Quickstart Guide

Building smart contracts for the Canto EVM is the same as doing so for Ethereum or any other EVM-compatible chain, with the only difference being the network itself.

Smart contracts can be deployed using your Ethereum tooling of choice, including Hardhat ,Truffle ,Foundry ,Remix , and others. To get started, simply configure your environment to use Canto's RPC and chain ID.

Mainnet

RPC URL :https://canto.slingshot.finance/

Chain ID: 7700

Explorer: https://www.oklink.com/canto

Alternative RPC URLs * https://canto.neobase.one * https://canto.evm.chandrastation.com * https://jsonrpc.canto.nodestake.top/ * https://canto.dexvaults.com/ * Alternative Block Explorers Canto EVM:

- https://cantoscan.xyz/
- https://www.gacanto.com/
- https://canto.dex.guru/
-

Canto EVM and Native:

- https://www.mintscan.io/canto *

Testnet

RPC URL :https://canto-testnet.plexnode.wtf

Chain ID :7701

Explorer: https://testnet.tuber.build/ Alternative Testnet Explorer URL: https://canto-test.dex.guru/

Libraries

When building frontends or other applications that require on-chain data, you'll likely want to use a library to retrieve that data.

Needless to say, you can interact with the Canto EVM using most Ethereum libraries, such as ethers.js ,web3.js , and web3.py – just initialize a provider with the Canto RPC using your library of choice. For example:

```

Copy constethers=require('ethers')

constprovider=newethers.providers.JsonRpcProvider("https://canto.slingshot.finance")

asyncfunctionprintCurrentBlock() { console.log(awaitprovider.getBlockNumber()) }

printCurrentBlock()

```

Beginner's Guide

In case you're new to Solidity development, here are step-by-step instructions on how you can deploy your first contract on Canto using Remix:

Writing a Smart Contract

The first step in deploying a smart contract on Canto is writing the contract's source code in Solidity. To do this, create a new file in your Remix workspace:

Name the file as desired with the.sol extension and begin writing your code. You can also use source code from contracts that are already deployed on Ethereum.

For this example, let's copy and paste the following source code based on the official Solidity documentation :

```

Copy pragmasolidity0.8.17;

contractExample{ uintstoredData;

functionset(uintx)public{ storedData=x; }

functionget()publicviewreturns(uint) { returnstoredData; } }

```

Once your code is ready, hit Ctrl+S to compile your smart contract.

Deploying a Smart Contract

To deploy your smart contract, start by making sure you have MetaMask installed andconnected to Canto . Since this contract is for testing purposes, we'll connect to the Canto testnet.

Returning to Remix, navigate to the deployment tab, which is the fifth icon down on the vertical menu. Click on the environment drop-down box and selectInjected Provider - Metamask :

A MetaMask prompt will appear asking you to connect your wallet to Remix. As always, make sure you are comfortable with the permissions that are being requested and clickConnect .

Once you've connected your wallet, you'll see your account address underAccount . Make sure this is where you want to deploy your contract from and then hitDeploy.

If your smart contract uses constructor arguments, enter them in the field adjacent to theDeploy button before attempting to deploy your contract. After you hitDeploy , a MetaMask prompt will appear with a contract deployment transaction. Confirm the transaction.

As soon as the block is mined, your smart contract will be live on the Canto EVM. You can find the address of your smart contract at the bottom left of the Remix interface underDeployed Contracts , or by looking for the recipient of the contract deployment transaction in your wallet or on the block explorer.

Previous Overview Next NOTE, DEX, and Lending Market Last updated1 month ago On this page *