DevOps Project

Name : Himanshu Vishwakarma
Email : himanshu1vishwakarma@gmail.com

Task :

1) Create a CI/CD Pipeline to Build, Test and Deploy a Website on a testing server then on production servers for a custom webpage.

2) After deploying the web page now I created a monitor service on both the servers to monitor the web server whether they are running or not.

Used Tools : Jenkins, Git & GitHub, Docker, Kubernetes, Selenium, Nagios, Apache Server and for Infrastructure AWS Cloud.

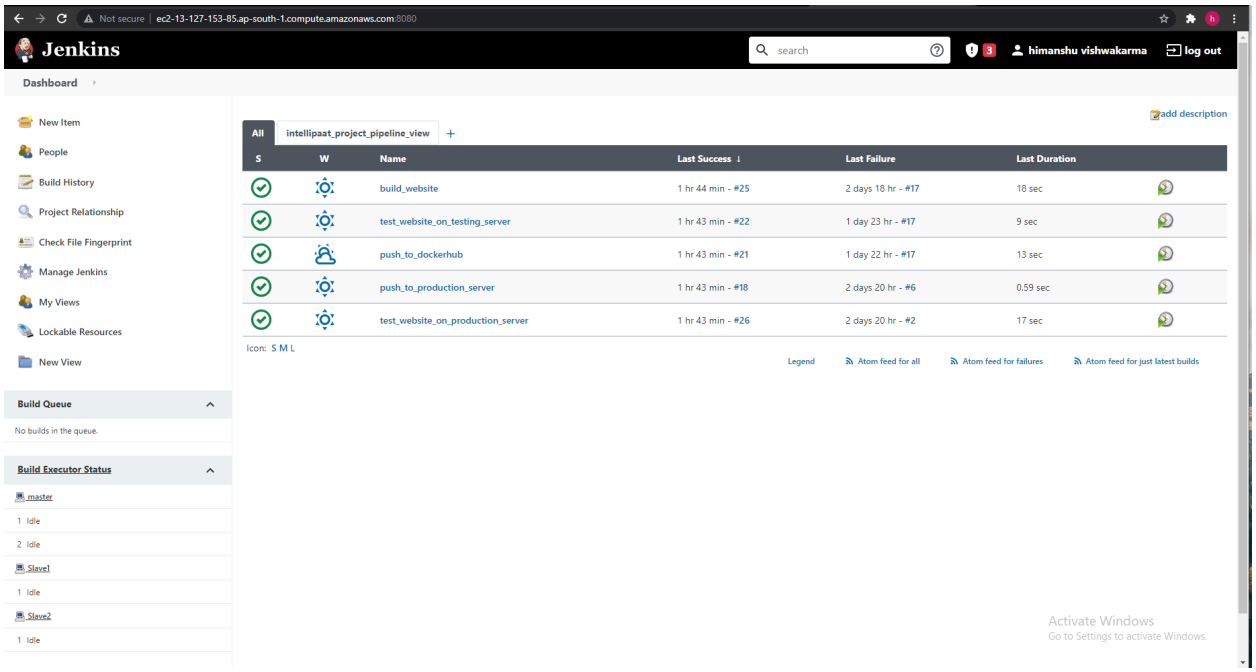
DevOps Project

Explanation of my Project 2 Solution :

1st I create 4 Servers =

1) Server 1 - (Jenkins, Ansible, Nagios) Masters
2) Server 2 - (Jenkins, Ansible, Nagios) Slave & Docker.
3) Server 3 - (Jenkins, Ansible, Nagios) Slave, Kubernetes Master & Docker.
4) Server 4 - Nagios Slave, Kubernetes Worker & Docker.

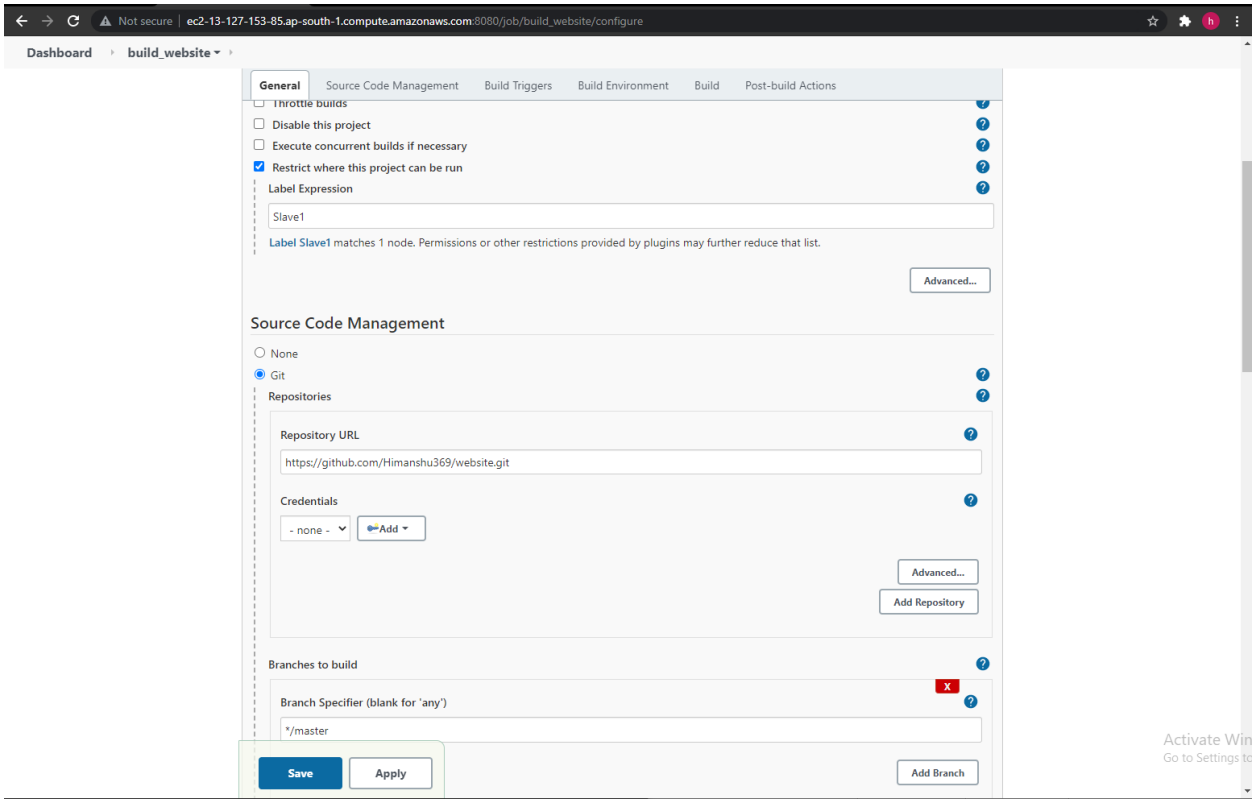
2nd I created Jenkins Pipeline with these Jobs :

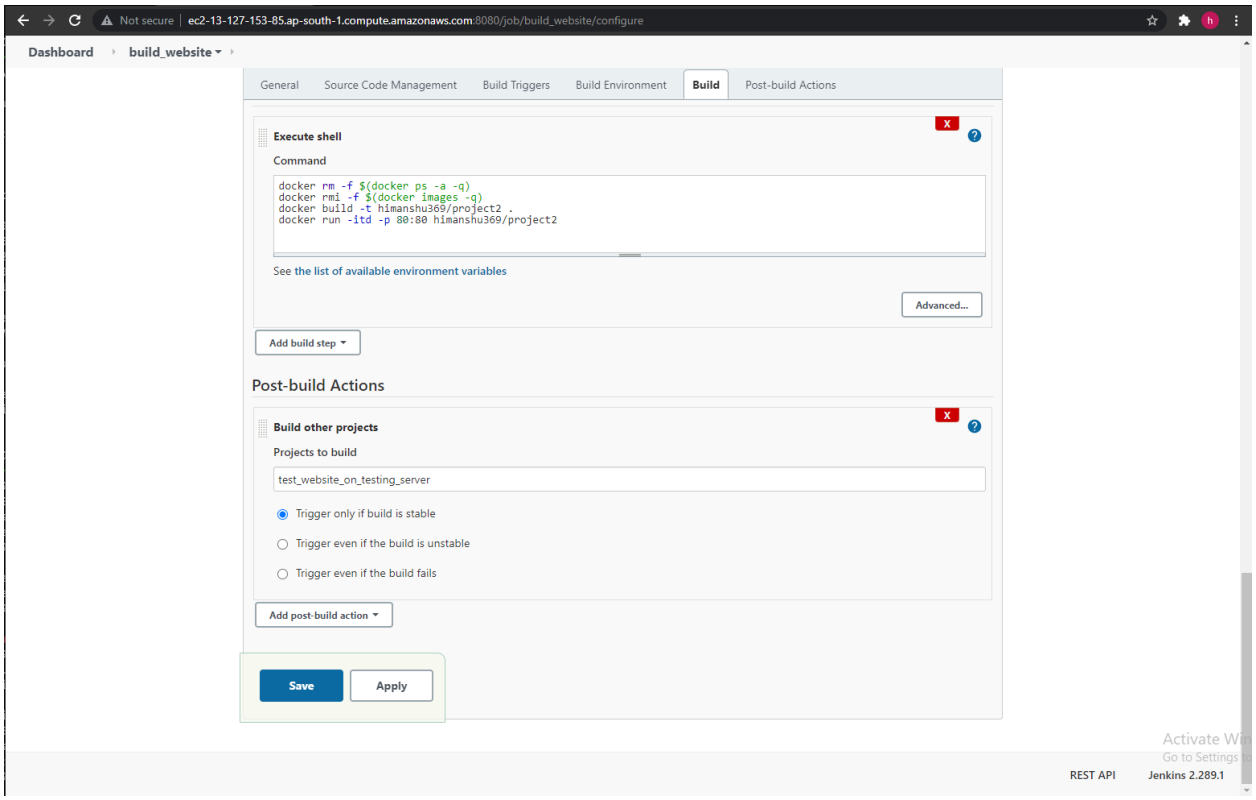


Where Jobs Configurations and there Logs are as :

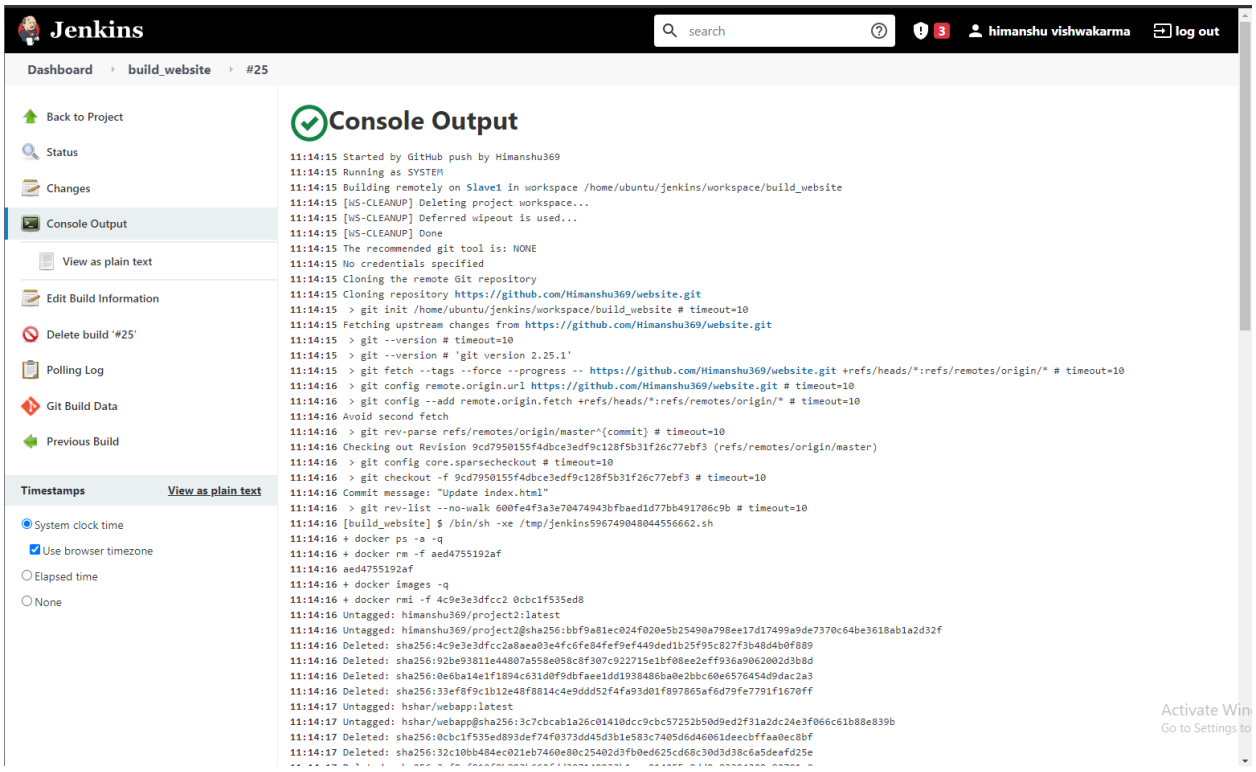
DevOps Project

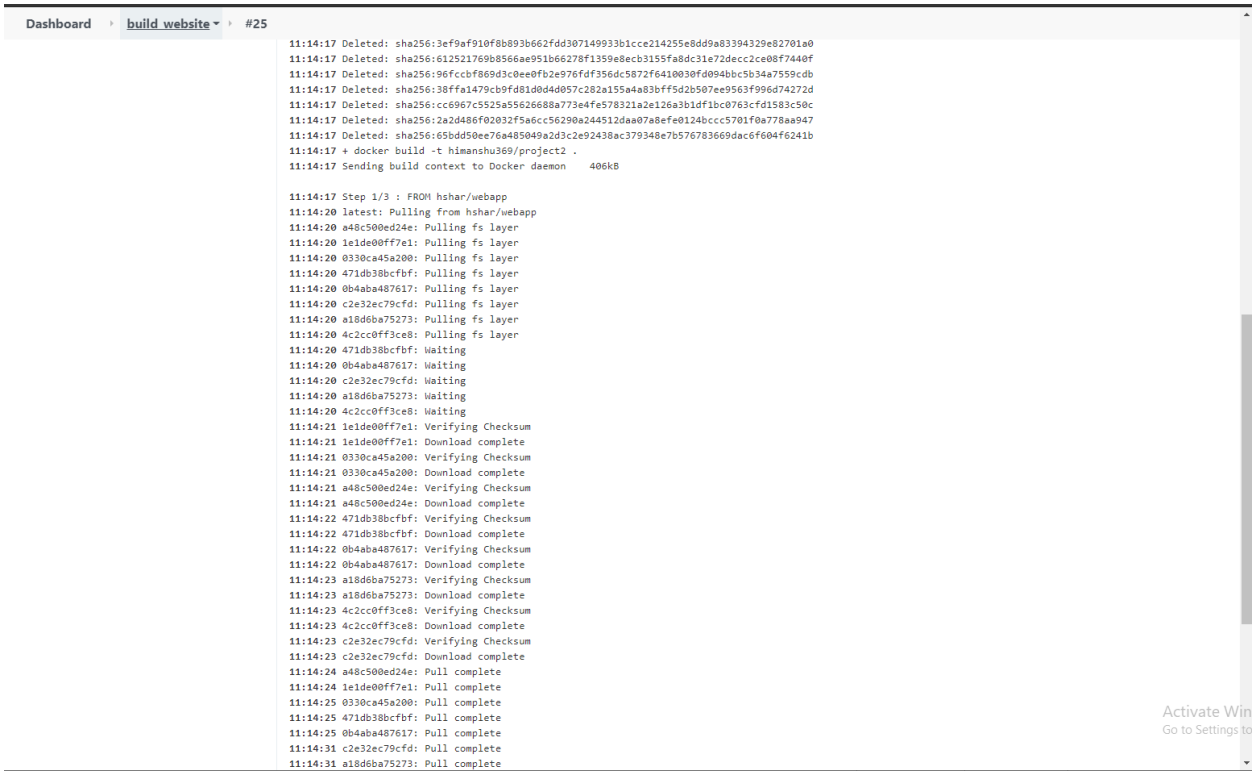
1) build_website =



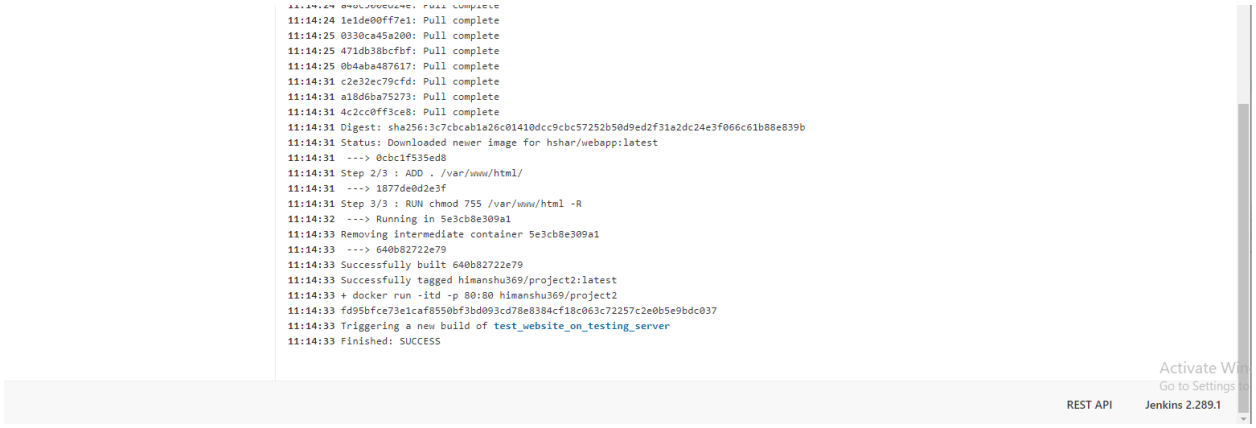


DevOps Project



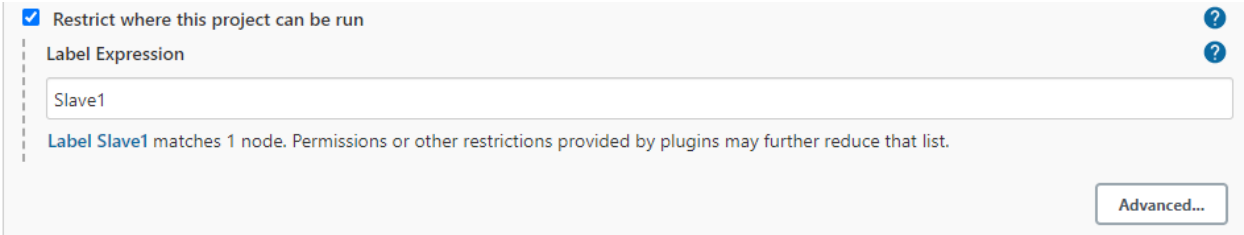


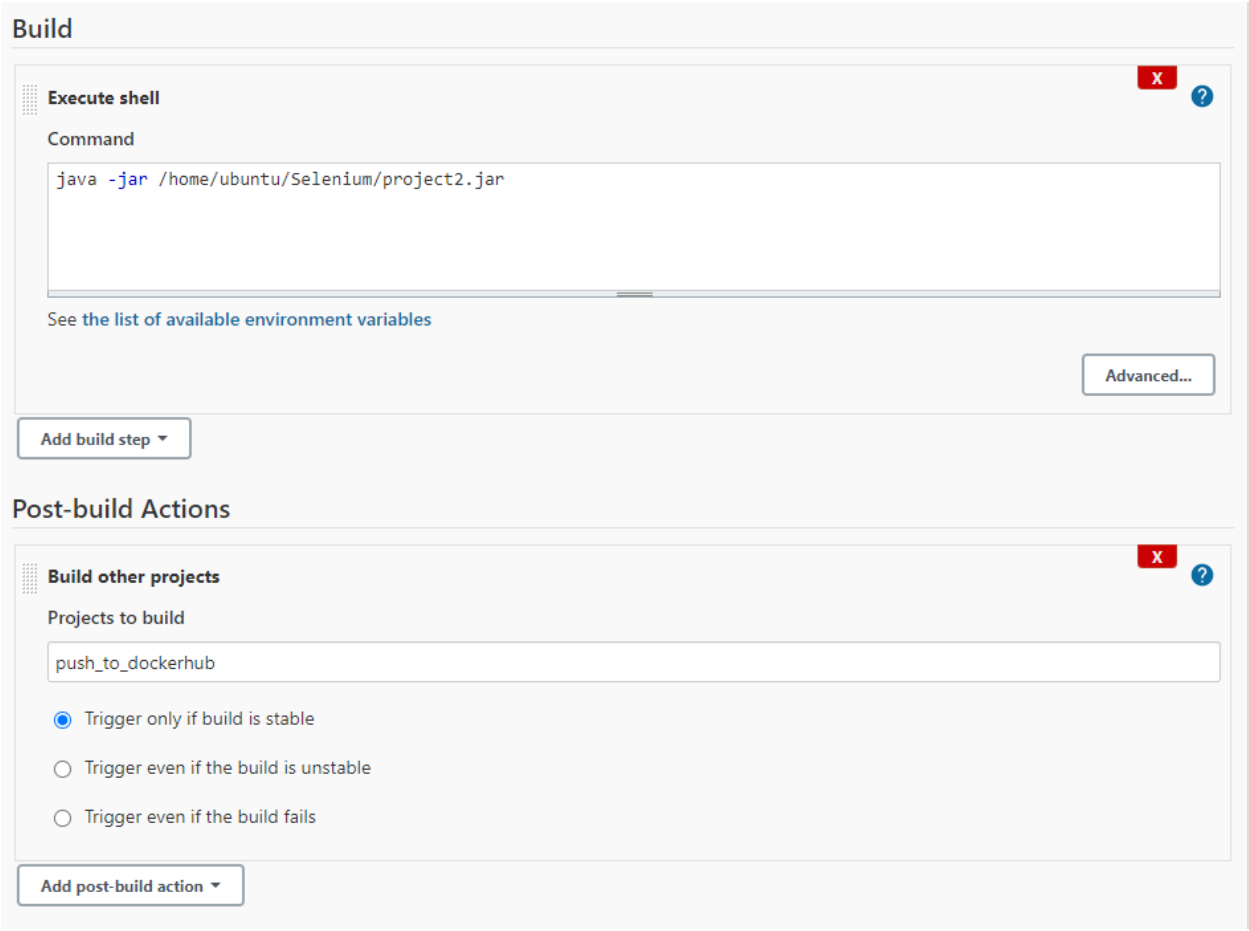
DevOps Project

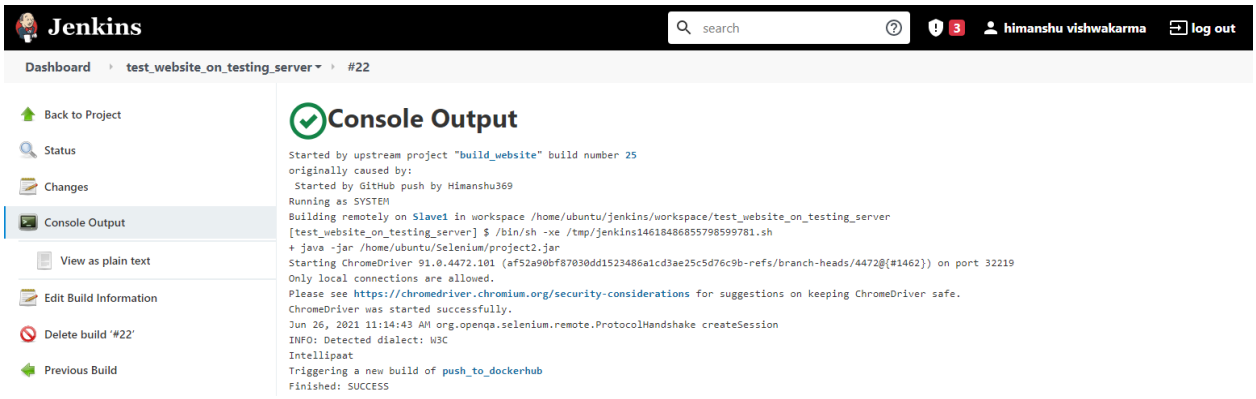


DevOps Project

2) test_website_on_testing_server =

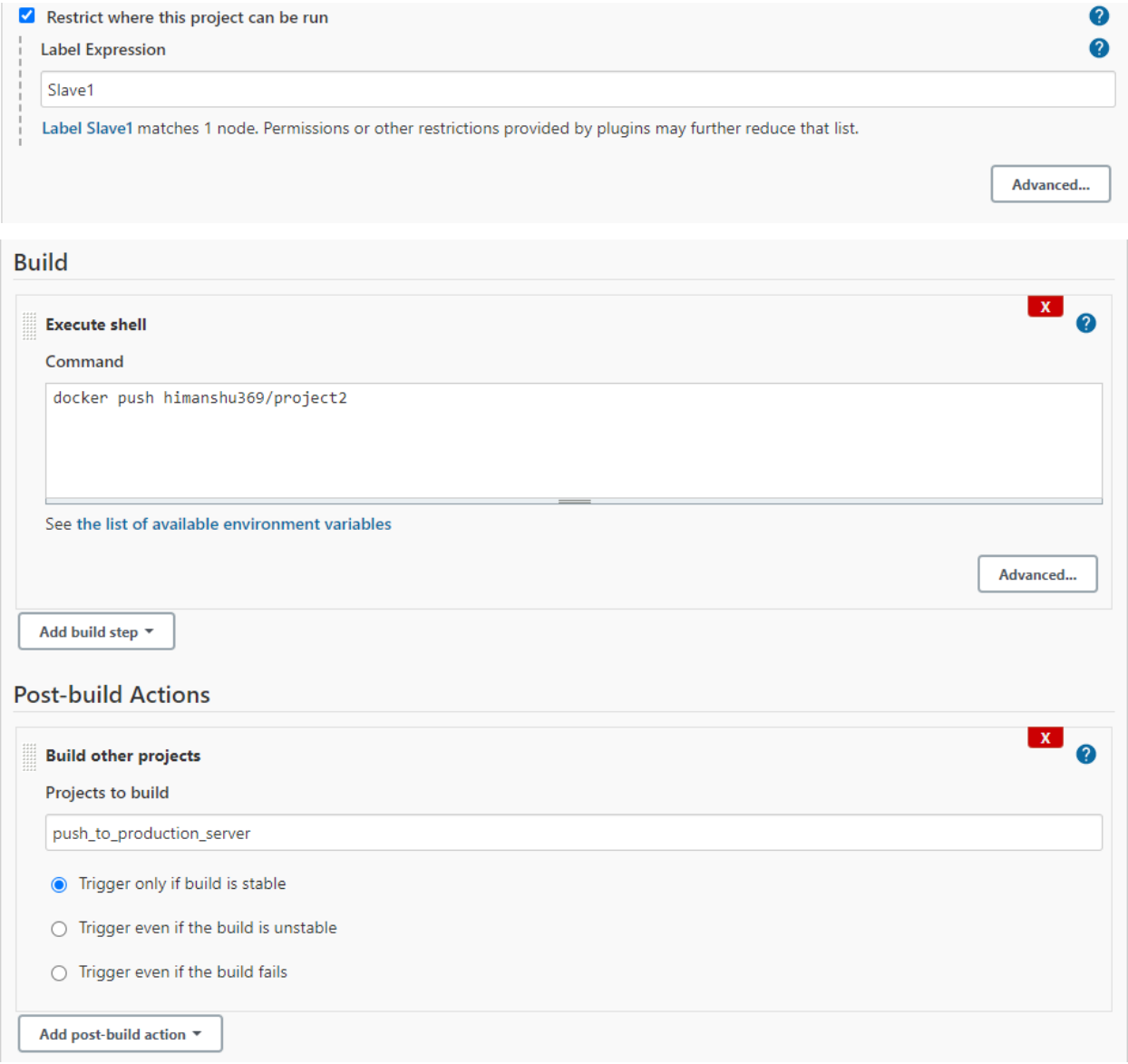




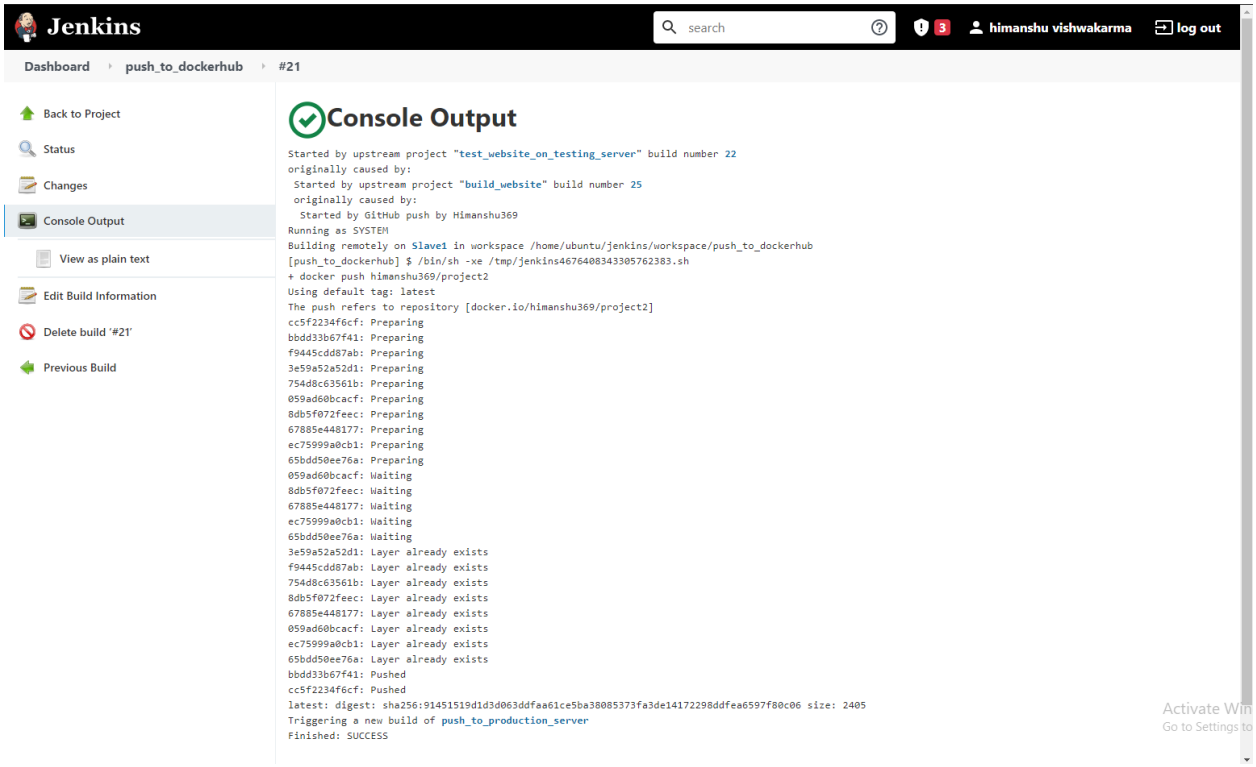


DevOps Project

3) push_to_dockerhub

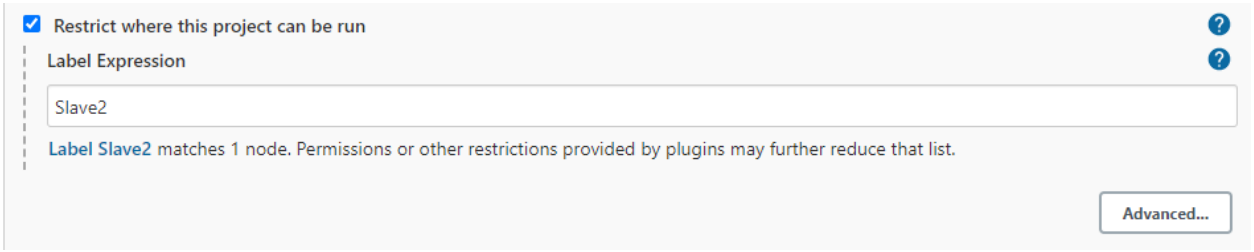


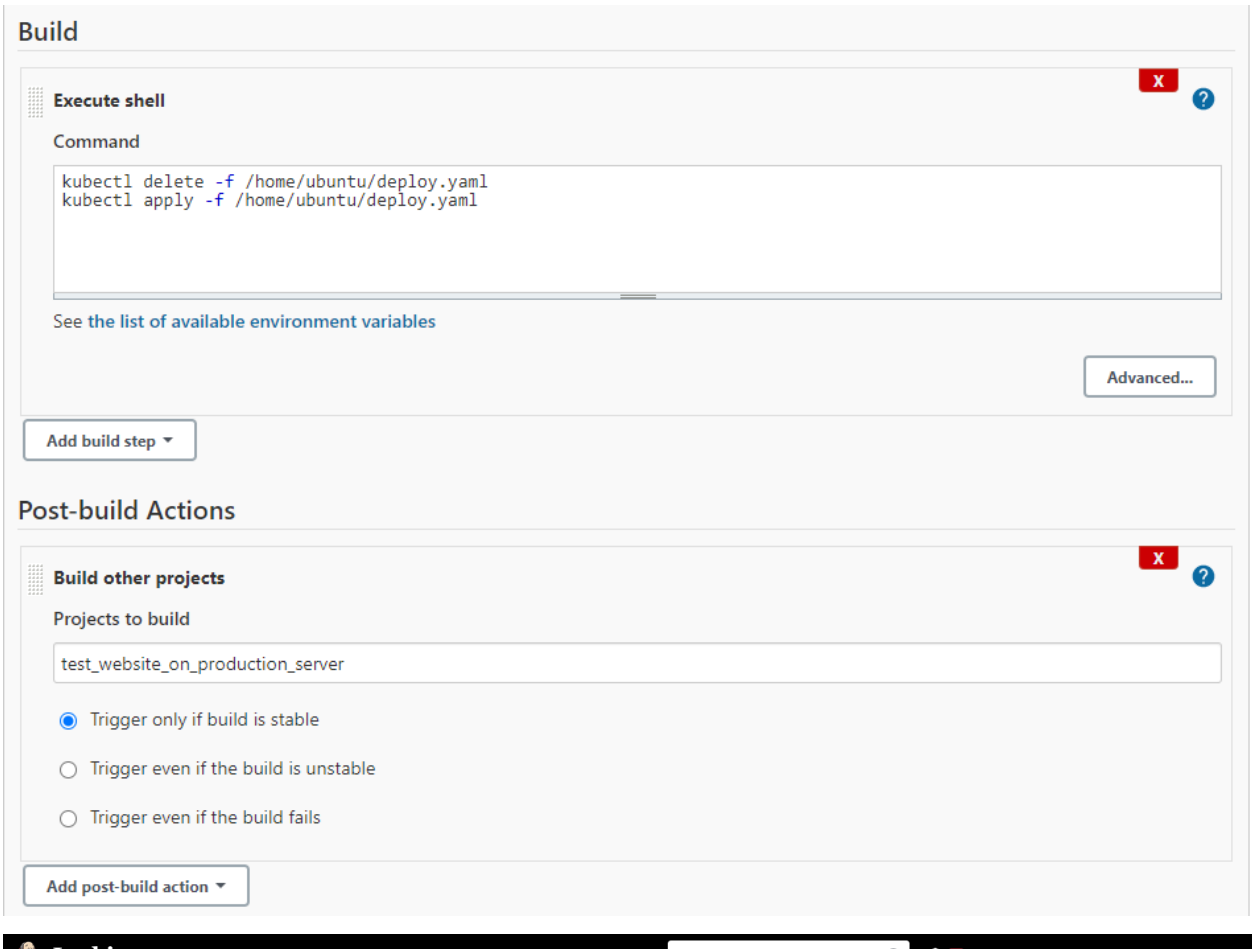
DevOps Project

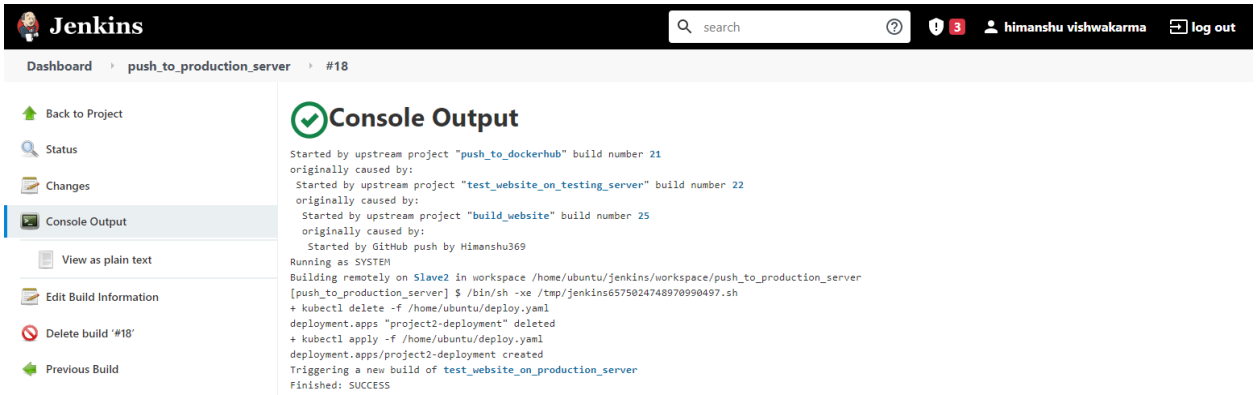


DevOps Project

4) push_to_production_server =

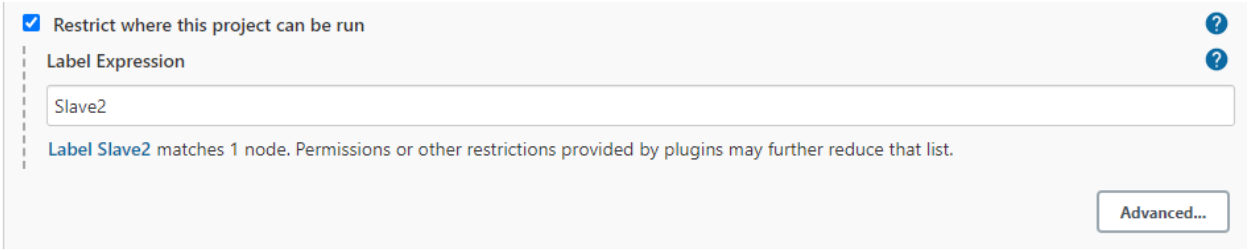


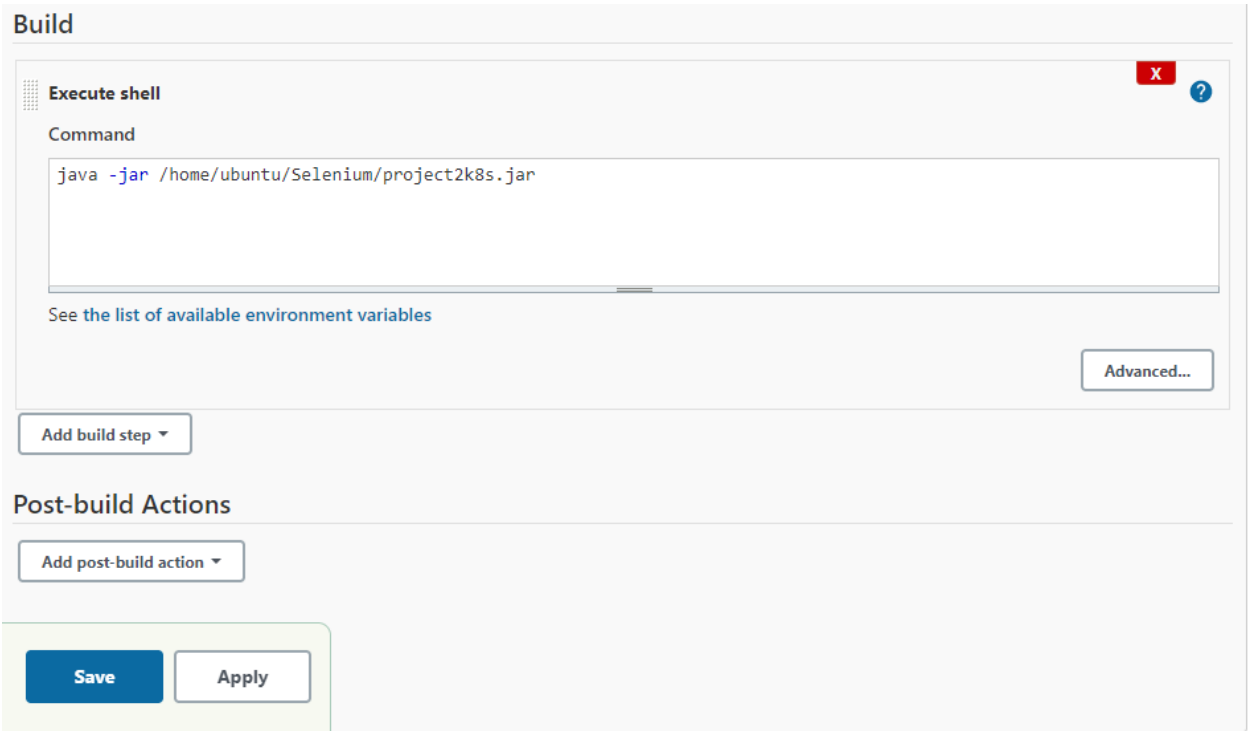


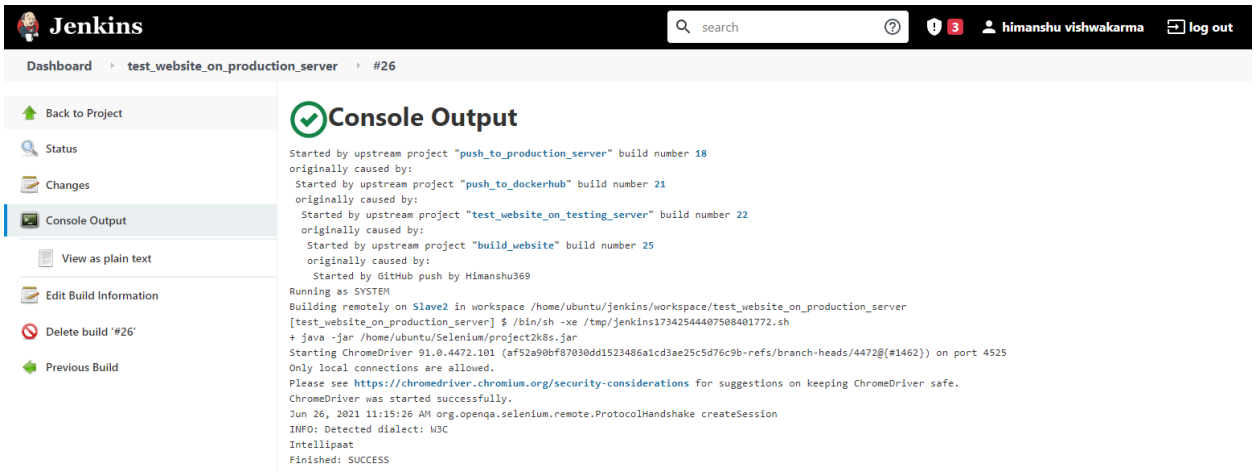


DevOps Project

5) test_website_on_production_server =

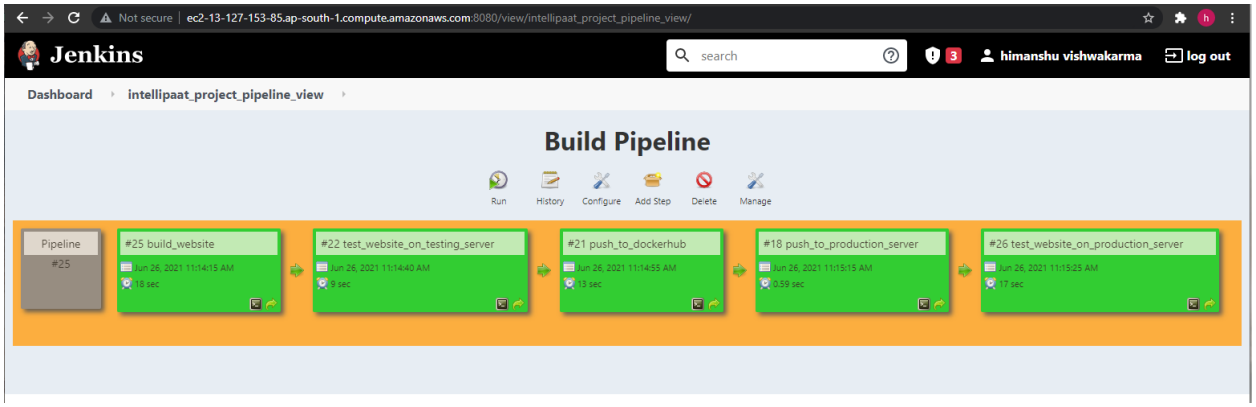






DevOps Project

Jenkins Pipeline View of Project 2 =

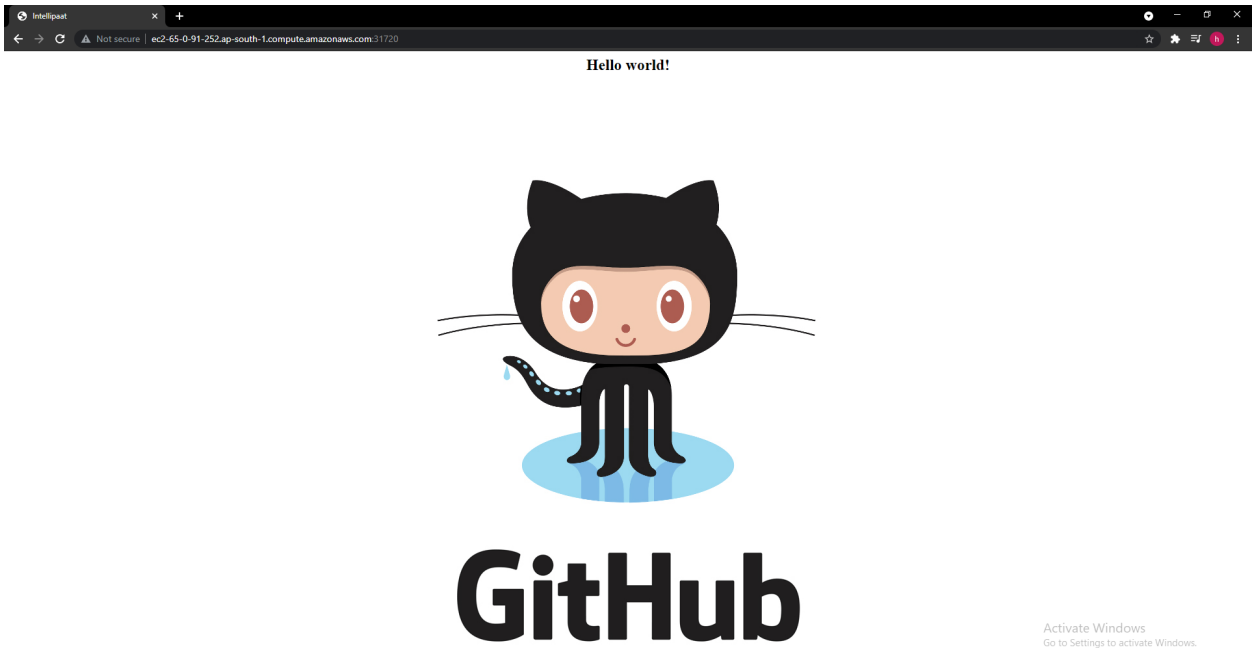


Jenkins Pipeline results :

1) Accessing the WebPage on Testing Server =

DevOps Project

2) Accessing the WebPage on Production Server =



DevOps Project

3rd Kubernetes configurations files :

1) Deployment k8s file and content =

```
root@ip-172-31-39-210:/home/ubuntu# cat deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: project2-deployment
  labels:
    app: project2
spec:
  replicas: 2
  selector:
    matchLabels:
      app: project2
  template:
    metadata:
      labels:
        app: project2
    spec:
      containers:
      - name: project2
        image: himanshu369/project2
        ports:
        - containerPort: 80
root@ip-172-31-39-210:/home/ubuntu#
```

```
root@ip-172-31-39-210:/home/ubuntu# kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
project2-deployment-54c4d86c66-cr76f   1/1     Running   0          24m
project2-deployment-54c4d86c66-d2zvl   1/1     Running   0          24m
root@ip-172-31-39-210:/home/ubuntu#
```

2) I have configured a NodePort service with tcp 80:80 ports =

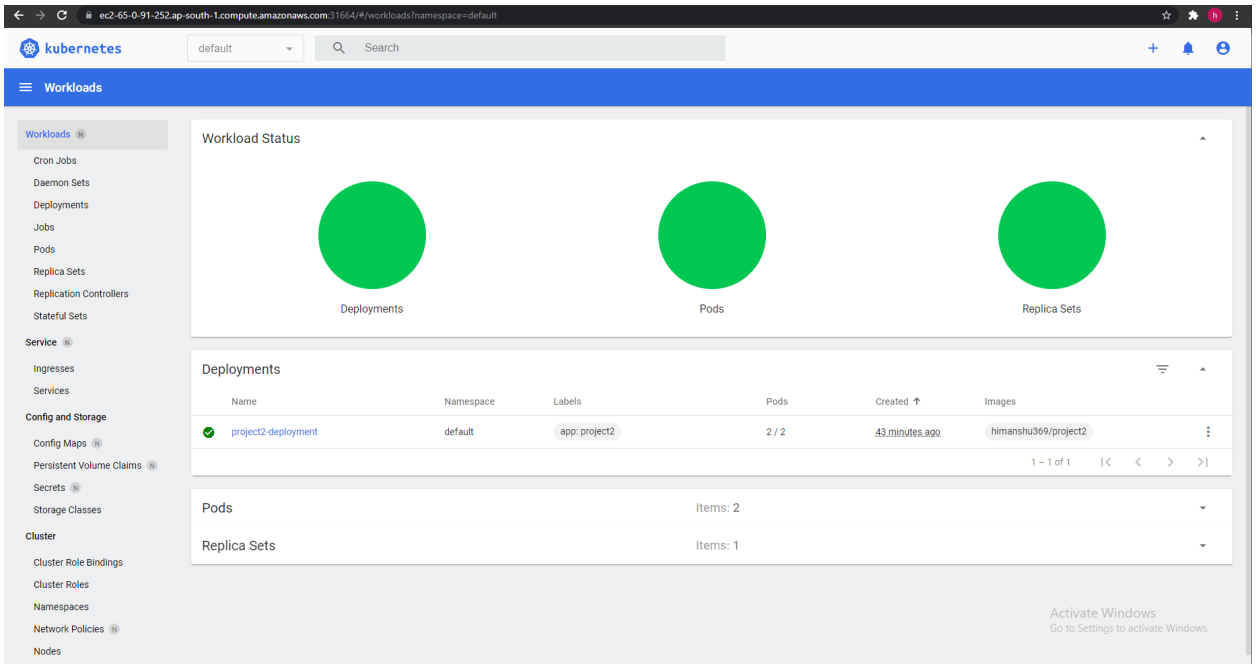
```
root@ip-172-31-39-210:/home/ubuntu# kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        2d20h
project2     NodePort    10.103.178.179   <none>        80:31720/TCP   44h
root@ip-172-31-39-210:/home/ubuntu#
```

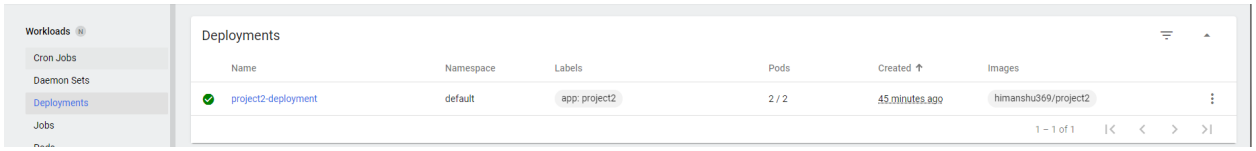
DevOps Project

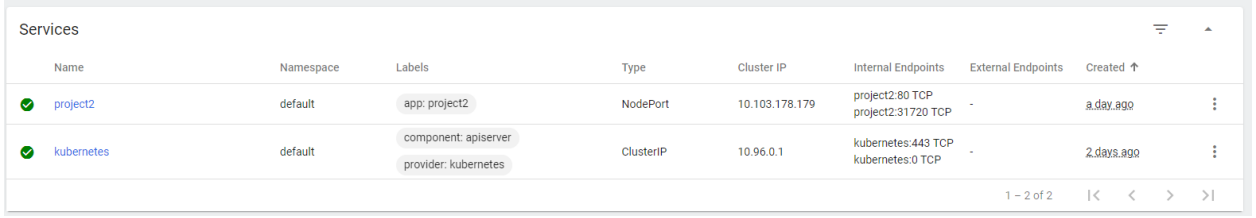
3) Kubernetes Nodes =

```
root@ip-172-31-39-210:/home/ubuntu# kubectl get nodes
NAME                STATUS   ROLES                  AGE     VERSION
ip-172-31-39-210    Ready    control-plane,master   2d20h   v1.21.2
ip-172-31-8-231     Ready    <none>                 2d20h   v1.21.2
root@ip-172-31-39-210:/home/ubuntu#
```

4) Kubernetes Dashboard and Monitoring created Deployment =







DevOps Project

4th Ansible configurations :

1) Ansible Slave 1 and Slave 2 =

```
root@ip-172-31-43-131:/home/ubuntu# ansible -m ping all
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host test_server should use /usr/bin/python3, but
 is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible
release will default to using the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
test_server | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[DEPRECATION WARNING]: Distribution Ubuntu 20.04 on host prod_server should use /usr/bin/python3, but
 is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible
release will default to using the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
prod_server | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

2) Ansible role and content of their files =

```
root@ip-172-31-43-131:/etc/ansible/roles# ls
configfile
root@ip-172-31-43-131:/etc/ansible/roles#
```

```
root@ip-172-31-43-131:/etc/ansible/roles/configfile/tasks# ls
configure.yml  main.yml
root@ip-172-31-43-131:/etc/ansible/roles/configfile/tasks# cat main.yml
---
# tasks file for configfile
 - include: configure.yml
root@ip-172-31-43-131:/etc/ansible/roles/configfile/tasks# cat configure.yml
---
 - name: server.config file
   copy: src=config.txt dest=/home/ubuntu/

root@ip-172-31-43-131:/etc/ansible/roles/configfile/tasks#
```

DevOps Project

```
root@ip-172-31-43-131:/etc/ansible/roles/configfile/files# cat config.txt
This is the configuration file for Production and Testing Servers.
root@ip-172-31-43-131:/etc/ansible/roles/configfile/files#
```

```
root@ip-172-31-43-131:/etc/ansible# ls
ansible.cfg  hosts  playbook.yaml  roles
root@ip-172-31-43-131:/etc/ansible# cat playbook.yaml
---
 - hosts: group
   roles:
    - configfile
root@ip-172-31-43-131:/etc/ansible#
```

3) After running the “ansible-playbook playbook.yaml” command on Ansible Master then the results on Ansible Slaves =

Slave 1

```
root@ip-172-31-12-142:/home/ubuntu# cat config.txt
This is the configuration file for Production and Testing Servers.
```

Slave 2

```
root@ip-172-31-39-210:/home/ubuntu# cat config.txt
This is the configuration file for Production and Testing Servers.
root@ip-172-31-39-210:/home/ubuntu#
```

DevOps Project

5th Nagios Configurations :

```
root@ip-172-31-43-131:/usr/local/nagios/etc/objects# ls
commands.cfg  localhost.cfg  printer.cfg  slave2.cfg  templates.cfg    windows.cfg
contacts.cfg  nano.save      slave1.cfg   switch.cfg  timeperiods.cfg
root@ip-172-31-43-131:/usr/local/nagios/etc/objects#
```

1) Content of slave1.cfg file

```
  GNU nano 4.8                                  slave1.cfg
define host {

    use                     linux-server
    host_name               slave1
    alias                   linuxhost
    address                 13.126.187.183
}


define service {

    use                     generic-service
    host_name               slave1
    service_description     HTTP
    check_command           check_http
    notifications_enabled   1
    check_interval          1
    contacts                Himanshu
    notification_interval   1
    notification_period     24x7
}
```

2) Content of slave2.cfg file

```
  GNU nano 4.8                                  slave2.cfg
define host {

    use                     linux-server
    host_name               slave2
    alias                   linuxhost
    address                 65.0.91.252
}


define service {

    use                     generic-service
    host_name               slave2
    service_description     HTTP
    check_command           check_http_slave2
    notifications_enabled   1
    check_interval          1
    _port_number            31720
    contacts                Himanshu
    notification_interval   1
    notification_period     24x7
}
```

DevOps Project

3) Content of edited commands.cfg file

```
  GNU nano 4.8                                        commands.cfg

define command {

    command_name    check_http_slave2
    command_line    $USER1$/check_http -I $HOSTADDRESS$ -p 31720 $ARG1$
}


define command {

    command_name    check_http
    command_line    $USER1$/check_http -I $HOSTADDRESS$ $ARG1$
}
```

4) Content of contacts.cfg file

```
  GNU nano 4.8                                  contacts.cfg
##############################################################################

# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contact temp>
    alias                   Nagios Admin            ; Full name of user
    email                   himanshu1vishwakarma@gmail.com        ; <<***** CHANGE THIS TO YOUR EMAIL >
}

define contact {

    contact_name            Himanshu            ; Short name of user
    use                     generic-contact         ; Inherit default values from generic-contact temp>
    alias                   Himanshu            ; Full name of user
    email                   himanshu1vishwakarma@gmail.com        ; <<***** CHANGE THIS TO YOUR EMAIL >
    host_notification_commands     notify-host-by-email
    host_notification_options      d,u,r
    host_notification_period       24x7
    service_notification_commands  notify-service-by-email
    service_notification_options   w,u,c,r
    service_notification_period    24x7
}
```

DevOps Project

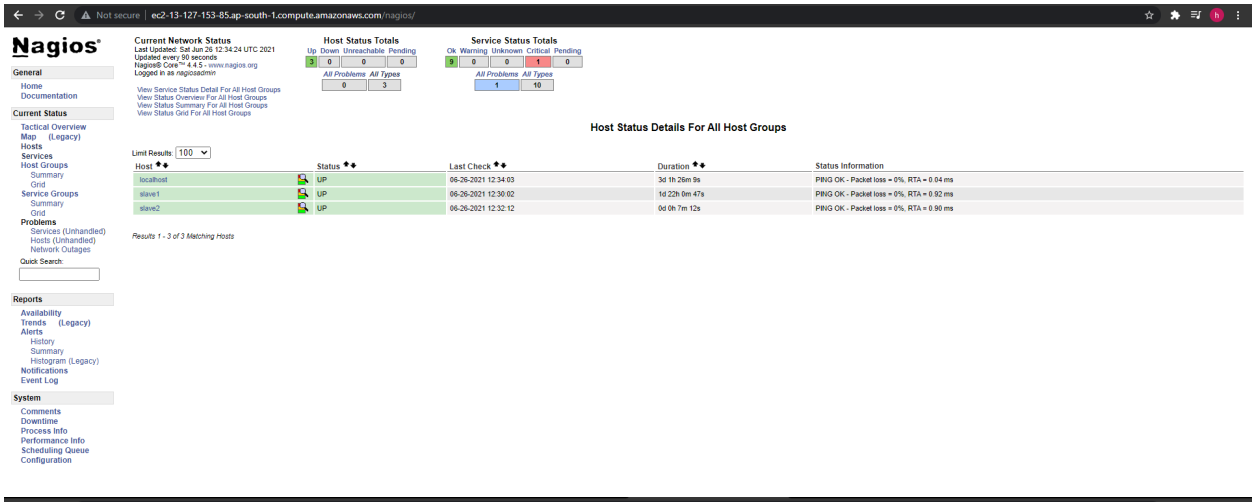
5) Content of nagios.cfg file

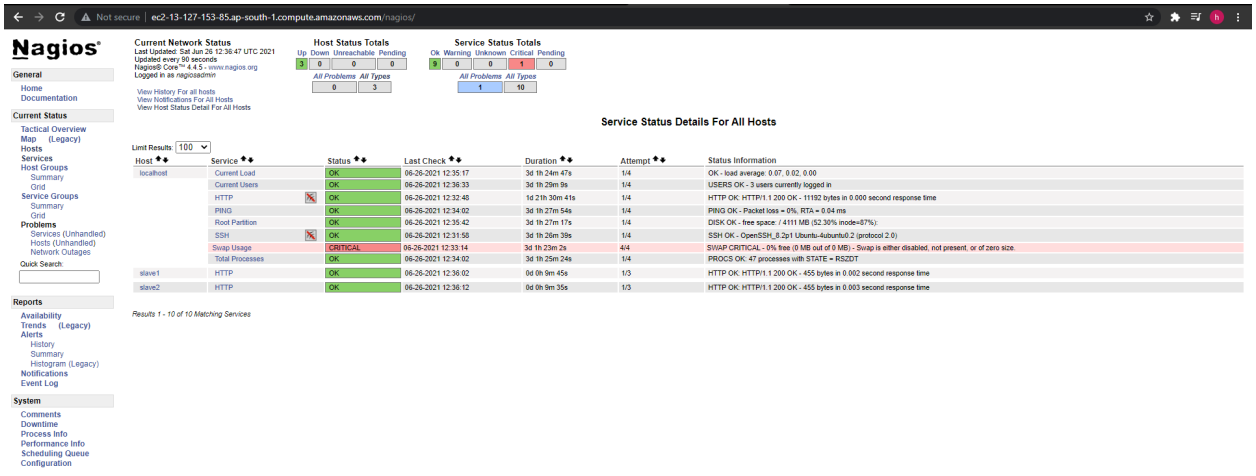
```
 GNU nano 4.8                                  nagios.cfg

# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg

# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
cfg_file=/usr/local/nagios/etc/objects/slave1.cfg
cfg_file=/usr/local/nagios/etc/objects/slave2.cfg
```

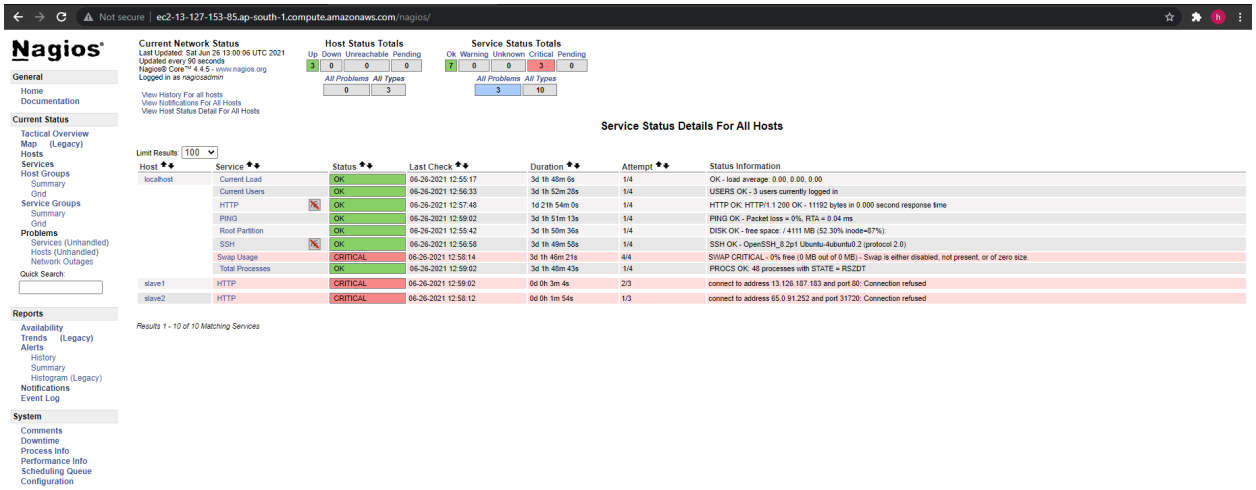
6) Accessing on Nagios Dashboard

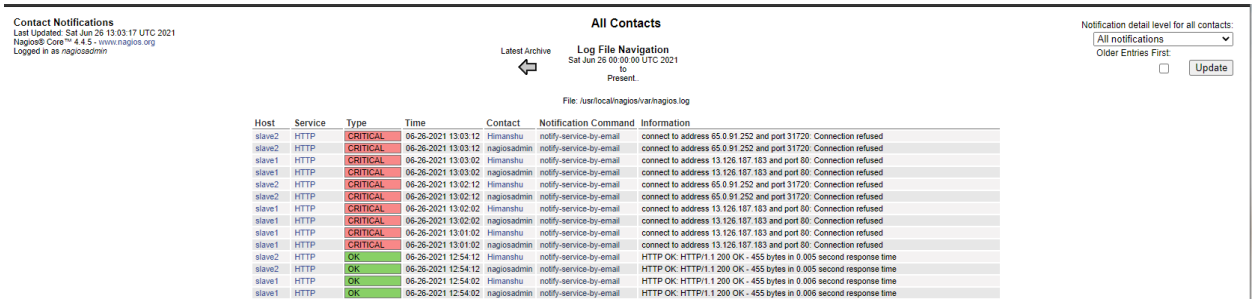




DevOps Project

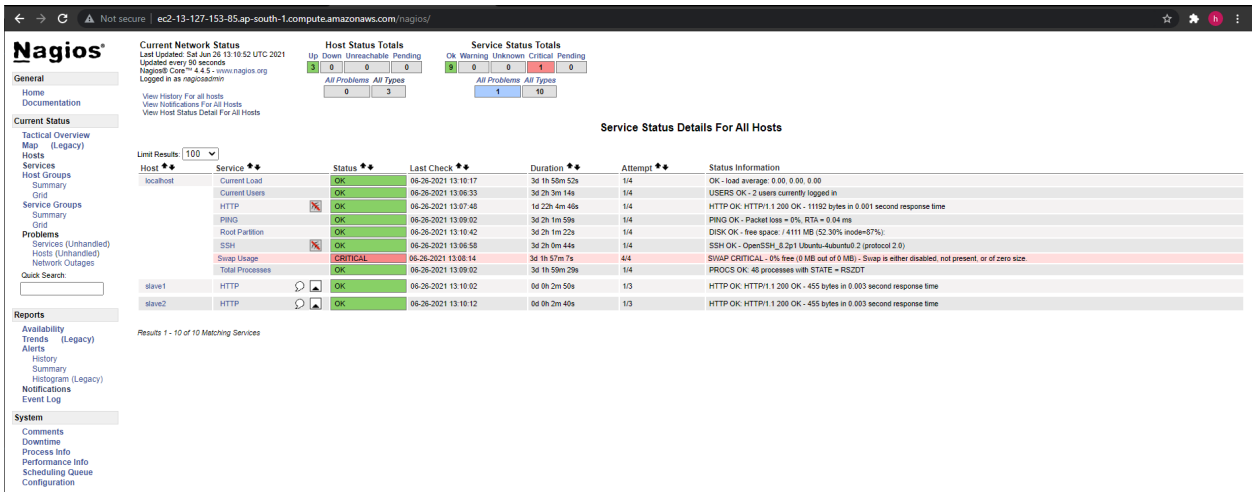
7) After disabling the http service on Testing and Production server and Contact notifications of Nagios.





DevOps Project

8) After enabling the http service on Testing and Production server.



DevOps Project

6th Commands of all the above tasks on :

1) Server 1 =

```
217  ansible-galaxy init configfile --offline
218  ls
219  cd configfile/
220  ls
221  cd files/
222  ls
223  nano config.txt
224  ls
225  cd ..
226  cd tasks/
227  ls
228  nano configure.yml
229  nano main.yml
230  cd ..
231  cd ../..
232  ls
233  nano playbook.yaml
234  ansible-playbook playbook.yaml --syntax-check
235  ansible-playbook playbook.yaml
236  cd /usr/local/nagios/etc/objects
237  ls
238  nano slave1.cfg
239  cd ..
240  ls
241  nano nagios.cfg
242  systemctl restart nagios
243  cd objects/
244  ls
245  nano slave1.cfg
246  cd /usr/local/nagios/etc/objects/
247  ls
248  nano contacts.cfg
249  systemctl restart nagios
250  nano contacts.cfg
251  nano slave1.cfg
252  nano contacts.cfg
253  nano slave1.cfg
254  nano contacts.cfg
255  nano slave1.cfg
256  systemctl restart nagios
257  cd ..
258  nano nagios.cfg
259  cd objects/
260  ls
261  nano co
262  nano contacts.cfg
263  systemctl restart nagios
264  nano contacts.cfg
265  nano slave1.cfg
```

DevOps Project

```
  266  systemctl restart nagios
  267  nano slave1.cfg
  268  systemctl restart nagios
  269  nano contacts.cfg
  270  cd /usr/local/nagios/etc/objects/
  271  ls
  272  nano contacts.cfg
  273  systemctl restart nagios
  274  sudo apt-get remove nagios-nrpe-server
  275  sudo apt-get remove nagios*
  276  which nagios
  277  cd /usr/local/nagios/etc/objects/
  278  ls
  279  nano slave1.cfg
  280  rm slave1.cfg
  281  ls
  282  systmectl restart nagios
  283  systemctl restart nagios
  284  nano contacts.cfg
  285  sudo nano /usr/local/nagios/etc/nagios.cfg && sudo mkdir sudo nano /usr/local/nagios/etc/servers
 && sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg && sudo service nagios start
&& sudo systemctl enable nagios && sudo systemctl restart nagios
  286  sudo mkdir sudo nano /usr/local/nagios/etc/servers && sudo /usr/local/nagios/bin/nagios -v /usr/
local/nagios/etc/nagios.cfg && sudo service nagios start && sudo systemctl enable nagios && sudo system
ctl restart nagios
  287  cd /usr/local/nagios/etc/objects/
  288  ls
  289  nano contacts.cfg
  290  systemctl restart nagios
  291  la
  292  ls
  293  nano commands.cfg
  294  nano contacts.cfg
  295  systemctl restart  nagios
  296  ls
  297  nano contacts.cfg
  298  nano localhost.cfg
  299  systemctl restart nagios
  300  sudo apt-get remove nagios-agent
  301  sudo apt-get --purge autoremove nagios-agent
  302  systemctl restart nagios
  303  journalctl -xe"
  304  cd /usr/local/nagios/etc/objects/
  305  ls
  306  cd ..
  307  nano nagios.cfg
  308  cd objects/
  309  ls
  310  rm slave3.cfg
  311  ls
  312  systemctl restart nagios
```

DevOps Project

```
312  systemctl restart nagios
313  naon slave1.cfg
314  nano slave1.cfg
315  systemctl restart nagios
316  nano /etc/ansible/hosts
317  cd /usr/local/nagios/etc/objects/
318  nano slave2.cfg
319  systemctl restart nagios
320  nano slave2.cfg
321  systemctl restart nagios
322  nano slave1.cfg
323  systemctl restart nagios
324  nano slave1.cfg
325  systemctl restart nagios
326  nano commands.cfg
327  nano contacts.cfg
328  nano templates.cfg
329  ls
330  nano commands.cfg
331  nano slave2.cfg
332  systemctl restart nagios
333  nano commands.cfg
334  systemctl restart nagios
335  nano commands.cfg
336  systemctl restart nagios
337  nano commands.cfg
338  systemctl restart nagios
339  nano commands.cfg
340  nano slave2.cfg
341  systemctl restart nagios
342  nano slave2.cfg
343  nano localhost.cfg
344  nano slave2.cfg
345  nano commands.cfg
346  nano slave2.cfg
347  systemctl restart nagios
348  nano slave2.cfg
349  nano slave1.cfg
350  systemctl restart nagios
351  ls
352  cd ...
353  cd ..
354  ls
355  cd ..
356  ls
357  cd libexec/
358  ls
359  cd ..
360  cd etc/objects/
```

DevOps Project

```
  361  ls
  362  nano contacts.cfg
  363  nano slave1.cfg
  364  nano slave2.cfg
  365  systemctl restart nagios
  366  cd /usr/local/nagios/etc/objects/
  367  nano slave1.cfg
  368  nano slave2.cfg
  369  nano contacts.cfg
  370  nano slave2.cfg
  371  systemctl restart nagios
  372  ls
  373  cd
  374  history
root@ip-172-31-43-131:~#
```

2) Server 2 =

```
   33  apt install unzip
   34  wget https://chromedriver.storage.googleapis.com/index.html?path=91.0.4472.101/
   35  ls
   36  rm 'index.html?path=91.0.4472.101%2F'
   37  wget https://chromedriver.storage.googleapis.com/91.0.4472.101/chromedriver_linux64.zip
   38  ls
   39  unzip chromedriver_linux64.zip
   40  wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
   41  sudo apt install ./google-chrome-stable_current_amd64.deb
   42  google-chrome -v
   43  google-chrome --version
   44  docker login
   45  git clone https://github.com/Himanshu369/Selenium.git
   46  ls
   47  cd Selenium/
   48  ls
   49  java -jar project2.jar
   50  docker images
   51  ls
   52  cd Selenium/
   53  ls
   54  java -jar project2.jar
   55  docker images
   56  docker rm -f himanshu369/project2
   57  docker rmi -f himanshu369/project2
   58  docker images
   59  docker rmi -f himanshu369/project2
   60  ls
   61  docker rm -f $(docker ps -a -q)
   62  docker rmi -f $(docker images -q)
   63  ls
   64  cat config.txt
   65  sudo apt-get update && sudo apt-get install -y nagios-nrpe-server nagios-plugins && sudo nano /e
tc/nagios/nrpe.cfg
   66  sudo /etc/init.d/nagios-nrpe-server restart
   67  ls
```

DevOps Project

```
   68  docker ps
   69  cd project2testcases/
   70  git pull
   71  java -jar project2.jar
   72  exit
   73  la
   74  docker images
   75  docker run -itd -p 80:80 bb
   76  ls
   77  git clone https://github.com/Himanshu369/Selenium.git
   78  historu
   79  history
root@ip-172-31-12-142:/home/ubuntu#
```

3) Server 3 =

```
   22  kubeadm init --pod-network-cidr 192.168.0.0/16
   23  kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
   24  kubectl get nodes
   25  cd /root/.ssh
   26  ls
   27  nano authorized_keys
   28  apt install python -y && apt install default-jdk -y
   29  apt install unzip
   30  wget https://chromedriver.storage.googleapis.com/index.html?path=91.0.4472.101/
   31  ls
   32  rm 'index.html?path=91.0.4472.101%2F'
   33  unzip chromedriver_linux64.zip
   34  wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
   35  sudo apt install ./google-chrome-stable_current_amd64.deb
   36  google-chrome --version
   37  ls
   38  git clone https://github.com/Himanshu369/Selenium.git
   39  kubectl get pods
   40  kubectl get pods --all-namespaces
   41  kubectl get svc
   42  kubectl giet svc -all-namespaces
   43  kubectl get svc -all-namespaces
   44  kubectl get svc --all-namespaces
   45  ls
   46  nano deploy.yaml
   47  kubectl apply -f [https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.47.0/
deploy/static/provider/baremetal/deploy.yaml](https://raw.githubusercontent.com/kubernetes/ingress-ngin
x/controller-v0.47.0/deploy/static/provider/baremetal/deploy.yaml)
   48  kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.47.0/d
eploy/static/provider/baremetal/deploy.yaml
```

DevOps Project

```
49  nano deploy.yaml
50  nano ingress.yaml
51  kubectl delete -f /home/ubuntu/ingress.yaml
52  kubectl delete service clusterip project2 --tcp=80:80
53  kubectl delete service clusterip project2
54  kubectl get svc
55  kubectl get pod
56  kubectl delete -f /home/ubuntu/deploy.yaml
57  kubectl get svc --all-namespaces
58  ls
59  cd Selenium/
60  git pull
61  kubectl delete service clusterip project2
62  kubectl delete -f /home/ubuntu/deploy.yaml
63  kubectl delete -f /home/ubuntu/ingress.yaml
64  kubectl get svc --all-namespaces
65  kubectl delete -f /home/ubuntu/ingress.yaml
66  kubectl delete -f /home/ubuntu/deploy.yaml
67  kubectl get svc --all-namespaces
68  kubectl create -f /home/ubuntu/deploy.yaml
69  kubectl create -f /home/ubuntu/ingress.yaml
70  kubectl create service clusterip project2 --tcp=80:80
71  kubectl get svc --all-namespaces
72  kubectl get pods
73  kubectl delete -f /home/ubuntu/ingress.yaml
74  kubectl get svc --all-namespaces
75  kubectl create -f /home/ubuntu/ingress.yaml
76  ls
77  kubectl get svc
78  kubectl get svc --all-namespaces
79  ls
80  java -jar project2k8s.jar
81  cd ..
82  ls
83  c
84  cd Selenium/
85  ls
86  git pull
87  java -jar project2k8s.jar
88  cd ..
89  ls
90  java -jar project2k8s.jar
91  cd Selenium/
92  java -jar project2k8s.jar
93  java -jar project2.jar
94  git push
95  git pull
96  ls
97  java -jar project2k8s.jar
98  ls
```

DevOps Project

```
   99  cat config.txt
  100  sudo apt-get update && sudo apt-get install -y nagios-nrpe-server nagios-plugins && sudo nano /e
tc/nagios/nrpe.cfg
  101  sudo /etc/init.d/nagios-nrpe-server restart
  102  nano /etc/nagios/nrpe.cfg
  103  sudo /etc/init.d/nagios-nrpe-server restart
  104  ls
  105  kubectl create -f deploy.yaml
  106  kubectl get svc
  107  kubectl get pods --all-namespaces
  108  kubectl get svc --all-namespaces
  109  ls
  110  rm -rf Selenium/
  111  git clone https://github.com/Himanshu369/project2testcases.git
  112  cd project2testcases/
  113  ls
  114  git pull
  115  ls
  116  git pull
  117  java -jar project2k8s.jar
  118  git pull
  119  java -jar project2k8s.jar
  120  ls
  121  kubectl get po
  122  kubectl get vcs
  123  kubectl get svc
  124  kubectl delete service project2
  125  kubectl create service nodeport project2
  126  kubectl create service nodeport project2 --tcp=80:80
  127  kubectl delete -f ingress.yaml
  128  kubectl get svc --all-namespaces
  129  exit
  130  kubectl get svc
  131  kubectl get po
  132  ls
  133  git clone https://github.com/Himanshu369/Selenium.git
  134  ls
  135  historu
  136  history
root@ip-172-31-39-210:/home/ubuntu#
```

DevOps Project

4) Server 4 =

```
   34  kubeadm join 172.31.39.210:6443 --token 24popk.gcy17msgcnoa8k2r         --discovery-token-ca-cer
t-hash sha256:4eea26c9478f2c59adb70f6c3a23511e9b4d0bca291d4f4479d18003335a878e
   35  cat /root/.ssh/authorized_keys
   36  which python
   37  which java
   38  apt install unzip
   39  wget https://chromedriver.storage.googleapis.com/index.html?path=91.0.4472.101/
   40  rm 'index.html?path=91.0.4472.101%2F'
   41  unzip chromedriver_linux64.zip
   42  wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
   43  sudo apt install ./google-chrome-stable_current_amd64.deb
   44  google-chrome --version
   45  git clone https://github.com/Himanshu369/Selenium.git
   46  ls
   47  ansible -m ping all
   48  which ansible
   49  cat config.txt
   50  sudo apt-get update && sudo apt-get install -y nagios-nrpe-server nagios-plugins && sudo nano /e
tc/nagios/nrpe.cfg
   51  sudo /etc/init.d/nagios-nrpe-server restart
   52  nano /etc/nagios/nrpe.cfg
   53  sudo /etc/init.d/nagios-nrpe-server restart
   54  history
root@ip-172-31-8-231:/home/ubuntu#
```

End of Project