Deploy using Hardhat

Depoly 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 theofficial Hardhat documentation .

Create New Project

Start with creating an npm project by going to an empty folder, runningnpm 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, runnpx hardhat init in your project folder:

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

Create deployer account

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

```

Copy 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* Click the vertical 3 dots in the upper-right corner of Metamask window * SelectAccount details * and then clickShow private key * Enter your Metamask password to reveal the private key * Copy the private key and paste it into the.env * file. * Verify that your .gitignore file contains.env to prevent your private key from being committed to a public repository. :::

Configure Hardhat

- Open thehardhat.config.js
- file and paste the code below:
-

```

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

module.exports={ solidity:"0.8.19", paths:{ artifacts:"./src", }, networks:{ raspberry:{ url:<your-rpc-url>, accounts: [process.env.ACCOUNT_PRIVATE_KEY], }, blueberry:{ url:<your-rpc-url>, accounts: [process.env.ACCOUNT_PRIVATE_KEY], }, blackberry:{ url:<your-rpc-url>, accounts: [process.env.ACCOUNT_PRIVATE_KEY], }, }, };
```

```

```

Copy import{ HardhatUserConfig }from"hardhat/config"; import"@nomicfoundation/hardhat-toolbox"; import*asdotenvfrom"dotenv";

dotenv.config({ path:__dirname+"/.env"}); constACCOUNT_PRIVATE_KEY=process.env.ACCOUNT_PRIVATE_KEY||""; console.log("PrivateKey set:",!!ACCOUNT_PRIVATE_KEY);

constconfig:HardhatUserConfig={ solidity:"0.8.19", paths:{ artifacts:"./src", }, networks:{ raspberry:{ url:<your-rpc-url>, accounts: [process.env.ACCOUNT_PRIVATE_KEY], }, ... }, };

exportdefaultconfig;

```

Write Smart Contract

Note: The existing smart contract code that comes with the sample project is aLock.sol contract. Feel free to delete it or leave it. * Create a new file, in the contracts folder, namedCounter.sol * : *

```

Copy touchcontracts/Counter.sol

```

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

```

Copy //SPDX-License-Identifier: MIT pragmasolidity^0.8.19;

contractCounter{ uint256currentCount=0;

functionincrement()public{ currentCount=currentCount+1; }

functionretrieve()publicviewreturns(uint256){ returncurrentCount; } }

```

Create Deploy Script

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

```

Copy consthre=require("hardhat");

asyncfunctionmain() { constdeployedContract=awaithre.ethers.deployContract("Counter"); awaitdeployedContract.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),
- ```
- Copy
- npxhardhatcompile
- ```
-

Deploy Contract
```

- Run the deploy script:
- ```
- Copy
- npxhardhatrunscripts/deploy.js--network
- ```
-

Previous Write A Contract Next Verify Contracts Last updated1 day ago On this page *Depoly using Hardhat * Create New Project * Hardhat Smart Contract * Create deployer account * Configure Hardhat * Write Smart Contract * Create Deploy Script * Compile Contract * Deploy Contract