1**) Install Hyperledger Firefly CLI**

FireFly CLI

The FireFly CLI can be used to create local FireFly stacks for offline development of blockchain apps. This allows developers to rapidly iterate on their idea without needing to set up a bunch of infrastructure before they can write the first line of code.

## Prerequisites

In order to run the FireFly CLI, you will need a few things installed on your dev machine:

* [Docker](https://www.docker.com/)
* [Docker Compose](https://docs.docker.com/compose/)
* Openssl
* Aws machine AMI should be upto 8GB RAM, or t2.medium, t2.large type.

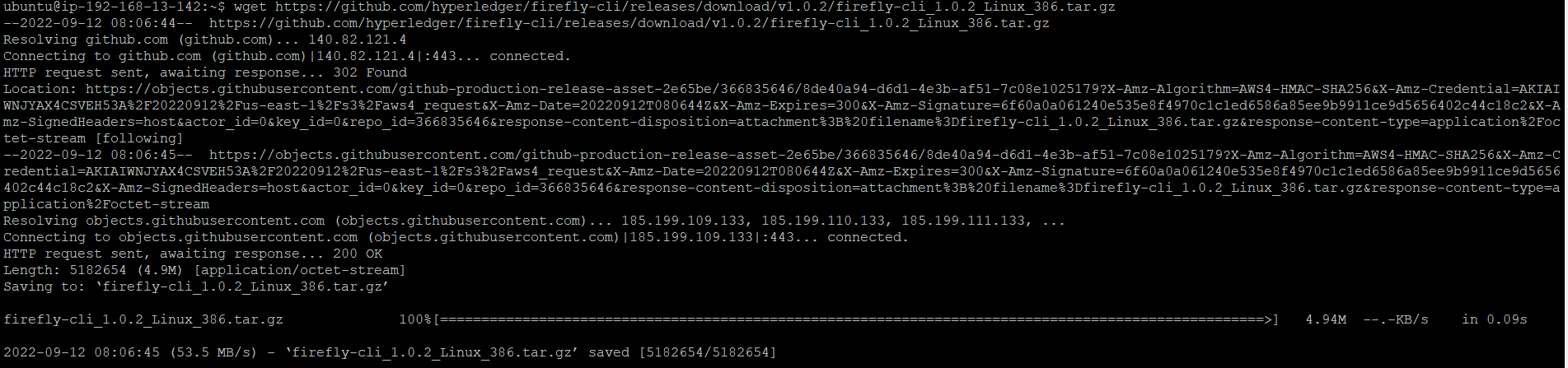
### **For Linux Users**

For Linux users, it is recommended that you add your user to the docker group so that you do not have to run ff or docker as root or with sudo.

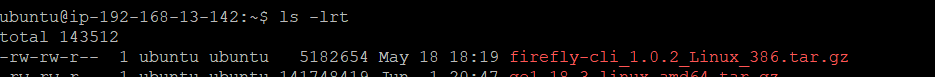
### **Download the package for your OS**

On ubuntu terminal use below command to download the package for your os.

wget <https://github.com/hyperledger/firefly-cli/releases/download/v1.0.2/firefly-cli_1.0.2_Linux_386.tar.gz>



once package is downloaded then extract the package

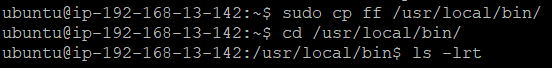


sudo tar -xzvf firefly-cli\_1.0.2\_Linux\_386.tar.gz

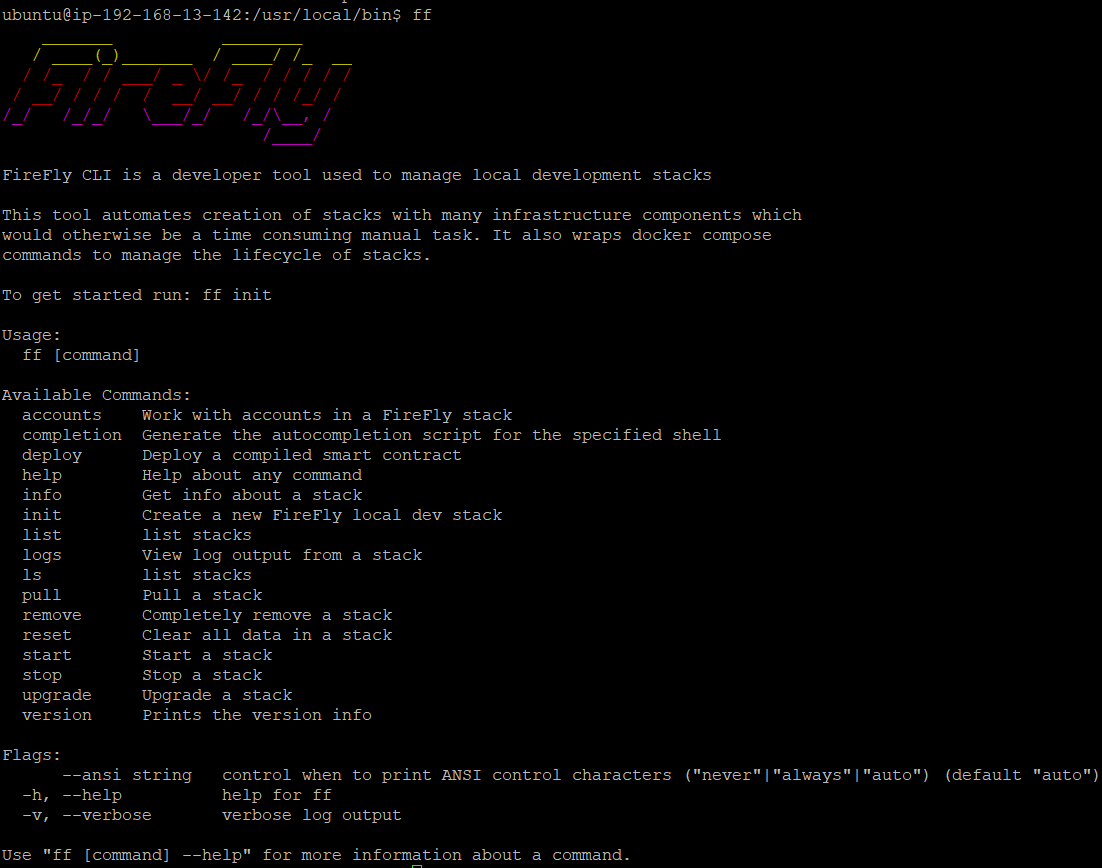
once the package is extracted then copy ff at the location /usr/bin/local

sudo cp ff /usr/local/bin/

go to the directory cd /usr/local/bin/

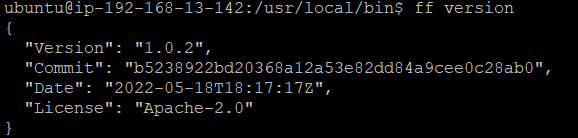


To check the firefly installation is completed of not just enter a command ‘ff ‘you will see the big icon of firefly as below snapshot.



**Verify the installation**

After using either installation method above, you can verify that the CLI is successfully installed by running ff version. This should print the current version like this:



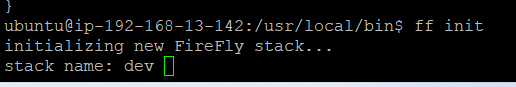
**2)  environment set up**

## Creating a new stack

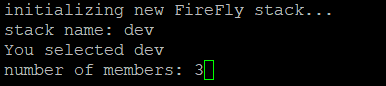
It’s really easy to create a new FireFly stack. The ff init command can create a new stack for you, and will prompt you for a few details such as the name, and how many members you want in your stack.

Run: ff init

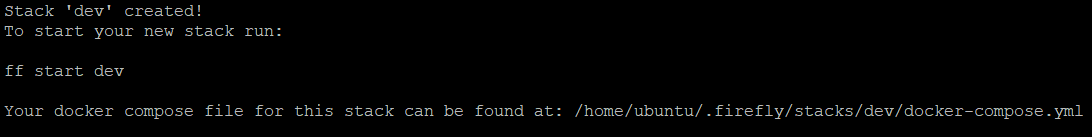
Choose a stack name. For this guide, I will choose the name dev, but you can pick whatever you want



Chose the number of members for your stack. For this guide, we have pick 3 members, so we can try out both public and private messaging use cases:



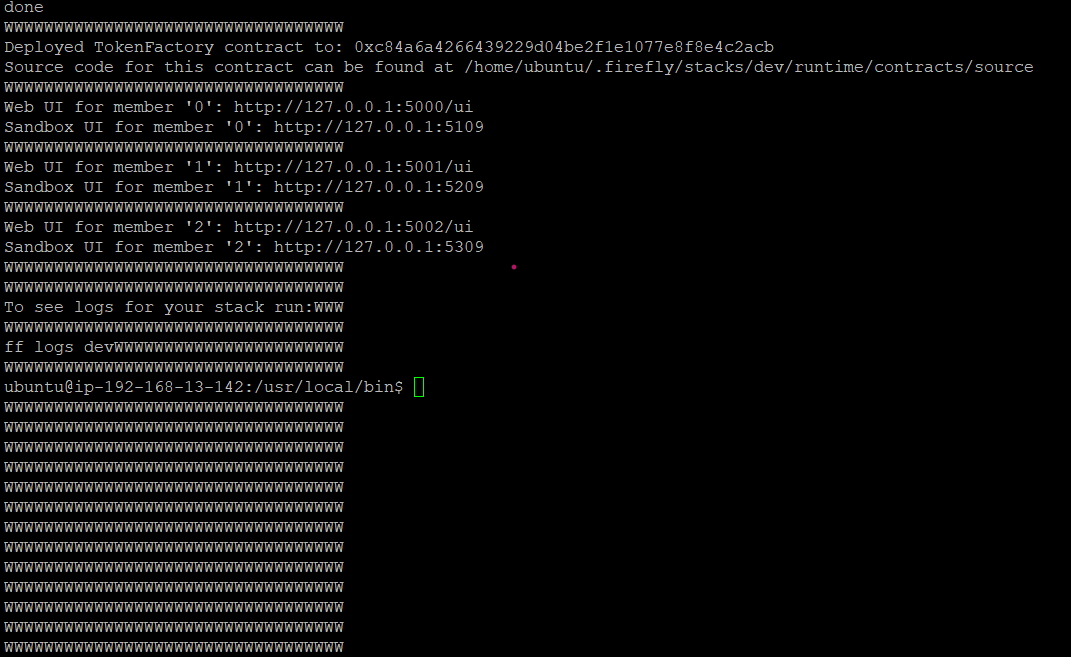
### **Start your stack**



This may take a minute or two and in the background the FireFly CLI will do the following for you:

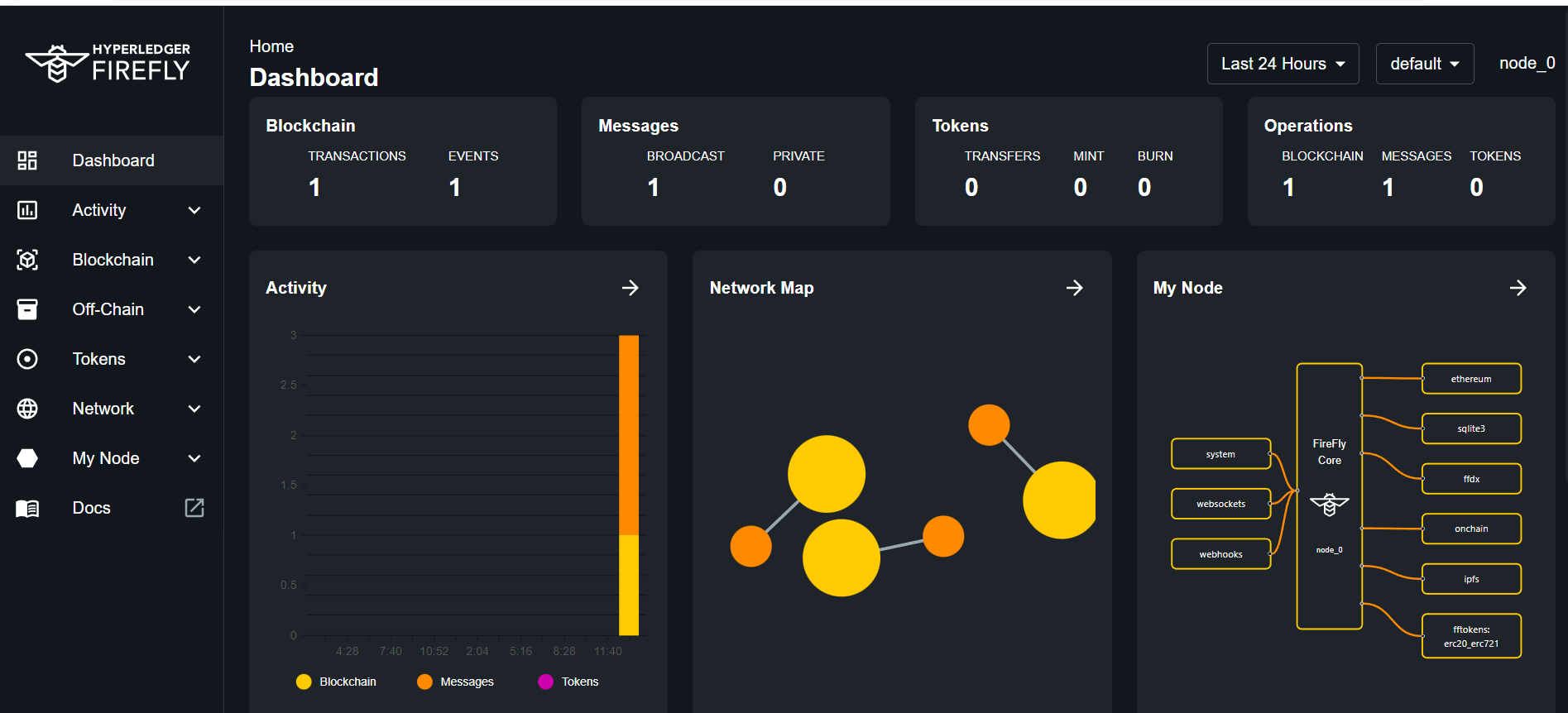
* Download Docker images for all of the components of the Supernode
* Initialize a new blockchain and blockchain node running inside a container
* Set up configuration between all the components
* Deploy FireFly’s BatchPin smart contract
* Deploy an ERC-1155 token smart contract
* Register an identity for each member and node

After your stack finishes starting it will print out the links to each member’s UI and the Sandbox for that node as below snapshot.



Now replace the http://127.0.0.1:5000/ui with your machine ip and access in browser you will see the firefly explorer of member 0 . In My case you can see the snapshot as below

Firefly explorer for member 0

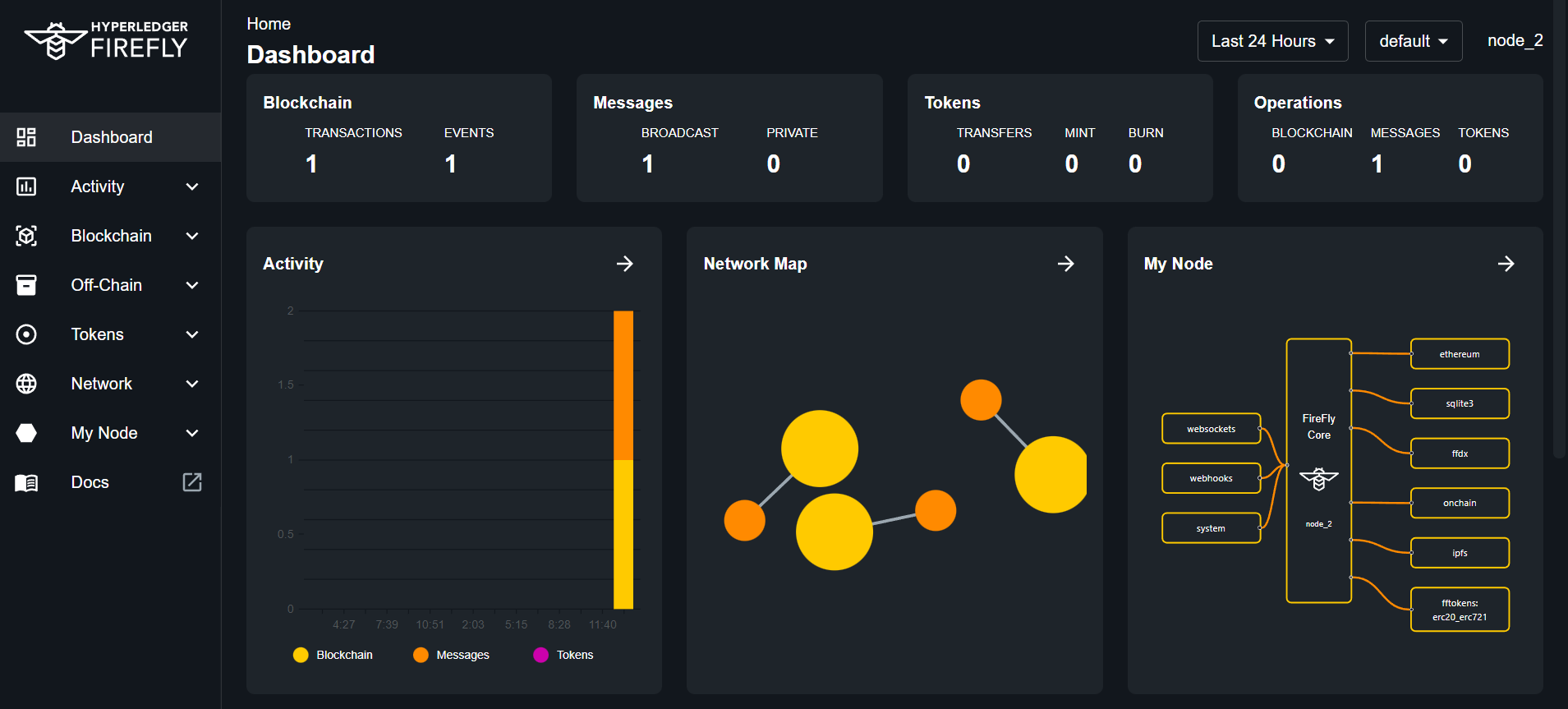


Do similarly for each member 1 and member and access the firefly explorer on browser as above.

Firefly explorer for member 1



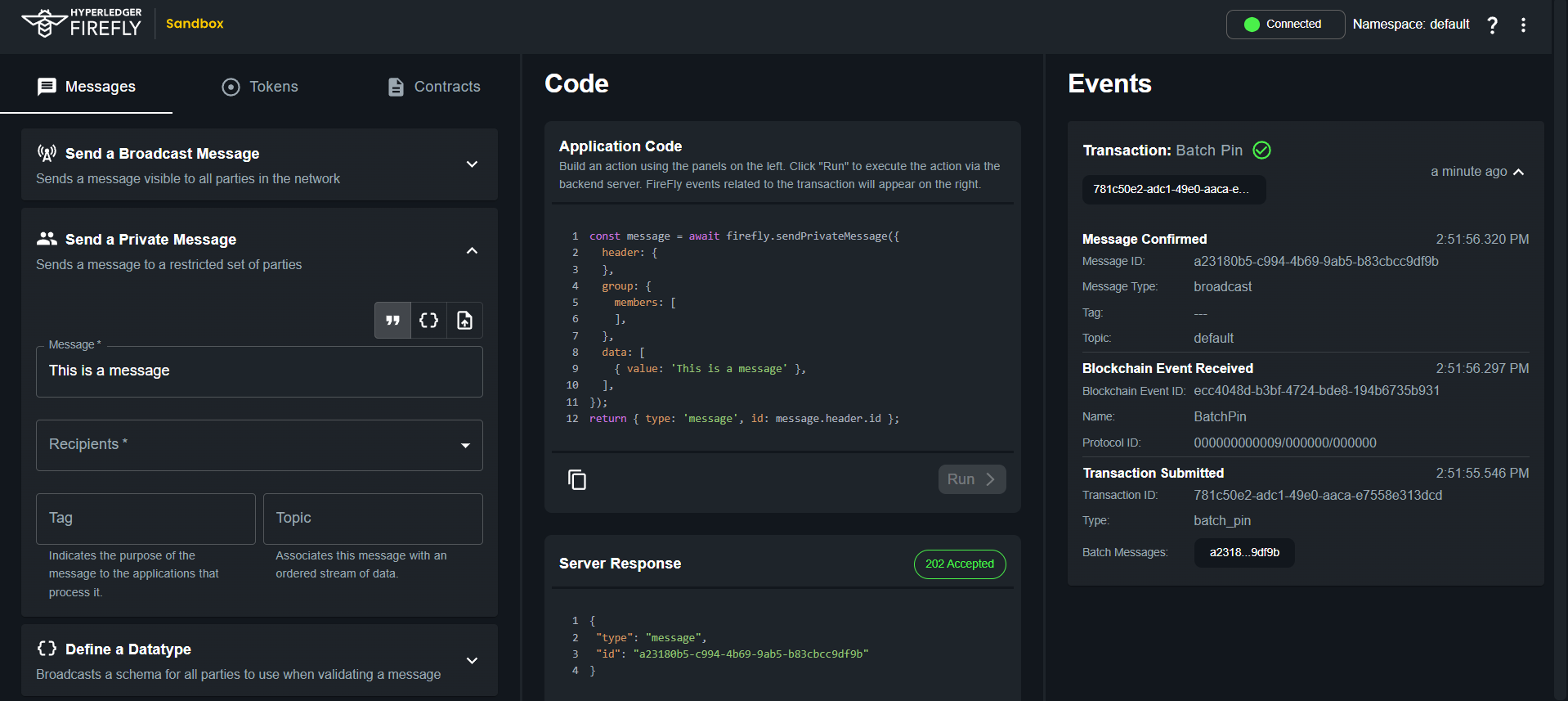
Firefly explorer for member 2



# **3) Use the Sandbox**

The FireFly Sandbox sits logically outside the Supernode, and it acts like an “end-user” application written to use FireFly’s API. In our setup, we have one Sandbox per member, each talking to their own FireFly API. The purpose of the Sandbox is to provide a quick and easy way to try out all of the fundamental building blocks that FireFly provides. It also shows developers, through example code snippets.

**Sandbox UI for member0**

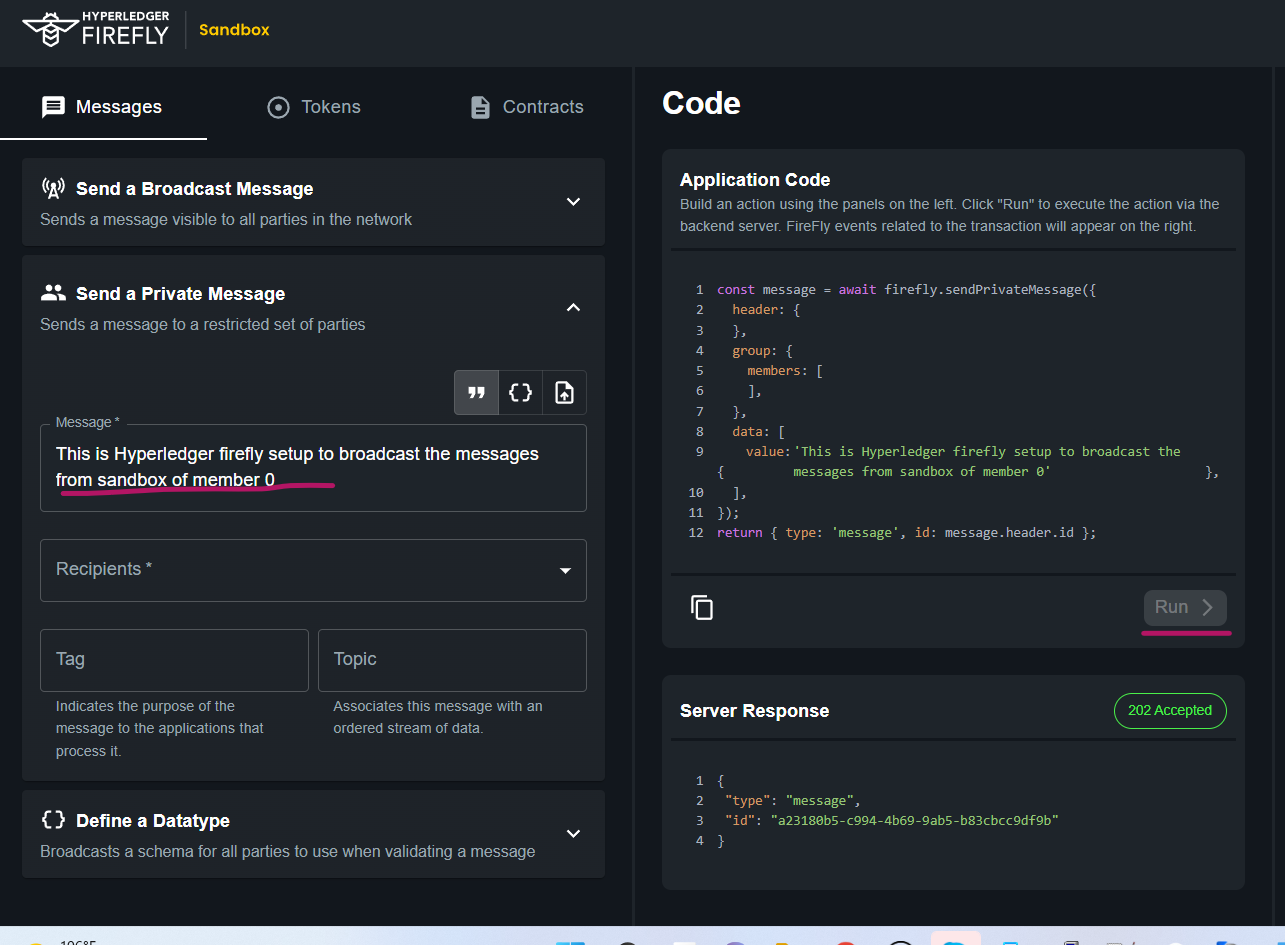


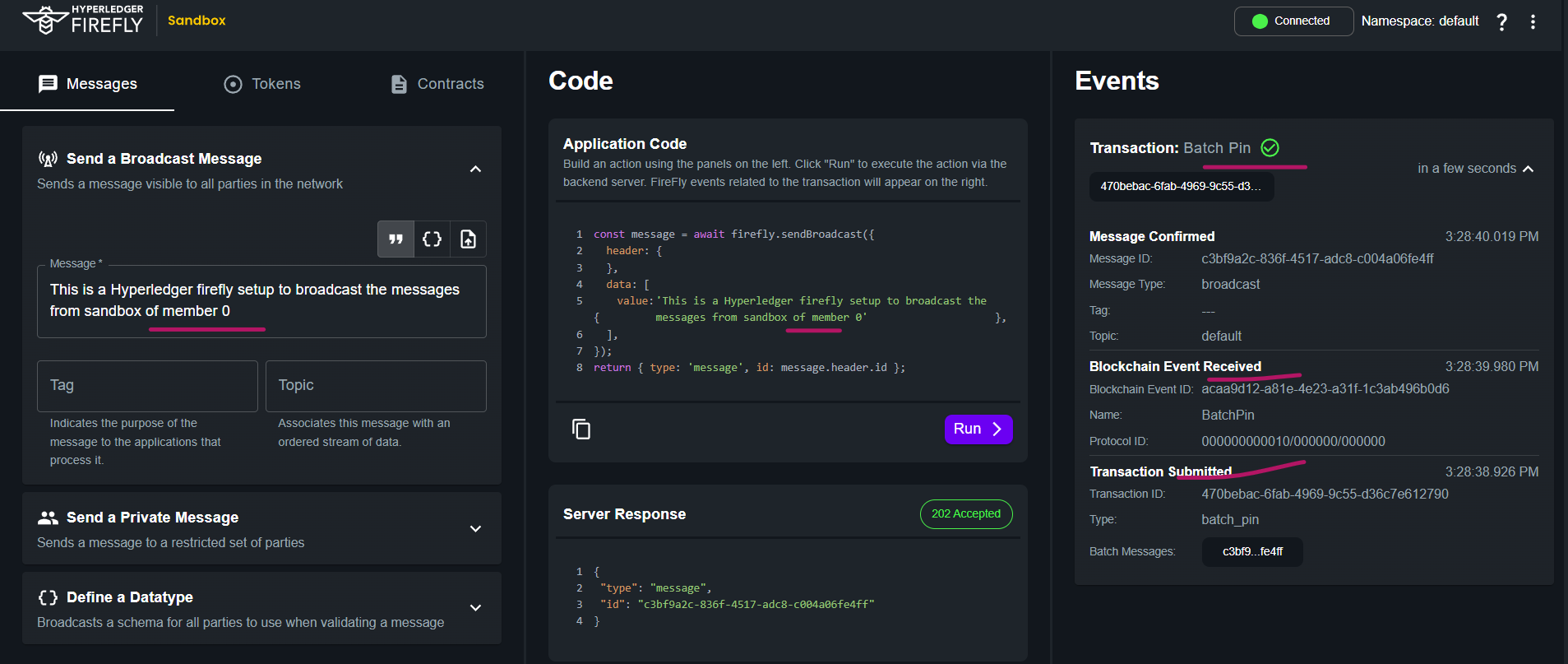
### **Things to try out**

* Send a broadcast message and view the data payload in every member’s FireFly Explorer
* Send a private message to one member, and verify that the data payload is not visible in the third member’s FireFly Explorer
* Send an image file and download it from another member’s FireFly Explorer

Let us send the message from sandbox UI of member 0 and check the data payload in every member’s FireFly explorer reflected on not.

Type the message at messages Tab and click on Run button.





Here we have broadcasted the message from sandbox of member 0 and message has been received on each organization firefly explorer as below snapshots.

