

Helm命令

artifacthub.io

命令整理

| 用途 | 命令 |
|---------------------------------|---|
| 列出仓库 | helm repo list -o yaml |
| 仓库更新 | helm repo update |
| 安装仓库 | helm repo add bitnami https://charts.bitnami.com/bitnami --username myuser --password mypass |
| 查找chart | helm search repo drupal --versions |
| 安装chart | helm install mysit bitnami/drupal / drupal.tgz |
| 列出已安装chart | helm list --all-namespaces |
| 升级chart配置 | helm upgrade mysit bitnami/drupal --values values.yaml --set a=b --reuse-values |
| 升级chart | helm upgrade mysit bitnami/drupal --version=11.0.1 |
| 安装或升级chart | helm upgrade --install mysit bitnami/drupal |
| 卸载安装 | helm uninstall mysit --keep-history |
| dry-run | helm install mysit bitnami/drupal --dry-run |
| template | helm template mysit bitnami/drupal |
| 查找发布的详细信息 | helm get (hooks、manifests、notes、values和all) |
| 查看chart的默认配置 | helm inspect values bitnami/drupal |
| 查看历史记录 | helm history mysit |
| 回滚 | helm rollback mysit 2 |
| 打包 | helm package deployment --destination charts/ |
| 校验chart | helm lint deployment |
| 解析和更新depedency | helm dependency update . |
| 如果Chart.lock已经有版本, 可以重建charts目录 | helm dependency build |
| 创建仓库index | helm repo index charts/ --merge charts/index-old.yaml |
| 从存储库下载chart | helm pull mycharts/superapp --version 0.1.0 |

| 用途 | 命令 |
|---------|--|
| 删除存储库 | helm repo remove mychars |
| 登陆登记点 | helm registry login -u myuser localhost:5000 |
| 注销登记站 | helm registry logout localhost:5000 |
| 创建chart | helm create anvil |
| 安装插件 | helm plugin install XXX |
| 列出已安装插件 | helm plugin list |
| 更新插件 | helm plugin update starter |
| 卸载插件 | helm plugin remove starter |
| 查看状态 | helm status tomcat |
| | |

属性说明

- dry-run
 - 当在命令行中添加此标志时，将导致Helm逐步完成前四个阶段（加载chart、确定值、渲染模板、格式化为YAML）。但是当第四阶段结束时，Helm会将大量信息转储到标准输出，包括所有渲染的模板。然后它将退出，而不将对象发送给Kubernetes，也不创建任何发布记录。
- helm template
 - template命令执行前4个阶段（加载chart、解析值、渲染模板、格式化为YAML）。但它也有一些额外的注意事项：·在helm template执行期间，Helm从不联系远程Kubernetes服务器。·template命令的作用始终类似于安装。·通常需要联系Kubernetes服务器的模板函数和指令将只返回默认数据。·chart只能访问默认的Kubernetes类型。
- 有时你需要截取YAML，用自己的工具修改它，然后将其加载到Kubernetes中。Helm提供了一种执行这个外部工具的方法，而不必使用helm template。install、upgrade、rollback和template上的--post-renderer标记会导致Helm将YAML数据发送到命令，然后将结果读回Helm。这是使用Kustomize等工具的好方法。
- 回滚
 - 这个命令告诉Helm获取wordpress version 2版本，并将清单重新提交给Kubernetes。回滚不会还原到集群的上一个快照。Helm没有追踪足够的信息来做到这一点。它会重新提交以前的配置，而Kubernetes尝试重置资源以匹配。

开发模板

helm模板语法

1. 动作
 - 逻辑、控制结构和数据计算用{{和}}包装。可以用-来删除前导或尾随空格

2. helm传递到模板的信息

传递到模板中的数据对象的属性名以大写字母开头

- .Values: values.yaml及传入的values
- .Release: 有关发布的信息
- .Chart: chart.yaml的信息
- .Capabilities: 关于集群能力的信息
- .Files: 自定义文件
- .Template: 模板的详细信息

3. 管道

4. 模板函数

<https://oreil.ly/Xtoya>

5. 方法

6. 在chart中查询k8s资源 - lookup函数

7. if/else/with

8. 变量

9. 循环

10. 无

命名模板

问题:

- 如果app有新版本发布, 是使用更新配置的方式好, 还是发布一个chart版本。chart怎么让自动更新到最新, 还是必须指定个版本?