| 192.168.25.102 | ccd079 | ubuntu@079 |
| --- | --- | --- |
| 192.168.25.103 | ccd080 | ubuntu@080 |
| 192.168.25.104 | ccd081 | ubuntu@081 |
| 192.168.25.105 | ccd082 | ubuntu@082 |

**To access VMs:**

Say your VM is ccd079 and it has IP 192.168.25.102

> ssh ccd079@192.168.25.102

It will then ask you password. Enter the password and you are good to go.

**Docker and Docker Compose Installation**

sudo apt-get install ca-certificates curl && sudo install -m 0755 -d /etc/apt/keyrings && sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc && sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

sudo chmod 777 /var/run/docker.sock

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

Cd pee

====

**Firewall Install and open ports for Docker Swarm**

sudo apt -y install firewalld

sudo systemctl start firewalld

sudo systemctl enable firewalld

sudo firewall-cmd --add-port=2376/tcp --permanent

sudo firewall-cmd --add-port=2377/tcp --permanent

sudo firewall-cmd --add-port=7946/tcp --permanent

sudo firewall-cmd --add-port=7946/udp --permanent

sudo firewall-cmd --add-port=4789/udp --permanent

sudo firewall-cmd --add-port=3001/tcp --permanent

sudo firewall-cmd --reload

sudo systemctl restart docker

sudo chmod 777 /var/run/docker.sock

=======

**Install Hyperledger**

curl -sSLO https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh && chmod +x install-fabric.sh

./install-fabric.sh docker samples binary

=====

**Create Docker Swarm**

docker swarm init --advertise-addr 192.168.25.102

docker swarm join-token manager

docker swarm join --token SWMTKN-1-0csduruek81aezmeugg215f6yzak2sdugwthkse31ayi0kv88c-2jw3abd76a1yshwm91xzavnyl 192.168.25.102:2377 --advertise-addr 192.168.25.103

docker swarm join --token SWMTKN-1-0csduruek81aezmeugg215f6yzak2sdugwthkse31ayi0kv88c-2jw3abd76a1yshwm91xzavnyl 192.168.25.102:2377 --advertise-addr 192.168.25.104

docker swarm join --token SWMTKN-1-0csduruek81aezmeugg215f6yzak2sdugwthkse31ayi0kv88c-2jw3abd76a1yshwm91xzavnyl 192.168.25.102:2377 --advertise-addr 192.168.25.105

docker network create --attachable --driver overlay first-network

=====

**Make the containers up**

========

**Install ethtool**

sudo apt-get install ethtool

sudo apt-get install net-tools

**Run ifconfig to see all containers**

ifconfig

**Bypass TCP Checksum of each interface**

sudo ethtool -K <interface> tx off

=========

**Clone the git file**

Go inside fabric samples directory

git clone https://github.com/ankit-gupta-24/4host-swarm.git

======

Bring up all containers on all 4 VMs

**Make mychannel up on 1st container**

./mychannelup.sh

**To check blockchain height**

docker exec peer0.org1.example.com peer channel getinfo -c mychannel

docker exec peer1.org1.example.com peer channel getinfo -c mychannel

docker exec peer0.org2.example.com peer channel getinfo -c mychannel

docker exec peer1.org2.example.com peer channel getinfo -c mychannel

========

**To go inside cli container**

docker exec -it cli bash

apt-get update

apt install git-all

apt install npm

**Clone the chaincode**

Git clone <https://github.com/hrishi-13/HMIS-dApp>

**To change directory**

pushd ./chaincode-javascript

**To install dependencies of javascript**

npm install

**To come out of directory change by pushd**

popd

**To package and make tar file**

peer lifecycle chaincode package kycjs.tar.gz --path /opt/gopath/src/github.com/hyperledger/fabric/peer/chaincode-javascript --label kycjs\_1 --lang node

**exit cli**

======

**Install chaincode on every peer**

# peer0.org1

docker exec cli peer lifecycle chaincode install kycjs.tar.gz

Chaincode code package identifier: kycjs\_1:1dd0576fbd84e57c0d3db302ad3a8a8018197787f395da5ababd98da3109e8c1

# peer1.org1

docker exec -e CORE\_PEER\_ADDRESS=peer1.org1.example.com:8051 -e CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer1.org1.example.com/tls/ca.crt cli peer lifecycle chaincode install kycjs.tar.gz

# peer0.org2

docker exec -e CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp -e CORE\_PEER\_ADDRESS=peer0.org2.example.com:9051 -e CORE\_PEER\_LOCALMSPID="Org2MSP" -e CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt cli peer lifecycle chaincode install kycjs.tar.gz

# peer1.org2

docker exec -e CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp -e CORE\_PEER\_ADDRESS=peer1.org2.example.com:10051 -e CORE\_PEER\_LOCALMSPID="Org2MSP" -e CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer1.org2.example.com/tls/ca.crt cli peer lifecycle chaincode install kycjs.tar.gz

======

**To check the package image container exists or not**

docker images dev-\*

**To approve the chaincode**

# for org1

docker exec cli peer lifecycle chaincode approveformyorg --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 --channelID mychannel --name kycjs --version 1 --sequence 1 --waitForEvent --package-id kycjs\_1:1dd0576fbd84e57c0d3db302ad3a8a8018197787f395da5ababd98da3109e8c1

# for org2

docker exec -e CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp -e CORE\_PEER\_ADDRESS=peer0.org2.example.com:9051 -e CORE\_PEER\_LOCALMSPID="Org2MSP" -e CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt cli peer lifecycle chaincode approveformyorg --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 --channelID mychannel --name kycjs --version 1 --sequence 1 --waitForEvent --package-id kycjs\_1:1dd0576fbd84e57c0d3db302ad3a8a8018197787f395da5ababd98da3109e8c1

**To check approval status**

docker exec cli peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name kycjs --version 1 --sequence 1

**Commit chaincode**

docker exec cli peer lifecycle chaincode commit -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 --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt --channelID mychannel --name kycjs --version 1 --sequence 1

**To check commit status on all peers**

docker exec cli peer lifecycle chaincode querycommitted --channelID mychannel --name kycjs

docker exec cli2 peer lifecycle chaincode querycommitted --channelID mychannel --name kycjs

docker exec cli3 peer lifecycle chaincode querycommitted --channelID mychannel --name kycjs

docker exec cli4 peer lifecycle chaincode querycommitted --channelID mychannel --name kycjs