Hydra PoC Local

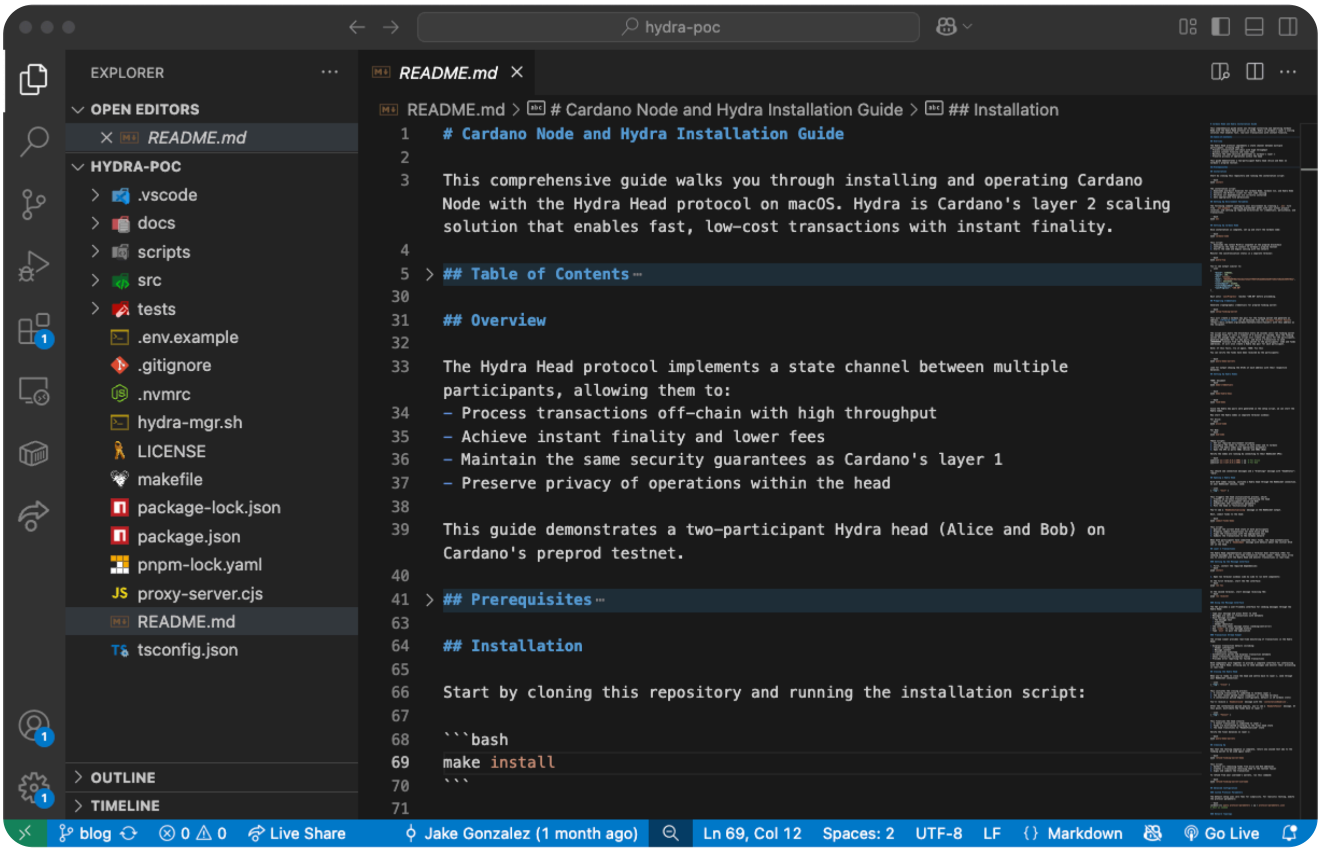
This document is related to cPoker Hydra Case Study: Implement Interactive dApp, and explains how to run of the Hydra POC on MacOS.

Getting started

Prepare a folder and clone the project

```
mkdir ~/poc && cd ~/poc
git clone git@github.com:Cardano-After-Dark/hydra-poc.git
cd hydra-poc
git checkout blog
```

Open the project and check the readme for the details.



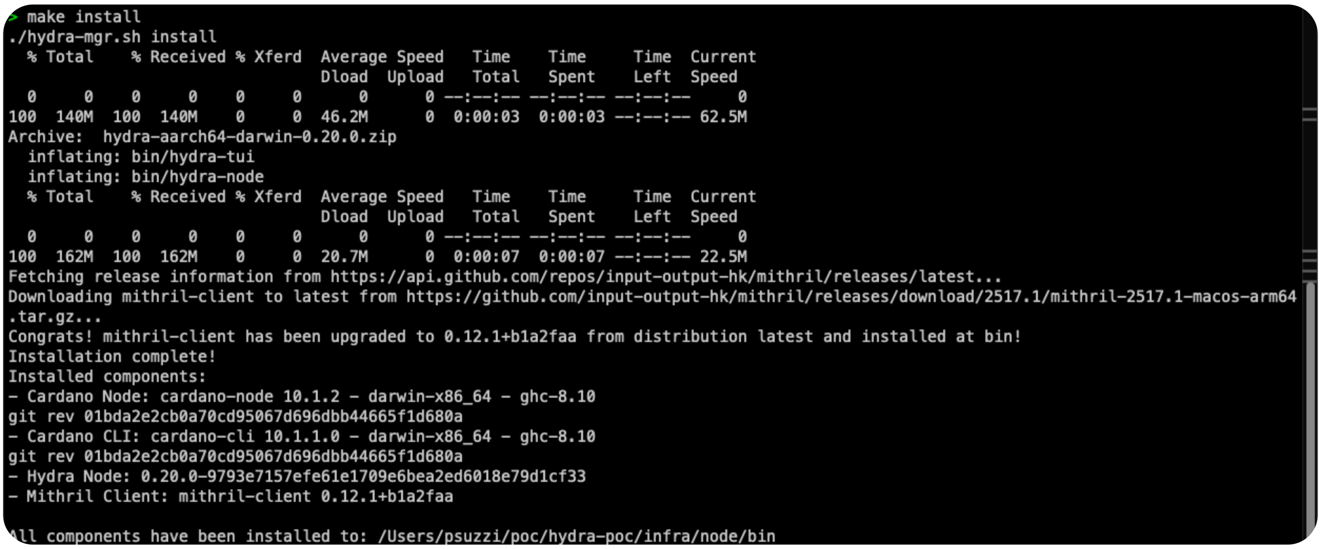
~~~
# Cardano Node and Hydra Installation Guide

This comprehensive guide walks you through installing and operating Cardano
Node with the Hydra Head protocol on macOS. Hydra is Cardano's layer 2 scaling
solution that enables fast, low-cost transactions with instant finality.

## Table of Contents

## Overview

The Hydra Head protocol implements a state channel between multiple
participants, allowing them to:
- Process transactions off-chain with high throughput
- Achieve instant finality and lower fees
- Maintain the same security guarantees as Cardano's layer 1
- Preserve privacy of operations within the head

This guide demonstrates a two-participant Hydra head (Alice and Bob) on
Cardano's preprod testnet.

## Prerequisites

## Installation

Start by cloning this repository and running the installation script:

```bash
make install
```
~~~

Installing and Setting Up Env Variables

Run the installation scripts and set up the environment variables to have a running environment.

First run install to setup the system, download mithryl, and prepare the `infra` directory

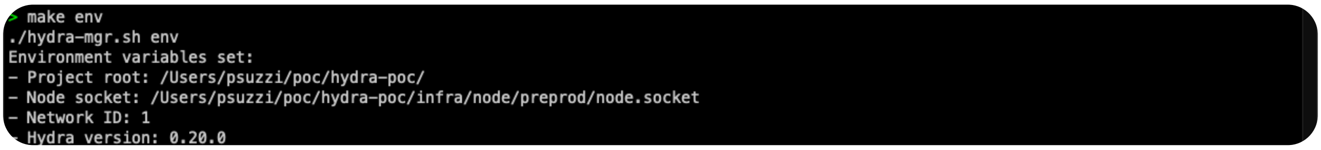
```
make install
```



```
> make install
./hydra-mgr.sh install
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  140M  100  140M    0     0  46.2M      0  0:00:03  0:00:03 --:--:-- 62.5M
Archive:  hydra-aarch64-darwin-0.20.0.zip
  inflating: bin/hydra-tui
  inflating: bin/hydra-node
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  162M  100  162M    0     0  20.7M      0  0:00:07  0:00:07 --:--:-- 22.5M
Fetching release information from https://api.github.com/repos/input-output-hk/mithril/releases/latest...
Downloading mithril-client to latest from https://github.com/input-output-hk/mithril/releases/download/2517.1/mithril-2517.1-macos-arm64
.tar.gz...
Congrats! mithril-client has been upgraded to 0.12.1+b1a2faa from distribution latest and installed at bin!
Installation complete!
Installed components:
- Cardano Node: cardano-node 10.1.2 - darwin-x86_64 - ghc-8.10
git rev 01bda2e2cb0a70cd95067d696dbb44665f1d680a
- Cardano CLI: cardano-cli 10.1.1.0 - darwin-x86_64 - ghc-8.10
git rev 01bda2e2cb0a70cd95067d696dbb44665f1d680a
- Hydra Node: 0.20.0-9793e7157efe61e1709e6bea2ed6018e79d1cf33
- Mithril Client: mithril-client 0.12.1+b1a2faa

All components have been installed to: /Users/psuzzi/poc/hydra-poc/infra/node/bin
```

Then run `env` to create an `.env` file to use for the demo

```
make env
```



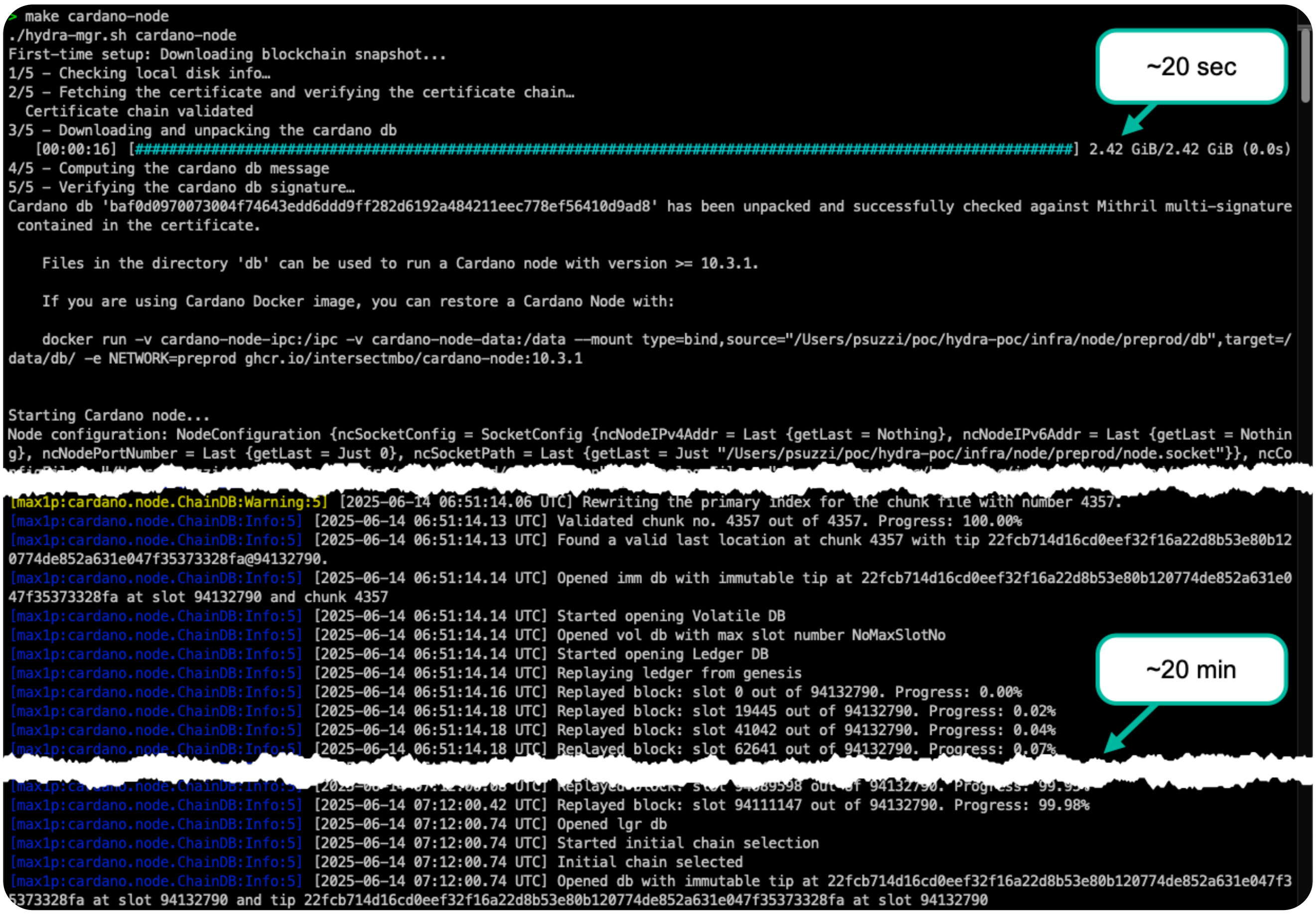
```
> make env
./hydra-mgr.sh env
Environment variables set:
- Project root: /Users/psuzzi/poc/hydra-poc/
- Node socket: /Users/psuzzi/poc/hydra-poc/infra/node/preprod/node.socket
- Network ID: 1
- Hydra version: 0.20.0
```

Setting up Cardano Node

Now, set up and start the Cardano node:

```
make cardano-node
```

This will download the cardano db, deflate it, and replay all its transactions. Wait until all the transactions are replayed and the cardano node is started.



```
> make cardano-node
./hydra-mgr.sh cardano-node
First-time setup: Downloading blockchain snapshot...
1/5 - Checking local disk info…
2/5 - Fetching the certificate and verifying the certificate chain…
  Certificate chain validated
3/5 - Downloading and unpacking the cardano db
   [00:00:16] [################################################################################################] 2.42 GiB/2.42 GiB (0.0s)
4/5 - Computing the cardano db message
5/5 - Verifying the cardano db signature…
Cardano db 'baf0d0970073004f74643edd6ddd9ff282d6192a484211eec778ef56410d9ad8' has been unpacked and successfully checked against Mithril multi-signature
 contained in the certificate.

    Files in the directory 'db' can be used to run a Cardano node with version >= 10.3.1.

    If you are using Cardano Docker image, you can restore a Cardano Node with:

    docker run -v cardano-node-ipc:/ipc -v cardano-node-data:/data --mount type=bind,source="/Users/psuzzi/poc/hydra-poc/infra/node/preprod/db",target=/
data/db/ -e NETWORK=preprod ghcr.io/intersectmbo/cardano-node:10.3.1


Starting Cardano node...
Node configuration: NodeConfiguration {ncSocketConfig = SocketConfig {ncNodeIPv4Addr = Last {getLast = Nothing}, ncNodeIPv6Addr = Last {getLast = Nothin
g}, ncNodePortNumber = Last {getLast = Just 0}, ncSocketPath = Last {getLast = Just "/Users/psuzzi/poc/hydra-poc/infra/node/preprod/node.socket"}}, ncCo

[max1p:cardano.node.ChainDB:Warning:5] [2025-06-14 06:51:14.06 UTC] Rewriting the primary index for the chunk file with number 4357.
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.13 UTC] Validated chunk no. 4357 out of 4357. Progress: 100.00%
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.13 UTC] Found a valid last location at chunk 4357 with tip 22fcb714d16cd0eef32f16a22d8b53e80b12
0774de852a631e047f35373328fa@94132790.
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.14 UTC] Opened imm db with immutable tip at 22fcb714d16cd0eef32f16a22d8b53e80b120774de852a631e0
47f35373328fa at slot 94132790 and chunk 4357
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.14 UTC] Started opening Volatile DB
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.14 UTC] Opened vol db with max slot number NoMaxSlotNo
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.14 UTC] Started opening Ledger DB
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.14 UTC] Replaying ledger from genesis
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.16 UTC] Replayed block: slot 0 out of 94132790. Progress: 0.00%
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.18 UTC] Replayed block: slot 19445 out of 94132790. Progress: 0.02%
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.18 UTC] Replayed block: slot 41042 out of 94132790. Progress: 0.04%
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 06:51:14.18 UTC] Replayed block: slot 62641 out of 94132790. Progress: 0.07%

[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 07:12:00.08 UTC] Replayed block: slot 94089598 out of 94132790. Progress: 99.95%
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 07:12:00.42 UTC] Replayed block: slot 94111147 out of 94132790. Progress: 99.98%
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 07:12:00.74 UTC] Opened lgr db
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 07:12:00.74 UTC] Started initial chain selection
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 07:12:00.74 UTC] Initial chain selected
[max1p:cardano.node.ChainDB:Info:5] [2025-06-14 07:12:00.74 UTC] Opened db with immutable tip at 22fcb714d16cd0eef32f16a22d8b53e80b120774de852a631e047f3
5373328fa at slot 94132790 and tip 22fcb714d16cd0eef32f16a22d8b53e80b120774de852a631e047f35373328fa at slot 94132790
```

~20 sec

~20 min

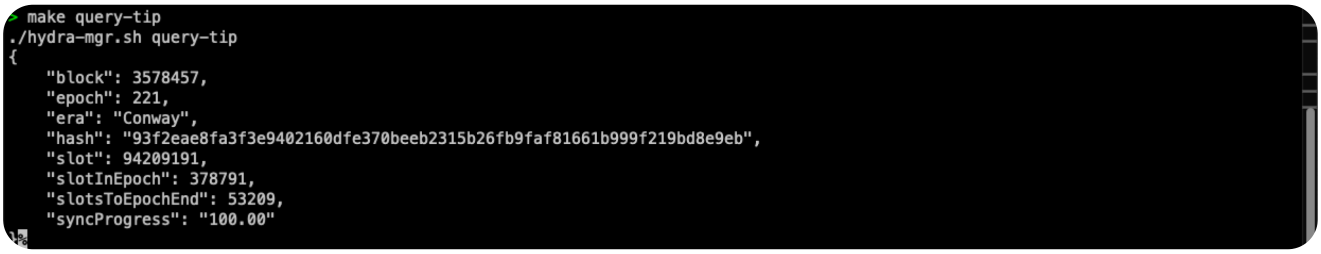
Notes:

- If requested, grant terminal the permission to access the local network.
- When completed, leave the terminal open with the cardano node running
- To restart run `make cardano-node` but chose `n` to not remove the existing database.

Check the cardano node is updated by querying the tip of the running node

```
make query-tip
```

In your terminal you should see a confirmation

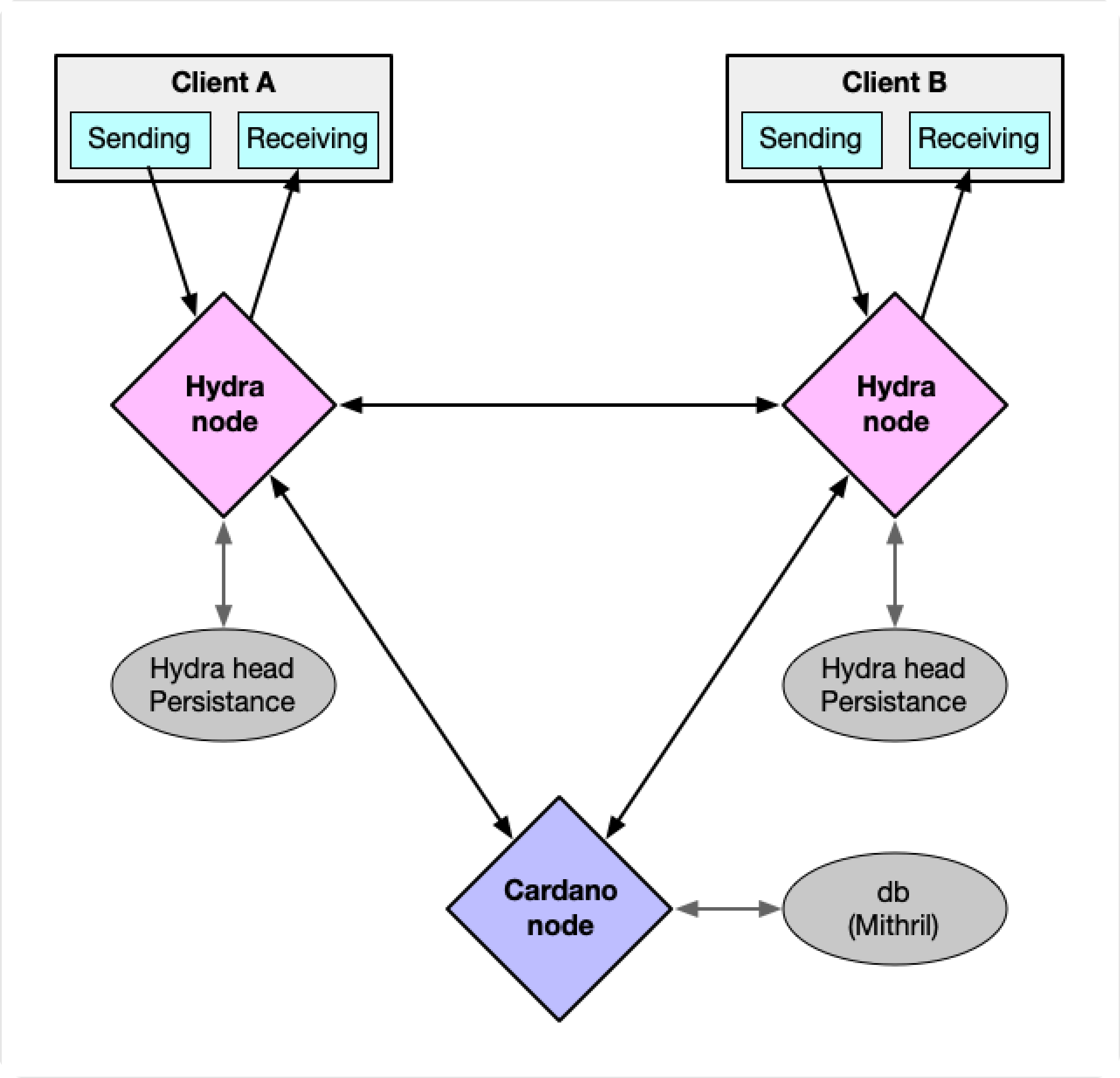


```
> make query-tip
./hydra-mgr.sh query-tip
{
    "block": 3578457,
    "epoch": 221,
    "era": "Conway",
    "hash": "93f2eae8fa3f3e9402160dfe370beeb2315b26fb9faf81661b999f219bd8e9eb",
    "slot": 94209191,
    "slotInEpoch": 378791,
    "slotsToEpochEnd": 53209,
    "syncProgress": "100.00"
```

Now we have the cardano node set up to run the PoCs

Hydra PoC Local

In this PoC we run two clients: A, and B, each one communicating through a two hydra nodes running agains the same Cardano node.

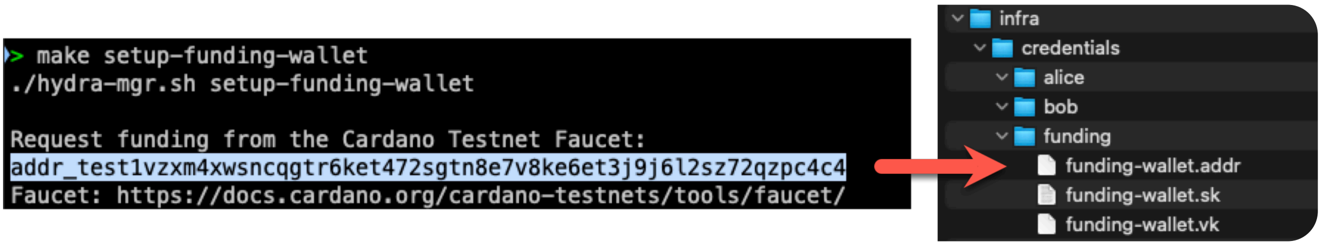


Preparing Credentials

Generate cryptographic credentials for preprod funding wallet:

```
make setup-funding-wallet
```

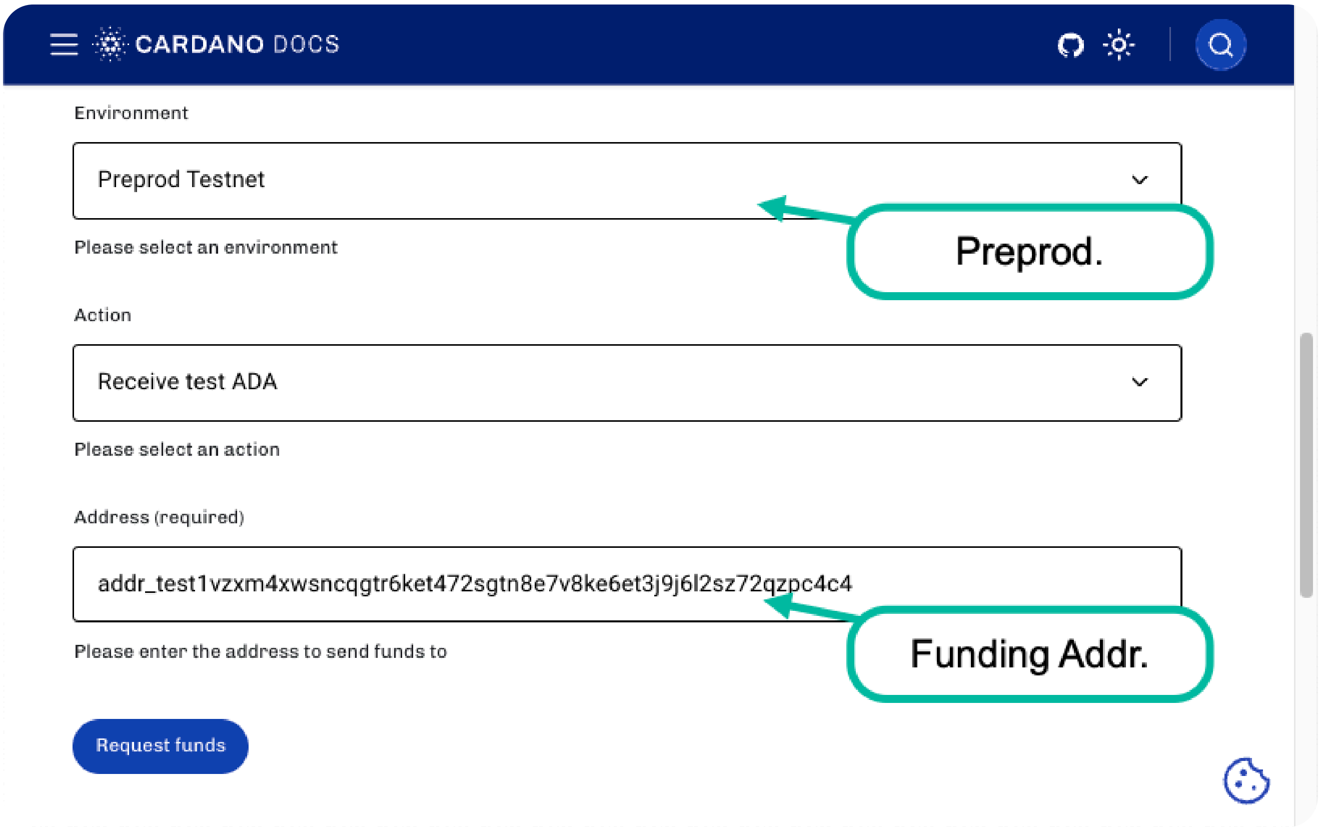
This will create a Cardano key pair for the funding wallet and generate an address.



```
> make setup-funding-wallet
./hydra-mgr.sh setup-funding-wallet

Request funding from the Cardano Testnet Faucet:
addr_test1vzxm4xwsncqgtr6ket472sgtn8e7v8ke6et3j9j6l2sz72qzpc4c4
Faucet: https://docs.cardano.org/cardano-testnets/tools/faucet/
```

```
infra
  credentials
    alice
    bob
    funding
      funding-wallet.addr
      funding-wallet.sk
      funding-wallet.vk
```

Preprod tADA can be obtained from the Cardano Testnet Faucet with this address as the recipient.



Environment

Preprod Testnet

Please select an environment

Action

Receive test ADA

Please select an action

Address (required)

addr_test1vzxm4xwsncqgtr6ket472sgtn8e7v8ke6et3j9j6l2sz72qzpc4c4

Please enter the address to send funds to

Preprod.

Funding Addr.

Creating credentials for Alice and bob in the folder `infra\credentials\alice|bob`:

```
# User credentials
make demo-credentials
# Hydra key pairs
make demo hydra-keys
```

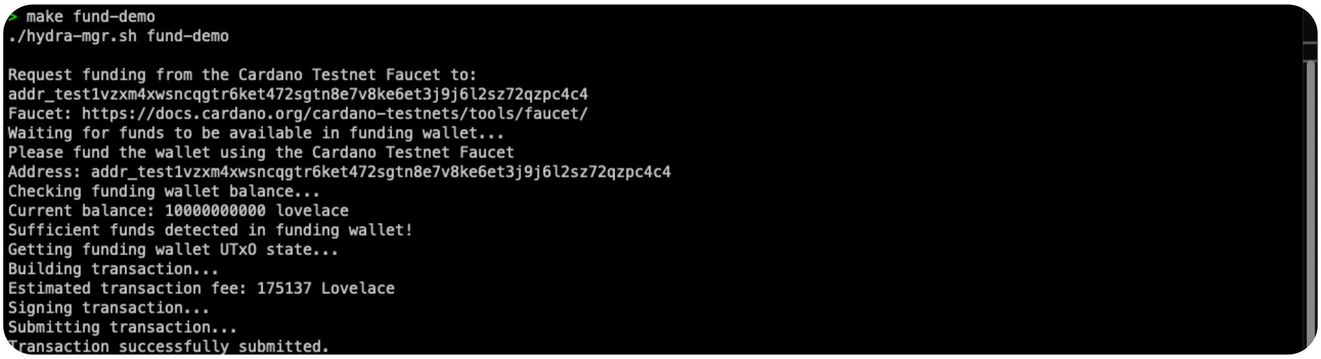
At this stage, the credentials will be into `infra\credentials\alice|bob`

- `<user>-node.*` : user's node credentials
- `<user>-funds.*` : user's funds credentials
- `<user>-hydra.*` : Hydra key pair for the user

Transfering the preprod tADA from funding wallet to the users wallets

```
make fund-demo
```

If you already transferred the tADA from the the Cardano Testnet Faucet, you'll see the balance is transfered to users. Otherwise, it waits for fund transfer.

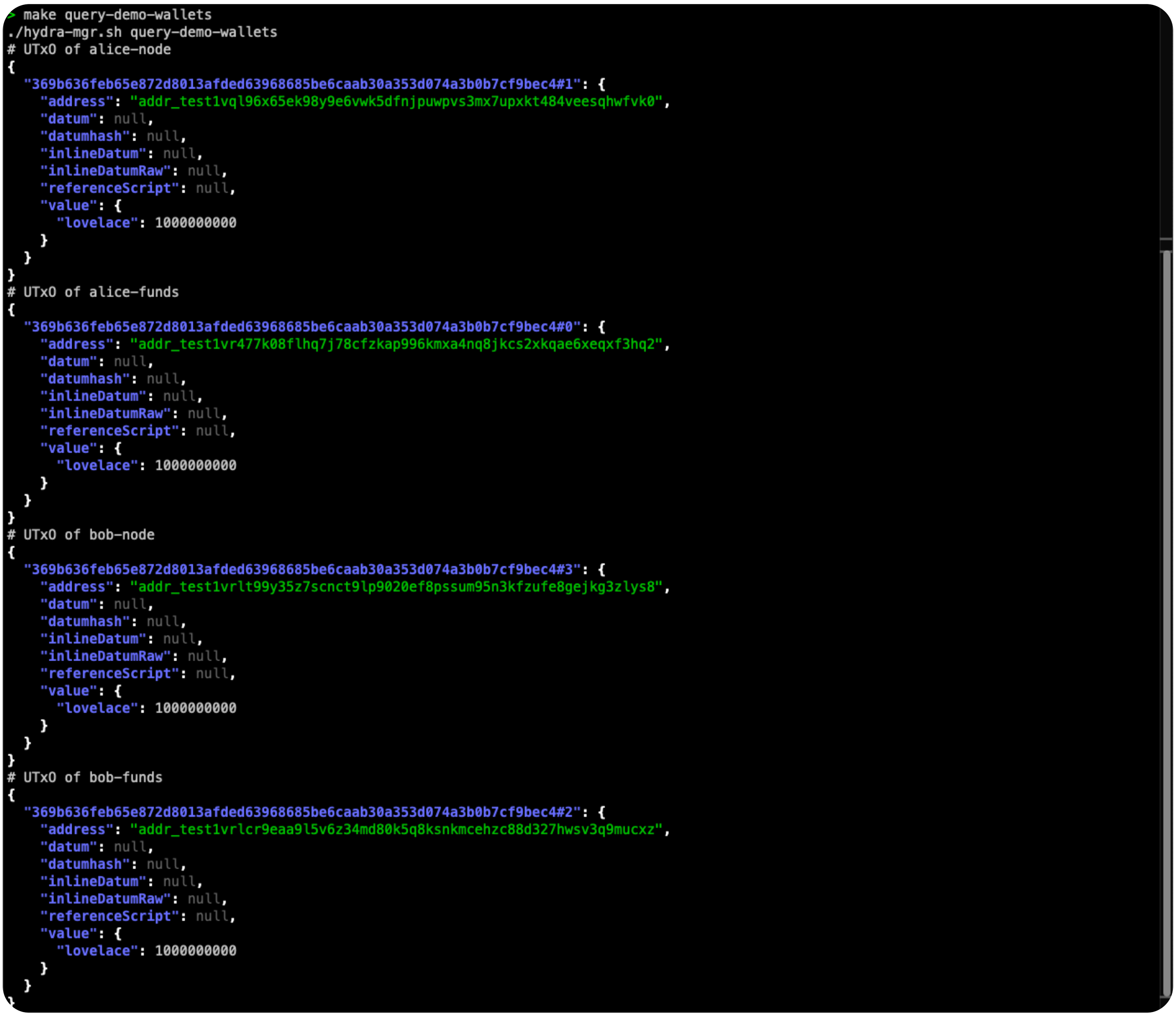


```
> make fund-demo
./hydra-mgr.sh fund-demo

Request funding from the Cardano Testnet Faucet to:
addr_test1vzxm4xwsncqgtr6ket472sgtn8e7v8ke6et3j9j6l2sz72qzpc4c4
Faucet: https://docs.cardano.org/cardano-testnets/tools/faucet/
Waiting for funds to be available in funding wallet...
Please fund the wallet using the Cardano Testnet Faucet
Address: addr_test1vzxm4xwsncqgtr6ket472sgtn8e7v8ke6et3j9j6l2sz72qzpc4c4
Checking funding wallet balance...
Current balance: 10000000000 lovelace
Sufficient funds detected in funding wallet!
Getting funding wallet UTxO state...
Building transaction...
Estimated transaction fee: 175137 Lovelace
Signing transaction...
Submitting transaction...
Transaction successfully submitted.
```

When done, you can check Alice's and Bob's funds by querying

```
make query-demo-wallets
```

See output example



```
> make query-demo-wallets
./hydra-mgr.sh query-demo-wallets
# UTxO of alice-node
{
  "369b636feb65e872d8013afded63968685be6caab30a353d074a3b0b7cf9bec4#1": {
    "address": "addr_test1vql96x65ek98y9e6vwk5dfnjpuwpvs3mx7upxkt484veesqhwfvk0",
    "datum": null,
    "datumhash": null,
    "inlineDatum": null,
    "inlineDatumRaw": null,
    "referenceScript": null,
    "value": {
      "lovelace": 1000000000
    }
  }
}
# UTxO of alice-funds
{
  "369b636feb65e872d8013afded63968685be6caab30a353d074a3b0b7cf9bec4#0": {
    "address": "addr_test1vr477k08flhq7j78cfzkap996kmxa4nq8jkcs2xkqae6xeqxf3hq2",
    "datum": null,
    "datumhash": null,
    "inlineDatum": null,
    "inlineDatumRaw": null,
    "referenceScript": null,
    "value": {
      "lovelace": 1000000000
    }
  }
}
# UTxO of bob-node
{
  "369b636feb65e872d8013afded63968685be6caab30a353d074a3b0b7cf9bec4#3": {
    "address": "addr_test1vrlt99y35z7scnct9lp9020ef8pssum95n3kfzufe8gejkg3zlys8",
    "datum": null,
    "datumhash": null,
    "inlineDatum": null,
    "inlineDatumRaw": null,
    "referenceScript": null,
    "value": {
      "lovelace": 1000000000
    }
  }
}
# UTxO of bob-funds
{
  "369b636feb65e872d8013afded63968685be6caab30a353d074a3b0b7cf9bec4#2": {
    "address": "addr_test1vrlcr9eaa9l5v6z34md80k5q8ksnkmcehzc88d327hwsv3q9mucxz",
    "datum": null,
    "datumhash": null,
    "inlineDatum": null,
    "inlineDatumRaw": null,
    "referenceScript": null,
    "value": {
      "lovelace": 1000000000
    }
  }
}
```

Starting Hydra Nodes

Start the hydra nodes in separate terminal windows:

For Alice:

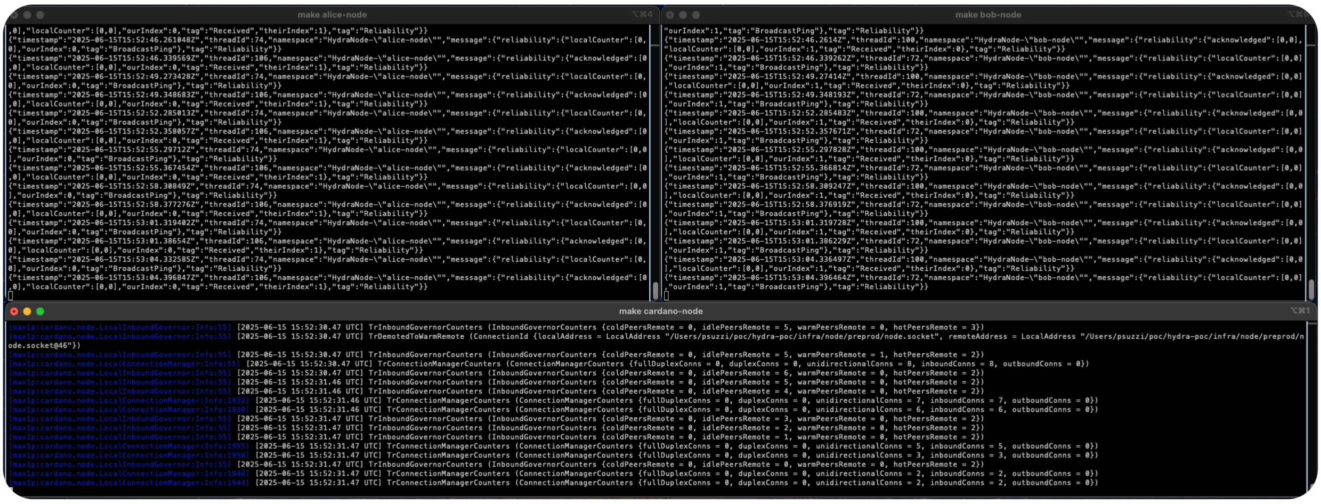
```
make alice-node
```

For Bob:

```
make bob-node
```

At this stage, you should have

- a terminal running the cardano node
- two more terminals running the Hydra nodes for Alice and Bob



```
,0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:46.261048Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,
0],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:46.339569Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0
,0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:49.273428Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,
0],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:49.348683Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0
,0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:52.285013Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,
0],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:52.358057Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0
,0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:55.29712Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,0
],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:55.367454Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0
,0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:58.30849Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,0
],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:58.377276Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0
,0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:01.319402Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,
0],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:01.38654Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0,
0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:04.332585Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,
0],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:04.396847Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0
,0],"localCounter":[0,0],"ourIndex":0,"tag":"Received","theirIndex":1},"tag":"Reliability"}}
```

```
"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:46.2614Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0],
"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:46.339262Z","threadId":72,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"localCounter":[0,0]
,"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:49.27414Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0]
,"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:49.348193Z","threadId":72,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"localCounter":[0,0]
,"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:52.285483Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0
],"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:52.357671Z","threadId":72,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"localCounter":[0,0]
,"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:55.297828Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0
],"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:55.366814Z","threadId":72,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"localCounter":[0,0]
,"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:58.309247Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0
],"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:52:58.376919Z","threadId":72,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"localCounter":[0,0]
,"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:01.319728Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0
],"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:01.386229Z","threadId":72,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"localCounter":[0,0]
,"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:04.336497Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0
],"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
{"timestamp":"2025-06-15T15:53:04.396464Z","threadId":72,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"localCounter":[0,0]
,"ourIndex":1,"tag":"BroadcastPing"},"tag":"Reliability"}}
```

```
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:30.47 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 5, warmPeersRemote = 0, hotPeersRemote = 3})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:30.47 UTC] TrDemotedToWarmRemote (ConnectionId {localAddress = LocalAddress "/Users/psuzzi/poc/hydra-poc/infra/node/preprod/node.socket", remoteAddress = LocalAddress "/Users/psuzzi/poc/hydra-poc/infra/node/preprod/n
ode.socket@46"})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:30.47 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 5, warmPeersRemote = 1, hotPeersRemote = 2})
[max1p:cardano.node.LocalConnectionManager:Info:55] [2025-06-15 15:52:30.47 UTC] TrConnectionManagerCounters (ConnectionManagerCounters {fullDuplexConns = 0, duplexConns = 0, unidirectionalConns = 8, inboundConns = 8, outboundConns = 0})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:30.47 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 6, warmPeersRemote = 0, hotPeersRemote = 2})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:31.46 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 5, warmPeersRemote = 0, hotPeersRemote = 2})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:31.46 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 4, warmPeersRemote = 0, hotPeersRemote = 2})
[max1p:cardano.node.LocalConnectionManager:Info:1932] [2025-06-15 15:52:31.46 UTC] TrConnectionManagerCounters (ConnectionManagerCounters {fullDuplexConns = 0, duplexConns = 0, unidirectionalConns = 7, inboundConns = 7, outboundConns = 0})
[max1p:cardano.node.LocalConnectionManager:Info:1930] [2025-06-15 15:52:31.46 UTC] TrConnectionManagerCounters (ConnectionManagerCounters {fullDuplexConns = 0, duplexConns = 0, unidirectionalConns = 6, inboundConns = 6, outboundConns = 0})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:31.47 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 3, warmPeersRemote = 0, hotPeersRemote = 2})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:31.47 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 2, warmPeersRemote = 0, hotPeersRemote = 2})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:31.47 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 1, warmPeersRemote = 0, hotPeersRemote = 2})
[max1p:cardano.node.LocalConnectionManager:Info:1955] [2025-06-15 15:52:31.47 UTC] TrConnectionManagerCounters (ConnectionManagerCounters {fullDuplexConns = 0, duplexConns = 0, unidirectionalConns = 5, inboundConns = 5, outboundConns = 0})
[max1p:cardano.node.LocalConnectionManager:Info:1950] [2025-06-15 15:52:31.47 UTC] TrConnectionManagerCounters (ConnectionManagerCounters {fullDuplexConns = 0, duplexConns = 0, unidirectionalConns = 3, inboundConns = 3, outboundConns = 0})
[max1p:cardano.node.LocalInboundGovernor:Info:55] [2025-06-15 15:52:31.47 UTC] TrInboundGovernorCounters (InboundGovernorCounters {coldPeersRemote = 0, idlePeersRemote = 0, warmPeersRemote = 0, hotPeersRemote = 2})
[max1p:cardano.node.LocalConnectionManager:Info:1940] [2025-06-15 15:52:31.47 UTC] TrConnectionManagerCounters (ConnectionManagerCounters {fullDuplexConns = 0, duplexConns = 0, unidirectionalConns = 2, inboundConns = 2, outboundConns = 0})
[max1p:cardano.node.LocalConnectionManager:Info:1944] [2025-06-15 15:52:31.47 UTC] TrConnectionManagerCounters (ConnectionManagerCounters {fullDuplexConns = 0, duplexConns = 0, unidirectionalConns = 2, inboundConns = 2, outboundConns = 0})
```

Let's keep all this infrastructure running, and launch two more terminals to verify the nodes are running by connecting to their websocket APIs

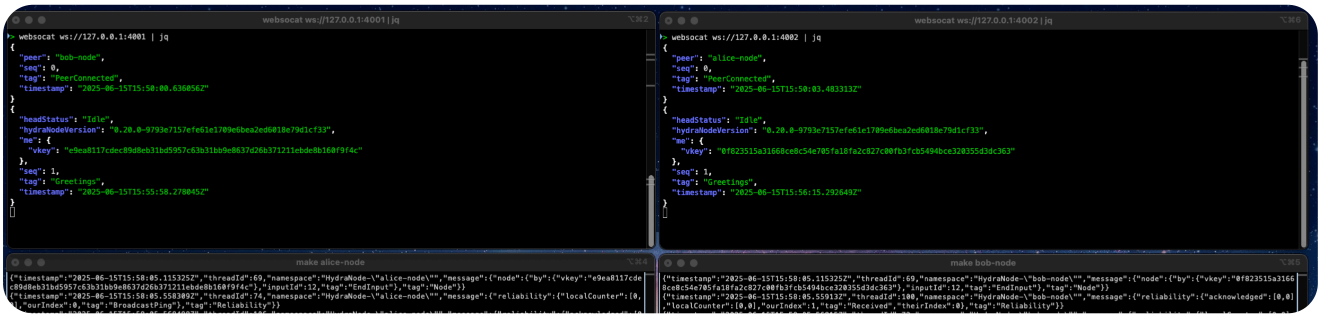
Open a terminal

```
websocat ws://127.0.0.1:4001 | jq # For Alice
```

Open another terminal

```
websocat ws://127.0.0.1:4002 | jq # For Bob
```

You should see connection messages and a "Greetings" message with "headStatus": "Idle".


websocat ws://127.0.0.1:4001 | jq
⌥⌘2
websocat ws://127.0.0.1:4002 | jq
⌥⌘6
make alice-node
⌥⌘4
make bob-node
⌥⌘5

```
> websocat ws://127.0.0.1:4001 | jq
{
  "peer": "bob-node",
  "seq": 0,
  "tag": "PeerConnected",
  "timestamp": "2025-06-15T15:50:00.636056Z"
}
{
  "headStatus": "Idle",
  "hydraNodeVersion": "0.20.0-9793e7157efe61e1709e6bea2ed6018e79d1cf33",
  "me": {
    "vkey": "e9ea8117cdec89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"
  },
  "seq": 1,
  "tag": "Greetings",
  "timestamp": "2025-06-15T15:55:58.278045Z"
}
```

```
> websocat ws://127.0.0.1:4002 | jq
{
  "peer": "alice-node",
  "seq": 0,
  "tag": "PeerConnected",
  "timestamp": "2025-06-15T15:50:03.483313Z"
}
{
  "headStatus": "Idle",
  "hydraNodeVersion": "0.20.0-9793e7157efe61e1709e6bea2ed6018e79d1cf33",
  "me": {
    "vkey": "0f823515a31668ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"
  },
  "seq": 1,
  "tag": "Greetings",
  "timestamp": "2025-06-15T15:56:15.292649Z"
}
```

```
{"timestamp":"2025-06-15T15:58:05.115325Z","threadId":69,"namespace":"HydraNode-\"alice-node\"","message":{"node":{"by":{"vkey":"e9ea8117cde
c89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"},"inputId":12,"tag":"EndInput"},"tag":"Node"}}
{"timestamp":"2025-06-15T15:58:05.558309Z","threadId":74,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"localCounter":[0,
0],"ourIndex":0,"tag":"BroadcastPing"},"tag":"Reliability"}}
```

```
{"timestamp":"2025-06-15T15:58:05.115325Z","threadId":69,"namespace":"HydraNode-\"bob-node\"","message":{"node":{"by":{"vkey":"0f823515a3166
8ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"},"inputId":12,"tag":"EndInput"},"tag":"Node"}}
{"timestamp":"2025-06-15T15:58:05.55913Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0]
,"localCounter":[0,0],"ourIndex":1,"tag":"Received","theirIndex":0},"tag":"Reliability"}}
```

Opening a Hydra Head

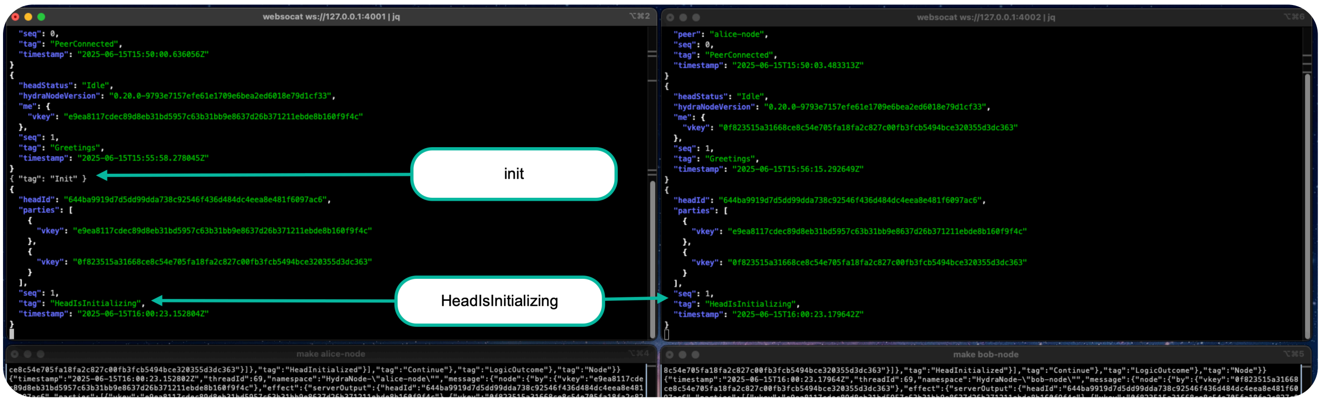
With both nodes running, initiate a Hydra head through the WebSocket connection. In your WebSocket session, send:

```
{ "tag": "Init" }
```

This triggers the head initialization process, which:

1. Creates an on-chain transaction referencing the head
2. Registers the participants (Alice and Bob)
3. Establishes the contestation parameters
4. Puts the head in "Initializing" state

You'll see a `HeadIsInitializing` message in the WebSocket output.


websocat ws://127.0.0.1:4001 | jq
⌥⌘2
init
HeadIsInitializing
websocat ws://127.0.0.1:4002 | jq
⌥⌘6
make alice-node
⌥⌘4
make bob-node
⌥⌘5

```
  "seq": 0,
  "tag": "PeerConnected",
  "timestamp": "2025-06-15T15:50:00.636056Z"
}
{
  "headStatus": "Idle",
  "hydraNodeVersion": "0.20.0-9793e7157efe61e1709e6bea2ed6018e79d1cf33",
  "me": {
    "vkey": "e9ea8117cdec89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"
  },
  "seq": 1,
  "tag": "Greetings",
  "timestamp": "2025-06-15T15:55:58.278045Z"
}
{ "tag": "Init" }
{
  "headId": "644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481f6097ac6",
  "parties": [
    {
      "vkey": "e9ea8117cdec89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"
    },
    {
      "vkey": "0f823515a31668ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"
    }
  ],
  "seq": 1,
  "tag": "HeadIsInitializing",
  "timestamp": "2025-06-15T16:00:23.152804Z"
}
```

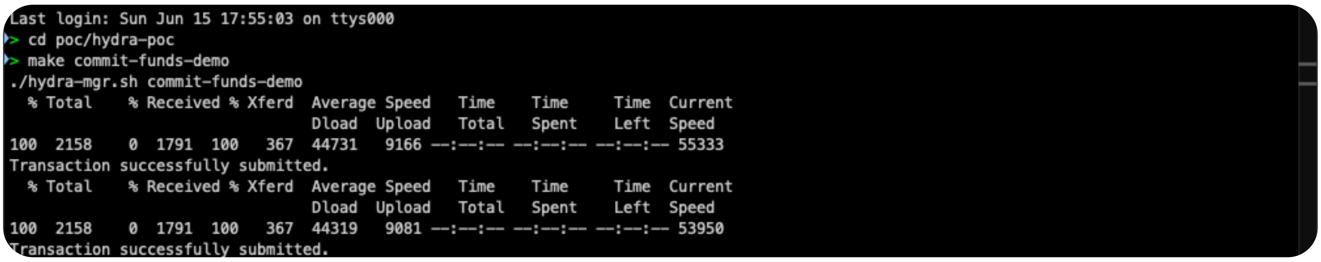
```
  "peer": "alice-node",
  "seq": 0,
  "tag": "PeerConnected",
  "timestamp": "2025-06-15T15:50:03.483313Z"
}
{
  "headStatus": "Idle",
  "hydraNodeVersion": "0.20.0-9793e7157efe61e1709e6bea2ed6018e79d1cf33",
  "me": {
    "vkey": "0f823515a31668ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"
  },
  "seq": 1,
  "tag": "Greetings",
  "timestamp": "2025-06-15T15:56:15.292649Z"
}
{
  "headId": "644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481f6097ac6",
  "parties": [
    {
      "vkey": "e9ea8117cdec89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"
    },
    {
      "vkey": "0f823515a31668ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"
    }
  ],
  "seq": 1,
  "tag": "HeadIsInitializing",
  "timestamp": "2025-06-15T16:00:23.179642Z"
}
```

```
ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"}]},"tag":"HeadInitialized"}],"tag":"Continue"},"tag":"LogicOutcome"},"tag":"Node"}}
{"timestamp":"2025-06-15T16:00:23.152802Z","threadId":69,"namespace":"HydraNode-\"alice-node\"","message":{"node":{"by":{"vkey":"e9ea8117cde
c89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"},"effect":{"serverOutput":{"headId":"644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481
```

```
8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"}]},"tag":"HeadInitialized"}],"tag":"Continue"},"tag":"LogicOutcome"},"tag":"Node"}}
{"timestamp":"2025-06-15T16:00:23.17964Z","threadId":69,"namespace":"HydraNode-\"bob-node\"","message":{"node":{"by":{"vkey":"0f823515a31668
ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"},"effect":{"serverOutput":{"headId":"644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481f60
```

Next, commit funds to the head:

```
make commit-funds-demo
```



```
Last login: Sun Jun 15 17:55:03 on ttys000
> cd poc/hydra-poc
> make commit-funds-demo
./hydra-mgr.sh commit-funds-demo
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2158    0  1791  100   367  44731   9166 --:--:-- --:--:-- --:--:-- 55333
Transaction successfully submitted.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2158    0  1791  100   367  44319   9081 --:--:-- --:--:-- --:--:-- 53950
Transaction successfully submitted.
```

This script:

1. Queries the current UTxO state of both participants
2. Prepares commit transactions for both Alice and Bob
3. Signs the transactions with the appropriate keys
4. Submits the transactions to the Cardano network

When both participants have committed their funds, the head automatically opens. You'll see a `HeadIsOpen` message with details about the initial UTxO set in the head.

Layer 2 transactions

The Hydra Head implementation includes a Terminal User Interface (TUI) chat for sending messages and a real-time transaction stream viewer. These tools allow you to interact with the Hydra Head and monitor transactions in real-time.

Setting Up the Chat

First, install the required dependencies:

```
pnpm install
```

Open two terminal windows side by side to run both message sender and receiver for Alice and Bob. For each terminal window do the following:

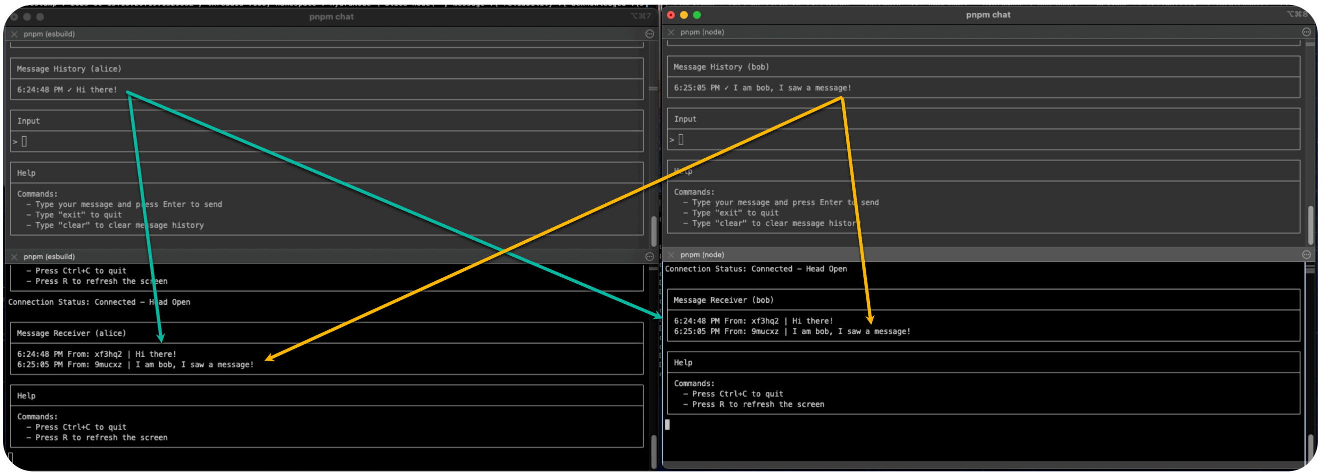
- Split the terminal horizontally
- In the upper part:
  - run `pnpm chat`,
  - select 1 to start the sender TUI
  - type the username (alice or bob)
- In the lower part:
  - run `pnpm chat`,
  - select 2 to start the sender TUI
  - type the username (alice or bob)

Using the Chat

The chat provides two components in a user-friendly interface:

- Sending: for sending messages through the Hydra Head
- Receiving: for monitoring in real time of the transactions in the Hydra Head

Here is an example of chat commuinication between Alice and Bob



```
Message History (alice)
6:24:48 PM ✓ Hi there!

Input
> 

Help
Commands:
  – Type your message and press Enter to send
  – Type "exit" to quit
  – Type "clear" to clear message history
```

```
  – Press Ctrl+C to quit
  – Press R to refresh the screen
Connection Status: Connected – Head Open

Message Receiver (alice)
6:24:48 PM From: xf3hq2 | Hi there!
6:25:05 PM From: 9mucxz | I am bob, I saw a message!

Help
Commands:
  – Press Ctrl+C to quit
  – Press R to refresh the screen
```

```
Message History (bob)
6:25:05 PM ✓ I am bob, I saw a message!

Input
> 

Help
Commands:
  – Type your message and press Enter to send
  – Type "exit" to quit
  – Type "clear" to clear message history
```

```
Connection Status: Connected – Head Open

Message Receiver (bob)
6:24:48 PM From: xf3hq2 | Hi there!
6:25:05 PM From: 9mucxz | I am bob, I saw a message!

Help
Commands:
  – Press Ctrl+C to quit
  – Press R to refresh the screen
```

Closing the Hydra Head

When you're ready to close the head and settle back to layer 1, send through your WebSocket connection:

```json
{ "tag": "Close" }
```

This initiates the closing process:

1. A closing transaction is submitted to Cardano layer 1
2. The most recent agreed state (snapshot) is recorded on-chain
3. A contestation period begins (configurable, default is 10 Cardano slots)

You'll receive a `HeadIsClosed` message with the `contestationDeadline`.

After the contestation period expires, you'll see a `ReadyToFanout` message. At this point, distribute the funds back to layer 1:

```json
{ "tag": "Fanout" }
```

This finalizes the head closure:

1. A fanout transaction is submitted to layer 1
2. Funds are distributed according to the final head state
3. The head transitions to "HeadIsFinalized" state

Verify the final balances on layer 1:

```bash
make query-demo-wallets
```

Cleaning Up

Now that the testing sequence is complete, return any unused test ada to the funding wallet to be used again later:

```bash
make refund-funding-wallet-demo
```

This script:

1. Gathers all remaining funds from Alice and Bob addresses
2. Creates a transaction returning them to the testnet faucet
3. Signs and submits the transaction

To refund from your username's wallets, run this command:

```
make refund-funding-wallet-username
```

This script:

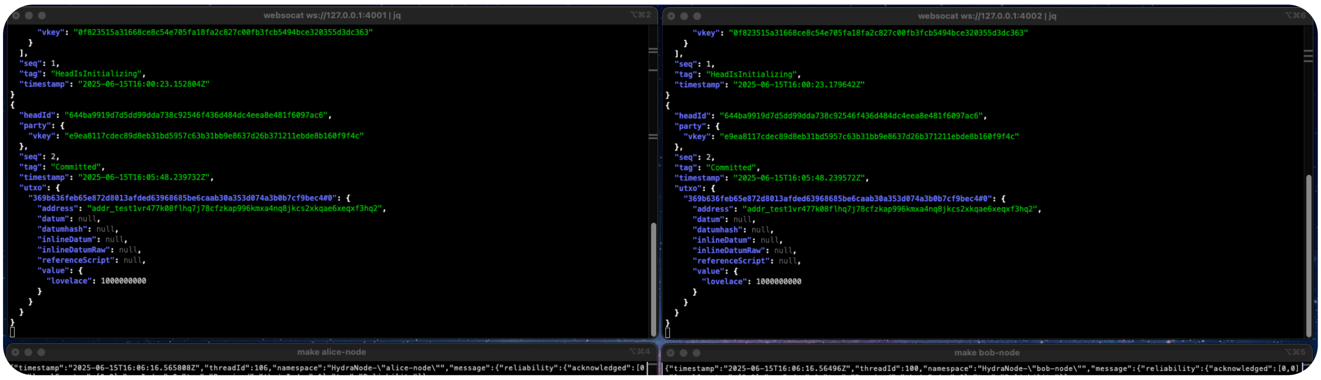
1. Gathers all remaining funds from Alice and Bob addresses
2. Creates a transaction returning them to the testnet faucet
3. Signs and submits the transaction

To refund from your username's wallets, run this command:

```
make refund-funding-wallet-username
```

Possible Error

If you find this error, just disregard it for now.

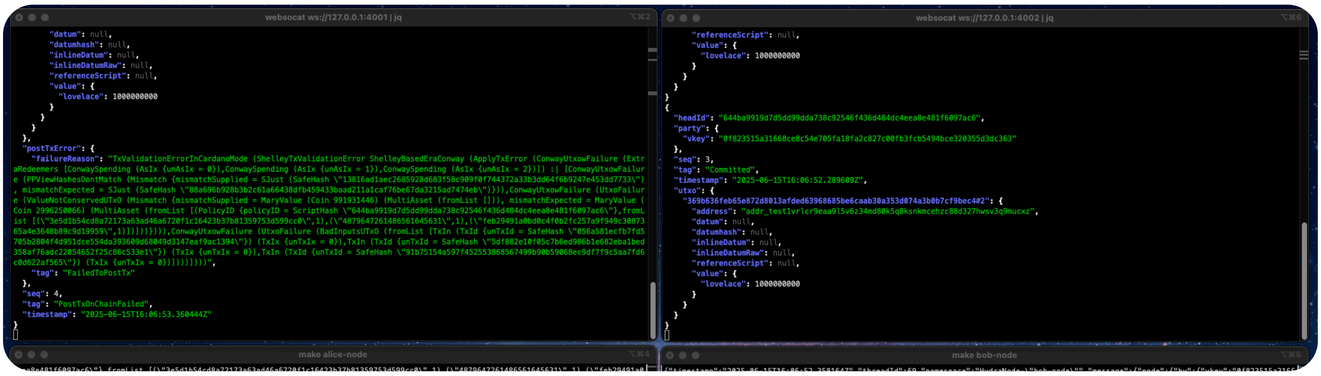

websocat ws://127.0.0.1:4001 | jq
⌥⌘2
websocat ws://127.0.0.1:4002 | jq
⌥⌘6
make alice-node
⌥⌘4
make bob-node
⌥⌘5

```
      "vkey": "0f823515a31668ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"
    }
  ],
  "seq": 1,
  "tag": "HeadIsInitializing",
  "timestamp": "2025-06-15T16:00:23.152804Z"
}
{
  "headId": "644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481f6097ac6",
  "party": {
    "vkey": "e9ea8117cdec89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"
  },
  "seq": 2,
  "tag": "Committed",
  "timestamp": "2025-06-15T16:05:48.239732Z",
  "utxo": {
    "369b636feb65e872d8013afded63968685be6caab30a353d074a3b0b7cf9bec4#0": {
      "address": "addr_test1vr477k08flhq7j78cfzkap996kmxa4nq8jkcs2xkqae6xeqxf3hq2",
      "datum": null,
      "datumhash": null,
      "inlineDatum": null,
      "inlineDatumRaw": null,
      "referenceScript": null,
      "value": {
        "lovelace": 1000000000
      }
    }
  }
}
```

```
      "vkey": "0f823515a31668ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"
    }
  ],
  "seq": 1,
  "tag": "HeadIsInitializing",
  "timestamp": "2025-06-15T16:00:23.179642Z"
}
{
  "headId": "644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481f6097ac6",
  "party": {
    "vkey": "e9ea8117cdec89d8eb31bd5957c63b31bb9e8637d26b371211ebde8b160f9f4c"
  },
  "seq": 2,
  "tag": "Committed",
  "timestamp": "2025-06-15T16:05:48.239572Z",
  "utxo": {
    "369b636feb65e872d8013afded63968685be6caab30a353d074a3b0b7cf9bec4#0": {
      "address": "addr_test1vr477k08flhq7j78cfzkap996kmxa4nq8jkcs2xkqae6xeqxf3hq2",
      "datum": null,
      "datumhash": null,
      "inlineDatum": null,
      "inlineDatumRaw": null,
      "referenceScript": null,
      "value": {
        "lovelace": 1000000000
      }
    }
  }
}
```

```
{"timestamp":"2025-06-15T16:06:16.565808Z","threadId":106,"namespace":"HydraNode-\"alice-node\"","message":{"reliability":{"acknowledged":[0
```

```
{"timestamp":"2025-06-15T16:06:16.56496Z","threadId":100,"namespace":"HydraNode-\"bob-node\"","message":{"reliability":{"acknowledged":[0,0]
```


websocat ws://127.0.0.1:4001 | jq
⌥⌘2
websocat ws://127.0.0.1:4002 | jq
⌥⌘6
make alice-node
⌥⌘4
make bob-node
⌥⌘5

```
        "datum": null,
        "datumhash": null,
        "inlineDatum": null,
        "inlineDatumRaw": null,
        "referenceScript": null,
        "value": {
          "lovelace": 1000000000
        }
      }
    }
  },
  "postTxError": {
    "failureReason": "TxValidationErrorInCardanoMode (ShelleyTxValidationError ShelleyBasedEraConway (ApplyTxError (ConwayUtxowFailure (Extr
aRedeemers [ConwaySpending (AsIx {unAsIx = 0}),ConwaySpending (AsIx {unAsIx = 1}),ConwaySpending (AsIx {unAsIx = 2})]) :| [ConwayUtxowFailur
e (PPViewHashesDontMatch (Mismatch {mismatchSupplied = SJust (SafeHash \"13816ad1aec2685928d683f58c909f0f744372a33b3dd64f6b9247e453dd7733\")
, mismatchExpected = SJust (SafeHash \"88a696b928b3b2c61a66438dfb459433baad211a1caf76be67da3215ad7474eb\")})),ConwayUtxowFailure (UtxoFailur
e (ValueNotConservedUTxO (Mismatch {mismatchSupplied = MaryValue (Coin 991931446) (MultiAsset (fromList [])), mismatchExpected = MaryValue (
Coin 2996250066) (MultiAsset (fromList [(PolicyID {policyID = ScriptHash \"644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481f6097ac6\"},fromL
ist [(\"3e5d1b54cd8a72173a63ad46a6720f1c16423b37b81359753d599cc0\",1),(\"4879647261486561645631\",1),(\"feb29491a0bd0c4f0b2fc257a9f949c30873
65a4e3648b89c9d19959\",1)])])))})),ConwayUtxowFailure (UtxoFailure (BadInputsUTxO (fromList [TxIn (TxId {unTxId = SafeHash \"056a581ecfb7fd5
705b2804f4d951dce554da393609d68049d3147eaf9ac1394\"}) (TxIx {unTxIx = 0}),TxIn (TxId {unTxId = SafeHash \"5df882e10f05c7b6ed986b1e682eba1bed
358af76adc22054652f25c86c533e1\"}) (TxIx {unTxIx = 0}),TxIn (TxId {unTxId = SafeHash \"91b75154a597f4525538685674990b59068ec9df7f9c5aa7fd6
c0d622af565\"}) (TxIx {unTxIx = 0})])))]))))",
    "tag": "FailedToPostTx"
  },
  "seq": 4,
  "tag": "PostTxOnChainFailed",
  "timestamp": "2025-06-15T16:06:53.360444Z"
}
```

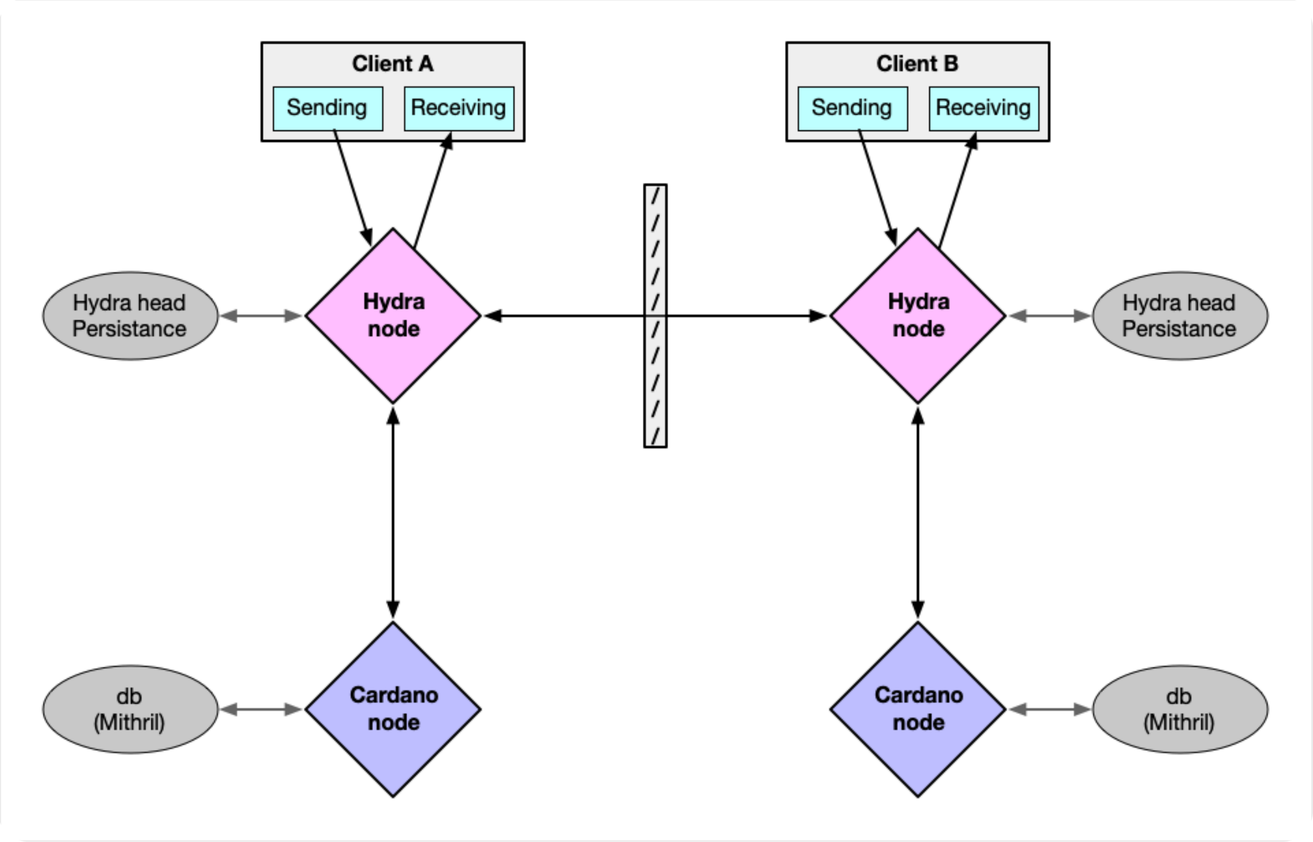
```
      "referenceScript": null,
      "value": {
        "lovelace": 1000000000
      }
    }
  }
}
{
  "headId": "644ba9919d7d5dd99dda738c92546f436d484dc4eea8e481f6097ac6",
  "party": {
    "vkey": "0f823515a31668ce8c54e705fa18fa2c827c00fb3fcb5494bce320355d3dc363"
  },
  "seq": 3,
  "tag": "Committed",
  "timestamp": "2025-06-15T16:06:52.289609Z",
  "utxo": {
    "369b636feb65e872d8013afded63968685be6caab30a353d074a3b0b7cf9bec4#2": {
      "address": "addr_test1vrlcr9eaa9l5v6z34md80k5q8ksnkmcehzc88d327hwsv3q9mucxz",
      "datum": null,
      "datumhash": null,
      "inlineDatum": null,
      "inlineDatumRaw": null,
      "referenceScript": null,
      "value": {
        "lovelace": 1000000000
      }
    }
  }
}
```

```
a8e481f6097ac6\"},fromList [(\"3e5d1b54cd8a72173a63ad46a6720f1c16423b37b81359753d599cc0\",1),(\"4879647261486561645631\",1),(\"feb29491a0
```

```
{"timestamp":"2025-06-15T16:06:52.358164Z","threadId":69,"namespace":"HydraNode-\"bob-node\"","message":{"node":{"by":{"vkey":"0f823515a316
```

Hydra PoC Remote

In the next PoC, we will see how to run two Hydra nodes on separate remote computers.



Setting up the Cardano Nodes

This part of the process is same as what explained for the Hydra PoC Local, but the cardano nodes are being set up on different computers.

Setup Networking

Each of the parties has to find their external ip address, and choose a port to use for communicating with the peer.

```
# find external ip address
> curl -4 ifconfig.me
78.55.20.191% # My public IP
```

Adding Hydra Nodes

Once the credentials are exchanged we will need to run different commands, in order to setup the credentials and establish the communication between the remote nodes.

```
make username-credentials
make setup-funding-wallet
make fund-username
```