Substrate

license GPL3/Apache2

pipeline manual

PRs welcome

Substrate Community & Support

Substrate is a next-generation framework for blockchain innovation 🚀 .

Getting Started

Head to `docs.substrate.io` and follow the installation instructions. Then try out one of the tutorials. Refer to the Docker instructions to quickly run Substrate, Substrate Node Template, Subkey, or to build a chain spec.

Community & Support

Join the highly active and supportive community on the Substrate Stack Exchange to ask questions about use and problems you run into using this software. Please do report bugs and issues here for anything you suspect requires action in the source.

Contributions & Code of Conduct

Please follow the contributions guidelines as outlined in `docs/contributor/CONTRIBUTING.md` . In all communications and contributions, this project follows the Contributor Covenant Code of Conduct.

Security

The security policy and procedures can be found in `docs/contributor/SECURITY.md` .

License

- Substrate Primitives (`sp-*`), Frame (`frame-*`) and the pallets (`pallets-*`), binaries (`/bin`) and all other utilities are licensed under Apache 2.0.
- Substrate Client (`/client/* / sc-*`) is licensed under GPL v3.0 with a classpath linking exception.

The reason for the split-licensing is to ensure that for the vast majority of teams using Substrate to create feature-chains, then all changes can be made entirely in Apache2-licensed code, allowing teams full freedom over what and how they release and giving licensing clarity to commercial teams.

In the interests of the community, we require any deeper improvements made to Substrate's core logic (e.g. Substrate's internal consensus, crypto or database code) to be contributed back so everyone can benefit.

Substrate Node Template

Installation Guide

Prerequisites

Ensure you have the following installed on your system:

- Rust
- Cargo (comes with Rust installation)
- Git

Cloning the Repository

First, clone the repository to your local machine:

```
git clone https://github.com/nhitranbtc/substrate-node-template.git
cd substrate-node-template
```

Building the Project

To build the project, run the following commands:

```
# Check the project
make check

# Build the project
make build

# Build the project in release mode
make build-release
```

Running the Project

To run the project, use the following commands:

```
# Run the project in development mode
make run

# Run the project in release mode
make release
```

Add Trusted Nodes

Tutorial Objectives

By completing this tutorial, you will accomplish the following objectives:

- Generate key pairs for use as a network authority.
- Create a custom chain specification file.
- Launch a private two-node blockchain network.

Generate Session Keys

Sr25519 Keys (BABE)

```
# Generate keys with mnemonic
./target/release/substrate-node-template key generate \
  --scheme Sr25519 \
  --password-filename ./bin/substrate-node/res/password.txt
```

Ed25519 Keys (GRANDPA)

```
# Generate keys with same mnemonic
./target/release/substrate-node-template key inspect \
  --scheme Ed25519 \
  --password-filename ./bin/substrate-node/res/password.txt \
  "<your-mnemonic-phrase>"
```

Start Node01

```
./target/release/substrate-node-template \
  --base-path /tmp/node01 \
  --chain=./bin/substrate-node/res/customSpecRaw.json \
  --port 30333 \
  --rpc-port 9944 \
  --validator \
  --rpc-methods Unsafe \
  --name MyNode01 \
  --node-key=c12b6d18942f5ee8528c8e2baf4e147b5c5c18710926ea492d09cbd9f6c9f82a
```

Start Node02

```
./target/release/substrate-node-template \
  --base-path /tmp/node02 \
  --chain=./bin/substrate-node/res/customSpecRaw.json \
  --port 30334 \
  --rpc-port 9945 \
  --validator \
  --rpc-methods Unsafe \
  --name MyNode02 \
  --node-key=6ce3be907dbcabf20a9a5a60a712b4256a54196000a8ed4050d352bc113f8c58 \
  --bootnodes /ip4/127.0.0.1/tcp/30333/p2p/12D3KooWBmAwcd4PJNJvfV89HwE48nwkRmAgo8Vy3uQE
```

Insert Keys for Node01

```
# Insert BABE key
./target/release/substrate-node-template key insert \
  --base-path /tmp/node01 \
  --chain=./bin/substrate-node/res/customSpecRaw.json \
  --password-filename ./bin/substrate-node/res/password.txt \
  --scheme Sr25519 \
  --key-type babe \
  --suri "<your-secret-seed>"

# Insert GRANDPA key
./target/release/substrate-node-template key insert \
  --base-path /tmp/node01 \
  --chain=./bin/substrate-node/res/customSpecRaw.json \
  --password-filename ./bin/substrate-node/res/password.txt \
  --scheme Ed25519 \
  --key-type gran \
  --suri "<your-secret-seed>"
```

Restart Node01

Insert Keys for Node02

```
# Insert BABE key
./target/release/substrate-node-template key insert \
  --base-path /tmp/node02 \
  --chain=./bin/substrate-node/res/customSpecRaw.json \
  --password-filename ./bin/substrate-node/res/password.txt \
  --scheme Sr25519 \
  --key-type babe \
  --suri "<your-secret-seed>"

# Insert GRANDPA key
./target/release/substrate-node-template key insert \
  --base-path /tmp/node02 \
  --chain=./bin/substrate-node/res/customSpecRaw.json \
  --password-filename ./bin/substrate-node/res/password.txt \
  --scheme Ed25519 \
  --key-type gran \
  --suri "<your-secret-seed>"
```

Restart Node02

Directory Structure

```
/tmp/
├── node01/
│   ├── chains/
│   │   └── local_testnet/
│   │       └── keystore/
│   │
│   └── keystore/
│
└── node02/
  ├── chains/
  │   └── local_testnet/
  │       └── keystore/
  │
  └── keystore/
```