MLNX-OS Up/Downgrade via Ansible v1.0

HyungKwang Choi (2021.9)

Table of Contents

1. Test environment 3

1.1 Topology. 3

1.2 Device info. 3

1.2.1 Server 3

1.2.1 SB7700 3

1.3 Scripts 5

1.4 License 5

1.4.1 if your MLNX-OS version is above v3.7, you do not need a license. 5

1.4.2 if your MLNX-OS version is lower than v3.7, you need a license. 5

2. Running 7

2.1 Introduction on my Ansible script 7

2.2.1 The Purpose 7

2.2.2 What it does 7

2.2.3 Exact steps 8

2.2.4 How to run continuously. 9

2.2 How to run 9

2.3 Running result : "Output.text" 9

3. Possible script failures 11

3.1 Ansible & Python version mis-match. 11

3.2 License invalid. 11

4. Reference 12

4.1 Ansible module for Mellanox 12

4.2 From NVIDIA Doc, how to 12

1. Test environment

1.1 Topology.

CentOS Server (ansible server)----------remote----------standalone SB7700

1.2 Device info.

1.2.1 Server

- CentOS 8.0
- Ansible 2.9.24
- Python 3.6.8
- Please place images, which you are running OS Up/Downgrade.

```
[root@NVIDIA ~]# cat /etc/redhat-release
CentOS Linux release 8.0.1905 (Core)

[root@NVIDIA ~]# ansible --version
ansible 2.9.24
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.6.8 (default, May 21 2019, 23:51:36) [GCC 8.2.1 20180905 (Red Hat 8.2.1-3)]

[root@NVIDIA ~]# ls -al | grep image
-rw-r--r--   1 root root 442345149 Aug 24 02:21 image-X86_64-3.6.2102.img
-rw-r--r--   1 root root 591062588 Aug 24 02:22 image-X86_64-3.6.6162.img
-rw-r--r--   1 root root 599453042 Aug 23 16:29 image-X86_64-3.6.8008.img
-rw-r--r--   1 root root 599644444 Aug 23 14:24 image-X86_64-3.6.8010.img
```

1.2.1 SB7700

```
NVIDIA [standalone: master] (config) # show  version
Product name:      MLNX-OS
Product release:   3.6.8010
Build ID:          #1-dev
Build date:        2018-08-20 18:04:19
Target arch:       x86_64
```

```
Target hw:         x86_64
Built by:          jenkins@XXXXXXX
Version summary:   X86_64 3.6.8010 2018-08-20 18:04:19 x86_64

NVIDIA [standalone: master] (config) # show  inventory
------------------------------------------------------------------------------
Module          Part Number       Serial Number       Asic Rev.    HW Rev.
------------------------------------------------------------------------------
CHASSIS         MSB7700-ES2F      MTXXXXXX        N/A         A7
MGMT            MSB7700-ES2F      MTXXXXXX        0           A7
FAN1            MTEF-FANF-A       MTXXXXXX        N/A         A3
FAN2            MTEF-FANF-A       MTXXXXXX        N/A         A3
FAN3            MTEF-FANF-A       MTXXXXXX        N/A         A3
FAN4            MTEF-FANF-A       MTXXXXXX        N/A         A3
PS1             MTEF-PSF-AC-A     MTXXXXXX        N/A         A2
PS2             MTEF-PSF-AC-A     MTXXXXXX        N/A         A2


NVIDIA [standalone: master] (config) # show  images

Installed images:
  Partition 1:
    version: X86_64 3.6.6162 2018-08-03 10:36:36 x86_64

  Partition 2:
    version: X86_64 3.6.8010 2018-08-20 18:04:19 x86_64

Last boot partition: 2
Next boot partition: 2

Images available to be installed:
  1:
    Image  : image-X86_64-3.6.8010.img
    Version: X86_64 3.6.8010 2018-08-20 18:04:19 x86_64

Serve image files via HTTP/HTTPS: no

No image install currently in progress.
Boot manager password is set.

Image signing  : trusted signature always required
Admin require signed images: yes

Settings for next boot only:
  Fallback reboot on configuration failure: yes (default)
```

1.3 Scripts

1. Ansible “OS_Up_downgrade.yml”

2. Inventory_hostname file

```
[root@NVIDIA ~]# cat /etc/ansible/hosts
[ONYX]
NVIDIA.test.labs ←=== modify properly.

[ONYX:vars]
ansible_network_os=onyx
ansible_become=yes
ansible_become_method=enable
ansible_ssh_user=admin
ansible_ssh_pass=admin
```

1.4 License

- License is not mandatory.
- If you want to check the SSD status or “Erase Count”, you need a license if version is lower than 3.7

1.4.1 if your MLNX-OS version is above v3.7, you do not need a license.

Because the command “#(config)fae show smart” is applicable as of v3.7.

1.4.2 if your MLNX-OS version is lower than v3.7, you need a license.

Because “#(config)fae show smart” does not work below v3.7
Then you need a license to check SSD status.

1). How to get license. You need mgmt mac-address and contact CSI lab admin

```
NVIDIA [standalone: master] (config) # show  interfaces mgmt0  ←==========
Interface mgmt0 status:
  Comment:
  Admin up:          yes
  Link up:           yes
  DHCP running:      yes
  IP address:        XXXX.XXXX.XXXX.XXXX
  Netmask:           255.255.252.0
  IPv6 enabled:      yes
  Autoconf enabled:  no
  Autoconf route:    yes
  Autoconf privacy:  no
  DHCPv6 running:    yes (but no valid lease)
```

```
  IPv6 addresses:     1
  IPv6 address:       XXXX::XXXX:XXXX:XXXX:XXXX/64
  Speed:              1000Mb/s (auto)
  Duplex:             full (auto)
  Interface type:     ethernet
  Interface source:   bridge
  MTU:                1500
  HW address:         XX:XX:XX:XX:XX:XX  <==============
```

2). How to see SSD status or “Erase Count”

```
(config) # license install <license key>
(config) # show  licenses
(config) # _exec iSmart -d /dev/sda

NVIDIA [standalone: master] # show  licenses

NVIDIA [standalone: master] # _exec iSmart -d /dev/sda

************************************************************************************************
* Innodisk iSMART V3.9.41                                        2018/05/25 *
************************************************************************************************
Model Name: InnoDisk Corp. - mSATA 3ME
FW Version: S141119O
Serial Number: XXXXXXXXXXXXXXXXXXX
Health: 0.00
Capacity: 14.914146 GB
P/E Cycle: 3000
Lifespan : 0 (Years : 0 Months : 0 Days : 0)
Write Protect: Disable
InnoRobust: Disable
------------------------------------------------------------------------------------------
ID    SMART Attributes                  Value         Raw Value
------------------------------------------------------------------------------------------
[09]  Power On Hours                   [53455]       [0900000000CFD000000000]
[0C]  Power Cycle Count                 [ 104]       [0C00000000680000000000]
[AA]  Total Bad Block Count             [  13]       [AA0300646400000D00000000]
[AD]  Erase Count Max.                 [ 5861]       [AD12006464DE15E516000000]
[AD]  Erase Count Avg.                 [ 5598]       [AD12006464DE15E516000000]
[C2]  Temperature                       [   0]       [000000000000000000000000]
[EB]  Later Bad Block                   [   0]       [EB020064000000000000000]
[EB]  Read Block                        [   0]       [EB020064000000000000000]
[EB]  Write Block                       [   0]       [EB020064000000000000000]
[EB]  Erase Block                       [   0]       [EB020064000000000000000]
[EC]  Unstable Power Count              [   0]       [EC02006464000000000000]

(config) # license delete 1
```

2. Running

2.1 Introduction on my Ansible script

2.2.1 The Purpose

- It does MLNX-OS upgrade and downgrade continuously for replication test purpose.

While I am handling cases, sometimes, there are some customer complaints about “during OS/FW upgrade or downgrade, SSD gets stuck or errored block, Erase count increases”.

To verify that we TAC engineers sets up a lab. So to run it automatically via Ansible, we can save time and focus on the SSD data collected during OS upgrade/downgrade.

2.2.2 What it does

1). Doing OS upgrade -> downgrade -> upgrade…..

3.6.2102 ->3.6.6162 -> 3.6.8010 -> 3.6.6162-> 3.6.2102…….

2). During that, it writes current version, collects SSD status and write to a output.txt file

```
# print/write the current version
   - name: print_current_version
     debug:
       msg: "Current version {{ current_version }}"
   - name: Write  Version
     local_action: shell echo "Current version {{ current_version }}" >> /root/output.txt

# Add license
   - name: Add license
     onyx_config:
          lines:
            - license install XXXX_XXXXXX_XXXXXXX_XXXXXXX_XXXXXX_XXXXXXXX
     register: add_license


# Get the SSD status
   - name: _exec iSmart -d /dev/sda
     onyx_command:
       commands: _exec iSmart -d /dev/sda
     register: ssd_data
```

```
# Write the SSD status to output.txt file
   - name: SSD status
     local_action: shell echo "{{ ssd_data.stdout }}" >> /root/output.txt
```

2.2.3 Exact steps

```
# Runing main task
- name: onyx_upgrade
~
~
  tasks:  ←=== main task

# get current switch Onyx version with command "show version concise" and save it to a variable named curr_version
   - name: get_current_switch_version

# save the onyx version as a fact named current_version
   - name: set_fact_current_version

# print/write the current version
   - name: print_current_version

# Add license
   - name: Add license

# Get the SSD status
   - name: _exec iSmart -d /dev/sda

# Write the SSD status to output.txt file
   - name: SSD status

# run the delete images command
   - name: delete images

# check if the img file exits in location
   - name: check_source_file_existence

# Scp file copy
   - name: copy_source_file_to_device

# Image install
   - name: image install

# # image boot next
   - name: image boot next

# # save config
   - name: save config

# run the reload command
   - name: reload
```

```
# use wait_for to wait for port 22 to be available on the switch after reboot
   - name: wait for switch to reboot
```

2.2.4 How to run continuously.

Above “Exact steps” is 1 cycle. So it repeated only changing version info. So if you want to run many cycle, just copy & paste whole steps as many as you want to run.

2.2 How to run

[root@NVIDIA ~]# ansible-playbook OS_Up_downgrade.yml

- For debugging

[root@NVIDIA ~]# ansible-playbook OS_Up_downgrade.yml -vvvv

2.3 Running result : “Output.text”

[root@NVIDIA ~]# ansible-playbook OS_Up_downgrade.yml

```
[root@NVIDIA ~]# cat output.txt

Current version 3.6.2102

*********************************************************************************************
* Innodisk iSMART V3.9.41                                    2018/05/25 *
*********************************************************************************************
Model Name: InnoDisk Corp. - mSATA 3ME
FW Version: S141119O
Serial Number: XXXXXXXXXXXXXXXXXXXX
Health: 0.00
Capacity: 14.914146 GB
P/E Cycle: 3000
Lifespan : 0 (Years : 0 Months : 0 Days : 0)
Write Protect: Disable
InnoRobust: Disable
---------------------------------------------------------------------------------------
ID    SMART Attributes                    Value         Raw Value
---------------------------------------------------------------------------------------
[09]  Power On Hours                    [53455]       [0900000000CFD00000000000]
```

[0C] Power Cycle Count	[104]	[0C00000000680000000000]
[AA] Total Bad Block Count	[13]	[AA0300646400000D00000000]
[AD] Erase Count Max.	[5861]	[AD12006464DE15E516000000]
[AD] Erase Count Avg.	[5598]	[AD12006464DE15E516000000]
[C2] Temperature	[0]	[000000000000000000000000]
[EB] Later Bad Block	[0]	[EB0200640000000000000000]
[EB] Read Block	[0]	[EB0200640000000000000000]
[EB] Write Block	[0]	[EB0200640000000000000000]
[EB] Erase Block	[0]	[EB0200640000000000000000]
[EC] Unstable Power Count	[0]	[EC0200646400000000000000]

3. Possible script failures

3.1 Ansible & Python version mis-match.

- My lab environment. Please check and compare it with yours

- CentOS 8.0
- Ansible 2.9.24
- Python 3.6.8

3.2 License invalid.

- One of my script running is “Adding license & SSD status.”. Please modify/add license properly.

```
# Add license

   - name: Add license
     onyx_config:
          lines:
            - license install XXXX_XXXXXX_XXXXXXX_XXXXXXX_XXXXXX_XXXXXXXX
     register: add_license


# Get the SSD status

   - name: _exec iSmart -d /dev/sda
     onyx_command:
       commands: _exec iSmart -d /dev/sda
     register: ssd_data
```

1). My script is written based on v3.6. So for this, license required to check SSD status.

2). If you want to skip to check SSD status, simply remove the above statement.

```
NVIDIA [standalone: master] # show  licenses
License 1: XXXX_XXXXXX_XXXXXXX_XXXXXXX_XXXXXX_XXXXXXXX
  Feature:          RESTRICTED_CMDS_GEN2
  Description:      Access to restricted system functionality
  Valid:            yes
  End date:         2023/07/13 (ok)
  Tied to MAC addr: XX:XX:XX:XX:XX:XX (ok)
  Active:           yes  <==========
```

4. Reference

4.1 Ansible module for Mellanox

https://docs.ansible.com/ansible/latest/collections/mellanox/onyx/onyx_command_module.html

4.2 From NVIDIA Doc, how to

https://docs.mellanox.com/display/ONYXv381208/Ansible

https://community.mellanox.com/s/article/Network-Automation-with-ZTP-and-Ansible-ONYX