

Guides & Tutorials (/docs/)

» Applications (/docs/applications/)

» Chef (/docs/applications/chef/)

» Deploy a Chef Server, Workstation and Node on Ubuntu 14.04

Deploy a Chef Server, Workstation and Node on Ubuntu 14.04

Updated Wednesday, June 10th, 2015 by Elle Krout

Use promo code **DOCS10** for \$10 credit on a new account.

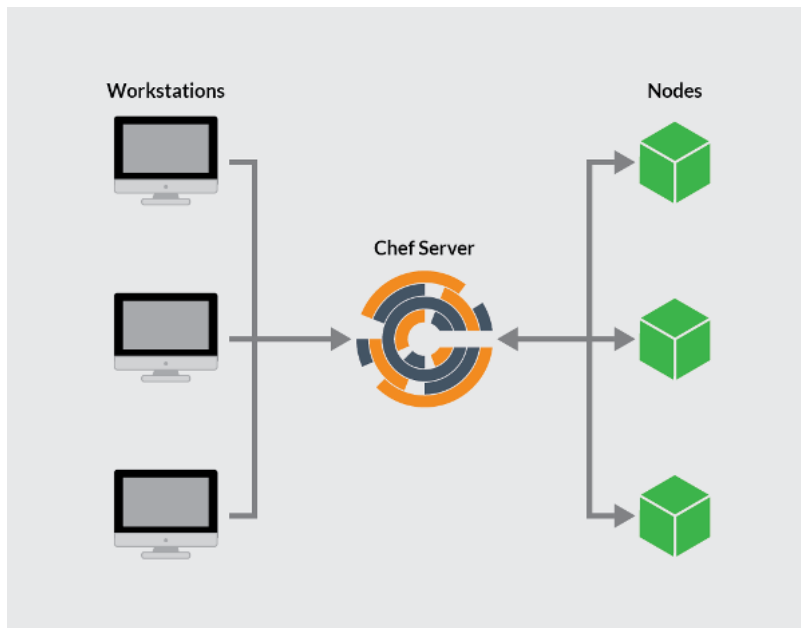
[Try this Guide](#)

Contribute on GitHub

View Project (<https://github.com/linode/docs>) | View File (<https://github.com/linode/docs/blob/master/docs/applications/chef/deploy-a-chef-server-workstation-and-node-on-ubuntu-14-04.md>) | Edit File (<https://github.com/linode/docs/edit/master/docs/applications/chef/deploy-a-chef-server-workstation-and-node-on-ubuntu-14-04.md>)

Chef is an automation platform that “turns infrastructure into code,” allowing users to manage and deploy resources across multiple servers, or *nodes*. Chef allows users to create and download recipes (stored in cookbooks) to automate content and policies on these nodes.

Chef is comprised of a Chef server, one or more workstations, and a number of nodes that are managed by the chef-client installed on each node.



(/docs/assets/chef_graph.png)

This guide will show users how to create and configure a Chef server, a virtual workstation, and how to bootstrap a node to run the chef-client, all on individual Linodes.

Note

This guide is written for a non-root user. Commands that require elevated privileges are prefixed with `sudo`. If you're not familiar with the `sudo` command, you can check our Users and Groups (/docs/tools-reference/linux-users-and-groups) guide.

Prerequisites

- One 4GB Linode to host the Chef server, running Ubuntu 14.04

- Two Linodes of any size to host a workstation and a node, each running Ubuntu 14.04
- Each Linode should be configured by following the Getting Started (/docs/getting-started) guide; also consider following the Securing Your Server (/docs/security/securing-your-server/) guide
- Each Linode needs to be configured to have a valid FQDN
- Ensure that all servers are up-to-date:

```
1 sudo apt-get update && sudo apt-get upgrade
```

The Chef Server

The Chef server is the hub of interaction between all workstations and nodes using Chef. Changes made through workstations are uploaded to the Chef server, which is then accessed by the chef-client and used to configure each individual node.

Install the Chef Server

1. Download (<https://downloads.chef.io/chef-server/ubuntu/>) the latest Chef server core (12.0.8 at the time of writing):

```
1 wget https://web-dl.packagecloud.io/chef/stable/packages/ubuntu/trusty/chef-server-core_12.0.8-1_amd64.deb
```

2. Install the server:

```
1 sudo dpkg -i chef-server-core_*.deb
```

3. Remove the download file:

```
1 rm chef-server-core_*.deb
```

4. Run the `chef-server-ctl` command to start the Chef server services:

```
1 sudo chef-server-ctl reconfigure
```

Create a User and Organization

1. In order to link workstations and nodes to the Chef server, an administrator and an organization need to be created with associated RSA private keys. From the home directory, create a `.chef` directory to store the keys:

```
1 mkdir .chef
```

2. Create an administrator. Change `username` to your desired username, `firstname` and `lastname` to your first and last name, `email` to your email, `password` to a secure password, and `username.pem` to your username followed by `.pem`:

```
1 sudo chef-server-ctl user-create username firstname lastname email password --filename ~/.chef/username.pem
```

3. Create an organization. The `shortname` value should be a basic identifier for your organization with no spaces, whereas the `fullname` can be the full, proper name of the organization. The `association_user` value `username` refers to the username made in the step above:

```
1 sudo chef-server-ctl org-create shortname fullname --association_user username --filename ~/.chef/shortname.pem
```

With the Chef server installed and the needed RSA keys generated, you can move on to configuring your workstation, where all major work will be performed for your Chef's nodes.

Workstations

Your Chef workstation will be where you create and configure any recipes, cookbooks, attributes, and other changes made to your Chef configurations. Although this can be a local machine of any OS, there is some benefit to keeping a remote server as your workstation since it can be accessed from anywhere.

Setting Up a Workstation

1. Download (<https://downloads.chef.io/chef-dk/ubuntu/>) the latest Chef Development Kit (0.5.1 at time of writing):

```
1 wget https://opscode-omnibus-packages.s3.amazonaws.com/ubuntu/12.04/x86_64/chefdk_0.5.1-1_amd64.deb
```

2. Install ChefDK:

```
1 sudo dpkg -i chefdk_*.deb
```

3. Remove the install file:

```
1 rm chefdk_*.deb
```

4. Verify the components of the development kit:

```
1 chef verify
```

It should output:

```
1 Running verification for component 'berkshelf'
2 Running verification for component 'test-kitchen'
3 Running verification for component 'chef-client'
4 Running verification for component 'chef-dk'
5 Running verification for component 'chefspect'
6 Running verification for component 'rubocop'
7 Running verification for component 'fauxhai'
8 Running verification for component 'knife-spork'
9 Running verification for component 'kitchen-vagrant'
10 Running verification for component 'package installation'
11 .....
12 -----
13 Verification of component 'rubocop' succeeded.
14 Verification of component 'kitchen-vagrant' succeeded.
15 Verification of component 'fauxhai' succeeded.
16 Verification of component 'berkshelf' succeeded.
17 Verification of component 'knife-spork' succeeded.
18 Verification of component 'test-kitchen' succeeded.
19 Verification of component 'chef-dk' succeeded.
20 Verification of component 'chef-client' succeeded.
21 Verification of component 'chefspect' succeeded.
22 Verification of component 'package installation' succeeded.
```

5. Generate the chef-repo and move into the newly-created directory:

```
1 chef generate repo chef-repo
2 cd chef-repo
```

6. Make the `.chef` directory:

```
1 mkdir .chef
```

Add the RSA Private Keys

1. The RSA private keys generated when setting up the Chef server will now need to be placed on the workstation. The process behind this will vary depending on if you are using SSH key pair authentication to log into your Linodes.

- If you are **not** using key pair authentication, then copy the file directly off of the Chef Server. replace `user` with your username on the server, and `123.45.67.89` with the URL or IP of your Chef Server:

```
1 scp user@123.45.67.89:~/.chef/*.pem ~/chef-repo/.chef/
```

- If you **are** using key pair authentication, then from your **local terminal** copy the `.pem` files from your server to your workstation using the `scp` command. Replace `user` with the appropriate username, and `123.45.67.89` with the URL or IP for your Chef Server and `987.65.43.21` with the URL or IP for your workstation:

```
1 scp -3 user@123.45.67.89:~/.chef/*.pem user@987.65.43.21:~/chef-repo/.chef/
```

2. Confirm that the files have been copied successfully by listing the contents of the `.chef` directory:

```
1 ls ~/chef-repo/.chef
```

Your `.pem` files should be listed.

Add Version Control

The workstation is used to add and edit cookbooks and other configuration files. It is beneficial to implement some form of version control. For this, Git proves to be useful.

1. Download Git:

```
1 sudo apt-get install git
```

2. Configure Git by adding your username and email, replacing the needed values:

```
1 git config --global user.name yourname
2 git config --global user.email user@email.com
```

3. From the `chef-repo`, initialize the repository:

```
1 git init
```

4. Add the `.chef` directory to the `.gitignore` file:

```
1 echo ".chef" > .gitignore
```

5. Add and commit all existing files:

```
1 git add .
2 git commit -m "initial commit"
```

6. Make sure the directory is clean:

```
1 git status
```

It should output:

```
1 nothing to commit, working directory clean
```

Generate knife.rb

1. Create a knife configuration file by navigating to your `~/chef-repo/.chef` folder and opening a file named `knife.rb` in your chosen text editor.
2. Copy the following configuration into the `knife.rb` file:

File: `~/chef-repo/.chef/knife.rb`

```
1 log_level           :info
2 log_location        STDOUT
3 node_name            'username'
4 client_key           '~/chef-repo/.chef/username.pem'
5 validation_client_name 'shortname-validator'
6 validation_key       '~/chef-repo/.chef/shortname.pem'
7 chef_server_url      'https://123.45.67.89/organizations/shortname'
8 syntax_check_cache_path '~/chef-repo/.chef/syntax_check_cache'
9 cookbook_path [ '~/chef-repo/cookbooks' ]
```

Change the following:

- The value for `node_name` should be the username that was created above.
 - Change `username.pem` under `client_key` to reflect your `.pem` file for your **user**.
 - The `validation_client_name` should be your organization's `shortname` followed by `-validator`.
 - `shortname.pem` in the `validation_key` path should be set to the shortname was was defined in the steps above.
 - Finally the `chef_server-url` needs to contain the IP address or URL of your Chef server, with the `shortname` in the file path changed to the shortname defined above.
3. Move to the `chef-repo` and copy the needed SSL certificates from the server:

```
1 cd ..
2 knife ssl fetch
```

4. Confirm that `knife.rb` is set up correctly by running the client list:

```
1 knife client list
```

This command should output the validator name.

With both the server and a workstation configured, it is possible to bootstrap your first node.

Bootstrap a Node

Bootstrapping a node installs the chef-client and validates the node, allowing it to read from the Chef server and make any needed configuration changes picked up by the chef-client in the future.

1. From your *workstation*, bootstrap the node either by using the node's root user, or a user with elevated privileges:

- As the node's root user, changing `password` to your root password and `nodename` to the desired name for your node. You can leave this off if you would like the name to default to your node's hostname:

```
1 knife bootstrap 123.45.67.89 -x root -P password --node-name nodename
```

- As a user with sudo privileges, change `username` to the username of a user on the node, `password` to the user's password and `nodename` to the desired name for the node. You can leave this off if you would like the name to default to your node's hostname:

```
1 knife bootstrap 123.45.67.89 -x username -P password --sudo --node-name nodename
```

2. Confirm that the node has been bootstrapped by listing the nodes:

```
1 knife node list
```

Your new node should be included on the list.

Download a Cookbook (Optional)

When using Chef you will want the chef-client to periodically run on your nodes and pull in any changes pushed to the Chef server. You will also want the `validation.pem` file that is uploaded to your node upon bootstrap to be deleted for security purposes. While these things can be done manually, it is often easier and more efficient to set it up as a cookbook.

This section is optional, but provides instructions on downloading a cookbook to your workstation, pushing it to a server, and includes the skeleton of a basic cookbook to expand and experiment with.

1. From your *workstation* download the cookbook and dependencies:

```
1 knife cookbook site install cron-delvalidate
```

2. Open the `default.rb` file to examine the default cookbook recipe:

File excerpt: `~/chef-repo/cookbooks/cron-delvalidate/recipes/default.rb`

```
1 #
2 # Cookbook Name:: cron-delvalidate
3 # Recipe:: Chef-Client Cron & Delete Validation.pem
4 #
5 #
6
7 cron "clientrun" do
8   minute '0'
9   hour '*/1'
10  command "/usr/bin/chef-client"
11  action :create
12 end
13
14 file "/etc/chef/validation.pem" do
15   action :delete
16 end
```

The resource `cron "clientrun" do` defines the cron action. It is set to run the chef-client action (`/usr/bin/chef-client`) every hour (`*/1` with the `*/` defining that it's every hour and not 1AM daily). The `action` code denotes that Chef is *creating* a new cronjob.

`file "/etc/chef/validation.pem" do` calls to the `validation.pem` file. The `action` defines that the file should be removed (`:delete`).

These are two very basic sets of code in Ruby, and provide an example of the code structure that will be used when creating Chef cookbooks. These examples can be edited and expanded as needed.

3. Add the recipe to your node's run list, replacing `nodename` with your node's name:

```
1 knife node run_list add nodename 'recipe[cron-delvalidate::default]'
```

4. Push the cookbook to the Chef server:

```
1 knife cookbook upload cron-delvalidate
```

This command is also used when updating cookbooks.

5. Switch to your *bootstrapped* node(s) and run the initial chef-client command:

```
1 chef-client
```

If running the node as a non-root user, append the above command with `sudo`.

The recipes in the run list will be pulled from the server and run. In this instance, it will be the `cron-delvalidate` recipe. This recipe ensures that any cookbooks made, pushed to the Chef Server, and added to the node's run list will be pulled down to bootstrapped nodes once an hour. This automated step eliminates connecting to the node in the future to pull down changes.

More Information

You may wish to consult the following resources for additional information on this topic. While these are provided in the hope that they will be useful, please note that we cannot vouch for the accuracy or timeliness of externally hosted materials.

- Chef (<http://www.chef.io>)

This guide is published under a CC BY-ND 4.0 (<https://creativecommons.org/licenses/by-nd/4.0>) license.

Get paid to write for Linode.

We're always expanding our docs. If you like to help people, can write, and want to earn some cash, learn how you can earn \$250 for every guide you write (</docs/contribute>) and we publish.

Get started in the Linode Cloud today.

Create an Account (<https://manager.linode.com/session/signup>)

Overview (<https://www.linode.com/linodes>)

Plans & Pricing (<https://www.linode.com/pricing>)

Features (<https://www.linode.com/linodes>)

Add-Ons (<https://www.linode.com/addons>)

Managed (<https://www.linode.com/managed>)

Professional Services (<https://www.linode.com/professional-services>)

Resources (<https://www.linode.com/docs>)

Guides & Tutorials (<https://www.linode.com/docs>)

Speed Test (<https://www.linode.com/speedtest>)

Forum (<https://forum.linode.com/>)

Chat (<https://www.linode.com/chat>)

System Status (<http://status.linode.com/>)

Company (<https://www.linode.com/about>)

About Us (<https://www.linode.com/about>)

Blog (<https://blog.linode.com>)

Press (<https://www.linode.com/press>)

Referral System (<https://www.linode.com/referrals>)

Careers (<https://www.linode.com/careers>)



Contact Us (<https://www.linode.com/contact>)

855-4-LINODE (tel:+18554546633)

(855-454-6633) (tel:+18554546633)

Intl.: +1 609-380-7100 (tel:+16093807100)

Email us (<mailto:support@linode.com>)

 (<https://facebook.com/linode>)  (<https://twitter.com/linode>)  (<https://plus.google.com/+linode/>)  (<https://linkedin.com/company/linode>)

© 2016 Linode, LLC

Terms of Service (<https://www.linode.com/tos>)

Privacy Policy (<https://www.linode.com/privacy>)

Security (<https://www.linode.com/security>)

Standards & Compliance (<https://www.linode.com/compliance>)

