**How to setup the DEVNET using Zombienet**

**Zombienet Configuration**

Note:

Use Rust Toolchain version 1.81.

Documentation: https://docs.google.com/document/d/1tFlRyX1f4aOoddvnN2EH-8wguhYKSOtpA3A0PiFCnzw/edit#heading=h.rzd4bkjyvplk

To install this follow this command:

sudo apt update

sudo apt install --assume-yes git clang curl libssl-dev llvm libudev-dev make protobuf-compiler

curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh

source ~/.cargo/env

rustup update stable

rustup install 1.81

rustup default 1.81

rustup update nightly

rustup target add wasm32-unknown-unknown --toolchain nightly

rustup component add rust-src --toolchain 1.81

rustup target add wasm32-unknown-unknown --toolchain 1.81

rustup component add rust-src --toolchain 1.81-x86\_64-unknown-linux-gnu

rustc --version

**Build the xode-node**

Clone the node repository by running the following command:

Command: git clone https://github.com/Blockspace-Corporation/xode-blockchain

**Change to the root of the directory**

Command: cd xode-blockchain

**Create a new branch** to save your work by running a command similar to the following:

Command: git switch -c my-learning-branch-yyyy-mm-dd

**Compile the node**

Command: cargo build –release

**Verify that your node is ready**

Command: ./target/release/xode-node --help

**Go to zombinet directory**

Command: cd zombienet

**Download the appropriate binaries for your operating system.**

Polkadot binaries: https://github.com/paritytech/polkadot/releases

Command:

wget https://github.com/paritytech/polkadot/releases/download/v1.0.0/polkadot

Zombienet binaries: https://github.com/paritytech/zombienet/releases

Command:

wget https://github.com/paritytech/zombienet/releases/download/v1.3.109/zombienet-linux-x64

**Run chmod +x all binaries and script**

Command: chmod +x polkadot

Command: chmod +x zombienet-linux-x64

**Create a config file to spawn ephemeral networks**

Config file is already created in the repository name xode-para-config.toml

If you want to create your own config file, enter this command:

Command: touch config.toml

This is an example of config file

[settings]

timeout = 1000

node\_verifier = "None"

[relaychain]

default\_command = "./polkadot"

default\_args = [ "-l=parachain=debug,xcm=trace" ]

chain = "rococo"

[[relaychain.nodes]]

name = "alice"

validator = true

[[relaychain.nodes]]

name = "bob"

validator = true

[[relaychain.nodes]]

name = "charlie"

validator = true

[[relaychain.nodes]]

name = "dave"

validator = true

[[parachains]]

id = 1000

cumulus\_based = true

[[parachains.collators]]

name = "xode-collator"

command = "./xode-node"

args = [ "-l=xcm=trace" ]

**Save the file and run the zombienet using this command:**

Command: ./zombienet-launch.sh or ./zombienet-linux-x64 spawn xode-para-config.toml -p native

**Open the appropriate link in polkadotJS**

**Xode Blockchain Template - Run in Zombienet and Run a Separate Xode Node**

**Run Xode-Blockchain-template using Zombienet**

**Step 1: Clone the Xode Blockchain Template Repository.**

git clone git@github.com:Xode-DAO/xode-blockchain-template.git

**Step 2: Compile the code.**

cd xode-blockchain-template

cargo build –release –package xode-node

**Step 3: Go to zombienet directory**

cd zombienet

**Step 4: Download Polkadot and Zombienet Binaries**

Polkadot: wget https://github.com/paritytech/polkadot/releases/download/v1.0.0/polkadot

Zombienet: wget https://github.com/paritytech/zombienet/releases/download/v1.3.109/zombienet-linux-x64

**Step 5: Make the binaries executable**

chmod +x polkadot zombienet-linux-x64

**Step 6: Modify zombienet.toml**

sudo nano zombienet.toml

Change default\_command = "../../polkadot/target/release/polkadot" under [relaychain] to default\_command = "./polkadot"

**Save the file Ctrl X then Y Enter.**

**Step 7: Modify the zombienet-launch.sh**

sudo nano zombienet-launch.sh

Change ./bin/zombienet-macos-x64 to ./zombienet-linux-x64

Save the file Ctrl X then Y Enter.

**Step 8: Run the Zombienet**

./zombienet-launch.sh

**Step 9: Go to PolkadotJS UI**

Click the link provided by the zombienet.

It will redirect to PolkadotJS UI

Check if there is xode staking pallet.

Go to Developer > Extrinsic > xodeStaking

**Runtime Upgrade: Version 1 to Version 2**

**Step 1: Identify the current version**.

Located at the top left of PolkadotJS. The current version is 1.

**Step 2: Modify the runtime code.**

Go to runtime/src/lib.rs and locate the pub const VERSION.

Change the spec\_version to 2.

Save the file.

**Step 3: Build the Code**

cargo build –release –package xode-node

**Step 4: Upload the new WASM**

Go back to PolkadotJS. Go to Developer > Extrinsics > sudo > sudoUncheckWeight > system > setCode . Upload the file located at xode-blockchain-template/target/ release/wbuild/xode-runtime/xode\_runtime.compact.compressed.wasm

Submit Transaction and Sign and Submit.

**Step 5: Check the Current version**

Check if the current version changes from 1 to 2

Runtime Upgraded Successfully.

**Run a Separate Xode Node with session key**

**Step 1: Check the command of collator in zombienet.**

Go to /tmp/zombie3424321312/dave.yaml. Copy the command from the collator of zombienet.

**Step 2: Generate node key**

Go to terminal and go to zombienet directory:

cd zombienet

./polkadot key generate-node-key

**Step 3: Run Collator in Terminal**

Paste the command from the collator in the terminal and modify the name, base path, rpc-port and the node key. Paste the generated node key here.

**Step 4: Generate Session Keys**

The session key needs to be set for a collator to start producing blocks. Generate a session key by sending an RPC call to the http endpoint of the parachain with the author\_rotateKeys method: Go to a new terminal and follow this:

curl http://127.0.0.1:7777 -H \

"Content-Type:application/json;charset=utf-8" -d \

'{

"jsonrpc":"2.0",

"id":1,

"method":"author\_rotateKeys",

"params": []

}'

An exemplary result: {"jsonrpc":"2.0","result":"0x56066a71efc51e4a6f0f838cac959a08b238e22d478bd5dc0cdc2ac5b40d2e66","id":1}

We use http://127.0.0.1:777 because 7777 is the rpc port of the new collator.

**Step 5: Bind collator account to the generated session key.**

Go to the PolkadotJS where the new Collator is running.

Go to Developers >Extrinsics > session > setKey with your associated account. There you will put the pubkey you just obtained, as “proof” you’ll simply put “0x” (0x is prefix for hex content and there is no validation implemented for the proof field, so we simply send empty content)

Click Submit Transaction and Sign and Submit.

**Step 6: Check if Ferdie has Session keys.**

Go to Developers > chain state > session > nextKeys. Push the + on the right and you must discover your key in the field at the bottom right of the page.

**Step 7: Add candidate to Xode Staking**

Go to Developers > Extrinsics > xodeStaking > stake(NewCandidate)

Click Submit Transaction Then Sign and Submit

**Step 8: Check Xode Staking Candidate**

Go to Developers > Chain State > xodeStaking > candidsates

So you can see that the session key has been added to xodeStakingCandidate and in Authority.

**Step 9: Check if it produces blocks.**

Go to Network > Explorer

No Ferdie has been displayed.

**Run a Separate Xode Node with node-key**

**Step 1: Check the command of collator in zombienet.**

Go to /tmp/zombie3424321312/dave.yaml. Copy the command from the collator of zombienet.

**Step 2: Generate node key**

Go to terminal and go to where the subkey binary is:

cd polkadot-sdk

./target/release/subkey inspect //Ferdie

**Step 3: Run Collator in Terminal**

Paste the command from the collator in the terminal and modify the name, base path, rpc-port and the node key. Paste the generated node key here.

**Step 4: Add candidate to Xode Staking**

Go to the PolkadotJS where the separate node is running. Go to Developers > Extrinsics > xodeStaking > stake(NewCandidate)

Paste the node key here.

Click Submit Transaction Then Sign and Submit

**Step 8: Check Xode Staking Candidate**

Go to Developers > Chain State > xodeStaking > candidates

So you can see that the session key has been added to xodeStakingCandidate and in Authority.

**Step 9: Check if it produces blocks.**

Go to Network > Explorer

No Ferdie has been displayed.