

# CÀI ĐẶT MÔI TRƯỜNG PHÁT TRIỂN BLOCKCHAIN\_\_\_\_\_

**Biên soạn:** Nhóm nghiên cứu về Blockchain Khoa HTTT

---

## 1. Giới thiệu về ngôn ngữ lập trình Solidity<sup>[1]</sup>

- Solidity là ngôn ngữ **đối tượng hợp đồng, ngôn ngữ lập trình bậc cao** dùng để thực thi Smart Contracts.
  - Solidity bị ảnh hưởng cao bởi ngôn ngữ lập trình C++, Python và JavaScript.
  - Ngôn ngữ này được thiết kế dành cho máy ảo Ethereum hay gọi cách khác là Ethereum Virtual Machine (EVM).
  - Solidity là ngôn ngữ lập trình thao tác tính, hỗ trợ tính kế thừa, thư viện và các kiểu dữ liệu phức tạp do người dùng định nghĩa.
  - **Ứng dụng của Smart Contract:** Tạo bình chọn, huy động vốn cộng đồng, đấu giá và ví điện tử, ...
- ❖ **Yêu cầu để học ngôn ngữ lập trình Solidity:** Biết công nghệ Blockchain và kiến thức lập trình cơ bản nói chung.

## 2. Thiết lập môi trường ảo Solidity<sup>[2]</sup>

- ❖ Cách 1. **Sử dụng IDE online: Remix IDE**
- **Đường dẫn:** <https://remix.ethereum.org/>
  - **Ưu điểm:** Dễ dàng sử dụng tiện lợi, không phải cài đặt.
  - **Nhược điểm:** Khó quản lý Source code trên Git, Tính bảo mật source không cao, khó tích hợp trực tiếp với HTML, CSS và Javascript.
- ❖ Cách 2. **Cài đặt môi trường bằng npm/Node.js**
- **Cài đặt Node.js**
    - Link tải về: <https://nodejs.org/en/>
    - Cài đặt Node.js và thiết lập biến môi trường npm
    - Vào terminal kiểm tra kết quả bằng lệnh: **npm --version**

```
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>npm --version
8.6.0

C:\Windows\system32>
```

## ▪ Cài đặt solc

- Cài đặt solcjs bằng câu lệnh như sau:

```
npm install -g solc
```

- Kiểm tra phiên bản solc:

```
solcjs --version
```

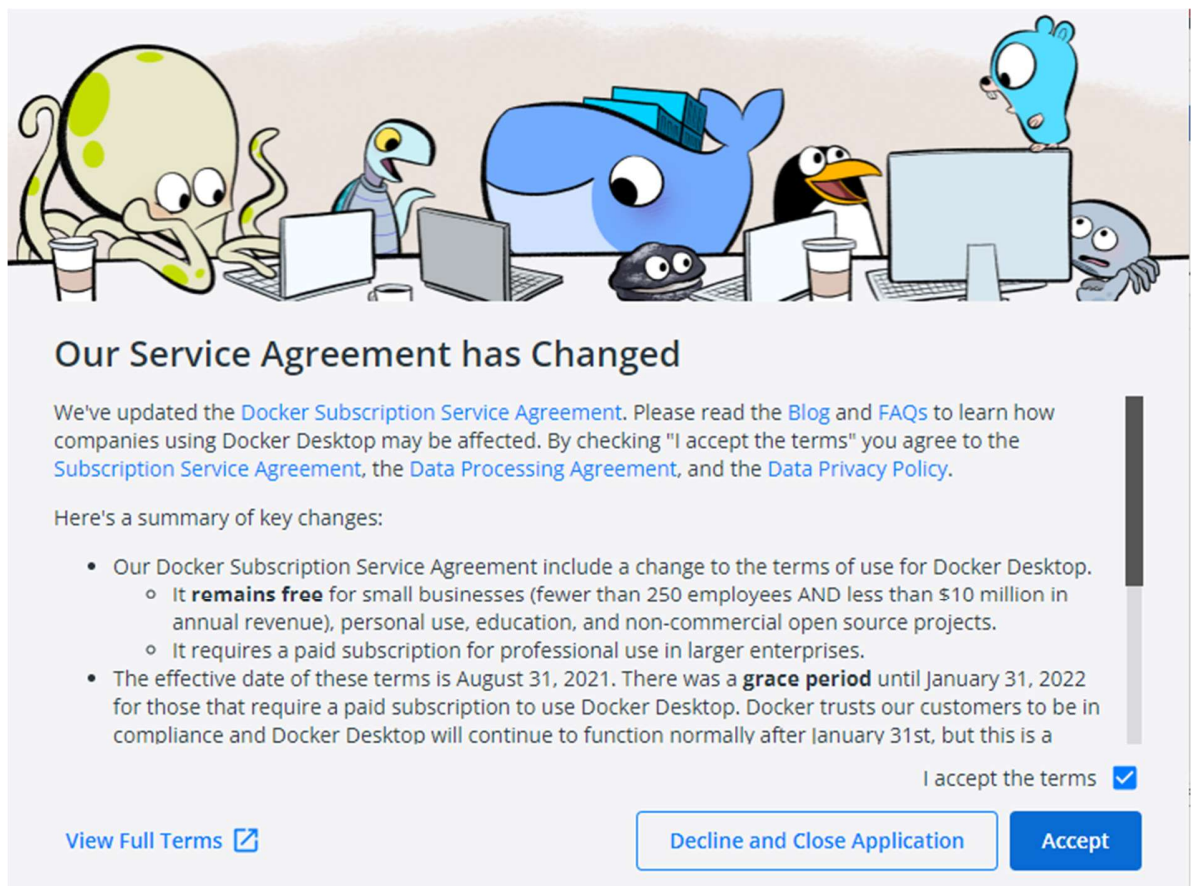
```
C:\Windows\system32>solcjs --version
0.8.13+commit.abaa5c0e.Emscripten.clang
C:\Windows\system32>
```

## ❖ Cách 2. **Cài bằng Docker Image**

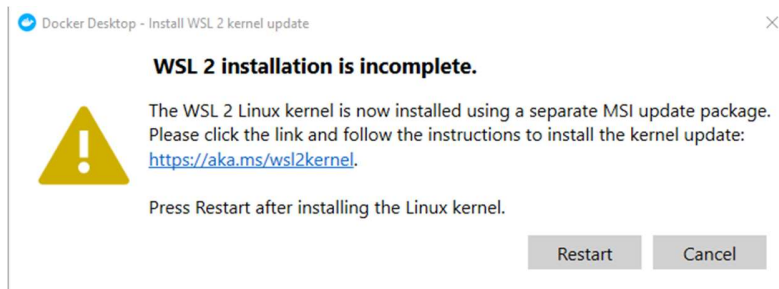
- Link tải về Docker cho Windows:

<https://docs.docker.com/desktop/windows/install/>

- Chấp nhận điều khoản docker → **Accept**



- Cài đặt WSL 2 theo đường dẫn cung cấp → Sau đó restart lại Docker



- Cài đặt solc: stable và kiểm tra bằng docker

#### Install Solc

> `docker pull ethereum/solc:stable`

```
C:\Users\nguye\Desktop\cmdr
λ docker pull ethereum/solc:stable
stable: Pulling from ethereum/solc
c7aac9bd019b: Pull complete
Digest: sha256:6c8791ffbe6ca05f35ead15b880c12b7683722d20c2df09fd1b33ef63e9c4be4
Status: Downloaded newer image for ethereum/solc:stable
docker.io/ethereum/solc:stable
```

#### Check Install

> `docker run ethereum/solc:stable --version`

```
C:\Users\nguye\Desktop\cmdr
λ docker run ethereum/solc:stable --version
solc, the solidity compiler commandline interface
Version: 0.8.13+commit.abaa5c0e.Linux.g++
```

### 3. Cài đặt ví điện tử lưu trữ tiền ảo Metamask

Link tham khảo về ví Metamask: <https://wiki.tino.org/metamask-la-gi/>

- Cài đặt Extensions ví Metamask bằng Extension của Chrome

**Đường dẫn cài đặt:**

<https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=en-US>

- Chọn **Get Started** để bắt đầu



## Welcome to MetaMask

Connecting you to Ethereum and the Decentralized Web.

We're happy to see you.

Get Started

- Chọn tạo ví MetaMask hoặc **import MetaMask** cũ với **Secret Key**



METAMASK

## New to MetaMask?



No, I already have a Secret Recovery Phrase

Import your existing wallet using a Secret Recovery Phrase

Import wallet

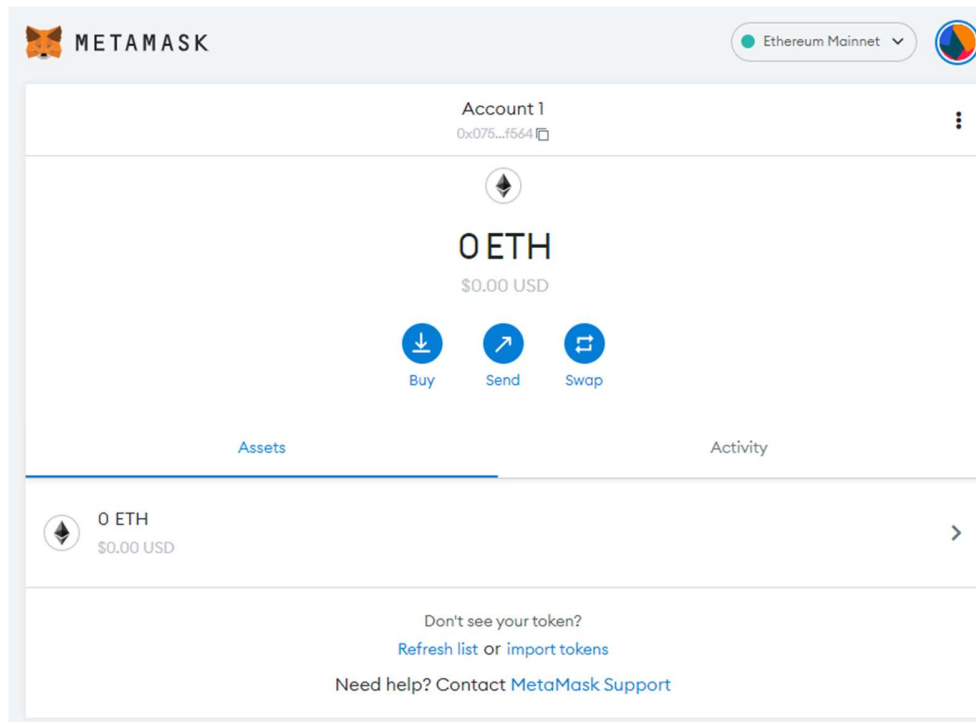


Yes, let's get set up!

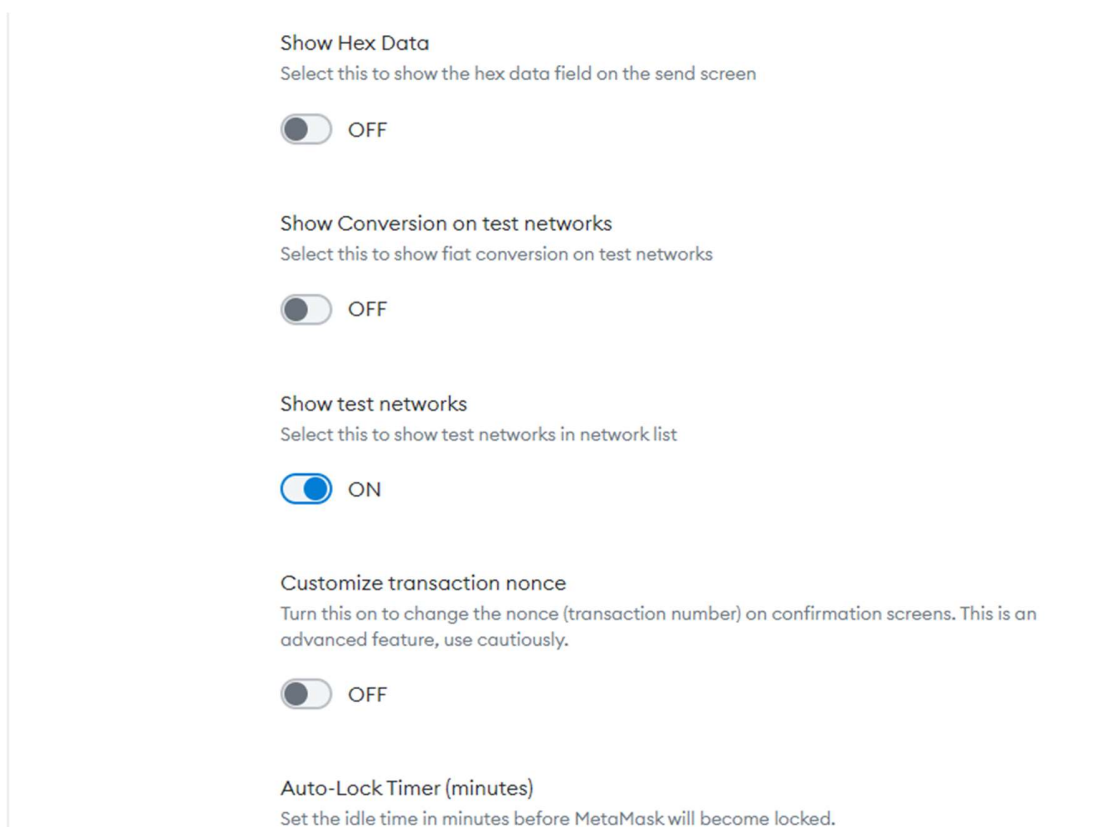
This will create a new wallet and Secret Recovery Phrase

Create a Wallet

- Giả sử tạo mới **một ví mới**. Kết quả sau khi tạo thành công **ví MetaMask** sẽ có kết quả như hình bên dưới.



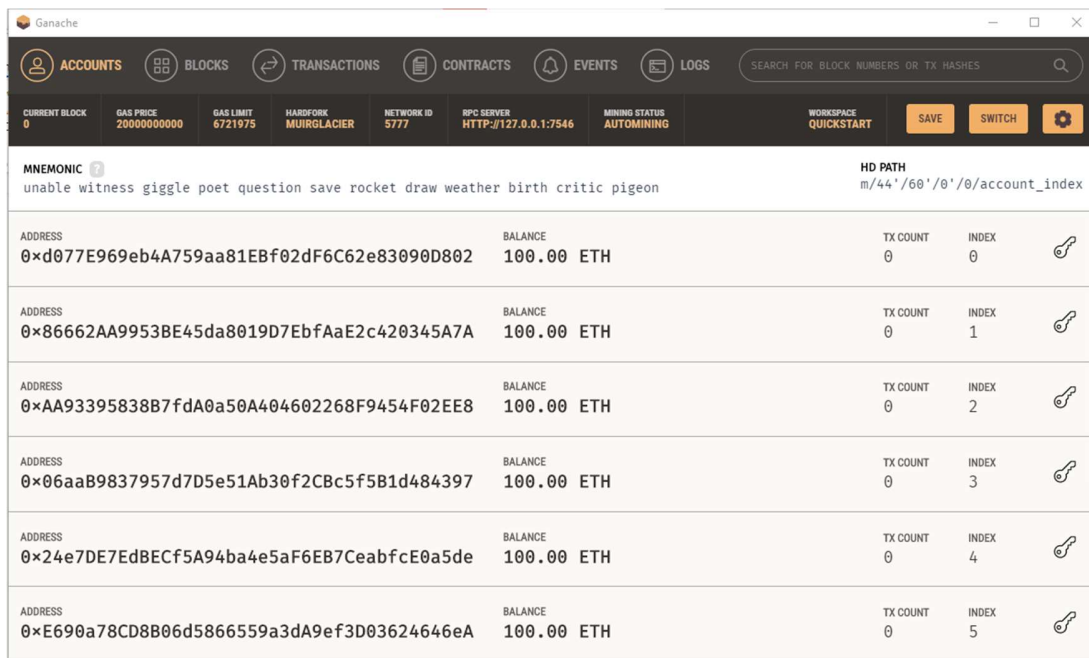
- Vào phần “**Settings**” vào phần “**Advanced**”.



- Chọn Show **Test network** để có thể thử ví **meta test**.
- Sau khi thiết lập xong ví **MetaMask**, tiến hành lên trang chính của từng ví để **xin tiền test** phục vụ cho các tác vụ.

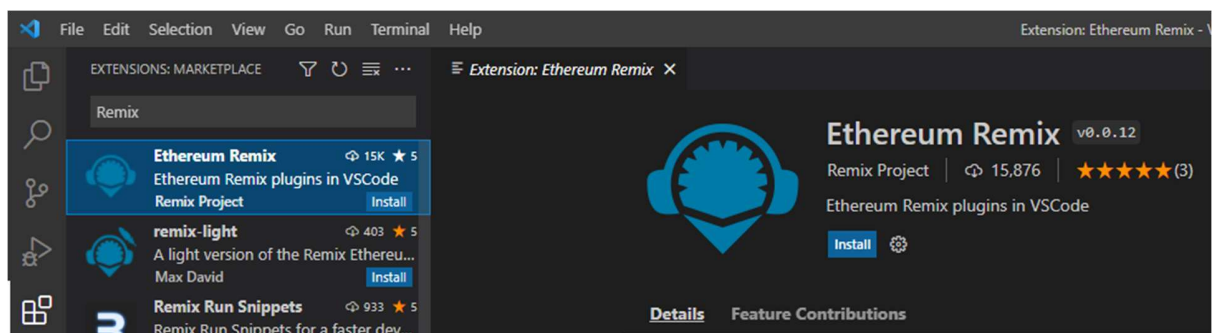
#### 4. Cài đặt Ganache<sup>[3]</sup>

- Link tải Ganache: <https://trufflesuite.com/ganache/>
- Ganache là một Blockchain cá nhân dùng để test và phát triển trên nền tảng Ethereum.
- Chọn chế độ QuickStart Truffle để bắt đầu một Smart Contract nhanh.



#### 5. Thiết lập môi trường phát triển solidity cho IDE Visual Code<sup>[3]</sup>

- Sau khi cài đặt môi trường ảo cho Solidity. Bước tiếp theo sẽ thiết lập môi trường phát triển IDE Visual Code.
- **IDE thực hiện:** Visual Code
- **Extension:** Ethereum Remix, Solidity

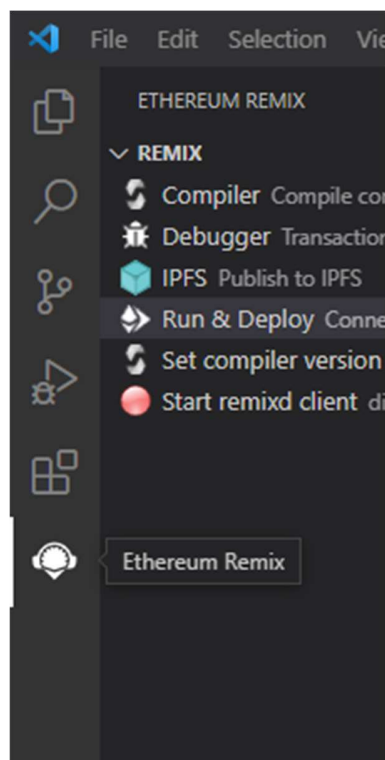


- Sau khi cài đặt xong tiến hành viết thử một Smart Contract đơn giản như sau:

## 00\_HelloWorld.sol

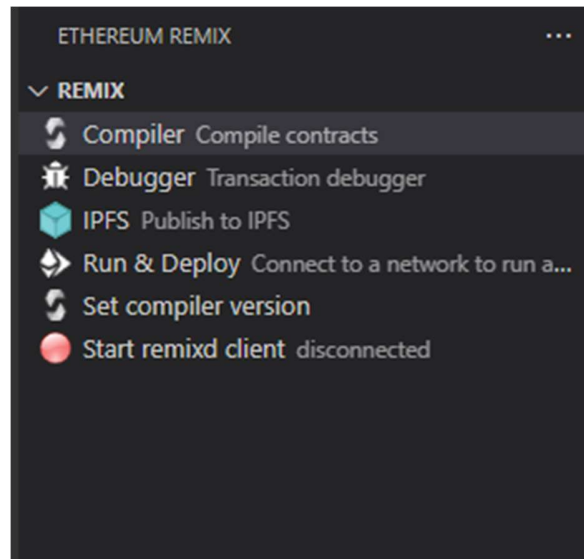
```
1  pragma solidity ^0.8.6;
2
3  contract HelloWorld {
4
5      string private message = "Hello world";
6
7      function getMessage() view public returns(string memory) {
8          return message;
9      }
10 }
11
```

- Nhấn save file để lưu lại. Nếu không lưu file máy tính sẽ lấy bản lưu cuối cùng.
- Chọn biểu tượng Ethereum Remix ở góc phải

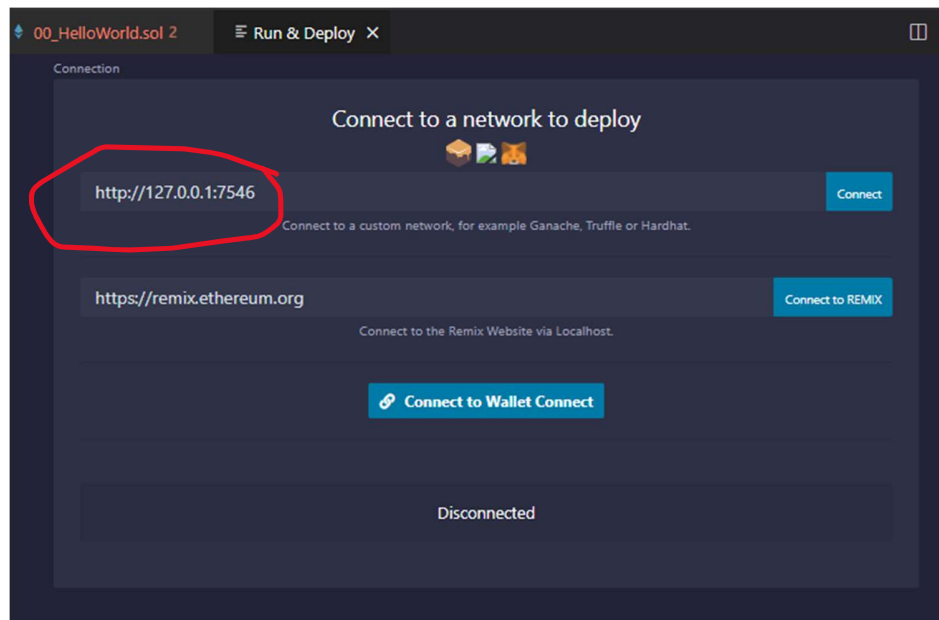


- Thực hiện các bước **compiler** và **run** như sau đây:
  - Bước 1: Chọn **Compiler** để Compiler Smart Contract





- Bước 2: Nhấn **Run & Deploy** → Active

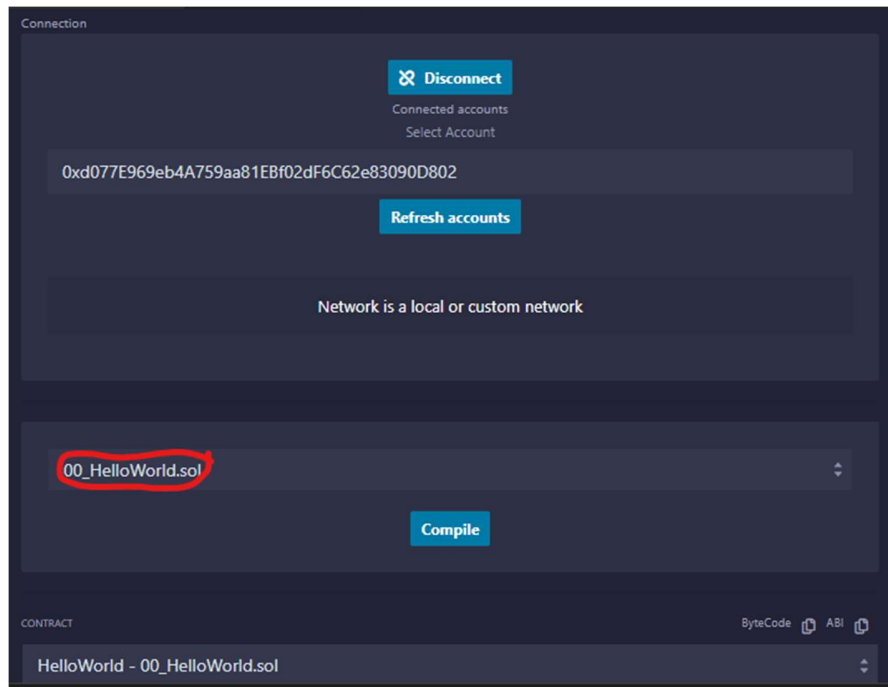


- Thay đổi địa chỉ **Run & Deploy** là địa chỉ trong **Truffle Ganache** (Nằm trong phần cài đặt → Server)

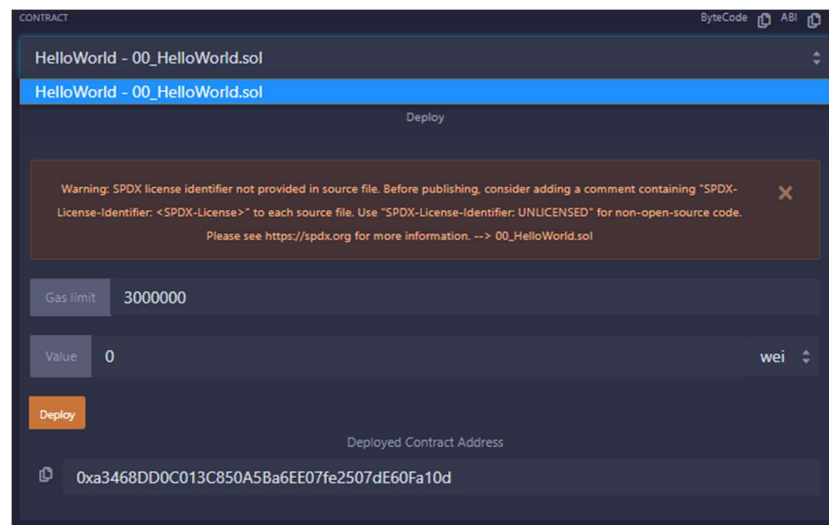
## 6. Chạy một Smart Contract

- Chọn file cần **Compile**. File cần compile có đuôi **\*.sol** → Nhấn nút **Compile**

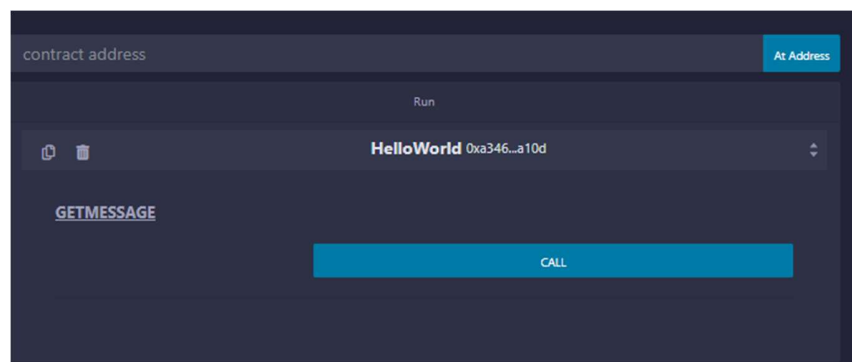




- Chọn **Contract** cần **Deploy** thiết lập giá trị **Gas Limit** và **Value**. Nhấn Deploy.



- Gọi những function cài đặt ở phần dưới đây:



## 7. Tài liệu tham khảo

- [1] <https://www.tutorialspoint.com/solidity/index.htm>, “Solidity – Home, Solidity – Overview”. [Online] [Thời gian truy cập: 26/04/2022].
- [2] <https://www.tutorialspoint.com/solidity/index.htm>, “Solidity - Environment Setup”. [Online] [Thời gian truy cập: 26/04/2022].
- [3] <https://www.geeksforgeeks.org/setting-up-smart-contract-development-environment/>, “Setting Up Smart Contract Development Environment”. [Online] [Thời gian truy cập: 26/04/2022].