



# **NVIDIA NVOS User Manual for NVLink Switches v25.02.2342**

# Table of Contents

<b>1</b>	<b>Quick Start Guide .....</b>	<b>24</b>
1.1	Prerequisites.....	24
1.2	Get Started .....	24
1.3	Manual Provisioning.....	25
1.3.1	Login Credentials .....	25
1.3.2	Serial Console Management .....	25
1.3.3	Wired Ethernet Management.....	26
1.3.4	Set Static IP Address.....	26
1.3.5	Configure the Hostname .....	26
1.3.6	Configure the System Clock .....	26
1.3.6.1	Manual Settings .....	26
1.3.6.1.1	Time Zone.....	26
1.3.6.1.2	Clock.....	27
1.3.6.2	NTP.....	27
1.3.6.2.1	NTP Servers from DHCP .....	27
1.3.6.2.2	Configure NTP Servers .....	27
1.3.7	Configure Syslog Servers.....	27
1.4	Test Cable Connectivity.....	28
<b>2</b>	<b>Installation Management.....</b>	<b>29</b>
2.1	Installing a New NVOS Image.....	29
2.1.1	Install Using a DHCP/Web Server With DHCP Options .....	30
2.1.2	Install Using a DHCP/Web Server Without DHCP Options .....	30
2.1.3	Install Using a Web Server With no DHCP .....	31
2.1.4	Install Using FTP Without a Web Server .....	31
2.1.5	Install Using a Local File .....	32
2.1.6	Installing using an image copied from the host machine .....	33
2.1.7	Install Using a USB Drive .....	33
2.1.7.1	Prepare for USB Installation.....	33
2.1.8	Initial login.....	35
2.1.9	Related Information .....	36
2.2	Upgrading/Downgrading NVOS Image.....	36
2.2.1	Upgrading Operating System Software .....	36
2.2.1.1	Important Notes Before Upgrading OS Software .....	36
2.2.1.2	Upgrading OS Software.....	36

2.2.2	Deleting Unused Images .....	37
2.3	Upgrading System Firmware .....	38
2.3.1	Importing Firmware and Changing the Default Firmware .....	38
2.3.1.1	Default Firmware Change .....	38
2.3.2	Upgrade Firmware .....	39
2.3.2.1	References .....	40
2.3.2.2	Bundles content and upgrade sequence.....	40
2.3.2.2.1	Bundles List .....	40
2.3.2.2.2	Upgrade Sequence .....	40
2.3.2.3	Component Image Update Using NVOS CLI.....	41
2.3.2.3.1	Transceiver Firmware Upgrade .....	41
2.3.2.4	Component Image Update Using RestAPI.....	42
2.3.2.4.1	REST API Commands .....	42
2.3.2.5	Recommended Full System Upgrade Sequence Example for Automation Reference .....	43
2.3.2.6	Error Status Catalog .....	44
2.4	Installation Management Commands.....	46
2.4.1	Version.....	47
2.4.1.1	nv show system version .....	47
2.4.2	Image .....	47
2.4.2.1	nv show system image .....	47
2.4.2.2	nv show system image files .....	48
2.4.2.3	nv action fetch system image.....	48
2.4.2.4	nv action install system image files.....	48
2.4.2.5	nv action rename system image files.....	49
2.4.2.6	nv action upload system image files.....	49
2.4.2.7	nv action uninstall system image.....	50
2.4.2.8	nv action delete system image files .....	50
2.4.2.9	nv action uninstall system image force .....	50
2.4.2.10	nv action boot-next system image .....	51
2.4.3	Firmware .....	51
2.4.3.1	nv show platform firmware .....	51
2.4.3.2	nv show platform firmware id .....	52
2.4.3.3	nv show platform firmware <erot-id> .....	52
2.4.3.4	nv show platform firmware files .....	53
2.4.3.5	nv set/unset platform firmware.....	53
2.4.3.6	nv action install platform firmware files.....	54
2.4.3.7	nv action delete platform firmware files .....	55
2.4.3.8	nv action rename platform firmware files.....	55
2.4.3.9	nv action fetch platform firmware.....	56

2.4.3.10	nv action upload platform firmware files .....	57
<b>3</b>	<b>NVIDIA User Experience (NVUE).....</b>	<b>58</b>
3.1	NVUE Object Model .....	58
3.2	NVUE CLI Overview .....	59
3.2.1	Command Syntax.....	59
3.2.2	Command Completion.....	60
3.2.3	Command Question Mark .....	60
3.2.4	Command Abbreviation .....	60
3.2.5	Command Help .....	60
3.2.6	Command List .....	61
3.2.7	Command History .....	61
3.2.8	Command Ranges .....	61
3.2.9	Command Categories.....	61
3.2.9.1	Configuration Commands .....	61
3.2.9.2	Monitoring Commands.....	62
3.2.9.3	Action Commands.....	64
3.2.9.4	Configuration Management Commands.....	64
3.2.9.5	List All NVUE Commands .....	65
3.2.10	Search for a Specific Configuration .....	66
3.2.11	Clear Switch Configuration .....	66
3.2.12	Example Configuration Commands.....	66
3.2.12.1	Configure the System Hostname.....	66
3.2.12.2	Configure an Interface .....	67
3.2.13	Example Monitoring Commands.....	67
3.2.13.1	Show Installed Software .....	67
3.2.13.2	Show Interface Configuration .....	67
3.2.14	Reset NVUE Configuration to Default Values.....	68
3.2.15	Add Configuration Apply Messages .....	68
3.2.16	Configure Auto Save .....	68
3.2.17	Example Configuration Management Commands.....	68
3.2.17.1	Apply and Save a Configuration.....	69
3.2.17.2	Detach a Pending Configuration .....	69
3.2.17.3	View Differences Between Configurations .....	69
3.2.17.4	Replace and Patch a Pending Configuration .....	70
3.2.18	Passwords and Special Characters.....	70
3.3	NVUE OpenAPI .....	73
3.3.1	Supported HTTP Methods .....	74

3.3.2	Secure the API .....	75
3.3.2.1	Certificates .....	75
3.3.2.1.1	Import a Certificate .....	75
3.3.2.1.2	Set the Certificate to Use .....	76
3.3.2.1.3	Delete Certificates .....	77
3.3.2.1.4	Show Certificate Information .....	77
3.3.2.2	Control Plane ACLs.....	77
3.3.3	Supported Objects .....	78
3.3.4	Use the API.....	78
3.3.4.1	API Port and Listening Address.....	79
3.3.4.1.1	Curl Command.....	79
3.3.4.2	Show NVUE REST API Information .....	79
3.3.4.3	Run cURL Commands .....	81
3.3.5	API Use Cases .....	81
3.3.5.1	View a Configuration .....	81
3.3.5.2	Replace an Entire Configuration.....	82
3.3.5.3	Make a Configuration Change .....	85
3.3.6	View Differences Between Configurations .....	88
3.3.7	Troubleshoot Configuration Changes.....	88
3.3.7.1	Configuration Apply Fails with Warnings .....	88
3.3.8	Save a Configuration .....	89
3.3.9	Unset a Configuration Change .....	89
3.3.10	Use the API for Active Monitoring .....	90
3.3.11	Retrieve View Types .....	91
3.3.12	Convert CLI Changes to Use the API.....	91
3.3.13	API Examples .....	93
3.3.13.1	Configure the System .....	94
3.3.13.2	Configure Services .....	95
3.3.13.3	Configure Users .....	97
3.3.13.4	Configure an Interface .....	99
3.3.13.5	Action Operations.....	100
3.3.14	Example Python Script .....	101
3.3.15	Try the API .....	103
3.3.16	Resources .....	103
3.3.17	Considerations.....	103
3.3.18	Related Information .....	104
3.3.19	Save the Applied Configurations.....	104

3.3.20	Send an Action.....	104
<b>4</b>	<b>System Management.....</b>	<b>105</b>
4.1	Access Control Lists .....	105
4.1.1	Firewall Rules .....	105
4.1.1.1	DoS Rules.....	105
4.1.1.2	Whitelist Rules .....	106
4.1.1.3	Unset the Default Firewall Rules .....	107
4.1.1.4	Add Firewall Rules .....	107
4.1.1.5	Show Firewall Rules .....	107
4.1.1.6	Log Messages .....	112
4.1.1.7	DoS Rules.....	112
4.1.1.8	Whitelist Rules .....	113
4.1.1.9	Unset the Default Firewall Rules .....	113
4.1.1.10	Add Firewall Rules .....	114
4.1.1.11	Show Firewall Rules .....	114
4.1.1.12	Log Messages .....	119
4.1.2	Access Control List Configuration .....	119
4.1.2.1	Traffic Rules .....	119
4.1.2.1.1	Chains.....	119
4.1.2.1.2	Rules .....	120
4.1.2.2	Install and Manage ACL Rules with NVUE .....	120
4.1.2.2.1	Rule Support .....	122
4.1.2.3	Default Action .....	122
4.1.2.4	Common Examples.....	122
4.1.2.4.1	Control Plane Limiters .....	122
4.1.2.4.2	Filter Specific TCP Flags .....	123
4.1.2.4.3	Control Who Can SSH into the Switch.....	124
4.1.2.4.4	Match on ECN Bits in the TCP IP Header .....	124
4.1.2.4.5	Set DSCP on Transit Traffic .....	125
4.1.3	Access Control List Commands.....	126
4.1.3.1	nv show acl .....	127
4.1.3.2	nv unset acl.....	128
4.1.3.3	nv show acl id .....	129
4.1.3.4	nv set/unset acl id.....	129
4.1.3.5	nv set/unset acl type .....	130
4.1.3.6	nv show acl rule.....	130
4.1.3.7	nv show acl rule id.....	130
4.1.3.8	nv set/unset acl rule .....	131
4.1.3.9	nv set/unset acl rule remark.....	131
4.1.3.10	nv show acl rule action .....	132

4.1.3.11	nv set/unset acl rule action permit.....	132
4.1.3.12	nv set/unset acl rule action deny .....	133
4.1.3.13	nv set/unset acl rule action log log-prefix.....	133
4.1.3.14	nv show acl rule match .....	134
4.1.3.15	nv set/unset acl rule match.....	134
4.1.3.16	nv show acl rule match ip .....	134
4.1.3.17	nv set/unset acl rule match ip.....	135
4.1.3.18	nv show acl rule match ip udp.....	135
4.1.3.19	nv show acl rule match ip udp dest-port.....	136
4.1.3.20	nv set/unset acl rule match ip udp dest-port .....	136
4.1.3.21	nv show acl rule match ip udp source-port .....	137
4.1.3.22	nv set/unset acl rule match ip udp source-port .....	137
4.1.3.23	nv show acl rule match ip tcp .....	138
4.1.3.24	nv show acl rule match ip tcp dest-port .....	138
4.1.3.25	nv set/unset acl rule match ip tcp dest-port.....	139
4.1.3.26	nv show acl rule match ip tcp source-port.....	139
4.1.3.27	nv set/unset acl rule match ip tcp source-port.....	140
4.1.3.28	nv show acl rule match ip tcp flags.....	140
4.1.3.29	nv set/unset acl rule match ip tcp flags .....	141
4.1.3.30	nv show acl rule match ip tcp mask .....	141
4.1.3.31	nv set/unset acl rule match ip tcp mask.....	142
4.1.3.32	nv set/unset acl rule match ip tcp mss .....	142
4.1.3.33	nv set/unset acl rule match ip tcp all-mss-except .....	143
4.1.3.34	nv set/unset acl rule match ip fragment.....	143
4.1.3.35	nv show acl rule match ip ecn .....	143
4.1.3.36	nv set/unset acl rule match ip ecn .....	144
4.1.3.37	nv set/unset acl rule match ip ecn ip-ect .....	144
4.1.3.38	nv set/unset acl rule match ip ecn flags.....	145
4.1.3.39	nv show acl rule match ip connection-state.....	145
4.1.3.40	nv set/unset acl rule match ip connection-state .....	145
4.1.3.41	nv show acl rule match ip extension-header .....	146
4.1.3.42	nv set/unset acl rule match ip extension-header type .....	146
4.1.3.43	nv show acl rule match ip routing-header .....	147
4.1.3.44	nv set/unset acl ACL rule match ip routing-header type .....	147
4.1.3.45	nv set/unset acl ACL rule match ip source-ip .....	147
4.1.3.46	nv set/unset acl ACL rule match ip dest-ip .....	148
4.1.3.47	nv set/unset acl rule match ip protocol.....	148
4.1.3.48	nv set/unset acl rule match ip icmp-type .....	149
4.1.3.49	nv set/unset acl rule match ip icmpv6-type.....	149
4.1.3.50	nv show acl rule match ip recent-list.....	150
4.1.3.51	nv set/unset acl rule match ip recent-list name.....	150

4.1.3.52	nv set/unset acl rule match ip recent-list action.....	151
4.1.3.53	nv set/unset acl rule match ip recent-list hit-count.....	151
4.1.3.54	nv set/unset acl rule match ip recent-list update-interval .....	152
4.1.3.55	nv show acl rule match ip hashlimit .....	152
4.1.3.56	nv set/unset acl rule match ip hashlimit name .....	153
4.1.3.57	nv set/unset acl rule match ip hashlimit rate-above .....	153
4.1.3.58	nv set/unset acl rule match ip hashlimit burst .....	154
4.1.3.59	nv set/unset acl rule match ip hashlimit expire .....	154
4.1.3.60	nv set/unset acl rule match ip hashlimit mode.....	154
4.1.3.61	nv set/unset acl rule match ip hashlimit destination-mask.....	155
4.1.3.62	nv set/unset acl rule match ip hashlimit source-mask.....	155
4.1.3.63	nv show interface acl .....	156
4.1.3.64	nv show interface acl id .....	156
4.1.3.65	nv show interface acl statistics.....	157
4.1.3.66	nv show interface acl statistics.....	157
4.1.3.67	nv show interface acl outbound .....	158
4.1.3.68	nv show interface acl outbound control-plane.....	158
4.1.3.69	nv show interface acl inbound.....	159
4.1.3.70	nv show interface acl inbound control-plane.....	159
4.1.3.71	nv set/unset interface acl inbound .....	160
4.1.3.72	nv set/unset interface acl inbound control-plane .....	160
4.1.3.73	nv set/unset interface acl outbound control-plane .....	160
4.1.3.74	nv set/unset interface acl outbound .....	161
4.1.3.75	nv action clear acl counters.....	161
4.1.3.76	nv set acl rule action set dscp .....	162
4.2	Attestation .....	162
4.2.1	SPDM .....	163
4.2.1.1	SPDM Commands .....	163
4.2.1.2	SPDM Commands .....	163
4.2.1.2.1	nv show system security spdm .....	163
4.2.1.2.2	nv show system security spdm certificates .....	164
4.2.1.2.3	nv show system security spdm measurements .....	164
4.2.1.2.4	nv action generate system security spdm.....	165
4.2.2	TPM .....	165
4.2.2.1	TPM Commands .....	165
4.2.2.2	TPM Commands .....	166
4.2.2.2.1	nv action generate system security tpm .....	166
4.2.2.2.2	nv action upload system security tpm.....	166
4.3	Authentication Authorization and Accounting .....	167
4.3.1	Authentication Order.....	167

4.3.2	Authentication Failthrough.....	167
4.3.3	User Accounts.....	168
4.3.3.1	First Login.....	168
4.3.3.2	Reset Local Users' Passwords.....	168
4.3.3.3	User Account Commands.....	169
4.3.3.4	User Account Commands.....	169
4.3.3.4.1	nv show system aaa user.....	169
4.3.3.4.2	nv show system aaa user id .....	170
4.3.3.4.3	nv show system aaa user ssh authorized-key .....	170
4.3.3.4.4	nv show system aaa user ssh authorized-key id.....	171
4.3.3.4.5	nv show system aaa user ssh .....	171
4.3.3.4.6	nv set/unset system aaa user ssh authorized-key .....	172
4.3.3.4.7	nv set/unset system aaa user .....	172
4.3.3.4.8	nv set/unset system aaa user full-name .....	173
4.3.3.4.9	nv set/unset system aaa user state .....	173
4.3.3.4.10	nv set/unset system aaa user role .....	174
4.3.3.4.11	nv set/unset system aaa user password.....	174
4.3.3.4.12	nv set/unset system aaa user hashed-password .....	175
4.3.3.4.13	nv set/unset system aaa allow-reset-local-passwords state .....	176
4.3.3.4.14	nv show system aaa allow-reset-local-passwords.....	176
4.3.4	LDAP Authentication and Authorization .....	176
4.3.4.1	Supported Features.....	176
4.3.4.2	LDAP Configuration .....	177
4.3.4.3	LDAP Groups and User Priviledges.....	177
4.3.4.3.1	LDAP Server Group Configuration Example .....	177
4.3.4.4	LDAP Secure Connection .....	177
4.3.4.5	LDAP Client Simple Configuration Example .....	178
4.3.4.6	LDAP Commands .....	178
4.3.4.7	LDAP Commands .....	178
4.3.4.7.1	nv show system aaa ldap .....	178
4.3.4.7.2	nv show system aaa ldap hostname.....	179
4.3.4.7.3	nv set system aaa ldap hostname .....	179
4.3.4.7.4	nv set system aaa ldap base-dn .....	180
4.3.4.7.5	nv set system aaa ldap bind-dn.....	180
4.3.4.7.6	nv set system aaa ldap port .....	180
4.3.4.7.7	nv set system aaa ldap timeout-bind .....	181
4.3.4.7.8	nv set system aaa ldap timeout-search .....	181
4.3.4.7.9	nv set system aaa ldap secret .....	181
4.3.4.7.10	nv set system aaa ldap map group cn .....	182
4.3.4.7.11	nv set system aaa ldap map group gidnumber .....	182
4.3.4.7.12	nv set system aaa ldap map group memberuid .....	182

4.3.4.7.13	nv set/unset system aaa ldap map passwd gidnumber .....	183
4.3.4.7.14	nv set system aaa ldap map passwd uid.....	183
4.3.4.7.15	nv set system aaa ldap map passwd uidnumber .....	183
4.3.4.7.16	nv set system aaa ldap map passwd userpassword .....	184
4.3.4.7.17	nv set system aaa ldap version .....	184
4.3.4.7.18	nv set system aaa ldap ssl mode .....	184
4.3.4.7.19	nv set system aaa ldap ssl cert-verify .....	185
4.3.4.7.20	nv set system aaa ldap ssl port .....	185
4.3.4.7.21	nv set system aaa ldap ssl ca-list.....	185
4.3.5	TACACS .....	186
4.3.5.1	Supported Features.....	186
4.3.5.2	TACACS Users .....	186
4.3.5.2.1	TACACS Server Setup and Usage Example .....	187
4.3.5.3	TACACS+ Accounting.....	187
4.3.5.3.1	TACACS Accounting Configuration .....	187
4.3.5.4	TACACS Commands .....	187
4.3.5.5	TACACS Commands .....	188
4.3.5.5.1	nv show system aaa tacacs accounting.....	188
4.3.5.5.2	nv show system aaa tacacs hostname .....	188
4.3.5.5.3	nv set system aaa tacacs accounting .....	189
4.3.5.5.4	nv set system aaa tacacs hostname.....	189
4.3.5.5.5	nv set system aaa tacacs port .....	189
4.3.5.5.6	nv set system aaa tacacs auth-type.....	190
4.3.5.5.7	nv set system aaa tacacs secret .....	190
4.3.5.5.8	nv set system aaa tacacs timeout.....	190
4.3.5.5.9	nv set system aaa tacacs hostname auth-port .....	191
4.3.5.5.10	nv set system aaa tacacs hostname auth-type .....	191
4.3.5.5.11	nv set system aaa tacacs hostname secret .....	192
4.3.5.5.12	nv set system aaa tacacs hostname priority.....	192
4.3.5.5.13	nv set system aaa tacacs hostname timeout .....	192
4.3.6	RADIUS.....	193
4.3.6.1	RADIUS Client.....	193
4.3.6.2	Radius Users .....	193
4.3.6.3	RADIUS Server Setup and Usage Example .....	193
4.3.6.3.1	Basic RADIUS Server Configuration .....	193
4.3.6.3.2	How To Set Up Basic FreeRADIUS Server .....	194
4.3.6.4	RADIUS Commands.....	194
4.3.6.5	RADIUS Commands.....	195
4.3.6.5.1	nv show system aaa radius .....	195
4.3.6.5.2	nv show system aaa radius hostname.....	195
4.3.6.5.3	nv show system aaa radius hostname.....	196

4.3.6.5.4	nv set system aaa radius hostname .....	196
4.3.6.5.5	nv set system aaa radius auth-port .....	197
4.3.6.5.6	nv set system aaa radius auth-type .....	197
4.3.6.5.7	nv set system aaa radius retransmit .....	197
4.3.6.5.8	nv set system aaa radius password.....	198
4.3.6.5.9	nv set system aaa radius statistics .....	198
4.3.6.5.10	nv set system aaa radius timeout .....	198
4.3.6.5.11	nv set system aaa radius hostname auth-port.....	199
4.3.6.5.12	nv set system aaa radius hostname auth-type .....	199
4.3.6.5.13	nv set system aaa radius hostname password .....	199
4.3.6.5.14	nv set system aaa radius hostname priority .....	200
4.3.6.5.15	nv set system aaa radius hostname retransmit .....	200
4.3.6.5.16	nv set system aaa radius hostname timeout.....	201
4.4	Certificates Management .....	201
4.4.1	Import Certificate .....	202
4.4.1.1	CA Certificate .....	202
4.4.1.2	Import Certificate .....	202
4.4.2	Set the Certificate to Use in NVUE REST API.....	203
4.4.3	Set the Certificate to Use for GNMI Service .....	204
4.4.4	Delete Certificates .....	204
4.4.5	Show Certificate Information .....	204
4.4.6	Certificate Management Commands .....	205
4.4.7	Certificate Management Commands .....	205
4.4.7.1	nv show system security ca-certificate.....	205
4.4.7.2	nv show system security certificate .....	205
4.4.7.3	nv action delete system security ca-certificate .....	206
4.4.7.4	nv action delete system security certificate .....	206
4.4.7.5	nv action import system security ca-certificate .....	207
4.4.7.6	nv action import system security certificate.....	207
4.5	Control and Power .....	208
4.5.1	Control and Power Commands .....	208
4.5.2	Control and Power Commands .....	209
4.5.2.1	nv show system reboot.....	209
4.5.2.2	nv show system reboot reason.....	209
4.5.2.3	nv show system reboot history.....	210
4.5.2.4	nv action reboot system .....	210
4.5.2.5	nv action power-cycle system .....	211
4.6	DNS Server .....	212
4.6.1	Supported Configurations and Limitations .....	212

4.6.2	DNS Server Commands .....	212
4.6.3	DNS Server Commands .....	212
4.6.3.1	nv show system dns.....	212
4.6.3.2	nv show system dns server .....	213
4.6.3.3	nv set/unset system dns server.....	213
4.7	Documentation .....	213
4.7.1	Documentation Commands .....	214
4.7.2	Documentation Commands .....	214
4.7.2.1	nv show system documentation .....	214
4.7.2.2	nv action upload system documentation files .....	214
4.8	Hostname .....	215
4.8.1	Hostname from DHCP .....	215
4.8.2	Static Hostname.....	215
4.8.3	Hostname Commands .....	215
4.8.4	Hostname Commands .....	215
4.8.4.1	nv set/unset system hostname .....	215
4.9	Management Interfaces .....	216
4.9.1	Configuring Management Interfaces with Static IP Addresses.....	216
4.9.2	Configuring IPv6 Address on the Management Interface .....	216
4.9.3	Default Gateway .....	217
4.9.4	IP DHCP Client.....	217
4.9.5	Management Interface Commands .....	217
4.9.5.1	nv show interface .....	218
4.9.5.2	nv show interface link.....	219
4.9.5.3	nv show interface ip.....	219
4.9.5.4	nv set/unset interface link state .....	220
4.9.5.5	nv set/unset interface description.....	221
4.9.5.6	nv set/unset interface ip .....	221
4.9.5.7	nv unset interface .....	222
4.9.5.8	nv set/unset interface link mtu .....	222
4.9.5.9	nv set/unset interface link speed .....	222
4.9.5.10	nv set/unset interface link duplex .....	223
4.9.5.11	nv set interface link auto-negotiate .....	223
4.9.5.12	nv set interface ip autoconf.....	224
4.9.5.13	nv set interface ip gateway .....	224
4.9.5.14	nv set interface ip arp-timeout.....	224
4.9.5.15	VRF .....	225
4.9.5.15.1	nv show vrf.....	225

4.9.5.15.2	nv set interface ip vrf .....	226
4.9.5.16	IP DHCP Client .....	227
4.9.5.16.1	nv set interface ip dhcp-client state .....	227
4.9.5.16.2	nv set interface ip dhcp-client set-hostname .....	227
4.9.5.16.3	nv set interface ip dhcp-client6 state .....	228
4.9.5.16.4	nv set interface ip dhcp-client6 set-hostname .....	228
4.9.5.16.5	nv action renew interface ip dhcp-client .....	229
4.9.5.16.6	nv action renew interface ip dhcp-client6 .....	229
4.10	Resource Management .....	229
4.10.1	Resource Management Commands .....	229
4.10.2	Resource Management Commands .....	230
4.10.2.1	nv show system .....	230
4.10.2.2	nv show system cpu .....	230
4.10.2.3	nv show system memory .....	231
4.11	Security .....	231
4.11.1	SSD Wipe .....	231
4.11.1.1	Get SSD PSID .....	231
4.11.1.2	Perform Disk Wipe .....	231
4.11.2	Recovery Flow After SSD Wipe .....	232
4.11.2.1	Prerequisites .....	232
4.11.2.2	Recovery Steps .....	232
4.11.3	Security Commands .....	233
4.11.4	Password Hardening .....	233
4.11.4.1	Password Hardening Commands .....	233
4.11.4.2	Password Hardening Commands .....	233
4.11.4.2.1	nv show system security password-hardening .....	234
4.11.4.2.2	nv set system security password-hardening state .....	234
4.11.4.2.3	nv set system security password-hardening digits-class .....	234
4.11.4.2.4	nv set system security password-hardening expiration .....	235
4.11.4.2.5	nv set system security password-hardening expiration-warning .....	235
4.11.4.2.6	nv set system security password-hardening history-cnt .....	236
4.11.4.2.7	nv set system security password-hardening len-min .....	236
4.11.4.2.8	nv set system security password-hardening lower-class .....	236
4.11.4.2.9	nv set system security password-hardening reject-user-passw- match .....	237
4.11.4.2.10	nv set system security password-hardening special-class .....	237
4.11.4.2.11	nv set system security password-hardening upper-class .....	237
4.11.5	SED Commands .....	238
4.11.5.1	nv action change system security sed-password .....	238
4.12	System API .....	238

4.12.1	Key Functionalities .....	238
4.12.2	System API Commands .....	239
4.12.3	System API Commands .....	239
4.12.3.1	nv show system api .....	239
4.12.3.2	nv set/unset system api state .....	239
4.12.3.3	nv set/unset system api port .....	240
4.12.3.4	nv show system api mtls .....	240
4.12.3.5	nv set system api mtls ca-certificate .....	240
4.12.3.6	nv set system api certificate .....	241
4.13	Time Synchronization .....	241
4.13.1	Date and Time .....	241
4.13.1.1	Time Zone .....	241
4.13.1.2	Clock .....	242
4.13.1.3	Date and Time Commands .....	242
4.13.1.4	Date and Time Commands .....	242
4.13.1.4.1	nv action change system date-time .....	242
4.13.1.4.2	nv set/unset system timezone .....	242
4.13.2	NTP .....	243
4.13.2.1	NTP Authenticate .....	243
4.13.2.2	NTP Authentication Key .....	243
4.13.2.3	NTP Commands .....	244
4.13.2.4	NTP Commands .....	244
4.13.2.4.1	nv show system ntp .....	244
4.13.2.4.2	nv show system ntp server .....	245
4.13.2.4.3	nv show system ntp key .....	245
4.13.2.4.4	nv set/unset system ntp .....	246
4.13.2.4.5	nv set/unset system ntp server .....	247
4.13.2.4.6	nv set/unset system ntp key .....	248
4.14	User Interfaces .....	248
4.14.1	Secure Shell (SSH) for Remote Access .....	249
4.14.1.1	Overview .....	249
4.14.1.2	Configure Timeouts and Sessions .....	249
4.14.1.2.1	Message of the Day .....	250
4.14.1.2.2	Generate and Install an SSH Key Pair .....	251
4.14.1.2.3	Troubleshooting .....	252
4.14.1.3	SSH Commands .....	252
4.14.1.3.1	show ssh-server .....	253
4.14.1.3.2	nv set/unset system ssh-server inactive-timeout .....	253
4.14.1.3.3	nv set/unset system ssh-server max-sessions .....	253
4.14.1.3.4	nv set/unset system ssh-server port .....	254

4.14.2	User Interface Commands .....	254
4.14.2.1	System Message .....	255
4.14.2.1.1	nv show system message.....	255
4.14.2.1.2	nv set/unset system message pre-login message .....	255
4.14.2.1.3	nv set/unset system message post-login message .....	256
4.14.2.1.4	nv set/unset system message post-logout .....	256
4.14.2.2	SSH .....	256
4.14.2.2.1	show ssh-server .....	256
4.14.2.2.2	nv set/unset system ssh-server inactive-timeout.....	257
4.14.2.2.3	nv set/unset system ssh-server max-sessions .....	257
4.14.2.2.4	nv set/unset system ssh-server port .....	258
4.14.2.3	Serial-Console .....	258
4.14.2.3.1	nv show system serial-console .....	258
4.14.2.3.2	nv set/unset system serial-console inactivity-timeout .....	258
4.14.2.3.3	nv set/unset system serial-console sysrq-capabilities .....	259
4.15	Zero-Touch Provisioning.....	259
4.15.1	Running DHCP-ZTP .....	260
4.15.2	Dynamic Content Configuration .....	261
4.15.2.1	DHCP Options.....	261
4.15.2.2	HTTP Headers .....	262
4.15.3	ZTP Configuration.....	262
4.15.3.1	ZTP Configuration File .....	262
4.15.3.2	ZTP Configuration image .....	263
4.15.3.3	ZTP Configuration startup-file .....	264
4.15.3.4	ZTP Configuration commands-list .....	265
4.15.3.5	ZTP Configuration connectivity-check.....	265
4.15.3.6	ZTP Configuration provisioning-script .....	266
4.15.3.7	ZTP Configuration nmx-commands-list .....	266
4.15.4	ZTP and OS Upgrade .....	267
4.15.5	Example Configurations.....	267
4.15.5.1	DHCPv4 Configuration .....	267
4.15.5.2	DHCPv6 Configuration .....	267
4.15.5.3	ZTP Configuration File Example .....	267
4.15.5.4	Commands List Example .....	268
4.15.5.5	YAML Formatted File Configuration .....	268
4.15.5.6	Provisioning Script Example .....	268
4.15.6	Zero-Touch Provisioning Commands.....	268
4.15.7	Zero-Touch Provisioning Commands.....	268
4.15.7.1	nv show system ztp .....	268
4.15.7.2	nv set system ztp config-save .....	269

4.15.7.3	nv action system ztp .....	270
<b>5</b>	<b>Chassis Management.....</b>	<b>271</b>
5.1	System Health Monitor .....	271
5.2	Leakage Sensors.....	271
5.3	Chassis Management Commands.....	271
5.4	Chassis Information and Inventory .....	272
5.4.1	Displaying Cable Cartridge (CBC) EEPROM .....	272
5.4.2	Chassis Information and Inventory Commands .....	272
5.4.3	Chassis Information and Inventory Commands .....	272
5.4.3.1	nv show platform .....	272
5.4.3.2	nv show platform chassis-location .....	273
5.4.3.3	nv show platform inventory .....	273
5.4.3.4	nv action reset platform bmc-password.....	274
5.4.3.5	nv show platform cable-cartridge .....	274
5.4.3.6	nv show platform cable-cartridge name .....	274
5.5	Environment .....	275
5.5.1	Key Functionalities .....	275
5.5.2	Environment Commands .....	275
5.5.3	Environment Commands .....	275
5.5.3.1	nv show platform environment .....	276
5.5.3.2	nv show platform environment fan .....	277
5.5.3.3	nv show platform environment led.....	277
5.5.3.4	nv show platform environment temperature .....	278
5.5.3.5	nv show platform environment voltage .....	279
5.5.3.6	nv show platform environment leakage .....	279
5.5.3.7	nv action turn-on/turn-off platform environment led UID .....	280
5.6	Software.....	280
5.6.1	Software Commands.....	280
5.6.2	Software Commands.....	280
5.6.2.1	nv show platform software installed.....	280
5.7	Transceiver.....	281
5.7.1	Key Functionalities .....	281
5.7.2	How to Install Transceiver Firmware.....	281
5.7.3	Transceiver Commands.....	282
5.7.4	Transceiver Commands.....	282
5.7.4.1	nv show platform transceiver .....	282
5.7.4.2	nv action reset platform transceiver id .....	286

5.7.4.3	nv action install platform transceiver firmware files .....	286
5.7.4.4	nv show platform transceiver firmware .....	287
5.7.4.5	nv show platform transceiver firmware files .....	287
5.7.4.6	nv show platform transceiver firmware files name .....	287
<b>6</b>	<b>Configuration Management.....</b>	<b>289</b>
6.1	Managing Configurations as Revisions.....	289
6.2	Restoring Factory Default Configuration .....	290
6.3	Configuration Management Commands.....	290
6.4	Configuration Management Commands.....	290
6.4.1	Config.....	291
6.4.1.1	nv config apply .....	291
6.4.1.2	nv config detach .....	292
6.4.1.3	nv config diff .....	292
6.4.1.4	nv config find .....	292
6.4.1.5	nv config history .....	293
6.4.1.6	nv config patch.....	293
6.4.1.7	nv config replace .....	294
6.4.1.8	nv config save .....	294
6.4.1.9	nv config show.....	294
6.4.1.10	nv show system config files .....	295
6.4.1.11	nv show system config files .....	295
6.4.1.12	nv show system config files brief.....	296
6.4.1.13	nv action export system config .....	296
6.4.1.14	nv action rename system config files.....	296
6.4.1.15	nv action delete system config files .....	297
6.4.1.16	nv action delete system config files .....	297
6.4.1.17	nv action upload system config files.....	297
6.4.1.18	nv action fetch system config files.....	298
6.4.2	Factory .....	298
6.4.2.1	nv action reset system factory-default.....	298
<b>7</b>	<b>Telemetry Streaming .....</b>	<b>300</b>
7.1	gNMI Streaming.....	300
7.1.1	Configure the gNMI Agent using NVUE CLI Commands.....	300
7.1.2	Supported Subscription Modes .....	300
7.1.2.1	Supported Models .....	301
7.1.3	gNMI Client Requests .....	302
7.1.4	Related Information .....	304
7.1.5	gNMI Streaming Commands.....	304

7.1.6	gNMI Streaming Commands.....	304
7.1.6.1	nv show system gnmi-server .....	304
7.1.6.2	nv set system gnmi-server state .....	304
7.1.6.3	nv unset system gnmi-server state .....	305
7.1.6.4	nv set sys gnmi-server certificate .....	305
7.2	SNMP .....	305
7.2.1	Standard MIBs.....	306
7.2.2	Configuring SNMP .....	306
7.2.3	SNMP Commands .....	306
7.2.4	SNMP Commands .....	306
7.2.4.1	nv show system snmp-server .....	307
7.2.4.2	nv show system snmp-server listening-address .....	307
7.2.4.3	nv show system snmp-server readonly-community .....	307
7.2.4.4	nv set system snmp-server state .....	308
7.2.4.5	nv set system snmp-server listening-address .....	308
7.2.4.6	nv set system snmp-server readonly-community .....	309
7.2.4.7	nv set system snmp-server auto-refresh-interval .....	309
7.2.4.8	nv set system snmp-server system-contact .....	309
7.2.4.9	nv set system snmp-server system-location .....	310
<b>8</b>	<b>Monitoring and Diagnostics .....</b>	<b>311</b>
8.1	Health Monitoring.....	311
8.1.1	Service Monitoring .....	311
8.1.2	Hardware Monitoring .....	311
8.1.3	Output Examples.....	311
8.1.4	Health Monitoring Commands.....	312
8.1.5	Health Monitoring Commands.....	312
8.1.5.1	nv show system health .....	312
8.1.5.2	nv show system health history .....	313
8.2	Logging .....	313
8.2.1	Logging .....	313
8.2.2	Logging Commands .....	313
8.2.3	Logging Commands .....	313
8.2.3.1	nv show system log .....	314
8.2.3.2	nv show system rotation .....	315
8.2.3.3	nv set/unset system log component .....	315
8.2.3.4	nv set/unset system rotation disk-percentage .....	316
8.2.3.5	nv set/unset system rotation frequency .....	316
8.2.3.6	nv set/unset system rotation max-number .....	316

8.2.3.7	nv set/unset system rotation size .....	317
8.2.3.8	nv set/unset system syslog trap .....	317
8.2.3.9	nv action rotate system.....	318
8.3	Remote Logging .....	318
8.3.1	Remote Logging Commands .....	318
8.3.2	Remote Logging Commands .....	319
8.3.2.1	nv show system syslog.....	319
8.3.2.2	nv show system syslog format .....	319
8.3.2.3	nv show system syslog server .....	320
8.3.2.4	nv show system syslog server id .....	320
8.3.2.5	nv unset system syslog .....	321
8.3.2.6	nv set/unset system syslog format.....	321
8.3.2.7	nv set/unset system syslog server filter exclude .....	321
8.3.2.8	nv set/unset system syslog server filter include .....	322
8.3.2.9	nv set/unset system syslog format welf firewall-name.....	322
8.3.2.10	nv unset system syslog server.....	323
8.3.2.11	nv set/unset system syslog server.....	323
8.3.2.12	nv set/unset system syslog server trap.....	323
8.3.2.13	nv set/unset system syslog server port.....	324
8.3.2.14	nv set/unset system syslog server protocol.....	324
8.3.2.15	nv set/unset system syslog server vrf .....	325
8.4	Link Diagnostic Per Port.....	325
8.4.1	Link Diagnostic Commands .....	327
8.4.2	Link Diagnostic Commands .....	327
8.4.2.1	nv show interface link diagnostics .....	327
8.4.2.2	nv show interface view link diagnostics.....	327
8.5	Event Management .....	328
8.5.1	Supported Events .....	328
8.5.2	Detailed Table of Events.....	329
8.5.3	Event Management Commands .....	334
8.5.4	Event Management Commands .....	334
8.5.4.1	nv show system events.....	334
8.5.4.2	nv show system events last.....	335
8.5.4.3	nv show system events recent .....	335
8.5.4.4	nv set/unset system events table-size .....	336
8.5.4.5	nv action clear system events .....	336
8.6	Statistics .....	336
8.6.1	Statistics Commands .....	337
8.6.2	Statistics Commands .....	337

8.6.2.1	nv show system stats .....	338
8.6.2.2	nv show system stats category.....	338
8.6.2.3	nv show system stats files .....	339
8.6.2.4	nv set/unset system stats state .....	339
8.6.2.5	nv set/unset system stats category state .....	340
8.6.2.6	nv set/unset system stats category interval.....	340
8.6.2.7	nv set/unset system stats category history-duration.....	341
8.6.2.8	nv action clear system stats category .....	341
8.6.2.9	nv action clear system stats category .....	341
8.6.2.10	nv action generate system stats category .....	342
8.6.2.11	nv action generate system stats.....	342
8.6.2.12	nv action clear system stats.....	342
8.6.2.13	nv action delete system stats files .....	343
8.6.2.14	nv action upload system stats files .....	343
8.7	Technical Support .....	344
8.7.1	How to Generate and Upload a Tech-Support File.....	344
8.7.2	Technical Support Commands .....	344
8.7.3	Technical Support Commands .....	344
8.7.3.1	nv show system tech-support files .....	344
8.7.3.2	nv action generate system tech-support .....	345
8.7.3.3	nv action delete system tech-support files .....	345
8.7.3.4	nv action upload system tech-support files.....	346
8.8	Troubleshooting .....	346
8.8.1	Resetting NVOS Password .....	346
8.8.2	Resetting BMC Root User Password.....	346
8.8.3	Image Upgrade Recovery .....	347
8.8.4	System Fatal Recovery .....	347
8.8.4.1	Detecting a Fatal State .....	347
8.8.4.2	Automatic Recovery Mechanism .....	347
8.8.4.3	Recovery Timeframe.....	348
<b>9</b>	<b>NVLink Switching.....</b>	<b>349</b>
9.1	NVLink Interface .....	349
9.1.1	Configuring and Monitoring Interfaces.....	349
9.1.2	NVLink Interface Commands .....	349
9.1.3	NVLink Interface Commands .....	350
9.1.3.1	nv show interface .....	350
9.1.3.2	nv show interface link.....	351
9.1.3.3	nv set interface description.....	353

9.1.3.4	nv set interface link state .....	354
9.1.3.5	nv action clear interface link counters.....	354
9.1.3.6	nv action clear interface counters .....	355
9.2	NVLink Fabric.....	355
9.2.1	NVLink Fabric Commands.....	355
9.2.2	NVLink Fabric Commands.....	355
9.2.2.1	show ib device.....	355
9.3	Cluster Management.....	356
9.3.1	Cluster Infrastructure and Cluster Applications.....	356
9.3.1.1	Installation and Upgrade.....	356
9.3.1.2	Management and Configuration .....	356
9.3.1.3	Persistence and Recovery.....	356
9.3.2	NMX-Controller .....	357
9.3.3	NMX-Telemetry .....	357
9.3.3.1	Key Features.....	357
9.3.4	Cluster Provisioning Flow .....	358
9.3.5	Partition Management.....	358
9.3.6	Protocol Buffers .....	358
9.3.7	Cluster Control .....	358
9.3.7.1	Chassis ID Update .....	358
9.3.7.2	Cluster Control Commands .....	358
9.3.7.3	Cluster Control Commands .....	359
9.3.7.3.1	nv show cluster.....	359
9.3.7.3.2	nv set/unset cluster state .....	359
9.3.7.3.3	nv action update cluster chassis-id .....	360
9.3.8	Cluster Applications .....	360
9.3.8.1	Key Functionalities .....	360
9.3.8.2	Cluster Applications Commands .....	361
9.3.8.3	Cluster Applications Commands .....	361
9.3.8.3.1	nv show cluster apps .....	361
9.3.8.3.2	nv show cluster apps name.....	362
9.3.8.3.3	nv show cluster apps installed .....	362
9.3.8.3.4	nv show cluster apps running .....	362
9.3.8.3.5	nv action start cluster apps .....	363
9.3.8.3.6	nv action stop cluster apps.....	363
9.3.8.3.7	nv show cluster apps log-level.....	364
9.3.8.3.8	nv action update cluster apps log-level .....	364
9.3.8.3.9	nv action restore cluster apps log-level.....	364
9.3.9	Cluster Manager .....	365

9.3.9.1	How to Use Cluster Manager (Non-Secure).....	365
9.3.9.2	Cluster Manager Security .....	365
9.3.9.2.1	Flow Description .....	366
9.3.9.2.2	Configuration for Enabling mTLS with Cluster Manager .....	366
9.3.9.2.3	Configuration for Enabling TLS with Cluster Manager .....	367
9.3.9.3	Cluster Manager Commands.....	367
9.3.9.4	Cluster Manager Commands.....	367
9.3.9.4.1	nv show cluster apps manager .....	367
9.3.9.4.2	nv show cluster apps manager encryption .....	368
9.3.9.4.3	nv show cluster apps manager certificate .....	368
9.3.9.4.4	nv show cluster apps manager ca-certificate .....	369
9.3.9.4.5	nv action update cluster apps manager .....	369
9.3.9.4.6	nv action update cluster apps manager certificate .....	369
9.3.9.4.7	nv action update cluster apps manager ca-certificate.....	370
9.3.9.4.8	nv action update cluster apps manager encryption .....	370
9.3.9.4.9	nv action restore cluster apps manager encryption .....	371
9.3.9.4.10	nv action restore cluster apps manager .....	371
9.3.9.4.11	nv action restore cluster apps manager certificate .....	372
9.3.9.4.12	nv action restore cluster apps manager ca-certificate .....	372
9.3.10	SDN.....	373
9.3.10.1	Key Functionalities .....	373
9.3.10.2	SDN Configuration and State Management.....	373
9.3.10.2.1	SDN Configuration .....	373
9.3.10.2.2	SDN State.....	373
9.3.10.3	SDN Partitions Management .....	374
9.3.10.3.1	Creating SDN Partitions .....	374
9.3.10.3.2	Updating SDN Partitions.....	374
9.3.10.4	SDN Reset to Factory Defaults .....	375
9.3.10.5	SDN Commands.....	375
9.3.10.6	SDN Commands.....	375
9.3.10.6.1	SDN Configuration and State Files Management .....	376
9.3.10.6.2	SDN Partition Management .....	381
9.3.10.6.3	SDN Reset .....	386
<b>10</b>	<b>Document Revision History .....</b>	<b>387</b>

## Welcome to NVOS Documentation

NVIDIA NVOS operating system enables the management and configuration of NVIDIA's switch system platforms.

NVOS provides a suite of management options, incorporates a CLI and OpenAPI, which enables administrators to easily configure and manage the system.

These pages provide information about the scope, organization, and command-line interface of NVOS as well as configuration examples.

### Document Revision History

A list of the changes made to the User Manual is provided in [Document Revision History](#) section.

---

# 1 Quick Start Guide

- [1.1 Prerequisites](#)
- [1.2 Get Started](#)
- [1.3 Manual Provisioning](#)
  - [1.3.1 Login Credentials](#)
  - [1.3.2 Serial Console Management](#)
  - [1.3.3 Wired Ethernet Management](#)
  - [1.3.4 Set Static IP Address](#)
  - [1.3.5 Configure the Hostname](#)
  - [1.3.6 Configure the System Clock](#)
    - [1.3.6.1 Manual Settings](#)
      - [1.3.6.1.1 Time Zone](#)
      - [1.3.6.1.2 Clock](#)
    - [1.3.6.2 NTP](#)
      - [1.3.6.2.1 NTP Servers from DHCP](#)
      - [1.3.6.2.2 Configure NTP Servers](#)
  - [1.3.7 Configure Syslog Servers](#)
- [1.4 Test Cable Connectivity](#)

This quick start guide provides an end-to-end setup process for installing and running NVOS.

 Before running NVUE commands, please refer to the [NVIDIA User Experience \(NVUE\)](#) section.

 NVOS supports configurations through NVUE only. Running Linux commands or directly modifying underlying Linux files might result in undefined behavior.

## 1.1 Prerequisites

This guide requires an intermediate-level Linux knowledge. An understanding of text editing, Unix file permissions, and process monitoring is necessary. A variety of text editors are pre-installed, including `vi` and `nano`.

Access to a Linux or UNIX shell is needed. For Windows users, employing a Linux environment like Cygwin as your command line tool is recommended for interacting with NVOS.

 Please align the firmware of all platform components (e.g., CPLD, BMC) before provisioning the switch. Refer to the NVIDIA NVOS Release Notes for the exact firmware versions in the package.

## 1.2 Get Started

NVOS provides two methods for initial provisioning:

- [Zero-Touch Provisioning \(ZTP\)](#)

- [Manual Provisioning](#)

## 1.3 Manual Provisioning

When starting NVOS for the first time, the management port send a DHCP request. To determine the IP address of the switch, you can cross reference the MAC address of the switch with your DHCP server. The MAC address is typically located on the side of the switch or on the box in which the unit ships.

### 1.3.1 Login Credentials

The default installation includes two accounts:

- The user account (admin) has `sudo` privileges. The admin account uses the default password `admin`.
- The user account (monitor) has `read only` privileges. The monitor account uses the default password `monitor`.

Upon first login, it will be required to change the default passwords for the admin and monitor accounts. The new password must comply with the default password hardening rules (see [Password Hardening](#) section).

The new configured password is treated like any other configuration. If the configuration is not saved, the user will need to reconfigure it upon the system's next boot.

```
You are required to change your password immediately (administrator enforced).
```

```
WARNING: Your password has expired.  
You must change your password now!  
New password:  
Retype new password:
```

In this quick start guide, use the *admin* account to configure NVOS.

All accounts except root can use remote SSH login. Note that the default SSH auto-logout is 15 minutes.

NVOS supports multiple services to authenticate users and authorize switch tasks, both local and remote authentication/authorization methods.

For more information, please refer to [Authentication Authorization and Accounting](#) section.

### 1.3.2 Serial Console Management

It is recommended to perform management and configuration over the network, either in-band or out-of-band. A serial console is fully supported.

Typically, switches ship from the manufacturer with a mating DB9 serial cable. Switches with ONIE are always set to a 115200 baud rate.

Recommended Serial Connection Settings

Parameter	Setting
Baud Rate	115200
Data bits	8
Stop bits	1
Parity	None
Flow Control	None

### 1.3.3 Wired Ethernet Management

An NVOS switch always provides two dedicated Ethernet management port called eth0 and eth1. This interface is specifically for out-of-band management use. The management interface uses DHCP for addressing by default.

### 1.3.4 Set Static IP Address

To set a static IP address, run the following (ipv6 supported as well):

```
admin@nvos:~$ nv set interface eth0 ip address 192.0.2.42/24
admin@nvos:~$ nv set interface eth0 ip gateway 192.0.2.1
admin@nvos:~$ nv config apply
```



Configuring static IP address will trigger unsolicited announcement messages to the gateway in order to reveal the device's MAC address.

### 1.3.5 Configure the Hostname

The hostname identifies the switch; as such, make sure the hostname is configured in a unique and descriptive way. For more information, see [Hostname](#) section.

### 1.3.6 Configure the System Clock

NVOS relies on the system clock for displaying the time and timestamping messages. User can control the system time zone and clock settings.

#### 1.3.6.1 Manual Settings

##### 1.3.6.1.1 Time Zone

Default time zone is set to Coordinated Universal Time (UTC). User may change the time zone configuration by executing the following:

```
admin@nvos:~$ nv set system timezone Etc/UTC
admin@nvos:~$ nv config apply
```

### 1.3.6.1.2 Clock

The system date and time can be manually changed. NTP servers configured on the switch will supersede any manually entered date-time settings.

```
admin@nvos:~$ nv action change system date-time 2024-12-24 10:25:13
```

### 1.3.6.2 NTP

NVOS supports [Network Time Protocol](#) (NTP) for synchronizing the time of the system.

#### 1.3.6.2.1 NTP Servers from DHCP

By default, NTP will obtain the NTP servers from DHCP. It is possible to disable it by executing the following:

```
admin@nvos:~$ nv set system ntp dhcp disabled
admin@nvos:~$ nv config apply
```

#### 1.3.6.2.2 Configure NTP Servers

A new NTP server can be added by executing the following:

```
admin@nvos:~$ nv set system ntp server 10.11.100.5
admin@nvos:~$ nv config apply
```

NTP can be configured to accept packets coming from an authenticated source only.

NTP authentication is disabled by default.

An authentication key may be created and used to authenticate incoming NTP packets. To ensure the key is utilized, the following conditions must be met:

- Enable NTP authentication

```
admin@nvos:~$ nv set system ntp authentication enabled
admin@nvos:~$ nv config apply
```

- Add a new NTP key, and associate it with the server:

```
admin@nvos:~$ nv set system ntp key 15 value SECRET
admin@nvos:~$ nv set system ntp key 15 trusted yes
admin@nvos:~$ nv set system ntp server 10.11.100.5 key 15
admin@nvos:~$ nv config apply
```

- NTP server must use the same key.

NVOS support various attributes for NTP servers and keys, for more information please see [NTP Commands](#).

### 1.3.7 Configure Syslog Servers

User can configure the system to send syslog logs to a centralized logging server ( one or more):

```
admin@nvos:~$ nv set system syslog server 10.20.30.40
admin@nvos:~$ nv config apply
```

For more syslog configurations, please see [Remote Logging](#) section.

## 1.4 Test Cable Connectivity

By default, NVOS enables all data plane ports (every NVLink port). To view link status, run the `nv show interface` command or `nv show platform transceiver`.



When running NVUE commands to configure the switch, run the `nv config save` command before you reboot. The command saves the applied configuration to the startup configuration so that the changes persist after the reboot.

```
admin@nvos:~$ nv config save
```

---

## 2 Installation Management

This section describes how to manage, install, and upgrade NVOS.

- [Installing a New NVOS Image](#)
- [Upgrading/Downgrading NVOS Image](#)
- [Upgrading System Firmware](#)
- [Installation Management Commands](#)

### 2.1 Installing a New NVOS Image

 The default password for the *admin* user account is `admin`.

 It is recommended to change the default password when logging in for the first time. ONIE includes options that allow to change the default password for the *admin* account automatically when installing a new NVOS image. Refer to [ONIE Installation Options](#).

A new NVOS image can be installed using ONIE—an open source project (equivalent to PXE on servers)—that enables the installation of network operating systems (NOS) on bare metal switches.

Before installing NVOS, the switch may be in one of the two following states:

1. The switch does not contain an image (the switch is only running ONIE).
2. NVOS is already on the switch, but NVUE commands are to be used to reinstall NVOS or upgrade to a newer version.

The sections below describe some of the different ways to install the NVOS image. Steps show how to install directly from ONIE (if no image is on the switch) and from NVOS (if the image is already on the switch). For additional methods to find and install the NVOS image, see the [ONIE Design Specification](#).

To get into ONIE, you need to interrupt the GRUB countdown screen by pressing the "ESC" or "F4" key and choose the appropriate menu entry.

The NVOS image can be downloaded from the [NVIDIA Enterprise Support Portal](#).

 Installing the NVOS image is destructive. Configuration files on the switch are not saved, so copy them to a different server before installation.

In the following procedures, the following is possible:

- The NVOS image can be named using any of the [ONIE naming schemes](#) mentioned here
- The `sudo onie-install -h` command can be run to show the ONIE installer options

## 2.1.1 Install Using a DHCP/Web Server With DHCP Options

To install NVOS using a DHCP or web server *with* DHCP options, set up a DHCP/web server on a laptop and connect the eth0 management port of the switch to the laptop. After connecting the cable, the installation proceeds as follows:

1. The switch boots up and requests an IP address (DHCP request).
2. The DHCP server acknowledges and responds with DHCP option 114 and the location of the installation image.
3. ONIE downloads the NVOS image, installs, and reboots.  
NVOS should now be running.



**⚠** The most common way is to send DHCP option 114 with the entire URL to the web server (this can be the same system). However, there are other ways to use DHCP even if you do not have full control over DHCP. See the ONIE user guide for information on [partial installer URLs](#) and [advanced DHCP options](#)—both articles list more supported DHCP options.

Example DHCP configuration with an ISC DHCP server:

```
subnet 172.0.24.0 netmask 255.255.255.0 {
    range 172.0.24.20 172.0.24.200;
    option default-url = "http://172.0.24.14/onie-installer-x86_64";
}
```

Example DHCP configuration with dnsmasq (static address assignment):

```
dhcp-host=sw4,192.168.100.14,6c:64:1a:00:03:ba,set:sw4
dhcp-option=tag:sw4,114,"http://roz.rtplab.test/onie-installer-x86_64"
```

If a web server is not accessible, [this free Apache example](#) can be used.

## 2.1.2 Install Using a DHCP/Web Server Without DHCP Options

Follow the steps below if logging into the switch on a serial console (ONIE), or log in on the console or with SSH (Install from NVOS).

Install from ONIE

1. Place the NVOS image in a directory on the web server.

2. Run the `onie-nos-install` command:

```
ONIE:/ #onie-nos-install http://10.0.1.251/path/to/nvos-amd64-25.01.0002.bin
```

### 2.1.3 Install Using a Web Server With no DHCP

Follow the steps below if logging into the switch on a serial console (ONIE), or log in on the console or with SSH (Install from NVOS) but *no* DHCP server is available.



A console connection is needed to access the switch. This procedure cannot be performed remotely.

Install from ONIE

1. ONIE is in **discovery mode**. Disable discovery mode with the following command:

```
onie# onie-discovery-stop
```

On older ONIE versions, if the `onie-discovery-stop` command is not supported, run the following command:

```
onie# /etc/init.d/discover.sh stop
```

2. Assign a static address to `eth0` with the `ip addr add` command:

```
ONIE:/ #ip addr add 10.0.1.252/24 dev eth0
```

3. Place the NVOS image in a directory on your web server.
4. Run the installer manually (because there are no DHCP options):

```
ONIE:/ #onie-nos-install http://10.0.1.251/path/to/nvos-install-x86_64.bin
```

### 2.1.4 Install Using FTP Without a Web Server

Follow the steps below if your laptop is on the same network as the switch `eth0` interface but *no* DHCP server is available.

Install from ONIE

1. Set up DHCP or static addressing for `eth0`. The following example assigns a static address to `eth0`:

```
ONIE:/ #ip addr add 10.0.1.252/24 dev eth0
```

2. If static addressing is being used, disable ONIE discovery mode:

```
onie# onie-discovery-stop
```

On older ONIE versions, if the `onie-discovery-stop` command is not supported, run the following:

```
onie# /etc/init.d/discover.sh stop
```

3. Place the NVOS image into a TFTP or FTP directory.
4. If DHCP options are not being used, run one of the following commands (tftp for TFTP or ftp for FTP):

```
ONIE# onie-nos-install ftp://local-ftp-server/nvos-amd64-25.01.0002.bin  
ONIE# onie-nos-install tftp://local-tftp-server/nvos-amd64-25.01.0002.bin
```

## 2.1.5 Install Using a Local File

Follow the steps below to install the NVOS image referencing a local file.

Install from ONIE

1. Set up DHCP or static addressing for eth0. The following example assigns a static address to eth0:

```
ONIE:/ #ip addr add 10.0.1.252/24 dev eth0
```

2. If you are using static addressing, disable ONIE discovery mode.

```
onie# onie-discovery-stop
```

On older ONIE versions, if the `onie-discovery-stop` command is not supported, run the following:

```
onie# /etc/init.d/discover.sh stop
```

3. Use `scp` to copy the NVOS image to the switch.
4. Run the installer manually from ONIE:

```
ONIE:/ #onie-nos-install /path/to/local/file/nvos-amd64-25.01.0002.bin
```

Install from NVOS

1. Fetch the NVOS image on the switch.

```
admin@nvos:~$ nv action fetch system image scp://<username>:<password>@<ip-address>/var/www/html/  
<new_image>
```

```
admin@nvos:~$ nv show system image files
Available image file
-----
nvos-amd64-25.01.0003.bin
```

2. From the NVOS command prompt, using NVUE command, install the new image and follow the instruction on the screen.

```
admin@nvos:~$ nv action install system image files nvos-amd64-25.01.0003.bin
Operation will reboot the system.
If you choose 'y', the system will install the image and reboot.
If you choose 'N', the operation will abort without installing the image and without rebooting the system.
Do you want to continue? [y/N]
```

3. After reboot, run `nv show system image` to review your images.

```
admin@nvos:~$ nv show system image --view detail
-----
operational      applied      description
-----
current          nvos-25.01.0003      Current running image
next             nvos-25.01.0003      Next image to boot from
partition1       nvos-25.01.0002      Image installed on partition 1
partition2       nvos-25.01.0003      Image installed on partition 2
```

## 2.1.6 Installing using an image copied from the host machine

1. Copy the image from the host machine to the switch.

 **Note:** you must copy an image to the predefined directory: `"/host/nos-images/"`

```
user@host:~$ scp <path-to-system-image> <switch-admin-username>@<switch-ip-address>:/host/nos-images/
<desired-name>.bin
```

2. To install the image login to the switch and perform steps 2-3 from the "Install from NVOS" section.

## 2.1.7 Install Using a USB Drive

Follow the steps below to install the NVOS image using a USB drive.

 **Tip:** Installing NVOS using a USB drive is not scalable. DHCP can scale to hundreds of switch installs with zero manual input, unlike USB installs.

### 2.1.7.1 Prepare for USB Installation

1. Download the appropriate NVOS image for your platform.
2. From a computer, prepare your USB drive by formatting it using one of the supported formats: FAT32, vFAT, or EXT2.
  - Optional: Prepare a USB drive inside NVOS
    - Insert the USB drive into the USB port on the switch running NVOS and log in to the switch. Examine output from `cat /proc/partitions` and `sudo fdisk -l [device]` to determine the location of your USB drive. For example, `sudo fdisk -l /dev/sdb`. These instructions assume the USB drive is the `/dev/sdb` device, which is typical if inserting the USB drive after the machine is already booted. However, if the USB drive

was inserted during the boot process, it is possible that the USB drive is the `/dev/sda` device. Make sure to modify the commands below to use the proper device for the USB drive.

- b. Create a new partition table on the USB drive. If the `parted` utility is not on the system, install it with `sudo -E apt-get install parted`.

```
sudo parted /dev/sdb mklabel msdos
```

- c. Create a new partition on the USB drive:

```
sudo parted /dev/sdb -a optimal mkpart primary 0% 100%
```

- d. Format the partition to your filesystem of choice using *one* of the examples below:

```
sudo mkfs.ext2 /dev/sdb1
sudo mkfs.msdos -F 32 /dev/sdb1
sudo mkfs.vfat /dev/sdb1
```

To use `mkfs.msdos` or `mkfs.vfat`, installation of the `dosfstools` package from the [Debian software repositories](#) is needed, as they are not included by default.

- e. To continue installing NVOS, mount the USB drive to move files:

```
sudo mkdir /mnt/usb
sudo mount /dev/sdb1 /mnt/usb
```

3. Copy the NVOS image to the USB drive, then rename the image file to `onie-installer-x86_64`.

Any of the [ONIE naming schemes mentioned here](#) can also be used.

When using a MAC or Windows computer to rename the installation file, the file extension can still be present. Make sure to remove the file extension so that ONIE can detect the file.

4. Insert the USB drive into the switch, then prepare the switch for installation:

- If the switch is offline, connect to the console and power on the switch
- If the switch is already online in ONIE, use the `reboot` command

SSH sessions to the switch get dropped after this step. To complete the remaining instructions, connect to the console of the switch. NVOS switches display their boot process to the console; you need to monitor the console specifically to complete the next step.

5. Monitor the console and interrupt the GRUB countdown with the "ESC" or "F4" key when proposed, then select the ONIE option from the first GRUB screen shown below.





Once the password has been changed, please save your configuration in order to retain the password during subsequent reboots.  
If the switch is being managed only via ZTP, disregard the password change request.

```
You are required to change your password immediately (administrator enforced).  
  
WARNING: Your password has expired.  
You must change your password now!  
New password:  
Retype new password:
```

## 2.1.9 Related Information

- [ONIE Design Specification](#)

## 2.2 Upgrading/Downgrading NVOS Image

The following pages provide information on upgrading and downgrading the operating system version on the device.

- [Upgrading Operating System Software](#)
- [Deleting Unused Images](#)

### 2.2.1 Upgrading Operating System Software

#### 2.2.1.1 Important Notes Before Upgrading OS Software

Consider the following items prior to upgrading the operating system:

- The system becomes unavailable while OS upgrade is in progress.
- The upgrade procedure burns the software image as well as the firmware.
- Before upgrading the software image, make sure to close all CLI sessions besides the one used to run the upgrade process.

#### 2.2.1.2 Upgrading OS Software

To upgrade NVOS, perform the following steps.

1. Display the image (.bin file) that is currently available.

```
admin@nvos:~$ nv show system image
```

```

----- operational ----- applied
current <old_image>
next <old_image>
partition1 <old_image>

```

2. In case there are partition1 and partition2, Uninstall the image that isn't used as 'current' or 'next' prior to fetching the new image. Use the command `nv action uninstall system image`, for this purpose.

```
admin@nvos:~$ nv action uninstall system image
```

Another option is to clean-up all the unused images.

```
admin@nvos:~$ nv action uninstall system image force
```

3. Upload the new software image to the system using fetch command.

```
admin@nvos:~$ nv action fetch system image scp://<username>:<password>@<ip-address>/var/www/html/
<new_image>
```

4. Install the new image.

```
admin@nvos:~$ nv action install system image files new-nvos-image.bin
Operation will reboot the system.
If you choose 'y', the system will install the image and reboot.
If you choose 'N', the operation will abort without installing the image and without rebooting the system.
Do you want to continue? [y/N]
```

5. After reboot, display available images and verify that the new image now appears.

```
admin@nvos:~$ nv show system image
----- operational ----- applied
current <new_image>
next <new_image>
partition1 <old_image>
partition2 <new_image>

```



- After software reboot, the software upgrade will also automatically upgrade the firmware version.
- The system will reboot twice if "nv action reboot system immediate" or "nv action power-cycle system" commands are used after NVOS installation. For optimized reboot time, use "nv action reboot system."
- To recover from image corruption (e.g., due to power interruption), there are two installed images on the system. For more information, see the [nv action boot-next system image](#) command.

## 2.2.2 Deleting Unused Images

To delete unused images, conduct the following steps:

1. Get a list of the unused images.

```
admin@nvos:~$ nv show system image
----- operational ----- applied
current <image1>
```

```
next <image1>
partition1 <image1>
partition2 <image2>
```

2. Uninstall the unused images.

```
admin@nvos:~$ nv action uninstall system image
```

## 2.3 Upgrading System Firmware

- [2.3.1 Importing Firmware and Changing the Default Firmware](#)
  - [2.3.1.1 Default Firmware Change](#)
- [2.3.2 Upgrade Firmware](#)
  - [2.3.2.1 References](#)
  - [2.3.2.2 Bundles content and upgrade sequence](#)
    - [2.3.2.2.1 Bundles List](#)
    - [2.3.2.2.2 Upgrade Sequence](#)
  - [2.3.2.3 Component Image Update Using NVOS CLI](#)
    - [2.3.2.3.1 Transceiver Firmware Upgrade](#)
  - [2.3.2.4 Component Image Update Using RestAPI](#)
    - [2.3.2.4.1 REST API Commands](#)
  - [2.3.2.5 Recommended Full System Upgrade Sequence Example for Automation Reference](#)
  - [2.3.2.6 Error Status Catalog](#)

NVOS software package version has a default switch firmware version. When updating the operating system software to a new version, an automatic firmware update process will be attempted by NVOS. This process is described below.

### 2.3.1 Importing Firmware and Changing the Default Firmware

To perform an automatic firmware update by the OS for a different switch firmware version without changing the OS version, import and install the firmware package as described below. The OS sets it as the new default firmware and performs the firmware update automatically.

#### 2.3.1.1 Default Firmware Change

1. Display the firmware that is currently available.

```
admin@nvos:~$ nv show platform firmware ASIC
----- operational ----- applied
part-number 920-9K36F-00MV-JS0_IPN_Ax
actual-firmware 35.2014.1482
auto-update enabled enabled
fw-source default default
```

2. Import the firmware image (.mfa file) to the switch.

```
admin@nvos:~$ nv action fetch platform firmware ASIC /path/to/fw-image.mfa
```

Alternatively, you can upload the FW file from the host to the switch:



The firmware file must be copied to the predefined directory: "/host/fw-images/asic"

```
user@host:~$ scp <path-to-fw-file> <switch-admin-username>@<switch-ip-address>:/host/fw-images/asic/  
<desired-name>.mfa
```

3. Configure default firmware source from user.

```
admin@nvos:~$ nv set platform firmware ASIC fw-source custom
```

4. Display system firmware component information.

```
admin@nvos:~$ nv show platform firmware ASIC  
operational          applied  
-----  
part-number          920-9K36F-00MV-JS0_IPN_Ax  
actual-firmware      35.2014.1482  
auto-update          enabled  
fw-source            default
```

5. Apply the configuration

```
admin@nvos:~$ nv config apply
```

6. Save the configuration.

```
admin@nvos:~$ nv config save
```

7. Install the firmware image.

```
admin@nvos:~$ nv action install platform firmware ASIC files fw-image.mfa  
The operation will initiate a component firmware update.  
Type [y] to install the firmware and reboot afterwards.  
Type [N] to install the firmware without reboot.  
  
Do you want to reboot? [y/N]
```

8. Press 'Y' will cause system reboot to install and activate firmware.

9. Press 'N' requires manual reboot from user later on.

```
admin@nvos:~$ nv action reboot system
```

## 2.3.2 Upgrade Firmware

This section provides step-by-step instructions to manually check and update software and firmware on GB200-NVL NVLink switch tray to ensure the system is up to date with the latest software and firmware versions.



- This guide only applies to upgrades on system of the same type (e.g., QS to QS)
- It is highly recommended to use nvfwupd Tool to conduct an automatic update GB200-NVL (see NVIDIA Firmware Update Tool: NVOnline Document ID 1107320)
- For CPLD only: Unpack the bundles using fwpkg-unpack CLI (see Firmware Package Unpacking Tool: NVOnline Document ID 1090243)

### 2.3.2.1 References

Document Title	NVOnline Document ID
GB200 NVL72 Rack System Specification and Integration Guide	1117886
GB200 NVL Technical Overview	1114121
NVLink Switch Tray	
NVIDIA NVLink GB200 Switch Systems User Manual	1115337
NVIDIA NVOS User Manual	1114436
Tools	
NVIDIA Firmware Update Tool	1107320
Firmware Package Unpacking Tool	1090243

### 2.3.2.2 Bundles content and upgrade sequence

#### 2.3.2.2.1 Bundles List

Bundle	Content	File type	Estimated file size	Estimated update time
BMC, ERoT and ERoT	BMC	.fwpkg	65 MB	12 minutes
	FPGA	.fwpkg	11 MB	2 minutes
	ERoT	.fwpkg	225 KB	15 seconds
BIOS and ERoT	BIOS	.fwpkg	17 MB	4 minutes
	ERoT	.fwpkg	225 KB	15 seconds
CPLD	CPLD	.vme / .bin	1.6 MB	10 minutes
NVOS (not part of the bundle)		.bin	2.3GB	10 minutes



NVOS is not part of the bundle. NVSwitch5 firmware is part of NVOS.

#### 2.3.2.2.2 Upgrade Sequence

1. BMC
2. FPGA
3. ERoT
4. CPLD
5. BIOS
6. NVOS



1. These updates are not done every release. See NVIDIA NVOS Release Notes to see which versions should be used.
2. A power cycle is needed at the end of the upgrade process.
3. The upgrade process will require maintenance window.
4. If necessary, retrieve logs for customer support using the command "nv action generate system tech-support".

### 2.3.2.3 Component Image Update Using NVOS CLI

Firmware updates can be done by NVOS CLI commands. CLI commands are blocking, meaning each command must be finished before another one can be.

There are two stages to upgrade each component:

1. Fetching a file from the unpacked bundle.

```
admin@nvos:~$ nv action fetch platform firmware <component-id> <remote-url>
```

For details, see [nv action fetch platform firmware](#).

2. Installing a component:

```
admin@nvos:~$ nv action install platform firmware <component-id> files <file-name>
```

For details, see [nv action install platform firmware files](#).

To save time, it is recommended to update one-by-one component and then to choose a power cycle.



- <component-id> can be one of the following: ASIC, BMC, BIOS, CPLD1, ERoT and FPGA.
- Once upgrading a specific CPLD, all other CPLDs will be upgraded as well.

1. Power cycle should be triggered if it was chosen after install command but if manual power cycle required, run the following:

```
admin@nvos:~$ nv action power-cycle system
```

2. To verify firmware versions after power cycle, run the following:

For details, see [nv show platform firmware](#).

#### 2.3.2.3.1 Transceiver Firmware Upgrade

Firmware updates can be done by NVOS CLI commands. CLI commands are blocking, meaning each command must be finished before another one can be.

There are two stages to upgrade each component:

1. Fetching a file from the unpacked bundle.

```
admin@nvos:~$ nv action fetch platform firmware transceiver <file-path>
```

For details, see [nv action fetch platform firmware](#).

## 2. Installing transceiver firmware.

```
admin@nvos:~$ nv action install platform transceiver <transceiver-id> firmware files <file-name>
```

For details, see [nv action install platform transceiver firmware files](#).

In order to activate the transceiver firmware, NVOS will reset the transceiver as part of the install action.

To verify firmware version, run the following:

```
admin@nvos:~$ nv show platform transceiver <transceiver-id> firmware
```

For details, see [nv show platform transceiver firmware](#).

## 2.3.2.4 Component Image Update Using RestAPI

RestAPI can be used from remote server to perform operations on the switch.

RestAPI is not blocking, meaning command can be sent before the previous finished. To deal with this nature, each command returns Task ID, use the Task ID to query for the result between the commands. State of “action\_success” means the operation ended successfully.

Upgrades consist of fetch, install, and power cycle at the end of the entire process.

### 2.3.2.4.1 REST API Commands

Query command, should be executed between commands:

```
admin@nvos:~$ curl -k --user <nvos-user>:<nvos-password> --request GET 'https://<switch-ip>/nvue_v1/action/<task-id>'
```

#### 1. Fetching component image file:

```
admin@nvos:~$ curl -k --user <nvos-user>:<nvos-password> --request POST 'https://<switch-ip>/nvue_v1/platform/firmware/<component>' -H 'Content-Type: application/json' -d '{"@fetch": {"state": "start", "parameters": {"remote-url": "scp://<server-user>:<server-password> >@<PATH_TO_FILE>"}}}'
```

#### 2. Install the component file:

```
admin@nvos:~$ curl -k --user <nvos-user>:<nvos-password> --request POST 'https://<switch-ip>/nvue_v1/platform/firmware/<component>/files/</><file-name>' -H 'Content-Type: application/json' -d '{"@install": {"state": "start", "parameters": {"force": false}}}'
```

#### 3. Power cycle:

```
admin@nvos:~$ curl -k --user <nvos-user>:<nvos-password> --request POST 'https://<switch-ip>/nvue_v1/system' -H 'Content-Type: application/json' -d '{"@power-cycle": {"state": "start", "parameters": {"force": true}}}'
```

#### 4. After power cycle, check firmware version:

```
admin@nvos:~$ curl -k --user <nvos-user>:<nvos-password> --request GET 'https://<nvos-ip>/nvue_v1/platform/firmware'
```

## 2.3.2.5 Recommended Full System Upgrade Sequence Example for Automation Reference

### 1. Fetch and install BMC.

```
admin@nvos:~$ nv action fetch platform firmware BMC <remote-url-to-BMC-bundle>
admin@nvos:~$ nv action install platform firmware BMC files <fetched-file-name> skip-reboot
```

### 2. Fetch and install FPGA.

```
admin@nvos:~$ nv action fetch platform firmware FPGA <remote-url-to-FPGA-bundle>
admin@nvos:~$ nv action install platform firmware FPGA files <fetched-file-name> skip-reboot
```

### 3. Fetch and install ERoT.

```
admin@nvos:~$ nv action fetch platform firmware ERoT <remote-url-to-ERoT-bundle>
admin@nvos:~$ nv action install platform firmware ERoT files <fetched-file-name> skip-reboot
```

### 4. Fetch and install CPLD.

For CPLD only: Unpack the bundles using fwpkg-unpack CLI (see Firmware Package Unpacking Tool: NVOnline Document ID [1090243](#))

```
admin@nvos:~$ nv action fetch platform firmware CPLD1 <remote-url-to-unpacked-CPLD-file>
admin@nvos:~$ nv action install platform firmware CPLD1 files <fetched-file-name> skip-reboot
```

### 5. Fetch and install BIOS:

```
admin@nvos:~$ nv action fetch platform firmware BIOS <remote-url-to-BIOS-bundle>
admin@nvos:~$ nv action install platform firmware BIOS files <fetched-file-name> skip-reboot
```

### 6. Fetch and install NVOS, and reboot the system:

```
admin@nvos:~$ nv action fetch system image files <remote-url-to-NVOS-file>
admin@nvos:~$ nv action install system image files <fetched-file-name> reboot no
admin@nvos:~$ nv action reboot system
```



- You can force reboot the system upon NVOS installation by using "force" option

```
admin@nvos:~$ nv action install system image files <fetched-file-name> force
```

- For automation scripts using NVOS SSH connection, it is recommended to set SSH inactivity timeout to at least 20 minutes prior starting the upgrade process, to enhance automation SSH resiliency

```
admin@nvos:~$ nv set system ssh-server inactivity-timeout 20
admin@nvos:~$ nv config apply
admin@nvos:~$ nv config save
```

- Every NVUE CLI has an equivalent REST API call. Please refer to the documentation for the relevant commands.
- The system will reboot twice if "nv action reboot system immediate" or "nv action power-cycle system" commands are used after NVOS installation. For optimized reboot time, use "nv action reboot system."

### 2.3.2.6 Error Status Catalog

Use the table below to identify the errors and their meaning.

#### Bundles List

BMC	
Scenario	Error
Selected file for installation doesn't exist	Failed to install BMC firmware file: No such firmware
Bad or corrupted file	Invalid file: /host/fw-images/bmc/bad_file.fwpkg
BMC is not accessible	Error: Timed out ...
Failed to login to BMC	Error: Timed out ...
Curl returns any other error when sending post request for BMC image installation	Error: X (being X the error returned by Curl)
During the installation process got responses in invalid format (responses should be in json format)	Error: Invalid JSON format
BMC returned an error code when triggering installation process (json response for installation command contained 'error' field)	Error returned by BMC
BMC returned not ok task status on installation response	Error: Return status is {status}
BMC response does not include task status	Error: Missing 'TaskStatus' field
BMC response task status is not OK during polling for installation	Error: Fail to execute the task - Taskstatus={status}
Error detected during installation process	Error: {err_msg}
Installation process was aborted (on BMC side)	Error: The task has been aborted
EROT (same errors as BMC)	
Scenario	Error
Selected file for installation doesn't exist	Failed to install EROT firmware file: No such firmware
Bad or corrupted file	Invalid file: /host/fw-images/erot/bad_file.fwpkg
BMC is not accessible	Error: Timed out ...
Failed to login to BMC	Error: Timed out ...
Curl returns any other error when sending post request for BMC image installation	Error: X (being X the error returned by Curl)

Error: Invalid JSON format	During the installation process got responses in invalid format (responses should be in json format)
BMC returned an error code when triggering installation process (json response for installation command contained 'error' field)	Error returned by BMC
BMC returned not ok task status on installation comand response	Error: Return status is {status}
BMC response doesn't include task status	Error: Missing 'TaskStatus' field
BMC response task status is not OK during polling for installation completion	Error: Fail to execute the task - Taskstatus={status}
Error detected during installation process	Error: {err_msg}
Installation process was aborted (on BMC side)	Error: The task has been aborted
Installation did not finish in 30 minutes	Wait task completion timeout
FPGA (same errors as BMC)	
Scenario	Error
Selected file for installation doesn't exist	Failed to install ER0T firmware file: No such firmware
Bad or corrupted file	Invalid file: /host/fw-images/fpga/bad_file.fwpkg
BMC is not accessible	Error: Timed out ...
Failed to login to BMC	Error: Timed out ...
Curl returns any other error when sending post request for BMC image installation	Error: X (being X the error returned by Curl)
Error: Invalid JSON format	During the installation process got responses in invalid format (responses should be in json format)
BMC returned an error code when triggering installation process (json response for installation command contained 'error' field)	Error returned by BMC
BMC returned not ok task status on installation comand response	Error: Return status is {status}
BMC response doesn't include task status	Error: Missing 'TaskStatus' field
BMC response task status is not OK during polling for installation completion	Error: Fail to execute the task - Taskstatus={status}
Error detected during installation process	Error: {err_msg}
Installation process was aborted (on BMC side)	Error: The task has been aborted
Installation didn't finish in 30 minutes	Wait task completion timeout

BIOS	
Scenario	Error
Selected file for installation doesn't exist	Failed to install BIOS firmware file: No such firmware
Bad or corrupted file	Invalid file: /host/fw-images/bios/bad_file.cab

Bad Onie version	ERROR: ONIE {} or later is required
Failed to enable ONIE firmware update mode	ERROR: failed to enable ONIE firmware update mode
Failed to disable ONIE firmware update mode	ERROR: failed to disable ONIE firmware update mode
Installation script was interrupted by signal	WARNING: Interrupted by \${_sig}: disable ONIE firmware update mode
CPLD	
Scenario	Error
Selected file for installation doesn't exist	Failed to install CPLD firmware file: No such firmware
Bad or corrupted file	Invalid file: /host/fw-images/cpld/bad_file.vme
MST service not started (never started or failed to start)	ERROR: mst driver is not loaded
MST device path doesn't exist or failed to it	ERROR: Failed to get mst device: pattern={}, devices={}
CPLD update command failed	ERROR: Failed to update {} firmware: {}

## 2.4 Installation Management Commands

- [2.4.1 Version](#)
  - [2.4.1.1 nv show system version](#)
- [2.4.2 Image](#)
  - [2.4.2.1 nv show system image](#)
  - [2.4.2.2 nv show system image files](#)
  - [2.4.2.3 nv action fetch system image](#)
  - [2.4.2.4 nv action install system image files](#)
  - [2.4.2.5 nv action rename system image files](#)
  - [2.4.2.6 nv action upload system image files](#)
  - [2.4.2.7 nv action uninstall system image](#)
  - [2.4.2.8 nv action delete system image files](#)
  - [2.4.2.9 nv action uninstall system image force](#)
  - [2.4.2.10 nv action boot-next system image](#)
- [2.4.3 Firmware](#)
  - [2.4.3.1 nv show platform firmware](#)
  - [2.4.3.2 nv show platform firmware id](#)
  - [2.4.3.3 nv show platform firmware <erot-id>](#)
  - [2.4.3.4 nv show platform firmware files](#)
  - [2.4.3.5 nv set/unset platform firmware](#)
  - [2.4.3.6 nv action install platform firmware files](#)
  - [2.4.3.7 nv action delete platform firmware files](#)
  - [2.4.3.8 nv action rename platform firmware files](#)
  - [2.4.3.9 nv action fetch platform firmware](#)

- [2.4.3.10 nv action upload platform firmware files](#)

## 2.4.1 Version

### 2.4.1.1 nv show system version

	<b>nv show system version</b> Display version information for system image that is currently running.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show system version ----- operational ----- applied kernel      5.10.0-8-2-amd64 build-date  Mon May 30 22:45:11 UTC 2022 image       nvos-25.01.0002           </pre>
REST API	GET https://<ip>/nvue_v1/system/version
Related Commands	nv show system nv show system image
Notes	

## 2.4.2 Image

### 2.4.2.1 nv show system image

	<b>nv show system image</b> Show system image information. Display version information related to system image.
Syntax Description	N/A
Default	N/A
History	25.02.1884 25.02.4257: Updated command output
Example	<pre> admin@nvos: ~\$ nv show system image ----- operational ----- current      2 next         2 partition1   build-id   nvos-25.02.1936 partition2   build-id   nvos-25.02.2930-010           </pre>
REST API	GET https://<ip>/nvue_v1/system/image
Related commands	nv show system image files
Notes	

### 2.4.2.2 nv show system image files

	nv show system image files Show files in the image directory on the switch file system.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system image files Available image file ----- nvos-amd64-25.01.2434-013.bin nvos-amd64-25.01.2504.bin</pre>	
REST API	GET https://<ip>/nvue_v1/system/image/files	
Related commands	nv show system image	
Notes		

### 2.4.2.3 nv action fetch system image

	nv action fetch system image <remote-url> Fetch/download a file from a remote server and stores it locally.	
Syntax Description	remote-url	<ul style="list-style-type: none"> <li>• Remote url path to fetch file from.</li> <li>• Format: [protocol]://username[:password]@hostname/path/filename</li> <li>• Supported protocols: SCP, HTTPS, FILE, FTP, and SFTP.</li> <li>• The password must be encoded if it contains special characters and is provided as part of the command.</li> </ul>
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action fetch system image scp://my_user:my_password@server/path/image.bin admin@nvos:~\$ nv action fetch system image https://some.domain.com/path/image.bin admin@nvos:~\$ nv action fetch system image file:///path/image.bin</pre>	
REST API	POST https://<ip>/nvue_v1/system/image	
Related Commands	nv show system image files	
Notes	Using https requires the remote server to have a valid CA certificate.	

### 2.4.2.4 nv action install system image files

	nv action install system image files {image-file} [force] [reboot <yes/no>] Install system image from a binary file and reboot the system.	
Syntax Description	image-file	Path to the binary file to install on the OS filesystems
	force	Force the action without asking for user confirmation
	reboot	Install image with or without system reboot

Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action install system image files nvos.bin The operation will install the image and initiate a reboot. Type [y] to install the image and reboot. Type [N] to abort.  Do you want to <b>continue</b>? [y/N] N Image install aborted by user</pre>
REST API	POST https://<ip>/nvue_v1/system/image/files/
Related commands	nv show system image
Notes	Executing the command for an image that is already installed on the other partition will not install the image again, it will only change the "boot-next" image.

### 2.4.2.5 nv action rename system image files

	nv action rename system image files <image> <new-name> Rename an image file.	
Syntax Description	image	Source image file name
	new-name	Destination image file name
Default	N/A	In case of an empty parameter performs a regular reboot without force.
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action rename system image files old_name.img new_name.img</pre>	
REST API	POST https://<ip>/nvue_v1/system/image/files/<file-name>	
Related Commands	nv show system image files	
Notes		

### 2.4.2.6 nv action upload system image files

	nv action upload system image files <image> <remote-url> Upload an image file to remote location.	
Syntax Description	image	Image file to upload
	remote-url	Destination image file name Remote url path to upload a file to. Format: [protocol]:// username[:password]@hostname/path/filename Supported protocols: scp ftp tftp and sftp
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action upload system image image_file.bin scp:// my_user:my_password@server/path/image_file.bin</pre>	

REST API	POST https://<ip>/nvue_v1/system/image/files/<file-name>
Related commands	nv show system image files
Notes	

### 2.4.2.7 nv action uninstall system image

	nv action uninstall system image Removes old unused images that are not current or boot-next.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action uninstall system image Cleaning up old unused images... Removing image nvos-old Images cleanup done Action succeeded  admin@nvos:~\$ nv action uninstall system image Cleaning up old unused images... No image(s) to cleanup Action succeeded</pre>
REST API	POST https://<ip>/nvue_v1/system/image
Related commands	nv show system image
Notes	

### 2.4.2.8 nv action delete system image files

	nv action delete system image files <image> Delete an image from file system.
Syntax Description	image      Image file name
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action delete system image files nvos.bin</pre>
REST API	POST https://<ip>/nvue_v1/system/image/files/<file-name>
Related Commands	nv show system image files
Notes	

### 2.4.2.9 nv action uninstall system image force

	nv action uninstall system image force Uninstall the system image. Uninstall old unused images that are not current image.
--	--

Syntax Description	image-id	System image identifier
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action uninstall system image force Uninstalling system image: nvos Image nvos uninstalled successfully Action succeeded  admin@nvos:~\$ nv action uninstall system image Uninstalling system image: nvos Action failed with the following issue: Image nvos can't be uninstalled. Image does not exist</pre>	
REST API	POST https://<ip>/nvue_v1/system/image	
Related commands	nv show system image	
Notes	Image ID can be retrieved by the "nv show system image" command.	

### 2.4.2.10 nv action boot-next system image

	nv action boot-next system image <image-id> Set boot-next system image. Set image to boot from at the next boot.	
Syntax Description	image-id	System image identifier
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action boot-next system image nvos Setting system image nvos as boot-next.. Image nvos set as boot-next Action succeeded  admin@nvos:~\$ nv action boot-next system image nvos Setting system image nvos as boot-next.. Action failed with the following issue: Failed to set boot-next: Image nvos does not exist</pre>	
REST API	POST https://<ip>/nvue_v1/system/image	
Related commands	nv show system image	
Notes	<ul style="list-style-type: none"> <li>Image ID can be retrieved by "nv show system image" command.</li> <li>Select which partition to boot from.</li> <li>Booting to an image with existing DB (with an image that is not freshly installed), will not migrate the configurations from the current image.</li> </ul>	

## 2.4.3 Firmware

### 2.4.3.1 nv show platform firmware

	nv show platform firmware Show platform firmware information. Show firmware information for platform components like ASIC, BIOS, SSD, CPLD, and transceiver.
--	--

Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show platform firmware Name          Actual FW          Part Number          FW Source ----- ASIC          35.2014.1498      920-9K36F-00MV-JS0_IPN_Ax  default BIOS          00.00.018         N/A                  N/A BMC           88.0002.0590      N/A                  N/A CPLD1        CPLD000370_REV0202 0x0172              N/A CPLD2        CPLD000377_REV0308 0x0179              N/A CPLD3        CPLD000373_REV0205 0x0175              N/A CPLD4        CPLD000390_REV0103 0x0186              N/A EROT         01.03.0235.0000_n04 N/A                  N/A EROT-ASIC1   01.03.0235.0000_n04 N/A                  N/A EROT-ASIC2   01.03.0235.0000_n04 N/A                  N/A EROT-BMC     01.03.0235.0000_n04 N/A                  N/A EROT-CPU     01.03.0235.0000_n04 N/A                  N/A EROT-FPGA    01.03.0235.0000_n04 N/A                  N/A FPGA         302e3139          N/A                  N/A SSD          CE00A400          Virtium VTPM24CEXI080-BM110006 N/A transceiver  N/A               N/A                  N/A </pre>
REST API	GET https://<ip>/nvue_v1/platform/firmware
Related commands	
Notes	

### 2.4.3.2 nv show platform firmware id

	<p>nv show platform firmware &lt;component-id&gt;  Show platform firmware information for specific component.  Show firmware information for platform components such as ASIC, BIOS, CPLD, SSD, and transceiver.</p>	
Syntax Description	component-id	Platform component name: ASIC, BIOS, CPLD1, SSD, and transceiver. For systems with BMC, additional firmware components are supported: EROT, FPGA, BMC.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform firmware ASIC operational      applied ----- part-number      920-9K36F-00MV-JS0 actual-firmware  35.2014.1476 auto-update      enabled fw-source        default </pre>	
REST API	GET https://<ip>/nvue_v1/platform/firmware/<component-id>	
Related Commands	nv show platform firmware	
Notes		

### 2.4.3.3 nv show platform firmware <erot-id>

	<p>nv show platform firmware &lt;erot-id&gt;  Show platform firmware information for specific ERoT component.  Show firmware information for platform components such as EROT-BMC, EROT-CPU, EROT-FPGA, EROT-ASIC1, EROT-ASIC2.</p>
--	---

Syntax Description	erot-id	Platform ERoT names: ERoT-BMC, ERoT-CPU, ERoT-FPGA, ERoT-ASIC1, ERoT-ASIC
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform firmware ERoT-CPU operational                applied ----- part-number                 920-9K36F-00MV-JS0 actual-firmware            35.2014.1476 auto-update                enabled fw-source                  default background-copy-status    pending debug-token-status        not-installed active                    flash1 inactive                  flash2 ap-boot-status             0x01234567 </pre>	
REST API	GET https://<ip>/nvue_v1/platform/firmware/<erot-id>	
Related Commands	nv show platform firmware	
Notes		

### 2.4.3.4 nv show platform firmware files

	nv show platform firmware <component-id> files Display the list of available firmware files for specific component. Display the list of available firmware files for platform components such as ASIC, BIOS, CPLD and transceiver.	
Syntax Description	component-id	Platform component name: ASIC, BIOS, CPLD, and transceiver. For system with BMC, more firmware like ERoT/FPGA/BMC are supported.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform firmware ASIC files Available firmware files      File path ----- fw-NVL5-rel-35_2014_1476.mfa /host/fw-images/fw-NVL5-rel-35_2014_1476.mfa  admin@nvos:~\$ nv show platform firmware transceiver files Available Firmware Files     File Path ----- fw_47_150_03003_dev_signed.bin /host/fw-images/module/fw_47_150_03003_dev_signed.bin </pre>	
REST API	GET https://<ip>/nvue_v1/platform/firmware/{component-id}/files	
Related commands	nv show platform firmware nv action install platform firmware <component-id> files <file-name> nv action install platform transceiver <transceiver-id> firmware files <file-name>	
Notes		

### 2.4.3.5 nv set/unset platform firmware

	nv set platform firmware [ASIC   BIOS] [auto-update {enabled,disabled}   fw-source {default,custom}] nv unset platform firmware [ASIC   BIOS] [auto-update {enabled,disabled}   fw-source {default,custom}] Update the platform ASIC or BIOS component configuration.	
--	---	--

Syntax Description	auto-update	Firmware component auto-update state
	fw-source	Firmware component default source
Default	auto-update	enabled
	fw-source	default
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set platform firmware ASIC fw-source custom admin@nvos:~\$ nv set platform firmware ASIC auto-update disabled</pre>	
REST API	PATCH https://<ip>/nvue_v1/platform/firmware/ASIC	
Related commands		
Notes	The configuration on BIOS is not available on system with BMC.	

### 2.4.3.6 nv action install platform firmware files

	<p>nv action install platform firmware &lt;component_id&gt; files &lt;file_path&gt; [force] [skip-reboot]</p> <p>Install platform firmware image.</p> <p>ASIC Image will be installed to stage location and will be installed at the next reboot time.</p>	
Syntax Description	component_id	Platform component name: ASIC, BIOS, CPLD1, SSD, and transceiver. For systems with BMC, additional firmware components are supported: EROT, FPGA, BMC.
	file_path	Path to the firmware file.
	force	Force the action without asking for user confirmation.
	skip-reboot	Skip reboot after firmware installation. If force was specified, the reboot will still be skipped.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action install platform firmware ASIC files fw-NVL5.mfa Installing firmware: /tmp/fw-NVL5.mfa Firmware fw-NVL5.mfa successfully installed Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/platform/firmware/{platform-component-id}/files/{file-name}	
Related commands	nv set/unset platform firmware ASIC auto-update nv action reboot system	
Notes	<ul style="list-style-type: none"> <li>• ASIC Image will be installed to stage location and will be installed at the next reboot time</li> <li>• Installation respects "auto-update" and "fw-source" configuration</li> <li>• To install firmware on a transceiver, use the following command: <a href="#">nv action install platform transceiver firmware files</a></li> <li>• After installing firmware, initiating system reboot via the "nv action reboot system" command will perform a power-cycle to load the new firmware.</li> </ul>	

### 2.4.3.7 nv action delete platform firmware files

	nv action delete platform firmware <component-id> files <file-name> Delete firmware file from the filesystem for specific component such as ASIC, BIOS, CPLD and transceiver.	
Syntax Description	file-name	firmware file name
	component-id	Platform component name: ASIC, BIOS, CPLD, and transceiver. For system with BMC, more firmware like EROT/FPGA/BMC are supported.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv action delete platform firmware ASIC files fw-test.mfa File delete successsfully Action succeeded  admin@nvos:~\$ nv action delete platform firmware transceiver files fw-test.bin File delete successsfully Action succeeded           </pre>	
REST API	POST https://<ip>/nvue_v1/platform/firmware/{component-id}/files/{file-name}	
Related commands	nv show platform firmware <component-id> files nv action fetch platform firmware <component-id> <url> nv action upload platform firmware <component-id> files <file-name> <remote-url> nv action rename platform firmware <component-id> files <file-name> <new-name> nv action install platform firmware <component-id> files <file-name> [force] nv show platform transceiver <transceiver-id> firmware nv action install platform transceiver <transceiver-id> firmware files	
Notes		

### 2.4.3.8 nv action rename platform firmware files

	nv action rename platform firmware <component-id> files <file-name> <new-name> Rename available firmware file for platform components such as ASIC, BIOS, CPLD and transceiver.	
Syntax Description	file-name	Firmware file name
	new-name	New name for firmware file
	component-id	Platform component name: ASIC, BIOS, CPLD, and transceiver. For system with BMC, more firmware like EROT/FPGA/BMC are supported.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv action rename platform firmware ASIC files fw-NVL5.mfa new-fw-NVL5.mfa File renamed successsfully Action succeeded  admin@nvos:~\$ nv action rename platform firmware transceiver files fw-test.bin new-fw-test.bin File renamed successsfully Action succeeded           </pre>	
REST API	POST https://<ip>/nvue_v1/platform/firmware/{component-id}/files/{file-name}	

Related commands	<pre> nv show platform firmware &lt;component-id&gt; files nv action fetch platform firmware &lt;component-id&gt; &lt;url&gt; nv action upload platform firmware &lt;component-id&gt; files &lt;file-name&gt; &lt;remote-url&gt; nv action delete platform firmware &lt;component-id&gt; files &lt;file-name&gt; nv action install platform firmware &lt;component-id&gt; files &lt;file-name&gt; [force] nv show platform transceiver &lt;transceiver-id&gt; firmware nv action install platform transceiver &lt;transceiver-id&gt; firmware files </pre>
Notes	

### 2.4.3.9 nv action fetch platform firmware

	<pre> nv action fetch platform firmware &lt;component-id&gt; &lt;remote-url&gt; </pre> <p>Fetch remote firmware file to local filesystem for platform components such as ASIC, BIOS, CPLD and transceiver.</p>	
Syntax Description	component-id	Platform component name: ASIC, BIOS, CPLD, and transceiver. For system with BMC, more firmware like EROT/FPGA/BMC are supported.
	remote-url	<ul style="list-style-type: none"> <li>Remote url path to fetch file from.</li> <li>Format: [protocol]://username[:password]@hostname/path/filename</li> <li>Supported protocols: SCP, HTTPS, FILE, FTP, and SFTP.</li> <li>The password must be encoded if it contains special characters and is provided as part of the command.</li> </ul>
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv action fetch platform firmware ASIC scp://my_user:my_password@hostname/ dir/file.mfa Fetching file ... File fetched successfully Action succeeded  admin@nvos:~\$ nv action fetch platform firmware transceiver scp:// my_user:my_password@hostname/dir/file.bin Fetching file ... File fetched successfully Action succeeded  admin@nvos:~\$ nv action fetch platform firmware ASIC file:///dir/file.mfa Fetching file ... File fetched successfully Action succeeded </pre>	
REST API	POST https://<ip>/nvue_v1/platform/firmware/{component-id}/files/{file-name}	
Related commands	<pre> nv show platform firmware &lt;component-id&gt; files nv action rename platform firmware &lt;component-id&gt; files &lt;file-name&gt; &lt;new-name&gt; nv action upload platform firmware &lt;component-id&gt; files &lt;file-name&gt; &lt;remote-url&gt; nv action delete platform firmware &lt;component-id&gt; files &lt;file-name&gt; nv action install platform firmware &lt;component-id&gt; files &lt;file-name&gt; [force] nv show platform transceiver &lt;transceiver-id&gt; firmware nv action install platform transceiver &lt;transceiver-id&gt; firmware files </pre>	
Notes		

## 2.4.3.10 nv action upload platform firmware files

	nv action upload platform firmware <component-id> files <file-name> <remote-url> Upload available firmware file for platform components such as ASIC, BIOS, CPLD and transceiver to remote server.	
Syntax Description	component-id	Platform component name: ASIC, BIOS, CPLD, and transceiver. For system with BMC, more firmware like EROT/FPGA/BMC are supported.
	file-name	Firmware file name
	remote-url	Remote url
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action upload platform firmware ASIC files fw-NVL5.mfa new-fw-NVL5.mfa Action executing ... File upload successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/platform/firmware/{component-id}/files/{file-name}	
Related commands	nv show platform firmware <component-id> files nv action fetch platform firmware <component-id> <url> nv action delete platform firmware <component-id> files <file-name> nv action install platform firmware <component-id> files <file-name> [force] nv show platform transceiver <transceiver-id> firmware nv action install platform transceiver <transceiver-id> firmware files	
Notes		

---

## 3 NVIDIA User Experience (NVUE)

NVUE is an object-oriented, schema driven model of a complete NVOS system (hardware and software) providing a robust API that allows for multiple interfaces to both view (show) and configure (set and unset) any element within a system running the NVUE software.

- [NVUE Object Model](#)
- [NVUE CLI Overview](#)
- [NVUE OpenAPI](#)

### 3.1 NVUE Object Model

The NVUE object model definition uses the [OpenAPI specification \(OAS\)](#). Similar to YANG ([RFC 6020](#) and [RFC 7950](#)), OAS is a data definition, manipulation, and modeling language (DML) that lets you build model-driven interfaces for both humans and machines. Although the computer networking and telecommunications industry commonly uses YANG (standardized by IETF) as a DML, the adoption of OpenAPI is broader, spanning cloud to compute to storage to IoT and even social media. The [OpenAPI Initiative \(OAI\) consortium](#) leads OpenAPI standardization, a chartered project under the Linux Foundation.

The OAS schema forms the management plane model with which you configure, monitor, and manage the NVOS switch. The v3.0.2 version of OAS defines the NVUE data model.

Like other systems that use OpenAPI, the NVUE OAS schema defines the endpoints (paths) exposed as RESTful APIs. With these REST APIs, you can perform various create, retrieve, update, delete, and eXecute (CRUDX) operations. The OAS schema also describes the API inputs and outputs (data models).

You can use the NVUE object model in these two ways:

- Through the NVUE REST API, where you run the GET, PATCH, DELETE, and other REST APIs on the NVUE object model endpoints to configure, monitor, and manage the switch. Because of the large user community and maturity of OAS, you can use several popular tools and libraries to create client-side bindings to use the NVUE REST API.
- Through the NVUE CLI, where you configure, monitor and manage the NVOS network elements. The CLI commands translate to their equivalent REST APIs, which NVOS then runs on the NVUE object model.

The CLI and the REST API are equivalent in functionality; you can run all management operations from the REST API or the CLI. The NVUE object model drives both the REST API and the CLI management operations. All operations are consistent; for example, the CLI `nv show` commands reflect any PATCH operation (create) you run through the REST API.

NVUE follows a declarative model, removing context-specific commands and settings. It is structured as a *big tree* that represents the entire state of a NVOS instance. At the base of the tree are high level branches representing objects, such as *system* and *interface*. Under each of these branches are further branches. As you navigate through the tree, you gain a more specific context. At the leaves of the tree are actual attributes, represented as key-value pairs. The path through the tree is similar to a filesystem path.

Please note that configuration with dependencies should be applied in two different steps.

NVOS installs NVUE by default and enables the NVUE service `nvued`.

## 3.2 NVUE CLI Overview

- [3.2.1 Command Syntax](#)
- [3.2.2 Command Completion](#)
- [3.2.3 Command Question Mark](#)
- [3.2.4 Command Abbreviation](#)
- [3.2.5 Command Help](#)
- [3.2.6 Command List](#)
- [3.2.7 Command History](#)
- [3.2.8 Command Ranges](#)
- [3.2.9 Command Categories](#)
  - [3.2.9.1 Configuration Commands](#)
  - [3.2.9.2 Monitoring Commands](#)
  - [3.2.9.3 Action Commands](#)
  - [3.2.9.4 Configuration Management Commands](#)
  - [3.2.9.5 List All NVUE Commands](#)
- [3.2.10 Search for a Specific Configuration](#)
- [3.2.11 Clear Switch Configuration](#)
- [3.2.12 Example Configuration Commands](#)
  - [3.2.12.1 Configure the System Hostname](#)
  - [3.2.12.2 Configure an Interface](#)
- [3.2.13 Example Monitoring Commands](#)
  - [3.2.13.1 Show Installed Software](#)
  - [3.2.13.2 Show Interface Configuration](#)
- [3.2.14 Reset NVUE Configuration to Default Values](#)
- [3.2.15 Add Configuration Apply Messages](#)
- [3.2.16 Configure Auto Save](#)
- [3.2.17 Example Configuration Management Commands](#)
  - [3.2.17.1 Apply and Save a Configuration](#)
  - [3.2.17.2 Detach a Pending Configuration](#)
  - [3.2.17.3 View Differences Between Configurations](#)
  - [3.2.17.4 Replace and Patch a Pending Configuration](#)
- [3.2.18 Passwords and Special Characters](#)

The NVUE CLI has a flat structure as opposed to a modal structure. This means that you can run all commands from the primary prompt instead of only in a specific mode.

### 3.2.1 Command Syntax

NVUE commands all begin with `nv` and fall into one of four syntax categories:

- Configuration ( `nv set` and `nv unset` )
- Monitoring ( `nv show` )

- Action commands ( `nv action` )
- Configuration management ( `nv config` ).

## 3.2.2 Command Completion

As you enter commands, you can get help with the valid keywords or options using the Tab key. For example, using Tab completion with `nv set` displays the possible options for the command and returns you to the command prompt to complete the command.

```
admin@nvos:~$ nv set <<press Tab>>
acl      interface platform system
```

## 3.2.3 Command Question Mark

You can type a question mark ( `?` ) after a command to display required information quickly and concisely. When you type `?`, NVUE specifies the value type, range, and options with a brief description of each; for example:

```
admin@nvos:~$ nv set system config auto-save ?
[Enter]
state          Enable/Disable configuration automatic save feature
```

NVUE also indicates if you need to provide specific values for the command:

```
admin@nvos:~$ nv set interface ?
<interface-id> Interface (interface-name)
```

## 3.2.4 Command Abbreviation

NVUE supports command abbreviation, where you can type a certain number of characters instead of a whole command to speed up CLI interaction. For example, instead of typing `nv show interface`, you can type `nv sh int`.

If the command you type is ambiguous, NVUE shows the reason for the ambiguity so that you can correct the shortcut. For example:

```
admin@nvos:~$ nv s i
Ambiguous Command:
set interface
show interface
```

## 3.2.5 Command Help

As you enter commands, you can get help with command syntax by entering `-h` or `--help` at various points within a command entry. For example, to examine the options available for `nv set interface`, enter `nv set interface -h` or `nv set interface --help`.

```
admin@nvos:~$ nv set interface -h
```

```
usage:
  nv [options] set interface <interface-id>

Description:
  interface Update all interfaces. Provide single interface or multiple interfaces using ranging (e.g.
  sw1-2p1-2 -> sw1p1,sw1p2,sw2p1,sw2p2).

Identifiers:
  <interface-id> Interface (interface-name)

General Options:
  -h, --help Show help.
```

## 3.2.6 Command List

You can list all the NVUE commands by running `nv list-commands`.

## 3.2.7 Command History

At the command prompt, press the Up Arrow and Down Arrow keys to move back and forth through the list of commands you entered. When you find a given command, you can run the command by pressing Enter. Optionally, you can modify the command before you run it.

## 3.2.8 Command Ranges

A single interface or multiple interfaces can be provided using a range such as `sw1-2p1-2 -> sw1p1,sw1p2,sw2p1,sw2p2`.

Example:

```
admin@nvos:~$ nv set interface sw1pls1,sw1pls2 description TEST
admin@nvos:~$ nv config diff
- set:
  interface:
    sw1pls1-2:
      description: TEST
      type: nvl
```

```
admin@nvos:~$ nv set interface sw1p1-2s1 description TEST
admin@nvos:~$ nv config diff
- set:
  interface:
    sw1p1-2s1,sw2p1-2s1:
      description: TEST
      type: nvl
```

## 3.2.9 Command Categories

The NVUE CLI has a flat structure; however, the commands are in four functional categories:

- Configuration
- Monitoring
- Action
- Configuration Management

### 3.2.9.1 Configuration Commands

The NVUE configuration commands modify switch configuration. You can set and unset configuration options.

The `nv set` and `nv unset` commands are in the following categories. Each command group includes subcommands. Use command completion (press the Tab key) to list the subcommands.

Command Group	Description
<code>nv set interface &lt;interface-id&gt;</code> <code>nv unset interface &lt;interface-id&gt;</code>	Configures the switch interfaces. Use this command to configure eth0/eth1/lo/nvl interfaces.
<code>nv set system</code> <code>nv unset system</code>	Configures the hostname of the switch, pre and post login messages, log level, users, password-hardening policies etc.
<code>nv set acl</code> <code>nv unset acl</code>	Configures ACLs.
<code>nv set platform</code> <code>nv unset platform</code>	Configures the platform firmware configuration.

### 3.2.9.2 Monitoring Commands

The NVUE monitoring commands show various parts of the network configuration. For example, you can show the complete network configuration or only interface configuration. The monitoring commands are in the following categories. Each command group includes subcommands. Use command completion (press the Tab key) to list the subcommands.

Command Group	Description
<code>nv show acl</code>	Shows Access Control List configuration.
<code>nv show action</code>	Shows information about the action commands job.
<code>nv show interface</code>	Shows interface configuration.
<code>nv show platform</code>	Shows platform configuration, such as hardware and software components.
<code>nv show system</code>	Shows global system settings, such as the images, tech-support files and log . You can also see system login messages and switch reboot history.
<code>nv show vrf</code>	Shows VRF configuration.
<code>nv show ib</code>	Shows system and MGMT GUIDs and LIDs



If there are no pending or applied configuration changes, the `nv show` command only shows the running configuration (under operational).

Additional options are available for the `nv show` commands. For example, you can choose the configuration you want to show (pending, applied, startup, or operational). You can also turn on colored output, and paginate specific output.

Option	Description
<code>--applied</code>	Shows configuration applied with the <code>nv config apply</code> command. For example, <code>nv show interface eth0 --applied</code> .
<code>--brief-help</code>	Shows help about the <code>nv show</code> command. For example, <code>nv show interface sw1p1 --brief-help</code>
<code>--color</code>	Turns colored output on or off. For example, <code>nv show interface eth0 --color on</code>
<code>--help</code>	Shows <code>help</code> for the NVUE commands.
<code>--filter</code>	Filters show command output on column data. For example, the <code>nv show interface --filter mtu=256</code> shows only the interfaces with operational MTU 256. To filter on multiple column outputs, enclose the filter types in parentheses; for example, <code>nv show interface --filter "type=ib&amp;mtu=256"</code> shows data for ib interfaces with MTU 256. You can use wildcards; for example, <code>nv show interface swp1 --filter "link.speed=200*"</code> shows all interfaces that have speed start with 200 . You can filter on all revisions (operational, applied, and pending); for example, <code>nv show interface --filter "ip.address=1*" --rev=applied</code> shows all IP addresses that start with 1 for in the applied revision.
<code>--hostname</code>	Shows system configuration for the switch with the specified hostname. For example, <code>nv show --hostname leaf01</code> .
<code>--operational</code>	Shows the running configuration (the actual system state). For example, <code>nv show interface eth0 --operational</code> shows the running configuration for eth0. The running and applied configuration should be the same. If different, inspect the logs.
<code>--output</code>	Shows command output in table format (auto), <code>json</code> format or <code>yaml</code> format. For example: <code>nv show interface eth0 --output auto</code> <code>nv show interface eth0 --output json</code> <code>nv show interface eth0 --output yaml</code>
<code>--paginate</code>	Paginates the output. For example, <code>nv show interface eth0 --paginate on</code> .
<code>--pending</code>	Shows configuration that is <code>set</code> and <code>unset</code> but not yet applied or saved. For example, <code>nv show interface eth0 --pending</code> .
<code>--rev &lt;revision&gt;</code>	Shows a detached pending configuration. See the <code>nv config detach</code> configuration management command below. For example, <code>nv show interface eth0 --rev changeset/admin/2021-06-11_16.16.41_FPKK</code> .
<code>--startup</code>	Shows configuration saved with the <code>nv config save</code> command. This is the configuration after the switch boots.
<code>--tab</code>	Show information in tab format. For example, <code>nv show interface sw1p1 --tab</code> .
<code>--view</code>	Shows different views. A view is a subset of information provided by certain <code>nv show</code> commands. To see the views available for an <code>nv show</code> command, run the command with <code>--view</code> and press TAB.

### 3.2.9.3 Action Commands

The NVUE action commands run immediate change in the switch.

The `nv action` command is in the following categories. Each command group includes subcommands. Use command completion (Tab key) to list the subcommands.

Command Group	Description
<code>nv action &lt;action&gt;</code> <code>system</code>	Run some actions on the switch. managing the software image, reboot the system ,generate tech-support etc.
<code>nv action &lt;action&gt;</code> <code>interface</code>	Run some actions on the switch. Clear counters, renew the DHCP-client on eth0 etc.
<code>nv action &lt;action&gt;</code> <code>platform</code>	Run some actions on the platform. turn-on or turn-off the UID led etc.

### 3.2.9.4 Configuration Management Commands

The NVUE configuration management commands manage and apply configurations.

Command	Description
<code>nv config apply</code>	<p>Applies the pending configuration ( <code>nv config apply</code> ) or a specific revision ( <code>nv config apply 2</code> ) to become the applied configuration. To see the list of revisions you can apply, run <code>nv config apply &lt;&lt;Tab&gt;&gt;</code> .</p> <p>You can also use these prompt options:</p> <ul style="list-style-type: none"> <li>• <code>--y</code> or <code>--assume-yes</code> to automatically reply <code>yes</code> to all prompts.</li> <li>• <code>--assume-no</code> to automatically reply <code>no</code> to all prompts.</li> </ul> <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> <p> NVOS applies but does not save the configuration; the configuration does not persist after a reboot.</p> </div> <p>You can also use these apply options:</p> <p><code>--confirm</code> applies the configuration change but you must confirm the applied configuration. If you do not confirm within ten minutes, the configuration rolls back automatically. You can change the default time with the apply <code>--confirm &lt;time&gt;</code> command. For example, <code>apply --confirm 60</code> requires you to confirm within one hour.</p> <p><code>--confirm-status</code> shows the amount of time left before the automatic rollback.</p> <p>To save the pending configuration to the startup configuration automatically when you run <code>nv config apply</code> so that you do not have to run the <code>nv config save</code> command, enable <a href="#">auto save</a>.</p>
<code>nv config detach</code>	<p>Detaches the configuration from the current pending configuration. NVOS names the detached configuration <code>pending</code> and includes a timestamp with extra characters. For example: <code>pending_20210128_212626_4WSY</code></p> <p>To list all the current detached pending configurations, run <code>nv config diff &lt;&lt;press tab&gt;</code> .</p>

Command	Description
<code>nv config diff</code> <code>&lt;revision&gt; &lt;revision&gt;</code>	Shows differences between configurations, such as the pending configuration and the applied configuration or the detached configuration and the pending configuration.
<code>nv config find</code> <code>&lt;string&gt;</code>	Finds a portion of the applied configuration according to the search string you provide. For example to find swp1 in the applied configuration, run <code>nv config find swp1</code> .
<code>nv config history</code>	Enables you to keep track of the configuration changes on the switch and shows a table with the configuration revision ID, the date and time of the change, the user account that made the change, and the type of change (such as CLI or REST API). The <code>nv config history &lt;revision&gt;</code> command shows the apply history for a specific revision.
<code>nv config patch &lt;nvue-file&gt;</code>	Updates the pending configuration with the specified YAML configuration file.
<code>nv config replace</code> <code>&lt;nvue-file&gt;</code>	Replaces the pending configuration with the specified YAML configuration file.
<code>nv config revision</code>	Shows all the configuration revisions on the switch.
<code>nv config save</code>	Overwrites the startup configuration with the applied configuration by writing to the <code>/etc/nvue.d/startup.yaml</code> file. The configuration persists after a reboot.
<code>nv config show</code>	Shows the currently applied configuration in <code>yaml</code> format. This command also shows NVUE version information.
<code>nv config show -o</code> <code>commands</code>	Shows the currently applied configuration commands.
<code>nv config diff -o</code> <code>commands</code>	Shows differences between two configuration revisions.

### 3.2.9.5 List All NVUE Commands

To show the full list of NVUE commands, run `nv list-commands`. For example:

```
admin@nvos:~$ nv list-commands
nv show platform environment
nv show platform environment fan
nv show platform environment fan <fan-id>
nv show platform environment temperature
nv show platform environment temperature <sensor-id>
nv show platform environment led
nv show platform environment led <led-id>
nv show platform software
nv show platform software installed
nv show platform software installed <installed-id>
...
```

```
admin@nvos:~$ nv list-commands system
nv show system
nv show system message
```

```
nv show system log
nv show system log files
nv show system log files <file-name>
nv show system log component
nv show system log component <component-name>
nv show system debug-log
nv show system debug-log files
nv show system debug-log files <file-name>
nv show system reboot
nv show system reboot reason
...
```

Use the Tab key to get help for the command lists you want to see. For example, to show the list of command options available for the interface eth0, run:

```
admin@nvos:~$ nv list-commands interface eth0 <<press Tab>>
ip      link      pluggable
```

You can show the list of commands for a command grouping. For example, to show the list of system commands:

### 3.2.10 Search for a Specific Configuration

To search for a specific portion of the NVUE configuration, run the `nv config find <search string>` command. The search shows all items above and below the search string. For example, to search the entire NVUE object model configuration for any mention of `ptm`:

```
admin@nvos:~$ nv config find system
- set:
  system:
    aaa:
      authentication:
        restrictions:
          fail-delay: 0
          lockout-state: disabled
      user:
        admin:
          password: '*'
      timezone: Asia/Jerusalem
```

### 3.2.11 Clear Switch Configuration

To reset the configuration on the switch back to the factory defaults, run the following command:

```
admin@nvos:~$ nv config apply empty
```

### 3.2.12 Example Configuration Commands

This section provides examples of how to configure a NVOS switch using NVUE commands.

#### 3.2.12.1 Configure the System Hostname

The example below shows the NVUE commands required to change the hostname for the switch to `nvos`:

```
admin@nvos:~$ nv set system hostname nvos
admin@nvos:~$ nv config apply
```

## 3.2.12.2 Configure an Interface

The example below shows the NVUE commands required to bring up sw1p1.

```
admin@nvos:~$ nv set interface sw1p1s1 link state up
admin@nvos:~$ nv config apply
```

## 3.2.13 Example Monitoring Commands

This section provides monitoring command examples.

### 3.2.13.1 Show Installed Software

The following example command lists the software installed on the switch:

```
admin@nvos:~$ nv show platform software installed
Installed software
-----
package                               description
-----
acl                                     access control list - utilities
acl                                     2.2.53-10
adduser                                 add and remove users and groups
adduser                                 3.118
apparmor                                user-space parser utility for AppArmor
apparmor                                2.13.6-10
apt                                      commandline package manager
apt                                      2.2.4
apt-transport-https                    transitional package for https support
apt-transport-https                    2.2.4
audisp-tacplus                          audisp module for TACACS+ accounting
audisp-tacplus                          1.0.2
auditd                                  User space tools for security auditing
auditd                                  1:3.0-2
base-files                              Debian base system miscellaneous files
base-files                              11.1+deb11u3
base-passwd                             Debian base system master password and group files
base-passwd                             3.5.51
bash                                    GNU Bourne Again SHell
bash                                    5.1-2+b3
bash-completion                         programmable completion for the bash shell
bash-completion                         1:2.11-2
bash-tacplus                             Bash TACACS+ plugin for per-command TACACS+ authorization.
bash-tacplus                             1.0.0
bridge-utils                            Utilities for configuring the Linux Ethernet bridge
bridge-utils                            1.7-1
bsdextrautils                           extra utilities from 4.4BSD-Lite
bsdextrautils                           2.36.1-8+deb11u1
bsdmainutils                             Transitional package for more utilities from FreeBSD
bsdmainutils                             12.1.7+nmu3
bsdutils                                basic utilities from 4.4BSD-Lite
bsdutils                                1:2.36.1-8+deb11u1
busybox                                  Tiny utilities for small and embedded systems
busybox                                  1:1.30.1-6+b3
ca-certificates                          Common CA certificates
ca-certificates                          20210119
...
```

### 3.2.13.2 Show Interface Configuration

The following example command shows the running, applied, and pending sw1p1s1 interface configuration.

```
admin@nvos:~$ nv show interface sw1p1s1
-----
operational      applied
-----
link
  auto-negotiate  on          on
  duplex          full
  speed           auto
counters
  in-bytes        0 Bytes
  in-pkts         0
  in-drops        0
```

```

in-errors          0
out-bytes          0 Bytes
out-pkts           0
out-drops          0
out-errors         0
in-symbol-errors  0
out-wait           0
[diagnostics]
mtu                4096
op-vls             VL0-VL7
lanes              1X,2X,4X
state              down
supported-speed    800G
max-supported-mtu  4096
physical-state     Polling
logical-state      Down
supported-lanes    1X,2X,4X
vl-capabilities    VL0-VL7
type               nv1

```

### 3.2.14 Reset NVUE Configuration to Default Values

To reset the NVUE configuration on the switch back to the default values, run the following command:

```

admin@nvos:~$ nv config replace /usr/lib/python3/dist-packages/cue_config_v1/initial.yaml
admin@nvos:~$ nv config apply

```

### 3.2.15 Add Configuration Apply Messages

When you run the `nv config apply` command, you can add a message that describes the configuration updates you make. You can see the message when you run the `nv config history` command.

To add a configuration apply message, run the `nv config apply -m <message>` command. If the message includes more than one word, enclose the message in quotes.

```

admin@nvos:~$ nv config apply -m "this is my message"

```

### 3.2.16 Configure Auto Save

By default, when you run the `nv config apply` command to apply a configuration setting, NVUE applies the pending configuration to become the applied configuration and automatically saves the changes to the startup configuration file ( `/etc/nvue.d/startup.yaml` ).

To disable auto save so that NVUE does not save applied configuration changes, run the `nv set system config auto-save enable off` command:

```

admin@nvos:~$ nv set system config auto-save enable off
admin@nvos:~$ nv config apply

```

When you disable auto save, you must run the `nv config save` command to save the applied configuration to the startup configuration so that the changes persist after a reboot.

### 3.2.17 Example Configuration Management Commands

This section provides examples of how to use the configuration management commands to apply, save, and detach configurations.

### 3.2.17.1 Apply and Save a Configuration

The following example command configures the front panel port interfaces sw1p1 state. The configuration is only in a pending configuration state. The configuration is not applied. NVUE has not yet made any changes to the running configuration.

```
admin@nvos:~$ nv set interface sw1p1 link state down
```

To apply the pending configuration to the running configuration, run the `nv config apply` command. The configuration does not persist after a reboot.

```
admin@nvos:~$ nv config apply
```

To save the applied configuration to the startup configuration, run the `nv config save` command. The configuration persists after a reboot.

```
admin@nvos:~$ nv config save
```

### 3.2.17.2 Detach a Pending Configuration

The following example configures the IP address of the eth0 interface, then detaches the configuration from the current pending configuration. NVOS saves the detached configuration to a file with a numerical value to distinguish it from other pending configurations.

```
admin@nvos:~$ nv set interface eth0 ip address 10.10.10.1
admin@nvos:~$ nv config detach
```

### 3.2.17.3 View Differences Between Configurations

To view differences between configurations, run the `nv config diff` command.

To view differences between two detached pending configurations, run the `nv config diff <TAB>` command to list all the current detached pending configurations, then run the `nv config diff` command with the pending configurations you want to diff:

```
admin@nvos:~$ nv config diff <<press Tab>>
1      2      3      4      5      6      applied empty startup
admin@nvos:~$ nv config diff 1 2
- set:
  system:
  security:
    password-hardening:
      state: disabled
```

To view differences between the applied configuration and the startup configuration:

```
admin@nvos:~$ nv config diff applied startup
```

### 3.2.17.4 Replace and Patch a Pending Configuration

The following example replaces the pending configuration with the contents of the YAML configuration file called `nv-02/13/2021.yaml` located in the `/deps` directory:

```
admin@nvos:~$ nv config replace /deps/nv-02/13/2021.yaml
```

The following example patches the pending configuration (runs the set or unset commands from the configuration in the `nv-02/13/2021.yaml` file located in the `/deps` directory):

```
admin@nvos:~$ nv config patch /deps/nv-02/13/2021.yaml
```

### 3.2.18 Passwords and Special Characters

If you use certain special characters in a password, you must quote or escape (with a backslash) these characters so that the system understands that they are part of the password.

The following table shows if you need to quote or escape a special character.

- Normal Use indicates that you can use the special character without quotes or a backslash.
- Single Quotes and Double Quotes indicate that the entire password needs to be enclosed in quotes.

Special Character	Normal Use	Single Quotes (")	Double Quotes ("" )	Escape ( \ )
backtick ( ` )	x	✓	1	✓
exclamation point ( ! )	x	✓	x	✓
semicolon ( ; )	x	✓	✓	✓
ampersand ( & )	x	✓	✓	✓
question mark ( ? )	x	✓	✓	x
tilde ( ~ )	x	✓	✓	✓
at-sign ( @ )	✓	✓	✓	✓
hash sign ( # )	x	✓	✓	✓
dollar sign ( \$ )	x	✓	x	✓
percent sign ( % )	✓	✓	✓	✓
caret ( ^ )	✓	✓	✓	✓
asterisk ( * )	✓	✓	✓	✓
parentheses ( ( ) )	x	✓	✓	✓
dash ( - )	✓	✓	✓	✓

Special Character	Normal Use	Single Quotes (")	Double Quotes ("" )	Escape ( \ )
underscore ( _ )	✓	✓	✓	✓
equals sign ( = )	✓	✓	✓	✓
plus sign ( + )	✓	✓	✓	✓
vertical bar	x	✓	✓	✓
brackets ( [ ] )	✓	✓	✓	✓
braces ( { } )	✓	✓	✓	✓
colon ( : )	✓	✓	✓	✓
single quote ( ‘ )	x	x	✓	✓
double quote ( “ )	x	✓	x	✓
comma ( , )	✓	✓	✓	✓
angle brackets ( < > )	x	✓	✓	✓
slash ( / )	✓	✓	✓	✓
dot ( . )	2	2	2	2
white space	x	x	3	x

1. Requires escape ( \ ) in addition to the double quotes ( "" ).
2. You cannot use this character at the beginning of a word.
3. A word cannot consist entirely of white space, even inside double quotes.

The following example shows a password that includes a question mark (?):

```
nvos@admin:~$ nv set system aaa user nvos password "Hello?world123"
```

The following example shows a password that includes a dot (.):

```
nvos@admin:~$ nv set system aaa user nvos password "Hello.world.123"
```

The following example shows a password that includes a dot (.) and tilde (~):

```
nvos@admin:~$ nv set system aaa user nvos password "Hello.world\~123"
```

You might need to encode special characters in a password, for example in a URL. The following table shows the special character encoding.

- ✓ indicates that encoding is not needed.
- %XX indicates that you need to replace the special character with %XX .

Symbol	Normal	Single Quotes (")	Double Quotes ("" )	Escape ( \ )
backtick ( ` )	%60	✓	%60	✓

Symbol	Normal	Single Quotes (")	Double Quotes ("" )	Escape (\ )
exclamation point ( ! )	%21	✓	%21	✓
semicolon ( ; )	%3B	✓	✓	✓
ampersand ( & )	%26	✓	✓	✓
question mark ( ? )	%3F	%3F	%3F	%3F
tilde ( ~ )	✓	✓	✓	✓
at-sign ( @ )	✓	✓	✓	✓
hash sign ( # )	%23	%23	%23	%23
dollar sign ( \$ )	%24	✓	%24	✓
percent sign ( % )	✓	✓	✓	✓
caret ( ^ )	✓	✓	✓	✓
asterisk ( * )	✓	✓	✓	✓
left parenthesis ( ( )	%28	✓	✓	✓
right parenthesis ( ) )	%29	✓	✓	✓
dash ( - )	✓	✓	✓	✓
underscore ( _ )	✓	✓	✓	✓
equals sign ( = )	✓	✓	✓	✓
plus sign ( + )	✓	✓	✓	✓
vertical bar	%7C	✓	✓	✓
left bracket ( [ )	%5B	%5B	%5B	%5B
right bracket ( ] )	%5D	%5D	%5D	%5D
braces ( { } )	✓	✓	✓	✓
colon ( : )	✓	✓	✓	✓
single quote ( ‘ )	%27	%27	✓	✓
double quote ( “ )	%22	✓	%22	✓
comma ( , )	✓	✓	✓	✓
left angle bracket ( < )	%3C	✓	✓	✓
right angle bracket ( > )	%3E	✓	✓	✓
slash ( / )	%2F	%2F	%2F	%2F
dot ( . )	✓	✓	✓	✓
white space	%20	✓	✓	✓

The following example fetches an image stored on a device with IP address 10.0.1.251 using the password `Pass#pass1` for user1:

```
nvos@admin:~$ nv action fetch system image scp://user1:Pass1%23pass1@10.0.1.251/host/nos-images/nvos-  
amd64-25.02.1857.bin
```

## 3.3 NVUE OpenAPI

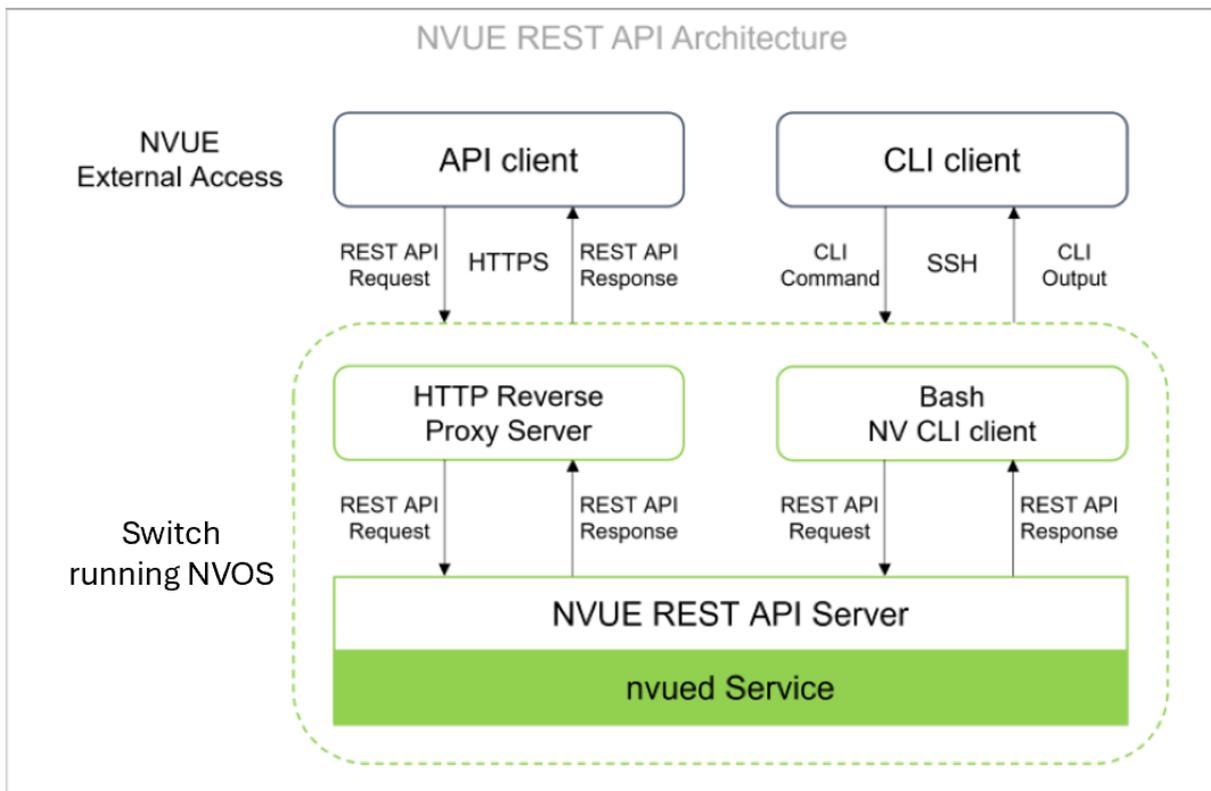
- [3.3.1 Supported HTTP Methods](#)
- [3.3.2 Secure the API](#)
  - [3.3.2.1 Certificates](#)
    - [3.3.2.1.1 Import a Certificate](#)
    - [3.3.2.1.2 Set the Certificate to Use](#)
    - [3.3.2.1.3 Delete Certificates](#)
    - [3.3.2.1.4 Show Certificate Information](#)
  - [3.3.2.2 Control Plane ACLs](#)
- [3.3.3 Supported Objects](#)
- [3.3.4 Use the API](#)
  - [3.3.4.1 API Port and Listening Address](#)
    - [3.3.4.1.1 Curl Command](#)
  - [3.3.4.2 Show NVUE REST API Information](#)
  - [3.3.4.3 Run cURL Commands](#)
- [3.3.5 API Use Cases](#)
  - [3.3.5.1 View a Configuration](#)
  - [3.3.5.2 Replace an Entire Configuration](#)
  - [3.3.5.3 Make a Configuration Change](#)
- [3.3.6 View Differences Between Configurations](#)
- [3.3.7 Troubleshoot Configuration Changes](#)
  - [3.3.7.1 Configuration Apply Fails with Warnings](#)
- [3.3.8 Save a Configuration](#)
- [3.3.9 Unset a Configuration Change](#)
- [3.3.10 Use the API for Active Monitoring](#)
- [3.3.11 Retrieve View Types](#)
- [3.3.12 Convert CLI Changes to Use the API](#)
- [3.3.13 API Examples](#)
  - [3.3.13.1 Configure the System](#)
  - [3.3.13.2 Configure Services](#)
  - [3.3.13.3 Configure Users](#)
  - [3.3.13.4 Configure an Interface](#)
  - [3.3.13.5 Action Operations](#)
- [3.3.14 Example Python Script](#)
- [3.3.15 Try the API](#)
- [3.3.16 Resources](#)
- [3.3.17 Considerations](#)
- [3.3.18 Related Information](#)
- [3.3.19 Save the Applied Configurations](#)
- [3.3.20 Send an Action](#)

This section provides information about using the NVUE API.

In addition to the CLI, NVUE supports a REST API. Instead of accessing NVOS using SSH, you can interact with the switch using an HTTP client, such as cURL or a web browser.

The `nvued` service provides access to the NVUE REST API. NVOS exposes the HTTP endpoint internally, which makes the NVUE REST API accessible locally within the NVOS switch. The NVUE CLI also communicates with the `nvued` service using internal APIs. To provide external access to the NVUE REST API, NVOS uses an HTTP reverse proxy server, and supports HTTPS and TLS connections from external REST API clients.

The following illustration shows the NVUE REST API architecture and illustrates how NVOS forwards the requests internally.



### 3.3.1 Supported HTTP Methods

The NVUE REST API supports the following methods:

- The GET method displays configuration and operational data, and is equivalent to the `nv show` commands.
- The POST method creates and submits operations. You typically use this method for `nv action` commands and for the `nv config` command to create revisions.
- The PATCH method replaces or unsets a configuration. You use this method for the `nv set` and `nv config apply` commands. You can either perform:
  - A *targeted* configuration patch to make a configuration change, where you run a specific NVUE REST API targeted at a particular OpenAPI end-point URI. Based on the NVUE schema definition, you need to direct the PATCH REST API request at a particular

endpoint (for example, `/nvue_v1/interface/<interface-id>/link/mtu`) and provide the payload that conforms to the schema. With a targeted configuration patch, you can control individual resources.

- A *root patch*, where you run the NVUE PATCH API on the root node of the schema so that a single PATCH operation can change one, some, or the entire configuration in a single payload. The payload of the PATCH method must be aware of the entire NVUE object model schema because you make the configuration changes relative to the root node `/nvue_v1`. You typically perform a *root patch* to push all configurations to the switch in bulk; for example, if you use an SDN controller or a network management system to push the entire switch configuration every time you need to make a change, regardless of how small or large. A root patch can also make configuration changes with fewer round trips to the switch.
- The input payload in a PATCH request can have either a `set` or `unset` json object for the same resource, but not both. The order in which the API executes the `set` and `unset` objects is not deterministic and not supported.
- The DELETE method deletes a configuration and is equivalent to the `nv unset` commands.

## 3.3.2 Secure the API

The NVUE REST API supports HTTP basic authentication, and the same underlying authentication methods for username and password that the NVUE CLI supports. User accounts work the same on both the API and the CLI.

### 3.3.2.1 Certificates

NVOS includes a self-signed certificate and private key to use on the server so that it works out of the box. The switch generates the self-signed certificate and private key when it boots for the first time. The X.509 certificate with the public key is in `/etc/ssl/certs/nvue.pem` and the corresponding private key is in `/etc/ssl/private/nvue.key`.

NVIDIA recommends you use your own certificates and keys. For the steps to generate self-signed certificates and keys, refer to the [Ubuntu Certificates and Security documentation](#).

NVOS lets you manage CA certificates (such as DigiCert or Verisign) and entity (end-point) certificates. Both a CA certificate and an entity certificate can contain a chain of certificates.

You can import certificates onto the switch (fetch certificates from an external source), set which certificate you want to use for the NVUE REST API, and show information about a certificate, such as the serial number, and the date and time during which the certificate is valid.

#### 3.3.2.1.1 Import a Certificate



- You can import a maximum of 25 entity certificates and a maximum of 25 CA certificates.
- The certificate you import contains sensitive private key information. NVIDIA recommends that you use a secure transport such as SFTP, SCP, or HTTPS.

- To import an entity certificate, run an `nv action import system security certificate <cert-id>` command.
- To import a CA certificate, run an `nv action import system security ca-certificate <cert-id>` command.

If the certificate is passphrase protected, you need to include the passphrase.

You must provide a certificate ID ( `<cert-id>` ) to uniquely identify the certificate you import.

The following example imports a CA certificate with a public key and calls the certificate `tls-cert-1`. The certificate is passphrase protected with `mypassphrase`. The public key is a Base64 ASCII encoded PEM string.

```
nvos@switch:~$ nv action import system security ca-certificate tls-cert-1 passphrase mypassphrase data "<public-key>"
```

The following example imports an entity certificate bundle and calls the certificate `tls-cert-1`. The certificate bundle is passphrase protected with `mypassphrase`.

A certificate bundle must be in .PFX or .P12 format.

```
nvos@switch:~$ nv action import system security certificate tls-cert-1 passphrase mypassphrase uri-bundle scp://user@pass:1.2.3.4:/opt/certs/cert.p12
```

The following example imports an entity certificate with the public key URI `scp://user@pass:1.2.3.4` and private key URI `scp://user@pass:1.2.3.4`, and calls the certificate `tls-cert-1`. The certificate is not passphrase protected.

A CA certificate must be in .pem, .p7a, or .p7c format.

```
nvos@switch:~$ nv action import system security certificate tls-cert-1 uri-public-key scp://user@pass:1.2.3.4 uri-private-key scp://user@pass:1.2.3.4
```

### 3.3.2.1.2 Set the Certificate to Use

You can configure the NVUE REST API to use a specific certificate.

The following example configures the API to use the certificate `tls-cert-1`:

```
nvos@switch:~$ nv set system api certificate tls-cert-1
nvos@switch:~$ nv config apply
```

The following example configures the API to use the self-signed certificate:

```
nvos@switch:~$ nv set system api certificate self-signed
nvos@switch:~$ nv config apply
```

To unset the certificate to use with the NVUE REST API:

```
nvos@switch:~$ nv unset system api certificate tls-cert-1
```

### 3.3.2.1.3 Delete Certificates

- To delete an entity certificate and the key data stored on the switch, run the `nv action delete system security certificate <cert-id>` command.
- To delete a CA certificate and the key data stored on the switch, run the `nv action delete system security ca-certificate <cert-id>` command.

The following command deletes the certificate `tls-cert-1`:

```
nvos@switch:~$ nv action delete system security certificate tls-cert-1
```

### 3.3.2.1.4 Show Certificate Information

- To show all the entity certificates on the switch, run the `nv show system security certificate` command.
- To show all the CA certificates on the switch, run the `nv show system security ca-certificate` command.

The following example shows all the entity certificates on the switch:

```
nvos@switch:~$ nv show system security certificate
```

- To show the applications that are using a specific entity certificate, run the `nv show system security certificate <cert-id> installed` command.
- To show the applications that are using a specific CA certificate, run the `nv show system security ca-certificate <cert-id> installed` command.

The following example shows the applications that are using a specific entity certificate.

```
nvos@switch:~$ nv show system security certificate tls-cert-1 installed
```

- To show detailed information about a specific entity certificate, run the `nv show system security certificate <cert-id> dump` command.
- To show detailed information about a specific CA certificate, run the `nv show system security ca-certificate <cert-id> dump` command.

The following example shows detailed information about the CA certificate `tls-cert-1`:

```
nvos@switch:~$ nv show system security ca-certificate tls-cert-1 dump
```

## 3.3.2.2 Control Plane ACLs

You can secure the API by configuring:

- A listening address; see [API Port and Listening Address](#) below.
- Control plane ACLs; see the following example.

This example shows how to create ACLs to allow users from the management subnet and the local switch to communicate with the switch using REST APIs, and restrict all other access.

```

nvos@switch:~$ nv set acl API-PROTECT type ipv4
nvos@switch:~$ nv set acl API-PROTECT rule 10 action permit
nvos@switch:~$ nv set acl API-PROTECT rule 10 match ip .protocol tcp .dest-port 8765 .source-ip 192.168.200.0/24
nvos@switch:~$ nv set acl API-PROTECT rule 10 remark "Allow the Management Subnet to talk to API"
nvos@switch:~$ nv set acl API-PROTECT rule 20 action permit
nvos@switch:~$ nv set acl API-PROTECT rule 20 match ip .protocol tcp .dest-port 8765 .source-ip 127.0.0.1
nvos@switch:~$ nv set acl API-PROTECT rule 20 remark "Allow the local switch to talk to the API"
nvos@switch:~$ nv set acl API-PROTECT rule 30 action deny
nvos@switch:~$ nv set acl API-PROTECT rule 30 match ip .protocol tcp .dest-port 8765
nvos@switch:~$ nv set acl API-PROTECT rule 30 remark "Block everyone else from talking to the API"
nvos@switch:~$ nv set system control-plane acl API-PROTECT inbound

```

### 3.3.3 Supported Objects

The NVUE object model supports most features on the NVOS switch. The following list shows the supported objects. The NVUE API supports more objects within each of these objects. You can find a full listing of the supported API endpoints [here](#).

High-level Objects	Description
<b>acl</b>	Access control lists.
<b>interface</b>	Interface configuration.
<b>platform</b>	Platform configuration, such as hardware and software components.
<b>system</b>	Global system settings, such as system login messages, switch reboot history, syslog, ntp etc..
<b>vrf</b>	Get VRF.
<b>ib</b>	IB Devices.

### 3.3.4 Use the API

The NVUE CLI and the REST API are equivalent in functionality; you can run all management operations from the REST API or from the CLI. The NVUE object model drives both the REST API and the CLI management operations. All operations are consistent; for example, the CLI `nv show commands` reflect any PATCH operation (create and update) you run through the REST API.

NVUE follows a declarative model, removing context-specific commands and settings. The structure of NVUE is like a big tree that represents the entire state of a NVOS instance. At the base of the tree are high level branches representing objects, such as system and interface. Under each of these branches are more branches. As you navigate through the tree, you gain a more specific context. At the leaves of the tree are actual attributes, represented as key-value pairs. The path through the tree is similar to a filesystem path.

NVOS enables the NVUE REST API by default. To disable the NVUE REST API, run the `nv set system api state disabled` command.



To use the NVUE REST API in NVOS, you must change the password for the admin user; otherwise you see 403 responses when you run commands.

### 3.3.4.1 API Port and Listening Address

This section shows how to:

- Set the NVUE REST API port. If you do not set a port, NVOS uses the default port 8765.
- Specify the NVUE REST API listening address; you can specify an IPv4 address, IPv6 address, or `localhost`. If you do not specify a listening address, NGINX listens on all addresses for the target port.

#### NVUE Commands

The following example sets the port to 8888:

```
nvos@switch:~$ nv set system api port 8888
nvos@switch:~$ nv config apply
```

You can listen on multiple interfaces by specifying different listening addresses:

```
nvos@switch:~$ nv set system api listening-address 10.10.10.1
nvos@switch:~$ nv set system api listening-address 10.10.20.1
nvos@switch:~$ nv config apply
```

The following example configures the listening address on eth0, which has IP address 172.0.24.0 and uses the management VRF by default:

```
nvos@switch:~$ nv set system api listening-address 172.0.24.0
nvos@switch:~$ nv config apply
```

#### 3.3.4.1.1 Curl Command

The following example sets the port to 8888:

```
nvos@switch:~$ curl -u 'admin:admin' -k --request PATCH https://localhost:8765/nvue_v1/system/api?rev=2 -H
'Content-Type:application/json' -d '{"port": 8888 }'
```

You can listen on multiple interfaces by specifying different listening addresses. The following example sets localhost, interface address 10.10.10.1, and 10.10.20.1 as listen-addresses.

```
nvos@switch:~$ curl -u 'admin:admin' -k --request PATCH https://localhost:8765/nvue_v1/system/api/listening-
address?rev=2 -H 'Content-Type:application/json' -d '{"localhost": {}, "10.10.10.1": {}, "10.10.20.1": {}}'
```

The following example configures the listening address on eth0, which has IP address 172.0.24.0 and uses the management VRF by default:

```
nvos@switch:~$ curl -u 'admin:admin' -k --request PATCH https://localhost:8765/nvue_v1/system/api/listening-
address?rev=2 -H 'Content-Type:application/json' -d '{"172.0.24.0": {}}'
```

### 3.3.4.2 Show NVUE REST API Information

To show REST API port configuration, state (enabled or disabled), certificate, listening address, and connection information:

## NVUE Commands

Run the `nv show system api` command:

```
nvos@switch:~$ nv show system api
----- operational applied -----
state          enabled        enabled
port           443            443
certificate    self-signed    self-signed
[listening-address] any
connections
  active        1
  accepted      2
  handled       2
  requests      2
  reading       0
  writing        1
  waiting       0
```

To show connection information only, run the `nv show system api connections` command:

```
nvos@switch:~$ nv show system api connections
----- operational -----
active        2
accepted      8
handled       8
requests      8
reading       0
writing        2
waiting       0
```

To show the configured listening address, run the `nv show system api listening-address` command:

```
nvos@switch:~$ nv show system api listening-address
-----
localhost
```

To show all the certificates installed on the switch, run the `nv show system security certificate` command. To show information about a specific certificate, such as the serial number and how long the certificate is valid, run the `nv show system security certificate <certificate>` command:

```
nvos@switch:~$ nv show system security certificate tls-cert-1
----- operational applied -----
installed
app          TLS
serial-number 67:03:3B:B4:6E:35:D3
valid-from   2023-02-14T00:35:18+00:00
valid-to     2033-02-11T00:35:18+00:00
```

## Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -k --request GET https://localhost:443/nvue_v1/system/api?rev=2 -H "accept: application/json"
{
  "certificate": "self-signed",
  "listening-address": {
    "10.10.10.1": {},
    "10.10.20.1": {},
    "172.0.24.0": {},
    "localhost": {}
  },
  "port": 8888,
  "state": "enabled"
}
```

To show the configured listening address:

```
nvos@switch:~$ curl -u 'admin:admin' -k --request GET https://localhost:443/nvue_v1/system/api/listening-address?rev=2 -H "accept: application/json"
{
  "10.10.10.1": {},
  "10.10.20.1": {}
}
```

To show the certificates on the switch:

```
nvos@switch:~$ curl -u 'admin:admin' -k --request GET https://localhost:443/nvue_v1/system/api/certificate?rev=2 -H "accept: application/json"
{
  "tls-cert-1": {},
  "tls-cert-2": {}
}
```

To show information about a specific certificate, such as the serial number and how long the certificate is valid:

```
nvos@switch:~$ curl -u 'admin:admin' -k --request GET https://localhost:443/nvue_v1/system/api/certificate/tls-cert-1?rev=2 -H "accept: application/json"
{
  "serial-number": "67:03:3B:B4:6E:35:D3",
  "valid-from": "2023-02-14T00:35:18+00:00",
  "valid-to": "2033-02-11T00:35:18+00:00"
}
```

### 3.3.4.3 Run cURL Commands

You can run the cURL commands from the command line. Use the username and password for the switch. For example:

```
nvos@switch:~$ curl -u 'admin:admin' --insecure https://127.0.0.1:443/nvue_v1/interface
{
  "eth0": {
    "ip": {
      "address": {
        "192.168.200.12/24": {}
      }
    },
    "link": {
      "mtu": 1500,
      "state": {
        "up": {}
      }
    },
    "stats": {
      "carrier-transitions": 2,
      "in-bytes": 184151,
      "in-drops": 0,
      "in-errors": 0,
      "in-pkts": 2371,
      "out-bytes": 117506,
      "out-drops": 0,
      "out-errors": 0,
      "out-pkts": 762
    }
  }
  ...
}
```

## 3.3.5 API Use Cases

The following examples show the primary API uses cases.

### 3.3.5.1 View a Configuration

Use the following example to obtain the current applied configuration on the switch. Change the `rev` argument to view any revision. Possible options for the `rev` argument include `startup`, `pending`, `operational`, and `applied`.

Curl Command

```

nvos@switch:~$ curl -k -u 'admin:admin' -X GET "https://127.0.0.1:443/nvue_v1/?rev=applied&filled=false"
{"interface": {
  "eth0": {
    "acl": {
      "ACL_MGMT_INBOUND_CP_DEFAULT": {
        "inbound": {
          "control-plane": {}
        }
      },
      "ACL_MGMT_INBOUND_CP_DEFAULT_IPV6": {
        "inbound": {
          "control-plane": {}
        }
      },
      "ACL_MGMT_INBOUND_DEFAULT": {
        "inbound": {}
      },
      "ACL_MGMT_INBOUND_DEFAULT_IPV6": {
        "inbound": {}
      },
      "ACL_MGMT_OUTBOUND_CP_DEFAULT": {
        "outbound": {
          "control-plane": {}
        }
      },
      "ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6": {
        "outbound": {
          "control-plane": {}
        }
      }
    },
    "type": "eth"
  },
  ...
}

```

## Python Code

```

#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

if __name__ == "__main__":
    r = requests.get(url=nvue_end_point + "/?rev=applied&filled=false",
                    auth=auth,
                    verify=False)
    print("====Current Applied Revision====")
    print(json.dumps(r.json(), indent=2))

```

## NVUE CLI

```

nvos@switch:~$ nv config show
- set:
  fae:
    system:
      events:
        table-size: 600
  interface:
    eth0-1:
      acl:
        ACL_MGMT_INBOUND_CP_DEFAULT:
          inbound:
            control-plane: {}
        ACL_MGMT_INBOUND_CP_DEFAULT_IPV6:
          inbound:
            control-plane: {}
        ACL_MGMT_INBOUND_DEFAULT:
          inbound: {}
        ACL_MGMT_INBOUND_DEFAULT_IPV6:
          inbound: {}
        ACL_MGMT_OUTBOUND_CP_DEFAULT:
          outbound:
            control-plane: {}
        ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6:
          outbound:
            control-plane: {}
      type: eth
  ...

```

### 3.3.5.2 Replace an Entire Configuration

To replace an entire configuration:

1. Create a new revision ID with a POST:

```
nvos@switch:~$ curl -u 'admin:admin' --insecure -X POST https://127.0.0.1:443/nvue_v1/revision
{
  "1": {
    "state": "pending",
    "transition": {
      "issue": {},
      "progress": ""
    }
  }
}
```

2. Record the revision ID. In the above example, the revision ID is "1".
3. Do a root patch to delete the whole configuration.

```
nvos@switch:~$ curl -u 'admin:admin' -d '{}' -H 'Content-Type: application/json' -k -X DELETE https://127.0.0.1:443/nvue_v1/?rev=1
{}

```

4. Do a root patch to update the switch with the new configuration.

```
nvos@switch:~$ curl -u 'admin:admin' -d '{
  "system": {
    "hostname": "switch01"
  },
  "interface": {
    "eth0": {
      "ip": {
        "address": {
          "192.168.200.6/24": {}
        },
        "type": "eth"
      }
    }
  }
}' -H 'Content-Type: application/json' -k -X PATCH https://127.0.0.1:443/nvue_v1/?rev=1
{}

```

5. Apply the changes with a PATCH to the revision changeset.

Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -H 'Content-Type:application/json' -d '{"state": "apply", "auto-prompt": {"ays": "ays_yes"}}' -k -X PATCH https://127.0.0.1:443/nvue_v1/revision/1
{
  "state": "apply",
  "transition": {
    "issue": {},
    "progress": ""
  }
}
```

NVUE CLI

```
nvos@switch:~$ nv config apply
```

6. Review the status of the apply and the configuration:

Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -k -X GET https://127.0.0.1:443/nvue_v1/revision/1
{
  "state": "applied",
  "transition": {
    "issue": {},
    "progress": ""
  }
}
```

```

nvos@switch:~$ curl -u 'admin:admin' --insecure https://127.0.0.1:443/nvue_v1/system
{
  "build": "NVOS Debian GNU/Linux 11 (bullseye)",
  "date-time": "2024-08-05 07:42:28",
  "health-status": "OK",
  "hostname": "switch",
  "platform": "x86_64-nvidia_q3200_ra-r0",
  "product-name": "nvos",
  "product-release": "25.02.0936-010",
  "status": "System is ready",
  "swap-memory": "0 MB used / 0 MB free / 0 MB total",
  "system-memory": "3250 MB used / 12564 MB free / 15814 MB total",
  "timezone": "Asia/Jerusalem",
  "uptime": 11659,
  "version": {
    "build-date": "Fri Jul 19 14:24:17 UTC 2024",
    "image": "nvos-25.02.0936-010",
    "kernel": "5.10.0-23-2-amd64",
    "onie": "2023.11-5.3.0012-rc6-115200-dev"
  }
}

```

## Python Code

```

#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
    print("Headers:", r.headers)
    print("Body:", r.body)

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def create_nvue_changset():
    r = requests.post(url=nvue_end_point + "/revision",
                     auth=auth,
                     verify=False)
    print_request(r.request)
    print_response(r)
    response = r.json()
    changeset = response.popitem()[0]
    return changeset

def apply_nvue_changeset(changeset):
    apply_payload = {"state": "apply", "auto-prompt": {"ays": "ays_yes"}}
    url = nvue_end_point + "/revision/" + requests.utils.quote(changeset,
                                                                safe="")
    r = requests.patch(url=url,
                      auth=auth,
                      verify=False,
                      data=json.dumps(apply_payload),
                      headers=mime_header)
    print_request(r.request)
    print_response(r)

def is_config_applied(changeset) -> bool:
    # Check if the configuration was indeed applied
    global RETRIES
    global POLL_APPLIED
    retries = RETRIES
    while retries > 0:
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),
                        auth=auth,
                        verify=False)
        response = r.json()
        print(response)
        if response["state"] == "applied":
            return True
        retries -= 1
        time.sleep(POLL_APPLIED)
    return False

def apply_new_config(path,payload):
    # Create a new revision ID
    changeset = create_nvue_changset()
    print("Using NVUE Changeset: {}".format(changeset))

    # Delete existing configuration

```

```

query_string = {"rev": changeset}
r = requests.delete(url=nvue_end_point + path,
                    auth=auth,
                    verify=False,
                    params=query_string,
                    headers=mime_header)
print_request(r.request)
print_response(r)

# Patch the new configuration
query_string = {"rev": changeset}
r = requests.patch(url=nvue_end_point + path,
                   auth=auth,
                   verify=False,
                   data=json.dumps(payload),
                   params=query_string,
                   headers=mime_header)
print_request(r.request)
print_response(r)

# Apply the changes to the new revision changeset
apply_nvue_changeset(changeset)

# Check if the changeset was applied
is_config_applied(changeset)

def nvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                    auth=auth,
                    verify=False)
    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    payload = {
        "system": {
            "hostname": "switch01"
        },
        "interface": {
            "eth0": {
                "ip": {
                    "address": {
                        "192.168.200.6/24": {}
                    },
                },
            },
            "type": "eth"
        }
    }
    apply_new_config("/", payload)
    time.sleep(DUMMY_SLEEP)
    print("====Verifying some of the configurations====")
    nvue_get("/system")
    nvue_get("/interface")

```

## NVUE CLI

```

nvos@switch:~$ nv show system
operational
-----
build          NVOS Debian GNU/Linux 11 (bullseye)
uptime         3:20:04
hostname       croc-88-mgmt2
product-name   nvos
product-release 25.02.0936-010
platform       x86_64-nvidia_q3200_ra-r0
system-memory  3268 MB used / 12546 MB free / 15814 MB total
swap-memory    0 MB used / 0 MB free / 0 MB total
health-status  Not OK
date-time      2024-08-05 07:48:13
status         System is ready
timezone       Asia/Jerusalem

```

### 3.3.5.3 Make a Configuration Change

To make a configuration change:

1. Create a new revision ID with a POST:

```

nvos@switch:~$ curl -u 'admin:admin' --insecure -X POST https://127.0.0.1:443/nvue_v1/revision
{
  "2": {
    "state": "pending",
    "transition": {
      "issue": {},
      "progress": ""
    }
  }
}

```

- Record the revision ID. In the above example, the revision ID is "2".
- Make the change with a PATCH and link it to the revision ID:

#### Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -d '{"99.99.99.99/32": {}}' -H 'Content-Type: application/json' -k -X PATCH https://127.0.0.1:443/nvue_v1/interface/eth0/ip/address?rev=2
{
  "99.99.99.99/32": {}
}
```

#### NVUE CLI

```
nvos@switch:~$ nv set interface eth0 ip address 99.99.99.99/32
```

- Apply the changes with a PATCH to the revision changeset:

#### Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -H 'Content-Type:application/json' -k -X PATCH https://127.0.0.1:443/nvue_v1/revision/2
{
  "state": "apply",
  "transition": {
    "issue": {},
    "progress": ""
  }
}
```

#### NVUE CLI

```
nvos@switch:~$ nv config apply
```

- Review the status of the apply and the configuration:

#### Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -k -X GET https://127.0.0.1:443/nvue_v1/revision/2
{
  "state": "applied",
  "transition": {
    "issue": {},
    "progress": ""
  }
}
```

```
nvos@switch:~$ curl -u 'admin:admin' -k -X GET https://127.0.0.1:443/nvue_v1/revision/2
{
  "state": "applied",
  "transition": {
    "issue": {},
    "progress": ""
  }
}
```

#### Python Code

```
#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
```

```

print("Headers:", r.headers)
print("Body:", r.body)

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def create_nvue_changest():
    r = requests.post(url=nvue_end_point + "/revision",
                     auth=auth,
                     verify=False)
    print_request(r.request)
    print_response(r)
    response = r.json()
    changeset = response.popitem()[0]
    return changeset

def apply_nvue_changeset(changeset):
    apply_payload = {"state": "apply", "auto-prompt": {"ays": "ays_yes"}}
    url = nvue_end_point + "/revision/" + requests.utils.quote(changeset,
                                                                safe="")

    r = requests.patch(url=url,
                       auth=auth,
                       verify=False,
                       data=json.dumps(apply_payload),
                       headers=mime_header)

    print_request(r.request)
    print_response(r)

def is_config_applied(changeset) -> bool:
    # Check if the configuration was indeed applied
    global RETRIES
    global POLL_APPLIED
    retries = RETRIES
    while retries > 0:
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),
                         auth=auth,
                         verify=False)

        response = r.json()
        print(response)
        if response["state"] == "applied":
            return True
        retries -= 1
        time.sleep(POLL_APPLIED)

    return False

def apply_new_config(path,payload):
    # Create a new revision ID
    changeset = create_nvue_changest()
    print("Using NVUE Changeset: {}".format(changeset))

    # Delete existing configuration
    query_string = {"rev": changeset}
    r = requests.delete(url=nvue_end_point + path,
                       auth=auth,
                       verify=False,
                       params=query_string,
                       headers=mime_header)

    print_request(r.request)
    print_response(r)

    # Patch the new configuration

    query_string = {"rev": changeset}
    r = requests.patch(url=nvue_end_point + path,
                       auth=auth,
                       verify=False,
                       data=json.dumps(payload),
                       params=query_string,
                       headers=mime_header)

    print_request(r.request)
    print_response(r)

    # Apply the changes to the new revision changeset
    apply_nvue_changeset(changeset)

    # Check if the changeset was applied
    is_config_applied(changeset)

def nvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                    auth=auth,
                    verify=False)

    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    payload = {
        "99.99.99.99/32": {}
    }
    apply_new_config("/interface/eth0/ip/address",payload)
    time.sleep(DUMMY_SLEEP)
    nvue_get("/interface/eth0/ip/address")

```

## NVUE CLI

```

nvos@switch:~$ nv show interface eth0 ip address
-----
99.99.99.99/32
127.0.0.1/8
::1/128

```

### 3.3.6 View Differences Between Configurations

To view differences between configurations, run the API `GET /nvue_v1/<resource>?rev=<rev-A>&diff=<rev-B>` method with the configurations you want to `diff`. This method is equivalent to the NVUE `nv config diff <rev-A> <rev-B>` command.

To see the difference between the startup revision and the applied revision:

```
$ curl -u 'admin:admin' --insecure -X GET /nvue_v1/interface?rev=startup&diff=applied
```

To see the difference between revision 1 and revision 2:

```
$ curl -u 'admin:admin' --insecure -X GET /nvue_v1/<resource>?rev=1&diff=2
```

The order of the revisions can be changed; for example, `GET /nvue_v1/<resource>?rev=2&diff=1`.

### 3.3.7 Troubleshoot Configuration Changes

When a configuration change fails, you see an error in the change request.

#### 3.3.7.1 Configuration Apply Fails with Warnings

In some cases, such as the first push with NVUE or if you change a file manually instead of using NVUE, you see a warning prompt and the apply fails.

```

nvos@switch:~$ curl -u 'admin:admin' --insecure -X GET https://127.0.0.1:443/nvue_v1/revision/6
{
  "6": {
    "state": "ays_fail",
    "transition": {
      "issue": {
        "0": {
          "code": "client_timeout",
          "data": {},
          "message": "Timeout while waiting for client response",
          "severity": "error"
        }
      },
      "progress": "Aborted apply after warnings"
    }
  }
}

```

To resolve this issue, observe the failures or errors, then inspect the configuration that you are trying to apply. After you resolve the errors, retry the API. If you prefer to overlook the errors and force an apply, add `"auto-prompt":{"ays": "ays_yes"}` to the configuration apply.

```

nvos@switch:~$ curl -u 'admin:admin' -d '{"state":"apply","auto-prompt":{"ays": "ays_yes"}}' -H 'Content-Type:application/json' --insecure -X PATCH https://127.0.0.1:443/nvue_v1/revision/6

```

## 3.3.8 Save a Configuration

To save an applied configuration change to the startup configuration file ( `/etc/nvue.d/startup.yaml` ) so that the changes persist after a reboot, use a PATCH to the applied revision with the `save` state.

### Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -k -X PATCH -d '{"state": "save", "auto-prompt": {"ays": "ays_yes"}}' -H
'Content-Type: application/json' https://127.0.0.1:443/nvue_v1/revision/applied
{
  "state": "save",
  "transition": {
    "issue": {},
    "progress": ""
  }
}
```

### Python Code

```
#!/usr/bin/env python3
import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
    print("Headers:", r.headers)
    print("Body:", r.body)

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def save_nvue_changeset():
    apply_payload = {"state": "save", "auto-prompt": {"ays": "ays_yes"}}
    url = nvue_end_point + "/revision/applied"
    r = requests.patch(url=url,
                      auth=auth,
                      verify=False,
                      data=json.dumps(apply_payload),
                      headers=mime_header)

    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    save_nvue_changeset()
```

### NVUE CLI

```
nvos@switch:~$ nv config save
saved
```

## 3.3.9 Unset a Configuration Change

To unset a change, use the `null` value to the key. For example, to delete start configuration from eth1 port, use the following syntax:

```
nvos@switch:~$ curl -u 'admin:admin' -d '{"eth1":null}' -H 'Content-Type: application/json' --insecure -X PATCH https://127.0.0.1:443/nvue_v1/interface/rev=4
```

When you unset a change, you must still use the `PATCH` action. The value indicates removal of the entry. The data is `{"eth1":null}` with the `PATCH` action.

### 3.3.10 Use the API for Active Monitoring

The example below fetches the counters for `interface sw1p1`.

#### Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -k -X GET https://127.0.0.1:443/nvue_v1/interface/sw1p1/link/counters
{
  "buffer-overflow-errors": 0,
  "in-bytes": 16128,
  "in-drops": 0,
  "in-errors": 0,
  "in-pkts": 56,
  "in-symbol-errors": 0,
  "link-downed": 0,
  "link-error-recovery": 0,
  "local-link-integrity-errors": 0,
  "out-bytes": 16128,
  "out-drops": 0,
  "out-errors": 0,
  "out-pkts": 56,
  "out-wait": 0,
  "port-rcv-constraint-errors": 0,
  "port-rcv-remote-physical-errors": 0,
  "port-rcv-switch-relay-errors": 0,
  "qpl-drops": 0,
  "rcv-icrc-errors": 0,
  "tx-parity-errors": 0
}
```

#### Python Code

```
#!/usr/bin/env python3
import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

if __name__ == "__main__":
    r = requests.get(url=nvue_end_point + "/interface/sw1p1/link/counters",
                    auth=auth,
                    verify=False)
    print("====Interface sw1p1 Counters====")
    print(json.dumps(r.json(), indent=2))
```

#### NVUE CLI

```
nvos@switch:~$ nv show interface sw1p1 link counters --view detail
operational description
-----
in-bytes          15.75 KB      Total number of bytes received on the interface
in-pkts           56             Total number of packets received on the interface
in-drops          0             Number of received packets dropped
in-errors         0             Number of received packets with errors
out-bytes         15.75 KB      Total number of bytes transmitted out of the interface
out-pkts          56             Total number of packets transmitted out of the interface
out-drops         0             The number of outbound packets that were chosen to be discarded even
                    though no errors had been detected to prevent their being
                    transmitted.
out-errors        0             The number of outbound packets that could not be transmitted because
                    of errors.
in-symbol-errors  0             Total number of minor errors detected on one or more physical lanes
out-wait          0             The number of ticks during which the port selected by PortSelect had
                    data to transmit but no data was sent during the entire tick because
                    of insufficient credits or of lack of arbitration.
link-error-recovery
successfuly       0             Total number of times the Port Training state machine has
                    completed the link error recovery process
link-downed       0             Total number of times the Port Training state machine has failed the
```

port-rcv-remote-physical-errors	0	link error recovery process and downed the link
port-rcv- <b>switch</b> -relay-errors	0	Total number of packets marked with the EBP delimiter received on the port
port-rcv-constraint-errors	0	Total number of packets received on the port that were discarded because they could not be forwarded by the <b>switch</b> relay
local-link-integrity-errors exceeded	0	Total number of packets received on the <b>switch</b> physical port that are discarded due to constraints
qpl-drops limitations	0	Total number of times that the count of local physical errors exceeded
buffer-overflow-errors	0	the threshold specified by Local Phy Errors
rcv-icrc-errors	0	Total number of QP1 MADs (packets) dropped due to resource
tx-parity-errors	0	The number of times that Overrun-Errors consecutive flow control update periods occur
		Total number of icrc payload corruption rx errors
		Total number of icrc payload corruption tx parity errors

### 3.3.11 Retrieve View Types

NVUE provides views for certain `show` commands. A view is a subset of information.

To see the views available for a `show` command, run the command with `--view` and press TAB:

```
nvos@switch:~$ nv show interface --view <<TAB>>
brief          detail          link-diagnostics  lldp          mac          small
```

To retrieve view types through the REST API, you use the `curl -u 'admin:admin' -k -X GET http://path?view=<brief>` syntax. For example, the equivalent REST API method for the NVUE `nv show interface --view=brief` command is:

```
nvos@switch:~$ curl -u 'admin:admin' -k -X GET https://127.0.0.1:443/nvue_v1/interface?view=brief
```

### 3.3.12 Convert CLI Changes to Use the API

You can take a configuration change from the CLI and use the API to configure the same set of changes.

1. Make your configuration changes on the system with the NVUE CLI.

```
nvos@switch:~$ nv set system hostname switch01
nvos@switch:~$ nv set interface eth0 ip address 192.168.200.6/24
```

2. View the changes as a JSON blob.

```
nvos@switch:~$ nv config diff -o json
[
  {
    "set": {
      "interface": {
        "eth0": {
          "ip": {
            "address": {
              "192.168.200.6/24": {}
            }
          }
        }
      },
      "system": {
        "hostname": "switch01"
      }
    }
  }
]
```

3. Staple the JSON blob to a root patch request as the payload.

```

nvos@switch:~$ curl -u 'admin:admin' -d '{
  "interface": {
    "eth0": {
      "ip": {
        "address": {
          "192.168.200.6/24": {}
        }
      }
    }
  },
  "system": {
    "hostname": "switch01"
  }
}' -k -X PATCH https://127.0.0.1:443/nvue_v1/?rev=3

```

#### 4. Apply the changes with a PATCH to the revision changeset.

```

nvos@switch:~$ curl -u 'admin:admin' -H 'Content-Type:application/json' -k -d '{"state": "apply", "auto-prompt": {"ays": "ays_yes"}}' -X PATCH https://127.0.0.1:443/nvue_v1/revision/3
{
  "state": "apply",
  "transition": {
    "issue": {},
    "progress": ""
  }
}

```

#### 5. Review the status of the apply and the configuration:

##### Curl Command

```

nvos@switch:~$ curl -u 'admin:admin' -k -X GET https://127.0.0.1:443/nvue_v1/revision/3
{
  "state": "applied",
  "transition": {
    "issue": {},
    "progress": ""
  }
}

```

##### Python Code

```

#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
    print("Headers:", r.headers)
    print("Body:", r.body)

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def create_nvue_changest():
    r = requests.post(url=nvue_end_point + "/revision",
                     auth=auth,
                     verify=False)
    print_request(r.request)
    print_response(r)
    response = r.json()
    changeset = response.popitem()[0]
    return changeset

def apply_nvue_changest(changeset):
    # apply_payload = {"state": "apply"}
    apply_payload = {"state": "apply", "auto-prompt": {"ays": "ays_yes"}}
    url = nvue_end_point + "/revision/" + requests.utils.quote(changeset,
                                                                safe="")
    r = requests.patch(url=url,
                       auth=auth,
                       verify=False,

```

```

        data=json.dumps(apply_payload),
        headers=mime_header)
print_request(r.request)
print_response(r)

def is_config_applied(changeset) -> bool:
    # Check if the configuration was indeed applied
    global RETRIES
    global POLL_APPLIED
    retries = RETRIES
    while retries > 0:
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),
                        auth=auth,
                        verify=False)
        response = r.json()
        print(response)

        if response["state"] == "applied":
            return True
            retries -= 1
            time.sleep(POLL_APPLIED)

    return False

def apply_new_config(path,payload):
    # Create a new revision ID
    changeset = create_nvue_changeset()
    print("Using NVUE Changeset: '{}'".format(changeset))

    # Delete existing configuration
    query_string = {"rev": changeset}
    r = requests.delete(url=nvue_end_point + path,
                       auth=auth,
                       verify=False,
                       params=query_string,
                       headers=mime_header)

    print_request(r.request)
    print_response(r)

    # Patch the new configuration

    query_string = {"rev": changeset}
    r = requests.patch(url=nvue_end_point + path,
                      auth=auth,
                      verify=False,
                      data=json.dumps(payload),
                      params=query_string,
                      headers=mime_header)

    print_request(r.request)
    print_response(r)

    # Apply the changes to the new revision changeset
    apply_nvue_changeset(changeset)

    # Check if the changeset was applied
    is_config_applied(changeset)

def nvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                    auth=auth,
                    verify=False)

    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    payload = {
        "interface": {
            "eth": {
                "ip": {
                    "address": {
                        "192.168.200.6/24": {}
                    }
                }
            }
        },
        "system": {
            "hostname": "switch01"
        }
    }
    apply_new_config("/",payload)
    time.sleep(DUMMY_SLEEP)
    nvue_get("/interface/eth0")
    nvue_get("/system")

```

### 3.3.13 API Examples

The following section provides practical API examples.

### 3.3.13.1 Configure the System

To set the system hostname, pre-login or post-login message, and time zone on the switch, send a targeted API request to `/nvue_v1/system`.

#### Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -d '{"system": {"hostname": "switch01", "timezone": "America/Los_Angeles", "message": {"pre-login": "Welcome to NVOS", "post-login": "You have successfully logged in to switch01"}}}' -k -X PATCH https://127.0.0.1:443/nvue_v1/?rev=4
```

#### Python Code

```
#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
    print("Headers:", r.headers)
    print("Body:", r.body)

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def create_nvue_changest():
    r = requests.post(url=nvue_end_point + "/revision",
                     auth=auth,
                     verify=False)
    print_request(r.request)
    print_response(r)
    response = r.json()
    changeset = response.popitem()[0]
    return changeset

def apply_nvue_changeset(changeset):
    # apply_payload = {"state": "apply"}
    apply_payload = {"state": "apply", "auto-prompt": {"ays": "ays_yes"}}
    url = nvue_end_point + "/revision/" + requests.utils.quote(changeset,
                                                                safe="")
    r = requests.patch(url=url,
                      auth=auth,
                      verify=False,
                      data=json.dumps(apply_payload),
                      headers=mime_header)
    print_request(r.request)
    print_response(r)

def is_config_applied(changeset) -> bool:
    # Check if the configuration was indeed applied
    global RETRIES
    global POLL_APPLIED
    retries = RETRIES
    while retries > 0:
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),
                        auth=auth,
                        verify=False)
        response = r.json()
        print(response)

        if response["state"] == "applied":
            return True
        retries -= 1
        time.sleep(POLL_APPLIED)
    return False

def apply_new_config(path,payload):
    # Create a new revision ID
    changeset = create_nvue_changest()
    print("Using NVUE Changeset: '{}'.format(changeset)

    # Delete existing configuration
    query_string = {"rev": changeset}
    r = requests.delete(url=nvue_end_point + path,
                       auth=auth,
```

```

        verify=False,
        params=query_string,
        headers=mime_header)
print_request(r.request)
print_response(r)

# Patch the new configuration
query_string = {"rev": changeset}
r = requests.patch(url=nvue_end_point + path,
                  auth=auth,
                  verify=False,
                  data=json.dumps(payload),
                  params=query_string,
                  headers=mime_header)
print_request(r.request)
print_response(r)

# Apply the changes to the new revision changeset
apply_nvvue_changeset(changeset)

# Check if the changeset was applied
is_config_applied(changeset)

def nvvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                    auth=auth,
                    verify=False)
    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    payload = {
        "system":
        {
            "hostname": "switch01",
            "timezone": "America/Los_Angeles",
            "message":
            {
                "pre-login": "Welcome to NVOS",
                "post-login": "You have successfully logged in to switch01"
            }
        }
    }
    apply_new_config("/", payload) # Root patch
    time.sleep(DUMMY_SLEEP)
    nvvue_get("/system")

```

## NVUE CLI

```

nvos@switch:~$ nv set system hostname switch01
nvos@switch:~$ nv set system timezone America/Los_Angeles
nvos@switch:~$ nv set system message pre-login "Welcome to NVOS"
nvos@switch:~$ nv set system message post-login "You have successfully logged into switch01"

```

## 3.3.13.2 Configure Services

To set up NTP, SYSLOG, and SNMP on the switch, send a targeted API request to `/nvvue_v1/system`.

### Curl Command

```

nvos@switch:~$ curl -u 'admin:admin' -d '{"system": { "ntp": { "server": {"4.nvos.pool.ntp.org": {}}}}' -k -X PATCH
https://127.0.0.1:443/nvvue_v1/?rev=5

```

### Python Code

```

#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvvue_end_point = "https://127.0.0.1:443/nvvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
    print("Headers:", r.headers)
    print("Body:", r.body)

```

```

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def create_nvue_changest():
    r = requests.post(url=nvue_end_point + "/revision",
                     auth=auth,
                     verify=False)
    print_request(r.request)
    print_response(r)
    response = r.json()
    changeset = response.popitem()[0]
    return changeset

def apply_nvue_changeset(changeset):
    # apply_payload = {"state": "apply"}
    apply_payload = {"state": "apply", "auto-prompt": {"ays": "ays_yes"}}
    url = nvue_end_point + "/revision/" + requests.utils.quote(changeset,
                                                                safe="")
    r = requests.patch(url=url,
                      auth=auth,
                      verify=False,
                      data=json.dumps(apply_payload),
                      headers=mime_header)
    print_request(r.request)
    print_response(r)

def is_config_applied(changeset) -> bool:
    # Check if the configuration was indeed applied
    global RETRIES
    global POLL_APPLIED
    retries = RETRIES
    while retries > 0:
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),
                        auth=auth,
                        verify=False)
        response = r.json()
        print(response)

        if response["state"] == "applied":
            return True
        retries -= 1
        time.sleep(POLL_APPLIED)

    return False

def apply_new_config(path,payload):
    # Create a new revision ID
    changeset = create_nvue_changest()
    print("Using NVUE Changeset: '{}'".format(changeset))

    # Delete existing configuration
    query_string = {"rev": changeset}
    r = requests.delete(url=nvue_end_point + path,
                       auth=auth,
                       verify=False,
                       params=query_string,
                       headers=mime_header)
    print_request(r.request)
    print_response(r)

    # Patch the new configuration
    query_string = {"rev": changeset}
    r = requests.patch(url=nvue_end_point + path,
                      auth=auth,
                      verify=False,
                      data=json.dumps(payload),
                      params=query_string,
                      headers=mime_header)
    print_request(r.request)
    print_response(r)

    # Apply the changes to the new revision changeset
    apply_nvue_changeset(changeset)

    # Check if the changeset was applied
    is_config_applied(changeset)

def nvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                    auth=auth,
                    verify=False)
    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    payload = {
        "system":
            {
                "ntp":
                    {
                        "server":
                            {
                                "4.nvos.pool.ntp.org": {}
                            }
                    }
            },
    }
    apply_new_config("/",payload) # Root patch
    time.sleep(DUMMY_SLEEP)
    nvue_get("/system/ntp")

```

## NVUE CLI

```
nvos@switch:~$ nv set system ntp server 4.nvos.pool.ntp.org
```

### 3.3.13.3 Configure Users

The following example creates a new user, then deletes the user.

#### Curl Command

This example creates a new user called `test1`.

```
nvos@switch:~$ curl -u 'admin:admin' -d '{"system": {"aaa": {"user": {"test1": {"hashed-  
password": "72b28582708d749c6c82f3b3f226041f1bd37090281641eaba8d44bd915d0042d609a92759d9f6fb96475cb0601cf428cd22613  
df8a53a09461e0b426cf0a35", "role": "monitor", "state": "enabled", "full-name": "Test User"}}}}}' -k -X PATCH https://  
127.0.0.1:443/nvue_v1/?rev=5
```

This example deletes the `test1` user.

```
nvos@switch:~$ curl -u 'admin:admin' -k -X DELETE https://127.0.0.1:443/nvue_v1/system/aaa/user/test1?rev=6
```

#### Python Code

```
#!/usr/bin/env python3  
  
import requests  
from requests.auth import HTTPBasicAuth  
import json  
import time  
  
auth = HTTPBasicAuth(username="admin", password="password")  
nvue_end_point = "https://127.0.0.1:443/nvue_v1"  
mime_header = {"Content-Type": "application/json"}  
  
DUMMY_SLEEP = 5 # In seconds  
POLL_APPLIED = 1 # in seconds  
RETRIES = 10  
  
def print_request(r: requests.Request):  
    print("====Request====")  
    print("URL:", r.url)  
    print("Headers:", r.headers)  
    print("Body:", r.body)  
  
def print_response(r: requests.Response):  
    print("====Response====")  
    print("Headers:", r.headers)  
    print("Body:", json.dumps(r.json(), indent=2))  
  
def create_nvue_changest():  
    r = requests.post(url=nvue_end_point + "/revision",  
                     auth=auth,  
                     verify=False)  
    print_request(r.request)  
    print_response(r)  
    response = r.json()  
    changeset = response.popitem()[0]  
    return changeset  
  
def apply_nvue_changeset(changeset):  
    # apply_payload = {"state": "apply"}  
    apply_payload = {"state": "apply", "auto-prompt": {"ays": "ays_yes"}}  
    url = nvue_end_point + "/revision/" + requests.utils.quote(changeset,  
                                                                safe="")  
    r = requests.patch(url=url,  
                      auth=auth,  
                      verify=False,  
                      data=json.dumps(apply_payload),  
                      headers=mime_header)  
    print_request(r.request)  
    print_response(r)  
  
def is_config_applied(changeset) -> bool:  
    # Check if the configuration was indeed applied  
    global RETRIES  
    global POLL_APPLIED  
    retries = RETRIES  
    while retries > 0:  
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),  
                        auth=auth,  
                        verify=False)
```

```

    response = r.json()
    print(response)

    if response["state"] == "applied":
        return True
    retries -= 1
    time.sleep(POLL_APPLIED)

return False

def apply_new_config(path,payload):
    # Create a new revision ID
    changeset = create_nvue_changest()
    print("Using NVUE Changeset: '{}'".format(changeset))

    # Delete existing configuration
    query_string = {"rev": changeset}
    r = requests.delete(url=nvue_end_point + path,
                        auth=auth,
                        verify=False,
                        params=query_string,
                        headers=mime_header)
    print_request(r.request)
    print_response(r)

    # Patch the new configuration
    query_string = {"rev": changeset}
    r = requests.patch(url=nvue_end_point + path,
                       auth=auth,
                       verify=False,
                       data=json.dumps(payload),
                       params=query_string,
                       headers=mime_header)
    print_request(r.request)
    print_response(r)

    # Apply the changes to the new revision changeset
    apply_nvue_changeset(changeset)

    # Check if the changeset was applied
    is_config_applied(changeset)

def delete_config(path):
    # Create an NVUE changeset
    changeset = create_nvue_changest()
    print("Using NVUE Changeset: '{}'".format(changeset))

    # Equivalent to JSON `null`
    payload = None

    # Stage the change
    query_string = {"rev": changeset}
    r = requests.delete(url=nvue_end_point + path,
                        auth=auth,
                        verify=False,
                        data=json.dumps(payload),
                        params=query_string,
                        headers=mime_header)
    print_request(r.request)
    print_response(r)

    # Apply the staged changeset
    apply_nvue_changeset(changeset)

    # Check if the changeset was applied
    is_config_applied(changeset)

def nvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                     auth=auth,
                     verify=False)
    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    # Need to create a hashed password - The supported password
    # Here in this example, we use SHA-512
    import crypt
    hashed_password = crypt.crypt("hello$world#2023", salt=crypt.METHOD_SHA512)
    payload = {
        "system": {
            "aaa": {
                "user": {
                    "test1": {
                        "hashed-password": hashed_password,
                        "role": "monitor",
                        "state": "enabled",
                        "full-name": "Test User",
                    }
                }
            }
        }
    }
    apply_new_config("/",payload) # Root patch
    time.sleep(DUMMY_SLEEP)
    nvue_get("/system/user/aaa")

    """Delete an existing user account using the AAA API."""
    delete_config("/system/aaa/user/test1")
    time.sleep(DUMMY_SLEEP)
    nvue_get("/system/user/aaa")

```

## NVUE CLI

This example creates a new user `test1`.

```
nvos@switch:~$ nv set system aaa user test1
nvos@switch:~$ nv set system aaa user test1 full-name "Test User"
nvos@switch:~$ nv set system aaa user test1 password "abcd@test"
nvos@switch:~$ nv set system aaa user test1 role monitor
nvos@switch:~$ nv set system aaa user test1 state enabled
```

This example deletes the user `test1`.

```
nvos@switch:~$ nv unset system aaa user test1
```

## 3.3.13.4 Configure an Interface

The following example configures an interface.

### Curl Command

```
nvos@switch:~$ curl -u 'admin:admin' -k -d '{"eth1": {"link":{"state":{"up": {}}}}}' -H 'Content-Type: application/json' -k -X PATCH https://127.0.0.1:443/nvue_v1/interface?rev=21
```

### Python Code

```
#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
    print("Headers:", r.headers)
    print("Body:", r.body)

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def create_nvue_changset():
    r = requests.post(url=nvue_end_point + "/revision",
                     auth=auth,
                     verify=False)
    print_request(r.request)
    print_response(r)
    response = r.json()
    changeset = response.popitem()[0]
    return changeset

def apply_nvue_changeset(changeset):
    # apply_payload = {"state": "apply"}
    apply_payload = {"state": "apply", "auto-prompt": {"ays": "ays_yes"}}
    url = nvue_end_point + "/revision/" + requests.utils.quote(changeset,
                                                                safe="")
    r = requests.patch(url=url,
                      auth=auth,
                      verify=False,
                      data=json.dumps(apply_payload),
                      headers=mime_header)
    print_request(r.request)
    print_response(r)

def is_config_applied(changeset) -> bool:
    # Check if the configuration was indeed applied
    global RETRIES
    global POLL_APPLIED
    retries = RETRIES
    while retries > 0:
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),
```

```

        auth=auth,
        verify=False)
    response = r.json()
    print(response)

    if response["state"] == "applied":
        return True
    retries -= 1
    time.sleep(POLL_APPLIED)

    return False

def apply_new_config(path,payload):
    # Create a new revision ID
    changeset = create_nvue_changeset()
    print("Using NVUE Changeset: {}".format(changeset))

    # Delete existing configuration
    query_string = {"rev": changeset}
    r = requests.delete(url=nvue_end_point + path,
                        auth=auth,
                        verify=False,
                        params=query_string,
                        headers=mime_header)
    print_request(r.request)
    print_response(r)

    # Patch the new configuration
    query_string = {"rev": changeset}
    r = requests.patch(url=nvue_end_point + path,
                       auth=auth,
                       verify=False,
                       data=json.dumps(payload),
                       params=query_string,
                       headers=mime_header)
    print_request(r.request)
    print_response(r)

    # Apply the changes to the new revision changeset
    apply_nvue_changeset(changeset)

    # Check if the changeset was applied
    is_config_applied(changeset)

def nvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                    auth=auth,
                    verify=False)
    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    payload = {
        "eth1": {
            "type": "eth",
            "link": {
                "state": "up"
            }
        }
    }
    apply_new_config("/interface",payload)
    time.sleep(DUMMY_SLEEP)
    nvue_get("/interface/eth1")

```

## NVUE CLI

```
nvos@switch:~$ nv set interface eth1
```

### 3.3.13.5 Action Operations

The NVUE action operations are ephemeral operations that do not modify the state of the configuration; they reset counters for interfaces, reboot the system etc...

#### Curl Command

To clear counters on sw1p1:

```

nvos@switch:~$ curl -u 'admin:admin' -H 'Content-Type:application/json' -d '{"clear": {"state": "start",
"parameters": {}}}' -k -X POST https://127.0.0.1:443/nvue_v1/interface/sw1p1/link/counters
1
nvos@switch:~$ curl -u 'admin:admin' -X GET https://127.0.0.1:443/nvue_v1/action/1 -k

```

```
{"detail": "swlpl counters cleared.", "http_status": 200, "issue": [], "state": "action_success", "status": "swlpl counters cleared.", "timeout": 60, "type": ""}
```

## Python Code

```
#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}

DUMMY_SLEEP = 5 # In seconds
POLL_APPLIED = 1 # in seconds
RETRIES = 10

def print_request(r: requests.Request):
    print("====Request====")
    print("URL:", r.url)
    print("Headers:", r.headers)
    print("Body:", r.body)

def print_response(r: requests.Response):
    print("====Response====")
    print("Headers:", r.headers)
    print("Body:", json.dumps(r.json(), indent=2))

def nvue_action():
    r = requests.post(url=nvue_end_point + path,
                     auth=auth,
                     verify=False,
                     data=json.dumps(apply_payload),
                     headers=mime_header)

    print_request(r.request)
    print_response(r)
    return response

def nvue_get(path):
    r = requests.get(url=nvue_end_point + path,
                    auth=auth,
                    verify=False)

    print_request(r.request)
    print_response(r)

if __name__ == "__main__":
    payload = {
        "@clear":
        {
            "state": "start",
            "parameters": {}
        }
    }
    action_id=nvue_action("/interface/swlpl/link/counters",payload)
    time.sleep(DUMMY_SLEEP)
    nvue_get(f"/action/{action_id}")
```

## NVUE CLI

```
nvos@switch:~$ nv action clear interface swlpl link counters
```

## 3.3.14 Example Python Script

In the following python example, the `full_config_example()` method sets the system pre-login message, and changes a few other configuration settings in a single bulk operation. The API end-point goes to the root node `/nvue_v1`.

```
#!/usr/bin/env python3

import requests
from requests.auth import HTTPBasicAuth
import json
import time

auth = HTTPBasicAuth(username="admin", password="password")
nvue_end_point = "https://127.0.0.1:443/nvue_v1"
mime_header = {"Content-Type": "application/json"}
```



```

# different switch configurations
payload = {
    "interface":{
        "eth0":{
            "description": "management port"
        }
    },
    "system":{
        "message":{
            "pre-login": pre_login_message,
            "post-login": post_login_message
        },
        "timezone": "Europe/Paris",
    },
}

# Stage the change
query_string = {"rev": changeset}
r = requests.patch(url=nvue_end_point + "/", # Root patch
                  auth=auth,
                  verify=False,
                  data=json.dumps(payload),
                  params=query_string,
                  headers=mime_header)

print_request(r.request)
print_response(r)

# Apply the staged changeset
apply_nvue_changeset(changeset)

# Check if the changeset was applied
is_config_applied(changeset)

def message_get():
    # Get the system pre-login/post-login
    # message that was configured.
    r = requests.get(url=nvue_end_point + "/system/message",
                    auth=auth,
                    verify=False)
    print_request(r.request)
    print_response(r)

def is_config_applied(changeset) -> bool:
    # Check if the configuration was indeed applied
    global RETRIES
    global POLL_APPLIED
    retries = RETRIES
    while retries > 0:
        r = requests.get(url=nvue_end_point + "/revision/" + requests.utils.quote(changeset, safe=""),
                        auth=auth,
                        verify=False)
        response = r.json()
        print(response)

        if response["state"] == "applied":
            return True
            retries -= 1
            time.sleep(POLL_APPLIED)

    return False

if __name__ == "__main__":
    sanity()
    time.sleep(DUMMY_SLEEP)
    full_config_example()
    time.sleep(DUMMY_SLEEP)
    message_get()

```

### 3.3.15 Try the API

To try out the NVUE REST API, use the [NVUE API Lab](#) available on NVIDIA Air. The lab provides a basic example to help you get started. You can also try out the other examples in this document.

### 3.3.16 Resources

For information about using the NVUE REST API, refer to the [NVUE API Swagger documentation](#). The full object model download is available [here](#).

### 3.3.17 Considerations

- Unlike the NVUE CLI, the NVUE API does not support configuring a plain text password for a user account; you must configure a hashed password for a user account with the NVUE API.
- If you need to make multiple updates on the switch, NVIDIA recommends you use a root patch, which can make configuration changes with fewer round trips to the switch. Running

many specific NVUE PATCH APIs to set or unset objects requires many round trips to the switch to set up the HTTP connection, transfer payload and responses, manage network utilization, and so on.

### 3.3.18 Related Information

- [NGINX documentation](#)
- [Ubuntu Certificates and Security documentation](#)
- [Python requests module](#)

### 3.3.19 Save the Applied Configurations

1. Send a PATCH request to save the currently applied configuration.

```
curl -u '<username:password>' --insecure --request PATCH 'https://<IP>/nvue_v1/revision/applied' -H 'Content-Type: application/json' -d '{"state": "save"}'
```

### 3.3.20 Send an Action

1. Run the nv action disconnect

```
$ curl -u '<username:password>' --insecure https://<IP>/nvue_v1/system/aaa/user -H 'Content-Type: application/json' -d '{
"@disconnect": {"state": "start", "parameters": {"user": "monitor"}
}'
3
```

2. Get the action ID from response; "3"

3. Check status

```
$ curl -u '<username:password>' --request GET 'https://<IP>/nvue_v1/action/3'
{"detail": "all sessions have been terminated", "http_status": 200, "issue": [], "state": "action_success", "status": "all sessions have been terminated", "timeout": 60, "type": ""}
```

---

## 4 System Management

The following pages provide information on configuring general management features on the switch system.

- [Access Control Lists](#)
- [Attestation](#)
- [Authentication Authorization and Accounting](#)
- [Certificates Management](#)
- [Control and Power](#)
- [DNS Server](#)
- [Documentation](#)
- [Hostname](#)
- [Management Interfaces](#)
- [Resource Management](#)
- [Security](#)
- [System API](#)
- [Time Synchronization](#)
- [User Interfaces](#)
- [Zero-Touch Provisioning](#)

### 4.1 Access Control Lists

Access Control Lists (ACLs) are rules on the switch that act as filters to manage traffic.

This section discusses the default Firewall Rules installed on the switch to protect the switch control plane and CPU from DOS and other potentially malicious network attacks and how to Configure Access Control Lists to manage traffic:

- [Firewall Rules](#)
- [Access Control List Configuration](#)
- [Access Control List Commands](#)

#### 4.1.1 Firewall Rules

The NVOS default firewall rules protect the switch control plane and CPU from DOS and other potentially malicious network attacks.

The default set of firewall rules consists of IP and transport level rules. See [Access Control List Configuration](#) for custom ACL rules configurations.



Please note that users cannot bind ACL rules to the Loopback interface (lo).

##### 4.1.1.1 DoS Rules

DoS rules protect the switch control plane and CPU from DOS attacks. NVOS provides firewall DoS rules to do the following:

- Allow only internal traffic to the loopback interfaces.
- Accept already established connections and outbound traffic.
- Drop packets if the first TCP segment is not SYN.
- Drop fragmented IP packets.
- Drop Christmas tree packets; packets with all TCP flags set.
- Drop NULL packets.
- Drop invalid packets.
- Drop strange MSS values.
- Provide brute-force protection.
- Drop packets with routing Header Type 0.
- Drop packets with a hop limit greater than 1.
- Limit excessive TCP reset packets.
- Protect against SYN flood.
- Rate limit new TCP connections for each IP address.
- Log all remaining packets, then drop them.

#### 4.1.1.2 Whitelist Rules

Whitelist rules specify the services or application ports enabled on the switch. NVOS provides firewall whitelist rules to enable TCP ports and UDP ports.

The following table lists the ports that NVOS enables by default.

Protocol	Port	Application
TCP	22	SSH
UDP	68	DHCP Client
UDP	67	DHCP Server
UDP	123	NTP
UDP	161	SNMP
TCP	389	LDAP
TCP	636	LDAP TLS
UDP	546	DHCPv6 Client
UDP	547	DHCPv6 Server
UDP	4500	IPSec-NAT
UDP	500	IKE
UDP	1812,1813,1645,1656	RADIUS
TCP	49	TACACS
UDP/TCP	53	DNS
UDP	5353	mDNS
UDP	514	remote syslog
TCP	443	HTTPS
TCP	9339	gNMI
ICMP	NA	Ping

### 4.1.1.3 Unset the Default Firewall Rules

To unset the default firewall rules to accept packets from all addresses and protocols:

```
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_CP_DEFAULT
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_CP_DEFAULT_IPV6
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_DEFAULT
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_DEFAULT_IPV6
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6
nvos@switch:~$ nv unset interface lo acl ACL_LOOPBACK_INBOUND_CP_DEFAULT
nvos@switch:~$ nv unset interface lo acl ACL_LOOPBACK_INBOUND_CP_DEFAULT_IPV6
nvos@switch:~$ nv config apply
```

To set the firewall rules back to the default setting:

```
nvos@switch:~$ nv unset interface
nvos@switch:~$ nv config apply
```

To set the firewall rules back to the default setting on specific interface:

```
nvos@switch:~$ nv unset interface eth0 acl
nvos@switch:~$ nv config apply
```

### 4.1.1.4 Add Firewall Rules

You cannot modify the `ACL_MGMT_INBOUND_CP_DEFAULT`, `ACL_MGMT_INBOUND_CP_DEFAULT_IPV6`, `ACL_MGMT_INBOUND_DEFAULT`, `ACL_MGMT_INBOUND_DEFAULT_IPV6`, `ACL_MGMT_OUTBOUND_CP_DEFAULT`, `ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6`, `ACL_LOOPBACK_INBOUND_CP_DEFAULT` and `ACL_LOOPBACK_INBOUND_CP_DEFAULT_IPV6` rules. However, you can append or insert additional rules.

If you use non-default ports for an application, NVIDIA recommends that you add a whitelist rule for the non-default port. For example, if you use ports 3020 and 3022 for radius server accounting and authentication instead of 1812 and 1813, you can add the following whitelist rules:

```
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 match ip udp source-port 3020
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 match ip connection-state new
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 match ip connection-state established
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 action permit
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 match ip udp source-port 3022
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 match ip connection-state new
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 match ip connection-state established
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 action permit
nvos@switch:~$ nv config apply
```

### 4.1.1.5 Show Firewall Rules

To show the default rules, run the `nv show acl <default-acl-id>` command, where `<default-acl-id>` is one of `ACL_MGMT_INBOUND_CP_DEFAULT`, `ACL_MGMT_INBOUND_CP_DEFAULT_IPV6`, `ACL_MGMT_INBOUND_DEFAULT`, `ACL_MGMT_INBOUND_DEFAULT_IPV6`, `ACL_MGMT_OUTBOUND_CP_DEFAULT`, `ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6`, `ACL_LOOPBACK_INBOUND_CP_DEFAULT` and `ACL_LOOPBACK_INBOUND_CP_DEFAULT_IPV6`:

```
nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT
-----
operational applied
type ipv4      ipv4
```

```

rule
=====
Number Summary
-----
10  action: deny
    match.ip.dest-ip: 127.0.0.0/8
20  action: permit
30  action: deny
    match.ip.protocol: tcp
40  action: deny
    match.ip.protocol: tcp
50  action: deny
    match.ip.protocol: tcp
60  action: deny
    match.ip.protocol: tcp
70  action: deny
80  action: deny
    match.ip.protocol: tcp
90  action: deny
    match.ip.protocol: tcp
100 action: deny
110 match.ip.protocol: tcp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: TCP
    match.ip.tcp.dest-port: 22
120 action: deny
    match.ip.protocol: tcp
    match.ip.recent-list.action: update
    match.ip.recent-list.hit-count: 100
    match.ip.recent-list.name: TCP
    match.ip.recent-list.update-interval: 60
    match.ip.tcp.dest-port: 22
130 match.ip.protocol: udp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: UDP
    match.ip.udp.dest-port: 161
140 action: deny
    match.ip.protocol: udp
    match.ip.recent-list.action: update
    match.ip.recent-list.hit-count: 100
    match.ip.recent-list.name: UDP
    match.ip.recent-list.update-interval: 60
    match.ip.udp.dest-port: 161
150 match.ip.protocol: tcp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: TCP
    match.ip.tcp.dest-port: 443
160 action: deny
    match.ip.protocol: tcp
    match.ip.recent-list.action: update
    match.ip.recent-list.hit-count: 150
    match.ip.recent-list.name: TCP
    match.ip.recent-list.update-interval: 60
    match.ip.tcp.dest-port: 443
170 match.ip.protocol: tcp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: TCP
    match.ip.tcp.dest-port: 9339
180 action: deny
    match.ip.protocol: tcp
    match.ip.recent-list.action: update
    match.ip.recent-list.hit-count: 100
    match.ip.recent-list.name: TCP
    match.ip.recent-list.update-interval: 60
    match.ip.tcp.dest-port: 9339
190 match.ip.protocol: tcp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: TCP
    match.ip.tcp.dest-port: 636
200 action: deny
    match.ip.protocol: tcp
    match.ip.recent-list.action: update
    match.ip.recent-list.hit-count: 100
    match.ip.recent-list.name: TCP
    match.ip.recent-list.update-interval: 60
    match.ip.tcp.dest-port: 636
210 match.ip.protocol: tcp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: TCP
    match.ip.tcp.dest-port: 389
220 action: deny
    match.ip.protocol: tcp
    match.ip.recent-list.action: update
    match.ip.recent-list.hit-count: 100
    match.ip.recent-list.name: TCP
    match.ip.recent-list.update-interval: 60
    match.ip.tcp.dest-port: 389
230 match.ip.protocol: tcp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: TCP
    match.ip.tcp.dest-port: 49
240 action: deny
    match.ip.protocol: tcp
    match.ip.recent-list.action: update
    match.ip.recent-list.hit-count: 100
    match.ip.recent-list.name: TCP
    match.ip.recent-list.update-interval: 60
    match.ip.tcp.dest-port: 49
250 match.ip.protocol: udp
    match.ip.recent-list.action: set
    match.ip.recent-list.name: UDP
    match.ip.udp.dest-port: 123
260 action: deny

```

```

match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    123
270 match.ip.protocol:          tcp
match.ip.recent-list.action: set
match.ip.recent-list.name:  TCP
match.ip.tcp.dest-port:    53
280 action:                    deny
match.ip.protocol:          tcp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  TCP
match.ip.recent-list.update-interval: 60
match.ip.tcp.dest-port:    53
290 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    53
300 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    53
310 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    514
320 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    514
330 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    5353
340 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    5353
350 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    68
360 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    68
370 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    67
380 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    67
390 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    4500
400 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    4500
410 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    500
420 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    500
430 match.ip.protocol:          udp
match.ip.recent-list.action: set
match.ip.recent-list.name:  UDP
match.ip.udp.dest-port:    1812
440 action:                    deny
match.ip.protocol:          udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name:  UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:    1812
450 match.ip.protocol:          udp

```

```

match.ip.recent-list.action:      set
match.ip.recent-list.name:       UDP
match.ip.udp.dest-port:         1813
460 action:                         deny
match.ip.protocol:              udp
match.ip.recent-list.action:     update
match.ip.recent-list.hit-count:  100
match.ip.recent-list.name:       UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:         1813
470 match.ip.protocol:              udp
match.ip.recent-list.action:     set
match.ip.recent-list.name:       UDP
match.ip.udp.dest-port:         1645
480 action:                         deny
match.ip.protocol:              udp
match.ip.recent-list.action:     update
match.ip.recent-list.hit-count:  100
match.ip.recent-list.name:       UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:         1645
490 match.ip.protocol:              udp
match.ip.recent-list.action:     set
match.ip.recent-list.name:       UDP
match.ip.udp.dest-port:         1646
500 action:                         deny
match.ip.protocol:              udp
match.ip.recent-list.action:     update
match.ip.recent-list.hit-count:  100
match.ip.recent-list.name:       UDP
match.ip.recent-list.update-interval: 60
match.ip.udp.dest-port:         1646
510 action:                         deny
match.ip.hashlimit.burst:        2
match.ip.hashlimit.expire:       30000
match.ip.hashlimit.mode:         src-ip
match.ip.hashlimit.name:         TCPRST
match.ip.hashlimit.rate-above:   5/min
match.ip.hashlimit.source-mask:  32
match.ip.protocol:              tcp
520 action:                         deny
match.ip.hashlimit.burst:        30
match.ip.hashlimit.expire:       30000
match.ip.hashlimit.mode:         src-ip
match.ip.hashlimit.name:         TCPGENRAL
match.ip.hashlimit.rate-above:   50/second
match.ip.hashlimit.source-mask:  32
match.ip.protocol:              tcp
530 action:                         deny
match.ip.hashlimit.burst:        30
match.ip.hashlimit.expire:       3000
match.ip.hashlimit.mode:         src-ip
match.ip.hashlimit.name:         TCPGENRAL
match.ip.hashlimit.rate-above:   50/second
match.ip.hashlimit.source-mask:  32
match.ip.protocol:              tcp
560 action:                         permit
match.ip.protocol:              udp
match.ip.udp.dest-port:         161
remark:                          Whitelist-snmp
570 action:                         permit
match.ip.protocol:              tcp
match.ip.tcp.dest-port:         443
remark:                          Whitelist-https
580 action:                         permit
match.ip.protocol:              tcp
match.ip.tcp.dest-port:         22
remark:                          Whitelist-ssh
590 action:                         permit
match.ip.protocol:              tcp
match.ip.tcp.dest-port:         9339
remark:                          Whitelist-gnmi
600 action:                         permit
match.ip.protocol:              tcp
match.ip.tcp.dest-port:         636
remark:                          Whitelist-ldap-tls
610 action:                         permit
match.ip.protocol:              udp
match.ip.udp.dest-port:         514
remark:                          Whitelist-rsyslog
620 action:                         permit
match.ip.protocol:              tcp
match.ip.tcp.dest-port:         389
remark:                          Whitelist-ldap
630 action:                         permit
match.ip.protocol:              tcp
match.ip.tcp.dest-port:         49
remark:                          Whitelist-tacacs
640 action:                         permit
match.ip.protocol:              udp
match.ip.udp.dest-port:         123
remark:                          Whitelist-ntp
650 action:                         permit
match.ip.protocol:              udp
match.ip.udp.dest-port:         53
remark:                          Whitelist-dns
660 action:                         permit
match.ip.protocol:              tcp
match.ip.tcp.dest-port:         53
remark:                          Whitelist-dns
670 action:                         permit
match.ip.protocol:              udp
match.ip.udp.dest-port:         5353
remark:                          Whitelist-mDNS
680 action:                         permit
match.ip.protocol:              udp

```

```

match.ip.udp.dest-port:      68
remark:                      Whitelist-dhcp
690 action:                    permit
match.ip.protocol:          udp
match.ip.udp.dest-port:     67
remark:                      Whitelist-dhcp
700 action:                    permit
match.ip.protocol:          udp
match.ip.udp.dest-port:     4500
remark:                      Whitelist-IPSec-NAT
710 action:                    permit
match.ip.protocol:          udp
match.ip.udp.dest-port:     500
remark:                      Whitelist-IKE
720 action:                    permit
match.ip.protocol:          udp
match.ip.udp.dest-port:     1812
remark:                      Whitelist-radius
730 action:                    permit
match.ip.protocol:          udp
match.ip.udp.dest-port:     1813
remark:                      Whitelist-radius
740 action:                    permit
match.ip.protocol:          udp
match.ip.udp.dest-port:     1645
remark:                      Whitelist-radius
750 action:                    permit
match.ip.protocol:          udp
match.ip.udp.dest-port:     1646
remark:                      Whitelist-radius
760 action:                    permit
match.ip.protocol:          icmp
remark:                      Whitelist-icmp
770 action:                    log
match.ip.hashlimit.burst:    5
match.ip.hashlimit.expire:   4294967295
match.ip.hashlimit.mode:     src-ip
match.ip.hashlimit.name:     LOGGING
match.ip.hashlimit.rate-above: 1/min
match.ip.hashlimit.source-mask: 32
780 action:                    deny

```

Run the `nv show acl ACL_MGMT_INBOUND_CP_DEFAULT --rev=applied -o json` command to show additional information, such as the connection state, hit count and update interval:

```

nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT --rev=applied -o json
...
"630": {
  "action": {
    "permit": {}
  },
  "match": {
    "ip": {
      "connection-state": {
        "established": {},
        "new": {}
      },
      "protocol": "tcp",
      "tcp": {
        "dest-port": {
          "49": {}
        }
      }
    }
  },
  "remark": "Whitelist-tacacs"
},
...
"500": {
  "action": {
    "deny": {}
  },
  "match": {
    "ip": {
      "connection-state": {
        "new": {}
      },
      "protocol": "udp",
      "recent-list": {
        "action": "update",
        "hit-count": 100,
        "name": "UDP",
        "update-interval": 60
      },
      "udp": {
        "dest-port": {
          "1646": {}
        }
      }
    }
  }
},
...

```

To show information about a specific rule, run the `nv show acl <default-acl-id> rule <rule>` command:

```
nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 500
-----
operational  applied
-----
match
  ip
  protocol      udp      udp
  udp
  [dest-port]   1646    1646
  recent-list
  name          UDP      UDP
  update-interval 60      60
  hit-count     100     100
  action        update  update
  deny          deny   deny

Run the nv show acl <default-acl-id> rule <rule> --rev=applied -o json command to see additional information, such
as the connection state:

nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 500 --rev=applied -o json {
  "action": {
    "deny": {}
  },
  "match": {
    "ip": {
      "connection-state": {
        "new": {}
      },
      "protocol": "udp",
      "recent-list": {
        "action": "update",
        "hit-count": 100,
        "name": "UDP",
        "update-interval": 60
      },
      "udp": {
        "dest-port": {
          "1646": {}
        }
      }
    }
  }
}
```

### 4.1.1.6 Log Messages

Default firewall rules include a log rule for packets that arrive in the control plane and do not match user defined or default firewall rules. The switch generates a log message in `/var/log/firewall_packet_capture.log` for packets that match the log rule.

The NVOS Linux default firewall rules protect the switch control plane and CPU from DOS and other potentially malicious network attacks.

The default set of firewall rules consists of IP and transport level rules. See [Access Control List Configuration](#) for custom ACL rules configurations.

### 4.1.1.7 DoS Rules

DoS rules protect the switch control plane and CPU from DOS attacks. NVOS provides firewall DoS rules to do the following:

- Allow only internal traffic to the loopback interfaces.
- Accept already established connections and outbound traffic.
- Drop packets if the first TCP segment is not SYN.
- Drop fragmented IP packets.
- Drop Christmas tree packets; packets with all TCP flags set.
- Drop NULL packets.
- Drop invalid packets.
- Drop strange MSS values.

- Provide brute-force protection.
- Drop packets with routing Header Type 0.
- Drop packets with a hop limit greater than 1.
- Limit excessive TCP reset packets.
- Protect against SYN flood.
- Rate limit new TCP connections for each IP address.
- Log all remaining packets, then drop them.

#### 4.1.1.8 Whitelist Rules

Whitelist rules specify the services or application ports enabled on the switch. NVOS provides firewall whitelist rules to enable TCP ports and UDP ports.

The following table lists the ports that NVOS enables by default.

Protocol	Port	Application
TCP	22	SSH
UDP	68	DHCP Client
UDP	67	DHCP Server
UDP	123	NTP
UDP	161	SNMP
TCP	389	LDAP
TCP	636	LDAP TLS
UDP	546	DHCPv6 Client
UDP	547	DHCPv6 Server
UDP	4500	IPSec-NAT
UDP	500	IKE
UDP	1812,1813,1645,1656	RADIUS
TCP	49	TACACS
UDP/TCP	53	DNS
UDP	5353	mDNS
UDP	514	remote syslog
TCP	443	HTTPS
TCP	9339	gNMI
ICMP	NA	Ping

#### 4.1.1.9 Unset the Default Firewall Rules

To unset the default firewall rules to accept packets from all addresses and protocols:

```

nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_CP_DEFAULT
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_CP_DEFAULT_IPV6
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_DEFAULT
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_INBOUND_DEFAULT_IPV6
nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT

```

```

nvos@switch:~$ nv unset interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6
nvos@switch:~$ nv unset interface lo acl ACL_LOOPBACK_INBOUND_CP_DEFAULT
nvos@switch:~$ nv unset interface lo acl ACL_LOOPBACK_INBOUND_CP_DEFAULT_IPV6
nvos@switch:~$ nv config apply

```

To set the firewall rules back to the default setting:

```

nvos@switch:~$ nv unset interface
nvos@switch:~$ nv config apply

```

To set the firewall rules back to the default setting on specific interface:

```

nvos@switch:~$ nv unset interface eth0 acl
nvos@switch:~$ nv config apply

```

#### 4.1.1.10 Add Firewall Rules

You cannot modify the `ACL_MGMT_INBOUND_CP_DEFAULT`, `ACL_MGMT_INBOUND_CP_DEFAULT_IPV6`, `ACL_MGMT_INBOUND_DEFAULT`, `ACL_MGMT_INBOUND_DEFAULT_IPV6`, `ACL_MGMT_OUTBOUND_CP_DEFAULT`, `ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6`, `ACL_LOOPBACK_INBOUND_CP_DEFAULT` and `ACL_LOOPBACK_INBOUND_CP_DEFAULT_IPV6` rules. However, you can append or insert additional rules.

If you use non-default ports for an application, NVIDIA recommends that you add a whitelist rule for the non-default port. For example, if you use ports 3020 and 3022 for radius server accounting and authentication instead of 1812 and 1813, you can add the following whitelist rules:

```

nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 match ip udp source-port 3020
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 match ip connection-state new
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 match ip connection-state established
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 765 action permit
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 match ip udp source-port 3022
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 match ip connection-state new
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 match ip connection-state established
nvos@switch:~$ nv set acl ACL_MGMT_INBOUND_CP_DEFAULT rule 766 action permit
nvos@switch:~$ nv config apply

```

#### 4.1.1.11 Show Firewall Rules

To show the default rules, run the `nv show acl <default-acl-id>` command, where `<default-acl-id>` is one of `ACL_MGMT_INBOUND_CP_DEFAULT`, `ACL_MGMT_INBOUND_CP_DEFAULT_IPV6`, `ACL_MGMT_INBOUND_DEFAULT`, `ACL_MGMT_INBOUND_DEFAULT_IPV6`, `ACL_MGMT_OUTBOUND_CP_DEFAULT`, `ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6`, `ACL_LOOPBACK_INBOUND_CP_DEFAULT` and `ACL_LOOPBACK_INBOUND_CP_DEFAULT_IPV6`:

```

nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT
operational applied
----
type  ipv4          ipv4

rule
=====
Number  Summary
-----
10      action:          deny
       match.ip.dest-ip: 127.0.0.0/8
20      action:          permit
30      action:          deny
       match.ip.protocol: tcp
40      action:          deny
       match.ip.protocol: tcp
50      action:          deny
       match.ip.protocol: tcp
60      action:          deny

```

```

70 match.ip.protocol: tcp
80 action: deny
90 match.ip.protocol: tcp
90 action: deny
100 match.ip.protocol: tcp
100 action: deny
110 match.ip.protocol: tcp
110 match.ip.recent-list.action: set
110 match.ip.recent-list.name: TCP
110 match.ip.tcp.dest-port: 22
120 action: deny
120 match.ip.protocol: tcp
120 match.ip.recent-list.action: update
120 match.ip.recent-list.hit-count: 100
120 match.ip.recent-list.name: TCP
120 match.ip.recent-list.update-interval: 60
130 match.ip.tcp.dest-port: 22
130 match.ip.protocol: udp
130 match.ip.recent-list.action: set
130 match.ip.recent-list.name: UDP
140 match.ip.udp.dest-port: 161
140 action: deny
140 match.ip.protocol: udp
140 match.ip.recent-list.action: update
140 match.ip.recent-list.hit-count: 100
140 match.ip.recent-list.name: UDP
140 match.ip.recent-list.update-interval: 60
150 match.ip.udp.dest-port: 161
150 match.ip.protocol: tcp
150 match.ip.recent-list.action: set
150 match.ip.recent-list.name: TCP
160 match.ip.tcp.dest-port: 443
160 action: deny
160 match.ip.protocol: tcp
160 match.ip.recent-list.action: update
160 match.ip.recent-list.hit-count: 150
160 match.ip.recent-list.name: TCP
160 match.ip.recent-list.update-interval: 60
170 match.ip.tcp.dest-port: 443
170 match.ip.protocol: tcp
170 match.ip.recent-list.action: set
170 match.ip.recent-list.name: TCP
180 match.ip.tcp.dest-port: 9339
180 action: deny
180 match.ip.protocol: tcp
180 match.ip.recent-list.action: update
180 match.ip.recent-list.hit-count: 100
180 match.ip.recent-list.name: TCP
180 match.ip.recent-list.update-interval: 60
190 match.ip.tcp.dest-port: 9339
190 match.ip.protocol: tcp
190 match.ip.recent-list.action: set
190 match.ip.recent-list.name: TCP
200 match.ip.tcp.dest-port: 636
200 action: deny
200 match.ip.protocol: tcp
200 match.ip.recent-list.action: update
200 match.ip.recent-list.hit-count: 100
200 match.ip.recent-list.name: TCP
200 match.ip.recent-list.update-interval: 60
210 match.ip.tcp.dest-port: 636
210 match.ip.protocol: tcp
210 match.ip.recent-list.action: set
210 match.ip.recent-list.name: TCP
220 match.ip.tcp.dest-port: 389
220 action: deny
220 match.ip.protocol: tcp
220 match.ip.recent-list.action: update
220 match.ip.recent-list.hit-count: 100
220 match.ip.recent-list.name: TCP
220 match.ip.recent-list.update-interval: 60
230 match.ip.tcp.dest-port: 389
230 match.ip.protocol: tcp
230 match.ip.recent-list.action: set
230 match.ip.recent-list.name: TCP
240 match.ip.tcp.dest-port: 49
240 action: deny
240 match.ip.protocol: tcp
240 match.ip.recent-list.action: update
240 match.ip.recent-list.hit-count: 100
240 match.ip.recent-list.name: TCP
240 match.ip.recent-list.update-interval: 60
250 match.ip.tcp.dest-port: 49
250 match.ip.protocol: udp
250 match.ip.recent-list.action: set
250 match.ip.recent-list.name: UDP
260 match.ip.udp.dest-port: 123
260 action: deny
260 match.ip.protocol: udp
260 match.ip.recent-list.action: update
260 match.ip.recent-list.hit-count: 100
260 match.ip.recent-list.name: UDP
260 match.ip.recent-list.update-interval: 60
270 match.ip.udp.dest-port: 123
270 match.ip.protocol: tcp
270 match.ip.recent-list.action: set
270 match.ip.recent-list.name: TCP
280 match.ip.tcp.dest-port: 53
280 action: deny
280 match.ip.protocol: tcp
280 match.ip.recent-list.action: update
280 match.ip.recent-list.hit-count: 100
280 match.ip.recent-list.name: TCP
280 match.ip.recent-list.update-interval: 60
280 match.ip.tcp.dest-port: 53

```



```

match.ip.recent-list.hit-count: 100
match.ip.recent-list.name: UDP
match.ip.recent-list.update-interval: 60
490 match.ip.udp.dest-port: 1645
match.ip.protocol: udp
match.ip.recent-list.action: set
match.ip.recent-list.name: UDP
match.ip.udp.dest-port: 1646
500 action: deny
match.ip.protocol: udp
match.ip.recent-list.action: update
match.ip.recent-list.hit-count: 100
match.ip.recent-list.name: UDP
match.ip.recent-list.update-interval: 60
510 match.ip.udp.dest-port: 1646
action: deny
match.ip.hashlimit.burst: 2
match.ip.hashlimit.expire: 30000
match.ip.hashlimit.mode: src-ip
match.ip.hashlimit.name: TCPRST
match.ip.hashlimit.rate-above: 5/min
match.ip.hashlimit.source-mask: 32
520 match.ip.protocol: tcp
action: deny
match.ip.hashlimit.burst: 30
match.ip.hashlimit.expire: 30000
match.ip.hashlimit.mode: src-ip
match.ip.hashlimit.name: TCPGENRAL
match.ip.hashlimit.rate-above: 50/second
match.ip.hashlimit.source-mask: 32
530 match.ip.protocol: tcp
action: deny
match.ip.hashlimit.burst: 30
match.ip.hashlimit.expire: 3000
match.ip.hashlimit.mode: src-ip
match.ip.hashlimit.name: TCPGENRAL
match.ip.hashlimit.rate-above: 50/second
match.ip.hashlimit.source-mask: 32
560 match.ip.protocol: tcp
action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 161
remark: Whitelist-snmpp
570 action: permit
match.ip.protocol: tcp
match.ip.tcp.dest-port: 443
remark: Whitelist-https
580 action: permit
match.ip.protocol: tcp
match.ip.tcp.dest-port: 22
remark: Whitelist-ssh
590 action: permit
match.ip.protocol: tcp
match.ip.tcp.dest-port: 9339
remark: Whitelist-gnmi
600 action: permit
match.ip.protocol: tcp
match.ip.tcp.dest-port: 636
remark: Whitelist-ldap-tls
610 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 514
remark: Whitelist-rsyslog
620 action: permit
match.ip.protocol: tcp
match.ip.tcp.dest-port: 389
remark: Whitelist-ldap
630 action: permit
match.ip.protocol: tcp
match.ip.tcp.dest-port: 49
remark: Whitelist-tacacs
640 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 123
remark: Whitelist-ntp
650 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 53
remark: Whitelist-dns
660 action: permit
match.ip.protocol: tcp
match.ip.tcp.dest-port: 53
remark: Whitelist-dns
670 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 5353
remark: Whitelist-mDNS
680 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 68
remark: Whitelist-dhcp
690 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 67
remark: Whitelist-dhcp
700 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 4500
remark: Whitelist-IPSec-NAT
710 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 500
remark: Whitelist-IKE
720 action: permit
match.ip.protocol: udp
match.ip.udp.dest-port: 1812

```

```

730    remark:                Whitelist-radius
       action:                permit
       match.ip.protocol:    udp
       match.ip.udp.dest-port: 1813
740    remark:                Whitelist-radius
       action:                permit
       match.ip.protocol:    udp
       match.ip.udp.dest-port: 1645
750    remark:                Whitelist-radius
       action:                permit
       match.ip.protocol:    udp
       match.ip.udp.dest-port: 1646
760    remark:                Whitelist-radius
       action:                permit
       match.ip.protocol:    icmp
770    remark:                Whitelist-icmp
       action:                log
       match.ip.hashlimit.burst: 5
       match.ip.hashlimit.expire: 4294967295
       match.ip.hashlimit.mode: src-ip
       match.ip.hashlimit.name: LOGGING
       match.ip.hashlimit.rate-above: 1/min
       match.ip.hashlimit.source-mask: 32
780    action:                deny

```

Run the `nv show acl ACL_MGMT_INBOUND_CP_DEFAULT --rev=applied -o json` command to show additional information, such as the connection state, hit count and update interval:

```

nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT --rev=applied -o json
...
"630": {
  "action": {
    "permit": {}
  },
  "match": {
    "ip": {
      "connection-state": {
        "established": {},
        "new": {}
      },
      "protocol": "tcp",
      "tcp": {
        "dest-port": {
          "49": {}
        }
      }
    }
  },
  "remark": "Whitelist-tacacs"
},
...
"500": {
  "action": {
    "deny": {}
  },
  "match": {
    "ip": {
      "connection-state": {
        "new": {}
      },
      "protocol": "udp",
      "recent-list": {
        "action": "update",
        "hit-count": 100,
        "name": "UDP",
        "update-interval": 60
      },
      "udp": {
        "dest-port": {
          "1646": {}
        }
      }
    }
  }
},
...

```

To show information about a specific rule, run the `nv show acl <default-acl-id> rule <rule> c` command:

```

nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 500
----- operational ----- applied -----
match
ip
  protocol      udp      udp
  udp
  [dest-port]  1646    1646

```

```

recent-list
name          UDP          UDP
update-interval 60          60
hit-count     100         100
action        update      update
              deny        deny

```

Run the `nv show acl <default-acl-id> rule <rule> --rev=applied -o json` command to see additional information, such as the connection state:

```

nvos@switch:~$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 500 --rev=applied -o json {
  "action": {
    "deny": {}
  },
  "match": {
    "ip": {
      "connection-state": {
        "new": {}
      },
      "protocol": "udp",
      "recent-list": {
        "action": "update",
        "hit-count": 100,
        "name": "UDP",
        "update-interval": 60
      },
      "udp": {
        "dest-port": {
          "1646": {}
        }
      }
    }
  }
}

```

#### 4.1.1.12 Log Messages

Default firewall rules include a log rule for packets that arrive in the control plane and do not match user defined or default firewall rules. The switch generates a log message in `/var/log/firewall_packet_capture.log` for packets that match the log rule.

### 4.1.2 Access Control List Configuration

Recent-list Limiter NVOS Linux provides the NVUE, an NVOS Linux-specific userspace tool to configure custom ACLs on mgmt interfaces including 'eth0' and 'loopback' interfaces.

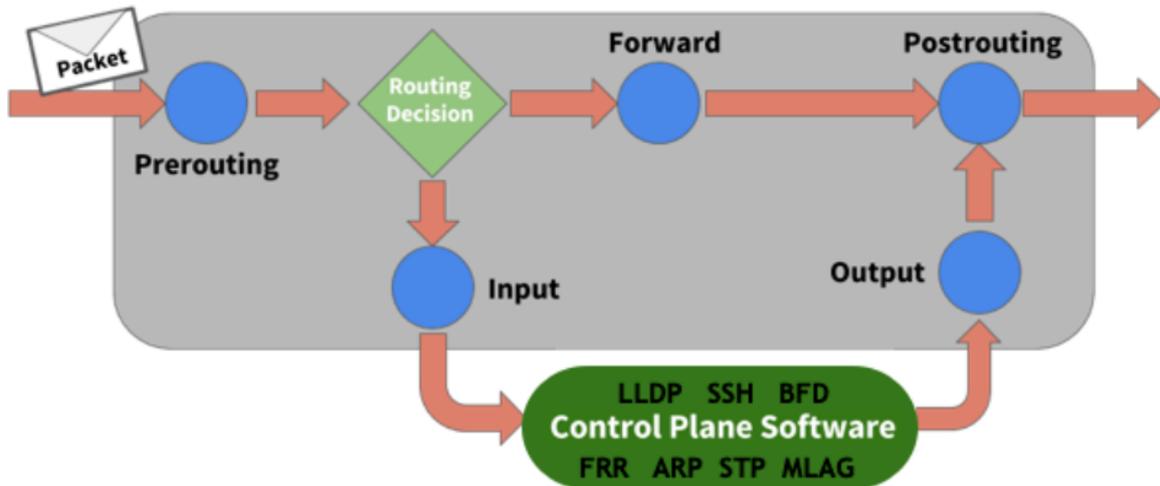
#### 4.1.2.1 Traffic Rules

##### 4.1.2.1.1 Chains

ACLs in NVOS Linux classify and control packets to and from the switch, asserting policies at layers 3 and 4 of the [OSI model](#) by inspecting packet headers according to a list of rules.

The rules inspect or operate on packets at several points (*chains*) in the life of the packet through the system:

## Traffic Inspection Points (Chains)

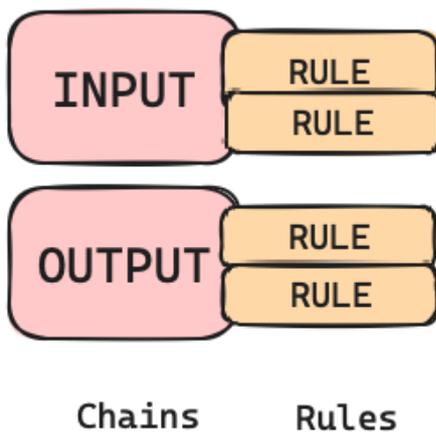


Supported chains are:

- PREROUTING touches packets before the switch routes them.
- INPUT touches packets after the switch determines that the packets are for the local system but before the control plane software receives them.
- OUTPUT touches packets from the control plane software before they leave the switch.
- POSTROUTING touches packets immediately before they leave the switch but after a routing decision.

### 4.1.2.1.2 Rules

Rules classify the traffic you want to control. You apply rules to chains.



### 4.1.2.2 Install and Manage ACL Rules with NVUE

NVOS provides a comfortable way to manage ACL rules configurations on the system using NVUE.

Consider the following example, where we want to accept packets matching the following criteria:

1. TCP packets.
2. Source IP address— 10.0.14.2/32
3. Any source port.
4. Destination IP address— 10.0.15.8/32
5. Any destination port.

The steps we need to perform are:

1. Set the rule type, the matching protocol, source IP address and port, destination IP address and port, and the action. You must provide a name for the rule (EXAMPLE1 in the commands below):

```
admin@nvos:~$ nv set acl EXAMPLE1 type ipv4
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip source-ip 10.0.14.2/32
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip tcp source-port ANY
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip dest-ip 10.0.15.8/32
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip tcp dest-port ANY
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 action permit
```

2. Apply the rule to an inbound or outbound interface with the `nv set interface <interface> acl` command.

- For rules affecting the PREROUTING or POSTROUTING chain, apply the rule to an inbound or outbound interface accordingly: For example:

```
admin@nvos:~$ nv set interface eth0 acl EXAMPLE1 inbound
admin@nvos:~$ nv config apply
```

- For rules affecting the INPUT or OUPUT chain, apply the rule to inbound control plane or outbound control plane interface accordingly. For example:

```
admin@nvos:~$ nv set interface eth0 acl EXAMPLE1 inbound control-plane
admin@nvos:~$ nv config apply
```



If an ACL is not applied to an interface it will not take any effect during the process of a packet.

To see the configured rule run the NVUE `nv show acl <rule-name> rule <ID>` command:

```
admin@nvos:~$ nv show acl EXAMPLE1 rule 10
----- operational applied -----
match
ip
  source-ip      10.0.14.2/32  10.0.14.2/32
  dest-ip        10.0.15.8/32  10.0.15.8/32
  protocol       tcp           tcp
  tcp
  [source-port] ANY          ANY
  [dest-port]   ANY          ANY
```

To remove this rule, run the `nv unset acl <acl-name>` and `nv unset interface <interface> acl <acl-name>` commands.

```
admin@nvos:~$ nv unset acl EXAMPLE1
admin@nvos:~$ nv unset interface eth0 acl EXAMPLE1
admin@nvos:~$ nv config apply
```

To show ACL statistics per interface, such as the total number of bytes that match the ACL rule, run the `nv show interface <interface-id> acl <acl-id> statistics` or `nv show interface <interface-id> acl <acl-id> statistics <rule-id>` command, for example:

```
admin@nvos:~$ nv show interface eth0 acl EXAMPLE1 statistics
Rule  In Packet  In Byte  Out Packet  Out Byte  Layer  Remark  Action  Summary
-----
10    0           0         0           0           ip      permit  match.ip.dest-ip: 10.0.15.8/32
match.ip.protocol: tcp
match.ip.source-ip: 10.0.14.2/32
match.ip.tcp.dest-port: ANY
match.ip.tcp.source-port: ANY
```

To see the list of all NVUE ACL commands, run the `nv list-commands acl` command.

#### 4.1.2.2.1 Rule Support

Rule Element	Supported
<b>Matches</b>	Src/Dst, IP protocol In/out interface IPv4: ecn, icmp IPv6: icmpv6, routing-header, extension-header IP common: tcp (with flags), udp, multiport, frag, mss, connection-state
<b>Standard Actions</b>	permit, deny, log (log with log prefix), empty (no action specified)
<b>Limiters</b>	Recent-list Hashlimit

#### 4.1.2.3 Default Action

The default action for each rule defined is no action. Packet matching such rules will not be affected and will continue to be processed by the proceeding rules in order.

#### 4.1.2.4 Common Examples

##### 4.1.2.4.1 Control Plane Limiters

NVUE allows you to configure rate limit on traffic, so incoming packets drop if they exceed certain thresholds. NVUE provides two limiters to achieve it: recent-list and hashlimit.

###### 4.1.2.4.1.1 Recent-List Limiter

Allows you to dynamically create a list of IP addresses and then match against that list in a few different ways, for example:

```
admin@nvos:~$ nv set acl EXAMPLE2 type ipv4
admin@nvos:~$
admin@nvos:~$ nv set acl EXAMPLE2 rule 10 match ip tcp dest-port 22
```

```

admin@nvos:~$ nv set acl EXAMPLE2 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl EXAMPLE2 rule 10 match ip recent-list action set
admin@nvos:~$ nv set acl EXAMPLE2 rule 10 match ip recent-list name TCP-SSH-LIMIT
admin@nvos:~$
admin@nvos:~$ nv set acl EXAMPLE2 rule 20 match ip tcp dest-port 22
admin@nvos:~$ nv set acl EXAMPLE2 rule 20 match ip protocol tcp
admin@nvos:~$ nv set acl EXAMPLE2 rule 20 match ip recent-list action update
admin@nvos:~$ nv set acl EXAMPLE2 rule 20 match ip recent-list update-interval 60
admin@nvos:~$ nv set acl EXAMPLE2 rule 20 match ip recent-list hit-count 100
admin@nvos:~$ nv set acl EXAMPLE2 rule 20 match ip recent-list name TCP-SSH-LIMIT
admin@nvos:~$ nv set acl EXAMPLE2 rule 20 action deny
admin@nvos:~$
admin@nvos:~$ nv set interface eth0 acl EXAMPLE2 inbound control-plane
admin@nvos:~$ nv config apply

```

The above example, limits any source IP address sending more than 100 packets per 60-second interval to the switch, if it exceeds this rate all packets from this source IP address will be blocked.



Configuring the recent-list limiter consists of two consecutive rules, both rules should contain the same matching criteria (protocol tcp and dest-port 22 in the above example) and the name of the recent-list (TCP-SSH-LIMIT in the above example).

1. The first rule is set to action 'set'
2. The second rule is set to recent-list action 'update' and specified with the requested threshold using 'hit-count' and 'update-interval' (100 packets per 60-second interval in the above example)
3. The second rule action is set to deny

#### 4.1.2.4.1.2 Hashlimit Limiter

Uses hash buckets to express a rate limiting match for a group of connections using a single rule. Grouping can be done per-hostgroup (source and/or destination address). It gives you the ability to express "Npackets per time quantum per group", for example:

```

admin@nvos:~$ nv set acl EXAMPLE3 type ipv4
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip tcp dest-port 22
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip hashlimit name TCP-SSH-LIMIT
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip hashlimit rate-above 5/min
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip hashlimit burst 2
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip hashlimit expire 3000
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip hashlimit mode src-ip
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 match ip hashlimit source-mask 32
admin@nvos:~$ nv set acl EXAMPLE3 rule 10 action deny
admin@nvos:~$ nv set interface eth0 acl EXAMPLE2 inbound control-plane
admin@nvos:~$ nv config apply

```

The above example, limits any source IP address sending more than 5 packets per minute to the switch with a burst of 2 packets, if it exceeds this rate all packets from this source IP address will be blocked for 3000 milliseconds as specified in the expire parameter and will be able to send again after this period.



Configuring the recent-list limiter can be configured in one single ACL rule. The following parameters need to be configured for the hashlimit: name, rate-above, burst, expire and mode. The source-mask or destination-mask are optional.

#### 4.1.2.4.2 Filter Specific TCP Flags

The example rule below drops ingress IPv4 TCP packets when you set the SYN bit and reset the RST, ACK, and FIN bits. The rule applies inbound on interface eth0. After configuring this rule, you

cannot establish new TCP sessions that originate from ingress mgmt port eth0. You can establish TCP sessions that originate from any other port.

#### 4.1.2.4.2.1 NVUE Commands

```
admin@nvos:~$ nv set acl EXAMPLE4 type ipv4
admin@nvos:~$ nv set acl EXAMPLE4 rule 20 match ip protocol tcp
admin@nvos:~$ nv set acl EXAMPLE4 rule 20 match ip tcp flags syn
admin@nvos:~$ nv set acl EXAMPLE4 rule 20 match ip tcp mask rst
admin@nvos:~$ nv set acl EXAMPLE4 rule 20 match ip tcp mask syn
admin@nvos:~$ nv set acl EXAMPLE4 rule 20 match ip tcp mask fin
admin@nvos:~$ nv set acl EXAMPLE4 rule 20 match ip tcp mask ack
admin@nvos:~$ nv set acl EXAMPLE4 rule 20 action deny
admin@nvos:~$ nv set interface eth0 acl EXAMPLE4 inbound
admin@nvos:~$ nv config apply
```

#### 4.1.2.4.3 Control Who Can SSH into the Switch

Run the following commands to control who can SSH into the switch. In the following example, 10.10.10.1/32 is the interface IP address (or loopback IP address) of the switch and 10.255.4.0/24 can SSH into the switch.

##### 4.1.2.4.3.1 NVUE Commands

```
admin@nvos:~$ nv set acl example2 type ipv4
admin@nvos:~$ nv set acl example2 rule 10 match ip source-ip 10.255.4.0/24
admin@nvos:~$ nv set acl example2 rule 10 match ip dest-ip 10.10.10.1/32
admin@nvos:~$ nv set acl example2 rule 10 action permit
admin@nvos:~$ nv set acl example2 rule 20 match ip source-ip ANY
admin@nvos:~$ nv set acl example2 rule 20 match ip dest-ip 10.10.10.1/32
admin@nvos:~$ nv set acl example2 rule 20 action deny
admin@nvos:~$ nv set interface eth0 acl example2 inbound
admin@nvos:~$ nv config apply
```

#### 4.1.2.4.4 Match on ECN Bits in the TCP IP Header

ECN allows end-to-end notification of network congestion without dropping packets. You can add ECN rules to match on the ECE, CWR, and ECT flags in the TCP IPv4 header.

By default, ECN rules match a packet with the bit set. You can reverse the match by using an explanation point (!).

##### 4.1.2.4.4.1 Match on the ECE Bit

After an endpoint receives a packet with the CE bit set by a router, it sets the ECE bit in the returning ACK packet to notify the other endpoint that it needs to slow down.

To match on the ECE bit:

NVUE Commands

```
admin@nvos:~$ nv set acl example2 type ipv4
admin@nvos:~$ nv set acl example2 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl example2 rule 10 match ip ecn flags tcp-ecce
admin@nvos:~$ nv set acl example2 rule 10 action permit
admin@nvos:~$ nv set interface eth0 acl example2 inbound
admin@nvos:~$ nv config apply
```

##### 4.1.2.4.4.2 Match on the CWR Bit

The CWR bit notifies the other endpoint of the connection that it received and reacted to an ECE.

To match on the CWR bit:

NVUE Commands

```
admin@nvos:~$ nv set acl example2 type ipv4
admin@nvos:~$ nv set acl example2 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl example2 rule 10 match ip ecn flags tcp-cwr
admin@nvos:~$ nv set acl example2 rule 10 action permit
admin@nvos:~$ nv set interface eth0 acl example2 inbound
admin@nvos:~$ nv config apply
```

#### 4.1.2.4.4.3 Match on the ECT Bit

The ECT codepoints negotiate if the connection is ECN capable by setting one of the two bits to 1. Routers also use the ECT bit to indicate that they are experiencing congestion by setting both the ECT codepoints to 1.

To match on the ECT bit:

NVUE Commands

```
admin@nvos:~$ nv set acl example2 type ipv4
admin@nvos:~$ nv set acl example2 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl example2 rule 10 match ip ecn ip-ect 1
admin@nvos:~$ nv set acl example2 rule 10 action permit
admin@nvos:~$ nv set interface eth0 acl example2 inbound
admin@nvos:~$ nv config apply
```

#### 4.1.2.4.5 Set DSCP on Transit Traffic

The following examples use the *mangle* table to modify the packet as it transits the switch. DSCP is in decimal notation in the examples below.

To set SSH as high priority traffic:

```
admin@nvos:~$ nv set acl EXAMPLE1 type ipv4
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip tcp dest-port 22
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 action set dscp 46
admin@nvos:~$ nv set interface eth0 acl EXAMPLE1 inbound
admin@nvos:~$ nv config apply
```

To set everything coming in swp1 as AF13:

```
admin@nvos:~$ nv set acl EXAMPLE1 type ipv4
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 action set dscp 14
admin@nvos:~$ nv set interface eth0 acl EXAMPLE1 inbound
admin@nvos:~$ nv config apply
```

To set Packets destined for 10.0.100.27 as best effort:

```
admin@nvos:~$ nv set acl EXAMPLE1 type ipv4
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip dest-ip 10.0.100.27/32
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 action set dscp 0
admin@nvos:~$ nv set interface eth0 acl EXAMPLE1 inbound
admin@nvos:~$ nv config apply
```

To use a range of ports for TCP traffic:

```
admin@nvos:~$ nv set acl EXAMPLE1 type ipv4
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip protocol tcp
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip source-ip 10.0.0.17/32
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip tcp source-port 10000:20000
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip dest-ip 10.0.100.27/32
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 match ip tcp dest-port 10000:20000
```

```
admin@nvos:~$ nv set acl EXAMPLE1 rule 10 action set dscp 34
admin@nvos:~$ nv set interface eth0 acl EXAMPLE1 inbound
admin@nvos:~$ nv config apply
```



To specify all ports on the switch in NVUE (swp+ in an iptables rule), you must set the range of interfaces on the switch as in the examples above (`nv set interface swp1-48`). This command creates as many rules in the `/etc/cumulus/acl/policy.d/50_nvue.rules` file as the number of interfaces in the range you specify.

### 4.1.3 Access Control List Commands

- [4.1.3.1 nv show acl](#)
- [4.1.3.2 nv unset acl](#)
- [4.1.3.3 nv show acl id](#)
- [4.1.3.4 nv set/unset acl id](#)
- [4.1.3.5 nv set/unset acl type](#)
- [4.1.3.6 nv show acl rule](#)
- [4.1.3.7 nv show acl rule id](#)
- [4.1.3.8 nv set/unset acl rule](#)
- [4.1.3.9 nv set/unset acl rule remark](#)
- [4.1.3.10 nv show acl rule action](#)
- [4.1.3.11 nv set/unset acl rule action permit](#)
- [4.1.3.12 nv set/unset acl rule action deny](#)
- [4.1.3.13 nv set/unset acl rule action log log-prefix](#)
- [4.1.3.14 nv show acl rule match](#)
- [4.1.3.15 nv set/unset acl rule match](#)
- [4.1.3.16 nv show acl rule match ip](#)
- [4.1.3.17 nv set/unset acl rule match ip](#)
- [4.1.3.18 nv show acl rule match ip udp](#)
- [4.1.3.19 nv show acl rule match ip udp dest-port](#)
- [4.1.3.20 nv set/unset acl rule match ip udp dest-port](#)
- [4.1.3.21 nv show acl rule match ip udp source-port](#)
- [4.1.3.22 nv set/unset acl rule match ip udp source-port](#)
- [4.1.3.23 nv show acl rule match ip tcp](#)
- [4.1.3.24 nv show acl rule match ip tcp dest-port](#)
- [4.1.3.25 nv set/unset acl rule match ip tcp dest-port](#)
- [4.1.3.26 nv show acl rule match ip tcp source-port](#)
- [4.1.3.27 nv set/unset acl rule match ip tcp source-port](#)
- [4.1.3.28 nv show acl rule match ip tcp flags](#)
- [4.1.3.29 nv set/unset acl rule match ip tcp flags](#)
- [4.1.3.30 nv show acl rule match ip tcp mask](#)
- [4.1.3.31 nv set/unset acl rule match ip tcp mask](#)
- [4.1.3.32 nv set/unset acl rule match ip tcp mss](#)
- [4.1.3.33 nv set/unset acl rule match ip tcp all-mss-except](#)
- [4.1.3.34 nv set/unset acl rule match ip fragment](#)
- [4.1.3.35 nv show acl rule match ip ecn](#)
- [4.1.3.36 nv set/unset acl rule match ip ecn](#)

- [4.1.3.37 nv set/unset acl rule match ip ecn ip-ect](#)
- [4.1.3.38 nv set/unset acl rule match ip ecn flags](#)
- [4.1.3.39 nv show acl rule match ip connection-state](#)
- [4.1.3.40 nv set/unset acl rule match ip connection-state](#)
- [4.1.3.41 nv show acl rule match ip extension-header](#)
- [4.1.3.42 nv set/unset acl rule match ip extension-header type](#)
- [4.1.3.43 nv show acl rule match ip routing-header](#)
- [4.1.3.44 nv set/unset acl ACL rule match ip routing-header type](#)
- [4.1.3.45 nv set/unset acl ACL rule match ip source-ip](#)
- [4.1.3.46 nv set/unset acl ACL rule match ip dest-ip](#)
- [4.1.3.47 nv set/unset acl rule match ip protocol](#)
- [4.1.3.48 nv set/unset acl rule match ip icmp-type](#)
- [4.1.3.49 nv set/unset acl rule match ip icmpv6-type](#)
- [4.1.3.50 nv show acl rule match ip recent-list](#)
- [4.1.3.51 nv set/unset acl rule match ip recent-list name](#)
- [4.1.3.52 nv set/unset acl rule match ip recent-list action](#)
- [4.1.3.53 nv set/unset acl rule match ip recent-list hit-count](#)
- [4.1.3.54 nv set/unset acl rule match ip recent-list update-interval](#)
- [4.1.3.55 nv show acl rule match ip hashlimit](#)
- [4.1.3.56 nv set/unset acl rule match ip hashlimit name](#)
- [4.1.3.57 nv set/unset acl rule match ip hashlimit rate-above](#)
- [4.1.3.58 nv set/unset acl rule match ip hashlimit burst](#)
- [4.1.3.59 nv set/unset acl rule match ip hashlimit expire](#)
- [4.1.3.60 nv set/unset acl rule match ip hashlimit mode](#)
- [4.1.3.61 nv set/unset acl rule match ip hashlimit destination-mask](#)
- [4.1.3.62 nv set/unset acl rule match ip hashlimit source-mask](#)
- [4.1.3.63 nv show interface acl](#)
- [4.1.3.64 nv show interface acl id](#)
- [4.1.3.65 nv show interface acl statistics](#)
- [4.1.3.66 nv show interface acl statistics](#)
- [4.1.3.67 nv show interface acl outbound](#)
- [4.1.3.68 nv show interface acl outbound control-plane](#)
- [4.1.3.69 nv show interface acl inbound](#)
- [4.1.3.70 nv show interface acl inbound control-plane](#)
- [4.1.3.71 nv set/unset interface acl inbound](#)
- [4.1.3.72 nv set/unset interface acl inbound control-plane](#)
- [4.1.3.73 nv set/unset interface acl outbound control-plane](#)
- [4.1.3.74 nv set/unset interface acl outbound](#)
- [4.1.3.75 nv action clear acl counters](#)
- [4.1.3.76 nv set acl rule action set dscp](#)

### 4.1.3.1 nv show acl

	nv show acl Display all available ACLs on the system.
Syntax Description	N/A

History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show acl ACL ----- ACL1          ipv4    rule: 1 Test         ipv4    rule: 2               rule: 3               rule: 4               ... ACL_MGMT_INBOUND_DEFAULT  ipv6    rule: 10               rule: 20               rule: 30               rule: 40 ... custom        ipv6    rule: 5 </pre>
REST API	GET https://<ip>/nvue_v1/acl
Related Commands	nv set acl
Notes	<ul style="list-style-type: none"> <li>By default, there are ACLs configured on the system as part of the default rules. The corresponding ACL names are as follows: <ul style="list-style-type: none"> <li>ACL_LOOPBACK_INBOUND_CP_DEFAULT—IPv4 default rules bound to the loopback interface in the inbound control-plane direction</li> <li>ACL_LOOPBACK_INBOUND_CP_DEFAULT_IPV6—IPv6 default rules bound to the loopback interface in the inbound control-plane direction</li> <li>ACL_MGMT_INBOUND_CP_DEFAULT—IPv4 default rules bound to the mgmt interface in the inbound control-plane direction</li> <li>ACL_MGMT_INBOUND_CP_DEFAULT_IPV6—IPv6 default rules bound to the mgmt interface in the inbound control-plane direction</li> <li>ACL_MGMT_INBOUND_DEFAULT—IPv4 default rules bound to the mgmt interface in the inbound direction</li> <li>ACL_MGMT_INBOUND_DEFAULT_IPV6—IPv6 default rules bound to the mgmt interface in the inbound direction</li> <li>ACL_MGMT_OUTBOUND_CP_DEFAULT— IPv4 default rules bound to the mgmt interface in the outbound control-plane direction</li> <li>ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6—IPv6 default rules bound to the mgmt interface in the outbound control-plane direction</li> </ul> </li> <li>Each ACL can have either IPv4 or IPv6.</li> </ul>

#### 4.1.3.2 nv unset acl

	nv unset acl Clear all the new configured ACLs and restore the original default ACLs.
Syntax Description	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv unset acl </pre>
REST API	DELETE https://<ip>/nvue_v1/acl
Related Commands	nv show acl
Notes	This command will remove the modifications/extra ACLs configured on the system and restore to the original default ACLs.

### 4.1.3.3 nv show acl id

	nv show acl <acl-id> Get ACL <acl-id> information (i.e., rule-ids and the ACL type: ipv4 or ipv6).	
Syntax Description	acl-id	ACL name
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_DEFAULT       operational applied ----- type  ipv4         ipv4  rule =====       Number  Summary -----       10      action:                deny               match.ip.protocol:          tcp               match.ip.tcp.all-mss-except: 536-65535           </pre>	
REST API	GET https://<ip>/nvue_v1/acl/<acl-id>	
Related Commands	nv show acl	
Notes		

### 4.1.3.4 nv set/unset acl id

	nv set acl <acl-id> nv unset acl <acl-id> Create a new custom ACL Delete an existing ACL.	
Syntax Description	acl-id	New, custom ACL name
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv set acl EXAMPLE_ACL           </pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/<acl-id>	
Related Commands	nv show acl	
Notes	<ul style="list-style-type: none"> <li>ACL name can be chosen to any generic name but is important later on binding multiple ACLs on the same interface and same direction since ACLs list of rules will be ordered with lexicographical order.             <ul style="list-style-type: none"> <li>For example, ACL with name 'A' that has 10 rules and acl with name 'B' with 5 rules, if bound to the same direction on the same interface, the 10 rules of acl 'A' will be before the 5 rules of acl 'B'.</li> </ul> </li> <li>This command is not enough for applying this custom acl, it needs to have at <b>least one rule</b> in it and needs to belong to the ip <b>type either ipv4 or ipv6</b>.</li> <li>The unset command will not remove the specified ACL if it is bound to an interface. The user must unbind it and then use this command to delete the ACL.</li> <li>Unset of default ACL will restore the original list of rules of that ACL.</li> </ul>	

### 4.1.3.5 nv set/unset acl type

	nv set acl <acl-id> type <acl-type> nv unset acl <acl-id> type <acl-type> Add ACL type, whether it is an IPv4 or IPv6 ACL.	
Syntax Description	acl-id	New, custom ACL name
	acl-type	Enum: ipv4   ipv6
History	25.02.1884	
Example		
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/<acl-id>/type/	
Related Commands	nv show acl	
Notes	Each ACL must have a type	

### 4.1.3.6 nv show acl rule

	nv show acl <acl-id> rule Display all the rules configured on the specified ACL.	
Syntax Description	acl-id	ACL name
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule Number  Summary -----  -  1      action:                                log  10     action:                                deny       match.ip.dest-ip:                    127.0.0.0/8  20     action:                                permit  30     action:                                deny       match.ip.protocol:                  tcp  40     action:                                deny       match.ip.protocol:                  tcp  50     action:                                deny       match.ip.protocol:                  tcp  60     action:                                deny       match.ip.protocol:                  tcp  70     action:                                deny  80     action:                                deny       match.ip.protocol:                  tcp  90     action:                                deny       match.ip.protocol:                  tcp 100     action:                                deny 110     match.ip.protocol:                  tcp           </pre>	
REST API	GET https://<ip>/nvue_v1/acl/<acl-id>/rule	
Related Commands	nv show acl <acl-id>	
Notes		

### 4.1.3.7 nv show acl rule id

	nv show acl <acl-id> rule <rule-id> Show ACL rule <rule-id> configurations.	
--	--	--

Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 10 ----- operational applied ----- match ip   dest-ip 127.0.0.0/8 127.0.0.0/8 action   deny         deny</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

#### 4.1.3.8 nv set/unset acl rule

	nv set acl <acl-id> rule <rule-id> nv unset acl <acl-id> rule <rule-id> Set/remove ACL rule <rule-id> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl user_custom_acl rule 10</pre>	
REST API	PATH https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• This command is used to declare the specified rule with the specified ACL.</li> <li>• Mere application of configuration is insufficient. Matching criteria on either the packet or action for this rule must be specified in order to be effective.</li> </ul>	

#### 4.1.3.9 nv set/unset acl rule remark

	nv set acl <acl-id> rule <rule-id> remark <string> nv unset acl <acl-id> rule <rule-id> remark <string> Set/remove ACL rule <rule-id> remark configurations (remark is the same as description).	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl user_custom_acl rule 20 remark "MY-PROTECTIVE-RULE"</pre>	

REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/action/deny
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	The remark acts the same as a description of a rule.

#### 4.1.3.10 nv show acl rule action

	nv show acl <acl-id> rule <rule-id> action Show ACL rule <rule-id> action configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 10 action operational  applied ----- deny          deny</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/action	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

#### 4.1.3.11 nv set/unset acl rule action permit

	nv set acl <acl-id> rule <rule-id> action permit nv unset acl <acl-id> rule <rule-id> action permit Set/remove ACL rule <rule-id> action permit.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl user_custom_acl rule 10 action permit</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/action/permit	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• Only one action per rule can be specified.</li> <li>• Any rule matching the specified rule will be accepted to the system.</li> <li>• Leaving a rule with no action will leave any packet matching the specified rule unaffected.</li> </ul>	

### 4.1.3.12 nv set/unset acl rule action deny

	nv set acl <acl-id> rule <rule-id> action deny nv unset acl <acl-id> rule <rule-id> action deny Set/remove ACL rule <rule-id> action deny.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl user_custom_acl rule 20 action deny</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/action/deny	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• Only one action per rule can be specified.</li> <li>• Any rule matching the specified rule will be rejected by the system and will be processed any further.</li> <li>• Leaving a rule with no action will leave any packet matching the specified rule unaffected.</li> </ul>	

### 4.1.3.13 nv set/unset acl rule action log log-prefix

	nv set acl <acl-id> rule <rule-id> action log log-prefix <str> nv unset acl <acl-id> rule <rule-id> action log log-prefix <str> Set/remove ACL rule <rule-id> action log log-prefix <str>.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	log-prefix-str	String
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl user_custom_acl rule 20 action log</pre> <pre>admin@nvos:~\$ nv set acl user_custom_acl rule 30 action log log-prefix "Dropped-by-custom-acl"</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/action/log PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/action/log/log-prefix/<log-prefix-str>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• Only one action per rule can be specified.</li> <li>• Log-prefix can be set to empty string.</li> <li>• Any packet matching specified rule with logging action will be logged to netfilter log.</li> <li>• Leaving a rule with no action will leave any packet matching the specified rule unaffected.</li> </ul>	

#### 4.1.3.14 nv show acl rule match

	nv show acl <acl-id> rule <rule-id> match Show ACL rule <rule-id> match configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 10 match               operational  applied ----- ip   dest-ip  127.0.0.0/8  127.0.0.0/8</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	Currently, displaying the matching criteria for the rule only contains layer 3 and 4 filtering criteria in the OSI model.	

#### 4.1.3.15 nv set/unset acl rule match

	nv set acl <acl-id> rule <rule-id> match nv unset acl <acl-id> rule <rule-id> match Set/remove ACL rule <rule-id> match.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl user_custom_acl rule 20 match</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Leaving a rule with empty matching criteria will cause the rule to match any packet.</li> <li>The unset form of the command will remove the match criteria of the rule.</li> </ul>	

#### 4.1.3.16 nv show acl rule match ip

	nv show acl <acl-id> rule <rule-id> match ip Show ACL rule <rule-id> match IP configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 10 match ip               operational  applied ----- dest-ip      127.0.0.0/8  127.0.0.0/8</pre>
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	Displays the matching IP criteria for the rule.

#### 4.1.3.17 nv set/unset acl rule match ip

	nv set acl <acl-id> rule <rule-id> match ip nv unset acl <acl-id> rule <rule-id> match ip Set/remove ACL rule <rule-id> match ip configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset acl user_custom_acl rule 20 match ip</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Leaving a rule with empty IP matching criteria will cause the rule to match any packet.</li> <li>The unset command will remove the IP match criteria of the rule.</li> </ul>	

#### 4.1.3.18 nv show acl rule match ip udp

	nv show acl <acl-id> rule <rule-id> match ip udp Show ACL rule <rule-id> match IP UDP configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 650 match ip udp               operational  applied ----- [dest-port]  53                53</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/udp	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Display the matching UDP IP criteria for the rule.</li> <li>The output primarily contains either the source port or destination port.</li> </ul>	

### 4.1.3.19 nv show acl rule match ip udp dest-port

	nv show acl <acl-id> rule <rule-id> match ip udp dest-port Show ACL rule <rule-id> match IP UDP dest-port configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl custom-acl rule 650 match ip udp dest-port Ports ----- 53 22</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/udp/dest-port	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>The command displays the matching dest-port of UDP IP criteria for the rule.</li> <li>The rule can have more than dest-port configured.</li> </ul>	

### 4.1.3.20 nv set/unset acl rule match ip udp dest-port

	nv set acl <acl-id> rule <rule-id> match ip udp dest-port <port-num> nv unset acl <acl-id> rule <rule-id> match ip udp dest-port <port-num> Configure/remove ACL rule <rule-id> match IP UDP dest-port <port-num> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	port-num	IP port ID (integer: 0-65535   enum: ANY, bootpc, bootps, clag, dhcp-client, dhcp-server, domain, ftp, http, https, imap2, ldap, ldaps, ntp, msdp, pop3, smtp, snmp, snmp-trap, ssh, telnet, tftp   ip-port-range)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip udp dest-port 22 admin@nvos:~\$ nv set acl custom-acl rule 650 match ip udp dest-port 53</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/udp/dest-port/<port-num>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>The rule can have more than dest-port configured.</li> <li>Rule cannot be configured with more than one source port and more than one dest-port at the same time in the same rule. For example, the user cannot configure ports 22, 53 on the dest-port and 1813 on the source-port, but can configure 22 on dest-port and 1813 on source-port.</li> <li>The user can configure more than one port on dest-port or source-port.</li> </ul>	

### 4.1.3.21 nv show acl rule match ip udp source-port

	nv show acl <acl-id> rule <rule-id> match ip udp source-port Show ACL rule <rule-id> match IP UDP source-port configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl custom-acl rule 650 match ip udp source-port Ports ----- 53 22</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/udp/source-port	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• Display the matching dest-port of UDP IP criteria for the rule.</li> <li>• The rule can have more than dest-port configured.</li> </ul>	

### 4.1.3.22 nv set/unset acl rule match ip udp source-port

	nv set acl <acl-id> rule <rule-id> match ip udp source-port <port-num> nv unset acl <acl-id> rule <rule-id> match ip udp source-port <port-num> Configure/remove ACL rule <rule-id> match IP UDP source-port <port-num> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	port-num	IP port ID (integer: 0-65535   enum:ANY, bootpc, bootps, clag, dhcp-client, dhcp-server, domain, ftp,http, https, imap2, ldap, ldaps, ntp, msdp, pop3, smtp,snmp, snmp-trap,ssh, telnet, tftp   ip-port-range)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip udp source-port 22 admin@nvos:~\$ nv set acl custom-acl rule 650 match ip udp source-port 53</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/udp/source-port/<port-num>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• The rule can have more than source-port configured.</li> <li>• Rule cannot be configured with more than one source port and more than one dest-port at the same time in the same rule. For example, the user cannot configure ports 22, 53 on the dest-port and 1813 on the source-port, but can configure 22 on dest-port and 1813 on source-port.</li> <li>• The user can configure more than one port on dest-port or source-port.</li> </ul>	

### 4.1.3.23 nv show acl rule match ip tcp

	nv show acl <acl-id> rule <rule-id> match ip tcp Show ACL rule <rule-id> match ip tcp configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 650 match ip tcp               operational  applied ----- [dest-port]  53           53</pre> <pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_DEFAULT rule 10 match ip tcp               operational  applied ----- all-mss-except  536-65535  536-65535</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• Display the matching UDP IP criteria for the rule.</li> <li>• The output primarily contains any of the source-port, dest-port, flags, mask, mss, all-mss-except.</li> </ul>	

### 4.1.3.24 nv show acl rule match ip tcp dest-port

	nv show acl <acl-id> rule <rule-id> match ip tcp dest-port Show ACL rule <rule-id> match IP TCP dest-port configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl custom-acl rule 650 match ip tcp dest-port Ports ----- 53 22</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/dest-port	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• Display the matching dest-port of TCP IP criteria for the rule.</li> <li>• The rule can have more than dest-port configured.</li> </ul>	

### 4.1.3.25 nv set/unset acl rule match ip tcp dest-port

	nv set acl <acl-id> rule <rule-id> match ip tcp dest-port <port-num> nv unset acl <acl-id> rule <rule-id> match ip tcp dest-port <port-num> Configure/remove ACL rule <rule-id> match ip tcp dest-port <port-num> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	port-num	IP port ID (integer: 0-65535   enum: ANY, bootpc, bootps, clag, dhcp-client, dhcp-server, domain, ftp,http, https, imap2, ldap, ldaps, ntp, msdp, pop3, smtp,snmp, snmp-trap,ssh, telnet, tftp   ip-port-range)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp dest-port 22 admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp dest-port 53</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/dest-port/<port-num>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• The rule can have more than source-port configured.</li> <li>• Rule cannot be configured with more than one source port and more than one dest-port at the same time in the same rule. For example, the user cannot configure ports 22, 53 on the dest-port and 1813 on the source-port, but can configure 22 on dest-port and 1813 on source-port.</li> <li>• The user can configure more than one port on dest-port or source-port.</li> </ul>	

### 4.1.3.26 nv show acl rule match ip tcp source-port

	nv show acl <acl-id> rule <rule-id> match ip tcp source-port Show ACL rule <rule-id> match IP TCP source-port configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl custom-acl rule 650 match ip tcp source-port Ports ----- 53 22</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/source-port	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• Display the matching dest-port of UDP IP criteria for the rule.</li> <li>• The rule can have more than the dest-port configured.</li> </ul>	

### 4.1.3.27 nv set/unset acl rule match ip tcp source-port

	nv set acl <acl-id> rule <rule-id> match ip tcp source-port <port-num> nv unset acl <acl-id> rule <rule-id> match ip tcp source-port <port-num> Configure/remove ACL rule <rule-id> match ip tcp source-port <port-num> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	port-num	IP port ID (integer: 0-65535   enum:ANY, bootpc, bootps, clag, dhcp-client, dhcp-server, domain, ftp,http, https, imap2, ldap, ldaps, ntp, msdp, pop3, smtp,snmp, snmp-trap,ssh, telnet, tftp   ip-port-range)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp source-port 22 admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp source-port 53</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/source-port/<port-num>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>The rule can have more than source-port configured.</li> <li>Rule cannot be configured with more than one source port and more than one dest-port at the same time in the same rule. For example, the user cannot configure ports 22, 53 on the dest-port and 1813 on the source-port, but can configure 22 on dest-port and 1813 on source-port.</li> <li>The user can configure more than one port on dest-port or source-port.</li> </ul>	

### 4.1.3.28 nv show acl rule match ip tcp flags

	nv show acl <acl-id> rule <rule-id> match ip tcp flags Show ACL rule <rule-id> match ip tcp flags configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 60 match ip tcp flags -o json {   "none": {} }</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/flags	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

### 4.1.3.29 nv set/unset acl rule match ip tcp flags

	nv set acl <acl-id> rule <rule-id> match ip tcp flags (syn   ack   fin   rst   urg   psh   all   none) nv unset acl <acl-id> rule <rule-id> match ip tcp flags (syn   ack   fin   rst   urg   psh   all   none) Configure/remove ACL rule <rule-id> match ip tcp flags <flag-id> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	flag-id	enum: (syn   ack   fin   rst   urg   psh   all   none)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp flags all admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp flags urg admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp flags psh admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp flags syn</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/flags/<flag-id>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>The user can configure multiple flags that are not 'none' or 'all'.</li> <li>The flag configurations must come with TCP mask configurations.</li> </ul>	

### 4.1.3.30 nv show acl rule match ip tcp mask

	nv show acl <acl-id> rule <rule-id> match ip tcp mask ACL rule <rule-id> match IP TCP mask configuration.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT rule 60 match ip tcp mask -o json {   "ack": {},   "fin": {},   "rst": {},   "syn": {} }</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/mask	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

### 4.1.3.31 nv set/unset acl rule match ip tcp mask

	nv set acl <acl-id> rule <rule-id> match ip tcp mask (syn   ack   fin   rst   urg   psh   all   none) nv unset acl <acl-id> rule <rule-id> match ip tcp mask (syn   ack   fin   rst   urg   psh   all   none) Configure/remove ACL rule <rule-id> match ip tcp mask <flag-id> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	flag-id	enum: (syn   ack   fin   rst   urg   psh   all   none)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp mask all admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp mask urg admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp mask psh admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp mask syn</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/flags/<flag-id>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>The user can configure multiple flags that are not 'none' or 'all'.</li> <li>The flag configurations must come with TCP mask configurations.</li> </ul>	

### 4.1.3.32 nv set/unset acl rule match ip tcp mss

	nv set acl <acl-id> rule <rule-id> match ip tcp mss <mss-format> nv unset acl <acl-id> rule <rule-id> match ip tcp mss <mss-format> Configure/remove ACL rule <rule-id> match ip tcp mss configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	mss-format	tcpmss value could be an integer or a range. Examples: "0-1", "536-65535", "65000", "128"
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp mss 536 admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp mss 536-65535</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/mss/<mss-format>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	The command will match TCP packets with the specified MSS values.	

### 4.1.3.33 nv set/unset acl rule match ip tcp all-mss-except

	nv set acl <acl-id> rule <rule-id> match ip tcp all-mss-except <mss-format> nv unset acl <acl-id> rule <rule-id> match ip tcp all-mss-except <mss-format> Configure/remove ACL rule <rule-id> match ip tcp all-mss-except configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	mss-format	tcpmss value could be an integer or a range. Examples: "0-1", "536-65535", "65000", "128"
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip tcp all-mss-except 536 admin@nvos:~\$ nv set acl custom-acl rule 660 match ip tcp all-mss-except 536-65535</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/tcp/all-mss-except/<mss-format>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	The command will match all TCP packets with MSS value different than the specified MSS values.	

### 4.1.3.34 nv set/unset acl rule match ip fragment

	nv set acl <acl-id> rule <rule-id> match ip fragment nv unset acl <acl-id> rule <rule-id> match ip fragment Configure/remove ACL rule <rule-id> match IP fragment configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip fragment</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/fragment	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	Match fragmented packets.	

### 4.1.3.35 nv show acl rule match ip ecn

	nv show acl <acl-id> rule <rule-id> match ip ecn Configure/remove ACL rule <rule-id> match IP ECN configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	

Example	<pre>admin@nvos:~\$nv show acl b rule 1 match ip ecn       operational  applied ----- ip-ect  3         3</pre>
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/ecn
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	

#### 4.1.3.36 nv set/unset acl rule match ip ecn

	nv set acl <acl-id> rule <rule-id> match ip ecn nv unset acl <acl-id> rule <rule-id> match ip ecn Configure/remove ACL rule <rule-id> match IP ECN configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset acl custom-acl rule 650 match ip ecn</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/ecn	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

#### 4.1.3.37 nv set/unset acl rule match ip ecn ip-ect

	nv set acl <acl-id> rule <rule-id> match ip ecn ip-ect <ip-ect-num> nv unset acl <acl-id> rule <rule-id> match ip ecn ip-ect <ip-ect-num> Configure/remove ACL rule <rule-id> match IP ECN ip-ect configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	ip-ect	ip-ect (integer: 0-3)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip ecn ip-ect 0</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/ecn/ip-ect	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

### 4.1.3.38 nv set/unset acl rule match ip ecn flags

	nv set acl <acl-id> rule <rule-id> match ip ecn flags <ecn-flag> nv unset acl <acl-id> rule <rule-id> match ip ecn flags <ecn-flag> Configure/remove ACL rule <rule-id> match IP ECN ip-ect configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	en-flag	enum: tcp-cwr   tcp-ece
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 650 match ip ecn flags tcp-cwr</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/ecn/flags/<flag-id>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

### 4.1.3.39 nv show acl rule match ip connection-state

	nv show acl <acl-id> rule <rule-id> match ip connection-state Show ACL rule <rule-id> match IP connection-state configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6 rule 10 match ip connection-state -o json {   "related": {},   "new": {} }</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/connection-state	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

### 4.1.3.40 nv set/unset acl rule match ip connection-state

	nv set acl <acl-id> rule <rule-id> match ip connection-state <state-id> nv unset acl <acl-id> rule <rule-id> match ip connection-state <state-id> Configure/remove ACL rule <rule-id> match IP connection-state <state-id> configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	state-id	state-id can be: established, invalid, new, related

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip connection-state new</pre>
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/ecn/ip-ect
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	Multiple connection-states can be configured.

#### 4.1.3.41 nv show acl rule match ip extension-header

	nv show acl <acl-id> rule <rule-id> match ip extension-header Show ACL rule <rule-id> match IP extension-header configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL1 rule 1 match ip extension-header operational          applied ----- type                 hop-by-hop          hop-by-hop</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/extension-header	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

#### 4.1.3.42 nv set/unset acl rule match ip extension-header type

	nv set acl <acl-id> rule <rule-id> match ip extension-header type <hop-by-hop> nv unset acl <acl-id> rule <rule-id> match ip extension-header type <hop-by-hop> Configure/remove ACL rule <rule-id> match IP extension-header configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip extension-header type hop-by-hop</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/extension-header/type/<type>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>This configuration is relevant to IPv6 ACLs only.</li> <li>Matches '-m hbh' in ip6tables tool</li> </ul>	

#### 4.1.3.43 nv show acl rule match ip routing-header

	nv show acl <acl-id> rule <rule-id> match ip routing-header Show ACL rule <rule-id> match ip routing-header configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT_IPV6 rule 850 match ip routing-header       operational  applied ----  - type  0           0</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/routing-header	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

#### 4.1.3.44 nv set/unset acl ACL rule match ip routing-header type

	nv set acl <acl-id> rule <rule-id> match ip routing-header type <hop-by-hop> nv unset acl <acl-id> rule <rule-id> match ip routing-header type <hop-by-hop> Configure/remove ACL rule <rule-id> match ip routing-header configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip extension-header type 0</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/routing-header/type/<type>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>• This configuration is relevant to IPv6 ACLs only.</li> <li>• Matches '-m rt' in ip6tables tool</li> </ul>	

#### 4.1.3.45 nv set/unset acl ACL rule match ip source-ip

	nv set acl <acl-id> rule <rule-id> match ip source-ip <ip-format> nv unset acl <acl-id> rule <rule-id> match ip source-ip <ip-format> Configure/remove ACL rule <rule-id> match ip source-ip configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	source-ip	(ANY   <ipv4>   <ipv6>   <ipv4-prefix>   <ipv6-prefix>   <ipv4-netmask>   <ipv6-netmask>)

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip source-ip 127.0.0.1/8</pre>
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/source-ip/<ip-format>
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	The user cannot configure IPv4 address on an ACL with IPv6 and vice versa.

#### 4.1.3.46 nv set/unset acl ACL rule match ip dest-ip

	nv set acl <acl-id> rule <rule-id> match ip dest-ip <ip-format> nv unset acl <acl-id> rule <rule-id> match ip dest-ip <ip-format> Configure/remove ACL rule <rule-id> match ip dest-ip configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	dest-ip	(ANY   <ipv4>   <ipv6>   <ipv4-prefix>   <ipv6-prefix>   <ipv4-netmask>   <ipv6-netmask>)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip dest-ip 127.0.0.1/8</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/dest-ip/<ip-format>	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	The user cannot configure IPv4 address on an ACL with IPv6 and vice versa.	

#### 4.1.3.47 nv set/unset acl rule match ip protocol

	nv set acl <acl-id> rule <rule-id> match ip protocol <protocol-format> nv unset acl <acl-id> rule <rule-id> match ip protocol <protocol-format> Configure/remove ACL rule <rule-id> match IP dest-ip configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	protocol-format	(0-255   tcp   udp   icmp   icmpv6)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip protocol tcp</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/protocol/<protocol-format>	

Related Commands	<code>nv set acl &lt;acl-id&gt; rule &lt;rule-id&gt;</code>
Notes	

#### 4.1.3.48 nv set/unset acl rule match ip icmp-type

	<code>nv set acl &lt;acl-id&gt; rule &lt;rule-id&gt; match ip icmp-type &lt;icmp-format&gt;</code> <code>nv unset acl &lt;acl-id&gt; rule &lt;rule-id&gt; match ip icmp-type &lt;icmp-format&gt;</code> Configure/remove ACL rule <rule-id> match IP ICMP-type configurations.	
Syntax Description	<code>acl-id</code>	ACL name
	<code>rule-id</code>	Rule number (integer: 1-65535)
	<code>icmp-format</code>	(0-255   echo-reply   echo-request   time-exceeded   dest-unreachable   port-unreachable)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip icmp-type echo-reply</pre> <pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip icmp-type 9</pre>	
REST API	PATCH/DELETE <a href="https://&lt;ip&gt;/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/icmp-type/&lt;icmp-format&gt;">https://&lt;ip&gt;/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/icmp-type/&lt;icmp-format&gt;</a>	
Related Commands	<code>nv set acl &lt;acl-id&gt; rule &lt;rule-id&gt;</code>	
Notes	<ul style="list-style-type: none"> <li>The protocol must be specified to be ICMP</li> </ul> <pre>\$nv set acl custom-acl rule 10 match ip protocol icmp</pre> <ul style="list-style-type: none"> <li>IPv4 type must be specified for the configured ACL.</li> </ul>	

#### 4.1.3.49 nv set/unset acl rule match ip icmpv6-type

	<code>nv set acl &lt;acl-id&gt; rule &lt;rule-id&gt; match ip icmpv6-type &lt;icmp-format&gt;</code> <code>nv unset acl &lt;acl-id&gt; rule &lt;rule-id&gt; match ip icmpv6-type &lt;icmp-format&gt;</code> Configure/remove ACL rule <rule-id> match IP ICMPv6-type configurations.	
Syntax Description	<code>acl-id</code>	ACL name
	<code>rule-id</code>	Rule number (integer: 1-65535)
	<code>icmpv6-format</code>	(0-255   router-solicitation   router-advertisement   neighbor-solicitation   neighbor-advertisement)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip icmp-type router-solicitation</pre> <pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip icmp-type 9</pre>	

REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/icmpv6-type/<icmpv6-format>
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	<ul style="list-style-type: none"> <li>The protocol must be specified to be ICMPv6.</li> </ul> <pre style="border: 1px solid black; padding: 5px;">\$nv set acl custom-acl rule 10 match ip protocol icmpv6</pre> <ul style="list-style-type: none"> <li>IPv6 type must be specified for the configured ACL.</li> </ul>

#### 4.1.3.50 nv show acl rule match ip recent-list

	nv show acl <acl-id> rule <rule-id> match ip recent-list Show ACL rule <rule-id> match ip recent-list configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT_IPV6 rule 600 match ip recent-list ----- name                UDP          UDP update-interval    60           60 hit-count           100          100 action              update       update</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/recent-list	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Matches the 'recent' iptables module.</li> <li>Used to filter IP address that passes a specific rate. In the above example, the rate is 100 packets per 60 seconds, if a source-ip sends more than this rate, the IP address will be blocked.</li> </ul>	

#### 4.1.3.51 nv set/unset acl rule match ip recent-list name

	nv set acl <acl-id> rule <rule-id> match ip recent-list name <generic-name> nv unset acl <acl-id> rule <rule-id> match ip recent-list name <generic-name> Configure/remove ACL rule <rule-id> match IP recent-list name configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip recent-list name "EXAMPLE"</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/recent-list	
Related Commands	nv set acl <acl-id> rule <rule-id>	

Notes	<ul style="list-style-type: none"> <li>In order to configure recent-list, the user needs to configure action and name for the recent-list <ul style="list-style-type: none"> <li>For action set: configure name and action=set</li> <li>For action update: configure name and action=update, hit-count and update-interval</li> </ul> </li> <li>There can be multiple recent-lists in the system and each is distinguished by a name.</li> <li>Refer to the documentation of 'recent' in 'iptables' for further information.</li> </ul>
-------	---

#### 4.1.3.52 nv set/unset acl rule match ip recent-list action

	nv set acl <acl-id> rule <rule-id> match ip recent-list action (set   update) nv unset acl <acl-id> rule <rule-id> match ip recent-list action (set   update) Configure/remove ACL rule <rule-id> match IP recent-list action configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip recent-list action update</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/recent-list	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>In order to configure recent-list, the user needs to configure action and name for the recent-list <ul style="list-style-type: none"> <li>For action set: configure name and action=set</li> <li>For action update: configure name and action=update, hit-count and update-interval</li> </ul> </li> <li>There can be multiple recent-lists in the system and each is distinguished by a name.</li> <li>Refer to the documentation of 'recent' in 'iptables' for further information.</li> </ul>	

#### 4.1.3.53 nv set/unset acl rule match ip recent-list hit-count

	nv set acl <acl-id> rule <rule-id> match ip recent-list hit-count (1-4294967295) nv unset acl <acl-id> rule <rule-id> match ip recent-list hit-count (1-4294967295) Configure/remove ACL rule <rule-id> match ip recent-list hit-count configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip recent-list hit-count 100</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/recent-list	
Related Commands	nv set acl <acl-id> rule <rule-id>	

Notes	<ul style="list-style-type: none"> <li>In order to configure recent-list, the user needs to configure action and name for the recent-list <ul style="list-style-type: none"> <li>For action set: configure name and action=set</li> <li>For action update: configure name and action=update, hit-count and update-interval</li> </ul> </li> <li>There can be multiple recent-lists in the system and each is distinguished by a name.</li> <li>Refer to the documentation of 'recent' in 'iptables' for further information.</li> </ul>
-------	---

#### 4.1.3.54 nv set/unset acl rule match ip recent-list update-interval

	nv set acl <acl-id> rule <rule-id> match ip recent-list update-interval (1-4294967295) nv unset acl <acl-id> rule <rule-id> match ip recent-list update-interval (1-4294967295) Configure/remove ACL rule <rule-id> match ip recent-list update-interval configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip recent-list update-interval 60</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/recent-list	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>In order to configure recent-list, the user needs to configure action and name for the recent-list <ul style="list-style-type: none"> <li>For action set: configure name and action=set</li> <li>For action update: configure name and action=update, hit-count and update-interval</li> </ul> </li> <li>There can be multiple recent-lists in the system and each is distinguished by a name.</li> <li>Refer to the documentation of 'recent' in 'iptables' for further information.</li> </ul>	

#### 4.1.3.55 nv show acl rule match ip hashlimit

	nv show acl <acl-id> rule <rule-id> match ip hashlimit Show ACL rule <rule-id> match ip hashlimit configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show acl ACL_MGMT_INBOUND_CP_DEFAULT_IPV6 rule 870 match ip hashlimit operational applied ----- name          LOGGING      LOGGING rate-above    1/min        1/min burst         5            5 source-mask   128          128 expire        4294967295   4294967295 mode          src-ip       src-ip</pre>	
REST API	GET https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/recent-list	
Related Commands	nv set acl <acl-id> rule <rule-id>	

Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>This is another way to filter IP addresses.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>
-------	--

#### 4.1.3.56 nv set/unset acl rule match ip hashlimit name

	nv set acl <acl-id> rule <rule-id> match ip hashlimit name <generic-name> nv unset acl <acl-id> rule <rule-id> match ip hashlimit name <generic-name> Configure/remove ACL rule <rule-id> match IP hashlimit name configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip hashlimit name "Limiter"</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/hashlimit	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>	

#### 4.1.3.57 nv set/unset acl rule match ip hashlimit rate-above

	nv [un]set acl <acl-id> rule <rule-id> match ip hashlimit rate-above <rate-format> Configure/remove ACL rule <rule-id> match IP hashlimit rate configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	rate-format	Rate limit, should be in the following format: integer/time-unit where time-unit is one of [second   min   hour]. The max supported rate is 1000000/second
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip hashlimit rate 2/min</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/hashlimit	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>	

### 4.1.3.58 nv set/unset acl rule match ip hashlimit burst

	nv [un]set acl <acl-id> rule <rule-id> match ip hashlimit burst <burst-int> Configure/remove ACL rule <rule-id> match IP hashlimit burst configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	burst-int	integer:1-4294967295
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip hashlimit burst 5</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/hashlimit	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>	

### 4.1.3.59 nv set/unset acl rule match ip hashlimit expire

	nv [un]set acl <acl-id> rule <rule-id> match ip hashlimit expire <expire-int> Configure/remove ACL rule <rule-id> match IP hashlimit expire configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	expire-int	integer:1-4294967295
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip hashlimit expire 3</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/hashlimit	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>	

### 4.1.3.60 nv set/unset acl rule match ip hashlimit mode

	nv set acl <acl-id> rule <rule-id> match ip hashlimit mode <mode> nv unset acl <acl-id> rule <rule-id> match ip hashlimit mode <mode> Configure/remove ACL rule <rule-id> match IP hashlimit mode configurations.	
--	---	--

Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	mode	(enum:src-ip, dst-ip   string)
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip hashlimit mode src-ip</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/hashlimit	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>	

#### 4.1.3.61 nv set/unset acl rule match ip hashlimit destination-mask

	nv set acl <acl-id> rule <rule-id> match ip hashlimit destination-mask <mask> nv unset acl <acl-id> rule <rule-id> match ip hashlimit destination-mask <mask> Configure/remove ACL rule <rule-id> match IP hashlimit destination-mask configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	mask	integer: for ipv4 the range is 0-32 and for ipv6 the range is 0-128
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip hashlimit destination-mask 32</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/hashlimit	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>	

#### 4.1.3.62 nv set/unset acl rule match ip hashlimit source-mask

	nv set acl <acl-id> rule <rule-id> match ip hashlimit source-mask <mask> nv unset acl <acl-id> rule <rule-id> match ip hashlimit source-mask <mask> Configure/remove ACL rule <rule-id> match ip hashlimit source-mask configurations.	
Syntax Description	acl-id	ACL name
	rule-id	Rule number (integer: 1-65535)
	mask	Integer: IPv4 range: 0-32 IPv6 range: 0-128
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set acl custom-acl rule 10 match ip hashlimit source-mask 32</pre>
REST API	PATCH/DELETE https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/match/ip/hashlimit
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	<ul style="list-style-type: none"> <li>Matches the 'hashlimit' iptables module.</li> <li>The following properties are required for configuration: name, rate-above, burst, expire, mode.</li> </ul>

#### 4.1.3.63 nv show interface acl

	nv show interface <iface-id> acl Display the acl bound to the interface.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show interface eth0 acl ACL Name                               Rule ID  In Packets  In Bytes  Out Packets  Out Bytes ----- ACL_MGMT_INBOUND_CP_DEFAULT            1        15620       2481722  10         0           0  20       14437       1118906  30         0           0 ...  740         0           0  750         0           0  760         28         2352  770         0           0  780         0           0 ACL_MGMT_INBOUND_CP_DEFAULT_IPV6       10         0           0  20         0           0  30         0           0 ...  800         0           0  810         0           0  820         0           0  830         0           0  840         0           0  850         0           0  860         0           0  870         0           0  880         0           0 ACL_MGMT_INBOUND_DEFAULT                10         5          296 ACL_MGMT_INBOUND_DEFAULT_IPV6           10         0           0 ACL_MGMT_OUTBOUND_CP_DEFAULT            10         0           0  20         0           9750       1885408 ACL_MGMT_OUTBOUND_CP_DEFAULT_IPV6       10         0           45         4680  20         0           160        13308</pre>	
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

#### 4.1.3.64 nv show interface acl id

	nv show interface <iface-id> acl <acl-id> Display the given acl-id bound to the interface.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT Statistics ===== Rule  In Packet  In Byte  Out Packet  Out Byte  Layer  Remark  Action  Summary -----  10              0         0           0         0      ip      deny   permit  20             9767      1902332</pre>
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	

#### 4.1.3.65 nv show interface acl statistics

	nv show interface <iface-id> acl <acl-id> statistics Display the given acl-id statistics bound to the interface.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT statistics Rule  In Packet  In Byte  Out Packet  Out Byte  Layer  Remark  Action  Summary -----  10              0         0           0         0      ip      deny   permit  20             9767      1902332</pre>	
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/statistics	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

#### 4.1.3.66 nv show interface acl statistics

	nv show interface <iface-id> acl <acl-id> statistics <rule-id> Display the given acl-id statistics bound to the interface.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show interface eth0 acl ACL_MGMT_INBOUND_DEFAULT statistics 10 operational applied ----- match  ip   protocol      tcp   tcp   all-mss-except 536-65535 action inbound  packet        5  byte          296</pre>	

REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/statistics/{rule-id}
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	

#### 4.1.3.67 nv show interface acl outbound

	nv show interface <iface-id> acl <acl-id> outbound Display the given acl-id bound to the interface in the outbound direction.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT outbound Statistics ===== No Data</pre>	
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/outbound	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	If an ACL is configured on one direction and not the other, it will be shown in the parent show (nv show interface <iface> acl <acl-id>) and not in the show of the direction it is not configured on.	

#### 4.1.3.68 nv show interface acl outbound control-plane

	nv show interface <iface-id> acl <acl-id> outbound control-plane Display the given acl-id bound to the interface in the outbound control-plane direction.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT outbound control-plane Statistics ===== Rule   In Packet  In Byte  Out Packet  Out Byte  Layer  Remark  Action  Summary -----  10                0         0           0         ip      deny    deny  20                9823      1908964      0         ip      permit  permit</pre>	
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/outbound/control-plane	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	If an ACL is configured on one direction and not the other, it will be shown in the parent show (nv show interface <iface> acl <acl-id>) and not in the show of the direction it is not configured on.	

### 4.1.3.69 nv show interface acl inbound

	nv show interface <iface-id> acl <acl-id> inbound Display the given acl-id bound to the interface in the inbound direction.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show interface eth0 acl ACL_MGMT_OUTBOUND_CP_DEFAULT inbound Statistics ===== Rule   In Packet  In Byte  Out Packet  Out Byte  Layer  Remark  Action  Summary ----- 10     5          296      0           0         ip      deny    match.ip.protocol: tcp except: 536-65535 match.ip.tcp.all-mss- </pre>	
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/inbound	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	If an ACL is configured on one direction and not the other, it will be shown in the parent show (nv show interface <iface> acl <acl-id>) and not in the show of the direction it is not configured on.	

### 4.1.3.70 nv show interface acl inbound control-plane

	nv show interface <iface-id> acl <acl-id> Display the given acl-id bound to the interface in the inbound direction.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show interface eth0 acl AAA inbound control-plane Statistics ===== Rule   In Packet  In Byte  Out Packet  Out Byte  Layer  Remark  Action  Summary ----- 10     5          296      0           0         ip      deny    match.ip.protocol: tcp except: 536-65535 match.ip.tcp.all-mss- </pre>	
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/inbound/control-plane	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	If an ACL is configured on one direction and not the other, it will be shown in the parent show (nv show interface <iface> acl <acl-id>) and not in the show of the direction it is not configured on.	

### 4.1.3.71 nv set/unset interface acl inbound

	nv set interface <iface-id> acl <acl-id> inbound nv unset interface <iface-id> acl <acl-id> inbound Configure/remove the binding of the given ACL on the specified interface.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
	acl-id	ACL name
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 acl ACL1 inbound</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/inbound	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	An ACL cannot be bound to inbound and inbound control-plane or cannot be bound to outbound and outbound control-plane on the same interface!	

### 4.1.3.72 nv set/unset interface acl inbound control-plane

	nv set interface <iface-id> acl <acl-id> inbound control-plane nv unset interface <iface-id> acl <acl-id> inbound control-plane Configure/remove the binding of the given ACL on the specified interface.	
Syntax Description	iface-id	interface could be one of 'eth0' or 'loopback'
	acl-id	ACL name
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 acl ACL1 inbound control-plane</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/inbound/control-plane	
	nv set acl <acl-id> rule <rule-id>	
Notes	An ACL cannot be bound to inbound and inbound control-plane or cannot be bound to outbound and outbound control-plane on the same interface!	

### 4.1.3.73 nv set/unset interface acl outbound control-plane

	nv set interface <iface-id> acl <acl-id> inbound control-plane nv unset interface <iface-id> acl <acl-id> inbound control-plane Configure/remove the binding of the given ACL on the specified interface.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
	acl-id	ACL name
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set <b>interface</b> eth0 acl ACL1 outbound control-plane</pre>
REST API	PATCH/DELETE https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/outbound/control-plane
Related Commands	nv set acl <acl-id> rule <rule-id>
Notes	An ACL cannot be bound to inbound and inbound control-plane or cannot be bound to outbound and outbound control-plane on the same interface!

#### 4.1.3.74 nv set/unset interface acl outbound

	nv set interface <iface-id> acl <acl-id> outbound nv unset interface <iface-id> acl <acl-id> outbound Configure/remove the binding of the given ACL on the specified interface.	
Syntax Description	iface-id	Interface could be one of 'eth0' or 'loopback'
	acl-id	ACL name
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set <b>interface</b> eth0 acl ACL1 outbound</pre>	
REST API	PATCH/DELETE https://<ip>/nvue_v1/interface/{interface-id}/acl/{acl-id}/outbound	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes	An ACL cannot be bound to inbound and inbound control-plane or cannot be bound to outbound and outbound control-plane on the same interface!	

#### 4.1.3.75 nv action clear acl counters

	nv action clear acl counters Clear the ACL counters in the show command.	
Syntax Description	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action clear acl counters</pre>	
REST API	POST https://<ip>/nvue_v1/acl	
Related Commands	nv set acl <acl-id> rule <rule-id>	
Notes		

### 4.1.3.76 nv set acl rule action set dscp

	nv set acl rule action set dscp Set DSCP value for packets.	
Syntax Description	acl-id	ACL ID to manipulate
	rule-id	Rule to configure dscp
	Dscp-value	It could be enum or an integer. Enums supported: <ul style="list-style-type: none"> <li>• af11</li> <li>• af12</li> <li>• af13</li> <li>• af21</li> <li>• af22</li> <li>• af23</li> <li>• af31</li> <li>• af32</li> <li>• af33</li> <li>• af41</li> <li>• af42</li> <li>• af43</li> <li>• cs1</li> <li>• cs2</li> <li>• cs3</li> <li>• cs4</li> <li>• cs5</li> <li>• cs6</li> <li>• cs7</li> <li>• be</li> <li>• ef</li> </ul> Or an integer in the range [0,63]
History	25.02.2141	
Example	<pre>admin@nvos:~\$ nv set acl ACL1 rule 10 action set dscp ef</pre>	
REST API	PATCH https://<ip>/nvue_v1/acl/{acl-id}/rule/{rule-id}/action/set	
Related Commands	nv show acl rule action	
Notes	Supported only for the management interface. Configurable only in inbound and outbound directions!	

## 4.2 Attestation

For more information, see the following sections:

- [SPDM](#)
- [TPM](#)

## 4.2.1 SPDM

SPDM (Security Protocol and Data Model) is a protocol that enables secure communication and attestation between system components. It ensures hardware integrity and authenticity through cryptographic measurements and certificates.

### 4.2.1.1 SPDM Commands

- [SPDM Commands](#)

### 4.2.1.2 SPDM Commands

- [4.2.1.2.1 nv show system security spdm](#)
- [4.2.1.2.2 nv show system security spdm certificates](#)
- [4.2.1.2.3 nv show system security spdm measurements](#)
- [4.2.1.2.4 nv action generate system security spdm](#)

#### 4.2.1.2.1 nv show system security spdm

	nv show system security spdm Show list of component integrity ERoT in the BMC, with their measurements if present and certificate-chains.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system security spdm Component      Id      HashingAlgorithm ----- ERoT_BMC_0     CertChain None ERoT_CPU_0     CertChain None ERoT_FPGA_0    CertChain None ERoT_NVSwitch_0 CertChain None ERoT_NVSwitch_1 CertChain None</pre> <pre>admin@nvos:~\$ nv show system security spdm ERoT_BMC_0 ----- operational ----- measurements   HashingAlgorithm None certificates   Id      CertChain</pre>	
REST API	GET https://<ip>/nvue_v1/system/security/spdm	
Related Commands	nv action generate system security spdm nv show system security spdm measurements nv show system security spdm certificates	

Notes	The output differs between JSON and regular "nv show" formats. The regular display will only show a list of ERoT and only if the relevant fields are present. In contrast, the JSON output is much longer, comprising five ERoT responses. In th output, when the name of the component is listed, the possible outcomes are as follows: (enum:ERoT_BMC_0, ERoT_CPU_0, ERoT_FPGA_0, ERoT_NVSwitch_0, ERoT_NVSwitch_1)
-------	---

#### 4.2.1.2.2 nv show system security spdm certificates

	nv show system security spdm certificates Shows certificate-chain of component integrity ERoT in the BMC.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show system security spdm ERoT_BMC_0 certificates -- operational -- ----- Id CertChain nv show sys sec spdm ERoT_CPU_0 certificates -ojson {   "CertificateString": "-----BEGIN CERTIFICATE-----\n..AX\n-----END CERTIFICATE-----\n",   "CertificateType": "PEMchain",   "CertificateUsageTypes": [     "Device"   ],   "Id": "CertChain",   "Name": "MGX_ERoT_CPU_0 Certificate Chain",   "SPDM": {     "SlotId": 0   } } </pre>
REST API	GET https://<ip>/nvue_v1/system/security/spdm/ERoT_BMC_0/ certificates
Related Commands	nv action generate system security spdm nv show system security spdm measurements nv show system security spdm
Notes	The output differs between JSON and regular "nv show" formats. The regular display will only show a list of ERoT and only if the relevant fields are present.

#### 4.2.1.2.3 nv show system security spdm measurements

	nv show system security spdm measurements Shows certificate-chain of component integrity ERoT in the BMC.
Syntax Description	N/A
Default	N/A
History	25.02.1884

Example	<pre> admin@nvos:~\$ nv show system security spdm ERoT_BMC_0 certificates operational ----- measurements   HashingAlgorithm  None  nv show sys sec spdm ERoT_CPU_0 measurements -ojson {   "HashingAlgorithm": "None",   "SignedMeasurements": "",   "SigningAlgorithm": "None",   "Version": "unknown" } </pre>
REST API	GET https://<ip>/nvue_v1/system/security/spdm/ERoT_BMC_0/ measurements
Related Commands	<pre> nv action generate system security spdm nv show system security spdm certificates nv show system security spdm </pre>
Notes	Result differs on Json and nv show format - regular show will show only list of ERoT and if fields are present

#### 4.2.1.2.4 nv action generate system security spdm

	nv action generate system security spdm Generate measurements on BMC.	
Syntax Description	nonce	64 char hex string
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv action generate system security spdm ERoT_BMC_0 Action executing ... Action succeeded </pre>	
REST API	POST https://<ip>/nvue_v1/system/security/spdm/{component-id}	
Related Commands	<pre> nv show system security spdm nv show system security spdm measurements nv show system security spdm certificates </pre>	
Notes		

## 4.2.2 TPM

TPM (Trusted Platform Module) is a hardware-based security technology that protects system integrity by securely storing cryptographic keys and measurements. It supports functionalities such as secure boot, attestation, and encryption.

### 4.2.2.1 TPM Commands

- [TPM Commands](#)

## 4.2.2.2 TPM Commands

- [4.2.2.2.1 nv action generate system security tpm](#)
- [4.2.2.2.2 nv action upload system security tpm](#)

### 4.2.2.2.1 nv action generate system security tpm

	nv action generate system security tpm <pcrs> <nonce> [algorithm] Generate quotes file.	
Syntax Description	pcrs	Platform Configuration Registers to be included in the quote <1-30>, divided by “,”. Both quote and PCRs use the same hash algorithm.
	nonce	Hex string, up to 512 bits (128 hex letters)
	algorithm	Hashing algorithm to be used (e.g., sha384)
Default	Algorithm-sha384	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action generate system security tpm 1,2 12 algorithm sha384</pre>	
REST API	POST https://<ip>/nvue_v1/system/security/tpm/quote	
Related Commands		
Notes		

### 4.2.2.2.2 nv action upload system security tpm

	nv action upload sys security tpm <file-name> <remote-url> Upload configuration file.	
Syntax Description	file-name	File to be uploaded (IAK.crt, quotes.json, or oIAK.crt). Note: quotes.json is a Base64-encoded JSON of quote.bin and signature, available after generating using nv action generate system security tpm.
	remote-url	ftp, scp and sftp are supported (e.g., scp://username[:password]@hostname/path/filename)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action upload sys security tpm IAK.crt scp://user:pass@host/path/IAK.crt</pre>	
REST API	POST https://<ip>/nvue_v1/system/security/tpm/upload	
Related commands		
Notes		

## 4.3 Authentication Authorization and Accounting

AAA (authentication, authorization, and accounting) supports configuring local accounts and remote servers using protocols like RADIUS, TACACS+, and LDAP. The AAA object model includes user management, general configurations, and per-protocol settings.

AAA configuration can be viewed on NVOS.

```
admin@nvos:~$ nv show system aaa
```

AAA authentication consists of authentication order and authentication failthrough.

### 4.3.1 Authentication Order

Authentication order specifies the sequence of protocols (radius, tacacs, ldap, local) used for authentication, separated by commas, for example:

```
admin@nvos:~$ nv set system aaa authentication order radius,local
# or
admin@nvos:~$ nv set system aaa authentication order local,ldap
```



Authentication order must include local and one of the following: radius, tacacs, or ldap.

### 4.3.2 Authentication Failthrough

Authentication failthrough defines the behavior of authentication when it is rejected locally or by an AAA server.

When authentication failthrough is disabled (default), the authentication process is blocked if the user password is rejected either locally or by the AAA server.

When authentication failthrough is enabled, the authentication process continues to the next AAA server or method if it is rejected.

```
admin@nvos:~$ nv set system aaa authentication failthrough ?
<arg>          Configure failthrough.
authentication errors.  "Enabled" login authentication continues to the next option on both server and
                        "Disabled" login authentication continues only on server errors.
                        (enum:enabled, disabled | string | default:disabled)
```



Authentication failthrough does not impact behavior when the AAA server is unavailable. After a server timeout, the switch will try the next server or method in order.

For more information on Authentication Authorization and Accounting, see the following sections:

- [User Accounts](#)
- [LDAP Authentication and Authorization](#)
- [TACACS](#)
- [RADIUS](#)

## 4.3.3 User Accounts

- [4.3.3.1 First Login](#)
- [4.3.3.2 Reset Local Users' Passwords](#)
- [4.3.3.3 User Account Commands](#)

By default, NVOS has two user accounts: *admin* and *monitor*.

The *admin* account:

- The default password of the account is "admin".
- The system administrator account. It is part of *sudo* group and it has sudo privileges.
- The account has permissions to run any show, set or action commands.

The *monitor* account:

- The default password of the account is "monitor".
- The system monitor account. It has read-only privileges.
- The account has permissions to run show commands only.

### 4.3.3.1 First Login

User will be required to change the default passwords for admin and monitor accounts upon first login. The new password must comply with the default password hardening rules (see [Password Hardening](#) section).

The new configured password is treated like any other configuration. If the configuration is not saved, the user will need to reconfigure it upon the system's next boot.

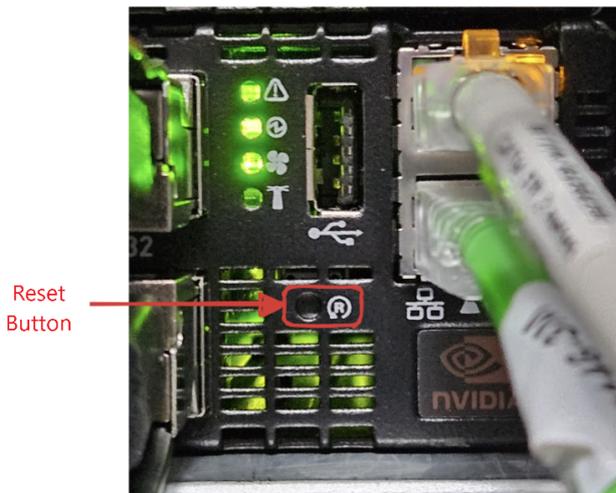
```
You are required to change your password immediately (administrator enforced).
```

```
WARNING: Your password has expired.  
You must change your password now!  
New password:  
Retype new password:
```

### 4.3.3.2 Reset Local Users' Passwords

If the passwords for the local user accounts on the switch are forgotten, a recovery method exists. Press and hold the Reset button for at least 15 seconds to reset the passwords for the 'admin' and 'monitor' accounts. This process also deletes any non-default users (i.e., users other than 'admin' and 'monitor'), resets the passwords for these default users, and expires them. The expiration forces a password change upon the first login.

Reset Button on the Physical Switch



**⚠** The image is for illustration purposes only. The button's location may vary depending on the switch model.

### 4.3.3.3 User Account Commands

- [User Account Commands](#)

### 4.3.3.4 User Account Commands

- [4.3.3.4.1 nv show system aaa user](#)
- [4.3.3.4.2 nv show system aaa user id](#)
- [4.3.3.4.3 nv show system aaa user ssh authorized-key](#)
- [4.3.3.4.4 nv show system aaa user ssh authorized-key id](#)
- [4.3.3.4.5 nv show system aaa user ssh](#)
- [4.3.3.4.6 nv set/unset system aaa user ssh authorized-key](#)
- [4.3.3.4.7 nv set/unset system aaa user](#)
- [4.3.3.4.8 nv set/unset system aaa user full-name](#)
- [4.3.3.4.9 nv set/unset system aaa user state](#)
- [4.3.3.4.10 nv set/unset system aaa user role](#)
- [4.3.3.4.11 nv set/unset system aaa user password](#)
- [4.3.3.4.12 nv set/unset system aaa user hashed-password](#)
- [4.3.3.4.13 nv set/unset system aaa allow-reset-local-passwords state](#)
- [4.3.3.4.14 nv show system aaa allow-reset-local-passwords](#)

#### 4.3.3.4.1 nv show system aaa user

	nv show system aaa user Displays list of users, their role and status.
Syntax Description	N/A
Default	N/A

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system aaa user Username  Full-name          Role    State ----- admin     System Administrator  admin  enabled monitor   System Monitor       monitor enabled</pre>
REST API	GET https://<ip>/nvue_v1/system/aaa/user
Related Commands	nv set system aaa user
Notes	

#### 4.3.3.4.2 nv show system aaa user id

	nv show system aaa user <user-id> Displays configuration of a user.	
Syntax Description	user-id	The user (e.g., monitor, test)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system aaa user admin ----- state      enabled          applied role       admin            admin full-name  System Administrator System Administrator password   *                * hashed-password *                *</pre>	
REST API	GET https://<ip>/nvue_v1/system/aaa/user/{user-id}	
Related Commands	nv show system aaa user nv set system aaa user	
Notes		

#### 4.3.3.4.3 nv show system aaa user ssh authorized-key

	nv show system aaa user <user-id> ssh authorized-key <authorized-key-id> Lists all SSH keys associated with the specified user.	
Syntax Description	user-id	The user (e.g., monitor, test)
	authorized-key-id	Name of the SSH key (item-name) (enum: saved keys of user)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show sys aaa user admin ssh authorized-key SSH Key Name      Key string  Key Type ----- key1               *           ssh-rsa key2               *           ssh-rsa</pre>	
REST API	GET https://<ip>/nvue_v1/system/aaa/user/<user>/ssh/authorized-key	

Related ommands	
Notes	The key string is obfuscated yet not regarded as a secret within NVOS.

#### 4.3.3.4.4 nv show system aaa user ssh authorized-key id

	nv show system aaa user <user-id> ssh authorized-key <ssh-authorized-key-id> Displays details for a specific SSH key.	
Syntax Description	user-id	The user (e.g., monitor, test)
	authorized-key-id	Name of the SSH key (item-name) (enum: saved keys of user)
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show sys aaa user admin ssh authorized-key key1       operational  applied ----- key    *            * type  ssh-rsa      ssh-rsa  admin@nvos:~\$ nv show sys aaa user admin ssh authorized-key key2       operational  applied ----- key type </pre>	
REST API	GET https://<ip>/nvue_v1/system/aaa/user/<user-id>/ssh/authorized-key/<ssh-authorized-key-id>	
Related Commands		
Notes	The key string is obfuscated yet not regarded as a secret within NVOS.	

#### 4.3.3.4.5 nv show system aaa user ssh

	nv show system aaa user <user-id> ssh Display user SSH configuration.	
Syntax Description	user-id	The user (e.g., monitor, test)
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show sys aaa user admin ssh       operational  applied ----- [authorized-key] key1      key1 </pre>	
REST API	GET https://<ip>/nvue_v1/system/aaa/user/<user-id>/ssh	
Related Commands		
Notes	Key string in obfuscated but is not considered a secret in nvos.	

#### 4.3.3.4.6 nv set/unset system aaa user ssh authorized-key

	nv set system aaa user <user-id> ssh authorized-key <ssh-authorized-key-id> {key   type} nv unset system aaa user <user> ssh authorized-key <ssh-authorized-key-id> {key   type} Authorized SSH key configuration. The unset form of the command clears configuration of SSH parameters for a user.	
Syntax Description	user-id	Name of the user (user-name) (enum: local users)
	authorized-key-id	Name of the SSH key (item-name) (enum: saved keys of user)
	key	The base64 contents of the key (key-string)
	type	The type of encoded key (string   enum:ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, ssh-ed25519, ssh-rsa   default:ssh-rsa)
Default	key=N/A type=ssh-rsa	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set sys aaa user admin ssh authorized-key key1 key AAAdnfs...== admin@nvos:~\$ nv set sys aaa user admin ssh authorized-key key1 type nistp384</pre> <pre>admin@nvos:~\$ nv unset sys aaa user admin ssh authorized-key admin@nvos:~\$ nv unset sys aaa user admin ssh authorized-key key1 admin@nvos:~\$ nv unset sys aaa user admin ssh authorized-key key1 type admin@nvos:~\$ nv unset sys aaa user admin ssh authorized-key key1 key</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/user/<user-id>/ssh/authorized-key/<ssh-authorized-key-id>	
Related Commands		
Notes		

#### 4.3.3.4.7 nv set/unset system aaa user

	nv set system aaa user <user-id> nv unset system aaa user <user-id> Specifies a username and creates a user account. New users are created initially with admin privileges. The unset form of the command deletes the user account.	
Syntax Description	user-id	The user. Username max length is 32 and it begins with a letter or an underscore, followed by letters, digits, underscores, or dashes. They can end with a dollar sign.
Default	The following usernames are available by default: * admin * monitor	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system aaa user test</pre>
REST API	PATCH https://<ip>/nvue_v1/system/aaa/user/{user-id}
Related Commands	nv show system aaa user nv set system aaa user password
Notes	<ul style="list-style-type: none"> <li>• New users must have a password.</li> <li>• Default users cannot be deleted.</li> </ul>

#### 4.3.3.4.8 nv set/unset system aaa user full-name

	nv set system aaa user <user-id> full-name <full-name> nv unset system aaa user <user-id> full-name <full-name> Configures user's full-name (Gecos Field). The unset form of the command sets user full-name (Gecos Field) to empty.	
Syntax Description	user-id	The user
	full-name	The full name of the user
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa user test full-name "Test User"</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/user/{user-id}	
Related Commands	nv show system aaa user nv set system aaa user	
Notes		

#### 4.3.3.4.9 nv set/unset system aaa user state

	nv set system aaa user <user-id> state <enable   disable> nv unset system aaa user <user-id> state Enables/disables the user account. The unset form of the command returns the user account state to its default state (enabled).	
Syntax Description	user-id	The user
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa user test state disabled</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/user/{user-id}	
Related Commands	nv show system aaa user nv set system aaa user	
Notes	Disabling a user account will terminate all user bash terminals.	

#### 4.3.3.4.10 nv set/unset system aaa user role

	nv set system aaa user <user-id> role <role-id> nv unset system aaa user <user-id> role Configures user role (capabilities). The unset form of the command return the user account role to its default (admin).	
Syntax Description	user-id	The user
	role-id	The name of the role
Default	admin	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa user test role monitor</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/user/{user-id}	
Related Commands	nv show system aaa user nv show system aaa role nv set system aaa user	
Notes		

#### 4.3.3.4.11 nv set/unset system aaa user password

	nv set system aaa user <user-id> password <password> nv unset system aaa user <user-id> password Configures a login password in cleartext. The unset form of the command clears the user password for non-default users. For default users, the default password will be expired and must be reconfigured in the next login.	
Syntax Description	user-id	The user
	password	A password for the user in string form. A string containing special Linux characters must be quoted or have the special characters escaped (i.e., add "\" before each special character). Examples: <pre>pass!word</pre> <pre>"pass!word"</pre> A leading dot is a special case and it must be escaped even if it is quoted: Examples: <pre>"\.password"</pre> <pre>\\.password</pre>
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa user test "pa!sswOrd"</pre> <pre>admin@nvos:~\$ nv set system aaa user test</pre> <pre>Enter new password:</pre> <pre>Confirm password:</pre>	

REST API	PATCH https://<ip>/nvue_v1/system/aaa/user/{user-id}
Related Commands	nv show system aaa user nv set system aaa user
Notes	<ul style="list-style-type: none"> <li>If no password was specified, user will be prompted to configure the password.</li> <li>If password hardening is enabled, the new password must match all configured policies.</li> <li>A string containing special Linux characters must be quoted or have the special characters escaped (i.e., add "\" before each special character).</li> </ul> <p>Examples:</p> <pre>pass!word "pass!word"</pre> <p>A leading dot is a special case and it must be escaped even if it is quoted:</p> <p>Examples:</p> <pre>".password" \\.password</pre> <ul style="list-style-type: none"> <li>A password is required, therefore a password must be configured before applying new configurations.</li> </ul>

#### 4.3.3.4.12 nv set/unset system aaa user hashed-password

	nv set system aaa user <user-id> hashed-password <hashed-password> nv unset system aaa user <user-id> hashed-password Configures a login password in encrypted format. The unset form of the command clears the user hashed-password.	
Syntax Description	user-id	The user
	hashed-password	A password for the user in encrypted text. Special Linux characters must be escaped (add "\" before each special character).
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa user test password "\$y\$j9T\$YwHwJEhi5c2oCgNNVJZgR0\ \$TboB1DcoS2iGmneLa/9Y54hsAgq9mi1QRGYmmkRffJC"</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/user/{user-id}	
Related Commands	nv show system aaa user nv set system aaa user nv set system aaa user password	
Notes	<ul style="list-style-type: none"> <li>If password hardening is enabled, hashed-password configuration will be blocked.</li> <li>Special Linux characters must be escaped (i.e., add "\" before each special character).</li> <li>A password is required, therefore a password must be configured before applying new configurations.</li> </ul>	

#### 4.3.3.4.13 nv set/unset system aaa allow-reset-local-passwords state

	nv set system aaa allow-reset-local-passwords state <enabled   disabled> nv unset system aaa allow-reset-local-passwords state Enables/disables the ability to reset local users' passwords upon long reboot press. The unset form of the command returns the state of feature to its default state (enabled).	
Syntax	state	enabled, disabled
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa allow-reset-local-passwords state disabled</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/allow-reset-local-passwords	
Related Commands	nv show system aaa allow-reset-local-passwords	
Notes		

#### 4.3.3.4.14 nv show system aaa allow-reset-local-passwords

	nv show system aaa allow-reset-local-passwords Displays state of resetting the local users passwords upon long reboot press	
Syntax	N/A	
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system allow-reset-local-passwords       operational  applied ----- state  enabled     enabled</pre>	
REST API	GET https://<ip>/nvue_v1/system/aaa/allow-reset-local-passwords	
Related Commands	nv set system allow-reset-local-passwords state <enabled disabled>	
Notes		

### 4.3.4 LDAP Authentication and Authorization

NVOS implements LDAP client AAA (Accounting, Authentication, and Authorization) in a transparent way with minimal configuration. There is no need to create accounts or directories on the switch.

NVOS uses Pluggable Authentication Modules (PAM) and Name Service Switch (NSS) for user authentication. NSS enables PAM to use LDAP to provide user authentication, group mapping, and information for other services on the system.

#### 4.3.4.1 Supported Features

- Authentication using PAM: Supports login, SSH, `sudo`, and `su`.

- Runs over the eth0 management interface.
- Supports up to eight LDAP servers.

#### 4.3.4.2 LDAP Configuration

LDAP configuration consists of two levels:

1. Global Configuration: Settings that apply to all LDAP servers unless overridden.
2. Per-Server Configuration: Specific settings for individual LDAP servers.

If a per-server configuration is not defined, the system will automatically use the settings from the global configuration.

All nv ldap commands are in [LDAP Commands](#) section. Global commands are under `/system/aaa/ldap`, and per-server commands are under `/system/aaa/ldap/hostname/<hostname-id>`.

#### 4.3.4.3 LDAP Groups and User Privileges

NVOS supports three types of users. User privileges are managed through the LDAP server by assigning users to specific LDAP groups. Membership in these groups determines the operations that a user is authorized to perform.

1. Admin privileged users (nv set, nv config apply): 1000(admin), 4(adm), 27(sudo), 999(docker), 1001(redis), 997(nvset), 996(nvapply)
2. Monitor privileged users (nv show): 4(adm), 998(nvshow)
3. Non-privileged users (no nv commands access)

##### 4.3.4.3.1 LDAP Server Group Configuration Example

Below is an example of configuring LDAP server groups. This configuration allows you to define a group of LDAP servers with common settings while enabling server-specific overrides when necessary.

```
dn: cn=nvset,ou=People,dc=itzgeek,dc=local
objectClass: posixGroup
cn: nvset
gidNumber: 997
memberUid: adminuser
```

#### 4.3.4.4 LDAP Secure Connection

The SSL section enables configuring the encryption mode for the LDAP client to ensure secure communication.

- Supported Encryption Modes: `start-tls`, `ssl`.
- Default CA Certificate Bundle: The LDAP client uses the default CA certificate bundle located at `/etc/ssl/certs/ca-certificates.crt`.
- Certificate Validation: Certificate validation may be skipped using SSL settings `cert-verify`. When certificate validation is skipped, the certificate is used only to establish a secure connection, without verifying its authenticity.

Ensure proper configuration to maintain secure and reliable LDAP connections.

### 4.3.4.5 LDAP Client Simple Configuration Example

Below is a simple example of configuring an LDAP client. This setup includes basic global settings and per-server configuration.

```
admin@nvos:~$ nv set system aaa ldap bind-dn <ldap-server-bind-dn>
admin@nvos:~$ nv set system aaa ldap secret "ldap-secret"
admin@nvos:~$ nv set system aaa ldap hostname <ldap-server-ip>
admin@nvos:~$ nv set system aaa ldap base-dn <ldap-server-base-dn>
# set global aaa configs
admin@nvos:~$ nv set system aaa authentication order ldap,local
admin@nvos:~$ nv config apply -y
```

### 4.3.4.6 LDAP Commands

- [LDAP Commands](#)

### 4.3.4.7 LDAP Commands

- [4.3.4.7.1 nv show system aaa ldap](#)
- [4.3.4.7.2 nv show system aaa ldap hostname](#)
- [4.3.4.7.3 nv set system aaa ldap hostname](#)
- [4.3.4.7.4 nv set system aaa ldap base-dn](#)
- [4.3.4.7.5 nv set system aaa ldap bind-dn](#)
- [4.3.4.7.6 nv set system aaa ldap port](#)
- [4.3.4.7.7 nv set system aaa ldap timeout-bind](#)
- [4.3.4.7.8 nv set system aaa ldap timeout-search](#)
- [4.3.4.7.9 nv set system aaa ldap secret](#)
- [4.3.4.7.10 nv set system aaa ldap map group cn](#)
- [4.3.4.7.11 nv set system aaa ldap map group gidnumber](#)
- [4.3.4.7.12 nv set system aaa ldap map group memberuid](#)
- [4.3.4.7.13 nv set/unset system aaa ldap map passwd gidnumber](#)
- [4.3.4.7.14 nv set system aaa ldap map passwd uid](#)
- [4.3.4.7.15 nv set system aaa ldap map passwd uidnumber](#)
- [4.3.4.7.16 nv set system aaa ldap map passwd userpassword](#)
- [4.3.4.7.17 nv set system aaa ldap version](#)
- [4.3.4.7.18 nv set system aaa ldap ssl mode](#)
- [4.3.4.7.19 nv set system aaa ldap ssl cert-verify](#)
- [4.3.4.7.20 nv set system aaa ldap ssl port](#)
- [4.3.4.7.21 nv set system aaa ldap ssl ca-list](#)

#### 4.3.4.7.1 nv show system aaa ldap

	nv show system aaa ldap Show LDAP configurations.
Syntax Description	N/A
Default	N/A
History	25.02.1884

Example	<pre> admin@nvos:~\$ nv show system aaa ldap operational                                applied ----- auth-port                                389 base-dn                                dc=itzgeek,dc=local bind-dn                                cn=ldapadm,dc=itzgeek,dc=local group-attribute                        member login-attribute                        cn password                                * timeout-bind                            5 timeout-search                          5 version                                  3 [hostname]                              10.209.1.250 </pre>
REST API	GET https://<ip>/nvue_v1/system/aaa/ldap
Related Commands	nv set system aaa ldap
Notes	LDAP feature in NVOS, the switch is basically an LDAP client that can be bind to an LDAP server, to support authentication to the switch via LDAP server instead local.

#### 4.3.4.7.2 nv show system aaa ldap hostname

	nv show system aaa ldap hostname Show remote LDAP servers.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show sys aa ldap hostname Hostname      Priority ----- 10.237.0.86  1 </pre>
REST API	GET https://<ip>/nvue_v1/system/aaa/ldap/hostname
Related Commands	nv set system aaa ldap hostname
Notes	Show LDAP configured servers.

#### 4.3.4.7.3 nv set system aaa ldap hostname

	nv set system aaa ldap hostname <hostname-id> Configure remote LDAP servers.
Syntax Description	hostname-id      LDAP server ID: ipv4, ipv4-unicas, idn-hostname, ipv6
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv set system aaa ldap hostname 1.2.3.4 </pre>
REST API	SET https://<ip>/nvue_v1/system/aaa/ldap/hostname/<hostname-id>
Related Commands	nv show system aaa ldap hostname <hostname-id> nv show system aaa ldap hostname nv show system aaa ldap
Notes	

#### 4.3.4.7.4 nv set system aaa ldap base-dn

	nv set system aaa ldap base-dn <base-dn> This command set the base-dn of the LDAP server.	
Syntax Description	base-dn	Configure base DN (Distinguished Name)
Default	ou=users dc=example dc=com	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap base-dn "dc=itzgeek,dc=local"</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/base-dn	
Related Commands	nv show system aaa ldap	
Notes	A base dn is the point from where a server will search for users.	

#### 4.3.4.7.5 nv set system aaa ldap bind-dn

	nv set system aaa ldap bind-dn <bind dn> This command sets the bind-dn of the ldap server.	
Syntax Description	bind dn	Configure bind DN (Distinguished Name)
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap bind-dn "cn=ldapadm,dc=itzgeek,dc=local"</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/bind-dn	
Related Commands	nv show system aaa ldap	
Notes	The Bind DN is the username that will be used to do the searching and request the authentication.	

#### 4.3.4.7.6 nv set system aaa ldap port

	nv set system aaa ldap port <1-65535> Set LDAP authentication port.	
Syntax Description	port	Integer: 1-65535
Default	389	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap port 389</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/ldap/port	
Related Commands	nv show system aaa ldap	

Notes	
-------	--

#### 4.3.4.7.7 nv set system aaa ldap timeout-bind

	nv set system aaa ldap timeout-bind <seconds> Set global LDAP max wait until bind timeout (seconds).	
Syntax Description	Seconds	Number of seconds
Default	5	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap timeout-bind 5</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/timeout-bind	
Related Commands	nv show system aaa ldap	
Notes		

#### 4.3.4.7.8 nv set system aaa ldap timeout-search

	nv set system aaa ldap timeout-search <seconds> Set global LDAP max wait until search timeout (seconds).	
Syntax Description	Seconds	Number of seconds
Default	cn	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap timeout-search 5</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/timeout-search	
Related Commands	nv show system aaa ldap	
Notes		

#### 4.3.4.7.9 nv set system aaa ldap secret

	nv set system aaa ldap secret <secret-value> Set global LDAP server secret in cleartext.	
Syntax Description	secret value	Secret string
Default	3	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap password 123asd</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/secret	
Related Commands	nv show system aaa ldap	

Notes	
-------	--

#### 4.3.4.7.10 nv set system aaa ldap map group cn

	nv set system aaa ldap map group cn <cn-str> Set LDAP search map for cn attribute for group database.	
Syntax Description	cn-str	Common name (string)
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap map group cn itzgeek</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/map/group/cn	
Related Commands	nv show system aaa ldap group	
Notes		

#### 4.3.4.7.11 nv set system aaa ldap map group gidnumber

	nv set system aaa ldap map group gidnumber <gidnumber> Set LDAP search map for gidNumber attribute for group database.	
Syntax Description	gidnumber	gidNumber string
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap map group gidNumber 1000</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/map/group/gidnumber	
Related Commands	nv show system aaa ldap group	
Notes		

#### 4.3.4.7.12 nv set system aaa ldap map group memberuid

	nv set system aaa ldap map group memberuid <memberuid> Set LDAP search map for memberUid attribute for group database.	
Syntax Description	memberuid	memberUid string
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap map group memberuid admingroup</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/map/group/memberuid	
Related Commands	nv show system aaa ldap group	

Notes	
-------	--

#### 4.3.4.7.13 nv set/unset system aaa ldap map passwd gidnumber

	nv set system aaa ldap map passwd gidnumber <gidnumber> Set LDAP map for gidNumber attribute for passwd database.	
Syntax Description	gidnumber	gidNumber string
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap map group gidnumber 1000</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/map/group/gidNumber	
Related Commands	nv show system aaa ldap passwd	
Notes		

#### 4.3.4.7.14 nv set system aaa ldap map passwd uid

	nv set system aaa ldap map group uid <uid> Set LDAP map for UID attribute for passwd database.	
Syntax Description	uid	uid string
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap map group uid 1000</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/map/group/uid	
Related Commands	nv show system aaa ldap passwd	
Notes		

#### 4.3.4.7.15 nv set system aaa ldap map passwd uidnumber

	nv set system aaa ldap map group uidnumber <uidnumber> Set LDAP map for uidNumber attribute for passwd database.	
Syntax Description	uidnumber	uidNumber string
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap map group uidnumber 1000</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/map/group/uidNumber	
Related Commands	nv show system aaa ldap passwd	

Notes	
-------	--

#### 4.3.4.7.16 nv set system aaa ldap map passwd userpassword

	nv set system aaa ldap map group userpassword <userpassword> Set LDAP map for userPassword attribute for passwd database.	
Syntax Description	userpassword	userpassword string
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap map group userpassword password</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/map/group/userpassword	
Related Commands	nv show system aaa ldap passwd	
Notes		

#### 4.3.4.7.17 nv set system aaa ldap version

	nv set system aaa ldap version <ldap-version> Set LDAP protocol version to be used.	
Syntax Description	ldap-version	2 or 3
Default	3	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa ldap version 2</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap	
Related Commands	nv show system aaa ldap	
Notes		

#### 4.3.4.7.18 nv set system aaa ldap ssl mode

	nv set system aaa ldap ssl mode <ssl-mode> Set the password of the LDAP server.	
Syntax Description	ssl-mode	none, ssl, start-tls
Default	None	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set sys aaa ldap ssl mode start-tls</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/ssl/mode	
Related Commands	nv show system aaa ldap	

Notes	
-------	--

#### 4.3.4.7.19 nv set system aaa ldap ssl cert-verify

	nv set system aaa ldap ssl cert-verify <enable   disable> Set CA certificate validation state.	
Syntax Description	enable	Validates certificate
	disable	Skips certificate validation
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set sys aaa ldap ssl mode cert-verify</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/ssl/cert-verify	
Related Commands	nv show system aaa ldap	
Notes		

#### 4.3.4.7.20 nv set system aaa ldap ssl port

	nv set system aaa ldap ssl port <1-65535> Set LDAP's authentication port.	
Syntax Description	port	Integer: 1-65535
Default	636	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set sys aaa ldap ssl port 636</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/ssl/port	
Related Commands	nv show system aaa ldap	
Notes		

#### 4.3.4.7.21 nv set system aaa ldap ssl ca-list

	nv set system aaa ldap ssl ca-list <none   default> Set LDAP CA certificate list.	
Syntax Description	none	LDAP does not use CA certificate
	default	LDAP uses the CA certificates in the following path: <code>/etc/ssl/certs/ca-certificates.crt</code>
Default	Default	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set sys aaa ldap ssl ca-list default</pre>
REST API	PATCH https://<ip>/nvue_v1/system/aaa/ldap/ssl/ca-list
Related Commands	nv show system aaa ldap
Notes	

## 4.3.5 TACACS

- [4.3.5.1 Supported Features](#)
- [4.3.5.2 TACACS Users](#)
  - [4.3.5.2.1 TACACS Server Setup and Usage Example](#)
- [4.3.5.3 TACACS+ Accounting](#)
  - [4.3.5.3.1 TACACS Accounting Configuration](#)
- [4.3.5.4 TACACS Commands](#)

NVOS implements TACACS+ client AAA (Accounting, Authentication, and Authorization) in a transparent way with minimal configuration. The client implements the TACACS+ protocol as described in this IETF document. There is no need to create accounts or directories on the switch. Accounting records go to all configured TACACS+ servers by default. Using per-command authorization requires additional setup on the switch.

### 4.3.5.1 Supported Features

- Authentication using PAM: includes `login`, `ssh`, `sudo` and `su`
- Runs over the eth0 management interface
- Up to eight TACACS+ servers

TACACS configuration is made of global configurations and per-server configurations. In general, if per-server configuration is not defined, the configuration will be taken from the global configuration.

All nv tacacs commands can be found in TACACS Commands, where global ones are direct under `/system/aaa/tacacs` and per-server ones or under `/system/aaa/tacacs/hostname/<hostname-id>`.

### 4.3.5.2 TACACS Users

NVOS supports three types of RADIUS users defined by priv-lvl configured in TACACS server.

- `priv-lvl=15` # admin privileged users (nv set, nv config apply)
- `priv-lvl=7` # monitor privileged users (nv show)
- `priv-lvl=1` # non-privileged users (no nv commands access)

### 4.3.5.2.1 TACACS Server Setup and Usage Example

TACACS server can be configured either on a remote host or on the switch itself (for testing or sanity-check).

Basic configuration for users and clients can be done in `/etc/tacplus_nss.conf` file.

#### Users Configuration

```
user = username {
    login = cleartext "login_password"
    pap = cleartext "pap_password"
    service = exec {
        priv-lvl=<15,7,1>
    }
}
```

#### Client Configuration

Client configuration allows specific client IPs and CIDR blocks.

```
key = "client-secret"
and:
acl = default {
    #permit = 192\.168\.0\.
    permit = 10\.7\.140\.30
    permit = .*
}
```

After configuring a tacacs server, configure the client:

```
admin@nvos:~$ nv set system aaa tacacs hostname <tacacs-server-ip> secret tacacs-secret
admin@nvos:~$ nv set system aaa authentication order tacacs,local
admin@nvos:~$ nv config apply -y
```

### 4.3.5.3 TACACS+ Accounting

#### 4.3.5.3.1 TACACS Accounting Configuration

TACACS accounting logs user activity and commands executed on the system, providing an audit trail for security and compliance. It ensures accountability by sending these logs to configured TACACS+ servers. The logs will be sent to the first server to respond.

TACACS accounting is managed under the `/etc/tacplus_nss.conf` file.

After configuring a TACACS server and client, enable accounting with the command `nv set system aaa tacacs accounting state enabled`.

#### 4.3.5.4 TACACS Commands

- [TACACS Commands](#)

## 4.3.5.5 TACACS Commands

- [4.3.5.5.1 nv show system aaa tacacs accounting](#)
- [4.3.5.5.2 nv show system aaa tacacs hostname](#)
- [4.3.5.5.3 nv set system aaa tacacs accounting](#)
- [4.3.5.5.4 nv set system aaa tacacs hostname](#)
- [4.3.5.5.5 nv set system aaa tacacs port](#)
- [4.3.5.5.6 nv set system aaa tacacs auth-type](#)
- [4.3.5.5.7 nv set system aaa tacacs secret](#)
- [4.3.5.5.8 nv set system aaa tacacs timeout](#)
- [4.3.5.5.9 nv set system aaa tacacs hostname auth-port](#)
- [4.3.5.5.10 nv set system aaa tacacs hostname auth-type](#)
- [4.3.5.5.11 nv set system aaa tacacs hostname secret](#)
- [4.3.5.5.12 nv set system aaa tacacs hostname priority](#)
- [4.3.5.5.13 nv set system aaa tacacs hostname timeout](#)

### 4.3.5.5.1 nv show system aaa tacacs accounting

	nv show system aaa tacacs accounting Show TACACS accounting configuration.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system aaa tacacs accounting ----- state  disabled      disabled ----- operational  applied</pre>
REST API	GET https://<ip>/nvue_v1/system/aaa/tacacs/accounting
Related Commands	nv set system aaa tacacs accounting state
Notes	Enable/disable tacacs accounting feature

### 4.3.5.5.2 nv show system aaa tacacs hostname

	nv show system aaa tacacs hostname Show remote TACACS servers.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system aaa tacacs hostname ----- Hostname  Auth type  Port  Priority  Secret  Timeout ----- 10.2.30.40  pap        49    1         *       5</pre>
REST API	GET https://<ip>/nvue_v1/system/aaa/tacacs/hostname

Related Commands	nv set system aaa tacacs hostname
Notes	Show TACACS configured servers.

#### 4.3.5.5.3 nv set system aaa tacacs accounting

	nv set system aaa tacacs accounting state <enabled   disabled> Configure remote TACACS servers.	
Syntax Description	state	enum: enabled   disabled
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs accounting state enabled</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/accounting/state/<enabled/disabled>	
Related Commands	nv show system aaa tacacs accounting nv show system aaa tacacs	
Notes		

#### 4.3.5.5.4 nv set system aaa tacacs hostname

	nv set system aaa tacacs hostname <hostname-id> Configure remote TACACS servers.	
Syntax Description	hostname-id	TACACS server ID: ipv4, ipv4-unicas, idn-hostname, ipv6
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs hostname 1.2.3.4</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/hostname/<hostname-id>	
Related Commands	nv show system aaa tacacs hostname <hostname-id> nv show system aaa tacacs hostname nv show system aaa tacacs	
Notes		

#### 4.3.5.5.5 nv set system aaa tacacs port

	nv set system aaa tacacs port <1-65535> Configure global TACACS authentication port.	
Syntax Description	port	Integer: 1-65535
Default	49	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system aaa tacacs port 51</pre>
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/port
Related Commands	nv show system aaa tacacs
Notes	

#### 4.3.5.5.6 nv set system aaa tacacs auth-type

	nv set system aaa tacacs auth-type <pap   chap   login> Configure global authentication type	
Syntax Description	auth-type	enum: chap, pap, login
Default	pap	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs auth-type chap</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/auth-type	
Related Commands	nv show system aaa tacacs	
Notes		

#### 4.3.5.5.7 nv set system aaa tacacs secret

	nv set system aaa tacacs secret <string   prompt> Configure global TACACS secret in cleartext.	
Syntax Description	secret	string   prompt
Default	""	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs secret tacacs-secret</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/secret	
Related Commands	nv show system aaa tacacs	
Notes		

#### 4.3.5.5.8 nv set system aaa tacacs timeout

	nv set system aaa tacacs timeout <1-60> Configure the global tacacs reply timeout (seconds)	
Syntax Description	timeout	Integer: 1-60
Default	3	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system aaa tacacs timeout 5</pre>
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/timeout
Related Commands	nv show system aaa tacacs
Notes	

#### 4.3.5.5.9 nv set system aaa tacacs hostname auth-port

	nv set system aaa tacacs hostname <hostname-id> port <1-65535> Configure tacacs authentication port for server	
Syntax Description	hostname-id	TACACS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	auth-port	Integer: 1-65535
Default	None (set by TACACS global configuration)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs hostname 1.2.3.4 port 50</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/hostname/<hostname-id>/port/<port>	
Related Commands	nv show system aaa tacacs hostname <hostname-id> nv show system aaa tacacs hostname nv show system aaa tacacs	
Notes		

#### 4.3.5.5.10 nv set system aaa tacacs hostname auth-type

	nv set system aaa tacacs hostname <hostname-id> auth-type <pap   chap   login> Configure TACACS authentication type for server.	
Syntax Description	hostname-id	TACACS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	auth-type	Enum: pap, chap, login
Default	None (set by tacacs global config)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs hostname 1.2.3.4 auth-type chap</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/hostname/<hostname-id>/auth-type/<auth-type>	
Related Commands	nv show system aaa tacacs hostname <hostname-id> nv show system aaa tacacs hostname nv show system aaa tacacs	
Notes		

#### 4.3.5.5.11 nv set system aaa tacacs hostname secret

	nv set system aaa tacacs hostname <hostname-id> secret <string   prompt> Configure server secret in cleartext.	
Syntax Description	hostname-id	TACACS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	secret	string   prompt
Default	None (set by tacacs global config)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs hostname 1.2.3.4 tacacs Tacacs-Server-Password</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/hostname/<hostname-id>/password/<SECRET>	
Related Commands	nv show system aaa tacacs hostname <hostname-id> nv show system aaa tacacs hostname nv show system aaa tacacs	
Notes		

#### 4.3.5.5.12 nv set system aaa tacacs hostname priority

	nv set system aaa tacacs hostname <hostname-id> priority <1-8> Configure tacacs priority for server.	
Syntax Description	hostname-id	Tacacs server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	priority	Integer: 1-8
Default	1	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs hostname 1.2.3.4 priority 5</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/hostname/<hostname-id>/priority/<priority-value>	
Related Commands	nv show system aaa tacacs hostname <hostname-id> nv show system aaa tacacs hostname nv show system aaa tacacs	
Notes		

#### 4.3.5.5.13 nv set system aaa tacacs hostname timeout

	nv set system aaa tacacs hostname <hostname-id> timeout <1-60> Configure the reply timeout for a TACACS server (seconds).	
Syntax Description	hostname-id	TACACS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	timeout	Integer: 1-60
Default	None (set by TACACS global configuration)	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set system aaa tacacs hostname 1.2.3.4 timeout 5</pre>
REST API	SET https://<ip>/nvue_v1/system/aaa/tacacs/hostname/<hostname-id>/timeout/<timeout-value>
Related Commands	<pre>nv show system aaa tacacs hostname &lt;hostname-id&gt; nv show system aaa tacacs hostname nv show system aaa tacacs</pre>
Notes	

## 4.3.6 RADIUS

Various add-on packages enable **RADIUS** users to log in to NVOS switches in a transparent way with minimal configuration. There is no need to create accounts or directories on the switch.

Authentication uses PAM and includes login, `ssh`, `restapi`, `sudo` and `su`.

### 4.3.6.1 RADIUS Client

RADIUS configuration is made of global configurations and per-server configurations. In general, if per-server configuration is not defined, the configuration will be taken from the global configuration.

All `nv radius` commands can be found in [.RADIUS Commands](#), where global ones are direct under `/system/aaa/radius` and per-server ones are under `/system/aaa/radius/hostname/<hostname-id>`

### 4.3.6.2 Radius Users

NVOS supports 3 types of RADIUS users, defined by Management-Privilege-Level configured in `radius-server`.

- Management-Privilege-Level := 15 # admin privileged users (`nv set`, `nv config apply`)
- Management-Privilege-Level := 7 # monitor privileged users (`nv show`)
- Management-Privilege-Level := 1 # non-privileged users (no `nv` commands access)

### 4.3.6.3 RADIUS Server Setup and Usage Example

Radius server can be configured either on a remote host, or on the switch itself (for testing or sanity-check).

#### 4.3.6.3.1 Basic RADIUS Server Configuration

To conduct a basic RADIUS server configuration, add sections to "users" and "clients.conf" files.

User File Example

```
radius_user Cleartext-Password := "radius_user_password"
Management-Privilege-Level := <15,7,1>
```

## Client File Example

```
client client_name {
    ipaddr      = 10.1.2.3
    secret      = radius-secret
}
# Or as CIDR block such as:
client 10.0.0.0/8 {
    secret      = testing-radius
}
```

### 4.3.6.3.2 How To Set Up Basic FreeRADIUS Server

1. Run the following command in a Debian machine or other similar Linux distributions.

```
sudo apt-get update
sudo apt-get install freeradius -y
```

2. Add your client IP to `/etc/freeradius/3.0/clients.conf` file as:

```
client client_name {
    ipaddr      = <CLIENT_IP>
    secret      = mysecret
}
```

or use CIDR block:

```
client 10.0.0.0/8 {
    secret      = global-secret
}
```

3. Add your required radius users to `/etc/freeradius/3.0/users` file as:

```
radius_admin_user Cleartext-Password := "radius_password"
                    Management-Privilege-Level := 15
radius_monitor_user Cleartext-Password := "radius_password"
                    Management-Privilege-Level := 7
radius_non_priv_user Cleartext-Password := "radius_password"
                    Management-Privilege-Level := 1
```

4. Reboot freeRADIUS service (and make sure it is running).

```
sudo service freeradius restart
sudo service freeradius status
```

5. Configure RADIUS client to use such server.

```
admin@nvos:~$ nv set system aaa radius hostname <radius-server-ip> secret radius-secret
admin@nvos:~$ nv set system aaa authentication order radius,local
admin@nvos:~$ nv config apply -y
```

6. Login with configured users.

### 4.3.6.4 RADIUS Commands

- [RADIUS Commands](#)

## 4.3.6.5 RADIUS Commands

- [4.3.6.5.1 nv show system aaa radius](#)
- [4.3.6.5.2 nv show system aaa radius hostname](#)
- [4.3.6.5.3 nv show system aaa radius hostname](#)
- [4.3.6.5.4 nv set system aaa radius hostname](#)
- [4.3.6.5.5 nv set system aaa radius auth-port](#)
- [4.3.6.5.6 nv set system aaa radius auth-type](#)
- [4.3.6.5.7 nv set system aaa radius retransmit](#)
- [4.3.6.5.8 nv set system aaa radius password](#)
- [4.3.6.5.9 nv set system aaa radius statistics](#)
- [4.3.6.5.10 nv set system aaa radius timeout](#)
- [4.3.6.5.11 nv set system aaa radius hostname auth-port](#)
- [4.3.6.5.12 nv set system aaa radius hostname auth-type](#)
- [4.3.6.5.13 nv set system aaa radius hostname password](#)
- [4.3.6.5.14 nv set system aaa radius hostname priority](#)
- [4.3.6.5.15 nv set system aaa radius hostname retransmit](#)
- [4.3.6.5.16 nv set system aaa radius hostname timeout](#)

### 4.3.6.5.1 nv show system aaa radius

	nv show system aaa radius Display RADIUS features configuration and state.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show system aaa radius               operational  applied ----- auth-port    1812          1812 auth-type    pap             pap password     *                   * retransmit   0                   0 statistics   disabled            disabled timeout      3                   3 [hostname]   1.1.1.1 [hostname]   1.2.3.4 [hostname]   10.7.34.20  10.7.34.20 </pre>
REST API	GET https://<ip>/nvue_v1/system/aaa/radius
Related Commands	
Notes	

### 4.3.6.5.2 nv show system aaa radius hostname

	nv show system aaa radius hostname Show remote RADIUS servers.
Syntax Description	N/A
Default	N/A

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system aaa radius hostname ----- Hostname      Auth port  Auth type  Password  Priority  Retransmit  Timeout ----- 1.1.1.1       1812      pap        *         5         0           3 10.7.34.20    1812      pap        *         1         0           3</pre>
REST API	GET https://<ip>/nvue_v1/system/aaa/radius/hostname
Related Commands	nv show system aaa radius
Notes	

#### 4.3.6.5.3 nv show system aaa radius hostname

	nv show system aaa radius hostname <hostname-id> Display RADIUS server configuration.	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicas, idn-hostname, ipv6
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system aaa radius hostname 10.7.34.20 ----- operational  applied ----- auth-port    1812 auth-type    pap password     * priority     1           1 retransmit   0 timeout      3</pre>	
REST API	GET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>	
Related Commands	nv show system aaa radius hostname nv show system aaa radius	
Notes		

#### 4.3.6.5.4 nv set system aaa radius hostname

	nv set system aaa radius hostname <hostname-id> Configure remote RADIUS servers.	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicas, idn-hostname, ipv6
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius hostname 1.2.3.4</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>	
Related Commands	nv show system aaa radius hostname <hostname-id> nv show system aaa radius hostname nv show system aaa radius	
Notes		

#### 4.3.6.5.5 nv set system aaa radius auth-port

	nv set system aaa radius auth-port <1-65535> Configure global RADIUS authentication port.	
Syntax Description	auth-port	Integer: 1-65535
Default	1812	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius auth-port 1813</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/auth-port	
Related Commands	nv show system aaa radius	
Notes		

#### 4.3.6.5.6 nv set system aaa radius auth-type

	nv set system aaa radius auth-type <pap   chap> Configure global authentication type.	
Syntax Description	auth-type	enum: chap, pap
Default	pap	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius auth-type chap</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/auth-type	
Related Commands	nv show system aaa radius	
Notes		

#### 4.3.6.5.7 nv set system aaa radius retransmit

	nv set system aaa radius retransmit <0-10> Configure the default RADIUS retransmit tries.	
Syntax Description	retransmit	integer: 0-10
Default	0	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius retransmit 2</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/retransmit	
Related Commands	nv show system aaa radius	
Notes		

#### 4.3.6.5.8 nv set system aaa radius password

	nv set system aaa radius password <string   prompt> Configure global radius server password in cleartext.	
Syntax Description	password	string, prompt
Default	""	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius password Radius-Password</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/password	
Related Commands	nv show system aaa radius	
Notes		

#### 4.3.6.5.9 nv set system aaa radius statistics

	nv set system aaa radius statistics <enabled   disabled> Enable/disable log RADIUS statistics.	
Syntax Description	statistics	enum: enabled, disabled
Default	Disabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius statistics enabled</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/statistics	
Related Commands	nv show system aaa radius	
Notes		

#### 4.3.6.5.10 nv set system aaa radius timeout

	nv set system aaa radius timeout <1-60> Configure the global RADIUS server reply timeout (seconds).	
Syntax Description	timeout	Integer: 1-60
Default	3	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius timeout 5</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/timeout	
Related Commands	nv show system aaa radius	
Notes		

#### 4.3.6.5.11 nv set system aaa radius hostname auth-port

	nv set system aaa radius hostname <hostname-id> auth-port <1-65535> Configure RADIUS authentication port for server.	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	auth-port	Integer: 1-65535
Default	None (set by RADIUS global config)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius hostname 1.2.3.4 auth-port 2812</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>/auth-port/<auth-port>	
Related Commands	nv show system aaa radius hostname <hostname-id> nv show system aaa radius hostname nv show system aaa radius	
Notes		

#### 4.3.6.5.12 nv set system aaa radius hostname auth-type

	nv set system aaa radius hostname <hostname-id> auth-type <pap   chap> Configure RADIUS authentication type for server.	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	auth-type	Enum: pap, chap
Default	None (set by RADIUS global config)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius hostname 1.2.3.4 auth-type chap</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>/auth-type/<auth-type>	
Related Commands	nv show system aaa radius hostname <hostname-id> nv show system aaa radius hostname nv show system aaa radius	
Notes		

#### 4.3.6.5.13 nv set system aaa radius hostname password

	nv set system aaa radius hostname <hostname-id> password <password> Configure server password in cleartext.	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	password	string, prompt
Default	None (set by RADIUS global config)	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set system aaa radius hostname 1.2.3.4 password Radius-Server-Password</pre>
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>/password/<PASSWORD>
Related Commands	nv show system aaa radius hostname <hostname-id> nv show system aaa radius hostname nv show system aaa radius
Notes	

#### 4.3.6.5.14 nv set system aaa radius hostname priority

	nv set system aaa radius hostname <hostname-id> priority <1-8> Configure RADIUS priority for server	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	priority	Integer: 1-8
Default	1	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius hostname 1.2.3.4 priority 5</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>/priority/<priority-value>	
Related Commands	nv show system aaa radius hostname <hostname-id> nv show system aaa radius hostname nv show system aaa radius	
Notes		

#### 4.3.6.5.15 nv set system aaa radius hostname retransmit

	nv set system aaa radius hostname <hostname-id> retransmit <1-10> Configure the RADIUS retransmit tries for server.	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	retransmit	Integer: 1-10
Default	None (set by RADIUS global config)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius hostname 1.2.3.4 retransmit 5</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>/retransmit/<retransmit-value>	

Related Commands	<pre>nv show system aaa radius hostname &lt;hostname-id&gt; nv show system aaa radius hostname nv show system aaa radius</pre>
Notes	

#### 4.3.6.5.16 nv set system aaa radius hostname timeout

	<pre>nv set system aaa radius hostname &lt;hostname-id&gt; timeout &lt;1-60&gt; Configure the reply timeout for a RADIUS server (seconds).</pre>	
Syntax Description	hostname-id	RADIUS server ID: ipv4, ipv4-unicast, idn-hostname, ipv6
	timeout	Integer: 1-60
Default	None (set by RADIUS global config)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system aaa radius hostname 1.2.3.4 timeout 5</pre>	
REST API	SET https://<ip>/nvue_v1/system/aaa/radius/hostname/<hostname-id>/timeout/<timeout-value>	
Related Commands	<pre>nv show system aaa radius hostname &lt;hostname-id&gt; nv show system aaa radius hostname nv show system aaa radius</pre>	
Notes		

## 4.4 Certificates Management

- [4.4.1 Import Certificate](#)
  - [4.4.1.1 CA Certificate](#)
  - [4.4.1.2 Import Certificate](#)
- [4.4.2 Set the Certificate to Use in NVUE REST API](#)
- [4.4.3 Set the Certificate to Use for GNMI Service](#)
- [4.4.4 Delete Certificates](#)
- [4.4.5 Show Certificate Information](#)
- [4.4.6 Certificate Management Commands](#)

NVOS includes a self-signed certificate and private key to use on the server so that it works out of the box. The switch generates the self-signed certificate and private key when it boots for the first time. The X.509 certificate with the public key is in `/etc/ssl/certs/nvue.pem` and the corresponding private key is in `/etc/ssl/private/nvue.key`.

NVIDIA recommends you use your own certificates and keys.

NVOS lets you manage CA certificates (such as DigiCert or Verisign) and entity (end-point) certificates. Both a CA certificate and an entity certificate can contain a chain of certificates. The CA certificates can be also addressed as trust bundles, which means that CA certs can also include intermediate certificates.

You can import certificates onto the switch (fetch certificates from an external source), set which certificate you want to use for the NVUE REST API, gNMI, NMX, and show information about a certificate, such as the serial number, and the date and time during which the certificate is valid.

## 4.4.1 Import Certificate

- A maximum of 25 entity certificates and a maximum of 25 CA certificates can be imported
  - A single CA certificate entry may contain up to 100 PEM strings and may also include intermediate certificates.
- The imported server/entity certificate contains sensitive private key information. NVIDIA recommends that you use a secure transport such as SFTP, SCP, or HTTPS.
- To import an entity certificate, run an `nv action import system security certificate <cert-id>` command.
- To import a CA certificate, run an `nv action import system security ca-certificate <cert-id>` command.

If the certificate is passphrase protected, the passphrase to be included.

You must provide a certificate ID (<cert-id>) to uniquely identify the certificate you import.

### 4.4.1.1 CA Certificate

The following example imports a CA certificate with a public key and calls the certificate `tls-cert-1`. The public key is a Base64 ASCII encoded PEM string.

```
nvos@switch:~$ nv action import system security ca-certificate tls-cert-1 data ""-----BEGIN CERTIFICATE----- TODO
-----END CERTIFICATE-----""
```

The following example imports a CA certificate with a public key and calls the certificate `tls-cert-1` with URI `scp://user@pass:1.2.3.4:/opt/certs/ca-cert.crt`.

```
nvos@switch:~$ nv action import system security ca-certificate tls-cert-1 uri scp://user@pass:1.2.3.4:/opt/certs/
ca-cert.crt
```

The following example imports a CA certificate with a public key and calls the certificate `tls-cert-1` with URI `scp://user@pass:1.2.3.4:/opt/certs/ca-cert.crt`. External makes CA certificate standalone, which means certificate is not attached to common system CA certificates bundle in `/etc/ssl/certs/ca-certificates.crt`.

```
nvos@switch:~$ nv action import system security ca-certificate tls-cert-1 uri scp://user@pass:1.2.3.4:/opt/certs/
ca-cert.crt external
```

### 4.4.1.2 Import Certificate

The following example imports an entity certificate bundle (public + private key) and calls the certificate `tls-cert-1`. The certificate bundle is passphrase protected with `mypassphrase`.

A certificate bundle must be in `.p12` format.

To generate a certificate bundle with the `.p12` format, use the following command:

```
openssl pkcs12 -export -out $cert_filename.p12 -in $cert_filename.pem -inkey $cert_filename.key -passout pass:$p12_pass
```

Example of importing the.p12 bundle file:

```
nvos@switch:~$ nv action import system security certificate tls-cert-1 passphrase mypassphrase uri-bundle scp://user@pass:1.2.3.4:/opt/certs/cert.p12
```

The following example imports an entity certificate bundle and calls the certificate `tls-cert-local`. The certificate is located on top of the local machine under `/home/admin` path:

```
nv action import system security certificate tls-cert-local uri-public-key file://127.0.0.1/home/admin/cert.crt uri-private-key file://127.0.0.1/home/admin/cert.key  
#Using absolute path is also possible. Example:  
nv action import system security certificate tls-cert-local uri-public-key file:///home/admin/cert.crt uri-private-key file:///home/admin/cert.key
```

The following example imports an entity certificate with the public key URI `scp://user@pass:1.2.3.4` and private key URI `scp://user@pass:1.2.3.4`, and calls the certificate `tls-cert-1`. The certificate is not passphrase protected.

```
nvos@switch:~$ nv action import system security certificate tls-cert-1 uri-public-key scp://user@pass:1.2.3.4 uri-private-key scp://user@pass:1.2.3.4
```

## 4.4.2 Set the Certificate to Use in NVUE REST API

 When updating a new CA, make sure to perform the "nv config save" before proceeding with a reboot.

You can configure the NVUE REST API to use a specific certificate.

The following example configures the API to use the certificate `tls-cert-1`:

```
nvos@switch:~$ nv set system api certificate tls-cert-1  
nvos@switch:~$ nv config apply
```

The following example configures the API to use the self-signed certificate:

```
nvos@switch:~$ nv set system api certificate self-signed  
nvos@switch:~$ nv config apply
```

To unset the certificate to use with the NVUE REST API:

```
nvos@switch:~$ nv unset system api certificate tls-cert-1
```

### 4.4.3 Set the Certificate to Use for GNMI Service

You can configure the GNMI to use a specific certificate.

The following example configures the API to use the certificate `tls-cert-1`:

```
nvos@switch:~$ nv set system gnmi-server certificate tls-cert-1
nvos@switch:~$ nv config apply
```

The following example configures the API to use the self-signed certificate:

```
nvos@switch:~$ nv set system gnmi-server certificate self-signed
nvos@switch:~$ nv config apply
```

To unset the certificate to use with the NVUE REST API:

```
nvos@switch:~$ nv unset system gnmi-server certificate tls-cert-1
```

### 4.4.4 Delete Certificates

- To delete an entity certificate and the key data stored on the switch, run the [nv action delete system security certificate <cert-id>](#) command.
- To delete a CA certificate and the key data stored on the switch, run the [nv action delete system security ca-certificate <cert-id>](#) command.

The following command deletes the certificate `tls-cert-1`:

```
nvos@switch:~$ nv action delete system security certificate tls-cert-1
```

### 4.4.5 Show Certificate Information

- To show all the entity certificates on the switch, run the [nv show system security certificate](#) command.
- To show all the CA certificates on the switch, run the [nv show system security ca-certificate](#) command.

The following example shows all the entity certificates on the switch:

```
nvos@switch:~$ nv show system security certificate
```

- To show the applications that are using a specific entity certificate, run the [nv show system security certificate <cert-id> installed](#) command.
- To show the applications that are using a specific CA certificate, run the [nv show system security ca-certificate <cert-id> installed](#) command.

The following example shows the applications that are using a specific entity certificate.

```
nvos@switch:~$ nv show system security certificate tls-cert-1 installed
```

- To show detailed information about a specific entity certificate, run the [nv show system security certificate <cert-id> dump](#) command.
- To show detailed information about a specific CA certificate, run the [nv show system security ca-certificate <cert-id> dump](#) command.

The following example shows detailed information about the CA certificate tls-cert-1:

```
nvos@switch:~$ nv show system security ca-certificate tls-cert-1 dump
```

## 4.4.6 Certificate Management Commands

- [Certificate Management Commands](#)

## 4.4.7 Certificate Management Commands

- [4.4.7.1 nv show system security ca-certificate](#)
- [4.4.7.2 nv show system security certificate](#)
- [4.4.7.3 nv action delete system security ca-certificate](#)
- [4.4.7.4 nv action delete system security certificate](#)
- [4.4.7.5 nv action import system security ca-certificate](#)
- [4.4.7.6 nv action import system security certificate](#)

### 4.4.7.1 nv show system security ca-certificate

	nv show system security ca-certificate Display owned CA certificates.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show sys security ca-certificate Certificate ID  Serial Number      Valid From      Valid To Type           Summary ----- cacert_id      49:***30:01    2024-08-28T07:46:54+00:00  2034-08-26T07:46:54+00:00 external       count: 100</pre>
REST API	GET https://<ip>/nvue_v1/system/security/ca-certificate/
Related Commands	nv action import system security ca-certificate
Notes	

### 4.4.7.2 nv show system security certificate

	nv show system security certificate Display owned certificates.
--	--

Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show sys security certificate Certificate ID  Serial Number      Valid From          Valid To Summary ----- cert_id       BE:F2:03:51:D8:F7:BF:71  2024-06-24T13:31:31+00:00  2034-06-22T13:31:31+00:00 0</pre>
REST API	GET https://<ip>/nvue_v1/system/security/ca-certificate/
Related Commands	nv action import system security certificate {cacert id}
Notes	

#### 4.4.7.3 nv action delete system security ca-certificate

	nv action delete system security caa-certificate <cacert-id> Delete system security CA certificate.	
Syntax Description	cacert-id	CA certificate ID removed during import
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action delete system security ca-certificate {cacert-id}</pre>	
REST API	DELETE https://<ip>/nvue_v1/system/security/ca-certificate/{cacert id}	
Related Commands	nv action import system security ca-certificate	
Notes		

#### 4.4.7.4 nv action delete system security certificate

	nv action delete system security certificate <cert-id> Delete system security CA certificate.	
Syntax Description	cert-id	Certificate ID removed during import
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action delete system security certificate {cert id}</pre>	
REST API	DELETE https://<ip>/nvue_v1/system/security/certificate/{cert id}	
Related Commands	nv action import system security certificate {cert id}	

Notes	
-------	--

#### 4.4.7.5 nv action import system security ca-certificate

	nv action import system security ca-certificate <cacert-id> Import system security CA certificate bundle.	
	uri	A local/remote URI from where the certificate file (containing the public-key) can be retrieved. Supports: ftp, scp and sftp (e.g., scp://user[:password]@hostname/path/filename, file:///absolute-path/filename)
Syntax Description	remote-url	A local/remote URI from where the certificate file (containing the CA certificate bundle) can be retrieved.
	data	The raw data bytes (e.g., PEM string) of the CA certificates bundle.
	external_ca	Optional parameter to import certificate without appending it to system CA certificates bundle at /etc/ssl/certs/ca-certificates.crt.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action import system security ca-certificate tls-cert-1 data "&lt;CA-certificate&gt;"</pre> <pre>admin@nvos:~\$ nv action import system security ca-certificate tls-cert-1 uri scp://user@pass:1.2.3.4:/ca-cert.crt</pre> <pre>admin@nvos:~\$ nv action import system security ca-certificate tls-cert-1 uri file:///ca-cert.crt</pre> <pre>admin@nvos:~\$ nv action import system security ca-certificate tls-cert-1 uri scp://user@pass:1.2.3.4:/ca-cert.crt external</pre>	
REST API	POST https://<ip>/nvue_v1/system/security/ca-certificate/{cacert-id}	
Related Commands	nv action delete system security ca-certificate {cacert-id} nv show sys security ca-certificate	
Notes		

#### 4.4.7.6 nv action import system security certificate

	nv action import system security certificate <cert-id> <uri-public-key <uri-path> uri-private-key <uri-path>   uri-bundle <uri-path> [passphrase]   data> Import system security certificate.	
Syntax Description	cert-id	Unique Certificate ID that was named by the user
	uri-public-key	A local/remote URI from where the public key file can be retrieved.

	<table border="1"> <tr> <td>uri-private-key</td> <td>A local/remote URI from where the private key file can be retrieved.</td> </tr> <tr> <td>uri-bundle</td> <td>A local/remote URI from where the certificate file containing the certificate bundle can be retrieved. Needs to be in .p12 format.</td> </tr> <tr> <td>uri-path</td> <td>A local/remote URI from where the certificate file (containing the CA certificate bundle) can be retrieved.</td> </tr> <tr> <td>data</td> <td>The raw data bytes (e.g., PEM string) of the certificates bundle</td> </tr> <tr> <td>passphrase</td> <td>Optional passphrase if certificate bundle is passphrase protected</td> </tr> </table>	uri-private-key	A local/remote URI from where the private key file can be retrieved.	uri-bundle	A local/remote URI from where the certificate file containing the certificate bundle can be retrieved. Needs to be in .p12 format.	uri-path	A local/remote URI from where the certificate file (containing the CA certificate bundle) can be retrieved.	data	The raw data bytes (e.g., PEM string) of the certificates bundle	passphrase	Optional passphrase if certificate bundle is passphrase protected
uri-private-key	A local/remote URI from where the private key file can be retrieved.										
uri-bundle	A local/remote URI from where the certificate file containing the certificate bundle can be retrieved. Needs to be in .p12 format.										
uri-path	A local/remote URI from where the certificate file (containing the CA certificate bundle) can be retrieved.										
data	The raw data bytes (e.g., PEM string) of the certificates bundle										
passphrase	Optional passphrase if certificate bundle is passphrase protected										
Default	N/A										
History	25.02.1884										
Example	<pre>admin@nvos:~\$ nv action import system security certificate tls-cert-1 passphrase mypassphrase uri-bundle scp://user@pass:1.2.3.4:/opt/certs/cert.p12</pre> <pre>admin@nvos:~\$ nv action import system security certificate tls-cert-1 uri-public-key scp://user@pass:1.2.3.4 uri-private-key scp://user@pass:1.2.3.4</pre> <pre>admin@nvos:~\$ nv action import system security certificate tls-cert-1 uri-public-key file://absolute-path/filename uri-private-key file://absolute-path/filename</pre> <pre>admin@nvos:~\$ nv action import system security certificate tls-cert-1 data "&lt;CA-certificate&gt;"</pre>										
REST API	POST https://<ip>/nvue_v1/system/security/certificate/{cert-id}										
Related Commands	<pre>nv action delete system security certificate {cert-id}</pre> <pre>nv show sys security certificate</pre>										
Notes											

## 4.5 Control and Power

NVOS allows to perform actions such as rebooting the system, initiating a power-cycle, and retrieving information about past reboots.



Operations like firmware installation ([nv action install platform firmware files](#)) may change the behavior of the next system reboot to do a full power-cycle to apply the newly burned firmware.

### 4.5.1 Control and Power Commands

- [Control and Power Commands](#)

## 4.5.2 Control and Power Commands

- [4.5.2.1 nv show system reboot](#)
- [4.5.2.2 nv show system reboot reason](#)
- [4.5.2.3 nv show system reboot history](#)
- [4.5.2.4 nv action reboot system](#)
- [4.5.2.5 nv action power-cycle system](#)

### 4.5.2.1 nv show system reboot

	nv show system reboot Show system reboot information.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show system reboot  ----- operational ----- applied [history] reason gentime 2022_06_02_07_32_05 reason Hardware - Other (Reset from ComEx) user N/A </pre>
REST API	GET https://<ip>/nvue_v1/system/reboot
Related commands	
Notes	

### 4.5.2.2 nv show system reboot reason

	nv show system reboot reason Show the system reboot reason.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show system reboot reason  ----- operational ----- applied gentime 2022_06_02_07_32_05 reason Hardware - Other (Reset from ComEx) user N/A </pre>
REST API	GET https://<ip>/nvue_v1/system/reboot/reason
Related commands	
Notes	

### 4.5.2.3 nv show system reboot history

	nv show system reboot history Show the system reboot history.		
Syntax Description	N/A		
Default	N/A		
History	25.02.1884		
Example	<pre> admin@nvos:~\$ nv show system reboot history   gentime          reason          user   -----          -   1  2022_06_02_07_32_05  Hardware - Other (Reset from ComEx)  N/A </pre>		
REST API	GET https://<ip>/nvue_v1/system/reboot/history		
Related commands			
Notes			

### 4.5.2.4 nv action reboot system

	nv action reboot system {flags} [force] Reboot switch system.		
Syntax Description	flags	proceed	<ul style="list-style-type: none"> <li>• Behavior: Standard reboot process but with force flag to bypass certain checks</li> <li>• Service Handling: Same graceful service shutdown as default mode</li> <li>• System State: Forces reboot even if system conditions normally prevent it</li> <li>• Best For: When system is busy or some checks fail but you still want controlled reboot</li> <li>• Risk: Low risk, same safety as default but more forceful</li> </ul>
		immediate	<ul style="list-style-type: none"> <li>• Behavior: Direct kernel reboot bypassing all NOS reboot infrastructure</li> <li>• Service Handling: No graceful shutdown—processes terminated abruptly</li> <li>• System State: Completely ignores system state</li> <li>• Best For: System recovery, emergency situations, frozen systems</li> <li>• Risk: Highest risk of data loss and file system corruption</li> </ul>
		halt	<ul style="list-style-type: none"> <li>• Behavior: Complete system shutdown instead of a restart</li> <li>• Service Handling: Full graceful shutdown like default mode</li> <li>• System State: Respects system busy state</li> <li>• Best For: Maintenance requiring complete power-off</li> <li>• Risk: Low risk, same safety as default mode</li> </ul>
		<none>	<ul style="list-style-type: none"> <li>• Behavior: Standard reboot with complete safety checks</li> <li>• Service Handling: Waits for services to shut down gracefully</li> <li>• System State: Respects system busy state</li> <li>• Best For: Normal operations, scheduled maintenance</li> <li>• Risk: Lowest risk of data loss or corruption</li> </ul>
	force	Force the action without asking for user confirmation.	
Default	N/A		

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action reboot system Configuration has been modified, but not saved. Type [y] to reboot the system without saving configuration. Type [N] to abort.  Do you want to <b>continue?</b> [Y/N] N System reboot aborted by user</pre> <pre>admin@nvos:~\$ nv action reboot system halt Type [y] to halt the system. Type [N] to abort. WARNING: This operation will shut down the system. You will NOT be able to turn on the system remotely.  Do you want to <b>continue?</b> [y/N]</pre>
REST API	POST https://<ip>/nvue_v1/system
Related commands	nv action install platform firmware files
Notes	<ul style="list-style-type: none"> <li>• The prompt will be displayed only if there is unsaved configuration on the switch. Otherwise, the reboot will be performed as usual.</li> <li>• Using the halt flag will show the prompt unless force flag is passed.</li> <li>• Using the immediate flag will not trigger the firmware upgrade flow that part of normal reboot.</li> <li>• If the switch is in a fatal state, it will exit that state once it comes up.</li> </ul>

#### 4.5.2.5 nv action power-cycle system

	nv action power-cycle system [force] [immediate] Performs system power cycle.	
Syntax Description	force	Force the action without asking for user confirmation.
	immediate	Performs power cycle without closing the system gracefully
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ \$ nv action power-cycle system Configuration has been modified, but not saved. Type [y] to power cycle the system without saving configuration. Type [N] to abort.  Do you want to <b>continue?</b> [y/N] N Action executing ... Power cycle aborted by user</pre> <pre>\$ nv action power-cycle system force Action executing ... System will power cycle in a few seconds Action succeeded \$ Connection to XXXX closed by remote host.</pre>	
REST API	POST https://<ip>/nvue_v1/system	
Related Commands	nv action reboot system	
Notes	<ul style="list-style-type: none"> <li>• Supported only in systems with BMC.</li> <li>• If the switch is in a fatal state, it will exit that state once it comes up.</li> </ul>	

## 4.6 DNS Server

Domain Name System (DNS) is an essential component of modern networking, translating human-readable domain names into IP addresses that systems can understand. Proper DNS configuration ensures seamless connectivity and efficient resolution of network requests.

NVOS provides an ability to manage DNS servers, allowing users to add, configure, and view DNS server configurations as needed.

### 4.6.1 Supported Configurations and Limitations

- Both IPv4 and IPv6 unicast DNS server addresses configurations are supported
- NVOS obtains dynamic DNS entries from DHCP server by default (this will be preserved if no static DNS entries are configured)
- Dynamic DNS configuration will not work if management interface uses static IP configuration
- Linux allows for a maximum of three DNS servers to be configured at the same time

### 4.6.2 DNS Server Commands

- [DNS Server Commands](#)

### 4.6.3 DNS Server Commands

- [4.6.3.1 nv show system dns](#)
- [4.6.3.2 nv show system dns server](#)
- [4.6.3.3 nv set/unset system dns server](#)

#### 4.6.3.1 nv show system dns

	nv show system dns Display DNS configuration.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system dns ----- operational  applied ----- [server] 8.8.8.8      8.8.8.8 [server] 10.7.77.135   10.7.77.135 [server] 10.7.77.192  10.7.77.192</pre>	
REST API	GET https://<ip>/nvue_v1/system/dns	
Related Commands	nv show system dns server nv set/unset system dns server	
Notes		

### 4.6.3.2 nv show system dns server

	nv show system dns server Display list of configured DNS servers.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system dns server  DNS Server ----- 8.8.8.8 10.7.77.135 10.7.77.192</pre>	
REST API	GET https://<ip>/nvue_v1/system/dns/server	
Related Commands	nv set/unset system dns server	
Notes		

### 4.6.3.3 nv set/unset system dns server

	nv set system dns server {dns-server-ip} nv unset system dns server {dns-server-ip} Update the DNS server configuration.	
Syntax Description	dns-server-ip	IPv4 or IPv6 unicast address of a DNS server
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system dns server 249.1.1.1 Error: '249.0.0.0' is not a 'dns-server-ip-address'. IP address must be unicast IPv4/ IPv6 address.  admin@nvos:~\$ nv set system dns server 8.8.8.8 admin@nvos:~\$ nv set system dns server 1.1.1.1 admin@nvos:~\$ nv set system dns server 10.7.77.192 admin@nvos:~\$ nv set system dns server 10.7.77.135 admin@nvos:~\$ nv config apply The maximum number 3 of DNS servers exceeded.  admin@nvos:~\$ nv unset system dns server 10.7.77.135 admin@nvos:~\$ nv unset system dns server admin@nvos:~\$ nv unset system dns</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/dns/server/{dns-server-ip}	
Related Commands	nv show system dns server	
Notes	The maximum number of DNS servers is limited to three by Linux.	

## 4.7 Documentation

Users can view and upload any of the documentation files provided by the NVOS as part of its release, including the EULA, User Manual, Release Notes, and Open Source Licenses.

## 4.7.1 Documentation Commands

- [Documentation Commands](#)

## 4.7.2 Documentation Commands

- [4.7.2.1 nv show system documentation](#)
- [4.7.2.2 nv action upload system documentation files](#)

### 4.7.2.1 nv show system documentation

	nv show system documentation Display system document list.	
Syntax Description	files	Displays system document list in brief mode.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system documentation Name                               Type                               Path -----                               - NVOS_EULA.pdf                       EULA                               /usr/share/nginx/html/ system_documents/eula/NVOS_EULA.pdf NVOS_NVL_Release_Notes.pdf          Release notes                       /usr/share/nginx/html/ system_documents/release_notes/NVOS_NVL_Release_Notes.pdf NVOS_NVL_User_Manual.pdf            User manual                          /usr/share/nginx/html/ system_documents/user_manual/NVOS_NVL_User_Manual.pdf Open_Source_Licenses.txt           Open source licenses                /usr/share/nginx/html/ system_documents/open_source_licenses/Open_Source_Licenses.txt </pre>	
Related Commands	nv action upload system documentation files	
Notes		

### 4.7.2.2 nv action upload system documentation files

	nv action upload system documentation files <file-name> <remote-url> Upload system document to remote server	
Syntax Description	file-name	Document to be uploaded.
	<remote-url>	ftp, tftp, scp and sftp are supported (e.g., scp://username[:password]@hostname/path/filename)
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv action upload system documentation files NVOS_EULA.pdf scp:// user:password@10.1.12.20/tmp/NVOS_EULA.pdf </pre>	
Related Commands	nv show system documentation	
Notes		

## 4.8 Hostname

NVOS provides the ability to set a hostname to the switch. Make sure that the hostname is unique and descriptive.

There are two options to set the hostname: Dynamically through DHCP or Statically in NVUE.

### 4.8.1 Hostname from DHCP

By default, DHCP is enabled, device receives the DHCP Hostname option inside the response and sets the device hostname.

By default, DHCP is enabled for both interfaces that may receive a hostname update during run. NVOS is always updating to the latest hostname received by DHCP.

It is possible to disable a hostname for a certain interface by executing the following:

```
admin@nvos:~$ nv set interface eth0 ip dhcp-client set-hostname disabled
admin@nvos:~$ nv config apply
```



Do not use an underscore (\_), apostrophe ('), or non-ASCII characters in the hostname.

The hostname convention needs to follow "idn-hostname" as defined by either RFC 1123 as for hostname, or an internationalized hostname as defined by [RFC 5890, section 2.3.2.3 \[RFC5890\]](#)

### 4.8.2 Static Hostname

To change the hostname, run the following:

```
admin@nvos:~$ nv set system hostname leaf01
admin@nvos:~$ nv config apply
```



The command prompt in the terminal does not reflect the new hostname until either logging out of the switch or starting a new shell.

### 4.8.3 Hostname Commands

- [Hostname Commands](#)

### 4.8.4 Hostname Commands

#### 4.8.4.1 nv set/unset system hostname

```
nv set/unset system hostname {hostname}
Set/unset the hostname of the switch.
```

Syntax Description	hostname	The new hostname to set
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system hostname switch01 admin@nvos:~\$ nv unset system hostname</pre>	
REST API	PATCH https://<ip>/nvue_v1/system	
Related Commands	nv show system	
Notes		

## 4.9 Management Interfaces

Management interfaces are used in order to provide access to switch management user interfaces (e.g., CLI, openAPI).

NVIDIA switches support out-of-band (OOB) dedicated interfaces (e.g., eth0). In addition, most NVIDIA switches feature a serial port that provides access to the CLI only.

### 4.9.1 Configuring Management Interfaces with Static IP Addresses

The management interface uses DHCP for addressing by default.

To set a static IP address (for example):

```
admin@nvos:~$ nv set interface eth0 ip address 192.0.2.42/24
admin@nvos:~$ nv set interface eth0 ip gateway 192.0.2.1
admin@nvos:~$ nv config apply
```



To help setups with static IP to detect new devices, configuring static IP address will trigger unsolicited announcement messages to the gateway in order to reveal the device's MAC address. This also applies to static IPv6 addresses. Physically disconnecting and connecting the cables will also trigger the same messages when static IP addresses are configured.

### 4.9.2 Configuring IPv6 Address on the Management Interface

1. Set a static IPv6 address. Run:

```
admin@nvos:~$ nv set interface eth0 ip address fdfd:fdfd:7:145:9a03:9bff:fe6b:6ac/64
admin@nvos:~$ nv config apply
```

2. Verify the IPv6 address is configured correctly. Run:

```
admin@nvos:~$ nv show interface eth0
```

## 4.9.3 Default Gateway

To configure manually the default gateway, use the “nv set interface eth0 ip gateway” command, with “0.0.0.0” as prefix and mask. The next-hop address must be within the range of one of the IP interfaces on the system.

## 4.9.4 IP DHCP Client

To enable DHCP client and DHCPv6 client state on Management (eth0, eth1) interfaces use the following commands:

1. nv set interface <interface-id> ip dhcp-client state <enabled | disabled>
2. nv set interface <interface-id> ip dhcp-client6 state <enabled | disabled>



Enabling DHCP client on management interface (eth0, eth1) automatically enables DHCPv6 client and vice-versa.

## 4.9.5 Management Interface Commands

- [4.9.5.1 nv show interface](#)
- [4.9.5.2 nv show interface link](#)
- [4.9.5.3 nv show interface ip](#)
- [4.9.5.4 nv set/unset interface link state](#)
- [4.9.5.5 nv set/unset interface description](#)
- [4.9.5.6 nv set/unset interface ip](#)
- [4.9.5.7 nv unset interface](#)
- [4.9.5.8 nv set/unset interface link mtu](#)
- [4.9.5.9 nv set/unset interface link speed](#)
- [4.9.5.10 nv set/unset interface link duplex](#)
- [4.9.5.11 nv set interface link auto-negotiate](#)
- [4.9.5.12 nv set interface ip autoconf](#)
- [4.9.5.13 nv set interface ip gateway](#)
- [4.9.5.14 nv set interface ip arp-timeout](#)
- [4.9.5.15 VRF](#)
  - [4.9.5.15.1 nv show vrf](#)
  - [4.9.5.15.2 nv set interface ip vrf](#)
- [4.9.5.16 IP DHCP Client](#)
  - [4.9.5.16.1 nv set interface ip dhcp-client state](#)
  - [4.9.5.16.2 nv set interface ip dhcp-client set-hostname](#)
  - [4.9.5.16.3 nv set interface ip dhcp-client6 state](#)
  - [4.9.5.16.4 nv set interface ip dhcp-client6 set-hostname](#)
  - [4.9.5.16.5 nv action renew interface ip dhcp-client](#)
  - [4.9.5.16.6 nv action renew interface ip dhcp-client6](#)

## 4.9.5.1 nv show interface

	<b>nv show interface [interface-id]</b> Displays details of an IPoB/eth0 interface. If not interface is selected, summary of attributes of all interfaces will be displayed.	
Syntax Description	interface-id	Name of the interface to display (e.g., eth0)
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show interface eth0 ----- operational ----- applied ----- ip vrf                default          default arp-timeout        1800            1800 autoconf           enabled         enabled dhcp-client state              enabled         enabled set-hostname      enabled         enabled is-running         yes            yes has-lease          yes            yes dhcp-client6 state              enabled         enabled set-hostname      enabled         enabled is-running         yes            yes has-lease          no             no [address]          10.7.89.7/20 [address]          fdfe:fdfe:7:80:eaeb:d3ff:fe4b:70b8/64 [gateway] link auto-negotiate    on             on duplex            full          full speed             1G           1G mac               e8:eb:d3:4b:70:b8 counters in-bytes          2.53 MB in-pkts           15719 in-drops          0 in-errors         0 out-bytes         144.50 KB out-pkts          917 out-drops         0 out-errors        0 carrier-transitions 4 mtu               1500          1500 state             up            up ifindex           2 type              eth           eth </pre> <pre> admin@nvos:~\$ nv show interface Interface State Speed MTU Type Description Logical state Physical state Summary ----- acp1      down          nv1          Down          Polling acp2      down          nv1          Down          Polling acp3      down          nv1          Down          Polling acp4      down          nv1          Down          Polling eth0      up            1G          1500         eth          Down          Polling ip.address: 10.7.145.68/21 ip.address: fdfe:fdfe:7:145:9e63:c0ff:fe72:b212/64 ip.address: fdfe:fdfe:7:145::1000:4240/128 fnm1      down          fnm          Down          Polling fnm2      down          fnm          Down          Polling lo        up            65536       loopback     Down          Polling ip.address: 127.0.0.1/16 ip.address: ::1/128 sw1pls1   down          nv1          Down          Disabled sw1pls2   down          nv1          Down          Disabled </pre>	
REST API	GET <a href="https://&lt;ip&gt;/nvue_v1/interface">https://&lt;ip&gt;/nvue_v1/interface</a>	
Related Commands	<b>nv set interface</b> <b>nv show interface</b>	
Notes		

## 4.9.5.2 nv show interface link

	nv show interface <interface-id> link {state   counters} Displays link information of an IPoIB/eth0 interface.	
Syntax Description	interface-id	Name of the interface to display (e.g., eth0)
	state	Show only the data relating to state
	stats	Show only the data relating to statistics
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show interface eth0 link                                 operational      applied ----- auto-negotiate                   on              on duplex                           full            full mtu                               1500           1500 speed                            1G             1G state                             up              up counters   carrier-transitions             6   in-bytes                       32.84 MB   in-drops                       0   in-errors                      0   in-pkts                       209466   out-bytes                      16.99 MB   out-drops                      0   out-errors                    0   out-pkts                      121334 mac                              08:c0:eb:58:7f:90 </pre> <pre> admin@nvos:~\$ nv show interface eth0 link state -- operational applied -- ----- up </pre> <pre> admin@nvos:~\$ nv show interface eth0 link counters                                 operational ----- in-bytes                       4.20 MB in-pkts                       5561 in-drops                      0 in-errors                     0 out-bytes                      1.42 MB out-pkts                      2942 out-drops                      0 out-errors                    0 carrier-transitions            5 </pre>	
REST API	GET https://<ip>/nvue_v1/interface/<interface-id>/link	
Related Commands	nv show interface nv set interface link nv unset interface link	
Notes		

## 4.9.5.3 nv show interface ip

	nv show interface <interface-id> ip {address   dhcp-client   dhcp-client6   gateway} Displays IP configuration and state of an IPoIB/eth0 interface.	
Syntax Description	interface-id	Name of the interface to display (e.g., ib0, eth0)
	address	Display the IP address configuration of an IPoIB/eth0 interface
	dhcp-client	Display the DHCPv4 configuration and state of an IPoIB/eth0 interface

	dhcp-client6	Display the DHCPv6 configuration and state of an IPoIB/eth0 interface
	gateway	Display the IP gateway configuration of an IPoIB/eth0 interface
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show interface eth0 ip ----- operational                                applied ----- vrf  default arp-timeout                                1800 autoconf                                   enabled dhcp-client   state                                    enabled   set-hostname                            enabled   is-running                               yes   has-lease                                yes dhcp-client6   state                                    enabled   set-hostname                            enabled   is-running                               yes   has-lease                                no [address]                                  10.7.89.7/20 [address]                                  fdfd:fdfd:7:80:eaeb:d3ff:fe4b:70b8/64 [gateway] </pre> <pre> admin@nvos:~\$ nv show interface eth0 ip address ----- 10.7.144.154/21 fdfd:fdfd:7:145::1000:454c/128 fdfd:fdfd:7:145:ac0:ebff:fe58:7f90/64 </pre> <pre> admin@nvos:~\$ nv show interface eth0 ip dhcp-client ----- operational    applied ----- set-hostname  enabled       enabled state         enabled       enabled has-lease     yes is-running    yes </pre>	
REST API	GET https://<ip>/nvue_v1/interface/<interface-id>/ip	
Related Commands	nv show interface nv set interface ip nv unset interface ip	
Notes		

#### 4.9.5.4 nv set/unset interface link state

	nv set interface <interface-id> link state {value} nv unset interface <interface-id> link state {value} Set/unset the administrative link state of a given IPoIB/eth0 interface.	
Syntax Description	interface-id	Name of the interface whose link state to set (e.g., eth0)
	value	New value for the link state: up, down
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv unset interface eth0 link state admin@nvos:~\$ nv set interface eth0 link state up </pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/link/state	

Related Commands	nv show interface nv set interface link nv unset interface link state
Notes	

#### 4.9.5.5 nv set/unset interface description

	nv set interface <interface-id> description nv unset interface <interface-id> description Sets the description of a given IPoB/eth0 interface. The unset form of the command sets the description of a given IPoB/eth0 interface to empty.	
Syntax Description	interface-id	Name of the interface whose link state to set, (e.g. ib0, eth0)
	value	New value for the description.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 description "mgmt interface" admin@nvos:~\$ nv unset interface eth0 description</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>	
Related Commands	nv show interface nv unset interface description	
Notes		

#### 4.9.5.6 nv set/unset interface ip

	nv set interface <interface-id> ip address <ip-prefix-id> nv unset interface <interface-id> ip address <ip-prefix-id> Sets the IP address of a given IPoB/eth0 interface. The unset form of the command deletes one or more IP addresses assigned to a given IPoB/eth0 interface.	
Syntax Description	interface-id	Name of the interface whose IP address to set (e.g., ib0, eth0)
	ip-prefix-id	IP address and netmask to assign to the interface
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip address 10.10.1.1/8</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip/address	
Related Commands	nv show interface nv unset interface ip	
Notes		

#### 4.9.5.7 nv unset interface

	nv unset interface <interface-id> Sets all attributes of an IPoIB/eth0 interface to default values.	
Syntax Description	interface-id	Name of the interface to set to default values (e.g., eth0)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset interface eth0</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>	
Related Commands	<a href="#">nv show interface</a> <a href="#">nv set interface</a>	
Notes		

#### 4.9.5.8 nv set/unset interface link mtu

	nv set interface <interface-id> link mtu <bytes> nv unset interface <interface-id> link mtu Sets the Maximum Transmission Unit (MTU) of this interface. The unset form of the command resets the MTU to its default.	
Syntax Description	interface-id	Name of the interface to display (e.g., eth0)
	bytes	Range: 1280-9000
Default	1500	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 link mtu 1500</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/link	
Related Commands	nv show interface link	
Notes		

#### 4.9.5.9 nv set/unset interface link speed

	nv set interface <interface-id> link speed <speed> nv unset interface <interface-id> link speed Sets the interface speed. The unset form of the command resets the speed setting for this interface to its default value.	
Syntax Description	interface-id	Name of the interface to display (e.g., eth0)
	speed	10M, 100M, 1000M
Default	1000M	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set interface eth0 link speed 100M</pre>
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/link
Related Commands	nv show interface link
Notes	

#### 4.9.5.10 nv set/unset interface link duplex

	nv set interface <interface-id> link duplex <duplex> nv unset interface <interface-id> link duplex Sets the interface duplex. The unset form of the command resets the duplex setting for this interface to its default value.	
Syntax Description	interface-id	The interface (e.g., eth0)
	duplex	full, half
Default	full	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 link duplex full</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/link	
Related Commands	nv show interface link	
Notes		

#### 4.9.5.11 nv set interface link auto-negotiate

	nv set interface <interface-id> link auto-negotiate <auto> Sets link speed and characteristic auto negotiation.	
Syntax Description	interface-id	The interface (e.g., eth0)
	auto	on, off
Default	on	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 link auto-negotiate on</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/link	
Related Commands	nv show interface link	
Notes		

#### 4.9.5.12 nv set interface ip autoconf

	nv set interface <interface-id> ip autoconf <autoconf> IPv6 Stateless Address Autoconfiguration (SLAAC)	
Syntax Description	interface-id	The interface
	autoconf	enable, disable
Default	Enable	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip autoconf enable</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip	
Related Commands	nv unset interface ip gateway	
Notes		

#### 4.9.5.13 nv set interface ip gateway

	nv set interface <interface-id> ip gateway {gateway-ip} Sets default gateway IP address for an interface.	
Syntax Description	interface-id	The interface
	gateway-ip	The gateway IP address
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip gateway 1.1.1.1</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip/gateway	
Related Commands	nv show interface ip	
Notes		

#### 4.9.5.14 nv set interface ip arp-timeout

	nv set interface <interface-id> ip arp-timeout <time> Sets IPv4 arp timeout (in seconds) for an interface.	
Syntax Description	interface-id	The interface
	time	Seconds. Range: 60-28800
Default	1800	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip arp-timeout 2100</pre>	

REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip
Related Commands	
Notes	

## 4.9.5.15 VRF

### 4.9.5.15.1 nv show vrf

	show vrf {vrf-id}{loopback   loopback ip   loopback ip address {ip-prefix-id}} Shows VRFs.	
Syntax Description	vrf-id	VRF
	loopback	Return loopback interface details of a VRF.
	loopback ip	Return IP details of a VRF loopback interface.
	loopback ip address	Return details of the IP addresses.
	ip-prefix-id	IPv4 or IPv6 address and route prefix in CIDR notation
	<none>	Shows all VRFs
Default	N/A	
Configuration Mode	config	
History	25.02.1884	

<p><b>Example</b></p>	<pre> admin@nvos:~\$ nv show vrf Name      Table  Summary ----- + default 254    IP Address: 127.0.0.1/8 + default      IP Address:  ::1/128 + mgmt     1001   IP Address: 127.0.0.1/8 + mgmt      IP Address:  ::1/128 </pre> <pre> admin@nvos:~\$ nv show vrf mgmt operational  applied  description ----- table        1001     auto     The routing table number, between 1001-1255, used by the named VRF.... evpn enable 'off'. loopback ip  [address] 127.0.0.1/8 127.0.0.1/8 static IPv4 or IPv6 address  [address] ::1/128      ::1/128 ptp enable 'on'. router bgp enable 'off'. ospf enable 'off'. [rib] [static] </pre> <pre> admin@nvos:~\$ nv show vrf mgmt loopback operational  applied  description ----- ip  [address] 127.0.0.1/8 127.0.0.1/8 static IPv4 or IPv6 address  [address] ::1/128      ::1/128 </pre> <pre> admin@nvos:~\$ nv show vrf mgmt loopback ip operational  applied  description ----- [address] 127.0.0.1/8 127.0.0.1/8 static IPv4 or IPv6 address [address] ::1/128      ::1/128 </pre> <pre> admin@nvos:~\$ nv show vrf mgmt loopback ip address ----- 127.0.0.1/8 ::1/128 </pre> <pre> admin@nvos:~\$ nv show vrf mgmt loopback ip address 127.0.0.1/8 operational  applied  description -- </pre>
<p>REST API</p>	<p>GET <a href="https://&lt;ip&gt;/nvue_v1/vrf/{vrf-id}">https://&lt;ip&gt;/nvue_v1/vrf/{vrf-id}</a></p>
<p>Related Commands</p>	
<p>Notes</p>	<p>FRR must be running to be able to use this command.</p>

### 4.9.5.15.2 nv set interface ip vrf

	<pre> nv set interface &lt;interface-id&gt; ip vrf {vrf-name} </pre> <p>Assigns an interface to a VRF.</p>		
<p>Syntax Description</p>	<table border="1"> <tr> <td data-bbox="440 1886 576 1921"> <p>interface-id</p> </td> <td data-bbox="582 1886 1394 1921"> <p>The interface</p> </td> </tr> </table>	<p>interface-id</p>	<p>The interface</p>
<p>interface-id</p>	<p>The interface</p>		

	vrf-name	The VRF: default/ mgmt
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip vrf mgmt</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip	
Related Commands	nv unset interface ip vrf	
Notes		

## 4.9.5.16 IP DHCP Client

### 4.9.5.16.1 nv set interface ip dhcp-client state

	nv set interface <interface-id> ip dhcp-client state {enabled   disabled} nv unset interface <interface-id> ip dhcp-client Enables/disables DHCP client for an interface. The unset form of the command returns the DHCP client to its default state.	
Syntax Description	interface-id	The interface
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip dhcp-client state enabled</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip/dhcp-client	
Related Commands	nv set interface ip dhcp-client6 state	
Notes	Configuring the DHCP client for either IPv4 or IPv6 will set the same configuration for both.	

### 4.9.5.16.2 nv set interface ip dhcp-client set-hostname

	nv set interface <interface-id> ip dhcp-client set-hostname {enabled   disabled} nv unset interface <interface-id> ip dhcp-client set-hostname Allows/disallows DHCP client to set system hostname from DHCP. The unset form of the command returns the system hostname to its default state.	
Syntax Description	interface-id	The interface
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip dhcp-client set-hostname enabled</pre>	

REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip/dhcp-client
Related Commands	nv set interface ip dhcp-client6 set-hostname nv set interface ip dhcp-client state
Notes	Configuring the DHCP client for either IPv4 or IPv6 will set the same configuration for both.

#### 4.9.5.16.3 nv set interface ip dhcp-client6 state

	nv set interface <interface-id> ip dhcp-client6 state {enabled   disabled} nv set interface <interface-id> ip dhcp-client6 Enables/disables DHCPv6 client for an interface. The unset form of the command returns the DHCPv6 client to its original state.	
Syntax Description	interface-id	The interface
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip dhcp-client6 state enabled</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip/dhcp-client6	
Related Commands	nv set interface ip dhcp-client state	
Notes	Configuring the DHCP client for either IPv4 or IPv6 will set the same configuration for both.	

#### 4.9.5.16.4 nv set interface ip dhcp-client6 set-hostname

	nv set interface <interface-id> ip dhcp-client6 set-hostname {enabled   disabled} nv unset interface <interface-id> ip dhcp-client6 set-hostname Allows/disallows DHCP client to set system hostname from DHCPv6. The unset form of the command returns the DHCP client to its default state.	
Syntax Description	interface-id	The interface
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface eth0 ip dhcp-client6 set-hostname enabled</pre>	
REST API	PATCH https://<ip>/nvue_v1/interface/<interface-id>/ip/dhcp-client6	
Related Commands	nv set interface ip dhcp-client set-hostname nv set interface ip dhcp-client6 state	
Notes	Configuring the DHCP client for either IPv4 or IPv6 will set the same configuration for both.	

#### 4.9.5.16.5 nv action renew interface ip dhcp-client

	nv action renew interface <interface-id> ip dhcp-client Renews DHCPv4 lease for this interface.	
Syntax Description	interface-id	The interface
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action renew <b>interface</b> eth0 ip dhcp-client</pre>	
REST API	POST https://<ip>/nvue_v1/interface/<interface-id>/ip	
Related Commands		
Notes		

#### 4.9.5.16.6 nv action renew interface ip dhcp-client6

	nv action renew interface <interface-id> ip dhcp-client6 Renews DHCPv6 lease for this interface.	
Syntax Description	interface-id	The interface
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action renew <b>interface</b> eth0 ip dhcp-client6</pre>	
REST API	POST https://<ip>/nvue_v1/interface/<interface-id>/ip/dhcp-client6	
Related Commands		
Notes		

## 4.10 Resource Management

NVOS provides commands for managing and monitoring essential system resources. These tools provide general information about the system, insights into the system's overall health, and resources utilization and usage.

### 4.10.1 Resource Management Commands

- [Resource Management Commands](#)

## 4.10.2 Resource Management Commands

- [4.10.2.1 nv show system](#)
- [4.10.2.2 nv show system cpu](#)
- [4.10.2.3 nv show system memory](#)

### 4.10.2.1 nv show system

	nv show system Show general system information.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show system operational ----- build          Debian GNU/Linux 11 (bullseye) uptime        0:26:21 hostname      sw-gorilla-07 product-name  nvos product-release 25.01.3000 platform      x86_64-mlnx_mqm9700-r0 system-memory 1913 MB used / 5722 MB free / 7635 MB total swap-memory   0 MB used / 0 MB free / 0 MB total health-status OK date-time     2024-02-11 15:59:21 status        System is ready timezone      Etc/UTC </pre>
REST API	GET https://<ip>/nvue_v1/system
Related commands	
Notes	

### 4.10.2.2 nv show system cpu

	nv show system cpu Show system CPU.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show system cpu operational ----- core-count    4 model         Intel(R) Pentium(R) CPU D1508 @ 2.20GHz utilization   3.0% </pre>
REST API	GET https://<ip>/nvue_v1/system/cpu
Related commands	

Notes	
-------	--

### 4.10.2.3 nv show system memory

	nv show system memory Show system memory.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system memory  B - Bytes, KB - Kilobytes, MB - Megabytes, GB - Gigabytes, % - Percent  Physical buffer: 53.12 MB Physical cache: 1.82 GB Physical free: 5.74 GB Physical total: 7.46 GB Physical used: 1.72 GB Physical utilization: 23.06 % Swap free: 0 B Swap total: 0 B Swap used: 0 B Swap utilization: 0.00 %</pre>
REST API	GET https://<ip>/nvue_v1/system/memory
Related commands	
Notes	

## 4.11 Security

### 4.11.1 SSD Wipe

To wipe the SED SSD (encrypted with a user-defined or default password), retrieve the PSID and perform a disk wipe.

#### 4.11.1.1 Get SSD PSID

The Physical Security ID (PSID) is required to unlock and wipe the SSD. The SSD PSID can be retrieved by running the following command:

```
sudo cat /var/run/hw-management/eeprom/vpd_data | grep PSID
```

#### 4.11.1.2 Perform Disk Wipe

Once you have the PSID, proceed to wipe the SSD.

Run the following Linux command to initiate the wipe process:

```
sudo sedutil-cli --yesIreallywanttoERASEALLmydatausingthePSID <psid> /dev/nvme0
```

The `psid` value needs to be taken from the earlier step in "[Get SSD PSID](#)".



In order to execute these commands, the user must have `sudo` capabilities.



This action is destructive and irreversible. Proceeding will completely erase all data on the switch's disk.

## 4.11.2 Recovery Flow After SSD Wipe

### 4.11.2.1 Prerequisites

Before starting the recovery process, ensure the following requirements are met:

- PXE Server Setup
  - A PXE (Preboot Execution Environment) server must be installed and running in the lab network
  - The PXE server should be configured to point to the NVIDIA ONIE image and automatically boot from it without user interaction
- Required Resources
  - NVIDIA ONIE image
  - NVOS image
  - Provisioning package



Performing an SSD wipe will erase the previous NVOS configuration, and it will not be recoverable. Ensure that any required configurations or data are backed up before proceeding.

### 4.11.2.2 Recovery Steps

1. Wait for NVIDIA ONIE installation to finish (should take up to 3 minutes).
2. Connect to ONIE via ssh. Please refer to UM for ONIE default credentials.
3. Once connected to ONIE, stop onie install by running `onie-stop`.
4. Copy provisioning package (e.g., `sed_provisioning_83.03.0001.tgz`) to `/tmp`.
5. Extract provisioning script and run it.

```
cd /tmp
tar -xzf sed_provisioning_83.03.0001.tgz
./sedutil_init.sh
```

6. Wait for the script to finish (which can take up to 2 minutes). The device will perform power-cycle.
7. Perform a new NVOS image install following the guidelines in the [Installing a New NVOS Image](#) section.

## 4.11.3 Security Commands

- [Password Hardening](#)
- [SED Commands](#)

## 4.11.4 Password Hardening

NVOS implements robust password hardening policies to enhance security and protect user accounts.

By default, passwords must meet the following requirements:

- Minimum length of 8 characters
- Inclusion of at least the following:
  - One uppercase letter
  - One lowercase letter
  - One number
  - One special character from the set ``~!@#$%^&*()-_+=|[]{};:','<.>/?` and white space
- The password cannot reuse any of the last 10 previously saved passwords
- The password cannot contain the username

The user can choose to disable/change one or more policies or even deactivate the feature entirely.



Password hardening checks are forced in NVUE during SET/PATCH operations. Any changes to the policies or feature state must be applied first and will only take effect for subsequent operations.

### 4.11.4.1 Password Hardening Commands

- [Password Hardening Commands](#)

### 4.11.4.2 Password Hardening Commands

- [4.11.4.2.1 nv show system security password-hardening](#)
- [4.11.4.2.2 nv set system security password-hardening state](#)
- [4.11.4.2.3 nv set system security password-hardening digits-class](#)
- [4.11.4.2.4 nv set system security password-hardening expiration](#)
- [4.11.4.2.5 nv set system security password-hardening expiration-warning](#)
- [4.11.4.2.6 nv set system security password-hardening history-cnt](#)
- [4.11.4.2.7 nv set system security password-hardening len-min](#)
- [4.11.4.2.8 nv set system security password-hardening lower-class](#)
- [4.11.4.2.9 nv set system security password-hardening reject-user-passw-match](#)
- [4.11.4.2.10 nv set system security password-hardening special-class](#)
- [4.11.4.2.11 nv set system security password-hardening upper-class](#)

#### 4.11.4.2.1 nv show system security password-hardening

	nv show system security password-hardening Displays the password hardening rules applied on top of the switch.	
Syntax Description	N/A	
Default	The example contains the default values of the feature	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system security password-hardening  ----- operational  applied state                enabled            enabled reject-user-passw-match  enabled            enabled lower-class          enabled            enabled upper-class          enabled            enabled digits-class         enabled            enabled special-class        enabled            enabled expiration-warning    15                 15 expiration            180                180 history-cnt           10                 10 len-min               8                  8 </pre>	
Rest API	GET https://<id>/nvue_v1/system/security/password_hardening	
Related Commands	nv set system security password-hardening	
Notes	Password hardening rules are applied only to locally stored passwords	

#### 4.11.4.2.2 nv set system security password-hardening state

	nv set system security password-hardening state <enabled   disabled> Enable or disable the password hardening feature	
Syntax Description	state	enabled, disabled
Default	enabled	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv set system security password-hardening state enabled </pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/state/	
Related Commands	nv show system security password-hardening	
Notes	When password hardening is enabled - the switch does not accept hashed passwords	

#### 4.11.4.2.3 nv set system security password-hardening digits-class

	nv set system security password-hardening digits-class<enabled   disabled> Enable or disable the requirement to enforce digits in the password	
Syntax Description	digits-class	enabled, disabled
Default	enabled	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system security password-hardening state enabled</pre>
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/digits-class/
Related Commands	nv show system security password-hardening
Notes	

#### 4.11.4.2.4 nv set system security password-hardening expiration

	nv set system security password-hardening expiration [<integer days>] Number of days for password validity, afterwards user will be prompted to change his password	
Syntax Description	expiration	expiration days (1-365)
Default	180	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system security password-hardening expiration 200</pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/expiration/	
Related Commands	nv show system security password-hardening	
Notes		

#### 4.11.4.2.5 nv set system security password-hardening expiration-warning

	nv set system security password-hardening expiration-warning [<integer days>] Number of days for password warning which will alert the user that he needs to change his password before it expires. The alert will appear on the login screen for the user	
Syntax Description	expiration-warning	warning days (1-30)
Default	15	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system security password-hardening expiration-warning 10</pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/expiration-warning/	
Related Commands	nv show system security password-hardening	
Notes		

#### 4.11.4.2.6 nv set system security password-hardening history-cnt

	nv set system security password-hardening history-cnt [<integer count>] Number of passwords the system will compare the current password against. If the password is equal to one of the previously configured password, the system will reject it.	
Syntax Description	history-cnt	history count (1-100)
Default	10	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system security password-hardening history-cnt 5</pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/history-cnt/	
Related Commands	nv show system security password-hardening	
Notes		

#### 4.11.4.2.7 nv set system security password-hardening len-min

	nv set system security password-hardening len-min [<integer length>] Set the minimum length for a password.	
Syntax Description	len-min	length (6-32)
Default	8	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system security password-hardening len-min 10</pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/len-min/	
Related Commands	nv show system security password-hardening	
Notes		

#### 4.11.4.2.8 nv set system security password-hardening lower-class

	nv set system security password-hardening lower-class <enabled/disabled> Enable or disable the requirement to enforce lower case letters in the password.	
Syntax Description	lower-class	enabled/disabled
Default	enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system security password-hardening lower-class &lt;enabled/disabled&gt;</pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/lower-class/	
Related Commands	nv show system security password-hardening	

Notes	
-------	--

#### 4.11.4.2.9 nv set system security password-hardening reject-user-passw-match

	nv set system security password-hardening reject-user-passw-match<enabled/ disabled> Enable or disable allowing the username and password to be identical.	
Syntax Description	reject-user-passw-match	enabled/disabled
Default	enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system security password-hardening state &lt;enabled/disabled&gt;</pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/reject-user-passw-match/	
Related Commands	nv show system security password-hardening	
Notes		

#### 4.11.4.2.10 nv set system security password-hardening special-class

	nv set system security password-hardening special-class <enabled/disabled> Enable or disable the requirement to enforce special characters in the password.	
Syntax Description	special-class	feature state
Default	enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system security password-hardening special-class disabled</pre>	
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/special-class/	
Related Commands	nv show system security password-hardening	
Notes	<ul style="list-style-type: none"> <li>The special characters allowed are: `~!@#%&amp;^&amp;*(-)_+= [[{}];:','&lt;.&gt;/? and white space</li> <li>These characters must be accompanied by quotation marks "" in order to be received correctly in the password string.</li> <li>Example: nv set system aaa user example password "123%\$Aabcv"</li> </ul>	

#### 4.11.4.2.11 nv set system security password-hardening upper-class

	nv set system security password-hardening upper-class <enabled/disabled> Enable or disable the requirement to enforce upper case letters in the password.	
Syntax Description	upper-class	enabled/disabled
Default	enabled	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set system security password-hardening upper-class enabled</pre>
REST API	PATCH https://<id>/nvue_v1/system/security/password_hardening/upper-class/
Related Commands	nv show system security password-hardening
Notes	

## 4.11.5 SED Commands

### 4.11.5.1 nv action change system security sed-password

	nv action change system security [sed-password] Change the SED password by setting a new password chosen by the user.	
Syntax Description	sed-password	Minimum password length: 8 characters Maximum password length: 250 characters
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action change system security sed-password 12345678</pre>	
REST API	POST https://<ip>/nvue_v1/system/security	
Related Commands		
Notes	The password can be up to 250 characters but it is hashed down to 32 bytes to conform to the size required by the drive.	

## 4.12 System API

NVOS offers commands for managing and monitoring its external REST API interface. These capabilities allow users to configure, enable, and monitor API settings, ensuring secure and efficient interaction with external systems.

### 4.12.1 Key Functionalities

- **API State Management:** Enable or disable external REST API access.
- **Port Configuration:** Specify the port number on which the REST API listens, with a default setting of 443.
- **Certificate Management:** View and configure the CA certificate for mutual TLS (mTLS) connections. For more examples on how to import and set certificates, Please see "[Certificates Management](#)" section.
- **Operational Metrics:** Retrieve detailed API usage statistics, including active connections, total requests, and the status of ongoing requests.

## 4.12.2 System API Commands

- [System API Commands](#)

## 4.12.3 System API Commands

- [4.12.3.1 nv show system api](#)
- [4.12.3.2 nv set/unset system api state](#)
- [4.12.3.3 nv set/unset system api port](#)
- [4.12.3.4 nv show system api mtls](#)
- [4.12.3.5 nv set system api mtls ca-certificate](#)
- [4.12.3.6 nv set system api certificate](#)

### 4.12.3.1 nv show system api

	nv show system api Show NVUE external REST API configuration.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system api ----- state           enabled        enabled port            443           443 certificate     self-signed   self-signed connections   active         1   accepted      15698   handled       15698   requests     14490   reading       0   writing        1   waiting       0           </pre>	
REST API	GET https://<ip>/nvue_v1/system/api	
Related Commands	nv set/unset system api state nv set/unset system api port	
Notes		

### 4.12.3.2 nv set/unset system api state

	nv set/unset system api state <enabled   disabled> Set the REST API external access state.	
Syntax Description	state	enabled, disabled
Default	enabled	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show system api admin@nvos:~\$ nv set system api state enabled</pre>
REST API	PATCH https://<ip>/nvue_v1/system/api
Related Commands	nv show system api
Notes	

### 4.12.3.3 nv set/unset system api port

	nv set/unset system api port <port-number> Set the REST API listen port.	
Syntax Description	port-number	Port number of the remote syslog server: 1-65535
Default	443	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system api port 443</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/api	
Related Commands	nv show system api	
Notes		

### 4.12.3.4 nv show system api mtls

	nv show system api mtls Show configured CA certificate for API mTLS connections.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system api mtls               operational  applied ----- ca-certificate nvue_root   nvue_root</pre>	
REST API	GET https://<ip>/nvue_v1/system/api/mtls	
Related Commands		
Notes	nv show system api	

### 4.12.3.5 nv set system api mtls ca-certificate

	nv set system api mtls ca-certificate <cacert-id> Set CA certificate for API mTLS connection.	
Syntax Description	cacert-id	CA certificate ID string

Default	none
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set system api mtls ca-certificate nvue_root</pre>
REST API	PATCH https://<ip>/nvue_v1/system/api/mtls?rev= Content-Type: application/json {"ca-certificate": "cacert_id"}
Related Commands	
Notes	

### 4.12.3.6 nv set system api certificate

	nv set system api certificate <cert-id> Set certificate for API connection.	
Syntax Description	cert-id	Certificate ID string
Default	none	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system api certificate cert cert_id</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/api/mtls?rev= Content-Type: application/json {"certificate": "cert_id"}	
Related Commands		
Notes		

## 4.13 Time Synchronization

NVOS relies on the system clock for displaying the time and timestamping messages. The date and time can be configured manually, or synchronization with Network Time Protocol (NTP) servers can be enabled.

For more information, see the following sections:

- [Date and Time](#)
- [NTP](#)

### 4.13.1 Date and Time

NVOS allows to control the system time zone and clock settings.

#### 4.13.1.1 Time Zone

Default time zone is set to Coordinated Universal Time (UTC). User may change the time zone configuration by executing the following:

```
admin@nvos:~$ nv set system timezone Etc/UTC
admin@nvos:~$ nv config apply
```

## 4.13.1.2 Clock

The system date and time can be manually changed. NTP servers configured on the switch will supersede any manually entered date-time settings.

```
admin@nvos:~$ nv action change system date-time 2024-12-24 10:25:13
```

## 4.13.1.3 Date and Time Commands

- [Date and Time Commands](#)

## 4.13.1.4 Date and Time Commands

### 4.13.1.4.1 nv action change system date-time

	nv action change system date-time <yyyy-mm-dd> <hh:mm:ss> Sets the time and date.	
Syntax Description	hh:mm:ss	Time
	yyyy-mm-dd	Date
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action change system date-time 2021-01-01 10:10:11</pre>	
REST API	POST https://<ip>/nvue_v1/system/date-time	
Related Commands	nv show system	
Notes	Unable to change date and time in case NTP is enabled.	

### 4.13.1.4.2 nv set/unset system timezone

	nv set system timezone <timezone> nv unset system timezone Sets the system time zone. The no form of the command resets time zone to its default (Etc/UTC).	
Syntax Description	timezone	A valid timezone value (e.g., Africa/Abidjan, Brazil/Acre, Africa/Accra, Chile/Continental)
Default	Etc/UTC	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system timezone Etc/UTC</pre>	

REST API	PATCH https://<ip>/nvue_v1/system
Related Commands	nv show system
Notes	<p>The time zone may be specified in one of three ways:</p> <ul style="list-style-type: none"> <li>• A nearby city whose time zone rules to follow. The system has a large list of cities which can be displayed by the help and completion system. They are organized hierarchically because there are too many of them to display in a flat list. A given city may be required to be specified in two, three, or four words, depending on the city</li> <li>• An offset from Etc/GMT. This will be in the form Etc/GMT-offset, Etc/GMT+&lt;0-14&gt;, Etc/GMT-&lt;1-12&gt;</li> <li>• Etc/UTC (Universal Time, which is almost identical to GMT), and this is the default time zone</li> </ul>

## 4.13.2 NTP

Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over variable-latency data networks. NTP is intended to synchronize all participating computers to within a few milliseconds of Coordinated Universal Time (UTC) and is designed to mitigate the effects of variable network latency. NTP can usually maintain time to within tens of milliseconds over the public Internet, and can achieve better than one millisecond accuracy in local area networks under ideal conditions.

### 4.13.2.1 NTP Authenticate

When authentication of incoming NTP packets is enabled, the switch ensures that they come from an authenticated time source before using them for time synchronization on the switch. Authentication keys are created and marked as trusted.

To add a key to be used for authentication, take the following steps

1. Create the key.

```
admin@nvos:~$ nv set system ntp key 1
```

2. Specify the value.

```
admin@nvos:~$ nv set system ntp key 1 value mystrongpassword
```

3. Trust new added key.

```
admin@nvos:~$ nv set system ntp key 1 trusted yes
```

4. Assign the key to the server.

```
admin@nvos:~$ nv set system ntp server 10.34.1.1 key 1
```

### 4.13.2.2 NTP Authentication Key

An authentication key may be created and used to authenticate incoming NTP packets. For the key to be used, make sure the following is in place.

1. It should be shared with the NTP server sending the NTP packet.
2. The key should be marked as trusted.
3. NTP authentication should be enabled on the system.

### 4.13.2.3 NTP Commands

- [NTP Commands](#)

### 4.13.2.4 NTP Commands

- [4.13.2.4.1 nv show system ntp](#)
- [4.13.2.4.2 nv show system ntp server](#)
- [4.13.2.4.3 nv show system ntp key](#)
- [4.13.2.4.4 nv set/unset system ntp](#)
- [4.13.2.4.5 nv set/unset system ntp server](#)
- [4.13.2.4.6 nv set/unset system ntp key](#)

#### 4.13.2.4.1 nv show system ntp

	nv show system ntp [-w status] Display NTP configuration and status. It shows the next information: status (offset, reference), authentication, listening interface, NTP state, NTP DHCP state, VRF device.	
Syntax Description	-w status	Status view: Show short NTP status
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system ntp operational          applied ----- offset               10.03 ms reference            1.2.3.4 status               <b>synchronized</b> authentication      disabled           disabled listen              eth0 state                enabled           enabled vrf                  <b>default</b>          <b>default</b> [server]             0.ua.ntp.pool.org 0.ua.ntp.pool.org [server]             1.1.1.1           1.1.1.1 [server]             1.il.ntp.pool.com 1.il.ntp.pool.com [server]             1.ua.pool.ntp.org 1.ua.pool.ntp.org [server]             10.11.100.5      10.11.100.5 [server]             ntpost1          ntpost1 [server]             ntpost2          ntpost2 [server]             time.google.com  time.google.com           </pre> <pre> admin@nvos:~\$ nv show system ntp -w status operational          applied ----- offset               10.03 ms reference            1.2.3.4 status               <b>synchronized</b>           </pre>	
REST API	GET <a href="https://&lt;ip&gt;/nvue_v1/system/ntp">https://&lt;ip&gt;/nvue_v1/system/ntp</a>	
Related Commands		
Notes	By default, NTP DHCP is enabled, so if the DHCP server sends out NTP servers to the clients, the switch will get it and will be synchronized with it.	

#### 4.13.2.4.2 nv show system ntp server

	nv show system ntp server [<server-id>   -w query] Display NTP servers configuration and query data. The command shows the following information: aggressive polling state, server type, authentication key, server IP address, server state, trustiness, protocol version.	
Syntax Description	server-id	Hostname or IP address of the NTP server. Show configuration for a specific configured NTP server.
	-w query	Query view: Show query data from NTP servers.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system ntp server NTP server      Aggressive  Type      KeyID  Resolve as      State      Trusted  Ver ----- 0.ua.ntp.pool.org  on          server    216.40.34.37  enabled        no        4 1.1.1.1          on          server    1.1.1.1       enabled        no        4 1.il.ntp.pool.com on          server    DNS resolution failed  enabled        no        4 1.ua.pool.ntp.org on          server    193.34.155.3  enabled        no        4 10.11.100.5      on          server    10.11.100.5   enabled        no        4 ntpghost1        on          pool      DNS resolution failed  disabled       no        3 ntpghost2        on          server    DNS resolution failed  enabled        no        4 time.google.com  on          server    216.239.35.12 enabled        no        4           </pre> <pre> admin@nvos:~\$ nv show system ntp server -w query NTP server      Stratum  Type  When  Auth  Delay  Jitter  Offset  Status  Poll  Reach  Ref clock ----- 0.ua.ntp.pool.org  none    0.000  0.000  +0.000  64    no    .INIT.  16 u 1.1.1.1          none    0.000  0.000  +0.000  64    no    .INIT.  16 u 10.11.100.5      none    0.000  0.000  +0.000  64    no    .INIT.  16 u 1.ua.pool.ntp.org none    81.419  0.884  +0.286  64    yes   129.134.28.123  2 u 43 time.google.com  none    61.700  0.965  +0.990  64    yes   .GOOG.  1 u 44           </pre>	
REST API	GET <a href="https://&lt;ip&gt;/nvue_v1/system/ntp/server">https://&lt;ip&gt;/nvue_v1/system/ntp/server</a> GET <a href="https://&lt;ip&gt;/nvue_v1/system/ntp/server/&lt;server-id&gt;">https://&lt;ip&gt;/nvue_v1/system/ntp/server/&lt;server-id&gt;</a>	
Related Commands		
Notes		

#### 4.13.2.4.3 nv show system ntp key

	nv show system ntp key [<key-id>] Display NTP authentication keys inventory. It shows the next information: key ID, key type, obfuscated value.	
Syntax Description	key-id	NTP authentication key ID. Show the configuration of a specific authentication key.
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show system ntp key KeyID  Trusted  Type  Value -----  -----  -----  ----- 1      yes      md5   * 42     yes      sha1  *</pre> <pre>admin@nvos:~\$ nv show system ntp key 42 -----  -----  -----  ----- operational  applied trusted yes      yes type sha1    sha1 value *      *</pre>
REST API	<p>GET <a href="https://&lt;ip&gt;/nvue_v1/system/ntp/key">https://&lt;ip&gt;/nvue_v1/system/ntp/key</a></p> <p>GET <a href="https://&lt;ip&gt;/nvue_v1/system/ntp/key/&lt;key-id&gt;">https://&lt;ip&gt;/nvue_v1/system/ntp/key/&lt;key-id&gt;</a></p>
Related Commands	
Notes	

#### 4.13.2.4.4 nv set/unset system ntp

	<p>nv {set,unset} system ntp [listen {eth0}   state {enabled,disabled}   dhcp {enabled,disabled}   vrf {default}   authentication {enabled,disabled}] Update NTP global configuration.</p>	
Syntax Description	listen	NTP interface to listen on. Limited to "eth0" only.
	state	NTP state configuration.
	dhcp	Use NTP servers leased from DHCP server.
	vrf	VRF to run NTP daemon in. Limited to "default" only.
	authentication	Enables NTP authentication.
Default	listen	eth0
	state	enabled
	dhcp	enabled
	vrf	default
	authentication	disabled
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset system ntp</pre> <pre>admin@nvos:~\$ nv set system ntp authentication enabled</pre> <pre>admin@nvos:~\$ nv unset system ntp state</pre>	
REST API	PATCH <a href="https://&lt;ip&gt;/nvue_v1/system/ntp">https://&lt;ip&gt;/nvue_v1/system/ntp</a>	
Related Commands	nv set system ntp server	
Notes		

#### 4.13.2.4.5 nv set/unset system ntp server

	nv {set,unset} system ntp server [<server-id> [aggressive-polling {on,off}   state {enabled,disabled}   key <1-65535>   trusted {yes,no}   version {3,4}   association-type {server,pool}]] Update the NTP servers configuration.	
Syntax Description	server-id	Hostname or IP address of the NTP server.
	aggressive-polling	Aggressive polling of the server.
	state	Temporarily disable/enable this NTP server.
	key	Specify the key ID to securely communicate with the remote NTP server.
	trusted	Trust that NTP server. If authentication is configured this will additionally force all time updates to only use trusted servers.
	version	The NTP protocol version to communicate with the remote server.
	association-type	NTP server association type.
Default	aggressive-polling	on
	state	disabled
	trusted	no
	version	4
	association-type	server
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset system ntp server</pre>	
	<pre>admin@nvos:~\$ nv unset system ntp server 10.10.10.10</pre>	
	<pre>admin@nvos:~\$ nv set system ntp server 10.10.10.10 trusted yes</pre>	
	<pre>admin@nvos:~\$ nv unset system ntp server 10.10.10.10 key</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/ntp/server PATCH https://<ip>/nvue_v1/system/ntp/server/<server-id>	
Related Commands	nv set system ntp key	

Notes	<p>When authentication of incoming NTP packets is enabled, the switch ensures that they come from an authenticated time source before using them for time synchronization on the switch.</p> <p>An authentication key may be created and used to authenticate incoming NTP packets. For the key to be used, make sure the following is in place.</p> <ol style="list-style-type: none"> <li>1. It should be shared with the NTP server sending the NTP packet.</li> <li>2. The key should be trusted.</li> <li>3. NTP authentication should be enabled on the system</li> </ol>
-------	---

#### 4.13.2.4.6 nv set/unset system ntp key

	<pre>nv {set,unset} system ntp key [&lt;key-id&gt; [trusted {yes,no}   type {md5,sha1}   value &lt;value&gt;]]</pre> <p>Update the NTP keys configuration.</p>	
Syntax Description	key-id	NTP authentication key ID.
	trusted	Trust that NTP authentication key.
	type	Authentication key type.
	value	Secret authentication key value.
Default	trusted	no
	type	md5
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset system ntp key</pre> <pre>admin@nvos:~\$ nv unset system ntp key 3</pre> <pre>admin@nvos:~\$ nv set system ntp key type sha1</pre> <pre>admin@nvos:~\$ nv unset system ntp key type</pre>	
REST API	<pre>PATCH https://&lt;ip&gt;/nvue_v1/system/ntp/key</pre> <pre>PATCH https://&lt;ip&gt;/nvue_v1/system/ntp/key/&lt;key-id&gt;</pre>	
Related Commands	nv set system ntp server	
Notes		

## 4.14 User Interfaces

This section provides information on the interfaces available for users to manage and validate the status of their switch system.

- [Secure Shell \(SSH\) for Remote Access](#)
- [User Interface Commands](#)

## 4.14.1 Secure Shell (SSH) for Remote Access

- [4.14.1.1 Overview](#)
- [4.14.1.2 Configure Timeouts and Sessions](#)
  - [4.14.1.2.1 Message of the Day](#)
  - [4.14.1.2.2 Generate and Install an SSH Key Pair](#)
    - [4.14.1.2.2.1 Generate an SSH Key Pair](#)
    - [4.14.1.2.2.2 Install an Authorized SSH Key](#)
    - [4.14.1.2.2.3 PKA-Only](#)
  - [4.14.1.2.3 Troubleshooting](#)

NVOS uses the OpenSSH package to provide access to the system using the Secure Shell (SSH) protocol.

### 4.14.1.1 Overview

You can configure SSH to provide login access to the root user and to specific user accounts, limit SSH to listen on a specific VRF, and configure timeouts and session options.



- SSH configuration changes take effect only in new SSH sessions and do not impact existing ones.
- SSH Strict Mode: By default, NVOS disables the following SSH server configurations: X11, TCP forwarding, and compression and enforces secure ciphers.
- Modified configurations take effect only in new SSH sessions and do not impact existing ones.

### 4.14.1.2 Configure Timeouts and Sessions

You can configure the following SSH timeout and session options:

- The number of login attempts allowed before rejecting the SSH session. You can specify a value between 3 and 100. The default value is 3 login attempts.
- The number of seconds allowed before login times out. You can specify a value between 1 and 600. The default value is 120 seconds.
- The TCP port numbers that listen for incoming SSH sessions. You can specify a value between 1 and 65535.
- The number of minutes a session can be inactive before the SSH server terminates the connection. The default value is 0 minutes.
- The maximum number of SSH sessions allowed per TCP connection. You can specify a value between 1 and 100. The default value is 10.
- Unauthenticated SSH sessions:
  - The maximum number of unauthenticated SSH sessions allowed. You can specify a value between 1 and 10000. The default value is 100.
  - The number of unauthenticated SSH sessions allowed before throttling starts. You can specify a value between 1 and 10000. The default value is 10.

- The starting percentage of connections to reject above the throttle start count before reaching the session count limit. You can specify a value between 1 and 100. The default value is 30.

The following example configures the number of login attempts allowed before rejecting the SSH session to 10 and the number of seconds allowed before login times out to 200:

```
admin@nvos:~$ nv set system ssh-server authentication-retries 10
admin@nvos:~$ nv set system ssh-server login-timeout 200
admin@nvos:~$ nv config apply
```

The following example configures the TCP port that listens for incoming SSH sessions to 443:

```
admin@nvos:~$ nv set system ssh-server port 443
admin@nvos:~$ nv config apply
```

The following example configures the amount of time a session can be inactive before the SSH server terminates the connection to 5 minutes (300 seconds) and the maximum number of SSH sessions allowed per TCP connection to 5. The default `inactive-timeout` is 15 minutes and the default `max-sessions` is 10:

```
admin@nvos:~$ nv set system ssh-server inactive-timeout 5
admin@nvos:~$ nv set system ssh-server max-sessions 5
admin@nvos:~$ nv config apply
```

#### 4.14.1.2.1 Message of the Day

When you log into the switch, NVOS shows system health information and login notifications.

Example:

```
Last login: Thu Jun 19 04:52:31 UTC 2025 from 10.20.30.40 on pts/0
Number of total successful connections since last 1 days: 6
```

### SSH Login Notifications

NVOS shows the following SSH login information on the console after authentication:

- The date and time of the last successful login.
- The number of unsuccessful logins after the last successful login.
- The date and time of the last unsuccessful login.
- Changes to a user account after the last login (password, role, group, and so on).
- The location (terminal or IP) of the last successful or unsuccessful login.
- The total number of successful logins after a specific date and time.

NVOS displays login notifications for both SSH and serial connections. The information can help to detect unwanted or malicious activities, such as suspicious logins or password and role changes.

To configure the time period in days during which to show login notifications, run the `nv set system ssh-server login-record-period <days>` command. You can specify a value between 1 and 30. The default value is 1.

The following example sets the SSH login notification period to 20 days:

```
admin@nvos:~$ nv set system ssh-server login-record-period 20
admin@nvos:~$ nv config apply
```

To set the SSH login notification period back to the default value (1 day), run the `nv unset system ssh-server login-record-period` command.

To show the configured SSH login notification period, run the `nv show system ssh-server` command. See [Troubleshooting](#) below.

#### 4.14.1.2.2 Generate and Install an SSH Key Pair

This section describes how to generate an SSH key pair on one system and install the key as an authorized key on another system.

##### 4.14.1.2.2.1 Generate an SSH Key Pair

To generate an SSH key pair, run the `ssh-keygen` command and follow the prompts.

NVOS does not support sha1 ssh key exchange methods.

To configure the system without a password, do not enter a passphrase when prompted in the following step.

```
admin@host01:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/admin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/admin/.ssh/id_rsa.
Your public key has been saved in /home/admin/.ssh/id_rsa.pub.
The key fingerprint is:
5a:b4:16:a0:f9:14:6b:51:f6:f6:c0:76:1a:35:2b:bb cumulus@leaf04
The key's randomart image is:
+----[RSA 2048]-----+
|  +.o  o |
| o * o . o |
| o + o O o |
| + . = O |
| . S o . |
| + . |
| . E |
+-----+

```

##### 4.14.1.2.2.2 Install an Authorized SSH Key

To install an authorized SSH key, you take the contents of an SSH public key and add it to the SSH authorized key file ( `~/.ssh/authorized_keys` ) of the user.

A public key is a text file with three space separated fields:

```
<type> <key string> <comment>
```

Field	Description
<type>	The algorithm you want to use to hash the key. The algorithm can be <code>ecdsa-sha2-nistp256</code> , <code>ecdsa-sha2-nistp384</code> , <code>ecdsa-sha2-nistp521</code> , <code>ssh-dss</code> , <code>ssh-ed25519</code> , or <code>ssh-rsa</code> (the default value).

Field	Description
<key string>	A base64 format string for the key.
<comment>	A single word string. By default, this is the name of the system that generated the key. NVUE uses the <comment> field as the key name.

The procedure to install an authorized SSH key is different based on whether the user is an NVUE managed user or a non-NVUE managed user.

#### NVUE Managed User

The following example adds an authorized key named `prod_key` to the user `admin2`. The content of the public key file is `ssh-rsa 1234 prod_key`.

```
admin@nvos:~$ nv set system aaa user admin2 ssh authorized-key prod_key key XABDB3NzaC1yc2EAAAADAQABAAQGCvjs/
RFPPhxLQMkckONg+1RE1PTIO2JQhzFN9TRg7ox7o0tFZ+IzSB99lr2dmmVe8FRWgxVjc...
admin@nvos:~$ nv set system aaa user admin2 ssh authorized-key prod_key type ssh-rsa
admin@nvos:~$ nv config apply
```

#### 4.14.1.2.2.3 PKA-Only

This configuration allows blocking password authentication from users that have a configured authorized key.

To enable this flag, run the following:

```
admin@nvos:~$ nv set system ssh-server pka-only enabled
admin@nvos:~$ nv config apply
```

#### 4.14.1.2.3 Troubleshooting

To show all the current SSH server configuration settings, run the NVUE `nv show system ssh-server` command:

```
admin@nvos:~$ nv show system ssh
-----
operational  applied
-----
authentication-retries 6          6
login-timeout          120         120
inactive-timeout       20          20
login-record-period    1           1
max-sessions           100         100
pka-only               disabled    disabled
[port]                 22          22
```

To show the TCP port numbers that listen for incoming SSH sessions, run the `nv show system ssh-server port` command. You can also show information for a specific port with the `nv show system ssh-server port <port>` command.

### 4.14.1.3 SSH Commands

- [4.14.1.3.1 show ssh-server](#)

- [4.14.1.3.2 nv set/unset system ssh-server inactive-timeout](#)
- [4.14.1.3.3 nv set/unset system ssh-server max-sessions](#)
- [4.14.1.3.4 nv set/unset system ssh-server port](#)

#### 4.14.1.3.1 show ssh-server

	nv show ssh-server Limit the maximum number of concurrent user sessions.	
Syntax Description	N/A	
Default	100	
History	25.02.1884 25.02.4257 Updated command output	
Example	<pre>admin@nvos:~\$ nv show system ssh-server ----- authentication-retries 6          6 login-timeout          120         120 inactive-timeout       15          15 login-record-period    1           1 max-sessions           100         100 pka-only               disabled    disabled [port]                 22          22</pre>	
REST API	GET https://<ip>/nvue_v1/system/ssh-server	
Related Commands	nv show system ssh-server	
Notes	Applied for new connections only.	

#### 4.14.1.3.2 nv set/unset system ssh-server inactive-timeout

	nv set/unset system ssh-server inactive-timeout [<integer time>] Configure inactive timeout for ssh connections in minutes <0-35000>.	
Syntax Description	integer time	Number of minutes: 0-35000 minutes
Default	15 minutes	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system ssh-server inactive-timeout 300</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/ssh-server	
Related Commands	nv show system ssh-server	
Notes	Applied for new connections only.	

#### 4.14.1.3.3 nv set/unset system ssh-server max-sessions

	nv set/unset system ssh-server max-sessions [<integer sessions>] Configuring the maximum number of ssh connections <3-100>.	
Syntax Description	integer sessions	Number of sessions: 3-100 sessions

Default	100
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set system ssh-server max-sessions 10</pre>
REST API	PATCH https://<ip>/nvue_v1/system/ssh-server
Related Commands	nv show system ssh-server
Notes	Applied for new connections only.

#### 4.14.1.3.4 nv set/unset system ssh-server port

	nv set/unset system ssh-server port <port-id> Configure the ports for the systems ssh-server.	
Syntax Description	port-id	TCP Port ID (integer: 1-65535)
Default	22	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system ssh-server port 777</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/ssh-server	
Related Commands	nv show system ssh-server	
Notes	Multiple ports can be configured. By default, port 22 is used. Any user configuration will remove the default, the user need to configure port 22 explicitly.	

### 4.14.2 User Interface Commands

- [4.14.2.1 System Message](#)
  - [4.14.2.1.1 nv show system message](#)
  - [4.14.2.1.2 nv set/unset system message pre-login message](#)
  - [4.14.2.1.3 nv set/unset system message post-login message](#)
  - [4.14.2.1.4 nv set/unset system message post-logout](#)
- [4.14.2.2 SSH](#)
  - [4.14.2.2.1 show ssh-server](#)
  - [4.14.2.2.2 nv set/unset system ssh-server inactive-timeout](#)
  - [4.14.2.2.3 nv set/unset system ssh-server max-sessions](#)
  - [4.14.2.2.4 nv set/unset system ssh-server port](#)
- [4.14.2.3 Serial-Console](#)
  - [4.14.2.3.1 nv show system serial-console](#)
  - [4.14.2.3.2 nv set/unset system serial-console inactivity-timeout](#)
  - [4.14.2.3.3 nv set/unset system serial-console sysrq-capabilities](#)

## 4.14.2.1 System Message

### 4.14.2.1.1 nv show system message

	<b>nv show system message</b> Display banner messages for different terminal session events: pre-login, post-login and post-logout messages.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system message operational                                applied ----- pre-login  NVOS switch                        NVOS switch post-login  post-logout           </pre>	
REST API	GET https://<ip>/nvue_v1/system/message	
Related Commands	nv set system message pre-login nv set system message post-login	
Notes	Printed as raw, unformatted output	

### 4.14.2.1.2 nv set/unset system message pre-login message

	<b>nv set system message pre-login {message}</b> <b>nv unset system message pre-login</b> Set/unset the pre-login message.	
Syntax Description	message	The new pre-login message to set
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv set system message pre-login new-pre admin@nvos:~\$ nv unset system message pre-login           </pre>	
REST API	PATCH https://<ip>/nvue_v1/system/message	
Related Commands	nv show system message	
Notes		

### 4.14.2.1.3 nv set/unset system message post-login message

	nv set system message post-login {message} nv unset system message post-login Set the post-login message. The unset form of the command unsets the post-login message.	
Syntax Description	message	The new post-login message to set
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system message post-login new-post admin@nvos:~\$ nv unset system message post-login</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/message	
Related Commands	nv show system message	
Notes		

### 4.14.2.1.4 nv set/unset system message post-logout

	nv set system message post-logout <message> nv unset system message post-logout Set/unset the post-logout message.	
Syntax Description	message	The new post-logout message to set
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system message post-logout Goodbye</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/message	
Related Commands	nv show system message	
Notes		

## 4.14.2.2 SSH

### 4.14.2.2.1 show ssh-server

	nv show ssh-server Limit the maximum number of concurrent user sessions.	
Syntax Description	N/A	
Default	100	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show system ssh-server ----- authentication-retries 6          6 inactive-timeout       15         15 login-record-period    1           1 login-timeout          120        120 max-sessions           100        100 [port]                 22         22</pre>
REST API	GET https://<ip>/nvue_v1/system/ssh-server
Related Commands	nv show system ssh-server
Notes	Applied for new connections only.

#### 4.14.2.2.2 nv set/unset system ssh-server inactive-timeout

	nv set/unset system ssh-server inactive-timeout [<integer time>] Configure inactive timeout for ssh connections in minutes <0-35000>.	
Syntax Description	integer time	Number of minutes: 0-35000 minutes
Default	15 minutes	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system ssh-server inactive-timeout 300</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/ssh-server	
Related Commands	nv show system ssh-server	
Notes	Applied for new connections only.	

#### 4.14.2.2.3 nv set/unset system ssh-server max-sessions

	nv set/unset system ssh-server max-sessions [<integer sessions>] Configuring the maximum number of ssh connections <3-100>.	
Syntax Description	integer sessions	Number of sessions: 3-100 sessions
Default	100	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system ssh-server max-sessions 10</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/ssh-server	
Related Commands	nv show system ssh-server	
Notes	Applied for new connections only.	

#### 4.14.2.2.4 nv set/unset system ssh-server port

	nv set/unset system ssh-server port <port-id> Configure the ports for the systems ssh-server.	
Syntax Description	port-id	TCP Port ID (integer: 1-65535)
Default	22	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system ssh-server port 777</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/ssh-server	
Related Commands	nv show system ssh-server	
Notes	Multiple ports can be configured. By default, port 22 is used. Any user configuration will remove the default, the user need to configure port 22 explicitly.	

#### 4.14.2.3 Serial-Console

##### 4.14.2.3.1 nv show system serial-console

	nv show system serial-console Show system serial console.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system serial-console ----- operational applied sysrq-capabilities disabled disabled inactivity-timeout 15 15 connected-to cpu cpu</pre>	
REST API	GET https://<ip>/nvue_v1/system/serial-console	
Related Commands	nv set system serial-console inactivity-timeout nv set system serial-console sysrq-capabilities	
Notes	Applied after reconnection only.	

##### 4.14.2.3.2 nv set/unset system serial-console inactivity-timeout

	nv set/unset system serial-console inactivity-timeout Configure inactivity timeout for serial console in minutes <0-35000>	
Syntax Description	N/A	
Default	15	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system serial-console inactivity-timeout 300</pre>
REST API	PATCH https://<ip>/nvue_v1/system/serial-console
Related Commands	nv show system serial-console
Notes	Applied after reconnection only.

#### 4.14.2.3.3 nv set/unset system serial-console sysrq-capabilities

	nv set/unset system serial-console sysrq-capabilities Enables or disables SysRq key capabilities.
Default	Disabled
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set system serial-console sysrq-capabilities enabled</pre>
REST API	PATCH https://<ip>/nvue_v1/system/ssh-server
Related Commands	nv show system serial-console
Notes	

## 4.15 Zero-Touch Provisioning

- [4.15.1 Running DHCP-ZTP](#)
- [4.15.2 Dynamic Content Configuration](#)
  - [4.15.2.1 DHCP Options](#)
  - [4.15.2.2 HTTP Headers](#)
- [4.15.3 ZTP Configuration](#)
  - [4.15.3.1 ZTP Configuration File](#)
  - [4.15.3.2 ZTP Configuration image](#)
  - [4.15.3.3 ZTP Configuration startup-file](#)
  - [4.15.3.4 ZTP Configuration commands-list](#)
  - [4.15.3.5 ZTP Configuration connectivity-check](#)
  - [4.15.3.6 ZTP Configuration provisioning-script](#)
  - [4.15.3.7 ZTP Configuration nmx-commands-list](#)
- [4.15.4 ZTP and OS Upgrade](#)
- [4.15.5 Example Configurations](#)
  - [4.15.5.1 DHCPv4 Configuration](#)
  - [4.15.5.2 DHCPv6 Configuration](#)
  - [4.15.5.3 ZTP Configuration File Example](#)
  - [4.15.5.4 Commands List Example](#)
  - [4.15.5.5 YAML Formatted File Configuration](#)
  - [4.15.5.6 Provisioning Script Example](#)
- [4.15.6 Zero-Touch Provisioning Commands](#)

Zero-touch Provisioning (ZTP) automates the initial configuration of switch systems at boot time. It helps minimize manual operation and reduce customer initial deployment costs. ZTP allows for automatic upgrade of the switch with a specified OS image, setting up a switch configuration, and checking connectivity to external resources.

A switch configuration is applied by using either of the following two formats: YAML format file configuration or CLI commands in the regular text file.

The user can create a YAML configuration file by saving the running configuration by “nv config save” and later uploading that file to an external provisioning server.

The textual configuration file can be created by running “nv config show -o commands” and pasting the output to the text file. That text file can be edited later to add more commands or replace the ones in the list.

ZTP can execute custom provisioning scripts. These scripts can be of any type, must include a shebang, and will run during the ZTP process.

## 4.15.1 Running DHCP-ZTP

There is no explicit command to enable ZTP. It is enabled by default. Disabling it is performed by a user-initiated configuration save (using the command “nv config save”). There are two ways to re-enable ZTP. The first one is to run a “reset factory” command, clearing the configuration of the switch and rebooting the system. The second one is to run “nv action run system ztp” which will just remove the startup configuration and restart ZTP. It is encouraged to run the “reset factory” command.

ZTP is based on DHCP. For ZTP to work, the software enables DHCP by default on all its management interfaces. The switch OS requests option 66 (tftp-server-name) and/or 67 (bootfile-name) from the DHCPv4 server or option 59 (bootfile-url) from the DHCPv6 server and waits for the DHCP responses containing the ZTP JSON configuration file URL. See [ZTP Configuration File](#) section for more information.

The DHCP server must be configured to send back the URL to the ZTP configuration file. For DHCPv4 there are two options supported: Option 66 and Option 67. Providing both options by DHCP server will result in combining them to build TFTP ULR to ZTP configuration file. When using Option 66 the TFTP protocol prefix is omitted. Providing only Option 67 will be considered as URL to the location of the ZTP configuration file.

For DHCPv6, the only option supported is Option 67. It will contain the complete URL to the ZTP JSON configuration file. The format of all the options is a string. Below is a summary table:

DHCP Option	Name	Description
66	tftp-server-name	TFTP-Server address. If specified by server, must be used together with option 67.
67	bootfile-name	URL to download the ZTP JSON configuration file. It can also specify the ZTP JSON file path on TFTP server.
59	dhcp6.bootfile-url	URL to download the ZTP JSON configuration file.

Examples of Options for DHCPv4:

```
option tftp-server-name "198.51.100.8";
option bootfile-name "ztp.json";
```



It will result in: <tftp://192.51.100.1/ztp.json>

Example of only Option 67 for DHCPv4:

```
option bootfile-name "scp://<user>:<pass>@198.51.100.8/ztp.json";
```

Example of only Option 67 for DHCPv6:

```
option dhcp6.bootfile-url "http://[2001:db8::8]/ztp.json";
```

## 4.15.2 Dynamic Content Configuration

When a switch requests a file specified by a URL, it is expected that the server which is handling the request knows which file to be returned. For example, when a switch requests for configuration file, the server needs to be able to give out a configuration file for the requesting switch. This selection of the file can be done in two ways:

- Provisioning Server side: DHCP Options, HTTP Headers
- ZTP Client side: Dynamic URL (see [ZTP Configuration File Section](#))

### 4.15.2.1 DHCP Options

To have the DHCP server discern the proper files based on switch-specific information, the NOS must provide identifying information for the server to classify the switches. NOS provides the following identifying information:

- Product Name—Option 61
- Serial Number—Option 61
- Vendor Class Identifier—Option 77 (Option 15 for DHCPv6)

Upon receiving such DHCP requests from a client, the server should be able to map the switch-specific information to the target file URLs according to predefined rules.

The following are the DHCP options sent by the NVOS switch in the DHCP request:

DHCP Option	Name	Description
61	dhcp-client-identifier	Used to uniquely identify the switch initiating DHCP request. NVOS switches set this value to "NVOS##product-name##serial-no".
77	user-class	Used to optionally identify the type or category of user or applications it represents. NVOS switches set this value to "NVOS-ZTP".
15	dhcp6.user-class	Used to optionally identify the type or category of user or applications it represents. NVOS switches set this value to "NVOS-ZTP".

## 4.15.2.2 HTTP Headers

For a server to make the decision on which file to serve out, it requires uniquely identifiable information about the requesting switch. Then, appropriate logic can be implemented on the server side to process the provided information, and identify and give out the requested file.

All HTTP/HTTPS requests made during ZTP contain switch identification information as part of HTTP headers. Below is the information that is included.

Header	Value	Example
User-Agent	NVOS-ZTP/0.1	
PRODUCT-NAME	<i>String specifying the switch model</i>	N511x_LD
SERIAL-NUMBER	<i>String specifying the manufacture provided serial number</i>	MT1234X56789
BASE-MAC-ADDRESS	<i>Ethernet MAC Address assigned to the switch by the manufacturer</i>	12:34:56:AB:CD:EF
NVOS-VERSION	<i>Version string as seen in 'nv show system version' command</i>	nvos-25.02.xxxx

## 4.15.3 ZTP Configuration

### 4.15.3.1 ZTP Configuration File

For ZTP to automate the provisioning process a user needs to provide a JSON configuration file. The ZTP JSON configuration file consists of configuration sections (objects). Each section has its own options. In other words, ZTP JSON configuration file is an instruction on how to perform provisioning and from where to get provisioning data. Here is an example of the ZTP JSON configuration file:

```
{
  "ztp": {
    "01-image": {
      "install": {
        "url": "http://198.51.100.2/images/nvos.bin"
      },
      "uninstall": true
    },
    "02-commands-list": {
      "url": "sftp://user:password@198.51.100.3/configs/commands.txt",
      "clear-config": true
    },
    "03-startup-file": {
      "url": "scp://user:password@198.51.100.4/configs/config.yaml",
      "clear-config": false
    },
    "04-connectivity-check": {
      "ping-hosts": [ "198.51.100.5", "localhost" ],
      "ping-count": 10,
      "ignore-result": true
    },
    "05-provisioning-script": {
      "url": "scp://user:password@198.51.100.5/scripts/script.sh",
      "timeout": 300
    }
  }
}
```

The main section is “ztp” and it must always be present in the ZTP JSON configuration file. Inside the main section may have the following configuration sections:

- image—to manage system images on the switch

- `commands-list`—to apply CLI commands in textual form
- `startup-file`—to apply YAML formatted configuration
- `connectivity-check`—to check a connectivity to predefined location

The configuration sections are processed by ZTP software in lexical order so to control the order of execution, a “xx-“ prefix to the section names is allowed (e.g., 02-commands-list).

Each configuration section of ZTP JSON includes some common parameters that can be used to influence its execution. The default value for a parameter is assumed when the parameter is not specified in the configuration section:

- `description`: Optional free-form text string used to describe a configuration section defined in the ZTP JSON configuration file.
- `ignore-result` (default: false):
  - `true`—ZTP service marks status as SUCCESS even if an error is encountered while processing this individual section.
  - `false`—ZTP service marks status as FAILED if an error is encountered while processing this individual section.
- `halt-on-failure` (default: false):
  - `true`—If configuration section result is FAILED, ZTP service stops and exits immediately marking ZTP status as FAILED. No other configuration sections are processed. User intervention is needed to restart ZTP.
  - `false`—ZTP service moves on to next configuration section.
- `restart-ztp-on-failure` (default: true):
  - `true`—ZTP procedure is restarted if the result of ZTP is FAILED after processing all of the configuration sections defined in the ZTP JSON configuration file. This happens up to 10 times.
  - `false`—ZTP service exits after processing all of the configuration sections defined in the ZTP JSON configuration file.

And the main “ztp” section has its own parameter:

- `restart-ztp-no-config` (default: false):
  - `true`—ZTP procedure is restarted if switch startup configuration file is not present after the completion of processing the configuration sections defined in the ZTP JSON configuration file. This happens up to 10 times.
  - `false`—ZTP service exits after processing all of the configuration sections defined in the ZTP JSON configuration file even if the Switch startup configuration file is not present.

ZTP service exits and marks the status as FAILED if any errors are encountered while parsing the ZTP JSON configuration file. It is encouraged to check for any JSON format correctness before rolling out the ZTP JSON configuration file for use. When processing a configuration section and provided data is found to be insufficient, it is marked as failed and ZTP moves on to the next section.

### 4.15.3.2 ZTP Configuration image

The *image* configuration section is used for image management on a switch. It can be used to install and uninstall system images.

Example config section to install a new image and boot into it:

```

"image": {
  "install": {
    "url": "http://198.51.100.2/images/nvos.bin",
    "skip-reboot ": false
  }
}
Example config section to uninstall a NVOS image from the Switch:
"image": {
  "uninstall": true
}

```

Following is the list of parameters supported in image configuration section and their brief description with a default values. The default value for a parameter is assumed when the parameter is not used:

- install—Used to install an image using URL.
  - url/dynamic-url—Specifies the URL string from where the system image file has to be downloaded in the form of url or dynamic-url object.
  - skip-reboot (default: false)—Specifies if a switch reboot operation is performed immediately after installing a new switch image. Reboot is skipped when set to true.
- uninstall (default: false)—Used to uninstall an existing image on the disk.
  - true—Uninstall the image from the second partition.
  - false—Do nothing.



uninstall is first processed followed by install if both are defined.

### 4.15.3.3 ZTP Configuration startup-file

The *startup-file* configuration section is used to apply the switch configuration in form of YAML file and apply the configuration. The YAML file format of the configuration is the same as switch startup configuration file. Here is an example of startup-file configuration section:

```

"startup-file": {
  "url": "http://198.51.100.3/startup.yaml",
  "clear-config": true,
  "save-config": true
}

```

Following is the list of parameters supported by the startup-file configuration section:

- url/dynamic-url—Define the URL string from where the startup.yaml file has to be downloaded in the form of url or dynamic-url object.
- clear-config (default: true)—Use this to specify if the existing configuration has to be cleared before loading the download startup.yaml file content . When set to true, ZTP replaces all current configuration with startup.yaml file content. When set to false it merges the configurations.
- save-config (default: false)—Use this to perform config save command after loading the downloaded startup.yaml file.



Setting save-config parameter to true will save the configuration and hence disable the ZTP at the end of the ZTP session.

### 4.15.3.4 ZTP Configuration commands-list

The *commands-list* configuration section is used to apply switch configuration in form of CLI text commands. The list of commands is provided in text file which should be downloaded and applied. Commands' file must contain only non-interactive commands, otherwise it will fail entire *commands-list* section. It is user responsibility to make sure commands are non-interactive, for example user should provide “force / -y / -n” or similar flags to the interactive CLI to prevent AYS (Are You Sure?) questions.



**Note:** Running “system reboot” command will restart the switch and will restart ZTP flow for current section from the beginning resulting in boot-loop. Keep in mind that when defining *commands-list* text file.

Example of *commands-list* configuration section:

```
"commands-list": {
  "url": "http://198.51.100.4/commands.txt",
  "clear-config": true,
  "save-config": true
}
```

Following is the list of parameters supported by the *commands-list* configuration section:

- *url/dynamic-url*—Define the URL string from where the *commands-list* file has to be downloaded in the form of *url* object or *dynamic-url* object.
- *clear-config* (default: true)—Use this to specify if the existing configuration has to be cleared before applying file content. If true, it applies empty (default) configuration prior executing commands.
- *save-config* (default: false)—Use this to perform a *config save* command after applying downloaded *commands-list*.



Setting *save-config* parameter to true will save the configuration and hence disable the ZTP at the end of the ZTP session. The same behavior applies to “nv config save” command specified in *commands-list* text file.

### 4.15.3.5 ZTP Configuration connectivity-check

The *connectivity-check* section is used to ping a remote host and verify if the switch is able to reach the remote host. It is possible to ping multiple hosts and the plugin result is marked as failed even if ping to one of the specified host fails.

```
"connectivity-check": {
  "ping-hosts": [ "198.51.100.5", "localhost" ]
}
```

Following is the list of objects supported by the *connectivity-check* section:

- *ping-hosts*—List of IPv4 hosts to ping.
- *ping6-hosts*—List of IPv6 hosts to ping.
- *retry-interval* (default: 5)—Specify a timeout, in seconds, before retrying ping to a host.

- `retry-count` (default: 12)—Stop ping to a host and move on to the next host specified in the list after retrying specified count of times.
- `ping-count` (default: 3)—Stop after sending *count* ECHO\_REQUEST packets. With *deadline* option, ping waits for *count* ECHO\_REPLY packets, until the timeout expires.
- `deadline` (default: N/A)—Specify a timeout, in seconds, before ping exits regardless of how many packets have been sent or received. In this case ping does not stop after *count* packet are sent, it waits either for *deadline* expire or until *count* probes are answered or for some error notification from network.
- `timeout` (default: N/A)—Time to wait for a response, in seconds. The option affects only timeout in absence of any responses, otherwise ping waits for two RTTs (Round Trip Time).

### 4.15.3.6 ZTP Configuration provisioning-script

The provisioning-script plugin is used to download a file and execute it.

Files must include a shebang (e.g., bash, python, etc.).

Files must contain only non-interactive commands.

A non-zero exit code during file execution will result in failure of the provisioning-script section.

```
// ztp.json
{
  "ztp": {
    "l-provisioning-script": {
      "url": "scp://user:password@server/path/to/script",
      "timeout": 300
    }
  }
}
```

The following objects are supported by the provisioning script:

- `url/dynamic-url`)—Defines the URL from which to download the file.
- `timeout`)—Specifies the timeout in seconds (default is 300). The script will terminate and fail if this time is exceeded. This object is optional.
- `Common objects`)—Includes options like `halt-on-failure`, `restart-ztp-on-failure`, and others.

### 4.15.3.7 ZTP Configuration nmx-commands-list

The *nmx-commands-list* configuration section is used to apply switch cluster and SDN configurations in form of CLI text commands on NVLink switch. The list of commands is provided in text file which should be downloaded and applied. Commands' file must contain only non-interactive commands, otherwise it will fail entire *nmx-commands-list* section. It is user responsibility to make sure commands are non-interactive, for example user should provide “force / -y / -n” or similar flags to the interactive CLI to prevent AYS (Are You Sure?) questions.



- Commands supported include “nv action \* cluster | sdn”
- Run these NMX commands after finishing all ZTP steps and enabling cluster
- Running “nv action start|stop cluster apps” command will start/stop specific cluster app and introduce uncertain effect to commands following
- Only allow one *nmx-commands-list* section in ZTP configuration file
- Result of execution of this section will not affect other ZTP steps

Example of nmx-commands-list configuration section:

```
"nmx-commands-list": {
  "url": "http://198.51.100.4/nmx-commands.txt",
}
```

Following is the list of parameters supported by the nmx-commands-list configuration section:

- url/dynamic-url—Define the URL string from where the nmx-commands-list file has to be downloaded in the form of url object or dynamic-url object.



Setting save-config parameter to true will save the configuration and hence disable the ZTP at the end of the ZTP session.

## 4.15.4 ZTP and OS Upgrade

Software upgrade from non-ZTP versions to ZTP versions and vice versa is supported. When upgrading from a non-ZTP version, ZTP is disabled because ZTP is always assumed to start with an empty configuration.

## 4.15.5 Example Configurations

### 4.15.5.1 DHCPv4 Configuration

The following is a configuration example for ISC DHCPv4 server:

```
host master {
  hardware ethernet 12:34:56:AB:CD:EF;
  fixed-address 192.51.100.201;
  option bootfile-name "scp://<user>:<password>@192.51.100.8/ztp.json";
}
```

### 4.15.5.2 DHCPv6 Configuration

The following is a DHCPv6 configuration example:

```
host master {
  .....
  option dhcp6.bootfile-url "http://[2001:db8::8]/ztp.json";
}
```

### 4.15.5.3 ZTP Configuration File Example

```
{
  "ztp": {
    "01-image": {
      "uninstall": true
    },
    "02-commands-list": {
      "url": "sftp://user:password@198.51.100.3/configs/commands.txt",
      "clear-config": "true"
    },
    "03-startup-file": {
```

```

    "url": "scp://user:password@198.51.100.4/configs/config.yaml"
  },
  "04-connectivity-check": {
    "ping-hosts": [ "198.51.100.5", "localhost" ],
  }
}
}

```

#### 4.15.5.4 Commands List Example

```

# Disable password hardening
nv set system security password-hardening state disabled

# Configure log rotation rules
nv set system log rotation frequency weekly

# Configure system message
nv set system message pre-login Hello

# Applying config
nv config apply -y

```

#### 4.15.5.5 YAML Formatted File Configuration

```

- set:
  system:
  log:
    rotation:
      size: 30.0
  ntp:
    dhcp: disabled
    server:
      localntpserver: {}
      ua.pool.ntp.org:
        association-type: pool
  security:
    password-hardening:
      state: disabled

```

#### 4.15.5.6 Provisioning Script Example

```

#!/bin/python
print("Hello world from Provisioning Script!")

```

### 4.15.6 Zero-Touch Provisioning Commands

- [Zero-Touch Provisioning Commands](#)

### 4.15.7 Zero-Touch Provisioning Commands

- [4.15.7.1 nv show system ztp](#)
- [4.15.7.2 nv set system ztp config-save](#)
- [4.15.7.3 nv action system ztp](#)

#### 4.15.7.1 nv show system ztp

	nv show system ztp Shows global ZTP status and the status for each section.	
Syntax Description	N/A	
Default	N/A	

History	25.02.1884
Example	<pre> admin@nvos:~\$ nv showsystem ztp operational    applied ----- runtime    31s service    inactive source     dhcp-opt67 (eth0) state      enabled status     success config-save disabled    disabled  ZTPstages ===== Stage      Exit Status Ignore Result Runtime Status ----- 02-startup-file success  no   09s  success 03-commands-list success  no   18s  success  admin@nvos:~\$ nv showsystem ztp operational    applied ----- runtime    05m 08s service     active-discovery state       enabled status      not-started config-save disabled    disabled  ZTPstages ===== No Data </pre>
REST API	GET https://<ip>/nvue_v1/system/ztp
Related Commands	nv set system ztp config-save
Notes	

#### 4.15.7.2 nv set system ztp config-save

	nv set system ztp config-save {enabled   disabled} Setting the ZTP configuration.	
Syntax Description	enabled, disabled	Enables or disables “nv config save”
Default	Disabled	
Configuration Mode	config	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv setsystem ztp config-save enabled admin@nvos:~\$ nv config apply admin@nvos:~\$ nv config save </pre>	
REST API	PATCH https://<ip>/nvue_v1/system/ztp	
Related Commands	nv show system ztp	
Notes	When ZTP is active, “nv config save” is suppressed because it may interfere with ZTP operation. Therefore, after running “nv set system ztp config-save enabled” and “nv config apply”, if “nv config save” is performed, then ZTP is disabled as a consequence of the database save.	

### 4.15.7.3 nv action system ztp

	nv action {abort   run} system ztp [force] ZTP action commands to interact with ZTP service.	
Syntax Description	abort	Abort ZTP session. This command is used to interrupt ongoing ZTP session.
	run	Rerun ZTP flow. Use this command to manually restart a new ZTP session from scratch or from when it failed or aborted.
	force	This option is used to skip AYS dialog. By default, each action command asks the user to confirm the execution of the specified command.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv action abort system ztp The operation will perform abort of the ZTP. Type [y] to perform abort of the ZTP. Type [N] to cancel an action.  Do you want to <b>continue?</b> [y/N] y Action executing ... Aborting ZTP session Action executing ... Action succeeded  admin@nvos:~\$ nv action run system ztp The operation will perform rerun of the ZTP. Type [y] to perform rerun of the ZTP. Type [N] to cancel an action.  Do you want to <b>continue?</b> [y/N] y Action executing ... Rerunning ZTP session Action executing ... Action succeeded           </pre>	
REST API	POST https://<ip>/nvue_v1/system/ztp	
Related Commands	nv show system ztp	
Notes	<ul style="list-style-type: none"> <li>• <b>nv action abort system ztp</b> command does not disable ZTP, in order to manually configure the device, please use "nv config save" first to fully disable ZTP process.</li> <li>• <b>nv action run system ztp</b> command will overwrite any currently saved configuration file, make sure to backup the file in case required.</li> </ul>	

---

## 5 Chassis Management

The chassis manager provides the user access to the following information:

Accessible Parameters	Description
switch temperatures	Displays system's temperature
switch leakage	Displays leakage sensors' status
fan unit	Displays system fans' status
power unit	Displays system power consumers
Flash memory	Displays information about system memory utilization.

Additionally, it monitors:

- AC power to the PSUs
- DC power out from the PSUs
- Chassis failures
- Leakage detection from the switch tray

### 5.1 System Health Monitor

The system health monitor scans the system to decide whether or not the system is healthy. When the monitor discovers that one of the system's modules (leaf, spine, fan, or power supply) is in an unhealthy state or has returned from an unhealthy state, it notifies the users through the following methods:

- System logs—accessible to the user at any time as they are saved permanently on the system
- Status LEDs—changed by the system health monitor when an error is found in the system and is resolved

### 5.2 Leakage Sensors

The system will have 6 leakage sensors located in different locations. Once sensors detect leakage, NVOS will publish an event immediately and update system health accordingly. User will be able to see on which sensor it was detected using the 'nv show platform environment leakage' show command. Once leakage is redeemed, the sensors can re-arm automatically without the need to clear the sensors' leak state and it takes up to 1 hour.

### 5.3 Chassis Management Commands

- [Chassis Information and Inventory](#)
- [Environment](#)
- [Software](#)
- [Transceiver](#)

## 5.4 Chassis Information and Inventory

NVOS provides commands to display general platform information and inventory details.

### 5.4.1 Displaying Cable Cartridge (CBC) EEPROM

Cable cartridge is a part of the rack and has its own EEPROM. This EEPROM contains an information about relevant cartridge: its serial and part number, slot and tray index, and manufacturing date. This information can be listed by running the [nv show platform cable-cartridge](#) command.

### 5.4.2 Chassis Information and Inventory Commands

- [Chassis Information and Inventory Commands](#)

### 5.4.3 Chassis Information and Inventory Commands

- [5.4.3.1 nv show platform](#)
- [5.4.3.2 nv show platform chassis-location](#)
- [5.4.3.3 nv show platform inventory](#)
- [5.4.3.4 nv action reset platform bmc-password](#)
- [5.4.3.5 nv show platform cable-cartridge](#)
- [5.4.3.6 nv show platform cable-cartridge name](#)

#### 5.4.3.1 nv show platform

	<b>nv show platform</b> Displays the types of data available under more specific platform commands.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform operational ----- system-mac      9C:63:C0:72:B2:12 manufacturer    Nvidia product-name    N5110_LD cpu              x86_64 AMD EPYC 3151 4-Core Processor x8 memory          16320876 kB disk-size       74.5G port-layout     18 x 400G-OSFP part-number     692-9K36F-00MV-JS0 serial-number   MT2416X02630 asic-model      Quantum3 system-uuid     317ab89a-032b-11ef-8000-b0cf0e0b5900                 </pre>	
REST API	GET https://<ip>/nvue_v1/platform	
Related Commands	<a href="#">nv show platform environment</a> <a href="#">nv show platform firmware</a> <a href="#">nv show platform software</a>	
Notes		

### 5.4.3.2 nv show platform chassis-location

	nv show platform chassis-location Display chassis location information.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show platform chassis-location operational ----- tray-index      0 slot-number     5 chassis-sn      999WWYY123456 topology-id     GB200 NVL36</pre>	
Related Commands		
Notes		

### 5.4.3.3 nv show platform inventory

	nv show platform inventory {<inventory-id>} Display the status of all platform components. Includes the following fields: hw-version, model, serial, state and type.	
Syntax Description	inventory-id	Display the status of a single platform component (with the same fields as the general command).
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show platform inventory    Hw version  Model          Serial          State  Type ----- FAN1/1      N/A            N/A             N/A    ok    fan FAN1/2      N/A            N/A             N/A    ok    fan FAN2/1      N/A            N/A             N/A    ok    fan FAN2/2      N/A            N/A             N/A    ok    fan FAN3/1      N/A            N/A             N/A    ok    fan FAN3/2      N/A            N/A             N/A    ok    fan FAN4/1      N/A            N/A             N/A    ok    fan FAN4/2      N/A            N/A             N/A    ok    fan FAN5/1      N/A            N/A             N/A    ok    fan FAN5/2      N/A            N/A             N/A    ok    fan FAN6/1      N/A            N/A             N/A    ok    fan FAN6/2      N/A            N/A             N/A    ok    fan SWITCH A2      692-9K36F-00MV-JS0 MT2416X02630  ok    switch  admin@nvos:~\$ nv show platform inventory SWITCH operational ----- state           ok hardware-version A2 model           692-9K36F-00MV-JS0 serial          MT2416X02630 type            switch</pre>	
REST API	GET https://<ip>/nvue_v1/platform/inventroy/ GET https://<ip>/nvue_v1/platform/inventroy/{inventroy-id}	
Related Commands	<a href="#">nv show platform</a>	
Notes		

### 5.4.3.4 nv action reset platform bmc-password

	nv action reset platform bmc-password Reset BMC root user password.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action reset platform bmc-password Action executing ...  The password has been reset to the default BMC password. To obtain the default password, please refer to the UM command description. For your security, it is highly recommended to update this temporary password to a new, strong password as soon as possible.  Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/platform/bmc-password	
Related Commands	<a href="#">nv show platform firmware BMC</a>	
Notes	This command resets the BMC root user password to 'OpenBmcTempPass!'. For security purposes, it is highly recommended to update this temporary password to a new, strong password immediately after the reset.	

### 5.4.3.5 nv show platform cable-cartridge

	nv show platform cable-cartridge Display the table with cable cartridges, their slot and tray ID, S/N, P/N and manufacturing date.	
Syntax Description	N/A	
Default	N/A	
History	25.02.2141	
Example	<pre>admin@nvos:~\$ nv show platform cable-cartridge Name      Slot ID  Tray ID  Serial          Part Number      Manufacturing Date ----- cartridge1 3        0        1783224040338  755-24972-0003-000 08/09/24 - 07:06:00 cartridge2 3        0        1783224040357  755-24972-0003-000 08/09/24 - 08:11:00 cartridge3 3        0        1783224040358  755-24972-0003-000 08/09/24 - 09:44:00 cartridge4 3        0        1783224040359  755-24972-0003-000 08/09/24 - 09:08:00</pre>	
REST API	GET https://<ip>/nvue_v1/platform/cable-cartridge	
Related Commands	nv show platform cable-cartridge <cartridge-name>	
Notes		

### 5.4.3.6 nv show platform cable-cartridge name

	nv show platform cable-cartridge <cartridge-name> Display specified cable-cartridge information.	
--	---	--

Syntax Description	cartridge-name	Cable cartridge name, listed in the <a href="#">nv show platform cable-cartridge</a>
Default	N/A	
History	25.02.2141	
Example	<pre>admin@nvos:~\$ nv show platform cable-cartridge cartridge1 operational ----- serial-number    1783224040338 part-number     755-24972-0003-000 slot-id         3 tray-id         0 manufacture-date 08/09/24 - 07:06:00</pre>	
REST API	GET https://<ip>/nvue_v1/platform/cable-cartridge/{cartridge-name}	
Related Commands	nv show platform cable-cartridge	
Notes		

## 5.5 Environment

NVOS includes robust features for monitoring and managing the platform's environmental conditions, ensuring optimal performance and system reliability. These features provide detailed insights into critical hardware components and allow users to take control of specific environmental aspects.

### 5.5.1 Key Functionalities

- Fan Monitoring: Retrieve real-time status and performance of system fans.
- Temperature Monitoring: Check temperature readings from various sensors.
- Voltage Monitoring: Inspect voltage levels and sensor data.
- Leakage Detection: Identify potential leakage scenarios.
- LED Control: Manage and monitor platform LEDs.

### 5.5.2 Environment Commands

- [Environment Commands](#)

### 5.5.3 Environment Commands

- [5.5.3.1 nv show platform environment](#)
- [5.5.3.2 nv show platform environment fan](#)
- [5.5.3.3 nv show platform environment led](#)
- [5.5.3.4 nv show platform environment temperature](#)
- [5.5.3.5 nv show platform environment voltage](#)
- [5.5.3.6 nv show platform environment leakage](#)
- [5.5.3.7 nv action turn-on/turn-off platform environment led UID](#)

### 5.5.3.1 nv show platform environment

	<b>nv show platform environment</b> Displays the types of data available under more specific platform environment commands.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform environment Name                                     Type      State ----- ASIC                                     temperature ok Ambient-Fan-Side-Temp                   temperature ok CPU-Pack-Temp                            temperature ok FAN1                                      led        off FAN1/1                                   fan        ok FAN1/2                                   fan        ok FAN2                                      led        off FAN2/1                                   fan        ok FAN2/2                                   fan        ok FAN3                                      led        off FAN3/1                                   fan        ok FAN3/2                                   fan        ok FAN4                                      led        off FAN4/1                                   fan        ok FAN4/2                                   fan        ok FAN5                                      led        off FAN5/1                                   fan        ok FAN5/2                                   fan        ok FAN6                                      led        off FAN6/1                                   fan        ok FAN6/2                                   fan        ok HSC-VinDC-In                            voltage    ok HSC-VinDC-Out                           voltage    ok PDB-1-Conv-In-1                         voltage    ok PDB-1-Conv-Out-1                       voltage    ok PDB-2-Conv-In-1                         voltage    ok PDB-2-Conv-Out-1                       voltage    ok PDB-3-Conv-In-1                         voltage    ok PDB-3-Conv-Out-1                       voltage    ok PDB-4-Conv-In-1                         voltage    ok PDB-4-Conv-Out-1                       voltage    ok PMIC-1-12V-VDD-ASIC1-In-1              voltage    ok PMIC-1-ASIC1-VDD-Out-1                  voltage    ok PMIC-2-12V-HVDD-DVDD-ASIC1-In-1        voltage    ok PMIC-2-ASIC1-DVDD-PL0-Out-2             voltage    ok PMIC-2-ASIC1-HVDD-PL0-Out-1             voltage    ok PMIC-3-12V-HVDD-DVDD-ASIC1-In-1        voltage    ok PMIC-3-ASIC1-DVDD-PL1-Out-2             voltage    ok PMIC-3-ASIC1-HVDD-PL1-Out-1             voltage    ok PMIC-4-12V-VDD-ASIC2-In-1              voltage    ok PMIC-4-ASIC2-VDD-Out-1                  voltage    ok PMIC-5-12V-HVDD-DVDD-ASIC2-In-1        voltage    ok PMIC-5-ASIC2-DVDD-PL0-Out-2             voltage    ok PMIC-5-ASIC2-HVDD-PL0-Out-1             voltage    ok PMIC-6-12V-HVDD-DVDD-ASIC2-In-1        voltage    ok PMIC-6-ASIC2-DVDD-PL1-Out-2             voltage    ok PMIC-6-ASIC2-HVDD-PL1-Out-1             voltage    ok PMIC-7-12V-MAIN-In-1                   voltage    ok PMIC-7-CEX-VDD-Out-1                   voltage    ok PMIC-8-COMEX-VDD-MEM-In-1               voltage    ok PMIC-8-COMEX-VDD-MEM-Out-1              voltage    ok SODIMM-1-Temp                           temperature ok STATUS                                    led        off UID                                       led        off </pre>	
REST API	GET https://<ip>/nvue_v1/platform/environment	
Related Commands	<a href="#">nv show platform environment fan</a> <a href="#">nv show platform environment led</a> <a href="#">nv show platform environment temperature</a>	
Notes		

### 5.5.3.2 nv show platform environment fan

	nv show platform environment fan {fan-id} Displays the maximum, minimum, current speed and state for one or all fans in the system.	
Syntax Description	fan-id	Name of the fan whose status to display. If not entered, all fans are displayed.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform environment fan Name      Fan State  Current Speed (RPM)  Max Speed  Min Speed  Fan Direction ----- FAN1/1   ok         23045                 33000     6000      B2F FAN1/2   ok         21634                 33000     6000      B2F FAN2/1   ok         23557                 33000     6000      B2F FAN2/2   ok         22085                 33000     6000      B2F FAN3/1   ok         22555                 33000     6000      B2F FAN3/2   ok         22085                 33000     6000      B2F FAN4/1   ok         22555                 33000     6000      B2F FAN4/2   ok         22085                 33000     6000      B2F FAN5/1   ok         22555                 33000     6000      B2F FAN5/2   ok         22085                 33000     6000      B2F FAN6/1   ok         22555                 33000     6000      B2F FAN6/2   ok         21634                 33000     6000      B2F  admin@nvos:~\$ nv show platform environment fan FAN1/1 operational ----- state      ok current-speed  22555 min-speed  6000 max-speed  33000 direction  B2F           </pre>	
REST API	GET https://<ip>/nvue_v1/platform/environment/fan GET https://<ip>/nvue_v1/platform/environment/fan/{fan-id}	
Related Commands	<a href="#">nv show platform environment</a>	
Notes		

### 5.5.3.3 nv show platform environment led

	nv show platform environment led {led-id} Displays the status of one or all LEDs in the system.	
Syntax Description	led-id	Name of the LED whose status to display. If not entered, all LEDs are displayed.
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show platform environment led LED Name  LED Color -----  - FAN1     off FAN2     off FAN3     off FAN4     off FAN5     off FAN6     off STATUS   green UID      blue</pre> <pre>admin@nvos:~\$ nv show platform environment led UID       operational ----- color  blue</pre>
REST API	GET https://<ip>/nvue_v1/platform/environment/led GET https://<ip>/nvue_v1/platform/environment/led/{led-id}
Related Commands	<a href="#">nv show platform environment</a>
Notes	

### 5.5.3.4 nv show platform environment temperature

	nv show platform environment temperature {sensor-id} Shows platform temperature and displays temperature information from different platform sensors.	
Syntax Description	sensor-id	Name of the sensor whose data to display. If not entered, all sensors are displayed.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show platform environment temperature Name          Cur Temp (°C)  Crit Temp  Max Temp  Min Temp  State ----- ASIC          47.00         105.00    120.00 Ambient-Fan-Side-Temp  33.75 CPU-Pack-Temp 70.00         100.00    95.00 SODIMM-1-Temp 36.50         95.00     85.00 ok ok ok ok</pre> <pre>admin@nvos:~\$ nv show platform environment temperature CPU-Pack-Temp       operational ----- state  ok current 46.12 max     95.00 crit    100.00</pre>	
REST API	GET https://<ip>/nvue_v1/platform/environment/temperature GET https://<ip>/nvue_v1/platform/environment/temperature/{sensor-id}	
Related Commands	<a href="#">nv show platform environment</a>	
Notes	Note the quotes needed for sensor ID containing a space.	

### 5.5.3.5 nv show platform environment voltage

	nv show platform environment voltage Displays a table with all voltage sensors located on the chassis.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform environment voltage Name                                     Actual (V)  Maximum (V)  Minimum (V)  State ----- HSC-VinDC-In                            54.85       89.53        0.26         ok HSC-VinDC-Out                            55.07       89.53        0.52         ok PDB-1-Conv-In-1                          53.75       64.00       35.56        ok PDB-1-Conv-Out-1                         13.43       16.00        8.20         ok PDB-2-Conv-In-1                          53.88       64.00       35.56        ok PDB-2-Conv-Out-1                         13.41       16.00        8.20         ok PDB-3-Conv-In-1                          53.81       64.00       35.56        ok PDB-3-Conv-Out-1                         13.40       16.00        8.20         ok PDB-4-Conv-In-1                          53.81       64.00       35.56        ok PDB-4-Conv-Out-1                         13.44       16.00        8.20         ok PMIC-1-12V-VDD-ASIC1-In-1                13.44       16.00        0.03         ok PMIC-1-ASIC1-VDD-Out-1                    1.43        1.64         1.44         ok PMIC-2-12V-HVDD-DVDD-ASIC1-In-1          13.44       16.00        0.03         ok PMIC-2-ASIC1-DVDD-PL0-Out-2               1.67        1.78         1.58         ok PMIC-2-ASIC1-HVDD-PL0-Out-1               1.20        1.35         1.05         ok PMIC-3-12V-HVDD-DVDD-ASIC1-In-1          13.38       16.00        0.03         ok PMIC-3-ASIC1-DVDD-PL1-Out-2               1.68        1.78         1.58         ok PMIC-3-ASIC1-HVDD-PL1-Out-1               1.20        1.35         1.05         ok PMIC-4-12V-VDD-ASIC2-In-1                13.12       16.00        0.03         ok PMIC-4-ASIC2-VDD-Out-1                    1.43        1.64         1.44         ok PMIC-5-12V-HVDD-DVDD-ASIC2-In-1          13.16       16.00        0.03         ok PMIC-5-ASIC2-DVDD-PL0-Out-2               1.68        1.78         1.58         ok PMIC-5-ASIC2-HVDD-PL0-Out-1               1.20        1.35         1.05         ok PMIC-6-12V-HVDD-DVDD-ASIC2-In-1          13.09       16.00        0.03         ok PMIC-6-ASIC2-DVDD-PL1-Out-2               1.70        1.78         1.58         ok PMIC-6-ASIC2-HVDD-PL1-Out-1               1.20        1.35         1.05         ok PMIC-7-12V-MAIN-In-1                     13.12       16.00        0.03         ok PMIC-7-CEX-VDD-Out-1                      1.06        1.26         0.64         ok PMIC-8-COMEX-VDD-MEM-In-1                 13.00       17.00        0.03         ok PMIC-8-COMEX-VDD-MEM-Out-1                1.20        1.29         0.90         ok </pre>	
REST API	GET https://<ip>/nvue_v1/platform/environment/voltage	
Related Commands	nv show platform environment voltage	
Notes		

### 5.5.3.6 nv show platform environment leakage

	nv show platform environment leakage Display leakage sensors' status.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ \$ nv show platform enviroment leakage Name                                     State ----- leakage1                                ok leakage2                                leak leakage3                                leak leakage4                                ok leakage1_rope                            leak leakage2_rope                            leak </pre>	
REST API	GET https://<ip>/nvue_v1/platform/environment/leakage	
Related Commands	nv show platform environment	

Notes	
-------	--

### 5.5.3.7 nv action turn-on/turn-off platform environment led UID

	nv action turn-on platform environment led UID nv action turn-off platform environment led UID Turns on/off the LED UID.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action turn-on platform environment led UID</pre>	
REST API	POST https://<ip>/nvue_v1/platform/environment/led/UID	
Related Commands		
Notes		

## 5.6 Software

NVOS provides tools to display the software packages installed on the system.

### 5.6.1 Software Commands

- [Software Commands](#)

### 5.6.2 Software Commands

#### 5.6.2.1 nv show platform software installed

	nv show platform software installed Displays the software packages installed in the system.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	

Example	<pre> admin@nvos:~\$ nv show platform software installed --w Installed Software      Description                                     Package Version ----- acl                    access control list - utilities                 acl   2.2.53-10 adduser                add and remove users and groups                adduser   3.118 apparmor              user-space parser utility for AppArmor         apparmor   2.13.6-10 apt                   commandline package manager                   apt   2.2.4 apt-transport-https   transitional package for https support        apt-transport- https 2.2.4 audisp-tacplus        audisp module for TACACS+ accounting          audisp-tacplus   1.0.2 auditd                User space tools for security auditing        auditd   1:3.0-2 base-files            Debian base system miscellaneous files        base-files   11.1+deb11u3 base-passwd           Debian base system master password and group files   3.5.51 ... </pre>
REST API	<pre> GET https://&lt;ip&gt;/nvue_v1/platform/software/installed GET https://&lt;ip&gt;/nvue_v1/platform/software/installed/{installed-id} </pre>
Related Commands	<a href="#">nv show platform</a>
Notes	

## 5.7 Transceiver

NVOS offers advanced features for managing and monitoring platform transceivers, ensuring seamless operation and optimal network performance. These features provide users with comprehensive tools for overseeing transceiver status, resetting devices, and managing firmware updates.

### 5.7.1 Key Functionalities

- **Transceiver Status Monitoring:** View detailed information about installed transceivers, including their operational state and specifications.
- **Transceiver Reset:** Perform targeted resets on specific transceivers to resolve issues or refresh their operational state.
- **Firmware Management:**
  - Install firmware files to update transceivers with the latest features and fixes
  - View the current firmware version of transceivers
  - List and inspect available firmware files

### 5.7.2 How to Install Transceiver Firmware

Follow the steps below for installing the transceiver firmware.

1. Fetch the firmware file.

```

admin@nvos:~$ nv action fetch platform firmware transceiver scp://username[:password]@hostname/path/
46_120_10010_dev_signed.bin
Password:
Action executing ...
File fetched successfully

```

```
Action succeeded
```

## 2. Install a transceiver firmware.

```
admin@nvos:~$ nv action install platform transceiver sw1 firmware files 46_120_10010_dev_signed.bin
Action executing ... 100%
Installed FW version: 46.120.10010
Action succeeded
```

## 5.7.3 Transceiver Commands

- [Transceiver Commands](#)

## 5.7.4 Transceiver Commands

- [5.7.4.1 nv show platform transceiver](#)
- [5.7.4.2 nv action reset platform transceiver id](#)
- [5.7.4.3 nv action install platform transceiver firmware files](#)
- [5.7.4.4 nv show platform transceiver firmware](#)
- [5.7.4.5 nv show platform transceiver firmware files](#)
- [5.7.4.6 nv show platform transceiver firmware files name](#)

### 5.7.4.1 nv show platform transceiver

	nv show platform transceiver {<transceiver-id>} Display the status of all transceivers. Includes fields such as: cable-type, cable-length, vendor-rev and identifier.	
Syntax Description	transceiver-id	Display the status of a single transceiver (with the same fields as the general command).
Default	N/A	
History	25.02.1884	

## Example

```
admin@nvos:~$ nv show platform transceiver
sw1:
  cable-type           : Optical module
  supported-cable-length : 30m OM3,50m OM4,50m OM5
  diagnostics-status    : Diagnostic Data Available
  status               : Inserted
  error-status         : N/A
  vendor-data-code     : 2022-08-19
  identifier           : OSFP 8X Pluggable Transceiver
  vendor-rev           : A3
  vendor-name          : NVIDIA
  vendor-pn            : MMA4Z00-NS
  vendor-sn            : MT2234FT11534

sw2:
  cable-type           : Copper cable
  cable-length         : 1.0
  diagnostics-status    : No Diagnostic Data Available. Module is not DDMI capable
  status               : Inserted
  error-status         : N/A
  vendor-data-code     : 2022-04-15
  identifier           : OSFP 8X Pluggable Transceiver
  vendor-rev           : A2
  vendor-name          : NVIDIA
  vendor-pn            : MCP7Y50-N001
  vendor-sn            : MT2216VS02521

sw3:
  cable-type           : Copper cable
  cable-length         : 1.0
  diagnostics-status    : No Diagnostic Data Available. Module is not DDMI capable
  status               : Inserted
  error-status         : N/A
  vendor-data-code     : 2021-12-22
  identifier           : OSFP 8X Pluggable Transceiver
  vendor-rev           : A1
  vendor-name          : NVIDIA
  vendor-pn            : MCP7Y00-N001
  vendor-sn            : MT2152VS04096

sw4:
  cable-type           : Copper cable
  cable-length         : 1.5
  diagnostics-status    : No Diagnostic Data Available. Module is not DDMI capable
  status               : Inserted
  error-status         : N/A
  vendor-data-code     : 2021-09-14
  identifier           : OSFP 8X Pluggable Transceiver
  vendor-rev           : A2
  vendor-name          : NVIDIA
  vendor-pn            : MCP7Y60-H01A
  vendor-sn            : MT2138VS01296

sw5:
  cable-type           : Copper cable
  cable-length         : 2.0
  diagnostics-status    : No Diagnostic Data Available. Module is not DDMI capable
  status               : Inserted
  error-status         : N/A
  vendor-data-code     : 2022-07-17
  identifier           : OSFP 8X Pluggable Transceiver
  vendor-rev           : A3
  vendor-name          : NVIDIA
  vendor-pn            : MCP7Y60-H002
  vendor-sn            : MT2233VS00911

sw6:
  diagnostics-status    : Non present module
  status               : Removed
  error-status         : N/A

...

sw30:
  diagnostics-status    : Non present module
  status               : Removed
  error-status         : N/A

sw31:
  cable-type           : Copper cable
  cable-length         : 0.5
  diagnostics-status    : No Diagnostic Data Available. Module is not DDMI capable
  status               : Inserted
  error-status         : N/A
  vendor-data-code     : 2021-08-31
  identifier           : OSFP 8X Pluggable Transceiver
  vendor-rev           : A3
  vendor-name          : NVIDIA
  vendor-pn            : MCP4Y10-N00A
  vendor-sn            : MT2134VS00903

sw32:
  cable-type           : Copper cable
  cable-length         : 0.5
  diagnostics-status    : No Diagnostic Data Available. Module is not DDMI capable
  status               : Inserted
  error-status         : N/A
  vendor-data-code     : 2021-08-31
  identifier           : OSFP 8X Pluggable Transceiver
```

```
vendor-rev      : A3
vendor-name     : NVIDIA
vendor-pn      : MCP4Y10-N00A
vendor-sn      : MT2134VS00903
```

```
admin@nvos:~$ nv show platform transceiver sw31
cable-type      : Copper cable
cable-length    : 0.5
diagnostics-status : No Diagnostic Data Available. Module is not DDMI capable
status         : Inserted
error-status    : N/A
vendor-data-code : 2021-08-31
identifier      : OSFP 8X Pluggable Transceiver
vendor-rev     : A3
vendor-name    : NVIDIA
vendor-pn     : MCP4Y10-N00A
vendor-sn     : MT2134VS00903
```

```
admin@nvos:~$ nv show platform transceiver sw1
sw1:
cable-type      : Optical module
supported-cable-length : 30m OM3,50m OM4,50m OM5
diagnostics-status : Diagnostic Data Available
status         : Inserted
error-status    : N/A
vendor-data-code : 2022-08-19
identifier      : OSFP 8X Pluggable Transceiver
vendor-rev     : A3
vendor-name    : NVIDIA
vendor-pn     : MMA4Z00-NS
vendor-sn     : MT2234FT11534
temperature:
temperature    : 57.00 C
high-alarm-threshold: 80.00 C
low-alarm-threshold : -10.00 C
voltage:
voltage       : 3.18 V
high-alarm-threshold: 3.50 V
low-alarm-threshold : 3.10 V
mod-fw-fault  : False
dp-fw-fault   : False
channel:
channel-1:
rx-power:
power         : 1.68 mW / 2.25 dBm
high-alarm-thresh: 5.00 dBm
low-alarm-thresh : -6.00 dBm
tx-power:
power         : 1.58 mW / 1.99 dBm
high-alarm-thresh: 5.00 dBm
low-alarm-thresh : -2.01 dBm
tx-bias-current:
current      : 54.60 mA
high-alarm-thresh: 145.00 mA
low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eg-fault: False
tx-cdr-lol   : False
tx-los       : False
tx-fault     : False
channel-2:
rx-power:
power         : 1.58 mW / 1.99 dBm
high-alarm-thresh: 5.00 dBm
low-alarm-thresh : -6.00 dBm
tx-power:
power         : 1.58 mW / 1.99 dBm
high-alarm-thresh: 5.00 dBm
low-alarm-thresh : -2.01 dBm
tx-bias-current:
current      : 48.38 mA
high-alarm-thresh: 145.00 mA
low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eg-fault: False
tx-cdr-lol   : False
tx-los       : False
tx-fault     : False
channel-3:
rx-power:
power         : 1.54 mW / 1.88 dBm
high-alarm-thresh: 5.00 dBm
low-alarm-thresh : -6.00 dBm
tx-power:
power         : 1.58 mW / 1.99 dBm
high-alarm-thresh: 5.00 dBm
low-alarm-thresh : -2.01 dBm
tx-bias-current:
current      : 50.83 mA
high-alarm-thresh: 145.00 mA
low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eg-fault: False
tx-cdr-lol   : False
```

```

tx-los      : False
tx-fault    : False

channel-4:
rx-power:
  power      : 1.74 mW / 2.41 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -6.00 dBm
tx-power:
  power      : 1.58 mW / 1.99 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -2.01 dBm
tx-bias-current:
  current     : 55.99 mA
  high-alarm-thresh: 145.00 mA
  low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eq-fault: False
tx-cdr-lol   : False
tx-los       : False
tx-fault     : False

channel-5:
rx-power:
  power      : 1.55 mW / 1.90 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -6.00 dBm
tx-power:
  power      : 1.59 mW / 2.01 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -2.01 dBm
tx-bias-current:
  current     : 62.47 mA
  high-alarm-thresh: 145.00 mA
  low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eq-fault: False
tx-cdr-lol   : False
tx-los       : False
tx-fault     : False

channel-6:
rx-power:
  power      : 1.64 mW / 2.15 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -6.00 dBm
tx-power:
  power      : 1.57 mW / 1.96 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -2.01 dBm
tx-bias-current:
  current     : 54.89 mA
  high-alarm-thresh: 145.00 mA
  low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eq-fault: False
tx-cdr-lol   : False
tx-los       : False
tx-fault     : False

channel-7:
rx-power:
  power      : 1.46 mW / 1.64 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -6.00 dBm
tx-power:
  power      : 1.61 mW / 2.07 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -2.01 dBm
tx-bias-current:
  current     : 53.54 mA
  high-alarm-thresh: 145.00 mA
  low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eq-fault: False
tx-cdr-lol   : False
tx-los       : False
tx-fault     : False

channel-8:
rx-power:
  power      : 1.46 mW / 1.64 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -6.00 dBm
tx-power:
  power      : 1.60 mW / 2.04 dBm
  high-alarm-thresh: 5.00 dBm
  low-alarm-thresh : -2.01 dBm
tx-bias-current:
  current     : 60.24 mA
  high-alarm-thresh: 145.00 mA
  low-alarm-thresh : 5.00 mA
rx-cdr-lol   : False
rx-los       : False
tx-ad-eq-fault: False
tx-cdr-lol   : False
tx-los       : False
tx-fault     : False

```

REST API	GET https://<ip>/nvue_v1/platform/transceiver/ GET https://<ip>/nvue_v1/platform/transceiver/{transceiver-id}
Related Commands	<a href="#">nv show platform</a>
Notes	

### 5.7.4.2 nv action reset platform transceiver id

	nv action reset platform transceiver <transceiver-id> Reset specific transceiver module	
Syntax Description	transceiver-id	The name of the transceiver
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action reset platform transceiver sw1 Action executing ... Resetting module sw1 ... OK Action succeeded  admin@nvos:~\$ nv action reset platform transceiver sw1 Action executing ... Resetting module sw1 ... Failed Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/platform/transceiver/{transceiver-id}	
Related Commands	nv show platform transceiver <transceiver-id> firmware	
Notes		

### 5.7.4.3 nv action install platform transceiver firmware files

	nv action install platform transceiver <transceiver-id> firmware files <file-name> Install firmware image file on the specific transceiver module.	
Syntax Description	transceiver-id	The name of the transceiver
	file-name	The name of the FW image file
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action install platform transceiver sw1 firmware files sec_issu_46_120_10010_dev_signed.bin Action executing ... 100% Installed FW version: 46.120.10010 Action succeeded  admin@nvos:~\$ nv action install platform transceiver sw5 firmware files sec_issu_46_120_10011_dev_signed.bin Error: Action failed with the following issue: FW update is not supported for this module: no FW mgmt data is found</pre>	
REST API	POST https://<ip>/nvue_v1/platform/transceiver/{transceiver-id}/firmware/files/{file-name}	
Related Commands	nv show platform transceiver <transceiver-id> firmware nv show platform transceiver <transceiver-id> firmware files	
Notes	Burning new firmware version is not supported for copper cables.	

### 5.7.4.4 nv show platform transceiver firmware

	nv show platform transceiver <transceiver-id> firmware Display firmware information for specific transceiver module.	
Syntax Description	transceiver-id	The name of the transceiver
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform transceiver sw1 firmware ----- operational      applied ----- actual-firmware      46.140.1013 fw-upgrade-status    N/A fw-upgrade-error-msg N/A  admin@nvos:~\$ nv show platform transceiver sw2 firmware ----- operational      applied ----- actual-firmware      47.120.10013 fw-upgrade-status    OK fw-upgrade-error-msg N/A </pre>	
REST API	GET https://<ip>/nvue_v1/platform/transceiver/{transceiver-id}/firmware	
Related Commands	nv action install platform transceiver <transceiver-id> firmware files {file-name} nv action reset platform transceiver <transceiver-id>	
Notes		

### 5.7.4.5 nv show platform transceiver firmware files

	nv show platform transceiver <transceiver-id> firmware files Display available firmware files for transceiver module.	
Syntax Description	transceiver-id	The name of the transceiver
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform transceiver sw3 firmware files Available firmware files  File path ----- fw1.bin                   /host/fw-images/modules/fw1.bin fw2.bin                   /host/fw-images/modules/fw2.bin </pre>	
REST API	GET https://<ip>/nvue_v1/platform/transceiver/{transceiver-id}/firmware/files	
Related Commands	nv action install platform transceiver <transceiver-id> firmware files {file-name} nv action reset platform transceiver <transceiver-id> nv show platform transceiver <transceiver-id> firmware	
Notes		

### 5.7.4.6 nv show platform transceiver firmware files name

	nv show platform transceiver <transceiver-id> firmware files <file-name> Display specific firmware file for transceiver module.	
Syntax Description	transceiver-id	The name of the transceiver

	file-name	The name of the FW image file
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show platform transceiver sw3 firmware files fw1.bin Available firmware files  File path ----- fw1.bin                   /host/fw-images/modules/fw1.bin </pre>	
REST API	GET https://<ip>/nvue_v1/platform/transceiver/{transceiver-id}/firmware/files/{file-name}	
Related Commands	nv action install platform transceiver <transceiver-id> firmware files {file-name} nv action reset platform transceiver <transceiver-id> nv show platform transceiver <transceiver-id> firmware	
Notes		

---

## 6 Configuration Management

- [6.1 Managing Configurations as Revisions](#)
- [6.2 Restoring Factory Default Configuration](#)
- [6.3 Configuration Management Commands](#)

NVOS provides commands to efficiently manage and apply configurations on the system. These commands allow users to create, view, modify, compare, and save configurations to ensure consistent and accurate network setup. Whether user is making incremental updates, examining configuration differences, or saving changes for persistence across reboots, these tools offer flexibility and control over the system's operational state.

### 6.1 Managing Configurations as Revisions

NVOS utilizes a revision-based method to manage system configurations effectively. Each configuration change creates a unique revision, providing a clear and structured way to track, compare, and revert modifications.

- **Creating Revisions:**
  - A new revision is automatically generated whenever user set a configuration using `nv set *`.
  - All subsequent commands will be attached to last created revision.
  - Revision can be applied on system using `nv config apply`.
  - After a revision is applied, it becomes the "applied" revision and cannot be modified further.
  - Each revision includes metadata such as the revision ID, timestamp, the user who made the change, and the method of application (e.g., CLI or API).
- **Viewing Revisions:**
  - Use the `nv config revision` command to list all stored revisions. This provides a chronological history of all configuration changes made on the system.
- **Tracking Changes:**
  - The `nv config history` command provides detailed information about revisions, including who made the changes, when they were made, and how they were applied.
- **Comparing Revisions:**
  - Use `nv config diff <revision> <revision>` to identify differences between two revisions.
  - Compare the current pending configuration with the applied configuration to preview changes before applying them.
- **Reverting to a Previous Revision:**
  - If a configuration change causes issues, revert to a previous revision using `nv config apply <revision>`.
  - This allows to restore the system to a stable state without manually re-entering configurations.
- **Persistence:**
  - Changes made to configurations do not persist after a reboot unless explicitly saved using `nv config save`. Saving writes the current configuration to the startup file, making it the default state after a system restart.

## 6.2 Restoring Factory Default Configuration

Restore to factory default is used when the configuration is corrupted or when there is a need to start the system with default configuration (e.g., when the system is being introduced into a new network).

The "keep" parameter allows to specify different levels of a factory reset. Each option determines which parts of the system configuration, files, and logs are retained or erased. Below are the available options:

1. **Full Factory Reset (default):** If no specific option is selected, a full reset is performed. This option erases all configuration settings, system files, and log files.

```
admin@nvos:~$ nv action reset system factory-default
```

2. **keep all-config :** This option preserves all system configurations but removes system files and log files. Use this option when you need to retain the configuration but clear any logs or system files.

```
admin@nvos:~$ nv action reset system factory-default keep all-config
```

3. **keep basic :** This option keeps only the basic configuration necessary to maintain system connectivity while removing most of the configuration, system files, and log files. The following settings are retained:
  - a. Management interface (eth0, eth1)
  - b. Local AAA users and their roles
  - c. Password hardening rules
  - d. SSH server configuration
  - e. DNS server configuration

```
admin@nvos:~$ nv action reset system factory-default keep basic
```

4. **keep only-files :** This option removes all system configuration but retains system and log files. This can be useful for debugging purposes while clearing the configuration.

```
admin@nvos:~$ nv action reset system factory-default keep only-files
```

## 6.3 Configuration Management Commands

- [Configuration Management Commands](#)

## 6.4 Configuration Management Commands

- [6.4.1 Config](#)
  - [6.4.1.1 nv config apply](#)
  - [6.4.1.2 nv config detach](#)
  - [6.4.1.3 nv config diff](#)

- [6.4.1.4 nv config find](#)
- [6.4.1.5 nv config history](#)
- [6.4.1.6 nv config patch](#)
- [6.4.1.7 nv config replace](#)
- [6.4.1.8 nv config save](#)
- [6.4.1.9 nv config show](#)
- [6.4.1.10 nv show system config files](#)
- [6.4.1.11 nv show system config files](#)
- [6.4.1.12 nv show system config files brief](#)
- [6.4.1.13 nv action export system config](#)
- [6.4.1.14 nv action rename system config files](#)
- [6.4.1.15 nv action delete system config files](#)
- [6.4.1.16 nv action delete system config files](#)
- [6.4.1.17 nv action upload system config files](#)
- [6.4.1.18 nv action fetch system config files](#)
- [6.4.2 Factory](#)
  - [6.4.2.1 nv action reset system factory-default](#)

## 6.4.1 Config

### 6.4.1.1 nv config apply

	nv config apply [--assume-yes   --y   --assume-no   --confirm <time>   --confirm-status] Applies the pending configuration to become the applied configuration.	
Syntax Description	--assume-yes   --y   --assume-no	Automatically reply yes/no to all prompts.
	--confirm	Applies the configuration change but you must confirm the applied configuration. If you do not confirm within ten minutes, the configuration rolls back automatically. You can change the default time with the time argument
	--confirm-status	Shows the amount of time left before the automatic rollback.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv config apply Warning: management interface config is changed. If you are connected with it, your connection may be temporary interrupted.  Are you sure? [y/N] y applied</pre>	
REST API	PATCH https://<ip>/nvue_v1/revision/{revision-id} -H 'Content-Type: application/json' -d '{"state": "apply", "auto-prompt": {"ays": "ays_yes"}}'	
Related Commands		
Notes	NVOS applies but does not save the configuration; the configuration does not persist after a reboot.	

### 6.4.1.2 nv config detach

	nv config detach Detaches the configuration from the current pending configuration.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv config detach</pre>	
REST API	DELETE https://<ip>/nvue_v1/revision/{revision-id}	
Related Commands		
Notes	NVOS names the detached configuration pending and includes a timestamp with extra characters. For example: pending_20210128_212626_4WSY	

### 6.4.1.3 nv config diff

	nv config diff <revision> <revision>   [ -o commands ] Shows differences between configurations, such as the pending configuration and the applied configuration or the detached configuration and the pending configuration.	
Syntax Description	revision	pending configuration / applied configuration / detached configuration / startup configuration
	-o commands	Shows differences between two configuration revisions.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv config diff applied pending - set:   interface:     swpl:       description: Test</pre>	
REST API	GET https://<ip>/nvue_v1/<resource>?rev=<rev-A>&diff=<rev-B>	
Related Commands		
Notes		

### 6.4.1.4 nv config find

	nv config find <string>   [ -o commands ] Finds a portion of the applied configuration according to the provided string.	
Syntax Description	string	Search string
	-o commands	Shows the configurations as commands
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv config find swp1 - set:   interface:     swp1:       description: Test</pre>
REST API	GET https://<ip>/nvue_v1/?rev=applied&filled=False&diff=&search-string=<string>
Related Commands	
Notes	

### 6.4.1.5 nv config history

	nv config history <revision> Shows the apply history for the revision.	
Syntax Description	revision	pending configuration / applied configuration / detached configuration / startup configuration
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv config history pending - apply-id: n/1   apply-meta:     method: CLI     reason: Config update     rev_id: changeset/admin/2022-06-01_09.57.46_93BE     state_controls: {}     user: admin     date: '2022-06-01T12:25:38+00:00'     message: Config update by admin via CLI     ref: apply/2022-06-01_12.25.34_93BG/done</pre>	
REST API	GET https://<ip>/nvue_v1/revision/{revision-id} OR GET https://<ip>/nvue_v1/revision	
Related Commands		
Notes		

### 6.4.1.6 nv config patch

	nv config patch <yaml_file> Updates the pending configuration with the specified YAML configuration file.	
Syntax Description	yaml_file	configuration file
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv config patch /deps/nv-02/13/2021.yaml</pre>	
REST API	N/A	
Related Commands	nv config apply	
Notes		

### 6.4.1.7 nv config replace

	nv config replace <yaml_file> Replaces the pending configuration with the specified YAML configuration file.	
Syntax Description	yaml_file	configuration file
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv config replace /deps/nv-02/13/2021.yaml</pre>	
REST API	POST https://<ip>/nvue_v1/revision & DELETE https://<ip>/nvue_v1/?rev=<rev-id> & Update the configuration: PATCH https://<ip>/nvue_v1/?rev=<rev-id> Apply changes: PATCH https://<ip>/nvue_v1/revision/{revision-id} -H 'Content-Type: application/json' -d '{"state": "apply", "auto-prompt": {"ays": "ays_yes"}}'	
Related Commands	nv config apply	
Notes		

### 6.4.1.8 nv config save

	nv config save Overwrites the startup configuration with the applied configuration. The configuration persists after a reboot.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv config save</pre>	
REST API	PATCH https://<ip>/nvue_v1/revision/{revision-id} -H 'Content-Type: application/json' -d '{"state": "save", "auto-prompt": {"ays": "ays_yes"}}'	
Related Commands		
Notes		

### 6.4.1.9 nv config show

	nv config show [ -o commands ] Shows the currently applied configuration in <code>yaml</code> format.	
Syntax Description	-o commands	Shows the currently applied configuration commands.
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv config show -o commands nv set interface swpl link state down nv set interface swpl type ib</pre>
REST API	GET https://<ip>/nvue_v1/?rev=applied&filled=false
Related Commands	
Notes	

### 6.4.1.10 nv show system config files

	<p>nv show system config files Lists the configuration files</p>
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system config files Available config files  File Path ----- config-file.yaml      /host/config_files/config-file.yaml</pre>
REST API	GET https://<ip>/nvue_v1/system/config/files
Related Commands	
Notes	

### 6.4.1.11 nv show system config files

	<p>nv show system config files &lt;file-id&gt; Get a configuration file.</p>
Syntax Description	file-id      the file name to interact with
Default	N/A
History	25.02.1884
Example	<pre>- header:   model: VX   nvue-api-version: nvue_v1   rev-id: 1.0   version: Cumulus Linux 5.4.0 - set:   system:     message:       pre-login: Hello       /etc/nvue.d/config_files/config_file.yaml (END)</pre>
REST API	GET https://<ip>/nvue_v1/system/config/files/{file-name}
Related Commands	
Notes	The command will display the content of the configuration file in 'less' format

### 6.4.1.12 nv show system config files brief

	nv show system config files <file-id> brief Get a configuration file.	
Syntax Description	file-id	the file name to interact with
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system config files config-file.yaml brief Available config files  File path ----- config-file.yaml       /etc/nvue.d/config_files/config-file.yaml</pre>	
REST API	GET https://<ip>/nvue_v1/system/config/files/{file-name}	
Related Commands		
Notes	the command will display the path to the file	

### 6.4.1.13 nv action export system config

	nv action export system config <file-id> Export configuration file.	
Syntax Description	file-id	the file name to interact with
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action export system config new_confif_file.yml Exporting completed Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system/config/files/{file-name}	
Related Commands		
Notes	This command will export the applied configuration a new config file.	

### 6.4.1.14 nv action rename system config files

	nv action rename system config files <file-id> <new-name> Rename config file.	
Syntax Description	file-id	the file name to interact with
	new-name	the new name to rename to
Default		
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action rename system config new_config_file.yml new_name.yaml config file new_config_file.yaml renamed to new_name.yaml Action succeeded</pre>	

REST API	POST https://<ip>/nvue_v1/system/config/files/{file-name}
Related Commands	
Notes	

### 6.4.1.15 nv action delete system config files

	nv action delete system config files Delete all config files.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action rename system config new_config_file.yml new_name.yml config file new_config_file.yml renamed to new_name.yml Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system/config/files	
Related Commands		
Notes		

### 6.4.1.16 nv action delete system config files

	nv action delete system config files <file-id> Delete config file.	
Syntax Description	file-id	the file name to interact with
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action rename system config new_config_file.yml new_name.yml config file new_config_file.yml renamed to new_name.yml Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system/config/files/{file-name}	
Related Commands		
Notes		

### 6.4.1.17 nv action upload system config files

	nv action upload system config files <file-id> <remote-url> Upload config file.	
Syntax Description	file-id	the file name to interact with
	remote-url	remote url to upload to
Default	N/A	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action upload system config files new_config_file.yml scp:// username:password@server/tmp/ Uploading file: new_config_file.yml ... File upload successfully Action succeeded</pre>
REST API	POST https://<ip>/nvue_v1/system/config/files/{file-name}
Related Commands	
Notes	

### 6.4.1.18 nv action fetch system config files

	nv action fetch system config files <remote-url> Fetch configuration file.	
Syntax Description	remote-url	<ul style="list-style-type: none"> <li>• Remote url path to fetch file from.</li> <li>• Format: [protocol]://username[:password]@hostname/path/filename</li> <li>• Supported protocols: SCP, HTTPS, FILE, FTP, and SFTP.</li> <li>• The password must be encoded if it contains special characters and is provided as part of the command.</li> </ul>
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action fetch system config scp://username:password@server/tmp/ new_config_file.yml admin@nvos:~\$ nv action fetch system config file:///tmp/new_config_file.yml</pre>	
REST API	POST https://<ip>/nvue_v1/system/config/files/{file-name}	
Related Commands		
Notes	<p>Alternatively, you can upload a config file from the host machine to the switch.</p> <div style="border: 1px solid orange; padding: 5px; margin: 5px 0;">  <b>Note:</b> you must copy a config to the predefined directory: "/host/config_files/" </div> <pre>user@host:~\$ scp &lt;path-to-config-file&gt; &lt;switch-admin-username&gt;@&lt;switch-ip-address&gt;:/ host/config_files/&lt;desired-name&gt;.yaml</pre>	

## 6.4.2 Factory

### 6.4.2.1 nv action reset system factory-default

	nv action reset system factory-default [force] Clears the system and resets it entirely to its factory state.	
Syntax Description	force	Forces a reboot and configuration reset regardless of the prompt answer

Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv action reset system factory-<b>default</b> The operation will reset the system configuration and initiate a reboot. Type [y] to reset the system configuration and reboot. Type [N] to abort.  Do you want to <b>continue</b>? [Y/N] N Reset factory aborted by user </pre>
REST API	POST https://<ip>/nvue_v1/system/
Related Commands	
Notes	Command will reboot the system.

---

# 7 Telemetry Streaming

The following pages provide telemetry-related information.

- [gNMI Streaming](#)
- [SNMP](#)

## 7.1 gNMI Streaming

- [7.1.1 Configure the gNMI Agent using NVUE CLI Commands](#)
- [7.1.2 Supported Subscription Modes](#)
  - [7.1.2.1 Supported Models](#)
- [7.1.3 gNMI Client Requests](#)
- [7.1.4 Related Information](#)
- [7.1.5 gNMI Streaming Commands](#)

The [gRPC Network Management Interface](#) (gNMI) can collect and export system resources, interface, and counter information from NVOS to your gNMI client.

### 7.1.1 Configure the gNMI Agent using NVUE CLI Commands

The gNMI server feature state can be set over NVOS using simple NVUE CLI commands:

Show command:

```
nvos@switch:~$ nv show system gnmi-server
operational    applied
-----
state          enabled       enabled
certificate    self-signed   self-signed
is-running     yes
version        4.13.0-3000-2
```

Set command:

```
nvos@switch:~$ nv set system gnmi-server state <enabled | disabled>
```

Unset command:

```
nvos@switch:~$ nv unset system gnmi-server state
```

The state is enabled by default and the unset command will restore the state to enabled, if it is not already.

### 7.1.2 Supported Subscription Modes

NVOS supports the following gNMI [subscription modes](#):

- **STREAM Mode:** In this mode, the client subscribes to receive updates whenever there is a change in the telemetry data. This mode is suitable for scenarios where you need real-time notifications of data changes.

- **ONCE Mode:** This mode retrieves the data once and then terminates the subscription. It's ideal for scenarios where a single snapshot of the data is needed without ongoing updates.
- **POLL Mode:** In this mode, the client periodically requests data from the server. This mode allows clients to fetch data at defined intervals, providing a balance between real-time and scheduled updates.

Supported stream modes:

- **ON\_CHANGE** - when a subscription is defined to be "on change", data updates are only sent when the value of the data item changes.
- **SAMPLE** - This mode allows clients to receive periodic samples of telemetry data at specified intervals. This mode is beneficial for scenarios where continuous streaming of data is not necessary, but periodic updates are required for monitoring and analytics.

Key Parameters for STREAM SAMPLE Mode:

- **sample\_interval** (mandatory): Defines the interval at which samples are sent to the client. This parameter controls the frequency of data transmission.
- **suppress\_redundant** (optional, default false): Determines whether redundant data updates, which have not changed since the last sample, should be suppressed. This helps in reducing unnecessary data transmission and optimizing network usage.
- **heartbeat\_interval** (optional, default disabled): Specifies the interval for sending heartbeat messages to indicate that the connection is still active. Heartbeats help in monitoring the health of the connection and detecting failures.

### 7.1.2.1 Supported Models

NVOS supports the following OpenConfig models ([v4.3.0](#))

Model	Supported Data
<a href="#">openconfig-interfaces</a>	Name, Operstatus, AdminStatus, IfIndex, Description, Enabled, Counters (InPkts, OutPkts, InOctets, InUnicastPkts, InDiscards, InMulticastPkts, InErrors, OutOctets, OutUnicastPkts, OutMulticastPkts, OutDiscards, OutErrors)
<a href="#">openconfig-platform-cpu</a>	OperStatus, Utilization
<a href="#">openconfig-platform-fan</a>	OperStatus, Speed
<a href="#">openconfig-transceiver</a>	OperStatus, InputPower, OutputPower, LaserBiasCurrent, LaserTemperature, Present, SupplyVoltage, FormFactor, VendorPart, SerialNo, Sevirty, Thresholds/LaserTemperatureLower, Thresholds/LaserTemperatureUpper
<a href="#">openconfig-system</a>	Hostname, MacAddress, BootTime

NVOS supports the following NVIDIA models:

Model
nvidia-interfaces-infiniband
nvidia-interfaces-infiniband-errors-ext
nvidia-platform-general-ext

Model
nvidia-platform-asic
nvidia-system-events
nvidia-if-phy-diag
nvidia-platform-transceiver-diag
nvidia-platform-general-ext-versions
nvidia-platform-types

The YANG models above can be found in the attached YANG zip file.

### 7.1.3 gNMI Client Requests

gNMI client on a host can request capabilities and data from the switch. The examples below use the [gNMIC client](#).

The following example shows a gNMIC `STREAM SAMPLE` mode request for specific Interface data, with a sample interval of 30 seconds, suppress redundant flag enabled, and heartbeat interval of 120 seconds:

```
gnmic -a "IP" --port 9339 --skip-verify subscribe --prefix "interfaces" --path "/interface[name=sw1p1]" --target nvos -u admin -p ***** --mode stream --stream-mode sample --sample-interval 30s --suppress-redundant --heartbeat-interval 120s
```

The following example shows a gNMIC `STREAM ON-CHANGE` mode request for system events, with an updates-only flag enabled:

```
gnmic -a "IP" --port 9339 --skip-verify subscribe --prefix "/system-events" --path "" --target nvos -u admin -p ***** --mode stream --stream-mode on-change --updates-only
```

The following example shows a gNMIC `ONCE` mode request and server response for interface MTU (-d for debug mode):

```
gnmic -a "IP" --port 9339 --skip-verify subscribe --prefix "interfaces" --path "/interface[name=sw1p1]/infiniband/state/mtu" -d --target nvos -u admin -p ***** --mode once
{
  "source": "IP",
  "subscription-name": "default-1709707931",
  "timestamp": 1709707925858795109,
  "time": "2024-03-06T08:52:05.858795109+02:00",
  "prefix": "interfaces/interface[name=sw1p1]",
  "target": "nvos",
  "updates": [
    {
      "Path": "infiniband/state/mtu",
      "values": {
        "infiniband/state/mtu": 256
      }
    }
  ]
}
```

The following example shows a gNMIC `ONCE` request for all supported paths:

```
gnmic -a "IP" --port 9339 --skip-verify subscribe --prefix "/" --path "" --target nvos -u admin -p ***** --mode once
```

The following example shows a gNMIC `POLL` mode request and server response for FAN1/1 speed:

```
gnmic -a "IP" --port 9339 --skip-verify subscribe --prefix "components" --path "component[name=FAN1/1]/fan/state/speed" --target nvos -u admin -p ***** --format flat --mode poll
components/component[name=FAN1/1]/fan/state/speed: 33
```

The following example shows a gNMIc **STREAM** mode request for specific system-event "text" leaf with **PROTO** encoding:

```
gnmic -a "IP" --port 9339 --skip-verify subscribe --prefix "system-events" --path "system-event[event-id=38]/state/text" --target nvos -u admin -p ***** --encoding proto --format prototext --mode stream

sync_response: true

update: {
  timestamp: 1719295967820127958
  prefix: {
    elem: {
      name: "system-events"
    }
    elem: {
      name: "system-event"
      key: {
        key: "event-id"
        value: "38"
      }
    }
  }
  target: "nvos"
}
update: {
  path: {
    elem: {
      name: "state"
    }
    elem: {
      name: "text"
    }
  }
  val: {
    string_val: "Interface admin state is up"
  }
}
```

A list of supported events can be found in the [Event Management](#) page.

The following example shows a gRPC curl command to describe the server using gRPC reflection service:

```
docker run fullstorydev/grpcurl -H username:admin -H password:***** -insecure "IP":9339 describe

gnmi.gNMI is a service:
service gNMI {
  rpc Capabilities ( .gnmi.CapabilityRequest ) returns ( .gnmi.CapabilityResponse );
  rpc Get ( .gnmi.GetRequest ) returns ( .gnmi.GetResponse );
  rpc Set ( .gnmi.SetRequest ) returns ( .gnmi.SetResponse );
  rpc Subscribe ( stream .gnmi.SubscribeRequest ) returns ( stream .gnmi.SubscribeResponse );
}
grpc.reflection.v1.ServerReflection is a service:
service ServerReflection {
  rpc ServerReflectionInfo ( stream .grpc.reflection.v1.ServerReflectionRequest ) returns
( stream .grpc.reflection.v1.ServerReflectionResponse );
}
grpc.reflection.v1alpha.ServerReflection is a service:
service ServerReflection {
  rpc ServerReflectionInfo ( stream .grpc.reflection.v1alpha.ServerReflectionRequest ) returns
( stream .grpc.reflection.v1alpha.ServerReflectionResponse );
}
```

The following example shows a gNMIc **ONCE** mode request for all the supported paths:

```
gnmic -a "IP" --port 9339 --skip-verify subscribe --prefix "/" --path "" --target nvos -u admin -p ***** --mode once --format flat
```

The following example shows a gNMIc **Capabilities** request to retrieve the set of capabilities that is supported by the server:

```
gnmic -a "IP" --port 9339 --skip-verify capabilities -u admin -p *****
```

## 7.1.4 Related Information

- [gNMI presentation to IETF](#)
- [gNMIc client](#)
- [gNMI subscription mode documentation](#)

## 7.1.5 gNMI Streaming Commands

- [gNMI Streaming Commands](#)

## 7.1.6 gNMI Streaming Commands

- [7.1.6.1 nv show system gnmi-server](#)
- [7.1.6.2 nv set system gnmi-server state](#)
- [7.1.6.3 nv unset system gnmi-server state](#)
- [7.1.6.4 nv set sys gnmi-server certificate](#)

### 7.1.6.1 nv show system gnmi-server

	nv show system gnmi-server Displays the gNMI server configured state, actual state, and version.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system gnmi-server operational  applied ----- state        enabled     enabled certificate  self-signed self-signed is-running   yes version      4.12.0-PS</pre>	
REST API	GET https://<ip>/nvue_v1/system/gnmi-server	
Related Commands	nv set system gnmi-server state disabled	

### 7.1.6.2 nv set system gnmi-server state

	nv set system gnmi-server state <enabled   disabled> Sets gNMI server state.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system gnmi-server state disabled</pre>	

REST API	PATCH https://<ip>/nvue_v1/system/gnmi-server
Related Commands	nv unset system gnmi-server state

### 7.1.6.3 nv unset system gnmi-server state

	nv unset system gnmi-server state Unsets gNMI server state back to default.	
Syntax Description	N/A	
Default	Enabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset system gnmi-server state</pre>	
REST API	DELETE https://<ip>/nvue_v1/system/gnmi-server	
Related Commands	nv set system gnmi-server state	

### 7.1.6.4 nv set sys gnmi-server certificate

	nv set sys gnmi-server certificate {cert id} Set CA certificate for API mTLS connection.	
Syntax Description	certificate	Certificate ID string
Default	self-signed	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set sys gnmi-server certificate cert_id</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/gnmi-server/certificate?rev= Content-Type: application/json {"certificate": "cert_id"}	
Related Commands		
Notes		

## 7.2 SNMP

Simple Network Management Protocol (SNMP) is a network protocol for management and monitoring of network devices.

NVOS supports:

- SNMP versions v1 and v2c
- Standard MIBs
- Query mode only

## 7.2.1 Standard MIBs

MIB	Standard
RFC1213-MIB	RFC 1213
IF-MIB	RFC 2863
ENTITY-MIB	RFC 2737
ENTITY-SENSOR-MIB	RFC 3433

## 7.2.2 Configuring SNMP

Activate the SNMP server on the switch by running:

```
admin@nvos:~$ nv set system snmp-server state enabled
admin@nvos:~$ nv set system snmp-server listening-address <IPv4|IPv6|all|all-v6|localhost|localhost-v6>
admin@nvos:~$ nv set system snmp-server readonly-community <community-string>
admin@nvos:~$ nv set system snmp-server system-contact <contact>
admin@nvos:~$ nv set system snmp-server system-location <location>
```



Community strings are case sensitive.



Multiple distinct IPv4/IPv6 addresses may be configured as listening addresses, as long as they are present on the system management interface.

Use all/all-v6 to listen on all available management addresses.

## 7.2.3 SNMP Commands

- [SNMP Commands](#)

## 7.2.4 SNMP Commands

- [7.2.4.1 nv show system snmp-server](#)
- [7.2.4.2 nv show system snmp-server listening-address](#)
- [7.2.4.3 nv show system snmp-server readonly-community](#)
- [7.2.4.4 nv set system snmp-server state](#)
- [7.2.4.5 nv set system snmp-server listening-address](#)
- [7.2.4.6 nv set system snmp-server readonly-community](#)
- [7.2.4.7 nv set system snmp-server auto-refresh-interval](#)
- [7.2.4.8 nv set system snmp-server system-contact](#)
- [7.2.4.9 nv set system snmp-server system-location](#)

### 7.2.4.1 nv show system snmp-server

	nv show system snmp-server Show SNMP server configuration.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system snmp-server  ----- operational  applied ----- [listening-address]    all             all [readonly-community]  public         public auto-refresh-interval 60             60 state                 enabled        enabled </pre>	
REST API	GET https://<ip>/nvue_v1/system/snmp-server	
Related Commands		
Notes		

### 7.2.4.2 nv show system snmp-server listening-address

	nv show system snmp-server listening-address [addr] Show SNMP server listening addresses configuration.	
Syntax Description	addr	Optional, show info of specific address. If omitted, will show a summary of all listening addresses
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system snmp-server listening-address        port ----- 1.1.1.1 161  switch (config) # nv show system snmp-server listening-address 1.1.1.1  ----- operational  applied ----- port 161             161 </pre>	
REST API	GET https://<ip>/nvue_v1/system/snmp-server/listening-address GET https://<ip>/nvue_v1/system/snmp-server/listening-address/{addr}	
Related Commands		
Notes		

### 7.2.4.3 nv show system snmp-server readonly-community

	nv show system snmp-server readonly-community Show SNMP server readonly communities.	
--	---	--

Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system snmp-server readonly-community  ----- community1 community2</pre>
REST API	GET https://<ip>/nvue_v1/system/snmp-server/readonly-community
Related Commands	
Notes	

#### 7.2.4.4 nv set system snmp-server state

	nv set system snmp-server state <enabled   disabled> Enable or disable the SNMP server.	
Syntax Description	N/A	
Default	Disabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system snmp-server state enabled</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/snmp-server	
Related Commands		
Notes	At least 1 listening-address and 1 readonly-community must be configured in order to enabled SNMP server.	

#### 7.2.4.5 nv set system snmp-server listening-address

	nv set system snmp-server listening-address <addr> [port <port>] Configure a listening-address for the SNMP server.	
Syntax Description	addr	The address to set. Both IPv4 and IPv6 are supported. Special addresses may be specified by designated keywords: all (0.0.0.0), localhost (127.0.0.1), all-v6 (:::), localhost-v6 (:::1)
	port	Optional, set the port for the listening-address. If omitted, will use the default SNMP port 161.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system snmp-server listening-address 1.1.1.1 port 44444</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/snmp-server/listening-address/{addr}	
Related Commands		

Notes	Multiple listening-addresses may be configured.
-------	---

### 7.2.4.6 nv set system snmp-server readonly-community

	nv set system snmp-server readonly-community <community> Configure a readonly community for the SNMP server	
Syntax Description	community	The community to set
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system snmp-server readonly-community community1</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/snmp-server/readonly-community/{community}	
Related Commands		
Notes	Up to 4 readonly communities may be configured. The same communities are used for both IPv4 and IPv6.	

### 7.2.4.7 nv set system snmp-server auto-refresh-interval

	nv set system snmp-server auto-refresh-interval {interval} Set the interval in seconds between SNMP data refreshes	
Syntax Description	interval	The interval to set, in seconds. The allowed range is [5-300]
Default	60	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system snmp-server auto-refresh-interval 5</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/snmp-server	
Related Commands		
Notes	In general, SNMP server refreshes its internal data every time period as set by the auto-refresh-interval configuration. However, sysName is an exception. It reflects the system hostname, and is being refreshed on a less frequent basis, every 12 time periods, as the system hostname is not expected to change very often.	

### 7.2.4.8 nv set system snmp-server system-contact

	nv set system snmp-server system-contact <contact> Set the value for sysContact in MIB-II.	
Syntax Description	contact	The contact to set, up to 255 characters
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system snmp-server system-contact "John Doe"</pre>
REST API	PATCH https://<ip>/nvue_v1/system/snmp-server
Related Commands	
Notes	

### 7.2.4.9 nv set system snmp-server system-location

	nv set system snmp-server system-location <location> Set the value for sysLocation in MIB-II.	
Syntax Description	location	The location to set, up to 255 characters
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system snmp-server system-location "2nd floor cabinet"</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/snmp-server	
Related Commands		
Notes		

---

## 8 Monitoring and Diagnostics

- [Health Monitoring](#)
- [Logging](#)
- [Remote Logging](#)
- [Link Diagnostic Per Port](#)
- [Event Management](#)
- [Statistics](#)
- [Technical Support](#)
- [Troubleshooting](#)

### 8.1 Health Monitoring

NVOS includes a health daemon that is responsible for collecting health events in the system and monitoring various components, including hardware components such as fans, power supply units, leakage sensors, and failing docker containers.

This health daemon runs every 3 seconds, analyzing components. If it detects an issue, it appears in [nv show system health](#) and publishes a health event in gNMI.

Health daemon monitors two main items: Service and Hardware

#### 8.1.1 Service Monitoring

The health daemon ensures the continuous operation of essential system services. It monitors:

- Critical Dockers and Services: Verifies that all vital dockers and services are running

#### 8.1.2 Hardware Monitoring

- Leakage Sensors: Detects any potential fluid leaks
- Temperature Sensors: Monitors the temperature of all hardware components to prevent overheating
- Voltage Sensors: Tracks voltage levels across hardware components to ensure proper functionality
- Fan Speeds: Checks the speed of system fans to maintain optimal airflow and cooling
- Power Supply Units (PSUs): Monitors the status of power supply units for stability
- ASIC Health Status: Verifies the health of ASIC components to prevent processing issues
- Disk: Checks the health and free space of the Disk
- CPU: Monitors the CPU utilization and temperature
- Transievers: Monitors the status of the transievers connected to the system

#### 8.1.3 Output Examples

Example output for a healthy system:

## healthy system

```
admin@nvos:~$ nv show system health
----- operational applied
status      OK
status-led  green
```

```
Health issues
=====
No Data
```

Example output for a faulty system:

## bad system

```
admin@nvos:~$ nv show system health
----- operational applied
status      Not OK
status-led  amber
```

```
Health issues
=====
Component          Status information
-----
LEAKAGE-2          detected leakage
PMIC 1 Temp        temperature is too hot, temperature=2008.0, threshold=125.0
```

## 8.1.4 Health Monitoring Commands

- [Health Monitoring Commands](#)

## 8.1.5 Health Monitoring Commands

- [8.1.5.1 nv show system health](#)
- [8.1.5.2 nv show system health history](#)

### 8.1.5.1 nv show system health

	nv show system health Show system health status.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system health ----- operational applied status      Not OK status-led  off  Health issues ===== Component          Status information ----- LEAKAGE-1          detected leakage</pre>	

Related Commands	nv show system health history
Notes	

## 8.1.5.2 nv show system health history

	nv show system health history [file-name] Show system health history file.									
Syntax Description	file-name	Show health history files in the system								
History	25.02.1884									
Example	<pre>admin@nvos:~\$ nv show system health history</pre> <pre>admin@nvos:~\$ nv show system health history files</pre> <table border="0"> <thead> <tr> <th>health history reports</th> <th>File path</th> </tr> <tr> <th>-----</th> <th>-----</th> </tr> </thead> <tbody> <tr> <td>health_history</td> <td>/var/log/health_history</td> </tr> <tr> <td>health_history.1</td> <td>/var/log/health_history.1</td> </tr> </tbody> </table> <pre>admin@nvos:~\$ nv show system health history files health_history</pre>		health history reports	File path	-----	-----	health_history	/var/log/health_history	health_history.1	/var/log/health_history.1
health history reports	File path									
-----	-----									
health_history	/var/log/health_history									
health_history.1	/var/log/health_history.1									
Related Commands	nv show system health									
Notes	<ul style="list-style-type: none"> <li>• When running the command via the CLI, the file open in “Less”</li> <li>• When no file is selected, the default file name that opens is “health_history”</li> </ul>									

## 8.2 Logging

### 8.2.1 Logging

To display syslog, run the following command:

```
admin@nvos:~$ nv show system log
```

To display specific syslog file, run the following command:

```
admin@nvos:~$ nv show system log files syslog.1
```

### 8.2.2 Logging Commands

- [Logging Commands](#)

### 8.2.3 Logging Commands

- [8.2.3.1 nv show system log](#)
- [8.2.3.2 nv show system rotation](#)

- [8.2.3.3 nv set/unset system log component](#)
- [8.2.3.4 nv set/unset system rotation disk-percentage](#)
- [8.2.3.5 nv set/unset system rotation frequency](#)
- [8.2.3.6 nv set/unset system rotation max-number](#)
- [8.2.3.7 nv set/unset system rotation size](#)
- [8.2.3.8 nv set/unset system syslog trap](#)
- [8.2.3.9 nv action rotate system](#)

### 8.2.3.1 nv show system log

	nv show system {log   debug-log} {files {file-name}   component {component-name}} {--view = follow} Displays the log file	
Syntax Description	log   debug-log	Displays the log file in interactive mode , similar to LINUX “less” utility
	files	Displays the list of log files
	file-name	Displays an archived log file
	component	Displays the log configuration of all the system components
	component-name	Displays the log configuration of specific system component
	--view follow	Displays the last few lines of the current log file and then continues to display new lines as they come in until the user hits Ctrl+C, similar to LINUX “tail” utility.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system log --view=follow May 31 15:59:15.041937 jaguar-70 INFO pmon#sensord: PSU-1(L) Temp 3: 36.2 C (min = -0.5 C, max = 60.0 C) May 31 16:00:01.312936 nvos INFO core_cleanup.py: Cleaning up core files May 31 16:00:01.313062 nvos INFO core_cleanup.py: Finished cleaning up core files  admin@nvos:~# nv show system log files Logs files names  Logs files path ----- syslog           /var/log/syslog  admin@nvos:~# nv show system log component Component        Level ----- nvued            debug orchagent        notice portsyncd        notice sai_api_port     notice sai_api_switch   notice syncd            notice  admin@nvos:~# nv show system log component syncd ----- operational applied pending description ----- level notice                               The component log level </pre>	
REST API	GET https://<ip>/nvue_v1/system/log GET https://<ip>/nvue_v1/system/log/component GET https://<ip>/nvue_v1/system/log/comonenpt/{comonenpt-name} GET https://<ip>/nvue_v1/system/log/files GET https://<ip>/nvue_v1/system/log/files/{file-name}	
Related Commands	nv show system debug-log nv set system log component nvued level <level>	

Notes	
-------	--

### 8.2.3.2 nv show system rotation

	nv show system {log-kind} rotation Show log rotation criteria configuration.	
Syntax Description	log-kind	Log kind: log or debug-log
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system log rotation ----- operational  applied ----- frequency    daily        daily max-number   20           20 size         10.0         10.0  admin@nvos:~\$ nv show system debug-log rotation ----- operational  applied ----- frequency    daily        daily max-number   10           10 size         20.0         20.0 </pre>	
REST API	GET https://<ip>/nvue_v1/system/log/rotation	
Related Commands	nv set system log rotation nv set system debug-log rotation	
Notes		

### 8.2.3.3 nv set/unset system log component

	nv set system log component <component-name> level <level> nv unset system log component <component-name> level Set/unset the system component minimum priority level of messages to log	
Syntax Description	component-name	The system component name: nvued, orchagent, portsyncd, sai_api_port, sai_api_switch, syncd
	level	The minimum priority level of messages to log: critical, debug, error, info, notice, warn
Default	notice	
History	25.02.1884	
Example	<pre> admin@nvos:~# nv set system log component nvued level debug </pre>	
REST API	PATCH https://<ip>/nvue_v1/system/log/component	
Related Commands	nv show system log component nv unset system log component level	
Notes	To get component-name run "nv show system log component"	

### 8.2.3.4 nv set/unset system rotation disk-percentage

	nv set system {log-kind} rotation disk-percentage <percentage> nv unset system {log-kind} rotation disk-percentage Set/unset the size of the log file to rotate based on disk size percentage.	
Syntax Description	log-kind	Log kind: log or debug-log
	percentage	Percentage value: 0.001 - 100
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system log rotation disk-percentage 10 admin@nvos:~\$ nv unset system log rotation disk-percentage</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/rotation	
Related Commands	nv show system log rotation nv show system debug-log rotation	
Notes		

### 8.2.3.5 nv set/unset system rotation frequency

	nv set system {log-kind} rotation frequency <frequency> nv unset system {log-kind} rotation frequency Set/unset the frequency of the file rotation.	
Syntax Description	log-kind	Log kind: log or debug-log
	frequency	Rotation frequency: daily, weekly, monthly, yearly
Default	daily	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system log rotation frequency weekly admin@nvos:~\$ nv unset system log rotation frequency</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/rotation	
Related Commands	nv show system log rotation nv show system debug-log rotation	
Notes		

### 8.2.3.6 nv set/unset system rotation max-number

	nv set system {log-kind} rotation max-number <count> nv unset system {log-kind} rotation max-number Set/unset the max number of file rotations.	
Syntax Description	log-kind	Log kind: log or debug-log
	count	Number of file rotations before being removed: 0-999999
Default	20	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set system log rotation max-number 10 admin@nvos:~\$ nv unset system log rotation max-number</pre>
REST API	PATCH https://<ip>/nvue_v1/system/rotation
Related Commands	nv show system log rotation nv show system debug-log rotation
Notes	

### 8.2.3.7 nv set/unset system rotation size

	nv set system {log-kind} rotation size <mebibytes> nv unset system {log-kind} rotation size Set the size of the file to be rotated.	
Syntax Description	log-kind	Log kind: log or debug-log
	mebibytes	The size threshold for a file to be rotated: 0.001-3500 MiB
Default	10	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system log rotation size 100.123 admin@nvos:~\$ nv unset system log rotation size</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/rotation	
Related Commands	nv show system log rotation nv show system debug-log rotation	
Notes		

### 8.2.3.8 nv set/unset system syslog trap

	nv set system syslog trap <severity> nv unset system syslog trap Set the minimum log level to send the logs to remote servers. The unset form of that command resets trap to default.	
Syntax Description	severity	debug   info   notice   warn   error   critical   none
Default	Notice	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog trap info admin@nvos:~\$ nv unset system syslog trap</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/trap	
Related Commands	nv show system log rotation	
Notes		

### 8.2.3.9 nv action rotate system

	nv action rotate system {log-kind} Force log file rotation. Action to trigger file rotation manually.	
Syntax Description	log-kind	Log kind: log or debug-log
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action rotate system log Performing syslog log file rotation... Log rotated successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system	
Related Commands	nv show system log rotation	
Notes		

## 8.3 Remote Logging

To configure remote syslog to send syslog messages to a remote syslog server, follow the steps below.

1. Set remote syslog server.

```
admin@nvos:~$ nv set system syslog server <IP address/hostname>
```

2. (Optional) Set the destination port of the remote host.

```
admin@nvos:~$ nv set system syslog server 10.20.30.40 port 1234
```

3. (Optional) Filter log messages according to an input regex.

```
admin@nvos:~$ nv set system syslog server 10.20.30.40 filter include ERROR
```

4. Set the minimum severity of the log level to info (it has no impact on local log level).

```
admin@nvos:~$ nv set system syslog server 10.20.30.40 trap info
```

5. Set the protocol over which to communicate with remote syslog server

```
admin@nvos:~$ nv set system syslog server 10.20.30.40 protocol tcp
```

### 8.3.1 Remote Logging Commands

- [Remote Logging Commands](#)

## 8.3.2 Remote Logging Commands

- [8.3.2.1 nv show system syslog](#)
- [8.3.2.2 nv show system syslog format](#)
- [8.3.2.3 nv show system syslog server](#)
- [8.3.2.4 nv show system syslog server id](#)
- [8.3.2.5 nv unset system syslog](#)
- [8.3.2.6 nv set/unset system syslog format](#)
- [8.3.2.7 nv set/unset system syslog server filter exclude](#)
- [8.3.2.8 nv set/unset system syslog server filter include](#)
- [8.3.2.9 nv set/unset system syslog format welf firewall-name](#)
- [8.3.2.10 nv unset system syslog server](#)
- [8.3.2.11 nv set/unset system syslog server](#)
- [8.3.2.12 nv set/unset system syslog server trap](#)
- [8.3.2.13 nv set/unset system syslog server port](#)
- [8.3.2.14 nv set/unset system syslog server protocol](#)
- [8.3.2.15 nv set/unset system syslog server vrf](#)

### 8.3.2.1 nv show system syslog

	nv show system syslog Show syslog configuration. It shows the next information: log format, servers list, trap level.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system syslog ----- trap          notice          notice format        standard        standard [server]      10.20.30.40    10.20.30.40 [server]      mysyslogsrv    mysyslogsrv           </pre>	
REST API	GET https://<ip>/nvue_v1/system/syslog	
Related Commands	nv show system log rotation	
Notes		

### 8.3.2.2 nv show system syslog format

	nv show system syslog format Show syslog format.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show system syslog format               operational  applied ----- format        welf          welf welf          firewall-name  NVOS switch  NVOS switch</pre>
REST API	GET https://<ip>/nvue_v1/system/syslog/format
Related Commands	nv show system log rotation
Notes	

### 8.3.2.3 nv show system syslog server

	nv show system syslog server Show remote syslog servers.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system syslog server Remote syslog servers ----- 10.20.30.40 20.30.40.50 mysyslogsrv</pre>
REST API	GET https://<ip>/nvue_v1/system/syslog/server
Related Commands	
Notes	

### 8.3.2.4 nv show system syslog server id

	nv show system syslog server <server-id> Show remote syslog server configuration.
Syntax Description	server-id Remote syslog server ipv4, or ipv6, or hostname
Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show system syslog server 10.20.30.40               operational  applied  description ----- filter exclude    ^bla      ^bla      Filter log messages sent to a remote server port       514      514      Set remote syslog server destination port protocol    udp       udp       Networking protocol to send the messages trap       notice   notice    Minimum severity of log messages to send vrf        mgmt     mgmt     VRF over which to send logs to remote server</pre>
REST API	GET https://<ip>/nvue_v1/system/syslog/server/{server-id}
Related Commands	
Notes	

### 8.3.2.5 nv unset system syslog

	nv unset system syslog Clear syslog global parameters. It completely removes syslog configuration.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset system syslog</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog	
Related Commands	nv show system log rotation	
Notes		

### 8.3.2.6 nv set/unset system syslog format

	nv set system syslog format [standard   welf] nv unset system syslog format Set the log format. The unset form of the command resets format to default value.	
Syntax Description	N/A	
Default	Standard	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog format welf admin@nvos:~\$ nv unset system syslog format</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/format	
Related Commands	nv set system syslog format welf firewall-name	
Notes		

### 8.3.2.7 nv set/unset system syslog server filter exclude

	nv set system server <server-id> filter exclude <regex> nv unset system server <server-id> filter exclude Define the filter to exclude logs from sending to remote server. The unset form of the command clears the filter.	
Syntax Description	server-id	Remote syslog server ipv4, or ipv6, or hostname
	regex	Pattern to exclude logs from sending
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system syslog server 10.20.30.40 filter exclude "New password" admin@nvos:~\$ nv unset system syslog server 10.20.30.40 filter exclude</pre>
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server/{server-id}/filter
Related Commands	nv set system syslog server filter include
Notes	Setting exclude filter clears include filter.

### 8.3.2.8 nv set/unset system syslog server filter include

	nv set system server <server-id> filter include <regex> nv unset system server <server-id> filter include Define the filter to include logs to send to remote server. The unset form of the command clears the filter.	
Syntax Description	server-id	Remote syslog server ipv4, or ipv6, or hostname
	regex	Pattern to include logs to send
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog server 10.20.30.40 filter include ERROR admin@nvos:~\$ nv unset system syslog server 10.20.30.40 filter include</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server/{server-id}/filter/include	
Related Commands	nv set system syslog server filter exclude	
Notes	Setting include filter clears exclude filter.	

### 8.3.2.9 nv set/unset system syslog format welf firewall-name

	nv set system format welf firewall-name <name> nv unset system format welf firewall-name Set WELF format firewall name. The unset form of that command clears the firewall name.	
Syntax Description	Name	Log firewall name to include into WELF log format.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog format welf firewall-name "NVOS switch 2" admin@nvos:~\$ nv unset system syslog format welf firewall-name</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/format/welf	
Related Commands		
Notes		

### 8.3.2.10 nv unset system syslog server

	nv unset system syslog server Clear remote syslog servers.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv unset system syslog server</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server	
Related Commands		
Notes		

### 8.3.2.11 nv set/unset system syslog server

	nv set system syslog server {server-id} nv unset system syslog server {server-id} Set new remote syslog server. The unset form of the command clears a specific server.	
Syntax Description	server-id	Remote syslog server ipv4, or ipv6, or hostname
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog server 10.20.30.40 admin@nvos:~\$ nv unset system syslog server 10.20.30.40</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server/{server-id}	
Related Commands		
Notes		

### 8.3.2.12 nv set/unset system syslog server trap

	nv set system server <server-id> trap <severity> nv unset system server <server-id> trap Set the minimum log level to send the NVOS daemon logs to remote server. The unset form of the command clears trap. Setting and unsetting has no impact on local logs level.	
Syntax Description	server-id	Remote syslog server ipv4, or ipv6, or hostname
	severity	Options: debug   info   notice   warn   error   critical   none
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system syslog server 10.20.30.40 trap warn admin@nvos:~\$ nv unset system syslog server 10.20.30.40 trap</pre>
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server/{server-id}
Related Commands	
Notes	Overrides global syslog trap.

### 8.3.2.13 nv set/unset system syslog server port

	<pre>nv set system server &lt;server-id&gt; port &lt;port-number&gt; nv unset system server &lt;server-id&gt; port</pre> <p>Set the port over which to communicate with remote syslog server. The unset form of the command returns port to default.</p>	
Syntax Description	server-id	Remote syslog server ipv4, or ipv6, or hostname
	port-number	Port number of the remote syslog server: 1-65535
Default	512	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog server 10.20.30.40 port 1234 admin@nvos:~\$ nv unset system syslog server 10.20.30.40 port</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server/{server-id}	
Related Commands		
Notes		

### 8.3.2.14 nv set/unset system syslog server protocol

	<pre>nv set system server &lt;server-id&gt; protocol [tcp   udp] nv unset system server &lt;server-id&gt; protocol</pre> <p>Set the protocol over which to communicate with remote syslog server. The unset form of the command returns the protocol to default.</p>	
Syntax Description	server-id	Remote syslog server ipv4, or ipv6, or hostname
Default	udp	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog server 10.20.30.40 protocol tcp admin@nvos:~\$ nv unset system syslog server 10.20.30.40 protocol</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server/{server-id}	
Related Commands		
Notes		

### 8.3.2.15 nv set/unset system syslog server vrf

	nv set system server <server-id> vrf [default   mgmt] nv unset system server <server-id> vrf [default   mgmt] Specify the VRF over which to communicate with the remote server. The unset form of the command returns VRF to default.	
Syntax Description	server-id	Remote syslog server ipv4, or ipv6, or hostname
Default	default	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system syslog server 10.20.30.40 vrf mgmt admin@nvos:~\$ nv unset system syslog server 10.20.30.40 vrf</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/syslog/server/{server-id}	
Related Commands		
Notes		

## 8.4 Link Diagnostic Per Port

When debugging a system, it is important to be able to quickly identify the root of a problem. The Diagnostic commands enables an insight into the physical layer components where the user is able to see information such as a cable status (plugged/unplugged) or if Auto-Negotiation has failed.

List of possible output messages:

Code	Firmware PHY Indication (0-1023)
0	No issue observed
1	Port is close by command
2-4	Auto Negotiation failure
5-8	Link training failure
9-13	Logical mismatch between link partners
14	Remote fault received
15	Bad Signal integrity
16	Compliance code mismatch (protocol mismatch between cable and port)
17	Bad signal integrity
18	Internal error
19	Internal error
22	Internal error
23	Internal error
24-32	Cable compliance code mismatch (protocol mismatch between cable and port)

34	Speed degradation
35	Speed degradation
38	Auto Negotiation failure
39	Auto Negotiation failure
40	VPI protocol do not match
41	Port is closed, module cannot be set to the enabled rate
42	Bad signal integrity
48	Bad signal integrity
49	Bad signal integrity
50	Internal error
52	Bad signal integrity
55	Internal error
56	module_lanes_frequency_not_synced
57	Signal not detected
60	No partner detected for long time
128	Troubleshooting in process
1023	Information not available
<b>Code</b>	<b>Firmware Management Issues (1024-2047)</b>
1024	Cable is unplugged
1025	Long range for non NVIDIA cable/module
1026	Bus stuck (I <sup>2</sup> C Data or clock shorted)
1027	Bad/unsupported EEPROM
1028	Part number list
1029	Unsupported cable
1030	Module temperature shutdown
1031	Shorted cable
1032	Power budget exceeded
1033	Management force down the port
1034	Module is disabled by command
1035	System Power is Exceeded therefore the module is powered off.
1036	Module's PMD type is not enabled (see PMTPS).
1040	pcie system power slot Exceeded
1042	Module state machine fault
1043-1046	Module's stamping speed degeneration
1047, 1048	Modules DataPath FSM fault
1050-1053	Module Boot Error
1054	Module Forced to Low Power by command

## 8.4.1 Link Diagnostic Commands

- [Link Diagnostic Commands](#)

## 8.4.2 Link Diagnostic Commands

- [8.4.2.1 nv show interface link diagnostics](#)
- [8.4.2.2 nv show interface view link diagnostics](#)

### 8.4.2.1 nv show interface link diagnostics

	nv show interface <interface-id> link diagnostics Display the link diagnostics information of the given interface.	
Syntax Description	interface-id	Name of the interface to display
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~-\$ nv show interface sw1pls1 link diagnostics Code      Status ----- 0          No issue was observed</pre>	
REST API	GET https://<ip>/nvue_v1/interface/<interface id>/link/diagnostics	
Related Commands	nv show interface --view link-diagnostics	
Notes		

### 8.4.2.2 nv show interface view link diagnostics

	nv show interface--view link-diagnostics Display the link diagnostics for all the interfaces.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~-\$ nv show interface --view link-diagnostics Interface Code Status ----- acp1      2      Negotiation failure acp2      2      Negotiation failure fnm1      2      Negotiation failure fnm2      2      Negotiation failure sw1pls1  1024   Cable is unplugged sw1pls2  1024   Cable is unplugged</pre>	
REST API	GET https://<ip>/nvue_v1/interface	
Related Commands	nv show interface link diagnostics	
Notes		

## 8.5 Event Management

- [8.5.1 Supported Events](#)
- [8.5.2 Detailed Table of Events](#)
- [8.5.3 Event Management Commands](#)

The primary objective of incorporating this feature is to address particular actions (such as port disable) or events (like port fast-recovery) and broadcast them in a standardized format along with descriptions. This aims to streamline remote system state monitoring for users.



In this current release, only the CLI is supported for accessing the list of broadcasted events. However, in the upcoming release, gNMI will be expanded to facilitate the remote publication of each event to clients. To subscribe to gNMI events, see [gNMI Streaming](#) section.

### 8.5.1 Supported Events

The following table presents the supported events with their description.

Resource	Event Description	Severity
System	System fatal state detected	CRITICAL
System	System is not ready—one or more services are not up	CRITICAL
System	System is not ready—one or more services failed	MAJOR
System	Restart all syncd-ibv0 dockers	MAJOR
System	Performing reboot	MAJOR
System	Health status is not ok	WARNING
System	System is ready	INFORMATIONAL
System	System recovered from fatal state	INFORMATIONAL
System	Health status is ok	INFORMATIONAL
Sensor or service name	<Repeats a message from the system health>	WARNING
Sensor or service name	Hardware component goes back to normal / Service goes back to normal	INFORMATIONAL
Interface name	Interface admin state is {Up/Down}	INFORMATIONAL
Interface name	Interface operational state is {Up/Down}	INFORMATIONAL
Interface name	Fast-recovery error event for trigger {trigger_name} was received	INFORMATIONAL

## 8.5.2 Detailed Table of Events

Event Category	Event Type ID ("event" in gNMI)	Severity	Resource ("component" in gNMI)	Text	Failure Reason	Suggested Repair Flow
<b>Fan-Related Events</b>						
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"FAN1/1 speed is out of range, speed=40%, range=[50,100]"	Fan speed out of range	<ul style="list-style-type: none"> <li>• Collect tech-support and submit NVIDIA support ticket.</li> <li>• Consider number of faulty fans: more than one fan requires immediate maintenance.</li> <li>• Power-cycle the switch.</li> <li>• If persists, submit NVIDIA support ticket to replace fan module.</li> </ul>
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"FAN1/1 is not working"	Fan status is not okay (status in the hardware)	
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"Failed to get actual speed data for FAN1/1"	Failed to get some information of fan data from hardware	
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"Failed to get target speed data for FAN1/1"	Failed to get some information of fan data from hardware	
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"Failed to get speed tolerance for FAN1/1"	Failed to get some information of fan data from hardware	
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"Failed to get speed status for FAN1/1"	Failed to get some information of fan data from hardware	
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"FAN1/1 is missing"	Fan is missing	
Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"FAN1/1 direction is not aligned with exhaust direction intake"	Fan direction is not aligned with other fans	

Fan failure	HEALTH_NOT_OK	WARNING	FAN1/1	"Invalid fan speed data for FAN1/1, speed=0x, target=50, tolerance=100"	Invalid speed		
Fan health	HEALTH_OK	INFORMATIONAL	FAN1/1	"HW component goes back to normal"	Fan is back to normal state	N/A	
<b>ASIC-Related Events</b>							
ASIC failure	HEALTH_NOT_OK	WARNING	ASIC-HEALTH	Switch ASIC in fatal state	ASIC in fatal	<ul style="list-style-type: none"> <li>• Check correct software and firmware bundle recipe of switch and compute trays.</li> <li>• Collect tech-support and submit NVIDIA support ticket.</li> <li>• Reboot the system.</li> <li>• If persists, power-cycle system.</li> </ul>	
ASIC failure	SYSTEM_FATAL_DETECTED	CRITICAL	System	System fatal state detected	Detect ASIC in fatal		
ASIC failure	HEALTH_NOT_OK	WARNING	ASIC1	ASIC1 temperature is too hot, temperature=120, threshold=105	ASIC temp too high	<ul style="list-style-type: none"> <li>• Collect tech-support and submit NVIDIA support ticket.</li> <li>• Continue to monitor switch temperature.</li> </ul>	
ASIC failure	SYSTEM_FATAL_REMEDY	MAJOR	System	Restart all syncd-ibv0 dockers	ASIC in fatal performing reboot of dockers	N/A	
ASIC failure	SYSTEM_FATAL_REMEDY	MAJOR	System	Performing reboot	ASIC in fatal performing reboot		
ASIC health	HEALTH_OK	INFORMATIONAL	ASIC1	"HW component goes back to normal"	ASIC1 is back to normal state		
ASIC health	SYSTEM_FATAL_RECOVERED	INFORMATIONAL	System	System recovered from fatal state	Recoverd from fatal		
<b>Leakage-Related Events</b>							

Leakage	LEAKAGE	CRITICAL	LEAKAGE-1	Leakage detected, inspect for water leakage and consider power down switch tray	Detected leakage	<ul style="list-style-type: none"> <li>Collect tech-support and submit NVIDIA support ticket.</li> <li>For additional instructions refer to NVONLINE 1115991 chapter "NVIDIA MGX Leak Detection Strategy and Remediation"</li> </ul>
Leakage	HEALTH_NOT_OK	WARNING	LEAKAGE-1	LEAKAGE-1 detected leakage	Detected leakage	

 Relevant only for liquid-cooled-based systems.

### Voltage-Related Events

Voltage	HEALTH_NOT_OK	WARNING	<Voltage-sensor-name>	Sensor voltage is out of range, voltage={}, range=[{},{}]	Voltage sensor not in range	<ul style="list-style-type: none"> <li>Collect tech-support and submit NVIDIA support ticket.</li> <li>Power cycle the switch.</li> <li>If persists, check busbar power supply if the sensor is one of the following: HSC-VinDC-In, PDB-1-Conv-In-1, PDB-2-Conv-In-1, PDB-3-Conv-In-1, PDB-4-Conv-In-1.</li> <li>If persists, consider replacing the system.</li> </ul>
Voltage	HEALTH_NOT_OK	WARNING	<Voltage-sensor-name>	Sensor status is failed	Voltage sensor status in hardware is failed	
Voltage	HEALTH_OK	INFORMATIONAL	<Voltage-sensor-name>	"HW component goes back to normal"	Voltage sensor value is back to normal state	N/A

### Temperature-Related Events

Temperature	HEALTH_NOT_OK	WARNING	<Temp-sensor-name>	<Temp-sensor-name> temperature is too hot, temperature={}, threshold={}	Temperature too hot	<ul style="list-style-type: none"> <li>Collect tech-support and submit NVIDIA support ticket.</li> <li>Power cycle the switch.</li> <li>If persists, see if the sensor is Ambient-MNG-Temp. If it is, check the environmental conditions (CDU and DC temperature).</li> <li>If persists, consider replacing the system.</li> </ul>
Temperature	HEALTH_NOT_OK	WARNING	<Temp-sensor-name>	Sensor status is failed	Sensor status in hardware is failed	
Temperature	HEALTH_OK	INFORMATIONAL	<Temp-sensor-name>	“HW component goes back to normal”	Temperature sensor value is back to normal state	N/A
<b>System-Services-Related Events</b>						
services	HEALTH_NOT_OK	WARNING	<container-name>	Container '<container-name>' is not running	Container is not running	Collect tech-support and submit NVIDIA support ticket.
services	HEALTH_OK	INFORMATIONAL	<container-name>	“Service goes back to normal”	Service goes back to normal state	N/A
<b>System-Initialization-Related Events</b>						
Init flow	SYSTEM_STATE_DOWN	CRITICAL	System	System is not ready—one or more services are not up	Some services are not up as part of init	<ul style="list-style-type: none"> <li>Collect tech-support and submit NVIDIA support ticket.</li> <li>If sensor is Ambient-MNG-Temp:</li> <li>Check environmental conditions (CDU (if exists) and DC temperature).</li> <li>If persists, power-cycle the switch.</li> <li>If still persists, replace the switch.</li> </ul>
Init flow	SYSTEM_STATE_FAILED	MAJOR	System	System is not ready—one or more services failed	Some services failed as part of init	
Init flow	SYSTEM_STATE_UP	INFORMATIONAL	System	System is ready	System finished initialization and is ready	N/A

Interface-Related Informational Events						
interface	INTERFACE_ADMIN_STATUS	INFORMATIONAL	<interface_name>	"Interface admin state is {admin_state}"	Informs of admin state change of interface	N/A
interface	INTERFACE_OPER_STATUS	INFORMATIONAL	<interface_name>	"Interface operational state is {up or down}"	Informs of operational state change of interface	N/A
interface	INTERFACE_LOGICAL_STATE	INFORMATIONAL	<interface_name>	"Interface logical state is {logical_state}"	Informs of logical state change of interface	N/A
System-Health-Related Events (The below events are summary and accompany the specific errors that were detailed above)						
system	HEALTH_SUMMARY_NOT_OK_CRITICAL	CRITICAL	System	Health status is not ok	Have some health not okay event—critical (e.g. leakage)	Collect tech-support and submit NVIDIA support ticket.
system	HEALTH_SUMMARY_NOT_OK	WARNING	System	Health status is not ok	Have some health not okay event—warning	
System	HEALTH_SUMMARY_OK	INFORMATIONAL	System	Health status is ok	System health with no issue	N/A
Transceiver-Related Events						
Transceiver failure	HEALTH_NOT_OK	WARNING	sw1	"Transceiver's temperature is higher than critical threshold [actual = 100, high threshold = 80]"	Temperature is critically high	N/A
Transceiver failure	HEALTH_NOT_OK	WARNING	sw1	"Transceiver's temperature is lower than critical threshold [actual = 1, low threshold = 5]"	Temperature is critically low	N/A

Transceiver health	HEALTH_OK	INFORMATIONAL	sw1	"HW component goes back to normal"	Transceiver's temperature is good now	N/A
--------------------	-----------	---------------	-----	------------------------------------	---------------------------------------	-----

## 8.5.3 Event Management Commands

- [Event Management Commands](#)

## 8.5.4 Event Management Commands

- [8.5.4.1 nv show system events](#)
- [8.5.4.2 nv show system events last](#)
- [8.5.4.3 nv show system events recent](#)
- [8.5.4.4 nv set/unset system events table-size](#)
- [8.5.4.5 nv action clear system events](#)

### 8.5.4.1 nv show system events

	nv show system events Display events generated by the system.
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$nv show system events  ----- operational ----- table-occupancy 80 table-size      1000  Events ===== Event ID  Severity      Component Description ----- 80        INFORMATIONAL System      Health status is ok        2024-04-23 08:29:05 79        INFORMATIONAL PSU2/FAN   HW component goes back to normal   2024-04-23 08:29:05 78        INFORMATIONAL PSU1/FAN   HW component goes back to normal   2024-04-23 08:29:05 77        WARNING       PSU1/FAN   PSU1/FAN speed is out of range, speed=30%, range=[35,100] 2024-04-23 08:29:02 76        WARNING       PSU2/FAN   PSU2/FAN speed is out of range, speed=30%, range=[35,100] 2024-04-23 08:28:56 75        WARNING       PSU2/FAN   PSU2/FAN speed is out of range, speed=29%, range=[35,100] 2024-04-23 08:28:50 74        WARNING       PSU1/FAN   PSU1/FAN speed is out of range, speed=29%, range=[35,100] 2024-04-23 08:28:50 ... </pre>
REST API	GET https://<ip>/nvue_v1/system/events
Related Commands	nv action clear system events
Notes	It shows only the last 50 events in order to not flood the screen with multiple lines at the same time. To get more events, 'last' option can be used.

### 8.5.4.2 nv show system events last

	nv show system events last <number> Show requested last events.	
Syntax Description	number	Requested number of events to show
Default	20 events	
History	25.02.1884	
Example	<pre> admin@nvos:~\$nv show system events last 5 operational ----- table-occupancy 80 table-size      1000  Events =====       Event ID  Severity      Component Description -----       80        INFORMATIONAL  System      Health status is ok          2024-04-23 08:29:05       79        INFORMATIONAL  PSU2/FAN    HW component goes back to normal     2024-04-23 08:29:05       78        INFORMATIONAL  PSU1/FAN    HW component goes back to normal     2024-04-23 08:29:05       77        WARNING       PSU1/FAN    PSU1/FAN speed is out of range, speed=30%, range=[35,100] 2024-04-23 08:29:02       76        WARNING       PSU2/FAN    PSU2/FAN speed is out of range, speed=30%, range=[35,100] 2024-04-23 08:28:56 </pre>	
REST API	GET https://<ip>/nvue_v1/system/events/last/{number}	
Related commands	nv action clear system events	
Notes	If no number is specified, this command will show the last 20 entries from the table.	

### 8.5.4.3 nv show system events recent

	nv show system events recent <minutes> Show events in the last requested minutes.	
Syntax Description	minutes	Time in past minutes to show events from
Default	5 minutes	
History	25.02.1884	
Example	<pre> admin@nvos:~\$nv show system events recent 15 operational ----- table-occupancy 80 table-size      1000  Events =====       Event ID  Severity      Component Description -----       80        INFORMATIONAL  System      Health status is ok          2024-04-23 08:29:05       79        INFORMATIONAL  PSU2/FAN    HW component goes back to normal     2024-04-23 08:29:05       78        INFORMATIONAL  PSU1/FAN    HW component goes back to normal     2024-04-23 08:29:05       77        WARNING       PSU1/FAN    PSU1/FAN speed is out of range, speed=30%, range=[35,100] 2024-04-23 08:29:02 </pre>	

REST API	GET https://<ip>/nvue_v1/system/events/recent/{minutes}
Related commands	nv action clear system events
Notes	If no minutes are specified, this command will display events from the past 5 minutes.

#### 8.5.4.4 nv set/unset system events table-size

	nv set/unset system events table-size <number-of-lines> Set/unset events table size.	
Syntax Description	number-of-lines	Number of lines shown in the events table
Default	1000 lines	
History	25.02.1884	
Example	<pre>admin@nvos:~\$nv set system events table-size 5000 admin@nvos:~\$nv unset system events table-size</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/events/table-size/{number-of-lines}	
Related commands	nv show system events	
Notes		

#### 8.5.4.5 nv action clear system events

	nv action clear system events Clear events.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$nv action clear system events Action executing ... Event table has been cleared Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system/events	
Related Commands	nv show system events	
Notes		

## 8.6 Statistics

NVOS collects samples and saves system statistics data. Sampling data is collected for tech-support.

Supporting CLI:

- Show, upload, clear data collected.
- Configure sampling configuration.

- Enable/disable sampling
- Sampling interval
- How long to save samples (cyclic buffer)

Statistics are split into multiple categories:

Category	Statistic	Description
Temperature	asic-<x>	Asic temperature (per asic)
	ambient COMEX	Ambient comex temperature
	Ambient Fan Side	Ambient fan side temperature
	Ambient Port Side	Ambient port-side temperature
	CPU Core <x>	CPU core temperature (per cpu)
	CPU Pack	CPU pack temperature
	PSU-<x>	PSU temperature (per PSU)
	Module <idx>	Module temperature
CPU	Free-ram	Free system ram memory
	CPU utilization	CPU utilization over the last 5 minutes (can be either 1/5/15)
	Reboot count	Number of reboots
Disk	Free space	Free space under "/"
	Wear level	Disk remaining time left
	Write rate	Average disk write/sec
	Read rate	Average disk read/sec
	Program fail count	Counts the number of flash program failures
	Erase fail count	counts the number of failed data deletion attempts
	Uncorrectable error count	count of errors that are impossible to recover
Power	Psu-power <x>	PSU power [W]
	Psu-current <x>	PSU current [A]
Fan	Fan-speed <x>	Fan speed [%]
Mgmt-interface	Eth<x> tx bytes	TX bytes diff from last sample
	Eth<x> rx bytes	RX bytes diff from last sample

## 8.6.1 Statistics Commands

- [Statistics Commands](#)

## 8.6.2 Statistics Commands

- [8.6.2.1 nv show system stats](#)

- [8.6.2.2 nv show system stats category](#)
- [8.6.2.3 nv show system stats files](#)
- [8.6.2.4 nv set/unset system stats state](#)
- [8.6.2.5 nv set/unset system stats category state](#)
- [8.6.2.6 nv set/unset system stats category interval](#)
- [8.6.2.7 nv set/unset system stats category history-duration](#)
- [8.6.2.8 nv action clear system stats category](#)
- [8.6.2.9 nv action clear system stats category](#)
- [8.6.2.10 nv action generate system stats category](#)
- [8.6.2.11 nv action generate system stats](#)
- [8.6.2.12 nv action clear system stats](#)
- [8.6.2.13 nv action delete system stats files](#)
- [8.6.2.14 nv action upload system stats files](#)

### 8.6.2.1 nv show system stats

	nv show system stats Display configuration for system statistics collection reports.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show system stats               operational  applied ----- state        enabled      enabled [category]   cpu [category]   disk [category]   fan [category]   mgmt-interface [category]   temperature </pre>	
REST API	GET https://<ip>/nvue_v1/system/stats	
Related Commands	nv set/unset system stats state	
Notes		

### 8.6.2.2 nv show system stats category

	nv show system stats category [<category-id>] Get statistic categories.	
Syntax Description	category-id	Display configuration for specified category.
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show system stats category ----- History Duration (days) Interval (minutes) State ----- cpu 365 5 enabled disk 365 30 enabled fan 365 5 enabled mgmt-interface 365 5 enabled temperature 365 5 enabled</pre> <pre>admin@nvos:~\$ nv show system stats category fan ----- operational applied ----- history-duration 365 365 interval 5 5 state enabled enabled</pre>
REST API	GET https://<ip>/nvue_v1/system/stats/category GET https://<ip>/nvue_v1/system/stats/category/< category-id >
Related Commands	nv set/unset system stats category
Notes	

### 8.6.2.3 nv show system stats files

	nv show system stats files [<file-name>] Display statistic report files.	
Syntax Description	file-name	Display content of specified CSV file.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show system stats files Statistic report file File path ----- stats_fan_gorilla-62_20230704_112407.csv /host/stats/ stats_fan_gorilla-62_20230704_112407.csv</pre> <pre>admin@nvos:~\$ nv show system stats files stats_fan_gorilla-62_20230704_112407.csv</pre>	
REST API	GET https://<ip>/nvue_v1/system/stats/files GET https://<ip>/nvue_v1/system/stats/files/< file-name >	
Related Commands	nv action generate system stats category	
Notes	Autocomplete returns also generated tar file (in case if it was generated before), but it can't be displayed, only *.csv files can be displayed	

### 8.6.2.4 nv set/unset system stats state

	nv set/unset system stats state [{ enabled   disabled }] Set/unset state.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv set system stats state disabled</pre> <pre>admin@nvos:~\$ nv unset system stats state</pre>
REST API	PATCH https://<ip>/nvue_v1/system/stats
Related Commands	nv show system stats
Notes	

### 8.6.2.5 nv set/unset system stats category state

	nv set/unset system stats category <category-id> state [{ enabled   disabled }] Set/unset category state.	
Syntax Description	category-id	Change state for specified category.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system stats category fan state disabled</pre> <pre>admin@nvos:~\$ nv unset system stats category fan state</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/stats/category/<category-id>	
Related Commands	nv show system stats category	
Notes		

### 8.6.2.6 nv set/unset system stats category interval

	nv set/unset system stats category <category-id> interval Set interval in minutes.	
Syntax Description	category-id	Change sampling interval for specified category.
Default	5 (30 for disk category)	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system stats category fan interval 60</pre> <pre>admin@nvos:~\$ nv unset system stats category fan interval</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/stats/category/<category-id>	
Related Commands	nv show system stats category	
Notes	Sampling interval is in minutes.	

### 8.6.2.7 nv set/unset system stats category history-duration

	nv set/unset system stats category <category-id> history-duration[ <b>{ 1-365 }</b> ] Sampling history duration in days.	
Syntax Description	category-id	Change history-duration for specified category.
	history-duration	1-365 days
Default	365	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set system stats category fan history-duration 31</pre> <pre>admin@nvos:~\$ nv unset system stats category fan history-duration</pre>	
REST API	PATCH https://<ip>/nvue_v1/system/stats/category/<category-id>	
Related Commands	nv show system stats category	
Notes	History duration is in days.	

### 8.6.2.8 nv action clear system stats category

	nv action clear system stats category <category-id> Clear statistic category sampled data.	
Syntax Description	category-id	Category to be cleared.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action clear system stats category fan</pre>	
REST API	POST https://<ip>/nvue_v1/system/stats/category/<category-id>	
Related Commands	nv action generate system stats category	
Notes		

### 8.6.2.9 nv action clear system stats category

	nv action clear system stats category <category-id> Clear statistic category sampled data.	
Syntax Description	category-id	Category to be cleared.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action clear system stats category fan</pre>	

REST API	POST https://<ip>/nvue_v1/system/stats/category/<category-id>
Related Commands	nv action generate system stats category
Notes	

### 8.6.2.10 nv action generate system stats category

	nv action generate system stats category <category-id> Generate statistic report file.	
Syntax Description	category-id	Category to be generated CSV file for.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action clear system stats category fan</pre>	
REST API	POST https://<ip>/nvue_v1/system/stats/category/<category-id>	
Related Commands	nv show system stats files	
Notes		

### 8.6.2.11 nv action generate system stats

	nv action generate system stats Generate tar file with reports from all categories.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action generate system stats</pre>	
REST API	POST https://<ip>/nvue_v1/system/stats	
Related Commands	nv show system stats files	
Notes		

### 8.6.2.12 nv action clear system stats

	nv action clear system stats Clear statistic sampled data for all categories.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv action clear system stats</pre>
REST API	POST https://<ip>/nvue_v1/system/stat
Related Commands	nv action generate system stats
Notes	

### 8.6.2.13 nv action delete system stats files

	nv action delete system stats files <file-name> Delete available stats report file.	
Syntax Description	file-name	Report file to be deleted.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action delete system stats files_stats_fan_gorilla-62_20230704_112407.csv</pre>	
REST API	POST https://<ip>/nvue_v1/system/stats/files/<file-name>	
Related Commands	nv action generate system stats category nv show system stats files	
Notes		

### 8.6.2.14 nv action upload system stats files

	nv action upload system stats files <file-name> <remote-url> Upload available stats report files to remote location.	
Syntax Description	file-name	Report file name to be uploaded.
	remote-url	FTP, SCP and SFTP are supported (e.g., scp://username[:password]@hostname/path/filename)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action upload system stats files stats_fan_gorilla-62_20230704_112407.csv scp://user@host/path/to/folder</pre>	
REST API	POST https://<ip>/nvue_v1/system/stats/files/<file-name>	
Related Commands	nv show system stats files	
Notes		

## 8.7 Technical Support

NVOS supports generating technical support files to assist with troubleshooting and issue resolution. These files contain critical system logs and diagnostic information that help support teams analyze and address reported issues efficiently.

It is recommended that users generate a technical support file whenever they encounter an issue and include it when filing a support request.

### 8.7.1 How to Generate and Upload a Tech-Support File

For troubleshooting your transceiver, generating and uploading a tech-support file is essential. This file includes detailed diagnostics to aid technical support in resolving your issues.

1. Generate the file.

```
admin@nvos:~$ nv action generate system tech-support
Generating system tech-support file, it might take a few minutes...
Generated tech-support /var/dump/nvos_dump_nvos_20220601_114011.tar.gz
Action succeeded
```

2. Upload generated file to a designated location.

```
admin@nvos:~$ nv action upload system tech-support files nvos_dump_nvos-switch_20230120_151401.tar.gz
scp://username@host-name/home/logs/
Password:
Action executing ...
Upload file nvos_dump_nvos-switch_20230120_151401.tar.gz
Action executing ...
File upload successfully
Action succeeded
```

### 8.7.2 Technical Support Commands

- [Technical Support Commands](#)

### 8.7.3 Technical Support Commands

- [8.7.3.1 nv show system tech-support files](#)
- [8.7.3.2 nv action generate system tech-support](#)
- [8.7.3.3 nv action delete system tech-support files](#)
- [8.7.3.4 nv action upload system tech-support files](#)

#### 8.7.3.1 nv show system tech-support files

	nv show system tech-support files Show the created tech-support files on the switch.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show system tech-support files File name                               Tech-support file path ----- nvos_dump_nvos-switch_20230120_151401.tar.gz /host/dump/nvos_dump_nvos- switch_20230120_151401.tar.gz</pre>
REST API	GET https://<ip>/nvue_v1/system/tech-support/files
Related Commands	nv action generate system tech-support nv action delete system tech-support files
Notes	

### 8.7.3.2 nv action generate system tech-support

	nv action generate system tech-support [--since <time>] Action to generate a tech-support file on switch.	
Syntax Description	--since	Collect logs and cores only from the given since date
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action generate system tech-support Generating system tech-support file, it might take a few minutes... Generated tech-support /var/dump/nvos_dump_jaguar-70_20220601_114011.tar.gz Action succeeded  admin@nvos:~\$ nv action generate system tech-support --since=1/1/2022 Generating system tech-support file, it might take a few minutes... Generated tech-support /var/dump/nvos_dump_jaguar-70_20220601_114315.tar.gz Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system/tech-support	
Related Commands	nv show system tech-support files nv action delete system tech-support files nv action upload system tech-support files	
Notes		

### 8.7.3.3 nv action delete system tech-support files

	nv action delete system tech-support files <file-name> Delete tech-support file from the file system.	
Syntax Description	file-name	Name of the tech-support file.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action delete system tech-support files nvos_dump_nvos- switch_20230120_151401.tar.gz File delete successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system/tech-support/files/{file-name}	

Related Commands	nv action show system tech-support files nv action generate system tech-support nv action upload system tech-support files
Notes	

### 8.7.3.4 nv action upload system tech-support files

	nv action upload system tech-support files <file-name> <remote-url> Upload tech-support file to the remote server.	
Syntax Description	file-name	Name of the tech-support file.
	remote-url	ftp, tftp, scp, and sftp are supported (e.g., scp://username[:password]@hostname/path/filename)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action upload system tech-support files nvos_dump_nvos- switch_20230120_151401.tar.gz scp://username@host-name/home/logs/ Password: Upload file 1 File upload successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/system/tech-support/files/{file-name}	
Related Commands	nv action show system tech-support files nv action generate system tech-support nv action delete system tech-support files	
Notes		

## 8.8 Troubleshooting

- [8.8.1 Resetting NVOS Password](#)
- [8.8.2 Resetting BMC Root User Password](#)
- [8.8.3 Image Upgrade Recovery](#)
- [8.8.4 System Fatal Recovery](#)
  - [8.8.4.1 Detecting a Fatal State](#)
  - [8.8.4.2 Automatic Recovery Mechanism](#)
  - [8.8.4.3 Recovery Timeframe](#)

### 8.8.1 Resetting NVOS Password

To reset forgotten password of default user accounts, see [Reset Local Users' Passwords](#) section.

### 8.8.2 Resetting BMC Root User Password

To reset the BMC root user password, use the [nv action reset platform bmc-password](#) command.

## 8.8.3 Image Upgrade Recovery

If the system encounters issues after an image upgrade, the user can switch back to the old partition.

1. Check the current partition.

```
admin@nvos:~$ nv show system image
operational
-----
current nvos-25.02.1500
next    nvos-25.02.1500
partition1 nvos-25.02.1500
partition2 nvos-25.02.1400
```

2. Change to the other partition.

```
admin@nvos:~$ nv action boot-next system image partition2
admin@nvos:~$ nv show system image
operational
-----
current nvos-25.02.1500
next    nvos-25.02.1400
partition1 nvos-25.02.1500
partition2 nvos-25.02.1400
```

3. Reboot the system.

```
admin@nvos:~$ nv action reboot system
```

## 8.8.4 System Fatal Recovery

The system has mechanism to detect if ASIC encountered health/firmware burn issue and try to recover from it.

During the fatal detection and recovery, events will be raised as well. For more information, see [ASIC-Related Events](#) in the [Event Management](#) section.

### 8.8.4.1 Detecting a Fatal State

The system's fatal state is indicated in the CLI prompt and in the [nv show system health](#) command. Example:

```
[System_Fatal_State]admin@nvos~$ nv show system health
operational applied
-----
status      FATAL
status-led  amber

Health issues
=====
Component   Status information
-----
ASIC-HEALTH Switch ASIC in fatal state.
```

### 8.8.4.2 Automatic Recovery Mechanism

The system has an internal mechanism to recover from a fatal state without user intervention. The recovery process involves the following steps:

1. Restart the ASICs of the system.
2. If restarting the ASICs does not resolve the issue, the system will attempt to recover through a system reboot.  
If after the reboot system still encounters ASIC issues, another reboot will be performed.
3. After the second reboot, the system will start without configuring the ASICs, leaving all ports down. NVOS is running, so logs can be collected for analysis.
4. To try to revive the switch, perform a power-cycle by the running the command [nv action power-cycle system](#).
5. If system entered fatal state again, please contact NVIDIA's support team.



Any reboot or power cycle initiated by the user will also reset the system's fatal detection and recovery mechanism. This process starts the recovery steps from the beginning.

#### 8.8.4.3 Recovery Timeframe

- After the recovery steps are completed and the system remains operational for 10 minutes without any health issues, it will exit the fatal state.
- During this 10-minute observation period, the system may still appear in a fatal state as reflected in the CLI prompt and system health command.
- Once the system exits the fatal state, the CLI prompt and system health command will confirm the recovery.

---

## 9 NVLink Switching

The following pages provide information on configuring NVLink protocols and features.

- [NVLink Interface](#)
- [NVLink Fabric](#)
- [Cluster Management](#)

### 9.1 NVLink Interface

NVLink switch tray has groups of NVLink interfaces with different roles:

Role	Name Example	Purpose
Access (backplane)	acp1	Establish connectivity between switch trays and compute trays in the same rack.
Front-panel	sw1p1s1 <ul style="list-style-type: none"><li>• sw&lt;x&gt; is the physical port number found on the front-panel</li><li>• p&lt;x&gt;s&lt;x&gt; is the logical port</li></ul>	When applicable, allow connectivity between switches from different racks.
Fabric Network Management (FNM)	fnm1	Establish connectivity between switch trays in the same rack, used for network provisioning by SDN entities (e.g., subnet manager).

#### 9.1.1 Configuring and Monitoring Interfaces

NVOS provides tools to configure and monitor network interfaces, ensuring seamless connectivity and performance.

- Enable/disable interfaces (all interfaces are enabled by default).

Examples:

```
admin@nvos:~$ nv set interface acp1 link state up
admin@nvos:~$ nv set interface acp1 link state down
```

- Set interface descriptions for easier identification and management.

Example:

```
admin@nvos:~$ nv set interface acp1 description "Access-1"
```

- Show link state, diagnostics, and counters.

```
admin@nvos:~$ nv show interface
admin@nvos:~$ nv show interface link-diagnostics
admin@nvos:~$ nv show interface acp1
admin@nvos:~$ nv show interface acp1 link counters
admin@nvos:~$ nv show interface acp1 link phy-detail
```

#### 9.1.2 NVLink Interface Commands

- [NVLink Interface Commands](#)

## 9.1.3 NVLink Interface Commands

- [9.1.3.1 nv show interface](#)
- [9.1.3.2 nv show interface link](#)
- [9.1.3.3 nv set interface description](#)
- [9.1.3.4 nv set interface link state](#)
- [9.1.3.5 nv action clear interface link counters](#)
- [9.1.3.6 nv action clear interface counters](#)

### 9.1.3.1 nv show interface

	nv show interface <interface-id> Displays details of a single NVLink interface.	
Syntax Description	interface-id	Name of the NVLink interface to display.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show interface acp21 ----- operational  applied link auto-negotiate  on speed           400G counters   in-bytes      108.00 KB   in-pkts       384   in-drops      0   in-errors     0   out-bytes     108.00 KB   out-pkts      384   out-drops     0   out-errors    0   in-symbol-errors  0   out-wait      0   link-error-recovery  0   link-downed   0   port-rcv-remote-physical-errors  0   port-rcv-switch-relay-errors     0   port-rcv-constraint-errors       0   local-link-integrity-errors      0   qpl-drops                         0   buffer-overflow-errors            0   rcv-icrc-errors                   0   tx-parity-errors                   0   unicast-in-pkts                    0   unicast-out-pkts                    0   multicast-in-pkts                   0   multicast-out-pkts                  0 [diagnostics] lanes           2X state           up supported-speed 400G max-supported-mtu 4096 physical-state  Polling logical-state   Active supported-lanes 2X vl-capabilities VL0-VL7 type           nv1          nv1 </pre>	
REST API	GET https://<ip>/nvue_v1/interface/{interface-id}	
Related Commands	set interface show interface management	
Notes	The data presented here is for an NVLink interface. If the id of a different type of interface is provided, the output will be different.	

### 9.1.3.2 nv show interface link

	nv show interface <interface-id> link {state   counters   phy-diag   phy-detail} Displays link information of a single NVLink interface.	
Syntax Description	interface-id	Name of the NVLink interface to display.
	state	Show only the data relating to state.
	counters	Show only the data relating to counters.
	phy-diag	Show PHY diagnostics and FSM states, part of AMBER.
	phy-detail	Show PHY statistics and BER measurements, part of AMBER.
Default	N/A	
History	25.02.1884	

Example

```
admin@nvos:~$ nv show interface acp21 link
-----
operational  applied
-----
auto-negotiate  on
speed          400G
counters
  in-bytes      108.00 KB
  in-pkts       384
  in-drops      0
  in-errors     0
  out-bytes     108.00 KB
  out-pkts      384
  out-drops     0
  out-errors    0
  in-symbol-errors  0
  out-wait      0
  link-error-recovery  0
  link-downed   0
  port-rcv-remote-physical-errors  0
  port-rcv-switch-relay-errors      0
  port-rcv-constraint-errors        0
  local-link-integrity-errors        0
  qpl-drops                            0
  buffer-overflow-errors              0
  rcv-icrc-errors                     0
  tx-parity-errors                     0
  unicast-in-pkts                      0
  unicast-out-pkts                     0
  multicast-in-pkts                     0
  multicast-out-pkts                    0
[diagnostics]
lanes          2X
state          up
supported-speed  400G
max-supported-mtu  4096
physical-state  LinkUp
logical-state   Active
supported-lanes  2X
vl-capabilities VL0-VL7
```

```
admin@nvos:~$ nv show interface acp21 link state
operational  applied
-----
up          up

admin@nvos:~$ nv show interface acp21 link counters
-----
operational
-----
in-bytes      108.00 KB
in-pkts       384
in-drops      0
in-errors     0
out-bytes     108.00 KB
out-pkts      384
out-drops     0
out-errors    0
in-symbol-errors  0
out-wait      0
link-error-recovery  0
link-downed   0
port-rcv-remote-physical-errors  0
port-rcv-switch-relay-errors      0
port-rcv-constraint-errors        0
local-link-integrity-errors        0
qpl-drops                            0
buffer-overflow-errors              0
rcv-icrc-errors                     0
tx-parity-errors                     0
unicast-in-pkts                      0
unicast-out-pkts                     0
multicast-in-pkts                     0
multicast-out-pkts                    0
```

```
admin@nvos:~$ nv show interface acp21 link phy-diag
-----
operational
-----
pd-fsm-state      0
eth-an-fsm-state  0
phy-hst-fsm-state LINKUP
psi-fsm-state     IDLE
phy-manager-link-width-enabled  256
phy-manager-link-protocol-enabled  1:20000
core-to-phy-link-width-enabled  256
core-to-phy-link-protocol-enabled  1:800000
cable-protocol-cap-ext  1:0
loopback-mode     NONE
retran-mode-request  0
retran-mode-active  0
fec-mode-request  2048
profile-fec-in-use  255
phy-manager-state  ACTIVE
sync-header-error-counter  0
port-local-physical-errors  0
```

	<pre> port-malformed-packet-errors      0 port-buffer-overflow-errors       0 port-dlid-mapping-errors          0 port-vl-mapping-errors            0 port-looping-errors               0 port-inactive-discards            0 port-neighbor-mtu-discards        0 plr-rcv-codes                     73634260371 plr-rcv-codes-err                 0 plr-rcv-uncorrectable-code        0 plr-xmit-codes                    73634123139 plr-xmit-retry-codes              0 plr-xmit-retry-events             0 plr-sync-events                   0 plr-codes-loss                    0 plr-xmit-retry-events-within-t-sec-max 0 plr-bw-loss-percent               0.0 rq-general-error                  0 ib-phy-fsm-state                  DISABLED zero-hist                          5 successful-recovery-events        0  admin@nvos:~\$ nv show interface acp21 link phy-detail ----- time-since-last-clear-min 1945856 phy-received-bits        778342400000000 symbol-errors            0 effective-errors         0 phy-raw-errors-lane0    1906686 phy-raw-errors-lane1    1659554 phy-raw-errors-lane2    0 phy-raw-errors-lane3    0 phy-raw-errors-lane4    0 phy-raw-errors-lane5    0 phy-raw-errors-lane6    0 phy-raw-errors-lane7    0 raw-ber                  4E-9 symbol-ber               15E-255 effective-ber            15E-255 raw-ber-lane0            4E-9 raw-ber-lane1            4E-9 raw-ber-lane2            0E-0 raw-ber-lane3            0E-0 raw-ber-lane4            0E-0 raw-ber-lane5            0E-0 raw-ber-lane6            0E-0 raw-ber-lane7            0E-0 rs-num-corr-err-bin0     151471657460 rs-num-corr-err-bin1     1690966 rs-num-corr-err-bin2     937603 rs-num-corr-err-bin3     16 rs-num-corr-err-bin4     5 rs-num-corr-err-bin5     0 rs-num-corr-err-bin6     0 rs-num-corr-err-bin7     0 rs-num-corr-err-bin8     0 rs-num-corr-err-bin9     0 rs-num-corr-err-bin10    0 rs-num-corr-err-bin11    0 rs-num-corr-err-bin12    0 rs-num-corr-err-bin13    0 rs-num-corr-err-bin14    0 rs-num-corr-err-bin15    0 </pre>
REST API	<pre> GET https://&lt;ip&gt;/nvue_v1/interface/{interface-id}/link GET https://&lt;ip&gt;/nvue_v1/interface/{interface-id}/link/counters GET https://&lt;ip&gt;/nvue_v1/interface/{interface-id}/link/state GET https://&lt;ip&gt;/nvue_v1/interface/{interface-id}/link/phy-diag GET https://&lt;ip&gt;/nvue_v1/interface/{interface-id}/link/phy-detail </pre>
Related Commands	set interface link
Notes	

### 9.1.3.3 nv set interface description

	<pre> nv set interface &lt;interface-id&gt; description {value} nv unset interface &lt;interface-id&gt; description {value} </pre> <p>Sets the description of a given NVLink interface. The unset sets the description of a given NVLink interface to empty.</p>	
Syntax Description	interface-id	Name of the interface whose description to set.
	Value	New value for the description.
Default	N/A	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv set interface sw1p1s1 description "sw1p1s1 description" admin@nvos:~\$ nv unset interface sw1p1s1 description</pre> <pre>admin@nvos:~\$ nv unset interface sw1p1s1 admin@nvos:~\$ nv unset interface</pre>
REST API	PATCH https://<ip>/nvue_v1/interface/{interface-id}
Related Commands	nv show interface
Notes	

### 9.1.3.4 nv set interface link state

	nv set interface <interface-id> link state {value} nv unset interface <interface-id> link state {value} Sets the administrative link state of a given NVLink interface. The unset sets the administrative link state of a given NVLink interface to the default value of "up".	
Syntax Description	interface-id	Name of the interface whose link state to set.
	Value	New value for the link state: {up, down}
Default	up	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set interface sw1p1s1 link state down admin@nvos:~\$ nv unset interface sw1p1s1 link state</pre>	
REST API	PATCH https://<id>/nvue_v1/interface/{interface-id}/link/state	
Related Commands	show interface nv unset interface link	
Notes		

### 9.1.3.5 nv action clear interface link counters

	nv action clear interface <interface-id> link counters Clears the interface counters for the user running the command.	
Syntax Description	interface-id	Name of the interface whose link stats to clear. or range of interfaces (e.g., sw1-2p1-2 → sw1p1,sw1p2,sw2p1,sw2p2)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action clear interface sw1p1s1 link counters</pre>	
REST API	POST https://<ip>/nvue_v1/interface/{interface-id}/link/counters	
Related Commands	nv show interface <id> link counters	

Notes	
-------	--

### 9.1.3.6 nv action clear interface counters

	nv action clear interface counters Clears all NVLink interfaces counters for the user running the command.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action clear interface counters</pre>	
REST API	POST https://<ip>/nvue_v1/interface	
Related Commands		
Notes		

## 9.2 NVLink Fabric

### 9.2.1 NVLink Fabric Commands

- [NVLink Fabric Commands](#)

### 9.2.2 NVLink Fabric Commands

#### 9.2.2.1 show ib device

	nv show ib device [device-id] Shows devices information.	
Syntax Description	device-id	Optional: show information only for the specified device
Default	N/A	
Configuration Mode	config	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show ib device Device  Type      IB Subnet      GUID                      LID -----  - ASIC1   NVLink-5  infiniband-default  00:00:00:01:FF:FF:00:00  0 ASIC2   NVLink-5  infiniband-default  00:00:00:01:FF:FF:01:00  0 SYSTEM  NVLink-5  infiniband-default  9C:63:C0:03:00:72:B2:12</pre>	
REST API	GET https://<ip>/nvue_v1/ib/device GET https://<ip>/nvue_v1/ib/device/{device-id}	
Related Commands		
Notes		

## 9.3 Cluster Management

- [9.3.1 Cluster Infrastructure and Cluster Applications](#)
  - [9.3.1.1 Installation and Upgrade](#)
  - [9.3.1.2 Management and Configuration](#)
  - [9.3.1.3 Persistence and Recovery](#)
- [9.3.2 NMX-Controller](#)
- [9.3.3 NMX-Telemetry](#)
  - [9.3.3.1 Key Features](#)
- [9.3.4 Cluster Provisioning Flow](#)
- [9.3.5 Partition Management](#)
- [9.3.6 Protocol Buffers](#)

### 9.3.1 Cluster Infrastructure and Cluster Applications

NVOS includes a robust infrastructure that enables the execution of cluster applications on the CPU of the switch. This release supports two cluster applications: NMX-Controller and NMX-Telemetry. The NVOS cluster infrastructure streamlines the management and monitoring of these applications, providing a seamless user experience.

#### 9.3.1.1 Installation and Upgrade

The NVOS cluster infrastructure includes the cluster applications package files within the NVOS image. The packages are automatically installed along with the NVOS image installation and upgrade process, ensuring a hassle-free setup.

#### 9.3.1.2 Management and Configuration

The NVOS cluster infrastructure provides a user-friendly Command Line Interface (CLI) and RESTful APIs to manage and configure the cluster applications. Users can perform the following tasks:

- Start and stop the execution of the cluster applications
- Manage the log verbosity level of the cluster applications
- Configure common functionalities across all cluster applications, such as the gRPC connection with an external manager
- Monitor the operational health of the cluster applications

The gRPC connection with the external manager supports three modes: unencrypted, TLS, and mTLS. The cluster applications act as the server-side of the gRPC connection. For encrypted gRPC modes (TLS and mTLS), the cluster applications facilitate the installation of security certificates and support key rotations to maintain a secure communication channel.

#### 9.3.1.3 Persistence and Recovery

Once the user starts the operation of the cluster applications, any subsequent NVOS boot will automatically restart the cluster applications, ensuring continuous availability. Additionally, upgrading the NVOS will also upgrade the cluster applications seamlessly. User configurations for the cluster infrastructure persist across NVOS reboots and upgrades, eliminating the need for manual

reconfiguration. In case of a factory reset of the NVOS, the cluster infrastructure and cluster applications will also be reset to their default state.

## 9.3.2 NMX-Controller

The NMX-Controller is a cluster application for fabric SDN services. In the GB200 NVL the SDN services are the subnet manager (SM) and global fabric manager (GFM).

The SDN services configuration and operation is managed via either gRPC interface with external manager or the NVUE CLIs and Rest-APIs.

## 9.3.3 NMX-Telemetry

NMX-T is a robust and scalable telemetry service application designed to collect, aggregate, filter, and stream telemetry data from various sources, such as network devices, compute nodes and various sensors, using multiple protocols. Its primary objective is to provide a centralized platform for ingesting and processing telemetry data, enabling real-time monitoring, analysis, and decision-making across diverse systems and applications.

### 9.3.3.1 Key Features

1. **Multi-Source Data Collection:** NMX supports ingesting telemetry data from a wide range of sources, including applications, network devices, sensors, and more. It can handle various protocols such as HTTP, gNMI, OTLP, Redfish, Syslog, and custom protocols, ensuring seamless integration with existing infrastructure.
2. **Data Aggregation and Filtering:** NMX employs advanced data aggregation and filtering techniques to process incoming telemetry data streams. It can aggregate data from multiple sources, apply custom filters based on predefined rules or conditions, and perform data transformations as needed. This feature enables efficient data management and reduces the overhead of processing irrelevant or redundant data.
3. **Real-Time Streaming:** NMX provides real-time streaming capabilities, allowing interested clients to subscribe to specific telemetry data streams. It leverages high-performance messaging protocols, such as gRPC, to ensure low-latency delivery of telemetry data to downstream consumers, such as monitoring tools, analytics platforms, or custom applications.
4. **Scalability and High Availability:** NMX is designed to handle large volumes of telemetry data and can scale horizontally to accommodate increasing data loads. It supports load balancing, failover mechanisms, and distributed deployment architectures, ensuring high availability and fault tolerance.
5. **Extensible Plugin Architecture:** NMX features a modular and extensible plugin architecture, enabling developers to create custom plugins for data ingestion, processing, and output. This flexibility allows NMX to adapt to new protocols, data formats, and integration requirements as needed.
6. **Security and Access Control:** NMX incorporates robust security measures, including authentication, authorization, and encryption mechanisms, ensuring that only authorized clients can access and consume specific telemetry data streams.
7. **Monitoring and Observability:** NMX provides comprehensive monitoring and observability capabilities, allowing administrators to track system health, performance metrics, and

operational insights. It integrates with popular monitoring tools like Prometheus and Grafana, enabling real-time visualization and alerting.

## 9.3.4 Cluster Provisioning Flow

In a domain of NVLink5 cluster, cluster function can be enabled on one of the NVL switches. More information about Cluster Provisioning Flow can be found in the "Cluster Provisioning Flow" section of the Nvidia GB200 NVL System Bring-up Guide that is part of the documentation package.

## 9.3.5 Partition Management

Partitions enable the creation of groups of GPUs dedicated to a specific tenant or workload, ensuring isolation from a security perspective. This means there is no communication path between GPUs assigned to different partitions. Within a partition, GPUs have multicast groups and shared memory mapping to enhance communication efficiency.

The NMX-Controller offers APIs with the following functionalities:

- Create a partition
- Add or remove GPUs from a partition
- View partition information
- Delete partitions

## 9.3.6 Protocol Buffers

For information about protobuf file, see the attached file. To view its content, use any standard OpenAPI viewer tool.

## 9.3.7 Cluster Control

NVOS allows users to view and control the cluster state.

Enabling the cluster starts all cluster applications, while disabling it stops them. The default state of the cluster is disabled.

### 9.3.7.1 Chassis ID Update

To update the chassis ID, run the following command:

```
admin@nvos:~$ nv action update cluster chassis-id <id>
```



Upon changing chassis-id, user must rerun nmx-controller app by stopping and running the application again. See [nv action update cluster chassis-id](#) command, for more information.

### 9.3.7.2 Cluster Control Commands

- [Cluster Control Commands](#)

### 9.3.7.3 Cluster Control Commands

- [9.3.7.3.1 nv show cluster](#)
- [9.3.7.3.2 nv set/unset cluster state](#)
- [9.3.7.3.3 nv action update cluster chassis-id](#)

#### 9.3.7.3.1 nv show cluster

	nv show cluster Display the settings of cluster.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show cluster           operational  applied -----  - state    enabled      enabled nmxc-conn up</pre>	
REST API	GET https://<ip>/nvue_v1/cluster	
Related Commands	nv set cluster state {enabled disabled}	
Notes	The field nmxc-conn show the status of the connection to NMX-Controller.	

#### 9.3.7.3.2 nv set/unset cluster state

	nv set cluster state {enabled   disabled} nv unset cluster state Enable/disable cluster functionality on a switch. The unset form of the command sets cluster functionality to default.	
Syntax Description	enabled	Enable cluster functionality
	disabled	Disable cluster functionality
Default	Disabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv set cluster state enabled admin@nvos:~\$ nv config apply</pre>	
REST API	PATCH https://<ip>/nvue_v1/cluster	
Related Commands	nv show cluster	
Notes	Enabling/Disabling cluster will run/stop all cluster applications automatically.	

### 9.3.7.3.3 nv action update cluster chassis-id

	nv action update cluster chassis-id <chassis-id> Update Chassis ID.	
Syntax Description	chassis-id	Chassis ID (integer)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action update cluster chassis-id 1 Action executing ... Chassis-id 1 is successfully updated Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/chassis-id	
Related Commands	nv show platform chassis-location	
Notes	<ul style="list-style-type: none"><li>• When cluster is disabled, action will fail.</li><li>• When chassis-sn is N/A, action will fail.</li></ul>	

## 9.3.8 Cluster Applications

NVOS provides commands to manage cluster applications, offering users visibility into the applications' status and control over their operation and log settings.

Current supported applications:

- NMX-Controller
- NMX-Telemetry

### 9.3.8.1 Key Functionalities

Viewing Applications State

- Display the status of all cluster applications or a specific application:
  - [nv show cluster apps](#)
  - [nv show cluster apps <app-name>](#)
- View lists of installed and currently running cluster applications:
  - [nv show cluster apps installed](#)
  - [nv show cluster apps running](#)

Controlling Application State

- Start or stop cluster applications:
  - [nv action start cluster apps <app-name>](#)
  - [nv action stop cluster apps <app-name>](#)

Managing Log Levels

- View the current log level for each application:
  - [nv show cluster apps <app-name> log-level](#)
- Update the log level of an application:

- [nv action update cluster apps <app-name> log-level <log-level>](#)
- Restore the log level of an application to its default settings:
  - [nv action restore cluster apps <app-name> log-level <log-level>](#)

## 9.3.8.2 Cluster Applications Commands

- [Cluster Applications Commands](#)

## 9.3.8.3 Cluster Applications Commands

- [9.3.8.3.1 nv show cluster apps](#)
- [9.3.8.3.2 nv show cluster apps name](#)
- [9.3.8.3.3 nv show cluster apps installed](#)
- [9.3.8.3.4 nv show cluster apps running](#)
- [9.3.8.3.5 nv action start cluster apps](#)
- [9.3.8.3.6 nv action stop cluster apps](#)
- [9.3.8.3.7 nv show cluster apps log-level](#)
- [9.3.8.3.8 nv action update cluster apps log-level](#)
- [9.3.8.3.9 nv action restore cluster apps log-level](#)

### 9.3.8.3.1 nv show cluster apps

	nv show cluster apps Display the aggregated information of cluster apps
Syntax Description	N/A
Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show cluster apps Name          ID          Version      Capabilities          Components Version Status Reason  Additional Information Summary ----- nmx-controller nmx-c-nvos  0.8.0_2024-11-27_11-25 sm, gfm, fib, gw-api  sm:2004.11.6, gfm:R570.52, fib-fe:0.8.1 ok CONTROL_PLANE_STATE_CONFIGURED nmx-telemetry nmx-telemetry 0.8.3      telemetry, gnmi aggregation telemetry- collector:1.19.9, gnmi-aggregator:0.8.3 ok </pre>
REST API	GET https://<ip>/nvue_v1/cluster/apps
Related Commands	nv show cluster apps <app-name>
Notes	

### 9.3.8.3.2 nv show cluster apps name

	nv show cluster apps <app-name> Display an specific cluster app.	
Syntax Description	app-name	The name of the cluster app.
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show cluster app nmx-controller operational ----- app-id          nmx-c-nvos app-ver         0.9.0_2025-01-11_09-51 capabilities    sm, gfm, fib, gw-api components-ver sm:2025.01.1, gfm:R570.90, fib-fe:0.9.0 status         not ok reason         GFM: UNINITIALIZED addition-info  CONTROL_PLANE_STATE_UNCONFIGURED manager   ca-certificate   certificate   encryption    disabled   state         disabled           </pre>	
REST API	GET https://<ip>/nvue_v1/cluster/apps/{app-name}	
Related Commands	nv show cluster apps <app-name>	
Notes		

### 9.3.8.3.3 nv show cluster apps installed

	nv show cluster apps installed Display the installed apps of cluster.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre> Name          ID          Version          Capabilities Components Version ----- nmx-controller nmx-c-nvos  0.9.0_2025-01-11_09-51 sm, gfm, fib, gw-api sm:2025.01.1, gfm:R570.90, fib-fe:0.9.0 nmx-telemetry  nmx-telemetry 0.9.1          nvl telemetry, gnmi aggregation, syslog aggregation nvl-telemetry:1.19.14, gnmi-aggregator:0.8.9, nmx-connector:0.8.9           </pre>	
REST API	GET https://<ip>/nvue_v1/cluster/apps/installed	
Related Commands	nv show cluster apps running	
Notes		

### 9.3.8.3.4 nv show cluster apps running

	nv show cluster apps running Display the running apps of cluster.	
Syntax Description	N/A	

Default	N/A
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show cluster apps running Name           Status  Reason                               Additional Information ----- nmx-controller not ok  GFM: UNINITIALIZED                 CONTROL_PLANE_STATE_UNCONFIGURED nmx-telemetry  ok</pre>
REST API	GET https://<ip>/nvue_v1/cluster/apps/running
Related Commands	nv show cluster apps installed
Notes	

### 9.3.8.3.5 nv action start cluster apps

	nv action start cluster apps <app-name> Start an cluster app.	
Syntax Description	app-name	The name of the cluster app.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action start cluster apps nmx-controller</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/{app-name}	
Related Commands	nv action stop cluster apps <app-name>	
Notes		

### 9.3.8.3.6 nv action stop cluster apps

	nv action stop cluster apps <app-name> Stop an cluster app.	
Syntax Description	app-name	The name of the cluster app.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action stop cluster apps nmx-controller</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/{app-name}	
Related Commands	nv action start cluster apps <app-name>	
Notes		

### 9.3.8.3.7 nv show cluster apps log-level

	nv show cluster apps <app-name> log-level Display an cluster app's log-level.	
Syntax Description	app-name	The name of the cluster app.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show cluster apps nmx-controller log-level operational ----- log-level      warn</pre>	
REST API	GET https://<ip>/nvue_v1/cluster/apps/{app-name}/log-level	
Related Commands	nv action update cluster apps <app-name> log-level <log-level> nv action restore cluster apps <app-name> log-level	
Notes		

### 9.3.8.3.8 nv action update cluster apps log-level

	nv action update cluster apps <app-name> log-level <log-level> Change an cluster app's log-level.	
Syntax Description	app-name	The name of the cluster app.
	log-level	The new log-level, should be one of critical, debug, error, info, notice, warn
Default	log-level: info	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action update cluster apps nmx-controller log-level error</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/{app-name}/log-level	
Related Commands	nv action restore cluster apps <app-name> log-level	
Notes		

### 9.3.8.3.9 nv action restore cluster apps log-level

	nv action restore cluster apps <app-name> log-level Restore an cluster app's log-level to default.	
Syntax Description	app-name	The name of the cluster app.
Default	app-name	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action restore cluster apps nmx-controller log-level</pre>	

REST API	POST https://<ip>/nvue_v1/cluster/apps/{app-name}/log-level
Related Commands	nv action update cluster apps <app-name> log-level <log-level>
Notes	

## 9.3.9 Cluster Manager

NMX Manager is a component of the NMX solution designed for collecting and processing data center telemetry, monitoring, and providing insights and predictions on system operability and health. The role of NMX Manager is to aggregate streamed telemetry from the NMX Telemetry subsystem, filter, sort, run local predictions, and stream the collected data into the NMX Oasis data lake for further analysis. Additionally, NMX Manager can control network behavior and configuration settings by sending control messages to the NMX Controller.

### 9.3.9.1 How to Use Cluster Manager (Non-Secure)

Presented below is the NVOS configuration for utilizing Cluster Manager in a non-secure environment.

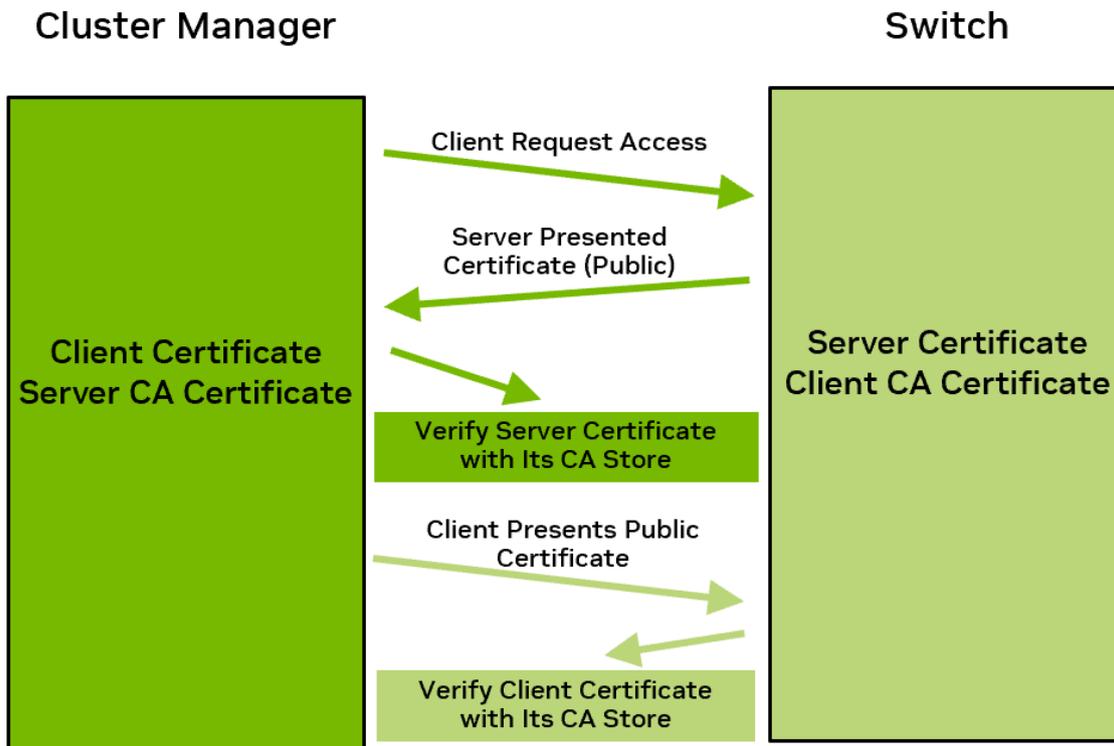
```
admin@nvos:~$ nv set cluster state enabled
admin@nvos:~$ nv config apply
admin@nvos:~$ nv action update cluster apps nmx-controller manager encryption disabled
admin@nvos:~$ nv action update cluster apps nmx-controller manager enabled
```

### 9.3.9.2 Cluster Manager Security

NMX Controller and NMX Telemetry offer security support.

They use GRPC for client communication, which works over TLS or MTLS configured via NVOS CLI. Below is a simple flow for using MTLS with Cluster Manager, along with a list of cluster commands attached to this manual.

## Mutual Authentication



The same CA could be used on both sides, or each side could choose a different CA.

### 9.3.9.2.1 Flow Description

In NVOS, the Cluster applications NMX-C/NMX-T function as GRPC servers, while the Cluster Manager device operates as the GRPC client.

The configuration from the NVOS CLI stores the Client CA certificate and the Server certificate on the NVOS side and binds the certificates to the apps for supporting the TLS/MTLS on top of GRPC.

### 9.3.9.2.2 Configuration for Enabling mTLS with Cluster Manager

To configure mutual TLS (mTLS) with Cluster Manager, ensure that the necessary certificates and configurations are set up across both the control plane and data plane components. Below is an example of how to configure mTLS in your Cluster Manager environment:

```
admin@nvos:~$ nv set cluster state enabled
admin@nvos:~$ nv config apply
admin@nvos:~$ nv action import system security certificate cert-name passphrase 12345678 uri-bundle scp://
your_username:your_password@1.2.3.4/path-to-cert/cert.pl2 //Saving the server certificate for TLS/MTLS
admin@nvos:~$ nv action import system security ca-certificate cacert-name uri scp://
your_username:your_password@1.2.3.4/path-to-cacert/ca.crt //Saving the Client CA certificate for mTLS
admin@nvos:~$ nv action update cluster apps nmx-controller manager enabled
admin@nvos:~$ nv action update cluster apps nmx-controller manager certificate cert-name //binding the imported
certificate to NMX
admin@nvos:~$ nv action update cluster apps nmx-controller manager ca-certificate cacert-name //binding the
imported CA certificate to NMX
admin@nvos:~$ nv action update cluster apps nmx-controller manager encryption mtls
```

### 9.3.9.2.3 Configuration for Enabling TLS with Cluster Manager

To enable TLS with Cluster Manager, you must configure the appropriate certificates and security settings for encrypted communication between services in the cluster. Below is an example configuration for setting up TLS in your Cluster Manager environment:

```
admin@nvos:~$ nv set cluster state enabled
admin@nvos:~$ nv config apply
admin@nvos:~$ nv action import system security certificate cert-name passphrase 12345678 uri-bundle scp://
your_username:your_password@1.2.3.4/path-to-cert/cert.p12
admin@nvos:~$ nv action update cluster apps nmx-controller manager enabled
admin@nvos:~$ nv action update cluster apps nmx-controller manager certificate cert-name
admin@nvos:~$ nv action update cluster apps nmx-controller manager encryption tls
```

### 9.3.9.3 Cluster Manager Commands



Action commands of Cluster Manager require a 5-second delay for execution between them. During this period, GPRC traffic will be paused. The command [nv action update cluster apps log-level](#) is an exception to this rule.

- [Cluster Manager Commands](#)

### 9.3.9.4 Cluster Manager Commands

- [9.3.9.4.1 nv show cluster apps manager](#)
- [9.3.9.4.2 nv show cluster apps manager encryption](#)
- [9.3.9.4.3 nv show cluster apps manager certificate](#)
- [9.3.9.4.4 nv show cluster apps manager ca-certificate](#)
- [9.3.9.4.5 nv action update cluster apps manager](#)
- [9.3.9.4.6 nv action update cluster apps manager certificate](#)
- [9.3.9.4.7 nv action update cluster apps manager ca-certificate](#)
- [9.3.9.4.8 nv action update cluster apps manager encryption](#)
- [9.3.9.4.9 nv action restore cluster apps manager encryption](#)
- [9.3.9.4.10 nv action restore cluster apps manager](#)
- [9.3.9.4.11 nv action restore cluster apps manager certificate](#)
- [9.3.9.4.12 nv action restore cluster apps manager ca-certificate](#)

#### 9.3.9.4.1 nv show cluster apps manager

	nv show cluster apps <app-name> manager Show a list of manager attributes.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show cluster apps nmx-telemetry manager ----- ca-certificate      ca certificate         cert_id_03 encryption          disabled state               disabled</pre>
REST API	GET https://<ip>/nvue_v1/cluster/apps/<app-name>/manager
Related Commands	nv show cluster apps <app-name> manager encryption nv show cluster apps <app-name> manager certificates
Notes	When cluster is disabled, no apps will be accepted as valid app names.

### 9.3.9.4.2 nv show cluster apps manager encryption

	nv show cluster apps <app-name> manager encryption Shows encryption mode which is disabled, mTLS, or TLS.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	Disabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show cluster apps nmx-telemetry manager encryption ----- encryption         disabled</pre>	
REST API	GET https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/encryption	
Related Commands	nv show cluster apps <app-name> manager certificates	
Notes	When the cluster is disabled, no apps will be accepted as valid app name	

### 9.3.9.4.3 nv show cluster apps manager certificate

	nv show cluster apps <app-name> manager certificate Shows the certificate ID which the cluster apps <app-name> manager is bounded to.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show cluster apps nmx-telemetry manager certificate ----- certificate         operational                    cert_id_03</pre>	
REST API	GET https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/certificate	
Related Commands	nv show cluster apps <app-name> manager ca-certificate	
Notes	When the cluster is disabled, no apps will be accepted as valid app name	

### 9.3.9.4.4 nv show cluster apps manager ca-certificate

	nv show cluster apps <app-name> manager ca-certificate Shows the ca certificate ID which the cluster apps <app-name> manager is bounded to.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show cluster apps nmx-telemetry manager ca-certificate ----- Ca-certificate      operational                     -----                     ca-name</pre>	
REST API	GET https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/ca-certificate	
Related Commands	nv show cluster apps <app-name> manager ca-certificate	
Notes	When the cluster is disabled, no apps will be accepted as valid app name	

### 9.3.9.4.5 nv action update cluster apps manager

	nv action update cluster apps <app-name> manager This action enables the cluster apps <app-name> manager communication. It opens an external port in the device for communication between the cluster app and the cluster manager.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	Disabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action update cluster apps nmx-telemetry manager Action executing ... Cluster App Manager Port updated successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager	
Related Commands	nv action update cluster apps <app-name> manager encryption	
Notes	Dependency that cluster state is enabled.	

### 9.3.9.4.6 nv action update cluster apps manager certificate

	nv action update cluster apps <app-name> manager certificate <cert-id> This command binds the certificate in the system to be used by the cluster manager.	
Syntax Description	cert-id	Certificate ID
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv action update cluster apps &lt;app-name&gt; manager certificate cert_id_03 Action executing ... Cluster App Manager Cert updated successfully Action succeeded</pre>
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/certificate
Related Commands	nv action update cluster apps <app-name> manager encryption
Notes	<p>The operation should be rejected if one of the following occurs:</p> <ul style="list-style-type: none"> <li>• cluster is disabled</li> <li>• The &lt;cert-id&gt; does not include: cert and private key</li> <li>• The certificate extended key usage does not have TLS client authentication</li> <li>• Given app is not valid</li> </ul>

#### 9.3.9.4.7 nv action update cluster apps manager ca-certificate

	nv action update cluster apps <app-name> manager ca-certificate <cacert-id> This command binds the ca certificate that exists in the system to be used by the cluster apps <app-name> manager.	
Syntax Description	cacert-id	CA certificate ID
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action update cluster apps &lt;app-name&gt; manager ca-certificate ca_id Action executing ... Cluster App Manager CA Cert updated successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/certificate @update {cacert-id: <cacert-id>}	
Related Commands	nv action update cluster apps <app-name> manager encryption	
Notes	<p>the &lt;cacert-id&gt; certificate was imported by nv action import system security ca-certificate “ca data”</p> <p>The operation should be rejected, if one of the following occurs:</p> <ul style="list-style-type: none"> <li>• cluster is disabled</li> <li>• The &lt;ca-cert-id&gt; does not include ca cert</li> <li>• Given app is not valid</li> </ul>	

#### 9.3.9.4.8 nv action update cluster apps manager encryption

	nv action update cluster apps <app-name> manager encryption <disabled   tls   mtls> This command sets which encryption mode will be used when using cluster apps <app-name> manager.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
	disabled	The cluster apps <app-name> manager feature will work with no encryption protocol
	tls	The cluster apps <app-name> manager feature will work with TLS protocol
	mtls	The cluster apps <app-name> manager feature will work with Mutual TLS (mTLS) protocol

Default	Disabled
History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action update cluster apps nmx-telemetry manager encryption tls Action executing ... Cluster App Manager Encryption updated successfully Action succeeded</pre>
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/encryption
Related Commands	nv action update cluster apps <app-name> manager certificate
Notes	<p>Requirement when mode is mTLS:</p> <ul style="list-style-type: none"> <li>• The cluster state is enabled</li> <li>• The certificate was bound to use TLS mode</li> <li>• The CA certificate was bound to use mTLS mode</li> <li>• Given app is valid</li> </ul> <p>Requirement when mode is TLS:</p> <ul style="list-style-type: none"> <li>• The cluster state is enabled</li> <li>• The certificate was bound to use TLS mode</li> <li>• Given app is valid</li> </ul> <p>Requirement when mode is disabled:</p> <ul style="list-style-type: none"> <li>• The cluster state is enabled</li> <li>• Given app is valid</li> </ul>

#### 9.3.9.4.9 nv action restore cluster apps manager encryption

	nv action restore cluster apps <app-name> manager encryption This command is setting encryption mode disabled.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	Disabled	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action restore cluster apps nmx-controller manager encryption Action executing ... Cluster App Manager Encryption restored successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/encryption	
Related Commands	nv action update cluster apps <app-name> manager certificate	
Notes	Command should be rejected when the following occurs: <ul style="list-style-type: none"> <li>• cluster is disabled</li> <li>• Given app is not valid</li> </ul>	

#### 9.3.9.4.10 nv action restore cluster apps manager

	nv action restore cluster apps <app-name> manager This command disables the communication between the NMX app and NMX manager. This action sends a notification to the NMX app to close the port which listens to the NMX manager's requests.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	Disabled	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action update cluster apps nmx-controller manager Action executing ... Cluster App Manager Port restored successfully Action succeeded</pre>
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager
Related Commands	nv action restore cluster apps <app-name> manager encryption
Notes	The operation should be rejected when the following occurs: <ul style="list-style-type: none"> <li>• cluster is disabled</li> <li>• given app is invalid</li> </ul>

### 9.3.9.4.11 nv action restore cluster apps manager certificate

	nv action restore cluster apps <app-name> manager certificate This command removes the bounded certificate used by the cluster apps <app-name> manager.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action restore cluster apps nmx-telemetry manager certificate Action executing ... Cluster App Manager Cert restored successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/certificate	
Related Commands	nv action restore cluster apps <app-name> manager encryption	
Notes	Command should be rejected when: <ul style="list-style-type: none"> <li>• cluster is disabled</li> <li>• cluster apps &lt;app-name&gt; manager encryption mode is "mtls/tls"</li> <li>• given app is invalid</li> </ul>	

### 9.3.9.4.12 nv action restore cluster apps manager ca-certificate

	nv action restore cluster apps <app-name> manager ca-certificate This command removes the CA certificate used by the cluster apps <app-name> manager.	
Syntax Description	app-name	Application name (e.g., nmx-telemetry, nmx-controller)
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action restore cluster apps nmx-telemetry manager ca-certificate Action executing ... Cluster App Manager CA Cert restored successfully Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/cluster/apps/<app-name>/manager/certificate	

Related Commands	nv action restore cluster apps <app-name> manager encryption
Notes	<p>Command should be rejected when the following occurs:</p> <ul style="list-style-type: none"> <li>• cluster is disabled</li> <li>• cluster apps &lt;app-name&gt; manager encryption mode is "mtls"</li> <li>• given app is invalid</li> </ul>

## 9.3.10 SDN

NVOS provides a comprehensive set of commands for managing Software-Defined Networking (SDN) configuration and state files and partitions. These tools enable users to generate, fetch, install, upload, and manage SDN-related configurations and states effectively.

### 9.3.10.1 Key Functionalities

- [SDN Configuration and State Management](#)
- [SDN Partitions Management](#)
- [SDN Reset to Factory Defaults](#)

### 9.3.10.2 SDN Configuration and State Management

#### 9.3.10.2.1 SDN Configuration

- **Generate and Fetch:** Create new SDN configuration files or retrieve existing ones based on application type

```
admin@nvos:~$ nv action generate sdn config app nmx-controller type fm_config
admin@nvos:~$ nv action fetch sdn config app nmx-controller type fm_config scp://<user>:<pswd>@<server_ip>/
<path_to_fm_config_file>
```

- **Install:** Apply SDN configuration files

```
admin@nvos:~$ nv action install sdn config app nmx-controller type fm_config files <fm_config_file_name>
```

- **Upload and Delete:** Upload/remove SDN configuration files to the system

```
admin@nvos:~$ nv action upload sdn config app nmx-controller type fm_config files nmx-
controller_fm_config_20241210_023001 scp://<user>:<pswd>@<server_ip>/<path_to_target_directory>
admin@nvos:~$ nv action delete sdn config app nmx-controller type fm_config files <fm_Config_to_delete>
```

- **View Files:** Display a list of available SDN configuration files

```
admin@nvos:~$ nv show sdn config app nmx-controller type fm_config files
Available config file      File path
-----
nmx-controller_fm_config_20241211_190246  /host/cluster_infra/app_config/nmx-controller/fm_config/
<file_name>
```

#### 9.3.10.2.2 SDN State

- **Generate State Files:** Create SDN state files for analysis or diagnostic purposes

```
admin@nvos:~$ nv action generate sdn state app nmx-controller type topology
admin@nvos:~$ nv action fetch sdn state app nmx-controller type topology scp://<user>:<pswd>@<server_ip>/<path_to_topology_file>
```

- **Upload and Delete State Files:** Upload state files to external systems or delete them from the system

```
admin@nvos:~$ nv action upload sdn state app nmx-controller type topology files nmx-controller_topology_20241210_023001 scp://<user>:<pswd>@<server_ip>/<path_to_target_directory>
admin@nvos:~$ nv action delete sdn state app nmx-controller type topology files <topology_to_delete>
```

- **View Files:** List existing SDN state files for easy reference and review

```
admin@nvos:~$ nv show sdn sdn app nmx-controller type topology files
Available config file          File path
-----
nmx-controller_topology_20241211_190246 /host/cluster_infra/app_config/nmx-controller/topology/<file_name>
```

## 9.3.10.3 SDN Partitions Management

### 9.3.10.3.1 Creating SDN Partitions

Partitions can be created to define the allocation of GPUs with attributes such as the following:

```
admin@nvos:~$ nv action create sdn partition <partition-id> name <name> resiliency-mode <resiliency-mode> mcast-limit <mcast-limit> [uuid <uuid>] [location <location-id>]
```

### 9.3.10.3.2 Updating SDN Partitions

#### 9.3.10.3.2.1 Add GPUs

Add GPUs to a partition using either a UUID or location.

Location-based:

```
admin@nvos:~$ nv action update sdn partition <partition-id> location <location-id> [no-reroute]
```

UUID-based:

```
admin@nvos:~$ nv action update sdn partition <partition-id> uuid <uuid> [no-reroute]
```

#### 9.3.10.3.2.2 Restore GPUs

Remove a GPU from a partition.

Location-based:

```
admin@nvos:~$ nv action restore sdn partition <partition-id> location <location-id> [no-reroute]
```

UUID-based:

```
admin@nvos:~$ nv action restore sdn partition <partition-id> uuid <uuid> [no-reroute]
```

### 9.3.10.3.2.3 Update Routing Table

Recalculate the routing table for a partition.

```
admin@nvos:~$ nv action update sdn partition <partition-id> reroute
```

### 9.3.10.3.2.4 Viewing SDN Partitions

List All Partitions

Display all partitions with details such as resiliency mode, multicast limit, and type.

```
admin@nvos:~$ nv show sdn partition
```

View Partition Details

Display detailed information about a specific partition, including associated GPUs and health status.

```
admin@nvos:~$ nv show sdn partition <partition-id>
```

### 9.3.10.3.2.5 Deleting SDN Partitions

Remove an existing partition.

```
admin@nvos:~$ nv action delete sdn partition <partition-id>
```

## 9.3.10.4 SDN Reset to Factory Defaults

Restore all SDN configurations, states, and partition settings to factory default values.

```
admin@nvos:~$ nv action reset sdn factory-default
```

## 9.3.10.5 SDN Commands

- [SDN Commands](#)

## 9.3.10.6 SDN Commands

- [9.3.10.6.1 SDN Configuration and State Files Management](#)
  - [9.3.10.6.1.1 nv action fetch sdn config apps type](#)
  - [9.3.10.6.1.2 nv action generate sdn config apps type](#)
  - [9.3.10.6.1.3 nv action install sdn config apps type files](#)
  - [9.3.10.6.1.4 nv action upload sdn config apps type files](#)

- [9.3.10.6.1.5 nv action delete sdn config apps type files](#)
- [9.3.10.6.1.6 nv show sdn config apps type files](#)
- [9.3.10.6.1.7 nv action generate sdn state apps type](#)
- [9.3.10.6.1.8 nv action upload sdn state apps type files](#)
- [9.3.10.6.1.9 nv action delete sdn state apps type files](#)
- [9.3.10.6.1.10 nv show sdn state apps type files](#)
- [9.3.10.6.2 SDN Partition Management](#)
  - [9.3.10.6.2.1 nv action create sdn partition name resiliency-mode mcast-limit](#)
  - [9.3.10.6.2.2 nv action delete sdn partition](#)
  - [9.3.10.6.2.3 nv action update sdn partition location](#)
  - [9.3.10.6.2.4 nv action restore sdn partition location](#)
  - [9.3.10.6.2.5 nv action update sdn partition uuid](#)
  - [9.3.10.6.2.6 nv action restore sdn partition uuid](#)
  - [9.3.10.6.2.7 nv action update sdn partition reroute](#)
  - [9.3.10.6.2.8 nv show sdn partition](#)
  - [9.3.10.6.2.9 nv show sdn partition id](#)
  - [9.3.10.6.2.10 nv show sdn partition id location](#)
  - [9.3.10.6.2.11 nv show sdn partition id uuid](#)
- [9.3.10.6.3 SDN Reset](#)
  - [9.3.10.6.3.1 nv action reset sdn factory-default](#)

## 9.3.10.6.1 SDN Configuration and State Files Management

### 9.3.10.6.1.1 nv action fetch sdn config apps type

	nv action fetch sdn config apps <app-name> type <file-type> <remote-url> Fetch an cluster app configuration file from a remote server.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The config file type.
	remote-url	<ul style="list-style-type: none"> <li>• The remote URL of the config file.</li> <li>• Format: [protocol]://username[:password]@hostname/path/filename</li> <li>• Supported protocols: SCP, HTTPS, FILE, FTP, and SFTP.</li> <li>• The password must be encoded if it contains special characters and is provided as part of the command.</li> </ul>
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action fetch sdn config apps nmx-controller type chassis_mapping scp:// user:pass@file-server/path/to/config/chassis_mapping_1 admin@nvos:~\$ nv action fetch sdn config apps nmx-controller type chassis_mapping file:///tmp/chassis_mapping_2</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/config/apps/{app-name}/type/{file-type}	
Related Commands	nv action install sdn config apps <app-name> type <file-type> files <file-name> nv show sdn config apps <app-name> type <file-type> files	
Notes		

### 9.3.10.6.1.2 nv action generate sdn config apps type

	nv action generate sdn config apps <app-name> type <file-type> Generate an cluster app config file.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The config file type.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action generate sdn config apps nmx-controller type chassis_mapping Action executing ... App config file nmx-controller_chassis_mapping_20241023_181411 is successfully generated Action succeeded</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/config/apps/{app-name}/type/{file-type}	
Related Commands	nv action upload sdn config apps <app-name> type <file-type> files <file-name> <remote-url> nv show sdn config apps <app-name> type <file-type> files	
Notes		

### 9.3.10.6.1.3 nv action install sdn config apps type files

	nv action install sdn config apps <app-name> type <file-type> files <file-name> Install an cluster app config file.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The config file type.
	file-name	The config file name.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action install sdn config apps nmx-controller type chassis_mapping files chassis_map_1</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/config/apps/{app-name}/type/{file-type}/files/{file-name}	
Related Commands	nv action fetch sdn config apps <app-name> type <file-type> <remote-url> nv show sdn config apps <app-name> type <file-type> files	
Notes		

### 9.3.10.6.1.4 nv action upload sdn config apps type files

	nv action upload sdn config apps <app-name> type <file-type> files <file-name> <remote-url> Upload an cluster app config file to a remote server.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The config file type.

	file-name	The config file name.
	remote-url	The remote server and path.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action upload sdn config apps nm-x-controller type chassis_mapping files chassis_map_1 scp://user:pass@file-server/path/to/config/</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/config/apps/{app-name}/type/{file-type}/files/{file-name}	
Related Commands	nv action generate sdn config apps <app-name> type <file-type> nv show sdn config apps <app-name> type <file-type> files	
Notes		

### 9.3.10.6.1.5 nv action delete sdn config apps type files

	nv action delete sdn config apps <app-name> type <file-type> files <file-name> Delete an cluster app local config file.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The config file type.
	file-name	The config file name.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action delete sdn config apps nm-x-controller type chassis_mapping files chassis_map_1</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/config/apps/{app-name}/type/{file-type}/files/{file-name}	
Related Commands	nv action generate sdn config apps <app-name> type <file-type> nv show sdn config apps <app-name> type <file-type> files	
Notes		

### 9.3.10.6.1.6 nv show sdn config apps type files

	nv show sdn config apps <app-name> type <file-type> files Display all the cluster app local config file for a file type.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The config file type.
Default	N/A	
History	25.02.1884	

Example	<pre>admin@nvos:~\$ nv show sdn config apps nmx-controller type chassis_mapping files File name      File path ----- nmx-controller_chassis_mapping_20240524_091045 /host/cluster/config_file/nmx- controller/chassis_mapping/nmx-controller_chassis_mapping_20240524_091045</pre>
REST API	GET https://<ip>/nvue_v1/sdn/config/apps/{app-name}/type/{file-type}/files
Related Commands	nv action generate sdn config apps <app-name> type <file-type> nv action delete sdn config apps <app-name> type <file-type> files <file-type>
Notes	

### 9.3.10.6.1.7 nv action generate sdn state apps type

	nv action generate sdn state apps <app-name> type <file-type> Generate an cluster app state file.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The state file type.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action generate sdn state apps nmx-controller type topology</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/state/apps/{app-name}/type/{file-type}	
Related Commands	nv action upload sdn state apps <app-name> type <file-type> files <file-name> <remote-url> nv show sdn state apps <app-name> type <file-type> files	
Notes		

### 9.3.10.6.1.8 nv action upload sdn state apps type files

	nv action upload sdn state apps <app-name> type <file-type> files <file-name> <remote-url> Upload an cluster app state file to remote server.	
Syntax Description	app-name	The name of the cluster app.
	file-type	The state file type
	file-name	The state file name
	remote-url	The remote server and path
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action upload sdn state apps nmx-controller type topology files topology_file_1 scp://user:pass@file-server/path/to/config/</pre>	

REST API	POST https://<ip>/nvue_v1/sdn/state/apps/{app-name}/type/{file-type}/files/{file-name}
Related Commands	nv action generate sdn state apps <app-name> type <file-type> nv show sdn state apps <app-name> type <file-type> files
Notes	

### 9.3.10.6.1.9 nv action delete sdn state apps type files

	nv action delete sdn state apps <app-name> type <file-type> files <file-name> Delete an cluster app local state file.	
Syntax Description	app-name	The name of the cluster app
	file-type	The state file type
	file-name	The state file name
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action delete sdn state apps nmx-controller type topology files topology_file_1</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/state/apps/{app-name}/type/{file-type}/files/{file-name}	
Related Commands	nv action generate sdn state apps <app-name> type <file-type> nv show sdn state apps <app-name> type <file-type> files	
Notes		

### 9.3.10.6.1.10 nv show sdn state apps type files

	nv show sdn state apps <app-name> type <file-type> files Display all the cluster app local state file for a file type.	
Syntax Description	app-name	The name of the cluster app
	file-type	The state file type
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show sdn state apps nmx-controller type topology files File name                               File path ----- - nmx-controller_topology_20240526_182245 /host/cluster/state_file/nmx-controller/ topology/nmx-controller_topology_20240526_182245</pre>	
REST API	GET https://<ip>/nvue_v1/sdn/state/apps/{app-name}/type/{file-type}/files	
Related Commands	nv action generate sdn state apps <app-name> type <file-type> nv action delete sdn state apps <app-name> type <file-type> files <file-type>	
Notes		

## 9.3.10.6.2 SDN Partition Management

### 9.3.10.6.2.1 nv action create sdn partition name resiliency-mode mcast-limit

	nv action create sdn partition <partition_id> name <name> resiliency-mode <resiliency-mode> mcast-limit <mcast-limit> [uuid <uuid>] [location <location-id>] Create a partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32765).
	name	The partition name, must be unique in the domain
	resiliency-mode	The resiliency-mode is one of full_bandwidth, adaptive_bandwidth and user_action: <ul style="list-style-type: none"> <li>• <b>Full Bandwidth:</b> On a trunk link failure, partition will attempt an automatic recovery. If spare trunk links are not available, GPUs will be excluded from the fabric to maintain full bandwidth for the rest of the GPUs.</li> <li>• <b>Adaptive Bandwidth:</b> On a trunk link failure, partition will attempt an automatic recovery. If spare trunk links are not available, the partition's GPUs will operate with a lower bandwidth than optimal.</li> <li>• <b>User Action Required:</b> On a trunk link failure, partition will attempt an automatic recovery. If spare trunk links are not available, the partition will go into an unhealthy state which requires user action for recovery. Example actions would be providing additional trunk links or removing GPUs from the partition.</li> </ul>
	mcast-limit	An integer presents the limit of multicast groups (0-1024)
	uuid	The GPU UID
	location-id	The location identifier of a GPU. The format of a location id is <chassis-id>.<slot-id>.<host-id>.<gpu-id>.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action create sdn partition 1 name part1 resiliency-mode adaptive_bandwidth mcast-limit 0 location-id 1.1.1.1</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/partition/{partition-id}	
Related Commands	nv action delete sdn partition <partition-id> nv show sdn partition <partition-id>	
Notes	<ul style="list-style-type: none"> <li>• A location or UUID can be provided when creating a partition, but not both</li> <li>• If both location and UUID has not be provide, an empty partition without any GPU will be created</li> <li>• 32766 is reserved for the ID of the default partition</li> </ul>	

### 9.3.10.6.2.2 nv action delete sdn partition

	nv action delete sdn partition <partition-id> Delete a partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766)
Default	N/A	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv action delete sdn partition 1</pre>
REST API	POST https://<ip>/nvue_v1/sdn/partition/{partition-id}
Related Commands	nv show sdn partition <partition-id>
Notes	

### 9.3.10.6.2.3 nv action update sdn partition location

	nv action update sdn partition <partition-id> location <location-id> [no-reroute] Add a GPU with location ID to a partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766)
	location-id	The location identifier of a GPU. The format of a location id is <chassis-id>.<slot-id>.<host-id>.<gpu-id>.
	no-route	Do not update routing table after GPU added. By default, the routing table is updated.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action update sdn partition 1 location 1.1.1.2</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/partition/{partition-id}/location/{location-id}	
Related Commands	nv action restore sdn partition <partition-id> location <location-id> nv show sdn config app <app-name> type <file-type> files	
Notes	<ul style="list-style-type: none"> <li>This command is only for the location-based partition, use "nv show sdn partition" to check if a partition is location-based or UUID-based</li> <li>The no-reroute option is only for scaleout setup</li> </ul>	

### 9.3.10.6.2.4 nv action restore sdn partition location

	nv action restore sdn partition <partition-id> location <location-id> [no-reroute] Remove a GPU with location ID from a partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766)
	location-id	The location identifier of a GPU. The format of a location id is <chassis-id>.<slot-id>.<host-id>.<gpu-id>.
	no-reroute	Do not update routing table after GPU removed. By default, the routing table is updated.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action restore sdn partition 1 location 1.1.1.2 no-reroute</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/partition/{partition-id}/location/{location-id}	

Related Commands	nv action update sdn partition <partition-id> location <location-id> nv show sdn partition <partition-id>
Notes	<ul style="list-style-type: none"> <li>This command is only for the location-based partition, use "nv show sdn partition" to check if a partition is location-based or UUID-based.</li> <li>The no-reroute option is only for scaleout setup.</li> </ul>

### 9.3.10.6.2.5 nv action update sdn partition uuid

	nv action update sdn partition <partition-id> uuid <uuid> [no-reroute] Add a GPU with UUID to a partition	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766).
	location-id	The location identifier of a GPU. The format of a location id is <chassis-id>.<slot-id>.<host-id>.<gpu-id>.
	no-route	Do not update routing table after GPU added. By default, the routing table is updated.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action update sdn partition 1 uuid 634426924859161775</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/partition/{partition-id}/uuid/{uuid}	
Related Commands	nv action restore sdn partition <partition-id> uuid <uuid> nv show sdn partition <partition-id>	
Notes	<ul style="list-style-type: none"> <li>This command is only for the uuid-based partition, use "nv show sdn partition" to check if a partition is location-based or UUID-based.</li> <li>The no-reroute option is only for scaleout setup.</li> </ul>	

### 9.3.10.6.2.6 nv action restore sdn partition uuid

	nv action restore sdn partition <partition-id> uuid <uuid> [no-reroute] Remove a GPU from from a partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766).
	location-id	The location identifier of a GPU. The format of a location id is <chassis-id>.<slot-id>.<host-id>.<gpu-id>.
	no-reroute	Do not update routing table after GPU removed. By default, the routing table is updated with the command.
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action restore sdn partition 1 uuid 634426924859161775 no-reroute</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/partition/{partition-id}/location/{location-id}	
Related Commands	nv action update sdn partition <partition-id> location <location-id> nv show sdn partition <partition-id>	

Notes	<ul style="list-style-type: none"> <li>This command is only for the uuid-based partition, use "nv show sdn partition" to check if a partition is location-based or UUID-based</li> <li>The no-reroute option is only for scaleout setup.</li> </ul>
-------	---

### 9.3.10.6.2.7 nv action update sdn partition reroute

	nv action update sdn partition <partition-id> reroute Update the routing table of a partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766).
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action update sdn partition 1 reroute</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/partition/{partition-id}	
Related Commands	nv action restore sdn partition <partition-id> uuid <uuid> nv show sdn partition <partition-id>	
Notes	The command can not be used on non-scaleout setup.	

### 9.3.10.6.2.8 nv show sdn partition

	nv show sdn partition Display all the sdn partitions.	
Syntax Description	N/A	
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv show sdn partition ID      Name      Num of GPUs  Health  Resiliency mode  Multicast groups limit  Partition type  Summary ----- 1      partition1  0            degraded  full_bandwidth    0 2      partition1  4            healthy   full_bandwidth    0 location_based 32766  Default Partition  4            unhealthy  adaptive_bandwidth 1024 gpuuid_based</pre>	
REST API	GET https://<ip>/nvue_v1/sdn/partition	
Related Commands	nv action create sdn partition <partition-id> name <name> resilience-mode <resilience-mode> mcast-limit <mcast-limit> location <location-id> nv action delete sdn partition <partition-id>	
Notes		

### 9.3.10.6.2.9 nv show sdn partition id

	nv show sdn partition <partition-id> Display all the sdn partitions.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766)

Default	N/A
History	25.02.1884
Example	<pre> admin@nvos:~\$ nv show sdn partition 32766 operational ----- name           Default Partition num-gpus       4 health         healthy resiliency-mode adaptive_bandwidth mcast-limit    1024 partition-type gpuuid_based  locations ===== GPU Location  UUID ----- 1.1.1.1      10000000000000000001 1.1.1.2      20000000000000000002 1.1.1.3      30000000000000000003 1.1.1.4      40000000000000000004 </pre>
REST API	GET https://<ip>/nvue_v1/nvos/partition/{partition-id}
Related Commands	nv action create sdn partition <partition-id> name <name> resilience-mode <resilience-mode> mcast-limit <mcast-limit> location <location-id> nv action delete control-plane partition <partition-id>
Notes	

#### 9.3.10.6.2.10 nv show sdn partition id location

	nv show sdn partition <partition-id> location Display all the locations of the GPUs in the SDN partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766)
Default	N/A	
History	25.02.1884	
Example	<pre> admin@nvos:~\$ nv show sdn partition 32766 location GPU Location  UUID ----- 1.1.1.1      8429764542702799123 1.1.1.2      14003344813302830705 1.1.1.3      634426924859161785 1.1.1.4      10713516454341026771 </pre>	
REST API	GET https://<ip>/nvue_v1/nvos/partition/{partition-id}/location	
Related Commands	nv action create sdn partition <partition-id> name <name> resilience-mode <resilience-mode> mcast-limit <mcast-limit> location <location-id> nv action delete control-plane partition <partition-id> nv show sdn partition id uuid	
Notes		

#### 9.3.10.6.2.11 nv show sdn partition id uuid

	nv show sdn partition <partition-id> uuid Display all the UUIDs of the GPUs in the SDN partition.	
Syntax Description	partition-id	The partition ID (an integer in range 1-32766)
Default	N/A	

History	25.02.1884
Example	<pre>admin@nvos:~\$ nv show sdn partition 32766 uuid GPU UUID          Location ----- 634426924859161785  1.1.1.3 8429764542702799123  1.1.1.1 10713516454341026771 1.1.1.4 14003344813302830705 1.1.1.2</pre>
REST API	GET https://<ip>/nvue_v1/nvos/partition/{partition-id}/uuid
Related Commands	<pre>nv action create sdn partition &lt;partition-id&gt; name &lt;name&gt; resilience-mode &lt;resilience-mode&gt; mcast-limit &lt;mcast-limit&gt; location &lt;location-id&gt; nv action delete control-plane partition &lt;partition-id&gt; nv show sdn partition id location</pre>
Notes	

### 9.3.10.6.3 SDN Reset

#### 9.3.10.6.3.1 nv action reset sdn factory-default

	<pre>nv action reset sdn factory-default [force] Restore SDN configuration to factory default.</pre>	
Syntax Description	force	Reset factory-default without prompting user
Default	N/A	
History	25.02.1884	
Example	<pre>admin@nvos:~\$ nv action reset sdn factory-<b>default</b> The operation will reset sdn configuration to factory-<b>default</b>. Type [y] to reset to factory-<b>default</b>. Type [N] to abort.</pre>	
REST API	POST https://<ip>/nvue_v1/sdn/factory-default	
Related Commands		
Notes		

---

# 10 Document Revision History

Version 25.02.2342—June 2025

Added:

- Note in [Certificates Management](#) section
- Note under [Recommended Full System Upgrade Sequence Example for Automation Reference](#) section
- Note under [Upgrading OS Software](#) section

Version 25.02.2141—March 2025

Updated:

- The subsection [Recovery Flow After SSD Wipe](#)
- [GRUB timeout style](#)

Added:

- The command [nv set acl rule action set dscp](#)
- The subsection [Set DSCP on Transit Traffic](#) in the [Access Control List Configuration](#) section
- The command [nv show platform cable-cartridge](#)
- The command [nv show platform cable-cartridge name](#)
- The subsection [ZTP Configuration provisioning-script](#)
- The subsection [Provisioning Script Example](#)
- The subsection [Recommended Full System Upgrade Sequence Example for Automation Reference](#)

Version 25.02.1884—February/March 2025

Updated:

- The command output for [nv show sdn partition](#)
- The commands in [SDN Configuration and State Files Management](#) section, changing "app" to "apps"

Added:

- The subsection [ZTP Configuration nmx-commands-list](#)
- The command [nv action reset platform bmc-password](#)
- The subsection [Resetting BMC Root User Password](#)
- The command [nv show sdn partition id location](#)
- The command [nv show sdn partition id uuid](#)
- The subsection [SSD Wipe](#)
- A note in the command [nv action system ztp](#)

Removed:

- [nv show platform environment psu](#)

Version 25.02.1786—December 2024

This is the first production-sample-level release of the NVLink NVOS software documentation.

## Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. Neither NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: "NVIDIA") make any representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

## Trademarks

NVIDIA, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of NVIDIA Corporation and/



or Mellanox Technologies Ltd. in the U.S. and in other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2024 NVIDIA Corporation & affiliates. All Rights Reserved.

