



Customizing BlueField Software Deployment Using bf.cfg

Table of contents

[Changing Default Credentials for "ubuntu" User via bf.cfg](#)

[Changing UEFI Password Using bf.cfg](#)

[Changing BMC Password Using bf.cfg](#)

[Advanced Customizations During BFB Installation](#)

[bf.cfg Parameters](#)

bf.cfg is an optional configuration file which may be used to customize the software deployment process on NVIDIA® BlueField® networking platforms (DPU or SuperNIC).

Note

To update the BMC components, it is required to provide the BMC_PASSWORD using bf.cfg to the BFB/ISO installation environment.

There are different ways to pass bf.cfg along with the BFB or ISO to customize the installation procedure:

- With BFB from the host:

```
# bfb-install -r <rshim device> -c <path to bf.cfg> -b <BFB>
```

- Using cat command:

```
# cat <BFB> <path to bf.cfg> > /dev/<rshim device>/boot
```

- By appending bf.cfg to the BFB and push it to RShim device on a host or BMC:

```
# cat <BFB> <path to bf.cfg> > <new BFB>
```

- In PXE environment using bfks parameter to provide a script that will be downloaded by the installation process and run on the Bluefield side at the beginning of installation:

```
cat > /etc/bf.cfg << 'EOF'  
BMC_PASSWORD="..."
```

EOF

- Or using `autoinstall.yaml`. See "[Deploying BlueField Software Using ISO with PXE](#)" for details.

Changing Default Credentials for "ubuntu" User via `bf.cfg`

Info

For a comprehensive list of the supported parameters to customize `bf.cfg` during BFB installation, refer to section "[bf.cfg Parameters](#)".

Ubuntu users are prompted to change the default password (`ubuntu`) for the default user (`ubuntu`) upon first login. Logging in will not be possible even if the login prompt appears until all services are up ("`DPU is ready`" message appears in `/dev/rshim0/misc`).

Note

Attempting to log in before all services are up prints the following message: `Permission denied, please try again.`

Alternatively, Ubuntu users can provide a unique password that will be applied at the end of the BFB installation. This password must be defined in a `bf.cfg` configuration file. To set the password for the `ubuntu` user:

1. Create password hash. Run:

```
# openssl passwd -1  
Password:  
Verifying - Password:
```

```
$1$3B0RlrfX$TIHry93NFUJzg3Nya00rE1
```

2. Add the password hash in quotes to the bf.cfg file:

```
# vim bf.cfg
ubuntu_PASSWORD='$1$3B0RlrfX$TIHry93NFUJzg3Nya00rE1'
```

The bf.cfg file is used with the bfb-install script in the steps that follow.

Changing UEFI Password Using bf.cfg

To change UEFI password add current UEFI password `UEFI_PASSWORD` and the new UEFI password `NEW_UEFI_PASSWORD` to bf.cfg.

Changing BMC Password Using bf.cfg

To change BMC root password, add current BMC root password `BMC_PASSWORD` and the new BMC root password `NEW_BMC_PASSWORD` to bf.cfg.

Advanced Customizations During BFB Installation

Using special purpose configuration parameters in the bf.cfg file, the BlueField's boot options and OS can be further customized. For a full list of the supported parameters to customize your BlueField during BFB installation, refer to section "[bf.cfg Parameters](#)". In addition, the bf.cfg file offers further control on customization of BlueField OS installation and software configuration through scripting.

Add any of the following functions to the bf.cfg file for them to be called by the install.sh script embedded in the BFB:

- `bfb_modify_os` – called after the file system is extracted on the target partitions. It can be used to modify files or create new files on the target file system mounted under `/mnt`. So the file path should look as follows: `/mnt/<expected_path_on_target_OS>`. This can be used to run a specific tool from the target OS (remember to add `/mnt` to the path for the tool).

- bfb_pre_install – called before eMMC/SSD partitions format and OS filesystem is extracted
- bfb_post_install – called as a last step before reboot. All eMMC/SSD partitions are unmounted at this stage.

For example, the bf.cfg script below disables OVS bridge creation upon boot:

```
# cat /root/bf.cfg

bfb_modify_os()
{
    log ===== bfb_modify_os =====
    log "Disable OVS bridges creation upon boot"
    sed -i -r -e 's/(CREATE_OVS_BRIDGES=).*\1"no"/ /mnt/etc/mellanox/mlnx-ovs.conf
}

bfb_pre_install()
{
    log ===== bfb_pre_install =====
}

bfb_post_install()
{
    log ===== bfb_post_install =====
}
```

bf.cfg Parameters

The following is a comprehensive list of the supported parameters to customize the bf.cfg file for BFB installation:

```
#####
# Configuration which can also be set in
# UEFI->Device Manager->System Configuration
#####
# Enable SMMU in ACPI.
#SYS_ENABLE_SMMU = TRUE
```

```

# Enable I2CO in ACPI.
#SYS_ENABLE_I2CO = FALSE

# Disable SPMI in ACPI.
#SYS_DISABLE_SPMI = FALSE

# Enable the second eMMC card which is only available on the BlueField Reference Platform.
#SYS_ENABLE_2ND_EMMC = FALSE

# Enable eMMC boot partition protection.
#SYS_BOOT_PROTECT = FALSE

# Enable SPCR table in ACPI.
#SYS_ENABLE_SPCR = FALSE

# Disable PCIe in ACPI.
#SYS_DISABLE_PCIE = FALSE

# Enable OP-TEE in ACPI.
#SYS_ENABLE_OPTEE = FALSE

#####
# Boot Order configuration
# Each entry BOOT<N> could have the following format:
# PXE:
# BOOT<N> = NET-<NIC_P0 | NIC_P1 | OOB | RSHIM>-<IPV4 | IPV6>
# PXE over VLAN (vlan-id in decimal):
# BOOT<N> = NET-<NIC_P0 | NIC_P1 | OOB | RSHIM>[.<vlan-id>]-<IPV4 | IPV6>
# UEFI Shell:
# BOOT<N> = UEFI_SHELL
# DISK: boot entries created during OS installation.
# BOOT<N> = DISK
#####
# This example configures PXE boot over the 2nd ConnectX port.
# If fails, it continues to boot from disk with boot entries created during OS
# installation.
#BOOT0 = NET-NIC_P1-IPV4
#BOOT1 = DISK

# UPDATE_ATF_UEFI - Updated ATF/UEFI (Default: yes)
# Relevant for PXE installation only as while using RSHIM interface ATF/UEFI
# will always be updated using capsule method
UPDATE_ATF_UEFI="yes"

```

```

# To change UEFI password set UEFI_PASSWORD to its current value and NEW_UEFI_PASSWORD to the
new UEFI password (clear text).
UEFI_PASSWORD=<current UEFI password>
NEW_UEFI_PASSWORD=<new UEFI password>

# UPDATE_DPU_OS - Update/Install BlueField Operating System (Default: yes)
UPDATE_DPU_OS="yes"

# grub_admin_PASSWORD - Hashed password to be set for the "admin" user to enter Grub menu
# Relevant for Ubuntu BFB only. (Default: is not set)
# E.g.:
grub_admin_PASSWORD='grub.pbkdf2.sha512.10000.5EB1FF92FDD89BDAF3395174282C77430656A6DBE
grub_admin_PASSWORD='grub.pbkdf2.sha512.10000.<hashed password>'

# ubuntu_PASSWORD - Hashed password to be set for "ubuntu" user during BFB installation process.
# Relevant for Ubuntu BFB only. (Default: is not set)
ubuntu_PASSWORD=<hashed password>

#####
# BMC Component Update
#####
# BMC_USER - User name to be used to access BMC (Default: root)
BMC_USER="root"

# BMC_PASSWORD - Password used by the BMC user to access BMC (Default: None)
BMC_PASSWORD=""

# NEW_BMC_PASSWORD - can be used to change BMC_PASSWORD to the new one (Default: None)
# Note: current BMC_PASSWORD is required
NEW_BMC_PASSWORD=<new BMC password>

# BMC_IP_TIMEOUT - Maximum time in seconds to wait for the connection to the
# BMC to be established (Default: 600)
BMC_IP_TIMEOUT=600

# BMC_TASK_TIMEOUT - Maximum time in seconds to wait for BMC task (BMC/CEC
# Firmware update) to complete (Default: 1800)
BMC_TASK_TIMEOUT=1800

# UPDATE_BMC_FW - Update BMC firmware (Default: yes)
UPDATE_BMC_FW="yes"

# BMC_REBOOT - Reboot BMC after BMC firmware update to apply the new version
# (Default: no). Note that the BMC reboot will reset the BMC console.

```



```

BMC_REBOOT="no"

# UPDATE_CEC_FW - Update CEC firmware (Default: yes)
UPDATE_CEC_FW="yes"

# UPDATE_DPU_GOLDEN_IMAGE - Update BlueField Golden Image (Default: yes)
UPDATE_DPU_GOLDEN_IMAGE="yes"

# UPDATE_NIC_FW_GOLDEN_IMAGE- Update NIC firmware Golden Image (Default: yes)
UPDATE_NIC_FW_GOLDEN_IMAGE="yes"

# pre_bmc_components_update - Shell function called by BFB's install.sh before
# updating BMC components (no communication to the BMC is established at this
# point)

# post_bmc_components_update - Shell function called by BFB's install.sh after
# updating BMC components

#####
# NIC Firmware update
#####
# WITH_NIC_FW_UPDATE - Update NIC Firmware (Default: yes)
WITH_NIC_FW_UPDATE="yes"

#####
# Other misc configuration
#####

# MAC address of the rshim network interface (tmfifo_net0).
#NET_RSHIM_MAC = 00:1a:ca:ff:ff:01

# DHCP class identifier for PXE (arbitrary string up to 32 characters)
#PXE_DHCP_CLASS_ID = NVIDIA/BF/PXE

# Create dual boot partition scheme (Ubuntu only)
# DUAL_BOOT=yes

# Upgrade NIC firmware
# WITH_NIC_FW_UPDATE=yes

# Target storage device for the BlueField Arm OS (Default SSD: /dev/nvme0n1)
device=/dev/nvme0n1

# bfb_modify_os - SHELL function called after the file system is extracted on the target partitions.

```

```
# It can be used to modify files or create new files on the target file system mounted under  
# /mnt. So the file path should look as follows: /mnt/<expected_path_on_target_OS>. This  
# can be used to run a specific tool from the target OS (remember to add /mnt to the path for  
# the tool).
```

```
# bfb_pre_install - SHELL function called before partitions format  
# and OS filesystem is extracted
```

```
# bfb_post_install - SHELL function called as a last step before reboot.  
# All partitions are unmounted at this stage.
```

© Copyright 2024, NVIDIA. PDF Generated on 08/20/2024