

SQL Server Configuration using Opscode Chef Book

Rohan Shah, M.S.

Computer Engineering Department, College of Engineering,
San Jose State University, San Jose, CA 95112, United States
Email: rohan.shah@sjsu.edu

Abstract – This paper walks you through the steps required to configure SQL Server using Chef Book Technology developed by Opscode. The environment used is Linux Ubuntu 12.10 LTS and the Chef Book Version installed is 11.8.2.

I. INTRODUCTION

Chef Cook book provides a ready to use infrastructure code that can be used anywhere on the go. It is nothing but a complete infrastructure residing in a single piece of code. Thus, a completely virtual platform is created by Chef that emulates the virtual and physical servers with the help of a code. The primary purpose of using Chef is to help scale your infrastructure, provide security and manage its integrity safeguard your system. Also, portability is the best feature of Chef as the piece of infrastructure code is mobile and hence can be loaded into any desired system.

II. Steps Needed to Install Chef

Following are the steps required to install Chef on you Linux platform (Ubuntu environment used)

The tutorial assumes that you have the following things installed on your system:

- Virtual Box
- Git Setup to manage and sync code repository
- Vagrant setup for command line interface

1. Git Bash and Git

Download the git installer from <http://git-scm.com/> and accept the default options while installing it.

Open the git bash and write the following commands

```
$ git config --global user.name "<YOUR NAME HERE>"  
$ git config --global user.email "<YOUR EMAIL HERE>"
```

2. Installation of Ruby

Chef runs on Ruby and hence install ruby using the omnibus installer and type the following commands:

```
$ curl -L https://www.opscode.com/chef/install.sh | sudo bash
```

Also, by typing the following command, you may need type the ruby path again and again.

```
$ echo 'export PATH="/opt/chef/embedded/bin:$PATH"' >> ~/.bash_profile && source ~/.bash_profile
```

3. Installing Chef

Type the following commands and you will see the following output:

```
root@intro:~# cd ~  
root@intro:~# curl -L https://www.opscode.com/chef/install.sh | bash  
Thank you for installing Chef!
```

Also, check the version of the chef which is installed on your system by typing:

```
root@intro:~# chef-solo -v
```

III. INSTALLATION OF APACHE2

Since, we need to configure SQL and also check whether we have it running successful in our system, we will create a PHP Application that will help us verify whether we have installed and configured Apache and SQL correctly in our system.

1. Install the Chef Repository

Chef repository provides a directory like structure for organizing our codes and various chef files and cookbooks.

Type the following command to install the chef repository in you root folder.

```
root@intro:~# wget http://github.com/opscode/chef-repo/tarball/master  
root@intro:~# tar -zxf master  
root@intro:~# mv opscode-chef-repo* chef-repo  
root@intro:~# rm master
```

Check the contents of the chef repository using the following commands. You will see the following files:

```

root@intro:~# cd chef-repo/
root@intro:~/chef-repo# ls
certificates  cheffignore  config  cookbooks  data_bags  environments  LICENSE  Rakefile
README.md  roles

```

2. Cook Book Path for Knife

Knife is a tool that helps manage cookbooks in the chef directory.

We need to create a Chef directory and provide a path of the cookbook to it using the following commands:

```

root@intro:~/chef-repo# mkdir .chef
root@intro:~/chef-repo# echo "cookbook_path [ '/root/chef-repo/cookbooks' ]" > .chef/knife.rb

```

This tells knife where to find the cookbook directory and to install all the upcoming cookbooks in this directory.

3. Creation of the PHP App

The following command installs the php app directory with the help of cookbook path provided:

```

root@intro:~/chef-repo# knife cookbook create phpapp
** Creating cookbook phpapp
** Creating README for cookbook: phpapp
** Creating CHANGELOG for cookbook: phpapp
** Creating metadata for cookbook: phpapp

```

Just verify what knife tool has created for us in the directory by typing:

```

root@intro:~/chef-repo# cd cookbooks/phpapp
root@intro:~/chef-repo/cookbooks/phpapp# ls
attributes  CHANGELOG.md  definitions  files  libraries  metadata.rb  provider
recipes  resources  templates

```

Our goal now is to install and configure Apache, SQL and PHP. We will see this in the remaining part of the document. The cookbooks can also be downloaded directly from the Opscode website community.opscode.com/cookbooks but we will use the direct knife command line tool to make our lives easy.

4. Installing Apache cookbook and Apt

Type following command to install apache cookbook using knife

```

root@intro:~/chef-repo/cookbooks/phpapp# cd ..
root@intro:~/chef-repo/cookbooks# knife cookbook site download apache2

```

Also, untar the tar file and remove the tar file after untaring with the help of following command;

```

root@intro:~/chef-repo/cookbooks# tar xzf apache2*
root@intro:~/chef-repo/cookbooks# rm apache2*.tar.gz

```

Also, follow the same steps to install Apt

```

root@intro:~/chef-repo/cookbooks# knife cookbook site download apt
root@intro:~/chef-repo/cookbooks# tar xzf apt*
root@intro:~/chef-repo/cookbooks# rm apt*.tar.gz

```

5. Editing metadata.rb and default.rb Files

This document assumes that you have some kind of text editor installed in you system and are comfortable using it. I have Sublime Text 2 installed on my Ubuntu PC and hence I use the same. It is very easy to use and user friendly as well.

Go into your phpapp folder and add the last line to the metadata.rb file as follows using your favorite text editor.

```

name          'phpapp'
maintainer     'YOUR_COMPANY_NAME'
maintainer_email 'YOUR_EMAIL'
license        'All rights reserved'
description    'Installs/Configures phpapp'
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version        '0.1.0'

depends "apache2"

```

Also, add the last line as shown to the default.rb file in the recipes folder of phpapp.

```

#
# Cookbook Name:: phpapp
# Recipe:: default
#
# Copyright 2013, YOUR_COMPANY_NAME
#
# All rights reserved - Do Not Redistribute
#

include_recipe "apache2"

```

IV. TESTING OF OUR FIRST COOKBOOK

It includes running the Apache server with the help of following steps.

1. Creation of solo.rb file

Create a solo.rb file going back in your chef-repo directory and add the following lines to it.

```
file_cache_path "/root/chef-solo"
cookbook_path  "/root/chef-repo/cookbooks"
```

2. Creation of web.json file

Also, create a web.json file in the same chef-repo directory and add the following lines to it.

```
{
  "run_list": [ "recipe[apt]", "recipe[phpapp]" ]
}
```

Thus, with this we tell the chef to run the apt cookbook followed by our php cookbook.

3. Run the Chef Cookbook

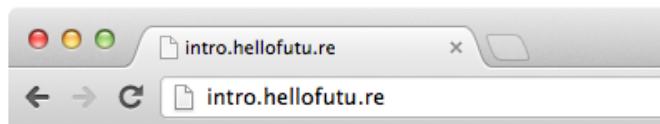
Now, if everything is set, run the chef cookbook by typing the following commands:

```
root@intro:~/chef-repo# chef-solo -c solo.rb -j web.json
Starting Chef Client, version 11.4.0
...
Chef Client finished, 14 resources updated
```

If you see the following message, as Chef Client finished, Congratulations!!! You have been successful in running your first cookbook and can visit the Apache server you just created.

4. Run the Apache Server

Run the apache server by typing on your web browser.



The web server is now running but you can see that no content is added yet. We will see the content really soon.

V. Configuration of SQL

It shows our main objective of successful installation of SQL server.

1. Installing Chef SQL Cook Book

Follow the same steps as before to install the sql cookbook available from the Opscode website by using the knife command line tool.

```
root@intro:~/chef-repo# cd cookbooks
root@intro:~/chef-repo/cookbooks# knife cookbook site download mysql
```

Also, untar and remove the tar files using the following commands;

```
root@intro:~/chef-repo/cookbooks# tar zxf mysql*
root@intro:~/chef-repo/cookbooks# rm mysql-*.tar.gz
```

2. Installing both MySQL Server as well as Client

Just verify the contents of mysql cookbook installed using the following commands. You should be able to see the following:

```
root@intro:~/chef-repo/cookbooks# cd mysql/recipes/
root@intro:~/chef-repo/cookbooks/mysql/recipes# ls
client.rb default.rb ruby.rb server_ec2.rb server.rb
```

This is a client server recipe and we need to include both of them as a dependency for our php app.

3. Editing of metadata.rb and default.rb files of php app

As we did before, we need to add the following lines to both the files of php app so that we can give the dependency of mysql to the phpapp. Type in the following lines at the end of the files.

```
name          'phpapp'
maintainer     'YOUR_COMPANY_NAME'
maintainer_email 'YOUR_EMAIL'
license        'All rights reserved'
description    'Installs/Configures phpapp'
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version        '0.1.0'

depends 'apache2'
depends 'mysql'
```

```
# Cookbook Name:: phpapp
# Recipe:: default
#
# Copyright 2013, YOUR_COMPANY_NAME
#
# All rights reserved - Do Not Redistribute
#

include_recipe "apache2"
include_recipe "mysql::client"
include_recipe "mysql::server"
```

VI. RUNNING CHEF THE SECOND TIME

It includes steps to run the sql cookbook

1. Run the Chef Cookbook

Just as done previously, run the step cookbook once more.

```
root@intro:~/chef-repo/cookbooks/phpapp# cd ../..
root@intro:~/chef-repo# chef-solo -c solo.rb -j web.json
Starting Chef Client, version 11.4.0
Compiling Cookbooks...
[2013-02-11T21:47:33+00:00] ERROR: Running exception handlers
[2013-02-11T21:47:33+00:00] ERROR: Exception handlers complete
Chef Client failed. 0 resources updated
[2013-02-11T21:47:33+00:00] FATAL: Stacktrace dumped to /root/chef-solo/chef-stacktrace.out
[2013-02-11T21:47:33+00:00] FATAL: Chef::Exceptions::CookbookNotFound: Cookbook build-essential not found. If
you're loading build-essential from another cookbook, make sure you configure the dependency in your metadata
a
```

If you see the above fata errors, don't woory you're on the right track. We have to specify the sql dependencies but we haven't done so.

2. Installing "buildessentail" and "openssl"

SQL depends on builddessesntail and openssl and thus we need to download and install them so that the metadata.rb file has their dependencies associated with them.

Type the following commands:

```
root@intro:~/chef-repo# cd cookbooks
root@intro:~/chef-repo/cookbooks# knife cookbook site download openssl
Downloading openssl from the cookbooks site at version 1.0.0 to /root/chef-repo/cookbooks/openssl-1.0.0.tar.g
z
Cookbook saved: /root/chef-repo/cookbooks/openssl-1.0.0.tar.gz
root@intro:~/chef-repo/cookbooks# tar zxvf openssl*.tar.gz
root@intro:~/chef-repo/cookbooks# rm openssl*.tar.gz
root@intro:~/chef-repo/cookbooks# knife cookbook site download build-essential
Downloading build-essential from the cookbooks site at version 1.3.4 to /root/chef-repo/cookbooks/build-essen
tial-1.3.4.tar.gz
Cookbook saved: /root/chef-repo/cookbooks/build-essential-1.3.4.tar.gz
root@intro:~/chef-repo/cookbooks# tar zxvf build-essential-*.tar.gz>
root@intro:~/chef-repo/cookbooks# rm build-essential-*.tar.gz
```

3. Re-run Chef

Now try to re-ru chef using the following commands:

```
root@intro:~/chef-repo/cookbooks# cd ..
root@intro:~/chef-repo# chef-solo -c solo.rb -j web.json
Starting Chef Client, version 11.4.0
Compiling Cookbooks...
[2013-02-12T18:44:18+00:00] WARN: Cloning resource attributes for service[apache2] from prior resource (CHEF-
3694)
[2013-02-12T18:44:18+00:00] WARN: Previous service[apache2]: /root/chef-repo/cookbooks/apache2/recipes/default
t.rb:24:in `from_file'
[2013-02-12T18:44:18+00:00] WARN: Current service[apache2]: /root/chef-repo/cookbooks/apache2/recipes/default
t.rb:221:in `from_file'
[2013-02-12T18:44:18+00:00] FATAL: You must set node['mysql']['server_debian_password'], node['mysql']['serve
r_root_password'], node['mysql']['server_rep_password'] in chef-solo mode. For more information, see https://
github.com/opscode-cookbooks/mysql#chef-solo-note
```

You might see some errors again but still you are on the right track.

4. Define root password for MySQL

It is an attribute of Chef to define a root password for its correct execution. So you need to edit the web.json file again and add the following lines to it.

```
{
  "mysql": { "server_root_password": "808052769e2c6d909027a2905b224bad", "server_debian_password": "569d1ed2d4
6870cc020fa87be83af98d", "server_rep_password": "476911180ee92a2ee5a471f33340f6f4"},
  "run_list": [ "recipe[apt]", "recipe[phpapp]" ]
}
```

5. Run Chef-Solo Again

Run the chef solo again and you will see the following message on the console:

```
root@intro:~/chef-repo# chef-solo -c solo.rb -j web.json
Starting Chef Client, version 11.4.0
...
Chef Client finished, 14 resources updated
```

It's time to rejoice ! We just installed our SQL Server.

VII.INSTALLING PHP CHEF COOKBOOK

Follow exactly the same commands as follows and as done before edit the metadata.rb and default.rb files of phpapp to install php cookbook into your system.

```
root@intro:~/chef-repo# cd cookbooks/
root@intro:~/chef-repo/cookbooks# knife cookbook site download php
root@intro:~/chef-repo/cookbooks# tar zxf php*.tar.gz
root@intro:~/chef-repo/cookbooks# rm php*.tar.gz
root@intro:~/chef-repo/cookbooks# knife cookbook site download xml
root@intro:~/chef-repo/cookbooks# tar zxf xml-*.tar.gz
root@intro:~/chef-repo/cookbooks# rm xml-*.tar.gz
```

```
name          'phpapp'
maintainer     'YOUR_COMPANY_NAME'
maintainer_email 'YOUR_EMAIL'
license        'All rights reserved'
description    'Installs/Configures phpapp'
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version        '0.1.0'

depends "apache2"
depends "mysql"
depends "php"
```

```
# Cookbook Name:: phpapp
# Recipe:: default
#
# Copyright 2013, YOUR_COMPANY_NAME
#
# All rights reserved - Do Not Redistribute
#

include_recipe "apache2"
include_recipe "mysql::client"
include_recipe "mysql::server"
include_recipe "php"
include_recipe "php::module_mysql"
include_recipe "apache2::mod_php5"

apache_site "default" do
  enable true
end
```

2. Run the Chef Solo

```
root@intro:~/chef-repo/cookbooks/phpapp# cd ../../
root@intro:~/chef-repo# chef-solo -c solo.rb -j web.json
Starting Chef Client, version 11.4.0
...
Chef Client finished, 8 resources updated
```

VIII. FINAL TESTING OF APACHE, SQL ANF PHP

1. Create a test.php file

Create a test file in your /var/www/php folder and add the following lines in your test.php file

```
<?php phpinfo(); ?>
```


2. You should see the following screen.

Browse from start to end and find if everything is working fine.

phpinfo() - Mozilla Firefox

localhost/test.php

PHP Version 5.3.10-1ubuntu3.8



System	Linux ubuntu 3.8.0-29-generic #42~precise1-Ubuntu SMP Wed Aug 14 16:19:23 UTC 2013 x86_64
Build Date	Sep 4 2013 19:44:20
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,NTS
PHP Extension Build	API20090626,NTS
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled

phpinfo() - Mozilla Firefox

localhost/test.php

Apache Environment

Variable	Value
HTTP_HOST	localhost
HTTP_USER_AGENT	Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:23.0) Gecko/20100101 Firefox/23.0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_CONNECTION	keep-alive
PATH	/usr/local/bin:/usr/bin:/bin
SERVER_SIGNATURE	<address>Apache Server at localhost Port 80</address>
SERVER_SOFTWARE	Apache
SERVER_NAME	localhost
SERVER_ADDR	127.0.0.1
SERVER_PORT	80
REMOTE_ADDR	127.0.0.1
DOCUMENT_ROOT	/var/www/
SERVER_ADMIN	ops@example.com
SCRIPT_FILENAME	/var/www/test.php
REMOTE_PORT	52701
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	no value
REQUEST_URI	/test.php

phpinfo() - Mozilla Firefox

localhost/test.php

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.5.34
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/run/mysqld/mysqld.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib/x86_64-linux-gnu -mysqlclient_r

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	no value	no value
mysql.default_password	no value	no value
mysql.default_port	no value	no value
mysql.default_socket	/var/run/mysqld/mysqld.sock	/var/run/mysqld/mysqld.sock
mysql.default_user	no value	no value
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

mysqli

CONCLUSION

Thus, we are successful in installing and configuring SQL with the help of Opscode Chef Book Technology. Also, we are able to test and verify the working condition of Apache, SQL and PHP with the help of test.php file created.

ACKNOWLEDGEMENT

I would like to deeply thank professor Rakesh Ranjan for allowing me to take an extra assignment and assigning me something of my choice. It was fun to learn the completely new concept of Chef and will help me in future endeavours.

APPENDIX

```

root@ubuntu: ~/chef-repo/cookbooks
root@ubuntu:~/chef-repo/cookbooks# pwd
/root/chef-repo/cookbooks
root@ubuntu:~/chef-repo/cookbooks# ls
apache2  apt  build-essential  mysql  openssl  php  phpapp  README.md  xml
root@ubuntu:~/chef-repo/cookbooks#

```

Proof of Concept showing Installed Cookbook in chef-repo directory