How to Use Madara with Avail

Introduction

Embark on setting up your own Madara-based Validium, leveraging Avail as the data availability layer. This guide is tailored for deploying on Ethereum's Sepolia testnet and integrating with the Avail Goldberg testnet. To gain a comprehensive understanding of Madara Starknet, review theMadara documentation (opens in a new tab) .

In this guide, you will conduct the following:

Prerequisites

Ensure you have installed the following software.

Installation commands are based on Ubuntu 20.04 LTS: Software VersionRust (opens in a new tab) rustc 1.69.0-nightly or later nvm/Node.js (opens in a new tab) Latest version Cairo (opens in a new tab) 1.0

Install Rust

curl

--proto

'=https'

--tlsv1.2

-sSf

https://sh.rustup.rs

|

sh rustup

toolchain

install

nightly

Install nvm and Node.js

curl

-o-

https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh

|

bash export NVM_DIR = "([-z "{XDG_CONFIG_HOME-}"] &&

printf

%s "{HOME}/.nvm" ||

printf

%s "{XDG_CONFIG_HOME}/nvm")" [-s

"NVM_DIR/nvm.sh"] &&

.

"NVM_DIR/nvm.sh"

This loads nvm

nvm

install

--lts

Install Cairo

curl

-L

https://github.com/franalgaba/cairo-installer/raw/main/bin/cairo-installer

|

bash

Hardware Requirements

The Madara stack, being a StarkNet sequencer, has specific hardware requirements for efficient operation. These requirements are particularly important when integrating Avail as the DA Layer.

Component Minimum Requirements Recommended Setup Suggested AWS Instance Sequencer Node 4-core CPU, 16GB RAM, 100 GB SSD 8-core CPU, 32GB RAM, 200 GB SSD m5a.xlarge Avail Node 4-core CPU, 8GB RAM, 50 GB SSD 8-core CPU, 16GB RAM, 100 GB SSD m5a.large The storage requirements can vary based on the volume of transactions and the length of the chain history you intend to maintain. For high-traffic networks, consider scaling your storage capacity accordingly.

Launch an Avail-Powered Madara Validium

1. Clone the Madara repository:
2. git
3. clone
4. git@github.com:keep-starknet-strange/madara.git
5. cd
6. madara
7. Build the chain in release mode:
8. cargo
9. build
10. --release

Running a Single-Node Development Chain

1. Start the development chain:
2. ./target/release/madara
3. --dev
4. Purge the chain's state:
5. ./target/release/madara
6. purge-chain
7. --dev

8. Start with detailed logging:
9. RUST_BACKTRACE
10. =
11. 1
12. ./target/release/madara
13. -ldebug
14. --dev
15. Create a plain chain spec:
16. ./target/release/madara
17. build-spec
18.

19. ./infra/chain-sepcs/chain-spec-plain.json

Setting Up Avail as the DA Layer

1. Configure Avail for Madara:

2. # Configuration for local Avail node

3. cat
4.

5. da-config.json
6. <<
7. EOF
8. {
9. "ws_provider": "ws://127.0.0.1:9945",
10. "app_id": 0,
11. "validate_codegen": false,
12. "seed": "//Alice"
13. }
14. EOF
15. Start Avail Node (in another terminal):
16. ./data-avail
17. --dev
18. --rpc-port
19. 9945
20. --port
21. 30334
22. Setup Madara for Avail:
23. ./madara
24. setup
25. --chain=dev
26. --from-remote
27. --base-path=../.madara
28. Launch Madara with Avail:
29. ./madara
30. --chain=dev
31. --base-path=../.madara
32. --da-layer=avail
33. --force-authoring
34. --alice

Deploying an Account on Your Chain

- After setting up your chain, you can deploy your own account and start making transactions.

Using Avail in Madara

- Specify--da-layer avail
- when launching a Madara node to use Avail for publishing the state_diff.

Remember to adjust theda-config.json file based on your setup, whether it's a local node or using the Avail network.

Overview Madara Starknet