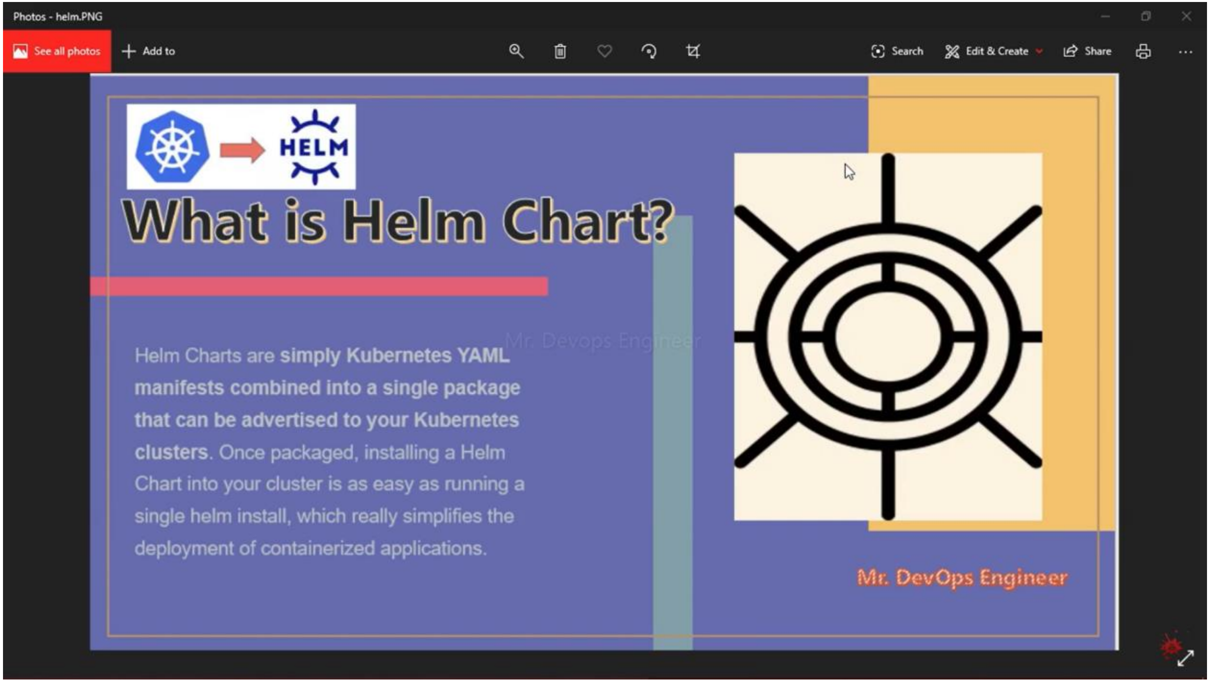
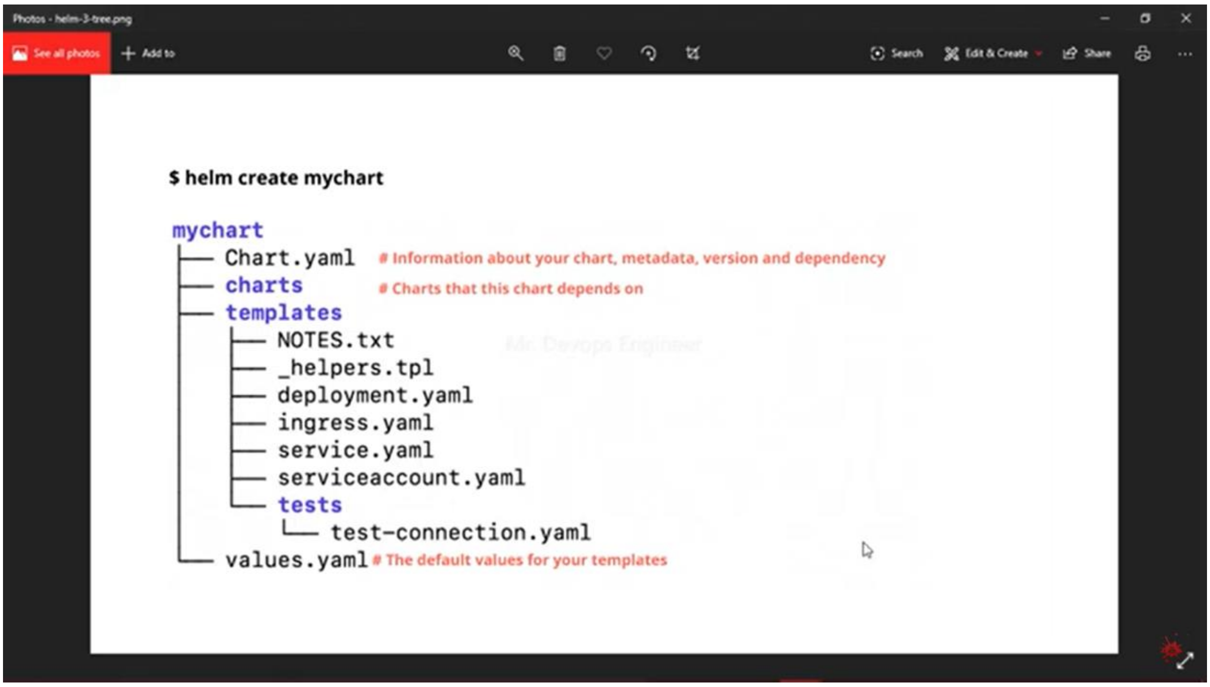
Helm chart is package manager for kubernetes clusters .



What is Helm Chart?

Helm Charts are **simply Kubernetes YAML manifests combined into a single package that can be advertised to your Kubernetes clusters**. Once packaged, installing a Helm Chart into your cluster is as easy as running a single helm install, which really simplifies the deployment of containerized applications.

In linux apt package manager is responsible for upgrading and installing packages .
We can install , upgrade , delete helm chart .



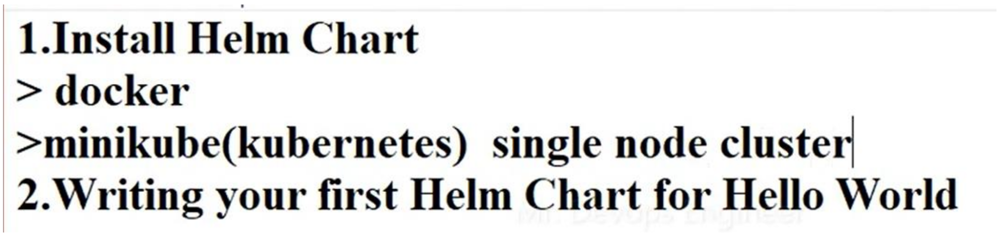
```
$ helm create mychart

mychart
├── Chart.yaml          # Information about your chart, metadata, version and dependency
├── charts              # Charts that this chart depends on
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   ├── ingress.yaml
│   ├── service.yaml
│   ├── serviceaccount.yaml
│   └── tests
│       └── test-connection.yaml
└── values.yaml # The default values for your templates
```

Helm create chart – using this command we get above structure .
Helm will create above chart .
Chart.yaml – contains metadata related to helm chart .

Helm will help in writing Kubernetes manifest .
Templates – it will contain all yaml files .
Values.yaml – basic configuration of files . we can add variables .

Install helm chart and create first helm chart .

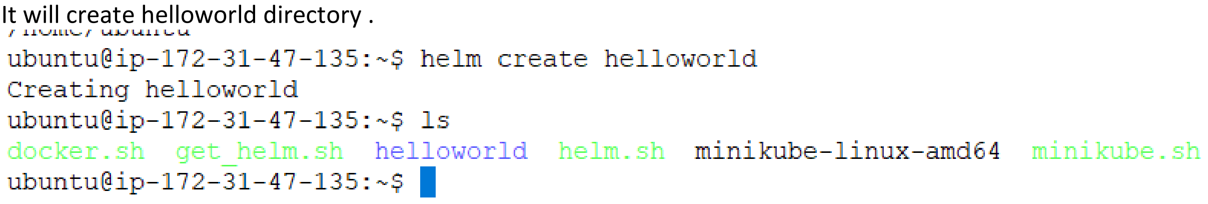
Create t2.medium ec2 instance . Install minikube .



1.Install Helm Chart
> docker
>minikube(kubernetes) single node cluster
2.Writing your first Helm Chart for Hello World

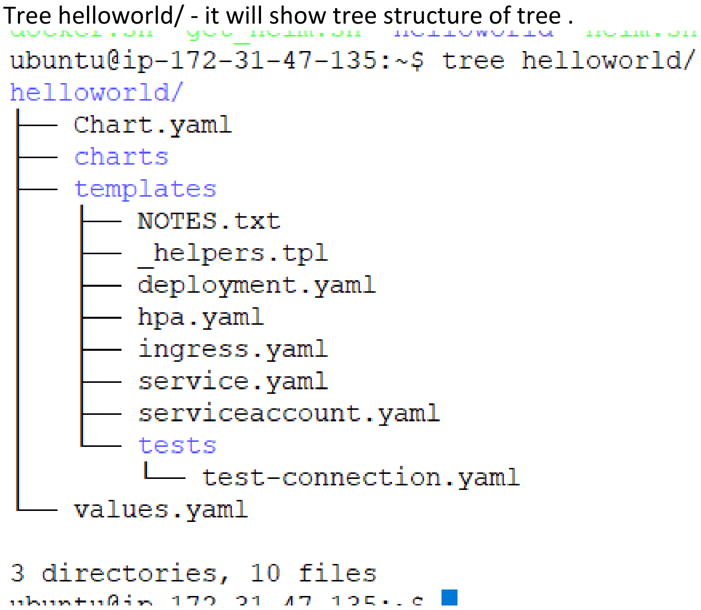
Install docker , minikube , helm
./docker.sh
./minikube.sh
./helm.sh

Creating first heml chart :
Helm create helloworld .
It will create helloworld directory .



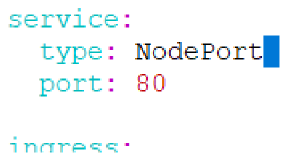
```
ubuntu@ip-172-31-47-135:~$ helm create helloworld
Creating helloworld
ubuntu@ip-172-31-47-135:~$ ls
docker.sh  get_helm.sh  helloworld  helm.sh  minikube-linux-amd64  minikube.sh
ubuntu@ip-172-31-47-135:~$
```

To check whether helm tree structure created or not we can use tree command .
Install tree .
Sudo apt install tree
Tree helloworld/ - it will show tree structure of tree .



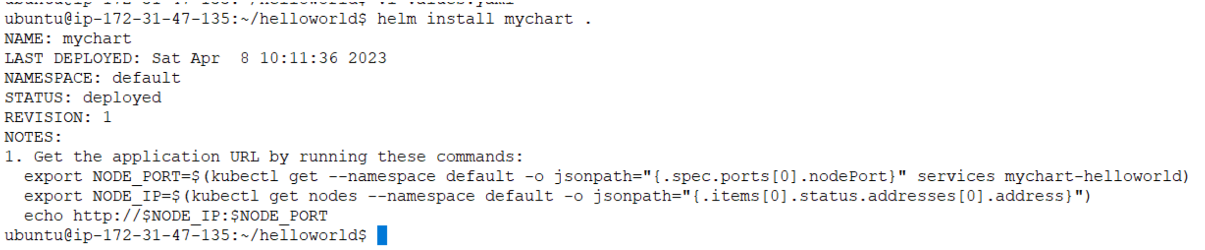
```
ubuntu@ip-172-31-47-135:~$ tree helloworld/
helloworld/
├── Chart.yaml
├── charts
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   ├── hpa.yaml
│   ├── ingress.yaml
│   ├── service.yaml
│   ├── serviceaccount.yaml
│   └── tests
│       └── test-connection.yaml
└── values.yaml

3 directories, 10 files
```

Change few values in values.yaml .
Change service from clusterip to nodeport (as clusterip can only be accessed inside nodeport)



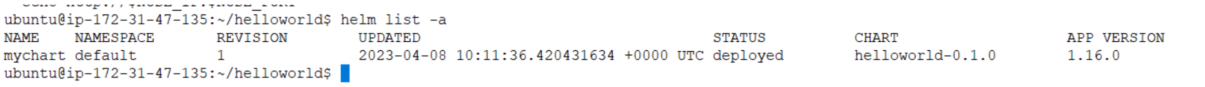
```
service:
  type: NodePort
  port: 80

ingress:
```

Helm install name directory_path_for_chart



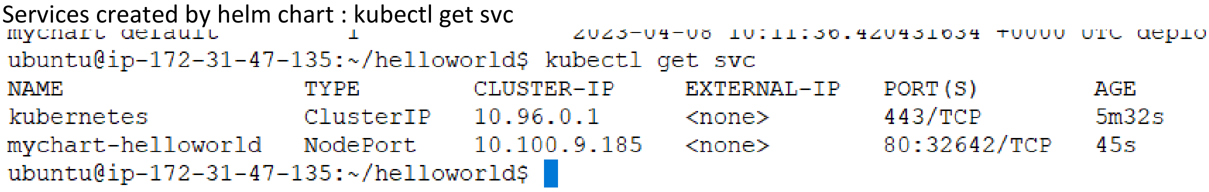
```
ubuntu@ip-172-31-47-135:~/helloworld$ helm install mychart .
NAME: mychart
LAST DEPLOYED: Sat Apr  8 10:11:36 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services mychart-helloworld)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
ubuntu@ip-172-31-47-135:~/helloworld$
```

Helm list -a – it will list all installed charts .



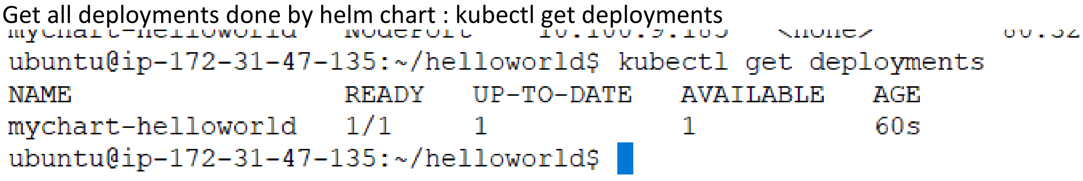
```
ubuntu@ip-172-31-47-135:~/helloworld$ helm list -a
NAME    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
mychart default         1               2023-04-08 10:11:36.420431634 +0000 UTC deployed        helloworld-0.1.0        1.16.0
ubuntu@ip-172-31-47-135:~/helloworld$
```

Services created by helm chart : kubectl get svc



```
mychart default         1               2023-04-08 10:11:36.420431634 +0000 UTC deplo
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl get svc
NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes           ClusterIP   10.96.0.1      <none>        443/TCP        5m32s
mychart-helloworld   NodePort    10.100.9.185   <none>        80:32642/TCP   45s
ubuntu@ip-172-31-47-135:~/helloworld$
```

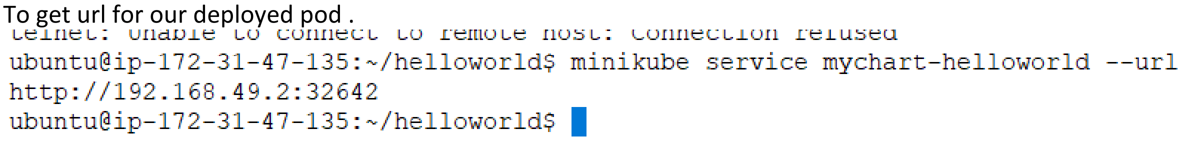
Get all deployments done by helm chart : kubectl get deployments



```
mychart-helloworld   NodePort    10.100.9.185   <none>        80:32
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl get deployments
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
mychart-helloworld   1/1     1            1           60s
ubuntu@ip-172-31-47-135:~/helloworld$
```

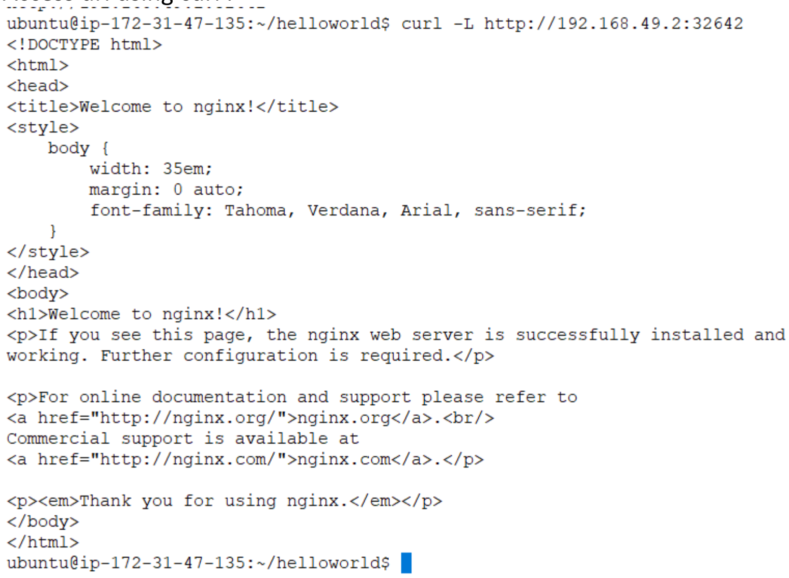
We can access pod using our server ip .

To get url for our deployed pod .



```
telnet: unable to connect to remote host: Connection refused
ubuntu@ip-172-31-47-135:~/helloworld$ minikube service mychart-helloworld --url
http://192.168.49.2:32642
ubuntu@ip-172-31-47-135:~/helloworld$
```

Access url using curl .

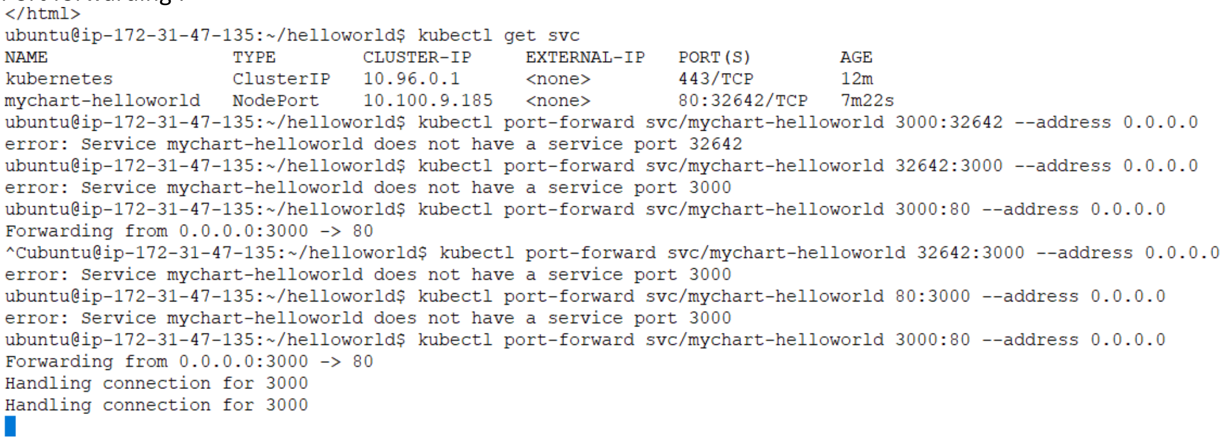


```
ubuntu@ip-172-31-47-135:~/helloworld$ curl -L http://192.168.49.2:32642
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ip-172-31-47-135:~/helloworld$
```

Url is accessible that means deployment is done .

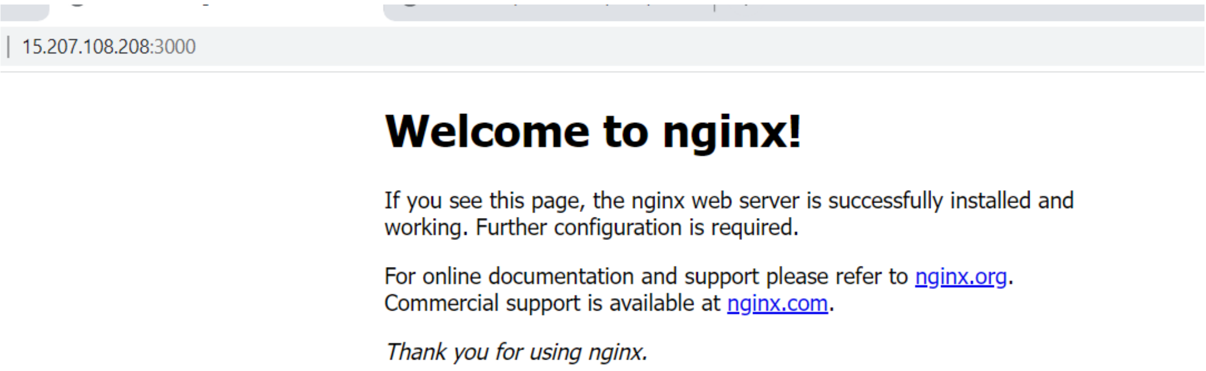
Expose application so that it can be accessible from browser .
Lets enable on port 3000 .
Enable in security group also .

Port forwarding :



```
</html>
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl get svc
NAME                 TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes           ClusterIP   10.96.0.1      <none>        443/TCP        12m
mychart-helloworld   NodePort    10.100.9.185   <none>        80:32642/TCP   7m22s
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl port-forward svc/mychart-helloworld 3000:32642 --address 0.0.0.0
error: Service mychart-helloworld does not have a service port 32642
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl port-forward svc/mychart-helloworld 32642:3000 --address 0.0.0.0
error: Service mychart-helloworld does not have a service port 3000
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl port-forward svc/mychart-helloworld 3000:80 --address 0.0.0.0
Forwarding from 0.0.0.0:3000 -> 80
^Cubuntu@ip-172-31-47-135:~/helloworld$ kubectl port-forward svc/mychart-helloworld 32642:3000 --address 0.0.0.0
error: Service mychart-helloworld does not have a service port 3000
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl port-forward svc/mychart-helloworld 80:3000 --address 0.0.0.0
error: Service mychart-helloworld does not have a service port 3000
ubuntu@ip-172-31-47-135:~/helloworld$ kubectl port-forward svc/mychart-helloworld 3000:80 --address 0.0.0.0
Forwarding from 0.0.0.0:3000 -> 80
Handling connection for 3000
Handling connection for 3000
```

After port forwarding nginx is accessible on port 3000 .



15.207.108.208:3000

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Nginx deployment done using helm chart .