Setting up a Hyperledger Fabric Network

The following are prerequisites for installing the required development tools:

- Operating Systems: Ubuntu Linux 14.04 / 16.04 LTS (both 64-bit), or Mac OS 10.12
- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Node: 8.9 or higher (note version 9 is not supported)
- npm: v5.x
- git: 2.9.x or higher
- Python: 2.7.x
- A code editor of your choice, we recommend VSCode.

****If installing Hyperledger Composer using Linux, be aware of the following advice:**

- Login as a normal user, rather than root.
- Do not use sudo su to root.
- When installing prerequisites, use curl, then unzip using sudo.
- Run prereqs-ubuntu.sh as a normal user. It may prompt for root password as some of its actions are required to be run as root.
- Do not use npm with sudo or su to root to use it.
- Avoid installing node globally as root.**

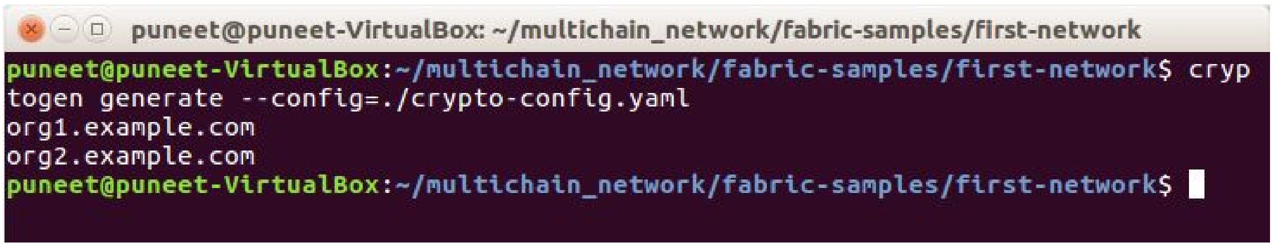
Prerequisites

- Download using - curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh
- Give permissions - chmod u+x prereqs-ubuntu.sh
- Run Script - ./prereqs-ubuntu.sh (restart system after it)
- Essential CLI tools - npm install -g composer-cli@0.19

Steps -

1. Create a directory - mkdir multichain_network
    a. cd multichain_network
    b. curl -sSL http://bit.ly/2ysbOFE | bash -s 1.4.0
    c. Copy bin folder from fabric-samples and paste it in first-network folder
    d. export PATH=<path to download location>/multichain_network/fabric-samples/first-network/bin:$PATH

2. Generate Certificates -
    a. cd first-network
    b. cryptogen generate --config=./crypto-config.yaml
        i. This will create all certificates for orderers and peers in crypto-config folder.



```
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ cryp
togen generate --config=./crypto-config.yaml
org1.example.com
org2.example.com
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$
```

puneet@puneet-VirtualBox: ~/multichain_network/fabric-samples/first-network

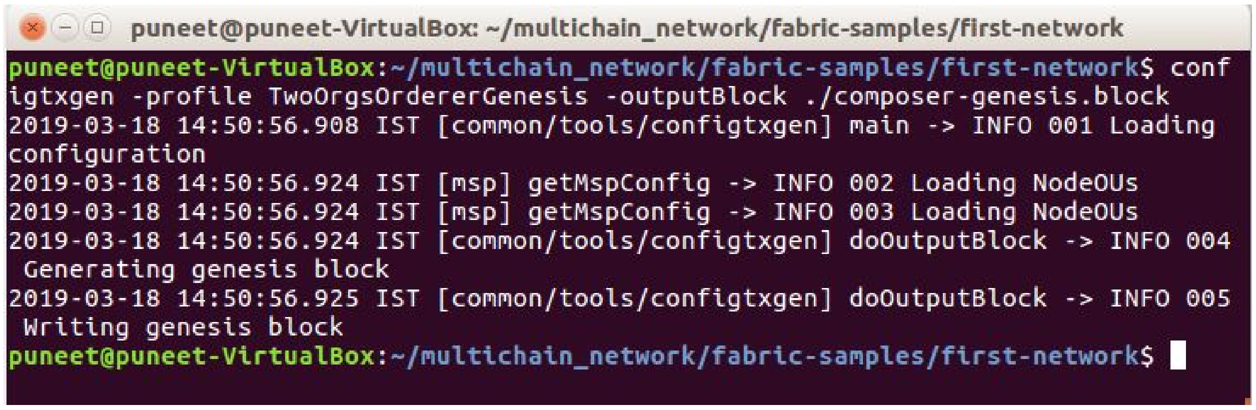
    c. export FABRIC_CFG_PATH=$PWD
    d. Copy certificates for all peers and orderer to temporary folder.
        i. In first-network folder run this command - mkdir -p tmp/composer/org1
        ii. awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt > ./tmp/composer/org1/ca-org1.txt
        iii. awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' ./crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/ca.crt > ./tmp/composer/ca-orderer.txt

iv. export ORG1=./crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

v. cp -p $ORG1/signcerts/A*.pem ./tmp/composer/org1

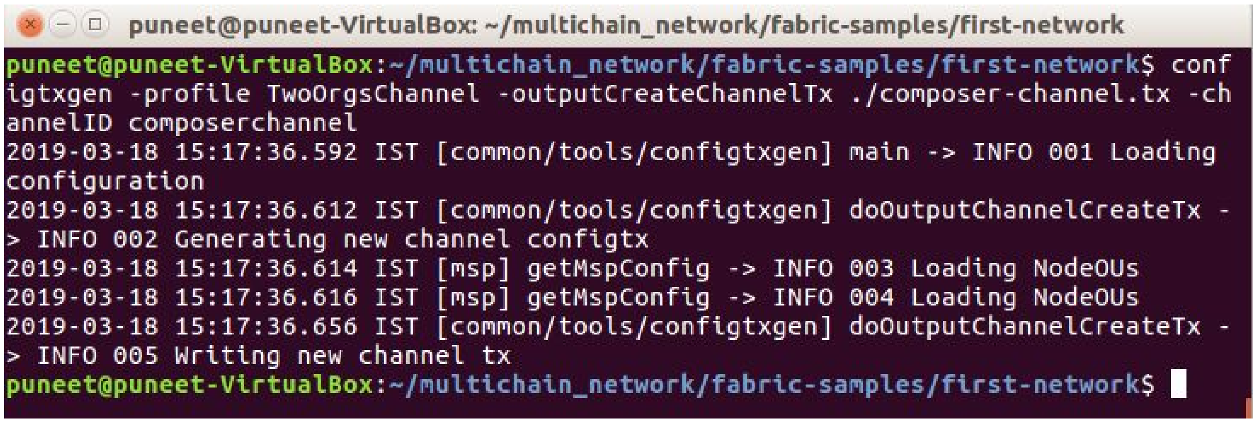
vi. cp -p $ORG1/keystore/*_sk ./tmp/composer/org1

3. Create genesis block and channeltx
   a. configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./composer-genesis.block



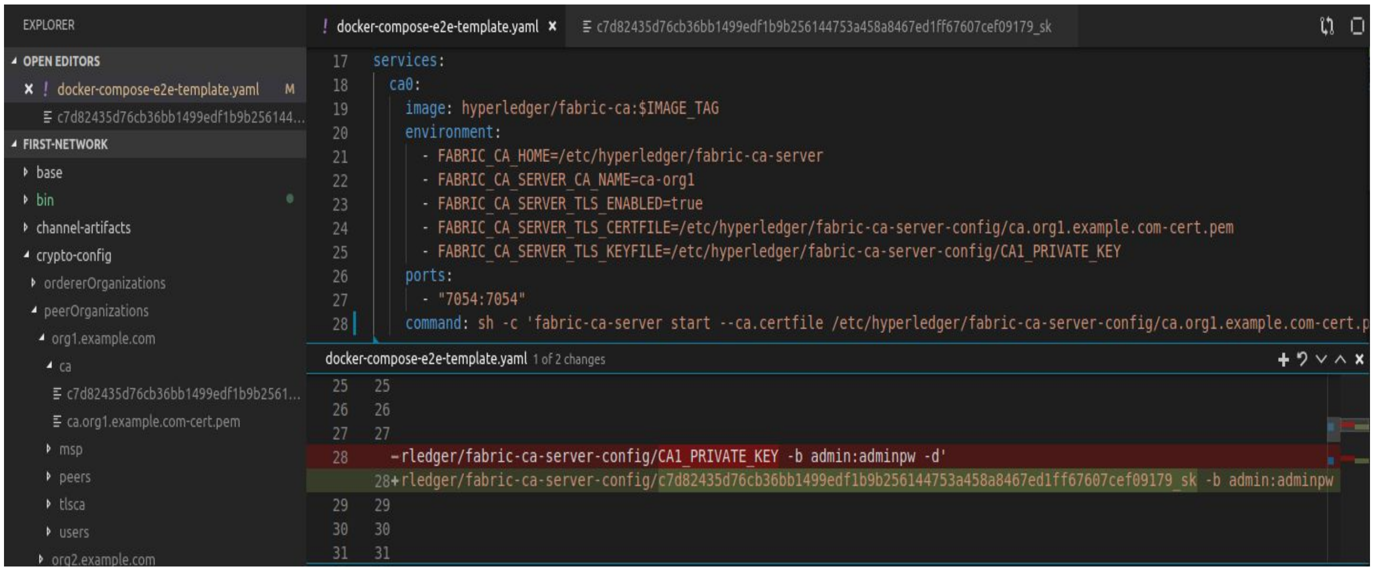
```
puneet@puneet-VirtualBox: ~/multichain_network/fabric-samples/first-network
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ conf
igtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./composer-genesis.block
2019-03-18 14:50:56.908 IST [common/tools/configtxgen] main -> INFO 001 Loading
configuration
2019-03-18 14:50:56.924 IST [msp] getMspConfig -> INFO 002 Loading NodeOUs
2019-03-18 14:50:56.924 IST [msp] getMspConfig -> INFO 003 Loading NodeOUs
2019-03-18 14:50:56.924 IST [common/tools/configtxgen] doOutputBlock -> INFO 004
 Generating genesis block
2019-03-18 14:50:56.925 IST [common/tools/configtxgen] doOutputBlock -> INFO 005
 Writing genesis block
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ 
```

   b. configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./composer-channel.tx -channelID composerchannel



```
puneet@puneet-VirtualBox: ~/multichain_network/fabric-samples/first-network
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ conf
igtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./composer-channel.tx -ch
annelID composerchannel
2019-03-18 15:17:36.592 IST [common/tools/configtxgen] main -> INFO 001 Loading
configuration
2019-03-18 15:17:36.612 IST [common/tools/configtxgen] doOutputChannelCreateTx -
> INFO 002 Generating new channel configtx
2019-03-18 15:17:36.614 IST [msp] getMspConfig -> INFO 003 Loading NodeOUs
2019-03-18 15:17:36.616 IST [msp] getMspConfig -> INFO 004 Loading NodeOUs
2019-03-18 15:17:36.656 IST [common/tools/configtxgen] doOutputChannelCreateTx -
> INFO 005 Writing new channel tx
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ 
```

4. Update CA keys in docker composer file
    a. Open project in vscode - goto first-network folder and run command in terminal “code .”
    b. We have to change CA keys in “docker-compose-e2e-template.yaml” file, therefore navigate to this file in vscode.
    c. Under services section we have two certificate authorities named “ca0” and “ca1”.
    d. For **ca0** - goto command section under ca0 and find CA1_PRIVATE_KEY and replace it with private key - **C7d82435d76cb36bb1499edf1b9b256144753a458a8467ed1ff67607cef09179_sk**
    This key can be found in - “first-network/crypto-config/peerOrganizations/org1.example.com/ca/c7d82435d76cb36bb1499edf1b9b256144753a458a8467ed1ff67607cef09179_sk”


EXPLORER
OPEN EDITORS
docker-compose-e2e-template.yaml M
c7d82435d76cb36bb1499edf1b9b256144...
FIRST-NETWORK
base
bin
channel-artifacts
crypto-config
ordererOrganizations
peerOrganizations
org1.example.com
ca
c7d82435d76cb36bb1499edf1b9b2561...
ca.org1.example.com-cert.pem
msp
peers
tlsca
users
org2.example.com
docker-compose-e2e-template.yaml
c7d82435d76cb36bb1499edf1b9b256144753a458a8467ed1ff67607cef09179_sk
docker-compose-e2e-template.yaml 1 of 2 changes

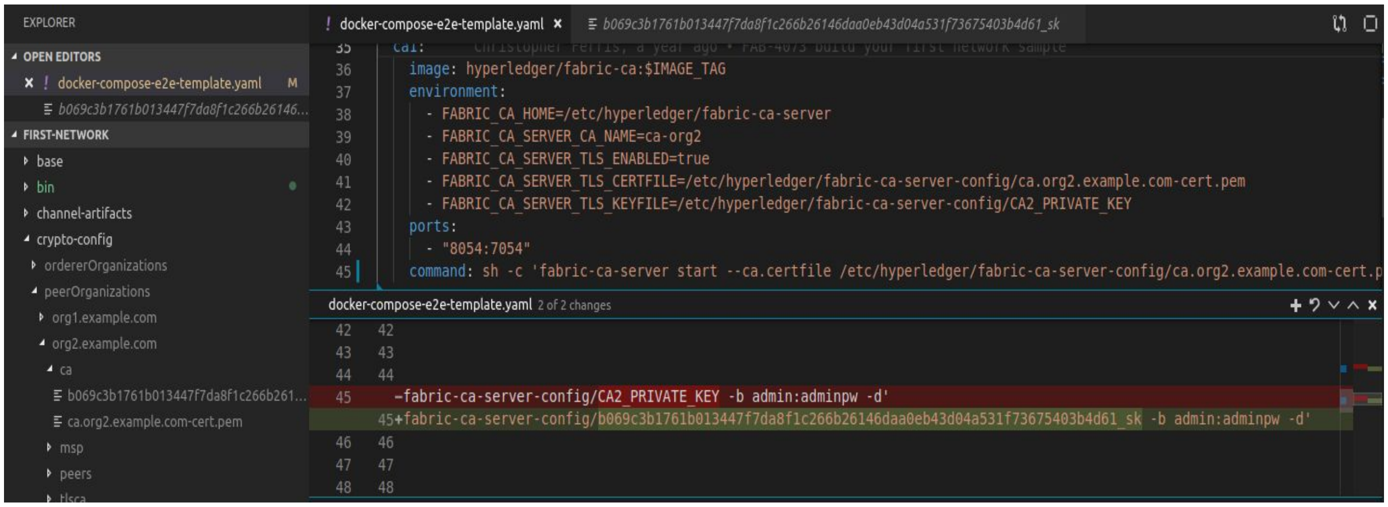
```
services:
  ca0:
    image: hyperledger/fabric-ca:$IMAGE_TAG
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-org1
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_TLS_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.pem
      - FABRIC_CA_SERVER_TLS_KEYFILE=/etc/hyperledger/fabric-ca-server-config/CA1_PRIVATE_KEY
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.p
```

```
-rledger/fabric-ca-server-config/CA1_PRIVATE_KEY -b admin:adminpw -d'
+rledger/fabric-ca-server-config/c7d82435d76cb36bb1499edf1b9b256144753a458a8467ed1ff67607cef09179_sk -b admin:adminpw
```

    e. For **ca1** - goto command section under ca1 and find CA2_PRIVATE_KEY and replace it with private key -

b069c3b1761b013447f7da8f1c266b26146daa0eb43d04a531f73675403b4d61_sk

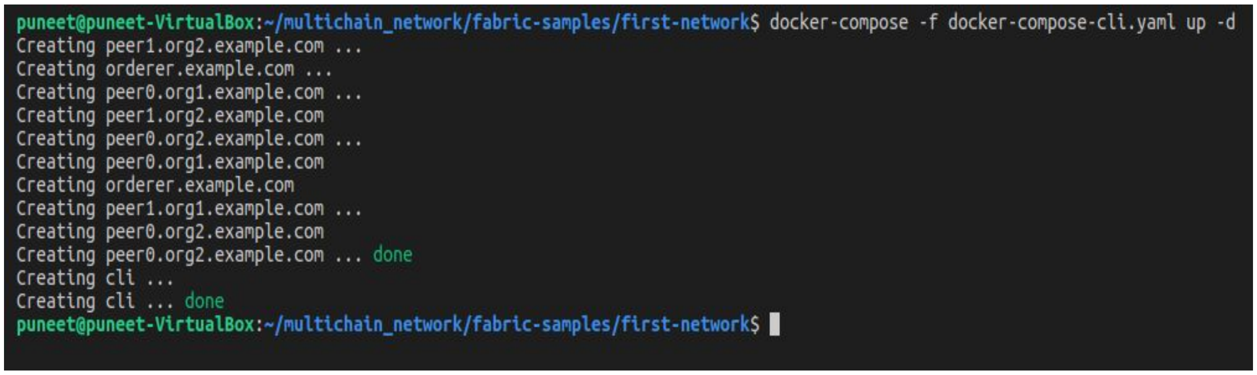
This key can be found in - "first-network/crypto-config/peerOrganizations/org1.example.com/ca/b069c3b1761b013447f7da8f1c266b26146daa0eb43d04a531f73675403b4d61_sk


EXPLORER
OPEN EDITORS
docker-compose-e2e-template.yaml M
b069c3b1761b013447f7da8f1c266b26146...
FIRST-NETWORK
base
bin
channel-artifacts
crypto-config
ordererOrganizations
peerOrganizations
org1.example.com
org2.example.com
ca
b069c3b1761b013447f7da8f1c266b261...
ca.org2.example.com-cert.pem
msp
peers
tlsca
docker-compose-e2e-template.yaml
b069c3b1761b013447f7da8f1c266b26146daa0eb43d04a531f73675403b4d61_sk
docker-compose-e2e-template.yaml 2 of 2 changes

```
  ca1:        Christopher Ferris, a year ago • FAB-4073 build your first network sample
    image: hyperledger/fabric-ca:$IMAGE_TAG
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-org2
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_TLS_CERTFILE=/etc/hyperledger/fabric-ca-server-config/ca.org2.example.com-cert.pem
      - FABRIC_CA_SERVER_TLS_KEYFILE=/etc/hyperledger/fabric-ca-server-config/CA2_PRIVATE_KEY
    ports:
      - "8054:7054"
    command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-config/ca.org2.example.com-cert.p
```

```
-fabric-ca-server-config/CA2_PRIVATE_KEY -b admin:adminpw -d'
+fabric-ca-server-config/b069c3b1761b013447f7da8f1c266b26146daa0eb43d04a531f73675403b4d61_sk -b admin:adminpw -d'
```

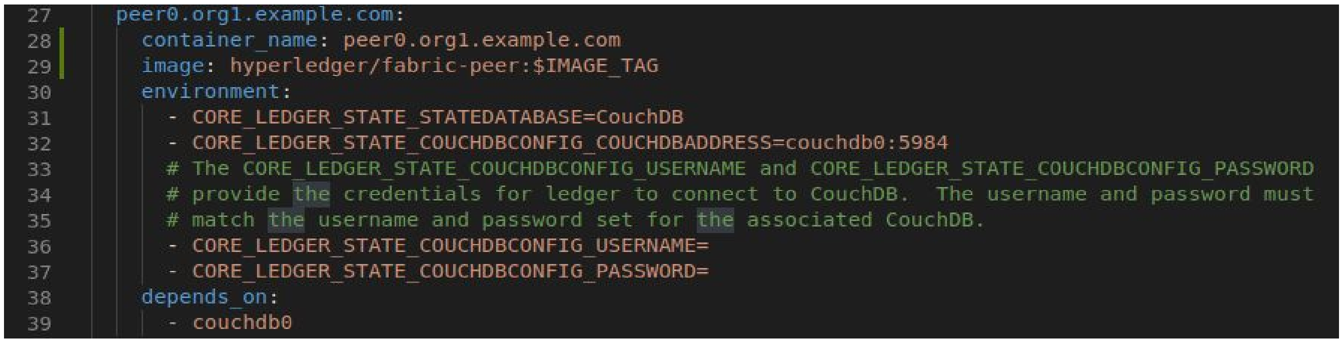
5. Start Fabric -
    a. docker-compose -f docker-compose-cli.yaml up -d



```
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ docker-compose -f docker-compose-cli.yaml up -d
Creating peer1.org2.example.com ...
Creating orderer.example.com ...
Creating peer0.org1.example.com ...
Creating peer1.org2.example.com
Creating peer0.org2.example.com ...
Creating peer0.org1.example.com
Creating orderer.example.com
Creating peer1.org1.example.com ...
Creating peer0.org2.example.com
Creating peer0.org2.example.com ... done
Creating cli ...
Creating cli ... done
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ 
```

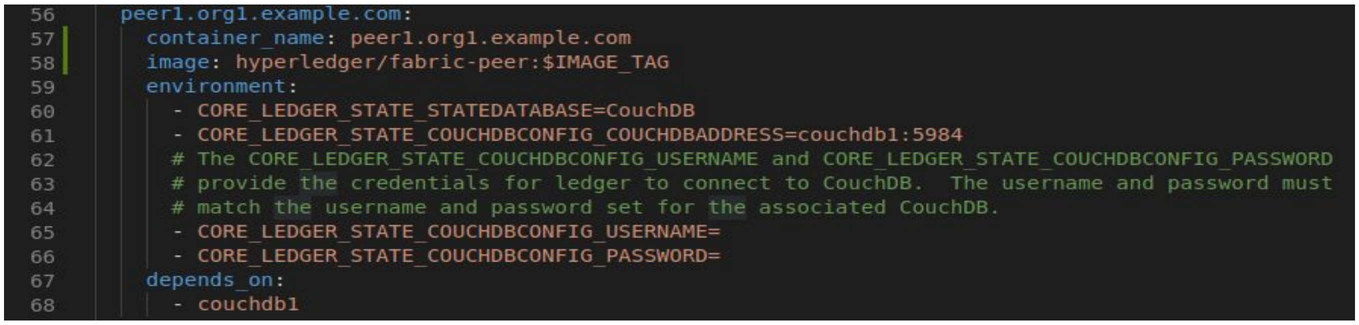
b. Now open docker-compose-couch.yaml file and set image and container namer for all four peers as shown below -

i. For peer0.org1.example.com



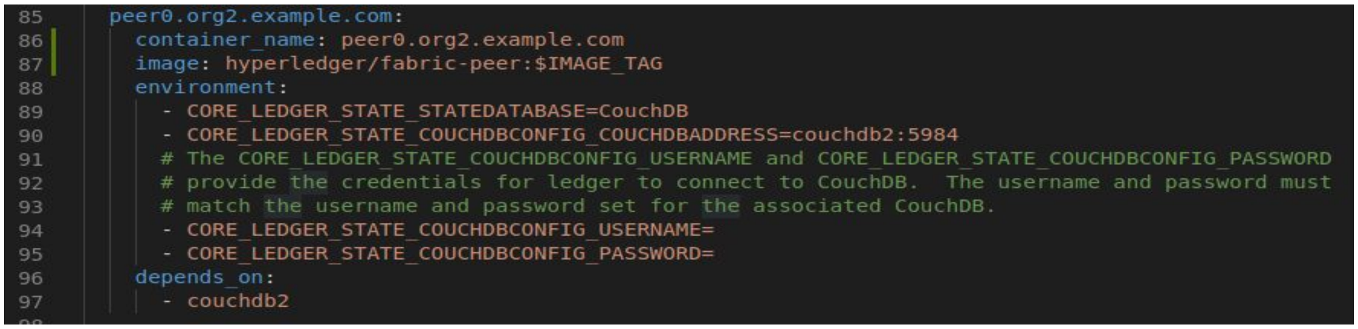
```
  peer0.org1.example.com:
    container_name: peer0.org1.example.com
    image: hyperledger/fabric-peer:$IMAGE_TAG
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb0:5984
      # The CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME and CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
      # provide the credentials for ledger to connect to CouchDB.  The username and password must
      # match the username and password set for the associated CouchDB.
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb0
```

ii. For peer1.org1.example.com



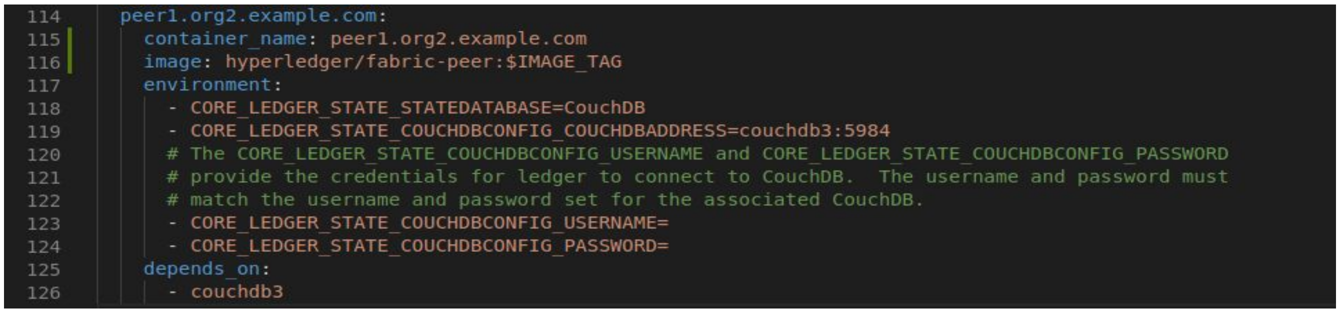
```
  peer1.org1.example.com:
    container_name: peer1.org1.example.com
    image: hyperledger/fabric-peer:$IMAGE_TAG
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb1:5984
      # The CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME and CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
      # provide the credentials for ledger to connect to CouchDB.  The username and password must
      # match the username and password set for the associated CouchDB.
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb1
```

iii. For peer0.org2.example.com



```
  peer0.org2.example.com:
    container_name: peer0.org2.example.com
    image: hyperledger/fabric-peer:$IMAGE_TAG
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb2:5984
      # The CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME and CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
      # provide the credentials for ledger to connect to CouchDB.  The username and password must
      # match the username and password set for the associated CouchDB.
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb2
```

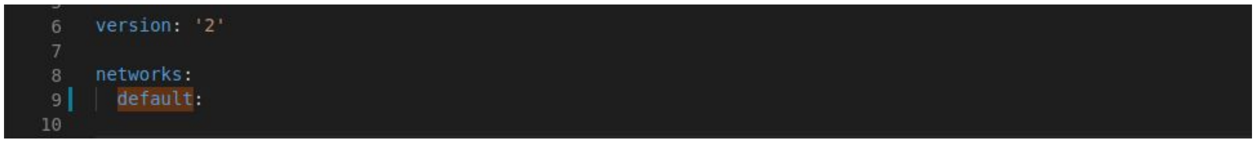
iv. For peer1.org2.example.com



```
  peer1.org2.example.com:
    container_name: peer1.org2.example.com
    image: hyperledger/fabric-peer:$IMAGE_TAG
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb3:5984
      # The CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME and CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
      # provide the credentials for ledger to connect to CouchDB.  The username and password must
      # match the username and password set for the associated CouchDB.
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb3
```

c. Also change network to **default** in docker-compose-cli.yaml at all places.

d. Also change network of couchdb to **default** on which all peers are running.

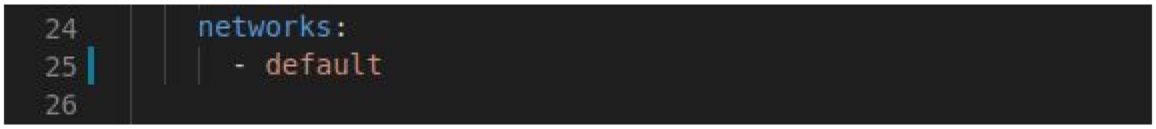


```
version: '2'

networks:
  default:

```

e. Also change subtree network name to default on all four places.



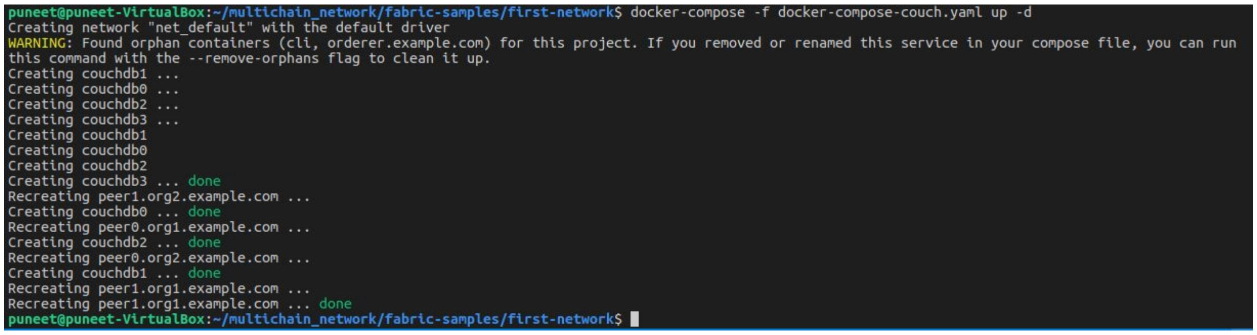
```
    networks:
      - default

```

f. Then run this command -

docker-compose -f docker-compose-couch.yaml up -d

g. And output will be like this -

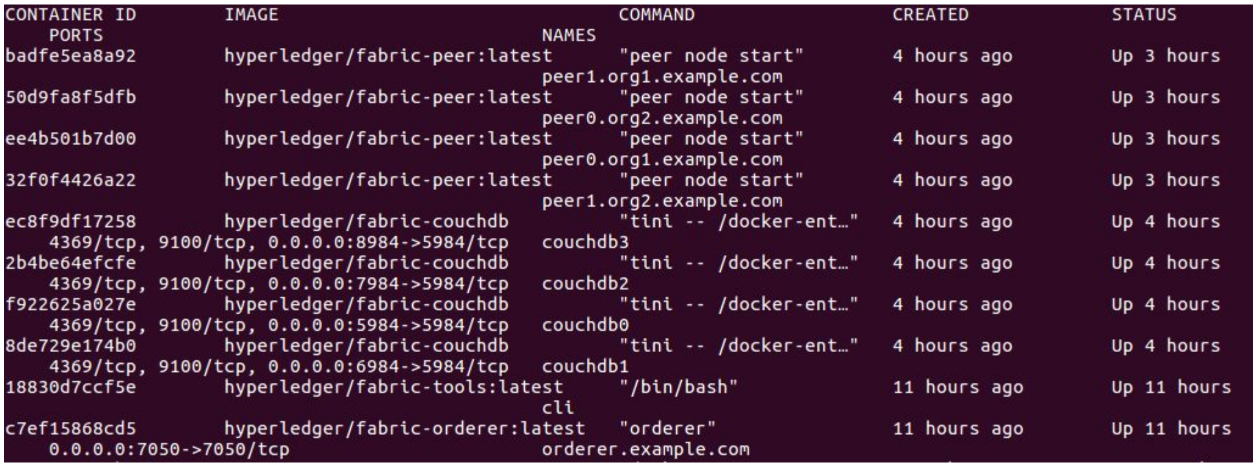


```
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ docker-compose -f docker-compose-couch.yaml up -d
Creating network "net_default" with the default driver
WARNING: Found orphan containers (cli, orderer.example.com) for this project. If you removed or renamed this service in your compose file, you can run
this command with the --remove-orphans flag to clean it up.
Creating couchdb1 ...
Creating couchdb0 ...
Creating couchdb2 ...
Creating couchdb3 ...
Creating couchdb1
Creating couchdb0
Creating couchdb2
Creating couchdb3 ... done
Recreating peer1.org2.example.com ...
Creating couchdb0 ... done
Recreating peer0.org1.example.com ...
Creating couchdb2 ... done
Recreating peer0.org2.example.com ...
Creating couchdb1 ... done
Recreating peer1.org1.example.com ...
Recreating peer1.org1.example.com ... done
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ 
```

h. After completing all these steps , you can run command -

   docker ps -a

   This will list all running containers regarding our network setup, in our case it will list 10 containers.



```
CONTAINER ID        IMAGE                               COMMAND                  CREATED             STATUS
    PORTS                                      NAMES
badfe5ea8a92        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                               peer1.org1.example.com
50d9fa8f5dfb        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                               peer0.org2.example.com
ee4b501b7d00        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                               peer0.org1.example.com
32f0f4426a22        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                               peer1.org2.example.com
ec8f9df17258        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:8984->5984/tcp   couchdb3
2b4be64efcfe        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp   couchdb2
f922625a027e        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp   couchdb0
8de729e174b0        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp   couchdb1
18830d7ccf5e        hyperledger/fabric-tools:latest     "/bin/bash"              11 hours ago        Up 11 hours
                                               cli
c7ef15868cd5        hyperledger/fabric-orderer:latest   "orderer"                11 hours ago        Up 11 hours
    0.0.0.0:7050->7050/tcp                     orderer.example.com
```

i. Proposed network setup is complete, our network have -
    i. One orderer
    ii. Two organizations
    iii. Four peers (two peers in each organization)
    iv. Couchdb for all peers