Chef Server, Workstation and Chef Client on Ubuntu 18.04

Chef is a configuration management technology used to automate the infrastructure provisioning. It is developed based on Ruby DSL language. It is used to streamline the task of configuration and managing the company's server. It has the capability to get integrated with any of the cloud technology.

In DevOps, we use Chef to deploy and manage servers and applications in-house and on the cloud.

Chef is an open source technology developed by Opscode.

Pre-requisites:

Supported Platforms

The following table lists the commercially-supported platforms and versions for the Chef Infra Server:

Platform	Architecture	Version
CentOS	`x86_64`	`7.x`, `8.x`
Oracle Enterprise Linux	`x86_64`	`7.x`, `8.x`
Red Hat Enterprise Linux	`x86_64`	`7.x`, `8.x`
SUSE Enterprise Linux Server	`x86_64`	`12.x`, `15.x`
Ubuntu	`x86_64`	`16.04`, `18.04`, `20.04`

Reference : https://docs.chef.io/server/install_server/

Install and Configure the Chef Server

The Chef server is the hub of interaction between all workstations and nodes under Chef management. Changes made to configuration code on workstations are pushed to the Chef server, and then pulled by a node's chef-client to apply the configurations.

#1: Download the latest Chef server package

```
wget https://packages.chef.io/files/stable/chef-server/14.3.14/ubuntu/18.04/chef-server-core_14.3.14-1_amd64.deb
```

```
root@chefserver:/home/ubuntu# wget https://packages.chef.io/files/stable/chef-server/14.3.14/ubuntu/18.04/chef-server-core_14.3.14-1_amd64.deb
--2021-05-06 06:20:39--  https://packages.chef.io/files/stable/chef-server/14.3.14/ubuntu/18.04/chef-server-core_14.3.14-1_amd64.deb
Resolving packages.chef.io (packages.chef.io)... 199.232.22.110
Connecting to packages.chef.io (packages.chef.io)|199.232.22.110|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 225665744 (215M) [application/x-debian-package]
Saving to: 'chef-server-core_14.3.14-1_amd64.deb'

chef-server-core_14.3.14-1_amd64.deb    100%[=============================================================================>] 215.21M  67.6MB/s

2021-05-06 06:20:56 (65.3 MB/s) - 'chef-server-core_14.3.14-1_amd64.deb' saved [225665744/225665744]

root@chefserver:/home/ubuntu#
```

#2: Installation of the chef server package

```
# dpkg -i chef-server-core_14.3.14-1_amd64.deb
```

```
root@chefserver:/home/ubuntu# ls
chef-server-core_14.3.14-1_amd64.deb
root@chefserver:/home/ubuntu#
root@chefserver:/home/ubuntu# dpkg -i chef-server-core_14.3.14-1_amd64.deb
Selecting previously unselected package chef-server-core.
(Reading database ... 57240 files and directories currently installed.)
Preparing to unpack chef-server-core_14.3.14-1_amd64.deb ...
Unpacking chef-server-core (14.3.14-1) ...
Setting up chef-server-core (14.3.14-1) ...
Thank you for installing Chef Infra Server!

Run 'chef-server-ctl reconfigure' to configure your Chef Infra Server

For more information on getting started see https://docs.chef.io/server/
root@chefserver:/home/ubuntu#
```

The Chef server includes a command line utility called chef-server-ctl. Run chef-server-ctl to start the Chef server services.

#3: Re-configure the chef server

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

```
root@chefserver:/home/ubuntu# chef-server-ctl reconfigure

Documentation: https://docs.chef.io/server_overview/
Patents:       https://www.chef.io/patents

+----------------------------------------------+
            Chef License Acceptance

Before you can continue, 3 product licenses
must be accepted. View the license at
https://www.chef.io/end-user-license-agreement/

Licenses that need accepting:
  * Chef Infra Server
  * Chef Infra Client
  * Chef InSpec

Do you accept the 3 product licenses (yes/no)?

> yes

Persisting 3 product licenses...
  3 product licenses persisted.

+----------------------------------------------+
Starting Chef Infra Client, version 15.17.4
resolving cookbooks for run list: ["private-chef::default"]
```

To Verify chef server running status run the below commad

```
# chef-server-ctl status
```

```
root@chefserver:/home/ubuntu# chef-server-ctl status
run: bookshelf: (pid 19367) 64s; run: log: (pid 18999) 90s
run: elasticsearch: (pid 19347) 64s; run: log: (pid 18806) 127s
run: nginx: (pid 19333) 65s; run: log: (pid 19162) 73s
run: oc_bifrost: (pid 19220) 66s; run: log: (pid 18704) 139s
run: oc_id: (pid 19269) 65s; run: log: (pid 18719) 133s
run: opscode-erchef: (pid 19385) 64s; run: log: (pid 19130) 84s
run: postgresql: (pid 19193) 66s; run: log: (pid 18184) 151s
run: redis_lb: (pid 19183) 67s; run: log: (pid 19182) 67s
root@chefserver:/home/ubuntu#
```

Create Chef User and Organization

In order to link the workstations and nodes to the Chef server, need to create an administrator and organization with associated RSA private keys. From the home directory, create a **.chef** directory to store the keys.

#1: Create the .chef folder

```
#mkdir .chef
```

```
root@chefserver:/home/ubuntu# mkdir .chef
```

Chef-server-ctl to create a user change the following values accordingly:
USER_NAME,
FIRST_NAME,
LAST_NAME, EMAIL,
PASSWORD. Adjust USER_NAME.pem –filename /home/ubuntu/.chef/chefadmin.pem

#2: User creation in the chef server

```
chef-server-ctl user-create USER_NAME FIRST_NAME LAST_NAME EMAIL 'PASSWORD' --
filename /home/ubuntu/.chef/
```

```
root@chefserver:/home/ubuntu# chef-server-ctl user-create chefadmin Chef Administrator lingamnatraj93@gmail.com 'Lingam@123' --filename /home/ubuntu/.chef/che
fadmin.pem
```

Create an organization and add the user created in the previous step. Replace ORG_NAME with a short identifier for the organization, ORG_FULL_NAME with the organizations' complete name, USER_NAME with the username created in the step above and ORG_NAME.pem with organization's short identifier followed by .pem

#3: Creation of organization in the chef server

```
chef-server-ctl org-create ORG_NAME "ORG_FULL_NAME" --association_user
USER_NAME --filename ~/.chef/ORG_NAME.pem
```

```
root@chefserver:/home/ubuntu#
root@chefserver:/home/ubuntu# chef-server-ctl org-create test-org "Chef Infrastructure on Ubuntu 18.04" --association_user chefadmin --filename /home/ubuntu/.
chef/test-org.pem
root@chefserver:/home/ubuntu#
```

To view the list of all organizations on your Chef server, use the following command:

```
#chef-server-ctl org-list
```

```
root@chefserver:/home/ubuntu# chef-server-ctl org-list
test-org
root@chefserver:/home/ubuntu#
```

Install and Configure the Chef Workstation

#1: Download the latest Chef Workstation

```
# wget https://packages.chef.io/files/stable/chef-workstation/21.4.365/ubuntu/18.04/chef-workstation_21.4.365-1_amd64.deb
```

```
root@workstation:/home/ubuntu# wget https://packages.chef.io/files/stable/chef-workstation/21.4.365/ubuntu/18.04/chef-workstation_21.4.365-1_amd64.deb
--2021-05-06 07:01:43--  https://packages.chef.io/files/stable/chef-workstation/21.4.365/ubuntu/18.04/chef-workstation_21.4.365-1_amd64.deb
Resolving packages.chef.io (packages.chef.io)... 199.232.22.110
Connecting to packages.chef.io (packages.chef.io)|199.232.22.110|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 174336370 (166M) [application/x-debian-package]
Saving to: 'chef-workstation_21.4.365-1_amd64.deb'

chef-workstation_21.4.365-1_amd64.deb   100%[=============================================================================>] 166.26M  75.9MB/s    in 2.2s

2021-05-06 07:02:03 (75.9 MB/s) - 'chef-workstation_21.4.365-1_amd64.deb' saved [174336370/174336370]
```

#2: Chef-Workstation installation

```
# ls

# dpkg – i chef-workstattion_21.4.365-1_amd64.deb
```

```
root@workstation:/home/ubuntu# ls
chef-workstation_21.4.365-1_amd64.deb
root@workstation:/home/ubuntu#
root@workstation:/home/ubuntu# dpkg -i chef-workstation_21.4.365-1_amd64.deb
Selecting previously unselected package chef-workstation.
(Reading database ... 57240 files and directories currently installed.)
Preparing to unpack chef-workstation_21.4.365-1_amd64.deb ...
Unpacking chef-workstation (21.4.365-1) ...
Setting up chef-workstation (21.4.365-1) ...

Chef Workstation ships with a toolbar application, the Chef Workstation App.
To run this application some additional dependencies must be installed.
Using your platform's package manager to install the 'electron' package is
the easiest way to meet the dependency requirements.

You can then launch the App by running 'chef-workstation-app'.
The App will then be available in the system tray.

Thank you for installing Chef Workstation!
You can find some tips on getting started at https://docs.chef.io/workstation/getting_started/

root@workstation:/home/ubuntu#
```

Now we need to create the Chef repository. The chef-repo directory will store your Chef cookbooks and other related files.

```
#chef generate repo chef-repo
```

```
root@workstation:/home/ubuntu# chef generate repo chef-repo
+---------------------------------------------+
            Chef License Acceptance

Before you can continue, 3 product licenses
must be accepted. View the license at
https://www.chef.io/end-user-license-agreement/

Licenses that need accepting:
  * Chef Workstation
  * Chef Infra Client
  * Chef InSpec

Do you accept the 3 product licenses (yes/no)?

> yes

Persisting 3 product licenses...
  3 product licenses persisted.

+---------------------------------------------+
Generating Chef Infra repo chef-repo
- Ensuring correct Chef Infra repo file content

Your new Chef Infra repo is ready! Type `cd chef-repo` to enter it.
root@workstation:/home/ubuntu#
```

Create a .chef subdirectory. The .chef subdirectory will store the Knife configuration file and the .pem files that are used for RSA key pair authentication with the Chef server. Move into the chef-repo directory:

```
root@workstation:~# mkdir ~/chef-repo/.chef

root@workstation:~# cd chef-repo
```

Authentication between the Chef server and workstation and/or nodes is done with public key encryption. This ensures that the Chef server only communicates with trusted machines.

Now We will generate a RSA key-pair on the workstation server. This key-pair will be used to gain access to the Chef server

```
# ssh-keygen -b 4096
```

```
# ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:sR+Nloq6vsc7rX4ZmMInP3SKdk4fYEJH1iLoKNm1YMg
The key's randomart image is:
+---[RSA 4096]----+
|.. .  o.         |
|.E+ oo. .        |
| * o.o...        |
|+ o...   o +     |
|.  .. ooS = .    |
|    +o=oo+ .     |
|     Oo+oo.      |
|    ooO.+.       |
|   o=B=*.        |
+----[SHA256]-----+
```

Upload the workstation node's public key to the Chef server node.

- Copy the generated public keys id_rsa.pub(/root/.ssh/id_rsa.pub) to the chef server under the /root/.ssh/authorized_keys
- Copy the .pem files from your Chef server which are generated in the chef server Keys will be saved in /home/ubuntu/.chef/

#3: Cookbook Devlopment

Generate a new Chef cookbook:

Create the Cookbook

On workstation, move to your chef-repo/cookbooks directory:

```
#cd chef-repo/cookbooks
```

Create the cookbook. In this instance the cookbook is titled test_cookbook:

```
# chef generate cookbook test_cookbook
```

```
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks# chef generate cookbook test_cookbook
Generating cookbook test_cookbook
- Ensuring correct cookbook content

Your cookbook is ready. Type `cd test_cookbook` to enter it.

There are several commands you can run to get started locally developing and testing your cookbook.
Type `delivery local --help` to see a full list of local testing commands.

Why not start by writing an InSpec test? Tests for the default recipe are stored at:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb

root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks#
```

Move to your cookbook's newly created directory:

```
# cd test_cookbook
```

```
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks# cd test_cookbook/
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks/test_cookbook#
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks/test_cookbook#
```

create attributes directory and default.rb file

```
#mkdir attributes

#touch attributes/default.rb
```

```
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks/test_cookbook# mkdir attributes
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks/test_cookbook#
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks/test_cookbook# touch attributes/default.rb
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks/test_cookbook#
```

add the below content to recipes (recipes/default.rb)

```
directory '/home/ubuntu/test' do
    owner 'root'
    group 'root'
    mode '0755'
    action :create
  end

  user 'test_user' do
    comment 'Test user'
    system true
    shell '/bin/bash'
    manage_home true
    password node.default['test_cookbook']['user']['password']
  end
```

Traverse to attributes/default.rb file add the below content

```
default['test_cookbook']['user']['password'] ='86d97f9f113c9339bf13fe9de3266226'
```

Upload cookbook to chef Server

```
Knife upload cookbook test_cookbook
```

```
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks# knife cookbook upload test_cookbook
Uploading test_cookbook  [0.1.0]
Uploaded 1 cookbook.
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks#
```

Configure Knife and Bootstrapping a Client Node

#1:Knife Configuration

The following configuration need to be added in the Workstation node

Create a knife configuration file by navigating to /home/ubuntu/chef-repo/.chef directory and creating a file named config.rb using your preferred text editor. Copy the following configuration into the config.rb file:

vi /home/ubuntu/chef-repo/.chef/config.rb

```
current_dir = File.dirname(__FILE__)
log_level                :info
log_location             STDOUT
node_name                'chefadmin'
client_key               "chefadmin.pem"
validation_client_name   'test-org-validator'
validation_key           "test-org-validator.pem"
trusted_certs_dir        "/home/ubuntu/chef-repo/.chef/trusted_certs"
chef_server_url          'https://ec2-13-234-30-21.ap-south-1.compute.amazonaws.com/organizations/test-org'
cache_type               'BasicFile'
cache_options( :path => "#{ENV['HOME']}/.chef/checksums" )
cookbook_path            ["#{current_dir}/../cookbooks/"]

ssl_verify_mode          :verify_none

# EC2 subcommand
knife[:availability_zone] = "ap-south-1a"
knife[:region] = "ap-south-1"
knife[:image] = "ami-04bde106886a53080"
knife[:flavor] = "t2.micro"

knife[:aws_access_key_id] = "AKIA42QCRBLVYH3QDCE7"
knife[:aws_secret_access_key] = "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"
knife[:aws_ssh_key_id] = "chef-server"
```

```
current_dir = File.dirname(__FILE__)
log_level                :info
log_location             STDOUT
node_name                'node_name'
client_key               "USER.pem"
validation_client_name   'ORG_NAME-validator'
validation_key           "ORGANIZATION-validator.pem"
chef_server_url          'https://ec2---.com/organizations/ORG_NAME'
cache_type               'BasicFile'
cache_options( :path => "#{ENV['HOME']}/.chef/checksums" )
cookbook_path            ["#{current_dir}/../cookbooks"]
```

knife ssl check

```
root@workstation:/home/ubuntu/chef-repo# knife ssl check
Connecting to host ec2-13-127-234-63.ap-south-1.compute.amazonaws.com:443
ERROR: The SSL certificate of ec2-13-127-234-63.ap-south-1.compute.amazonaws.com could not be verified
Certificate issuer data: /C=US/O=YouCorp/OU=Operations/CN=chefserver

Configuration Info:

OpenSSL Configuration:
* Version: OpenSSL 1.0.2y  16 Feb 2021
* Certificate file: /opt/chef-workstation/embedded/ssl/cert.pem
* Certificate directory: /opt/chef-workstation/embedded/ssl/certs
Chef Infra Client SSL Configuration:
* ssl_ca_path: nil
* ssl_ca_file: nil
* trusted_certs_dir: "/home/ubuntu/chef-repo/.chef/trusted_certs"
```

To Conform the config.rb is set up correctly by running the below command

```
# knife client list
```

```
root@workstation:/home/ubuntu/chef-repo/cookbooks# knife client list
test-org-validator
root@workstation:/home/ubuntu/chef-repo/cookbooks#
```

Bootstrap the client node

The bootstrapping process involves setting up a Chef Client on a node. Chef Client Communicates with the Chef Server to receive directions for its own configuration. After the client receives the policy, it applies to the node to ensure the client is configured as per the directions of the server.

Knife-ec2 is official Chef Knife plugin for EC2. This plugin gives knife the ability to create, bootstrap, and manage EC2 instances.

Note: Workstation need to have proper iam role to provision ec2-machine.

Install the knife-ec2 plugin on your workstation

```
#apt-get install gcc g++ make autoconf

#apt-get install libxml2 libxml2-dev libxslt1-dev

#/opt/chef-workstation/embedded/bin/gem install knife-ec2 –V
```

```
root@workstation:/home/ubuntu# /opt/chef-workstation/embedded/bin/gem install knife-ec2 -V
Fetching knife-ec2-2.1.4.gem
Successfully installed knife-ec2-2.1.4
```

#1: Knife command to Provision the ec2

```
knife ec2 server create --image ami-04bde106886a53080 --flavor t2.micro --region ap-south-1 --ssh-user ubuntu --ssh-key chef-server -i /home/ubuntu/chef-server.pem --subnet subnet-2fe3a263  --node-name final_1  -Z ap-south-1b -g sg-7bc1f21d -g sg-0ee1f53f1092ebd58 --associate-public- --run-list 'recipe[test_cookbook]' --sudo --node-ssl-verify-mode none
```

Installation of chef automate

Login in Back to Chef Server and install the chef automate this will configure the chef sever Dashboard

#1: Download Chef automate

wget https://packages.chef.io/files/current/latest/chef-automate-cli/chef-automate_linux_amd64.zip

```
root@chefserver:/home/ubuntu# wget https://packages.chef.io/files/current/latest/chef-automate-cli/chef-automate_linux_amd64.zip
--2021-05-06 09:52:04--  https://packages.chef.io/files/current/latest/chef-automate-cli/chef-automate_linux_amd64.zip
Resolving packages.chef.io (packages.chef.io)... 151.101.154.110
Connecting to packages.chef.io (packages.chef.io)|151.101.154.110|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11454997 (11M) [application/zip]
Saving to: 'chef-automate_linux_amd64.zip.1'

chef-automate_linux_amd64.zip.1         100%[=======================================================================================>]

2021-05-06 09:52:04 (105 MB/s) - 'chef-automate_linux_amd64.zip.1' saved [11454997/11454997]

root@chefserver:/home/ubuntu#
```

#2: Configuration of Chef automate

Configuration of the chef automate below are the commands

```
apt install unzip
unzip chef-automate_linux_amd64.zip
mv chef-automate /usr/local/bin/
chmod +x /usr/local/bin/chef-automate
chef-automate deploy --accept-terms-and-mlsa
chef-manage-ctl reconfigure
```

Tomcat Chef Cookbook

#1: Clone this cookbook:

Git repo : https://github.com/natrajnix/tomcat.git

This cookbook will install & configure Java and Tomcat includes the tomcat service, complies and deploy the app to tomcat server

Usage of the Cookbook:

Add 'apt' cookbook to the workstation

```
knife supermarket download apt
```

Upload Cookbook's to the Chef Server

```
knife cookbook upload tomcat
knife cookbook upload apt
```

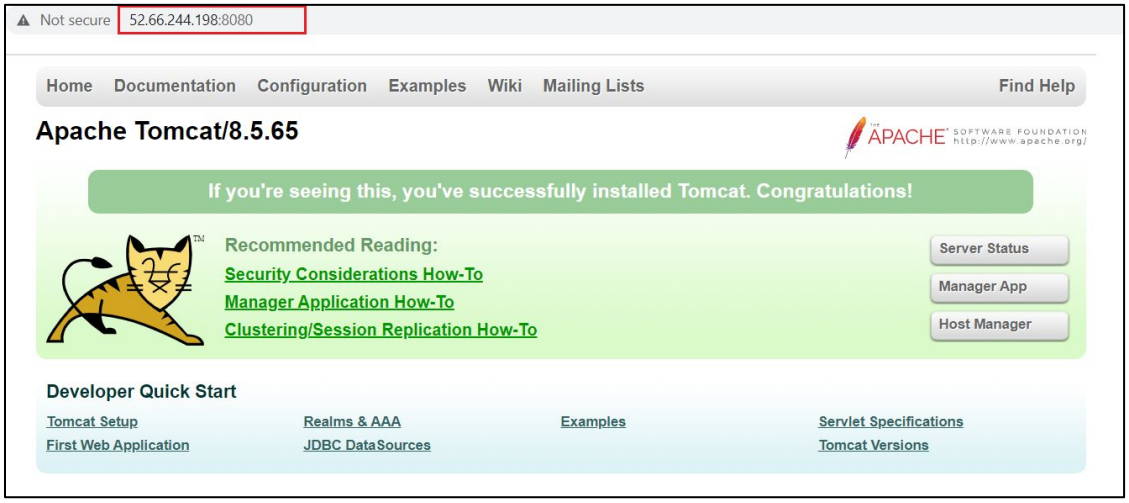
```
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks# knife cookbook upload tomcat
Uploading tomcat          [0.1.0]
Uploaded 1 cookbook.
root@ip-172-31-6-149:/home/ubuntu/chef-repo/cookbooks#
```

Bootstrap the using the knife ec2 create

```
knife ec2 server create --image ami-04bde106886a53080 --flavor t2.micro --region ap-south-1 --ssh-user ubuntu --ssh-key chef-server -i /home/ubuntu/chef-server.pem --subnet subnet-2fe3a263 --node-name tomcat_webapp_deploy_final  -Z ap-south-1b -g sg-7bc1f21d -g sg-0ee1f53f1092ebd58 --associate-public-ip --run-list 'recipe[apt],recipe[tomcat]' --sudo --node-ssl-verify-mode none
```

Once knife ec2 creation is successful Now in the browser hit the Public url this will load the default tomcat page:

Public ip : x.x.x.x :8080



Application: Page

Public ip : x.x.x.x:8080/hello/sayhello