|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version | Command to check version and install | commands |
| Docker | Docker version 26.1.3 | docker –version | - |
| Docker Compose | Docker Compose version v2.18.0 | docker-compose --version | sudo apt install docker-compose-plugin -y |
| Go | go1.22.2 | go version |  |
| Node.js | v18.19.1 | node -v | curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash -  sudo apt install -y nodejs |
| npm | 11.1.0 | npm --version |  |
| Git | git version 2.43.0 | git --version | sudo apt install git -y |
| Python | Python 3.12.3 | python3 --version | sudo apt install python3 python3-pip -y |
| WSL2 | WSL2 (Windows Subsystem for Linux) | *Installed on system* | wsl --set-default-version 2 |
| Hyperledger Fabric | Hyperledger Fabric 3.1.0 | - | curl –sSL <http://bit.ly/2ysbOFE> | bash -s 3.1.0) |

1. Make sure you have all requirements installed.
2. Mkdir scm, make sure you have the bin folder from fabric samples: mv ~/fabric-samples/bin/\* ~/scm/
3. Write your cryptoconfig file and generate certificates using configenx
4. Create ca certificates by running command: cryptogen generate --config=crypto-config.yaml --output="crypto-config”
5. Write the configtx.yaml file
6. Create genesis block and channel : configtxgen -profile SupplyChainGenesis -channelID system-channel -outputBlock ./genesis.block  
    configtxgen -profile SupplyChainChannel -outputCreateChannelTx ./foodscmchannel.tx -channelID foodscmchannel
7. Update anchor peers: eg: configtxgen -profile SupplyChainChannel \

-outputAnchorPeersUpdate ./MiddlemenMSPAnchor.tx \

-channelID foodscmchannel \

-asOrg MiddlemenMSP

1. Export keys

export MANUFACTURER\_CA\_PRIVATE\_KEY=$(cd ./artifacts/network/crypto-config/peerOrganizations/manufacturer.example.com/ca && ls \*\_sk)

export MIDDLEMEN\_CA\_PRIVATE\_KEY=$(cd ./artifacts/network/crypto-config/peerOrganizations/middlemen.example.com/ca && ls \*\_sk)

export CONSUMER\_CA\_PRIVATE\_KEY=$(cd ./artifacts/network/crypto-config/peerOrganizations/consumer.example.com/ca && ls \*\_sk)

1. Up the network: docker-compose -f artifacts/docker-compose.yaml up -d
2. Get into cli and export channel name

Commands for channel and chaincode

peer channel create -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/channel.tx --tls $CORE\_PEER\_TLS\_ENABLED --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

11. peer 0 manufacturer join channel

peer channel join -b supplychainchannel.block

12. peer 0 middlemen join channel

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/users/[Admin@middlemen.example.com](mailto:Admin@middlemen.example.com)/msp CORE\_PEER\_ADDRESS=peer0.middlemen.example.com:8051 CORE\_PEER\_LOCALMSPID="MiddleMenMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/peers/peer0.middlemen.example.com/tls/ca.crt peer channel join -b $CHANNEL\_NAME.block

peer 1 middlemen join channel

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/users/[Admin@middlemen.example.com](mailto:Admin@middlemen.example.com)/msp CORE\_PEER\_ADDRESS=peer1.middlemen.example.com:9051 CORE\_PEER\_LOCALMSPID="MiddleMenMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/peers/peer1.middlemen.example.com/tls/ca.crt peer channel join -b $CHANNEL\_NAME.block

peer 2 middlemen join channel

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/users/[Admin@middlemen.example.com](mailto:Admin@middlemen.example.com)/msp CORE\_PEER\_ADDRESS=peer2.middlemen.example.com:10051 CORE\_PEER\_LOCALMSPID="MiddleMenMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/peers/peer2.middlemen.example.com/tls/ca.crt peer channel join -b $CHANNEL\_NAME.block

peer 0 consumer join channel

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/consumer.example.com/users/[Admin@consumer.example.com](mailto:Admin@consumer.example.com)/msp CORE\_PEER\_ADDRESS=peer0.consumer.example.com:11051 CORE\_PEER\_LOCALMSPID="ConsumerMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/consumer.example.com/peers/peer0.consumer.example.com/tls/ca.crt peer channel join -b $CHANNEL\_NAME.block

13. update anchor peers

peer 0 manufacturer

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/manufacturer.example.com/users/[Admin@manufacturer.example.com](mailto:Admin@manufacturer.example.com)/msp CORE\_PEER\_ADDRESS=peer0.manufacturer.example.com:7051 CORE\_PEER\_LOCALMSPID="ManufacturerMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/manufacturer.example.com/peers/peer0.manufacturer.example.com/tls/ca.crt peer channel update -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/ManufacturerMSPanchors.tx --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer 0 middlemen

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/users/[Admin@middlemen.example.com](mailto:Admin@middlemen.example.com)/msp CORE\_PEER\_ADDRESS=peer0.middlemen.example.com:8051 CORE\_PEER\_LOCALMSPID="MiddleMenMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/middlemen.example.com/peers/peer0.middlemen.example.com/tls/ca.crt peer channel update -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/MiddleMenMSPanchors.tx --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

peer 0 consumer

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/consumer.example.com/users/[Admin@consumer.example.com](mailto:Admin@consumer.example.com)/msp CORE\_PEER\_ADDRESS=peer0.consumer.example.com:11051 CORE\_PEER\_LOCALMSPID="ConsumerMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/consumer.example.com/peers/peer0.consumer.example.com/tls/ca.crt peer channel update -o orderer.example.com:7050 -c $CHANNEL\_NAME -f ./channel-artifacts/ConsumerMSPanchors.tx --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem

14. install chaincode

peer 0 manufacturer

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/manufacturer.example.com/users/[Admin@manufacturer.example.com](mailto:Admin@manufacturer.example.com)/msp CORE\_PEER\_ADDRESS=peer0.manufacturer.example.com:7051 CORE\_PEER\_LOCALMSPID="ManufacturerMSP" CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/manufacturer.example.com/peers/peer0.manufacturer.example.com/tls/ca.crt peer chaincode install -n supplychaincc -v 1.0 -p github.com/chaincode/

15. instantiate or upgrade chaincode

peer chaincode instantiate -o orderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -l golang -v 1.0 -c '{"Args":[""]}' -P "OR ('ManufacturerMSP.peer','MiddleMenMSP.peer', 'ConsumerMSP.peer')"

peer chaincode upgrade -o orderer.example.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -l golang -v 1.2 -c '{"Args":[""]}' -P "OR ('ManufacturerMSP.peer','MiddleMenMSP.peer', 'ConsumerMSP.peer')"

16. invoke chaincode

create users

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["createUser","tanisha1","[tanisha@gmail.com](mailto:kk@asdf.asdf)","manufacturer","mumbai"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["createUser","tanisha2","[tanisha@gmail.com](mailto:kk@asdf.asdf)","wholesaler","mumbai"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["createUser","tanisha3","[tanisha@gmail.com](mailto:kk@asdf.asdf)","distributor","mumbai"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["createUser","tanisha4","[tanisha@gmail.com](mailto:kk@asdf.asdf)","retailer","mumbai"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["createUser","tanisha5","[tanisha@gmail.com](mailto:kk@asdf.asdf)","consumer","mumbai"]}'

create product

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["createProduct","Kohinoor diamond","User1","1000"]}'

update product

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["updateProduct","Product1","User1","Kohinoor heera","1500"]}'

query product

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["queryAsset","Product1"]}'

sending product

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["sendToWholesaler","Product1","User2"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["sendToDistributer","Product1","User3"]}'

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["sendToRetailer","Product1","User4"]}'

order product

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["orderProduct","User5","Product1"]}'

sell to consumer product

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["sellToConsumer","Product1"]}'

delivered product

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["deliveredProduct","Product1"]}'

query asset

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["queryAsset","Product1"]}'

query all

peer chaincode invoke -o orderer.example.com:7050 --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C $CHANNEL\_NAME -n supplychaincc -c '{"Args":["queryAll","User"]}'

17. down the network

docker-compose -f artifacts/docker-compose.yaml down -v

sudo rm -fR artifacts/network/

docker kill $(docker ps -aq)

docker rm $(docker ps -aq)

docker ps

Summary of steps

1. Generate certificates by CA
2. Create Genesis block and channel artifacts
3. Export channel path
4. Create and update anchor peers
5. Bring up docker network
6. Create and join channel
7. Initialize deploy and invoke chaincode
8. Down the network