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 prequisites needed and also have cloned thezksync-era repository.

Installation commands are based onUbuntu 20.04 LTS :

The below instalation 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

build-essential

pkg-config

cmake

clang

lldb

lld

libssl-dev

postgresql

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}

Stop default postgres (as we'll use the docker one)

sudo

systemctl

stop

postgresql sudo

systemctl

disable

postgresql

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

curl

-L

https://foundry.paradigm.xyz

|

bash foundryup

--branch

master

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

Clone the repo to the desired location

git

clone

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

zksync-era git

submodule

update

--init

--recursive

Installing & running the ZK Stack

1. Installing zkstack

zkstack creates and manages Elastic Chain ecosystem and ZK chains using the zksync stack. To installzkstack , run the following command:

cargo

install

--git

https://github.com/matter-labs/zksync-era/

--locked

zkstack

--force

2. Clone the zksync-era repository from github:

git

clone

https://github.com/matter-labs/zksync-era * Checkout to branch that is worked on for DA clients:

git

checkout

dz-sync-layer-da-clients * Change directory to Eracontracts * :

cd

contracts * Install submodules in the contracts directory:

git

submodule

update

--init

--recursive * Checkout branchdz-upd-da-contracts * in contracts folder:

git

checkout

dz-upd-da-contracts

3. Add custom chain logic

Looking back, we have successfully installedzkstack and cloned thezksync-era repository. We have also checked out branch on the mainzksync-era repository and another branch in thecontracts/ folder. Now, we can add our custom chain logic in files/contracts/LayerNameL1DAValidator.sol and/contracts/LayerNameL2DAValidator.sol .

You may also rename these two files, but you will have to change all instances of the old name in the code. These instances

are in these files:

/contracts/da-contracts/contracts/da-layer/LayerNameL1DAValidator.sol /contracts/l2-contracts/contracts/data-availability/da-layers/LayerNameL2DAValidator.sol /contracts/l1-contracts/src.ts/deploy.ts /contracts/l1-contracts/src.ts/deploy-process.ts * Run the following command:

./bin/zkt

4. Running the containers

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

zk

clean

--all Now, we are ready to start 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

5. 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 initalized 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 * Now that the contracts have been deployed, we just need to go to the chain's general config and add our own config that we may need for any specific usecase. You can get these here:

zksync-era/chainsCHAIN_NAME/configs/general.yaml

6. Run ZK chain

Finally, you just need to start the server which will be running the ZK chain. By default, it will be running onlocalhost:3050 .

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