

模块七：Client-go AIOps 实战

王炜 / 前腾讯云 CODING 高级架构师

目录

- 1 Golang CLI 实战：Cobra SDK
- 2 实战一：从零开发 Chat K8s 命令行工具
- 3 实战二：从零开发 K8sGPT 故障诊断工具

1. Golang CLI 实战：Cobra SDK

Cobra 简介

Cobra 是一个用于创建 Golang CLI 应用程序的库，提供了命令行参数解析、自动生成帮助文档和命令补全功能。它简化了多级命令行结构的开发，使开发者能够轻松创建层级命令和子命令。



Cobra 优势

- 很容易实现子命令，类似于 `kubectl get pod`、`kubectl get deployment`
- 自动生成 `--help` flag
- 自动命令建议（针对输错命令的情况）
- 自动生成 `bash` complete
- 根据命令自动生成使用文档
- 支持设置全局、本地和级联 flag
 - 例如对于 `kubectl --kubeconfig` flag 而言是全局的
- 社区活跃：`kubectl`、`helm`、`docker` 等 cli 工具也用

Cobra 核心概念

- 命令 (COMMAND)：命令表示要执行的操作
- 参数 (ARG)：命令的参数，一般用来表示操作的对象
- 标志 (FLAG)：是命令的修饰，可以调整操作的行为

kubectl	edit	service	nginx	-n	kube-system
AppName	COMMAND	SUB COMMAND	ARG	FLAG	

Quick Start

- 使用 cobra-cli 初始化项目
- `go install github.com/spf13/cobra-cli@latest`
- 创建 k8scopilot && `go mod init github.com/lyzhang1999/k8scopilot`
- 初始化项目: `cobra-cli init --author "aiops" --license mit`
- 修改 root.go 描述
- 构建: `go build -o k8scopilot`
- 添加第一个 command: hello
 - `cobra-cli add hello`

添加第多级子命令

- `cobra-cli add world -p 'helloCmd'`
 - `-p` 指定子命令所属的父命令
 - 父命令只输出使用方法，不写业务逻辑

Flag 类型

- String 类型
 - `StringVar(&variable, "flag", "default", "description")` -----> 不带缩写
 - `StringVarP(&variable, "flag", "shorthand", "default", "description")` -----> 带缩写
 - `StringSliceVar(&variable, "flag", []string{}, "description")` -----> 可接受数组, cli 通过传多个 flag
 - `StringSliceVarP(&variable, "flag", "shorthand", []string{}, "description")`
- Bool 类型
 - `BoolVar(&variable, "flag", false, "description")`
 - `BoolVarP(&variable, "flag", "shorthand", false, "description")`
- Int 类型
 - `IntVar(&variable, "flag", 0, "description")`
 - `IntVarP(&variable, "flag", "shorthand", 0, "description")`
- Float 类型
 - `Float64Var(&variable, "flag", 0.0, "description")`
 - `Float64VarP(&variable, "flag", "shorthand", 0.0, "description")`

添加 Flag

- PersistentFlags: 持久 Flag, 定义它的命令和子命令都可以使用, 例如在 rootCmd 里添加全局 Flag
 - `rootCmd.PersistentFlags().StringVarP(&kubeconfig, "kubeconfig", "k", defaultKubeconfig, "kubeconfig file")`
 - `rootCmd.PersistentFlags().StringVarP(&namespace, "namespace", "n", "default", "namespace")`
 - 在 world.go 里可以直接读取
- 本地 Flags: 仅限于定于定义它的命令可以使用
 - `worldCmd.Flags().StringVarP(&Source, "source", "s", "", "Source directory to read from")`
- 默认 Flag 都是可选的, 如果是必须的 Flag 可通过以下方式设置
 - `rootCmd.MarkFlagRequired("source")`
 - 不设置则会报错

其他基础设置

- 设置 Version
 - 在 `root.go` `&cobra.Command{}` 里设置 Version
- 标记命令已废弃
 - 在子命令中 `&cobra.Command{}` 里设置 `Deprecated`, 打印输出提示

2. 实战一：从零开发 Chat K8s 命令行工具

实现效果

- Command: k8scopilot ask chatgpt
 - 进入交互模式
- 部署资源，例如：
 - 帮我部署一个 Deploy，镜像是 nginx
 - 部署一个 Pod，镜像是 nginx
- 查询资源，例如
 - 查一下 default NS 的 Pod/SVC/Deploy 资源
- 删除资源，例如
 - 删除 default NS 的 nginx deploy

步骤

- 添加 ask chatgpt command
 - cobra-cli add ask
 - cobra-cli add chatgpt -p 'askCmd'
- 实现一个简单的回复
- 封装 utils/openai.go, 调用 OpenAI 得到回复
- 调用 OpenAI Function calling
- 解析 OpenAI 返回的消息, 手动调用对应的函数

3. 实战二：从零开发 K8sGPT 故障诊断工具

实现效果

- Command: k8scopilot analyze event
- 分析集群的 Event 事件
- 获取故障的 Pod 日志
- ChatGPT 给出分析建议

步骤

- 添加 analyze event command
 - cobra-cli add analyze
 - cobra-cli add event -p 'analyzeCmd'
- 使用 Client-go 获取集群 Event 事件
- 加 Event 和 Log 发给 ChatGPT 获取建议

效果

```
analyze.sh

1 > ./k8scopilot analyze event
2
3 找到以下 Pod Waring 事件及其日志:
4
5 Pod 名称: demo-1-7977c8646d-sf4sv
6 Event Message: Back-off restarting failed container model-6-demo-1 in pod demo-
1-7977c8646d-sf4sv_default(40833261-b322-462d-8dca-8b5aab869930)
7 Namespace: default
8 Logs:
9 err pods is forbidden: User "system:serviceaccount:default:default" cannot list
resource "pods" in API group "" in the namespace "default"err deployments.apps
is forbidden: User "system:serviceaccount:default:default" cannot list resource
"deployments" in API group "apps" in the namespace "default"
10
11 GPT 建议:
12 1. 创建角色 (Role) : kubectl create clusterrole demo-role --verb=list --
resource=pods,deployments
13 2. 创建角色绑定: kubectl create clusterrolebinding demo-role-binding --
clusterrole=demo-role --serviceaccount=default:default
14 3. 重启 Pod: kubectl delete pod demo-1-7977c8646d-sf4sv
```

课后作业

- 完善 chatgpt.go, 实现 deleteResource 方法, 能其能以对话的方式删除 K8s 资源

THANKS