



# Red Hat Training and Certification

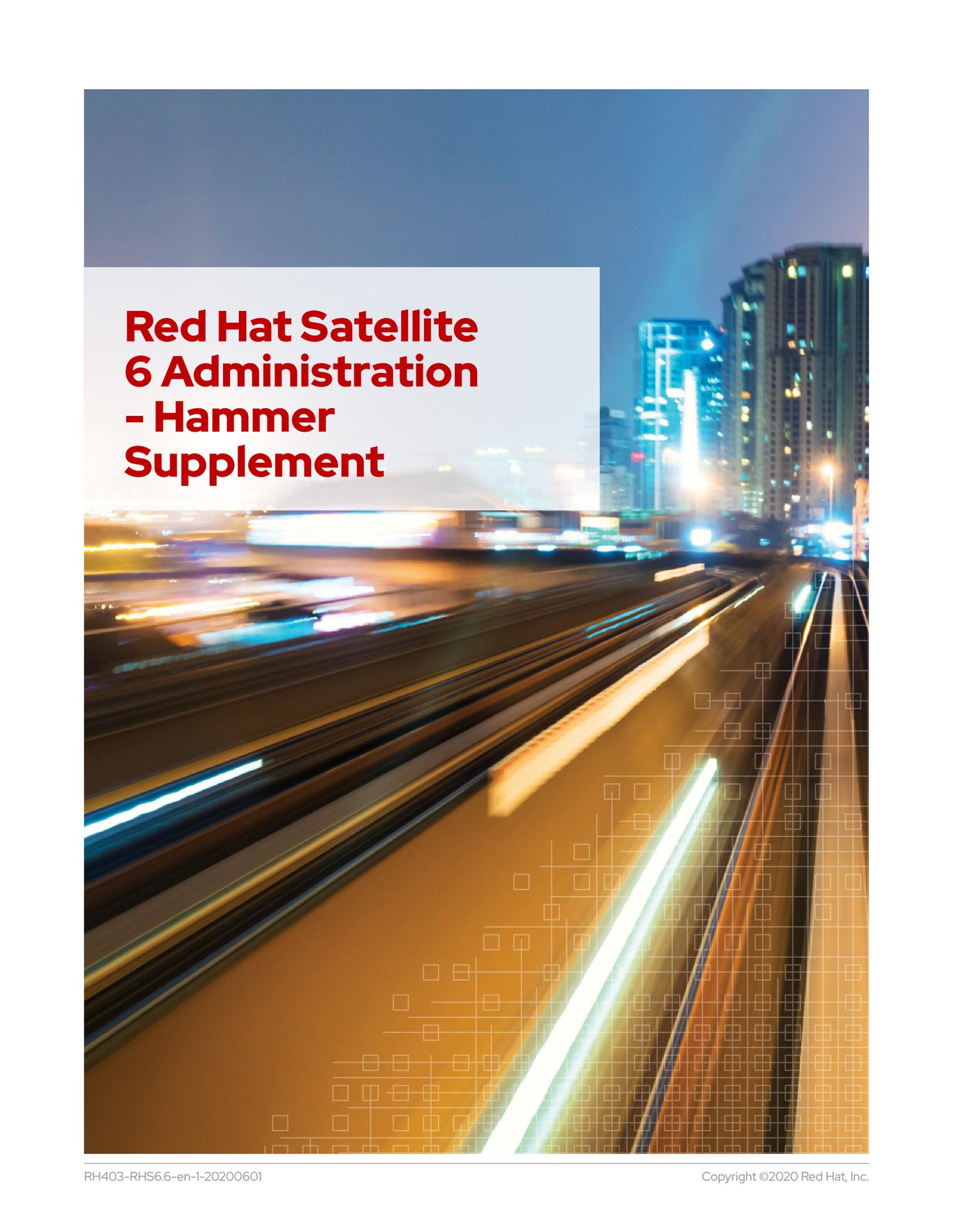
## Student Workbook

Red Hat Satellite 6.6 RH403

## Red Hat Satellite 6 Administration - Hammer Supplement

Edition 1



A long-exposure photograph of a city street at night, showing light trails from cars and buildings in the background. A semi-transparent white box is overlaid on the left side of the image, containing the title text. A faint grid pattern is visible in the bottom right corner of the image.

# **Red Hat Satellite 6 Administration - Hammer Supplement**

# **Red Hat Satellite 6.6 RH403**

## **Red Hat Satellite 6 Administration - Hammer Supplement**

### **Edition 1 20200601**

#### **Publication date 20200601**

Author: Travis Michette

Copyright © 2020 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2020 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please e-mail [training@redhat.com](mailto:training@redhat.com) or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, Hibernate, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

|   |          |
|---|----------|
| <b>1. Red Hat Satellite Server Administration and Maintenance</b> | <b>1</b> |
| Using and Understanding the Hammer CLI .....                      | 2        |
| Guided Exercise: Hammer CLI Exploration .....                     | 9        |



## Chapter 1

# Red Hat Satellite Server Administration and Maintenance

### Goal

Maintain Red Hat Satellite for security, recoverability, and growth.

### Objectives

- Utilize the Hammer CLI to perform routine administration and maintenance tasks on Satellite Server

### Sections

- Using and Understanding the Hammer CLI (and Guided Exercise)

# Using and Understanding the Hammer CLI

## Objectives

After completing this section, students should be able to:

- Setup and use the Hammer CLI Shell
- Setup and configure the **foreman.yml** file for use with Hammer.
- Utilize the Hammer CLI to perform various administrative tasks to Red Hat Satellite server from the command line.

## Introduction to Hammer

Hammer is a CLI tool that is provided with Red Hat Satellite 6. It relies on the Satellite 6 Rest API and can be used as a friendly interface to interact with the Satellite API. The Hammer CLI requires access to the Satellite server's shell (typically via SSH) which can limit user access when compared with the Satellite WebUI.

The Hammer CLI can be used to create commands for use in scripts and automating complex tasks. The output from Hammer commands can be sent to other tools or simple text files for processing and integration for other systems within the environment. The Hammer command can also show API calls/responses by using the **-d** option when issuing the command.



### Important

It is important to note that while Hammer commands mirror the tasks done with the WebUI, some features may not be implemented in Hammer as the WebUI has development priority. There may be some items that are only implemented from within the Satellite WebUI.

## Configuring and Using Hammer

Hammer is generally used one of two ways. Either Hammer is used from the regular CLI and part of scripts with the following format:

```
[root@satellite ~]# hammer settings list | grep footer
```

In the example above, the Hammer command is used on the shell to list settings. The output has been piped to **grep** so that the keyword "footer" can be located in the search and have results returned. This method of using Hammer is often used in scripts. In order to use Hammer from the CLI, there must be a configuration of a Hammer configuration file to provide authentication and the URL for the Satellite.

The second method of using Hammer interactively is by utilizing the Hammer shell. The Hammer shell is an interactive shell for Hammer requiring authentication only when opening/launching the Hammer shell. If the Hammer authentication file has been configured, there is no need to provide authorization information. If the Hammer authentication file has not been configured, the Hammer shell can be launched with the following command syntax:

```
[root@satellite ~]# hammer --username admin_user_name --password admin_password
shell
```

The Hammer shell is useful for administrators wanting to work interactively with Satellite and Hammer. By utilizing the Hammer shell, administrators don't need to type **hammer** before each operation. The other benefit of the Hammer shell is that it understands and support command-line completion.

## Getting Help with Hammer

Hammer has numerous commands to interact with the Satellite server. Each command is somewhat unique in syntax requirements and the output that is provided. The **hammer --help** command can be used to list all available commands.

The **hammer --help** command also allows subcommands and options to be explored.

```
[root@satellite ~]# hammer activation-key --help
Usage:
  hammer activation-key [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:
  SUBCOMMAND          Subcommand
  [ARG] ...           Subcommand arguments

Subcommands:
  add-host-collection  Associate a resource
  add-subscription     Add subscription
  content-override     Override product content defaults
  copy                 Copy an activation key
  create               Create an activation key
  delete               Destroy an activation key
  host-collections    List associated host collections
  info                 Show an activation key
  list                 List activation keys
  product-content     List associated products
  remove-host-collection Disassociate a resource
  remove-subscription Remove subscription
  subscriptions        List associated subscriptions
  update              Update an activation key

Options:
  -h, --help          Print help
```

Each command and subcommand can also yield more detailed help. For example, if the goal is to create an activation key, the syntax to get help would be the following:

```
[root@satellite ~]# hammer activation-key create --help
Usage:
  hammer activation-key create [OPTIONS]

Options:
  --content-view CONTENT_VIEW_NAME  Content view name to search
  by
```

```

--content-view-id CONTENT_VIEW_ID          Content view numeric
identifier
--description DESCRIPTION                  Description
--environment ENVIRONMENT_NAME            Lifecycle environment name to
search by (--environment is deprecated: Use --lifecycle-environment instead)
--environment-id ENVIRONMENT_ID           (--environment-id is
deprecated: Use --lifecycle-environment-id instead)
--lifecycle-environment LIFECYCLE_ENVIRONMENT_NAME Lifecycle environment name to
search by
--lifecycle-environment-id LIFECYCLE_ENVIRONMENT_ID
--max-hosts MAX_HOSTS                     Maximum number of registered
content hosts
--name NAME                                Name
--organization ORGANIZATION_NAME          Organization name to search
by
--organization-id ORGANIZATION_ID         Organization ID
--organization-label ORGANIZATION_LABEL   Organization label to search
by
--unlimited-hosts                          Set hosts max to unlimited
-h, --help                                Print help

```

The **hammer --help** commands can continue to expand and provide more detailed help for each of the primary commands and subcommands. There will be no additional help provided once the level is reached where only options are remaining.

## The Hammer Configuration File and Authentication

There are three (3) primary methods for Hammer authentication:

- Hammer authentication session
- Storing credentials in the Hammer configuration file
- Providing credentials with each hammer command



### Tip

The Hammer configuration file method is recommended for running commands automatically when using Hammer with scripts or cron jobs.

### Hammer Authentication Sessions

Hammer authentication session is a cache capable of storing credentials that are provided to Hammer. The authentication credentials are provided once at the beginning of the session. By utilizing Hammer authentication sessions, it is possible to avoid storing credentials in a configuration file or script. In order to enable Hammer authentication sessions, the **~/hammer/cli.modules.d/foreman.yml** must be properly configured and you must both start and end the hammer sessions.

- Enabling Hammer Sessions in the Config File

```

[root@satellite ~]# vim ~/.hammer/cli.modules.d/foreman.yml
:foreman:
:use_sessions: true

```

- Starting a Hammer Session

```
[root@satellite ~]# hammer auth login
[Foreman] Username: admin
[Foreman] Password for admin:
Successfully logged in as 'admin'
```

- Viewing Hammer Session Status

```
[root@satellite ~]# hammer auth status
Session exists, currently logged in as 'admin'.
```

- Ending a Hammer Session

```
[root@satellite ~]# hammer auth logout
Logged out.
```

## Hammer Configuration File

The hammer configuration file gets created for the root user automatically if Satellite was installed and provisioned using the **--foreman-initial-admin-username** and **--foreman-initial-admin-password** options. This would store the credentials in **~/.hammer/cli.modules.d/foreman.yml** automatically. It is also possible to store the credentials manually by editing the file and adding the **username** and **password** options to the file.

```
[root@satellite ~]# cat ~/.hammer/cli.modules.d/foreman.yml
:foreman:
  :username: 'admin'
  :password: 'redhat'
  :refresh_cache: false
  :request_timeout: 120
```



### Important

It is very important that when modifying the **foreman.yml** file that only spaces are used for indentation. The **tab** key should never be used for indentation in hammer configuration files.

## Providing Credentials on the Command Line

If Hammer authentication sessions are not used and Satellite credentials aren't saved in the **~/.hammer/cli.modules.d/foreman.yml** Hammer configuration file, the Hammer command will prompt for credentials each time a command is executed. It is also possible to specify and provide the credentials on the command line by using:

```
[root@satellite ~]# hammer -u <admin_username> -p <admin_password> subcommands/options
```

Each time the **hammer** command is run, the credentials must be provided.

## The Hammer Shell

The Hammer shell is an interactive shell for issuing **hammer** commands. Once the shell is invoked, all Hammer subcommands can be entered directly without typing **hammer** for the hammer commands. The Hammer shell is often used for testing and debugging of commands before using them in a script.

- Opening the Hammer Shell

```
[root@satellite ~]# hammer shell
Welcome to the hammer interactive shell.
Type 'help' for usage information.
hammer>
```

- Using the Hammer Shell

```
hammer> organization list
Warning: Option --environments is deprecated. Use --puppet-environments instead
Warning: Option --environment-ids is deprecated. Use --puppet-environment-ids
instead
---|-----|-----|-----|-----
ID | TITLE                | NAME                | DESCRIPTION | LABEL
---|-----|-----|-----|-----
1 | Default Organization | Default Organization |             |
  Default_Organization
3 | Operations            | Operations          | Operations  | Operations
---|-----|-----|-----|-----
```

- Exiting the Hammer Shell

```
hammer> exit
[root@satellite ~]#
```



### Tip

It is also possible to exit the Hammer shell by pressing **[CTRL + D]**

## Using Hammer to Generate Formatted Output

Hammer commands allow administrators to produce output in multiple default formats. The production of output in a different format is often useful for providing high-level reporting and for use with data analysis tools and procedures. In order to generate output, the **--output output\_format** directives must be specified. Hammer can produce outputs in the following supported formats:

- **table** - generates output in the form of a table. This is the *default* setting.
- **base** - generates output in form of key-value pairs.
- **yaml** - generates output in the YAML format
- **csv** - generates output in Comma Separated Values (CSV) format. It is possible to define custom separators by using the **--csv** and **--csv-separator** options.

- **json** - generates output in JavaScript Object Notation (JSON) format.
- **silent** - suppresses output

**Tip**

It is possible to hide header information in the hammer output by appending the **--no-headers** directive to the command.

```
[root@satellite ~]# hammer --no-headers organization list
Warning: Option --environments is deprecated. Use --puppet-environments
instead
Warning: Option --environment-ids is deprecated. Use --puppet-environment-ids
instead
1 | Default Organization | Default Organization |
  Default_Organization
3 | Operations           | Operations           | Operations | Operations
```

In order to properly use Hammer to get the data to a file, it is necessary to use console redirection to place the output in a file.

```
[root@satellite ~]# hammer --output --csv organization list
Warning: Option --environments is deprecated. Use --puppet-environments instead
Warning: Option --environment-ids is deprecated. Use --puppet-environment-ids
instead
Id:          1
Title:       Default Organization
Name:        Default Organization
Description:
Label:       Default_Organization

Id:          3
Title:       Operations
Name:        Operations
Description: Operations
Label:       Operations
```

```
[root@satellite ~]# hammer --output --csv organization list > org_list.csv
Warning: Option --environments is deprecated. Use --puppet-environments instead
Warning: Option --environment-ids is deprecated. Use --puppet-environment-ids
instead
```

## Troubleshooting with Hammer

The Hammer command can also be used to troubleshoot core Satellite services. The **hammer ping** command will check and return the status of the core services within Red Hat Satellite. The **hammer ping** command can be used together with the **satellite-maintain service status** command to ensure that Satellite is running properly and also used to diagnose and troubleshoot issues with Satellite services.

```
[root@satellite ~]# hammer ping
candlepin:
  Status:          ok
  Server Response: Duration: 9ms
candlepin_auth:
  Status:          ok
  Server Response: Duration: 8ms
pulp:
  Status:          ok
  Server Response: Duration: 28ms
pulp_auth:
  Status:          ok
  Server Response: Duration: 14ms
foreman_tasks:
  Status:          ok
  Server Response: Duration: 2ms
```



## References

### Getting Started with Satellite 6 Command Line

<https://gsw-hammer.documentation.rocks/>

The Getting Started with Satellite 6 Command Line book is a very useful book looking at Satellite 6 for command line features. It provides some step-by-step examples for several Hammer commands.

### Use Hammer in Satellite 6 Without a Password

<https://gsw-hammer.documentation.rocks/>

The Use Hammer in Red Hat Satellite 6 Without a Password shows an older method of using a **cli\_config.yml** file to configure hammer credentials.



## References

For more information, refer to the *Hammer CLI Guide* at

[https://access.redhat.com/documentation/en-us/red\\_hat\\_satellite/6.6/html/hammer\\_cli\\_guide/index](https://access.redhat.com/documentation/en-us/red_hat_satellite/6.6/html/hammer_cli_guide/index)

For more information, refer to the *Hammer Cheat Sheet* at

[https://access.redhat.com/documentation/en-us/red\\_hat\\_satellite/6.6/html/hammer\\_cheat\\_sheet/index](https://access.redhat.com/documentation/en-us/red_hat_satellite/6.6/html/hammer_cheat_sheet/index)

For more information, refer to the *hammer commands fail with Invalid Username or Password after Red Hat Satellite 6.3 installation when setting up cli\_config.yml* at

<https://access.redhat.com/solutions/3436701>

## ▶ Guided Exercise

# Hammer CLI Exploration

In this exercise, you will be using the Hammer command to explore and manage the Satellite server from the command line. You will be configuring the Hammer CLI to run from an authentication file as the student user and you will be performing various administration tasks with the hammer command.

## Outcomes

You should be able to:

- Create the `~/.hammer/cli.modules.d/foreman.yml` allowing use of Hammer without entering credentials
- List Satellite Organizations using Hammer
- Output a list of Satellite subscriptions using Hammer as a CSV File
- Set the Download/Content Synchronization Policy at the Satellite level using Hammer
- Use Hammer to set a login banner
- Use the `hammer ping` command to diagnose and verify Satellite services

## Before You Begin

Ensure that the Satellite server is fully functional and that you can login with the UN/PW: **admin/redhat** in the WebUI.

Log in as the **student** user on **workstation** using **student** as the password.

- ▶ **1.** On **workstation**, use **ssh** to log in to **satellite** as **student**.

```
[student@workstation ~]$ ssh student@satellite
[student@satellite ~]$
```

- ▶ **2.** Create the `~/.hammer/cli.modules.d/foreman.yml` directory, subdirectory, and file.
  - 2.1. Create the directory structure with the **mkdir -p** option to create all parent directories.

```
[student@satellite ~]$ mkdir -p ~/.hammer/cli.modules.d
```

- 2.2. Create the **foreman.yml** file with the needed Hammer credentials and enter the contents as shown below.

```
[student@satellite ~]$ vim ~/.hammer/cli.modules.d/foreman.yml
:foreman:
  :username: 'admin'
  :password: 'redhat'
```



### Warning

Please note that the configuration of the `~/.hammer/cli.modules.d/foreman.yml` will allow the student user to authenticate as the Red Hat Satellite user **admin**. By creating this configuration file, both the username and password are stored in plaintext on the system.

- ▶ **3.** Verify Satellite services are operational using the **hammer ping** command.

```
[student@satellite ~]$ hammer ping
candlepin:
  Status:          ok
  Server Response: Duration: 9ms
candlepin_auth:
  Status:          ok
  Server Response: Duration: 9ms
pulp:
  Status:          ok
  Server Response: Duration: 31ms
pulp_auth:
  Status:          ok
  Server Response: Duration: 18ms
foreman_tasks:
  Status:          ok
  Server Response: Duration: 2ms
```



### Tip

The **hammer ping** command is one of the few commands that does not require authentication in order to execute. It is possible to perform a **hammer ping** without having the authentication file in place and without knowing the passwords.

- ▶ **4.** Use Hammer to obtain a listing of Satellite organizations in table format.

```
[student@satellite ~]$ hammer organization list
Warning: Option --environments is deprecated. Use --puppet-environments instead
Warning: Option --environment-ids is deprecated. Use --puppet-environment-ids
instead
---|-----|-----|-----|
ID | TITLE                | NAME                | DESCRIPTION | LABEL
---|-----|-----|-----|
1  | Default Organization | Default Organization |             |
  | Default_Organization
3  | Operations            | Operations          | Operations  | Operations
---|-----|-----|-----|
```

- ▶ 5. Use Hammer to output a list of subscriptions for the **Operations** organization as a CSV file.

5.1. Using the **hammer organization list** command, obtain a list of organizations.

```
[student@satellite ~]$ hammer organization list
Warning: Option --environments is deprecated. Use --puppet-environments instead
Warning: Option --environment-ids is deprecated. Use --puppet-environment-ids
instead
-----|-----|-----|-----
ID | TITLE                | NAME                | DESCRIPTION | LABEL
-----|-----|-----|-----|-----
1  | Default Organization | Default Organization |             |
   | Default_Organization
3  | Operations           | Operations          | Operations  | Operations
-----|-----|-----|-----|-----
```

5.2. Use either the **ORG\_ID**, **ORG\_Name**, or **ORG\_Label** to provide to the **hammer subscription list** command and verify the output.

```
[student@satellite ~]$ hammer --output --csv subscription list --organization-id 3
ID:          1
UUID:        2c997a8f6f6cd6a5016f6cd86ce10807
Name:        Red Hat Enterprise Linux Server, Standard (Physical or Virtual Nodes)
Type:        Physical
Contract:    12017022
Account:     5508397
Support:     Standard
End Date:    2024/09/30 03:59:59
Quantity:    10
Consumed:    1

ID:          2
UUID:        2c997a8f6f6cd6a5016f6cd86d1a080d
Name:        Red Hat Satellite Infrastructure Subscription
Type:        Physical
Contract:    12055880
Account:     5508397
Support:     Premium
End Date:    2024/11/26 04:59:59
Quantity:    10
Consumed:    1
```

5.3. Use either the **ORG\_ID**, **ORG\_Name**, or **ORG\_Label** to provide to the **hammer subscription list** command. Redirect the output to the **Operations\_Subs.csv** file.

```
[student@satellite ~]$ hammer --output --csv subscription list --organization-id 3 >
Operations_Subs.csv
```

5.4. Verify the contents of the **Operations\_Subs.csv** file.

```
[student@satellite ~]$ cat Operations_Sub.csv
ID:      1
UUID:    2c997a8f6f6cd6a5016f6cd86ce10807
Name:    Red Hat Enterprise Linux Server, Standard (Physical or Virtual Nodes)
Type:    Physical
Contract: 12017022
Account: 5508397
Support: Standard
End Date: 2024/09/30 03:59:59
Quantity: 10
Consumed: 1

ID:      2
UUID:    2c997a8f6f6cd6a5016f6cd86d1a080d
Name:    Red Hat Satellite Infrastructure Subscription
Type:    Physical
Contract: 12055880
Account: 5508397
Support: Premium
End Date: 2024/11/26 04:59:59
Quantity: 10
Consumed: 1
```

- ▶ 6. Use Hammer to set the Default Download and Content Synchronization Policy

```
[student@satellite ~]$ hammer settings set --name default_download_policy --value
immediate
Setting [default_download_policy] updated to [immediate].
```

- ▶ 7. Use Hammer to set a Login Warning banner on the Satellite WebUI login page. The message should be the following: *Warning, you are accessing a sensitive computer system. This message is required by the DISA STIG settings. Anything you do is being monitored.*

```
[student@satellite ~]$ hammer settings set --name login_text --value "Warning, you are
accessing a sensitive computer system. This message is required by the DISA STIG settings.
Anything you do is being monitored."
Setting [login_text] updated to [Warning, you are accessing a sensitive computer
system. This message is required by the DISA STIG settings. Anything you do is
being monitored.].
```

This concludes the guided exercise.