Hewlett Packard
Enterprise

HPE Cray XD675 Ansible Firmware Update Tool User Guide

Part Number: 10-150146-Q324

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Table of Contents

Overview

The HPE Cray XD Ansible Firmware Update Tool (AFUT) provides a mechanism to quickly update the firmware components of HPE Cray XD Server nodes, whether individually or many at a time. It has support for the XD675 models. This Ansible based tool can be executed from Linux either from management or administrator nodes to update the components. This tool can also be used to create firmware inventory reports for HPC cluster nodes. It is necessary to install a full Ansible environment and other prerequisites on the management workstation as it is an Ansible script-based tool.

Supported operating system

The tool supports execution from Linux operating system.

1. Ubuntu (24.04)
2. Red Hat Enterprise Linux (RHEL 8, and RHEL 9)
3. SLES Linux Enterprise Server 15 SP6

Prerequisites

1. Ensure that Ansible is installed.
2. The Ansible collection community.general version 8.2.0 or later must be installed.
3. To install the package, use:
   i. sudo apt install ansible
   ii. ansible-galaxy collection install community.general

To install all the remaining prerequisites, use the setup.yml file based on operating system

4. Ensure request toolbelt version 0.9.1 or later is installed.
   i. To install, run pip install requests-toolbelt.

Supported Models for Update

a. HPE Cray XD675 – Wistron using ASPEED/AMI BMC firmware

Supported Targets for Update

- HPE Cray XD675
  - BMC
  - BIOS
  - HPM CPLD
  - DC-SCM CPLD
  - HIB CPLD
  - GPU

Install AFUT

1. You can download AFUT by cloning this repository: https://github.com/Hewlettpackard/CrayXD_AFUT_XD675
2. Install and unzip the Ansible_Firmware_Update_Tool.zip package at the Management Workstation where you want to deploy the firmware.
3. The unzipped folder contains:
    - `system_firmware_update.yml` – The main Ansible playbook file, consisting of tasks and which is responsible for flashing firmware.
    - `get_system_firmware_inventory.yml` – The Ansible playbook file, which is responsible for report generation with firmware inventory information.
    - `get_gpu_inventory.yml` – The Ansible playbook file, which is responsible for report generation with GPU inventory information.
    - `power_state_XD675.yml` – The Ansible playbook is used to fetch and change the power states of the HPE Cray XD675.
    - `Inventory` – Create an Ansible inventory file that lists the target systems you want to update or retrieve inventory details. This file should contain the IP addresses to connect to the systems.
    - `system_credentials.yml` –This file contains the credentials of the target systems.
    - `config.ini` – Configuration file contains four sections Target, Image Path, Power and Firmware Type. Under these sections, options need to be filled with valid keys.
    - `Setup playbook` – This playbook can be run to install all pre-requisites on the host system.

        i) `ubuntu_setup.yml` - This playbook is used to install pre-requisites on the ubuntu system. Use the following command to run this playbook.

        ```
        $ansible-playbook ubuntu_setup.yml --ask-become-pass -i inventory
        ```

        ii) `RHEL_setup.yml` – This playbook is used to install pre-requisites on the RHEL system. Use the following command to run this playbook.

        ```
        $ansible-playbook RHEL_setup.yml --ask-become-pass -i inventory
        ```

        iii) `SUSE_setup.yml` – This playbook is used to install pre-requisites on the SUSE system. Use the following command to run this playbook.

        ```
        $ansible-playbook SUSE_setup.yml --ask-become-pass -i inventory
        ```

Notes

- Enter the correct details in config.ini file to start update. If wrong information is given, AFUT might not be able to detect the firmware file of the update or might print the error message on the command line.
- For HPE Cray XD675, power state must be off for BIOS before the firmware update. The power_state_XD675.yml playbook can be used to switch the power state to off. BIOS also requires reboot after firmware update, using Redfish APIs.
- For the three CPLD firmware updates (HPM CPLD, DC-SCM CPLD and HIB CPLD), the power state must be turned off after completion of the update and before resetting the CPLD component. After CPLD reset, the power state must be turned on.
- After GPU update, perform an AC cycle through GUI. Go to GPU Management -> Firmware -> Perform AC cycle for GPU -> Click Start. AC cycle after GPU update can be done through both GUI as well as manual plug-out and plug-in of power cables.

- IPV6 addresses are not supported with the current AFUT. IP addresses refer to IPv4 only and not IPv6.
- Check the generated CSV output file to confirm the firmware update status.
- AFUT has no explicit limitation on the number of nodes it can handle. This is not specified in the official Ansible documentation, as you can verify here: Ansible Documentation."

Firmware files

Download the latest component firmware packs from the HPE Support Centre.

Supported files for the update:

- Use .bin files for BMC update and .rom files for BIOS update.
- CPLD firmware updates requires .rpd files for the update.
- For GPU updates, use .pldm files.

Ensure that the file is present on the local machine and provide the correct file path in the configuration file for execution.

Configuration

- `system_credentials.yml` contains the credentials of the server in the form of its username and password.

  An example of `system_credentials.yml` is as follows:

```
---
inputs:
10.xx.xx.xx:
user: "username"
password: "password"
10.xx.xx.xx:
user: "username"
password: "password"
10.xx.xx.xx:
user: "username"
password: "password"
```

- `inventory.txt` file contains the details of the server's IP, mention all the IP's that needs to be used for execution. If an IP is marked with "#", then it will not be considered for execution.

  An example of `inventory` is as follows:

```
[xds]
#IP1
IP2
IP3
```

- `config.ini` – This configuration file maintains the Target, Image path and Options (for power state) for the firmware update.

Following is an example of the configuration file:

```
$ cat config.ini
[Image]
update_image_path_xd675 =


[Target]
update_target =
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.


[Options]
power_state =
##Allowed options are: NA on off
##applicable only for XD675
```

Ansible Vault Implementation

Ansible-Vault helps to encrypt files using a password, here we encrypt our system credentials file using a password to make it more secure.

Following are examples for using an Ansible vault

```
cat system_credentials.yml
---
  inputs:
  10.xx.xx.xx:
    user: "username"
    password: "password"
  10.xx.xx.xx:
    user: "username"
    password: "password"
  10.xx.xx.xx:
    user: "username"
    password: "password"


ansible-vault encrypt system_credentials.yml

New Vault password:
Confirm New Vault password:
Encryption successful
```

- cat system_credentials.yml

```
$ANSIBLE_VAULT;1.1;AES256
393134316238356433306239353962326237623662373864623136316565323430313435396666564
333639636136346230336335653831393535396330326534 0a373563656536343136393731616531

663139396330356637343465373033323930333963643165376132623837636539643834376133 63
386461653731656361 0a623633326564653537323630373065363436303337623765666162333339
376232643766376332623038393836396634393831613534353335326530613637373133623033 31
666339623162383362393735346232306331643633383332393731623165303839346630356437 35
393236626334323939326137303931356534306365636531393162653061383334663530656366 31
636232316537383633343338313635326164333961313936633035643364333338656532633466 38
643132353434376536633765356266616536663732653662616338376139663231643531656337 30
653539373237363530313132373538633663326237326363343666313039616134313034306264 37
386434336661363330653430631623039383232363562623861376536373639396163643038 31
616639396234306133363465626531336162323732323737343830316437343232643531333966 35
626431613936386234306635663039393265363334633834373834363832383665633964303234 35
623938646339643233666534616466623962383764373836373 6
```

- ansible-vault edit system_credentials.yml

```
Vault password:
```

- ansible-vault view system_credentials.yml

```
Vault password:
- --
inputs:
  10.xx.xx.xx:
    user: "username"
    password: "password"

 10.xx.xx.xx:
    user: "username"
    password: "password"
  10.xx.xx.xx:
    user: "username"
    password: "password"
```

- cat system_credentials.yml

```
$ANSIBLE_VAULT;1.1;AES256
393134316238356433306239353962326237623662373864623136316565323430313435396666564
333639636136346230336335653831393535396330326534 0a373563656536343136393731616531
663139396330356637343465373033323930333963643165376132623837636539643834376133 63
386461653731656361 0a623633326564653537323630373065363436303337623765666162333339
376232643766376332623038393836396634393831613534353335326530613637373133623033 31
666339623162383362393735346232306331643633383332393731623165303839346630356437 35
393236626334323939326137303931356534306365636531393162653061383334663530656366 31
636232316537383633343338313635326164333961313936633035643364333338656532633466 38
643132353434376536633765356266616536663732653662616338376139663231643531656337 30
653539373237363530313132373538633663326237326363343666313039616134313034306264 37
386434336661363330653430631623039383232363562623861376536373639396163643038 31
616639396234306133363465626531336162323732323737343830316437343232643531333966 35
626431613936386234306635663039393265363334633834373834363832383665633964303234 35
623938646339643233666534616466623962383764373836373 6
```

- ansible-vault decrypt system_credentials.yml

```
Vault password:
Decryption successful
```

- `cat system_credentials.yml`

```
---

inputs:
  10.xx.xx.xx:
    user: "username"
    password: "password"
  10.xx.xx.xx:
    user: "username"
    password: "password"
  10.xx.xx.xx:
    user: "username"
    password: "password"
```

Below is an example for usage of encrypted file in command execution. “–ask-vault-pass” needs to be used while execution of command if any of the file is encrypted. Vault password is needed during the execution.

```
$ ansible-playbook get_system_firmware_inventory.yml -i inventory --ask-vault-pass -e
  @system_credentials.yml

  Vault password:
```

Below is a similar example for “`system_firmware_update.yml`”, while it’s execution with encrypted “`system_credentials.yml`”.

```
$ ansible-playbook system_firmware_update.yml -i inventory --ask-vault-pass -e
  @system_credentials.yml

  Vault password:
```

Commands

usage: ansible-playbook: [-e EXTRA_VARS] [-i INVENTORY] [--ask-vault-password]

options:
 --ask-vault-password, --ask-vault-pass
 ask for vault password
-e EXTRA_VARS, --extra-vars EXTRA_VARS
 set additional variables as key=value or YAML/JSON, if filename prepend with @
-i inventory, --inventory inventory, --inventory-file inventory
 specify inventory host path or comma separated host list. --inventory-file is deprecated

Running the Utility for Report

The “`get_system_firmware_inventory.yml`” playbook generates a report consisting of the IP details, Models and the components version in a CSV file.
Before execution, ensure that the system credentials and the inventory details are correctly filled in the respective files.

Following command is used for the execution of Inventory file:

```
$ ansible-playbook -i inventory get_system_firmware_inventory.yml -e @system_credentials.yml

PLAY [version 1.0 Fetches the AFUT supported Cray XD servers System Firmware Inventory Details
along with Model name] *********************************************************************

TASK [Gathering Facts]
*********************************************************************************************
*************************************************************

ok: [10.xx.xx.xx]
```

```
TASK [All System Firmware Inventory Details will be stored in the below csv file]
*********************************************************************************************
*********

ok: [10.xx.xx.xx]

TASK [Fetching System Firmware Inventory Details]
*********************************************************************************************
****************************************

[WARNING]: Collection community.general does not support Ansible version 2.10.8

ok: [10.xx.xx.xx]

TASK [Writing inventory details to All_System_FW_Inventory_2024-11-12_10:05:22.csv file]
*********************************************************************************************
**

changed: [10.xx.xx.xx]

PLAY RECAP
*********************************************************************************************
*************************************************************************

10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0

rescued=0    ignored=0
```

Following is an example output of a CSV file:

```
$ cat  All_System_FW_Inventory_2024-11-12_10:05:22.csv

IP_Address,Model,BMC,BIOS,DCSCM_FPGA,MB_FPGA,HIB_FPGA,RoT

10.xx.xx.xx,HPE Cray XD675,oct-09-24-2.36.0.1,v3.01.00,v1.06,v1.04,v2.18,v1.00
```

The "get_gpu_inventory.yml" playbook generates a report consisting of the IP details, Models and the components version in a CSV file.

Following command is used for the execution of GPU Inventory file:

```
$ ansible-playbook -i inventory get_gpu_inventory.yml -e @system_credentials.yml

PLAY [version 1.0 Fetches the AFUT supported Cray XD servers GPU Inventory Details along with

Model name] *********************************************************************

TASK [Gathering Facts]

*********************************************************************************************

*****************************************************************

ok: [10.xx.xx.xx]

TASK [All GPU Inventory Details will be stored in the below csv file]

*********************************************************************************************

*********

ok: [10.xx.xx.xx]
```

```
TASK [Fetching GPU Inventory Details]
*****************************************************************************************************
************************************
[WARNING]: Collection community.general does not support Ansible version 2.10.8
ok: [10.xx.xx.xx]
TASK [Writing GPU inventory details to GPU_FW_Inventory_2024-12-12_15:05:39.csv file]
*****************************************************************************************************
*********
changed: [10.xx.xx.xx]
PLAY RECAP
*****************************************************************************************************
*************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
$ cat  GPU_FW_Inventory_2024-12-12_15:05:39.csv
IP_Address,Model,AMC_ACTIVE,AMC_FPGA_ACTIVE,APCFG_ACTIVE,BUNDLE_ACTIVE,IFWI_ACTIVE,RETIMER_ACTIVE,
RMI_ACTIVE,ROT_ACTIVE,UBB_FPGA_ACTIVE,VR_BUNDLE_ACTIVE
10.14.56.170,HPE Cray
XD675,2.11.0.65.0F,00.26.10,1,01.24.12.10,01.04.114328,2.11.075,4.0.12,0x00730000,00.26.24,{OAM_0:
01.009, OAM_1: 01.009, OAM_2: 01.009, OAM_3: 01.009, OAM_4: 01.009, OAM_5: 01.009, OAM_6: 02.0BE,
OAM_7: 01.009, UBB: 01.04}
```

Running the Utility for Power State

To check the current power state, the power_state in the configuration file should be 'NA', then to change the power state use 'on' or 'off' in the power state section of the configuration file.

Following command is used for the execution of power state file:

```
$ ansible-playbook -i inventory power_state_XD675.yml -e @system_credentials.yml
PLAY [version 1.0 Get Power State of HPE Cray XD675 model nodes]
*****************************************************************************************************
*************************
TASK [Gathering Facts]
*****************************************************************************************************
*****************************************************************
ok: [10.xx.xx.xx]
TASK [Power states of the nodes will be uploaded to the below csv file]
*****************************************************************************************************
******************
ok: [10.xx.xx.xx]
TASK [Getting Power State of Cray XD675 Server nodes]
*****************************************************************************************************
**********************************
[WARNING]: Collection community.general does not support Ansible version 2.10.8
ok: [10.xx.xx.xx]
```

```
TASK [Writing Power status details to Power_State_CrayXD675_2024-12-02_18:03:54.csv file]
*********************************************************************************************
*
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************
*************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

$ cat Power_State_CrayXD675_2024-12-02_18:03:54.csv
IP_Address,Model,Power_State
10.xx.xx.xx,HPE Cray XD675,On

$ cat config.ini
[Image]
update_image_path_xd675 =

[Target]
update_target =
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.

[Options]
power_state = off
##Allowed options are: NA on off
##applicable only for XD675


$ ansible-playbook -i inventory power_state_XD675.yml -e @system_credentials.yml
PLAY [version 1.0 Get Power State of HPE Cray XD675 model nodes]
*********************************************************************************************
*************************
TASK [Gathering Facts]
*********************************************************************************************
*****************************************************************
ok: [10.xx.xx.xx]
TASK [Power states of the nodes will be uploaded to the below csv file]
*********************************************************************************************
******************
ok: [10.xx.xx.xx]
TASK [Getting Power State of Cray XD675 Server nodes]
*********************************************************************************************
***********************************
[WARNING]: Collection community.general does not support Ansible version 2.10.8
ok: [10.xx.xx.xx]
TASK [Writing Power status details to Power_State_CrayXD675_2024-12-02_17:59:59.csv file]
*********************************************************************************************
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************
*************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

```
$ cat Power_State_CrayXD675_2024-12-02_17:59:59.csv
IP_Address,Model,Power_State
10.xx.xx.xx,HPE Cray XD675,Off
```

Running the Utility for Firmware Update

The "`system_firmware_update.yml`" playbook is used to update HPC nodes. The available update targets depend on the server model, and the supported targets for each model can be found in the supported targets section.

Before performing the update:

- Specify the image path, target and firmware file type details in the playbook.
- Update the inventory and system credentials input files with the respective details of target system.

After the update:

- The status of the update tasks will be saved in a CSV file in the current directory.

Here is an example of the update for HPE Cray XD675 BMC from version sep-30-24-2.35.0.3 to oct-09-24-2.36.0.1

```
$ cat config.ini
[Image]
update_image_path_xd675 = bmc_2_36_0_1_scap.bin

[Target]
update_target = bmc
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.

[Options]
power_state = NA
##Allowed options are: NA on off
##applicable only for XD675


$ ansible-playbook -i inventory system_firmware_update.yml -e @system_credentials.yml

PLAY [version 1.0 System Firmware Update for HPE Cray675 model systems]
**********************************************************************************************
****************************

TASK [Gathering Facts]
**********************************************************************************************
**********************************************************************************************
******************
ok: [10.xx.xx.xx]

TASK [System Firmware Update Status result will be uploaded to the below csv file]
**********************************************************************************************
*****************************************************

TASK [Running Firmware Update for Cray XD Servers]
**********************************************************************************************
*****************************************************************************
ok: [10.xx.xx.xx]

TASK [Writing Firmware Upgrade status details to System_FW_Update_2024-11-22_03:37:18.csv file]
**********************************************************************************************
****************************************
changed: [10.xx.xx.xx]
```

```
PLAY RECAP
*********************************************************************************************************
*********************************************************************************************************
*****************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Following is an example output of a CSV file:

```
$ cat System_FW_Update_2024-11-22_03:37:18.csv

IP_Address,Model,bmc_Pre_Ver,bmc_Post_Ver,Update_Status
10.xx.xx.xx,HPE Cray XD675,sep-30-24-2.35.0.3,oct-09-24-2.36.0.1,success
```

Here is an example of update for Cray XD675 BIOS from version 3.01.01 to 3.1.0

```
$ cat config.ini
[Image]
update_image_path_xd675 = DS_v310_signed.rom

[Target]
update_target = bios
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.

[Options]
power_state = NA
##Allowed options are: NA on off
##applicable only for XD675


$ ansible-playbook system_firmware_update.yml -i inventory -e @system_credentials.yml
PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*****************************************************************
TASK [Gathering Facts]
*********************************************************************************************************
*********************************************************************************
ok: [10.xx.xx.xx]
TASK [System Firmware Update Status result will be uploaded to the below csv file]
*********************************************************************************************************
*****************************
ok: [10.xx.xx.xx]
TASK [Running Firmware Update for Cray XD Servers]
*********************************************************************************************************
**************************************************
ok: [10.xx.xx.xx]
TASK [Writing Firmware Upgrade status details to System_FW_Update_2024-11-26_10:36:29.csv file]
*********************************************************************************************************
*
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************************
*********************************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Following is an example output of a CSV file:

```
$ cat  System_FW_Update_2024-11-26_10:36:29.csv
IP_Address,Model,bios_Pre_Ver,bios_Post_Ver,Update_Status
10.xx.xx.xx,HPE Cray XD675,v3.01.01,v3.1.0,success
```

Here is an example of the force update for HPE Cray XD675 HPM/MB CPLD

```
$ cat config.ini
[Image]
update_image_path_xd675 = MiramarHPM_FPGA_V0201.rpd

[Target]
update_target = mb_fpga
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.

[Options]
power_state = on
##Allowed options are: NA on off
##applicable only for XD675


$ ansible-playbook power_state_XD675.yml -i inventory -e @system_credentials.yml
PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*********************************************************************
TASK [Gathering Facts]
*****************************************************************************************************
*****************************************************************************
ok: [10.xx.xx.xx]
TASK [Power states of the nodes will be uploaded to the below csv file]
*****************************************************************************************************
*********************************
ok: [10.xx.xx.xx]
TASK [Getting Power State of Cray XD675 Server nodes]
*****************************************************************************************************
*************************************************
ok: [10.xx.xx.xx]
TASK [Writing Power status details to Power_State_CrayXD675_2024-11-29_15:55:20.csv file]
*****************************************************************************************************
*****
changed: [10.xx.xx.xx]
PLAY RECAP
*****************************************************************************************************
*****************************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

$ cat Power_State_CrayXD675_2024-11-29_15:55:20.csv
IP_Address,Model,Power_State
10.xx.xx.xx,HPE Cray XD675,On


$ ansible-playbook -i inventory system_firmware_update.yml -e @system_credentials.yml

PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*********************************************************************
TASK [Gathering Facts]
*****************************************************************************************************
*****************************************************************************
ok: [10.xx.xx.xx]
TASK [System Firmware Update Status result will be uploaded to the below csv file]
*****************************************************************************************************
**********************
ok: [10.xx.xx.xx]
TASK [Running Firmware Update for Cray XD Servers]
*****************************************************************************************************
*************************************************
ok: [10.xx.xx.xx]
```

```
TASK [Writing Firmware Upgrade status details to System_FW_Update_2024-11-29_15:59:24.csv file]
*********************************************************************************************
**********
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************
*********************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0


$ cat  System_FW_Update_2024-11-29_15:59:24.csv
IP_Address,Model,Update_Status,Remarks
10.xx.xx.xx,HPE Cray XD675,success,MB_FPGA firmware update is successfully completed.
```

Here is an example of the update for HPE Cray XD675 DC-SCM CPLD

```
$ cat config.ini
[Image]
update_image_path_xd675 = MiramarSCM_FPGA_V0201.rpd

[Target]
update_target = dcscm_fpga
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.

[Options]
power_state = on
##Allowed options are: NA on off
##applicable only for XD675


$ ansible-playbook power_state_XD675.yml -i inventory -e @system_credentials.yml
PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*********************************************************************
TASK [Gathering Facts]
*********************************************************************************************
*********************************************************************************
ok: [10.xx.xx.xx]
TASK [Power states of the nodes will be uploaded to the below csv file]
*********************************************************************************************
*********************************
ok: [10.xx.xx.xx]
TASK [Getting Power State of Cray XD675 Server nodes]
*********************************************************************************************
************************************************
ok: [10.xx.xx.xx]
TASK [Writing Power status details to Power_State_CrayXD675_2024-11-29_16:46:17.csv file]
*********************************************************************************************
*****
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************
*********************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

$ cat Power_State_CrayXD675_2024-11-29_16:46:17.csv
IP_Address,Model,Power_State
10.xx.xx.xx,HPE Cray XD675,On
```

```
$ ansible-playbook -i inventory system_firmware_update.yml -e @system_credentials.yml

PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*********************************************************************
TASK [Gathering Facts]
*****************************************************************************************************
*************************************************************************
ok: [10.xx.xx.xx]
TASK [System Firmware Update Status result will be uploaded to the below csv file]
*****************************************************************************************************
***********************
ok: [10.xx.xx.xx]
TASK [Running Firmware Update for Cray XD Servers]
*****************************************************************************************************
*******************************************************
ok: [10.xx.xx.xx]
TASK [Writing Firmware Upgrade status details to System_FW_Update_2024-11-29_16:31:02.csv file]
*****************************************************************************************************
**********
changed: [10.xx.xx.xx]
PLAY RECAP
*****************************************************************************************************
*************************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0


$ cat  System_FW_Update_2024-11-29_16:31:02.csv
IP_Address,Model,Update_Status,Remarks
10.xx.xx.xx,HPE Cray XD675,success,DCSCM_FPGA firmware update is successfully completed.
```

Here is an example of the update for HPE Cray XD675 HIB CPLD

```
$ cat config.ini
[Image]
update_image_path_xd675 = MiramarHIB_CPLD_PVT_0222_cfm0_auto.rpd

[Target]
update_target = hib_fpga
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.

[Options]
power_state = on
##Allowed options are: NA on off
##applicable only for XD675


$ ansible-playbook power_state_XD675.yml -i inventory -e @system_credentials.yml
PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*********************************************************************
TASK [Gathering Facts]
*****************************************************************************************************
*************************************************************************
ok: [10.xx.xx.xx]
TASK [Power states of the nodes will be uploaded to the below csv file]
*****************************************************************************************************
*********************************
ok: [10.xx.xx.xx]
TASK [Getting Power State of Cray XD675 Server nodes]
*****************************************************************************************************
************************************************
ok: [10.xx.xx.xx]
```

```
TASK [Writing Power status details to Power_State_CrayXD675_2024-11-29_16:58:01.csv file]
*********************************************************************************************
*****
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************
*********************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

$ cat Power_State_CrayXD675_2024-11-29_16:58:01.csv
IP_Address,Model,Power_State
10.xx.xx.xx,HPE Cray XD675,On


$ ansible-playbook -i inventory system_firmware_update.yml -e @system_credentials.yml

PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*********************************************************************
TASK [Gathering Facts]
*********************************************************************************************
*********************************************************************************
ok: [10.xx.xx.xx]
TASK [System Firmware Update Status result will be uploaded to the below csv file]
*********************************************************************************************
***********************
ok: [10.xx.xx.xx]
TASK [Running Firmware Update for Cray XD Servers]
*********************************************************************************************
*****************************************************
ok: [10.xx.xx.xx]
TASK [Writing Firmware Upgrade status details to System_FW_Update_2024-11-29_17:01:36.csv file]
*********************************************************************************************
**********
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************
*********************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0

$ cat  System_FW_Update_2024-11-29_17:01:36.csv
IP_Address,Model,Update_Status,Remarks
10.xx.xx.xx,HPE Cray XD675,success,HIB_FPGA firmware update is successfully completed.
```

Here is an example of the update for HPE Cray XD675 GPU

The GPU update must be followed by an AC power cycle. AC power cycle can be done through both GUI as well as manual plug-out and plug-in of power cables.

```
$ cat config.ini
[Image]
update_image_path_xd675 = BKC_24.12.10.75_PROD.pldm

[Target]
update_target = gpu
##Allowed options are:
#XD675 - bmc, bios, dcscm_fpga, mb_fpga, hib_fpga, rot, gpu
##all targets are case-sensitive. Please follow the exact same cases.

[Options]
power_state = NA
##Allowed options are: NA on off
##applicable only for XD675
```

```
$ ansible-playbook -i inventory system_firmware_update.yml -e @system_credentials.yml
PLAY [version 1.0 System Firmware Update for HPE Cray XD675 model systems]
*****************************************
TASK [Gathering Facts]
*********************************************************************************************************
*****************************************************************
ok: [10.xx.xx.xx]
TASK [System Firmware Update Status result will be uploaded to the below csv file]
*********************************************************************************************************
********
ok: [10.xx.xx.xx]
TASK [Running Firmware Update for Cray XD Servers]
*********************************************************************************************************
************************************
[WARNING]: Collection community.general does not support Ansible version 2.10.8
ok: [10.xx.xx.xx]
TASK [Writing Firmware Upgrade status details to System_FW_Update_2024-12-13_09:55:52.csv file]
*********************************************************************************************
changed: [10.xx.xx.xx]
PLAY RECAP
*********************************************************************************************************
*************************************************************************
10.xx.xx.xx                : ok=4    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0


$ cat System_FW_Update_2024-12-13_09:55:52.csv
IP_Address,Model,Update_Status,Remarks
10.14.56.170,HPE Cray XD675,success,GPU firmware update completed successfully. Please perform an
AC power cycle to activate the latest firmware.
```

Using the Docker image for Firmware Update, Report and Power state

- Ensure Docker is installed in the system and docker daemon is running. Clone the AFUT repository from https://github.hpe.com/hpe/CrayXD_AFUT_XD675 , and cd to the cloned repository.

  ```
  $ cd CrayXD_AFUT_XD675
  ```

- Execute below shell command to extract the AFUT build version from "`system_firmware_update.yml`" and store it in the variable "VERSION".

  ```
  $ VERSION=$(grep -m1 'name:' system_firmware_update.yml | awk '{print $3}')
  ```

- Build the docker image using below command, where VERSION is passed as an argument to Dockerfile to include image version.

  ```
  $ docker build --build-arg VERSION=$VERSION -t <image-name>:$VERSION .
  ```

  Here, <image-name> should be replaced with user-defined custom name of the image being built.

- Ensure the image is built successfully by checking the image details after executing below command:

  ```
  $ docker images
  ```

- Create and run a docker container from the installed docker image using below command:

  ```
  $ docker run -it --name <container-name> -d <image-name>:<tag>
  ```

  Here,
  <container-name> is replaced by the name that the user intends to give to the container being created.
  <image-name> is the name of the image whose container is being created.
  <tag> is the tag corresponding to <image-name> when “docker images” command is executed.

- Ensure the container is running by checking the container status after executing below command:

  ```
  $ docker ps
  ```

- Every time we run the "docker run" command, it creates a new container of that image. In case, to run the already existing container, execute:

  ```
  $ docker start <container-name>
  ```

- To copy a file/directory from host machine to the container, use below command:

  ```
  $ docker cp <source-path> <container-name>:/app
  ```

- Enter the CLI of the running container using:

  ```
  $ docker exec -it <container-name> /bin/bash
  ```

- Follow the same procedure mentioned in earlier sections to perform firmware update, get/set power state, get system firmware inventory and get GPU firmware inventory operations from the container's CLI.

- To exit the container’s CLI, type “exit”. To stop the running container, execute below command:

  ```
  $ docker stop <container-name>
  ```

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, use the **Feedback** button and icons (located at the bottom of an opened document) on the Hewlett Packard Enterprise Support Centre portal (**https://www.hpe.com/support/hpesc**) to send any errors, suggestions, or comments. All document information is captured by the process.