Project : Capstone II

- Ruchitha Bt

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

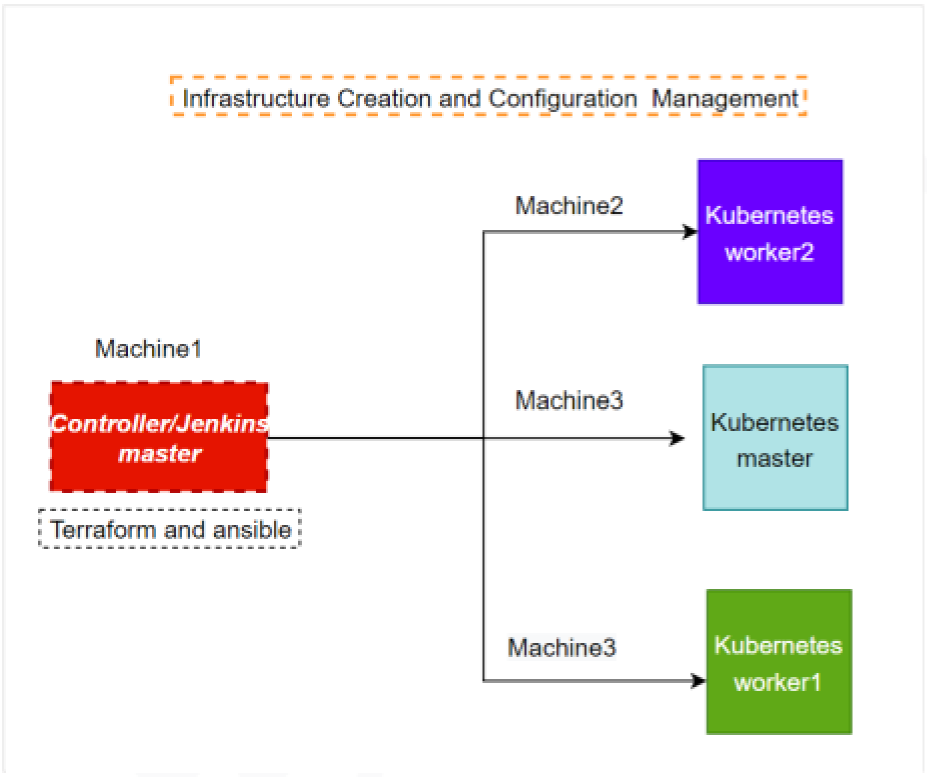
Following are the specifications of the lifecycle:

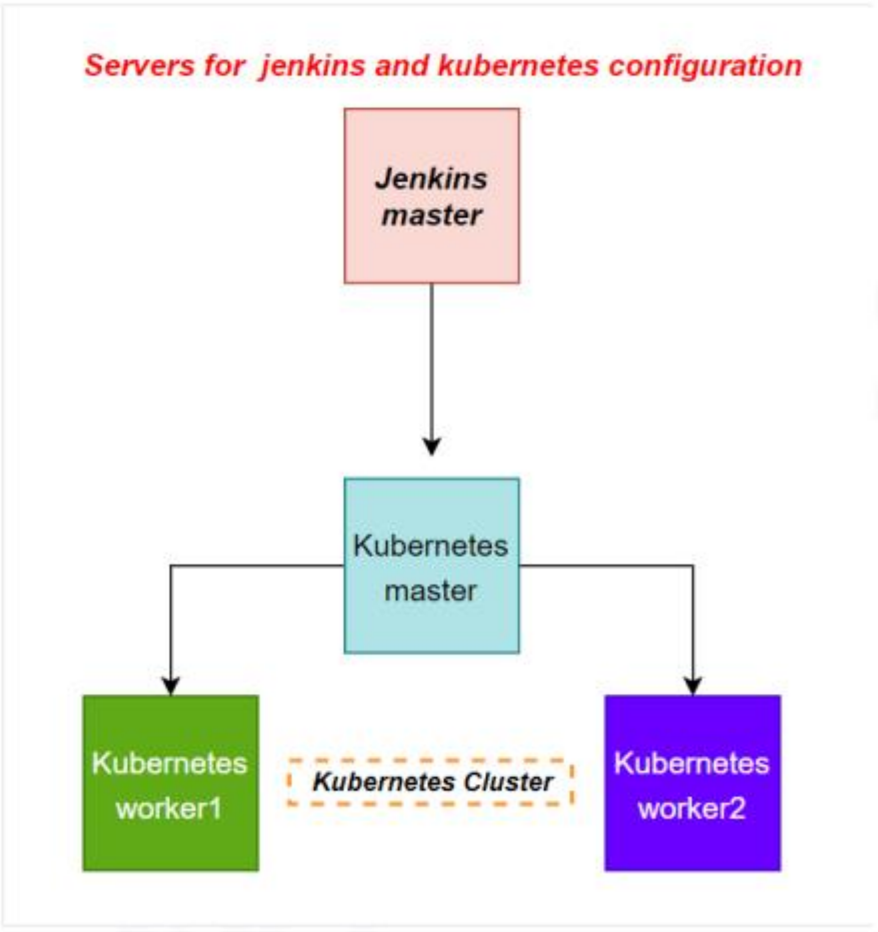
1. Git workflow should be implemented. Since the company follows a monolithic architecture of development, you need to take care of version control. The release should happen only on the 25th of every month.
2. CodeBuild should be triggered once the commits are made in the master branch.
3. The code 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:
Softwares to be installed on the respective machines using configuration management.

Worker1: Jenkins, Java

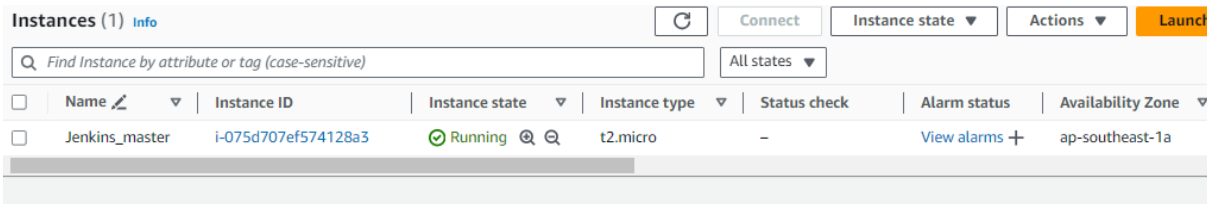
Worker2: Docker, Kubernetes
Worker3: Java, Docker, Kubernetes
Worker4: Docker, Kubernetes


Infrastructure Creation and Configuration Management
Machine1
Controller/Jenkins master
Terraform and ansible
Machine2
Kubernetes worker2
Machine3
Kubernetes master
Machine3
Kubernetes worker1


Servers for jenkins and kubernetes configuration
Jenkins master
Kubernetes master
Kubernetes worker1
Kubernetes Cluster
Kubernetes worker2

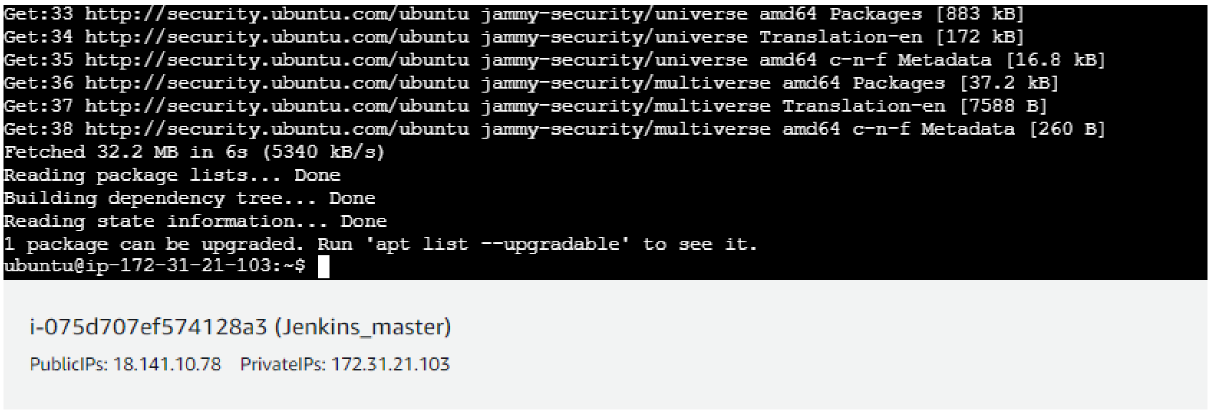
Step1:

Create an instance for Jenkins master



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Jenkins_master	i-075d707ef574128a3	Running	t2.micro	–	View alarms +	ap-southeast-1a

Connect the instance and update the machine

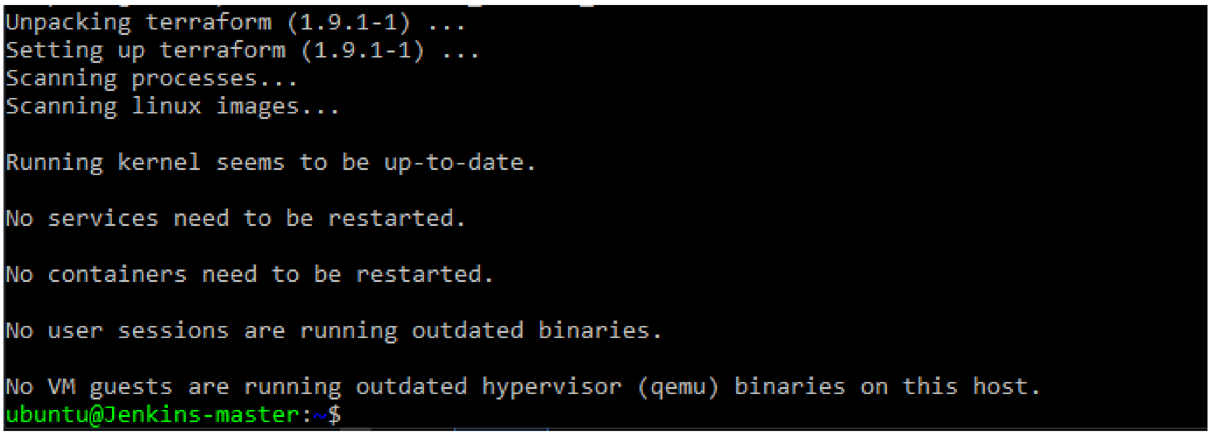


```
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [883 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [172 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7588 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 32.2 MB in 6s (5340 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
ubuntu@ip-172-31-21-103:~$
```

i-075d707ef574128a3 (Jenkins_master)

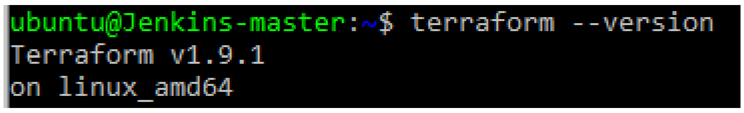
PublicIPs: 18.141.10.78 PrivateIPs: 172.31.21.103

Step 2: Install terraform in Jenkins master



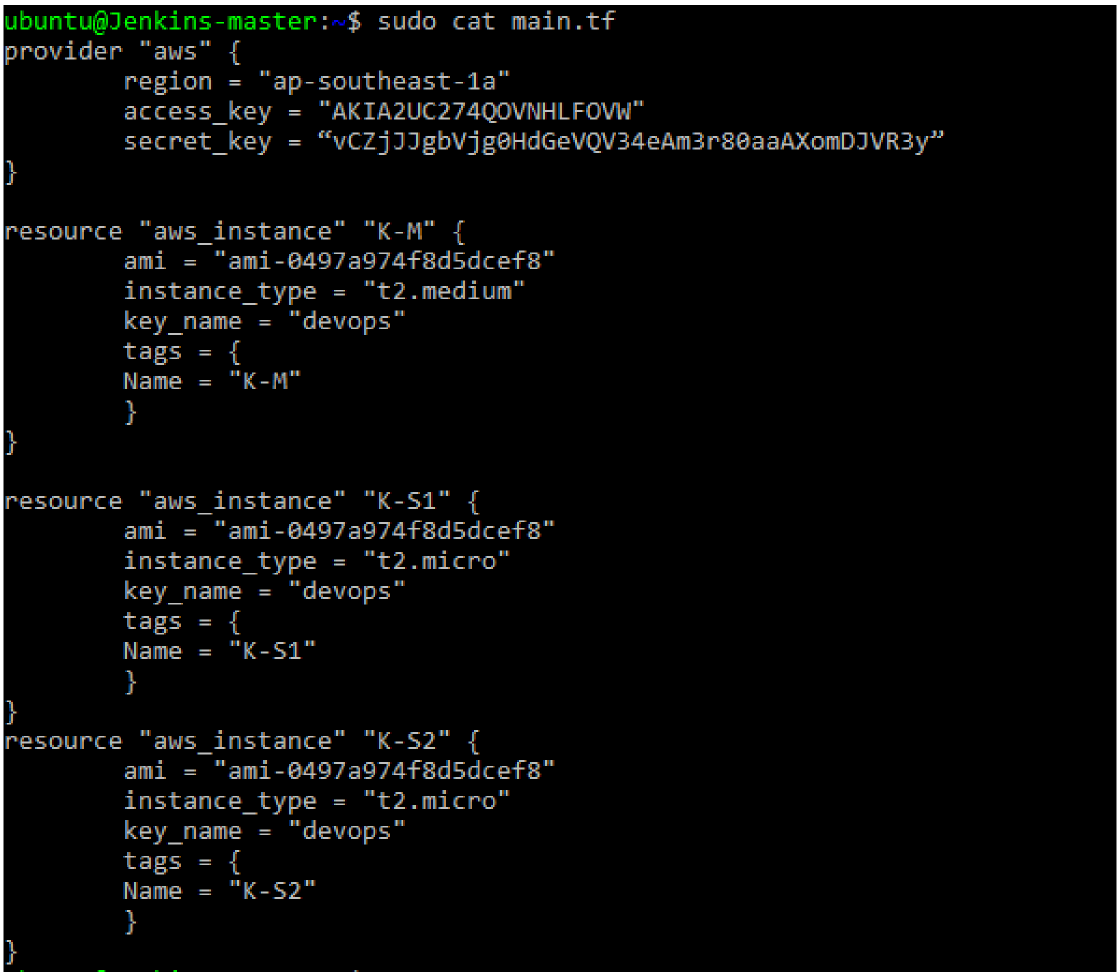
```
Unpacking terraform (1.9.1-1) ...
Setting up terraform (1.9.1-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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.
ubuntu@Jenkins-master:~$
```

Check the version



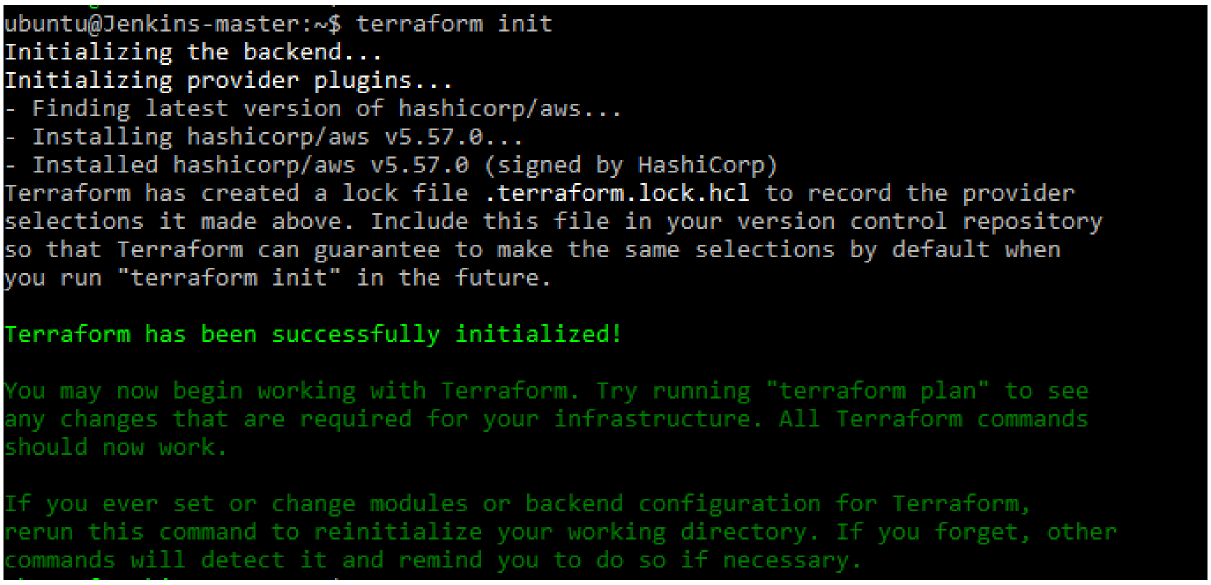
```
ubuntu@Jenkins-master:~$ terraform --version
Terraform v1.9.1
on linux_amd64
```

Create one tf file to install kubernetes master and slaves



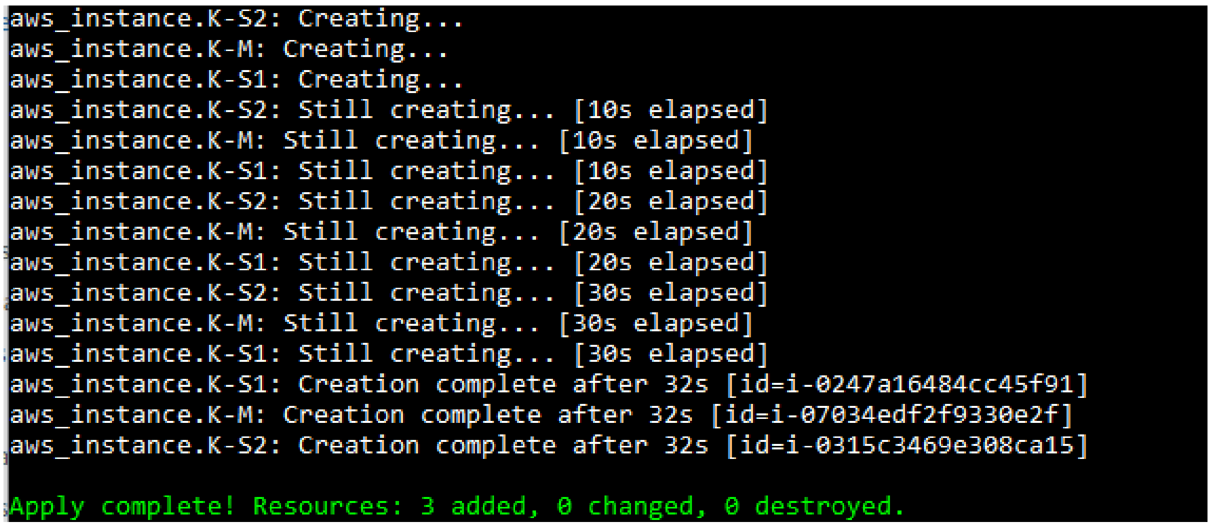
```
ubuntu@Jenkins-master:~$ sudo cat main.tf
provider "aws" {
        region = "ap-southeast-1a"
        access_key = "AKIA2UC274QOVNHLFOVW"
        secret_key = “vCZjJJgbVjg0HdGeVQV34eAm3r80aaAXomDJVR3y”
}

resource "aws_instance" "K-M" {
        ami = "ami-0497a974f8d5dcef8"
        instance_type = "t2.medium"
        key_name = "devops"
        tags = {
        Name = "K-M"
        }
}

resource "aws_instance" "K-S1" {
        ami = "ami-0497a974f8d5dcef8"
        instance_type = "t2.micro"
        key_name = "devops"
        tags = {
        Name = "K-S1"
        }
}
resource "aws_instance" "K-S2" {
        ami = "ami-0497a974f8d5dcef8"
        instance_type = "t2.micro"
        key_name = "devops"
        tags = {
        Name = "K-S2"
        }
}
```

Initialize the terraform



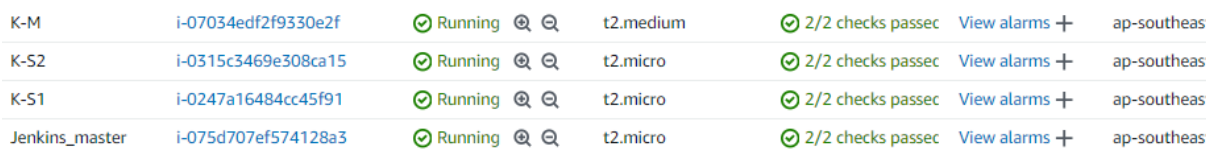
```
ubuntu@Jenkins-master:~$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.57.0...
- Installed hashicorp/aws v5.57.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to 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.
```

Plan and apply the terraform file.



```
aws_instance.K-S2: Creating...
aws_instance.K-M: Creating...
aws_instance.K-S1: Creating...
aws_instance.K-S2: Still creating... [10s elapsed]
aws_instance.K-M: Still creating... [10s elapsed]
aws_instance.K-S1: Still creating... [10s elapsed]
aws_instance.K-S2: Still creating... [20s elapsed]
aws_instance.K-M: Still creating... [20s elapsed]
aws_instance.K-S1: Still creating... [20s elapsed]
aws_instance.K-S2: Still creating... [30s elapsed]
aws_instance.K-M: Still creating... [30s elapsed]
aws_instance.K-S1: Still creating... [30s elapsed]
aws_instance.K-S1: Creation complete after 32s [id=i-0247a16484cc45f91]
aws_instance.K-M: Creation complete after 32s [id=i-07034edf2f9330e2f]
aws_instance.K-S2: Creation complete after 32s [id=i-0315c3469e308ca15]

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

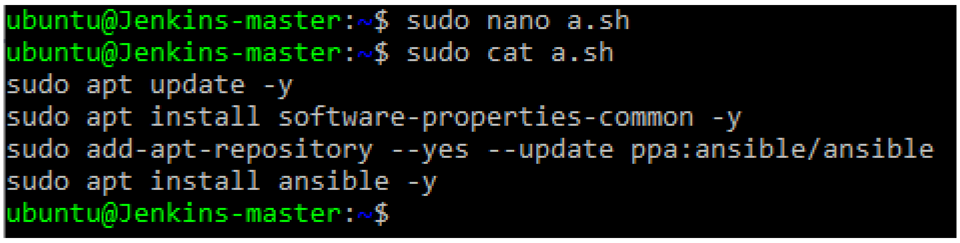
Check the instances created, connect and update the machine



K-M	i-07034edf2f9330e2f	Running	t2.medium	2/2 checks passec	View alarms +	ap-southeas
K-S2	i-0315c3469e308ca15	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeas
K-S1	i-0247a16484cc45f91	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeas
Jenkins_master	i-075d707ef574128a3	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeas

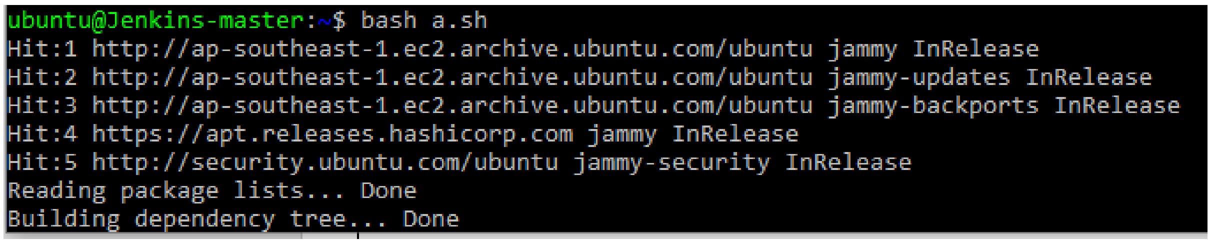
Step 3: Ansible installation

Create a script file and give the command of ansible installation



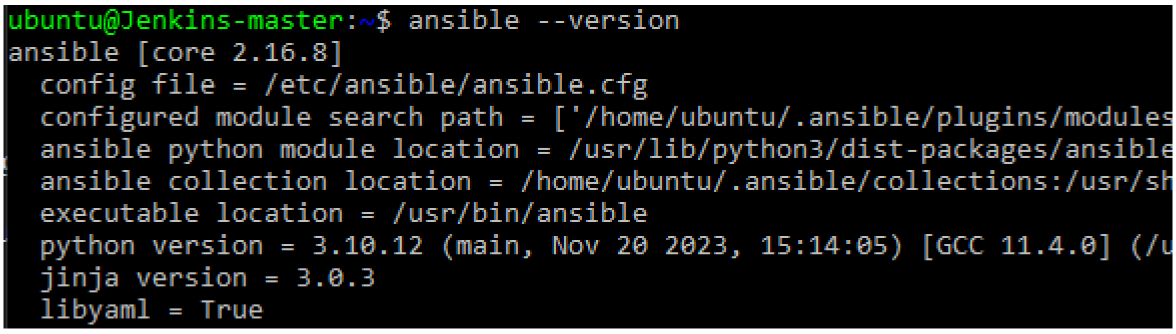
```
ubuntu@Jenkins-master:~$ sudo nano a.sh
ubuntu@Jenkins-master:~$ sudo cat a.sh
sudo apt update -y
sudo apt install software-properties-common -y
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y
ubuntu@Jenkins-master:~$
```

Execute the shell script



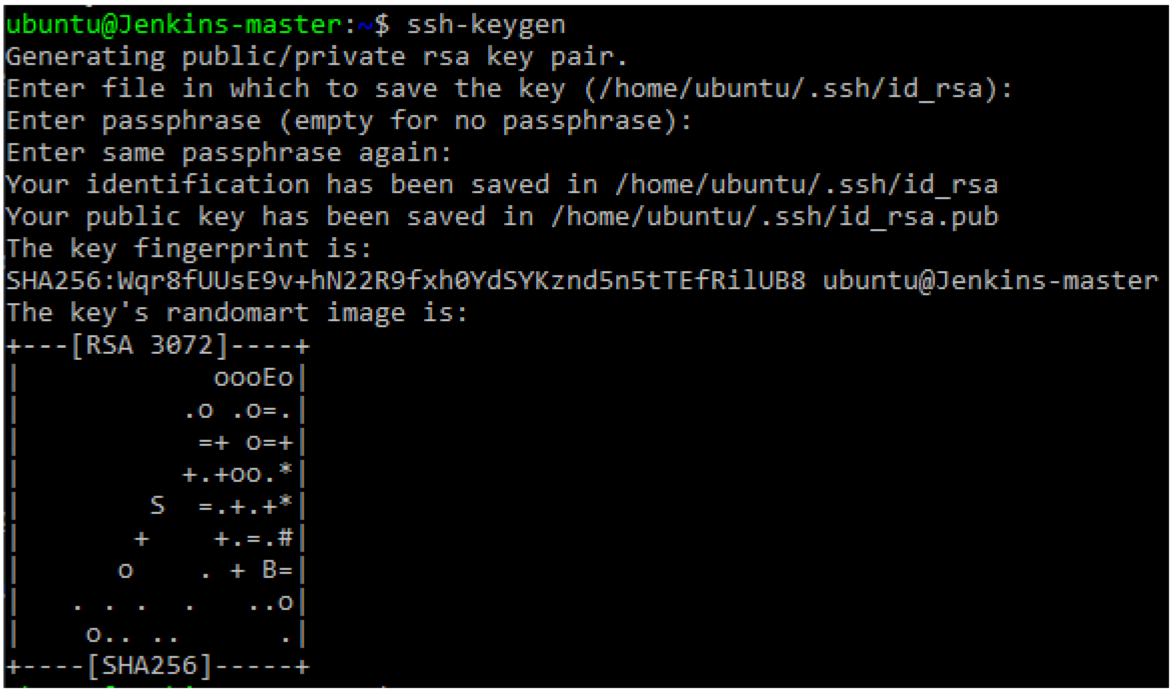
```
ubuntu@Jenkins-master:~$ bash a.sh
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 https://apt.releases.hashicorp.com jammy InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
```

Check the version



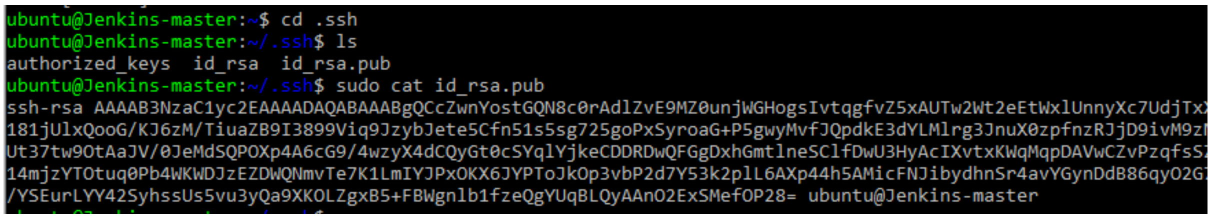
```
ubuntu@Jenkins-master:~$ ansible --version
ansible [core 2.16.8]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/sh
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/u
  jinja version = 3.0.3
  libyaml = True
```

Step 4: Crete keypair



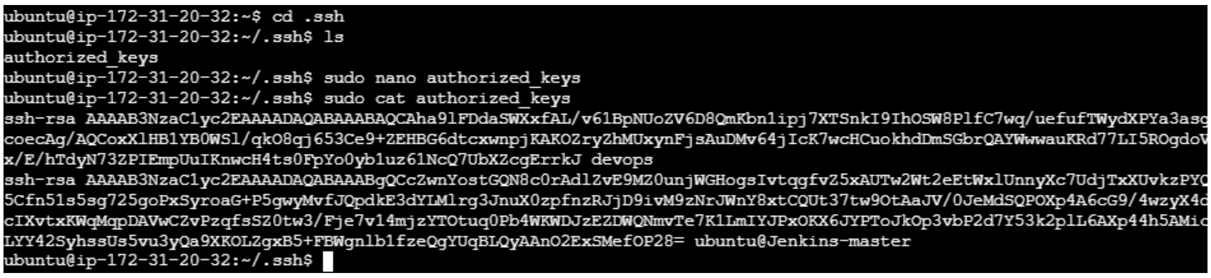
```
ubuntu@Jenkins-master:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Wqr8fUUsE9v+hN22R9fxh0YdSYKznd5n5tTEfRilUB8 ubuntu@Jenkins-master
The key's randomart image is:
+---[RSA 3072]----+
|            oooEo|
|          .o .o=.|
|           =+ o=+|
|          +.+oo.*|
|        S  =.+.+*|
|       +    +.=.#|
|      o    . + B=|
|   . . .  .   ..o|
|    o.. ..      .|
+----[SHA256]-----+
```

Go to the ssh directory and copy the public key created



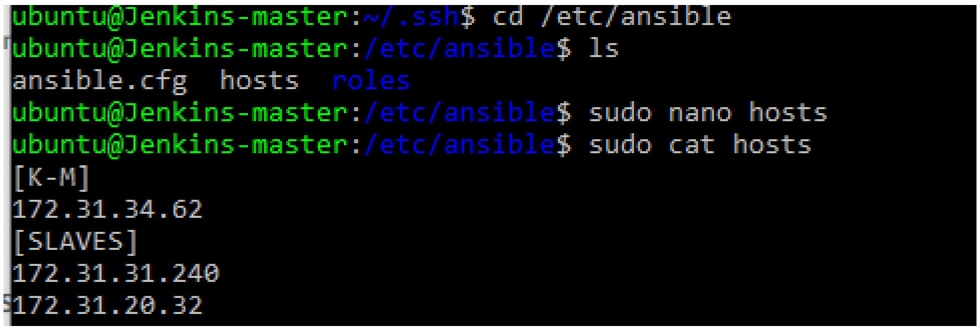
```
ubuntu@Jenkins-master:~$ cd .ssh
ubuntu@Jenkins-master:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@Jenkins-master:~/.ssh$ sudo cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCcZwnYostGQN8c0rAdlZvE9MZ0unjWGHogsIvtqgfvZ5xAUTw2Wt2eEtWxlUnnyXc7UdjTx
181jUlxQooG/KJ6zM/TiuaZB9I3899Viq9JzybJete5Cfn51s5sg725goPxSyroaG+P5gwyMvfJQpdkE3dYLMlrg3JnuX0zpfnzRJjD9ivM9z
Ut37tw9OtAaJV/0JeMdSQPOXp4A6cG9/4wzyX4dCQyGt0cSYqlYjkeCDDRDwQFGgDxhGmtlneSClfDwU3HyAcIXvtxKWqMqpDAVwCZvPzqfsS
14mjzYTOtuq0Pb4WKWDJzEZDWQNmvTe7K1LmIYJPxOKX6JYPToJkOp3vbP2d7Y53k2plL6AXp44h5AMicFNJibydhnSr4avYGynDdB86qyO2G
/YSEurLYY42SyhssUs5vu3yQa9XKOLZgxB5+FBWgnlb1fzeQgYUqBLQyAAnO2ExSMefOP28= ubuntu@Jenkins-master
```

Paste it to the kubernetes master, kubernetes slave1 and slave2

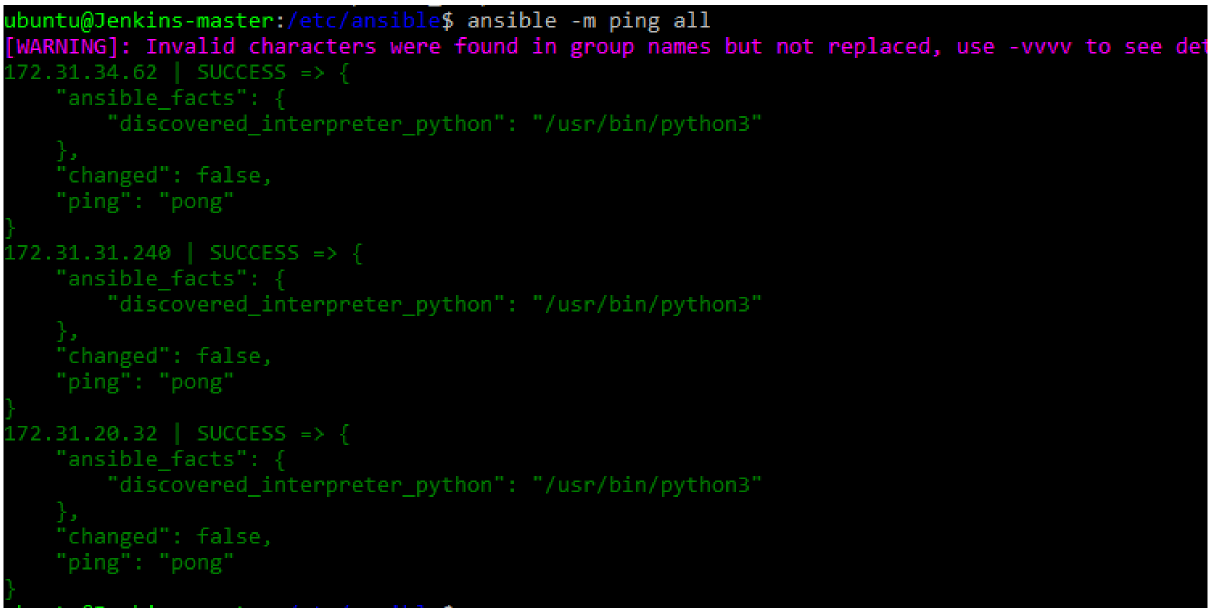

```
ubuntu@ip-172-31-20-32:~$ cd .ssh
ubuntu@ip-172-31-20-32:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-20-32:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-20-32:~/.ssh$ sudo cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCAha91FDdaSWXxfAL/v61BpNUoZV6D8QmKbn1ipj7XTSnkI9IhOSW8PlfC7wq/uefufTWydXPYa3asg
coecAg/AQCoxX1HB1YB0WS1/qkO8qj653Ce9+ZEHBG6dtcxwnpjKAKOZryZhMUxynFjsAuDMv64jIcK7wcHCuokhdDmSGbrQAYWwwauKRd77LI5ROgdoV
x/E/hTdyN73ZPIEmpUuIKnwcH4ts0FpYo0yb1uz61NcQ7UbXZcgErrkJ devops
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCcZwnYostGQN8c0rAdlZvE9MZ0unjWGHogsIvtqgfvZ5xAUTw2Wt2eEtWxlUnnyXc7UdjTxXUvkzPYQ
5Cfn51s5sg725goPxSyroaG+P5gwyMvfJQpdkE3dYLMlrg3JnuX0zpfnzRJjD9ivM9zNrJWnY8xtCQUt37tw9OtAaJV/0JeMdSQPOXp4A6cG9/4wzyX4d
cIXvtxKWqMqpDAVwCZvPzqfsSZ0tw3/Fje7v14mjzYTOtuq0Pb4WKWDJzEZDWQNmvTe7K1LmIYJPxOKX6JYPToJkOp3vbP2d7Y53k2plL6AXp44h5AMic
LYY42SyhssUs5vu3yQa9XKOLZgxB5+FBWgnlb1fzeQgYUqBLQyAAnO2ExSMefOP28= ubuntu@Jenkins-master
ubuntu@ip-172-31-20-32:~/.ssh$
```

Step 5: Fill up the host information in Jenkins master

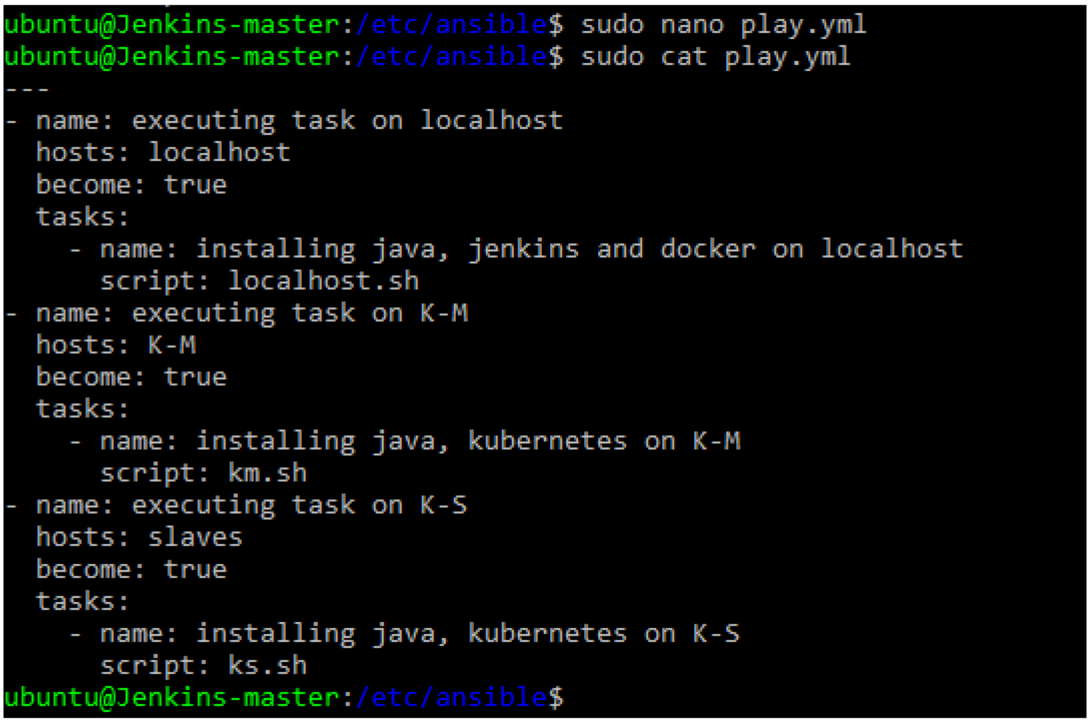
Go to the etc ansible location you will find the hosts file


```
ubuntu@Jenkins-master:~/.ssh$ cd /etc/ansible
ubuntu@Jenkins-master:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@Jenkins-master:/etc/ansible$ sudo nano hosts
ubuntu@Jenkins-master:/etc/ansible$ sudo cat hosts
[K-M]
172.31.34.62
[SLAVES]
172.31.31.240
172.31.20.32
```

Ping check the hosts connected

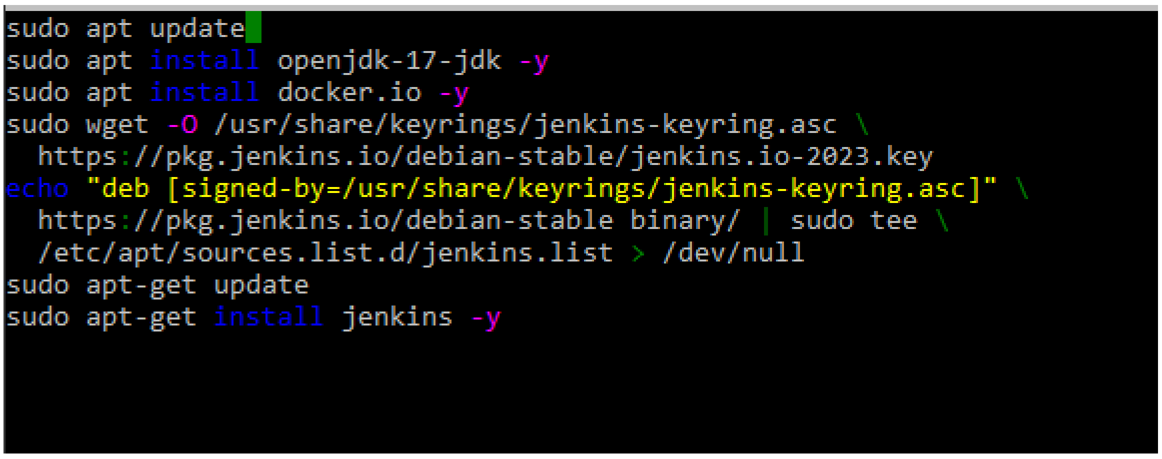

```
ubuntu@Jenkins-master:/etc/ansible$ ansible -m ping all
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see det
172.31.34.62 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.31.240 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.20.32 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Now create the ansible playbook



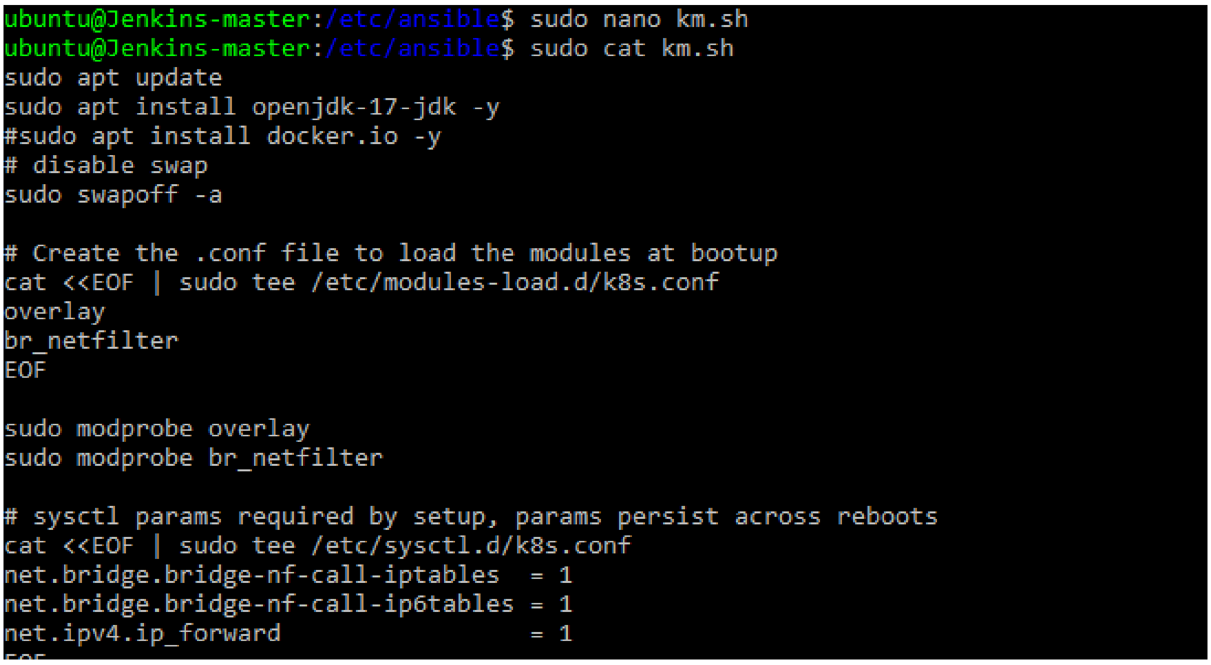
```
ubuntu@Jenkins-master:/etc/ansible$ sudo nano play.yml
ubuntu@Jenkins-master:/etc/ansible$ sudo cat play.yml
---
- name: executing task on localhost
  hosts: localhost
  become: true
  tasks:
    - name: installing java, jenkins and docker on localhost
      script: localhost.sh
- name: executing task on K-M
  hosts: K-M
  become: true
  tasks:
    - name: installing java, kubernetes on K-M
      script: km.sh
- name: executing task on K-S
  hosts: slaves
  become: true
  tasks:
    - name: installing java, kubernetes on K-S
      script: ks.sh
ubuntu@Jenkins-master:/etc/ansible$
```

$ sudo nano localhost.sh



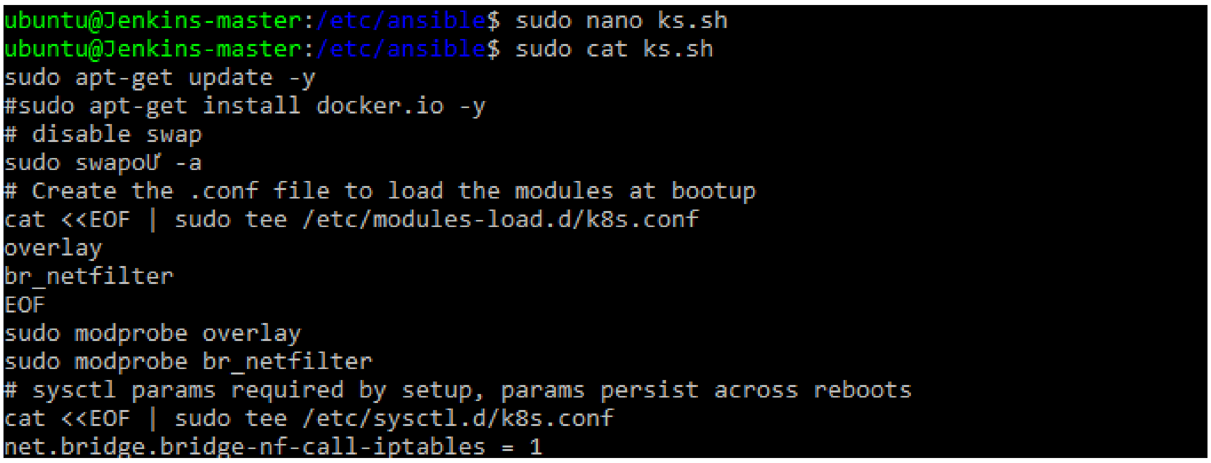
```
sudo apt update
sudo apt install openjdk-17-jdk -y
sudo apt install docker.io -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
```

$ sudo nano km.sh



```
ubuntu@Jenkins-master:/etc/ansible$ sudo nano km.sh
ubuntu@Jenkins-master:/etc/ansible$ sudo cat km.sh
sudo apt update
sudo apt install openjdk-17-jdk -y
#sudo apt install docker.io -y
# disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables  = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward                 = 1
```

$ sudo nan ks.sh



```
ubuntu@Jenkins-master:/etc/ansible$ sudo nano ks.sh
ubuntu@Jenkins-master:/etc/ansible$ sudo cat ks.sh
sudo apt-get update -y
#sudo apt-get install docker.io -y
# disable swap
sudo swapoƯ -a
# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
```

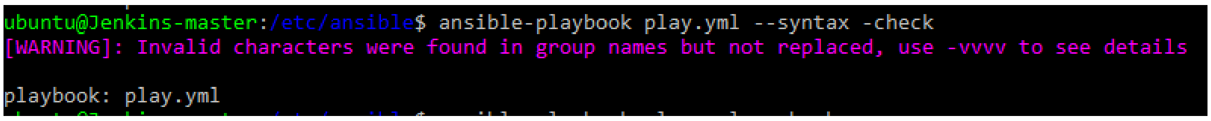
Commands:

```
sudo apt update
sudo apt install openjdk-17-jdk -y
#sudo apt install docker.io -y
# disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
```

```
sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables  = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward                 = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRIO Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-
certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg]
https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee
/etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor
-o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet
```

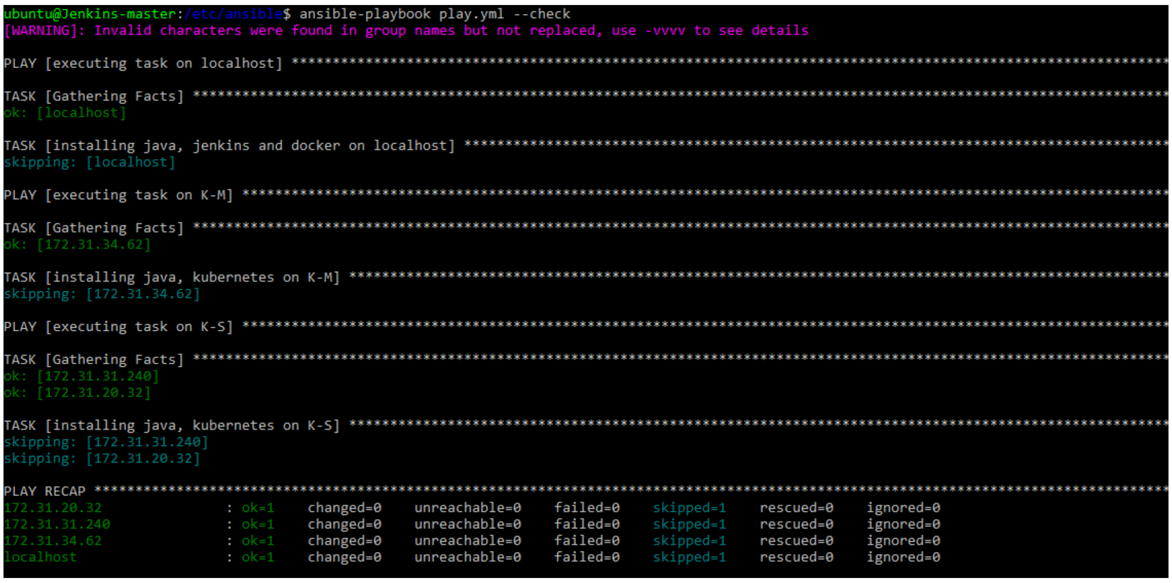
Step 8: Execute the ansible playbook

Do the syntax check



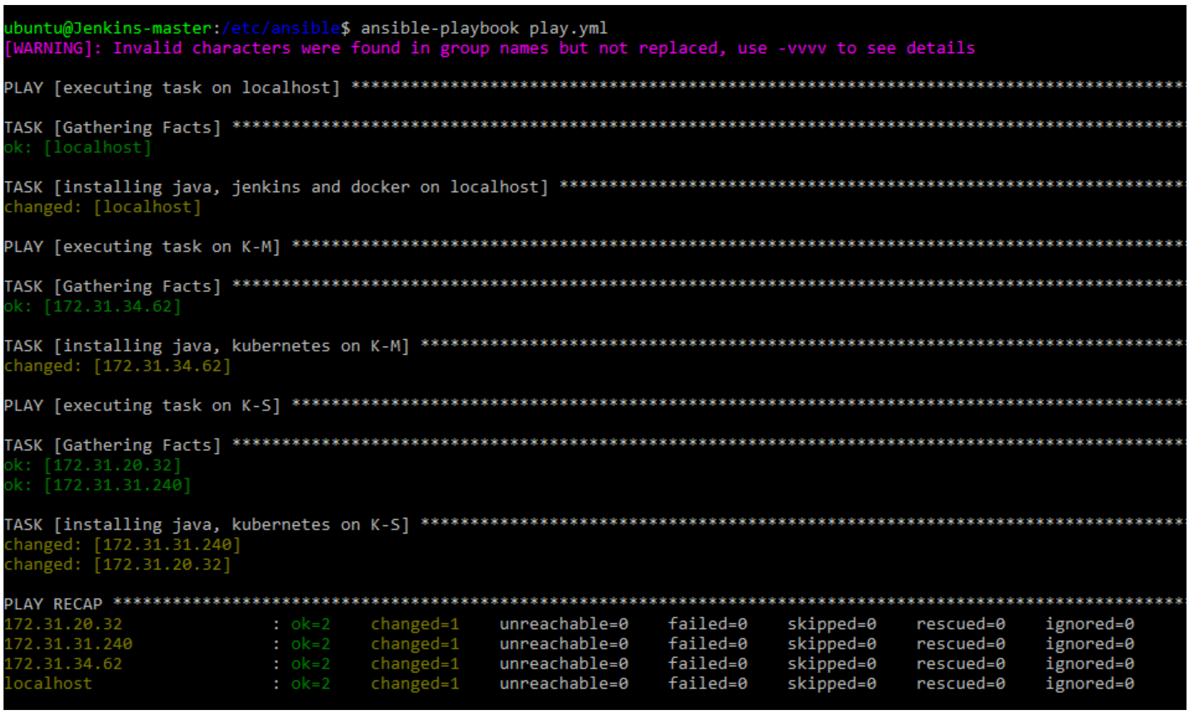
```
ubuntu@Jenkins-master:/etc/ansible$ ansible-playbook play.yml --syntax -check
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

playbook: play.yml
```

Dry run the playbook



```
ubuntu@Jenkins-master:/etc/ansible$ ansible-playbook play.yml --check
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [executing task on localhost] **********************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [localhost]

TASK [installing java, jenkins and docker on localhost] *************************************************************
skipping: [localhost]

PLAY [executing task on K-M] ****************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [172.31.34.62]

TASK [installing java, kubernetes on K-M] ***************************************************************************
skipping: [172.31.34.62]

PLAY [executing task on K-S] ****************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [172.31.31.240]
ok: [172.31.20.32]

TASK [installing java, kubernetes on K-S] ***************************************************************************
skipping: [172.31.31.240]
skipping: [172.31.20.32]

PLAY RECAP **********************************************************************************************************
172.31.20.32               : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.31.31.240              : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.31.34.62               : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Perform the actual run

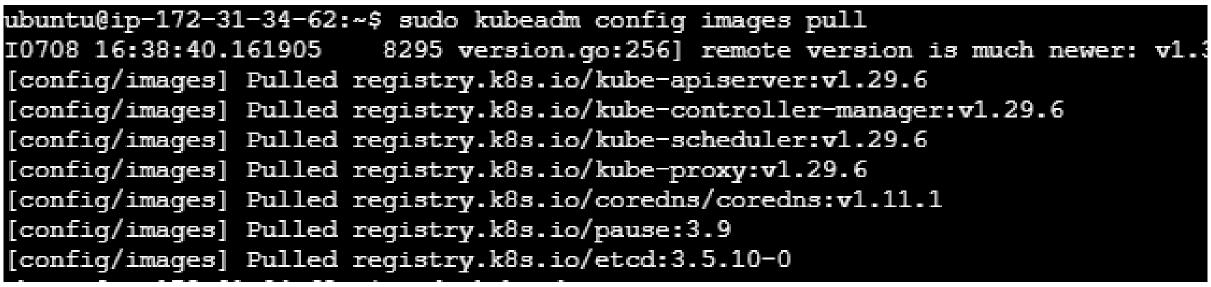


```
ubuntu@Jenkins-master:/etc/ansible$ ansible-playbook play.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [executing task on localhost] **********************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [localhost]

TASK [installing java, jenkins and docker on localhost] *************************************************************
changed: [localhost]

PLAY [executing task on K-M] ****************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [172.31.34.62]

TASK [installing java, kubernetes on K-M] ***************************************************************************
changed: [172.31.34.62]

PLAY [executing task on K-S] ****************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [172.31.20.32]
ok: [172.31.31.240]

TASK [installing java, kubernetes on K-S] ***************************************************************************
changed: [172.31.31.240]
changed: [172.31.20.32]

PLAY RECAP **********************************************************************************************************
172.31.20.32               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.31.240              : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.34.62               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Step 9: Creating kubeadm cluster in kubernetes master node

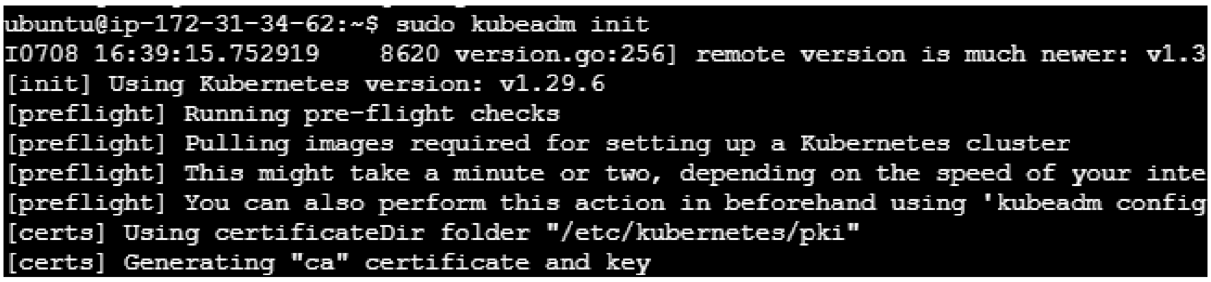
# Execute ONLY on "Master Node"

$ sudo kubeadm config images pull



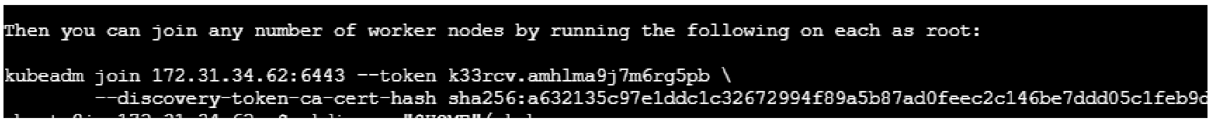
```
ubuntu@ip-172-31-34-62:~$ sudo kubeadm config images pull
I0708 16:38:40.161905    8295 version.go:256] remote version is much newer: v1.3
[config/images] Pulled registry.k8s.io/kube-apiserver:v1.29.6
[config/images] Pulled registry.k8s.io/kube-controller-manager:v1.29.6
[config/images] Pulled registry.k8s.io/kube-scheduler:v1.29.6
[config/images] Pulled registry.k8s.io/kube-proxy:v1.29.6
[config/images] Pulled registry.k8s.io/coredns/coredns:v1.11.1
[config/images] Pulled registry.k8s.io/pause:3.9
[config/images] Pulled registry.k8s.io/etcd:3.5.10-0
```

$ sudo kubeadm init



```
ubuntu@ip-172-31-34-62:~$ sudo kubeadm init
I0708 16:39:15.752919    8620 version.go:256] remote version is much newer: v1.3
[init] Using Kubernetes version: v1.29.6
[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 inte
[preflight] You can also perform this action in beforehand using 'kubeadm config
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
```

You will receive the token

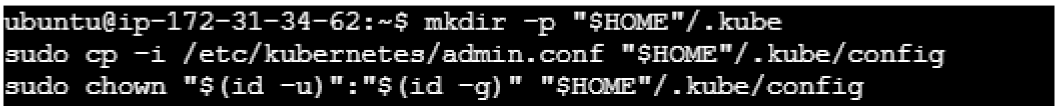


```
Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.34.62:6443 --token k33rcv.amhlma9j7m6rg5pb \
        --discovery-token-ca-cert-hash sha256:a632135c97e1ddc1c32672994f89a5b87ad0feec2c146be7ddd05c1feb9d
```

```
kubeadm join 172.31.34.62:6443 --token k33rcv.amhlma9j7m6rg5pb \
    --discovery-token-ca-cert-hash
sha256:a632135c97e1ddc1c32672994f89a5b87ad0feec2c146be7ddd05c1feb9d8f97
```

Run the following command:

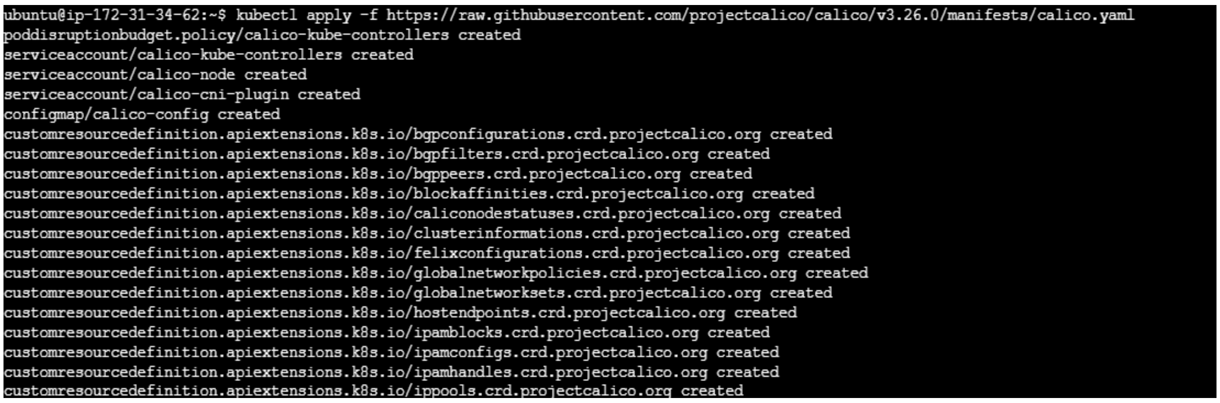
```
mkdir -p "$HOME"/.kube
sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config
sudo chown "$(id -u)":"$(id -g)" "$HOME"/.kube/config
```



```
ubuntu@ip-172-31-34-62:~$ mkdir -p "$HOME"/.kube
sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config
sudo chown "$(id -u)":"$(id -g)" "$HOME"/.kube/config
```

Network plugin:

kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml

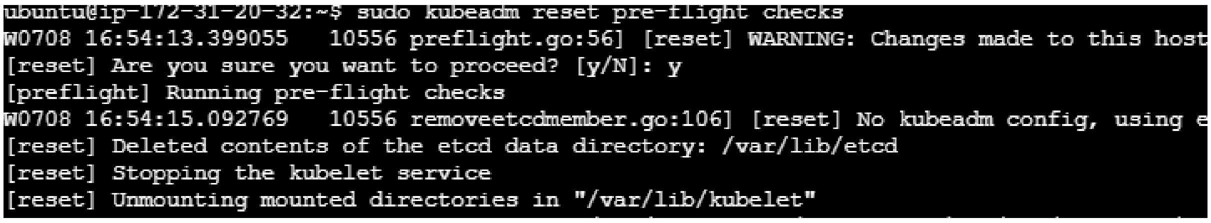


```
ubuntu@ip-172-31-34-62:~$ kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml
poddisruptionbudget.policy/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
serviceaccount/calico-cni-plugin created
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
```

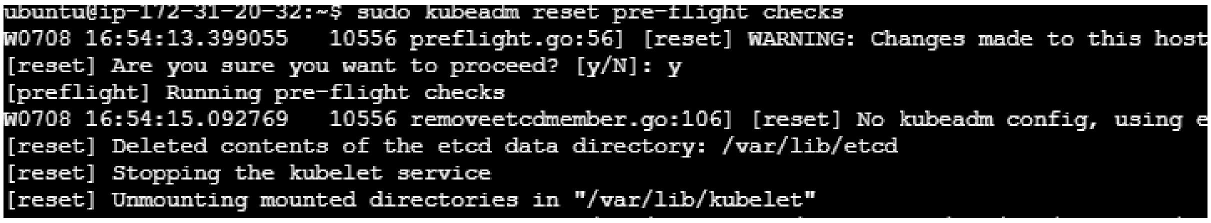
#Execute on all the worker nodes → K-S1 and K-S2

First perform the pre-flight checks

$ sudo kubeadm reset pre-flight checks



```
ubuntu@ip-172-31-20-32:~$ sudo kubeadm reset pre-flight checks
W0708 16:54:13.399055   10556 preflight.go:56] [reset] WARNING: Changes made to this host
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0708 16:54:15.092769   10556 removeetcdmember.go:106] [reset] No kubeadm config, using e
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
```

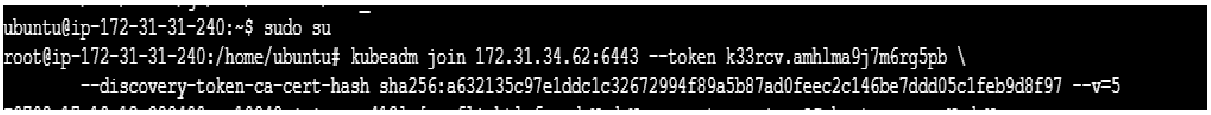


```
ubuntu@ip-172-31-20-32:~$ sudo kubeadm reset pre-flight checks
W0708 16:54:13.399055   10556 preflight.go:56] [reset] WARNING: Changes made to this host
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0708 16:54:15.092769   10556 removeetcdmember.go:106] [reset] No kubeadm config, using e
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
```

Paste the token along with - -v=5 for both slaves

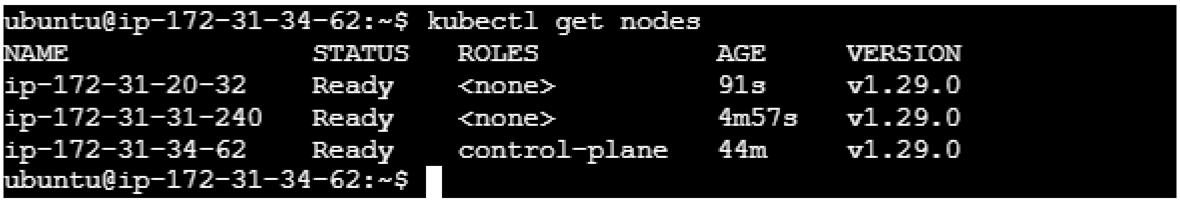
Inside the root directory

$ sudo su



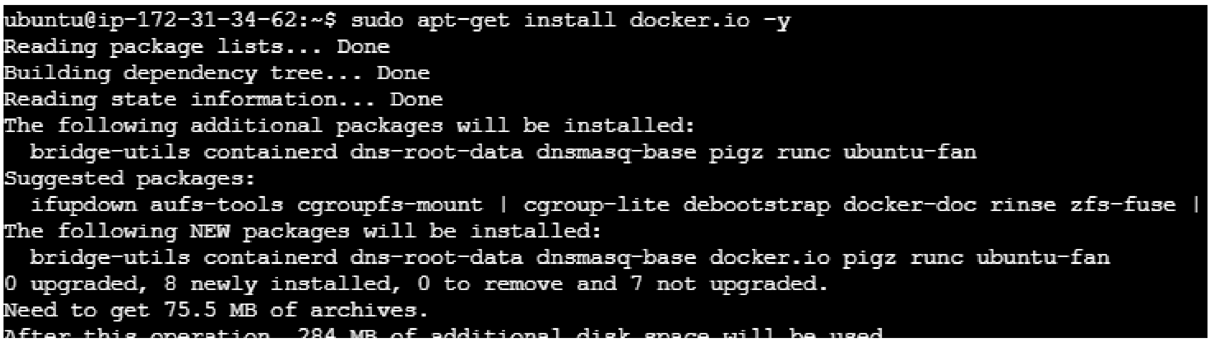
```
ubuntu@ip-172-31-31-240:~$ sudo su
root@ip-172-31-31-240:/home/ubuntu# kubeadm join 172.31.34.62:6443 --token k33rcv.amhlma9j7m6rg5pb \
        --discovery-token-ca-cert-hash sha256:a632135c97e1ddc1c32672994f89a5b87ad0feec2c146be7ddd05c1feb9d8f97 --v=5
```

Check on kubernetes master node

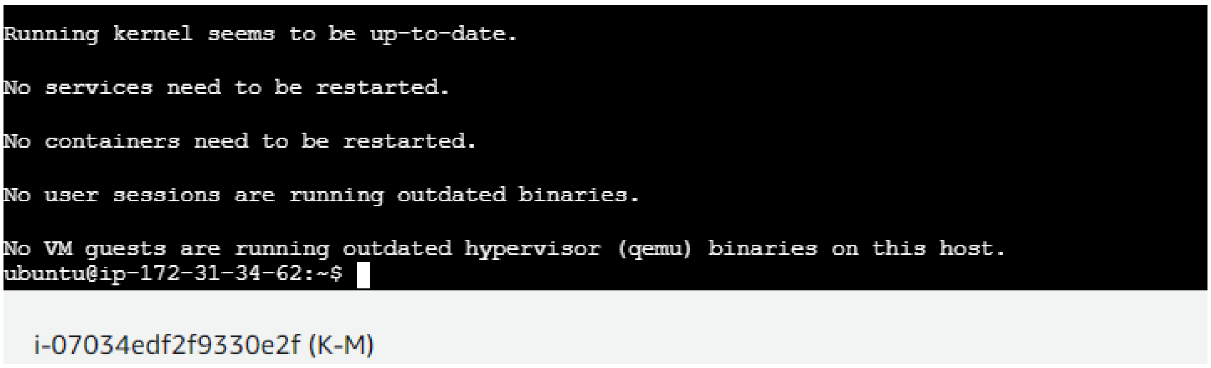


```
ubuntu@ip-172-31-34-62:~$ kubectl get nodes
NAME                STATUS   ROLES           AGE     VERSION
ip-172-31-20-32     Ready    <none>          91s     v1.29.0
ip-172-31-31-240    Ready    <none>          4m57s   v1.29.0
ip-172-31-34-62     Ready    control-plane   44m     v1.29.0
ubuntu@ip-172-31-34-62:~$
```

Step 10: After successfully join the cluster install docker on K-M



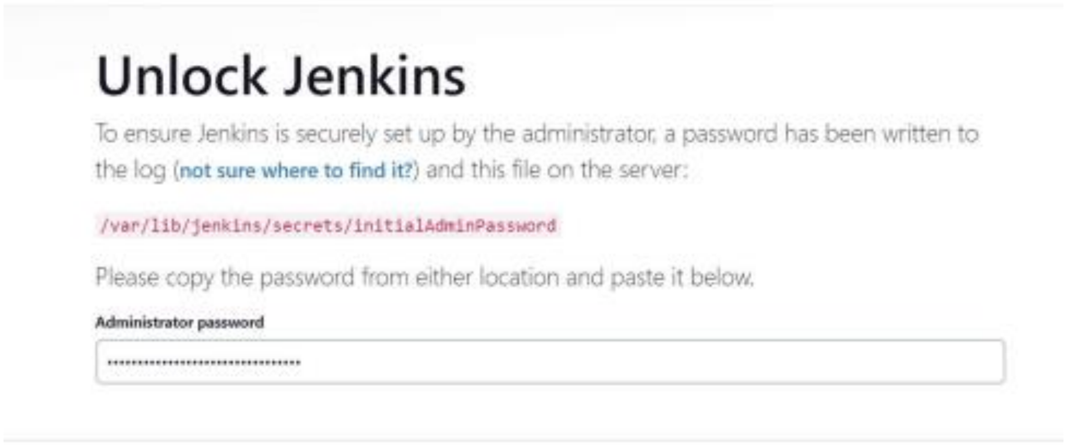
```
ubuntu@ip-172-31-34-62:~$ sudo apt-get install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse |
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 7 not upgraded.
Need to get 75.5 MB of archives.
After this operation, 284 MB of additional disk space will be used.
```



```
Running kernel seems to be up-to-date.

No services need to be restarted.

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.
ubuntu@ip-172-31-34-62:~$
```

i-07034edf2f9330e2f (K-M)

Step 11: Setup the Jenkins dashboard and create new Node called slave



Unlock Jenkins

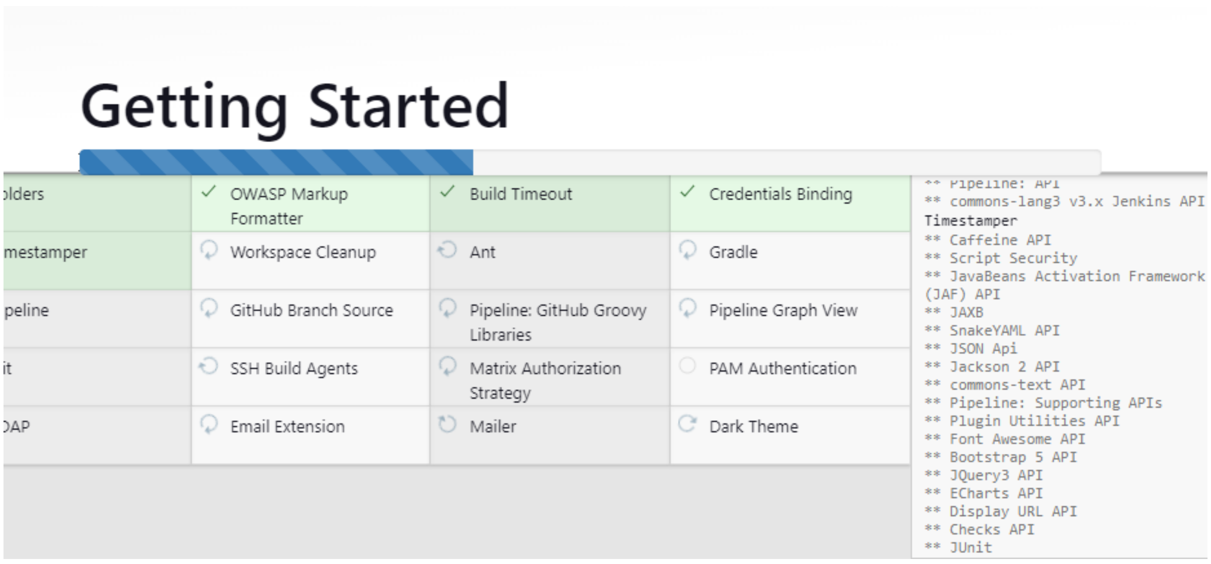
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

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

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

Administrator password

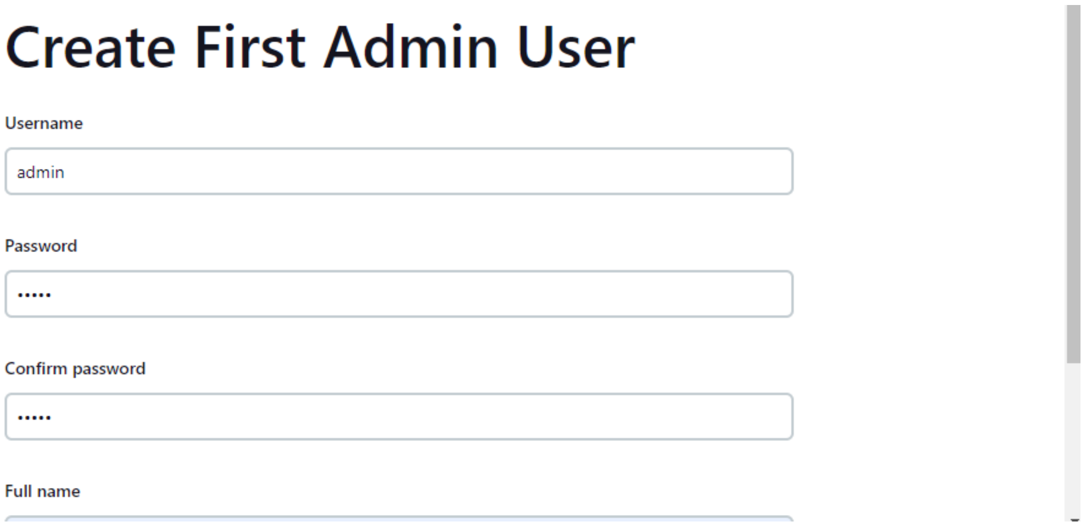
································



Getting Started

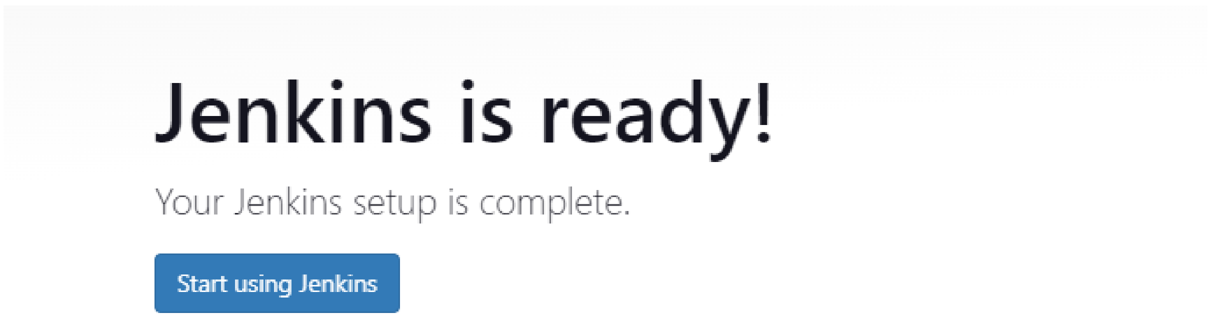
olders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
mestamper	Workspace Cleanup	Ant	Gradle
peline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline Graph View
it	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
AP	Email Extension	Mailer	Dark Theme

```
** Pipeline: API
** commons-lang3 v3.x Jenkins API
Timestamper
** Caffeine API
** Script Security
** JavaBeans Activation Framework
(JAF) API
** JAXB
** SnakeYAML API
** JSON Api
** Jackson 2 API
** commons-text API
** Pipeline: Supporting APIs
** Plugin Utilities API
** Font Awesome API
** Bootstrap 5 API
** JQuery3 API
** ECharts API
** Display URL API
** Checks API
** JUnit
```



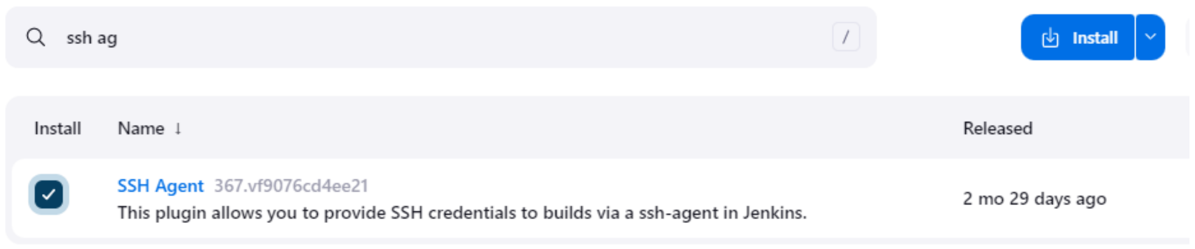
Create First Admin User

Username	admin
Password	•••••
Confirm password	•••••
Full name	



Jenkins is ready!

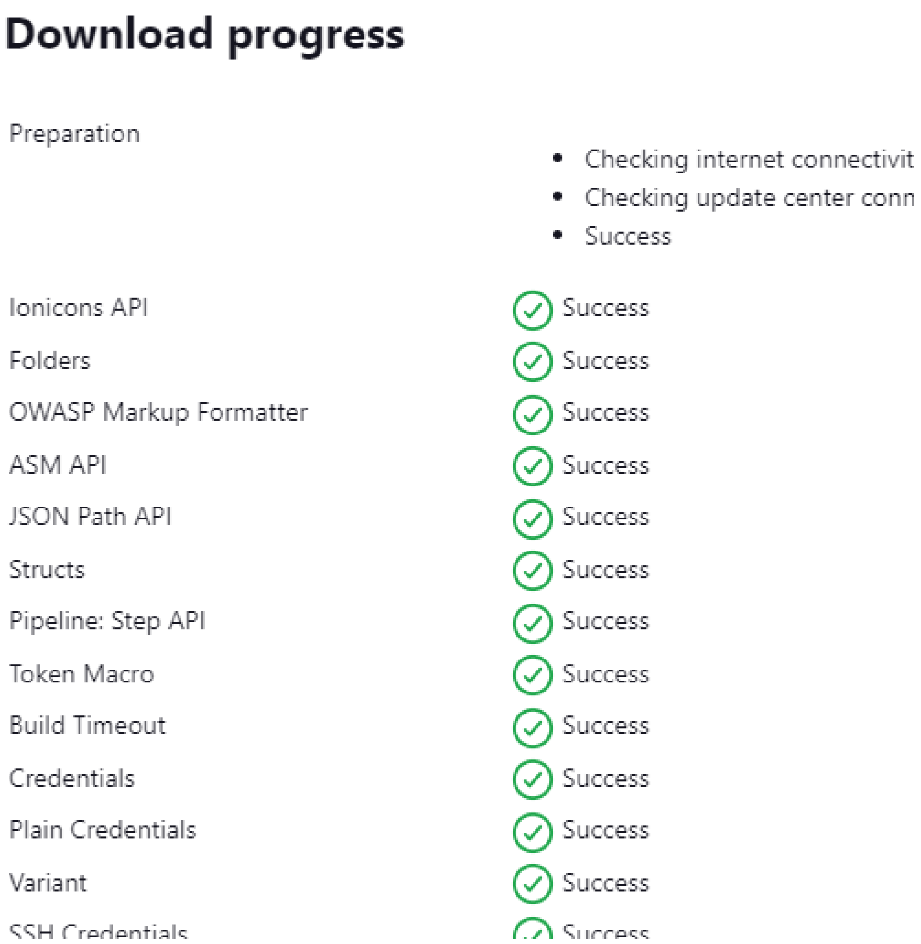
Your Jenkins setup is complete.



ssh ag

Install

Install	Name ↓	Released
☑	SSH Agent 367.vf9076cd4ee21 This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.	2 mo 29 days ago

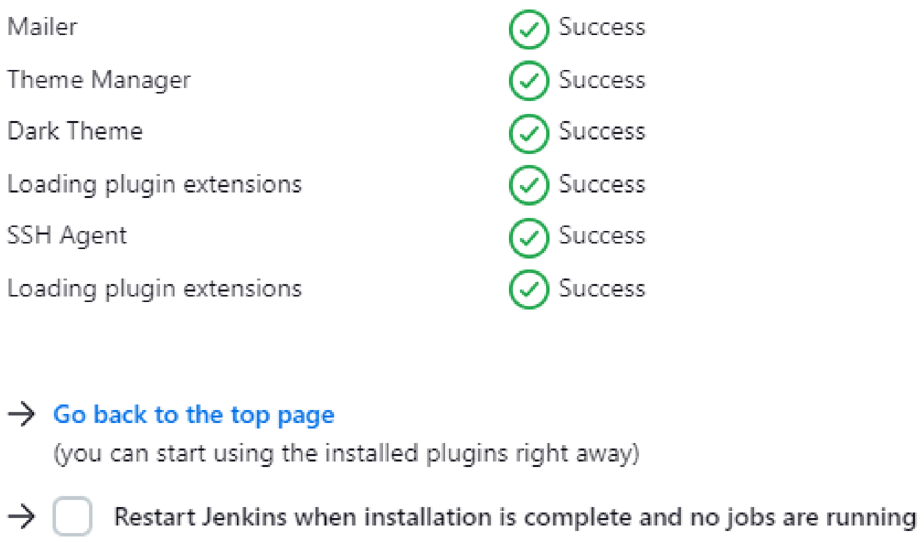


Download progress

Preparation

- Checking internet connectivit
- Checking update center conn
- Success

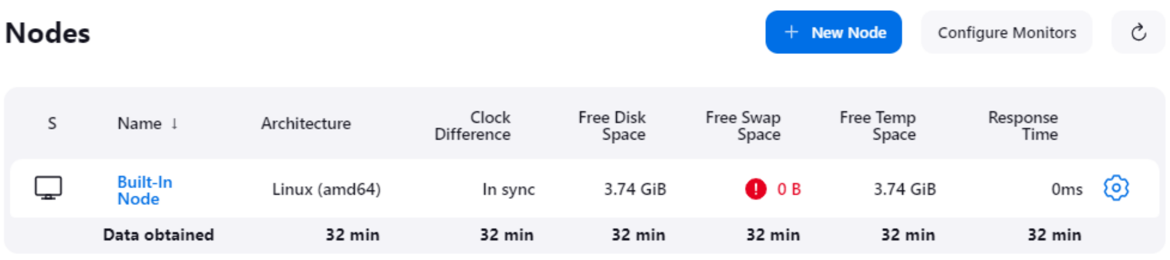
Ionicons API	✓ Success
Folders	✓ Success
OWASP Markup Formatter	✓ Success
ASM API	✓ Success
JSON Path API	✓ Success
Structs	✓ Success
Pipeline: Step API	✓ Success
Token Macro	✓ Success
Build Timeout	✓ Success
Credentials	✓ Success
Plain Credentials	✓ Success
Variant	✓ Success
SSH Credentials	✓ Success



Mailer	✓ Success
Theme Manager	✓ Success
Dark Theme	✓ Success
Loading plugin extensions	✓ Success
SSH Agent	✓ Success
Loading plugin extensions	✓ Success

→ Go back to the top page
(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

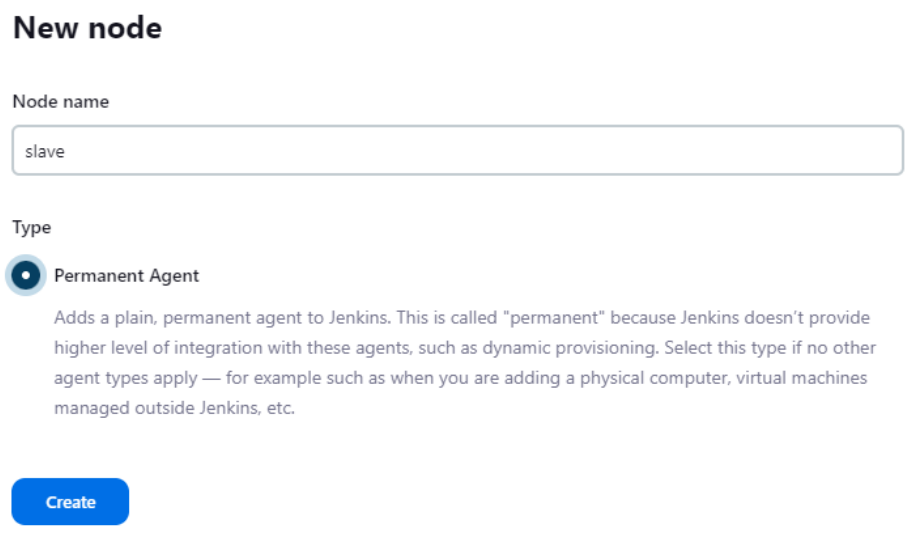


Nodes

+ New Node | Configure Monitors

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.74 GiB	0 B	3.74 GiB	0ms
	Data obtained	32 min	32 min	32 min	32 min	32 min	32 min

Now create a new node:



New node

Node name

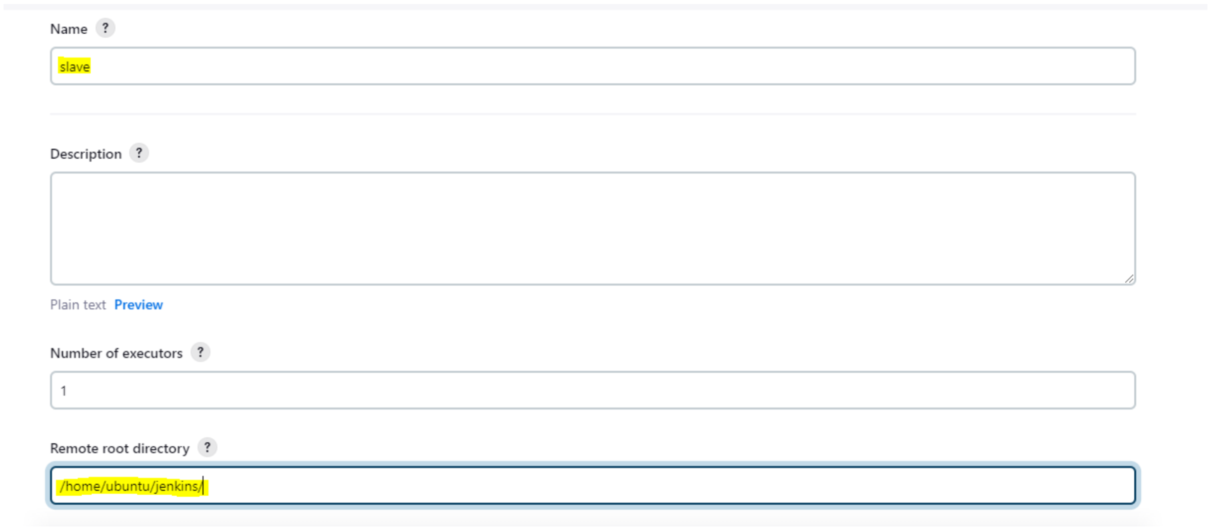
slave

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.

Create



Name ?

slave

Description ?

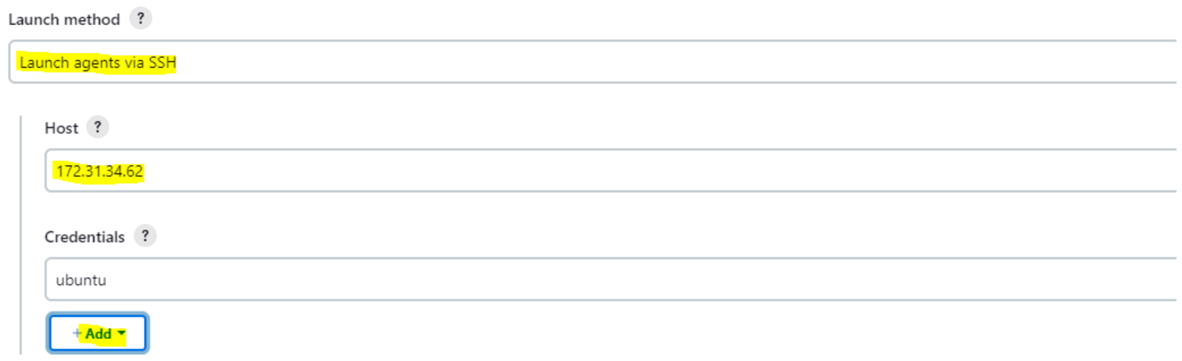
Plain text Preview

Number of executors ?

1

Remote root directory ?

/home/ubuntu/jenkins/


Launch method

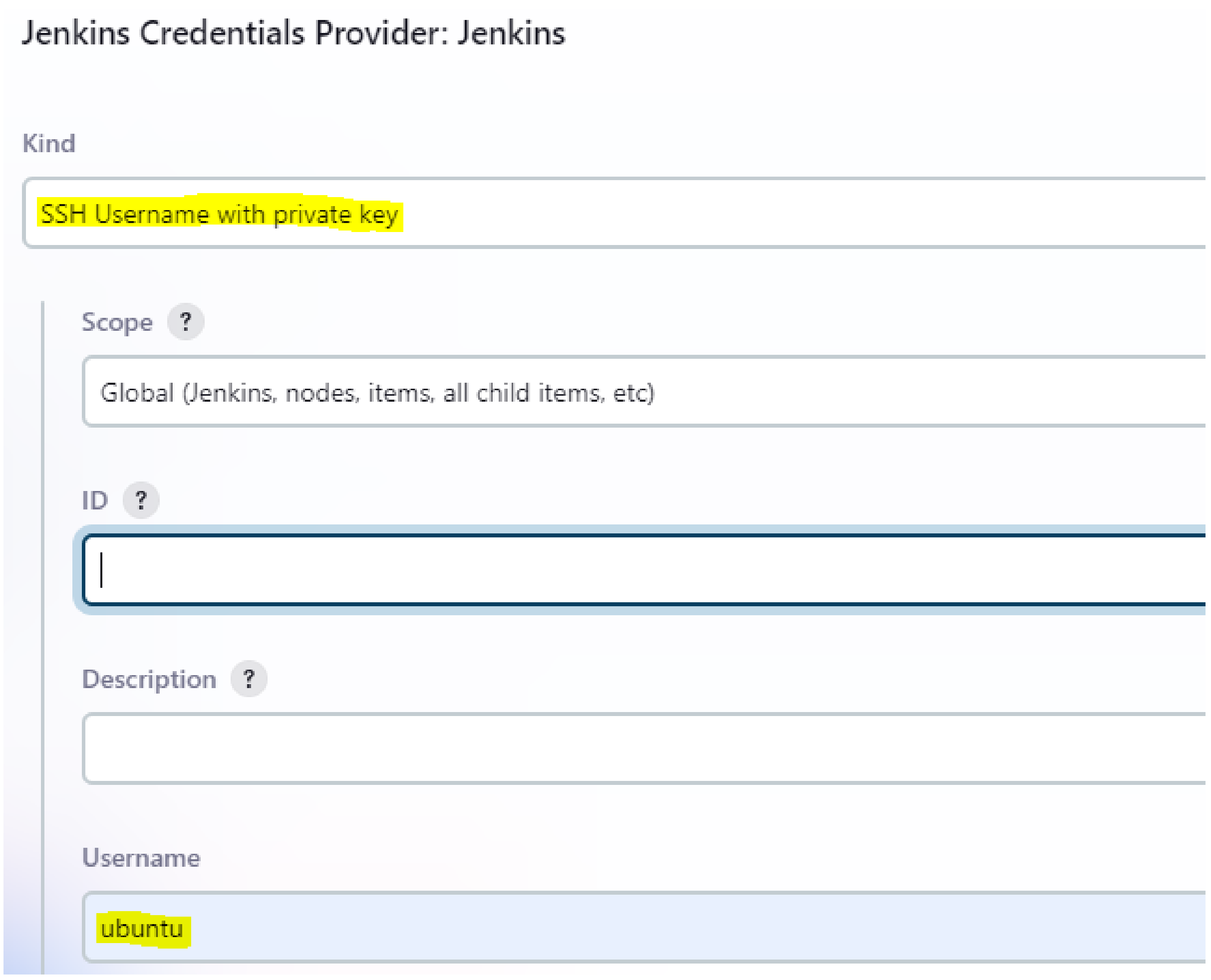
Launch agents via SSH

Host

172.31.34.62

Credentials

ubuntu


Jenkins Credentials Provider: Jenkins

Kind

SSH Username with private key

Scope

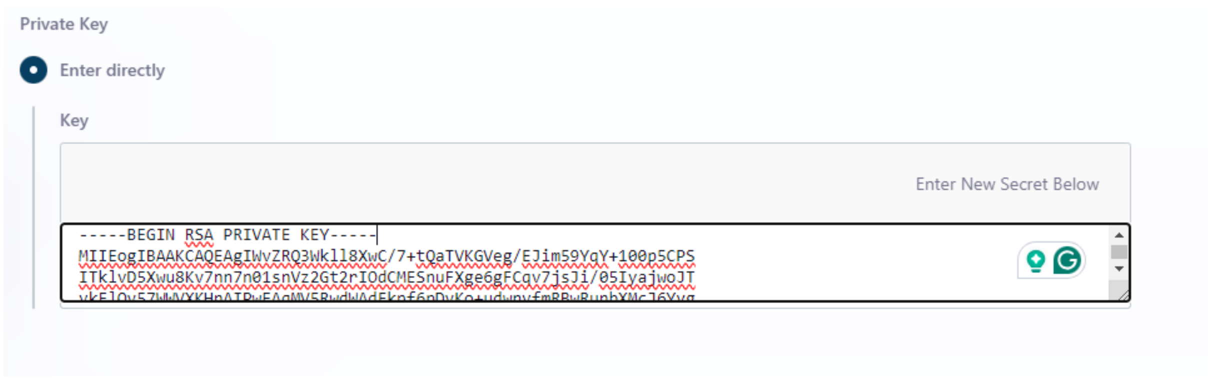
Global (Jenkins, nodes, items, all child items, etc)

ID

Description

Username

ubuntu



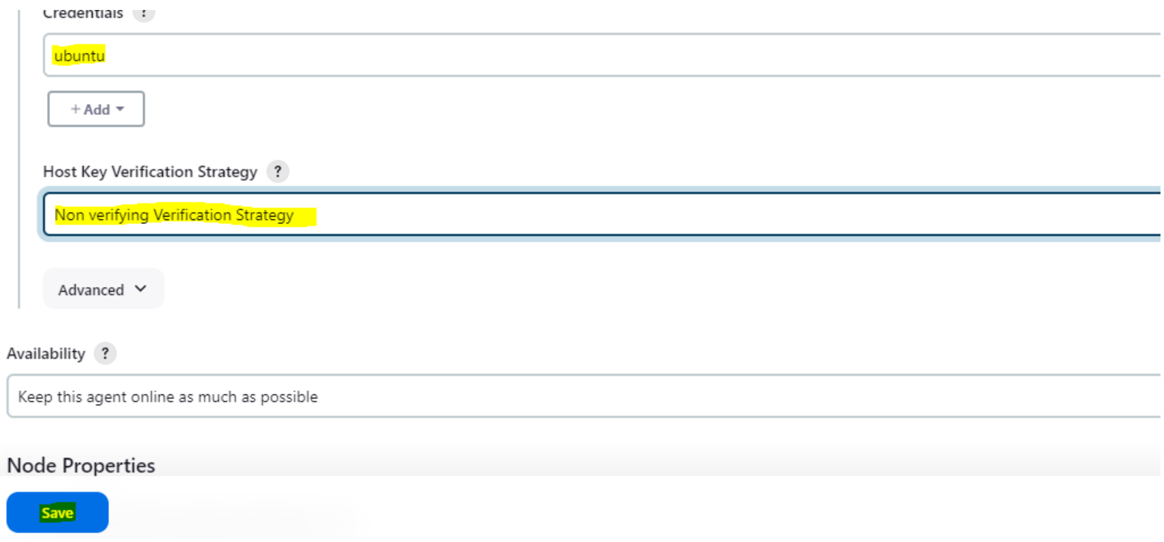
Private Key

● Enter directly

Key

Enter New Secret Below

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAgIWvZRQ3Wkll8XwC/7+tQaTVKGVeg/EJim59YqY+100p5CPS
ITklvD5Xwu8Kv7nn7n01snVz2Gt2rIOdCMESnuFXge6gFCqv7jsJi/05IyajwoJT
```



Credentials

ubuntu

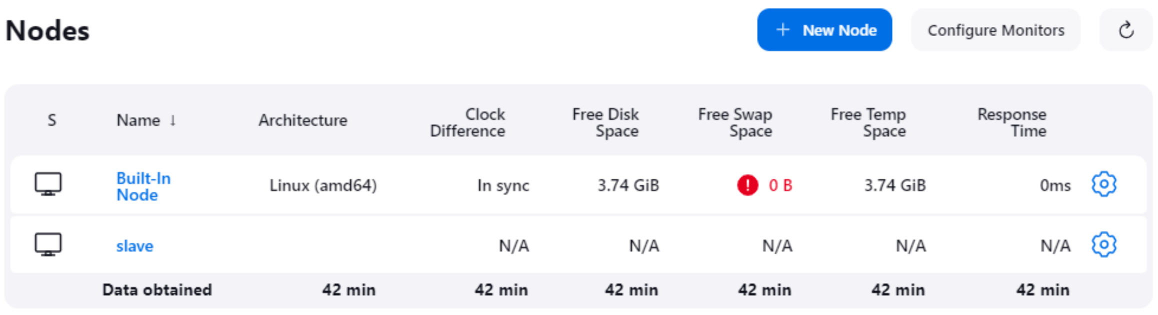
Host Key Verification Strategy

Non verifying Verification Strategy

Availability

Keep this agent online as much as possible

Node Properties



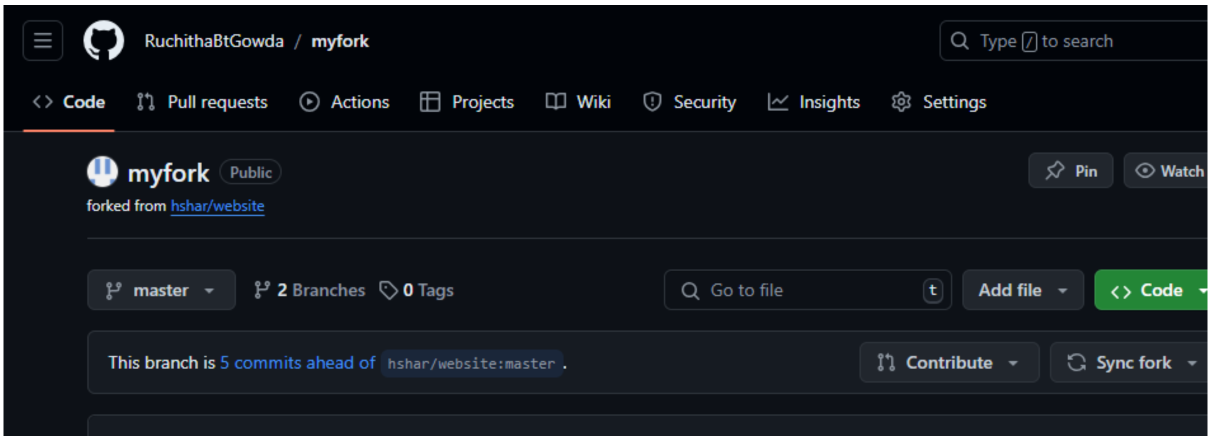
Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.74 GiB	0 B	3.74 GiB	0ms
	slave		N/A	N/A	N/A	N/A	N/A
	Data obtained	42 min	42 min	42 min	42 min	42 min	42 min

Step 11: Fork the given repository and create new 2 yaml file for deployment and service

https://github.com/hshar/website

https://github.com/RuchithaBtGowda/myfork/new/master



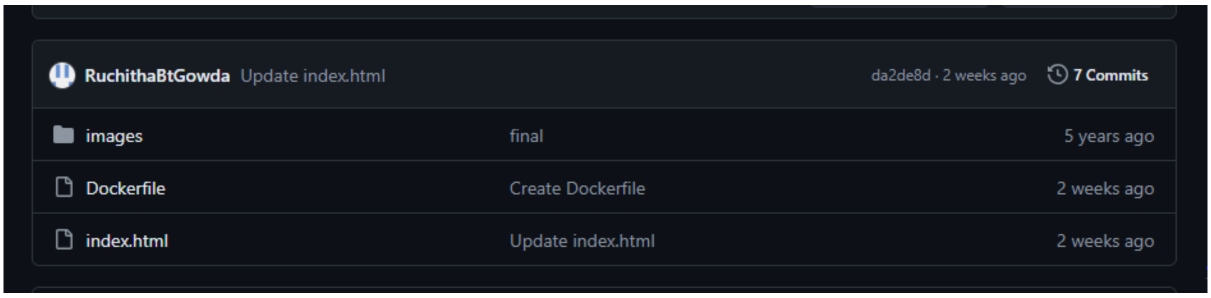
myfork Public

forked from hshar/website

2 Branches 0 Tags

This branch is 5 commits ahead of `hshar/website:master`.

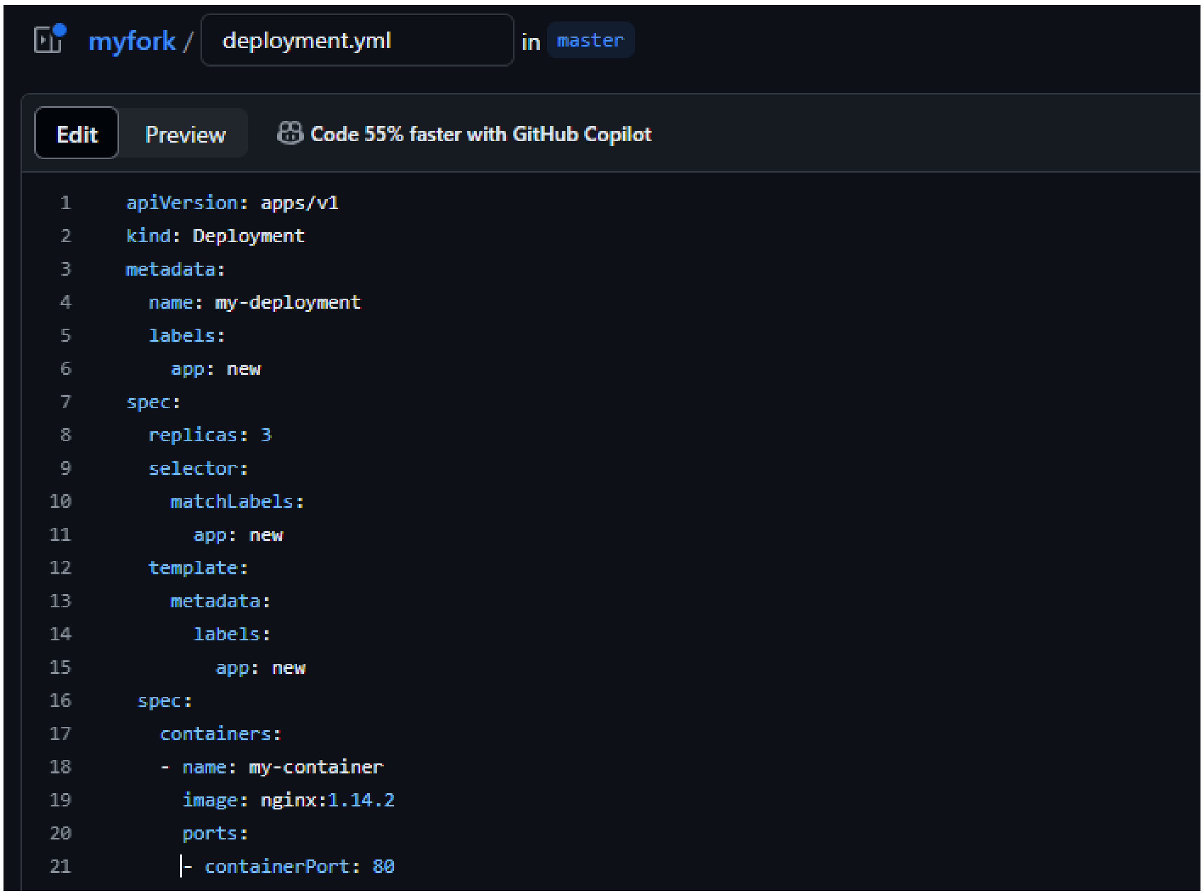
Available file on the repo



RuchithaBtGowda Update index.html da2de8d · 2 weeks ago 7 Commits

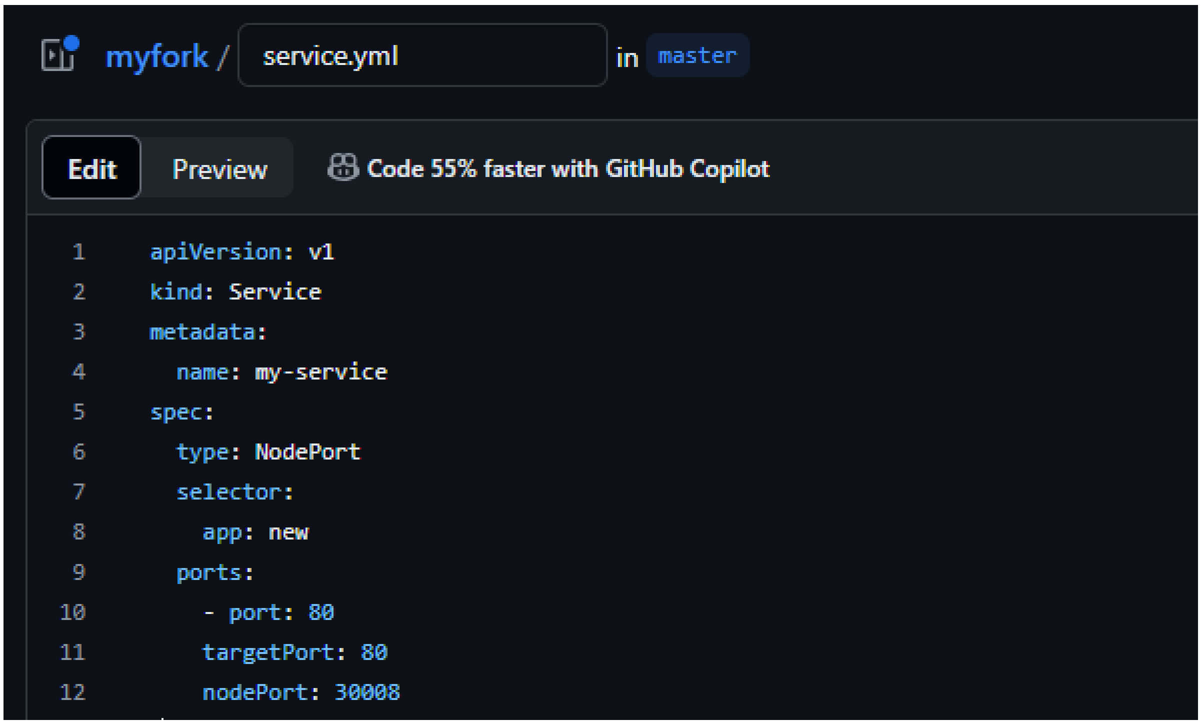
images	final	5 years ago
Dockerfile	Create Dockerfile	2 weeks ago
index.html	Update index.html	2 weeks ago

Create deployment.yml file


myfork / deployment.yml in master
Edit
Preview
Code 55% faster with GitHub Copilot

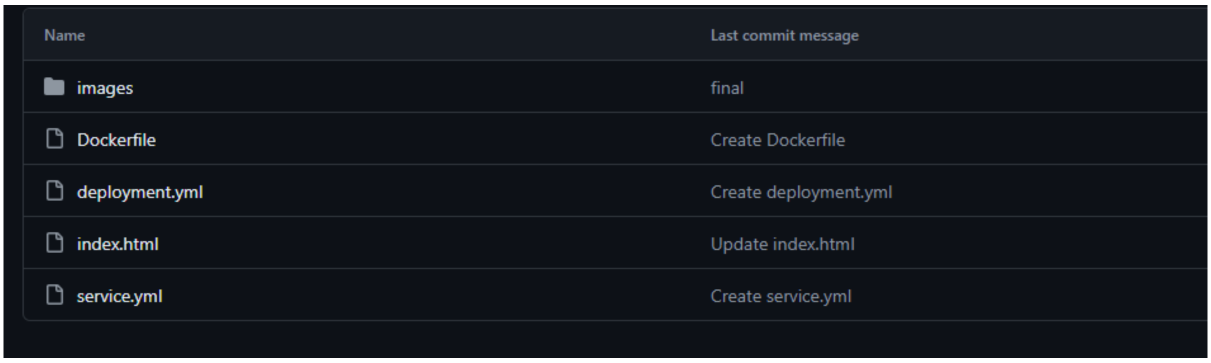
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  labels:
    app: new
spec:
  replicas: 3
  selector:
    matchLabels:
      app: new
  template:
    metadata:
      labels:
        app: new
   spec:
     containers:
     - name: my-container
       image: nginx:1.14.2
       ports:
       - containerPort: 80
```

Service.yml file


myfork / service.yml in master
Edit
Preview
Code 55% faster with GitHub Copilot

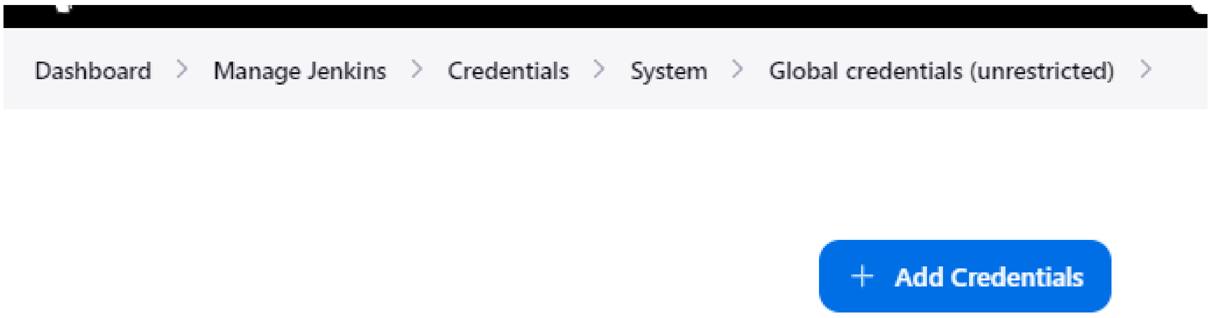
```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: new
  ports:
    - port: 80
    targetPort: 80
    nodePort: 30008
```

Successfully created yaml files for deploying

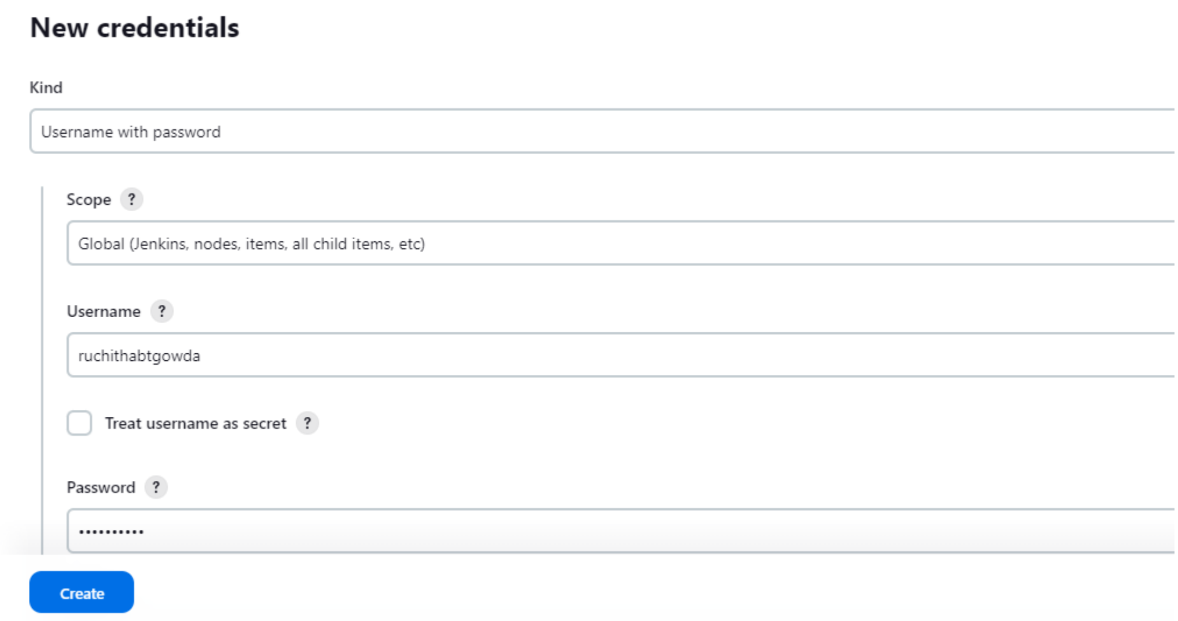


Name	Last commit message
images	final
Dockerfile	Create Dockerfile
deployment.yml	Create deployment.yml
index.html	Update index.html
service.yml	Create service.yml

Step 13: Setting up the docker hub credentials before creating the image


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

Give the user name and password of docker hub



New credentials

Kind: Username with password

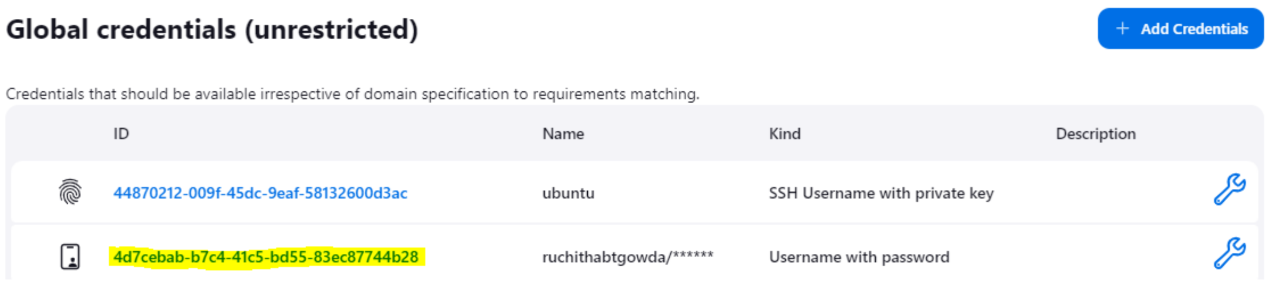
Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: ruchithabtgowda

[ ] Treat username as secret

Password: ••••••••••

Save the credential

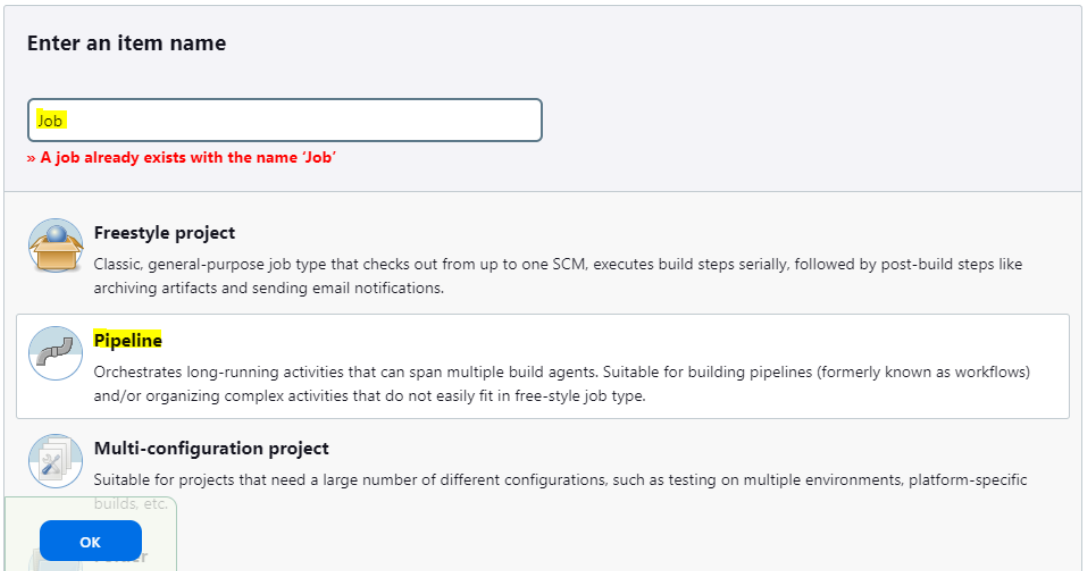

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
44870212-009f-45dc-9eaf-58132600d3ac	ubuntu	SSH Username with private key	
4d7cebab-b7c4-41c5-bd55-83ec87744b28	ruchithabtgowda/******	Username with password	

Step 14: Creating a Job

Here we will create a pipeline job


Enter an item name

Job

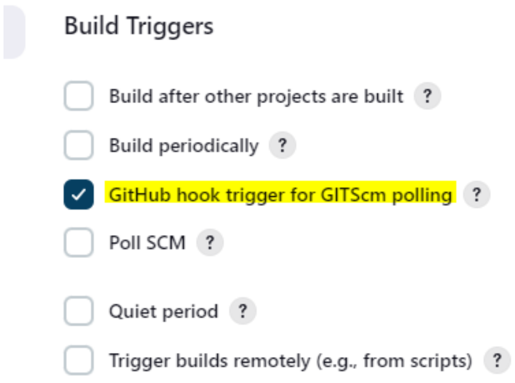
» A job already exists with the name 'Job'

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

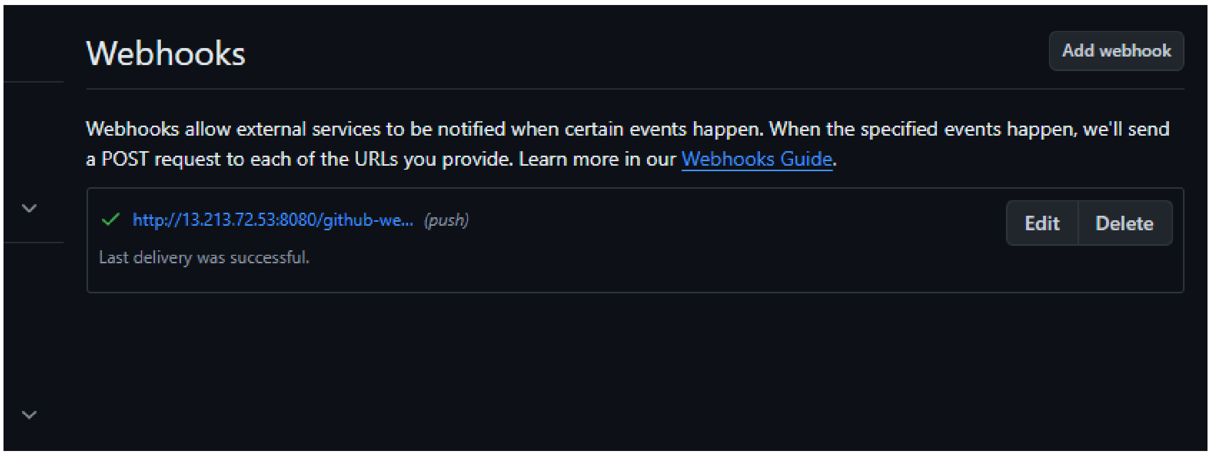
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Set webhook trigger


Build Triggers

- [ ] Build after other projects are built
- [ ] Build periodically
- [x] GitHub hook trigger for GITScm polling
- [ ] Poll SCM
- [ ] Quiet period
- [ ] Trigger builds remotely (e.g., from scripts)

Go to git hub and add the webhook

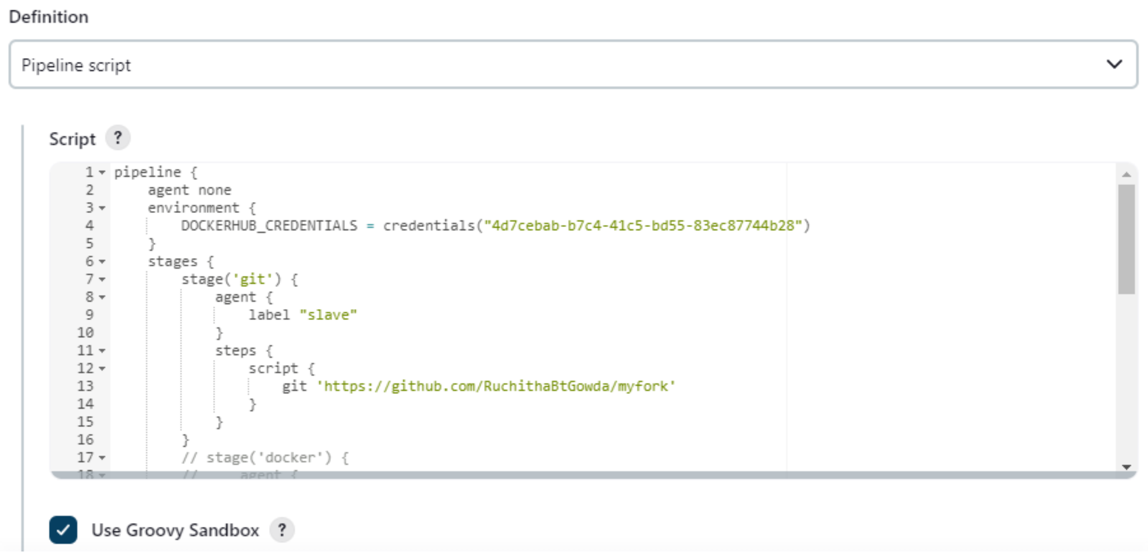


Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://13.213.72.53:8080/github-we... (push)

Last delivery was successful.

Add the pipeline code



Definition: Pipeline script

Script

```
pipeline {
    agent none
    environment {
        DOCKERHUB_CREDENTIALS = credentials("4d7cebab-b7c4-41c5-bd55-83ec87744b28")
    }
    stages {
        stage('git') {
            agent {
                label "slave"
            }
            steps {
                script {
                    git 'https://github.com/RuchithaBtGowda/myfork'
                }
            }
        }
        // stage('docker') {
```

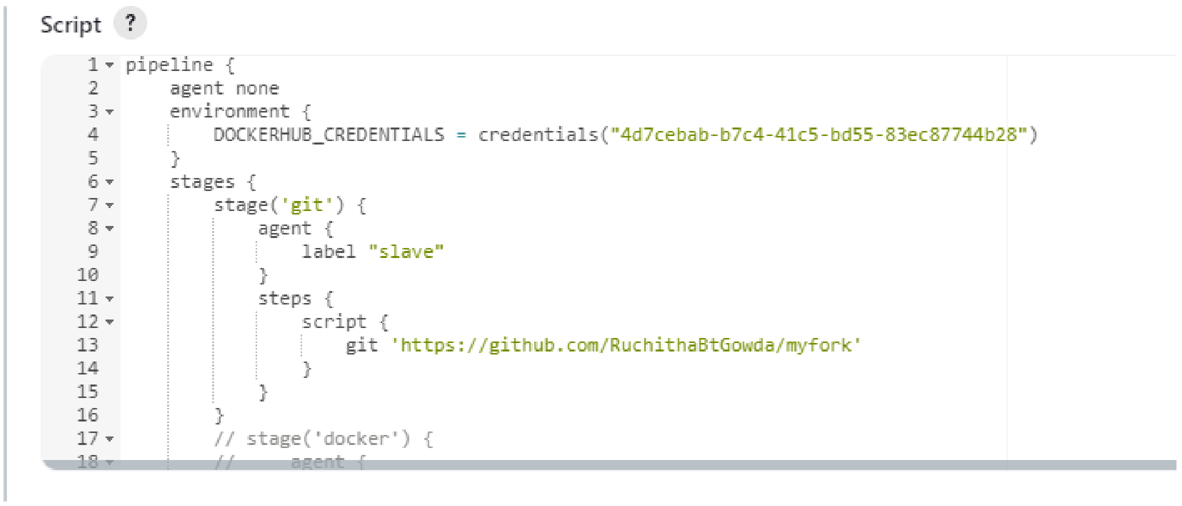
✓ Use Groovy Sandbox

There are three stage in the code:

1st stage- source code management, takes file from git hub adding to slave
2nd stage – build docker image, as for deploying we need an image I e through a Docker file.
3rd stage – kubernetes deployment

```
pipeline {
   agent none
   environment {
      DOCKERHUB_CREDENTIALS = credentials("cb26ac19-f954-40ca-a12e-
d790594bcca7")
   }
   stages {
      stage('git') {
         agent {
            label "K8-Master"
         }
         steps {
            script {
               git'https://github.com/RuchithaBtGowda /website.git'
            }
         }
      }
      stage('docker') {
         agent {
            label "K8-Master"
         }
         steps {
            script {
               sh 'sudo docker build /home/ubuntu/jenkins/workspace/test-pipeline/ -t
intellipaatsai/proj2'
               sh 'sudo docker login -u ${DOCKERHUB_CREDENTIALS_USR} -p
${DOCKERHUB_CREDENTIALS_PSW}'
               sh 'sudo docker push intellipaatsai/proj2'
            }
         }
      }
      stage('kubernetes') {
         agent {
            label "K8-Master"
         }
         steps {
            script {
               sh 'kubectl delete deploy nginx-deployment'
               sh 'kubectl apply -f deployment.yaml'
               sh 'kubectl delete service my-service'
               sh 'kubectl apply -f service.yaml'
            }
         }
      }
   }
}
```

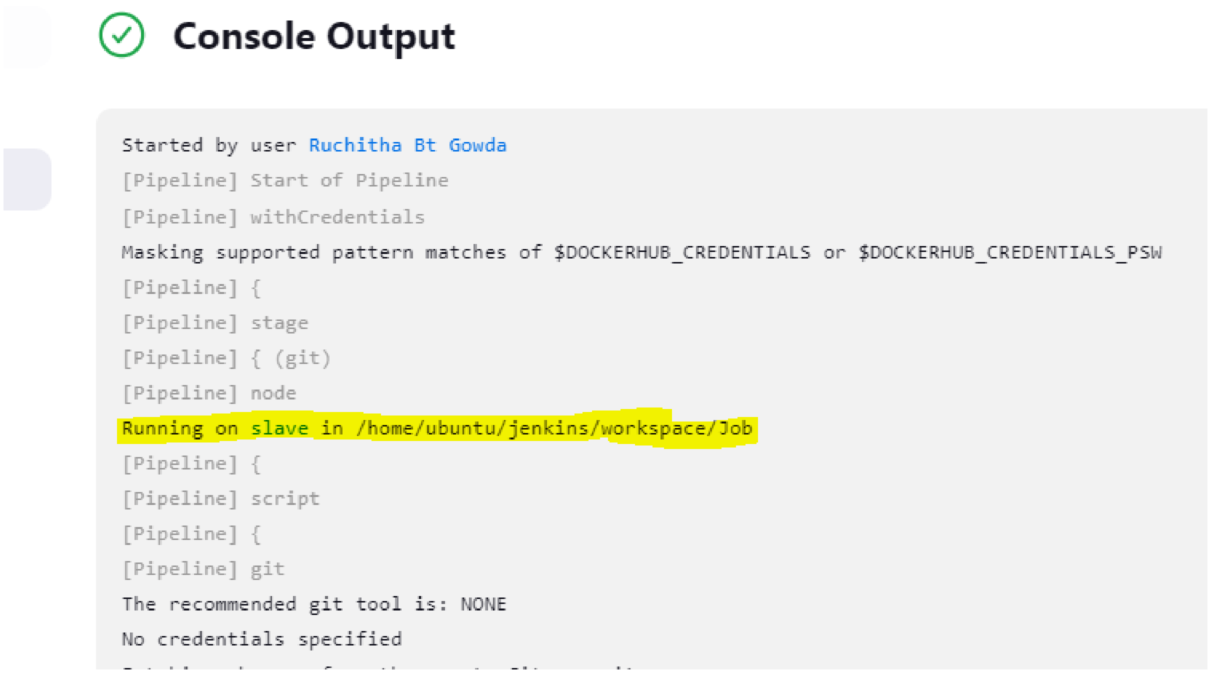
Build the script up to 1st stage, stash the remaining



Script ?

```
pipeline {
    agent none
    environment {
        DOCKERHUB_CREDENTIALS = credentials("4d7cebab-b7c4-41c5-bd55-83ec87744b28")
    }
    stages {
        stage('git') {
            agent {
                label "slave"
            }
            steps {
                script {
                    git 'https://github.com/RuchithaBtGowda/myfork'
                }
            }
        }
        // stage('docker') {
        //     agent {
```

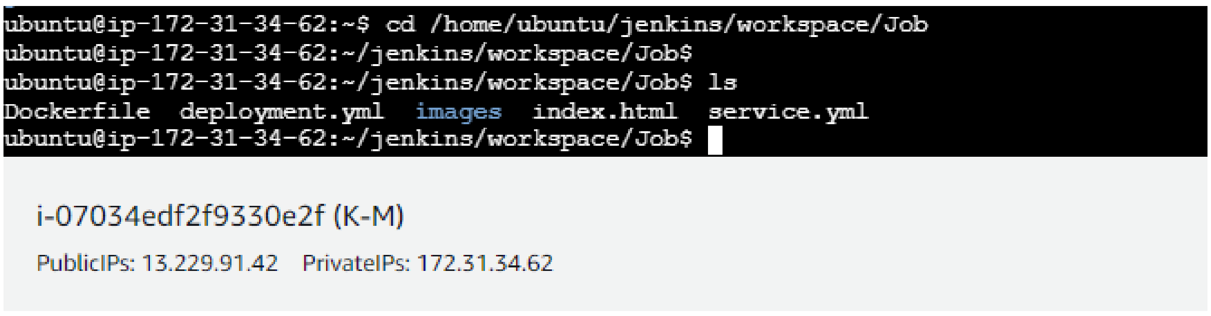
Console output after building



Console Output

```
Started by user Ruchitha Bt Gowda
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (git)
[Pipeline] node
Running on slave in /home/ubuntu/jenkins/workspace/Job
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
```

Here slave is Kubernetes master, check the git hub files in it

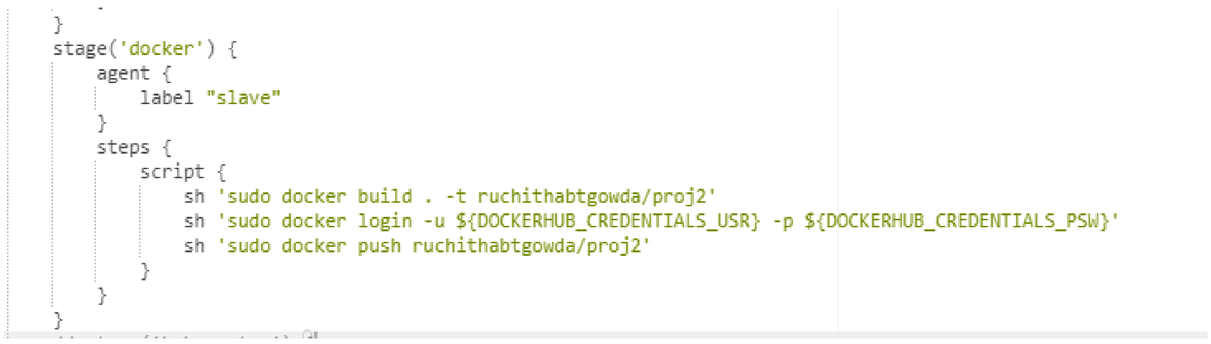


```
ubuntu@ip-172-31-34-62:~$ cd /home/ubuntu/jenkins/workspace/Job
ubuntu@ip-172-31-34-62:~/jenkins/workspace/Job$
ubuntu@ip-172-31-34-62:~/jenkins/workspace/Job$ ls
Dockerfile  deployment.yml  images  index.html  service.yml
ubuntu@ip-172-31-34-62:~/jenkins/workspace/Job$
```

i-07034edf2f9330e2f (K-M)

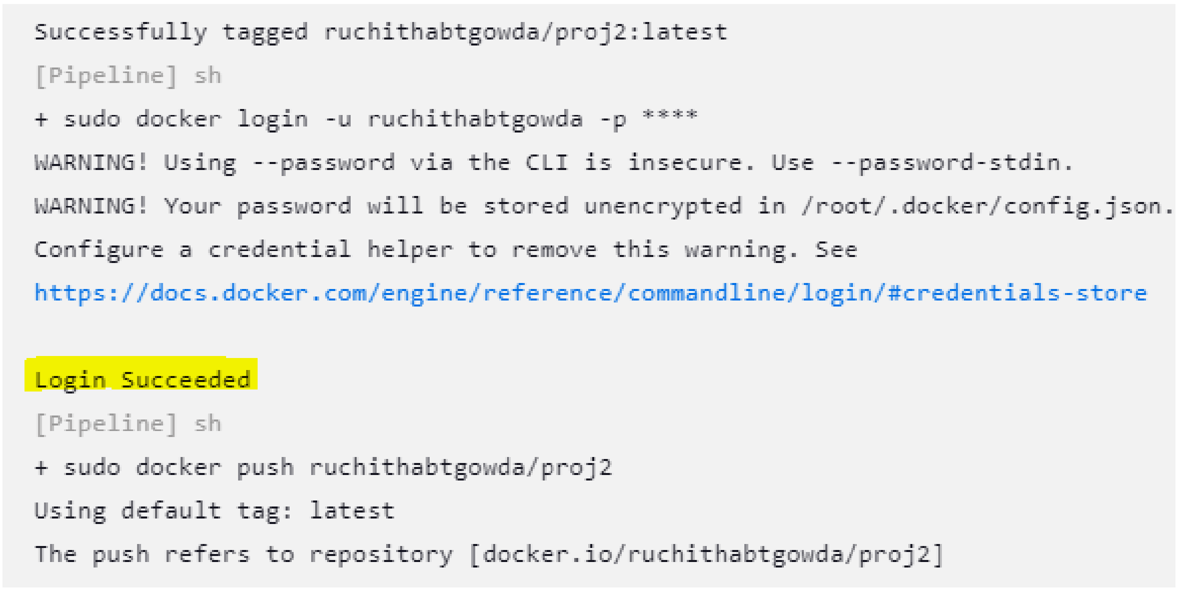
PublicIPs: 13.229.91.42   PrivateIPs: 172.31.34.62

Now uncomment the stage 2



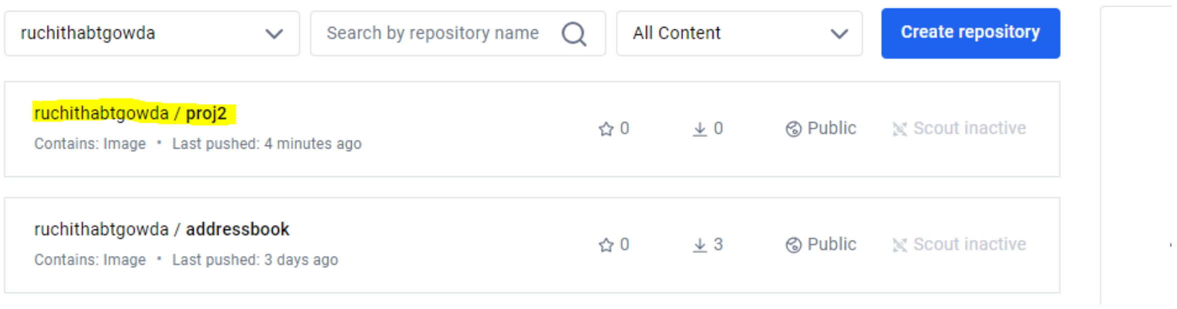
```
    }
    stage('docker') {
        agent {
            label "slave"
        }
        steps {
            script {
                sh 'sudo docker build . -t ruchithabtgowda/proj2'
                sh 'sudo docker login -u ${DOCKERHUB_CREDENTIALS_USR} -p ${DOCKERHUB_CREDENTIALS_PSW}'
                sh 'sudo docker push ruchithabtgowda/proj2'
            }
        }
    }
```

Build the job



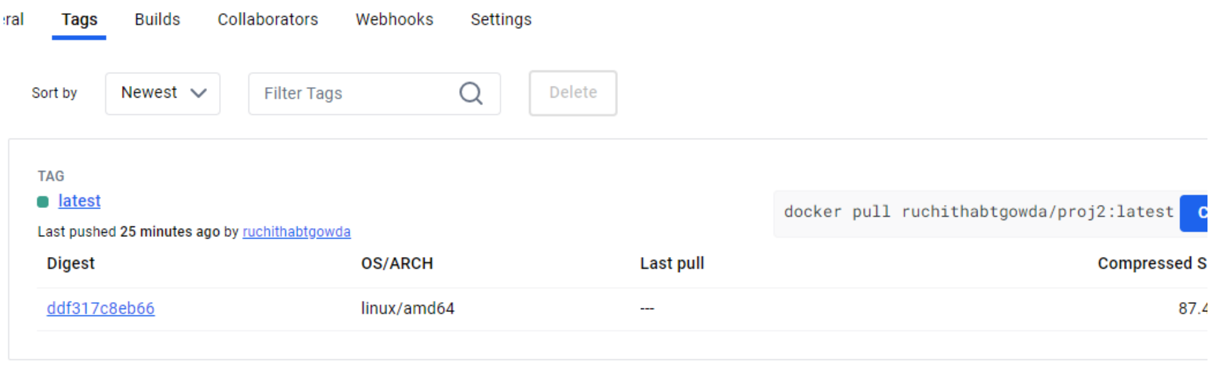
```
Successfully tagged ruchithabtgowda/proj2:latest
[Pipeline] sh
+ sudo docker login -u ruchithabtgowda -p ****
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] sh
+ sudo docker push ruchithabtgowda/proj2
Using default tag: latest
The push refers to repository [docker.io/ruchithabtgowda/proj2]
```

Login to the docker hub and check


ruchithabtgowda | Search by repository name | All Content | Create repository

ruchithabtgowda / proj2
Contains: Image • Last pushed: 4 minutes ago   ☆ 0   ⤓ 0   Public   Scout inactive

ruchithabtgowda / addressbook
Contains: Image • Last pushed: 3 days ago   ☆ 0   ⤓ 3   Public   Scout inactive


ral | Tags | Builds | Collaborators | Webhooks | Settings

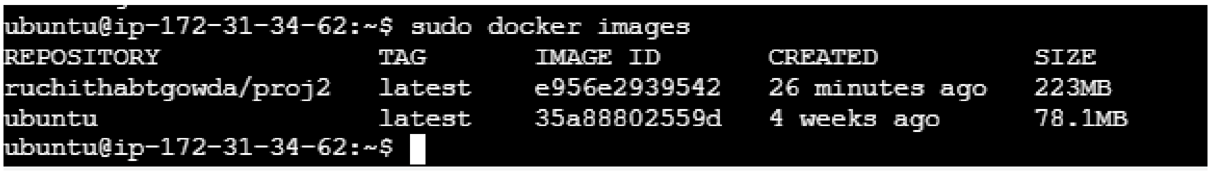
Sort by Newest | Filter Tags | Delete

TAG
● latest
Last pushed 25 minutes ago by ruchithabtgowda

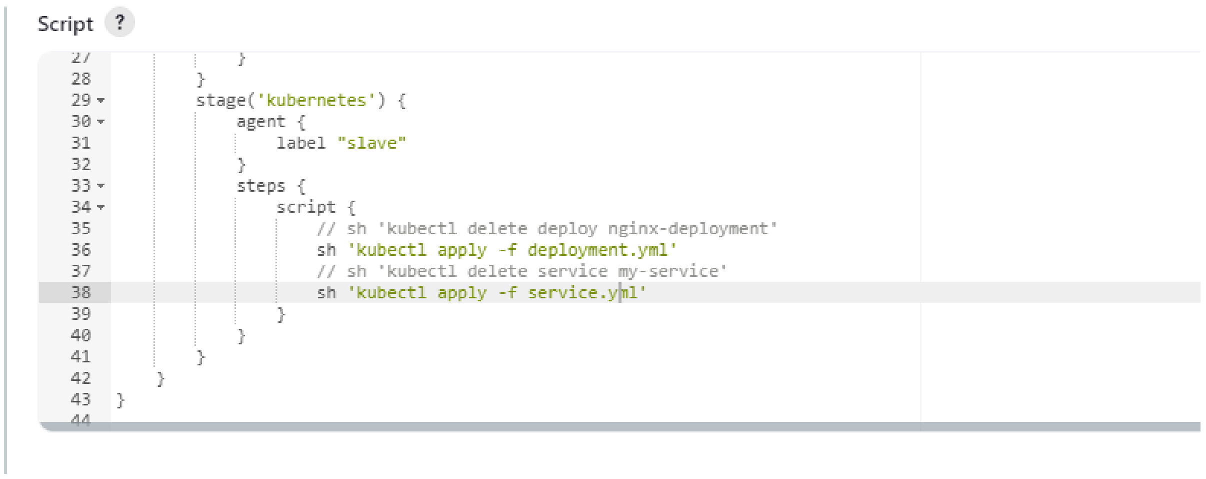
```
docker pull ruchithabtgowda/proj2:latest
```

Digest	OS/ARCH	Last pull	Compressed S
ddf317c8eb66	linux/amd64	---	87.4

Check the image on slave machine

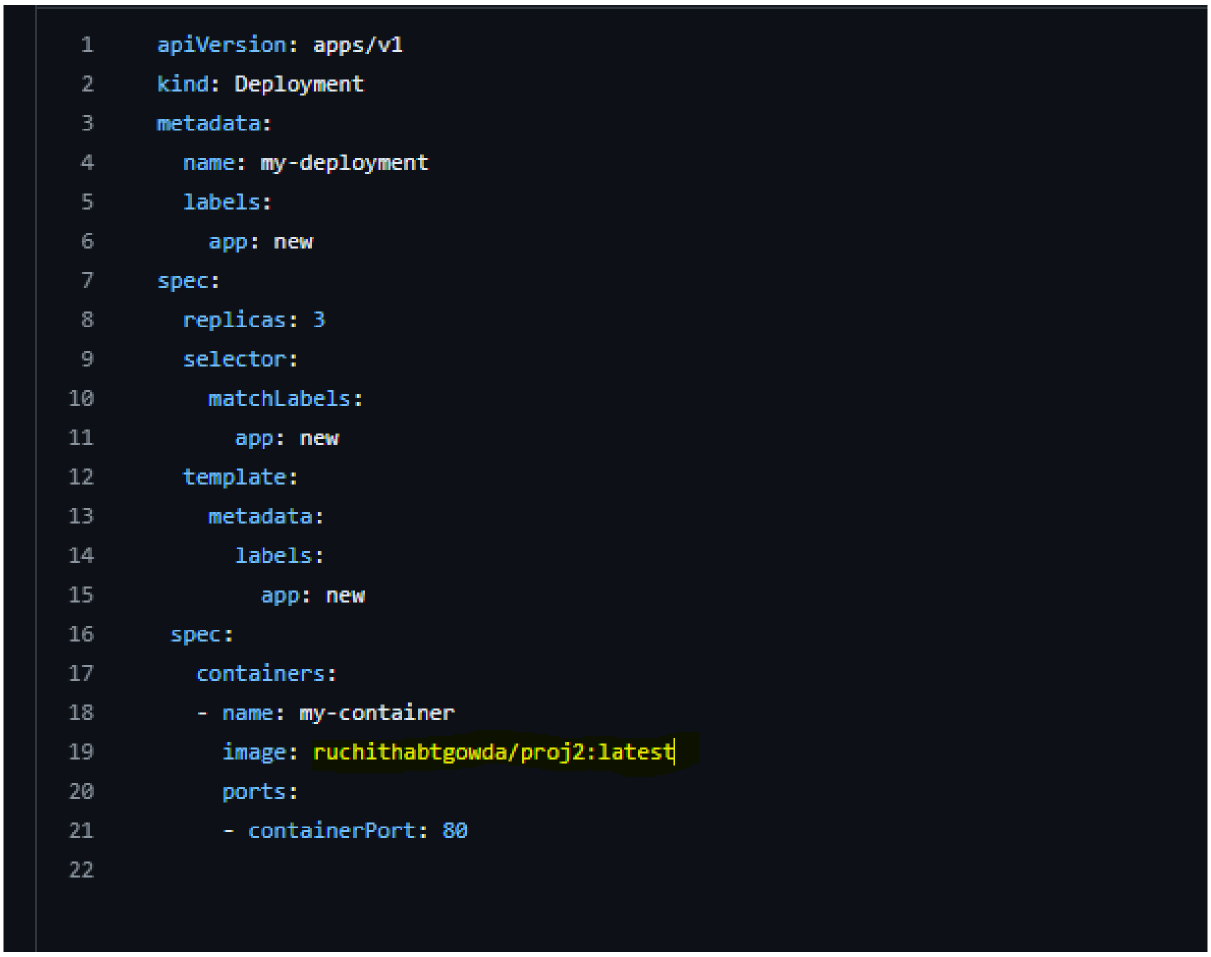

```
ubuntu@ip-172-31-34-62:~$ sudo docker images
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
ruchithabtgowda/proj2  latest    e956e2939542   26 minutes ago   223MB
ubuntu                 latest    35a88802559d   4 weeks ago      78.1MB
ubuntu@ip-172-31-34-62:~$
```

Now unstash the kubernetes stage


Script ?

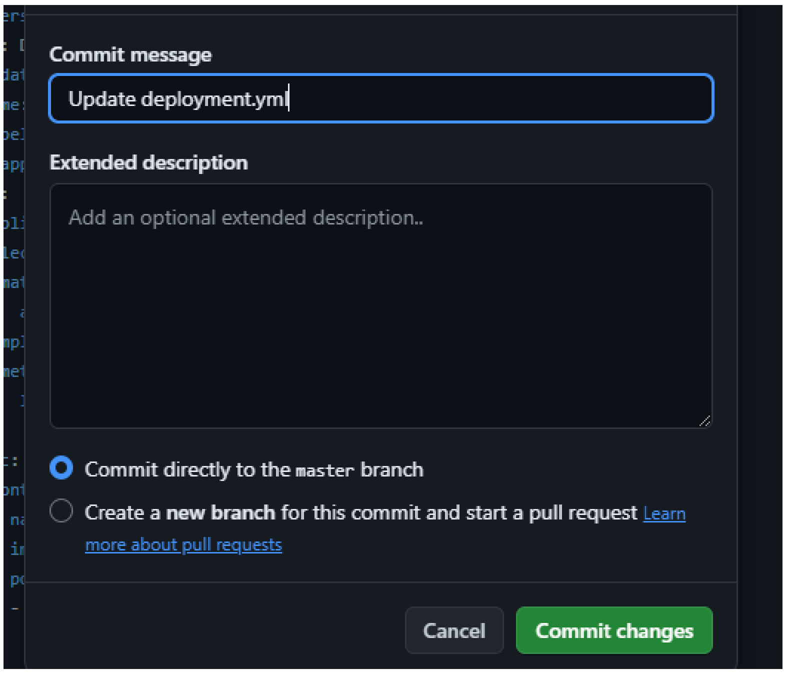
```
        }
    }
    stage('kubernetes') {
        agent {
            label "slave"
        }
        steps {
            script {
                // sh 'kubectl delete deploy nginx-deployment'
                sh 'kubectl apply -f deployment.yml'
                // sh 'kubectl delete service my-service'
                sh 'kubectl apply -f service.yml'
            }
        }
    }
  }
}
```

Add the docker image in git hub deployment file



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  labels:
    app: new
spec:
  replicas: 3
  selector:
    matchLabels:
      app: new
  template:
    metadata:
      labels:
        app: new
   spec:
     containers:
     - name: my-container
       image: ruchithabtgowda/proj2:latest
       ports:
       - containerPort: 80

```

Commit the changes



Commit message

Update deployment.yml

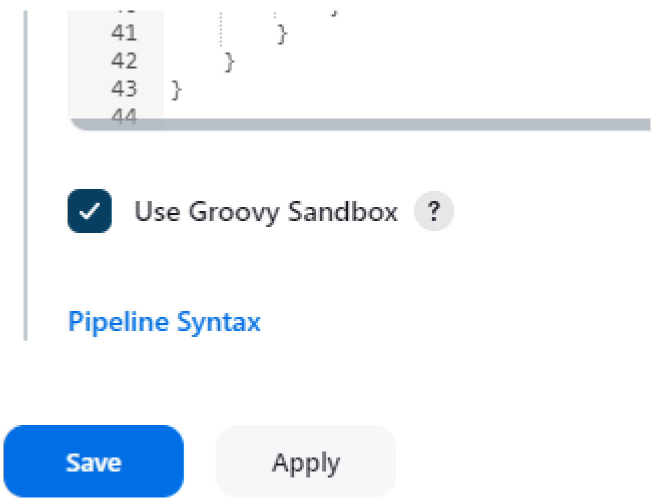
Extended description

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

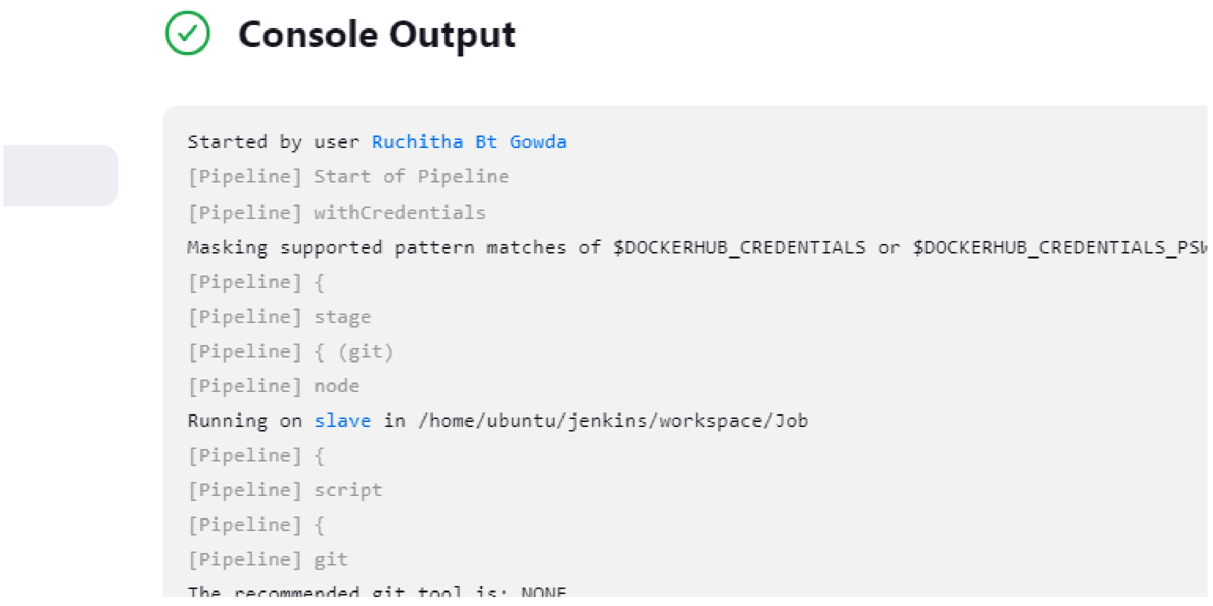
Apply and save the pipeline code



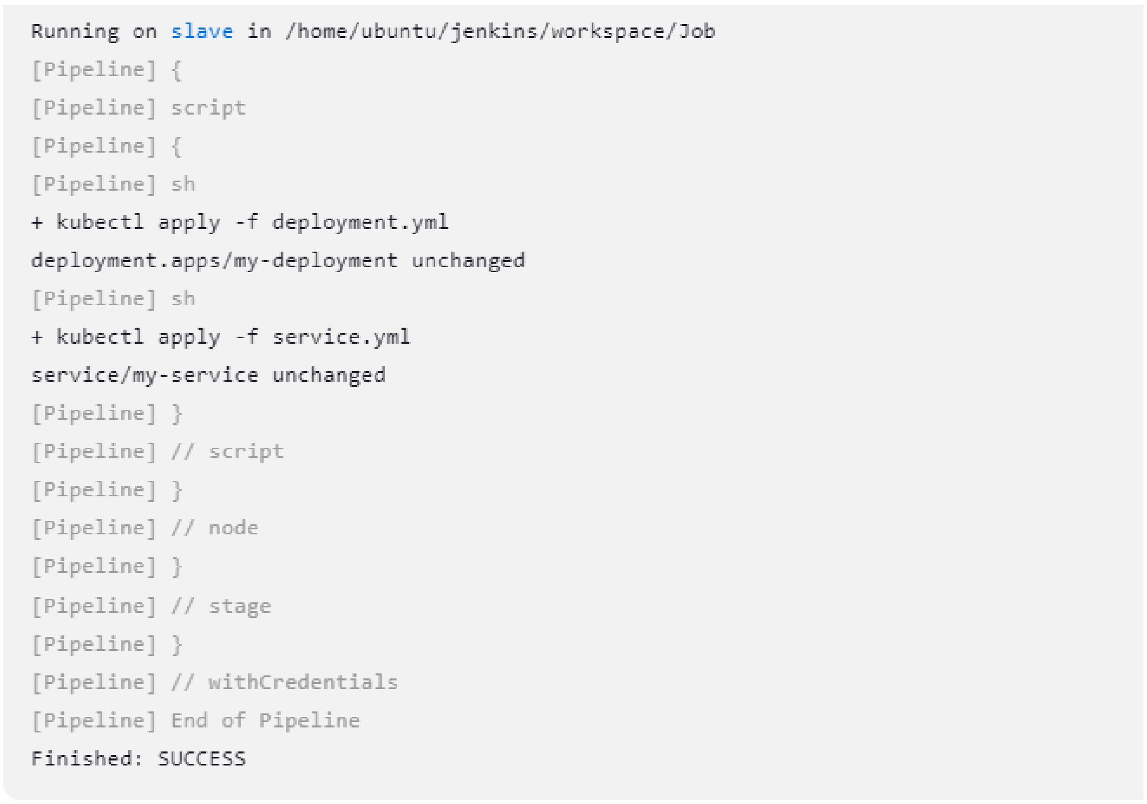
```
            }
        }
}
```

Build the job

View the condole output

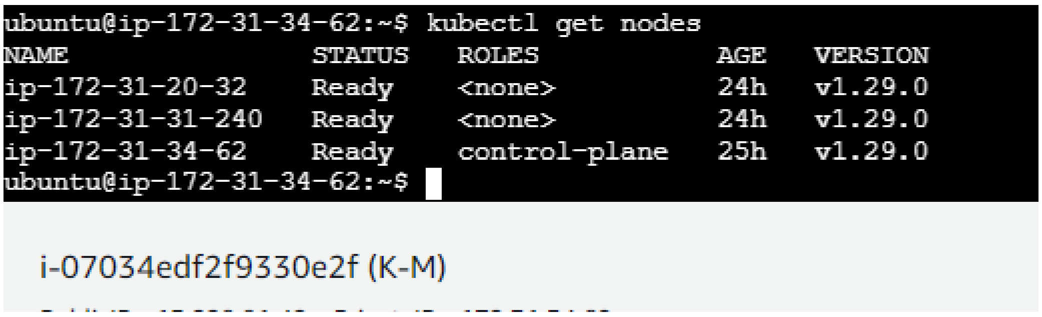


```
Started by user Ruchitha Bt Gowda
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (git)
[Pipeline] node
Running on slave in /home/ubuntu/jenkins/workspace/Job
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
```



```
Running on slave in /home/ubuntu/jenkins/workspace/Job
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ kubectl apply -f deployment.yml
deployment.apps/my-deployment unchanged
[Pipeline] sh
+ kubectl apply -f service.yml
service/my-service unchanged
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS
```

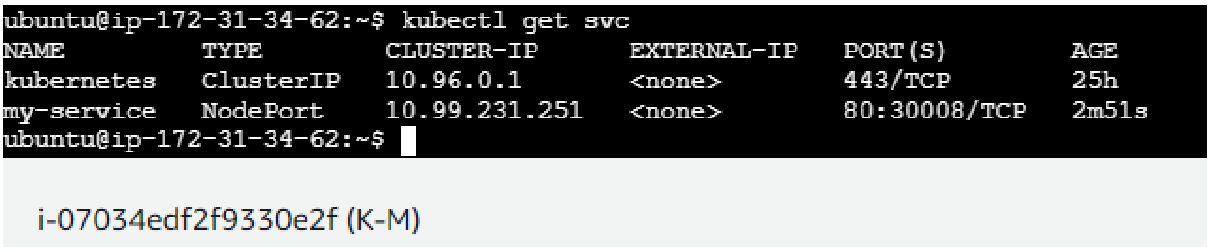
Check the nodes



```
ubuntu@ip-172-31-34-62:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-20-32    Ready    <none>          24h   v1.29.0
ip-172-31-31-240   Ready    <none>          24h   v1.29.0
ip-172-31-34-62    Ready    control-plane   25h   v1.29.0
ubuntu@ip-172-31-34-62:~$
```

i-07034edf2f9330e2f (K-M)

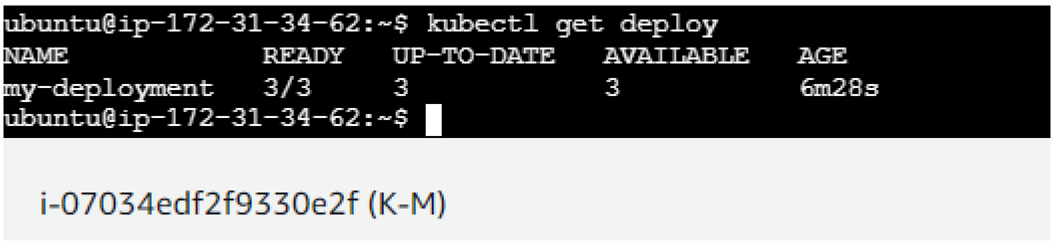
Check the services



```
ubuntu@ip-172-31-34-62:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        25h
my-service   NodePort    10.99.231.251   <none>        80:30008/TCP   2m51s
ubuntu@ip-172-31-34-62:~$
```

i-07034edf2f9330e2f (K-M)

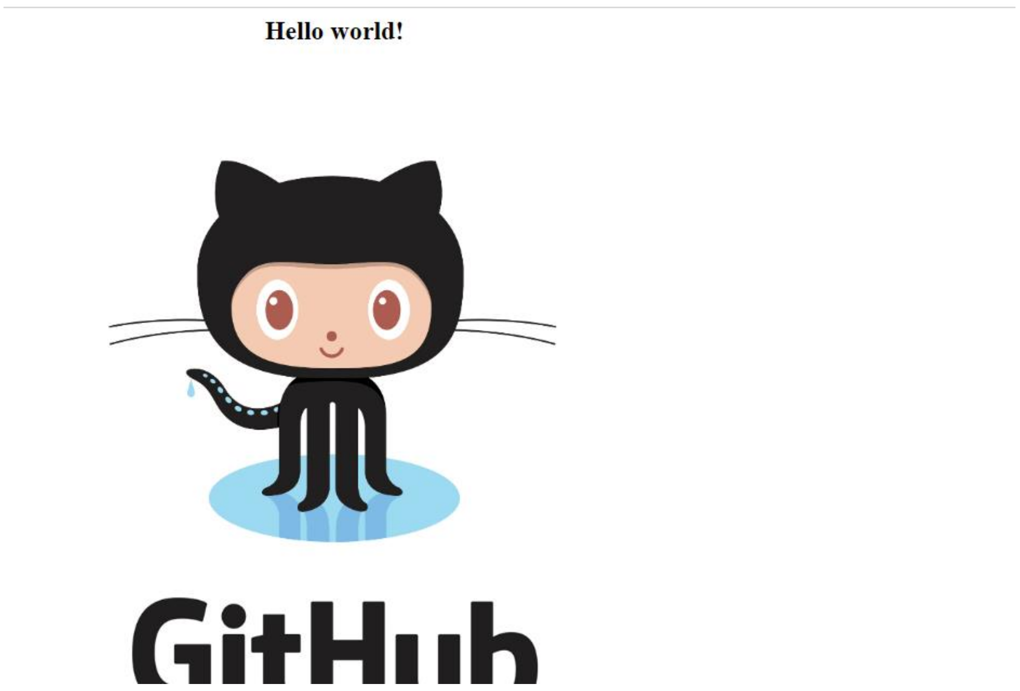
Check the deployment



```
ubuntu@ip-172-31-34-62:~$ kubectl get deploy
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
my-deployment   3/3     3            3           6m28s
ubuntu@ip-172-31-34-62:~$
```

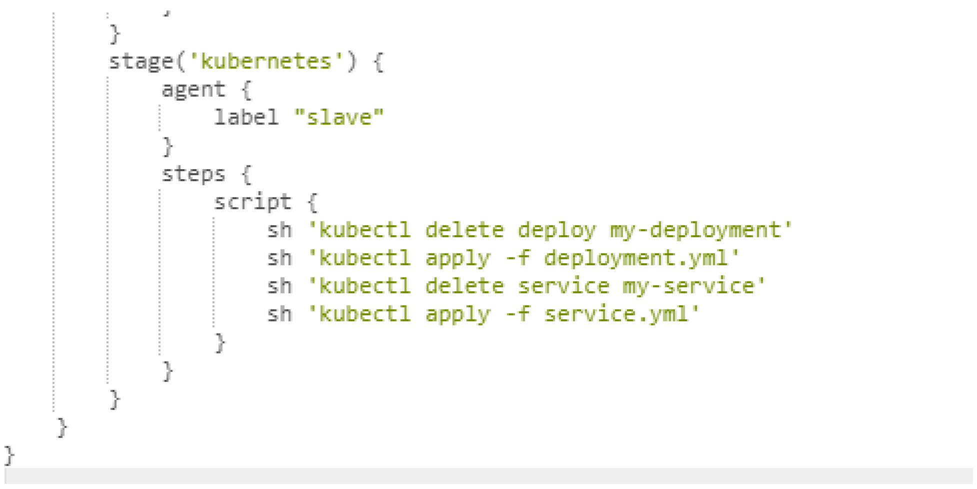
i-07034edf2f9330e2f (K-M)

Copy public ip of Kubernetes master followed by the port number – 30008



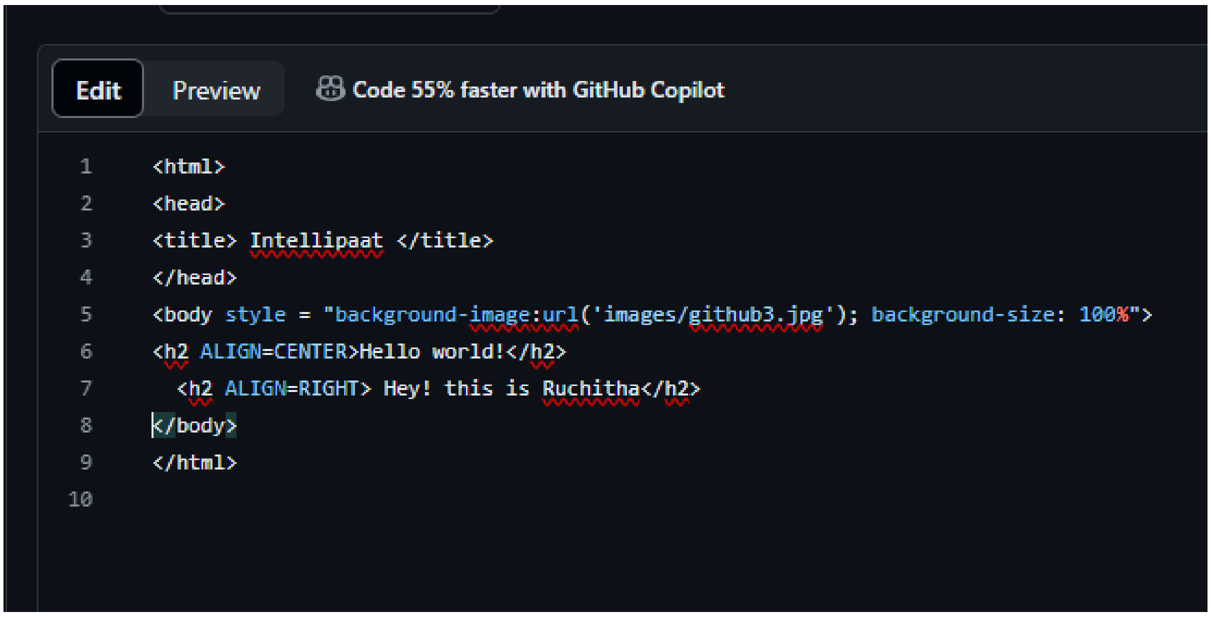
Hello world!

Now comment the line which says delete k8s before building new



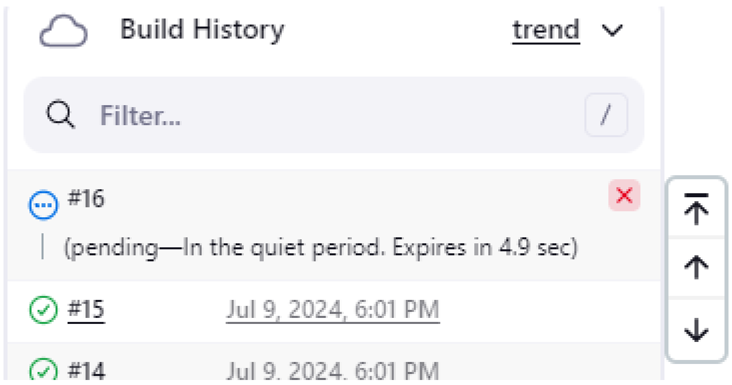
```
        }
        stage('kubernetes') {
            agent {
                label "slave"
            }
            steps {
                script {
                    sh 'kubectl delete deploy my-deployment'
                    sh 'kubectl apply -f deployment.yml'
                    sh 'kubectl delete service my-service'
                    sh 'kubectl apply -f service.yml'
                }
            }
        }
    }
}
```

You can perform some changes in html file and commit the changes again



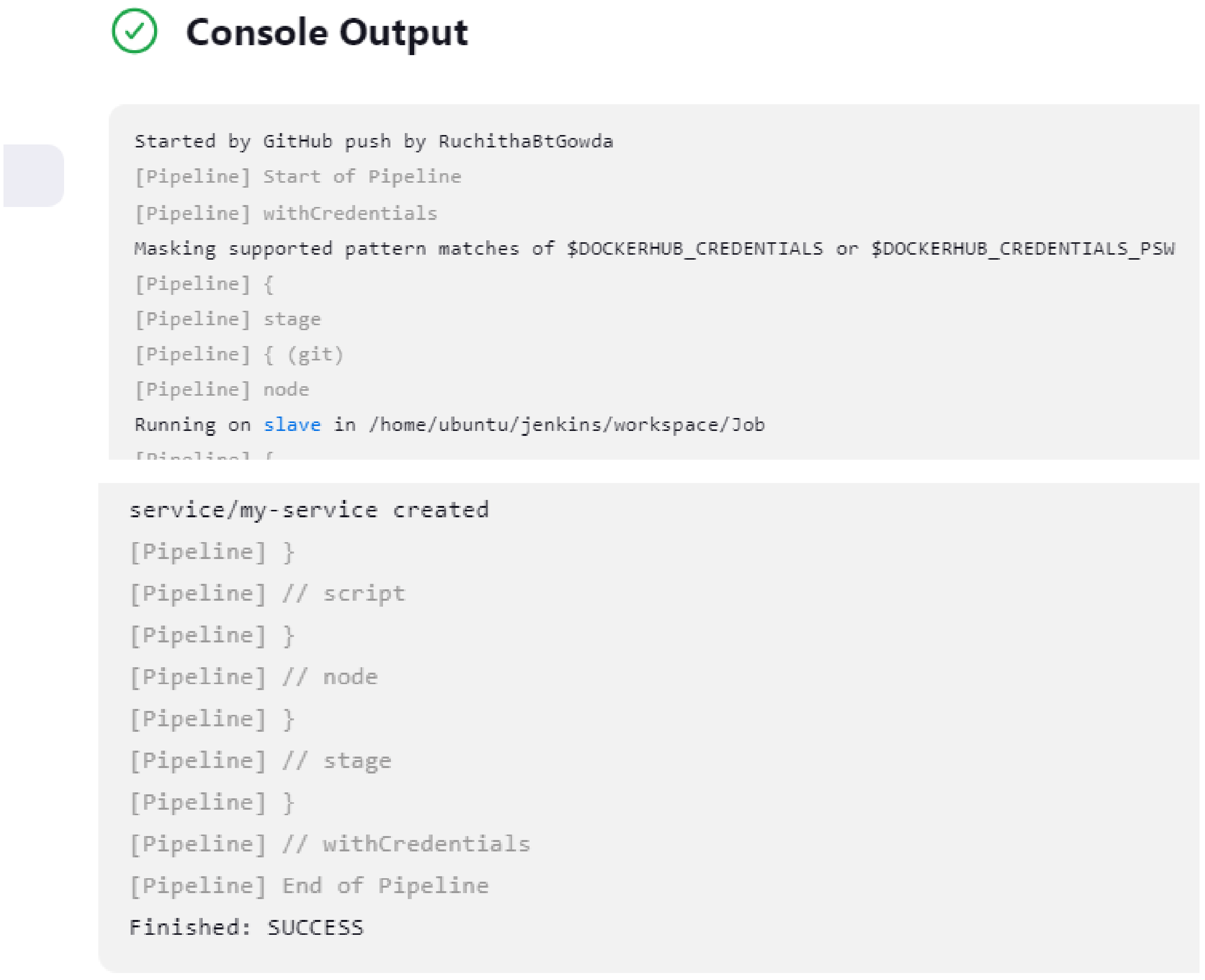
```
<html>
<head>
<title> Intellipaat </title>
</head>
<body style = "background-image:url('images/github3.jpg'); background-size: 100%">
<h2 ALIGN=CENTER>Hello world!</h2>
  <h2 ALIGN=RIGHT> Hey! this is Ruchitha</h2>
</body>
</html>
```

Job building will happen automatically



```
#16
(pending—In the quiet period. Expires in 4.9 sec)
#15    Jul 9, 2024, 6:01 PM
#14    Jul 9, 2024, 6:01 PM
```

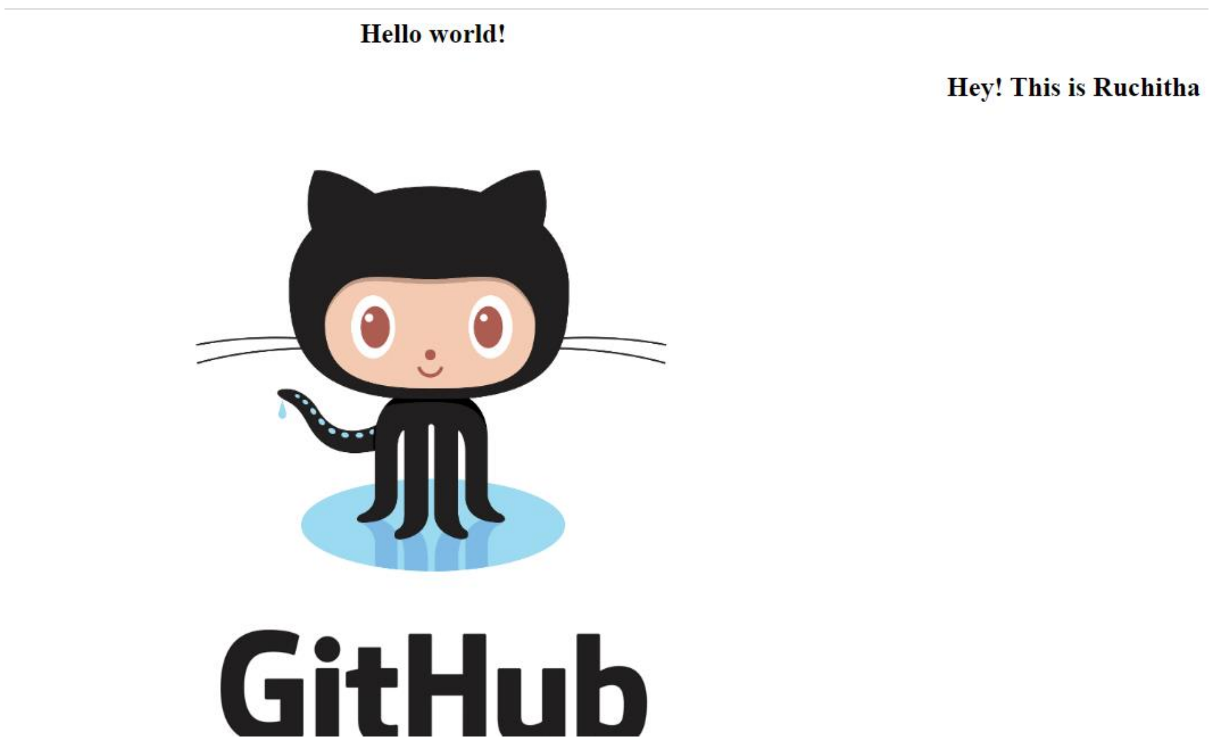
Check the console output



```
Started by GitHub push by RuchithaBtGowda
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (git)
[Pipeline] node
Running on slave in /home/ubuntu/jenkins/workspace/Job
```

```
service/my-service created
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Refresh the page, you can see the change.



Hello world!

Hey! This is Ruchitha