BNB Smart Chain Truffle Box¶

Table of contents generated with markdown-toc

This Truffle BNB Smart Chain box provides you with the boilerplate structure necessary to start coding on the BNB Smart Chain. For detailed information on how the BNB Smart Chain works, please see their documentationhere .

As a starting point, this box contains only theSimpleStorage Solidity contract. Including minimal code was a conscious decision as this box is meant to provide the initial building blocks needed to get to work on BNB Smart Chain without pushing developers to write any particular sort of application. With this box, you will be able to compile, migrate, and test Solidity code against several instances of BNB Smart Chain networks.

The BNB Smart Chain is fully compatible with the EVM. This means you will not need a new compiler to deploy Solidity contracts, and should be able to add your own Solidity contracts to this project. The main difference developers will encounter is in accessing and interacting with the BNB Smart Chain network.

Requirements¶

The BSC box has the following requirements:

- Node.js
- 10.x or later
- NPM
- version 5.2 or later
- Windows, Linux or MacOS

Helpful, but optional:

- AMetaMask
- account

Installation¶

```
truffle unbox bnb-chain/BSC-Truffle-Starter-Box
```

Setup¶

Using the env File¶

You will need at least one mnemonic to use with the network. The.dotenv npm package has been installed for you, and you will need to create a.env file for storing your mnemonic and any other needed private information.

The.env file is ignored by git in this project, to help protect your private data. In general, it is good security practice to avoid committing information about your private keys to github. Thetruffle-config.bsc.js file expects aMNEMONIC value to exist in.env for running migrations on the networks listed intruffle-config.bsc.js .

If you are unfamiliar with using.env for managing your mnemonics and other keys, the basic steps for doing so are below:

1. Usetouch .env
2. in the command line to create a.env
3. file at the root of your project.
4. Open the.env
5. file in your preferred IDE
6. Add the following, filling in your own mnemonic:

MNEMONIC="" 1. As you develop your project, you can put any other sensitive information in this file. You can access it from other files withrequire('dotenv').config() 2. and refer to the variable you need withprocess.env[''] 3. .

New Configuration File¶

A new configuration file exists in this project:truffle-config.bsc.js . This file contains a reference to the new file location of thecontracts_build_directory for BNB Smart Chain contracts and lists several networks that are running the BNB Smart Chain network instance (seebelow).

Please note, the classictruffle-config.js configuration file is included here as well, because you will eventually want to deploy contracts to on localhost for local development. All normal truffle commands (truffle compile ,truffle migrate , etc.) will use this config file and save built files tobuild/local-contracts . You can save Solidity contracts that you wish to deploy to Ethereum in thecontracts/local-dev folder.

New Directory Structure for Artifacts¶

When you compile or migrate, the resultingjson files will be atbuild/bsc-contracts/ . This is to distinguish them from contracts you build for any other network other than BSC. As we have included the appropriatecontracts_build_directory in each configuration file, Truffle will know which set of built files to reference!

BNB Smart Chain¶

Compiling¶

You do not need to add any new compilers or settings to compile your contracts for the BNB Smart Chain, as it is fully EVM compatible. Thetruffle-config.bsc.js configuration file indicates the contract and build paths for BSC-destined contracts.

If you are compiling contracts specifically for the BNB Smart Chain network, use the following command, which indicates the appropriate configuration file:

npm run compile:bsc If you would like to recompile previously compiled contracts, you can manually run this command withtruffle compile --config=truffle-config.bsc.js and add the--all flag.

Migrating¶

To migrate on the BNB Smart Chain network, runnpm run migrate:bsc --network=(bscTestnet | bscMainnet) (remember to choose a network from these options!).

As you can see, you have two BSC networks to choose from:

- bscTestnet
- : This is the BNB Smart Chain testnet.
- bscMainnet
- : This is the BNB Smart Chain mainnet. Caution! If you deploy to this network using a connected wallet, the fees are charged in mainnet BNB.

If you would like to migrate previously migrated contracts on the same network, you can runtruffle migrate --config truffle-config.bsc.js --network= (bscTestnet | bscMainnet) and add the--reset flag.

Paying for Migrations¶

To pay for your deployments, you will need to have an account with BNB available to spend. You will need your mnemomic phrase (saved in the.env file or through some other secure method). The first account generated by the seed needs to have the BNB you need to deploy.

If you do not have a wallet with funds to deploy, you will need to connect a wallet to at least one of the networks above. For testing, this means you will want to connect a wallet to theBSC Testnet network. We recommend usingMetaMask .

Documentation for how to set up MetaMask to configure custom network like BSc Testnet can be foundhere .

Follow the steps in the documentation above using the BNB Smart Chain RPC endpoints (https://docs.bnbchain.org/docs/rpc). ThechainId values are the same as those in thetruffle-config.bsc.js networks entries.

To get testnet BNB tokens use the official faucet .

Basic Commands¶

The code here will allow you to compile, migrate, and test your code on the BNB Smart Chain. The following commands can be run (more details on each can be found in the next section):

To compile:

npm run compile:bsc To migrate:

npm run migrate:bsc --network=(bscTestnet | bscMainnet) To test:

npm run test:bsc --network=(bscTestnet | bscMainnet)

Testing¶

In order to run the test currently in the boilerplate, use the following command:npm run test:bsc --network=(bscTestnet | bscMainnet) (remember to choose a network!). The current test file just has some boilerplate tests to get you started. You will likely want to add network-specific tests to ensure your contracts are behaving as expected.

Support¶

Support for this box is available via the Truffle community here or on our official Discord Channel .