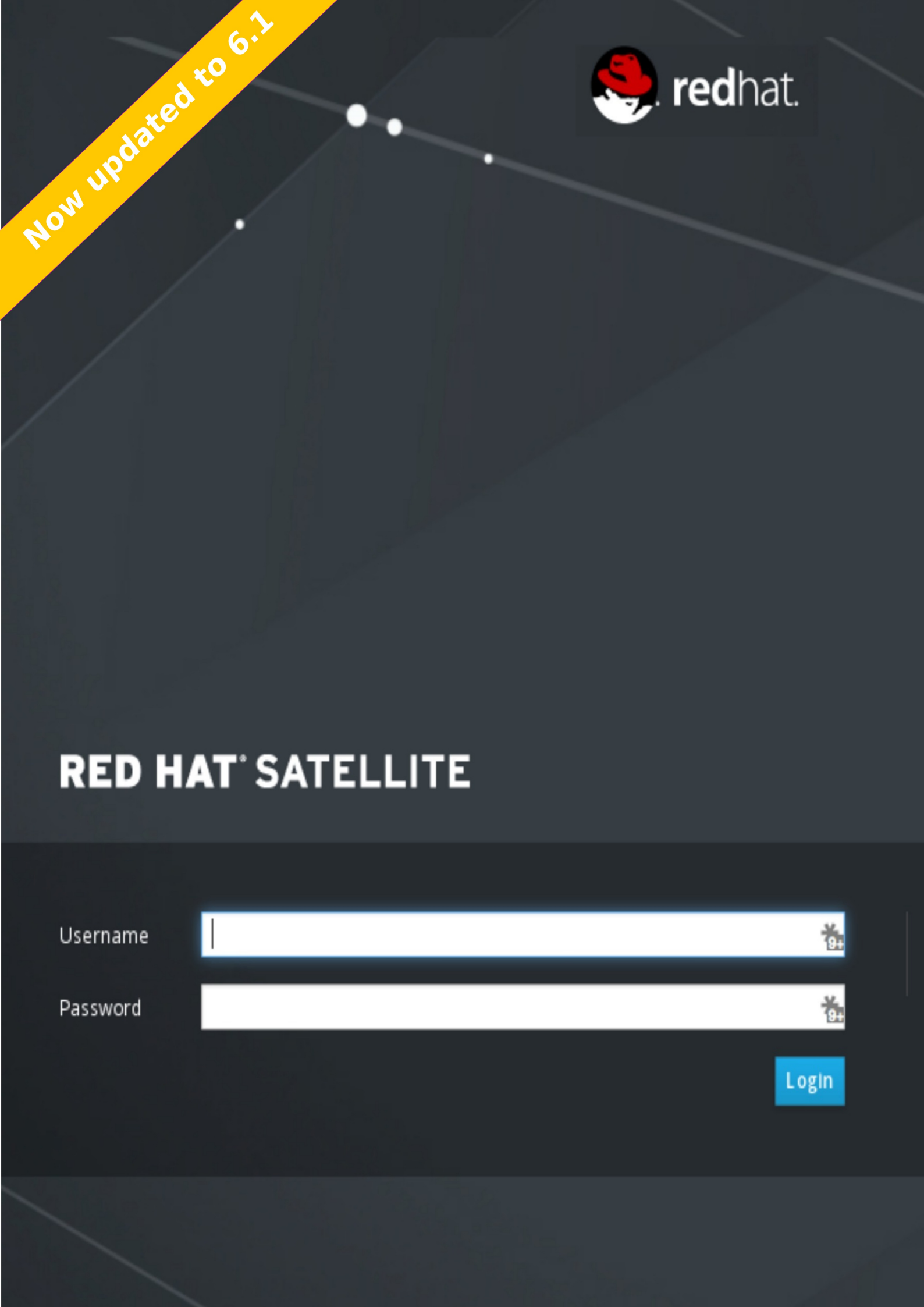

Now updated to 6.1
redhat.
RED HAT® SATELLITE
Username
Password
Login

Table of Contents

Getting Started with Satellite 6 Command Line

> **NOTE**: This book has now been updated to Satellite 6.2
>
> The previous version of the book had a number of work arounds included that are now, no longer necessary

---

In this, the second book in the series, we look at using Satellite 6's command line tool, called Hammer

While the web interface is very nice, its often more efficient to use the command line, especially as it enables scripting.

We shall again start of with focusing on configuring a freshly installed Satellite so that it can provision machines. We will use an "all in one" setup, where the Satellite box will also perform DNS, DHCP and TFTP

Many of the first steps are identical to those in the sister book **Getting Started with Satellite 6**, but we have included them here anyway

Installation

Pre-Requisites

Before we start, you need

- vanilla install of RHEL. (we will be using RHEL6, but RHEL7 is also supported)
- valid entitlement for RHEL and entitlement for Satellite
- a login to access.redat.com (for creating and downloading the manifest)

Firewall configuration

Its worth getting the firewall configured at this stage, so that we dont forget later. I shall assume a default firewall config exists. Configure the firewall any way you feel confortable, there is a quick option below.

RHEL 7 Firewall

RHEL 7 uses firewalld, which is new to all of us. Below are some simple steps to get the firewall setup as we need it

Start off by verifying what the default zone is

```
firewall-cmd --get-active-zones
```

You should see something like this that indicates its active zone is *public*

```
public
  interfaces: eth0
```

If the active zones list is empty, its probably because the NIC has not been assigned to a group:

```
firewall-cmd --zone=public \
 --change-interface=nicInterfaceName
```

OK, now we have verified that, lets take a look at the default rules in place

```
firewall-cmd --zone=public --list-all
...
(output omited)
```

Of course this is not what we require, so lets add in all the ports we require including the ones for the additional capsules (DNS & DHCP).

```
firewall-cmd  --permanent --zone=public  \
 --add-service=RH-Satellite-6 --add-service=dhcp \
 --add-service=dns --add-service=tftp
# 6.2 still seems to needs also 8000 TCP for foreman-proxy
# https://bugzilla.redhat.com/show_bug.cgi?id=1248665
# [root@satellite ~]# lsof -i :8000
# COMMAND  PID          USER   FD   TYPE DEVICE SIZE/OFF NODE NA
ME
# ruby    2740 foreman-proxy   10u  IPv4  22409      0t0  TCP *:
irdmi (LISTEN)

firewall-cmd --zone=public --add-port="8000/tcp" --permanent
```

Now run that command again.

```
firewall-cmd --zone=public --list-all
public (default, active)
  interfaces: eth0
  sources:
  services: RH-Satellite-6 dhcp dhcpv6-client dns ssh tftp
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

As I have used `--permanent` above, we will have to restart the firewall for this to take effect. We use `firewall-cmd --reload` as it will keep the existing state information. See the man page for more details

```
firewall-cmd  --reload
```

More information on what each of these ports are for can be found in Installation Guide Prerequisites

Base Install

Start with a fresh installation of RHEL 7.2, which was installed via PXE and kickstart with an FTP source. I prefer this, over templates or machines deployed by Satellite/Foreman as it keeps it as clean as possible and there is no existing puppet configuration, might have lead to problems later on.

Plus, my test Sat 6 has a 300GiB vda, not keeping 2 Satelites on the hypervisor.

When installing, be sure to just use "Base" as the installation type.

Although I do add the following for comfort

- acpid
- chrony
- kexec-tools

Once installed, we will register the machine to Red Hat, and fully update it. Then we will change its Satellite version, add some subscriptions and download its manifest file and get the Satellite software installed.

Registration

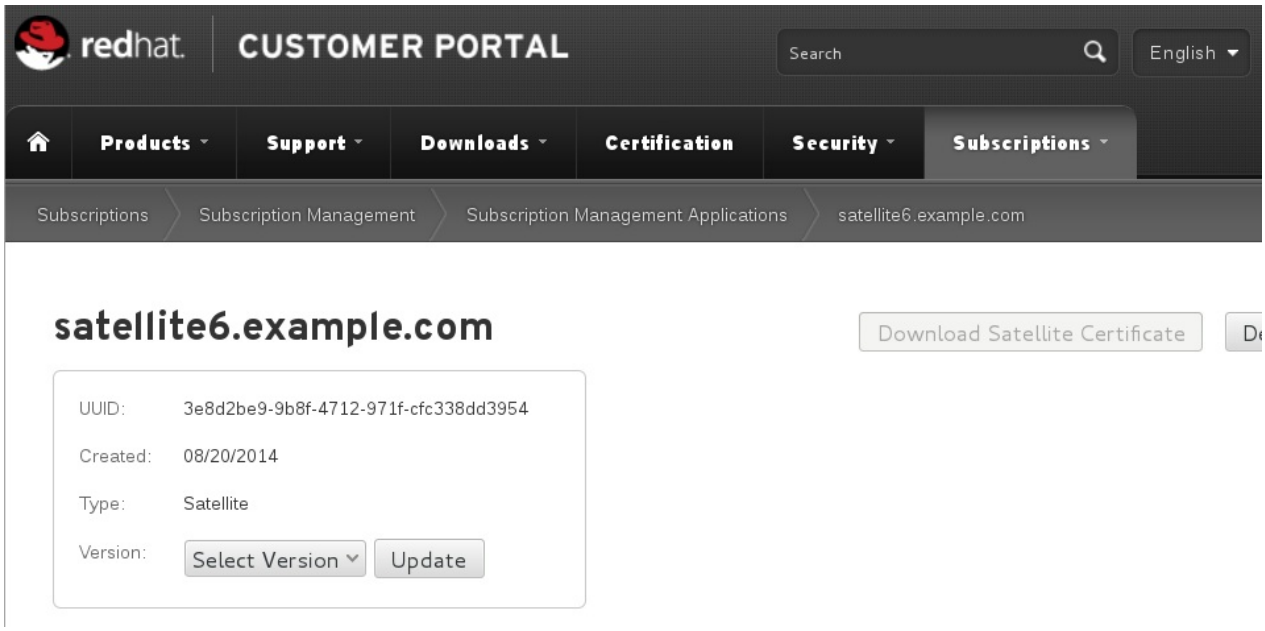
I like to follow a very precise way of registering the box, one that I have worked out over many installation methods. While its not the way currently mentioned in the official documentation, it works for me 100% of the time.

From a command prompt type the following

```
subscription-manager register --type=satellite
```

You will then be prompted for your credentials, enter them and the system should be correctly registered, but not attached/consuming entitlements

At this point its important to logon to the Red Hat Customer Portal and set the version of Satellite to 6.0



Attach Entitlements

OK, lets get a list of whats availble to you, you are specifically looking for the **Pool IDs** here

```
subscription-manager list --available --all|less
```

Search for the Satellite Subscription, as a Red Hat employee, some of the output I see is as follows, yours will look a little different

```
Subscription Name: Red Hat Satellite Subscription
Provides:          Red Hat Software Collections (for RHEL Server
)
                   Red Hat Satellite Capsule
                   Red Hat Satellite
                   ....

SKU:               SER---US
Pool ID:           aaaabbbbccccddddeeeeffffgggghhh
Available:         17
Suggested:         1
Service Level:     Self-Support
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              01/01/22
System Type:       Physical
```

You are interested in the **Pool ID** of the subscription

You can **attach** this to your server as follows

```
subscription-manager attach --pool=aaaabbbbccccddddeeeeffffggggh
hh
```

> **NOTE**: If you get an error that reads like this
>
> ```
> Too many content sets for certificate Red Hat
> Satellite Employee Subscription.
> A newer client may be available to address this
> problem.
> See kbase
> https://access.redhat.com/knowledge/node/129003 for
> more information.
> ```
>
> then make sure you log on to Red Hat and select the verison of Satellite, as mentioned in the previous section. Be sure to click the update button

You will then be able to attach to that pool. However, depending on your entitlemenmts, the pool you have atached to may not have **Software Collections**. If this is the case, use subscription manager to list all availble pools and attach one that contains **Software Collections**

This may enable too many repositories. The Satellite documentation makes clear which repositories you will need, and shows how to disable the ones you dont.

RHEL 7 Repos

```
subscription-manager repos --disable "*"

subscription-manager repos --enable rhel-7-server-rpms \
--enable rhel-server-rhscl-7-rpms \
--enable rhel-7-server-satellite-6.2-rpms
```

Once done, check that you have access to **exactly** three repos

```
# yum repolist

repo id                                    repo name
rhel-7-server-rpms/7Server/x86_64       Red Hat Enterprise Linux
 7 Server (RPMs)
rhel-7-server-satellite-6.2-rpms/x86_64 Red Hat Satellite 6.2 (f
or RHEL 7 Server) (RPMs)
rhel-server-rhscl-7-rpms/7Server/x86_64 Red Hat Software Collect
ions RPMs for Red Hat Enterprise Linux 7 Server
```

Once you have confirmed that you have access to exactly those three repositories, carry on to the next part.

Installation

Now we have the correct repos configured, update the server with the latest updates from Red Hat.

```
yum -y update
```

Next we perform the actual instalation.

```
yum install satellite
```

Initial Setup

The **katello-installer** is used to perform the initial setup and any future changes to the existing config. Its a puppet based installation and so can be re-run without overiding the previous settings.

We will create an "all-in-one" deployment, meaning that the Satellite will have the additional roles of TFTP proxy, DHCP server and DNS server added at install time.

The Satellite has 2 network interfaces and does not forward packets between them (to simulate a corporate firewall).

- the network "default" is the outside link (to sync repos)
- the network "SatTesting" is where DNS, DHCP and PXE are handled by the Satellite

```
# provided you put your CA cert and the signed cert of the Satel
lite there
# otgerwise skip this and drop the cert options when installing
cd /root/SSL-cert
katello-certs-check \
  -c 2016-06-17.crt \
  -r 2016-06-17.csr \
  -k 2016-06-17_unencrypted.key \
  -b pcfe-CA-pem.crt

cd /root/SSL-cert # provided you put your CA cert and the signed
 cert of rthe Satellite there
satellite-installer --scenario satellite \
  --foreman-initial-organization "Sat Test" \
  --foreman-initial-location "Engineering Building" \
  --foreman-proxy-dns true \
  --foreman-proxy-dns-forwarders 8.8.8.8 \
  --foreman-proxy-dns-zone sattest.example.com \
  --foreman-proxy-dns-reverse 2.168.192.in-addr.arpa \
  --foreman-proxy-dhcp true \
  --foreman-proxy-dhcp-interface eth0 \
  --foreman-proxy-dhcp-range  "192.168.2.100 192.168.2.150" \
  --foreman-proxy-dhcp-gateway 192.168.2.2 \
  --foreman-proxy-dhcp-nameservers 192.168.2.2 \
  --foreman-proxy-tftp true \
  --foreman-proxy-tftp-servername $(hostname) \
  --capsule-puppet true \
  --certs-server-cert \$(pwd)/2016-06-17.crt \
  --certs-server-cert-req \$(pwd)/2016-06-17.csr \
  --certs-server-key \$(pwd)/2016-06-17_unencrypted.key \
  --certs-server-ca-cert \$(pwd)/pcfe-CA-pem.crt
```

> **Note**: forwarders are stored in `/etc/named/options.conf` should you wish to change them later.
>
> **Note**: if you omit `--capsule-dhcp-nameservers` then a default nameservers option will be created in the dhcp config for the satellite servers own address, meaning that the DHCP server will advertise the satellite as the DNS server. If this is not what you want, you can add something like `--capsule-dhcp-nameservers 10.0.0.1, 10.0.0.2` to the main katello-installer options above

A full list of other katello-setup options are available via

```
katello-installer --help
```

Once the installer has finished, you should be able to login by pointing your browser to `https://<servername>` (assuming you have made the necessary firewall changes).

Initial Configuration, adding Red Hat Repos

The following sections will cover adding different repositories (or repos) to the Satellite server, starting with the official Red Hat repos. Adding other repos will be left until the next book

Each type of repo can either be syncronised on demand or alternatively can be scheuled to syncronise on a regular basis

Before we can add Red Hat repositories, we need to create an organisation and location then create and import a manifest file however.

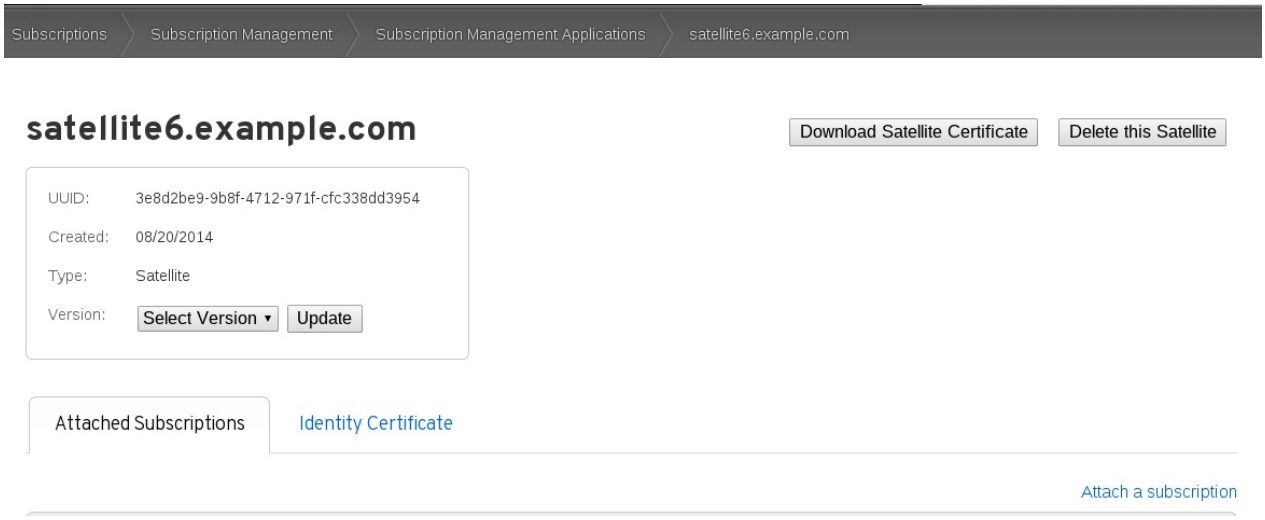
Creating the Manifest

Login to access.redhat.com to generate our manefest file.

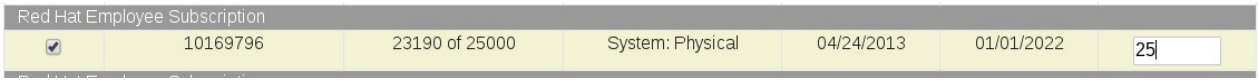
Locate the system, within access.redhat.com.

Be sure to re-set the **Version**, if it has become unset (as in the screenshot below) and hit **Update**

Then locate the "Attach a subscription" (bottom right in the screenshot below).



Navigate to the subscription you wish to add, select the checkbox on the left and enter a quantity on the right before selecting **Attach Selected**



It may take some time to attach the subscription, as it will say on the screen (see screenshot below).

Hit refresh from time to time until it completes and then click on the **Download Manifest** button and save the mainfest somewhere safe

Hammer Credentials

To run hammer you need to provide authentication, you can specify your username and password each time

```
hammer -u admin -p <password> <subcommands>
```

or simply create **~/.hammer/cli_config.yml**

```
:foreman:
    :enable_module: true
    :host: 'https://satellite.example.com/'
    :username: 'admin'
    :password: '<password>'
```

Personally I find the second option much more convenient,ne so I shall assume you have done this for all further examples

With the release of GA, the setup program was changed to randomise this password. The password is dumped to the screen, once the **katello-installer** has finished.

Just incase you didnt make a note of it, it can be recovered by running

```
awk '/^ *admin_password:/ { print $2 }' \
/etc/foreman-installer/scenarios.d/satellite-answers.yaml
```

Once you have edited this file, it might be worth `chmod 600`
the config file and stripping the password from the answers file just to safe

Quick Introduction to Hammer

The command line tool that is shipped with Satellite 6 is called Hammer

You can examime which commands Hammer is capable of with

`hammer --help`

You will see output line this

```
Usage:
    hammer [OPTIONS] SUBCOMMAND [ARG] ...

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

Subcommands:
    activation-key                Manipulate activation keys.
    architecture                  Manipulate architectures.
    capsule                       Manipulate capsule
    compute-resource              Manipulate compute resources.
    content-host                  manipulate content hosts on th
e server
    content-view                  Manipulate content views.
    domain                        Manipulate domains.
    environment                   Manipulate environments.
    fact                          Search facts.
    global-parameter              Manipulate global parameters.
    gpg                           manipulate GPG Key actions on
the server
    host                          Manipulate hosts.
    host-collection               Manipulate host collections
    hostgroup                     Manipulate hostgroups.

...
```

To get more specific help, you can be more specific with your request

```
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
    create                        Create an activation key
    host-collections              List associated host collectio
ns
    info                          Show an activation key
    list                          List activation keys
    remove-repository             Disassociate a resource
    remove-subscription           Remove subscription
    subscriptions                 List associated subscriptions
    update                        Update an activation key

Options:
    -h, --help                    print help
```

The more specific the request, the more specific the answer

```
hammer activation-key create --help
```

```
Usage:
    hammer activation-key create [OPTIONS]

Options:
    --content-view CONTENT_VIEW_NAME
    --content-view-id CONTENT_VIEW_ID content view id
    --description DESCRIPTION     description
    --environment ENVIRONMENT_NAME
    --environment-id ENVIRONMENT_ID environment id
    --name NAME                   name
    --organization ORGANIZATION_NAME
    --organization-id ORGANIZATION_ID organization identifier
    --organization-label ORGANIZATION_LABEL
    --usage-limit USAGE_LIMIT     maximum number of registered c
ontent hosts, or 'unlimited'
    -h, --help                    print help
```

Prepare for Manifest Import

Once the manifest has been created, we simple need to import it into our Satellite server.

However, first we must create our **Organization** and **Location**

Organisation

The oganisation that I shall use for this book is **Example Org** and the location I shall use is **Europe**

Lets get our orgainisation created, starting off with getting some help on the syntax

```
hammer organization create  --help
Usage:
    hammer organization create [OPTIONS]

Options:
    --description DESCRIPTION     description
    --label LABEL                 unique label
    --name NAME                   name
    -h, --help                    print help
```

So its probably enough to just specify the **--name** option

Lets setup some variables to use throughout the book, replace them with the values that are meaningful to you

```
ORG="Example Org"
LOC="Europe"
```

Ok, now we have those set, we need to get an organisation and a location set, before we import the manifest.

```
hammer organization create  --name "${ORG}"
Organization created
```

Location

Now the location

```
hammer location create --help
Usage:
    hammer location create [OPTIONS]

Options:
    --name NAME
    -h, --help                    print help
```

OK, that also looks simple then

```
hammer location create --name "${LOC}"
Location created
```

Congratulations, your first hammer commands have completed successfully. Remember to make good use of the **--help** option

Importing the Manifest

Earlier on we created and downloaded our manifest, now we should import it

A quick look at the help shows it should be easy

```
hammer subscription upload --help
Usage:
    hammer subscription upload [OPTIONS]

Options:
    --async                       Do not wait for the task
    --file MANIFEST               Subscription manifest file
    --organization ORGANIZATION_NAME
    --organization-id ORGANIZATION_ID Organization id
    --organization-label ORGANIZATION_LABEL
    --repository-url REPOSITORY_URL repository url
    -h, --help                    print help
```

So the following is all we need

```
hammer subscription upload --organization "${ORG}" --file <filen
ame here>
```

> **NOTE**: At the time of writing using the organisation **name** wasnt working for me, so I had to use **--organization-id** method instead. The ID was found by doing **hammer organization list**
>
> hammer subscription upload --organization-id 5 --file <filename>

Adding Red Hat Repositories

OK, once the manifest is imported we can look at enabling the Red Hat repositories to be downloaded to the Satellite server

The hammer command for enabling these Red Hat repositories is

```
hammer repository-set
```

There are a large number of repositories available, as can be seen by doing

```
hammer repository-set  list --product "Red Hat Enterprise Linux
Server" \
--organization  "${ORG}"
```

To get us started we shall only enable three Red Hat repositories - these are enough to enable us to perform provisioning

Example RHEL 6 Channels

```
# hammer repository-set enable  --organization "${ORG}" \
 --product "Red Hat Enterprise Linux Server" \
 --name "Red Hat Enterprise Linux 6 Server (Kickstart)" \
 --releasever "6.5" --basearch "x86_64"

# hammer repository-set enable  --organization "${ORG}" \
 --product "Red Hat Enterprise Linux Server" \
 --name "Red Hat Enterprise Linux 6 Server (RPMs)" \
 --releasever "6.5" --basearch "x86_64"

hammer repository-set enable  --organization "${ORG}"  \
 --product "Red Hat Enterprise Linux Server"  \
 --name "Red Hat Satellite Tools 6.2 (for RHEL 6 Server) (RPMs)"
 \
 --basearch "x86_64"

# hammer repository-set enable  --organization "${ORG}" \
 --product "Red Hat Enterprise Linux Server" \
 --name "Red Hat Enterprise Linux 6 Server - RH Common (RPMs)" \
 --basearch "x86_64"
```

Example RHEL 7 Channels

```
hammer repository-set enable  --organization "${ORG}" \
 --product "Red Hat Enterprise Linux Server" \
 --name "Red Hat Enterprise Linux 7 Server (Kickstart)" \
 --releasever "7.2" --basearch "x86_64"

hammer repository-set enable  --organization "${ORG}" \
 --product "Red Hat Enterprise Linux Server" \
 --name "Red Hat Enterprise Linux 7 Server (RPMs)" \
 --releasever "7Server" --basearch "x86_64"

hammer repository-set enable  --organization "${ORG}"  \
 --product "Red Hat Enterprise Linux Server"  \
 --name "Red Hat Satellite Tools 6.2 (for RHEL 7 Server) (RPMs)"
 \
 --basearch "x86_64"

hammer repository-set enable  --organization "${ORG}" \
 --product "Red Hat Enterprise Linux Server" \
 --name "Red Hat Enterprise Linux 7 Server - RH Common (RPMs)" \
 --releasever "7Server" --basearch "x86_64"
```

> **NOTE**: In Satellite 6.1, you will need to add the Satellite Tools repository but the RH Common repository can be removed (unless you use it for other things like the rhevm-guest-agent, in which case you will need both)

So now you have enabled s few repositories, but they are not syncronised. See the next section for this

Synchronising Repositories

Before we can create hosts to be provisioned, we need to synchronise the repositories that we selected in the previous section. This may take time, depending on the speed of your internet connection

Repository synchronisation is perfromed with

`hammer repository synchronize`

So in theory this should work

```
hammer repository synchronize --product "Red Hat Enterprise Linu
x Server" \
 --name "Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.1"
 \
 --organization "${ORG}"
```

However, once you have started creating content views, you may see errors due to the repository existing more than once

```
hammer repository synchronize --product "Red Hat Enterprise Linu
x Server" \
 --name "Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.1"
 \
 --organization "${ORG}"
Could not synchronize the repository:
  Error: repository found more than once
```

If we take a look at the repositories we have, we can confirm this

```
hammer repository list --organization "${ORG}"
---|----------------------------------------------------------------
-------|---------------------------------|-------------|-------
-----------------------------------------------------------------
----------
ID | NAME
       | PRODUCT                           | CONTENT TYPE | URL


---|----------------------------------------------------------------
-------|---------------------------------|-------------|-------
-----------------------------------------------------------------
----------
1  | Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.1
       | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7.1/x86_64/kickstart

4  | Red Hat Enterprise Linux 7 Server - RH Common RPMs x86_64 7
Server | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7Server/x86_64/rh-co
mmon/os
2  | Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server
       | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7Server/x86_64/os

3  | Red Hat Satellite Tools 6.1 for RHEL 7 Server RPMs x86_64
       | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7Server/x86_64/sat-t
ools/6....
---|----------------------------------------------------------------
-------|---------------------------------|-------------|-------
-----------------------------------------------------------------
----------
```

As a general rule, you should use the lowest ID for each duplicate and then you can synchronise via ID

```
hammer repository synchronize --id 1 --organization "${ORG}"
```

Unless you are happy to wait for it to finish you can also add the **--async** option to the command

```
 hammer repository synchronize --id 1 --organization "${ORG}" --
async
```

Lifecycle Environments

Satellite 6 has the concept of **Lifecycle Environments**. These should generally match the names of your tiers, such as **Crash, Development, QA, Production** etc etc

The idea is that your hosts or clients will exist in one of these tiers. A **Content View** describing how the host should be configured is defined and pushed or **promoted** to the first tier (Crash in our example) where it is tested and refined before it is promoted to the next environment for the next team to test.

We will discuss **Content Views** in more detail in a later section

For now lets go ahead and create the four **Lifecycle Environments** mentioned above

```
hammer lifecycle-environment create --name "Engineering" \
--description "For Engineering" --organization "${ORG}" \
--prior "Library"

hammer lifecycle-environment create --name "Development" \
--description "Initial testing for the App guys" \
--organization "${ORG}" --prior "Engineering"

hammer lifecycle-environment create --name "QA" \
--description "QA testing for the App guys" \
--organization "${ORG}" --prior "Development"

hammer lifecycle-environment create --name "Production" \
--description "Production Environment" \
--organization "${ORG}" --prior "QA"
```

Content Views

At this point we will have to wait for our Red Hat Repositories to be syncronised, before we define content views.

Content views contain RPM packages from repos as well as puppet modules from puppet repos. Everything needed in order to correctly configure the hosts

Once defined, your content views will be **published** and **promoted** to the appropriate **Lifecycle Environment**

Defining Content Views

Once the repositories that we need are syncronised, we can get our content view created. The content view will create a frozen view of the repositories until further updates are added to it and published.

OK, lets remind ourself of the IDs of our syncronised repositories, as its simpler to define the repositories we want to add to our **content view**, by ID

Again, to simplify cut and paste, lets define another variable

```
CV1="cv-rhel7-base"
```

```
hammer repository list --organization "${ORG}"
---|-------------------------------------------------------------|
---------------------------------|-------------|--------------
-------------------------------------------------------------
--
ID | NAME                                                        |
 PRODUCT                          | CONTENT TYPE | URL
---|-------------------------------------------------------------|
---------------------------------|-------------|--------------
-------------------------------------------------------------
--
3  | Red Hat Satellite Tools 6.2 for RHEL 7 Server RPMs x86_64 |
 Red Hat Enterprise Linux Server | yum          | https://cdn.re
dhat.com/content/dist/rhel/server/7/7Server/x86_64/sat-tools/6..
..
6  | Red Hat Satellite Tools 6.2 for RHEL 6 Server RPMs x86_64 |
 Red Hat Enterprise Linux Server | yum          | https://cdn.re
dhat.com/content/dist/rhel/server/6/6Server/x86_64/sat-tools/6..
..
2  | Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server     |
 Red Hat Enterprise Linux Server | yum          | https://cdn.re
dhat.com/content/dist/rhel/server/7/7Server/x86_64/os
1  | Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.2    |
 Red Hat Enterprise Linux Server | yum          | https://cdn.re
dhat.com/content/dist/rhel/server/7/7.2/x86_64/kickstart
5  | Red Hat Enterprise Linux 6 Server RPMs x86_64 6.5         |
 Red Hat Enterprise Linux Server | yum          | https://cdn.re
dhat.com/content/dist/rhel/server/6/6.5/x86_64/os
4  | Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.5    |
 Red Hat Enterprise Linux Server | yum          | https://cdn.re
dhat.com/content/dist/rhel/server/6/6.5/x86_64/kickstart
---|-------------------------------------------------------------|
---------------------------------|-------------|--------------
-------------------------------------------------------------
--
```

We dont need the kickstart repo once the anaconda has provisioned the machine, so we will exclude that and incluse the others

Lets create our **content view** now, called **cv-rhel7-base**

```
hammer content-view create --name "${CV1}" \
 --description "Our initial first content view" \
 --organization "${ORG}"
```

Next we add the repositories to the **content view**

```
hammer content-view update --repository-ids 3,2 \
 --name "${CV1}" --organization "${ORG}"
```

Finally we can publish our **content view**

```
hammer content-view publish --name "${CV1}" \
--organization "${ORG}"
```

NOTE: Its possible to also add the `--async` to this publish command if required

Capsules / Smart Proxies

Satellite 6 has the concept of **Capsules** which are analagous to **Smart Proxies** in Foreman

A **Capsule** provides functionality to the Satellite server. Examples of **Capsules** are

- DHCP Capsule - enabling Satellite 6 to reserve IP addresses on a DHCP server, including all the options necessary for a PXE boot
- DNS Capsule - enabling the Satellite to create, update and remove forward and reverse DNS records
- Realm Capsule - enabling Satellite to create Kerberos Host Principles on a Kerberos Server
- TFTP Capsule - enabling the Satellite server to place files required for PXE booting a Host
- Puppet Capsule - Providing Puppet functionality to Satellite (usually the Satellite server itself)

These are usually, but not always, on remote servers and not on the main Satellite server itself. However, that said, in this introductory session we configured our Satellite to have multiple local **Capsules**.

We chose to run TFTP,DHCP & DNS **Capsules** on our main Satellite server, during the initial configuration.

Capsules can be used with `hammer capsule` subcommands

```
hammer capsule --help
Usage:
    hammer capsule [OPTIONS] SUBCOMMAND [ARG] ...

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

Subcommands:
    content                       Manage the capsule content
    info                          Show the capsule details
    list                          List all capsules

Options:
    -h, --help                    print help
```

Further discussion of **capsules** I shall save for the next book.

Configure the Server for Provisioning

There are a number of things that need to be defined before we can add a new host to be provisioned.

The following items need to be defined

- Architecture
- Domain
- Activation Key
- Partition Table
- Subnet
- DHCP Proxy
- DNS Proxy
- Realm Proxy
- TFTP Proxy
- Provisioning Templates
- Operating Systems

Each will be discussed in its own section

Architectures

Double check that the **Architectures** are created already

```
hammer  architecture list
---|-------
ID | NAME
---|-------
2  | i386
1  | x86_64
---|-------
```

As **architectures** dont cant be allocated to a **location** or **organisation** there is nothing to do here other than verify that the configure script created them

Domains

Also verify that your first **Domain** was created by the initial configuration task

```
hammer domain list
---|------------
ID | NAME
---|------------
1  | example.com
---|------------
```

Domains can and indeed should be part of your **location** and **organisation**, so we shall probably have to move it. We can do this with

```
hammer location add-domain --domain "example.com" --name "${LOC}
"
```

Next the **location**

```
hammer organization add-domain --domain "example.com" --name "${
ORG}"
```

Finally we can verify that this has worked, using the ID we can see in step 1

```
hammer domain info --id 1
Id:             1
Name:           example.com
Description:
DNS Id:
Subnets:

Locations:
    *** your location
Organizations:
    *** your org name
Parameters:

Created at:     2014/09/18 17:20:41
Updated at:     2014/09/18 17:20:41
```

Make a mental note that there is no **subnet** assigned to this domain so far, we shall take care of this later

Activation Keys

Activation keys are required, when registering the host, in order to apply the correct settings to the host

> **NOTE** Unfortunatley the version of subscription manager shipped in RHEL6.5 (and below) does not function correctly with Activation Keys. The default **subscription_manager_registration** snippet has a fix to ensure that the RH Common repo is included, so that provisioning should work fine. RHEL6.6 and RHEL7 do work correctly.

NOTE: We need to wait for our **content view** to publish and have an activation key created, before we continue

Once again we will use a variable to aid copy and pasting

```
AK1="ak-rhel7-base-1"
```

```
hammer activation-key create --name "${AK1}" \
 --content-view "${CV1}" --lifecycle-environment Library \
 --organization "${ORG}"

Activation key created
```

Even when using the UI, its easy to miss this step.

```
hammer activation-key update --release-version "7Server"  \
 --organization "${ORG}" --name "${AK1}"
Activation key updated
```

Now the key is created but needs additional configuration such as adding **Subscriptions**

This step requires some interim steps to find the **IDs** we need. First step is to list the available **Subscriptions**. We need the **ID** (the last but one column)

```
hammer subscription list --organization "${ORG}"
```

We also need to list the **activation keys** as we need the **ID** from that also

```
hammer activation-key list --organization "${ORG}"
```

Then using the result from the **ID** columns, attach the subscription

```
hammer activation-key add-subscription --id <activation key ID>
\
--subscription-id <subscription ID>
```

Partition Tables

Check that your **Partition Table** is associated with your **Operating System**. This should have been done during the initial configuration, but when you create your own, custom ones, dont forget this step

```
hammer partition-table list
---|------------------------------|----------
ID | NAME                         | OS FAMILY
---|------------------------------|----------
1  | AutoYaST entire SCSI disk    | Suse
2  | AutoYaST entire virtual disk | Suse
3  | AutoYaST LVM                 | Suse
4  | FreeBSD                      | Freebsd
5  | Jumpstart default            | Solaris
6  | Jumpstart mirrored           | Solaris
10 | Junos default fake           | Junos
7  | Kickstart default            | Redhat
9  | Preseed custom LVM           | Debian
8  | Preseed default              | Debian
---|------------------------------|----------
```

Get some more info

```
hammer partition-table info --id 7
Id:                 7
Name:               Kickstart default
OS Family:          Redhat
Operating systems:
    RedHat 7.2
    RedHat 7.1
Created at:         2015/12/07 09:31:13
Updated at:         2015/12/07 09:31:13
```

We will fix this later but just FYI, you would use `hammer os add-ptable --id 1 --ptable-id 7` but as I say we will do it later

Subnets

We need to heck that the initial configuration has created your subnet **and** that its in the correct **Organisation** and **Location**

```
hammer subnet list
---|-------------|-------------|--------------
ID | NAME        | NETWORK     | MASK
---|-------------|-------------|--------------
1  | 172.16.30.0 | 172.16.30.0 | 255.255.255.0
---|-------------|-------------|--------------
```

You can either use the **ID** or the **Name** to see if your subnet is in the correct **Organisation** and **Location**

```
hammer subnet info --id 1
Id:             1
Name:           172.16.30.0
Network:        172.16.30.0
Mask:           255.255.255.0
Priority:
DNS:            satellite6.example.com (https://satellite6.exampl
e.com:9090)
Primary DNS:    172.16.30.200
Secondary DNS: 10.0.0.9
TFTP:           satellite6.example.com (https://satellite6.exampl
e.com:9090)
DHCP:           satellite6.example.com (https://satellite6.exampl
e.com:9090)
VLAN ID:
Gateway:        172.16.30.1
From:           172.16.30.100
To:             172.16.30.150
Domains:
    example.com
Locations:
    *** Your location
Organizations:
    *** Your Organisation
```

If your subnet doesnt exist, use hammer to create it

```
hammer subnet create --help
Usage:
    hammer subnet create [OPTIONS]

Options:
    --dhcp-id DHCP_ID             DHCP Proxy to use within this
subnet
    --dns-id DNS_ID               DNS Proxy to use within this s
ubnet
    --dns-primary DNS_PRIMARY     Primary DNS for this subnet
    --dns-secondary DNS_SECONDARY Secondary DNS for this subnet
    --domain-ids DOMAIN_IDS       Domains in which this subnet i
s part
                                  Comma separated list of values
.
    --from FROM                   Starting IP Address for IP aut
o suggestion
    --gateway GATEWAY             Primary DNS for this subnet
    --mask MASK                   Netmask for this subnet
    --name NAME                   Subnet name
    --network NETWORK             Subnet network
    --tftp-id TFTP_ID             TFTP Proxy to use within this
subnet
    --to TO                       Ending IP Address for IP auto
suggestion
    --vlanid VLANID               VLAN ID for this subnet
    -h, --help                    print help
```

If it does exist but is in the wrong **Organisation** or **Location** then use hammer to move it

```
hammer organization add-subnet --help
Usage:
    hammer organization add-subnet [OPTIONS]

Options:
    --id ID
    --name NAME                    Organization name
    --subnet SUBNET_NAME           Subnet name
    --subnet-id SUBNET_ID
    -h, --help                     print help
```

and

```
hammer location add-subnet --help
Usage:
    hammer location add-subnet [OPTIONS]

Options:
    --id ID
    --name NAME                    Location name
    --subnet SUBNET_NAME           Subnet name
    --subnet-id SUBNET_ID
    -h, --help                     print help
```

If your subnet isn't even created, then we shall have to do it manually, lets get some help

```
hammer subnet create --help
Usage:
    hammer subnet create [OPTIONS]

Options:
    --dhcp-id DHCP_ID             DHCP Proxy to use within this
subnet
    --dns-id DNS_ID               DNS Proxy to use within this s
ubnet
    --dns-primary DNS_PRIMARY     Primary DNS for this subnet
    --dns-secondary DNS_SECONDARY Secondary DNS for this subnet
    --domain-ids DOMAIN_IDS       Domains in which this subnet i
s part
                                  Comma separated list of values
.
    --from FROM                   Starting IP Address for IP aut
o suggestion
    --gateway GATEWAY             Primary DNS for this subnet
    --mask MASK                   Netmask for this subnet
    --name NAME                   Subnet name
    --network NETWORK             Subnet network
    --tftp-id TFTP_ID             TFTP Proxy to use within this
subnet
    --to TO                       Ending IP Address for IP auto
suggestion
    --vlanid VLANID               VLAN ID for this subnet
    -h, --help                    print help
```

We will need most of these settings, we also need to know the ID of the **Capules/Smart Proxies** we installed to take care of DNS & DHCP

```
hammer proxy list
---|------------------------------|------------------------------
------------|--------------------------
ID | NAME                         | URL
            | FEATURES
---|------------------------------|------------------------------
------------|--------------------------
1  | satellite6.example.com | https://satellite6.example.com:909
0 | TFTP, DNS, DHCP, Puppe...
---|------------------------------|------------------------------
------------|--------------------------
```

We also need to verify that the **domain ID** before we create the subnet

```
hammer domain list
---|------------
ID | NAME
---|------------
1  | example.com
---|------------
```

OK, we have enough information to create the subnet

```
hammer subnet create --dhcp-id 1 --dns-id 1  \
 --dns-primary 172.16.30.250 --domain-ids 1 \
 --from "172.16.30.100" --gateway "172.16.30.1"\
 --mask "255.255.255.0" --name "172.16.30.0/24"\
 --network "172.16.30.0" --tftp-id 1 --to "172.16.30.199"
```

OK, that created the subnet, now we need to associate it with our **location** and **organisation**. The command is very similar to the one we used for our **domain** earlier. Check out the syntax

```
 hammer location add-subnet --help
Usage:
    hammer location add-subnet [OPTIONS]

Options:
    --id ID
    --name NAME                    Location name
    --subnet SUBNET_NAME           Subnet name
    --subnet-id SUBNET_ID
    -h, --help                     print help
```

So lets associate it

```
hammer subnet update --name "172.16.0.0/24" --locations "${LOC}"
 --organizations  "${ORG}"
```

Finally, lets check that every thing has worked as we expected

```
hammer subnet info --id 1
Id:             1
Name:           172.16.30.0
Network:        172.16.30.0
Mask:           255.255.255.0
Priority:
DNS:            satellite6.example.com (https://satellite6.exampl
e.com:9090)
Primary DNS:    172.16.30.200
Secondary DNS:
TFTP:           satellite6.example.com (https://satellite6.exampl
e.com:9090)
DHCP:           satellite6.example.com (https://satellite6.exampl
e.com:9090)
VLAN ID:
Gateway:        172.16.30.1
From:           172.16.30.100
To:             172.16.30.199
Domains:
    example.com
Locations:
    *** Your location
Organizations:
    *** Your organisation
```

Provisioning Templates

One of the changes from the beta version is that now, copies of provisioning templates are copied to your location and organisation, but they are read only copies.

This is a nice last minute change (from the beta) as editing one template no longer affects other orgs.

If you want to change one of them, then you will need to clone it

> **NOTE**: At the time of writing (just after GA, hammer still had not been updated to include this clone functionality. If you need to clone a template, you will need to use the UI Bugzilla 1160292

You can get a list of **Provisioning Templates** by doing the following (note that this command produces paged output by defualt, so Ive used the **--per-page 9999** option)

```
hammer template list  --per-page 9999
---|--------------------------------------|----------
ID | NAME                                 | TYPE
---|--------------------------------------|----------
5  | Alterator default                    | provision
6  | Alterator default finish             | finish
7  | Alterator default PXELinux           | PXELinux
34 | alterator_pkglist                    | snippet
8  | AutoYaST default                     | provision
10 | AutoYaST default PXELinux            | PXELinux
9  | AutoYaST SLES default                | provision
44 | Boot disk iPXE - generic host        | Bootdisk
43 | Boot disk iPXE - host                | Bootdisk
35 | epel                                 | snippet
36 | fix_hosts                            | snippet
11 | FreeBSD (mfsBSD) finish              | finish
12 | FreeBSD (mfsBSD) provision           | provision
13 | FreeBSD (mfsBSD) PXELinux            | PXELinux
37 | freeipa_register                     | snippet
```

```
 14 | Grubby default                          | script
 38 | http_proxy                              | snippet
 48 | idm_register                            | snippet
 15 | Jumpstart default                       | provision
 16 | Jumpstart default finish                | finish
 17 | Jumpstart default PXEGrub               | PXEGrub
 33 | Junos default finish                    | finish
 31 | Junos default SLAX                      | provision
 32 | Junos default ZTP config                | ZTP
 18 | Kickstart default                       | provision
 20 | Kickstart default finish                | finish
 22 | Kickstart default iPXE                  | iPXE
 21 | Kickstart default PXELinux              | PXELinux
 23 | Kickstart default user data             | user_data
 39 | kickstart_networking_setup              | snippet
 19 | Kickstart RHEL default                  | provision
 24 | Preseed default                         | provision
 25 | Preseed default finish                  | finish
 27 | Preseed default iPXE                    | iPXE
 26 | Preseed default PXELinux                | PXELinux
 28 | Preseed default user data               | user_data
 40 | puppet.conf                             | snippet
 4  | PXEGrub default local boot              | PXEGrub
 2  | PXELinux default local boot             | PXELinux
 3  | PXELinux default memdisk                | PXELinux
 1  | PXELinux global default                 | PXELinux
 41 | redhat_register                         | snippet
 42 | saltstack_minion                        | snippet
 45 | Satellite Kickstart Default             | provision
 47 | Satellite Kickstart Default Finish      | finish
 46 | Satellite Kickstart Default User Data | user_data
 49 | subscription_manager_registration       | snippet
 29 | UserData default                        | user_data
 30 | WAIK default PXELinux                   | PXELinux
----|-----------------------------------------|----------
```

The two that we require for provisioning are **Kickstart default PXELinux** and **Satellite Kickstart Default**. The later brings in the **subscription_manager_registration** snippet also

Lets stick with the builtin provisioning template. Lets see what we need to do.

```
hammer template info --id 45
Id:                 45
Name:               Satellite Kickstart Default
Type:               provision
Operating systems:
    RedHat 7.2
    RedHat 7.1
Locations:
    Default_Location
    *** Your location
Organizations:
    Default_Organization
    *** Your organisation
```

Associate the PXE Linux Template with the OS

The kickstart template was take care of above, now we need to make sure that the PXE template is associated with the OS correctly

```
# hammer template info --id 21
Id:                 21
Name:               Kickstart default PXELinux
Type:               PXELinux
Operating systems:
    RedHat 7.2
    RedHat 7.1
Locations:
    Default Location
    *** Your location
Organisations:
    Default Organization
    *** Your organisation
```

This page is significantly short than the 6.0 one,as many of the 6.0 bugs have been resolved in 6.1

Operating Systems

There are a number of things that need to be set on the **Operating System**

Make sure that the following items have values

- Partition tables
- Default templates:
- Architectures:
- Installation media

```
hammer os list
---|------------|--------------|-------
ID | TITLE      | RELEASE NAME | FAMILY
---|------------|--------------|-------
1  | RedHat 7.2 |              | Redhat
2  | RedHat 7.1 |              | Redhat
---|------------|--------------|-------
```

```
hammer os info --id 1
Id:                 1
Title:              RedHat 7.2
Release name:
Family:             Redhat
Name:               RedHat
Major version:      7
Minor version:      2
Partition tables:
    Kickstart default
Default templates:
    Kickstart default PXELinux (PXELinux)
    Kickstart default iPXE (iPXE)
    Satellite Kickstart Default (provision)
    Satellite Kickstart Default Finish (finish)
    Satellite Kickstart Default User Data (user_data)
Architectures:

Installation media:

Templates:
    Kickstart default iPXE (iPXE)
    Kickstart default PXELinux (PXELinux)
    Satellite Kickstart Default (provision)
    Satellite Kickstart Default Finish (finish)
    Satellite Kickstart Default User Data (user_data)
Parameters:
```

Again, this page is much short in 6.1 as many of the additional steps required in 6.0 are no longer required

Installation Media

Next item to address is the fact that the installation media currently doent have your location/organisation set correctly

In order to take care of this, get a list of *installation media*

```
hammer medium list
```

Make of note of the ID of the media you want more info on

```
hammer medium info --id 7
Id:                 7
Name:               Example_Org/Library/Red_Hat_Server/Red_Hat_En
terprise_Linux_7_Server_Kickstart_x86_64_7_1
Path:               http://satellite.example.com/pulp/repos/Examp
le_Org/Library/content/dist/rhel/server/7/7.1/x86_64/kickstart/
OS Family:          Redhat
Operating systems:
    RedHat 7.1
Organisations:
    Example Org
Created at:         2015/12/07 10:11:21
Updated at:         2015/12/07 10:11:21
```

Looks like the organisation **is** set but the lociation isnt, so we will correct that

```
hammer location add-medium --medium-id 7 --name "${LOC}"
```

Final check

```
hammer medium info --id 7
Id:                 7
Name:               Example_Org/Library/Red_Hat_Server/Red_Hat_En
terprise_Linux_7_Server_Kickstart_x86_64_7_1
Path:               http://satellite.example.com/pulp/repos/Examp
le_Org/Library/content/dist/rhel/server/7/7.1/x86_64/kickstart/
OS Family:          Redhat
Operating systems:
    RedHat 7.1
Locations:
    Europe
Organisations:
    Example Org
Created at:         2015/12/07 10:11:21
Updated at:         2015/12/07 10:11:21
```

Looks like I forgot to get the 7.2 kickstart as well, so lets add that repo and get it syncronised

```
hammer repository-set enable  --organization "${ORG}"  --product
 "Red Hat Enterprise Linux Server"  --name "Red Hat Enterprise L
inux 7 Server (Kickstart)"  --releasever "7.2" --basearch "x86_6
4"
 Repository enabled

hammer repository list --organization "${ORG}"
---|-----------------------------------------------------------
-------|---------------------------------|-------------|-------
-------------------------------------------------------------
----------
ID | NAME
       | PRODUCT                          | CONTENT TYPE | URL


---|-----------------------------------------------------------
-------|---------------------------------|-------------|-------
-------------------------------------------------------------
----------
1  | Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.1
```

```
       | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7.1/x86_64/kickstart

11 | Red Hat Enterprise Linux 7 Server Kickstart x86_64 7.2
       | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7.2/x86_64/kickstart

4  | Red Hat Enterprise Linux 7 Server - RH Common RPMs x86_64 7
Server | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7Server/x86_64/rh-co
mmon/os
2  | Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server
       | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7Server/x86_64/os

3  | Red Hat Satellite Tools 6.2 for RHEL 7 Server RPMs x86_64
       | Red Hat Enterprise Linux Server | yum          | https:
//cdn.redhat.com/content/dist/rhel/server/7/7Server/x86_64/sat-t
ools/6....
---|-----------------------------------------------------------
-------|---------------------------------|--------------|-------
-------------------------------------------------------------
----------

hammer repository synchronize --id 11 --organization "${ORG}"
```

We will also need to add the location to this one

```
hammer location add-medium --medium-id 8 --name "${LOC}"

hammer medium info --id 8
Id:                 8
Name:               Example_Org/Library/Red_Hat_Server/Red_Hat_En
terprise_Linux_7_Server_Kickstart_x86_64_7_2
Path:               http://satellite.example.com/pulp/repos/Examp
le_Org/Library/content/dist/rhel/server/7/7.2/x86_64/kickstart/
OS Family:          Redhat
Operating systems:
    RedHat 7.2
Locations:
    Europe
Organisations:
    Example Org
Created at:         2015/12/07 13:08:54
Updated at:         2015/12/07 13:08:54
```

Puppet Environment - Location

Now we need to make sure that the **puppet environment** is assigned to our **location**. So lets list the environments

```
hammer environment list
---|--------------------------------------
ID | NAME
---|--------------------------------------
2  | KT_Example_Org_Library_cv_rhel7_base_3
1  | production
---|--------------------------------------
```

Now to get more info about our puppet environment

```
# hammer environment info --id 2
id:            2
Name:          KT_Example_Org_Library_cv_rhel7_base_3
Puppetclasses:
    access_insights_client
Locations:
    Default Location
Organisations:
    Example Org
Created at:    2015/12/07 12:23:58
Updated at:    2015/12/07 12:23:58
```

OK, so we can see that the location isnt set (its only in the **Default Location** right now), lets fix that

```
# hammer environment update --help
Usage:
    hammer environment update [OPTIONS]

Options:
 --id ID
 --location-ids LOCATION_IDS         Comma separated list of val
ues.
 --locations LOCATION_NAMES          Comma separated list of val
ues.
 --name NAME                         Environment name
 --new-name NEW_NAME
 --organization-ids ORGANIZATION_IDS organization ID
                                     Comma separated list of val
ues.
 --organizations ORGANIZATION_NAMES  Comma separated list of val
ues.
 -h, --help                          print help
```

So that looks straighforward then

```
# hammer environment update --id 2 --locations "${LOC}"
Environment updated
```

Lets check

```
hammer environment info --id 2
Id:            2
Name:          KT_Example_Org_Library_cv_rhel7_base_3
Puppetclasses:
    access_insights_client
Locations:
    Europe
Organisations:
    Example Org
Created at:    2015/12/07 12:23:58
Updated at:    2015/12/07 12:23:58
```

Looks good now

Host Groups

Rather than applying multiple settings to an individual host, Satellite 6 uses hostgroups.

Multiple **HostGroups** can be added and they can also be nested in a hierarchical manner.

Examples

```
DC North
    Applications
    Database
        PostgSQL
        MariaDB
    Infrastructure
    Online

DC South
    Applications
    Database
        Oracle
        Mongo
    Infrastructure
    Online
        Webservers
```

Configuration can be applied at any level.

For example, your DNS servers can be set at the top level (DC North/DC South) where as mailserver settings may be defined lower down the hierachy

Defining Host Groups

This page has been entirely re-written in the update to 6.1, nearly all of the work arounds are now not required thankfully.

```
hammer hostgroup create --help
Usage:
    hammer hostgroup create [OPTIONS]

Options:
 --architecture ARCHITECTURE_NAME                         Architect
ure name
 --architecture-id ARCHITECTURE_ID
 --content-source-id CONTENT_SOURCE_ID
 --content-view CONTENT_VIEW_NAME                         Name to s
earch by
 --content-view-id CONTENT_VIEW_ID                        content v
iew numeric identifier
 --domain DOMAIN_NAME                                     Domain na
me
 --domain-id DOMAIN_ID                                    Numerical
 ID or domain name
 --environment ENVIRONMENT_NAME                           Environme
nt name
 --environment-id ENVIRONMENT_ID
 --lifecycle-environment LIFECYCLE_ENVIRONMENT_NAME       Name to s
earch by
 --lifecycle-environment-id LIFECYCLE_ENVIRONMENT_ID      ID of the
 environment
 --location-ids LOCATION_IDS                              Comma sep
arated list of values.
 --locations LOCATION_NAMES                               Comma sep
arated list of values.
 --medium MEDIUM_NAME                                     Medium na
me
 --medium-id MEDIUM_ID
 --name NAME
 --operatingsystem OPERATINGSYSTEM_TITLE                  Operating
```

```
 system title
 --operatingsystem-id OPERATINGSYSTEM_ID
 --organization-ids ORGANIZATION_IDS                    organizat
ion ID
                                                        Comma sep
arated list of values.
 --organizations ORGANIZATION_NAMES                     Comma sep
arated list of values.
 --parent PARENT_NAME                                   Name of p
arent hostgroup
 --parent-id PARENT_ID
 --partition-table, --ptable PARTITION_TABLE_NAME       Partition
 table name
 --partition-table-id, --ptable-id PARTITION_TABLE_ID
 --puppet-ca-proxy PUPPET_CA_PROXY_NAME                 Name of p
uppet CA proxy
 --puppet-ca-proxy-id PUPPET_CA_PROXY_ID
 --puppet-class-ids, --puppetclass-ids PUPPETCLASS_IDS List of p
uppetclass ids
                                                        Comma sep
arated list of values.
 --puppet-classes PUPPET_CLASS_NAMES                    Comma sep
arated list of values.
 --puppet-proxy PUPPET_PROXY_NAME                       Name of p
uppet proxy
 --puppet-proxy-id PUPPET_PROXY_ID
 --realm REALM_NAME                                     Name to s
earch by
 --realm-id REALM_ID                                    Numerical
 ID or realm name
 --subnet SUBNET_NAME                                   Subnet na
me
 --subnet-id SUBNET_ID
 -h, --help                                             print hel
p
```

We will create a host group called **"DC North"** in the **Library** - **Lifecycle Environment**. From the previous section, we know that our **--operatingsystem-id** is **1** and the **--partition-table** is **"Kickstart default"**

```
hammer hostgroup create --name "DC North" \
 --architecture "x86_64" --domain "example.com"\
 --environment "KT_Example_Org_Library_cv_rhel7_base_3"\
 --medium "Example_Org/Library/Red_Hat_Server/Red_Hat_Enterprise
_Linux_7_Server_Kickstart_x86_64_7_2"\
 --operatingsystem-id 1 --partition-table "Kickstart default"\
 --puppet-ca-proxy-id 1 --puppet-proxy-id 1 \
 --subnet "172.16.0.0/24" --content-source-id "1" \
 --organizations "${ORG}"  --locations "${LOC}" \
 --lifecycle-environment "Library" --content-view "${CV1}"
```

There are two things we cant do with the above command

- Set the activation key
- Set the root password

To fix the first one

```
hammer hostgroup set-parameter --hostgroup "DC North"  --name "k
t_activation_keys"   --value "${AK1}"
```

Unfortunately the second one (setting the root password at the **host group** level still cant be done, although it looks like the code has been merged upstream and so is hopefully coming to Satellite soon.

We will just live with this for now and set the root password at the **host** creation time, in the next section

Host Creation

Just as for the previous section, this page is now dramatically simpler, the workarounds have been removed, as in Satellite 6.1, they are no longer needed as there are many more options in **hammer host create**

```
hammer host create --help

Usage:
    hammer host create [OPTIONS]

Options:
 --architecture ARCHITECTURE_NAME                         Architec
ture name
 --architecture-id ARCHITECTURE_ID
 --ask-root-password ASK_ROOT_PW                          One of t
rue/false, yes/no, 1/0.
 --build BUILD                                            One of t
rue/false, yes/no, 1/0.
                                                          Default:
 "true"
 --comment COMMENT                                        Addition
al information about this host
 --compute-attributes COMPUTE_ATTRS                       Compute
resource attributes.
                                                          Comma-se
parated list of key=value.
 --compute-profile COMPUTE_PROFILE_NAME                   Name to
search by
 --compute-profile-id COMPUTE_PROFILE_ID
 --compute-resource COMPUTE_RESOURCE_NAME                 Compute
resource name
 --compute-resource-id COMPUTE_RESOURCE_ID
 --domain DOMAIN_NAME                                     Domain n
ame
 --domain-id DOMAIN_ID                                    Numerica
l ID or domain name
```

```
 --enabled ENABLED                                          One of t
rue/false, yes/no, 1/0.
                                                            Default:
 "true"
 --environment ENVIRONMENT_NAME                             Environm
ent name
 --environment-id ENVIRONMENT_ID
 --hostgroup HOSTGROUP_NAME                                 Hostgrou
p name
 --hostgroup-id HOSTGROUP_ID
 --image IMAGE_NAME                                         Name to
search by
 --image-id IMAGE_ID
 --interface INTERFACE                                      Interfac
e parameters.
                                                            Comma-se
parated list of key=value.
                                                            Can be s
pecified multiple times.
 --ip IP                                                    not requ
ired if using a subnet with DHCP proxy
 --location LOCATION_NAME                                   Location
 name
 --location-id LOCATION_ID
 --mac MAC                                                  required
 for managed host that is bare metal, not required if it's a
                                                            virtual
machine
 --managed MANAGED                                          One of t
rue/false, yes/no, 1/0.
                                                            Default:
 "true"
 --medium MEDIUM_NAME                                       Medium n
ame
 --medium-id MEDIUM_ID
 --model MODEL_NAME                                         Model na
me
 --model-id MODEL_ID
 --name NAME
 --operatingsystem OPERATINGSYSTEM_TITLE                    Operatin
```

```
g system title
 --operatingsystem-id OPERATINGSYSTEM_ID
 --organization ORGANIZATION_NAME                          Organisa
tion name
 --organization-id ORGANIZATION_ID                         organiza
tion ID
 --owner OWNER_LOGIN                                       Login of
 the owner
 --owner-id OWNER_ID                                       ID of th
e owner
 --owner-type OWNER_TYPE                                   Host's o
wner type
                                                           Possible
 value(s): 'User', 'Usergroup'
 --parameters PARAMS                                       Host par
ameters.
                                                           Comma-se
parated list of key=value.
                                                           Comma-se
parated list of key=value.
 --partition-table, --ptable PARTITION_TABLE_NAME          Partitio
n table name
 --partition-table-id, --ptable-id PARTITION_TABLE_ID
 --progress-report-id PROGRESS_REPORT_ID                   UUID to
track orchestration tasks status, GET
                                                           /api/orc
hestration/:UUID/tasks
 --provision-method METHOD                                 Possible
 value(s): 'build', 'image'
 --puppet-ca-proxy PUPPET_CA_PROXY_NAME
 --puppet-ca-proxy-id PUPPET_CA_PROXY_ID
 --puppet-class-ids, --puppetclass-ids PUPPET_CLASS_IDS Comma se
parated list of values.
 --puppet-classes PUPPET_CLASS_NAMES                       Comma se
parated list of values.
 --puppet-proxy PUPPET_PROXY_NAME
 --puppet-proxy-id PUPPET_PROXY_ID
 --realm REALM_NAME                                        Name to
search by
 --realm-id REALM_ID                                       Numerica
```

```
l ID or realm name
 --root-pass ROOT_PASS                                      required
 if host is managed and value is not inherited from host group o
r
                                                            default
password in settings
 --root-password ROOT_PW
 --subnet SUBNET_NAME                                       Subnet n
ame
 --subnet-id SUBNET_ID
 --volume VOLUME                                            Volume p
arameters
                                                            Comma-se
parated list of key=value.
                                                            Can be s
pecified multiple times.
 -h, --help                                                 print he
lp
```

As also mentioned in the previous section, setting of the root password at the **host-group** level is not possible, so we will set it here at the host creation stage (we can even have it prompt us for the password, of you dont want such things in your bash history)

All we really need now, for bare metal provisioning (physical or virtual without compute resources configured) is a **mac-address**

```
hammer host create --hostgroup "DC North" --name="satellite-prov
ision-test" \
 --mac "00:1a:4a:16:01:7a" --root-password "redhat00" \
 --organization "${ORG}" --location "${LOC}"
Host created
```

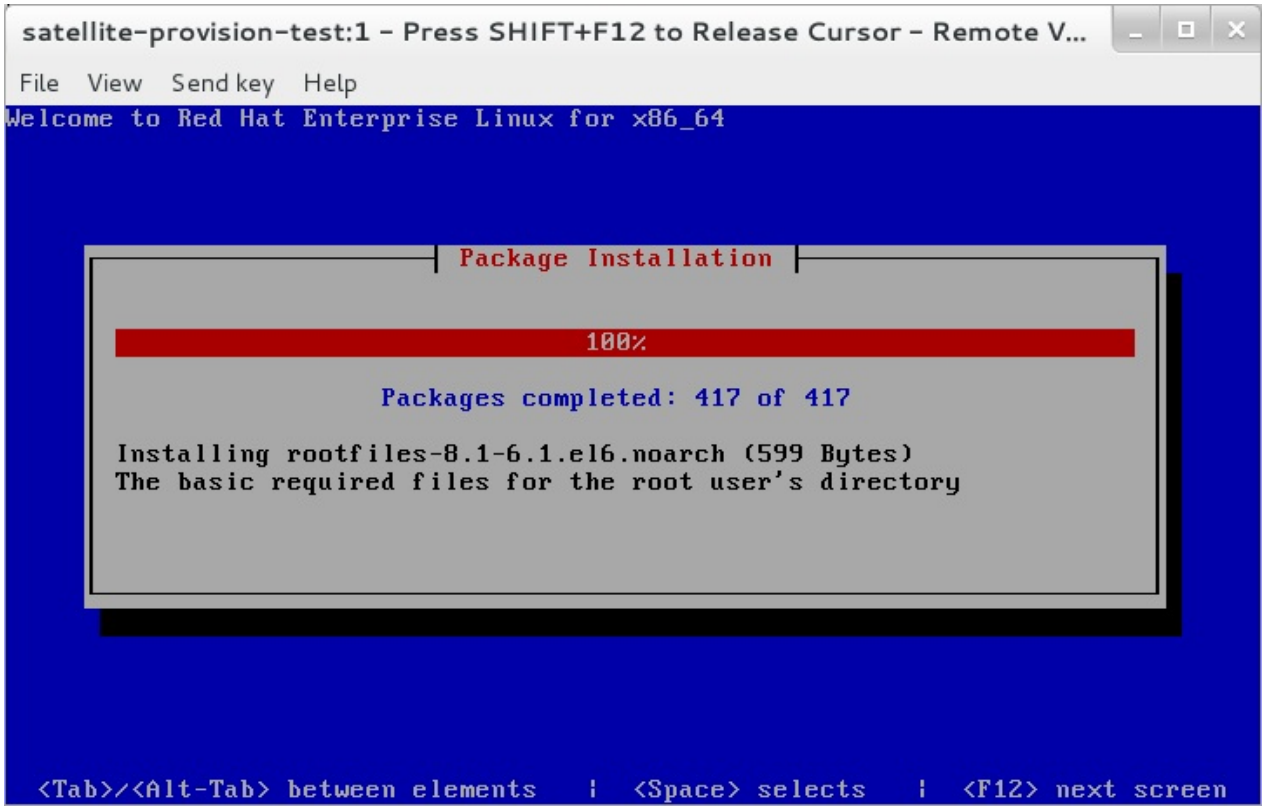
or get it to prompt us for the password

```
hammer host create --hostgroup "DC North" --name="satellite-prov
ision-test" \
 --mac "00:1a:4a:16:01:7a" --ask-root-password yes \
 --organization "${ORG}" --location "${LOC}"
Enter the root password for the host:
Host created
```

Now power on the host to be provisioned.

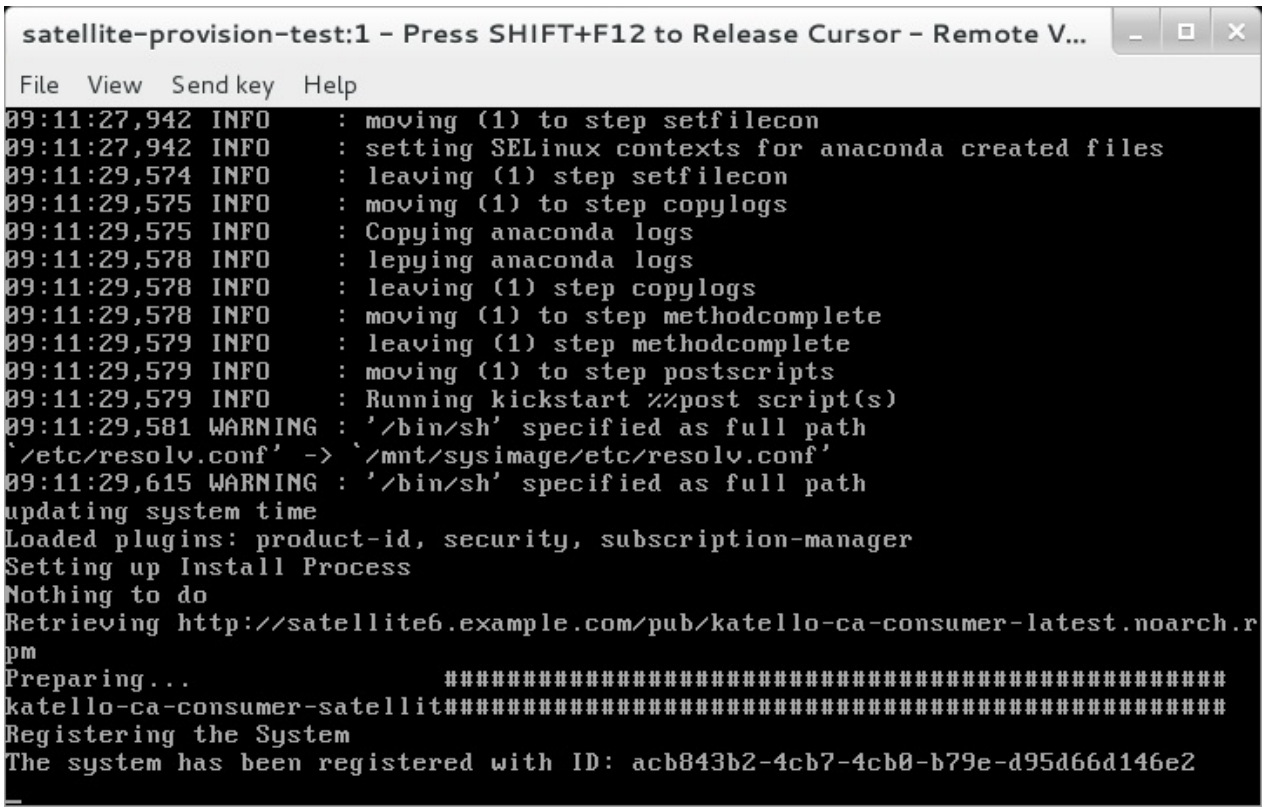
The build should progress in these distinct stages

The initial Anaconda package install stage

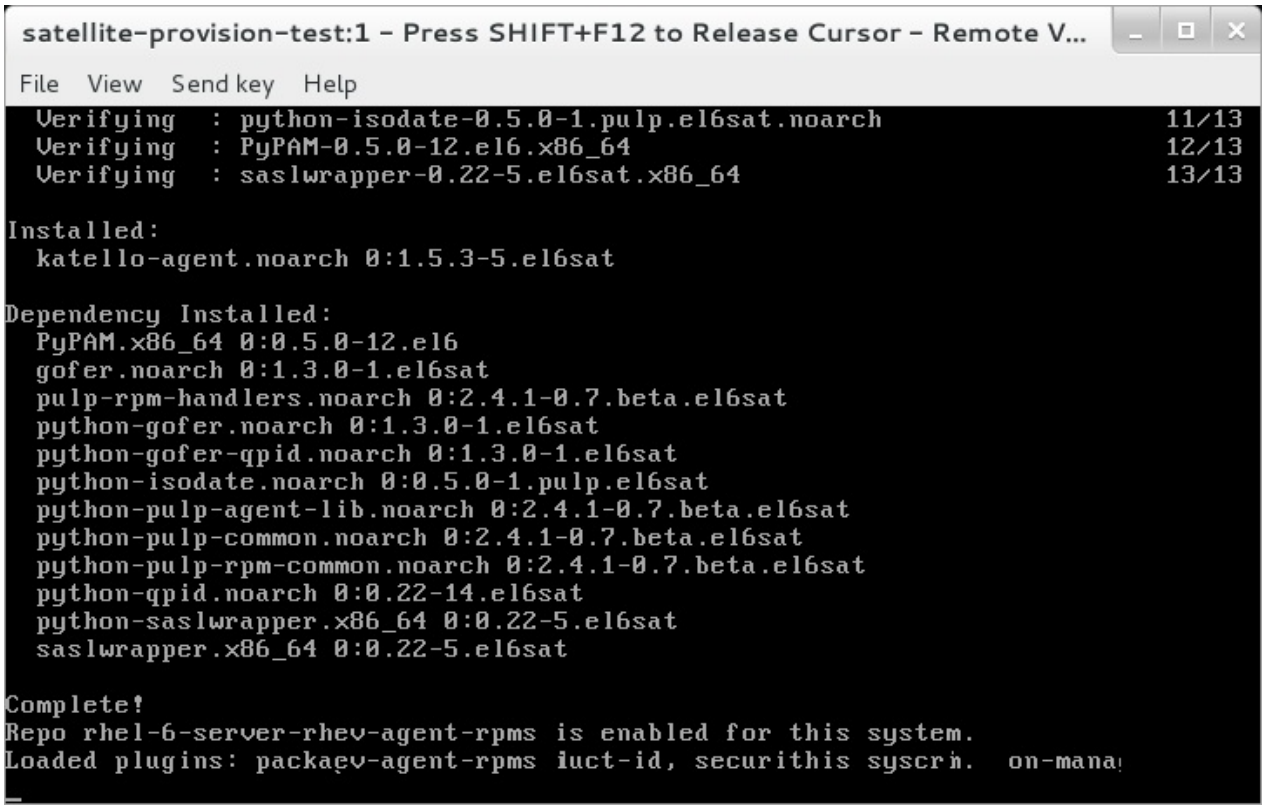


Next the post section will run, switching you to VT3 so that you can follow.

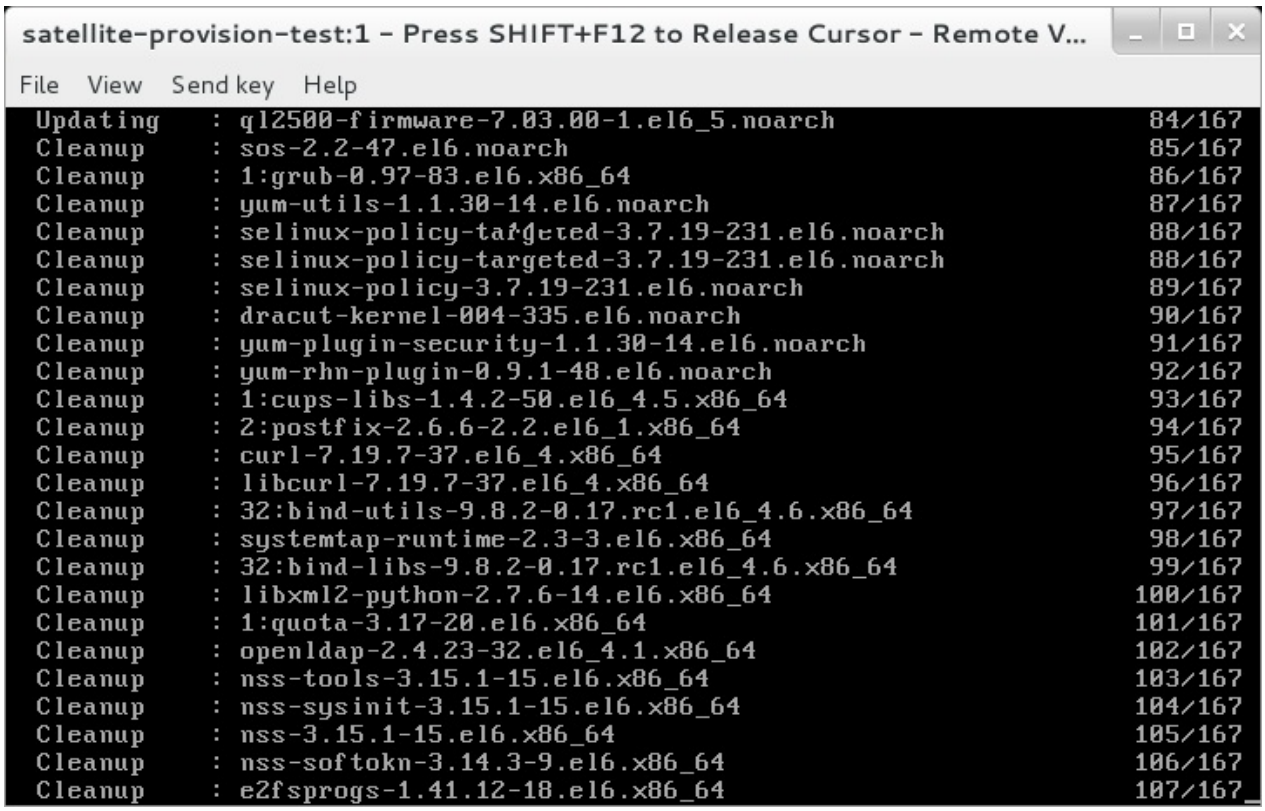
First it will register, via **subscription-manager**, to the Satellite



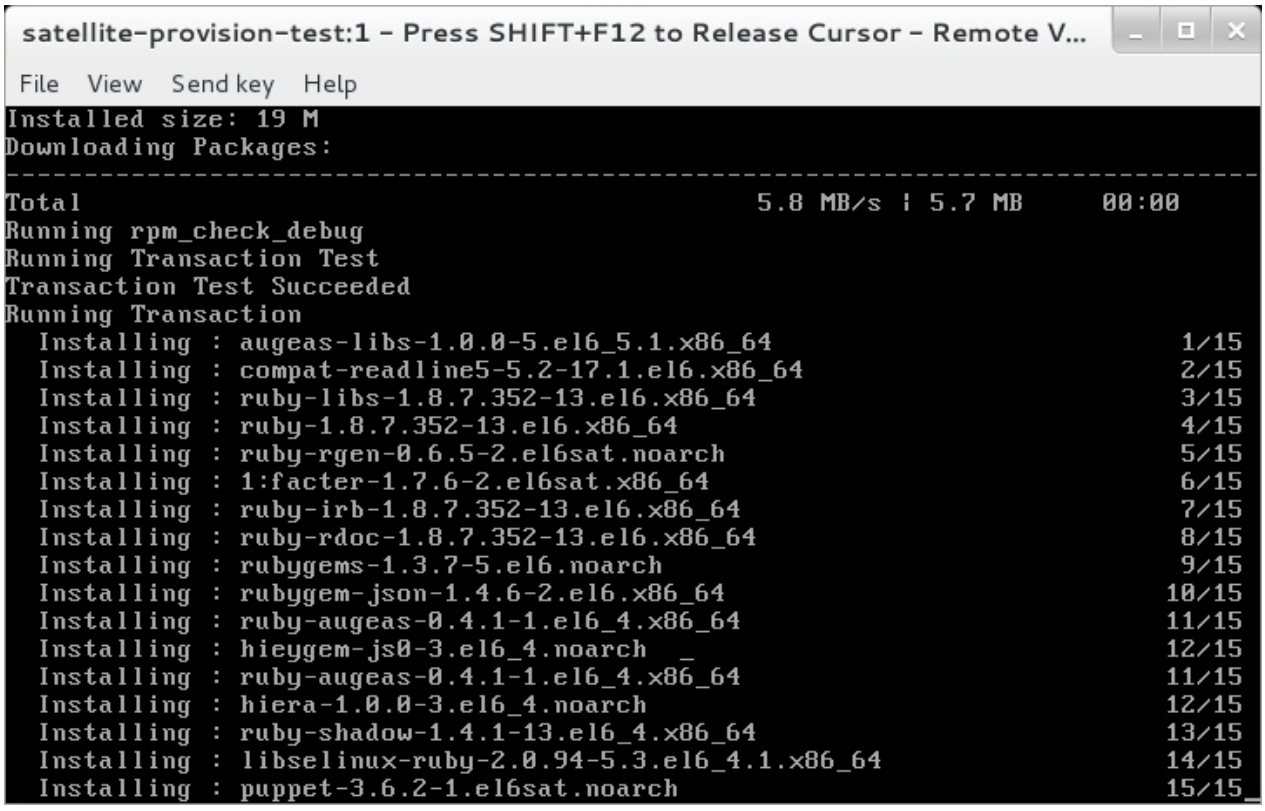
Next it will install the katello-agent



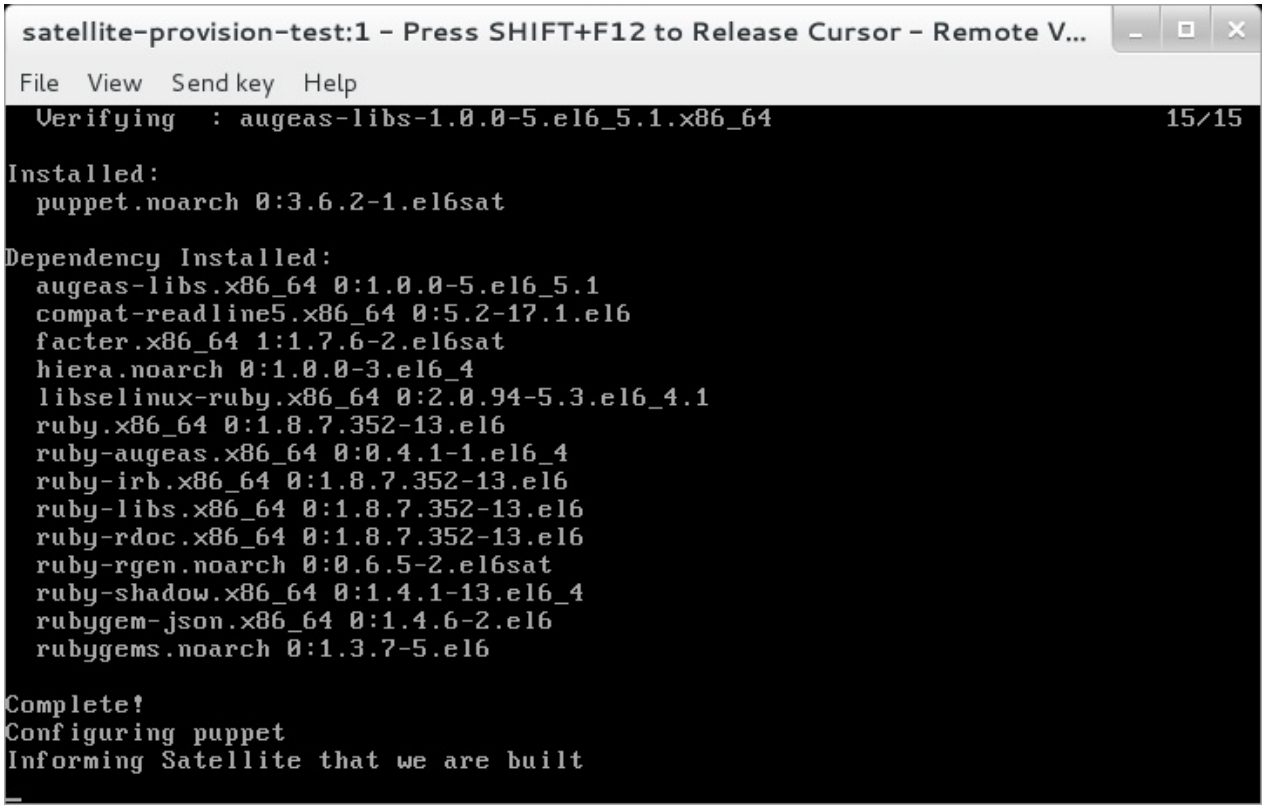
This will be followed by a `yum update`



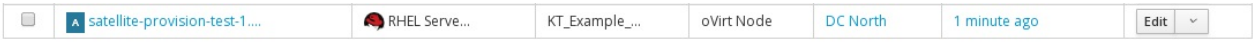
After the full update, the final install will happen, it will install **puppet**



Finally, once puppet installs, it will configure puppet and inform the Satellite server that it is built

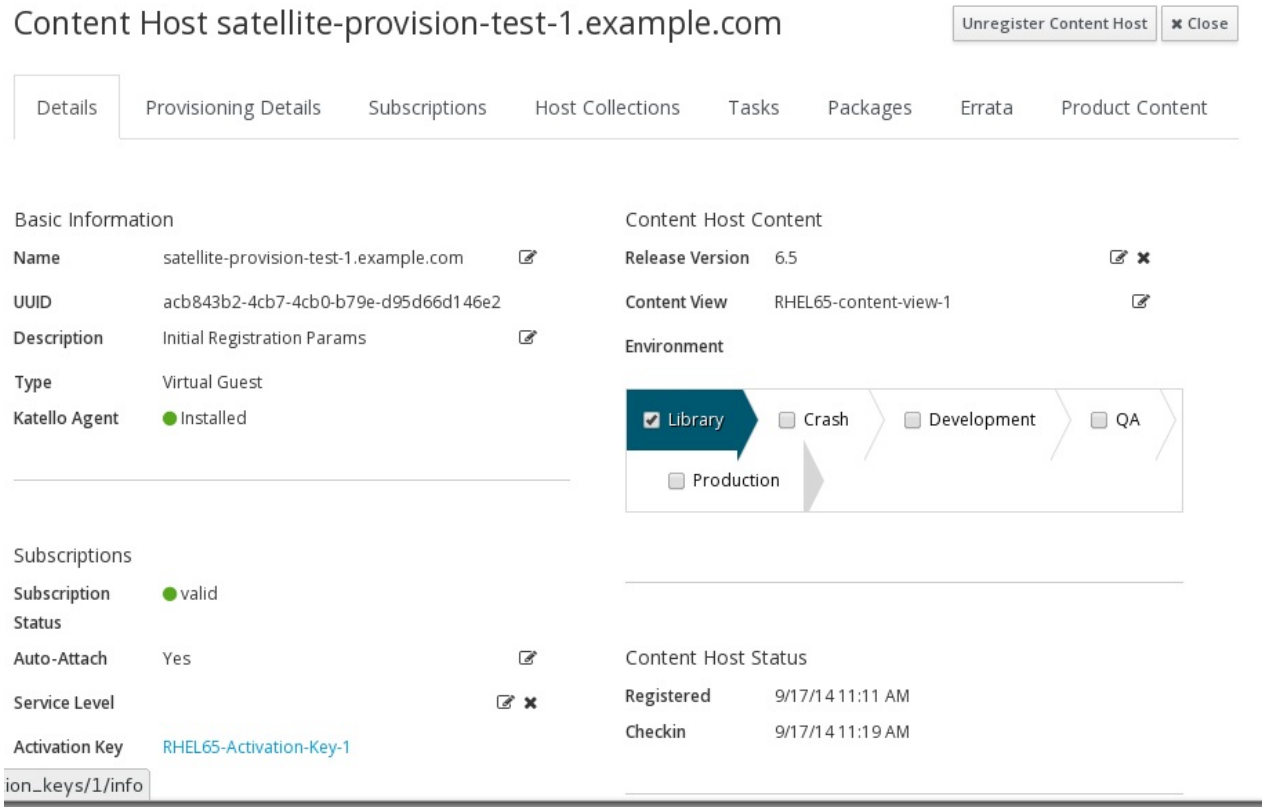


Back on the Satellite Server, under `Hosts > All Hosts` , you will see the new host initally has a blue A (Active) next to it. This simply means that puppet has made changes during its initial run. It will change to a green O (no changes) next time puppet runs -in about 30 mins time.



Also on the Satellite Server, check the status of the **Content Hosts** `Hosts > Content Hosts`

Click on the **Content Host** to see more details (*the screenshots abave and below need updating*)


Content Host satellite-provision-test-1.example.com
Unregister Content Host
Close
Details
Provisioning Details
Subscriptions
Host Collections
Tasks
Packages
Errata
Product Content
Basic Information
Name
satellite-provision-test-1.example.com
UUID
acb843b2-4cb7-4cb0-b79e-d95d66d146e2
Description
Initial Registration Params
Type
Virtual Guest
Katello Agent
Installed
Subscriptions
Subscription Status
valid
Auto-Attach
Yes
Service Level
Activation Key
RHEL65-Activation-Key-1
ion_keys/1/info
Content Host Content
Release Version
6.5
Content View
RHEL65-content-view-1
Environment
Library
Crash
Development
QA
Production
Content Host Status
Registered
9/17/14 11:11 AM
Checkin
9/17/14 11:19 AM

Troubleshooting

Its quite possible, with all the configuration that we have just done, that we missed something or that a bug (this is a *.0 release afterall) has meant that we are unable to provision.

In this section I will give you a few places to check that have been common issues with my own Satellite configuration.

Locations and Organisations

These are relatively new in the development lifecycle, and while the GA is better than the beta, there is still room for improvement.

A nice way to check that we have all the elements we need in each **Location** and **Organisation** is to use the hammer command to list all info about each one

```
hammer  location list
---|-----------------
ID | NAME
---|-----------------
2  | Default_Location
9  | Europe
---|-----------------
```

```
hammer  location info --id 9
Id:                 9
Name:               Europe
Users:

Smart proxies:

Subnets:
    172.16.30.0/24 (172.16.30.0/24)
Compute resources:

Installation media:

Templates:
    freeipa_register ()
    idm_register ()
    Kickstart default iPXE (iPXE)
    Kickstart default PXELinux (PXELinux)
    puppet.conf ()
    PXELinux global default (PXELinux)
    Satellite Finish Default (finish)
    Satellite Kickstart Default (provision)
    Satellite User Data Default (user_data)
    subscription_manager_registration ()
Domains:
    Example.com
Environments:

Hostgroups:
    DC North
Parameters:

Created at:         2014/09/18 17:58:16
Updated at:         2014/09/20 16:03:21
```

The things that jump out to me above are

- missing Smart Proxies
- Missing Installation Media

- Missing (puppet) Envronments
- Missing Parameters

Next lets take a look for **organisations**

```
hammer organization list
---|----------------------|----------------------|--------------
--------------------
ID | NAME                 | LABEL                | DESCRIPTION
---|----------------------|----------------------|--------------
--------------------
1  | Default_Organization | Default_Organization | Default_Organ
ization Organization
10 | Example Org          | Example_Org          |
---|----------------------|----------------------|--------------
--------------------
```

```
hammer organization info --id 10
Id:                     10
Name:                   Example Org
Users:

Smart proxies:
    sat6-hammer-test.example.com
Subnets:
    172.16.30.0/24 (172.16.30.0/24)
Compute resources:

Installation media:
    Example_Org/Library/Red_Hat_6_Server_Kickstart_x86_64_6_5
Templates:
    freeipa_register ()
    idm_register ()
    Kickstart default iPXE (iPXE)
    Kickstart default PXELinux (PXELinux)
    puppet.conf ()
    PXELinux global default (PXELinux)
    Satellite Finish Default (finish)
    Satellite Kickstart Default (provision)
    Satellite User Data Default (user_data)
    subscription_manager_registration ()
Domains:
    example.com
Environments:
    KT_Example_Org_Library_RHEL65_Content_View_1_5
Hostgroups:
    DC North
Parameters:

Created at:             2014/09/18 17:58:19
Updated at:             2014/09/20 16:03:12
Label:                  Example_Org
Description:
Red Hat Repository URL: https://cdn.redhat.com
```

Tasks

From time to time a task with either get stuck or fail. The steps listed on this upstream webpage can be very useful to help on these occasions

http://www.katello.org/troubleshooting/index.html

Next Steps

In the next book in the series, Next Steps with Satellite 6 Command Line, I will cover

- Adding Other Repositories
  - Third Party RPM Repositories
  - Puppet Repositories
- Adding Existing Hosts to Satellite
- Backing Up the Satellite
- Working with Capsules

The sister book this one, Getting Started with Satellite 6 Command line is aimed to achieving the same results as this book but by using only the command line interface, called **Hammer**

Other Books

Here are details of other books I have planned.

- Getting Started with Satellite 6 (sister to this book)
- Getting Started with Satellite 6 Command line (THIS BOOK)
- Next Steps with Satellite 6 (Working on the outline)

- *Next Steps with Satellite 6 Command line (sister to this book, not started)*
- *Doing More with Satellite 6 (not started)*
- *Doing More with Satellite 6 Command line (not started)*