

Substrate环境部署--基于Docker

开发环境

- Centos7
- Docker

Substrate

开发之前我们需要明确几个东西，首先是需要对substrate是什么有一个了解，这里只做简单介绍，如果需要更详细的信息，请参考[What is substrate](#)

或者参考我之前写的一篇关于substrate的架构:[substrate架构解析](#)

那么substrate到底是什么呢？简单来说，它就是一个框架，是区块链的开发框架，能够帮你快速实现个性化区块链的开发，它帮你实现了区块链通用的一些底层模块如共识、网络、数据流等。你可以简单将它理解为java界的spring框架就会直观很多，也就是说我们只需要专注于业务本身，而不是再去考虑一些底层的细节设计。

环境部署

在开始正式开发之前我们需要先进行环境部署，这里所有的环境部署都基于Docker来实现。

- 拉取ubuntu镜像

```
$ docker pull ubuntu
```

- 通过镜像去启动容器，并取名为substrate，并且将容器的80端口映射到本地的9900

```
$ docker run -d -it --name substrate -p 9900:80 ubuntu
```

- 进入容器

```
$ docker exec -it substrate bash
```

- 修改apt源，修改/etc/apt/sources.list

```
deb http://cn.archive.ubuntu.com/ubuntu bionic main multiverse restricted universe
deb http://cn.archive.ubuntu.com/ubuntu bionic-updates main multiverse restricted
universe
deb http://cn.archive.ubuntu.com/ubuntu bionic-security main multiverse restricted
universe
deb http://cn.archive.ubuntu.com/ubuntu bionic-proposed main multiverse restricted
universe
```

- 更新apt源

```
$ apt update
```

- 下载vim、curl、**clang**（不然部署substrate过程会报错）、wget

```
$ apt install -y vim curl clang wget
```

- 下载node并进行编译部署，过程耗时较长

```
$ wget https://nodejs.org/dist/v10.15.3/node-v10.15.3.tar.gz
$ tar zxvf node-v10.15.3.tar.gz
$ cd node-v10.15.3
$ ./configure
$ make && make install
```

结束之后进行测试

```
root@6db0590b60e6:~# node -v
v10.15.3
root@6db0590b60e6:~# npm -v
6.4.1
```

- 下载并安装substrate

```
$ curl https://getsubstrate.io -ssf | bash
```

我们来简单看一看这个命令里面到底做了些什么事情.脚本源码: getsubstrate.io

更新apt源并安装相关依赖包

```
elif [ -f /etc/debian_version ]; then
    echo "Ubuntu/Debian Linux detected."
    $MAKE_ME_ROOT apt update #更新apt源
    $MAKE_ME_ROOT apt install -y cmake pkg-config libssl-dev git gcc build-essential
    clang libclang-dev #下载相关依赖包
```

安装rust环境

```
if ! which rustup >/dev/null 2>&1; then
    curl https://sh.rustup.rs -ssf | sh -s -- -y
    source ~/.cargo/env
else
    rustup update
fi
```

安装substrate

```
cargo install --force --git https://github.com/paritytech/substrate subkey #安装subkey
cargo install --force --git https://github.com/paritytech/substrate substrate #安装
substrate

f=`mktemp -d`
git clone https://github.com/paritytech/substrate-up $f #安装substrate相关脚本
cp -a $f/substrate-* ~/.cargo/bin
```

- 更新环境

通过研究脚本源码我们已经发现其实它只是帮助我们部署好了环境，但是最后安装完substrate之后并没有更新环境

```
$ source ~/.cargo/env
```

安装成功之后进行测试，如果没有问题代表已经安装成功了

```
root@6db0590b60e6:~# rustc --version
rustc 1.33.0 (2aa4c46cf 2019-02-28)
root@6db0590b60e6:~# substrate --version
substrate 0.11.0-88772b50-x86_64-linux-gnu
root@6db0590b60e6:~# substrate-node-new
Usage: substrate-node-new <NAME> <AUTHOR>
```

然后我们可以简单看一下.cargo中bin目录下面的文件

```
root@6db0590b60e6:~# tree ~/.cargo/bin/
/root/.cargo/bin/
|-- cargo
|-- cargo-clippy
|-- cargo-fmt
|-- cargo-miri
|-- clippy-driver
|-- rls
|-- rust-gdb
|-- rust-lldb
|-- rustc
|-- rustdoc
|-- rustfmt
|-- rustup
|-- subkey
|-- substrate
|-- substrate-module-new
|-- substrate-node-new
|-- substrate-ui-new
`-- wasm-gc
```

至此，我们的项目环境已经成功部署了，我们可以先将docker镜像简单保存一下。新开一个窗口

```
$ docker commit -m "substrate dev environment" -a "作者" 打包的镜像名:版本号
```

以下是我的操作

```
$ docker commit -m "substrate dev environment" -a "sherlzp" sherlzp/substrate:0.1
```

然后将镜像push到自己的dockerhub上面即可

```
$ docker push sherlzp/substrate:0.1
```

一步到位

由于搭建环境过程当中会遇到各种各样的问题，而且没有科学上网下载速度特别特别慢，所以如果各位想一步到位可以直接使用我已经搭建好的镜像。

```
$ docker pull sherlzp/substrate:0.3
$ docker run -d -it --name substrate -p 8800:80 -p 8000:8000 -p 9944:9944
sherlzp/substrate:0.1
$ docker exec -it substrate bash
进入容器之后
$ source ~/.cargo/env
```