

Chef installation and configuration

What is chef?

Chef is a powerful automation platform that transforms infrastructure into code. Whether you're operating in the cloud, on-premises, or in a hybrid environment, Chef automates how infrastructure is configured, deployed, and managed across your network, no matter its size

Chef Server:

The server acts as a hub for configuration data. The server stores cookbooks, the policies that are applied to nodes, and metadata that describes each registered node that is being managed by the chef-client. Nodes use the chef-client to ask the server for configuration details, such as recipes, templates, and file distributions. Starting with the release of Chef 11.x, the front-end for the server is written using Erlang.

Workstations:

A workstation is a computer that is configured to run Knife, to synchronize with the chef-repo, and interact with a single server. The workstation is the location from which most users will do most of their work, including:

- Developing cookbooks and recipes (and authoring them using Ruby).
- Keeping the chef-repo synchronized with version source control.
- Using Knife to upload items from the chef-repo to the server.
- Configuring organizational policy, including defining roles and environments and ensuring that critical data is stored in data bags.
- Interacting with nodes, as (or when) required, such as performing a bootstrap operation

Node:

A node is any server or virtual server that is configured to be maintained by a chef-client. A node can be any physical, virtual, or cloud machine that can run the chef-client. A chef-client is an agent that runs locally on every node that is registered with the server. When a chef-client is run, it will perform all of the steps that are required to bring the node into the expected state, including:

- Registering and authenticating the node with the server.
- Building the node object.
- Synchronizing cookbooks.
- Compiling the resource collection by loading each of the required cookbooks, including recipes, attributes, and all other dependencies.
- Taking the appropriate and required actions to configure the node.
- Looking for exceptions and notifications, handling each as required.

RSA public key-pairs are used to authenticate the chef-client with the server every time a chef-client needs access to data that is stored on the server. This prevents any node from accessing data that it shouldn't and it ensures that only nodes that are properly registered with the server can be managed.

Prerequisite

- 1) Host should have fully configured hostname.
- 2) Should have DNS entry in place.
- 3) Following package are required.

```
# yum install wget curl -y
```

Chef Server Installation

- 1) Go to <https://downloads.chef.io/chef-server/>
- 2) Select the **operating system**, **version**, and **architecture**.
- 3) Select the version of **Chef Server 11.x** to download, and then click the link that appears to download the package.
- 4) Install the downloaded package using the correct method for the operating system on which Chef Server 11.x will be installed

```
# wget https://packages.chef.io/stable/el/6/chef-server-11.1.0-1.el6.x86_64.rpm  
#rpm -ivh chef-server-11.1.0-1.el6.x86_64.rpm
```

Configure Chef Server 11.x by running the following command:

```
# chef-server-ctl reconfigure
```

The `*chef-server-ctl*` command will set up all of the required components, including Erchef, RabbitMQ, PostgreSQL, and all of the cookbooks that are used by chef to maintain Chef Server 11.x.

Verify the the hostname for the server by running the `*hostname*` command. The hostname for the server must be a FQDN

```
# hostname
```

Verify the installation of Chef Server 11.x by running the following command:

```
# chef-server-ctl test
```

Note: Try to stop apache before running this test.

You can explore the Chef Server URL using your favorite browser:

```
# https://FQDN-OR-IP-OF-CHEF-SERVER
```

Note: Default **UserName/Password** is **admin/p@ssw0rd1**

The `*chef-server-ctl*` command is used on the Chef Server system for management. It has built-in help (-h) that will display the various sub-commands.

Chef WorkStation Installation

```
# curl -L https://www.opscode.com/chef/install.sh | bash
% Total % Received % Xferd Average Speed Time Time Time Current
      Dload Upload Total Spent Left Speed
101 6790 101 6790 0 0 3826 0 0:00:01 0:00:01 --:--:-- 12190
Downloading Chef for el...
Installing Chef
warning: /tmp/tmp.KnyQTnqz/chef-.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a:
NOKEY
Preparing... ##### [100%]
1:chef ##### [100%]
Thank you for installing Chef!
```

2) When the installation is finished enter the ***chef-client*** command to verify that the chef-client was installed:

```
# chef-client -v
Chef: 11.6.0
```

Create the **“.chef”** directory

The **.chef** directory is used to store three files:

- knife.rb
- ORGANIZATION-validator.pem
- USER.pem

a) Copy **Cert Keys** from **Chef Server** to your **Workstation** User Folder:

```
$ mkdir ~/.chef
$ scp root@chef-server:/etc/chef-server/admin.pem ~/.chef
$ scp root@chef-server:/etc/chef-server/chef-validator.pem ~/.chef
```

Create the **“.chef”** directory

The **.chef** directory is used to store three files:

- knife.rb
- ORGANIZATION-validator.pem
- USER.pem

a) Copy **Cert Keys** from **Chef Server** to your **Workstation** User Folder:

```
$ mkdir ~/.chef
$ scp root@chef-server:/etc/chef-server/admin.pem ~/.chef
$ scp root@chef-server:/etc/chef-server/chef-validator.pem ~/.chef
```

Your Knife config file (**knife.rb**) will look like:

```
$ cat ~/.chef/knife.rb
log_level :info
log_location STDOUT
```

```
node_name      'knife-user1'
client_key     '/root/.chef/knife-user1.pem'
validation_client_name 'chef-validator'
validation_key '/root/.chef/admin.pem'
chef_server_url 'https://chef-server.example.com:443/'
syntax_check_cache_path '/root/.chef/syntax_check_cache'
```

Your Knife config file (**knife.rb**) will look like:

```
$ cat ~/.chef/knife.rb
log_level      :info
log_location   STDOUT
node_name      'knife-user1'
client_key     '/root/.chef/knife-user1.pem'
validation_client_name 'chef-validator'
validation_key '/root/.chef/admin.pem'
chef_server_url 'https://chef-server.example.com:443/'
syntax_check_cache_path '/root/.chef/syntax_check_cache'
```

III) Chef Node Installation

1) Run the following command that appears (for UNIX and Linux environments):

```
# curl -L https://www.opscode.com/chef/install.sh | bash
% Total    % Received % Xferd Average Speed   Time    Time     Current
   Dload  Upload  Total  Spent    Left  Speed
101 6790 101 6790    0   3826    0 0:00:01 0:00:01 --:--:-- 12190
Downloading Chef for el...
Installing Chef
warning: /tmp/tmp.KnyQTnqz/chef-.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a:
NOKEY
Preparing...      ##### [100%]
1:chef            ##### [100%]
Thank you for installing Chef!
```

2) Create the Chef Directory.

```
# mkdir /etc/chef
```

3) Copy Chef Server Validation Cert Keys from Chef Server to your Node at **“/etc/chef”**:

III) Chef Node Installation

1) Run the following command that appears (for UNIX and Linux environments):

```
# curl -L https://www.opscode.com/chef/install.sh | bash
% Total    % Received % Xferd Average Speed   Time    Time     Current
   Dload  Upload  Total  Spent    Left  Speed
101 6790 101 6790    0   3826    0 0:00:01 0:00:01 --:--:-- 12190
Downloading Chef for el...
Installing Chef
warning: /tmp/tmp.KnyQTnqz/chef-.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a:
NOKEY
Preparing...      ##### [100%]
1:chef            ##### [100%]
Thank you for installing Chef!
```

2) Create the Chef Directory.

```
# mkdir /etc/chef
```

3) Copy Chef Server Validation Cert Keys from Chef Server to your Node at “**/etc/chef**”:

3) Copy Chef Server Validation Cert Keys from Chef Server to your Node at “**/etc/chef**”:

```
# scp root@chef-server:/etc/chef-server/chef-validator.pem /etc/chef
```

4) Log in to Chef client and run the following command in order for a client to register itself with Chef Server:

```
# chef-client -S https://FQDN-OR-IP-OF-CHEF-SERVER -K /etc/chef/chef-validator.pem
```

5) Once the client is verified, we need to create a “**client.rb**” file inside “**/etc/chef**”.

```
# vi /etc/chef/client.rb
log_level      :info
log_location   STDOUT
chef_server_url 'https://FQDN-OR-IP-OF-CHEF-SERVER'
```

6) Verify the Node is successfully registered with Chef Server using:

a) From Workstation Machine:

```
# knife node list
```

b) From Chef Server Web UI (Node List):

```
# https://FQDN-OR-IP-OF-CHEF-SERVER
```

7) Run the Chef Client to check if the respective cookbook (recipe's) are pushed to that node:

```
# chef-client
# chef-client -l debug (In case if you want to debug)
```

8) Starts the chef-client which will poll the chef-server every 3600 seconds for changes.

```
# chef-client -i 3600
```