Deploy Using Hardhat

Hardhat is a popular smart contract development frameworks. In this tutorial, we will be using Hardhat to deploy a simple Counter smart contract to the Custom Rollup Testnet. We will explore the basics of creating a Hardhat project with a sample contract and a script to deploy it. For the full instruction on how to use Hardhat, please refer to the official Hardhat documentation.

Create New Project

Start with creating an npm project by going to an empty folder, running npm init, and following its instructions. You can use another package manager, like yarn, but Hardhat recommends you use npm 7 or later, as it makes installing Hardhat plugins simpler.

Hardhat Smart Contract

To create the sample project, run npx hardhat init in your project folder:

- Press <ENTER> choose javascript, typescript or empty project
- Press <ENTER> to set the project root
- Press <ENTER> again to accept addition of .gitignore
- Press <ENTER> to install hardhat @nomicfoundation/hardhat-toolbox

Create deployer account

- Create the .env file in your project root folder and add the following line:

```
ACCOUNT_PRIVATE_KEY='my private key'
```

- Populate the .env file with your private key. You can get your private key from Metamask. See the section below on how to get your private key from Metamask.

How to get your Private Key in Metamask
warning
Do not commit your private key to a public repository!
Verify that your .gitignore file contains .env to prevent your private key from being committed to a public repository.

Configure Hardhat

- Open the hardhat.config.js file and paste the code below:
- Javascript
- Typescript

```
require("dotenv").config();
require("@nomicfoundation/hardhat-toolbox");

module.exports = {
  solidity: "0.8.19",
  paths: {
    artifacts: "./src",
  },
  networks: {
    opencampus: {
      url: `https://rpc.open-campus-codex.gelato.digital/`,
      accounts: [process.env.ACCOUNT_PRIVATE_KEY],
    },
  },
  etherscan: {
    apiKey: {
      opencampus: "your-etherscan-api-key",
    },
    customChains: [
      {
        network: "opencampus",
        chainId: 656476,
        urls: {
          apiURL: "https://opencampus-codex.blockscout.com/api",
          browserURL: "https://opencampus-codex.blockscout.com",
        },
      },
    ],
  },
};
```

Write Smart Contract

info

Note: The existing smart contract code that comes with the sample project is a Lock.sol contract. Feel free to delete it or leave it.

- Create a new file, in the contracts folder, named Counter.sol:

```
touch contracts/Counter.sol
```

- Copy the below code and paste it in the Counter.sol contract code:

```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

contract Counter {
uint256 currentCount = 0;

   function increment() public {
       currentCount = currentCount + 1;
   }
   function retrieve() public view returns (uint256){
       return currentCount;
   }
}
```

Create Deploy Script

- Delete the content of the scripts/deploy.js file and add the code below:

```
const hre = require("hardhat");

async function main() {
 const deployedContract = await hre.ethers.deployContract("Counter");
 await deployedContract.waitForDeployment();
 console.log(`Counter contract deployed to ${deployedContract.target}`);
}

main().catch((error) => {
 console.error(error);
 process.exitCode = 1;
});
```

Compile Contract

- Install dotenv package: npm install dotenv
- Compile your contract code (i.e., go back to the project root in the CLI),
- npx hardhat compile

Deploy Contract

- Run the deploy script:
- npx hardhat run scripts/deploy.js --network opencampus