LESSON INTRO 3 - Intro to Remix IDE

University of West Attica

Department of Electrical and Electronics Engineering

Ioannis Christidis

Christoforos Kachris

Support by Ethereum Foundation ESP

What is Remix IDE

Remix IDE is an open-source, web-based development environment for creating, testing, debugging, and deploying smart contracts (SCs) on the Ethereum blockchain.

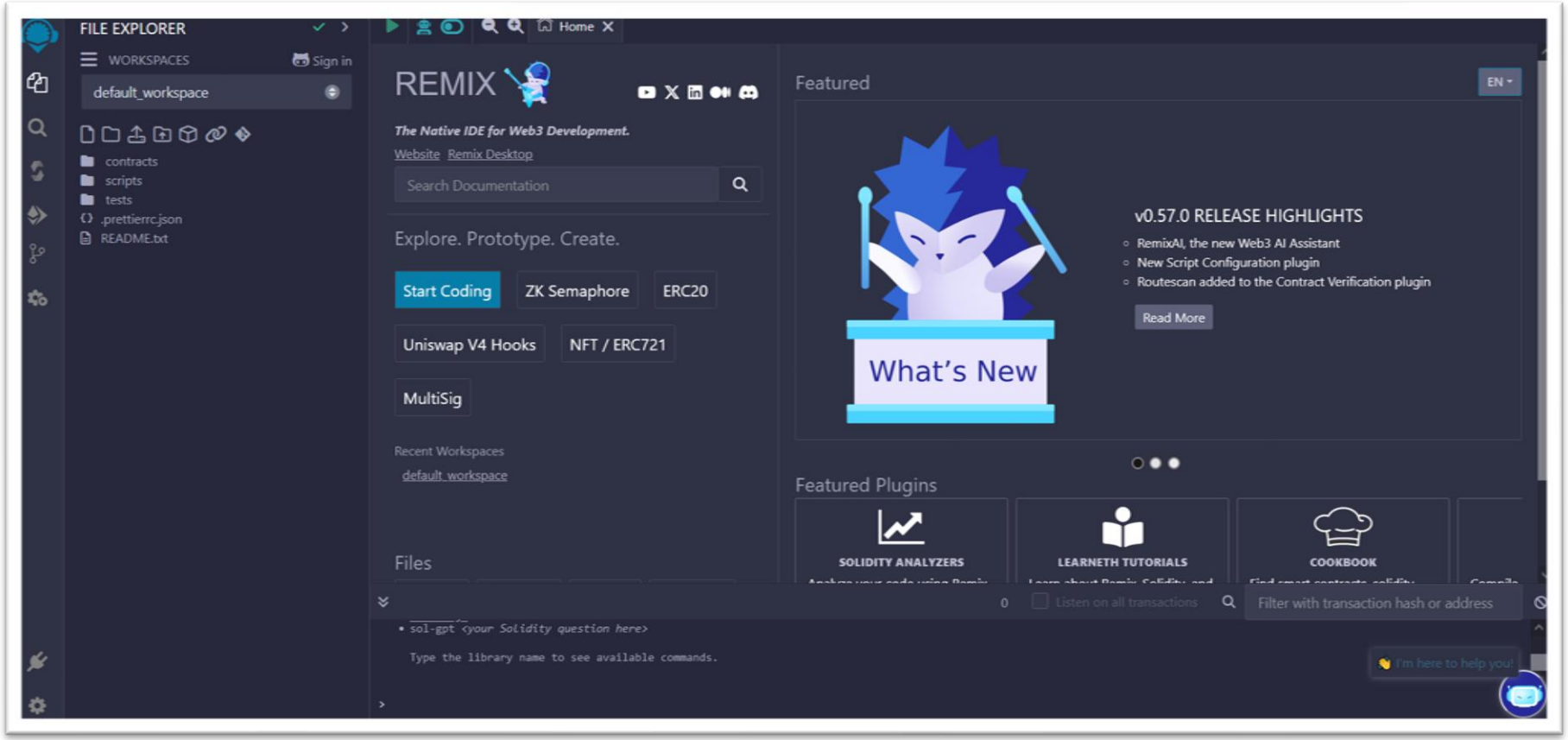
In this course, we will use Remix IDE to learn the basics of SCs.

The following link will transfer you to the landing page:

https://remix.ethereum.org

How to use Remix IDE

By clicking the link you will land to a page similar to this.



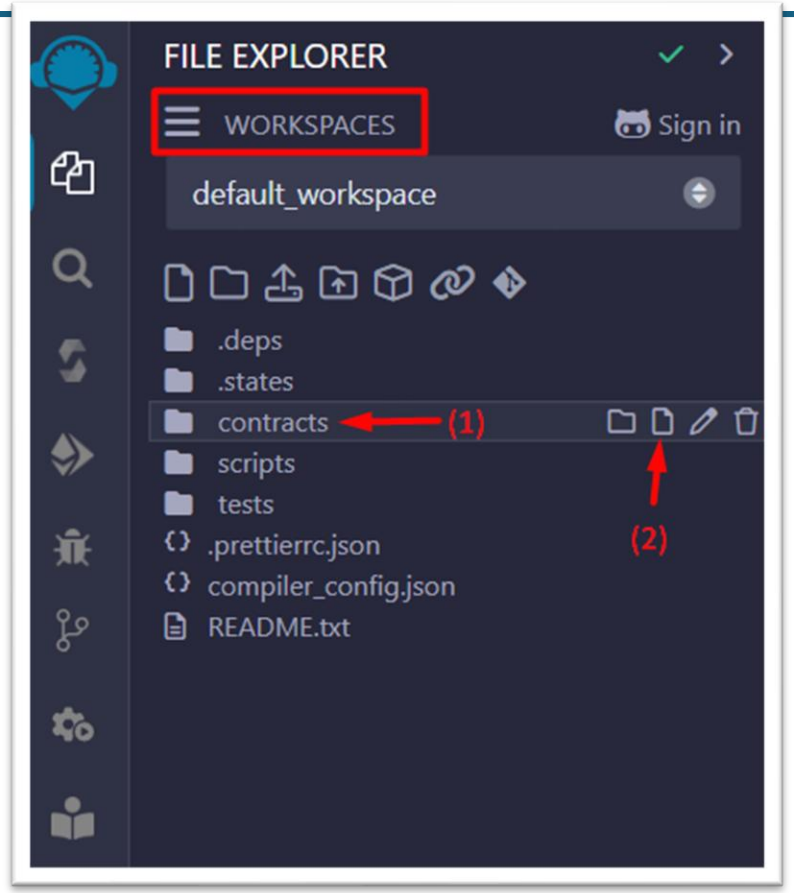
Usage of Remix IDE

In the workspaces you can see some folders.

We are only going to use the `contracts` folder (1).

If you click on it you can see some already created SCs.

By clicking the small file icon (2) you can create a new SC.



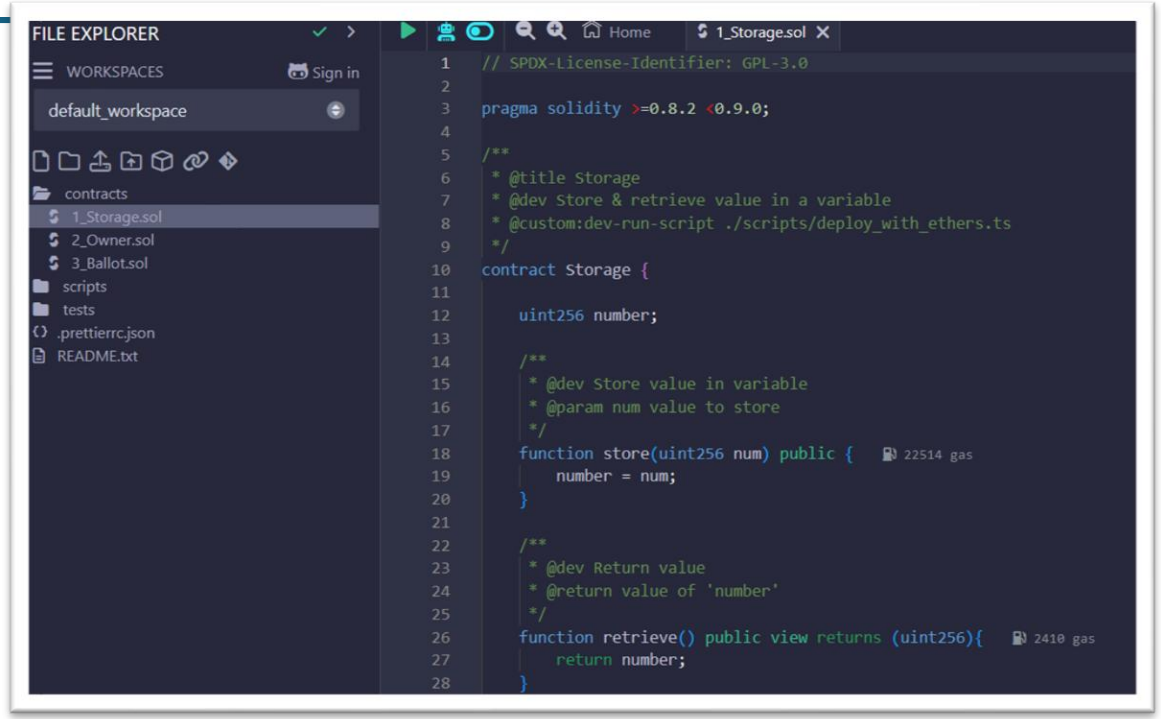
Opening up a Smart Contract

Click on the *contracts* folder and you will see some already created SCs.

Click the `Storage.sol` (The name may be different like `1_Storage.sol` etc).

The SC code will appear in the *code editor* next to the *workspace*.

The SC that you see is written in ***Solidity***.

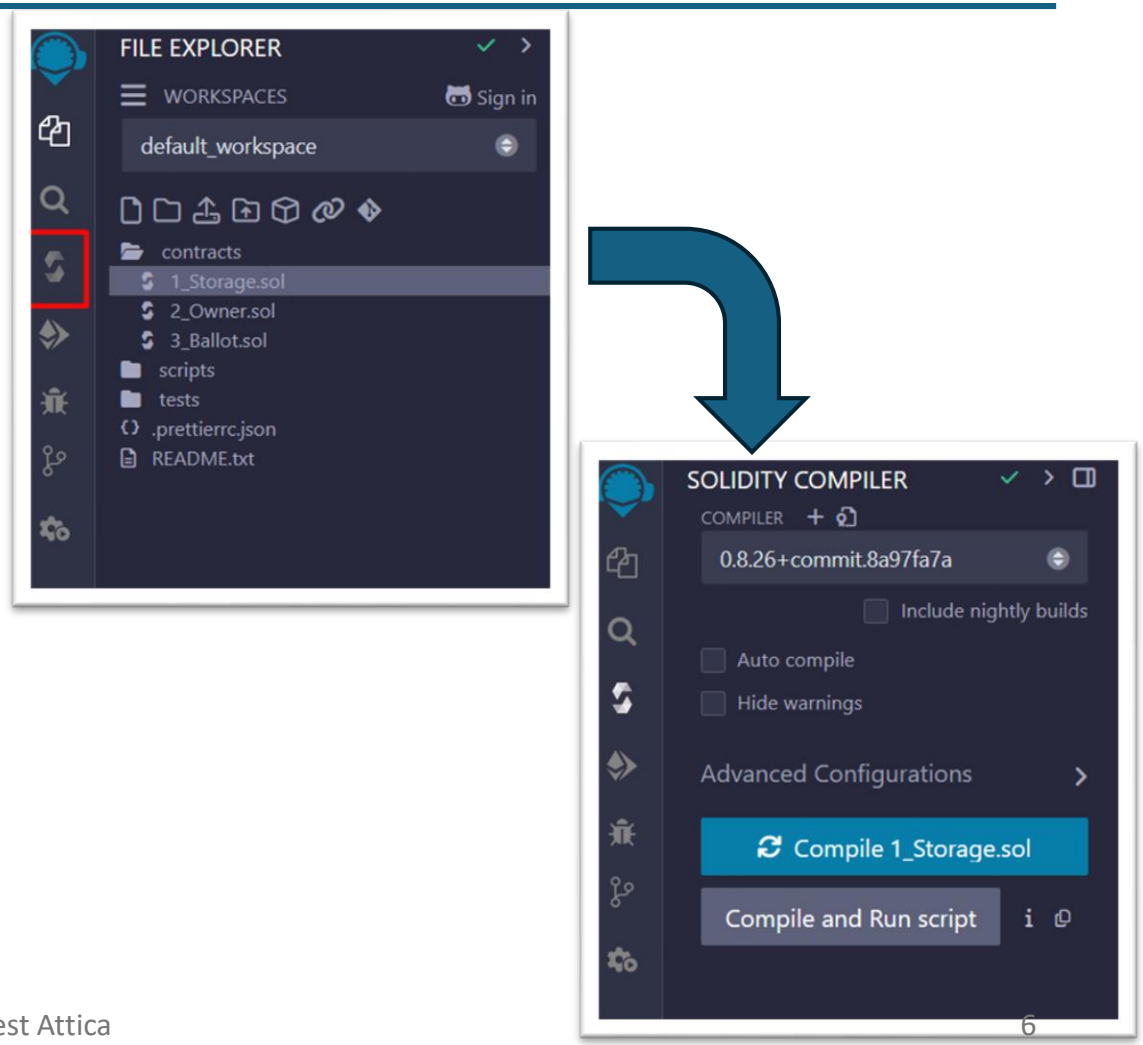


Solidity is a high-level, statically-typed programming language specifically designed for developing smart contracts on Ethereum and other blockchain platforms. It is similar in syntax to JavaScript, C++, and Python. The ***solidity*** files have the extension `.sol`.

Compiling a Smart Contract

The process of publishing a SC in the blockchain is called *deployment*. During *deployment*, the *Ethereum Virtual Machine (EVM)* executes the SC's bytecode. But, in order to get the bytecode, we need to *compile* the SC.

Click on the icon pointed in the image and you will see the solidity compiler.

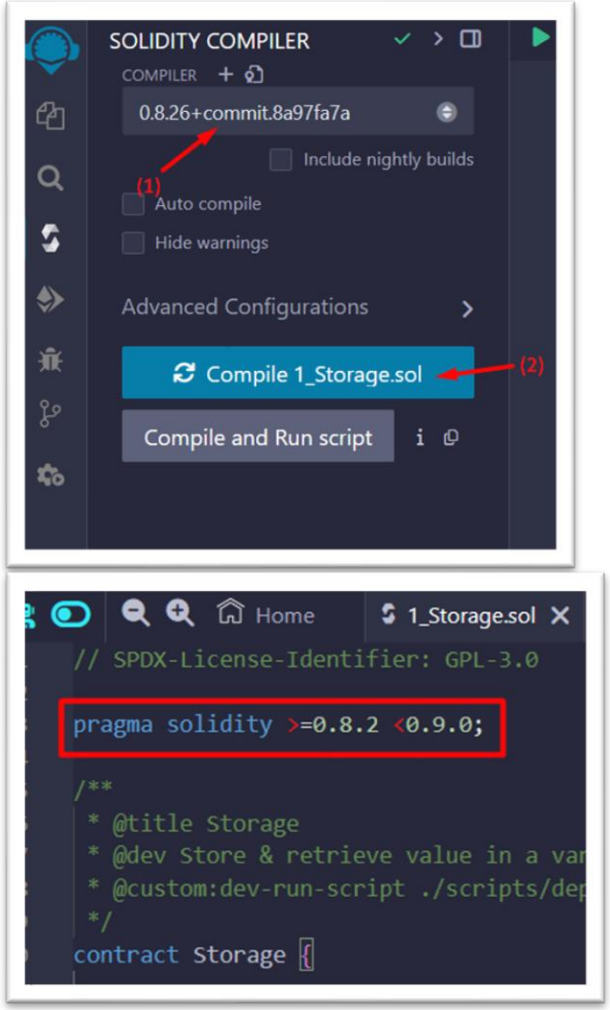


Compiling a Smart Contract

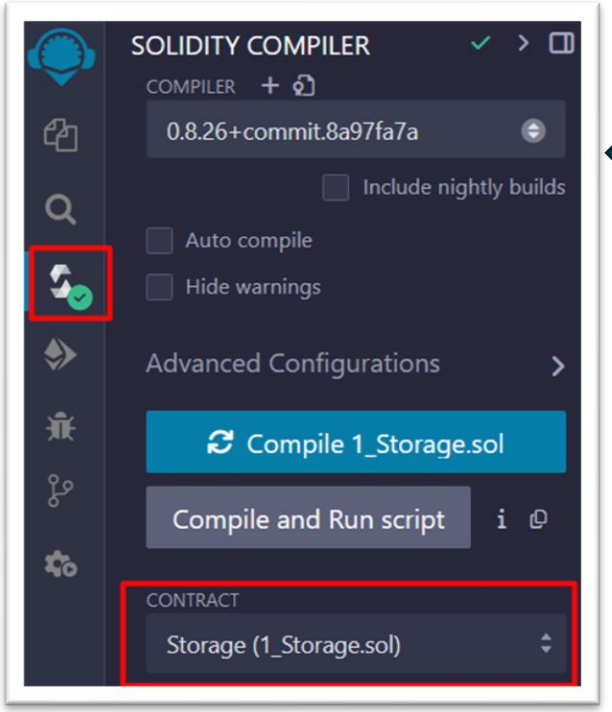
At the top (1) you will see the *compiler version*. In solidity we need to specify in which version we code at the top of the SC. You can click on the dropdown to see the different versions.

Bellow you will see the compile button (2). While having open the `Storage.sol` press this button and the SC will be compiled.

To compile a SC in Remix you can also `Ctrl+S`.

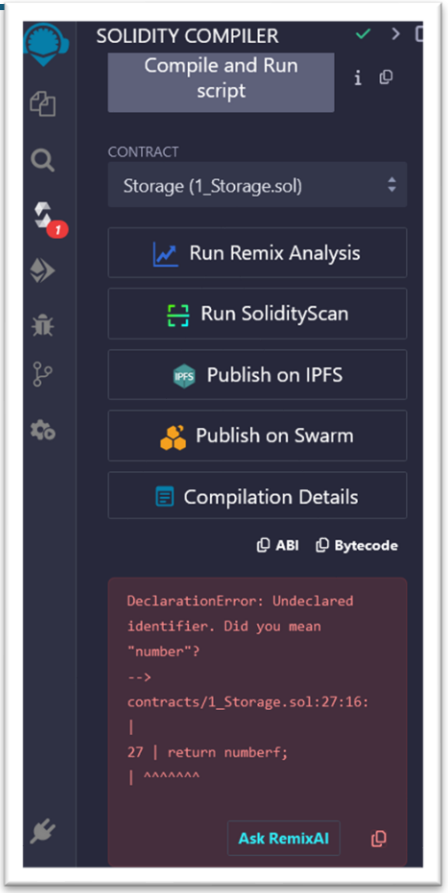


Compiling a Smart Contract



If you do not have any errors in your SC you will see a success mark in the compiler.

Else you will see errors or warnings by scrolling down on the compiler.



Deploying a Smart Contract

By clicking the button (1) in the sidebar, you will see the deployment area.

There you will see the following fields:

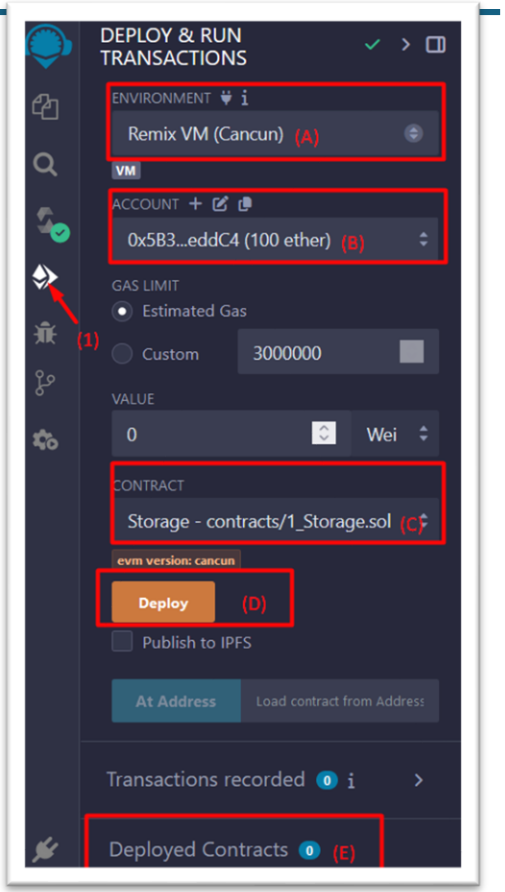
A) *Environment*: The blockchain you are deploying to.

B) *Account*: The accounts you have available.

C) *Contract*: The SC to be deployed.

D) The *Deploy* button.

E) *Deployed Contracts*: The SCs you deployed while using Remix (These are lost if you close it).



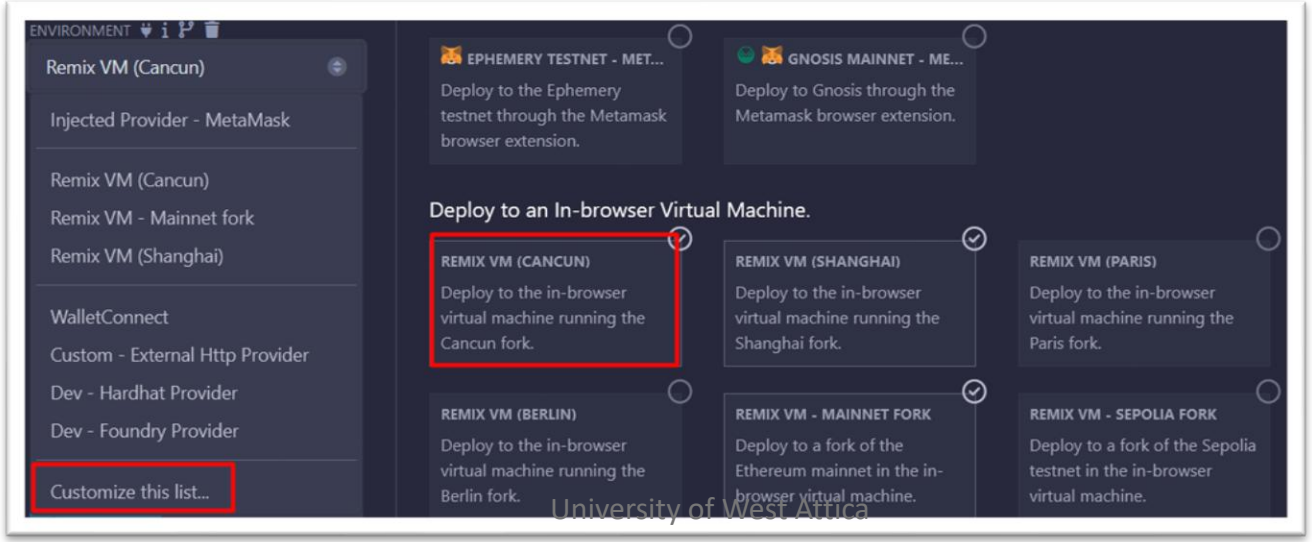
Deploying a Smart Contract

We will deploy the `Storage.sol` locally in a *VM*.

In the *Environment* you can see the *VM* you are currently on.

By clicking on the dropdown, you can see the different options you have.

We will use the Remix VM (Cancun) which is a local blockchain built just for the purpose of testing. There are no other users currently running on this blockchain since it local. If it is not in the options, click on *Customize this list...* in the dropdown and a window with all the different options will pop up.



Deploying a Smart Contract

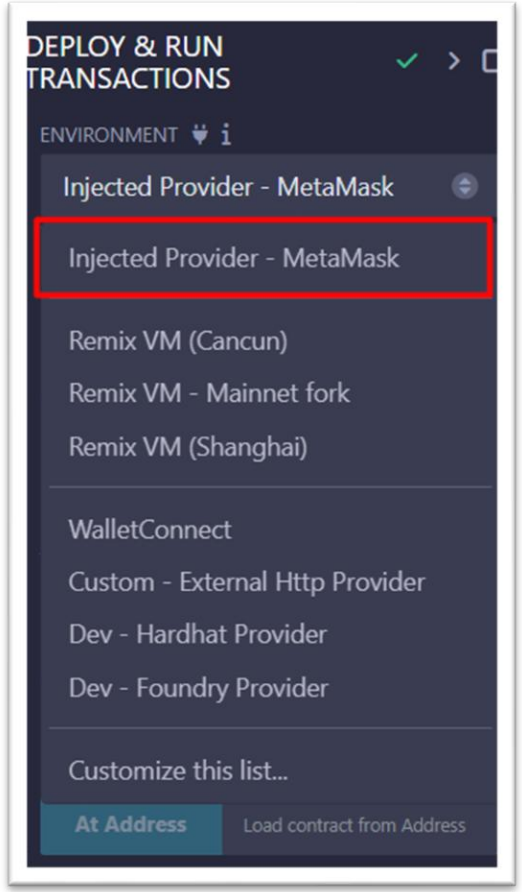
To deploy a SC to real network or a *testnet,* you have to use your own wallet (e.g. MetaMask) and on the environment dropdown click *Injected Provider – MetaMask* option.

You will be asked to unlock your account if it locked and connect it to Remix (no sensitive info will be used).

The environment will change to whatever is the current network on MetaMask.

The account will be the active account on MetaMask.

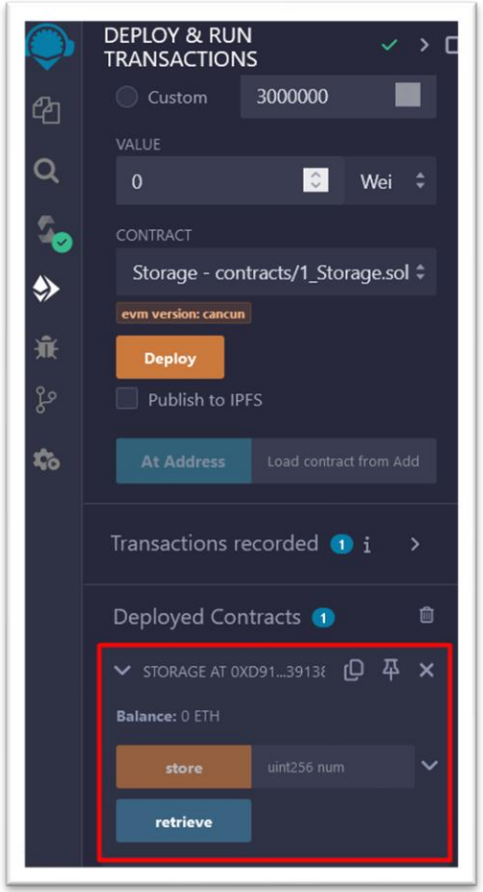
In order to deploy on a live network or testnet you will need to have some of its native currency (e.g. ETH for Ethereum).



Deploying a Smart Contract

In the *Account* you can see a dropdown list. These are test accounts with each one having its own address and all of them have 100 ETH. The one that is shown is the active account and it will be used to make the transaction of deploying the SC.

Click the *Deploy button* and you will see under the *Deployed Contracts* that the `Storage` SC has been deployed, and it has a few *functions* that we can use to interact with it.



Interacting with a Smart Contract

Under the *code editor*, that currently the SC is open, you can see the *console* and if you deployed the SC successfully, you would see some information about the SC's creation, similar to the image below.

We will talk more about them later.

Outro

Next, we will write our first SC, and we will interact with it through Remix IDE.