To create a custom Helm chart for deploying an Nginx server, follow these steps:

**Prerequisites**

1. A working Kubernetes cluster.
2. Helm installed on your workstation.

**Step-by-Step Guide to Creating a Custom Helm Chart**

**1. Create a Helm Chart**

Run the following command to generate a new Helm chart named nginx-demo:

helm create nginx-demo

This will create a directory structure with default files.

**2. Customize Chart.yaml**

Edit Chart.yaml to include metadata about your chart:

yaml

apiVersion: v2

name: nginx-demo

description: My First Helm Chart

type: application

version: 0.1.0

appVersion: "1.0.0"

maintainers:

- email: amit@openwriteup.com

name: mychart

**3. Customize values.yaml**

Specify the values for deployment in values.yaml:

replicaCount: 2

image:

repository: nginx

tag: "1.16.0"

pullPolicy: IfNotPresent

service:

type: ClusterIP

port: 80

targetPort: 9000

env:

name: dev

**4. Create and Customize Template Files**

**deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: {{ .Release.Name }}-nginx

labels:

app: nginx

spec:

replicas: {{ .Values.replicaCount }}

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: {{ .Chart.Name }}

image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"

imagePullPolicy: {{ .Values.image.pullPolicy }}

ports:

- name: http

containerPort: 80

protocol: TCP

**service.yaml**

apiVersion: v1

kind: Service

metadata:

name: {{ .Release.Name }}-service

spec:

selector:

app.kubernetes.io/instance: {{ .Release.Name }}

type: {{ .Values.service.type }}

ports:

- protocol: {{ .Values.service.protocol | default "TCP" }}

port: {{ .Values.service.port }}

targetPort: {{ .Values.service.targetPort }}

**configmap.yaml**

apiVersion: v1

kind: ConfigMap

metadata:

name: {{ .Release.Name }}-html-configmap

namespace: default

data:

index.html: |

<html>

<h1>Welcome</h1>

</br>

<h1>Hi! I got deployed in {{ .Values.env.name }} Environment using Helm Chart </h1>

</html>

**Notes.txt**

Thank you for installing the nginx-demo Helm chart!

To access your Nginx server, follow these steps:

1. Get the Service URL:

The Nginx server is exposed as a ClusterIP service. To access it, you need to use port forwarding or access it within the cluster.

- Port Forwarding:

kubectl port-forward svc/{{ .Release.Name }}-service 8080:80

You can then access the server at [http://127.0.0.1:8080](http://127.0.0.1:8080).

- Access within the cluster:

The service is available at `{{ .Release.Name }}-service` on port `80` inside the cluster.

2. Check the status of the deployment:

- To view the status of the deployment, run:

kubectl get deployment -l app=nginx

- To list all the services, run:

kubectl get services -l app=nginx

- To view the pods created by the deployment, run:

kubectl get pods -l app=nginx

**3. Accessing Logs:**

**To view the logs for the Nginx server, run:**

**kubectl logs -l app=nginx**

**5. Verify the Helm Chart**

Run the following commands to verify the configuration

helm lint .

helm template .

helm install --dry-run my-nginx-release nginx-demo

**6. Deploy the Helm Chart**

Deploy the chart to your Kubernetes cluster:

helm install helm-demo nginx-demo

**7. Verify the Deployment**

Check the status of the deployment

helm list

kubectl get deployment

kubectl get services

kubectl get configmap

kubectl get pods

**8. Use External values.yaml for Different Environments**

To deploy using a specific values file:

helm install helm-demo nginx-demo --values env/prod-values.yaml

**9. Upgrade and Rollback Releases**

Upgrade the chart:

helm upgrade helm-demo nginx-demo

Rollback to a previous version:

helm rollback helm-demo <revision-number>

**10. Uninstall the Helm Release**

Uninstall the release and remove all associated resources:

helm uninstall helm-demo