# Hyperledger Fabric 环境配置

# 操作系统

常见的 Linux 系统(Ubuntu、CentOs、Redhat)和 MacOS 等 都可以运行fabric 如果是 window 电脑,可以通过安装 Ubuntu 虚拟机。

# 安装 cURL

mac 系统自带 cURL 所以不需要安装

## Ubuntu系统

sudo apt install curl

# **Docker 和 Docker Compose**

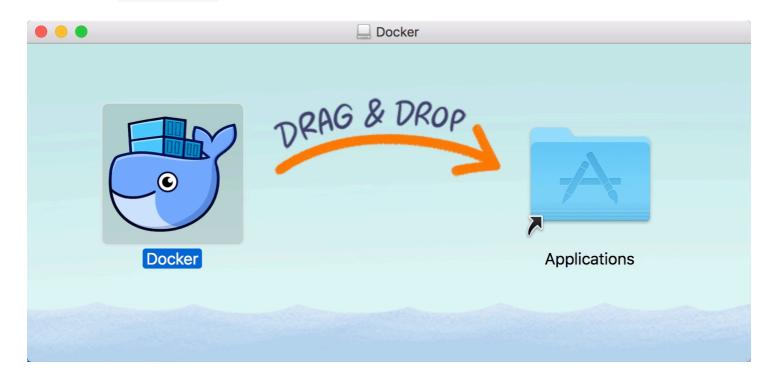
### mac 系统

下载 Docker

https://store.docker.com/editions/community/docker-ce-desktop-mac



双击下载的 Docker dmg ,将 Docker 拖到 应用程序中。



#### 测试 Docker

```
docker-compose --version
```

### Ubuntu系统

#### 通过仓库安装Docker

• 建立仓库

```
stable"
```

• 安装 Docker CE

```
$ sudo apt-get update
$ sudo apt-get install docker-ce
$ sudo docker run hello-world
```

参考链接: https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/#prerequisites

### Docker Compose 安装

```
sudo apt install docker-compose
```

# Go 开发语言

### mac 系统

#### 安装 Homebrew

• 官网

```
https://brew.sh/index_zh-cn.html
```

• 安装 Homebrew

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/in
stall/master/install)"
```

### 安装 go

```
$ brew install go
```

#### 配置 GOPATH

```
vi ~/.bashrc

# 添加如下内容

export GOPATH=$HOME/go

export PATH=$PATH:$GOPATH/bin
```

## ubuntu 系统

• 下载 go1.9.2.linux-amd64.tar.gz

https://golang.org/dl/

• 解压 go1.9.2.linux-amd64.tar.gz

```
sudo tar -xzf go1.9.2.linux-amd64.tar.gz -C /usr/local
```

• 配置全局变量

\$ sudo gedit ~/.bashrc

• 添加如下代码

```
export GOPATH=/usr/local/go
export PATH=$GOPATH/bin:$PATH
```

• 重启配置文件

source ~/.bashrc

# 安装 Node.js

## mac系统

第一种方法 brew 安装

\$ brew install node

### 第二种方法 nvm 安装

• 安装 nvm

```
$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.7/install.
sh | bash
```

• 配置 nvm

```
# 编辑 ~/.bashrc

# 添加如下代码

export NVM_DIR="$HOME/.nvm"
. "/usr/local/opt/nvm/nvm.sh"

# 重新 source
source ~/.bashrc
```

### ubuntu系统

• 下载 nodejs

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
sudo apt-get install -y nodejs
```

• 检测 node 版本

```
nodejs -∨
```

# **Python**

由于 Ubuntu 16.04 安装的 python 版本为 3.5.1, Fabric Node.js SDK 要求python2.7版本通过如下命令恢复到2.7

```
$ sudo apt-get install python
```

# 查询 python 版本

\$ python --version