Chef Workstation Configuration

You can consider Chef Workstation as a place where all the development work of chef happens. This is the place where most of the administrators will start working on creating cookbooks and recipes. The workstation contains a local chef repository. This repository can then be synchronized with the central chef server.

Chef workstation also will have a command line utility called "Knife", which will be used to interact with the central chef server.

Chef workstation consists of two primary components as mentioned below:

1. **Knife Utility**
2. **Chef Repository**

1. Knife Utility installation

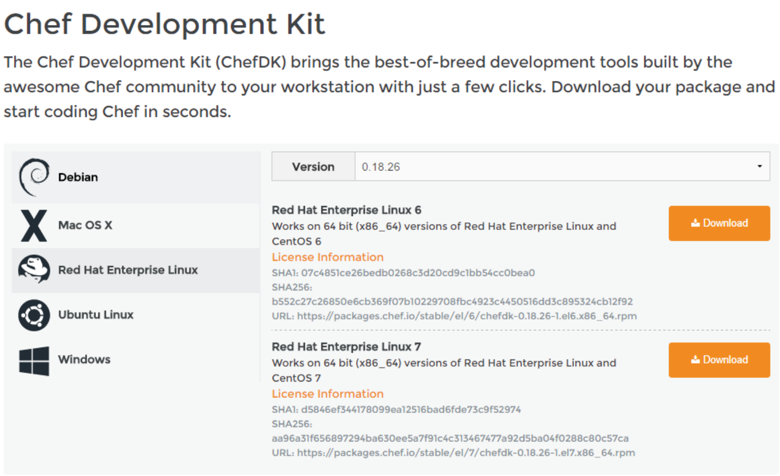
Let's first install Knife utility on our chef workstation. Knife utility and chef client can be installed easily on the workstation by simply firing up the below curl command. The below curl command downloads a bash shell script provided by chef, and then executes it.

```
curl -L https://omnitruck.chef.io/install.sh | sudo bash
```

Alternatively you can also do the following steps to install chef workstation specific components.

Chef officially provides a debian and rpm package called **chefdk**. chefdk stands for chef development kit. You can download your operating system specific package by navigating to the below URL:

https://downloads.chef.io/chef-dk/



Chef Development Kit

The Chef Development Kit (ChefDK) brings the best-of-breed development tools built by the awesome Chef community to your workstation with just a few clicks. Download your package and start coding Chef in seconds.

Version 0.18.26

Red Hat Enterprise Linux 6
Works on 64 bit (x86_64) versions of Red Hat Enterprise Linux and CentOS 6
License Information
SHA1: 07c4851ce26bedb0268c3d20cd9c1bb54cc0bea0
SHA256:
b552c27c26850e6cb369f07b10229708fbc4923c4450516dd3c895324cb12f92
URL: https://packages.chef.io/stable/el/6/chefdk-0.18.26-1.el6.x86_64.rpm

Red Hat Enterprise Linux 7
Works on 64 bit (x86_64) versions of Red Hat Enterprise Linux and CentOS 7
License Information
SHA1: d5846ef344178099ea12516bad6fde73c9f52974
SHA256:
aa96a31f656897294ba630ee5a7f91c4c313467477a92d5ba04f0288c80c57ca
URL: https://packages.chef.io/stable/el/7/chefdk-0.18.26-1.el7.x86_64.rpm

For Ubuntu, you will be doing something like the below:

```
wget https://packages.chef.io/stable/ubuntu/12.04/chefdk_0.15.15-1_amd64.deb
dpkg -i chefdk*
```

In the case of RedHat, you should be doing the below:

```
wget https://packages.chef.io/stable/el/6/chefdk-0.15.15-1.el6.x86_64.rpm
rpm -ivh chefdk*
```

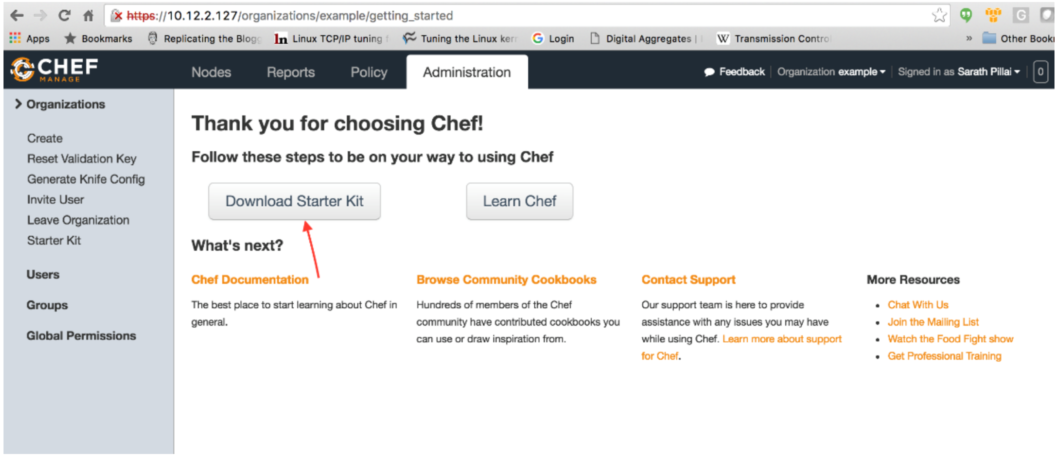
2. Starter Kit download from Chef Server

Now is the time to get the starter kit for your organization. You need to login to your chef server via a browser and download the chef starter kit.

What is Chef Starter kit?

Chef starter kit is nothing but an archive that contains the directory structure for doing development work on chef workstation. It contains the cookbook and knife settings.

You can simply download it and copy the archive to your chef workstation server. The below image shows the download page for chef starter kit from chef server manage user interface (which we have already installed during our previous tutorial)



Thank you for choosing Chef!

Follow these steps to be on your way to using Chef

What's next?

Chef Documentation

The best place to start learning about Chef in general.

Browse Community Cookbooks

Hundreds of members of the Chef community have contributed cookbooks you can use or draw inspiration from.

Contact Support

Our support team is here to provide assistance with any issues you may have while using Chef. Learn more about support for Chef.

More Resources

***Note:** When you click on the download button above, you will be shown a message with regard to resetting of the user keys. Proceed with it. This new key will be part of the starter kit that we are downloading now.*

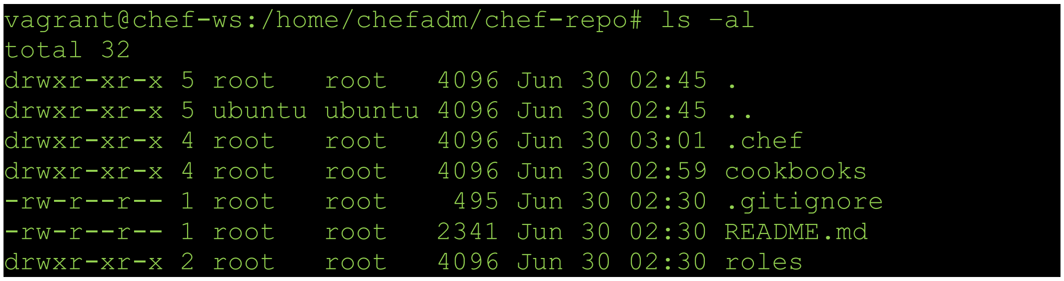
Once downloaded (chef-starter.zip), copy it down to the chef workstation server. Once copied, you can then decompress it as shown below.

```
unzip chef-starter.zip
```

The above command will create a directory called chef-repo in the current location. This directory contains all the basic settings for interacting with chef server from workstation.

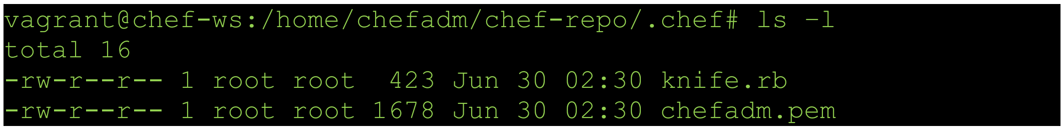
It also has a cookbook directory inside, where all cookbooks will be stored and later pushed to central chef server.

Let's get inside the "chef-repo" directory, and see what's inside.



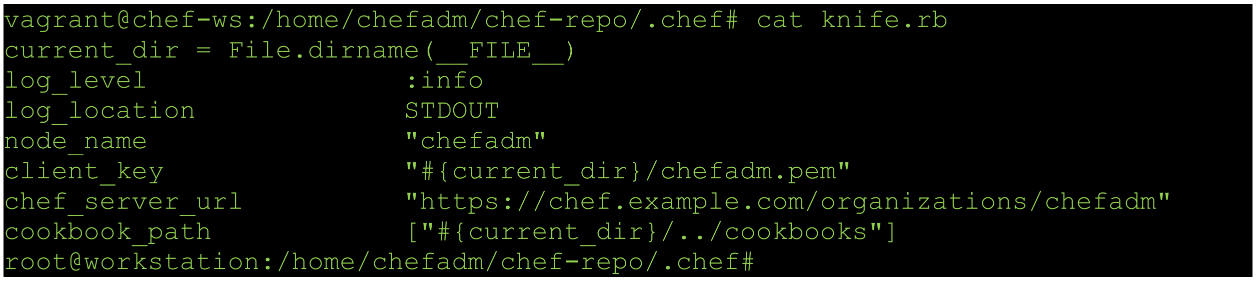
```
vagrant@chef-ws:/home/chefadm/chef-repo# ls -al
total 32
drwxr-xr-x 5 root   root   4096 Jun 30 02:45 .
drwxr-xr-x 5 ubuntu ubuntu 4096 Jun 30 02:45 ..
drwxr-xr-x 4 root   root   4096 Jun 30 03:01 .chef
drwxr-xr-x 4 root   root   4096 Jun 30 02:59 cookbooks
-rw-r--r-- 1 root   root    495 Jun 30 02:30 .gitignore
-rw-r--r-- 1 root   root   2341 Jun 30 02:30 README.md
drwxr-xr-x 2 root   root   4096 Jun 30 02:30 roles
```

There is a directory called ".chef" as evident from the directory listing above. This .chef directory contains the private key for the user. Also this ".chef" directory contains knife configuration file called knife.rb.



```
vagrant@chef-ws:/home/chefadm/chef-repo/.chef# ls -l
total 16
-rw-r--r-- 1 root root  423 Jun 30 02:30 knife.rb
-rw-r--r-- 1 root root 1678 Jun 30 02:30 chefadm.pem
```

Knife.rb file will be read by knife tool while interacting with central chef server from workstation. The contents of knife.rb file looks like the below:



```
vagrant@chef-ws:/home/chefadm/chef-repo/.chef# cat knife.rb
current_dir = File.dirname(__FILE__)
log_level                :info
log_location             STDOUT
node_name                "chefadm"
client_key               "#{current_dir}/chefadm.pem"
chef_server_url          "https://chef.example.com/organizations/chefadm"
cookbook_path            ["#{current_dir}/../cookbooks"]
root@workstation:/home/chefadm/chef-repo/.chef#
```

***Note**: https://docs.getchef.com/config_rb_knife.html for more information on knife configuration options*

Knife.rb file options explained

client_key: Specifies the path of the private key file associated with the user. This private key will be used to authenticate workstation against the chef server.

chef_server_url: This is the full URL of the chef server with the organization path (we did create an organization during our previous tutorial of installing chef server. Our organization's name is "cheflab")

cookbook_path: The absolute path where cookbooks will be stored on the workstation. This is because knife will be syncing cookbooks to server, it will also be downloading cookbooks, will be creating new cookbooks etc.

client_name: This is the user name associated with the organization. We created a user named "chefadm".

Instead of downloading the chef starter kit (which includes everything required for the workstation, along with the user key) for setting up the workstation, we can alternatively create chef repo ourselves as shown below:

1. *Install the chefdk package (as described earlier) on the workstation.*
2. *Run the command "**chef generate repo chef-repo**"*
3. *Then create .chef directory inside the repository created*

```
cd chef-repo
mkdir .chef
cd .chef
```

4. *Then create knife.rb and also copy private key of the user to the .chef directory (knife.rb should have the exact same content that we saw earlier. The private key can be found on the chef server where we created the user using the chef-server-ctl command above.)*

***Note**: The chef_server_url setting inside knife.rb uses dns name of central chef server. What this means is...You need to have proper dns resolution working on chef workstation as well. In other words, chef workstation should be able to reach chef server using the dns name mentioned in chef_server_url*

If its non-production environment, then simply adding an entry to /etc/hosts file would do. Else, it's recommended to make a proper DNS entry for chef server name.

Once all the above mentioned items are configured and ready. Let's verify the connectivity between chef workstation and chef server. As mentioned earlier, knife is going to be our tool while interacting with central chef.

Please remember the fact that you should execute knife commands only after navigating to the chef-repo directory. This is because knife looks for a directory called .chef, which contains our knife.rb settings file.

```
vagrant@chef-ws:/home/chefadm/chef-repo# knife ssl fetch
WARNING: Certificates from chef.example.com will be fetched and placed
in your trusted_cert dir (/home/chefadm/chef-repo/.chef/trusted_certs)

Knife has no means to verify these are the correct certificates. You
should verify the authenticity of these certificates after downloading.

Adding certificate for chef.example.com in /home/chefadm/chef-repo/.chef/
trusted_certs/chef_example_com.crt
```

The above command interacts with the chef server (by using the URL defined inside knife.rb file), and grabs the SSL certificates of the server. This SSL certificate is then stored inside a new directory called **trusted_certs**.

Once the SSL certificates are added to the trusted list of knife. You can then verify the connectivity to central chef server using the below command.

```
root@workstation:/home/chefadm/chef-repo# knife ssl check
Connecting to host chef.example.com:443
Successfully verified certificates from `chef.example.com'
```

If things are correctly configured, you should be able to see the above output. Now knife can do all the operations on the chef server using the user we have created (which is defined inside knife.rb), and the organization associated with that user.

Chef workstation configuration is now complete and it should be talking the chef server. In order to verify this, let's run the following command on your chef workstation:

```
root@workstation:/home/chefadm/chef-repo# knife node list
node1
node2
node3
```

That completes the configuration and verification for Chef Workstation and now you can do things like creating and working with cookbooks and recipes and bootstrapping new nodes to your existing infrastructure.

Using Chef Workstation to manage nodes/cookbooks

1. How to Bootstrap a node using chef workstation?

Bootstrapping a node is nothing but the job of installing and configuring chef agent on a server that needs to be automated via chef. Installing and configuring a node with chef agent which will then start pulling configuration from central chef server, is a single knife command away as shown below.

Syntax:

```
knife bootstrap fqdn --ssh-user username --ssh-password 'password' --node-name node1
```

Example:

```
knife bootstrap 172.16.12.12 --ssh-user root --ssh-password 'p@ssword' --node-name node1
```

The above command connects to the server 172.16.12.12 (an example node which is automated in this case), and then downloads chef agent and configures it to connect to our central chef server.

You can alternatively use the below command to bootstrap, in case your node only supports key based authentication.

```
knife bootstrap 172.16.12.12 --ssh-user username --sudo --identity-file
privatekey.pem --node-name node1.example.com
```

Replace privatekey.pem with the real private key file. And we are also using sudo so that commands executed while installing chef agent on that node will be prepend with sudo.

The node name that we gave in both the commands will add that name for the node on the chef server. Chef server will be able to identify this node with that name.

2. How to create and Upload a cookbook to chef server from workstation?

The second operation that the chef workstation does is to create a cookbook which acts as the main basic block of configuration that will be applied on nodes.

You can create a skeleton cookbook by using the below command.

```
knife cookbook create test
```

***Note**: Please do not forget the fact that knife commands needs to from chef-repo directory (as it requires ".chef" directory, which has the "knife.rb" settings file)*

You can then upload this cookbook that we just created to central chef server using the below command.

```
knife cookbook upload test
```