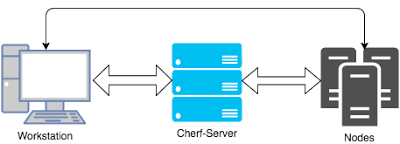
### **Chef**

Chef is an open source automation tool to manage your infrastructure as code. Provision, orchestrate and do configuration management without worrying about the type of cloud and scale of your infrastructure.

### **Chef Architecture**

[](https://2.bp.blogspot.com/-SZ5H-fg4eRU/V8h3P4brULI/AAAAAAAADLI/lT6UoyU6NDsZ37zsvSP0oTa8SNCp0WNkgCLcB/s1600/Untitled+Diagram.png)

### **Setup Chef workstation**

* [Install Chef development kit](https://downloads.chef.io/chef-dk/)

mkdir ~/mychefworkstation

Install chek-dk from below url on Radhat

 wget **https**://**packages**.chef.io/files/stable/**chefdk**/1.3.43/**el**/7/**chefdk**-1.3.43-1.el7.x86\_64.rpm

sudo rpm -ivh **chefdk**-1.3.43-1.el7.x86\_64.rpm

### **Chef Server**

#### **Install chef-server**

* SH the instance on which you want to setup chef server and run the following command

curl -L https://omnitruck.chef.io/install.sh | sudo bash -s -- -P chef-server

* Get a text editor (VIM)
* Create chef server configuration file  ***/etc/opscode/chef-server.rb*** (if not present) and add the following settings with a proper domain

server\_name = "chef\_server\_domain"

api\_fqdn server\_name

bookshelf['vip'] = server\_name

nginx['url'] = "https://#{server\_name}"

nginx['server\_name'] = server\_name

nginx['ssl\_certificate'] = "/var/opt/opscode/nginx/ca/#{server\_name}.crt"

nginx['ssl\_certificate\_key'] = "/var/opt/opscode/nginx/ca/#{server\_name}.key"

* Now apply the configuration by running the following command

sudo chef-server-ctl reconfigure

#### **Install the management console and reporting features**

* The **management console** is the web-based interface into Chef server
* **Chef reporting** tracks what happens when chef-client runs on your nodes. Chef server uses this information to build reports.
* **Run the following commands on your Chef server to install the management console.**

sudo chef-server-ctl install chef-manage

sudo chef-server-ctl reconfigure

sudo chef-manage-ctl reconfigure

**Then run these commands to install the reporting feature**

sudo chef-server-ctl install opscode-reporting

sudo chef-server-ctl reconfigure

sudo opscode-reporting-ctl reconfigure

#### **Create admin user Syntax command:**

sudo chef-server-ctl user-create ADMIN\_USER\_NAME ADMIN\_FIRST\_NAME ADMIN\_LAST\_NAME ADMIN\_EMAIL ADMIN\_PASSWORD --filename ADMIN\_USER\_NAME.pem

**Actual command:**

 sudo chef-server-ctl user-create jsmith Joe Smith joe.smith@example.com p4ssw0rd --filename jsmith.pem

This will generate an RSA private key (.pem) file in the current directory, which allows us to run knife commands against chef server as an authenticated user.

In the coming step we will copy this file to our workstation.

#### **Create the organization**

**command syntax:**

sudo chef-server-ctl org-create ORG\_SHORT\_NAME "ORG\_LONG\_NAME" --association\_user ADMIN\_USER\_NAME

**Actual command:**  
  
sudo chef-server-ctl org-create 4thcoffee "Fourth Coffee, Inc." --association\_user jsmith

### **Setup knife on workstation by downloading starter kit from chef server**

Knife is a command line tool to interact with chef server and nodes for example uploading the chef code i.e. cookbooks to chef server.

From your workstation:

* In web browser, hit the domain for which chef server was configured
* Login with the username and password you provided in the previous step.
* From the Administration tab, select your organization.
* Select Starter Kit from the menu on the left.
* Click the Download Starter Kit button.
* Click Proceed. Save the file chef-starter.zip to your computer.
* Extract chef-starter.zip to your ~/mychef directory.

You are free to choose a directory other than ~/mychef

Now verify that the ***~/mychef/chef-repo/.chef*** directory on your workstation contains the knife configuration file and your RSA key.  
ls ~/mychef/chef-repo/.chef

#### **Download SSL certificate**

The communication between node and chef server is done over HTTPS(port 443) .  
During bootstrap process knife copies SSL certificate from workstation to node. To be able to copy certificate, the workstation should have the copy of SSL certificate.

From your ***~/mychef/chef-repo*** directory, run the ***knife ssl fetch*** command to retrieve a copy of the certificate.

knife ssl fetch

#### **Test the connection to Chef server**

Run on workstation to check the connection with chef-server

knife client list

## **Bootstrap a node with chef**

## **Create First Cookbook in Chef**

Before bootstrapping a node, will first create a cookbook which will install **tmux** on the node while bootstrapping. If you are confused with the term bootstrap, It is a process of installing chef-client on the node so that it can communicate to the chef-server.

Let's create a cookbook. All cookbooks will be stored in the directory **cookbooks**

mkdir ~/mychef/chef-repo/cookbooks

Now go to mychef directory  
cd ~/mychef/chef-repo

Generate a cookbook **common\_packages**, which will have installation of basic packages on the node.

chef generate cookbook cookbooks/common\_packages

This will generate the basic scaffolding which is as follow:

cookbooks

├── chefignore

├── common\_packages

│ ├── Berksfile

│ ├── README.md

│ ├── chefignore

│ ├── metadata.rb

│ ├── recipes

│ │ └── default.rb

│ ├── spec

│ │ ├── spec\_helper.rb

│ │ └── unit

│ │ └── recipes

│ │ └── default\_spec.rb

│ └── test

│ └── recipes

│ └── default\_test.rb

└── starter

├── attributes

│ └── default.rb

├── files

│ └── default

│ └── sample.txt

├── metadata.rb

├── recipes

│ └── default.rb

└── templates

└── default

└── sample.erb

Let's write a recipe (a part of cookbook which do some tasks) in ***~/mychef/chef-repo/cookbooks/common\_packages/recipes/default.rb***. Write the following code to install**tmux** package.

package 'tmux'

Let's create a role now which would be applied to all the node as it will contain the runlist to be applied which is common to all node.

vi ~/mychef/chef-repo/roles/common.json

Put the following content in it

{

"name": "common",

"description": "This role installs the common packages to be present on all servers",

"run\_list": [

"recipe[common\_packages]"

]

}

We have created the cookbook, now we will upload the cookbook to chef-server

knife cookbook upload common\_packages

To verify if uploaded, run:

knife cookbook list

Go to ***~/mychef/chef-repo*** directory and run the following command to bootstrap a node:

**Syntax:**  
knife bootstrap ADDRESS --ssh-user USER --sudo --identity-file IDENTITY\_FILE --node-name node1 --run-list 'recipe[COOKBOOK\_NAME]'

**Actual Command:**  
knife bootstrap X.X.X.X --ssh-user ec2-user --sudo --identity-file ~/.ssh/pemfile/ajeet.pem --node-name api-server --run-list 'recipe[common\_packages]'

This will bootstrap the node. Now check if the node is being registered to chef-server or not by running following command

knife node list

This will give you the name of the node given while bootstrapping it. You can also verify the node is registered to chef-server or not by seeing nodes tab in chef-web-UI.

Also SSH on node and check if the required package '**tmux**' is installed or not.