Exploring The Demo

Canton Network Quickstart Guide | 2025

Version: 1.0.0-2025-02-13

Contents

Exploring the Demo

The CN-QS and its guides are a work-in-progress (WIP). As a result, the CN-QS guides may not accurately reflect the state of the application. If you find errors or other inconsistencies, please contact your representative at Digital Asset.

This section works through a complete business operation within the CN-QS.

Prerequisites

You should have successfully installed the CN-QS before beginning this demonstration.

Access to the [CN-Quickstart Github repository](#) and [CN Docker repository](#) is needed to successfully pull the Digital Asset artifacts from JFrog Artifactory.

Access to the *Daml-VPN* connection or [a SV Node](#) that is whitelisted on the CN is required to connect to `DevNet`. The GSF publishes a [list of SV nodes](#) who have the ability to sponsor a Validator node. To access `DevNet`, contact your sponsoring SV agent for VPN connection information.

If you need access, email support@digitalasset.com.

The CN-QS is a Dockerized application and requires [Docker Desktop](#). Running CN-QS on `LocalNet` is resource intensive. It is recommended to allocate 32 GB of memory and 3 GB of Swap memory to properly run the required Docker containers. If you witness unhealthy containers, please consider allocating additional resources, if possible.
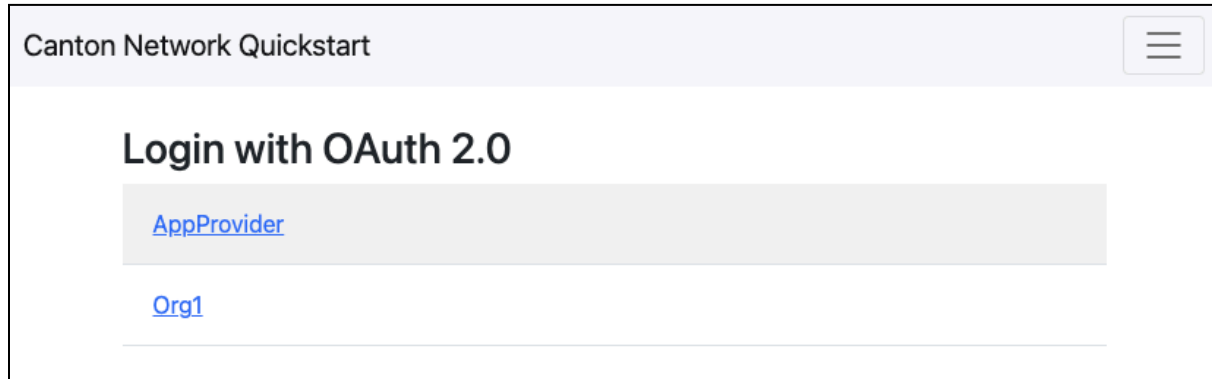
`DevNet` is not as intensive because the SVs and other `LocalNet` containers are hosted outside of your local machine.

Walkthrough

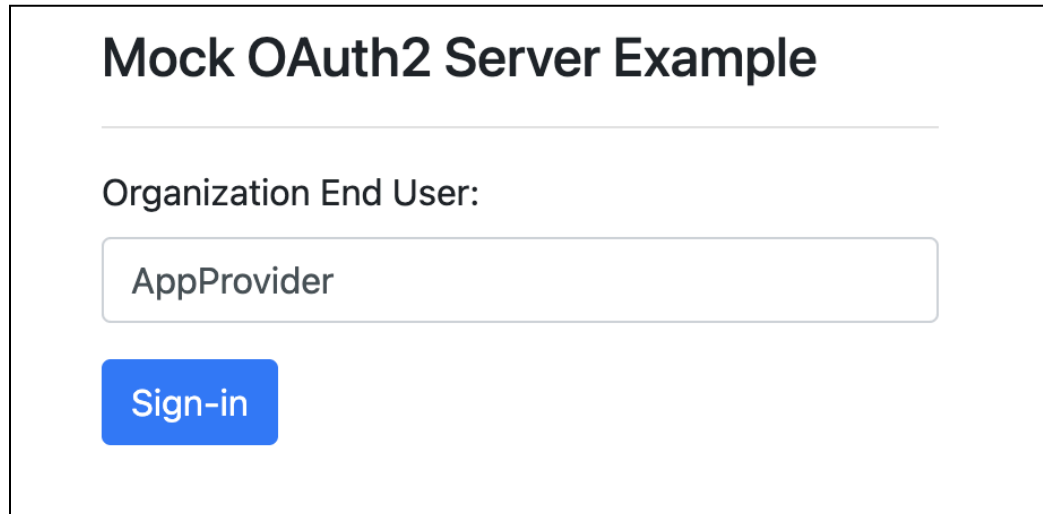After the QS is installed and running, confirm that you are in the `quickstart` subdirectory of the CN-QS.

Open a new incognito browser.

Navigate to `localhost:3000/login` or run `make open-app-ui` in the terminal.

Login as the AppProvider.

Select "AppInstallRequests" in the menu.

| Canton Network Quickstart | Home | AppInstallRequests | AppInstalls | Licenses | LicenseRenewalRequests | OAuth2 | **AppProvider** | Logout |

Note: Run `make create-app-install-request` to submit an AppInstallRequest from the AppUser participant.

App Install Requests

Contract ID	DSO	Provider	User	Meta	Actions

Open a terminal.
From `/quickstart/` run `make create-app-install-request` to create an App Installation Request on behalf of the Participant.

```
[(base) quickstart ~ % make create-app-install-request
docker compose -f docker/app-user-shell/compose.yaml --env-file .env run --rm create-app-install-request ||
 true
get_token ledger-api-user AppProvider
get_user_party AppProvider participant-app-provider
http://participant-app-provider:7575/v2/users/AppProvider
get_token ledger-api-user Org1
get_user_party Org1 participant-app-user
http://participant-app-user:7575/v2/users/Org1
get_token administrator Org1
http://validator-app-user:5003/api/validator/v0/scan-proxy/dso-party-id
http://participant-app-user:7575/v2/commands/submit-and-wait
--data-raw {
        "commands" : [
          { "CreateCommand" : {
              "template_id": "#quickstart-licensing:Licensing.AppInstall:AppInstallRequest",
              "create_arguments": {
                  "dso": "DSO::12209a3af80af3fa93853be8a8b9f5887055edc3b0a94f4b198f486d08d197784c09",
                  "provider": "AppProvider::1220b3de80de523473aa2ca56745ef1fb29a2203dfd13029e0448336bbcf3
82aaf86",
                  "user": "Org1::1220e69bc6115cd8ed360aa6caafce122c2a24561262c2f6ce6a6070e170e9e8244d",
                  "meta": {"values": []}
              }
            }
          }
        ],
        "workflow_id" : "create-app-install-request",
        "application_id": "ledger-api-user",
        "command_id": "create-app-install-request",
        "deduplication_period": { "Empty": {} },
        "act_as": ["Org1::1220e69bc6115cd8ed360aa6caafce122c2a24561262c2f6ce6a6070e170e9e8244d"],
        "read_as": ["Org1::1220e69bc6115cd8ed360aa6caafce122c2a24561262c2f6ce6a6070e170e9e8244d"],
        "submission_id": "create-app-install-request",
        "disclosed_contracts": [],
        "domain_id": "",
        "package_id_selection_preference": []
    }
{"update_id":"1220aebcd64fc960c2aeec834d415967899994f21a69ef9cab5639b8521169a0ca67","completion_offset":81}
```

If your machine is not powerful enough to host `LocalNet` or if the docker containers are not responsive then the response may show a failure with status code `404` or `000`. Increasing Docker memory limit to at least 32 GB should allow the `LocalNet` containers to operate properly.

```
(base) quickstart ~ % make create-app-install-request
docker compose -f docker/app-user-shell/compose.yaml --env-file .env run --rm create-app-install-request || true
[+] Building 0.0s (0/0)                                                    docker:desktop-linux
[+] Building 0.0s (0/0)                                                    docker:desktop-linux
get_token ledger-api-user AppProvider
get_user_party AppProvider participant-app-provider
http://participant-app-provider:7575/v2/users/AppProvider
get_token ledger-api-user Org1
get_user_party Org1 participant-app-user
http://participant-app-user:7575/v2/users/Org1
get_token administrator Org1
http://validator-app-user:5003/api/validator/v0/scan-proxy/dso-party-id
Request failed with HTTP status code 404
Response body: The requested resource could not be found.
```

The install request appears in the list.

Canton Network Quickstart	Home AppInstallRequests AppInstalls Licenses LicenseRenewalRequests OAuth2	AppProvider Logout

Note: Run `make create-app-install-request` to submit an AppInstallRequest from the AppUser participant.

App Install Requests

Contract ID	DSO	Provider	User	Meta	Actions
00a2400ea9a6...	DSO::12209a3af...	AppProvider::12...	Org1::1220e69b...	{"data":{}}	Actions

Select "Actions."

App Install Requests

Contract ID	DSO	Provider	User	Meta	Actions
00a2400ea9a6...	DSO::12209a3af...	AppProvider::12...	Org1::1220e69b...	{"data":{}}	Accept Reject Cancel

Select "Accept" and the request moves to the "AppInstalls" view.

Select "Actions" and then select "Create License."

The License is created.

Notice in the screenshot above that the User field is set to "Org1."
At this time, the license also appears in Org1's view.

In the AppProvider account, navigate to the Licenses menu and select "Actions."

Licenses

Existing Licenses

Contract ID	DSO	Provider	User	Params	Expires At	License Num	Actions
0050da1bc6f07e8605...	DSO::12209a3af...	AppProvider::12...	Org1::1220e69b...	{"meta":{"data":{}}}	2025-02-11T11:23:19.5...	1	Actions

An "Actions for License" modal opens with an option to renew or expire the license. Per the Daml contract, licenses are created in an expired state. To activate the license, it must be renewed.

Actions for License 0050da1bc6f07e860545cf65

Renew License

Extension: 30 days (P30D), **Payment Acceptance:** 7 days (P7D), **Fee:** 100 CC

Description:

> e.g. "Renew for next month"

Renew

Expire License

Description:

> e.g. "License expired"

Expire

Close

To renew the license, enter a description then click the green "Renew" button.

The license renewal process is initiated.

Navigate to the License Renewal Requests menu.

Clicking "Complete Renewal" reveals that the license must be paid for in order to be renewed.

Return to `localhost:3000/login` in an incognito browser.

Login as an Org1 user.

Go to the LicenseRenewealRequests View and click the "Pay" button.

Canton Network Quickstart · Home · AppInstallRequests · AppInstalls · Licenses · LicenseRenewalRequests · Org1 · Logout

License Renewal Requests

Contract ID	Provider	User	DSO	License Number	Fee (CC)	Extension Duration	Actions
00c9285b226956419d...	AppProvider::12...	Org1::1220e69b...	DSO::12209a3af...	1	100	30 days	Pay

Clicking "Pay" redirects to the Canton Coin (CC) Wallet at `wallet.localhost:2000`.
Log in with OAUTH2.
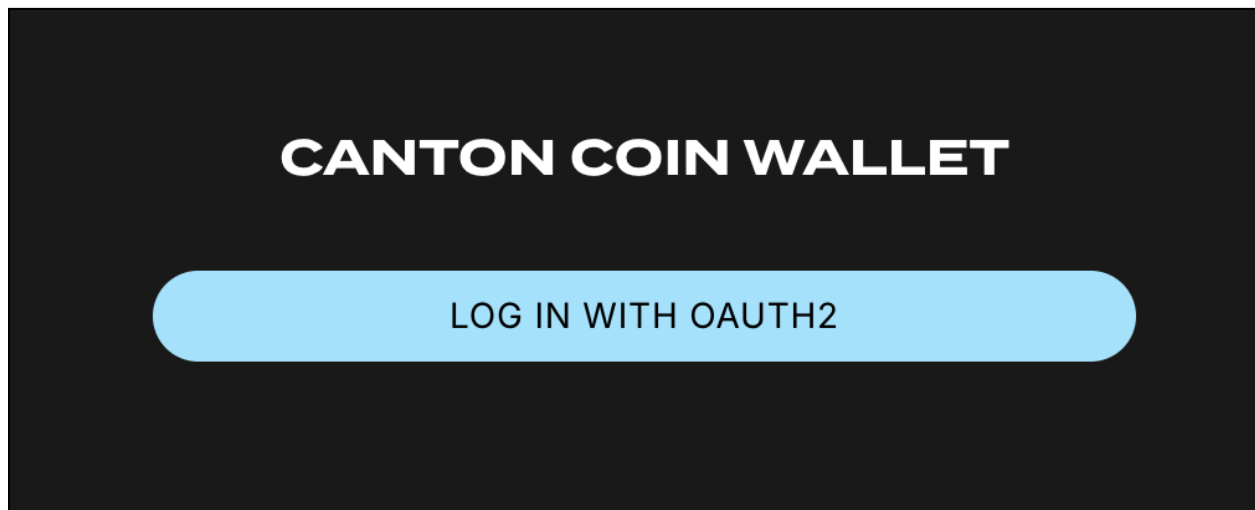
Logging in with OAUTH2 directs to a payment view. However, the total available balance is 0 CC. The Renewal fee is 100 CC.

Open CC Wallet from the terminal. In the `quickstart/` directory, use `make open-sv-wallet`.
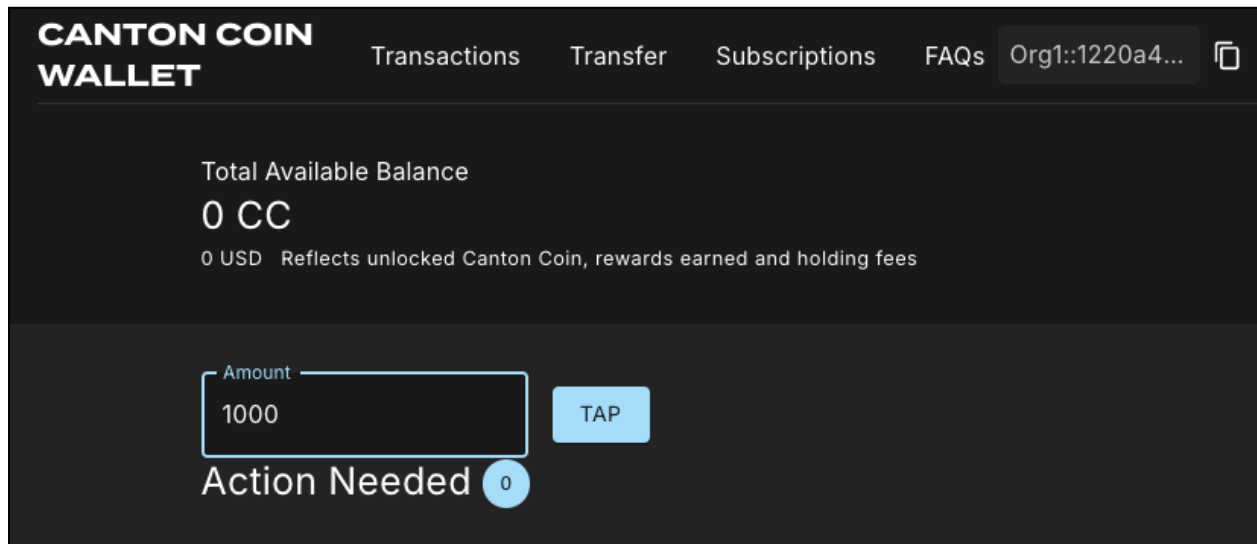
Populate the wallet with CCs.
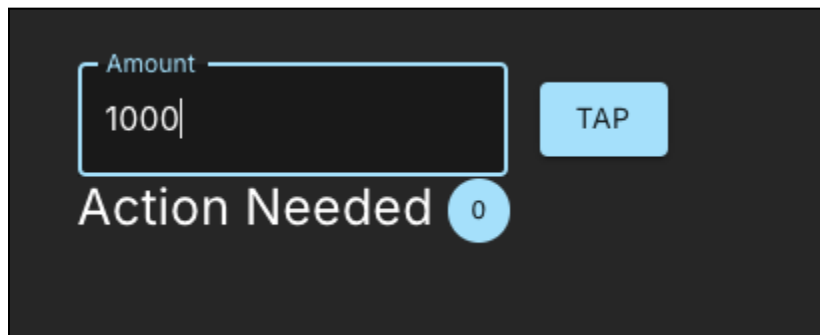
Click "TAP." `LocalNet` recognizes the update in your wallet's value, as well as the total available CCs in the CN.

Return to the License Renewal Request as Org1. Click Pay.

The CC Wallet balance is now sufficient to send payment to the Provider.

Return to the AppProvider's License Renewal Requests View.
The AppProvider may now Complete the Renewal.

Clicking "Complete Renewal" results in a Success.

Org1's Licenses view shows the activated license.

Congratulations. You've successfully created and activated a license with a payment transfer!

Canton Console

The Canton Console connects to the running application ledger. The console allows a developer to bypass the UI to interact with the CN in a more direct manner. For example, in Canton Console you can connect to the Participant to see the location of the Participant and their domain.

Activate the Canton Console in a terminal from the `quickstart/` directory. Run:

```
make console
```

After the console initiates, run the participant and participant.domains commands, respectively.

`participant` returns their location in the ledger.
`participant.domains` shows the Participant's domain.

```
@ participant
res0: com.digitalasset.canton.console.RemoteParticipantReference =
Participant 'participant'

@ participant.domains
res1: participant.domains.type =
com.digitalasset.canton.console.commands.ParticipantAdministration$do
mains$@2da16d31
```

Daml Shell

The Daml Shell connects to the running PQS database of the application provider's Participant. In the Shell, the assets and their details are available in real time.

Run the following commands to see the data:

`active` - shows unique identifiers and the asset count

`active quickstart-assets:Assets.Main:Asset` - shows the asset details

`archives quickstart-assets:Assets.Main:Asset` - shows any archived assets

Connect to DevNet

Stop the `LocalNet` containers to change the connection from `LocalNet` to `DevNet`.

In the terminal, run:

```
make stop
make clean-all
```

To edit the connection and observability parameters run:

```
make setup
```

When prompted to enable `LocalNet`, enter "n". This enables `DevNet`

Optionally, enter "Y" to enable observability. This starts additional containers which may require more memory for Docker.

```
(base) quickstart ~ % make setup
Starting setup...
Enable LocalNet? (Y/n): n
  LOCALNET_ENABLED set to 'false'.

Enable Observability? (Y/n): Y
  OBSERVABILITY_ENABLED set to 'true'.

.env.local updated successfully.
```

The application is now connected to `DevNet`.

In `DevNet` mode, you can configure a non-default `SPONSOR_SV_ADDRESS`, `SCAN_ADDRESS` and `ONBOARDING_SECRET_URL` or `ONBOARDING_SECRET` in the `quickstart/.env` file.

Connecting to `DevNet` requires a connection to an [approved SV](#). If your organization provides access to the DAML-VPN, then connect to it to access the Digital Asset-sponsored SV. Your organization may sponsor another CN-approved SV. If this is the case, speak with your administrator for privileged access. Review the DevNet Global Synchronizer documentation to learn more about the [SV onboarding process](#).

In an incognito browser navigate to `localhost:3000/login`. Login as the `Org1` user and create and archive assets, as before. Logout and do the same as the `AppProvider`.

SV UIs

Navigate to `localhost:4000` for a list of viewable Web UI options.
Click "Visit Super Validator Web UI" to view information from the validator.

Super Validator UIs

Wallet Web UI

Interface for managing user wallets.

Visit Wallet Web UI

Super Validator Web UI

Interface to manage Super Validator functionalities.

Visit Super Validator Web UI

Scan Web UI

Interface to monitor transactions.

Visit Scan Web UI

Login as 'administrator'.
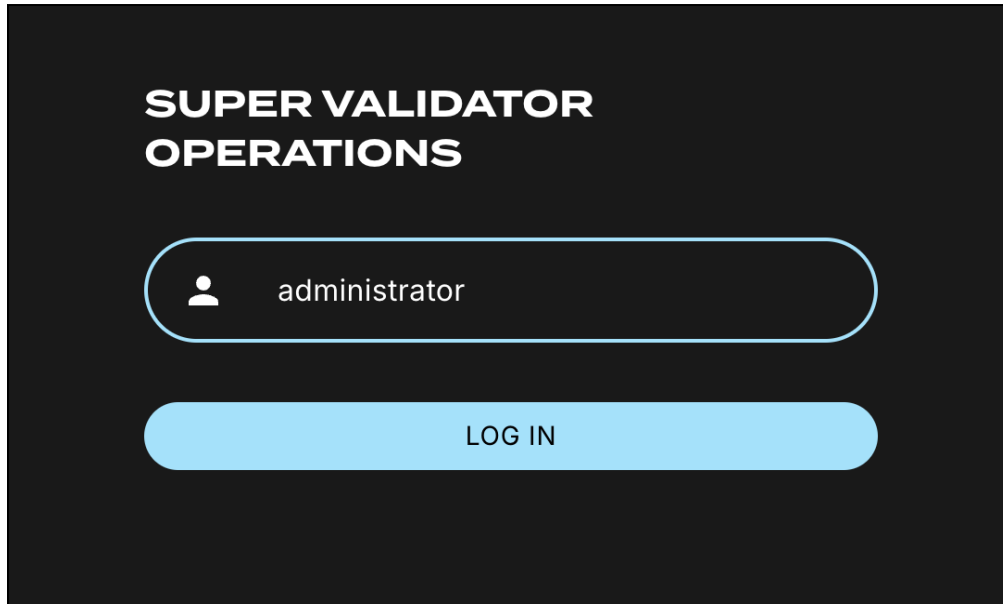
The UI shows information about the SV and lists the active SVs.
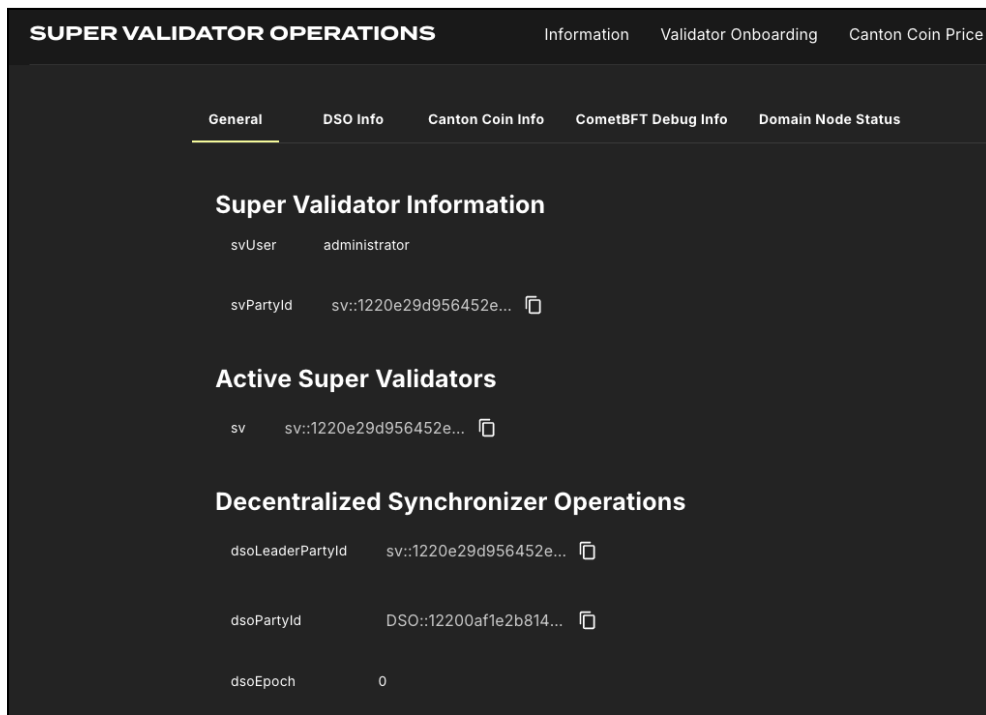
The Validator Onboarding menu allows for the creation of validator onboarding secrets.

The CC Price menu option has an option to set the price for open mining rounds.

Update the price of the CC.

The updated coin price reflects the new open mining rounds.

Navigate to the CC Scan Web UI by returning to `localhost:4000` and select "Visit Scan Web UI" or navigate directly to http://scan.localhost:4000/.

The activity view shows the total CC balance and the Validator rewards.

Select the Network Info menu to view SV identification.

The Validators menu shows that the local validator has been registered with the SV.

Observability Dashboard

In a web browser, navigate to http://localhost:3030/dashboards to view the observability dashboards. Select "Quickstart - consolidated logs".
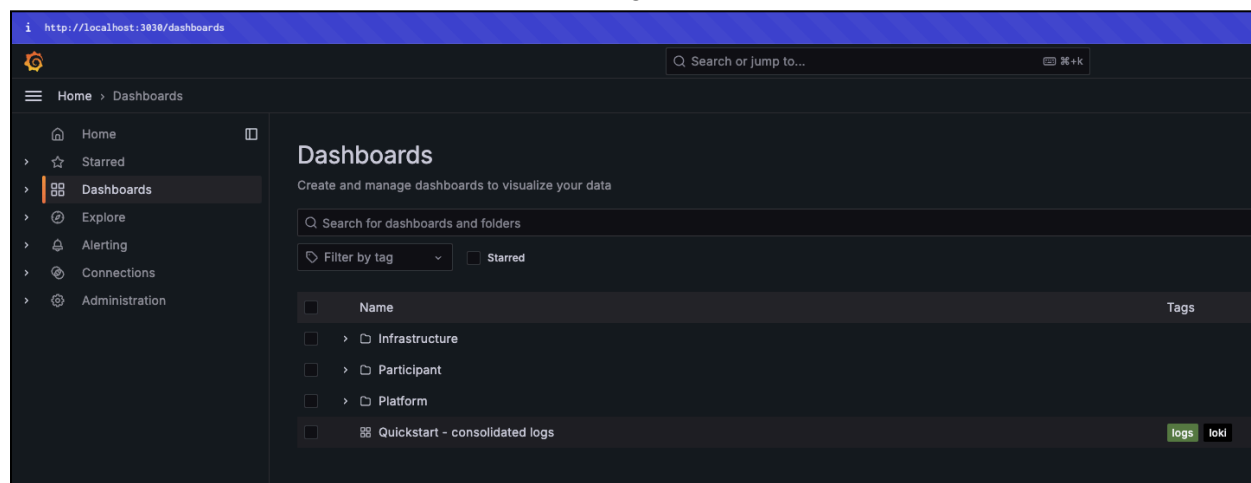
The default view shows a running stream of all services.

Change the services filter from "All" to "participant" to view participant logs. Select any log entry to view its details.

Next Steps

You've completed a business operation in the CN-QS and have been introduced to the basics of the Canton Console, Daml Shell, and the SV UIs.

Learn more about Daml Shell and the project structure in the Project Structure Guide.