In this twe use the following two EC2 instances:

1. Chef server on an AWS EC2 Ubuntu instance that was setup in EC2 Ubuntu 14.04
2. Workstation on an AWS EC2 Ubuntu 14.04 instance that we're going to setup in this chapter.

Setup SSH - EC2 Ubuntu 14.04

On EC2 instance for our Workstation, we need to make sure ports for SSH, HTTP, HTTPS, and DNS are opened:

Now, we can login to our workstation on EC2 we've just created using`ssh`. The workstation shares its private key with Chef Server (**Chef_Server.pem**):

As we've done for the server, we may want to set alias with shorter name. Modify the file called `~/.ssh/config`, add the following line:

```
Host Chef_Workstation
  Hostname      ec2-54-172-74-156.compute-1.amazonaws.com
  User          ubuntu
  IdentityFile  ~/.ssh/Chef_Workstation.pem
```

Then, our login to EC2 will be much easier:

```
$ ssh Chef_Workstation
ubuntu@ip-172-31-49-135:~$
```

omnibus installer

The omnibus installer is used to set up a workstation. The omnibus installer uses a single command to install the chef-client and all of its dependencies, including an embedded version of Ruby, RubyGems, OpenSSL, key-value stores, parsers, libraries, and command line utilities. The omnibus installer puts everything into a unique directory (`opt/opscode/`) so that the chef-client will not interfere with other applications that may be running on the target machine.

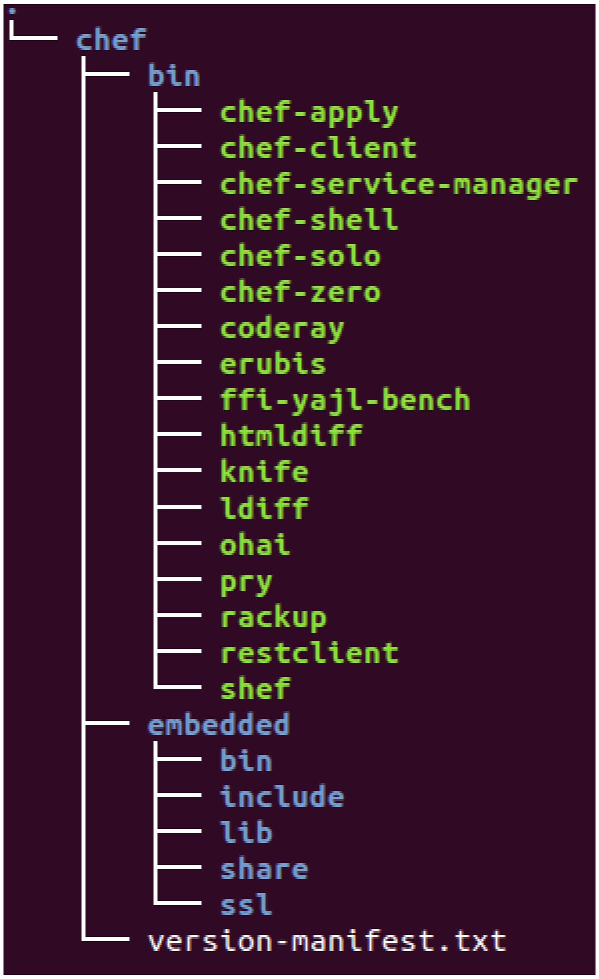
To install the chef-client on a workstation, we must run the omnibus installer. To run the omnibus installer, w need to download and run the client installation script from the Chef website:

```
ubuntu@ip-172-31-48-124:~$ curl -L
https://www.opscode.com/chef/install.sh | sudo bash

...

Downloading Chef  for ubuntu...

...

downloading https://opscode-omnibus-
packages.s3.amazonaws.com/ubuntu/13.04/x86_64/chef_11.16.4-
1_amd64.deb

  to file /tmp/install.sh.1411/chef_11.16.4-1_amd64.deb

...

Preparing to unpack .../chef_11.16.4-1_amd64.deb ...

Unpacking chef (11.16.4-1) ...

Setting up chef (11.16.4-1) ...

Thank you for installing Chef!

ubuntu@ip-172-31-48-124:~$
```

When the installation is finished enter the following to verify that the chef-client was installed. When the chef-client is installed correctly, the command shell will return a note that says the version of that was installed:

```
ubuntu@ip-172-31-48-124:~$ chef-client -v
Chef: 11.16.4
```

After the chef-client has been installed, the following folder structure will be present on the local machine:



```
.
└── chef
    ├── bin
    │   ├── chef-apply
    │   ├── chef-client
    │   ├── chef-service-manager
    │   ├── chef-shell
    │   ├── chef-solo
    │   ├── chef-zero
    │   ├── coderay
    │   ├── erubis
    │   ├── ffi-yajl-bench
    │   ├── htmldiff
    │   ├── knife
    │   ├── ldiff
    │   ├── ohai
    │   ├── pry
    │   ├── rackup
    │   ├── restclient
    │   └── shef
    ├── embedded
    │   ├── bin
    │   ├── include
    │   ├── lib
    │   ├── share
    │   └── ssl
    └── version-manifest.txt
```

install git

`git` must be installed before the `chef-repo` can be cloned to the workstation from `GitHub`:

```
ubuntu@ip-172-31-48-124:~$ sudo apt-get update
```

```
ubuntu@ip-172-31-48-124:~$ sudo apt-get install git
```

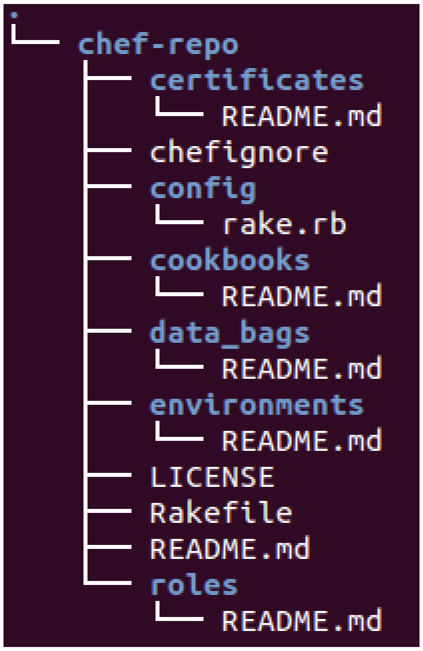
Set up the chef-repo

Let's clone the chef-repo skeleton directory.

The chef-repo on GitHub must be cloned to every workstation that will interact with a Chef server. To clone the `chef-repo`, in a command window, open the home directory, and then clone the chef-repo:

```
ubuntu@ip-172-31-48-124:~$ cd ~

ubuntu@ip-172-31-48-124:~$ git clone git://github.com/opscode/chef-repo.git
```

After the chef-repo has been cloned, the following folder structure will be present on the local machine:



```
.
└── chef-repo
    ├── certificates
    │   └── README.md
    ├── chefignore
    ├── config
    │   └── rake.rb
    ├── cookbooks
    │   └── README.md
    ├── data_bags
    │   └── README.md
    ├── environments
    │   └── README.md
    ├── LICENSE
    ├── Rakefile
    ├── README.md
    └── roles
        └── README.md
```

Setup email and name with git:

```
ubuntu@ip-172-31-48-124:~/chef-repo$ git config --global user.email
"k@xyz.com"

ubuntu@ip-172-31-48-124:~/chef-repo$ git config --global user.name
"xyz"
```

Create .chef Directory

The `.chef` directory is used to store three files:

- knife.rb
- ORGANIZATION-validator.pem
- USER.pem

Where ORGANIZATION and USER represent strings that are unique to each organization. These files must be present in the `.chef` directory in order for a workstation to be able to connect to a Chef server.

To create the `.chef` directory:

```
ubuntu@ip-172-31-48-124:~$ sudo mkdir -p ~/chef-repo/.chef
```

Add `.chef` to the `.gitignore` file to prevent uploading the contents of the `.chef`folder to `GitHub`:

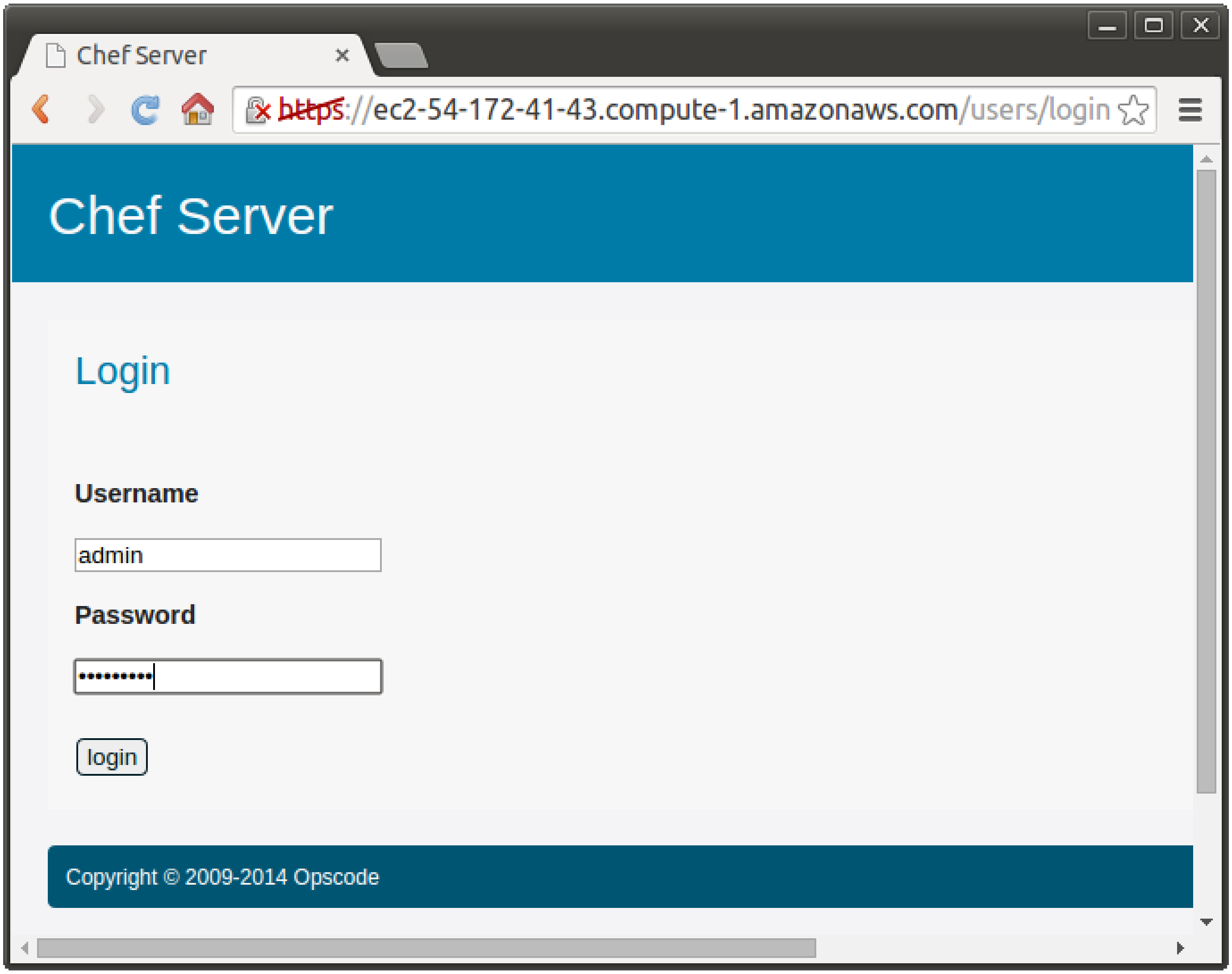
```
ubuntu@ip-172-31-48-124:~$ echo '.chef' >> ~/chef-repo/.gitignore
```

Login to Chef server

Login to the Chef Server with the admin credentials. we can access the web interface by typing `https://domain` from our browser. Because the SSL certificate is signed by an authority not recognized by our browser, we may get a warning. Click on the "Proceed anyway" button.

Login with the default admin credentials:

```
username: admin
password: p@ssw0rd1
```



~~https~~://ec2-54-172-41-43.compute-1.amazonaws.com/users/login

Chef Server

Login

Username

admin

Password

•••••••••

Copyright © 2009-2014 Opscode

When we try to login, it may won't allow, and keep giving us the same window. To fix this issue, we need to modify `session_store.rb` file under `/opt/chef-server/embedded/service/chef-server-webui/config/initializers` file:

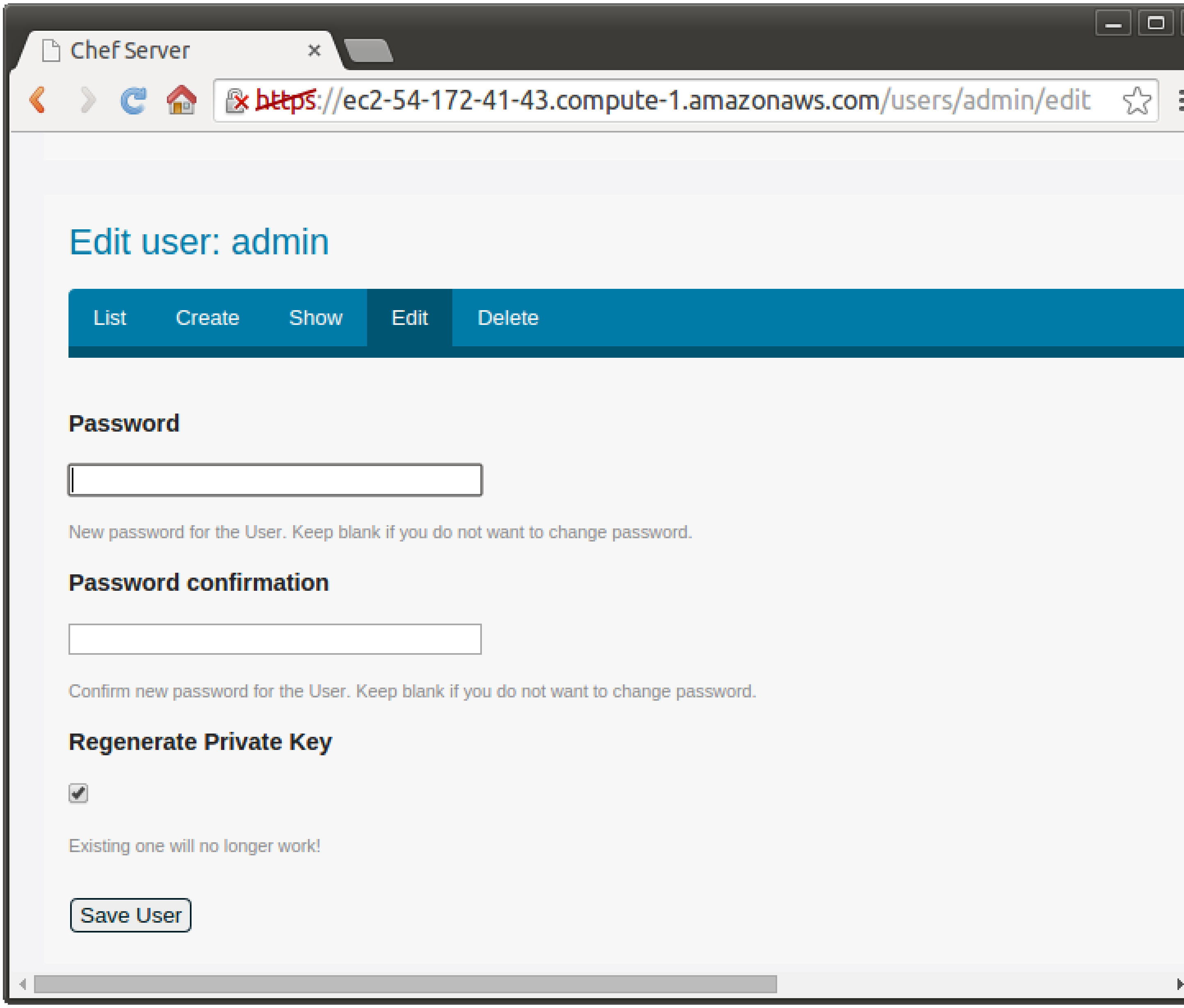
```
# Be sure to restart your server when you modify this file.


#ChefServerWebui::Application.config.session_store    :cookie_store,
:key => '_sandbox_session', :domain => :all

ChefServerWebui::Application.config.session_store     :cookie_store,
:key  =>  '_sandbox_session',  :domain  =>  'ec2-54-172-41-43.compute-
1.amazonaws.com'
```

Note that we changed `:all` to `public domain name`.

Then, we need to restart chef server:

```
ubuntu@ip-172-31-52-254:~$ sudo chef-server-ctl restart
```

After login, we get a new window for password confirmation/change:


Chef Server
~~https~~://ec2-54-172-41-43.compute-1.amazonaws.com/users/admin/edit
List
Create
Show
Edit
Delete
Save User

Edit user: admin

Password

New password for the User. Keep blank if you do not want to change password.

Password confirmation

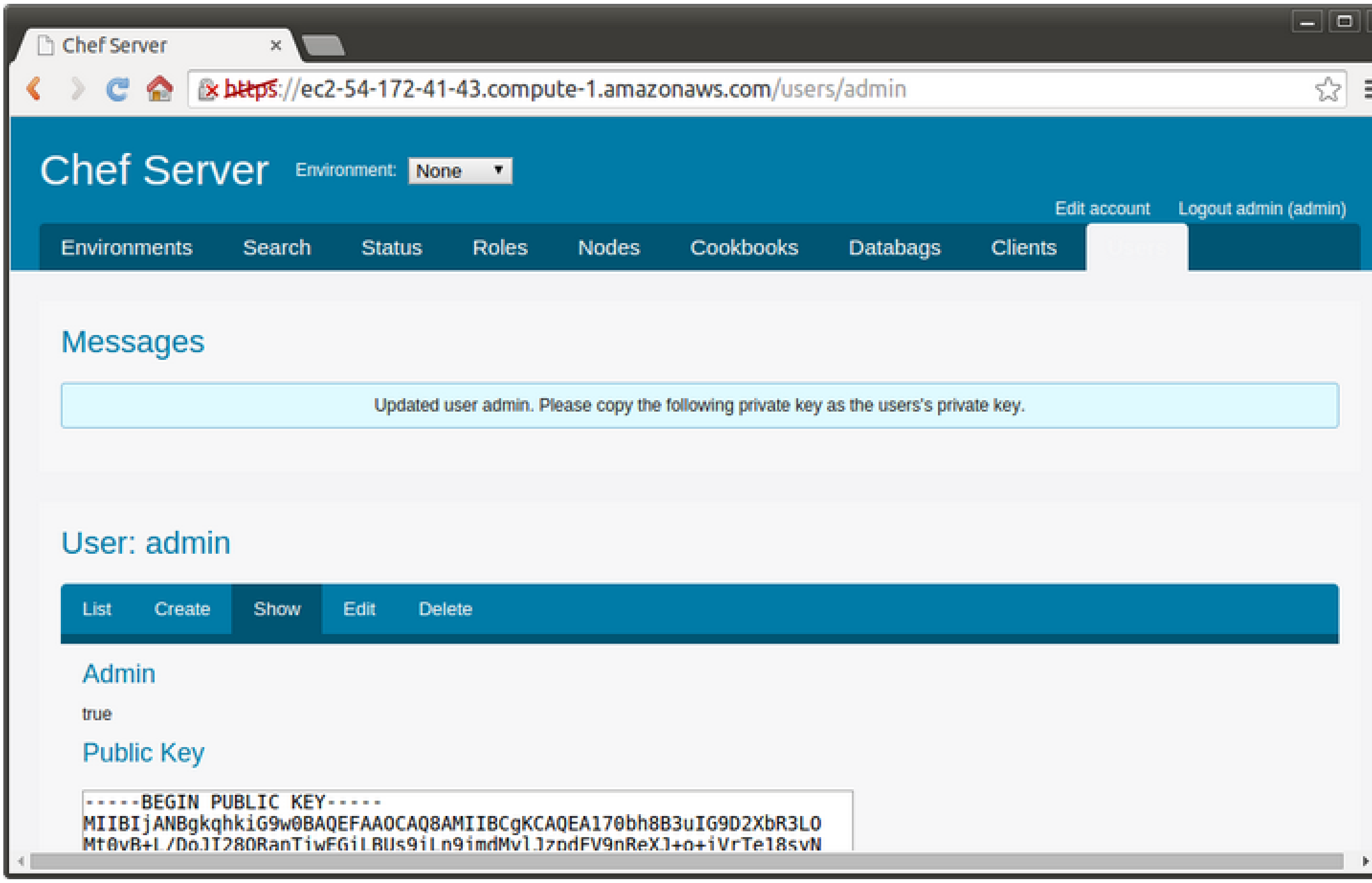
Confirm new password for the User. Keep blank if you do not want to change password.

Regenerate Private Key

☑

Existing one will no longer work!

Click on "Save User":



Messages

Updated user admin. Please copy the following private key as the users's private key.

User: admin

Admin

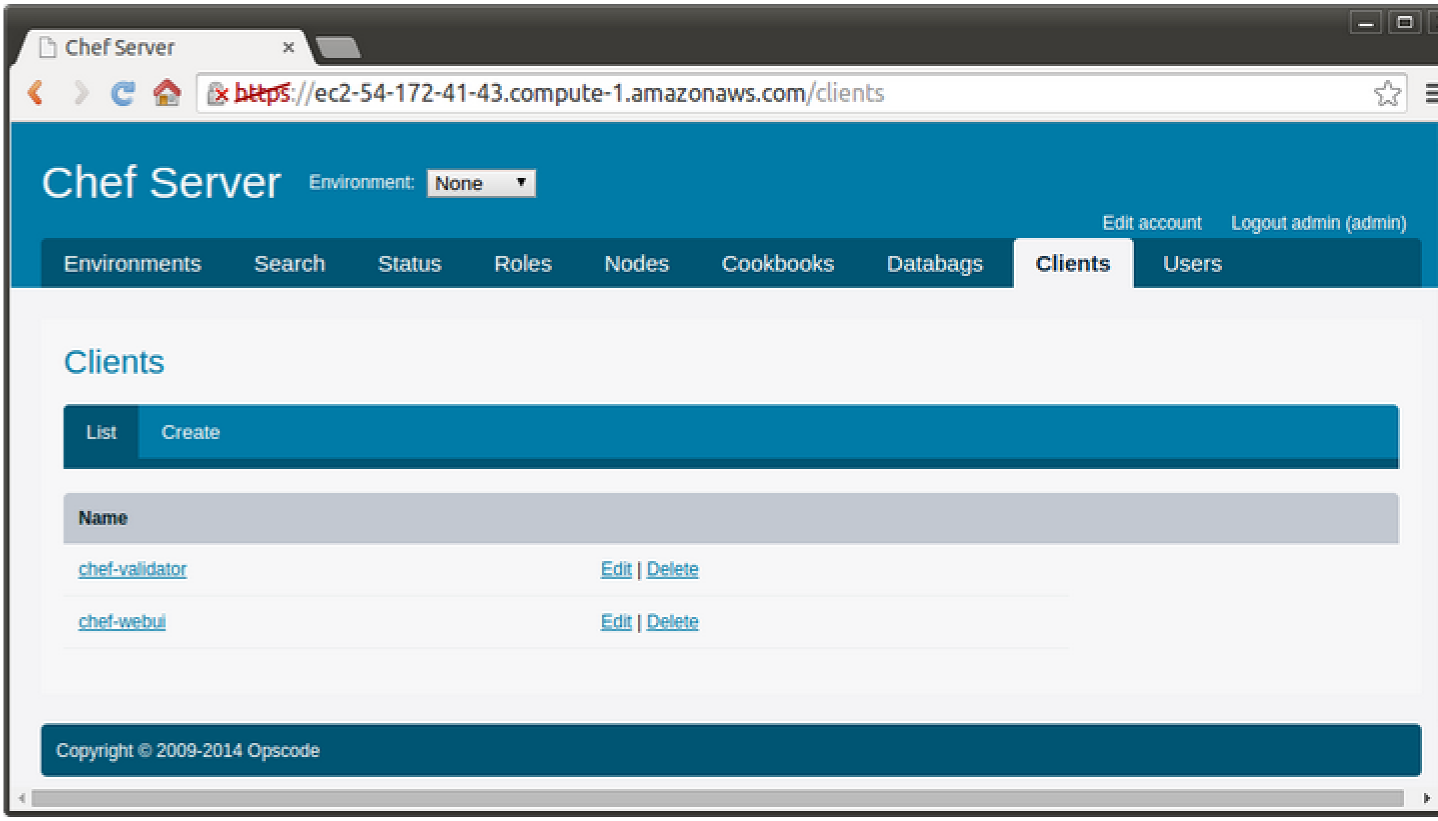
true

Public Key

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA170bh8B3uIG9D2XbR3LO
Mt0vB+L/DoJI28ORanTiwFGiLBUs9iLn9imdMvlJzpdFV9nReXJ+o+iVrTel8syN
```

Copy the private key and save it in `admin.pem` file in `~/chef-repo/.chef` directory.

Click on the "Clients" tab in the top navigation bar:

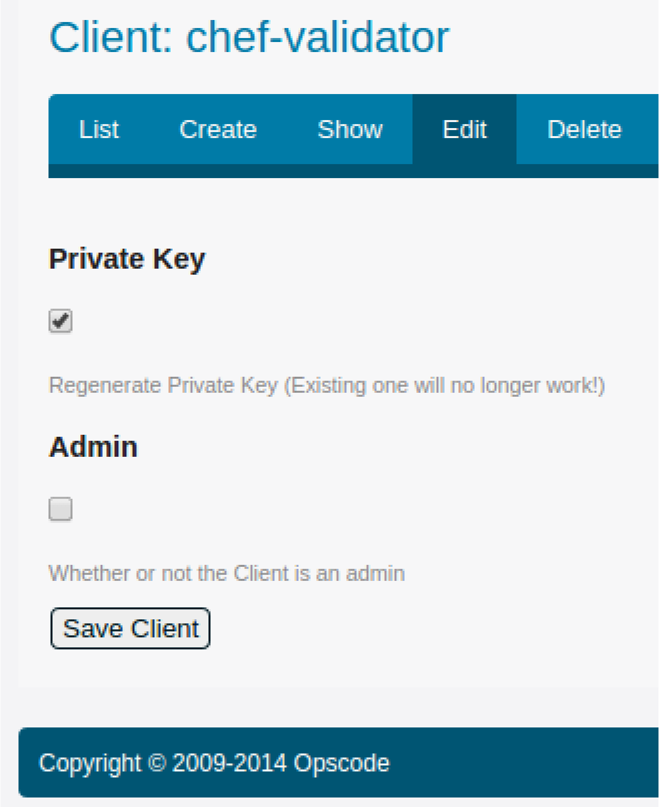


Clients

Name	
chef-validator	Edit | Delete
chef-webui	Edit | Delete

Copyright © 2009-2014 Opscode

Click on the "Edit" button associated with the `chef-validator` client. Regenerate the private key by selecting that check box and clicking "Save Client":



Client: chef-validator

Private Key

☑

Regenerate Private Key (Existing one will no longer work!)

Admin

□

Whether or not the Client is an admin

Copyright © 2009-2014 Opscode

Copy the private key and save it in the `chef-validator.pem` file in`~/chef-repo/.chef` directory.

Config Knife

`Knife` is a command-line tool that provides an interface between a local chef-repo and the Chef server. Knife helps provisioning resources, manage recipes/cookbooks, nodes etc.

Create a `knife.rb` file. This configuration file must be created by running the `knife configure --initial` command on the machine that will be run as a workstation.

The validation_key attribute in the `knife.rb` file must specify the path to the validation key. The validation_client_name attribute defaults to`chef-validator` (which is the `chef-validator.pem` private key created by the open source Chef server on startup). When prompted for the URL for the Chef server, use the FQDN for the Chef server:

```
ubuntu@ip-172-31-48-124:~/chef-repo$ sudo knife configure --initial

WARNING: No knife configuration file found

Where should I put the config file? [/home/ubuntu/.chef/knife.rb] /home/ubuntu/chef-repo/.chef/knife.rb

Please enter the chef server URL: [https://ip-172-31-48-124.ec2.internal:443] https://ec2-54-172-41-43.compute-1.amazonaws.com

Please enter a name for the new user: [ubuntu]

Please enter the existing admin name: [admin]

Please enter the location of the existing admin's private key: [/etc/chef-server/admin.pem] /home/ubuntu/chef-repo/.chef/admin.pem

Please enter the validation clientname: [chef-validator]

Please enter the location of the validation key: [/etc/chef-server/chef-validator.pem] /home/ubuntu/chef-repo/.chef/chef-validator.pem

Please enter the path to a chef repository (or leave blank): /home/ubuntu/chef-repo

Creating initial API user...

Please enter a password for the new user:

Created user[xyz]

Configuration file written to /home/ubuntu/chef-repo/.chef/knife.rb

ubuntu@ip-172-31-48-124:~/chef-repo$ knife user list

admin

xyz
```

Verify that the files are in the `.chef` folder:

```
ubuntu@ip-172-31-48-124:~/chef-repo/.chef$ pwd

/home/ubuntu/chef-repo/.chef
```

```
ubuntu@ip-172-31-48-124:~/chef-repo/.chef$ ls
admin.pem  xyz.pem  chef-validator.pem  knife.rb
```

Add Ruby to the $PATH environment variable

The chef-client includes a stable version of Ruby as part of the omnibus installer. The path to this version of Ruby must be added to the `$PATH` environment variable and saved in the configuration file for the command shell that is used on the workstation:

```
ubuntu@ip-172-31-48-124:~/chef-repo$ echo 'export
PATH="/opt/chef/embedded/bin:$PATH"' >> ~/.configuration_file &&
source ~/.configuration_file
```

where configuration_file is the name of the configuration file for the specific command shell. For example, if Bash were the command shell and the configuration file were named `bash_profile`:

```
ubuntu@ip-172-31-48-124:~/chef-repo$ echo 'export
PATH="/opt/chef/embedded/bin:$PATH"' >> ~/.bash_profile && source
~/.bash_profile
```

Verify the chef-client install

A workstation is installed correctly when it is able to use `knife` to communicate with the `Chef server`. To verify that a workstation can connect to the Chef server:

```
ubuntu@ip-172-31-48-124:~$ cd chef-repo/
ubuntu@ip-172-31-48-124:~/chef-repo$ knife user list
admin
```

```
xyz

ubuntu@ip-172-31-48-124:~/chef-repo$ knife client list
chef-validator
chef-webui
```

If this is successful, then our workstation can successfully communicate with our server.

knife.rb

Just for reference, here is `knife.rb` in `/home/ubuntu/chef-repo/.chef/` folder:

```
log_level                :info
log_location             STDOUT
node_name                'xyz'
client_key               '/home/ubuntu/chef-repo/.chef/xyz.pem'
validation_client_name   'chef-validator'
validation_key           '/home/ubuntu/chef-repo/.chef/chef-validator.pem'
chef_server_url          'https://ec2-54-172-41-43.compute-1.amazonaws.com'
syntax_check_cache_path  '/home/ubuntu/chef-repo/.chef/syntax_check_cache'
cookbook_path [ '/home/ubuntu/chef-repo/cookbooks' ]
```

Note

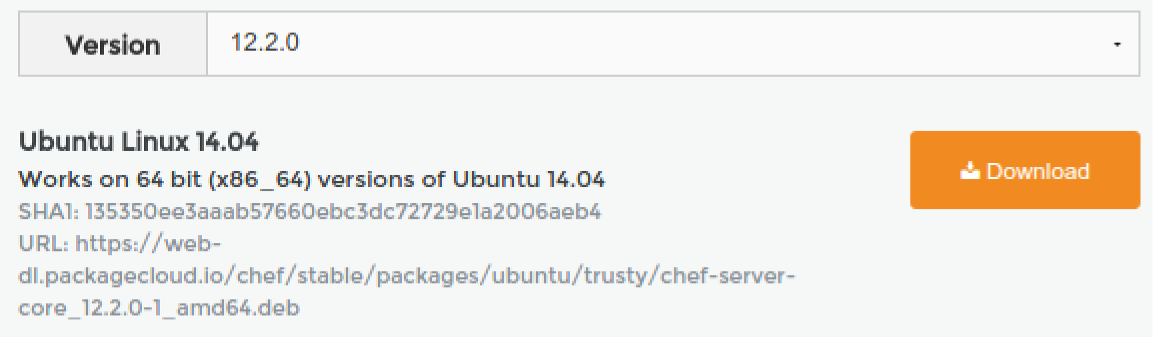
If we don't prepend `https://` to the url when we do code>knife configure --initial, we may get the following error:

```
ubuntu@ip-172-31-48-124:~/chef-repo$ knife client list
ERROR: TypeError: can't dup NilClass
```

Install Chef server on Amazon EC2 Ubuntu 14.04

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.

1. Get Chef 12.2.0 package from https://downloads.chef.io/chef-server/ubuntu/



Version 12.2.0

Ubuntu Linux 14.04

Works on 64 bit (x86_64) versions of Ubuntu 14.04

SHA1: 135350ee3aaab57660ebc3dc72729e1a2006aeb4

URL: https://web-dl.packagecloud.io/chef/stable/packages/ubuntu/trusty/chef-server-core_12.2.0-1_amd64.deb

```
ubuntu@ip-172-31-10-101:~$          wget          https://web-
dl.packagecloud.io/chef/stable/packages/ubuntu/trusty/chef-
server-core_12.2.0-1_amd64.deb
```

2. Install the Chef server:

```
ubuntu@ip-172.31.10.101:~$ sudo dpkg -i chef-server*
```

This will install the base Chef 12 system onto the server.

3. Once the installation is complete, we should run `chef-server-ctl reconfigure` command to start the Chef server services. It configures the components that make up the server to work together in our specific environment:

```
ubuntu@ip-172.31.10.101:~$ sudo chef-server-ctl reconfigure
...
```

```
Chef Server Reconfigured!
```

Once the installation is complete, you must call the command, which configures the components that make up the server to work together in your specific environment:

```
ubuntu@ip-172.31.10.101:~$ sudo chef-server-ctl status
run: bookshelf: (pid 4486) 39s; run: log: (pid 4526) 39s
run: nginx: (pid 4269) 44s; run: log: (pid 4679) 35s
run: oc_bifrost: (pid 4199) 46s; run: log: (pid 4246) 45s
run: oc_id: (pid 4253) 45s; run: log: (pid 4258) 44s
run: opscode-erchef: (pid 4591) 36s; run: log: (pid 4580) 38s
run: opscode-expander: (pid 4367) 41s; run: log: (pid 4469) 40s
run: opscode-expander-reindexer: (pid 4435) 40s; run: log: (pid 4475) 40s
run: opscode-solr4: (pid 4308) 42s; run: log: (pid 4344) 42s
run: postgresql: (pid 4151) 46s; run: log: (pid 4174) 46s
run: rabbitmq: (pid 1685) 136s; run: log: (pid 4138) 47s
run: redis_lb: (pid 4086) 64s; run: log: (pid 4673) 35s
ubuntu@ip-172.31.10.101:~$ sudo chef-server-ctl test
```

The Chef core server is now installed and started. The next steps is to configure it to allow us to log in.

Create a default user and organization

The next step is to create a default user and organization for the chef-server.

Next, we need to create an admin user. This will be the username that will have access to make changes to the infrastructure components in the organization we will be creating.

We can do this using the **user-create** subcommand of the **chef-server-ctl** command. The command requires a number of fields to be passed in during the creation process.

We will create a user with the following information:

- Username: admin
- First Name: admin
- Last Name: admin
- Email: admin@xyz.com
- Password: password
- Filename: admin.pem

Also, we will create an organization with the following information:

- Short Name: xyz
- Long Name: xyz.com
- Association User: admin
- Filename: xyz.pem

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:

```
ubuntu@ip-172-31-10-101:~$ mkdir .chef
```

2. Let's create an administrator:

```
3. ubuntu@ip-172-31-10-101:~$  sudo  chef-server-ctl  user-create
   admin admin admin admin@xyz.com password -f ~/.chef/admin.pem

4.

5. We should now have a private key called admin.pem in ~/.chef/
   directory.
```

```

```

6. Create an organization with the **org-create** subcommand:

```
ubuntu@ip-172-31-10-101:~$ sudo chef-server-ctl org-create xyz
"xyz.com" --association_user admin -f ~/.chef/xyz.pem
```

Now, we should have two .pem key files in **~/.chef/** directory:

```
ubuntu@ip-172-31-10-101:~/.chef$ ls

admin.pem  xyz.pem
```

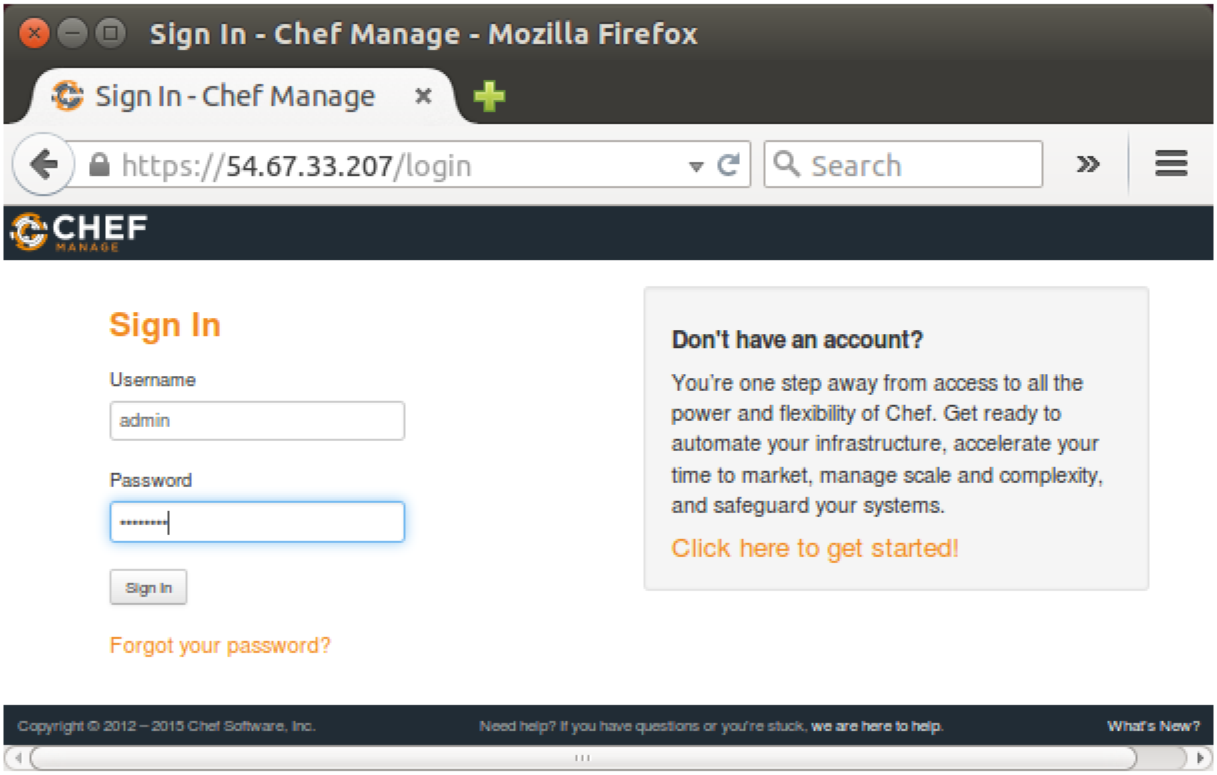
We will need to connect to this server and download these keys to our workstation momentarily. For now though, our Chef server installation is complete.

Opscode Manage (GUI)

Let's install the GUI plugin for the Chef:

```
ubuntu@ip-172-31-10-101:~$  sudo  chef-server-ctl  install  opscode-
manage

ubuntu@ip-172-31-10-101:~$ sudo opscode-manage-ctl reconfigure

ubuntu@ip-172-31-10-101:~$ sudo chef-server-ctl reconfigure
```

We can access the web interface by typing **https://domain** from our browser. Because the SSL certificate is signed by an authority not recognized by our browser, we may get a warning. Click on the "Proceed anyway" button.



https://54.67.33.207/login

Sign In

Username: admin

Password: ••••••••

Don't have an account?

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.

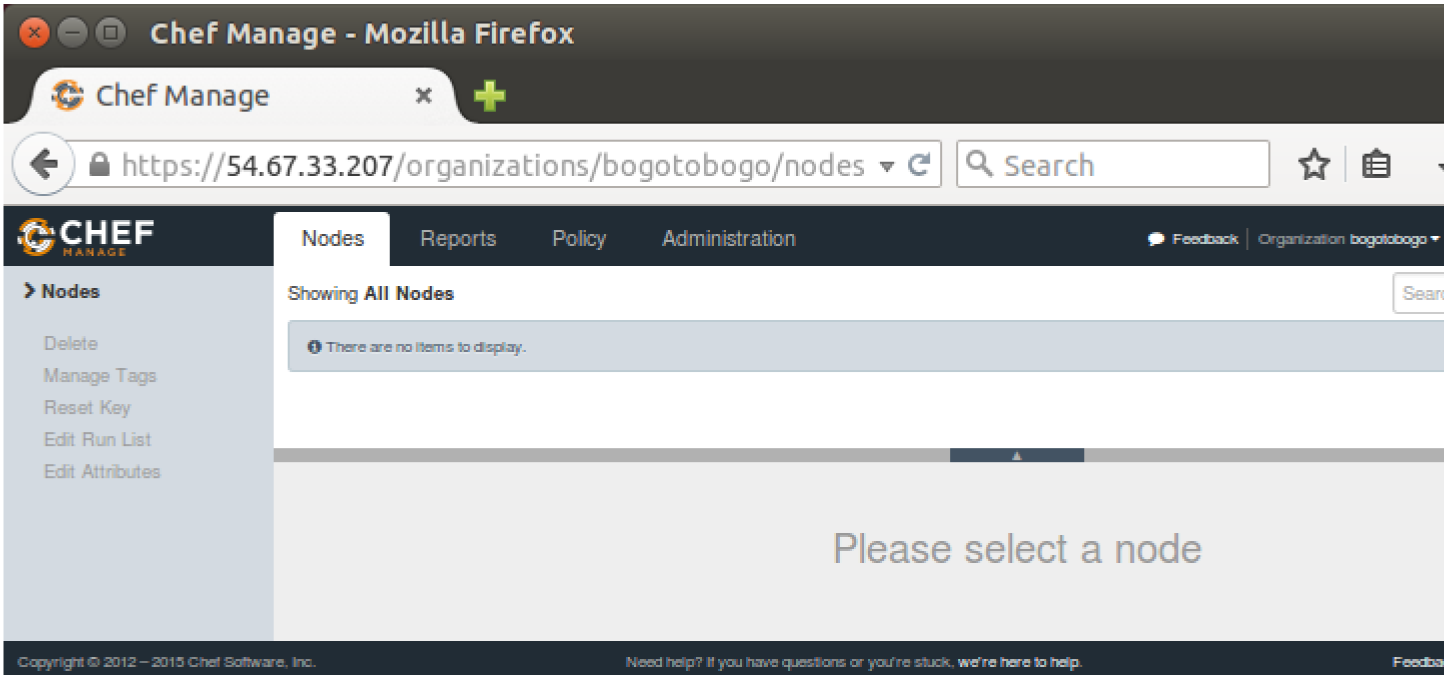
Click here to get started!

Copyright © 2012 – 2015 Chef Software, Inc.

Need help? If you have questions or you're stuck, **we are here to help**.

What's New?

After login with **admin** credentials we setup earlier, we get "Chef Manage" page:



There are no items to display.

Please select a node

Copyright © 2012 – 2015 Chef Software, Inc.

Need help? If you have questions or you're stuck, we're here to help.

Feedba

At this point, we do not have anything to display!

Other Chef plugins

There are other useful plugins:

1. **Opscode Push Jobs plugin**

```
2. $ sudo chef-server-ctl install opscode-push-jobs-server
3. $ sudo opscode-push-jobs-server-ctl reconfigure
```

```
$ sudo chef-server-ctl reconfigure
```

4. **Analytics plugin**

```
5. $ sudo chef-server-ctl install opscode-analytics
6. $ echo 'analytics_fqdn "FQDN"' | sudo tee -a /etc/opscode-analytics/opscode-analytics.rb
7. $ sudo opscode-analytics-ctl reconfigure
```

```
$ sudo chef-server-ctl reconfigure
```