

作者：李晓辉

联系方式：

1. 微信：Lxh_Chat

2. 邮箱：939958092@qq.com

Helm基本概念

Helm 是 Kubernetes 的包管理工具，它能够简化 Kubernetes 应用程序的部署和管理。在 Kubernetes 中，一个应用程序可能由多个资源组成，比如 Deployment、Service、ConfigMap、Ingress 等。Helm 可以将这些资源打包成一个 Helm Chart（包），方便用户进行安装、升级、回滚和删除等操作，就和系统里的 YUM/DNF/APT非常的类似。

主要组件

1. Helm Client

- 这是用户与 Helm 交互的客户端工具。用户通过在本地上运行 Helm 命令来操作 Helm Chart，例如安装、升级、查询等操作。它会直接与 Kubernetes API 服务器交互，客户端可以直接从它的官方GitHub下载，地址是：<https://github.com/helm/helm/releases>

2. Helm Chart

- Helm Chart 是一个打包的 Kubernetes 应用程序，包含所有必要的资源定义和配置文件。它类似于软件包管理工具中的“包”，用于简化 Kubernetes 应用的部署和管理。一个 Helm Chart 包含以下部分：
 - **Chart.yaml**：
 - 定义：这是 Helm Chart 的元数据文件，包含 Chart 的基本信息。大概内容：
 - name：Chart 的名称。
 - version：Chart 的版本号（遵循语义化版本规范）。
 - description：Chart 的简要描述。
 - dependencies：依赖的其他 Chart（可选）。
 - **values.yaml**：它是 Chart 的默认配置文件。用户可以通过修改这个文件或者在安装时覆盖其中的值来定制 Chart 的行为。比如，它可能包含应用的副本数量、镜像版本、资源限制等配置项。
 - **templates/**：包含 Kubernetes 资源的模板文件，这些模板文件在安装或升级时会被渲染成实际的 Kubernetes 资源，渲染的时候，将用values.yaml里的值替换掉templates模板里的变量，例如，一个 Deployment 的模板文件可以根据 values.yaml 中定义的副本数量和镜像版本来生成对应的 Deployment 资源，具体渲染的命令是：`helm template`，可以用--help等方式获取语法，不过需要注意的是，这会渲染出实际的 Kubernetes 资源文件，但不会实际部署到集群中。

- **charts/**: 如果当前 Chart 依赖其他 Chart，这些依赖的 Chart 会存放在这个目录下。这样可以实现 Chart 之间的组合和复用。

3. Helm Repository (仓库)

- 它是一个存储 Helm Chart 的仓库，可以是本地的，也可以是远程的。远程仓库通常是一个 HTTP 服务器，用户可以从仓库中搜索、下载和更新 Helm Chart。例如，Helm 官方维护了一个默认的公共仓库，里面包含了许多常用的 Helm Chart，用户可以通过 Helm 命令将这些 Chart 添加到本地的 Chart 仓库列表中，然后进行安装等操作。

官方网址 <http://helm.sh>

Helm安装

下载安装Helm

```
wget https://get.helm.sh/helm-v3.17.3-linux-amd64.tar.gz
tar xf helm-v3.17.3-linux-amd64.tar.gz
mv linux-amd64/helm /usr/local/bin/helm
helm completion bash > /etc/bash_completion.d/helm
```

默认情况下，helm内置了一个hub，用于软件搜索和安装，搜索软件是否可被安装，用以下格式命令：

```
helm search hub Packages
```

命令行显示有点奇怪，也不够丰富，可以考虑用浏览器打开搜索：<https://hub.helm.sh>

添加仓库

官方仓库网速慢，可以考虑一下以下仓库或自己部署仓库

```
http://mirror.azure.cn/kubernetes/charts/
```

添加方式

```
helm repo add azurerepo http://mirror.azure.cn/kubernetes/charts/
```

Helm 实验案例

wordpress安装

手工定制安装

本次安装一个wordpress

```
helm search repo wordpress
```

NAME	CHART VERSION	APP VERSION	DESCRIPTION
azurerepo/wordpress	9.0.3	5.3.2	DEPRECATED Web publishing platform

ok, 发现一个wordpress包, 那这个包里有什么? 把包下出来研究研究

```
root@k8s-master:~# helm pull azurerepo/wordpress
root@k8s-master:~# ls
wordpress-9.0.3.tgz
```

下完是一个压缩包, 解压看看

```
root@k8s-master:~# tar xf wordpress-9.0.3.tgz
root@k8s-master:~# ls
wordpress  wordpress-9.0.3.tgz
root@k8s-master:~# cd wordpress/
root@k8s-master:~/wordpress# ls
charts  Chart.yaml  README.md  requirements.lock  requirements.yaml  templates  values.sche
```

果然看到了 `Chart.yaml` `templates` `values.yaml`

看看都有哪些模板

```
root@k8s-master:~/wordpress# ls templates/
deployment.yaml  externaldb-secrets.yaml  _helpers.tpl  ingress.yaml  NOTES.txt  pvc.yaml
```

模板里面都是各种变量, 稍后需要用values.yml来填充

```

root@k8s-master:~/wordpress# head templates/deployment.yaml
apiVersion: {{ template "wordpress.deployment.apiVersion" . }}
kind: Deployment
metadata:
  name: {{ template "wordpress.fullname" . }}
  labels: {{- include "wordpress.labels" . | nindent 4 }}
spec:
  selector:
    matchLabels: {{- include "wordpress.matchLabels" . | nindent 6 }}
    {{- if .Values.updateStrategy }}
  strategy: {{ toYaml .Values.updateStrategy | nindent 4 }}

```

ok, 模板看到了, 我们试试用values.yml里的变量来渲染模板, 生成最终的yaml文件

```

root@k8s-master:~/wordpress# helm template --version 9.0.3 azurerepo/wordpress

```

从仓库中读出values.yml, 将templates目录里的模板渲染成功

```

---
# Source: wordpress/charts/mariadb/templates/secrets.yaml
apiVersion: v1
kind: Secret
metadata:
  name: release-name-mariadb
  labels:
    app: "mariadb"
    chart: "mariadb-7.3.12"
    release: "release-name"
    heritage: "Helm"
type: Opaque
data:
  mariadb-root-password: "QmhiblBvMXJmZQ=="

  mariadb-password: "VWZxWm5CbnVicA=="
---
# Source: wordpress/templates/secrets.yaml
apiVersion: v1
kind: Secret

```

如果有一个值不满意怎么办? 可以手工指定某一个参数, 或者干脆用本地的values.yaml

先试试手工指定, 我们看了一下仓库里的value, 要用这个镜像, 那我来随便改一个看看

```
root@k8s-master:~/wordpress# helm show values azurerepo/wordpress | grep ^image: -A 10
image:
  registry: docker.io
  repository: bitnami/wordpress
  tag: 5.3.2-debian-10-r32
  ## Specify a imagePullPolicy
  ## Defaults to 'Always' if image tag is 'latest', else set to 'IfNotPresent'
  ## ref: http://kubernetes.io/docs/user-guide/images/#pre-pulling-images
  ##
  pullPolicy: IfNotPresent
  ## Optionally specify an array of imagePullSecrets.
  ## Secrets must be manually created in the namespace.
```

渲染时手工指定某个参数，比如替换一个镜像的registry

我通过--set参数指定了镜像的registry，并且生成了模板

```
root@k8s-master:~/wordpress# helm template --version 9.0.3 azurerepo/wordpress --set image.
WARNING: This chart is deprecated
root@k8s-master:~/wordpress# grep linuxcenter custom.yml
    image: linuxcenter.cn/bitnami/wordpress:5.3.2-debian-10-r32
    image: linuxcenter.cn/bitnami/wordpress:5.3.2-debian-10-r32
```

当然，我上面的set是瞎写的，镜像仓库不存在，只是为了演示参数而已

我们重新用默认参数渲染出最终的yaml，然后向集群部署

这一步将安装mariadb和wordpress，我们这里用的是存储章节设置的默认存储类，所以会自动创建pv以及pvc，你要是还没有默认存储类，往上翻，重新做一下存储类并标记为默认即可

```
root@k8s-master:~/wordpress# helm template --version 9.0.3 azurerepo/wordpress > lixiaohui.
WARNING: This chart is deprecated
root@k8s-master:~/wordpress# kubectl apply -f lixiaohui.yml
secret/release-name-mariadb created
secret/release-name-wordpress created
configmap/release-name-mariadb created
configmap/release-name-mariadb-tests created
persistentvolumeclaim/release-name-wordpress created
service/release-name-mariadb created
service/release-name-wordpress created
deployment.apps/release-name-wordpress created
statefulset.apps/release-name-mariadb created
pod/release-name-mariadb-test-8u7uw created
pod/release-name-credentials-test created
```

不需要看到pod工作正常，我们只研究helm自身，如果你需要它工作正常，需要搞定镜像、机器内存需要再加一下，不然数据库起不来

以上就是研究的过程，其实一般来说，不用这么麻烦，比如我们最后实际上就是用的默认value，如果默认value你就满意，可以用下面的方法直接让helm来管理，我们不用改东西，如果你并不是用的 `helm install`，后面就不受helm管理，因为当你使用 `helm template` 命令生成 Kubernetes 资源清单文件并手动将其部署到集群中时，这些资源不会被 Helm 的生命周期管理所跟踪。因此，当你运行 `helm list` 命令时，不会看到任何相关的 Helm Release，因为 Helm 的 Release 管理机制没有被触发。

helm仓库直接安装

这一步将安装mariadb和wordpress，我们这里用的是存储章节设置的默认存储类，所以会自动创建pv以及pvc，你要是还没有默认存储类，往上翻，重新做一下存储类并标记为默认即可

`helm install` 也是只是 `--set` 参数的，不一定非要用默认的值

```
root@k8s-master:~/wordpress# helm install wordpress azurerepo/wordpress
WARNING: This chart is deprecated
NAME: wordpress
LAST DEPLOYED: Thu Apr 24 13:10:09 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
```

```
root@k8s-master:~/wordpress# helm list
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS
wordpress	default	1	2025-04-24 13:10:09.708968227 +0800 CST dep	

ok，安装完毕，现在来看看这个服务怎么样了

查询服务端口

```
kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
wordpress	LoadBalancer	10.102.156.226	<pending>	80:31194/TCP,443:30386/TCP
wordpress-mariadb	ClusterIP	10.106.8.85	<none>	3306/TCP

查询pod所在节点

```
root@k8s-master:~# kubectl get pod -o wide
```

不需要看到pod工作正常，我们只研究helm自身，如果你需要它工作正常，需要搞定镜像、机器内存需要再加一下，不然数据库起不来

以上案例都是基于自己安装的k8s，我们来看一下教材上的helm操作

添加Charts仓库

helm必须要将仓库添加到本地，才能从仓库中安装软件

```
[student@workstation ~]$ helm repo add do280-repo http://helm.ocp4.example.com/charts
[student@workstation ~]$ helm repo list
```

NAME	URL
do280-repo	http://helm.ocp4.example.com/charts

搜索仓库中的helm包

如果不加--versions，同一个软件有不同版本时，只显示最新版

```
[student@workstation ~]$ helm search repo --versions
```

查询Charts基本信息与values

```
[student@workstation ~]$ helm show chart do280-repo/etherpad
apiVersion: v2
appVersion: latest
description: A Helm chart for etherpad lite
home: https://github.com/redhat-cop/helm-charts
icon: https://pbs.twimg.com/profile_images/1336377123964145665/2gTadaDt_400x400.jpg
maintainers:
- name: eformat
name: etherpad
type: application
version: 0.0.7
```

```
[student@workstation ~]$ helm show values do280-repo/etherpad
# Default values for etherpad.
replicaCount: 1

defaultTitle: "Labs Etherpad"
defaultText: "👉 Assign yourself a user and share your ideas! 👉"

image:
  repository: etherpad
  name:
  tag:
  pullPolicy: IfNotPresent

imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""

podSecurityContext: {}
securityContext: {}

service:
  type: ClusterIP
  port: 9001

ingress:
  enabled: false
  hosts:
    - name: etherpad.organization.com
  annotations: {}

route:
  enabled: true
  host: null
  targetPort: http
```

安装Helm Chart

安装helm包，需要创建values.yaml文件为软件提供参数值


```
cat > values.yaml <<-EOF
image:
  repository: registry.ocp4.example.com:8443/etherpad
  name: etherpad
  tag: 1.8.18
route:
  host: development-etherpad.apps.ocp4.example.com
EOF
```

安装0.0.6版本

```
[student@workstation ~]$ helm install lixiaohui-app do280-repo/etherpad -f values.yaml --verbose
NAME: lixiaohui-app
LAST DEPLOYED: Thu Dec 19 01:24:44 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

看看部署的效果

```
[student@workstation ~]$ helm list
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS
lixiaohui-app	default	1	2024-12-19 01:24:44.705048565 -0500 EST	deployed


```
[student@workstation ~]$ oc get route
```

NAME	HOST/PORT	PATH	SERVICES
lixiaohui	lixiaohui-default.apps.ocp4.example.com		lixiaohui
lixiaohui-app-etherpad	development-etherpad.apps.ocp4.example.com		lixiaohui-app-etherpad


```
[student@workstation ~]$ curl -s https://development-etherpad.apps.ocp4.example.com | grep
<title>Labs Etherpad</title>
```

升级Chart

helm upgrade 命令可以将更改应用到现有发行版，例如更新值或图表版本。这里我们升级到0.0.7

```
[student@workstation ~]$ helm list
NAME                NAMESPACE    REVISION    UPDATED                               STATUS
lixiaohui-app      default       1           2024-12-19 01:24:44.705048565 -0500 EST dep
[student@workstation ~]$ helm upgrade lixiaohui-app do280-repo/etherpad -f values.yaml --
Release "lixiaohui-app" has been upgraded. Happy Helming!
NAME: lixiaohui-app
LAST DEPLOYED: Thu Dec 19 01:33:01 2024
NAMESPACE: default
STATUS: deployed
REVISION: 2
TEST SUITE: None
[student@workstation ~]$ helm list
NAME                NAMESPACE    REVISION    UPDATED                               STATUS
lixiaohui-app      default       2           2024-12-19 01:33:01.298788916 -0500 EST dep
```

服务照样在线

```
[student@workstation ~]$ curl -s https://development-etherpad.apps.ocp4.example.com | grep
<title>Labs Etherpad</title>
```

回滚Chart

天有不测风云，有时候需要回退，我们回到上一版

```
[student@workstation ~]$ helm history lixiaohui-app
REVISION    UPDATED             STATUS    CHART              APP VERSION
1           Thu Dec 19 01:24:44 2024    superseded etherpad-0.0.6     latest
2           Thu Dec 19 01:33:01 2024    deployed  etherpad-0.0.7     latest
[student@workstation ~]$ helm rollback lixiaohui-app 1
Rollback was a success! Happy Helming!
```

本文档在线版本：<https://www.linuxcenter.cn>