Blockchain #13

Development Tools for Smart Contract

Prof. Byung Il Kwak

Outline

- Geth
- Ganache
- Node.js
- Truffle
- Visual studio code
- METAMASK

Geth: Go Ethereum

- Geth
  - Go Ethereum은 프로그래밍 언어인 고(Go)에서 구현된 전체 이더넷 노드를 실행하기 위한 프로그램
    - 이더리움 재단에서 제공하는 공식 클라이언트 소프트웨어
    - Geth를 이용해 스마트 컨트랙트 실행을 수행
    - Go, C++, python 등 다양한 언어로 구동할 수 있는 클라이언트들이 개발됨

https://geth.ethereum.org/downloads/

Geth: Go Ethereum

- Geth

Go Ethereum Install Downloads Documentation Search site...

Download Geth – Vallhallan Threshold (v1.10.12) – Release Notes

You can download the latest 64-bit stable release of Geth for our primary platforms below. Packages for all supported platforms, as well as develop builds, can be found further down the page. If you're looking to install Geth and/or associated tools via your favorite package manager, please check our installation guide.

Geth 1.10.12 for Linux | Geth 1.10.12 for macOS | Geth 1.10.12 for Windows | Geth 1.10.12 sources

Specific Versions

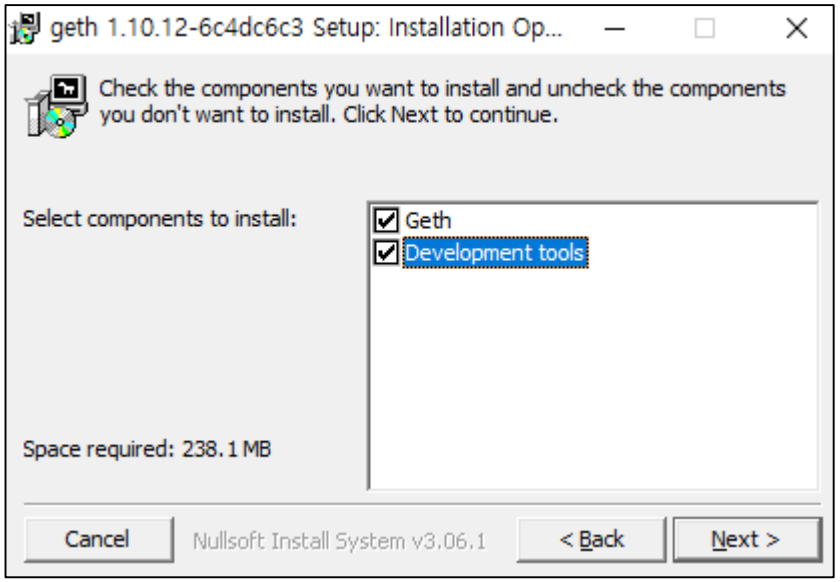
If you're looking for a specific release, operating system or architecture, below you will find:

- All stable and develop builds of Geth and tools
- Archives for non-primary processor architectures
- Android library archives and iOS XCode frameworks

Please select your desired platform from the lists below and download your bundle of choice. Please be aware that the `MD5` checksums are provided by our binary hosting platform (Azure Blobstore) to help check for download errors. **For security guarantees please verify any downloads via the attached PGP signature files** (see OpenPGP Signatures for details).

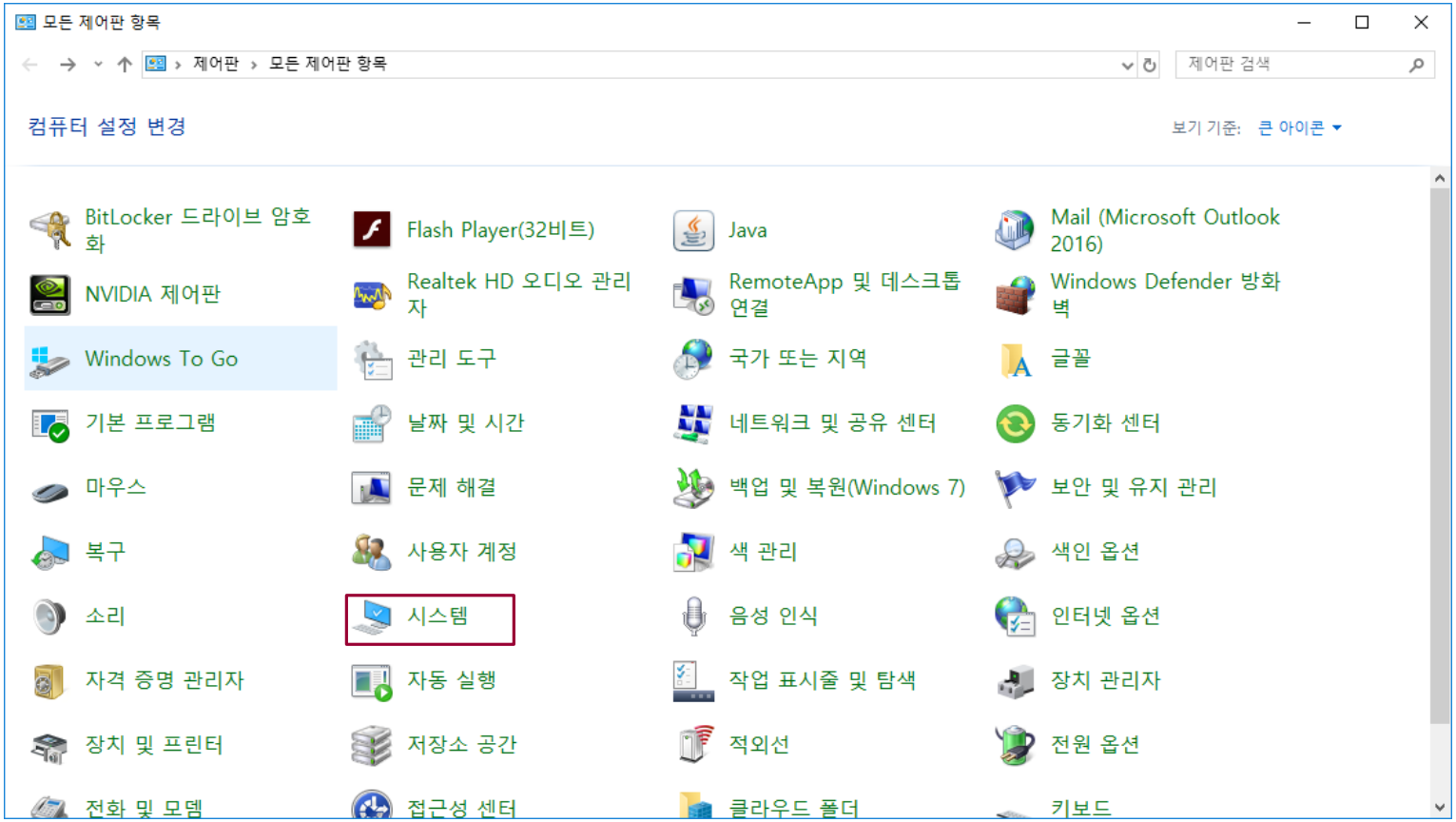
Geth: Go Ethereum

- Geth



Geth: Go Ethereum

- Geth
  - 설치시 에러 발생할 경우, 경로 설정

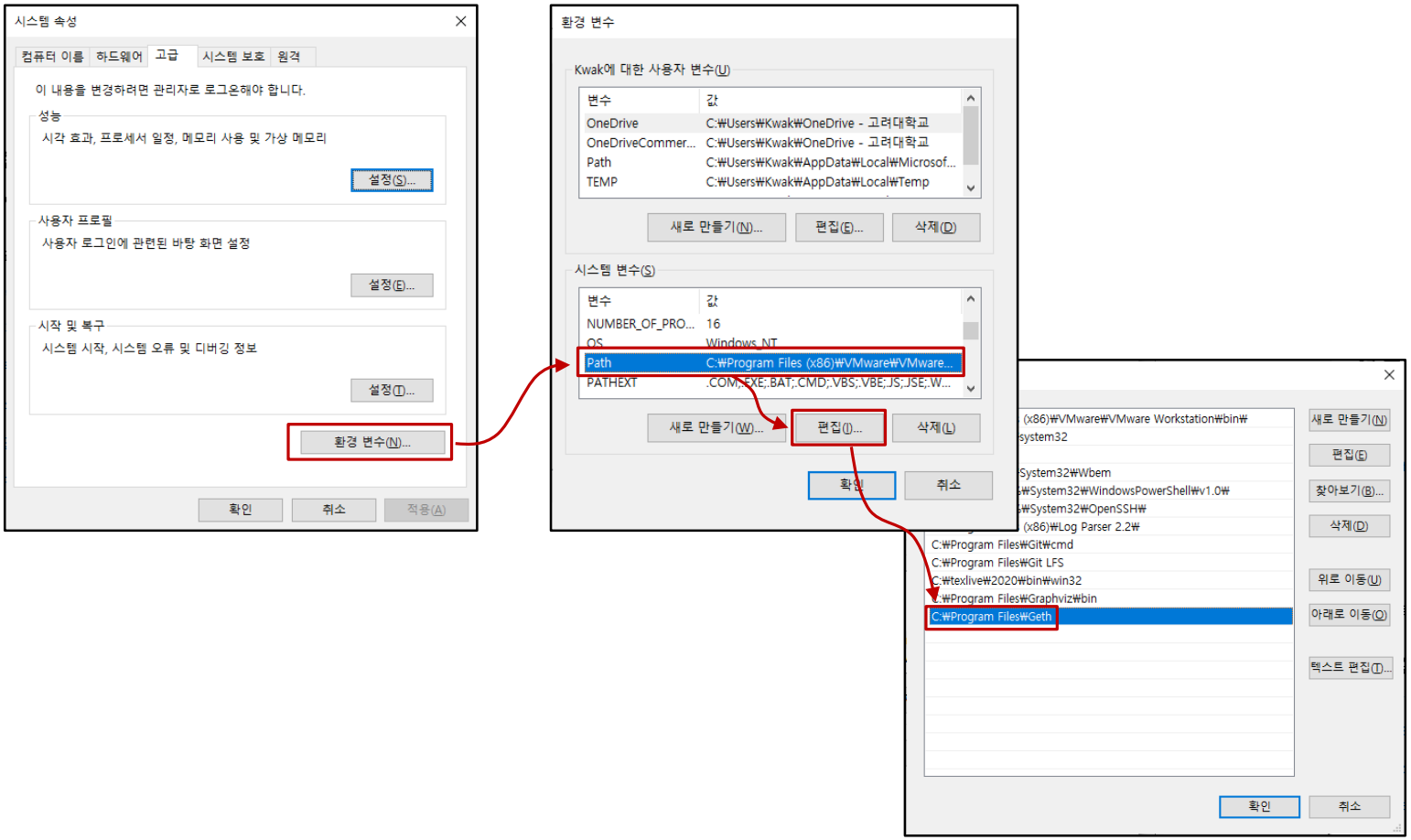


Geth: Go Ethereum

- Geth
  - 설치시 에러 발생할 경우, 경로 설정

Geth: Go Ethereum

- Geth
  - 설치시 에러 발생할 경우, 경로 설정



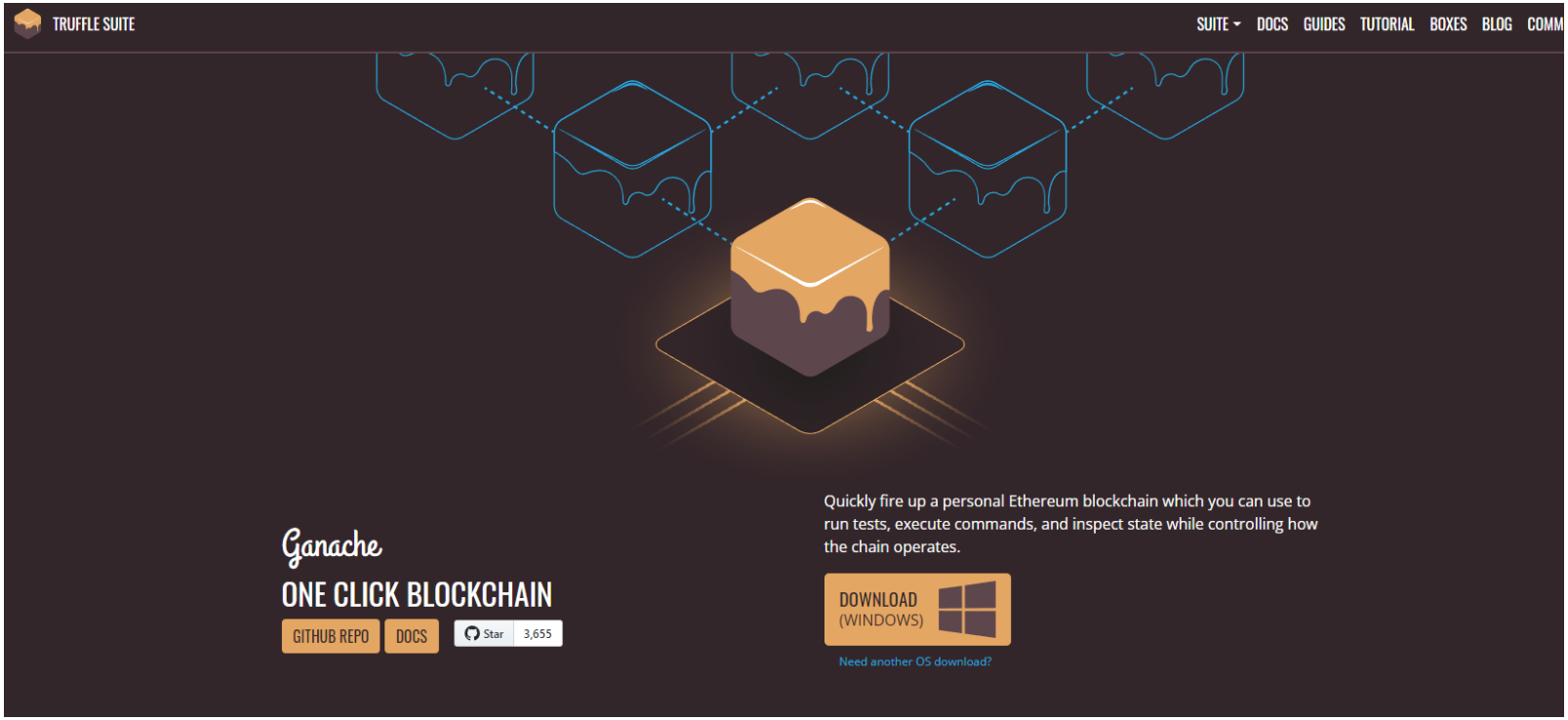
Geth: Go Ethereum

- Geth

```
C:₩Users₩Kwak>geth version
Geth
Version: 1.10.12-stable
Git Commit: 6c4dc6c38827296dec5a49a6ea25fd7f0eb4ac77
Git Commit Date: 20211108
Architecture: amd64
Go Version: go1.17.2
Operating System: windows
GOPATH=
GOROOT=go

C:₩Users₩Kwak>
```

Ganache

- https://truffleframework.com/ganache

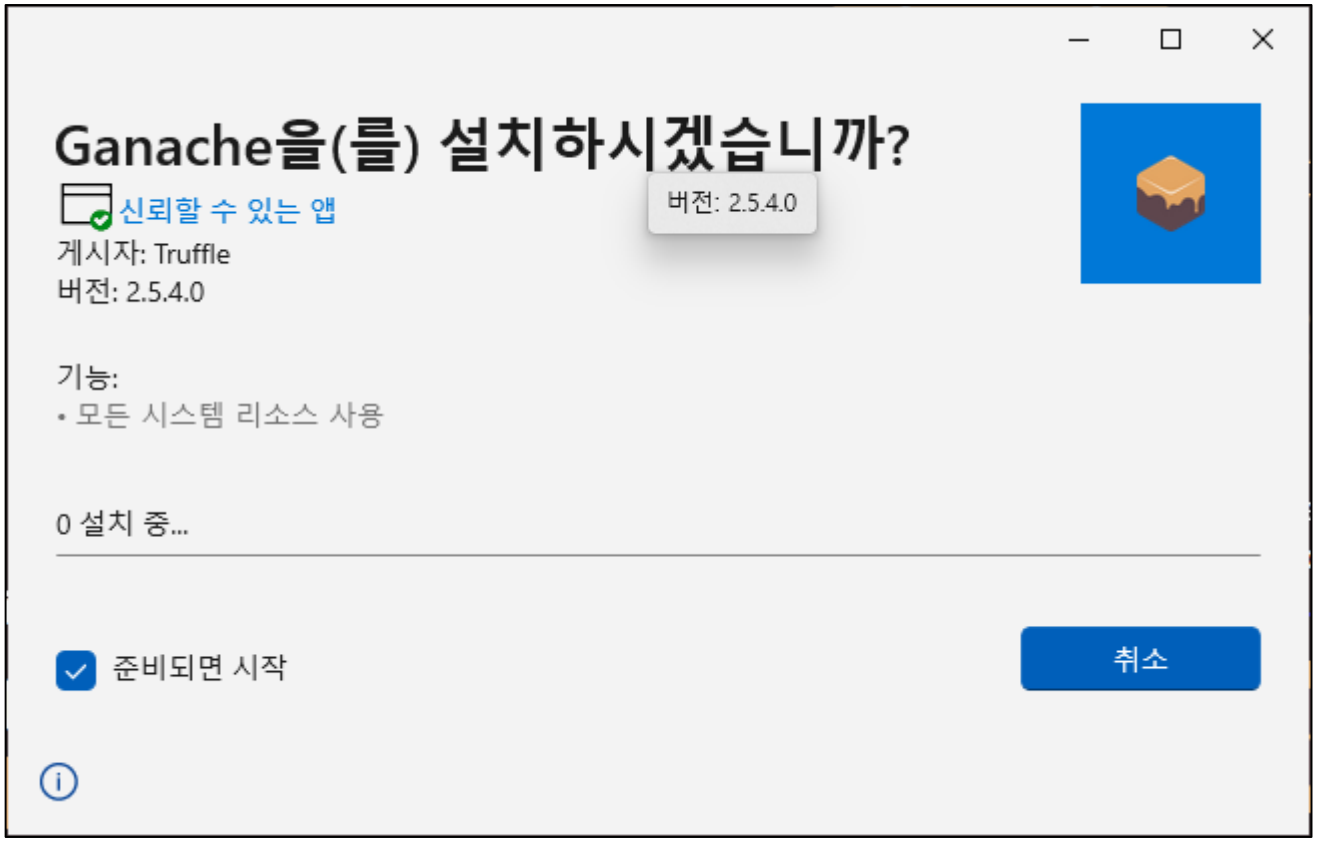


Ganache

- What is Ganache?
  - 이더리움 기반 블록체인 Dapp 개발에 사용하는 개인용 블록체인 프로그램 (UI 기반의 블록체인 툴)
    - 테스트 목적으로 PC에 설치해서 사용할 수 있는 일종의 간이 블록체인
    - Geth는 개발을 위해 트랜잭션을 실행하는데 대략 15초 이상 필요하며, 개발속도가 늦어질 수 있음
  - Ganache는 트랜잭션과 블록을 mining process 없이 생성할 수 있음 (메모리 내에서 수행)

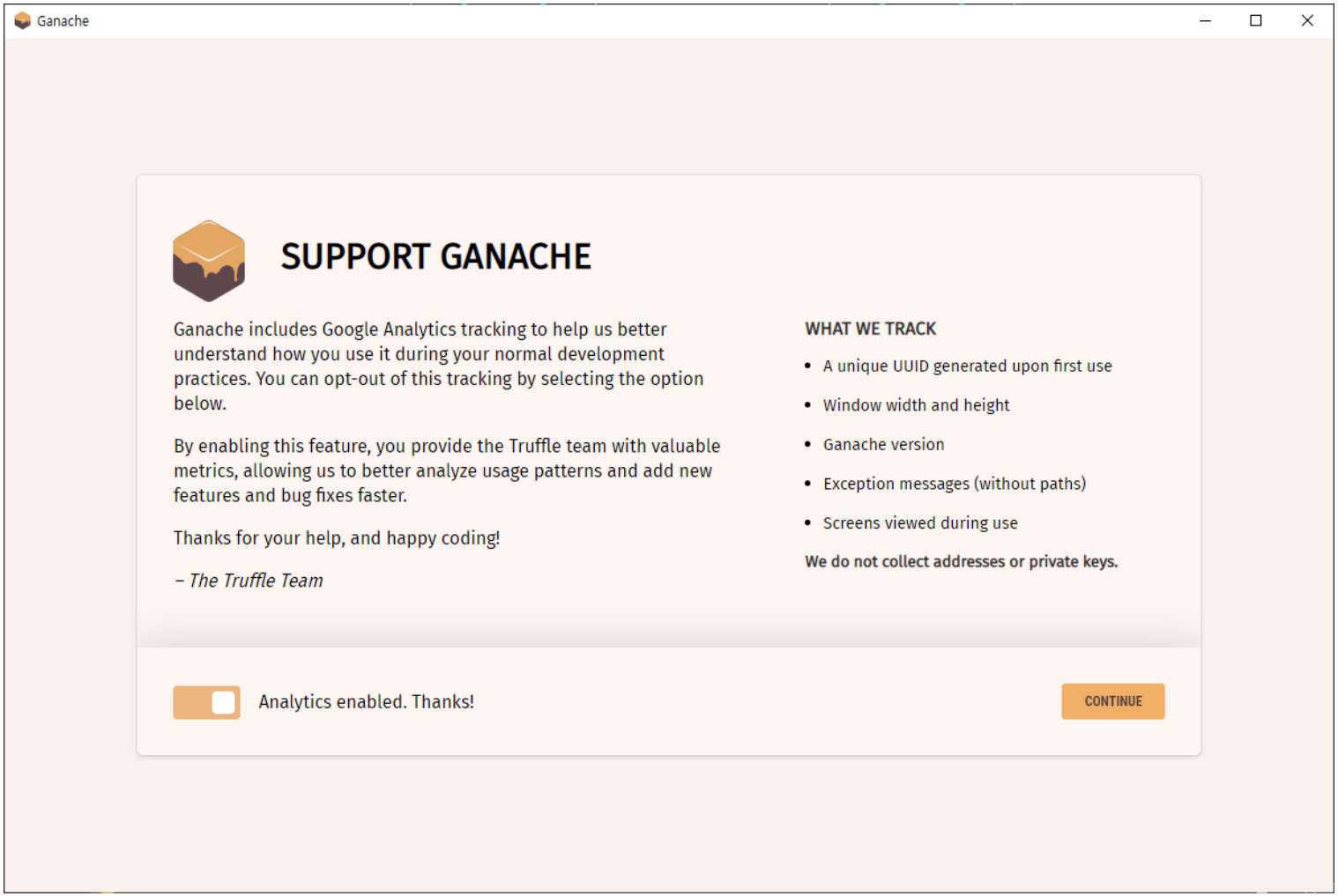
Ganache

- What is Ganache?



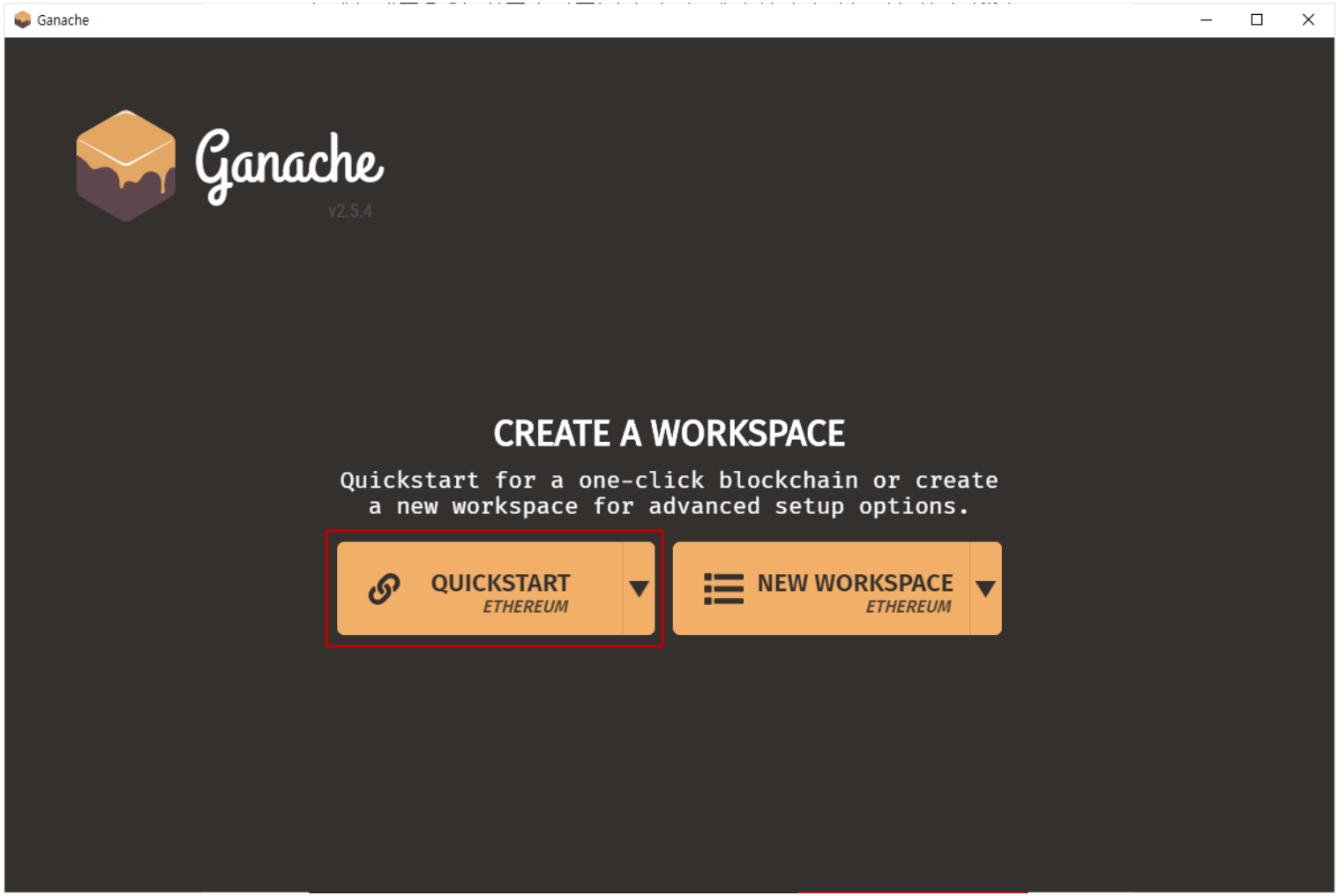
Ganache

- What is Ganache?

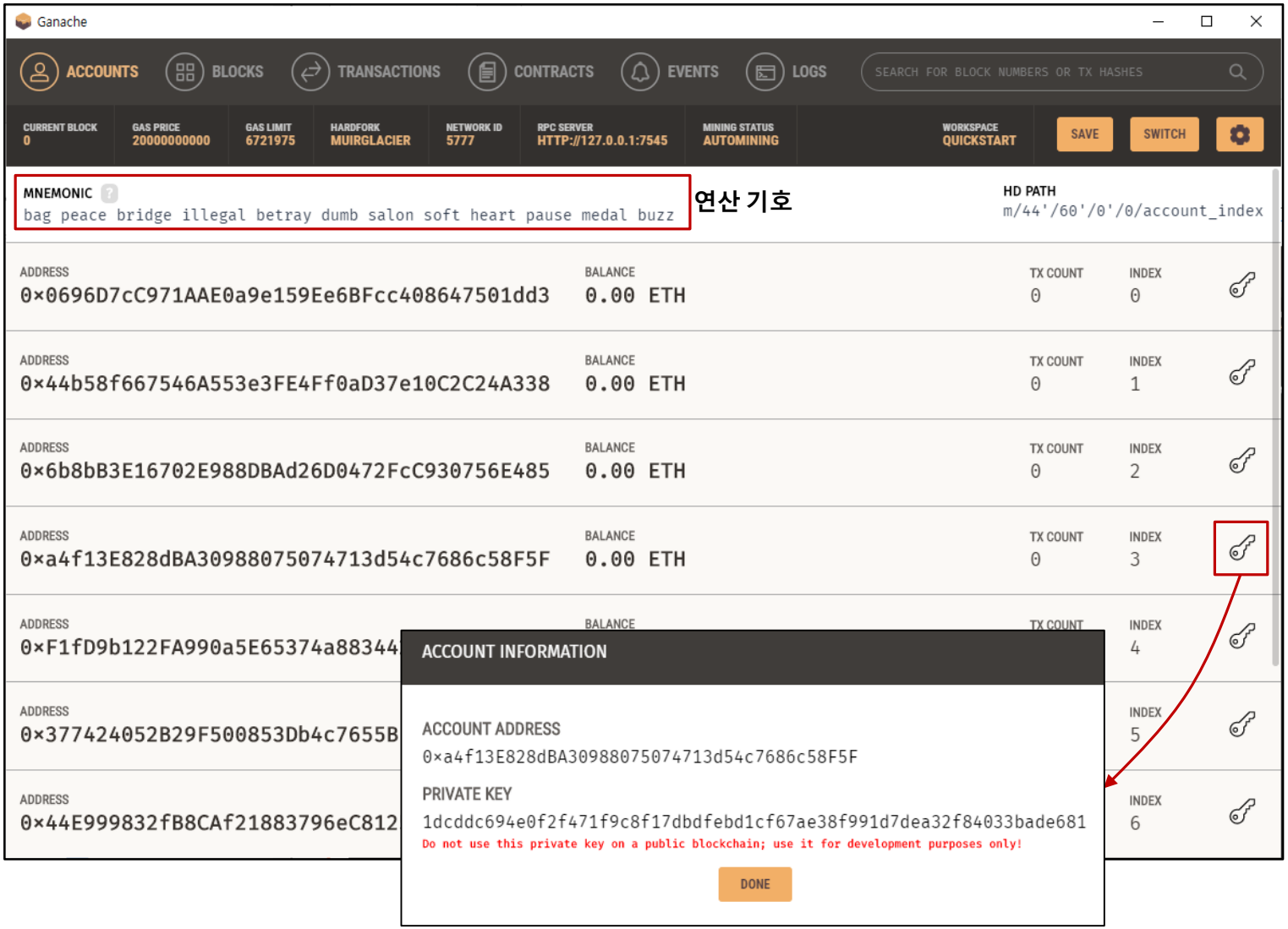


Ganache

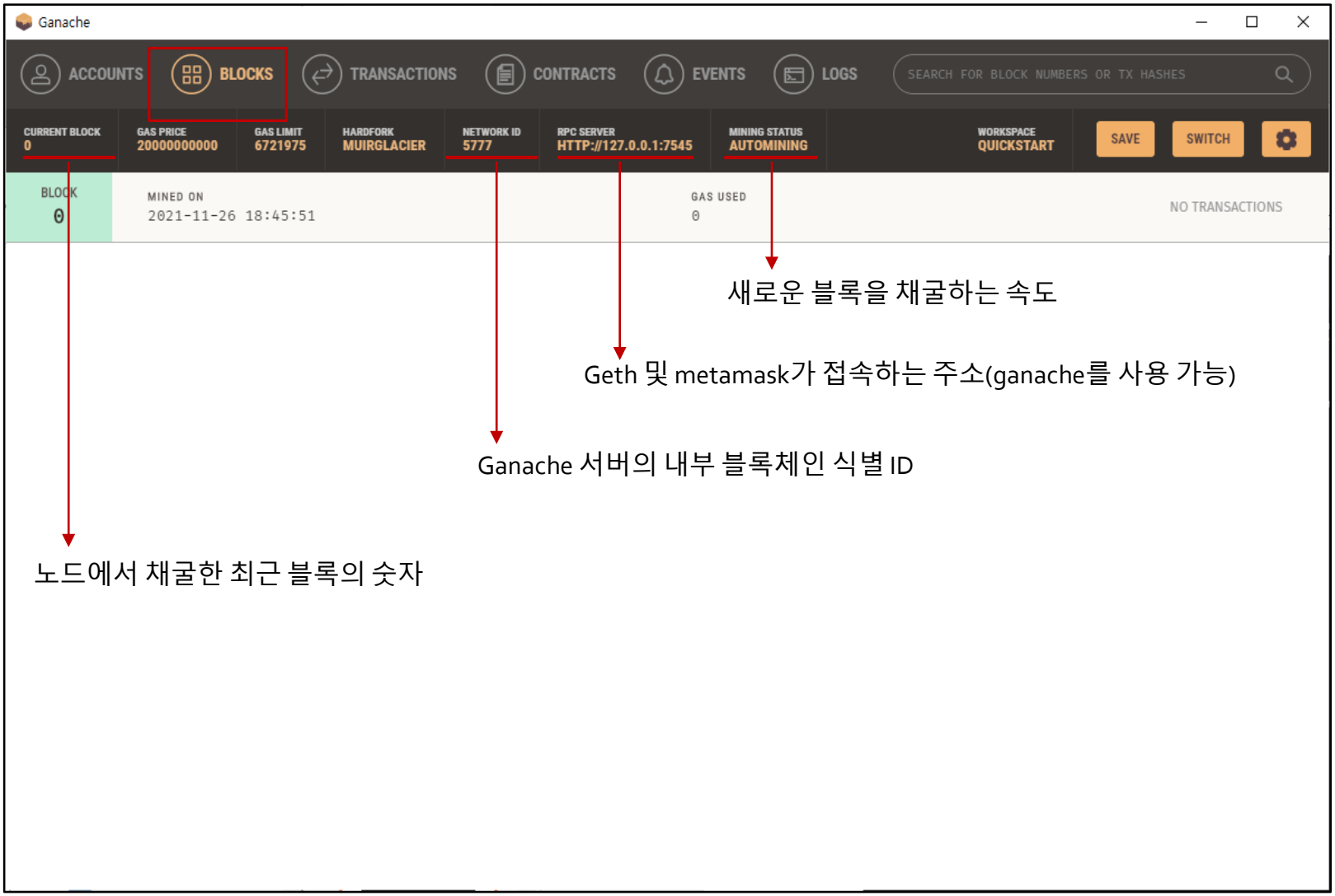
- What is Ganache?



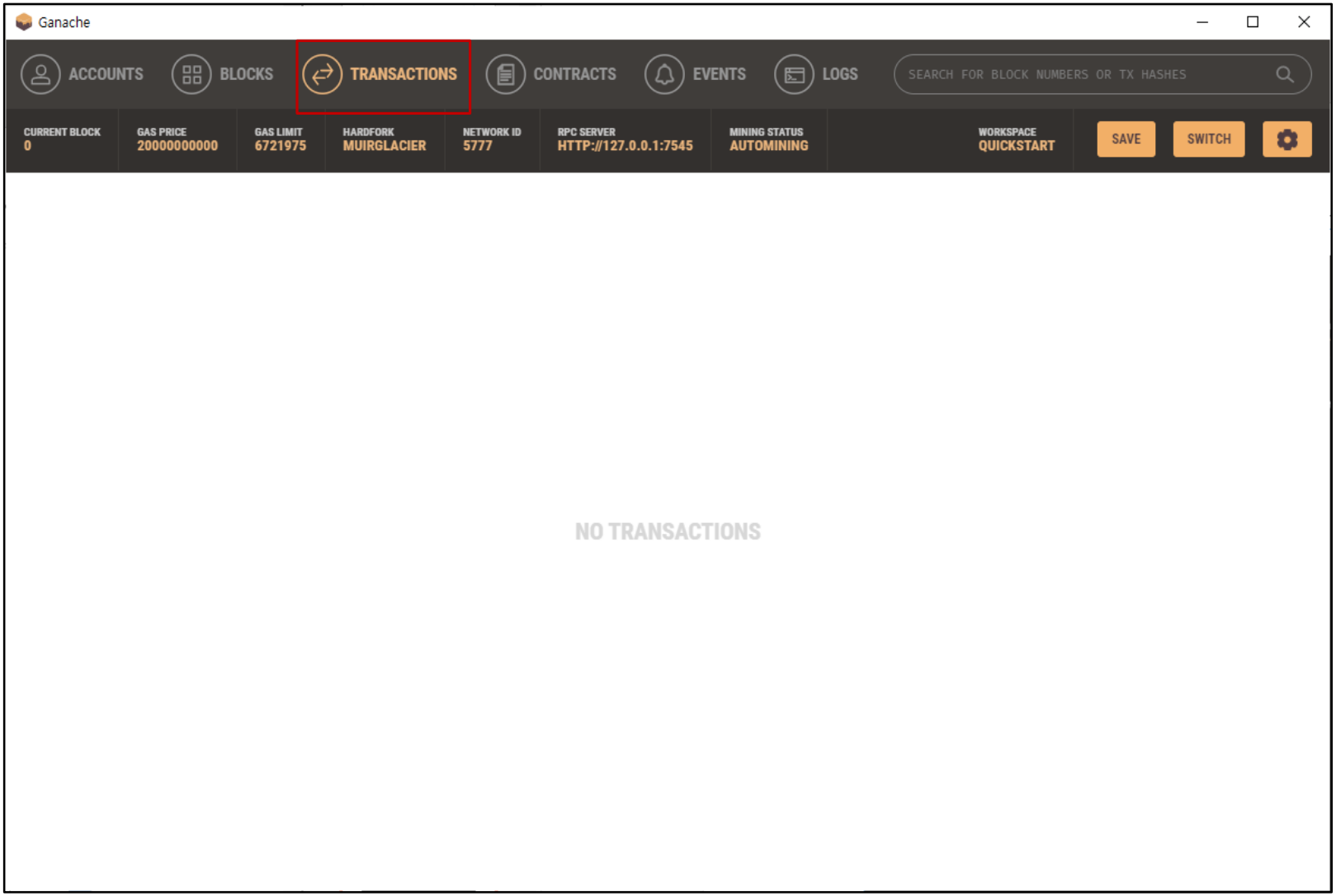
Ganache


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART
SAVE
SWITCH
MNEMONIC
bag peace bridge illegal betray dumb salon soft heart pause medal buzz
연산 기호
HD PATH
m/44'/60'/0'/0/account_index
ADDRESS 0x0696D7cC971AAE0a9e159Ee6BFcc408647501dd3 BALANCE 0.00 ETH TX COUNT 0 INDEX 0
ADDRESS 0x44b58f667546A553e3FE4Ff0aD37e10C2C24A338 BALANCE 0.00 ETH TX COUNT 0 INDEX 1
ADDRESS 0x6b8bB3E16702E988DBAd26D0472FcC930756E485 BALANCE 0.00 ETH TX COUNT 0 INDEX 2
ADDRESS 0xa4f13E828dBA30988075074713d54c7686c58F5F BALANCE 0.00 ETH TX COUNT 0 INDEX 3
ADDRESS 0xF1fD9b122FA990a5E65374a88344 BALANCE TX COUNT INDEX 4
ADDRESS 0x377424052B29F500853Db4c7655B INDEX 5
ADDRESS 0x44E999832fB8CAf21883796eC812 INDEX 6
ACCOUNT INFORMATION
ACCOUNT ADDRESS
0xa4f13E828dBA30988075074713d54c7686c58F5F
PRIVATE KEY
1dcddc694e0f2f471f9c8f17dbdfebd1cf67ae38f991d7dea32f84033bade681
Do not use this private key on a public blockchain; use it for development purposes only!
DONE

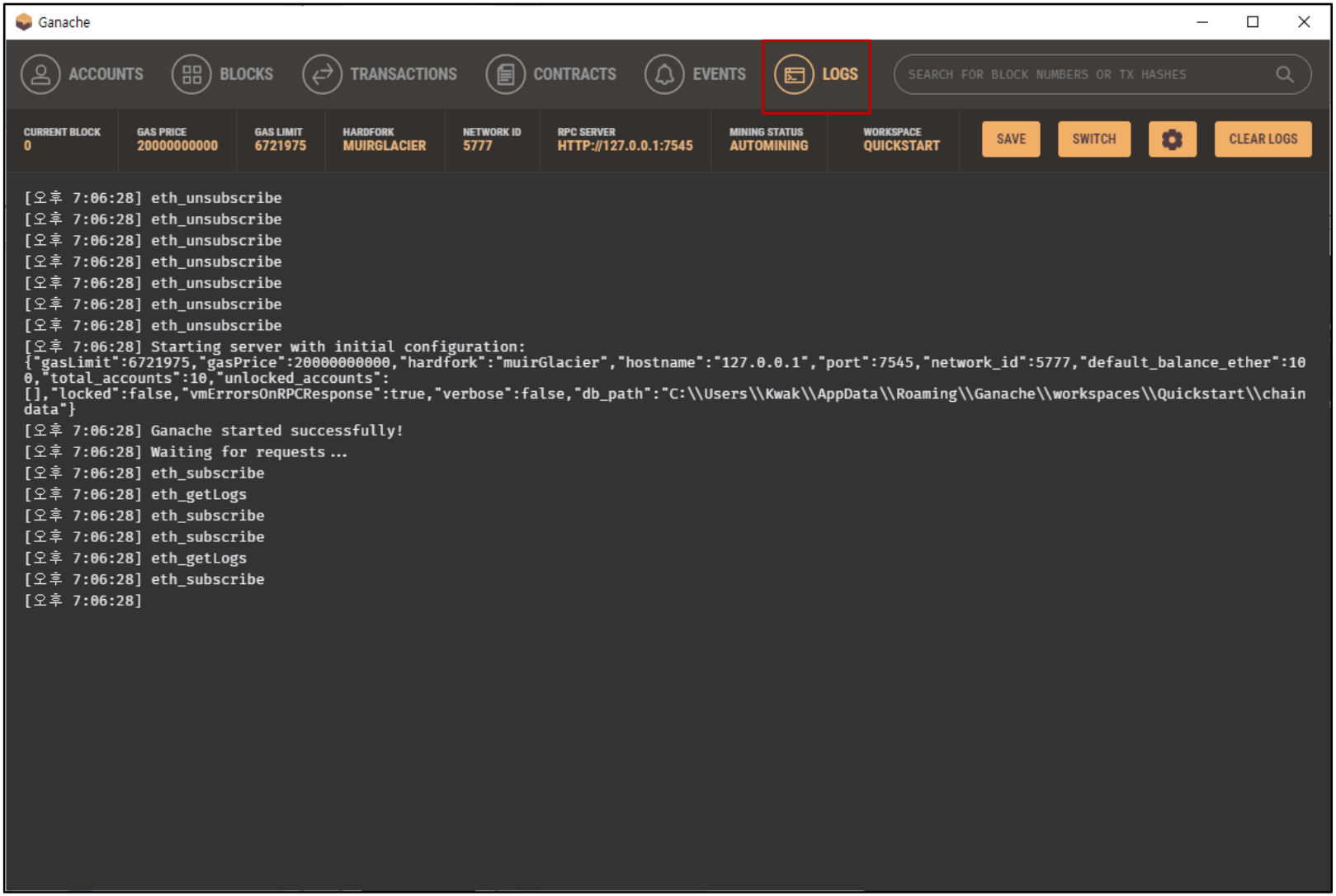
Ganache


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART
SAVE
SWITCH
BLOCK 0
MINED ON 2021-11-26 18:45:51
GAS USED 0
NO TRANSACTIONS
새로운 블록을 채굴하는 속도
Geth 및 metamask가 접속하는 주소(ganache를 사용 가능)
Ganache 서버의 내부 블록체인 식별 ID
노드에서 채굴한 최근 블록의 숫자

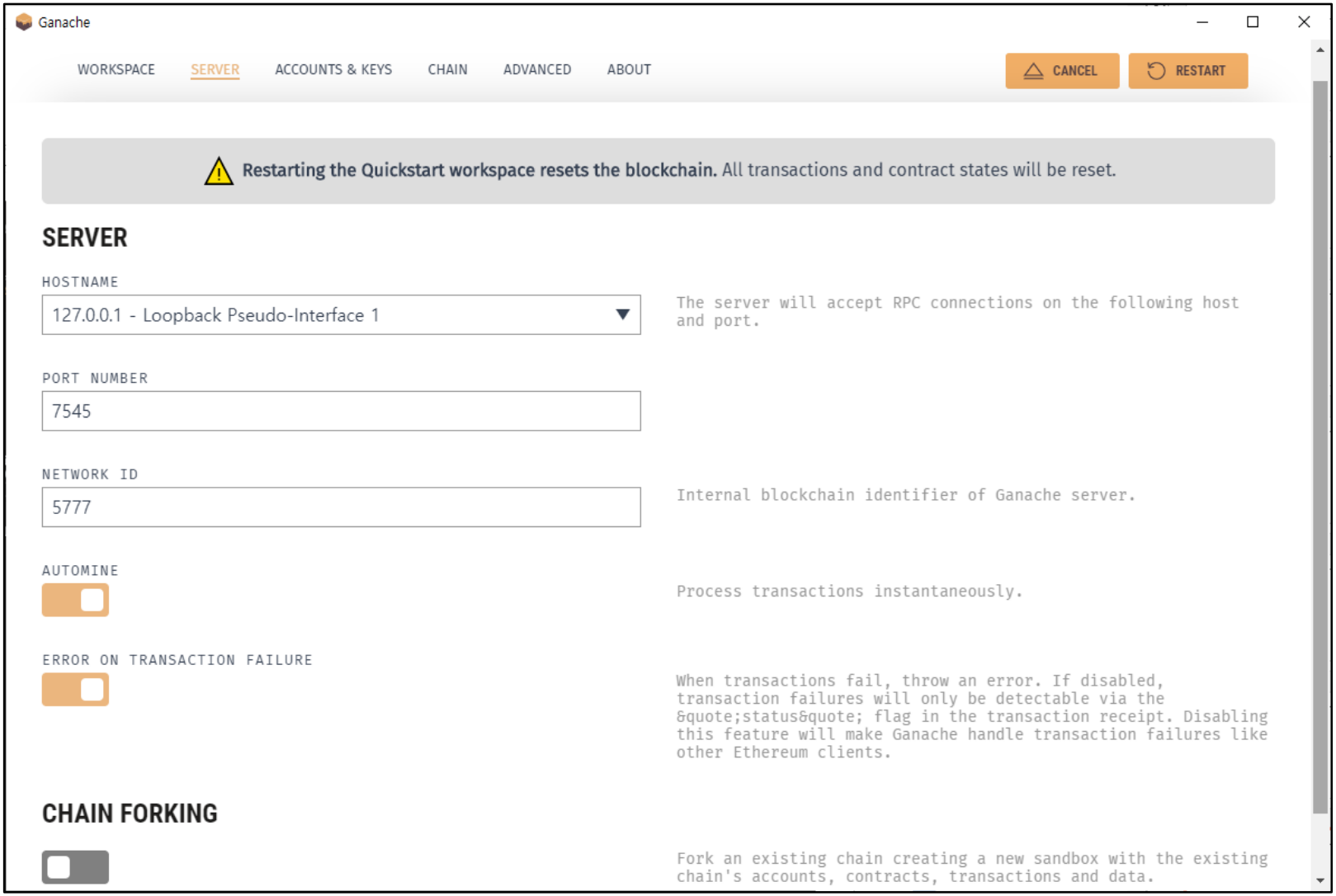
Ganache


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK
0
GAS PRICE
20000000000
GAS LIMIT
6721975
HARDFORK
MUIRGLACIER
NETWORK ID
5777
RPC SERVER
HTTP://127.0.0.1:7545
MINING STATUS
AUTOMINING
WORKSPACE
QUICKSTART
SAVE
SWITCH
NO TRANSACTIONS

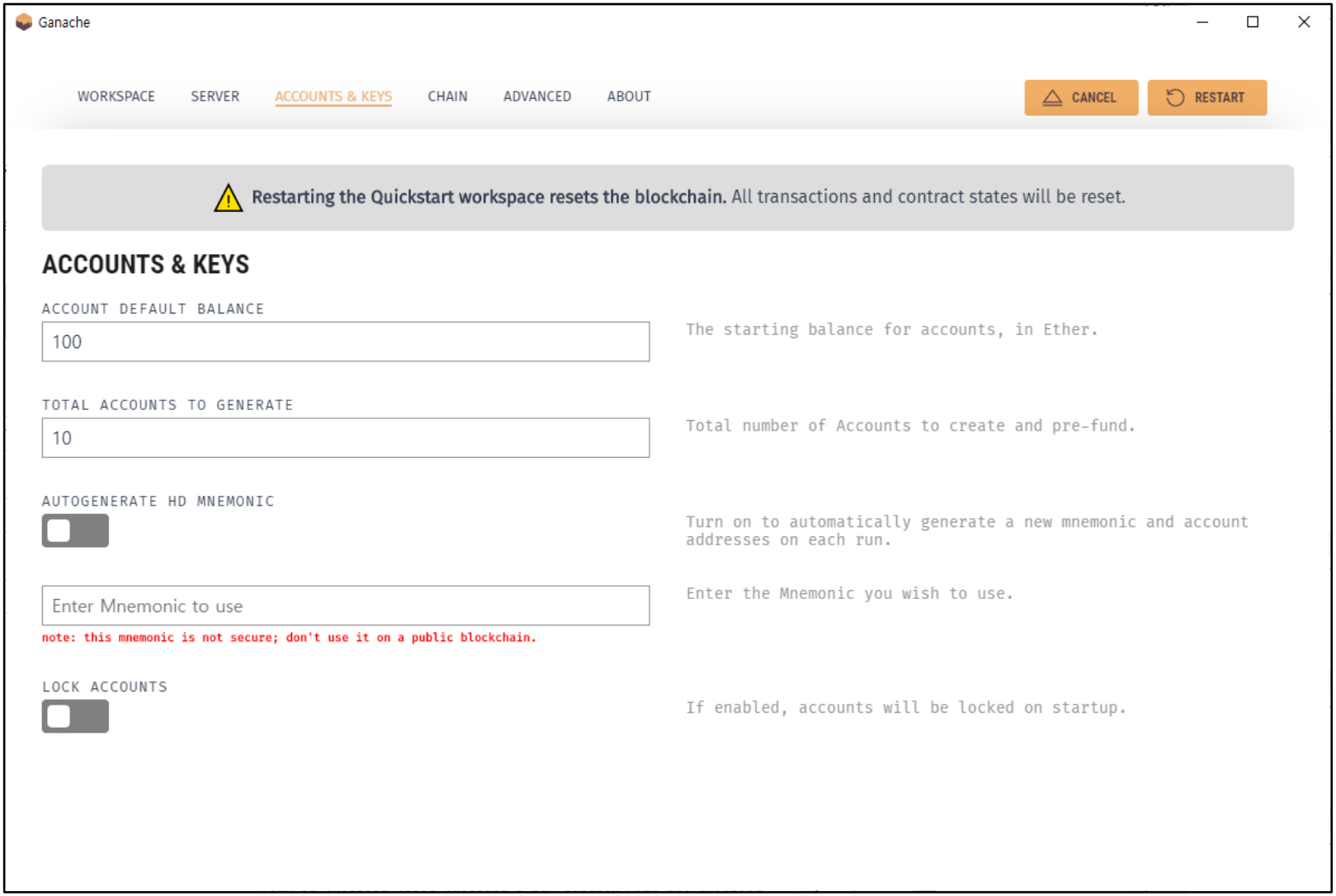
Ganache


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART
SAVE
SWITCH
CLEAR LOGS
[오후 7:06:28] eth_unsubscribe
[오후 7:06:28] eth_unsubscribe
[오후 7:06:28] eth_unsubscribe
[오후 7:06:28] eth_unsubscribe
[오후 7:06:28] eth_unsubscribe
[오후 7:06:28] eth_unsubscribe
[오후 7:06:28] eth_unsubscribe
[오후 7:06:28] Starting server with initial configuration:
{"gasLimit":6721975,"gasPrice":20000000000,"hardfork":"muirGlacier","hostname":"127.0.0.1","port":7545,"network_id":5777,"default_balance_ether":100,"total_accounts":10,"unlocked_accounts":[],"locked":false,"vmErrorsOnRPCResponse":true,"verbose":false,"db_path":"C:\\Users\\Kwak\\AppData\\Roaming\\Ganache\\workspaces\\Quickstart\\chain data"}
[오후 7:06:28] Ganache started successfully!
[오후 7:06:28] Waiting for requests ...
[오후 7:06:28] eth_subscribe
[오후 7:06:28] eth_getLogs
[오후 7:06:28] eth_subscribe
[오후 7:06:28] eth_subscribe
[오후 7:06:28] eth_getLogs
[오후 7:06:28] eth_subscribe
[오후 7:06:28]

Ganache

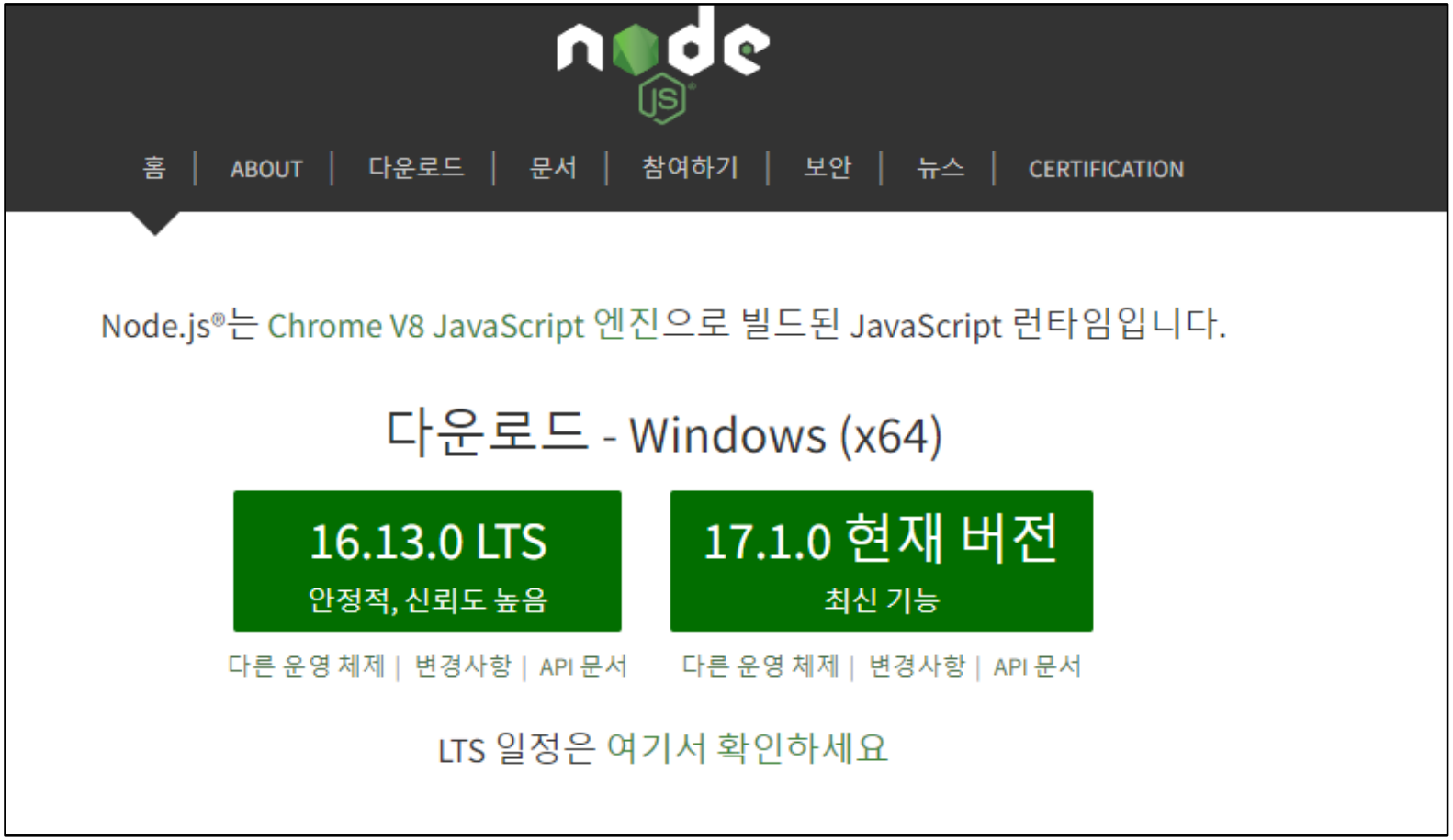

Ganache
WORKSPACE
SERVER
ACCOUNTS & KEYS
CHAIN
ADVANCED
ABOUT
CANCEL
RESTART
Restarting the Quickstart workspace resets the blockchain. All transactions and contract states will be reset.
SERVER
HOSTNAME
127.0.0.1 - Loopback Pseudo-Interface 1
The server will accept RPC connections on the following host and port.
PORT NUMBER
7545
NETWORK ID
5777
Internal blockchain identifier of Ganache server.
AUTOMINE
Process transactions instantaneously.
ERROR ON TRANSACTION FAILURE
When transactions fail, throw an error. If disabled, transaction failures will only be detectable via the &quote;status&quote; flag in the transaction receipt. Disabling this feature will make Ganache handle transaction failures like other Ethereum clients.
CHAIN FORKING
Fork an existing chain creating a new sandbox with the existing chain's accounts, contracts, transactions and data.

Ganache


Ganache
WORKSPACE
SERVER
ACCOUNTS & KEYS
CHAIN
ADVANCED
ABOUT
CANCEL
RESTART
Restarting the Quickstart workspace resets the blockchain. All transactions and contract states will be reset.
ACCOUNTS & KEYS
ACCOUNT DEFAULT BALANCE
100
The starting balance for accounts, in Ether.
TOTAL ACCOUNTS TO GENERATE
10
Total number of Accounts to create and pre-fund.
AUTOGENERATE HD MNEMONIC
Turn on to automatically generate a new mnemonic and account addresses on each run.
Enter Mnemonic to use
Enter the Mnemonic you wish to use.
note: this mnemonic is not secure; don't use it on a public blockchain.
LOCK ACCOUNTS
If enabled, accounts will be locked on startup.

Node.js

- https://nodejs.org/ko/



  - Node.js는 서버사이드 js 플랫폼 분산 어플리케이션에 필수

Node.js

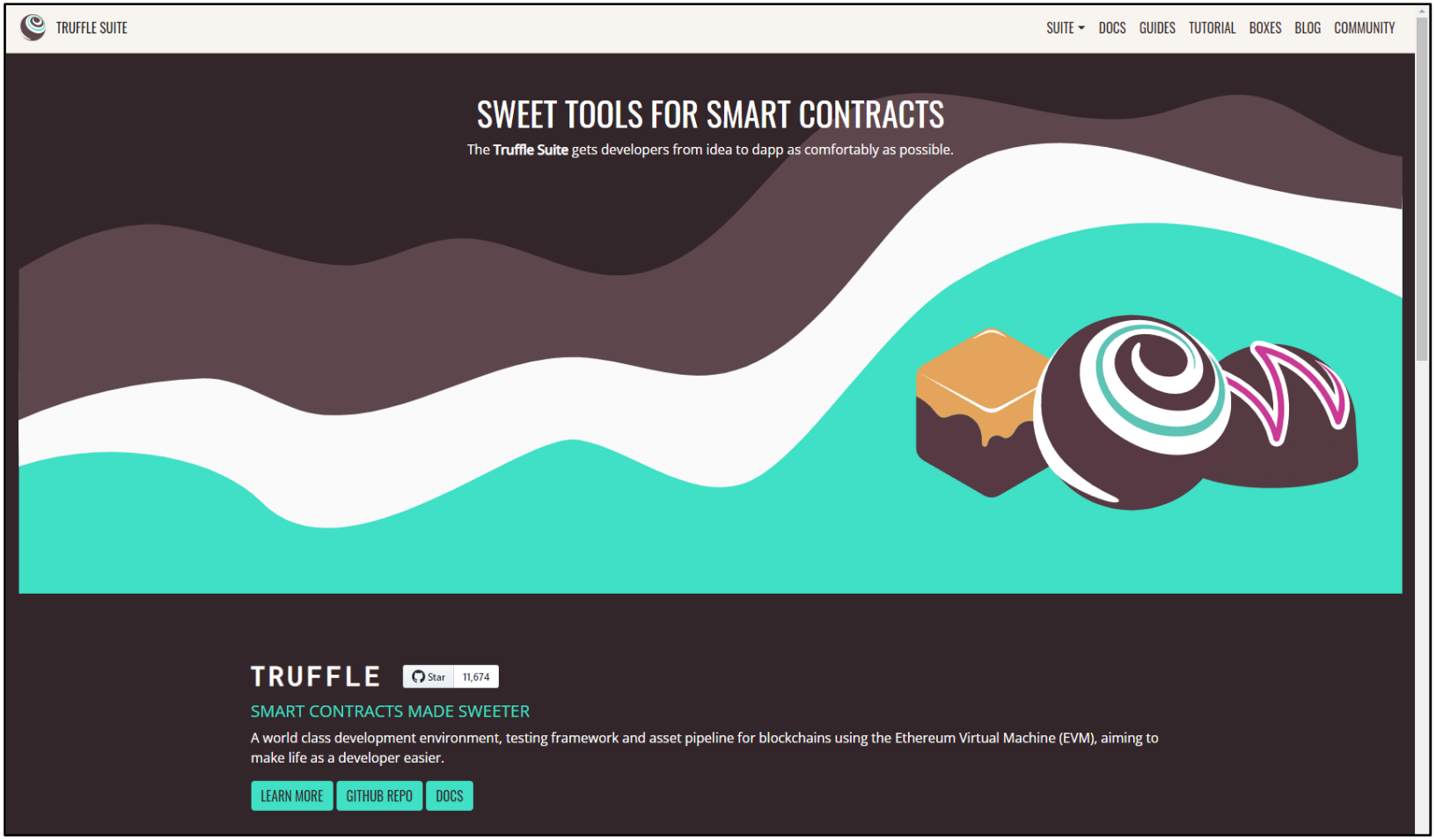
- Node.js 버전 체크

```
명령 프롬프트
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:₩Users₩Kwak>node -v
v16.13.0

C:₩Users₩Kwak>npm -v
8.1.0

C:₩Users₩Kwak>
```

  - npm (패키지 관리자)은 개발하면서 툴이나 라이브러리 다운로드를 위해 사용

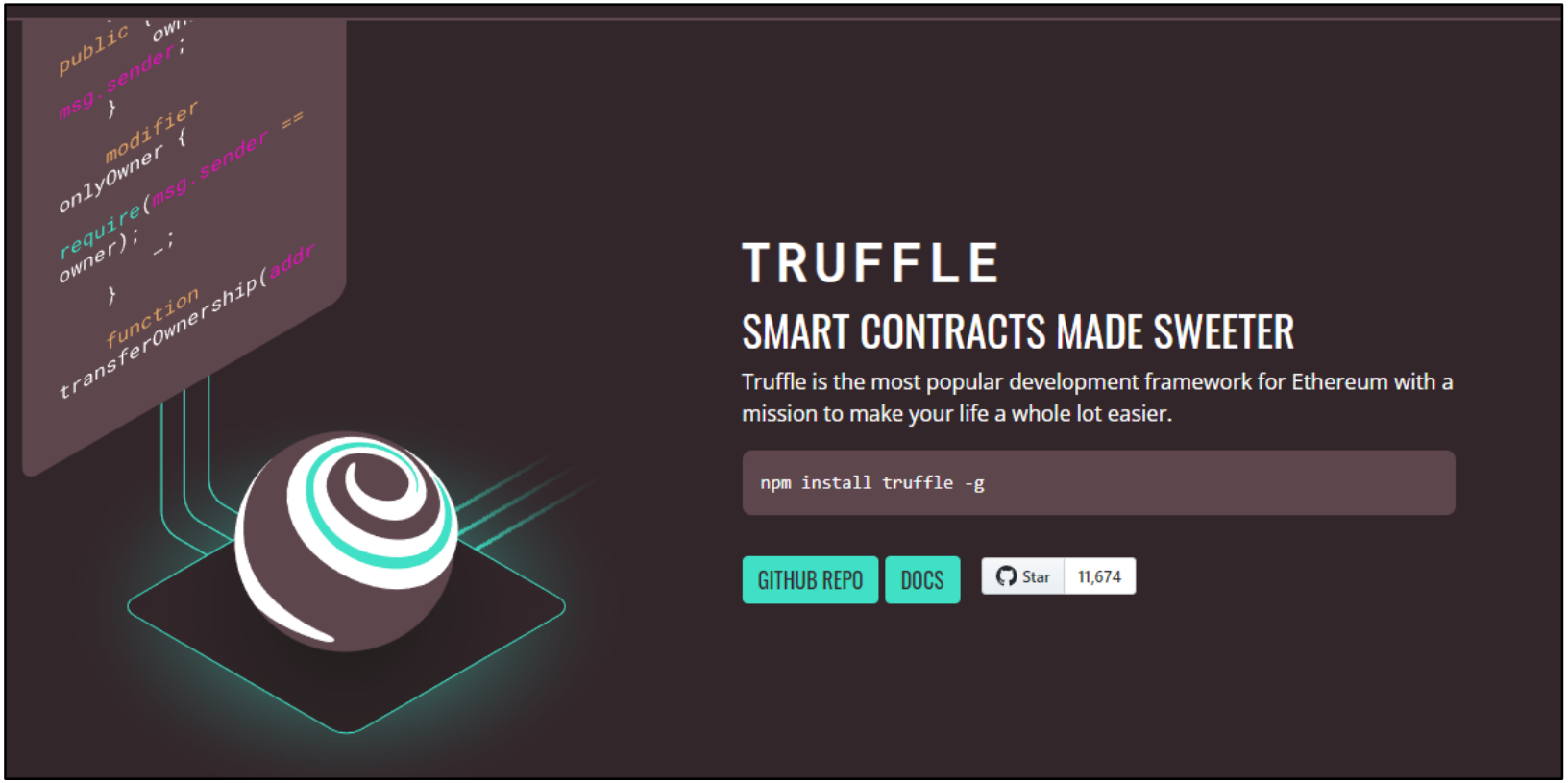
Truffle

- https://truffleframework.com/



- 스마트 컨트랙트를 컴파일, 테스트, 배포를 위해 사용

Truffle


TRUFFLE
SMART CONTRACTS MADE SWEETER
Truffle is the most popular development framework for Ethereum with a mission to make your life a whole lot easier.
npm install truffle -g
GITHUB REPO
DOCS
Star
11,674

Truffle

npm install truffle

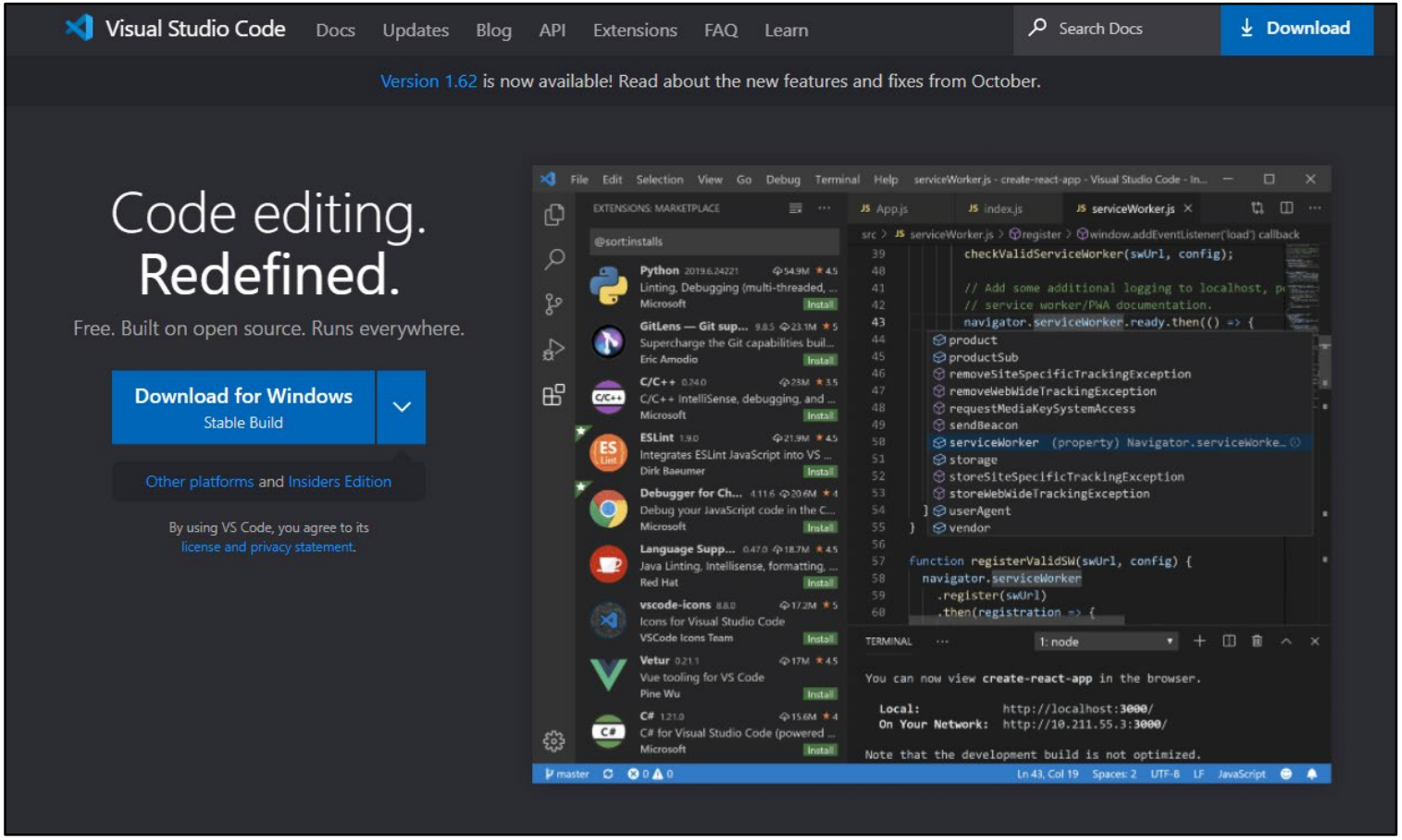
```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:₩Users₩Kwak>node -v
v16.13.0

C:₩Users₩Kwak>npm -v
8.1.0

C:₩Users₩Kwak>npm install -g truffle
[                    ] - idealTree:npm: timing idealTree:#root Completed in 2723ms
```

Truffle

```
C:₩WINDOWS₩system32₩cmd.exe
86 packages are looking for funding
  run `npm fund` for details

89 vulnerabilities (7 low, 60 moderate, 15 high, 7 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible, run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 8.1.0 -> 8.1.4
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.1.4
npm notice Run npm install -g npm@8.1.4 to update!
npm notice

C:₩Users₩Kwak>truffle version
Truffle v5.4.22 (core: 5.4.22)
Solidity v0.5.16 (solc-js)
Node v16.13.0
Web3.js v1.5.3

C:₩Users₩Kwak>_
```

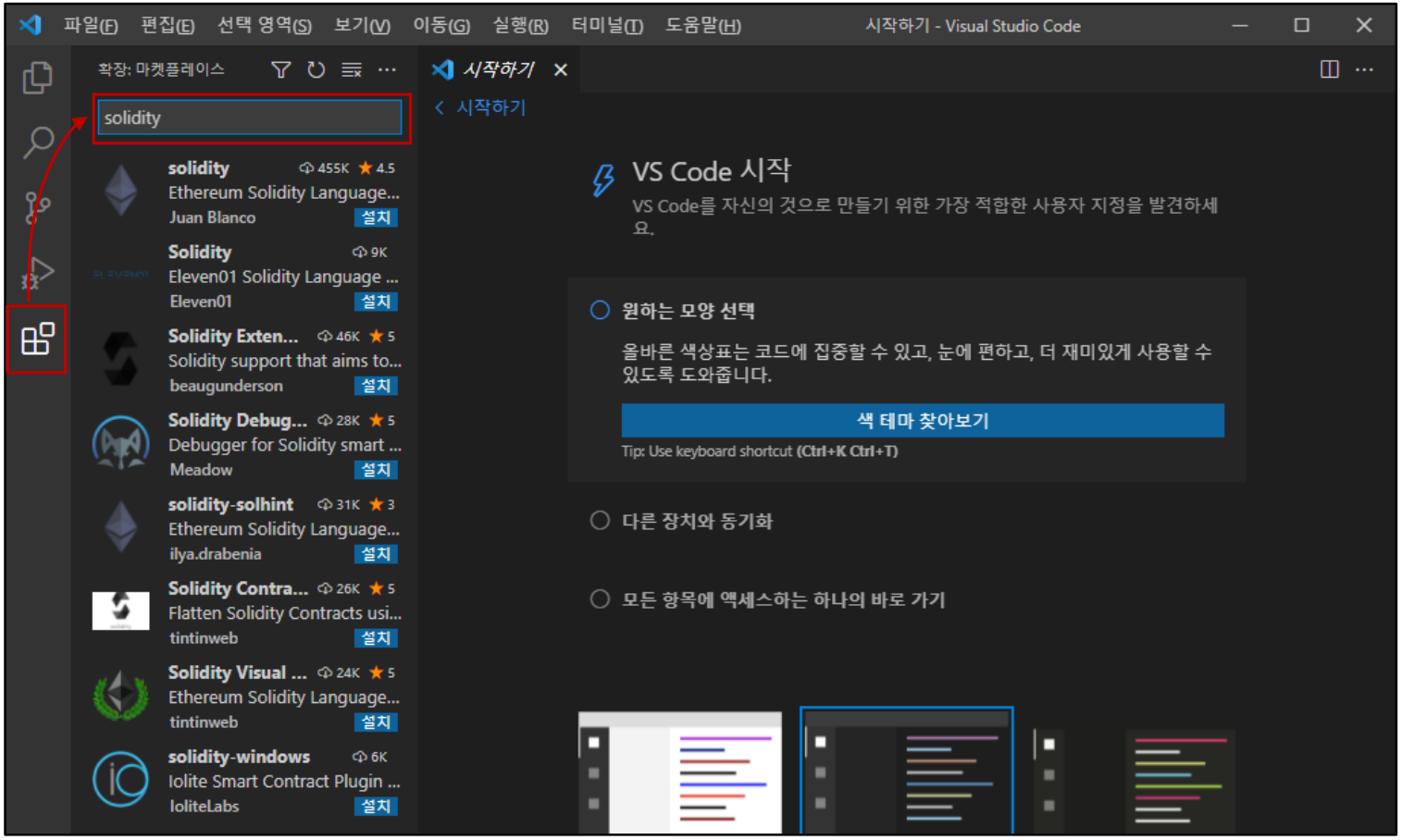
Visual Studio Code

- https://code.visualstudio.com/



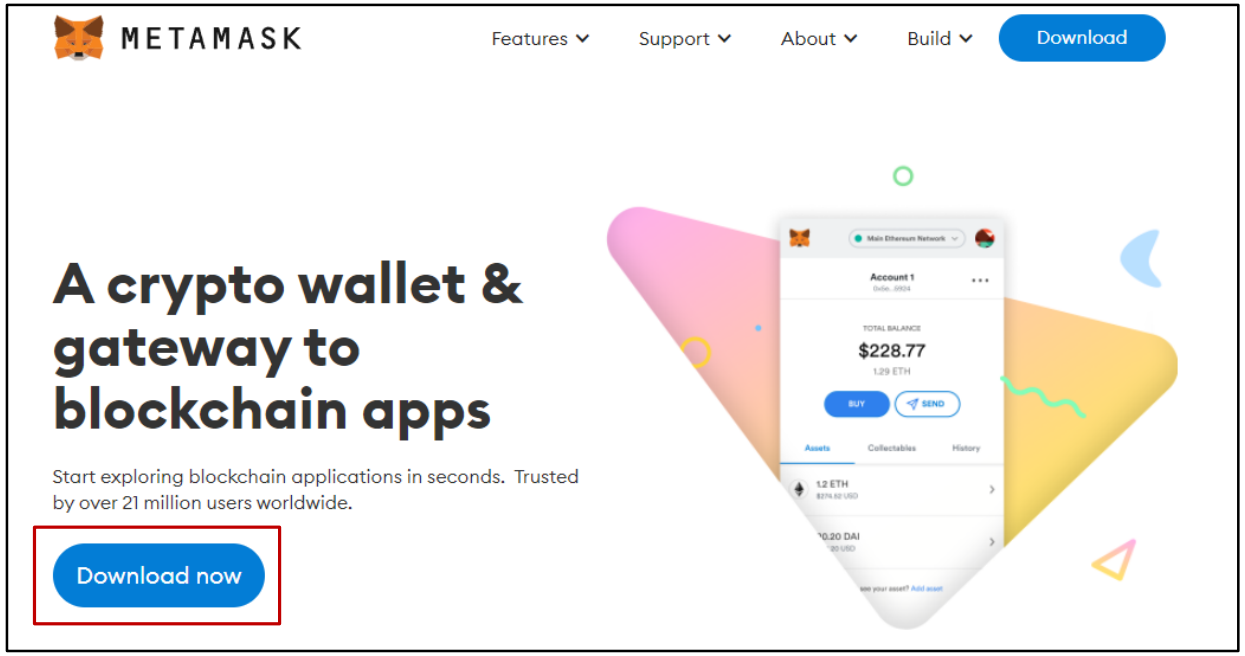
Visual Studio Code

- https://code.visualstudio.com/

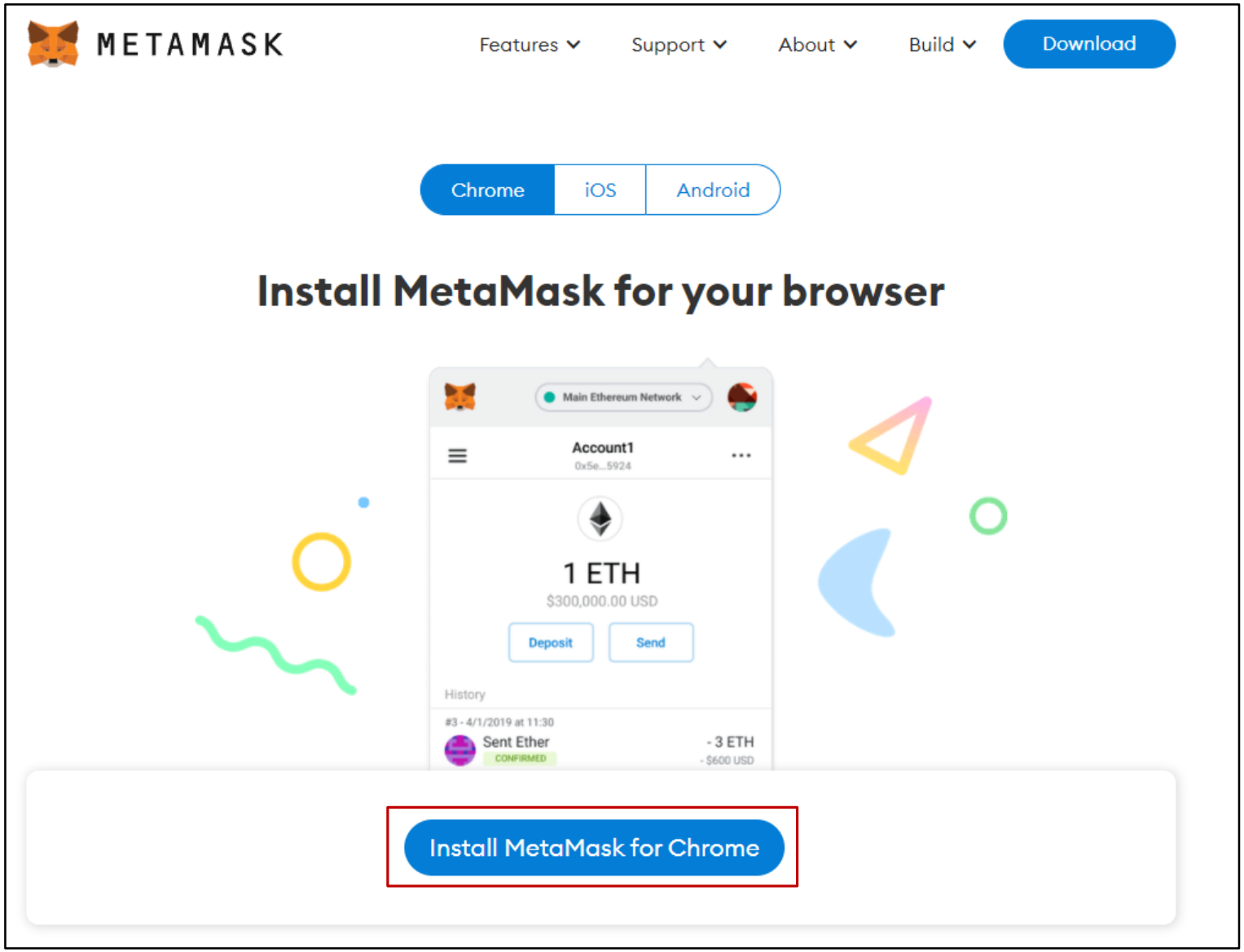


METAMASK

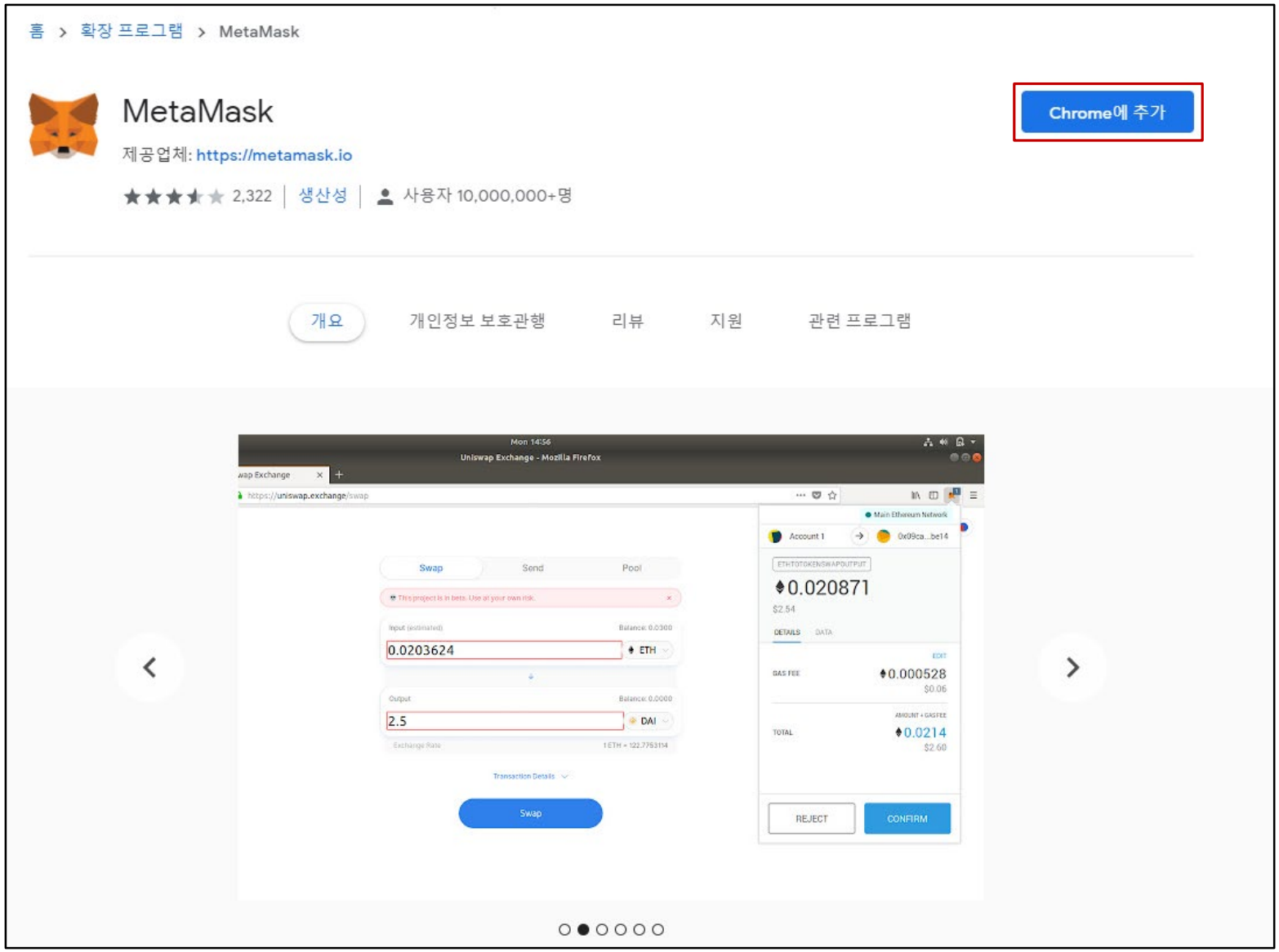
- Chrome-based wallet
- Transactions can be processed by METAMASK
  - https://metamask.io/



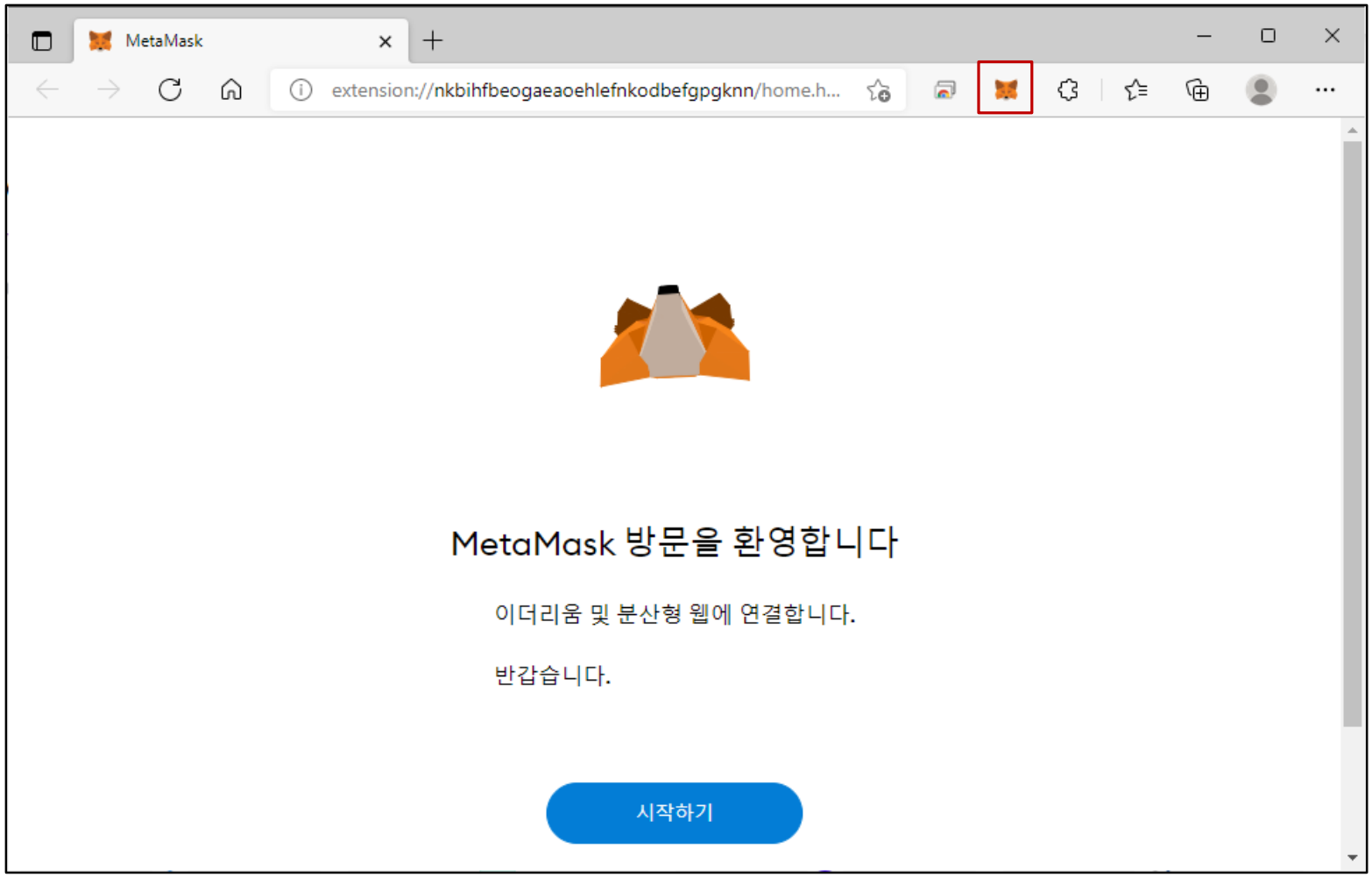
METAMASK


METAMASK
Features
Support
About
Build
Download
Chrome
iOS
Android
Install MetaMask for your browser
Main Ethereum Network
Account1
0x5e...5924
1 ETH
$300,000.00 USD
Deposit
Send
History
#3 - 4/1/2019 at 11:30
Sent Ether
CONFIRMED
- 3 ETH
- $600 USD
Install MetaMask for Chrome

METAMASK


홈 > 확장 프로그램 > MetaMask
MetaMask
제공업체: https://metamask.io
2,322 | 생산성 | 사용자 10,000,000+명
Chrome에 추가
개요
개인정보 보호관행
리뷰
지원
관련 프로그램
Uniswap Exchange - Mozilla Firefox
Swap
Send
Pool
0.0203624
ETH
2.5
DAI
Swap
Account 1
0.020871
$2.54
DETAILS
DATA
GAS FEE
0.000528
$0.06
TOTAL
0.0214
$2.60
REJECT
CONFIRM

METAMASK


MetaMask
extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.h...
MetaMask 방문을 환영합니다
이더리움 및 분산형 웹에 연결합니다.
반갑습니다.
시작하기

METAMASK

METAMASK

MetaMask 개선에 참여

MetaMask는 사용자가 확장 프로그램과 상호작용하는 방식을 자세히 이해하기 위해 사용 데이터를 수집하려 합니다. 이 데이터는 당사의 제품과 이더리움 에코시스템의 사용 편의성 및 사용자 경험을 지속적으로 개선하는 데 사용됩니다.

MetaMask에서는..

- ✔ 언제든 설정을 통해 옵트아웃할 수 있습니다.
- ✔ 익명화된 클릭 및 페이지뷰 이벤트 보내기

- ✘ 키, 주소, 거래, 잔액, 해시 또는 개인 정보를 절대 수집하지 않습니다.
- ✘ 전체 IP 주소를 절대 수집하지 않습니다.
- ✘ 수익을 위해 데이터를 절대 판매하지 않습니다. 결코 그렇습니다.

괜찮습니다 | 동의함

이 데이터는 집계되며 일반 데이터 보호 규칙(EU) 2016/679의 목적에 따라 익명으로 관리됩니다. 당사의 개인정보보호 관행에 관한 자세한 내용은 개인정보보호정책을(를) 참조하세요.

METAMASK

METAMASK

지갑 보호하기

시작하기 전에 이 짧은 동영상을 보고 복구 구문과 지갑을 안전하게 보호하는 방법에 대해 알아보세요.

다음

'복구 구문'이란 무엇인가요?

복구 구문은 지갑과 자금의 '마스터 키'입니다.

복구 구문은 어떻게 저장하나요?

- 암호 관리자에 저장
- 은행 금고에 보관.
- 대여 금고에 보관.
- 적어서 여러 비밀 장소에 보관하세요.

복구 구문을 공유해야 하나요?

절대로, MetaMask와도 시드 구문을 공유하면 안 됩니다!

복구 구문을 요청하는 사람은 사기를 치려는 것입니다.

METAMASK

METAMASK

< 뒤로

Secret Recovery Phrase

비밀 백업 구문을 이용하면 계정을 쉽게 백업하고 복구할 수 있습니다.

경고: 백업 구문은 절대로 공개하지 마세요. 이 구문이 있는 사람은 귀하의 Ether를 영원히 소유할 수 있습니다.

비밀 단어를 표시하려면 여기를 클릭하세요.

나중에 알림

다음

팁:

이 구문을 1Password 같은 암호 관리자에 저장하세요.

메모지에 이 구문을 적어 안전한 곳에 보관하세요. 보안을 더욱 강화하고 싶다면 여러 메모지에 적은 다음 2~3곳에 보관하세요.

이 구문을 기억합니다.

이 비밀 백업 구문을 다운로드하고 암호화된 외장 하드 드라이브나 저장 매체에 안전하게 보관하세요.

METAMASK

METAMASK

축하합니다.

테스트를 통과하셨습니다. 비밀 복구 구문을 안전하게 보관할 책임은 본인에게 있습니다.

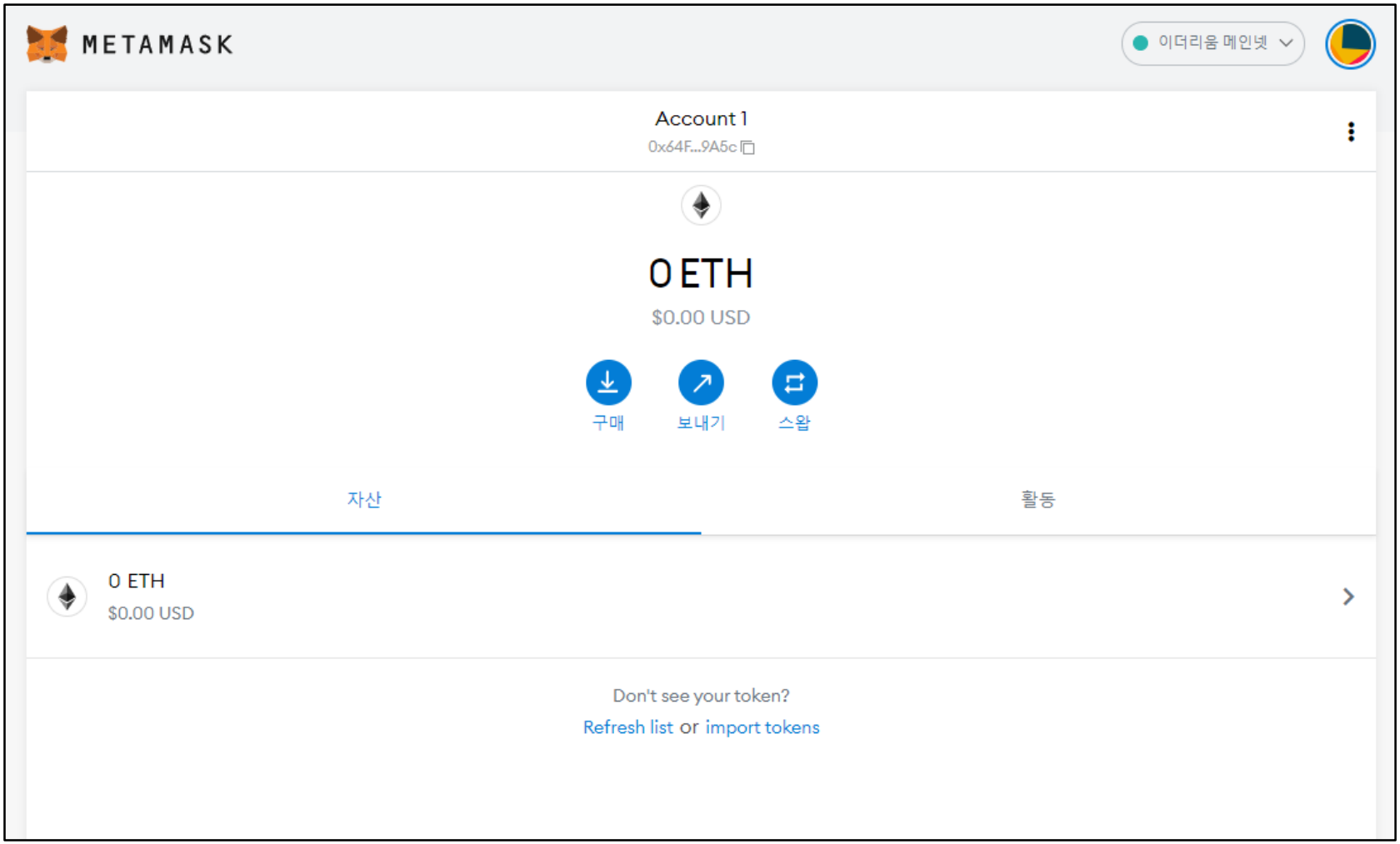
안전한 보관 관련 팁

- 백업을 여러 장소에 보관하세요.
- 구문을 누구와도 공유하지 마세요.
- 피싱에 유의하세요. MetaMask에서는 절대로 비밀 복구 구문을 갑자기 물어보지 않습니다.
- 비밀 복구 구문을 다시 백업해야 한다면 설정 -> 보안에서 해당 구문을 찾을 수 있습니다.
- 질문이 있거나 의심스러운 행위를 목격했다면 지원을 요청하세요(여기).

*MetaMask에서는 계정 시드 구문을 복구할 수 없습니다. 자세한 내용을 알아보십시오.

모두 완료

METAMASK


METAMASK
이더리움 메인넷
Account 1
0x64F...9A5c
0 ETH
$0.00 USD
구매
보내기
스왑
자산
활동
0 ETH
$0.00 USD
Don't see your token?
Refresh list or import tokens

Ganache

Ganache

WORKSPACE SERVER ACCOUNTS & KEYS CHAIN ADVANCED ABOUT

CANCEL RESTART

Restarting the Quickstart workspace resets the blockchain. All transactions and contract states will be reset.

SERVER

HOSTNAME

127.0.0.1 - Loopback Pseudo-Interface 1

The server will accept RPC connections on the following host and port.

PORT NUMBER

8545

NETWORK ID

5777

Internal blockchain identifier of Ganache server.

AUTOMINE

Process transactions instantaneously.

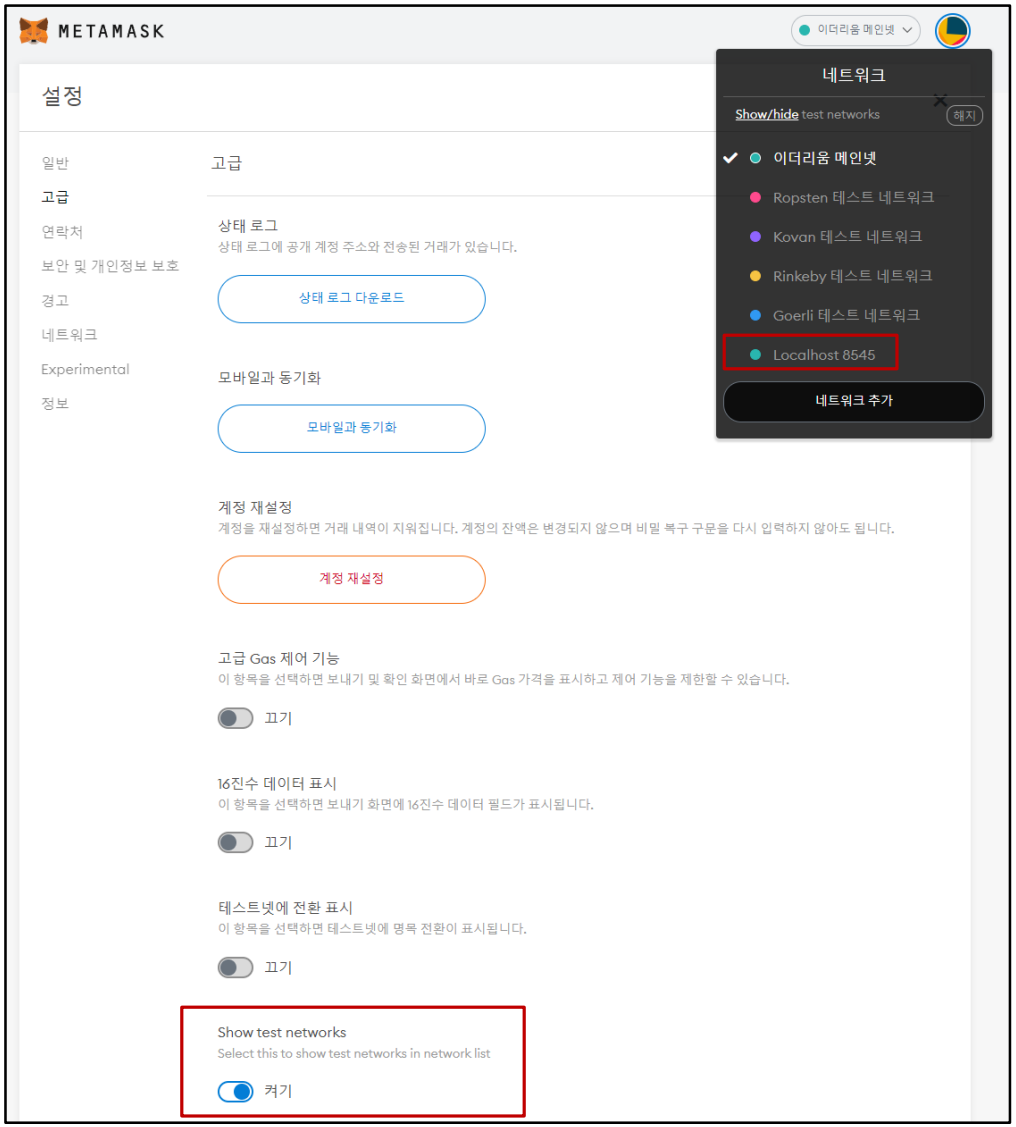
ERROR ON TRANSACTION FAILURE

When transactions fail, throw an error. If disabled, transaction failures will only be detectable via the &quote;status&quote; flag in the transaction receipt. Disabling this feature will make Ganache handle transaction failures like other Ethereum clients.

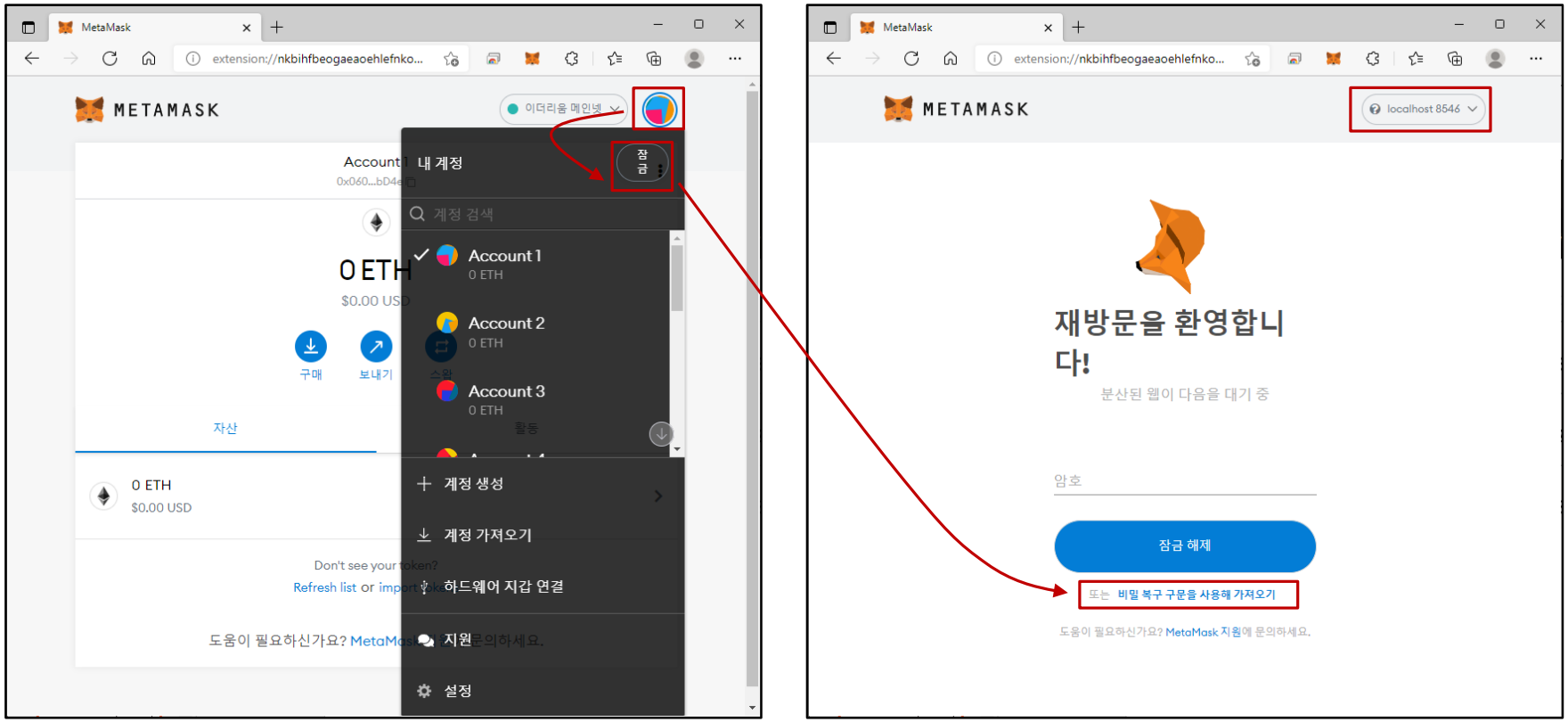
CHAIN FORKING

Fork an existing chain creating a new sandbox with the existing

Ganache


METAMASK
이더리움 메인넷
설정
일반
고급
연락처
보안 및 개인정보 보호
경고
네트워크
Experimental
정보
고급
상태 로그
상태 로그에 공개 계정 주소와 전송된 거래가 있습니다.
상태 로그 다운로드
모바일과 동기화
모바일과 동기화
계정 재설정
계정을 재설정하면 거래 내역이 지워집니다. 계정의 잔액은 변경되지 않으며 비밀 복구 구문을 다시 입력하지 않아도 됩니다.
계정 재설정
고급 Gas 제어 기능
이 항목을 선택하면 보내기 및 확인 화면에서 바로 Gas 가격을 표시하고 제어 기능을 제한할 수 있습니다.
끄기
16진수 데이터 표시
이 항목을 선택하면 보내기 화면에 16진수 데이터 필드가 표시됩니다.
끄기
테스트넷에 전환 표시
이 항목을 선택하면 테스트넷에 명목 전환이 표시됩니다.
끄기
Show test networks
Select this to show test networks in network list
켜기
네트워크
Show/hide test networks
해지
이더리움 메인넷
Ropsten 테스트 네트워크
Kovan 테스트 네트워크
Rinkeby 테스트 네트워크
Goerli 테스트 네트워크
Localhost 8545
네트워크 추가

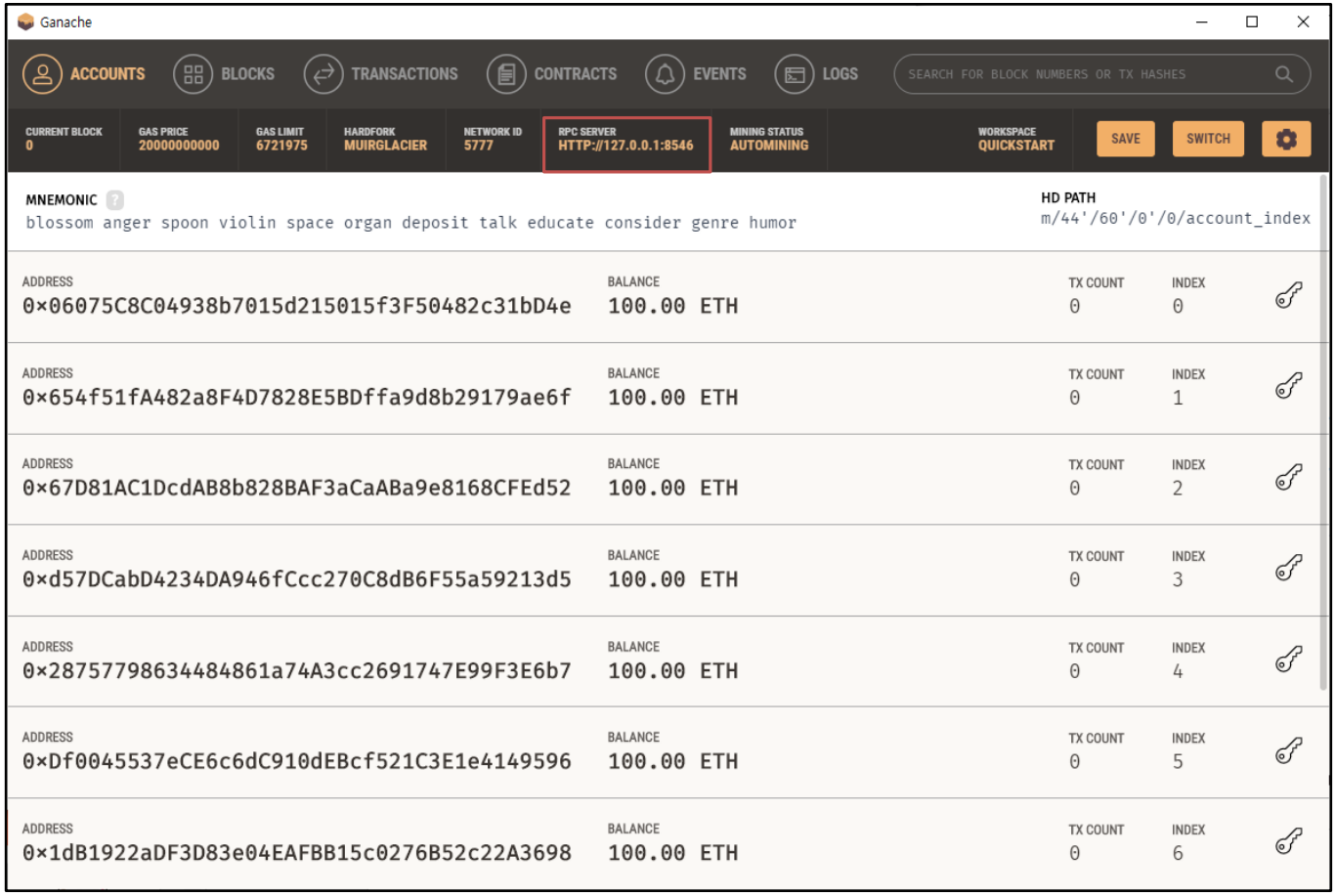
Ganache



- Localhost 8545로 안될 경우, localhost 8546으로 다시 설정

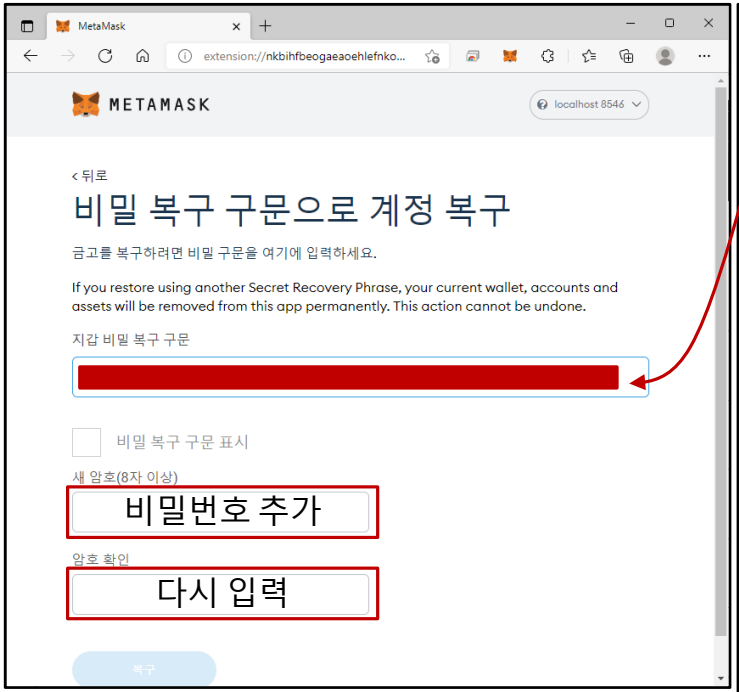
Ganache

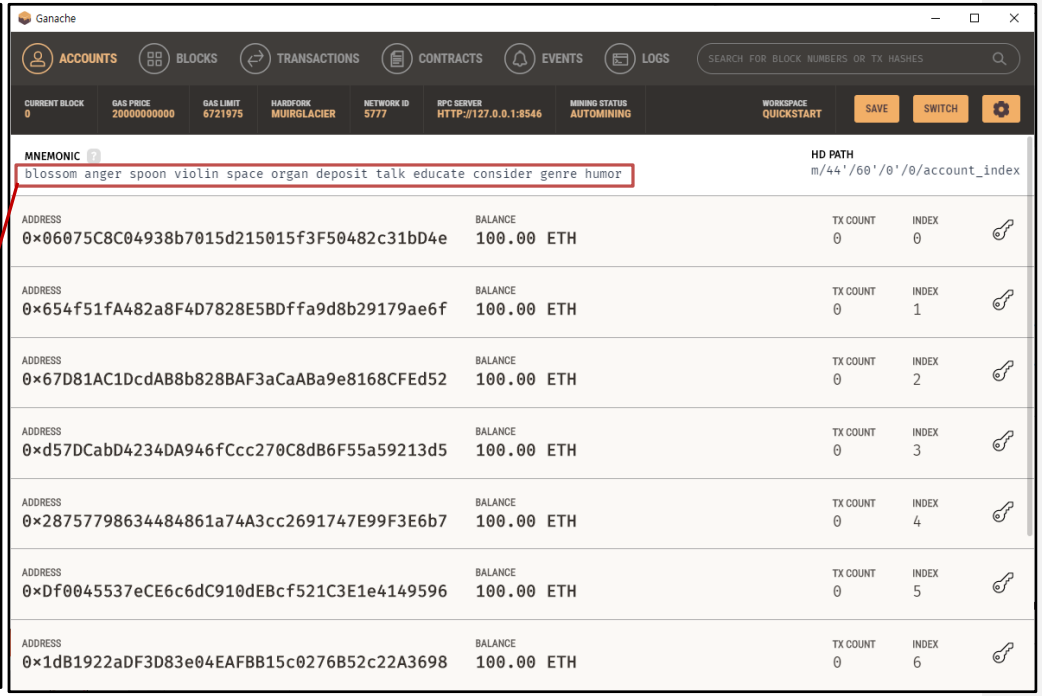
- Port번호 8546으로 다시 설정할 경우



Ganache

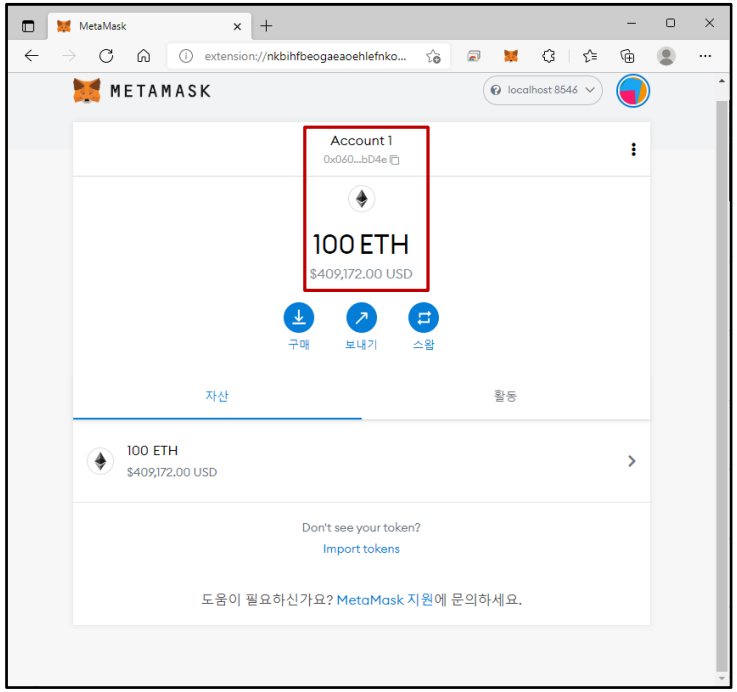
Localhost 8546

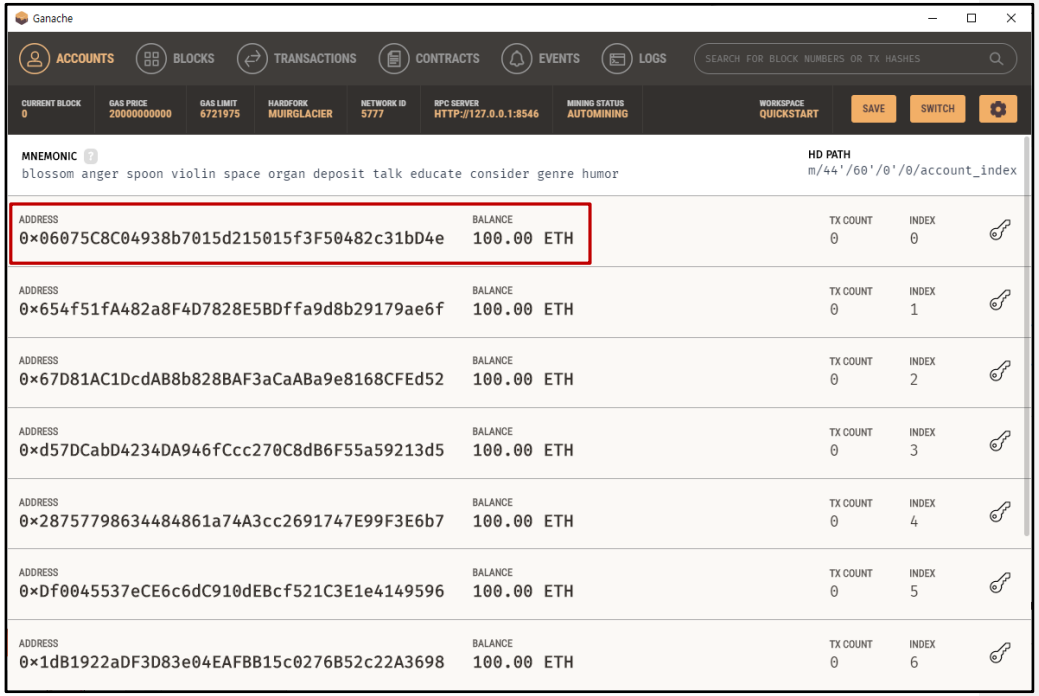




Ganache

- Localhost 8546





Geth: Go Ethereum (Private Blockchain)

- Geth를 이용한 Private 노드 구축

```
C:₩WINDOWS₩system32₩cmd.exe - puppeth

C:₩>mkdir blockchain

C:₩>cd blockchain

C:₩blockchain>pupeth
'pupeth'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

C:₩blockchain>puppeth
+-----------------------------------------------------------+
| Welcome to puppeth, your Ethereum private network manager |
|                                                           |
| This tool lets you create a new Ethereum network down to  |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail.         |
|                                                           |
| Puppeth uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset.                                   |
+-----------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters please)
>
```

Geth: Go Ethereum (Private Blockchain)

- 네트워크 생성, 제네시스 블록 생성, PoW 설정

```
C:₩WINDOWS₩system32₩cmd.exe - puppeth
| docker-compose toolset.                                    |
+------------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters please)
> mynetwork

Sweet, you can set this via --network=mynetwork next time!

←[32mINFO ←[0m[11-26|23:14:36.677] Administering Ethereum network          ←[32mname←[0m=mynetwork
←[33mWARN ←[0m[11-26|23:14:36.763] No previous configurations found        ←[33mpath←[0m=.puppeth₩myn
etwork

What would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
> 2

What would you like to do? (default = create)
 1. Create new genesis from scratch
 2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
>
```

Geth: Go Ethereum (Private Blockchain)

```
C:₩WINDOWS₩system32₩cmd.exe - puppeth

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 1

Which accounts should be pre-funded? (advisable at least one)
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
>

Specify your chain/network ID if you want an explicit one (default = random)
> 1305
←[32mINFO ←[0m[11-26|23:16:29.942] Configured new genesis block

What would you like to do? (default = stats)
 1. Show network stats
 2. Manage existing genesis
 3. Track new remote server
 4. Deploy network components
>
```

1: Main 네트워크
2: 모던 테스트 네트워크 (이제 안씀)
3: Ropsten 테스트 네트워크
4: Rinkeby 테스트 네트워크
42: Kovan 테스트 네트워크

Geth: Go Ethereum (Private Blockchain)

```
C:₩WINDOWS₩system32₩cmd.exe - puppeth
Specify your chain/network ID if you want an explicit one (default = random)
> 1305
←[32mINFO ←[0m[11-26|23:16:29.942] Configured new genesis block

What would you like to do? (default = stats)
 1. Show network stats
 2. Manage existing genesis
 3. Track new remote server
 4. Deploy network components
> 2

 1. Modify existing configurations
 2. Export genesis configurations
 3. Remove genesis configuration
> 2

Which folder to save the genesis specs into? (default = current)
  Will create mynetwork.json, mynetwork-aleth.json, mynetwork-harmony.json, mynetwork-parity.json
>
←[32mINFO ←[0m[11-26|23:18:34.350] Saved native genesis chain spec         ←[32mpath←[0m=mynetwork.json
←[32mINFO ←[0m[11-26|23:18:34.352] Saved genesis chain spec                ←[32mclient←[0m=aleth ←[32mpath←[0m=mynetwork-aleth.json
←[32mINFO ←[0m[11-26|23:18:34.354] Saved genesis chain spec                ←[32mclient←[0m=parity ←[32mpath←[0m=mynetwork-parity.json
←[32mINFO ←[0m[11-26|23:18:34.356] Saved genesis chain spec                ←[32mclient←[0m=harmony ←[32mpath←[0m=mynetwork-harmony.json

What would you like to do? (default = stats)
 1. Show network stats
 2. Manage existing genesis
 3. Track new remote server
 4. Deploy network components
>
>
```

Ctrl + C

이름	수정한 날짜	유형	크기
.puppeth	2021-11-26 오후 11:16	파일 폴더	
mynetwork.json	2021-11-26 오후 11:18	JSON 원본 파일	21KB
mynetwork-aleth.json	2021-11-26 오후 11:18	JSON 원본 파일	23KB
mynetwork-harmony.json	2021-11-26 오후 11:18	JSON 원본 파일	21KB
mynetwork-parity.json	2021-11-26 오후 11:18	JSON 원본 파일	25KB

Geth: Go Ethereum (Private Blockchain)

명령 프롬프트

```
Microsoft Windows [Version 10.0.19043
(c) Microsoft Corporation. All rights

C:₩Users₩Kwak>cd /

C:₩>cd blockchain

C:₩blockchain>code mynetwork.json

C:₩blockchain>_
```

mynetwork.json - Visual Studio Code

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)

탐색기

열려 있는 편집기
- 확장: solidity
- Untitled-1
- MetaMask 상태 로그.json C:\Us...
- mynetwork.json C:\blockchain

열린 폴더 없음

개요

확장: solidity | Untitled-1 | MetaMask 상태 로그.json | mynetwork.json

C: > blockchain > mynetwork.json > ...

```
1  {
2    "config": {
3      "chainId": 1305,
4      "homesteadBlock": 0,
5      "eip150Block": 0,
6      "eip150Hash": "0x0000000000000000000000000000000000000000000000000000
7      "eip155Block": 0,
8      "eip158Block": 0,
9      "byzantiumBlock": 0,
10     "constantinopleBlock": 0,
11     "petersburgBlock": 0,
12     "istanbulBlock": 0,
13     "ethash": {}
14   },
15   "nonce": "0x0",
16   "timestamp": "0x61a0ebd3",
17   "extraData": "0x0000000000000000000000000000000000000000000000000000000
18   "gasLimit": "0x47b760",
19   "difficulty": "0x80000",
20   "mixHash": "0x0000000000000000000000000000000000000000000000000000000
21   "coinbase": "0x0000000000000000000000000000000000000000",
22   "alloc": {
23     "0000000000000000000000000000000000000000": {
24       "balance": "0x1"
25     },
26     "0000000000000000000000000000000000000001": {
27       "balance": "0x1"
28     },
29     "0000000000000000000000000000000000000002": {
30       "balance": "0x1"
31     },
32     "0000000000000000000000000000000000000003": {
33       "balance": "0x1"
34     },
35     "0000000000000000000000000000000000000004": {
```

⊗ 0 ⚠ 0 | 줄 1, 열 1 | 공백: 2 | UTF-8 | LF | JSON

...

```
791   },
792   "number": "0x0",
793   "gasUsed": "0x0",
794   "parentHash": "0x0000000000000000000000000000000000000000000000000000
795   "baseFeePerGas": null
796 }
```

Geth: Go Ethereum (Private Blockchain)

- Private 노드 초기화
  - > geth --datadir . init mynetwork.json

```
C:₩blockchain>code mynetwork.json

C:₩blockchain>geth --datadir . init mynetwork.json
INFO [11-27|14:36:50.080] Maximum peer count                       ETH=50 LES=0 total=50
INFO [11-27|14:36:50.131] Set global gas cap                       cap=50,000,000
INFO [11-27|14:36:50.136] Allocated cache and file handles         database=C:₩blockchain₩geth₩chaindata cache=16.00MiB handles=16
INFO [11-27|14:36:50.167] Writing custom genesis block
INFO [11-27|14:36:50.182] Persisted trie from memory database      nodes=354 size=50.23KiB time=1.3756ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-27|14:36:50.194] Successfully wrote genesis state         database=chaindata                    hash=02983c..4f2cdc
INFO [11-27|14:36:50.201] Allocated cache and file handles         database=C:₩blockchain₩geth₩lightchaindata cache=16.00MiB handles=16
INFO [11-27|14:36:50.216] Writing custom genesis block
INFO [11-27|14:36:50.223] Persisted trie from memory database      nodes=354 size=50.23KiB time=1.5559ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-27|14:36:50.233] Successfully wrote genesis state         database=lightchaindata                    hash=02983c..4f2cdc

C:₩blockchain>_
```

.puppeth	2021-11-26 오후 11:16	파일 폴더
geth	2021-11-27 오후 2:36	파일 폴더
keystore	2021-11-27 오후 2:36	파일 폴더
mynetwork.json	2021-11-26 오후 11:18	JSON 원본 파일
mynetwork-aleth.json	2021-11-26 오후 11:18	JSON 원본 파일
mynetwork-harmony.json	2021-11-26 오후 11:18	JSON 원본 파일
mynetwork-parity.json	2021-11-26 오후 11:18	JSON 원본 파일

Geth: Go Ethereum (Private Blockchain)

계정 생성

```
명령 프롬프트

C:₩blockchain>geth --datadir . account new
INFO [11-27|14:39:36.814] Maximum peer count                       ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0x6CD5b74f16d7dFdc77e87191e7F719EFc55E79c5
Path of the secret key file: keystore₩UTC--2021-11-27T05-39-52.545525700Z--6cd5b74f16d7dfdc77e87191e7f719efc55e79c5

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!


C:₩blockchain>
```

in › keystore keystore 검색

이름	수정한 날짜	유형	크기
UTC--2021-11-27T05-39-52.54552570...	2021-11-27 오후 2:39	545525700Z--6C...	1KB

계정을 2개 추가 생성 후 확인

- > geth --datadir . account list

```
C:₩blockchain>geth --datadir . account list
INFO [11-27|14:43:51.815] Maximum peer count                       ETH=50 LES=0 total=50
INFO [11-27|14:43:51.861] Set global gas cap                       cap=50,000,000
Account #0: {6cd5b74f16d7dfdc77e87191e7f719efc55e79c5} keystore://C:₩blockchain₩keystore₩UTC--2021-11-27T05-39-52.545525700Z--6cd5b74f16d7dfdc77e87191e7f719efc55e79c5
Account #1: {c40b835bf684d0cc966fe03a4001e27594da6600} keystore://C:₩blockchain₩keystore₩UTC--2021-11-27T05-42-08.387343700Z--c40b835bf684d0cc966fe03a4001e27594da6600
Account #2: {e0ddc7c6bf2d0a9a6e85fa973089e0bb0c841ca5} keystore://C:₩blockchain₩keystore₩UTC--2021-11-27T05-42-19.889946400Z--e0ddc7c6bf2d0a9a6e85fa973089e0bb0c841ca5
```

Geth: Go Ethereum (Private Blockchain)

- Private node 실행

```
C:₩blockchain>code nodestart.cmd

C:₩blockchain>
```

  - geth --networkid 1305 --mine --miner.threads 2 --datadir "./" --nodiscover --http --http.port "8545" --http.corsdomain "*" --nat "any" --http.api eth,web3,personal,net --http.addr “127.0.0.1” --allow-insecure-unlock --password ./password.sec

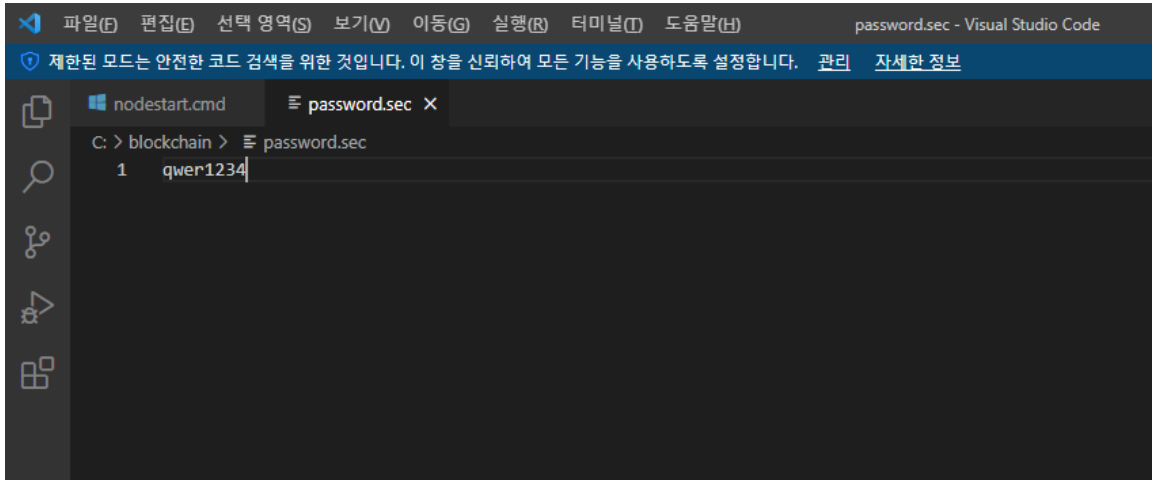
  - 슬라이드 #41에서 port번호를 8546으로 했을 경우, --http.port는 8546으로 변경하여 설정

Geth: Go Ethereum (Private Blockchain)

- Password.sec 파일 생성

```
C:₩blockchain>code password.sec
```

  - 패스워드 설정



Geth: Go Ethereum (Private Blockchain)

```
WARN [11-28|18:55:02.301] Unclean shutdown detected        booted=2021-11-28T18:50:50+0900 age=4m12s
WARN [11-28|18:55:02.312] Unclean shutdown detected        booted=2021-11-28T18:51:40+0900 age=3m22s
WARN [11-28|18:55:02.325] Unclean shutdown detected        booted=2021-11-28T18:52:48+0900 age=2m14s
INFO [11-28|18:55:02.331] Starting peer-to-peer node       instance=Geth/v1.10.13-stable-7a0c19f8/windows
INFO [11-28|18:55:02.368] IPC endpoint opened              uri=\\.\pipe\geth.ipc
INFO [11-28|18:55:02.380] New local node record            seq=1,638,092,296,747 id=6506d8faaeba36bb ip=1
30303
INFO [11-28|18:55:02.401] Started P2P networking           self="enode://27634067d4987124e9379417b7e42c34
3275b048ebd554f32b57031b46140210696245 4d6cc1cfd81be46529d1dad77d9b9b5ca2e3805@127.0.0.1:30303?discport=0"
INFO [11-28|18:55:02.391] HTTP server started              endpoint=127.0.0.1:8545 prefix= cors=* vhosts=
INFO [11-28|18:55:02.436] Transaction pool price threshold updated price=1,000,000,000
INFO [11-28|18:55:02.445] Updated mining threads           threads=2
INFO [11-28|18:55:02.452] Transaction pool price threshold updated price=1,000,000,000
INFO [11-28|18:55:02.458] Etherbase automatically configured address=0xC9b3f70a59170c985d9cA5e8e0E449667dF3
INFO [11-28|18:55:02.468] Commit new mining work           number=1 sealhash=e67bf4..94e000 uncles=0 txs=
ed=0s
INFO [11-28|18:55:04.464] Generating DAG in progress       epoch=0 percentage=0 elapsed=1.424s
INFO [11-28|18:55:05.890] Generating DAG in progress       epoch=0 percentage=1 elapsed=2.849s
INFO [11-28|18:55:07.313] Generating DAG in progress       epoch=0 percentage=2 elapsed=4.273s
INFO [11-28|18:55:08.735] Generating DAG in progress       epoch=0 percentage=3 elapsed=5.695s
INFO [11-28|18:55:10.143] Generating DAG in progress       epoch=0 percentage=4 elapsed=7.103s
INFO [11-28|18:55:11.572] Generating DAG in progress       epoch=0 percentage=5 elapsed=8.532s
INFO [11-28|18:55:13.099] Generating DAG in progress       epoch=0 percentage=6 elapsed=10.058s
INFO [11-28|18:55:14.580] Generating DAG in progress       epoch=0 percentage=7 elapsed=11.540s
```

Geth: Go Ethereum (Private Blockchain)

```
INFO [11-28|18:57:17.875] Generating DAG in progress          epoch=0 percentage=93 elapsed=2m14.834s
INFO [11-28|18:57:19.293] Generating DAG in progress          epoch=0 percentage=94 elapsed=2m16.253s
INFO [11-28|18:57:20.699] Generating DAG in progress          epoch=0 percentage=95 elapsed=2m17.659s
INFO [11-28|18:57:22.111] Generating DAG in progress          epoch=0 percentage=96 elapsed=2m19.070s
INFO [11-28|18:57:23.508] Generating DAG in progress          epoch=0 percentage=97 elapsed=2m20.467s
INFO [11-28|18:57:24.912] Generating DAG in progress          epoch=0 percentage=98 elapsed=2m21.871s
INFO [11-28|18:57:26.480] Generating DAG in progress          epoch=0 percentage=99 elapsed=2m23.440s
INFO [11-28|18:57:26.491] Generated ethash verification cache epoch=0 elapsed=2m23.450s
INFO [11-28|18:57:34.545] Successfully sealed new block       number=1 sealhash=e67bf4..94e000 hash=4458c4..2de683 elapsed=2m32
.077s
INFO [11-28|18:57:34.560] ◆ mined potential block              number=1 hash=4458c4..2de683
INFO [11-28|18:57:34.554] Commit new mining work              number=2 sealhash=7bba9b..dd0ddc uncles=0 txs=0 gas=0 fees=0 elap
sed="598.4µs"
INFO [11-28|18:57:35.697] Generating DAG in progress          epoch=1 percentage=0  elapsed=2.237s
INFO [11-28|18:57:37.906] Successfully sealed new block       number=2 sealhash=7bba9b..dd0ddc hash=533742..461ffd elapsed=3.35
2s
INFO [11-28|18:57:37.919] ◆ mined potential block              number=2 hash=533742..461ffd
INFO [11-28|18:57:37.945] Generating DAG in progress          epoch=1 percentage=1  elapsed=4.485s
INFO [11-28|18:57:37.917] Commit new mining work              number=3 sealhash=9b30d1..9cbe9b uncles=0 txs=0 gas=0 fees=0 elap
sed=0s
INFO [11-28|18:57:40.196] Generating DAG in progress          epoch=1 percentage=2  elapsed=6.736s
INFO [11-28|18:57:42.035] Successfully sealed new block       number=3 sealhash=9b30d1..9cbe9b hash=a51426..b71984 elapsed=4.11
8s
INFO [11-28|18:57:42.123] ◆ mined potential block              number=3 hash=a51426..b71984
INFO [11-28|18:57:42.053] Commit new mining work              number=4 sealhash=78923b..5ce0b2 uncles=0 txs=0 gas=0 fees=0 elap
sed=0s
INFO [11-28|18:57:42.489] Generating DAG in progress          epoch=1 percentage=3  elapsed=9.028s
```

Geth: Go Ethereum (Private Blockchain)

- 새로운 PowerShell로 입력

```
INFO [11-28|19:08:06.883] Transaction pool price threshold updated
INFO [11-28|19:08:06.892] Commit new mining work
psed=0s
INFO [11-28|19:08:09.178] Successfully sealed new block
85s
INFO [11-28|19:08:09.178] block reached canonical chain
INFO [11-28|19:08:09.185] Commit new mining work
psed=6.601ms
INFO [11-28|19:08:09.187] mined potential block
INFO [11-28|19:08:09.756] Successfully sealed new block
.131ms
INFO [11-28|19:08:09.757] block reached canonical chain
INFO [11-28|19:08:09.788] Commit new mining work
psed=32.115ms
INFO [11-28|19:08:09.790] mined potential block
INFO [11-28|19:08:09.813] Successfully sealed new block
318ms
INFO [11-28|19:08:09.813] block reached canonical chain
INFO [11-28|19:08:09.888] Commit new mining work
psed=75.503ms
INFO [11-28|19:08:09.891] mined potential block
INFO [11-28|19:08:15.603] Updated mining threads
INFO [11-28|19:08:15.609] Transaction pool price threshold updated
INFO [11-28|19:08:15.621] Commit new mining work
psed="829µs"
INFO [11-28|19:08:21.586] Successfully sealed new block
64s
INFO [11-28|19:08:21.586] block reached canonical chain
INFO [11-28|19:08:21.610] Commit new mining work
psed=24.560ms
INFO [11-28|19:08:21.610] mined potential block
INFO [11-28|19:08:24.647] Successfully sealed new block
61s
INFO [11-28|19:08:24.647] block reached canonical chain
INFO [11-28|19:08:24.674] Commit new mining work
psed=27.033ms
INFO [11-28|19:08:24.674] mined potential block
INFO [11-28|19:08:30.837] Successfully sealed new block
90s
INFO [11-28|19:08:30.837] block reached canonical chain
INFO [11-28|19:08:30.858] Commit new mining work
psed=20.980ms
INFO [11-28|19:08:30.858] mined potential block
INFO [11-28|19:08:32.071] Successfully sealed new block
34s
INFO [11-28|19:08:32.071] block reached canonical chain
INFO [11-28|19:08:32.100] Commit new mining work
psed=29.162ms
INFO [11-28|19:08:32.101] mined potential block
```

```
관리자: Windows PowerShell
PS C:₩Users₩kwack₩Blockchain> geth attach ipc:₩₩.₩pipe₩geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.13-stable-7a0c19f8/windows-amd64/go1.17.2
coinbase: 0xc9b3f70a59170c985d9ca5e8e0e449667df3889d
at block: 71 (Sun Nov 28 2021 19:06:38 GMT+0900 (KST))
 datadir: C:₩Users₩kwack₩Blockchain
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0

To exit, press ctrl-d or type exit
> eth.coinbase
"0xc9b3f70a59170c985d9ca5e8e0e449667df3889d"
> eth.accounts
["0xc9b3f70a59170c985d9ca5e8e0e449667df3889d", "0x5ae3657f97982e9a
b5236a8c64afa15a"]
> eth.getBalance(eth.accounts[1])
0
> eth.getBalance(eth.coinbase)
150000000000000000000
> web3.fromWei(eth.getBalance(eth.coinbase), "ether")
172
> miner.stop()
null
> miner.start()
null
> miner.stop()
null
> miner.start(2)
null
> miner.stop()
null
>
```

Geth: Go Ethereum (Private Blockchain)

- Send transaction

```
> eth.sendTransaction({from:eth.coinbase, to:eth.accounts[1], value:web3.toWei(20,"ether")})
"0x686cc78b2ff3f364a9daa082dd0c61ce6693df7db80ba74e3c5d7b56dedcb498"
```

Transactions

```
INFO [11-28|19:18:15.228] Submitted transaction    hash=0x686cc78b2ff3f364a9daa082dd0c61ce6693df7db80ba74e3c5d7b56de
dcb498 from=0xC9b3f70a59170c985d9cA5e8e0E449667dF3889d nonce=0 recipient=0x5aE3657f97982e9Af35b03A942a96E7Ac0294db1 value=20,000,000
,000,000,000,000
```

Balance checking

```
> web3.fromWei(eth.getBalance(eth.accounts[1]), "ether")
20
>
```

Geth: Go Ethereum (Private Blockchain)

- Send transaction
  - 다음과 같은 에러가 발생했을 경우

```
> eth.sendTransaction({from:eth.coinbase, to:eth.accounts[1], value:web3.toWei(20,"ether")})
Error: authentication needed: password or unlock
        at web3.js:6357:37(47)
        at send (web3.js:5091:62(35))
        at <eval>:1:20(19)
```

  - 아래와 같이 입력

```
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xc9b3f70a59170c985d9ca5e8e0e449667df3889d
Passphrase:
true
```

Truffle

- 새로운 PowerShell or Cmd

관리자: Windows PowerShell

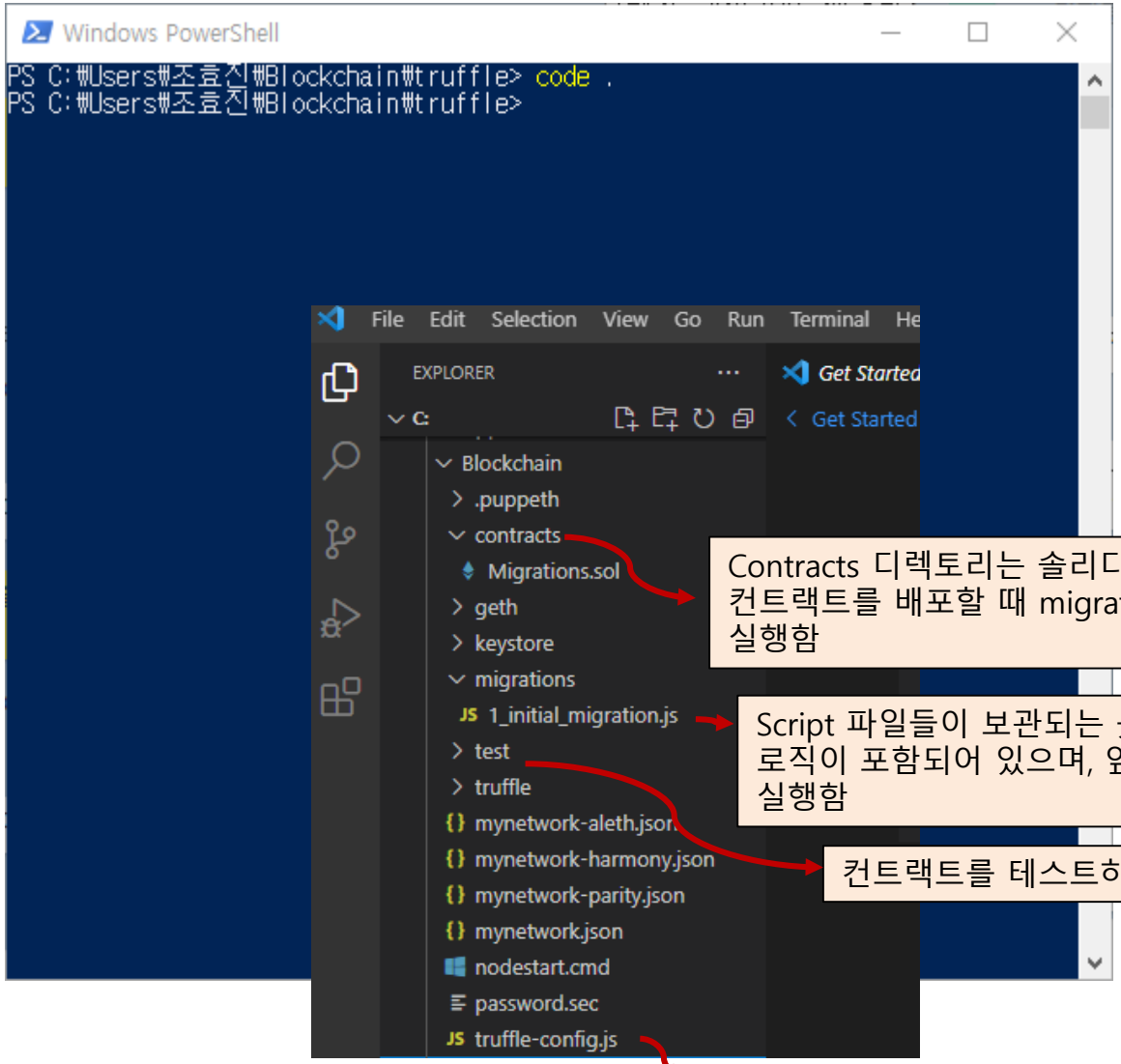
```
PS C:₩Users₩kwack₩Blockchain> mkdir truffle


    디렉터리: C:₩Users₩kwack₩Blockchain


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----      2021-11-28   오후 7:25                truffle


PS C:₩Users₩kwack₩Blockchain> dir


    디렉터리: C:₩Users₩kwack₩Blockchain


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----      2021-11-28   오후 6:00                .puppeth
d-----      2021-11-28   오후 6:55                geth
d-----      2021-11-28   오후 6:38                keystore
d-----      2021-11-28   오후 7:25                truffle
-a----      2021-11-28   오후 6:01          22736 mynetwork-aleth.json
-a----      2021-11-28   오후 6:01          21318 mynetwork-harmony.json
-a----      2021-11-28   오후 6:01          24793 mynetwork-parity.json
-a----      2021-11-28   오후 6:01          21318 mynetwork.json
-a----      2021-11-28   오후 6:54            242 nodestart.cmd
-a----      2021-11-28   오후 6:38              8 password.sec


PS C:₩Users₩kwack₩Blockchain>
```

Truffle

- PowerShell에서 실행 안되면, Cmd 창에서 실행

```
C:₩Users₩kwack₩Blockchain>truffle init

Starting init...
================

> Copying project files to C:₩Users₩kwack₩Blockchain

Init successful, sweet!

Try our scaffold commands to get started:
  $ truffle create contract YourContractName # scaffold a contract
  $ truffle create test YourTestName         # scaffold a test

http://trufflesuite.com/docs


C:₩Users₩kwack₩Blockchain>
```

Truffle



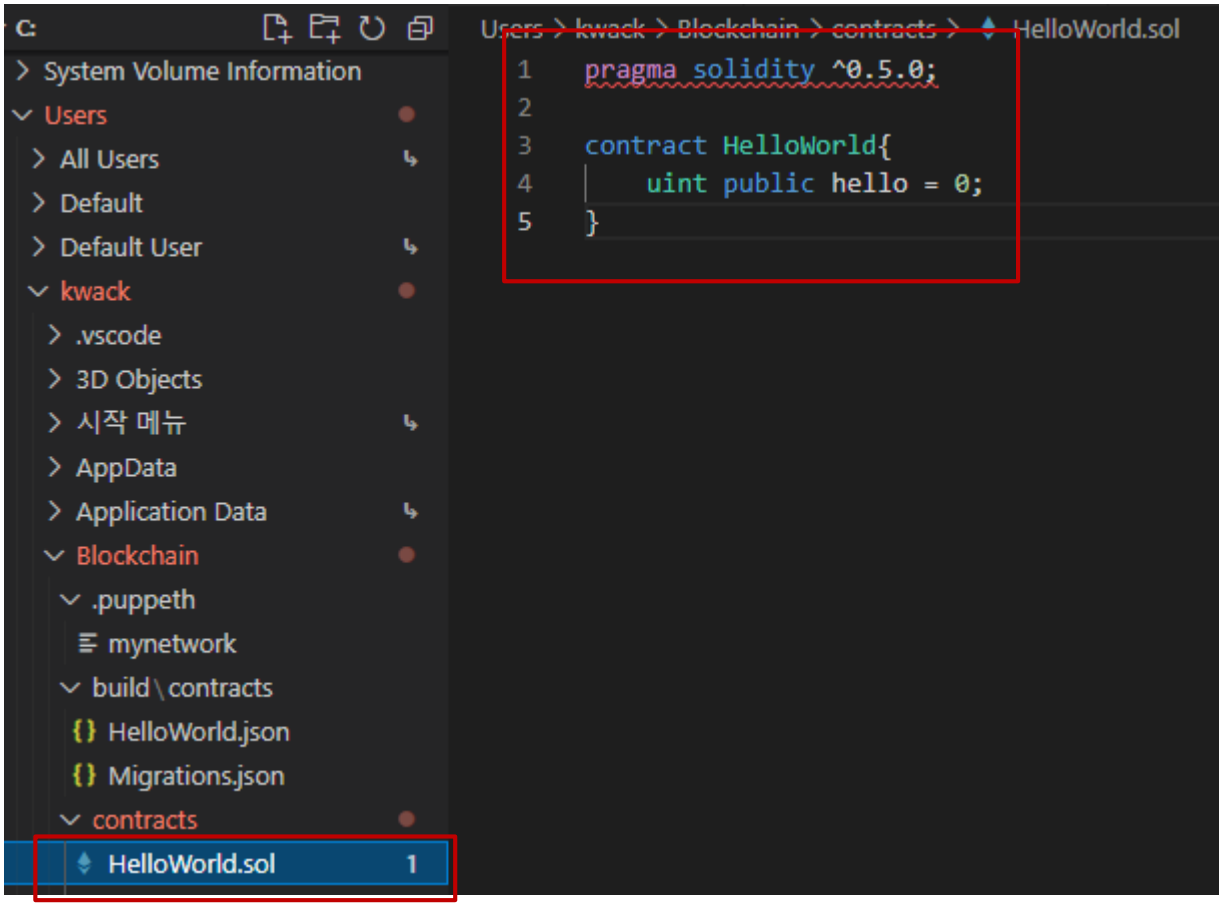
Contracts 디렉토리는 솔리디티 컨트랙트들을 보관하는 곳으로, 컨트랙트를 배포할 때 migrations 폴더에 있는 script 파일들을 실행함

Script 파일들이 보관되는 곳으로, script 파일에 배포 과정에 대한 로직이 포함되어 있으며, 앞의 숫자를 통해 순차적으로 script를 실행함

컨트랙트를 테스트하는데 쓰이는 디렉토리

컨트랙트 설정을 바꿀 수 있음

Truffle + Ganache



Truffle + Ganache

- HelloWorld.sol 파일을 생성 후,
  - > truffle compile

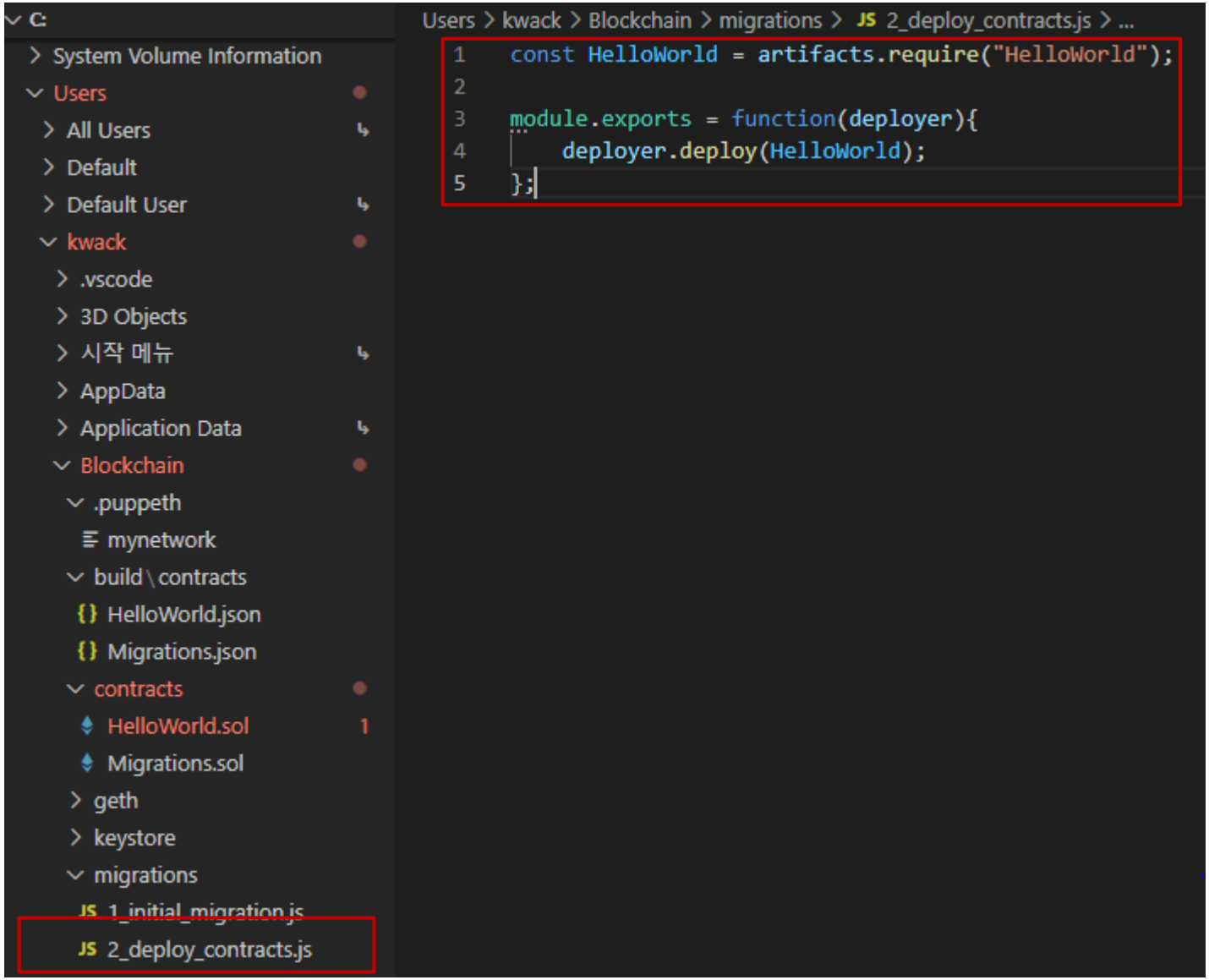
```
C:₩bc>truffle compile

Compiling your contracts...
===========================
> Compiling .₩contracts₩HelloWorld.sol
> Compiling .₩contracts₩Migrations.sol
> Artifacts written to C:₩bc₩build₩contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

  - 컴파일 하고 난 뒤에, “build” 폴더가 생성되며 “.json” 파일이 생성되는데 해당 파일에는 ABI와 Bytecode가 생성됨
    - ABI는 웹 어플리케이션에서 사용
    - Bytecode는 블록체인 내부에 올라감

Truffle + Ganache

- Ganache는 로컬 가상 이더리움
  - 로컬에서 Ganache를 이용해서 컨트랙트를 배포하는 것
  - 배포를 위해
    - Migration 폴더에 '2_deploy_contract.js' 파일을 생성

Truffle + Ganache



Truffle + Ganache

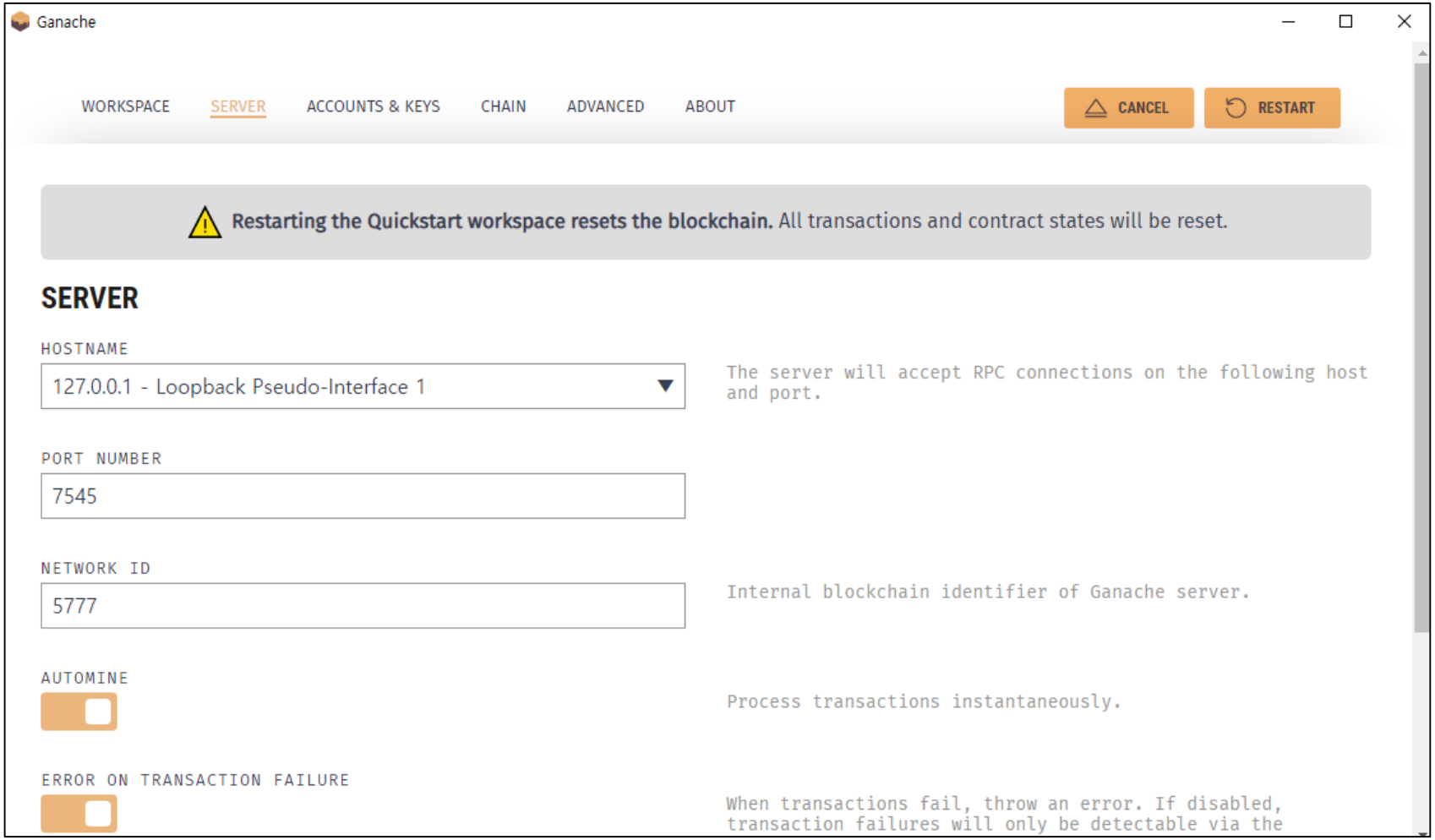
```
migrations
  1_initial_migration.js
  2_deploy_contracts.js
test
  .gitkeep
truffle
mynetwork-aleth.json
mynetwork-harmony.json
mynetwork-parity.json
mynetwork.json
nodestart.cmd
package.json
password.sec
truffle-config.js
```

```
35    */
36
37    networks: {
38      ganache: {
39        host: "127.0.0.1",
40        port: 7545,
41        network_id: "*",
42      }
43      // Useful for testing. The `development` name is special - truffle uses it
44      // if it's defined here and no other network is specified at the command l
45      // You should run a client (like ganache-cli, geth or parity) in a separat
46      // tab if you use this network and you must also set the `host`, `port` an
47      // options below to some value.
48      //
49      // development: {
50      //  host: "127.0.0.1",     // Localhost (default: none)
51      //  port: 8545,            // Standard Ethereum port (default: none)
```

...

```
// Configure your compilers
compilers: {
  solc: {
    version: "0.5.1",
    optimizer:{
      enabled: true,
      runs: 200
    }
    // Fetch exact version from solc-b
```

Truffle + Ganache (설정)



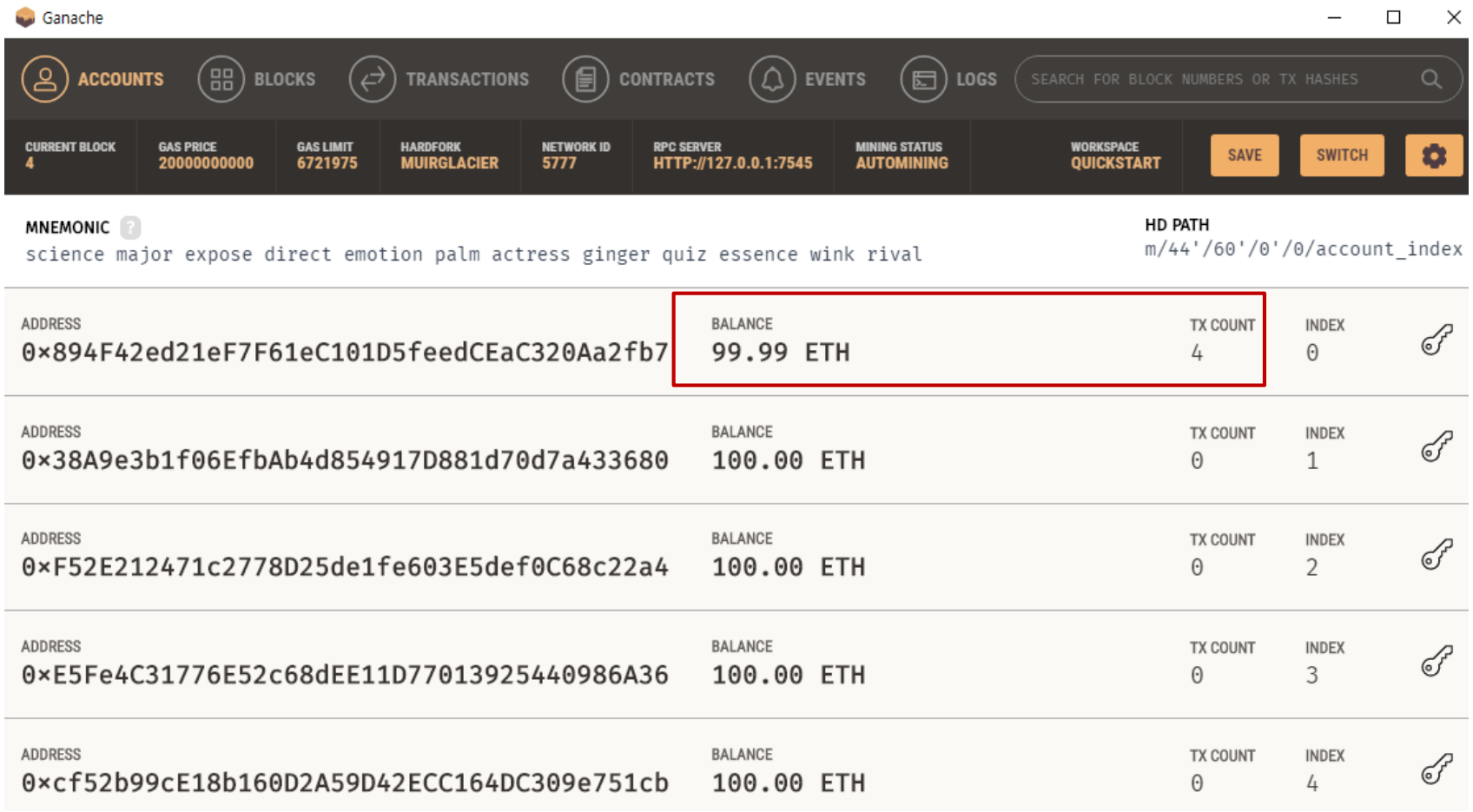
Truffle + Ganache (Truffle migration)

- PowerShell/Cmd 창에서 “> truffle migrate”

C:₩Windows₩system32₩cmd.exe

```
C:₩Users₩kwack₩Blockchain>truffle migrate

Compiling your contracts...
===========================
> Compiling .₩contracts₩HelloWorld.sol
> Compiling .₩contracts₩Migrations.sol
> Artifacts written to C:₩Users₩kwack₩Blockchain₩build₩contracts
> Compiled successfully using:
   - solc: 0.5.1+commit.c8a2cb62.Emscripten.clang



Starting migrations...
======================
> Network name:    'ganache'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0x65a08f60621da808a5c3b4a36380eff00399e48f544cf2a497d6a82d74e7b826
   > Blocks: 0            Seconds: 0
   > contract address:    0x7eD67723F680d7D7AedE01D4C0ACF6251fC9955f
   > block number:        1
   > block timestamp:     1638100808
   > account:             0x894F42ed21eF7F61eC101D5feedCEaC320Aa2fb7
   > balance:             99.9958945
   > gas used:            205275 (0x321db)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0041055 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.0041055 ETH


2_deploy_contracts.js
=====================

   Deploying 'HelloWorld'
   ----------------------
```

Truffle + Ganache (컨트랙트 전송 확인)



Truffle + Ganache (컨트랙트 확인)

- > Truffle console
- > HelloWorld.deployed()

```
truffle(ganache)> HelloWorld.deployed()
TruffleContract {
  constructor: [Function: TruffleContract] {
    _constructorMethods: {
      configureNetwork: [Function: configureNetwork],
      setProvider: [Function: setProvider],
      new: [Function: new],
      at: [AsyncFunction: at],
      deployed: [AsyncFunction: deployed],
      defaults: [Function: defaults],
      hasNetwork: [Function: hasNetwork],
      isDeployed: [Function: isDeployed],
      detectNetwork: [AsyncFunction: detectNetwork],
      setNetwork: [Function: setNetwork],
      setNetworkType: [Function: setNetworkType],
      setWallet: [Function: setWallet],
      resetAddress: [Function: resetAddress],
      link: [Function: link],
      clone: [Function: clone],
      addProp: [Function: addProp],
      toJSON: [Function: toJSON],
      decodeLogs: [Function: decodeLogs]
    },
    _properties: {
      contract_name: [Object],
      contractName: [Object],
      gasMultiplier: [Object],
      timeoutBlocks: [Object],
      autoGas: [Object],
      numberFormat: [Object],
      abi: [Object],
      metadata: [Function: metadata],
      network: [Function: network],
      networks: [Function: networks],
      address: [Object],
      transactionHash: [Object],
      links: [Function: links],
      events: [Function: events],
      binary: [Function: binary],
      deployedBinary: [Function: deployedBinary],
      unlinked_binary: [Object],
```

References

- Lecture slides from BLOCKCHAIN @ BERKELEY
- https://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/
- "Mastering Ethereum - Building Smart Contracts and Dapps"
- https://slides.com/ironpark/parity-smart-contract#/5

Q & A