Application Packaging & Deployment with Helm

Install Helm on the VM

```
tlxp@ip-172-31-35-46:~$ curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11929  100 11929    0     0   141k      0 --:--:-- --:--:-- --:--:--  142k
Downloading https://get.helm.sh/helm-v3.19.4-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
[sudo] password for tlxp:
helm installed into /usr/local/bin/helm
tlxp@ip-172-31-35-46:~$ helm version
version.BuildInfo{Version:"v3.19.4", GitCommit:"7cfb6e486dac026202556836bb910c37d847793e", GitTreeState:"clean", GoVersion:"go1.24.1
1"}
```

Create a Helm Chart

```
tlxp@ip-172-31-35-46:~$ helm create my-5gc-nginx
Creating my-5gc-nginx
```

Customize values.yaml

```
tlxp@ip-172-31-35-46:~$ cd my-5gc-nginx
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ nano values.yaml
```

```
replicaCount: 2
```

```
  #    - ALL
  # readOnlyRootFilesyst
  # runAsNonRoot: true
  # runAsUser: 1000

# This is for setting up
service:
  # This sets the servic
  type: NodePort
  # This sets the ports
  port: 80
```

Update Deployment Template

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ cd templates
tlxp@ip-172-31-35-46:~/my-5gc-nginx/templates$ nano deployment.yaml
```

```
    containers:
      - name: {{ nginx }}
        {{- with .Values.securityContext }}
        securityContext:
          {{- toYaml . | nindent 12 }}
        {{- end }}
        image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
        imagePullPolicy: {{ .Values.image.pullPolicy }}
        ports:
          - name: http
            containerPort: 80
```

Install the Helm Chart

```
tlxp@ip-172-31-35-46:~$ cd my-5gc-nginx
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ helm install my-5gc-release .
NAME: my-5gc-release
LAST DEPLOYED: Wed Dec 31 08:59:53 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services my-5gc-release-my-5gc-nginx)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
```

Verify Kubernetes resources

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ kubectl get pods
kubectl get svc
NAME                                        READY   STATUS             RESTARTS   AGE
my-5gc-pod                                  1/1     Running            0          101m
my-5gc-release-my-5gc-nginx-7df8f84767-pmnkd   0/1     InvalidImageName   0          32s
my-5gc-release-my-5gc-nginx-7df8f84767-rd8f2   0/1     InvalidImageName   0          32s
nginx-deploy-77bf8679f9-tt77g               1/1     Running            0          115m
nginx-deploy-77bf8679f9-xnrwb               1/1     Running            0          115m
NAME                          TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)        AGE
kubernetes                    ClusterIP      10.96.0.1        <none>           443/TCP        168m
my-5gc-externalname           ExternalName   <none>           telcolearn.com   <none>         148m
my-5gc-release-my-5gc-nginx   NodePort       10.106.167.163   <none>           80:30899/TCP   32s
my-5gc-svc-clusterip          ClusterIP      10.105.135.162   <none>           80/TCP         151m
my-5gc-svc-lb                 LoadBalancer   10.102.218.162   <pending>        80:31260/TCP   150m
my-5gc-svc-nodeport           NodePort       10.109.173.108   <none>           80:31419/TCP   150m
nginx-service                 ClusterIP      10.100.53.63     <none>           80/TCP         111m
```

Upgrade the Application (Rolling Update)

```
# This will set the re
replicaCount: 3
```

Upgrade the release

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ helm upgrade my-5gc-release .
Release "my-5gc-release" has been upgraded. Happy Helming!
NAME: my-5gc-release
LAST DEPLOYED: Wed Dec 31 09:03:04 2025
NAMESPACE: default
STATUS: deployed
REVISION: 2
NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].nodePort}" services my-5gc-release-my-5gc-nginx)
  export NODE_IP=$(kubectl get nodes --namespace default -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
```

Verify upgrade

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ kubectl get pods
NAME                                        READY   STATUS             RESTARTS   AGE
my-5gc-pod                                  1/1     Running            0          104m
my-5gc-release-my-5gc-nginx-7df8f84767-pmnkd   0/1     InvalidImageName   0          4m24s
my-5gc-release-my-5gc-nginx-7df8f84767-rd8f2   0/1     InvalidImageName   0          4m24s
my-5gc-release-my-5gc-nginx-7df8f84767-scnx6   0/1     InvalidImageName   0          73s
nginx-deploy-77bf8679f9-tt77g               1/1     Running            0          119m
nginx-deploy-77bf8679f9-xnrwb               1/1     Running            0          119m
```

Rollback

Check release history

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ helm history my-5gc-release
REVISION        UPDATED                         STATUS          CHART                   APP VERSION     DESCRIPTION
1               Wed Dec 31 08:59:53 2025        superseded      my-5gc-nginx-0.1.0      1.16.0          Install complete
2               Wed Dec 31 09:03:04 2025        deployed        my-5gc-nginx-0.1.0      1.16.0          Upgrade complete
```

Rollback to revision 1

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ helm rollback my-5gc-release 1
Rollback was a success! Happy Helming!
```

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ helm rollback my-5gc-release 1
Rollback was a success! Happy Helming!
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ kubectl get pods
NAME                                        READY   STATUS             RESTARTS   AGE
my-5gc-pod                                  1/1     Running            0          109m
my-5gc-release-my-5gc-nginx-7df8f84767-pmnkd   0/1     InvalidImageName   0          8m58s
my-5gc-release-my-5gc-nginx-7df8f84767-rd8f2   0/1     InvalidImageName   0          8m58s
nginx-deploy-77bf8679f9-tt77g               1/1     Running            0          123m
nginx-deploy-77bf8679f9-xnrwb               1/1     Running            0          123m
tlxp@ip-172-31-35-46:~/my-5gc-nginx$
```

Uninstall the Application

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ helm uninstall my-5gc-release
release "my-5gc-release" uninstalled
```

```
tlxp@ip-172-31-35-46:~/my-5gc-nginx$ kubectl get pods
kubectl get svc
NAME                          READY   STATUS    RESTARTS   AGE
my-5gc-pod                    1/1     Running   0          110m
nginx-deploy-77bf8679f9-tt77g   1/1     Running   0          125m
nginx-deploy-77bf8679f9-xnrwb   1/1     Running   0          125m
NAME                  TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)          AGE
kubernetes            ClusterIP      10.96.0.1        <none>           443/TCP          178m
my-5gc-externalname   ExternalName   <none>           telcolearn.com   <none>           158m
my-5gc-svc-clusterip  ClusterIP      10.105.135.162   <none>           80/TCP           161m
my-5gc-svc-lb         LoadBalancer   10.102.218.162   <pending>        80:31260/TCP     159m
my-5gc-svc-nodeport   NodePort       10.109.173.108   <none>           80:31419/TCP     160m
nginx-service         ClusterIP      10.100.53.63     <none>           80/TCP           121m
```

In this exercise, Helm was used to package and deploy an Nginx application in Kubernetes. A Helm chart was created, and application configuration such as replica count and image version was defined in the values.yaml file. The chart was installed as a Helm release, creating Kubernetes resources automatically. The application was upgraded using helm upgrade, demonstrating rolling updates, and then rolled back to a previous version using helm rollback. Finally, the application was completely removed using helm uninstall, showcasing Helm's lifecycle management capabilities.