DOP: / /2023 DOS: / / 2023

Experiment No: 02

Aim: To deploy a smart contract on test network.

Theory:

smart contract:

A smart contract on a test network is a self-executing program or code that is deployed on a blockchain test network for testing and experimentation purposes. A test network is a separate blockchain network that mimics the main blockchain network but is designed for testing and experimentation purposes only.

In the context of smart contracts, test networks allow developers to deploy and test their smart contracts without incurring any costs associated with deploying to the main network. This enables developers to experiment with new features, test different scenarios, and identify and fix any bugs or issues before deploying to the main network.

Smart contracts on test networks are similar to those on the main network, with the key difference being that test network ether, which is used to pay for gas fees, has no real-world value. Test networks provide a safe and secure environment for developers to test their smart contracts and ensure that they are functioning correctly before they are deployed to the main network.

Overall, smart contracts on test networks provide a valuable tool for blockchain developers to test, refine, and improve their code before deploying it to the production network, helping to reduce the risk of errors or issues and ensure that smart contracts perform as expected.

Here are Steps deploy smart contract on test network

1. Choose a test network: There are several test networks available such as Rinkeby, Ropsten, Kovan, etc. Choose one that suits your requirements.

2. Set up your development environment: You need to have a development environment set up to deploy a smart contract on a test network. You can use tools like Truffle, Remix, or Hardhat to write and deploy your smart contract.

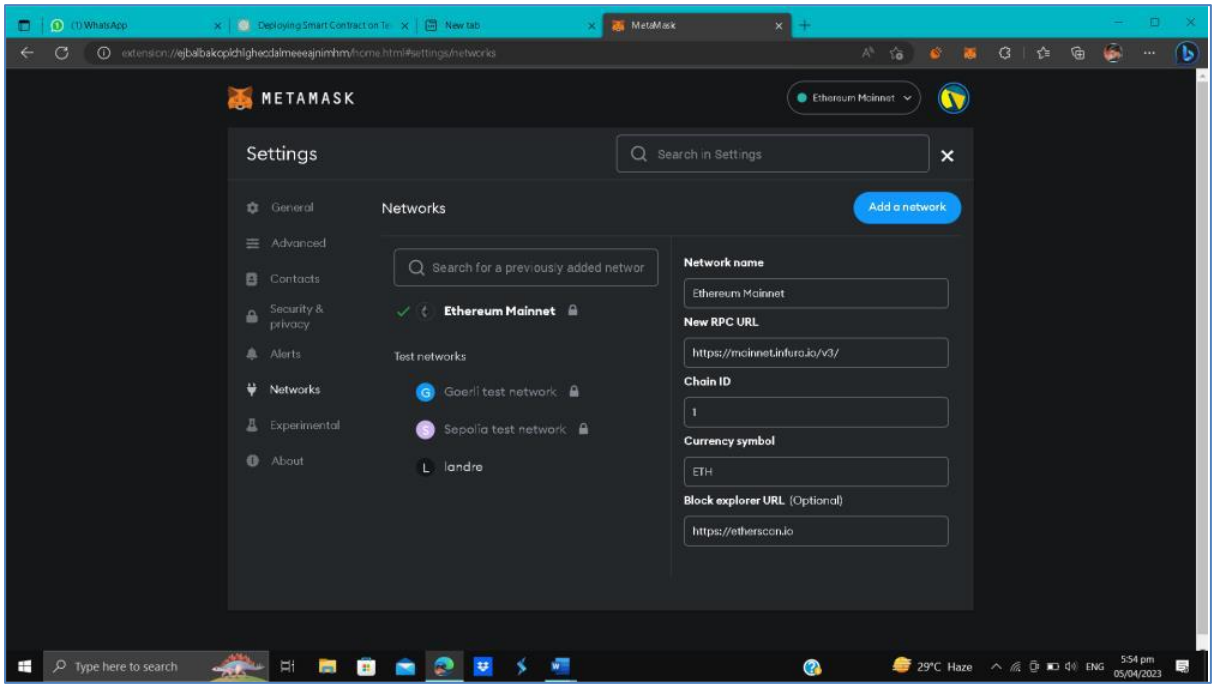
truffle init

3. Write your smart contract: Write your smart contract code in Solidity, which is the most popular programming language for writing smart contracts.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SimpleStorage {
    uint256 private _value;

    function set(uint256 value) public {
        _value = value;
    }

    function get() public view returns (uint256) {
        return _value;
    }
}
```

4. Compile your smart contract: Use a compiler such as Solc to compile your Solidity code into bytecode.

5. Connect to the test network: Use tools like Metamask or Geth to connect to the test network of your choice.



6. Fund your test network account: You need to have test network ether to pay for gas fees while deploying the smart contract.

7. Deploy the smart contract: Once you have connected to the test network and have funded your account, you can deploy the smart contract by sending a transaction to the network. This will create a new smart contract instance on the network.

```
Compiling your contracts...
===========================
> Compiling ./contracts/SimpleStorage.sol
> Compiling @openzeppelin/contracts/access/Ownable.sol
> Compiling @openzeppelin/contracts/token/ERC20/IERC20.sol
> Compiling @openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol
> Compiling @openzeppelin/contracts/token/ERC20/ERC20.sol
> Artifacts written to /path/to/your/truffle-project/build/contracts
> Compiled successfully using:
   - solc: 0.8.7+commit.e28d00a7.Emscripten.clang
```

8. Interact with the smart contract: You can interact with the smart contract by sending transactions to it and verifying that it behaves as expected.

npxTruffle migrate --reset

```
Starting migrations...
======================
> Network name:    'rinkeby'
> Network id:      4
> Block gas limit: 0x989680


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

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0x12aaf84c8610b7fa57cfe9340cb7197ec3c3f91d7c4e4f4dc8f221b
   > Blocks: 2            Seconds: 13
```

Conclusion:

Here we have successfully deploy a smart contract on test network without any error and properly implemented the program of smart contract.