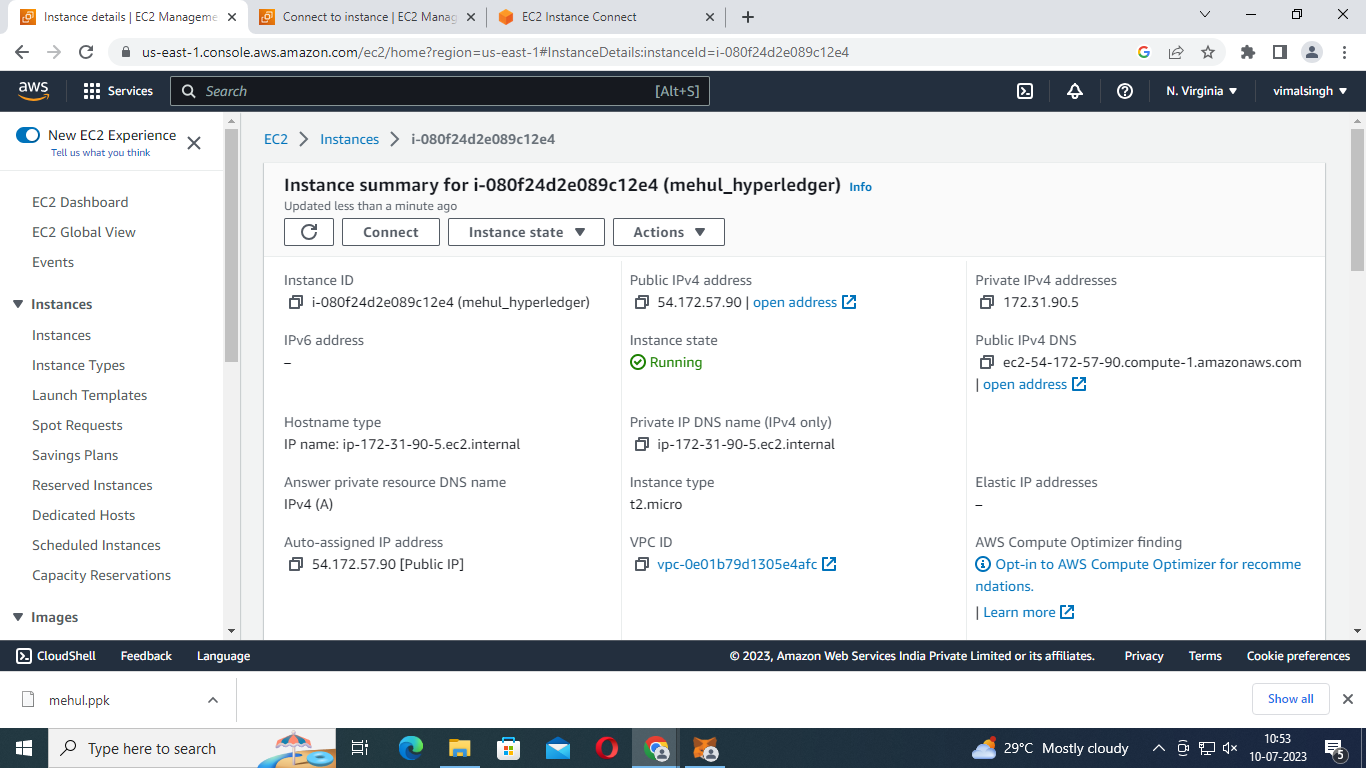
**Hyperledger-Fabric**

Prerequisites:

Create an instance for Ubuntu using AWS



**1.Install Docker**

Sudo apt-get update

sudo apt-get -y install docker-compose

**2. Install Golang-go**

sudo apt install golang-go

**3. Install jq**

sudo apt install jq

**4. Install Node/java**

sudo apt install npm

npm install node

**5. Installing Fabric-samples Repository**

sudo curl -sSLO <https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh>

sudo chmod +x install-fabric.sh

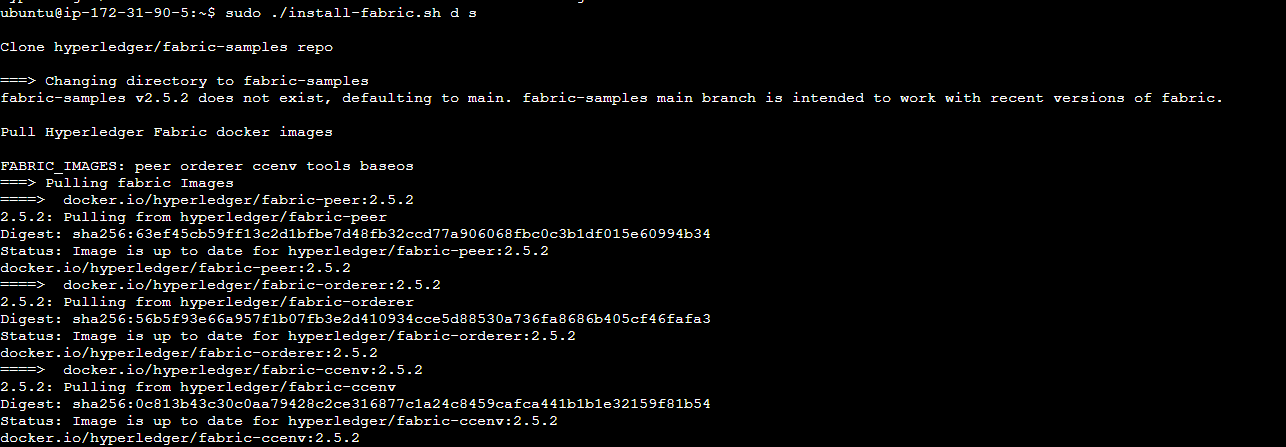
Then you can pull docker containers by running one of these commands:

sudo ./install-fabric.sh docker samples

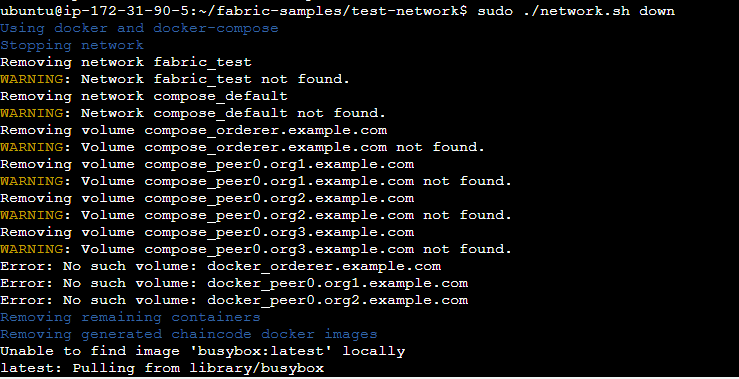
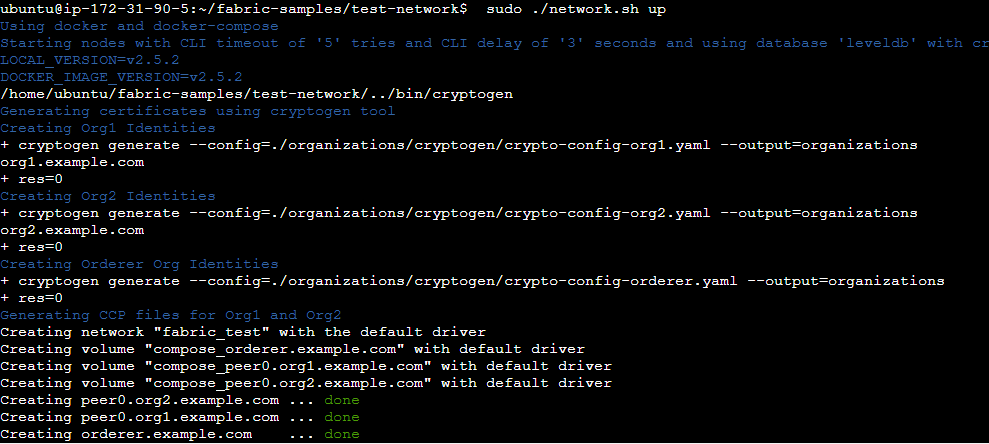
sudo ./install-fabric.sh d s

To install binaries for Fabric samples you can use the command below:

sudo ./install-fabric.sh binary



**Building First Network**

1. cd fabric-samples/test-network
2. sudo ./network.sh down
3. 
4. sudo ./network.sh up
5. 

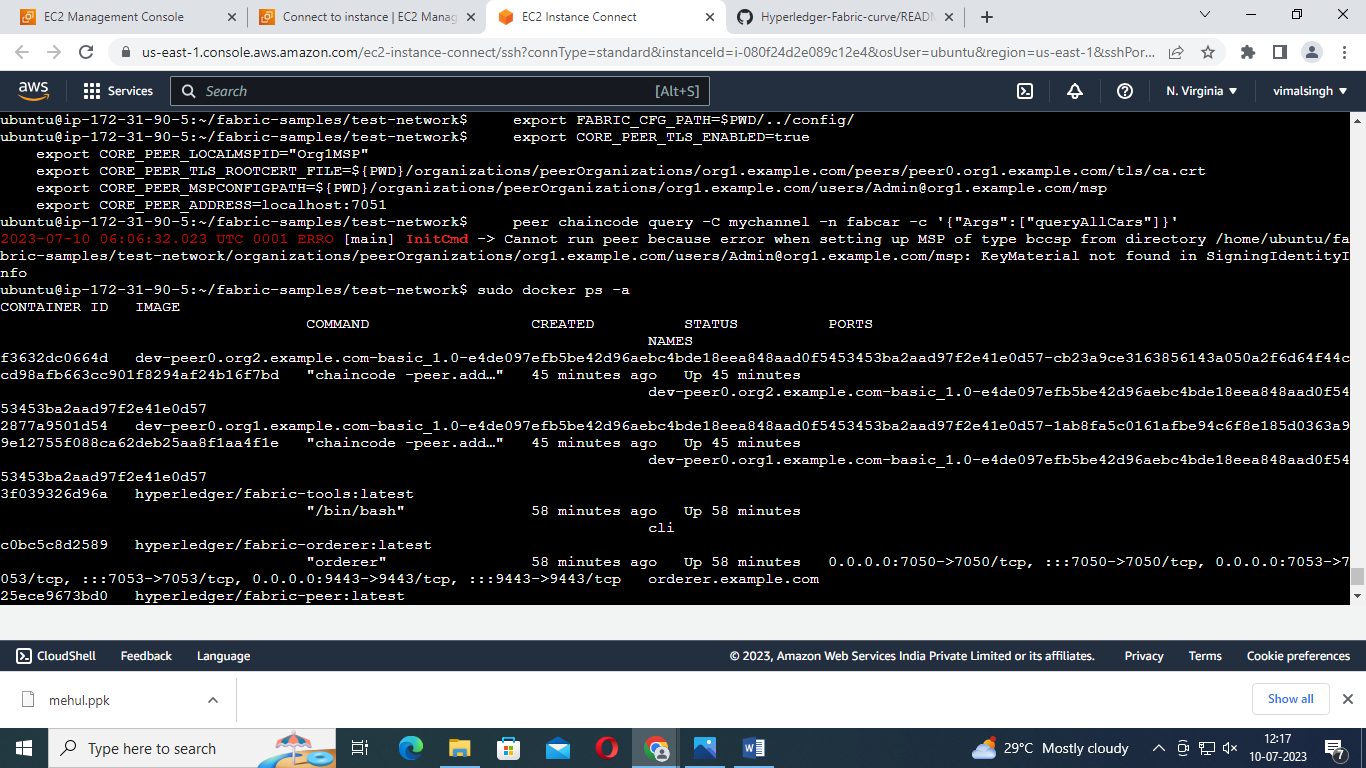
**Now our first Hyperledger Fabric Network is successfully running.**

**The components of the test network:**

Run the below command to list all of Docker containers that are running on your machine:

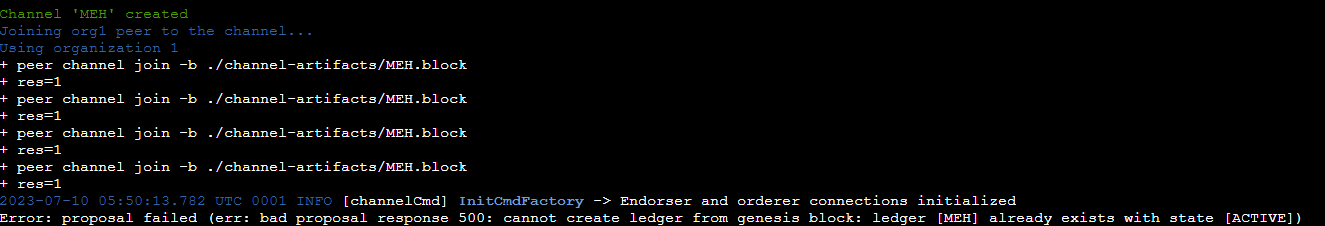
sudo docker ps -a

Peers are the fundamental components of any Fabric network.



**Creating a channel**

sudo ./network.sh createChannel –c meh



**Starting a chaincode on the channel**

sudo ./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go

**Interacting with the network**

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

export FABRIC\_CFG\_PATH=$PWD/../config/

export CORE\_PEER\_TLS\_ENABLED=true

export CORE\_PEER\_LOCALMSPID="Org1MSP"

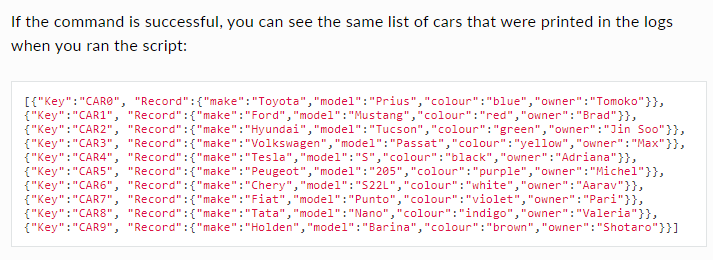
export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:7051

you can now query the ledger from your CLI. Run the following command to get the list of cars that were added to your channel ledger:

peer chaincode query -C mychannel -n fabcar -c '{"Args":["queryAllCars"]}'



Since we already queried the Org1 peer, we can take this opportunity to query the chaincode running on the Org2 peer. Set the following environment variables to operate as Org2 (Run one by one):

export CORE\_PEER\_TLS\_ENABLED=true

export CORE\_PEER\_LOCALMSPID="Org2MSP"

export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

export CORE\_PEER\_ADDRESS=localhost:9051

You can now query the fabcar chaincode running on peer0.org2.example.com:

peer chaincode query -C mychannel -n fabcar -c '{"Args":["queryCar","CAR9"]}'



## Bring down the network

Sudo ./network.sh down