CHEF

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

Chef is a configuration management technology developed by Opscode to manage infrastructure on physical or virtual machines. It is an open source developed using Ruby, which helps in managing complex infrastructure on the fly.

This tutorial provides a basic understanding of the infrastructure and fundamental concepts of managing an infrastructure using Chef.

Audience

This tutorial has been prepared for those who want to understand the features and functionality of Chef and how Chef can help in reducing the complexity of managing an infrastructure.

After completing this tutorial one would have moderate level understanding of Chef and its key building blocks. It will also give a fair idea on how to configure Chef in a preconfigured infrastructure and how to use it.

Prerequisites

We assume anyone who wants to learn Chef should have an understanding of system administration, infrastructure and network protocol communication. To automate the infrastructure provisioning, one should have a command over basic Ruby script writing and the underlying system where one wants to use Chef.

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

Basic Chef

1. Chef – Overview

Chef is an open source technology developed by Opscode. Adam Jacob, co-founder of Opscode is known as the founder of Chef. This technology uses Ruby encoding to develop basic building blocks like recipe and cookbooks. Chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.

Chef have got its own convention for different building blocks, which are required to manage and automate infrastructure.

Why Chef?

Chef is a configuration management technology used to automate the infrastructure provisioning. It is developed on the basis of 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.

Features of Chef

Following are the most prominent features of Chef:

- Chef uses popular Ruby language to create a domain-specific language.
- Chef does not make assumptions on the current status of a node. It uses its mechanisms to get the current status of machine.
- Chef is ideal for deploying and managing the cloud server, storage, and software.

Advantages of Chef

Chef offers the following advantages:

- **Lower barrier for entry**: As Chef uses native Ruby language for configuration, a standard configuration language it can be easily picked up by anyone having some development experience.
- **Excellent integration with cloud**: Using the knife utility, it can be easily integrated with any of the cloud technologies. It is the best tool for an organization that wishes to distribute its infrastructure on multi-cloud environment.

Disadvantages of Chef

Some of the major drawbacks of Chef are as follows:

- One of the huge disadvantages of Chef is the way cookbooks are controlled. It needs constant babying so that people who are working should not mess up with others cookbooks.

- Only Chef solo is available.

- In the current situation, it is only a good fit for AWS cloud.

- It is not very easy to learn if the person is not familiar with Ruby.

- Documentation is still lacking.

Key Building Blocks of Chef

Recipe

It can be defined as a collection of attributes which are used to manage the infrastructure. These attributes which are present in the recipe are used to change the existing state or setting a particular infrastructure node. They are loaded during Chef client run and comparted with the existing attribute of the node (machine). It then gets to the status which is defined in the node resource of the recipe. It is the main workhorse of the cookbook.

Cookbook

A cookbook is a collection of recipes. They are the basic building blocks which get uploaded to Chef server. When Chef run takes place, it ensures that the recipes present inside it gets a given infrastructure to the desired state as listed in the recipe.

Resource

It is the basic component of a recipe used to manage the infrastructure with different kind of states. There can be multiple resources in a recipe, which will help in configuring and managing the infrastructure. For example -

- **package**: Manages the packages on a node
- **service**: Manages the services on a node
- **user**: Manages the users on the node
- **group**: Manages groups
- **template**: Manages the files with embedded Ruby template
- **cookbook_file**: Transfers the files from the files subdirectory in the cookbook to a location on the node

- **file**: Manages the contents of a file on the node
- **directory**: Manages the directories on the node
- **execute**: Executes a command on the node
- **cron**: Edits an existing cron file on the node

Attribute

They are basically settings. They can be thought of as a key value pair of anything which one wants to use in the cookbook. There are several different kinds of attributes that can be applied, with a different level of precedence over the final settings that the node operates under.

File

It's a subdirectory within the cookbook that contains any static file which will be placed on the nodes that uses the cookbooks. A recipe then can be declared as a resource that moves the files from that directory to the final node.

Templates

They are similar to files, but they are static. Template files end with the **.ebr** extension, which means they contain embedded Ruby. They are mainly used to substitute an attribute value into the files to create the final file version that will be placed on the node.

Metadata.rb

It is used to manage the metadata about the package. This includes details like the name and details of the package. It also includes things such as dependency information that tells which cookbooks this cookbook needs to operate. This allows the Chef server to build the run-list of the node correctly and ensures that all of the pieces are transferred correctly.

Default Cookbook Structure

```
C:\chef\cookbooks\nginx>tree
Folder PATH listing for volume Local Disk
Volume serial number is BE8B-6427
C:
├───attributes
├───definitions
├───files
│   └───default
```

```
├──────libraries

├──────providers

├──────recipes

├──────resources

└──────templates

    └──────default
```

Chef –hRelated Technologies

Following is the list of Chef related technologies.

Puppet

Puppet provides a standard way of delivering and operating software, no matter where it runs. It is an automated administrative engine for Linux, Unix, and Windows system that performs administrative tasks based on centralized specification.

The primary **features of Puppet** are as follows:

- Implementing new systems with a uniform configuration.
- Updating the systems and upgrading the security and software packages.
- Incorporating new features and adding dexterous capabilities.
- Customizing configurations for ensuring the availability of data sources.
- Optimizing the available resources and minimizing the cost.
- Simplifying the roles and enabling the team to focus on core and productive issues.
- Getting a bird’s eye view of the available infrastructure.

Ansible

Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications — automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

The primary **features of Ansible** are as follows:

- Simple and easy to learn
- Written in Python

- Agentless
- YAML-based Playbooks
- Ansible galaxy

SaltStack

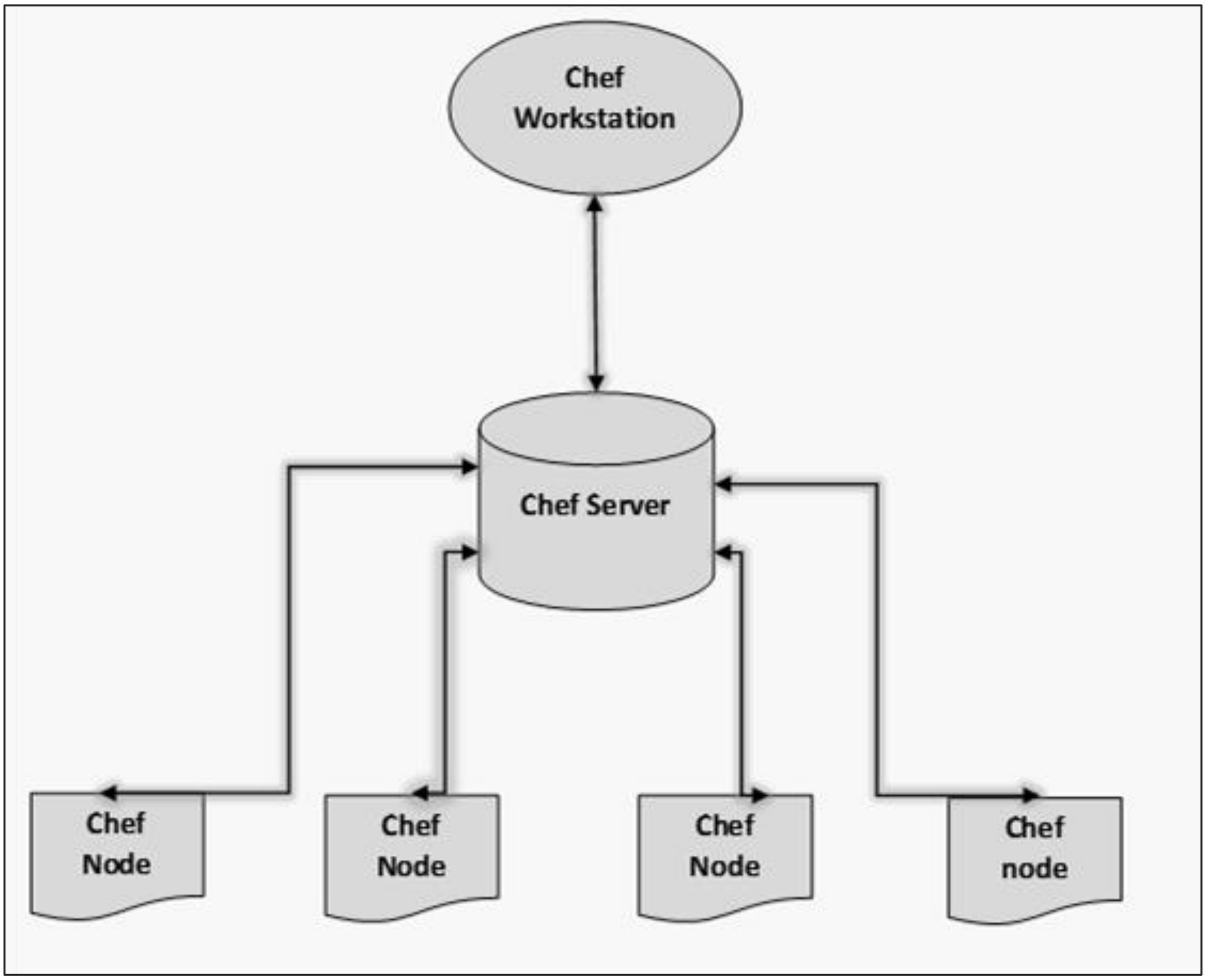
SaltStack is used for data-driven configuration. It is a new approach of infrastructure management built on dynamic communication bus. It is used for data-driven orchestration, remote execution for any infrastructure, and configuration management for any app stack.

Fabric

Fabric is a Python-based programming language, which is developed as an API of Python which needs to be imported in Python code in order to configure and manage an infrastructure.

2. Chef – Architecture

Chef works on a three-tier client server model wherein the working units such as cookbooks are developed on the Chef workstation. From the command line utilities such as knife, they are uploaded to the Chef server and all the nodes which are present in the architecture are registered with the Chef server.



In order to get the working Chef infrastructure in place, we need to set up multiple things in sequence.

In the above setup, we have the following components.

Chef Workstation

This is the location where all the configurations are developed. Chef workstation is installed on the local machine. Detailed configuration structure is discussed in the later chapters of this tutorial.

Chef Server

This works as a centralized working unit of Chef setup, where all the configuration files are uploaded post development. There are different kinds of Chef server, some are hosted Chef server whereas some are built-in premise.

Chef Nodes

They are the actual machines which are going to be managed by the Chef server. All the nodes can have different kinds of setup as per requirement. Chef client is the key component of all the nodes, which helps in setting up the communication between the Chef server and Chef node. The other components of Chef node is Ohai, which helps in getting the current state of any node at a given point of time.

3. Chef – Version Control System Setup

Using Version Control system is a fundamental part of infrastructure automation. There are multiple kinds of version control system such as SVN, CVS, and GIT. Due to the popularity of GIT among the Chef community, we will use the GIT setup.

Note: Don't think of building an infrastructure as a code without a version control system.

On Windows

Step 1: Download the Windows installer from www.git-scm.org and follow the installation steps.

Step 2: Sign up for a central repository on GitHub.

Step 3: Upload the ssh key to the GitHub account, so that one can interact with it easily. For details on ssh key visit the following link https://help.github.com/articles/generating-ssh-keys.

Step 4: Finally create a repo on the github account by visiting https://github.com/new with the name of chef-repo.

Before actually starting to write a cookbook, one can set up an initial GIT repository on the development box and clone the empty repository provided by Opscode.

Step 1: Download Opscode Chef repository empty structure.

```
$ wget https://github.com/opscode/chef-repo/tarball/master
```

Step 2: Extract the tar ball.

```
$ tar –xvf master
```

Step 3: Rename the directory.

```
$ mv opscode-chef-repo-2c42c6a/ chef-repo
```

Step 4: Change the current working directory to chef repo.

```
$ cd chef-repo
```

Step 5: Initialize a fresh get repo.

```
$ git init .
```

Step 6: Connect to your repo on the git hub.

```
$ git remote add origin git@github.com:vipin022/chef-
```

Step 7: Push the local repo to github.

```
$ git add .
$ git commit –m “empty repo structure added”
$ git push –u origin master
```

By using the above procedure, you will get an empty chef repo in place. You can then start working on developing the recipes and cookbooks. Once done, you can push the changes to the GitHub.

4. Chef – Workstation Setup

Chef follows the concept of client-server architecture, hence in order to start working with Chef one needs to set up Chef on the workstation and develop the configuration locally. Later it can be uploaded to Chef server to make them working on the Chef nodes, which needs to be configured.

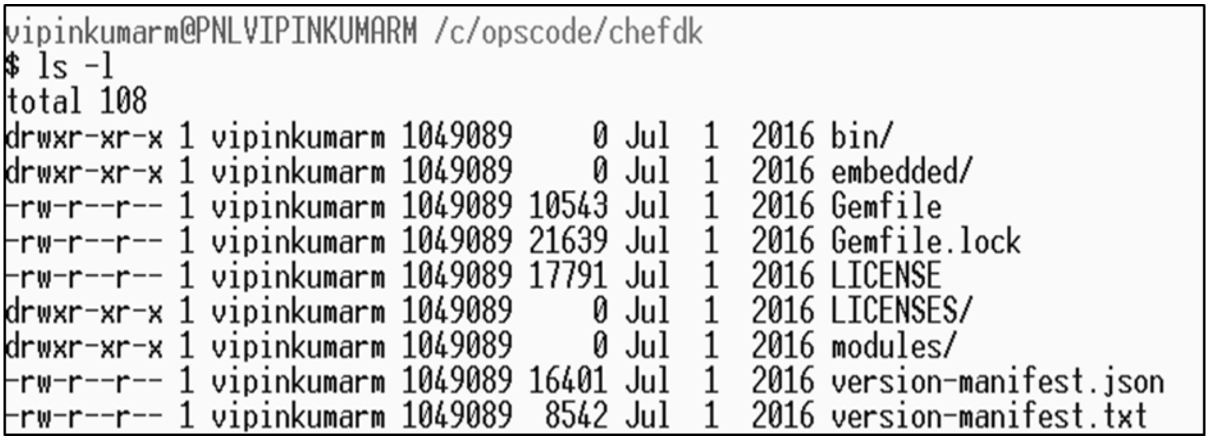
Opscode provides a fully packaged version, which does not have any external prerequisites. This fully packaged Chef is called the **omnibus installer.**

On Windows Machine

Step 1: Download the setup .msi file of chefDK on the machine.

Step 2: Follow the installation steps and install it on the target location.

The setup will look as shown in the following screenshot.



```
vipinkumarm@PNLVIPINKUMARM /c/opscode/chefdk
$ ls -l
total 108
drwxr-xr-x 1 vipinkumarm 1049089     0 Jul  1  2016 bin/
drwxr-xr-x 1 vipinkumarm 1049089     0 Jul  1  2016 embedded/
-rw-r--r-- 1 vipinkumarm 1049089 10543 Jul  1  2016 Gemfile
-rw-r--r-- 1 vipinkumarm 1049089 21639 Jul  1  2016 Gemfile.lock
-rw-r--r-- 1 vipinkumarm 1049089 17791 Jul  1  2016 LICENSE
drwxr-xr-x 1 vipinkumarm 1049089     0 Jul  1  2016 LICENSES/
drwxr-xr-x 1 vipinkumarm 1049089     0 Jul  1  2016 modules/
-rw-r--r-- 1 vipinkumarm 1049089 16401 Jul  1  2016 version-manifest.json
-rw-r--r-- 1 vipinkumarm 1049089  8542 Jul  1  2016 version-manifest.txt
```

ChefDK Path Variable

```
$ echo $PATH
/c/opscode/chef/bin:/c/opscode/chefdk/bin:
```

On Linux Machine

In order to set up on the Linux machine, we need to first get curl on the machine

Step 1: Once curl is installed on the machine, we need to install Chef on the workstation using Opscode's omnibus Chef installer.

```
$ curl –L https://www.opscode.com/chef/install.sh | sudo bash
```

Step 2: Install Ruby on the machine.

Step 3: Add Ruby to path variable.

```
$ echo ‘export PATH=”/opt/chef/embedded/bin:$PATH”’ >> ~/.bash_profile && source
~/.bash_profile
```

The Omnibus Chef will install Ruby and all the required Ruby gems into **/opt/chef/embedded** by adding **/opt/chef/embedded/bin** directory to the .bash_profile file.

If Ruby is already installed, then install the Chef Ruby gem on the machine by running the following command.

```
$ gem install chef
```