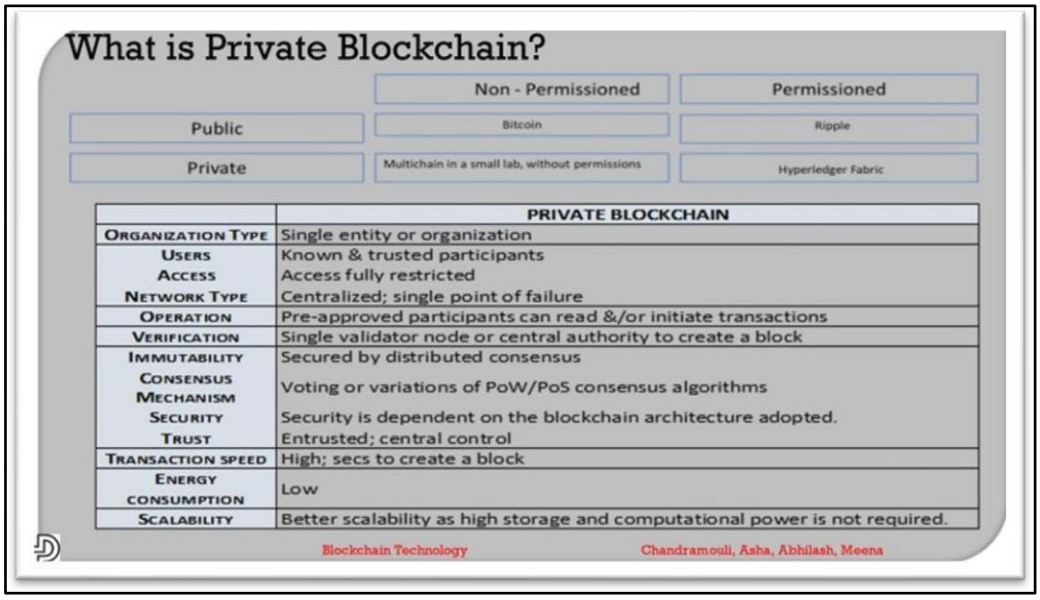
Experiment No. 7

Aim:

Case Study on Hyperledger

Theory:

Private Blockchain



Private blockchains are used by individual hobbyists or by private enterprises or organizations with a specific purpose (e.g., an NGO may like to keep a record of money spentin various schools). Organizations prefer usinga private blockchain, if they would like to control:

- who can use the system
- who can write to the system
- who can read the system

Besides, organizations need a solution with a mechanism to ensure users are added via process, and user rightsarecreated, changed, or deleted by an authorized user. These needs arose and gave birth to a need for private blockchain. In addition, the solution needs faster transactions, proper audit trail, interactions with the organization's existing IT systems.

Hyperledger:

The Linux Foundation took up the challenge for an open-source enterprise-grade distributedledger technology andannounced the Hyperledger Project in December 2015.

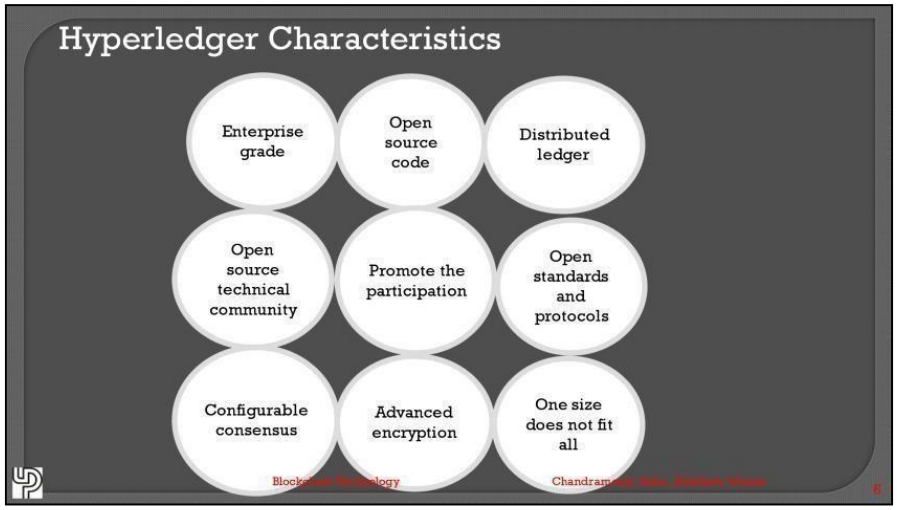
The approach was to ensure that the best practices of computer science related todistributed computing are used in blockchain for enterprise solutions.

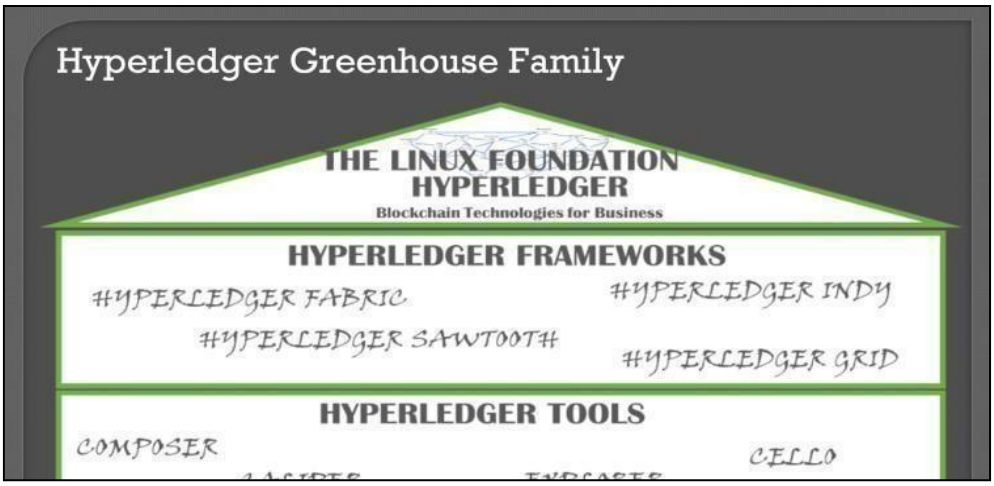
It needs to be noted that Hyperledger has stated they will not be issuing itscryptocurrency

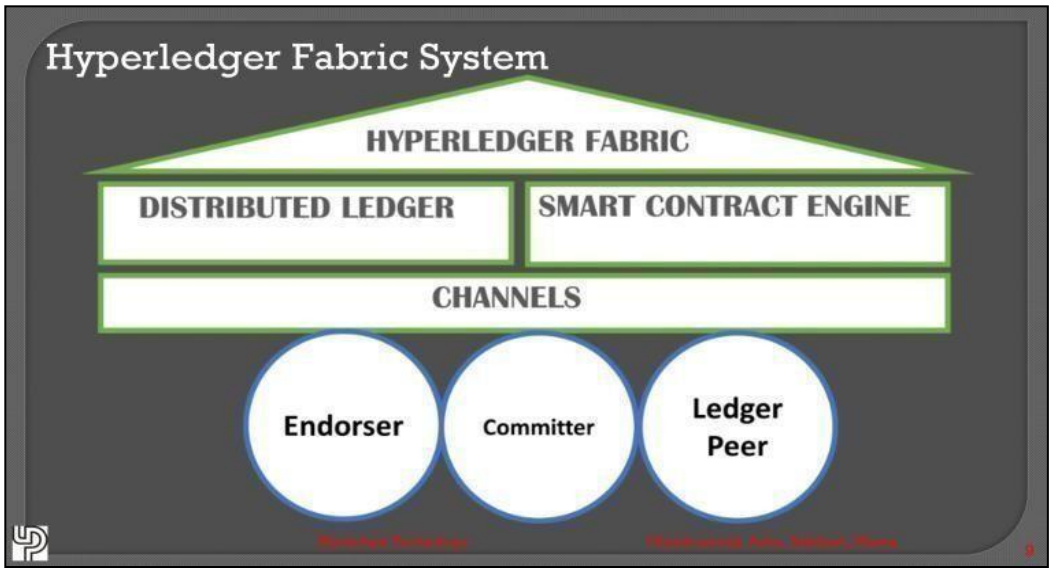
Mision of hyperledger:

As per hyperledger.org, the mission of Hyperledger Project (HLP) is to

- create an enterprise-grade, open-source distributed ledger framework and codebase
- create an open-source, technical community to benefit the ecosystem of the HLP(Hyper LedgerProject) solution
- promote the participation of leading members of the ecosystem, including developers,service and solution providers and end-users.







Hyperledger Indy:

Hyperledger INDY, a part of the hyperledger framework, is a blockchain tool for digital identity. An organizationcalled sovrin.org donated the code base for INDY.

INDY is known for providing digital identities in a decentralized environment. INDY provides tools, libraries, and reusable components for the creation of digital identities. INDY's status is incubation, although it has documented specifications for identity along witha sample implementation are available.

As Hyperledger, INDY is related to identity, and being a blockchain, an identity created once cannot be altered. Designers and administrators of INDY are requested to have propertraining for foundational concepts, which includes the following:

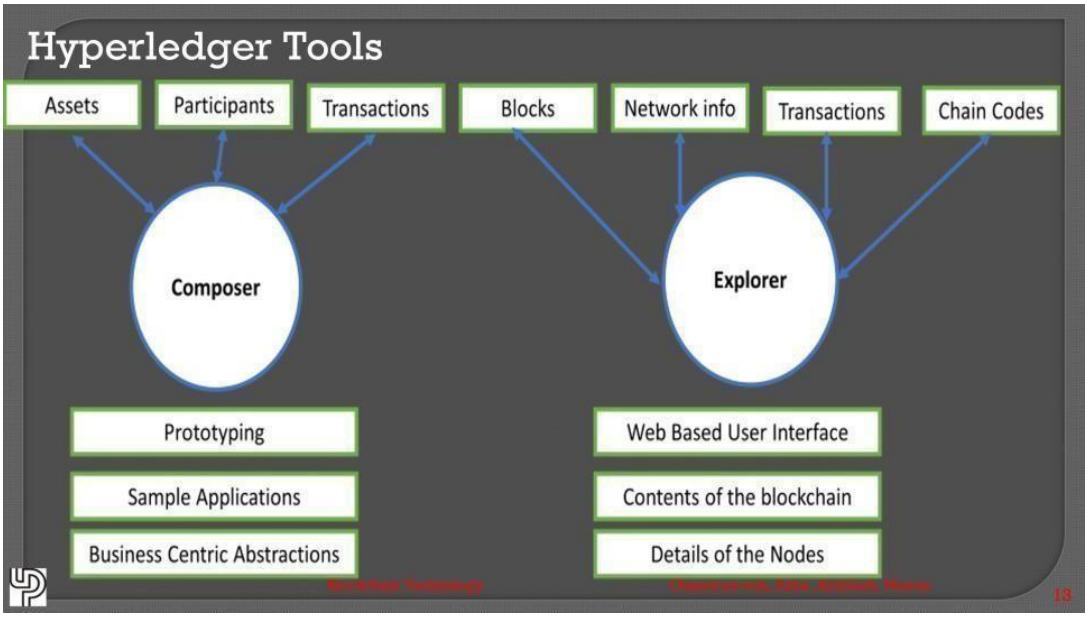
- Privacy by design
- Privacy-preserving technologies

Hyperledger Sawtooth :

Sawtooth is a distributed ledger technology and powered with a smart contract engine. The sawtooth project started with a contribution via Intel. Sawtooth offers a robust runtimeenvironment, even allowing change of consensus approach in run time. Sawtooth being a permissioned layer bring restrictions via Access Control Lists, nodes are put into these restrictions:Who can connect to the network?Who can send consensus messages?Who can submit transactions to the network? Sawtooth is the only project within Hyperledger Project: that uses Ethereum: and smart contracts are written via solidity, as it is written in Ethereum. Even thesmart contracts can be deployed on the fly to the sawtooth network.

Hyperledger Grid:

Hyperledger Grid is a framework (a framework is a set of best practices to achieve a task). Grid is focussed on only one segment: Supply chain Management. Being a framework, Hyperledger grid does not contain rules; instead it is an ecosystem detailing a blockchain for supply chain management. It includes data sets, frameworks that work together, letting application developers choose the best-suited technology or methodology as per company's requirement.



Hyperledger tools:

Composer:

Composer offers business-centric abstractions as well as sample apps, which are used to test or replicate businessproblems.Composer is handy when you have to build an applicationpart of Proof-of-Concepts (PoC), and you have a concise timeline.Composer operates as a rapid prototyping tool for user-facing solutions. Hyperledger fabric is the underlying mechanism to operate a blockchain for the composer. Composer allows for creation/modification of the following:

- Assets
- Participants
- Transactions

Explorer: is another tool hosted in the Hyperledger greenhouse, which provides aweb-based user interface.

Hyperledger Explorer allows the user to view contents in the blockchain, and list the nodesHyperledger Explorercan view, invoke, deploy or query. These nodes include the following:

- Blocks
- Transactions
- Network information (name, status, list of nodes)
- Chain codes.

Hyperledger Fabric :is the most popular of the Hyperledger project. All blockchains strive to solve the problemof trust and time, and blockchain Hyperledger fabricis no different. Hyperledger fabric is amongst the best solutions where the organization can choose the trust mechanism it needs to use. The issue of trust hasa profoundimpact on Supply Chain Management, which is the most, talked and piloted use-case of blockchain.

Before you install Hyperledger Fabric, you must first download and install the prerequisites that are required torun a Docker-based Fabric test network on your local machine from https://hyperledger-fabric.readthedocs.io/en/latest/prereqs.html.

Hyperleder Fabric Prerequisites Setup:

Curl Installation
Run below command to install Curl.

$ sudo apt-get install curl

Verify the installation and check the version of Curl using below command.

$ curl –version

NodeJs Installation

Open the terminal window and run below command to download and execute the nodejs file.

$ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash –

Then run below command.

$ sudo apt-get update

Run below command to start the installation for NodeJs.

$ sudo apt-get install nodejs

Run below command to check if Nodejs is successfully installed or not. This should return the versionof NodeJs.

$ node –version

Git Installation

Open the terminal window and run below command. This will start the installation for Git.

$ sudo apt-get install git

Run below command to check if Git is successfully installed or not. This should return the version of Git.

$ git –version

Python Installation

In the terminal window, run below command to install Python.

$ sudo apt-get install python

Verify the installation by running below command and that should return the version of Python.

$ python –version

Lib Tools Installation

Install Lib tools using below command.

$ sudo apt-get install libltdl-dev

Install Docker CE (Community Edition)

First download and then install it using below commands.

$ wget https://download.docker.com/linux/ubuntu/dists/xenial/pool/stable/amd64/docker-ce_18.06.3~ce~3-0~ubuntu_amd64.deb

$ sudo dpkg -i docker-ce_18.06.3~ce~3-0~ubuntu_amd64.deb

Check the version of docker using below command and this should return the version of docker.

$ docker –version

Install Docker Compose

Run below commands to setup Docker compose.

$ sudo apt-get install python-pip

$ pip --version

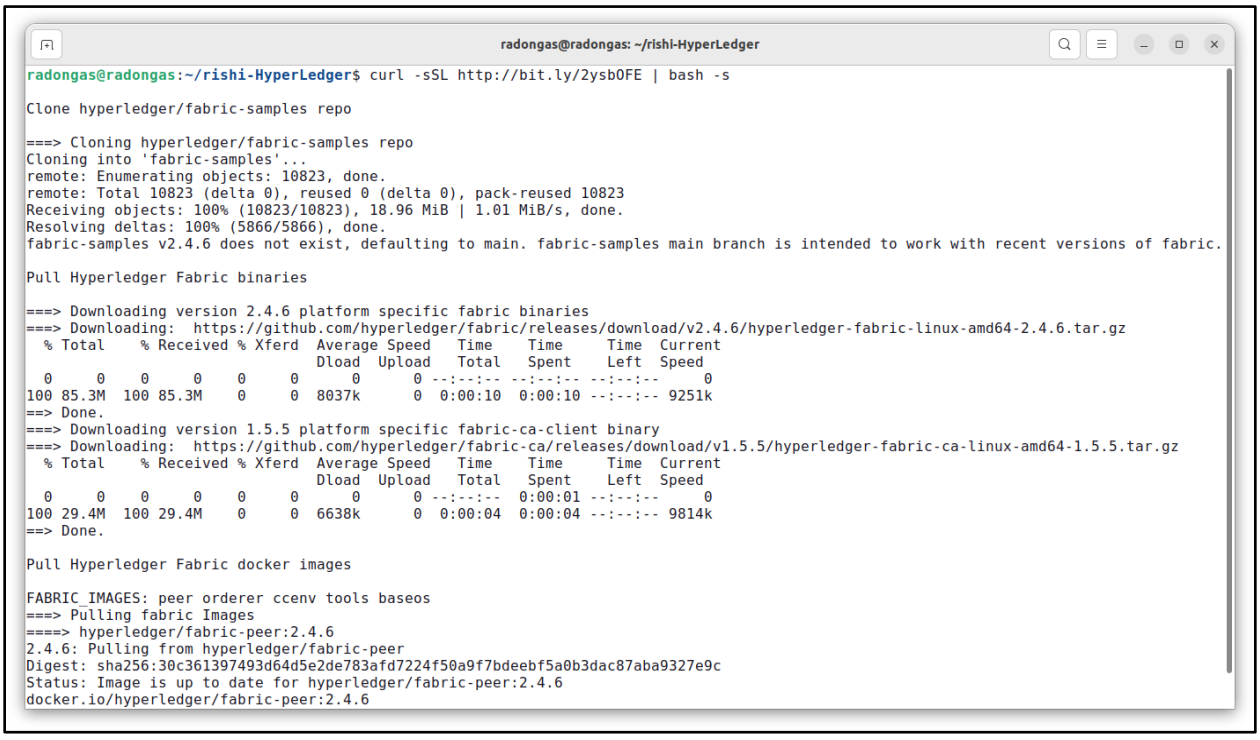
$ sudo pip install docker-compose

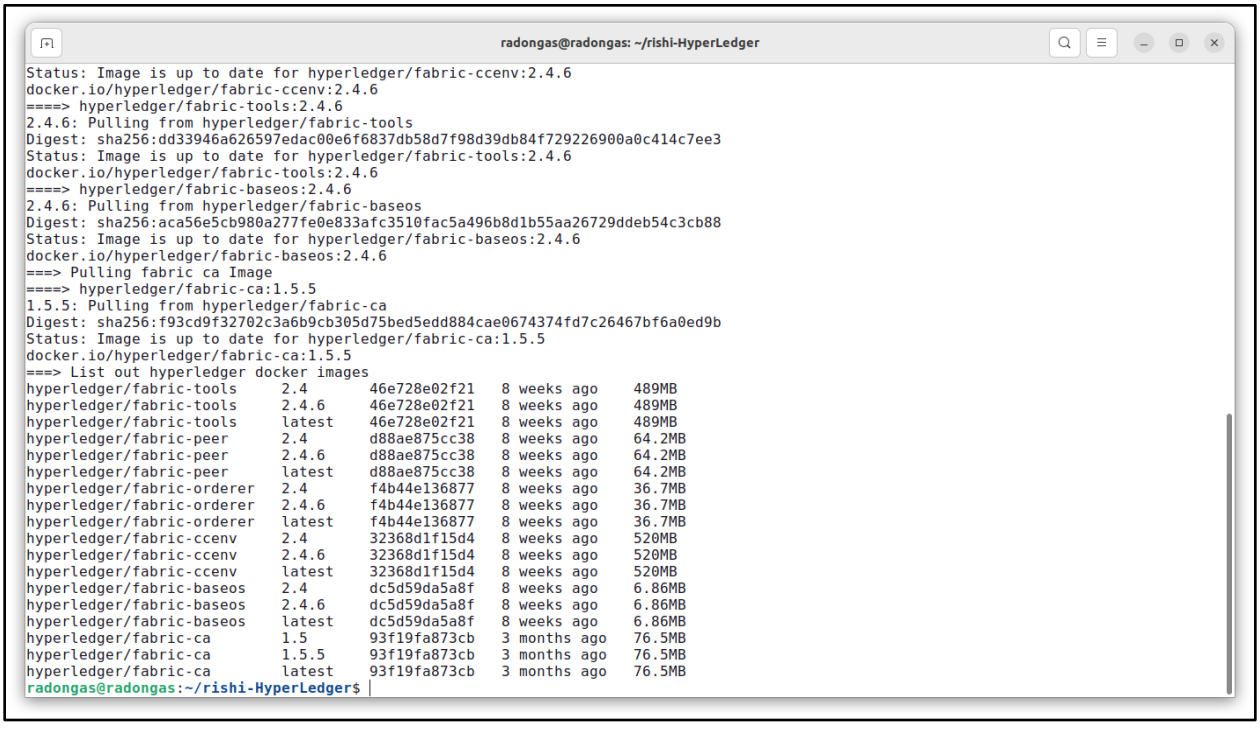
Verify the installation and check the version from below command.

$ docker-compose version

Hyperledger Installation:

Step 1: Run below command to download and setup Fabric.
$ curl -sSL https://bit.ly/2ysbOFE | bash -s



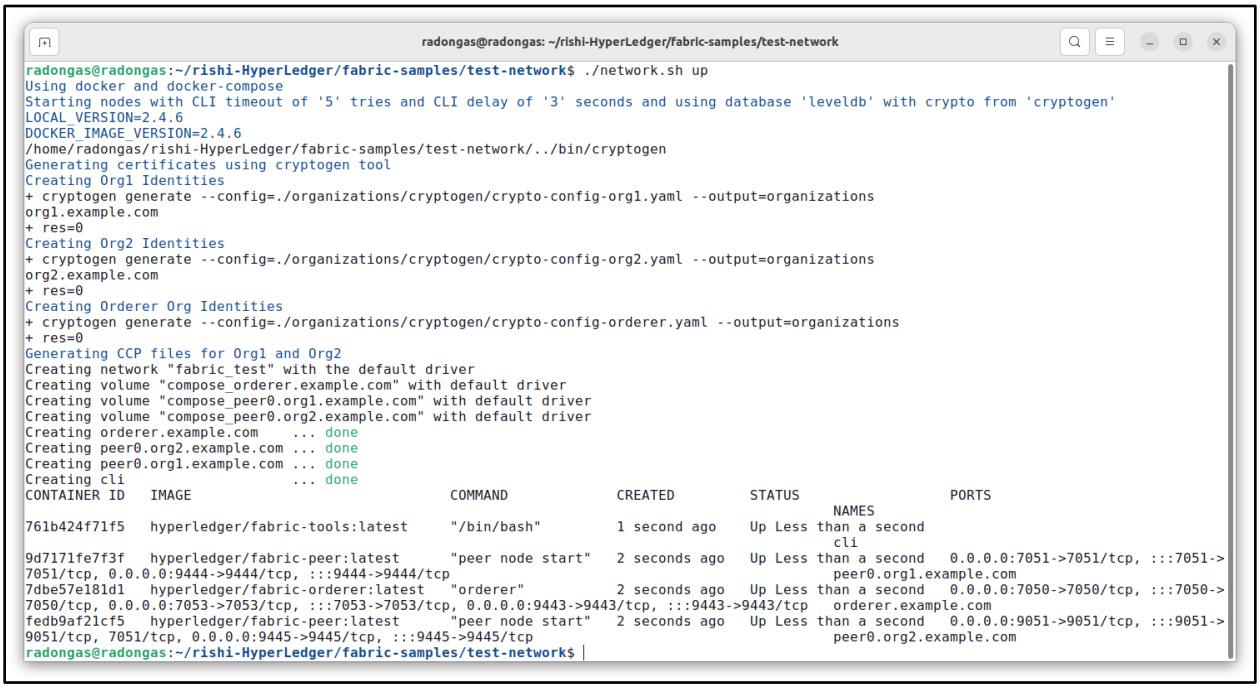


Running Hyperleder Fabric Testnetwork:

Step 1: Go to fabric-samples folder by using below command.
$ cd fabric-samples

Step 2: Go to test-network folder by using below command.
$ cd test-network

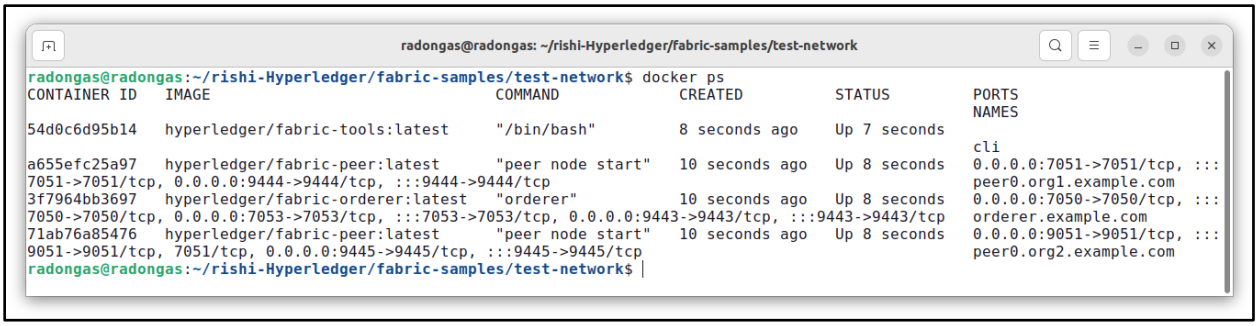
Step 3: Run below command to start your test-network
$ sudo ./network.sh up



This start the network, you can run below command to check docker containers.
$ sudo docker ps

This shows you three docker containers

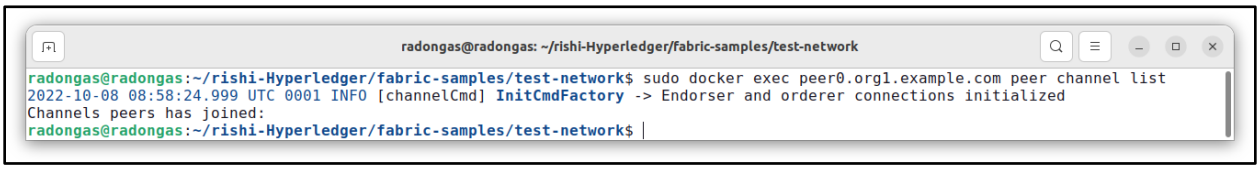
- One for Org1 peer node
- One for Org2 peer node
- One for Orderer



When you start the network, you will also not get any channel by default. You can check the channel by usingbelow command.

$ sudo docker exec peer0.org1.example.com peer channel list

This command shows you that, you don't have any channel created.



Step 4: Create new channel by using below command.

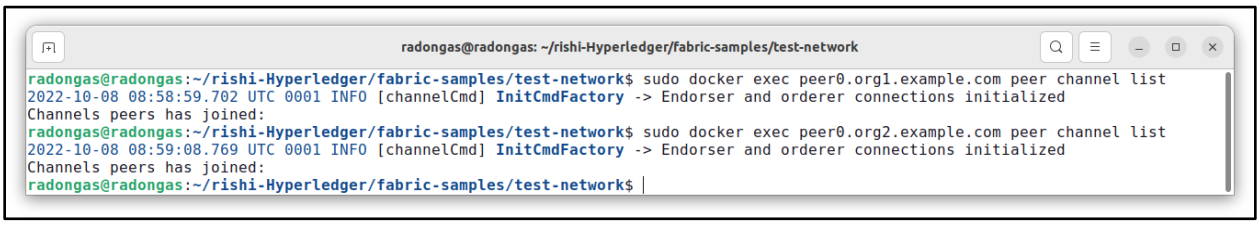
$ sudo ./network.sh createChannel -c testchannel

This will create a new channel with the name test channel.

To verify this channel creation, run below command on both the peers.

$ sudo docker exec peer0.org1.example.com peer channellist

$ sudo docker exec peer0.org2.example.com peer channellist



Step 5: To stop the network, you need to run below command.

$ sudo ./network.sh down

Working with State DataBase (Couch DB):

Step 1: Go to fabric-samples folder by using below command.

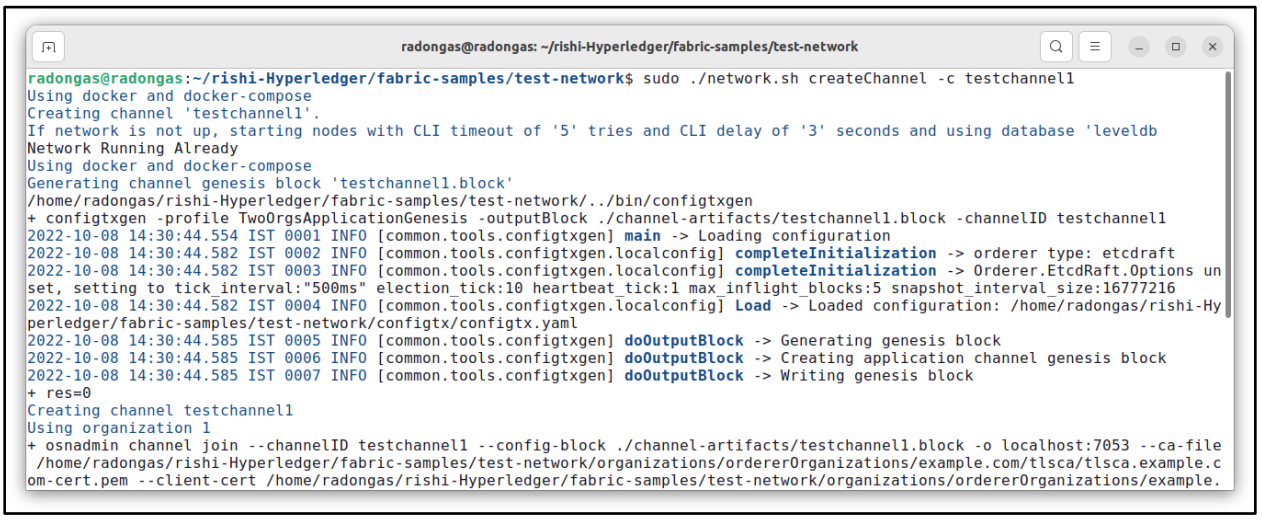
$ cd fabric-samples

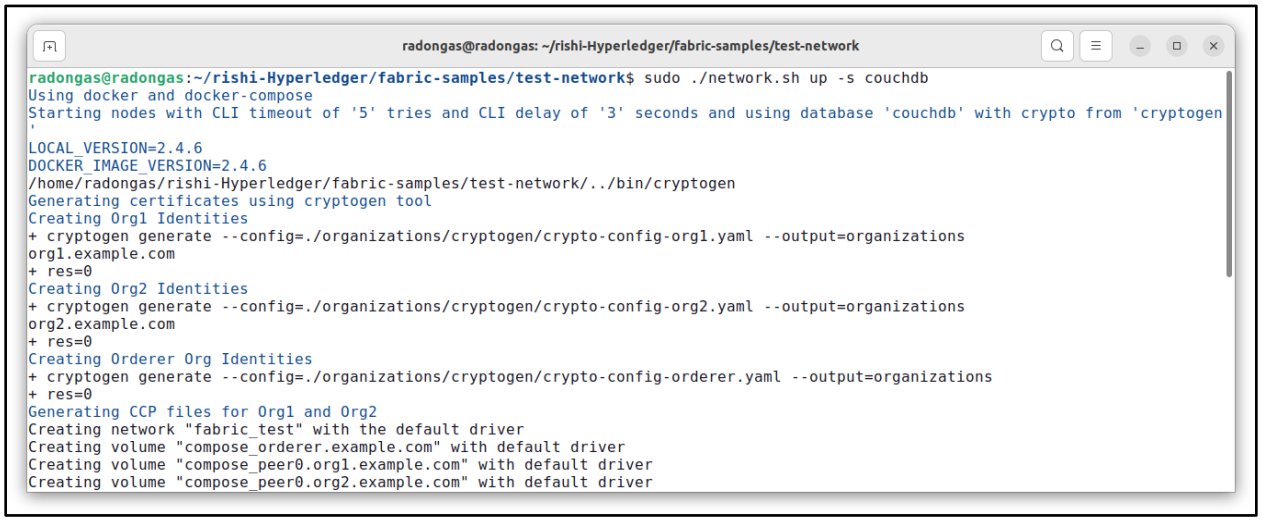
Step 2: Go to test-network folder by using below command.

$ cd test-network

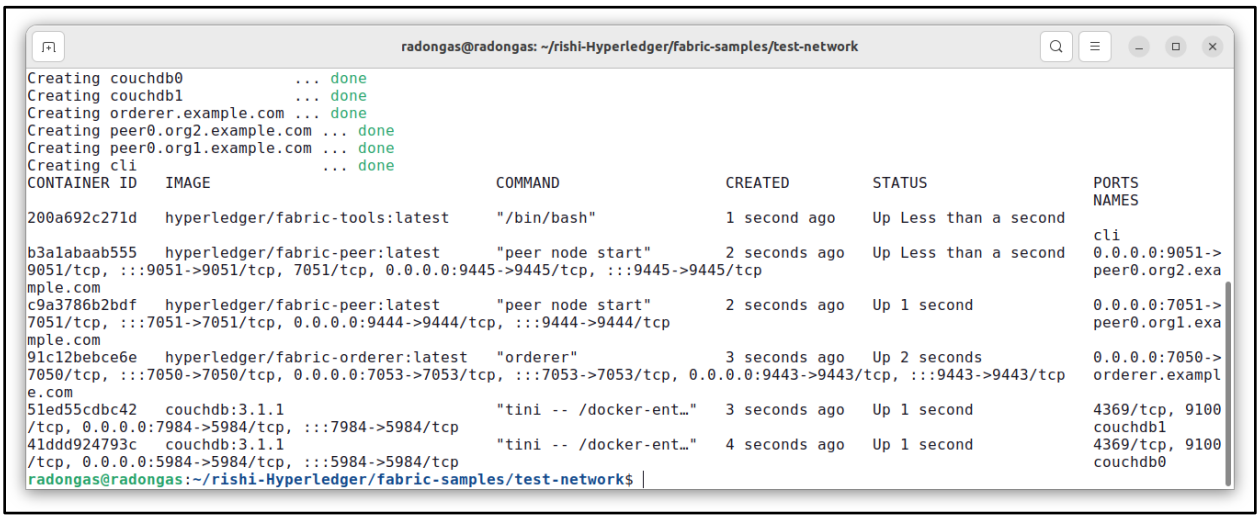
Step 3: Run below command to start the network and create couchDB containers as well.

$ sudo ./network.sh up -s couchdb





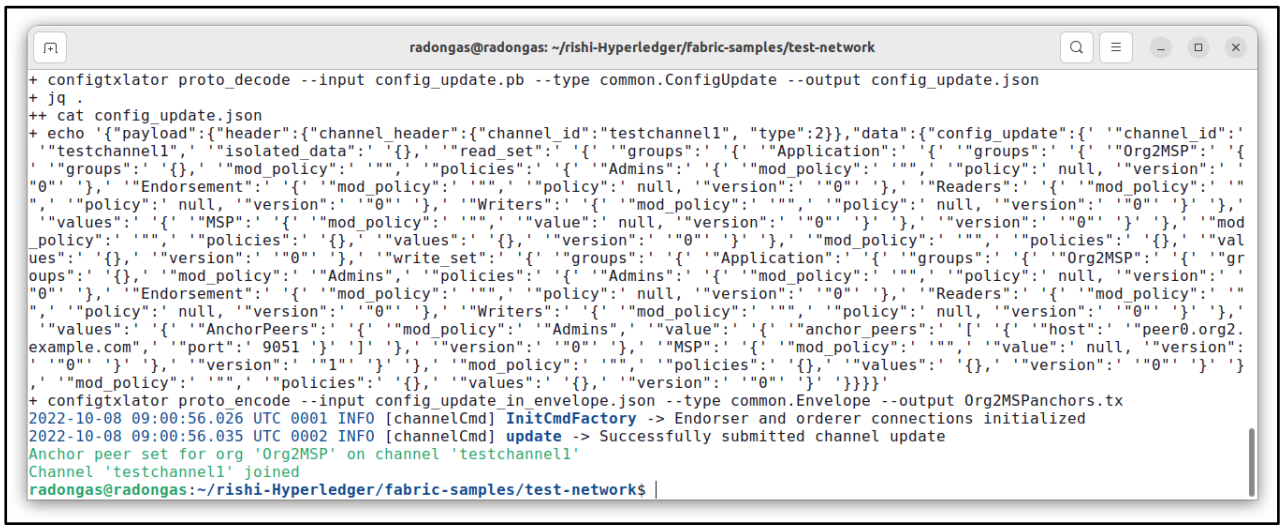
This command starts your network and create couchdb container for each peer as well.



Step 4: Create new channel by using below command.

$ sudo ./network.sh createChannel -c testchannel1

This will create a new channel with the name testchannel1.



Step 5: To stop the network, you need to run below command.

$ sudo ./network.sh down

Set up the Blockchain Network:

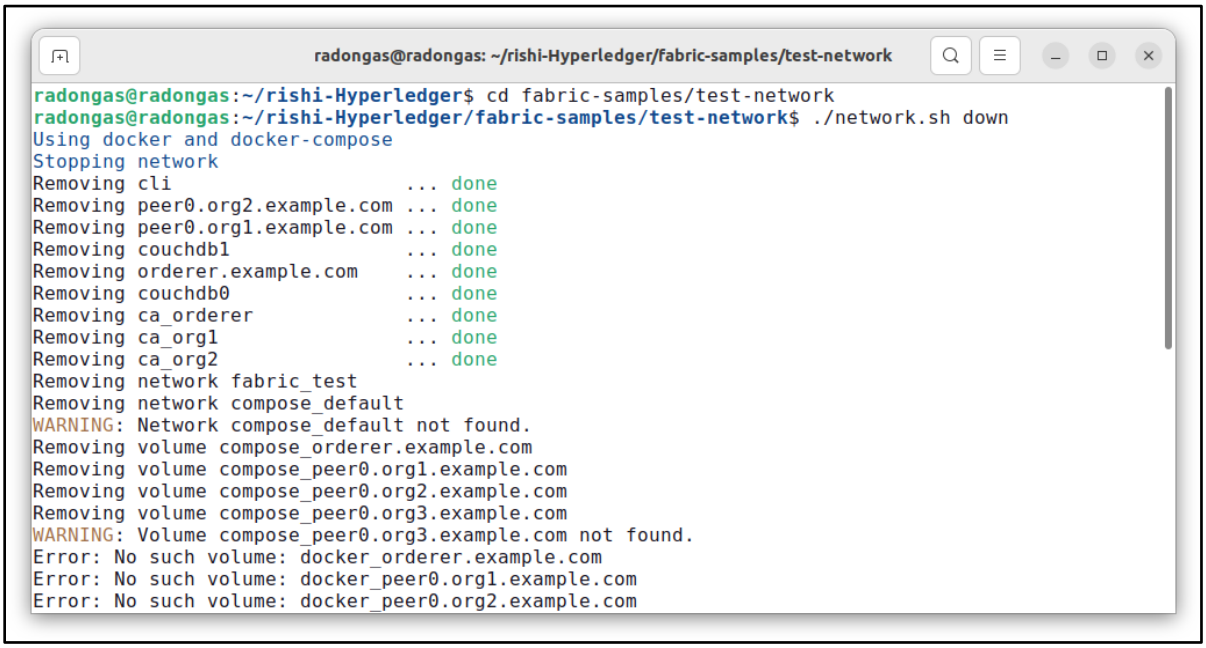
If you've already run through Using the Fabric test network tutorial and have a network up and running, this tutorial will bring down your running network before bringing up a new one.

$ cd fabric-samples/test-network

Navigate to the test-network subdirectory within your local clone of the fabric-samples repository.

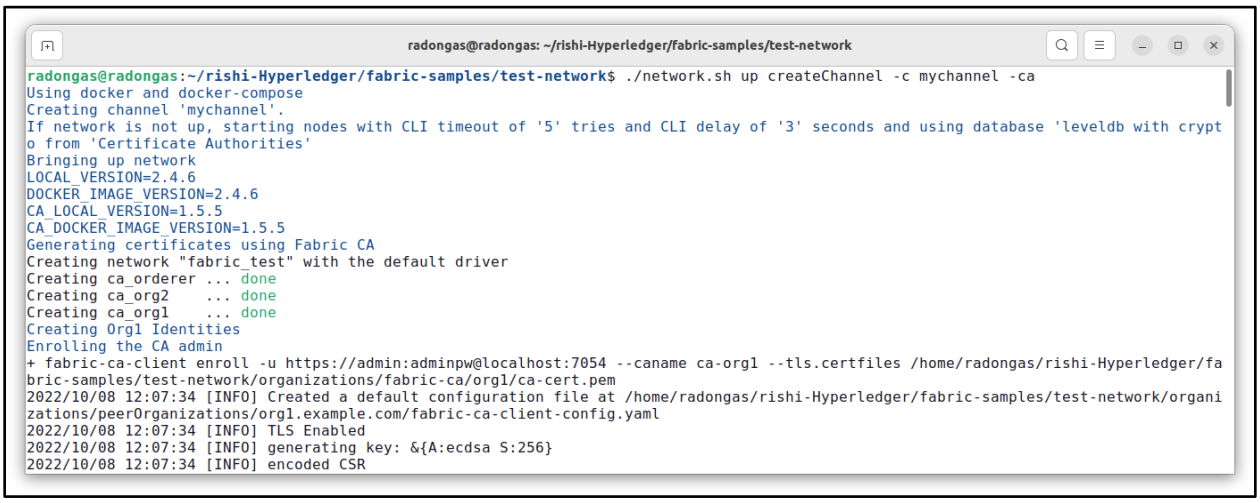
If you already have a test network running, bring it down to ensure the environment is clean.

$ ./network.sh down



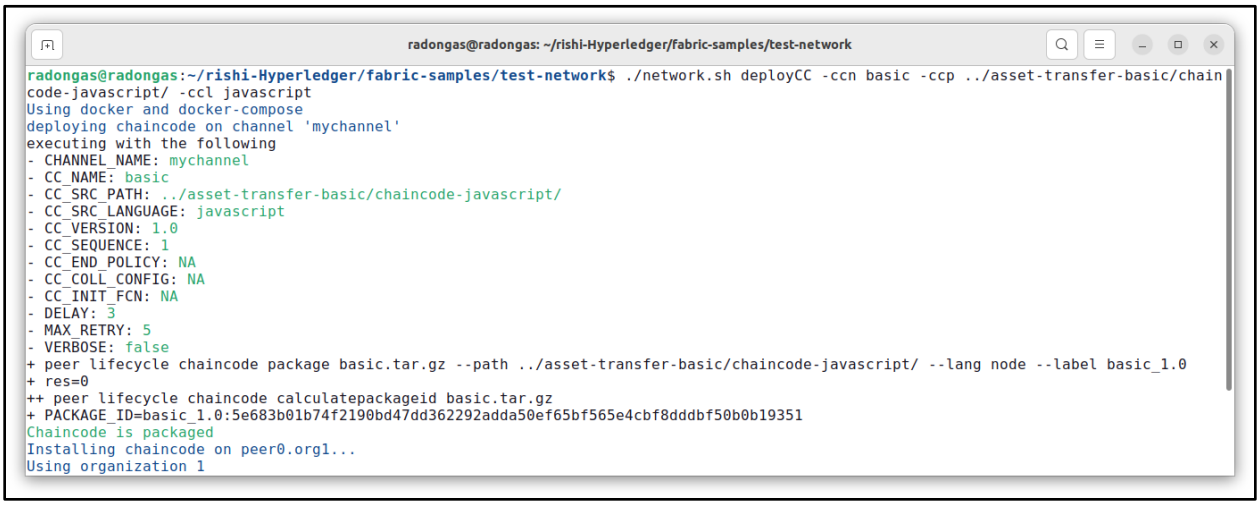
Launch the Fabric test network using the network.sh shell script.

$ ./network.sh up createChannel -c mychannel -ca



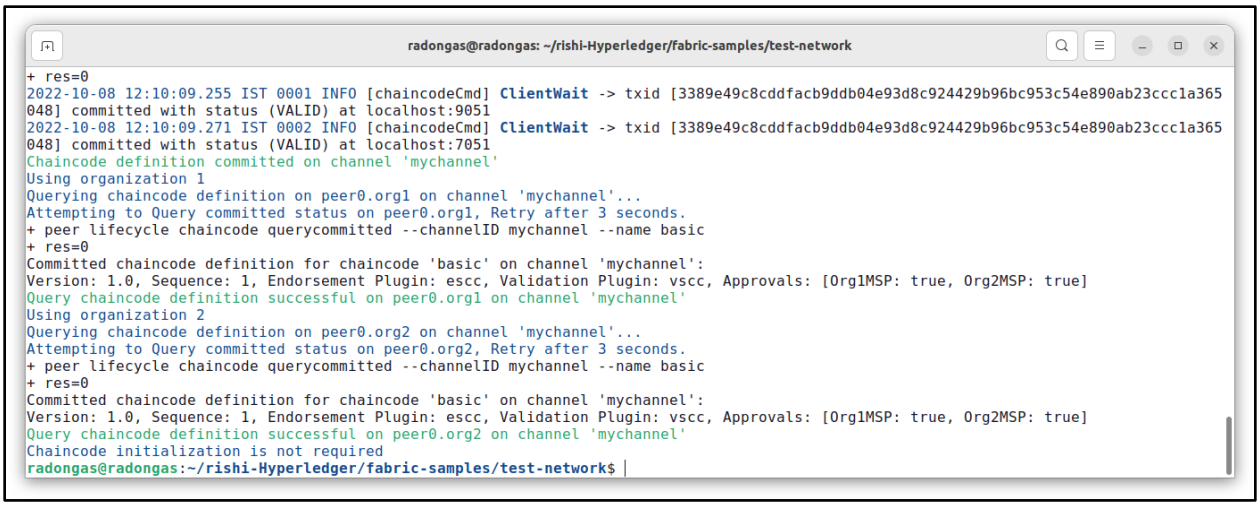
This command will deploy the Fabric test network with two peers, an ordering service, and three certificate authorities (Orderer, Org1, Org2). Instead of using the cryptogen tool, we bring up the test network using Certificate Authorities, hence the -ca flag. Additionally, the org admin user registration is bootstrapped when the Certificate Authority is started. In a later step, we will show how the sample application completes the admin enrollment.

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



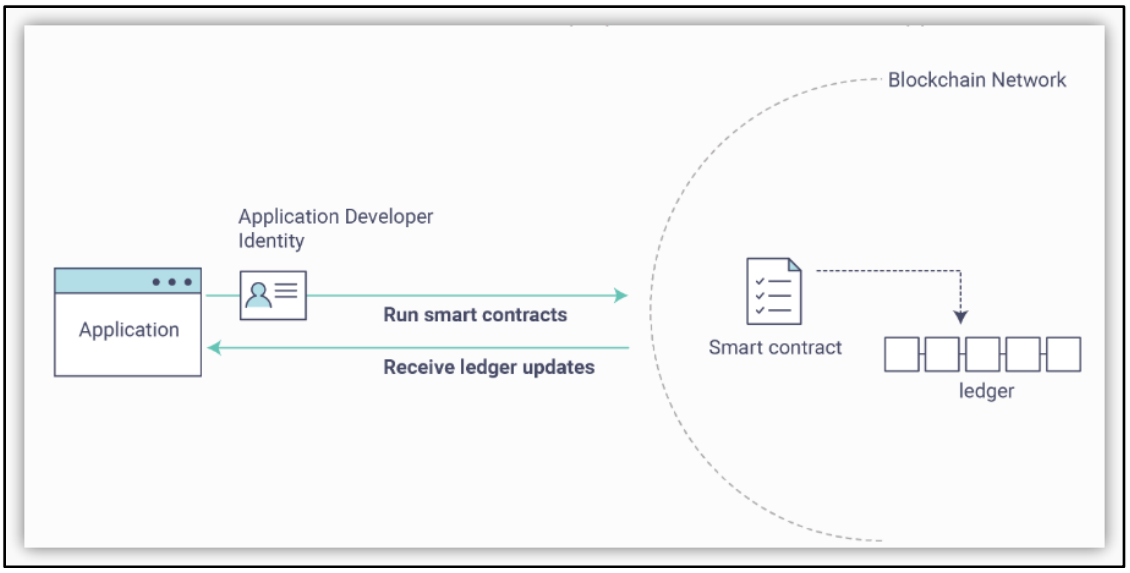
Behind the scenes, this script uses the chaincode lifecycle to package, install, query installed chaincode, approvechaincode for both Org1 and Org2, and finally commit the chaincode.

If the chaincode is successfully deployed, the end of the output in your terminal should look like below:



Next, let's prepare the sample Asset Transfer Javascript application that will be used to interact with the deployed chaincode.

JavaScript Application:



Open a new terminal, and navigate to the application-javascript folder.

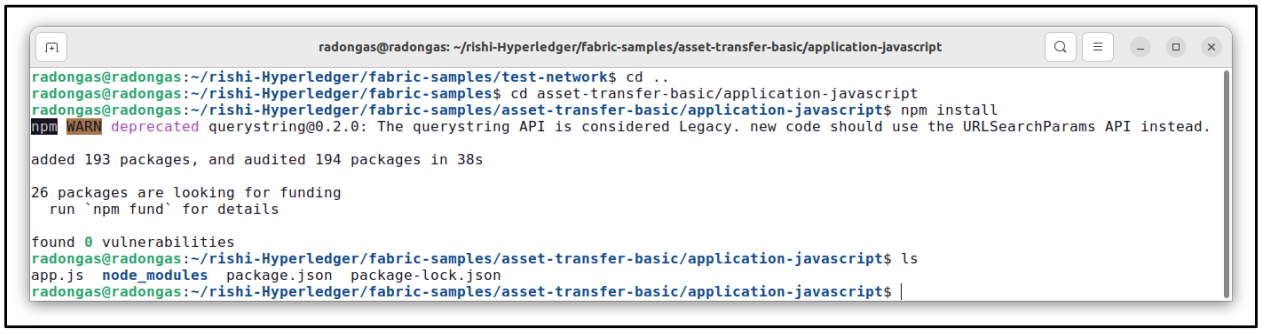
$ cd asset-transfer-basic/application-javascript

This directory contains sample programs that were developed using the Fabric SDK for Node.js. Run thefollowing command to install the application dependencies. It may take up to a minute to complete:

Once npm install completes, everything is in place to run the application. Let's take a look at the sample JavaScript application files we will be using in this tutorial. Run the following command to list the files in this directory:

$ ls

You should see the following:



Let's run the application and then step through each of the interactions with the smart contract functions.

From the asset-transfer-basic/application-javascript directory, run the following command:

$ node app.js

- the application enrolls the admin user.
- the application registers and enrolls an application user.
- the sample application prepares a connection to the channel and smart contract.
- the application initializes the ledger with some sample data.
- the application invokes each of the chaincode functions.

```
radongas8r adongas :-/ rishi- Hyper1edqer/I ob ric - samples/asset -t ronsI er- bosic/opplicution- jovosc ript5  node app. js
Loaded the netwo rk cunt i gurati on located at /tiome/ radongas/ ri shi - Hyperledger/fab ric - sanples/test - ne two rk/o rganizations/peer0rganizations/org i

Built a l°!s s em at et a ı home ra ngas r sh Hype ledg° tJ'abr1c- samptes/asset- I ransfer- bas1c/appt1cat1on- j avascrlpt/vattet
Successfully registered and enrolled user appUser and imported it into the wallet
-> Submit Transaction: InltLedger . function creates the 1n1t1at set of assets on the ledger
t  evaluate Transaction: GetAttAssets , function returns aft the current assets on the ledger

  "ID": "asset1",
  "Owner": "Tomoko",
  "Size": 5,

  'ID': 'asset2',
  "S1ze" : 5,

  "AppralsedVatue" : 500.
  "ID" : "asse t3 " ,
  'Size': 10,
```

```
radongas@radongas: ~/rishi-Hyperledger/fabric-samples/asset-transfer-basic/application-javascript/wallet
radongas@radongas:~/rishi-Hyperledger/fabric-samples/asset-transfer-basic/application-javascript$ cd wallet/
radongas@radongas:~/rishi-Hyperledger/fabric-samples/asset-transfer-basic/application-javascript/wallet$ cat admin.id && echo ""
{"credentials":{"certificate":"-----BEGIN CERTIFICATE-----\nMIIB8jCCAZmgAwIBAgIUNDDGVP690gYYrhe6lsIYcutGwlUwCgYIKoZIzj0EAwIw\ncDELMAkGA1UEBhM
CVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMQ8wDQYDVQQH\nEwZEdXJoYW0xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh\nLm9yZzEuZXhhbXBsZS5jb20w
HhcNMjIxMDA4MDYzMzAwWhcNMjMxMDA4MDY0MzAw\nWjAhMQ8wDQYDVQQLEwZjbGllbnQxDjAMBgNVBAMTBWFkbWluMFkwEwYHKoZIzj0C\nAQYIKoZIzj0DAQcDQgAEi0rOz7IIF/8Mt
IyGDlJs0WZ7j8Fhtbs5FSDTKIf+kG37\nvxT1tCaPe2Dxt9go+XU5D5GUXKXRfUqcQxe7vVrYRqNgMF4wDgYDVR0PAQH/BAQD\nAgeAMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFIR39P
8jBlLYMtyLoU5mJj0Mz81r\nMB8GA1UdIwQYMBaAFPa5KaLUlqbDYWqL6gRq4gaJjnjoMAoGCCqGSM49BAMCA0cA\nMEQCIFsEqHD2Cuyx1c6goZe3ZLhK3iZH3YjjFgHj5dCLmr8QAiB
J+GJN8RrbQYGj\n0cIB9XAmTRNsGsXIZLjR9YBVHRKA5A==\n-----END CERTIFICATE-----\n","privateKey":"-----BEGIN PRIVATE KEY-----\r\nMIGHAgEAMBMGByqGSM
49AgEGCCqGSM49AwEHBG0wawIBAQQgrbPYrMfur2tNOUZo\r\nJb5E4bpcIH3iZItQ8dx+efXU6iOhRANCAASLSs7PsggX/wy0jIYOUmzRZnuPwWG1\r\nuzkVINMoh/6Qbfu/FPW0Jo9
7YPG32Cj5dTkPkZRcpdF9SpxDF7u9WthG\r\n-----END PRIVATE KEY-----\r\n"},"mspId":"Org1MSP","type":"X.509","version":1}
radongas@radongas:~/rishi-Hyperledger/fabric-samples/asset-transfer-basic/application-javascript/wallet$ cat appUser.id && echo ""
{"credentials":{"certificate":"-----BEGIN CERTIFICATE-----\nMIIChDCCAiqgAwIBAgIUW81vUnp0ltdLsFrGbmf18fwihocwCgYIKoZIzj0EAwIw\ncDELMAkGA1UEBhM
CVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMQ8wDQYDVQQH\nEwZEdXJoYW0xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh\nLm9yZzEuZXhhbXBsZS5jb20w
HhcNMjIxMDA4MDYzMzAwWhcNMjMxMDA4MDY0MzAw\nWjBEMTAwCwYDVQQLEwRvcmcxMA0GA1UECxMGY2xpZW50MBIGA1UECxMLZGVwYXJ0\nbWVudDExEDAOBgNVBAMTB2FwcFVzZXIwW
TATBgcqhkjOPQIBBggqhkjOPQMBBwNC\nAAQh90yVImsgwBchEmELjyPQSz2i88/khjsie3bhtbOBqn9t4UGxusZks1UKgo7T\nV/NFhuaehx1yLkLAHjnDqcM7o4HNMIHKMA4GA1UdDw
EB/wQEAwIHgDAMBgNVHRMB\nAf8EAjAAMB0GA1UdDgQWBBSo78uVFOvFMTFiOzdYPMazu8I4TzAfBgNVHSMEGDAW\ngBT2uSmi1Jamw2Fqi+oEauIGiY546DBqBggqAwQFBgcIAQReeyJ
hdHRycyI6eyJo\nZi5BZmZpbGlhdGlvbiI6Im9yZzEuZGVwYXJ0bWVudDEiLCJoZi5FbnJvbGxtZW50\nSUQiOiJhcHBVc2VyIiwiaGYuVHlwZSI6ImNsaWVudCJ9fTAKBggqhkjOPQQD
AgNI\nADBFAiEA/Gpuulnt1cnLNVGcLN7DQ7IikAiYS0jWBnkJh8+i3EACIHx3tppvHE6d\n5vF833maZRLCir+16Xfks0opinvxzrZv\n-----END CERTIFICATE-----\n","priva
teKey":"-----BEGIN PRIVATE KEY-----\r\nMIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQg29pUzEWCYoeRZG8G\r\nmj2wyicH0UFCV06Ar0guBqbVFVWhRANCAA
Qh90yVImsgwBchEmELjyPQSz2i88/k\r\nhjsie3bhtbOBqn9t4UGxusZks1UKgo7TV/NFhuaehx1yLkLAHjnDqcM7\r\n-----END PRIVATE KEY-----\r\n"},"mspId":"Org1MS
P","type":"X.509","version":1}
radongas@radongas:~/rishi-Hyperledger/fabric-samples/asset-transfer-basic/application-javascript/wallet$
```

When you are finished using the asset-transfer sample, you can bring down the test network

$ ./network.sh down

Using network.sh script.

Conclusion:

Q. Can you provide a real-world case study of how a company implemented Hyperledger technology in its business operations, and what benefits were realized?