Course Content:

- Introduced to Configuration Management in Automation.
- Chef Server
    - Chef Introduction and Flavors
    - Chef Architecture
    - Server Configuration
- Chef Workstation
    - Workstation Introduction and Configuration
    - Chef Domain Specific Language (DSL)
    - Chef Recipe
    - Chef Cookbook
    - Knife
- Chef Node
- Chef Other topic

Chef Introduction and Types

- Chef is a powerful tool for automation that transforms infrastructure into code. Whether you're operating in the cloud or on-premises or in Hybrid Environment. Chef automates how infrastructure is configured, deployed and managed across your network
- Chef is a Ruby based powerful tool for automation that transforms infrastructure into code.
- Chef is available in two Types:
- Chef is available in two flavors:
    - **HOSTED CHEF**

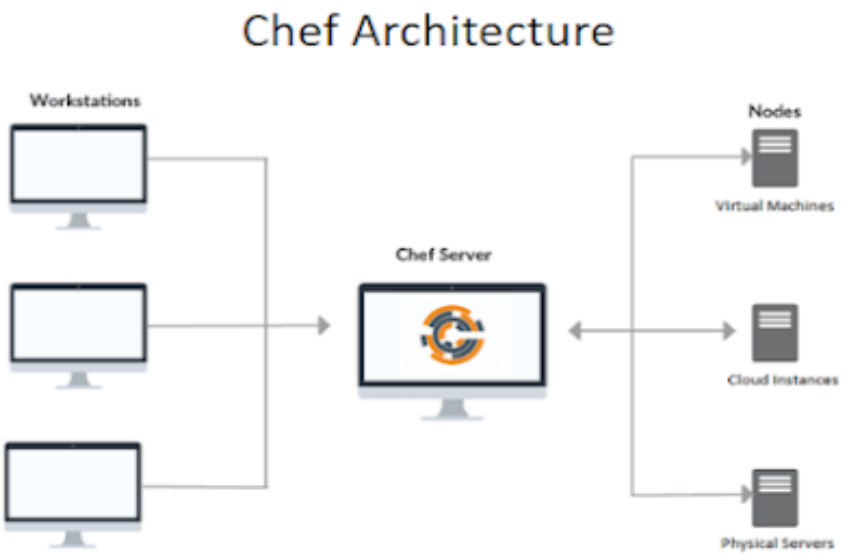
- **ONE ERMISES CHEF**

HOSTED CHEF

- Hosted Enterprise Chef is the quickest and easiest way to get started with chef. This chef is hosted by Chef itself.
- just signup to https://manage.chef.io/signup and create or organization.
- Its Highly available and highly scalable.

ON PERMISES CHEF

- With On Premises Chef, a Chef server to be run on-premises is provisioned by the customer.
- The main advantage over Hosted Chef, of course, is that full control over the server is maintained. Faster rollout and better integration is also possible since the server is likely to be physically closer to the rest of the customer's network.
- On Premises Chef servers reside behind the customer's own firewalls, the machines are shielded from any public global issues that may affect Hosted Chef customers.



Server configuration(ON PERMISES CHEF)

- set up and manage your own Chef server
- Go to https://downloads.chef.io/chef-server and select your OS flavor
- Pre-Requisites:
    - Persistent FQDN
    - Hostname
    - System Requirements
        - 4vCPU
        - 4-8GB RAM
    - Support Platform
        - RHEL | CentOS | Ubuntu |SUSE
        - No Windows Support
    - Firewall Settings
        - Port 80 | 443
    - Time Sync

- Install the Chef server package.
- Run “sudo chef-server-ctl install chef-manage” to install management console
- Run “sudo chef-server-ctl reconfigure and “sudo chef-manage-ctl reconfigure"

Setup Chef User and Organization

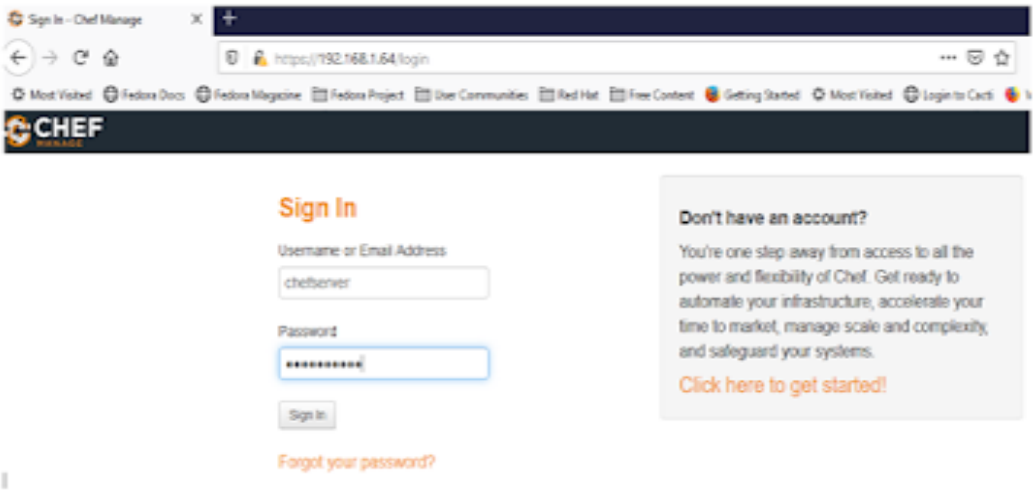
- Chef server connects workstation and client nodes. To link them, will create an admin and organizer with their private keys.
- Firstly, create a directory to store the keys.
    - Eg. **mkdir .chef**
- **Run the following command to create an administrator:**
    - **chef-server-ctl user-create user_name first_name last_name email ‘password’ --filename FILE_NAM**

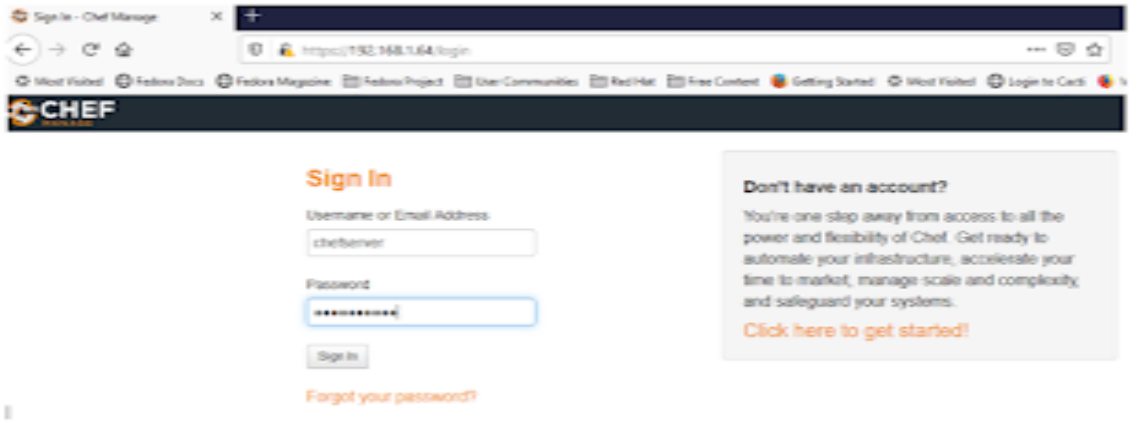
- eg. **chef-server-ctl user-create ramesh ramesh awasthi email@ramesh.com ‘Strongpassword’ --filename ~/.chef/ramesh.pem**

- Run the following command to create an Organization:
    - **sudo chef-server-ctl org-create ORG_NAME “ORG_FULL_NAME” –association_user USER_NAME –filename FILE_NAME**
    - eg. **Sudo chef-server-ctl org-create ramesh-org “ramesh chef infrastructure” –association-user ramesh --filename ~/.chef/ramesh-org.pem**

- Run the following command to check the list of user and organization on the chef server
    - **sudo chef-server-ctl user-list**
    - **sudo chef-server-ctl org-list**

- You should be able to access the Chef web admin dashboard on https://your-chef-serverip/login





Chef Workstation Introduction and Configuration

· Chef workstation is our local desktop or laptop with Chef development kit software installed on it.

· The Chef workstation where we create and configure any recipes, cookbook, attributes and other changes necessary to manage our nodes.

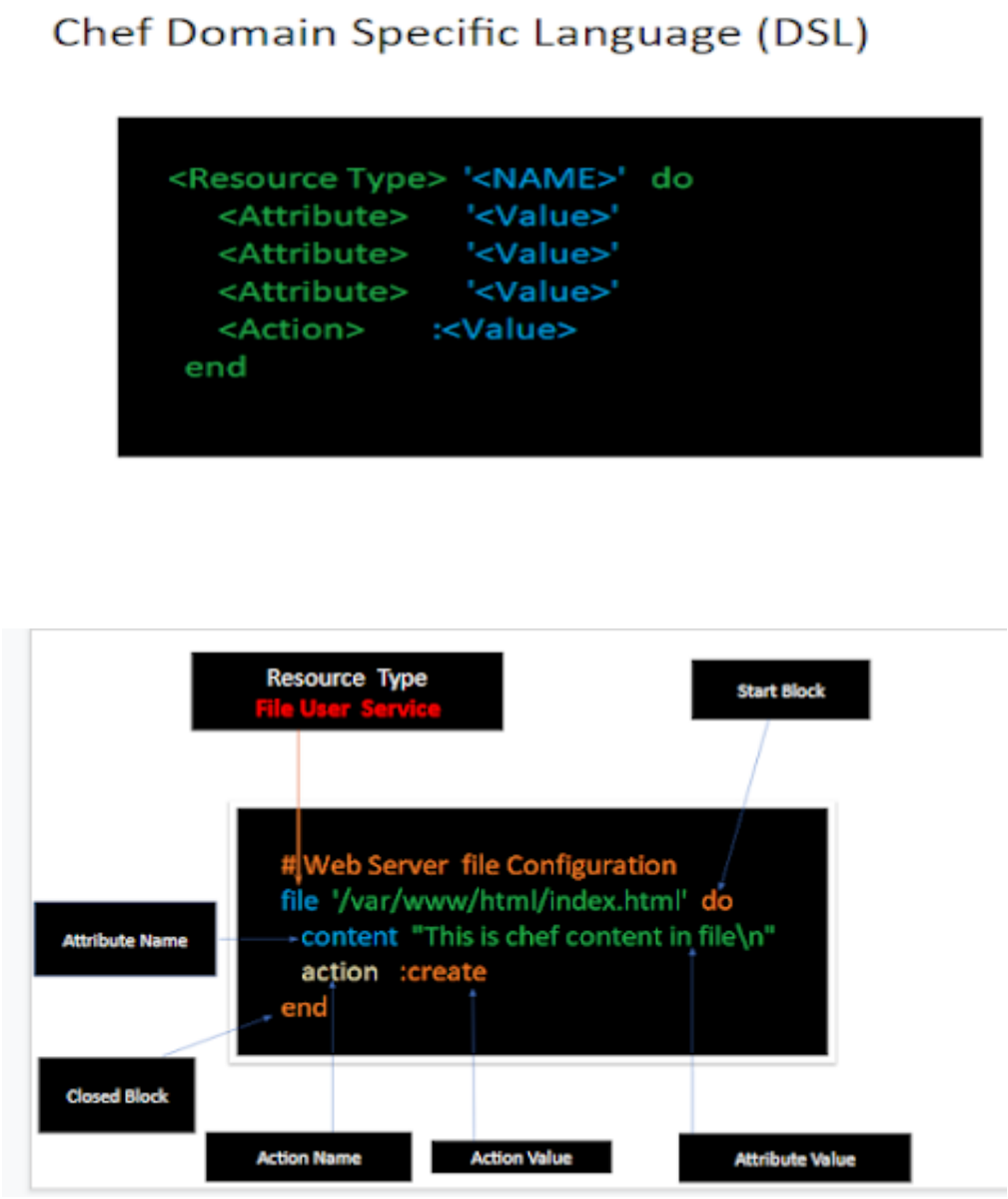
· There is no particular system requirements for Chef Workstation.

· Chef workstation can be run on a variety of OS platforms which includes Windows and Mac OS.

· Download the latest Chef Workstation:
https://downloads.chef.io/chef-workstation/

· Install the Chef server package

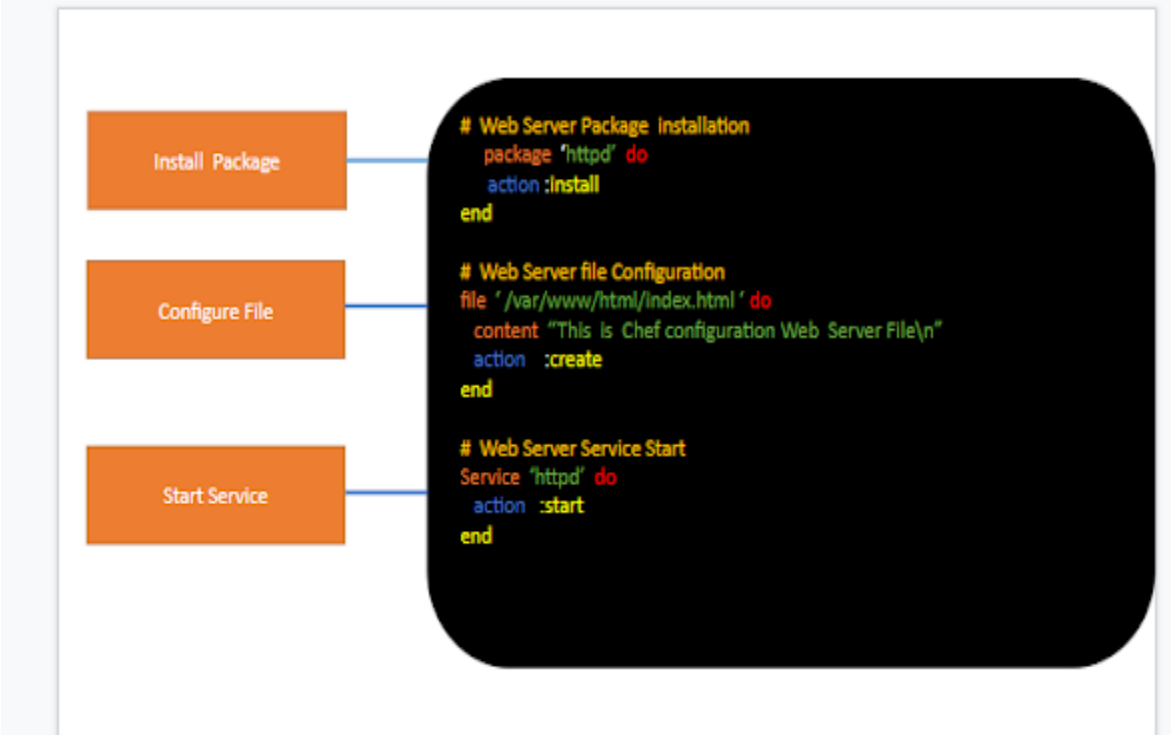
Chef Domain Specific Language (DSL)



Recipes

- Configuration files that describe resources and their desired state. Creating file with .rb extension.

- Recipes can:
    - install and configure software components
    - Manage files
    - Deploy applications
    - Execute other recipes
- four step process to create, check, test and run Chef Recipes

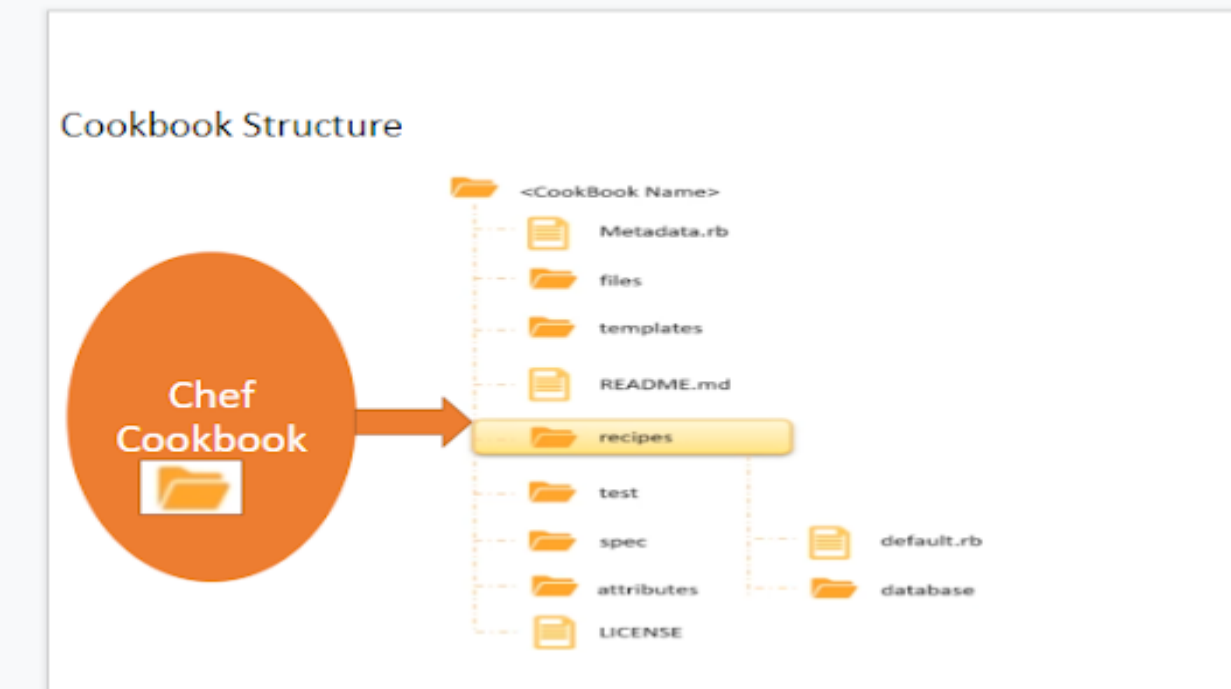


Cookbook

· Cookbooks are a set of recipes grouped together aiming to solve a similar problem.

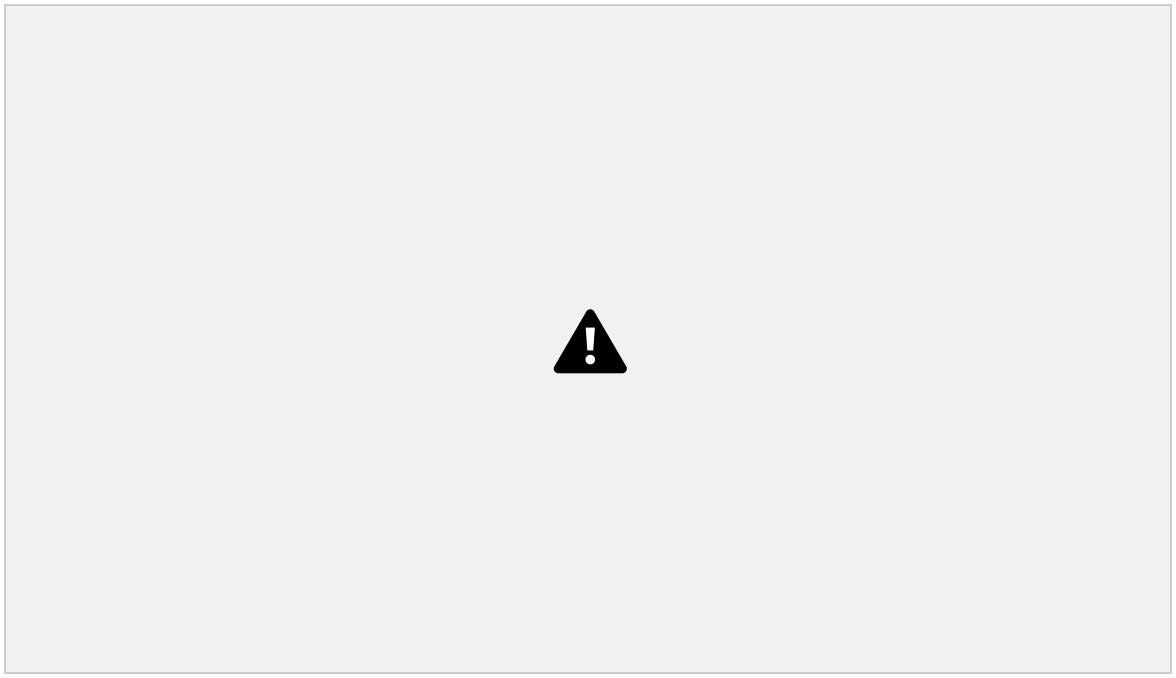
· All cookbooks follow similar standards such as they all have a name, a section with the Table of Contents, set of chapters, set of pages, a summary, contact, information, license, etc.

· The default path for the Cookbooks directory can be found under the .chef directory located in the chef-repo path, which we extracted during workstations setup using starter kit.



Create a Chef Cookbook

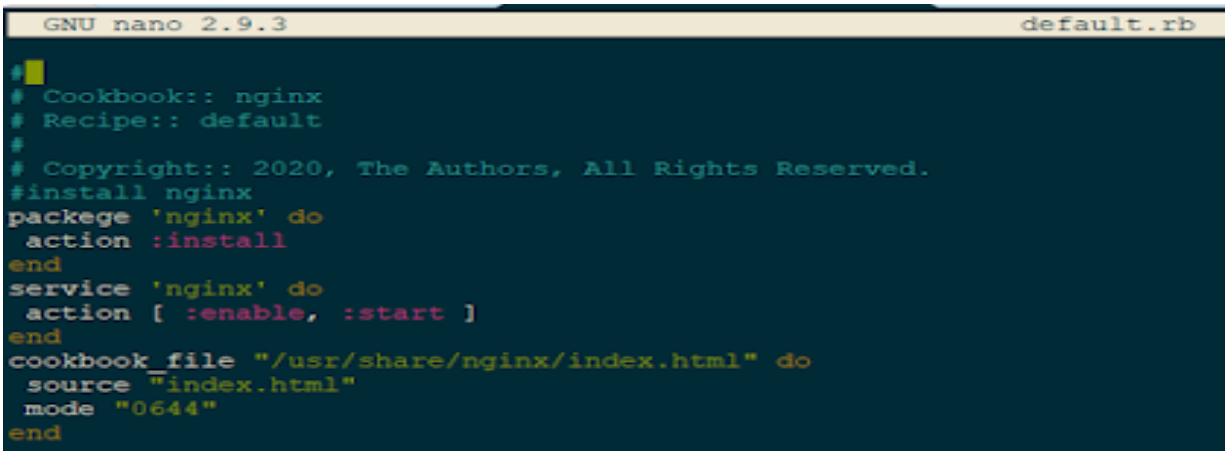
- Go to chef-repo directory.
- chef generate cookbook cookbook_name



```
ubuntu@workstation:~/chef-repo/cookbooks$ ls
chef-repo  example  nginx  README.md
ubuntu@workstation:~/chef-repo/cookbooks$ cd nginx/
ubuntu@workstation:~/chef-repo/cookbooks/nginx$ ls
Berksfile  CHANGELOG.md  chefignore  LICENSE  metadata.rb  README.md  recipes  spec  test
```

```
ubuntu@workstation:~/chef-repo/cookbooks/nginx$ cd recipes/
ubuntu@workstation:~/chef-repo/cookbooks/nginx/recipes$ ls
default.rb
```

- **nano default.rb**



Nodes

- Nodes may represent physical servers, virtual servers or cloud instances.
- Chef Client installation can be done in two ways.
  - Install and configure Chef Client
    - https://downloads.chef.io/chef
  - Knife Bootstrap Method

Knife

Knife is a Command-line tool that provides an interface between a local chef- repo and chef server. knife helps users to manage:

- Nodes
- cookbooks and recipes Roles
- Cloud resources, including provisioning
- The installation of the chef-client on management workstation.

```
ubuntu@workstation:~$ sudo nano ~/chef-repo/.chef/knife.rb
```

```
 GNU nano 2.9.3                        /home/ubuntu/chef-repo/.chef/knife.rb

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

```
ubuntu@workstation:~/chef-repo$ knife client list
chef-org-validator
```

Bootstrap a Node

- Bootstrapping a node installs the Chef client on the node and validates the node. This allows the node to read from the Chef server and pull down and apply any needed configuration updates detected by the chef-client.

```
ubuntu@workstation:~/chef-repo$ knife bootstrap 192.168.1.67 -N ubuntu-node1 --ssh-user ubuntu --su
do --identity-file /home/ubuntu/chef-repo/.chef/chefserver.pem
```

```
ubuntu@workstation:~/chef-repo$ knife bootstrap 192.168.1.67 -N ubuntu-node1 --ssh-user ubuntu --su
do --identity-file /home/ubuntu/chef-repo/.chef/chefserver.pem
Creating new client for ubuntu-node1
Creating new node for ubuntu-node1
Connecting to 192.168.1.67
ubuntu@192.168.1.67's password:
192.168.1.67 knife sudo password:
Enter your password:
192.168.1.67
192.168.1.67 -----> Existing Chef installation detected
192.168.1.67 Starting the first Chef Client run...
192.168.1.67 Starting Chef Infra Client, version 16.1.0
192.168.1.67 resolving cookbooks for run list: []
192.168.1.67 Synchronizing Cookbooks:
192.168.1.67 Installing Cookbook Gems:
192.168.1.67 Compiling Cookbooks...
192.168.1.67 [2020-05-21T13:09:17+05:45] WARN: Node ubuntu-node1 has an empty run list.
192.168.1.67 Converging 0 resources
192.168.1.67
192.168.1.67 Running handlers:
192.168.1.67 Running handlers complete
192.168.1.67 Chef Infra Client finished, 0/0 resources updated in 02 seconds
ubuntu@workstation:~/chef-repo$
```

```
ubuntu@workstation:~/chef-repo$ knife node list
ubuntu-node1
```

Chef Other Topics

- Ohai
- Kitchen Testing
- Roles
- Environments
- Supermarket

Ohai

· Ohai is run by the chef-client at the beginning of every Chef run to determine the current system state.

· This data consists of host, kernel and network information, IP address, CPU, memory, file system and disk details and much more.

· The returned output of Ohai is in a Json format.

Kitchen Testing

· Kitchen is a command line application that helps us manage the testing lifecycle.

Roles

· Roles represent the types of servers in our infrastructure

- Load Balancer

- Application Server
- Database
- Monitoring
- Networking Devices

· Roles may include a list of Chef configuration files that should be applied.

· We call this list a Run List

Environments

- Model the life-stages of your applications
- Every Organization starts with a single environment
- Environments to reflect your patterns and workflow
  - Development
  - Test
  - Staging
  - Production
  - Etc.
- Environments may include data attributes necessary for configuring your infrastructure
  - The location of your package repository
  - The version of the Chef configuration files that that should be used

Supermarket

- Chef supermarket is the official website for users to search, download, install and use Chef Cookbooks.
- It is a repository of over 37 hundred pre-written cookbooks.
  You can explore more about supermarket at https://supermarket.chef.io/