Software Management Operations

This chapter provides information about software management operations on the system.

Understanding the Local File System

The local file system on the VPC VM is made up of files that are stored on the following:

- **/flash** Flash memory allocated as vHDD-1 on the M via the hypervisor is the default storage media for the StarOS image, CLI configuration, and crash log files used by the system.
- **/hd-raid** This is the storage space allocated as vHDD-2 on the CF VM by the hypervisor. It is used to store CDRs (Charging Data Records) and UDRs (Usage Data Records).

File Types Used by the Local File System

The following file types can be located in the local file system:

- **Operating System Software Image File**: This binary file type is identified by its **.bin** extension. The file is the operating system that is loaded by the system upon startup or reloading. This is an executable, read-only file that cannot be modified by end users.
- **CLI Configuration File**: This file type is identified by its **.cfg** extension. These are text files that contain CLI commands that work in conjunction with the operating system software image. These files determine services to be provided, hardware and software configurations, and other functions performed by the system. The files are typically created by the end user. You can modify the files both on and off-line and use descriptive long filenames.

- **System File**: Only one file identified by a **.sys** extension is used by the system. The boot.sys file contains system-specific information, which describes how the system locates, and in what priority it loads, file groups (paired .bin and .cfg files) from its boot stack.
- **Abridged Crash Log**: The abridged crash log, identified by its **crashlog** filename, contains summary information about software or hardware failures that occur on the system. This file is located in the **/flash/crsh2/** directory on the device. You can view the contents of this file through the CLI, but you cannot modify the file.

Understanding the boot.sys File

The system uses the boot.sys file to store the prioritized boot stack parameters and file groups the system uses during startup. Modify this file only through system CLI commands and not through external means. Boot parameters contain information the system needs to locate the operating system image file, including:

- **bootmode**: This setting is typically configured to normal, and identifies how the system starts.
- **boot stack information**: The boot stack is made up of prioritized file group entries that designate the operating system image file and the CLI configuration file to load.

When a system is started for the first time, the boot.sys file is configured to use the normal boot mode and load the operating system software image from the /flash directory.

There is no CLI configuration file contained on the local file system. This causes the system to automatically start its CLI-based Quick Setup Wizard upon the first successful boot. Refer to *Getting Started* for more information on using the Quick Setup Wizard.

Maintaining the Local File System

Use CLI commands to manage and maintain the devices that make up the local file system. Execute all the commands described in this section in the Exec Mode. Unless otherwise specified, you must have security administrator or administrator privileges to execute these commands.

Note You must be logged into the active CF VM to run the commands described below.

File System Management Commands

Use the commands in this section to manage and organize the local file system.

Important For complete information on the commands listed below, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference.*

Creating Directories

Use the **mkdir** command to create a new directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

`[local]host_name# mkdir { /flash | /usb1 | /hd-raid } /dir_name`

Use the following command to create a directory named *configs*:

`[local]host_name# mkdir /flash/configs`

Renaming Files and Directories

Use the **rename** command to change the name of a file from its original name to a different name. Remember to use the same file extension, if applicable, to ensure that the file type remains unchanged.

Use the following command to rename a file named *iot_test.cfg* to *iot_accept.cfg* on the */flash* local device.

`[local]host_name# rename /flash/iot_test.cfg /flash/iot_accept.cfg -noconfirm`

Important Use the **rename** command only within the same local device. You cannot rename a file and place it onto another local device at the same time. To move a renamed file, you must use the **copy** command.

Copying Files

These instructions assume that you are at the root prompt for the Exec mode. To save your current configuration, enter the following command:

`[local]host_name# copy from_url to_url [-noconfirm]`

To copy a configuration file called *system.cfg* from a directory that was called *cfgfiles* to a directory named *configs_old*, enter the following command:

```
[local]host_name# copy /flash/cfgfiles/system.cfg
/flash/configs_old/system_2011.cfg
```

To copy a configuration file called *simple_ip.cfg* from a directory called *host_name_configs* to an FTP server with an IP address of *192.168.34.156*, on which you have an account with a username of *administrator* and a password of *secure*, use the following command:

```
[local]host_name# copy /flash/host_name_configs/simple_ip.cfg
ftp://administrator:secure@192.168.34.156/host_name_configs/ simple_ip.cfg
```

To copy a configuration file called *init_config.cfg* to the root directory of a TFTP server with a hostname of *config_server*, enter the following command:

```
[local]host_name# copy /flash/cfgfiles/init_confg.cfg
tftp://config_server/init_config.cfg
```

Deleting Files

The **delete** command removes a designated file from its specified location on the local file system.

Important This command does not support wildcard entries; each filename must be specified in its entirety.

Caution Do not delete the boot.sys file. If deleted, the system will not reboot on command and will be rendered inoperable.

```
[local]host_name# delete { /flash | /usb1 | /hd-raid }/filename [ -noconfirm
]
```

The following command deletes a file named *test.cfg* from the */flash* directory.

```
[local]host_name# delete /flash/test.cfg
```

Removing Directories

The **rmdir** command deletes a current directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

Important The directory you want to remove (delete) must be empty before executing the **rmdir** command. If the directory is not empty, the CLI displays a "Directory not empty" message and will not execute.

```
[local]host_name# rmdir url /dir_name
```

The following command deletes an empty directory named *configs* in the */flash* directory.

```
[local]host_name# rmdir /flash/configs
```

Formatting Local Devices

The **format** command performs a low-level format of a local device. This operation formats the device to use the FAT16 formatting method, which is required for proper read/write functionality with the operating system.

Important Local devices that have been formatted using other methods such as NTFS or FAT32 may be used to store various operating system, CLI configuration, and crash log files. However, when placing a new local device into the MIO/UMIO/MIO2 for regular use, you should format the device via the system prior to use. This ensures that the proper file allocation table format is used, preventing any possible discrepancies between other formats used with other operating systems.

Caution The **filesystem format** command removes all files and information stored on the device.

To format a local device for use by the local file system, enter the following command:

```
[local]host_name# filesystem format { /flash | /usb1 | /hd-raid }
```

Applying Pre-existing CLI Configuration Files

A pre-existing CLI configuration file is any .cfg file created to provide utility functions (such as clearing all statistics during testing) or created off-line using a text editor. There may be pre-existing configuration files stored on the local file system that can be applied to a running system at any time.

Caution If a configuration file is applied to a system currently running another CLI configuration, any like contexts, services, logical interfaces, physical ports, IP address pools, or other configured items will be overwritten if the same command exists in the configuration file being applied. Take caution to ensure that you are knowledgeable of the contents of the file being applied and understand what the service ramifications are if a currently running command is overwritten. Also note that changes will not be saved automatically.

A CLI configuration file, or script containing CLI commands, can be applied to a running system by entering the following command at the Exec mode prompt:

`[local]host_name# configure url [ verbose ]`

url specifies the location of the CLI configuration file to be applied. It may refer to a local or a remote file.

The following command applies a pre-existing CLI configuration file named *clearcmds.cfg* in the */flash* directory.

`[local]host_name# configure /flash/clearcmds.cfg`

Viewing Files on the Local File System

This section describes how to view a variety of files.

Viewing the Contents of a Local Device

The contents, usage information, and file system directory structure of any local device can be viewed by entering the following command at the Exec mode prompt:

`directory { /flash | /usb1 | /hd-raid }`

Viewing CLI Configuration and boot.sys Files

The contents of CLI configuration and boot.sys files, contained on the local file system, can be viewed off-line (without loading them into the OS) by entering the following command at the Exec mode prompt:

`[local]host_name# show file url { /flash | /usb1 | /hd-raid } filename`

Where: *url* is the path name for the location of the file and *filename* is the name of the file, including any extension.

Important Operator and inspector-level users can execute the **show file** command but cannot execute the **directory** command.

Validating an Operating System File

The operating system software image file, identified by its .bin extension, is a non-readable, non-editable file that executes on the system, creating its runtime operating system (OS).

It is important to verify a new operating system image file before attempting to load it. To accomplish this, a proprietary checksum algorithm is used to create checksum values for each portion of the application stored within the .bin file during program compilation.

This information can be used to validate the actual file against the checksum values stored within the file during its compilation. If any portion of the image file has become corrupted (for example, the file was truncated or was transferred using ASCII mode instead of binary mode), then this information is reported and the file is deemed unusable.

To validate an operating system software image file, enter the following command at the Exec mode prompt:

`[local]host_name# show version { /flash | /usb1 | /hd-raid } /[directory]/filename [all]`

The output of this command displays the following information:

- Version number
- Description
- Date
- Boot Image
- Size
- Flags
- Platform

If an invalid file is found, the system displays a failure message similar to these:

```
Failure: Image /flash/image_version.bin CRC check failed!
Failure: /flash/image_version.bin, has a bad magic number
```

Cloud Initialization Support for Elastic Services Controller

When Elastic Services Controller (ESC) uses Cinder Multi-Attach volume on Control Function (CF), Active and Standby for QvPC-DI, the invoked Openstack API version for virtual machine (VM) Orchestration is 2.60 or higher. In this API version, ESC encodes and injects the configuration files into the VM for security reasons. Since, the VM is unable to read the encoded configuration files, ESC uses the **user_data** compressed file. This **user_data** file contains the configuration files that are required to boot VM.

Configuring the Boot Stack

The boot stack consists of a prioritized listing of operating system software image-to-CLI configuration file associations. These associations determine the software image and configuration file that gets loaded during system startup or upon a reload/reboot. Though multiple associations can be configured, the system uses the association with the highest priority. If there is an error processing this association (for example, unable to locate one of the files), the system attempts to use the association with the next highest priority.

For VPC-SI and VPC-DI platforms, when the configuration file in the highest configured boot priority is not available (but the image file is), the system boots up with the configuration setup wizard after reloading instead of using the next available boot system priority. Priorities range from 1 to 100, with 1 being the highest priority. The maximum number of boot stack entries that may be configured in the boot.sys file is 10.

Boot stack information is contained in the boot.sys file, described in Understanding the boot.sys File, on page 2. In addition to boot stack entries, the boot.sys file contains any configuration commands required to define the system boot method as explained in the section that follows.

System Boot Methods

The local-boot method uses software image and configuration files stored locally on the system. On system startup or reboot, the system looks on one of its local devices or **/hd-raid** for the specific software image and accompanying configuration text file. When using the local-booting method, you only need to configure boot stack parameters.

The system can also be configured to obtain its software image from a specific external network server while it is paired with a configuration text file that resides on the system. When using network booting, you need to configure the following:

- Boot stack parameters, which define the files to use and in what priority to use them
- Boot interface and network parameters defining the remote management LAN interface and the methods to use to reach the external network server
- Network booting delay time and optional name server parameters defining the delay period (in seconds) to allow for network communications to be established, and the IP address of any Domain Name Service (DNS) name server that may be used

Viewing the Current Boot Stack

To view the boot stack entries contained in the boot.sys file run the Exec mode **show boot** command.

Important Operator and inspector-level users can execute the **show boot** command.

The examples below shows the command output for a local booting configuration. Notice that in these examples both the image file (operating system software) and configuration file (CLI commands) are located on the **/flash** device.

Important The StarOS image filename format "asr5500-*image_number*.bin".

Example :

```
boot system priority 18 \
    image /flash/16-1-builds/asr5500-16.1.3.bin \
    config /flash/general_config.cfg

boot system priority 19 \
    image /flash/16-1-builds/asr5500-16.1.1.bin \
    config /flash/general_config_3819.cfg

boot system priority 20 \
    image /flash/16-1-builds/asr5500-16.1.0.bin \
    config /flash/general_config_3665.cfg
```

The example below shows the output for a combination network booting and local booting configuration. Notice in this example that the first two boot stack entries (Priorities 18 and 19) load the image file (operating system software) from an external network server using the Trivial File Transfer Protocol (TFTP), while all configuration files are located on the **/flash** device.

Also notice the boot network interface and boot network configuration commands located at the top of the boot stack. These commands define what remote management LAN interface(s) to use and information about communicating with the external network server that hosts the operating system software image file.

```
boot networkconfig static ip address mio1 192.168.1.150 netmask 255.255.255.0
boot delay 15
boot system priority 18 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 19 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 20 image /flash/image_version.bin \config /flash/general_config.cfg
```

To identify the boot image priority that was loaded at the initial boot time enter:

show boot initial-config

The example below displays the output:

```
[local]host_name# show boot initial-config
Initial (boot time) configuration:
      image tftp://192.168.1.161/tftpboot/image_version.bin \
       config /flash/config_name.cfg
      priority 1
```

Adding a New Boot Stack Entry

Important Before performing this procedure, verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (i.e. that minimally there is no priority 1 entry in the boot stack). Refer to *Viewing the Current Boot Stack* for more information.

If priority 1 is in use, then you must renumber the existing entry(ies) to ensure that at least that priority is available. The maximum number of boot stack entries that can be contained in the *boot.sys* file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Refer to Deleting a Boot Stack Entry, on page 9 for more information.

This procedure details how to add new boot stack entries to the *boot.sys* file. Make sure you are at the Exec mode prompt and enter the following commands:

configure
boot system priority *number* **image** *image_url* **config** *cfg_url*

The following command creates a new boot stack entry, using a boot priority of *3*.

boot system priority 3 image /flash/*image_filename***.bin config /flash/***config_name***.cfg**

Important Boot stack changes saved to the boot.sys file are not executed until the system is rebooted.

Synchronize the local file systems on the CF VMs with the following command:

```
filesystem synchronize all
```

Deleting a Boot Stack Entry

This procedure details how to remove an individual boot stack entry from the boot.sys file. Make sure you are at the Exec mode prompt and enter the following commands:

```
configure
  no boot system priority number
```

Where *number* specifies the boot priority used for the boot stack entry. This command removes that specific entry from the boot stack, causing the boot.sys file to be overwritten.

Upgrading the Operating System Software

This section describes how to manually perform the StarOS binary image upgrade procedure. For the automated bin upgrade procedure, see the Automated StarOS Bin Upgrade, on page 32 section.

Prior to initiating the StarOS software upgrade procedure, make sure the conditions described in the *Prerequisites* section are met.

Caution Undeploying/redeploying VPC is not supported after the bin upgrade. Deactivating VPC removes the upgraded StarOS bin image.

To upgrade the StarOS software manually:

1. *Obtain VIP Addresses for AutoVNF, CF, ESC and UEM*
2. Identify OS Release Version and Build Number, on page 11
3. Download the Software Image from the Support Site, on page 12
4. *Verify Zookeeper Database*
5. *Verify ESC Database*
6. Verify Free Space on the /flash Device, on page 14
7. Transfer StarOS Image to /flash, on page 15
8. Save the Running Configuration, on page 18
9. Synchronize File Systems, on page 26
10. Reboot the System, on page 21

Prerequisites

Prior to performing an upgrade of StarOS software containing CF and SF VNFCs, check if the following prerequisites are met:

- You'll need the login credentials and IP address of AutoDeploy, AutoVNF, ESC, UEM, and CF VMs. You should have administrative rights to the OpenStack setup.
- Verify the OpenStack status. The Ansible output should all pass.

```
cd /home/stack/
source stackrc
cd /home/stack/ansible/
ansible-playbook -i inventory openstack_verify.yml
```

- Check if the health of AutoVNF/ESC/EM/VNF VM is normal through the UltraM health logs on AutoIT. If any of the VM(s) are not normal, then take necessary actions to rcover the health of the corresponding VM(s).
- You should have the new StarOS binary image file (for manual upgrade) or new StarOS CF QCOW2 image (for automated upgrade).
- Ensure that there are no pending transactions between ESC, UEM and CF.
- Be sure to take a backup of the original StarOS bin file.

Note Taking backup is not necessary for the automated bin upgrade as this will be automatically taken care through the upgrade RPC.

Obtain VIP Addresses for AutoVNF, CF, ESC and UEM

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To collect the VIP addresses for AutoVNF, CF, ESC and UEM VMs:

1. Log on to the AutoDeploy VM as the default user, *ubuntu*.

   ssh ubuntu@*<ad_vm_address>*

2. Switch to the *root* user.

   sudo -i

3. Enter the ConfD CLI.

   confd_cli -u admin -C

4. Enter the *admin* user credentials when prompted.
5. Collect the VIP address of AutoVNF, ESC, UEM and CF VMs.

   show vnfr

   Example output:

```
vnfr autoit-f-autovnf

 vnfd     f-autovnf

 vnf-type usp-uas
```

```
 state    deployed
 external-connection-point avf
  virtual-link-ref    management
  ip-address          192.168.100.26
  floating-ip-address 10.225.202.94
vnfr sj-autovnf-esc
 vnfd     esc
 vnf-type esc
 state    deployed
 external-connection-point esc
  virtual-link-ref management
  ip-address       192.168.100.22
vnfr sj-autovnf-vpc
 vnfd     vpc
 vnf-type ugp
 state    alive
 external-connection-point cf
  virtual-link-ref management
  ip-address       192.168.100.38
 external-connection-point em
  virtual-link-ref management
  ip-address       192.168.100.21
```

Identify OS Release Version and Build Number

The operating system can be configured to provide services and perform pre-defined functions through commands issued from the CLI.

The operating system software is delivered as a single binary file (**.bin** file extension) and is loaded as a single instance for the entire system.

A starfile image must be signed with an REL key before being released. A deployable image will be signed with an REL key having a ".bin.SPA" extension, where "A" identifies the revision level of the signing key. For example, **asr5500-20.0.0.bin.SPA**. If a signing key becomes compromised, a new key is created and the revision level increments to "B".

Trusted images have been introduced. The difference between a Trusted build and a Normal build is the absence of unsecure programs ftpd, telnet and tcpdump, as well as the addition of a staros.conf file for security

options. Trusted images are identifiable by the presence of "_T" in the platform name. For example, **asr5500_T-20.0.0.bin.SPA**.

To identify the StarOS software version and build information:

1. Log on to the VNF to be upgraded.
2. Enter the following Exec mode command in the StarOS command line interface:

   show version

   Example output:

```
Active Software:

  Image Version:             21.9.0.69918

  Image Build Number:        69918

  Image Description:         Deployment_Build

  Image Date:                Sun Jul 22 12:08:55 EDT 2018

  Boot Image:                /flash/staros.bin

  Source Commit ID:          94797337b6c1691541ea0dd86f2f29b0f2c3630c
```

3. Execute the following Exec mode command to display additional information about the StarOS build release.

   show build

Download the Software Image from the Support Site

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

Access to the Cisco support site and download facility is username and password controlled. You must have an active customer account to access the site and download the StarOS image.

Download the software image to a network location or physical device (USB stick) from which it can be uploaded to the */flash* device. Contact your Cisco representative or Cisco TAC for additional information.

For UGP-based VNF, perform the following steps to download the new bin file to AutoVNF or OSPD VM.

1. Log on to the AutoVNF of the corresponding VNF.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

   ssh ubuntu@10.225.202.94

2. Create a directory to download the new StarOS qvpc-di binary file to AutoVNF/OSPD.

   cd /home/ubuntu/

   mkdir StarOSBinUpgrade

3. Download the new StarOS qvpc-di binary file from the Cisco support site and copy the file to the *StarOSBinUpgrade* directory.

   cd StarOSBinUpgrade

Then, use the following command to verify if the directory contains the new bin file.

ls -lrt /home/ubuntu/StarOSBinUpgrade

Example output:

```
total 172560

-r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
```

Verify Zookeeper Database

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To verify the zookeeper database:

1. Log on to the AutoVNF using the floating IP.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

   ssh ubuntu@10.225.202.94

2. Log on to the UEM VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.21

3. Become the *root* user.

   sudo -i

4. Collect the UEM orchestration IP address for Zookeeper database connection.

   #ifconfig

   ```
   eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

             inet addr:209.165.200.240  Bcast: 209.165.200.255  Mask: 255.255.255.224
   ```

5. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* directory.

6. Execute the following script from the command line to access the UEM Zookeeper database.

   zkCli.sh -server *ip_addr:port_num*

   For example:

   zkCli.sh -server 209.165.200.240:2181

7. Check the zookeeper database and ensure that there are no pending requests between UEM and CF VMs.

   ls /request

   Example output:

```
[]

<Ctrl+D to exit Zookeeper shell>
```

Verify ESC Database

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To verify the ESC database:

1. Log on to the AutoVNF using the floating IP.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

   ssh ubuntu@10.225.202.94

2. Log on to the ESC VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

   ssh admin@*<vip-addr>*

   Command example:

   ssh admin@192.168.100.22

3. Check the ESC database to ensure there are no pending transactions.

   sudo /opt/cisco/esc/pgsql/bin/psql -U esc -p 7878 -h localhost -c 'select * from esc_schema.workitem';

   ```
   config_id | request_id | mo_type | config_action | config_state
   -----------+------------+---------+---------------+--------------
   (0 rows)
   ```

4. Execute the following command to check the transaction details.

   escadm ip_trans

   Example output:

   ```
   Number of in-progress transaction events = 0
   ```

Verify Free Space on the /flash Device

Verify that there is enough free space on the **/flash** device to accommodate the new StarOS image file.

To verify the available space on *flash* directory:

1. Log on to the CF VM using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.38

2. Enter the following Exec mode command:

```
[local]host_name# directory /flash
```

The following is an example of the type of directory information displayed:

```
-rwxrwxr-x    1  root     root         7334 May    5 17:29 asr-config.cfg
-rwxrwxr-x    1  root     root         399 Jun    7 18:32 system.cfg
-rwxrwxr-x    1  root     root         10667 May 14 16:24 testconfig.cfg
-rwxrwxr-x    1  root     root         10667 Jun    1 11:21 testconfig_4.cfg
-rwxrwxr-x    1  root     root         5926 Apr    7 16:27  tworpcontext.cfg
-rwxrwxr-x    1  root     root         15534 Aug    4 13:31 test_vlan.cfg
-rwxrwxr-x    1  root     root         2482 Nov 18 11:09 gateway2.cfg
-rwxrwxr-x    1  root     root         159106048 Dec 31  2011 image_filename
1136352  /flash
Filesystem           1k-blocks      Used Available Use% Mounted on
/var/run/storage/flash/part1 3115468   1136352  30018336   4%
/mnt/user/.auto/onboard/flash
```

Note the "Available" blocks in the last line of the display. After displaying the directory information, the CLI returns to root and the following prompt appears:

```
[local]host_name#
```

Transfer StarOS Image to /flash

For StarOS-based VNF, transfer the new operating system image file to the */flash* directory on the MIO/UMIO/MIO2 VPC-DI active CF or VPC-SI using one of the following methods:

- Transfer the file to the */flash* device using an FTP client with access to the system.

Important Whenever transferring a operating system software image file using the file transfer protocol (FTP), the FTP client must be configured to transfer the file using binary mode. Failure to use binary transfer mode will make the transferred operating system image file unusable. FTP is not supported.

- Transfer the file to the */flash* device using an SFTP client with access to the system.

For UGP-based VNF, copy the new StarOS bin to the active CF by following these steps.

1. Log on to the AutoVNF or OSPD VM where the new bin file is downloaded.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

   ssh ubuntu@10.225.202.94

2. Navigate to the directory where the new bin file is downloaded from the Cisco support site.

   cd /home/ubuntu/StarOSBinUpgrade/ && ls -lrt

   Example output:

   ```
   total 172560

   -r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
   ```

3. SFTP to the CF VM.

   For example:

sftp ubuntu@192.168.100.38

4. Navigate to the *sftp* directory.

```
#sftp>pwd
Remote working directory: /
#sftp>ls
hd-raid sftp
#sftp>cd sftp
```

5. Upload the new binary file to the *sftp* directory.

```
#sftp>put image_filename.bin
```

Example output:

```
#sftp>put qvpc-di-21.9.0.69932.bin
Uploading qvpc-di-21.9.0.69932.bin to /.auto/onboard/flash/sftp/qvpc-di-21.9.0.69932.bin
qvpc-di-21.9.0.69932.bin                      100%  169MB 168.5MB/s   00:01
```

6. Log on to the CF VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

ssh ubuntu@<*vip-addr*>

Command example:

ssh ubuntu@192.168.100.38

7. Copy the new bin from *sftp* to *flash* directory.

copy /flash/sftp/*image_filename.bin* **/flash/updated.bin**

Example output:

```
#copy /flash/sftp/qvpc-di-21.9.0.69932.bin /flash/updated.bin

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

Transferred 176698880 bytes in 2.718 seconds (63486.9 KB/sec)
```

8. Delete the new bin from *sftp* directory.

delete /flash/sftp/*image_filename.bin*

Example output:

```
delete /flash/sftp/qvpc-di-21.9.0.69932.bin
Are you sure? [Yes|No]: yes
File /flash/sftp/qvpc-di-21.9.0.69932.bin removed
```

9. Verify that the image file was successfully transferred to the */flash* device by running the following Exec mode command:

```
[local]host_name# directory /flash
```

The image filename should appear in the displayed output.

10. Execute the following command to verify the build information.

show version /flash/*image_filename***.bin**

Saving a Copy of the Current Configuration File

Prior to upgrading to a new software release, you should copy and rename the current configuration file to the **/flash** device and to an off-chassis location (external memory device or network URL). This renamed copy assures that you will have a fallback, loadable configuration file should a problem be encountered during the upgrade.

Off-line Software Upgrade

An off-line software upgrade can be performed for any system, upgrading from any version of operating system software to any version, regardless of version number. This process is considered off-line because while many of the steps can be performed while the system is currently supporting sessions, the last step of this process requires a reboot to actually apply the software upgrade.

This procedure assumes that you have a CLI session established and are placing the new operating system image file onto the local file system. To begin, make sure you are at the Exec mode prompt:

```
[local]host_name#
```

To perform offline software upgrade:

1. Configure a Newcall Policy, on page 17
2. Configure a Message of the Day Banner, on page 18
3. Back up the Current CLI Configuration File , on page 18
4. Save the Running Configuration, on page 18
5. Create a New Boot Stack Entry, on page 20
6. Synchronize File Systems, on page 26
7. Reboot the System, on page 21

Configure a Newcall Policy

Configure a newcall policy from the Exec mode to meet your service requirements. When enabled the policy redirects or rejects new calls in anticipation of the system reload that completes the upgrade process. This reduces the amount of service disruption to subscribers caused by the system reload that completes the upgrade.

Important Newcall policies are created on a per-service basis. If you have multiple services running on the chassis, you can configure multiple newcall policies.

The syntax for newcall policies is described below:

```
[local]host_name# newcall policy { asngw-service | asnpc-service | sgsn-service
 } { all | name service_name } reject
[local]host_name# newcall policy { fa-service | lns-service | mipv6ha-service
 } { all | name service_name } reject
[local]host_name# newcall policy { ha-service | pdsn-service |
pdsnclosedrp-service  } { all | name service_name } { redirect target_ip_address
 [ weight  weight_num ] [ target_ipaddress2 [ weight weight_num ] ...
```

```
target_ip_address16 [ weight weight_num ] | reject }
[local]host_name# newcall policy ggsn-service { apn name apn_name | all | name
 service_name } reject
[local]host_name# newcall policy hnbgw-service { all | name service_name } reject
[local]host_name# newcall policy { pcc-af-service | pcc-policy-service } {
all | name service_name } reject
[local]host_name# newcall policy {pcc-af-service | pcc-policy-service } { all
 | name service_name } reject
[local]host_name# newcall policy mme-service { all | name service_name } reject
```

For complete information about the above commands, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

Configure a Message of the Day Banner

Optional: Configure a "Message of the Day" banner informing other management users that the system will be rebooted by entering the following command from the Global Configuration mode prompt.

```
[local]host_name(config)# banner motd "banner_text"
```

banner_text is the message that you would like to be displayed and can be up to 2048 alphanumeric characters. Note that *banner_text* must begin with and end in quotation marks (" "). For more information in entering CLI banner information, see the *CLI Reference*. The banner is displayed when an administrative user logs onto the CLI.

Back up the Current CLI Configuration File

Back up the current CLI configuration file by entering the following command:

```
[local]host_name# copy from_url to_url [ -noconfirm ]
```

This creates a mirror-image of the CLI configuration file linked to the operating system defined in the current boot stack entry.

The following command example creates a backup copy of a file called *general.cfg* located on the **/flash** device to a file called *general_3652.cfg*:

```
[local]host_name# copy /flash/general.cfg /flash/general_3652.cfg
```

Save the Running Configuration

Save the currently running, upgraded configuration prior to rebooting the chassis.

Important For the automated upgrade of StarOS binary image, perform only the Steps from 1 through 3 in the following procedure.

To save the boot configuration:

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10 .

   ```
   ssh ubuntu@<vip-addr>
   ```

   Command example:

   ```
   ssh ubuntu@192.168.100.38
   ```

2. *Optional*. Execute the following command in the Exec mode.

   chassis key value 1234

   ```
   Save config before reload chassis, EVEN IF the same old key value is used.
   Old config scripts will become invalid after reload.
   ```

Important This step is optional, and needed only if the chassis key is not set.

3. Save the boot configuration in the flash directory.

   save configuration /flash/system.cfg

   ```
   Warning: About to overwrite boot configuration file
   Are you sure? [Yes|No]: yes
   ```

   This will update the boot configuration to use the new bin image.

   Use the following command to check the boot configuration.

   # show boot

   ```
   Monday May 21 20:39:57 UTC 2018

   boot system priority 8 \
       image /flash/sftp/production.YYYYY.qvpc-di.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 9 \
       image /flash/staros.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 10 \
       image /flash/staros.bin \
       config /flash/system.cfg
   ```

4. Enter the configuration mode to change the boot priority of new StarOS bin file.

   #config

   #boot system priority 1 image /flash/updated.bin config /flash/system.cfg

   #end

5. Verify the new boot priority.

   #show boot

   ```
   boot system priority 1 \

       image /flash/updated.bin \

       config /flash/system.cfg

   boot system priority 10 \

       image /flash/staros.bin \

       config /flash/system.cfg
   ```

6. Verify whether the flash directory contains the boot configuration and new bin.

```
dir /flash
```

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

Create a New Boot Stack Entry

Create a new boot stack entry for the new file group, consisting of the new operating system image file and the currently used CLI configuration file by entering the following Global Configuration command:

```
[local]host_name(config)# boot system priority number image image_url /flash filename
 config cfg_url /flash/filename
```

Assign the next highest priority to this entry, by using the <N-1> method, wherein you assign a priority number that is one number less than your current highest priority.

Important Run the Exec mode **show boot** command to verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (minimally there is no priority 1 entry in the boot stack).

If priority 1 is in use, you must renumber the existing entries to ensure that at least that priority is available.

The maximum number of boot stack entries that can be contained in the boot.sys file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Use the no boot system priority command to delete a book stack entry.

```
[local]host_name# configure
[local]host_name(config)# no boot system priority number
```

To add new boot stack entries to the boot.sys file enter the following commands:

```
[local]host_name# configure
[local]host_name(config)# boot system priority number image image_url config cfg_url
```

For information on using the **boot system priority** command, refer to the Adding a New Boot Stack Entry, on page 8 .

Synchronize File Systems

Synchronize the local file systems on the management cards by entering the following command:

```
[local]host_name# filesystem synchronize all
```

Reboot the System

To reboot the system (VNF):

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10 .

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.38

2. Enter the following Exec mode command:

   `[local]host_name#` **reload [-noconfirm]**

   As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

3. *Optional for PDSN:* If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

4. After the reload is complete, log on to the VNF and make sure it is loaded with the intended StarOS version and all the cards have booted up and are in active or stand-by state as expected.

   show version

   Example output:

```
Active Software:
  Image Version:                 21.9.0.69977
  Image Build Number:            69977
  Image Description:             Build
  Image Date:                    Mon Jul 30 06:48:34 EDT 2018
  Boot Image:                    /flash/updated.bin
  Source Commit ID:              abde005a31c93734c89444b8aec2b6bb2d2e794d
```

   show card table

   Example output:

```
Slot         Card Type                            Oper State     SPOF  Attach
-----------  -----------------------------------  -------------  ----  ------
 1: CFC      Control Function Virtual Card        Active         No
 2: CFC      Control Function Virtual Card        Standby        -
 3: FC       4-Port Service Function Virtual Card Standby        -
 4: FC       4-Port Service Function Virtual Card Standby        -
 5: FC       4-Port Service Function Virtual Card Standby        -
 6: FC       4-Port Service Function Virtual Card Standby        -
 7: FC       4-Port Service Function Virtual Card Standby        -
 8: FC       4-Port Service Function Virtual Card Standby        -
 9: FC       4-Port Service Function Virtual Card Standby        -
10: FC       4-Port Service Function Virtual Card Standby        -
```

5. Run the following Exec mode command to display additional information about the running StarOS build release.

   show build

6. *Optional*. Verify the operational state of CF and SF VNFCs.

Note This step is relevant only for the upgrade of CF and SF VNFCs.

a. Repeat the steps in *Verify Zookeeper Database* and *Verify ESC Database* sections.

b. Log on to the UEM using either the floating IP or from the AutoVNF using the UEM VIP.

ssh ubuntu@*<vip-addr>*

Command example:

ssh ubuntu@192.168.100.21

c. Become the *root* user.

sudo -i

d. Collect the UEM orchestration IP address for Zookeeper database connection.

#ifconfig

```
eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

          inet addr:209.165.200.225  Bcast:209.165.200.255  Mask:255.255.255.224
```

e. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* directory.

f. Run Zookeeper tool to access the UEM Zookeeper database.

zkCli.sh -server *<vip-addr>:port_num*

Command example:

zkCli.sh -server 209.165.200.225:2181

Make sure there are no outstanding requests between UEM and CF.

g. Verify the "state": "alive" for each of the CFs and SFs using the following commands:

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf2**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf2**

Command examples:

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2

h. Look for the state Alive in the console output.

```
zk: localhost:2181(CONNECTED) 2] get
/oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
{"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
```

"cpts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"},
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1da1e1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"},
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d3cdd","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"},{"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b111e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0

Note You can use use **CTRL+D** to exit the zookeeper CLI.

i. From the UEM VM as a root user, log on to the *ncs_cli* and check for devices live status.

~$ sudo -i

ncs_cli -C -u admin

 # show devices device *device_name* **live-status**

Verify that the command output reflects the correct 'state' and 'card-state' of each card.

Example output:

 # show devices device tb1-autovnf1_vpc-vpc-core-cf-nc live-status

```
<snip>

VNFC  CURRENT VNFC  VDU   CARD  CARD  NUMBER CPU   DISK  START UPTIME NOVA  ID    DATE  FROM  TO
REF   STATE   INSTANCE REF   TYPE  SLOT  OF     UTILIZATION SPACE TIME       LAUNCH      AND   STATE STATE
              ID                  ID    CORES                               CMD         TIME
-------------------------------------------------------------------------------------------------
cf1   -       cf1   cf    cardtype 1     -      -     -     -     -      -
cf2   -       cf2   cf    cardtype 2     -      -     -     -     -      -
sf1   -       sf1   sf    servicetype 3  -      -     -     -     -      -
sf2   -       sf2   sf    servicetype 4  -      -     -     -     -      -


live-status vnfd sj-autovnf-vpc-abc
```

```
 version            6.0

 vnfm vim-tenant-name abc

 vnfm tenant-name abc

 vnfm ipaddr     192.168.100.22

 vnfm port       830

 vnfm username   ubuntu

 vnfm password   "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmkn1tRe1f84byNakWEa9sJ8sY/
cwfFME3aG0UaBC\nvvNNAMkuXQI9Ksfu5IiQQ9ViWbbHwl6IEFQ="

 virtual-link vl-di-internal1

  auto-vnf-connection-ref di-internal1

 virtual-link vl-management

  auto-vnf-connection-ref management

 virtual-link vl-orchestration

  auto-vnf-connection-ref orchestration

 virtual-link vl-abc-vpc-svc

  auto-vnf-connection-ref sj-autovnf-abc-vpc-svc

 vdu cf

  ssh-keygen         false

  vm-image           076c887a-a12c-4a0b-b4d6-b2d213f64b9e

  lifecycle-event-initialization staros_config.txt

   source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt


  lifecycle-event-initialization staros_param.cfg

   source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg


  ned cisco-staros-nc

   user        "$4$+HLzsE1kLJOeuoFWyOsmBWY2LHjOi2WtJdKy/OIux7YHhsNY/
O8hnA9/WwWuFD5trHrW3ZHs\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="

   password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNSJumC7n5rnkVxwI1s\
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="

   netconf

   port-number 830

  card-type          control-function

  usp-auto-vnf-id    sj-autovnf-vpc-abc-cf
```

```
    vnfc cf-vnfc-ugp

<snip>
```

Save the Running Configuration

Save the currently running, upgraded configuration prior to rebooting the chassis.

Important For the automated upgrade of StarOS binary image, perform only the Steps from 1 through 3 in the following procedure.

To save the boot configuration:

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10 .

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.38

2. *Optional*. Execute the following command in the Exec mode.

   chassis key value 1234

   ```
   Save config before reload chassis, EVEN IF the same old key value is used.
   Old config scripts will become invalid after reload.
   ```

   Important This step is optional, and needed only if the chassis key is not set.

3. Save the boot configuration in the flash directory.

   save configuration /flash/system.cfg

   ```
   Warning: About to overwrite boot configuration file
   Are you sure? [Yes|No]: yes
   ```

   This will update the boot configuration to use the new bin image.

   Use the following command to check the boot configuration.

   # show boot

   ```
   Monday May 21 20:39:57 UTC 2018

   boot system priority 8 \
       image /flash/sftp/production.YYYYY.qvpc-di.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 9 \
       image /flash/staros.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 10 \
   ```

```
    image /flash/staros.bin \
    config /flash/system.cfg
```

4. Enter the configuration mode to change the boot priority of new StarOS bin file.

   #config

   #boot system priority 1 image /flash/updated.bin config /flash/system.cfg

   #end

5. Verify the new boot priority.

   #show boot

```
boot system priority 1 \

    image /flash/updated.bin \

    config /flash/system.cfg

boot system priority 10 \

    image /flash/staros.bin \

    config /flash/system.cfg
```

6. Verify whether the flash directory contains the boot configuration and new bin.

   dir /flash

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

Synchronize File Systems

To synchronize the file systems:

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

2. Synchronize the local file systems on the management cards by entering the following command:

   `[local]`*host_name*`#` **filesystem synchronize all**

   Example output:

```
Updating /flash/system.cfg
********************************************************************************
Updating /flash/updated.bin
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-bulkstats-config.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-bulkstats.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-cli-config.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-confd-config.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-config.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-exec.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-kpi.yang
********************************************************************************
Updating /flash/sftp/yang/cisco-staros-notif.yang
********************************************************************************
Updating /flash/boot.sys
********************************************************************************
12 updated on card 2
           /flash/system.cfg
           /flash/updated.bin
           /flash/sftp/yang/cisco-staros-bulkstats-config.yang
           /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang
           /flash/sftp/yang/cisco-staros-bulkstats.yang
           /flash/sftp/yang/cisco-staros-cli-config.yang
           /flash/sftp/yang/cisco-staros-confd-config.yang
```

```
/flash/sftp/yang/cisco-staros-config.yang

/flash/sftp/yang/cisco-staros-exec.yang

/flash/sftp/yang/cisco-staros-kpi.yang

/flash/sftp/yang/cisco-staros-notif.yang

/flash/boot.sys
```

Reboot the System

To reboot the system (VNF):

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10 .

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.38

2. Enter the following Exec mode command:

   `[local]`*host_name*`#` **reload [-noconfirm]**

   As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

3. *Optional for PDSN:* If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

4. After the reload is complete, log on to the VNF and make sure it is loaded with the intended StarOS version and all the cards have booted up and are in active or stand-by state as expected.

   show version

   Example output:

```
Active Software:
  Image Version:                 21.9.0.69977
  Image Build Number:            69977
  Image Description:             Build
  Image Date:                    Mon Jul 30 06:48:34 EDT 2018
  Boot Image:                    /flash/updated.bin
  Source Commit ID:              abde005a31c93734c89444b8aec2b6bb2d2e794d
```

   show card table

   Example output:

```
Slot         Card Type                            Oper State     SPOF  Attach
-----------  -----------------------------------  -------------  ----  ------
 1: CFC      Control Function Virtual Card        Active         No
 2: CFC      Control Function Virtual Card        Standby        -
 3: FC       4-Port Service Function Virtual Card Standby        -
 4: FC       4-Port Service Function Virtual Card Standby        -
 5: FC       4-Port Service Function Virtual Card Standby        -
 6: FC       4-Port Service Function Virtual Card Standby        -
 7: FC       4-Port Service Function Virtual Card Standby        -
 8: FC       4-Port Service Function Virtual Card Standby        -
```

```
 9: FC       4-Port Service Function Virtual Card    Standby        -
10: FC       4-Port Service Function Virtual Card    Standby        -
```

5. Run the following Exec mode command to display additional information about the running StarOS build release.

   show build

6. *Optional*. Verify the operational state of CF and SF VNFCs.

Note This step is relevant only for the upgrade of CF and SF VNFCs.

   a. Repeat the steps in *Verify Zookeeper Database* and *Verify ESC Database* sections.

   b. Log on to the UEM using either the floating IP or from the AutoVNF using the UEM VIP.

      ssh ubuntu@*<vip-addr>*

      Command example:

      ssh ubuntu@192.168.100.21

   c. Become the *root* user.

      sudo -i

   d. Collect the UEM orchestration IP address for Zookeeper database connection.

      #ifconfig

      ```
      eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

                inet addr:209.165.200.225  Bcast:209.165.200.255  Mask:255.255.255.224
      ```

   e. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* directory.

   f. Run Zookeeper tool to access the UEM Zookeeper database.

      zkCli.sh -server *<vip-addr>:port_num*

      Command example:

      zkCli.sh -server 209.165.200.225:2181

      Make sure there are no outstanding requests between UEM and CF.

   g. Verify the "state": "alive" for each of the CFs and SFs using the following commands:

      get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf1**

      get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf2**

      get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf1**

      get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf2**

      Command examples:

      get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1

      get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2

      get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2

h. Look for the state Alive in the console output.

```
zk: localhost:2181(CONNECTED) 2] get
/oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
{"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
"cpts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"},
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1da1e1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"},
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d3cdd","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"},{"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b111e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0
```

Note You can use use **CTRL+D** to exit the zookeeper CLI.

i. From the UEM VM as a root user, log on to the *ncs_cli* and check for devices live status.

~$ sudo -i

ncs_cli -C -u admin

 # show devices device *device_name* **live-status**

Verify that the command output reflects the correct 'state' and 'card-state' of each card.

Example output:

 # show devices device tb1-autovnf1_vpc-vpc-core-cf-nc live-status

```
<snip>

 VNFC  CURRENT  VNFC  VDU   CARD  CARD  NUMBER  CPU   DISK  START  UPTIME  NOVA  ID    DATE  FROM  TO
 REF   STATE    INSTANCE REF  TYPE  SLOT  OF      UTILIZATION SPACE TIME        LAUNCH      AND   STATE STATE
              ID                  ID    CORES                                 CMD         TIME
 ------------------------------------------------------------------------------------------------------
 cf1   -      cf1   cf    cf-dihdd 1     -       -     -      -       -       -
```

```
 cf2  -    cf2  cf    [illegible] 2    -    -    -    -    -    -

 sf1  -    sf1  sf    [illegible] 3    -    -    -    -    -    -

 sf2  -    sf2  sf    [illegible] 4    -    -    -    -    -    -


live-status vnfd sj-autovnf-vpc-abc

 version            6.0

 vnfm vim-tenant-name abc

 vnfm tenant-name abc

 vnfm ipaddr     192.168.100.22

 vnfm port       830

 vnfm username   ubuntu

 vnfm password   "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmkn1tRe1f84byNakWEa9sJ8sY/
cwfFME3aG0UaBC\nvvNNAMkuXQI9Ksfu5IiQQ9ViWbbHwl6IEFQ="

 virtual-link vl-di-internal1

  auto-vnf-connection-ref di-internal1

 virtual-link vl-management

  auto-vnf-connection-ref management

 virtual-link vl-orchestration

  auto-vnf-connection-ref orchestration

 virtual-link vl-abc-vpc-svc

  auto-vnf-connection-ref sj-autovnf-abc-vpc-svc

 vdu cf

  ssh-keygen         false

  vm-image           076c887a-a12c-4a0b-b4d6-b2d213f64b9e

  lifecycle-event-initialization staros_config.txt

   source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt


  lifecycle-event-initialization staros_param.cfg

   source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg


  ned cisco-staros-nc

   user        "$4$+HLzsE1kLJOeuoFWyOsmBWY2LHjOi2WtJdKy/OIux7YHhsNY/
O8hnA9/WwWuFD5trHrW3ZHs\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="

   password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNSJumC7n5rnkVxwI1s\
```

```
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="
   netconf
   port-number 830
  card-type          control-function
  usp-auto-vnf-id    sj-autovnf-vpc-abc-cf
  vnfc cf-vnfc-ugp
<snip>
```

Restoring the Previous Software Image

If for some reason you need to undo the upgrade, perform the upgrade again except:

- Specify the locations of the upgrade software image and configuration files.

then

- Specify the locations of the original software image and configuration files.

Automated StarOS Bin Upgrade

This section describes the automated StarOS bin upgrade procedure. For the manual bin upgrade procedure, see the Upgrading the Operating System Software, on page 9 section.

Prior to initiating the StarOS software upgrade procedure, make sure the conditions described in the Prerequisites section are met.

To automate the StarOS software upgrade:

1. Download the StarOS CF QCOW2 Image, on page 33
2. Verify Zookeeper Database, on page 13
3. Verify ESC Database, on page 14
4. Verify Free Space on the /flash Device, on page 14
5. Save the Running Configuration, on page 18
6. Initiate the StarOS Upgrade, on page 37

Prerequisites

Prior to performing an upgrade of StarOS software containing CF and SF VNFCs, check if the following prerequisites are met:

- You'll need the login credentials and IP address of AutoDeploy, AutoVNF, ESC, UEM, and CF VMs. You should have administrative rights to the OpenStack setup.

- Verify the OpenStack status. The Ansible output should all pass.

```
cd /home/stack/
source stackrc
cd /home/stack/ansible/
ansible-playbook -i inventory openstack_verify.yml
```

- Check if the health of AutoVNF/ESC/EM/VNF VM is normal through the UltraM health logs on AutoIT. If any of the VM(s) are not normal, then take necessary actions to rcover the health of the corresponding VM(s).
- You should have the new StarOS binary image file (for manual upgrade) or new StarOS CF QCOW2 image (for automated upgrade).
- Ensure that there are no pending transactions between ESC, UEM and CF.
- Be sure to take a backup of the original StarOS bin file.

Note Taking backup is not necessary for the automated bin upgrade as this will be automatically taken care through the upgrade RPC.

Download the StarOS CF QCOW2 Image

This section provides instructions that are applicable only to the upgrade of CF VNFCs.

Access to the Cisco support site and download facility is username and password controlled. You must have an active customer account to access the site and download the StarOS image.

The image files required to deploy the UGP are distributed as part of an RPM bundle. The bundle is called *usp-ugp-bundle-<version>.x86_64.rpm* and it is distributed as part of the USP ISO image.

For UGP-based VNF, perform the following steps to download the new StarOS binary image file to AutoVNF or OSPD VM.

1. Download the USP ISO bundle for the required release from the Cisco support site.
2. Extract the UGP image files from the USP ISO. The UGP RPM bundle contains *qvpc-di-<version>.qcow2.tgz*.

   rpm2cpio /var/usp-iso/repo/usp-ugp-bundle-*<version>***-1.x86_64.rpm | cpio -idmv ./opt/cisco/usp/bundles/ugp-bundle/qvpc-di-***<version>***.qcow2.tgz**
3. Ensure that the images have been extracted.

   ls -l ./opt/cisco/usp/bundles/ugp-bundle/qvpc-di-*<version>***.qcow2.tgz**
4. Extract the CF qcow2 image.

   tar -zxvf qvpc-di-*<version>***.qcow2.tgz qvpc-di-cf-***<version>***.qcow2**

Verify Zookeeper Database

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To verify the zookeeper database:

1. Log on to the AutoVNF using the floating IP.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

   ssh ubuntu@10.225.202.94

2. Log on to the UEM VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.21

3. Become the *root* user.

   sudo -i

4. Collect the UEM orchestration IP address for Zookeeper database connection.

   #ifconfig

   ```
   eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

             inet addr:209.165.200.240  Bcast: 209.165.200.255  Mask: 255.255.255.224
   ```

5. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* directory.

6. Execute the following script from the command line to access the UEM Zookeeper database.

   zkCli.sh -server *ip_addr:port_num*

   For example:

   zkCli.sh -server 209.165.200.240:2181

7. Check the zookeeper database and ensure that there are no pending requests between UEM and CF VMs.

   ls /request

   Example output:

   ```
   []

   <Ctrl+D to exit Zookeeper shell>
   ```

Verify ESC Database

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To verify the ESC database:

1. Log on to the AutoVNF using the floating IP.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

```
ssh ubuntu@10.225.202.94
```

2. Log on to the ESC VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

   ssh admin@<*vip-addr*>

   Command example:

   ```
   ssh admin@192.168.100.22
   ```

3. Check the ESC database to ensure there are no pending transactions.

   ```
   sudo /opt/cisco/esc/pgsql/bin/psql -U esc -p 7878 -h localhost -c
   'select * from esc_schema.workitem';
   ```

   ```
   config_id | request_id | mo_type | config_action | config_state
   ----------+------------+---------+---------------+--------------
   (0 rows)
   ```

4. Execute the following command to check the transaction details.

   ```
   escadm ip_trans
   ```

   Example output:

   ```
   Number of in-progress transaction events = 0
   ```

Verify Free Space on the /flash Device

Verify that there is enough free space on the **/flash** device to accommodate the new StarOS image file.

To verify the available space on *flash* directory:

1. Log on to the CF VM using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10.

   ssh ubuntu@<*vip-addr*>

   Command example:

   ```
   ssh ubuntu@192.168.100.38
   ```

2. Enter the following Exec mode command:

   [local]*host_name*# **directory /flash**

   The following is an example of the type of directory information displayed:

   ```
   -rwxrwxr-x    1  root     root         7334 May    5 17:29 asr-config.cfg
   -rwxrwxr-x    1  root     root         399 Jun    7 18:32 system.cfg
   -rwxrwxr-x    1  root     root         10667 May 14 16:24 testconfig.cfg
   -rwxrwxr-x    1  root     root         10667 Jun    1 11:21 testconfig_4.cfg
   -rwxrwxr-x    1  root     root         5926 Apr    7 16:27  tworpcontext.cfg
   -rwxrwxr-x    1  root     root         15534 Aug    4 13:31 test_vlan.cfg
   -rwxrwxr-x    1  root     root         2482 Nov 18 11:09 gateway2.cfg
   -rwxrwxr-x    1  root     root         159106048 Dec 31  2011 image_filename
   1136352  /flash
   Filesystem           1k-blocks      Used Available Use% Mounted on
   /var/run/storage/flash/part1 3115468   1136352  30018336   4%
   /mnt/user/.auto/onboard/flash
   ```

Note the "Available" blocks in the last line of the display. After displaying the directory information, the CLI returns to root and the following prompt appears:

```
[local]host_name#
```

Save the Running Configuration

Save the currently running, upgraded configuration prior to rebooting the chassis.

Important For the automated upgrade of StarOS binary image, perform only the Steps from 1 through 3 in the following procedure.

To save the boot configuration:

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 10 .

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.38

2. *Optional*. Execute the following command in the Exec mode.

   chassis key value 1234

   ```
   Save config before reload chassis, EVEN IF the same old key value is used.
   Old config scripts will become invalid after reload.
   ```

   Important This step is optional, and needed only if the chassis key is not set.

3. Save the boot configuration in the flash directory.

   save configuration /flash/system.cfg

   ```
   Warning: About to overwrite boot configuration file
   Are you sure? [Yes|No]: yes
   ```

   This will update the boot configuration to use the new bin image.

   Use the following command to check the boot configuration.

   # show boot

   ```
   Monday May 21 20:39:57 UTC 2018

   boot system priority 8 \
       image /flash/sftp/production.YYYYY.qvpc-di.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 9 \
       image /flash/staros.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 10 \
   ```

```
    image /flash/staros.bin \
    config /flash/system.cfg
```

4. Enter the configuration mode to change the boot priority of new StarOS bin file.

   #config

   #boot system priority 1 image /flash/updated.bin config /flash/system.cfg

   #end

5. Verify the new boot priority.

   #show boot

```
boot system priority 1 \

    image /flash/updated.bin \

    config /flash/system.cfg

boot system priority 10 \

    image /flash/staros.bin \

    config /flash/system.cfg
```

6. Verify whether the flash directory contains the boot configuration and new bin.

   dir /flash

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

Initiate the StarOS Upgrade

StarOS upgrades are automated through the RPC executed from the ConfD CLI or a NETCONF API.

Via the CLI

To perform an upgrade using the CLI, log on to AutoVNF as the ConfD CLI *admin* user and execute the following command:

vnfm-staros-bin-upgrade nsd-id *<nsd_id>* **qcow-url** *<qcow_url>* **[vnfid** *<vnfid>* **]**

NOTES:

- <*nsd_id*> is the name of the network service descriptor (NSD).
- <*qcow_url*> is the fully qualified path to StarOS CF QCOW file from the installation ISO image. Typically, the name of the file is *qvpc-di-cf.qcow2* or *qvpc-di-cf_T.qcow2*.
- <*vnfid*> is the optional ID of the StarOS VNF to upgrade. That is, it represents the VPC VNF ID as known to the UEM.

Note This ID must be obtained from the UEM.

If this ID is not specified, all the StarOS VNFs in the deployment are upgraded. This ID can be retrieved by executing **show running vnfdservice:vnfd** under ncs_cli on the master UEM.

- The upgrade could take several minutes to complete. The transaction status indicates whether the operation is successful or not. To monitor the progress or failures, check the *autovnf.log* in the */var/log/upstart/* directory.

Via the NETCONF API

Operation: nsd:vnfm-staros-bin-upgrade

Namespace: xmlns:nsd="http://www.cisco.com/usp/nfv/usp-nsds"

Parameters:

Parameter Name	Required	Type	Description
nsd-id	M	string	NSD name
qcow-url	M	string	Path to qcow file
vnfid	O	string	ID of StarOS device to upgrade

NOTES:

- <*nsd-id*> is the name of the network service descriptor (NSD).
- <*qcow-url*> is the fully qualified path to StarOS CF qcow file from the installation ISO image. Typically, the name of the file is *qvpc-di-cf.qcow2* or *qvpc-di-cf_T.qcow2*.
- <*vnfid*> is the StarOS VNF identifier on the UEM. This ID is optional. If not specified, all the StarOS VNFs in the deployment are upgraded.

Note This ID must be obtained from the UEM.

- The upgrade could take several minutes to complete. The transaction status indicates whether the operation is successful or not. To monitor the progress or failures, check the *autovnf.log* in the */var/log/upstart/* directory.

Performing Dynamic Software Updates

This section describes the dynamic software update (DSU) process that can be used to incrementally update plugins without having to update StarOS and reload the system.

Overview

StarOS allows the runtime loading of plugins. All StarOS builds include a "default" baseline plugin.

This feature is currently used to dynamically update the detection logic used to filter P2P applications.

Patching is the process used to install a plugin as an incremental update to a StarOS release. One plugin can be provided to multiple, compatible, concurrent product releases. A plugin is distributed in the form of a compressed distribution kit via the internet or by other means (USB stick, CD, etc.).

A plugin is a functional software entity that provides incremental updates to a pre-existing StarOS software component. Plugins have the characteristic of being dynamically loadable at runtime and do not require a system restart. A plugin has a name and one or more versions. All plugin names are known to the system at product release.

A plugin module is a specific instance of a plugin version consisting of at least one file that can be added to a running, in-service system. The module contains the information or instructions for a specific component's incremental update. Typically this would be a single file.

The Version Priority List (VPL) is a linked list of module versions associated with a specific plugin. Each plugin has one VPL. The list is sorted in ascending order by the priority number that is assigned by the administrator. When updating, the lowest priority number is tried first. If that version is not successful, the version in the VPL with the next sequentially greater priority number is tried. This list is iterated until a successful version is found. The VPL also supports manual rollback to a previous version (higher priority number) via a CLI command.

The basic sequence for the dynamic software process is as follows:

-
-
-
-
-
-

Downloading the Patch Kit

The Exec mode **patch plugin** command copies a patch intended for a specific plugin onto the running system.

The plugin kit includes a compressed plugin file (.tgz extension) and a certificate file (.cert extension). The command syntax is as follows:

patch plugin *name* **[http | ftp | sftp]://***host***/***directory***/***file* **certificate [http | ftp | sftp]://***host***/***directory***/***file*

For example:

```
[local]asr5k# patch plugin p2p http://192.168.1.2/tmp/patch_libp2p-1.17.4343.tgz certificate
 http://192.168.1.2/tmp/patch_libp2p-1.17.4343.cert
New patch for plugin p2p available for installation
```

The patch kit is copied to the **/flash/patch/***module-name* directory on the system.

Unpacking the Patch Kit

The Exec mode **install patch** command unpacks a patch kit and validates its contents. This command does not actually distribute the contents to the packet processing card **/var/opt** directories

The command syntax is as follows:

install plugin *plugin-name patch-file-name*

For example:

```
[local]host_name# install patch plugin p2p patch_libp2p-1.17.4343.tgz
  Install patch file patch_libp2p-1.17.4343.tgz [Yes/No]&quest;
  Install module p2p version 1.17.4343 successful
```

When a new plugin module is installed, the patch file is checked against the certificate which is provided with the patch file for authenticity of contents. If the certificate matches the patch file, the contents are extracted to the **/flash/install/***plugin-name***/contents** directory.

Once the contents are extracted, the Plugin Manager looks for the file **patch.bom** which contains a list of files which are considered a complete set for the patch. In addition, for each file listed, the.bom file contains an intended final destination directory where the content file should reside at the end of the installation.

For each file listed in the.bom file, the installation procedure copies the specified file to the destination directory. Once this has completed, the installation procedure is finished. The primary MIO/UMIO/MIO2 is responsible for ensuring that the **/flash** partitions on the standby MIO/UMIO/MIO2 are up-to-date.

Configuring the Plugin

The Global Configuration mode **plugin** command specifies a plugin to be configured and enters the Plugin Configuration mode. The plugin name must match the name of a plugin which has been copied to and unpacked on the system or an error message is displayed.

The command syntax is as follows:

plugin *plugin-name*
  module priority *number* **version** *module_version*

For example:

```
[local]host_name# configure
[local]host_name(config)# plugin p2p
[local]host_name(plugin-p2p)# module priority 1 version 1.17.4343
```

If a module priority number (1 through 100) was specified in an earlier configuration, that number is preserved. If the user does not explicitly replace or remove a module priority number, that number and the version associated with it will remain configured.

The Plugin Configuration mode also allows a user to enter values for any attribute value pairs (AVPs) associated with the plugin, as well as set the priority for this version of the plugin in the VPL.

Loading the Plugin

The Exec mode **update module** command loads the specified module with the lowest priority number from the VPL.

The command syntax is as follows:

update module *plugin-name*

For example:

```
[local]host_name# update module p2p
Update to module p2p version 1.17.4343 successful
```

The **update module** command takes the configured VPL (/flash/module.sys) and copies it into the internal data structures within the Plugin Manager. The VPL determines the sequence of module versions to try to load for each of the cards.

Current subscriber sessions will not be updated to the latest plugin version. New subscriber sessions will be affected by the newly updated plugin version.

Rolling Back to a Previous Plugin Version

The Exec mode **rollback module** command loads a module with the next higher priority number from the VPL.

The command syntax is as follows:

rollback module *plugin-name*

For example:

```
[local]host_name# rollback module p2p
Rollback to module p2p version 1.17.4340 successful
```

Current subscriber sessions will continue to be affected by the previous plugin version. New subscriber sessions will be affected by the newly downgraded plugin version.

DSU show Commands

There are two **show** commands associated with the dynamic software update process – **show module** and **show plugin**.

The **show module** command displays the current status of the internal VPL. The command syntax is as follows:

show module [*plugin-name***]**

For example:

```
[local]asr5# show module p2p
Module p2p
Priority version    loaded location                  time         status
    1    1.17.4343      no  /var/opt/lib    07/04/2012:05:06:59  success
    2    1.17.4340      yes  /var/opt/lib    03/06/2012:04:05:53  success
    3    1.17.4335      no  /var/opt/lib    03/05/2012:03:04:35  success
    4     1.17.4328     no  /var/opt/lib    03/04/2012:02:03:42  failed
    X    1.17.4289      no  /var/opt/lib    01/12/2012:05:44:22  success
```

Important Priority "X" identifies the "default" module that shipped with the StarOS build.

The **show plugin** command displays the current configuration of the plugin – AVPs and version priorities. The command syntax is as follows:

show plugin [*plugin-name***]**

For example:

```
[local]host_name# show plugin p2p
  plugin p2p
    module priority 1 version 1.17.4343
    module priority 2 version 1.17.4340
    module priority 3 version 1.17.4328
    module priority 4 version 1.17.4340
    module priority X version 1.17.4289
```

Managing License Keys

License keys define capacity limits (number of allowed subscriber sessions) and available features on your system. Adding new license keys allows you to increase capacity and add new features as your subscriber base grows.

New System License Keys

New systems are delivered with no license keys installed. In most cases, you receive the license key in electronic format (usually through e-mail).

When a system boots with no license key installed a default set of restricted session use and feature licenses is installed. The following Exec Mode command lists the license information:

```
[local]host_name# show license information
```

Important With no license key installed, the session use licenses for PDSN, HA, GGSN, and L2TP LNS are limited to 10,000 sessions.

Session Use and Feature Use Licenses

Session use and feature use licenses are software mechanisms that provide session limit controls and enable special features within the system. These electronic licenses are stored in the system's configuration file that is loaded as part of the system software each time the system is powered on or restarted.

- Session use licenses limit the number of concurrent sessions that a system is capable of supporting per service type and are acquired on an as-needed basis. This allows carriers to pay only for what they are using and easily increase capacity as their subscriber base grows.
- Feature use licenses enable specific features/functionality within the system and are distributed based on the total number of sessions supported by the system.

Installing New License Keys

Use the instructions below to install a new license key.

Cutting and Pasting the Key

If you have a copy of the license, use the following configuration to cut and paste just the license key part:

Procedure

Step 1 From the Exec mode, enter the following:

```
configure
   license key license
   exit
```

license is the license key string. The license can be an alphanumeric string of 1 through 1023 characters that is case sensitive. Copy the license key as shown in the example below, including the "\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\
STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|
LSP=000000|LSH=000000|LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|
FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/
XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

Step 2 Verify that the license key just entered was accepted by entering the following command at the Exec mode prompt:

```
[local]host_name# show license key
```

The new license key should be displayed. If it is not, return to the Global configuration mode and re-enter the key using the **license key** command.

Important An invalid license will not be accepted. A Failure error will appear in the output of the **license key** command when you attempt to configure an invalid license key. If you use the **-force** option to install an invalid license key, the license will be placed into a 30-day grace period. StarOS will generate daily syslog error messages and SNMP traps during the grace period. The output of the **show license information** command will indicate "License State" as "Not Valid".

Step 3 Verify that the license key enabled the correct functionality by entering the following command:

```
[local]host_name# show license information
```

All license keys and the new session capacity or functionality enabled should be listed. If the functionality or session capacity enabled by the new key is incorrect, please contact your service representative.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Caution Failure to save the new license key configuration in the current CLI configuration file will result in the loss of any of the new features enabled by the license key once the system is reloaded.

Adding License Keys to Configuration Files

License keys can be added to a new or existing configuration file.

Important License key information is maintained as part of the CLI configuration. Each time a key is installed or updated, you must re-save the configuration file.

Procedure

Step 1 Open the configuration file to which the new license key commands are to be copied.

Step 2 Copy the license as shown in the example, including the "\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\STCB21M82003R80411A4|
DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|LSP=000000|LSH=000000|
LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|
SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

Step 3 Paste the license key into the configuration

Important Paste the license key information at the beginning of the configuration file to ensure the system has the expected capacity and features before it configures contexts.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

License Expiration Behavior

When a license expires, there is a built-in grace period of 30 days that allows normal use of the licensed session use and feature use licenses. This allows you to obtain a new license without any interruption of service.

The following Exec mode command lists the license information including the date the grace period is set to expire:

`show license information`

Requesting License Keys

License keys for the system can be obtained through your Cisco account representative. Specific information is required before a license key may be generated:

- Sales Order or Purchase Order information
- Desired session capacity
- Desired functionality

Viewing License Information

To see the license detail, enter the following command from the Exec mode:

```
[local]host_name# show license information [ full | key [ full ] ]
```

Activating New License Keys

To activate new license keys, you must reboot the active CF by exciting the Exec mode **reload** command.

Note All SFs are rebooted when the **reload** command is executed on the active CF.

Deleting a License Key

Use the procedure below to delete the session and feature use license key from a configuration. You must be a security administrator or administrator.

```
configure
   no license key
   exit
show license key
```

The output of this command should display: "No license key installed".

Managing Local-User Administrative Accounts

Unlike context-level administrative accounts which are configured via a configuration file, information for local-user administrative accounts is maintained in a separate file in flash memory and managed through the software's Shared Configuration Task (SCT). Because local-user accounts were designed to be compliant with ANSI T1.276-2003, the system provides a number of mechanisms for managing these types of administrative user accounts.

For additional information, see Disable AAA-based Authentication for Console and Limit local-user Login on Console/vty Lines.

Configuring Local-User Password Properties

Local-user account password properties are configured globally and apply to all local-user accounts. The system supports the configuration of the following password properties:

- **Complexity:** Password complexity can be forced to be compliant with ANSI T1.276-2003.
- **History length:** How many previous password versions should be tracked by the system.
- **Maximum age:** How long a user can use the same password.
- **Minimum number of characters to change:** How many characters must be changed in the password during a reset.

- **Minimum change interval:** How often a user can change their password.
- **Minimum length:** The minimum number of characters a valid password must contain.
- **Expiry warning:** Password expiry warning interval in days.
- **Auto-generate:** Automatically generates password with option to specify length of password.

Refer to the **local-user password** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details on each of the above parameters.

Configuring Local-User Account Management Properties

Local-user account management includes configuring account lockouts and user suspensions.

Local-User Account Lockouts

Local-user accounts can be administratively locked for the following reasons:

- **Login failures:** The configured maximum login failure threshold has been reached. Refer to the **local-user max-failed-logins** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details
- **Password Aging:** The configured maximum password age has been reached. Refer to the **local-user password** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details.

Accounts that are locked out are inaccessible to the user until either the configured lockout time is reached (refer to the **local-user lockout-time** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference*) or a security administrator clears the lockout (refer to the **clear local-user** command in the *Exec Mode Commands* chapter of the *Command Line Interface Reference*).

Important Local-user administrative user accounts could be configured to enforce or reject lockouts. Refer to the **local-user username** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details.

Local-User Account Suspensions

Local-user accounts can be suspended as follows:

```
configure
  suspend local-user name
```

A suspension can be removed by entering:

```
configure
  no suspend local-user name
```

Changing Local-User Passwords

Local-user administrative users can change their passwords using the **password change** command in the Exec mode. Users are prompted to enter their current and new passwords.

Security administrators can reset passwords for local-users by entering the following command from the root prompt in the Exec mode:

`[local]host_name# password change username name`

name is the name of the local-user account for which the password is to be changed. When a security administrator resets a local-user's password, the system prompts the user to change their password the next time they login.

All new passwords must adhere to the password properties configured for the system.

Resetting, Stopping, Starting or Deleting VMs in the VPC-DI Instance

You can create a script that with reset, stop, start or delete one or more VMs in the VPC-DI instance using the hypervisor.

KVM

You can create a KVM script that runs a series of commands to reset, stop, start, or delete specified VMs.

The actual script will contain command sequences appropriate to the installation requirements of the VPC-DI instance as deployed in your site.

Script Commands

Available commands include:

```
 reset
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh reset {VMNAME}
done

 stop
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh destroy {VMNAME}
done

 start
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh start {VMNAME}
done

 delete
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh undefine {VMNAME}
done
```

Sample Script

A sample script is provided below:

```
cat ./vm-start

 start local
for CARD in {01,02}-cf {03,04,05,06,17,18,19,20}-sf do
  VMNAME=BLADE5-DI-{CARD}
  virsh start {VMNAME}
done

 start remote
ssh testbed3 '/home/luser/vm-start-BLADE5-DI'

roottestc1b5:/home/luser cat ./vm-stop

 reset local
for CARD in {01,02}-cf {03,04,05,06,17,18,19,20}-sf do
  VMNAME=BLADE5-DI-{CARD}
  virsh destroy {VMNAME}
done

 reset remote
ssh testbed3 '/home/luser/vm-stop-BLADE5-DI'
```

VMware ESXi

VSphere GUI

To reset VMs in a VPC-DI instance you can use the vSphere GUI.

Procedure

Step 1 From vSphere select the host for the target VPC-DI VMs.

Step 2 Select **Power Off** from the Commands list to shut down the VM(s).

Step 3 Select **Power On** from the Commands list to start the VM(s).

Step 4 If multiple vSphere hosts are used to provision the VMs in this VPC-DI instance, repeat steps 2 and 3 for each host.

PowerShell Script

You can also create a PowerShell script that uses the commands described in XREF to restart target VMs.