Project 3

You are hired as a DevOps Engineer for Analytics Pvt Ltd. This company is a product-based organization which uses Docker for their containerization needs within the company. The final product received a lot of traction in the first few weeks of launch. Now with the increasing demand, the organization needs to have a platform for automating deployment, scaling and operations of application containers across clusters of hosts. As a DevOps Engineer, you need to implement a DevOps lifecycle such that all the requirements are implemented without any change in the Docker containers in the testing environment.

Up until now, this organization used to follow a monolithic architecture with just 2 developers. The product is present on: https://github.com/hshar/website.git Following are the specifications of the lifecycle:

1. Git workflow should be implemented. Since the company follows a monolithic architecture of development, you need to take care of version control. The release should happen only on the 25th of every month.
2. CodeBuild should be triggered once the commits are made in the master branch.
3. The code should be containerized with the help of the Dockerfile. The Dockerfile should be built every time if there is a push to GitHub. Create a custom Docker image using a Dockerfile.
4. As per the requirement in the production server, you need to use the Kubernetes cluster and the containerized code from Docker Hub should be deployed with 2 replicas. Create a NodePort service and configure the same for port 30008.
5. Create a Jenkins Pipeline script to accomplish the above task.
6. For configuration management of the infrastructure, you need to deploy the configuration on the servers to install necessary software and configurations.
7. Using Terraform, accomplish the task of infrastructure creation in the AWS cloud provider.

Architectural Advice:

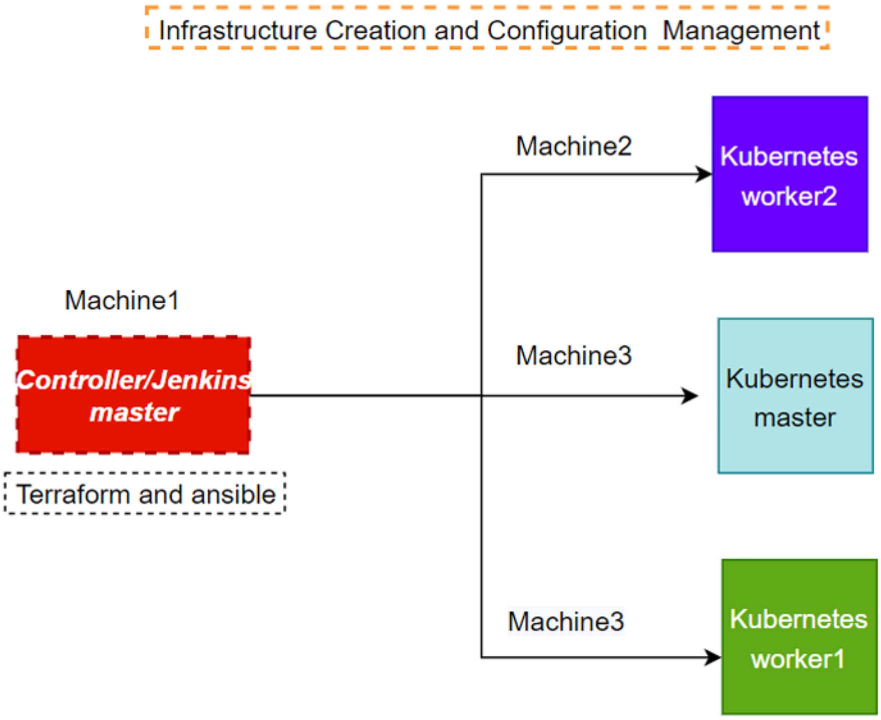
Software's to be installed on the respective machines using configuration management.

Worker1: Jenkins, Java

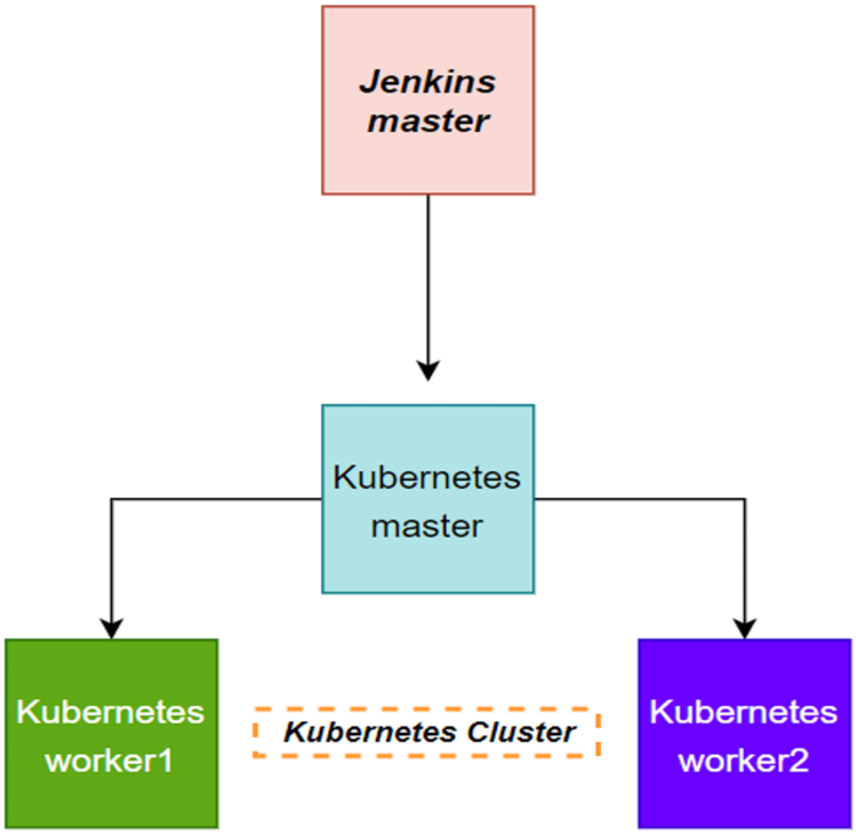
Worker2: Docker, Kubernetes

Worker3: Java, Docker, Kubernetes

Worker4: Docker, Kubernetes



Servers for jenkins and kubernetes configuration



DevOps Project

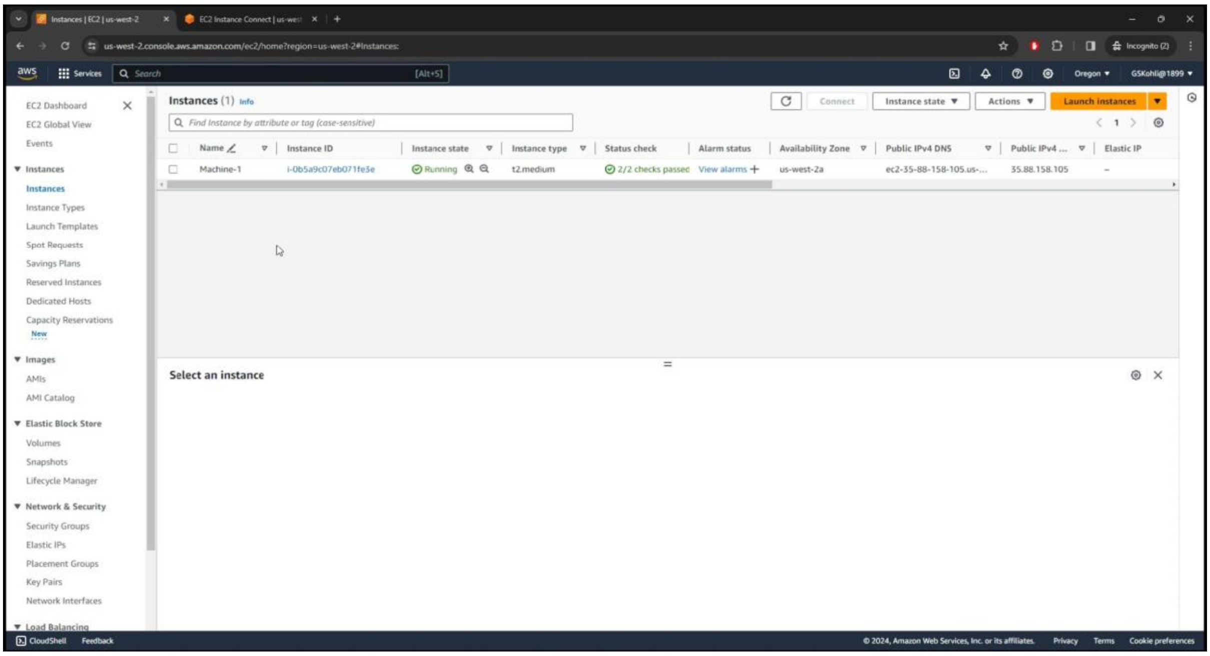
Architecture

We have total 4 Machine. Machine 1, 2, 3 and 4

- Machine-1 - Terraform, Ansible, Java and Jenkins Installed
- Machine-2 – Docker and Kubernetes
- Machine-3 – Java, Docker and Kubernetes
- Machine-4 - Docker and Kubernetes

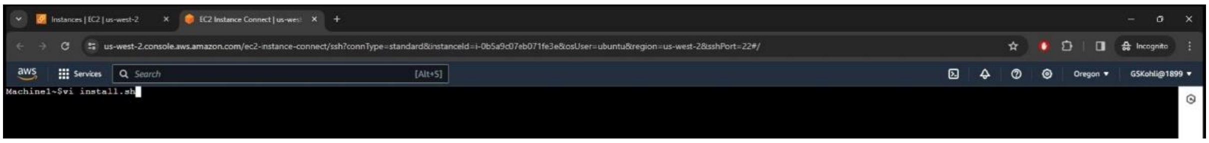
Machine 1 (Jenkins Master)→Machine 2(Kubernetes Master)→ Machine 3 and Machine 4

1. Creating Machine-1



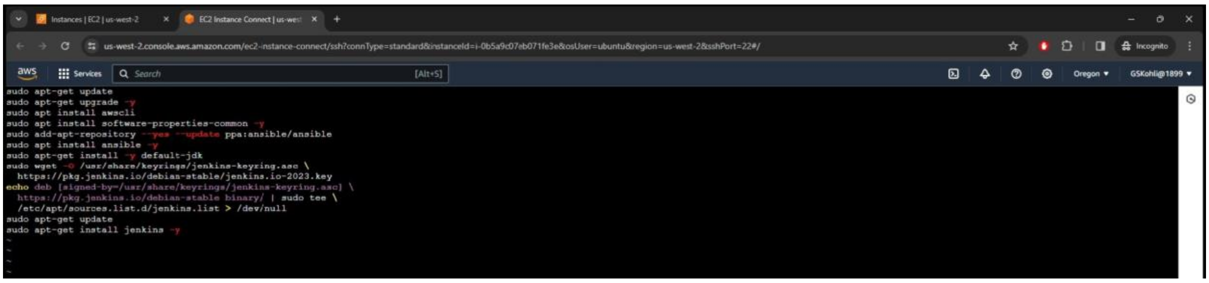
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Machine-1	i-0b5a9c07eb071fe3e	Running	t2.medium	2/2 checks passed	View alarms +	us-west-2a	ec2-35-88-158-105.us-...	35.88.158.105	–

2. Making script to install required packages for Machine 1



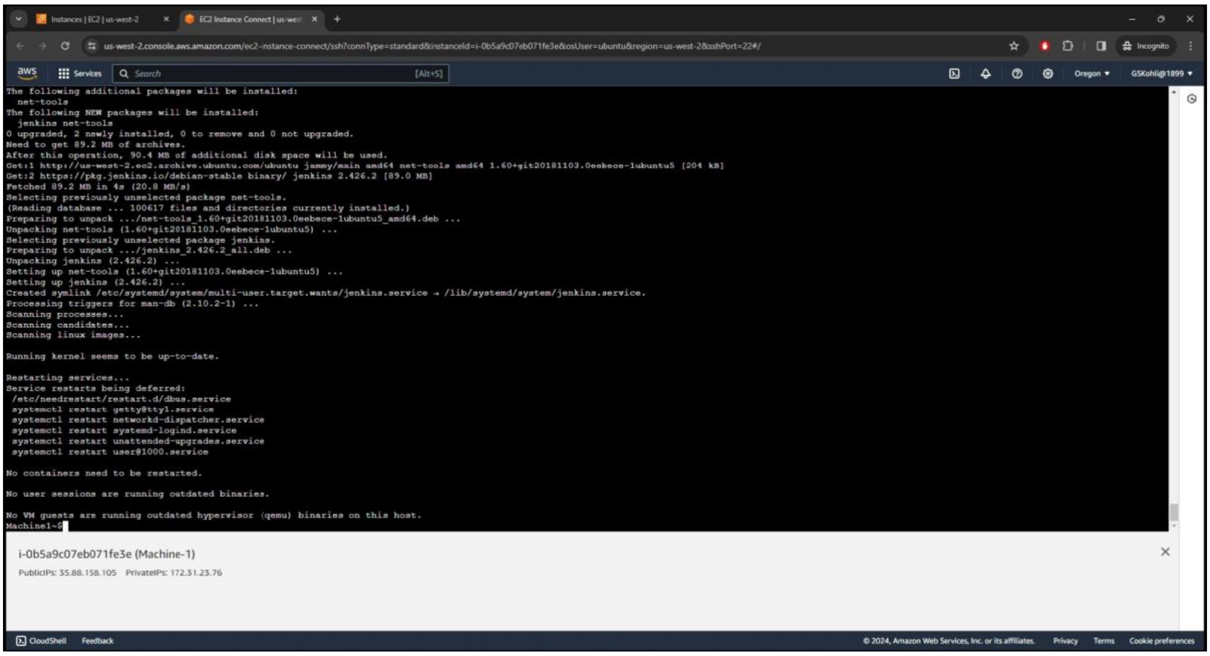
```
Machine1~$vi install.sh
```

Instances | EC2 | us-west-2
EC2 Instance Connect | us-west
us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0b5a9c07eb071fe3e&osUser=ubuntu&region=us-west-2&sshPort=22#/
Services
Search
[Alt+S]
Oregon
GSKohli@1899



```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt install awscli
sudo apt install software-properties-common -y
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y
sudo apt-get install -y default-jdk
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
~
~
~
~
```

Instances | EC2 | us-west-2
EC2 Instance Connect | us-west
us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0b5a9c07eb071fe3e&osUser=ubuntu&region=us-west-2&sshPort=22#/
Services
Search
[Alt+S]
Oregon
GSKohli@1899



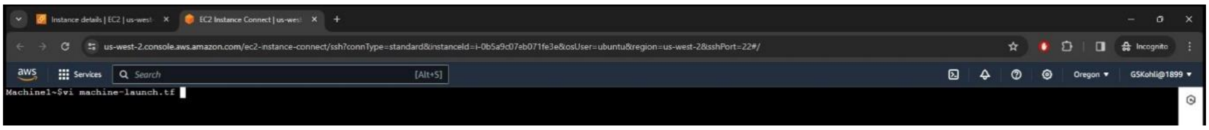
```
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 89.2 MB of archives.
After this operation, 90.4 MB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.426.2 [89.0 MB]
Fetched 89.2 MB in 4s (20.8 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 100617 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.426.2_all.deb ...
Unpacking jenkins (2.426.2) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.426.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
Service restarts being deferred:
 /etc/needrestart/restart.d/dbus.service
 systemctl restart getty@tty1.service
 systemctl restart networkd-dispatcher.service
 systemctl restart systemd-logind.service
 systemctl restart unattended-upgrades.service
 systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
Machine1~$
```

i-0b5a9c07eb071fe3e (Machine-1)

PublicIPs: 35.88.158.105 PrivateIPs: 172.31.23.76

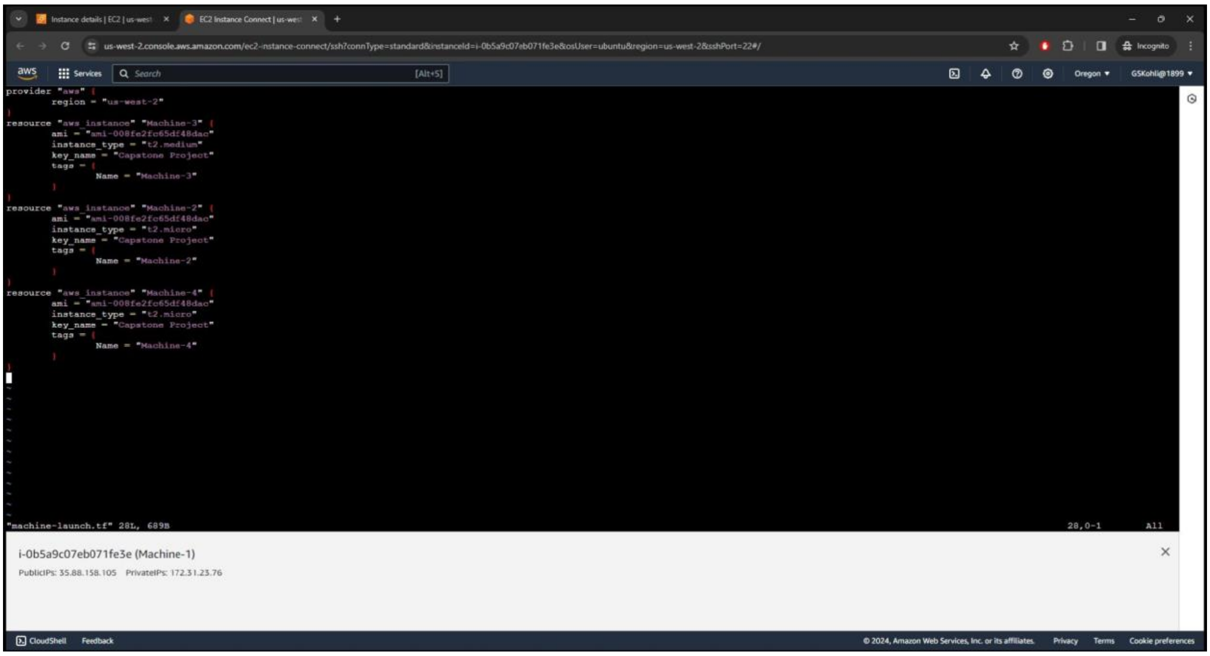
Instances | EC2 | us-west-2
EC2 Instance Connect | us-west
us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0b5a9c07eb071fe3e&osUser=ubuntu&region=us-west-2&sshPort=22#/
Incognito
Services
Search
[Alt+S]
Oregon
GSKohli@1899
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

3. Making a terraform script to create all other Machines.



```
Machine1~$vi machine-launch.tf
```

Instance details | EC2 | us-west
EC2 Instance Connect | us-west
us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0b5a9c07eb071fe3e&osUser=ubuntu&region=us-west-2&sshPort=22#/
Incognito
Services
Search
[Alt+S]
Oregon
GSKohli@1899

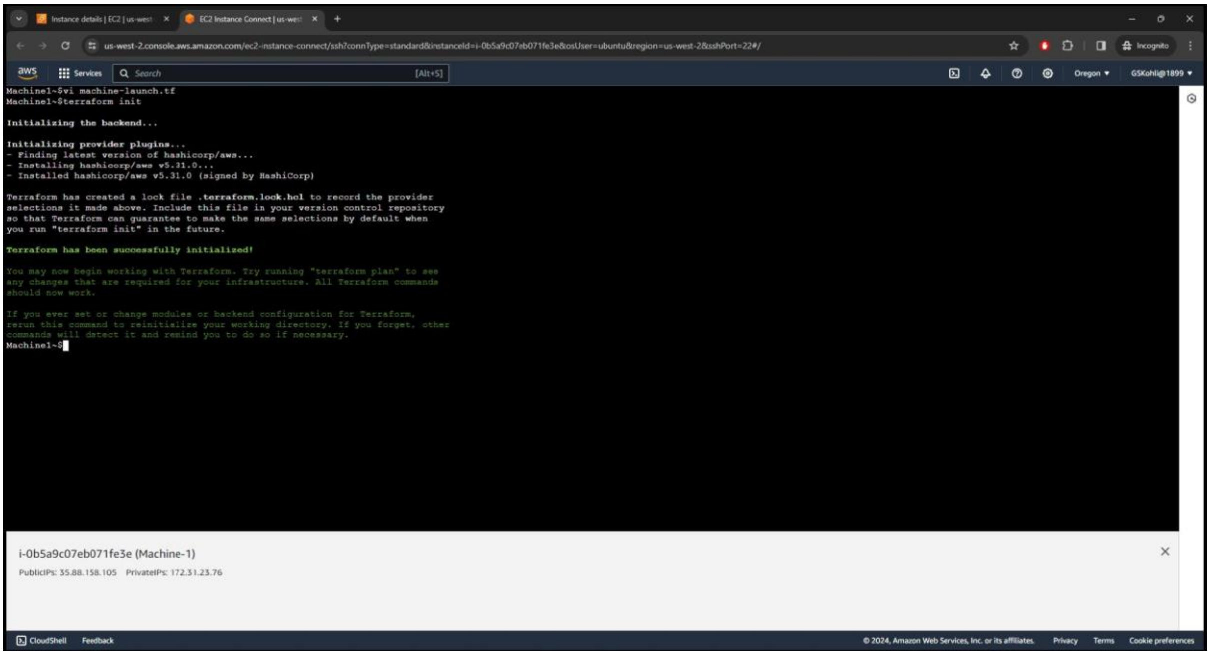


```
provider "aws" {
        region = "us-west-2"
}
resource "aws_instance" "Machine-3" {
        ami = "ami-008fe2fc65df48dac"
        instance_type = "t2.medium"
        key_name = "Capstone Project"
        tags = {
                Name = "Machine-3"
        }
}
resource "aws_instance" "Machine-2" {
        ami = "ami-008fe2fc65df48dac"
        instance_type = "t2.micro"
        key_name = "Capstone Project"
        tags = {
                Name = "Machine-2"
        }
}
resource "aws_instance" "Machine-4" {
        ami = "ami-008fe2fc65df48dac"
        instance_type = "t2.micro"
        key_name = "Capstone Project"
        tags = {
                Name = "Machine-4"
        }
}

~
~
~
~
~
~
~
~
~
~
~
~
~
"machine-launch.tf" 28L, 689B                                                    28,0-1        All
```

i-0b5a9c07eb071fe3e (Machine-1)

PublicIPs: 35.88.158.105 PrivateIPs: 172.31.23.76

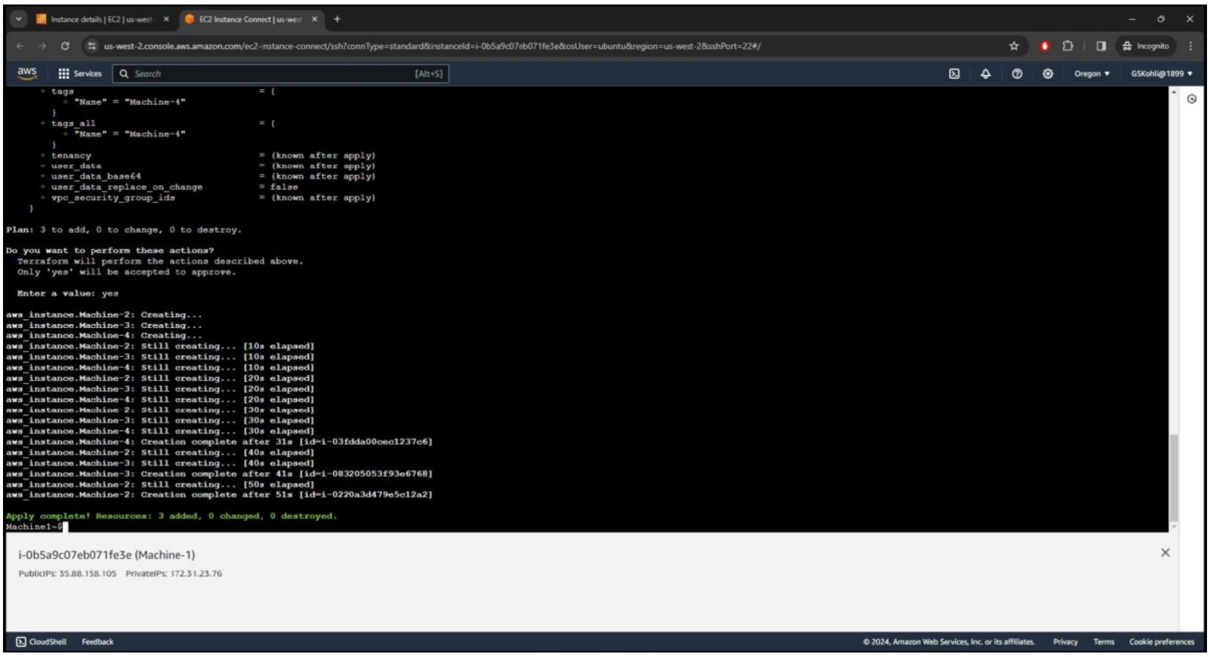
Instance details | EC2 | us-west
EC2 Instance Connect | us-west
us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0b5a9c07eb071fe3e&osUser=ubuntu&region=us-west-2&sshPort=22#/
Incognito
Services
Search
[Alt+S]
Oregon
GSKohli@1899
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



```
Machine1~$vi machine-launch.tf
Machine1~$terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
Machine1~$
```

i-0b5a9c07eb071fe3e (Machine-1)

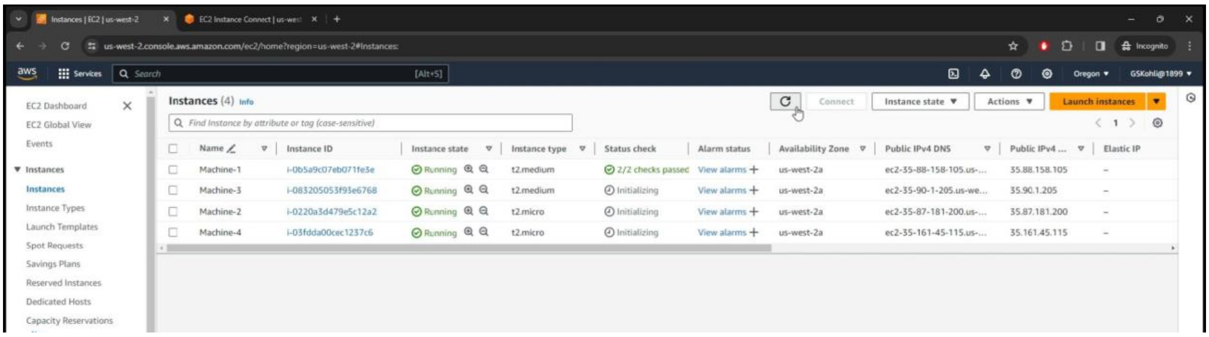
PublicIPs: 35.88.158.105 PrivateIPs: 172.31.23.76



```
      + tags                                 = {
          + "Name" = "Machine-4"
        }
      + tags_all                             = {
          + "Name" = "Machine-4"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.Machine-2: Creating...
aws_instance.Machine-3: Creating...
aws_instance.Machine-4: Creating...
aws_instance.Machine-2: Still creating... [10s elapsed]
aws_instance.Machine-3: Still creating... [10s elapsed]
aws_instance.Machine-4: Still creating... [10s elapsed]
aws_instance.Machine-2: Still creating... [20s elapsed]
aws_instance.Machine-3: Still creating... [20s elapsed]
aws_instance.Machine-4: Still creating... [20s elapsed]
aws_instance.Machine-2: Still creating... [30s elapsed]
aws_instance.Machine-3: Still creating... [30s elapsed]
aws_instance.Machine-4: Still creating... [30s elapsed]
aws_instance.Machine-4: Creation complete after 31s [id=i-03fdda00cec1237c6]
aws_instance.Machine-2: Still creating... [40s elapsed]
aws_instance.Machine-3: Still creating... [40s elapsed]
aws_instance.Machine-3: Creation complete after 41s [id=i-083205053f93e6768]
aws_instance.Machine-2: Still creating... [50s elapsed]
aws_instance.Machine-2: Creation complete after 51s [id=i-0220a3d479e5c12a2]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
Machine1~$
```

i-0b5a9c07eb071fe3e (Machine-1)

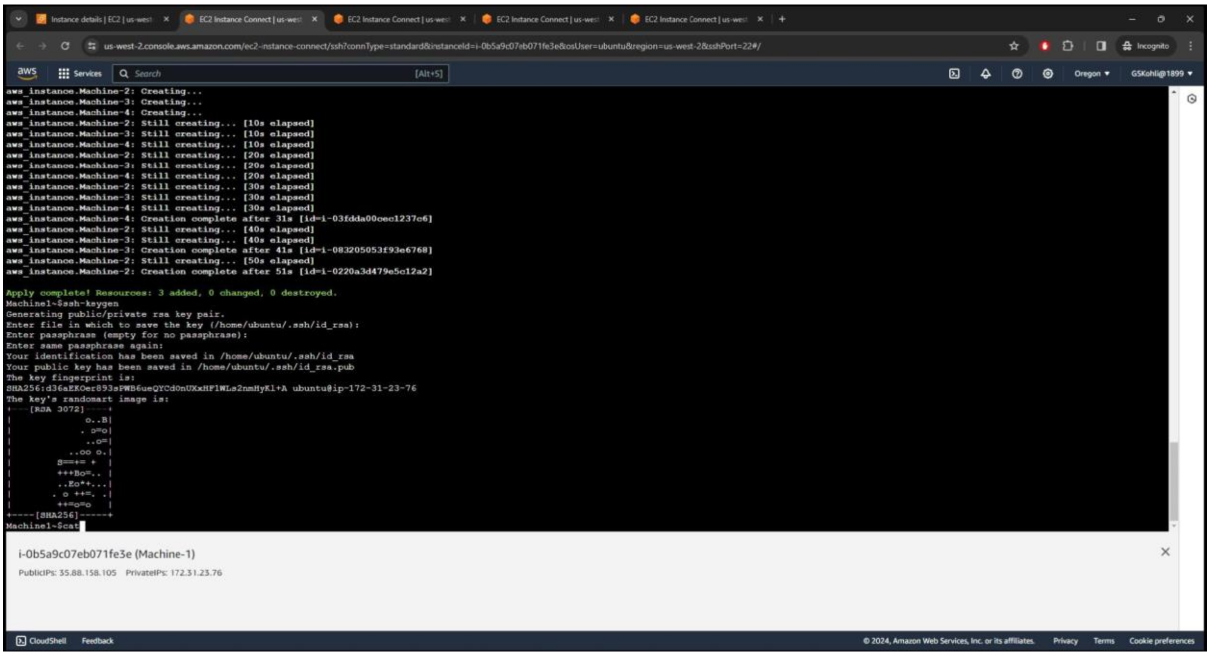
PublicIPs: 35.88.158.105 PrivateIPs: 172.31.23.76



Instances (4)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Machine-1	i-0b5a9c07eb071fe3e	Running	t2.medium	2/2 checks passed	View alarms +	us-west-2a	ec2-35-88-158-105.us-...	35.88.158.105	–
Machine-3	i-083205053f93e6768	Running	t2.medium	Initializing	View alarms +	us-west-2a	ec2-35-90-1-205.us-we...	35.90.1.205	–
Machine-2	i-0220a3d479e5c12a2	Running	t2.micro	Initializing	View alarms +	us-west-2a	ec2-35-87-181-200.us-...	35.87.181.200	–
Machine-4	i-03fdda00cec1237c6	Running	t2.micro	Initializing	View alarms +	us-west-2a	ec2-35-161-45-115.us-...	35.161.45.115	–

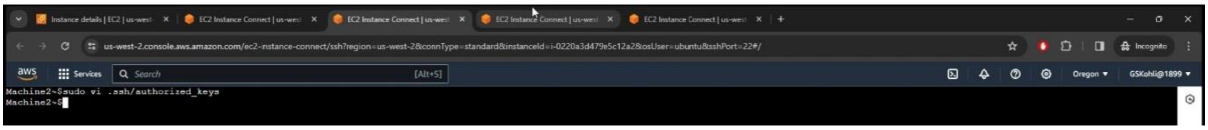
4. Setting up Ansible



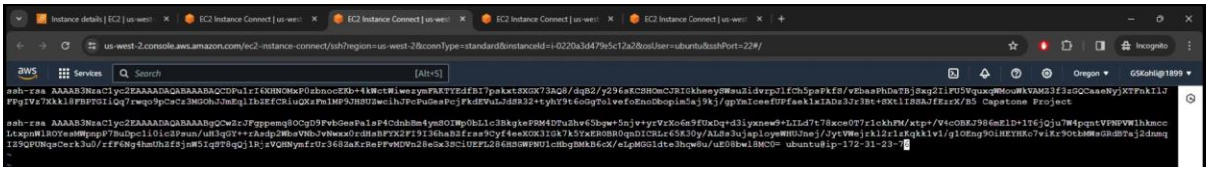
```
aws_instance.Machine-2: Creating...
aws_instance.Machine-3: Creating...
aws_instance.Machine-4: Creating...
aws_instance.Machine-2: Still creating... [10s elapsed]
aws_instance.Machine-3: Still creating... [10s elapsed]
aws_instance.Machine-4: Still creating... [10s elapsed]
aws_instance.Machine-2: Still creating... [20s elapsed]
aws_instance.Machine-3: Still creating... [20s elapsed]
aws_instance.Machine-4: Still creating... [20s elapsed]
aws_instance.Machine-2: Still creating... [30s elapsed]
aws_instance.Machine-3: Still creating... [30s elapsed]
aws_instance.Machine-4: Still creating... [30s elapsed]
aws_instance.Machine-4: Creation complete after 31s [id=i-03fdda00cec1237c6]
aws_instance.Machine-2: Still creating... [40s elapsed]
aws_instance.Machine-3: Still creating... [40s elapsed]
aws_instance.Machine-3: Creation complete after 41s [id=i-083205053f93e6768]
aws_instance.Machine-2: Still creating... [50s elapsed]
aws_instance.Machine-2: Creation complete after 51s [id=i-0220a3d479e5c12a2]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
Machine1~$ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:d36aEKOer893sPWB6ueQYCd0nUXxHFlWLs2nmHyKl+A ubuntu@ip-172-31-23-76
The key's randomart image is:
+---[RSA 3072]----+
|           o..B|
|          . o=o|
|           ..o=|
|        ..oo o.|
|      S==+= +  |
|      +++Bo=.. |
|      ..Eo*+...|
|     . o ++=. .|
|      ++=o=o   |
+----[SHA256]-----+
Machine1~$cat
```

i-0b5a9c07eb071fe3e (Machine-1)

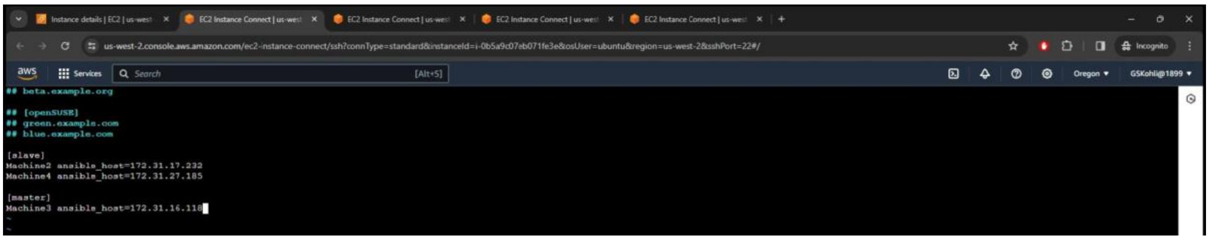
PublicIPs: 35.88.158.105 PrivateIPs: 172.31.23.76



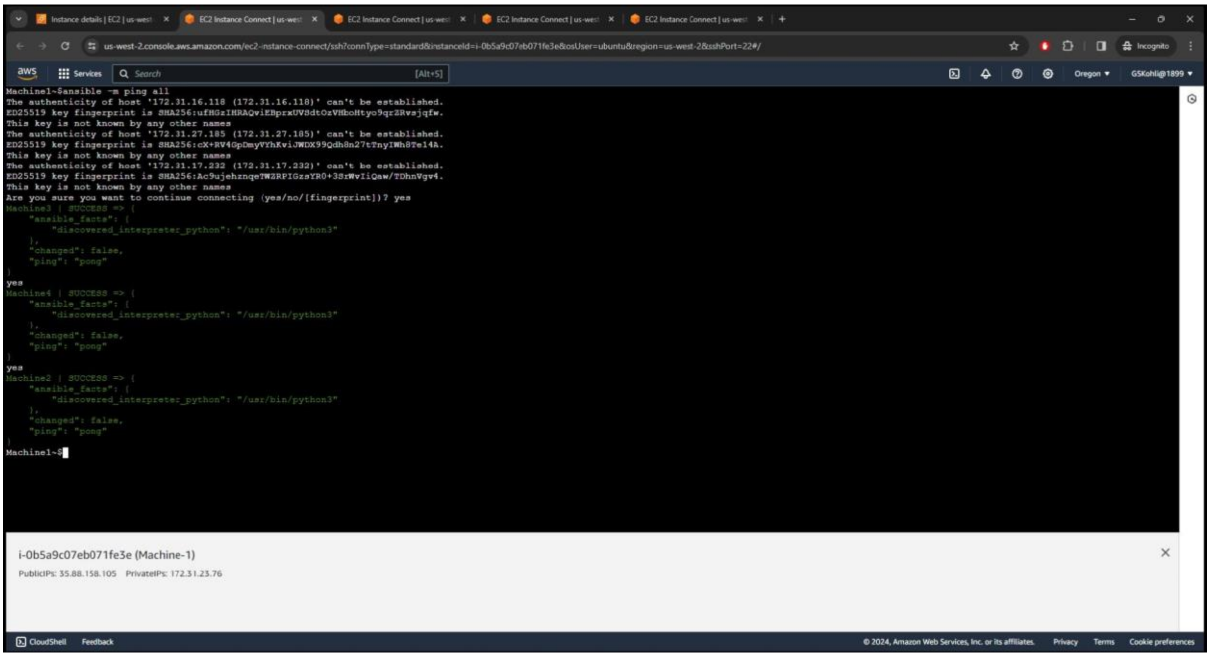
```
Machine2~$sudo vi .ssh/authorized_keys
Machine2~$
```



```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCDPu1rI6XHNOMxP0zbnocEKb+4kWctWiwezymFAKTYEdfBI7pskxtSXGX73AQ8/dqB2/y296sKCSHOmCJRIGkheeySWsuZidvrpJlfCh5psPkfS/vEbasPhDaTBjSxg2IiFU5VquxqWMouWkVAMZ3f3zGQCaaeNyjXTFnkIlJ
FPgIVz7Xkkl8FBPTGIiQq7rwqo9pCsCz3MGOhJJmEqlIbZEfCRiuQXzFm1MP9JHSUZwcihJPcPuGesPcjFkdEVuLJdSR32+tyhY9t6oGgTolvefoEnoDbopim5aj9kj/gpYmIceefUPfaek1xIADz3Jr3Bt+SXtlISSAJfEzrX/B5 Capstone Project

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCwZrJFgppemq8OCgD9FvbGesPalsP4CdnbBm4ymSOIWp0bLlc3BkgkePRM4DTuZhv65bgw+5njv+yrVrXo6m9fUxDq+d3iyxnew9+LILd7t78xce0T7rlckhFM/xtp+/V4cOBKJ986mElD+1T6jQju7W4pqntVPNPVW1hkmcc
LtxpnWlROYesMWpnpP7BuDpcli0icZPsun/uH3qGY++rAsdp2WbsVNbJvNwxx0rdHsBFYX2FI9I36haBZfrss9Cyf4eeXOX3IGk7k5YxEROBR0qnDICRLr65K30y/ALSs3ujaployeWHUJnej/JytVWejrkl2rlzKqkklv1/g1OEng9OiHEYHKc7viKr9OtbMWsGRdBTaj2dnmq
IZ9QPUNqsCerk3u0/rfF6Ng4hmUhZfSjnW5IqST8qQj1RjzVQHNymfrUr3682aKrRePFvMDVn28eGx3SCiUEFL286HSGWPNU1cHbgBMkB6cX/eLpMGG1dte3hqw8u/uE08bwl8MC0= ubuntu@ip-172-31-23-76
~
~
```



```
## beta.example.org

## [openSUSE]
## green.example.com
## blue.example.com

[slave]
Machine2 ansible_host=172.31.17.232
Machine4 ansible_host=172.31.27.185

[master]
Machine3 ansible_host=172.31.16.118
~
~
```

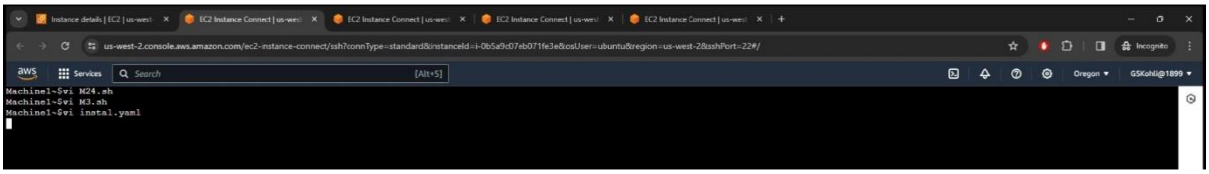


```
Machine1~$ansible -m ping all
The authenticity of host '172.31.16.118 (172.31.16.118)' can't be established.
ED25519 key fingerprint is SHA256:ufHGzIMRAQviEBprxUVSdtOzVHboHtyo9qrZRvsjqfw.
This key is not known by any other names
The authenticity of host '172.31.27.185 (172.31.27.185)' can't be established.
ED25519 key fingerprint is SHA256:cX+RV4GpDmyVYhKviJWDX99Qdh8n27tTnyIWh8Te14A.
This key is not known by any other names
The authenticity of host '172.31.17.232 (172.31.17.232)' can't be established.
ED25519 key fingerprint is SHA256:Ac9ujehznqeTWZRPIGzsYR0+3SrWvIiQaw/TDhnVgv4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Machine3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
yes
Machine4 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
yes
Machine2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
Machine1~$
```

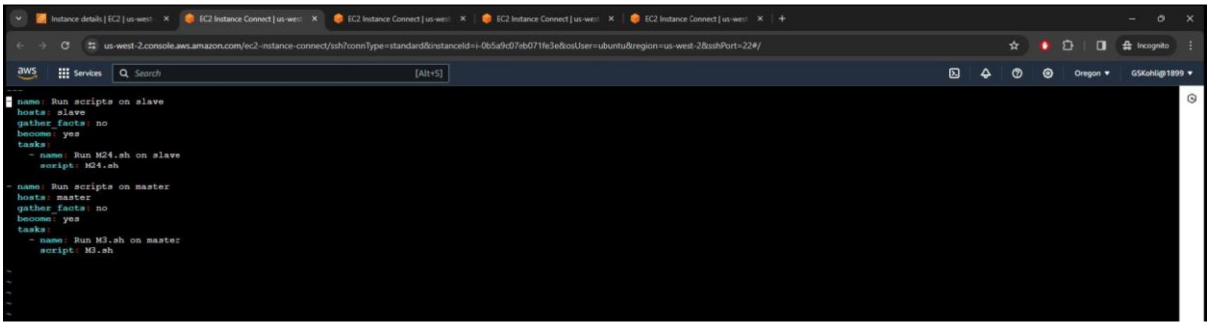
i-0b5a9c07eb071fe3e (Machine-1)

PublicIPs: 35.88.158.105 PrivateIPs: 172.31.23.76

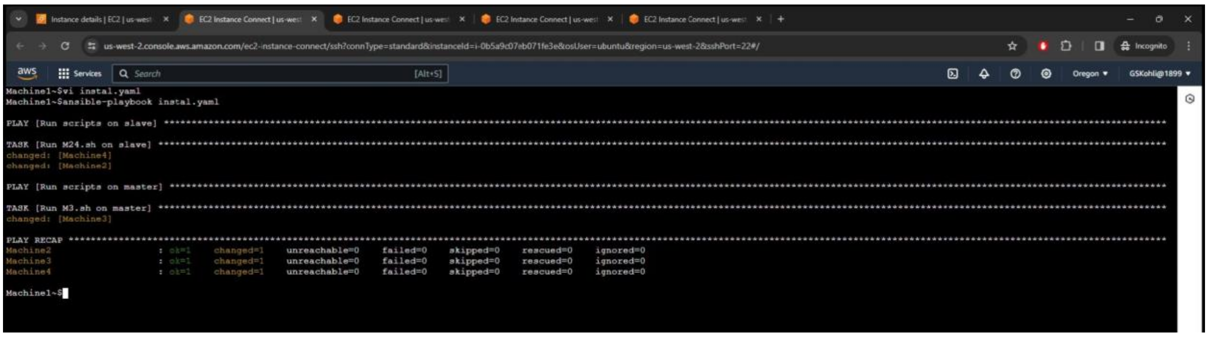
5. Making 2 shell script and 1 yaml script to install required packages in machine 2, 3 and 4



```
Machine1~$vi M24.sh
Machine1~$vi M3.sh
Machine1~$vi instal.yaml
```

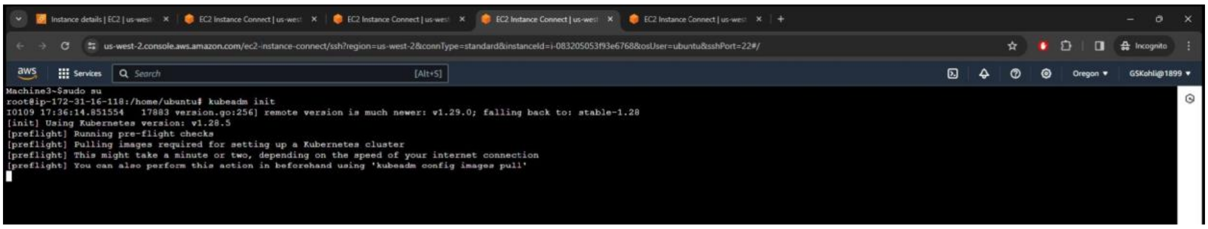


```
---
- name: Run scripts on slave
  hosts: slave
  gather_facts: no
  become: yes
  tasks:
    - name: Run M24.sh on slave
      script: M24.sh

- name: Run scripts on master
  hosts: master
  gather_facts: no
  become: yes
  tasks:
    - name: Run M3.sh on master
      script: M3.sh

~
~
~
~
~
```

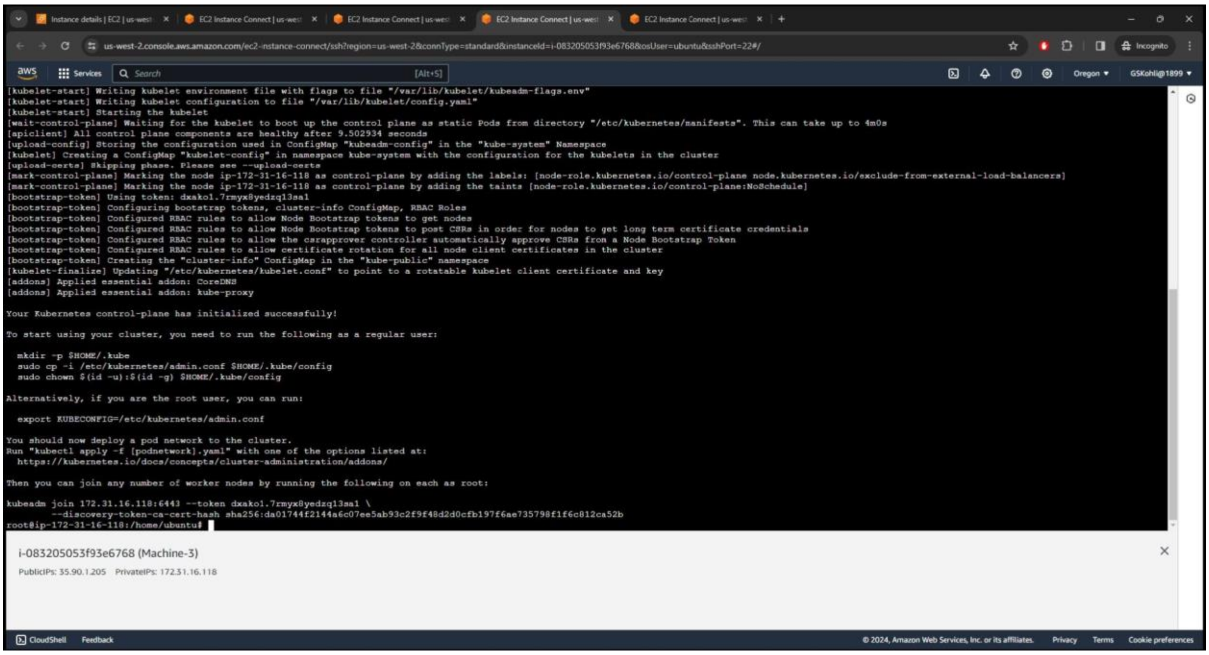


```
Machine1~$vi instal.yaml
Machine1~$ansible-playbook instal.yaml

PLAY [Run scripts on slave] *****************************************************************************

TASK [Run M24.sh on slave] ******************************************************************************
changed: [Machine4]
changed: [Machine2]

PLAY [Run scripts on master] ****************************************************************************

TASK [Run M3.sh on master] ******************************************************************************
changed: [Machine3]

PLAY RECAP **********************************************************************************************
Machine2                   : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Machine3                   : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Machine4                   : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Machine1~$
```

6. Setting Kubernetes Cluster



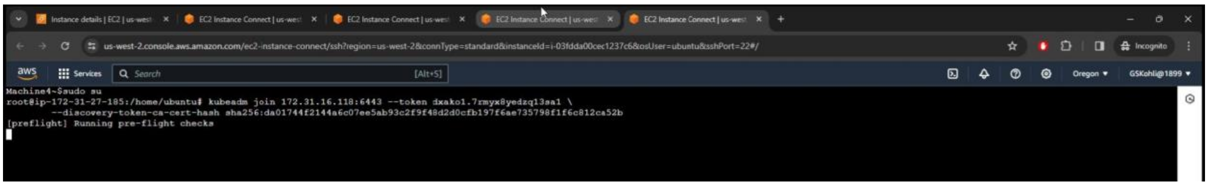
```
Machine3~$sudo su
root@ip-172-31-16-118:/home/ubuntu# kubeadm init
I0109 17:36:14.851554   17883 version.go:256] remote version is much newer: v1.29.0; falling back to: stable-1.28
[init] Using Kubernetes version: v1.28.5
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
```



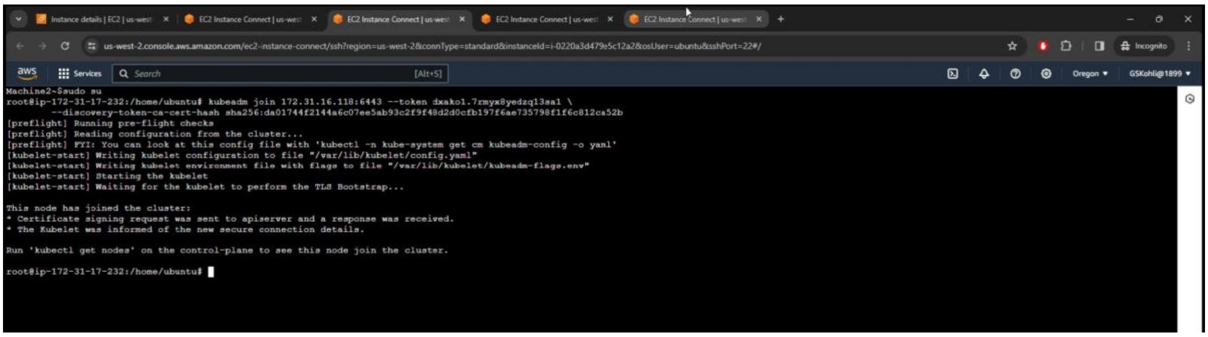
```
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
[apiclient] All control plane components are healthy after 9.502934 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node ip-172-31-16-118 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernetes.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node ip-172-31-16-118 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: dxakol.7rmyx8yedzq13sal
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.16.118:6443 --token dxakol.7rmyx8yedzq13sal \
        --discovery-token-ca-cert-hash sha256:da01744f2144a6c07ee5ab93c2f9f48d2d0cfb197f6ae735798f1f6c812ca52b
root@ip-172-31-16-118:/home/ubuntu#
```

i-083205053f93e6768 (Machine-3)

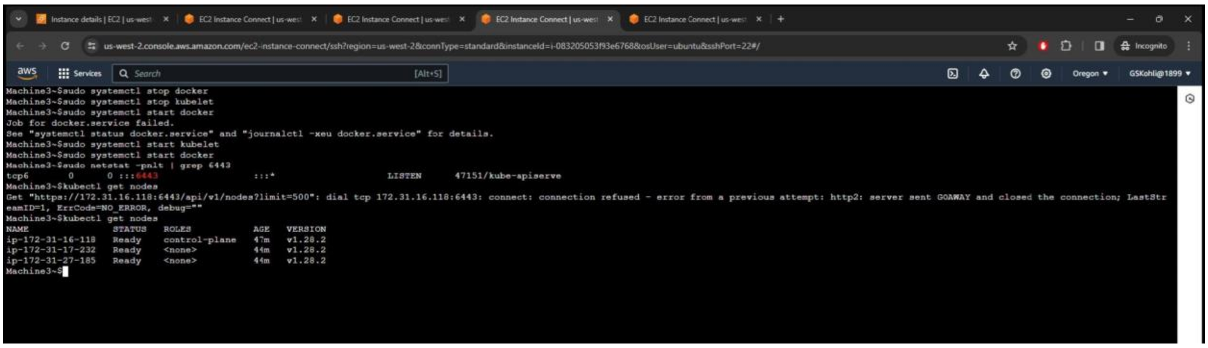
PublicIPs: 35.90.1.205   PrivateIPs: 172.31.16.118



```
Machine4~$sudo su
root@ip-172-31-27-185:/home/ubuntu# kubeadm join 172.31.16.118:6443 --token dxako1.7rmyx8yedzq13sa1 \
        --discovery-token-ca-cert-hash sha256:da01744f2144a6c07ee5ab93c2f9f48d2d0cfb197f6ae735798f1f6c812ca52b
[preflight] Running pre-flight checks
```

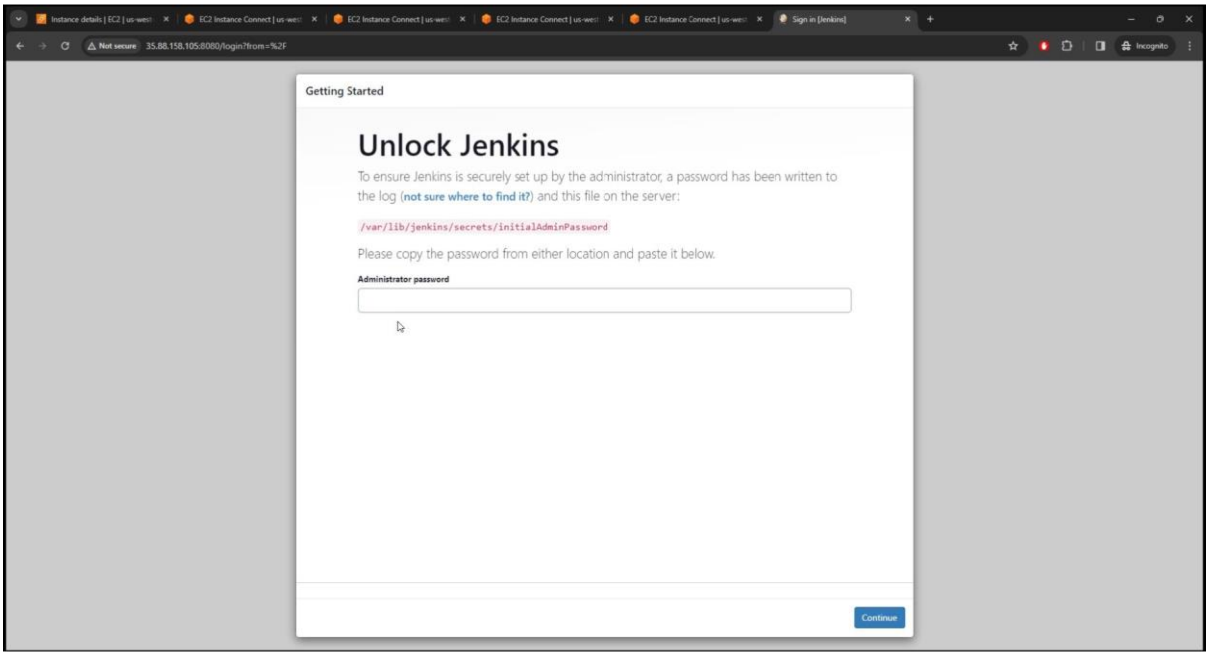


```
Machine2~$sudo su
root@ip-172-31-17-232:/home/ubuntu# kubeadm join 172.31.16.118:6443 --token dxako1.7rmyx8yedzq13sa1 \
        --discovery-token-ca-cert-hash sha256:da01744f2144a6c07ee5ab93c2f9f48d2d0cfb197f6ae735798f1f6c812ca52b
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

root@ip-172-31-17-232:/home/ubuntu#
```



```
Machine3~$sudo systemctl stop docker
Machine3~$sudo systemctl stop kubelet
Machine3~$sudo systemctl start docker
Job for docker.service failed.
See "systemctl status docker.service" and "journalctl -xeu docker.service" for details.
Machine3~$sudo systemctl start kubelet
Machine3~$sudo systemctl start docker
Machine3~$sudo netstat -pnlt | grep 6443
tcp6       0      0 :::6443                 :::*                    LISTEN      47151/kube-apiserve
Machine3~$kubectl get nodes
Get "https://172.31.16.118:6443/api/v1/nodes?limit=500": dial tcp 172.31.16.118:6443: connect: connection refused - error from a previous attempt: http2: server sent GOAWAY and closed the connection; LastStr
eamID=1, ErrCode=NO_ERROR, debug=""
Machine3~$kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-16-118   Ready    control-plane   47m   v1.28.2
ip-172-31-17-232   Ready    <none>          44m   v1.28.2
ip-172-31-27-185   Ready    <none>          44m   v1.28.2
Machine3~$
```

7. Now Setting Jenkins



Getting Started

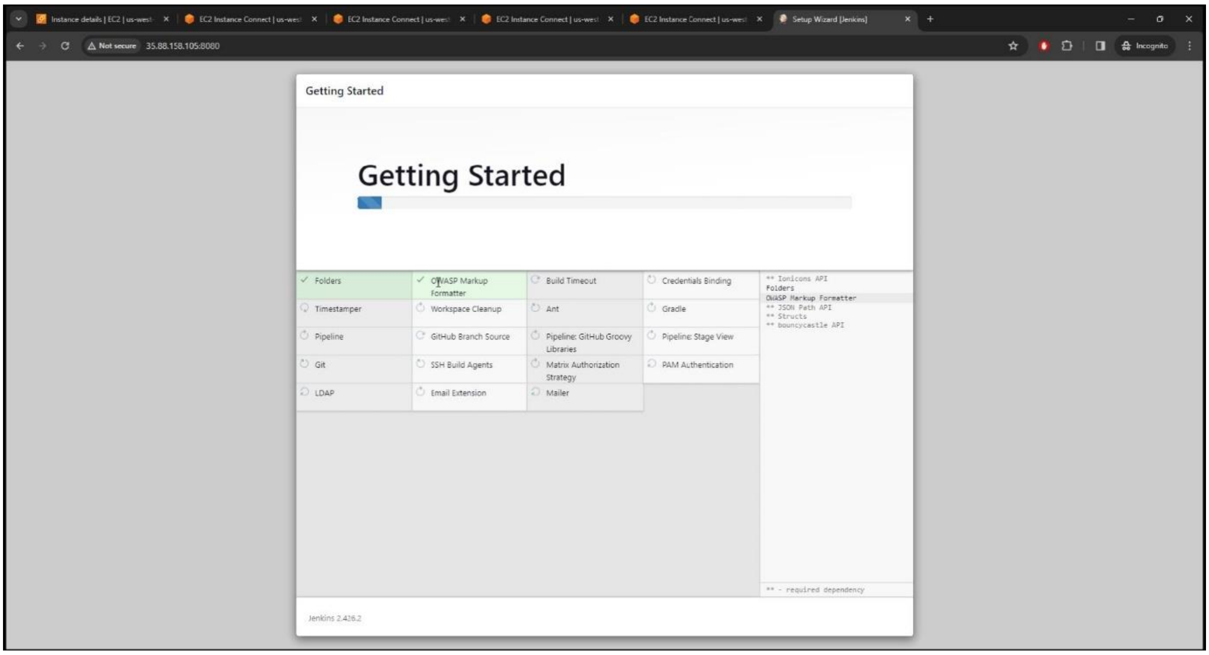
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password



Getting Started

Getting Started

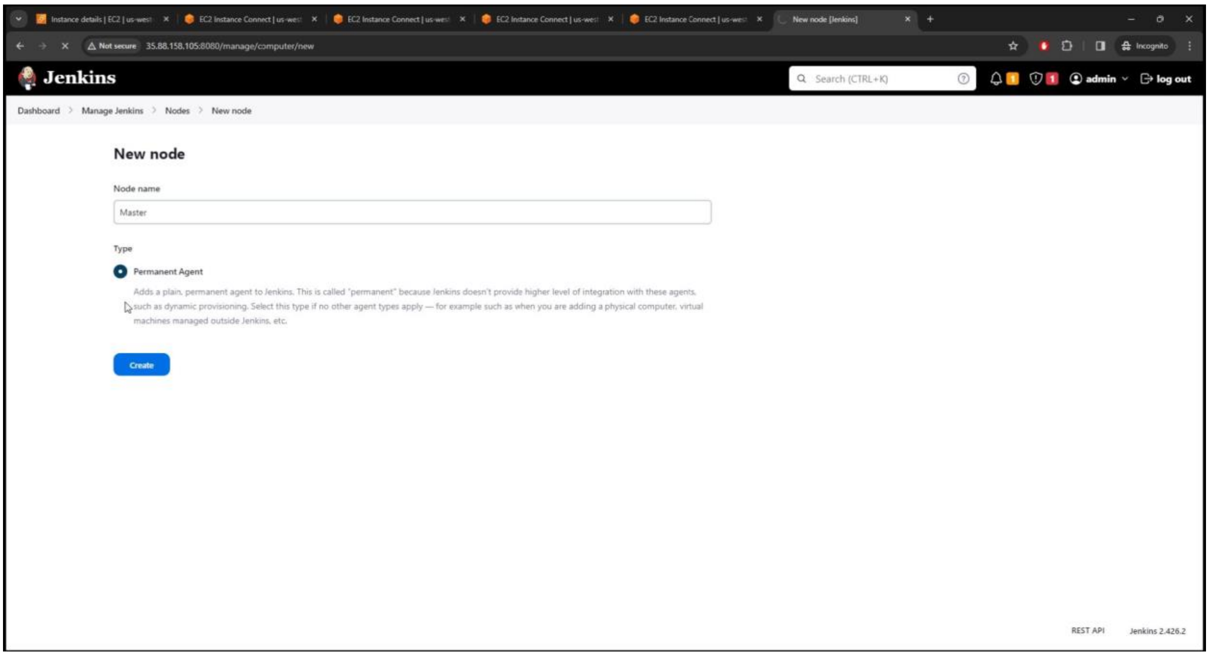
✓ Folders	✓ OWASP Markup Formatter	Build Timeout	Credentials Binding	
Timestamper	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		

```
** Ionicons API
Folders
OWASP Markup Formatter
** JSON Path API
** Structs
** bouncycastle API
```

```
** - required dependency
```

Jenkins 2.426.2

8. Adding Machine 3 as a node.



Dashboard > Manage Jenkins > Nodes > New node

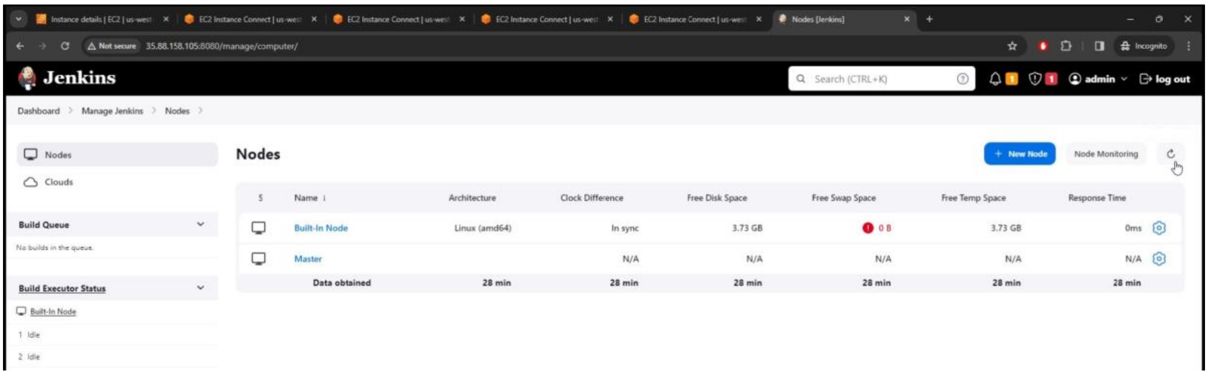
New node

Node name: Master

Type

● Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.



Dashboard > Manage Jenkins > Nodes >

Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.73 GB	0 B	3.73 GB	0ms
	Master		N/A	N/A	N/A	N/A	N/A
	Data obtained	28 min	28 min	28 min	28 min	28 min	28 min

Build Queue

No builds in the queue.

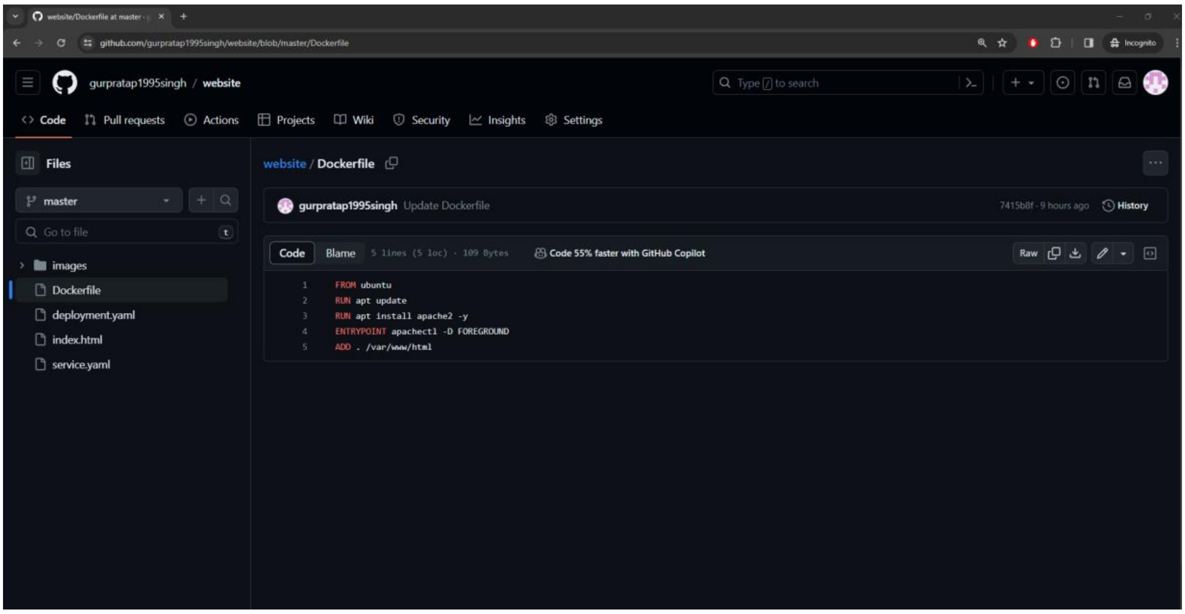
Build Executor Status

Built-In Node

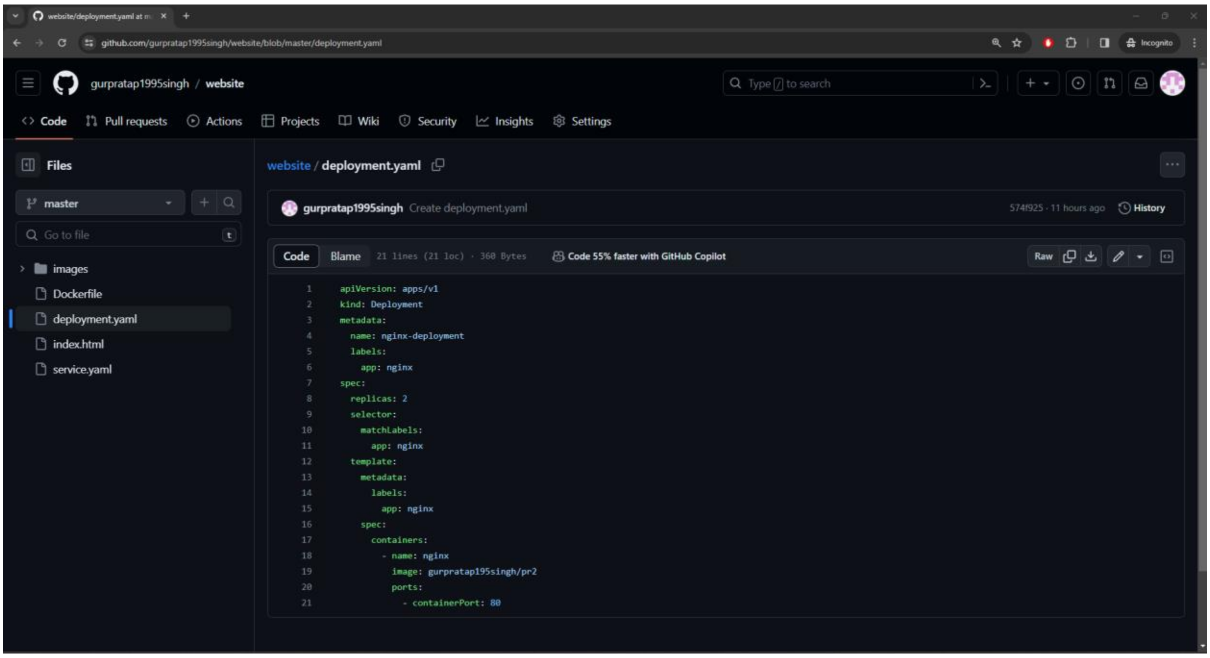
1 Idle

2 Idle

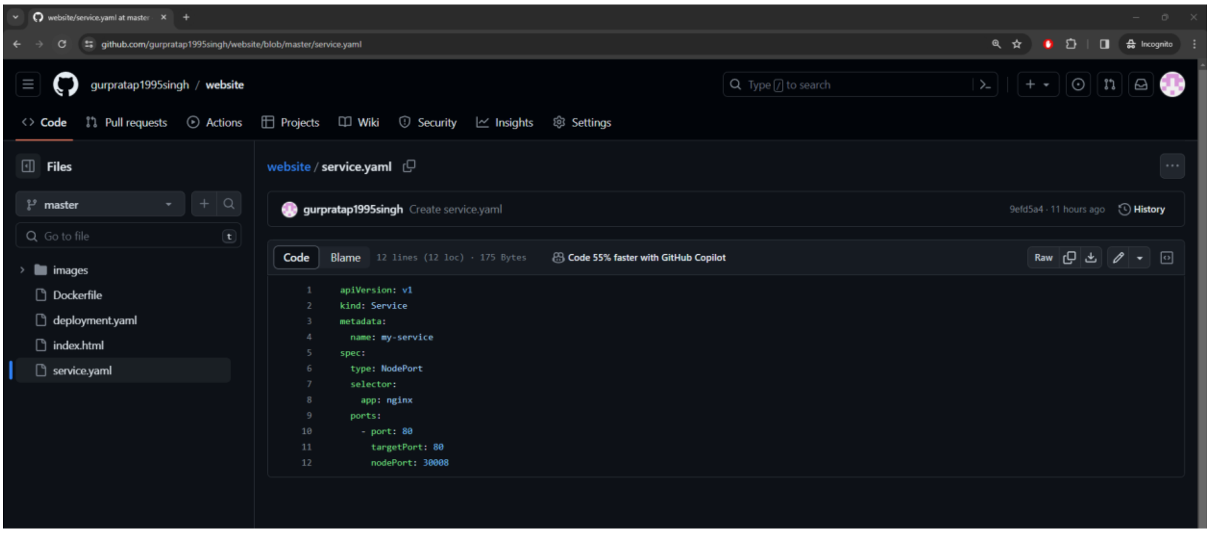
9. Cloning GIT repository and adding Dockerfile, deployment.yaml and service.yaml



```
FROM ubuntu
RUN apt update
RUN apt install apache2 -y
ENTRYPOINT apachectl -D FOREGROUND
ADD . /var/www/html
```

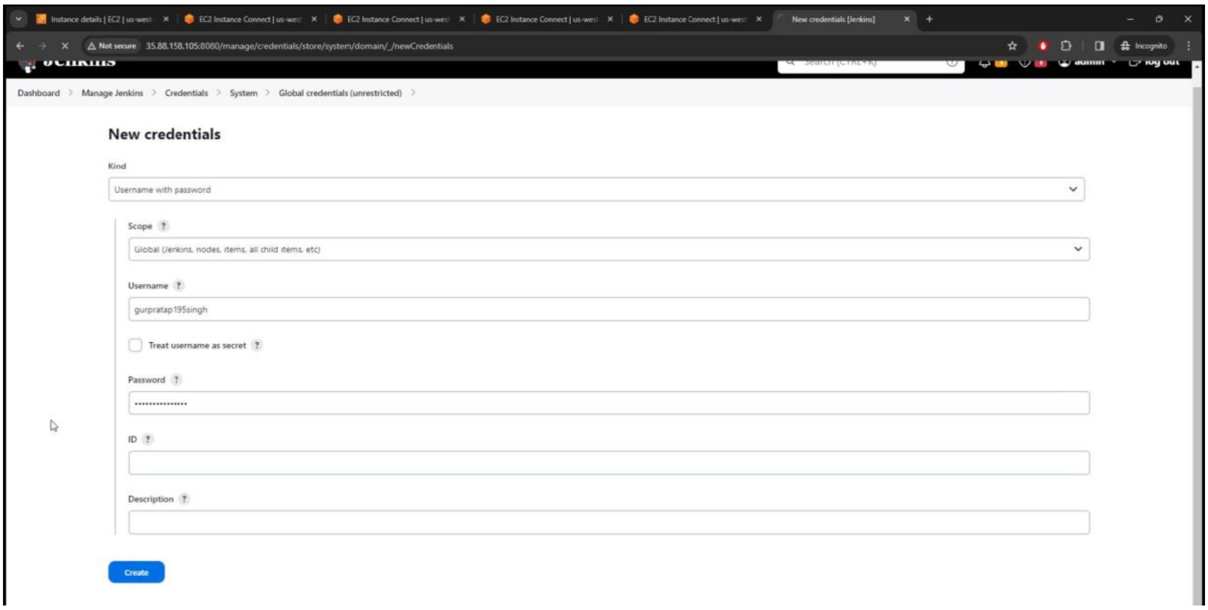

github.com/gurpratap1995singh/website/blob/master/deployment.yaml
gurpratap1995singh / website
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Files
master
Go to file
images
Dockerfile
deployment.yaml
index.html
service.yaml
website / deployment.yaml
gurpratap1995singh Create deployment.yaml
574f925 · 11 hours ago
History
Code
Blame
21 lines (21 loc) · 360 Bytes
Code 55% faster with GitHub Copilot
Raw

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: gurpratap195singh/pr2
          ports:
            - containerPort: 80
```



```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30008
```

10. Adding Credentials



Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: gurpratap195singh

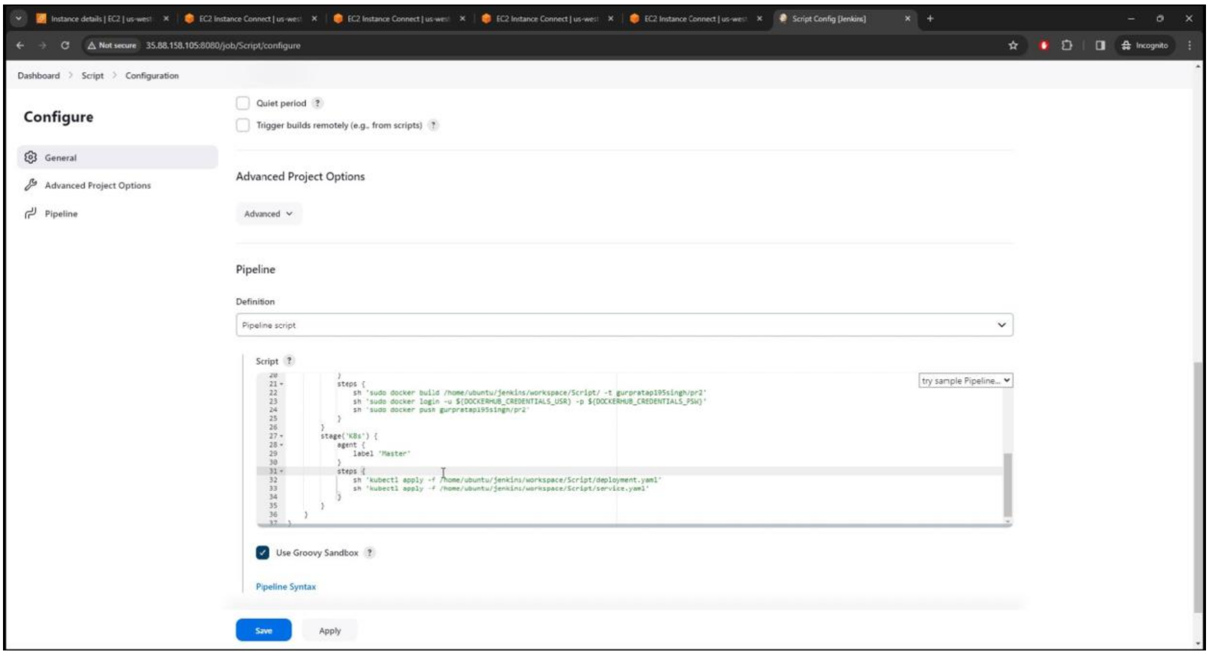
□ Treat username as secret

Password: •••••••••••••••

ID:

Description:

11. Creating Job (Pipeline script).



Dashboard > Script > Configuration

□ Quiet period

□ Trigger builds remotely (e.g., from scripts)

Advanced Project Options

Pipeline

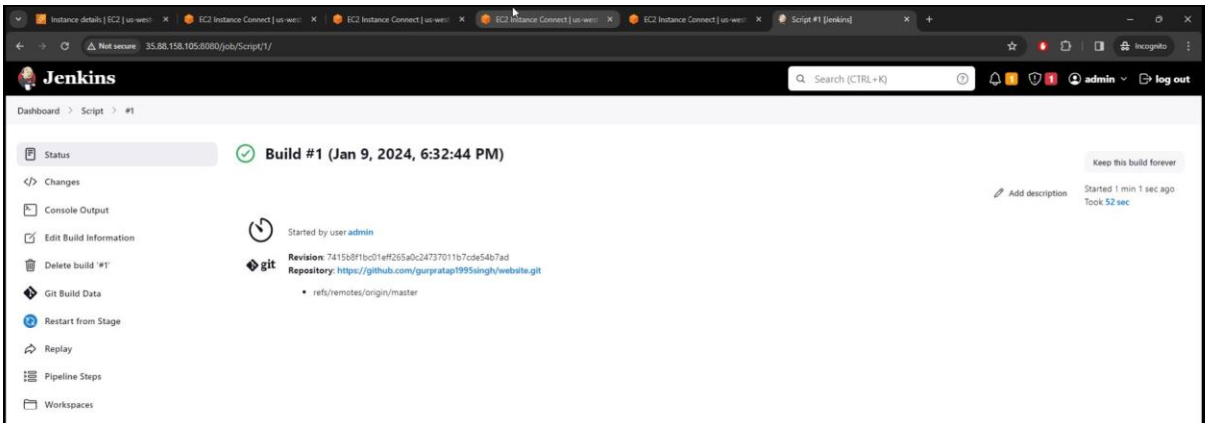
Definition: Pipeline script

Script

```
            }
            steps {
                sh 'sudo docker build /home/ubuntu/jenkins/workspace/Script/ -t gurpratap195singh/pr2'
                sh 'sudo docker login -u ${DOCKERHUB_CREDENTIALS_USR} -p ${DOCKERHUB_CREDENTIALS_PSW}'
                sh 'sudo docker push gurpratap195singh/pr2'
            }
        }
        stage('K8s') {
            agent {
                label 'Master'
            }
            steps {
                sh 'kubectl apply -f /home/ubuntu/jenkins/workspace/Script/deployment.yaml'
                sh 'kubectl apply -f /home/ubuntu/jenkins/workspace/Script/service.yaml'
            }
        }
    }
}
```

☑ Use Groovy Sandbox

12. Running the Job.



Dashboard > Script > #1

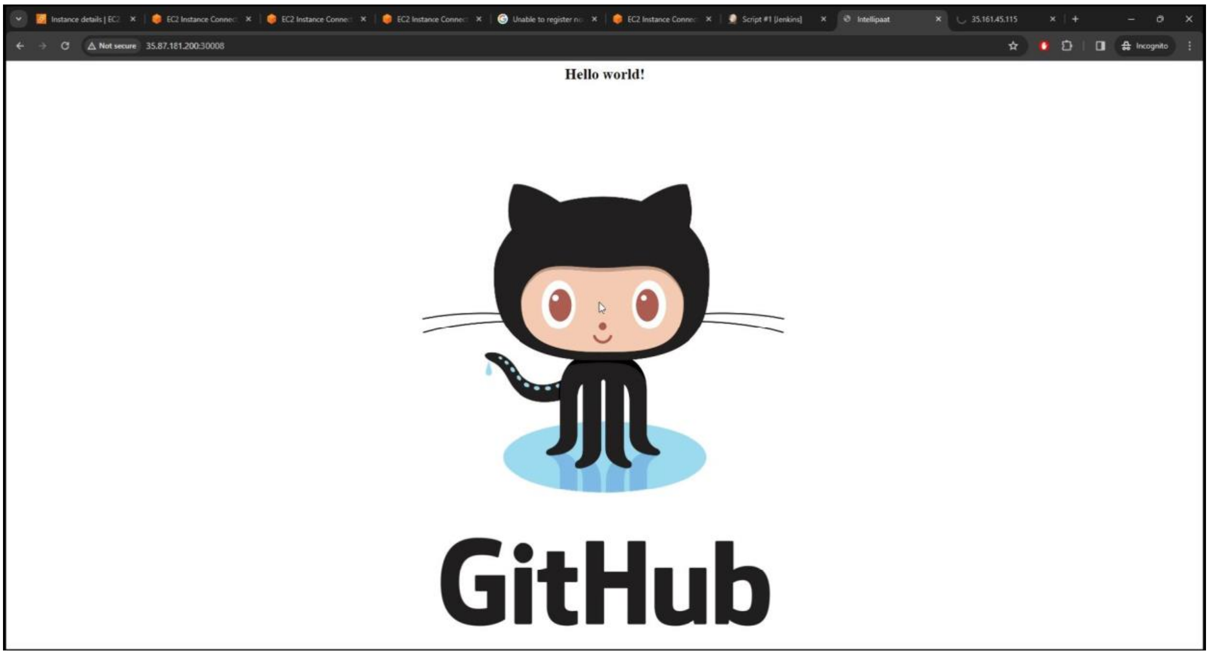
Build #1 (Jan 9, 2024, 6:32:44 PM)

Started 1 min 1 sec ago
Took 52 sec

Started by user admin

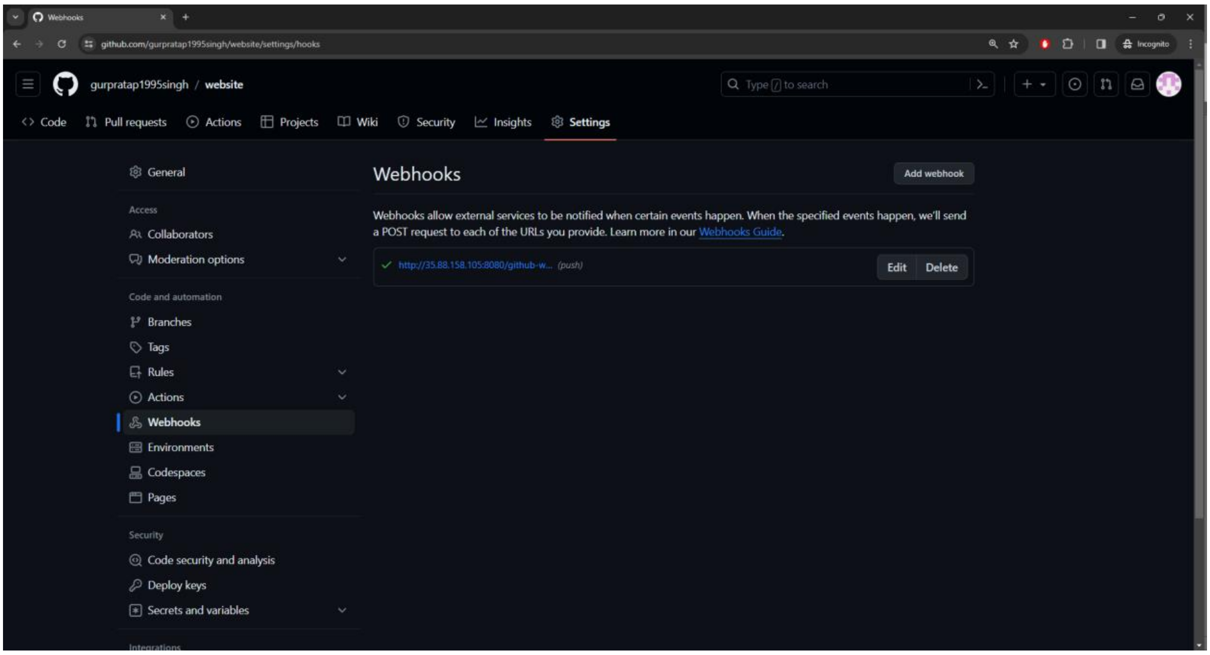
Revision: 7415b8f1bc01eff265a0c24737011b7cde54b7ad
Repository: https://github.com/gurpratap1995singh/website.git

- refs/remotes/origin/master



Hello world!

13. Now adding the GitHub Webhook to auto trigger the job.

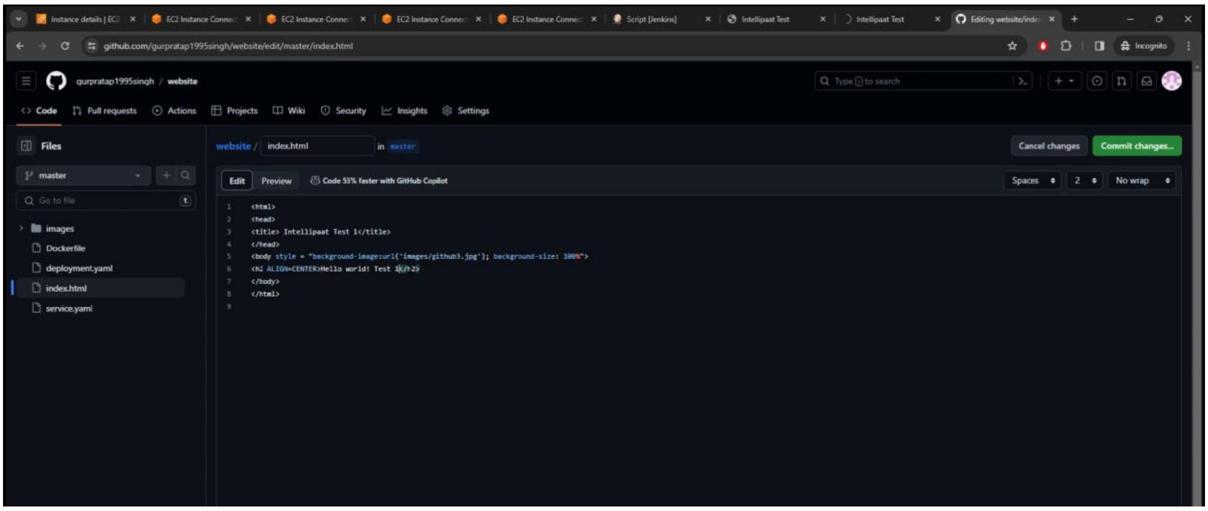


Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

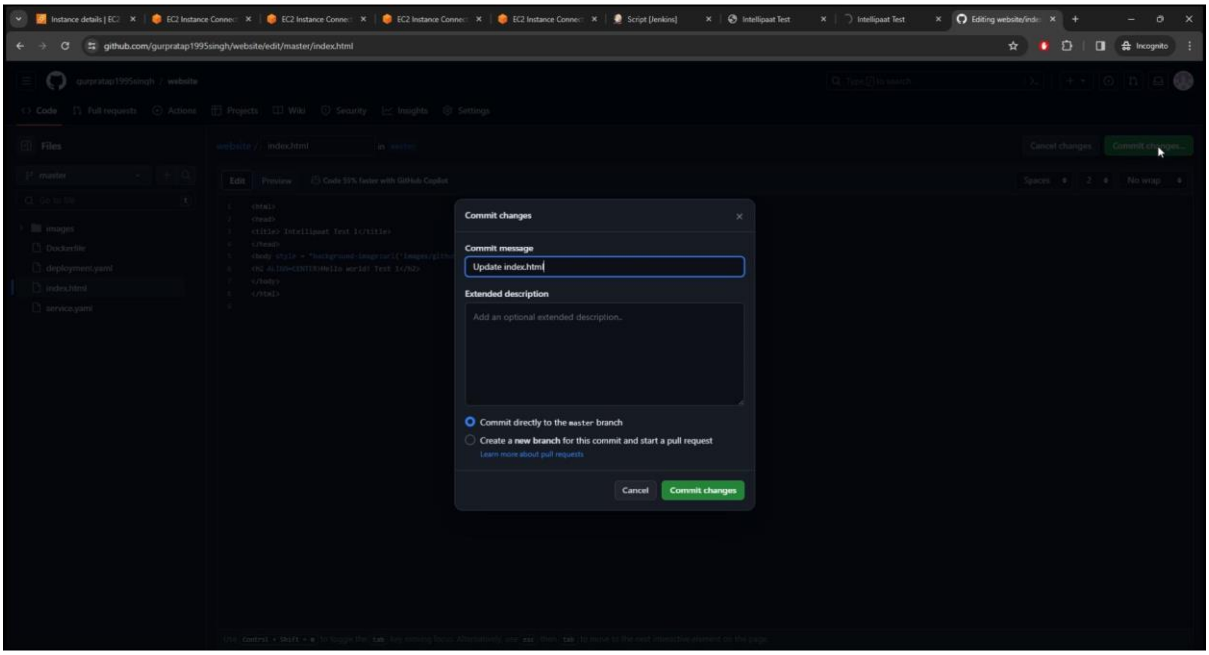
✓ http://35.88.158.105:8080/github-w... (push)

14. Changing in the index.html to verify.



website / index.html in master

```
<html>
<head>
<title> Intellipaat Test 1</title>
</head>
<body style = "background-image:url('images/github3.jpg'); background-size: 100%">
<h2 ALIGN=CENTER>Hello world! Test 1</h2>
</body>
</html>
```



```
<html>
<head>
<title> Intellipaat Test 1</title>
</head>
<body style = "background-image:url('images/githu
<h2 ALIGN=CENTER>Hello world! Test 1</h2>
</body>
</html>
```

Commit changes

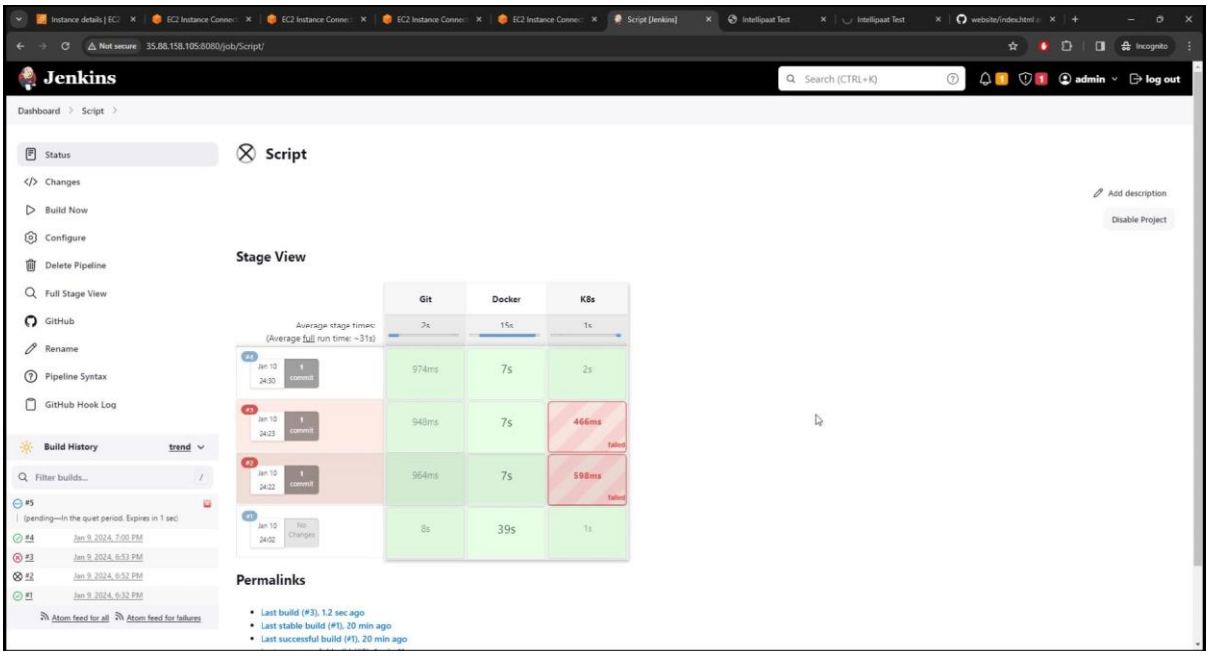
Commit message: Update index.html

Extended description: Add an optional extended description...

● Commit directly to the master branch

○ Create a **new branch** for this commit and start a pull request

Learn more about pull requests



Script

Stage View

	Git	Docker	K8s
Average stage times: (Average full run time: ~31s)	2s	15s	1s
#4 Jan 10 24:30 1 commit	974ms	7s	2s
#3 Jan 10 24:23 1 commit	948ms	7s	466ms failed
#2 Jan 10 24:22 1 commit	964ms	7s	598ms failed
#1 Jan 10 24:02 No Changes	8s	39s	1s

Build History trend

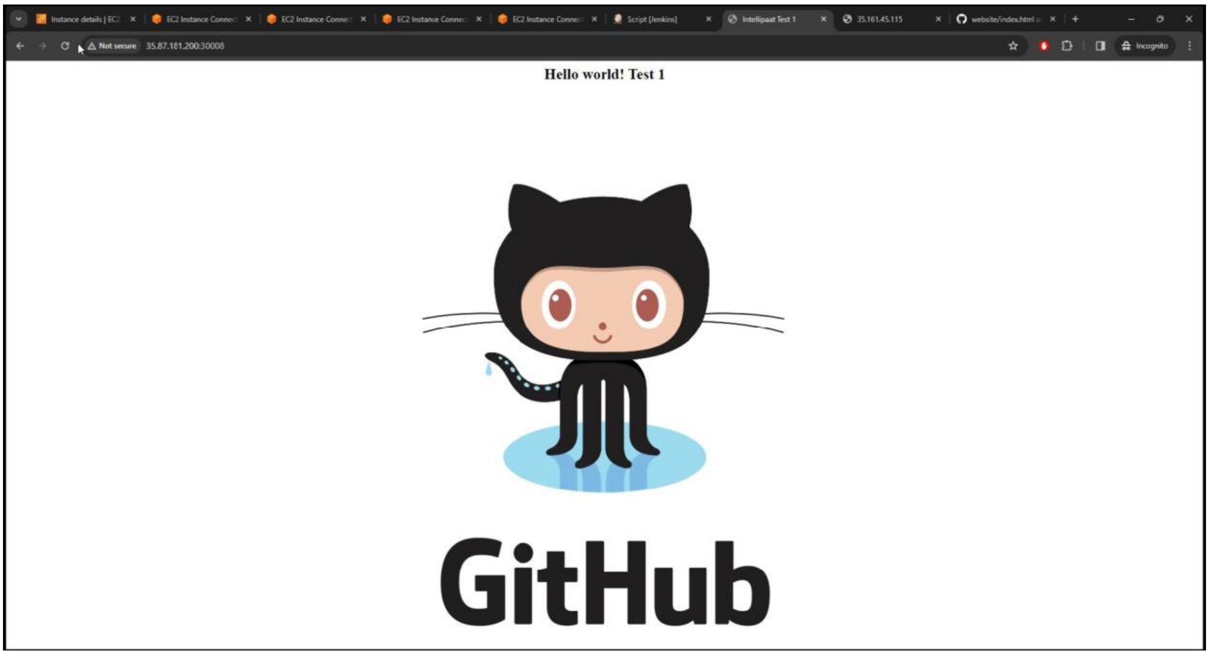
Filter builds...

- #5 (pending—In the quiet period. Expires in 1 sec)
- #4 Jan 9, 2024, 7:00 PM
- #3 Jan 9, 2024, 6:53 PM
- #2 Jan 9, 2024, 6:52 PM
- #1 Jan 9, 2024, 6:32 PM

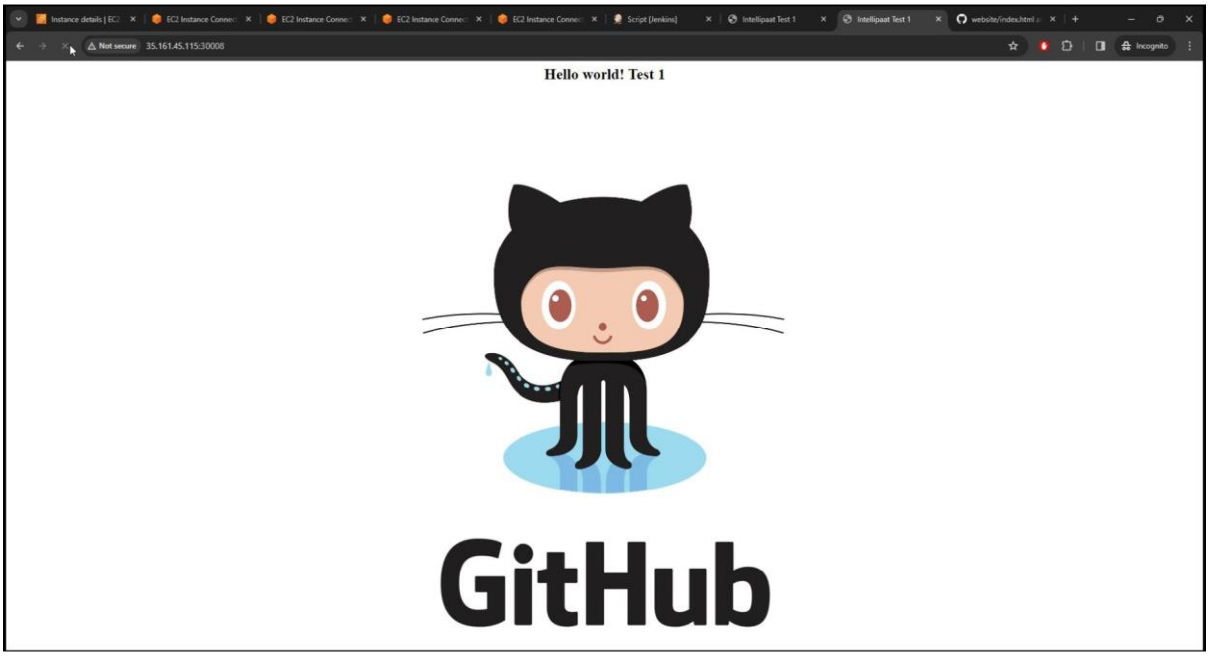
Atom feed for all Atom feed for failures

Permalinks

- Last build (#3), 1.2 sec ago
- Last stable build (#1), 20 min ago
- Last successful build (#1), 20 min ago



Hello world! Test 1



Hello world! Test 1