Deploy on Celo with Hardhat

How to deploy a smart contract to Celo testnet, mainnet, or a local network using Hardhat .

Introduction to Hardhat

Hardhat is a development environment to compile, deploy, test, and debug your Ethereum or Celo software. It helps developers manage and automate the recurring tasks that are inherent to the process of building smart contracts and dApps, as well as easily introducing more functionality around this workflow. This means compiling, running, and testing smart contracts at the very core.

Prerequisites

To deploy on Celo using Hardhat, you should have Celo set up Celo in your local environment. If you prefer to deploy without a local environment, you can deploy using Remix or Replit.

- Using Windows
- Using Mac
- Using Replit

Create Hardhat Project

Choose one of the following items to prepare a dApp to deploy on Celo.

- Follow the installation instructions and quickstart
- to build and deploy your smart contract.

Update the hardhat.config.js file

Open hardhat.config.js in a text editor and replace its contents with this Celo configuration code . This code is similar to Hardhat settings with a few configuration updates needed to deploy to a Celo network. You will need to create a .env file in the project root directory and install dotenv with npm or yarn in order to read the process.env.MNEMONIC variable in the config file.

Connect to Local Network

Using Celo Ganache CLI creates test accounts at the localhost on port 7545. The private network setup connects to your localhost on this port and gives you access to your accounts on ganache-cli.

localhost :

{ url :

"http://127.0.0.1:7545" } , tip If you choose to Set up a Local Development Chain , your blockchain will also be hosted on a private network on localhost. This same configuration can be used to connect to the local development chain.

Connect to Testnet using Forno

Using Forno allows you to connect to the Celo test blockchain without running a local node. The testnet configuration uses Forno to connect you to the Celo Testnet (Alfajores) using HDWalletProvider and the mnemonic stored in your .env file.

alfajores :

{ url :

"https://alfajores-forno.celo-testnet.org" , accounts :

{ mnemonic : process . env . MNEMONIC , path :

"m/44'/52752'/0'/0" } , chainId :

44787 } note Celo uses a different account derivation path than Ethereum, so you have to specify "m/44'/52752'/0'/0" as the path.

Connect to Mainnet using Forno

Using Forno also allows you to connect to the Celo main blockchain without running a local node. The mainnet configuration

uses Forno to connect you to the Celo Mainnet using HDWalletProvider and the mnemonic stored in your.env file.

celo :

{ url :

"https://forno.celo.org" , accounts :

{ mnemonic : process . env . MNEMONIC , path :

"m/44'/52752'/0'/0" } , chainId :

42220 } tip [Forno](#) is a cLabs hosted node service for interacting with the Celo network. This allows you to connect to the Celo Blockchain without having to run your own node.

Deploy to Celo

Run the following command from your root project directory to deploy to Celo Alfajores testnet.

npx hardhat run scripts/sample-script.js --network alfajores ...or run this command to deploy to Celo mainnet.

npx hardhat run scripts/sample-script.js --network celo

View Contract Deployment

Copy yourcontract address from the terminal and navigate to the[block explorer](#) to search for your deployed contract. Switch between networks to find your contract using the dropdown by the search bar.

tip Learn more about building and deploying dApps using the[HardHat documentation](#) .

Verify Contracts on Celo

- [Using Celo Explorer](#)
- [Using Remix](#)
- [Using CeloScan](#)
- [Using Hardhat](#) [Edit this page](#) [Previous Deploy with Remix](#) [Next Celo Libraries & SDKs](#)