Hyperledger Explorer

Introduction

Hyperledger Explorer is a tool for visualizing blockchain operations of the Hyperledger Fabric platform.

Hyperledger Explorer is an open source blockchain utility module that allows users to create a user-friendly web-based application, with which a user can view, initiate, organize or query various artifacts and developments that form an integral part of the blockchain network. It is to be used specifically on deployments of blockchains created using the Hyperledger umbrella.

Features

- Get the latest status blocks, network, and chaincodes, view blocks, and transactions.
- Blocks and transaction metrics by hours, and minutes.
- Search, and filter blocks, transactions by date range and channels.
- Dynamically discover new channels and switch data presentation by channels.
- Get real time notification of new blocks.

Working

Hyperledger Explorer's architecture includes a web server that runs in the backend and is responsible for interacting with all other components and maintaining the necessary query-server response. The web sockets are used to communicate between the server and the various client components of Hyperledger Explorer. A PostgreSQL database is used to store the necessary details about blockchain components like information about blocks, transactions, and smart contracts, and this can be queried for any necessary information. A security repository takes care of ensuring only secure and authorized access is maintained for accessing the Hyperledger Explorer.

Hyperledger Explorer allows for a unified enterprise-level visualization, which may be needed in real-time by a blockchain developer developing a particular feature or component on the blockchain, or by a researcher seeking to study historical developments, or by blockchain operators who are responsible for managing the blockchain, or by top management.

Prerequisites

- Docker
- Docker Compose
- fabric samples

1) Start the network:

```
cd fabric-samples/test-network
./network.sh up createChannel
```

Deploy Chaincode

```
./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-javascript -ccl javascript
```

Add fabric peer binaries to our CLI Path:

```
export PATH=${PWD}/../bin:$PATH
```

set the FABRIC_CFG_PATH to point to the core.yaml file in the fabric-samples repository:

```
export FABRIC_CFG_PATH=$PWD/../config/
# Environment variables for Org1
export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org1MSP"
export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
export CORE_PEER_ADDRESS=localhost:7051
```

2) Run the following command to initialize the ledger with assets.

```
  peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls
--cafile"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlsc
acerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051
--tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt"
--peerAddresses localhost:9051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c
'{"function":"InitLedger","Args":[]}'
```

3) query the ledger from your CLI

```
peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'
```

Chaincodes are invoked when a network member wants to transfer or change an asset on the ledger.

```
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/t
lsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt"
--peerAddresses localhost:9051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c
'{"function":"TransferAsset","Args":["asset6","Christopher"]}'
```

HyperLedger Explorer

1.Create a new directory (e.g. explorer) in the root directory/ where fabric-sample is present

```
mkdir explorer
```

```
cd explorer
```

Copy the following files from the repository

- docker-compose.yaml
- examples/net1/connection-profile/test-network.json
- examples/net1/config.json

```
wget https://raw.githubusercontent.com/hyperledger/blockchain-explorer/main/examples/net1/config.json
```

```
wget https://raw.githubusercontent.com/hyperledger/blockchain-explorer/main/examples/net1/connection-profile/test-network.json -P connection-profile
```

```
wget https://raw.githubusercontent.com/hyperledger/blockchain-explorer/main/docker-compose.yaml
```

2. Copy entire crypto artifact directory (organizations/) from your fabric network (e.g /fabric-samples/test-network)

```
cp -r ../fabric-samples/test-network/organizations/ .
```

Now, you should have the following files and directory structure.
Docker-compose.yaml

Config.json

connection-profile/test-network.json

organizations/ordererOrganizations/

organizations/peerOrganizations/

3. Edit environmental variables in docker-compose.yaml to align with your environment

```
export EXPLORER_CONFIG_FILE_PATH=./config.json
export EXPLORER_PROFILE_DIR_PATH=./connection-profile
export FABRIC_CRYPTO_PATH=./organizations
```

Start container services

- Run the following to start up explore and explorer-db services after starting your fabric network:

```
docker-compose up -d
```

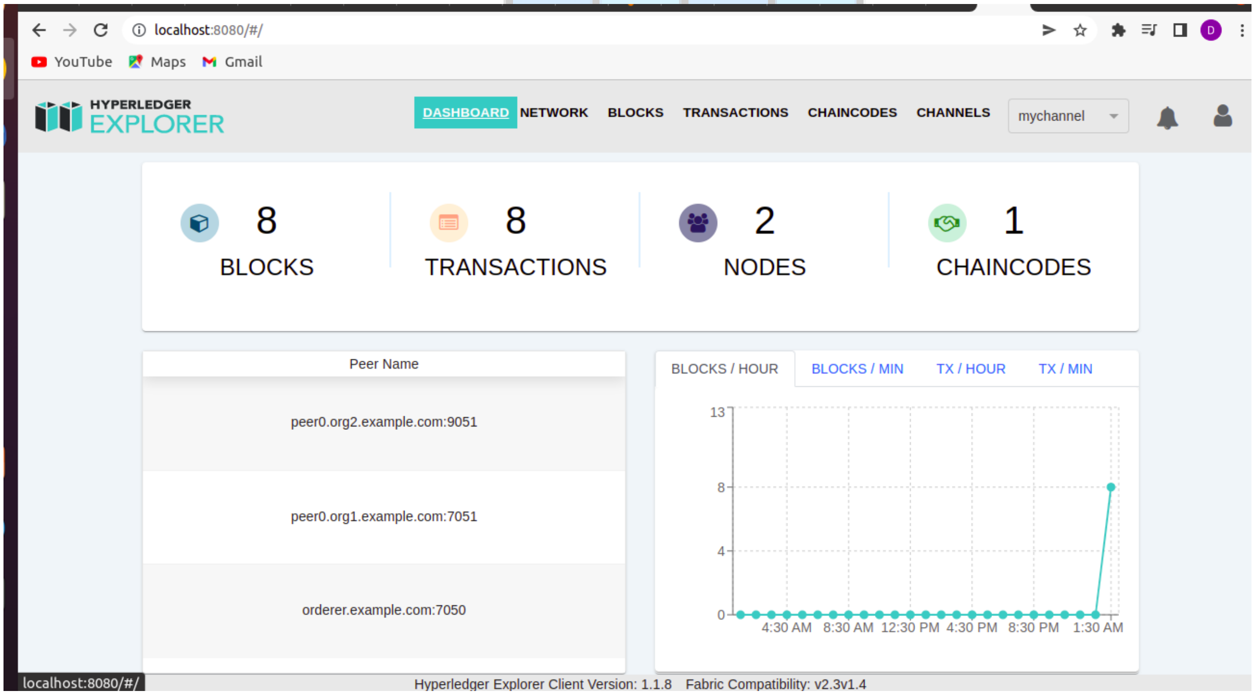
Goto

http://localhost:8080/#/

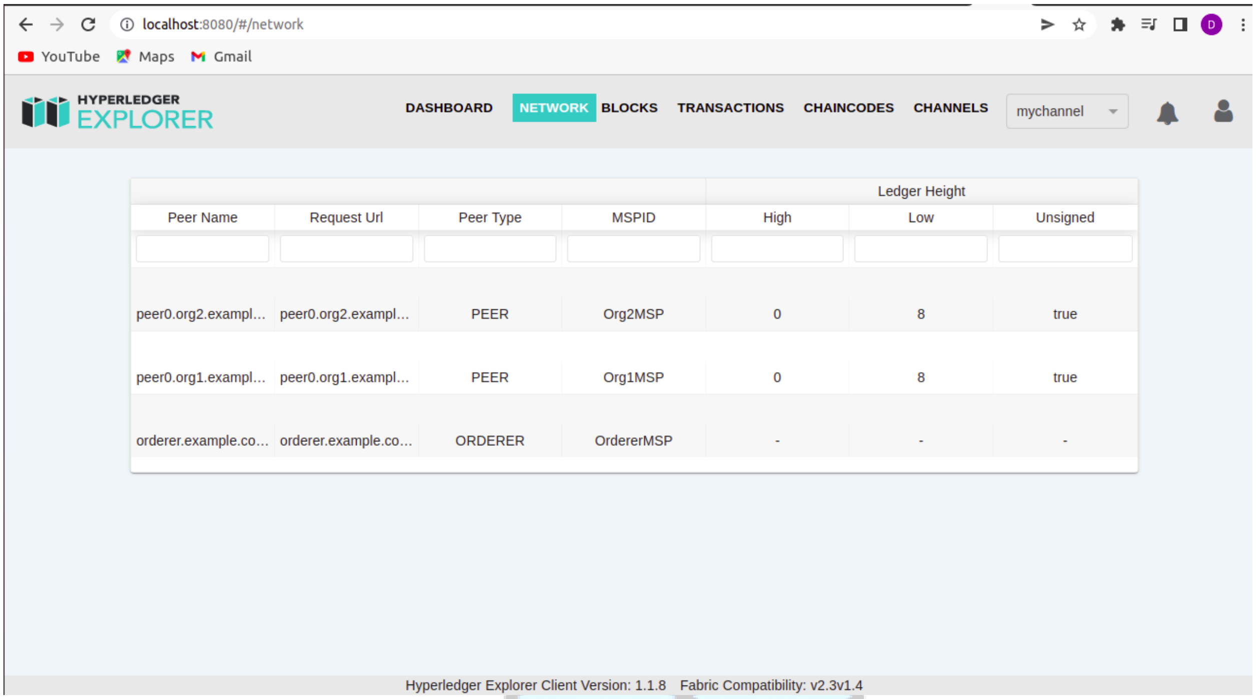
to view explorer dashboard and use the below credentials

username: exploreradmin

Password: exploreradminpw

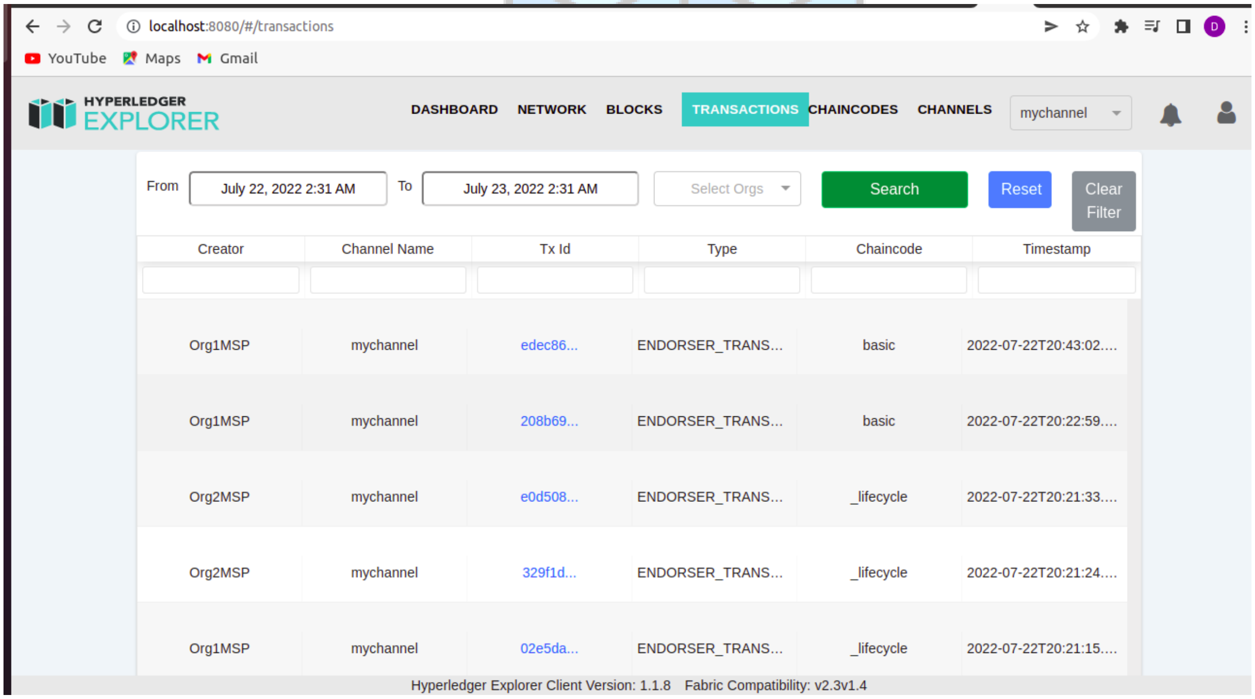


Peer Name
peer0.org2.example.com:9051
peer0.org1.example.com:7051
orderer.example.com:7050



				Ledger Height		
Peer Name	Request Url	Peer Type	MSPID	High	Low	Unsigned
peer0.org2.exampl...	peer0.org2.exampl...	PEER	Org2MSP	0	8	true
peer0.org1.exampl...	peer0.org1.exampl...	PEER	Org1MSP	0	8	true
orderer.example.co...	orderer.example.co...	ORDERER	OrdererMSP	-	-	-

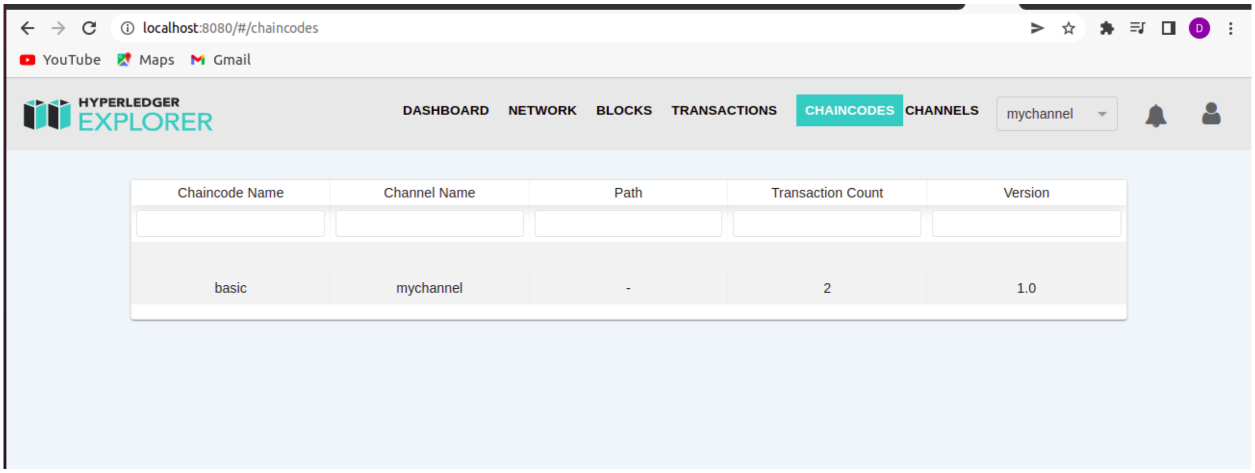
Hyperledger Explorer Client Version: 1.1.8 Fabric Compatibility: v2.3v1.4



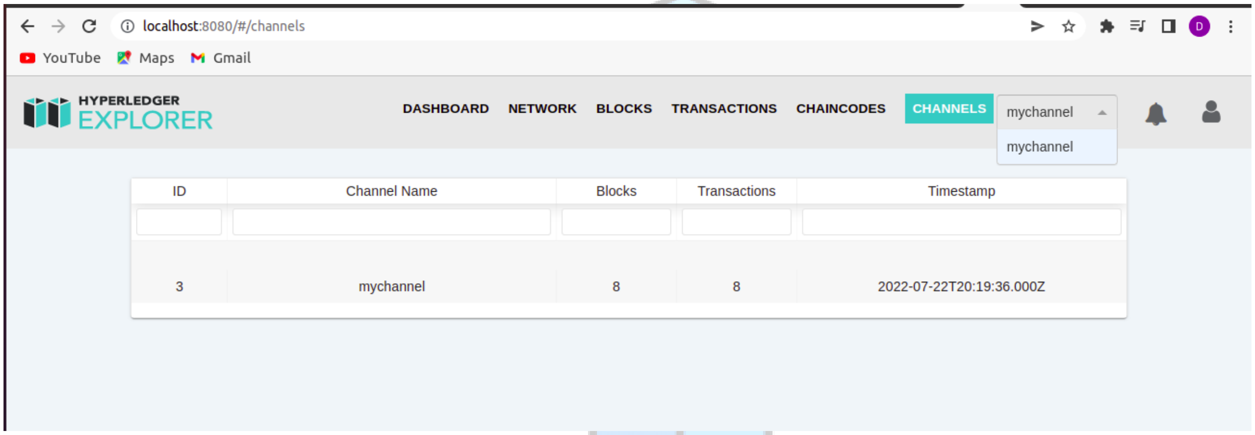
From July 22, 2022 2:31 AM To July 23, 2022 2:31 AM

Creator	Channel Name	Tx Id	Type	Chaincode	Timestamp
Org1MSP	mychannel	edec86...	ENDORSER_TRANS...	basic	2022-07-22T20:43:02....
Org1MSP	mychannel	208b69...	ENDORSER_TRANS...	basic	2022-07-22T20:22:59....
Org2MSP	mychannel	e0d508...	ENDORSER_TRANS...	_lifecycle	2022-07-22T20:21:33....
Org2MSP	mychannel	329f1d...	ENDORSER_TRANS...	_lifecycle	2022-07-22T20:21:24....
Org1MSP	mychannel	02e5da...	ENDORSER_TRANS...	_lifecycle	2022-07-22T20:21:15....

Hyperledger Explorer Client Version: 1.1.8 Fabric Compatibility: v2.3v1.4



Chaincode Name	Channel Name	Path	Transaction Count	Version
basic	mychannel	-	2	1.0



ID	Channel Name	Blocks	Transactions	Timestamp
3	mychannel	8	8	2022-07-22T20:19:36.000Z

Clean up

- To stop services without removing persistent data, run the following:
  $ docker-compose down
- In the docker-compose.yaml, two named volumes are allocated for persistent data (for Postgres data and user wallet). If you would like to clear these named volumes up, run the following:
  $ docker-compose down -v

Reference:

1. https://github.com/hyperledger/blockchain-explorer
2. Hyperledger Caliper: A Benchmark Tool For Multiple Technologies - https://www.youtube.com/watch?v=z5QqXaldDwc

3. HyperledgerExplorer Documentation:https://blockchain-explorer.readthedocs.io/_/downloads/en/main/pdf/

KBA