

Helmify

The DevOps team will use a basic application that was developed in one of the earlier labs and convert them into simple helm charts which can be easily deployed to a kubernetes cluster.

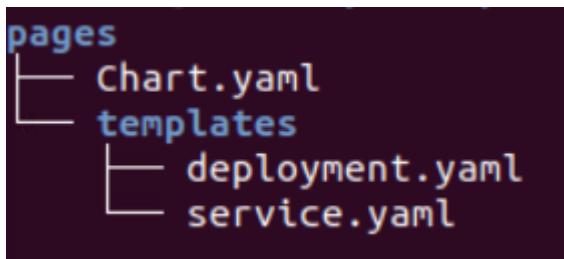
Learning Outcomes

After completing the lab, you will be able to understand

1. Chart structure
2. Helm concepts – Release, Revision, Chart Version
3. Helm commands
4. Install, Upgrade, Rollback and Delete application

Creating helm directory structure

1. Create the directory structure as below



2. Create `helm-charts` directory
3. Verify helm is installed
4. Create `templates` directory and `Chart.yaml` file.

```
mkdir ~/workspace/helm-charts
cd ~/workspace/helm-charts
helm version
```

```
mkdir pages
mkdir pages/templates
touch pages/Chart.yaml
```



5. Update `pages/Chart.yaml` containing the name of the chart with a short description, chart version and application version

`pages/Chart.yaml`

```
apiVersion: v2
name: pages
description: A Helm chart for Pages Application
type: application
version: 0.1.0
appVersion: "1.0"
```



Create the manifest files

1. Reuse the existing manifest files for deployment and service. Copy from the earlier lab or refer below

`pages/templates/deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: pages
    name: pages
spec:
  replicas: 1
  selector:
    matchLabels:
      app: pages
  strategy: {}
  template:
```

```
metadata:
  labels:
    app: pages
spec:
  containers:
    - image: [image-from-Lab 01 - Dockerize an Application]/pages:1.0
      name: pages
      imagePullPolicy: IfNotPresent
      ports:
        - containerPort: 8080
```



pages/templates/service.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: pages
  name: pages
spec:
  ports:
    - port: 8080
      protocol: TCP
      targetPort: 8080
  selector:
    app: pages
  type: NodePort
```



Deploy using helm chart

1. Before installing the helm chart create the namespace in which you want to install the chart and set the kubectl context to point to the right namespace.

```
kubectl create ns [name-of-your-team]-dev  
kubectl config set-context --current --namespace [name-of-your-team]-dev
```



2. The template command helps us to see the output of the generated manifest files before installing. Once you are satisfied with the the manifest, go ahead and install the chart in the namespace that you created.

```
helm template pages  
helm install api pages --dry-run --debug  
helm install api pages -n [name-of-your-team]-dev
```



3. Verify the installation and deployment

```
helm list  
kubectl get deploy pages  
kubectl get svc pages
```



4. Test the application running in the cluster

```
kubectl port-forward svc/pages 8080:8080  
curl localhost:8080
```



Upgrade

1. Update the name of the image in the `deployment.yaml` to `dellcloud/pages:service`

```
helm template pages  
helm upgrade api pages  
helm list  
kubectl get deploy pages  
kubectl get pods
```



2. You can notice that the deployment fails. What could be the reason?

Rollback

1. Let us rollback to the earlier version

```
helm list  
helm status api  
helm rollback api  
helm status api  
kubectl get deploy pages
```



Un-install the application

1. The devops team would want to uninstall the application

```
helm list  
helm uninstall api  
helm list  
kubectl get deploy
```

