<u>**Helm Chart Documentation: Nginx**</u>

HELM: A package manager for Kubernetes that allows you to package, configure, and deploy applications using Helm Charts. It simplifies the deployment of Kubernetes resources by managing collections of YAML files

HELM CHART: A Helm Chart is a bundle of Kubernetes YAML files that define an application or service. You can create your own Helm Charts and distribute them via public or private repositories or use existing ones.
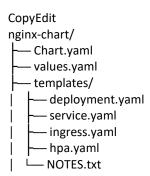
Ingress: In Kubernetes, Ingress is a resource that manages external access to services, typically HTTP. It defines rules for routing traffic to different backends based on hostnames or paths.

Overview

This Helm chart deploys an Nginx server with customizable configurations for use in Kubernetes clusters. It supports features like autoscaling, ingress configuration, and resource management.

Chart Structure

The chart includes the following files and directories:

```
CopyEdit
nginx-chart/
├── Chart.yaml
├── values.yaml
├── templates/
│   ├── deployment.yaml
│   ├── service.yaml
│   ├── ingress.yaml
│   ├── hpa.yaml
│   └── NOTES.txt
```

Files Description

- **Chart.yaml**: Contains metadata about the Helm chart.
- **values.yaml**: Default configurations for the chart.
- **templates/**: Kubernetes manifest templates for deploying resources.
    - deployment.yaml: Defines the Nginx Deployment.
    - service.yaml: Configures the Service for Nginx.
    - ingress.yaml: Sets up the Ingress resource.
    - hpa.yaml: Configures the Horizontal Pod Autoscaler.

Templates Overview

Values.yaml

- **File:** values.yaml
- Purpose: Defines default values for the Helm chart.
- Sets replica count, image details, service, ingress, and autoscaling options.

Chart.yaml

- **File:** chart.yaml
- Purpose: Contains metadata for the Helm chart.
- Defines chart name, version, and application version.

Hpa.yaml

- **File:** templates/hpa.yaml
- Purpose: Configures Horizontal Pod Autoscaling (HPA).
- Sets scaling parameters based on CPU and memory utilization.

Deployment

- **File:** templates/deployment.yaml
- Deploys the Nginx application using the specified image and replica count.
- Exposes port 80 for the container.

Service

- **File:** templates/service.yaml
- Configures the service to expose the deployment.
- Supports LoadBalancer type for external access.

Ingress

- **File:** templates/ingress.yaml
- Configures ingress rules to route traffic to the service.
- Supports custom hostnames (e.g., nginx.local).

ServiceAccount

- **File:** templates/serviceaccount.yaml
- Creates a dedicated service account for the deployment.

Steps

1. create nginx-chart
   helm create nginx-chart
2. Install the chart
   helm install my-nginx ./nginx-chart

3. List all pods
   kubectl get pod

```
PS D:\helm\nginx-chart2> kubectl get pods
NAME                             READY    STATUS    RESTARTS    AGE
my-nginx-nginx-55f9f6b49f-jlnn5   1/1     Running   0           44h
my-nginx-nginx-55f9f6b49f-mnjkt   1/1     Running   0           44h
```

4. Check the service type and external IP
   kubectl get svc my-nginx-nginx

```
PS D:\helm\nginx-chart> kubectl get svc my-nginx-nginx
NAME             TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
my-nginx-nginx   LoadBalancer   10.98.6.189   127.0.0.1     80:30080/TCP   44h
PS D:\helm\nginx-chart>
```

5. Verify the ingress controller is running
   kubectl get pods -n ingress-nginx

```
PS D:\helm\nginx-chart> kubectl get pods -n ingress-nginx
NAME                                        READY    STATUS      RESTARTS    AGE
ingress-nginx-admission-create-x7d4n        0/1     Completed   0           45h
ingress-nginx-admission-patch-4jlhm         0/1     Completed   1           45h
ingress-nginx-controller-6899bd8f5b-7xgt7   1/1     Running     0           18h
PS D:\helm\nginx-chart>
```

6. Add the host to `/etc/hosts` with the correct IP.

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.

# Added by Docker Desktop
172.20.1.29 host.docker.internal
172.20.1.29 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
127.0.0.1 nginx.local
# End of section
```

7. Access the application via the configured ingress host

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.