**CI/CD Deployment Using Ansible CM Tool**

**Problem:-**

DESCRIPTION

You are a DevOps engineer at XYZ Ltd. Your company is working on a Java application and wants to automate WAR file artifact deployment so that they don’t have to perform WAR deployment on Tomcat/Jetty web containers. Automate Ansible integration with Jenkins CI server so that we can run and execute playbooks to deploy custom WAR files to a web container and then perform restart for the web container.

**Steps to Perform:**

1. Configure Jenkins server as Ansible provisioning machine
2. Install Ansible plugins in Jenkins CI server
3. Prepare Ansible playbook to run Maven build on Jenkins CI server
4. Prepare Ansible playbook to execute deployment steps on the remote web container with restart of the web container post deployment

**Solution:-**

Pre-requisite:-

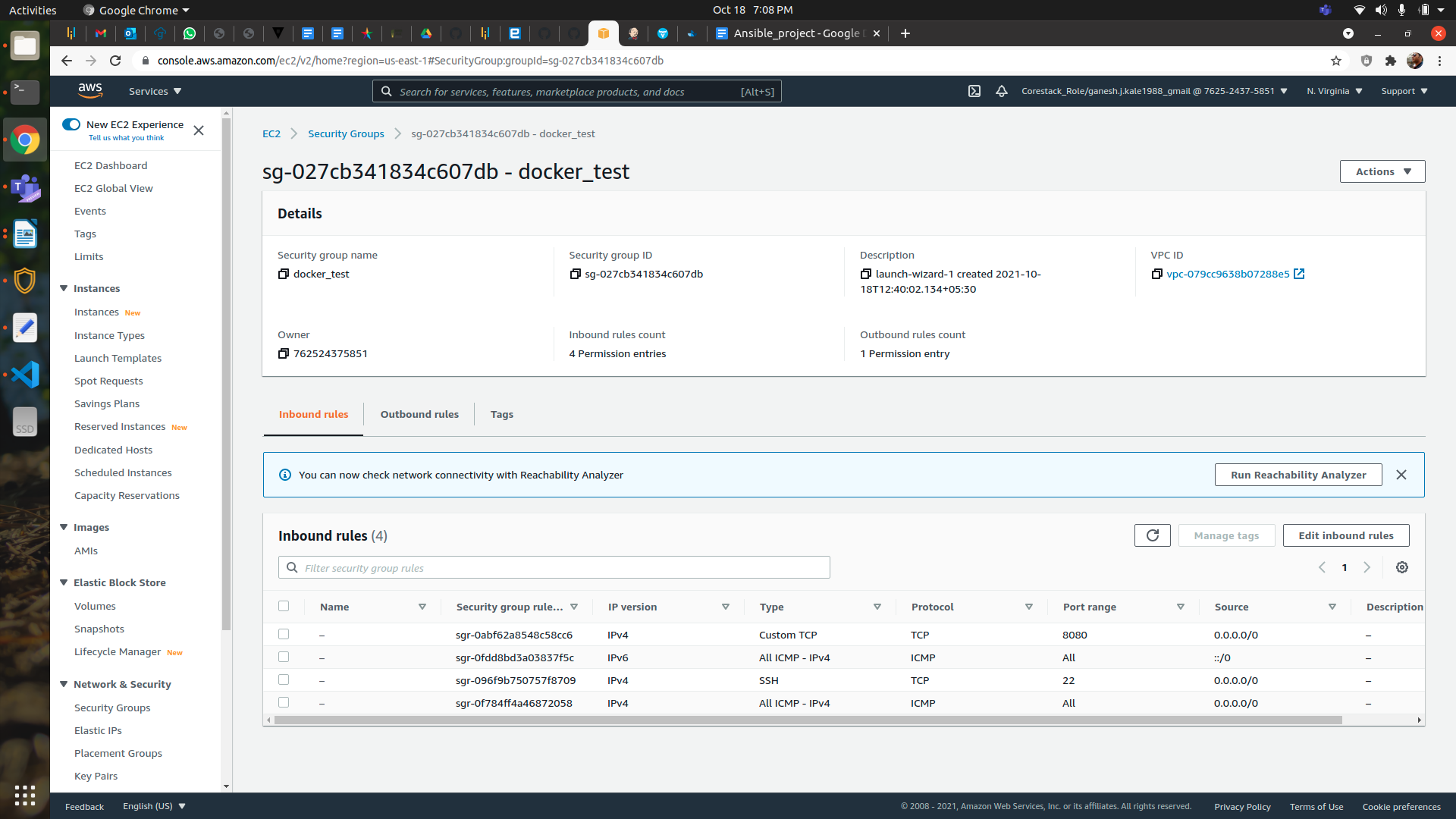
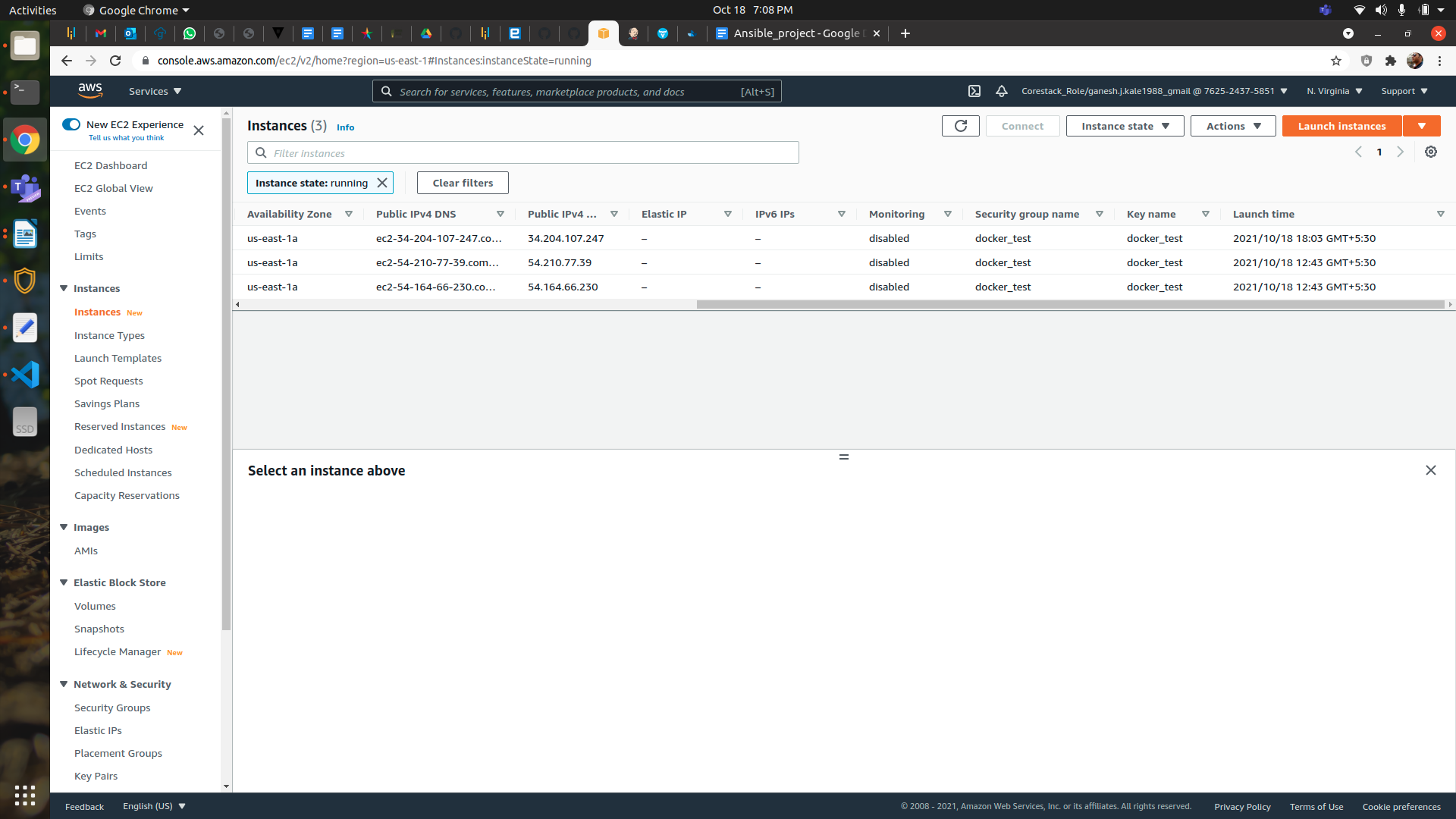
1:-Create 2 vm on AWS (Amazon Linux 2 AMI (HVM), SSD Volume Type )

2:-one vm for jenkins running in docker

3:-second vm for maven build and ansible controller

**AWS vm creation steps:-**

while creating ec2 instance we need to create ssh keypair to access newly created vm

****

**Steps:-**

1:-Run jenkins in docker container in first vm

sudo apt update

sudo apt install docker\*

sudo usermod -aG docker ec2-user

sudo service docker start

newgrp docker

docker pull ganeshkale/jenkins\_docker:1.0

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

sudo chmod 777 /usr/bin/docker-compose

mkdir jenkins

cd jenkins

vi docker-compose.yaml

version: '3.3'

services:

jenkins:

image: ganeshkale/jenkins\_docker:1.0

ports:

- '8080:8080'

- '50000:50000'

networks:

- jenkins

volumes:

- jenkins-data:/var/jenkins\_home

- jenkins-docker-certs:/certs/client:ro

restart: always

networks:

jenkins:

volumes:

jenkins-data:

external: true

jenkins-docker-certs:

external: true

~

docker-compose up -d

docker ps

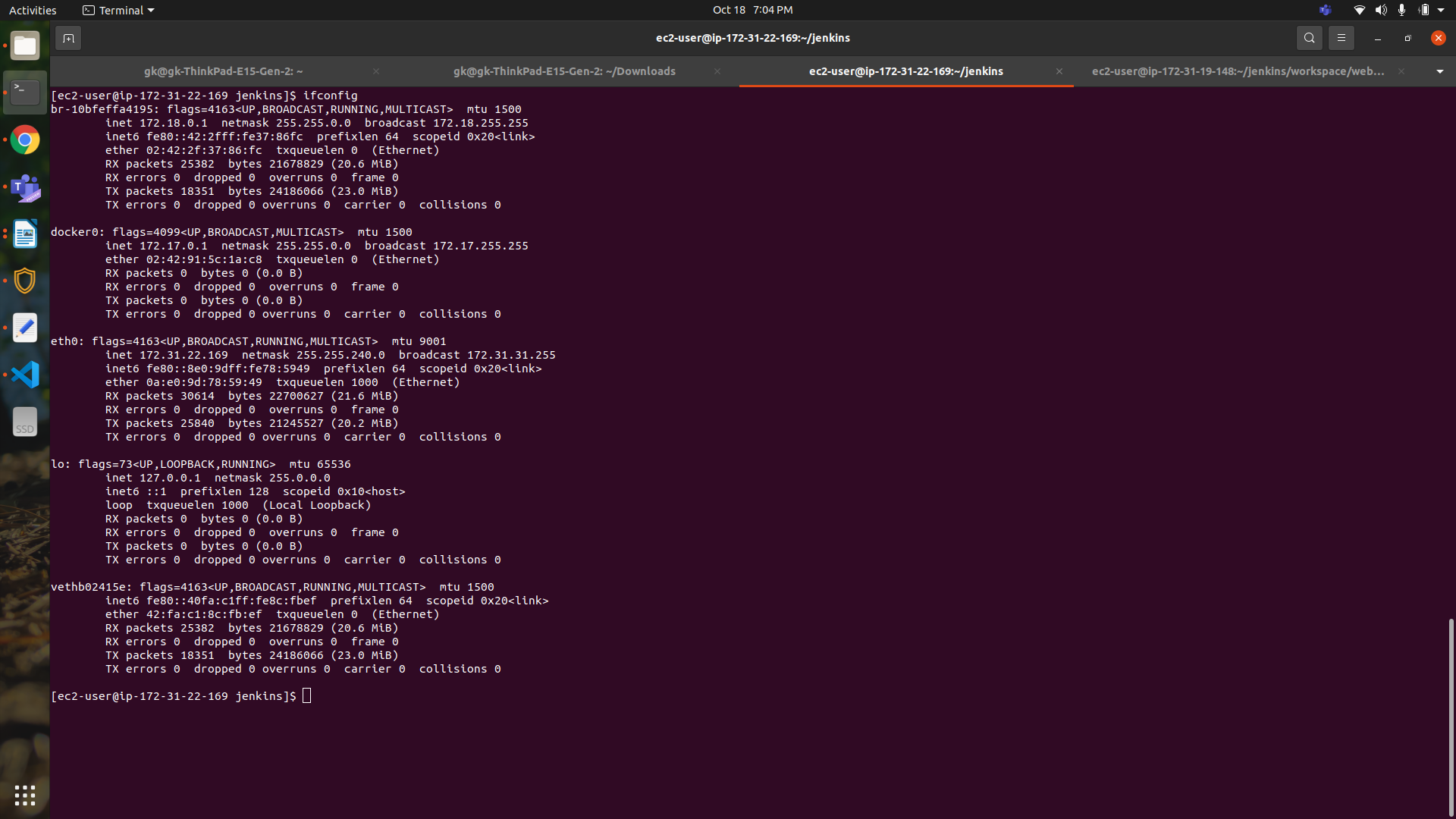
#Goto to webbrowser and enter publick ip of jenkins vm

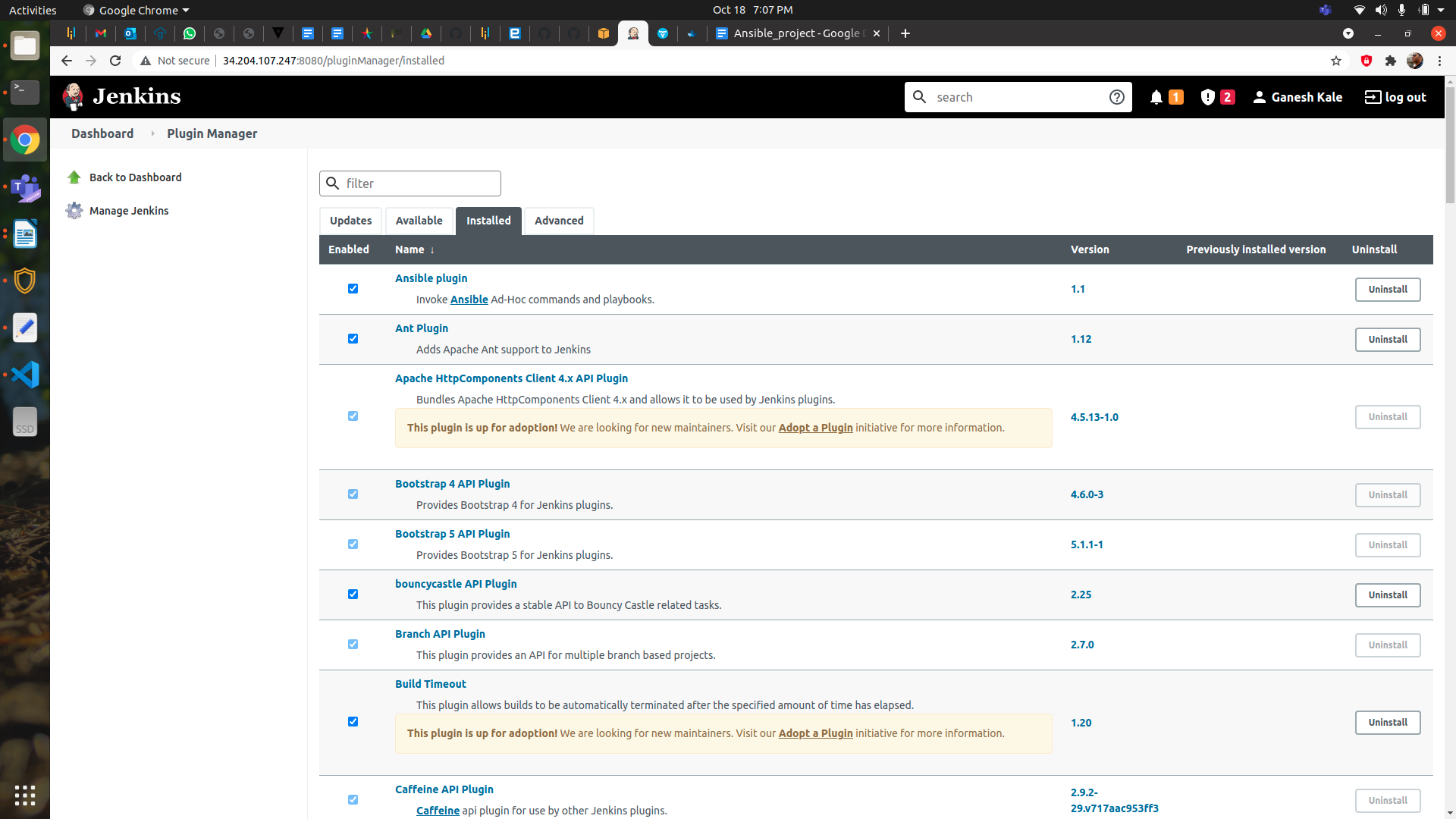
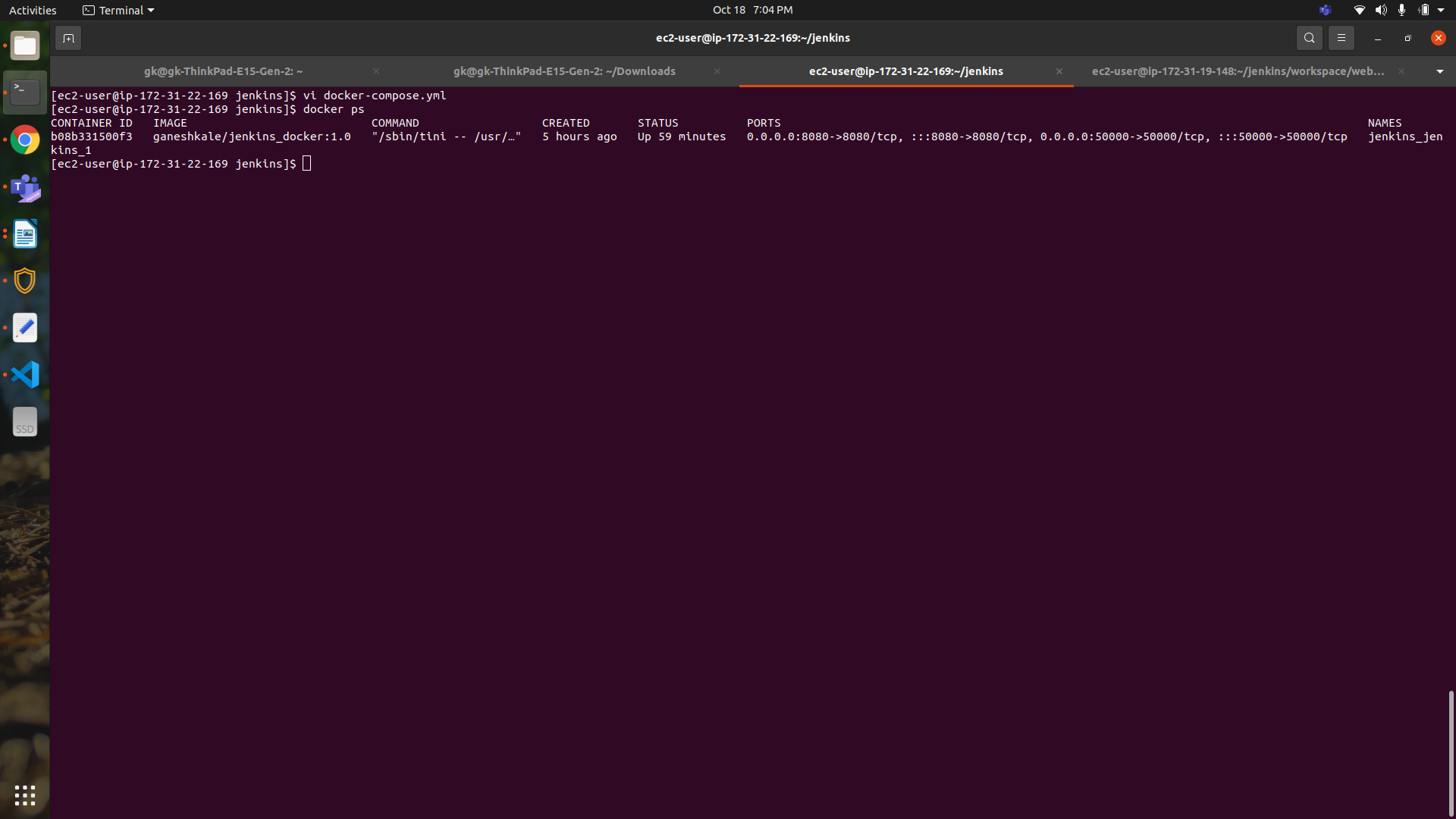
http://34.204.107.247:8080/

#Enter intial admin password and create our own user and password and install recommended plugins and wait#Login with new user id and password

#Goto Manage jenking -> Plugin Manager -> Available

#Search for ansible plugin and install it





2:-Write jenkins declarative pipeline for CI/CD process

pipeline {

agent none

stages{

stage ('Git clone') {

steps {

sh """

rm -rf addressbook

git clone https://github.com/ganeshjkale/addressbook.git

echo "current path = `pwd`"

"""

}

agent{

label 'compile'

}

tools {

git 'git'

}

}

stage ('Code Build'){

steps {

sh """

/usr/sbin/ifconfig

cd /home/ec2-user/jenkins/workspace/web\_container/addressbook

echo "Perform Validate"

echo "current path = `pwd`"

mvn validate

mvn compile

mvn test

mvn package

"""

}

agent{

label 'compile'

}

tools {

git 'git'

}

}

stage ('deployment'){

agent {

label 'compile'

}

steps{

sh """

cd /home/ec2-user/jenkins/workspace/web\_container/addressbook

echo "Run ansible playbook"

/usr/sbin/ifconfig

"""

ansiblePlaybook colorized: true, credentialsId: 'ec2-user', disableHostKeyChecking: true, playbook: '/home/ec2-user/jenkins/workspace/web\_container/addressbook/ansible\_playbook.yaml'

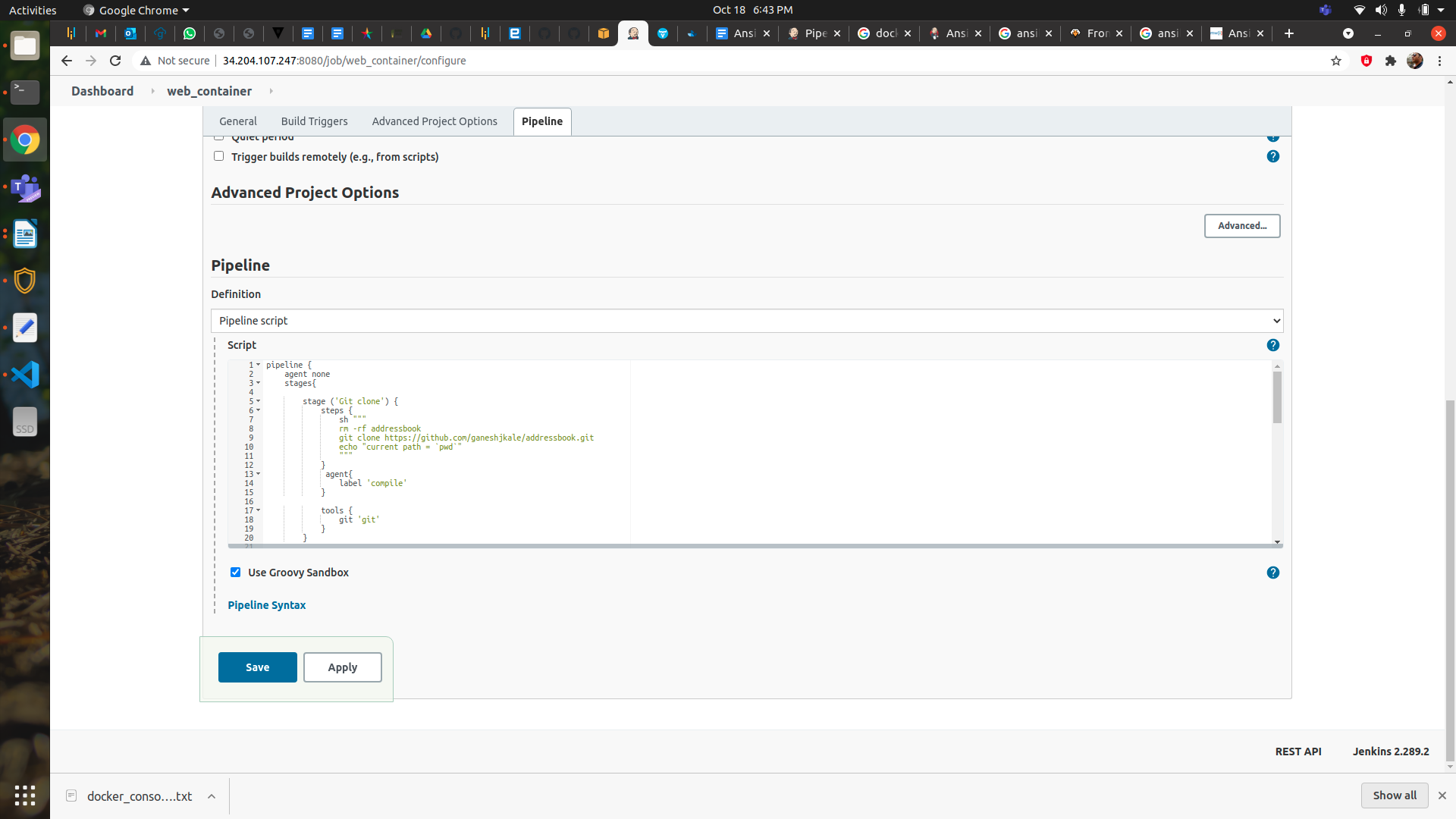
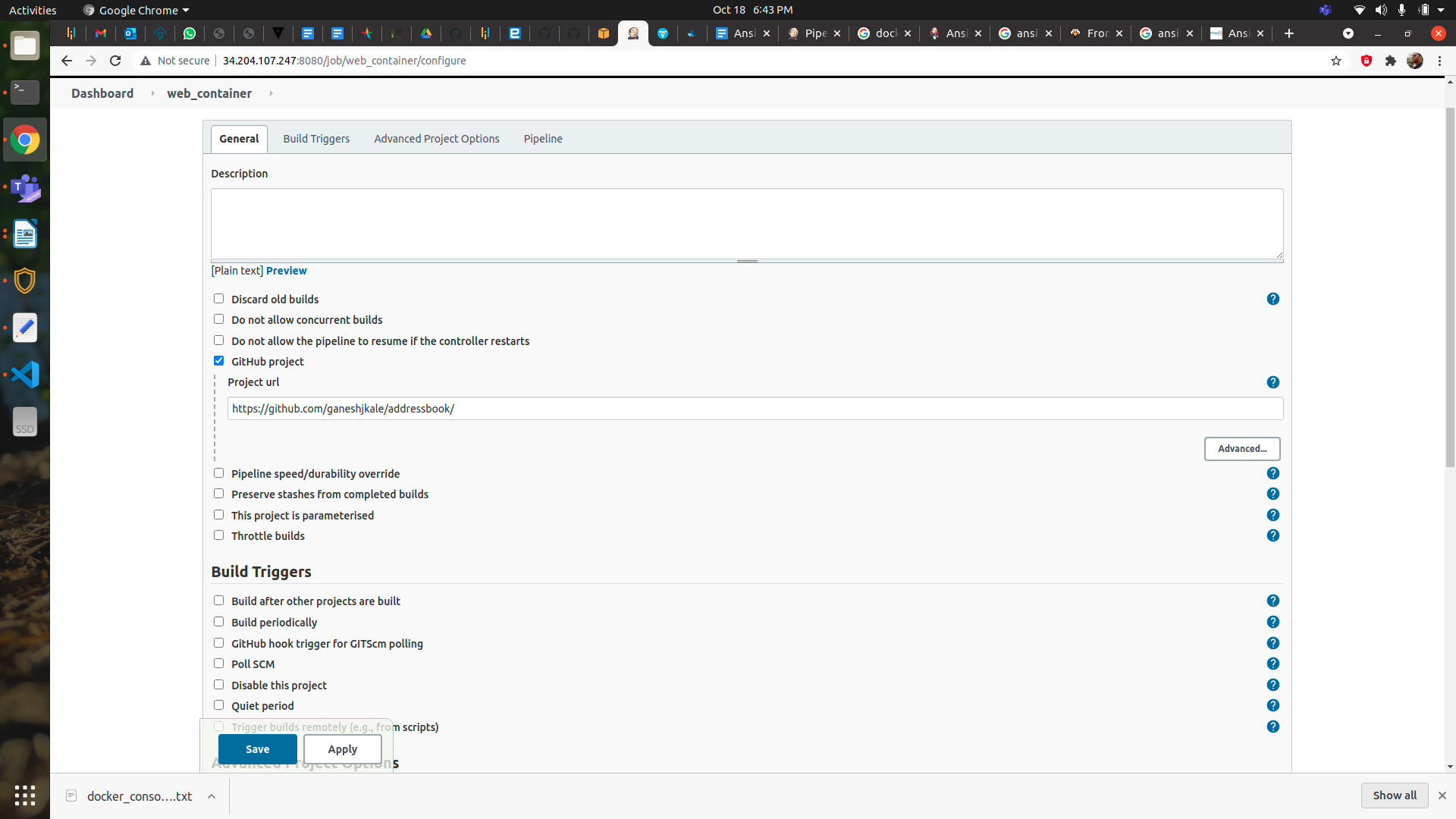
}

}

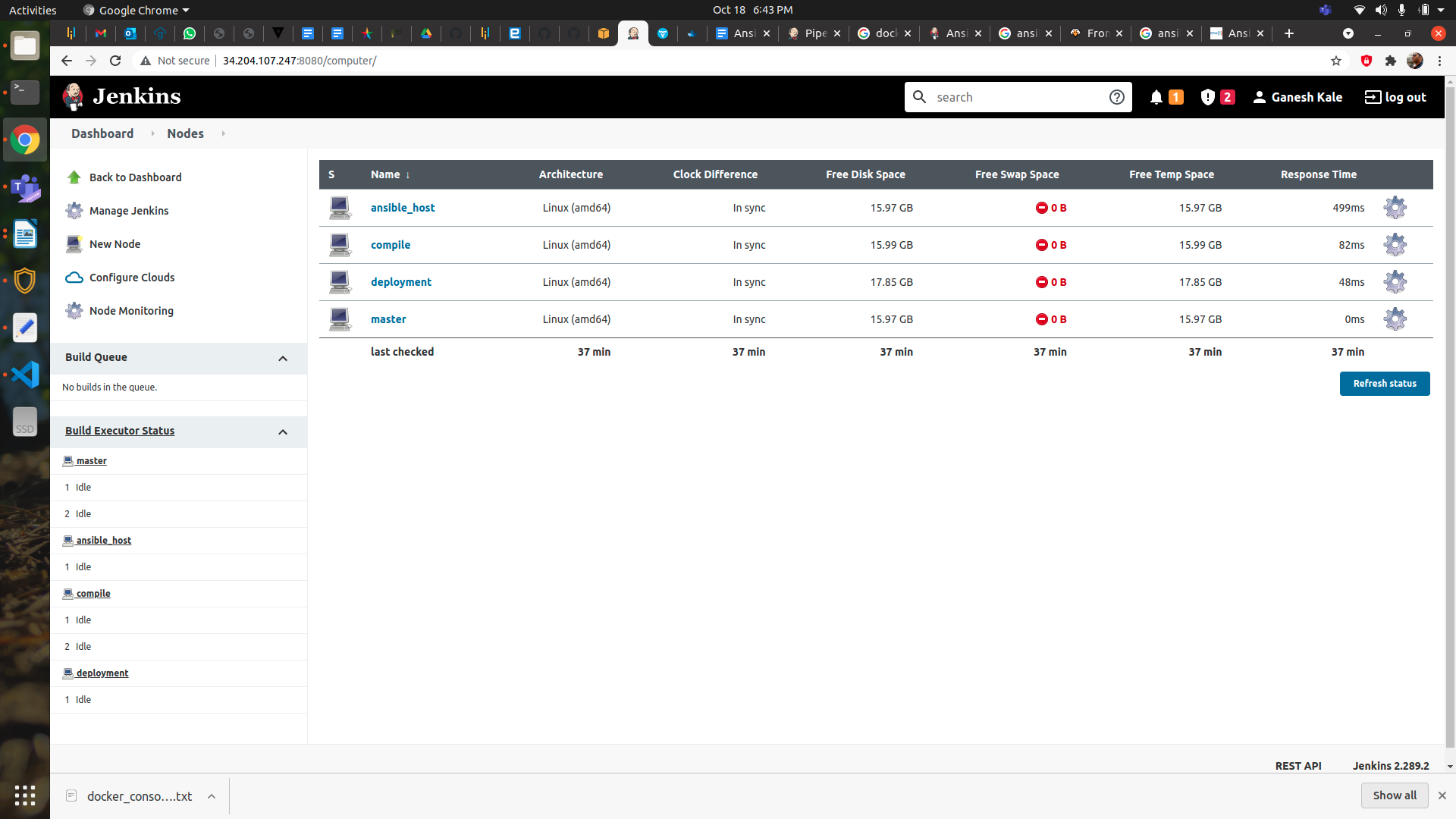
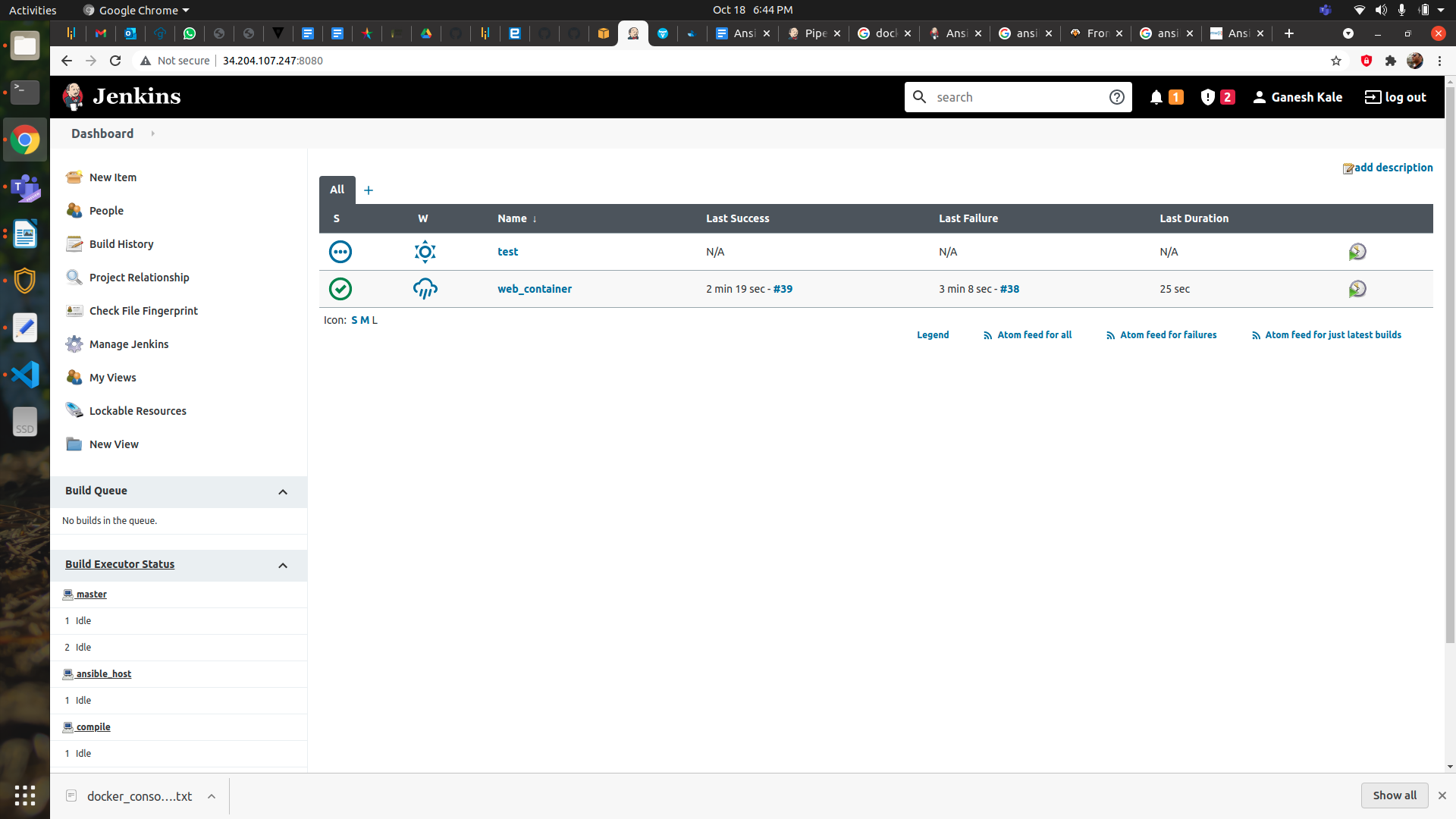
}

}

#Now create new pipeline job in jenkins web console as below screenshot and add above code in Pipeline section



#Add jenkins agents to run jobs on it also add ec2-user private key as secret to communicate with agents node



**3:-Write Ansible plabook and add into github : https://github.com/ganeshjkale/addressbook**

---

#Ansible Playbook to build and run a Docker containers

- name: Playbook to build and run Docker

hosts: localhost

become: true

gather\_facts: false

tasks:

- name: copying file with playbook

become: true

copy:

src: /home/ec2-user/jenkins/workspace/web\_container/addressbook/target

dest: /home/ec2-user/web\_docker\_path/

copy:

src: /home/ec2-user/jenkins/workspace/web\_container/addressbook/pom.xml

dest: /home/ec2-user/web\_docker\_path/

copy:

src: /home/ec2-user/jenkins/workspace/web\_container/addressbook/Dockerfile

dest: /home/ec2-user/web\_docker\_path/

- name: Build a Docker image using webapp.war file

docker\_image:

name: simple-docker-image

build:

path: /home/ec2-user/web\_docker\_path

pull: false

source: build

- name: Run Docker container using simple-docker-image

docker\_container:

name: simple-docker-container

image: maventest:1.0

state: started

recreate: yes

detach: true

ports:

- "8080:8080"

4:- Write docker web container Dockerfile and add into github : https://github.com/ganeshjkale/addressbook

FROM maven

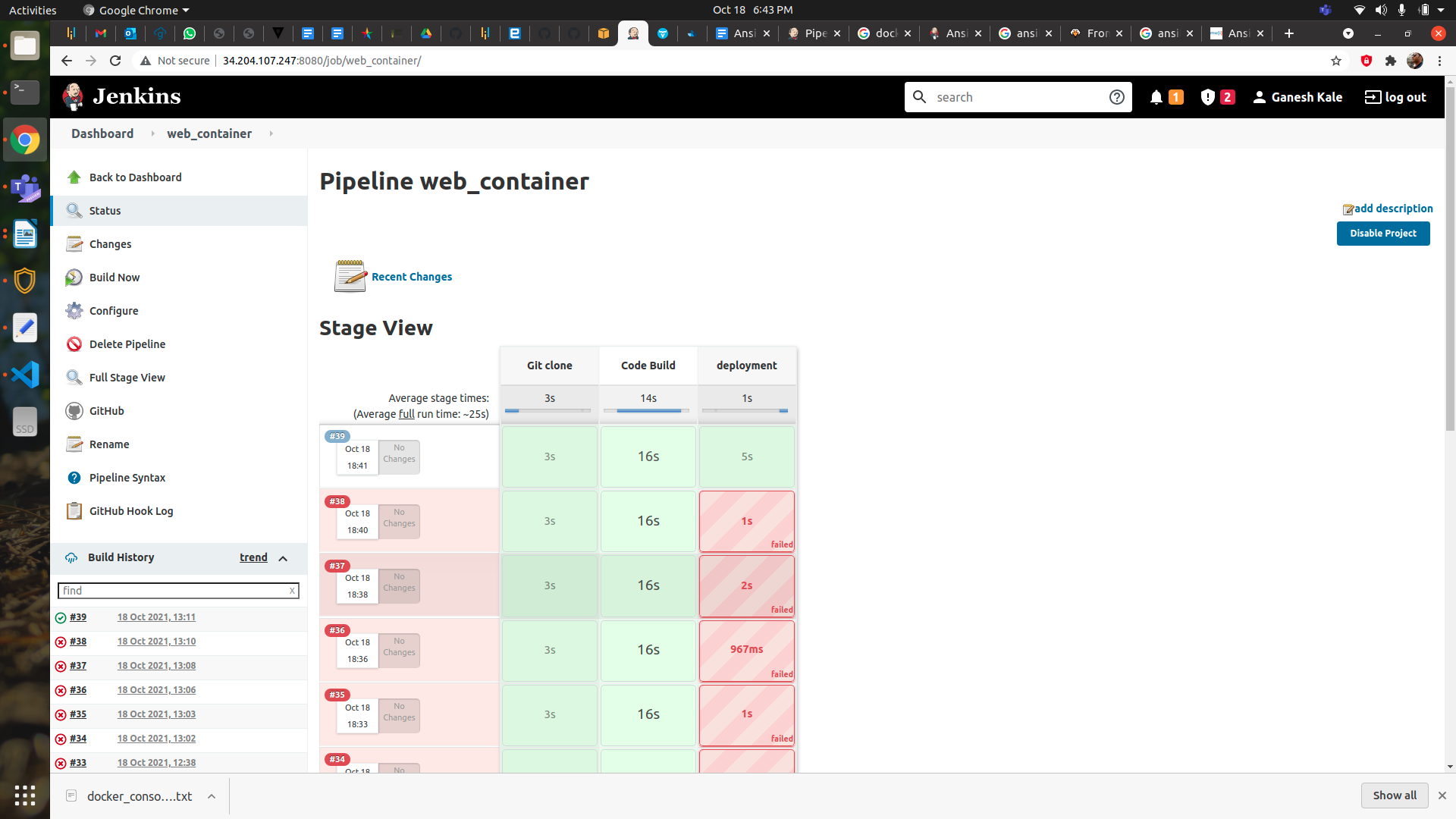
LABEL maven addressbook

WORKDIR addressbook

COPY . .

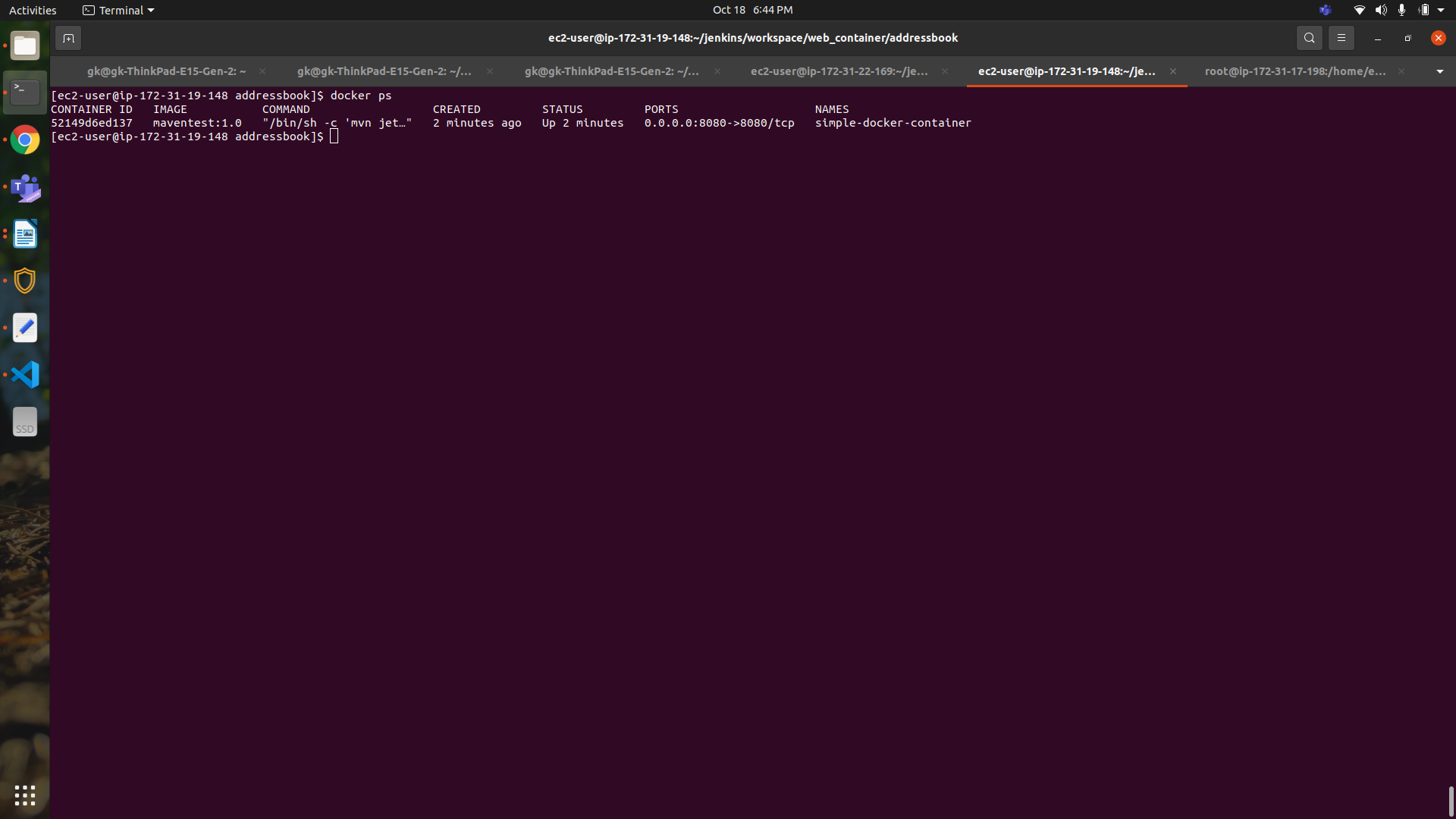
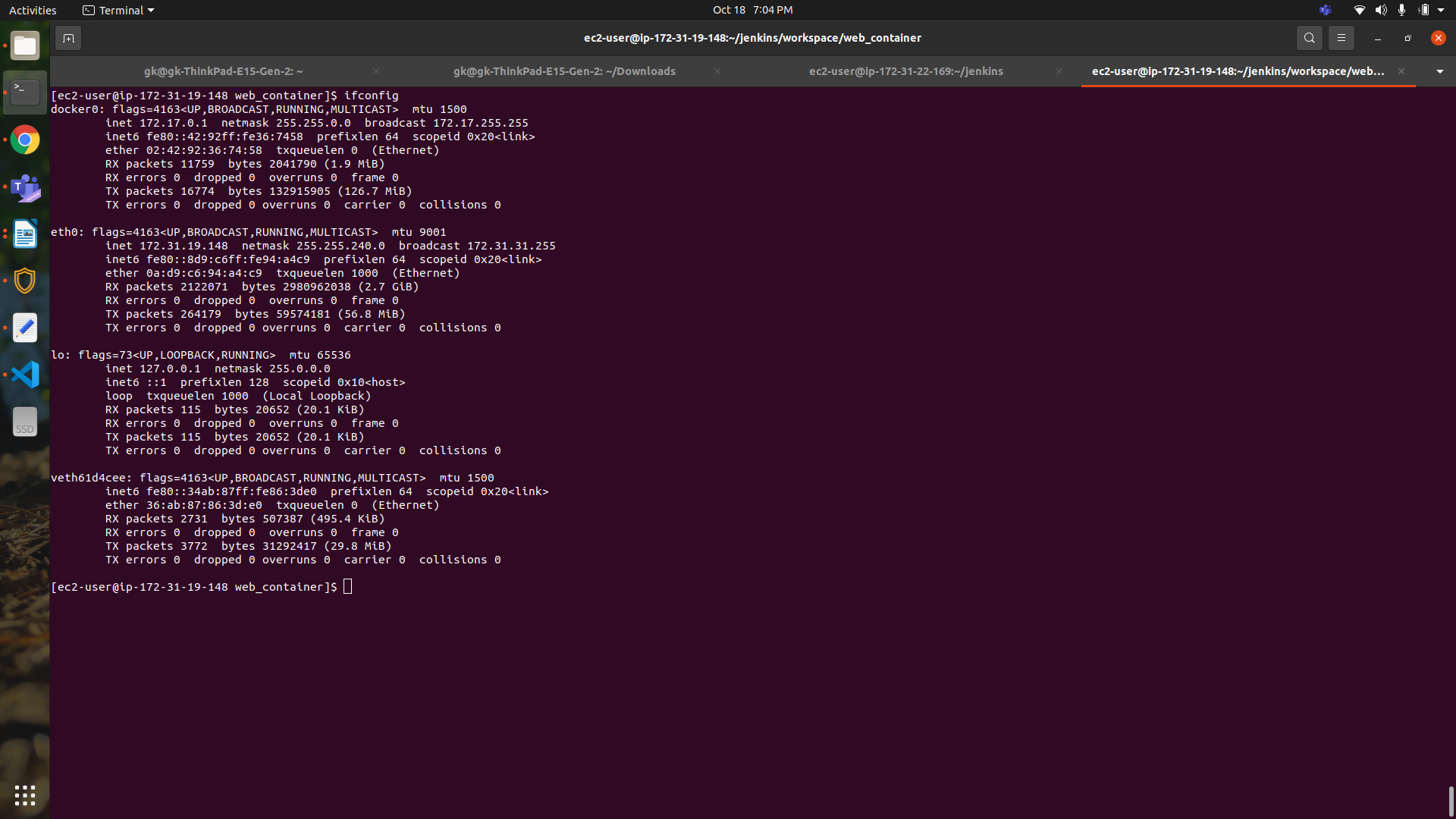
ENTRYPOINT mvn jetty:run

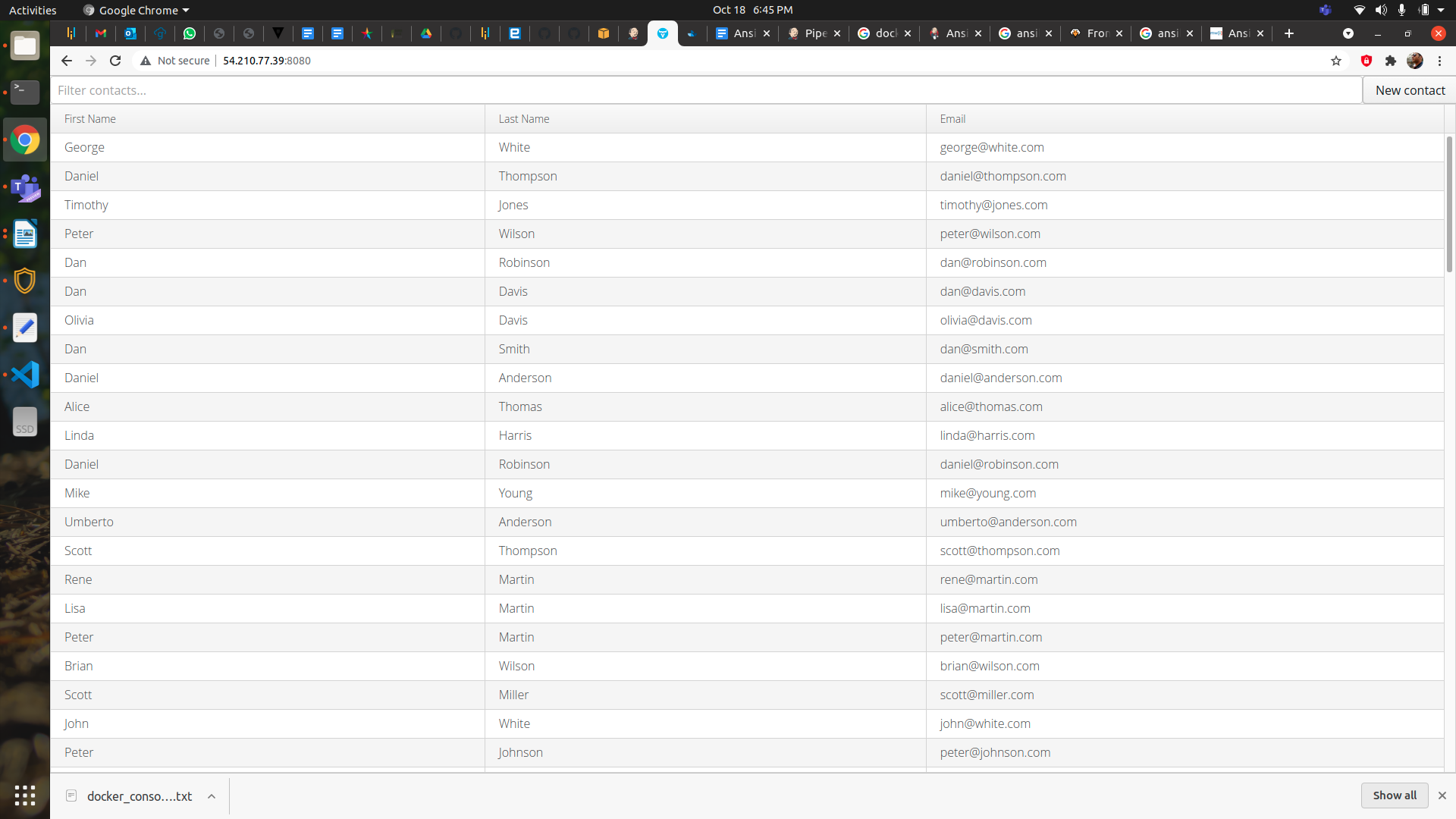
5:-Now run Build now on job “web\_container” Jenkins web console



6:-**After job is successfully working then goto vm 2 to verify web\_container running with new**

**data and also check** [**http://54.210.77.39:8080/**](http://54.210.77.39:8080/) **on browser**





**7:-Reference :-**

<https://github.com/ganeshjkale/addressbook> (Forked from adressbook opensource project)