InfiniBand Bring-up Tool v3.0.0

Software version 3.0.0

Table of Contents

1 About This Document

Cluster Bring Up is a web-based framework for InfiniBand clusters which serves as an orchestration tool for tasks such as burning firmware images, burning cable transceivers' firmware, installing software packages, etc.

1.1 Technical Support

Customers who purchased NVIDIA products directly from NVIDIA are invited to contact us through the following methods:

- E-mail: enterprisesupport@nvidia.com
- Enterprise Support page: https://www.nvidia.com/en-us/support/enterprise

Customers who purchased NVIDIA M-1 Global Support Services, please see your contract for details regarding technical support.

Customers who purchased NVIDIA products through an NVIDIA-approved reseller should first seek assistance through their reseller.

1.2 Related Documentation

- Clonezilla software
- KVM
- VMware – What is a Hypervisor?
- HPE ProLiant DL380p Gen8
- Kubernetes
- AWX User Guide

2 Introduction

The cluster bring-up is a web-based framework for InfiniBand cluster orchestration which can be used for:

- Burning firmware images
- Burning cables transceivers' firmware
- Installing software packages
- Performing network verification

This framework exposes a static file server on a cluster bring-up host. A file server is useful to access files (e.g., packages, images, etc.) which are not available on the WEB.

3 Cluster Bring-Up Web Installation Steps

3.1 Deploying Cluster Bring-Up WEB Framework

This section describes how to deploy Cluster Bring-Up Web on a Linux machine.

3.1.1 Prerequisites

Python 3.6 or greater is required on the host where the framework is to be deployed.

3.1.2 System Requirements

The system that runs the cluster bring-up framework must satisfy the following requirements:

- At least 4GB of memory
- At least 2 CPU cores
- At least 30GB of space
- Running Kubernetes

3.1.3 Supported Operating Systems

- CentOS 8 or later 64-bit (x86)
- Red Hat Enterprise Linux 8.2 or later 64-bit (x86)
- Ubuntu 20.04 or later 64-bit (x86)

3.1.4 Supported Deployment Platforms

NVIDIA currently supports running cluster bring-up framework as a containerized application using Docker images deployed to a Kubernetes cluster.

In the following sections, you'll find deployment details and instructions for a Kubernetes platform.

3.2 Deploying via Kubernetes

This section describes how to deploy Cluster Bring-Up WEB in Kubernetes cluster.

3.2.1 Installation

The installation is performed by using a virtual machine (VM) image which includes COT with all its dependencies.

3.2.1.1 Installation with Image

This section shows how to install the and deploy the cluster-bring up in offline mode which requires the user to download and restore a machine image with most of the dependencies already located on the machine.

3.2.1.1.1 Prerequisites

The following is a list of requirements that must be met:

- Clonezilla version 3.0.1.8

3.2.1.1.2 Installation Steps

For offline installation, perform the following steps:

1. Download the tar image file located here.
2. Move the download file to the data center and untar the file.
3. Restore the image on your machine via Clonezilla. See section Restore Image for procedure.
4. Log into the installation machine as the `root` user with the password "`password`".
5. Make sure Kubernetes is running in `Ready` status:

```
$ kubectl get nodes
NAME              STATUS   ROLES                  AGE   VERSION
ib-node-01-cot    Ready    control-plane,master   39m   v1.24.2+k3s1
```

6. Change directory to the location of the installation script located under `/cot`:

```
$ pwd
/cot
$ ls -la
total 24
drwxr-xr-x  4 root root 4096 Jul 21 14:36 .
drwxr-xr-x 21 root root 4096 Jul 21 15:03 ..
drwxr-xr-x  3 root root 4096 Jul 21 14:35 ansible
drwxr-xr-x  5 root root 4096 Jul 21 14:36 installer
-rwxr-xr-x  1 root root  583 Jul 21 14:36 install.sh
-rwxr-xr-x  1 root root  393 Jul 21 14:36 uninstall.sh
```

7. Run the installation script with the `--offline-mode` flag:

```
$ ./install.sh --offline-mode
Installing cluster-bringup-service
Installing awx-operator
Installing awx-cluster-bringup
AWX is currently upgrading
Importing AWX resources
Installation finished successfully
AWX interface URL: http://10.43.144.44:80
AWX username: admin
AWX password: SxFLNsjpjAuoUICJDl0XUvdjDDmQmBWf
AWX OAuth token: jacb63Ac3bzyXXTTzsYbzAdA1mymaP
API URL: http://cluster-bringup:5000/api
PyPI URL: http://cluster-bringup:5000/pypi/simple/
Downloads URL: http://cluster-bringup:5000/downloads/
Files folder: /opt/nvidia/cot/files
```

3.2.1.2 Restore Image

As part of the installation process, an image with Kubernetes and AWX-Operator already present must be restored on a machine. To restore, the Clonezilla software must be utilized.

3.2.1.2.1 Restore VM Using Hypervisor

The Clonezeilla restoration procedure can also be used for virtualization.

The following subsections provide the list of virtualization solutions that are supported.

3.2.1.2.1.1 KVM

Kernel-based Virtual Machine, or KVM, is a full virtualization solution for Linux on x86 hardware containing virtualization extensions. Using KVM, users can run multiple VMs running unmodified Linux or Windows images. Each VM has private virtualized hardware: A network card, disk, graphics adapter, etc.

3.2.1.2.1.2 Dependencies

The following is a list of required dependencies:

- virt-manager application

3.2.1.2.1.3 Restoration Steps

Follow these steps to restore the image on a VM. Each step has a name prepended to the step which indicated from which machine to perform the action:

1. On the machine running a hypervisor, check if there is enough space in the root and `/images` directories.

```
$ df -h
Filesystem        Size  Used Avail Use% Mounted on
devtmpfs           91G     0   91G   0% /dev
tmpfs              91G     0   91G   0% /dev/shm
tmpfs              91G   35M   91G   1% /run
tmpfs              91G     0   91G   0% /sys/fs/cgroup
/dev/sda2          44G  8.1G   33G  20% /
/dev/sda1         2.0G  226M  1.6G  13% /boot
/dev/sda5         392G   99G  274G  27% /images
tmpfs              19G  4.0K   19G   1% /run/user/0
l1:/vol/s1        2.9T  2.0T  931G  69% /auto/s1
```

2. On the machine running the hypervisor, download Clonezilla ISO and move it to the `/tmp` directory.

```
$ ll /tmp
total 396484
-rw-r--r--   1 qemu qemu 379584512 Jul 17 16:06 clonezilla-live-3.0.1-8-amd64.iso
```

3. On the machine running the hypervisor, create a new directory in the `/images` directory with the name of the newly created machine.
4. On the machine running the hypervisor, create a disk image with 65G.

```
$ qemu-img create -f raw /images/<machine_name>/<machine_name>-bk-disk.img 65G
Formatting '/images/<machine_name>/<machine_name>-bk-disk.img', fmt=raw size=69793218560
```

5. On the machine running the hypervisor, open the Virtual Manager GUI.

```
$ virt-manager
```

6. In the Virtual Manager GUI, click the "Create a virtual machine" icon on the top left.
7. Create a new VM (5 steps):
    a. Select "Local install media".
    b. For "Choose ISO", select the Clonezilla ISO placed in `/tmp`, uncheck "Automatically detect from the installation media", type and select the OS of choice (must be supported).
    c. Memory: 4096; CPUs: 2

    d. For "Select or create custom storage" and browse to the image disk created earlier.
    e. Type in a unique machine name and check the "Customize configuration before install" box
    f. Click "Finish".
8. In the Virtual Manager GUI, change the boot order:
    a. Open the settings of the VM you are restoring on.
    b. Boot Options.
    c. Check the "Clonezilla CDROM" box which is linked to the Clonezilla ISO from step 2 above.
    d. Click the up arrow to move it up in the boot order.
    e. Click "Apply".
    f. Click "Begin Installation".
9. After restarting the machine, the Clonezilla software will boot. Follow these steps to successfully restore the image:
    a. Clonezilla live.
    b. English.
    c. Keep.
    d. Start.
    e. device-image.
    f. ssh_server.
    g. dhcp.
    h. Type the IP address of the machine which stores the untar file from step 2 of section "Installation Steps".
    i. Port stays at "22" (default ssh).
    j. Keep "root" as user.
    k. Type the directory path which stores the untar file from step 2 of section "Installation Steps".
    l. Type password to `root` user.
    m. Mode: Beginner.
    n. restoredisk.
    o. Select the name of your image.
    p. Select the name of your storage.
    q. Yes, check.
    r. Power off.
10. In the Virtual Manager GUI, select "Change Boot Order". Then move disk image created in step 4 to the top of the list ahead of Clonezilla (CDROM).
11. In the Virtual Manager GUI, select "Force off" and "Start VM".
12. After booting, log in as root user with the password "password".
13. (Restore) Change the name of the machine since it has the cloned machine name configured.

```
$ vi /etc/hostname
$ vi /etc/hosts
```

14. (Restore) If no Internet access is available on the machine, change the network interface in use.

```
$ ifconfig -a
$ ethtool ens3 # Link detected: no
$ dhclient
```

```
$ ethtool ens3
$ ifconfig -a
$ vim /etc/netplan/00-installer-config.yaml
```

15. (Restore) Reboot machine → reboot.

3.2.1.2.2 Restore on Bare Metal

This section explains how to restore the image on a physical computer server.

3.2.1.2.2.1 Supported Hardware

ProLiant DL380p Gen8

3.2.1.2.2.2 Restoration Steps

1. Connect to machine's remote management, ILO for HPE.
2. Mount/add Clonezilla ISO via: Virtual Drives → Image File CDROM → Select Clonezilla ISO
3. Reset the machine: Power Switch → Reset.
4. Boot via Clonezilla ISO: Press F11 on startup → select CDROM Clonezilla ISO for boot.
5. Continue from step 9 of section Restore on VM Machine to the end.

> ⚠ For additional information on HPE's remote management, visit HPE's support website.

3.2.2 Installation Script

The installation script, `install.sh`, performs the following operations:

- Creates a new virtual environment for installation
- Ensures the dependencies for the installer are installed
- Deploys cluster bring-up WEB framework on Kubernetes platform
- Deploys cluster bring-up AWX framework on Kubernetes platform
- Configures AWX resources for cluster orchestration

Usage:

> ⚠ Make sure to be located in the folder of the installation script (under `/cot`).

```
./install.sh [OPTIONS]
```

The following options are available for the installation script:

Option	Description
`--hostfile`	Specify path to hosts file that contains hostnames for the inventory
`--hostname`	Specify end-host list expression that represents hostnames for the inventory
`--ib-host-manager`	Specify hostname to be a member of the `ib_host_manager` group

Option	Description
`--username`	Specify username to authenticate against the hosts
`--password`	Specify password (encoded in base64) to authenticate against the hosts
`--offline-mode`	Specify to run the installation script in offline mode. Supported only when using COT image.
`--config_file`	Specify the path to the configuration file to incorporate into the installation

For example:

```
$ ./install.sh --hostname ib-node-0[1-2,5] --ib-host-manager ib-node-01

Installing cluster-bringup-web
Installing awx-operator
Installing awx-cluster-bringup
AWX is currently upgrading
Importing AWX resources
Installation finished successfully
AWX interface URL: http://cluster-bringup:31873
AWX username: admin
AWX password: NDaXP7ULFjoHdxNwEYxLPRYx6PNWxwoX
AWX OAuth token: ihj219yX6w5cpmgqvHy923nyQTjuoB
API URL: http://cluster-bringup:5000/api
PyPI URL: http://cluster-bringup:5000/pypi/simple/
Downloads URL: http://cluster-bringup:5000/downloads/
Files folder: /opt/nvidia/cot/files
```

In this example, 3 hosts named `ib-node-01`, `ib-node-02`, and `ib-node-05` are added to the inventory.

In addition, the `ib-node-01` host configured to be a member of the `ib_host_manager` group for the In-Band operations.

3.2.2.1 Configuration File

This section provides the required information to add a YAML configuration file during the installation process.

> ⚠ Currently, the configuration file only supports adding inventory variables so that they are included in the IB Cluster Inventory variable list when AWX loads for the first time.

The YAML file must consist of an `extra_variables` parent key paired with a dictionary value. That dictionary must include an `inventory_vars` key which also has its own dictionary value. It will consist of a list of key-value pairs that are added to the inventory variables.

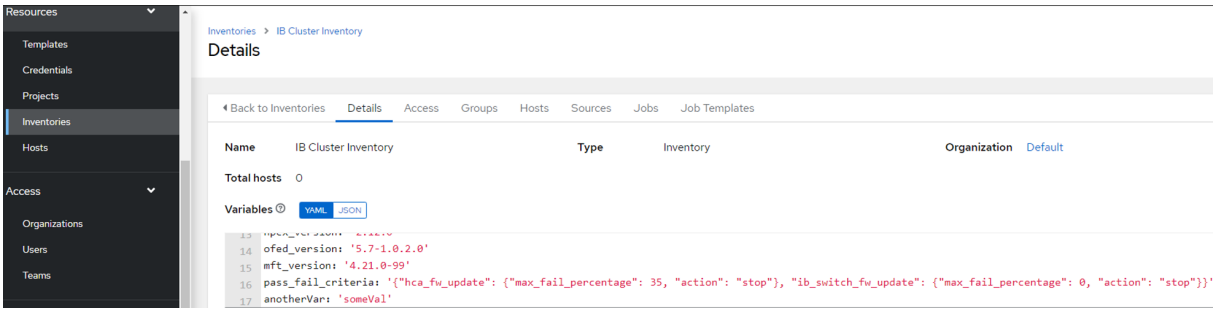
YAML configuration file example:

```
---
extra_variables:
  inventory_vars:
    - pass_fail_criteria:
        hca_fw_update:
          max_fail_percentage: 35
          action: stop
        ib_switch_fw_update:
          max_fail_percentage: 0
          action: stop
    - anotherVar: someVal
```

In this example, there are two variables, `pass_fail_criteria` and `anotherVar`, that will be added to the inventory variables list in AWX.

Example usage with the configuration file flag:

```
$ ./install.sh --config_file /PATH/TO/YAML
```



As seen in the picture above, `pass_fail_criteria` and `anotherVar` have been added to the IB Cluster Inventory after AWX loads for the first time.

3.2.3 Upgrading Framework Script

The `upgrade.sh` script upgrades the COT containers and configuration files, including the COT API itself, while preserving the existing data.

To upgrade the COT:

1. Download `tar.gz` upgrade file from the COT download center.
2. Extract the upgrade file.
3. Run the `upgrade.sh` script located in the extracted folder.

Example:

```
root@cot-server:/cot/upgrade_example# ./upgrade.sh
Upgrading COT API
Building COT snapshot
Snapshot built successfully. Path: /tmp/cot_snapshot_26-03-23_08-28.tar.gz
Removing cluster-bringup-service
Removing local registry
Removing awx-cluster-bringup
Removing awx-operator
Installing awx-operator
Installing local registry
Installing awx-cluster-bringup
AWX is currently upgrading
Installing cluster-bringup-service
Importing snapshot
Removing snapshot /tmp/cot_snapshot_26-03-23_08-28.tar.gz
Successfully upgraded using /cot/upgrade_example/upgrade_data
```

3.2.4 COT API

This section details the operations that could be performed once the installation process concludes.

The following code block demonstrates all the available actions:

```
$ cot [-h] [-v] {install,update,show,uninstall}
```

⚠ The install and uninstall operations must be utilized via the `install.sh` and `uninstall.sh` scripts.

3.2.4.1 Update

The update command allows updating certain components of the Cluster Bring-up Tool.

```
$ cot update [-h] --cot_dir <PATH> {playbooks,awx_templates,cot_client}
```

⚠ The update command relies on the `cot_dir` argument, which refers to the path of the folder extracted from the `tar.gz` file given .

Mandatory arguments:

Arguments	Description
`--cot_dir`	Specify the path of the folder extracted from the new `tar.gz` file. The tool uses the data inside the folder as the new data for the update operation.

Optional arguments:

Arguments	Description
`playbooks`	Update the ansible playbooks
`awx_templates`	Update the AWX templates (job templates and workflows). This updates the ansible playbooks as a pre-task.
`cot_client`	Update the COT client (on the `ib_host_manager` specified host)

3.2.4.2 Show

Usage:

```
$ cot show [-h] [--awx_info] [--file_server_info] [--api_url]
```

Options:

Option	Description
`--awx_info`	Get AWX URL and credentials
`--file_server_info`	Get file server URL and files folder
`--api_url`	Get the REST API URL

3.2.4.3 Export

The export operation allows creating a snapshot of the data within an existing COT environment. This may be used to transport the data between environments.

Usage:

```
cot export [-h] [--dest_path PATH] [--components {all,playbooks,file_server,database,awx}
[{all,playbooks,file_server,database,awx} ...]]
```

Options:

Option	Description
`--dest_path`	Directory path to save the snapshot. Default: `/tmp`.
`--components`	List of components to export, separated by spaces. Default: `all`.

Example:

```
root@cot-server:/ # cot export --dest_path /tmp/example/ --components playbooks database
```

This command builds a snapshot containing the playbooks and the database of the current COT environment. The `.tar.gz` snapshot file produced is saved to `/tmp/example/<snapshot_name>`.

Output:

```
Exporting playbooks
Exporting database
Wrapping
Finished Export. File located at: /tmp/example/cot_playbooks_database_22-03-23_12-22.tar.gz
```

3.2.4.4 Import

The import operation allows importing data of a given snapshot into an existing COT environment.

Usage:

```
cot import [-h] -s PATH [-f] [--merge_file_server_files] [--components {all,playbooks,file_server,database,awx}
[{all,playbooks,file_server,database,awx} ...]]
```

Options:

Option	Description
`--merge_file_server_files`	Adds the file server files from the snapshot to the existing files in the file server of the COT environment. ⚠ Without this flag, the files in the file server are overridden.
`-s`	Path to snapshot file.

Option	Description
`--components`	List of components to import, separated by spaces. ⚠ If not provided, the command imports the data of all the components contained in the snapshot.

Example:

```
root@cot-server:/# cot import -s /tmp/cot_snapshot_22-03-23_12-24.tar.gz --merge_file_server_files --components
file_server database
```

This command imports the file server files and the database content from the snapshot into the COT environment. The file server files from the snapshot are added to the files that already exist in the file server.

Output:

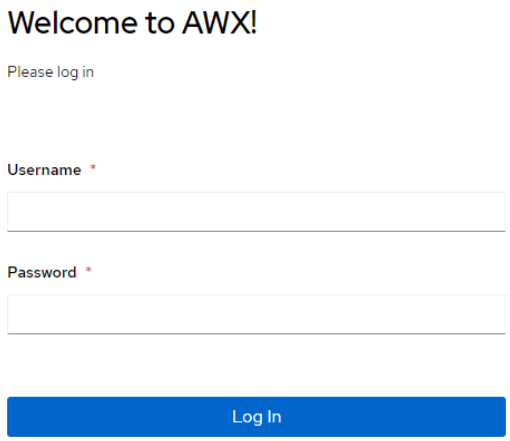
```
Importing File Server
Importing database
Import finished successfully from snapshot: /tmp/cot_snapshot_22-03-23_12-24.tar.gz
```

4 AWX Cluster Bring-Up

4.1 AWX Interface

This section describes how to use the AWX web-based interface.

To log into AWX, browse to AWX interface at `http://<AWX-server-name>/` and log in with a valid AWX username and password.



The default username set during installation has `admin` capabilities. The credentials can be accessed in the `awx-cluster-bringup-admin-password` secret. TBD verify

To retrieve the admin password, run the following command on your AWX host:

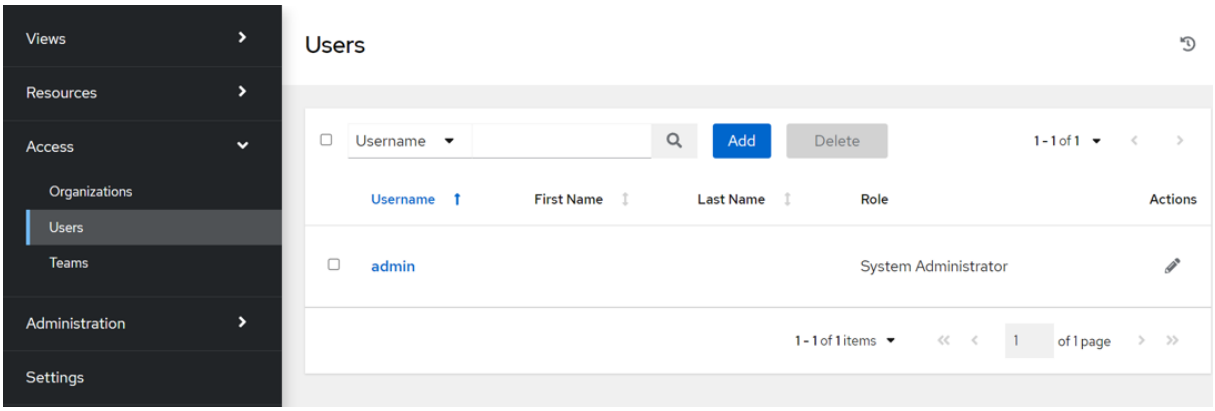
```
$ kubectl get secret awx-cluster-bringup-admin-password -o jsonpath="{.data.password}" | base64 --decode
```

⚠ You can specify the `controller_oauthtoken` variable with OAuth token for your AWX controller instance instead of using `controller_username` and `controller_password` variables.

4.1.1 Users

Refer to the official Ansible Tower Users documentation for information on users.

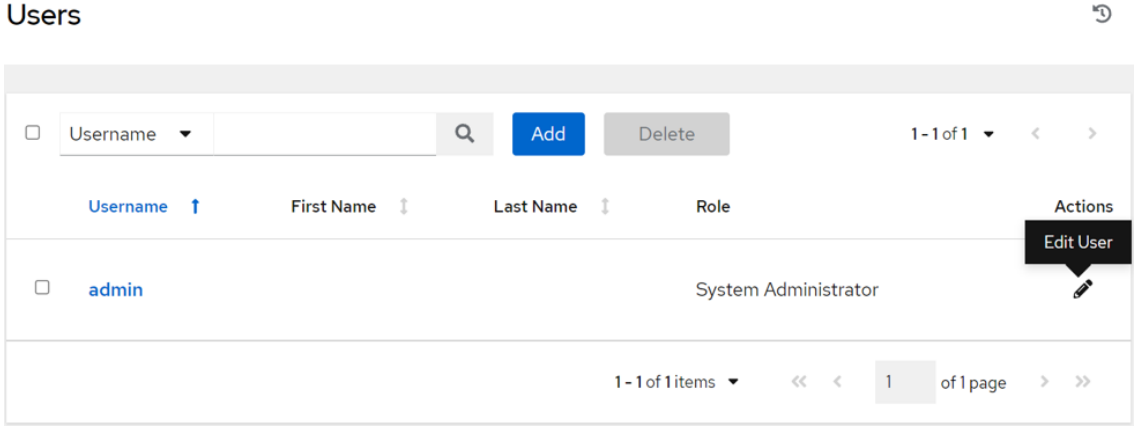
To view a list of the users currently configured, navigate to Access > Users:



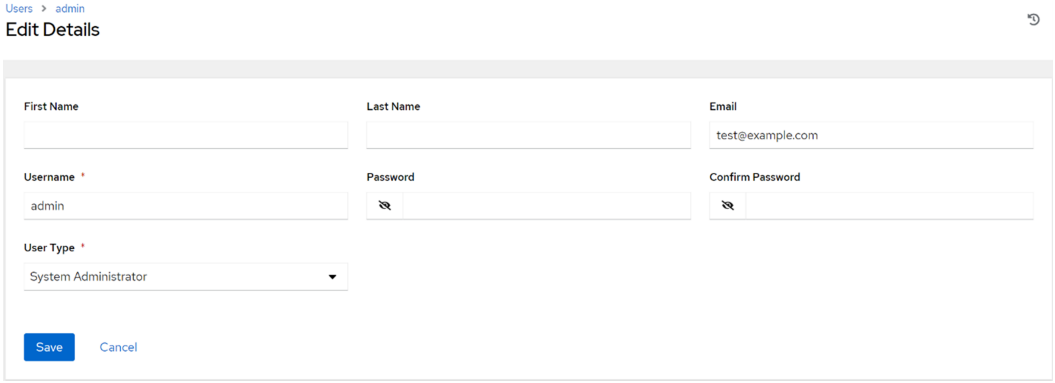
When modifying your own password, log out and log back in for it to take effect.

The following example shows how to update admin user details:

1. Navigate to Access > Users.
2. Click the "Edit User" icon which opens the "Edit Details" dialog box.



3. Update the appropriate details for the following fields:
    - First Name
    - Last Name
    - Email
    - Password
    - Confirmation Password

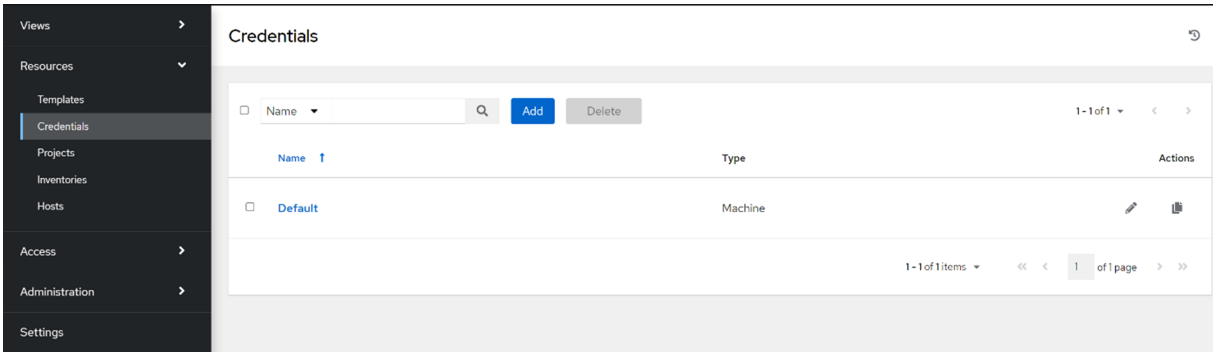


4. Click the "Save" button when finished.

4.1.2 Credentials

Credentials are utilized by AWX for authentication when launching jobs against machines, synchronizing with inventory sources and importing project content from a version control system. Refer to the official Ansible Tower Credentials documentation for further information.

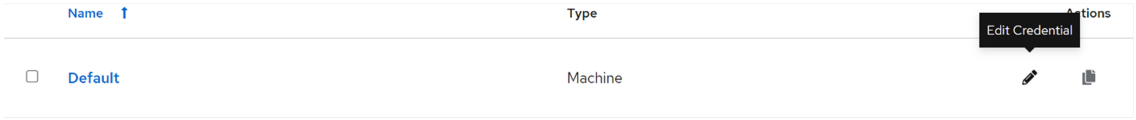
To view a list of the credentials that are currently configured, navigate to Resources > Credentials.



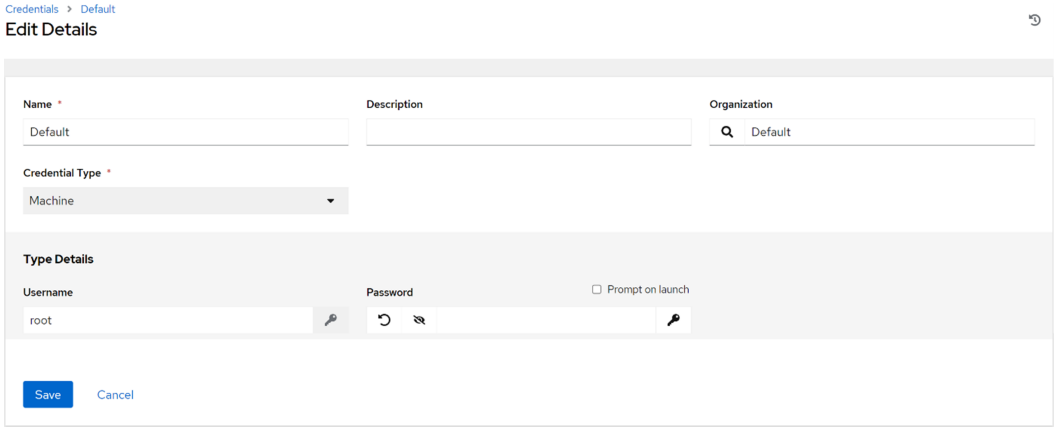
"Default" is configured to use the SSH protocol with the `root` username.

The following example shows how to update credentials:

1. Click the "Edit Credential" icon, which opens the "Edit Details" dialog.



2. Update the appropriate details for the following fields to use for SSH authentication:
   - Username
   - Password

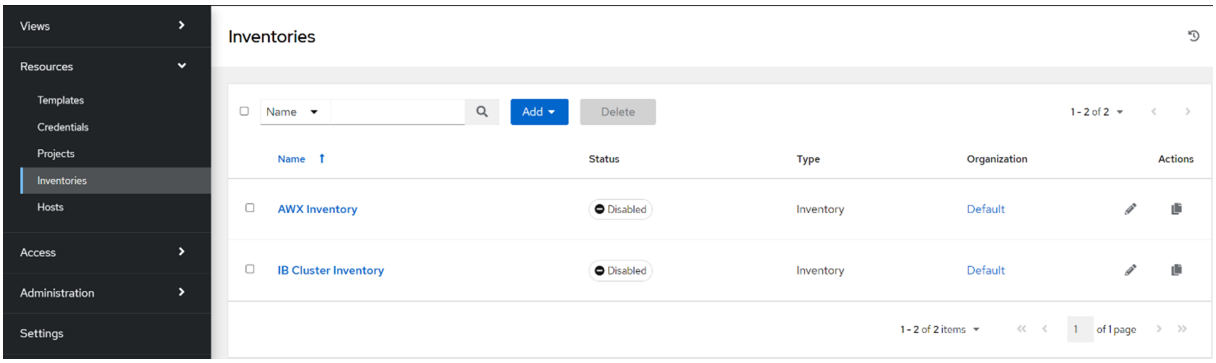


3. Clock "Save" when finished.

4.1.3 Inventories

The inventory includes a collection of hosts against which jobs may be launched. Inventories are divided into groups containing the actual hosts. Refer to the official Ansible Tower Inventories documentation for further information.

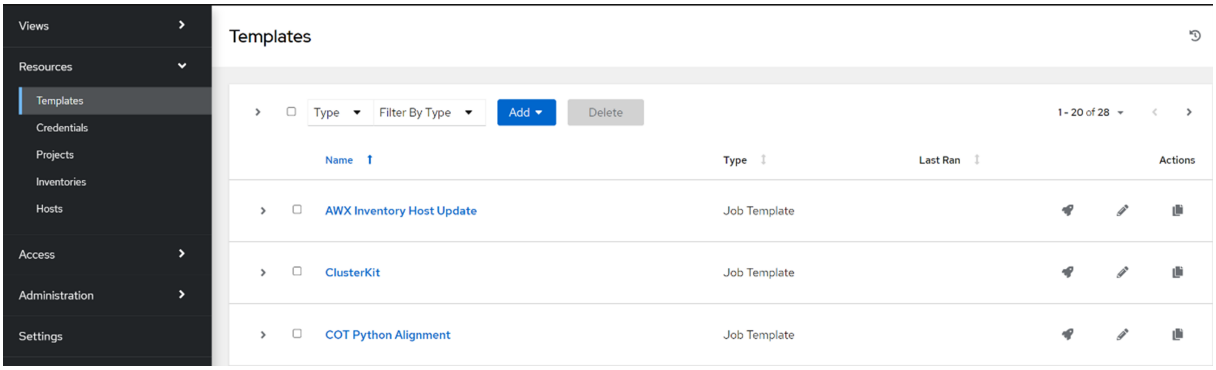
To view a list of the inventories currently available, go to Resources > Inventories.



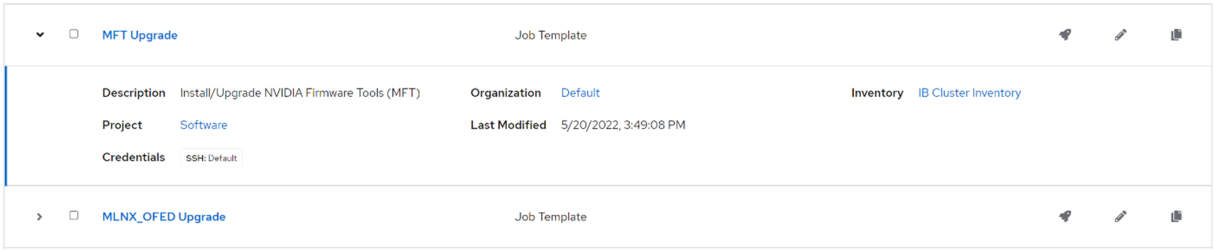
4.1.4 Job Templates

A job template is a definition and set of parameters for running an Ansible job. Job templates are useful to execute a single task many times. Refer to the official Ansible Tower Job Templates documentation for further information.

To view a list of the job templates currently available, go to Resources > Templates.

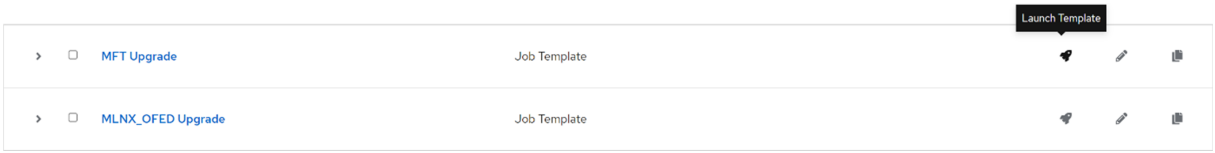


You can view a short description of the job template by selecting one from the list.

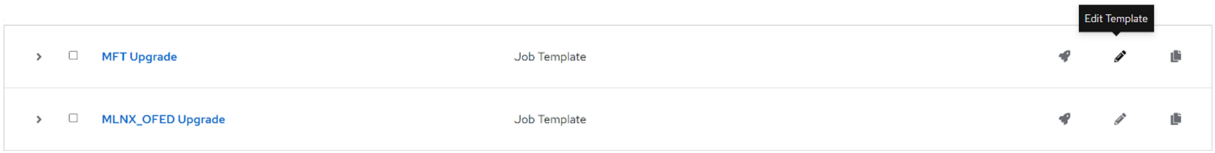


In this example, the inventory to be used for this job template is defined as "IB Cluster Inventory" and the credentials for the hosts are defined as "Default".

To launch a job template, click the "Launch Template" icon.



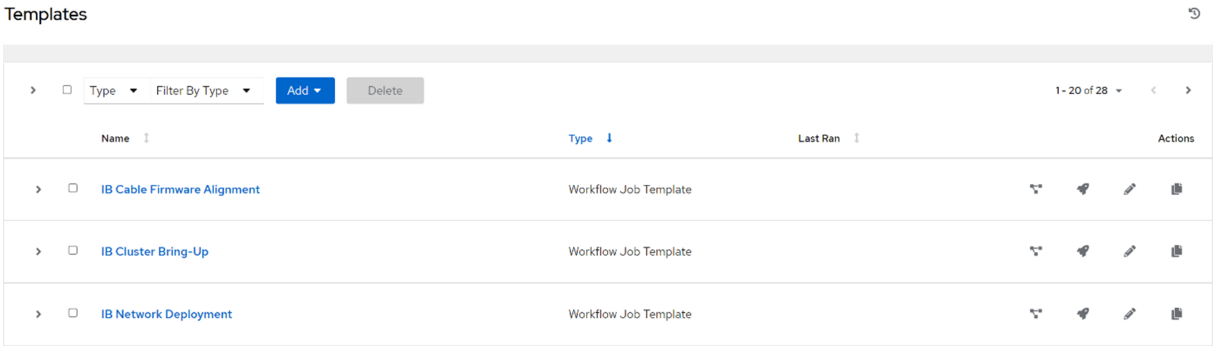
To edit settings for a job template, click the "Edit Template" icon:



4.1.5 Workflow Job Templates

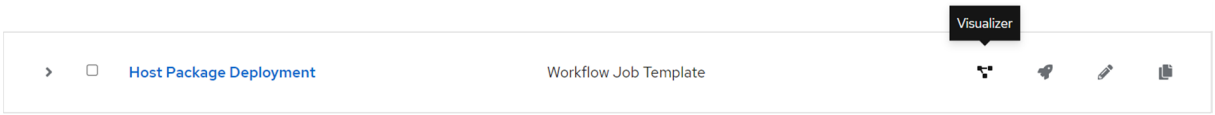
A workflow job template links together a sequence of disparate resources that accomplishes the task of tracking the full set of jobs that are part of the release process as a single unit. Refer to the official Ansible Tower Workflows and Ansible Tower Workflow Job Templates documentation for further information.

To view a list of the workflow job templates currently available, go to Resources > Templates.

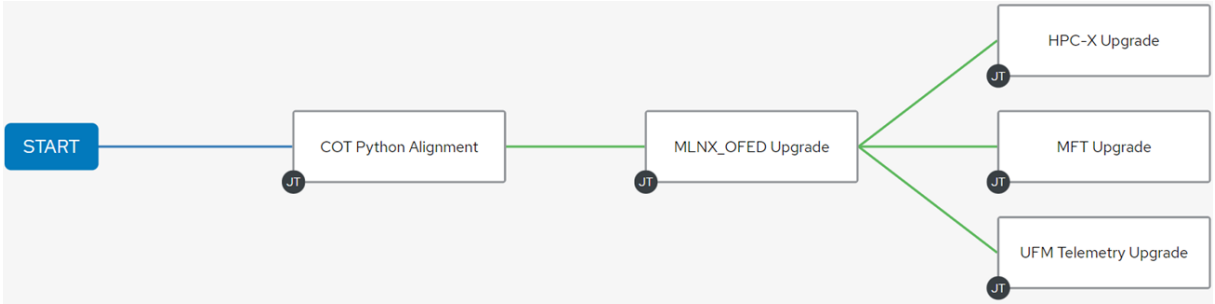


Workflow templates have a visualizer action that displays a graphical way of linking together job templates, workflow templates, project syncs, and inventory syncs.

To view the graph-like structure, click the "Visualizer" icon on a particular workflow:



Example visualization:



4.2 AWX InfiniBand Cluster Deployment

This section describes how to deploy an InfiniBand cluster via AWX interface.

- IB Cluster Inventory
- IB Cluster Deployment
- IB Cluster Deployment Example

- Supported Workflow Templates
- Supported Job Templates

4.2.1 IB Cluster Inventory

This inventory is a collection of hosts against which jobs may be launched to deploy the InfiniBand cluster.

The predefined group named `ib_host_manager` must contain a single host for in-band tasks.

4.2.1.1 Requirements

All hosts in this inventory must have Python 3.6 or greater.

All the hosts associated to ib_host_manager group must have the following:

- Python ≥ 3.6
- MLNX_OFED ≥ 5.6
- MFT ≥ 4.20

> ⚠ MFT and MLNX_OFED packages are installed using the Host Package Deployment workflow.
>
> Make sure both packages are installed when this workflow is not part of the bring-up flow.

4.2.1.2 Pass/Fail Criteria

If the user wants to define specific pass/fail criteria, the `pass_fail_criteria` variable should be utilized. This variable must consist of a dictionary as its value which will have a mapping of a job template (playbook name) to its user-defined criteria (dictionary). The criteria dictionary should contain two special keys, `max_fail_percentage` and `action`.

- `max_fail_percentage` key expects an integer from 0-100 (percentage) as its value. The value represents a percentage (as integer) of failures which are acceptable during the execution of the supported job template. Its default value is 0, which means that in the case of any failures (one host or more) the job template will fail.
- `action` defines the operation to perform if the actual failure percentage is greater than the `max_fail_percentage` value

Supported job template actions (operation types):

Action/Operation	Description
`stop`	Fails the execution of the job

Playbook name (key names supported for `pass_fail_criteria`) to job template name mapping:

Playbook Name	Job Template Name
`hca_fw_update`	HCA Firmware Update
`ib_hca_fw_update`	IB HCA Firmware Update

Playbook Name	Job Template Name
`ib_cable_fw_update`	IB Cable Firmware Update
`ib_switch_fw_update`	IB Externally Managed Switch Firmware Update
`mlnxos_configure`	MLNX-OS Configure
`mlnxos_upgrade`	MLNX-OS Upgrade

Example for `pass_fail_criteria` variable example (placed in the inventory variables list):

```
pass_fail_criteria: '{"hca_fw_update": {"max_fail_percentage": 40, "action": "stop"}, "ib_switch_fw_update":
{"max_fail_percentage": 80, "action": "stop"}}'
```

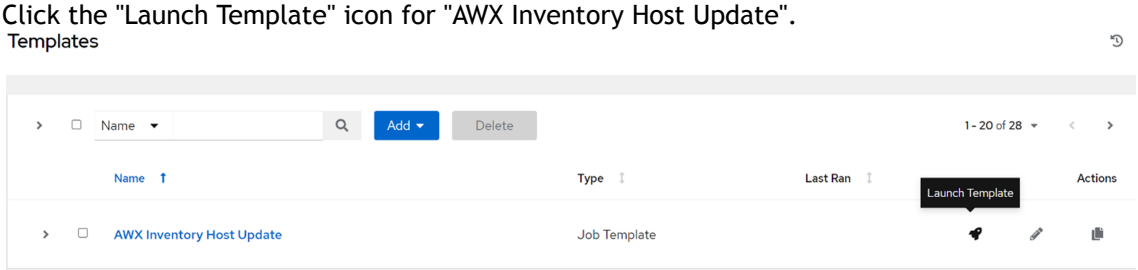
In this example, the user provides criteria for two job templates: HCA Firmware Update (`hca_fw_update`) and IB Externally Managed Switch Firmware Update (`ib_switch_fw_update`).

- For the HCA Firmware Update job template, the `max_fail_percentage` is set to 40. Supposing we have 3 total hosts. If only one host fails, then the job template will pass (33% actual failure which is smaller than 40%). If two hosts fail, the job template will fail (66% actual failure which is greater than 40%).
- For the IB Externally Managed Switch Firmware Update job template, the `max_fail_percentage` is set to 80. For this job template to fail, over 80% of the hosts must fail.

4.2.1.3 IB Cluster Inventory Hosts Example

Add one or more hosts to IB Cluster Inventory.

1. Go to Resources > Templates.
2. Click the "Launch Template" icon for "AWX Inventory Host Update".



3. Specify the following required variables:
   - `api_url` - URL to your cluster bring-up REST API
   - `controller_host` - URL to your AWX controller instance
   - `controller_username` - username for your AWX controller instance
   - `controller_password` - password for your AWX controller instance
   - `inventory` - inventory the host(s) should be made a member of (default: IB Cluster Inventory)
   - `hostname` - a hostname or a hostname expression of the end-host(s) to add/remove
4. Click the Next button.
5. Click the Launch button.

> ⚠ You may specify the `controller_oauthtoken` variable with OAuth token for your AWX controller instance instead of using `controller_username` and `controller_password` variables.

6. Select the Groups tab and click on a group named `ib_host_manager`.
7. Select the Hosts tab and click the "Add" button to add a new host to the group.
8. Select the "Add existing host" option and choose one host to be member of the group.

4.2.1.4 IB Cluster Inventory Variables Example

You can specify variable definitions and values to be applied to all hosts in this inventory.

To define variables for the IB Cluster Inventory:

1. Go to Inventories > IB Cluster Inventory and select the Details tab.
2. Click the Edit button, which opens the Edit details dialog.
3. Enter variables using either JSON or YAML syntax.
4. Click Save when finished.

This section describes all available variables for this inventory.

4.2.1.4.1 Cluster Bring-Up Web Framework Variables

Name	Description
`api_url`	URL to your cluster bring-up REST API
`pypi_url`	URL to your cluster bring-up PyPI repository

4.2.1.4.2 MLNX_OFED Upgrade Variables

Name	Description
`ofed_package_url`	URL of the MLNX_OFED package to download (default: auto-detection). In addition, you must specify the `ofed_version` parameter or use its default value.
`ofed_dependencies`	List of all package dependencies for the MLNX_OFED package
`ofed_install_options`	List of optional arguments for the installation command
`ofed_version`	Version number of the MLNX_OFED package to install

4.2.1.4.3 MFT Upgrade Variables

Name	Description
`mft_package_url`	URL of the MFT package to download (default: auto-detection). In addition, you must specify the `mft_version` parameter or use its default value.

Name	Description
`mft_dependencies`	List of all package dependencies for the MFT package
`mft_install_options`	List of optional arguments for the installation command
`mft_version`	Version number of the MFT package to install

4.2.1.4.4 UFM Telemetry Upgrade Variables

Name	Description
`ufm_telemetry_package_url`	URL for NVIDIA® UFM® Telemetry to download

4.2.1.4.5 HPC-X Upgrade Variables

Name	Description
`hpcx_dir`	Target path for HPC-X installation folder (default: `/opt/nvidia/hpcx`)
`hpcx_package_url`	URL of the HPC-X package to download (default: auto-detection). In addition, you must specify the `hpcx_version` parameter or use its default value.
`hpcx_version`	Version number of the HPC-X package to install
`hpcx_install_once`	Specify whether to install HPC-X package via single host. May be used to install the package on a shared directory (default: false).
`ofed_version`	Version number of the OFED package that compatible to the HPC-X package (default: auto-detection). This variable is mandatory when MLNX_OFED is not installed on the host.

4.2.1.4.6 ClusterKit Variables

Name	Description
`clusterkit_hostname`	Hostname expression that represents the hostname to run tests on (default: all hosts in the inventory)
`clusterkit_options`	List of optional arguments for the tests

4.2.1.4.7 MLNX-OS Upgrade Variables

Name	Description
`mlnxos_switch_hostname`	Hostname expression that represents the names of the switches to upgrade. To skip this parameter using auto-detection of the MLNX-OS switches, UFM Telemetry is required. Make sure to run IB Network Discovery with `ufm_telemetry_path` parameter.
`mlnxos_image_url`	URL of the NVIDIA® MLNX-OS® image to download
`mlnxos_switch_username`	Username to authenticate against target switches
`mlnxos_switch_password`	Password to authenticate against target switches

4.2.1.4.8 Externally Managed Switch Firmware Upgrade Variables

Name	Description
`switch_fw_image_url`	URL of the firmware image to download
`switch_psid`	PSID of the externally managed switch device to upgrade

4.2.1.4.9 HCA Firmware Upgrade Variables

Name	Description
`hca_fw_image_url`	URL of the firmware image to download
`hca_psid`	PSID of the HCA device to upgrade

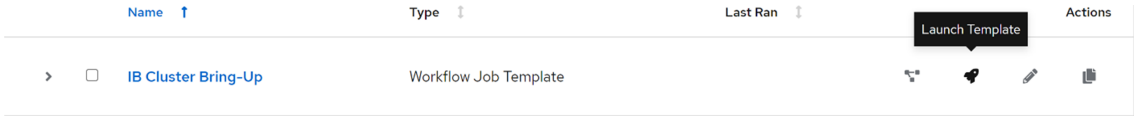
4.2.1.4.10 Cable Firmware Upgrade (IFFU) Variables

Name	Description
`iffu_auto_update`	Specify whether to update all supported cables/transceivers connected to the host/switch (default: true)
`iffu_fw_version`	Firmware version number of the cable image to update. This variable is mandatory when the cable image is not queryable.
`iffu_image_url`	URL of the firmware image to download
`cable_identifier`	Identifier of the cable/transceiver to update (e.g., OSFP, QSFP56)
`cable_part_number`	Part number of the cable/transceiver to upgrade

4.2.2 IB Cluster Deployment

To deploy the InfiniBand cluster:

1. Go to Resources > Templates > IB Cluster Bring-Up.
2. Click the Launch icon.



4.2.3 IB Cluster Deployment Example

This example:

1. Configures 3 hosts to IB Cluster Inventory.
2. Configures a single host to be member of `ib_host_manager`.
3. Deploys an InfiniBand cluster.

The following example shows how to deploy an InfiniBand cluster that performs the following:

1. Updates MLNX_OFED to version number 5.6-1.0.3.3 on all hosts of this inventory.
2. Updates MFT to version number 4.20.0-34 on all hosts of this inventory.
3. Updates HPC-X to version number 2.11.0 on all hosts of this inventory.
4. Updates NVIDIA® MLNX-OS® to version number 3.9.3124 on 5 switches.
5. Updates firmware for NVIDIA® Quantum InfiniBand to version number 27.2008.3328.
6. Updates firmware for NVIDIA® ConnectX®-6 InfiniBand to version number 20.31.1014.
7. Updates firmware for NVIDIA® AOC InfiniBand HDR cables to version number 38.100.121.
8. Runs ClusterKit tests for 1 minute on 2 hosts of this inventory.

Using YAML syntax, the following variables are used in this example:

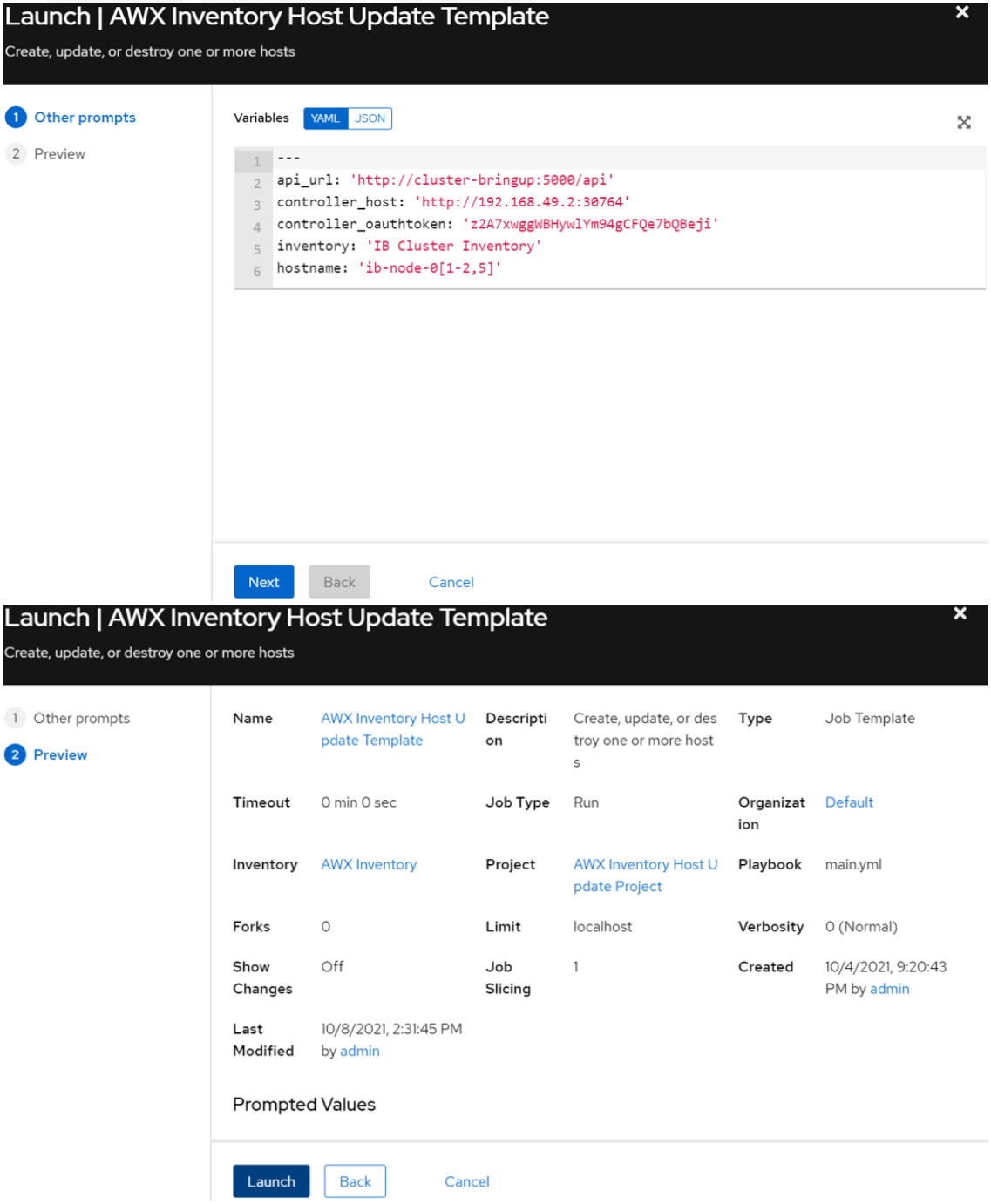
```
# Ansible parameters
ansible_python_interpreter: '/usr/bin/python3'

# Cluster bring-up WEB framework parameters
api_url: 'http://cluster-bringup:5000/api'

# UFM Telemetry Upgrade parameters
ufm_telemetry_package_url: 'http://cluster-bringup:5000/downloads/collectx-1.10.5-5968674.x86_64_el8.2-
bringup.tar.gz'

# MLNX-OS switches parameters
mlnxos_switch_hostname: 'ib-switch-t[1-2],ib-switch-l[1-2],ib-switch-s1'
mlnxos_switch_username: 'admin'
mlnxos_switch_password: 'my_admin_password'
mlnxos_image_url: 'http://cluster-bringup:5000/downloads/sx_mlnx_os/sx_mlnx_os-3.9.3124/sx_mlnx_os-3.9.3124-X86_64/
image-X86_64-3.9.3124.img'

# IB cables firmware update (IFFU) parameters
iffu_image_url: 'http://cluster-bringup:5000/downloads/linkx/rel-38_100_121/iffu/hercules2.bin'
iffu_auto_update: true
cable_part_number: 'MFS1S00-H0(03|05|10)E_QP'
iffu_fw_version: '38.100.121'

 # IB externally managed switch firmware update parameters
switch_fw_image_url: 'https://www.mellanox.com/downloads/firmware/fw-Quantum-rel-27_2008_3328-MQM8790-
HS2X_Ax.bin.zip'
switch_psid: 'MT_0000000063'

# HCA firmware update parameters
hca_fw_image_url: 'https://www.mellanox.com/downloads/firmware/fw-ConnectX6-rel-20_31_1014-MCX654106A-HCA_Ax-
UEFI-14.24.13-FlexBoot-3.6.403.bin.zip'
hca_psid: 'MT_0000000228'

# Software packages parameters
hpcx_version: '2.11.0'
ofed_version: '5.6-1.0.3.3'
ofed_dependencies: ['python3-devel', 'createrepo', 'kernel-rpm-macros', 'kernel-modules-extra']
mft_version: '4.20.0-34'
```

```
mft_dependencies: ['rpm-build']

# ClusterKit parameters
clusterkit_hostname: 'ib-node-0[1-2]'
clusterkit_options: ['--traffic', '1']
```

4.2.3.1 IB Cluster Inventory Hosts Example

To configure the hosts for this inventory:

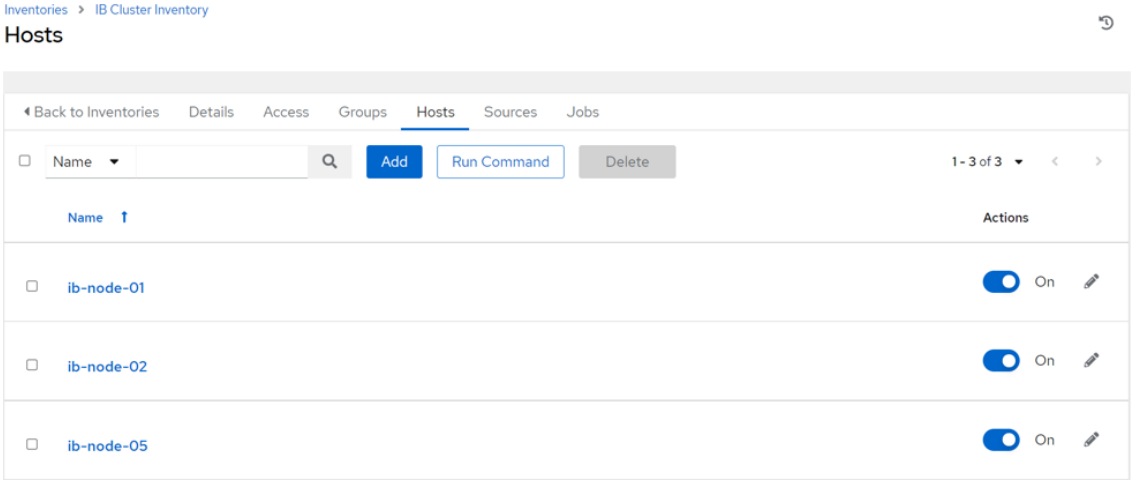
1. Add `ib-node-01`, `ib-node-02`, and `ib-node-05` hosts to IB Cluster Inventory.



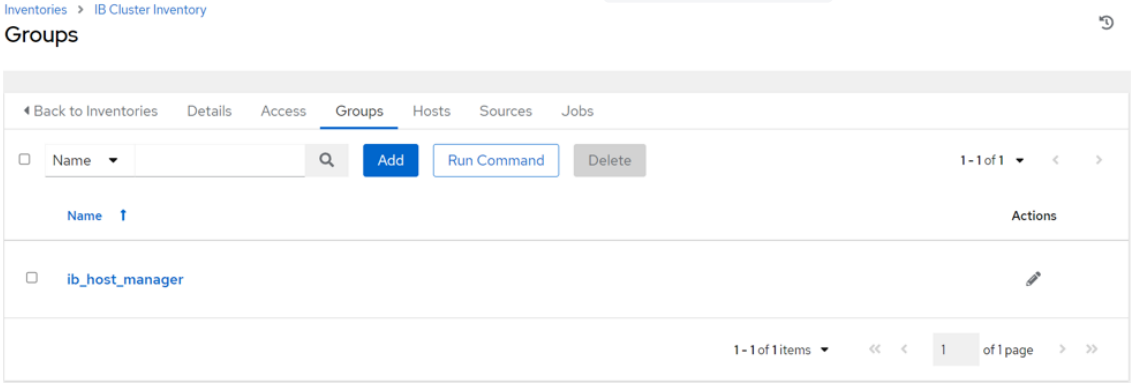
2. Verify the job output.

```
TASK [Show status details about hosts update] ********************************** 13:46:38
ok: [localhost] => (item=ib-node-01) => {
    "msg": "ib-node-01 host was loaded successfully"
}
ok: [localhost] => (item=ib-node-02) => {
    "msg": "ib-node-02 host was loaded successfully"
}
ok: [localhost] => (item=ib-node-05) => {
    "msg": "ib-node-05 host was loaded successfully"
}

TASK [Show status details about inventory update] ****************************** 13:46:38
ok: [localhost] => {
    "msg": "Inventory update completed for IB Cluster Inventory, loaded 3 hosts, skipped 0 hosts\n"
}

PLAY RECAP ********************************************************************* 13:46:38
localhost                  : ok=8    changed=1    unreachable=0    failed=0    skipped=8    rescued=0    ignored=0
```

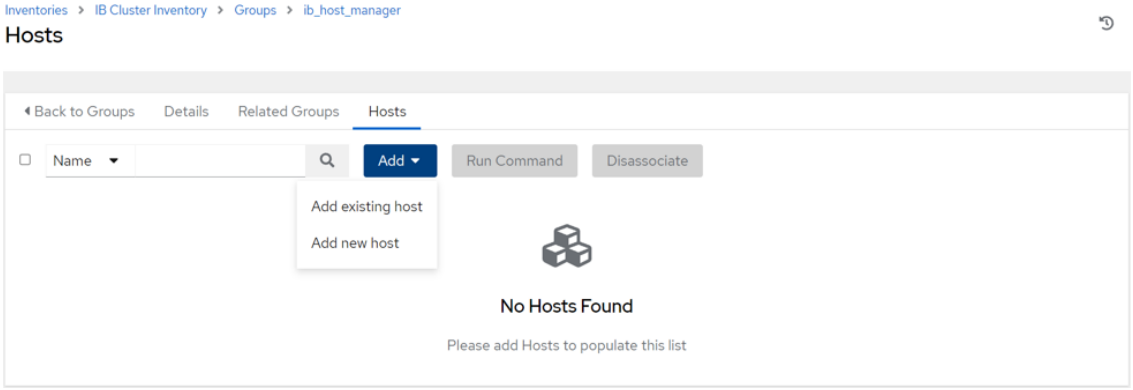
3. See the added hosts by going to Inventories > IB Cluster Inventory and selecting the Hosts tab.



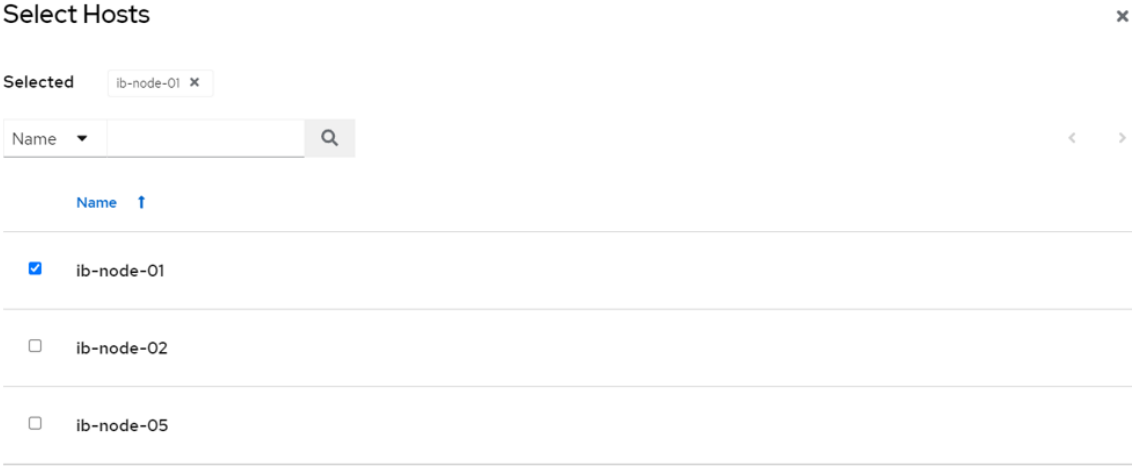
4. Select the Groups tab and click on a group named `ib_host_manager`.



5. Select the Hosts tab and click the Add button to add a new host to the group.

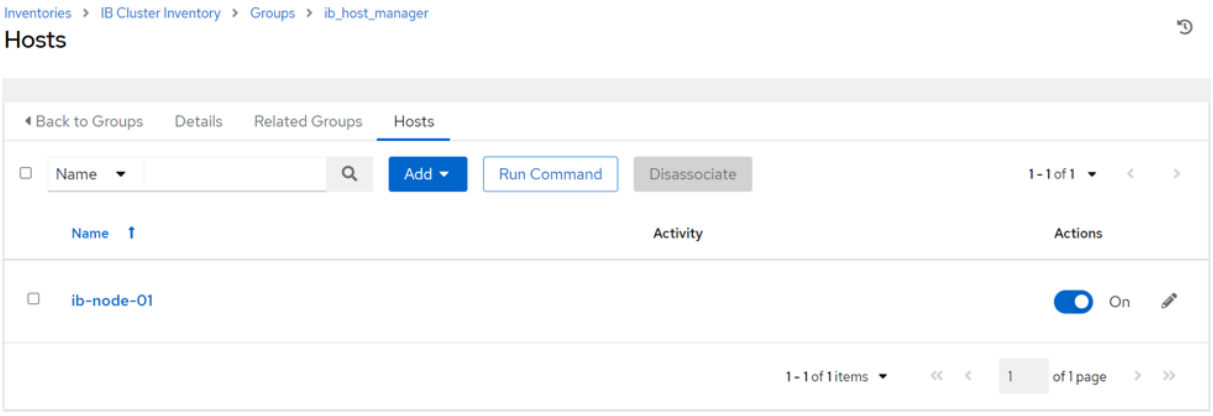


6. Select the "Add existing host" option and mark one the hosts to be a member of the group.



7. Click Save when finished.

Once the host is successfully added, it will be member of the `ib_host_manager` group.

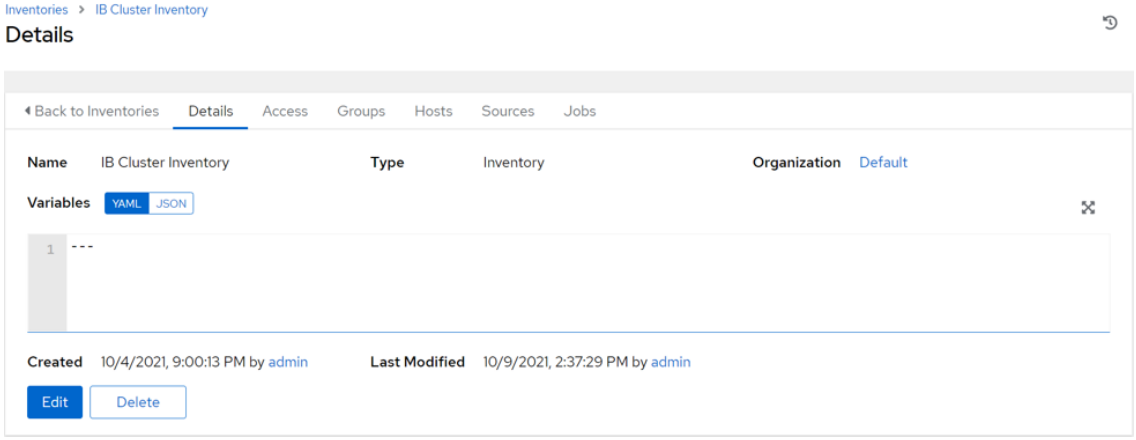


> When the host is not member of this inventory, you need to select the "Add new host" option instead of "Add existing host" option.

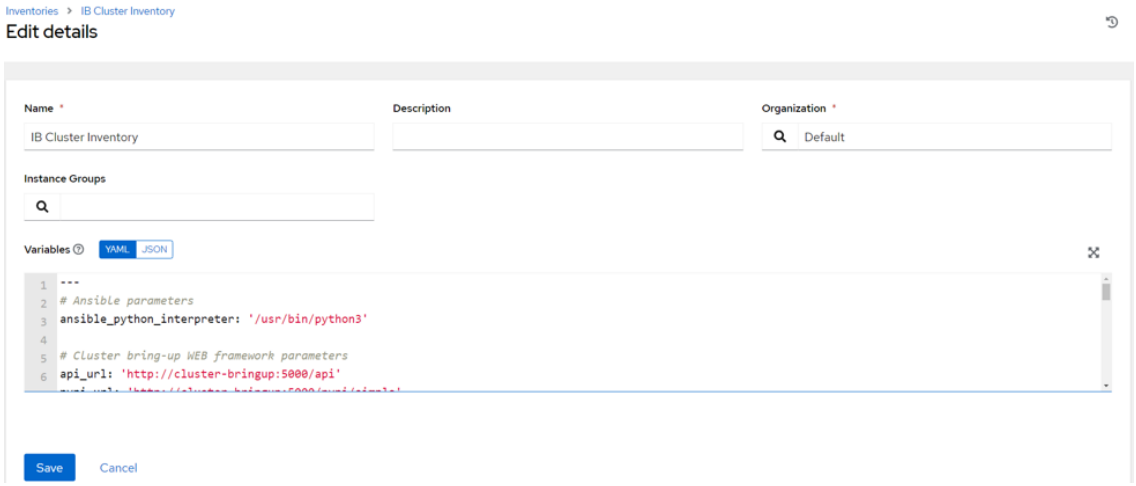
4.2.3.2 IB Cluster Inventory Variables Example

To configure the variables for this inventory:

1. Go to Inventories > IB Cluster Inventory and select the Details tab.



2. Click the Edit icon which opens the "Edit details" dialog.

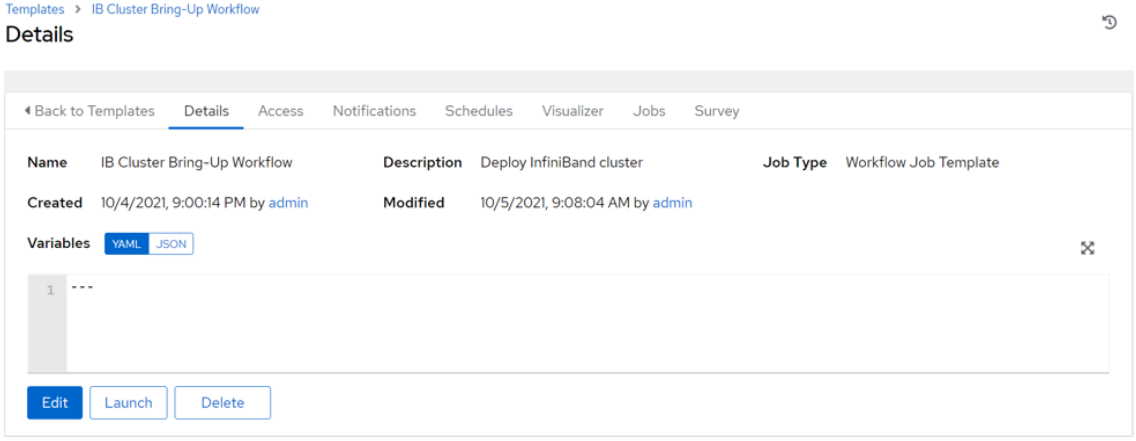


3. Enter variables using either JSON or YAML syntax.
4. Click Save when finished.

4.2.3.3 IB Cluster Deployment Example

To deploy the InfiniBand cluster:

1. Go to Resources > Templates > IB Cluster Bring-Up.
2. Click the Launch button.



Once the job is completed successfully, the output of the job should look like this:

4.2.4 Supported Workflow Templates

The following subsections describe the currently supported workflow templates.

4.2.4.1 IB Cluster Bring-Up

This section describes how to deploy the InfiniBand cluster.

This procedure is a sequence of the following workflow templates:

1. IB Network Deployment
2. IB Network Verification

These workflow templates are linked together to deploy the InfiniBand cluster:

1. Deploy InfiniBand network.
2. Verify the InfiniBand network.

The following diagram shows the nodes of this workflow:

The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on the "IB Cluster Bring-Up".

4.2.4.2 IB Network Deployment

This section describes how to deploy InfiniBand network.

This procedure is a sequence of the following workflow and job templates:

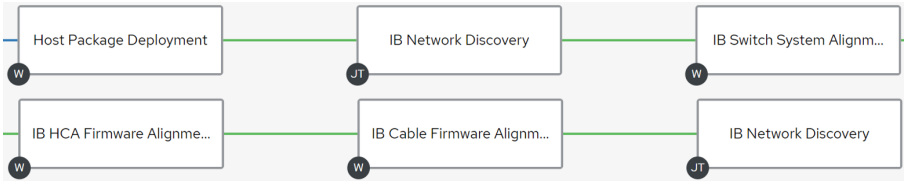
1. Host Package Deployment
2. IB Network Discovery
3. IB Switch System Alignment
4. IB HCA Firmware Alignment
5. IB Cable Firmware Alignment
6. IB Network Discovery

These workflow templates and job templates are linked together to deploy the InfiniBand cluster:

1. Ensure software packages are installed on the hosts.
2. Discover InfiniBand topology and update the database with the discovered topology.
3. Update system firmware/MLNX-OS software on InfiniBand switches.
4. Update firmware on InfiniBand HCAs.
5. Update cables' transceivers' firmware on InfiniBand cable devices.

6. Discover InfiniBand topology and update the database with the discovered topology.

The following diagram shows the nodes of this workflow:



The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on the "IB Network Deployment".

> ⚠ Make sure that all variables for this workflow are defined.

4.2.4.3 IB Network Verification

This section describes how to verify the InfiniBand network.

This procedure is a sequence of the following job templates:

1. IB Topology Comparison
2. ClusterKit
3. IB Topology Comparison
4. IB Fabric Health Checks
5. Fabric Health Counters Collection

These workflow templates are linked together to deploy the InfiniBand cluster:

1. Discover InfiniBand topology and create a file with the discovered topology.
2. Run ClusterKit tests.
3. Discover InfiniBand topology and compare against the discovered topology.
4. Performs diagnostic fabric health check of the fabric's state.
5. Performs the collection of fabric counters with and without traffic.

The following diagram shows the nodes of this workflow:

The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on the "IB Network Verification".

4.2.4.4 Host Package Deployment

This section describes how to deploy NVIDIA software packages on one or more hosts.

Refer to the official NVIDIA Software Products documentation for further information.

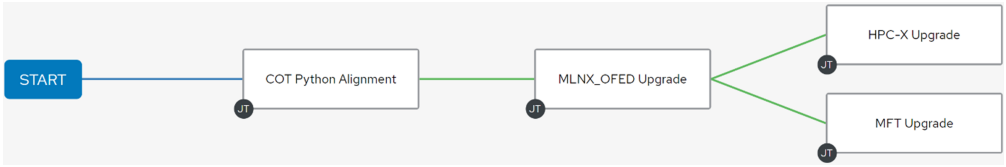
This procedure is a sequence of the following job templates:

1. COT Python Alignment
2. MLNX_OFED Upgrade
3. MFT Upgrade
4. HPC-X Upgrade
5. UFM Telemetry Upgrade

These job templates are linked together to deploy NVIDIA software packages:

1. Ensure the Python environment for the cluster orchestration tool (COT) is installed.
2. Ensure the MLNX_OFED Linux driver is installed.
3. Ensure the HPC-X Software Toolkit is installed.
4. Ensure the MFT is installed.
5. Install UFM Telemetry if package URL is provided.

The following diagram shows the nodes of this workflow:



The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on the "Host Package Deployment".

The following variables are available for deploying software packages:

Name	Description
`force`	Install the packages even if the packages are already up to date
`hpcx_checksum`	Checksum of the HPC-X package to download
`hpcx_dir`	Target path for HPC-X installation folder
`hpcx_install_once`	Specify whether to install HPC-X package via single host. May be used to install the package on a shared directory.
`hpcx_package_url`	URL of the HPC-X package to download (default: auto-detection). In addition, you must specify the `hpcx_version` parameter or use its default value.
`hpcx_version`	Version number of the HPC-X package to install
`mft_checksum`	Checksum of the MFT package to download
`mft_dependencies`	List of all package dependencies for the MFT package
`mft_install_options`	List of optional arguments for the installation command
`mft_package_url`	URL of the MFT package to download (default: auto-detection). In addition, you must specify the `hpcx_version` parameter or use its default value.
`mft_version`	Version number of the MFT package to install
`ofed_checksum`	Checksum of the MLNX_OFED package to download

Name	Description
`ofed_dependencies`	List of all package dependencies for the MLNX_OFED package
`ofed_install_options`	List of optional arguments for the installation command
`ofed_package_url`	URL of the MLNX_OFED package to download (default: auto-detection). In addition, you must specify the `ofed_version` parameter or use its default value.
`ofed_version`	Version number of the MLNX_OFED package to install
`working_dir`	Path to the working directory on the host

The following are variable definitions and default values for deploying software packages:

Name	Default	Type
`force`	`false`	Boolean
`hpcx_checksum`	`''`	String
`hpcx_dir`	`'/opt/nvidia/hpcx'`	String
`hpcx_install_once`	`false`	Boolean
`hpcx_package_url`	`''`	String
`hpcx_version`	`'2.15.0'`	String
`mft_checksum`	`''`	String
`mft_dependencies`	`[]`	List[String]
`mft_install_options`	`[]`	List[String]
`mft_package_url`	`''`	String
`mft_version`	`'4.24.0-72'`	String
`ofed_checksum`	`''`	String
`ofed_dependencies`	`[]`	List[String]
`ofed_install_options`	`[]`	List[String]
`ofed_package_url`	`''`	String
`ofed_version`	`'23.04-0.5.3.3'`	String
`working_dir`	`'/tmp'`	String

4.2.4.5 IB Cable Firmware Alignment

This section describes how to update the firmware of the transceivers on one or more cable devices.

Refer to the official NVIDIA Cable Firmware Update documentation for further information.

This procedure is a sequence of the following job templates:

1. Lookup InfiniBand Cables
2. IB Cable Firmware Update

These job templates are linked together to update cable transceiver firmware:

1. Lookup for InfiniBand cables by a specific part number.
2. Update cable transceiver firmware on the specified cable devices.

> ⚠ This workflow relies on updated topology, so make sure the topology is up to date by running network discovery.

The following diagram shows the nodes of this workflow:

The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on the "IB Cable Firmware Alignment".

> ⚠ Make sure that all required variables described below are defined before running this job. You can define these variables either as inventory variables or as job template variables.

The following variables are required for updating cable transceiver firmware:

Name	Default	Type
`api_url`	URL to your Cluster Bring-up REST API	String
`iffu_image_url`	URL of the firmware image to download	String
`cable_part_number`	Part number of the cable/transceiver to update. Can be provided as a regular expression (e.g., `'MFS1S00-H0(03|05|10)E_QP'`).	String

The following is an additional variable required for updating cable transceiver firmware on hybrid products (e.g., NVIDIA AOC splitter, IB twin port HDR, OSFP-to-2xQSFP56):

Name	Default	Type
`cable_identifier`	Identifier of the cable/transceiver to update (e.g., OSFP, QSFP56)	String

> ⚠ Cable firmware may be provided as a zip file. For this purpose, either `unzip` or `zipinfo` must be installed when using Ansible. For more information, refer to Ansible's documentation.

The following variables are available to update cable transceiver firmware:

Name	Description
activate_delay	Time (in seconds) to wait before activating all updated cables
activate_delay_factor	Multiplying factor used to adjust delay after loading new firmware. Its value must be greater than or equal to 1.
activate_image_retries	Maximum number of retries available for activate task to complete
activate_image_wait	Time (in seconds) to wait for activate task to complete
burn_image_retries	Maximum number of retries available for burn task to complete
burn_image_wait	Time (in seconds) to wait for burn task to complete
clear_semaphore	Specify to clear the flash semaphore before update started
cot_python_interpreter	Path to cluster orchestration Python interpreter
exclude_devices	List of GUIDS/LIDs representing the InfiniBand devices to ignore
exclude_ports	Ports labels that represent the cable devices to ignore
ib_device	Specify the name of the In-Band device to use (e.g., mlx5_0)
iffu_activate_auto_update	Specify whether to activate all updated cables/transceivers connected to the host/switch. This variable is not available when iffu_auto_update is set to true.
iffu_auto_update	Specify whether to update all supported cables/transceivers connected to the host/switch
iffu_fw_version	Firmware version number of the cable image to update. This variable is mandatory when the cable image is not queryable.
iffu_image_checksum	Checksum of firmware image to download
max_device_ports	Limit the number of cables/transceivers to burn on each host/switch device. This variable is not available when iffu_auto_update is set to true.
query_image_retries	Maximum number of retries available for query task to complete
query_image_wait	Time (in seconds) to wait for query to complete
stop_on_failure	Specifies to stop the update firmware execution on the first failure
working_dir	Path to the working directory on the host

The following are variables definitions and default values for update cables transceivers' firmware:

Name	Default	Type
activate_delay	60	Integer
activate_delay_factor	2	Decimal
activate_image_retries	10	Integer
activate_image_wait	120	Integer

Name	Default	Type
`burn_image_retries`	`20`	Integer
`burn_image_wait`	`120`	Integer
`clear_semaphore`	`false`	Boolean
`cot_python_interpreter`	`'/opt/nvidia/cot/client/bin/python'`	String
`exclude_devices`	`[]`	List[String]
`exclude_ports`	`[]`	List[String]
`ib_device`	`''`	String
`iffu_activate_auto_update`	`false`	Boolean
`iffu_auto_update`	`true`	Boolean
`iffu_fw_version`	`''`	String
`iffu_image_checksum`	`''`	String
`max_device_ports`	`-1`	Integer
`query_image_retries`	`120`	Integer
`query_image_wait`	`10`	Integer
`stop_on_failure`	`false`	Boolean
`working_dir`	`'/tmp'`	String

The following are the formats of port labels for each product:

- NVIDIA Quantum-2 - `<Node GUID>/P<ASIC>/<cage>/<port>` (e.g., `0x900a84030040aab0/P1/3/1`)
- NVIDIA Quantum - `<Node GUID>/P<port>` (e.g., `0x900a84030040bbb0/P3`)
- NVIDIA® ConnectX®-6 - `<Node GUID>/P1` (e.g., `0xb8cef60300ff8727/P1`)
- NVIDIA® ConnectX®-7 - `<Node GUID>/P1` (e.g., `0x08c0eb0300e877c4/P1`)

4.2.4.6 IB HCA Firmware Alignment

This section describes how to update the firmware on one or more InfiniBand HCAs.

Refer to the official NVIDIA Firmware Downloads documentation for further information.

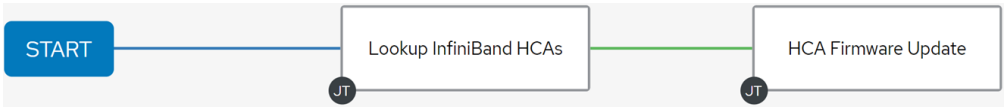
This procedure is a sequence of the following job templates:

1. Lookup InfiniBand HCAs
2. HCA Firmware Update

These job templates are linked together to update firmware on InfiniBand HCAs:

1. Lookup for InfiniBand HCAs by a specific PSID.
2. Update firmware on the specified InfiniBand HCAs.

The following shows a diagram of this workflow:



The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "IB HCA Firmware Alignment".

> ⚠ HCA firmware update on the SM (subnet manager) host requires stopping the SM service before running the job.

The following variables are required to update HCAs firmware:

Name	Default	Type
`api_url`	URL to your cluster bring-up REST API	String
`hca_fw_image_url`	URL of the firmware image to download	String
`hca_psid`	PSID of the HCA device to update	String

> ⚠ HCA firmware may be provided as a zip file. For this purpose, either `unzip` or `zipinfo` must be installed when using Ansible. For more information, refer to Ansible's documentation.

The following variables are available for update HCAs firmware:

Name	Description
`burn_image_retries`	Maximum number of retries available for burn task to complete
`burn_image_wait`	Time (in seconds) to wait for burn task to complete
`clear_semaphore`	Specify to clear the flash semaphore before update started
`ib_device`	Specify the name of the in-band device to use (e.g., 'mlx5_0')
`exclude_devices`	List of GUID/LIDs representing the HCAs to ignore
`hca_fw_image_checksum`	Checksum of firmware image to download
`psid`	Alias name for `hca_psid`. This variable is not available when the `hca_psid` variable is set.
`query_image_retries`	Maximum number of retries available for query task to complete
`query_image_wait`	Time (in seconds) to wait for query task to complete

Name	Description
subnet	Subnet name which the HCAs are member of
working_dir	Path to the working directory on the host

The following variables are available for update HCAs firmware:

Name	Default	Type
burn_image_retries	10	Integer
burn_image_wait	120	Integer
clear_semaphore	false	Boolean
ib_device	''	String
exclude_devices	[]	List[String]
hca_fw_image_checksum	''	String
query_image_retries	5	Integer
query_image_wait	30	Integer
subnet	'infiniband-default'	String
working_dir	'/tmp'	String

The following example shows the firmware image for NVIDIA® ConnectX®-6 VPI adapter cards on the ConnectX VPI/InfiniBand Firmware Download Center:


```
hca_fw_image_url: 'https://www.mellanox.com/downloads/firmware/fw-ConnectX6-rel-20_31_1014-MCX654106A-HCA_Ax-
UEFI-14.24.13-FlexBoot-3.6.403.bin.zip'
hca_fw_image_checksum: 'md5:8055b27dd7a3ac7ae60300a37455a7a4'
hca_psid: 'MT_0000000228'
```

4.2.4.7 IB Switch System Alignment

This section describes how to update system firmware/software on one or more InfiniBand switches.

This procedure is a sequence of the following job templates:

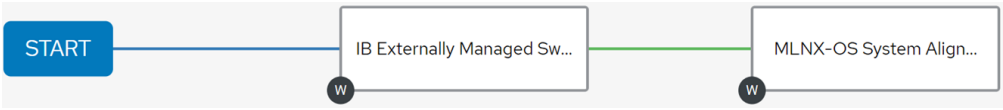
1. IB Externally Managed Switch Firmware Alignment
2. MLNX-OS System Alignment

These job templates are linked together to update firmware on InfiniBand switches:

1. Update firmware on externally managed InfiniBand switches.
2. Upgrade ASIC firmware/MLNX-OS software on InfiniBand switches.

> This workflow relies on the updated topology. Therefore, make sure the topology is up-to-date by running network discovery.

The following diagram shows the nodes of this workflow:



The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "IB Switch System Alignment".

4.2.4.8 IB Externally Managed Switch Firmware Alignment

This section describes how to update firmware on one or more externally managed InfiniBand switches.

Refer to the official NVIDIA Firmware Downloads documentation for further information.

This procedure is a sequence of the following job templates:

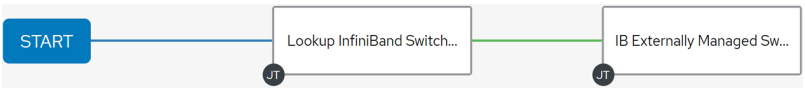
1. Lookup InfiniBand Switches
2. IB Externally Managed Switch Firmware Update

These job templates are linked together to update firmware on InfiniBand switches:

1. Lookup for externally managed InfiniBand switches by a specific PSID.
2. Update firmware on the specified externally managed InfiniBand switches.

> This workflow relies on the updated topology. Therefore, make sure the topology is up-to-date by running network discovery.

The following diagram shows the nodes of this workflow:



The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "IB Externally Managed Switch Firmware Alignment".

⚠ Make sure all required variables described below are defined before running this job. You can define these variables either as inventory variables or as job template variables.

The following variables are required to update externally managed InfiniBand switch firmware:

Name	Default	Type
`api_url`	URL to your cluster bring-up REST API	String
`switch_fw_image_url`	URL of the firmware image to download	String
`switch_psid`	PSID of the externally managed switch device to update	String

⚠ Switch firmware may be provided as a zip file. For this purpose, either `unzip` or `zipinfo` must be installed when using Ansible. For more information, refer to Ansible's documentation.

The following variables are available to update externally managed switch firmware:

Name	Description
`burn_image_retries`	Maximum number of retries available for burn task to complete
`burn_image_wait`	Time (in seconds) to wait for burn task to complete
`clear_semaphore`	Specify to clear the flash semaphore before update started
`exclude_devices`	List of GUIDS/LIDs representing the switches to ignore
`ib_device`	Specifies the name of the in-band device to use (e.g., 'mlx5_0')
`psid`	Alias name for `switch_psid`. This variable item is not available when the `switch_psid` variable is set.
`query_image_retries`	Maximum number of retries available for query task to complete
`query_image_wait`	Time (in seconds) to wait for query task to complete
`subnet`	Subnet name which the externally managed switches are member of
`switch_fw_image_checksum`	Checksum of firmware image to download
`working_dir`	Path to the working directory on the host

The following are variables definitions and default values for update externally managed switches firmware:

Name	Default	Type
burn_image_retries	10	Integer
burn_image_wait	120	Integer
clear_semaphore	false	Boolean
exclude_devices	[]	List[String]
ib_device	''	String
query_image_retries	5	String
query_image_wait	30	Integer
subnet	'infiniband-default'	String
switch_fw_image_checksum	''	String
working_dir	'/tmp'	String

The following example shows firmware image for NVIDIA Quantum-based InfiniBand switch platforms on the Quantum InfiniBand Firmware Download Center:

Quantum InfiniBand Firmware Download Center

Current Versions | Archive Versions | START OVER

Version (Current)	OPN	PSID	Download/ Documentation
27.2008.3328	MQM8790	MT_0000000063	**Quantum:** fw-Quantum-rel-27_2008_3328-MQM8790-HS2X_Ax **MD5SUM:** 953dca31ed40e0a90e991b4291f0fa2d **SHA256:** 0ee31c68dbd7982120d7176461da79101e0e310bd7cd280835b289b2e734e66a **Release Date:** 21 September 2021 **Documentation:** Release Notes

```
switch_fw_image_url: 'https://www.mellanox.com/downloads/firmware/fw-Quantum-rel-27_2008_3328-MQM8790-
HS2X_Ax.bin.zip'
switch_fw_image_checksum: 'md5:953dca31ed40e0a90e991b4291f0fa2d'
switch_psid: 'MT_0000000063'
```

4.2.4.9 MLNX-OS System Alignment

This section describes how to update system firmware/MLNX-OS software on one or more switches.

Refer to the official NVIDIA® MLNX-OS® documentation for further information.

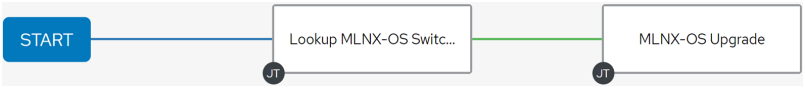
This procedure is a sequence of the following job templates:

1. Lookup MLNX-OS Switches
2. MLNX-OS Upgrade

These job templates are linked together to update software on InfiniBand switches:

1. Lookup for MLNX-OS switches hostnames.
2. Update system firmware/OS software on the specified switches.

The following diagram shows the nodes of this workflow:



The following instructions describe how to run this workflow:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "MLNX-OS System Alignment".

> ⚠ Make sure all required variables described below are defined before running this job. You can define these variables either as inventory variables or as job template variables.

The following variables are required to update a MLNX-OS system:

Name	Default	Type
`api_url`	URL to your cluster bring-up REST API	String
`mlnxos_image_url`	URL of the MLNX-OS image to download	String
`switch_username`	Username to authenticate against target switches	String
`switch_password`	Password to authenticate against target switches	String
`mlnxos_switch_hostname`	Hostname expression that represents the names of the switches to upgrade. To skip this parameter using auto-detection of the MLNX-OS switches, NVIDIA® UFM® Telemetry is required. Make sure to run IB Network Discovery with `ufm_telemetry_path` parameter.	String

The following variables are available to update a MLNX-OS system:

Name	Description
`command_timeout`	Time (in seconds) to wait for the command to be completed
`force`	Specify to update MLNX-OS system even if it is already up to date
`image_url`	Alias name for `mlnxos_image_url`. This variable item is not available when the `mlnxos_image_url` is set.
`mlnxos_switch_username`	Alias name for `switch_username`. This variable item is not available when the `switch_username` is set.

Name	Description
`mlnxos_switch_username`	Alias name for `switch_password`. This variable item is not available when the `switch_password` is set.
`reload_command`	Specify an alternative command for reload switch system
`reload_timeout`	Time (in seconds) to wait for the switch system to be reloaded
`remove_images`	Determine whether to remove all images on disk before system upgrade started

The following are variable definitions and default values to update internally managed switch software:

Name	Default	Type
`command_timeout`	`240`	Integer
`force`	`false`	Boolean
`reload_command`	`'"reload noconfirm"'`	String
`reload_timeout`	`200`	Integer
`remove_images`	`false`	Boolean

4.2.5 Supported Job Templates

The following subsections describe the currently supported job templates.

4.2.5.1 AWX Inventory Host Update

Create, update, or destroy one or more hosts on a specific AWX inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "AWX Inventory Host Update".

> ⚠ Make sure that all required variables described below are defined before running this job. You can define these variables either as inventory variables or as job template variables.

The following variables are required to update inventory:

Variable	Default	Type
`controller_host`	URL to the AWX controller instance	String
`controller_oauthtoken`	OAuth token for the AWX controller instance	String
`hostname`	URL to the AWX controller instance	String

Alternatively, you can specify the following variables for update inventory:

Variable	Default	Type
`controller_host`	URL to the AWX controller instance	String
`controller_username`	Username for the AWX controller instance	String
`controller_password`	Password for the AWX controller instance	String
`hostname`	Hostname or a hostname expression of the host(s) to update	String

The following variables are available to update inventory:

Variable	Description
`api_url`	URL to your cluster bring-up REST API. This variable item is required when the `hostname_regex_enabled` is set to true.
`description`	Description to use for the host(s)
`host_enabled`	Determine whether the host(s) should be enabled
`hostname_regex_enabled`	Determine whether to use hostname expression to create the hostnames
`host_state`	State of the hosts resources. Options: present; or absent.
`inventory`	Name of the inventory the host(s) should be made a member of

The following are variable definitions and default values to update inventory:

Variable	Default	Type
`api_url`	`''`	String
`description`	`''`	String
`host_enabled`	`true`	Boolean
`hostname_regex_enabled`	`true`	Boolean
`host_state`	`'present'`	String
`inventory`	`'IB Cluster Inventory'`	String

4.2.5.2 Cable Validation

Perform cable validation according to a given topology file.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "Cable Validation".

> ⚠ make sure that the filenames you provide in the `ip_files` and `topo_files` parameters, are names of files located at `/opt/nvidia/cot/cable_validation_files`.

The following variables are required to run cable validation:

Variable	Description
`api_url`	URL to your cluster bring-up REST API.
`ip_files`	List of IP filenames to use for cable validation.
`topo_files`	List of topology filenames to use for cable validation.

Alternatively, you can specify the following variables for cable validation:

Variable	Description
`remove_agents`	Specify to remove the agents from the switches once validation is complete.
`delay_time`	Time (in seconds) to wait between queries of async requests.

The following are variable definitions and default values to run cable validation:

Variable	Default	Type
`remove_agents`	true	Boolean
`delay_time`	10	Integer

The following example shows how to provide the ip_files and topo_files parameters:

```
ip_files: ['test-ip-file.ip']
topo_files: ['test-topo-file.topo']
```

In this example, the cable validation tool would expect to find the `test-ip-file.ip` and `test-topo-file.topo` files at `/opt/nvidia/cot/cable_validation_files`.

4.2.5.3 COT Python Alignment

Ensure that Python environment for the COT client is installed on one or more hosts.

> ⚠ By default, this job template is configured to run against the `ib_host_manager` group of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "COT Python Alignment".

The following variables are available for cluster orchestration Python environment installation:

Variable	Description
`cot_dir`	Target path to installation root folder
`force`	Install the package even if it is already up to date
`working_dir`	Path to the working directory on the host

The following are variable definitions and default values for cluster bring-up client installation:

Variable	Default	Type
`cot_dir`	`'/opt/nvidia/cot'`	String
`force`	`false`	Boolean
`working_dir`	`'/tmp'`	String

4.2.5.4 ClusterKit

This job runs high performance tests on the hosts of the inventory.

> ⚠ By default, this job template is configured to run against the `ib_host_manager` group of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "ClusteKit".

> ⚠ ClusterKit relies on the HPC-X package. Make sure HPC-X package is installed.

The following variables are available for running ClusterKit:

Variable	Description
`clusterkit_hostname`	Hostname expressions that represent the hostnames to run tests on
`clusterkit_options`	List of optional arguments for the tests
`clusterkit_path`	Path to the clusterkit executable script
`ib_device`	Name of the RDMA device of the port used to connect to the fabric
`inventory_group`	Name of the inventory group for the hostnames to run tests on. This variable item is not available when either the `use_hostfile` is set to false or the `clusterkit_hostname` is set.
`max_hosts`	Limit the number of hostnames. This variable item is not available when the `use_hostfile` is set to false.

Variable	Description
`use_hostfile`	Determine whether to use a file for hostnames to run tests on
`working_dir`	Path to the working directory on the host

The following are variable definitions and default values for running ClusterKit:

Variable	Default	Type
`clusterkit_hostname`	`null`	String
`clusterkit_options`	`[]`	List[String]
`clusterkit_path`	`'/opt/nvidia/hpcx/clusterkit/bin/clusterkit.sh'`	String
`ib_device`	`'mlx5_0'`	String
`inventory_group`	`all`	String
`max_hosts`	`-1`	Integer
`use_hostfile`	`true`	Boolean
`working_dir`	`'/tmp'`	String

The ClusterKit results are uploaded to the database after each run and can be accessed via the API.

The following are REST requests to retrieve ClusterKit results:

URL	Response	Method Type
`/api/performance/clusterkit/results`	Get a list of all the ClusterKit run IDs stored in the database	GET
`/api/performance/clusterkit/results/<run_id>`	Get a ClusterKit run's results based on its run ID	GET
`/api/performance/clusterkit/results/<run_id>?raw_data=true`	Get a ClusterKit run's test results as they are stored in the ClusterKit JSON output file based on its run ID. Using the query param "raw_data".	GET
`/api/performance/clusterkit/results/<run_id>?test=<test name>`	Get a specific test result of the ClusterKit run based on its run ID. Using the query param "test".	GET

Query Param	Description
`test`	Returns a specific test result of the ClusterKit run
`raw`	Returns the data as it is stored in the ClusterKit output JSON files

Examples:

```
$ curl 'http://cluster-bringup:5000/api/performance/clusterkit/results' ["20220721_152951", "20220721_151736",
"20220721_152900", "20220721_152702"]

$ curl 'http://cluster-bringup:5000/api/performance/clusterkit/results/20220721_152951?
raw_data=true&test=latency' {
  "Cluster": "Unknown",
  "User": "root",
  "Testname": "latency",
  "Date_and_Time": "2022/07/21 15:29:51",
  "JOBID": 0,
  "PPN": 28,
  "Bidirectional": "True",
  "Skip_Intra_Node": "True",
  "HCA_Tag": "Unknown",
  "Technology": "Unknown",
  "Units": "usec",
  "Nodes": {"ib-node-01": 0, "ib-node-02": 1},
  "Links": [[0, 41.885]]
}
```

4.2.5.5 Fabric Health Counters Collection

This job collects fabric counters with and without traffic based on CollectX and ClusterKit tools.

> ⚠ By default, this job template is configured to run with the `ib_host_manager` group specified in the IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "Fabric Health Counters Collection".

The following are available for running Fabric Health Counters Collection:

Variable	Description
`clusterkit_path`	Path to the ClusterKit executable script
`collection_interval`	Interval of time between counter samples in minutes
`cot_executable`	Path to the installed `cotclient` tool
`counters_output_dir`	Directory path to save counters data
`ib_device`	Name of the RDMA device of the port used to connect to the fabric
`idle_test_time`	Time to run monitor counters without traffic in minutes
`format_generate`	Formats the collection counters data with the specified type
`hpcx_dir`	Path to the HPC-X directory
`reset_counters`	Specify to reset counters before starting the counters collection
`stress_test_time`	Time to run monitor counters with traffic in minutes
`ufm_telemetry_path`	Path for the UFM Telemetry directory located in the `ib_host_manager_server`
`working_dir`	Path to the working directory on the host

The following are variable definitions and default values for the fabric health counters collection:

Variable	Default	Type
`clusterkit_path`	`'{hpcx_dir}/clusterkit/bin/clusterkit.sh'`	String
`collection_interval`	`5`	Integer
`cot_executable`	`'/opt/nvidia/cot/client/bin/cotclient'`	String
`counters_output_dir`	`'/tmp/collectx_counters_{date}_{time}/'`	String
`ib_device`	`'mlx5_0'`	String
`idle_test_time`	`30`	Integer
`format_generate`	`'basic'`	String
`hpcx_dir`	`'/opt/nvidia/hpcx'`	String
`reset_counters`	`true`	Boolean
`stress_test_time`	`30`	Integer
`ufm_telemetry_path`	`'{working_dir}/ufm_telemetry'`	String
`working_dir`	`'/tmp'`	String

4.2.5.6 IB Fabric Health Checks

This job performs diagnostics on the fabric's state based on `ibdiagnet` checks, SM files, and switch commands.

> ⚠ By default, this job template is configured to run against the `ib_host_manager` group of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "IB Fabric Health Checks".

The following variables are available for running IB Fabric Health Checks:

Variable	Description
`check_max_failure_percentage`	Max failure percentage for fabric health checks
`cot_executable`	Path to the installed `cotclient` tool
`exclude_scope`	List of node GUIDs and their ports to be excluded
`ib_device`	Name of the RDMA device of the port used to connect to the fabric
`routing_check`	Specify for routing check

Variable	Description
`sm_configuration_file`	Path for SM configuration file; supported only when the SM is running on the `ib_host_manager`
`sm_unhealthy_ports_check`	Specify for SM unhealthy ports check; supported only when the SM is running on the `ib_host_manager`
`topology_type`	Type of topology to discover
`mlnxos_switch_hostname`	Hostname expression that represents switches running MLNX-OS
`mlnxos_switch_username`	Username to authenticate against the target switches
`mlnxos_switch_password`	Password to authenticate against the target switches

The following are variable definitions and default values for the health check:

Variable	Default	Type
`check_max_failure_percentage`	`1`	Float
`cot_executable`	`'/opt/nvidia/cot/client/bin/cotclient'`	String
`exclude_scope`	`NULL`	List(String)
`ib_device`	`'mlx5_0'`	String
`routing_check`	`True`	Boolean
`sm_configuration_file`	`'/etc/opensm/opensm.conf'`	String
`sm_unhealthy_ports_check`	`false`	Boolean
`topology_type`	`'infiniband'`	String
`mlnxos_switch_hostname`	`NULL`	String
`mlnxos_switch_username`	`NULL`	String
`mlnxos_switch_password`	`NULL`	String

The following example shows how to exclude ports using the `exclude_scope` variable:

```
exclude_scope: ['0x1234@1/3', '0x1235']
```

In this example, IB Fabric Health Check runs over the fabric except on ports 1 and 3 of node GUID 0x1234 and all ports of node GUID 0x1235.

The following example shows how to configure switch variables:

```
mlnxos_switch_hostname: 'ib-switch-t[1-2],ib-switch-s1'
mlnxos_switch_username: 'admin'
mlnxos_switch_password: 'my_admin_password'
```

In this example, IB Fabric Health Check performs a check that requires switch connection over `ib-switch-t1`, `ib-switch-t2`, and `ib-switch-s1` using the username `admin` and password `my_admin_password` for the connection.

4.2.5.7 IB Network Discovery

This job discovers network topology and updates the database.

> ⚠ By default, this job template is configured to run against the `ib_host_manager` group of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "IB Network Discovery".

The following variables are required for network discovery:

Name	Default	Type
`api_url`	URL to your cluster bring-up REST API	String

> ⚠ For the network discovery to find the IPs of MLNX-OS switches, the `ufm_telemetry_path` variable is required. This feature is supported for UFM Telemetry version 1.11.0 and above.

The following variables are available for network discovery:

Variable	Description
`clear_topology`	Use to clear previous topology data.
`ufm_telemetry_path`	Path for the UFM Telemetry folder located on the `ib_host_manager_server`. Specify for using UFM Telemetry's ibdiagnet tool for the network discovery (e.g., `'/tmp/ufm_telemetry'`).
`switch_username`	Username to authenticate against MLNX-OS switches
`switch_password`	Password to authenticate against MLNX-OS switches
`cot_python_interpreter`	Path to cluster orchestration Python interpreter
`ib_device`	Name of the in-band HCA device to use (e.g., `'mlx5_0'`)
`subnet`	Name of a subnet which the topology nodes of the are member of

The following are variables definitions and default values for network discovery:

Variable	Default	Type
`clear_topology`	`false`	Boolean

Variable	Default	Type
`ufm_telemetry_path`	`NULL`	String
`cot_python_interpreter`	`'/opt/nvidia/cot/client/bin/python/'`	String
`ib_device`	`'mlx5_0'`	String
`subnet`	`'infiniband-default'`	String

4.2.5.8 UFM Telemetry Upgrade

This job installs NVIDIA® UFM® Telemetry on one or more hosts.

> ⚠ By default, this job template is configured to run against the `ib_host_manager` group of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "UFM Telemetry Upgrade".

The following variables are required for UFM Telemetry installation:

Variable	Description
`ufm_telemetry_package_url`	URL for UFM Telemetry to download

The following variables are available for UFM Telemetry installation:

Variable	Description
`working_dir`	Destination path for installing UFM Telemetry. The package will be placed in a subdirectory called `ufm_telemetry`. Default: `/tmp`.
`ufm_telemetry_checksum`	Checksum of the UFM Telemetry package to download

4.2.5.9 MLNX_OFED Upgrade

This job installs NVIDIA® MLNX_OFED driver on one or more hosts.

Refer to the official NVIDIA Linux Drivers documentation for further information.

> ⚠ By default, this job template is configured to run against the hosts of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "MLNX_OFED Upgrade".

⚠ By default, the MLNX_OFED package is downloaded from the MLNX_OFED download center. You must specify the `ofed_version` (or use its default value) and the `ofed_package_url` variables when the download center is not available.

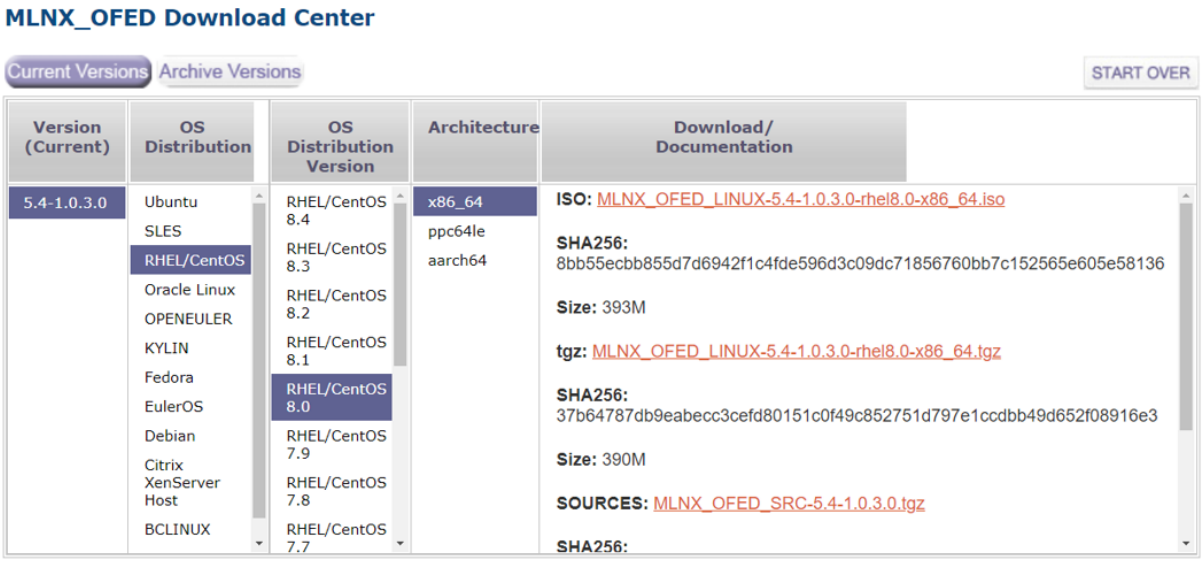
The following variables are available for MLNX_OFED installation:

Variable	Description
`force`	Install MLNX_OFED package even if it is already up to date
`ofed_checksum`	Checksum of the MLNX_OFED package to download
`ofed_dependencies`	List of all package dependencies for the MLNX_OFED package
`ofed_install_options`	List of optional arguments for the installation command
`ofed_package_url`	URL of the MLNX_OFED package to download (default: auto-detection). In addition, you must specify the `ofed_version` parameter or use its default value.
`ofed_version`	Version number of the MLNX_OFED package to install
`working_dir`	Path to the working directory on the host

The following are variable definitions and default values for MLNX_OFED installation:

Variable	Default	Type
`force`	`false`	Boolean
`ofed_checksum`	`''`	String
`ofed_dependencies`	`[]`	List
`ofed_install_options`	`[]`	List
`ofed_package_url`	`''`	String
`ofed_version`	`23.04-0.5.3.3`	String
`working_dir`	`'/tmp'`	String

The following example shows MLNX_OFED for RHEL/CentOS 8.0 on the MLNX_OFED Download Center:




```
ofed_checksum: 'SHA256: 37b64787db9eabecc3cefd80151c0f49c852751d797e1ccdbb49d652f08916e3' ofed_version:
'5.4-1.0.3.0'
```

4.2.5.10 MLNX-OS Upgrade

This job installs updates system firmware/OS software on one or more MLNX-OS switches.

> ⚠ By default, this job template is configured to run against the `ib_host_manager` group of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "MLNX-OS Upgrade".

> ⚠ Make sure all required variables described below are defined before running this job. You can define these variables either as inventory variables or as job template variables.

The following variables are required to update MLNX-OS system:

Variable	Description	Type
`mlnxos_image_url`	URL of the MLNX-OS image to download	String
`switch_username`	Username to authenticate against target switches	String
`switch_password`	Password to authenticate against target switches	String
`switches`	List of IP addresses/hostnames of the switches to upgrade	List[String]

The following variables are available to update MLNX-OS system:

Variable	Description
`command_timeout`	Time (in seconds) to wait for the command to complete
`force`	Update MLNX-OS system even if it is already up to date
`image_url`	Alias name for `mlnxos_image_url`. This variable item is not available when the `mlnxos_image_url` is set.
`reload_command`	Specify an alternative command to reload switch system
`reload_timeout`	Time (in seconds) to wait for the switch system to reload
`remove_images`	Determine whether to remove all images on disk before system upgrade starts

The following are variable definitions and default values for update MLNX-OS system:

Variable	Default	Type
`command_timeout`	`240`	Integer
`force`	`false`	Boolean
`reload_command`	`'"reload noconfirm"'`	String
`reload_timeout`	`200`	Integer
`remove_images`	`false`	Boolean

4.2.5.11 MLNX-OS Configure

This job executes configuration commands on one or more MLNX-OS switches.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "MLNX-OS Configure".

The following variables are required to configure MLNX-OS system:

Variable	Description	Type
`switch_config_commands`	List of configuration commands to execute	List[String]
`switch_username`	Username to authenticate against target switches	String
`switch_password`	Password to authenticate against target switches	String
`switches`	List of IP addresses/hostnames of the switches to configure	List[String]

The following variables are available to configure MLNX-OS system:

Variable	Description
`save_config`	Indicates to save the system configuration after the execution completed

The following are variable definitions and default values to configure MLNX-OS system:

Variable	Default	Type
`save_config`	`true`	Boolean

4.2.5.12 MFT Upgrade

This job installs NVIDIA® MFT package on one or more hosts.

Refer to the official Mellanox Firmware Tools documentation for further information.

> ⚠ By default, this job template is configured to run against the hosts of IB Cluster Inventory.

To run this job template:

1. Go to Resources > Templates.
2. Click the Launch Template button on "MFT Upgrade".

> ⚠ By default, the MFT package is downloaded from the MFT download center. You must specify the `mft_version` (or use its default value) and the `mft_package_url` variables when the download center is not available.

The following variables are available for MFT installation:

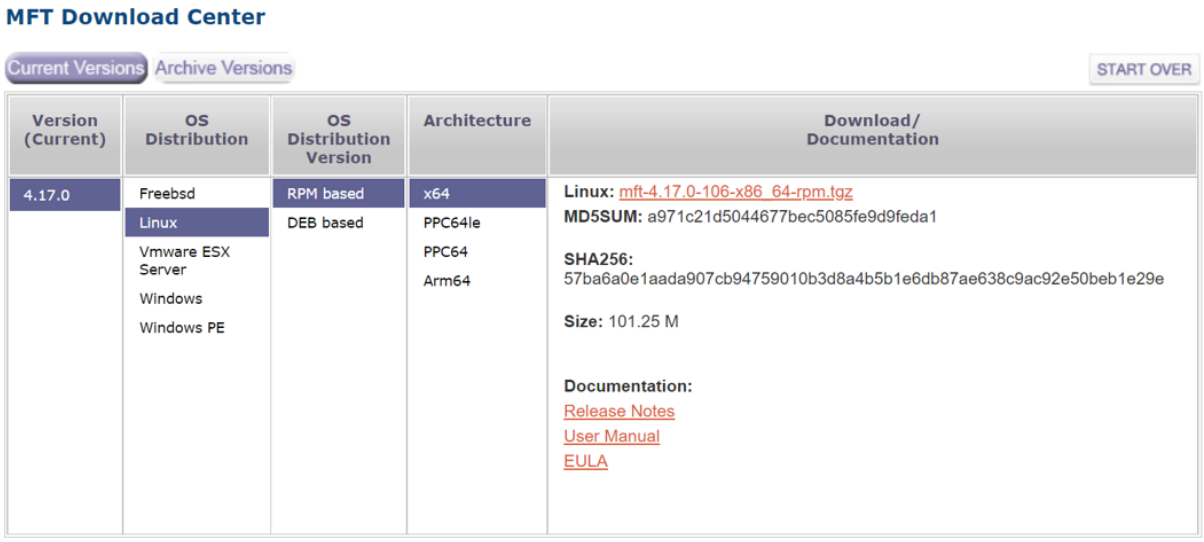
Variable	Description
`force`	Install MFT package even if it is already up to date
`mft_checksum`	Checksum of MFT package to download
`mft_dependencies`	List of all package dependencies for the MFT package
`mft_install_options`	List of optional arguments for the installation command
`mft_package_url`	URL of the MFT package to download (default: auto-detection). In addition, you must specify the `mft_version` parameter or use its default value.
`mft_version`	Version number of the MFT package to install
`working_dir`	Path to the working directory on the host

The following are variable definitions and default values for MFT installation:

Variable	Default	Type
`force`	`false`	Boolean

Variable	Default	Type
`mft_checksum`	`''`	String
`mft_dependencies`	`[]`	List
`mft_install_options`	`[]`	List
`mft_package_url`	`''`	String
`mft_version`	`'4.24.0-72'`	String
`working_dir`	`'/tmp'`	String

The following example shows MFT for RedHat on the MFT Download Center:




```
mft_checksum: 'sha256: 57ba6a0e1aada907cb94759010b3d8a4b5b1e6db87ae638c9ac92e50beb1e29e' mft_version: '4.17.0-106'
```

4.2.5.13 HPC-X Upgrade

This job installs NVIDIA® HPC-X® package on one or more hosts.

Refer to the official NVIDIA HPC-X documentation for further information.

> ⚠ By default, this job template is configured to run against the hosts of IB Cluster Inventory. You must set the `hpcx_install_once` variable to `true` when installing the HPC-X package to a shared location.

To run this job template:

1. Go to Resources > Templates.
2. Click the "Launch Template" button on "HPC-X Upgrade".

⚠ By default, the HPC-X package is downloaded from the HPC-X download center. You need to specify the `hpcx_version` (or use its default value) and the `hpcx_package_url` variables when the download center is not available.

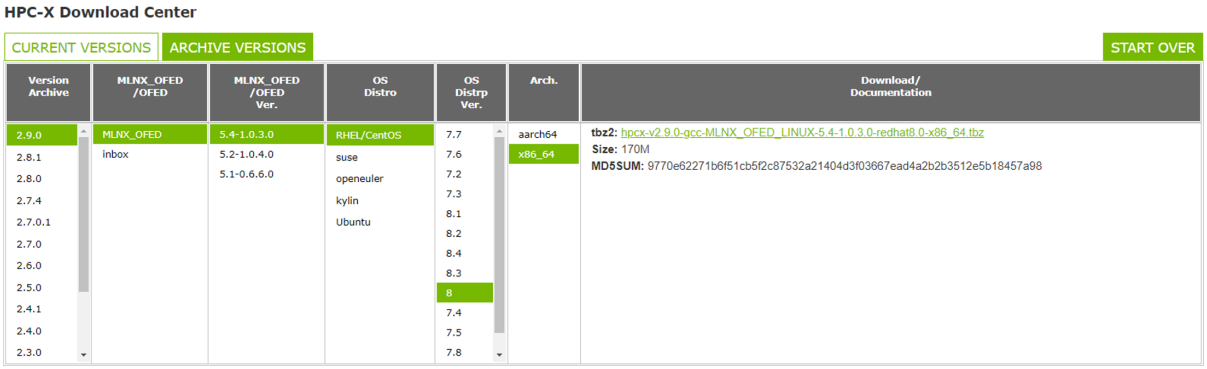
The following variables are available for HPC-X installation:

Variable	Description
`force`	Install HPC-X package even if it is already up to date
`hpcx_checksum`	Checksum of the HPC-X package to download
`hpcx_dir`	Target path for HPC-X installation folder
`hpcx_install_once`	Specify whether to install HPC-X package via single host. May be used to install the package on a shared directory.
`hpcx_package_url`	URL of the HPC-X package to download (default: auto-detection). In addition, you must specify the `hpcx_version` parameter or use its default value.
`hpcx_version`	Version number of the HPC-X package to install
`ofed_version`	Version number of the OFED package compatible to the HPC-X package. This variable item is required when MLNX_OFED is not installed on the host.
`working_dir`	Path to the working directory on the host

The following are variable definitions and default values for HPC-X installation:

Variable	Default	Type
`force`	`false`	Boolean
`hpcx_checksum`	`''`	String
`hpcx_dir`	`'/opt/nvidia/hpcx'`	String
`hpcx_install_once`	`false`	Boolean
`hpcx_package_url`	`''`	String
`hpcx_version`	`'2.15.0'`	String
`ofed_version`	`''`	String
`working_dir`	`'/tmp'`	String

The following example shows HPC-X for RedHat 8.0 on the HPC-X Download Center:




```
hpcx_checksum: 'sha256: 57ba6a0e1aada907cb94759010b3d8a4b5b1e6db87ae638c9ac92e50beb1e29e' hpcx_version: '2.9.0'
ofed_version: ''
```

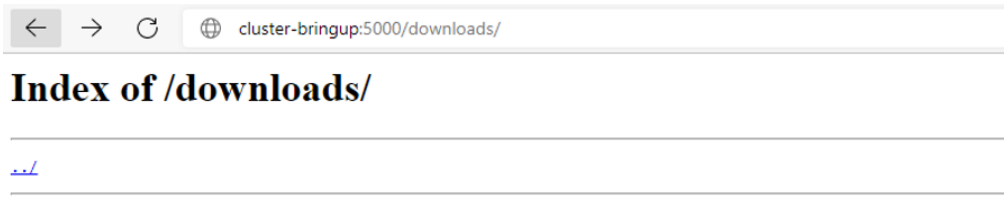
4.2.5.14 File Server

A file server is useful when you must access files (e.g., packages, images, etc.) that are not available on the WEB.

The files can be accessed over the following URL: `http://<host>:<port>/downloads/` where `host` (IP address/hostname) and `port` are the address of your cluster bring-up host.

For example, if `cluster-bringup` is the hostname of your cluster bring-up host and the TCP port is 5000 as defined in the suggested configuration, then files can be accessed over the URL `http://cluster-bringup:5000/downloads/`.

To see all available files, open your browser and navigate to `http://cluster-bringup:5000/downloads/`.



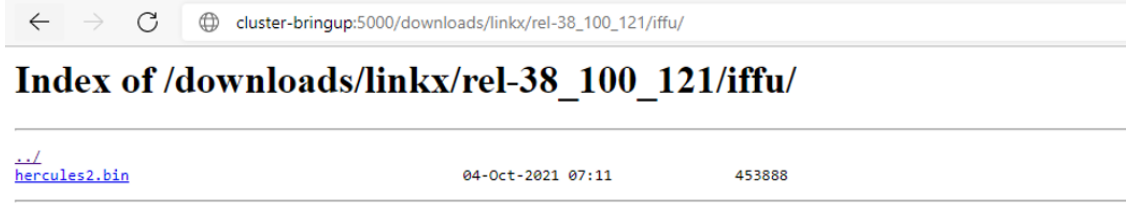
1. Create a directory for a specific cable firmware image and copy a binary image file into it. Run:

   ```
   [root@cluster-bringup ~]# mkdir -p \
   /opt/nvidia/cot/files/linkx/rel-38_100_121/iffu
   [root@cluster-bringup ~]# cp /tmp/hercules2.bin \
   /opt/nvidia/cot/files/linkx/rel-38_100_121/iffu
   ```

   The file can be accessed over the URL `http://cluster-bringup:5000/downloads/linkx/rel-38_100_121/iffu/hercules2.bin`.

2. To see all available files, open a browser and navigate to `http://cluster-bringup:5000/downloads/`.

3. To see the image file, navigate to `http://cluster-bringup:5000/downloads/linkx/rel38_100_121/iffu/`.



5 Document Revision History

5.1 Rev 3.0.0 – May 10, 2023

Added:

- Section "Installation Steps"
- Section "Upgrading Framework Script"
- Section "Export"
- Section "Import"
- Section "Cable Validation"
- Section "Fabric Health Counters Collection"

Updated:

- Section "Restoration Steps"
- Section "MLNX-OS Upgrade Variables" by adding `mlnxos_switch_hostname`
- Example code in section "IB Cluster Deployment Example"
- Section "IB Network Verification"
- Section "ClusterKit" with the `ib_device` variable
- Section "IB Cluster Health Checks" naming to "IB Fabric Health Checks"
- Section "IB Network Discovery" with `clear_topolgoy` parameter

5.2 Rev 1.2.0 – November 07, 2022

Added:

- Section "Configuration File"
- Section "COT API"
- Section "IB Cluster Health Checks"
- Section "Pass/Fair Criteria"
- Section "UFM Telemetry Upgrade Variables"

Updated:

- Section "Prerequisites" with new Kubernetes server and kubectl versions
- Section "Restoration Steps"
- Section "Installation Script" with `--config_file` option
- Section "MLNX-OS Upgrade Variables" by removing `mlnxos_switch_hostname`
- Example code in section "IB Cluster Deployment Example"
- Section "IB Network Verification"
- Section "MLNX_OFED Upgrade Variables"
- Section "MFT Upgrade Variables"
- Section "Host Package Deployment"
- Section "MLNX-OS System Alignment"
- Section "IB Network Discovery"
- Section "UFM Telemetry Upgrade"
- Section "MLNX_OFED Upgrade"
- Section "MFT Upgrade"

- Section "HPC-X Upgrade"

5.3 Rev 1.0.1 – July 27, 2022

First release

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.