



环境准备

本节主要介绍 Kubernetes 源码获取以及编译环境、IDE 配置等。

1. 运行环境

2. 源码下载

3. IDE

1. 运行环境

我们这里使用 MAC 作为源码分析操作的系统，后续如果需要调试环境最好使用 Linux，比如 CentOS 系统。

本地 Golang 环境，直接前往官方网站 [<https://golang.org/>] 下载安装即可：

```
$ go version
go version go1.14.4 darwin/amd64
```

基于最新版本 Kubernetes v1.18.5 进行分析。

2. 源码下载

直接 Clone 官方 Kubernetes 源码即可获得代码：

```
$ git clone https://github.com/kubernetes/kubernetes.git
```

但是由于 Kubernetes 源码非常大，而且我们访问 GitHub 又比较慢，为解决这个问题我们可以使用加速器进行 Clone，只需要将 github.com 替换成 github.com.cnpmjs.org 即可：

```
$ git clone https://github.com.cnpmjs.org/kubernetes/kubernetes.git
```

Clone 完成后本地目录如下所示：



```
$ kubernetes [master] ✗ tree . -L 1
.
```

```
|— BUILD.bazel -> build/root/BUILD.root
|— CHANGELOG
|— CHANGELOG.md -> CHANGELOG/README.md
|— CONTRIBUTING.md
|— Godeps
|— LICENSE
|— Makefile -> build/root/Makefile
|— Makefile.generated_files -> build/root/Makefile.generated_files
|— OWNERS
|— OWNERS_ALIASES
|— README.md
|— SECURITY_CONTACTS
|— SUPPORT.md
|— WORKSPACE -> build/root/WORKSPACE
|— _output
|— api
|— build
|— cluster
|— cmd
|— code-of-conduct.md
|— docs
|— go.mod
|— go.sum
|— hack
|— logo
|— pkg
|— plugin
|— staging
|— test
|— third_party
|— translations
|— vendor

17 directories, 15 files
```

我们可以看到 Kubernetes 源码目录非常多，本身也是一个几百万行代码的大型项目，我们也不可能把每一段代码都讲解到，主要是围绕 Kubernetes 的主线来讲解，我们先看一下几个主要的目录：

Copy of 源码目录

 目录名	 用途
<u>cmd</u>	每个组件代码入口（main函数）
<u>pkg</u>	各个组件的具体功能实现
<u>staging</u>	已经分库的项目
<u>vendor</u>	依赖

我们知道 Kubernetes 的各个组件都是二进制文件，这些二进制文件的入口就在 cmd 目录下面，比如 kube-apiserver、kubeadmin、kubectl、kubelet 等等，都分别在

cmd 目录下面有一个入口文件，在入口文件的 main 函数中都是通过 Cobra 这个 CLI 工具来进行初始化的，所以的配置参数或者配置文件也是通过 Cobra 来进行解析的。

而 pkg 目录下面就是各个组件的具体功能的实现，比如 kube-apiserver 的认证鉴权准入控制器等功能都是在该目录下面实现的。

staging 这里的代码都存放在独立的仓库中，以引用包的方式添加到项目中来。

vendor 下面就是项目依赖的第三方包

3. IDE

由于 Kubernetes 源码非常庞大，所以使用 IDE 查看源码更为方便快捷，这里我们推荐使用 GoLand，如果你习惯使用 VSCode 也是可以的，只是整体来说 GoLand 更为方便。

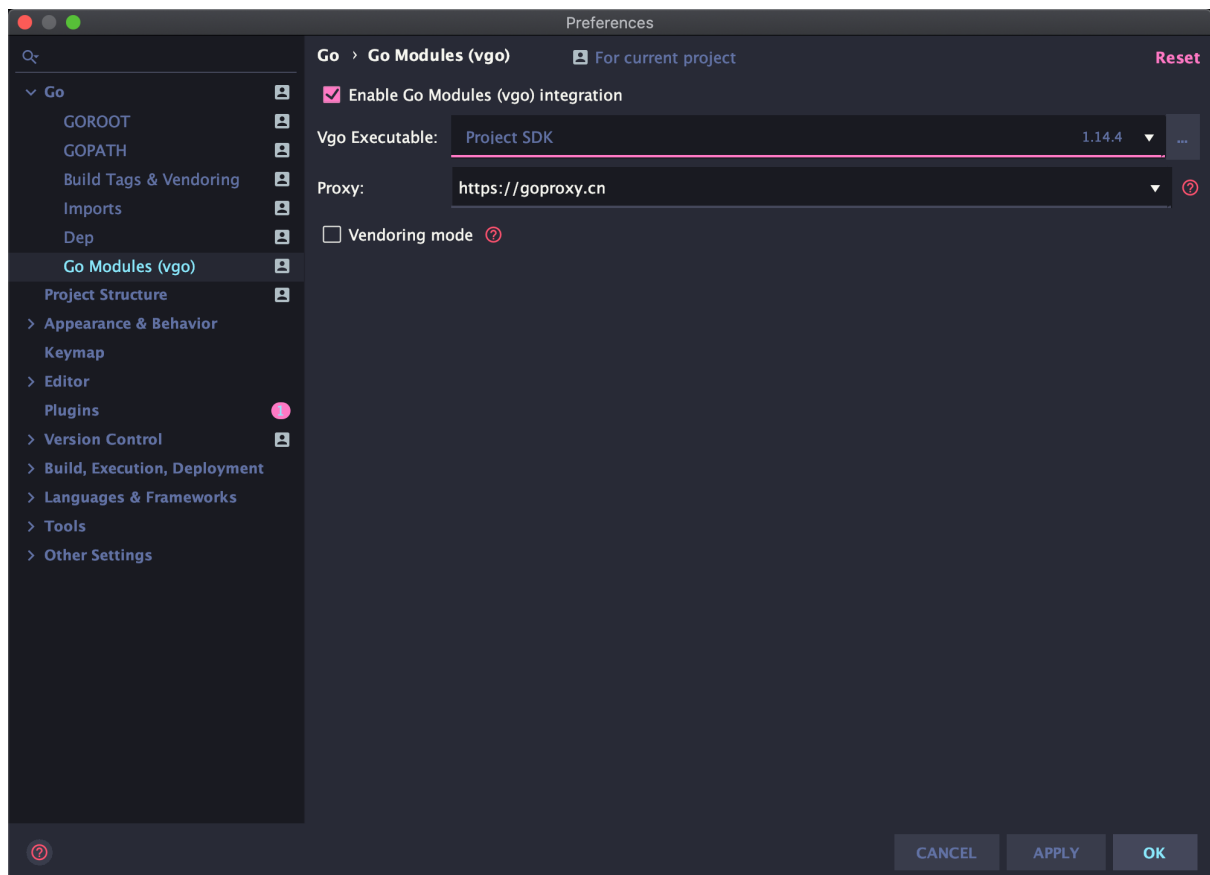
首先切换到我们分析的 v1.18.5 分支：

```
$ kubernetes [master] git checkout v1.18.5
```

然后基于该分支切换一个名为 dev 的分支方便我们测试：

```
$ kubernetes [e6503f8d8f7] git checkout -b dev  
Switched to a new branch 'dev'
```

使用 GoLand 打开 Kubernetes 源码根目录，然后将项目配置成 Go Modules 模式，GoLand → Preference → Go → Go Modules (vgo)：



然后切换到终端中获取依赖包：

```
$ export GOMODULE=on # 1.14 版本的 Go 不需要手动开启
$ export GOPROXY=https://goproxy.cn
$ go mod tidy
```

到这里就完成了对 IDE 的配置，下面就可以开始来分析我们的源码了。