

SKALE Network Box📦

- [Overview](#)
- [Requirements](#)
- [Install](#)
- [Setup](#)
- - [Get a SKALE Chain](#)
- - [Get a Wallet](#)
- - - [Fund your Wallet with sFUEL](#)
- - [Update Truffle Config](#)
- [Deploy to the SKALE Network](#)
- [Test](#)
- [SKALE Network Resource Links](#)
- [Support](#)

Overview

Welcome to the SKALE Network Box! This Truffle SKALE Network box provides you with all the basics necessary to kick-start scaling your application with the SKALE Network. You can find out more information about getting started with SKALE within the [Developer Getting Started Guide](#) .

This box comes with everything you need to start using smart contracts from a react app on the SKALE Network.

The SKALE Network is fully decentralized, and is supported by a validator community represented by over 48 validator orgs running over 150 nodes. Since the network is EVM compatible, all existing tools built for Ethereum will work directly within the SKALE Network as well. For example, the SKALE Network is fully compatible with the all Ethereum native wallets. Check out the SKALE Network Dev Portal for easy integration scripts fo connecting your favorite wallet, [here](#) .

For example, connecting MetaMask to SKALE in a seamless way can be seen in this document:

<https://docs.skale.network/develop/wallets/metamask>

All SKALE Chains contain the following features:

- Full EVM Support for Solidity Smart Contracts
- Interchain Messaging for managing Tokens (ETH, ERC20, ERC721, etc.)
- Decentralized Storage
- Integration Support for All Ethereum Tools
- Wallet Support for API and HSM Wallets

Lastly, the SKALE Network uses a [unique combination of several technologies](#) to achieve scalability, security, interoperability, and progressive decentralization:

Technology Impact Pooled Validation Proof-of-Stake [Scalable security model across validators and delegators](#) Hybrid Container Architecture [Agile allocation of on-demand composable compute resources across the network](#) Threshold Cryptography Supermajority signature signing with ABBA consensus supports Byzantine Fault Tolerance and resolves [data-availability](#) issues Trusted-Execution Environment [Fast block signing and multiple chain support using threshold cryptography](#) Asynchronous Binary Byzantine Agreement (ABBA) Consensus [Mathematically provable, fast-finality, leaderless, and Byzantine Fault Tolerant](#) Ethereum Network Public, open-source, and decentralized operation of the SKALE Network via SKALE Manager contracts

Requirements📦

The SKALE Network box has the following requirements:

- [Node.js](#)
- 12.x or later
- [NPM](#)
- version 5.2 or later
- Windows, Linux or MacOS

Installation📦

truffle unbox skalenetwork/truffle-skale-network-box

npm install

Setup

1) Get a SKALE Chain

The SKALE Network is a multichain platform, and there are many ways to get access to the SKALE Network testing environments. To obtain your SKALE Chain endpoint and Chain ID, check out the [SKALE Network getting started guide](#).

Alternatively, for faster testing please feel free to use the available [SDK Repo](#) and [Documentation](#).

2) Get a Wallet

The SKALE Network works with a variety of wallet solutions. If you already have a wallet and private key to use, you can skip this step and go to the next section: [Update Truffle Config](#).

You can pick and setup your favorite wallet from the documentation [here](#).

We recommend starting with MetaMask, and obtaining a wallet address and private key to use for testing from the [MetaMask wallet](#).

Fund your Wallet with sFUEL

SKALE Chains enable dApps to run scalable, cost-free transactions. This provides an effective way to scale your dApp not only for speed but also to scale economically. SKALE Chains do require sFUEL to conduct transactions – this sFUEL is a gas token with no economic value and provides a way for dApps to facilitate permissive access to SKALE Chains and protects against DDoS attacks.

There are a few ways developers can manage and allocate SKALE FUEL (sFUEL), depending on your setup.

You can learn more about sFUEL [here](#).

3) Update Truffle Config

You can add the SKALE Chain endpoint and private key directly into thetruffle-config.js file by replacing the following:

const privateKey = process.env.PRIVATE_KEY; const skale = process.env.SKALE_CHAIN; For example:

```
const privateKey = "0x0000000000000000000000000000000000000000000000000000000000000000"; const skale = "https://this-is-my-skale-chain.com:1234";
```

A more secure way would be to leave thetruffle-config.js as is, and create a.env file at the root directory of your project to hold onto your private key and SKALE Chain endpoint.

Your.env file should look like:

```
SKALE_CHAIN=https://this-is-my-skale-chain.com:1234
```

```
PRIVATE_KEY=0x0000000000000000000000000000000000000000000000000000000000000000 To learn more about using environment variables within a.env file, see:
```

<https://www.npmjs.com/package/dotenv>

Using a mnemonic

If you'd prefer to use a mnemonic (vs a private key), you can just omit thePRIVATE_KEY entry in your.env file and include aMNEMONIC entry instead. Note thatHDWalletProvider will use the first account that's generated from the mnemonic. For example:

```
SKALE_CHAIN=https://this-is-my-skale-chain.com:1234 MNEMONIC=test test test test test test test test test test test
```

Deploy to the SKALE Network

You can use Truffle to compile and migrate your smart contracts to the SKALE Network. The following command will compile and migrate the smart contract(s) to SKALE:

truffle migrate --network skale --compile-all --config=truffle-config.skale.js For more information on available Truffle commands, please see the Truffle documentation [here](#):

Testing¶

Truffle can run tests written in Solidity or JavaScript against your smart contracts. Note the command varies slightly if you're in or outside of the development console.

```
// inside the truffle development console. test --network skale --config=truffle-config.skale.js
```

```
// outside the truffle development console. truffle test --network skale --config=truffle-config.skale.js
```

Resources¶

SKALE Network Resources Links [-Website](#) [-Blog](#) [-Github](#) [-Whitepaper](#) [-Developer Docs](#)

SKALE Network Community Links [-Discord](#) [-Twitter](#) [-Telegram](#)

Support¶

Support for this box is available via the Truffle community [here](#) . In addition, SKALE Network support is available [here](#) .