In this project we're going to deploy the Reddit Clone Application on the Kubernetes Server Manually (Not a Pipeline Script).

• What is Reddit?

- Reddit is a social network with a forum-style discussion structure. Users create posts in topic-based communities — called subreddits — and interact in comment threads. Every thread has an OP (original poster) who started it. Users can also vote content by others "up" or "down" the algorithm.

Steps We are Involving in this project:-

Step 1: Clone the source code

Step 2: Containerize the Application using Docker

Step 3) Building Docker-Image

Step 4) Push the Image To DockerHub

Step 5) Write a Kubernetes Manifest File

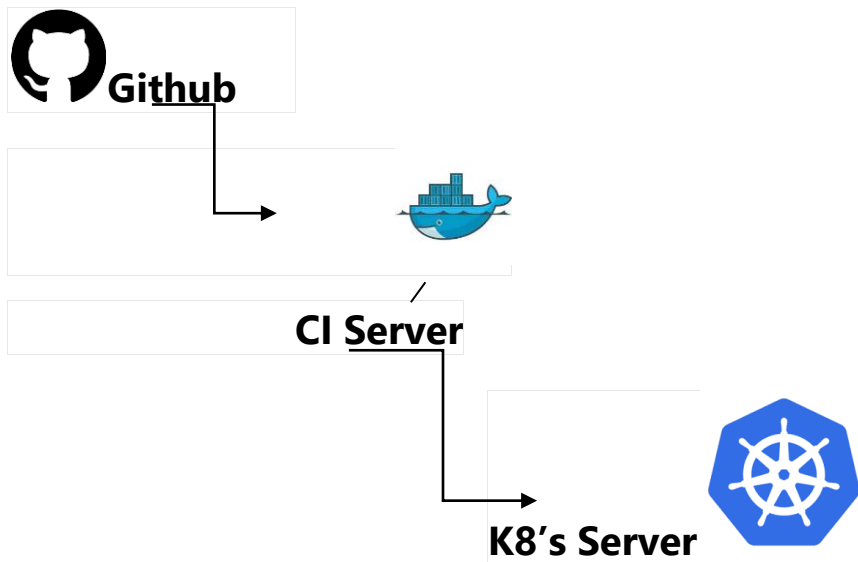Write Deployment.yml file

Write Service.yml file

Step 6) Deploy the app to Kubernetes & Create a Service For It

Step 7) Expose the app

Step 8) Let's Configure Ingress

Test Ingress

WorkFlow :-

Github

CI Server

K8's Server

- **Cloning the Code from Github to CI Server**
- **Building docker image in CI Server**
- **Pushing to the docker hub from CI Server**
- **Loging to K8's Server & deploying our app by pulling the image from the docker hub.**

Prerequisite: First You have to install some Prerequisites for this deployment:-

Before we begin with the Project, we need to make sure we have the following prerequisites installed:

EC2 (AMI- Ubuntu, Type- t2.micro) :–For CI Server

EC2 (AMI- Ubuntu, Type- t2.medium) :– For K8's Server

Docker

Minikube

Kubectl

You can Install all this by doing the below steps one by one. and these steps are for Ubuntu AMI.

☐	CI Server	i-02164730209c00e90	⊘ Running ⊕⊖	t2.micro	🕐 Initializing	No alarms	+
☐	K8s Server	i-0ba5cbc030176cb11	⊘ Running ⊕⊖	t2.medium	🕐 Initializing	No alarms	+

In Action:-

Connect to CI Server by SSH

```
Hussain@DESKTOP-572PBGQ MINGW64 ~/OneDrive/Desktop
$ ssh -i horizon.pem ubuntu@54.221.16.176
```

Once Connected, Install Docker in it.

For Docker Installation

sudo apt-get update

sudo apt-get install docker.io -y

sudo usermod -aG docker $USER && newgrp docker

Once Installed, Check our user is added in Docker Group or not.

Id $USER

```
ubuntu@ip-172-31-27-82:~$ id $USER
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdro
m),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),119(netdev),120(l
xd),122(docker)
ubuntu@ip-172-31-27-82:~$ |
```

Now, Connect to K8's Server and install docker and Minikube in it by running these commands :-

```
Hussain@DESKTOP-572PBGQ MINGW64 ~/OneDrive/Desktop
$ ssh -i horizon.pem ubuntu@3.80.80.15
The authenticity of host '3.80.80.15 (3.80.80.15)' can't be
ED25519 key fingerprint is SHA256:e5GnUkipCrZWyMk729wx8LpG/c
This key is not known by any other names
```

For Docker Installation

sudo apt-get update

sudo apt-get install docker.io -y

sudo usermod -aG docker $USER && newgrp docker

For Minikube & Kubectl

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube

sudo snap install kubectl --classic

minikube start --driver=docker

Great! You're all set for the project. Your Minikube cluster is now prepared for deploying the Reddit clone application. Now Lets Start...

Step 1: Clone the source code in CI Server:-

The first step is to clone the source code in the **CI Server** for the app. You can do this by using this command git clone https://github.com/Hussain147/reddit-clone-k8s.git

```
ubuntu@ip-172-31-27-82:~$ git clone https://github.com/Hussain147/reddit-clone-k8s.git
Cloning into 'reddit-clone-k8s'...
remote: Enumerating objects: 130, done.
remote: Counting objects: 100% (130/130), done.
remote: Compressing objects: 100% (115/115), done.
remote: Total 130 (delta 4), reused 130 (delta 4), pack-reused 0
Receiving objects: 100% (130/130), 1.47 MiB | 29.61 MiB/s, done.
Resolving deltas: 100% (4/4), done.
ubuntu@ip-172-31-27-82:~$ |
```

Now, do ls and change the directory

```
ubuntu@ip-172-31-27-82:~$ ls
reddit-clone-k8s
ubuntu@ip-172-31-27-82:~$ cd reddit-clone-k8s/
ubuntu@ip-172-31-27-82:~/reddit-clone-k8s$ ls
README.md  firebase.json  functions  next-env.d.ts  next.config.js  package-lock.json  package.json  public  src  tsconfig.json
ubuntu@ip-172-31-27-82:~/reddit-clone-k8s$ |
```

Now, Write a Dockerfile for all the files in it.

Vi Dockerfile

Follow Step 2

Step 2: Containerize the Application using Docker:-

Write a Dockerfile with the following code:

You can copy from the github link:- https://github.com/Hussain147/reddit-clone-k8s.git

#Taking base image as Node

FROM node:19-alpine3.15

#Docker will Create a Working Environment as reddit-clone

WORKDIR /reddit-clone

#Copy all the present working directory files into the environment of Reddit-clone

COPY . /reddit-clone

#Docker will start the process by npm install command

RUN npm install

#Exposing to 3000 Port

EXPOSE 3000

#Giving commands to start the application

CMD ["npm","run","dev"]

Step 3: Building Docker-Image:-

Now it's time to build Docker Image from this Dockerfile. docker build -t <DockerHub_Username>/<Imagename> . use this command to build a docker image.

docker build . –t kubehussain/reddit-clone

```
ubuntu@ip-172-31-27-82:~/reddit-clone-k8s$ docker build . -t kubehussain/reddit-clone
Sending build context to Docker daemon   4.21MB
Step 1/6 : FROM node:19-alpine3.15
19-alpine3.15: Pulling from library/node
9621f1afde84: Already exists
79ef350afea0: Already exists
d3ad581b5912: Already exists
ee1cc07adb60: Already exists
Digest: sha256:12d9c7253f232bb88a9ef6d6e974afd90e296cb8383572dbb7f28c39f828b07e
Status: Downloaded newer image for node:19-alpine3.15
 ---> e5f532dbeb15
Step 2/6 : WORKDIR /reddit-clone
 ---> Running in f2bc973b9bf6
Removing intermediate container f2bc973b9bf6
 ---> 114c38cb9f1d
Step 3/6 : COPY . /reddit-clone
 ---> 4761650dc2ff
Step 4/6 : RUN npm install -g
 ---> Running in f294323e80bf

added 1 package, and audited 3 packages in 473ms

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 8.19.3 -> 9.6.1
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v9.6.1>
npm notice Run `npm install -g npm@9.6.1` to update!
npm notice
Removing intermediate container f294323e80bf
 ---> 19b56c5d23fa
Step 5/6 : EXPOSE 3000
 ---> Running in 55b80b4f517e
Removing intermediate container 55b80b4f517e
 ---> 7f1f41a63a8d
Step 6/6 : CMD ["npm","run","dev"]
 ---> Running in 6c32dd5771ba
Removing intermediate container 6c32dd5771ba
 ---> 8d21fcc60f1e
Successfully built 8d21fcc60f1e
Successfully tagged kubehussain/reddit-clone:latest
```

Step 4: Push the Image To DockerHub:-

Now push this Docker Image to DockerHub so our Deployment file can pull this image & run the app in Kubernetes pods.

- First login to your DockerHub account using Command i.e docker login and give your username & password.

    docker login

- Then use docker push <DockerHub_Username>/<Imagename> for pushing to the DockerHub.

  docker push <username>/<imagename>:<tag>

  Example :- docker push kubehussain/reddit-clone:latest

- **You can use an existing docker image i.e kubehussain/reddit-clone:latest for deployment.**

See the below picture, our Docker image is stored in Docker Hub.

Step 5: Write a Kubernetes Manifest File:-

Now, You might be wondering what this manifest file in Kubernetes is. Don't worry, I'll tell you in brief.

When you are going to deploy to Kubernetes or create Kubernetes resources like a pod, replica-set, config map, secret, deployment, etc, you need to write a file called manifest that describes that object and its attributes either in yaml or json. Just like you do in the ansible playbook.

Of course, you can create those objects by using just the command line, but the recommended way is to write a file so that you can version control it and use it in a repeatable way.

Login to **K8s Server:-**

Create a directory to couple all the files in one directory

mkdir k8s

cd k8s

```
ubuntu@ip-172-31-18-126:~$ mkdir k8s
ubuntu@ip-172-31-18-126:~$ cd k8s
ubuntu@ip-172-31-18-126:~/k8s$ |
```

Validate the kubectl whether our minikube is installed or not.

```
ubuntu@ip-172-31-18-126:~/k8s$ kubectl get pods
No resources found in default namespace.
```

1. Write Deployment.yml file:-

Let's Create a Deployment File For our Application. Use the following code for the *Deployment.yml* file.

You can copy from here :- https://github.com/Hussain147/reddit-clone-k8s/blob/main/Deployment.yaml

vi Deployment.yaml

```
ubuntu@ip-172-31-18-126:~/k8s$ vi Deployment.yaml|
```

Paste the Script, Save & Run the Deployment.yaml file

Kubectl apply –f Deployment.yaml

```
ubuntu@ip-172-31-18-126:~/k8s$ kubectl apply -f Deployment.yaml|
```

```
ubuntu@ip-172-31-18-126:~/k8s$ kubectl get deployment
NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
reddit-clone-deployment   2/2     2            2           84s
ubuntu@ip-172-31-18-126:~/k8s$ kubectl get pods
NAME                                       READY   STATUS    RESTARTS   AGE
reddit-clone-deployment-7549854884-29qw2   1/1     Running   0          90s
reddit-clone-deployment-7549854884-6qhk8   1/1     Running   0          90s
ubuntu@ip-172-31-18-126:~/k8s$ |
```

apiVersion: apps/v1

kind: Deployment

metadata:

```yaml
  name: reddit-clone-deployment
  labels:
    app: reddit-clone
spec:
  replicas: 2
  selector:
    matchLabels:
      app: reddit-clone
  template:
    metadata:
      labels:
        app: reddit-clone
    spec:
      containers:
      - name: reddit-clone
        image: kubehussain/reddit-clone
        ports:
        - containerPort: 3000
```

----

vi Service.yaml

2. Write Service.yml file:-

You can copy from here:- https://github.com/Hussain147/reddit-clone-k8s/blob/main/Service.yaml

```
ubuntu@ip-172-31-18-126:~/k8s$ vi Service.yaml
ubuntu@ip-172-31-18-126:~/k8s$ kubectl apply -f Service.yaml
service/reddit-clone-service created
ubuntu@ip-172-31-18-126:~/k8s$ kubectl get service
NAME                   TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes             ClusterIP   10.96.0.1      <none>        443/TCP          73m
reddit-clone-service   NodePort    10.96.222.3    <none>        3000:31000/TCP   11s
ubuntu@ip-172-31-18-126:~/k8s$ |
```

apiVersion: v1

kind: Service

metadata:

  name: reddit-clone-service

  labels:

    app: reddit-clone

spec:

  type: NodePort

  ports:

  - port: 3000

    targetPort: 3000

    nodePort: 31000

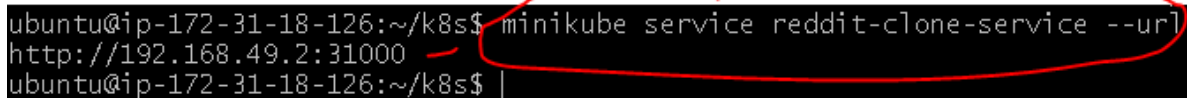 selector:

app: reddit-clone

Step 6: Deploy the app to Kubernetes & Create a Service For It :-

Now, we have a deployment file for our app and we have a running Kubernetes cluster, we can deploy the app to Kubernetes. To deploy the app you need to run following the command: kubectl apply -f Deployment.yml Just Like this create a Service using kubectl apply -f Service.yml

If You want to check your deployment & Service use the command kubectl get deployment & kubectl get services

Now, **Start the service**

      minikube service reddit-clone-service --url

```
ubuntu@ip-172-31-18-126:~/k8s$ minikube service reddit-clone-service --url
http://192.168.49.2:31000
ubuntu@ip-172-31-18-126:~/k8s$ |
```

Then run this command to validate our app running or not:-

    Curl –L http://<ip:31000>

```
ubuntu@ip-172-31-18-126:~/k8s$ minikube service reddit-clone-service  --url
http://192.168.49.2:31000
ubuntu@ip-172-31-18-126:~/k8s$ curl -L http://192.168.49.2:31000
<!DOCTYPE html><html><link rel="preconnect" href="https://fonts.googleapis.com"/><l
ink rel="preconnect" href="https://fonts.gstatic.com"/><link href="https://fonts.go
ogleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;1,300;1,
400&amp;display=swap" rel="stylesheet"/><head><style data-next-hide-fouc="true">bod
y{display:none}</style><noscript data-next-hide-fouc="true"><style>body{display:blo
ck}</style></noscript><meta charSet="utf-8"/><meta name="viewport" content="width=d
evice-width"/><meta name="next-head-count" content="2"/><noscript data-n-css=""></n
oscript><script defer="" nomodule="" src="/_next/static/chunks/polyfills.js?ts=1678
```

If you get like this, then our app running internally.

Now, We need to run this application externally(on the browser), for that we need to expose our deployment.

-

Step 7: Expose the app:-

1. First We need to expose our deployment so use kubectl expose deployment reddit-clone-deployment --type=NodePort command.

```
ubuntu@ip-172-31-18-126:~/k8s$ kubectl expose deployment reddit-clone-deployment
 --type=NodePort
service/reddit-clone-deployment exposed
```

2. Then We have to expose our app service kubectl port-forward svc/reddit-clone-service 3000:3000 --address 0.0.0.0 &

```
ubuntu@ip-172-31-18-126:~/k8s$  kubectl port-forward svc/reddit-clone-service 3000:3000 --address 0.0.0.0 &
[1] 11022
ubuntu@ip-172-31-18-126:~/k8s$ |
```

Optional:-

3. **You can test your deployment using curl - L http://192.168.49.2:31000. 192.168.49.2 is a minikube ip & Port 31000 is defined in Service.yml**

4. You can also see the deployed application on **Ec2_ip:3000**
5. *Note:- Make sure you open the 3000 port in a security group of your Ec2 Instance.*
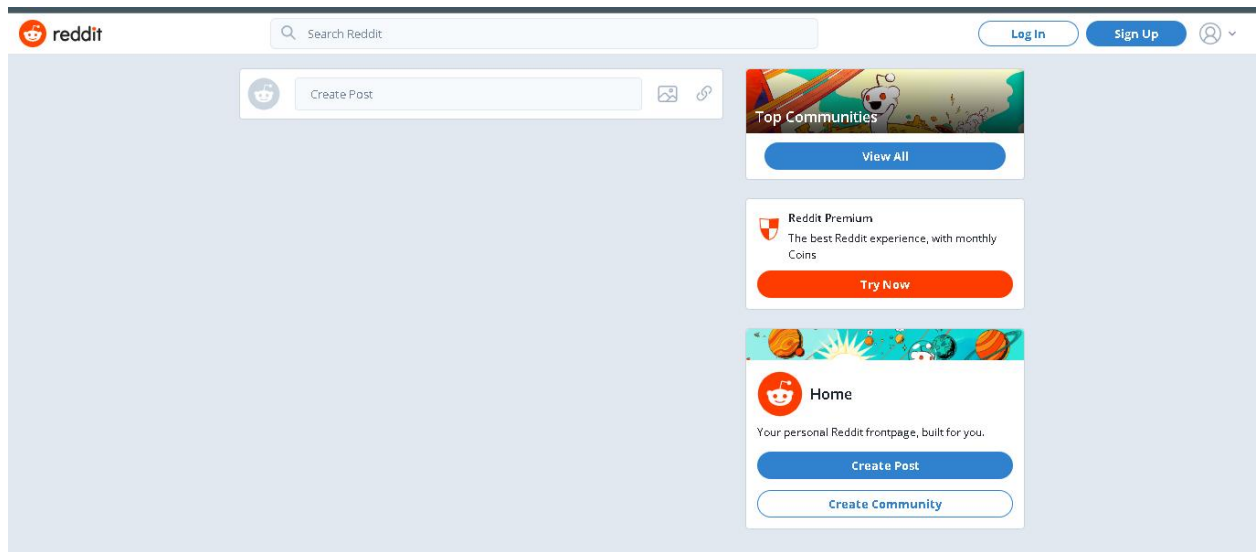
6. **You have successfully Deployed a Reddit Clone on Kubernetes.**

Step 8: Let's Configure Ingress:-

Ingress:

Pods and services in Kubernetes have their own IP; however, it is normally not the interface you'd provide to the external internet. Though there is a service with node IP configured, the port in the node IP can't be duplicated among the services. It is cumbersome to decide which port to manage with which service. Furthermore, the node comes and goes, it wouldn't be clever to provide a static node IP to an external service. Ingress defines a set of rules that allows the inbound connection to access Kubernetes cluster services. It brings the traffic into the cluster at L7 and allocates and forwards a port on each VM to the service port. This is shown in the following figure. We define a set of rules and post them as source type ingress to the API server. When the traffic comes in, the ingress controller will then fulfill and route the ingress by the ingress rules. As shown in the following figure, ingress is used to route external traffic to the Kubernetes endpoints by different URLs:
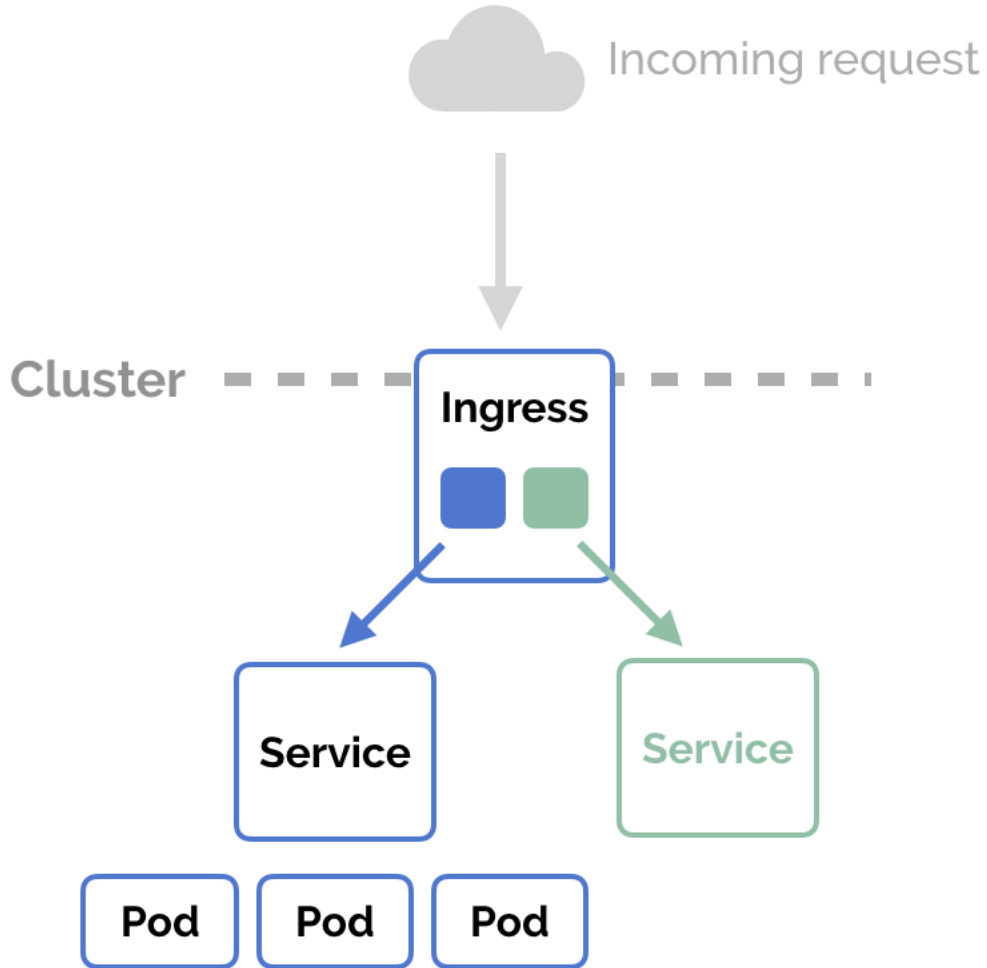
Minikube doesn't enable ingress by default; we have to enable it first using minikube addons enable ingress command.

- If you want to check the current setting for addons in minikube use minikube addons list command.

K8s Server:-

Minikube addons enable ingress

```
ubuntu@ip-172-31-18-126:~/k8s$ minikube addons enable ingress
* ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20220916-gd32f8c343
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20220916-gd32f8c343
  - Using image registry.k8s.io/ingress-nginx/controller:v1.5.1
* Verifying ingress addon...
```

Let's write ingress.yml and put the following code in it:

https://github.com/Hussain147/reddit-clone-k8s/blob/main/ingress.yaml

```
ubuntu@ip-172-31-18-126:~/k8s$ vi ingress.yaml
```

Ingress.yml:-

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

  name: ingress-reddit-app

spec:

  rules:

  - host: "domain.com"

   http:

    paths:

    - pathType: Prefix

     path: "/test"

     backend:

      service:

```
    name: reddit-clone-service

    port:

      number: 3000

- host: "*.domain.com"

  http:

    paths:

    - pathType: Prefix

      path: "/test"

      backend:

        service:

          name: reddit-clone-service

          port:

            number: 3000
```

Now you can able to create ingress for your service. kubectl apply -f ingress.yml use this command to apply ingress settings.

Verify that the ingress resource is running correctly by using kubectl get ingress ingress-reddit-app command.

```
ubuntu@ip-172-31-18-126:~/k8s$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/ingress-reddit-app created
```

Test Ingress:-

Now It's time to test your ingress so use the *curl -L domain/test* command in the terminal.

```
ubuntu@ip-172-31-18-126:~/k8s$ curl -L domain.com/test
```

You have successfully Deployed a Reddit Clone on Kubernetes with Ingress Enabled.