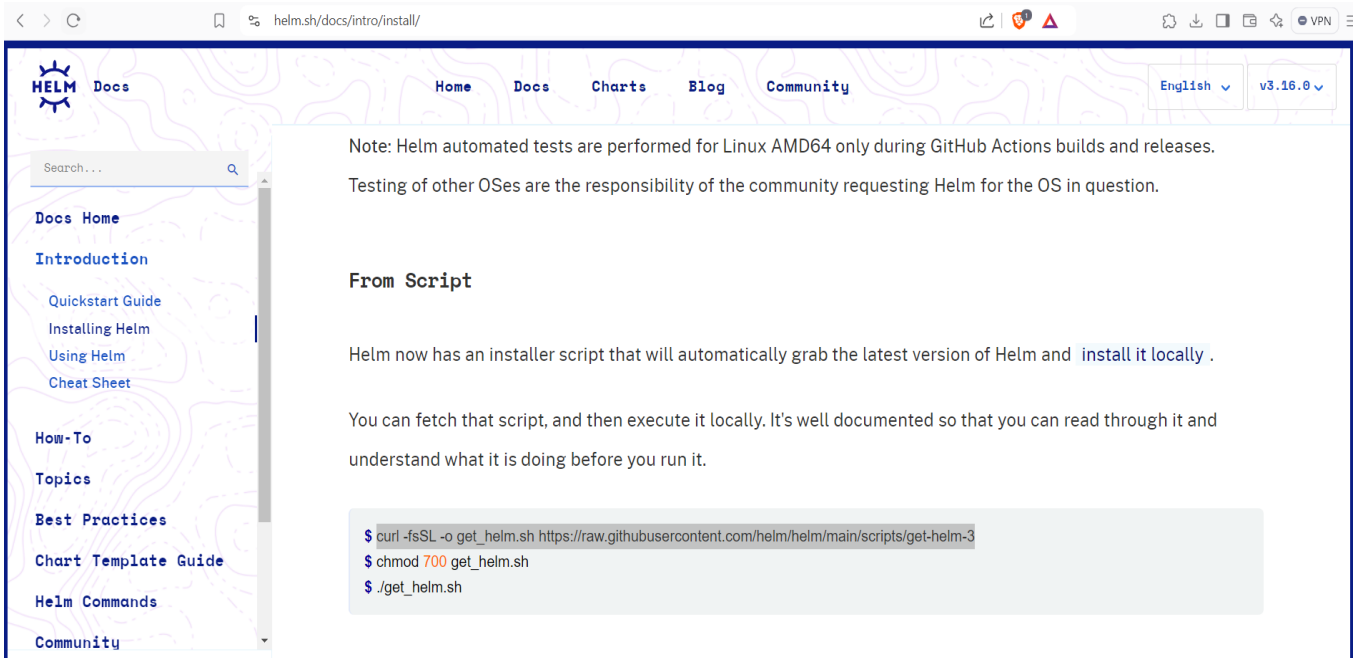
HELM

1. Install helm using the official website of Helm in your K8s cluster.

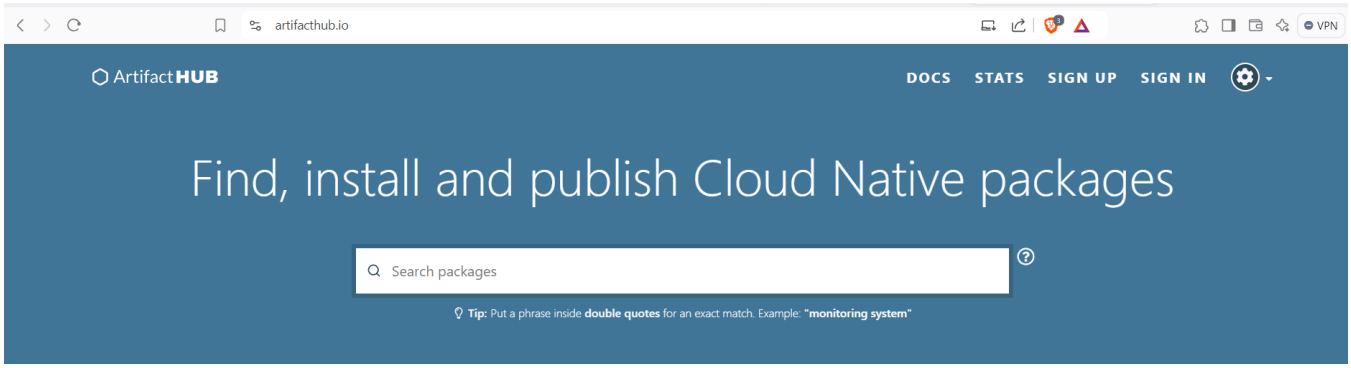


2. Validate the Helm Installation

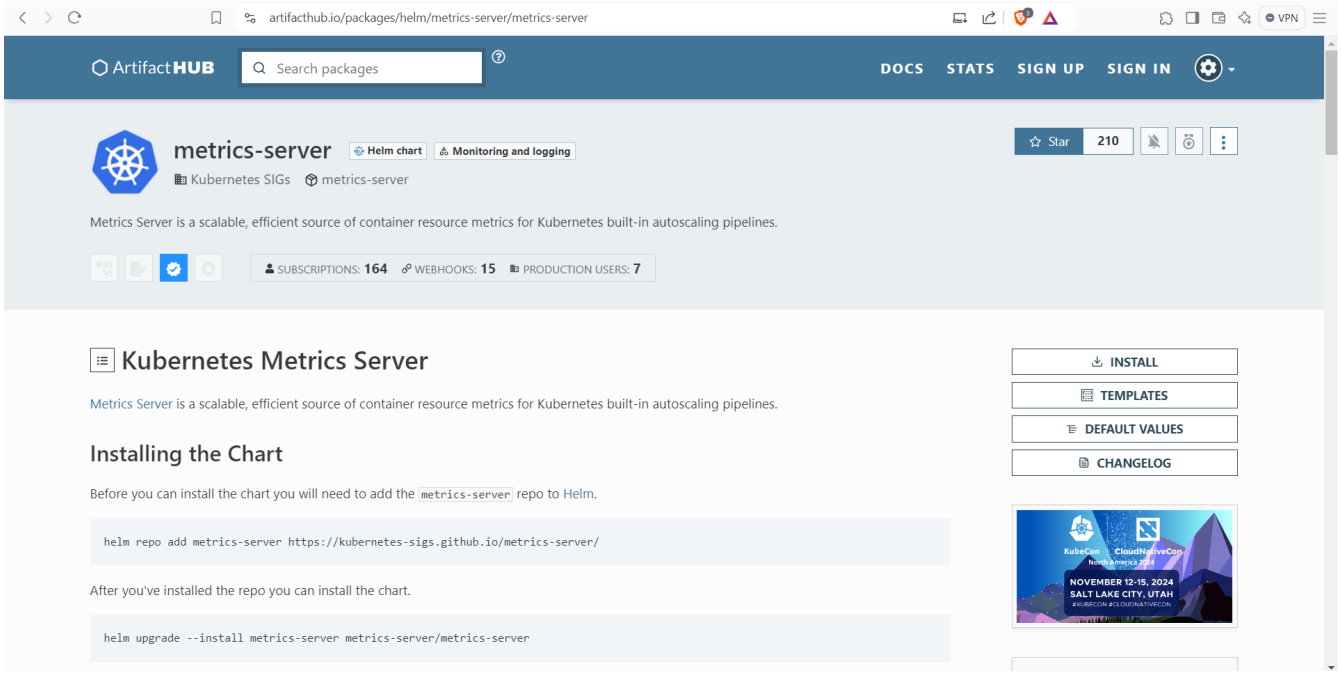
```
[ec2-user@ip-192-168-48-154 ~]$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
[ec2-user@ip-192-168-48-154 ~]$ ls
aws  awscliv2.zip  clusterautoscaler.yaml  get_helm.sh  javawebapp.yaml  kubectl  kubernetes-ingress  nginxdep.yaml  RBAC_ClusterRole.yml  rbac.yml
[ec2-user@ip-192-168-48-154 ~]$ chmod 700 get_helm.sh
[ec2-user@ip-192-168-48-154 ~]$ ./get_helm.sh
Downloading https://get.helm.sh/helm-v3.16.1-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
[ec2-user@ip-192-168-48-154 ~]$ kubectl get nodes
NAME                                            STATUS   ROLES    AGE   VERSION
ip-192-168-191-75.ap-south-1.compute.internal   Ready    <none>   12m   v1.30.4-eks-a737599
ip-192-168-252-121.ap-south-1.compute.internal  Ready    <none>   12m   v1.30.4-eks-a737599
[ec2-user@ip-192-168-48-154 ~]$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
javawebappdeployment-84cfd97467-7vhc6   1/1     Running   0          10h
javawebappdeployment-84cfd97467-tppbt   1/1     Running   0          10h
nginxdeployment-6c94fcfc47-jnl8l        1/1     Running   0          10h
nginxdeployment-6c94fcfc47-l5z7l        1/1     Running   0          10h
[ec2-user@ip-192-168-48-154 ~]$ helm
The Kubernetes package manager

Common actions for Helm:

- helm search:    search for charts
- helm pull:      download a chart to your local directory to view
- helm install:   upload the chart to Kubernetes
- helm list:      list releases of charts

Environment variables:

| Name                                | Description                                                                                                                |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| $HELM_CACHE_HOME                    | set an alternative location for storing cached files.                                                                      |
| $HELM_CONFIG_HOME                   | set an alternative location for storing Helm configuration.                                                                |
| $HELM_DATA_HOME                     | set an alternative location for storing Helm data.                                                                         |
| $HELM_DEBUG                         | indicate whether or not Helm is running in Debug mode                                                                      |
| $HELM_DRIVER                        | set the backend storage driver. Values are: configmap, secret, memory, sql.                                                |
| $HELM_DRIVER_SQL_CONNECTION_STRING  | set the connection string the SQL storage driver should use.                                                               |
| $HELM_MAX_HISTORY                   | set the maximum number of helm release history.                                                                            |
| $HELM_NAMESPACE                     | set the namespace used for the helm operations.                                                                            |
```

```
[ec2-user@ip-192-168-48-154 ~]$ helm version
version.BuildInfo{Version:"v3.16.1", GitCommit:"5a5449dc42be07001fd5771d56429132984ab3ab", GitTreeState:"clean", GoVersion:"go1.22.7"}
[ec2-user@ip-192-168-48-154 ~]$
```

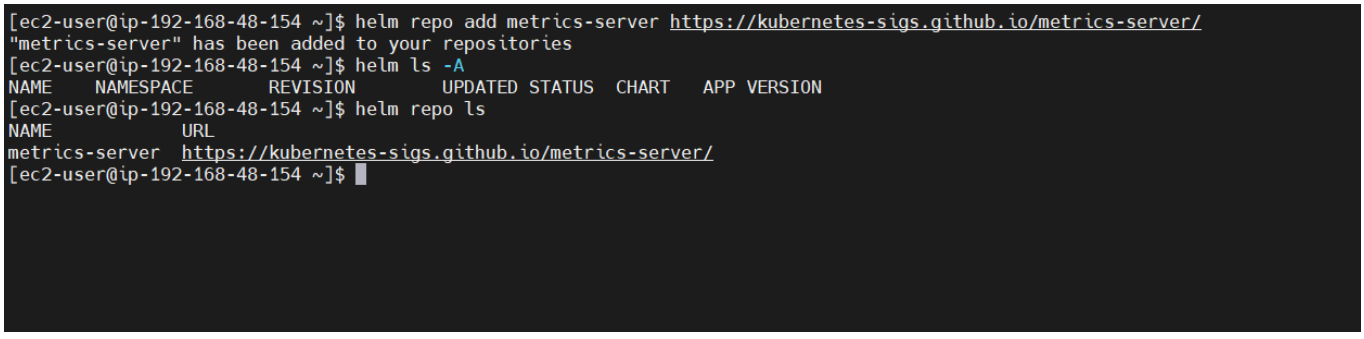
3. Like we have dockerhub for docker , similarly we have https://artifacthub.io for Helm.



4. We will install metric-server using Helm.



5. Repo added locally



6. Now, will install the chart using the below hem install command.

helm upgrade --install metrics-server metrics-server/metrics-server -n kube-system

```
[ec2-user@ip-192-168-48-154 ~]$ helm upgrade --install metrics-server metrics-server/metrics-server -n kube-system
Release "metrics-server" does not exist. Installing it now.
NAME: metrics-server
LAST DEPLOYED: Sun Oct  6 09:15:35 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
**********************************************************************
* Metrics Server                                                     *
**********************************************************************
  Chart version: 3.12.1
  App version:   0.7.1
  Image tag:     registry.k8s.io/metrics-server/metrics-server:v0.7.1
**********************************************************************
[ec2-user@ip-192-168-48-154 ~]$ kubectl get all -n kube-system
NAME                                  READY   STATUS    RESTARTS   AGE
pod/aws-node-lrnqc                    2/2     Running   0          35m
pod/aws-node-tmp5z                    2/2     Running   0          35m
pod/coredns-6c55b85fbb-ct76m          1/1     Running   0          11h
pod/coredns-6c55b85fbb-pjt8b          1/1     Running   0          10h
pod/kube-proxy-qrlxr                  1/1     Running   0          35m
pod/kube-proxy-vzfqt                  1/1     Running   0          35m
pod/metrics-server-7998667b79-rdxjl   1/1     Running   0          42s

NAME                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                  AGE
service/kube-dns         ClusterIP   10.100.0.10      <none>        53/UDP,53/TCP,9153/TCP   3d1h
service/metrics-server   ClusterIP   10.100.125.136   <none>        443/TCP                  42s

NAME                        DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/aws-node     2         2         2       2            2           <none>          3d1h
daemonset.apps/kube-proxy   2         2         2       2            2           <none>          3d1h

NAME                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/coredns          2/2     2            2           3d1h
deployment.apps/metrics-server   1/1     1            1           42s

NAME                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/coredns-6c55b85fbb          2         2         2       3d1h
replicaset.apps/metrics-server-7998667b79   1         1         1       42s
[ec2-user@ip-192-168-48-154 ~]$
```

```
[ec2-user@ip-192-168-48-154 ~]$ helm ls -n kube-system
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
metrics-server  kube-system     1               2024-10-06 09:15:35.038137807 +0000 UTC deployed        metrics-server-3.12.1   0.7.1
[ec2-user@ip-192-168-48-154 ~]$
```

7. HPA metrics-server working as expected.

```
ec2-user@ip-192-168-48-154 ~]$ kubectl top nodes
AME                                            CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
p-192-168-191-75.ap-south-1.compute.internal    48m          2%     691Mi           9%
p-192-168-252-121.ap-south-1.compute.internal   49m          2%     686Mi           9%
ec2-user@ip-192-168-48-154 ~]$ kubectl top pods
AME                                    CPU(cores)   MEMORY(bytes)
avawebappdeployment-84cfd97467-7vhc6   2m           140Mi
avawebappdeployment-84cfd97467-tppbt   3m           149Mi
ginxdeployment-6c94fcfc47-jnl8l        0m           2Mi
ginxdeployment-6c94fcfc47-l5z7l        0m           2Mi
ec2-user@ip-192-168-48-154 ~]$
```

8. Now, will increase the replicas using --set replicas

   helm upgrade --install metrics-server metrics-server/metrics-server -n kube-system --set replicas=2

```
[ec2-user@ip-192-168-48-154 ~]$ helm upgrade --install metrics-server metrics-server/metrics-server -n kube-system --set replicas=2
Release "metrics-server" has been upgraded. Happy Helming!
NAME: metrics-server
LAST DEPLOYED: Sun Oct  6 09:32:38 2024
NAMESPACE: kube-system
STATUS: deployed
REVISION: 2
TEST SUITE: None
NOTES:
**********************************************************************
* Metrics Server                                                     *
**********************************************************************
  Chart version: 3.12.1
  App version:   0.7.1
  Image tag:     registry.k8s.io/metrics-server/metrics-server:v0.7.1
**********************************************************************
[ec2-user@ip-192-168-48-154 ~]$ kubectl get pods -n kube-system
NAME                              READY   STATUS    RESTARTS   AGE
aws-node-lrnqc                    2/2     Running   0          52m
aws-node-tmp5z                    2/2     Running   0          52m
coredns-6c55b85fbb-ct76m          1/1     Running   0          11h
coredns-6c55b85fbb-pjt8b          1/1     Running   0          11h
kube-proxy-qrlxr                  1/1     Running   0          52m
kube-proxy-vzfqt                  1/1     Running   0          52m
metrics-server-7998667b79-rdxjl   1/1     Running   0          17m
metrics-server-7998667b79-tntfq   1/1     Running   0          32s
[ec2-user@ip-192-168-48-154 ~]$
```

9. We can the revision after the increasing the replicas of metrics-server pod.

```
[ec2-user@ip-192-168-48-154 ~]$ helm ls -n kube-system
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
metrics-server  kube-system     2               2024-10-06 09:32:38.434780774 +0000 UTC deployed        metrics-server-3.12.1   0.7.1
[ec2-user@ip-192-168-48-154 ~]$
[ec2-user@ip-192-168-48-154 ~]$
```

10. In case something is not working we can rollback to the previous version.

    helm rollback metrics-server -n kube-system

```
[ec2-user@ip-192-168-48-154 ~]$ helm rollback metrics-server -n kube-system
Rollback was a success! Happy Helming!
[ec2-user@ip-192-168-48-154 ~]$ kubectl get pods -n kube-system
NAME                              READY   STATUS    RESTARTS   AGE
aws-node-lrnqc                    2/2     Running   0          58m
aws-node-tmp5z                    2/2     Running   0          58m
coredns-6c55b85fbb-ct76m          1/1     Running   0          11h
coredns-6c55b85fbb-pjt8b          1/1     Running   0          11h
kube-proxy-qrlxr                  1/1     Running   0          58m
kube-proxy-vzfqt                  1/1     Running   0          58m
metrics-server-7998667b79-rdxjl   1/1     Running   0          23m
[ec2-user@ip-192-168-48-154 ~]$
```

11. Revisions are saved in secrets objects of Kubernetes cluster.

```
[ec2-user@ip-192-168-48-154 ~]$ kubectl get secrets -n kube-system|grep "metrics-server"
sh.helm.release.v1.metrics-server.v1   helm.sh/release.v1   1      26m
sh.helm.release.v1.metrics-server.v2   helm.sh/release.v1   1      9m34s
sh.helm.release.v1.metrics-server.v3   helm.sh/release.v1   1      3m19s
[ec2-user@ip-192-168-48-154 ~]$
```

12. Uninstalling the chart metrics-server.

    helm uninstall metrics-server -n kube-system

```
[ec2-user@ip-192-168-48-154 ~]$ helm uninstall metrics-server -n kube-system
release "metrics-server" uninstalled
[ec2-user@ip-192-168-48-154 ~]$ kubectl get pods -n kube-system
NAME                              READY   STATUS    RESTARTS   AGE
aws-node-lrnqc                    2/2     Running   0          66m
aws-node-tmp5z                    2/2     Running   0          66m
coredns-6c55b85fbb-ct76m          1/1     Running   0          11h
coredns-6c55b85fbb-pjt8b          1/1     Running   0          11h
kube-proxy-qrlxr                  1/1     Running   0          66m
kube-proxy-vzfqt                  1/1     Running   0          66m
[ec2-user@ip-192-168-48-154 ~]$
```

Creating chart for our applications

1. Create Helm chart using below command for the application.

   helm create nginxchart
   sudo yum install tree -y → (if tree command not available – its optional)

```
ec2-user@ip-192-168-48-154 ~]$ mkdir Helm_Charts
ec2-user@ip-192-168-48-154 ~]$ cd Helm_Charts/
ec2-user@ip-192-168-48-154 Helm_Charts]$ helm create nginxchart
reating nginxchart
```

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ tree
.
└── nginxchart
    ├── charts
    ├── Chart.yaml
    ├── templates
    │   ├── deployment.yaml
    │   ├── _helpers.tpl
    │   ├── hpa.yaml
    │   ├── ingress.yaml
    │   ├── NOTES.txt
    │   ├── serviceaccount.yaml
    │   ├── service.yaml
    │   └── tests
    │       └── test-connection.yaml
    └── values.yaml

4 directories, 10 files
[ec2-user@ip-192-168-48-154 Helm_Charts]$ ls nginxchart/
charts  Chart.yaml  templates  values.yaml
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

2. Now, we need to modify the values.yaml file according to our requirement.

```
# This will set the replicaset count more information can be found here: https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/
replicaCount: 2

# This sets the name of the container
containerName: nginxcontainer
```

```
mage:
 repository: nginx
 # This sets the pull policy for images.
 pullPolicy: Always
 # Overrides the image tag whose default is the chart appVersion.
 tag: "latest"

 This is for the secretes for pulling an image from a private repository more information can be found here: https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/
magePullSecrets: []
 This is to override the chart name.
ameOverride: "nginxpod"
ullnameOverride: nginxdeployment
```

```
resources:
  # We usually recommend not to specify default resources and to leave this as a conscious
  # choice for the user. This also increases chances charts run on environments with little
  # resources, such as Minikube. If you do want to specify resources, uncomment the following
  # lines, adjust them as necessary, and remove the curly braces after 'resources:'.
  requests:
    memory: 128Mi
    cpu: 250m
  limits:
    memory: 256Mi
    cpu: 500m
```

```
# This is to setup the liveness and readiness probes more information can be found here: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/
readinessProbe:
  httpGet:
    path: /
    port: 80
  initialDelaySeconds: 5
  periodSeconds: 10
  timeoutSeconds: 2
  successThreshold: 1
  failureThreshold: 3
livenessProbe:
  httpGet:
    path: /
    port: 80
  initialDelaySeconds: 15
  periodSeconds: 20
  timeoutSeconds: 2
  successThreshold: 1
  failureThreshold: 3
```

3. Once the required changes are we can run the below to check the fine template.
   helm template nginx nginxchart/

```
# Source: nginxchart/templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginxdeployment
  labels:
    helm.sh/chart: nginxchart-0.1.0
    app.kubernetes.io/name: nginxpod
    app.kubernetes.io/instance: nginx
    app.kubernetes.io/version: "1.16.0"
    app.kubernetes.io/managed-by: Helm
spec:
  replicas: 2
  selector:
    matchLabels:
      app.kubernetes.io/name: nginxpod
      app.kubernetes.io/instance: nginx
  template:
    metadata:
      labels:
        helm.sh/chart: nginxchart-0.1.0
        app.kubernetes.io/name: nginxpod
        app.kubernetes.io/instance: nginx
        app.kubernetes.io/version: "1.16.0"
        app.kubernetes.io/managed-by: Helm
    spec:
      serviceAccountName: nginxdeployment
      containers:
        - name: nginxcontainer
          image: nginx:latest
          imagePullPolicy: Always
          ports:
            - name: http
              containerPort: 80
              protocol: TCP
          livenessProbe:
            failureThreshold: 3
            httpGet:
              path: /
              port: 80
            initialDelaySeconds: 15
            periodSeconds: 20
            successThreshold: 1
            timeoutSeconds: 2
          readinessProbe:
            failureThreshold: 3
            httpGet:
              path: /
              port: 80
            initialDelaySeconds: 5
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 2
          resources:
            limits:
              cpu: 500m
              memory: 256Mi
            requests:
```

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm template nginx nginxchart/
---
# Source: nginxchart/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginxdeployment
  labels:
    helm.sh/chart: nginxchart-0.1.0
    app.kubernetes.io/name: nginxpod
    app.kubernetes.io/instance: nginx
    app.kubernetes.io/version: "1.16.0"
    app.kubernetes.io/managed-by: Helm
automountServiceAccountToken: true
---
# Source: nginxchart/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginxdeployment
  labels:
    helm.sh/chart: nginxchart-0.1.0
    app.kubernetes.io/name: nginxpod
    app.kubernetes.io/instance: nginx
    app.kubernetes.io/version: "1.16.0"
    app.kubernetes.io/managed-by: Helm
spec:
  type: ClusterIP
  ports:
    - port: 80
      targetPort: http
      protocol: TCP
      name: http
  selector:
    app.kubernetes.io/name: nginxpod
    app.kubernetes.io/instance: nginx
---
```

4. Once all the checks are done, we can install the chart.

   helm install nginx nginxchart

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm install nginx nginxchart
NAME: nginx
LAST DEPLOYED: Sun Oct  6 13:05:43 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=nginxpod,app.kubernetes.io/instance=nginx" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(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
[ec2-user@ip-192-168-48-154 Helm_Charts]$ kubetcl get all
-bash: kubetcl: command not found
[ec2-user@ip-192-168-48-154 Helm_Charts]$ kubectl get all
NAME                                        READY   STATUS    RESTARTS   AGE
pod/javawebappdeployment-84cfd97467-7vhc6   1/1     Running   0          14h
pod/javawebappdeployment-84cfd97467-tppbt   1/1     Running   0          14h
pod/nginxdeployment-58d78b5dcb-5cbdl        1/1     Running   0          25s
pod/nginxdeployment-58d78b5dcb-ztltz        1/1     Running   0          25s

NAME                        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/javawebappsvc       ClusterIP   10.100.137.72   <none>        80/TCP    27h
service/kubernetes          ClusterIP   10.100.0.1      <none>        443/TCP   3d5h
service/nginxdeployment     ClusterIP   10.100.2.57     <none>        80/TCP    25s

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/javawebappdeployment   2/2     2            2           27h
deployment.apps/nginxdeployment        2/2     2            2           25s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/javawebappdeployment-6d5bb44f96   0         0         0       27h
replicaset.apps/javawebappdeployment-84cfd97467   2         2         2       27h
replicaset.apps/nginxdeployment-58d78b5dcb        2         2         2       25s
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

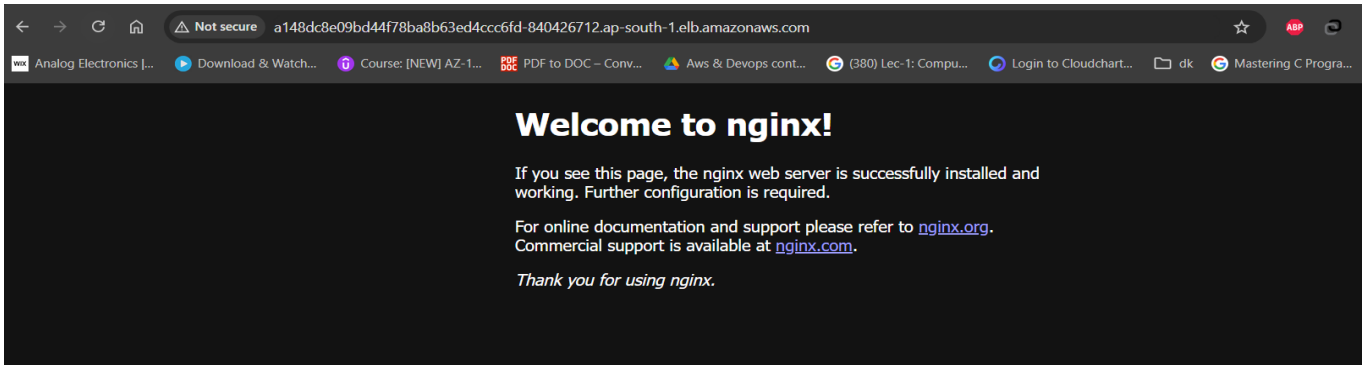
5. I did a change in the service from ClusterIP changed to LoadBalancer.

```
# This is for setting up a service more information can be found here: https://kubernetes.io/docs/concepts/services-networking/service/
service:
  # This sets the service type more information can be found here: https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types
  type: LoadBalancer
  # This sets the ports more information can be found here: https://kubernetes.io/docs/concepts/services-networking/service/#field-spec-ports
  port: 80
```

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm upgrade --install nginx nginxchart/
Release "nginx" has been upgraded. Happy Helming!
NAME: nginx
LAST DEPLOYED: Sun Oct  6 13:30:10 2024
NAMESPACE: default
STATUS: deployed
REVISION: 2
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
     NOTE: It may take a few minutes for the LoadBalancer IP to be available.
           You can watch its status by running 'kubectl get --namespace default svc -w nginxdeployment'
  export SERVICE_IP=$(kubectl get svc --namespace default nginxdeployment --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
  echo http://$SERVICE_IP:80
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm ls
NAME    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
nginx   default         2               2024-10-06 13:30:10.188645981 +0000 UTC deployed        nginxchart-0.1.0        1.16.0
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ export SERVICE_IP=$(kubectl get svc --namespace default nginxdeployment --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
[ec2-user@ip-192-168-48-154 Helm_Charts]$ echo http://$SERVICE_IP:80
http://a148dc8e09bd44f78ba8b63ed4ccc6fd-840426712.ap-south-1.elb.amazonaws.com:80
[ec2-user@ip-192-168-48-154 Helm_Charts]$ kubectl get svc
NAME            TYPE          CLUSTER-IP      EXTERNAL-IP                                                               PORT(S)       AGE
javawebappsvc   ClusterIP     10.100.137.72   <none>                                                                    80/TCP        27h
kubernetes      ClusterIP     10.100.0.1      <none>                                                                    443/TCP       3d6h
nginxdeployment LoadBalancer  10.100.2.57     a148dc8e09bd44f78ba8b63ed4ccc6fd-840426712.ap-south-1.elb.amazonaws.com   80:31058/TCP  27m
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

export SERVICE_IP=$(kubectl get svc --namespace default nginxdeployment --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
$ echo http://$SERVICE_IP:80
http://a148dc8e09bd44f78ba8b63ed4ccc6fd-840426712.ap-south-1.elb.amazonaws.com:80



6. To rollback to previous version we can use the below command.
   Usage:
   helm rollback <RELEASE> [REVISION] [flags]

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm ls
NAME    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
nginx   default         2               2024-10-06 13:30:10.188645981 +0000 UTC deployed        nginxchart-0.1.0        1.16.0
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm rollback nginx 1
Rollback was a success! Happy Helming!
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

   In Revision 1 we had ClusterIp for the service instead of LoadBalancer

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ kubectl get svc
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
javawebappsvc     ClusterIP   10.100.137.72   <none>        80/TCP    28h
kubernetes        ClusterIP   10.100.0.1      <none>        443/TCP   3d6h
nginxdeployment   ClusterIP   10.100.2.57     <none>        80/TCP    35m
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

7. To uninstall the chart

   Usage:
   helm uninstall RELEASE_NAME [...] [flags]

   Ex -helm uninstall nginx

8. To list the repo in helm

   Usage: helm repo ls

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm repo ls
NAME            URL
metrics-server  https://kubernetes-sigs.github.io/metrics-server/
```

9. To search repo
   Usage:
   helm search [command]

   Available Commands:
   hub search for charts in the Artifact Hub or your own hub instance
   repo search repositories for a keyword in charts

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm search repo metrics-server
NAME                        CHART VERSION   APP VERSION     DESCRIPTION
metrics-server/metrics-server   3.12.1          0.7.1           Metrics Server is a scalable, efficient source ...
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

10. To Render chart templates locally and display the output.
    Usage:
    helm template [NAME] [CHART] [flags]

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm template metrics-server metrics-server/metrics-server
---
# Source: metrics-server/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: metrics-server
  namespace: default
  labels:
    helm.sh/chart: metrics-server-3.12.1
    app.kubernetes.io/name: metrics-server
    app.kubernetes.io/instance: metrics-server
    app.kubernetes.io/version: "0.7.1"
    app.kubernetes.io/managed-by: Helm
---
# Source: metrics-server/templates/clusterrole-aggregated-reader.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: system:metrics-server-aggregated-reader
  labels:
    helm.sh/chart: metrics-server-3.12.1
    app.kubernetes.io/name: metrics-server
    app.kubernetes.io/instance: metrics-server
    app.kubernetes.io/version: "0.7.1"
    app.kubernetes.io/managed-by: Helm
    rbac.authorization.k8s.io/aggregate-to-admin: "true"
    rbac.authorization.k8s.io/aggregate-to-edit: "true"
    rbac.authorization.k8s.io/aggregate-to-view: "true"
rules:
  - apiGroups:
      - metrics.k8s.io
    resources:
      - pods
      - nodes
    verbs:
      - get
      - list
      - watch
---
```

11. Helm show values command

Commands

1. **Show Default Values**: To see the default values for the `metrics-server` Helm chart, use:

bash Copy code

```
helm show values metrics-server/metrics-server
```

2. **Show Chart Information**: To get metadata about the chart, including its version and description:

```bash
helm show chart metrics-server/metrics-server
```

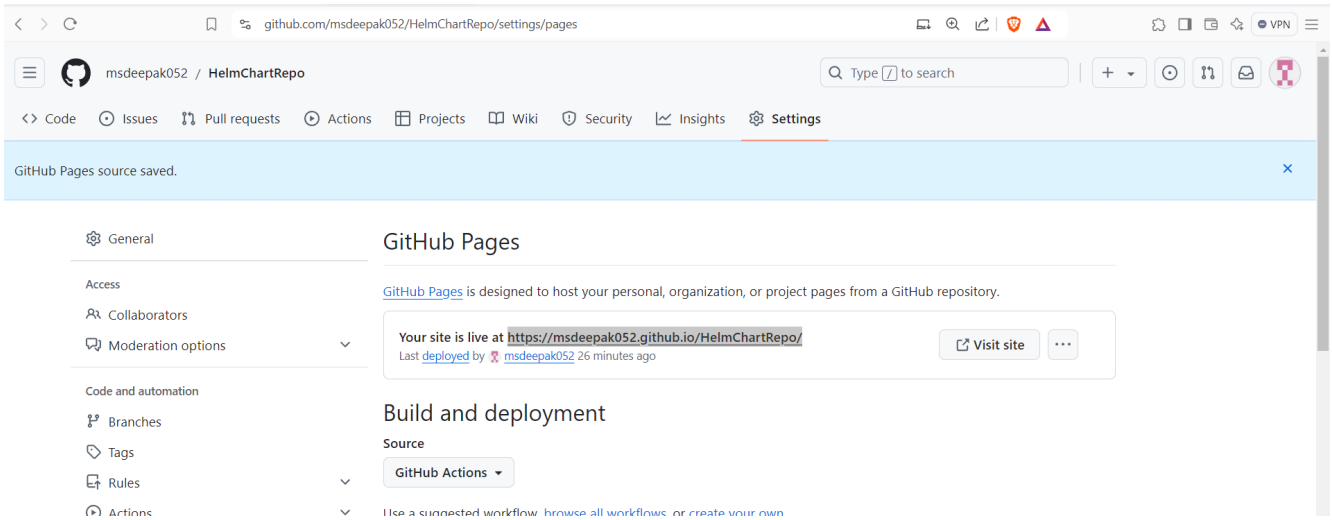
3. **Show Readme**: To view the README file for the `metrics-server` Helm chart, which typically includes installation instructions, configuration options, and usage examples:

```bash
helm show readme metrics-server/metrics-server
```

4. **Show All Information**: To show all information about the chart (values, chart, and readme), you can use:

```bash
helm show all metrics-server/metrics-server
```

12. Integrating with Github for managing the Helm charts.



13. Helm lint & package command

a. Helm lint

This command takes a path to a chart and runs a series of tests to verify that the chart is well-formed.

If the linter encounters things that will cause the chart to fail installation, it will emit [ERROR] messages. If it encounters issues that break with convention or recommendation, it will emit [WARNING] messages.

Usage:

helm lint PATH [flags]

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm lint nginxchart/
==> Linting nginxchart/
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, 0 chart(s) failed
```

b. helm package

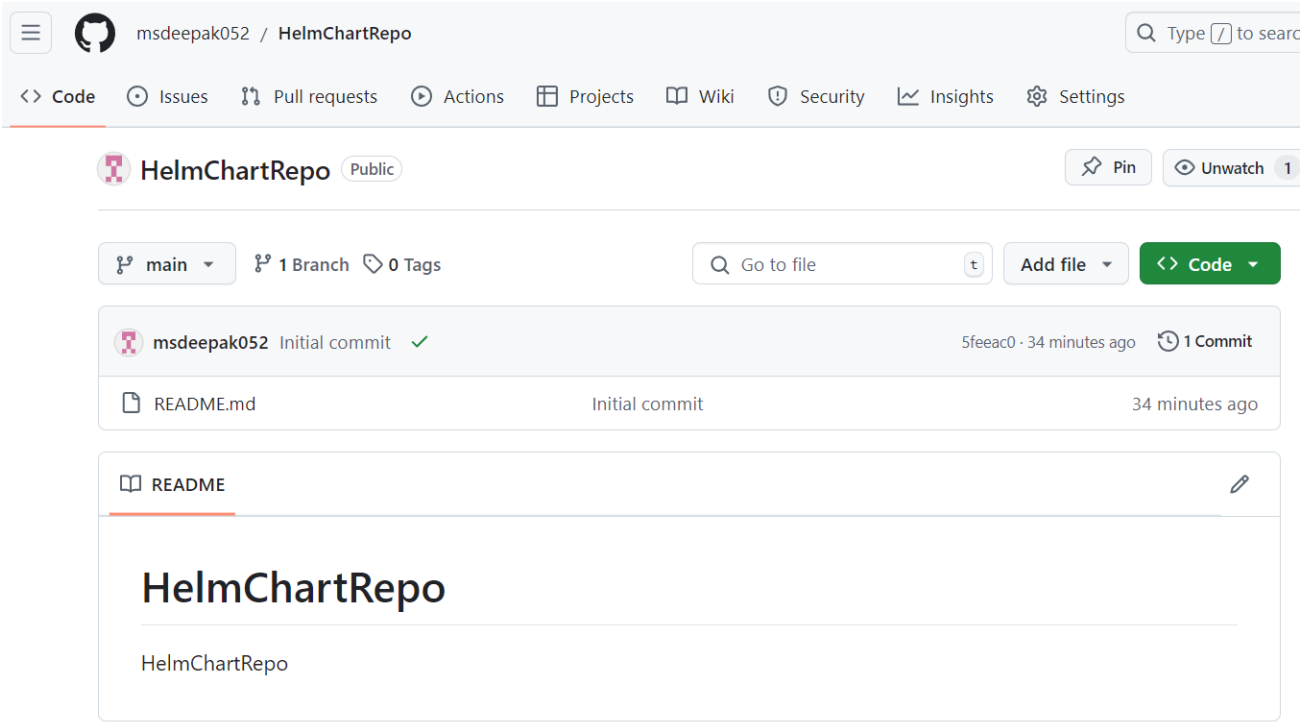
This command packages a chart into a versioned chart archive file. If a path is given, this will look at that path for a chart (which must contain a Chart.yaml file) and then package that directory.

Versioned chart archives are used by Helm package repositories.
Usage:
helm package [CHART_PATH] [...] [flags]

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm package nginxchart
Successfully packaged chart and saved it to: /home/ec2-user/Helm_Charts/nginxchart-0.1.0.tgz
[ec2-user@ip-192-168-48-154 Helm_Charts]$ ls
nginxchart  nginxchart-0.1.0.tgz
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

14. Clone the repo

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ git clone https://github.com/msdeepak052/HelmChartRepo.git
Cloning into 'HelmChartRepo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```



15. Move the tgz file inside the HelmChartRepo

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ ls
HelmChartRepo  nginxchart  nginxchart-0.1.0.tgz
[ec2-user@ip-192-168-48-154 Helm_Charts]$ mv nginxchart-0.1.0.tgz HelmChartRepo/
[ec2-user@ip-192-168-48-154 Helm_Charts]$ ls HelmChartRepo/
nginxchart-0.1.0.tgz  README.md
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

16. Run the below command
    helm repo index HelmChartRepo --url https://msdeepak052.github.io/HelmChartRepo/

    Note: Above URL(Repo) is GitHub Page Link Which We can get from Git Hub Pages of our repo.

```
ec2-user@ip-192-168-48-154 Helm_Charts]$ ls
elmChartRepo  nginxchart
ec2-user@ip-192-168-48-154 Helm_Charts]$ helm repo index HelmChartRepo --url https://msdeepak052.github.io/HelmChartRepo/
ec2-user@ip-192-168-48-154 Helm_Charts]$ cd HelmChartRepo/
ec2-user@ip-192-168-48-154 HelmChartRepo]$ ls
ndex.yaml  nginxchart-0.1.0.tgz  README.md
ec2-user@ip-192-168-48-154 HelmChartRepo]$
```

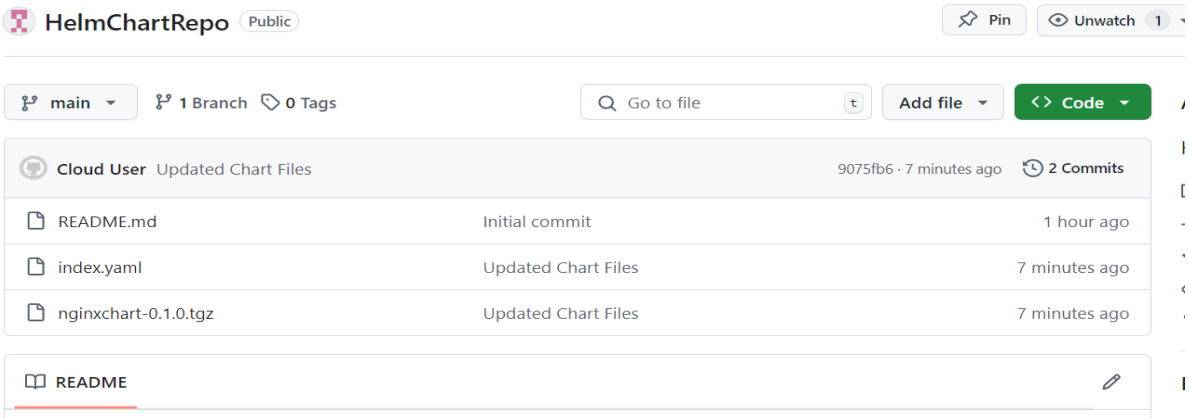
```
[ec2-user@ip-192-168-48-154 HelmChartRepo]$ cd ..
[ec2-user@ip-192-168-48-154 Helm_Charts]$ cat HelmChartRepo/index.yaml
apiVersion: v1
entries:
  nginxchart:
  - apiVersion: v2
    appVersion: 1.16.0
    created: "2024-10-06T16:36:56.335484668Z"
    description: A Helm chart for Kubernetes
    digest: 3be24686cb10d37fcb3caf486870db610c25cb100062e40a6c4681ac326d1b2d
    name: nginxchart
    type: application
    urls:
    - https://msdeepak052.github.io/HelmChartRepo/nginxchart-0.1.0.tgz
    version: 0.1.0
generated: "2024-10-06T16:36:56.334620538Z"
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

17. Perform the below git command.

    $ cd HelmChartRepo
    $ git add .
    $ git commit -a -m "Updated Chart Files"
    $ git push origin

```
ec2-user@ip-192-168-48-154 Helm_Charts]$ cd HelmChartRepo
ec2-user@ip-192-168-48-154 HelmChartRepo]$ git init
Reinitialized existing Git repository in /home/ec2-user/Helm_Charts/HelmChartRepo/.git/
ec2-user@ip-192-168-48-154 HelmChartRepo]$ git add .
ec2-user@ip-192-168-48-154 HelmChartRepo]$ git commit -a -m "Updated Chart Files"
main 9075fb6] Updated Chart Files
 Committer: Cloud User <ec2-user@ip-192-168-48-154.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 14 insertions(+)
 create mode 100644 index.yaml
 create mode 100644 nginxchart-0.1.0.tgz
ec2-user@ip-192-168-48-154 HelmChartRepo]$ git push origin
Username for 'https://github.com': yadav.deepak012@gmail.com
Password for 'https://yadav.deepak012@gmail.com@github.com':
```

```
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 4.79 KiB | 4.79 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/msdeepak052/HelmChartRepo.git
   5feeac0..9075fb6  main -> main
[ec2-user@ip-192-168-48-154 HelmChartRepo]$
```



18. Add the repo to the helm

    helm repo add HelmChartRepo https://msdeepak052.github.io/HelmChartRepo/

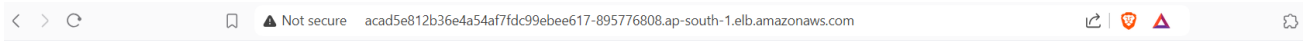
```
[ec2-user@ip-192-168-48-154 HelmChartRepo]$ cd ..
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm repo add HelmChartRepo https://msdeepak052.github.io/HelmChartRepo/
"HelmChartRepo" has been added to your repositories
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm repo ls
NAME            URL
metrics-server  https://kubernetes-sigs.github.io/metrics-server/
HelmChartRepo   https://msdeepak052.github.io/HelmChartRepo/
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

19. Once the helm charts are added in the repo we can install the k8s resources using helm install

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm repo ls
NAME            URL
metrics-server  https://kubernetes-sigs.github.io/metrics-server/
HelmChartRepo   https://msdeepak052.github.io/HelmChartRepo/
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm search repo
NAME                            CHART VERSION   APP VERSION     DESCRIPTION
HelmChartRepo/nginxchart        0.1.0           1.16.0          A Helm chart for Kubernetes
metrics-server/metrics-server   3.12.1          0.7.1           Metrics Server is a scalable, efficient source ...
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm ls
NAME    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
nginx   default         3               2024-10-06 13:40:17.078157246 +0000 UTC deployed        nginxchart-0.1.0        1.16.0
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm uninstall nginx
release "nginx" uninstalled
[ec2-user@ip-192-168-48-154 Helm_Charts]$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
javawebappdeployment-84cfd97467-7vhc6   1/1     Running   0          19h
javawebappdeployment-84cfd97467-tppbt   1/1     Running   0          19h
[ec2-user@ip-192-168-48-154 Helm_Charts]$ helm upgrade --install nginx HelmChartRepo/nginxchart
Release "nginx" does not exist. Installing it now.
NAME: nginx
LAST DEPLOYED: Sun Oct  6 17:24:02 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
     NOTE: It may take a few minutes for the LoadBalancer IP to be available.
           You can watch its status by running 'kubectl get --namespace default svc -w nginxdeployment'
  export SERVICE_IP=$(kubectl get svc --namespace default nginxdeployment --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
  echo http://$SERVICE_IP:80
[ec2-user@ip-192-168-48-154 Helm_Charts]$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
javawebappdeployment-84cfd97467-7vhc6   1/1     Running   0          19h
javawebappdeployment-84cfd97467-tppbt   1/1     Running   0          19h
nginxdeployment-58d78b5dcb-ds7dv        0/1     Running   0          8s
nginxdeployment-58d78b5dcb-m94pw        0/1     Running   0          8s
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```

```
[ec2-user@ip-192-168-48-154 Helm_Charts]$ kubectl get all
NAME                                        READY   STATUS    RESTARTS   AGE
pod/javawebappdeployment-84cfd97467-7vhc6   1/1     Running   0          19h
pod/javawebappdeployment-84cfd97467-tppbt   1/1     Running   0          19h
pod/nginxdeployment-58d78b5dcb-ds7dv        1/1     Running   0          116s
pod/nginxdeployment-58d78b5dcb-m94pw        1/1     Running   0          116s

NAME                        TYPE           CLUSTER-IP       EXTERNAL-IP                                                               PORT(S)        AGE
service/javawebappsvc       ClusterIP      10.100.137.72    <none>                                                                    80/TCP         31h
service/kubernetes          ClusterIP      10.100.0.1       <none>                                                                    443/TCP        3d10h
service/nginxdeployment     LoadBalancer   10.100.213.193   acad5e812b36e4a54af7fdc99ebee617-895776808.ap-south-1.elb.amazonaws.com   80:31521/TCP   116s

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/javawebappdeployment   2/2     2            2           31h
deployment.apps/nginxdeployment        2/2     2            2           116s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/javawebappdeployment-6d5bb44f96   0         0         0       31h
replicaset.apps/javawebappdeployment-84cfd97467   2         2         2       31h
replicaset.apps/nginxdeployment-58d78b5dcb        2         2         2       116s
[ec2-user@ip-192-168-48-154 Helm_Charts]$
```



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.