

Bhartiya Vidya Bhavan’s

Sardar Patel Institute of Technology, Mumbai-400058

Department of Electronics and Telecommunication Engineering

**OEIT1:Blockchain Technology and Applications**

**Lab-8A: Hyperledger Fabric Application- Setup HF and Supply Chain Management (SCM)**

**Name: Adwait Purao**

**Class: TE COMPS B**

**UID: 2021300101**

**Objective:** Building a Supply Chain Management application using Hyperledger Fabric

**Outcomes:** After completing this lab, students will be able to:

[1] Implement a dApp using Hyperledger fabric

[2] Demonstrate the basic concepts of Hyperledger fabric and how they work.

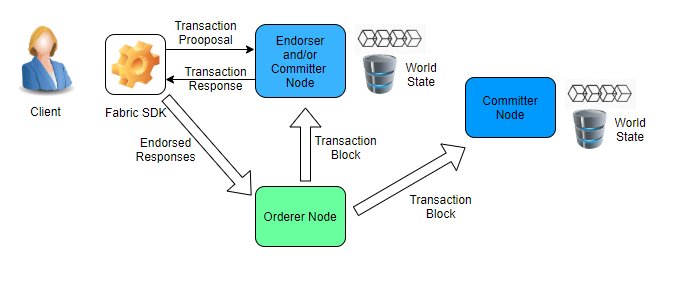
[3] Demonstrate the importance of Certifying authority.

**System Requirements:**

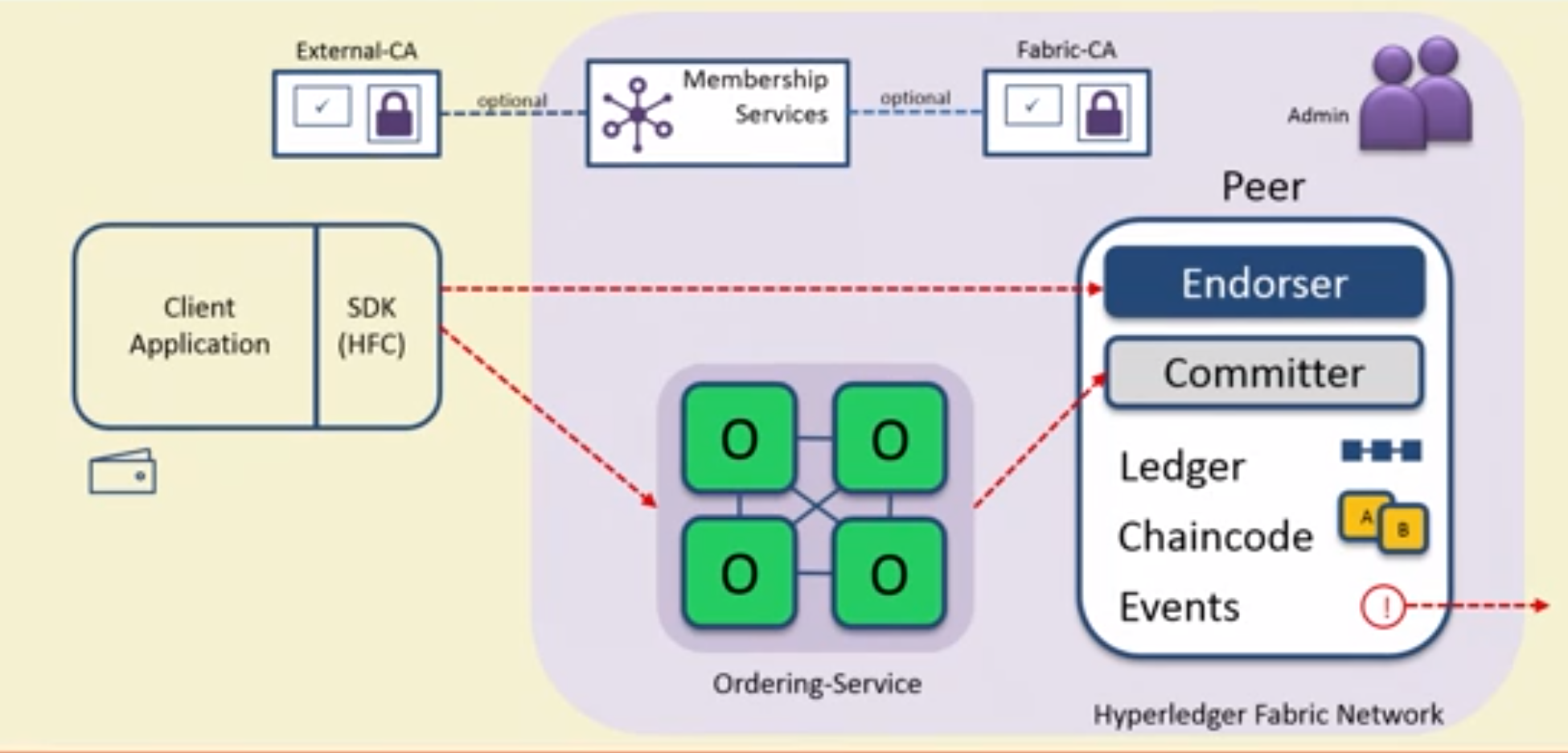
PC (C2D, 8GB RAM, 100GB HDD space and NIC),Ubuntu Linux 14.04/20.04 Internet connectivity,Python Cryptography and Pycrypto,Nodejs, Truffle,Ganache-cli, solidity,REST API

**Theory:**

Hyperledger Fabric is a permissioned blockchain framework, with a modular architecture (plug-and-play). It leverages container technology to host smart contract (Chaincode) which contains application logic.



Hyperledger Fabric Basic Transaction Flow



**Hyperledger Fabric capabilities**

Hyperledger has a full-stack, enterprise-grade business solution to deliver secure and scalable value with added security, confidentiality, and performance. Hyperledger Fabric delivers the following functionalities and core capabilities:

**Identity management:** To turn a permissioned network, Hyperledger Fabric provides a membership identity service that maintains user IDs and then authenticates each one of them in the network. One user ID can invoke a chain code application but can be blocked to turn up a new chain code.

**Efficient processing:** Hyperledger assigns a role for each node based on transaction ordering and commitment. The overall performance improves as the concurrent execution increases and improves the time to deliver each order.

**Privacy and confidentiality:** Private channels restrict the messaging paths to provide transaction privacy and confidentiality for specific network members. All data, including member information, transactions, and channel details, remains invisible and inaccessible to other network members.

**Chaincode functionality:** This regards chaincode applications and is the business logic of Hyperledger Fabric. Chaincode ensures that all transactions that transfer ownership are subject to its rules and requirements. The operating parameters of the channel are usually defined by the system chaincode, whereas the validation system chaincode defines the requirements for endorsing and validating transactions.

Stage-0: Use official document of Hyperledger fabric:

Setup Hyperledger Fabric using Docker and Docker-compose

<https://hyperledger-fabric.readthedocs.io/en/release-2.0/getting_started.html>

Stage-1: Tuna Application Development

**Procedure:**

Cloning the education.git repository of Hyperledger from github.

$mkdir lab8a

$cd lab8a

$git clone <https://github.com/hyperledger/education.git>

Installing docker.io

$sudo apt install docker.io

Installing docker:

$sudo snap install docker

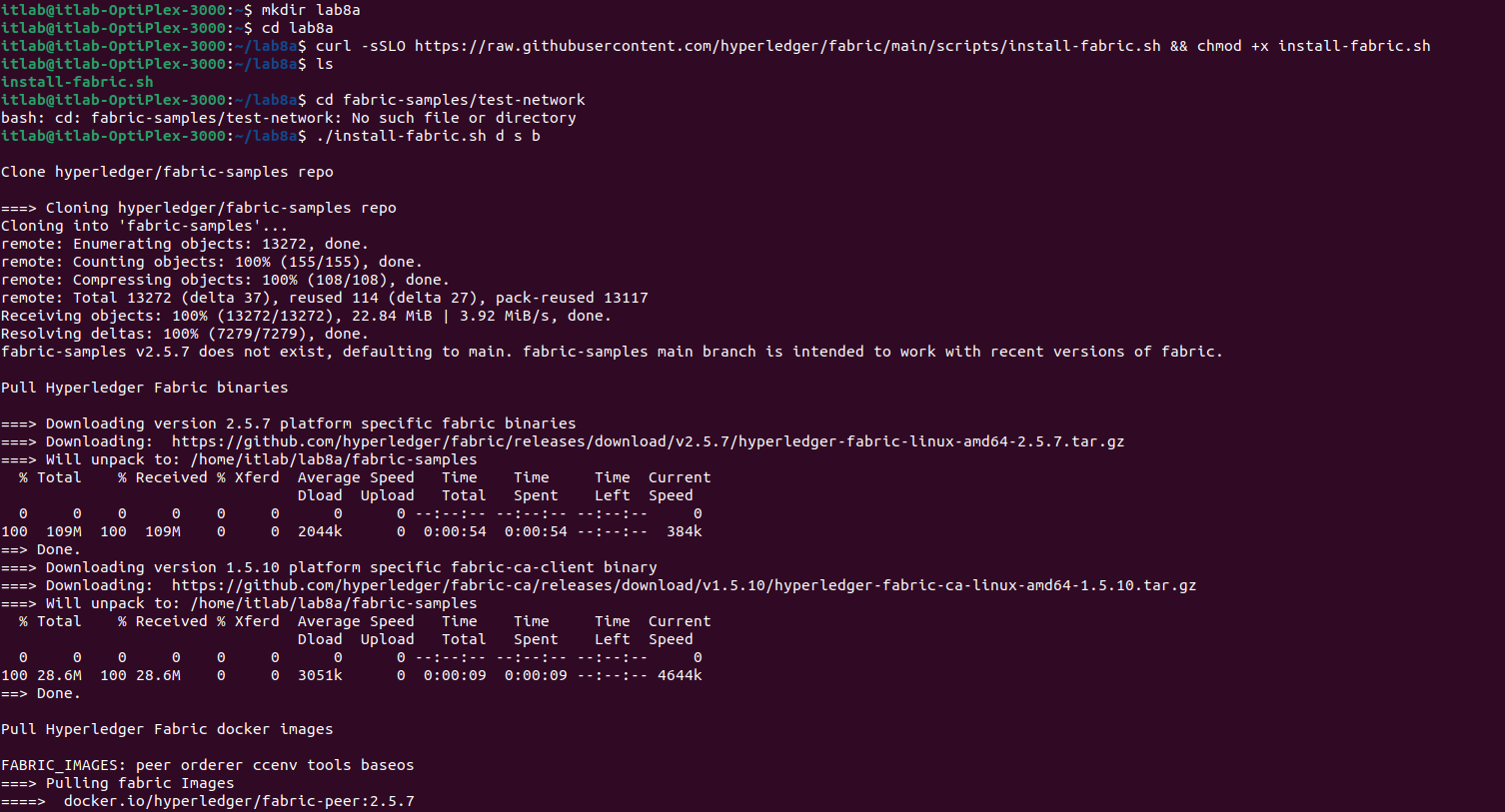
Going into the tuna-app folder:

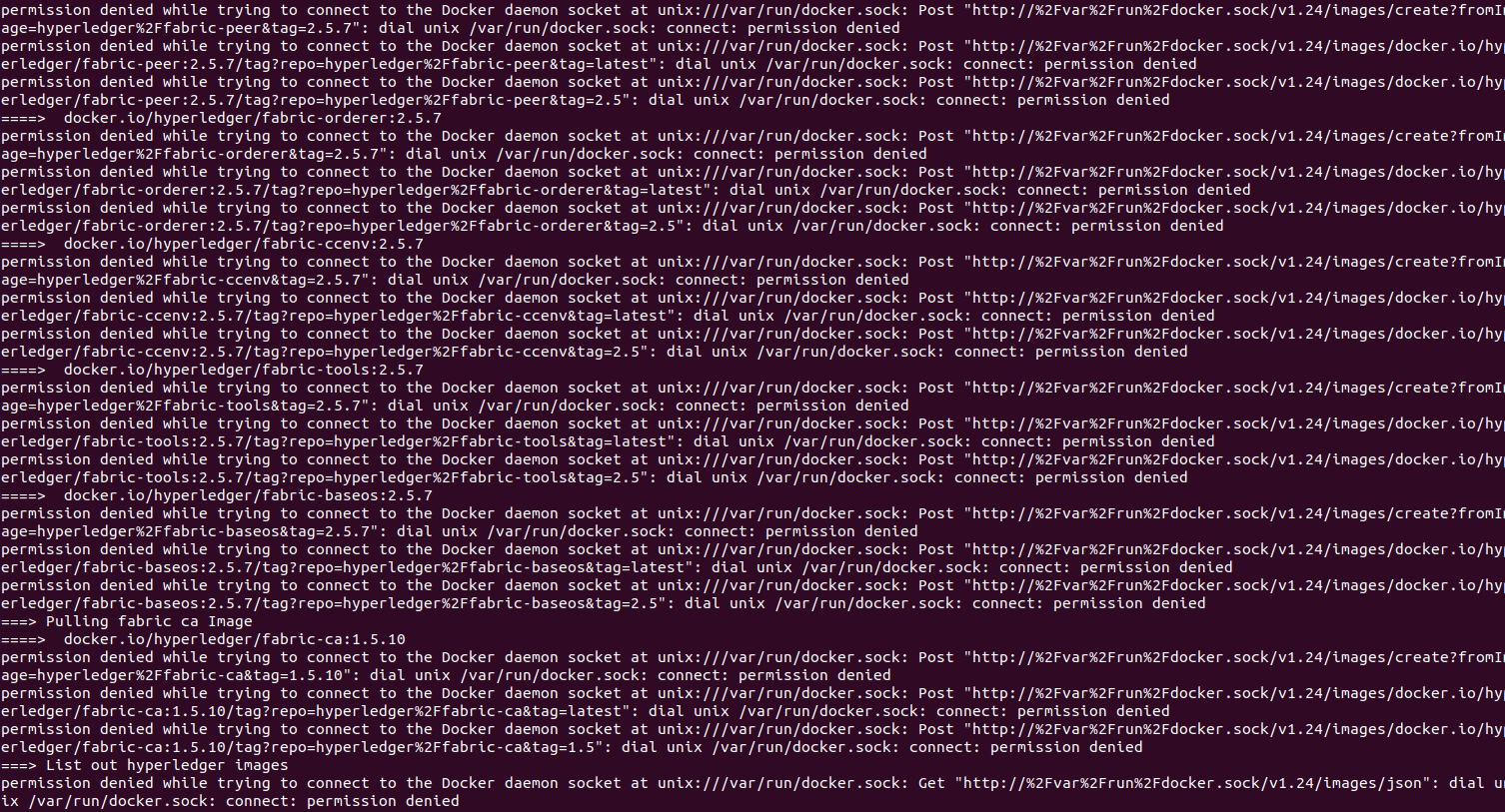
$cd education/LFS171x/fabric-material/tuna-app

Changing permissions for docker.sock since connection gets denied

$sudo chmod 666 /var/run/docker.sock

$./startFacbric.sh





Starting hyperledger fabric, it also starts listening on channels created.

We have to use node v8.9.0

Run the following commands:

nvm install v8.9.0

nvm use v8.9.0

rm -rf node\_modules to delete existing modules

and then

npm install

Running node registerAdmin.js

Certificate gets created.

$node registerAdmin.js

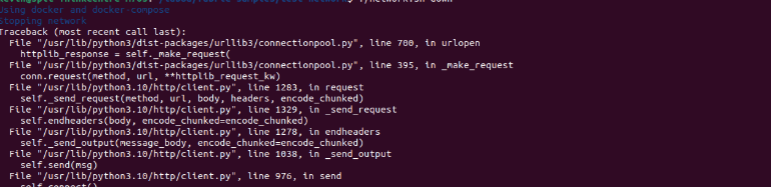
$node registerUser.js

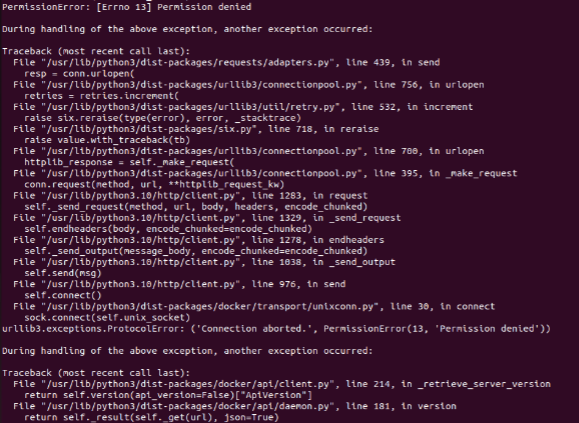
Running registerUser.js, this file registers user with CA.

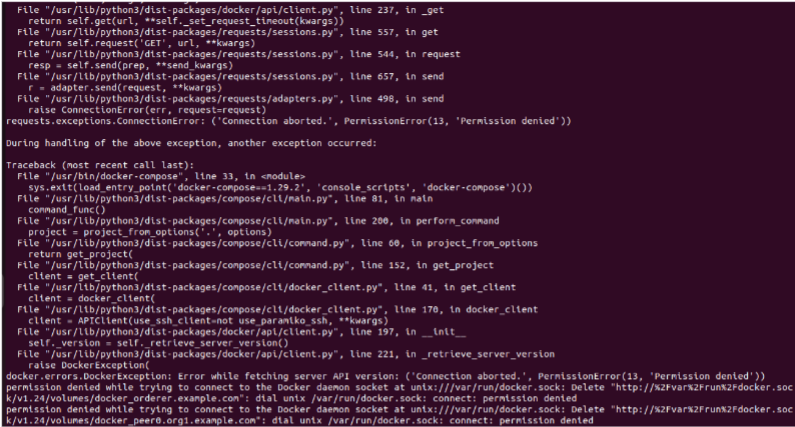
$node server,js

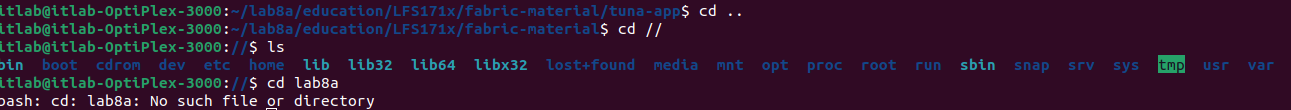
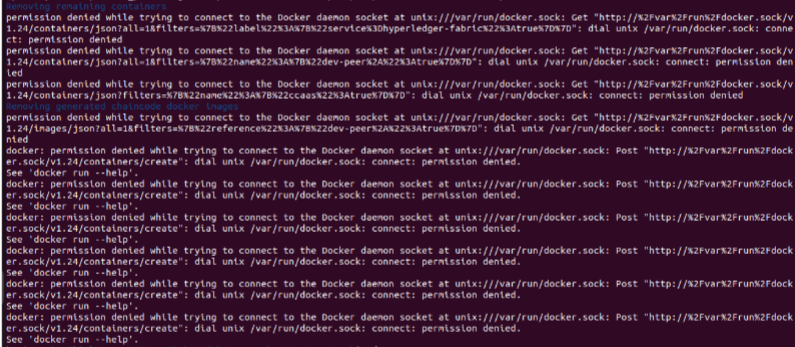
Running server.js

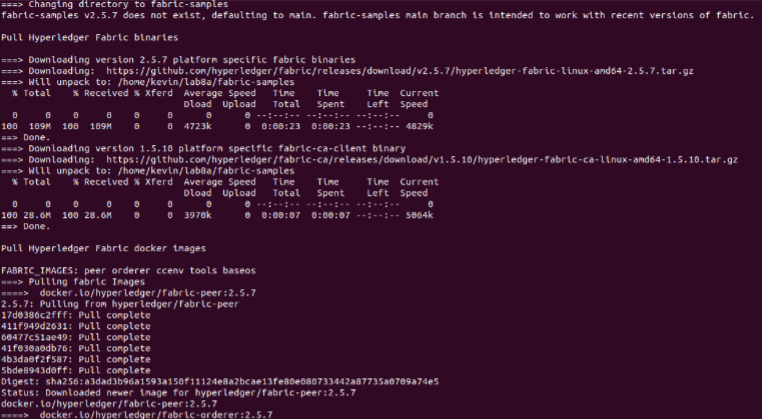


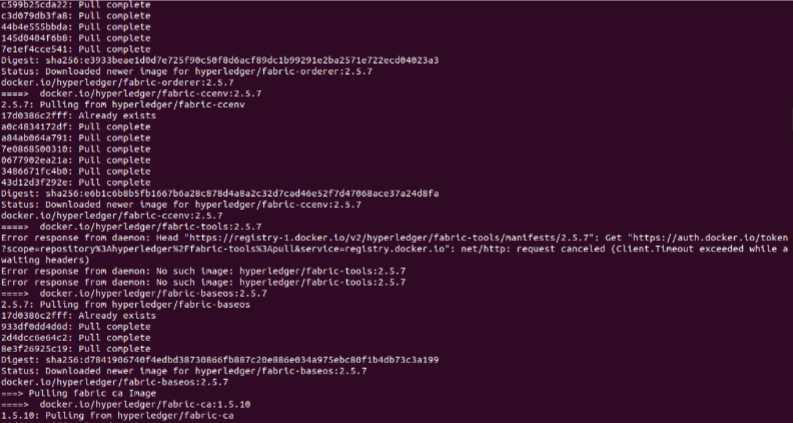




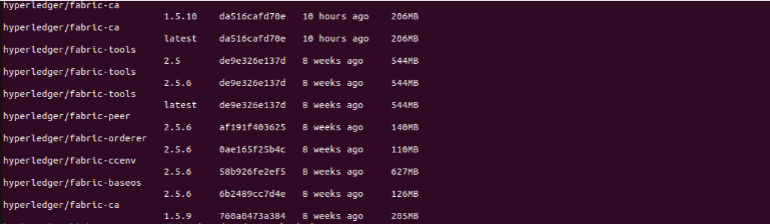
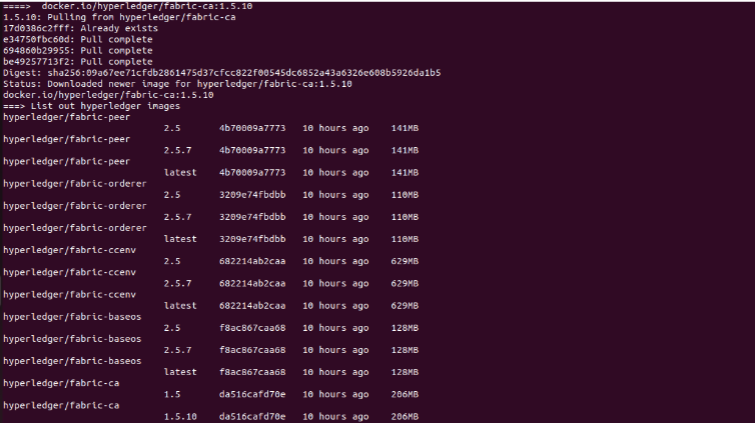


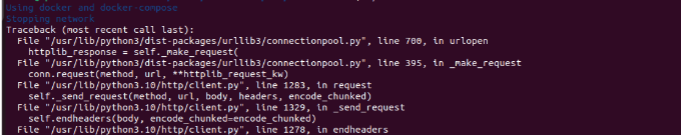




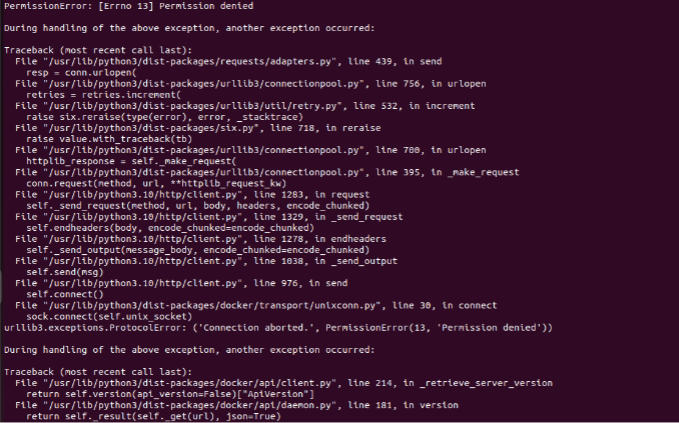


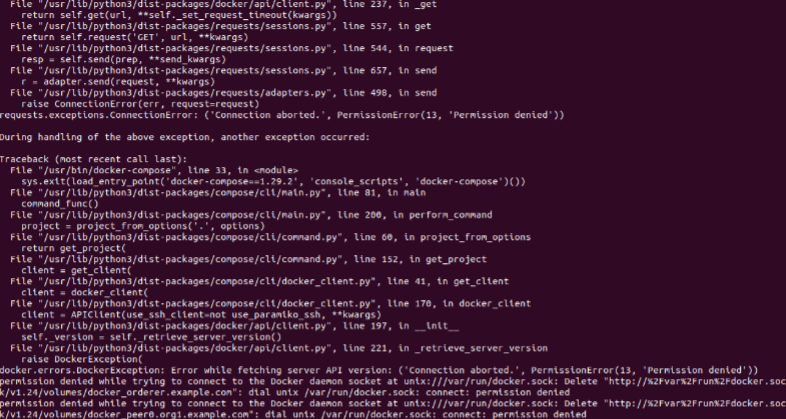






**Conclusion:**





**References:**

[1] <https://hyperledger-fabric.readthedocs.io/en/release-2.0/getting_started.html>

[2] Supply Chain- Tuna Shipment Application

https://github.com/hyperledger/education.git

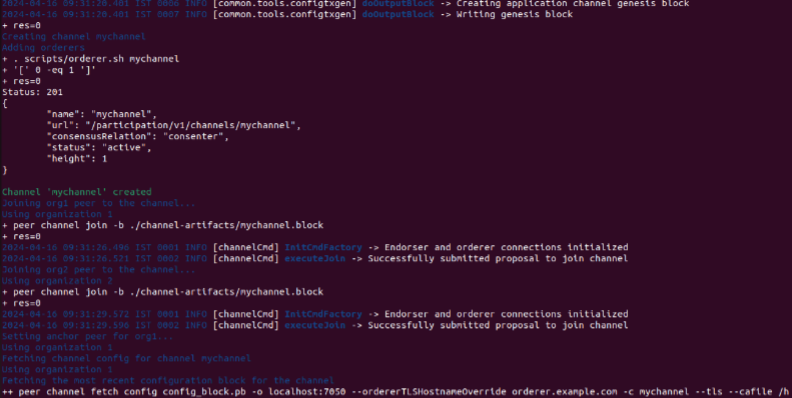
[3] How To Install Node.js on Ubuntu 20.04

[How To Install Node.js on Ubuntu 20.04 | DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04)

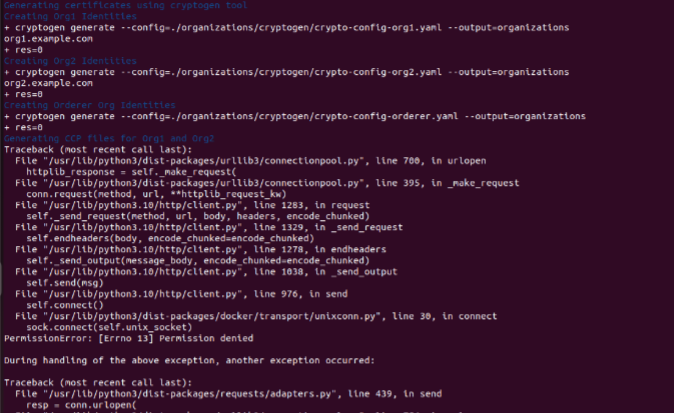
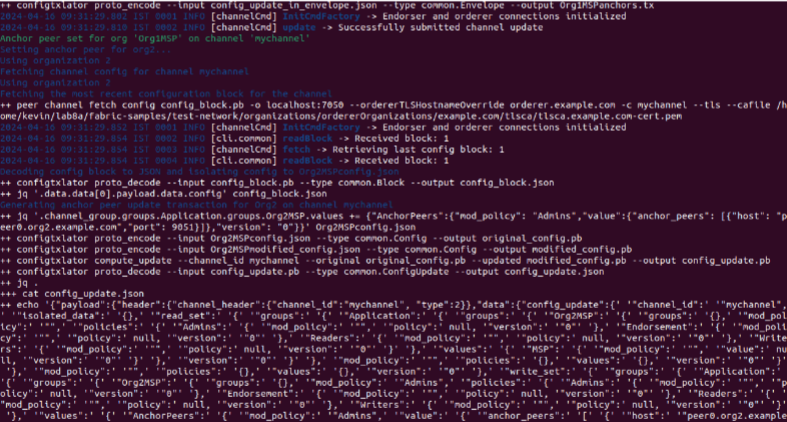


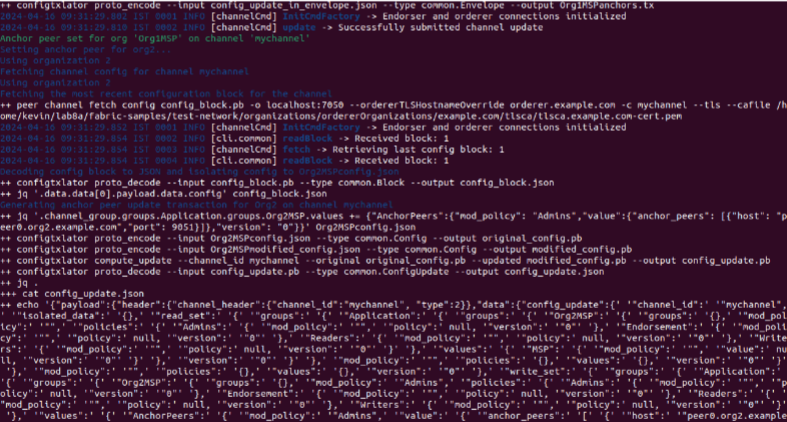


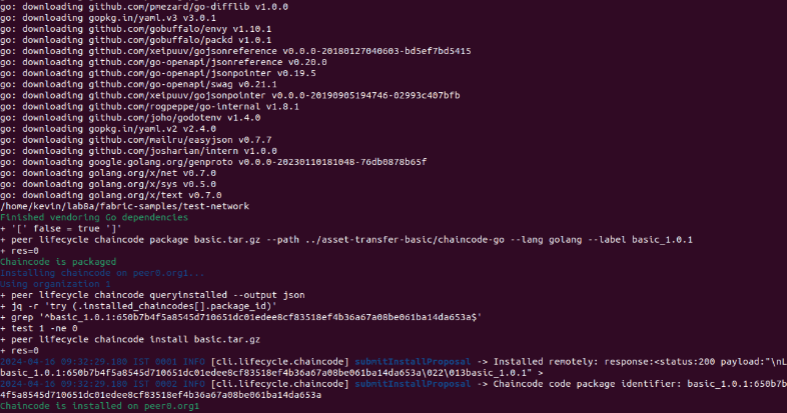


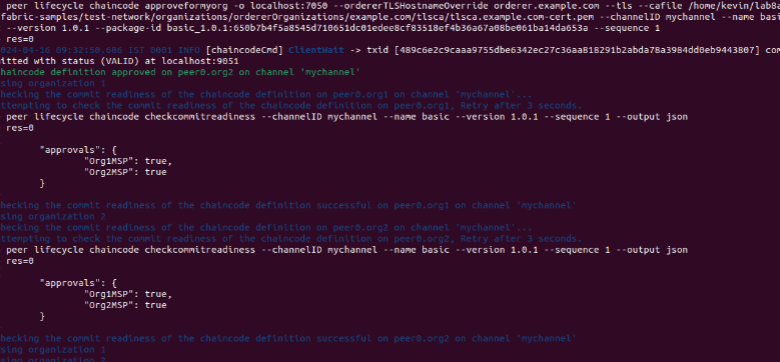


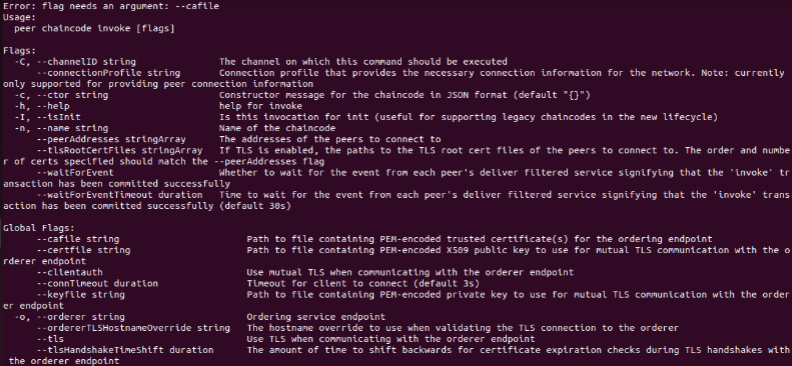
xx

xx 









**Conclusion:**

Therefore, our conclusion was that the experiment provided valuable hands-on experience in deploying Hyperledger Fabric using Docker and Docker-compose. This encompassed understanding its modular architecture and the utilization of containers for smart contract hosting. We also gained insights into permissioned networks, identity management for user authentication, privacy features like private channels, and the critical role of chaincode in enforcing transaction rules. Practical tasks such as registering identities, initializing the network, and building applications with Node.js contributed significantly to comprehending blockchain's potential applications in Supply Chain Management and enterprise solutions.

**References:**

1. Hyperledger Fabric. (n.d.). Getting Started with Hyperledger Fabric. Retrieved from<https://hyperledger-fabric.readthedocs.io/en/release-2.0/getting_started.html>
2. Hyperledger. (n.d.). Tuna Shipment Application - Hyperledger Education. Retrieved from<https://github.com/hyperledger/education.git>
3. DigitalOcean. (n.d.). How To Install Node.js on Ubuntu 20.04. Retrieved from https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04