1. Create cookbook using knife

Ansible: [https://www.udemy.com/ansible-the-complete-automation-han…/…](https://www.udemy.com/ansible-the-complete-automation-hands-on-guide-devops-beginners/?couponCode=JGAFB0)  
Docker: [https://www.udemy.com/devops-docker-complete-guide-hands-…/…](https://www.udemy.com/devops-docker-complete-guide-hands-on-with-practical/?couponCode=JGAFB0)  
AWS:[https://www.udemy.com/aws-certified-solution-architect-as…/…](https://www.udemy.com/aws-certified-solution-architect-associate-2017/?couponCode=FACEBOOKJOY2)  
Ethereum: [https://www.udemy.com/ethereum-cryptocurrency-blockchain-…/…](https://www.udemy.com/ethereum-cryptocurrency-blockchain-solidity-ethereum-bitcoin-crypto/?couponCode=FACEBOOKJOY3)  
Steem: [https://www.udemy.com/steemit-blogging-social-media-that-…/…](https://www.udemy.com/steemit-blogging-social-media-that-pays-work-from-home/?couponCode=FBJOY0)

HTML5: <https://www.udemy.com/html-5-hands-on>  
Chess: [https://www.udemy.com/learn-chess-opening-caro-kann-defen…/…](https://www.udemy.com/learn-chess-opening-caro-kann-defense-a-to-z-concept/?couponCode=FACEBOOKJOY2)  
Python: [https://www.udemy.com/python-programming-mini-boot-camp/…](https://www.udemy.com/python-programming-mini-boot-camp/?couponCode=FACEBOOKJOY)  
Chess: [https://www.udemy.com/play-chess-the-chess-boot-camp-step…/…](https://www.udemy.com/play-chess-the-chess-boot-camp-step-by-step-for-beginner/?couponCode=SANGAM)  
Scala: [https://www.udemy.com/scala-programming-complete-mini-boot-…](https://www.udemy.com/scala-programming-complete-mini-boot-camp-beginner-to-advance)  
CCNA: [https://www.udemy.com/ccna-bootcamp-from-beginner-to-adva…/…](https://www.udemy.com/ccna-bootcamp-from-beginner-to-advance/?couponCode=FACEBOOKJOY2)  
SCALA: [https://www.udemy.com/scala\_scalable\_functional\_programmi…/…](https://www.udemy.com/scala_scalable_functional_programming_language_kids_beginners_scratch/?couponCode=REALDISCOUNT)  
Java: [https://www.udemy.com/java-programming-for-complete-beginne…](https://www.udemy.com/java-programming-for-complete-beginners-to-advance)  
PHP: [https://www.udemy.com/php-for-beginners-to-advance-comple…/…](https://l.facebook.com/l.php?u=https%3A%2F%2Fwww.udemy.com%2Fphp-for-beginners-to-advance-complete-course%2F%3FcouponCode%3DFACEBOOKJOY&h=ATOkIlh-Y3MYQ-zuyq25Ugsj2ou9SrimDXq0EM0Rzh0k-vTJ3CqqPYivGF7oeguSOhvGAZe1-LgppMWCNWvHJKyiEo6c-NfyofJy_G0F6GNVN9xoMOz8zGhzWYcoYL_Y0v0CuwhATWZz05aKrTKlNN4oBQYFs1ugGP7zshXoz8ohjygJCg4eFz-Bc7TIlKA4wjLILyAU-pHzdeTIO7-oX76YVG0CLDlwugby-fuJtJCgOlLyzv108-XTgRZG7516EzLGPMBtzPgpTnCrTO6FI9URdrfWxllspVcCm3gv0tCYFmA)  
Python: [https://www.udemy.com/python-django-programming-beginner-…/…](https://www.udemy.com/python-django-programming-beginner-to-advance-tutorial-step-by-step/?couponCode=FACEBOOKJOY2)  
Ebay Dropshipping: [https://www.udemy.com/top-ebay-dropshiping-course/…](https://www.udemy.com/top-ebay-dropshiping-course/?couponCode=FACEBOOKJOY2)  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Code :

Git

Subversion

JIRA

Eclipse

# Build :

Maven

Ant

Gradle

# Test :

Sellinium

Junit

# Release :

Jenkins

Bamboo

# Monitering tools:

Nagios,

Splunck

Sensu

New relic

# Deploy tools :

Puppet

Ansible

Chef

Saltstack

# Cmd : Knife cookbook create “test”

chef generate cookbook

[vagrant@checflab chef-repo]$ ls -la cookbooks/testcb/

total 52

drwxrwxr-x 10 vagrant vagrant 4096 Jul 7 02:44 .

drwxr-xr-x 4 vagrant vagrant 4096 Jul 7 02:44 ..

drwxrwxr-x 2 vagrant vagrant 4096 Jul 7 02:44 attributes

-rw-rw-r-- 1 vagrant vagrant 433 Jul 7 02:44 CHANGELOG.md

drwxrwxr-x 2 vagrant vagrant 4096 Jul 7 02:44 definitions

drwxrwxr-x 3 vagrant vagrant 4096 Jul 7 02:44 files

drwxrwxr-x 2 vagrant vagrant 4096 Jul 7 02:44 libraries

-rw-rw-r-- 1 vagrant vagrant 276 Jul 7 02:44 metadata.rb

drwxrwxr-x 2 vagrant vagrant 4096 Jul 7 02:44 providers

-rw-rw-r-- 1 vagrant vagrant 1455 Jul 7 02:44 README.md

drwxrwxr-x 2 vagrant vagrant 4096 Jul 7 02:44 recipes

drwxrwxr-x 2 vagrant vagrant 4096 Jul 7 02:44 resources

drwxrwxr-x 3 vagrant vagrant 4096 Jul 7 02:44 templates

## Cmd : tree

If tree is not installed do sudo yum install tree -y

Tree testcb/

[vagrant@checflab cookbooks]$ tree

.

|-- chefignore

|-- starter

| |-- attributes

| | `-- default.rb

| |-- files

| | `-- default

| | `-- sample.txt

| |-- metadata.rb

| |-- recipes

| | `-- default.rb

| `-- templates

| `-- default

| `-- sample.erb

`-- testcb

|-- attributes

|-- CHANGELOG.md

|-- definitions

|-- files

| `-- default

|-- libraries

|-- metadata.rb

|-- providers

|-- README.md

|-- recipes

| `-- default.rb

|-- resources

`-- templates

`-- default

18 directories, 10 files

Cp <.rb > file to default.rb

# Upload cookbook:

Step1: create directory in cookbook

Cmd : chef generate cookbook testcb

Step2 : go to testcb and update default.rb file

Step3 : from cookbooks directory issue cmd

Cmd : knife upload cookbook testcb

To know the cook book list

Cmd : knife cookbook list

# bootstrap:

knife bootstrap 192.168.33.50 -x vagrant -P vagrant -N jenkins --sudo

# Roles:

# databags:

# packages:

# Attributes:

# servers packages:

packages : httpd 🡪 centos

apache2 🡪 Ubuntu

lab : copy rpm to files dir

sudo chef-client

docker commit <old image> <new imaga name (with all the files of old image)>

docker rm <container id>

docker stop <container id>

code commit

pipeline

deploy

ops works

# Issues :

**echo** "source /opt/rh/rh-ruby22/enable" to fix this add file ruby22.sh in profile.d and add path /opt/rh/rh-ruby22/enable in it.

# Jenkins installation :

### **Port check :**

/etc/init.d 🡪 all installation service can see here

/var/log 🡪logs will store here

Config global security :

LDAP :

Master

Slaves

Create job :

### **Step 1: Update your CentOS 7 system**

One of the Linux system administrator's best practices is keeping a system up to date. Install the latest stable packages, then reboot.

sudo yum install epel-release

sudo yum update

sudo reboot

When the reboot finishes, login with the same sudo user.

### **Step 2: Install Java**

Before you can install Jenkins, you need to setup a Java virtual machine on your system. Here, let's install the latest OpenJDK Runtime Environment 1.8.0 using YUM:

sudo yum install java-1.8.0-openjdk.x86\_64

After the installation, you can confirm it by running the following command:

java -version

This command will tell you about the Java runtime environment that you have installed:

openjdk version "1.8.0\_91"

OpenJDK Runtime Environment (build 1.8.0\_91-b14)

OpenJDK 64-Bit Server VM (build 25.91-b14, mixed mode)

In order to help Java-based applications locate the Java virtual machine properly, you need to set two environment variables: "JAVA\_HOME" and "JRE\_HOME".

sudo cp /etc/profile /etc/profile\_backup

echo 'export JAVA\_HOME=/usr/lib/jvm/jre-1.8.0-openjdk' | sudo tee -a /etc/profile

echo 'export JRE\_HOME=/usr/lib/jvm/jre' | sudo tee -a /etc/profile

source /etc/profile

Finally, you can print them for review:

echo $JAVA\_HOME

echo $JRE\_HOME

### **Step 3: Install Jenkins**

Use the official YUM repo to install the latest stable version of Jenkins, which is 1.651.2 at the time of writing:

cd ~

sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo

sudo rpm --import http://pkg.jenkins-ci.org/redhat-stable/jenkins-ci.org.key

sudo yum install jenkins

Start the Jenkins service and set it to run at boot time:

sudo systemctl start jenkins.service

sudo systemctl enable jenkins.service

In order to allow visitors access to Jenkins, you need to allow inbound traffic on port 8080:

sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp

sudo firewall-cmd --reload

Now, test Jenkins by visiting the following address from your web browser:

http://<your-Vultr-server-IP>:8080

### Jenkins slave :

### SSH connect remote machine :

If your username is different on the remote system, you can specify it by using this syntax:

* ssh remote\_username@remote\_host

Once you have connected to the server, you will probably be asked to verify your identity by providing a password.

Later, we will cover how to generate keys to use instead of passwords.

### How Does Key-based Authentication Work?

Key-based authentication works by creating a pair of keys: a private key and a public key.

The private key is located on the client machine and is secured and kept secret.

The public key can be given to anyone or placed on any server you wish to access.

When you attempt to connect using a key-pair, the server will use the public key to create a message for the client computer that can only be read with the private key.

The client computer then sends the appropriate response back to the server and the server will know that the client is legitimate.

This entire process is done in the background automatically after you set up keys.

### How To Create SSH Keys

SSH keys should be generated on the computer you wish to log in from. This is usually your local computer.

Enter the following into the command line:

* ssh-keygen -t rsa

Press enter to accept the defaults. Your keys will be created at ~/.ssh/id\_rsa.pub and ~/.ssh/id\_rsa.

Change into the .ssh directory by typing:

* cd ~/.ssh

Look at the permissions of the files:

* ls -l

Output

-rw-r--r-- 1 demo demo 807 Sep 9 22:15 authorized\_keys

-rw------- 1 demo demo 1679 Sep 9 23:13 id\_rsa

-rw-r--r-- 1 demo demo 396 Sep 9 23:13 id\_rsa.pub

As you can see, the id\_rsa file is readable and writable only to the owner. This is how it should be to keep it secret.

The id\_rsa.pub file, however, can be shared and has permissions appropriate for this activity.

### How To Transfer Your Public Key to the Server

You can copy the public key to the remote server by issuing this command:

* ssh-copy-id remote\_host

This will start an SSH session, which you will need to authenticate with your password.

After you enter your password, it will copy your public key to the server's authorized keys file, which will allow you to log in without the password next time.

## Client-Side Options

There are a number of optional flags that you can select when connecting through SSH.

Some of these may be necessary to match the settings in the remote host's sshd configuration.

For instance, you if you changed the port number in your sshd configuration, you will need to match that port on the client-side by typing:

* ssh -p port\_number remote\_host

If you only wish to execute a single command on a remote system, you can specify it after the host like so:

* ssh remote\_host command\_to\_run

You will connect to the remote machine, authenticate, and the command will be executed.

As we said before, if X11 forwarding is enabled on both computers, you can access that functionality by typing:

* ssh -X remote\_host

Providing you have the appropriate tools on your computer, GUI programs that you use on the remote system will now open their window on your local system.

# How to Disable Password Authentication for SSH

Once you have [**SSH Keys configured**](http://support.hostgator.com/articles/ssh-keys), you can add some extra security to your server by disabling password authentication for SSH. (Note that if you do lose your private key, this will make the server inaccessible and you will need to contact HostGator to have this re-enabled.)

To disable this setting, you can do the following:

nano /etc/ssh/sshd\_config

In this file, set the following settings to the following values. If these settings are already in the file, set them to "no" rather than add new lines.

ChallengeResponseAuthentication no  
PasswordAuthentication no  
UsePAM no

Once this is done, restart the SSH daemon to apply the settings.

/etc/init.d/sshd restart

# docker installation :

# docker issues :

Error response from daemon: Get https://index.docker.io/v1/search?q=httpd&n=25: dial tcp: lookup index.docker.io on 10.0.2.3:53: read udp 10.0.2.15:52250->10.0.2.3:53: i/o timeout

Solution : vi /etc/resolv.conf

change nameserver to 8.8.8.8

Create repo in /etc/repo.d/docker.repo

Install docker

Service docker start

Docker pull images

Docker ps

Docker ps –a

Docker run –it image id /bin/bash bash

docker commit c3f279d17e0a(container id ) svendowideit/testimage:version3

**How to mount volumes or transfer files to container**

Cmd : docker run –it –name myimage -p 1234:80 -v ${PWD}/.:/home/test\_copy image

**How to go back to container once we come out :**

Docker ps

Docker attach <names from docker ps /container name>

Note: This will take you back to container where exactly you left

Check t webserver is working or not :

**Install apache**

If u get error below :

root@e2ee9a596dc7:/home/test\_copy\_data# service apache2 start

\* Starting Apache httpd web server apache2 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message

Then

root@e2ee9a596dc7:/etc/apache2# tree

.

|-- apache2.conf <At the end of this file add ServerName localhost>

Come back to browser type ipaddress:port

**How to start container :**

Docker start <name from docker ps –a names>

Docker exec -it container name (to get interactive mode)