

金融服务平台——数据区块链存储

部署文档

目录

| | |
|-----------------------------|---|
| 一.部署环境..... | 1 |
| 1.1 服务器配置..... | 1 |
| 1.2 客户机配置..... | 1 |
| 1.2.1 客户端机器配置要求..... | 1 |
| 1.2.2 客户端系统软件环境..... | 1 |
| 二.系统安装与配置..... | 1 |
| 2.1 总体说明..... | 1 |
| 2.2 启动网络..... | 1 |
| 2.3 停止网络..... | 2 |
| 2.4 重启网络..... | 2 |
| 2.5 终止（删除）网络..... | 2 |
| 2.6 实现区块链操作..... | 2 |
| 2.7 调试网络..... | 2 |
| 2.8 调用 java 应用以启动区块链服务..... | 2 |
| 三.软件环境配置参考..... | 2 |
| 3.1 cURL 配置..... | 2 |
| 3.2 GO 安装配置..... | 3 |
| 3.3 Docker 配置..... | 3 |
| 3.4 Docker Compose 配置..... | 3 |

一.部署环境

1.1 服务器配置

- 服务器推荐使用 Ubuntu 16.04 LTS 作为服务器系统；
- 服务器应保证至少有 2G 空余内存和 10G 左右的空闲硬盘空间；
- 服务器应保证空闲时有至少 1 核左右的空余 CPU；
- 服务器应保证互联网通畅并拥有公网 IP；
- 服务器应保证有一块与公网 IP 相对应的网卡并且有空闲端口。

1.2 客户机配置

1.2.1 客户端机器配置要求

硬件设施需要能够满足前端运行要求。

1.2.2 客户端系统软件环境

- 操作系统: Mac OSX, *nix, Windows(需在 WSL 内)
- cURL: 7.54.0 及以上版本
- Docker: 17.06.2-ce 及以上版本
- Docker Compose: 1.14.0 及以上版本
- Go: 1.12.x 及以上
- JRE 8.0 及以上

二.系统安装与配置

2.1 总体说明

在管理员权限下，启动 Docker 后，运行 basic-network/start.sh 脚本启动底层区块链网络，并使用 BlockChain 接口对区块链进行查询与调用。

2.2 启动网络

启动 Docker 后，运行 basic-network/start.sh 脚本，运行 basic-network/start.sh

脚本将自动启动区块链网络，创建通道(Channel)，并将各个底层节点(Peer)添加到通道中。

2.3 停止网络

若要使得网络停止，则运行脚本 `basic-network/stop.sh`。

2.4 重启网络

若要在网络停止时重启网络，则运行 `basic-network/restart.sh` 脚本即可，重启网络后区块链恢复正常工作状态。

2.5 终止（删除）网络

运行脚本 `basic-network/teardown.sh` 即可在系统中彻底终止（删除）网络。

2.6 实现区块链操作

使用 `BlockChain` 目录下的 `Java SDK` 可实现对区块链的操作，其中的接口提供了十个与区块链进行交互的方法，详见《用户使用说明书》。

2.7 调试网络

通过 `docker exec -it cli bash` 命令进入 `CLI` 容器调试网络。

2.8 调用 java 应用以启动区块链服务

对编译好的 `jar` 文件夹 `Blockchain_jar.zip` 进行解压

进入文件目录后，执行命令 `java -jar Blockchain.jar`

三.软件环境配置参考

3.1 cURL 配置

`cURL` 是一个基于 `URL` 规则在命令行下工作的文件传输工具，支持上传和下载文件。

可在 <https://curl.haxx.se/download.html> 获取对应操作系统的发行版本，安装后即可使用。

若要使用 Windows 操作系统，可以给 Windows 增加 curl 命令的环境变量，增加 CURL_HOME 环境变量，给 PATH 环境变量加上%CURL_HOME%。即可在命令窗口的任意目录下使用 curl 命令。

3.2 GO 安装配置

Go（又称 Golang）是 Google 开发的一种静态强类型、编译型、并发型，并具有垃圾回收功能的编程语言。

下载对应系统的安装包并将其解压缩到 /usr/local，在 /usr/local/go 中创建一个 Go 树。例如：

```
tar -C /usr/local -xzf go$VERSION.$OS-$ARCH.tar.gz
```

注意，如果要在 Linux 上为 64 位 x86 安装 Go 版本 1.2.1，则所需的存档称为 go1.2.1.linux-amd64.tar.gz，以此类推（通常这些命令必须以 root 身份或通过 sudo 运行。）

将 /usr/local/go/bin 添加到 PATH 环境变量。此操作可以通过将下面这行指令添加到/etc/profile（用于系统范围的安装）或 \$HOME/.profile 来实现：

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

3.3 Docker 配置

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，而且也可以实现虚拟化。通过 Docker 可以简化部署多种应用实例工作。

对于 Docker 的配置，以 Ubuntu 为例：

获取最新版本的 Docker 安装包：\$ wget -qO- https://get.docker.com/ | sh

启动 docker 后台服务：\$ sudo service docker start

3.4 Docker Compose 配置

Docker Compose 是运行多容器 Docker 应用程序的工具。

以 Ubuntu 为例：pip 安装：\$ sudo pip install docker-compose

安装 curl 后也可使用以下命令安装配置：

```
# curl -L
```

```
https://github.com/docker/compose/releases/download/1.14.0/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose
```