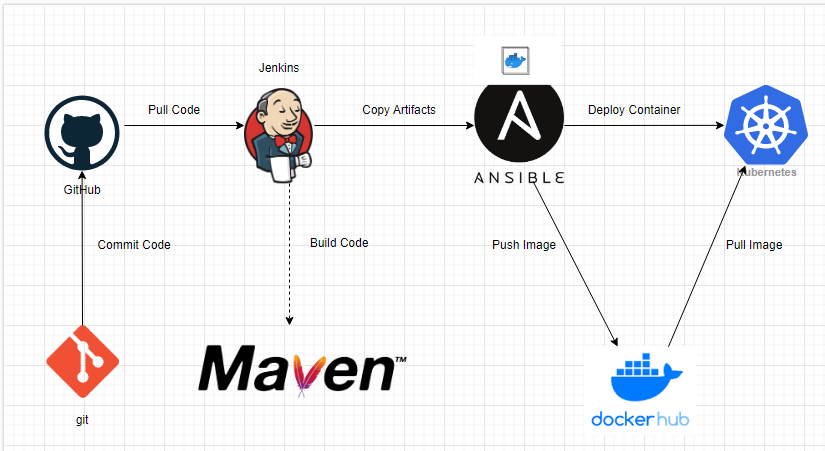
**Devops Project**

**CI/CD pipeline System**

**Requirements :**

* Git - local version control system.
* GitHub - As Distributed version control system.
* Jenkins - Continous Integration tool.
* Maven - As a Build Tool.
* Anisible - Configuration Management & Deployment tool.
* docker -Containerization
* Kubernetes - As Container Management Tool.

**Flow Diagram :**



**Resources to Setup CI and CD pipeline.**

* Free Tier AWS account.
* GitHub account (for source code and documentation).
* MobaXterm – enhanced terminal for windows with X11 Server tabbed SSH clients, network tool and much more.
* Git – local version control system.

**Setup Jenkins Server**

* Setup a Linux EC2 instance
* Install Java
* Install Jenkins
* Start Jenkins
* Access Web UI on port 8080

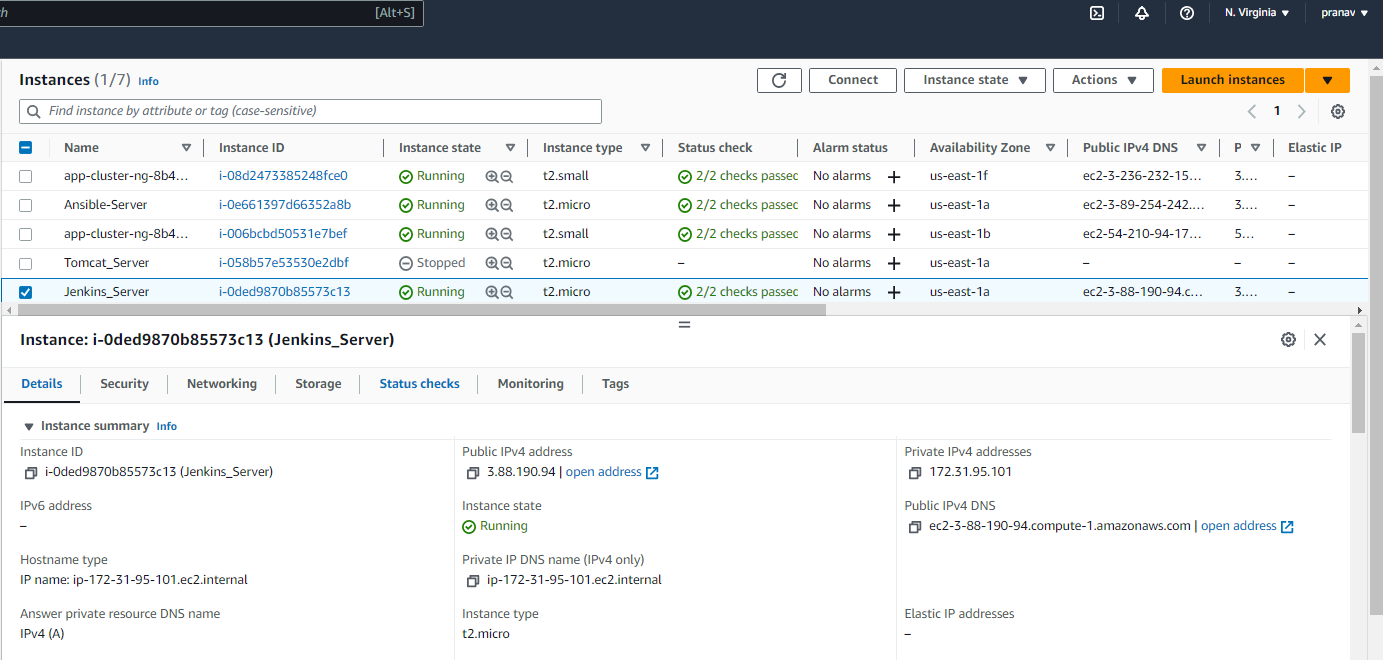
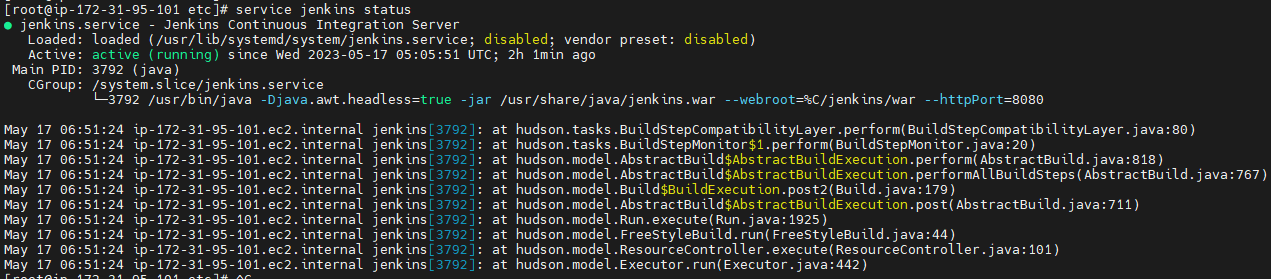


Fig.EC2 Instance for Jenkins Server

[root@ip-172-31-95-101 ~]# service jenkins start

Starting jenkins (via systemctl): [ OK ]



**Integrate Git with Jenkins**

* Install Git on Jenkins Instances
* Install GitHub plug in on Jenkins GUI
* Configure Git on Jenkins GUI

**Install Git on Jenkins Instances**



**Install GitHub plug in on Jenkins GUI**

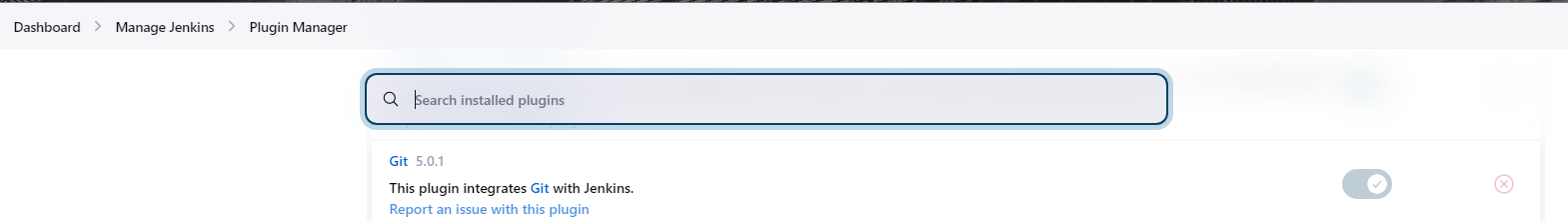


Fig. Plugin Integrates git with Jenkins.

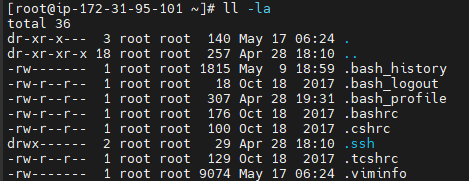
**Configure Git on Jenkins GUI**



Fig. Configure Git on Jenkins GUI

**Integrate Maven with Jenkins**

* Setup Maven on Jenkins Server
* Setup Environment Variables
* JAVA\_HOME, M2, M2\_HOME
* Install Maven Plugin
* Configure Maven and Java



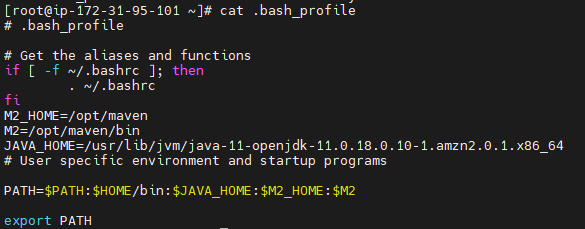
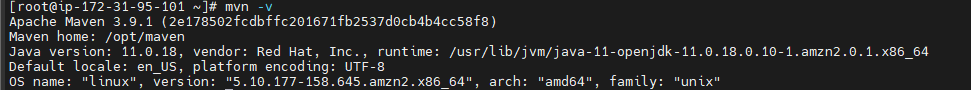


Fig:Setup Environment Variables



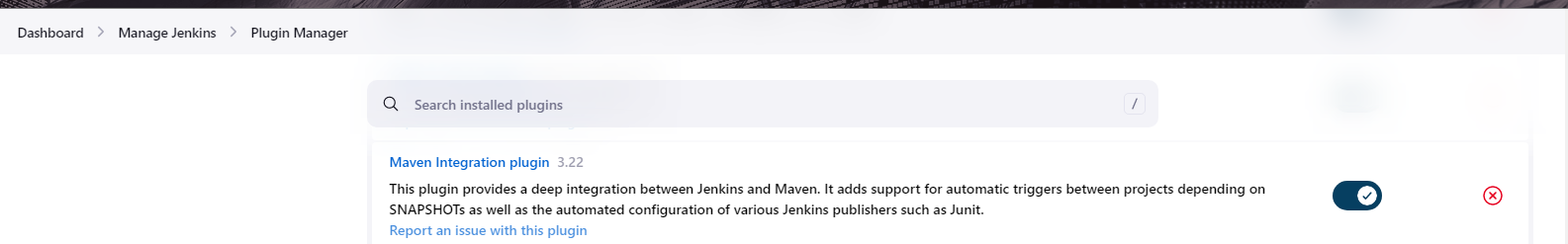


Fig: Install Maven Plugin



Fig. Java Global Tool Configuration.

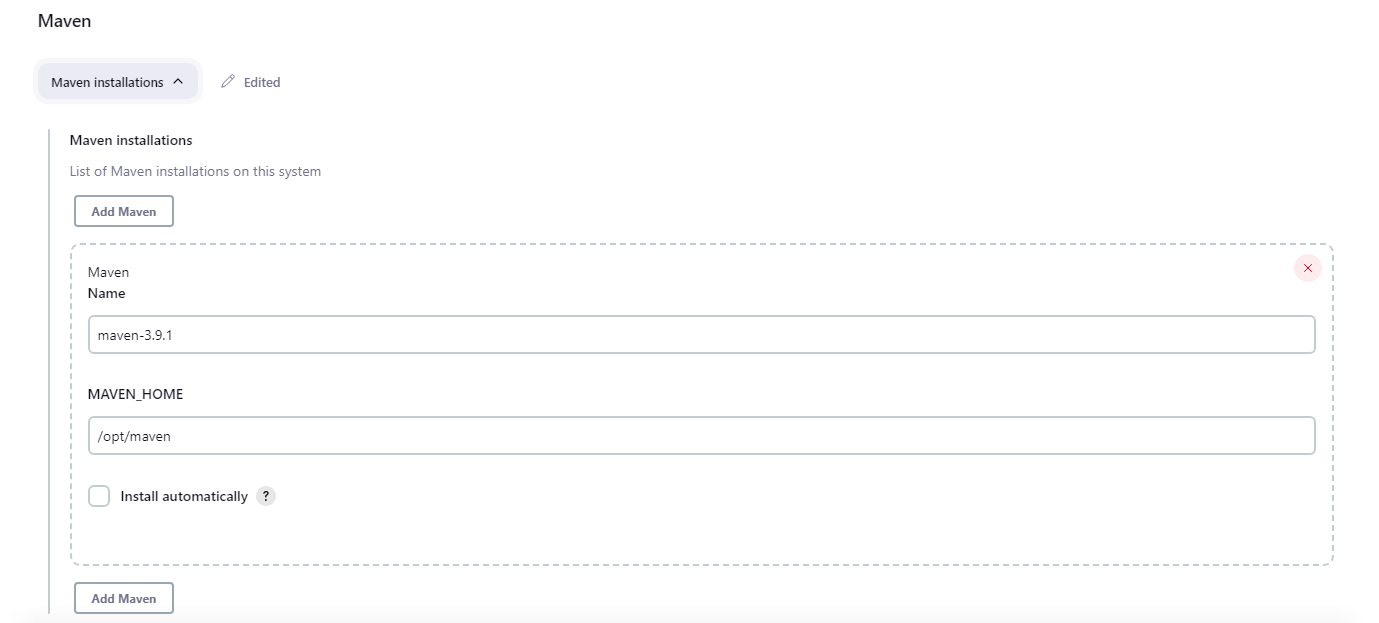


Fig. Maven Global Tool Configuration

**Integrating Docker in CI/CD pipeline**

**Setup a Docker Environment**

**Setup Docker Host**

* Setup a Linux EC2 Instance
* Install Docker
* Start docker services

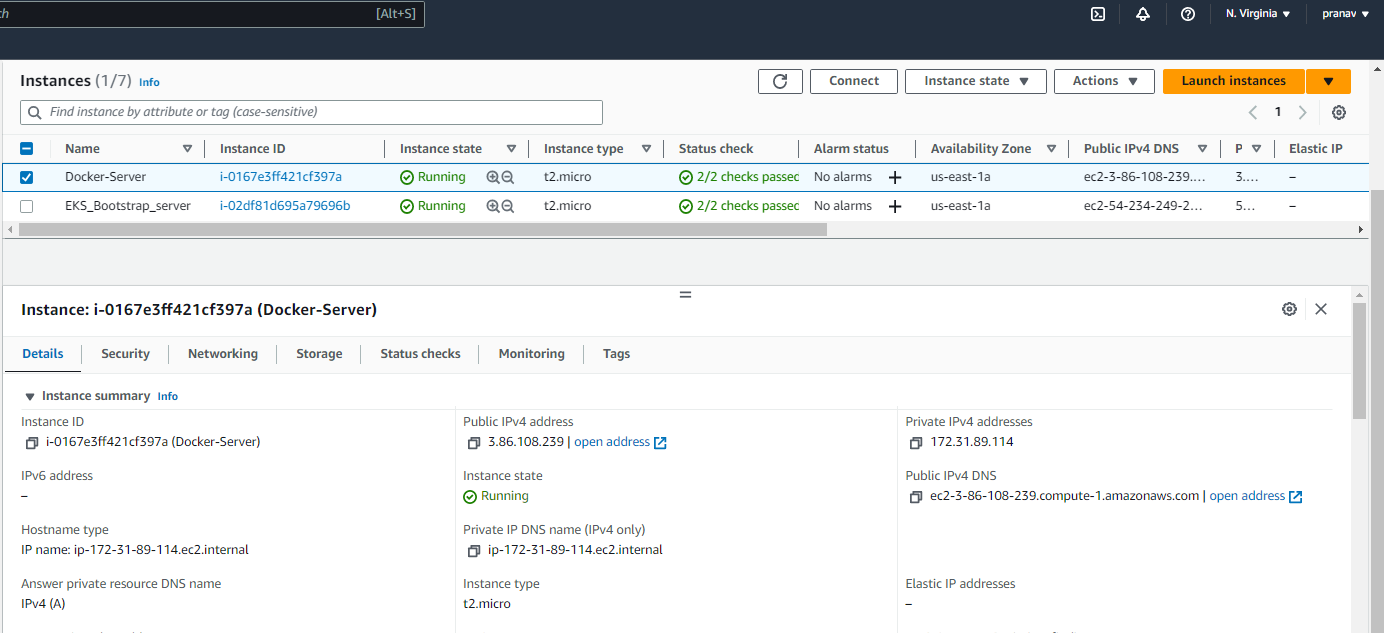


Fig. Docker Server EC2 Instance





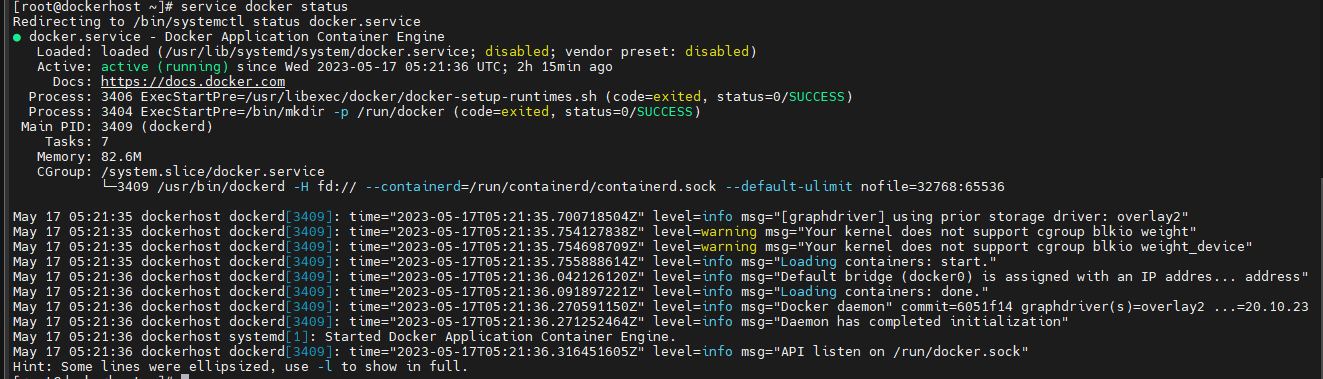


Fig.Starting Docker Server

**Integrate Docker with Jenkins**

* Create a docker admin user
* Install “Publish Over SSH “plugin
* Add Docker Host to Jenkins “configure systems”

[root@dockerhost ~]# useradd dockeradmin // Creating Users

[root@dockerhost ~]# passwd dockeradmin // Creating Password

Changing password for user dockeradmin.

New password:

BAD PASSWORD: The password contains the user name in some form

Retype new password:

[root@dockerhost ~]# usermod -aG docker dockeradmin

// Modifying Docker Group

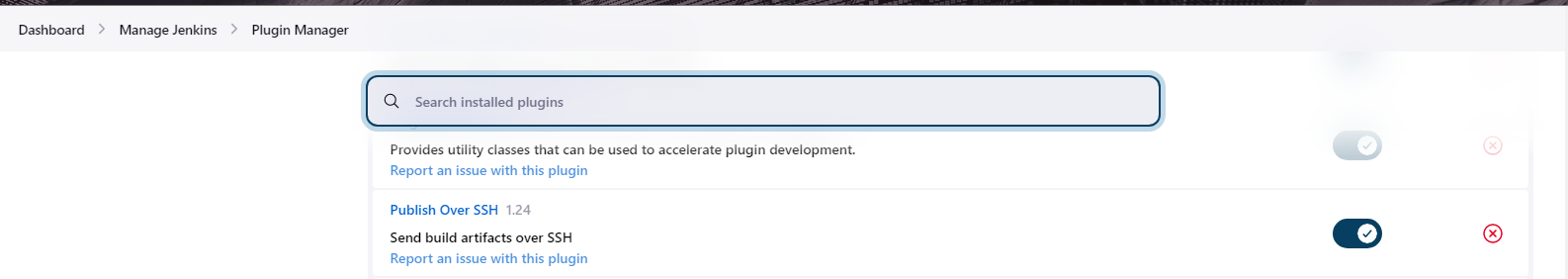
[root@dockerhost ~]# id dockeradmin

uid=1001(dockeradmin) gid=1001(dockeradmin) groups=1001(dockeradmin),992(docker)

[root@dockerhost ~]# vi /etc/ssh/sshd\_config // Configure path password

[root@dockerhost ~]# service sshd reload

**Install “Publish Over SSH “plugin**

****

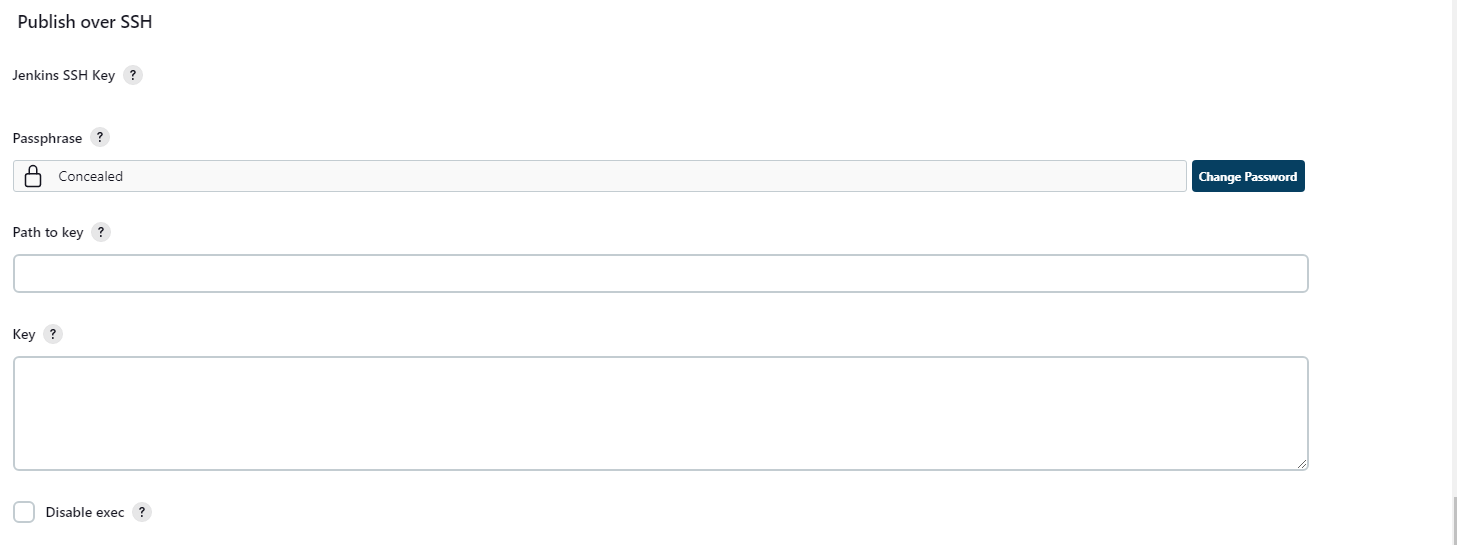
****

Fig. Publish Over SSH plugin

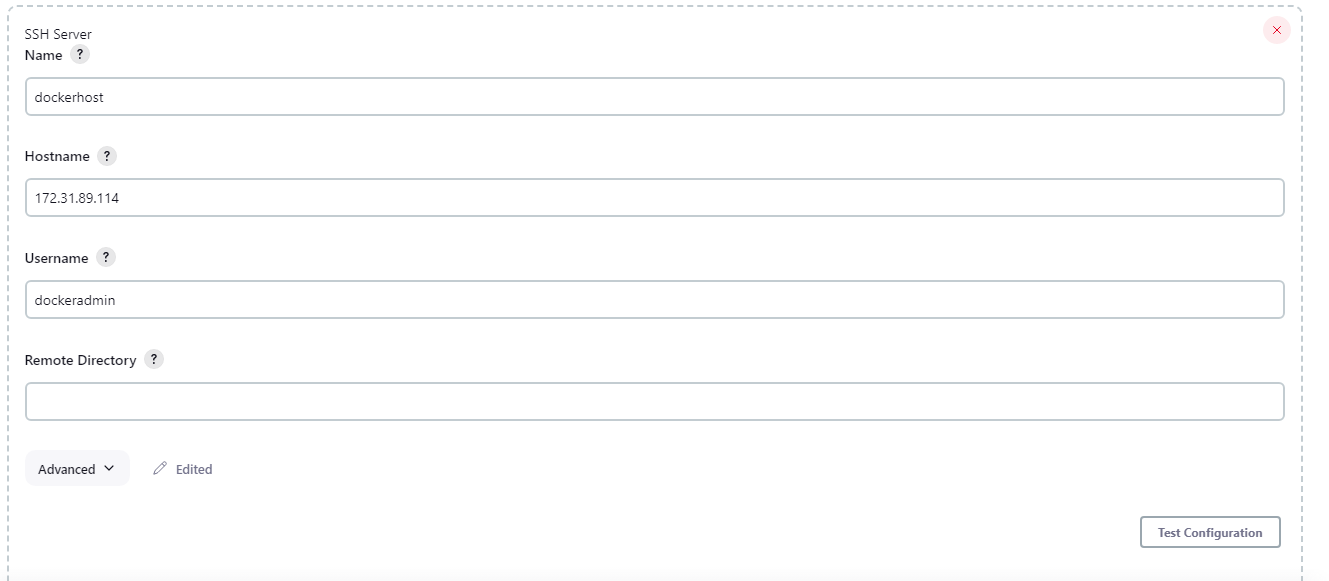
****

Fig. Adding Docker Host to Jenkins “configure systems”

**Integrate Ansible in CI/CD Pipeline:**

**Prepare Ansible Server:**

* Setup EC2 Instance
* Setup hostname
* Create ansadmin users
* Add Users to sudoers file
* Generate ssh keys
* Enable Password Based Login
* Install Ansible

**Setup EC2 Instance**

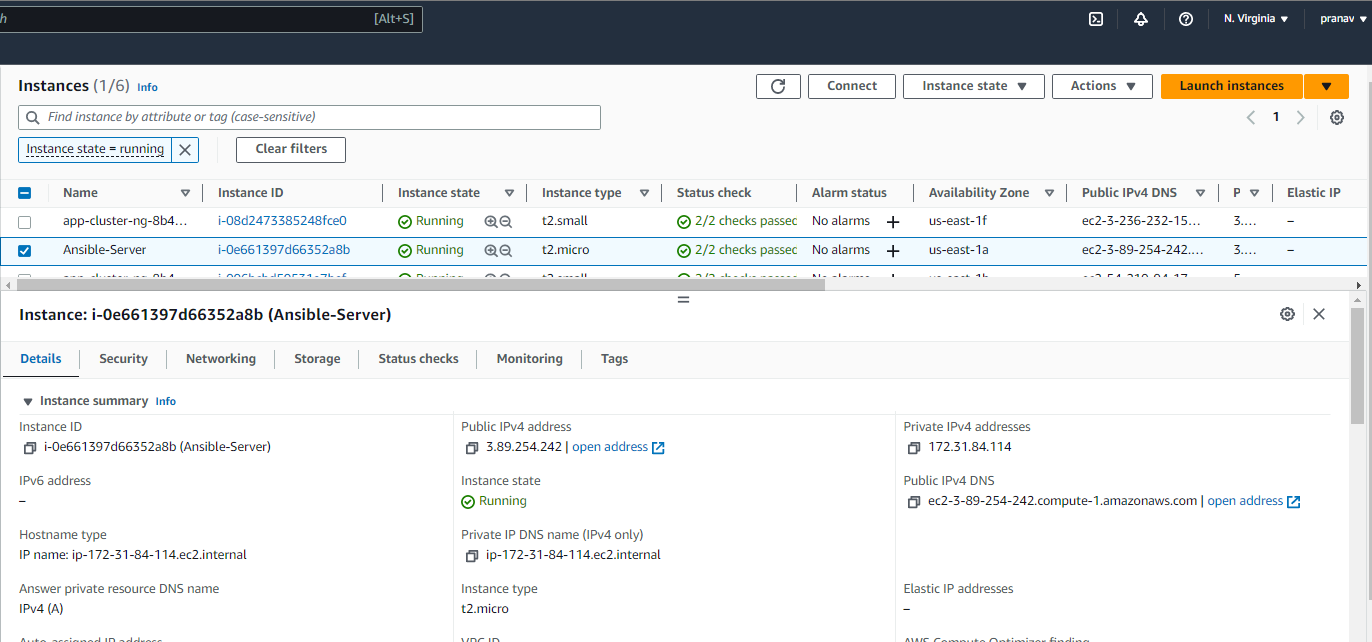


Fig.EC2 Instance Setup For Ansible

**Generated SSH Keys:**

[ansadmin@nsible-server ~]$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/ansadmin/.ssh/id\_rsa):

Created directory '/home/ansadmin/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/ansadmin/.ssh/id\_rsa.

Your public key has been saved in /home/ansadmin/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:1eWQGH9lwcDotDMlxHjHjpVn6nQOtefY+iL1KBSxF+Q ansadmin@nsible-server

The key's randomart image is:

+---[RSA 2048]----+

| =+=+\*.+|

| ..Oo%.B |

| = @oE..|

| . \*o\*.o.|

| S =o++.|

| ..o.o|

| . . + |

| o + .|

| o o.|

+----[SHA256]-----+

**Install Ansible:**

[root@nsible-server ~]# python --version

**Python 2.7.18**

[root@nsible-server ~]# ansible --version

**ansible 2.9.23**

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /usr/bin/ansible

python version = 2.7.18 (default, Feb 28 2023, 02:51:06) [GCC 7.3.1 20180712 (Red Hat 7.3.1-15)]

**Integrate Docker with Ansible**:

**Create ansadmin**

[root@dockerhost ~]# useradd ansadmin

[root@dockerhost ~]# passwd ansadmin

Changing password for user ansadmin.

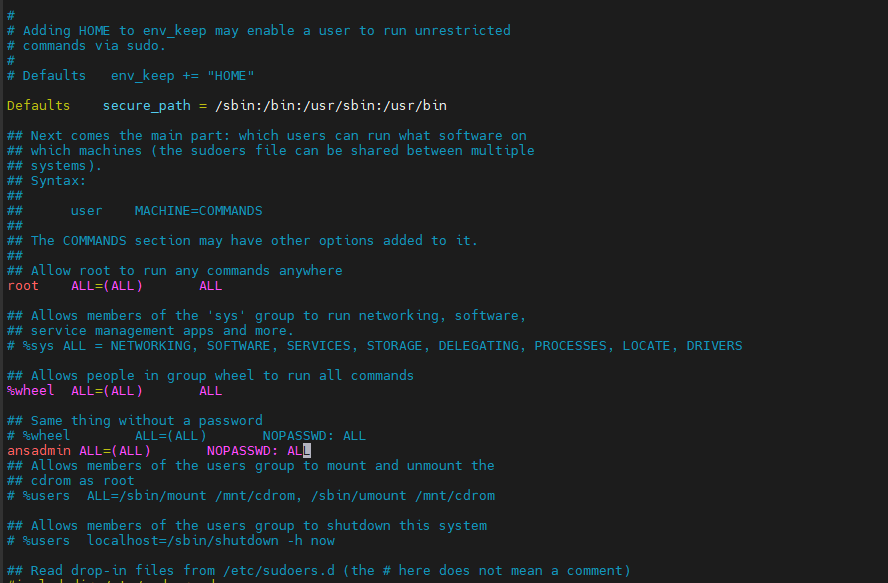
New password:

BAD PASSWORD: The password is shorter than 8 characters

Retype new password:

passwd: all authentication tokens updated successfully.

**Add ansadmin to sudoers file**

****

**Enable Password Based Login**

[root@dockerhost ~]# grep Password /etc/ssh/sshd\_config

PasswordAuthentication yes

#PermitEmptyPasswords no

#PasswordAuthentication no

# PasswordAuthentication. Depending on your PAM configuration,

# PAM authentication, then enable this but set PasswordAuthentication

**Copy public key to target system**

**Copy ssh keys**

[ansadmin@nsible-server .ssh]$ ssh-copy-id 172.31.89.114

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id\_rsa.pub"

The authenticity of host '172.31.89.114 (172.31.89.114)' can't be established.

ECDSA key fingerprint is SHA256:VyyK5S3p9Ejp2w2vBnHiHfoWMiuCz+0orNnNuwRU10c.

ECDSA key fingerprint is MD5:95:25:97:7b:f4:10:7f:ab:fb:8d:37:f4:88:c7:50:5b.

Are you sure you want to continue connecting (yes/no)? yes

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

ansadmin@172.31.89.114's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '172.31.89.114'"

and check to make sure that only the key(s) you wanted were added.

**Copying sshkey to docker host.**

[root@dockerhost ~]# sudo su - ansadmin

[ansadmin@dockerhost ~]$ ll -la

total 12

drwx------ 3 ansadmin ansadmin 74 May 6 18:17 .

drwxr-xr-x 5 root root 57 May 6 18:07 ..

-rw-r--r-- 1 ansadmin ansadmin 18 Jul 15 2020 .bash\_logout

-rw-r--r-- 1 ansadmin ansadmin 193 Jul 15 2020 .bash\_profile

-rw-r--r-- 1 ansadmin ansadmin 231 Jul 15 2020 .bashrc

drwx------ 2 ansadmin ansadmin 29 May 6 18:17 .ssh

[ansadmin@dockerhost ~]$ cd .ssh

[ansadmin@dockerhost .ssh]$ ll

total 4

-rw------- 1 ansadmin ansadmin 404 May 6 18:17 authorized\_keys

[ansadmin@dockerhost .ssh]$ date

Sat May 6 18:18:46 UTC 2023

**Test the Connection**

[ansadmin@nsible-server .ssh]$ ansible all -m ping

[WARNING]: Platform linux on host 172.31.89.114 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python

interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.

172.31.89.114 | SUCCESS => {

"ansible\_facts": {

"discovered\_interpreter\_python": "/usr/bin/python"

},

"changed": false,

"ping": "pong"

}

**Connection Success**

[ansadmin@dockerhost ~]$ uptime

18:25:05 up 19 min, 1 user, load average: 0.00, 0.00, 0.00

**Integrate Ansible with Jenkins:**

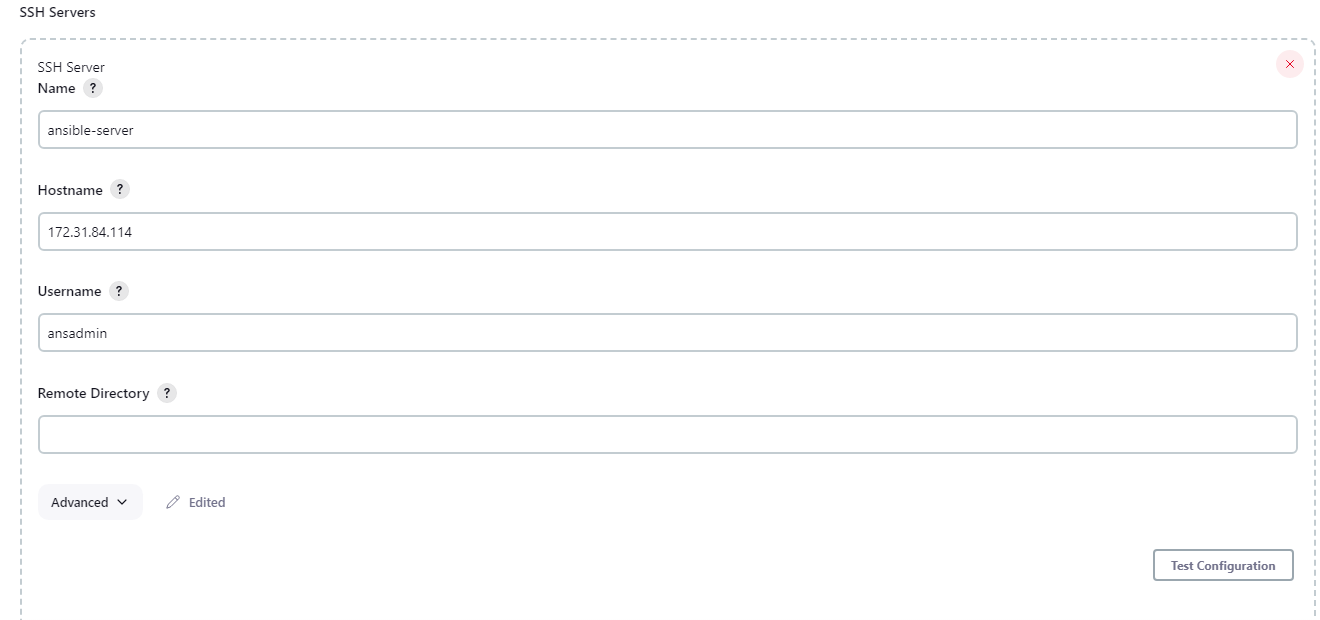
****

Fig. Configuring Ansible with Jenkins

**Creating docker directory on Ansible Server:**

[ansadmin@nsible-server ~]# cd /opt

[ansadmin@nsible-server opt]# mkdir docker

[ansadmin@nsible-server opt]# chown ansadmin:ansadmin docker

[ansadmin@nsible-server opt]$ ll

total 0

drwxr-xr-x 4 root root 33 Apr 20 18:29 aws

drwx--x--x 4 root root 28 May 7 17:18 containerd

drwxr-xr-x 2 ansadmin ansadmin 60 May 7 19:44 docker

drwxr-xr-x 2 root root 6 Aug 16 2018 rh

**Install docker**

**Creating Docker File:**

[ansadmin@ansible-server docker]$vi Dockerfile

[ansadmin@ansible-server docker]$ cat Dockerfile

FROM tomcat

RUN cp -R /usr/local/tomcat/webapps.dist/\* /usr/local/tomcat/webapps

COPY ./\*.war /usr/local/tomcat/webapps

**Adding address for host**

[ansadmin@ansible-server docker]$ cat hosts

[ansible]

172.31.84.114

[dockeradmin]

172.31.89.114

[ansadmin@ansible-server docker]$ ansible all -a uptime

[WARNING]: Platform linux on host 172.31.89.114 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python

interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.

172.31.89.114 | CHANGED | rc=0 >>

18:13:20 up 54 min, 2 users, load average: 0.00, 0.00, 0.00

[WARNING]: Platform linux on host 172.31.84.114 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python

interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.

172.31.84.114 | CHANGED | rc=0 >>

18:13:20 up 1:25, 2 users, load average: 0.09, 0.04, 0.01

**Creating playbook:**

[ansadmin@ansible-server docker]$ cat create\_image\_regapp.yml

---

- hosts: ansible

tasks:

- name: create docker image

command: docker build -t regapp:latest .

args:

chdir: /opt/docker

- name: create tag to push image onto dockerhub

command: docker tag regapp:latest pranav27/regapp:latest

- name: push docker image

command: docker push pranav27/regapp:latest

[ansadmin@ansible-server docker]$ cat docker\_deployment.yml

---

- hosts: dockerhost

tasks:

- name: Stop exisitng container

command: docker stop regapp-server

ignore\_errors: yes

- name: remove the container

command: docker rm regapp-server

ignore\_errors: yes

- name: remove image

command: docker rmi pranav27/regapp

ignore\_errors: yes

- name: create container

command: docker run -d --name regapp-server -p 8082:8080 pranav27/regapp:latest

**Complete CI and CD job to build and deploy code on Kubernetes:**

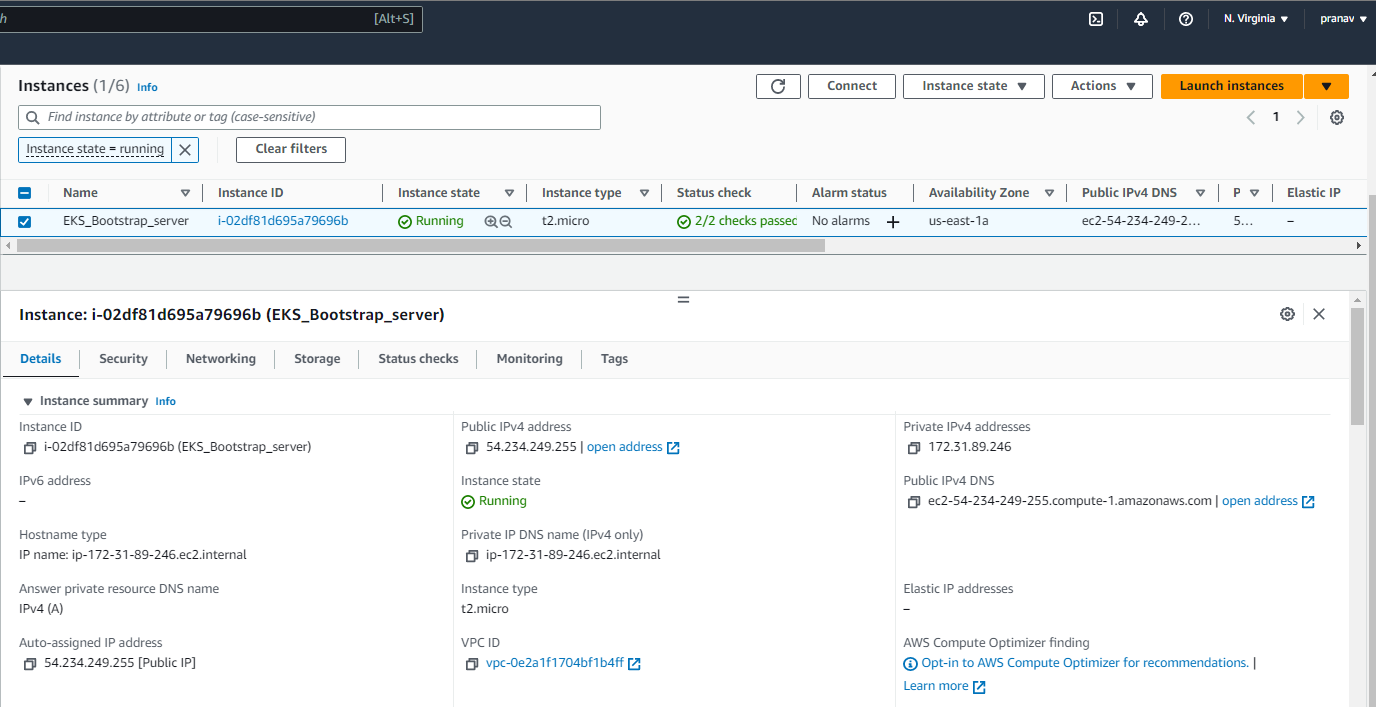


Fig. Bootstrap Server Running

**Install AWSCLI latest version**

[ec2-user@ip-172-31-89-246 ~]$ sudo su –

[root@ip-172-31-89-246 ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 54.9M 100 54.9M 0 0 281M 0 --:--:-- --:--:-- --:--:-- 281M

[root@ip-172-31-89-246 ~]# unzip awscliv2.zip

Archive: awscliv2.zip

[root@ip-172-31-89-246 ~]# sudo ./aws/install

You can now run: /usr/local/bin/aws --version

[root@ip-172-31-89-246 ~]# aws --version

aws-cli/2.11.18 Python/3.11.3 Linux/5.10.177-158.645.amzn2.x86\_64 exe/x86\_64.amzn.2 prompt/off

**Setup kubectl**

**Download kubectl version 1.26**

[root@ip-172-31-89-246 ~]# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.26.2/2023-03-17/bin/linux/amd64/kubectl

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 45.8M 100 45.8M 0 0 5044k 0 0:00:09 0:00:09 --:--:-- 5158k

**Grant execution permissions to kubectl executable**

[root@ip-172-31-89-246 ~]# chmod +x ./kubectl

**Move kubectl onto /usr/local/bin**

[root@ip-172-31-89-246 ~]# mv ./kubectl /usr/local/bin

**Test that your kubectl installation was successful**

[root@ip-172-31-89-246 ~]# kubectl version

WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version.

Client Version: version.Info{Major:"1", Minor:"26+", GitVersion:"v1.26.2-eks-a59e1f0", GitCommit:"8b68f4b95d7121d039ceebd30870e48acc7772e4", GitTreeState:"clean", BuildDate:"2023-03-09T20:03:04Z", GoVersion:"go1.19.6", Compiler:"gc", Platform:"linux/amd64"}

Kustomize Version: v4.5.7

The connection to the server localhost:8080 was refused - did you specify the right host or port?

**Setup eksctl**

[root@ip-172-31-89-246 ~]# curl -sL "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_checksums.txt" | grep $PLATFORM | sha256sum --check

eksctl\_Linux\_amd64.tar.gz: OK

**Move the extracted binary to /usr/local/bin**

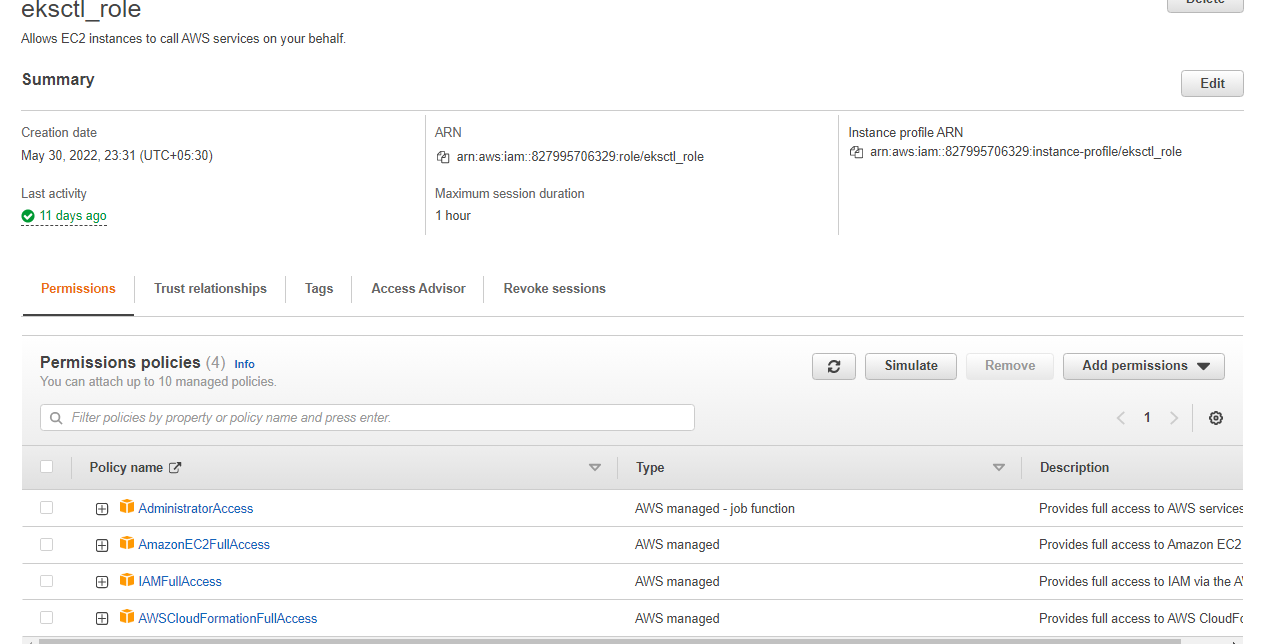
[root@ip-172-31-89-246 tmp]# mv eksctl /usr/local/bin

**Test that your eksctl installation was successful**

[root@ip-172-31-89-246 tmp]# eksctl version

0.140.0

**Create an IAM Role and attach it to EC2 instance**



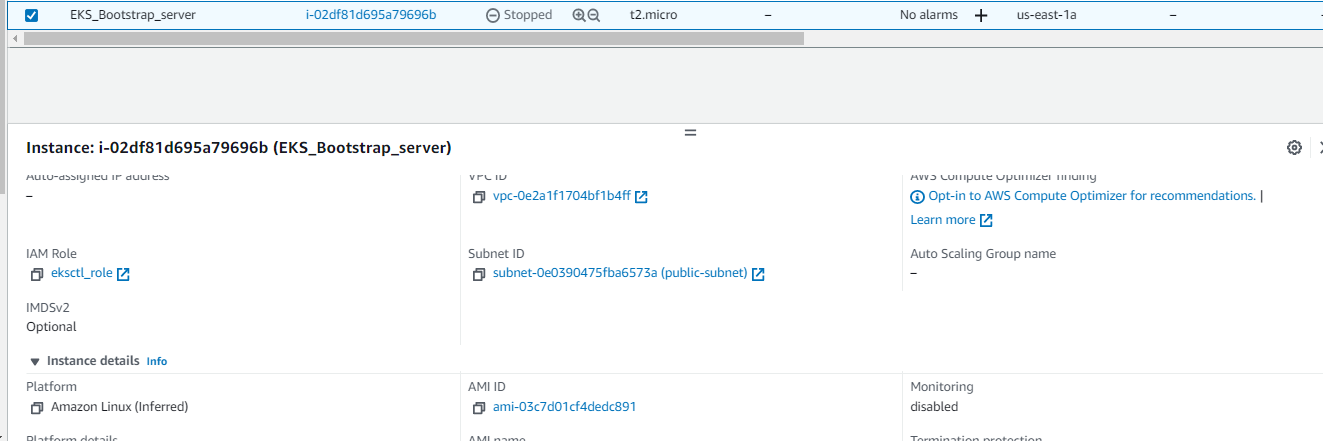


Fig.Attached eksctl\_role to bootstrap server

**Create your cluster and nodes**

[root@ip-172-31-89-246 ~]# cd /tmp

[root@ip-172-31-89-246 tmp]# eksctl create cluster --name app-cluster \

> --region us-east-1 \

> --node-type t2.small

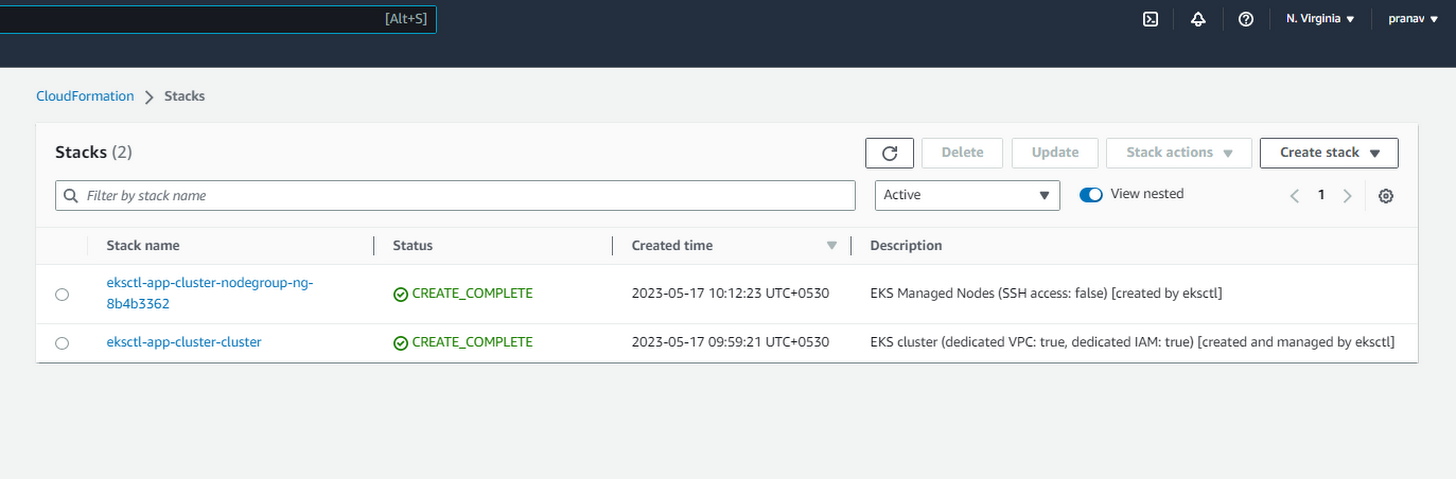


Fig. Stacks in CloudFormation

**Creating a deployment file:**

[root@ip-172-31-89-246 ~]# vi regapp-deployment.yml

[root@ip-172-31-89-246 ~]# cat regapp-deployment.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: app-regapp

labels:

app: regapp

spec:

replicas: 3

selector:

matchLabels:

app: regapp

template:

metadata:

labels:

apps: regapp

spec:

containers:

- name: regapp

image: pranav27/regapp

imagePullPolicy: Always

ports:

- containerPort: 8080

strategy:

type: RollingUpdate

rollingUpdate:

maxSurge: 1

maxUnavailable: 1

**Creating a service file:**

[root@ip-172-31-89-246 ~]# cat regapp-service.yml

apiVersion: v1

kind: Service

metadata:

name: regapp-service

labels:

app: regapp

spec:

selector:

app: regapp

ports:

- port: 8080

targetPort: 8080

type: LoadBalancer

**Integrate Kubernetes Bootstrap Server with Ansible:**

**On Bootstrap Server:**

**Create ansadmin**

[root@ip-172-31-89-246 ~]# useradd ansadmin  
[root@ip-172-31-89-246 ~]# passwd ansadmin

Changing password for user ansadmin.

New password:

BAD PASSWORD: The password is shorter than 8 characters

Retype new password:

passwd: all authentication tokens updated successfully.

[root@ip-172-31-89-246 ~]# visudo

**Add ansadmin to sudoers file**

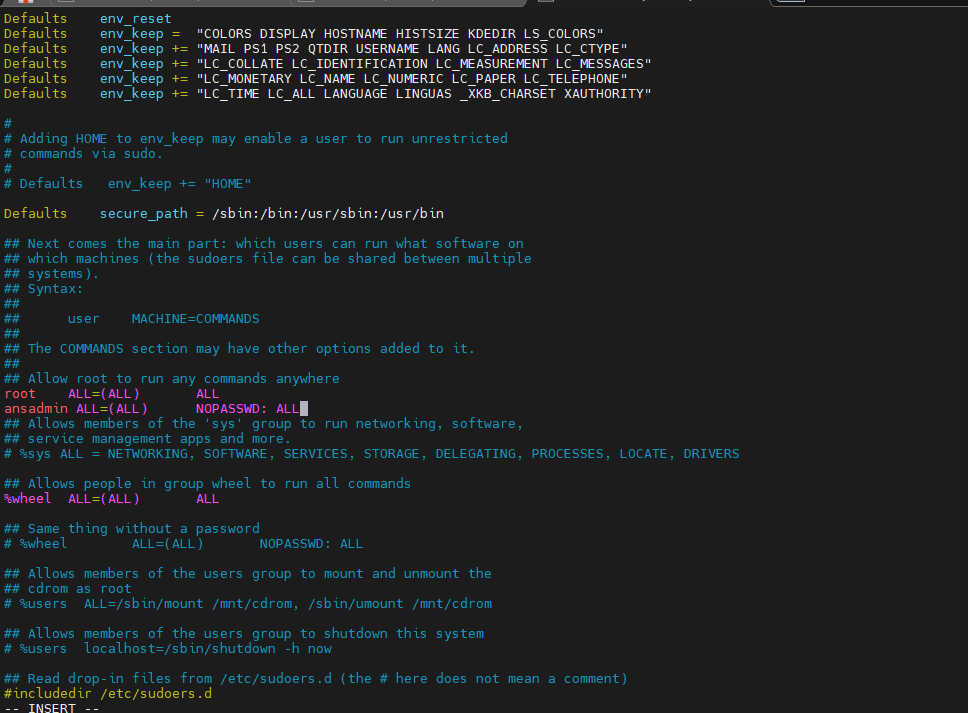


Fig. Add ansadmin to sudoers file

**Enable Password based login**

[root@ip-172-31-89-246 ~]# vi /etc/ssh/sshd\_config

[root@ip-172-31-89-246 ~]# service sshd reload

Redirecting to /bin/systemctl reload sshd.service

**Create Ansible Playbooks for deployment and Service File:**

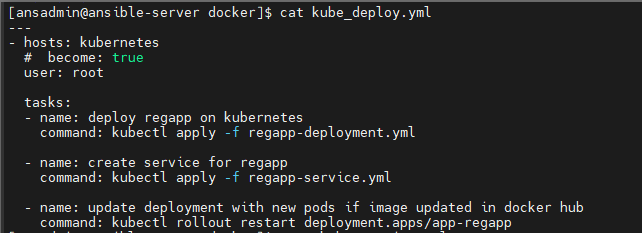


Fig. Kubernetes deploy file created

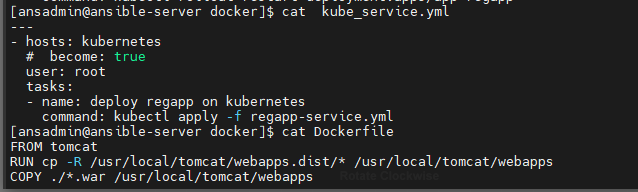
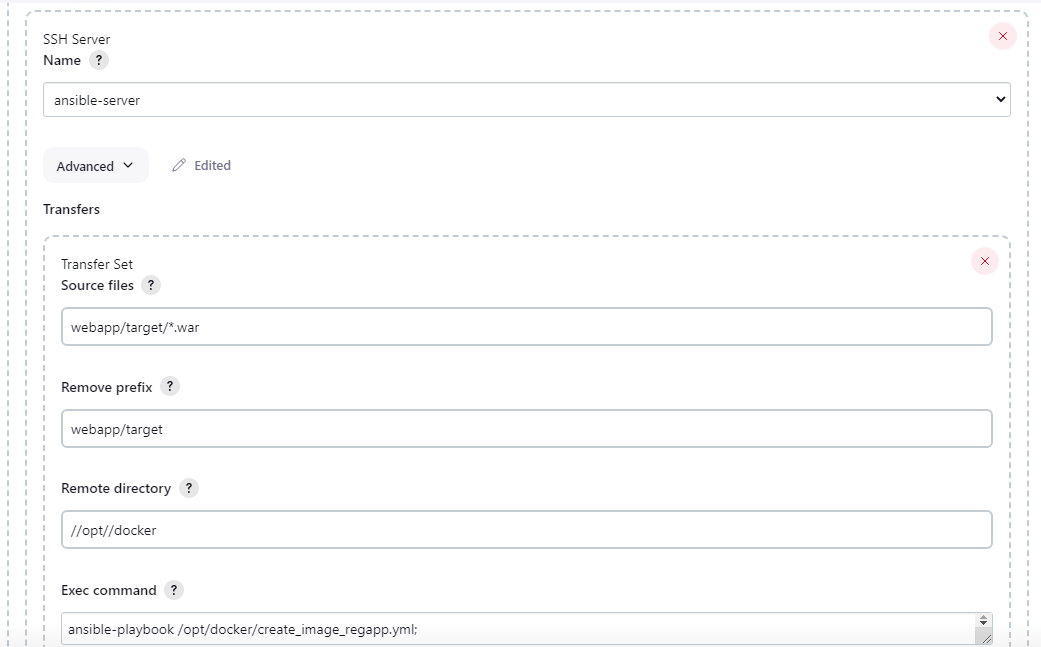
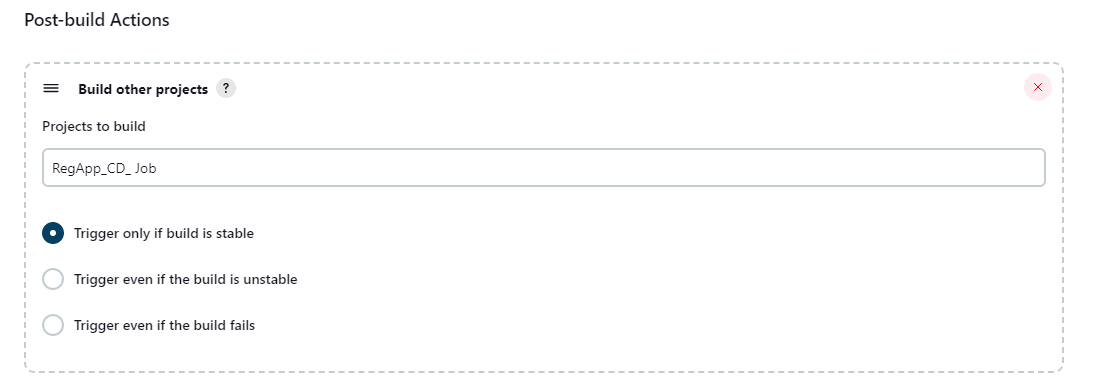
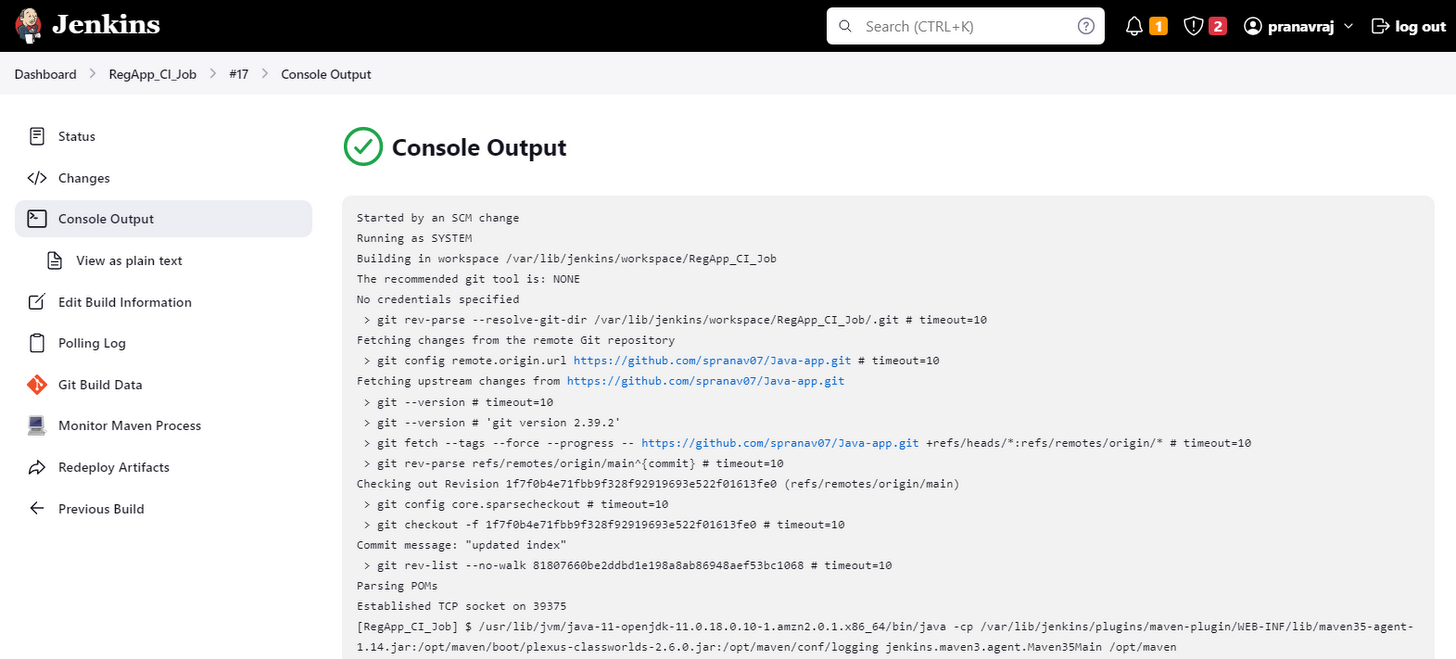


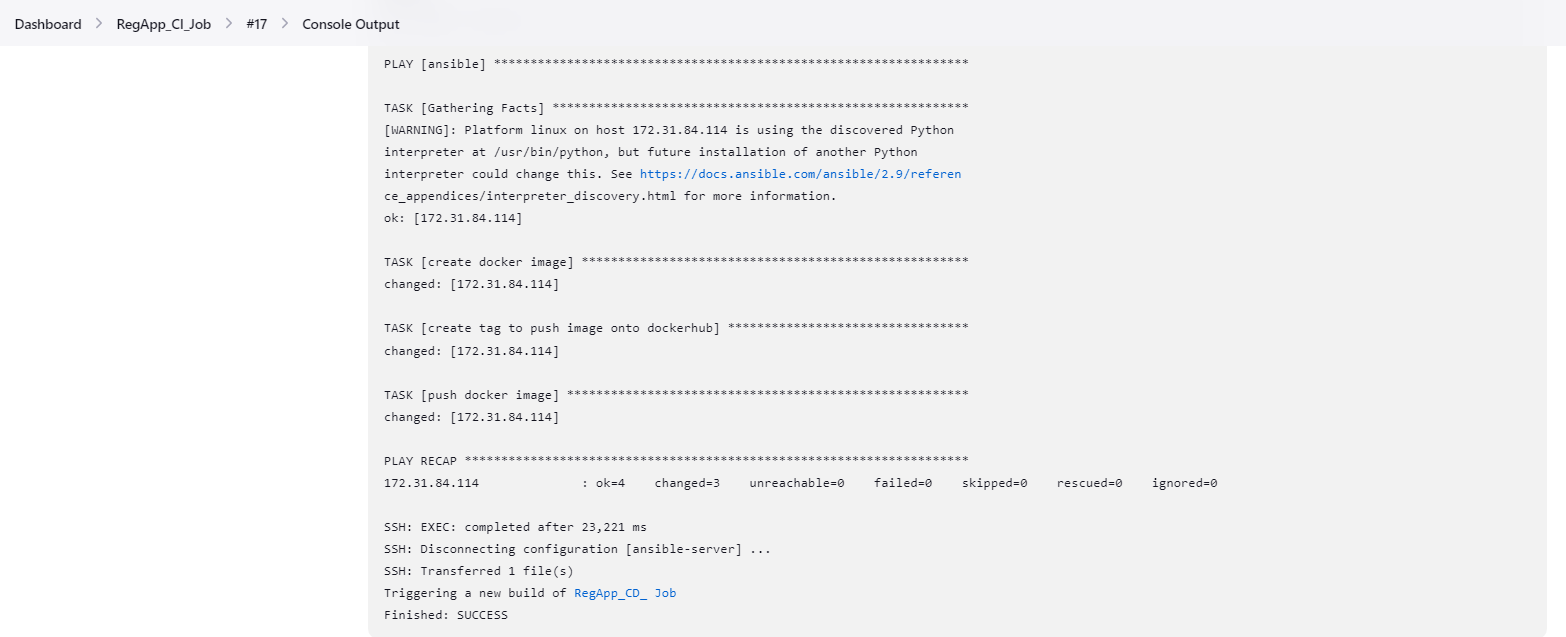
Fig. Kubernetes service file created

**CI Job to create Image for Kubernetes:**

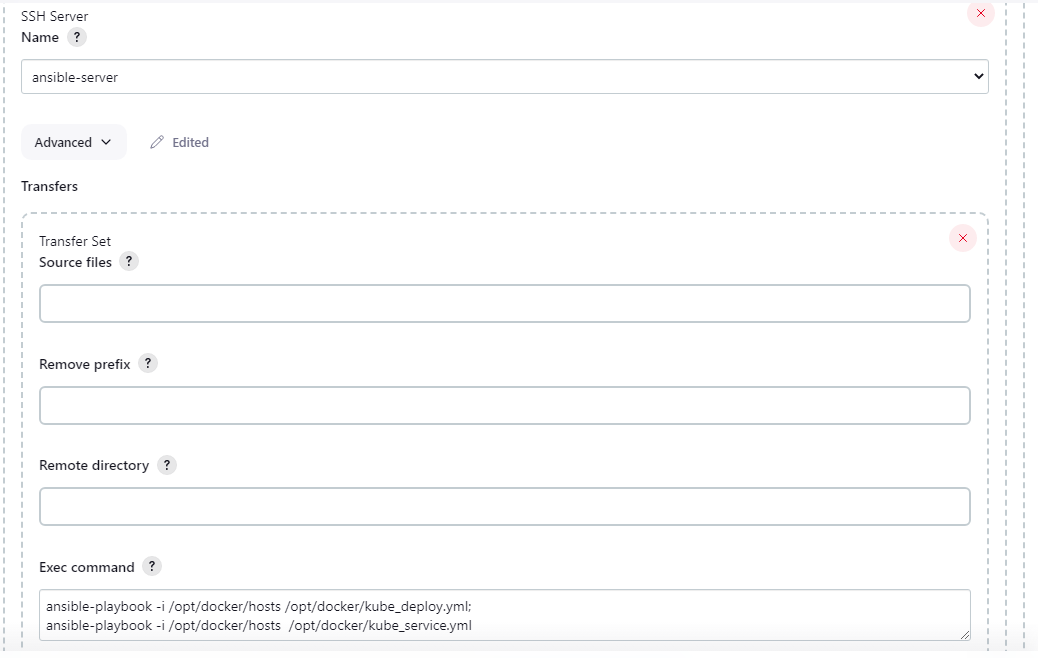
****

****

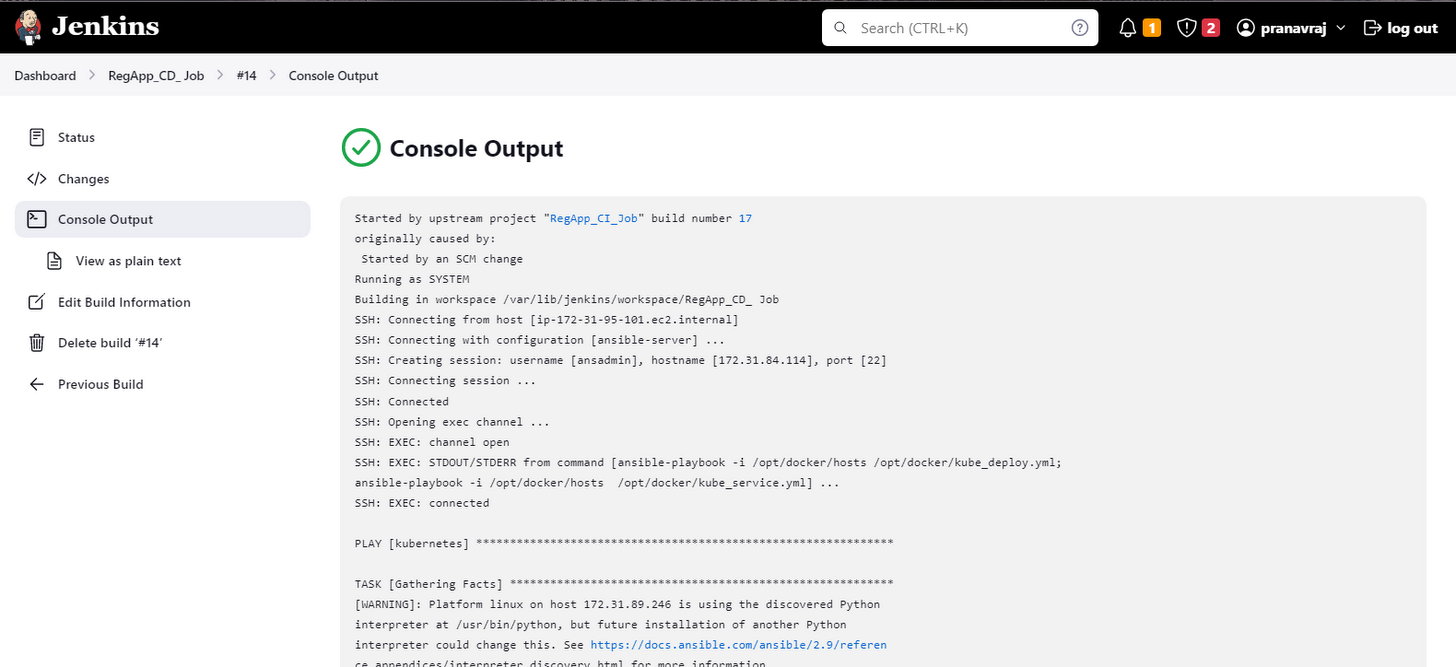


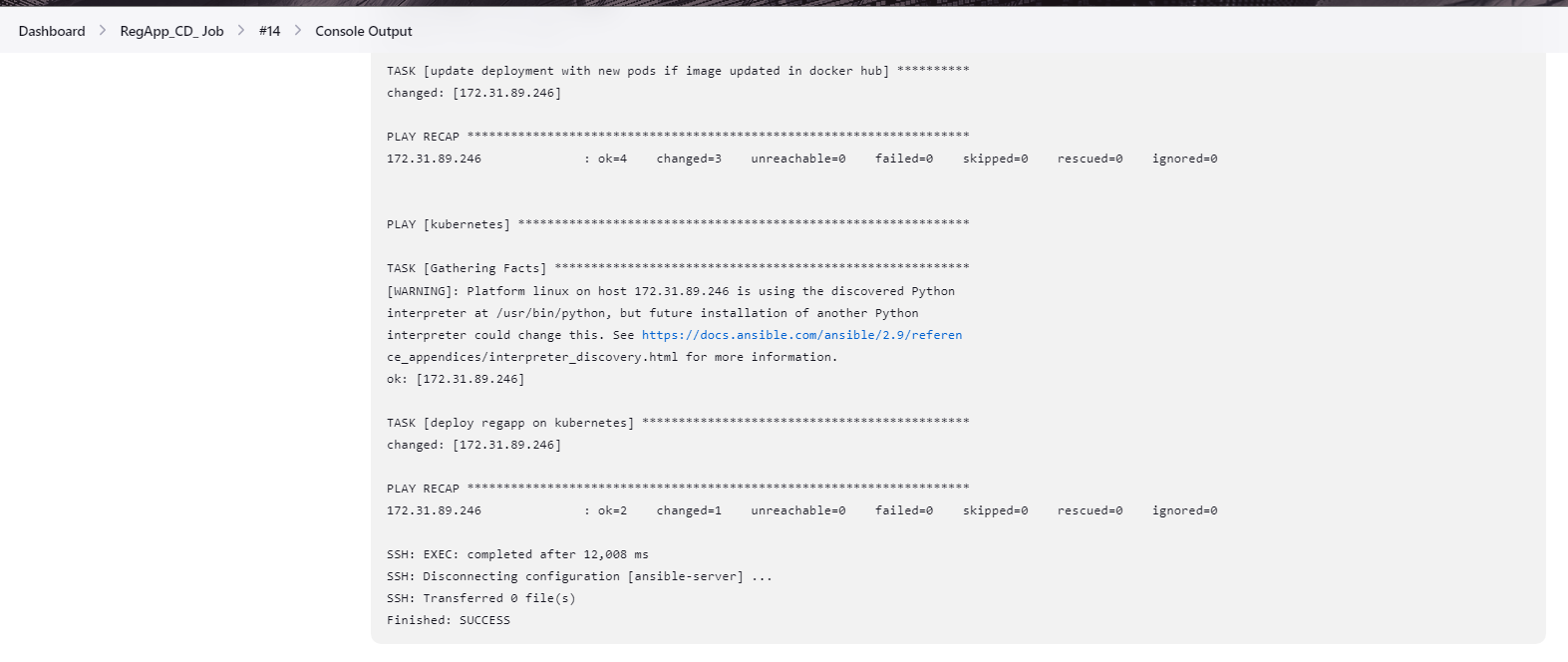


**RegApp\_CD\_ Job**

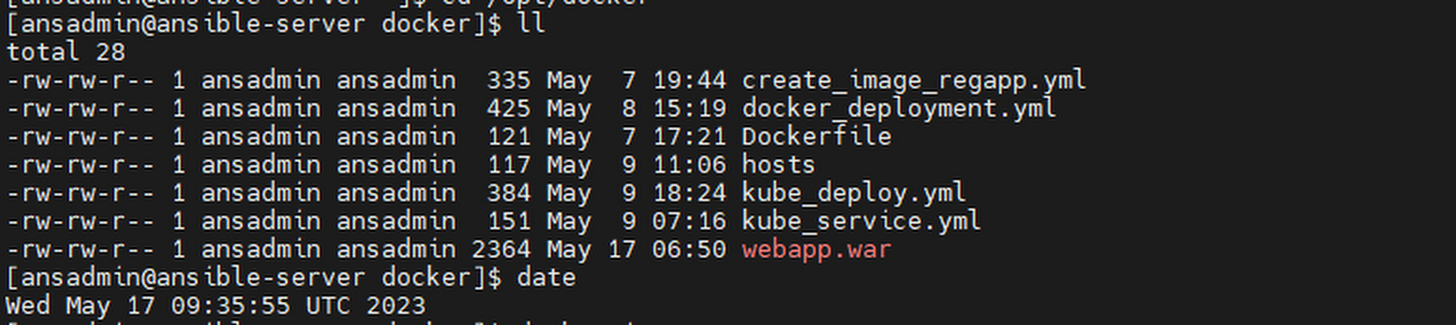
****

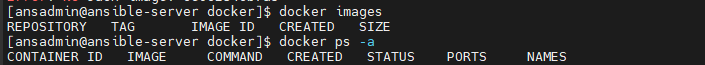
**Complete CI and CD job to build and deploy code on Kubernetes:**





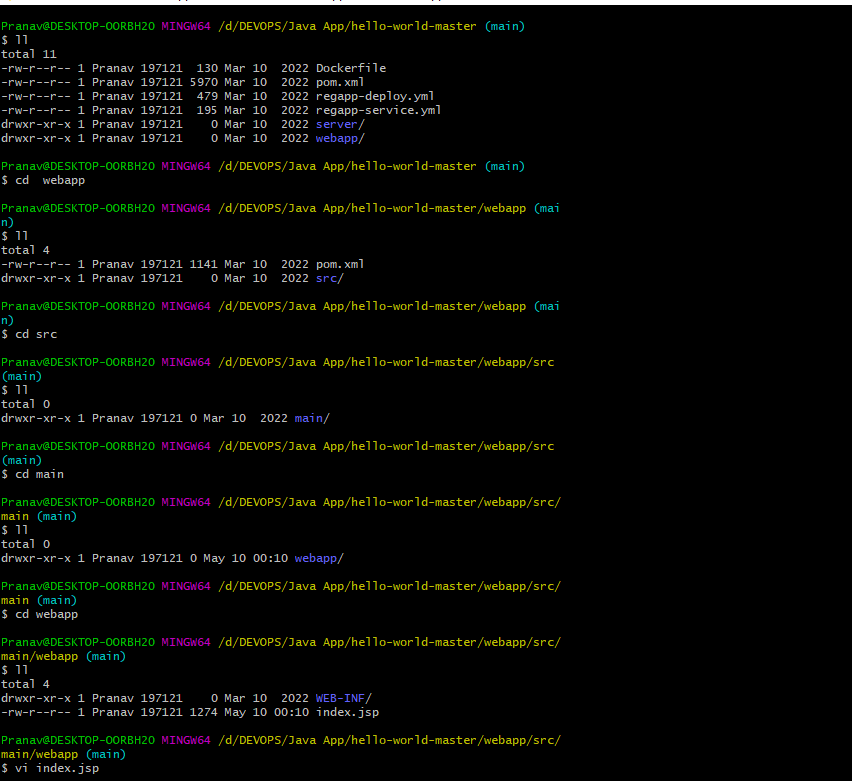
**Output :**

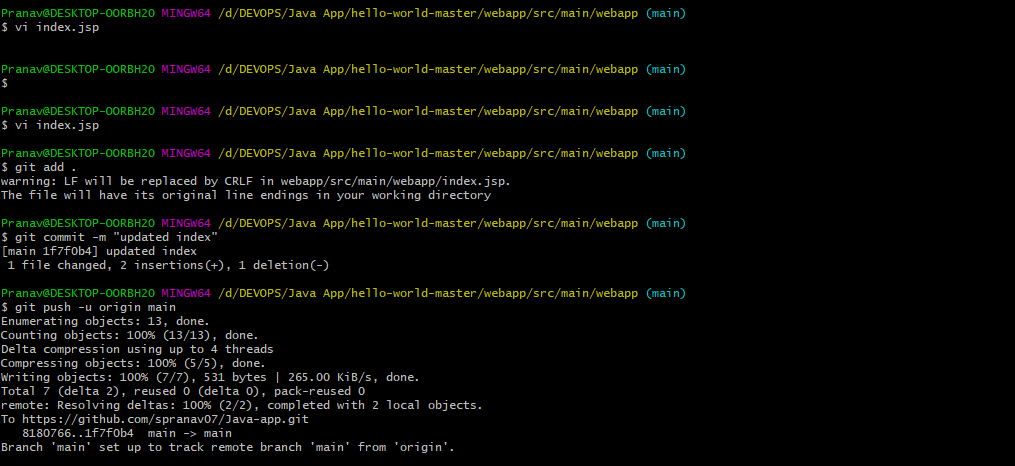
**Before**



**After**

**Changing the source code**

****

****

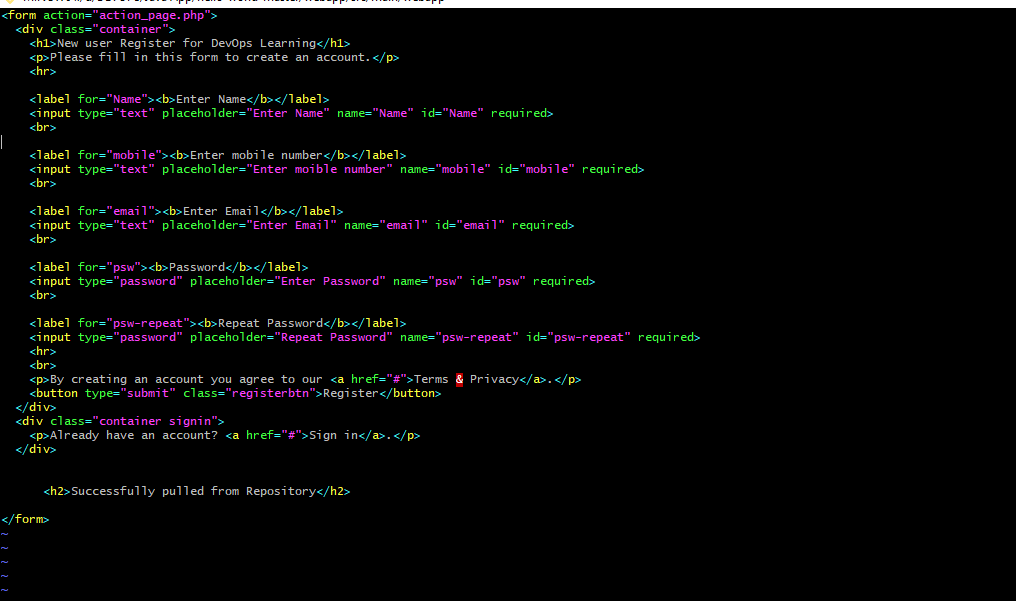
****

Fig.index.isp edited

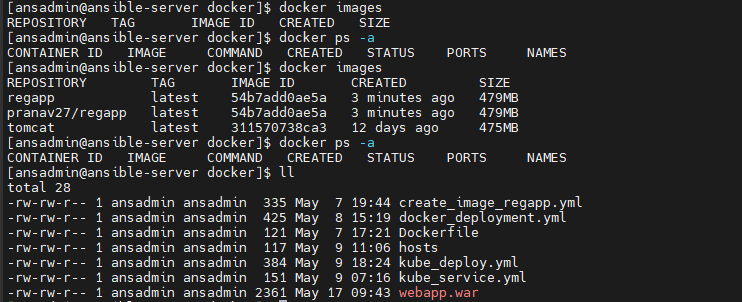


Fig. Ansible Server Webapp.war time change after success



Fig. Bootstrap Server Running

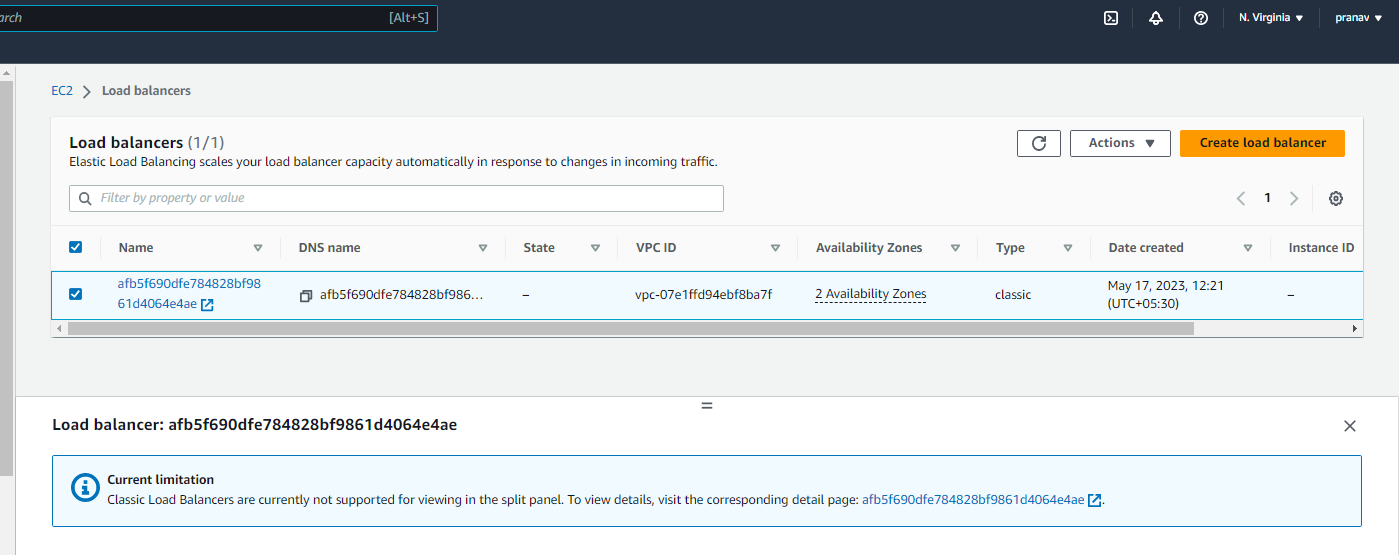


Fig.Load balancer created

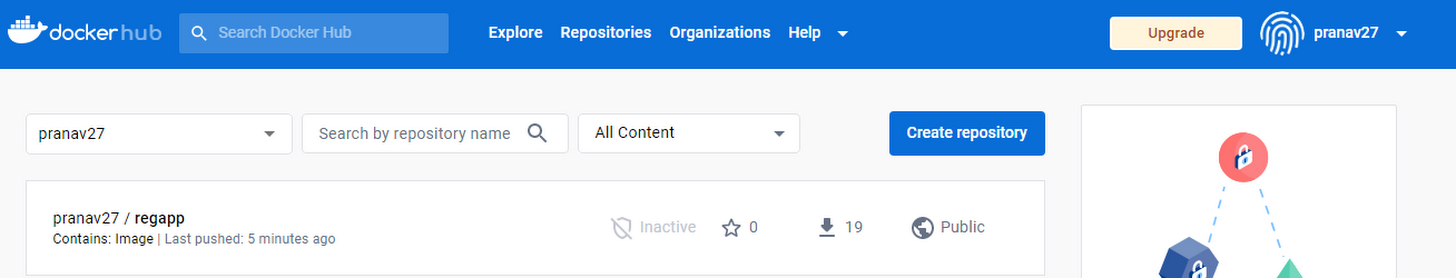


Fig. Docker latest Image

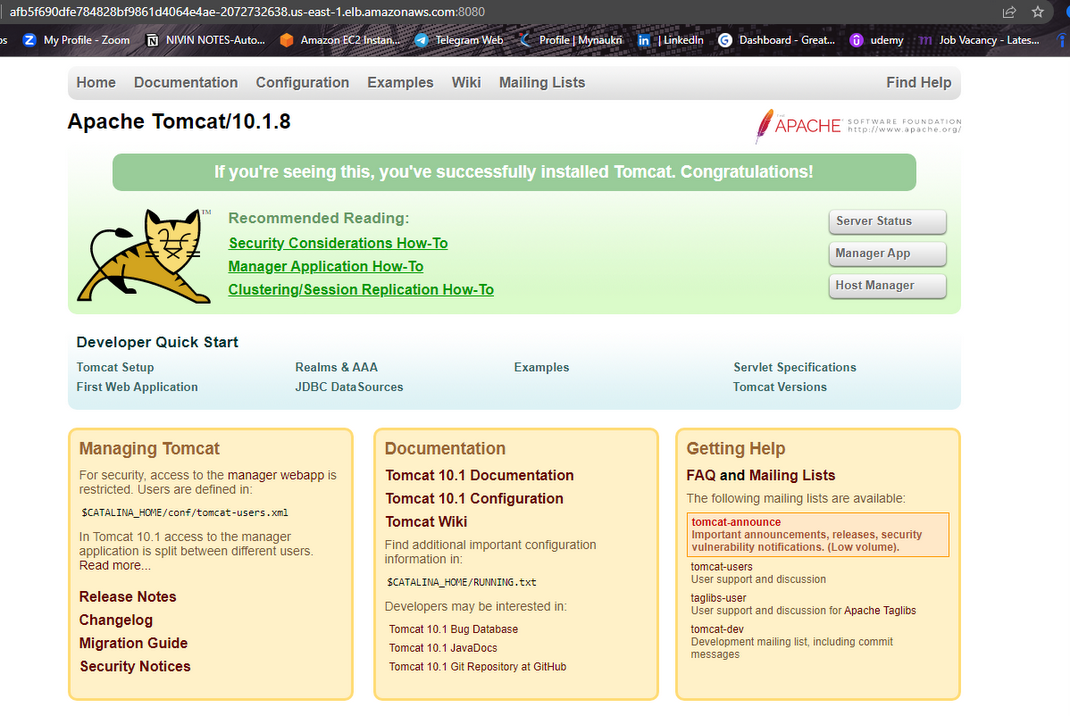


Fig. Access to load balancer on port 8080

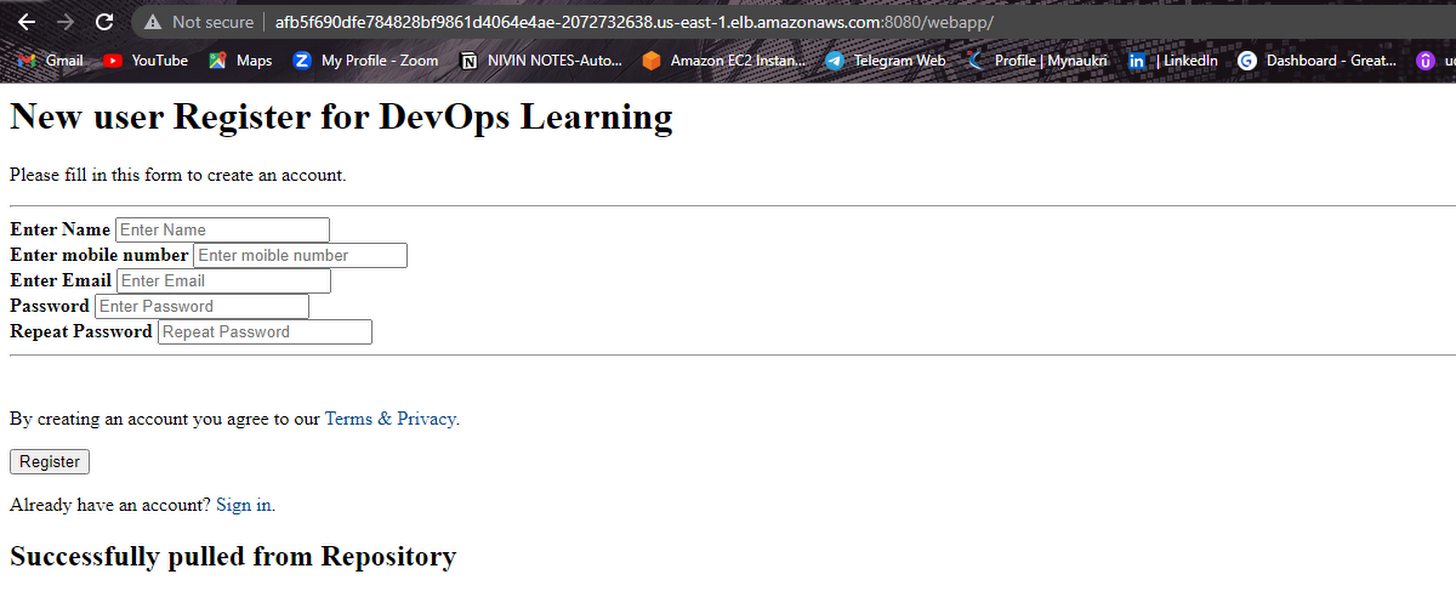
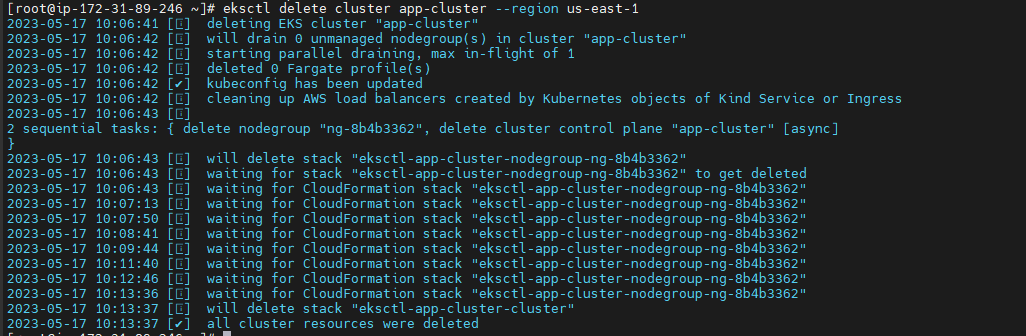


Fig.Latest application

**Cleaning Setup:**



**Deleting Cluster:**



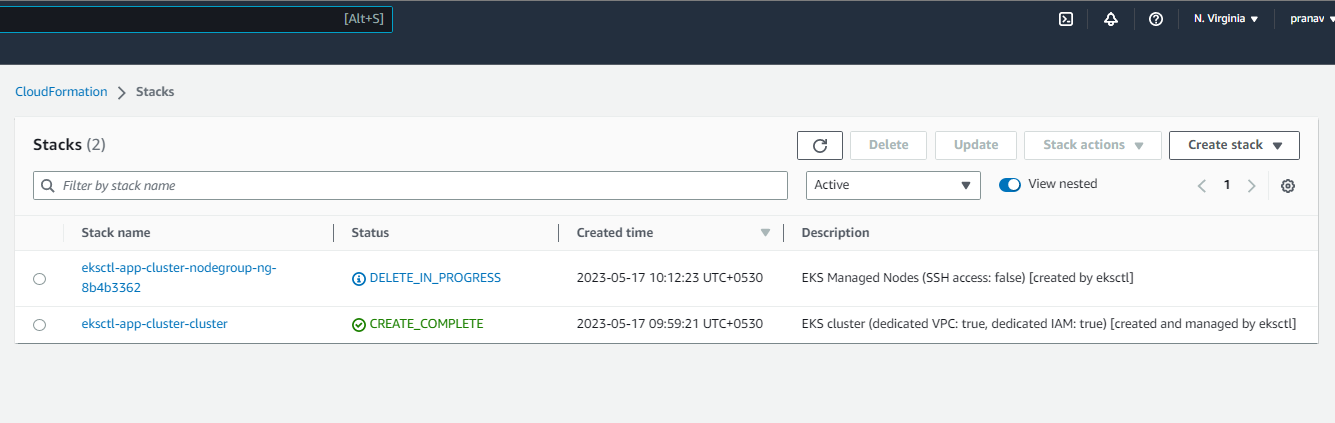


Fig . Cluster Deleted

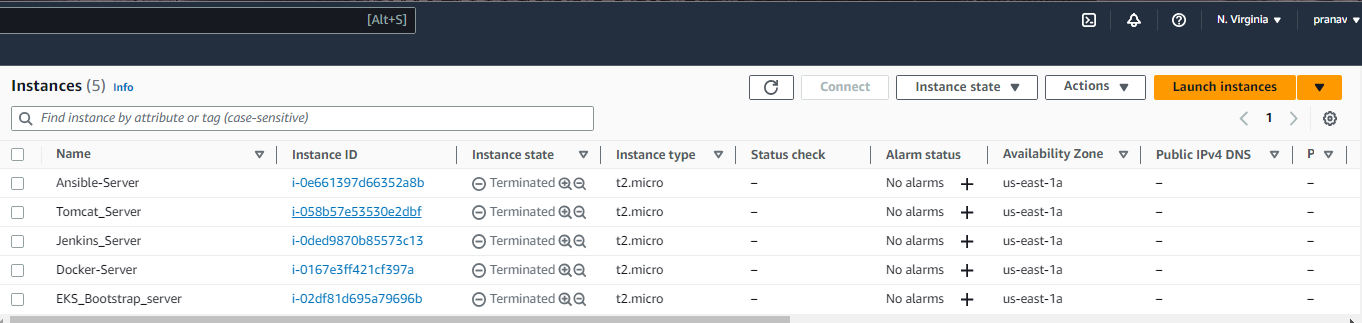


Fig. Terminates all Instances