

What is Chef?

“Chef turns infrastructure into code. With Chef, you can automate how you build, deploy, and manage your infrastructure. Your infrastructure becomes as versionable, testable, and repeatable as application code.”

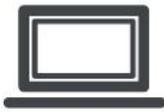
Chef relies on reusable definitions known as recipes to automate infrastructure tasks. Examples of recipes are instructions for configuring web servers, databases and load balancers. Together, recipes describe what your infrastructure consists of and how each part of your infrastructure should be deployed, configured and managed.”

- Packages to be installed
- Files to be created
- Directories to be created
- Services to be started
- Config files to be updated
- Commands to be executed

Chef Components

Chef has the following major components:

Component	Description
-----------	-------------



workstation

One (or more) workstations are configured to allow users to author, test, and maintain cookbooks. Cookbooks are uploaded to the Chef server from the workstation. Some cookbooks are custom to the organization and others are based on community cookbooks available from the Chef Supermarket.



cookbook

Ruby is the programming language that is the authoring syntax for cookbooks. Most recipes are simple patterns (blocks that define properties and values that map to specific configuration items like packages, files, services, templates, and users). The full power of Ruby is available for when you need a programming language.



Ruby

Often, a workstation is configured to use the Chef development kit as the development toolkit. The Chef development kit is a package from Chef that provides an optional (but recommended) set of tooling, including Chef itself, the chef command line tool, Kitchen, ChefSpec, Berkshelf, and more.



node

A node is any machine—physical, virtual, cloud, network device, etc.—that is under management by Chef.

A chef-client is installed on every node that is under management by Chef. The chef-client performs all of the configuration tasks that are specified by the run-list and will pull down any required configuration data from the Chef server as it is needed during the chef-client run.



Chef Client



Chef Server

The Chef server acts as a hub of information. Cookbooks and policy settings are uploaded to the Chef server by users from workstations. (Policy settings may also be maintained from the Chef server itself, via the Chef management console web user interface.)

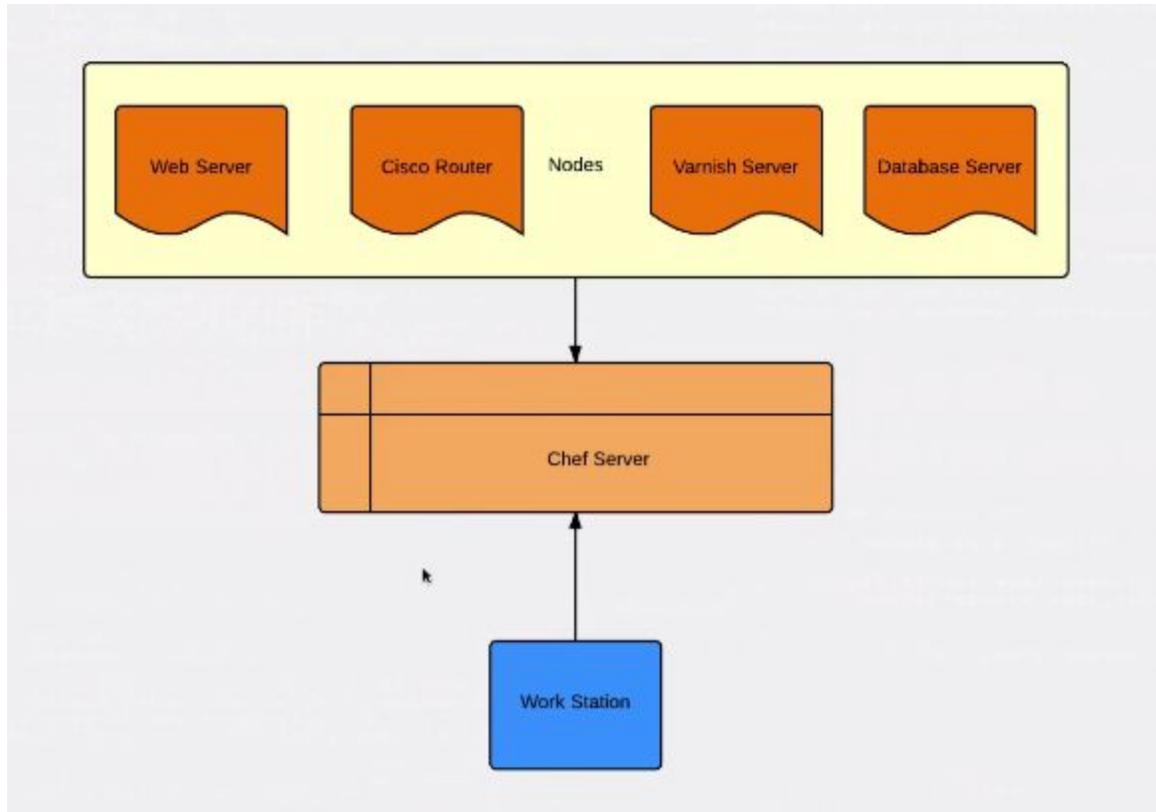
The chef-client accesses the Chef server from the node on which it's installed to get configuration data, perform searches of historical chef-client run data, and then pull down the necessary configuration data. After the chef-client run is finished, the chef-client uploads updated run data to the Chef server as the updated node object.

Chef management console is the user interface for the Chef server. It is used to manage data bags, attributes, run-lists, roles, environments, and cookbooks, and also to configure role-based access for users and groups.



Chef Supermarket

Chef Supermarket is the location in which community cookbooks are authored and maintained. Cookbooks that are part of the Chef Supermarket may be used by any Chef user. How community cookbooks are used varies from organization to organization.



Chef Terminology

Common Chef Terminology

Recipes:

Fundamental configuration element within an organization

Cookbook:

Defines a scenario and is the fundamental unit of configuration and policy distribution

Chef-Client:

Agent that runs locally on the node that is registered with the chef server

Convergence:

Occurs when chef-client configures the system/node based off the information collected from chef-server

Configuration Drift:

Occurs when the node state does not reflect the updated state of polices/configurations on the chef server

Resources:

A statement of configuration policy within a recipe

Describes the desired state of an element in the infrastructure and steps needed to configure

Provider:

Defines the steps that are needed to bring the piece of the system from its current state to the desired state

Attributes:

Specific details about the node, used by chef-client to understand current state of the node, the state of the node on the previous chef-client run, and the state of the node at the end of the client run

Data-bags:

A global variables stored as JSON data and is accessible from the Chef server

Workstation:

A computer configured with Knife and used to synchronize with chef-repo and interact with chef server

Chef Server:

Chef server is the hub for all configuration data, stores cookbooks, and the policies applied to the node

Knife:

Command line tool which provides an interface between the local chef-repo and chef-server

client.rb:

Configuration file for chef-client located at `/etc/chef/client.rb` on each node

Ohai:

Tool used to detect attributes on a node and then provide attributes to chef-client at the start of every chef-client run

Node Object:

Consists of run-list and node attributes that describe states of the node

Chef-Repo:

Located on the workstation and installed with the starter kit, should be synchronized with a version control system and stores Cookbooks, roles, data bags, environments, and configuration files

Organization:

Used in chef enterprise server to restrict access to objects, nodes environments, roles, data-bags etc

Environments:

Used to organize environments (Prod/Staging/Dev/QA) generally used with cookbook **versions**

Idempotence:

Means a recipe can run multiple times on the same system and the results will always be identical

Chef Server

Two types of Chef-server:

1. OpenSource Chef-server
 - Free version of Chef
 - Each instance of the server must be configured and managed locally (includes all aspects of managing the server, updates, migrations, scalability, etc).
2. Chef-server enterprise (hosted)
 - Scalable by design
 - Available organizations
 - Always available
 - Resource-based access control
3. Chef-server enterprise (on-premise)
 - Scalable by design
 - Available organizations
 - Hosted on-premise behind your firewall
 - Managed yourself

Chef Server: Enterprise

- Allows creation of organizations
 - Organizations separate the infrastructure, policies, and cookbooks
 - Nodes are registered in organizations
 - Nothing can be shared between organizations
 - Enterprise chef server can contain many different organizations
 - OpenSource chef the local individual server acts as an organization and does not allow creation of organizations
 - Organizations can represent different companies, department, infrastructures, applications, and so forth
- For each organization in order to start bootstrapping nodes you need to download the starter kit
- Starter kit provides security credentials (validation.pem keys) to authenticate each node to the chef server
- Chef enterprise scales by design to handle thousands of nodes and different organizations

Chef Server: Role of the server

- Stores system configuration information (policies for nodes)
- Authenticates workstations and nodes
- Delivers configurations to nodes
- Chef server holds the configuration and the node checks-in to receive instructions on its desired state
- The node downloads configuration instructions from the server and does all of the work