

Hyperledger Fabric开发环境搭建（MacOS系统）

1、安装Homebrew：

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

2、安装Go：

(1) 安装命令：

```
brew install go
```

(2) 检查安装版本（安装前可查看是否已安装）

```
wenzildeiMac:~ wenzil$ go version  
go version go1.9.2 darwin/amd64
```

(3) 配置环境变量

```
#GOPATH路径可能因人而异，注意先用"which go"检查一下path路径（如后面的一段）  
export GOPATH=/usr/local/go  
export GOBIN=$GOPATH/bin  
export PATH=$PATH:$GOBIN
```

vim .bash_profile：进入vim，输入i切换为编辑模式，输入上面三段配置信息，然后按Esc键，输入:wq保存并退出编辑。

source ~/.bash_profile：作为执行命令，会立即生效，然后再检查一下Go

```
wenzildeiMac:go wenzil$ which go  
/usr/local/go/bin/go  
wenzildeiMac:~ wenzil$ cd ~  
wenzildeiMac:~ wenzil$ vim .bash_profile  
wenzildeiMac:~ wenzil$ source ~/.bash_profile  
wenzildeiMac:~ wenzil$ go version  
go version go1.9.2 darwin/amd64
```

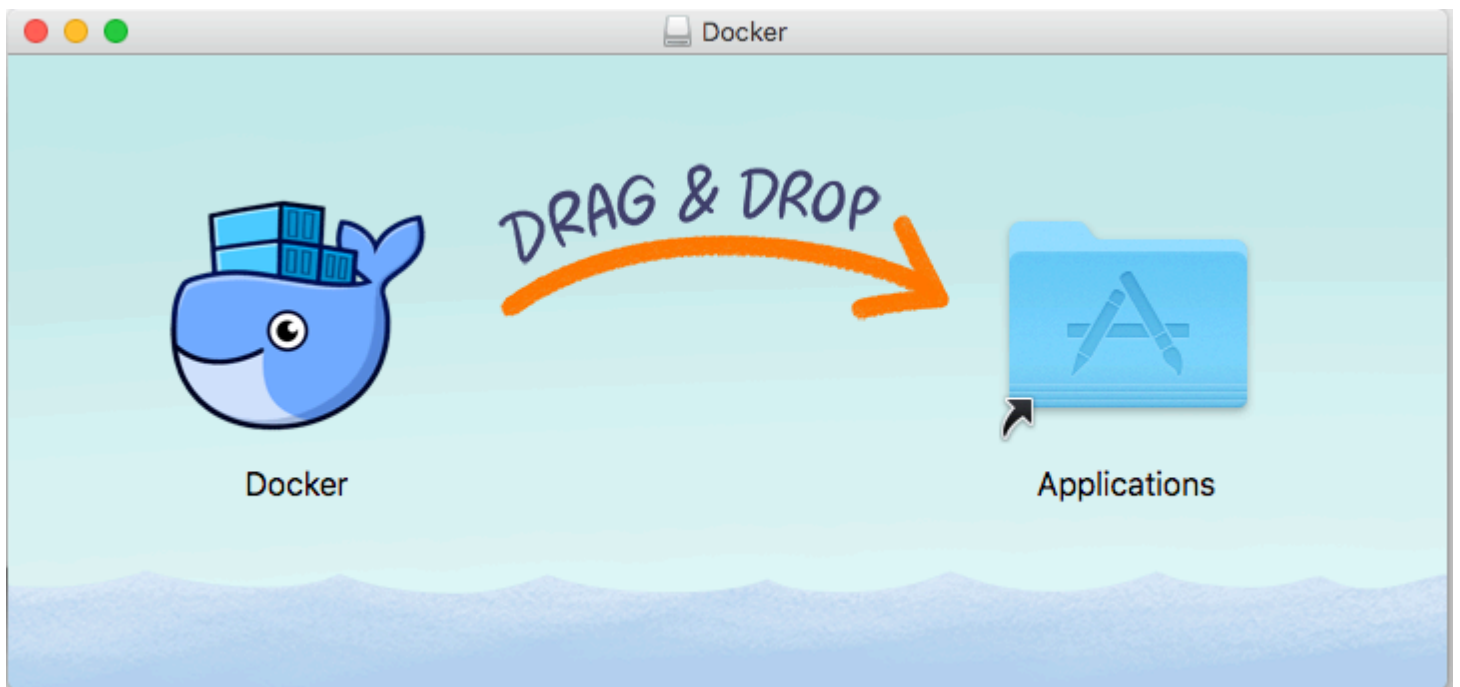
(4) 检查Go的配置信息

```
wenzildeiMac:~ wenzil$ go env
GOARCH="amd64"
GOBIN=""
GOEXE=""
GOHOSTARCH="amd64"
GOHOSTOS="darwin"
GOOS="darwin"
GOPATH="/Users/wenzil/.rvm/gems/ruby-2.4.0/bin:/Users/wenzil/.rvm/gems/ruby-2.4.0@global/bin:/Users/wenzil/.rvm/rubies/ruby-2.4.0/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/go/bin:/Users/wenzil/.rvm/bin:/usr/local/go/bin"
GORACE=""
GOROOT="/usr/local/go"
GOTOOLDIR="/usr/local/go/pkg/tool/darwin_amd64"
GCCGO="gccgo"
CC="clang"
GOGCCFLAGS="-fPIC -m64 -pthread -fno-caret-diagnostics -Qunused-arguments -fmessage-length=0 -fdebug-prefix-map=/var/folders/rd/b67k290j5xbcfw59chnjm1qh0000gn/T/go-build472270299=/tmp/go-build -gno-record-gcc-switches -fno-common"
CXX="clang++"
CGO_ENABLED="1"
CGO_CFLAGS="-g -O2"
CGO_CPPFLAGS=""
CGO_CXXFLAGS="-g -O2"
CGO_FFLAGS="-g -O2"
CGO_LDFLAGS="-g -O2"
PKG_CONFIG="pkg-config"
```

3、安装Docker：

Docker界面化软件方便操作和管理，下载地址如下
(点击页面的“Get Docker”按钮即可下载)

```
# 下载地址
https://store.docker.com/editions/community/docker-ce-desktop-mac
```



终端测试命令

```
wenzildeiMac:~ wenzil$ docker --version
Docker version 18.03.1-ce, build 9ee9f40
wenziliangdeiMac:~ wenzil$ docker-compose --version
docker-compose version 1.21.1, build 5a3f1a3
```

4、安装nvm

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

安装成功后显示这么一段话（版本不同可能显示不同）：

```
=> If you wish to uninstall them at a later point (or re-install them under
your
```

```
=> `nvm` Nodes), you can remove them from the system Node as follows:
```

```
$ nvm use system
$ npm uninstall -g a_module
```

```
=> Close and reopen your terminal to start using nvm or run the following to
use it now:
```

```
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This l
oads nvm bash_completion
```

打开终端，编辑.bash_profile：

```
wenzildeiMac:~ wenzil$ cd ~
wenzildeiMac:~ wenzil$ vim .bash_profile
```

在.bash_profile中直接复制添加上面返回打印的内容：

```
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This l
oads nvm bash_completion
```

然后执行如下命令，让配置立即生效：

```
source .bash_profile
```

5、安装node指定版本

```
Node.js version 7.x is not supported at this time.
Node.js - version 6.9.x or greater
```

（意思说Fabric暂时不支持7.x，需要安装6.9.x或者更多版本）
官网对应的上面一句话地址：

```
http://hyperledger-fabric.readthedocs.io/en/v1.1.0-alpha/prereqs.html#node-j
s-runtime-and-npm
```

6、下载Hyperledger的一个Demo源码

```
git clone https://github.com/hyperledger/fabric-samples
```

7、下载特定平台的二进制文件（注意这里有坑，也可以跳过此步骤，先操作第7和第8步，再回来看下）

坑说明：有人直接让你下载某个平台某个版本的二进制文件，然后发现运行不了，无法进行下一步操作。

先打开Fabric的官网，会自动跳转到对应的版本：

<http://hyperledger-fabric.readthedocs.io/>

然后在左边的输入框，输入“fabric-samples”，右边会显示对应的结果，点击第一个

The screenshot shows the Hyperledger Fabric documentation website. On the left, a sidebar contains a search input field with 'fabric-samples' entered. The main content area is titled 'Search Results' and shows 'Search finished, found 7 page(s) matching the search query.' The first result, 'Hyperledger Fabric Samples (from project hyperledger-fabric)', is highlighted with a red box. Below it, other results are listed: 'Getting Started (from project hyperledger-fabric)', 'Chaincode for Developers (from project hyperledger-fabric)', 'Writing Your First Application (from project hyperledger-fabric)', 'Adding an Org to a Channel (from project hyperledger-fabric)', 'Building Your First Network (from project hyperledger-fabric)', and 'Upgrading Your Network Components (from project hyperledger-fabric)'.

然后往下看，会看到框住的两部分，第一个就是说终端执行curl命令下载特定平台的二进制文件，然后可以打开第二个部分的那个网址了解最新的curl获取正确环境的其他信息

The screenshot shows the Hyperledger Fabric documentation page for downloading platform-specific binaries. The page has a sidebar on the left with a navigation menu. The main content area is titled "Download Platform-specific Binaries".

Download Platform-specific Binaries

Next, we will install the Hyperledger Fabric platform-specific binaries. This process was designed to complement the Hyperledger Fabric Samples above, but can be used independently. If you are not installing the samples above, then simply create and enter a directory into which to extract the contents of the platform-specific binaries.

Please execute the following command from within the directory into which you will extract the platform-specific binaries:

```
curl -sSL https://goo.gl/6wtTN5 | bash -s 1.1.0
```

Note

If you get an error running the above curl command, you may have too old a version of curl that does not handle redirects or an unsupported environment.

Please visit the [Prerequisites](#) page for additional information on where to find the latest version of curl and get the right environment. Alternately, you can substitute the un-shortened URL: <https://github.com/hyperledger/fabric/blob/master/scripts/bootstrap.sh>

Note

You can use the command above for any published version of Hyperledger Fabric. Simply replace '1.1.0' with the version identifier of the version you wish to install.

The command above downloads and executes a bash script that will download and extract all of the platform-specific binaries you will need to set up your network and place them into the cloned repo you created above. It retrieves four platform-specific binaries:

Navigation Menu:

- Getting Started
 - Prerequisites
- Hyperledger Fabric Samples
 - Download Platform-specific Binaries
 - Install Prerequisites
 - Install Binaries and Docker Images
 - Hyperledger Fabric Samples
 - API Documentation
 - Hyperledger Fabric SDKs
 - Hyperledger Fabric CA
- Key Concepts
- Tutorials
- Operations Guides
- Commands Reference

Read the Docs v: release-1.1

#第一部分的地址

```
curl -sSL https://goo.gl/6wtTN5 | bash -s 1.1.0
```

#第二部分的地址

```
https://github.com/hyperledger/fabric/blob/master/scripts/bootstrap.sh
```

如图所示，说明需要下载"1.1.0"版本

注意：直接下载会很慢，而且很容易失败（VPN模式下中断了好几次）

建议离线下载，网址：

```
https://nexus.hyperledger.org/content/repositories/releases/org/hyperledger/fabric/hyperledger-fabric/
```

打开网址，一定要找到对应操作系统对应版本号的文件（不然会失败），如图：

Index of /repositories/releases/org/hyperledger/fabric/hyperledger-fabric

Name	Last Modified	Size	Description
Parent Directory			
chaintool-1.0.0/	Mon Jul 24 01:43:57 UTC 2017		
chaintool-1.0.1/	Wed Sep 06 19:02:29 UTC 2017		
chaintool-1.1.0/	Thu Mar 22 23:26:21 UTC 2018		
chaintool-1.1.1/	Sun Apr 22 02:05:29 UTC 2018		
darwin-amd64-1.0.0-rc1/	Fri Jun 23 22:11:07 UTC 2017		
darwin-amd64-1.0.0/	Tue Jul 11 16:35:25 UTC 2017		
darwin-amd64-1.0.1/	Thu Aug 10 10:46:11 UTC 2017		
darwin-amd64-1.0.2/	Fri Sep 01 22:57:33 UTC 2017		
darwin-amd64-1.0.3/	Tue Oct 03 13:01:36 UTC 2017		
darwin-amd64-1.0.4/	Tue Oct 31 19:08:46 UTC 2017		
darwin-amd64-1.0.5/	Wed Dec 06 16:02:37 UTC 2017		
darwin-amd64-1.0.6/	Mon Feb 19 22:54:11 UTC 2018		
darwin-amd64-1.1.0-alpha/	Fri Jan 26 16:50:14 UTC 2018		
darwin-amd64-1.1.0-preview/	Wed Nov 01 19:38:02 UTC 2017		
darwin-amd64-1.1.0-rc1/	Thu Mar 01 20:01:31 UTC 2018		
darwin-amd64-1.1.0/	Thu Mar 15 22:18:12 UTC 2018		
linux-amd64-1.0.0-rc1/	Fri Jun 23 22:10:54 UTC 2017		
linux-amd64-1.0.0/	Tue Jul 11 16:35:04 UTC 2017		

打开MacOS对应版本的文件夹，还有Windows、Linux其他系统，如图Windows系统可以打开该文件夹：

windows-amd64-1.0.5/	Wed Dec 06 16:02:30 UTC 2017	
windows-amd64-1.0.6/	Mon Feb 19 22:54:05 UTC 2018	
windows-amd64-1.1.0-alpha/	Fri Jan 26 16:50:03 UTC 2018	
windows-amd64-1.1.0-preview/	Wed Nov 01 19:37:56 UTC 2017	
windows-amd64-1.1.0-rc1/	Thu Mar 01 20:01:26 UTC 2018	
windows-amd64-1.1.0/	Thu Mar 15 22:18:05 UTC 2018	
maven-metadata.xml	Mon Apr 23 01:30:06 UTC 2018	3295
maven-metadata.xml.md5	Mon Apr 23 01:30:06 UTC 2018	32
maven-metadata.xml.sha1	Mon Apr 23 01:30:06 UTC 2018	40

MacOS对应的文件夹，如图，点击下载压缩包文件：

Index of /repositories/releases/org/hyperledger/fabric/hyperledger-fabric/darwin-amd64-1.1.0

Name	Last Modified	Size	Description
Parent Directory			
hyperledger-fabric-darwin-amd64-1.1.0.pom	Thu Mar 15 22:18:12 UTC 2018	458	
hyperledger-fabric-darwin-amd64-1.1.0.pom.md5	Thu Mar 15 22:18:12 UTC 2018	32	
hyperledger-fabric-darwin-amd64-1.1.0.pom.sha1	Thu Mar 15 22:18:12 UTC 2018	40	
hyperledger-fabric-darwin-amd64-1.1.0.tar.gz	Thu Mar 15 22:18:10 UTC 2018	29676840	
hyperledger-fabric-darwin-amd64-1.1.0.tar.gz.md5	Thu Mar 15 22:18:12 UTC 2018	32	
hyperledger-fabric-darwin-amd64-1.1.0.tar.gz.sha1	Thu Mar 15 22:18:12 UTC 2018	40	

下载成功后解压出来，是一个bin文件夹，然后复制该文件夹到fabric-samples根目录下，如图：

fabric-samples				
名称	修改日期	大小	种类	
balance-transfer	今天 下午1:38	--	文件夹	
basic-network	今天 下午1:38	--	文件夹	
bin	2018年1月27日 上午12:41	--	文件夹	
chaincode	今天 下午1:38	--	文件夹	
chaincode-docker-devmode	今天 下午1:38	--	文件夹	
fabcar	今天 下午1:38	--	文件夹	
fabric-ca	今天 下午1:38	--	文件夹	
first-network	今天 下午5:41	--	文件夹	
high-throughput	今天 下午1:38	--	文件夹	
init.sh	今天 下午1:41	1 KB	Shell Script	
LICENSE	今天 下午1:38	11 KB	文本编辑 文稿	
MAINTAINERS.md	今天 下午1:38	470 字节	Markdown	
README.md	今天 下午1:38	522 字节	Markdown	
scripts	今天 下午1:38	--	文件夹	

8、设置Docker容器代理

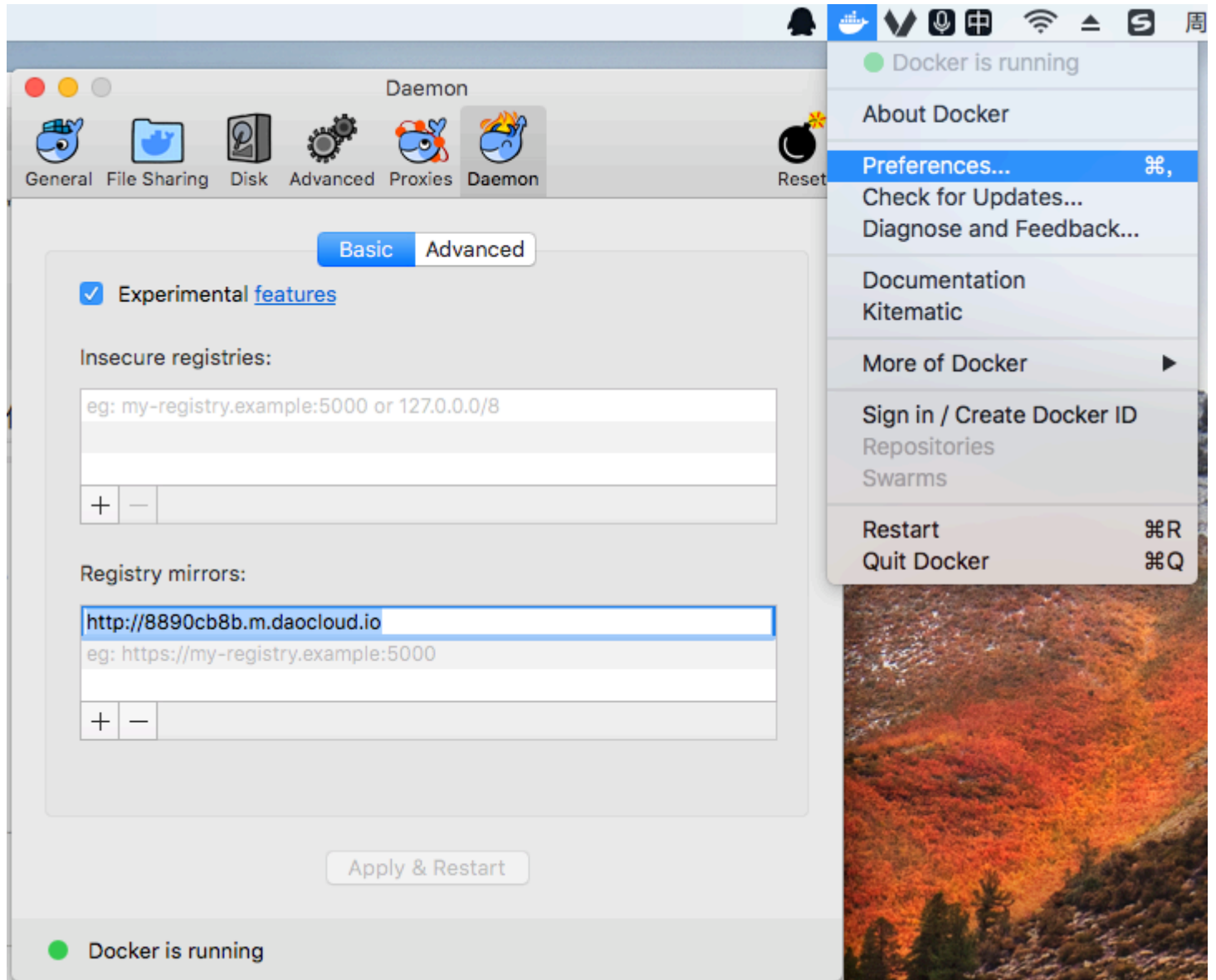
设置Docker代理可提高镜像下载速度，否则很慢（慢到可能需要下载大半天甚至一天以上）。点击"Docker"的"Preferences..."菜单，切换到"Daemon"菜单，在"Registry mirrors"中添加代理，地址如下：

```
# 试了该代理，速度快了很多
http://8890cb8b.m.daocloud.io
```


大牛推荐使用Docker官方提供的中国镜像加速（还没试过），网址：

<https://www.docker-cn.com/registry-mirror>

添加成功后，"Apply & Restart"按钮会变成可点击状态，点击进行配置自动重启Docker。



9、安装Docker镜像前准备

先打开如下网址（这个可能需要翻墙）：

<https://goo.gl/byy2Qj>

在fabric-samples根目录下创建"init.sh"脚本文件（终端执行"vi init.sh"），将网站的内容复制到"init.sh"文件中。

```
← → ↻ 安全 | https://raw.githubusercontent.com/hyperledger/fabric/v1.0.5/scripts/bootstrap.sh ☆ ❷ ⋮

#!/bin/bash
#
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

# current version of fabric released
export VERSION=${1:-1.0.5}
# current version of fabric-ca released
export CA_VERSION=${2:-$VERSION}
export ARCH=$(echo "$(uname -s|tr '[:upper:]' '[:lower:]'|sed 's/mingw64_nt.*/windows/')-$(uname -m | sed 's/x86_64/amd64/g')" | awk '{print tolower($0)}')
#Set MARCH variable i.e ppc64le,s390x,x86_64,i386
MARCH=`uname -m`

dockerFabricPull() {
    local FABRIC_TAG=$1
    for IMAGES in peer orderer couchdb ccenv javaenv kafka zookeeper tools; do
        echo "====> FABRIC IMAGE: $IMAGES"
        echo
        docker pull hyperledger/fabric-$IMAGES:$FABRIC_TAG
        docker tag hyperledger/fabric-$IMAGES:$FABRIC_TAG hyperledger/fabric-$IMAGES
    done
}

dockerCaPull() {
    local CA_TAG=$1
    echo "====> FABRIC CA IMAGE"
    echo
    docker pull hyperledger/fabric-ca:$CA_TAG
    docker tag hyperledger/fabric-ca:$CA_TAG hyperledger/fabric-ca
}

: ${CA_TAG:="$MARCH-$CA_VERSION"}
: ${FABRIC_TAG:="$MARCH-$VERSION"}

echo "====> Downloading platform binaries"
curl https://nexus.hyperledger.org/content/repositories/releases/org/hyperledger/fabric/hyperledger-fabric/${ARCH}-${VERSION}/hyperledger-fabric-${ARCH}-${VERSION}.tar.gz | tar xz

echo "====> Pulling fabric Images"
dockerFabricPull ${FABRIC_TAG}

echo "====> Pulling fabric ca Image"
dockerCaPull ${CA_TAG}
echo
echo "====> List out hyperledger docker images"
docker images | grep hyperledger*
```

注释掉下载平台二进制文件代码（注释echo和curl两行），然后保存并退出（按Esc，输入":wq"）

```
# echo "====> Downloading platform binaries"
# curl https://nexus.hyperledger.org/content/repositories/releases/org/hyperledger/fabric/hyperledger-fabric/${ARCH}-${VERSION}/hyperledger-fabric-${ARCH}-${VERSION}.tar.gz | tar xz
```

跳过了第7步的注意看下：当时是按照7、8、9步骤操作的，现在想想应该可以不用注释上面这两行代码，然后省略第7步手动下载二进制文件，从让系统自动下载特定平台的二进制文件并自动解压，有机会可以试下，成功后应该会在fabric-samples根目录下生成bin文件夹。

设置"init.sh"文件最高权限777，执行"./init.sh"命令之前确保已经启动Docker。

```
vi init.sh
chmod 777 init.sh
./init.sh
```

10、安装Docker镜像

执行了"./init.sh"命令之后，会下载很多镜像文件，目测十几G的样子，成功后如图：

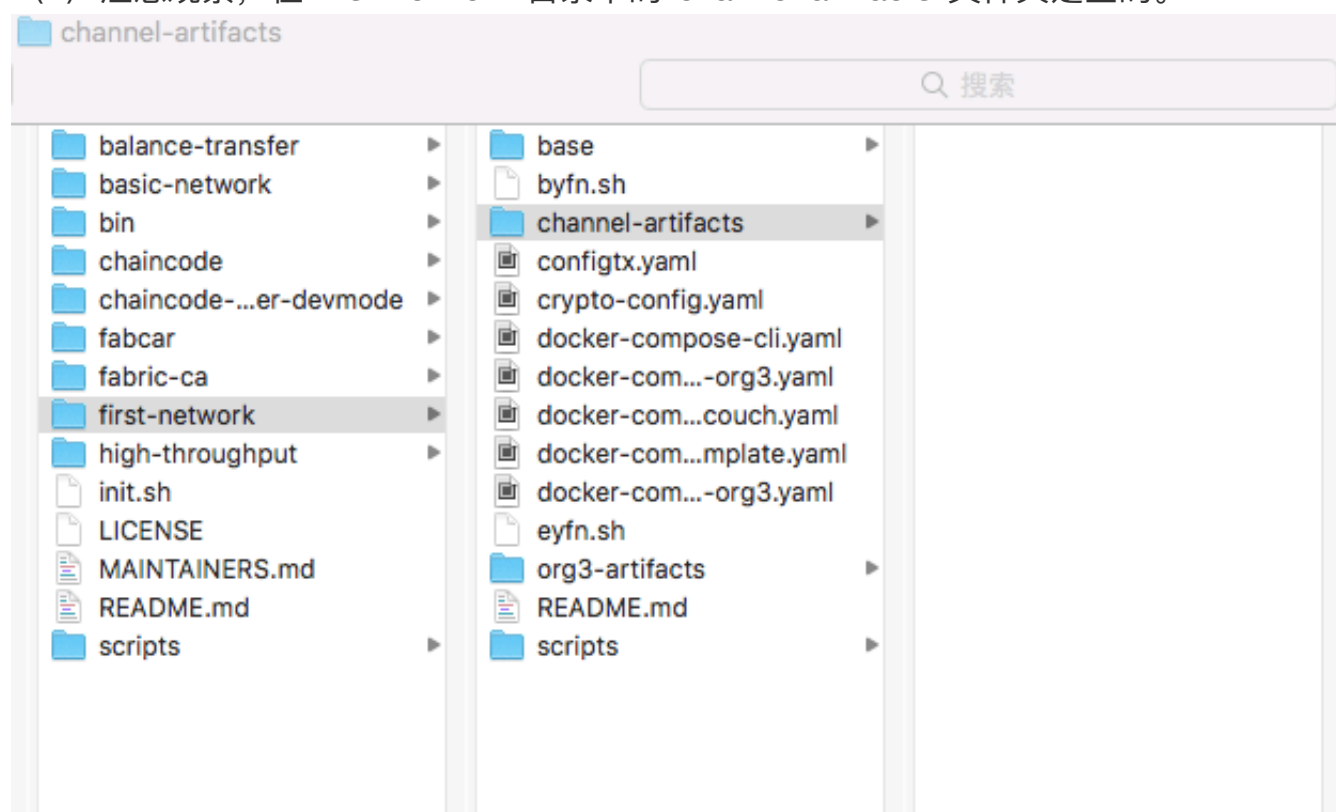
```

==> List out hyperledger docker images
hyperledger/fabric-tools      latest      6a8993b718c8      4 months ago      1.33GB
hyperledger/fabric-tools      x86_64-1.0.5 6a8993b718c8      4 months ago      1.33GB
hyperledger/fabric-couchdb    latest      9a58db2d2723      4 months ago      1.5GB
hyperledger/fabric-couchdb    x86_64-1.0.5 9a58db2d2723      4 months ago      1.5GB
hyperledger/fabric-kafka      latest      b8c5172bb83c      4 months ago      1.29GB
hyperledger/fabric-kafka      x86_64-1.0.5 b8c5172bb83c      4 months ago      1.29GB
hyperledger/fabric-zookeeper  latest      68945f4613fc      4 months ago      1.32GB
hyperledger/fabric-zookeeper  x86_64-1.0.5 68945f4613fc      4 months ago      1.32GB
hyperledger/fabric-orderer     latest      368c78b6f03b      4 months ago      151MB
hyperledger/fabric-orderer     x86_64-1.0.5 368c78b6f03b      4 months ago      151MB
hyperledger/fabric-peer        latest      c2ab022f0bdb      4 months ago      154MB
hyperledger/fabric-peer        x86_64-1.0.5 c2ab022f0bdb      4 months ago      154MB
hyperledger/fabric-javaenv     latest      50890cc3f0cd      4 months ago      1.41GB
hyperledger/fabric-javaenv     x86_64-1.0.5 50890cc3f0cd      4 months ago      1.41GB
hyperledger/fabric-ccenv       latest      33feadb8f7a6      4 months ago      1.28GB
hyperledger/fabric-ccenv       x86_64-1.0.5 33feadb8f7a6      4 months ago      1.28GB
hyperledger/fabric-ca          latest      002c9089e464      4 months ago      238MB
hyperledger/fabric-ca          x86_64-1.0.5 002c9089e464      4 months ago      238MB

```

11、创建第一个区块链网络

(1) 注意观察，在"first-network"目录下的"channel-artifacts"文件夹是空的。



(2) 创建第一个区块链网络前，先关闭所有服务，执行命令：

```
./byfn.sh -m down
```

结果如图：

```

:first-network wenzil$ ./byfn.sh -m down
Stopping with channel 'mychannel' and CLI timeout of '10' seconds and CLI delay of '3' seconds
Continue? [Y/n] y
proceeding ...
Removing network net_byfn
WARNING: Network net_byfn not found.
Removing volume net_orderer.example.com
WARNING: Volume net_orderer.example.com not found.
Removing volume net_peer0.org1.example.com
WARNING: Volume net_peer0.org1.example.com not found.
Removing volume net_peer1.org1.example.com
WARNING: Volume net_peer1.org1.example.com not found.
Removing volume net_peer0.org2.example.com
WARNING: Volume net_peer0.org2.example.com not found.
Removing volume net_peer1.org2.example.com
WARNING: Volume net_peer1.org2.example.com not found.
Removing network net_byfn
WARNING: Network net_byfn not found.
Removing volume net_orderer.example.com
WARNING: Volume net_orderer.example.com not found.
Removing volume net_peer0.org1.example.com
WARNING: Volume net_peer0.org1.example.com not found.
Removing volume net_peer1.org1.example.com
WARNING: Volume net_peer1.org1.example.com not found.
Removing volume net_peer0.org2.example.com
WARNING: Volume net_peer0.org2.example.com not found.
Removing volume net_peer1.org2.example.com
WARNING: Volume net_peer1.org2.example.com not found.
---- No containers available for deletion ----
---- No images available for deletion ----

```

(3) 创建区块链网络，执行如下命令：

```
./byfn.sh -m generat
```

1.没有下载特定平台二进制文件的情况（被坑的结果）：

```

:~:first-network wenzil$ ./byfn.sh -m generate
Generating certs and genesis block for with channel 'mychannel' and CLI timeout of '10' seconds and CLI del
ay of '3' seconds
Continue? [Y/n] y
proceeding ...
/Users/wenzil/~/first-network/fabric-samples/first-network/./bin/cryptogen

#####
#### Generate certificates using cryptogen tool ####
#####
+ cryptogen generate --config=./crypto-config.yaml
org1.example.com
org2.example.com
+ res=0
+ set +x

/Users/wenzil/Desktop/test/fabric-samples/first-network/./bin/configtxgen
#####
##### Generating Orderer Genesis block #####
#####
+ configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block
2018-05-05 17:33:02.697 CST [common/configtx/tool] main -> INFO 001 Loading configuration
2018-05-05 17:33:02.736 CST [common/configtx/tool/localconfig] Load -> CRIT 002 Error unmarshaling config i
nto struct: 4 error(s) decoding:

* '' has invalid keys: capabilities
* 'Profiles[TwoOrgsChannel].Application' has invalid keys: Capabilities
* 'Profiles[TwoOrgsOrdererGenesis]' has invalid keys: Capabilities
* 'Profiles[TwoOrgsOrdererGenesis].Orderer' has invalid keys: Capabilities
+ res=1
+ set +x
Failed to generate orderer genesis block...

```

2.重新下载了正确的特定平台二进制文件的情况，成功通过：


```
first-network — -bash — 113x65
first-network wenzil$ ./byfn.sh -m generate
Generating certs and genesis block for with channel 'mychannel' and CLI timeout of '10' seconds and CLI delay of '3' seconds
[Continue? [Y/n] y
[proceeding ...

first-network/./bin/cryptogen

#####
#### Generate certificates using cryptogen tool #####
#####
+ cryptogen generate --config=./crypto-config.yaml
org1.example.com
org2.example.com
+ res=0
+ set +x

first-network/./bin/configtxgen

#####
##### Generating Orderer Genesis block #####
#####
+ configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./channel-artifacts/genesis.block
2018-05-05 18:57:30.390 CST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2018-05-05 18:57:30.456 CST [common/tools/configtxgen] doOutputBlock -> INFO 002 Generating genesis block
2018-05-05 18:57:30.457 CST [common/tools/configtxgen] doOutputBlock -> INFO 003 Writing genesis block
+ res=0
+ set +x

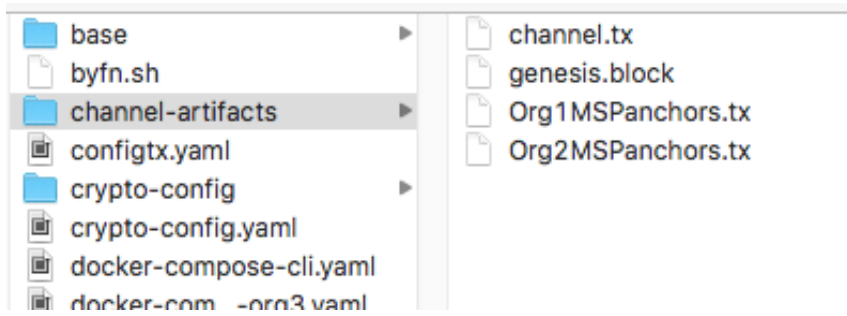
#####
### Generating channel configuration transaction 'channel.tx' ###
#####
+ configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/channel.tx -channelID mychannel
2018-05-05 18:57:30.510 CST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2018-05-05 18:57:30.519 CST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 002 Generating new channel
configtx
2018-05-05 18:57:30.541 CST [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 003 Writing new channel tx
+ res=0
+ set +x

#####
##### Generating anchor peer update for Org1MSP #####
#####
+ configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org1MSPanchors.tx -channelID m
ychannel -asOrg Org1MSP
2018-05-05 18:57:30.595 CST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2018-05-05 18:57:30.606 CST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor pe
er update
2018-05-05 18:57:30.606 CST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer
update
+ res=0
+ set +x

#####
##### Generating anchor peer update for Org2MSP #####
#####
+ configtxgen -profile TwoOrgsChannel -outputAnchorPeersUpdate ./channel-artifacts/Org2MSPanchors.tx -channelID m
ychannel -asOrg Org2MSP
2018-05-05 18:57:30.651 CST [common/tools/configtxgen] main -> INFO 001 Loading configuration
2018-05-05 18:57:30.664 CST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating anchor pe
er update
2018-05-05 18:57:30.664 CST [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anchor peer
update
+ res=0
+ set +x

first-network wenzil$
```

打开"channel-artifacts"目录，会发现多了四个文件



此时，Hyperledger Fabric开发环境基本搭建完毕。

接下来的是创建和加入通道，安装和实例化链码，进行链码的调用等操作。

由于涉及内容过多，可以查看另外一篇文章“《Hyperledger Fabric运行体验（MacOS系统）》做参考”