**Fabric编译制作镜像**

**fabric**

Go Tools安装

mkdir -p $GOPATH/src/golang.org/x

cd $GOPATH/src/golang.org/x

git clone <https://github.com/golang/tools.git>

git clone <https://github.com/golang/lint.git>

git clone [https://github.com/golang/crypto.git](https://github.com/golang/tools.git)

git clone https://github.com/golang/sys.git

下载完毕，安装Fabric可能用到的Go工具：

go get github.com/kardianos/govendor

go get github.com/golang/lint/golint

go get golang.org/x/tools/cmd/goimports

go get github.com/onsi/ginkgo/ginkgo

go get github.com/axw/gocov/...

go get github.com/client9/misspell/cmd/misspell

go get github.com/AlekSi/gocov-xml

go get github.com/golang/protobuf/protoc-gen-go

go get github.com/estesp/manifest-tool

go get github.com/maxbrunsfeld/counterfeiter

go get github.com/vektra/mockery/cmd/mockery

直接编译可如下操作(1.2版本之前是build,1.2版本及以上是.build)

cd $GOPATH/src/github.com/hyperledger/fabric

mkdir -p build/docker/gotools/bin

cp $GOPATH/bin/\* build/docker/gotools/bin

（此前安装的gotools可能会由于go环境变量配置的原因而存在$GOPATH/bin或者$GOROOT/bin里面，此处路径对应填写）

make docker

分步编译（用fabric-samples测试时需要用到分步编译生成的几个可执行文件，所以可以采用分步编译）

cd $GOPATH/src/github.com/hyperledger/fabric

mkdir -p build/docker/gotools/bin

cp $GOPATH/bin/\* build/docker/gotools/bin

1.编译orderer

make orderer

2. 编译peer

make peer

3. 编译tools

make configtxgen

make cryptogen

make configtxlator

4.生成镜像

make orderer-docker

make peer-docker

make tools-docker

make docker