基础环境

- ubuntu20.04

安装基础依赖

```
# 安装curl
sudo apt install curl
# 为了curl能够支持https协议，安装openssl
sudo apt install openssl libssl-dev
# 安装git
sudo apt install git
# 安装cmake
sudo apt install cmake
# 安装jq
sudo apt install jq
# 安装libtool
sudo apt install libtool libltdl-dev
```

安装docker

> #### 安装docker
>
> ```
> sudo apt update
> sudo apt install apt-transport-https ca-certificates curl gnupg-agent
> software-properties-common
> curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
> add -
> sudo add-apt-repository "deb [arch=amd64]
> https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
> sudo apt update
> sudo apt install docker-ce docker-ce-cli containerd.io
> ```

⚠建议用方法2

> #### 以非root用户的身份运行Docker
>
> ```
> # 如果出现Cannot connect to the Docker daemon. Is the docker daemon running
> on this host?
> # 解决办法1
> ```

```
sudo chown seed:docker /var/run/docker.sock
# seed改成自己的用户名

# 解决办法2
# 创建一个名为docker的组
sudo groupadd docker
# 将你的用户添加到docker组
sudo usermod -aG docker $USER
# 重启
```

安装Golang

- go官网链接

```
# 下载官网的linux x86-64 Archive
wget https://golang.google.cn/dl/go1.19.3.linux-amd64.tar.gz --no-check-certificate
# 解压
sudo tar -xzf go1.19.3.linux-amd64.tar.gz -C /usr/local
# 主目录下建立goDir文件夹，做为GOPATH路径
mkdir ~/goDir
# 在zsh配置文件中添加系统PATH
vim ~/.zshrc
export GOPATH=/home/seed/goDir
export GOROOT=/usr/local/go
export PATH=$PATH:$GOPATH/bin
export PATH=$PATH:$GOROOT/bin
# 刷新配置文件
source ~/.zshrc
```

```
# 让root也能执行go命令（如果root没有安装zsh，则修改 /root/.bashrc）
sudo su
vim ~/.zshrc
export GOPATH=/home/seed/goDir
export GOROOT=/usr/local/go
export PATH=$PATH:$GOPATH/bin
export PATH=$PATH:$GOROOT/bin
```

搭建fabric环境

```
mkdir ~/blockchain
cd blockchain
# github克隆fabric-samples
git clone https://github.com/hyperledger/fabric-samples
```

```
cd fabric-samples
# 下载安装脚本并构建docker
wget
https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-
fabric.sh --no-check-certificate
chmod +x install-fabric.sh
./install-fabric.sh docker
./install-fabric.sh b
# 启动docker
cd test-network
./network.sh up
# 查看是否创建成功
docker ps
# 关闭docker
./network.sh down
```