Problem Statement

This project will be about how to do deploy code to dev/stage/prod etc, just on a click of button.

Link for the sample PHP application: <https://github.com/edureka-devops/projCert.git>

Business challenge/requirement

As soon as the developer pushes the updated code on the GIT master branch, a new test server should be provisioned with all the required software. Post this, the code should be containerized and deployed on the test server.

The deployment should then be built and pushed to the prod server.

All this should happen automatically and should be triggered from a push to the GitHub master branch.

Steps for executing the solution:

•Use the Master VM for Jenkins, Ansible, GIT etc.

•Use the fresh instance for Jenkins Slave Node (Test Server)

•Change the IP address of the VMs accordingly

•Add Build Pipeline Plugin and Post-build task plugin to Jenkins on the master VM

•Install python, openssh-server and git on the slave node manually

•Use the image devopsedu/webapp and add your PHP website to it using a Dockerfile

•Push the PHP website, and the Dockerfile to a git repository

Below tasks should be automated through Jenkins by creating a pipeline:

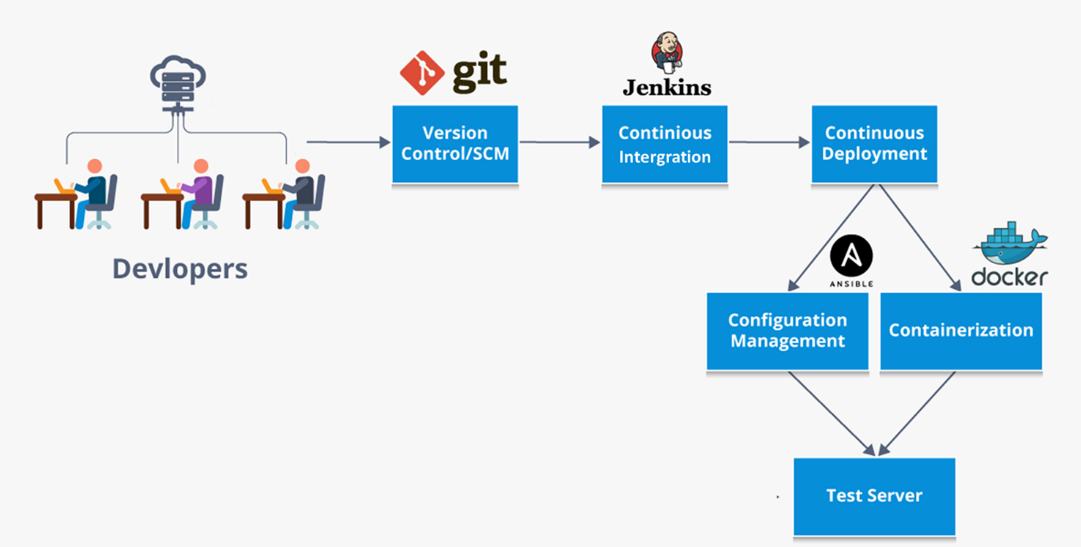
1.Install and configure puppet agent on the slave node (Job 1)

2.Push an Ansible configuration on test server to install docker (Job 2)

3.Pull the PHP website, and the Dockerfile from the git repo and build and deploy your PHP

docker container. After. (Job 3)

4.If Job 3 fails, delete the running container on Test Server.



Solution:

Use the Master VM for Jenkins, Ansible, GIT etc.

*Create a centos server and install Jenkins.*

sudo yum install epel-release java-11-openjdk-devel

sudo yum -y install wget

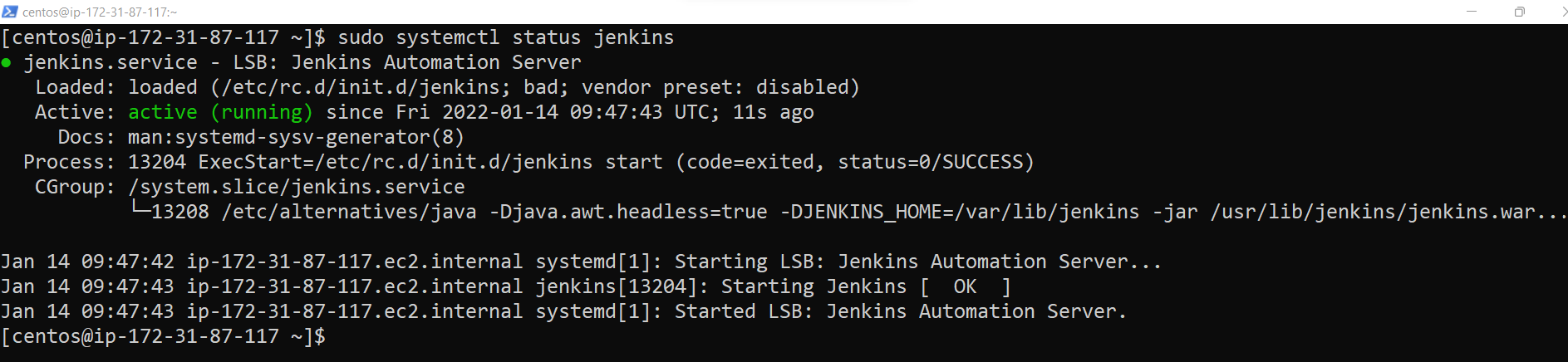
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo --no-check-certificate

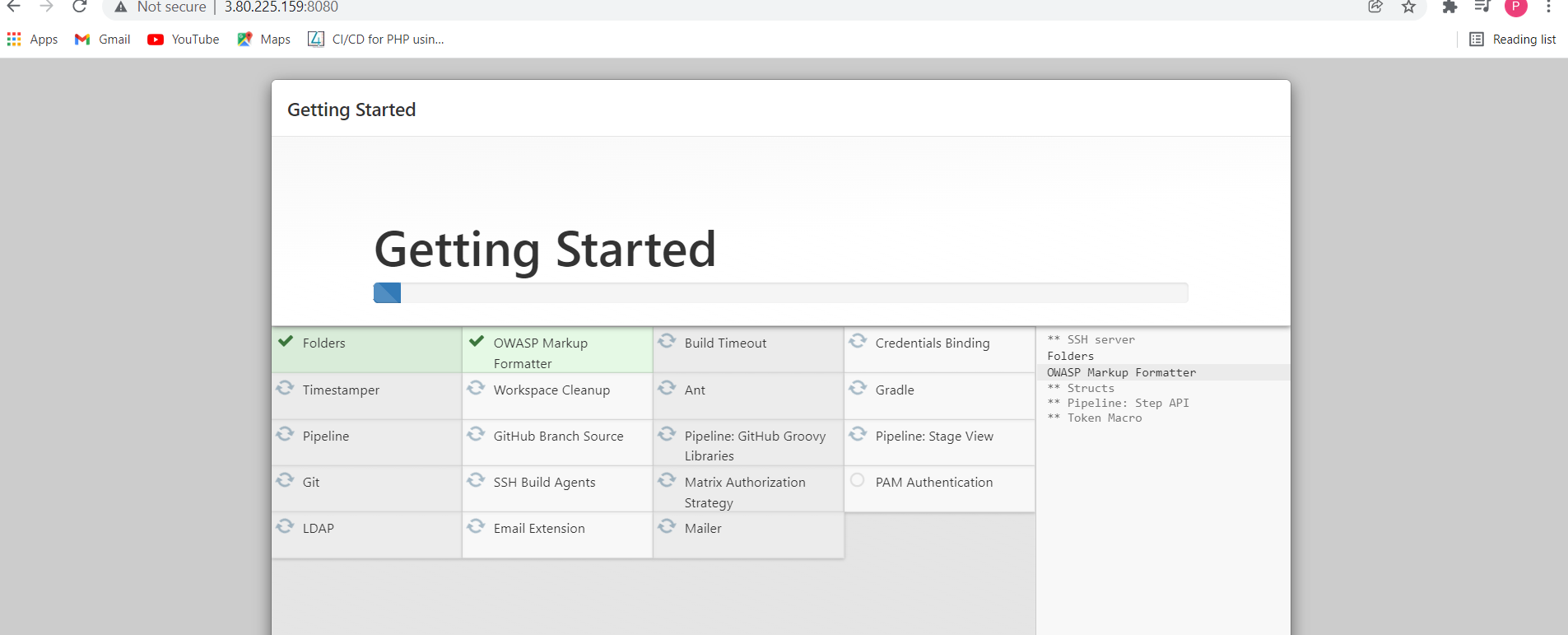
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

sudo yum upgrade

sudo yum install jenkins

sudo systemctl start Jenkins





*Install Git:*

sudo yum -y install git



*Install Ansible:*

sudo yum -y install epel-release

sudo yum -y update

sudo yum -y install ansible

Use the fresh instance for Jenkins Slave Node (Test Server)

RUN an ec2 CentOS instance (using AWS console) and got public Ip as 3.84.43.63

Install python, openssh-server and git on the slave node manually

sudo yum install -y python3

sudo yum -y install git

sudo yum –y install openssh-server openssh-clients

sudo systemctl start sshd

sudo systemctl enable sshd

sudo systemctl status sshd

Install php on the slave node manually

|  |
| --- |
|  |
|  |
| 1. To install PHP 7, you have to install and enable EPEL and Remi repository on your CentOS 7 system with the commands below.  # yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  # yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm  2. Next, you need to install yum-utils, a collection of useful programs for managing yum repositories and packages. It has tools that basically extend yum’s default features.  It can be used for managing (enabling or disabling) yum repositories as well as packages without any manual configuration and so much more.  # yum install yum-utils  3. One of the programs provided by yum-utils is yum-config-manager, which you can use to enable Remi repository as the default repository for installing different PHP versions as shown.  # yum-config-manager --enable remi-php70 [Install PHP 7.0]  4. Now install PHP 7 with all necessary modules with the command below.  # yum install php php-mcrypt php-cli php-gd php-curl php-mysql php-ldap php-zip php-fileinfo  5. Afterwards, double check the installed version of PHP on your system.  # php -v |

Use the image devopsedu/webapp and add your PHP website to it using a Dockerfile

Give Jenkins access as sudo user:

sudo usermod -a -G docker Jenkins

sudo reboot

git clone <https://github.com/lvchiney/projCert.git>

cd projCert/

create a Docker file here.

*Dockerfile:*

FROM devopsedu/webapp

COPY . /usr/src/myapp

*To build the image:*

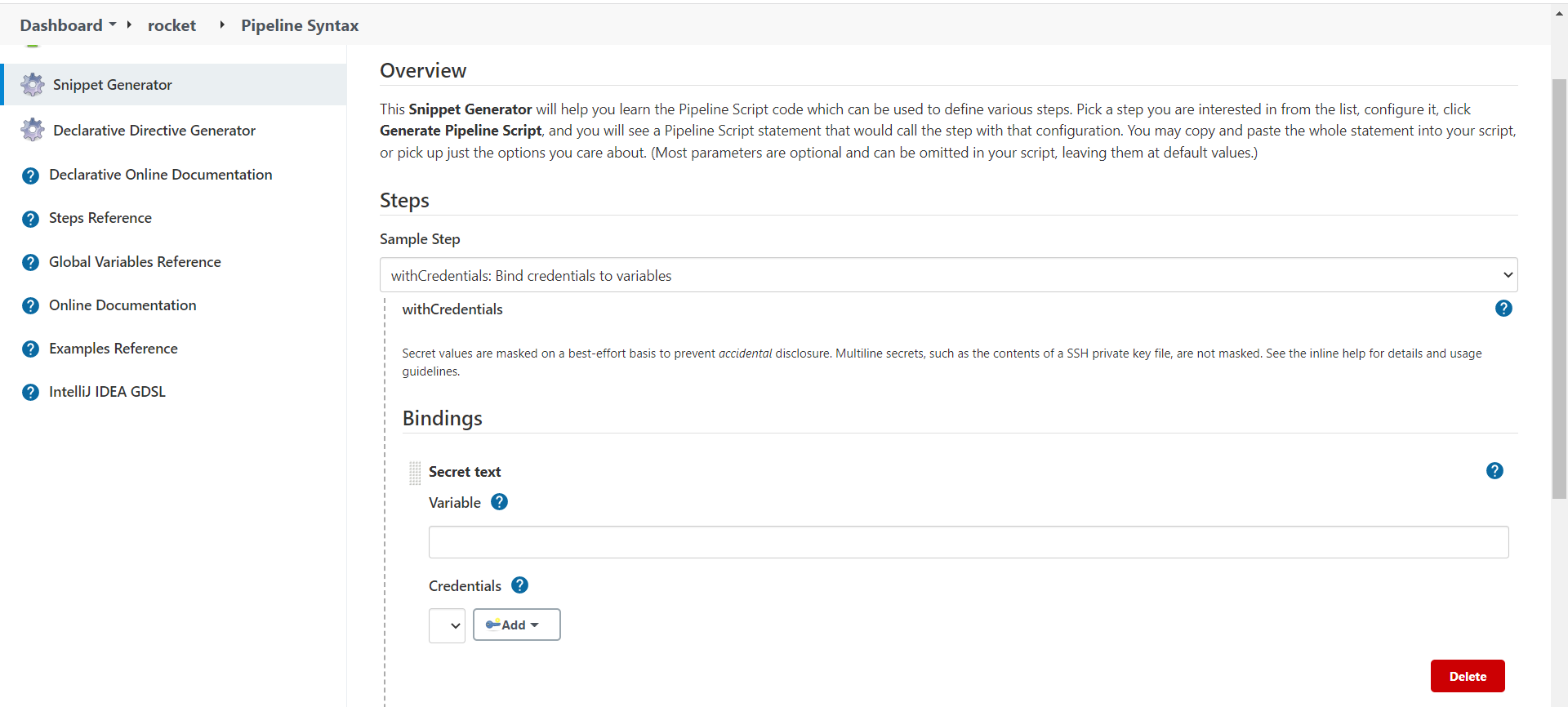
docker build -t lvchiney/projcert .

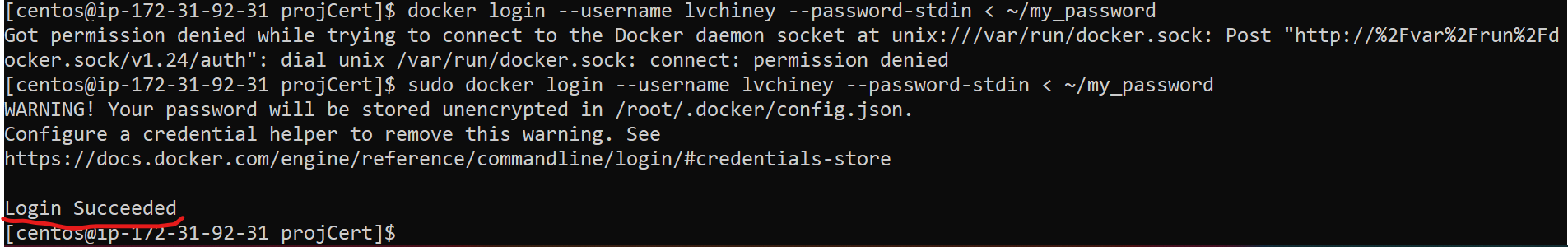
*•Push the image to* [*Docker Hub*](https://hub.docker.com/)

Keep your password in the my\_password file.

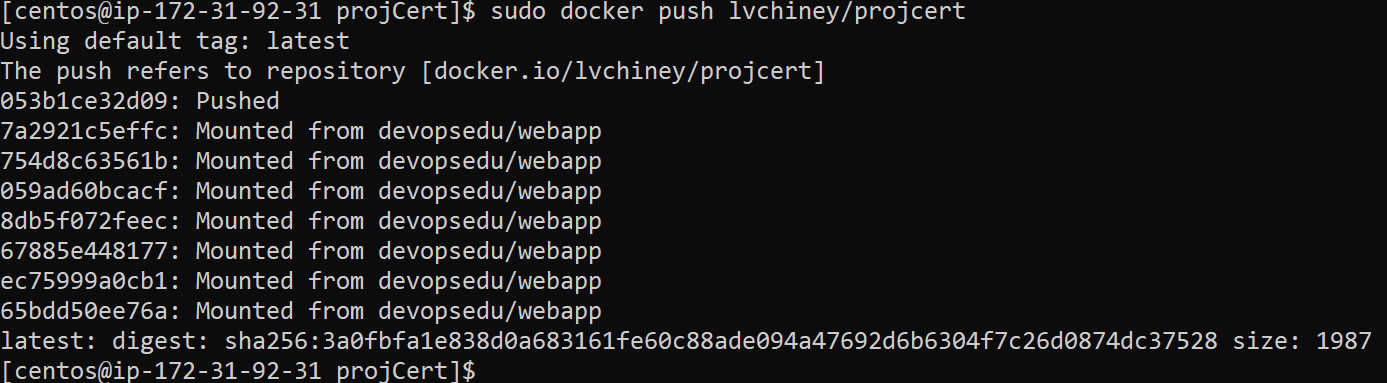
docker login --username lvchiney --password-stdin < ~/my\_password

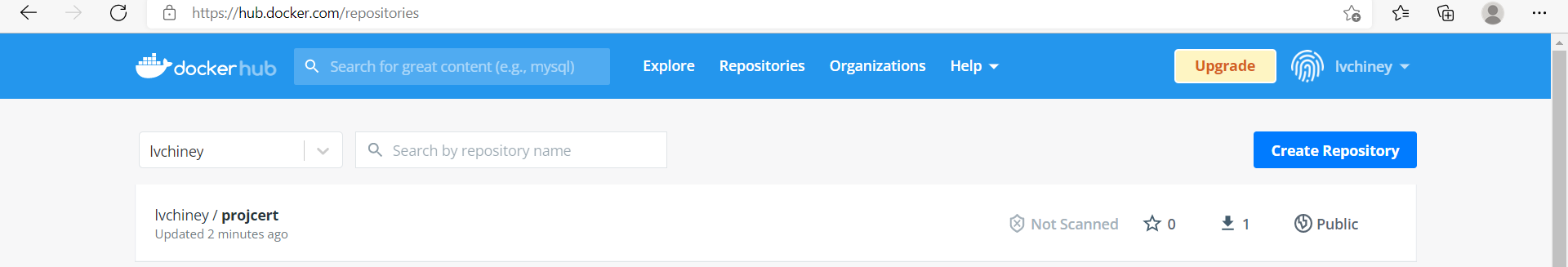
or use below steps in Jenkins to add password.





sudo docker push lvchiney/projcert





Push an Ansible configuration on test server to install docker

Copy the .pem file in main server for our system:

scp -i .\AWSCHINEY.pem .\AWSCHINEY.pem centos@18.206.120.207:~

Change the host file in server1 location:

sudo vi /etc/ansible/hosts

[Web]

W1 ansible\_host=3.84.43.63 ansible\_user=centos ansible\_ssh\_private\_key\_file=/home/centos/AWSCHINEY.pem

#docker-configuration.yml

- name: Update web servers

hosts: W1

become: true

tasks:

- name: Create yum repository for docker

yum\_repository:

name: docker-repo

description: "Repo for docker"

baseurl: "https://download.docker.com/linux/centos/7/x86\_64/stable/"

enabled: yes

gpgcheck: no

- name: Install Docker

package:

name: "docker-ce-18.09.1-3.el7.x86\_64"

state: present

- name: "Start docker service"

service:

name: "docker"

state: started

ansible-playbook docker-configure.yml

Install Docker-compose on test server

1. Run this command to download the current stable release of Docker Compose:

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

2. Apply executable permissions to the binary:

sudo chmod +x /usr/local/bin/docker-compose

Create Docker-compose.yml

version: "3.9" # optional since v1.27.0

services:

httpd:

image: lvchiney/projcert

ports:

- "7777:80"

Run:

docker-compose up -d

Final Output:

