



# Managing Backup and Restore Operations

The following topics describe Cisco NFVI management node backup and restore operations.

- [Managing Backup and Restore Operations, on page 1](#)
- [Backing Up VIM UM, on page 3](#)
- [Restoring the Management Node, on page 7](#)
- [Management Node Autobackup, on page 9](#)
- [Managing Software Distribution Servers \(SDS\), on page 10](#)

## Managing Backup and Restore Operations

The management node hosts critical services such as Cisco VIM REST API, Cobbler for PXE, ELK for Logging/Kibana dashboard, and VMTP for the cloud validation in Cisco VIM.

The management node is not redundant during the initial Cisco VIM offering, hence it is recommended to take backup of the management node. Using the saved management node information, you can restore the management node if you are facing any issues with the platform.

## Backing Up the Management Node

An administrator must maintain the number of backup snapshots on the management node. The backup of the management node is possible only after complete deployment of at least one Cisco VIM. Two copies of backup folders are maintained at the management node itself and the older copy will be overwritten when a next backup is performed.

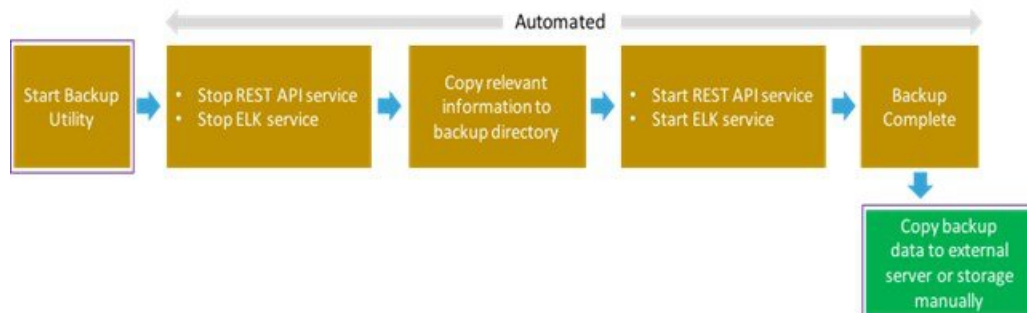
During the backup operation, activities such as pod management, software update or upgrade, and addition or deletion or replacement of nodes cannot be performed.

The REST API and ELK services are stopped during the backup operation, the OpenStack Logs are cached on the control, compute, and storage nodes until the restoration of the management node is completed.

As part of the backup operation, two files are created: `.backup_files` and `.backup_hash`. `.backup_files` is a list of files that are backed up, while the second one is the hash. These two files are placed under the backup folder `/var/cisco/backup_<tag>_<date-time>` at the management node and also at the `/var/cisco/` folder of all three controllers. These two files are used during the restore validation. When you attempt to restore from a particular backup, these two files within this backup are compared to the files that are kept in the controllers. If there is any discrepancy, the restore validation fails and you are prompted to either terminate the restore operation or continue despite the validation failure. Only one copy of the `.backup_files` and `.backup_hash` are

kept at the controllers, that is every time a new backup is created, these two files are overwritten with the most recent ones. Hence the restore validation passes only when the latest backup is used for restore.

**Figure 1: Cisco NFVI Management Node Backup Operation**



### Before you begin

- Save the management node information (for example, IP address of the management node) for use during the restore operation.
- Ensure that you have the br\_mgmt and br\_api IP addresses and respective network information.

**Step 1** Launch a SSH session to the Cisco NFVI management node.

**Step 2** Navigate to the <installer-ws>/tools/mgmt/ directory.

**Step 3** Execute `mgmt_node_backup.py`.

### What to do next

The backup operation takes approximately 30 minutes and creates the backup\_<tag>\_<date-time> directory in the /var/cisco/ path.

Copy the directory to a remote server to recover the management node using rsync.

For example, to copy the backup directory to the remote server 20.0.0.5 /var/cisco/directory , execute the following command sequence:

```
rsync -e ssh -go -rtvpX --numeric-ids /var/cisco/
```



**Note** On the remote server, protect the backup directory for any unauthorized access as the backup files may contain sensitive information

At the remote server, change directory to where the backup directory is copied to; in this example /var/cisco/backup\_<tag>\_<date-time>/.

To verify if the backup is not corrupted or modified, execute `./check_integrity`.

Check\_integrity depends on the following packages, the packages are installed on the server where check\_integrity is executed.

```
python-prettytable
python-jinja2
python-babel
python-markupsafe
python-setuptools
pytz
```

## Backup with Forwarding ELK Logs to External Syslog Server

When the feature Forwarding ELK Logs to External Syslog Server is enabled, during the backup process, in both the autobackup and manual backup, the ELK Logs are not collected. For manual backups, you can override by appending the `-a` or `--add-elk` option to the backup command. The `-s` or `--skip-elk` option is to skip the ELK Logs collection regardless of the forwarding feature is enabled or not.

```
# cd installer/tools/mgmt
# ./mgmt_node_backup.py --help
Usage: ./mgmt_node_backup.py [options]
Options:
  -h, --help            show this help message and exit
  -s, --skip-elk        do not collect ELK logs during backup
  -a, --add-elk         force to also collect ELK logs on backup
```

## Backing Up VIM UM

Administrator maintains the backup for Unified Management on the management node. The backup of the Unified Management is done only after the complete deployment of the Unified Management bootstrap. Only two copies of backup directory are maintained at the management node. The older copy is overwritten when a next Unified Management backup or autobackup takes place.

Unified Management backup is stored at the default backup location `/var/cisco/insight_backup/insight_backup_<release_tag>_<date>_<time>`. If you want to take a backup of Unified Management at a different location use `-backupdir/-b` option from `bootstrap_insight`; details of which are provided later in this section.

Unified Management UI triggers an autobackup whenever it detects an operation relating to MySQL database entry to preserve the latest state of Unified Management.




---

**Note** Unified Management backup is not allowed after an update. Update is an intermediate stage between rollback and commit. Any change that is made relating to MySQL database entry after an update from UM UI is not backed up.

---

## Autobackup Unified Management

If there is a change, Unified Management Installation automatically run a daemon process to take the autobackup.

Live status of the process is determined by checking the log located at `"/var/log/insight/insight_autobackup/insight_autobackup.logs"` or `systemctl status insight-autobackup`.



**Note** Max of 10-log files of size 1024\*1024 are maintained in the directory.

Following are the scenarios where autobackup is initiated:

Unified Management Operation	Auto-backup Performed
Adding or Deleting POD	Yes
Changing POD REST Password and Certificate	Yes
Add/Edit/Delete all types of users	Yes
Add/Edit/Delete Roles	Yes
Modify User and Role association	Yes
Revoking or Adding user permission	Yes
Log in or Logout	No
Context Switching	No
Change User Password	Yes

**Step 1** To check the status of the Unified Management perform the following steps:

```
systemctl status insight-autobackup
insight-autobackup.service - Insight Autobackup Service
  Loaded: loaded (/usr/lib/systemd/system/insight-autobackup.service; enabled; vendor preset:
disabled)
  Active: active (running) since Wed 2017-08-30 01:17:18 PDT; 19s ago
  Main PID: 19949 (python)
  Memory: 12.4M
  CGroup: /system.slice/insight-autobackup.service
          └─19949 /usr/bin/python /root/<installer-tag>/insight/playbooks/../../insight_autobackup.py
```

**Step 2** To stop Unified Management autobackup do the following:

```
systemctl stop insight-autobackup
insight-autobackup.service - Insight Autobackup Service
  Loaded: loaded (/usr/lib/systemd/system/insight-autobackup.service; enabled; vendor preset:
disabled)
  Active: inactive (dead) since Mon 2017-09-04 00:43:43 PDT; 5s ago
  Process: 19993 ExecStop=/bin/kill ${MAINPID} (code=exited, status=0/SUCCESS)
  Main PID: 19984
  Memory: 56.0K
  CGroup: /system.slice/insight-autobackup.service
```

**Step 3** The following are the steps to start Unified Management autobackup:

```
systemctl start insight-autobackup
insight-autobackup.service - Insight Autobackup Service
  Loaded: loaded (/usr/lib/systemd/system/insight-autobackup.service; enabled; vendor preset:
disabled)
  Active: active (running) since Wed 2017-08-30 01:17:18 PDT; 19s ago
  Main PID: 19949 (python)
  Memory: 12.4M
```

```
CGroup: /system.slice/insight-autobackup.service
└─19949 /usr/bin/python /root/<installer-tag>/insight/playbooks/../../insight_autobackup.py
```

**Step 4** The way Unified Management works is as follows:

#### 1. Install

- As soon as galera db and Unified Management containers are up the script will be invoked.
- Log dir : tailf /var/log/insight/insight\_autobackup\_logs/insight\_autobackup.log.
- It has a 10-seconds pulse which tells if the service is up or not.
  - [2017-09-04 00:49:01,504] INFO [Insight Autobackup] Insight Autobackup Service Running.
  - [2017-09-04 00:49:11,514] INFO [Insight Autobackup] Insight Autobackup Service Running.
  - [2017-09-04 00:49:21,525] INFO [Insight Autobackup] Insight Autobackup Service Running.
- If there is any change it takes a backup (time to check Sql diff is 30 seconds).
- It creates "rbac\_latest.sql" and "insight\_latest.tar.gz" and dump in the latest backup dir.
- During restore the bootstrap script checks if "rbac\_latest.sql" or "insight\_latest.tar.gz" is present in the backup dir.

#### 2. Update

- During update bootstrap insight does not support backup.
- Autobackup service would be terminated and no backup would be maintained in the intermediate state.

#### 3. Rollback

- Script are invoked again from the previous workspace.

#### 4. Commit

- Script are invoked again from the new workspace.

#### 5. Uninstall

- Service files are deleted.
- Log directory remains as the same.

---

## Back Up Unified Management at Default Back Up Location

---

**Step 1** Launch an SSH session to Cisco Unified Management management node and follow steps:

```
# cd <insight-ws>
#./bootstrap_insight.py -help

usage: bootstrap_insight.py [-h] --action ACTION
                           [--regenerate_secrets] [--setpassword]
                           [--file INSIGHTSETUPDATA] [--keep] [--verbose]
```

```

                                [--backupdir BACKUPDIR] [-y]

Insight install setup helper.
optional arguments:
  -h, --help                show this help message and exit
  --action ACTION, -a ACTION
                            install - Install Insight UI
                            install-status - Display Insight Install Status
reconfigure - Reconfigure Insight DB password or TLS                Certificate
                            update - Update Insight UI
                            update-status - Display Insight Update Status
                            rollback - Rollback Insight UI update
                            commit - Commit Insight UI update
                            backup - Backup Insight UI
                            uninstall - Uninstall Insight UI
  --regenerate_secrets, -r
                            System generated INSIGHT_DB_PASSWORD
  --setpassword, -s         User supplied INSIGHT_DB_PASSWORD,
  --file INSIGHTSETUPDATA, -f INSIGHTSETUPDATA
                            Location of insight_setup_data.yaml
  --keep, -k                Preserve Insight artifacts during uninstall
  --verbose, -v             Verbose on/off
  --backupdir BACKUPDIR, -b BACKUPDIR
                            Path to backup Insight
  -y, --yes                 Option to skip reconfigure or uninstall steps without prompt

```

## Step 2 Run the bootstrap command to view the Cisco VIM Unified Management backup details:

```

# ./bootstrap_insight.py -a backup
VIM Insight backup logs are at: /var/log/insight/<bootstrap_insight_<date>_<time>.log

Cisco VIM Insight backup Info!
+-----+-----+-----+-----+
| Description          | Status| Details
+-----+-----+-----+-----+
| Insight backup Status| PASS  | Backup done @
|                      |       | /var/cisco/insight_backup/insight_backup_<release_tag>_<date_time>|
+-----+-----+-----+-----+
Done with VIM Insight backup!

```

# Backup Unified Management at user defined backup location

## Step 1 Launch a SSH session to Cisco Unified Management management node and follow the below steps:

```

# cd <insight-ws>
# ./bootstrap_insight.py -help
usage: bootstrap_insight.py [-h] --action ACTION
                            [--regenerate_secrets] [--setpassword]
                            [--file INSIGHTSETUPDATA] [--keep] [--verbose]
                            [--backupdir BACKUPDIR] [-y]

Insight install setup helper.
optional arguments:
  -h, --help                show this help message and exit
  --action ACTION, -a ACTION

```

```

install - Install Insight UI
install-status - Display Insight Install Status
reconfigure - Reconfigure Insight DB password or TLS Certificate
update - Update Insight UI
update-status - Display Insight Update Status
rollback - Rollback Insight UI update
commit - Commit Insight UI update
backup - Backup Insight UI
uninstall - Uninstall Insight UI

--regenerate_secrets, -r
System generated INSIGHT_DB_PASSWORD
--setpassword, -s User supplied INSIGHT_DB_PASSWORD,
--file INSIGHTSETUPDATA, -f INSIGHTSETUPDATA
Location of insight_setup_data.yaml
--keep, -k Preserve Insight artifacts during uninstall
--verbose, -v Verbose on/off
--backupdir BACKUPDIR, -b BACKUPDIR
Path to backup Insight
-y, --yes Option to skip reconfigure or uninstall steps without prompt
    
```

**Step 2** Run the following command to view the Cisco VIM Unified Management backup details:

```

# ./bootstrap_insight.py -a backup --backupdir <user_defined_path>
VIM Insight backup logs are at: /var/log/insight/<bootstrap_insight_<date>_<time>.log
    
```

```

Cisco VIM Insight backup Info!
-----+-----+-----+
| Description          | Status | Details                               |
+-----+-----+-----+
| Insight backup Status | PASS   | Backup done @ <user_defined_path>    |
+-----+-----+-----+
Done with VIM Insight backup!
    
```

**What to do next**

Copy the backup directory to a remote server using rsync to recover the Insight later. We recommend you to copy backup dir using rsync as it preserves the permissions of the files.

For example, to copy the backup directory to the remote server 20.0.0.5 /var/cisco/insight\_backup/directory, execute the following command sequence: .

```

rsync -e ssh -rtvpX --numeric-ids
/var/cisco/insight_backup/insight_backup_2.1.5_2017-01-09_14-04-38
root@20.0.0.5:/var/cisco/insight_backup/
    
```

On the remote server, protect the backup directory for any unauthorized access, as the backup files may contain sensitive information

# Restoring the Management Node

As an administrator, you have to reimage the management node with the same ISO version when the backup is performed, before initiating the restore operation. Restore fails when there is a version mismatch.

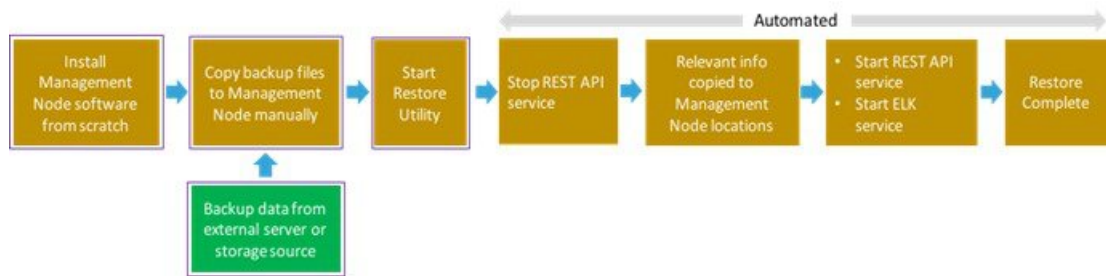


**Note** Version checking is available only for offline installation.

As part of the restore operation, system checks for the management node's IP address information to match the prior configuration. Logs are cached on the control, compute, and storage nodes from the moment of the management node fails until its restoration.

If you are using Cisco VIM Unified Management (in Tech Preview), in the same management node, you have to rebootstrap it for installation. During installation, RBAC and Pod registration information is lost, hence it is advised to make a note of the RBAC and Pod information.

**Figure 2: Cisco NFVI Management Node Restore Operation**



### Before you begin

Ensure that you have the `br_mgmt` and `br_api` IP addresses of the failed management node.

- Step 1** Reimage the management node with the ISO version with which you want to restore the node, and with the same IP address that is used before the failure of the node.
- Step 2** Navigate to `/var/cisco/directory` at the remote server where the backup folder is copied during the backup operation. Execute `./check_integrity` to verify if the backup is not corrupted or modified.
- Step 3** Copy the backup file to the `/var/cisco/directory` of the reimaged management node.
- For example, to copy the backup folder from the remote host `20.0.0.5` to the management node `/var/cisco/directory`, execute the following command sequence:

```
rsync -e ssh -rtvpX --numeric-ids root@20.0.0.5:/var/cisco/backup_2017-01-09_14-04-38 /var/cisco/
```

- Step 4** Navigate to the backup folder and execute the following command to verify if the backup is not corrupted or modified.

```
# cd /var/cisco/backup_<date-time>
# ./check-integrity
```

- Step 5** In `/var/cisco/backup_<date-time>` folder, execute the following command:

```
/var/cisco/backup_<date-time> # ./restore
```

The restore operation takes around 45 minutes.

- Step 6** Before restoration, the restore script performs validation of the backup folder. If validation fails, restore operation is halted and an error message is displayed. The script also verifies the last performed backup folder in the Management Node, and if any defects are detected, the you does confirm to proceed with restore operation.

```
...
2017-02-02 21:25:23 INFO Starting Cisco VIM restore...
```



```

2017-02-02 21:25:23 INFO Cisco VIM restore: estimated run time is approx. 45 mins...
2017-02-02 21:25:23 INFO Please see progress log for restore at
/var/log/mercury/installer/restore_2017-02-02_21:25:23.log
2017-02-02 21:25:27 ERROR Error: Backup id is not the one expected
Error: Found hashID file only in controller(s): j10-controller-2, j10-controller-3
Management backup files are ok (as per j10-controller-2)
Management backup files are ok (as per j10-controller-3)
The management node changed after the last backup was stored. Do you still want to proceed restoring
this management node? [Y/n] y
2017-02-02 22:17:55 INFO Workspace restored to /root/installer-6518
2017-02-02 22:17:55 INFO Cisco VIM restore: Executing restore playbook ...
2017-02-02 22:18:47 INFO Cisco VIM restore: Executing bootstrap playbook ...

```

**Note** The default behavior is to continue by keying **Return** or **Y**. Keying **N** terminates the restore operation.

```

...
2017-02-02 21:25:23 INFO Starting Cisco VIM restore...
2017-02-02 21:25:23 INFO Cisco VIM restore: estimated run time is approx. 45 mins...
2017-02-02 21:25:23 INFO Please see progress log for restore at
/var/log/mercury/installer/restore_2017-02-02_21:25:23.log
2017-02-02 21:25:27 ERROR Error: Backup id is not the one expected
Error: Found hashID file only in controller(s): j10-controller-2, j10-controller-3
Management backup files are ok (as per j10-controller-2)
Management backup files are ok (as per j10-controller-3)
The management node changed after the last backup was stored. Do you still want to proceed restoring
this management node? [Y/n] n
Aborting the restore operation as per user request

```

Once, restore operation ends, several health check points are automatically executed and the summary of results for that particular cloud availability is displayed.

**Step 7** Run the following checks manually to verify the status of the restore:

- Check the status of the REST API server:

```

# cd installer-<tagid>/tools
# ./restapi.py -a status
Status of the REST API Server: active (running) since Thu 2016-08-18 09:15:39 UTC; 9h ago
REST API launch directory: /root/installer-<tagid>/

```

- Check the setup\_data and runtime consistency of the management node:

```

# cd installer-<tagid>/; ciscovim run --perform 1,3 -y

```

- Execute the cloud sanity command:

```

# cd installer-<tagid>/tools
# ./cloud_sanity.py -c all

```

## Management Node Autobackup

After the successful completion of certain Pod management operations, a backup of the management node is performed automatically. Only one copy of the autobackup folder is kept at /var/cisco/ at any given time. Directory format for the autobackup\_<tag>\_<timestamp>.

Following are the list of operations:

- Fresh install of Cisco VIM

- Commit an update
- Replace controller
- Add or Remove compute nodes
- Add or Remove the storage node
- Reconfigure
- NFVIMON

Enabling or disabling the variable autobackup, is defined in the setup\_data.yaml file. It is enabled by default.

Add the following setup-data.yaml file snippet:

```
#####
# AutoBackup configuration
#####
#Default is True
#autobackup: True or False
```

The following tables shows when an auto-backup is performed during update or rollback or commit.

POD operation	Autobackup performed
Update	No
Rollback	No
Commit	Yes
Update fail with auto rollback	No

After creating a successful autobackup folder, you can copy it to an external server for later restoration as mentioned in [Restoring the Management Node, on page 7](#).

During the autobackup, if **Forwarding ELK Logs to the External Syslog server** option is enabled, the Elasticsearch database will not be maintained and the ELK Logs will not be recovered after restoring the management node.

## Managing Software Distribution Servers (SDS)

Software Distribution Servers (SDS) helps mitigate the need to ship USBs across different pods during installing or updating Cisco VIM. To ensure the long-term viability of SDS, it is designed to handle the following Day 2 scenarios:

- [Updating SDS TLS Certificate and Registry Credentials, on page 11](#)
- [SDS Server Backup and Restore, on page 11](#)
- [Resolving Low Disk Space, on page 13](#)
- [Manually Updating Packages, on page 14](#)

## Updating SDS TLS Certificate and Registry Credentials

Before installing the release artifacts from the SDS server, you must provide a valid TLS certificate and SDS registry credentials in the `sds_setup_data.yaml` file. Taking into account the security policies of an organization, SDS allows you to update the TLS certificate and registry credentials on the SDS server as required.

**Step 1** Navigate to the last installed release workspace using the `ls -lrt` command.

**Step 2** Replace the TLS certificate in the `openstack-configs` directory.

**Step 3** Modify the credentials in the `sds_setup_data.yaml` file.

**Step 4** Run the following command for the changes to take effect:

```
# cd /root/cvim_sds-<last-tag> # directory of last installed release and execute the following command.  
# ./sds_runner/runner.py
```

This operation validates the changes in the `sds_setup_data.yaml` file and the new TLS certificate. It reconfigures the SDS server components with this new information.

**Note** The SDS registry credentials of the pods that rely on SDS are also reconfigured.

## SDS Server Backup and Restore

SDS triggers an autobackup operation when a new Cisco VIM release is installed on the SDS server. It takes a backup of the relevant files from the SDS server, and saves it in the following location on the SDS server:

```
directory /var/cisco/autobackup_<tag>_<date-time>
```

Only the latest two backup directories are maintained on SDS. The older copy is overwritten when the next autobackup operation is triggered. If you want to use an older backup directory for a restore operation later, you need to save it to another location before it is overwritten. You can use the `rsync` or `scp` commands to save it to an RHEL7/CentOS based system, which is outside the SDS server.

## Checking Integrity of Autobackup Files

You can use the script provided in the autobackup directory to check the integrity of the autobackup files after using the `rsync` or `scp` commands.

### Before you begin

Ensure that the following packages are installed on the backup server using `yum`:

- `python-prettytable`
- `python-jinja2`
- `python-babel`
- `python-markupsafe`
- `python-setuptools`
- `pytz`

- 
- Step 1** Navigate to the autobackup directory.
- Step 2** Execute the following command to run the script:
- ```
# ./check_integrity
```
- 

## Restoring SDS from Backup

An SDS restore operation is usually performed when the original SDS server is being replaced by a new one.

---

- Step 1** Re-image the SDS server with the ISO version with which you want to restore the node, and with the same IP address that is used before the failure of the node.
- Step 2** Navigate to the location where the backup directory is copied during the backup operation.
- Step 3** Verify the integrity of the backup files as described in [Checking Integrity of Autobackup Files, on page 11](#).
- Step 4** Copy the backup file to the directory of the re-imaged SDS node.

For example, you can copy the backup directory from the remote host 20.0.0.5 to the SDS node directory `/var/cisco/` as follows:

```
rsync -e ssh -rtvpX --numeric-ids root@20.0.0.5:/var/cisco/autobackup_2017-01-09_14-04-38 /var/cisco/
```

- Step 5** Navigate to the backup directory and execute the following command to verify if the backup is not corrupted or modified.

```
# cd /var/cisco/autobackup_<tag>_<date-time>
# ./check-integrity
```

- Step 6** In the `/var/cisco/autobackup_<tag>_<date-time>` directory, execute the following commands:

```
# cd /var/cisco/backup_<date-time>
# ./restore
```

It may take about 45 minutes for the restore operation to complete.

**Note** Before restoring a backup directory, the restore script validates the backup directory. If the validation fails, the restore operation is interrupted and an error message is displayed. The restore script also verifies the latest backup directory in the SDS Node. If defects are detected, you need to confirm whether you want to proceed with the restore operation.

For example:

```
2017-02-02 21:25:23 INFO Starting Cisco VIM restore...
2017-02-02 21:25:23 INFO Cisco VIM restore: estimated run time is approx. 45 mins...
2017-02-02 21:25:23 INFO Please see progress log for restore at
/var/log/mercury/installer/restore_2017-02-02_21:25:23.log
2017-02-02 21:25:27 ERROR Error: Backup id is not the one expected
Error: Found hashID file only in controller(s): j10-controller-2, j10-controller-3 Management backup
files are ok (as per j10controller-2)
Management backup files are ok (as per j10-controller-3)
The management node changed after the last backup was stored. Do you still want to proceed restoring
this management node? [Y/n] y
2017-02-02 22:17:55 INFO Workspace restored to /root/installer-6518
2017-02-02 22:17:55 INFO Cisco VIM restore: Executing restore playbook ...
2017-02-02 22:18:47 INFO Cisco VIM restore: Executing bootstrap playbook ...
```

**Note** To continue the restore operation, you can press the **Enter** key or the **Y** key. If you want to abort the restore operation, you need to press the **N** key.

```

2017-02-02 21:25:23 INFO Starting Cisco VIM restore...
2017-02-02 21:25:23 INFO Cisco VIM restore: estimated run time is approx. 45 mins...
2017-02-02 21:25:23 INFO Please see progress log for restore at
/var/log/mercury/installer/restore_2017-02-02_21:25:23.log
2017-02-02 21:25:27 ERROR Error: Backup id is not the one expected
Error: Found hashID file only in controller(s): j10-controller-2, j10-controller-3 Management backup
files are ok (as per j10controller-2)
Management backup files are ok (as per j10-controller-3)
The management node changed after the last backup was stored. Do you still want to proceed restoring
this management node? [
Y/n] n
Aborting the restore operation as per user request

```

## Resolving Low Disk Space

Installing releases on SDS server is not allowed if the free disk space is less than 20%. Hence, a utility to remove docker images from the container registry running on the SDS server is provided. You can find the cleanup script at the following location:

```
/root/cvim_sds-<last-tag>/sds/registry_cleanup.py
```

Example of running the cleanup script:

```
# ./registry_cleanup.py -h
usage: registry_cleanup.py [-h] (--list | --delete DELETE | --unused_tags)
                        [-u USERNAME] [-p PASSWORD] [-r REGISTRY]
```

List/Delete image tags in the registry

optional arguments:

```

-h, --help                Show this help message and exit
--list                    List Image Tags in Registry
--delete DELETE          Delete Images of provided tags from registry
--unused_tags            List unused Tags in SDS registry
-u USERNAME, --username USERNAME
                        Registry Username
-p PASSWORD, --password PASSWORD
                        Registry Password
-r REGISTRY, --registry REGISTRY
                        Registry URL

```

The cleanup script requires three mandatory parameters, namely, Registry URL, Registry username, and Registry password. The script supports the following three options:

- **List Image Tags**—The option lists all the images and corresponding tags present in the docker registry.
- **Unused Tags**—This option lists all the releases present on the SDS server but are not used by any of the Cisco VIM pods. By default, the pods are registered with the SDS server. When a pod is installed, updated, roll backed, or upgraded, the release information is sent to SDS. You can use this command to identify the releases that can be safely removed from the SDS server.
- **Delete Tags**—You can specify the releases that you want to remove from the docker registry. The script removes these images and frees the disk space.

## Manually Updating Packages

SDS installs repositories inside docker containers so that all the packages to be installed are obtained from those repositories. These repositories are updated when you install a later version of Cisco VIM release on the SDS server. Once the repositories are updated, all the packages, except httpd package and its dependencies are updated. Updating httpd is deferred because when httpd is updated, all downstream connections are disrupted thereby requiring you to restart the Cisco VIM pod install.

To update httpd and its dependent packages, you can use the update script found in the tools directory. Ensure that you run this script during the maintenance phase so that non of the Cisco VIM pods are currently attempting to get artifacts from the SDS server.

Run the following command to execute the update script:

```
# cd /root/cvim_sds-<last-tag> # directory of last installed release and execute the following command.  
# ./update_httpd.sh
```