Deploy a ZK Stack Validium with Avail DA

Introduction

In this guide, we will deploy a ZK Stack validium powered by Avail DA for secure, cost-efficient, and verifiable data availability. By following along, at the end of this guide, you will have a version of a ZK Stack chain running on your machine onlocalhost .

- Learn more about the ZK Stack(opens in a new tab)

In this guide, you will go over the following:

- Setting up your developer environment
- Installing & running the ZK Stack

Setting up your developer environment

Ensure you have installed the following prerequisites.

Software Version Node.js (opens in a new tab) Latest LTS Version Git (opens in a new tab) OS Default Docker (opens in a new tab) Latest Docker Compose (opens in a new tab) Latest Yarn (opens in a new tab) v1.22.19 cargo-nextest (opens in a new tab) Latest sqlx-cli (opens in a new tab) Latest Foundry (opens in a new tab) Latest Postgres (opens in a new tab) Latest After executing the following commands you will have installed all the prerequisites needed and also have cloned thezksync-era repository.

Installation commands are based onUbuntu 20.04 LTS :

The below installation commands are taken fromsetup-dev.md(opens in a new tab) from thezksync-era repository. For any more troubleshooting, please refer to the original repository.

For VMs only! They don't have SSH keys, so we override SSH with HTTPS

```
git
config
--global
url. "https://github.com/" .insteadOf
git@github.com: git
config
--global
url. "https://" .insteadOf
git://
```

Rust

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

sh

NVM

curl

-o-

https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh

|

bash

All necessary stuff

sudo

apt-get

update sudo

apt-get

install

-y

build-essential

pkg-config

cmake

clang

lldb

lld

libssl-dev

libpq-dev

apt-transport-https

ca-certificates

curl

software-properties-common

Install docker

curl

-fsSL

https://download.docker.com/linux/ubuntu/gpg

|

sudo

apt-key

add

- sudo

add-apt-repository

"deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable" sudo

apt

install

docker-ce sudo

usermod

-aG

docker {USER}

Start docker.

sudo

systemctl

start

docker

You might need to re-connect (due to usermod change).

Node & yarn

nvm

install

20

Important: there will be a note in the output to load

new paths in your local session, either run it or reload the terminal.

npm

install

-g

yarn yarn

set

version

1.22 .19

For running unit tests

cargo

install

cargo-nextest

SQL tools

cargo

install

sqlx-cli

--version

0.8 .1

Foundry ZKsync

curl

-L

https://raw.githubusercontent.com/matter-labs/foundry-zksync/main/install-foundry-zksync

|

bash foundryup-zksync

Non CUDA (GPU) setup, can be skipped if the machine has a CUDA installed for provers

Don't do that if you intend to run provers on your machine. Check the prover docs for a setup instead.

echo

"export ZKSYNC_USE_CUDA_STUBS=true"

~/.bashrc

You will need to reload your *rc file here

Installing & running the ZK Stack

1. Install ZK Stack CLI

ZK Stack CLI(opens in a new tab) is a toolkit that facilitates the creation and management of ZK Stacks. To installzkstack , run the following command:

curl

-L

https://raw.githubusercontent.com/matter-labs/zksync-era/main/zkstack_cli/zkstackup/install

|

bash source

~/.bashrc zkstackup

2. Clone the zksync-era repository from github

git

clone

git@github.com:matter-labs/zksync-era.git cd

zksync-era * Install submodules:

git

submodule

update

--init

--recursive

3. Running the containers

- We can start running the containers and ecosystem by first cleaning the docker containers and then running the following command:

zkstack

dev

clean

all Now, we are ready to start the DB and Reth containers usingzkstack . The great thing about this is thatzkstack handles everything on its own. If you encounter any errors, just make sure your ports are not busy and theobservability feature is disabled during container setting up.

zkstack

containers

4. Initialize Elastic Chain ecosystem

- We need to create an Elastic Chain ecosystem now. We will choose all the default options.

zkstack

ecosystem

init * Once we have successfully initialized an elastic chain ecosystem, we can create new chains and can choose to create a Validium when prompted. For simplicity, we will choose the default options.

zkstack

chain

create * Once we have created the chain, we have to finally initialize it and deploy the contracts of the ZK chain:

zkstack

chain

init

5. Configure Avail DA

BEFORE YOU BEGIN

Before configuring Avail DA, complete these essential steps:

1. Create an Avail Account (opens in a new tab)
2. : Set up your account to interact with the Avail network
3. Get Testnet Tokens (opens in a new tab)
4. : Obtain AVAIL tokens on the Turing testnet for transactions
5. Create an AppID (opens in a new tab)
6. : Required to identify your application on the Avail network After completing these prerequisites, configure your chain to use Avail as the data availability layer by updating two configuration files with the appropriate settings.

Please choose between theFull Client orGas Relay configuration below.

Full Client

Using the full client requires you tocreate an AppID(opens in a new tab)and maintain the Avail balance in your designated data submission account.

1. Add the following to your chain's general config:zksync-era/chainsCHAIN_NAME/configs/general.yaml

Turing Mainnet da_client : avail : bridge_api_url :

https://turing-bridge-api.avail.so timeout_ms :

7200000 full_client : api_node_url :

wss://turing-rpc.avail.so/ws app_id :

YOUR_APP_ID 1. Add the following to your chain's secrets config:zksync-era/chainsCHAIN_NAME/configs/secrets.yaml

da : avail : seed_phrase :

YOUR_SEED_PHRASE

Gas Relay

The gas relay API is inprivate beta . Using the gas relay allows you to post data to Avail using a relayer and pay fees in a different token (only if supported) without needing to maintain a balance. It is recommended tocreate an AppID(opens in a new tab) to use within the gas relay service. Please contact the Avail team if you want access to the gas relay API.

1. Add the following to your chain's general config:zksync-era/chainsCHAIN_NAME/configs/general.yaml

Turing Mainnet da_client : avail : bridge_api_url :

https://turing-bridge-api.avail.so timeout_ms :

7200000 gas_relay : gas_relay_api_url :

https://turing.turbo-api.availproject.org max_retries :

3 1. Add the following to your chain's secrets config:zksync-era/chainsCHAIN_NAME/configs/secrets.yaml

da : avail : gas_relay_api_key :

YOUR_API_KEY

6. Run ZK chain

Start the server which will be running the ZK chain. By default, it will be running onlocalhost:3150 .

zkstack

server

--chain CHAIN_NAME Your ZK chain is now running and you can start using it. For any troubleshooting or issues, please refer to the original repository which you can findhere(opens in a new tab) .

ZKsync's ZK Stack Sovereign Rollups