



# Configure

- General
- 源码管理
- Triggers
- Environment
- Build Steps
- 构建后操作

## General

Enabled ☐

描述

纯文本 [预览](#)

- ☐ GitHub 项目
- ☐ Throttle builds
- ☐ 丢弃旧的构建
- ☐ 参数化构建过程
- ☐ 在必要的时候并发构建

高级

### 源码管理

Connect and manage your code repository to automatically pull the latest code for your builds.



Save

应用

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- ☐ 触发远程构建 (例如,使用脚本)
- ☐ 其他工程构建后触发
- ☐ 定时构建
- ☐ GitHub hook trigger for GITScm polling
- ☐ 轮询 SCM

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant
- ☐ 在构建日志中添加时间戳前缀

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

执行 shell

命令

查看 可用的环境变量列表

```
#!/bin/bash
# 提权 远程复制文件
set -ex
DATE=$(date +%Y%m%d%H%M)
source /etc/profile
go version

# 打印环境变量验证
echo "HOME: $HOME"
echo "GOPATH: $GOPATH"
echo "GOCACHE: $GOCACHE"
echo "GOMODCACHE: $GOMODCACHE"
echo "PATH: $PATH"

# 检查目录是否存在
```

```
ls -ld $HOME
ls -ld $GOPATH
ls -ld $GOMODCACHE

# 设置基本环境变量 /home/hroot/jenkins/run-config/workspace/go_
export HOME=/home/hroot/jenkins/run-config/workspace/go_server
export GOPATH=$HOME/go
export GOPATH=$HOME/.cache/go-build
export GOMODCACHE=$GOPATH/pkg/mod
export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin

# 打印环境变量验证
echo "HOME: $HOME"
echo "GOPATH: $GOPATH"
echo "GOPATH: $GOPATH"
echo "GOPATH: $GOPATH"
echo "GOPATH: $GOPATH"
echo "GOPATH: $GOPATH"

# 确保缓存目录存在
mkdir -p $GOPATH
mkdir -p $GOMODCACHE

# 检查目录是否存在
ls -ld $HOME
ls -ld $GOPATH
ls -ld $GOMODCACHE

go version

echo "[start] 开始 $DATE ..."
cd goapijob-main
go mod tidy
CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -o bin/coder_j
CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -o bin/coder_c

CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -o bin/coder_t
CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -o bin/coder_j
cp -r config/config.yaml template bin/
cd bin/
chmod +x coder_*
exit 0

#nohup ./coder_job -config config.yaml > /dev/null 2>&1 &
#nohup ./coder_queue -config config.yaml > /dev/null 2>&1 &
#nohup ./coder_translate -config config.yaml > /dev/null 2>&1 &
#nohup ./coder_job_html -config config.yaml > /dev/null 2>&1 &
```

高级 ▾

增加构建步骤 ▾

构建后操作

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

增加构建后操作步骤 ▾