

## 用 Geth 搭建以太坊私链

这节课让我们用 Geth 来搭建一个属于自己的以太坊私链。

### 安装 Geth

安装 Geth 有很多种方式，这里主要就 Linux 环境给出两种：系统包管理器（apt-get）安装和源码安装。更加推荐大家用源码安装，在整个过程中可以看到 Geth 各组件的构建步骤。

#### 一、apt-get

```
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository -y ppa:ethereum/ethereum
$ sudo apt-get update
$ sudo apt-get install ethereum
```

#### 二、源码安装

##### 1. 克隆 github 仓库

我们的第一步是克隆 git 仓库，以获取源代码的副本。

```
$ git clone https://github.com/ethereum/go-ethereum.git
```

##### 2. 从源码构建 Geth

要构建 Geth，切换到下载源代码的目录并使用 make 命令：

```
$ cd go-ethereum
$ make geth
```

如果一切顺利，我们将看到 Go 编译器构建每个组件，直到它生成 geth 可执行文件：

```
build/env.sh go run build/ci.go install ./cmd/geth
>>> /usr/local/go/bin/go install -ldflags -X
main.gitCommit=58a1e13e6dd7f52a1d5e67bee47d23fd6cfdee5c -v ./cmd/geth
github.com/ethereum/go-ethereum/common/hexutil
github.com/ethereum/go-ethereum/common/math
github.com/ethereum/go-ethereum/crypto/sha3 github.com/ethereum/go-ethereum/rlp
github.com/ethereum/go-ethereum/crypto/secp256k1
github.com/ethereum/go-ethereum/common [...]
github.com/ethereum/go-ethereum/cmd/utls
```

```
github.com/ethereum/go-ethereum/cmd/geth Done building. Run "build/bin/geth" to
launch geth.
```

查看 geth version，确保在真正运行之前安装正常：

```
$ ./build/bin/geth version
Geth
Version: 1.8.0-unstable
Git Commit: e37f7be97e47a032d723db16d8b195998547805a
Architecture: amd64
Protocol Versions: [63 62]
Network Id: 1
Go Version: go1.9
Operating System: linux
GOPATH=/home/ubuntu/project
GOROOT=/usr/local/go
```

## 启动节点同步

安装好了 Geth，现在我们可以尝试运行一下它。执行下面的命令，geth 就会开始同步区块，并存储在当前目录下。这里的 `--syncmode fast` 参数表示我们会以“快速”模式同步区块。在这种模式下，我们只会下载每个区块头和区块体，但不会执行验证所有的交易，直到所有区块同步完毕再去获取一个系统当前的状态。这样就节省了很多交易验证的时间。

```
$ geth --datadir . --syncmode fast
```

通常，在同步以太坊区块链时，客户端会一开始就下载并验证每个块和每个交易，也就是说从创世区块开始。毫无疑问，如果我们不加 `--syncmode fast` 参数，同步将花费很长时间并且具有很高的资源要求（它将需要更多的 RAM，如果你没有快速存储，则需要很长时间）。

有些文章会把这个参数写成 `--fast`，这是以前快速同步模式的参数写法，现在已经被 `--syncmode fast` 取代。

如果我们想同步测试网络的区块，可以用下面的命令：

```
$ geth --testnet --datadir . --syncmode fast
```

`--testnet` 这个参数会告诉 geth 启动并连接到最新的测试网络，也就是 Ropsten。测试网络的区块和交易数量会明显少于主网，所以会更快一点。但即使是用快速模式同步测试网络，也会需要几个小时的时间。

## 搭建自己的私有链

因为公共网络的区块数量太多，同步耗时太长，我们为了方便快速了解 Geth，可以试着用它来搭一个只属于自己的私链。

首先，我们需要创建网络的“创世”（genesis）状态，这写在一个小小的 JSON 文件里（例如，我们将其命名为 genesis.json）：

```
{
  "config": {
    "chainId": 15
  },
  "difficulty": "2000",
  "gasLimit": "2100000",
  "alloc": {
    "7df9a875a174b3bc565e6424a0050ebc1b2d1d82": { "balance": "300000" },
    "f41c74c9ae680c1aa78f42e5647a62f353b7bdde": { "balance": "400000" }
  }
}
```

要创建一条以它作为创世块的区块链，我们可以使用下面的命令：

```
geth --datadir path/to/custom/data/folder init genesis.json
```

在当前目录下运行 geth，就会启动这条私链，注意要将 networked 设置为与创世块配置里的 chainId 一致。

```
geth --datadir path/to/custom/data/folder --networkid 15
```

我们可以看到节点正常启动：

```
WARN [10-23|02:38:19] No etherbase set and no accounts found as default
INFO [10-23|02:38:19] Starting peer-to-peer node
instance=Geth/v1.8.0-unstable-e37f7be9/linux-amd64/go1.9
...
INFO [10-23|02:38:21] IPC endpoint opened:
/home/ubuntu/project/go_ethereum_test/geth.ipc
INFO [10-23|02:38:21] Mapped network port                proto=tcp
export=30303 intport=30303 interface="UPNP IGDv1-IP1"
```

恭喜！我们已经成功启动了一条自己的私链。

尚硅谷