Satellite 6 - Reference Guide

Document control

Title	Satellite 6 - Reference Guide
Prepared By	Ranjith Plakkil
Document Version	1.0
Contact	Pepsunix.in@capgemini.com

Version History

Revision Number	Revision Date	Comment
1.0		Original document

Table of Content

1. Configuring Organizations, Locations and Life Cycle Environments

Red Hat Satellite 6 takes a consolidated approach to Organization and Location management. System administrators define multiple Organizations and multiple Locations in a single Satellite Server. For example, a company might have three Organizations (Finance, Marketing, and Sales) across three countries (United States, United Kingdom, and Japan).

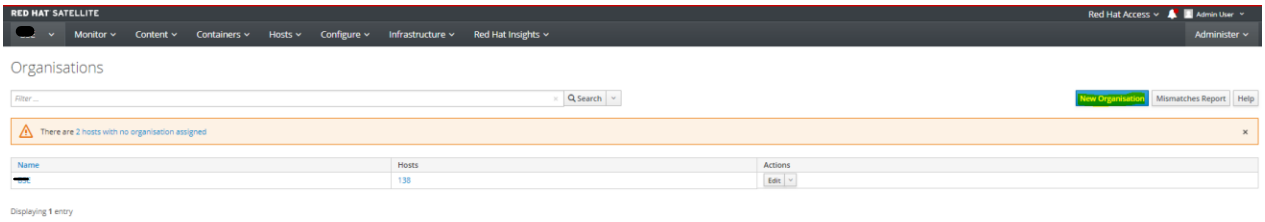
1.1. Organizations

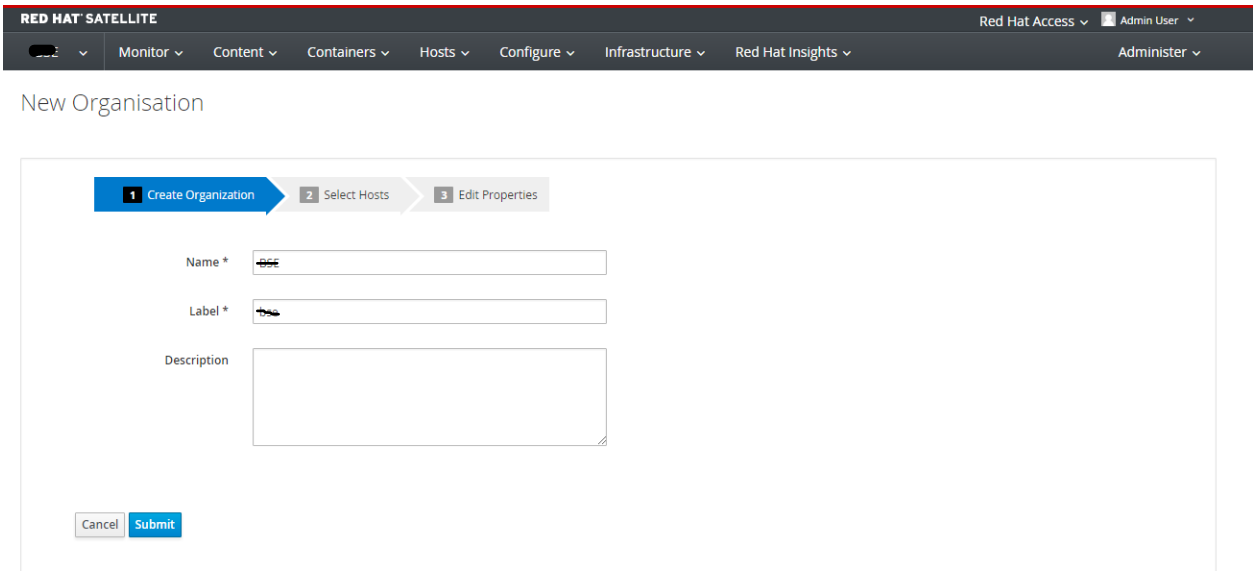
Organizations divide hosts into logical groups based on ownership, purpose, content, security level, or other divisions. Multiple organizations can be viewed, created, and managed within the web UI. Software and host entitlements can be allocated across many organizations and access to those organizations controlled.

1.1.1. Creating an Organization

- Navigate to **Administer → Organizations**.

-

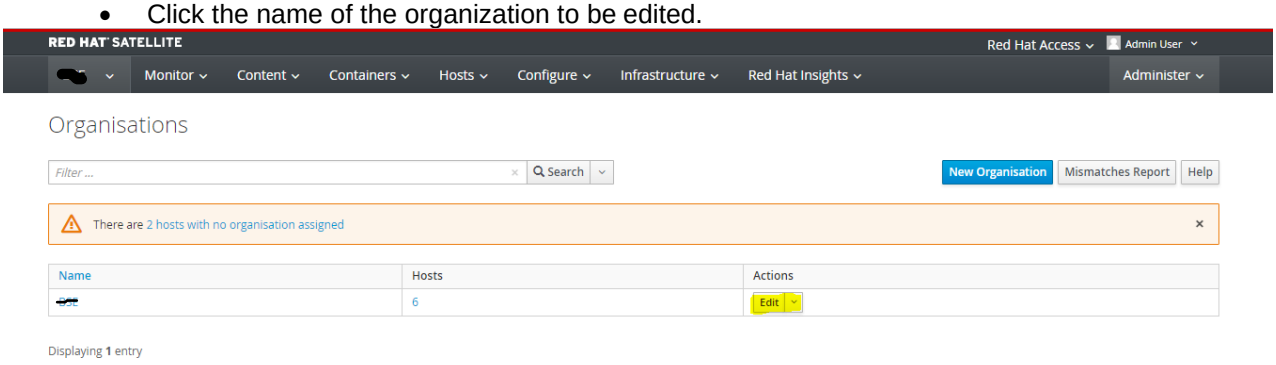




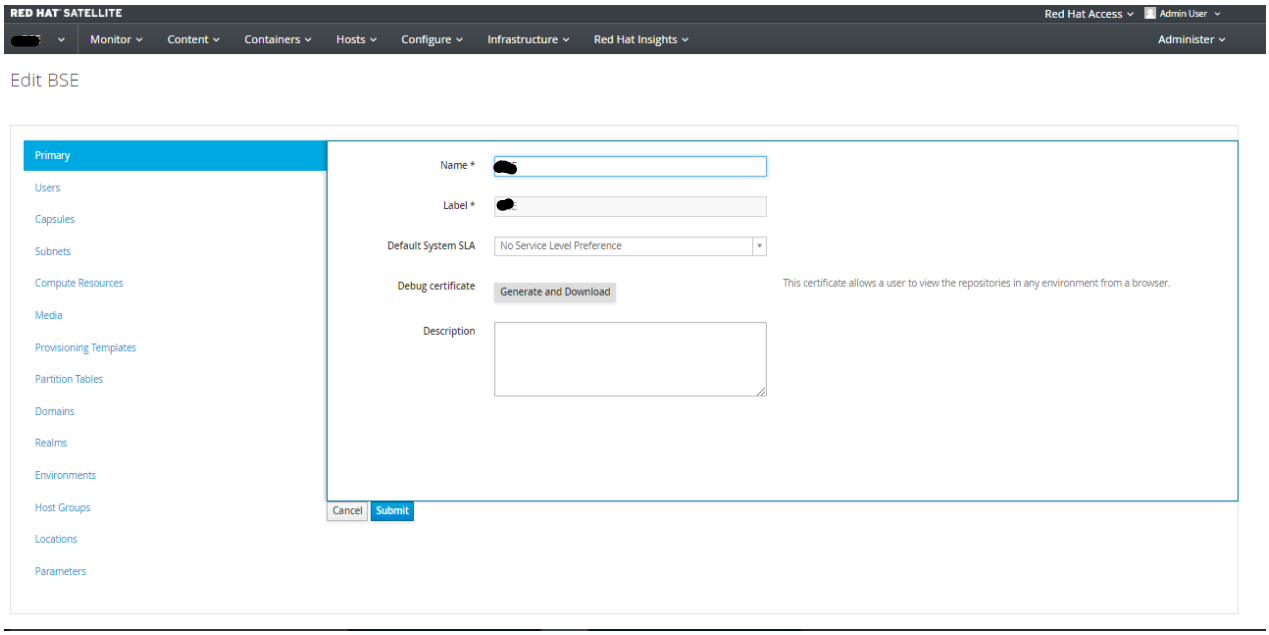
- Select the hosts to assign to the new organization.
- Click **Assign All** to assign all hosts with no organization to the new organization.
- Click **Manually Assign** to manually select and assign the hosts with no organization.
- Click **Proceed to edit** to skip assigning hosts.
- Click **Submit**.

1.1.2. Editing an Organization

- Navigate to **Administer** → **Organizations**.
- Click the name of the organization to be edited.

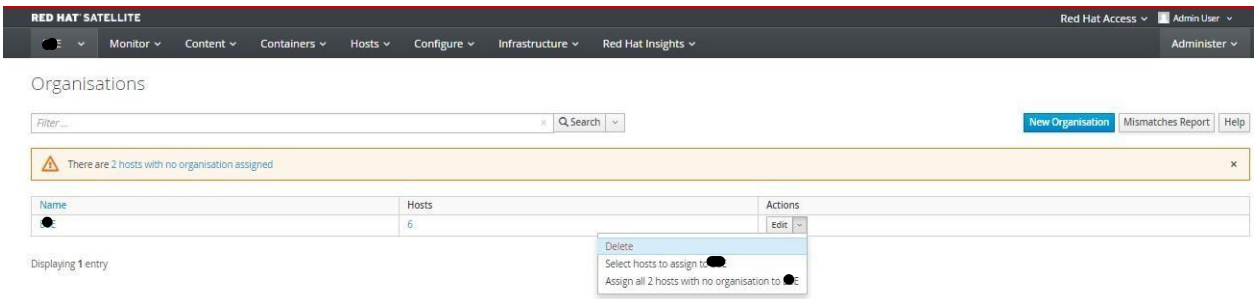


- Select the resource to edit from the list on the left.
- Click the name of the desired items to add them to the **Selected Items** list.
- Click **Submit**.



1.1.3. Removing an Organization

- Navigate to **Administer** → **Organizations**.
- Select **Delete** from the drop-down menu to the right of the name of the organization
- You want to remove. An alert box appears: **Delete *Organization*?**
- Click **OK** to delete the organization.



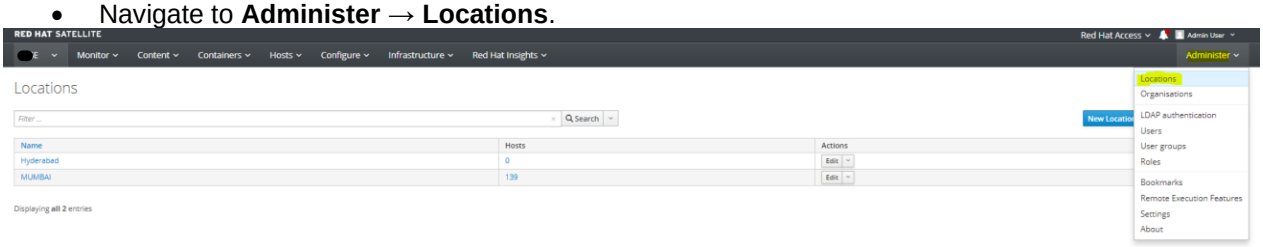
1.2. Locations

Locations divide organizations into logical groups based on geographical location. The Red Hat Satellite installation process creates a location called **Default Location** unless another name is specified. If a new user is not assigned a default location their access will be limited. To grant system rights to users, assign a default location and have them log out and log in again.

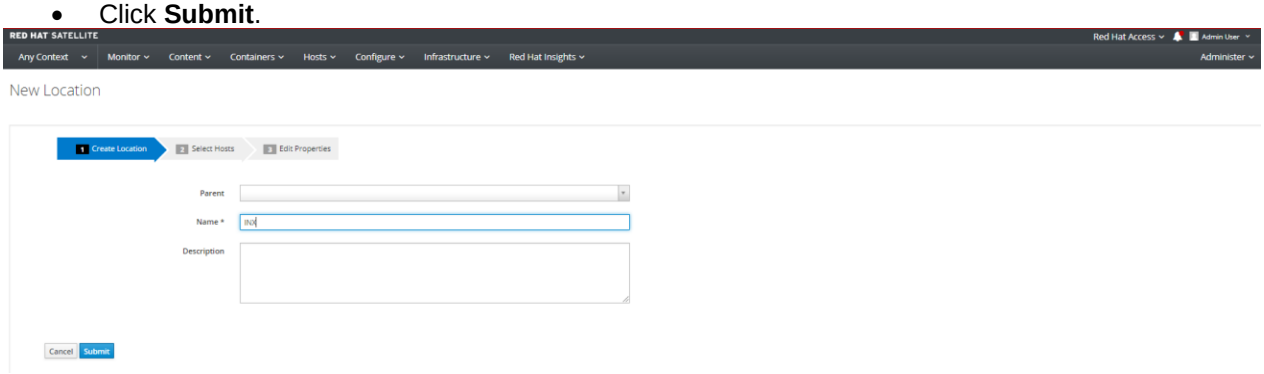
1.2.1. Creating a Location

These steps show how to create a location.

- Navigate to **Administer** → **Locations**.

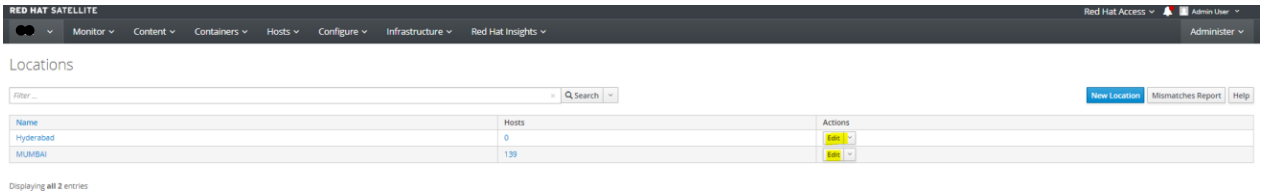


- Click **New Location**.
- Insert the name of the new location in the **Name** field. If you want to create a nested location, select a **Parent** location from the drop-down menu. Optionally, specify a **Description** of the location. Click **Submit**.
- Select the hosts to assign to the new location.
- Click **Assign All** to assign all hosts with no location to the new location.
- Click **Manually Assign** to manually select and assign the hosts with no location.
- Click **Proceed to Edit** to skip assigning hosts.
- Click **Submit**.

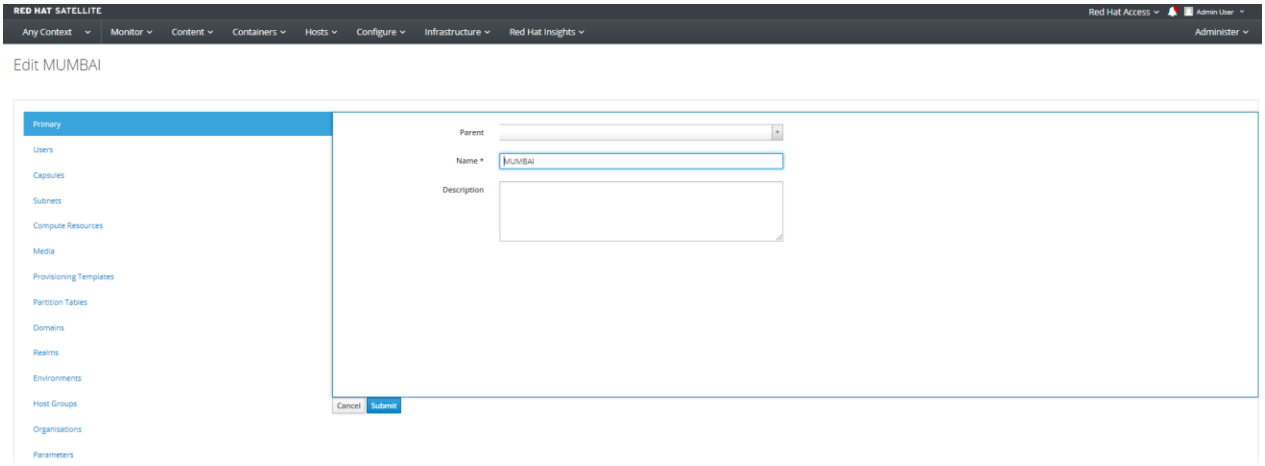


1.2.2. Editing a Location

- Navigate to **Administer** → **Locations**.
- Click the name of the location to be edited.
- Select the resource to edit from the list on the left.



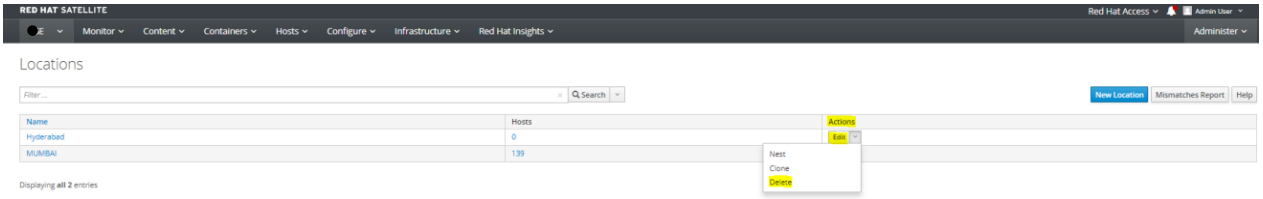
- Click the name of the desired items to add them to the **Selected Items** list.
- Click **Submit**.



1.2.3. Removing a Location

These steps show how to remove an existing location. Deleting the default location created during installation is currently not supported.

- Navigate to **Administer** → **Locations**.
- Select **Delete** from the drop-down menu to the right of the name of the location you want to remove. An alert box appears: Delete *Location*?
- Click **OK**.



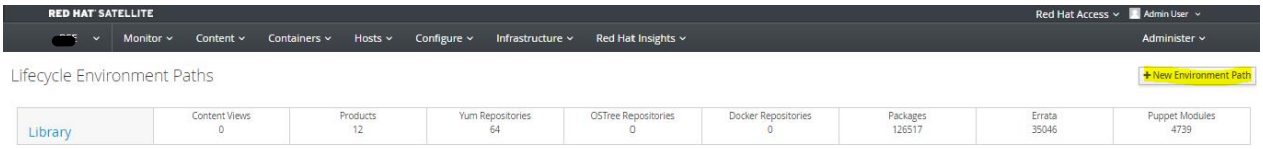
1.3. Life Cycle Environments

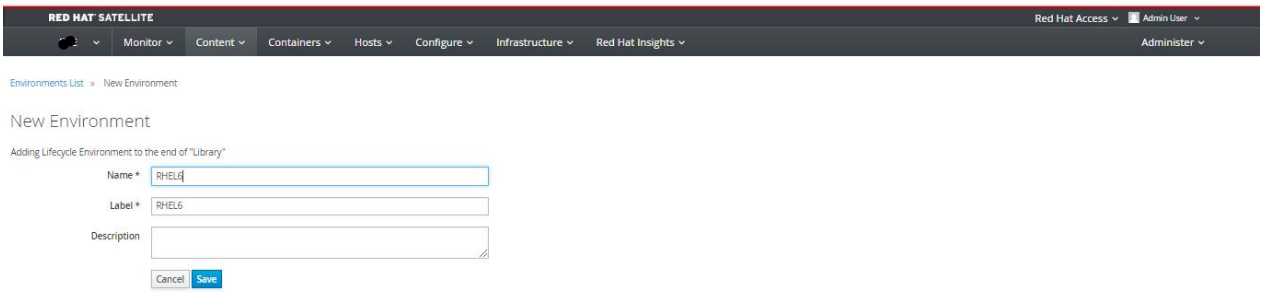
Application life cycles are divided into *life cycle environments*, which represent each stage of the application life cycle. Life cycle environments are linked to form an *environment path*. You can promote content along the environment path to the next life cycle environment when required. For example, if development ends on a particular version of an application, you can promote this version to the testing environment and start development on the next version.

1.3.1. Creating Life Cycle Environments

This procedure describes how to create a life cycle environment in Red Hat Satellite.

- Select an organization from the menu in the top left hand corner.
- Click **Content** → **Life Cycle Environments** and then click **New Environment Path**.
- Insert a name and a label (automatically fills in the **Name** field input) for the life cycle environment. The **Description** field is optional.
- Click **Save** to create the environment.





1.3.2. Removing Life Cycle Environments from Satellite Server

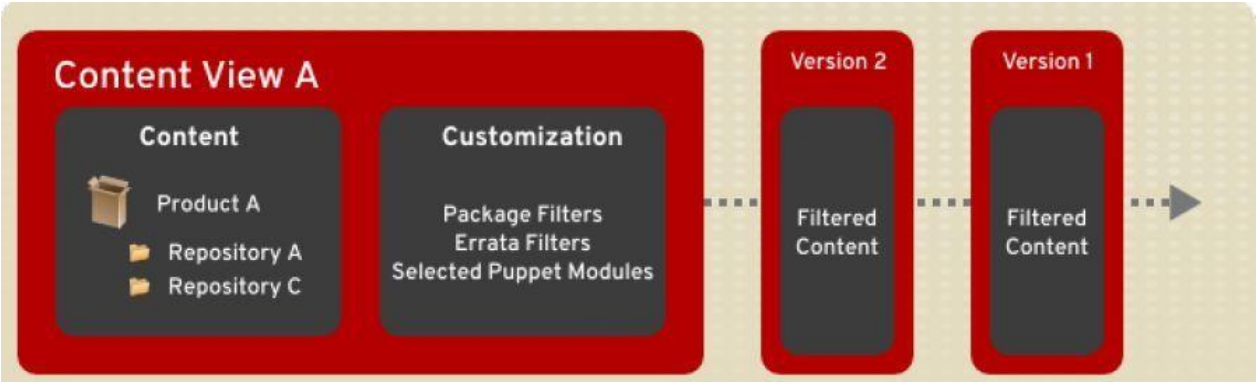
This procedure describes how to remove a life cycle environment from Red Hat Satellite.

To Remove a Life Cycle Environment:

- On the main menu, click **Content** → **Life Cycle Environments**.
- Click the name of the life cycle environment that you want to remove, and then click
- **Remove Environment**.
- In the confirmation dialog box, click **Remove** to remove the environment.

1.4. Using Content Views

Content views are managed selections of content, which contain one or more repositories (yum, puppet, or containers) with optional filtering. These filters can be either inclusive or Exclusive, and tailor a system view of content for life cycle management. They are used to customize content to be made available to client systems. Keeping repositories for yum, Puppet, and containers in separate Content Views has the advantage that any updates to one repository only requires republishing the relevant Content View. You can use Composite Content Views to combine published Content Views for ease of management.

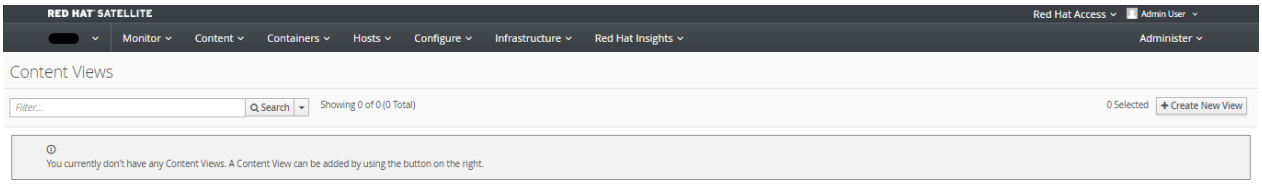


Published content views are used with life cycle environments.

1.4.1. Creating a Content View

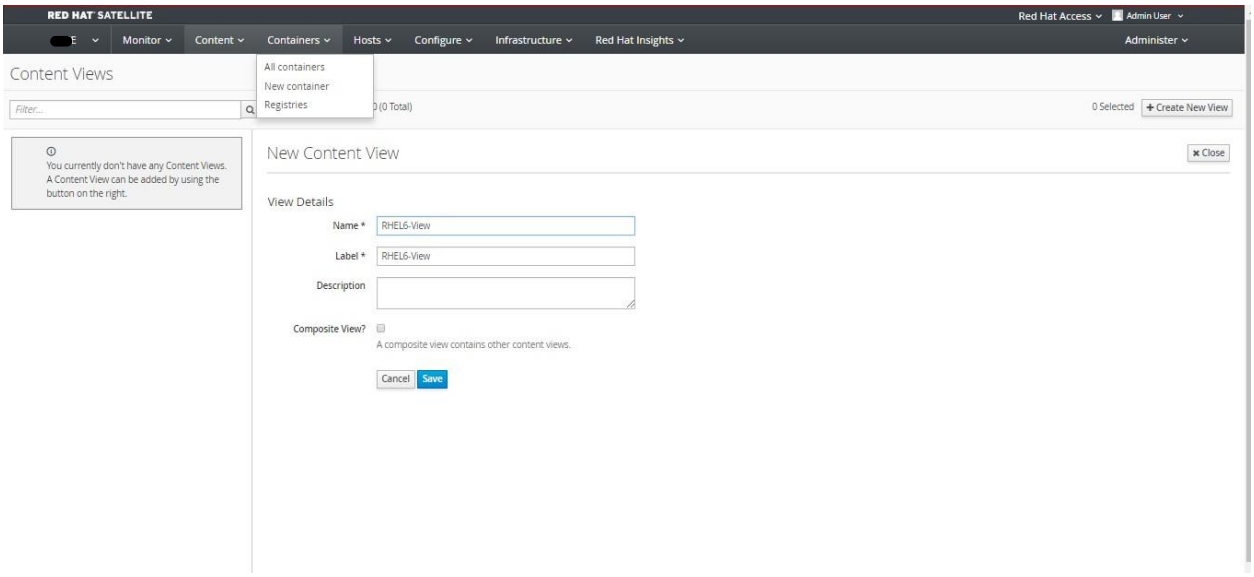
A user with administrator privileges can create content views for use within the life cycle environments.
To Create a Content View:

- Log in as a Satellite administrator.
- Click **Content** → **Content Views**.



- Click **Create New View**.
- Specify the **Name** of the content view. The **Label** field is automatically populated when the **Name** field is filled out. Optionally, provide a description of the content view.
- Select the **Composite Content View** check box to combine a series of published content views into one and choose which content view.
- 6. Click **Save**.

Note: If you select **Composite Content View** it will override any filtering and allow you to choose a group of published content views and bundle those views into a composite one.



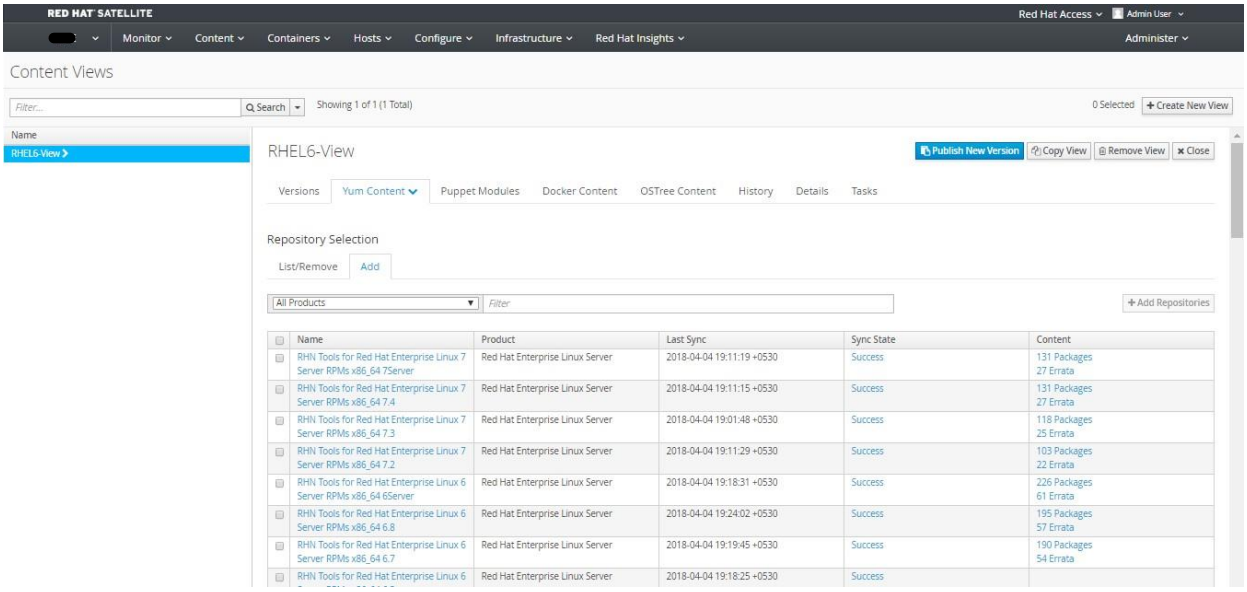
1.4.2. Adding Repositories to the Content View

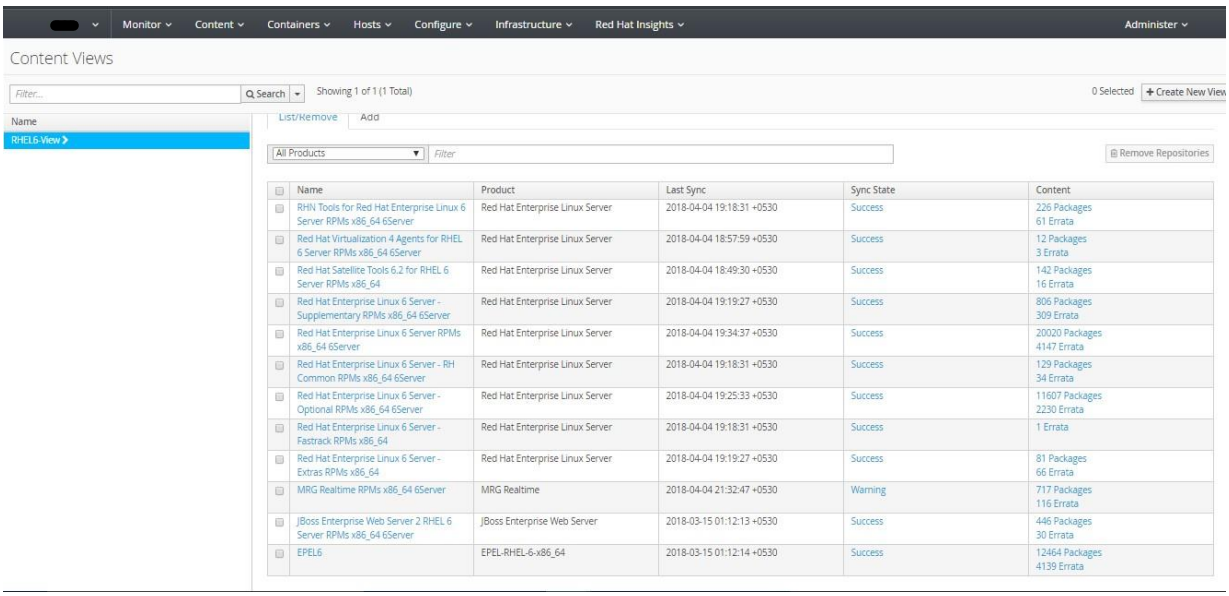
A repository provides storage for content. For example, a YUM repository, Puppet repository, or a Docker repository.
The **Content View** page contains a searchable list of content views. When searching for content views using the **Search** field, use an asterisk (*) to perform a partial string search. For example, if searching for a content view

named **RHEL7_Base**, entering **RHEL7** will not return any results, instead enter **RHEL7***. Alternatively, ***Base*** also retrieves the content view **RHEL7_Base**.

To Associate a Repository with a Content View:

- Click **Content** → **Content Views** and choose the Content View to add repositories to.
- Depending on the type of content you want to store: To add a Yum repository, click **Yum Content** and select **Repositories** from the dropdown menu. From the submenu, click **Add**. For example, to be able to install Katello agent on your host, you need to enable the **Satellite Tools** repository.
- To add a Puppet repository, click **Puppet Modules** and click **Add New Module**.
- To add a Docker repository, click **Docker Content** and click **Add** in the submenu.
- Select the repositories to add and click **Add Repositories**.





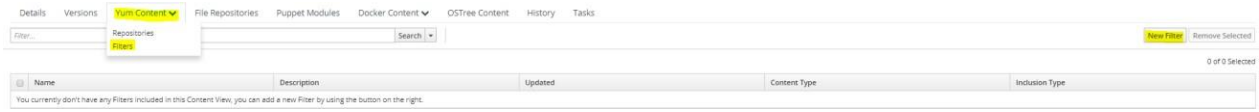
1.4.3. Filtering Content

Filters provide a mechanism to prevent packages from being promoted to subsequent environments. You can use package names or regular expressions in the filter to create the rules to blacklist packages. Then you can associate the filter to entire products or individual repositories within any product.

1.4.4. Creating a Filter

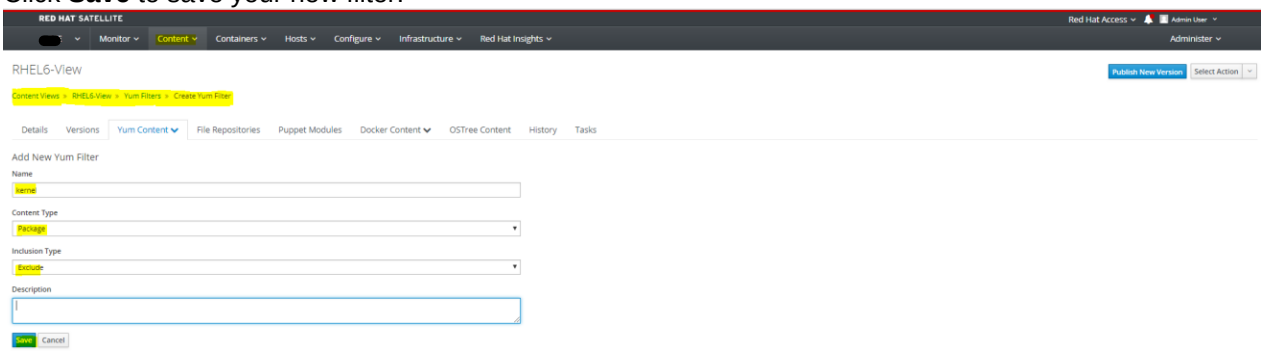
The following procedure shows how to create a filter for packages.
To Create a Filter:

- Navigate to **Content** → **Content Views** and select the Content View you want to filter.
- Click **Yum Content** → **Filters** and click **New Filter**



- In the **Name** field, specify the name of the new filter and choose a content type from the **Content Type** drop-down menu. Choose whether the filter includes or excludes the selected content type by selecting the **Type** drop-down menu. Optionally, insert a description in the **Description** field.

Click **Save** to save your new filter.

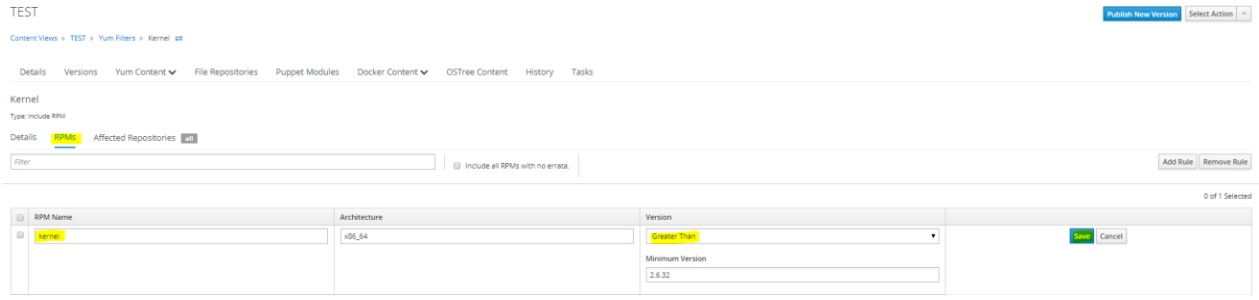


1.4.5. Adding Content to a Filter

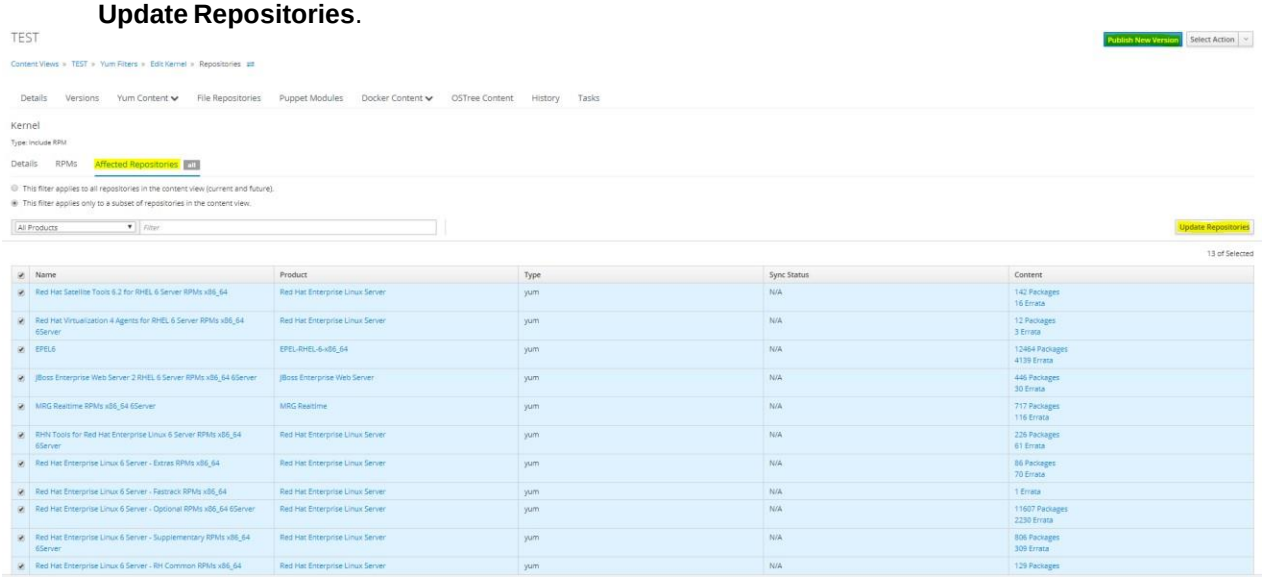
The following procedure shows how to add content to a package filter.
To Add Content to a Filter:

- Navigate to **Content** → **Content Views** and select the Content View you want to filter.
- Click **Yum Content** → **Filters** and click the name of the filter you want

to edit. Depending on the type of filter selected, perform the following actions:
a. If the filter is made for packages, specify a package name on the **Packages** subtab, and select a **Detail** value from the drop-down menu. Click **Add** to add the package to the filter.
b. If the filter is made for package groups, click the **Add** sub tab, and choose the desired package group. Click **Add Package Group**.
c. If the filter is made for errata, click the **Add** subtab. Select the errata type (**Security**, **Enhancement**, or **Bugfix**), and specify a start date and end date. Click **Add Errata**.
d. If the filter is made for errata - date and type, on the **Erratum Date Range** sub tab, select the errata type (**Security**, **Enhancement**, or **Bug fix**) and specify a start date and end date. Click **Save**.
e. Add specific Rules



- On the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, select the desired repositories and click **Update Repositories**.



- 4. Click **Publish New Version**. Insert a comment if desired, then click **Save**.

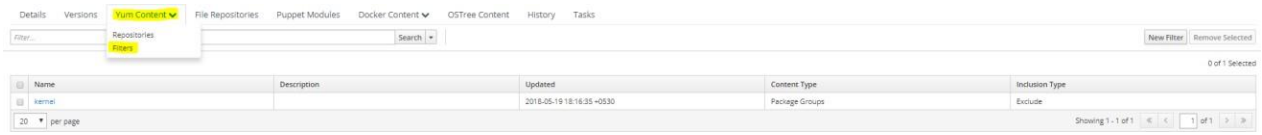
1.4.6. Removing a Filter

The following procedure shows how to remove a filter.

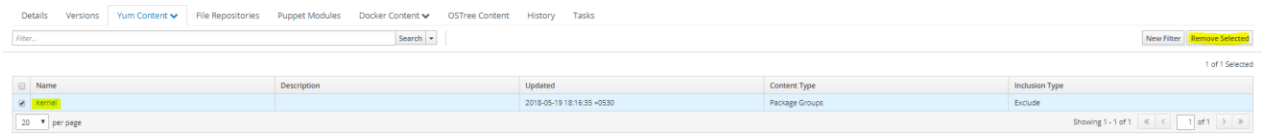
To Remove a Filter:

- Navigate to **Content** → **Content Views** and select the Content View you want to filter.

- Click **Yum Content** → **Filters** and select the check box next to the name of the package filter you want to remove.



- Click **Remove Filters**.

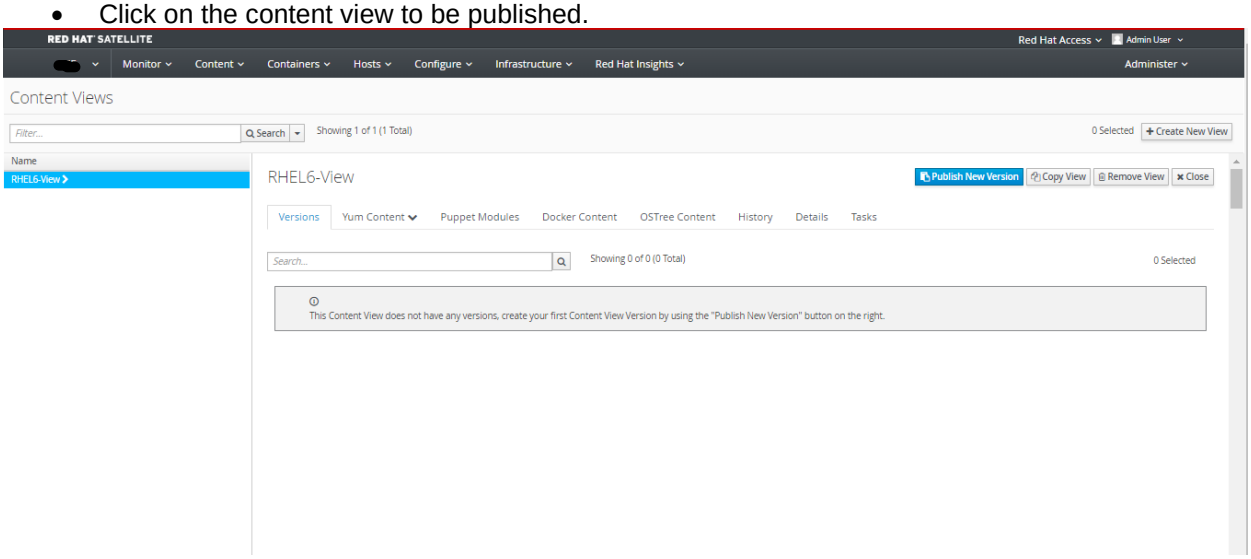


1.4.7. Publishing a Content View

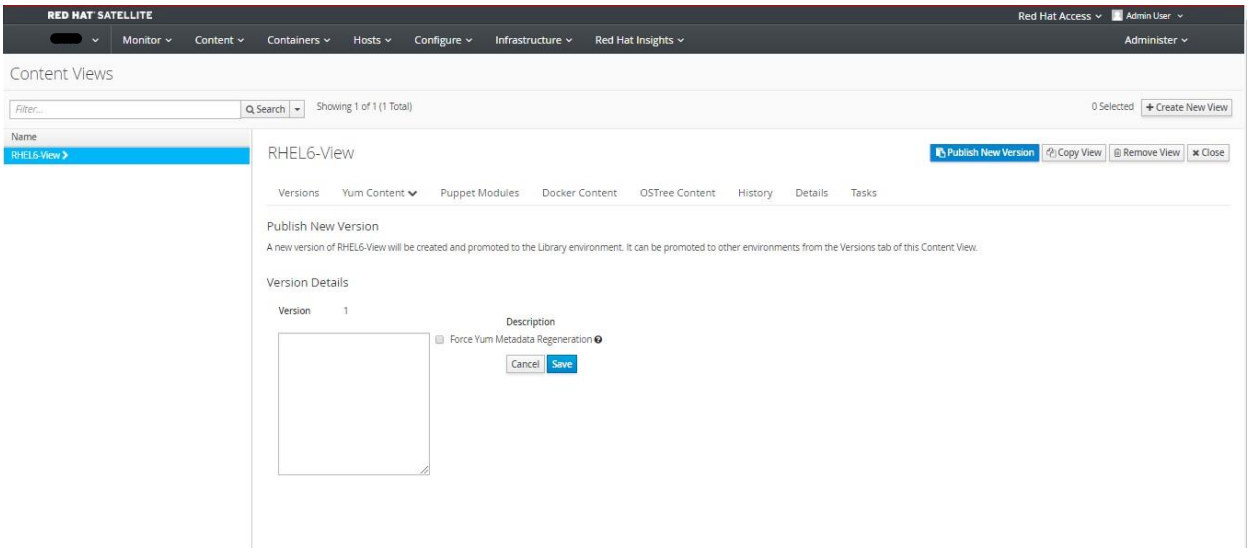
After a content view has been created, it needs to be published in order for it to be visible and usable by hosts. Before publishing the content view definition, make sure that the content view definition has the necessary products, repositories and filters.

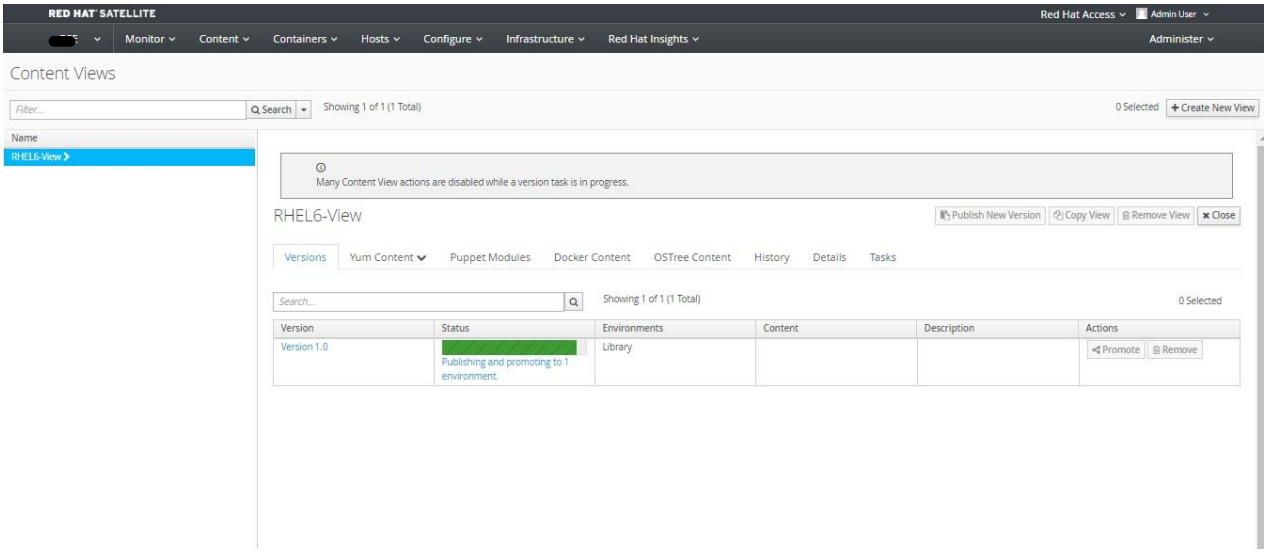
To Publish a Content View Definition:

- Click **Content** → **Content Views**.
- Click on the content view to be published.



- Click **Publish New Version**.
- Fill in a comment.
- Click **Save**.

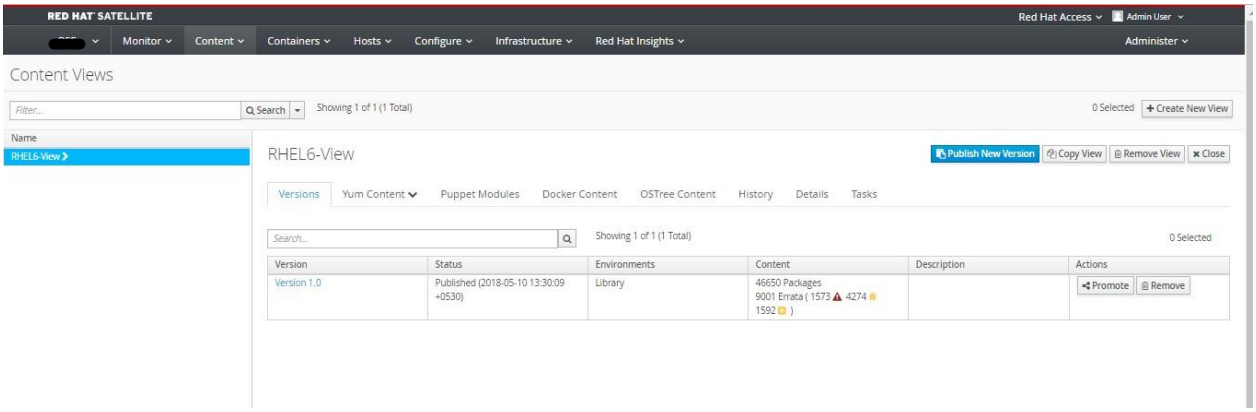




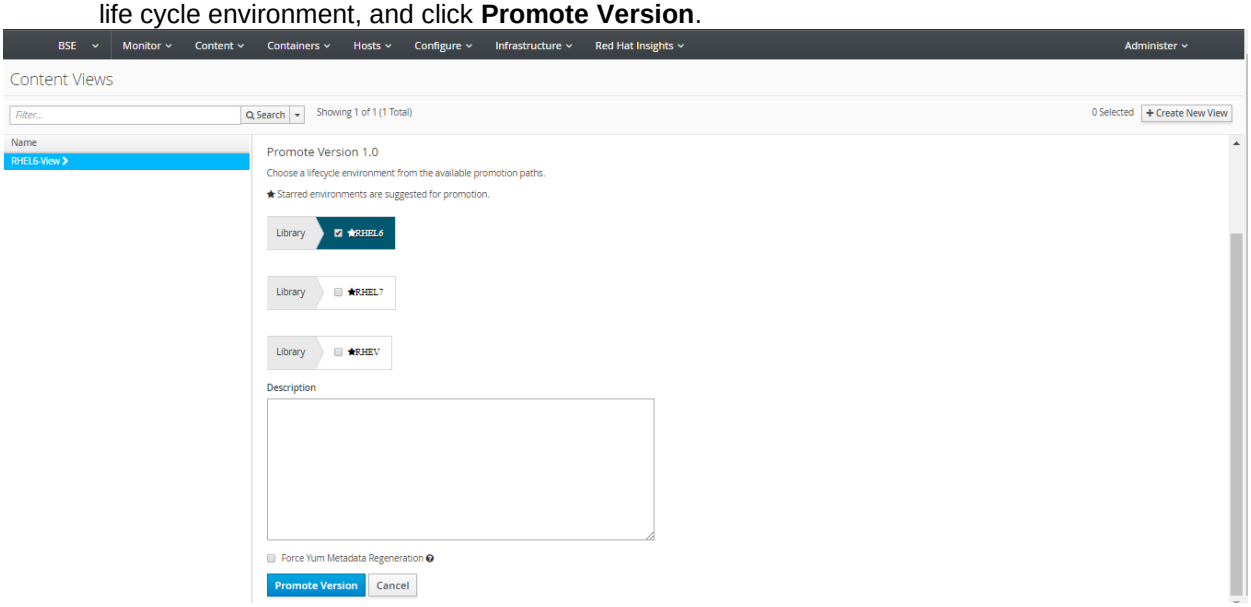
1.4.8. Promoting Content Views

After you have created a Content View and an environment path consisting of two or more life cycle environments, you can promote the Content View from one environment to the next as required. This means that the most recent version of the Content View that exists in a specified environment will be promoted, or copied, to the next environment in the life cycle environment path. You can promote a Content View to any environment where that version does not exist. The system automatically suggests the next environment in the life cycle environment path, but you can override this and promote to a different environment if required.

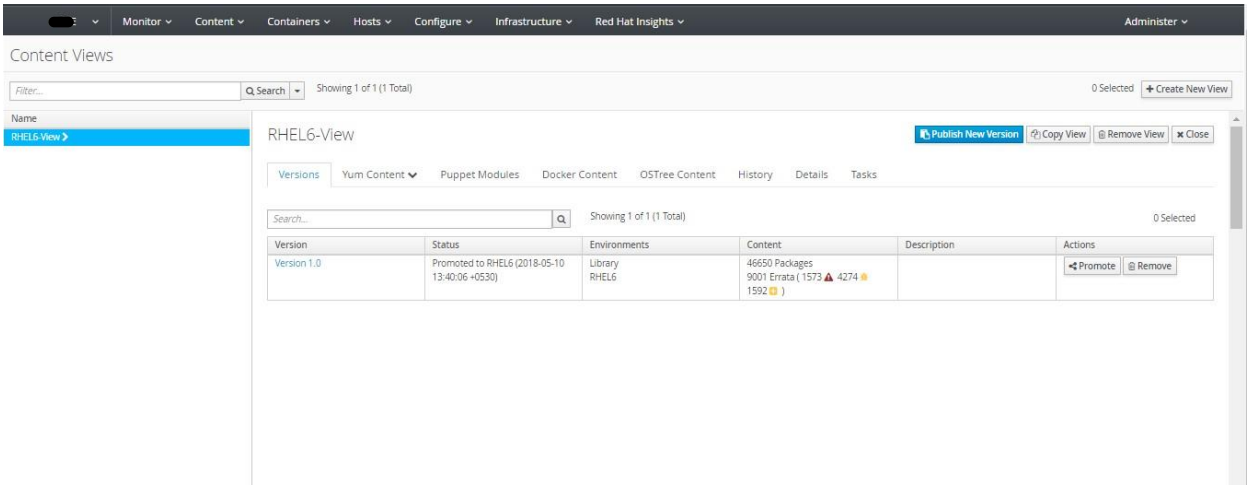
- On the main menu, click **Content** → **Content Views**.
- In the **Name** column, click the name of the Content View that you want to promote.
- On the **Versions** tab, identify the latest version, and click **Promote**.



- Identify the promotion path where you want to promote the Content View, select the appropriate life cycle environment, and click **Promote Version**.



- After the promotion has completed, the **Versions** tab updates to display the new status of your Content Views.



- Promoted version to Hosts Environment(ie;RHEL6)

1.5. Viewing and Applying Errata

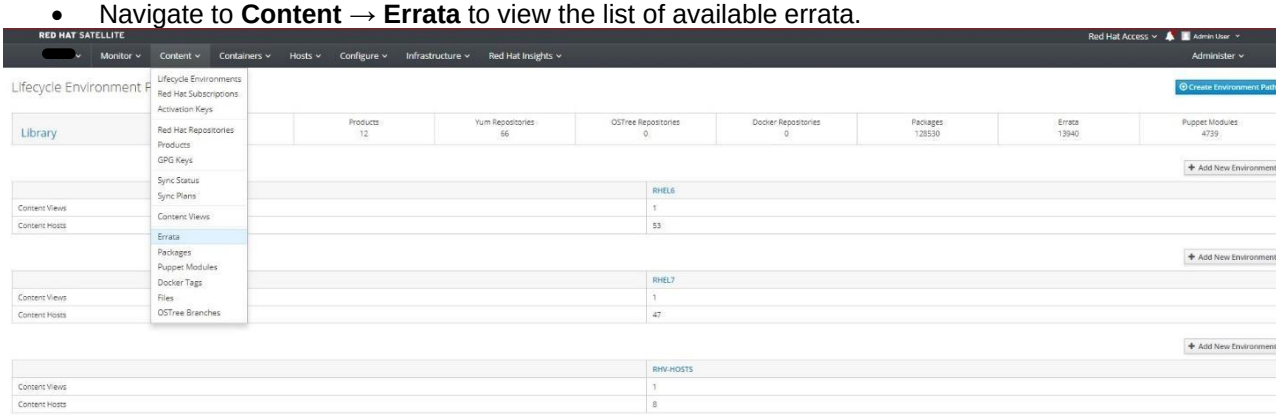
Navigate to **Monitor** → **Content Dashboard** to see the overview of errata synchronization.
Errata contain advisories that describe the changes introduced by the update. There are three types of advisories (in order of importance):
Security Advisory describes fixed security issues found in the package. The security impact of the issue can be *Low*, *Moderate*, *Important*, or *Critical*.
Bug Fix Advisory describes bug fixes for the package.
Product Enhancement Advisory describes enhancements and new features added to the package.

In Red Hat Satellite, there are two keywords that describe an erratum's relationship to the available content hosts: **Applicable**: erratum applies to one or more content hosts, which means it updates packages present on the content host. Applicable errata are not yet accessible by the content host.
Installable: erratum applies to one or more content hosts and it has been made available to the content host. Installable errata are present in the content host's life cycle environment and content view, but are not yet installed. This way, errata can be installed by users who have permissions to manage content hosts, but are not entitled for errata management at higher levels.

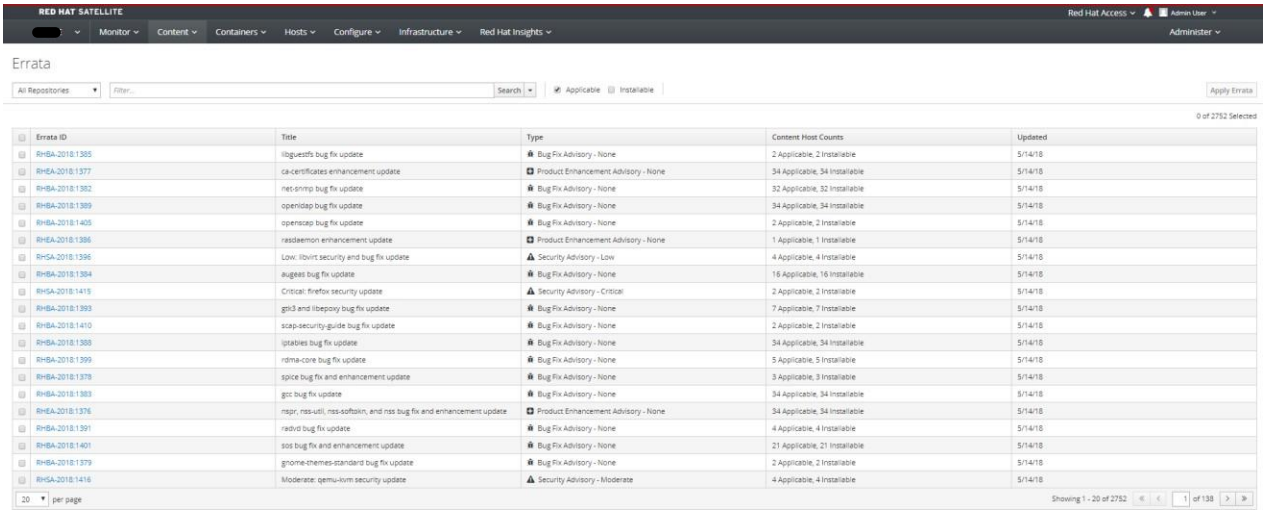
1.5.1. Applying Errata to Content Hosts

The following procedures show how to apply one or more errata to content hosts.
To Apply a Single Erratum to Content Hosts:

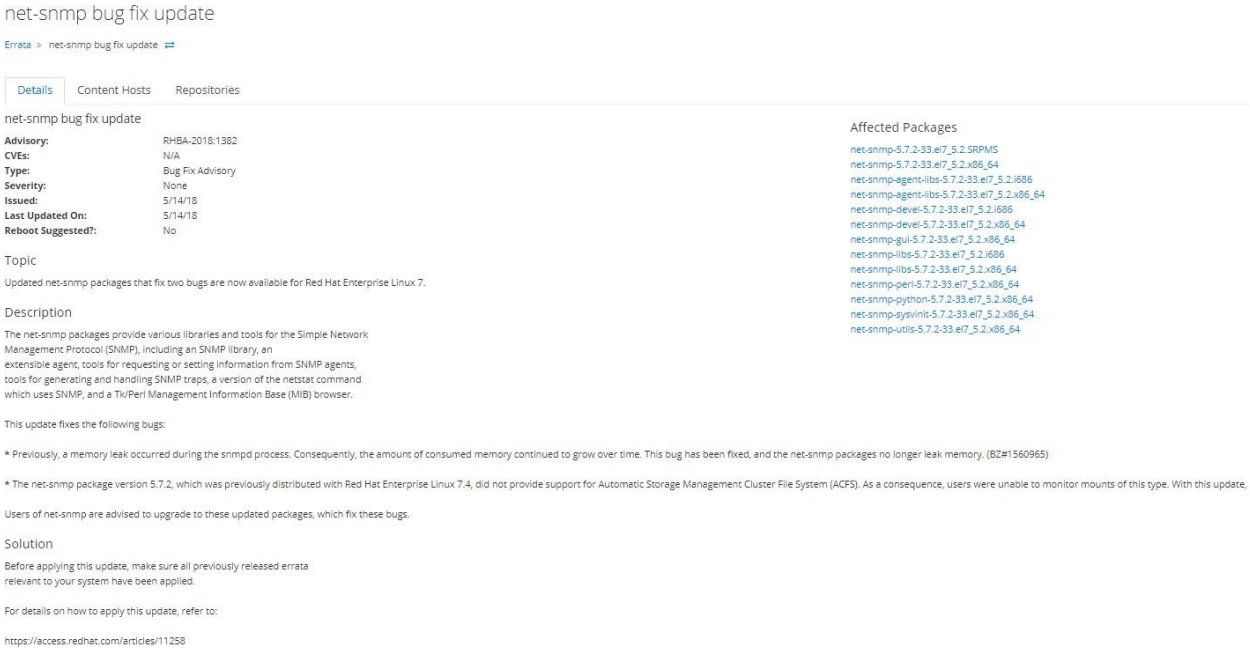
- Navigate to **Content** → **Errata** to view the list of available errata.



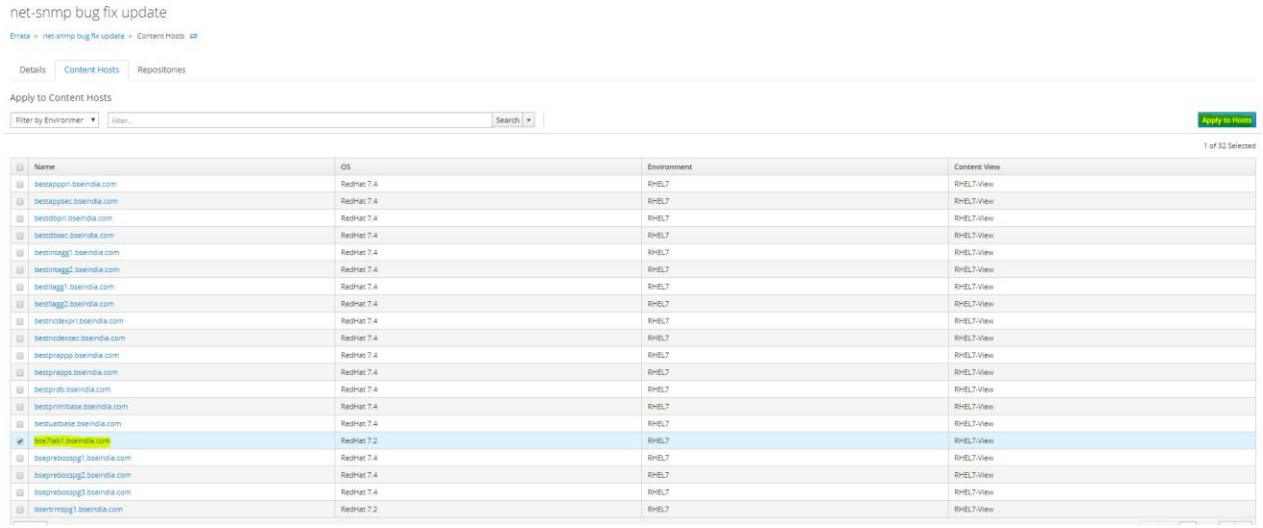
- Click the **Errata ID** of the erratum you want to apply.



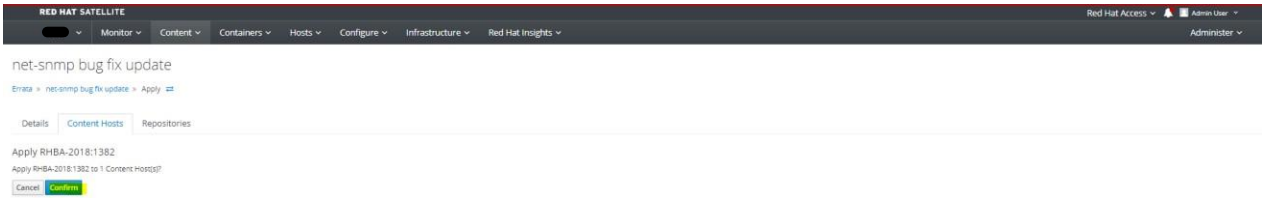
- On the **Content Hosts** tab, select one or more content hosts to be updated. You can filter the available content hosts by the environment, and search for them by name. If you select the check box at the top of the page, only the content hosts that already have the installable erratum in their life cycle environment are displayed.



- 4. Click **Apply to Hosts**.

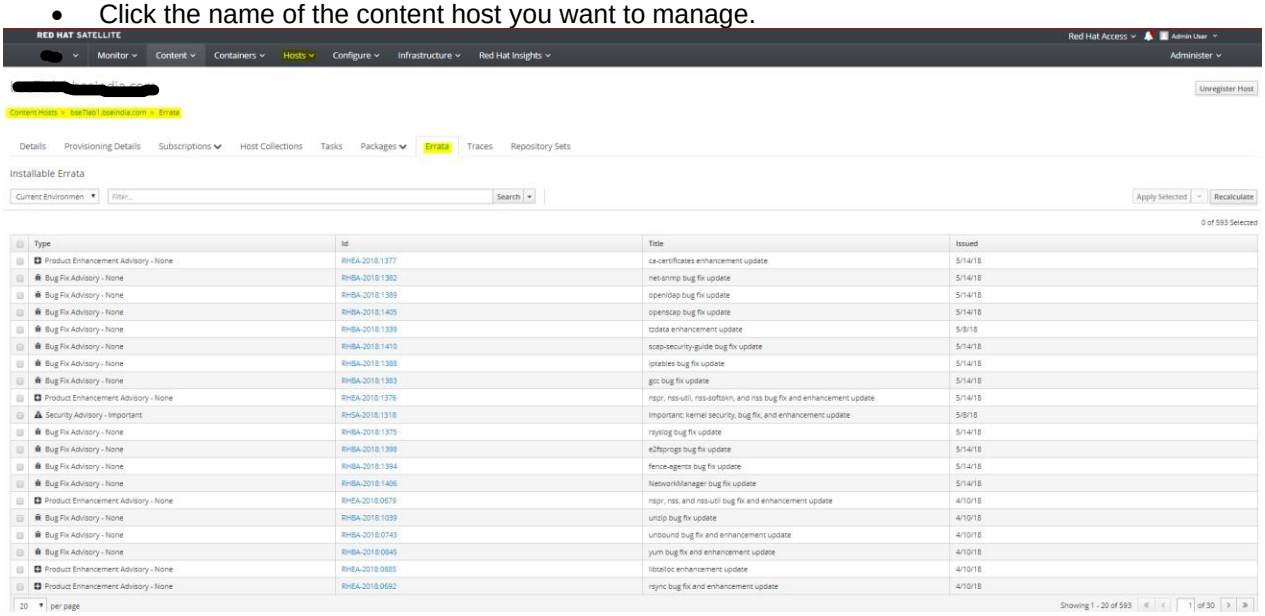


- 5. Click **Confirm**.

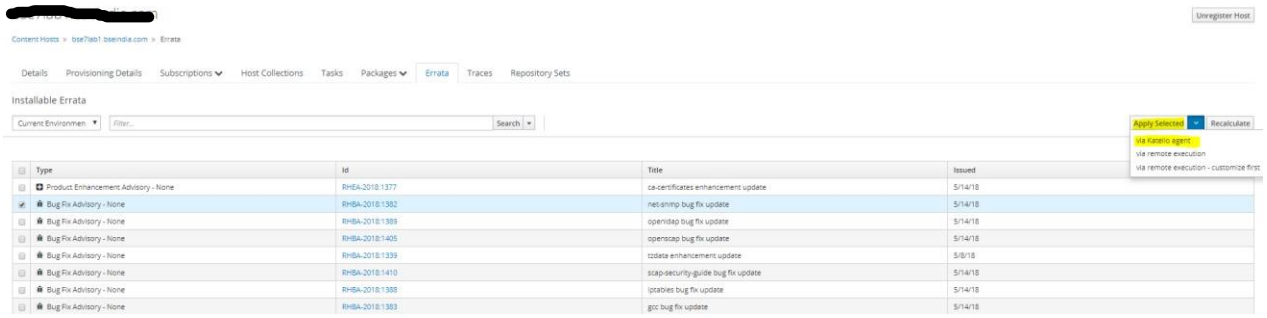


To Apply Installable Errata to a Content Host:

- Navigate to **Hosts** → **Content Hosts**.
- Click the name of the content host you want to manage.



- On the **Errata** tab, select errata you want to install.



- 4. Click **Apply Selected** to install the selected updates.

1.5.2. Adding Errata to a Host Collection

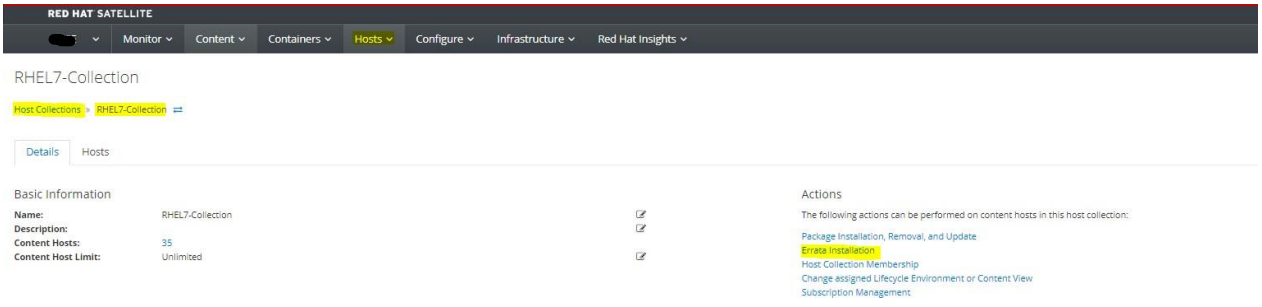
The following procedure shows how to add errata to host collections.

Prerequisites

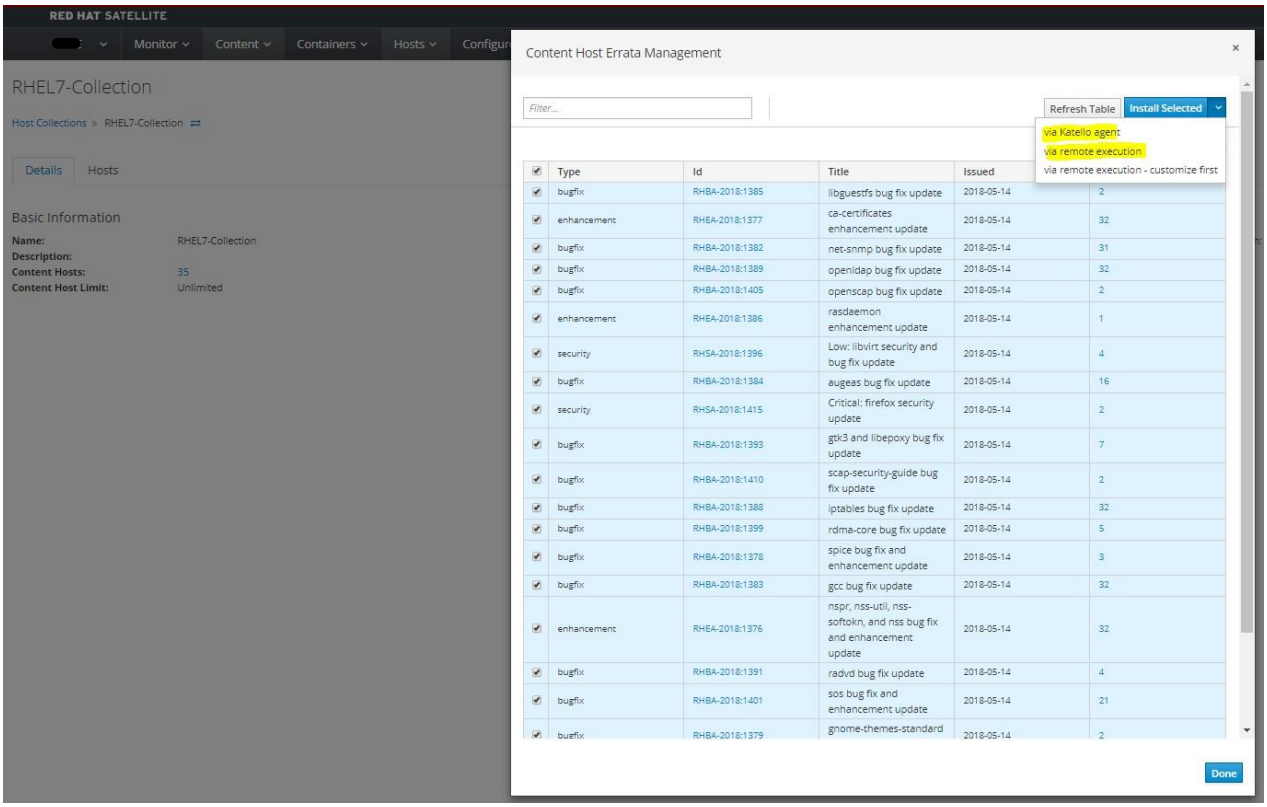
The errata to be added should be available in one of the existing repositories or added prior to this procedure. Errata should be promoted to the environment where the hosts are assigned.

To Add Errata to a Host Collection:

- Click **Hosts** → **Host Collections**.
- Select the host collection where the errata should be added.
- Click **Errata Installation**.



- Select the errata you want to add to the host collection and click the **Install Selected** button to use the default method. Alternatively, select the drop-down icon to the right of the button to select a method to use. Selecting **via remote execution – customize first** menu entry will take you to the **Job invocation** page where you can customize the action.



1.6. Configuring Activation Keys

Activation keys define selected properties of content hosts. You can use activation keys during content host registration to improve the speed, simplicity and consistency of the process.

- Activation can specify:
- Associated subscriptions and subscription attach behavior.
- Available products and repositories.
- A life cycle environment and a content view.
- Host collection membership.

The same activation key can be applied to multiple content hosts, as long as it contains enough subscriptions. However, activation keys only set the initial configuration for a content host. When it is registered to an organization, other content which that organization possesses can be attached to the content host manually.

1.7. Creating an Activation Key

This section describes how to create an activation key.

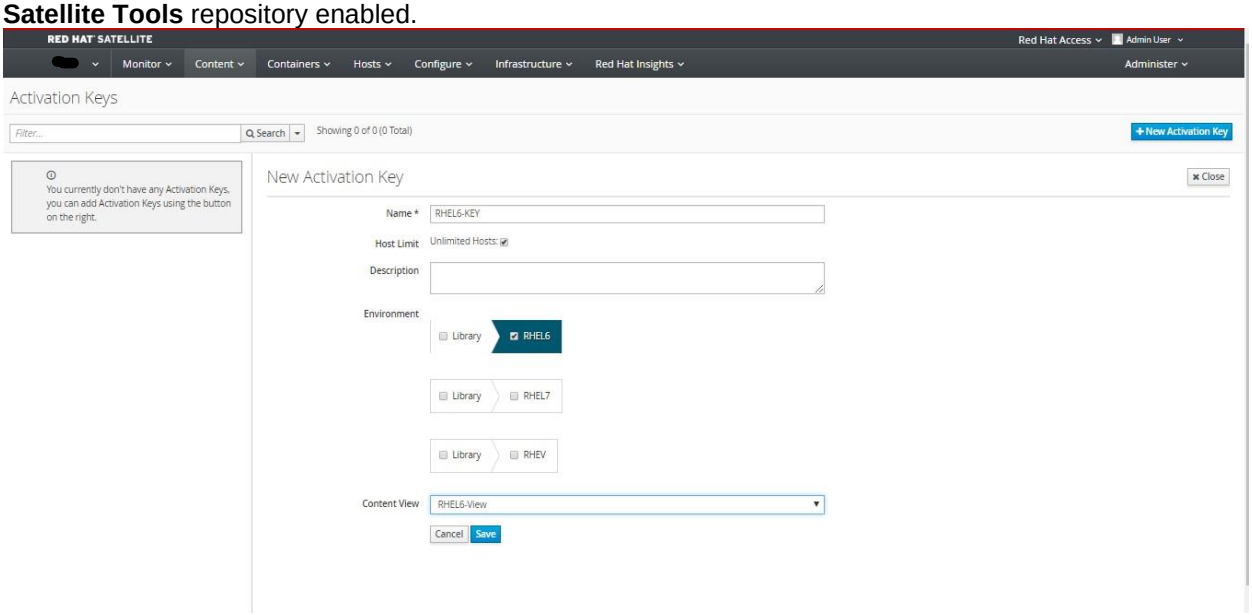
To Create an Activation Key:

- Click **Content** → **Activation keys**.
- Click **New Activation Key**. Perform the following actions:

a. Specify the activation key name. This setting is required.

b. Optionally, clear the **Unlimited Hosts** check box if you want to limit the number of host that can be associated with the activation key. Specify the number in the **Limit** field. .

c. Optionally, enter a suitable description in the **Description** field. You can also select the **Environment** and **Content View** to which this key should apply. For host registration, select a content view that has the **Satellite Tools** repository enabled.

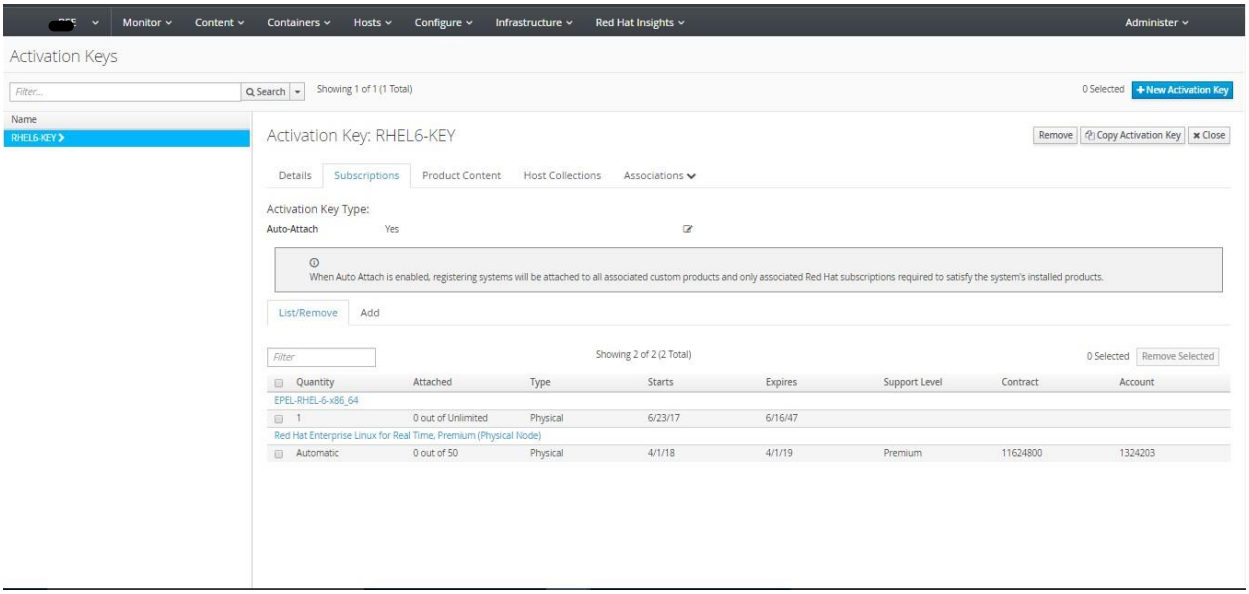


- Click **Save** to create the activation key.

1.7.1. Adding and Removing Subscriptions

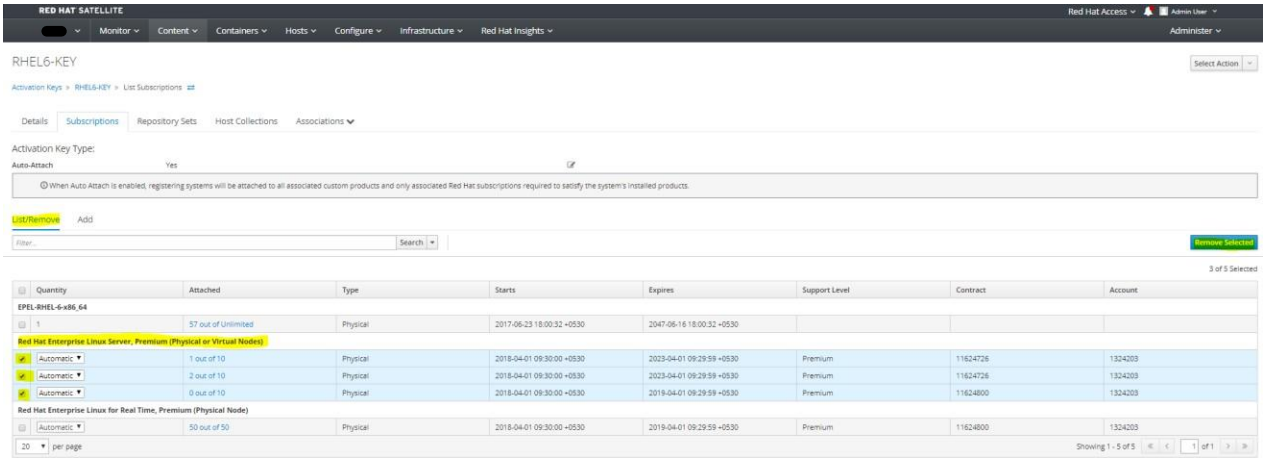
To Add a Subscription to an Activation Key:
Click **Content** → **Activation keys**.
Click the activation key name you want to edit. On the **Subscriptions** tab, select the **Add** subtab.

- From the list of available subscriptions, select the subscriptions you want to add.
- Click **Add Selected**.



1.7.2. To Remove Subscriptions from an Activation Key:

- Click **Content** → **Activation keys**.
- A list of activation keys is displayed. Click the activation key you want to remove subscriptions from.
- Click the **Subscriptions** tab.
- Under the **List/Remove** subtab, a list of attached subscriptions is displayed. Select the subscriptions to be removed.
- Click **Remove Selected**.

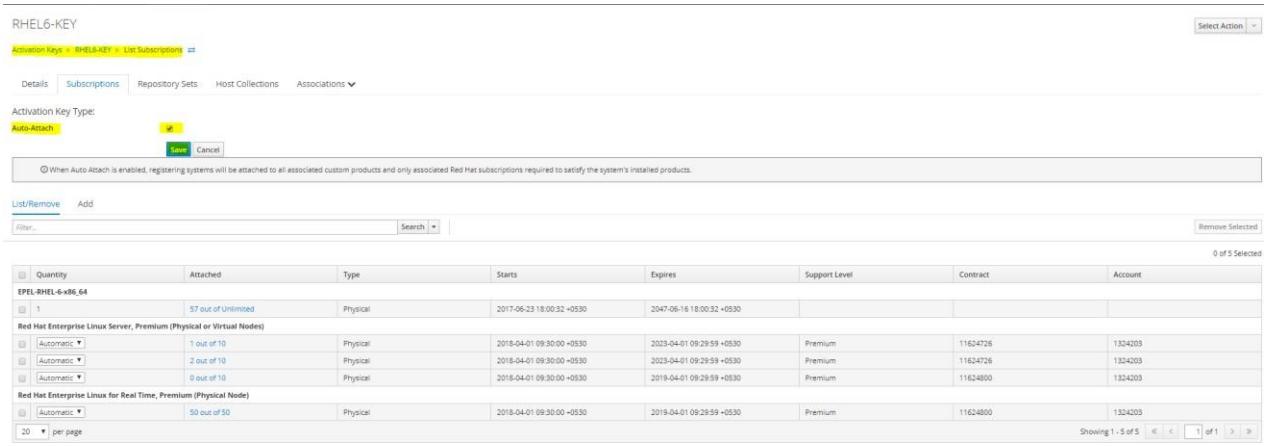


1.7.3. Enabling Auto-Attach

The auto-attach setting of an activation key determines what subscriptions are automatically attached during registration.

To Enable Auto-Attach on an Activation Key:

- Click **Content** → **Activation keys**.
- Click the activation key name that you want to edit.
- Click the **Subscriptions** tab.
- Click the edit icon next to **Auto-Attach**.
- Select the check box to enable auto-attach.
- Click **Save**.

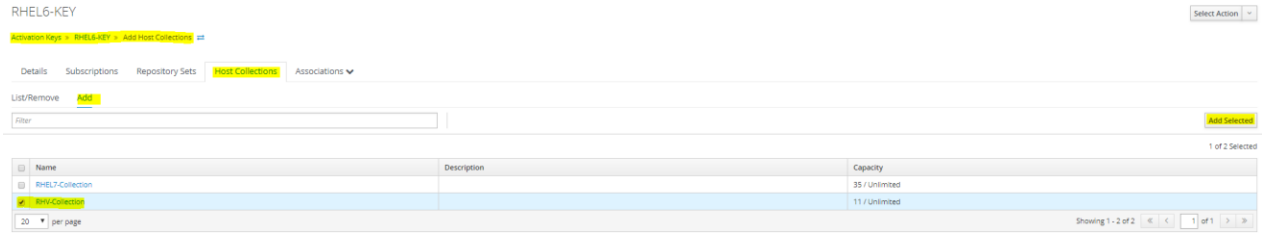


1.7.4. Adding and Removing Host Collections

These steps show how to add host collections to an activation key. Host collections can be associated with activation keys so that hosts utilizing the activation keys will automatically be added to the associated host collections upon registering with the Satellite Server.

To Add Host Collections to an Activation Key:

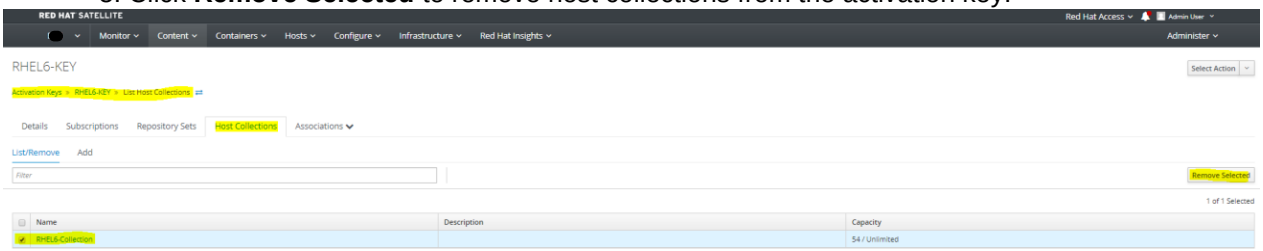
- Click **Content** → **Activation keys**.
- Click the activation key that you want to add a host collection to.
- On the **Host Collections** tab click the **Add** subtab to display the list of available host collections.
- Select the host collections you want to add, and then click **Add Selected**.



To Remove Host Collections from the Activation Key:

- Click **Content** → **Activation keys**.
- A list of activation keys is displayed. Click the activation key you want to remove host collections from.
- Click the **Host Collections** tab.

- 4. Under **List/Remove** subtab, a list of host collections attached to the activation key is displayed. Select the check box of the host collections you want to remove.
- 5. Click **Remove Selected** to remove host collections from the activation key.

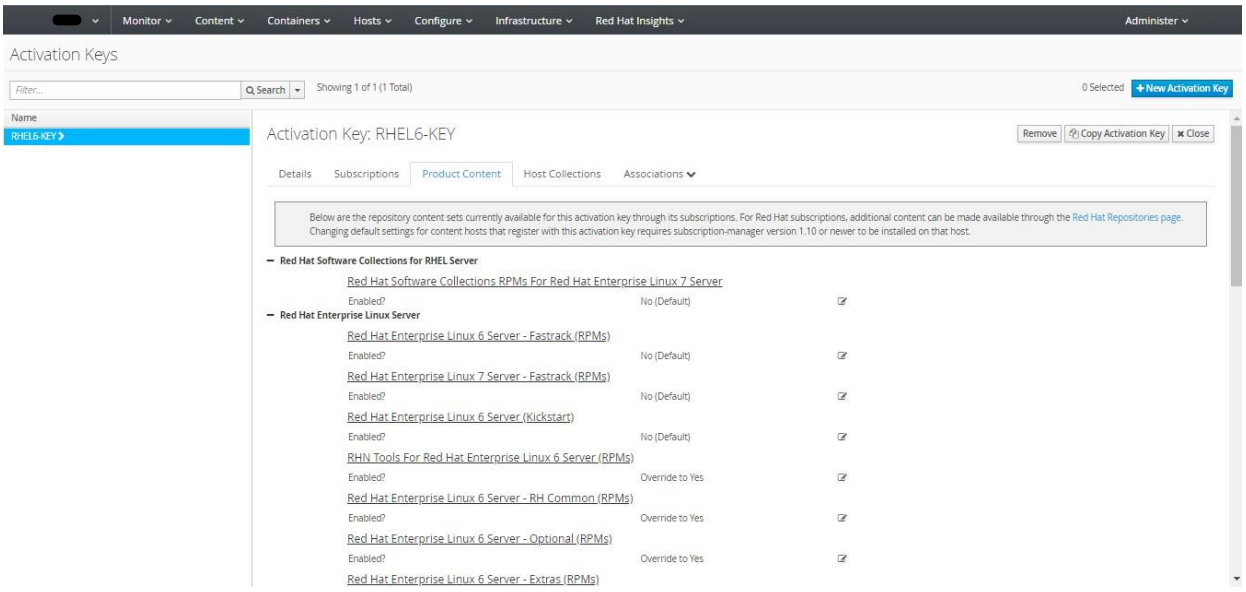


1.7.5. Editing Product Content

The number of products available for the activation key is determined by associated subscriptions. You can change which repositories in products are enabled on the **Product Content** tab.

To Edit Product Content on an Activation Key:

- Click **Content** → **Activation keys**.
- Click the activation key name that you want to edit.
- Click the **Product Content** tab to view the products and repositories associated with the activation key through subscriptions.
- Click the edit icon next to the repository you want to edit.
- From the drop-down menu, select if the repository will be enabled or disabled. Click **Save** to apply the change.

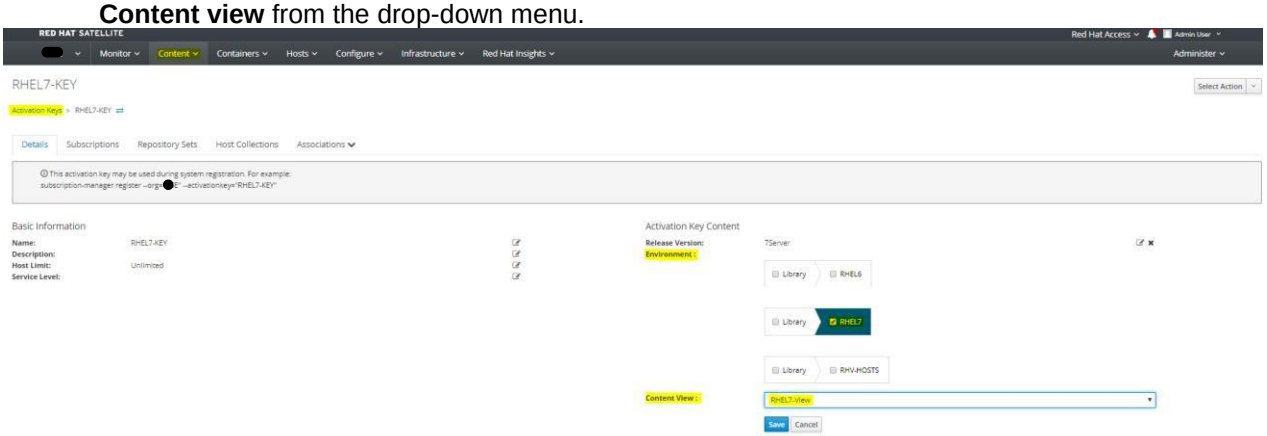


1.7.6. Setting a Life Cycle Environment and a Content View

You can set a life cycle environment and a content view for an activation key when creating it.

It is also possible to modify these settings afterwards using the following procedure.

- Click **Content** → **Activation keys**.
- Click the activation key name that you want to edit.
- Click the check box next to the **Environment** you want to associate with the activation key. Select a **Content view** from the drop-down menu.



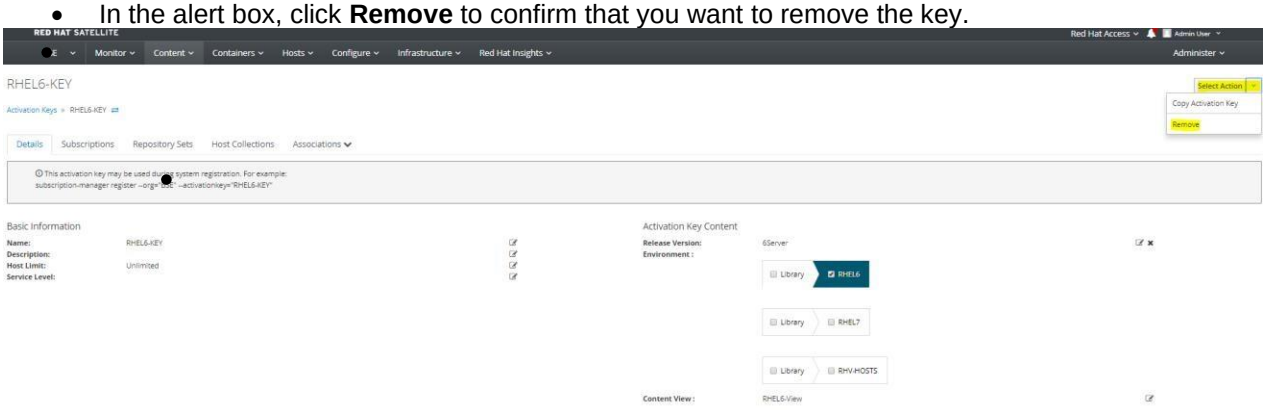
- 4. Click **Save**.

1.7.7. Removing an Activation Key

This section describes how to remove an activation key.

To Remove an Activation Key:

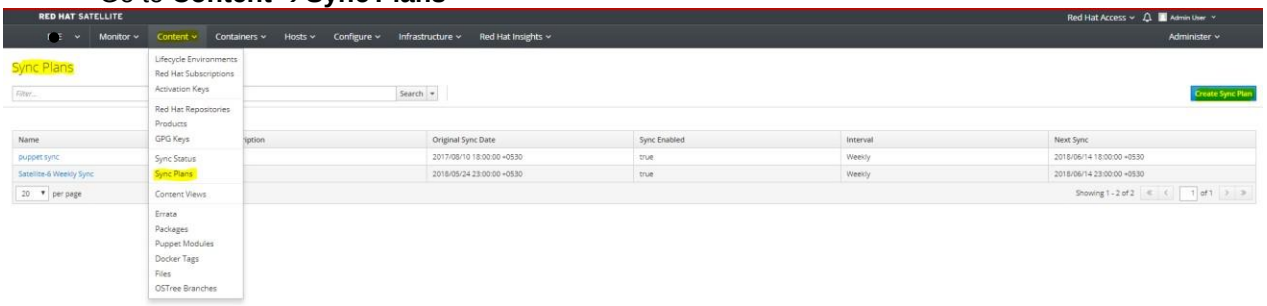
- Click **Content** → **Activation keys**.
- Click the activation key name that you want to remove.
- In the upper right of the **Activation Key** details panel, click **Select Action**.
- In the alert box, click **Remove** to confirm that you want to remove the key.



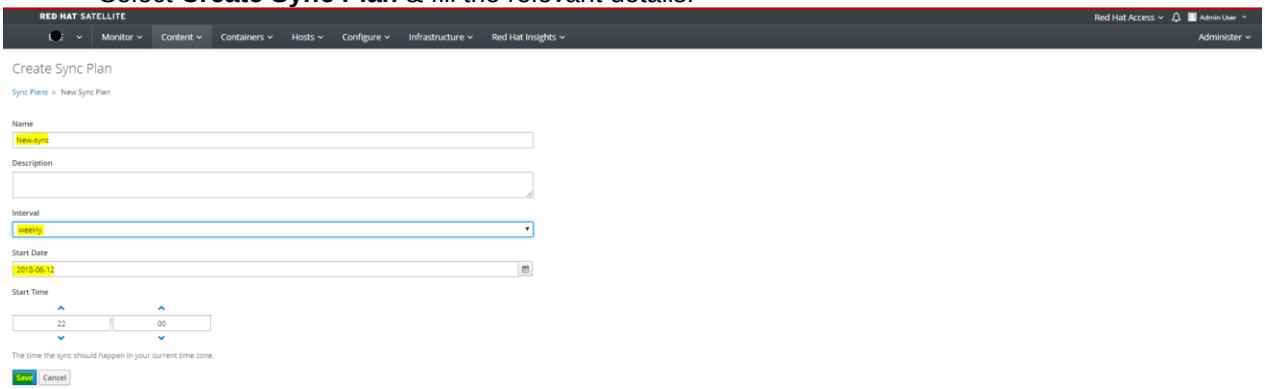
1.7.8. Creating Repository Sync Plan

This Section describes how to create a sync plan

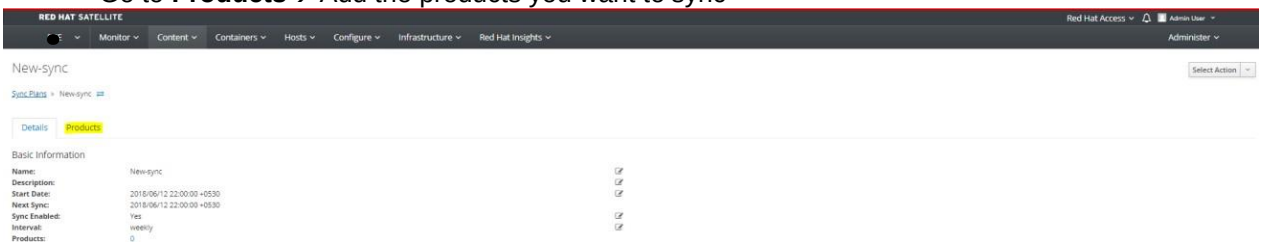
- Go to **Content →Sync Plans**



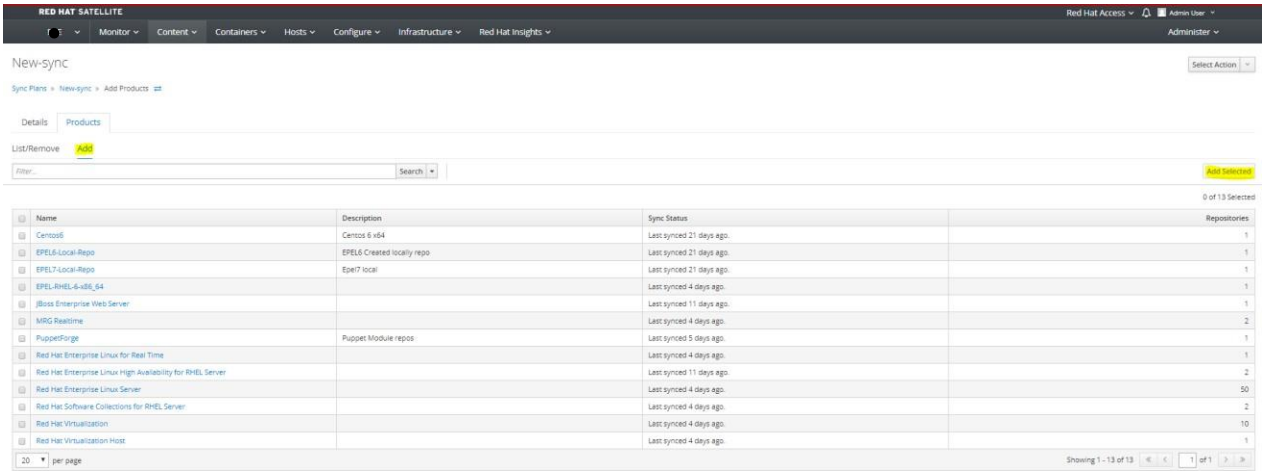
- Select **Create Sync Plan** & fill the relevant details.



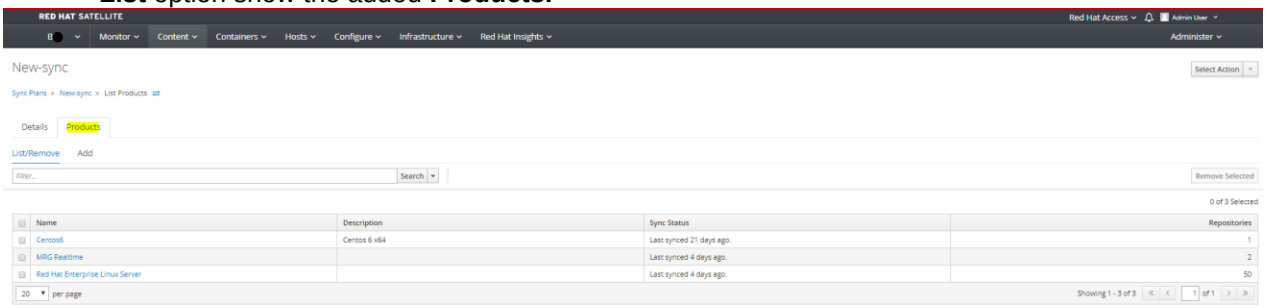
- Go to **Products→** Add the products you want to sync



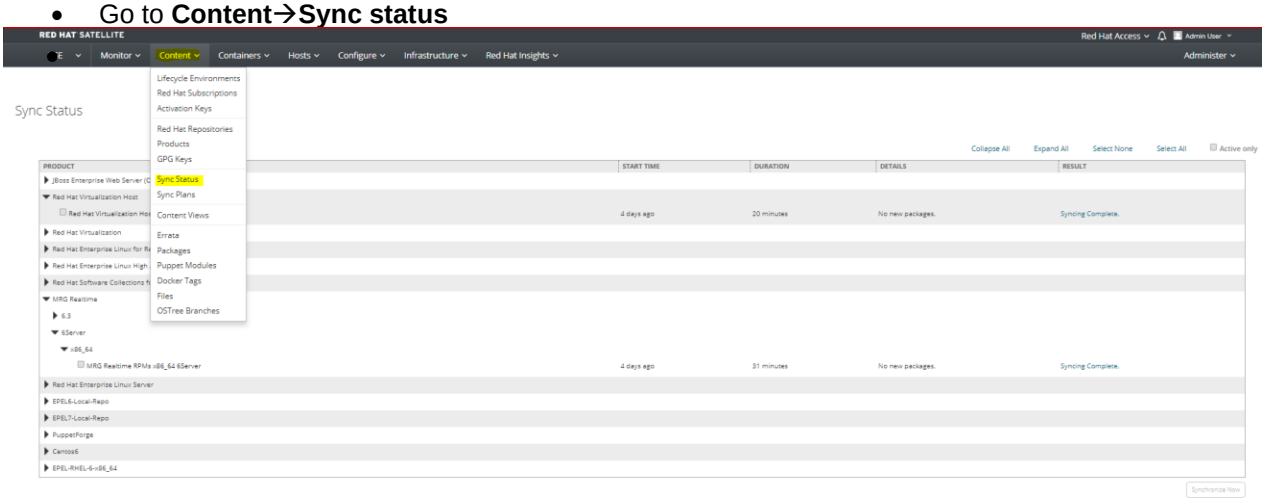
- Click **Add Selected**



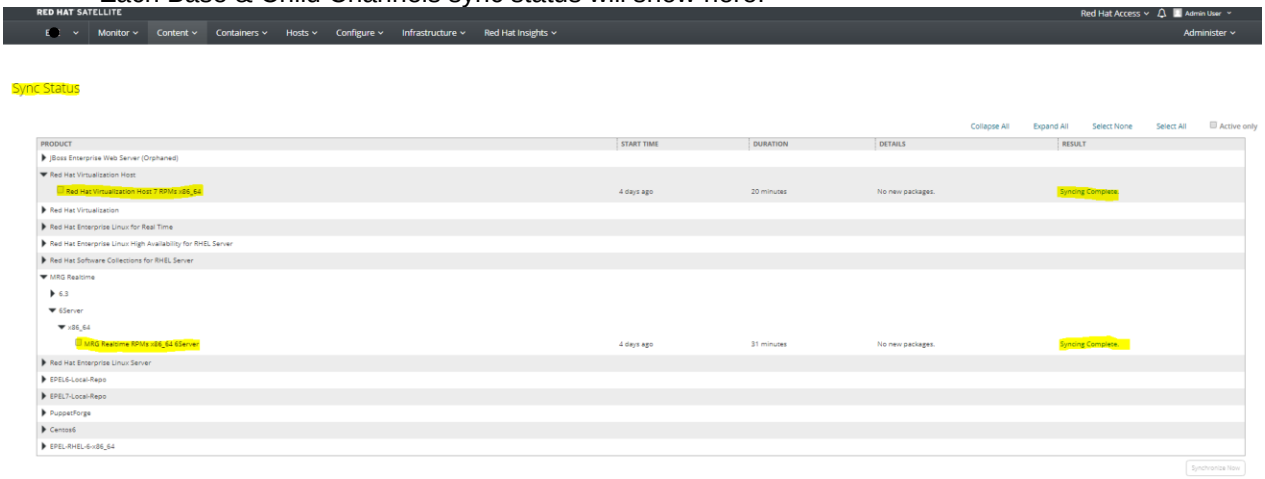
- **List** option show the added **Products.**



- After Sync you can check the status on sync status.
- Go to **Content→Sync status**



- Each Base & Child Channels sync status will show here.



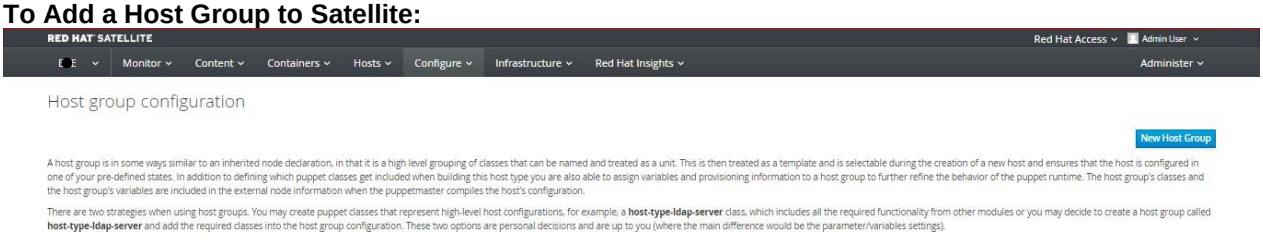
1.8. Configuring the Provisioning Environment

Provisioning refers to a process that starts with a bare physical or virtual compute resource and ends with a fully configured, ready-to-use operating system. Red Hat Satellite provides an ability to define and automate fine-grained provisioning for a large number of hosts.

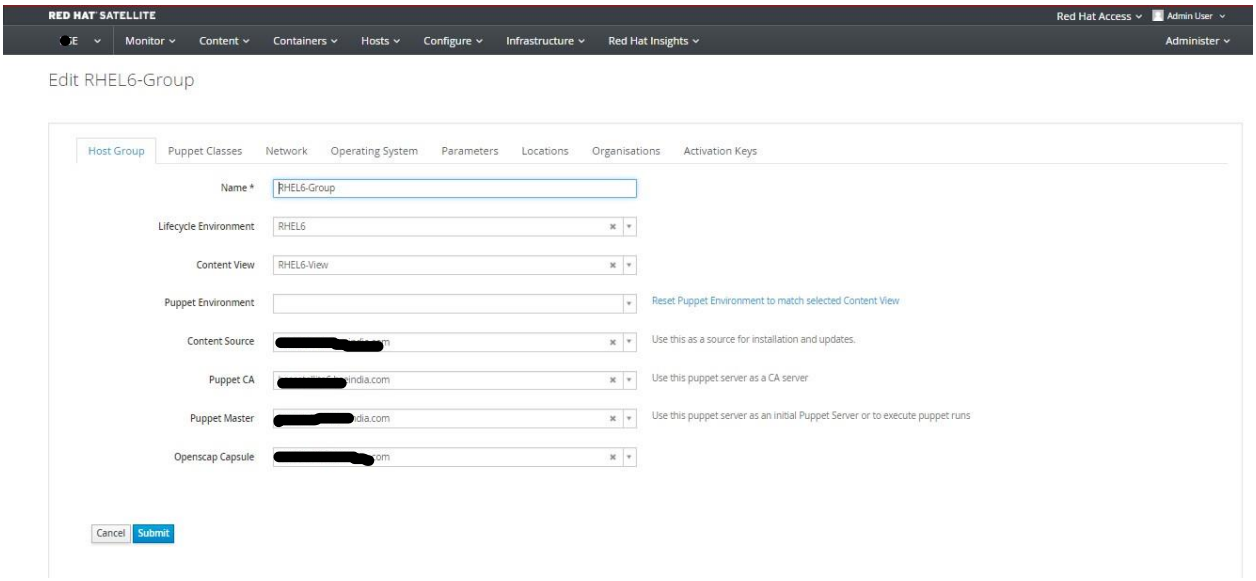
1.8.1. Creating a Host Group

A host group defines a set of default values that hosts inherit when placed in that group. Hosts can belong to only one host group, but host groups can be nested in hierarchies. You can create a "base" or "parent" host group that represents all hosts in your organization, and then create nested or "child" host groups under that parent to provide specific settings. This section describes how to create a host group.

To Add a Host Group to Satellite:



- Click **Configure** → **Host Groups** and then click **New Host Group**.
- Enter the required details for the Host Group, and then click **Submit**.



1.8.2. Host Group Attributes

The following table describes the attributes that apply to Satellite Host Groups.

Submenu	Options	Description
Host Group	Parent	The parent Host Group for the new Host Group.
	Name	The name of the Host Group.
	Life Cycle Environment	The environment containing this Host Group.
	Puppet CA	The Red Hat Satellite Capsule Server to use for the Puppet CA server.
	Puppet Master	The Red Hat Satellite Capsule Server to use as the Puppet Master.
Puppet Classes	Included Classes	The Puppet Classes included with the Host Group.
	Available Classes	The Puppet Classes available to use with the Host Group.
Network	Domain	The domain for hosts in the Host Group.
	Subnet	The subnet for hosts in the Host Group.
Operating System	Architecture	The default architecture for systems in the Host Group.
	Operating Systems	The default operating system for systems in the Host Group.
	Media	The location of the installation media for the operating system.
	Partition Table	A file system partition layout for the operating system installation.
	Root Password	The root password for the operating system.

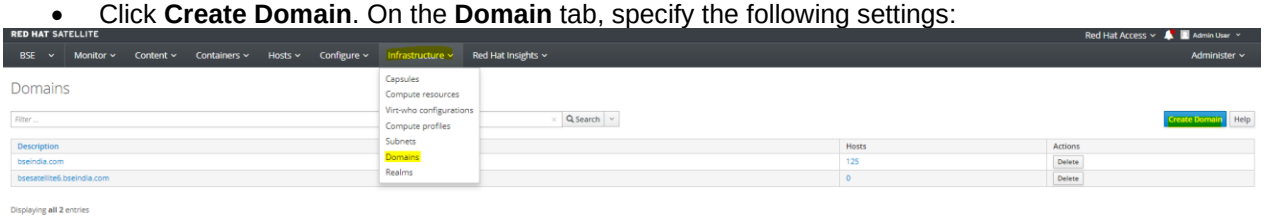
Parameters	Add Parameter	Provides a Name and Value pair to set parameters for the Host Group.
Organizations	Organizations	The organizations that own this host group.
Activation Keys	Content Environment	Defines the activation keys made available in templates as `@host.params['kt_activation_keys']`.

1.8.3. Domains

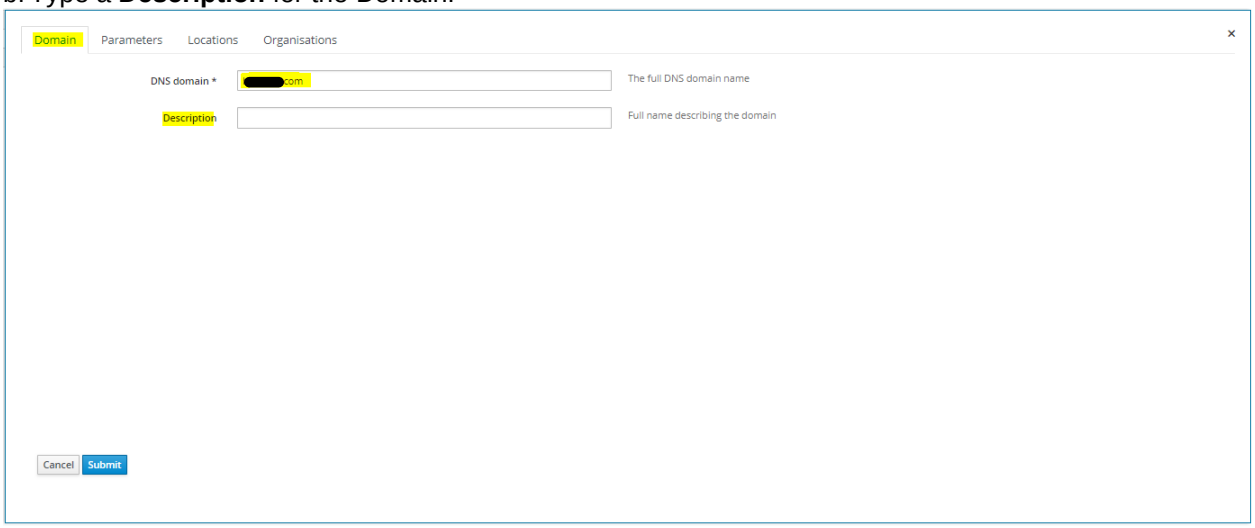
Satellite has the ability to assign domain names with Red Hat Satellite Capsule Server DNS. This provides users with a means to group and name hosts within a particular domain.

To Create a Domain:

- Click **Infrastructure** → **Domains**.
- Click **Create Domain**. On the **Domain** tab, specify the following settings:

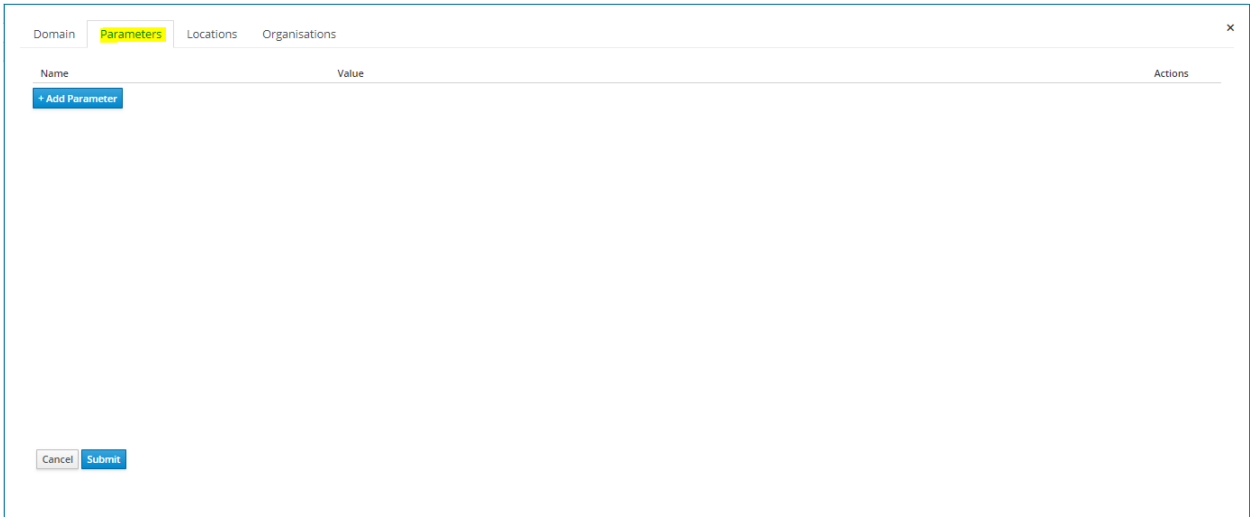


a. Specify a **Name** for the Domain. This is the required DNS domain name.
b. Type a **Description** for the Domain.

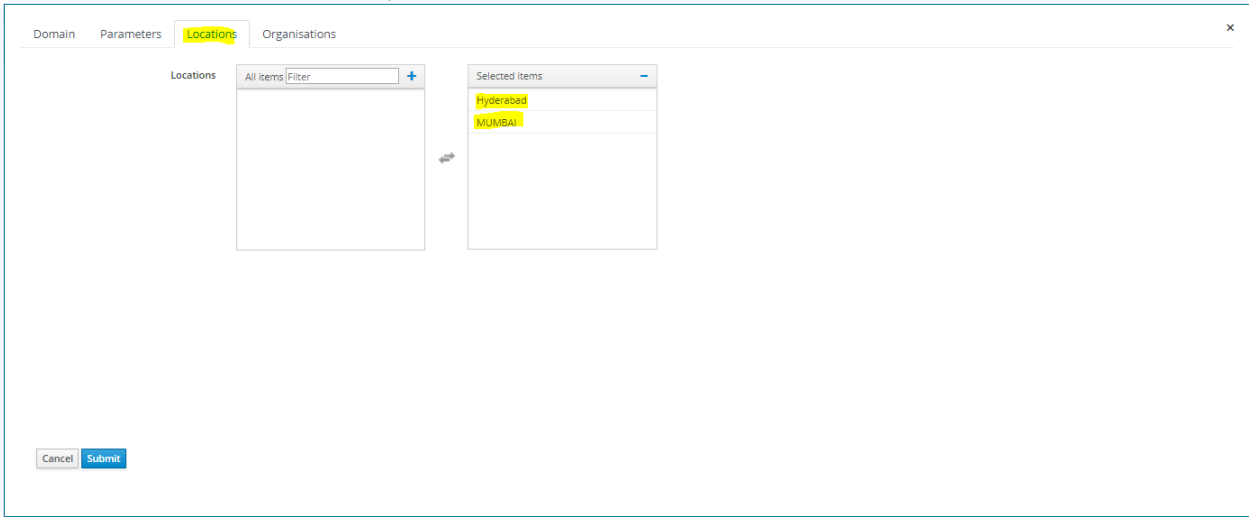


c. Select a DNS-enabled Capsule Server.

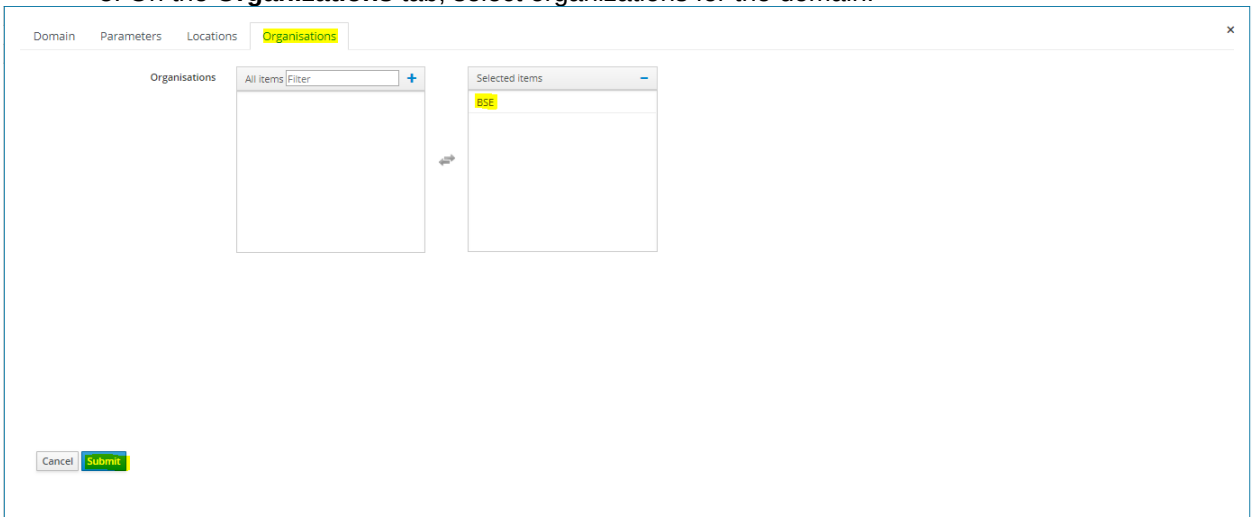
- On the **Parameters** tab, specify domain parameters.



- 4. On the **Locations** tab, select locations for the domain.



- 5. On the **Organizations** tab, select organizations for the domain.



- Click **Submit**.

1.8.4. Subnets

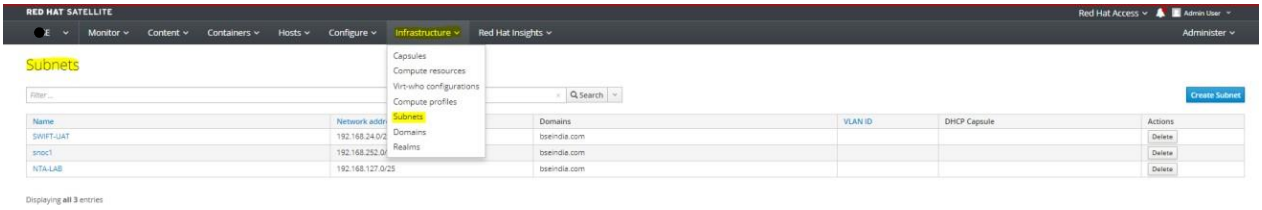
Satellite has the ability to create networks for groups of systems. Subnets use standard IP address settings to define the network and use the Red Hat Satellite Capsule Server's DHCP features to assign IP addresses to systems within the subnet.

1.8.5. Creating a Subnet

The following procedure shows how to create a subnet.

To Create a Subnet:

- Click **Infrastructure → Subnets**.

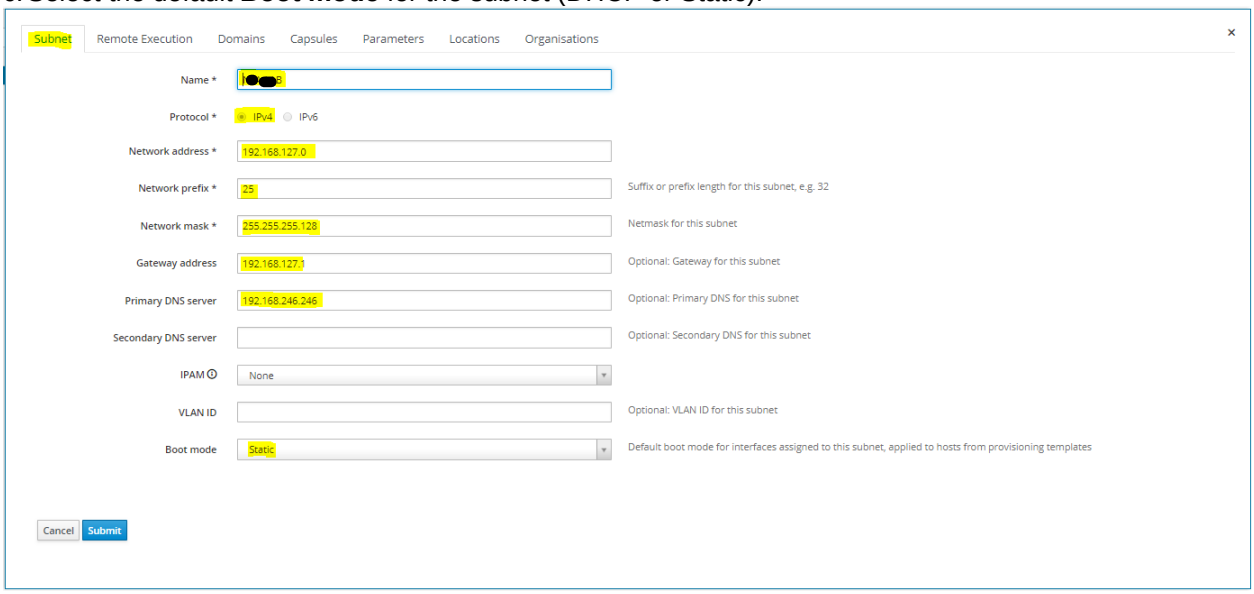


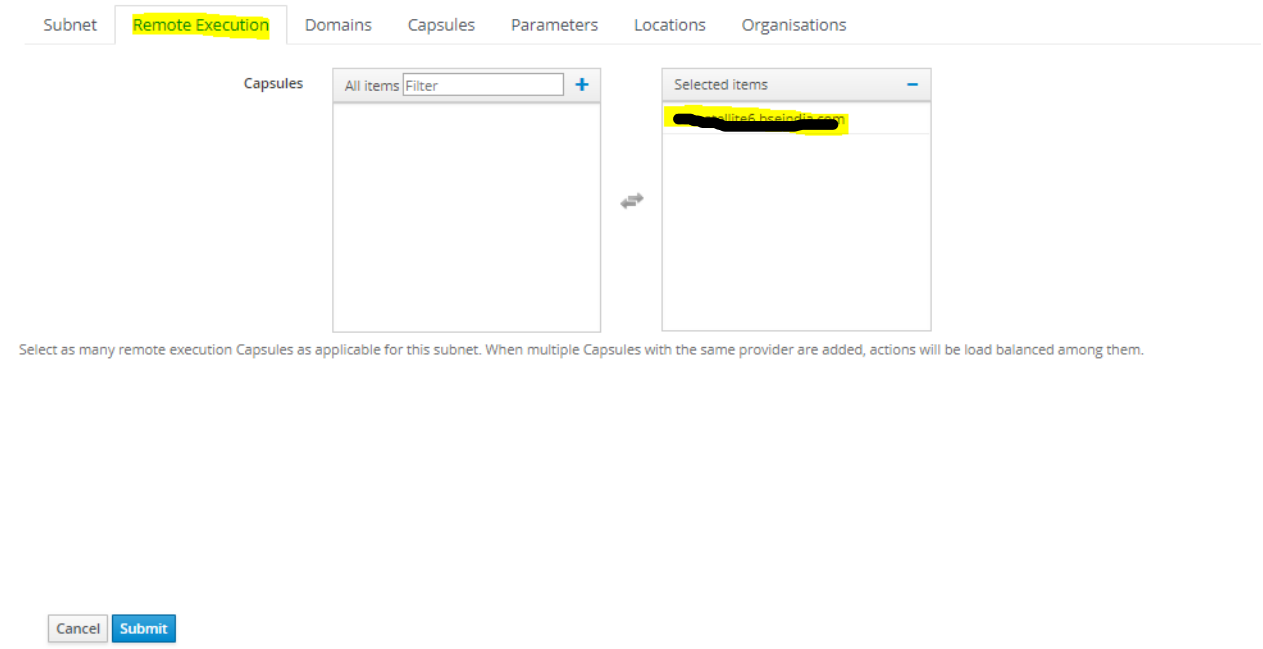
- Click New Subnet. On the Subnet tab, specify the following settings:

a. Specify a **Name**, **Network address** (IP address), and **Network mask** for the subnet. These settings are required.

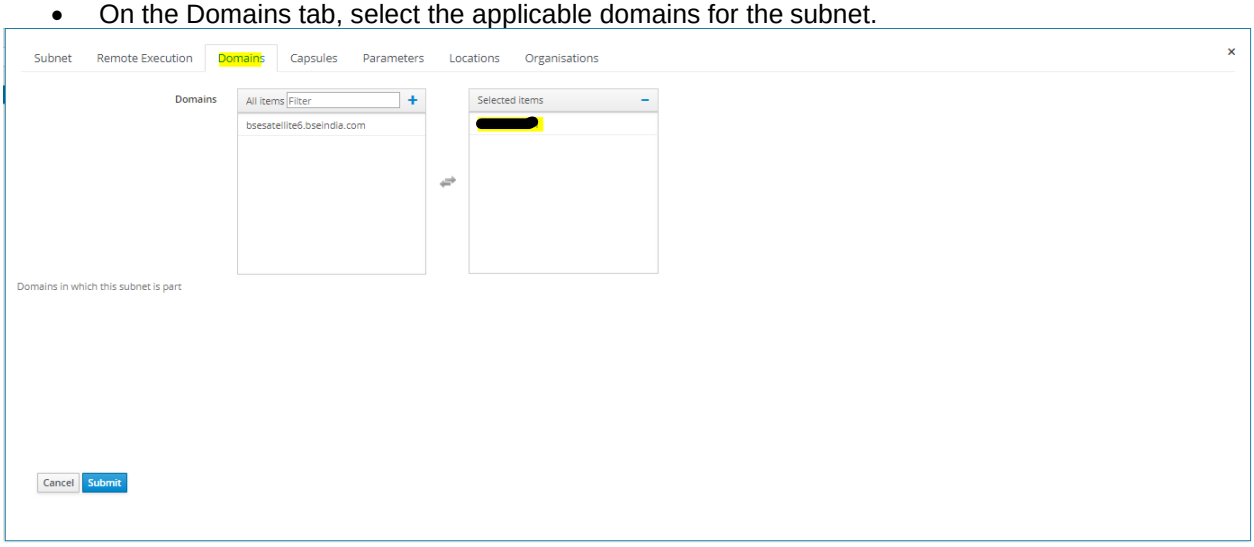
b. Optionally, specify the **Gateway address**, **Primary DNS server**, **Secondary DNS server**, and **VLAN ID**. Note that the gateway address and DNS server settings are optional only with IPAM and Boot modes set to DHCP (default). If you decide to change these default modes, you also have to specify gateway and DNS. You can also select the **IPAM** mode (DHCP, Internal DB, or None) and define the IP assignment range with the **Start of IP range** and **End of IP range** fields.

c. Select the default **Boot mode** for the subnet (DHCP or Static).

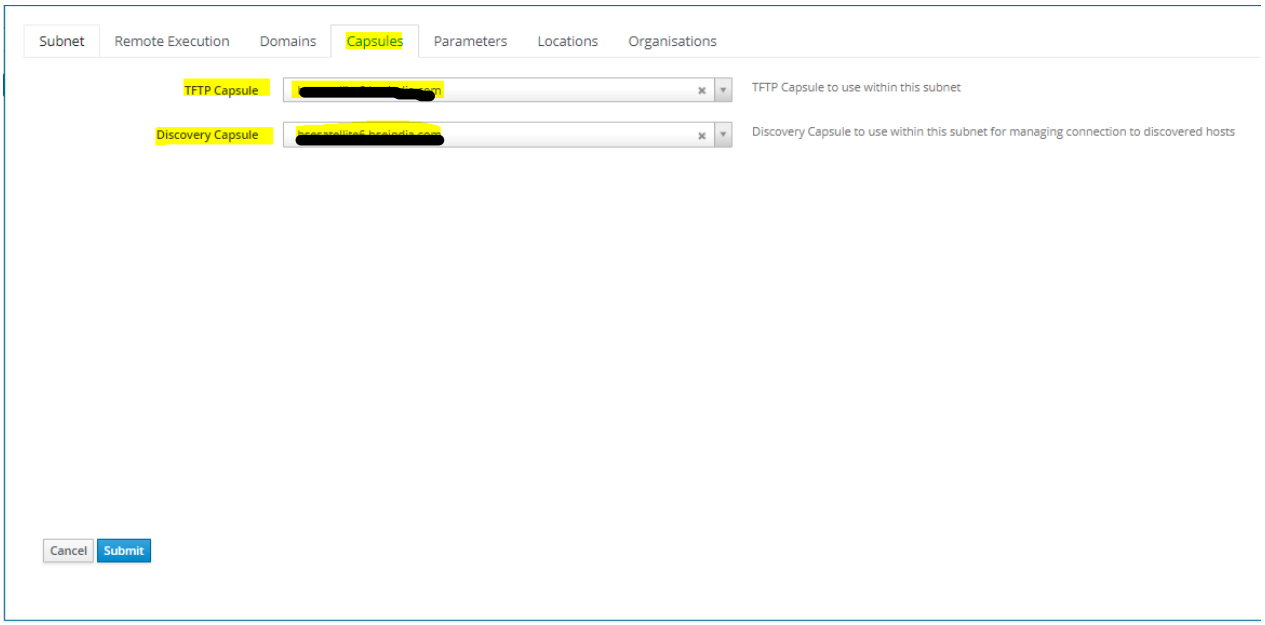




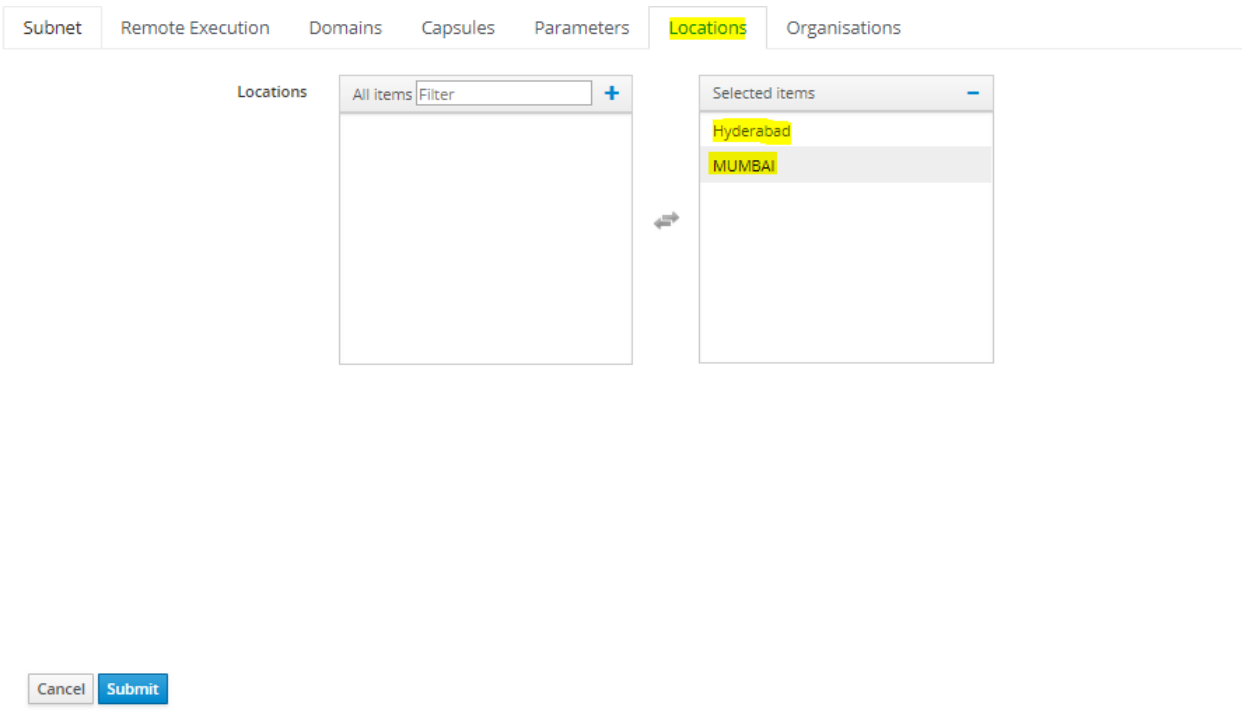
- On the Domains tab, select the applicable domains for the subnet.



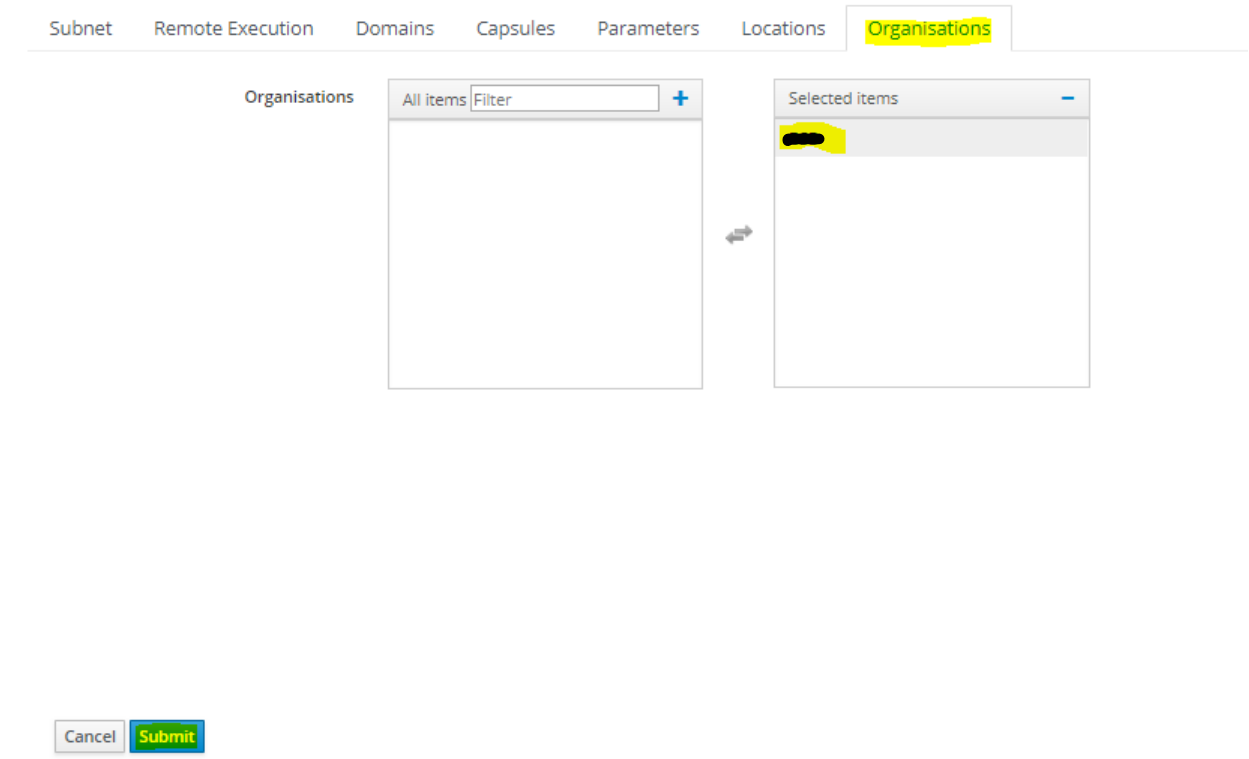
4. On the **Capsules** tab, select the Capsule Servers to be used for hosting the **DHCP Proxy**, **TFTP Proxy**, **DNS Proxy**, and **Discovery Proxy** services.



- 5. On the **Locations** tab, select locations for the subnet.



- 6. On the **Organizations** tab, select organizations for the subnet.



- 7. Click Submit.

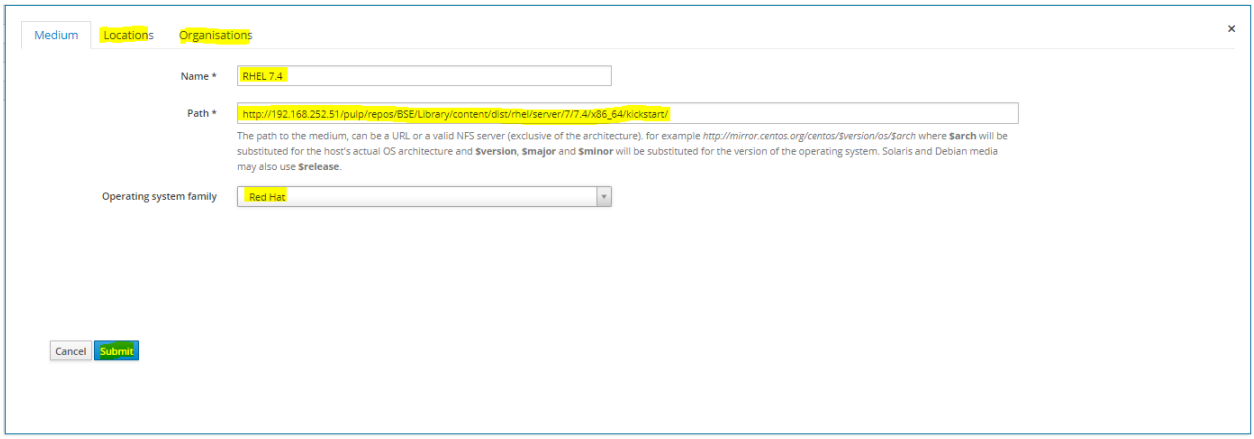
1.8.6. Installation Media

Red Hat Satellite uses installation media (ISO images) as content for kickstart trees and new host installations.
To Add an Installation Medium:

- Click **Hosts** → **Installation Media**.
- Click **Create Medium**. On the **Medium** tab, specify the following settings:

a. Type a **Name** for the Installation Media. This setting is required.
b. Type a **Path** to the Installation Medium. Options include either a URL or a valid NFS server. This setting is required.
c. Select an **Operating System Family** to define the type of the Installation Medium.

- On the **Locations** tab, select the desired locations to add them to the **Selected Items** list.
- On the **Organizations** tab, select the desired organizations to add them to the **Selected Items** list.
- Click **Submit**.

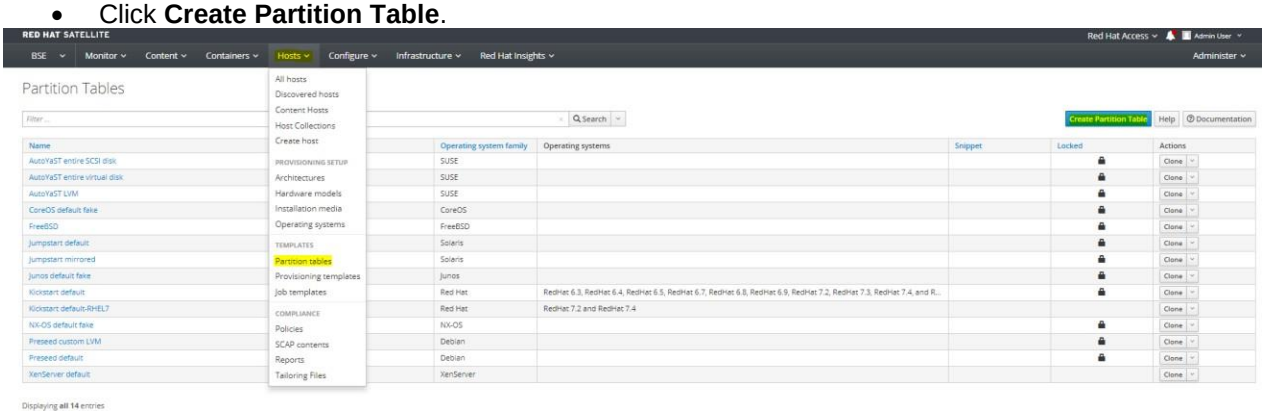


1.8.7. Partition Tables

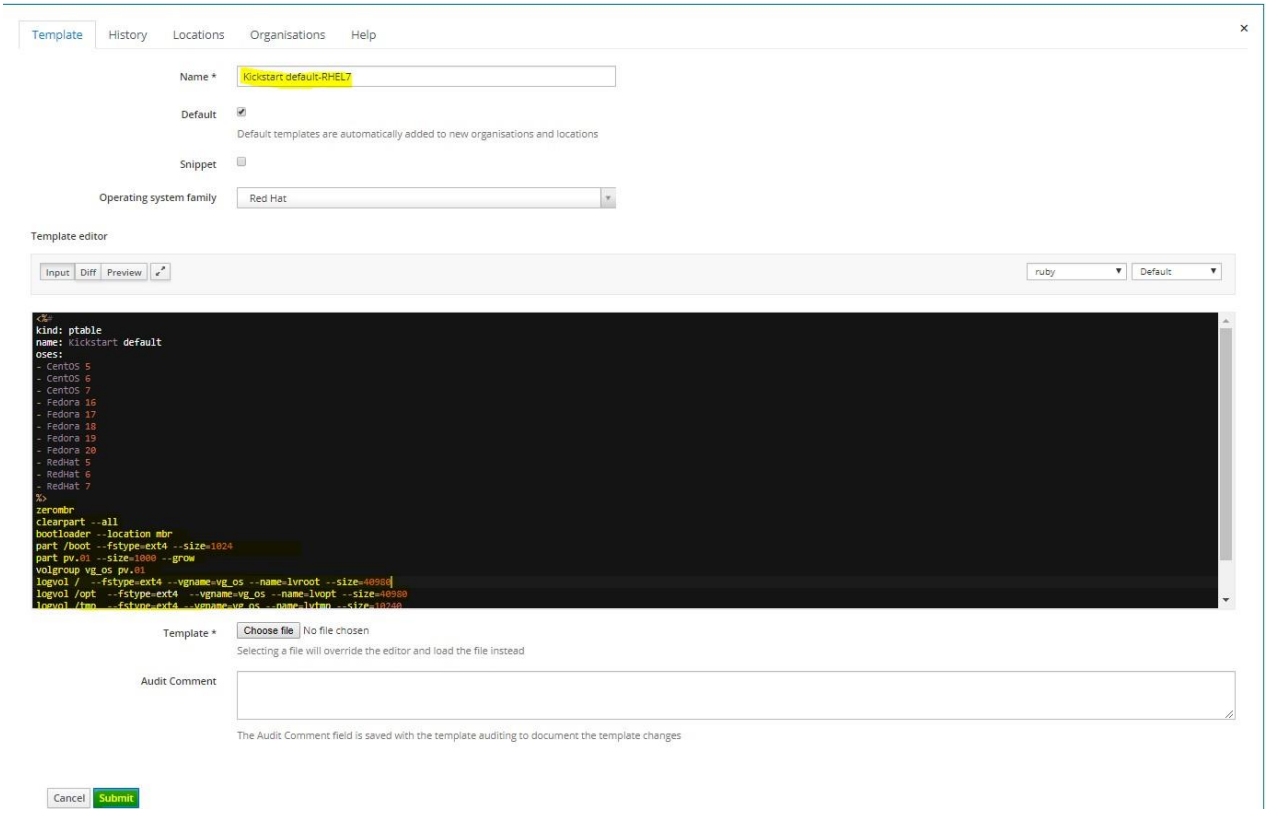
Partition tables define the partitions and file system layout for new installations when provisioning systems. Satellite users specify the host's disk layout as an explicit sequence of Partitions or use a dynamic disk layout script.

To Create a Partition Table:

- Click Hosts → Partition Tables.
- Click **Create Partition Table**.



- Type a **Name** for the partition table.
- Specify the **Layout** of the partition table. The **Layout** field also accepts dynamic disk partitioning scripts.
- Select the operating system from the **OS Family** drop-down list.
- Click **Submit**.

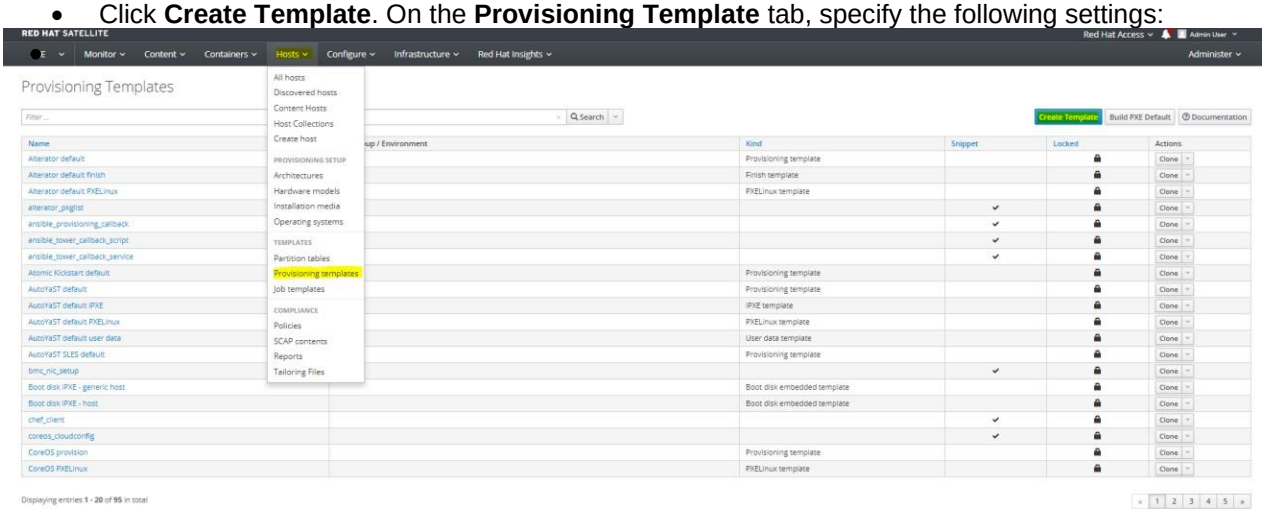


1.8.8. Provisioning Templates

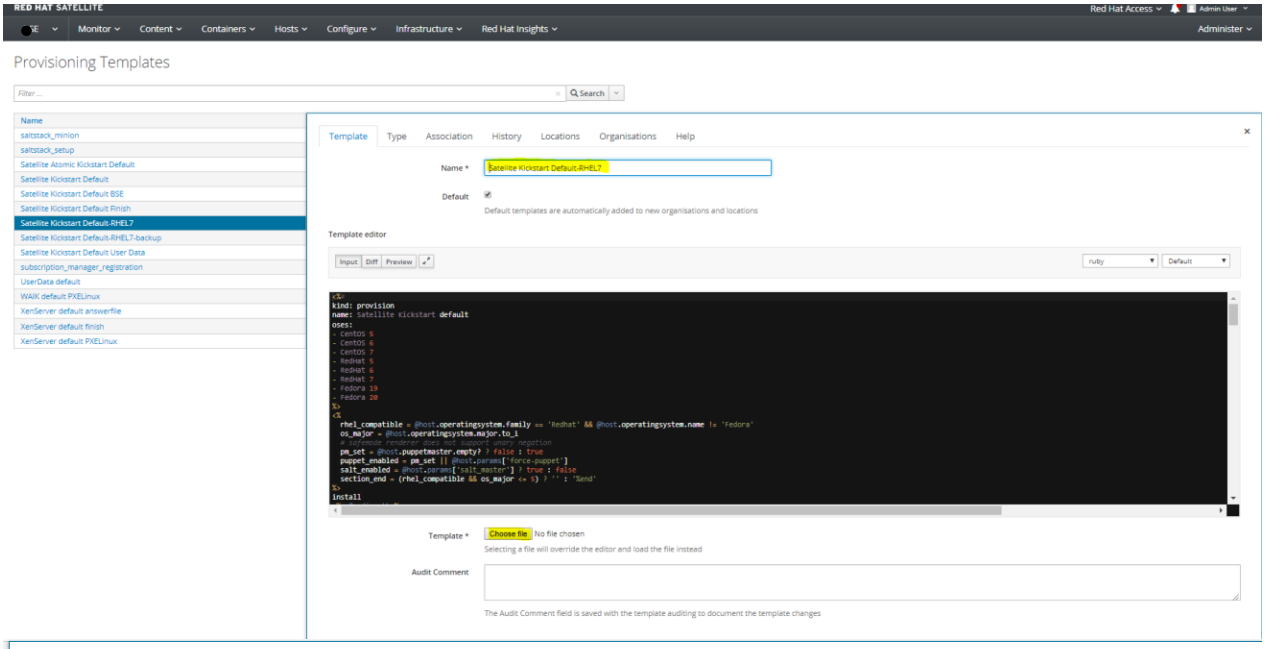
Provisioning templates provide the systematic means to run unattended installations.
Provisioning templates can be executed via several methods including bash scripts, kickstart scripts, and PXE-based installations.

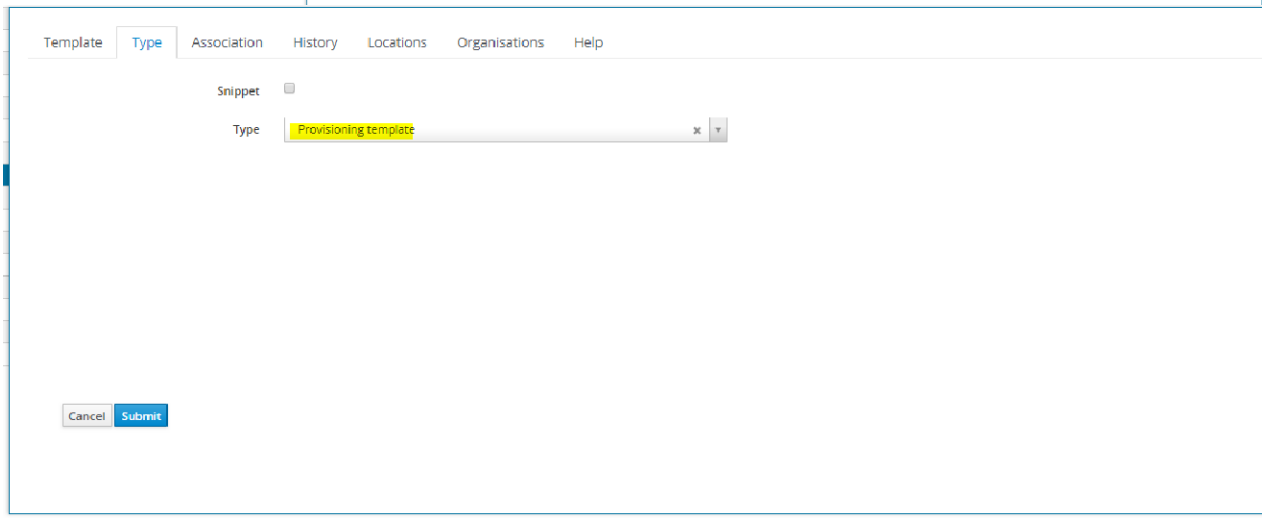
To Create a Provisioning Template:

- Click **Hosts** → **Provisioning Templates**.
- Click **Create Template**. On the **Provisioning Template** tab, specify the following settings:



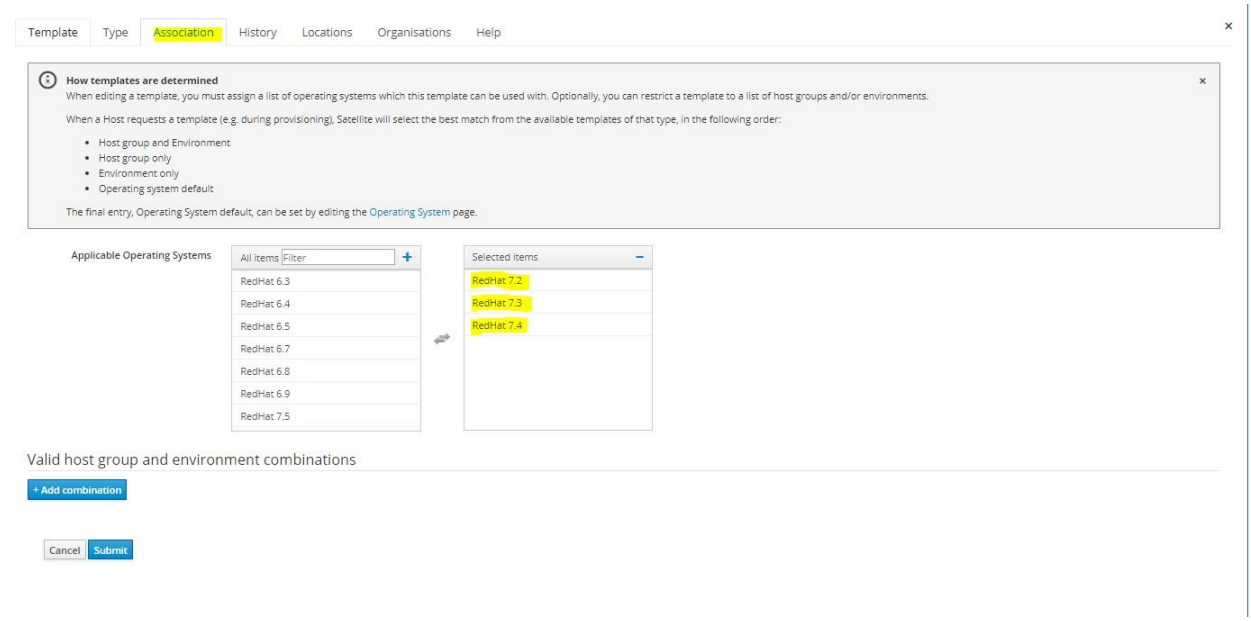
a. Specify a **Name** for the template.
b. Insert your template in the **Template editor** field. Alternatively, click **Browse** to upload the template. This replaces the content in the **Template editor** field with the content of your chosen file.



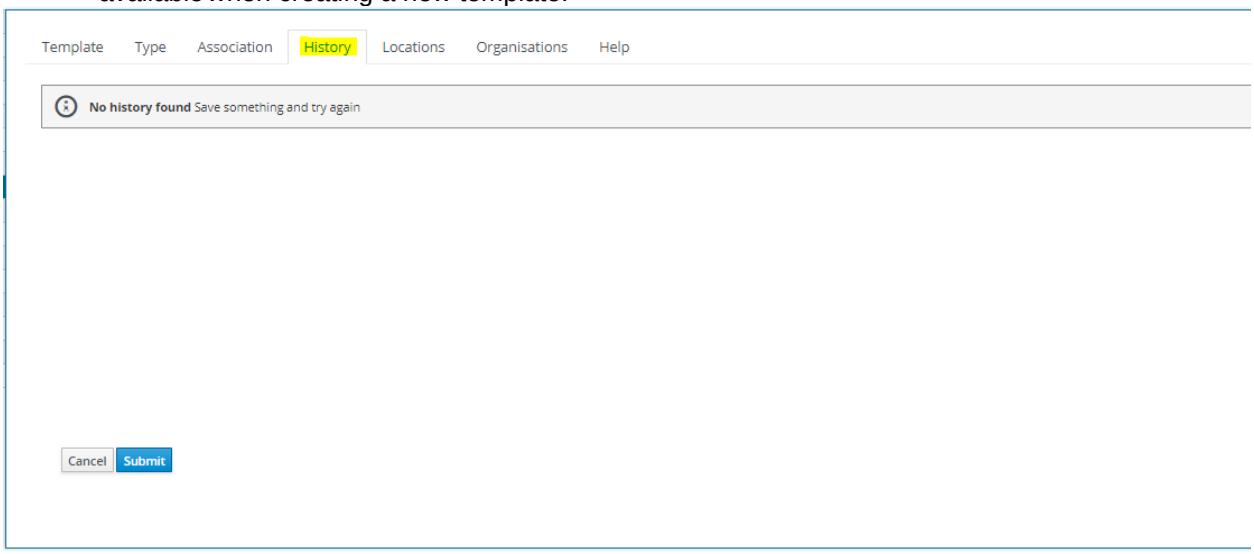


c. Optionally, type a comment in the **Audit Comment** field. Satellite adds the comment to the template history to track changes. View the template history under the **History** tab.

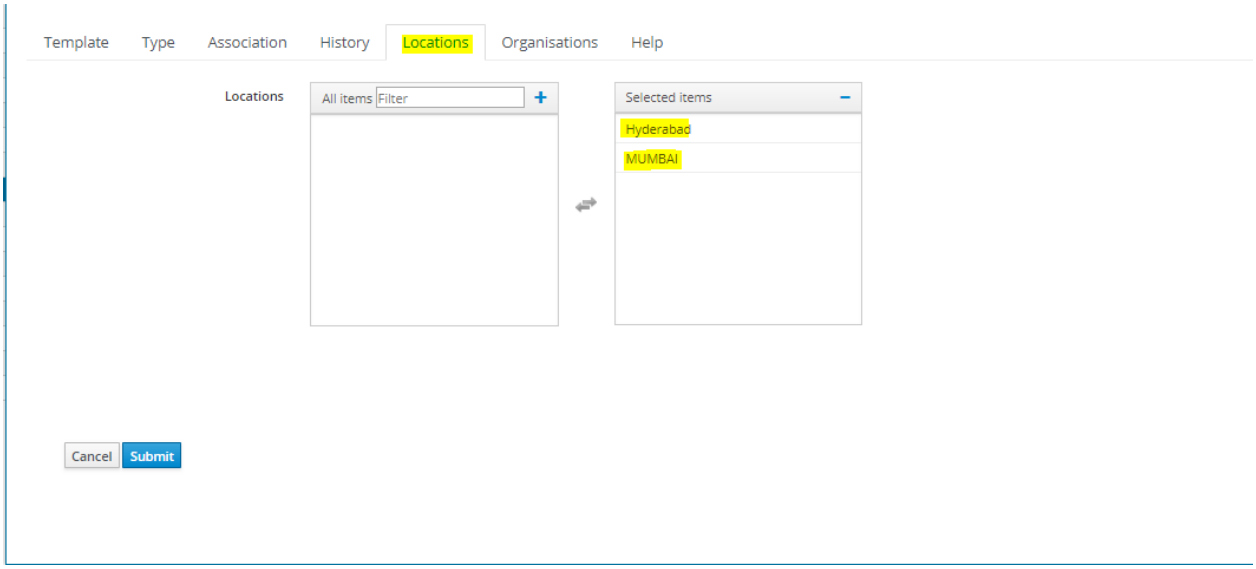
- On the **Type** tab, select **Snippet** to store the template code without defining it as particular script or template type, or select the type from the **Type** drop-down menu.
- On the **Association** tab, select host groups, environments and operating systems to be associated with the template. Select the operating systems from the **Applicable Operating Systems** list. Click **Add Combination** and select a **Hostgroup** and **Environment** to limit the template's use. Note that associations are not available for templates of type snippet.



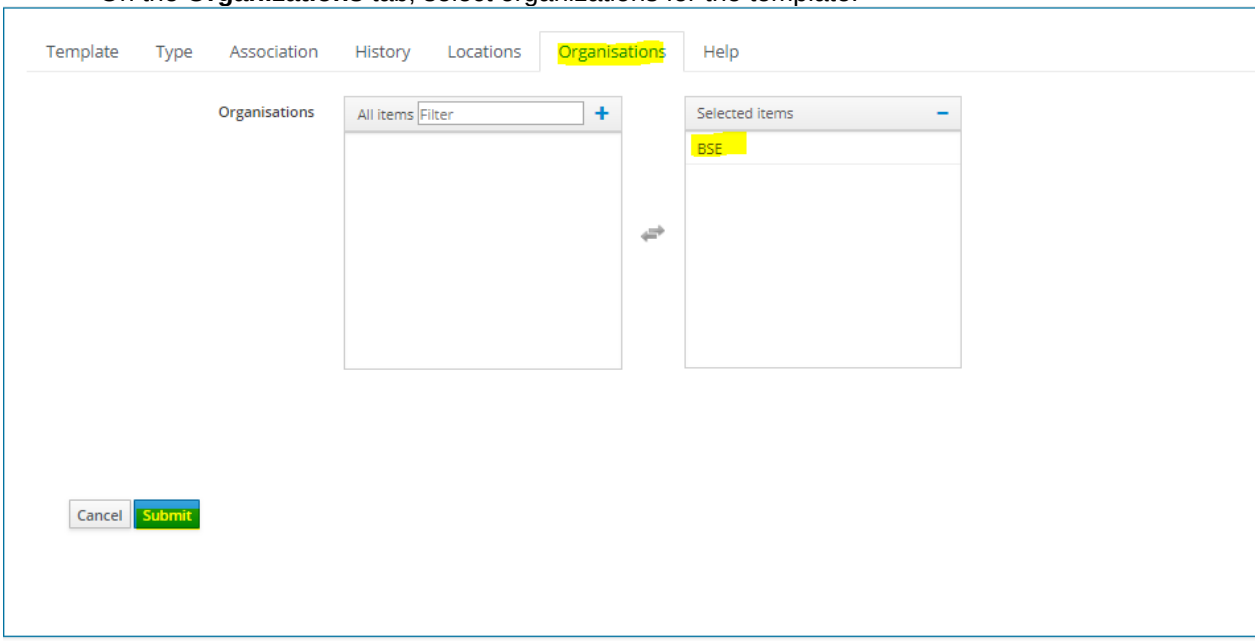
- On the **Association** tab, you can view the history of existing templates. No history is available when creating a new template.



- On the **Locations** tab, select locations for the template.



- On the **Organizations** tab, select organizations for the template.



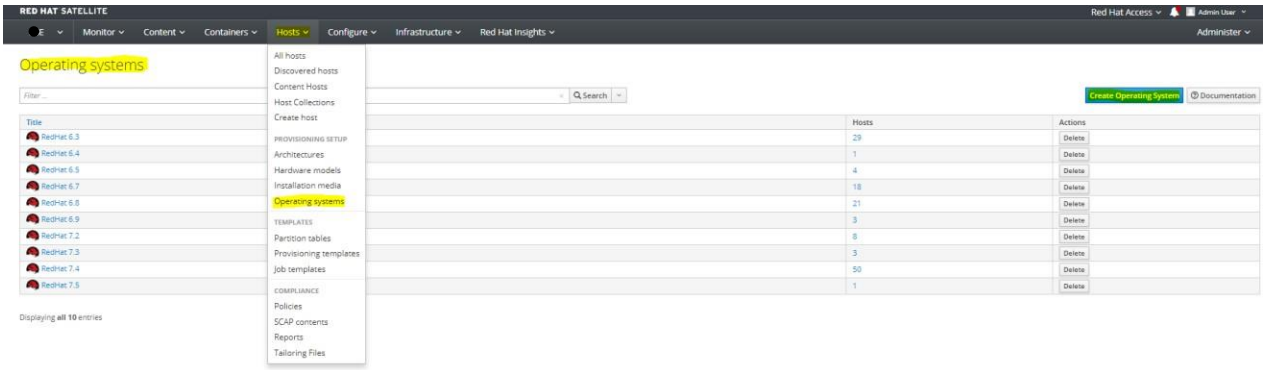
- Click **Submit**.

1.8.9. Operating Systems

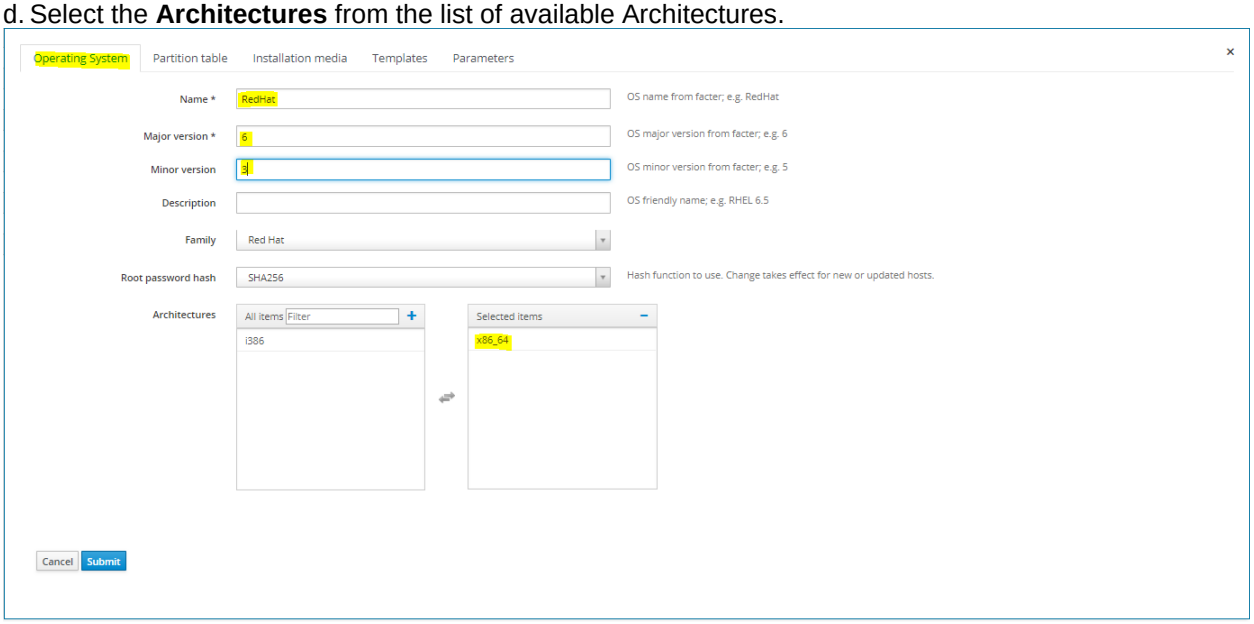
Operating Systems define combinations of installation methods and media and are grouped within families. As a default, Red Hat Satellite uses a **RedHat** family. Families allow Satellite to change certain behaviors when provisioning hosts.

To Add an Operating System:

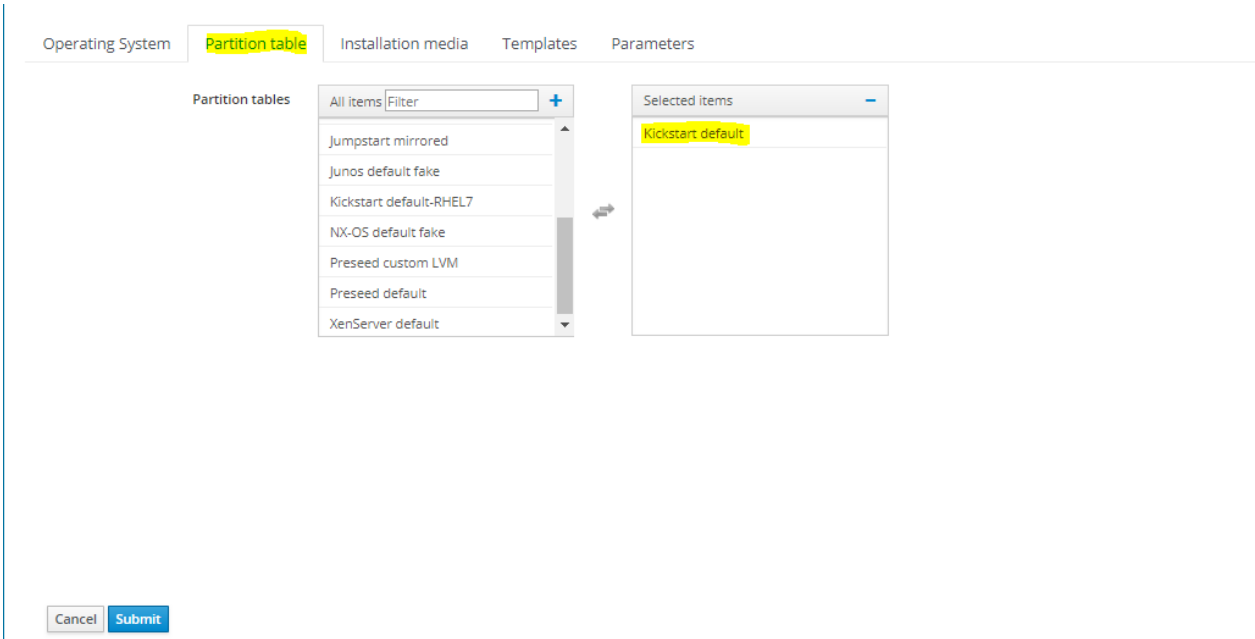
- Click **Hosts** → **Operating Systems**.
- Click **Create Operating system**. On the **Operating System** tab, specify the following settings:



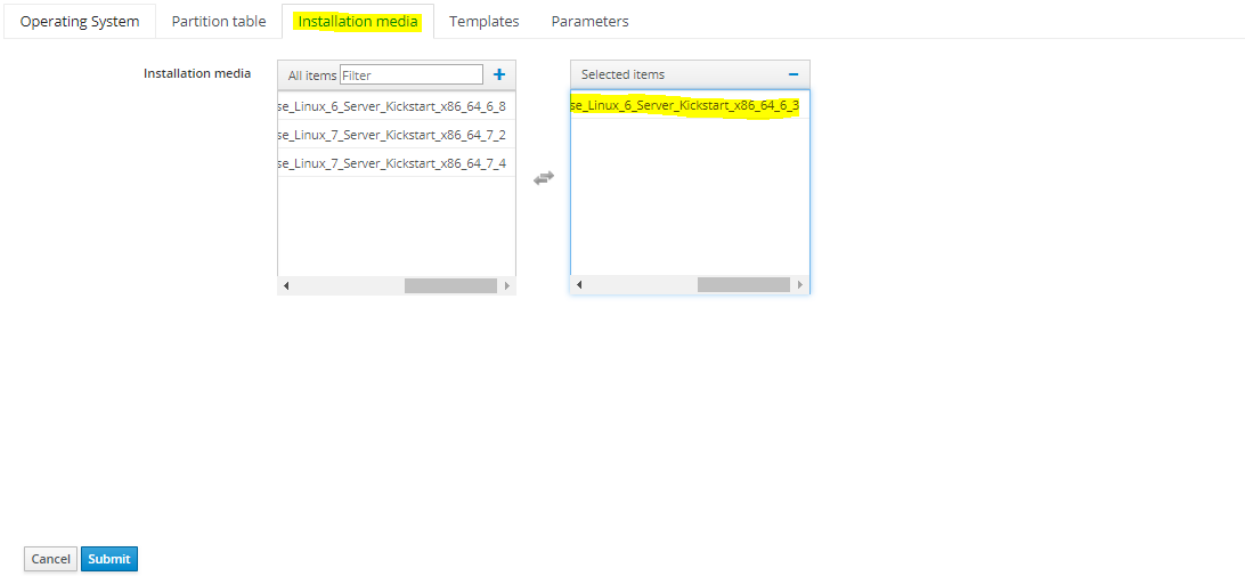
a. Type the **Name** of the Operating System and its **Major Version**. These settings are required.
b. Optionally, define the **Minor Version**, select the **OS Family**, and add a **Description** of the operating system.
c. Select a **Root password hash** (MD5, SHA256, of SHA512).
d. Select the **Architectures** from the list of available Architectures.



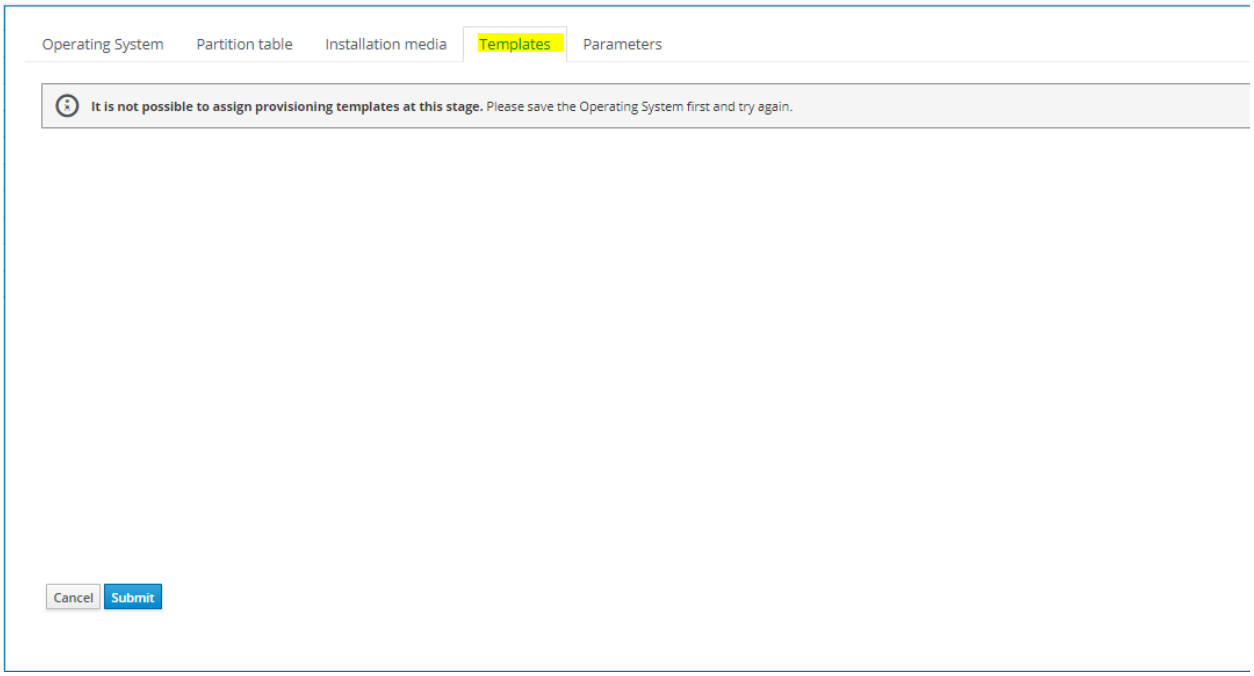
- On the **Partition tables** tab, select the applicable file system layouts from the list



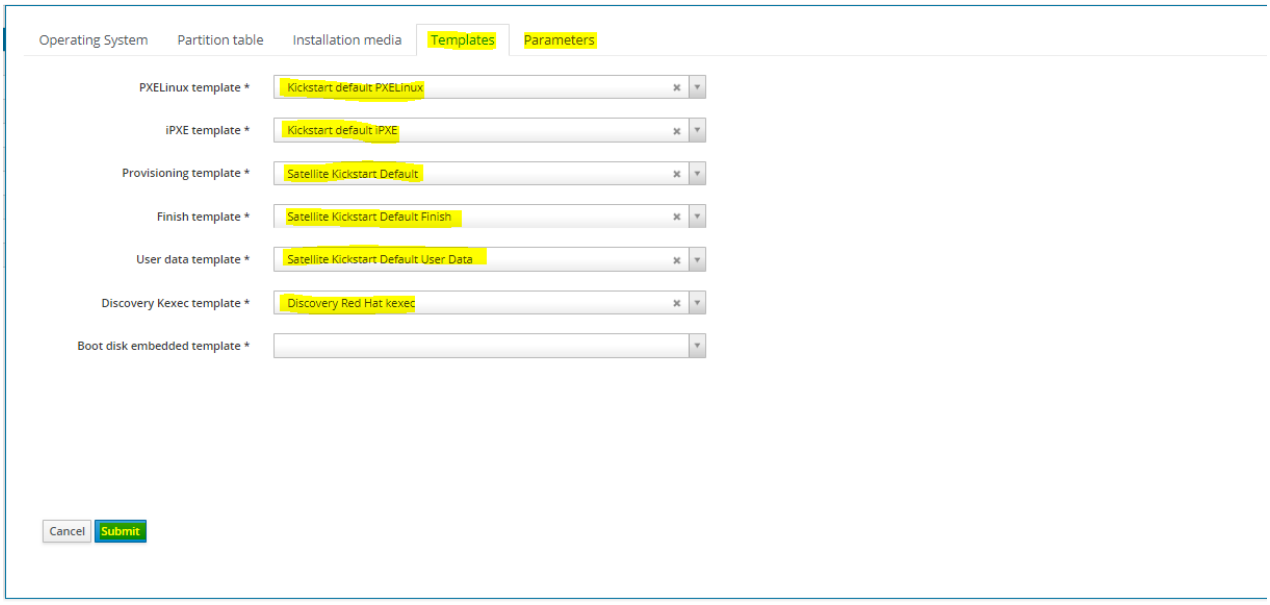
- 4. On the **Installation Media** tab, select the applicable installation media from the list.



- On the **Templates** tab, you can assign provisioning templates when editing an existing operating system.



- 6. On the **Parameters** tab, you can add parameters for the operating system.
- 7. Click **Submit**.



1.9. Configuring Host Collections

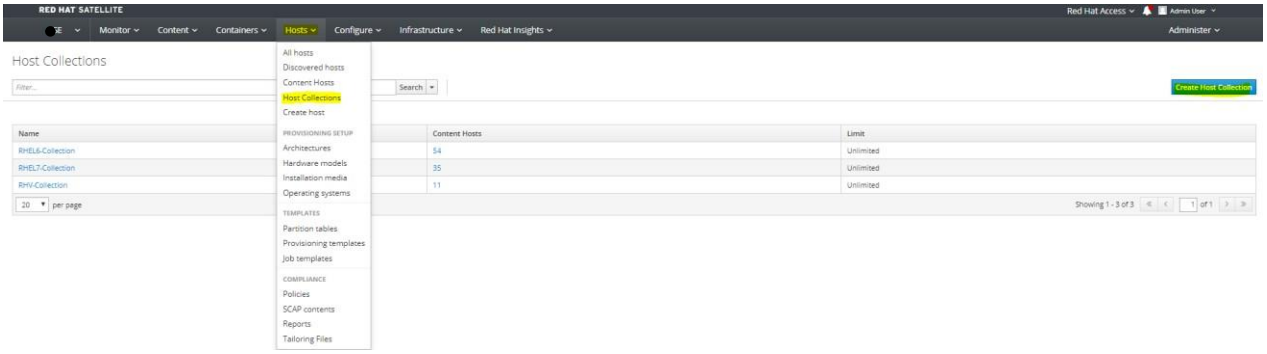
The Host Collections application tab is a system management tool that allows the administrator to: Add hosts to a collection. Apply a mass installation of packages, errata, or package groups to all host members of a host collection. Update specific packages, errata, or specific package groups to all host members.

1.9.1. Creating a Host Collection

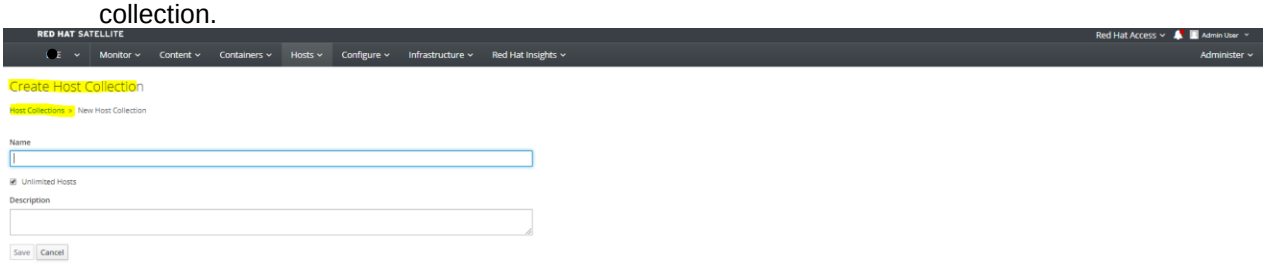
The following procedure shows how to create host collections.

To Create a Host Collection:

- Click **Hosts** → **Host Collections**.



- Click **Create Host Collection**.
- Add the Name and Description of the host collection.
- Deselect **Unlimited Content Hosts** to specify the maximum number of hosts that will be allowed to the group. Otherwise, leave it checked to allow unlimited hosts to join the host collection.



- Click **Save**.

1.9.2. Adding Hosts to a Host Collection

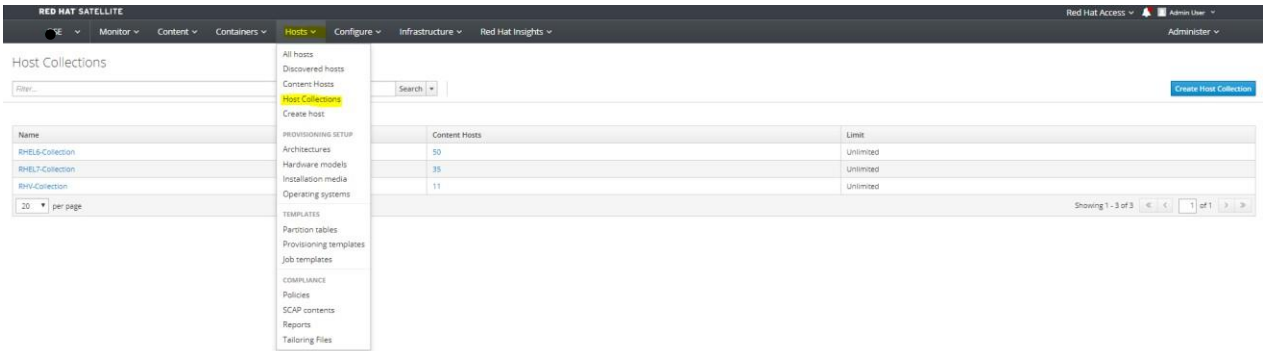
The following procedure shows how to add hosts to host collections.

Prerequisites

A host must be registered to Red Hat Satellite in order to add it to a Host Collection

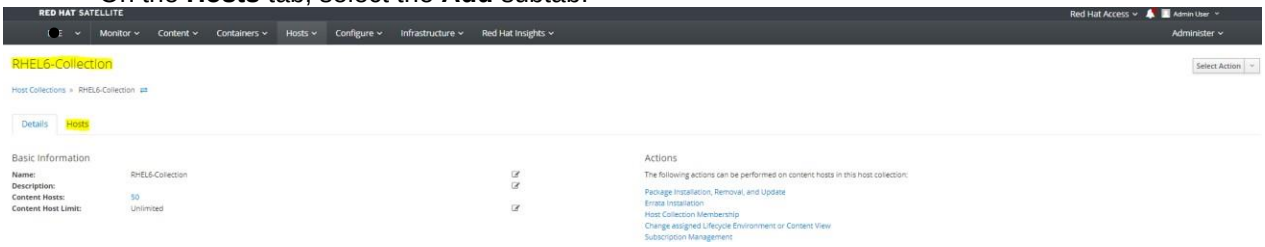
To Add Hosts to a Host Collection:

- Click **Hosts** → **Host Collections**.

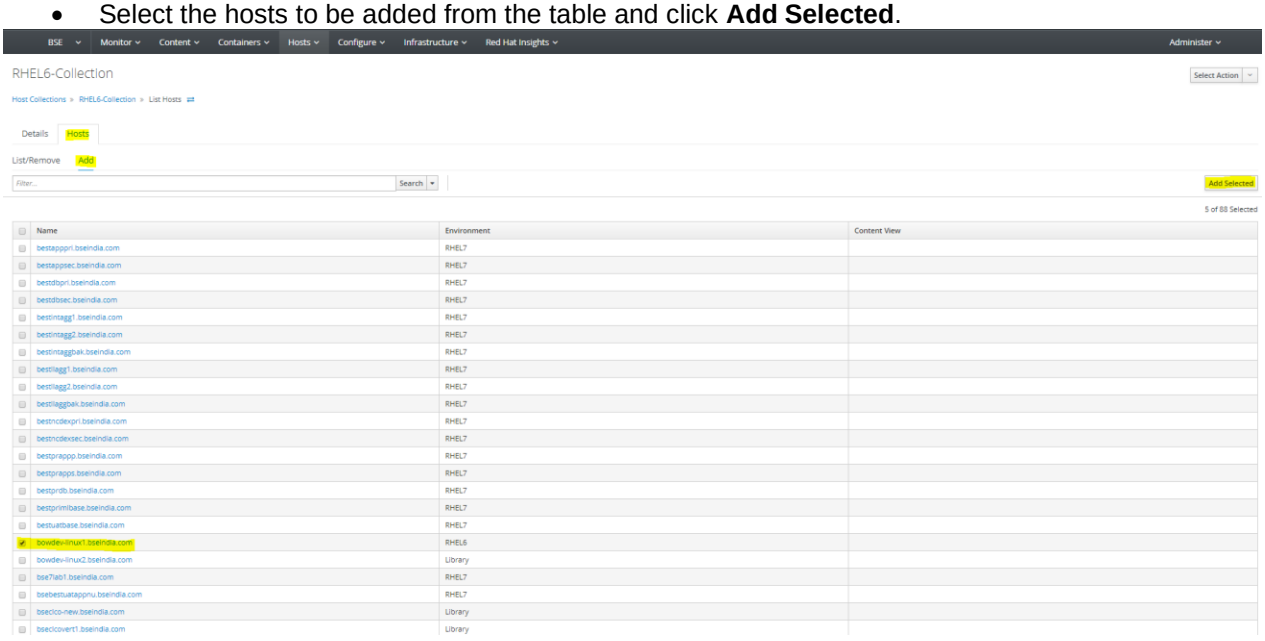


- Click the host collection where the host should be added.

- On the **Hosts** tab, select the **Add** subtab.



- Select the hosts to be added from the table and click **Add Selected**.



1.9.3. Adding Content to Host Collections

These steps show how to add content to host collections in Red Hat Satellite.

1.9.4. Adding Packages to a Host Collection

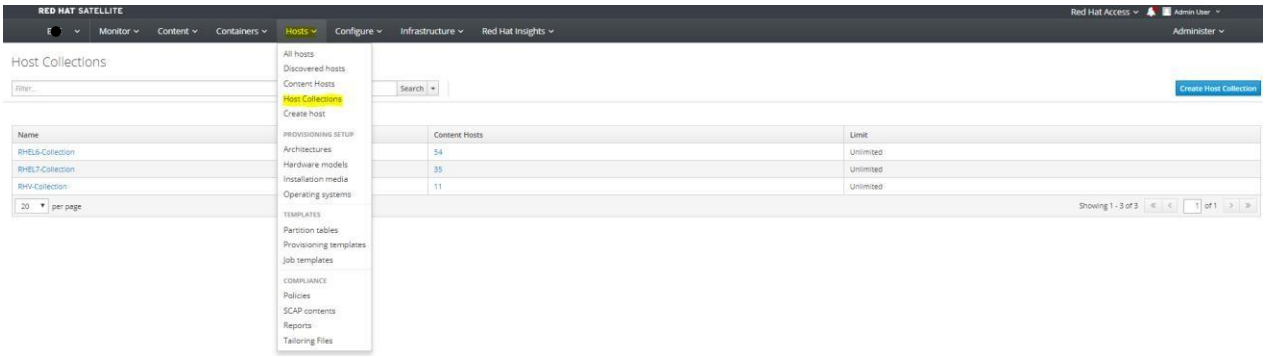
The following procedure shows how to add packages to host collections.

Prerequisites

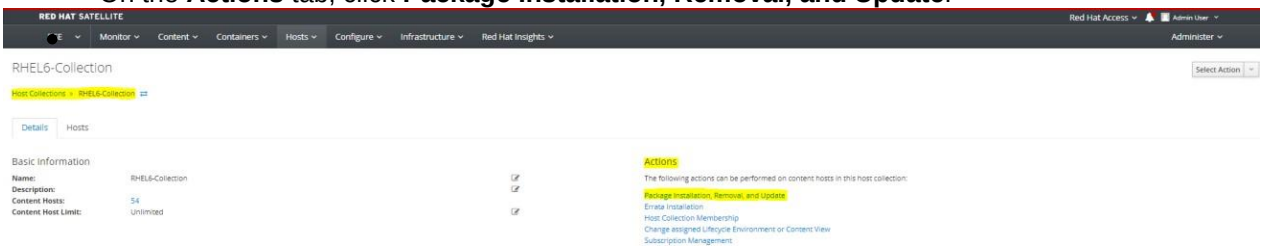
The content to be added should be available in one of the existing repositories or added prior to this procedure. Content should be promoted to the environment where the hosts are assigned.

To Add Packages to Host Collections:

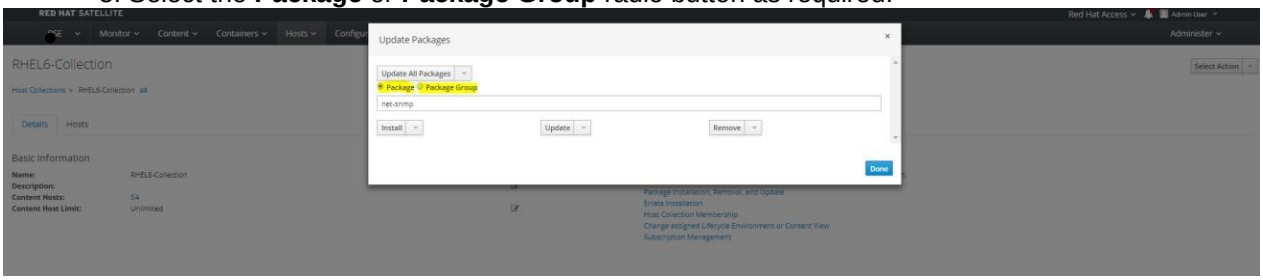
- Click **Hosts** → **Host Collections**.



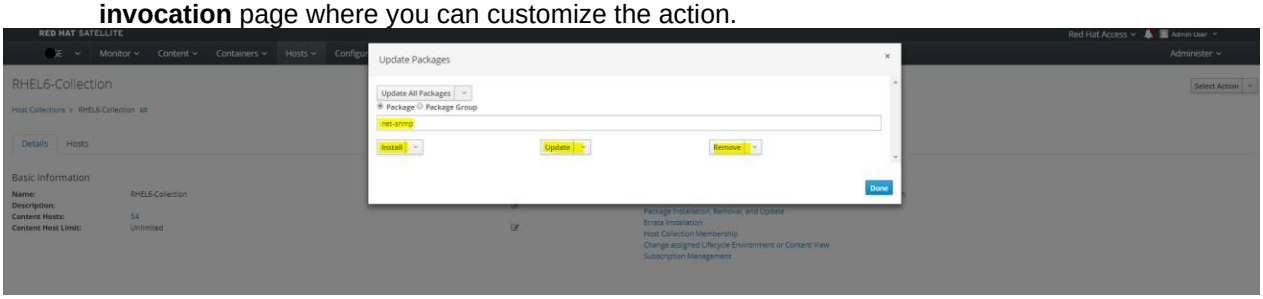
- Click the host collection where the package should be added.
- On the **Actions** tab, click **Package Installation, Removal, and Update**.



- 4. To update all packages, click the **Update All Packages** button to use the default method. Alternatively, select the drop-down icon to the right of the button to select a method to use. Selecting the **via remote execution – customize first** menu entry will take you to the **Job invocation** page where you can customize the action.
- 5. Select the **Package** or **Package Group** radio button as required.



- 6. In the field provided, specify the package or package group name. Then click: **Install** — to install a new package using the default method. Alternatively, select the drop-down icon to the right of the button and select a method to use. Selecting the **via remote execution - customize first** menu entry will take you to the **Job invocation** page where you can customize the action. **Update** — to update an existing package in the host collection using the default method. Alternatively, select the drop-down icon to the right of the button and select a method to use. Selecting the **via remote execution - customize first** menu entry will take you to the **Job invocation** page where you can customize the action.

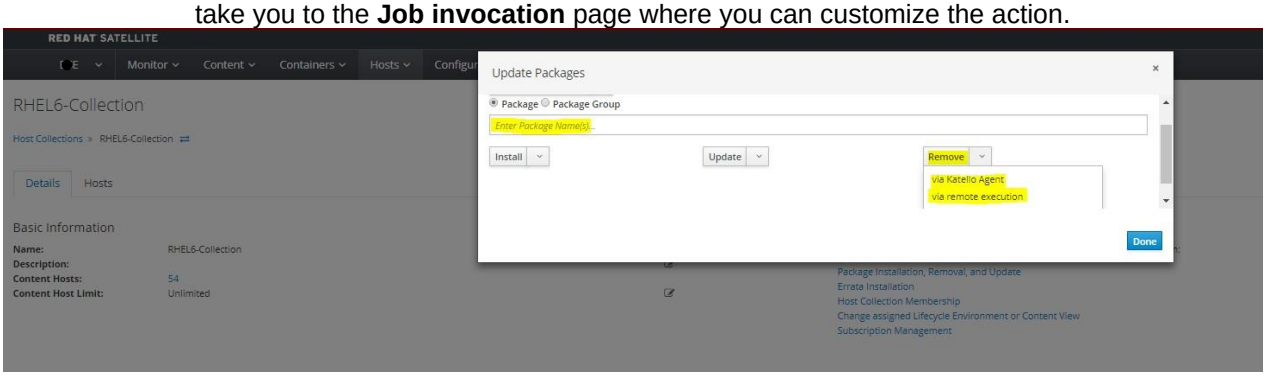


1.9.5. Removing Content from a Host Collection

The following procedure shows how to remove packages from host collections.

To Remove Content from a Host Collection:

- Click **Hosts** → **Host Collections**.
- Click the host collection where the package should be removed.
- On the **Collection Actions** tab, click **Package Installation, Removal, and Update**.
- Select the **Package** or **Package Group** radio button as required.
- In the field provided, specify the package or package group name.
- Click the **Remove** button to remove the package or package group using the default method. Alternatively, select the drop-down icon to the right of the button and select a method to use. Selecting the **via remote execution - customize first** menu entry will take you to the **Job invocation** page where you can customize the action.

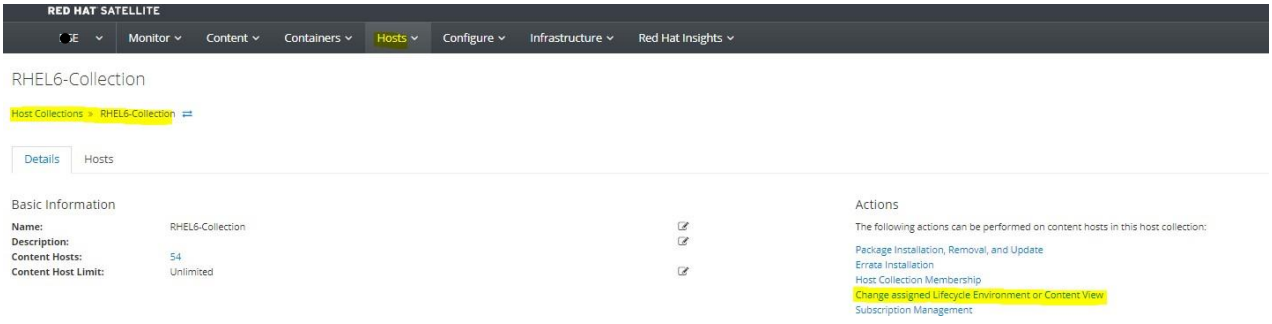


1.9.6. Changing the Life Cycle Environment or Content View of a Host Collection

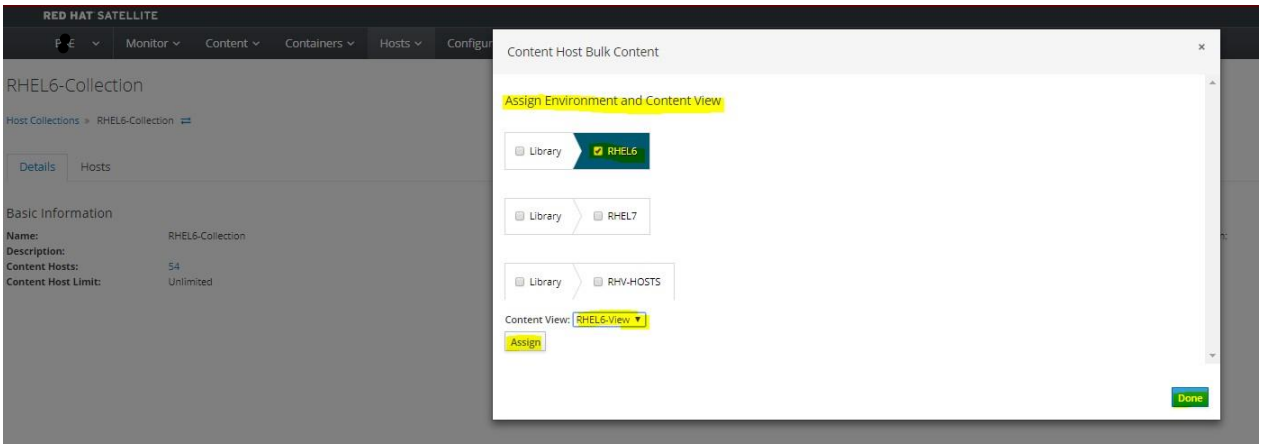
The following procedure shows how to change the assigned life cycle environment or content view of host collections.

To Change the Life Cycle Environment or Content View of a Host Collection:

- Click **Hosts** → **Host Collection**.
- Selection the host collection where the life cycle environment or content view should be changed.
- On the **Collection Actions** tab, click **Change assigned Life Cycle Environment or Content View**.



- Select the life cycle environment to be assigned to the host collection.



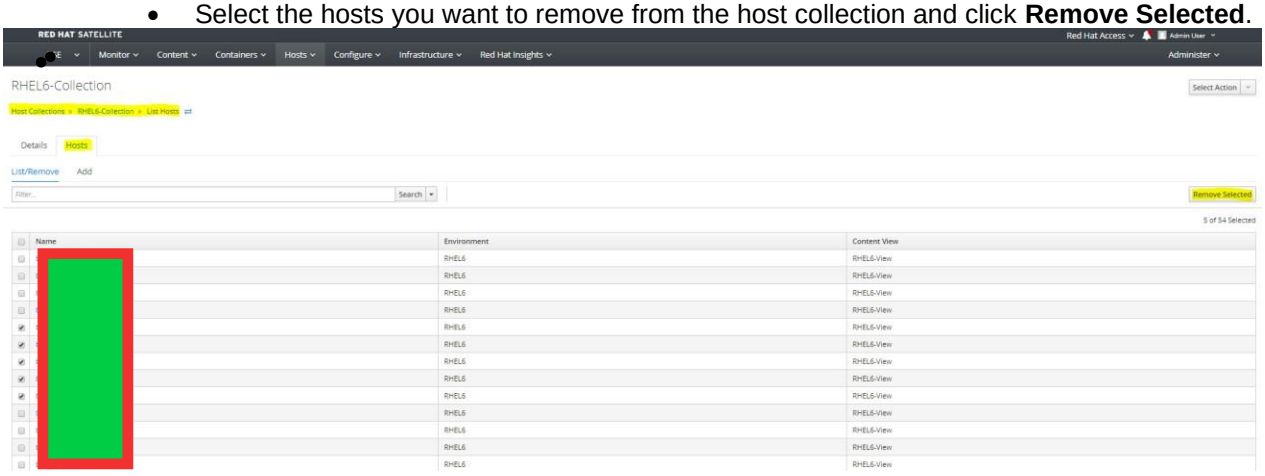
- Select the required content view from the drop-down list.
- Click **Assign**.

1.9.7. Removing a Host from a Host Collection

The following procedure shows how to remove hosts from host collections.

To Remove Hosts from a Host Collection:

- Click **Hosts** → **Host Collections**.
- Choose the desired host collection.
- On the **Hosts** tab, select the **List/Remove** subtab.
- Select the hosts you want to remove from the host collection and click **Remove Selected**.

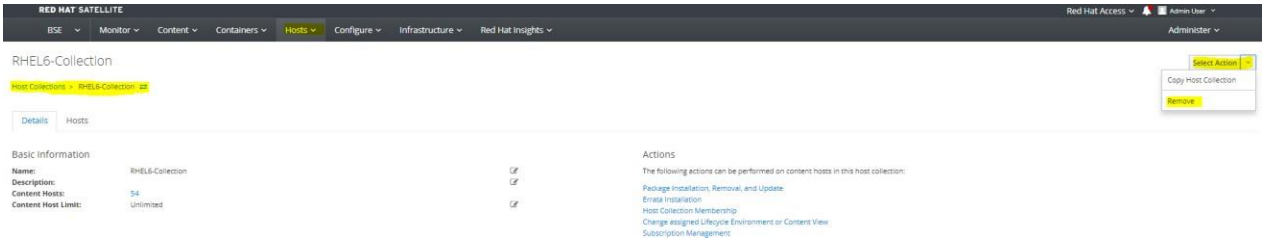


1.9.8. Removing a Host Collection

The following procedure shows how to remove a host collection.

To Remove a Host Collection:

- Click **Hosts** → **Host Collections**.
- Choose the host collection to be removed.
- Click **Select Action.**
- Click **Remove**.



2. Registering hosts in satellite6

- Remove host profile from satellite 5
- Delete rhn id from host
  $sudo rm –rf /etc/sysconfig/rhn/systemid
- Add host entry in /etc/hosts
  192.168.252.51 example.satellite.com [redacted]satellite6
- Check subscription-manager package version (should be latest)
  $cd /usr/bin/
  $sudo wget http://example.satellite.com/pub/bootstrap.py

```
[hranjith@[redacted] bin]$ cd /usr/bin/
[hranjith@[redacted] bin]$ sudo wget http://192.168.252.51/pub/bootstrap.py
--2018-05-08 11:23:37--  http://192.168.252.51/pub/bootstrap.py
Connecting to 192.168.252.51:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 41377 (40K)
Saving to: “bootstrap.py.1”

100%[==========================================================================================================>] 41,377      --.-K/s   in 0s

2018-05-08 11:23:37 (126 MB/s) - “bootstrap.py.1” saved [41377/41377]

[hranjith@[redacted] bin]$
```

$sudo chmod +x bootstrap.py

```
[hranjith@[redacted] bin]$ sudo chmod +x bootstrap.py
[hranjith@[redacted] bin]$ ls bootstrap.py
bootstrap.py
[hranjith@[redacted] bin]$
```

For RHEL 6 Servers
$ sudo ./bootstrap.py -l admin -s example.satellite.com -L MUMBAI -o [redacted] -a RHEL6-KEY -g RHEL6-Group -- force --skip-foreman

For RHEL 7 Servers
$ sudo ./bootstrap.py -l admin -s example.satellite.com -L MUMBAI -o [redacted] -a RHEL7-KEY -g RHEL7-Group -- force --skip-foreman

Remove **katello-agent** package.

2.1. Registering RHV Hypervisors

$sudo rm -rf

/etc/sysconfig/rhn/systemid Add host

entry in /etc/hosts

192.168.252.51 example.satellite.com [illegible]satellite6

$sudo rpm -Uvh http://example.satellite.com/pub/katello-ca-consumer-latest.noarch.rpm

$sudo subscription-manager unsubscribe –all

$sudo subscription-manager clean

$sudo subscription-manager refresh

$sudo subscription-manager register --org="[illegible]" --activationkey="RHV-KEY"

$sudo yum repolist

2.2. Registering RHV VM’S

Add host entry in /etc/hosts

192.168.252.51 example.satellite.com [illegible]satellite6

$sudo rm -rf /etc/sysconfig/rhn/systemid

$sudo rpm -Uvh http://example.satellite.com/pub/katello-ca-consumer-latest.noarch.rpm

$sudo subscription-manager unsubscribe –all

$sudo subscription-manager clean

$sudo subscription-manager refresh

$sudo subscription-manager register --org="[illegible]" --

auto-attach Username: admin

Password : *****

Environment:

Library

2.3. Registering hosts without katello agent

Add host entry in /etc/hosts

192.168.252.51 example.satellite.com satellite6

Check subscription-manager package version (should

be latest)

$sudo rm -rf

/etc/sysconfig/rhn/systemid delete

system profile from satellite 5.

$sudo rpm -Uvh http://example.satellite.com/pub/katello-ca-consumer

latest.noarch.rpm Activation Key for RHEL6 servers

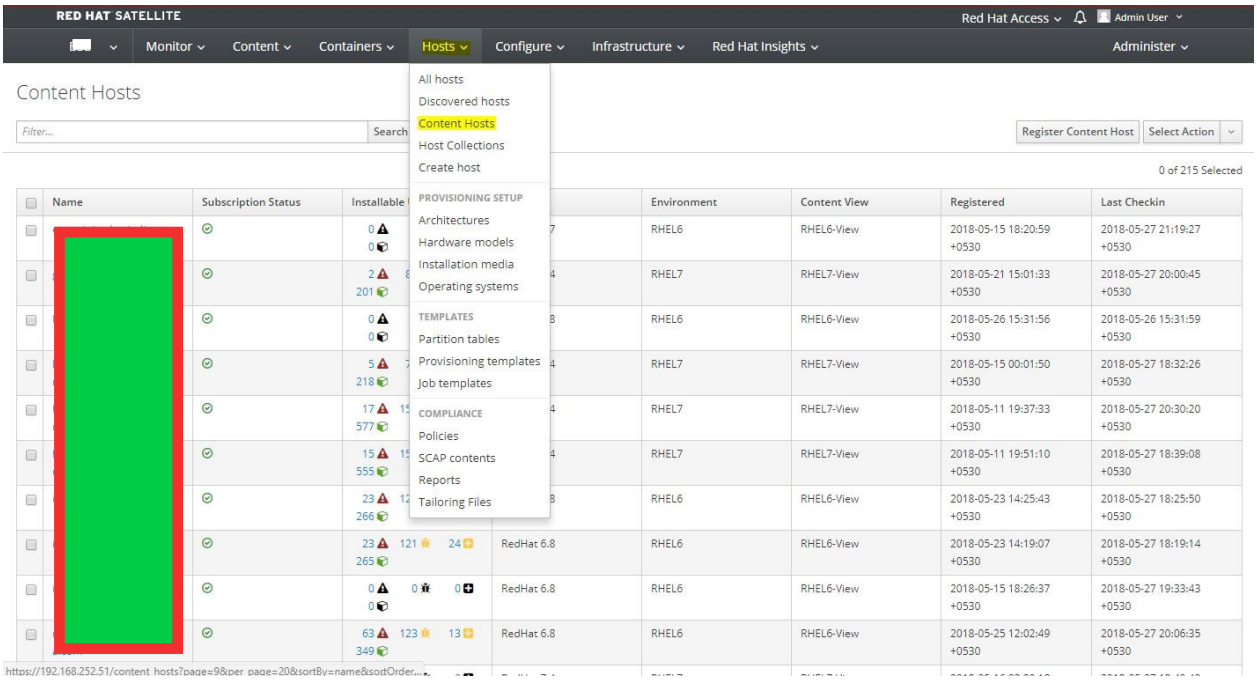
$sudo subscription-manager register --org="DOE" --

activationkey="RHEL6-KEY" Activation Key for RHEL7 servers

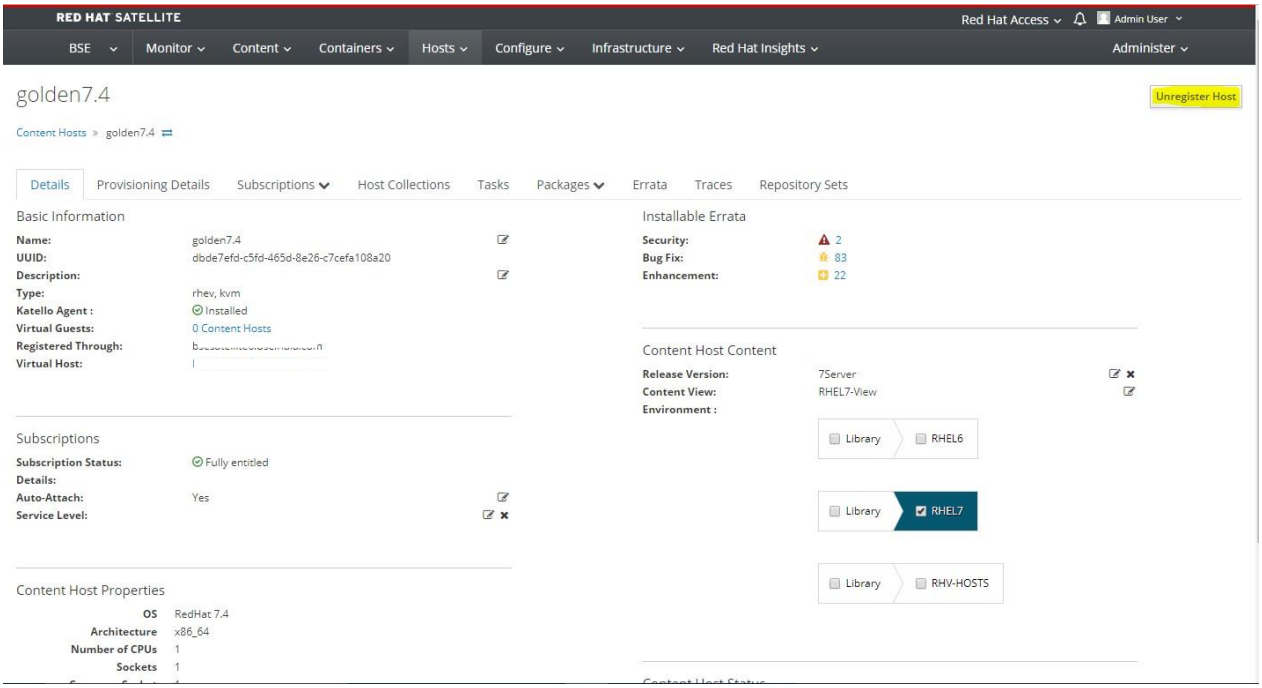
$sudo subscription-manager register --org="DOE" --activationkey="RHEL7-KEY"

2.4. Unregistering hosts

- Goto **Hosts→ Content Hosts** → Select the host you want to unregister



- Click **Unregister Host**

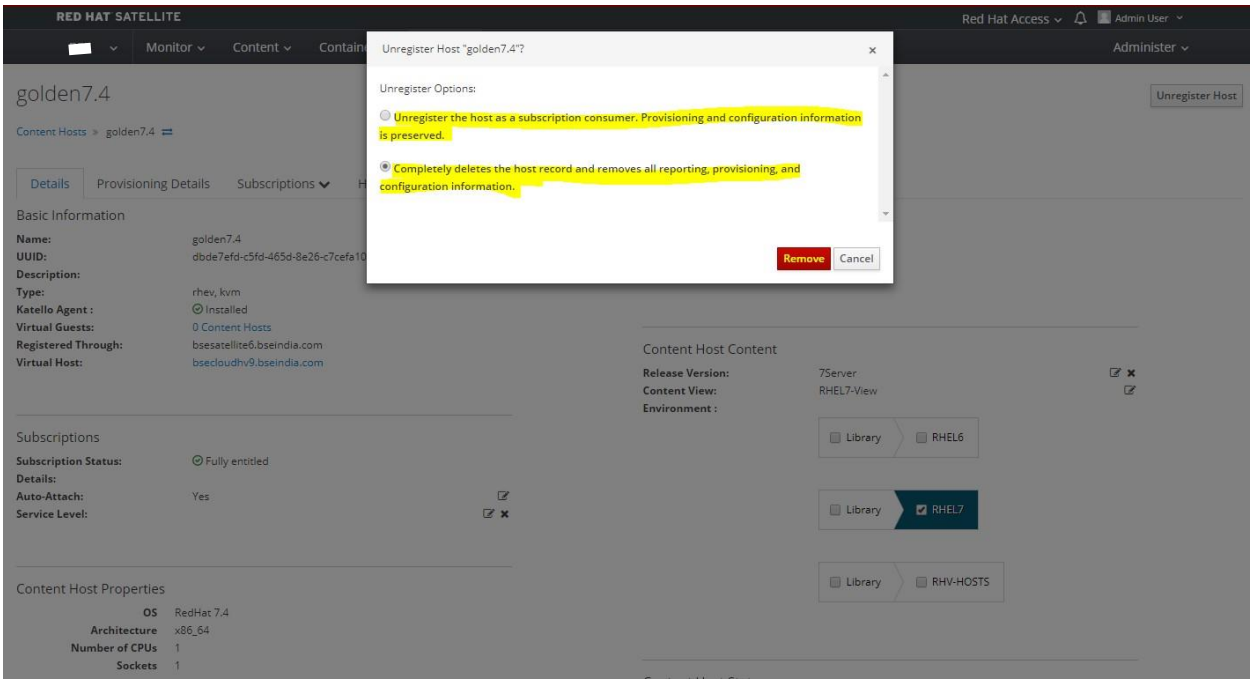


Select below option to just remove the subscription

- **Unregister the host as a subscription consumer, provisioning and configuration information is preserved.**

Select below option to completely remove the host record

- **Completely deletes the host record and removes all reporting, provisioning, and Configuration information.**



2.5. Virt-who Configuration in RHV Hypervisors

- As RHVH is a read-write filesystem, RPM packages can be directly installed on it. Install consumer RPM (http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm) and register it with Red Hat Satellite 6 server.
- Attach **Red Hat Enterprise Linux Server (Unlimited Guests)** or equivalent subscription which provides 1 Physical and Unlimited Guest subscription to the hypervisor profile.
- By default, virt-who is installed on RHVH 4.1 hypervisor, so directly follow the instructions for configuring listed below

  # rpm -qa |grep virt-who
  virt-who-0.12-
  10.el6.noarch

- Edit the virt-who configuration file and enable only the below parameters.

#vi /etc/sysconfig/virt-who

VIRTWHO_DEBUG=0
VIRTWHO_BACKGROUND=1
VIRTWHO_INTERVAL=3600
VIRTWHO_SATELLITE6=1

```
[root@[redacted] ~]# cat /etc/sysconfig/virt-who | grep -v ^# | grep .
VIRTWHO_DEBUG=0
VIRTWHO_BACKGROUND=1
VIRTWHO_INTERVAL=3600
VIRTWHO_SATELLITE6=1
[root@[redacted] ~]#
```

- Create a new configuration file inside /etc/virt-who.d/

$sudo vi /etc/virt-who.d/config.conf

```
[root@[redacted] ~]# cat /etc/virt-who.d/config.conf
[[redacted]]
type=vdsm
hypervisor_id=[redacted]
[root@[redacted] ~]#
```

[[redacted]]
type=vdsm
hypervisor_id=[redacted]

- After completing the above steps, restart the virt-who service

$sudo systemctl restart virt-who
$sudo systemctl enable virt-who
$sudo tailf /var/log/rhsm/rhsm.log

- Try to register guests created on the hypervisor and confirm whether they are using subscription attached to the hypervisor or not.

3. Provisioning

We explore four main ways to provision bare metal instances with Red Hat Satellite 6. These include:

- **Unattended Provisioning** - You identify a host using a MAC address and the Satellite Server provisions it using a PXE boot process.
- **Unattended Provisioning with Discovery** - New hosts use PXE boot to load the Satellite Discovery service. This service identifies hardware information about the host and lists it as an available host to provision.
- **PXE-less provisioning** - The ability to provision new hosts using a boot disk or PXE-less discovery image that the Satellite Server generates.
- **PXE-less Provisioning with Discovery** - New hosts use an ISO boot disk that loads the Satellite Discovery service. This service identifies hardware information about the host and lists it as an available host to provision.

3.1. Configuring RED HAT Satellite's Discovery Service

Red Hat Satellite provides a Discovery feature. This provides a method to automatically detect blank hosts on a network. These hosts boot a special image that performs hardware detection and relays this information back to the Satellite Server. This provides a method to create a pool of ready-to provision hosts on the Satellite Server and without needing to enter the MAC address of each host.

3.2. Discovering Bare-metal Hosts on Satellite

Red Hat Satellite 6.2 ships with the Discovery plug-in already installed. The Discovery plug-in enables automatic bare-metal discovery of unknown hosts on the provisioning network. These new hosts are registered to the Satellite Server and the Puppet agent on the client uploads system facts collected by Factor, such as serial ID, network interface, memory, and disk information. After registration, the hosts are displayed on the Discovered Hosts page in the Satellite web UI. You can then initiate provisioning either manually (using the web UI, CLI, or API) or automatically, using predefined discovery rules.

The Discovery plug-in communicates through the Satellite Capsule Server, which has direct access both to the provisioning network and the Satellite Server instance. It is possible to discover hosts directly from the Satellite Server, but Red Hat recommends the following scheme be used:
Satellite Server (Satellite Server Discovery plug-in) <--> Satellite Capsule (Satellite Capsule Discovery plug-in) <--> Discovered Host (Satellite Discovery image)

The Satellite Discovery plug-in consists of three different components:

The Satellite Server Discovery plug-in

This runs on the Satellite Server and provides API and UI functionality for working with discovered hosts. The tfm- rubygem-foreman_discovery package contains this plug-in.

The Satellite Capsule Server Discovery plug-in

This is a communication proxy between discovered hosts on a provisioning network and the Satellite Server. The **rubygem-smart_proxy_discovery** package contains this plug-in.

The Satellite Discovery image

This is the minimal operating system based on Red Hat Enterprise Linux that is PXEbooted on hosts to acquire initial hardware information and to check in to the Satellite Server. Discovered hosts keep running the Satellite Discovery image until they are rebooted into Anaconda, which then initiates the provisioning process. The foreman discovery- image package contains this image. It must be installed on the Satellite Capsule Server that provides TFTP services.

Configuring the Satellite Discovery Plug-in

The following sections describe how to configure the Satellite Discovery plug-in and how to prepare the PXE-boot template on the Satellite Server.

Deploying the Satellite Discovery Image

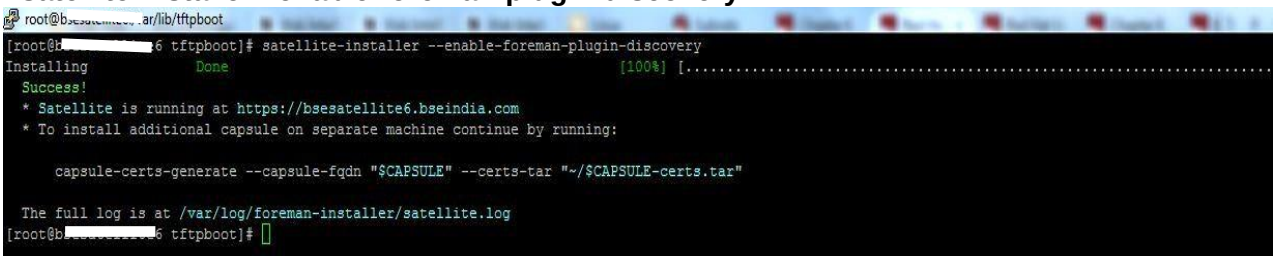
Install the package containing the Satellite Discovery image on the Satellite Capsule Server that provides TFTP services (not on the Satellite Server itself):

 # yum install foreman-discovery-image rubygem-smart_proxy_discovery
This package contains the Linux kernel and initial RAM disk image as a bootable ISO file which is used for PXE- booting discovered hosts. You can run the following command to investigate the contents of the package. This produces output similar to the following:

$ rpm -ql foreman-discovery-image /usr/share/foreman-discovery-image
/usr/share/foreman-discovery- image/fdi-image-rhel_7-2.1.0-20150212.1.iso

When you install this package, it extracts the kernel and image from the ISO file into the TFTP directory and creates symbolic links to the latest versions of the image and kernel. Use the symbolic links in the PXE-boot provisioning template to make sure that you do not need to change the version in the template every time the foreman- discovery-image package is upgraded. For example:
$ find /var/lib/tftpboot/boot
/var/lib/tftpboot/boot
/var/lib/tftpboot/boot/fdi-image-rhel_7-2.1.0-20150212.1-img
/var/lib/tftpboot/boot/fdi-image-rhel_7-2.1.0-20150212.1-vmlinuz
/var/lib/tftpboot/boot/fdi-image-rhel_7-img /var/lib/tftpboot/boot/fdi-image-rhel_7-vmlinuz

#satellite-installer –enable-foreman-plugin-discovery



This installs and enables the Discovery service plugin on the Satellite Server. After installation completes, install the following packages:

#Katello-service restart

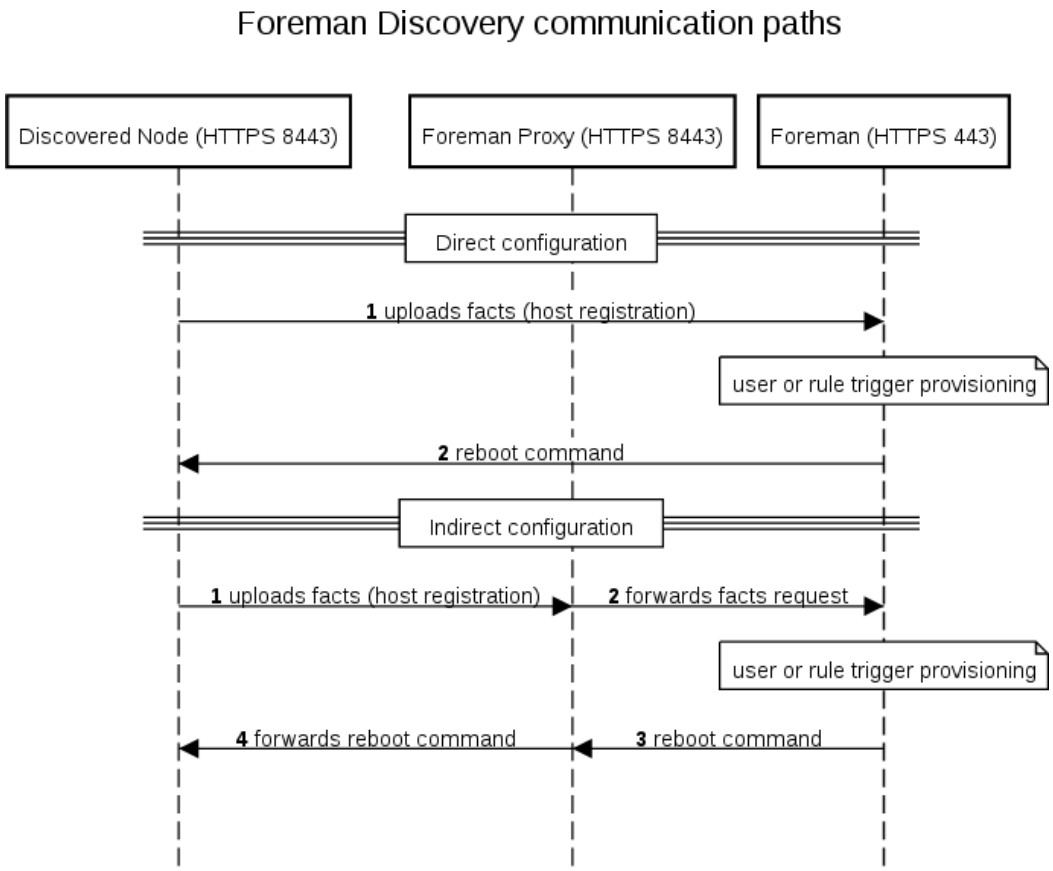
The **PXELinux global default template** in the **Hosts > Provisioning templates** section includes an entry for the Discovery service.

LABEL discovery
MENU LABEL (discovery)
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0
nomodeset proxy.url=https://SATELLITE_SERVER_URL:9090
proxy.type=proxy IPAPPEND 2

The **KERNEL** and **APPEND** options boot the Discovery image and ramdisk. Also note the **APPEND** option contains a **proxy.url** parameter, which specifies the URL of the Satellite Server to use for provisioning. Edit the **SATELLITE_SERVER_URL** to the name of the provisioning capsule that you want. In this scenario, it is the Satellite Server's integrated Capsule:

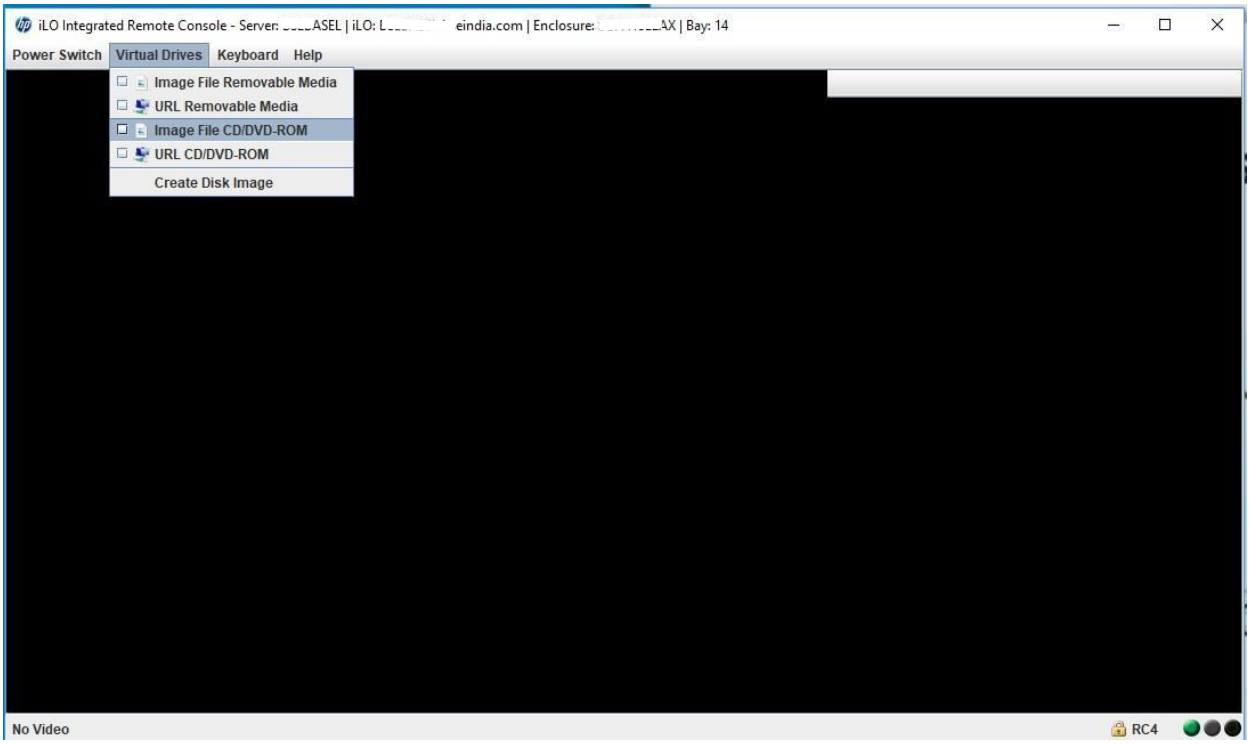
You can change the Discovery service to be the default service that boots for blank hosts. Edit the **ONTIMEOUT** value in the **PXELinux global default** to the following **ONTIMEOUT discovery.**

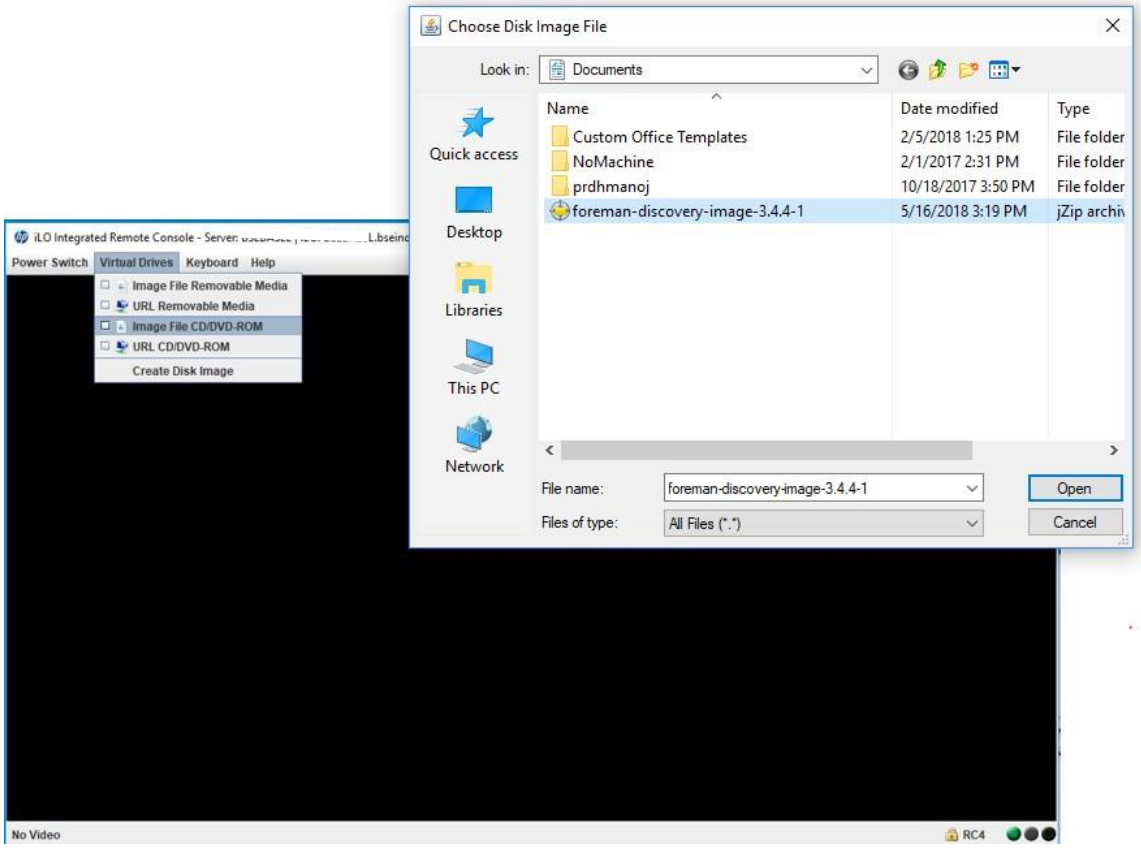
You need to push the changes from the **PXELinux global default template** to the Satellite Server's default PXE template. Navigate to **Hosts** > **Provisioning templates** and **click Build PXE Default**. This refreshes the default PXE template on the Satellite Server.



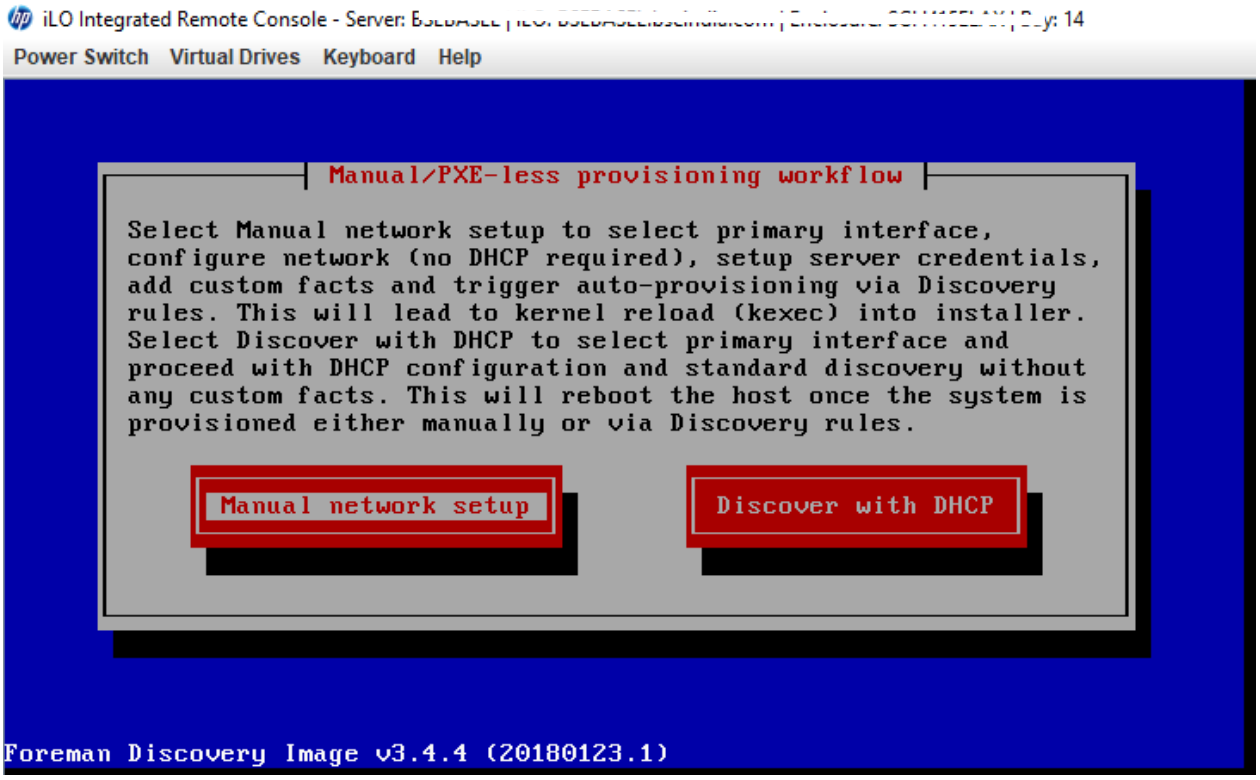
3.3. Provisioning Hosts

- Login with ILO & mount foreman discovery image iso.

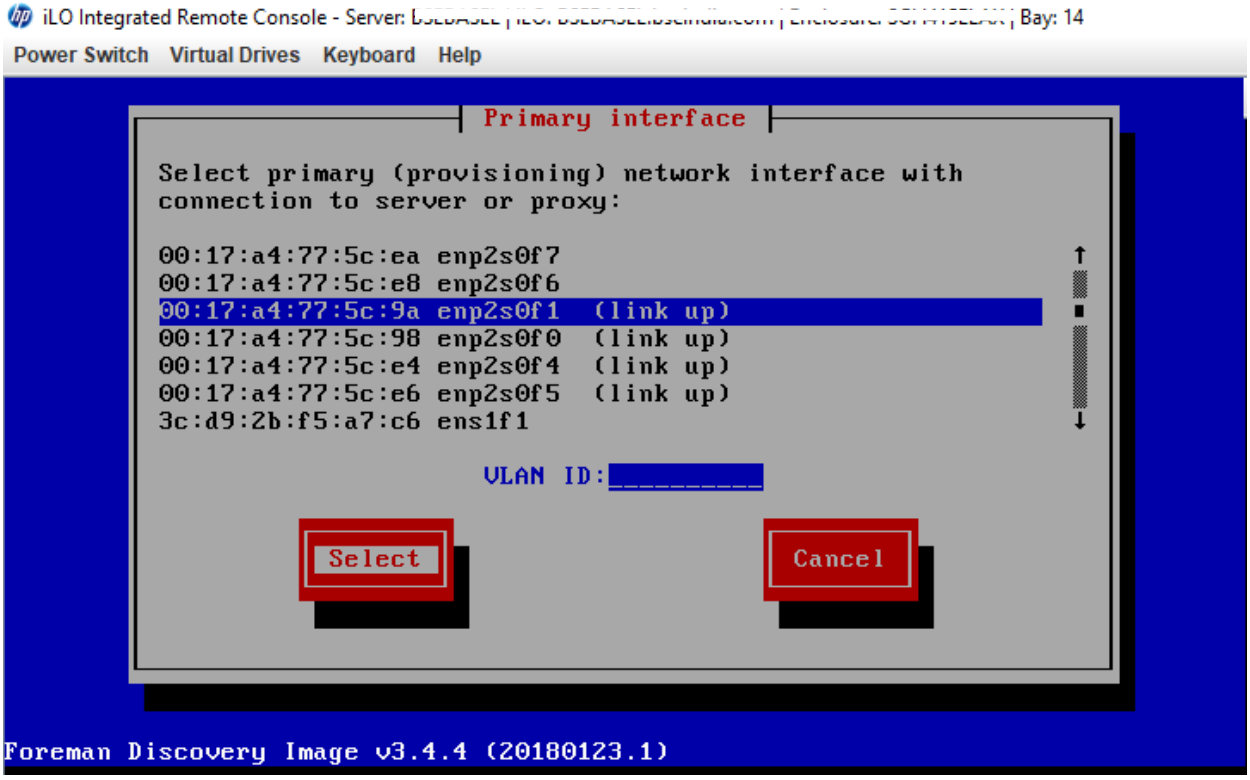

iLO Integrated Remote Console - Server:
eindia.com | Enclosure:
Bay: 14
Power Switch
Virtual Drives
Keyboard
Help
Image File Removable Media
URL Removable Media
Image File CD/DVD-ROM
URL CD/DVD-ROM
Create Disk Image
No Video
RC4



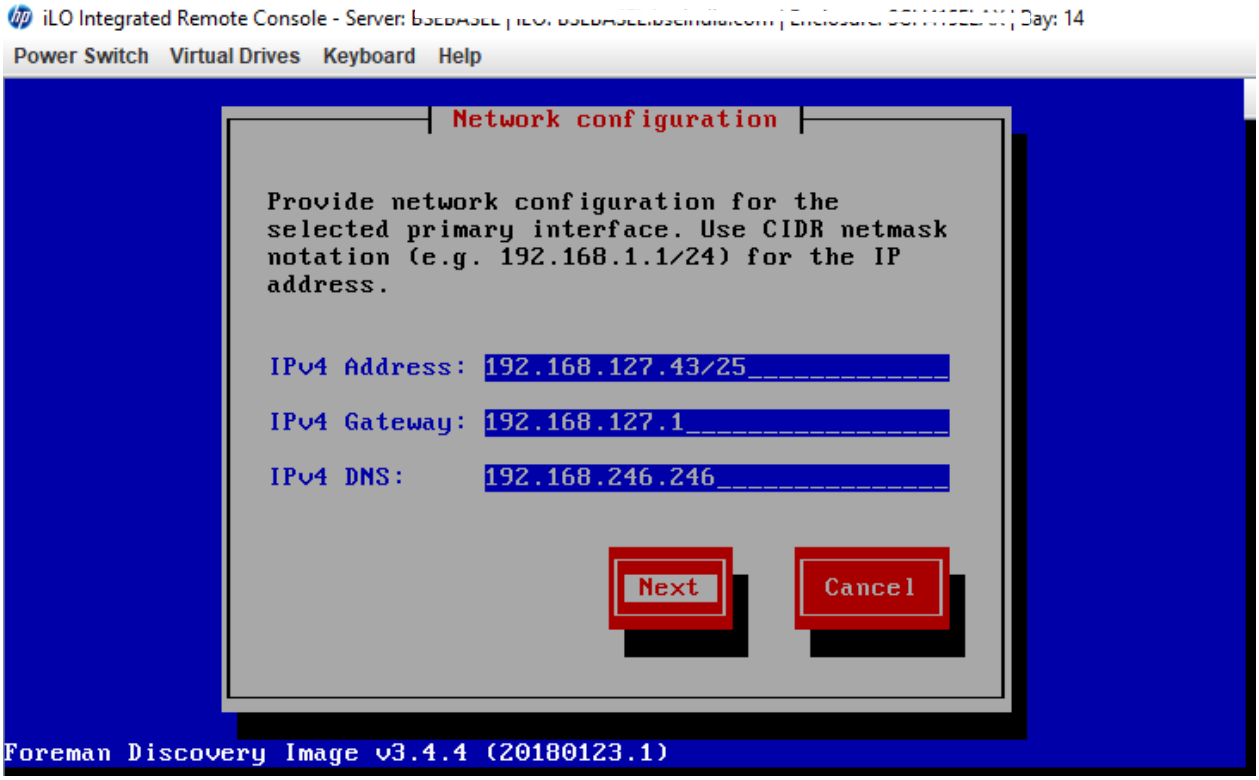
- Boot with foreman discovery ISO & select Manual network setup



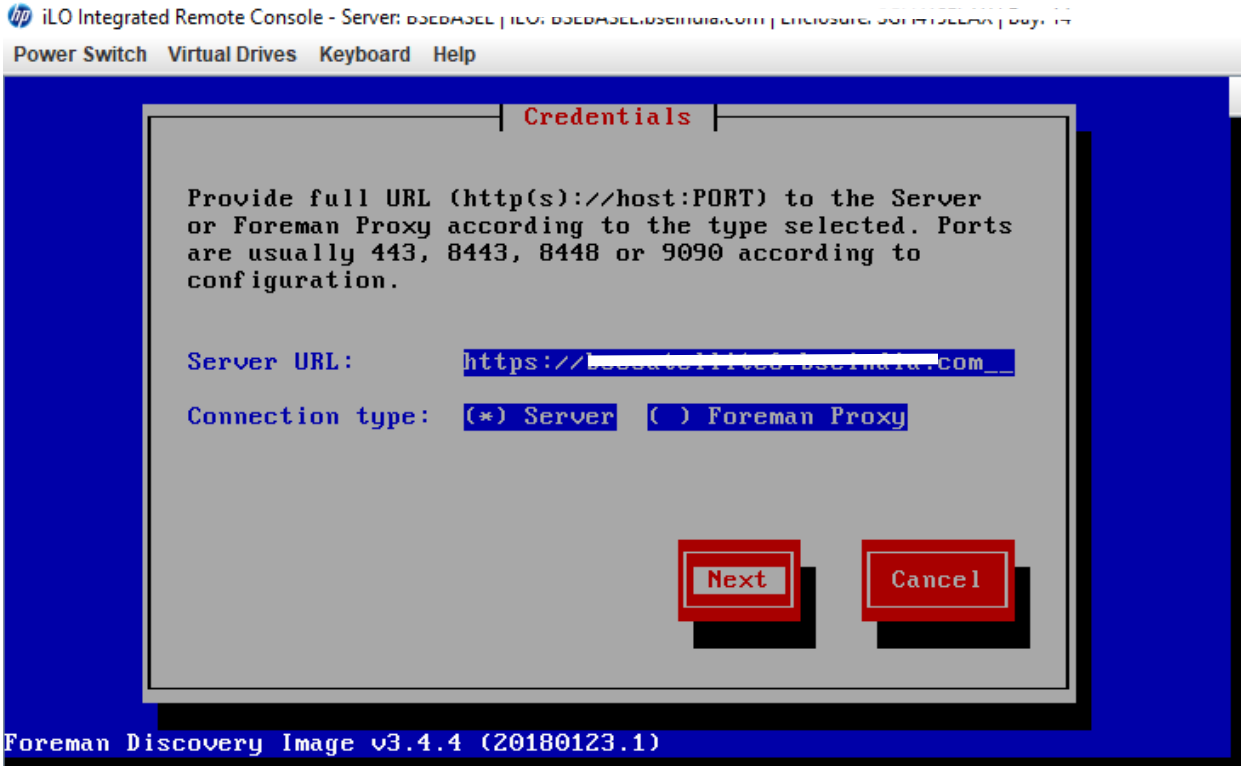
- Select the primary interface



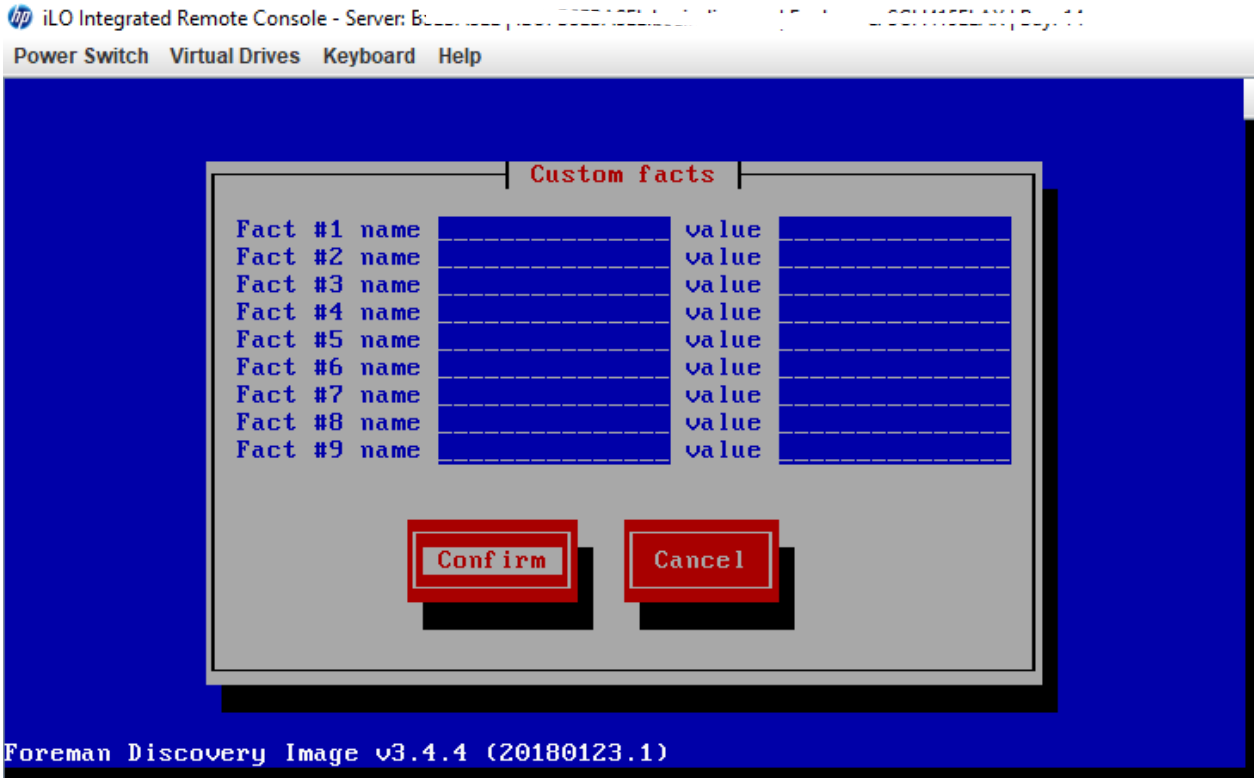
- Assign IP address, Gateway & DNS details



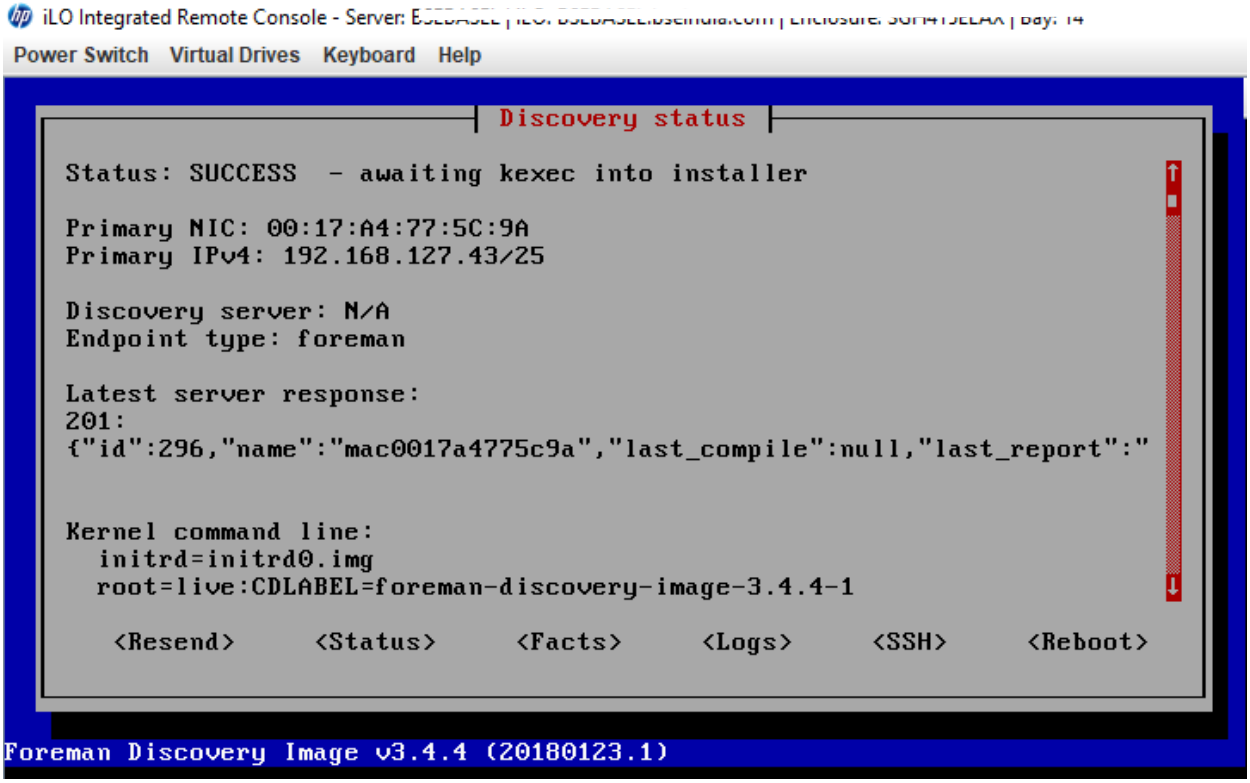
- Enter satellite URL & select connection type as server.



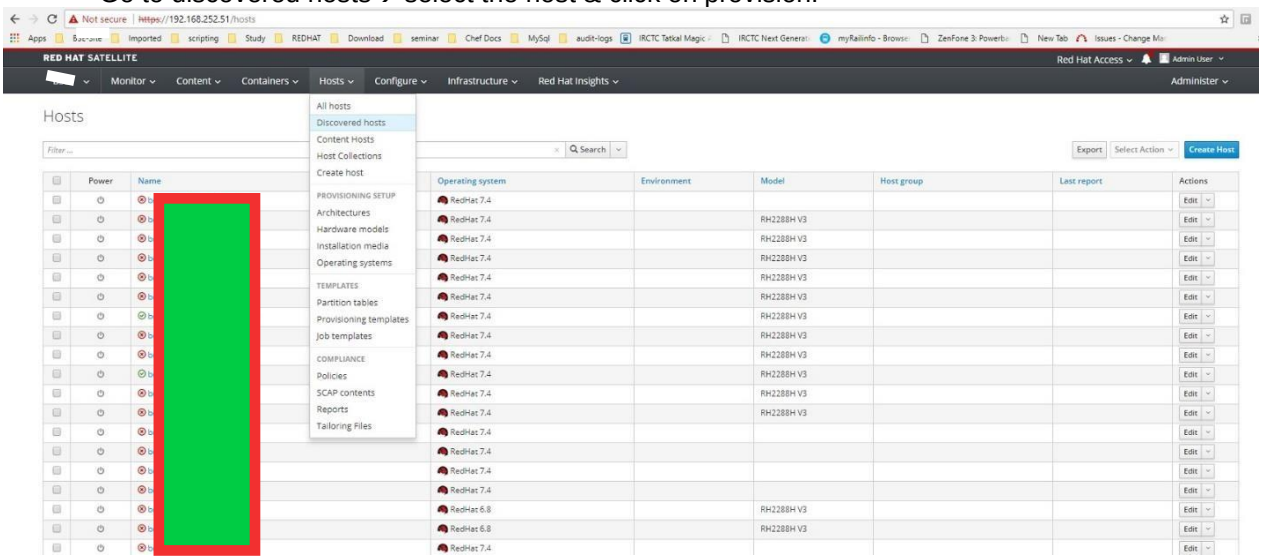
- In the custom facts section, do not enter facts if you want to have the provisioned server check against the search parameters of our defined web server and database server discovery rules. Depending on the hardware setup of the server, one of the rules will match, and the system will be auto-provisioned accordingly.
- Select confirm option.

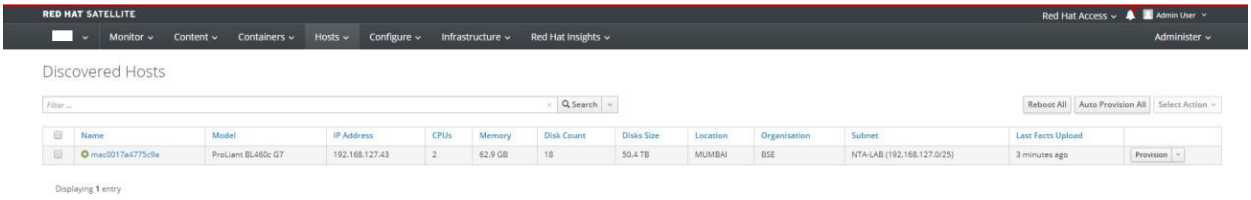


- The status showing success means the host will discovered in satellite 6 & ready for provision.

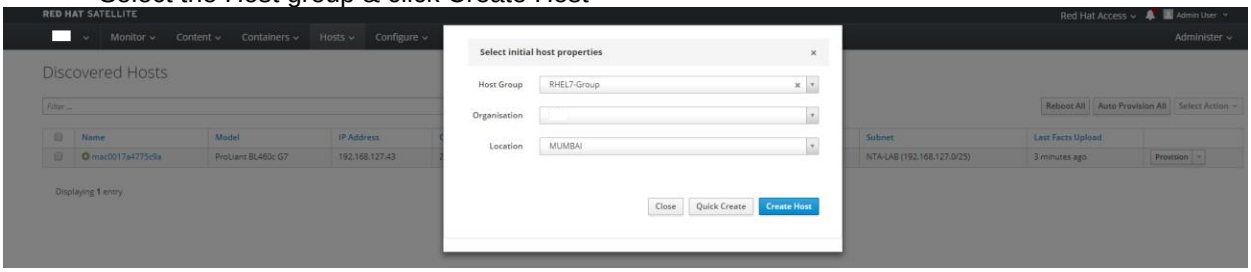


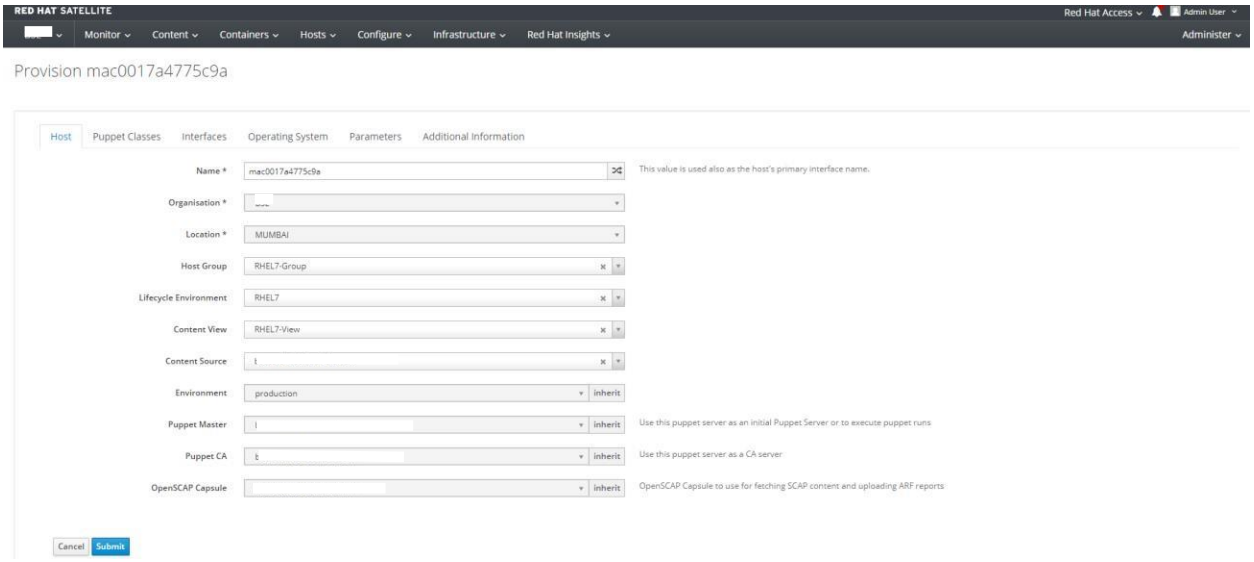
- Go to discovered hosts→ select the host & click on provision.



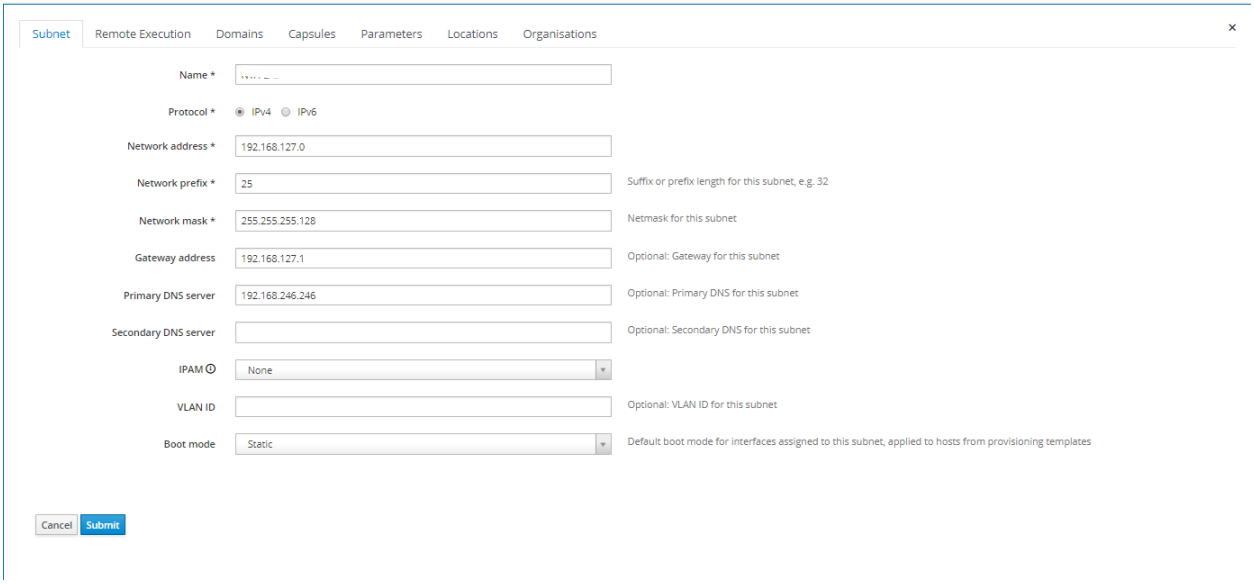


- Select the Host group & click Create Host

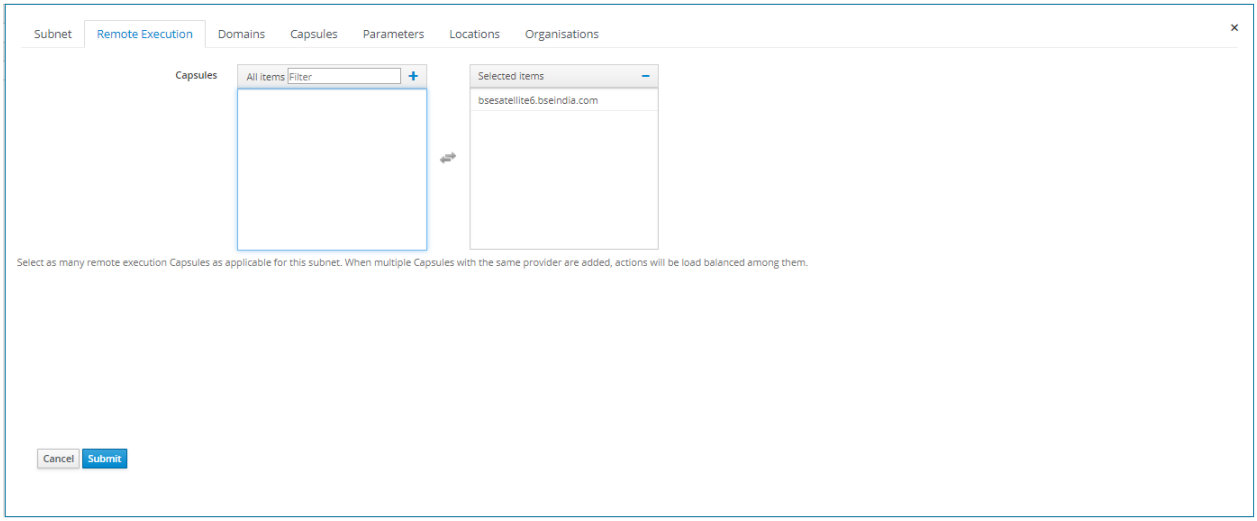




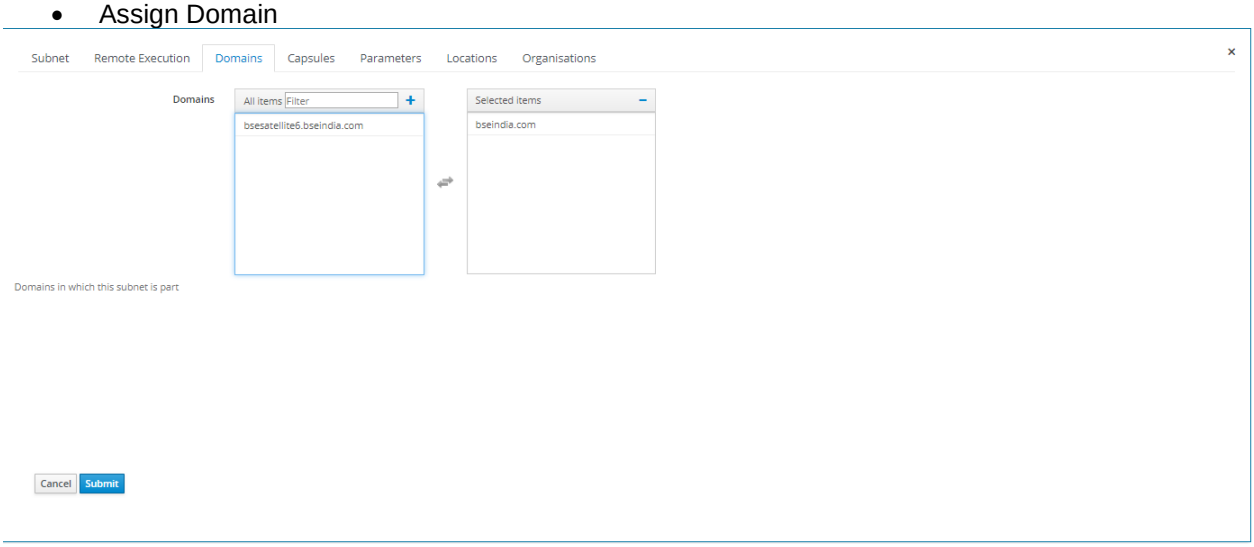
- In Interface tab select the appropriate subnet for the network.
- Create subnet if it is not there.
- Go to Infrastructure → subnets → Create subnet



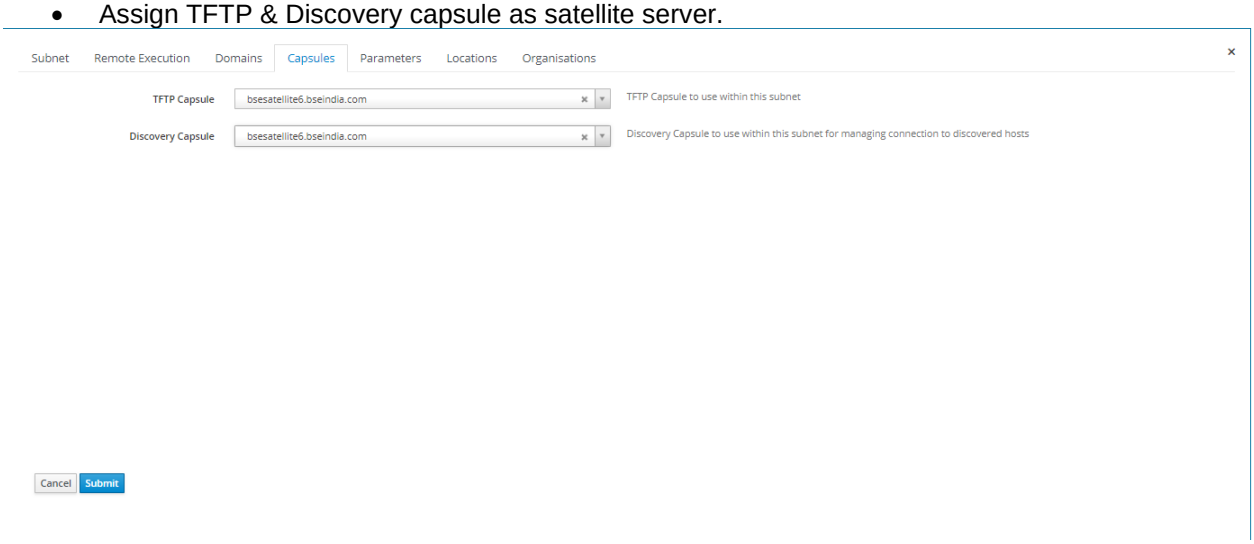
- Assign Remote Execution capsule



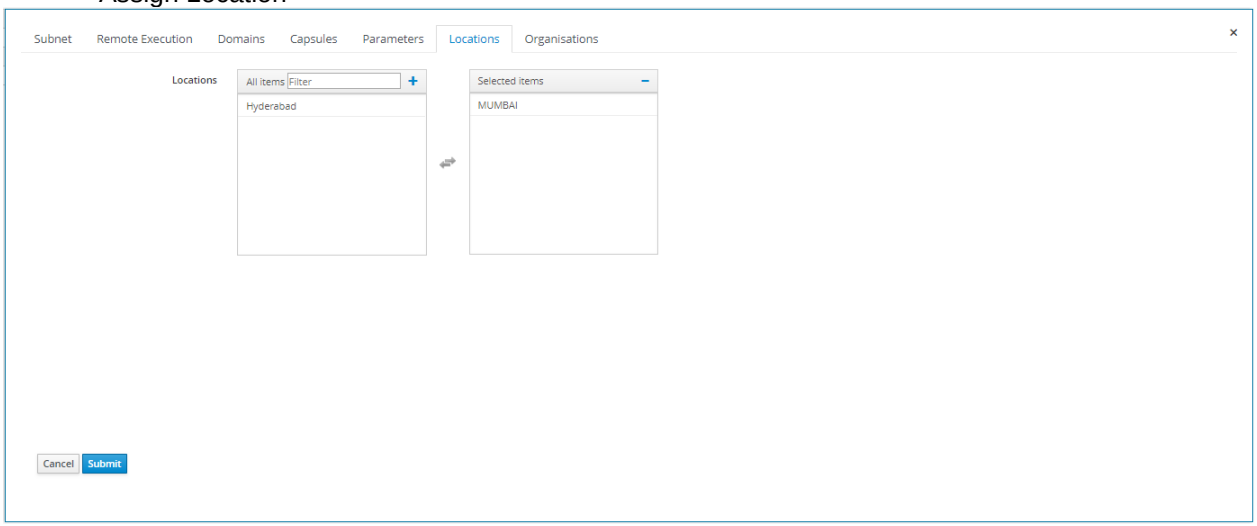
- Assign Domain



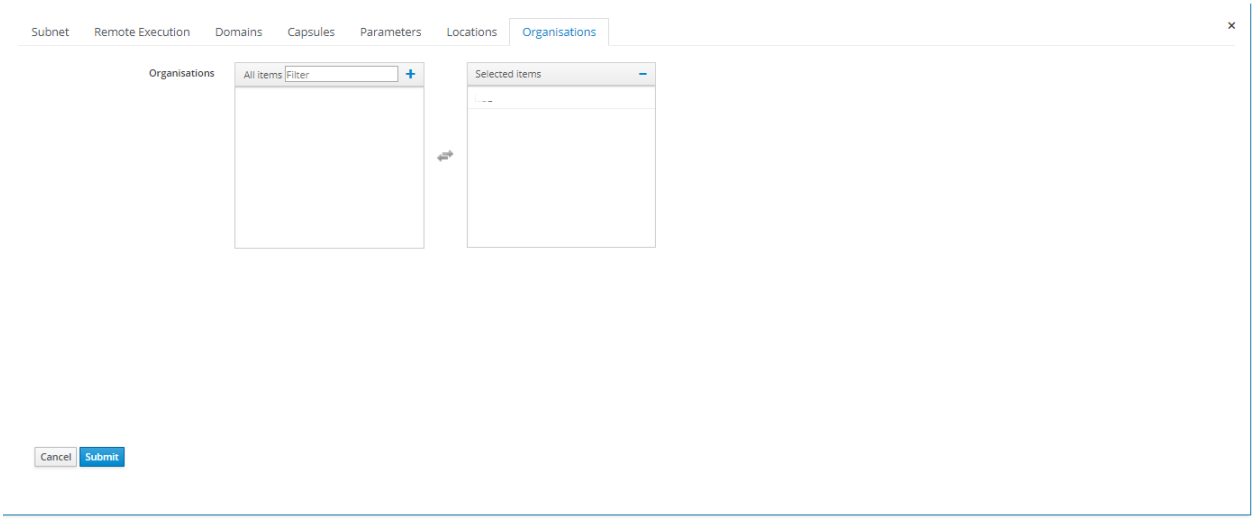
- Assign TFTP & Discovery capsule as satellite server.



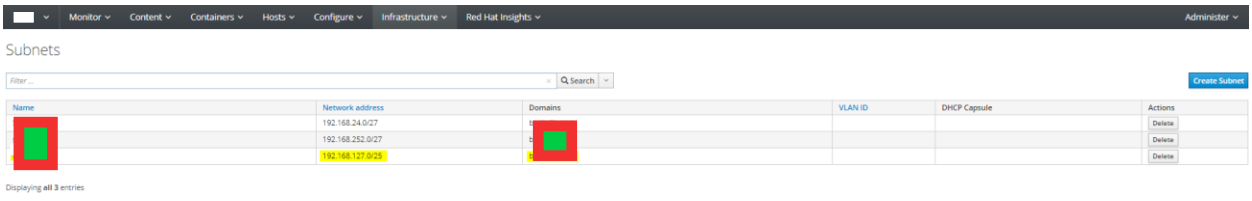
- Assign Location

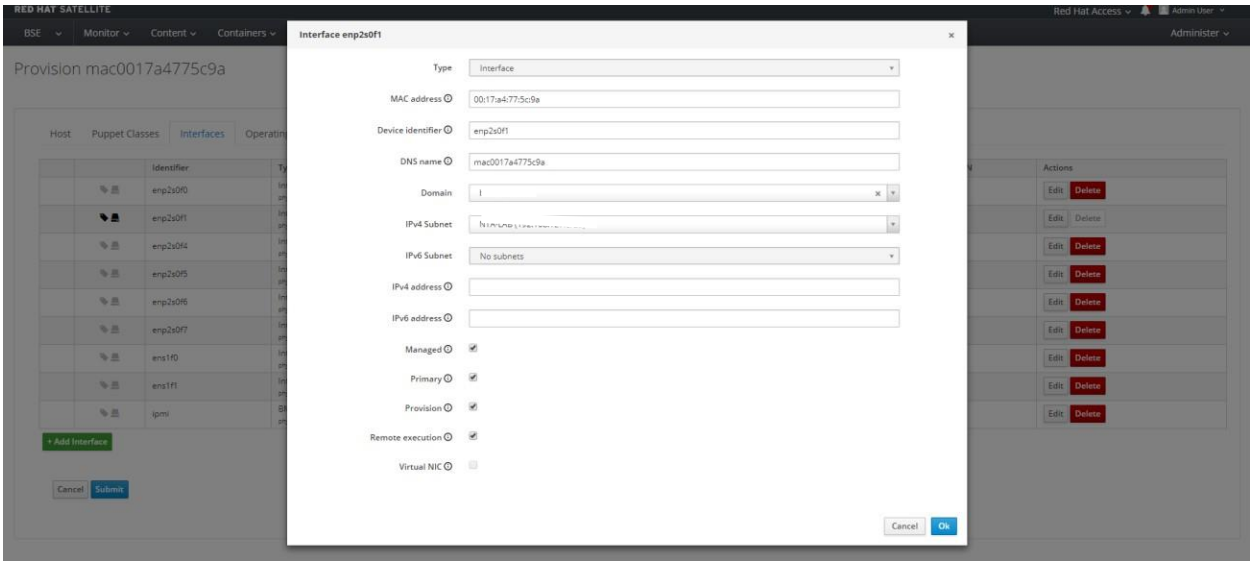


- Assign Organisation

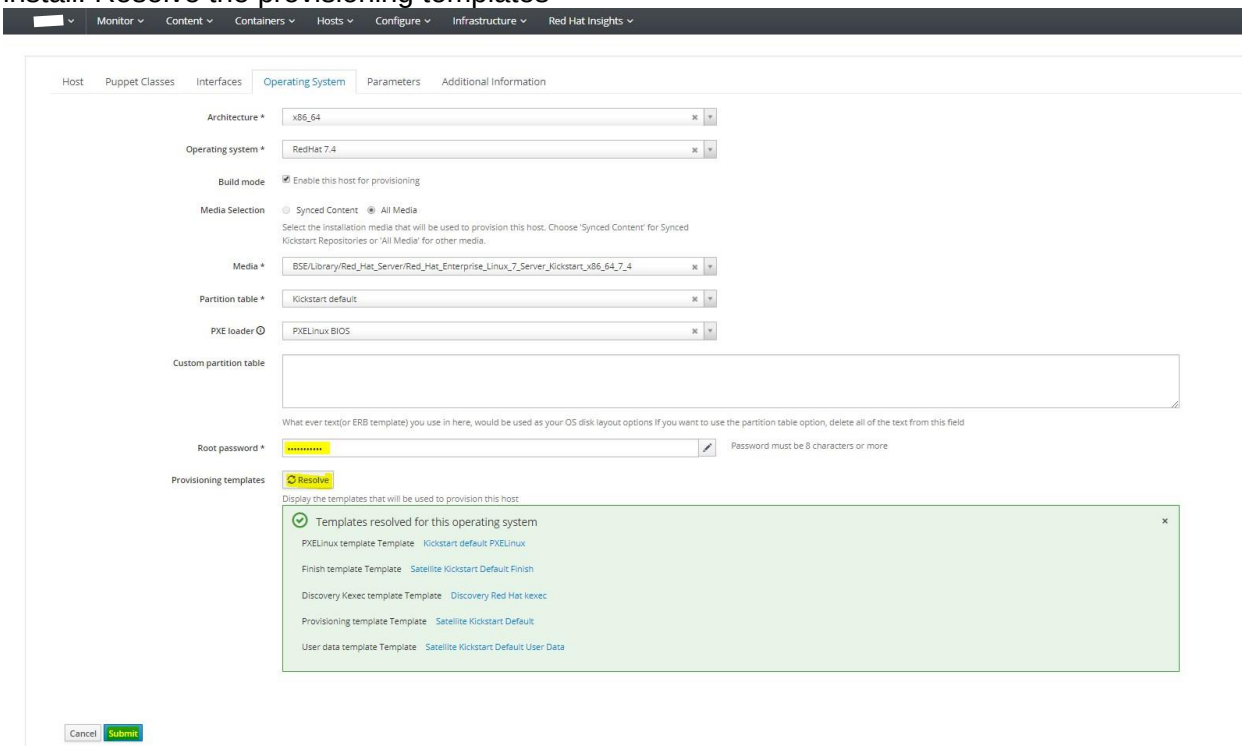


- Click on submit

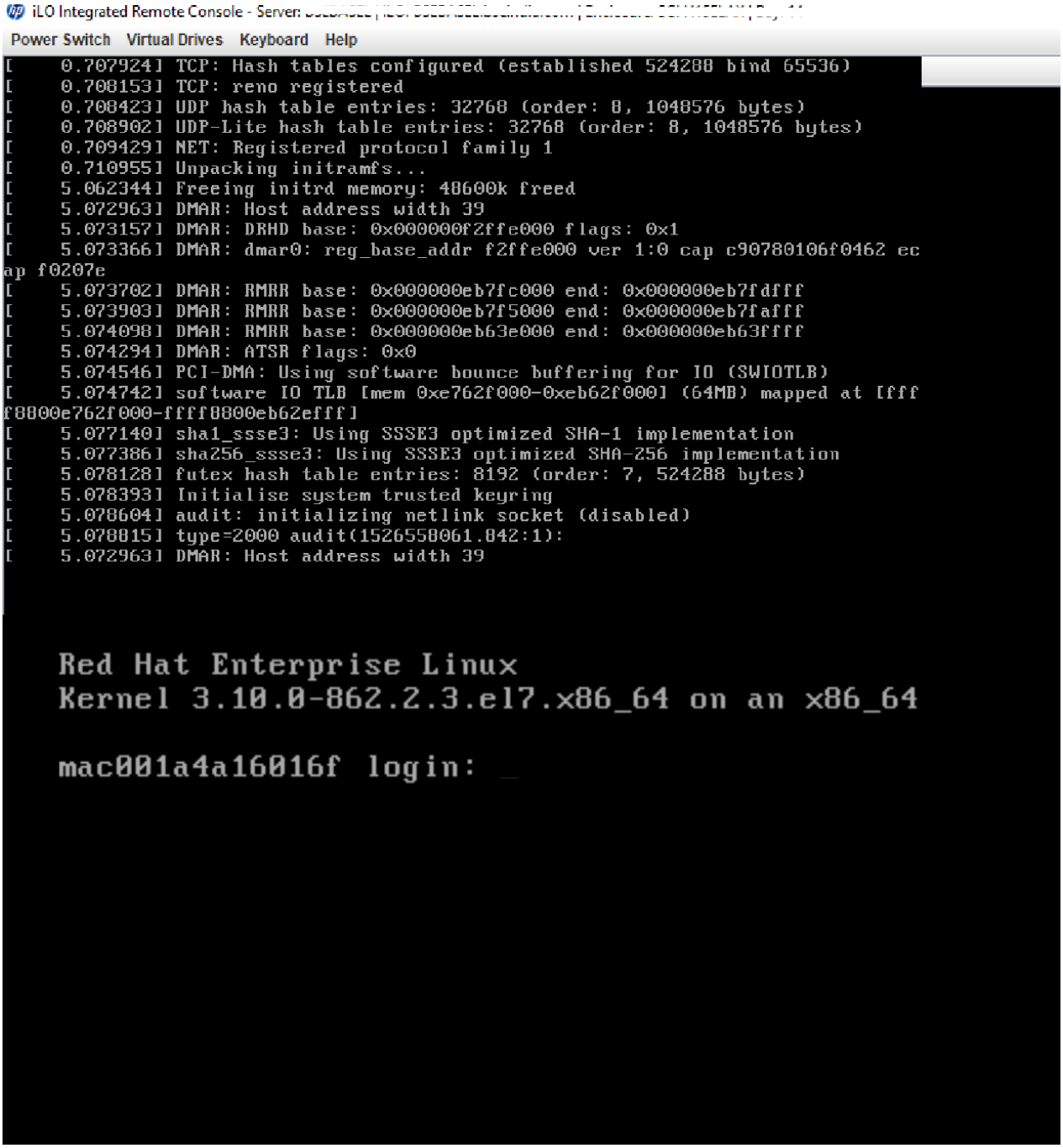




Select the OS, Media, and Partition Table you want to install. Resolve the provisioning templates



- Click on submit.



4. Puppet

Puppet is a tool for applying and managing system configurations. Puppet collects system information, or facts, and uses this information to create a customized system configuration using a set of modules.

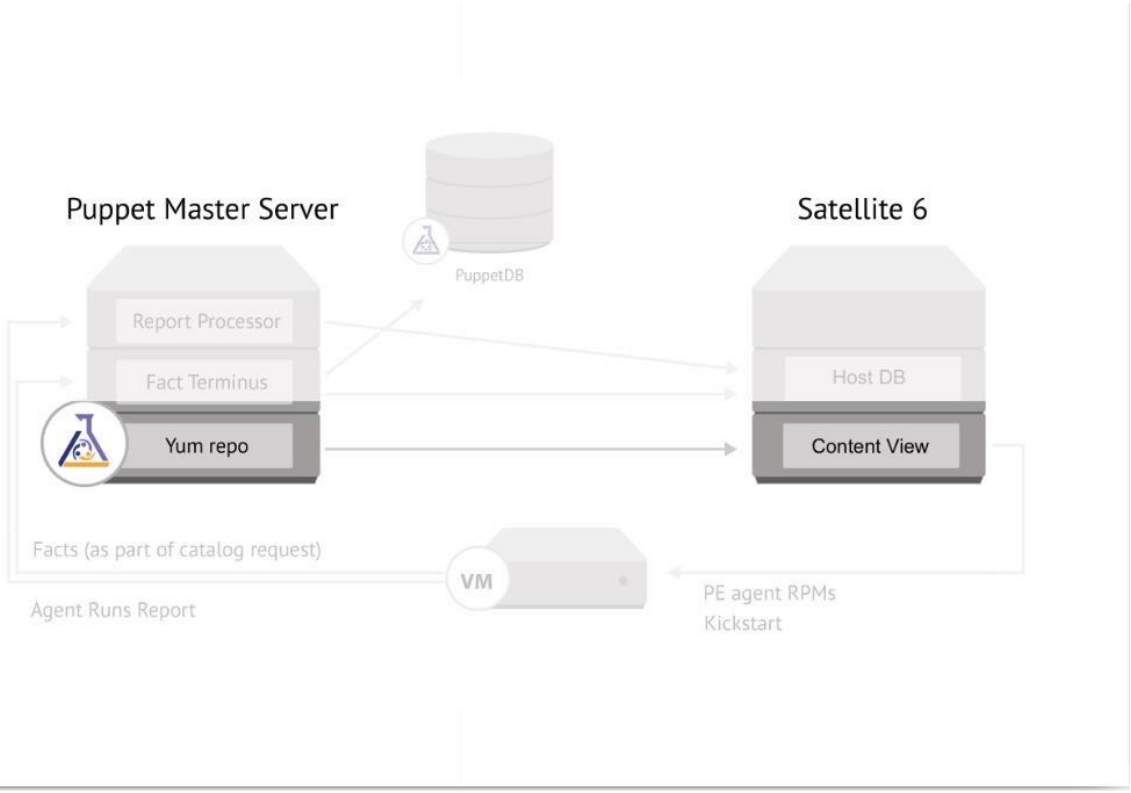
These modules contain parameters, conditional arguments, actions, and templates. Puppet is used as either a local system command line tool or in a client-server relationship where the server acts as the Puppet master and applies configuration to multiple client systems using a Puppet agent. This provides a way to automatically configure newly provisioned systems, either individually or simultaneously to create a specific infrastructure.

Defining the Puppet Workflow

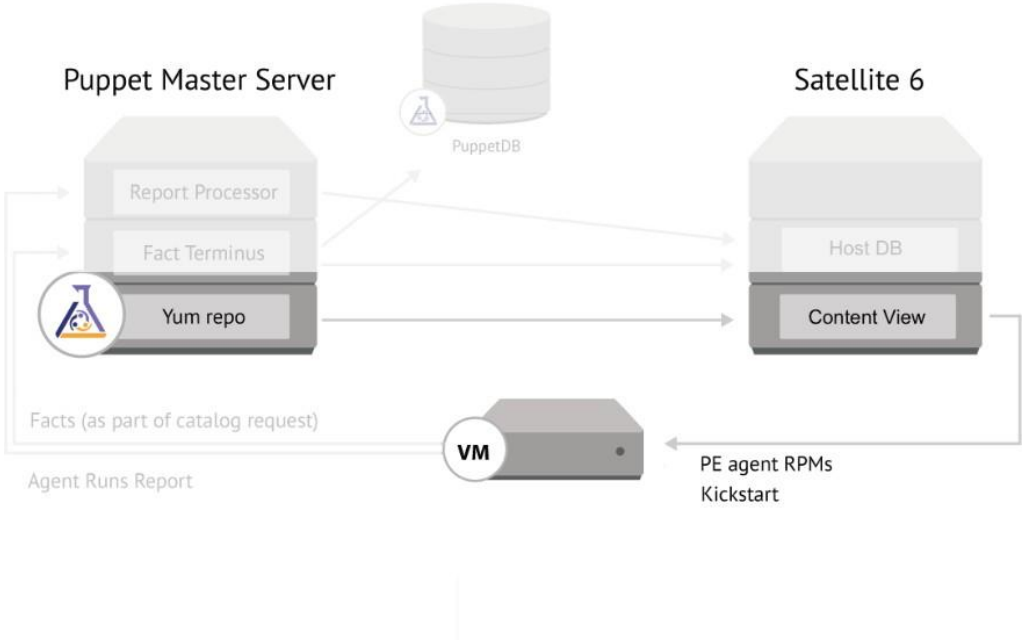
- Puppet uses the following workflow to apply configuration to a system.
- Collect facts about each system. These facts can include hardware, operating systems, package versions, and other information. The Puppet agent on each system collects this information and sends it to the Puppet master.
- The Puppet master generates a custom configuration for each system and sends it to the Puppet agent. This custom configuration is called a catalog.
- The Puppet agent applies the configuration to the system.
- The Puppet agent sends a report back to the Puppet master that indicates the changes applied and if any changes were unsuccessful.
- Third-party applications can collect these reports using Puppet's API.

How It Works

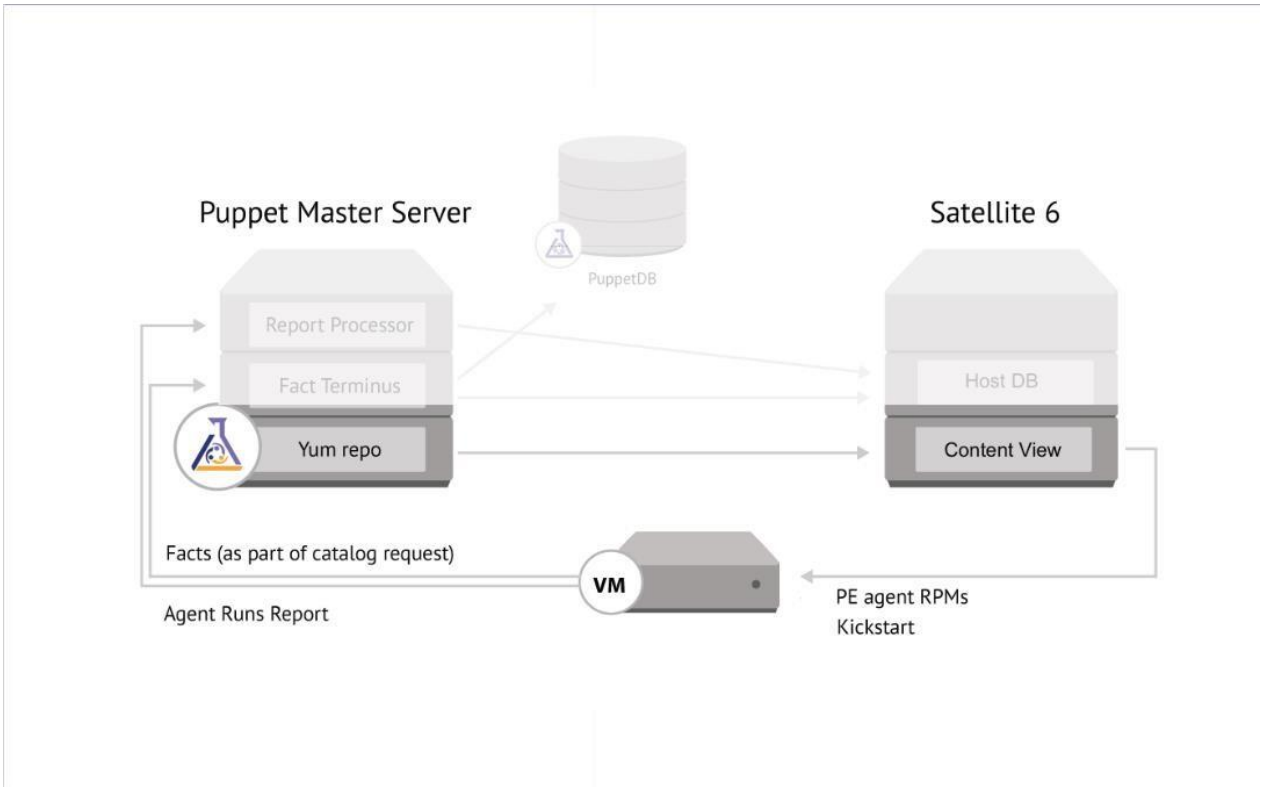
First, Red Hat Satellite is used to synchronize in the Puppet Enterprise (PE) Agent RPMs for the versions of RHEL that will be managed. This is done by means of a Custom Product, and a repository of type **yum**.
This allows the customer to deploy the Agent RPM during provisioning as part of a Content View. Additionally, as with all custom product, subscription tracking capabilities (via **hammer subscription list** for example)



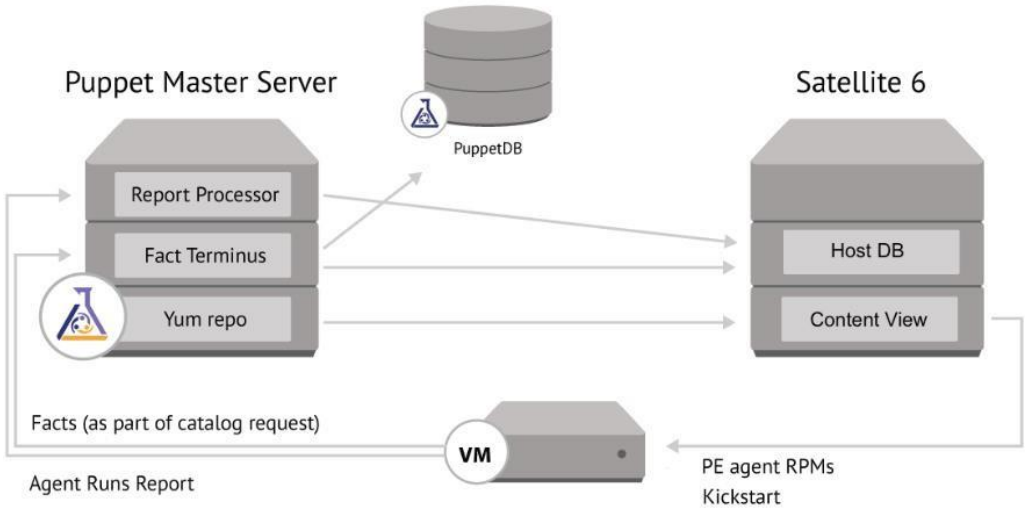
Next, the system is provisioned using any of the supported methods in Satellite. We provide a customized provisioning template & snippet (more on these below), which install the PE agent and not the Puppet Agent as shipped in the Satellite tools repository. Additionally, the provisioning templates configure the pe-puppet.conf file with the user-provided hostname of the Puppet Master that is to be used. A Hostgroup Parameter is used to configure the hostname of the Puppet Master and Puppet Certificate Authority.



After the node completes its first puppet run, its agent certificate is approved and it is classified (either manually or via rules). Ongoing puppet runs are reported to whichever Puppet Master was designated during provisioning.



Lastly, ongoing facts and reporting data are reported to Satellite via the custom reports processor and facts terminus. Additionally this data is stored in Puppet DB.



Satellite 6 uses Puppet in several ways:

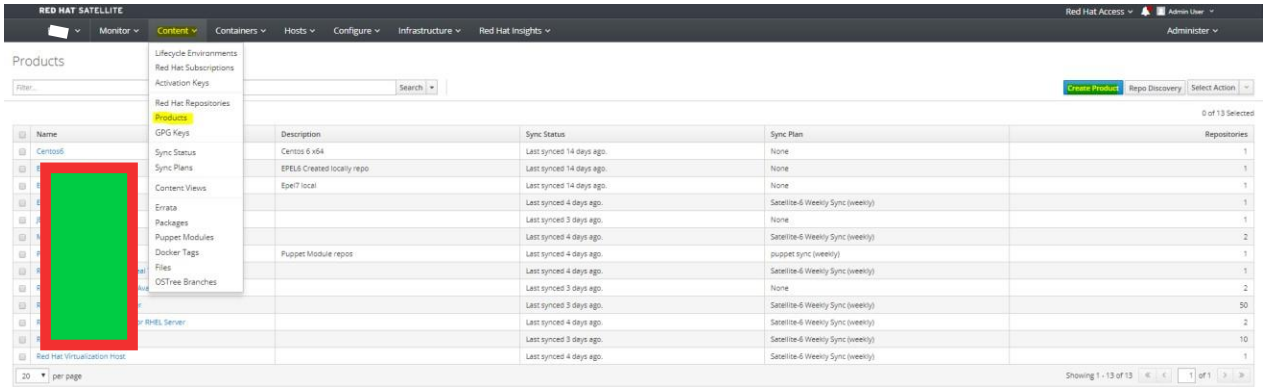
- Satellite 6 imports Puppet modules used to define the system configuration. This includes control over module versions and their environments.
- Satellite 6 imports sets of parameters, also known as Puppet class parameters, from Puppet modules. Users can accept the default values from Puppet classes or provide their own at a global or system-specific level.
- Satellite 6 triggers the execution of Puppet between the master and the respective agents on each system. Puppet runs can occur either:
- Automatically, such as after the provisioning process completes or as a daemon that checks and manages the machine's configuration over its lifecycle.
- Manually, such as needing to trigger an immediate Puppet run.
- Satellite 6 collects reports from Puppet after the configuration workflow completes. This helps with auditing and archiving system configuration over long term periods.

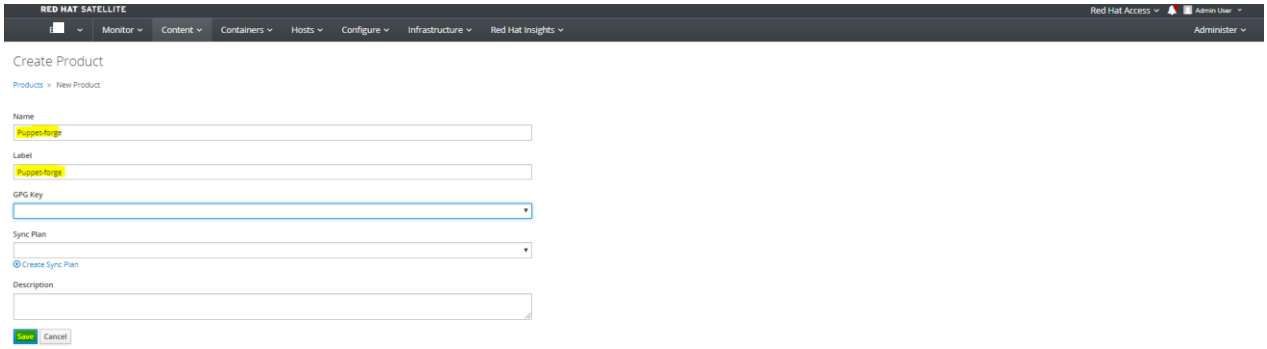
These functions provide an easy way for users to control system configuration aspects of the application lifecycle using Puppet.

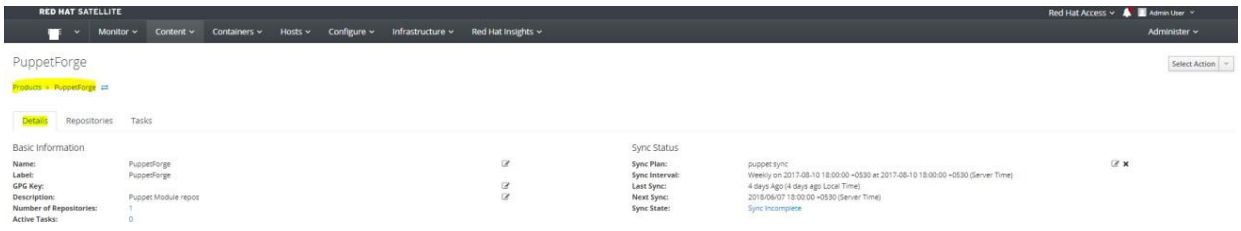
- **Creating Product**
- **Building puppet module**
- **Configuring Clients**
- **Deploying Puppet Modules**

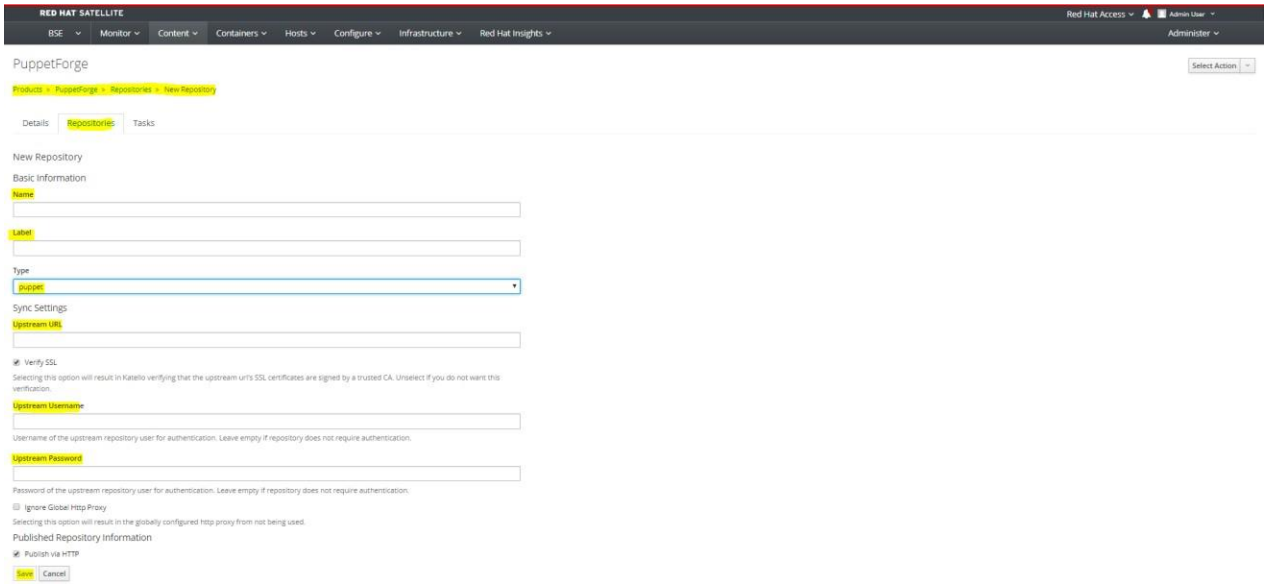
4.1. Creating Product

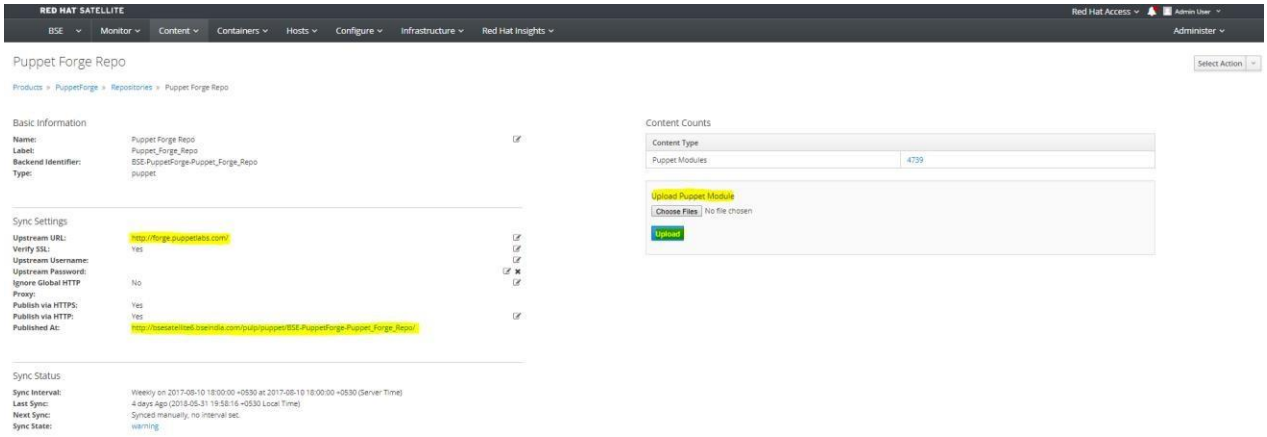
- Create Products → New Product → Puppet name & Save puppet repositories →type →Puppet →Enter URL→ Save











4.2. Building puppet module

- Creating simple **motd** module
  #**puppet module generate cloud (author name)-motd** →it will generate puppet module at /root/cloud-motd/

```
[root@h[redacted] ~]# puppet module generate cloud-motd
We need to create a metadata.json file for this module.  Please answer the
following questions; if the question is not applicable to this module, feel free
to leave it blank.

Puppet uses Semantic Versioning (semver.org) to version modules.
What version is this module?  [0.1.0]
-->

Who wrote this module?  [cloud]
-->

What license does this module code fall under?  [Apache 2.0]
-->

How would you describe this module in a single sentence?
-->

Where is this module's source code repository?
-->

Where can others go to learn more about this module?
-->

Where can others go to file issues about this module?
-->

----------------------------------------
{
  "name": "cloud-motd",
  "version": "0.1.0",
  "author": "cloud",
  "summary": null,
  "license": "Apache 2.0",
  "source": "",
  "project_page": null,
  "issues_url": null,
  "dependencies": [
    {"name":"puppetlabs-stdlib","version_requirement":">= 1.0.0"}
  ]
}
----------------------------------------

About to generate this metadata; continue? [n/Y]
--> y

Notice: Generating module at /root/cloud-motd...
Notice: Populating templates...
Finished; module generated in cloud-motd.
cloud-motd/Rakefile
cloud-motd/Gemfile
cloud-motd/manifests
cloud-motd/manifests/init.pp
cloud-motd/tests
cloud-motd/tests/init.pp
cloud-motd/metadata.json
cloud-motd/README.md
cloud-motd/spec
cloud-motd/spec/spec_helper.rb
cloud-motd/spec/classes
cloud-motd/spec/classes/init_spec.rb
[root@h[redacted] ~]#
```

#cd /root/cloud-motd/

#**vi metadata.json** → Basic Information about the puppet module. Remove dependencies from the file.

```
[root@b[redacted]6 bse-motd]# cd ../cloud-motd/
[root@b[redacted]6 cloud-motd]# ll
total 28
-rw-r----- 1 root root  242 Jun  4 21:25 Gemfile
drwxr-x--- 2 root root 4096 Jun  4 21:25 manifests
-rw-r----- 1 root root  270 Jun  4 21:25 metadata.json
-rw-r----- 1 root root  633 Jun  4 21:25 Rakefile
-rw-r----- 1 root root 2891 Jun  4 21:25 README.md
drwxr-x--- 3 root root 4096 Jun  4 21:25 spec
drwxr-x--- 2 root root 4096 Jun  4 21:25 tests
[root@bsesatellite6 cloud-motd]# vi metadata.json
{
  "name": "cloud-motd",
  "version": "0.1.0",
  "author": "cloud",
  "summary": null,
  "license": "Apache 2.0",
  "source": "",
  "project_page": null,
  "issues_url": null,
  "dependencies": [
    {"name":"puppetlabs-stdlib","version_requirement":">= 1.0.0"}
  ]
}
```

```
[redacted]~/cloud-motd
{
  "name": "cloud-motd",
  "version": "0.1.0",
  "author": "cloud",
  "summary": null,
  "license": "Apache 2.0",
  "source": "",
  "project_page": null,
  "issues_url": null,
  "dependencies": [
  ]
}
~
~
~
```

manifests/: Contains all of the manifests in the module.
init.pp: Contains a class definition. The init.pp class, if used, is the main class of the module. This class's name must match the module's name.

#cd
manifests
#vi init.pp

```
[root@[redacted] manifests]# pwd
/root/cloud-motd/manifests
[root@[redacted]ellite6 manifests]# cat init.pp
# == Class: motd
#
# Full description of class motd here.
#
# === Parameters
#
# Document parameters here.
#
# [*sample_parameter*]
#   Explanation of what this parameter affects and what it defaults to.
#   e.g. "Specify one or more upstream ntp servers as an array."
#
# === Variables
#
# Here you should define a list of variables that this module would require.
#
# [*sample_variable*]
#   Explanation of how this variable affects the funtion of this class and if
#   it has a default. e.g. "The parameter enc_ntp_servers must be set by the
#   External Node Classifier as a comma separated list of hostnames." (Note,
#   global variables should be avoided in favor of class parameters as
#   of Puppet 2.6.)
#
# === Examples
#
#  class { 'motd':
#    servers => [ 'pool.ntp.org', 'ntp.local.company.com' ],
#  }
#
# === Authors
#
# Author Name <author@domain.com>
#
# === Copyright
#
# Copyright 2018 Your name here, unless otherwise noted.
#
class motd {


}
[root@[redacted] manifests]#
```

root@[redacted]/cloud-motd/manifests

```
# == Class: motd
#
# Full description of class motd here.
#
# === Parameters
#
# Document parameters here.
#
# [*sample_parameter*]
#   Explanation of what this parameter affects and what it defaults to.
#   e.g. "Specify one or more upstream ntp servers as an array."
#
# === Variables
#
# Here you should define a list of variables that this module would require.
#
# [*sample_variable*]
#   Explanation of how this variable affects the funtion of this class and if
#   it has a default. e.g. "The parameter enc_ntp_servers must be set by the
#   External Node Classifier as a comma separated list of hostnames." (Note,
#   global variables should be avoided in favor of class parameters as
#   of Puppet 2.6.)
#
# === Examples
#
#  class { 'motd':
#    servers => [ 'pool.ntp.org', 'ntp.local.company.com' ],
#  }
#
# === Authors
#
# Author Name <author@domain.com>
#
# === Copyright
#
# Copyright 2018 Your name here, unless otherwise noted.
#
class motd ($message = "Welcome to [redacted] "){
file {'/etc/motd':
ensure => 'present',
content => "$message \n",
 }
}
~
```

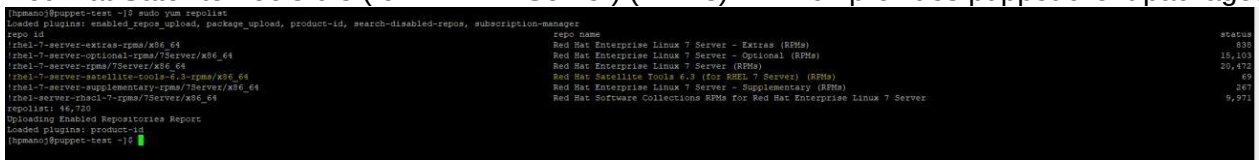
class motd($message = “ This is the default
message”){ file { ‘/etc/motd’:
ensure => ‘present’,
content => “$message \n”,
}
}

To build the puppet module
#puppet module build cloud-motd

```
[root@b[redacted]6 manifests]# cd
[root@b[redacted]6 ~]# puppet module build cloud-motd
Notice: Building /root/cloud-motd for release
Module built: /root/cloud-motd/pkg/cloud-motd-0.1.0.tar.gz
[root@b[redacted]6 ~]#
```

4.3. Configuring Clients

- Configuring Puppet agent
  #Subscription-manager repos –enable=”reponame”
  Red Hat Satellite Tools 6.3 (for RHEL 7 Server) (RPMs) →which provides puppet client packages



#yum install puppet

#VI

/etc/puppet/puppet.conf

Adding server parameter
echo " **server** = **example.satellite.com**" >> **/etc/puppet/puppet.conf**
echo " **environment** = **KT_BSE_Library_Puppet_View_51**" >> **/etc/puppet/puppet.conf**

```
[hpmanoj@puppet-test ~]$ cat /etc/puppet/puppet.conf
[main]
    # The Puppet log directory.
    # The default value is '$vardir/log'.
    logdir = /var/log/puppet

    # Where Puppet PID files are kept.
    # The default value is '$vardir/run'.
    rundir = /var/run/puppet

    # Where SSL certificates are kept.
    # The default value is '$confdir/ssl'.
    ssldir = $vardir/ssl

[agent]
    # The file in which puppetd stores a list of the classes
    # associated with the retrieved configuratiion.  Can be loaded in
    # the separate ``puppet`` executable using the ``--loadclasses``
    # option.
    # The default value is '$confdir/classes.txt'.
    classfile = $vardir/classes.txt

    # Where puppetd caches the local configuration.  An
    # extension indicating the cache format is added automatically.
    # The default value is '$confdir/localconfig'.
    localconfig = $vardir/localconfig
    server = [redacted].com
    environment = KT_BSE_Library_Puppet_View_51
[hpmanoj@puppet-test ~]$
```

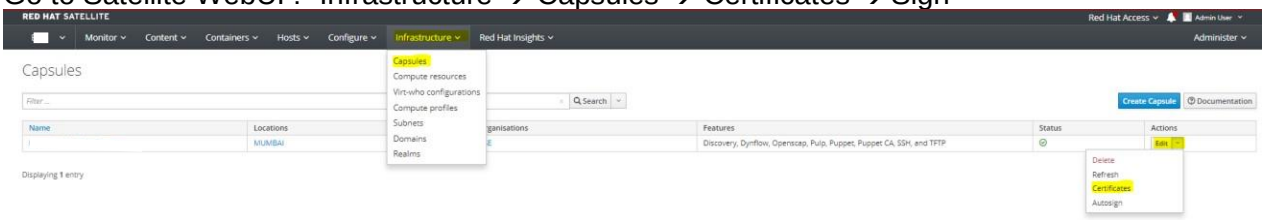
Sending certificate request to puppet master

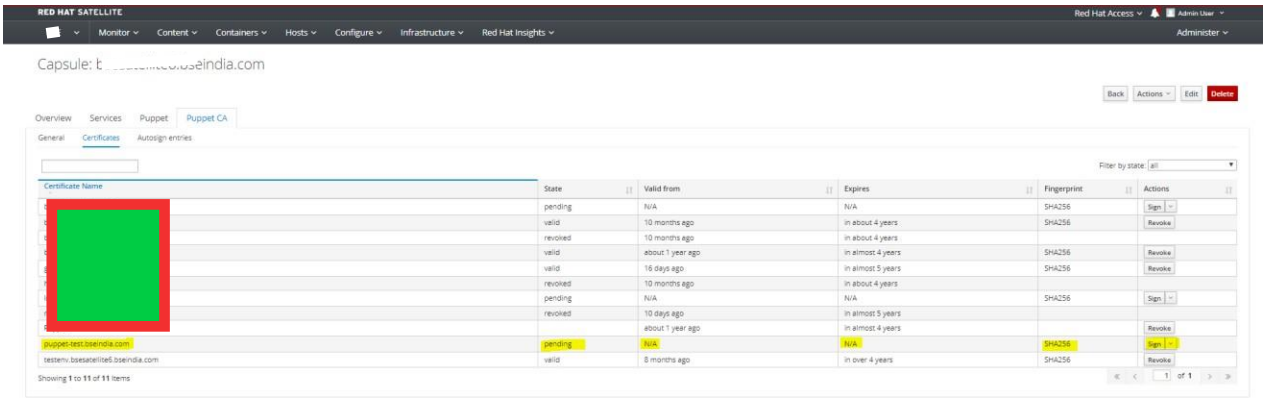
#puppet agent –tv

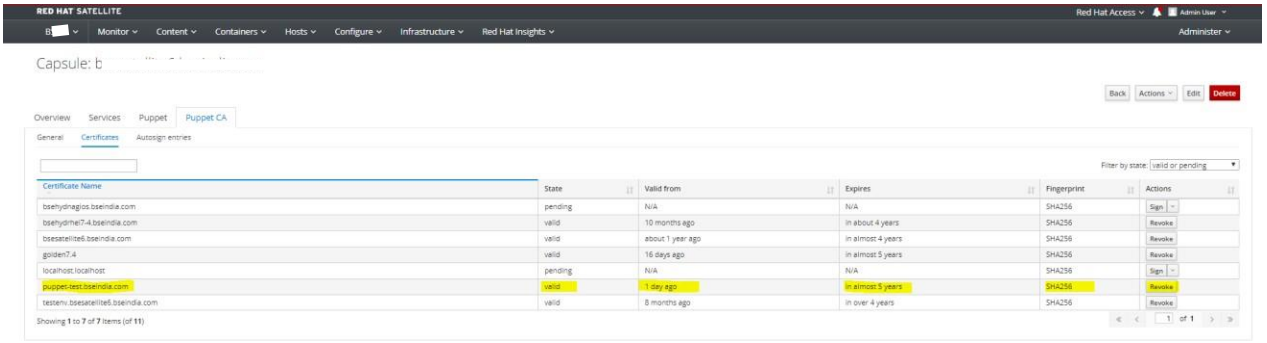
```
[hpmanoj@puppet-test ~]$ sudo puppet agent -tv
Info: Caching certificate for ca
Info: csr_attributes file loading from /etc/puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for puppet-test.bseindia.com
Info: Certificate Request fingerprint (SHA256): AC:15:74:69:7D:08:0D:0C:E9:53:02:EF:DC:43:FC:1B:C3:EB:17:A4:14:CC:9F:F9:09:74:85:A4:E5:53:CF:E5
Info: Caching certificate for ca
Exiting; no certificate found and waitforcert is disabled
[hpmanoj@puppet-test ~]$
```

4.4. Deploying Puppet Modules

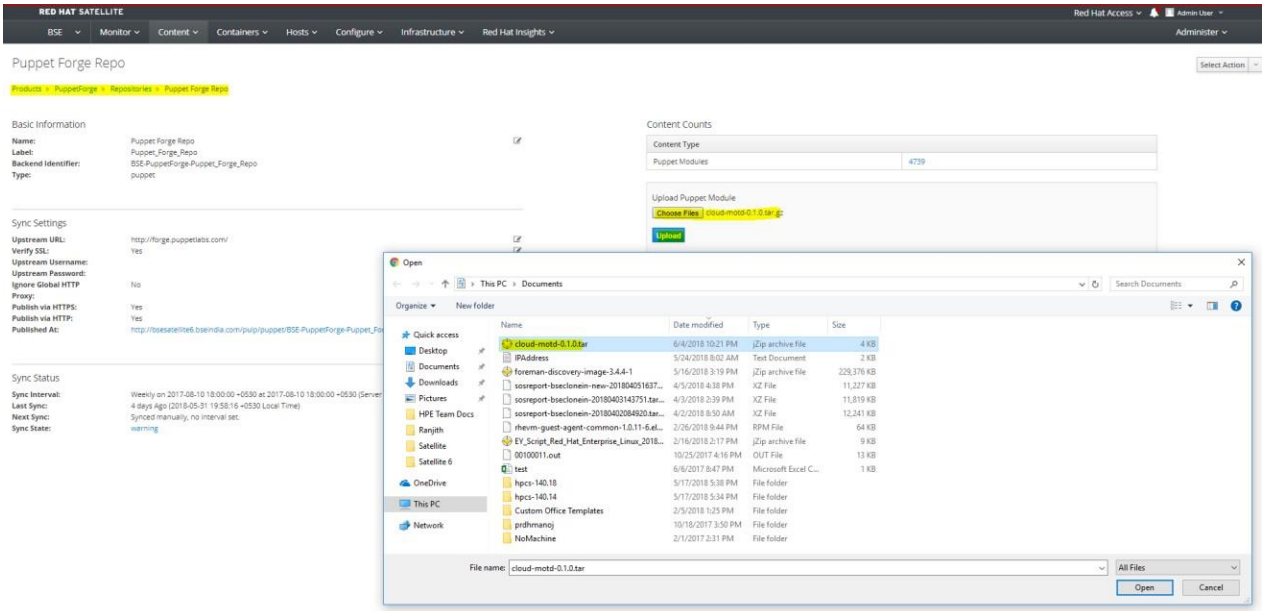
- Go to Satellite WebUI :- Infrastructure → Capsules → Certificates → Sign

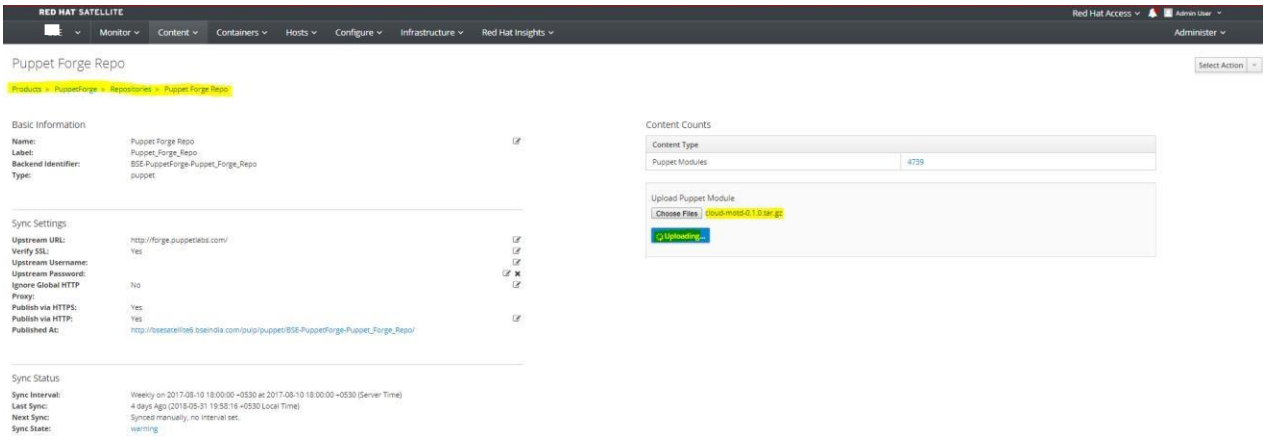




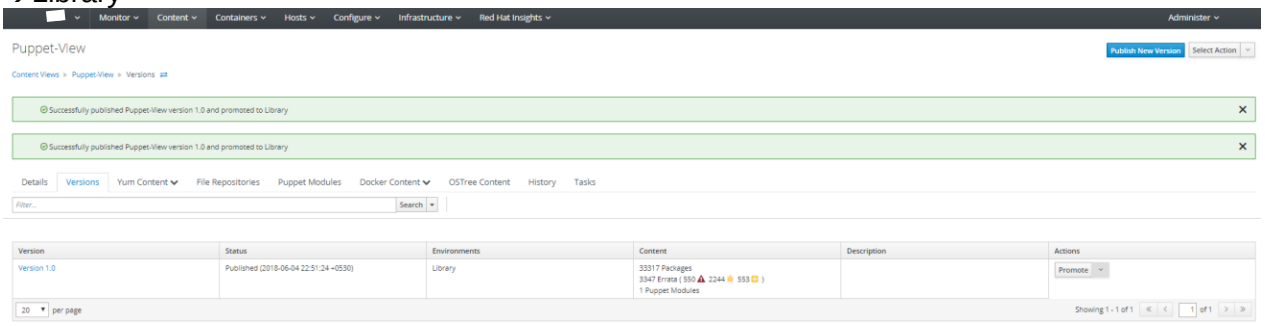


Uploading Puppet Module
Products → Puppet module name → Repositories → Browse puppet module →**cloud-motd.tar.gz** →Open→ Upload

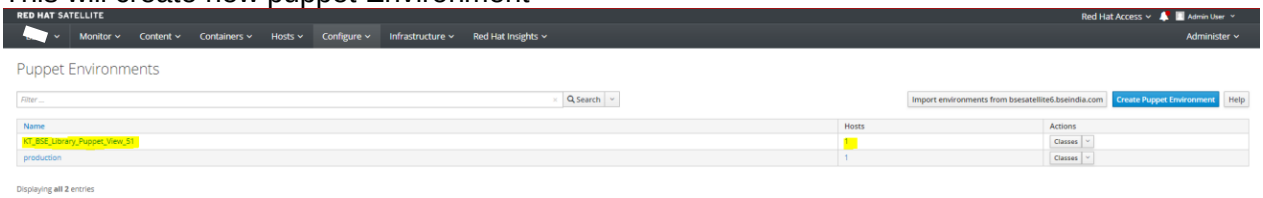




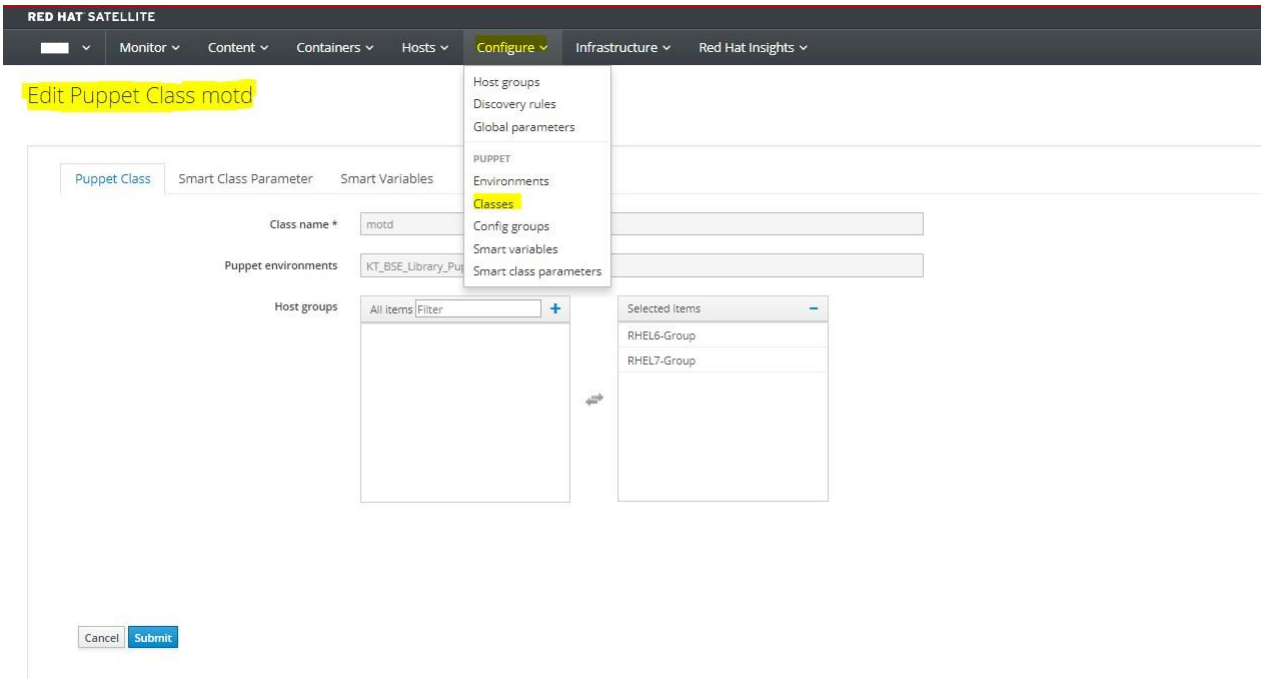
Content Views→ Content name→Puppet Modules→Select Version→ Publish New Version →Library

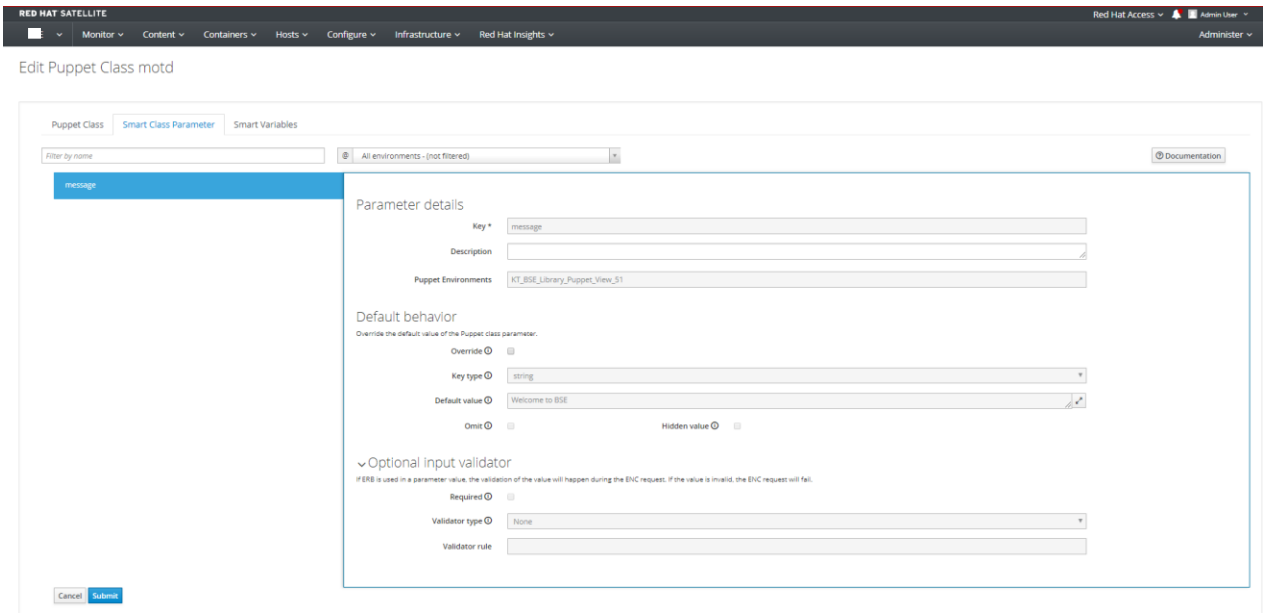


This will create new puppet Environment

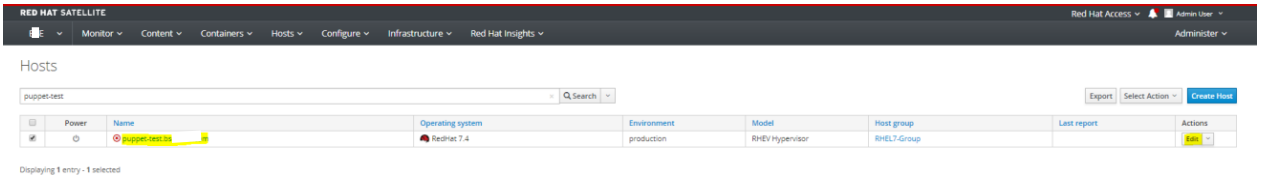


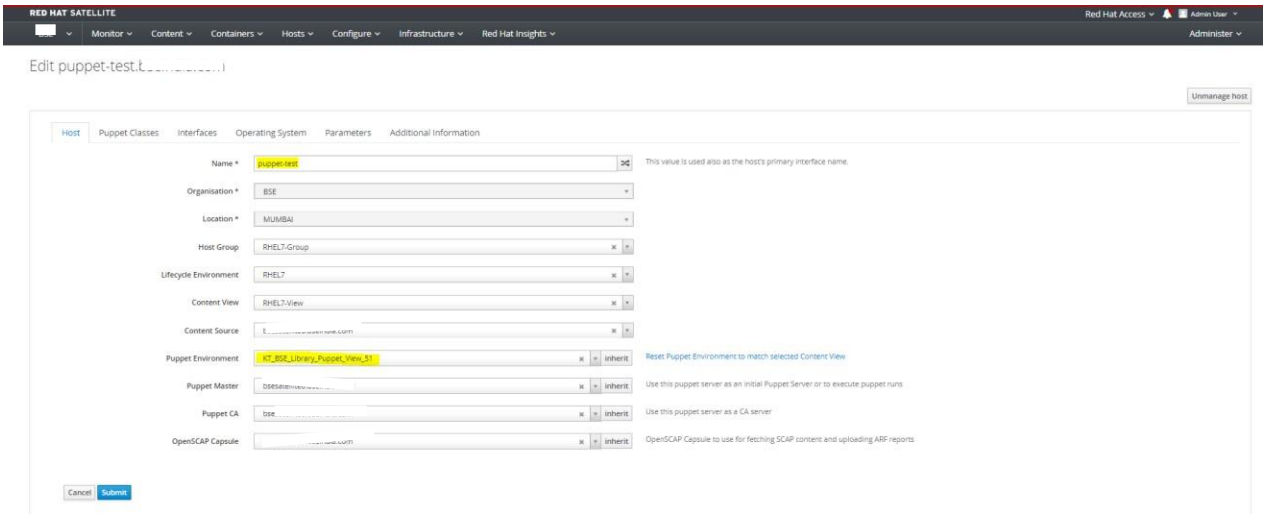
Goto Puppet Classes→smart class parameter →message→Default value(Change select Override this)→Submit

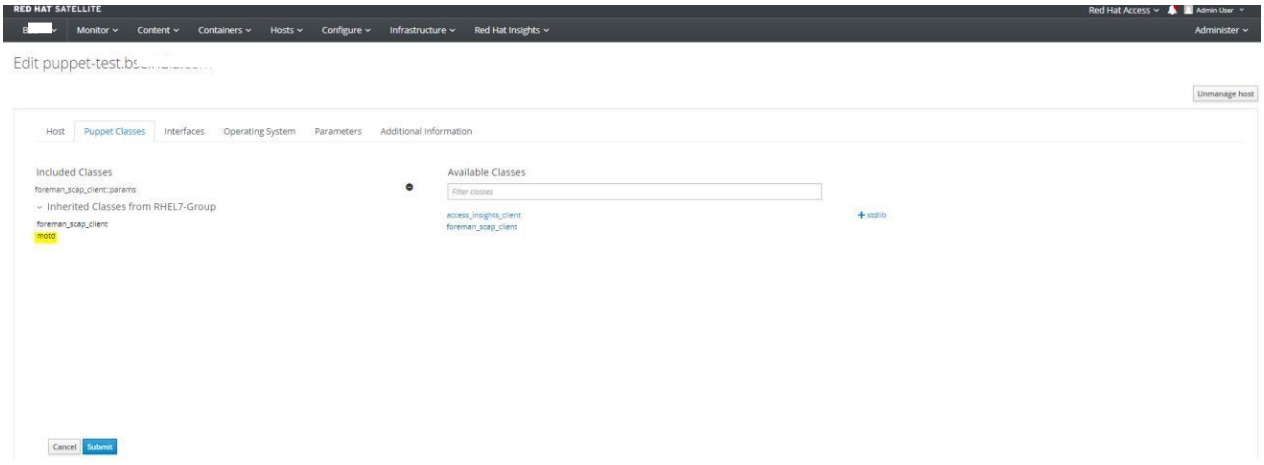




Hosts→Select Host→Select action→Edit the host→ Add Lifecycle Environment & Content View

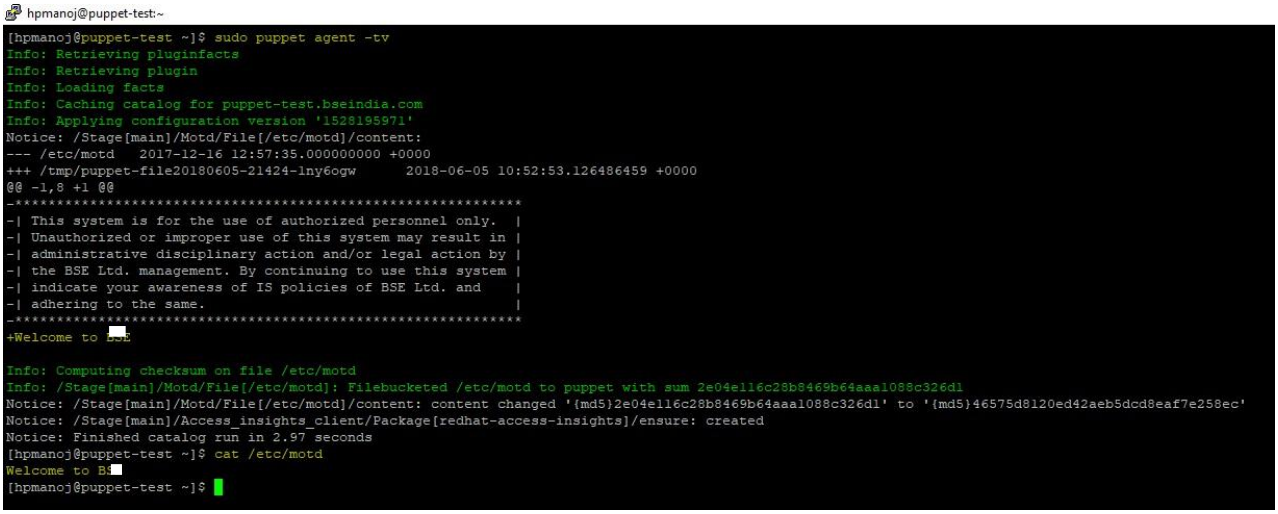







```
[hpmanoj@puppet-test ~]$ cat /etc/motd
************************************************************
| This system is for the use of authorized personnel only.  |
| Unauthorized or improper use of this system may result in |
| administrative disciplinary action and/or legal action by |
| the BSE Ltd. management. By continuing to use this system |
| indicate your awareness of IS policies of BSE Ltd. and    |
| adhering to the same.                                     |
************************************************************
[hpmanoj@puppet-test ~]$ date
Tue Jun  5 10:52:15 UTC 2018
[hpmanoj@puppet-test ~]$
```

On client server
#puppet agent –
tv



5. Remote Execution

Remote execution, enabling users to take multiple actions against a group of systems while automating workflows. With these new capabilities, users can reboot a system after a patch install, or conduct rolling upgrades across hundreds of systems with the same ease and consistency as updating a single system.
New scheduling features and dashboards enhance interaction with systems under Red Hat Satellite management, bringing increased efficiency to its provisioning and discovery processes.

5.1. Remote Execution using non-root users

- On the client machine first create a user and add it to the sudoers.

```
[root@mac001a4a16016f ~]# useradd -m hpmanoj
[root@mac001a4a16016f ~]# echo "hpmanoj   ALL=NOPASSWD:   ALL" > /etc/sudoers.d/hpmanoj
[root@mac001a4a16016f ~]#
```

- Check if hpmanoj user can run the sudo commands without password.

```
[root@mac001a4a16016f ~]# su - hpmanoj
Last login: Sat May 19 14:20:41 UTC 2018 on pts/0
[hpmanoj@mac001a4a16016f ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[hpmanoj@mac001a4a16016f ~]$ sudo cat /etc/shadow
root:$5$m0FcvlBi$uwRBPf9UZtd/m8PTbxScwBor0wtaq5mIyxUm35L0gN9::0:99999:7:::
bin:*:16925:0:99999:7:::
daemon:*:16925:0:99999:7:::
adm:*:16925:0:99999:7:::
lp:*:16925:0:99999:7:::
sync:*:16925:0:99999:7:::
shutdown:*:16925:0:99999:7:::
halt:*:16925:0:99999:7:::
mail:*:16925:0:99999:7:::
operator:*:16925:0:99999:7:::
games:*:16925:0:99999:7:::
ftp:*:16925:0:99999:7:::
nobody:*:16925:0:99999:7:::
systemd-network:!!:17668::::::
dbus:!!:17668::::::
polkitd:!!:17668::::::
ntp:!!:17668::::::
postfix:!!:17668::::::
chrony:!!:17668::::::
sshd:!!:17668::::::
puppet:!!:17668::::::
hpmanoj:!!:17670:0:99999:7:::
[hpmanoj@mac001a4a16016f ~]$
```

- Create ssh keygen for the user & Copy over the foreman-proxy public key using curl into authorized keys file under hpmanoj user account

```
[hpmanoj@mac001a4a16016f ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hpmanoj/.ssh/id_rsa):
Created directory '/home/hpmanoj/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hpmanoj/.ssh/id_rsa.
Your public key has been saved in /home/hpmanoj/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:nuQf+EoBi/D9zUlREchKDM+4HvPz7p9s5kwjnf+GxRY hpmanoj@mac001a4a16016f.bseindia.com
The key's randomart image is:
+---[RSA 2048]----+
|        .o . .+o |
|         +o o.   |
|  .   . ..o..    |
|   o o o ..  . E |
|    o o S   .  ..|
|       * X o . .+|
|        O * o =+ |
|       . o + =++.|
|        ..oo+=*.+|
+----[SHA256]-----+
[hpmanoj@mac001a4a16016f ~]$ curl https://[redacted]satellite6.[redacted].com:9090/ssh/pubkey >> .ssh/authorized_keys
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   413  100   413    0     0   1324      0 --:--:-- --:--:-- --:--:--  1327
[hpmanoj@mac001a4a16016f ~]$ chmod 600 .ssh/authorized_keys
[hpmanoj@mac001a4a16016f ~]$ restorecon -Rv .ssh/
[hpmanoj@mac001a4a16016f ~]$
```

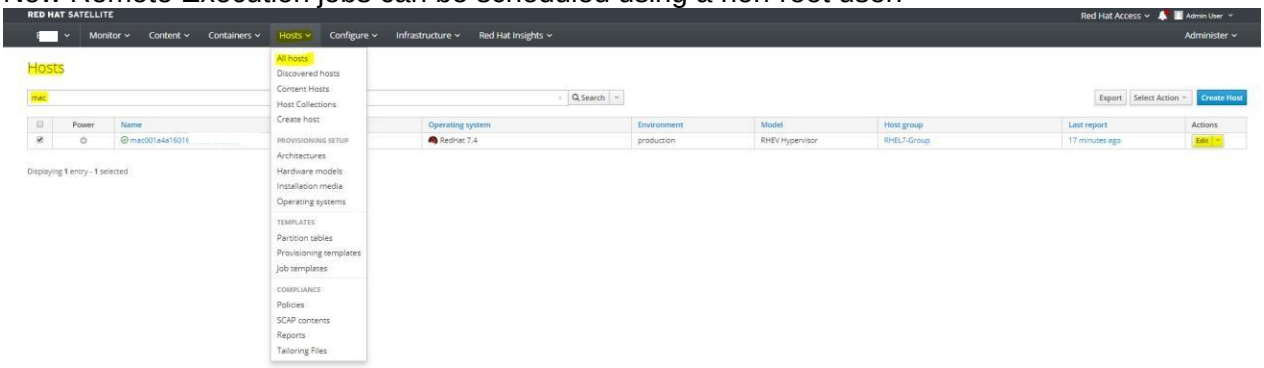
$ssh-keygen –t rsa
$ curl https://example.satellite.com:9090/ssh/pubkey >> .ssh/authorized_keys
$ chmod 600 .ssh/authorized_keys
$ restorecon –Rv .ssh/

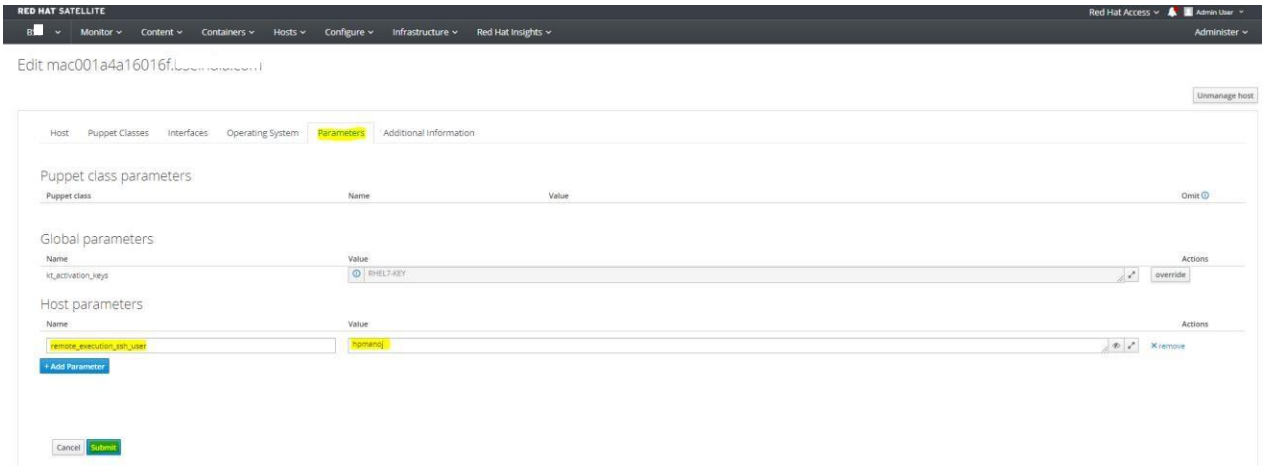
Now check from the Satellite server if **hpmanoj** user can execute the **sudo** commands without requiring any password interactions.

```
[root@b[redacted] ~]# su - foreman-proxy -s /bin/bash
Last login: Sat May 19 20:44:09 IST 2018 on pts/0
-bash-4.2$ ssh -t -i .ssh/id_rsa_foreman_proxy hpmanoj@192.168.127.55 'sudo df -h'
Filesystem                              Size  Used Avail Use% Mounted on
/dev/mapper/rhel_mac001a4a16016f-root   47G  1.8G   46G   4% /
devtmpfs                               875M     0  875M   0% /dev
tmpfs                                  887M     0  887M   0% /dev/shm
tmpfs                                  887M  8.7M  878M   1% /run
tmpfs                                  887M     0  887M   0% /sys/fs/cgroup
/dev/sda1                             1014M  168M  847M  17% /boot
tmpfs                                  178M     0  178M   0% /run/user/0
tmpfs                                  178M     0  178M   0% /run/user/1000
Connection to 192.168.127.55 closed.
-bash-4.2$
```

$su - foreman-proxy -s /bin/bash
$ssh -t -i .ssh/id_rsa_foreman_proxy hpmanoj@192.168.127.55 'sudo df -h'
If foreman-proxy user can execute commands without any issues, then add the following parameter to the client host from the Satellite Server.
Satellite webUI >> Hosts >> All Hosts >> Edit the mac001a4a16016f.[redacted].com >> Parameters tab >> Host Parameters >> Add Parameter >> Specify Name as **remote_execution_ssh_user** and set its value to hpmanoj >> click Submit

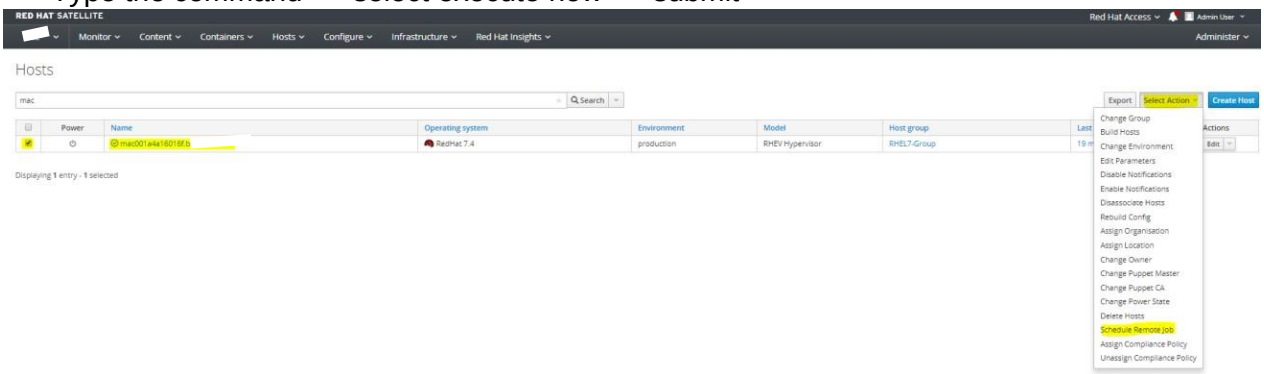
Now Remote Execution jobs can be scheduled using a non-root user.

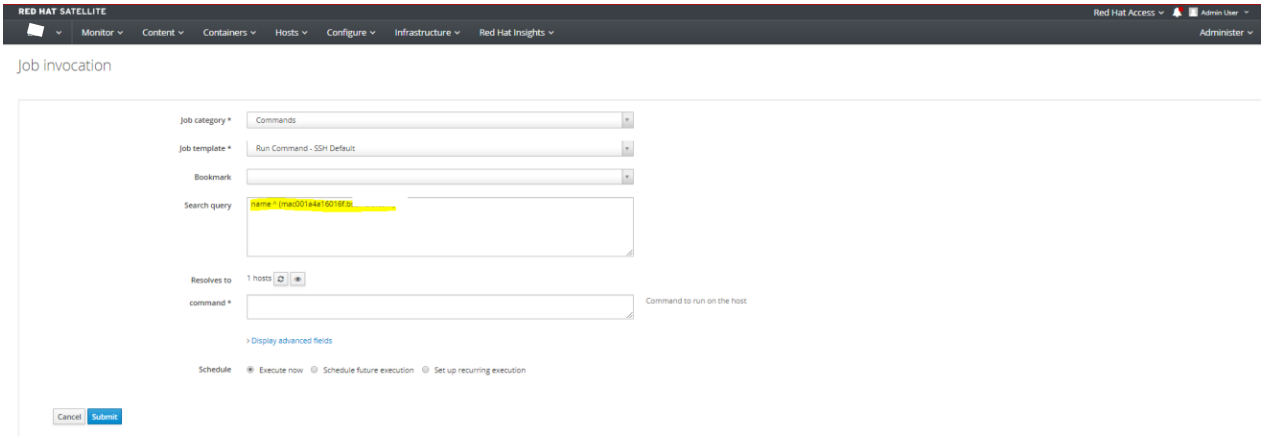


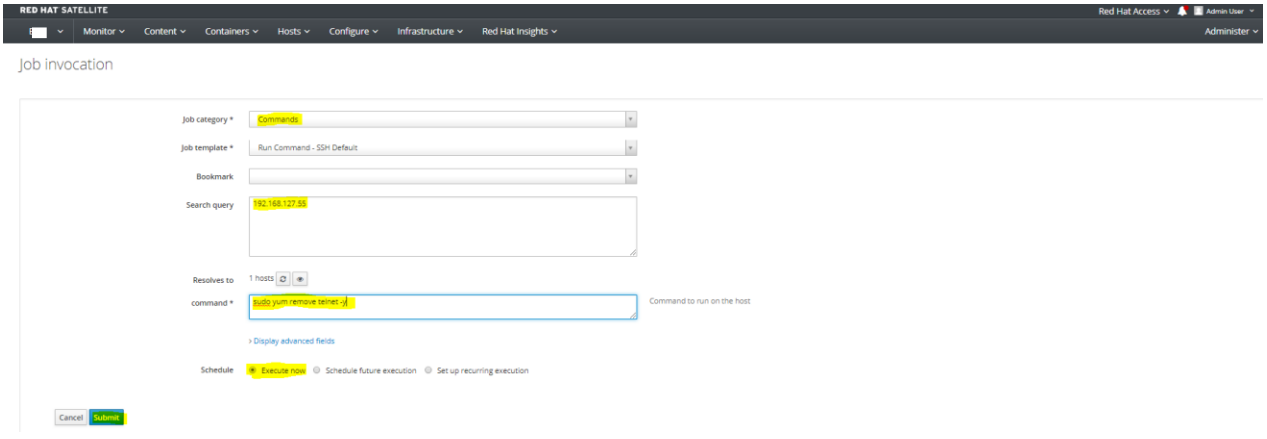


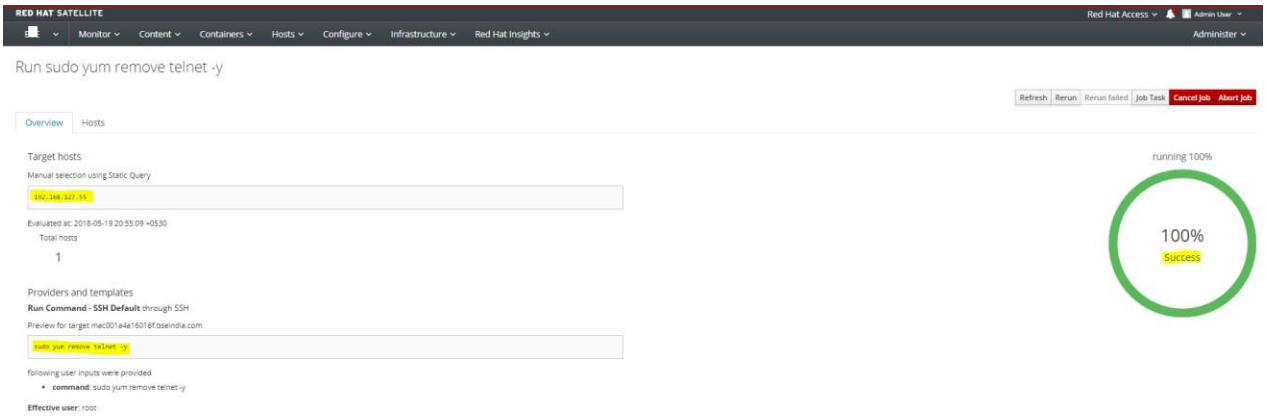
- Go to Satellite webUI >> Hosts >> All Hosts >> Select the host >> Select Action >> Schedule Remote Job >> Run commands >> in search query specify the ip address of the host otherwise you need to add host entry in satellite server /etc/hosts file.

Type the command >> select execute now >> submit







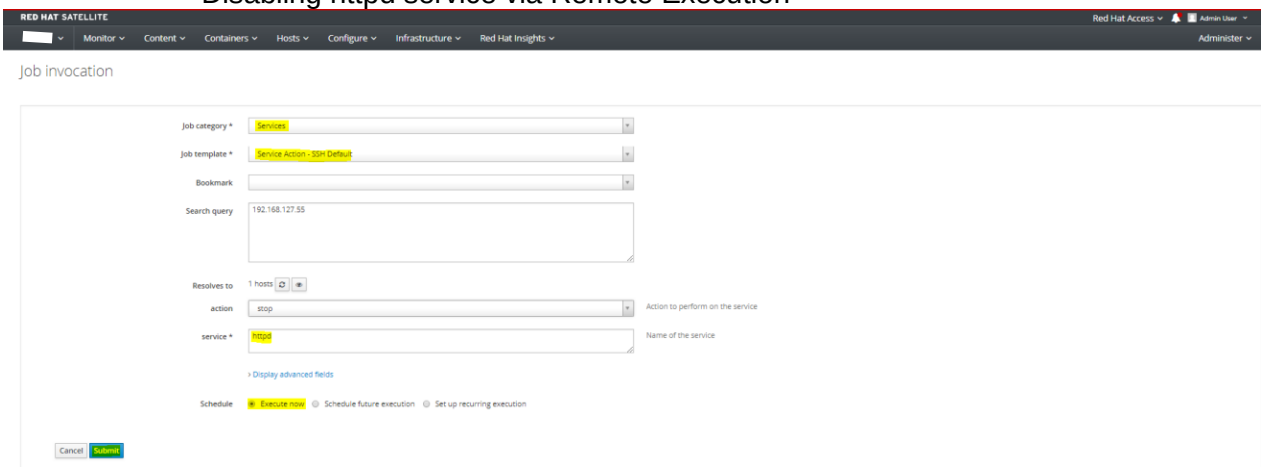


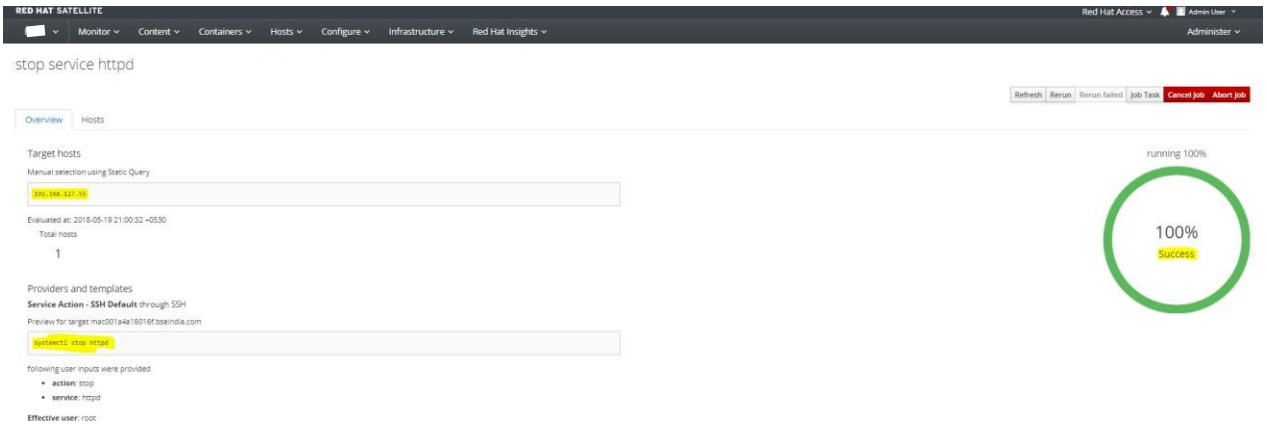
hpmanoj@mac001a4a16016f:~

```
[hpmanoj@mac001a4a16016f ~]$ sudo yum history
Loaded plugins: enabled_repos_upload, package_upload, product-id, search-disabled-repos, subscription-manager
ID     | Login user               | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------
     7 |  <hpmanoj>               | 2018-05-19 15:25 | Erase          |    1
     6 |  <hpmanoj>               | 2018-05-19 15:20 | Install        |    1
     5 |  <hpmanoj>               | 2018-05-19 15:10 | Install        |    5
     4 | System <unset>           | 2018-05-17 15:58 | Install        |   20
     3 | System <unset>           | 2018-05-17 15:56 | I, O, U        |  198 EE
     2 | System <unset>           | 2018-05-17 15:55 | Install        |   13 E<
     1 | System <unset>           | 2018-05-17 15:51 | Install        |  333 >
history list
Uploading Enabled Repositories Report
Loaded plugins: product-id, subscription-manager
[hpmanoj@mac001a4a16016f ~]$ sudo yum history info 7
Loaded plugins: enabled_repos_upload, package_upload, product-id, search-disabled-repos, subscription-manager
Transaction ID : 7
Begin time     : Sat May 19 15:25:07 2018
Begin rpmdb    : 377:15bed860ecd6fb6e4306ldf1faca16237fda81b3
End time       :           15:25:10 2018 (3 seconds)
End rpmdb      : 376:f9faca74298a0f06eaee3146b0fd83a70518f0da
User           :  <hpmanoj>
Return-Code    : Success
Command Line   : remove telnet -y
Transaction performed with:
    Installed     rpm-4.11.3-32.el7.x86_64                  @rhel-7-server-rpms
    Installed     subscription-manager-1.20.11-1.el7_5.x86_64 @rhel-7-server-rpms
    Installed     yum-3.4.3-158.el7.noarch                  @rhel-7-server-rpms
Packages Altered:
    Erase telnet-1:0.17-64.el7.x86_64 @rhel-7-server-rpms
history info
Uploading Enabled Repositories Report
Loaded plugins: product-id, subscription-manager
[hpmanoj@mac001a4a16016f ~]$ rpm -qa | grep telnet
[hpmanoj@mac001a4a16016f ~]$
```

```
[hpmanoj@mac001a4a16016f ~]$ systemctl status httpd
â httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2018-05-19 15:28:25 UTC; 1min 15s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 2910 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ââ2910 /usr/sbin/httpd -DFOREGROUND
           ââ2911 /usr/sbin/httpd -DFOREGROUND
           ââ2912 /usr/sbin/httpd -DFOREGROUND
           ââ2913 /usr/sbin/httpd -DFOREGROUND
           ââ2914 /usr/sbin/httpd -DFOREGROUND
           ââ2915 /usr/sbin/httpd -DFOREGROUND
[hpmanoj@mac001a4a16016f ~]$
```

- Disabling httpd service via Remote Execution





```
[hpmanoj@mac001a4a16016f ~]$ systemctl status httpd
â httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[hpmanoj@mac001a4a16016f ~]$ date
Sat May 19 15:31:27 UTC 2018
[hpmanoj@mac001a4a16016f ~]$
```

6. Security Compliance Management with OpenScap

Security compliance management is the ongoing process of defining security policies, auditing for compliance with those policies and resolving instances of non-compliance. Once a security policy is defined, an audit is conducted to verify compliance with the policy. Any non-compliance is managed according to the organization's configuration management policies. Security policies vary in their scope, from being host-specific to industry-wide, so there is a need for flexibility in their definition.

The Security Content Automation Protocol (SCAP) enables the definition of security configuration policies. For example, a security policy might specify that for hosts running Red Hat Enterprise Linux, login via SSH is not permitted for the root account. In Satellite 6, tools provided by the OpenSCAP project are used to implement security compliance auditing. For more information about OpenSCAP see the Red Hat Enterprise Linux 7 Security Guide. The Satellite web UI enables scheduled compliance auditing and reporting on all hosts under management by Red Hat Satellite.

The following specifications are supported by OpenSCAP:

- XCCDF: The Extensible Configuration Checklist Description Format (version 1.2)
- OVAL: Open Vulnerability and Assessment Language (version 5.11)
- Asset Identification (version 1.1)
- ARF: Asset Reporting Format (version 1.1)
- CCE: Common Configuration Enumeration (version 5.0)
- CPE: Common Platform Enumeration (version 2.3)
- CVE: Common Vulnerabilities and Exposures
- CVSS: Common Vulnerability Scoring System (version 2.0)

SCAP Content

SCAP content is a datastream format containing the configuration and security baseline against which hosts are checked. Checklists are described in the extensible checklist configuration description format (XCCDF) and vulnerabilities in the open vulnerability and assessment language (OVAL). Checklist items,

also known as rules express the desired configuration of a system item. For example, you may specify that no one can log in to a host over SSH using the root user account. Rules can be grouped into one or more profiles, allowing multiple profiles to share a rule. SCAP content consists of both rules and profiles.

XCCDF Profile

An XCCDF profile is a checklist against which a host or host group is evaluated. Profiles are generally created to verify compliance with a standard, whether that be an industry standard or a custom standard.

To list all available profiles, open the Satellite web UI, navigate to Hosts → Policies, select Edit from the drop-down list next to the policy of interest and select the SCAP Content tab. Select the SCAP Content of interest and browse the available profiles in the XCCDF Profile drop-down list.

6.1. Installing and Configuring OpenScap

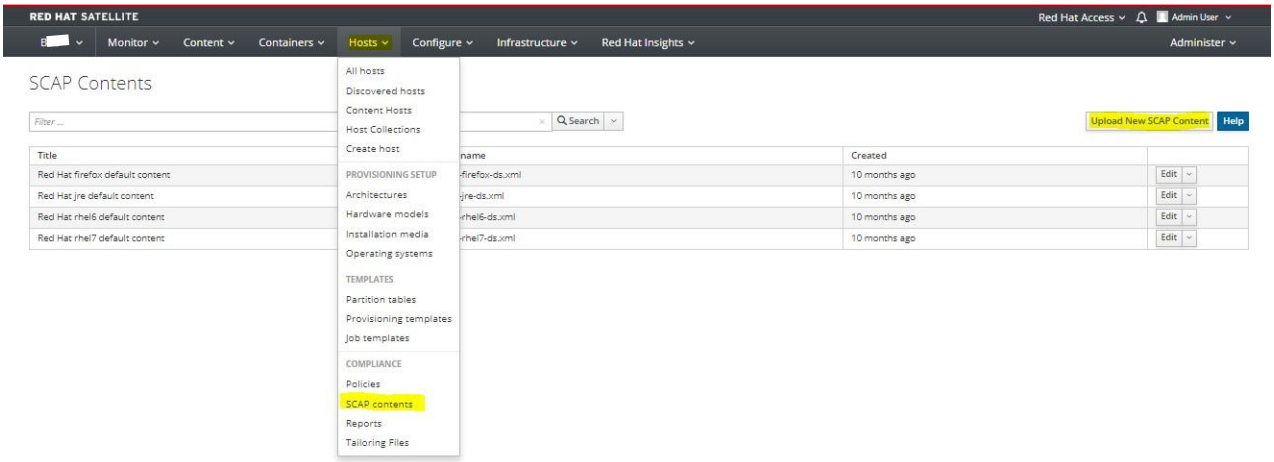
- On Satellite Server.

[root@bsesatellite6 ~]# **satellite-installer --enable-foreman-plugin-openscap** [root@bsesatellite6 ~]# **yum -y install puppet-foreman_scap_client** [root@bsesatellite6 ~]# **foreman-rake foreman_openscap:bulk_upload:default**

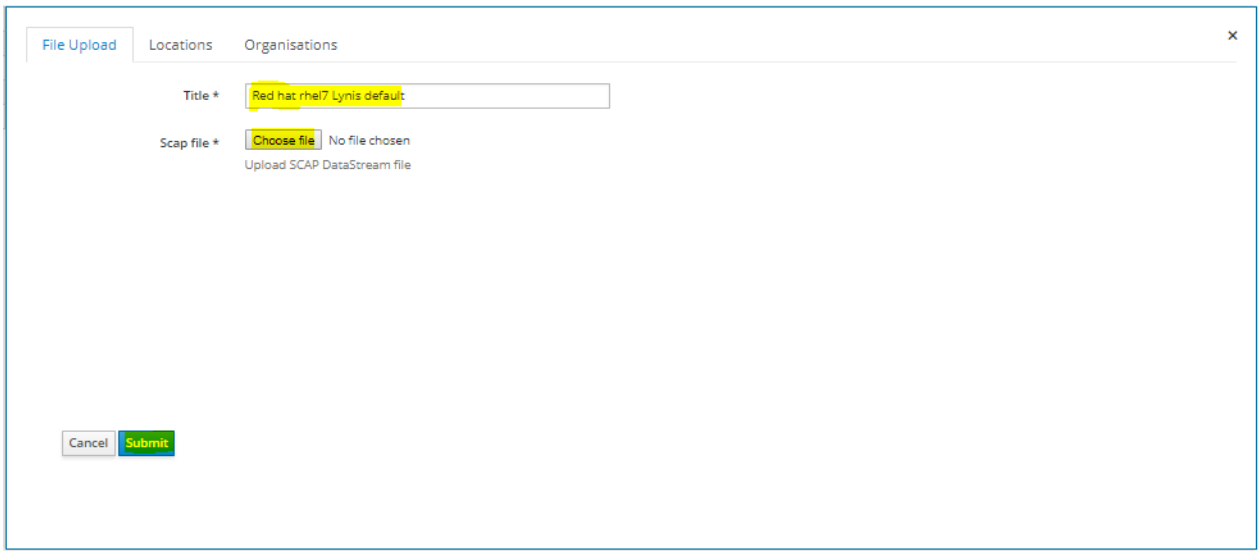
- 2. OpenSCAP Content Requirements.

On Satellite command line:
[root@ bsesatellite6 ~]# **yum -y install scap-security-guide**

- 3. On Satellite Web UI
- 4. Go to **Hosts** >> **Scap Contents** to **Upload New SCAP Content** file.

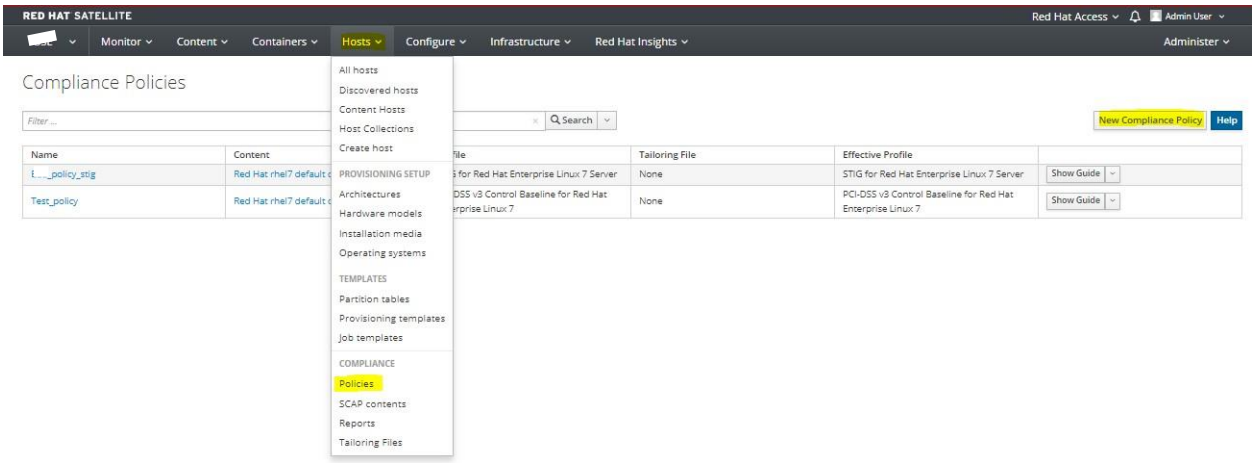


- 5. Select Upload New SCAP Content >> Enter the Title & click the choose file option to upload the Scap file.
- Assign the Location & Organisation
- Submit

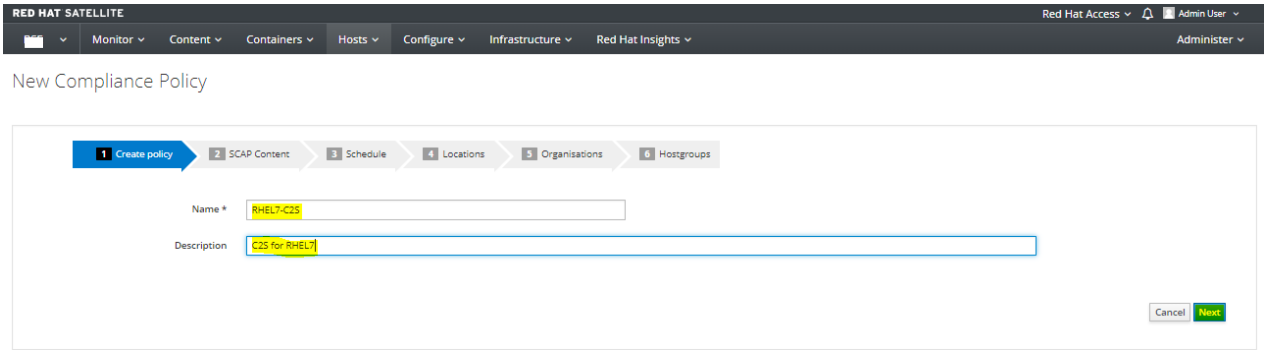


Create New Compliance Policy

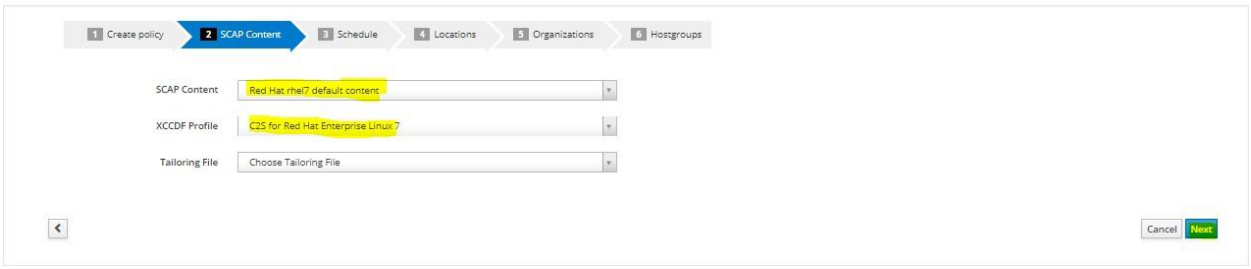
- Hosts >> Policies >> New Compliance Policy



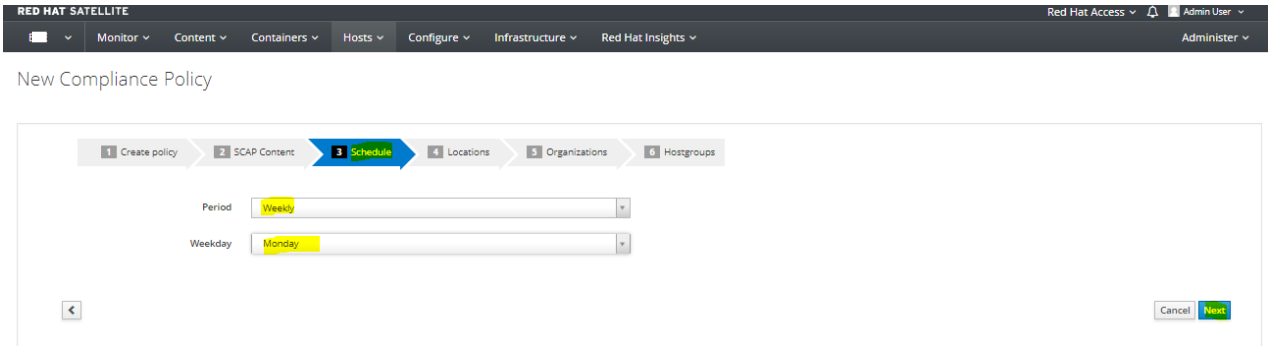
- Create New Compliance Policy



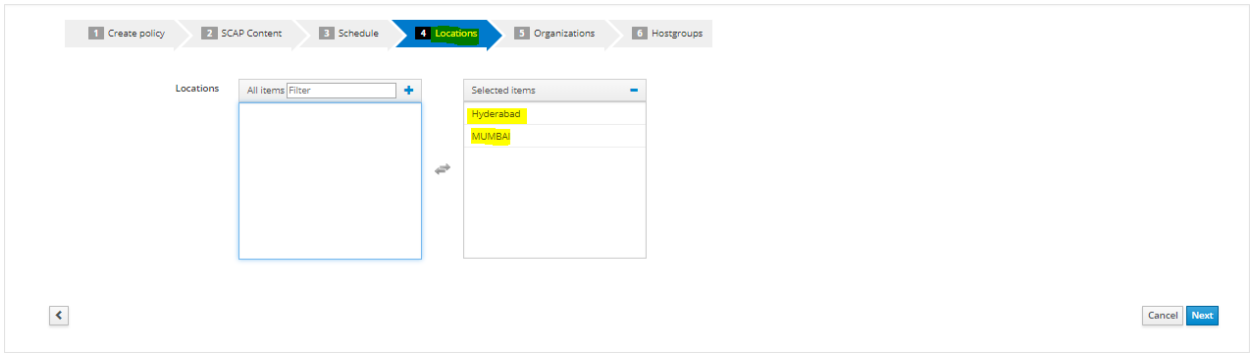
- Select the scap content



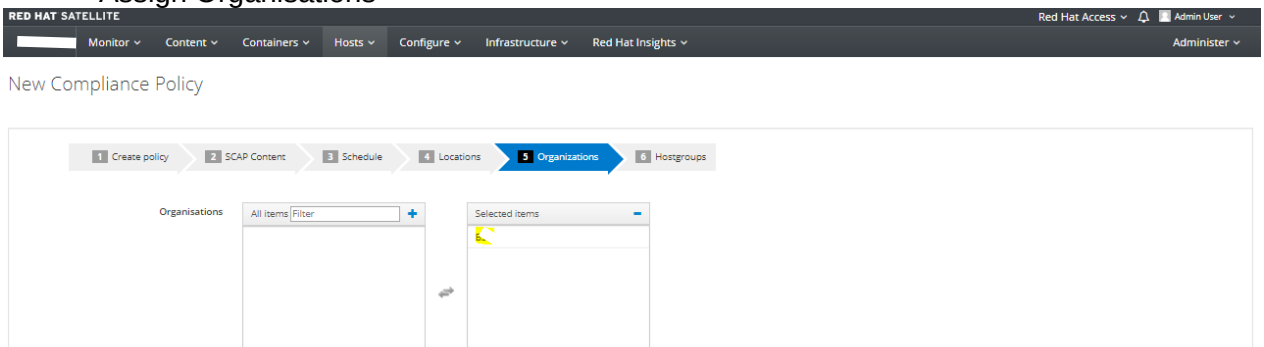
- Assign schedule time period



- Assign the Locations

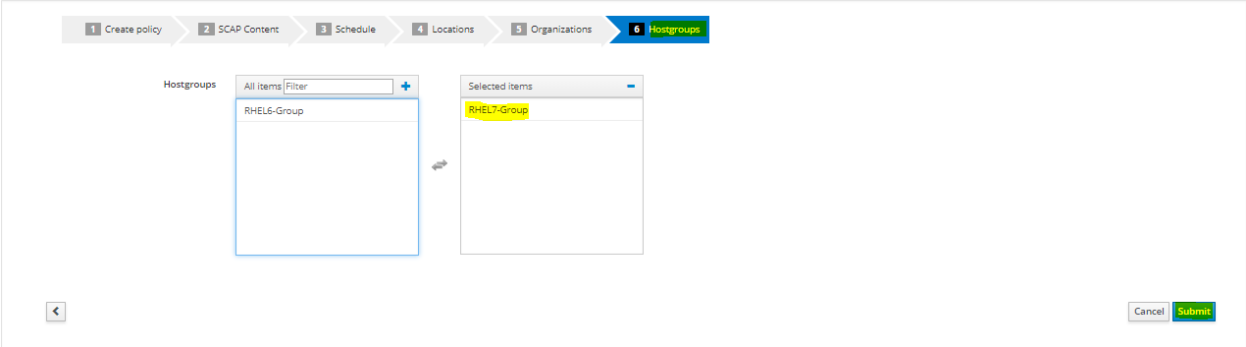


- Assign Organisations

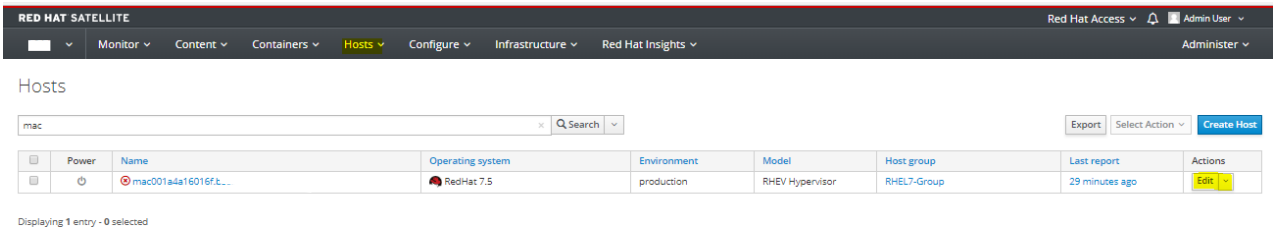


- Select the Hostgroup

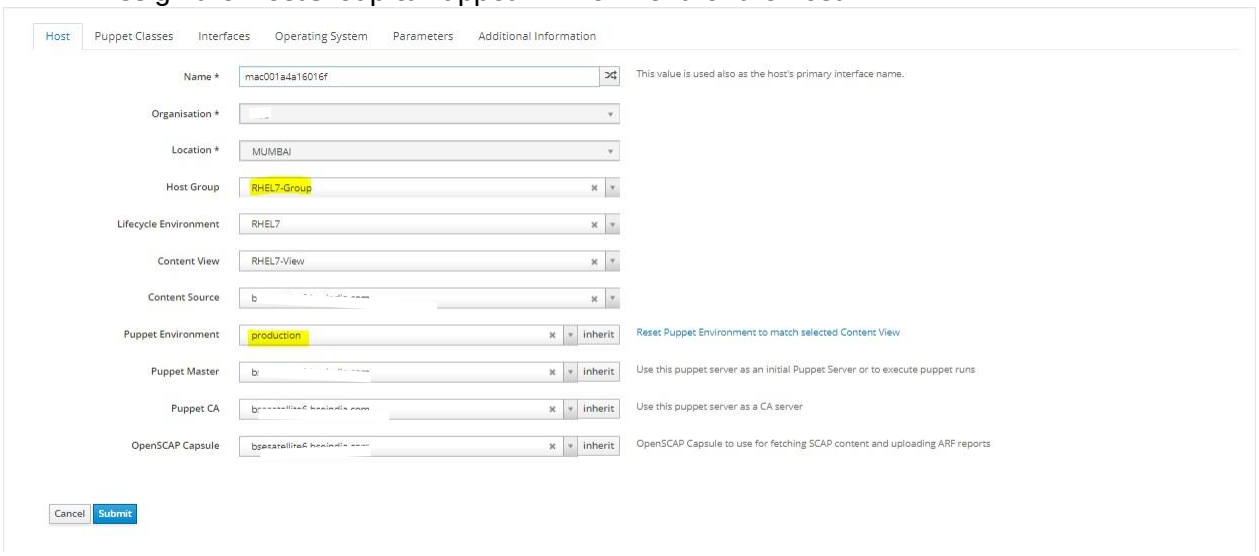
New Compliance Policy



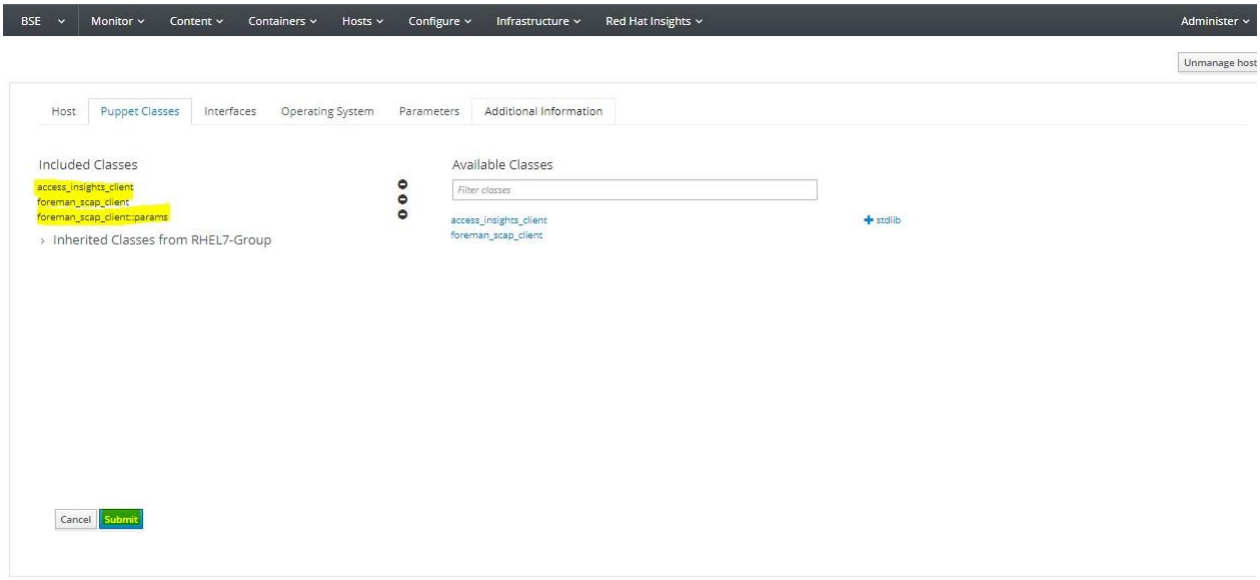
- Goto Hosts→ Select the host and Edit it.



- Assign the HostGroup & Puppet Environment for the host.



- Goto Puppet classes and add **foreman_scap_client** Classes to this host & submit it.



- **Prepare RHEL7 Client**

```
root@mac001a4a16016f:~
[root@mac001a4a16016f ~]# rpm -qa | grep puppet
[root@mac001a4a16016f ~]# yum -y install puppet
Loaded plugins: enabled_repos_upload, package_upload, product-id, search-disabled-repos, security, subscription-manager
Setting up Install Process
BSE_EPEL-RHEL-6-x86_64_file___EPEL6                                                                    | 2.5 kB     00:00
rhel-6-server-extras-rpms                                                                              | 1.8 kB     00:00
rhel-6-server-optional-rpms                                                                            | 2.0 kB     00:00
rhel-6-server-rh-common-rpms                                                                           | 2.1 kB     00:00
rhel-6-server-rhn-tools-rpms                                                                           | 1.8 kB     00:00
rhel-6-server-rpms                                                                                     | 2.0 kB     00:00
rhel-6-server-satellite-tools-6.3-rpms                                                                 | 2.1 kB     00:00
rhel-server-rhscl-6-rpms                                                                               | 2.0 kB     00:00
Resolving Dependencies
--> Running transaction check
---> Package puppet.noarch 0:3.8.6-2.el6sat will be installed
--> Processing Dependency: ruby-rgen >= 0.6.5 for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: ruby >= 1.8.7 for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: ruby >= 1.8 for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: hiera >= 1.0.0 for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: facter >= 1:1.7.0 for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: rubygem-json for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: ruby-shadow for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: ruby-augeas for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: ruby(selinux) for package: puppet-3.8.6-2.el6sat.noarch
--> Processing Dependency: /usr/bin/ruby for package: puppet-3.8.6-2.el6sat.noarch
--> Running transaction check
---> Package facter.x86_64 1:2.4.6-3.el6sat will be installed
--> Processing Dependency: ruby(abi) = 1.8 for package: 1:facter-2.4.6-3.el6sat.x86_64
---> Package hiera.noarch 0:1.3.1-2.el6sat will be installed
---> Package libselinux-ruby.x86_64 0:2.0.94-7.el6 will be installed
---> Package ruby.x86_64 0:1.8.7.374-5.el6 will be installed
---> Package ruby-augeas.x86_64 0:0.4.1-1.el6_4 will be installed
---> Package ruby-rgen.noarch 0:0.6.5-2.el6sat will be installed
---> Package ruby-shadow.x86_64 0:1.4.1-13.el6_4 will be installed
---> Package rubygem-json.x86_64 0:1.4.6-2.el6 will be installed
--> Processing Dependency: rubygems for package: rubygem-json-1.4.6-2.el6.x86_64
--> Running transaction check
---> Package ruby-libs.x86_64 0:1.8.7.374-5.el6 will be installed
--> Processing Dependency: libreadline.so.5()(64bit) for package: ruby-libs-1.8.7.374-5.el6.x86_64
---> Package rubygems.noarch 0:1.3.7-5.el6 will be installed
--> Processing Dependency: ruby-rdoc for package: rubygems-1.3.7-5.el6.noarch
--> Running transaction check
---> Package compat-readline5.x86_64 0:5.2-17.1.el6 will be installed
---> Package ruby-rdoc.x86_64 0:1.8.7.374-5.el6 will be installed
--> Processing Dependency: ruby-irb = 1.8.7.374-5.el6 for package: ruby-rdoc-1.8.7.374-5.el6.x86_64
--> Running transaction check
---> Package ruby-irb.x86_64 0:1.8.7.374-5.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

```
[root@mac001a4a16016f ~]# echo "    server = b                    >> /etc/puppet/puppet.conf
[root@mac001a4a16016f ~]# echo "    environment = production" >> /etc/puppet/puppet.conf
[root@mac001a4a16016f ~]# cat /etc/puppet/puppet.conf | grep -v ^# | grep .
[main]
    # The Puppet log directory.
    # The default value is '$vardir/log'.
    logdir = /var/log/puppet
    # Where Puppet PID files are kept.
    # The default value is '$vardir/run'.
    rundir = /var/run/puppet
    # Where SSL certificates are kept.
    # The default value is '$confdir/ssl'.
    ssldir = $vardir/ssl
[agent]
    # The file in which puppetd stores a list of the classes
    # associated with the retrieved configuratiion.  Can be loaded in
    # the separate ``puppet`` executable using the ``--loadclasses``
    # option.
    # The default value is '$confdir/classes.txt'.
    classfile = $vardir/classes.txt
    # Where puppetd caches the local configuration.  An
    # extension indicating the cache format is added automatically.
    # The default value is '$confdir/localconfig'.
    localconfig = $vardir/localconfig
    server = b                    ia.com
    environment = production
[root@mac001a4a16016f ~]#
```

Install and Configure Puppet on Client
[root@puppet-test ~]# **yum -y install**
puppet
[root@puppet-test ~]# echo " **server = example.satellite.com**" >>
/etc/puppet/puppet.conf [root@puppet-test ~]# echo " **environment = production**" >>
/etc/puppet/puppet.conf [root@puppet-test ~]# **systemctl start puppet && systemctl**
enable puppet

```
[root@mac001a4a16016f ~]# puppet agent -tv
Exiting; no certificate found and waitforcert is disabled
[root@mac001a4a16016f ~]#
```

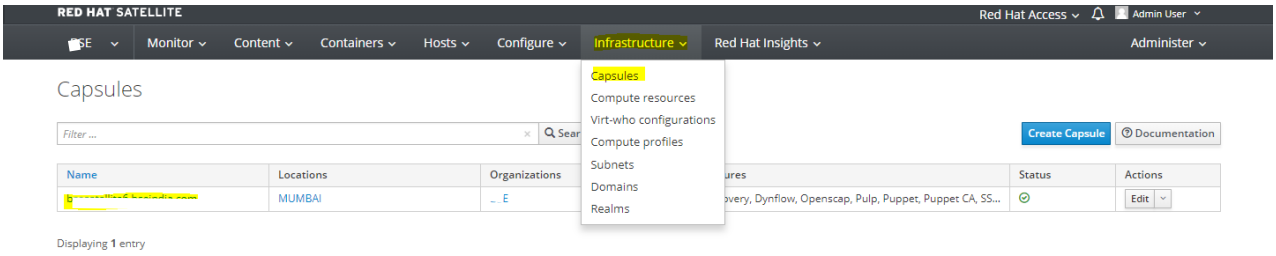
[root@puppet-test ~]# **puppet agent -t**
Exiting; no certificate found and waitforcert is disabled

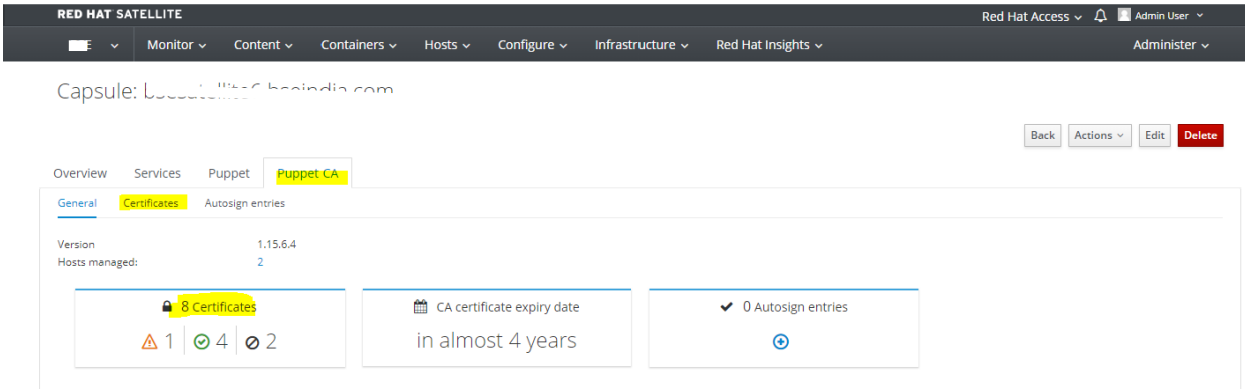
- **Go to next step to sign**

certificate Sign Certificate on

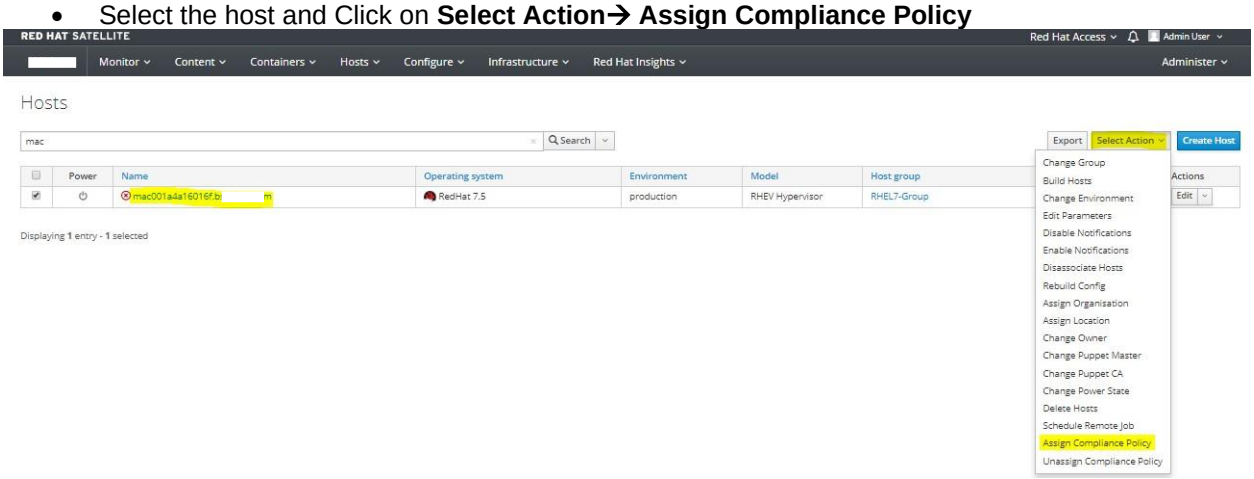
Capsule
Satellite Web UI: **Infrastructure > Capsule**
For this demo, we only have the all-in-one **Satellite/Capsule/Puppet**
Master On the Actions column, click on the available actions and
select Certificates Click Sign to **sign** the certificate
A scan will run base on the "Cron line:" setting in the earlier step. Monitor **/var/log/messages** on the client to see activities.

- Run "**puppet agent -t**" again on the client.

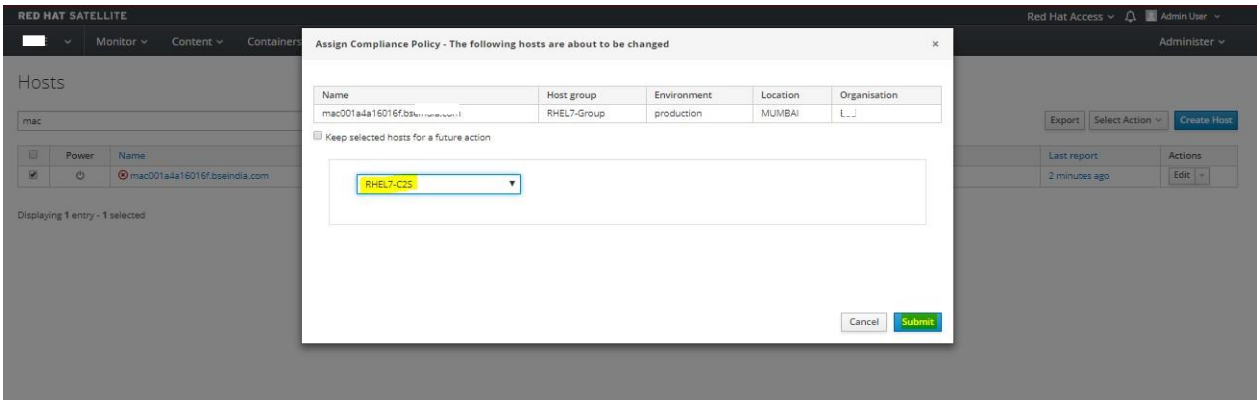




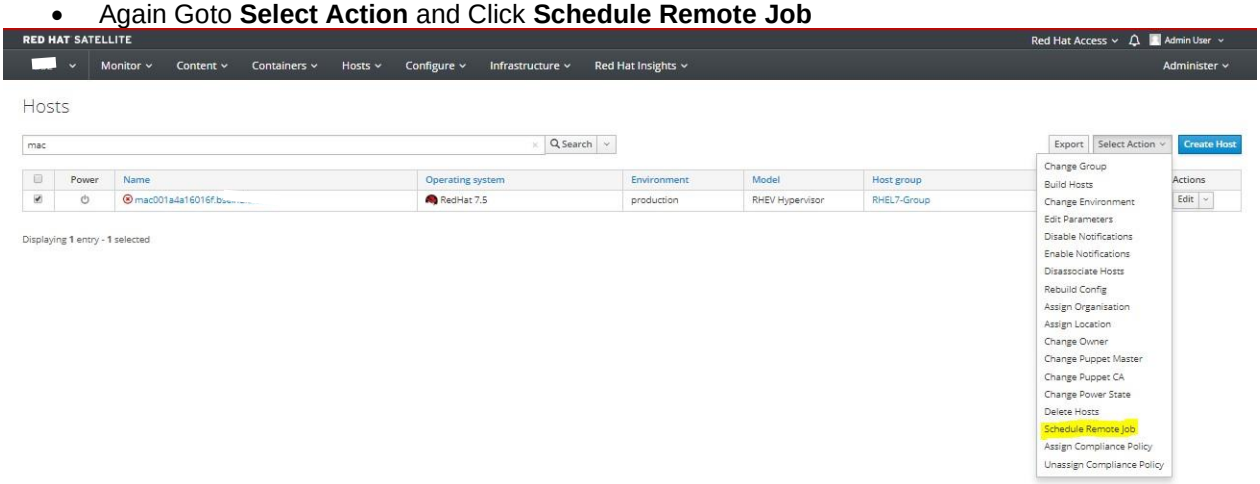
- Goto **Puppet** section to know the certificate signing steps.
- Select the host and Click on **Select Action→ Assign Compliance Policy**



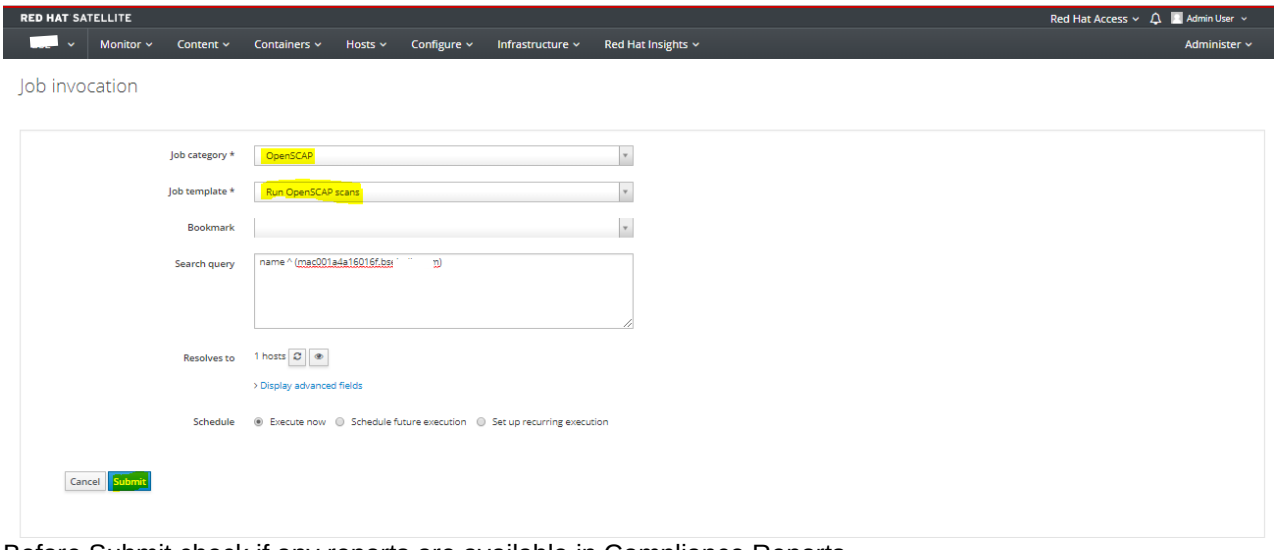
- Select the **Policy** and **Submit** it.



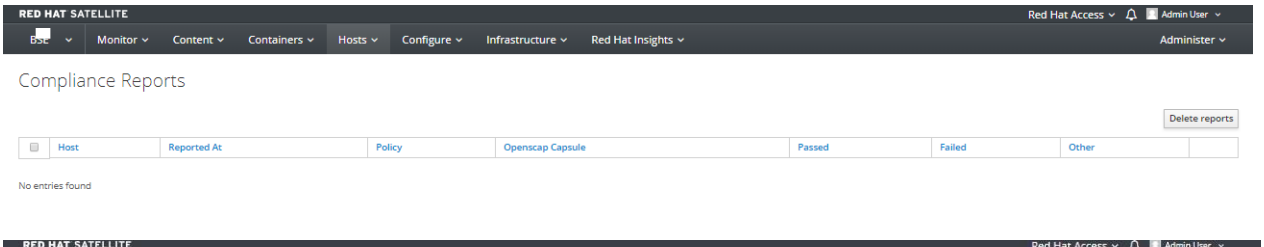
- Again Goto **Select Action** and Click **Schedule Remote Job**

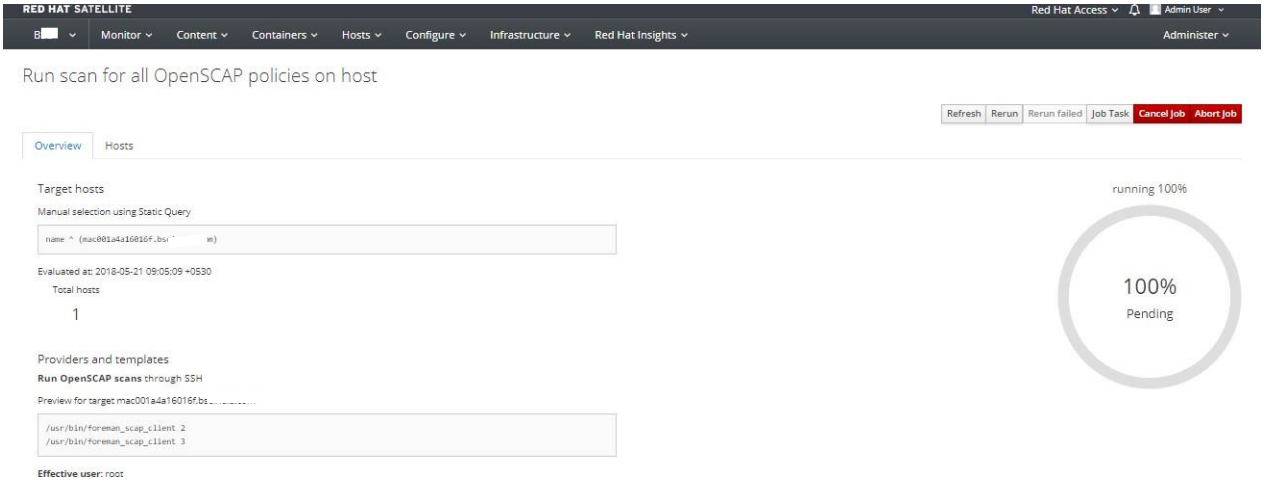


- Select **OpenSCAP** & **Submit**

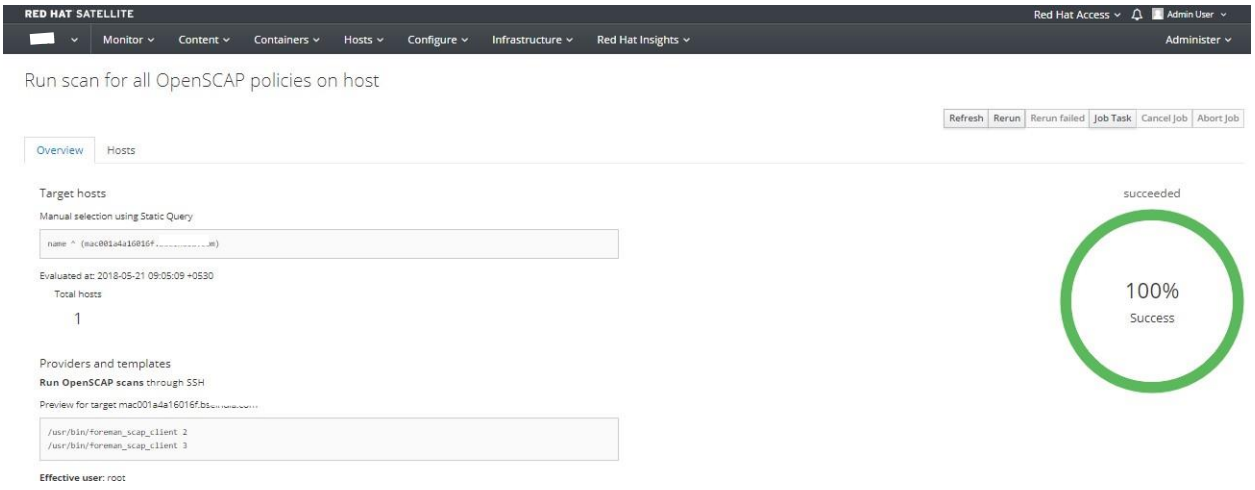


Before Submit check if any reports are available in Compliance Reports

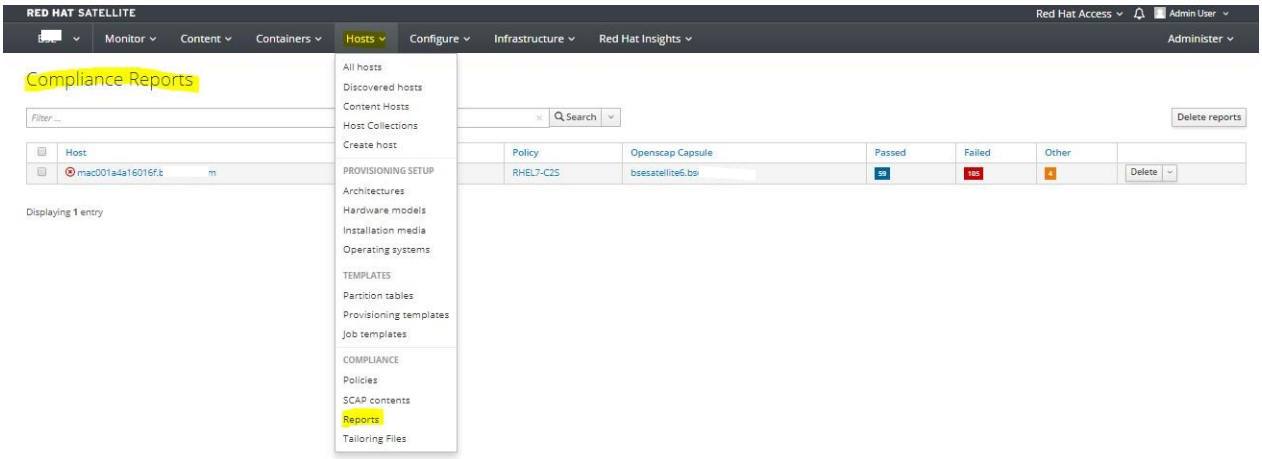




After Submit it will showing 100% Success

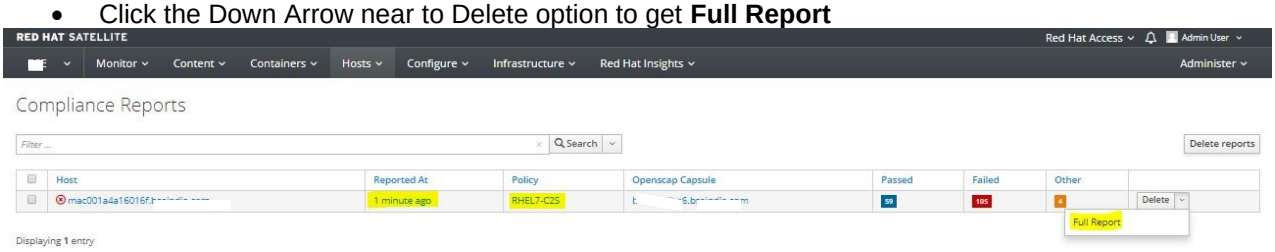


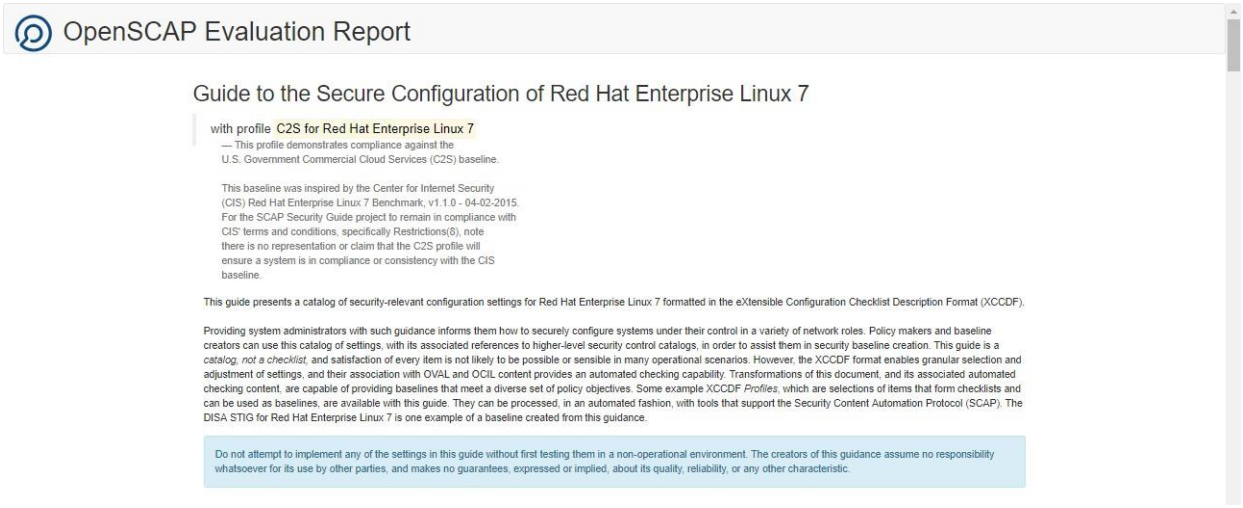
- Goto Hosts→Reports

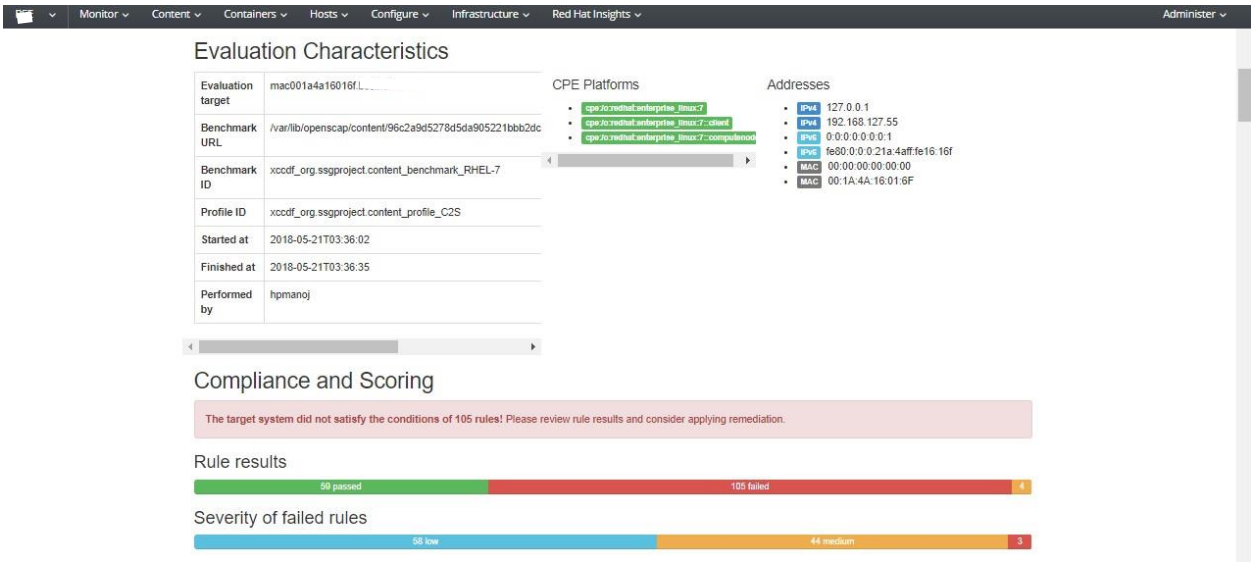


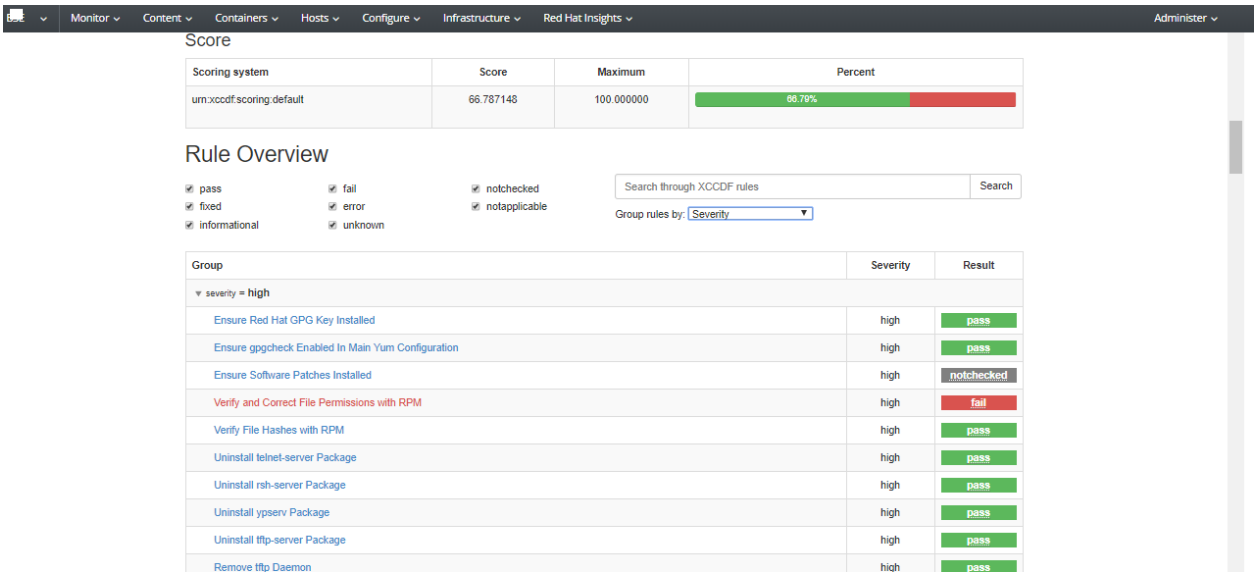
Now the Compliance report generated for this particular host.

- Click the Down Arrow near to Delete option to get **Full Report**



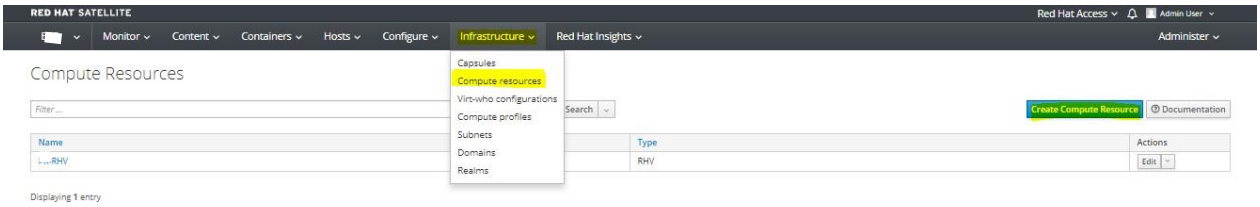






7. Compute Resources

Compute resources are hardware abstractions from virtualization and cloud providers. Satellite uses compute resources to provision virtual machines and containers. Supported private providers include **Red Hat Enterprise Virtualization**, **oVir**t, **OpenStack**, **VMware**, **Libvirt**, and **Docker**. Supported public cloud providers include **Amazon EC2, Google Compute Engine,** and **Rackspace.**



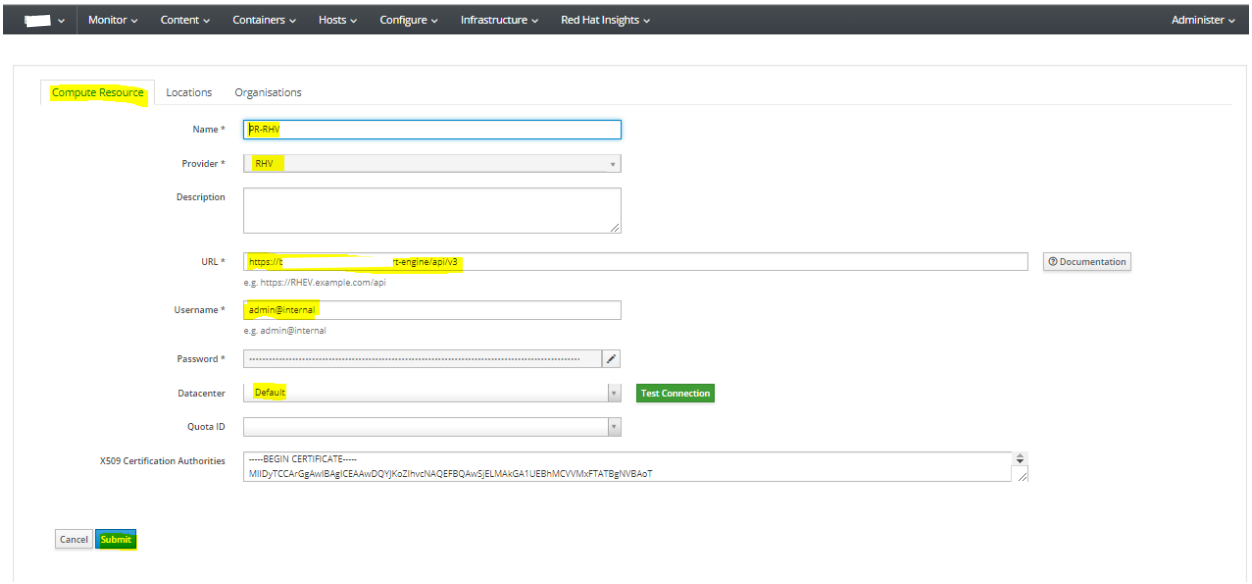
To Add a Compute Resource:

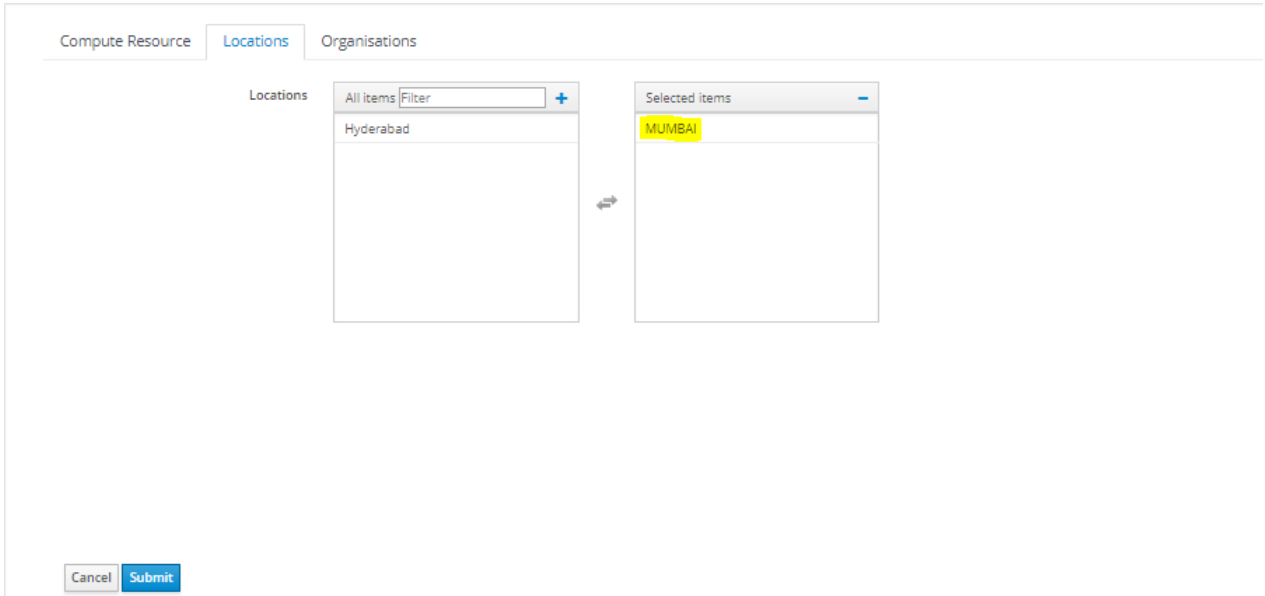
- Navigate to **Infrastructure → Compute Resources.**
- Click **New Compute Resource**.
- On the Compute Resource tab, specify the following settings:
- Specify a Name and a Provider type for the Compute Resource. Optionally, insert a Description.
- Depending on the provider type chosen, the next few fields ask for authentication and datacentre details.

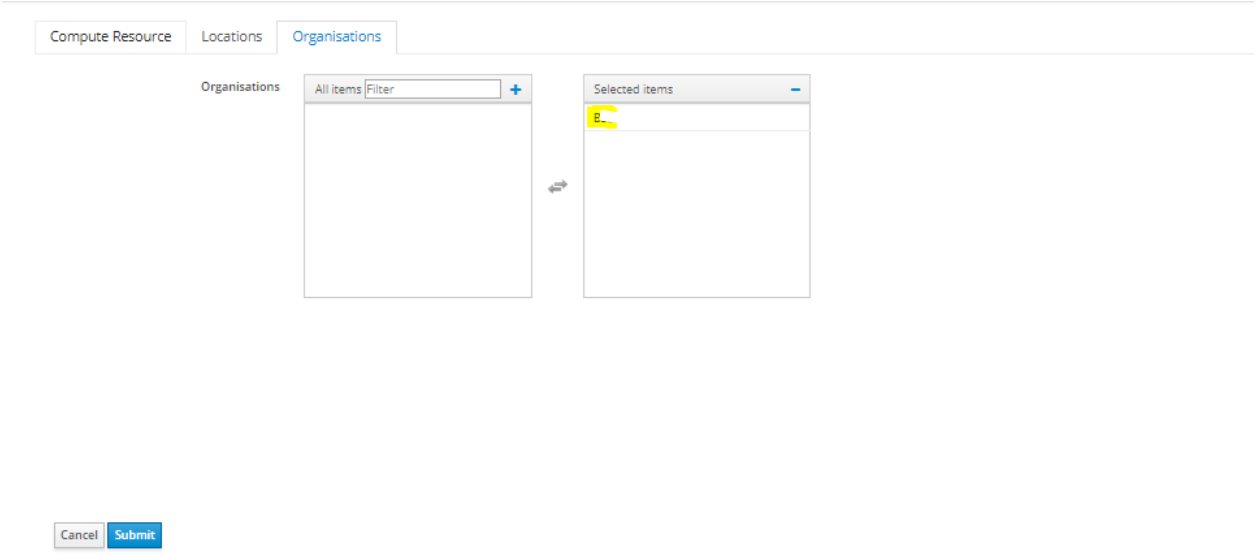
- 3. Provide RHV URL **https://[redacted]/ovirt-engine/api/v3**
- Username :

admin@internal Password :
****

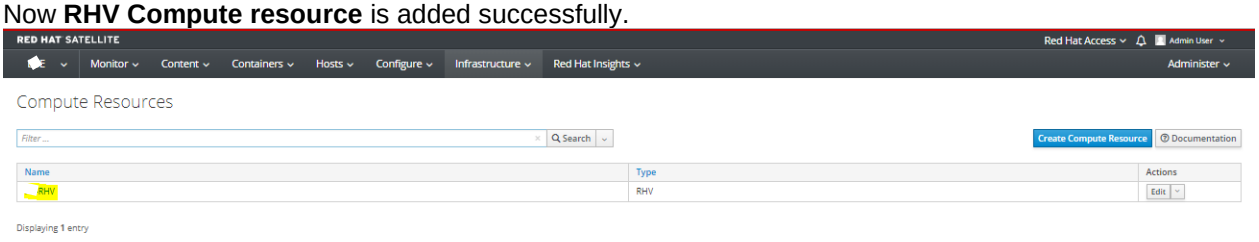
- 4. Click on **Test Connection** option to test the connectivity.
- 5. Select the **DataCenter**→ Default
- 6. Assign Location & Organisation
- Click on Submit

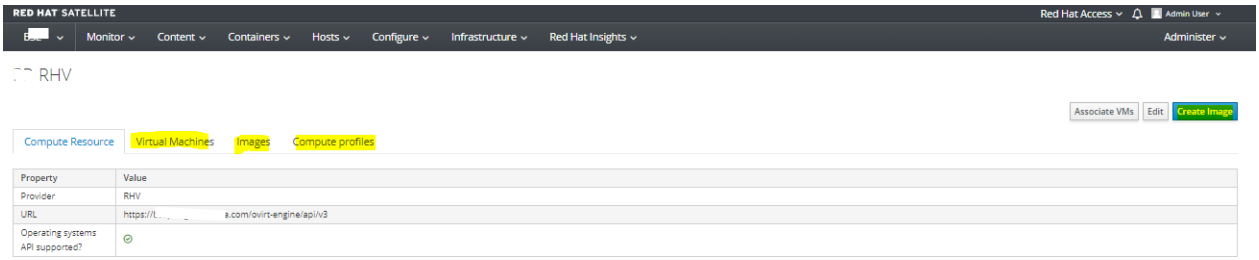




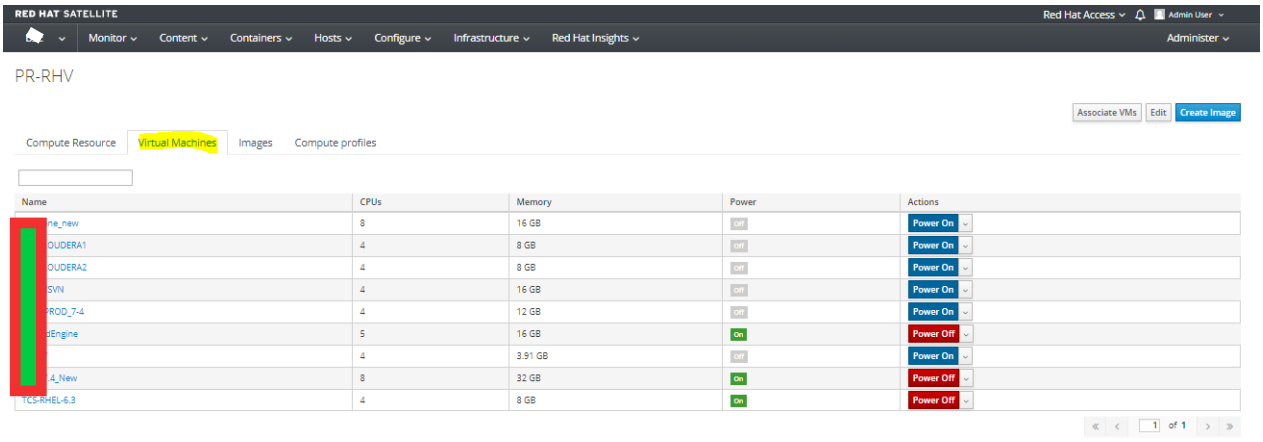


Now **RHV Compute resource** is added successfully.

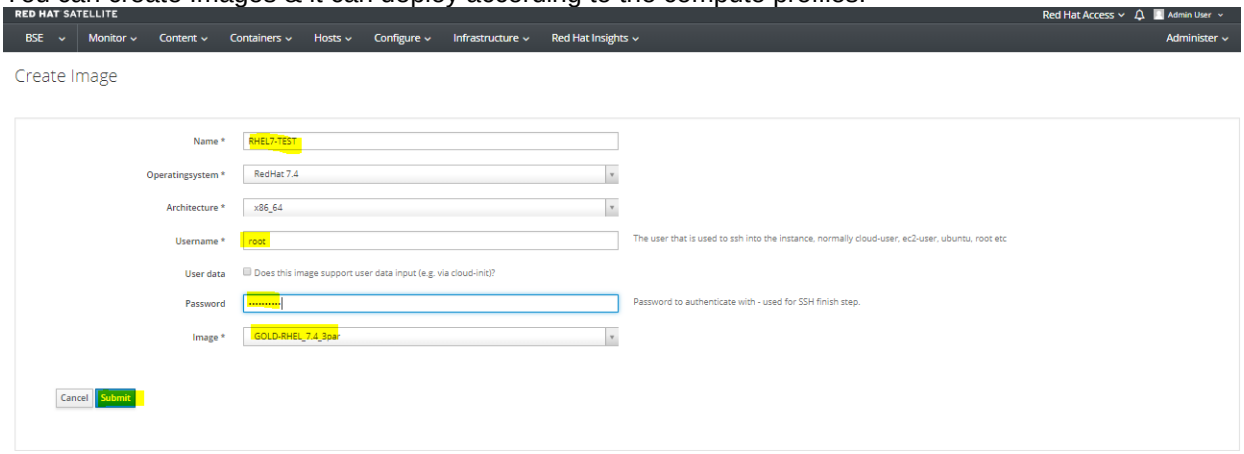


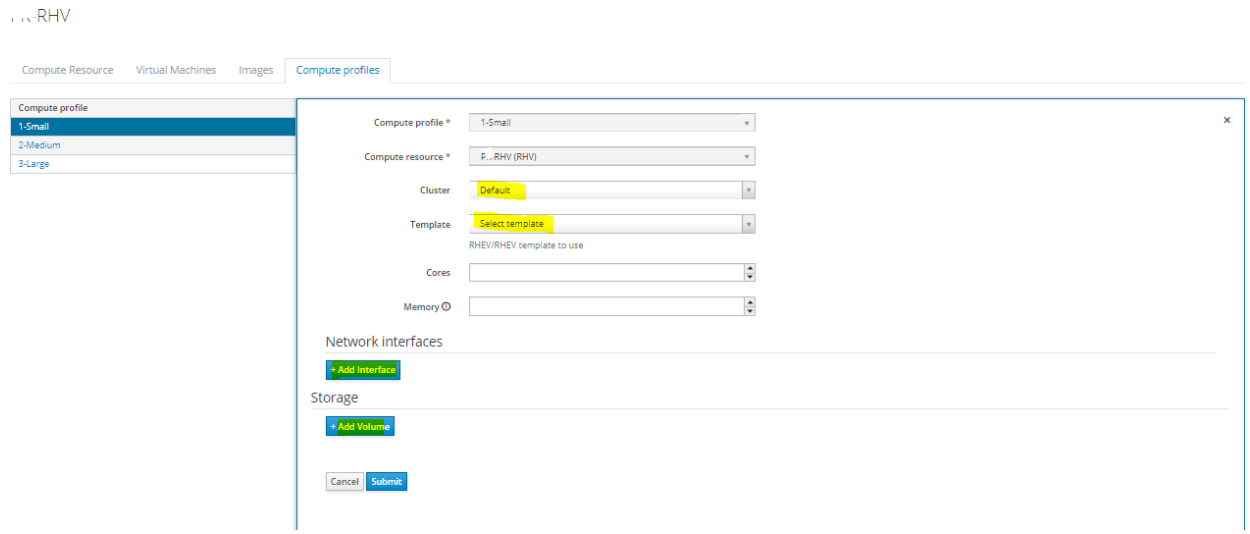


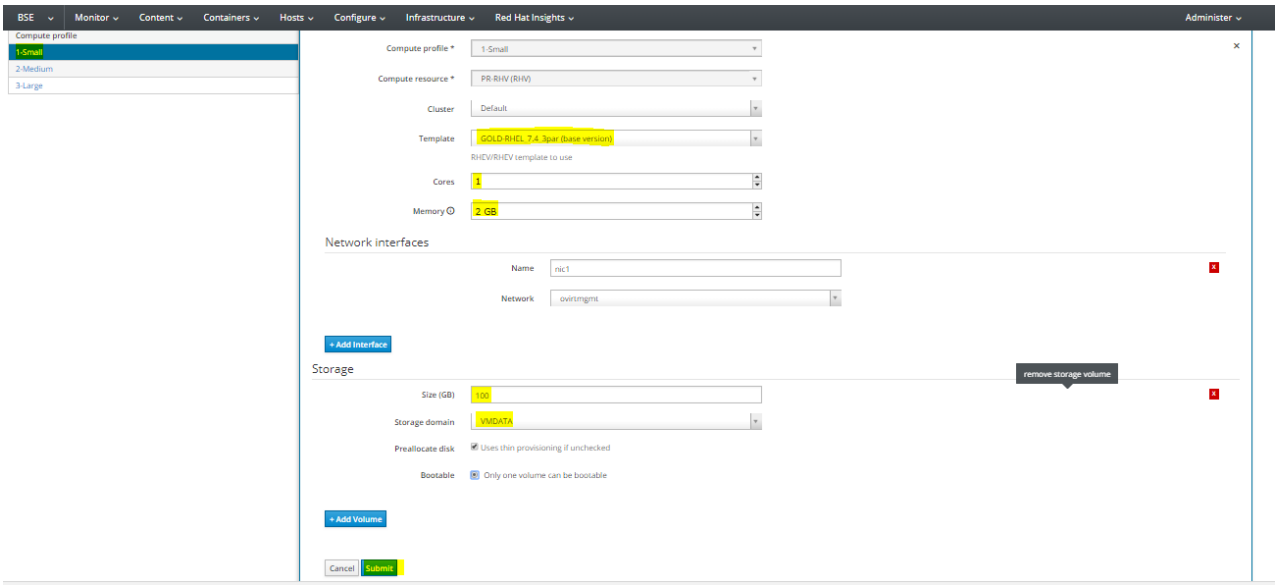
You can control the VM's from here.



You can create Images & it can deploy according to the compute profiles.







8. Updating Between Minor Versions of Satellite

Updating is the process of migrating Satellite Server, Capsule Server, and Content Hosts to a new minor version. Updates typically patch security vulnerabilities and correct minor issues discovered after code is released. Generally speaking, updates require little time and are non-disruptive to your operating environment

Prerequisites

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

Updating Satellite Server to the Next Minor Version

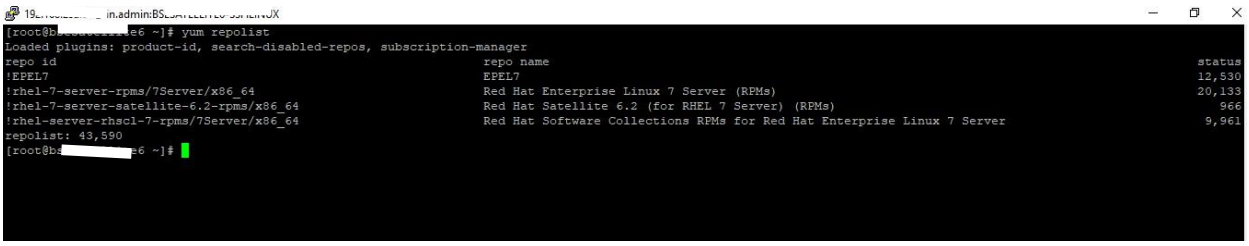
To Update Satellite Server:

1. Check that only the correct repositories are enabled:
    - List the enabled repositories:

      # subscription-manager repos --list-enabled

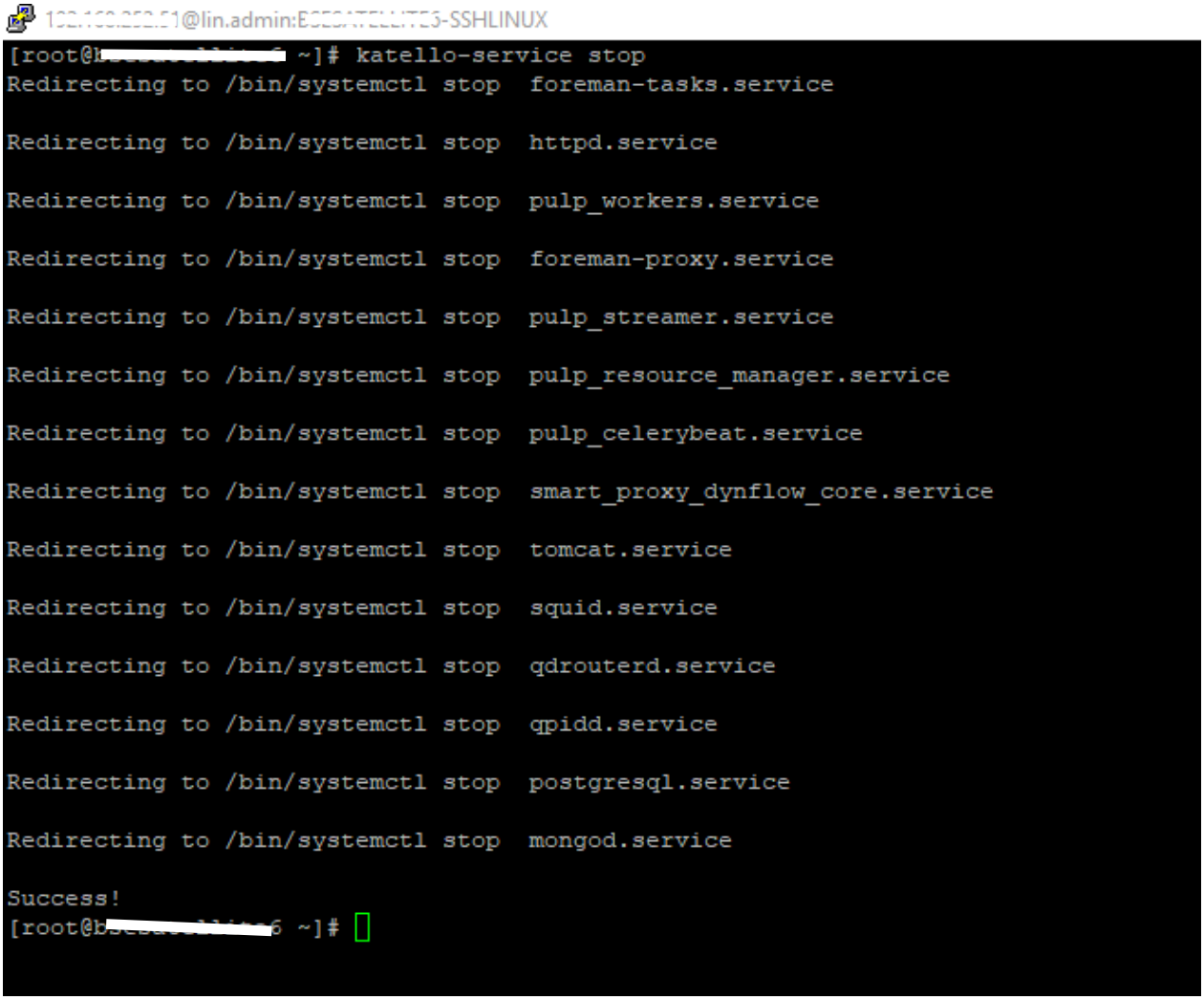
    - Ensure you only have the following repositories enabled:

      rhel-X-server-rpms
      rhel-X-server-satellite-6.2-
      rpms rhel-server-rhscl-X-
      rpms



- Stop Katello services:

```
# katello-service stop
```



- Update all packages:

```
[root@[redacted]6 ~]# yum update --disablerepo=EPEL7
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
---> Package NetworkManager.x86_64 1:1.8.0-9.el7 will be updated
---> Package NetworkManager.x86_64 1:1.10.2-13.el7 will be an update
---> Package NetworkManager-config-server.x86_64 1:1.4.0-19.el7_3 will be updated
---> Package NetworkManager-config-server.noarch 1:1.10.2-13.el7 will be an update
---> Package NetworkManager-libnm.x86_64 1:1.8.0-9.el7 will be updated
---> Package NetworkManager-libnm.x86_64 1:1.10.2-13.el7 will be an update
---> Package NetworkManager-team.x86_64 1:1.8.0-9.el7 will be updated
---> Package NetworkManager-team.x86_64 1:1.10.2-13.el7 will be an update
---> Package NetworkManager-tui.x86_64 1:1.8.0-9.el7 will be updated
---> Package NetworkManager-tui.x86_64 1:1.10.2-13.el7 will be an update
---> Package OpenIPMI-modalias.x86_64 0:2.0.19-15.el7 will be updated
---> Package OpenIPMI-modalias.x86_64 0:2.0.23-2.el7 will be an update
---> Package acl.x86_64 0:2.2.51-12.el7 will be updated
---> Package acl.x86_64 0:2.2.51-14.el7 will be an update
---> Package alsa-lib.x86_64 0:1.1.1-1.el7 will be updated
---> Package alsa-lib.x86_64 0:1.1.4.1-2.el7 will be an update
---> Package apache-commons-daemon.x86_64 0:1.0.13-6.el7 will be updated
---> Package apache-commons-daemon.x86_64 0:1.0.13-7.el7 will be an update
---> Package atk.x86_64 0:2.14.0-1.el7 will be updated
---> Package atk.x86_64 0:2.22.0-3.el7 will be an update
---> Package audit.x86_64 0:2.6.5-3.el7_3.1 will be updated
---> Package audit.x86_64 0:2.8.1-3.el7 will be an update
---> Package audit-libs.x86_64 0:2.6.5-3.el7_3.1 will be updated
---> Package audit-libs.x86_64 0:2.8.1-3.el7 will be an update
---> Package audit-libs-python.x86_64 0:2.6.5-3.el7_3.1 will be updated
---> Package audit-libs-python.x86_64 0:2.8.1-3.el7 will be an update
---> Package augeas-libs.x86_64 0:1.4.0-2.el7 will be updated
---> Package augeas-libs.x86_64 0:1.4.0-5.el7 will be an update
---> Package avahi-autoipd.x86_64 0:0.6.31-17.el7 will be updated
---> Package avahi-autoipd.x86_64 0:0.6.31-19.el7 will be an update
---> Package avahi-libs.x86_64 0:0.6.31-17.el7 will be updated
---> Package avahi-libs.x86_64 0:0.6.31-19.el7 will be an update
---> Package bash.x86_64 0:4.2.46-21.el7_3 will be updated
---> Package bash.x86_64 0:4.2.46-30.el7 will be an update
---> Package bind-libs-lite.x86_64 32:9.9.4-51.el7_4.2 will be updated
---> Package bind-libs-lite.x86_64 32:9.9.4-61.el7 will be an update
---> Package bind-license.noarch 32:9.9.4-51.el7_4.2 will be updated
---> Package bind-license.noarch 32:9.9.4-61.el7 will be an update
```

- Perform the update by running the installer script with the --upgrade option.

```
# satellite-installer --scenario satellite --upgrade
```

```
[root@[redacted]6 ~]# satellite-installer --scenario satellite --upgrade
Upgrading...
Upgrade Step: stop_services...
Redirecting to /bin/systemctl stop foreman-tasks.service

Redirecting to /bin/systemctl stop httpd.service

Redirecting to /bin/systemctl stop pulp_workers.service

Redirecting to /bin/systemctl stop foreman-proxy.service

Redirecting to /bin/systemctl stop pulp_streamer.service

Redirecting to /bin/systemctl stop pulp_resource_manager.service

Redirecting to /bin/systemctl stop pulp_celerybeat.service

Redirecting to /bin/systemctl stop smart_proxy_dynflow_core.service

Redirecting to /bin/systemctl stop tomcat.service

Redirecting to /bin/systemctl stop squid.service

Redirecting to /bin/systemctl stop qdrouterd.service

Redirecting to /bin/systemctl stop qpidd.service

Success!

Upgrade Step: start_databases...
Redirecting to /bin/systemctl start mongod.service

Redirecting to /bin/systemctl start postgresql.service

Success!

Upgrade Step: update_http_conf...

Upgrade Step: fix_pulp_httpd_conf...
Upgrade Step: upgrade_qpid_paths...
Backing up /var/lib/qpidd in case of migration failure
tar: Removing leading `/' from member names
```

Upgrade process started..

```
Moving pulp.agent.d166ac2c-4e6b-4ff8-8673-07af17a7ae5a

Moving pulp.agent.75aa2e5b-a8a8-4c77-b92f-a95b7387613f

Moving pulp.agent.9a932d51-9242-4aa1-9352-543f620517e4

Moving pulp.agent.4bea88cc-8065-433f-a69d-a726b396b064

Moving pulp.agent.0963c438-5441-45e6-92b2-834b490302db

Moving pulp.agent.0f607dec-f241-4c3c-821e-4162d31a86c0

Moving pulp.task

Moving katello_event_queue

Moving pulp.agent.541cdc29-2800-46fa-9c1d-920de4e69df6

Moving pulp.agent.1b95ab69-dbbe-4cc0-a4b8-d76de3d11e66

Moving pulp.agent.c8a3cb35-45f8-4b9f-af92-60fd47868c50
```

```
Moving pulp.agent.e8b563ad-a220-4d05-be66-6553d88cdb82

Moving pulp.agent.2dd46878-af21-4135-93d7-1ff2202ba4c8

Moving pulp.agent.8914f60b-6961-4fc8-8a76-14443079a691

Moving pulp.agent.fc0858da-1897-4abc-b807-592c38f4ba0e

Moving pulp.agent.0374026e-b1f1-4db7-b74f-9bb2c5c9e880

Moving pulp.agent.0e60ecf0-d750-4a95-96de-7ea7c731f31b

Moving pulp.agent.d61e99f8-ad7f-4ab6-b496-340f502d0a8a

Moving pulp.agent.fe5974cc-775d-4d00-9bc4-a1ab4197edc5

Moving resource_manager

Moving pulp.agent.2fcd173b-4242-4dc3-9763-d45c57c694fc

Moving pulp.agent.29c7eaf3-694b-4538-9ca4-1f713266b165
```

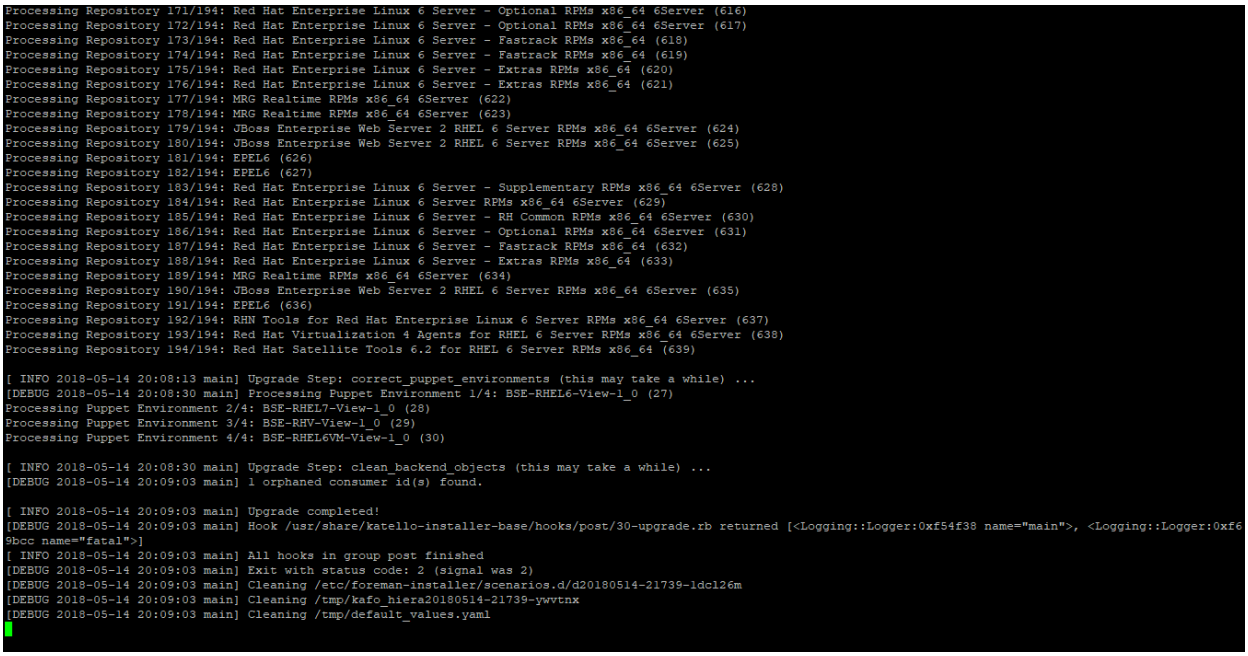
```
Upgrade Step: migrate_pulp...
22182

Attempting to connect to localhost:27017
Attempting to connect to localhost:27017
Write concern for Mongo connection: {}
Loading content types.
Loading type descriptors []
Parsing type descriptors
Validating type descriptor syntactic integrity
Validating type descriptor semantic integrity
Loading unit model: docker_blob = pulp_docker.plugins.models:Blob
Loading unit model: docker_manifest = pulp_docker.plugins.models:Manifest
Loading unit model: docker_image = pulp_docker.plugins.models:Image
Loading unit model: docker_tag = pulp_docker.plugins.models:Tag
Loading unit model: erratum = pulp_rpm.plugins.db.models:Errata
Loading unit model: distribution = pulp_rpm.plugins.db.models:Distribution
Loading unit model: package_group = pulp_rpm.plugins.db.models:PackageGroup
Loading unit model: package_category = pulp_rpm.plugins.db.models:PackageCategory
Loading unit model: iso = pulp_rpm.plugins.db.models:ISO
Loading unit model: package_environment = pulp_rpm.plugins.db.models:PackageEnvironment
Loading unit model: drpm = pulp_rpm.plugins.db.models:DRPM
Loading unit model: srpm = pulp_rpm.plugins.db.models:SRPM
Loading unit model: rpm = pulp_rpm.plugins.db.models:RPM
Loading unit model: yum_repo_metadata_file = pulp_rpm.plugins.db.models:YumMetadataFile
Loading unit model: puppet_module = pulp_puppet.plugins.db.models:Module
Loading auxiliary model: erratum_pkglists = pulp_rpm.plugins.db.models:ErratumPkglist
Updating the database with types []
Found the following type definitions that were not present in the update collection [puppet_module, docker_tag, docker_manifest, docker_blob, erratum, yum_repo_metadata
_file, package_group, package_category, iso, package_environment, drpm, srpm, rpm, distribution, docker_image]
Updating the database with types [puppet_module, drpm, erratum, docker_blob, docker_manifest, yum_repo_metadata_file, package_group, package_category, iso, package_envi
ronment, docker_tag, distribution, rpm, srpm, docker_image]
Content types loaded.
Ensuring the admin role and user are in place.
Admin role and user are in place.
Beginning database migrations.
Migration package pulp.server.db.migrations is up to date at version 24
Migration package pulp_docker.plugins.migrations is up to date at version 2
```

```
Upgrade Step: start_httpd...
Redirecting to /bin/systemctl start httpd.service

Success!

Upgrade Step: start_qpidd...
Redirecting to /bin/systemctl start qpidd.service

Redirecting to /bin/systemctl start qdrouterd.service

Success!

Upgrade Step: start_pulp...
Redirecting to /bin/systemctl start pulp_celerybeat.service

Redirecting to /bin/systemctl start pulp_resource_manager.service

Redirecting to /bin/systemctl start pulp_workers.service

Success!

Upgrade Step: migrate_candlepin...
Migrating Candlepin database
--driver=org.postgresql.Driver --classpath=/usr/share/java/postgresql-jdbc.jar:/var/lib/tomcat/webapps/candlepin/WEB-INF/classes/ --changeLogFile=db/changelog/changelog
-update.xml --url=jdbc:postgresql:candlepin --username=candlepin --password=TcbZvfB4a2Hren9ognGKA3VwURAXSzwJ --logLevel=severe
Liquibase Update Successful

Upgrade Step: start_tomcat...
Redirecting to /bin/systemctl start tomcat.service

Success!

Upgrade Step: migrate_foreman...
API controllers newer than Apipie cache! Run apipie:cache rake task to regenerate cache.
true

API controllers newer than Apipie cache! Run apipie:cache rake task to regenerate cache.
== 20161003204325 AddUserToKatelloSubscriptionFacets: migrating ==============
-- add_column(:katello_subscription_facets, :user_id, :integer)
   -> 0.1774s
-- add_index(:katello_subscription_facets, [:user_id], {:unique=>true})
   -> 0.3469s
-- add_foreign_key("katello_subscription_facets", "users", {:column=>"user_id"})
```

```
   -> 0.0017s
-- add_index(:katello_installed_packages, [:name, :nvra])
   -> 0.0879s
-- add_foreign_key("katello_host_installed_packages", "hosts", {:name=>"katello_host_installed_packages_host_id", :column=>"host_id"})
   -> 0.0113s
-- add_foreign_key("katello_host_installed_packages", "katello_installed_packages", {:name=>"katello_host_installed_packages_installed_package_id", :column=>"installed_
package_id"})
   -> 0.0170s
-- add_index(:katello_installed_packages, [:nvra], {:unique=>true})
   -> 0.1258s
-- add_index(:katello_host_installed_packages, [:host_id, :installed_package_id], {:unique=>true, :name=>:katello_host_installed_packages_h_id_ip_id})
   -> 0.1030s
== 20171114150937 CleanupInstalledPackages: migrated (1.5783s) ===============


API controllers newer than Apipie cache! Run apipie:cache rake task to regenerate cache.
false

Upgrade Step: Running installer...
Installing             Done                                       [100%] [.......................................................................................]
  The full log is at /var/log/foreman-installer/satellite.log
Upgrade Step: restart_services...
Redirecting to /bin/systemctl stop foreman-tasks.service

Redirecting to /bin/systemctl stop httpd.service

Redirecting to /bin/systemctl stop pulp_workers.service

Redirecting to /bin/systemctl stop foreman-proxy.service

Redirecting to /bin/systemctl stop pulp_streamer.service

Redirecting to /bin/systemctl stop pulp_resource_manager.service

Redirecting to /bin/systemctl stop pulp_celerybeat.service

Redirecting to /bin/systemctl stop smart_proxy_dynflow_core.service

Redirecting to /bin/systemctl stop tomcat.service

Redirecting to /bin/systemctl stop squid.service

Redirecting to /bin/systemctl stop qdrouterd.service
```

```
Processing Repository 170/194: Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 6Server (615)
Processing Repository 171/194: Red Hat Enterprise Linux 6 Server - Optional RPMs x86_64 6Server (616)
Processing Repository 172/194: Red Hat Enterprise Linux 6 Server - Optional RPMs x86_64 6Server (617)
Processing Repository 173/194: Red Hat Enterprise Linux 6 Server - Fastrack RPMs x86_64 (618)
Processing Repository 174/194: Red Hat Enterprise Linux 6 Server - Fastrack RPMs x86_64 (619)
Processing Repository 175/194: Red Hat Enterprise Linux 6 Server - Extras RPMs x86_64 (620)
Processing Repository 176/194: Red Hat Enterprise Linux 6 Server - Extras RPMs x86_64 (621)
Processing Repository 177/194: MRG Realtime RPMs x86_64 6Server (622)
Processing Repository 178/194: MRG Realtime RPMs x86_64 6Server (623)
Processing Repository 179/194: JBoss Enterprise Web Server 2 RHEL 6 Server RPMs x86_64 6Server (624)
Processing Repository 180/194: JBoss Enterprise Web Server 2 RHEL 6 Server RPMs x86_64 6Server (625)
Processing Repository 181/194: EPEL6 (626)
Processing Repository 182/194: EPEL6 (627)
Processing Repository 183/194: Red Hat Enterprise Linux 6 Server - Supplementary RPMs x86_64 6Server (628)
Processing Repository 184/194: Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server (629)
Processing Repository 185/194: Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 6Server (630)
Processing Repository 186/194: Red Hat Enterprise Linux 6 Server - Optional RPMs x86_64 6Server (631)
Processing Repository 187/194: Red Hat Enterprise Linux 6 Server - Fastrack RPMs x86_64 (632)
Processing Repository 188/194: Red Hat Enterprise Linux 6 Server - Extras RPMs x86_64 (633)
Processing Repository 189/194: MRG Realtime RPMs x86_64 6Server (634)
Processing Repository 190/194: JBoss Enterprise Web Server 2 RHEL 6 Server RPMs x86_64 6Server (635)
Processing Repository 191/194: EPEL6 (636)
Processing Repository 192/194: RHN Tools for Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server (637)
Processing Repository 193/194: Red Hat Virtualization 4 Agents for RHEL 6 Server RPMs x86_64 6Server (638)
Processing Repository 194/194: Red Hat Satellite Tools 6.2 for RHEL 6 Server RPMs x86_64 (639)

Upgrade Step: correct_puppet_environments (this may take a while) ...
Processing Puppet Environment 1/4: BSE-RHEL6-View-1_0 (27)
Processing Puppet Environment 2/4: BSE-RHEL7-View-1_0 (28)
Processing Puppet Environment 3/4: BSE-RHV-View-1_0 (29)
Processing Puppet Environment 4/4: BSE-RHEL6VM-View-1_0 (30)

Upgrade Step: clean_backend_objects (this may take a while) ...
1 orphaned consumer id(s) found.

Upgrade completed!
[root@[redacted]6 ~]#
```

- Upgrade completed successfully. Review the **/var/log/foreman-installer/satellite.log #tailf /var/log/foreman-installer/satellite.log**



9. Upgrading Satellite Server Version to Satellite 6.3

Use this procedure for a Satellite Server connected to the Red Hat Content Delivery Network.

Prerequisites

- Create a backup.
    - On a virtual machine, take a snapshot.
    - On a physical machine, create a backup.
- Back up the DNS and DHCP configuration files **/etc/zones.conf** and **/etc/dhcp/dhcpd.conf** as the installer only supports one domain or subnet, and therefore restoring changes from these backups might be required.
- 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.

- In the Satellite web UI, navigate to **Content** > **Red Hat Subscriptions**, and then click **Manage Manifest**. In the Subscription Manifest pane, click the **Actions** tab, and then click **Refresh Manifest** to download the latest copy of the Subscription Manifest.

- Configure the repositories in the Satellite web UI:

    - In the Satellite web UI, navigate to **Content** > **Red Hat Repositories** and select the **RPMs** tab.

    - From the **PRODUCT** list, find and expand **Red Hat Enterprise Linux Server**.

    - From the **REPOSITORY SET** list, find and expand **Red Hat Satellite Tools 6.3 (for RHEL7 Server) (RPMs)**.

    - Select **Red Hat Satellite Tools 6.3 for RHEL 7 Server RPMs x86_64**.

- Synchronize the newly enabled repositories:

    - In the Satellite web UI, navigate to **Content** > **Sync Status**.

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

    - Select the repositories for 6.3.

    - Click **Synchronize Now**.

        - If you get an error when trying to update a repository, ensure 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. Refresh the manifest and if the problem persists, raise a support request.

- Update any pre-existing Content Views that utilize 6.2 version repositories with the new version for 6.3. Publish and promote updated versions of any Content Views that now have the new 6.3 version repositories.

- Refresh your subscription:

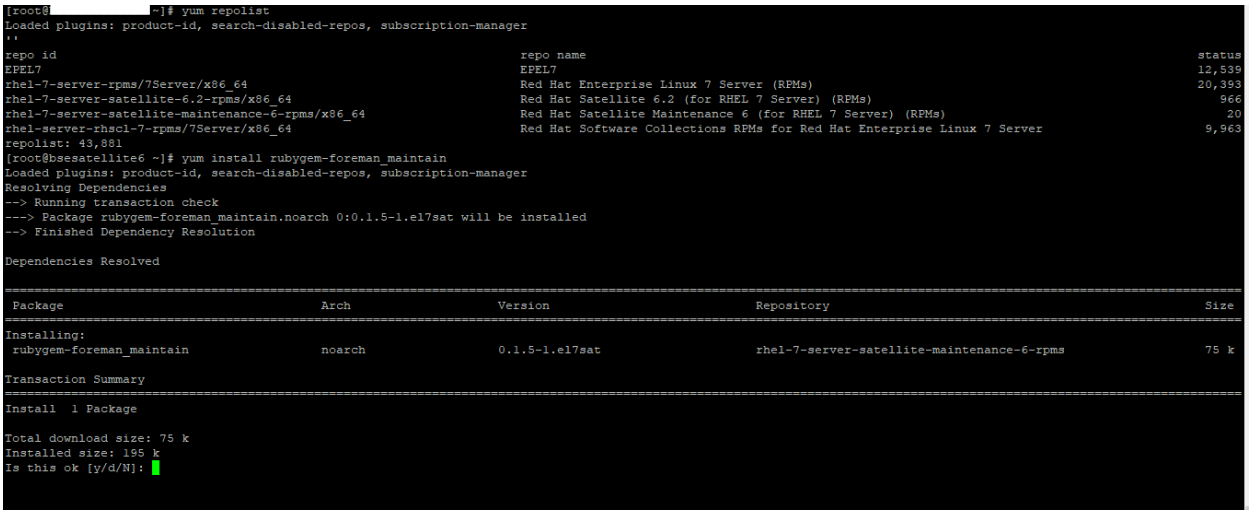
```
# subscription-manager refresh
```

- Enable the Satellite Maintenance repository:

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

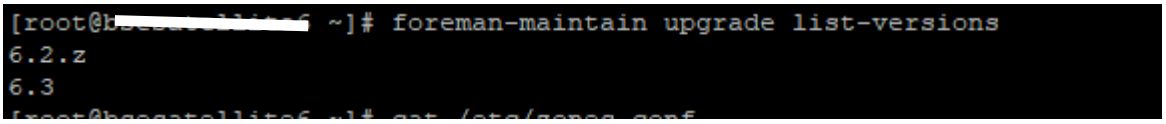
- Install foreman-maintain:

```
# yum install rubygem-foreman_maintain
```



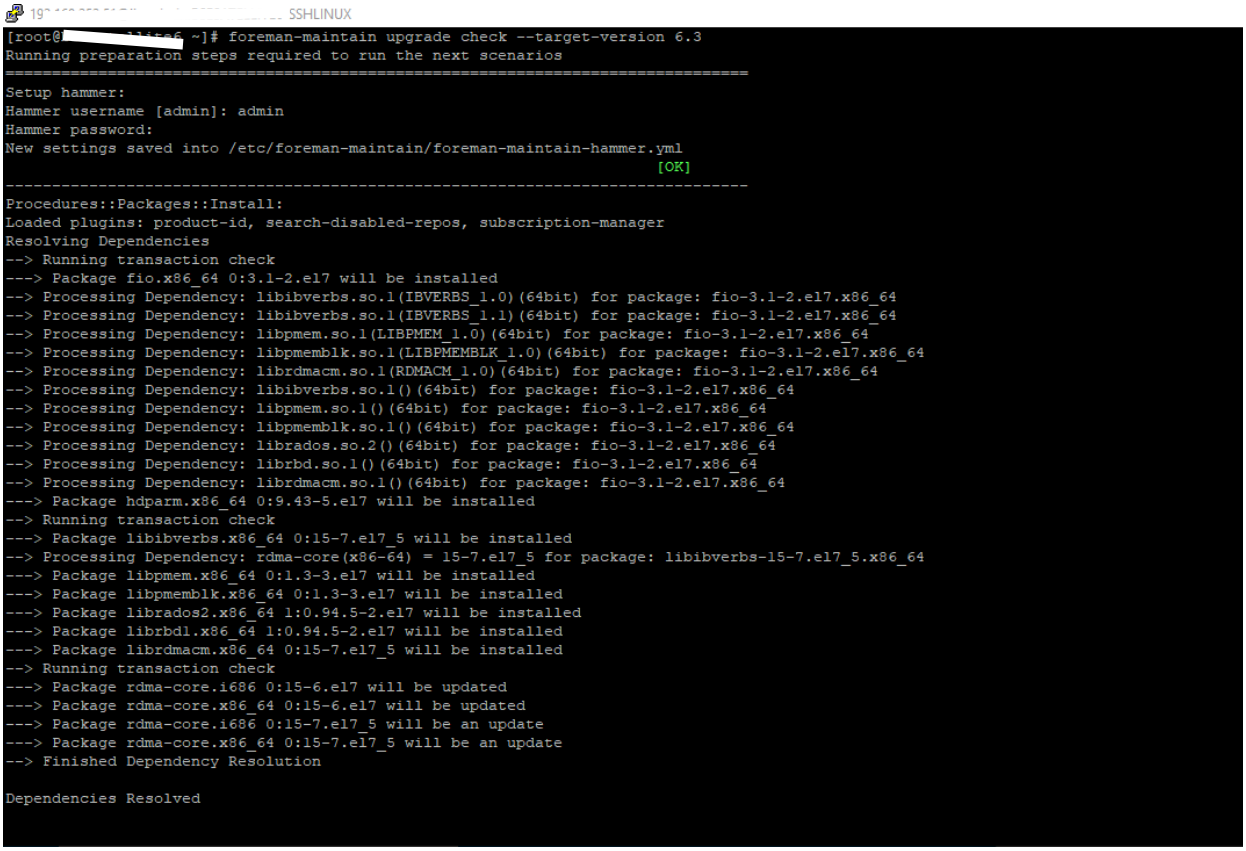
- Check the available versions to confirm 6.3 is listed:

```
# foreman-maintain upgrade list-versions
```



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

```
# foreman-maintain upgrade check --target-version 6.3
```



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

```
[root@[redacted] ~]# foreman-maintain upgrade check --target-version 6.3
Running Checks before upgrading to Satellite 6.3
================================================================================
Check for paused tasks:                                                  [OK]
--------------------------------------------------------------------------------
Check whether all services are running using hammer ping:                [OK]
--------------------------------------------------------------------------------
Check to validate candlepin database:                                    [OK]
--------------------------------------------------------------------------------
Check for running tasks:                                                 [OK]
--------------------------------------------------------------------------------
Check for old tasks in paused/stopped state:                             [FAIL]
Found 2850 paused or stopped task(s) older than 30 days
--------------------------------------------------------------------------------
Continue with step [Delete old tasks]?, [y(yes), n(no), q(quit)] y
Delete tasks:
- Deleted old stopped and paused tasks: 2850                             [OK]
--------------------------------------------------------------------------------
Rerunning the check after fix procedure
Check for old tasks in paused/stopped state:                             [OK]
--------------------------------------------------------------------------------
Check for pending tasks which are safe to delete:                        [OK]
--------------------------------------------------------------------------------
Check for tasks in planning state:                                       [OK]
--------------------------------------------------------------------------------
Check for recommended disk speed of pulp, mongodb, pgsql dir.:
| Finished

/var/lib/pulp : 269 MB/sec
/var/lib/mongodb : 203 MB/sec
/var/lib/pgsql : 64 MB/sec                                               [FAIL]
Slow disk detected /var/lib/pgsql mounted on /dev/mapper/vg_rhn-lv_rhnpgsql.
             Actual disk speed: 64 MB/sec
             Expected disk speed: 80 MB/sec.
--------------------------------------------------------------------------------
Check to verify remote_execution_ssh settings already exist:             [OK]
--------------------------------------------------------------------------------
Validate availability of repositories:
- Validating availability of repositories for 6.3                       [OK]
--------------------------------------------------------------------------------
Scenario [Checks before upgrading to Satellite 6.3] failed.

The following steps ended up in failing state:
```

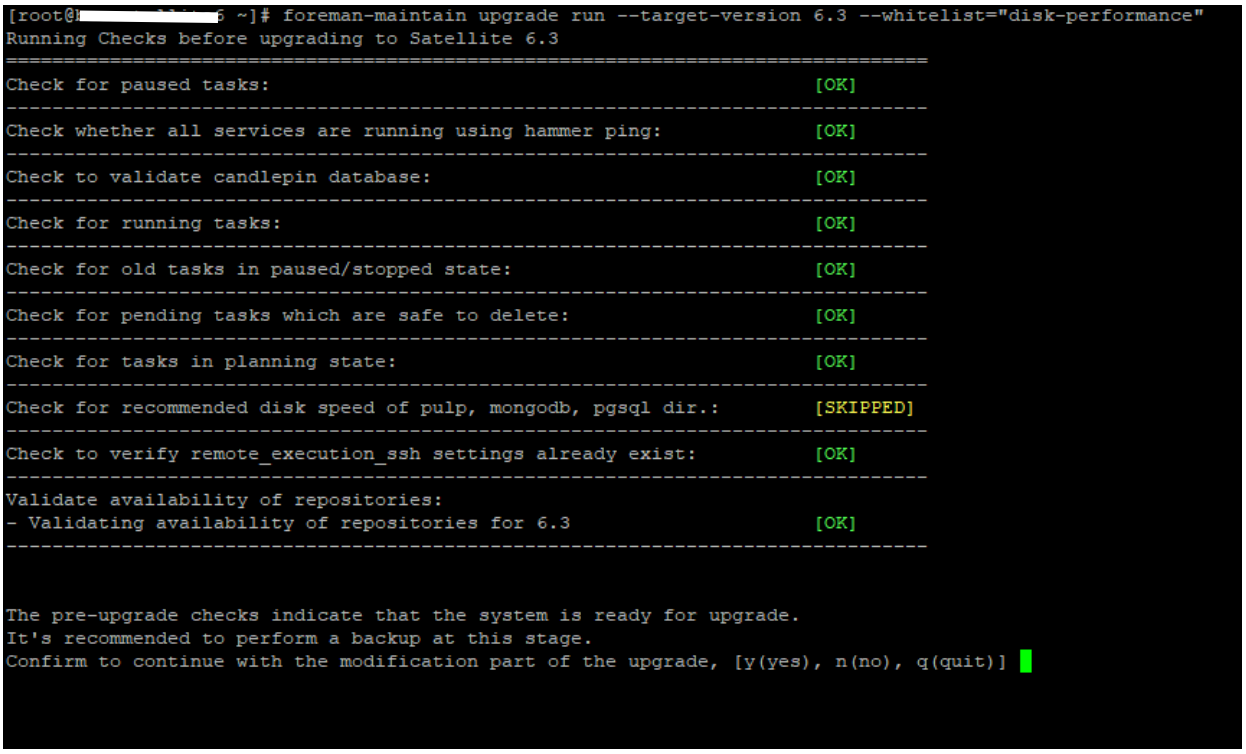
```
The following steps ended up in failing state:

  [disk-performance]

Resolve the failed steps and rerun
the command. In case the failures are false positives,
use --whitelist="disk-performance"


[root@bsesatellite6 ~]#
```

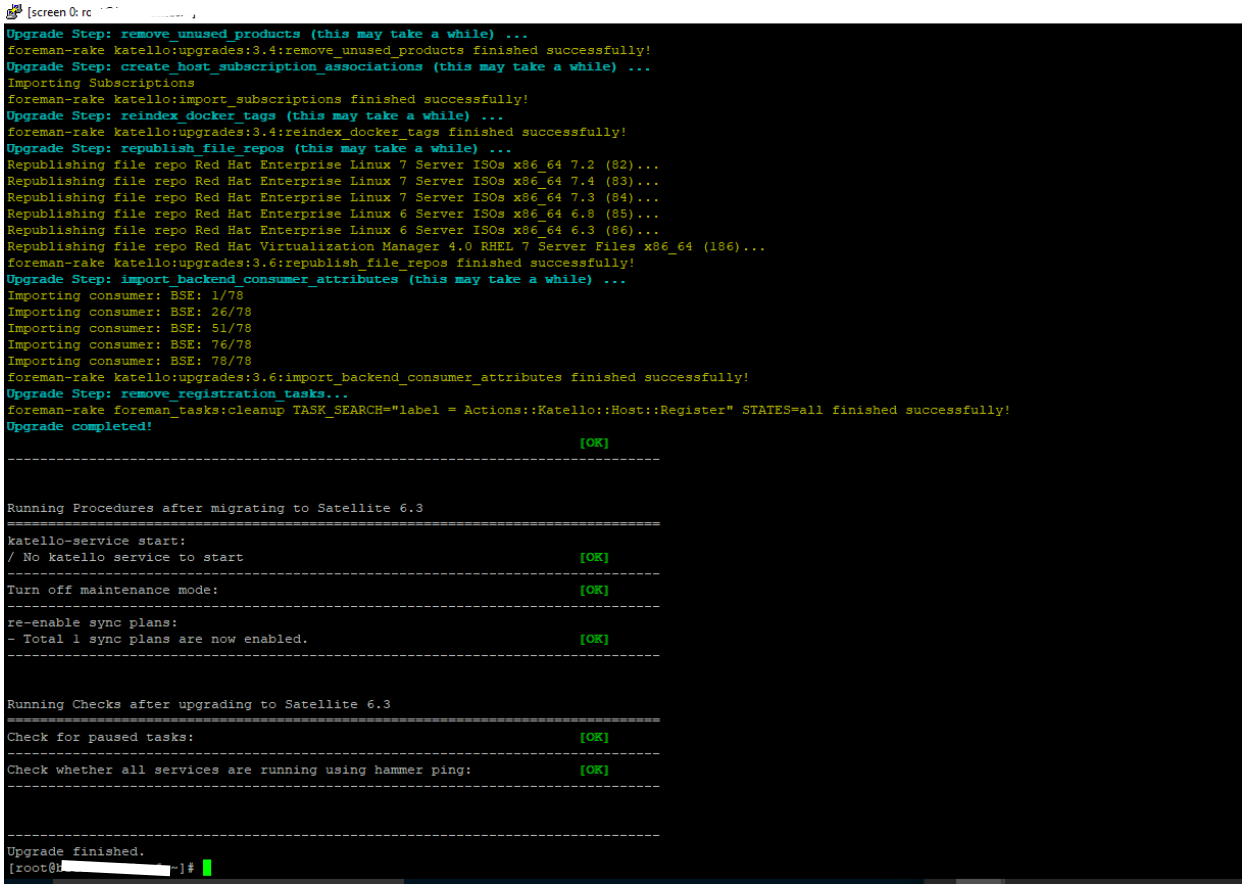
Due to the lengthy upgrade time, use a utility such as screen to suspend and reattach a communication session. You can then check the upgrade progress without staying connected to the command shell continuously

- 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.

    - Perform the upgrade:

```
# foreman-maintain upgrade run --target-version 6.3 --whitelist=”disk-performance”
```





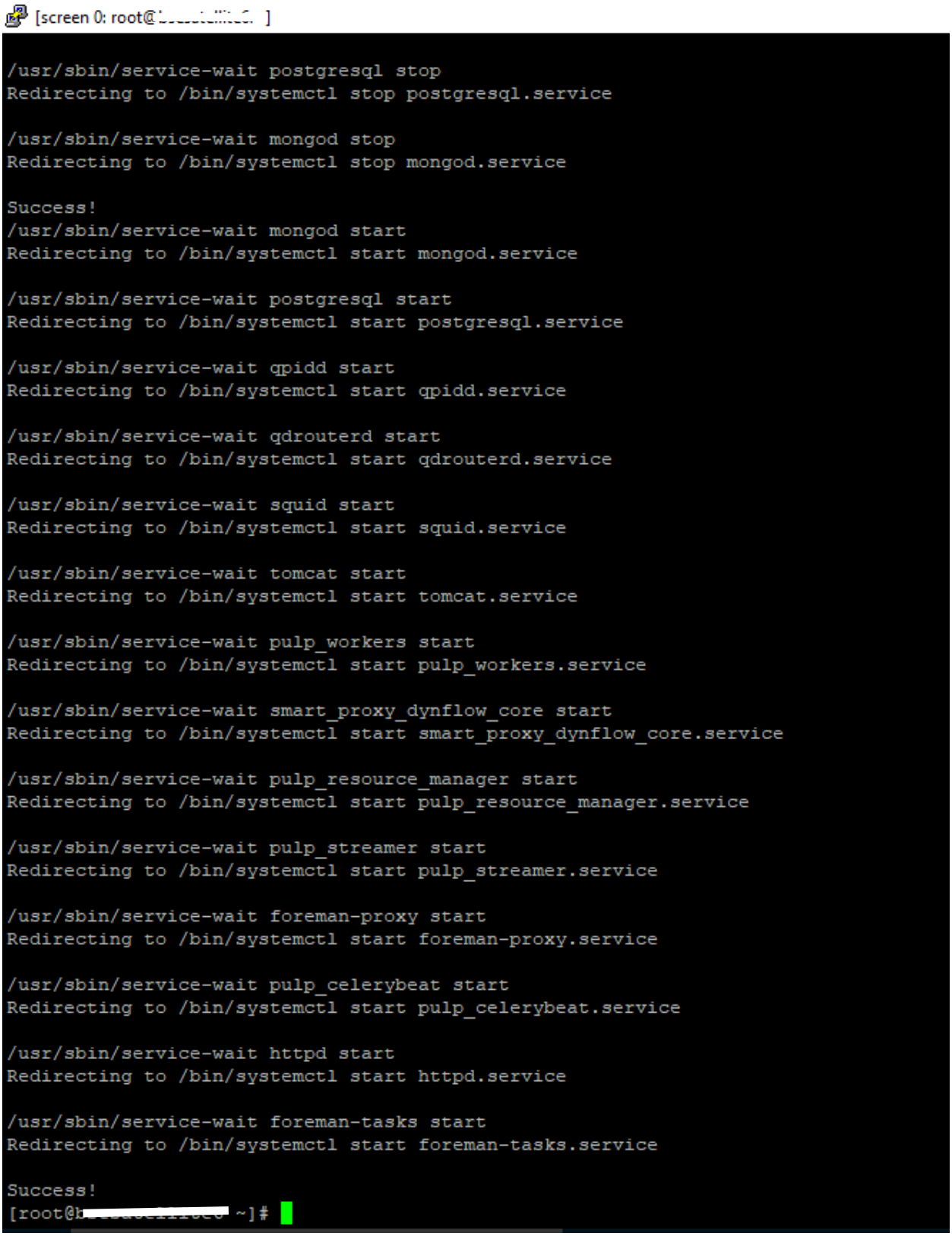
If using a BASH shell, after a successful or failed upgrade, enter:

 # hash -d katello-service 2> /dev/null

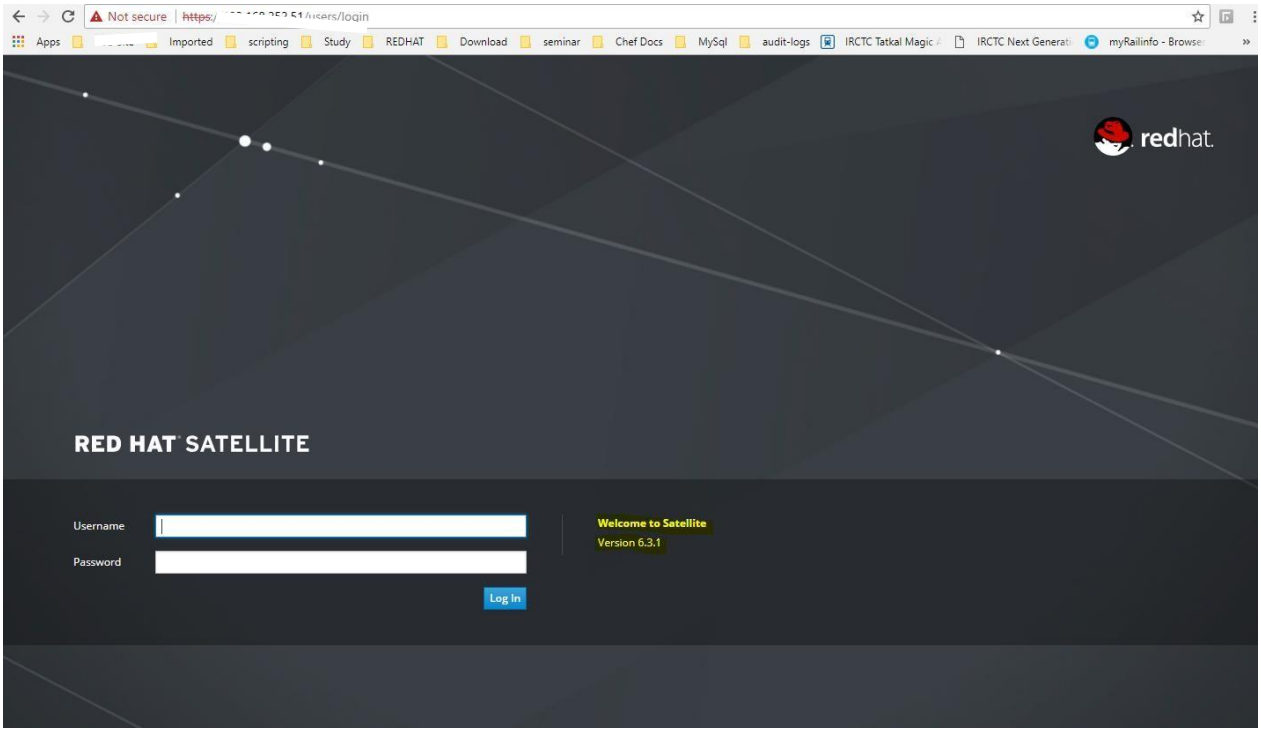
- Check and restore any changes required to the DNS and DHCP configuration files using the backups made earlier.

- If you made changes in the previous step, restart **Katello** services:

  # katello-service restart


[screen 0: root@
/usr/sbin/service-wait postgresql stop
Redirecting to /bin/systemctl stop postgresql.service
/usr/sbin/service-wait mongod stop
Redirecting to /bin/systemctl stop mongod.service
Success!
/usr/sbin/service-wait mongod start
Redirecting to /bin/systemctl start mongod.service
/usr/sbin/service-wait postgresql start
Redirecting to /bin/systemctl start postgresql.service
/usr/sbin/service-wait qpidd start
Redirecting to /bin/systemctl start qpidd.service
/usr/sbin/service-wait qdrouterd start
Redirecting to /bin/systemctl start qdrouterd.service
/usr/sbin/service-wait squid start
Redirecting to /bin/systemctl start squid.service
/usr/sbin/service-wait tomcat start
Redirecting to /bin/systemctl start tomcat.service
/usr/sbin/service-wait pulp_workers start
Redirecting to /bin/systemctl start pulp_workers.service
/usr/sbin/service-wait smart_proxy_dynflow_core start
Redirecting to /bin/systemctl start smart_proxy_dynflow_core.service
/usr/sbin/service-wait pulp_resource_manager start
Redirecting to /bin/systemctl start pulp_resource_manager.service
/usr/sbin/service-wait pulp_streamer start
Redirecting to /bin/systemctl start pulp_streamer.service
/usr/sbin/service-wait foreman-proxy start
Redirecting to /bin/systemctl start foreman-proxy.service
/usr/sbin/service-wait pulp_celerybeat start
Redirecting to /bin/systemctl start pulp_celerybeat.service
/usr/sbin/service-wait httpd start
Redirecting to /bin/systemctl start httpd.service
/usr/sbin/service-wait foreman-tasks start
Redirecting to /bin/systemctl start foreman-tasks.service
Success!
[root@ ~]#

- Check the **Satellite 6.3 WebUI** Login



Now it successfully upgraded to **Satellite 6.3.**

10. References

- https://access.redhat.com/documentation/en-us/red_hat_satellite/6.2/html/server_administration_guide/index
- https://access.redhat.com/solutions/2650071
- https://access.redhat.com/solutions/3145861
- https://access.redhat.com/documentation/en-us/red_hat_satellite/6.2/html/provisioning_guide/index
- https://access.redhat.com/solutions/2599261

****