

Geth 실습

이민현

minbba@chainstream.io



목표

Geth의 설치부터 Mining까지 ~

Session 1에서 공부한 이론을 Geth 실습을 통해 이해



Geth

이더리움 CLI 클라이언트

- 이더리움의 전체 기능을 사용할 수 있는 풀 클라이언트로서 다중 인터페이스 제공
: [커맨드라인 모드](#) 의 부가 명령어로 [대화형 자바스크립트 콘솔](#) 과 [JSON-RPC 서버](#)

- Geth 커맨드라인 : `geth [options] command [--command options] [arguments...]`

```
$ geth --identity "JayChain" --rpc --rpcport "8800" --rpccorsdomain "*" --datadir "/Users/jay/privatechain" --port "30303" --nodiscover --rpcapi "db,eth,net,web3" --networkid 2017 console
```

--identity "JayChain" // 내 프라이빗 노드 식별자.

--rpc // RPC 인터페이스 가능하게 함.

--rpcport "8800" // RPC 포트 지정

--rpccorsdomain "*" // 접속가능한 RPC 클라이언트 URL 지정 , 가능한 *(전체 허용) 보다는 URL을 지정하는 게 보안상 좋음.

--datadir "/Users/jay/privatechain" // 커스텀 디렉토리 지정

--port "30303" // 네트워크 Listening Port 지정

--nodiscover // 같은 제네시스 블록과 네트워크 ID에 있는 블록에 연결 방지

--rpcapi "db,eth,net,web3" // RPC에 의해서 접근을 허락할 API

--networkid 2017

console // 출력을 콘솔로 함.

Geth & eth

이더리움 CLI 클라이언트

- 대화형 자바스크립트 옵션 (console , attach , js)

- 이더리움은 자바스크립트 런타임 환경(JSRE)을 대화형 콘솔 또는 비대화형 스크립트 모드로 제공

```
$ geth console // 콘솔에서 커맨드 라인 모드로 JSRE 사용
```

```
$ geth attach [ipc:/some/custom/path | rpc:http://127.0.0.1:8545] // 작동중인 geth 노드의 콘솔에 연결 후 사용
```

```
$ geth console 2 >> /dev/null // 화면에 log를 출력하지 않는다.
```

```
$ geth js demo.js 2 >> geth.log // 비대화형 모드, demo.js 파일을 실행한다.
```

- 커맨드라인 모드 : JSON-RPC 서버 옵션

- JSON-RPC는 경량 RPC 프로토콜을 통해 이더리움을 호출하고 그 결과를 JSON 포맷으로 받음.
 - 이더리움 RPC 인터페이스를 제공하는 **Web3.js 라이브러리를 사용**하여 자바 스크립트 응용 프로그램내에서 이더리움 노드 호출
 - 기본 JSON-RPC 기본 엔드포인트 : <http://localhost:8545>
 - Geth 구동 : \$ geth --rpc --rpcaddr <IP주소> --rpcport <포트번호>
 - Geth 콘솔: >> admin.startRPC(IP주소, 포트번호)

Step 1. 시작하기

Geth 설치 및 실행

Step 1. Geth 설치 및 실행

1. Geth 다운로드

운영체제에 맞는 Geth 클라이언트를 다운로드하세요.[링크](#)

2. Geth 설치하기

Windows: 인스톨러의 가이드에 따라 설치하세요.

Mac: 원하는 디렉토리에 설치하세요.

3. Geth 실행하기

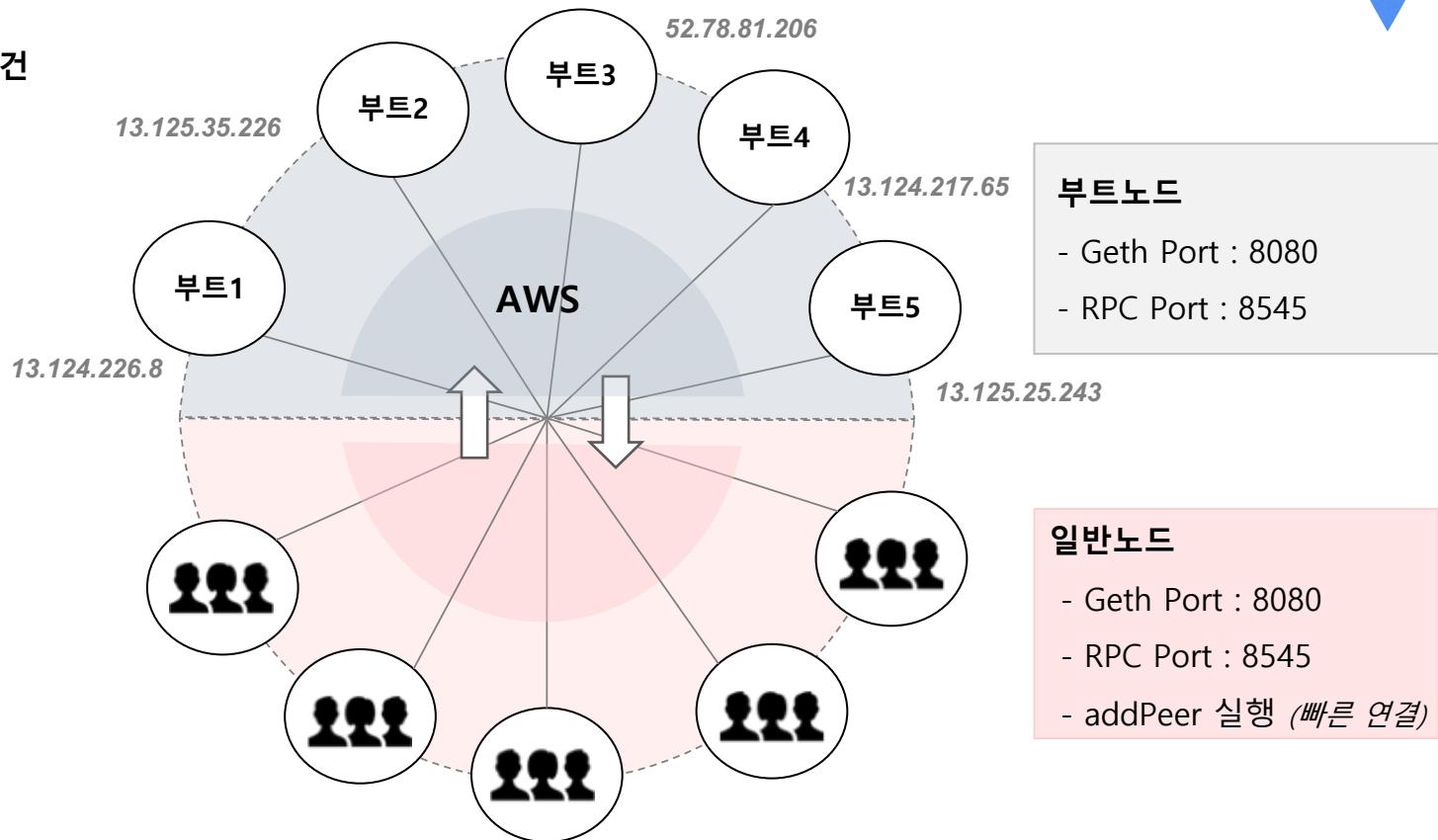
콘솔에 아래와 같이 입력하여 Geth를 실행하세요.

```
geth console
```

[참고] 실습용 프라이빗 네트워크 환경

* 프라이빗 네트워크 조건

- 동일한
genesis.json
- 동일한
Networkid : 15



Step 2. Private network 구성하기

Custom Genesis Block 생성하기

Step 2. Private network 접속(1)

1. Genesis.json 다운로드

실습용 Private chain에 연결하기 위해 Genesis.json 파일을 다운로드하세요.[링크](#)

※ Genesis.json 다운로드 Directory(권장)

Windows: C:\Users\Username

Mac: Users\Username

2. Genesis initialization

콘솔에 아래와 같이 입력하여 Geth를 init하세요.

```
geth --datadir=".ether_study" init genesis.json
```

Step 2. Private network 구성하기

Custom Genesis Block 생성하기

Step 3. Private network 접속(2)

1. Console mode로 Geth 실행하기

콘솔에 아래와 같이 입력하여 Geth를 실행하세요.

```
geth --datadir=".ether_study" --networkid 15 --port 8080 console
```

Step 2. Private network 구성하기

Custom Genesis Block 생성하기

2. CLI를 이용해 Bootnode에 직접 접속하기

콘솔에 아래와 같이 입력하여 Bootnode에 직접 접속하세요.

Bootnode 1 좌석번호 1번 ~ 8번까지 1번 Bootnode에 접속해주세요.

```
admin.addPeer("enode://ae34a288e3a61b7c161bf2c407b2b44dc9161f9277fb2f0e0ce1a729be999f  
50dbf2054a5dfb925b5e81317e8b4803a5886cd3c346cc32a78693764d6511437@13.124.226.8:8080")
```

Step 3. CLI 실습

1. Account 생성하기

콘솔에 아래와 같이 입력하여 Account를 생성하세요.

```
personal.newAccount()
```

2. 생성한 Account 확인하기

콘솔에 아래와 같이 입력하여 보유한 Account의 List를 확인할 수 있습니다.

```
personal.listAccounts
```

Step 3. CLI 실습

3. 생성한 Account의 Balance 확인하기

콘솔에 아래와 같이 입력하여 생성한 Account의 Balance를 확인할 수 있습니다.

```
eth.getBalance(personal.listAccounts[0])
```

4. 생성한 Account 삭제하기

실습을 위해 ./ether_study/keystore 에 있는 UTC--... 파일을 삭제하세요.

5. 실습용 Account 다운로드하기

이 [링크](#)에서 본인의 좌석번호와 같은 실습용 Account를 다운로드한 후 ./ether_study/keystore에 압축을 해제하세요.

Step 3. CLI 실습

6. 실습용 Account의 Balance 확인하기

콘솔에 아래와 같이 입력하여 실습용 Account의 Balance를 확인할 수 있습니다.

```
eth.getBalance(personal.listAccounts[0])
```

Step 4. Transaction의 처리과정

1. Ether transfer

콘솔에 아래와 같이 입력하여 강사의 Account로 1000ETH를 송금하세요.

```
personal.sendTransaction({from:personal.listAccounts[0],to:"0x968a86f451ebc6a1ed58793e1be74183f17aba32",value:web3.toWei(100,"ether")}, "1")
```

Step 4. Transaction의 처리과정

2. Pending Tx 확인하기

아직 블럭에 담기지 못한 Tx은 각 노드 Local의 Transaction pool에 저장됩니다.

Txpool 상태확인

```
txpool.status
```

Txpool 상세내역 확인

```
txpool.content
```

송금한 금액과 수수료 확인

```
txpool.inspect
```

Step 4. Transaction의 처리과정

3. Mining으로 Tx을 Block에 담아 처리하기(강사만 진행합니다.)

Block reward를 수취할 Account를 coinbase로 세팅

```
miner.setEtherbase(personal.listAccounts[0])
```

Mining 시작

```
miner.start()
```

Step 4. Transaction의 처리과정

Coinbase Account의 Balance 증가 확인

```
eth.getBalance(personal.listAccounts[0])
```

Mining한 Block 확인

```
eth.getBlock(확인할 블럭 넘버)
```

Step 4. Transaction의 처리과정

강사의 Account Balance 증가 확인

```
eth.getBalance(personal.listAccounts[0])
```

Q&A

