内部资料

注意保密

Internal Document

Confidential

IBM-Fabric安装部署手册

**编 号： V0.8**

**版 本： 初稿**

**拟制人： 李涛**

**文件修订记录**

**版本修改记录** （A --- 增加 M --- 修改 D --- 删除）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **当前版本** | **日期** | **修改位置**  **（图、表、章节）** | **A/M/D** | **简单描述**  **（原因和修改简介）** | **修改人** | **评审人** |
| 初稿 | 2018-4-21 | ALL | A | 创建 | 李涛 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

[IBM-Fabric安装部署手册 1](#_Toc513029149)

[1. 概述 5](#_Toc513029150)

[1.1. IBM-Fabric版本信息 5](#_Toc513029151)

[1.2. Linux版本信息 5](#_Toc513029152)

[1.3. Docker版本信息 5](#_Toc513029153)

[1.4. GoLang版本信息 5](#_Toc513029154)

[2. Fabric网络结构 6](#_Toc513029155)

[2.1. Chaicode 6](#_Toc513029156)

[2.2. CA 6](#_Toc513029157)

[2.3. Orderer节点 6](#_Toc513029158)

[2.4. Peer节点 6](#_Toc513029159)

[2.5. MSP 6](#_Toc513029160)

[2.6. Channel 6](#_Toc513029161)

[2.7. Gossip 6](#_Toc513029162)

[2.8. Events 6](#_Toc513029163)

[3. 环境构建(外网) 6](#_Toc513029164)

[3.1. Docker安装 6](#_Toc513029165)

[1. 移除旧版本的Docker信息（如果之前安装过docker） 6](#_Toc513029166)

[2. 安装Docker CE 7](#_Toc513029167)

[3. 验证Docker安装结果 7](#_Toc513029168)

[4. Docker启动 7](#_Toc513029169)

[5. Docker开机自启动 7](#_Toc513029170)

[6. Docker配置 8](#_Toc513029171)

[3.2. Docker-Compose安装 9](#_Toc513029172)

[1. 获取Docker-Compose 9](#_Toc513029173)

[2. 验证Docker-Compose版本 9](#_Toc513029174)

[3.3. Go语言安装 9](#_Toc513029175)

[1. 获取GoLang安装包 9](#_Toc513029176)

[2. 设置环境变量 10](#_Toc513029177)

[3.4. Linux内核升级 10](#_Toc513029178)

[1. 获取Linux内核源码 10](#_Toc513029179)

[2. 编译源码（root用户） 10](#_Toc513029180)

[4. 环境构建(内网) 11](#_Toc513029181)

[5. Fabric源码&镜像文件 11](#_Toc513029182)

[5.1. 获取fabric源码 11](#_Toc513029183)

[1. 执行命令： 12](#_Toc513029184)

[2. 执行结果：（源码结构） 12](#_Toc513029185)

[5.2. 获取fabric –samples 12](#_Toc513029186)

[1. 进入到与fabric平级的目录 12](#_Toc513029187)

[2. 下载源码 12](#_Toc513029188)

[3. 进入fabric-samples目录,查看分支并切换到1.1 12](#_Toc513029189)

[5.3. 获取镜像文件 13](#_Toc513029190)

[1. 获取镜像文件 13](#_Toc513029191)

[2. 镜像文件明细 13](#_Toc513029192)

[3. 镜像文件打包 13](#_Toc513029193)

[5.4. 获取2进制文件 14](#_Toc513029194)

[1. 进入到fabric-samples文件 14](#_Toc513029195)

[2. 通过wget下载二进制文件 14](#_Toc513029196)

[3. 解压下载的文件 15](#_Toc513029197)

[4. 配置环境变量 15](#_Toc513029198)

[6. 运行测试（单机） 15](#_Toc513029199)

[6.1. 运行文件 15](#_Toc513029200)

[1. 生成配置文件 15](#_Toc513029201)

[2. 启动项目 15](#_Toc513029202)

[3. 关闭项目 16](#_Toc513029203)

[4. 阿里云环境 16](#_Toc513029204)

[6.2. 测试脚本 16](#_Toc513029205)

[1. 执行部署脚本 17](#_Toc513029206)

[2. 关闭部署脚本 17](#_Toc513029207)

[7. Fabric多节点部署 17](#_Toc513029208)

[7.1. 创建多节点部署 17](#_Toc513029209)

[7.2. 启动多节点部署 17](#_Toc513029210)

[7.3. 多节点集群生产部署 17](#_Toc513029211)

[7.4. 多节点集群生产启动 17](#_Toc513029212)

[8. 智能合约 17](#_Toc513029213)

[9. CouchDB 17](#_Toc513029214)

[10. SDK-Java应用 17](#_Toc513029215)

[11. Orderer分布式方案 17](#_Toc513029216)

# 概述

## IBM-Fabric版本信息

|  |  |  |
| --- | --- | --- |
| **Fabric版本信息** | | |
| **编号** | **版本号** | **说明** |
| 1 | 1.1.0 |  |

## Linux版本信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Linux版本信息** | | | | |
| **编号** | **Linux名称** | **版本信息** | **内核版本号** | **说明** |
| 1 | CentOS | 7 | 3.10以上 | 建议最新的内核稳定版本 |

## Docker版本信息

|  |  |  |  |
| --- | --- | --- | --- |
| **Docker版本信息** | | | |
| **编号** | **名称** | **版本信息** | **说明** |
| 1 | Docker | 18.04.0-ce |  |
| 2 | Docker-Compose | 1.21.0 |  |

## GoLang版本信息

|  |  |  |
| --- | --- | --- |
| **GoLang版本信息** | | |
| **编号** | **版本号** | **说明** |
| 1 | 1.8.3 | linux/amd64 |

# Fabric网络结构

## Chaicode

## CA

## Orderer节点

## Peer节点

## MSP

## Channel

## Gossip

## Events

# 环境构建(外网)

## Docker安装

### 移除旧版本的Docker信息（如果之前安装过docker）

**运行命令：**

sudo yum remove docker

sudo yum remove docker-common

sudo yum remove docker-selinux

sudo yum remove docker-engine

### 安装Docker CE

**运行命令：（yum模式安装Docker CE）**

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

sudo yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

sudo yum-config-manager --enable docker-ce-edge

sudo yum-config-manager --enable docker-ce-test

sudo yum-config-manager --disable docker-ce-edge

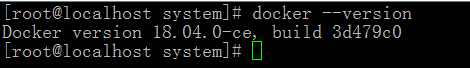
sudo yum makecache fast

sudo yum install docker-ce

**网络原因，可能会比较慢。**

### 验证Docker安装结果

执行查询docker版本号，看是否安装成功，结果如下：

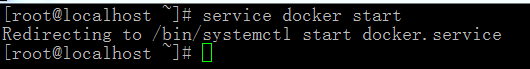


### Docker启动

**执行命令：**

Service docker start

**执行结果：**



### Docker开机自启动

**执行命令：**

chkconfig docker on

### Docker配置

#### 设置DOCKER\_OPTS

**执行命令：**

vi /etc/default/docker

**docker文件中，增加如下内容：**

DOCKER\_OPTS="-H tcp://127.0.0.1:5096 -H unix:///var/run/docker.sock"

**保存退出**

#### 设置DOCKER\_HOST

**在应用用户下，打开.bashrc文件，设置DOCKER\_HOST环境变量：**

export DOCKER\_HOST=tcp://localhost:5096

**执行命令，使环境变量生效：**

source .bashrc

**查看环境变量是否生效：**

env|grep HOST

#### docker.service配置文件

**打开docker.service文件：**

vi /usr/lib/systemd/system/docker.service

**配置【Service】：**

**增加：**

EnvironmentFile=-/etc/sysconfig/docker

ExecStart=/usr/bin/dockerd-current \

$DOCKER\_OPTS

**配置【Install】：**

**增加：**

EnvironmentFile=-/etc/sysconfig/docker

#### 重新加载Docker配置

**执行命令：**

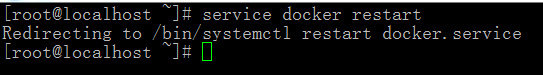
systemctl daemon-reload

#### 重启服务

**执行命令：**

service docker restart

**执行结果：**

****

## Docker-Compose安装

### 获取Docker-Compose

1. **获取应用程序，执行命令：**

curl -L https://github.com/docker/compose/releases/download/1.21.0/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

1. **增加执行权限：**

Chmod +x /usr/local/bin/docker-compose

1. **拷贝执行程序：**

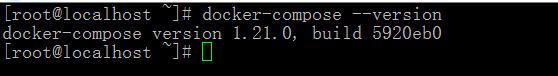
cp /usr/local/bin/docker-compose /usr/bin

### 验证Docker-Compose版本

**执行命令：**

docker-compose –version

**执行结果：**



## Go语言安装

### 获取GoLang安装包

1. **获取安装包 ，执行：**

curl -O https://storage.googleapis.com/golang/go1.8.3.linux-amd64.tar.gz

1. **解压**

tar -C /usr/local -xzf go1.8.3.linux-amd64.tar.gz

### 设置环境变量

**编辑/etc/profile:**

vi /etc/profile

**增加环境变量:**

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

export GOPATH=/home/fabric --**根据实际情况配置**

source /etc/profile

## Linux内核升级

### 获取Linux内核源码

1. **在Linux官网获取最新的Kernel源码：(以4.16.4为例)**

[www.kernel.org](http://www.kernel.org)

**源码文件：**

linux-4.16.4.tar.xz

1. **上传到服务器**

**执行命令：**

xz –d linux-4.16.4.tar.xz

tar xvf linux-4.16.4.tar

### 编译源码（root用户）

1. **清理老版本的相关配置**

**执行名命令：**

make mrproper

1. **安装依赖包**

**执行命令：**

yum install ncurses ncurses-devel --install the ncurses

yum install elfutils-libelf-devel

yum install openssl-devel.x86\_64

yum install bc

1. **生成配置文件**

**执行命令：**

make menuconfig

**进入配置界面：**

1. **编译源码**

**执行命令：**

make

1. **组件安装**

**执行命令：**

make modules\_install

1. **内核安装**

**执行命令：**

make install

1. **修改内核启动项**

* **CentOS 6**

vi /etc /grub.conf:

default = 0

* **CentOS 7**

**察看内核版本列表:**

cat /boot/grub2/grub.cfg |grep menuentry

配置从默认内核启动，下面命令的内核名称根据系统内部查到的实际名称来替换:

grub2-set-default "CentOS Linux (4.16.4) 7 (Core)"

验证是否配置成功：

 grub2-editenv list

1. **重新启动系统**

**执行命令:**

**reboot**

# 环境构建(内网)

# Fabric源码&镜像文件

## 获取fabric源码

**执行命令：（github上获取）**

go get github.com/hyperledger/fabric

cd ~/src/github.com/hyperledger/fabric/

git checkout -b v1.1.0

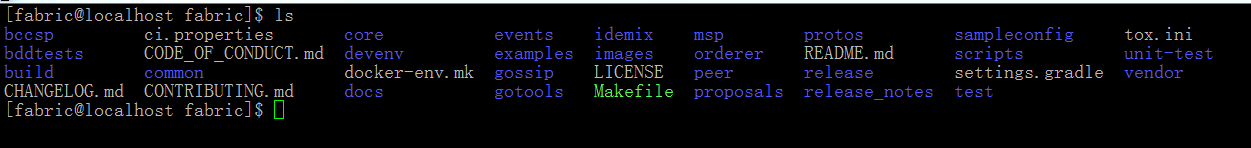
### 执行命令：

go get github.com/hyperledger/fabric

cd ~/src/github.com/hyperledger/fabric/

git checkout -b v1.1.0

### 执行结果：（源码结构）

****

## 获取fabric –samples

### 进入到与fabric平级的目录

cd $GOPATH/src/github.com/hyperledger

### 下载源码

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

### 进入fabric-samples目录,查看分支并切换到1.1

cd fabric-samples/

git branch -a

git checkout release-1.1

## 获取镜像文件

### 获取镜像文件

**执行脚本：**

~/src/github.com/hyperledger/fabric/script/bootstrap.sh

### 镜像文件明细

**镜像文件明细：**

hyperledger/fabric-tools

hyperledger/fabric-orderer

hyperledger/fabric-peer

hyperledger/fabric-couchdb

hyperledger/fabric-kafka

hyperledger/fabric-ca

hyperledger/fabric-ccenv

hyperledger/fabric-baseimage

### 镜像文件打包

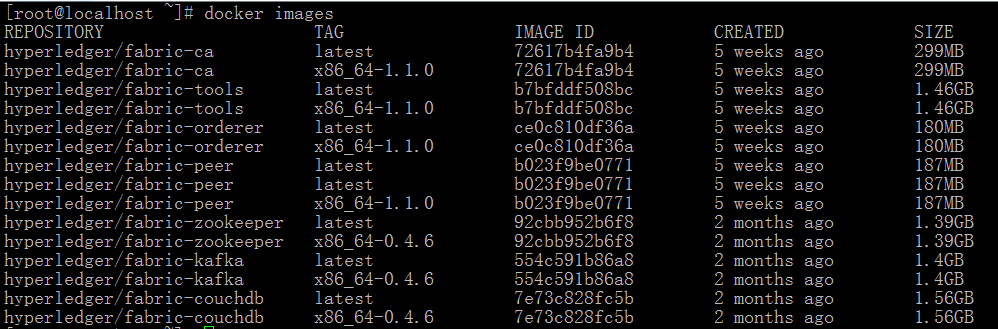
HyperLedger/Fabric镜像数量较多且容量需求大，一套基本的服务镜像可达10G左右，如果在多台服务器上部署，会耽误很多时间。因此，对于上述已经下载至本地的镜像，我们需要使用docker save命令来备份，并通过scp命令来将这些镜像文件拷贝至其他服务器。再通过docker load命令来导入。

1. **查看镜像ID**

**执行命令：**

docker images

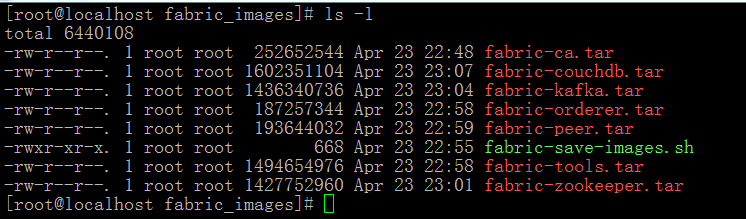
**执行结果：**



1. **保存镜像文件：**

docker save IMAGEID > FileName ------把对应ID的镜像文件保存到本地的压缩包

**镜像文件明细：**

****

1. **导入镜像文件：**

**执行命令：**

docker load < 镜像文件名

## 获取2进制文件

### 进入到fabric-samples文件

cd fabric-samples/

### 通过wget下载二进制文件

wget https://nexus.hyperledger.org/content/repositories/releases/org/hyperledger/fabric/hyperledger-fabric/linux-amd64-1.1.0/hyperledger-fabric-linux-amd64-1.1.0.tar.gz

### 解压下载的文件

tar -zxvf hyperledger-fabric-linux-amd64-1.1.0.tar.gz

会发现多了bin和config两个个文件夹：

### 配置环境变量

root用户,修改/etc/profile：

增加PATH路径：

C:\Users\thinkpad\AppData\Local\Temp\1524790408(1).png

# 运行测试（单机）

## 运行文件

### 生成配置文件

**进到fabric-samples/first-network：**

cd fabric-samples/first-network/

**生成配置文件：**

./byfn.sh -m generate

### 启动项目

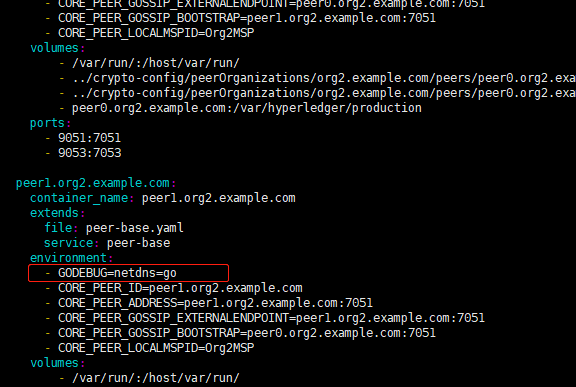
./byfn.sh -m up

### 关闭项目

./byfn.sh -m down

### 阿里云环境

阿里云服务器要修改base文件夹的docker-compose-base.yaml，所有的environment下添加 -GODEBUG=netdns=go,如下图：



## 测试脚本

**examples/e2e\_cli/network\_setup.sh:**

network\_setup.sh是一个测试脚本，该脚本启动5个docker容器，其中4个容器运行peer节点和1个容器运行orderer节点，它组成一个Fabric集群。另外还有一个cli容器用于执行创建channel、加入channel、安装和执行chaincode等操作。

测试用的chaincode定义了两个变量，在实例化的时候会给这两个变量赋予了初始值，并通过invoke操作可以使两个变量的值发生变化。

### 执行部署脚本

### 关闭部署脚本

# Fabric多节点部署

## 创建多节点部署

## 启动多节点部署

## 多节点集群生产部署

## 多节点集群生产启动

# 智能合约

# CouchDB

# SDK-Java应用

# Orderer分布式方案