Centurion University
Shaping Lives... Empowering Communities...

School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : Dive into Ethereum – Clients and EVM

Objective/Aim:

To study and understand **Ethereum Clients** and the **Ethereum Virtual Machine (EVM)** – their roles, architecture, and how they execute smart contracts in the Ethereum blockchain ecosystem.

Apparatus/Software Used:

- **Ethereum Test Network (Sepolia)**
- **Remix IDE** (for writing and deploying smart contracts)
- **Metamask** (for wallet and transaction management)
- **Etherscan / Block Explorer** (to view contract deployment and execution details)

Theory:

Ethereum is a decentralized, open-source blockchain platform that supports **smart contracts**—self-executing code that runs on the blockchain without intermediaries.

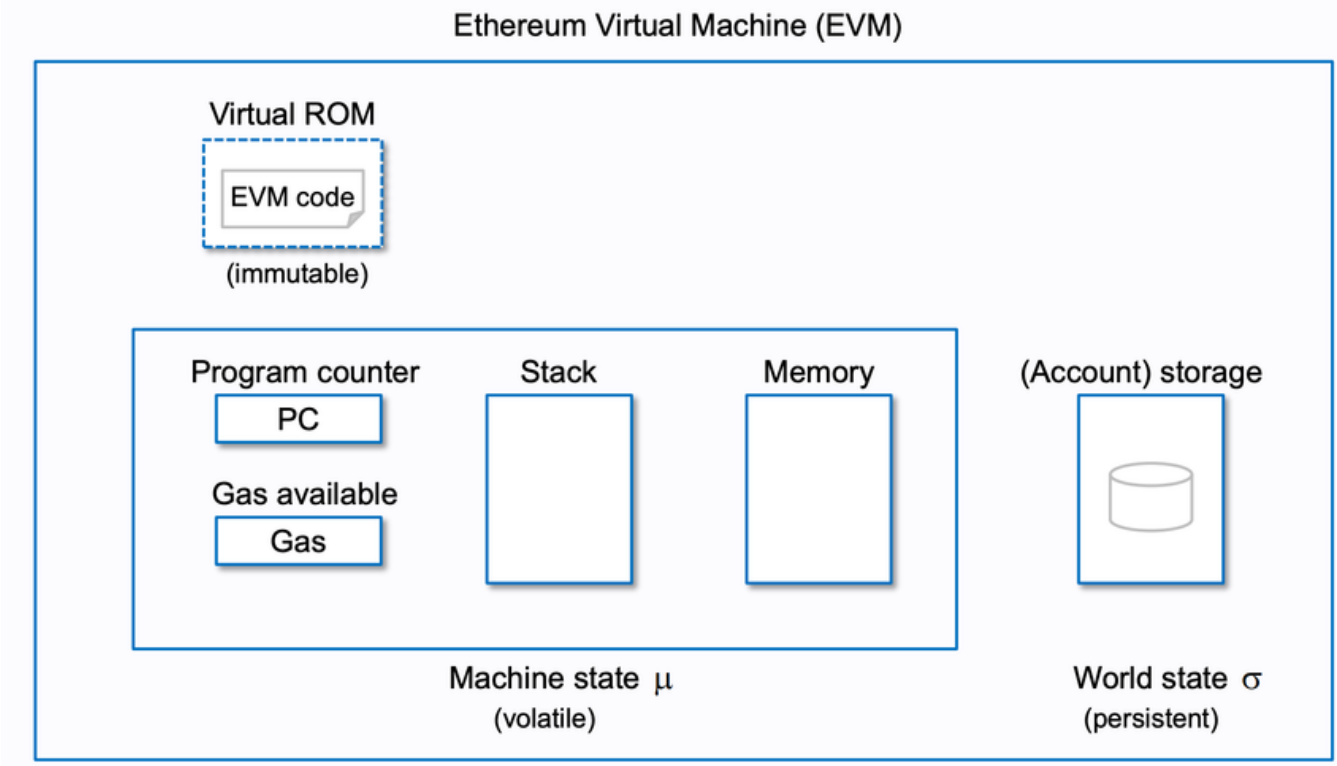
Ethereum Virtual Machine (EVM)

The **EVM** is the execution environment for Ethereum smart contracts.
It is a **stack-based virtual machine** that executes bytecode (compiled from Solidity or Vyper code).
Each node runs the EVM locally to verify transactions and update the blockchain state.

EVM Components:

1. **Stack:** Holds temporary values for computation.
2. **Memory:** Temporary, per-transaction storage.
3. **Storage:** Persistent storage linked to a contract address.
4. **Program Counter:** Tracks instruction execution.
5. **Gas:** Measures computation cost and prevents infinite loops.

Theory:



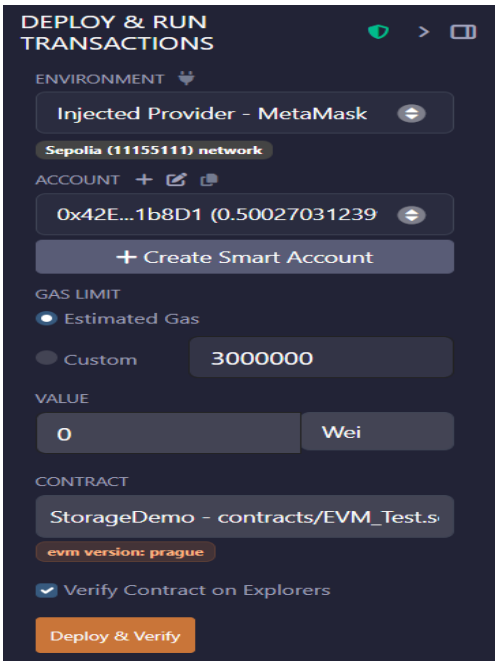
Procedure:

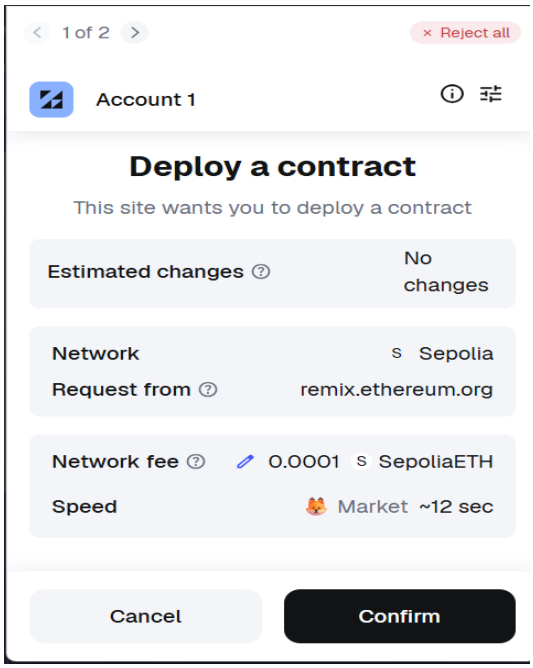
1.Open remix IDE and create a file eg. EVM_Test.sol and compile it.

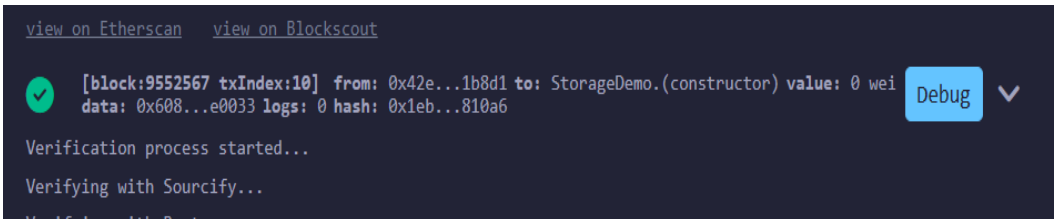
Compiled | Home | EVM_Test.sol

```
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.8.0;
3
4  contract StorageDemo {
5      uint public number;
6
7      function setNumber(uint _num) public {    22492 gas
8          number = _num;
9      }
10
11     function getNumber() public view returns (uint) {    2453 gas
12         return number;
13     }
14 }
```

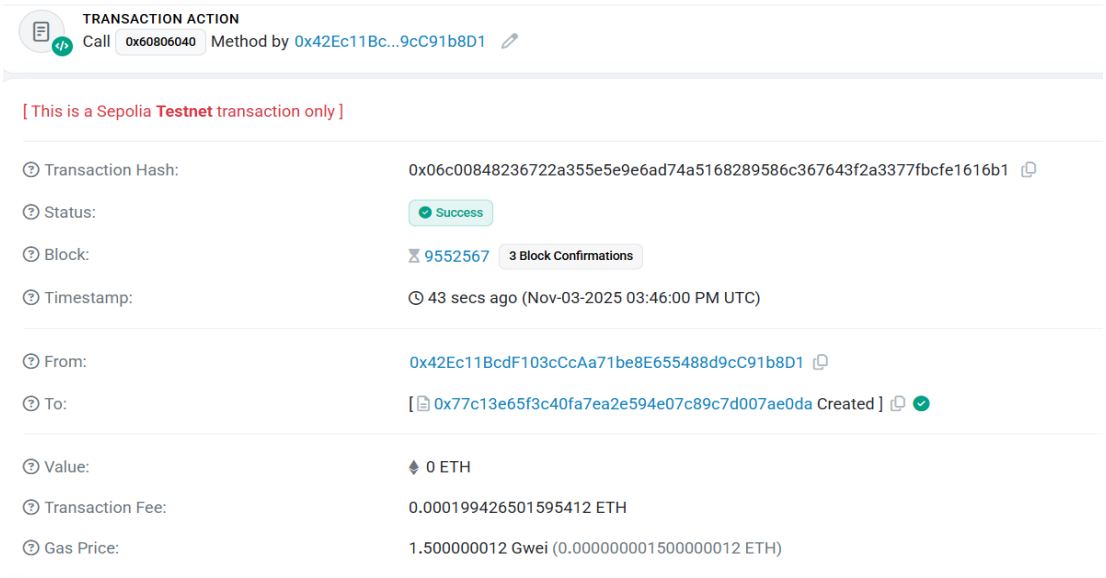
2. **Connect Metamask** to Remix (choose Sepolia testnet).
3. **Deploy** the contract using a test account.







4. After deployment, **view the transaction hash** on Etherscan.



5. Observe:
   - Gas used
   - Bytecode generated
   - Execution steps in Remix Debugger

Call the setNumber() and getNumber() functions to observe state changes.
Review the **EVM bytecode** generated in the compilation details.

Observation:

Through this experiment, we understood:

- The **role of Ethereum clients** in maintaining the decentralized network.
- How the **EVM** ensures deterministic and secure execution of smart contracts.
- That **smart contracts** are converted into low-level bytecode and executed identically across all nodes.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Signature of the Faculty:

Regn. No. :

**As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.