Red Hat Satellite 6.10

Upgrading and Updating Red Hat Satellite

Upgrading and updating Red Hat Satellite Server and Capsule Server

Last Updated: 2022-04-13
Abstract

This guide describes upgrading and updating Red Hat Satellite Server, Capsule Server, and hosts.
# Table of Contents

**CHAPTER 1. UPGRADE OVERVIEW**  ................................................................. 3  
1.1. PREREQUISITES  3  
1.2. UPGRADE PATHS  4  
1.3. FOLLOWING THE PROGRESS OF THE UPGRADE  5  
1.4. UPGRADING CAPSULES SEPARATELY FROM SATELLITE  5  

**CHAPTER 2. CLONING SATELLITE SERVER**  .................................................. 6  
2.1. CLONING PROCESS OVERVIEW  6  
2.2. PREREQUISITES  6  
2.3. PULP DATA CONSIDERATIONS  7  
2.4. CLONING SATELLITE SERVER  7  
   2.4.1. Preparing the source server for cloning  7  
   2.4.2. Cloning to the Target Server  8  

**CHAPTER 3. UPGRADING RED HAT SATELLITE**  .......................................... 10  
3.1. UPGRADING SATELLITE SERVER  10  
   3.1.1. Preparing to Migrate Content to Pulp 3  12  
   3.1.2. Upgrading a Connected Satellite Server  16  
   3.1.3. Upgrading a Disconnected Satellite Server  17  
3.2. SYNCHRONIZING THE NEW REPOSITORIES  22  
3.3. UPGRADING CAPSULE SERVERS  23  
3.4. UPGRADING SATELLITE CLIENTS  25  

**CHAPTER 4. POST-UPGRADE TASKS**  ......................................................... 28  
4.1. UPGRADING DISCOVERY  28  
   4.1.1. Upgrading Discovery on Satellite Server  28  
   4.1.2. Upgrading Discovery on Capsule Servers  28  
   4.1.3. Verifying Subnets have a Template Capsule  29  
4.2. UPGRADING VIRT-WHO  29  
4.3. REMOVING THE PREVIOUS VERSION OF THE SATELLITE TOOLS REPOSITORY  30  
4.4. RECLAIMING POSTGRESQL SPACE  30  
4.5. UPDATING TEMPLATES, PARAMETERS, LOOKUP KEYS AND VALUES  31  
4.6. TUNING SATELLITE SERVER WITH PREDEFINED PROFILES  31  

**CHAPTER 5. UPDATING SATELLITE SERVER, CAPSULE SERVER, AND CONTENT HOSTS**  .......... 34  
5.1. UPDATING SATELLITE SERVER  34  
5.2. UPDATING DISCONNECTED SATELLITE SERVER  35  
5.3. UPDATING CAPSULE SERVER  39
CHAPTER 1. UPGRADE OVERVIEW

This chapter details the prerequisites and available upgrade paths to Red Hat Satellite 6.10. Review this information before upgrading your current Red Hat Satellite 6 installation.

In this guide, the terms update, upgrade, and migrate have the following meanings:

Upgrading
The process of advancing your Satellite Server and Capsule Server installations from a y-stream release to the next, for example Satellite 6.9 to Satellite 6.10.

Updating
The process of advancing your Satellite Server and Capsule Server installations from a z-stream release to the next, for example Satellite 6.10.0 to Satellite 6.10.1.

Migrating
The process of moving an existing Satellite installation to another Red Hat Enterprise Linux server.

For interactive upgrade instructions, you can also use the Red Hat Satellite Upgrade Helper on the Red Hat Customer Portal. This application provides you with an exact guide to match your current version number. You can find instructions that are specific to your upgrade path, as well as steps to prevent known issues. For more information, see Satellite Upgrade Helper on the customer portal.

Note that you can upgrade Capsules separately from Satellite. For more information, see Section 1.4, “Upgrading Capsules Separately from Satellite”.

1.1. PREREQUISITES

Upgrading to Satellite 6.10 affects your entire Satellite infrastructure. Before proceeding, complete the following:

- Read the Release Notes.
- Review this guide so that you are aware of the upgrade process and its impact.
- Plan your upgrade path. For more information, see Section 1.2, “Upgrade Paths”.
- Plan for the required downtime. Satellite services are shut down during the upgrade. The upgrade process duration might vary depending on your hardware configuration, network speed, and the amount of data that is stored on the server. Upgrading Satellite takes approximately 1 - 2 hours.

Upgrading Capsule takes approximately 10 - 30 minutes.

However, upgrading from 6.9 to 6.10 also migrates Pulp content, this step can take some considerable time. For more information on preparing for Pulp migration and the upgrade process, see Section 3.1, “Upgrading Satellite Server”.

- Ensure that you have sufficient storage space on your server. For more information, see Preparing your Environment for Installation in Installing Satellite Server from a Connected Network and Preparing your Environment for Installation in Installing Capsule Server.
- Back up your Satellite Server and all Capsule Servers. For more information, see Backing Up Satellite Server and Capsule Server in the Administering Red Hat Satellite 6.9 guide.
- Plan for updating any scripts you use that contain Satellite API commands because some API
commands differ between versions of Satellite. For more information about changes in the API, see the Knowledgebase article API Changes Between Satellite Versions on the Red Hat Customer Portal.

**WARNING**

If you customize configuration files, manually or use a tool such as Hiera, these customizations are overwritten when the installation script runs during upgrading or updating. You can use the `--noop` option with the satellite-installer script to test for changes. For more information, see the Red Hat Knowledgebase solution How to use the noop option to check for changes in Satellite config files during an upgrade.

1.2. UPGRADE PATHS

You can upgrade to Red Hat Satellite 6.10 from Red Hat Satellite 6.9. Satellite Servers and Capsule Servers on earlier versions must first be upgraded to Satellite 6.9. For more details, see the Satellite 6.9 Upgrading and Updating Red Hat Satellite guide.

**Figure 1.1. Overview of Satellite 6.10 Upgrade Paths**

The high level steps in upgrading to Satellite 6.10 are as follows.

1. Clone your existing Satellite Servers. For more information, see Chapter 2, Cloning Satellite Server.

2. Upgrade Satellite Server and all Capsule Servers to Satellite 6.10. For more information, see Section 3.1, “Upgrading Satellite Server”.

**WARNING**

Upgrading from the Beta to GA version is not supported.
3. Upgrade to Satellite Tools 6.10 on all Satellite clients. For more information, see Section 3.4, “Upgrading Satellite Clients”.

Considerations for Upgrades of Satellite to Future Versions

Before you begin the upgrade to Satellite 6.10, which includes an upgrade to Pulp 3 on Satellite Servers, it is highly recommended that you complete pre-migration of Pulp content. If you perform no migrations on 6.9, they will all be performed during the upgrade to 6.10. For smaller content setups, or if you can afford longer downtimes, you can proceed without the pre-migration.

For Capsules, you can choose to deploy new 6.10 Capsules rather than to upgrade.

For future upgrades following Satellite 6.11, you will be required to upgrade the operating system from RHEL 7 to RHEL 8 on your Satellite Servers and Capsules. You can upgrade the operating system in-place or through a cloning process. The latter includes migration of all data, configuration, and synced content.

NOTE

If you are planning to avoid the upgrade from Pulp 2 to Pulp 3 and deploy a new Satellite 6.10 infrastructure due to the Pulp 3 changes instead, you might want to wait for Satellite 6.11 to deploy with RHEL 8 directly.

1.3. FOLLOWING THE PROGRESS OF THE UPGRADE

Because of the lengthy upgrade time, use a utility such as `screen` to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously. For more information about using the screen command, see How do I use the screen command? article in the Red Hat Knowledge Base. You can also see the `screen` manual page for more information.

If you lose connection to the command shell where the upgrade command is running you can see the logs in `/var/log/foreman-installer/satellite.log` to check if the process completed successfully.

1.4. UPGRADING CAPSULES SEPARATELY FROM SATELLITE

You can upgrade Satellite to version 6.10 and keep Capsules at version 6.9 until you have the capacity to upgrade them too.

All the functionality that worked previously works on 6.9 Capsules. However, the functionality added in the 6.10 release will not work until you upgrade Capsules to 6.10.

Upgrading Capsules after upgrading Satellite can be useful in the following example scenarios:

1. If you want to have several smaller outage windows instead of one larger window.

2. If Capsules in your organization are managed by several teams and are located in different locations.

3. If you use a load-balanced configuration, you can upgrade one load-balanced Capsule and keep other load-balanced Capsules at 1 version lower. This allows you to upgrade all Capsules one after another without any outage.
CHAPTER 2. CLONING SATELLITE SERVER

When you upgrade Satellite Server, you can optionally create and upgrade a clone of your Satellite to ensure that you do not lose any data while you upgrade. After your upgrade is complete, you can then decommission the earlier version of Satellite Server.

Use the following procedures to clone your Satellite instances to preserve your environments in preparation for upgrade.

The Satellite clone tool does not support migrating a Capsule Server to Red Hat Enterprise Linux 7. Instead you must backup the existing Capsule Server, restore it on Red Hat Enterprise Linux 7, then reconfigure Capsule Server.

Terminology

Ensure that you understand the following terms:

Source server: the server that you clone.

Target server: the new server that you copy files to and clone the source server to.

2.1. CLONING PROCESS OVERVIEW

1. Back up the source server.

2. Clone the source server to the target server.

3. Power off the source server.

4. Update the network configuration on the target server to match the target server’s IP address with its new host name.

5. Restart goferd in Content hosts and Capsules to refresh the connection.

6. Test the new target server.

2.2. PREREQUISITES

To clone Satellite Server, ensure that you have the following resources available:

- A minimal install of Red Hat Enterprise Linux 7 server to become the target server. Do not install Red Hat Enterprise Linux 7 software groups, or third-party applications. Ensure that your server complies with all the specifications of Preparing your Environment for Installation in Installing Satellite Server.

- A backup from Satellite 6.9 that you make using the satellite-maintain backup script. You can use a backup with or without Pulp data.

- A Satellite subscription for the target server.

Before you begin cloning, ensure the following conditions exist:

- The target server is on an isolated network. This avoids unwanted communication with Capsule Servers and hosts.

- The target server has the capacity to store all your backup files from the source server.
Customized configuration files
If you have any customized configurations on your source server that are not managed by the satellite-installer tool or Satellite backup process, you must manually back up these files.

2.3. PULP DATA CONSIDERATIONS
You can clone Satellite server without including Pulp data. However, for your cloned environment to work, you do require Pulp data. If the target server does not have Pulp data, it is not a fully working Satellite.

To transfer Pulp data to a target server, you have two options:

- Clone using backup with Pulp data
- Clone using backup without Pulp data and copy /var/lib/pulp manually from the source server.

If your pulp_data.tar file is greater than 500 GB, or if you use a slow storage system, such as NFS, and your pulp_data.tar file is greater than 100 GB, do not include pulp_data.tar in the backup because this can cause memory errors during extraction. Copy the pulp_data.tar file from the source server to the target server.

To back up without Pulp data
Follow the steps in the procedure in Section 2.4, “Cloning Satellite Server” and replace the steps that involve cloning with Pulp data with the following steps:

1. Perform a backup with PostgreSQL databases active excluding the Pulp data:
   
   ```
   # satellite-maintain backup offline --skip-pulp-content \
   --assumeyes /var/backup
   ```

2. Stop and disable the satellite-maintain services
   
   ```
   # satellite-maintain service stop
   # satellite-maintain service disable
   ```

3. Copy the Pulp data to the target server:
   
   ```
   # rsync --archive --partial --progress --compress \
   /var/lib/pulp target_server.example.com:/var/lib/pulp
   ```

Proceed to Section 2.4.2, “Cloning to the Target Server”.

2.4. CLONING SATELLITE SERVER
Use the following procedures to clone Satellite Server. Note that because of the high volume of data that you must copy and transfer as part of these procedures, it can take a significant amount of time to complete.

2.4.1. Preparing the source server for cloning
On the source server, complete the following steps:

1. Verify the Pool ID of your Satellite subscription:
2. Remove the Red Hat Satellite subscription:

```
# subscription-manager remove --serial=$(subscription-manager list --consumed --matches 'Red Hat Satellite' | grep "Serial:" | awk '{print $2}')
```

3. Determine the size of the Pulp data:

```
# du -sh /var/lib/pulp/
```

4. If you have less than 500 GB of Pulp data, perform a backup with PostgreSQL databases active including the Pulp data. If you have more than 500 GB of Pulp data, skip the following steps and complete the steps in Section 2.3, “Pulp Data Considerations” before you continue.

```
# satellite-maintain backup offline --assumeyes /var/backup
```

5. Stop and disable the `satellite-maintain` services:

```
# satellite-maintain service stop
# satellite-maintain service disable
```

Proceed to Section 2.4.2, “Cloning to the Target Server”.

### 2.4.2. Cloning to the Target Server

To clone your server, complete the following steps on your target server:

1. The `satellite-clone` tool defaults to using `/backup/` as the backup folder. If you copy to a different folder, update the `backup_dir` variable in the `/etc/satellite-clone/satellite-clone-vars.yml` file.

2. Place the backup files from the source Satellite in the `/backup/` folder on the target server. You can either mount the shared storage or copy the backup files to the `/backup/` folder on the target server.

3. Power off the source server.

4. Enter the following commands to register to the Customer Portal, attach subscriptions, and enable only the required subscriptions:

```
# subscription-manager register your_customer_portal_credentials
# subscription-manager attach --pool=pool_ID
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-server-rhscl-7-rpms
# subscription-manager repos --enable=rhel-7-server-satellite-maintenance-6-rpms
# subscription-manager repos --enable=rhel-7-server-satellite-6.9-rpms
```
5. Install the **satellite-clone** package:

   ```shell
   # yum install satellite-clone
   ```

   After you install the **satellite-clone** tool, you can adjust any configuration to suit your own deployment in the `/etc/satellite-clone/satellite-clone-vars.yml` file.

6. Run the **satellite-clone** tool:

   ```shell
   # satellite-clone
   ```

7. Reconfigure DHCP, DNS, TFTP, and remote execution services. The cloning process disables these services on the target Satellite Server to avoid conflict with the source Satellite Server.

8. Reconfigure and enable DHCP, DNS, and TFTP in the Satellite web UI. For more information, see [Configuring External Services on Satellite Server](#) in *Installing Satellite Server*.

9. Enable remote execution:

   ```shell
   # satellite-installer --scenario satellite \
   --enable-foreman-plugin-remote-execution \
   --enable-foreman-proxy-plugin-remote-execution-ssh
   ```

10. Log on to the Satellite web UI, with the username **admin** and the password **changeme**. Immediately update the admin password to secure credentials.

11. Ensure that the correct organization is selected.

12. Navigate to **Content > Subscriptions**, then click **Manage Manifest**.

13. Click the **Refresh** button, then click **Close** to return to the list of subscriptions.

14. Verify that the available subscriptions are correct.

15. Follow the instructions in the `/usr/share/satellite-clone/logs/reassociate_capsules.txt` file to restore the associations between Capsules and their lifecycle environments.

16. Update your network configuration, for example, DNS, to match the target server’s IP address with its new host name. The **satellite-clone** tool changes the host name to the source server’s host name. If you want to change the host name to something different, you can use the **satellite-change-hostname** tool. For more information, see [Renaming a Satellite or Capsule Server](#) in *Administrating Red Hat Satellite*.

17. If the source server uses the **virt-who** daemon, install and configure it on the target server. Copy all the **virt-who** configuration files in the `/etc/virt-who.d/` directory from the source server to the same directory on the target server. For more information, see [Configuring Virtual Machine Subscriptions in Red Hat Satellite](#). After you perform an upgrade using the following chapters, you can safely decommission the source server.
CHAPTER 3. UPGRADING RED HAT SATELLITE

WARNING
If you have Satellite 6 installed in a high availability configuration, contact Red Hat Support before upgrading to Satellite 6.10.

Use the following procedures to upgrade your existing Red Hat Satellite to Red Hat Satellite 6.10:

1. Section 3.1, “Upgrading Satellite Server”
2. Section 3.2, “Synchronizing the New Repositories”
3. Section 3.3, “Upgrading Capsule Servers”
4. Section 3.4, “Upgrading Satellite Clients”
5. Chapter 4, Post-Upgrade Tasks

Before upgrading, see Section 1.1, “Prerequisites”.

3.1. UPGRADING SATELLITE SERVER

This section describes how to upgrade Satellite Server from 6.9 to 6.10. You can upgrade from any minor version of Red Hat Satellite Server 6.9. Ensure that you have updated to the latest z-stream before upgrading to 6.10. For more information about updating Satellite Server, see Section 5.1, “Updating Satellite Server”.

Although the process of upgrading from 6.9 to 6.10 also migrates Pulp content, this can take some considerable time. You can reduce this time by following the procedure in Section 3.1.1, “Preparing to Migrate Content to Pulp 3”.

If you have not pre-migrated content from Pulp 2 to Pulp 3, you must upgrade to the latest version of Satellite Server before starting to migrate Pulp content.

Before You Begin

- Note that you can upgrade Capsules separately from Satellite. For more information, see Section 1.4, “Upgrading Capsules Separately from Satellite”.
- Review and update your firewall configuration prior to upgrading your Satellite Server. For more information, see Preparing your environment for installation in Installing Satellite Server.
- Ensure that you do not delete the manifest from the Customer Portal or in the Satellite web UI because this removes all the entitlements of your content hosts.
- Back up and remove all Foreman hooks before upgrading. Restore any hooks only after Satellite is known to be working after the upgrade is complete. Note that Foreman Hooks functionality is deprecated and will be removed in the next Satellite version.
If you have edited any of the default templates, back up the files either by cloning or exporting them. Cloning is the recommended method because that prevents them being overwritten in future updates or upgrades. To confirm if a template has been edited, you can view its History before you upgrade or view the changes in the audit log after an upgrade. In the Satellite web UI, navigate to Monitor > Audits and search for the template to see a record of changes made. If you use the export method, restore your changes by comparing the exported template and the default template, manually applying your changes.

- Pulp content migration occurs at 6.9, before the upgrade to 6.10.

- Ensure there is sufficient storage space for /var/lib/pulp/published to double in size before starting the migration process. Check the size of /var/lib/pulp/published with:

  ```bash
  # du -sh /var/lib/pulp/published/
  ```

- Content migration can be run online, but uses processor, disk, and memory resources. Sync and content view publishing operations might take longer as a result.

- If you have not pre-migrated Pulp content, the **PULP_CONTENT_PREMIGRATION_BATCH_SIZE** setting defines the number of content units processed at the same time. It affects the amount of RAM used and the I/O load. The lower the value, the longer **satellite-maintain content prepare** takes to complete. The upgrade downtime is also longer if you have content left to migrate at that point.
  
  - The default value is 1000.
  
  - If the system has a hard disk drive, or there are concerns about I/O load, the recommended value is 50.
  
  - A lower value is not recommended.

- Use the following method to set **PULP_CONTENT_PREMIGRATION_BATCH_SIZE**:

  1. Create a directory and file:

     ```bash
     $ mkdir /etc/systemd/system/pulpcore-worker@.service.d/
     $ vi /etc/systemd/system/pulpcore-worker@.service.d/settings.conf
     ```

     with contents:

     ```
     [Service]
     User=pulp
     Environment=PULP_CONTENT_PREMIGRATION_BATCH_SIZE=1000
     ```

  2. Restart the **satellite-maintain** services using:

     ```bash
     # systemctl daemon-reload
     # satellite-maintain service restart
     ```

     For further information, see [Preparing to Migrate Content to Pulp 3](#).

**Capsule Considerations**

- If you use Content Views to control updates to a Capsule Server’s base operating system, or for Capsule Server repository, you must publish updated versions of those Content Views.
Note that Satellite Server upgraded from 6.9 to 6.10 can use Capsule Servers still at 6.9.

WARNING

If you implemented custom certificates, you must retain the content of both the /root/ssl-build directory and the directory in which you created any source files associated with your custom certificates.

Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

Upgrade Scenarios

- To upgrade a Satellite Server connected to the Red Hat Content Delivery Network, proceed to Section 3.1.2, "Upgrading a Connected Satellite Server".
- To upgrade a Satellite Server not connected to the Red Hat Content Delivery Network, proceed to Section 3.1.3, "Upgrading a Disconnected Satellite Server".

You cannot upgrade a self-registered Satellite. You must migrate a self-registered Satellite to the Red Hat Content Delivery Network (CDN) and then perform the upgrade. To migrate a self-registered Satellite to the CDN, see Migrating Self-Registered Satellites in the Satellite 6.10 Upgrading and Updating Red Hat Satellite guide.

FIPS mode

You cannot upgrade Satellite Server from a RHEL base system that is not operating in FIPS mode to a RHEL base system that is operating in FIPS mode.

To run Satellite Server on a Red Hat Enterprise Linux base system operating in FIPS mode, you must install Satellite on a freshly provisioned RHEL base system operating in FIPS mode. For more information, see Preparing your environment for installation in Installing Satellite Server.

3.1.1. Preparing to Migrate Content to Pulp 3

The time to prepare content for Pulp 3 depends on the amount of content and the number of Content Views. For large systems, this can mean several days of downtime. To prevent this, pre-migrate Pulp content while running the latest version of Satellite Server 6.9. This reduces the overall upgrade downtime.

During migration to Pulp3, the amount of data stored in the /var/lib/pulp/published/ directory can double in size. Ensure that there is enough space on the /var/lib/pulp volume. If the user runs satellite-maintain prep-6.10-upgrade, the content in /var/lib/pulp/content does not need to be duplicated. After Pulp2 is removed, the extra space can be claimed back, however, shrinking of XFS volumes is not possible and copying data to a different partition might be necessary in order to bring volume size down to the previous value.

During migration to Pulp 3, the amount of data stored in the PostgreSQL data directory can grow significantly. Ensure that you have enough space for the migration. A rough estimation of the space you require is 2-3 times the size of the MongoDB Pulp 2 database.
Pre-Migration Checks

The sequence should always be if you have any Composite Content Views, take action on those first, and then work on the Content Views.

- Ensure that all enabled repositories complete the synchronization process.
- If any repositories are synced with a Warning or Error state, fix the underlying problem and resync the repository.
- Ensure that no paused tasks in the system are related to any repositories or Content Views.
- If you no longer require specific Content Views or Composite Content View Versions, you can perform a cleanup action.
- For more information about how to perform the cleanup, see How to remove old content view versions in Red Hat Satellite 6 using CLI/hammer?.
- If some Content View or Composite Content View versions are not required anymore, please consider performing a cleanup of them.
- For more information about how to perform the cleanup of older Content View or Composite Content View versions using hammercli, see How to remove old content view versions in Red Hat Satellite 6 using CLI/hammer?.
- If you no longer require or use certain repositories, disable those specific repositories.
- When the cleanup process finishes, ensure that you clean up the orphan data by following the steps in How to delete orphaned content in /var/lib/pulp on Capsule?.
- Ensure that you can enter the following command on Satellite without error:

  `# foreman-rake katello:correct_repositories COMMIT=true --trace`

Use this procedure to begin migrating content from Pulp 2 to Pulp 3.

Procedure

1. Update the file permissions before upgrading Satellite Server using the following command:

   `# satellite-maintain prep-6.10-upgrade`

   This might take some time on high latency systems.

2. View details of the content you are pre-migrating using the following command:

   `# satellite-maintain content migration-stats`

   Use this command as often as necessary during the migration process to determine how long the process will take. It also identifies corrupted or missing content that might cause the migration to fail. Output is similar to the following:

   `Running Retrieve Pulp 2 to Pulp 3 migration statistics
   ===============Retrieve Pulp 2 to Pulp 3 migration statistics:
   =========Migration Summary==============`
Migrated/Total RPMs: 0/367633  
Migrated/Total errata: 0/20780140  
Migrated/Total repositories: 0/33924  
Estimated migration time based on yum content: 47 hours, 23 minutes  
Note: ensure there is sufficient storage space for /var/lib/pulp/published to double in size before starting the migration process.
Check the size of /var/lib/pulp/published with `du -sh /var/lib/pulp/published/

3. Prepare your content for Pulp migration using the following command:

```
# satellite-maintain content prepare
```

As part of its final step, `satellite-maintain content prepare` checks whether any content units are unmigrated. If it identifies corrupted or missing content, you might see something similar to the following:

```
=============Missing/Corrupted Content Summary=================
WARNING: MISSING OR CORRUPTED CONTENT DETECTED
Corrupted or Missing Rpm: 1000/104583
Corrupted or missing content has been detected, you can examine the list of content in /tmp/unmigratable_content-20211025-74422-16cxfae and take action by either:
  1. Performing a 'Verify Checksum' sync under Advanced Sync Options, let it complete, and re-running the migration
  2. Deleting/disabling the affected repositories and running orphan cleanup (foreman-rake katello:delete_orphaned_content) and re-running the migration.
  3. Manually correcting files on the filesystem in /var/lib/pulp/content/ and re-running the migration.
  4. Mark currently corrupted or missing content as skipped (foreman-rake katello:approve_corrupted_migration_content). This will skip migration of missing or corrupted content.
```

There are several causes of corrupted or missing content:

- A Repository synchronization or Content View publish or promotion occurred during the migration.
- Files on disk in /var/lib/pulp/content/ became corrupted due to disk rot.
- Files in /var/lib/pulp/content/ are missing due to past events such as disk loss with a partial restore.
- There is some mismatch between subsystems in Satellite such as Katello and Pulp. This can happen if a repository or content view is improperly deleted by skipping steps in a paused Red Hat Satellite Task. Running `foreman-rake katello:correct_repositories COMMIT=true` can correct this.

You can review the list of corrupt or missing RPMs written to disk as part of the `satellite-maintain content migration-stats` command, `/tmp/unmigratable_content-20211025-74422-16cxfae20211120-2149-1j4pmfae` in the example above.

For more information about determining which repository the unmigratable contents belong to, see How to determine the repository to run Verify Checksum on for reported corrupted RPMs.
In most cases, where repositories with unmigratable contents are configured on Satellite with the **On-demand** download policy and the number of missing or corrupt RPMs are low, you can mark these as skipped using the command:

```
# foreman-rake katello:approve_corrupted_migration_content
```

If you approve corrupted or missing content and proceed with the upgrade, that content will not appear in the repositories or content view versions after upgrade.

If these packages exist in the upstream repositories, they are added again when you re-synchronize the repositories after upgrade. They are not restored to the Content Views unless you republish those Content Views. As these packages are likely unusable, the upgrade process only detects that fact.

If you suspect a Repository synchronization or Content View publish or promotion occurred during the migration, re-running `satellite-maintain content prepare` migrates the remaining content. If this is the first time you have run `satellite-maintain content prepare`, Red Hat recommends running it as often as necessary to try to reduce the number of corrupt or missing RPMs. Migrating the remaining content then takes less time.

**NOTE**

You cannot use **Ctrl + C** to terminate the `satellite-maintain content prepare` process. If you attempt to halt the process using **Ctrl + C** or by disconnecting your SSH session, the process does not terminate but continues in the background. You can use the following command to terminate the process gracefully, whenever necessary, so that you can continue later.

```
# satellite-maintain content prepare-abort
```

Note that `satellite-maintain content-prepare abort` can take several minutes to terminate the process. You can continue the migration process using `satellite-maintain content-prepare` whenever it is convenient.

4. The process does not confirm that migration is complete. You can determine how near to completion the process is by using the following command:

```
# satellite-maintain content migration-stats
```

at intervals until the indicated migration time is at or near zero.

5. The final steps of Pulp content migration are completed when upgrading Satellite Server from 6.9 to 6.10.

**NOTE**

If problems occur, you can restart the pre-migration process from the beginning using the following command:

```
# satellite-maintain content migration-reset
```
The final steps of Pulp content migration are completed when upgrading Satellite Server from 6.9 to 6.10.

### 3.1.2. Upgrading a Connected Satellite Server

Use this procedure for a Satellite Server with access to the public internet

---

**WARNING**

If you customize configuration files, manually or using a tool such as Hiera, these changes are overwritten when the installation script runs during upgrading or updating. You can use the `--noop` option with the satellite-installer script to test for changes. For more information, see the Red Hat Knowledgebase solution How to use the noop option to check for changes in Satellite config files during an upgrade.

---

**Upgrade Satellite Server**

1. Create a backup.
   - On a virtual machine, take a snapshot.
   - On a physical machine, create a backup.

   For more information about backups, see Backing Up Satellite Server and Capsule Server in the Administering Red Hat Satellite 6.9 guide.

2. Optional: If you made manual edits to DNS or DHCP configuration in the `/etc/zones.conf` or `/etc/dhcp/dhcpd.conf` files, back up the configuration files because the installer only supports one domain or subnet, and therefore restoring changes from these backups might be required.

3. Optional: If you made manual edits to DNS or DHCP configuration files and do not want to overwrite the changes, enter the following command:

   ```sh
   # satellite-installer --foreman-proxy-dns-managed=false \ 
   --foreman-proxy-dhcp-managed=false
   ```

4. In the Satellite web UI, navigate to Hosts > Discovered hosts. On the Discovered Hosts page, power off and then delete the discovered hosts. From the Select an Organization menu, select each organization in turn and repeat the process to power off and delete the discovered hosts. Make a note to reboot these hosts when the upgrade is complete.

5. Ensure that the `apache::purge_configs: false` entry is either not present or commented out in the `/etc/foreman-installer/custom-hiera.yaml` file of the Satellite 6.9/Capsule 6.9 servers which will be upgraded to 6.10.

6. Ensure that the Satellite Maintenance repository is enabled:

   ```sh
   # subscription-manager repos --enable \ 
   rhel-7-server-satellite-maintenance-6-rpms
   ```

7. Check the available versions to confirm the version you want is listed:
# satellite-maintain upgrade list-versions

8. Use the health check option to determine if the system is ready for upgrade. When prompted, enter the hammer admin user credentials to configure satellite-maintain with hammer credentials. These changes are applied to the /etc/foreman-maintain/foreman-maintain-hammer.yml file.

# satellite-maintain upgrade check --target-version 6.10

Review the results and address any highlighted error conditions before performing the upgrade.

9. Because of the lengthy upgrade time, use a utility such as screen to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously. For more information about using the screen command, see How do I use the screen command? article in the Red Hat Knowledge Base. If you lose connection to the command shell where the upgrade command is running you can see the logged messages in the /var/log/foreman-installer/satellite.log file to check if the process completed successfully.

10. Perform the upgrade:

   # satellite-maintain upgrade run --target-version 6.10

11. Check when the kernel packages were last updated:

    # rpm -qa --last | grep kernel

12. Optional: If a kernel update occurred since the last reboot, reboot the system:

    # reboot

13. If using a BASH shell, after a successful or failed upgrade, enter:

    # hash -d satellite-maintain service 2> /dev/null

14. If you have migrated content from Pulp 2 to Pulp 3, remove all Pulp 2 content.

    # satellite-maintain content remove-pulp2

   This removes Pulp 2 RPMs, content in /var/lib/pulp/content/, the mongo database, and migration content in the Pulp 3 database.

### 3.1.3. Upgrading a Disconnected Satellite Server

Use this procedure if your Satellite Server is not connected to the Red Hat Content Delivery Network.
WARNING

- If you customized configuration files, either manually or using a tool such as Hiera, these changes are overwritten when you enter the `satellite-maintain` command during upgrading or updating. You can use the `--noop` option with the `satellite-installer` command to review the changes that are applied during upgrading or updating. For more information, see the Red Hat Knowledgebase solution How to use the noop option to check for changes in Satellite config files during an upgrade.

- The hammer import and export commands have been replaced with `hammer content-import` and `hammer content-export` tooling. If you have scripts that are using `hammer content-view version export`, `hammer content-view version export-legacy`, `hammer repository export`, or their respective import commands, you have to adjust them to use the `hammer content-export` command instead, along with its respective import command.

- If you implemented custom certificates, you must retain the content of both the `/root/ssl-build` directory and the directory in which you created any source files associated with your custom certificates. Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

Before You Begin

- Review and update your firewall configuration before upgrading your Satellite Server. For more information, see Ports and Firewalls Requirements in Installing Satellite Server from a Disconnected Network.

- Ensure that you do not delete the manifest from the Customer Portal or in the Satellite Web UI because this removes all the entitlements of your content hosts.

- Back up and remove all Foreman hooks before upgrading. Reinstate hooks only after Satellite is known to be working after the upgrade is complete.

WARNING

If you implemented custom certificates, you must retain the content of both the `/root/ssl-build` directory and the directory in which you created any source files associated with your custom certificates.

Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.
Upgrade Disconnected Satellite Server

1. Create a backup.
   - On a virtual machine, take a snapshot.
   - On a physical machine, create a backup.

2. A pre-upgrade script is available to detect conflicts and list hosts which have duplicate entries in Satellite Server that can be unregistered and deleted after upgrade. In addition, it will detect hosts which are not assigned to an organization. If a host is listed under **Hosts > All hosts** without an organization association and if a content host with the same name has an organization already associated with it then the content host will automatically be unregistered. This can be avoided by associating such hosts to an organization before upgrading. Run the pre-upgrade check script to get a list of hosts that can be deleted after upgrading. If any unassociated hosts are found, associating them to an organization before upgrading is recommended.

   ```bash
   # foreman-rake katello:upgrade_check
   ```

3. Optional: If you made manual edits to DNS or DHCP configuration in the `/etc/zones.conf` or `/etc/dhcp/dhcpd.conf` files, back up the configuration files because the installer only supports one domain or subnet, and therefore restoring changes from these backups might be required.

4. Optional: If you made manual edits to DNS or DHCP configuration files and do not want to overwrite the changes, enter the following command:

   ```bash
   # satellite-installer --foreman-proxy-dns-managed=false \
   --foreman-proxy-dhcp-managed=false
   ```

5. Optional: If you use PostgreSQL as an external database, on the PostgreSQL server, install the `rh-postgresql12-postgresql-evr` package, which is available from the `rhel-7-server-satellite-6.10-rpms` repository:

   ```bash
   # yum install rh-postgresql12-postgresql-evr
   ```

6. In the Satellite web UI, navigate to **Hosts > Discovered hosts**. If there are discovered hosts available, turn them off and then delete all entries under the **Discovered hosts** page. Select all other organizations in turn using the organization setting menu and repeat this action as required. Reboot these hosts after the upgrade has completed.

7. Ensure that the `apache::purge_configs: false` entry is either not present or commented out in the `/etc/foreman-installer/custom-hiera.yaml` file of the Satellite 6.9/Capsule 6.9 servers which will be upgraded to 6.10.

8. Make sure all external Capsule Servers are assigned to an organization, otherwise they might get unregistered due to host-unification changes.

9. Remove old repositories:

   ```bash
   # rm /etc/yum.repos.d/*
   ```

10. Stop the `satellite-maintain` services.

    ```bash
    # satellite-maintain service stop
    ```
For more information with additional steps, see Section 5.2 Updating Disconnected Satellite Server.

1. Optional: If you have applied custom Apache server configurations, note that the custom configurations are reverted to the installation defaults when you perform the upgrade. To preview the changes that are applied during the upgrade, enter the satellite-installer command with the --noop (no operation) option. These changes are applied when you enter the satellite-maintain upgrade command in a following step.

   a. Add the following line to the /etc/httpd/conf/httpd.conf configuration file.

      ```
      Include /etc/httpd/conf.modules.d/*.conf
      ```

   b. Restart the httpd service.

      ```
      # systemctl restart httpd
      ```

   c. Start the postgresql and rh-mongodb34-mongod database services.

      ```
      # systemctl start postgresql
      # systemctl start rh-mongodb34-mongod
      ```

   d. Enter the satellite-installer command with the --noop option:

      ```
      # satellite-installer --scenario satellite --upgrade --verbose --noop
      ```

      Review the /var/log/foreman-installer/satellite.log to preview the changes that are applied during the upgrade. Locate the +++ and --- symbols that indicate the changes to the configurations files. Although entering the satellite-installer command with the --noop option does not apply any changes to your Satellite, some Puppet resources in the module expect changes to be applied and might display failure messages.

   e. Stop the satellite-maintain services:

      ```
      # satellite-maintain service stop
      ```

2. Because of the lengthy upgrade time, use a utility such as screen to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously. For more information about using the screen command, see How do I use the screen command? article in the Red Hat Knowledge Base. If you lose connection to the command shell where the upgrade command is running you can see the logs in /var/log/foreman-installer/satellite.log to check if the process completed successfully.

3. Check the available versions to confirm the version you want is listed:

   ```
   # satellite-maintain upgrade list-versions
   ```

4. Use the health check option to determine if the system is ready for upgrade. When prompted, enter the hammer admin user credentials to configure satellite-maintain with hammer credentials. These changes are applied to the /etc/foreman-maintain/foreman-maintain-hammer.yml file.

   ```
   # satellite-maintain upgrade check --target-version 6.10 \ 
   --whitelist="repositories-validate,repositories-setup"
   ```
Review the results and address any highlighted error conditions before performing the upgrade.

5. Perform the upgrade:

```shell
# satellite-maintain upgrade run --target-version 6.10 \ 
--whitelist="repositories-validate,repositories-setup"
```

**WARNING**

If you run the command from a directory containing a `config` subdirectory, you will encounter the following error:

```
ERROR: Scenario (config/satellite.yaml) was not found, can not continue.
```

In such a case, change directory, for example to the `root` user’s home directory, and run the command again.

If the script fails due to missing or outdated packages, you must download and install these separately. For more information, see the Resolving Package Dependency Errors section in the Installing Satellite Server from a Disconnected Network guide.

6. If using a BASH shell, after a successful or failed upgrade, enter:

```shell
# hash -d satellite-maintain service 2> /dev/null
```

7. If you have migrated content from Pulp 2 to Pulp 3, remove all Pulp 2 content:

```shell
# satellite-maintain content remove-pulp2
```

This removes uploaded and synchronized Pulp 2 packages, content in `/var/lib/pulp/content/`, MongoDB, and migration content in the Pulp 3 database.

8. Check when the kernel packages were last updated:

```shell
# rpm -qa --last | grep kernel
```

9. Optional: If a kernel update occurred since the last reboot, stop the `satellite-maintain` services and reboot the system:

```shell
# satellite-maintain service stop
# reboot
```

10. Optional: If you made manual edits to DNS or DHCP configuration files, check and restore any changes required to the DNS and DHCP configuration files using the backups that you made.

11. If you make changes in the previous step, restart the `satellite-maintain` services.
12. If you have the OpenSCAP plug-in installed, but do not have the default OpenSCAP content available, enter the following command.

```
# foreman-rake foreman_openscap:bulk_upload:default
```

13. In the Satellite web UI, go to Configure > Discovery Rules and associate selected organizations and locations with discovery rules.

### 3.2. SYNCHRONIZING THE NEW REPOSITORIES

You must enable and synchronize the new 6.10 repositories before you can upgrade Capsule Servers and Satellite clients.

**Procedure**

1. In the Satellite web UI, navigate to Content > Red Hat Repositories.

2. Toggle the Recommended Repositories switch to the On position.

3. From the list of results, expand the following repositories and click the Enable icon to enable the repositories:
   - To upgrade Satellite clients, enable the Red Hat Satellite Tools 6.10 repositories for all Red Hat Enterprise Linux versions that clients use.
   - If you have Capsule Servers, to upgrade them, enable the following repositories too:
     - Red Hat Satellite Capsule 6.10 (for RHEL 7 Server) (RPMs)
     - Red Hat Satellite Maintenance 6 (for RHEL 7 Server) (RPMs)
     - Red Hat Ansible Engine 2.9 RPMs for Red Hat Enterprise Linux 7 Server
     - Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server

   **NOTE**

   If the 6.10 repositories are not available, refresh the Subscription Manifest. Navigate to Content > Subscriptions, click Manage Manifest, then click Refresh.

4. Navigate to Content > Sync Status.

5. Click the arrow next to the product to view the available repositories.

6. Select the repositories for 6.10.

7. Click Synchronize Now.
IMPORTANT

If an error occurs when you try to synchronize a repository, refresh the manifest. If the problem persists, raise a support request. Do not delete the manifest from the Customer Portal or in the Satellite web UI; this removes all the entitlements of your content hosts.

8. If you use Content Views to control updates to the base operating system of Capsule Server, update those Content Views with new repositories, publish, and promote their updated versions. For more information, see Managing Content Views in the Content Management Guide.

3.3. UPGRADING CAPSULE SERVERS

This section describes how to upgrade Capsule Servers from 6.9 to 6.10.

Before You Begin

- You must upgrade Satellite Server before you can upgrade any Capsule Servers. Note that you can upgrade Capsules separately from Satellite. For more information, see Section 1.4, “Upgrading Capsules Separately from Satellite”.
- Ensure the Red Hat Satellite Capsule 6.10 repository is enabled in Satellite Server and synchronized.
- Ensure that you synchronize the required repositories on Satellite Server. For more information, see Section 3.2, “Synchronizing the New Repositories”.
- If you use Content Views to control updates to the base operating system of Capsule Server, update those Content Views with new repositories and publish their updated versions. For more information, see Managing Content Views in the Content Management Guide.
- Ensure the Capsule’s base system is registered to the newly upgraded Satellite Server.
- Ensure the Capsule has the correct organization and location settings in the newly upgraded Satellite Server.
- Review and update your firewall configuration prior to upgrading your Capsule Server. For more information, see Preparing Your Environment for Capsule Installation in Installing Capsule Server.

WARNING

If you implemented custom certificates, you must retain the content of both the /root/ssl-build directory and the directory in which you created any source files associated with your custom certificates.

Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.
Upgrading Capsule Servers

1. Create a backup.
   - On a virtual machine, take a snapshot.
   - On a physical machine, create a backup. For information on backups, see Backing Up Satellite Server and Capsule Server in the Administering Red Hat Satellite 6.9 guide.

2. Clean yum cache:
   ```
   # yum clean metadata
   ```

3. Ensure that the rubygem-foreman_maintain package that provides satellite-maintain is installed and up to date:
   ```
   # yum install rubygem-foreman_maintain
   ```

4. On Capsule Server, verify that the foreman_url setting points to the Satellite FQDN:
   ```
   # grep foreman_url /etc/foreman-proxy/settings.yml
   ```

5. Ensure that the apache::purge_configs: false entry is either not present or commented out in the /etc/foreman-installer/custom-hiera.yaml file of the Satellite 6.9/Capsule 6.9 servers which will be upgraded to 6.10.

6. Check the available versions to confirm the version you want is listed:
   ```
   # satellite-maintain upgrade list-versions
   ```

7. Because of the lengthy upgrade time, use a utility such as screen to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously. For more information about using the screen command, see How do I use the screen command? article in the Red Hat Knowledge Base.
   If you lose connection to the command shell where the upgrade command is running you can see the logged messages in the /var/log/foreman-installer/satellite.log file to check if the process completed successfully.

8. Use the health check option to determine if the system is ready for upgrade:
   ```
   # satellite-maintain upgrade check --target-version 6.10
   ```
   Review the results and address any highlighted error conditions before performing the upgrade.

9. Perform the upgrade:
   ```
   # satellite-maintain upgrade run --target-version 6.10
   ```
**WARNING**

If you run the command from a directory containing a `config` subdirectory, you will encounter the following error:

```
ERROR: Scenario (config/capsule.yaml) was not found, can not continue.
```

In such a case, change directory, for example to the `root` user’s home directory, and run the command again.

10. If you have migrated content from Pulp 2 to Pulp 3, remove all Pulp 2 content.

```
# satellite-maintain content remove-pulp2
```

This removes Pulp 2 RPMs, content in `/var/lib/pulp/content/`, the mongo database, and migration content in the Pulp 3 database.

11. Check when the kernel packages were last updated:

```
# rpm -qa --last | grep kernel
```

12. Optional: If a kernel update occurred since the last reboot, reboot the system:

```
# reboot
```

13. Optional: If you made manual edits to DNS or DHCP configuration files, check and restore any changes required to the DNS and DHCP configuration files using the backups made earlier.

14. Optional: If you use custom repositories, ensure that you enable these custom repositories after the upgrade completes.

15. On Satellite Server, perform a complete synchronization of the upgraded Capsule as the MongoDB and RPM repositories are not automatically migrated with Satellite.

```
# hammer capsule content synchronize --name $(Capsule) --skip-metadata-check true --async
```

### 3.4. UPGRADING SATELLITE CLIENTS

The Satellite Tools 6.10 repository provides `katello-agent` and `katello-host-tools`, which provide communication services for managing Errata.

**NOTE**

The Katello agent is deprecated and will be removed in a future Satellite version. Migrate your workloads to use the remote execution feature to update clients remotely. For more information, see [Migrating from Katello Agent to Remote Execution](Managing Hosts Guide) in the *Managing Hosts Guide*.  

---

*CHAPTER 3. UPGRADING RED HAT SATELLITE*
For deployments using **katello-agent** and **goferd**, update all clients to the new version of **katello-agent**. For deployments not using **katello-agent** and **goferd**, update all clients to the new version of **katello-host-tools**. Complete this action as soon as possible so that your clients are fully compatible with Satellite Server.

**Prerequisites**

- You must have upgraded Satellite Server.
- You must have enabled the new Satellite Tools 6.10 repositories on the Satellite.
- You must have synchronized the new repositories in the Satellite.
- If you have not previously installed **katello-agent** on your clients and you want to install it, use the manual method. For more information, see [Upgrade Satellite Clients Manually](#).

**WARNING**

If you implemented custom certificates, you must retain the content of both the `/root/ssl-build` directory and the directory in which you created any source files associated with your custom certificates.

Failure to retain these files during an upgrade causes the upgrade to fail. If these files have been deleted, they must be restored from a backup in order for the upgrade to proceed.

**Upgrade Satellite Clients Using the Bulk Repository Set UI:**

1. In the Satellite web UI, navigate to **Hosts > Content Hosts** and select the Content Hosts that you want to upgrade.
2. From the **Select Action** list, select **Manage Repository Sets**.
3. From the **Repository Sets Management** list, select the **Red Hat Satellite Tools 6.9** check box.
4. From the **Select Action** list, select **Override to Disabled**, and click **Done**.
5. When the process completes, on the same set of hosts from the previous steps, from the **Select Action** list, select **Manage Repository Sets**.
6. From the **Repository Sets Management** list, select the **Red Hat Satellite Tools 6.10** check box.
7. From the **Select Action** list, select **Override to Enabled**, and click **Done**.
8. When the process completes, on the same set of hosts from the previous steps, from the **Select Action** list, select **Manage Packages**.
9. In the **Package** search field, enter one of the following options depending on your configuration:
   - If your deployment uses **katello-agent** and **goferd**, enter **katello-agent**.
   - If your deployment does not use **katello-agent** and **goferd**, enter **katello-host-tools**.
10. Until BZ#1649764 is resolved, from the Update list, you must select via remote execution. This is required because if you update the package using the Katello agent, the package update disrupts the communication between the client and Satellite or Capsule Server, which causes the update to fail. For more information, see Configuring and Setting Up Remote Jobs in the Managing Hosts guide.

Upgrade Satellite Clients Manually

1. Log into the client system.

2. Disable the repositories for the previous version of Satellite.

   ```
   # subscription-manager repos \
   --disable rhel-7-server-satellite-tools-6.9-rpms
   ```

3. Enable the Satellite Tools 6.10 repository for this version of Satellite.

   ```
   # subscription-manager repos \
   --enable=rhel-7-server-satellite-tools-6.10-rpms
   ```

4. Depending on your configuration, complete one of the following steps:

   - If your deployment uses katello-agent and goferd, enter the following command to install or upgrade katello-agent:
     ```
     # yum install katello-agent
     ```

   - If your deployment does not use katello-agent and goferd, enter the following command to install or upgrade katello-host-tools:
     ```
     # yum install katello-host-tools
     ```
CHAPTER 4. POST-UPGRADE TASKS

Some of the procedures in this section are optional. You can choose to perform only those procedures that are relevant to your installation.

If you use the PXE-based discovery process, then you must complete the discovery upgrade procedure on Satellite and on any Capsule Server with hosts that you want to be listed in Satellite on the Hosts > Discovered hosts page.

4.1. UPGRAдинG DISCOVERY

This section describes updating the PXELinux template and the boot image passed to hosts that use PXE booting to register themselves with Satellite Server.

From Satellite 6.10, provisioning templates now have a separate association with a subnet, and do not default to using the TFTP Capsule for that subnet. If you create subnets after the upgrade, you must specifically enable the Satellite or a Capsule to provide a proxy service for discovery templates and then configure all subnets with discovered hosts to use a specific template Capsule.

During the upgrade, for every subnet with a TFTP proxy enabled, the template Capsule is set to be the same as the TFTP Capsule. After the upgrade, check all subnets to verify this was set correctly.

These procedures are not required if you do not use PXE booting of hosts to enable Satellite to discover new hosts.

4.1.1. Upgrading Discovery on Satellite Server

1. Update the Discovery template in the Satellite web UI:
   a. Navigate to Hosts > Provisioning templates.
   b. On the PXELinux global default line, click Clone.
   c. Enter a new name for the template in the Name field, for example ACME PXE global default.
   d. In the template editor field, change the line ONTIMEOUT local to ONTIMEOUT discovery and click Submit.
   e. Navigate to Administer > Settings.
   f. Locate Global default PXELinux template and click on its Value.
   g. Select the name of the newly created template from the menu and click the tick button.
   h. Navigate to Hosts > Provisioning templates.
   i. Click Build PXE Default, then click OK.

2. In the Satellite web UI, go to Configure > Discovery Rules and associate selected organizations and locations with discovery rules.

4.1.2. Upgrading Discovery on Capsule Servers

1. Verify that the Foreman Discovery package is current on Satellite Server.
2. If an update occurred in the previous step, restart the satellite-maintain services.

3. Upgrade the Discovery image on the Satellite Capsule that is either connected to the provisioning network with discovered hosts or provides TFTP services for discovered hosts.

4. On the same instance, install the package which provides the Proxy service, and then restart foreman-proxy service.

5. In the Satellite web UI, go to Infrastructure > Capsules and verify that the relevant Capsule lists Discovery in the features column. Select Refresh from the Actions drop-down menu if necessary.

6. Go to Infrastructure > Subnets and for each subnet on which you want to use discovery:
   a. Click the subnet name.
   b. On the Capsules tab, ensure the Discovery Capsule is set to a Capsule you configured above.

### 4.1.3. Verifying Subnets have a Template Capsule

Ensure all subnets with discovered hosts have a template Capsule:

1. In the Satellite web UI, navigate to Infrastructure > Subnets.
2. Select the subnet you want to check.
3. On the Capsules tab, ensure a Template Capsule has been set for this subnet.

For more information about configuring subnets with template Capsules, see Configuring the Discovery Service in the Provisioning Guide

### 4.2. UPGRADING VIRT-WHO

If virt-who is installed on Satellite Server or a Capsule Server, it will be upgraded when they are upgraded. No further action is required. If virt-who is installed elsewhere, it must be upgraded manually.

**Before You Begin**

If virt-who is installed on a host registered to Satellite Server or a Capsule Server, first upgrade the host to the latest packages available in the Satellite Tools 6.10 repository. For information about upgrading hosts, see Section 3.4, “Upgrading Satellite Clients”.

Upgrade virt-who Manually
1. Upgrade virt-who.
   
   ```bash
   # yum upgrade virt-who
   ```

2. Restart the virt-who service so the new version is activated.
   
   ```bash
   # systemctl restart virt-who.service
   ```

### 4.3. REMOVING THE PREVIOUS VERSION OF THE SATELLITE TOOLS REPOSITORY

After completing the upgrade to Satellite 6.10, the Red Hat Satellite Tools 6.9 repository can be removed from Content Views and then disabled.

Disable Version 6.9 of the Satellite Tools Repository:

1. In the Satellite web UI, navigate to Content > Red Hat Repositories.

2. In the Enabled Repositories area, locate Red Hat Satellite Tools 6.9 for RHEL 7 Server RPMs x86_64.

3. Click the Disable icon to the right.

If the repository is still contained in a Content View then you cannot disable it. Packages from a disabled repository are removed automatically by a scheduled task.

### 4.4. RECLAIMING POSTGRESQL SPACE

The PostgreSQL database can use a large amount of disk space especially in heavily loaded deployments. Use this procedure to reclaim some of this disk space on Satellite.

**Procedure**

1. Stop all services, except for the `postgresql` service:
   
   ```bash
   # satellite-maintain service stop --exclude postgresql
   ```

2. Switch to the `postgres` user and reclaim space on the database:
   
   ```bash
   # su - postgres -c 'vacuumdb --full --dbname=foreman'
   ```

3. Start the other services when the vacuum completes:
   
   ```bash
   # satellite-maintain service start
   ```

4. Confirm that the files exist in the `/var/lib/pgsql/` directory:
   
   ```bash
   # ls -l /var/lib/pgsql/
   # du -sh /var/lib/pgsql/
   ```

5. Delete the data from the `/var/lib/pgsql/` directory:
# rm -rf /var/lib/pgsql/*

## 4.5. Updating Templates, Parameters, Lookup Keys and Values

During the upgrade process, Satellite attempts to locate macros that are deprecated for Satellite 6.10 and converts old syntax to new syntax for the default Satellite templates, parameters, and lookup keys and values. However, Satellite does not convert old syntax in the custom templates that you have created and in the cloned templates.

The process uses simple text replacement, for example:

```plaintext
@host.params['parameter1'] -> host_param('parameter1')
@host.param_true?('parameter1') -> host_param_true?('parameter1')
@host.param_false?('parameter1') -> host_param_false?('parameter1')
@host.info['parameters'] -> host_enc['parameters']
```

**WARNING**

If you use cloned templates in Satellite, verify whether the cloned templates have diverged from the latest version of the original templates in Satellite. The syntax for the same template can differ between versions of Satellite. If your cloned templates contain outdated syntax, update the syntax to match the latest version of the template.

To ensure that this text replacement does not break or omit any variables in your files during the upgrade, check all templates, parameters, and lookup keys and values for the old syntax and replace manually.

The following error occurs because of old syntax remaining in files after the upgrade:

```plaintext
undefined method '#params' for Host::Managed::Jail
```

### Fixing the outdated subscription_manager_registration snippet

Satellite 6.4 onwards uses the `redhat_register` snippet instead of the `subscription_manager_registration` snippet.

If you upgrade from Satellite 6.3 and earlier, ensure to replace the `subscription_manager_registration` snippet in your custom templates as follows:

```plaintext
<%== snippet "subscription_manager_registration" %>  
<%== snippet 'redhat_register' %>
```

## 4.6. Tuning Satellite Server with Predefined Profiles

If your Satellite deployment includes more than 5000 hosts, you can use predefined tuning profiles to...
If your Satellite deployment includes more than 5000 hosts, you can use predefined tuning profiles to improve performance of Satellite.

Note that you cannot use tuning profiles on Capsules.

You can choose one of the profiles depending on the number of hosts your Satellite manages and available hardware resources.

The tuning profiles are available in the `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes` directory.

When you run the `satellite-installer` command with the `--tuning` option, deployment configuration settings are applied to Satellite in the following order:

1. The default tuning profile defined in the `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` file
2. The tuning profile that you want to apply to your deployment and is defined in the `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/` directory
3. Optional: If you have configured a `/etc/foreman-installer/custom-hiera.yaml` file, Satellite applies these configuration settings.

Note that the configuration settings that are defined in the `/etc/foreman-installer/custom-hiera.yaml` file override the configuration settings that are defined in the tuning profiles.

Therefore, before applying a tuning profile, you must compare the configuration settings that are defined in the default tuning profile in `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml`, the tuning profile that you want to apply and your `/etc/foreman-installer/custom-hiera.yaml` file, and remove any duplicated configuration from the `/etc/foreman-installer/custom-hiera.yaml` file.

default

    Number of managed hosts: 0–5000
    RAM: 20G

    Number of CPU cores: 4

medium

    Number of managed hosts: 5001–10000
    RAM: 32G

    Number of CPU cores: 8

large

    Number of managed hosts: 10001–20000
    RAM: 64G

    Number of CPU cores: 16

extra-large

    Number of managed hosts: 20001–60000
    RAM: 128G

    Number of CPU cores: 32
Procedure

1. **Optional:** If you have configured the `custom-hiera.yaml` file on Satellite Server, back up the `/etc/foreman-installer/custom-hiera.yaml` file to `custom-hiera.original`. You can use the backup file to restore the `/etc/foreman-installer/custom-hiera.yaml` file to its original state if it becomes corrupted:

   ```bash
   # cp /etc/foreman-installer/custom-hiera.yaml /etc/foreman-installer/custom-hiera.original
   ```

2. **Optional:** If you have configured the `custom-hiera.yaml` file on Satellite Server, review the definitions of the default tuning profile in `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` and the tuning profile that you want to apply in `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/`. Compare the configuration entries against the entries in your `/etc/foreman-installer/custom-hiera.yaml` file and remove any duplicated configuration settings in your `/etc/foreman-installer/custom-hiera.yaml` file.

3. Enter the `satellite-installer` command with the `--tuning` option for the profile that you want to apply. For example, to apply the medium tuning profile settings, enter the following command:

   ```bash
   # satellite-installer --tuning medium
   ```
CHAPTER 5. UPDATING SATELLITE SERVER, CAPSULE SERVER, AND CONTENT HOSTS

Use this chapter to update your existing Satellite Server, Capsule Server, and Content Hosts to a new minor version, for example, from 6.10.0 to 6.10.1.

Updates patch security vulnerabilities and minor issues discovered after code is released, and are often fast and non-disruptive to your operating environment.

Before updating, back up your Satellite Server and all Capsule Servers. For more information, see Backing Up Satellite Server and Capsule Server in the Administering Red Hat Satellite guide.

5.1. UPDATING SATELLITE SERVER

Prerequisites

- Ensure that you have synchronized Satellite Server repositories for Satellite, Capsule, and Satellite Tools 6.10.
- Ensure each external Capsule and Content Host can be updated by promoting the updated repositories to all relevant Content Views.

WARNING

If you customize configuration files, manually or use a tool such as Hiera, these customizations are overwritten when the installation script runs during upgrading or updating. You can use the --noop option with the satellite-installer script to test for changes. For more information, see the Red Hat Knowledgebase solution How to use the noop option to check for changes in Satellite config files during an upgrade.

Updating Satellite Server to the Next Minor Version

To Update Satellite Server:

1. Ensure the Satellite Maintenance repository is enabled:

   ```bash
   # subscription-manager repos --enable \r
erhel-7-server-satellite-maintenance-6-rpms
   ```

2. Check the available versions to confirm the next minor version is listed:

   ```bash
   # satellite-maintain upgrade list-versions
   ```

3. Use the health check option to determine if the system is ready for upgrade. On first use of this command, satellite-maintain prompts you to enter the hammer admin user credentials and saves them in the /etc/foreman-maintain/foreman-maintain-hammer.yml file.

   ```bash
   # satellite-maintain upgrade check --target-version 6.10.z
   ```
Review the results and address any highlighted error conditions before performing the upgrade.

4. Because of the lengthy update time, use a utility such as `screen` to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously. For more information about using the screen command, see How do I use the screen command? article in the Red Hat Knowledge Base. If you lose connection to the command shell where the upgrade command is running, you can see the logged messages in the `/var/log/foreman-installer/satellite.log` file to check if the process completed successfully.

5. Perform the upgrade:

```bash
# satellite-maintain upgrade run --target-version 6.10.z
```

6. Check when the kernel packages were last updated:

```bash
# rpm -qa --last | grep kernel
```

7. Optional: If a kernel update occurred since the last reboot, stop the `satellite-maintain` services and reboot the system:

```bash
# satellite-maintain service stop
# reboot
```

### 5.2. UPDATING DISCONNECTED SATELLITE SERVER

This section describes the steps needed to update in an Air-gapped Disconnected setup where the connected Satellite Server (which synchronizes content from CDN) is air gapped from a disconnected Satellite Server.

Complete the following steps on the connected Satellite Server.

1. Ensure that you have synchronized the following repositories in your connected Satellite Server.

   - `rhel-7-server-ansible-2.9-rpms`
   - `rhel-7-server-rpms`
   - `rhel-7-server-satellite-6.10-rpms`
   - `rhel-7-server-satellite-maintenance-6-rpms`
   - `rhel-server-rhscl-7-rpms`

2. Download the debug certificate of the organization and store it locally at, for example, `/etc/pki/katello/certs/org-debug-cert.pem` or a location of your choosing.

3. Create a Yum configuration file under `/etc/yum.repos.d` with the following repository information:

```ini
[rhel-7-server-ansible-2.9-rpms]
name=Ansible 2.9 RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/server/7/$releasever/$basearch/ansible/2.9/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
```
4. In the configuration file, replace `/etc/pki/katello/certs/org-debug-cert.pem` in `sslclientcert` and `sslclientkey` with the location of the downloaded organization debug certificate.

5. Update `satellite.example.com` with correct FQDN for your deployment.

6. Replace `My_Organization` with the correct organization label in the `baseurl`. To obtain the organization label, enter the command:

   ```bash
   # hammer organization list
   ```

7. Enter the `reposync` command:

   ```bash
   # reposync --delete --download-metadata -p ~/Satellite-repos -n \
   -r rhel-7-server-ansible-2.9-rpms \
   ```
This downloads the contents of the repositories from the connected Satellite Server and stores them in the directory `~/Satellite-repos`. The `reposync` command in Red Hat Enterprise Linux 7 downloads the RPMs but not the Yum metadata. Because of this, you must manually run `createrepo` in each sub-directory of `Satellite-repos`. Make sure you have the `createrepo` rpm installed. If not use the following command to install it.

```bash
# satellite-maintain packages install createrepo
```

Run the following command to create repodata in each sub-directory of `~/Satellite-repos`:

```bash
# cd ~/Satellite-repos
# for directory in */
do
echo "Processing $directory"
  cd $directory
  createrepo .
cd..
done
```

8. Verify that the RPMs have been downloaded and the repository data directory is generated in each of the sub-directories of `~/Satellite-repos`.

9. Archive the contents of the directory

```bash
# cd ~
# tar czf Satellite-repos.tgz Satellite-repos
```

10. Use the generated `Satellite-repos.tgz` file to upgrade in the disconnected Satellite Server.

Perform the following steps on the disconnected Satellite Server

1. Copy the generated `Satellite-repos.tgz` file to your disconnected Satellite Server

2. Extract the archive to anywhere accessible by the root user. In the following example `/root` is the extraction location.

```bash
# cd /root
# tar zxf Satellite-repos.tgz
```

3. Create a Yum configuration file under `/etc/yum.repos.d` with the following repository information:

```plaintext
[rhel-7-server-ansible-2.9-rpms]
name=Ansible 2.9 RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-ansible-2.9-rpms
enabled=1

[rhel-7-server-rpms]
```
name=Red Hat Enterprise Linux 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-rpms
enabled=1

[rhel-7-server-satellite-6.10-rpms]
name=Red Hat Satellite 6 for RHEL 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-satellite-6.10-rpms
enabled=1

[rhel-7-server-satellite-maintenance-6-rpms]
name=Red Hat Satellite Maintenance 6 for RHEL 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-satellite-maintenance-6-rpms
enabled=1

[rhel-server-rhscl-7-rpms]
name=Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=file:///root/Satellite-repos/rhel-server-rhscl-7-rpms
enabled=1

4. In the configuration file, replace the /root/Satellite-repos with the extracted location.

5. Check the available versions to confirm the next minor version is listed:

   # satellite-maintain upgrade list-versions

6. Use the health check option to determine if the system is ready for upgrade. On first use of this command, satellite-maintain prompts you to enter the hammer admin user credentials and saves them in the /etc/foreman-maintain/foreman-maintain-hammer.yml file.

   # satellite-maintain upgrade check --whitelist="check-upstream-repository,repositories-validate" --target-version 6.10.z

7. Review the results and address any highlighted error conditions before performing the upgrade.

8. Because of the lengthy update time, use a utility such as screen to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously. For more information about using the screen command, see How do I use the screen command? article in the Red Hat Knowledge Base.

   If you lose connection to the command shell where the upgrade command is running, you can see the logged messages in the /var/log/foreman-installer/satellite.log file to check if the process completed successfully.

9. Perform the upgrade:

   # satellite-maintain upgrade run --whitelist="check-upstream-repository,repositories-validate"
   --target-version 6.10.z

10. Check when the kernel packages were last updated:

    # rpm -qa --last | grep kernel

11. Optional: If a kernel update occurred since the last reboot, stop the satellite-maintain services and reboot the system:
CHAPTER 5. UPDATING SATELLITE SERVER, CAPSULE SERVER, AND CONTENT HOSTS

5.3. UPDATING CAPSULE SERVER

Use this procedure to update Capsule Servers to the next minor version.

Procedure

1. Ensure that the Satellite Maintenance repository is enabled:

```bash
# subscription-manager repos --enable \
  rhel-7-server-satellite-maintenance-6-rpms
```

2. Check the available versions to confirm the next minor version is listed:

```bash
# satellite-maintain upgrade list-versions
```

3. Use the health check option to determine if the system is ready for upgrade:

```bash
# satellite-maintain upgrade check --target-version 6.10.z
```

Review the results and address any highlighted error conditions before performing the upgrade.

4. Because of the lengthy update time, use a utility such as `screen` to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously. For more information about using the `screen` command, see [How do I use the `screen` command?](https://access.redhat.com/solutions/1877) article in the *Red Hat Knowledge Base*. If you lose connection to the command shell where the upgrade command is running, you can see the logged messages in the `/var/log/foreman-installer/satellite.log` file to check if the process completed successfully.

5. Perform the upgrade:

```bash
# satellite-maintain upgrade run --target-version 6.10.z
```

6. Check when the kernel packages were last updated:

```bash
# rpm -qa --last | grep kernel
```

7. Optional: If a kernel update occurred since the last reboot, stop the `satellite-maintain` services and reboot the system:

```bash
# satellite-maintain service stop
# reboot
```