DFX Project Setup Guide

This document provides two approaches for setting up your development environment. One approach uses pre-built binaries (the ones I built and tested on my machine, would only work if you had the same Ubuntu, wsl and go version), while the other rebuilds the project binaries from source code(The code that I modified, if you build the binaries on your own pc using “make geth” command, then then there is no strict version matching of wsl and Ubuntu)

# Approach 1: Use Pre-built Binaries (Recommended)

Step 1: System Requirements (steps for installation are at the end)

- Ubuntu 22.04.5 LTS:   
- Windows Subsystem for Linux 2 (WSL2):   
- Go Programming Language (1.23.1)  
- Foundry (forge & cast): <https://book.getfoundry.sh/getting-started/installation>

Step 2: Verify Installed Tools

Run the following commands to check your environment, first open powershell.

wsl #starts the Ubuntu terminal  
uname -a # Check Linux kernel  
lsb\_release -a # Verify Ubuntu version  
go version # Verify Go installation

Step 3: Running the Project

Navigate to the ethereum-pos-testnet folder

chmod +x testnet.sh

./testnet.sh

# Approach 2: Rebuild from Source Code (DFX Folder)

Step 1: Prerequisites

- curl: https://curl.se/download.html  
- jq: https://jqlang.github.io/jq/download/  
- Foundry (forge & cast): https://book.getfoundry.sh/getting-started/installation  
- Go: (Follow steps from approach 1)

Note: you could use cli commands for these if you prefer

### Install Dependencies (CLI)

sudo apt update && sudo apt install -y curl jq build-essential

curl -L https://foundry.paradigm.xyz | bash

foundryup

### Step 3: Build the Project

Move into the source folder:

cd project/DFX

make all

This compiles all required binaries from source.

### Step 4: Run the Testnet

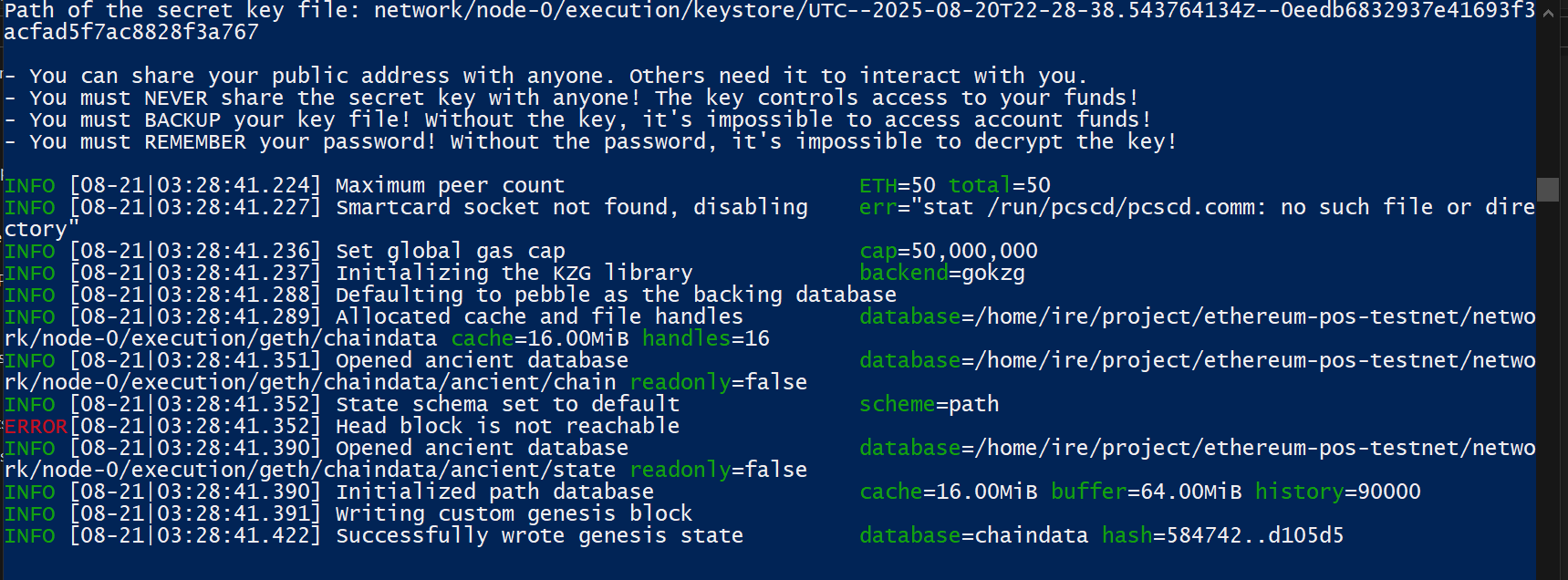
Once built, you can read the contents within testnet.sh to understand it better. Switch to the testnet folder and launch:

cd ../ethereum-pos-testnet

chmod +x testnet.sh

./testnet.sh

If all works well, it should show:



The head block error will go away as more blocks are mined. That’s supposed to happen.

Same goes for “nil block finalized”.

## Steps to Install WSL2 + Ubuntu 22.04.5 LTS for windows:

### 1. Enable WSL & Virtual Machine Platform

Open **PowerShell as Administrator** and run:

wsl --install

This installs **WSL2**, the **kernel**, and defaults to Ubuntu (latest LTS).  
If WSL is already installed, make sure it’s updated:

wsl --update

### 2. Set WSL2 as Default

wsl --set-default-version 2

### 3. Install Ubuntu 22.04.5 LTS

Check available distros:

wsl --list --online

Install Ubuntu 22.04:

wsl --install -d Ubuntu-22.04

(this will pull the latest 22.04.x — currently ***22.04.5 LTS***)

### 4. Launch Ubuntu

After install finishes, launch it:

“wsl -d Ubuntu-22.04” or simply “wsl”

On first launch:

* It will set up the filesystem
* Ask you to create a **UNIX username + password**

### 5. Update Ubuntu packages

Inside Ubuntu:

sudo apt update && sudo apt upgrade -y

### 6. Verify versions

Check WSL version (from PowerShell):

wsl -l -v

You should see:

Ubuntu-22.04 Running 2

Check Ubuntu release (inside WSL):

lsb\_release -a

Expected:

Description: Ubuntu 22.04.5 LTS

**Go Installation Guide (WSL2 - Ubuntu)**

1. Remove old Go (if installed) sudo rm -rf /usr/local/go

2. Download Go 1.23.1 tarball wget https://go.dev/dl/go1.23.1.linux-amd64.tar.gz (If your system is ARM64, replace `amd64` with `arm64`)

3. Extract it to /usr/local sudo tar -C /usr/local -xzf go1.23.1.linux-amd64.tar.gz

4. Add Go to PATH nano ~/.bashrc

5. export PATH=$PATH:/usr/local/go/bin

6. source ~/.bashrc