

Connect WorldLand test network

Ver. 2023-02-02

Writer : Jong-nam Hong, @Libervance

Updated on 2022-02-02, by Libervance

목차

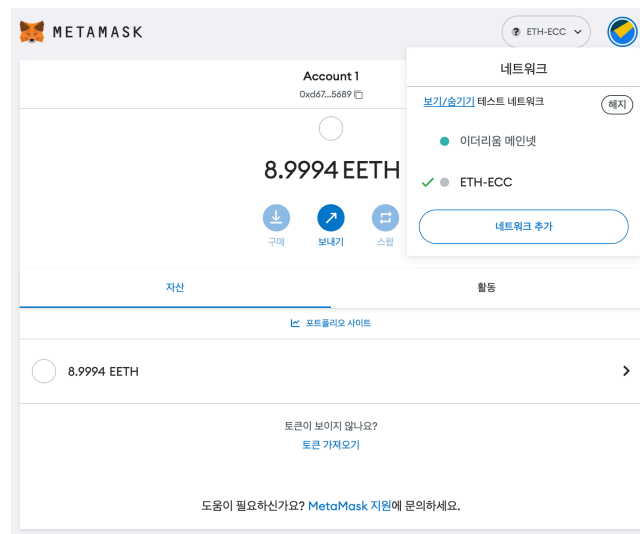
1. WorldLand test network 와 지갑연동 방법
2. WorldLand test network 에 SmartContract 배포
3. Front-end SmartContract API 활용.

1. WorldLand test network 와 지갑연동 방법

1-1. 크롬브라우저 확장탭에서 메타마스크를 설치 후 확장합니다.

- Link : <https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaochlefnkodbefgpgknn>

1-2. 메타마스크 계정을 생성해 주시고, 우측 상단에 네트워크 네트워크 추가를 누릅니다.



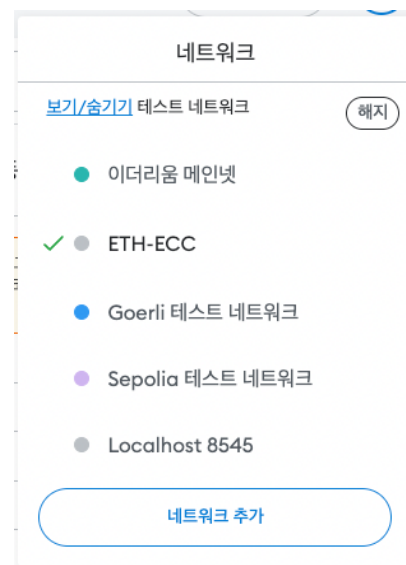
1-3. 아래의 네트워크 설정값을 입력합니다.

네트워크 이름 : ETH-ECC

새 RPC URL : <https://rpc.lvscan.io>

체인 ID : 12345

통화기호 : eETH

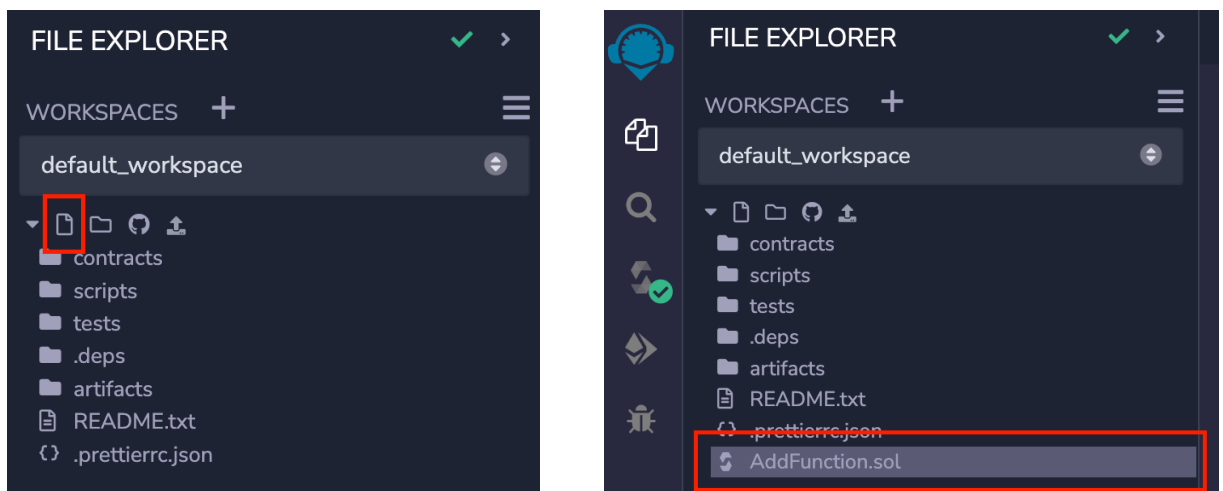


2. WorldLand test network SmartContract 배포 방법

2-1. Remix-IDE : <https://remix.ethereum.org/> 접속합니다.

2-2. 그림과 같이 파일을 추가하여 Solidity 파일을 생성합니다.

(파일명은 상관없습니다. 저는 AddFunction.sol 파일로 만들었습니다.)



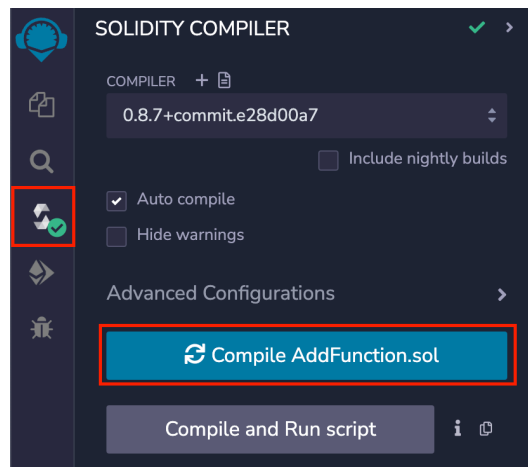
2-3. 아래의 간단한 코드를 생성한 파일에 붙여 넣습니다.

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

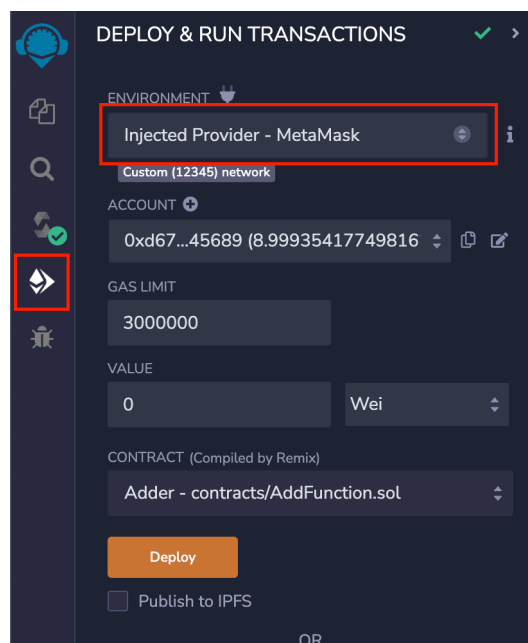
contract Adder {
    function add(uint256 a, uint256 b) public pure returns (uint256) {
        return a + b;
    }
}
```

2-4. Compile tap 으로 이동 후 Compile 버튼을 누릅니다.

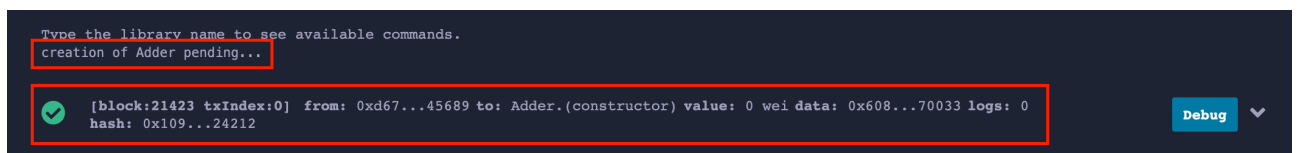


2-5. Deploy tap 으로 이동 후 Environment : Injected Provider - MetaMask 로 설정합니다.

설정이 완료 되었으면 Deploy 버튼을 누릅니다.



2-6. WorldLand test net 배포를 위해 최소 3분에서 최대 10분정도 기다립니다.



위의 그림과 같이 트랜잭션이 처리 되었다면 WorldLand test net 에 배포가 완료된 상태가 됩니다.
링크를 통해 처리된 트랜잭션을 확인할 수 있습니다. Link : <https://lyscan.io/#/>

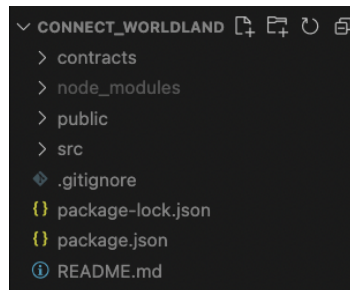
3. Front-end SmartContract API 활용

3-1. 배포한 SmartContract 의 API 를 Front-end code를 통해 연동하겠습니다.

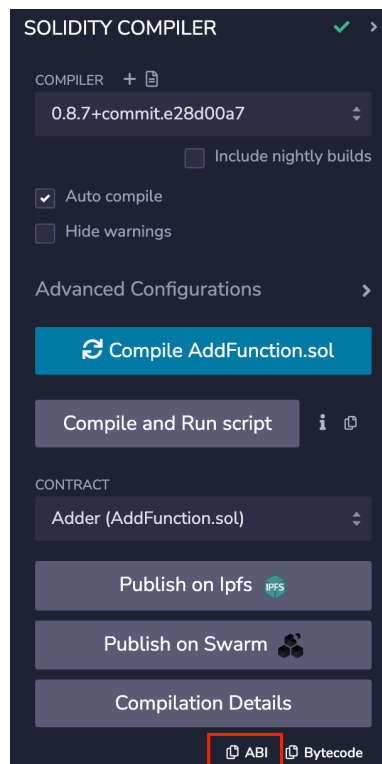
3-2. Github 링크로 이동 Connect_WorldLand 레파지토리를 Clone 합니다.

Github Link : https://github.com/cryptoecc/connect_worldland

3-3. Clone 후 npm install 합니다.



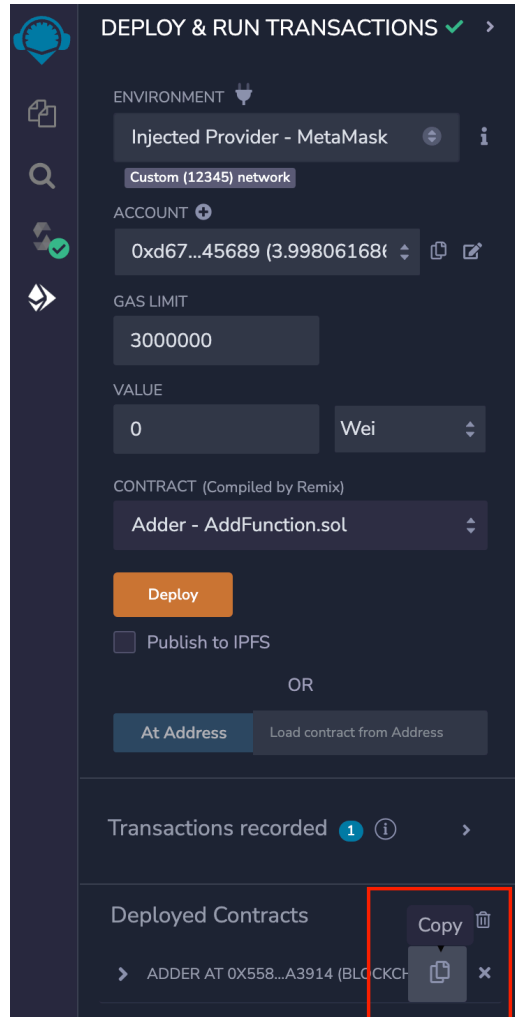
3-4. 이전에 배포했던 스마트컨트랙트를 사용하기 위해 abi 파일을 그림과 같이 복사합니다.



(Remix)

3-5. Src / abis / add_function.sol 파일로 이동하여 복사한 Abi 코드로 기존의 코드를 덮어씹습니다.

3-6. App.js 로 이동하여 CA = 값을 지우고 배포한 컨트랙트의 CA값으로 교체합니다.



(Remix)

3-7. 교체를 하셨으면 터미널에서 `npm start`

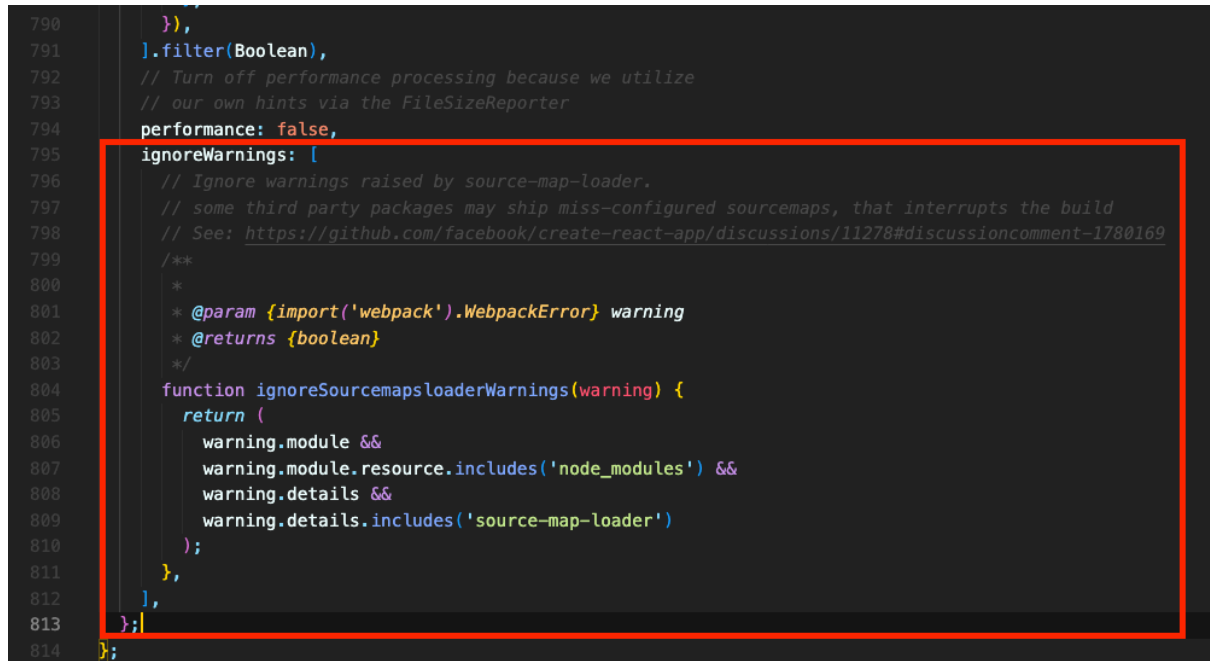
**** (React-script 버전 문제로 수많은 Warning 문구가 뜰 경우 아래내용 참조) ****

```

ignoreWarnings: [
  // Ignore warnings raised by source-map-loader.
  // some third party packages may ship miss-configured
  sourcemaps, that interrupts the build
  // See: https://github.com/facebook/create-react-app/discussions/11278#discussioncomment-1780169
  /**
   *
   * @param {import('webpack').WebpackError} warning
   * @returns {boolean}
   */
  function ignoreSourcemapsloaderWarnings(warning) {
    return (
      warning.module &&
      warning.module.resource.includes('node_modules') &&
      warning.details &&
      warning.details.includes('source-map-loader')
    );
  },
],

```

위의 코드를 복사하시고 node_modules > react-scripts > config > webpack.config.js 경로로 이동



```

790     },
791     .filter(Boolean),
792     // Turn off performance processing because we utilize
793     // our own hints via the FileSizeReporter
794     performance: false,
795     ignoreWarnings: [
796       // Ignore warnings raised by source-map-loader.
797       // some third party packages may ship miss-configured sourcemaps, that interrupts the build
798       // See: https://github.com/facebook/create-react-app/discussions/11278#discussioncomment-1780169
799       /**
800        *
801        * @param {import('webpack').WebpackError} warning
802        * @returns {boolean}
803        */
804       function ignoreSourcemapsloaderWarnings(warning) {
805         return (
806           warning.module &&
807           warning.module.resource.includes('node_modules') &&
808           warning.details &&
809           warning.details.includes('source-map-loader')
810         );
811       },
812     ],
813   },
814 ];

```

Performance : false, 코드 아래에 위의 코드를 붙여넣습니다. 그리고 다시 npm start

3-8. 아래와 같이 순서대로 진행 합니다.

2. 숫자입력 3. 숫자입력 1. 지갑연결

<input type="text"/>	+	<input type="text"/>	=	Result: 0	지갑연결
----------------------	---	----------------------	---	-----------	------

4. 계산버튼

Connecting Wallet :

트랜잭션 처리시간 : 3분 ~ 10분 사이로 기다리시면 결과값이 출력됩니다.