

Project A

CI/CD Pipeline to Deploy A Simple Web Application to Kubernetes Cluster

Flow Diagram:-

1. CI Job:- Any change in github repo will trigger CI job which will pull the code from github repo and jenkins will trigger the playbook on Ansible. Ansible playbook will build docker image, It will push the docker image to the docker hub.
 2. CD Job:- CD job will run Ansible playbook and it will pull images from docker hub and use this image to deploy pods on kubernetes cluster.
-

Steps done:-

1. I created an Azure Resource Group MyLab-RG and Azure virtual machine for jenkins server. (Install and Configure jenkins).
 2. I used MobaXterm -> remote toolbox for remote computing.
 3. Connect to Jenkins server through SSH -> enter public and private key.
 4. Install default jdk.
 5. Command to see where app installed java on ubuntu
update-alternatives --config java.
 6. I enter sudo nano /etc/environment -> to add the path of Java -> every user would be able to access the JAVA.
 7. source /etc/environment this command will force ubuntu terminal to load the environment configuration file.
 8. echo \$JAVA_HOME.
 9. Install jenkins on server -> go to jenkins installation page and copy command under weekly release <https://www.jenkins.io/doc/book/installing/linux/> .
 10. sudo systemctl enable jenkins -> allow jenkins to boot.
 11. sudo systemctl start jenkins.
 12. systemctl status jenkins.
 13. Go to azure networking and add inbound port rules. On port 8080 as this is the port of jenkins. Which in the following example is <http://57.151.115.214:8080/login?from=%2F> -> jenkins IP.
 14. Open jenkins path using command
sudo nano /var/lib/jenkins/secrets/initialAdminPassword
- Note:- Simply put, inbound firewall rules protect the network from unwanted incoming traffic from the internet or other networks -- in particular, disallowed connections, malware and DoS attacks.
15. Create a new Azure Virtual machine and install on it Ansible Server.
 16. Add inbound rule 8080-8090
 17. Copy public ip and access it within a terminal.
 18. Add a user inside the ansible server -> sudo adduser ansibleadmin
 19. Add sudo rights to this user -> sudo usermod -aG sudo ansible -> sudo init 6 for rebooting
 20. Open another session with public key and ansibleadmin as username
 21. Type ssh-keygen to generate rsa key-pair for ansibleadmin

22. cd /home/ansibleadmin
23. cd .ssh
24. Ls
25. Generate public key and install ansible in this server
26. sudo apt-add-repository ppa:ansible/ansible
27. sudo apt update
28. sudo apt install ansible-core
29. Integrate Ansible with jenkins and install docker on Ansible Server.
30. Add plugin -> install plugin over ssh
31. Click system under system consideration add ansible server and click save
32. On ansible session go to
33. Cd /opt
34. Sudo mkdir docker
35. Make ansible admin owner of docker directory

sudo chown ansibleadmin:ansibleadmin docker

36. ll
37. cd docker
38. Give all the rights to ansible admin on docker

Sudo usermod -aG docker ansibleadmin

39. Reboot system -> sudo init 6
40. Create ansible playbook -> to create a docker image and push it to docker hub.
41. On ansible server check ip of this server

Ifconfig -> check for ip address

sudo apt install net-tools

42. Enter nano hosts
43. Ssh-copy-id 10.0.0.5 -> self ssh key so that ansible playbook can connect to localhost as
docker image will be created on ansible host itself
44. cd /opt/docker
45. nano cafe-app.yml
46. Write the following ansible-playbook

Task to clone repo

Task to create docker image

```

---
- hosts: ansible
  tasks:
    - name: clone repository
      git:
        repo: https://github.com/ashfaque-9x/Cafe-App.git
        dest: /opt/docker/cafe-app
        clone: yes
        update: yes
    - name: create docker image
      command: docker build -t cafe-app:latest /opt/docker/cafe-app
      args:

```

chdir: /opt/docker/

47. ansible-playbook cafe-app.yml --check

48. For running the playbook -> ansible-playbook cafe-app.yml

Note:- you can create a docker image from ansible playbook

49. Push docker image to your docker hub account.

50. Add two more tasks in Ansible playbook

Task to tag docker image

Task to push docker image

51. Save the updated ansible playbook and run it

- hosts: ansible

tasks:

- name: clone repository

git:

repo: <https://github.com/Ashfaque-9x/Cafe-App.git>

dest: /opt/docker/cafe-app

clone: yes

update: yes

- name: create docker image

command: docker build -t cafe-app:latest /opt/docker/cafe-app

args:

chdir: /opt/docker/

- name: create tag to push image onto dockerhub

command: docker build -t cafe-app:latest momousa1997/cafe-app:latest

- name: push docker image

command: docker push momousa1997/cafe-app:latest ->

Note:- make sure to write your dockerhub username instead of momousa1997

52. Ansible-playbook cafe-app.yml

53. Create Jenkins Job to run ansible playbook

- Click on git and select type your git repo and enter */main

- Select Post-Build Actions

- Type Ansible-Server under Source Files

- Type ansible-playbook /opt/docker/cafe-app.yml under Exec command

54. Click on Apply

55. Click on Save

56. Click on Build Now

57. After the job is completed type docker images to make sure that image is created with a tag.

58. Then also go to the docker hub and check that image is pushed.

59. Create Kubernetes Cluster

60. Create a virtual machine from where you will operate the kubernetes cluster.

61. Open session with kube-server VM

62. Type sudo su -> to go to root

63. Install azure-cli on linux -> refer to documentations and copy the command

curl -sL <https://aka.ms/InstallAzureCLIDeb> | sudo bash

64. From azure portal go to kubernetes services -> create kubernetes cluster

65. Make sure you are in the root@kube-server

66. Give command az login

A web browser has been opened at

<https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize>. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with `az login --use-device-code`.

az login --use device-code -> enter the code in the microsoft page

67. az aks install-cli -> install kubernetes cli

68. To connect to kubernetes cluster

Go to the azure portal on my-lab-cluster -> click on connect -> copy command -> download cluster credentials and paste it on terminal.

69. You will see " Merged "mylab-cluster" as current context in /root/.kube/config "

70. Type kubectl get nodes and you will find two nodes. Here you have successfully set up kubernetes cluster and get connected to it through kubernetes cli

71. Create a deployment

72. Create deployment file

73. nano cafe-app-deployment.yml

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: mylab-cafe-app
```

```
  labels:
```

```
    app: cafe-app
```

```
spec:
```

```
  replicas: 2
```

```
  selector:
```

```
    matchLabels:
```

```
      app: cafe-app
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: cafe-app
```

```
    spec:
```

```
      containers:
```

```
        - name: cafe-app
```

```
          image: momousa1997/cafe-app
```

```
          imagePullPolicy: Always
```

```
          ports:
```

```
            - containerPort: 80
```

```
  strategy:
```

```
    type: RollingUpdate
```

```
rollingUpdate:
  maxSurge: 1
  maxUnavailable: 1
74. Create service manifest file for kubernetes cluster
75. Nano cafe-app-service.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: mylab-service
  labels:
    app: cafe-app
spec:
  selector:
    app: cafe-app
  ports:
    - port: 80
      targetPort: 80
  type: LoadBalancer
```

- 76. Integrate with the Kube-Server with ansible server
- 77. Check for kube-server you will find it with private ip 10.0.0.6

Note if they are in the same vnet you can only provide private IP while they are in different vNets you must also add a public IP address.

- 78. Enable password based login for root user
- 79. nano /etc/ssh/ssh_config
- 80. Open file and look for

PasswordAuthentication yes

PermitRootLogin yes

Make sure you remove “#”

- 81. Enabled password authentication for kube-server
- 82. Now we will go back to ansible server and we will copy the ssh key of ansibleadmin user to the root user of the kube-server so playbook can interact with kubernetes. As we have connected kubernetes with root user of kube-server
- 83. Give command ssh-copy-id root@10.0.0.6 on ansible-server
- 84. ssh ‘root@10.0.0.6’ from ansible server you are connected to kube-server

Note: For connectivity between Jenkins server and Ansible Server we have used ansibleadmin, and for the connectivity between Ansible server and kube server we have used root.

We have connected to kubernetes cluster with username root

- 85. Create Ansible playbook for Deployment and Service Manifest files for deploying resources in the kubernetes cluster.
- 86. Nano kubernetes-deploy.yml

```
- hosts: kubernetes
  users: root
  tasks:
```

- name: deploy cafeapp on kubernetes
command: kubectl apply -f cafe-app-deployment.yml
- name: create service for cafeapp
command: kubectl apply -f cafe-app-service.yml
- name: update deployment with new pods if image updated in docker hub
command: kubectl rollout restart deployment.apps/mylab-cafe-app
- 87. Create Jenkins Deployment Job for kubernetes -> CD
- 88. Create CD_Job_CafeApp
- 89. Choose Post Build Actions
- 90. Name -> Ansible-Server, Source files -> , exec command -> ansible-playbook
/opt/docker/kubernetes-deploy.yml -> will deploy resources on kubernetes.
- 91. In the first CI job configure and click on post build build other project -> move to Projects
to build -> CD_Job_CafeApp.
- 92. Click on poll SCM -> * * * * *
- 93. Verify CI/CD Pipeline by Doing best commit
- 94. Git clone your repo
- 95. nano index.html
- 96. Make a change in for example price of Ice americano to 11.55
- 97. git add .
- 98. git commit -m "Change Ice Americano price"
- 99. git push origin main
- 100. You will see jobs triggered in Jenkins and when you open the website you will see
the changes.