

# DevOps Capstone Project 2;

Running Instance(t2.medium);

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, there are several sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main content area displays a table with one row for 'Machine-1 (Main)'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability. The instance is listed as 'Running' on the 't2.medium' type. Below the table, a detailed view for 'i-0eab16ea279ce59af (Machine-1 (Main))' is shown with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is selected, showing the Instance summary. It includes fields for Instance ID (i-0eab16ea279ce59af), Public IPv4 address (54.208.109.220), Private IPv4 addresses (172.31.85.206), Public IPv4 DNS (ec2-54-208-109-220.compute-1.amazonaws.com), and Instance state (Running). Other fields like IPv6 address, Hostname type, and Private IP DNS name (IPv4 only) are also present.

Installing the terraform;

The image contains three screenshots of a web browser window showing EC2 Instance Connect sessions. The top screenshot shows the AWS EC2 Instances page with a single instance named 'Machine-1 (Main)'. The middle screenshot shows the EC2 Instance Connect interface for this instance, displaying a terminal session where the user runs 'history' to show a command history. The commands listed are: sudo apt update, sudo apt upgrade -y, sudo apt-get install -y gnupg software-properties-common curl, curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg, echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com \$(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list, sudo apt update, sudo apt-get install terraform, terraform -v, clear, and history. The bottom screenshot shows another EC2 Instance Connect session for the same instance, where the user runs 'terraform -v' and gets the output 'Terraform v1.9.7' and 'on linux amd64'.

Run “Terraform Script” to Create Other Three Instances;

Creating main.tf file;

```
GNU nano 7.2
[Alt+S] main.tf *
provider "aws" {
    region = "us-east-2"
}
resource "aws_instance" "Kubernetes_Master" {
    ami           = "ami-0ea3c35c5c3284d82"
    instance_type = "t2.medium"
    subnet_id     = "subnet-044e8983b687d0058"
    key_name      = "Jipsy"
    tags = [
        Name = "Machine-3"
    ]
}
resource "aws_instance" "Kubernetes_Slave1" {
    ami           = "ami-0ea3c35c5c3284d82"
    instance_type = "t2.micro"
    subnet_id     = "subnet-0efe3e08a4f44f06b"
    key_name      = "Jipsy"
    tags = [
        Name = "Machine-2"
    ]
}
resource "aws_instance" "Kubernetes_Slave2" {
    ami           = "ami-0ea3c35c5c3284d82"
    instance_type = "t2.micro"
    subnet_id     = "subnet-06a93ed1ed336a340"
    key_name      = "Jipsy"
    tags = [
        Name = "Machine-4"
    ]
}

^G Help          ^C Write Out    ^W Where Is      ^R Cut          ^T Execute      ^C Location      M-U Undo
^X Exit          ^R Read File    ^A Replace       ^U Paste         ^J Justify      ^Y Go To Line    M-E Redo
M-A Set Mark    M-C Copy       M-J To Bracket
M-G Copy        M-Q Where Was
```

i-02097984ee63fdcb5 (Machine-1 (Main))  
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127  
Untitled - Notepad  
File Edit Format View Help

```
provider "aws" {
    region = "us-east-2"
}
resource "aws_instance" "Kubernetes_Master" {
    ami           = "ami-0ea3c35c5c3284d82"
    instance_type = "t2.medium"
    subnet_id     = "subnet-044e8983b687d0058"
    key_name      = "Jipsy"
    tags = [
        Name = "Machine-3"
    ]
}
resource "aws_instance" "Kubernetes_Slave1" {
    ami           = "ami-0ea3c35c5c3284d82"
    instance_type = "t2.micro"
    subnet_id     = "subnet-0efe3e08a4f44f06b"
    key_name      = "Jipsy"
    tags = [
        Name = "Machine-2"
    ]
}
resource "aws_instance" "Kubernetes_Slave2" {
    ami           = "ami-0ea3c35c5c3284d82"
    instance_type = "t2.micro"
    subnet_id     = "subnet-06a93ed1ed336a340"
    key_name      = "Jipsy"
    tags = [
        Name = "Machine-4"
    ]
}
```

Terraform init;

```
ubuntu@ip-172-31-27-127:~$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.71.0...
- Installed hashicorp/aws v5.71.0 (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.

```
ubuntu@ip-172-31-27-127:~$
```

i-02097984ee63fdcb5 (Machine-1 (Main))  
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127

## Configuring my AWS account details

The screenshot shows two terminal windows side-by-side. The left window displays the command `aws configure` being run on an EC2 instance, with the AWS Access Key ID, Secret Access Key, Region, and Output Format all set to 'None'. The right window shows the output of the `terraform plan` command, which generates an execution plan for creating a Kubernetes Master instance. The plan includes resource details like AMI, ARN, and instance type, along with their known states after apply.

```
ubuntu@ip-172-31-27-127:~$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: us-east-2
Default output format [None]: json
ubuntu@ip-172-31-27-127:~$ [Alt+S]

ubuntu@ip-172-31-27-127:~$ terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.Kubernetes_Master will be created
+ resource "aws_instance" "Kubernetes_Master" {
  + ami = "ami-0ea3c35c5c3284d82"
  + arn = "(known after apply)"
  + associate_public_ip_address = "(known after apply)"
  + availability_zone = "(known after apply)"
  + cpu_core_count = "(known after apply)"
  + cpu_threads_per_core = "(known after apply)"
  + disable_api_stop = "(known after apply)"
  + disable_api_termination = "(known after apply)"
  + ebs_optimized = "(known after apply)"
  + get_password_data = "false"
  + host_id = "(known after apply)"
  + host_resource_group_arn = "(known after apply)"
  + iam_instance_profile = "(known after apply)"
  + id = "(known after apply)"
  + instance_initiated_shutdown_behavior = "(known after apply)"

i-02097984ee63fdcb5 (Machine-1 (Main))
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127
```

## terraform apply;

The screenshot shows a terminal window where the `terraform apply` command is being run. It prompts for confirmation and then executes the plan. The output shows the creation of three instances: a Kubernetes Master and two slaves. The status messages indicate the progress of each instance's creation, with some taking longer than others due to network latency.

```
ubuntu@ip-172-31-27-127:~$ [Alt+S]
}
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.Kubernetes_Slave1: Creating...
aws_instance.Kubernetes_Master: Creating...
aws_instance.Kubernetes_Slave2: Creating...
aws_instance.Kubernetes_Slave1: Still creating... [10s elapsed]
aws_instance.Kubernetes_Master: Still creating... [10s elapsed]
aws_instance.Kubernetes_Slave2: Still creating... [10s elapsed]
aws_instance.Kubernetes_Master: Creation complete after 13s [id=i-027d7477da832a469]
aws_instance.Kubernetes_Slave2: Creation complete after 13s [id=i-0f34141aec9505e5e]
aws_instance.Kubernetes_Slave1: Still creating... [20s elapsed]
aws_instance.Kubernetes_Slave1: Still creating... [30s elapsed]
aws_instance.Kubernetes_Slave2: Creation complete after 32s [id=i-0193524f40d25bd58]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-27-127:~$ [Alt+S]

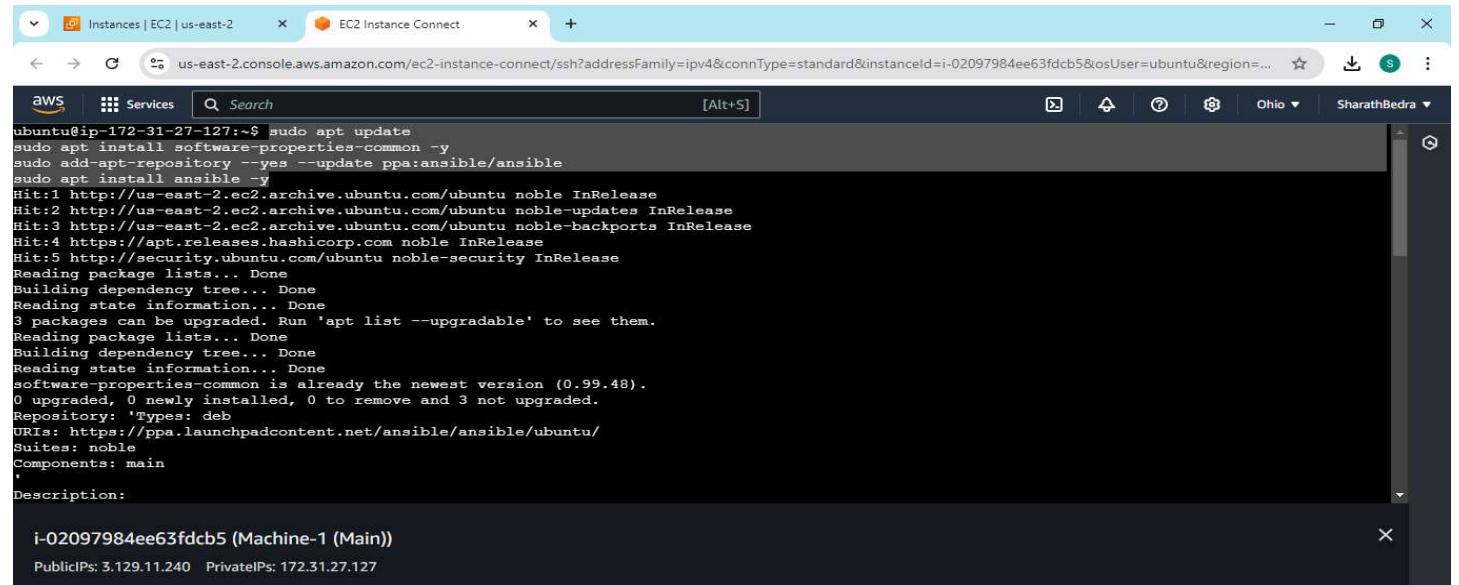
i-02097984ee63fdcb5 (Machine-1 (Main))
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127
```

## Instances got launched by terraform;

The screenshot shows the AWS EC2 Instances page with four instances listed: 'Kub-Slave2' (running, t2.micro), 'Kub-Master' (running, t2.micro), 'Machine-1 (Main)' (running, t2.medium), and 'Kub-Slave1' (running, t2.medium). The 'Kub-Slave2' instance is currently selected. The page also shows instance-level settings like auto-recovery, lifecycle, and stop-hibernate behavior.

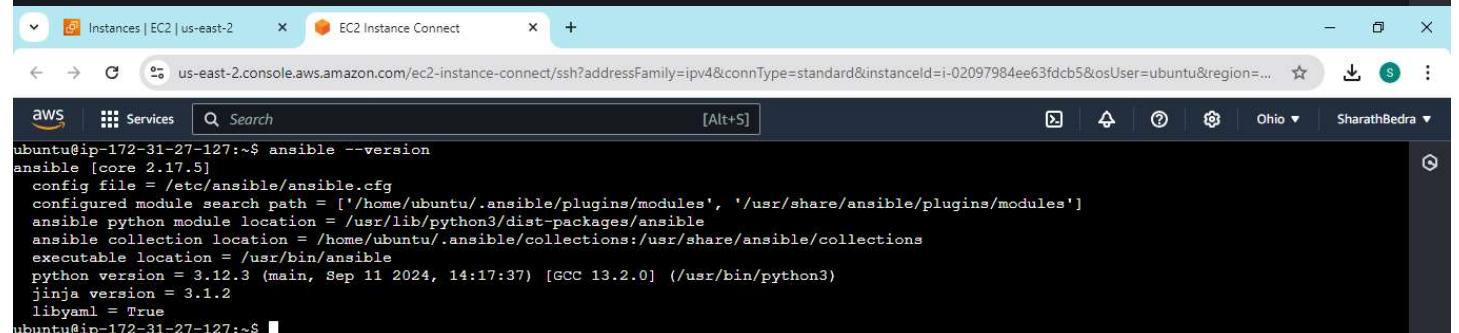
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
Kub-Slave2	i-0f34141aec9505e5e	Running	t2.micro	Initializing	View alarms	us-east-2a
Kub-Master	i-0193524f40d25bd58	Running	t2.micro	2/2 checks passed	View alarms	us-east-2a
Machine-1 (Main)	i-02097984ee63fdcb5	Running	t2.medium	2/2 checks passed	View alarms	us-east-2a
Kub-Slave1	i-027d7477da832a469	Running	t2.medium	Initializing	View alarms	us-east-2a

## Install “Ansible” on Machine 1 (main);



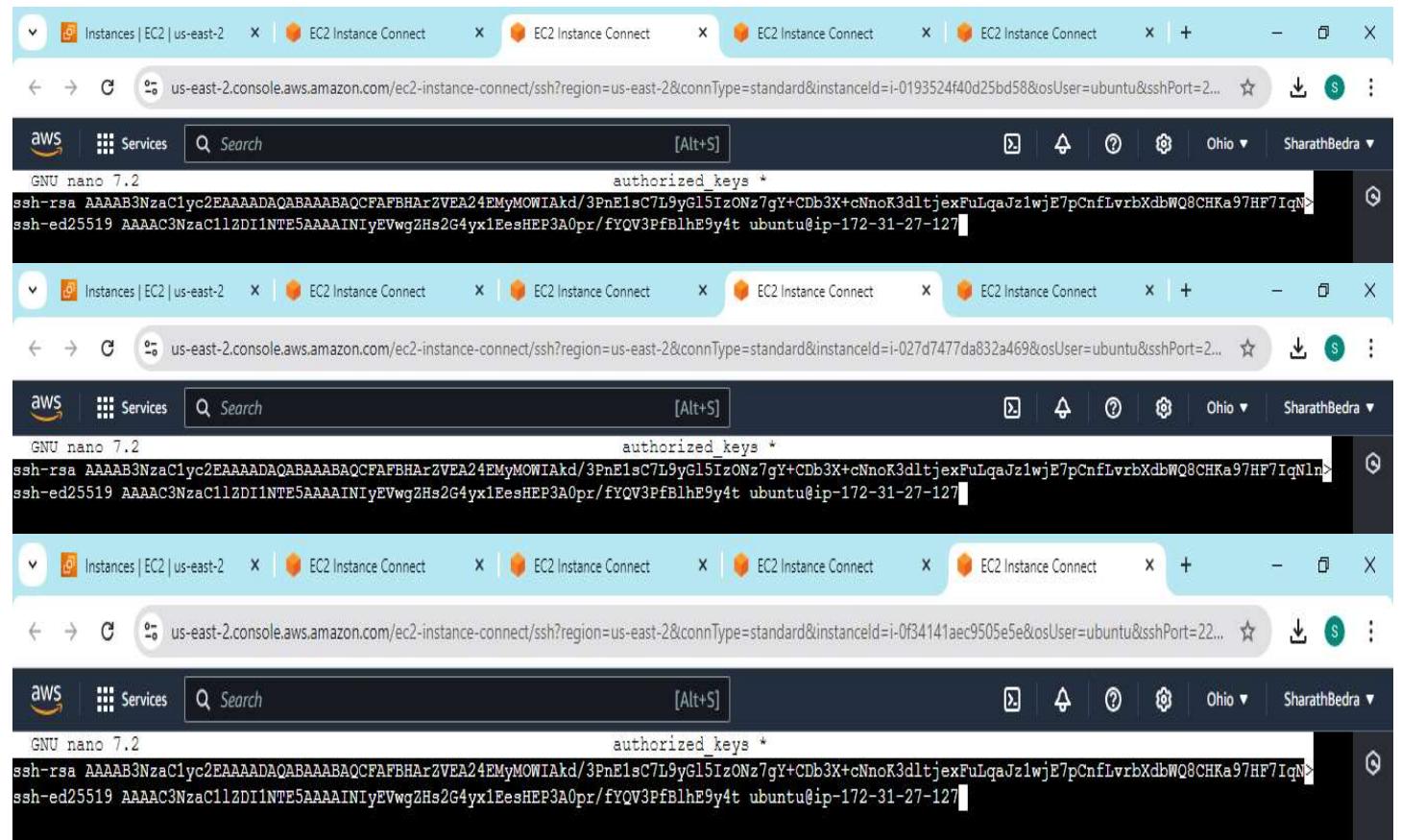
```
ubuntu@ip-172-31-27-127:~$ sudo apt update
sudo apt install software-properties-common -y
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://apt.releases.hashicorp.com noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.48).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
'
Description:

i-02097984ee63fdcb5 (Machine-1 (Main))
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127
```



```
ubuntu@ip-172-31-27-127:~$ ansible --version
ansible [core 2.17.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-27-127:~$
```

We shall Paste this key content to all three machines individually.



```
GNU nano 7.2
authorized_keys *
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCFAFBHAzZVEA24EMyMOWIAkd/3PnElsc7L9yGl5IzONz7gY+CDb3X+cNnoK3dltjexFuLqaJz1wjE7pCnfLvrkXdbWQ8CHKa97HF7IqN>
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINiEVwgZHs2G4yx1EesHEP3A0pr/fYQV3PfBlhE9y4t ubuntu@ip-172-31-27-127

GNU nano 7.2
authorized_keys *
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCFAFBHAzZVEA24EMyMOWIAkd/3PnElsc7L9yGl5IzONz7gY+CDb3X+cNnoK3dltjexFuLqaJz1wjE7pCnfLvrkXdbWQ8CHKa97HF7IqNln>
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINiEVwgZHs2G4yx1EesHEP3A0pr/fYQV3PfBlhE9y4t ubuntu@ip-172-31-27-127

GNU nano 7.2
authorized_keys *
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCFAFBHAzZVEA24EMyMOWIAkd/3PnElsc7L9yGl5IzONz7gY+CDb3X+cNnoK3dltjexFuLqaJz1wjE7pCnfLvrkXdbWQ8CHKa97HF7IqN>
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINiEVwgZHs2G4yx1EesHEP3A0pr/fYQV3PfBlhE9y4t ubuntu@ip-172-31-27-127
```

# We shall change the names of our instances as per the requirement of the project;

The screenshot shows the AWS Management Console with the EC2 Instances page open. There are four instances listed:

Name	Instance ID	Instance state	Type	Status check	Alarm status	Availability zone
Machine-2	i-0f34141aec9505e5e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a
Machine-4	i-0193524f40d25bd58	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a
Machine-1 (Main)	i-02097984ee63fdcb5	Running	t2.medium	2/2 checks passed	View alarms +	us-east-2b
Machine-3	i-027d74777da832a469	Running	t2.medium	2/2 checks passed	View alarms +	us-east-2b

Paste the slaves & master private IP Addresses and pinging them all;

The screenshot shows the AWS CloudShell terminal. The inventory file is displayed:

```
GNU nano 7.2
## 10.25.1.56
## 10.25.1.57

# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':
## [Debian]
## alpha.example.org
## beta.example.org

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

[master]
Machine-3 ansible_host=172.31.27.128
[slaves]
Machine-2 ansible_host=172.31.34.129
Machine-4 ansible_host=172.31.12.52
```

The screenshot shows the AWS CloudShell terminal executing the command `ansible -m ping all`:

```
ubuntu@ip-172-31-27-127:~$ ansible -m ping all
[WARNING]: Platform linux on host Machine-2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
Machine-2 | SUCCESS => {
    "ansible_facts": [
        "discovered_interpreter_python": "/usr/bin/python3.12"
    ],
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host Machine-3 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
Machine-3 | SUCCESS => {
    "ansible_facts": [
        "discovered_interpreter_python": "/usr/bin/python3.12"
    ],
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host Machine-4 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
Machine-4 | SUCCESS => {
    "ansible_facts": [
        "discovered_interpreter_python": "/usr/bin/python3.12"
    ],
    "changed": false,
    "ping": "pong"
}
```

# Create Three Scripts for Installing Required Tools on Machines;

```
script1.sh *
sudo apt-get update
sudo apt-get install openjdk-17-jre-headless -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y

i-02097984ee63fdcb5 (Machine-1 (Main))
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127
```

```
script2.sh *
sudo apt-get update
sudo apt-get install openjdk-17-jre-headless -y
sudo apt-get install docker.io -y
sudo systemctl enable --now docker
sudo swapoff -a
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/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:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
```

```
script3.sh *
sudo apt-get update
sudo apt-get install docker.io -y
sudo systemctl enable --now docker
sudo swapoff -a
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/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:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
```

```
i-02097984ee63fdcb5 (Machine-1 (Main))
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127
```

## Create the Playbooks to Run these Scripts to Install the Much Needed Tools;

```
aws Services Search [Alt+S] Ohio SharathBedra
ubuntu@ip-172-31-27-127:~$ sudo cat play.yaml
---
- name: install Jenkins & Java on Machine-1
  become: true
  hosts: localhost
  tasks:
    - name: running script1
      script: script1.sh

- name: install Java, Docker & Kubernetes on Machine-3
  become: true
  hosts: master
  tasks:
    - name: running script2
      script: script2.sh

- name: install Docker & Kubernetes on Machine-2&4
  become: true
  hosts: slaves
  tasks:
    - name: running script3
      script: script3.sh
ubuntu@ip-172-31-27-127:~$
```

i-02097984ee63fdcb5 (Machine-1 (Main))  
PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127

```
aws Services Search [Alt+S] Ohio SharathBedra
ubuntu@ip-172-31-27-127:~$ ansible-playbook play.yaml --check
PLAY [install Jenkins & Java on Machine-1] ****
TASK [Gathering Facts] ****
ok: [localhost]
TASK [running script1] ****
skipping: [localhost]
PLAY [install Java, Docker & Kubernetes on Machine-3] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host Machine-3 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [Machine-3]
TASK [running script2] ****
skipping: [Machine-3]
PLAY [install Docker & Kubernetes on Machine-2&4] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host Machine-2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [Machine-2]
[WARNING]: Platform linux on host Machine-4 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [Machine-4]
TASK [running script3] ****
skipping: [Machine-2]
skipping: [Machine-4]
PLAY RECAP ****
Machine-2 : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
Machine-3 : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
Machine-4 : 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
```

Now, we will run the “play.yaml”;

```
aws Services Search [Alt+S] Ohio SharathBedra
ubuntu@ip-172-31-27-127:~$ ansible-playbook play.yaml
PLAY [install Jenkins & Java on Machine-1] ****
TASK [Gathering Facts] ****
ok: [localhost]
TASK [running script1] ****
changed: [localhost]
PLAY [install Java, Docker & Kubernetes on Machine-3] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host Machine-3 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [Machine-3]
TASK [running script2] ****
changed: [Machine-3]
PLAY [install Docker & Kubernetes on Machine-2&4] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host Machine-2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [Machine-2]
[WARNING]: Platform linux on host Machine-4 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [Machine-4]
TASK [running script3] ****
changed: [Machine-2]
changed: [Machine-4]
PLAY RECAP ****
Machine-2 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Machine-3 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Machine-4 : 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
```

## Configure Kubernetes Slaves Properly on Machine-3;

```
ubuntu@ip-172-31-27-128:~$ sudo kubeadm init --apiserver-advertise-address=172.31.27.128
I1015 15:45:33.541189 8232 version.go:256] remote version is much newer: v1.31.1; falling back to: stable-1.28
[kubelet] Using Kubernetes version: v1.28.14
[preflight] Running pre-flight checks
[WARNING FileExisting-socat]: socat not found in system path
[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'
W1015 15:45:41.702368 8232 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent
with that used by kubeadm. It is recommended that using "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-27-128 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.27.128]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-27-128 localhost] and IPs [172.31.27.128 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-27-128 localhost] and IPs [172.31.27.128 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
```

i-027d7477da832a469 (Machine-3)

PublicIPs: 18.118.6.221 PrivateIPs: 172.31.27.128

```
[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 172.31.27.128:6443 --token 6o8xpk.7blf3j5lc4mbxwus \
--discovery-token-ca-cert-hash sha256:1fba67e73a66375a5c2b3ebb799315ea4ab1920bdd7ad0b5170929551faf1d88
ubuntu@ip-172-31-27-128:~$
```

i-027d7477da832a469 (Machine-3)

PublicIPs: 18.118.6.221 PrivateIPs: 172.31.27.128

Copy this token & Paste it into Machine-2 & Machine-4 one by one using the “sudo” command.

```
ubuntu@ip-172-31-34-129:~$ sudo kubeadm join 172.31.27.128:6443 --token 6o8xpk.7blf3j5lc4mbxwus \
--discovery-token-ca-cert-hash sha256:1fba67e73a66375a5c2b3ebb799315ea4ab1920bdd7ad0b5170929551faf1d88 --v=5
I1015 15:50:53.988755 5592 join.go:412] [preflight] found NodeName empty; using OS hostname as NodeName
I1015 15:50:53.989222 5592 initConfiguration.go:117] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
I1015 15:50:53.989650 5592 preflight.go:93] [preflight] Running general checks
I1015 15:50:53.989826 5592 checks.go:280] validating the existence of file /etc/kubernetes/kubelet.conf
I1015 15:50:53.989959 5592 checks.go:280] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I1015 15:50:53.990049 5592 checks.go:104] validating the container runtime
I1015 15:50:54.279297 5592 checks.go:639] validating whether swap is enabled or not
I1015 15:50:54.279426 5592 checks.go:370] validating the presence of executable crictl
I1015 15:50:54.279623 5592 checks.go:370] validating the presence of executable conntrack
I1015 15:50:54.279667 5592 checks.go:370] validating the presence of executable ip
I1015 15:50:54.279739 5592 checks.go:370] validating the presence of executable iptables
I1015 15:50:54.279865 5592 checks.go:370] validating the presence of executable mount
I1015 15:50:54.279953 5592 checks.go:370] validating the presence of executable nsenter
I1015 15:50:54.280021 5592 checks.go:370] validating the presence of executable ebttables
I1015 15:50:54.280890 5592 checks.go:370] validating the presence of executable ethtool
I1015 15:50:54.281059 5592 checks.go:370] validating the presence of executable socat
[WARNING FileExisting-socat]: socat not found in system path
I1015 15:50:54.282039 5592 checks.go:370] validating the presence of executable tc
I1015 15:50:54.282148 5592 checks.go:370] validating the presence of executable touch
I1015 15:50:54.282264 5592 checks.go:516] running all checks
I1015 15:50:54.295535 5592 checks.go:401] checking whether the given node name is valid and reachable using net.LookupHost
```

i-0f34141aec9505e5e (Machine-2)

PublicIPs: 52.14.142.163 PrivateIPs: 172.31.34.129

Instances | EC2 | us-east-2 | EC2 Instance Connect | Instances | Services | Search | [Alt+S] | Ohio | SharathBedra

```
I1015 15:50:54.375588 5592 interface.go:443] Found active IP 172.31.34.129
I1015 15:50:54.380040 5592 preflight.go:104] [preflight] Running configuration dependant checks
I1015 15:50:54.380359 5592 controlplaneprepare.go:225] [download-certs] Skipping certs download
I1015 15:50:54.380440 5592 kubelet.go:121] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I1015 15:50:54.380987 5592 kubelet.go:136] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I1015 15:50:54.381641 5592 kubelet.go:157] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-34-129" and status is "Ready"
I1015 15:50:54.384680 5592 kubelet.go:172] [kubelet-start] Stopping the kubelet
[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...
I1015 15:50:55.624368 5592 cert_rotation.go:137] Starting client certificate rotation controller
I1015 15:50:55.624944 5592 kubelet.go:220] [kubelet-start] preserving the crisocket information for the node
I1015 15:50:55.624993 5592 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31-34-129" as an annotation

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.

ubuntu@ip-172-31-34-129:~$
```

### i-0f34141aec9505e5e (Machine-2)

PublicIPs: 52.14.142.163 PrivateIPs: 172.31.34.129

Instances | EC2 | us-east-2 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Instances | Services | Search | [Alt+S] | Ohio | SharathBedra

```
ubuntu@ip-172-31-12-52:~$ sudo kubeadm join 172.31.27.128:6443 --token 668xpk.7bf1f3j5lc4mbxwus \
--discovery-token-ca-cert-hash sha256:1fba67e73a66375a5c2b3eb799315ea4ab1920bdd7ad0b5170929551faf1d88 --v=5
I1015 15:51:29.560959 5621 join.go:412] [preflight] found NodeName empty; using OS hostname as NodeName
I1015 15:51:29.562170 5621 initconfiguration.go:117] detected and using CRI socket: unix:///var/run/containerd/containerd.sock
[preflight] Running pre-flight checks
I1015 15:51:29.562230 5621 preflight.go:93] [preflight] Running general checks
I1015 15:51:29.562298 5621 checks.go:280] validating the existence of file /etc/kubernetes/kubelet.conf
I1015 15:51:29.562311 5621 checks.go:280] validating the existence of file /etc/kubernetes/bootstrap-kubelet.conf
I1015 15:51:29.562322 5621 checks.go:104] validating the container runtime
I1015 15:51:29.831684 5621 checks.go:639] validating whether swap is enabled or not
I1015 15:51:29.832070 5621 checks.go:370] validating the presence of executable crictl
I1015 15:51:29.832148 5621 checks.go:370] validating the presence of executable conctrack
I1015 15:51:29.832199 5621 checks.go:370] validating the presence of executable ip
I1015 15:51:29.832297 5621 checks.go:370] validating the presence of executable iptables
I1015 15:51:29.832509 5621 checks.go:370] validating the presence of executable mount
I1015 15:51:29.832611 5621 checks.go:370] validating the presence of executable nsenter
I1015 15:51:29.832857 5621 checks.go:370] validating the presence of executable ebtables
I1015 15:51:29.832982 5621 checks.go:370] validating the presence of executable ethtool
I1015 15:51:29.833240 5621 checks.go:370] validating the presence of executable socat
[WARNING FileExisting-socat]: socat not found in system path
I1015 15:51:29.833545 5621 checks.go:370] validating the presence of executable tc
I1015 15:51:29.833692 5621 checks.go:370] validating the presence of executable touch
I1015 15:51:29.833772 5621 checks.go:516] running all checks
I1015 15:51:29.854116 5621 checks.go:401] checking whether the given node name is valid and reachable using net.LookupHost
```

### i-0193524f40d25bd58 (Machine-4)

PublicIPs: 3.145.180.160 PrivateIPs: 172.31.12.52

Instances | EC2 | us-east-2 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Instances | Services | Search | [Alt+S] | Ohio | SharathBedra

```
I1015 15:51:29.970527 5621 interface.go:443] Found active IP 172.31.12.52
I1015 15:51:29.978016 5621 preflight.go:104] [preflight] Running configuration dependant checks
I1015 15:51:29.978290 5621 controlplaneprepare.go:225] [download-certs] Skipping certs download
I1015 15:51:29.978528 5621 kubelet.go:121] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I1015 15:51:29.979507 5621 kubelet.go:136] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I1015 15:51:29.980571 5621 kubelet.go:157] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-12-52" and status is "Ready"
I1015 15:51:29.983756 5621 kubelet.go:172] [kubelet-start] Stopping the kubelet
[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...
I1015 15:51:31.367029 5621 cert_rotation.go:137] Starting client certificate rotation controller
I1015 15:51:31.367984 5621 kubelet.go:220] [kubelet-start] preserving the crisocket information for the node
I1015 15:51:31.368231 5621 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/containerd/containerd.sock" to the Node API object "ip-172-31-12-52" as an annotation

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.

ubuntu@ip-172-31-12-52:~$
```

### i-0193524f40d25bd58 (Machine-4)

PublicIPs: 3.145.180.160 PrivateIPs: 172.31.12.52

## Verifying the nodes;

Instances | EC2 | us-east-2 | EC2 Instance Connect | Instances | Services | Search | [Alt+S] | Ohio | SharathBedra

```
ubuntu@ip-172-31-27-128:~$ kubectl get nodes
NAME        STATUS    ROLES     AGE      VERSION
ip-172-31-12-52  NotReady <none>    23s   v1.28.14
ip-172-31-27-128  NotReady control-plane 6m2s   v1.28.14
ip-172-31-34-129  NotReady <none>    58s   v1.28.14
ubuntu@ip-172-31-27-128:~$
```

# Install the “Calico Network” to run the cluster using the below-given command

The screenshot shows three separate AWS EC2 Instance Connect sessions, each running on a different EC2 instance. The top session shows the download of the Calico manifest file from GitHub. The middle session shows the creation of the Calico YAML file. The bottom session shows the application of the Calico manifest using kubectl, resulting in numerous custom resource definitions being created.

```
ubuntu@ip-172-31-27-128:~$ curl https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/calico.yaml -o calico.yaml
ubuntu@ip-172-31-27-128:~$ ls
calico.yaml
ubuntu@ip-172-31-27-128:~$ kubectl apply -f 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.apirextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ipservations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-kube-controllers created
deployment.apps/calico-node created
```

## Enabling Jenkins;

```
aws Services Search [Alt+S] Ohio ▾ SharathBedra ▾
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
  Active: active (running) since Tue 2024-10-15 15:33:39 UTC; 46min ago
    Main PID: 9374 (java)
       Tasks: 44 (limit: 4676)
      Memory: 700.7M (peak: 701.2M)
        CPU: 19.641s
       CGroup: /system.slice/jenkins.service
               └─9374 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --http
Port=8080

Oct 15 15:33:35 ip-172-31-27-127 jenkins[9374]: 6aab8eb3ce81489a927fa92e94955c5
Oct 15 15:33:35 ip-172-31-27-127 jenkins[9374]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 15 15:33:35 ip-172-31-27-127 jenkins[9374]: ****
Oct 15 15:33:35 ip-172-31-27-127 jenkins[9374]: ****
Oct 15 15:33:35 ip-172-31-27-127 jenkins[9374]: ****
Oct 15 15:33:35 ip-172-31-27-127 jenkins[9374]: 2024-10-15 15:33:39.504+0000 [id=33]           INFO      jenkins.InitReactorRunner$1
#onAttained
Oct 15 15:33:39 ip-172-31-27-127 jenkins[9374]: 2024-10-15 15:33:39.527+0000 [id=23]           INFO      hudson.lifecycle.Lifecycle$#
onReady: Jenkins
Oct 15 15:33:39 ip-172-31-27-127 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Oct 15 15:33:39 ip-172-31-27-127 jenkins[9374]: 2024-10-15 15:33:39.700+0000 [id=48]           INFO      h.m.DownloadService$Downloadable#load
Oct 15 15:33:39 ip-172-31-27-127 jenkins[9374]: 2024-10-15 15:33:39.700+0000 [id=48]           INFO      hudson.util.Retriger#start:
Performed t
Performe
lines 1-20/20 (END)
```

i-02097984ee63fdcb5 (Machine-1 (Main))

PublicIPs: 3.129.11.240 PrivateIPs: 172.31.27.127

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

.....

Created a Node "Kubernetes Master";

Instances | EC2 | us-east-1 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Nodes [Jenkins]

Not secure 3.129.11.240:8080/computer/

# Jenkins

Search (CTRL+K) Sharath Bedra log out

Dashboard > Nodes >

Nodes

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	2.39 GiB	! 0 B	2.39 GiB	0ms
2	Kubernetes Master	Linux (amd64)	In sync	2.27 GiB	! 0 B	2.27 GiB	32ms

Build Queue: No builds in the queue.

Build Executor Status: Built-In Node 0/2

Clouds: + New Node Configure Monitors

## Create the DockerHub Credentials for Jenkins Pipeline Creation;

Global credentials (unrestricted)

ID	Name	Kind	Description
pwdless	ubuntu (pwdless)	SSH Username with private key	pwdless
d4034313-614a-45b8-a0b6-ea6ced0bdf43	sharathkumar5802@gmail.com/*****	Username with password	

Icon: S M L

## Forked the given repository;

SharathBedra / website

Code Pull requests Actions Projects Wiki Security Insights Settings

website Public  
forked from hshar/website

master 1 Branch 0 Tags

This branch is up to date with hshar/website:master.

Ubuntu modified 683b439 · 5 years ago 2 Commits

images final 5 years ago

index.html modified 5 years ago

About

No description, website, or topics provided.

Activity

0 stars 0 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

HTML 100.0%

## Creating a Dockerfile in master branch;

https://github.com/SharathBedra/website/new/master

YouTube WhatsApp Adobe Acrobat Photo - Google Photo... All Bookmarks

SharathBedra / website

Code Pull requests Actions Projects Wiki Security Insights Settings

website / Dockerfile in master

Edit Preview Code 55% faster with GitHub Copilot

```
1 FROM ubuntu/apache2
2 COPY . /var/www/html
```

Commit changes

Commit message

Create Dockerfile

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

Cancel Commit changes

SharathBedra / website

Code Pull requests Actions Projects Wiki Security Insights Settings

master / website /

SharathBedra Create Dockerfile

This branch is 1 commit ahead of hshar/website:master .

Name Last commit message Last commit date

images	final	5 years ago
Dockerfile	Create Dockerfile	4 minutes ago
index.html	modified	5 years ago

## Create a Pipeline to Automate the Tasks;

Jenkins

Search (CTRL+K)

Sharrath Bedra log out

Dashboard > All > New Item

New Item

Enter an item name

Testpipeline

Select an item type

**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.

**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.

**Multi-configuration project**

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

**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

## Configuring the pipeline script;

Dashboard > Testpipeline > Configuration

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

Configure

General Advanced Project Options Advanced Pipeline

Pipeline

Definition

Pipeline script

Script ?

```

1> pipeline {
2>   agent none
3>   environment {
4>     DOCKERHUB_CREDENTIALS=credentials("d4034313-614a-45b8-a0b6-ea6ced0bdf43")
5>   }
6>   stages {
7>     stage('Hello') {
8>       steps {
9>         echo 'Hello World'
10>      }
11>    }
12>  }
13>
14>

```

try sample Pipeline... ▾

Save Apply

```

Started by user Sharath Bedra
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  (Hello)
[Pipeline] echo
Hello World
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Now, we will use the below-given script to check whether the pipeline script is working properly or not.

Dashboard > Testpipeline > Configuration

## Configure

General

Advanced Project Options

Pipeline

### Pipeline

#### Definition

Pipeline script

```

1 pipeline {
2   agent none
3   environment {
4     DOCKERHUB_CREDENTIALS=credentials('d4034313-614a-45b8-a0b6-ea6ced0bdf43')
5   }
6   stages {
7     stage('Hello') {
8       steps {
9         echo 'Hello World'
10      }
11    }
12    stage('Git') {
13      agent {
14        label 'Kubernetes-Master'
15      }
16      steps {
17        git 'https://github.com/SharathBedra/website.git'
18      }

```

Use Groovy Sandbox ?

Pipeline Syntax

**Save**

Apply

Dashboard > Testpipeline > #4

## Console Output

Download Copy View as plain text

```

Started by user Sharath Bedra
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  (Hello)
[Pipeline] echo
Hello World
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] stage
[Pipeline] {
  (Git)
[Pipeline] node
Running on Kubernetes Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/SharathBedra/website.git
> git init /home/ubuntu/jenkins/workspace/Testpipeline # timeout=10
Fetching upstream changes from https://github.com/SharathBedra/website.git
> git --version # timeout=10
> git --version # git version 2.43.0
> git fetch --tags --force --progress -- https://github.com/SharathBedra/website.git +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 1bcbb05136fb6cc53d54595f61564063185c55 (refs/remotes/origin/master)
> git config remote.origin.url https://github.com/SharathBedra/website.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git config core.sparsecheckout # timeout=10
> git checkout -f 1bcbb05136fb6cc53d54595f61564063185c55 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b master 1bcbb05136fb6cc53d54595f61564063185c55 # timeout=10
Commit message: "Create Dockerfile"
First time build. Skipping changelog.
[Pipeline]
[Pipeline] // node
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Again, paste the below-given script in the “Pipeline” section. Click on “Save”.

```
*Untitled - Notepad
File Edit Format View Help
pipeline {
    agent none
    environment {
        DOCKERHUB_CREDENTIALS=credentials('d4034313-614a-45b8-a0b6-ea6ced0bdf43')
    }
    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
        stage('Git') {
            agent {
                label 'Kubernetes Master'
            }
            steps {
                git 'https://github.com/SharathBedra/website.git'
            }
        }
        stage('Docker') {
            agent {
                label 'Kubernetes Master'
            }
            steps {
                sh 'sudo docker build /home/ubuntu/jenkins/workspace/Testpipeline -t sharathbedra/project2'
                sh 'sudo echo $DOCKERHUB_CREDENTIALS_PSW | sudo docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
                sh 'sudo docker push sharathbedra/project2'
            }
        }
    }
}
```

The screenshot shows the Jenkins Pipeline configuration interface. At the top, there are tabs for 'Instances | EC2 | us-east-1' and 'EC2 Instance Connect'. Below the tabs, the URL is 3.129.11.240:8080/job/Testpipeline/configure. The main area has a breadcrumb navigation: Dashboard > Testpipeline > Configuration. On the left, there are three tabs: 'General' (selected), 'Advanced Project Options', and 'Pipeline'. The 'Pipeline' tab is currently active. It contains a 'Definition' section with a dropdown menu set to 'Pipeline script'. Below it is a code editor titled 'Script' with the Jenkins pipeline script pasted in. A 'try sample Pipeline...' button is located to the right of the code editor. At the bottom of the editor are two buttons: 'Save' and 'Apply'.

The screenshot shows the Jenkins Pipeline console output for build #5. The left sidebar contains links: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#5', Timings, Git Build Data, Pipeline Overview, Pipeline Console, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main area is titled 'Console Output' and shows the following log:

```
Started by user Sharath Bedra
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Git)
[Pipeline] node
Running on Kubernetes Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision 1bcbb6b5136fb6cc53d8545955f61564063185c55 (refs/remotes/origin/master)
Commit message: "Create Dockerfile"
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Testpipeline/.git # timeout=10
> git config remote.origin.url https://github.com/SharathBedra/website.git # timeout=10
Fetching upstream changes from https://github.com/SharathBedra/website.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
```

Dashboard > Testpipeline > #5

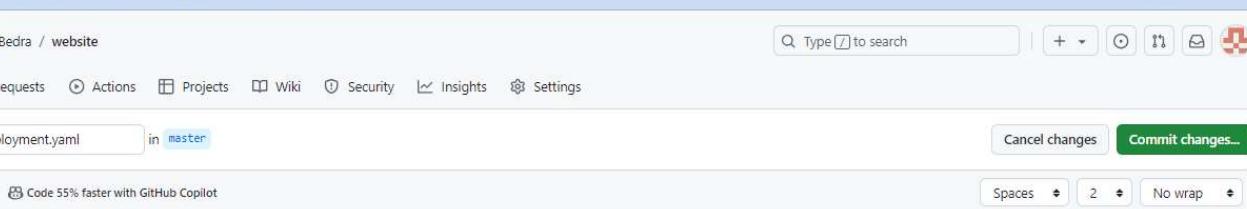
```
WARNING: Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] sh
+ sudo docker push sharathbedra/project2
Using default tag: latest
The push refers to repository [docker.io/sharathbedra/project2]
90ac36489388: Preparing
62978bc9d4e5: Preparing
068ff270fbe5: Preparing
f36fd4bb7334: Preparing
068ff270fbe5: Mounted from ubuntu/apache2
62978bc9d4e5: Mounted from ubuntu/apache2
f36fd4bb7334: Mounted from ubuntu/apache2
90ac36489388: Pushed
latest: digest: sha256:ce4594b1f95ad39b2650d766e0b946e30c4e43d6df018912e96f057ca
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The screenshot shows a web browser window with the following details:

- Address Bar:** https://hub.docker.com/repositories/sharathbedra
- Toolbar:** YouTube, WhatsApp, Adobe Acrobat, Photo - Google Pho..., All Bookmarks
- Docker Hub Header:** dockerhub, Explore, **Repositories**, Organizations, Usage, Search Docker Hub (ctrl+K), Help, Settings, Notifications, Profile (S)
- Search and Filter:** A dropdown menu shows "sharathbedra" with a dropdown arrow, a search bar "Search by repository name", a dropdown menu "All Content" with a dropdown arrow, and a blue "Create repository" button.
- Repository Card:** sharathbedra / project2. It includes a star icon (0), a down arrow icon (0), a Public icon, and a Scout inactive icon. Below the card, it says "Contains: Image" and "Last pushed: 1 minute ago".

Now, we will create the “deployment.yaml” & “service.yaml” file to deploy the website using the “Kubernetes” tool.



The screenshot shows the GitHub interface for a repository named 'SharathBedra/website'. The user is viewing the 'deployment.yaml' file in the 'master' branch. The file contains a Kubernetes Deployment configuration for an Nginx application. The code is as follows:

```
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: sharathbedra/project2:latest
          ports:
            - containerPort: 80
```

At the bottom of the code editor, there is a note: "Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page."

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: my-service
5  spec:
6    type: NodePort
7    selector:
8      app: nginx
9    ports:
10   - port: 80
11     targetPort: 80
12     nodePort: 30000

```

Name	Last commit message	Last commit date
images	final	5 years ago
Dockerfile	Create Dockerfile	39 minutes ago
deployment.yaml	Create deployment.yaml	now
index.html	modified	5 years ago
service.yaml	Create service.yaml	now

Go to the “Configure” in the “Jenkins” & paste the below-given code to create the Kubernetes Deployment for deploying the application and click on build now:

```

pipeline {
    agent none
    environment {
        DOCKERHUB_CREDENTIALS=credentials('d4034313-614a-45b8-a0b6-ea6ced0bdf43')
    }
    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
        stage('Git') {
            agent {
                label 'Kubernetes Master'
            }
            steps {
                git 'https://github.com/SharathBedra/website.git'
            }
        }
        stage('Docker') {
            agent {
                label 'Kubernetes Master'
            }
            steps {
                sh 'sudo docker build /home/ubuntu/jenkins/workspace/Testpipeline -t sharathbedra/project2'
                sh 'sudo echo $DOCKERHUB_CREDENTIALS_PSW | sudo docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
                sh 'sudo docker push sharathbedra/project2'
            }
        }
        stage('K8s') {
            agent {
                label 'Kubernetes Master'
            }
            steps {
                sh 'kubectl apply -f deployment.yaml'
                sh 'kubectl apply -f service.yaml'
            }
        }
    }
}

```

Instances | EC2 | us-east-1 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Testpipeline Config

Not secure 3.129.11.240:8080/job/Testpipeline/configure

Dashboard > Testpipeline > Configuration

## Configure

**Pipeline**

- General
- Advanced Project Options**
- Pipeline

**Definition**

Pipeline script

```

Script ? try sample Pipeline...
24  sh 'sudo docker build /home/ubuntu/jenkins/workspace/Testpipeline -t sharathbedra/p...
25  sh 'sudo echo $DOCKERHUB_CREDENTIALS_PSW | sudo docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
26  sh 'sudo docker push sharathbedra/project2'
27
28
29 }
stage('K8s') {
30   agent {
31     label 'Kubernetes-Master'
32   }
33   steps {
34     sh 'kubectl apply -f deployment.yaml'
35     sh 'kubectl apply -f service.yaml'
36   }
37 }
38
39 }
40

```

Use Groovy Sandbox ?

**Save** **Apply**

Instances | EC2 | us-east-1 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Testpipeline #7 Configuration

Not secure 3.129.11.240:8080/job/Testpipeline/7/console

Jenkins

Search (CTRL+K)

Sharath Bedra

log out

Dashboard > Testpipeline > #7

**Console Output**

Status Changes Console Output Edit Build Information Delete build #7 Timings Git Build Data Pipeline Overview Pipeline Console Restart from Stage Replay Pipeline Steps

```

Started by user Sharath Bedra
[Pipeline] Start of Pipeline
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Git)
[Pipeline] node
Running on Kubernetes Master in /home/ubuntu/jenkins/workspace/Testpipeline
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified

```

Paste both the Slaves' IP one by one in the browser address bar & the website will be successfully deployed through "Kubernetes"(IP-Address:30008).

Instances | EC2 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Testpipeline #7 | Intellipaat

Not secure 52.14.142.163:30008

Hello world!



# GitHub



Automate the Pipeline using Github Webhooks.

Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at <https://docs.github.com/webhooks/ping-event>.

Do the Changes & Test the Pipeline also choose “GitHub hook trigger for GITScm polling” in “Build Triggers”.

Here add `sh 'kubectl delete deploy nginx-deployment'`

GitHub project

Pipeline speed/durability override ?

Preserve stashes from completed builds ?

This project is parameterized ?

Throttle builds ?

**Build Triggers**

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Quiet period ?

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

Instances | EC2 Instance | EC2 Instance | EC2 Instance | EC2 Instance | Testpipeline | Intellipaat | Intellipaat

Not secure 3.129.11.240:8080/job/Testpipeline/configure

Dashboard > Testpipeline > Configuration

## Configure Pipeline

**Definition**

**Pipeline script**

```

25   sh 'sudo docker build /home/ubuntu/jenkins/workspace/Testpipeline -t sharathbedra/project2'
26   sh 'sudo echo $DOCKERHUB_CREDENTIALS_PSW | sudo docker login -u $DOCKERHUB_CREDENTIALS_USR'
27   sh 'sudo docker push sharathbedra/project2'
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 stage('K8s') {
36   agent {
37     label 'Kubernetes Master'
38   }
39   steps {
40     sh 'kubectl delete deploy nginx-deployment'
41     sh 'kubectl apply -f deployment.yaml'
42     sh 'kubectl apply -f service.yaml'
43   }
44 }
45 }

```

**Save** **Apply**

Change “Title” to “DevOps Capstone Project 2”.

https://github.com/SharathBedra/website/edit/master/index.html

YouTube WhatsApp Adobe Acrobat Photo - Google Photo All Bookmarks

SharathBedra / website Type to search

Code Pull requests Actions Projects Wiki Security Insights Settings

Files master index.html in master Cancel changes Commit changes...

Edit Preview Code 55% faster with GitHub Copilot Spaces 2 No wrap

```

1 <html>
2 <head>
3 <title> DevOps Capstone Project 2 </title>
4 </head>
5 <body style = "background-image:url('images/github3.jpg'); background-size: 100%">
6 <h2 ALIGN=CENTER>Hello world!</h2>
7 </body>
8 </html>
9

```

Instances | EC2 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Testpipeline #9 Console

Not secure 3.129.11.240:8080/job/Testpipeline/9/console

Jenkins Search (CTRL+K) Sharath Bedra log out

Dashboard > Testpipeline > #9

Status Changes Console Output Edit Build Information Delete build '#9' Polling Log Timings Git Build Data Pipeline Overview Pipeline Console Restart from Stage

Console Output

Started by Github push by SharathBedra  
[Pipeline] Start of Pipeline  
[Pipeline] withCredentials  
Masking supported pattern matches of \$DOCKERHUB\_CREDENTIALS or \$DOCKERHUB\_CREDENTIALS\_PSW  
[Pipeline] {  
[Pipeline] stage  
[Pipeline] { (Hello)  
[Pipeline] echo  
Hello World  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Git)  
[Pipeline] node  
Running on Kubernetes Master in /home/ubuntu/jenkins/workspace/Testpipeline  
[Pipeline] }

Verifying;

