

## DevOps Project 2

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 implement a DevOps life cycle, 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 life-cycle:

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

### Architectural Advice

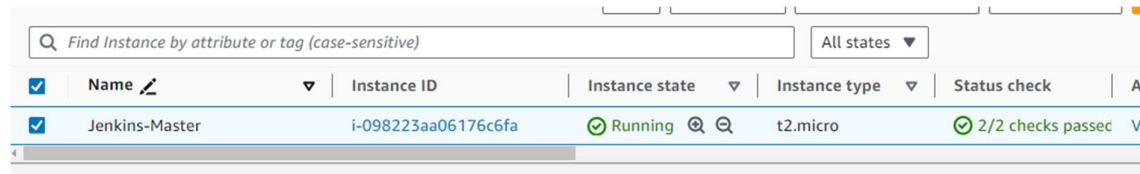
Software's to be installed on the respective machines using configuration management.

**Worker1:** Jenkins, Java. **Worker2:** Docker, Kubernetes.

**Worker3:** Java, Docker, Kubernetes **Worker4:** Docker, Kubernetes.

Solution:

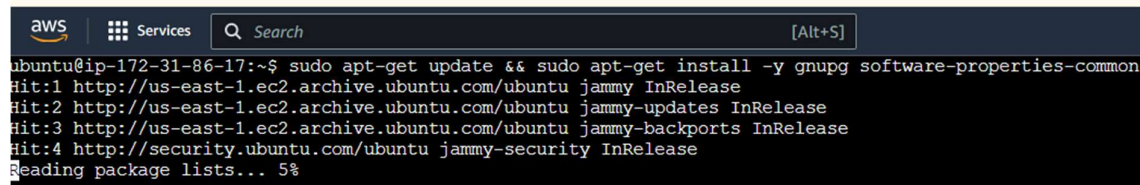
Jenkins Master launched



The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there is a search bar with the text "Find Instance by attribute or tag (case-sensitive)" and a dropdown menu set to "All states". Below this is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, and a final column with a checkmark icon. The table contains one entry: "Jenkins-Master" with Instance ID "i-098223aa06176c6fa", Instance state "Running" (indicated by a green checkmark icon), Instance type "t2.micro", and Status check "2/2 checks passed" (indicated by a green checkmark icon).

Name	Instance ID	Instance state	Instance type	Status check	
Jenkins-Master	i-098223aa06176c6fa	Running	t2.micro	2/2 checks passed	

Installing terraform



The screenshot shows a terminal window with the following commands and output:

```
ubuntu@ip-172-31-86-17:~$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... 5%
```

Commands used to install Terraform:

```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
```

```
wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee
/usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
```

```
gpg --no-default-keyring --keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg --
fingerprint
```

```
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
```

main.tf file created

```

ubuntu@ip-172-31-86-17:~$ cat main.tf
provider "aws" {
  region = "us-east-1"
  access_key = "AKIAQ3EGVQL5LD52DTEJ"
  secret_key = "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"
}

resource "aws_instance" "K-M" {
  ami = "ami-0a0e5d9c7acc336f1"
  instance_type = "t2.medium"
  key_name = "new{privatekeyNV"
  tags = {
    Name = "K-M"
  }
}

resource "aws_instance" "K-S1" {
  ami = "ami-0a0e5d9c7acc336f1"
  instance_type = "t2.micro"
  key_name = "new{privatekeyNV"
  tags = {
    Name = "K-S1"
  }
}

resource "aws_instance" "K-S2" {
  ami = "ami-0a0e5d9c7acc336f1"
  instance_type = "t2.micro"
  key_name = "new{privatekeyNV"
  tags = {
    Name = "K-S2"
  }
}

```

Terraform init

```

ubuntu@ip-172-31-86-17:~$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.62.0...
- Installed hashicorp/aws v5.62.0 (signed by Hashi
Terraform has created a lock file .terraform.lock
selections it made above. Include this file in you

```

Terraform plan

```

commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-86-17:~$ terraform plan

Terraform used the selected providers to generate the following execution plan:
+ create

Terraform will perform the following actions:

# aws_instance.K-M will be created
+ resource "aws_instance" "K-M" {

```

Terraform apply

```

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

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

Enter a value: yes

aws_instance.K-M: Creating...
aws_instance.K-S2: Creating...
aws_instance.K-S1: Creating...

aws_instance.K-S2: Still creating... [20s elapsed]
aws_instance.K-S1: Still creating... [20s elapsed]
aws_instance.K-M: Creation complete after 22s [id=i-0ce3b919a9d4c61d1]
aws_instance.K-S2: Still creating... [30s elapsed]
aws_instance.K-S1: Still creating... [30s elapsed]
aws_instance.K-S1: Creation complete after 32s [id=i-049a68947d9ac7b17]
aws_instance.K-S2: Creation complete after 32s [id=i-073f2474f87028bfd]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-86-17:~$

```

Instances (4) Info

🔄













Connect

Instance state ▼

Actions ▼

🔍 Find Instance by attribute or tag (case-sensitive)

All states ▼

<input type="checkbox"/>	Name 	Instance ID	Instance state ▼	Instance type ▼	Status check
<input type="checkbox"/>	K-MandJ-S	i-0ce3b919a9d4c61d1	<span>Running</span>  	t2.medium	 Initializing
<input type="checkbox"/>	K-S2	i-073f2474f87028bfd	<span>Running</span>  	t2.micro	 Initializing
<input type="checkbox"/>	Jenkins-Master	i-098223aa06176c6fa	<span>Running</span>  	t2.micro	<span>2/2 checks passed</span>
<input type="checkbox"/>	K-S1	i-049a68947d9ac7b17	<span>Running</span>  	t2.micro	 Initializing

Installing ansible

```
aws | Services | Search
ubuntu@ip-172-31-86-17:~$ sudo nano a.sh
ubuntu@ip-172-31-86-17:~$ cat a.sh
sudo apt update
sudo apt install software-properties-common
sudo apt-add-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
ubuntu@ip-172-31-86-17:~$
```

Ansible is installed

```
aws | Services | Search
ubuntu@ip-172-31-86-17:~$ ansible --version
ansible [core 2.16.9]
  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/modules/core
  ansible collection location = /home/ubuntu/.ansible/collect
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
  jinja version = 3.0.3
  libyaml = True
```

Creating keys in J-Master

```
| =.. |
+----[SHA256]-----+
ubuntu@Jenkins-Master:~$ cd .ssh
ubuntu@Jenkins-Master:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@Jenkins-Master:~/.ssh$
```

i-098223aa06176c6fa (Jenkins-Master)

PublicIPs: 54.165.168.27 PrivateIPs: 172.31.86.17

```
aws | Services | Search [Alt+S]
ubuntu@Jenkins-Master:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@Jenkins-Master:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCssvK/9G8FU3Rv7O1jiHIPgc4oOCDBElxwJfhzVTgtgi
4f2z8eXUHE/tbOjxN5xtpyt1VRdSae44wzwyUDYabwfx0zUzYAfMY/9516492RIXRPNlekYe0DfwbYwpse
leXos5DUM5RJ4P37O4vWP4U440d11WzOdBZme2RDc0bCwVsdMD0WDlywjVORH8/W9bSbXmHmGJpXfdkug+
Z0w2gmWjArqv9LXJOVo/RzxFFOH2XPb3bukTi3n3c+RUDGgwOYRUklgcxNtTrFgHvY7lhVvixNra7KXvnl
ubuntu@Jenkins-Master:~/.ssh$
```

Entering keys in K-S1, K-S2 and K-M



```
aws Services Search [Alt+S]
ubuntu@K-M:~$ sudo nano .ssh/authorized_keys
ubuntu@K-M:~$ cd .ssh
ubuntu@K-M:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCPwvZgfKfHSw1raoGxmQAnkF3dafep+aAh7UeEY
zuqBe08iogWN5S1fkX+0yjryVXRQ31W/c+f9gjztW21LKdHxJ4vTdaKBDrlJyWb1tOeusMdpe9Lf
Q1jarxA88LEA9aKEz6ZvPxx7cmY3Ycf82LYXnikCp8ZHE17xAwk6DXb/H8eKouTgi8M0z new{pr
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCssvK/9G8FU3Rv701jiHIPgc4oOCDBELxwJfhzV
2z8eXUHE/tbOjxN5xtpyt1VRdSae44wzWYuDYabwfx0zUzYAfMY/9516492RIXRPN1ekYe0DfwbYw
s5DUM5RJ4P3704vWP4U440d11WzOdBZme2RDc0bCwVsdMD0WDlywjVORH8/W9bSbXmHmGJpXfdkug
WjArqv9LXJOVo/RzxFFOH2XPb3bukTi3n3c+RUdGgwOYRUklgcxNtTrFgHvY7lhVvixNra7KXvnlW
ubuntu@K-M:~/.ssh$
```

```
ubuntu@K-S1:~$ sudo nano .ssh/authorized_keys
ubuntu@K-S1:~$ cd .ssh
ubuntu@K-S1:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCPwvZgfKfHSw1raoGxmQAnkF3dafep+aAh7UeEYQtKTCCTNkjaxGZo
zuqBe08iogWN5S1fkX+0yjryVXRQ31W/c+f9gjztW21LKdHxJ4vTdaKBDrlJyWb1tOeusMdpe9Lf7zsqxPJsbgsluU1
Q1jarxA88LEA9aKEz6ZvPxx7cmY3Ycf82LYXnikCp8ZHE17xAwk6DXb/H8eKouTgi8M0z new{privatekeyNV
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCssvK/9G8FU3Rv701jiHIPgc4oOCDBELxwJfhzVTgtgiPGIwQON12O
2z8eXUHE/tbOjxN5xtpyt1VRdSae44wzWYuDYabwfx0zUzYAfMY/9516492RIXRPN1ekYe0DfwbYwpseWRVGP3fDoHs+
s5DUM5RJ4P3704vWP4U440d11WzOdBZme2RDc0bCwVsdMD0WDlywjVORH8/W9bSbXmHmGJpXfdkug+GxEmsy0BbyZb/q
WjArqv9LXJOVo/RzxFFOH2XPb3bukTi3n3c+RUdGgwOYRUklgcxNtTrFgHvY7lhVvixNra7KXvnlWQfQ49/8ic= ubun
ubuntu@K-S1:~/.ssh$
```

```
ubuntu@K-S2:~$ sudo nano .ssh/authorized_keys
ubuntu@K-S2:~$ cd .ssh
ubuntu@K-S2:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCPwvZgfKfHSw1raoGxmQAnkF3dafep+aAh7UeEYQtKTCCTNkjaxGZot+aFt
fBzuqBe08iogWN5S1fkX+0yjryVXRQ31W/c+f9gjztW21LKdHxJ4vTdaKBDrlJyWb1tOeusMdpe9Lf7zsqxPJsbgsluU1L4F
C+tnQ1jarxA88LEA9aKEz6ZvPxx7cmY3Ycf82LYXnikCp8ZHE17xAwk6DXb/H8eKouTgi8M0z new{privatekeyNV
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCssvK/9G8FU3Rv701jiHIPgc4oOCDBELxwJfhzVTgtgiPGIwQON12OEC1Y
4f2z8eXUHE/tbOjxN5xtpyt1VRdSae44wzWYuDYabwfx0zUzYAfMY/9516492RIXRPN1ekYe0DfwbYwpseWRVGP3fDoHs+4eY
leXos5DUM5RJ4P3704vWP4U440d11WzOdBZme2RDc0bCwVsdMD0WDlywjVORH8/W9bSbXmHmGJpXfdkug+GxEmsy0BbyZb/q
Z0w2gmWjArqv9LXJOVo/RzxFFOH2XPb3bukTi3n3c+RUdGgwOYRUklgcxNtTrFgHvY7lhVvixNra7KXvnlWQfQ49/8ic= ubu
ubuntu@K-S2:~/.ssh$
```

Adding hosts

```
ubuntu@Jenkins-Master:/etc/ansible$ sudo nano hosts
ubuntu@Jenkins-Master:/etc/ansible$ cat hosts
[K-M]
172.31.27.137
[slaves]
172.31.90.157
172.31.93.250
ubuntu@Jenkins-Master:/etc/ansible$
```

All slaves are connected

```

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
172.31.27.137 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
172.31.90.157 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
172.31.93.250 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@Jenkins-Master:/etc/ansible$

```

Play.yaml is created

```

ubuntu@Jenkins-Master:/etc/ansible$ cat play.yaml
---
- name: install on localhost
  become: true
  hosts: localhost
  tasks:
    - name: running script to install java, jenkins and docker on localhost
      script: localhost.sh
- name: install on K-M
  become: true
  hosts: K-M
  tasks:
    - name: running script to install java, docker and K8s on K-M
      script: km.sh
- name: install on K-slaves
  become: true
  hosts: slaves
  tasks:
    - name: running script to install docker and K8s on K-slaves
      script: ks.sh
ubuntu@Jenkins-Master:/etc/ansible$

```

i-098223aa06176c6fa (Jenkins-Master)

Run dry check

```

TASK [running script to install docker and K8s on K-slaves] *****
skipping: [172.31.90.157]
skipping: [172.31.93.250]

PLAY RECAP *****
172.31.27.137      : ok=1    changed=0    unreachable=0    failed=0
172.31.90.157     : ok=1    changed=0    unreachable=0    failed=0
172.31.93.250     : ok=1    changed=0    unreachable=0    failed=0
localhost         : ok=1    changed=0    unreachable=0    failed=0

ubuntu@Jenkins-Master:/etc/ansible$

```

i-098223aa06176c6fa (Jenkins-Master)

PublicIPs: 54.165.168.27 PrivateIPs: 172.31.86.17

## K-M execution

```

ubuntu@ip-172-31-18-69:~$ sudo kubeadm config images pull
I0813 10:30:04.277002 8104 version.go:256] remote version is much newer:
[config/images] Pulled registry.k8s.io/kube-apiserver:v1.29.7
[config/images] Pulled registry.k8s.io/kube-controller-manager:v1.29.7
[config/images] Pulled registry.k8s.io/kube-scheduler:v1.29.7
[config/images] Pulled registry.k8s.io/kube-proxy:v1.29.7

```

```

PLAY RECAP *****
172.31.27.137      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.90.157     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.93.250     : 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

ubuntu@Jenkins-Master:/etc/ansible$

```

i-098223aa06176c6fa (Jenkins-Master)

PublicIPs: 54.165.168.27 PrivateIPs: 172.31.86.17

## Creating containers

```

kubeadm join 172.31.18.69:6443 --token xdurl1.vk6kr3xu72lbp6p9 \
--discovery-token-ca-cert-hash sha256:40ed8a9df2fb36bb88d4b51d7332fed0b041d988f717d7443f264
ubuntu@ip-172-31-18-69:~$ mkdir -p "$HOME"/.kube
sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config
sudo chown "$(id -u)": "$(id -g)" "$HOME"/.kube/config
ubuntu@ip-172-31-18-69:~$ kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v
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

```

Nodes can be seen from K-M



```
aws | Services | Search
ubuntu@ip-172-31-18-69:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-172-31-18-69                     Ready    control-plane   6m1s   v1.29.0
ip-172-31-83-111                    Ready    <none>         117s   v1.29.0
ip-172-31-90-156                    Ready    <none>         99s    v1.29.0
ubuntu@ip-172-31-18-69:~$
```

Logged in Jenkins

Project1 / service.yaml in master

Edit Preview Code 55% faster with GitHub Copilot

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: my-service
5  spec:
6    selector:
7      app.kubernetes.io/name: new
8    tupe: NodePort
9    ports:
10     - protocol: TCP
11       port: 80
12       targetPort: 30008
```


Code Blame 21 lines (21 loc) · 339 Bytes Code 55% faster with GitHub Copilot

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: my-deployment
5    labels:
6      app: new
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: new
12   template:
13     metadata:
14       labels:
15         app: new
16     spec:
17       containers:
18         - name: my-container
19           image: nginx:1.14.2
20           ports:
21             - containerPort: 80
```


52c7401d-00ad-4f27-ae76-71f87b9d9415

job

Select an item 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.


 Pipeline

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


Hello world script is working


Dashboard > new-job > #1

 Status

 Changes

 Console Output

 Edit Build Information

 Delete build '#1'

 Timings

 Pipeline Overview

 Pipeline Console

## Console Output


```
Started by user Minal
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Slave in /home/ubuntu/jenkins/workspace/new-job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello) (hide)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
```


Dashboard > job > #8

 Status

 Changes

 Console Output

 Edit Build Information

 Delete build '#8'

 Timings

## Console Output

```
Started by user Minal
[Pipeline] Start of Pipeline
[Pipeline] End of Pipeline
ERROR: f4b3a7f9-abbb-422a-8a00-308d1a3d0e9a
Finished: FAILURE
```

Kubernetes cluster not working