#KubernetesIn30Days Challenge :-

#Day22 :-

Everything about Helm and usage in K8s.

Helm :-

→ Helm is a package manager for kubernetes applications. It simplifies the process of deploying and managing applications on K8s by providing a standardized way of define, install and upgrade applications.

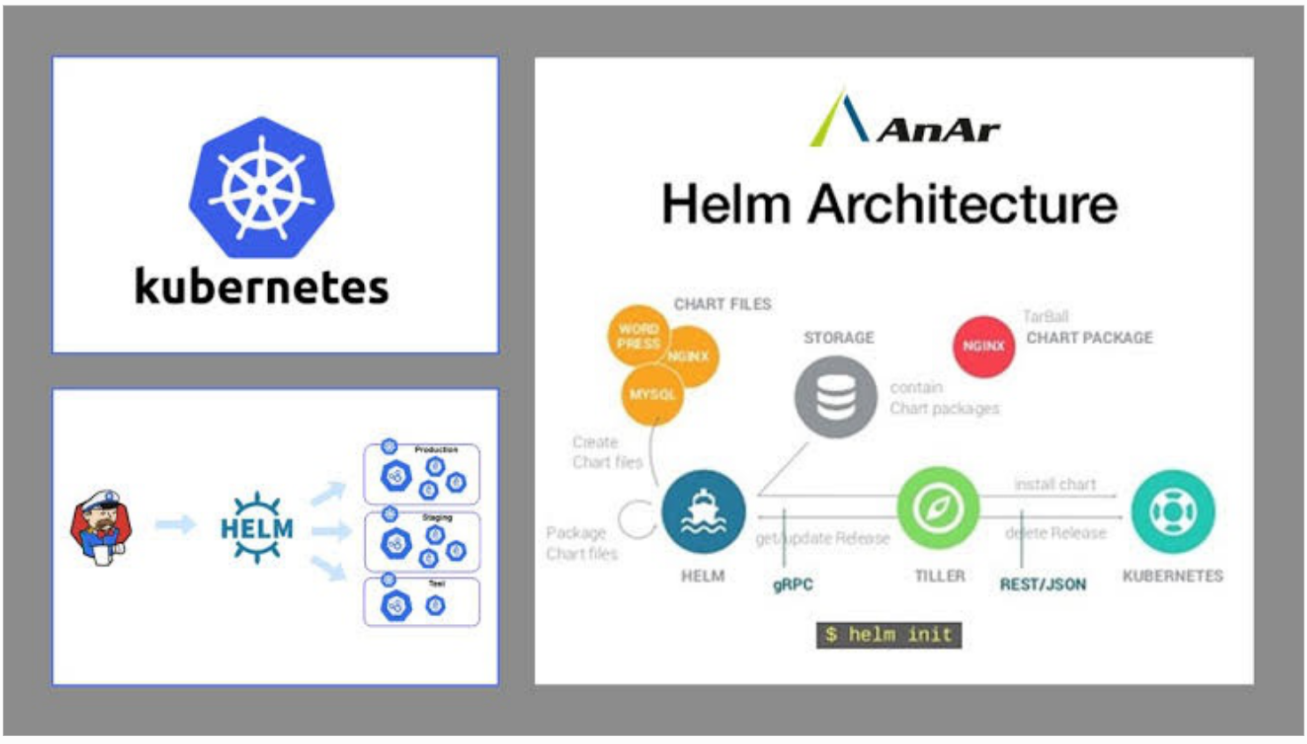
→ Helm uses charts, which are packages of pre-configured K8s resources.

Helm Basics :-

① chart :-

→ A Helm package is called a chart. A chart contains all the K8s resources needed to run an application, including deployments, services, ingress rules and more

→ charts are versioned and can be easily shared and reused.



② Templates :-

→ Inside a chart, you have templates written in YAML that defines K8s resources. Helm uses these templates to generate the actual K8s manifests with customized values.

③ Values :-

→ Helm allows you to parameterize your chart using values. These values can be provided at installation time, making charts flexible and reusable in different environments.

Helm workflow :-

Install → Upgrade → Rollback → delete

helm install my-release ./mychart

Is used to install particular chart

helm upgrade my-release ./mychart

Is used to upgrade package to latest version

helm rollback my-release 1

Is used to rollback package to previous version.

helm uninstall my-release

Is used to uninstall package.

Helm In Real time scenarios :-

1. Application Packaging
2. configuration management
3. Versioning and Rollbacks
4. community charts
5. customization.
6. CI/CD Integration

Handson - Helm :-

① Install helm on your Host / EC2 / VM using Installation guide

② create the Helm chart.

→ create new directory

→ go to respective directory and create webapp.

```
# helm create my-webapp
```

③ Edit the values.yaml file in webapp and change tag to stable keeping everything same

Its just NGINX server to install helm.

④ Have the Kubernetes cluster. (Here it is Minikube cluster)

```
#helm install my-webapp-release ./webapp
```

⑤ Verify deployments

① deployments

② pods

③ services.

⑥ Access the application using NodePort service at the minikube.

⑦ Now change the tag to latest and use below cmd:

```
# helm upgrade my-webapp-release ./webapp
```

⑧ Verify deployments

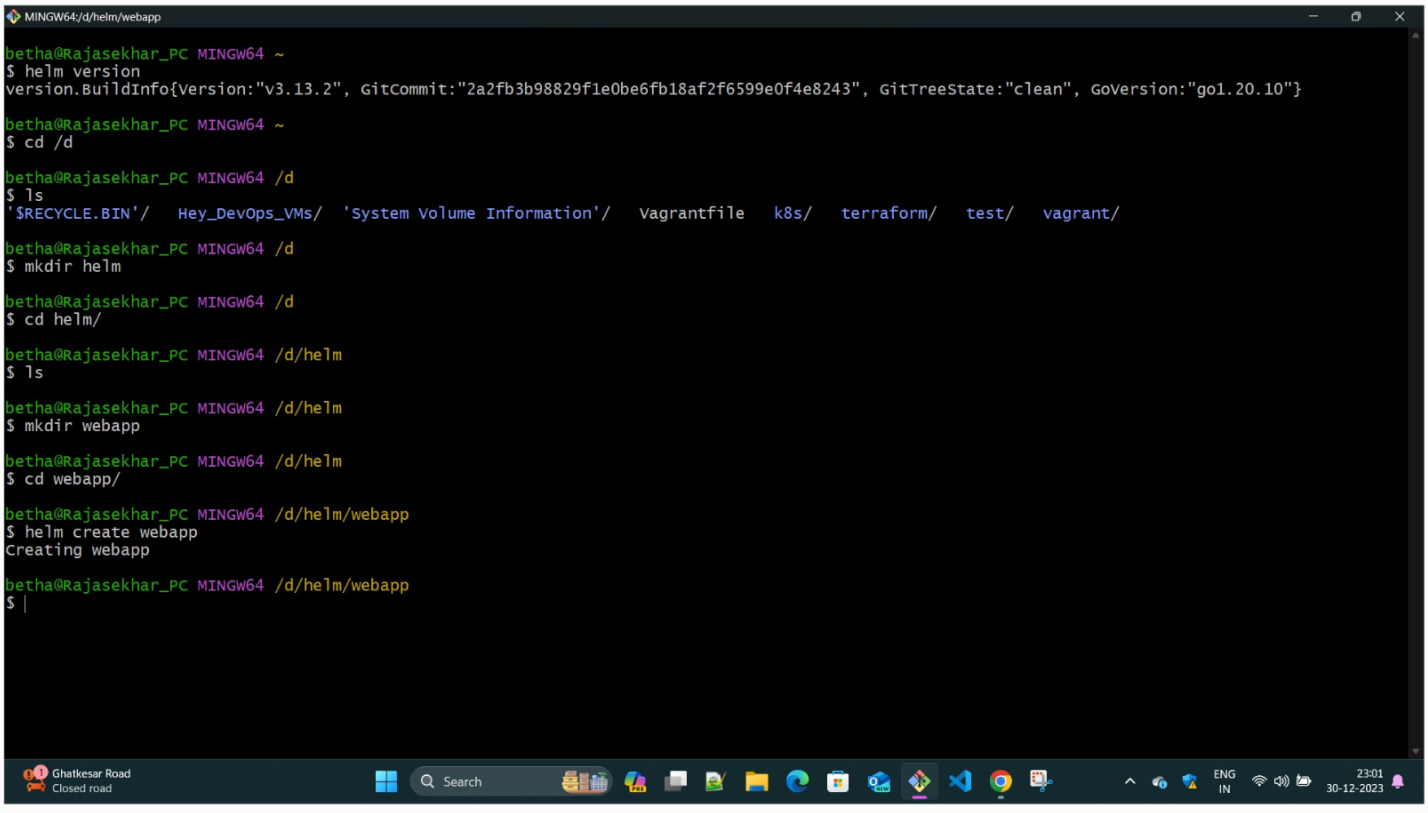
① deployments
② pods
③ services.

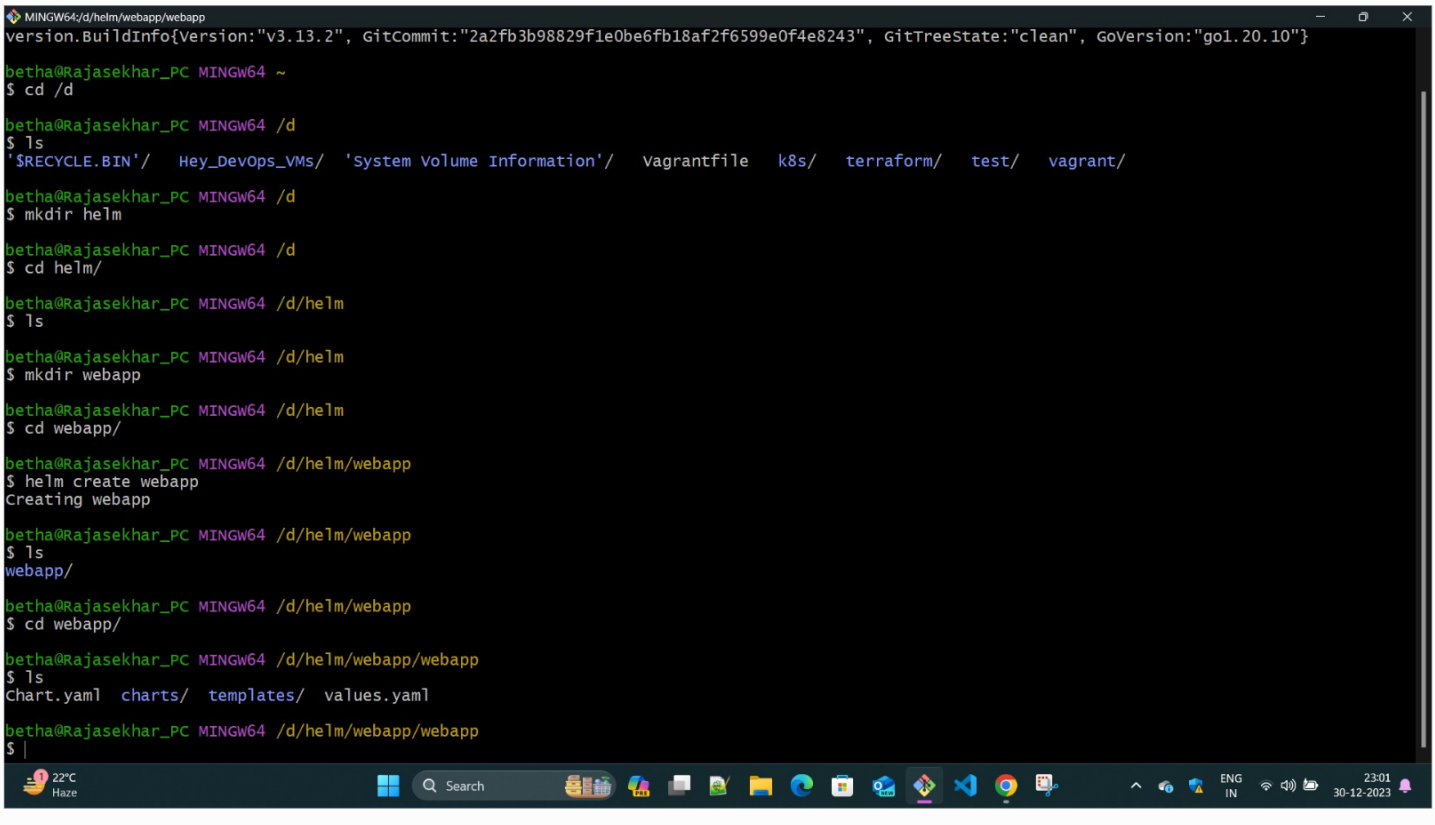
⑨ uninstall the webapp for helm

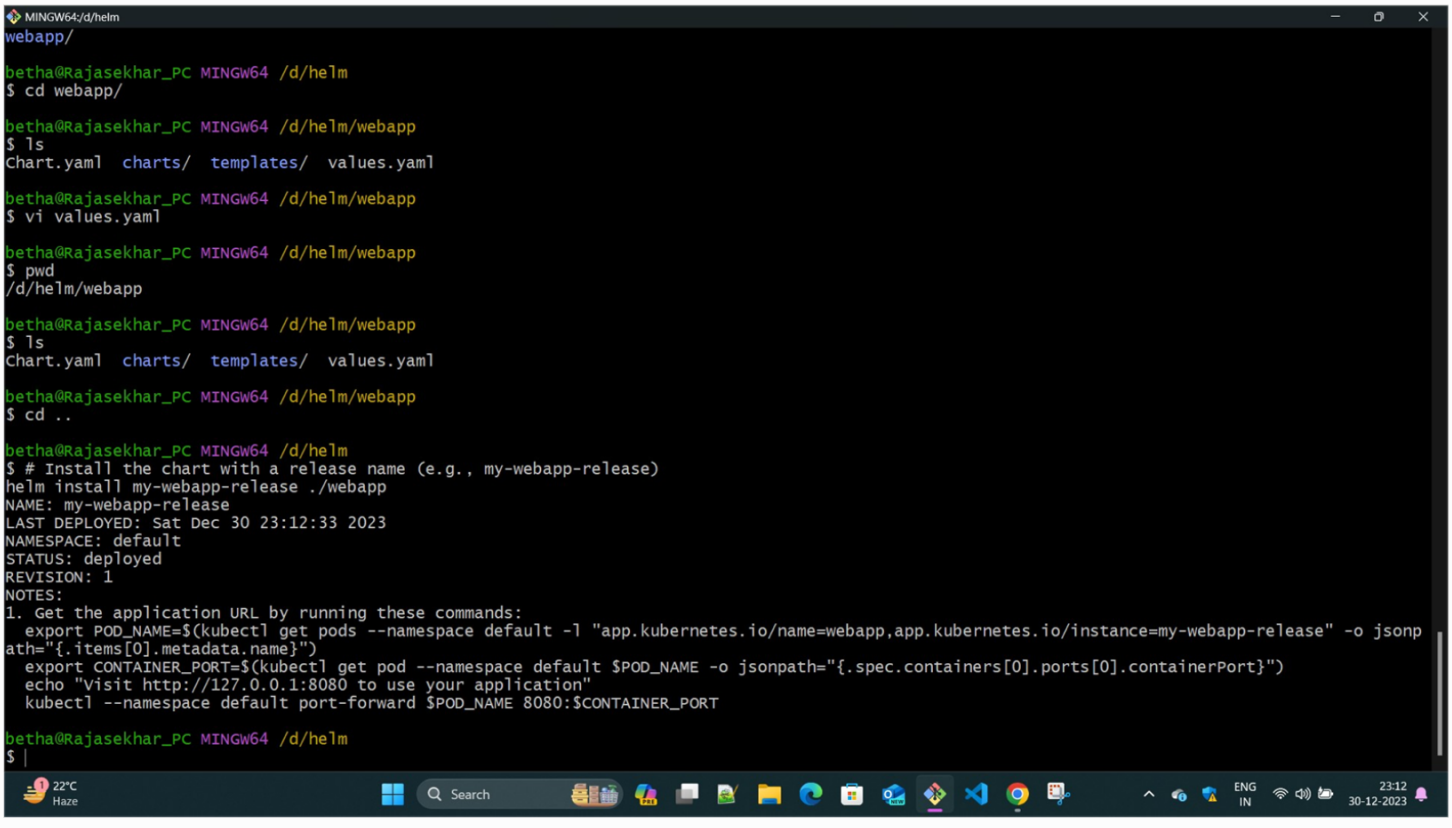
```
#helm uninstall my-webapp-release
```

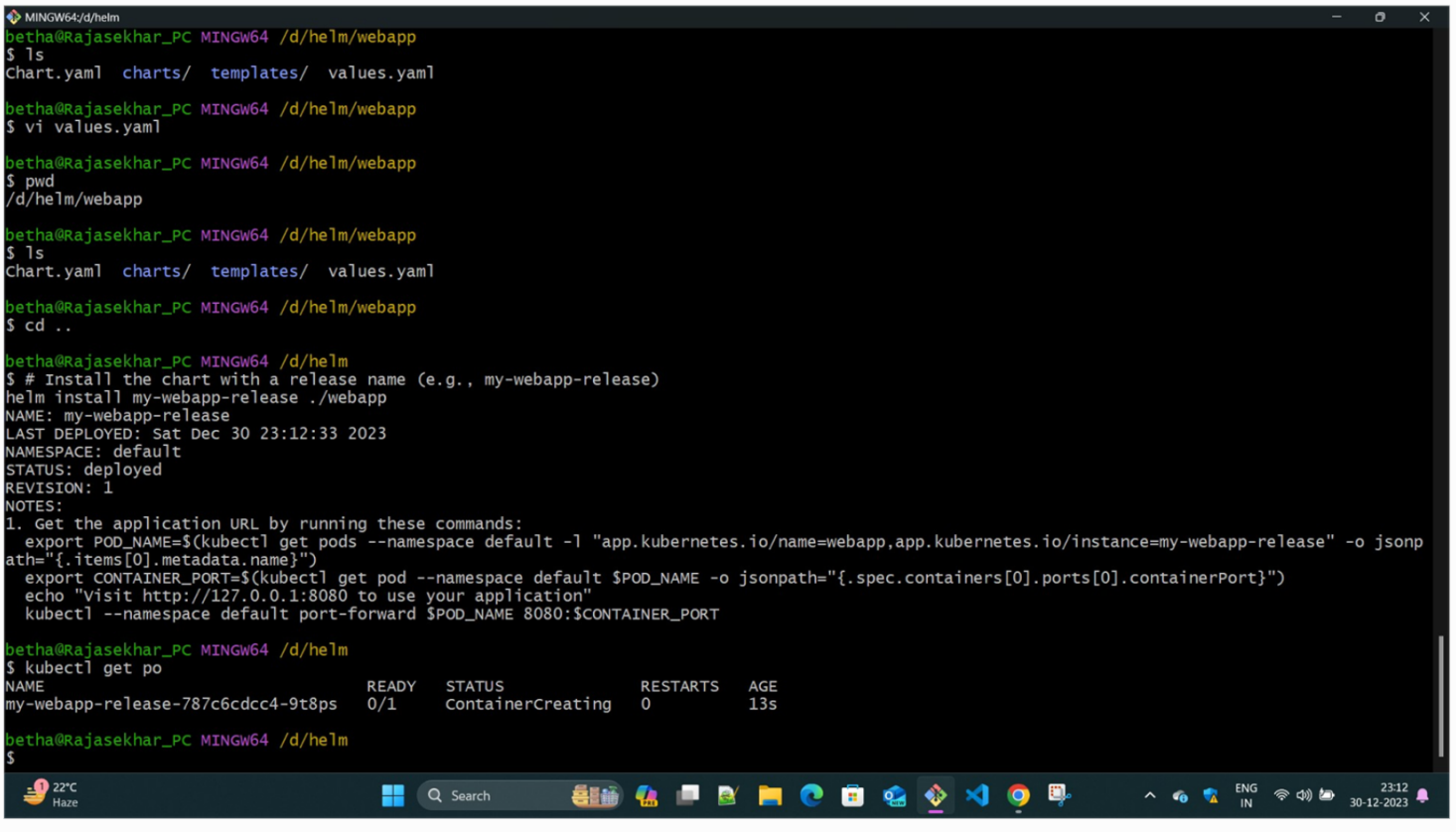
Thats it you have done your Demo on Helm.

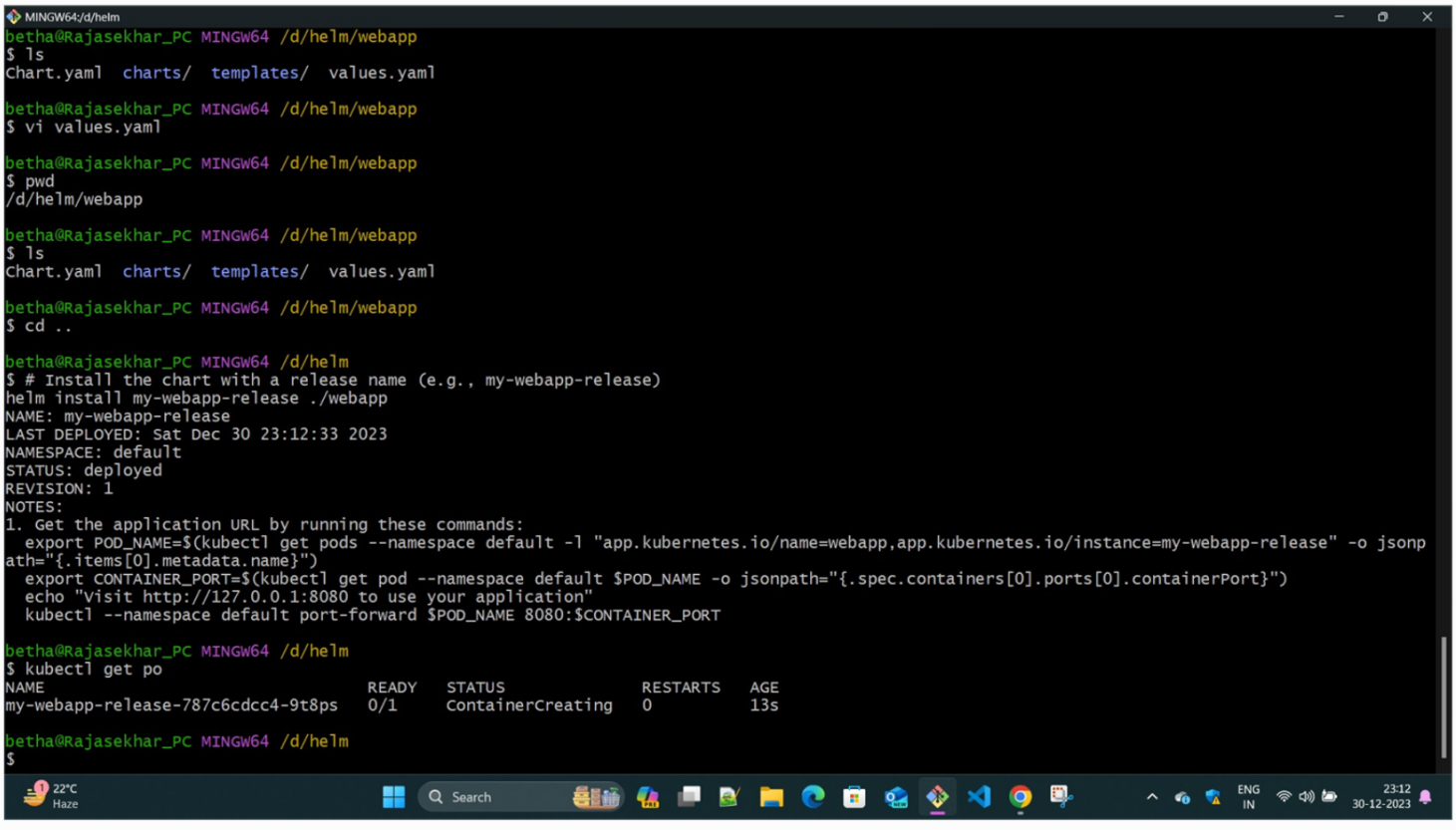
You can checkout Work samples below. on this Handson. :-

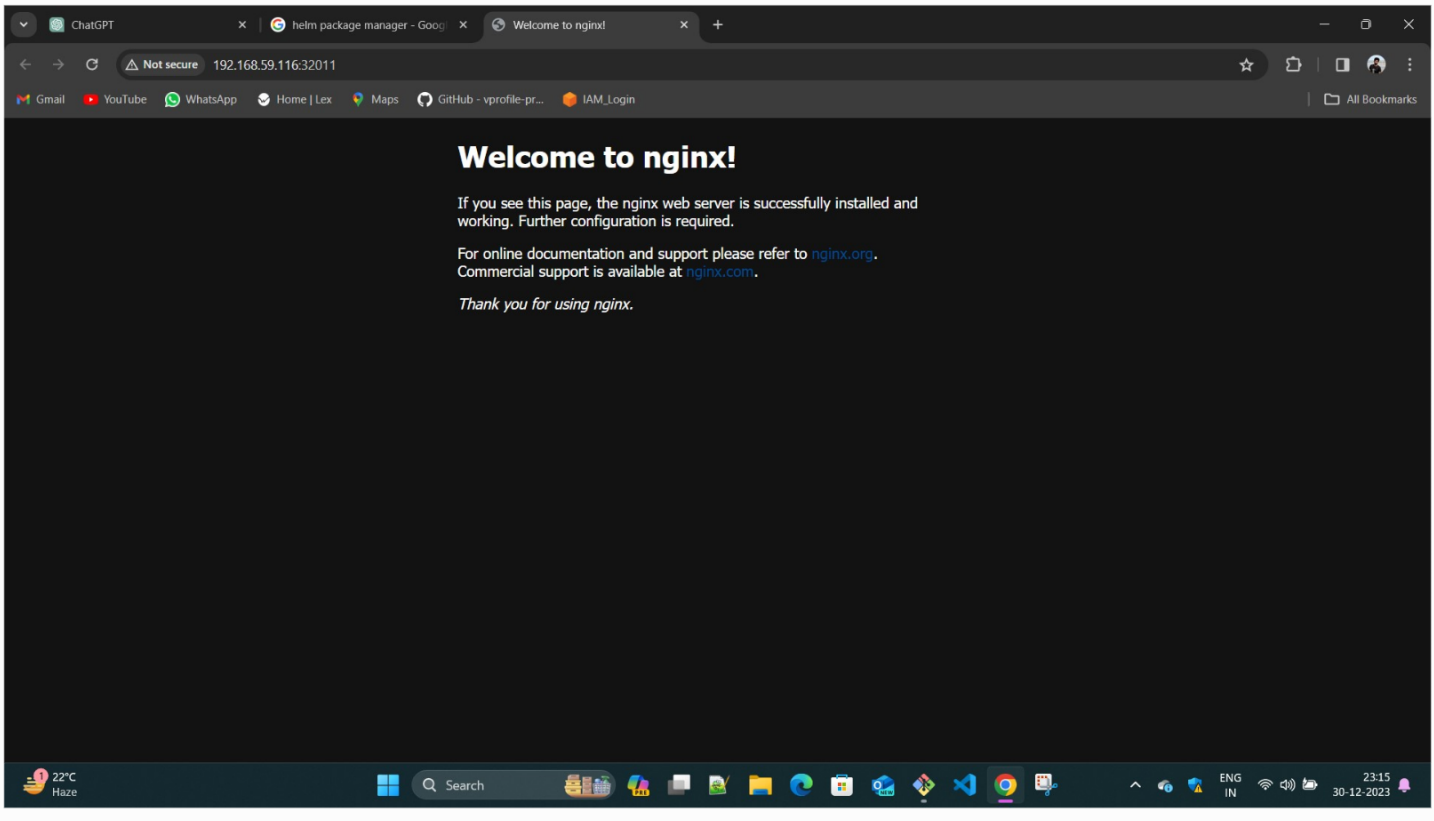

MINGW64:/d/helm/webapp
betha@Rajasekhar_PC MINGW64 ~
$ helm version
version.BuildInfo{Version:"v3.13.2", GitCommit:"2a2fb3b98829f1e0be6fb18af2f6599e0f4e8243", GitTreeState:"clean", GoVersion:"go1.20.10"}
betha@Rajasekhar_PC MINGW64 ~
$ cd /d
betha@Rajasekhar_PC MINGW64 /d
$ ls
'$RECYCLE.BIN'/ Hey_DevOps_VMs/ 'System Volume Information'/ Vagrantfile k8s/ terraform/ test/ vagrant/
betha@Rajasekhar_PC MINGW64 /d
$ mkdir helm
betha@Rajasekhar_PC MINGW64 /d
$ cd helm/
betha@Rajasekhar_PC MINGW64 /d/helm
$ ls
betha@Rajasekhar_PC MINGW64 /d/helm
$ mkdir webapp
betha@Rajasekhar_PC MINGW64 /d/helm
$ cd webapp/
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ helm create webapp
Creating webapp
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$
Ghatkesar Road
Closed road
Search
ENG
IN
23:01
30-12-2023

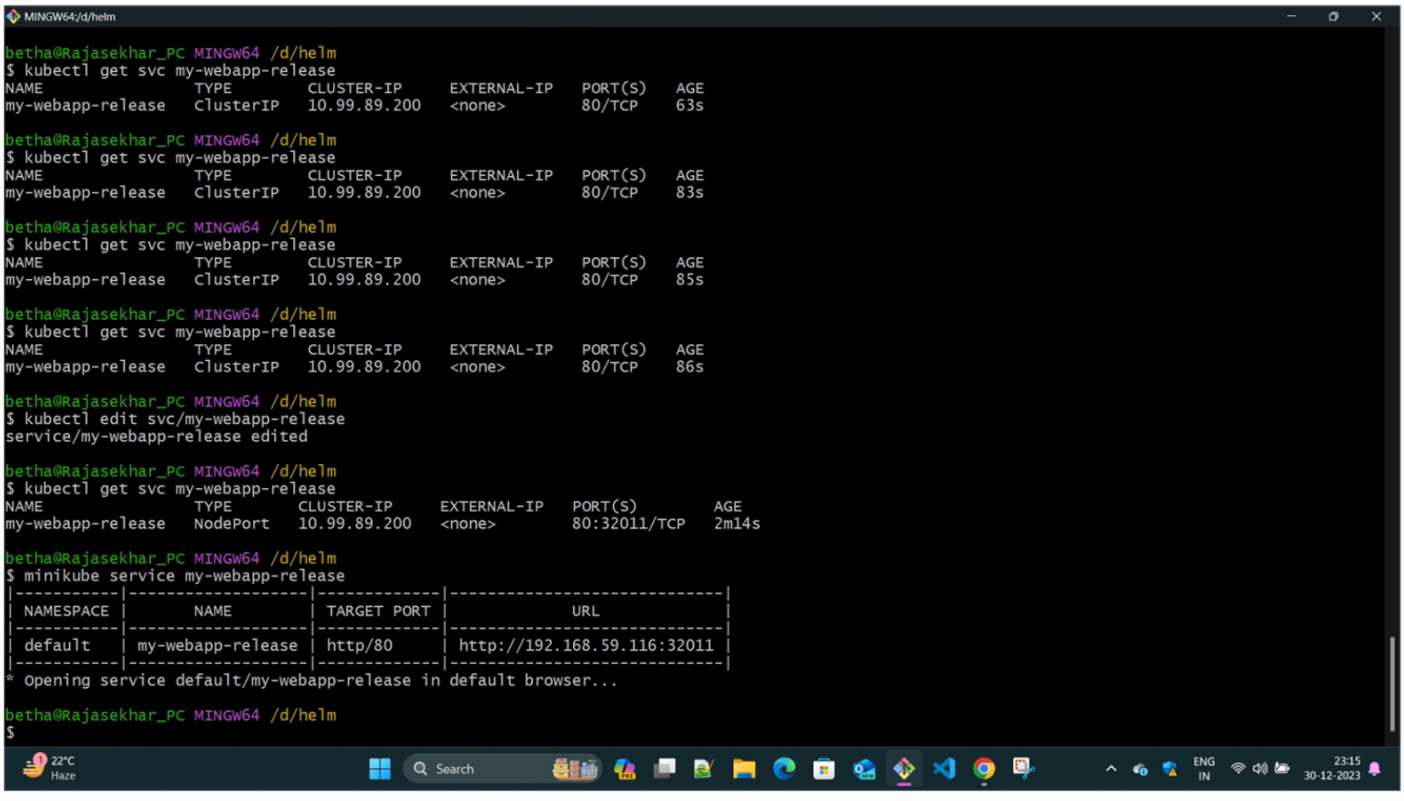

MINGW64:/d/helm/webapp/webapp
version.BuildInfo{Version:"v3.13.2", GitCommit:"2a2fb3b98829f1e0be6fb18af2f6599e0f4e8243", GitTreeState:"clean", GoVersion:"go1.20.10"}
betha@Rajasekhar_PC MINGW64 ~
$ cd /d
betha@Rajasekhar_PC MINGW64 /d
$ ls
'$RECYCLE.BIN'/ Hey_DevOps_VMs/ 'System Volume Information'/ Vagrantfile k8s/ terraform/ test/ vagrant/
betha@Rajasekhar_PC MINGW64 /d
$ mkdir helm
betha@Rajasekhar_PC MINGW64 /d
$ cd helm/
betha@Rajasekhar_PC MINGW64 /d/helm
$ ls
betha@Rajasekhar_PC MINGW64 /d/helm
$ mkdir webapp
betha@Rajasekhar_PC MINGW64 /d/helm
$ cd webapp/
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ helm create webapp
Creating webapp
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ ls
webapp/
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ cd webapp/
betha@Rajasekhar_PC MINGW64 /d/helm/webapp/webapp
$ ls
Chart.yaml charts/ templates/ values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp/webapp
$
22°C
Haze
Search
ENG
IN
23:01
30-12-2023

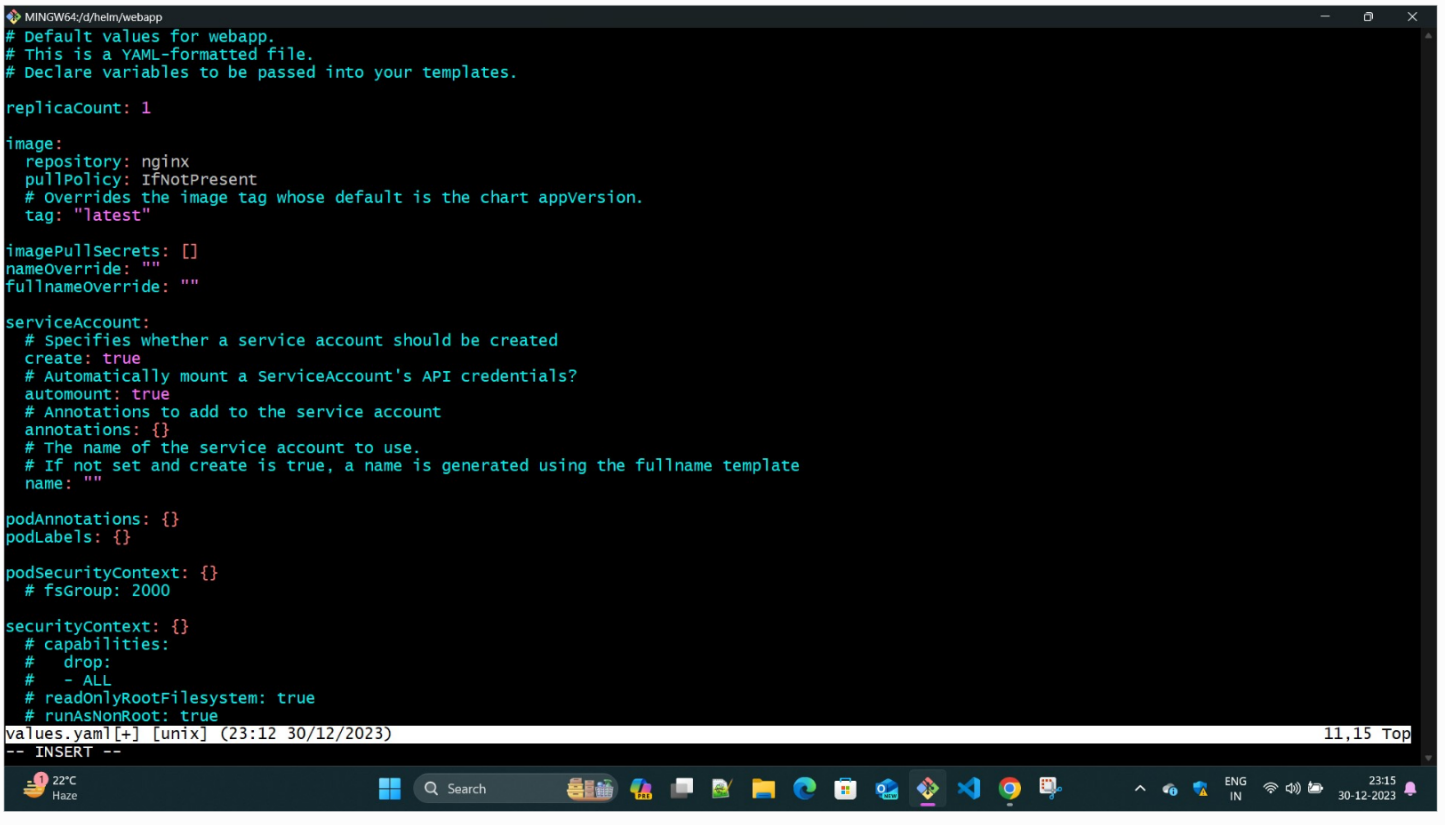

MINGW64:/d/helm
webapp/
betha@Rajasekhar_PC MINGW64 /d/helm
$ cd webapp/
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ ls
Chart.yaml charts/ templates/ values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ vi values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ pwd
/d/helm/webapp
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ ls
Chart.yaml charts/ templates/ values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ cd ..
betha@Rajasekhar_PC MINGW64 /d/helm
$ # Install the chart with a release name (e.g., my-webapp-release)
helm install my-webapp-release ./webapp
NAME: my-webapp-release
LAST DEPLOYED: Sat Dec 30 23:12:33 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=webapp,app.kubernetes.io/instance=my-webapp-release" -o jsonp
ath="{.items[0].metadata.name}")
export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
echo "Visit http://127.0.0.1:8080 to use your application"
kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
betha@Rajasekhar_PC MINGW64 /d/helm
$
22°C Haze
Search
ENG IN
23:12 30-12-2023

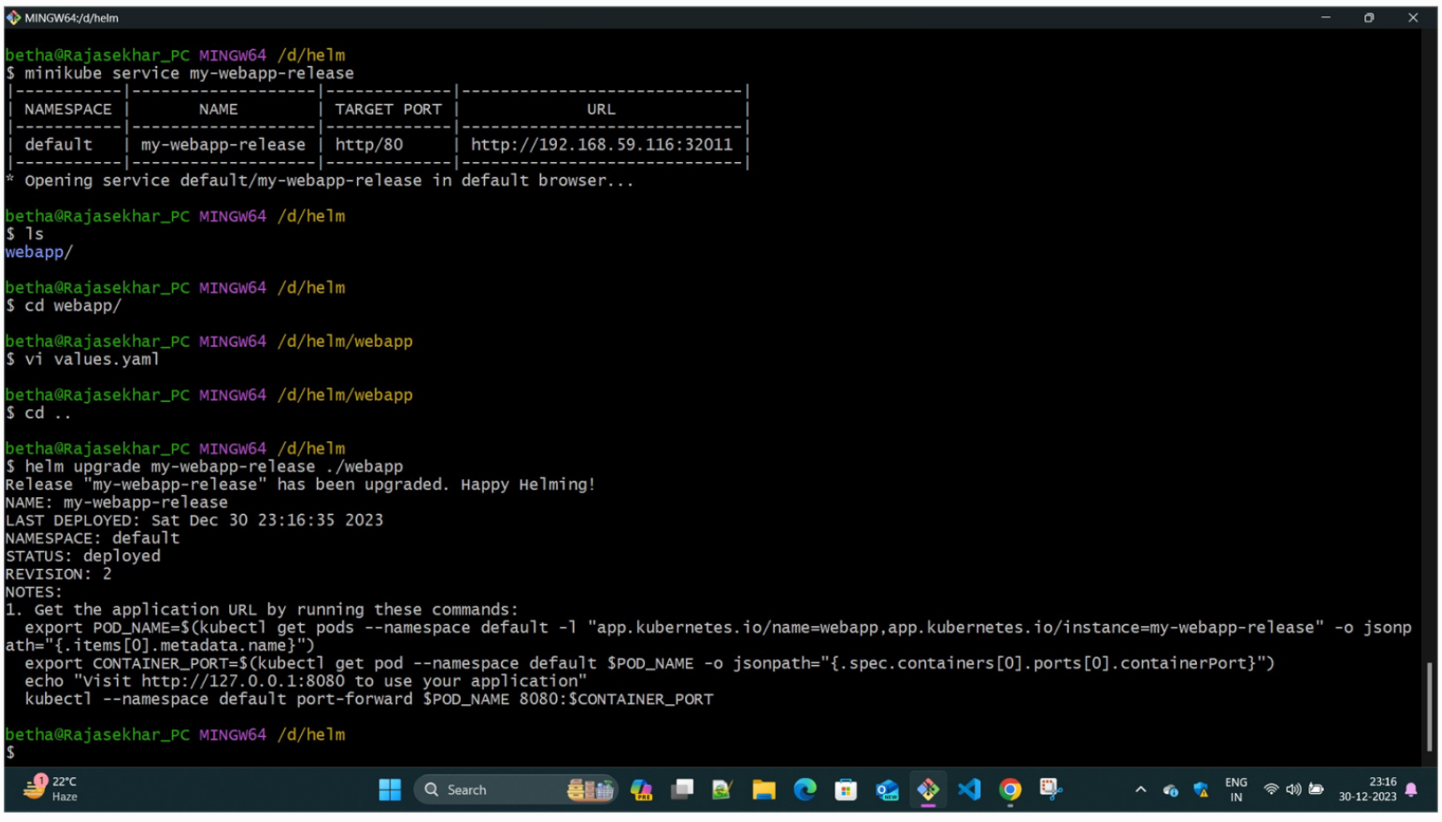

MINGW64:/d/helm
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ ls
Chart.yaml charts/ templates/ values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ vi values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ pwd
/d/helm/webapp
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ ls
Chart.yaml charts/ templates/ values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ cd ..
betha@Rajasekhar_PC MINGW64 /d/helm
$ # Install the chart with a release name (e.g., my-webapp-release)
helm install my-webapp-release ./webapp
NAME: my-webapp-release
LAST DEPLOYED: Sat Dec 30 23:12:33 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=webapp,app.kubernetes.io/instance=my-webapp-release" -o jsonp
ath="{.items[0].metadata.name}")
export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
echo "Visit http://127.0.0.1:8080 to use your application"
kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get po
NAME READY STATUS RESTARTS AGE
my-webapp-release-787c6cdcc4-9t8ps 0/1 ContainerCreating 0 13s
betha@Rajasekhar_PC MINGW64 /d/helm
$
22°C Haze
Search
ENG IN
23:12 30-12-2023

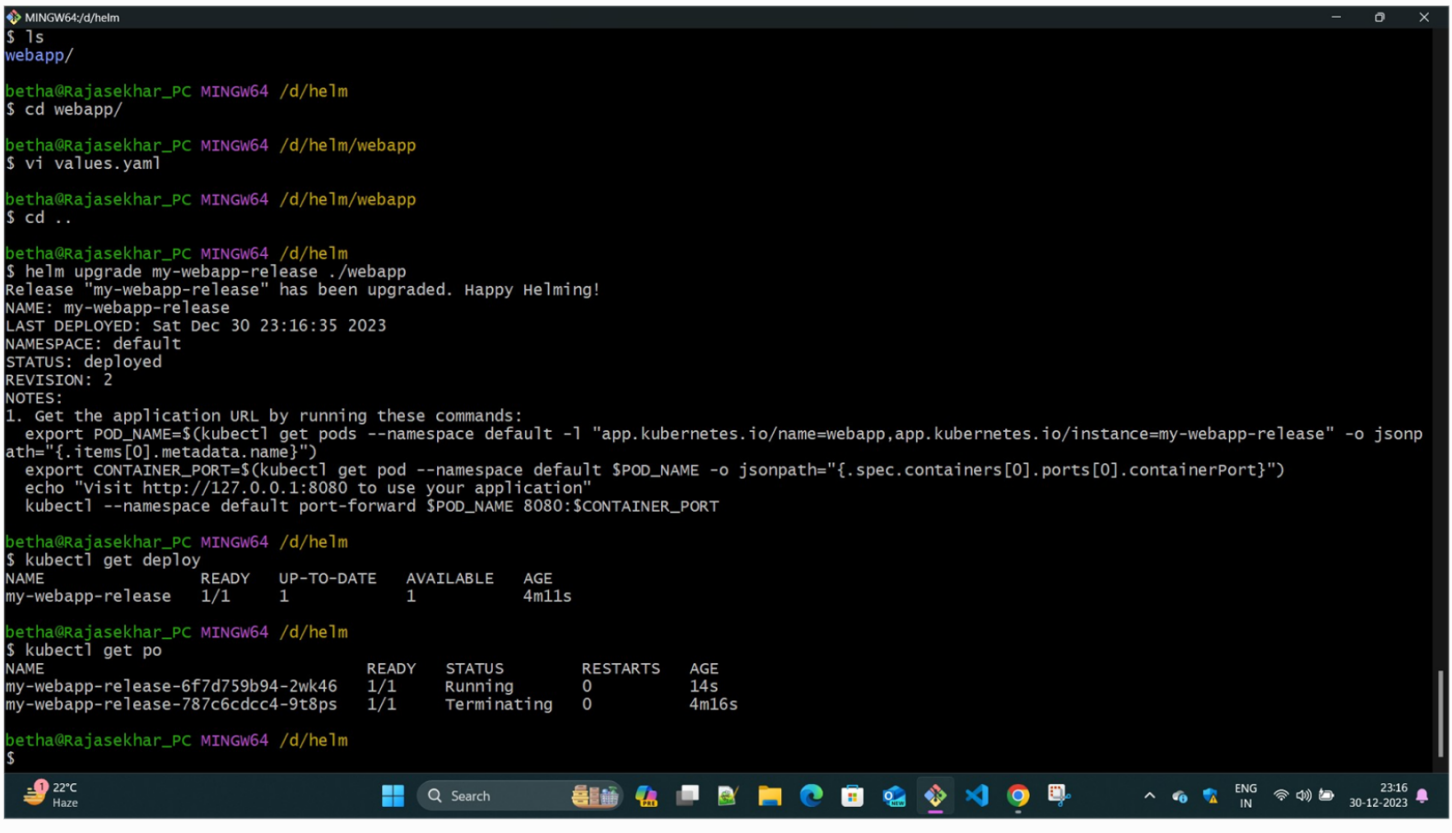

MINGW64:/d/helm
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ ls
Chart.yaml charts/ templates/ values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ vi values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ pwd
/d/helm/webapp
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ ls
Chart.yaml charts/ templates/ values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ cd ..
betha@Rajasekhar_PC MINGW64 /d/helm
$ # Install the chart with a release name (e.g., my-webapp-release)
helm install my-webapp-release ./webapp
NAME: my-webapp-release
LAST DEPLOYED: Sat Dec 30 23:12:33 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=webapp,app.kubernetes.io/instance=my-webapp-release" -o jsonpath="{.items[0].metadata.name}")
export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
echo "Visit http://127.0.0.1:8080 to use your application"
kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get po
NAME READY STATUS RESTARTS AGE
my-webapp-release-787c6cdcc4-9t8ps 0/1 ContainerCreating 0 13s
betha@Rajasekhar_PC MINGW64 /d/helm
$
22°C Haze
Search
ENG IN
23:12 30-12-2023

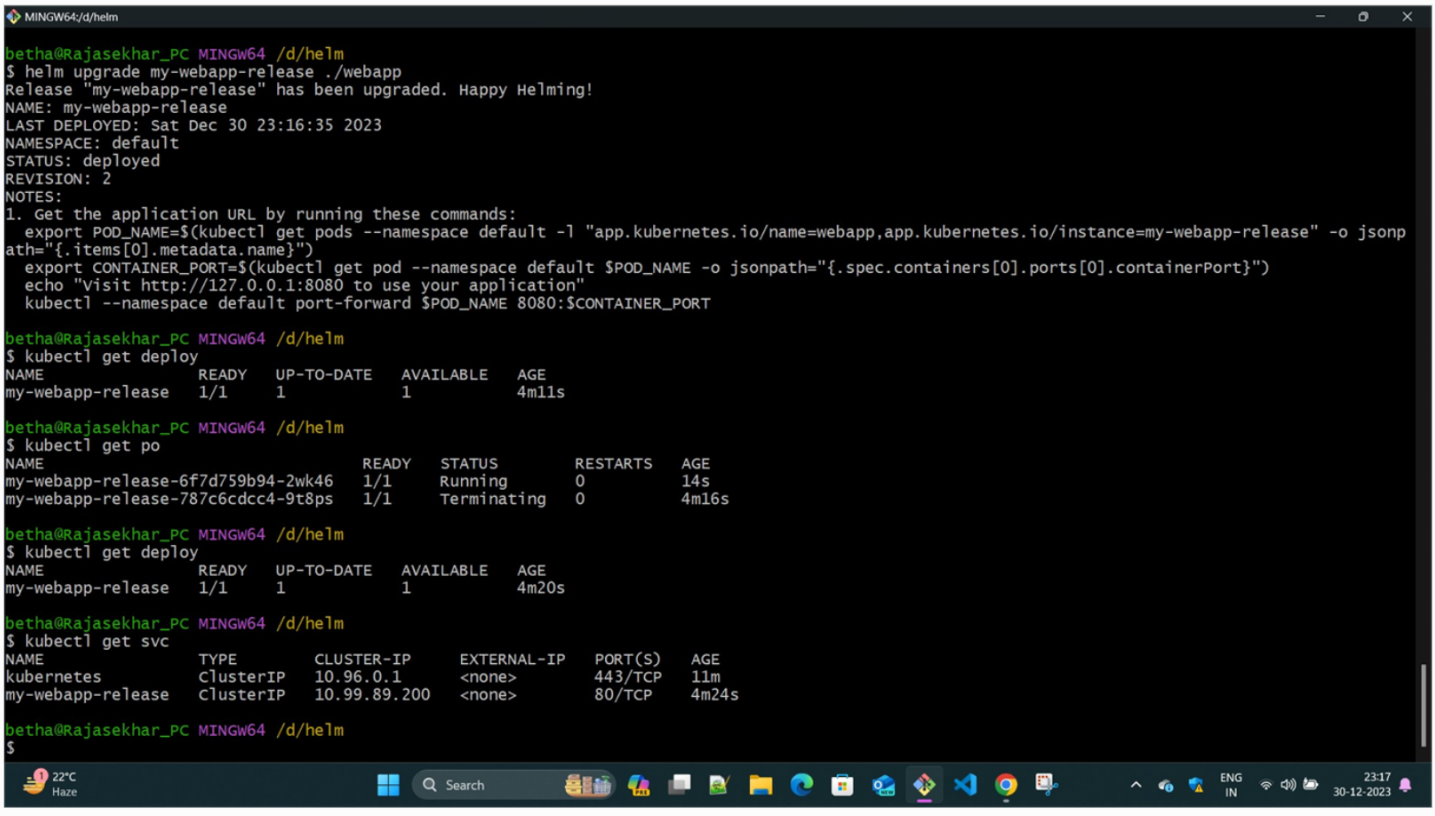

ChatGPT
helm package manager - Goog
Welcome to nginx!
Not secure 192.168.59.116:32011
Gmail YouTube WhatsApp Home | Lex Maps GitHub - vprofile-pr... IAM_Login
All Bookmarks
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.
22°C Haze
Search
ENG IN
23:15 30-12-2023

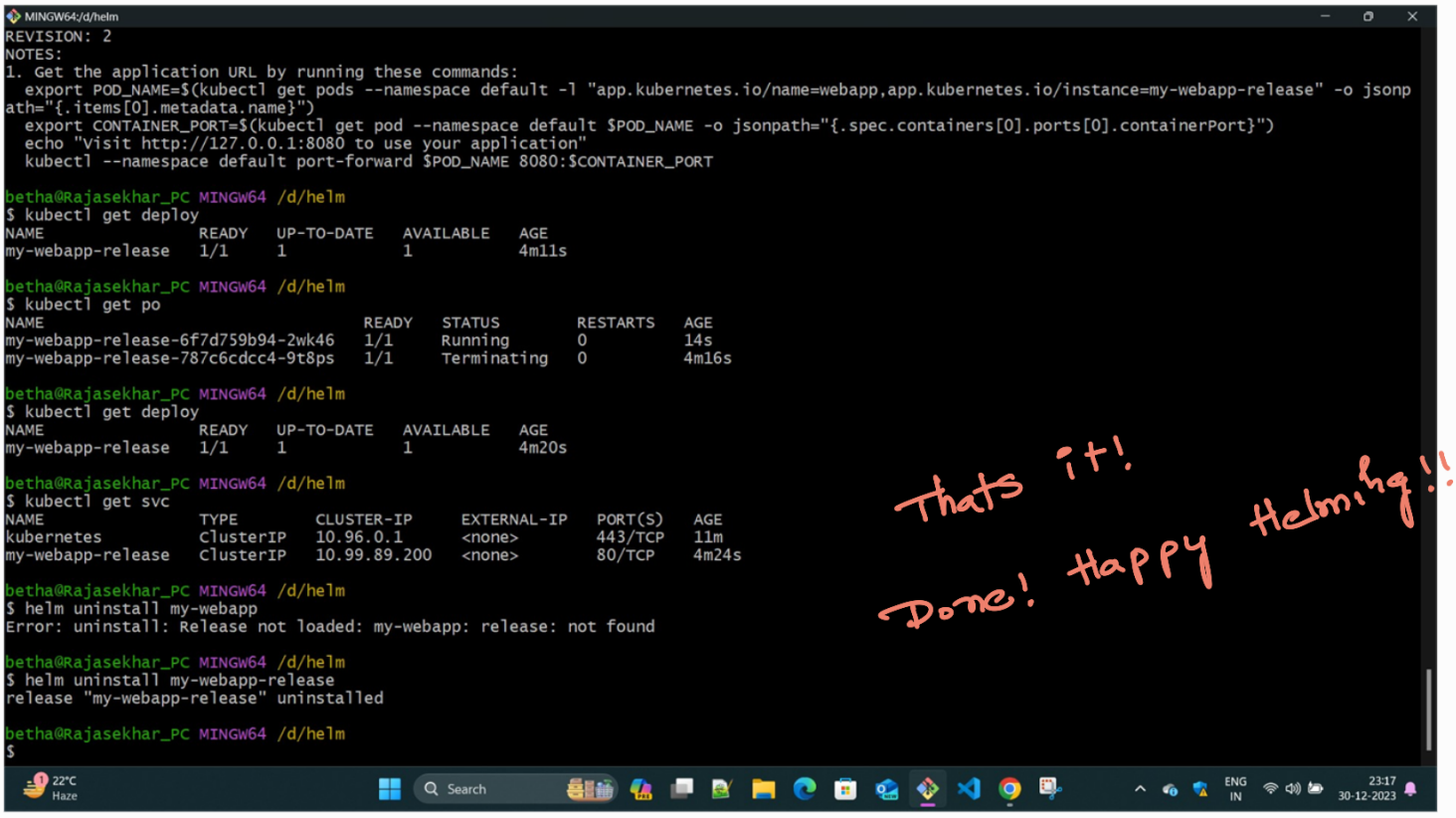

MINGW64:/d/helm
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get svc my-webapp-release
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
my-webapp-release ClusterIP 10.99.89.200 <none> 80/TCP 63s
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get svc my-webapp-release
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
my-webapp-release ClusterIP 10.99.89.200 <none> 80/TCP 83s
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get svc my-webapp-release
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
my-webapp-release ClusterIP 10.99.89.200 <none> 80/TCP 85s
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get svc my-webapp-release
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
my-webapp-release ClusterIP 10.99.89.200 <none> 80/TCP 86s
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl edit svc/my-webapp-release
service/my-webapp-release edited
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get svc my-webapp-release
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
my-webapp-release NodePort 10.99.89.200 <none> 80:32011/TCP 2m14s
betha@Rajasekhar_PC MINGW64 /d/helm
$ minikube service my-webapp-release
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-webapp-release | http/80 | http://192.168.59.116:32011 |
* Opening service default/my-webapp-release in default browser...
betha@Rajasekhar_PC MINGW64 /d/helm
$
22°C Haze
Search
ENG IN
23:15 30-12-2023


MINGW64:/d/helm/webapp
Default values for webapp.
This is a YAML-formatted file.
Declare variables to be passed into your templates.
replicaCount: 1
image:
repository: nginx
pullPolicy: IfNotPresent
Overrides the image tag whose default is the chart appVersion.
tag: "latest"
imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""
serviceAccount:
Specifies whether a service account should be created
create: true
Automatically mount a ServiceAccount's API credentials?
automount: true
Annotations to add to the service account
annotations: {}
The name of the service account to use.
If not set and create is true, a name is generated using the fullname template
name: ""
podAnnotations: {}
podLabels: {}
podSecurityContext: {}
fsGroup: 2000
securityContext: {}
capabilities:
drop:
- ALL
readOnlyRootFilesystem: true
runAsNonRoot: true
values.yaml[+] [unix] (23:12 30/12/2023) 11,15 Top
-- INSERT --
22°C Haze
Search
ENG IN
23:15 30-12-2023


MINGW64:/d/helm
betha@Rajasekhar_PC MINGW64 /d/helm
$ minikube service my-webapp-release
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-webapp-release | http/80 | http://192.168.59.116:32011 |
* Opening service default/my-webapp-release in default browser...
betha@Rajasekhar_PC MINGW64 /d/helm
$ ls
webapp/
betha@Rajasekhar_PC MINGW64 /d/helm
$ cd webapp/
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ vi values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ cd ..
betha@Rajasekhar_PC MINGW64 /d/helm
$ helm upgrade my-webapp-release ./webapp
Release "my-webapp-release" has been upgraded. Happy Helming!
NAME: my-webapp-release
LAST DEPLOYED: Sat Dec 30 23:16:35 2023
NAMESPACE: default
STATUS: deployed
REVISION: 2
NOTES:
1. Get the application URL by running these commands:
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=webapp,app.kubernetes.io/instance=my-webapp-release" -o jsonpath="{.items[0].metadata.name}")
export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
echo "Visit http://127.0.0.1:8080 to use your application"
kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
betha@Rajasekhar_PC MINGW64 /d/helm
$
22°C Haze
Search
ENG IN
23:16 30-12-2023


MINGW64:/d/helm
$ ls
webapp/
betha@Rajasekhar_PC MINGW64 /d/helm
$ cd webapp/
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ vi values.yaml
betha@Rajasekhar_PC MINGW64 /d/helm/webapp
$ cd ..
betha@Rajasekhar_PC MINGW64 /d/helm
$ helm upgrade my-webapp-release ./webapp
Release "my-webapp-release" has been upgraded. Happy Helming!
NAME: my-webapp-release
LAST DEPLOYED: Sat Dec 30 23:16:35 2023
NAMESPACE: default
STATUS: deployed
REVISION: 2
NOTES:
1. Get the application URL by running these commands:
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=webapp,app.kubernetes.io/instance=my-webapp-release" -o jsonpath="{.items[0].metadata.name}")
export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
echo "Visit http://127.0.0.1:8080 to use your application"
kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get deploy
NAME READY UP-TO-DATE AVAILABLE AGE
my-webapp-release 1/1 1 1 4m11s
betha@Rajasekhar_PC MINGW64 /d/helm
$ kubectl get po
NAME READY STATUS RESTARTS AGE
my-webapp-release-6f7d759b94-2wk46 1/1 Running 0 14s
my-webapp-release-787c6cdcc4-9t8ps 1/1 Terminating 0 4m16s
betha@Rajasekhar_PC MINGW64 /d/helm
$
22°C Haze
Search
ENG IN
23:16 30-12-2023





Thats it!

Done! Happy Helming!!!

Thanks for Reading

☺