DevOps
Chef for Configuration Management

Trainer: Abhijith V G – AWS, eCommerce, Mobile & DevOps Architect

Magento Certified

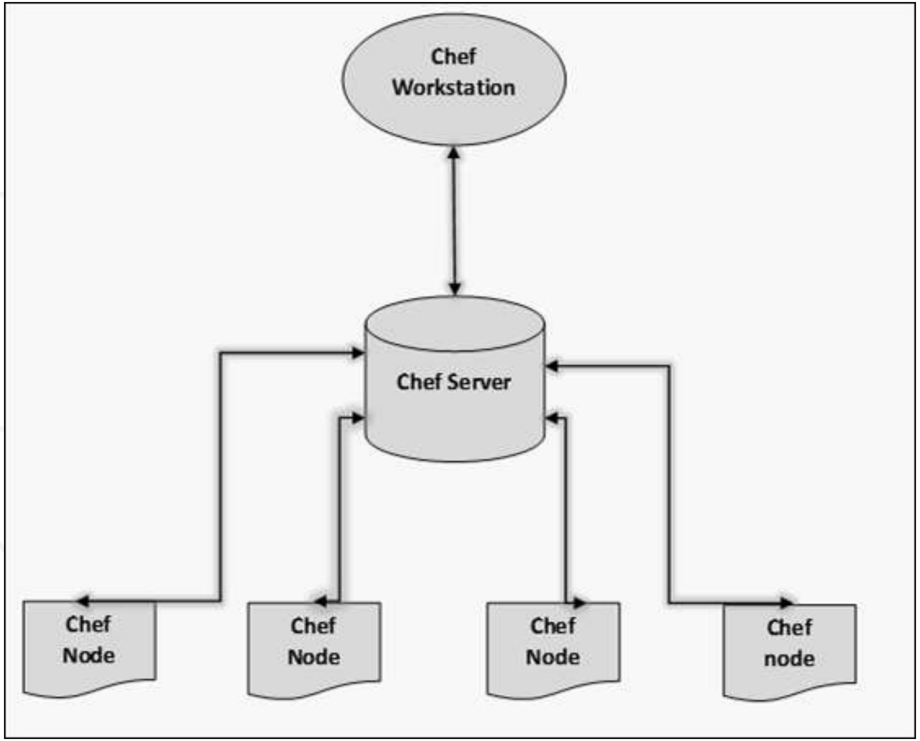
DEVELOPER PLUS

amazon web services | Certified

Developer - Associate

Chef: Overview of Chef

Chef is a powerful automation platform that transforms infrastructure into code.



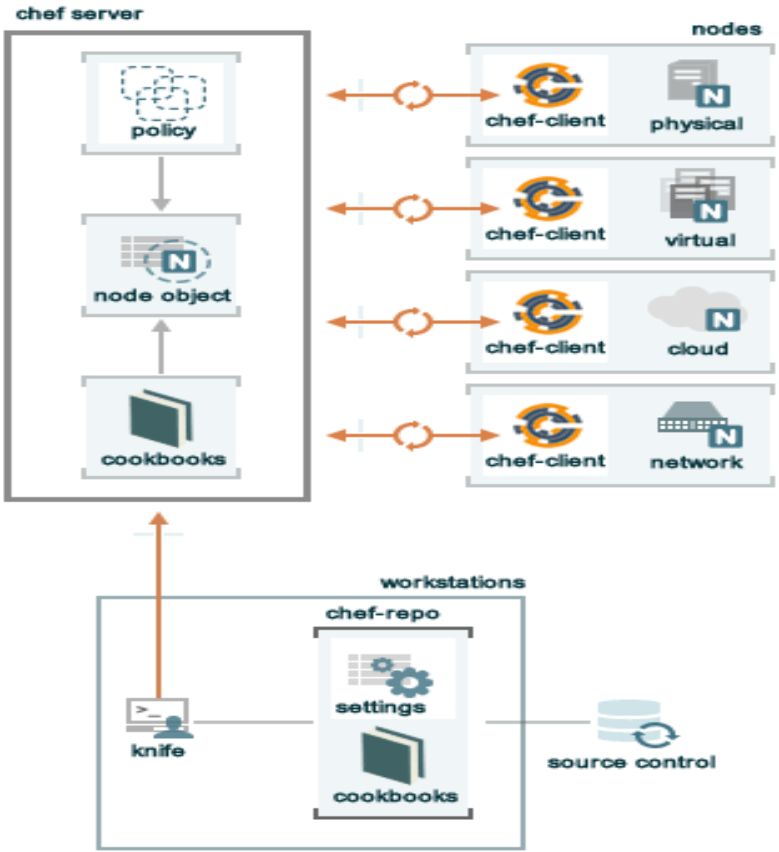
Pull
Chef Recipe

ChefDK

Chef: Overview of Chef: Workstation, Server and Nodes

1. **Workstation:** The workstation is the location from which users (DevOps personnel/technical people) interact with Chef
2. **Nodes (Android, Windows, Linux etc):** Nodes are the machines—physical, virtual, cloud, and so on—that are under management by Chef. The chef-client is installed on each node. Nodes use the chef-client to ask the Chef server for configuration details, such as recipes, templates, and file distributions
3. **Chef Server:** The Chef server acts as a hub for configuration data. 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.

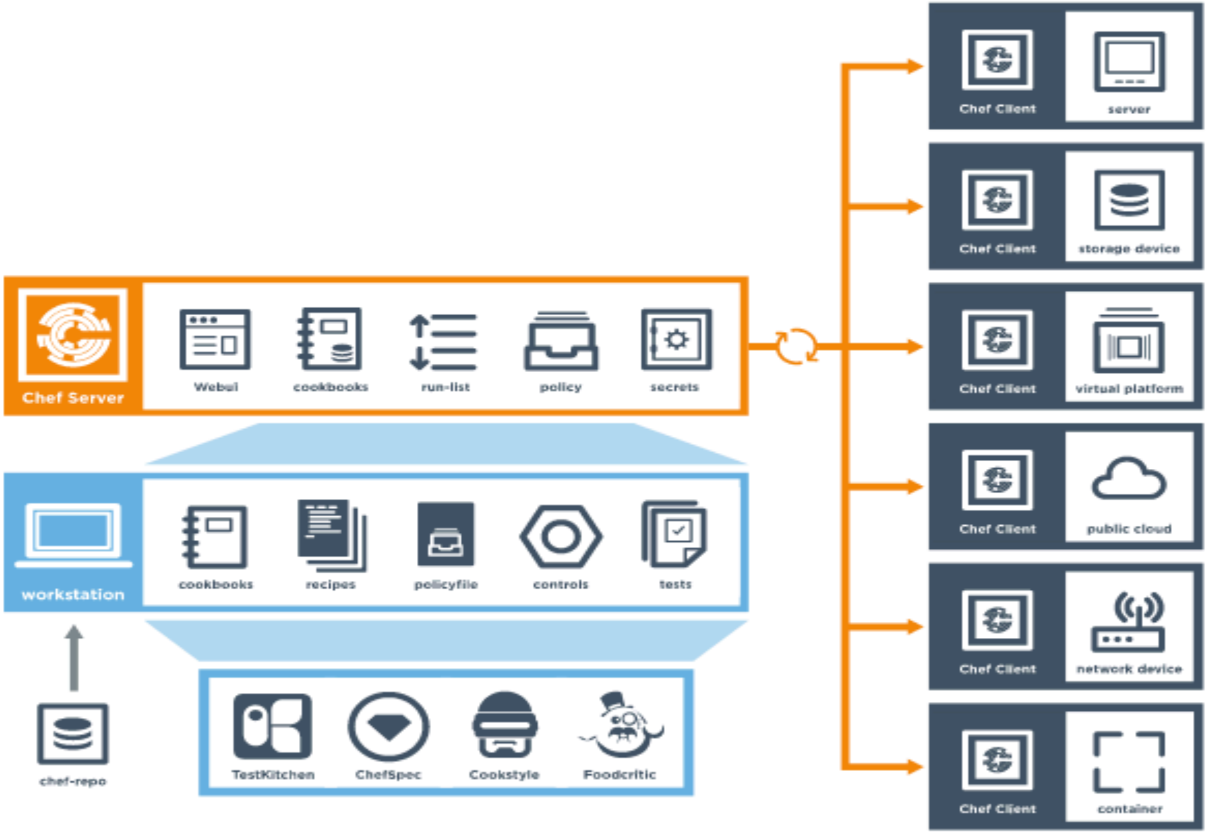
Chef: Overview of Chef: Workstation, Server and Nodes



Chef: Overview of Chef: Terminologies: Nodes

Nodes

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



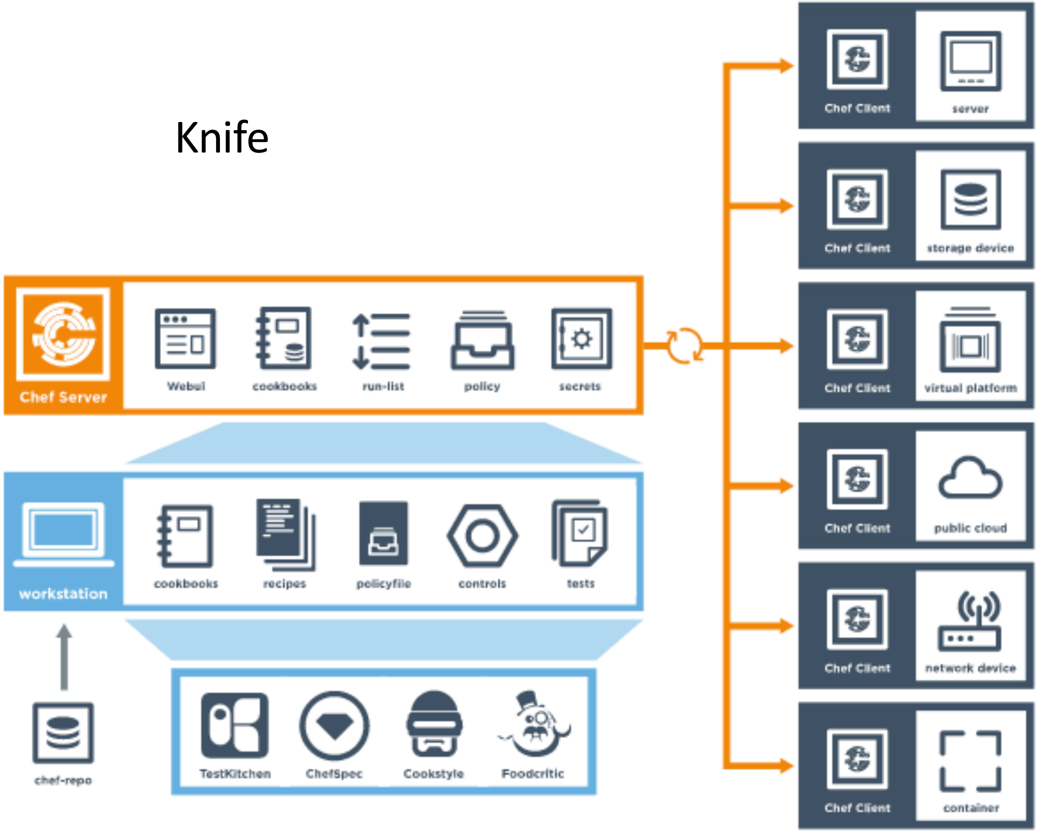
Chef: Overview of Chef: Terminologies: Workstations

Workstations

A workstation is a computer running the Chef Development Kit (ChefDK) that is used to author cookbooks, interact with the Chef server, and interact with nodes.

The workstation is where developing and testing cookbooks and recipes, testing Chef code, keeping the chef-repo synchronized with version source control, configuring organizational policy.. etc will happen.

Knife

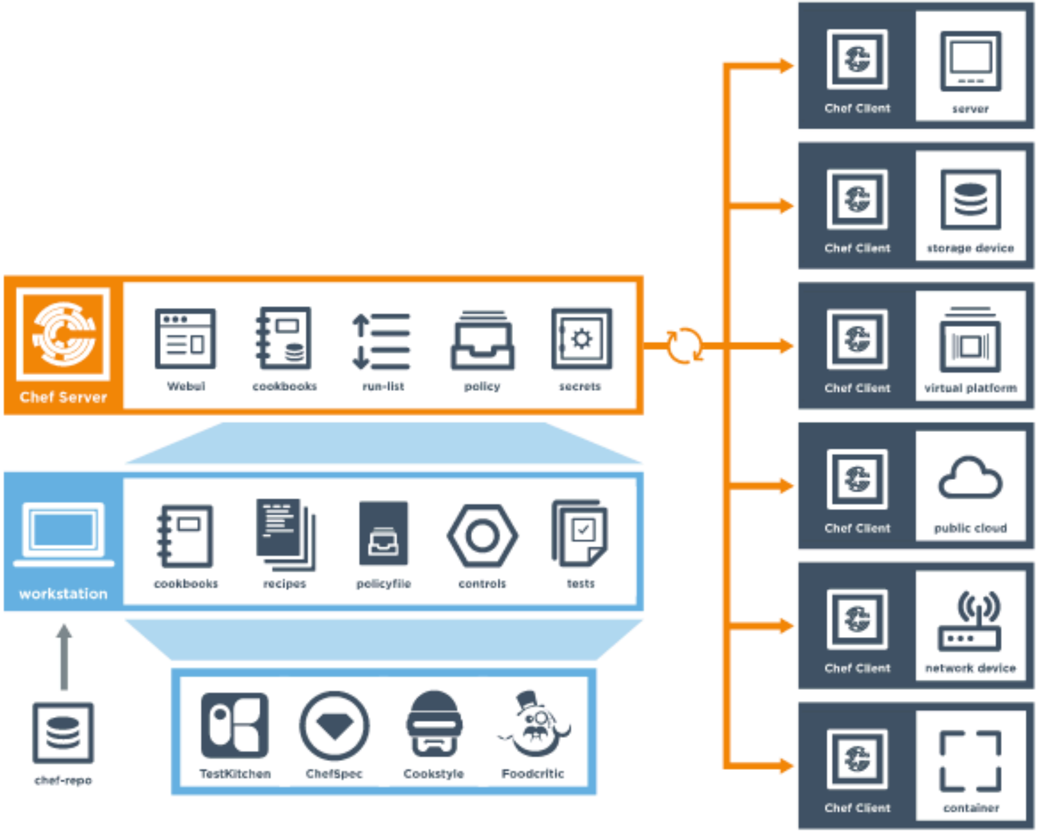


Chef: Overview of Chef: Terminologies: Knife

Knife

knife is a 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, Environments, and Data Bags
- Resources within various cloud environments
- The installation of the chef-client onto nodes
- Searching of indexed data on the Chef server



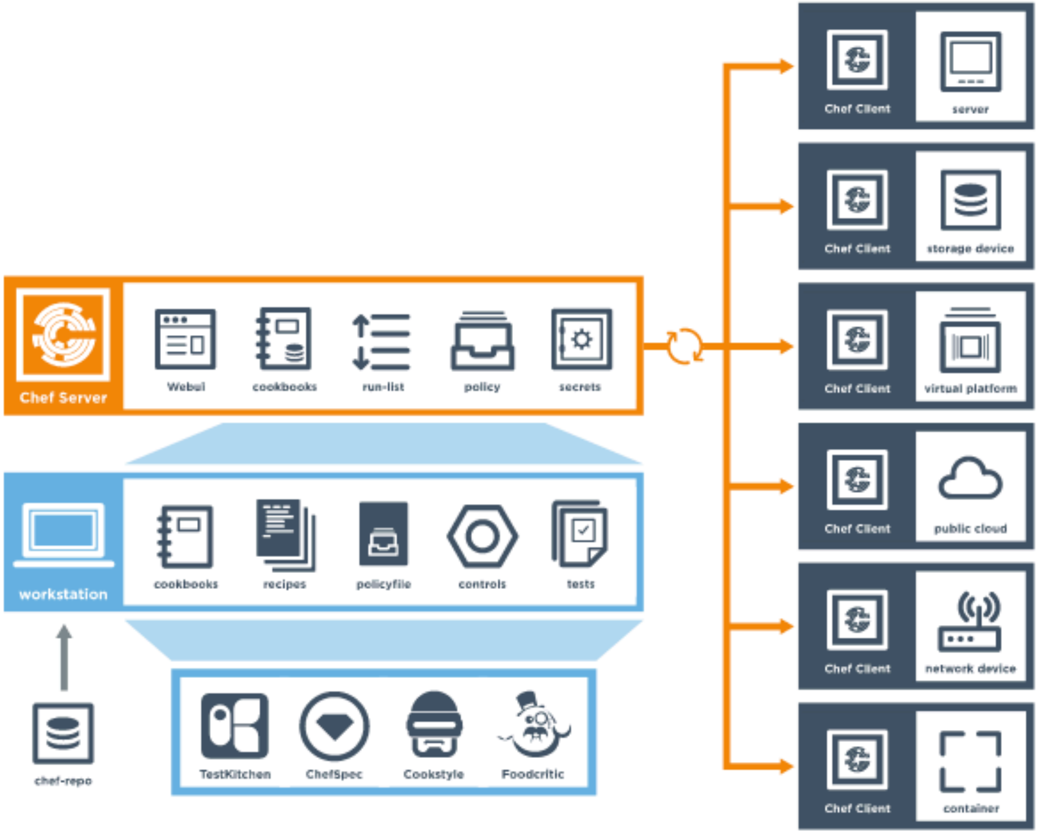
Chef: Overview of Chef: Terminologies: Repository

Repository (chef-repo)

The chef-repo is a directory on your workstation that stores:

- Cookbooks (authored, tested, and maintained)
- Roles
- Data bags
- Environment

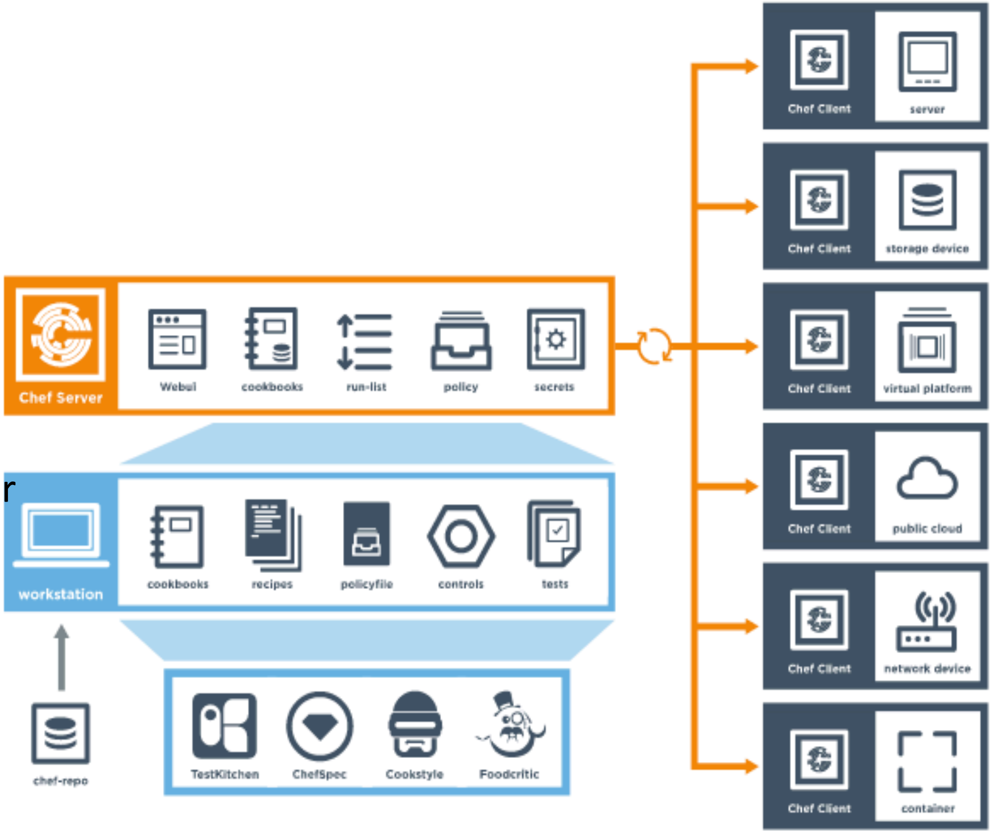
The chef-repo should be synchronized with a version control system (such as git), and then managed as if it were source code. ‘git’ is the most commonly-used for chef-repo.



Chef: Overview of Chef: Terminologies: Chef Server

Chef Server

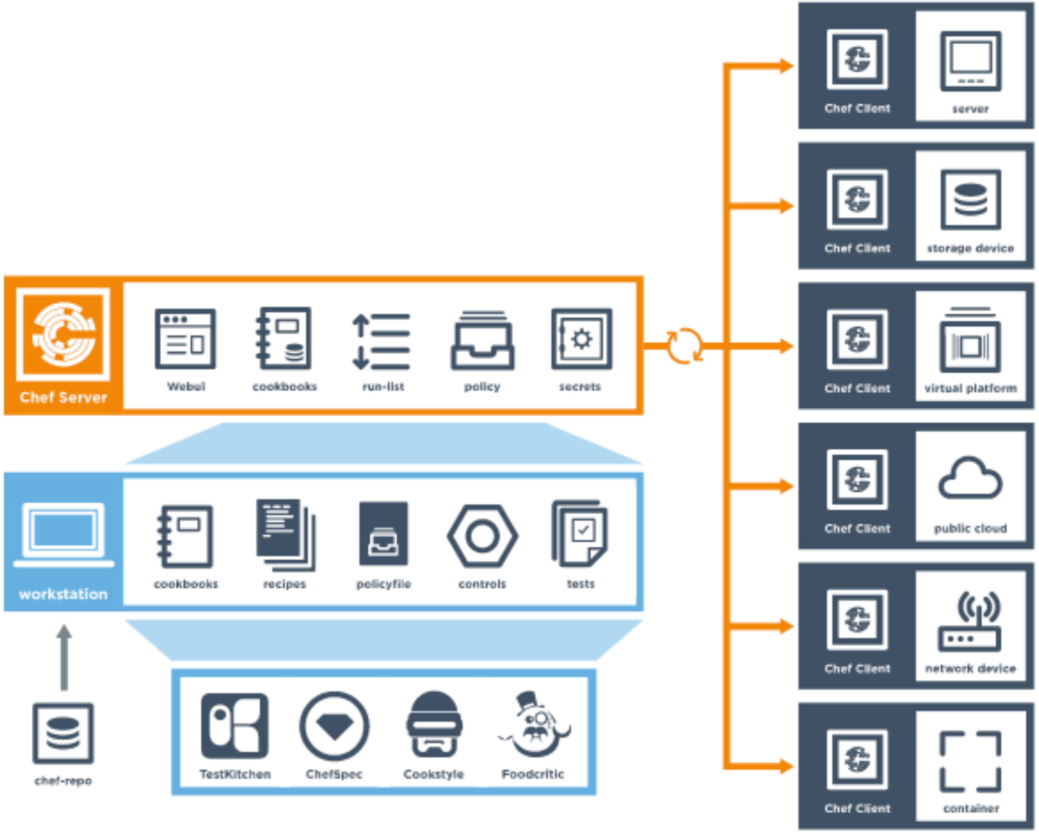
- The Chef server acts as a hub for configuration data.
- The Chef server stores cookbooks, the policies that are applied to nodes, and metadata that describes each registered node that is being managed by the chef-client.
- Nodes use the chef-client to ask the Chef server for configuration details, such as recipes, templates, and file distributions.
- The chef-client then does as much of the configuration work as possible on the nodes themselves (and not on the Chef server).
- This scalable approach distributes the configuration effort throughout the organization.



Chef: Overview of Chef: Terminologies: The Hosted Chef Server

The Hosted Chef Server

- The hosted Chef server is a version of the Chef server that is hosted by Chef. The hosted Chef server is cloud-based, scalable, and available (24x7/365), with resource-based access control. The hosted Chef server has the same automation capabilities of any Chef server, but without requiring it to be set up and managed from behind the firewall.



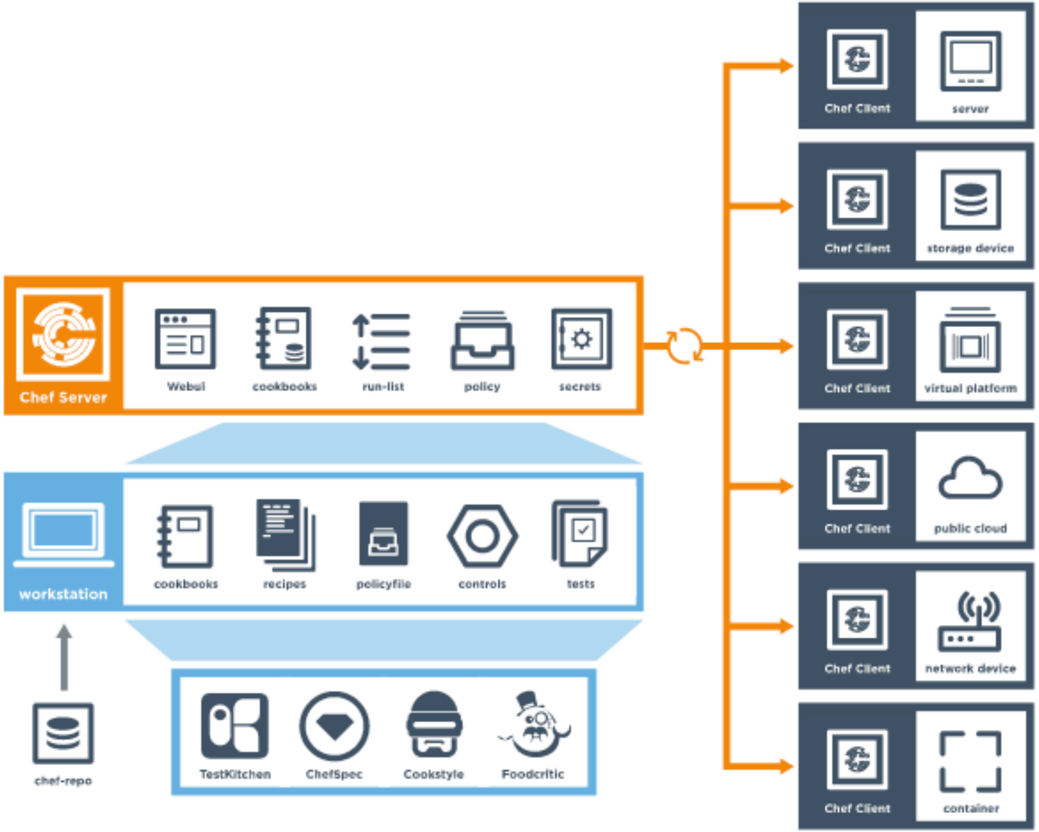
IDEMPOTENT

Chef: Overview of Chef: Terminologies: Cookbooks

Cookbooks

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

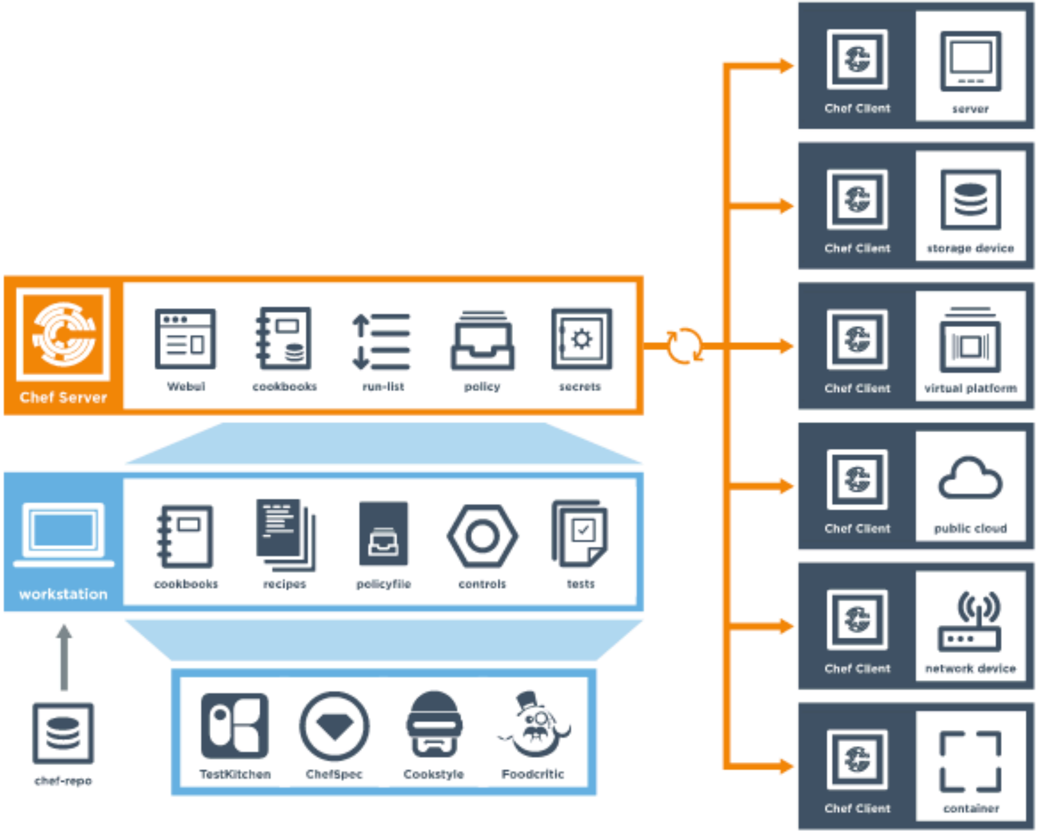
Recipes that specify the resources to use and the order in which they are to be applied Attribute values, File distributions, Templates. The chef-client uses Ruby as its reference language for creating cookbooks and defining recipes. It also has a DSL that can be extended.



Chef: Overview of Chef: Terminologies: Chef Supermarket

Chef Supermarket

Chef Supermarket is the site for community cookbooks. It provides an easily searchable cookbook repository and a friendly web UI. Cookbooks that are part of the Chef Supermarket are accessible by any Chef user. ... The public Chef Supermarket is hosted by Chef and is located at https://supermarket.chef.io .



Chef: Overview of Chef: Chef Configuration Concepts: Policy

Policies describe how you want Chef to build your system

Policy: A collection of system configurations that you define (roles/databags/environments). The policy states the state that each node should be in but not how to get there. Chef-client will pull the policy and configure the node with the appropriate resources so that it matches the state of the policy.

Policy concept examples:

- If it should be installed
- If it is not installed then install it
- If it is already installed then do nothing
- A file should exist - if not, create it
- If a file exists but does not have correct content, correct it

Chef: Overview of Chef: Chef Configuration Concepts:

Resource represents a piece of a system or a **single** configuration item (and its desired state) present on a node that is under management by Chef.

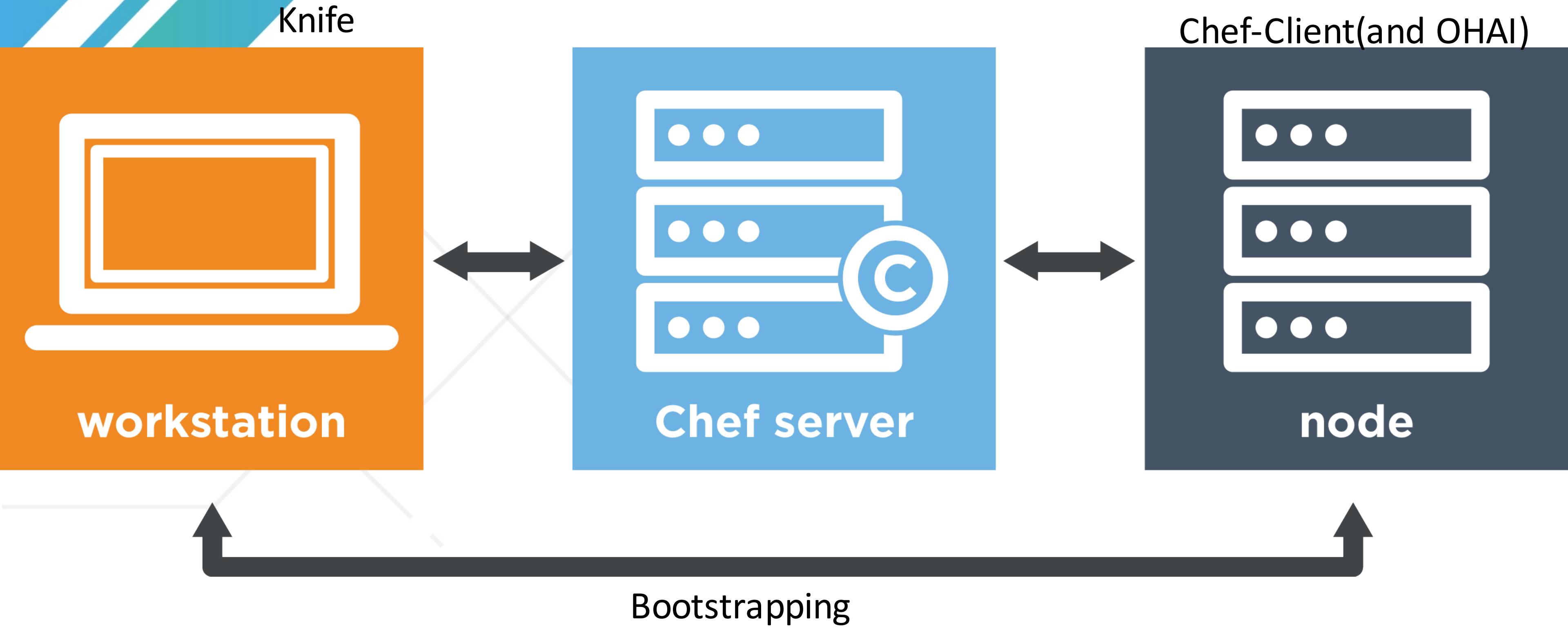
Building blocks of Chef configuration

When chef-client is run on a node, the resource is executed by the provider which is handled by Chef and the OS itself. What provider to use (ie what package manager to use, such as yum or apt-get) is populated when **Ohai** is run at the start of each chef-client run.

Most common resource types are:

- **Package**: Used to manage packages such as installing packages
- **Template**: Used to manage the contents of a Ruby template in the cookbook
- **Service**: Manage system services - what run levels to start the service in, current state of the service (running/stopped/etc)

Chef: Chef WS----Chef Server----Nodes



Chef: Organization Setup: Create organization

www.getchef.com

https://getchef.opscode.com/signup

[OR]

https://api.chef.io/signup

[OR]

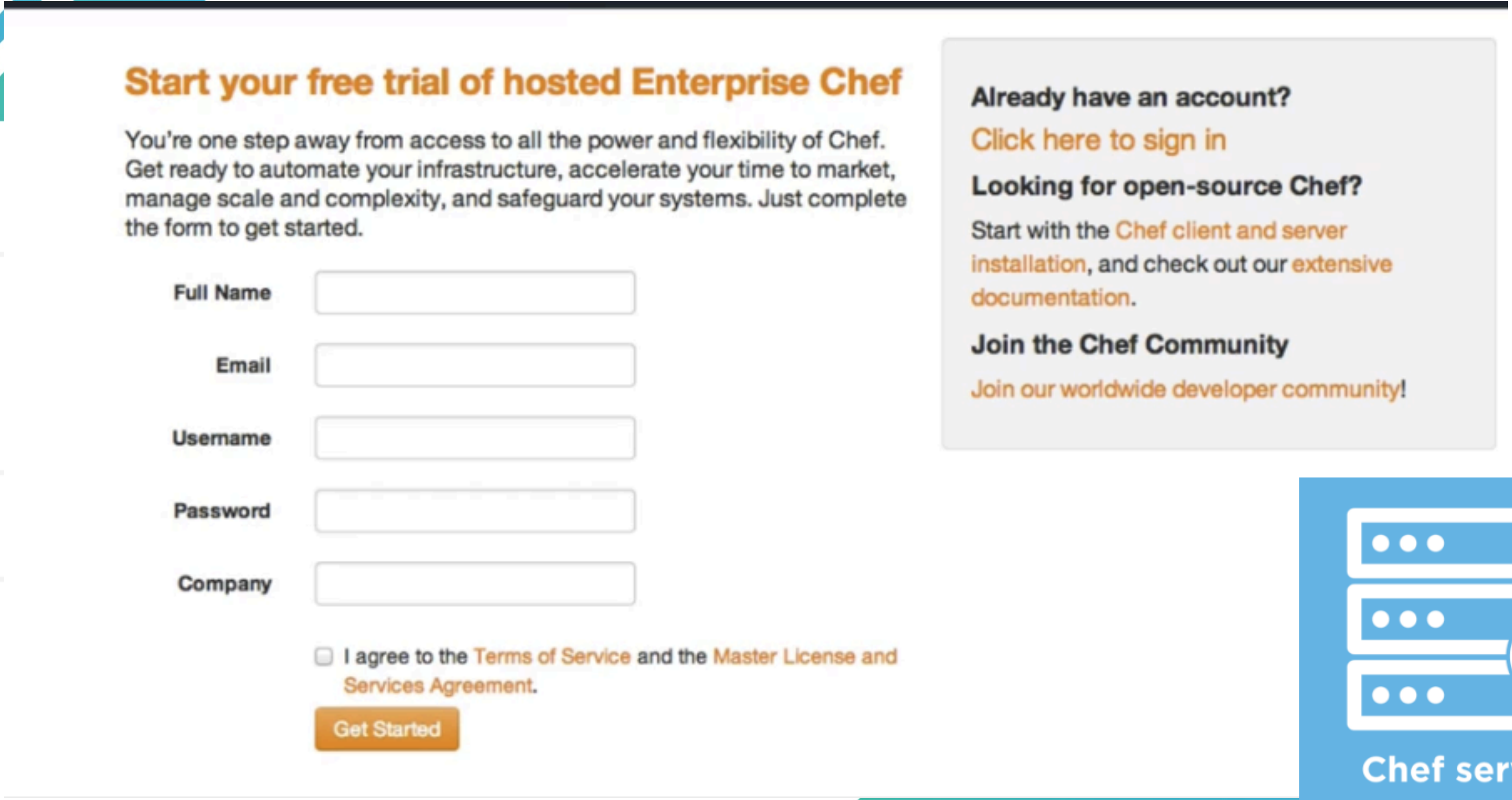
https://manage.chef.io/signup

[OR]

https://www.chef.io/

Chef server

Chef: Organization Setup: Create organization



Start your free trial of hosted Enterprise Chef

You're one step away from access to all the power and flexibility of Chef. Get ready to automate your infrastructure, accelerate your time to market, manage scale and complexity, and safeguard your systems. Just complete the form to get started.

Full Name	
Email	
Username	
Password	
Company	

□ I agree to the Terms of Service and the Master License and Services Agreement.

Already have an account?

Click here to sign in

Looking for open-source Chef?

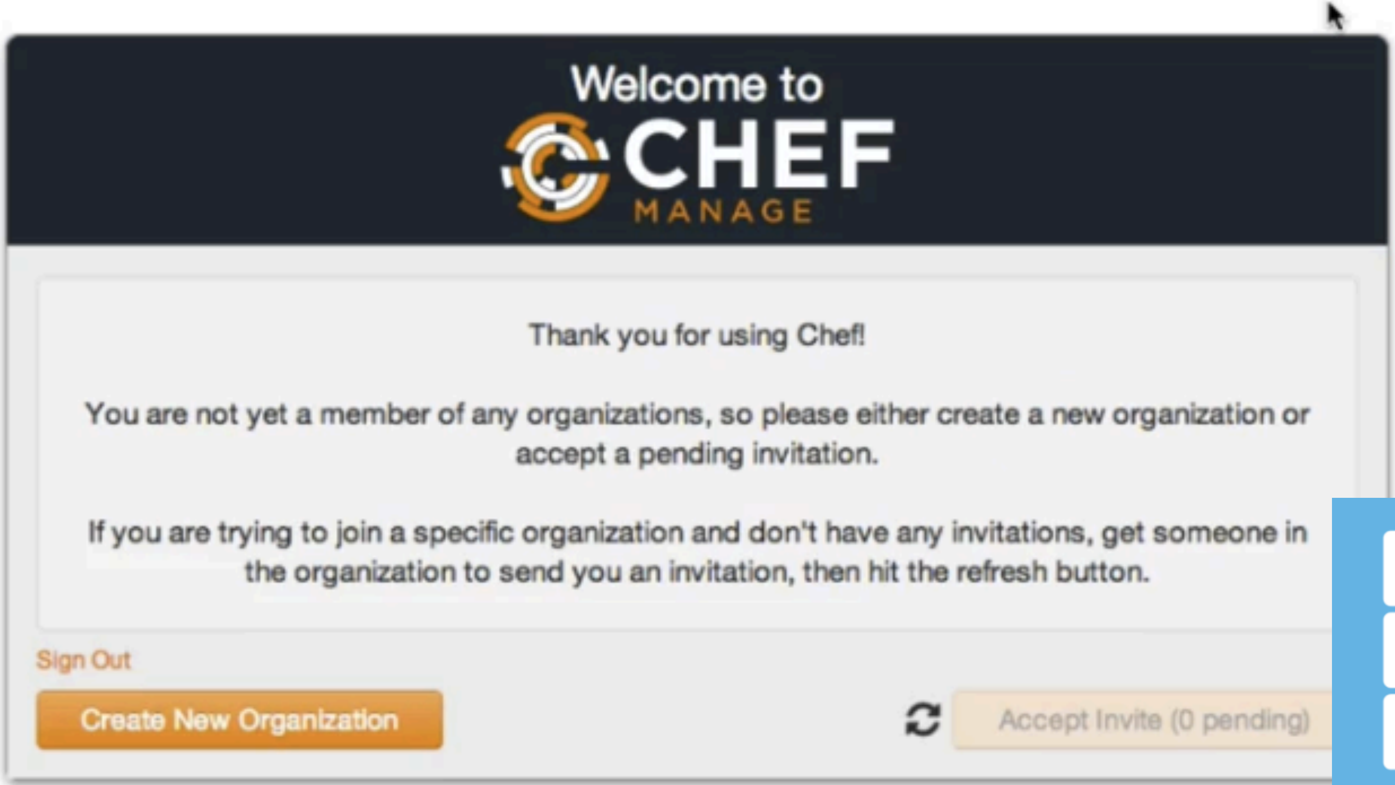
Start with the Chef client and server installation, and check out our extensive documentation.

Join the Chef Community

Join our worldwide developer community!

Chef server

Chef: Organization Setup: Create organization TODO-11



Welcome to CHEF MANAGE

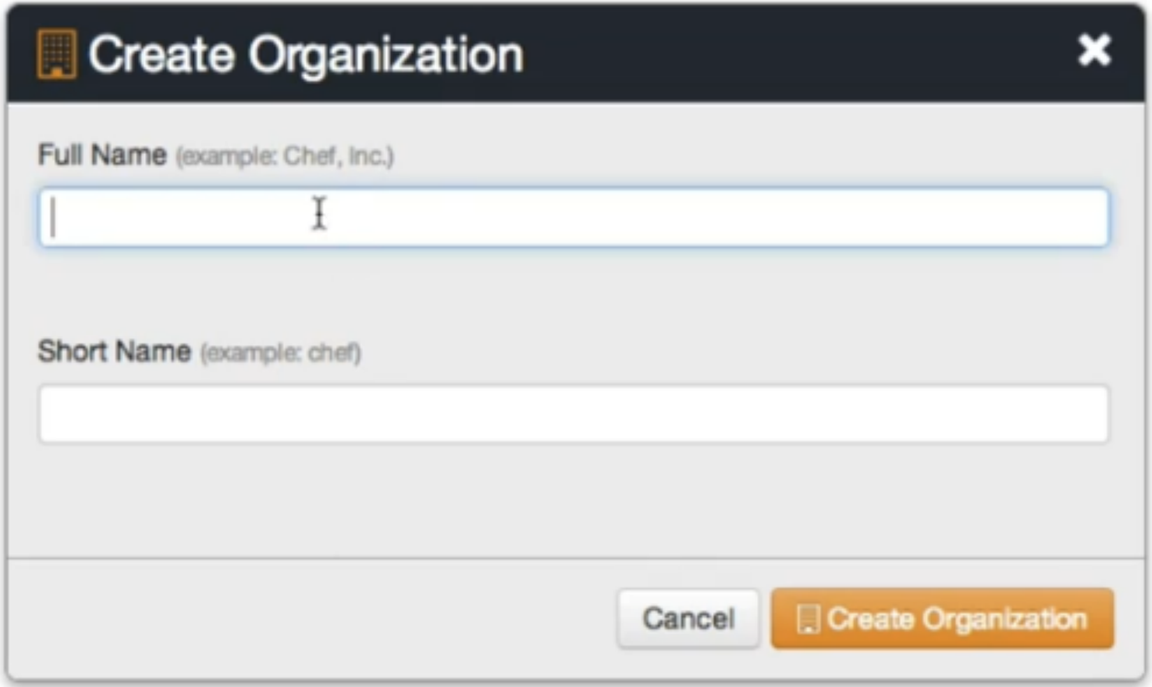
Thank you for using Chef!

You are not yet a member of any organizations, so please either create a new organization or accept a pending invitation.

If you are trying to join a specific organization and don't have any invitations, get someone in the organization to send you an invitation, then hit the refresh button.

Chef server

Chef: Organization Setup: Create organization TODO-11

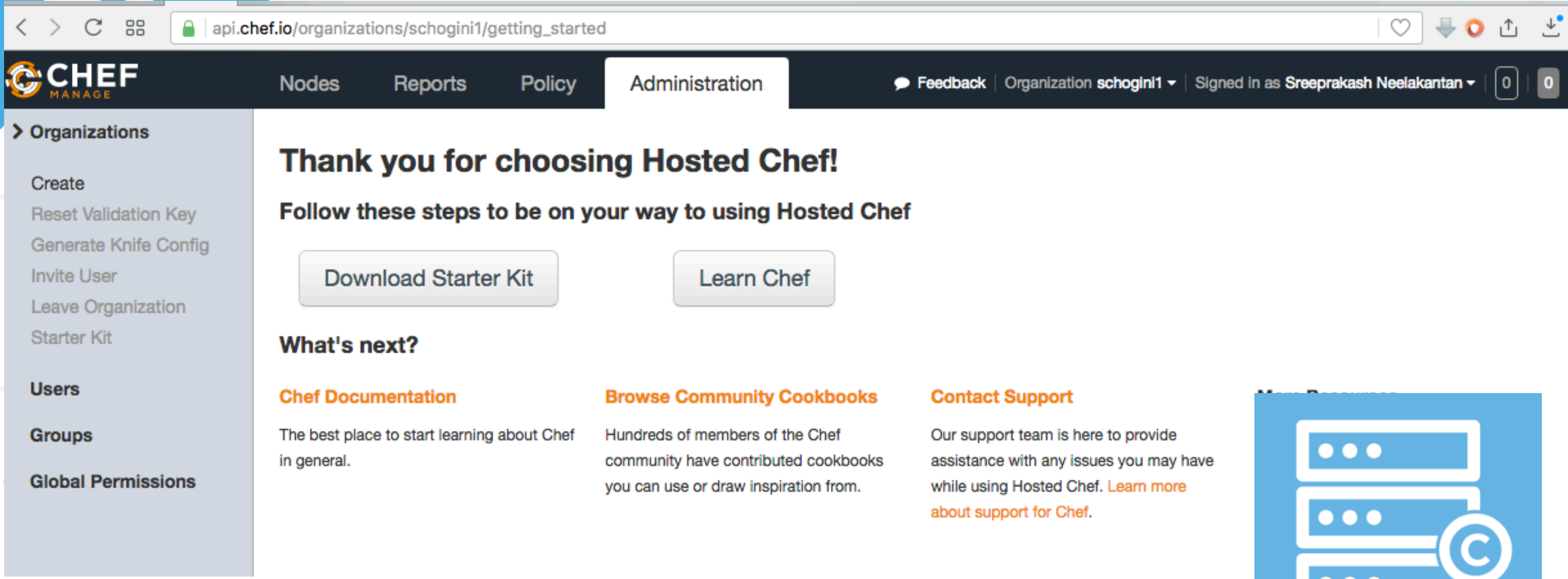


Full Name (example: Chef, Inc.)

Short Name (example: chef)

Chef server

Chef: Organization Setup: Create organization



api.chef.io/organizations/schogini1/getting_started

Thank you for choosing Hosted Chef!

Follow these steps to be on your way to using Hosted Chef

What's next?

Chef Documentation

The best place to start learning about Chef in general.

Browse Community Cookbooks

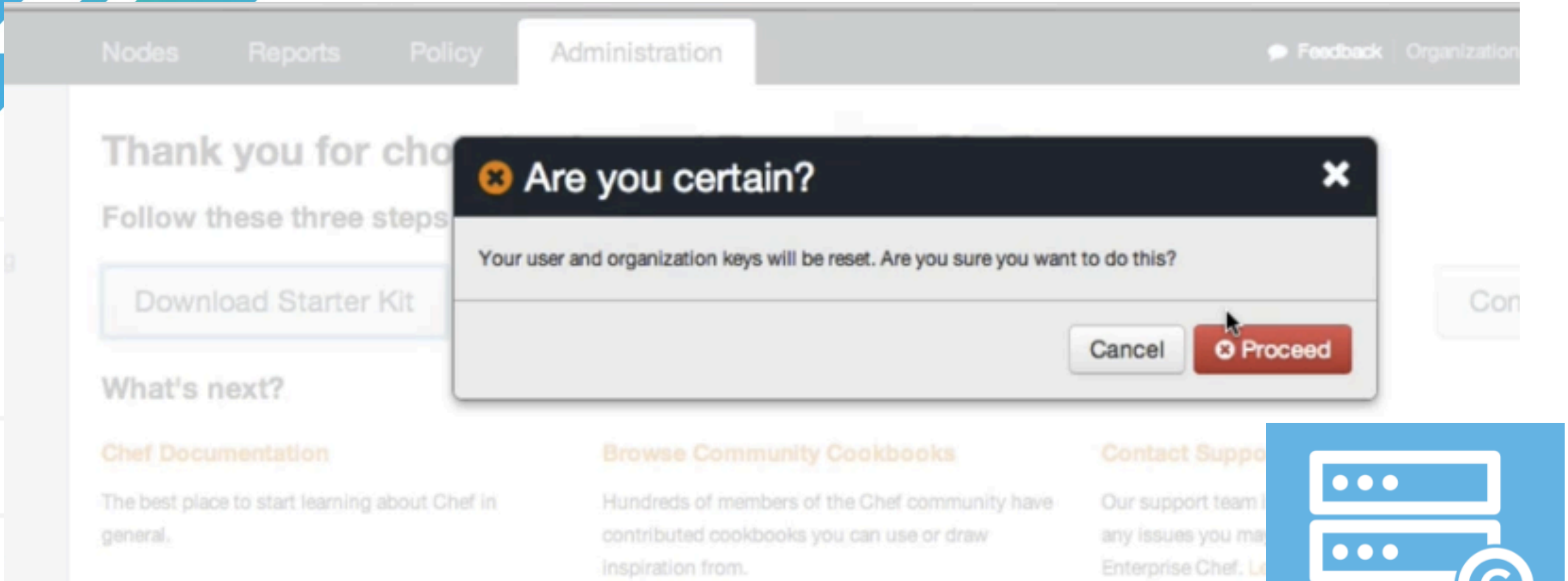
Hundreds of members of the Chef community have contributed cookbooks you can use or draw inspiration from.

Contact Support

Our support team is here to provide assistance with any issues you may have while using Hosted Chef. Learn more about support for Chef.

Chef server

Chef: Organization Setup: Create organization TODO-11



Thank you for cho

Follow these three steps

Download Starter Kit

Con

Are you certain?

Your user and organization keys will be reset. Are you sure you want to do this?

What's next?

Chef Documentation

The best place to start learning about Chef in general.

Browse Community Cookbooks

Hundreds of members of the Chef community have contributed cookbooks you can use or draw inspiration from.

Contact Suppo

Our support team i any issues you ma Enterprise Chef. L

This *chef-starter.zip* is what we used when we created our Chef Workstation container

Chef server

Chef: Organization Setup: Readying Chef Workstation

Download and setup the chef-starter.zip into the workstation as /root/chef-repo

Chef: Workstation Setup: Configure knife using Docker

STEP 1: Create the folders

```
# mkdir chef-ws
# cd chef-ws
```

STEP 2: Get the starter kit for your Chef account.

```
# unzip chef-starter.zip   🡐==  DOWNLOAD THIS FROM CHEF WEBSITE
```

STEP 3: Start the Chef Workstation

```
# docker network create chef
# docker run --rm -ti --name chefws --network chef -v $PWD/chef-repo:/root/chef-repo
    schogini/chef-ws
```

STEP 4: Enter the Workstation and setup your Git user

```
# git config --global user.email abhijithvg@example.com
# git config --global user.name ”Abhijith V G"
```

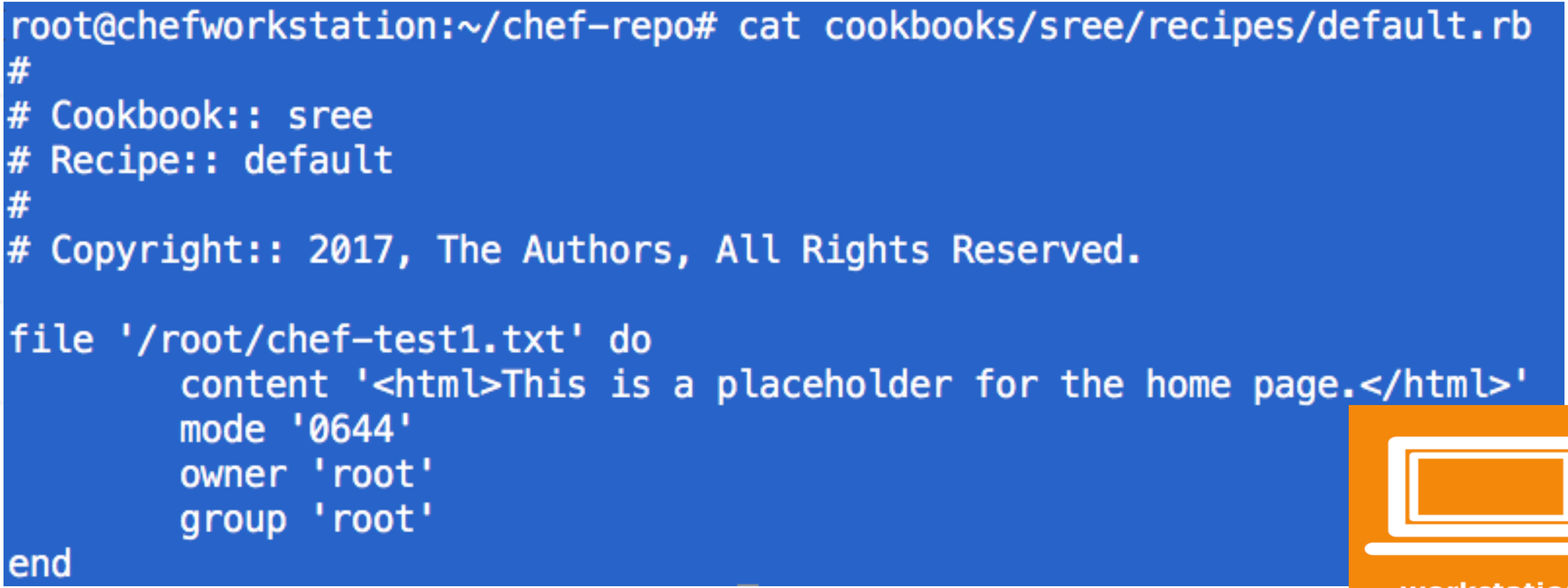
STEP 5: Run the Chef Client (in local mode)

```
# chef-client -z
```

workstation

Chef: Workstation Setup: Create a simple recipe

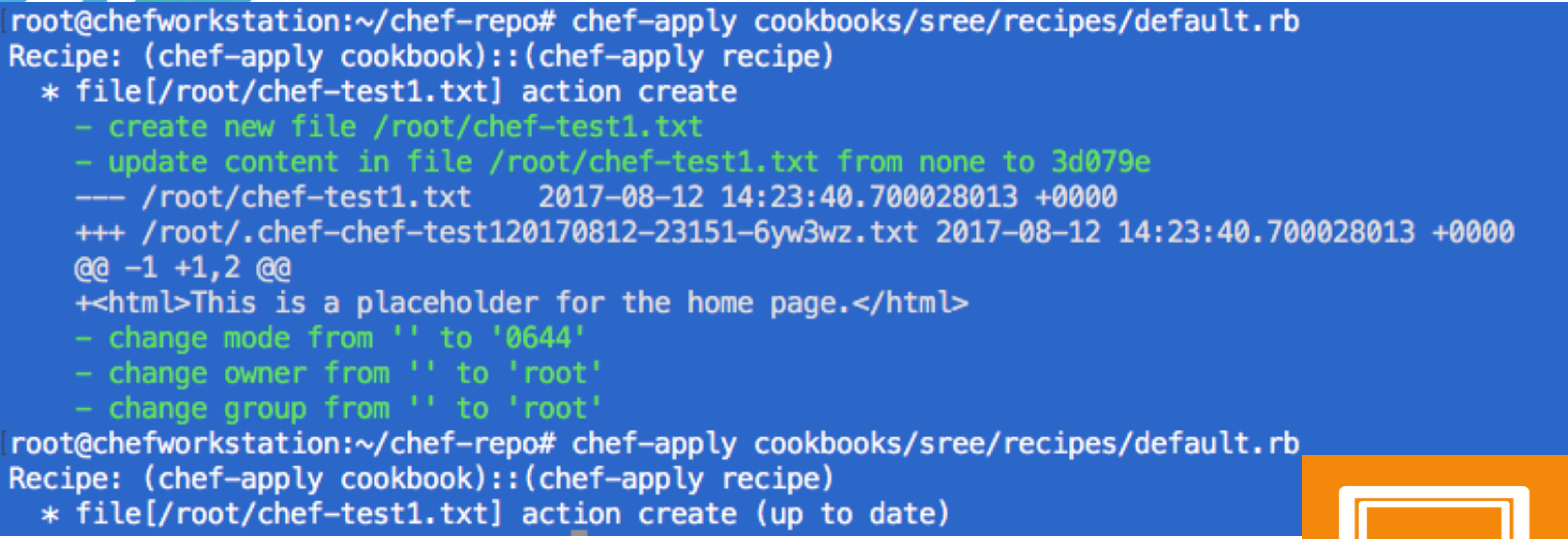
cd chef-repo/ cookbooks → *(while testing Chef assumes that all recipes are in the cookbooks folder)*
chef generate cookbook sree
nano cookbooks/sree/recipes/default.rb



```
root@chefworkstation:~/chef-repo# cat cookbooks/sree/recipes/default.rb
#
# Cookbook:: sree
# Recipe:: default
#
# Copyright:: 2017, The Authors, All Rights Reserved.

file '/root/chef-test1.txt' do
        content '<html>This is a placeholder for the home page.</html>'
        mode '0644'
        owner 'root'
        group 'root'
end
```

workstation

Chef: Workstation Setup: Applying Recipe-1: chef-apply (Locally)

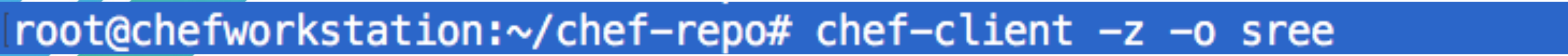


```
root@chefworkstation:~/chef-repo# chef-apply cookbooks/sree/recipes/default.rb
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * file[/root/chef-test1.txt] action create
    - create new file /root/chef-test1.txt
    - update content in file /root/chef-test1.txt from none to 3d079e
    --- /root/chef-test1.txt    2017-08-12 14:23:40.700028013 +0000
    +++ /root/.chef-chef-test120170812-23151-6yw3wz.txt 2017-08-12 14:23:40.700028013 +0000
    @@ -1 +1,2 @@
    +<html>This is a placeholder for the home page.</html>
    - change mode from '' to '0644'
    - change owner from '' to 'root'
    - change group from '' to 'root'
root@chefworkstation:~/chef-repo# chef-apply cookbooks/sree/recipes/default.rb
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * file[/root/chef-test1.txt] action create (up to date)
```

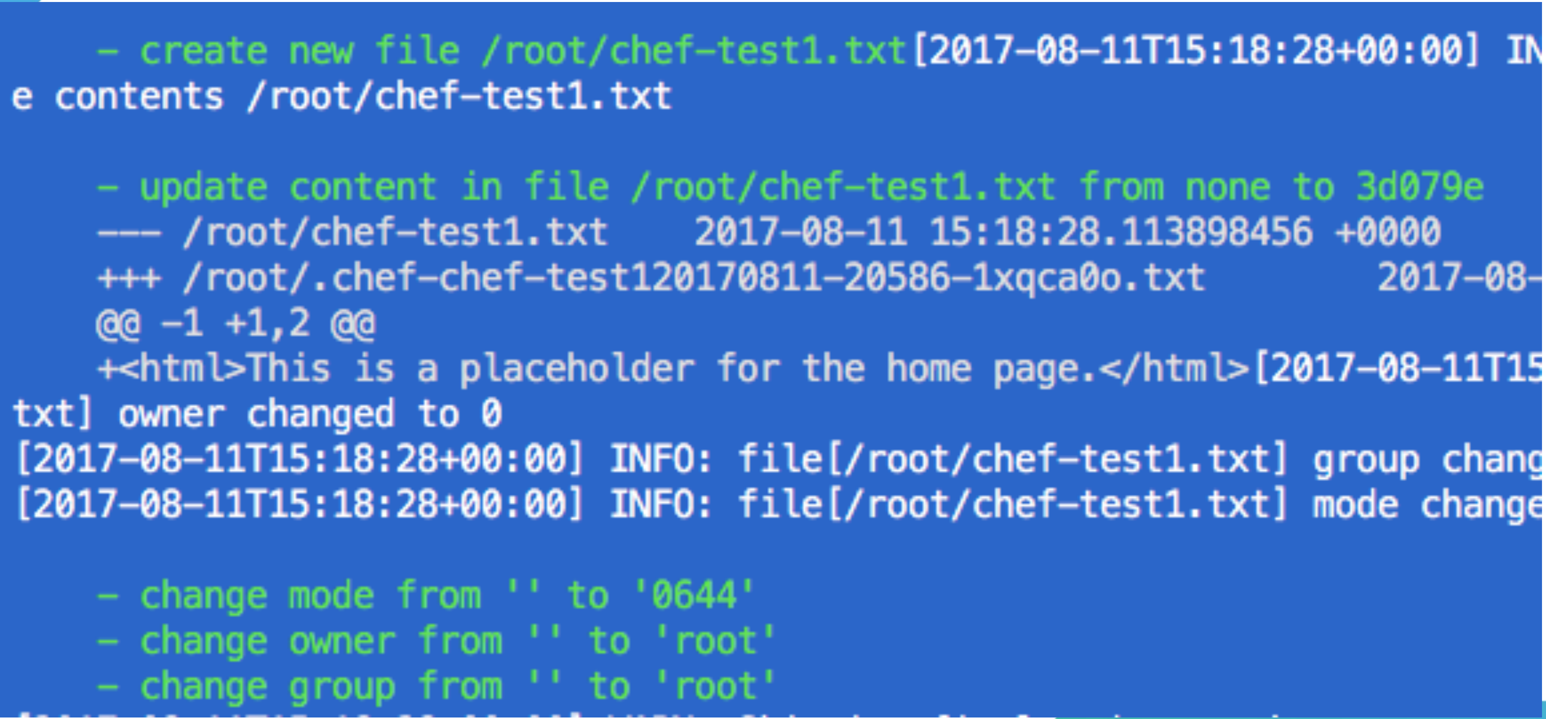
workstation

Chef: Workstation Setup: Applying Recipe-2: chef-client local mode

chef-client -z -o sree



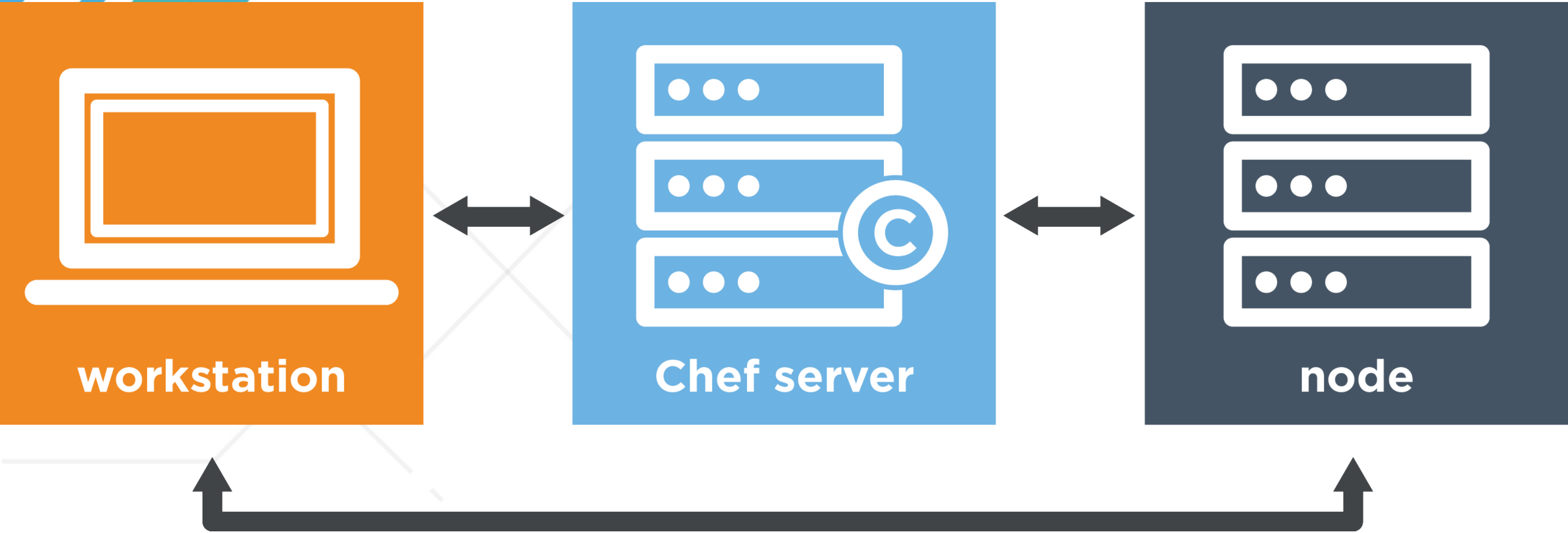
```
root@chefworkstation:~/chef-repo# chef-client -z -o sree
```



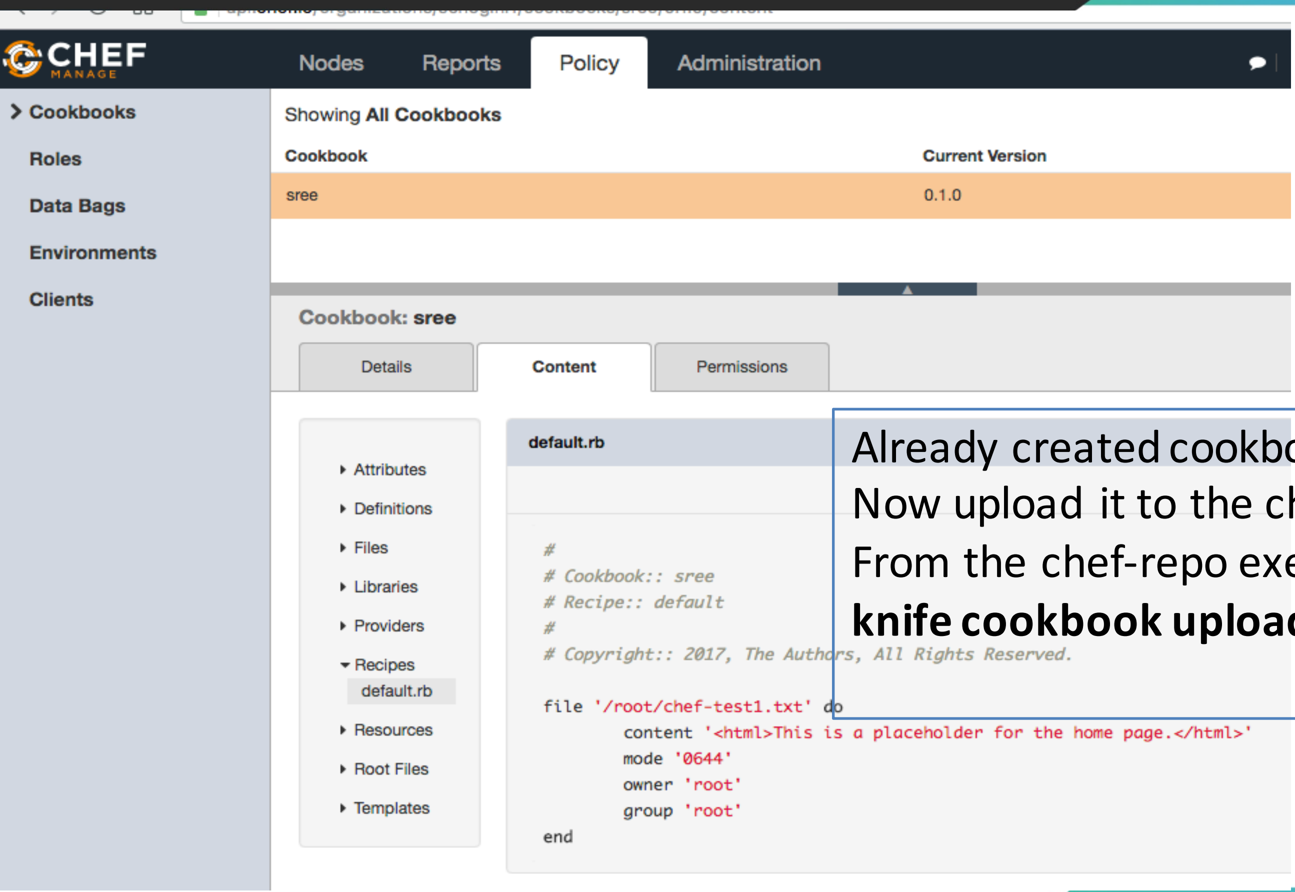
```
    - create new file /root/chef-test1.txt[2017-08-11T15:18:28+00:00] IN
e contents /root/chef-test1.txt

    - update content in file /root/chef-test1.txt from none to 3d079e
    --- /root/chef-test1.txt    2017-08-11 15:18:28.113898456 +0000
    +++ /root/.chef-chef-test120170811-20586-1xqca0o.txt       2017-08-
    @@ -1 +1,2 @@
    +<html>This is a placeholder for the home page.</html>[2017-08-11T15
txt] owner changed to 0
[2017-08-11T15:18:28+00:00] INFO: file[/root/chef-test1.txt] group chang
[2017-08-11T15:18:28+00:00] INFO: file[/root/chef-test1.txt] mode change

    - change mode from '' to '0644'
    - change owner from '' to 'root'
    - change group from '' to 'root'
```

workstation

Chef: Chef Server



Chef: Node Objects and Search: How to Add Run list to Node



Showing **All Cookbooks**

Cookbook	Current Version
sree	0.1.0

Cookbook: sree

default.rb

```
#
# Cookbook:: sree
# Recipe:: default
#
# Copyright:: 2017, The Authors, All Rights Reserved.

file '/root/chef-test1.txt' do
        content '<html>This is a placeholder for the home page.</html>'
        mode '0644'
        owner 'root'
        group 'root'
end
```

Already created cookbook sree with a default recipe

Now upload it to the chef server.

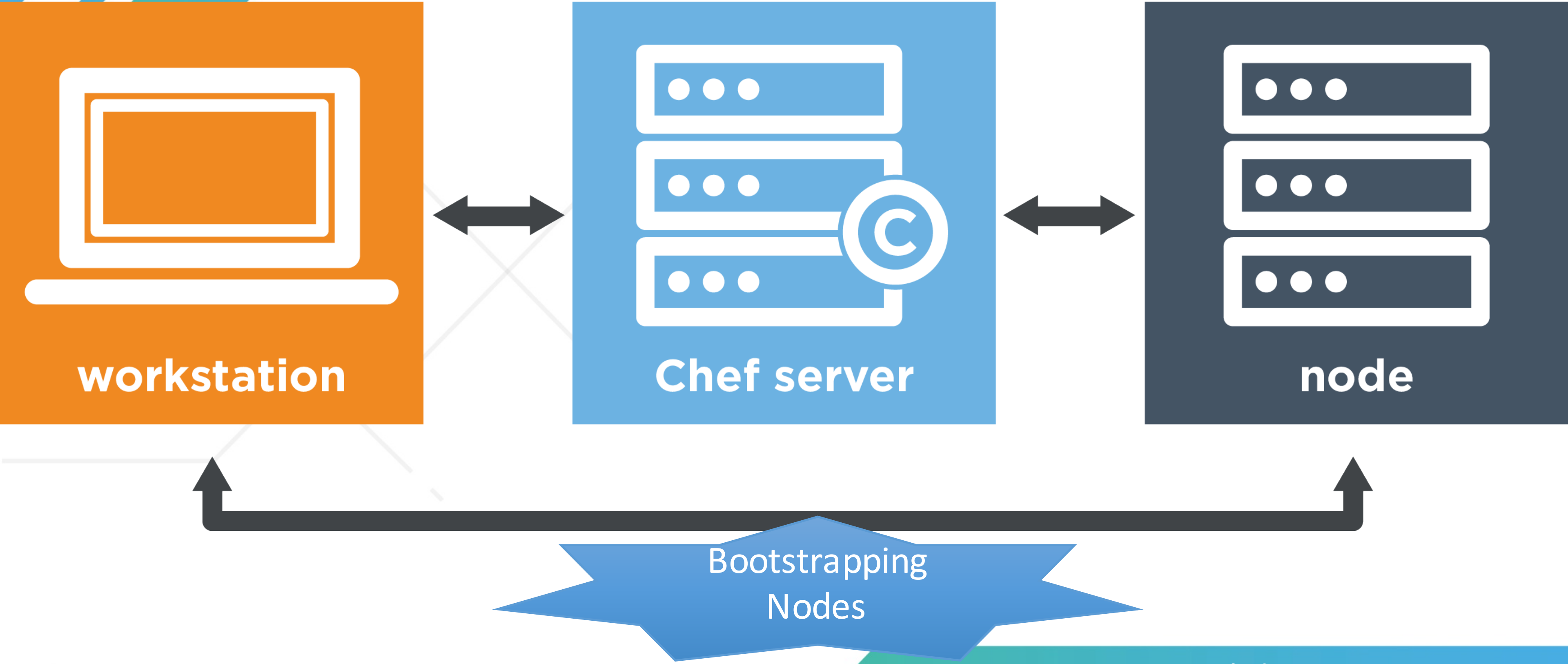
From the chef-repo execute:

knife cookbook upload sree

Chef: Workstation Setup: How to configure nodes/servers

```
docker  run --rm -ti --name chefnode1  --network chef schogini/chef-node
```

Chef: Chef Server



Chef: Organization Setup: Adding Servers as Nodes

1. From the workstation’s above chef-repo folder run this command for each node

```
knife bootstrap chefnode1 -x root -P <root-password> -N <node-name>
```

eg: knife bootstrap chefnode1 -x root -P sree1234 -N chefnode1

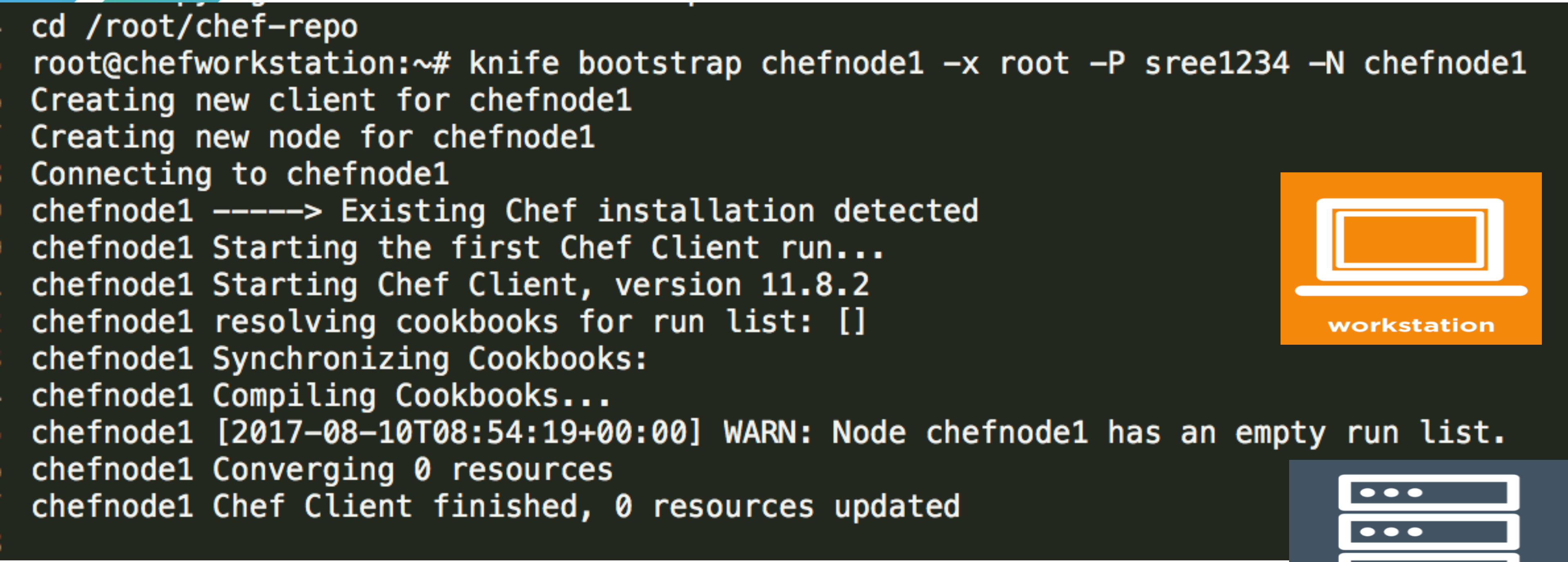
Notes:

1) In production setup, do not specify the node name, it will be auto populated by the nodes hostname.

2) Nodes must allow root ssh login during the initial setup, this must be blocked after that.

Chef: Workstation Setup: Bootstrapping nodes

Login to the workstation and run the knife command to talk to the nodes



```
cd /root/chef-repo
root@chefworkstation:~# knife bootstrap chefnode1 -x root -P sree1234 -N chefnode1
Creating new client for chefnode1
Creating new node for chefnode1
Connecting to chefnode1
chefnode1 -----> Existing Chef installation detected
chefnode1 Starting the first Chef Client run...
chefnode1 Starting Chef Client, version 11.8.2
chefnode1 resolving cookbooks for run list: []
chefnode1 Synchronizing Cookbooks:
chefnode1 Compiling Cookbooks...
chefnode1 [2017-08-10T08:54:19+00:00] WARN: Node chefnode1 has an empty run list.
chefnode1 Converging 0 resources
chefnode1 Chef Client finished, 0 resources updated
```

workstation

node

Chef: Organization Setup: Adding Nodes

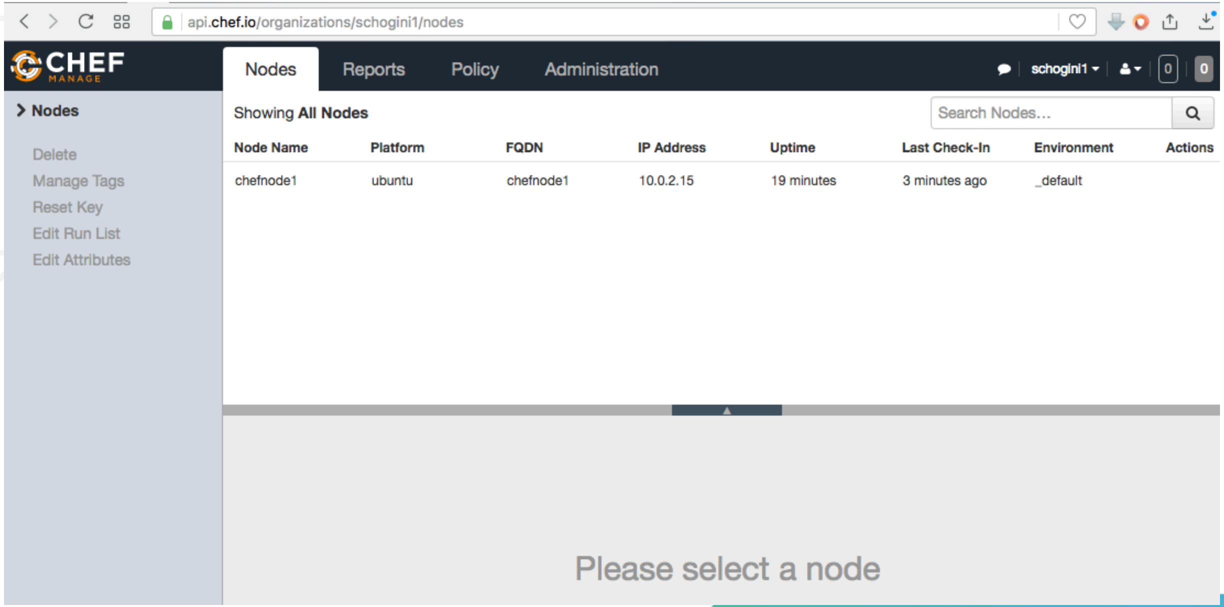
After you bootstrap a node, the chef-client will poll the chef-server and the server will become aware of the chef node.

eg: knife bootstrap chefnode1 -x root -P sree1234 -N chefnode1

<u>NOTE:</u> If you get an error like this:

***ERROR:** Net::SSH::HostKeyMismatch: fingerprint 7b:12:4c:1e:05:d5:61:55:66:97:6e:92:27:97:37:91 does not match for "chefnode1,172.28.128.30"*

Remove the chefnode1 line from the ~/.ssh/known_hosts line



Showing **All Nodes**

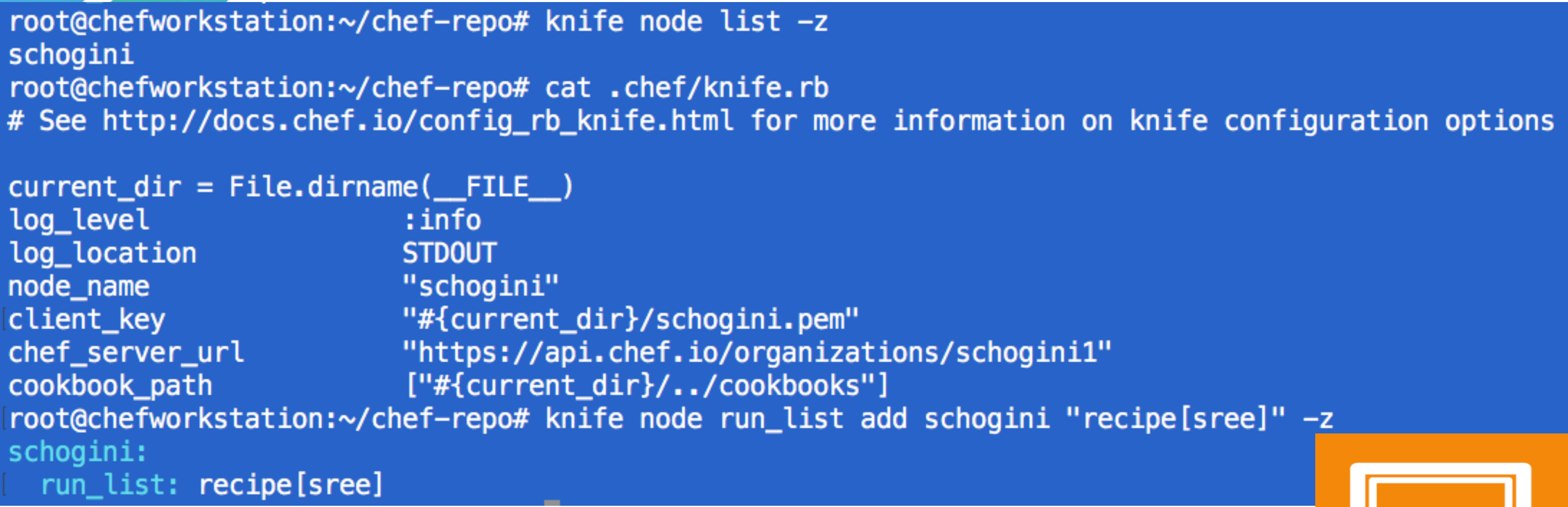
Node Name	Platform	FQDN	IP Address	Uptime	Last Check-In	Environment	Actions
chefnode1	ubuntu	chefnode1	10.0.2.15	19 minutes	3 minutes ago	_default	

Please select a node

Chef server

Chef: Workstation Setup: Applying Recipe-3: knife local mode

First find the node name, then add run list! Then run chef client



```
root@chefworkstation:~/chef-repo# knife node list -z
schogini
root@chefworkstation:~/chef-repo# cat .chef/knife.rb
# See http://docs.chef.io/config_rb_knife.html for more information on knife configuration options

current_dir = File.dirname(__FILE__)
log_level                :info
log_location             STDOUT
node_name                "schogini"
client_key               "#{current_dir}/schogini.pem"
chef_server_url          "https://api.chef.io/organizations/schogini1"
cookbook_path            ["#{current_dir}/../cookbooks"]
root@chefworkstation:~/chef-repo# knife node run_list add schogini "recipe[sree]" -z
schogini:
  run_list: recipe[sree]
```

chef-client -z

Chef: HW-1

Chef

1. Setup Chef Workstation
2. Create Chef Server account
3. Experiment with different recipes
4. Upload the recipe to Chef Server
5. Create Chef Node
6. Add Chef Node to CS

Cognixia

THANK YOU