

# Chef

Ramit Surana  
@ramitsurana  
/in/ramitsurana



# Agenda

Introduction to Chef

Chef Architecture

Building Cookbook

Recipe

Knife

Chef Server

Chef-Client

Solving Cookbook Issues

RSpec

And much more...



**CHEF**™

# What is Chef ?

Chef is a automation provisioning tool.

It is a cloud infrastructure automation framework that makes it easy to deploy servers and applications to any physical, virtual, or cloud location, no matter the size of the infrastructure.

It has an advanced and better working principle than usage of regular shell scripts.

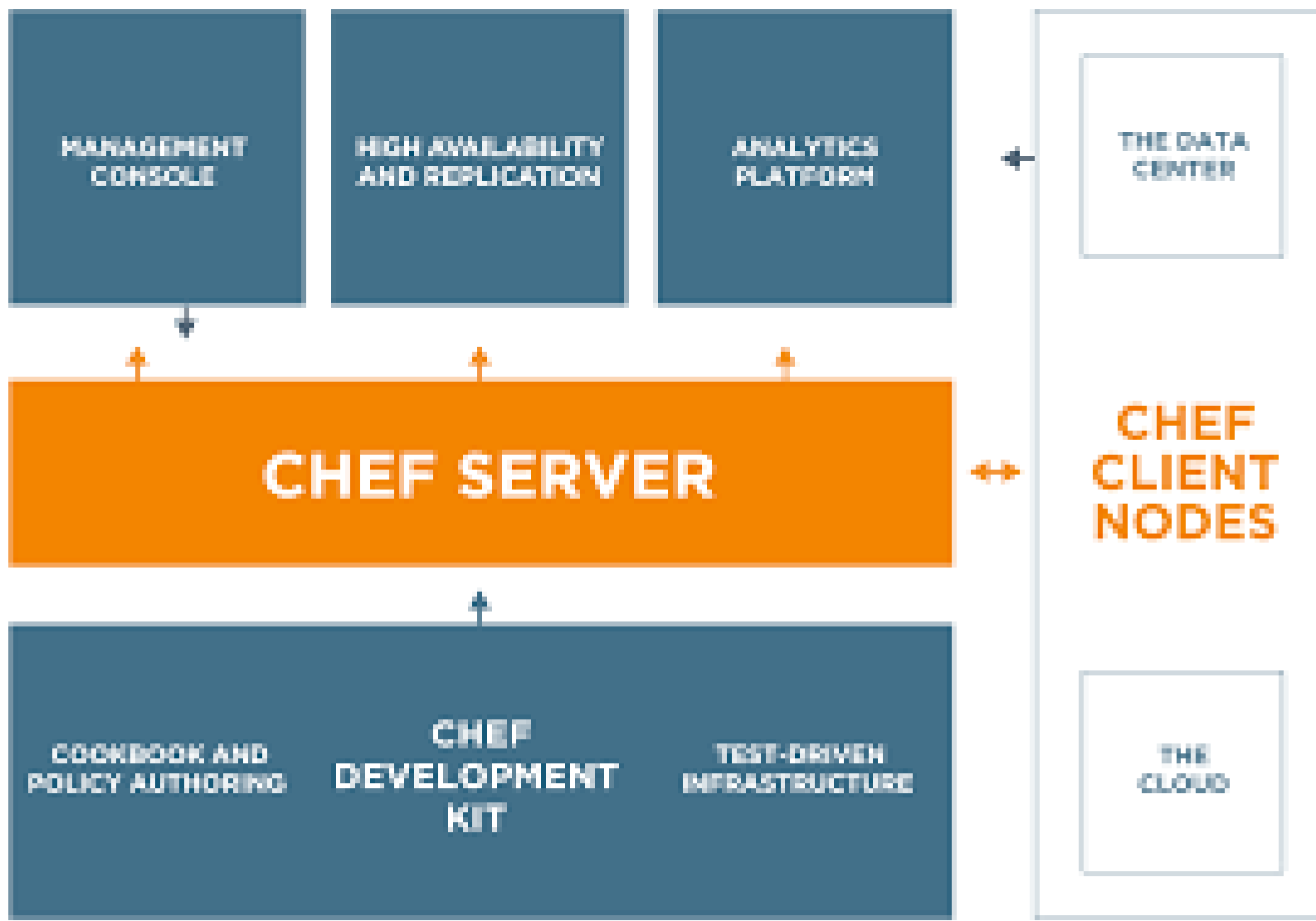


CHEF™



CHEF™

# Architecture



# Cookbook

The fundamental unit of configuration and policy distribution. A cookbook defines a scenario and contains everything that is required to support that scenario:

Recipes that specify the resources to use and the order in which they are to be applied

Attribute values

File distributions

Templates

Extensions to Chef, such as libraries, definitions, and custom resources



**CHEF**™

# Recipe

It is the most fundamental configuration element within the organization. A recipe:

- It Is authored using Ruby, which is a programming language designed to read and behave in a predictable manner

- Is mostly a collection of resources, defined using patterns (resource names, attribute-value pairs, and actions); helper code is added around this using Ruby, when needed

- Must define everything that is required to configure part of a system

- Must be stored in a cookbook

- May be included in a recipe

- May use the results of a search query and read the contents of a data bag (including an encrypted data bag)

- May have a dependency on one (or more) recipes

- May tag a node to facilitate the creation of arbitrary groupings

- Must be added to a run-list before it can be used by the chef-client

- Is always executed in the same order as listed in a run-list



**CHEF**™

# Knife

Chef's command-line tool called to interact with the Chef Server.

Use it for uploading cookbooks and managing other aspects of Chef.

command-line tool that provides an interface between a local chef-repo and the Chef server.  
knife helps users to manage:

Nodes, Cookbooks and recipes

Roles

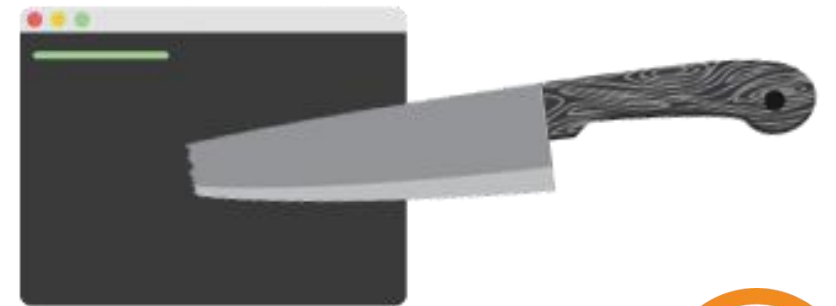
Stores of JSON data (data bags), including encrypted data

Environments, Cloud resources, including provisioning

The installation of the chef-client on management workstations

Searching of indexed data on the Chef server

## — SERVER PROVISIONING WITH — KNIFE SOLO



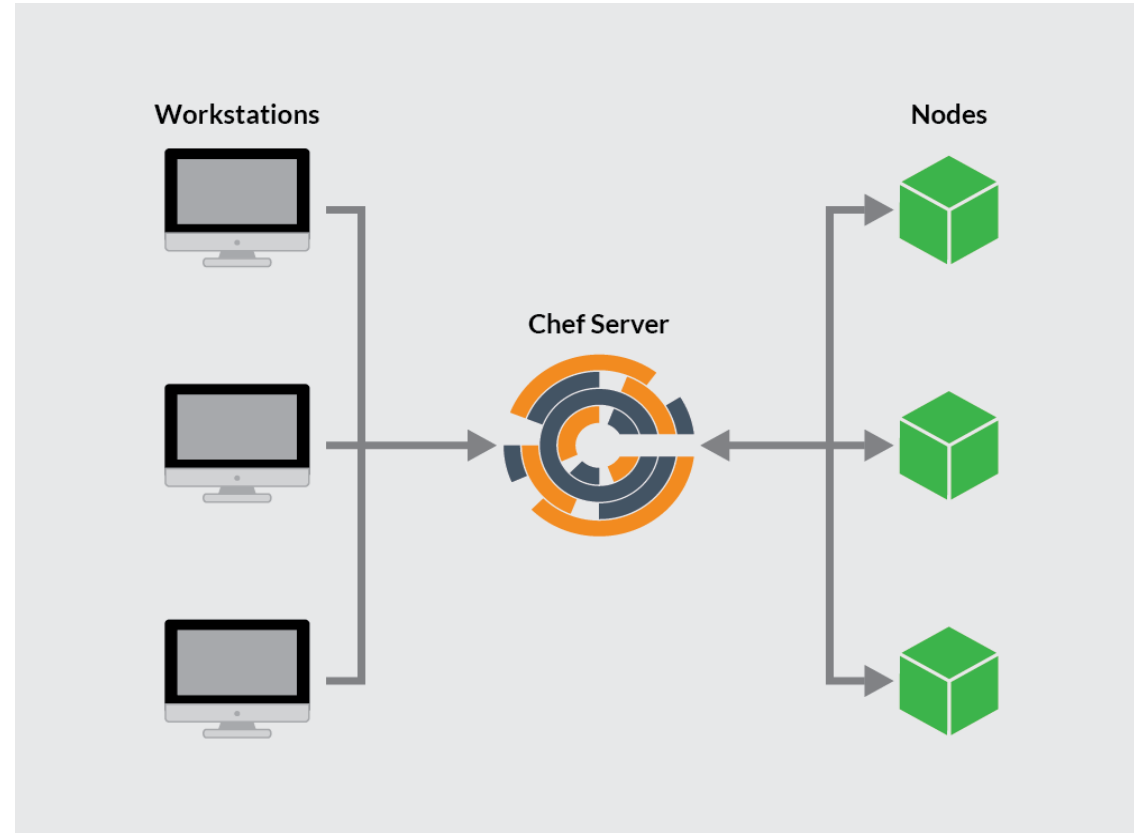
CHEF™

# Chef-Server

The Chef 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 Chef server for configuration details, such as recipes, templates, and file distributions.

The chef-client then does as much of the configuration work as possible on the nodes themselves





# Chef-Client

It is a provisioning which works on server.

Agent that runs locally on every node that is under management by Chef. 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 Chef server

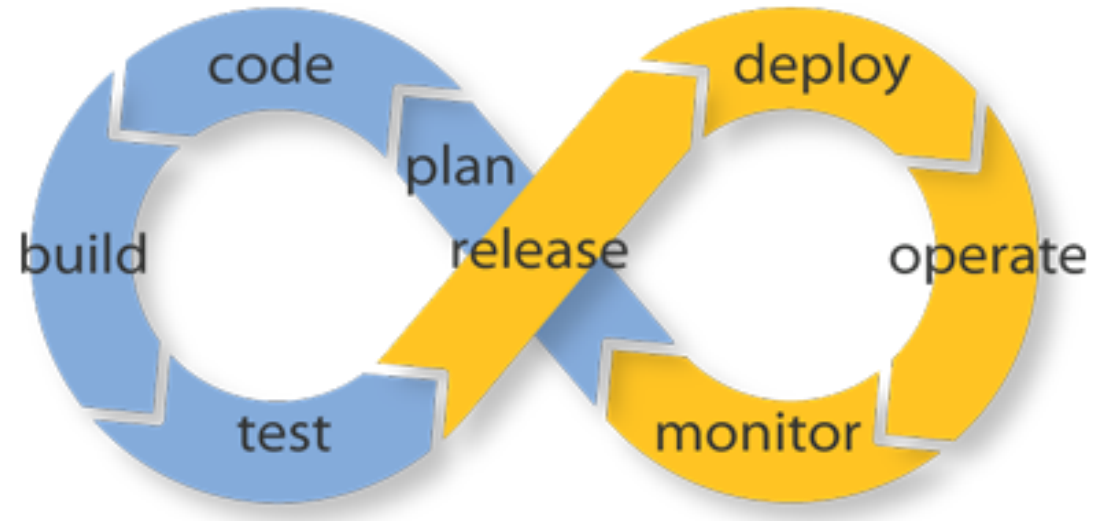
Building the node object

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

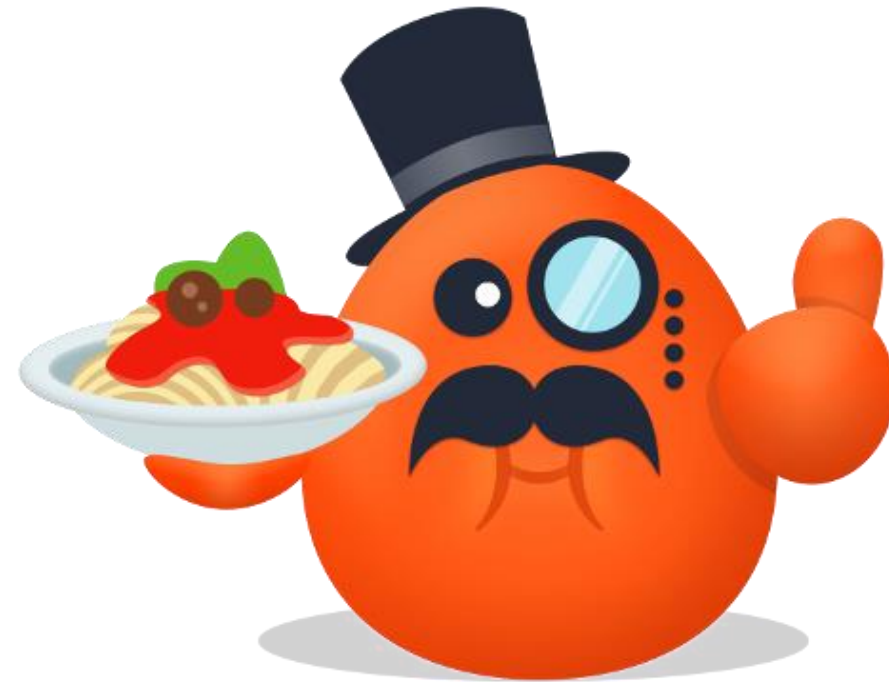
Looking for exceptions and notifications, handling each as required



# Solving Cookbook Issues: Foodcritic

Tries to identify possible issues with the logic and style of your cookbooks.

It comes with rules concerning various areas: style, correctness, attributes, strings, portability, search, services, files, metadata, and so on.



CHEF™

# RSpec Framework

Composed of multiple libraries, which are designed to work together, or can be used independently with other testing tools like Cucumber or Minitest.

The parts of RSpec are:

**rspec-core:** The spec runner, providing a rich command line program, flexible and customizable reporting, and an API to organize your code examples.

**rspec-expectations:** Provides a readable API to express expected outcomes of a code example.

**rspec-mocks:** Test double framework, providing multiple types of fake objects to allow you to tightly control the environment in which your specs run.

**rspec-rails:** Supports using RSpec to test Ruby on Rails applications in place of Rails' built-in test framework.



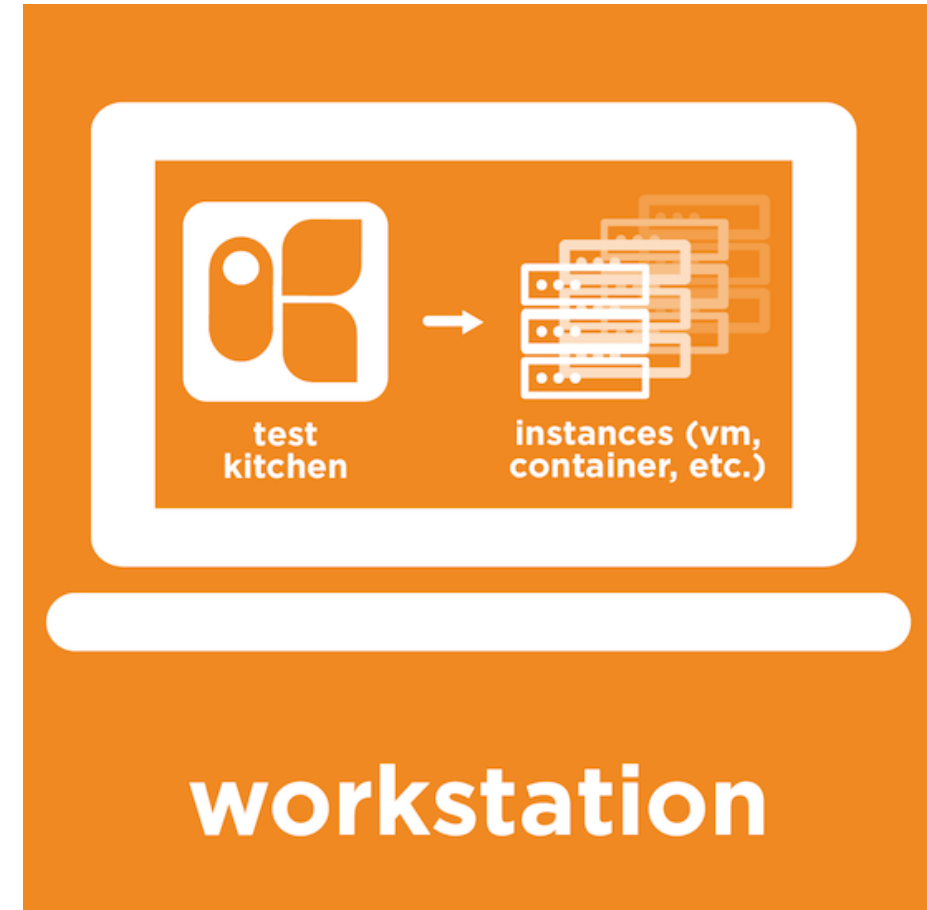
**CHEF**™

# Chef's Test Kitchen

Test harness tool to execute your configured code on one or more platforms in isolation.

A driver plugin architecture is used which lets you run your code on various cloud providers and virtualization technologies such as Amazon EC2, Blue Box, CloudStack, Digital Ocean, Rackspace, OpenStack, Vagrant, Docker, LXC containers, and more.

Many testing frameworks are already supported out of the box including Bats, shUnit2, RSpec, Serverspec, with others being created weekly.

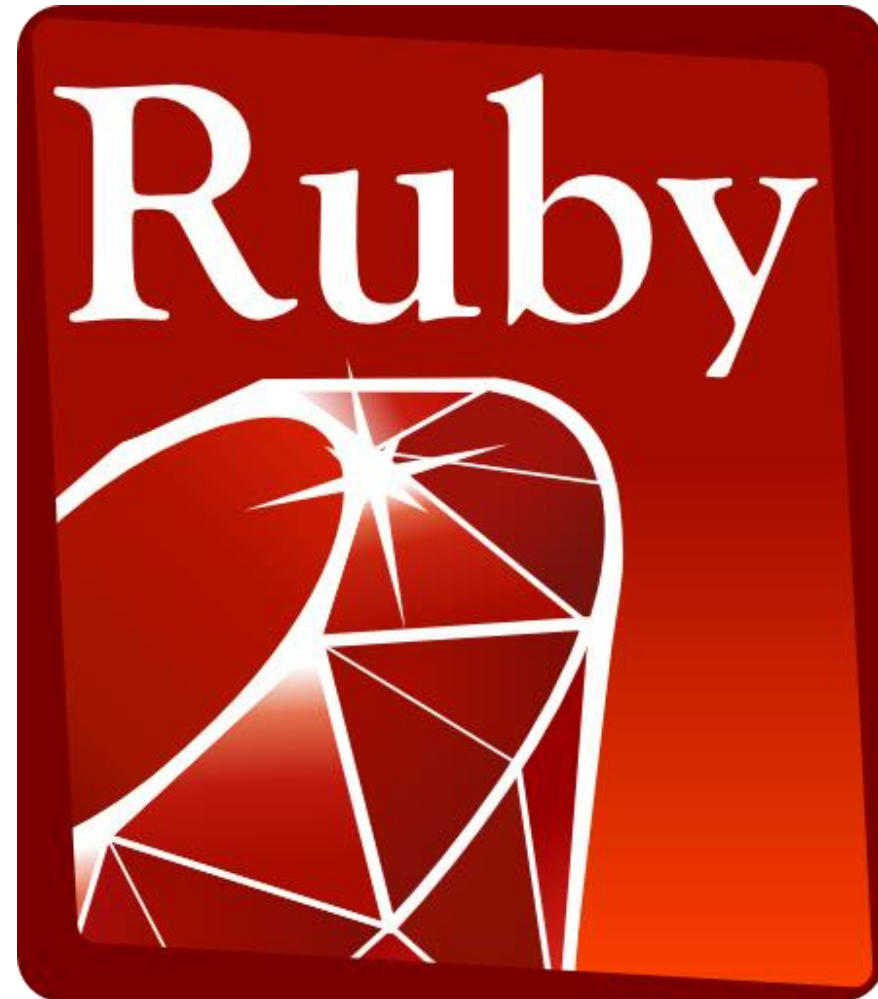


**CHEF**™

# Chef DSL(Domain Specific Language)

Recipe DSL helps ensure that recipes interact with nodes (and node properties) in the desired manner.

Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.



# Chef Analytics

Feature of Chef that provides real-time visibility into what is happening on the Chef server, what's changing, who made those changes, and when they occurred.

The relationships between the various elements of Chef Analytics, including how information is routed from various nodes to the Chef Analytics server (through the Chef server) nodes.

It is where reports about chef-client run outcomes may be viewed, where rules are processed, and where Chef Analytics data may be viewed.



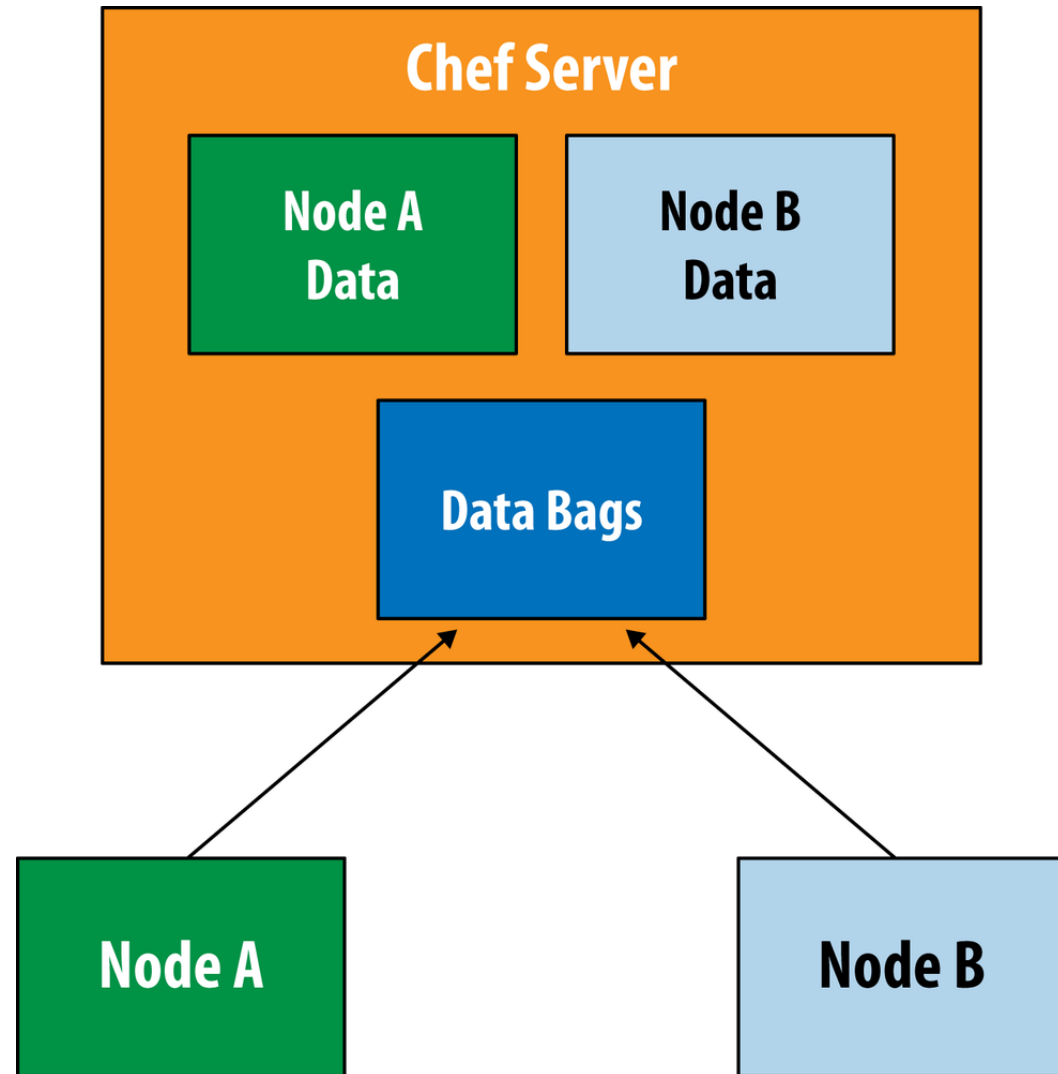
**CHEF**™

# Data Bags

Global variable that is stored as JSON data and is accessible from a Chef server.

It can be created in two ways: using knife or manually.

It helps you manage and configure nodes information on the server in an easy way.



# Chef v/s Puppet

## Chef

Uses a D.S.L but based on the Ruby syntax.

Per the Common Vulnerabilities and Exposures (CVE) database, Chef has a total of 3 reported vulnerabilities of medium severity.

Chef Delivery, a set of tools that add yet more developer-friendly features like comprehensive codebase change histories, metrics, and permissions management to the platform.

## Puppet

Uses a D.S.L (Domain Specific Language) which is a mixture between JSON and Ruby.

55 Puppet vulnerabilities have been documented with an average severity level of medium.

Puppet's Node Manager enables the creation of rules around node attributes, which allows for easier more efficient node management. Nodes can be managed based on their job rather than name, eliminating the need to manually classify each node.



**CHEF**™



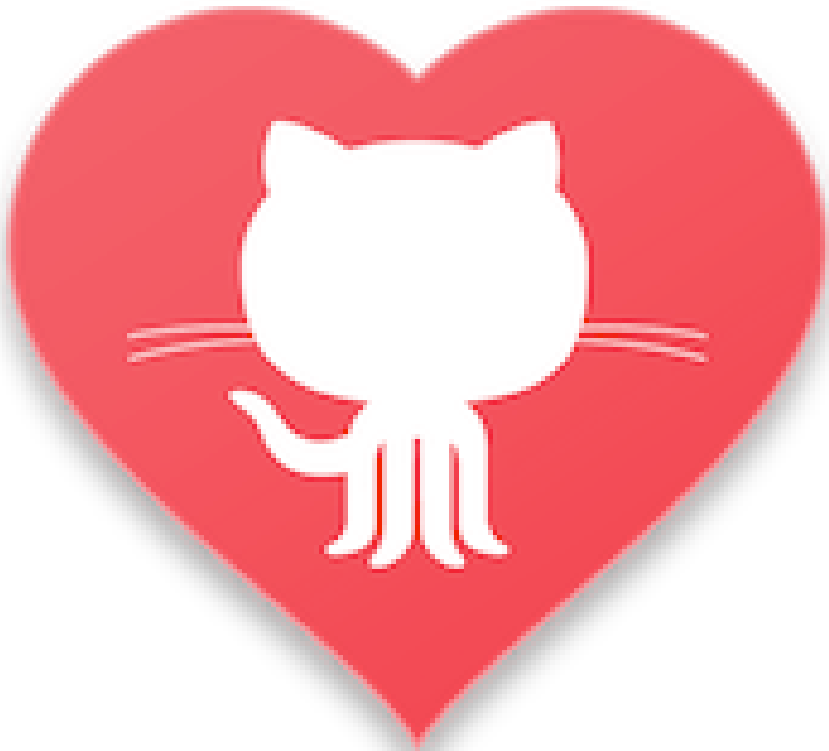
Customers



# Questions ?



# Please Contribute !!



<https://github.com/chef>



#Thank You

**thanks for  
listening!**



Like it Share it!!

