Project Working

STEP1: Create directories and copy essential files

- Create a folder named “Automobile”
- Create subfolders in **Automobile** folder with names
    - “client",
    - “server"
    - “ca-server-yaml”
    - “Ca-client-yaml”

- Copy **“fabric-ca-server-config.yaml”** file from the folder and paste in “**ca-server-yaml”** folder
- Copy **“fabric-ca-client-config.yaml”** file from the folder and paste in **“ca-client-yaml”** folder.

Step 2 :Creating root server

- Copy **“fabric-ca-server-config.yaml”** file from the folder nd paste in **“server”** folder
- Open terminal and run the following commands
    - Open a secure shell connection to virtual machine using:

```
vagrant ssh
```

    - Create and configure guest machines according to your Vagrantfile using:

```
vagrant up
```

- Move to **Automobile** directory using:
  cd [folder name] to enter into a folder
  cd .. to come out of a directory
- Set environment using following command:

```
export FABRIC_CA_SERVER_HOME=$PWD/server
```

- Start server using command:

```
fabric-ca-server start
```

- Create a subfolder with name caserver in client folder.
- Create a subfolder admin in caserver folder
- Open another terminal
  Note:
    - Make sure vagrant is up
    - Move to automobile directory

- Set environment variable using command:

```
export FABRIC_CA_CLIENT_HOME=$PWD/client/caserver/admin
```

- Enroll bootstrap identity using command:

```
fabric-ca-client enroll -u http://admin:pw@localhost:7054
```

At this point server has been started and bootstrap identity has been registered

Step 3:Creating Organizational identities

- Create subfolder in **client** folder with names:
  - **manufacture**
  - **excise**
  - **fbr**
  - **Orderer**

- Make **admin** subfolder in the following folders
  - **manufacture**
  - **excise**
  - **fbr**
  - **Orderer**
- Copy **fabric-ca-client-config.yaml** file from the following location:
  - ***file/ca-client-yaml/[organization_name]*** to ***Automobile/client/[organization_name]/admin*** *folder.*

- Do this for all 4 organizations.
- Set environment variables using command:

```
export FABRIC_CA_CLIENT_HOME=$PWD/client/caserver/admin
```

- Register identities using following command:

```
      # SET ATTRIBUTE
ATTRIBUTES='"hf.Registrar.Roles=peer,user,client","hf.AffiliationMgr=true","hf.Revoker=true"'
      # Register EXCISE ADMIN
fabric-ca-client register --id.type client --id.name Excise-admin --id.secret pw --id.affiliation excise --id.attrs $ATTRIBUTES

      # SET ATTRIBUTE
ATTRIBUTES='"hf.Registrar.Roles=peer,user,client","hf.AffiliationMgr=true","hf.Revoker=true"'
      # Register fbr-ADMIN
```

```
fabric-ca-client register --id.type client --id.name Fbr-admin
--id.secret pw --id.affiliation fbr --id.attrs $ATTRIBUTES

      # SET ATTRIBUTES
ATTRIBUTES='"hf.Registrar.Roles=orderer"'
      # Register ORDERER ADMIN
fabric-ca-client register --id.type client --id.name Orderer-admin
--id.secret pw --id.affiliation excise --id.attrs $ATTRIBUTES

      # SET ATTRIBUTES
ATTRIBUTES='"hf.Registrar.Roles=peer,user,client","hf.AffiliationMgr
=true","hf.Revoker=true"'
      # Register MANUFACTURER ADMIN
fabric-ca-client register --id.type client --id.name
Manufacturer-admin --id.secret pw --id.affiliation manufacturer
--id.attrs $ATTRIBUTES
```

- At this point we have registered all organizations admin's

- Set environment variable

```
export FABRIC_CA_CLIENT_HOME=$PWD/client/excise/admin
```

- Enroll excise admin

```
fabric-ca-client enroll -u http://Excise-admin:pw@localhost:7054
```

- Set environment variable

```
export FABRIC_CA_CLIENT_HOME=$PWD/client/fbr/admin
```

- Enroll fbr admin

```
fabric-ca-client enroll -u http://Fbr-admin:pw@localhost:7054
```

- Set environment variable

```
export FABRIC_CA_CLIENT_HOME=$PWD/client/manufacturer/admin
```

- Enroll manufacturer admin

```
fabric-ca-client enroll -u
http://Manufacturer-admin:pw@localhost:7054
```

- Set environment variable

```
export FABRIC_CA_CLIENT_HOME=$PWD/client/orderer/admin
```

- Enroll orderer admin

```
  fabric-ca-client enroll -u http://Orderer-admin:pw@localhost:7054
```

Step 4:Add *admincerts* manually:

- Make subfolder **msp** within each organization **admin** folder as ***[organization_name]/admin/msp***
- Make a subfolder **admincerts** within each orgnization **msp** subfolder of **admin** ***[organization_name]/admin/msp/admincerts***
- Copy ***caserver/admin/signcerts/cert.pem*** file
- Paste the copied file in **admincerts** folder of all organizations
- Eg: Paste the file in the following folders
    - *orderer/admin/msp/admincerts*
    - *fbr/admin/msp/admincerts*

Step 5: Locally setting MSP for organization

- Make subfolder **msp** within each organization folder as ***automobile/client / [organization_name]***
- Eg:
    - ***automobile/client/excise*** will have a folder ***msp***
- Make **3** subfolders in **msp** folder of each organization with names:
    - **cacerts**
    - **admincerts**
    - **keystore**
- Eg: ***automobile/client/excise/msp*** will have **cacerts, admincerts, & keystore**
- Copy **ca-cert.pem** file from ***automobile/server*** folder
- Paste the copied file in ***[organization_name]/msp/cacerts*** for all organization.
- Eg: Paste it in:
    - *excise/msp/cacerts* folder

- This confirms root CA
- Copy **cert.pem** fille available in ***[organization_name]/admin/msp/signcerts***
- Paste the copied file to ***[organization_name]/msp/admincerts***
- Eg:
    - Copy file from ***excise/admin/msp/signcerts*** and paste the file in ***excise/msp/admincerts***

- Repeat the process for all organizations ie copy & paste **cert.pem** file for respective orgnizations.

Step 6: Orderer Setup

- Orderer depend upon **genesis block, orderer msp & orderer yaml file**
- Copy **configtx.yaml** file given in file.
- Paste the copied file in **Automobile** folder

- Copy **orderer.yaml** file given in file folder
- Paste the copied file in **Automobile** folder
  COMMANDS
- Note:
  - Vagrant is up
  - All commands will run in **Automobile** directory so use ***cd [folder_name]*** to move to a directory

```
# SET ENVIRONMENT
export FABRIC_LOGGING_SPEC=INFO
export FABRIC_CFG_PATH=$PWD


# Create the Genesis Block
configtxgen -profile AutomobileOrdererGenesis -outputBlock
./automobile-genesis.block -channelID ordererchannel


```

- Genesis block has been created
- Now we have to register & enroll orderer identity and generate crypto material for orderer identity.

```
# Set environment variable
IDENTITY="admin"
 CA_CLIENT_FOLDER="client/orderer"
export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"
ADMIN_CLIENT_HOME=$FABRIC_CA_CLIENT_HOME
# Register orderer
fabric-ca-client register --id.type orderer --id.name orderer
--id.secret pw --id.affiliation excise
```

```
# Set Environment variables
IDENTITY="orderer"
CA_CLIENT_FOLDER="client/orderer"
export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"

# Enroll admin
fabric-ca-client enroll -u http://orderer:pw@localhost:7054
mkdir -p $FABRIC_CA_CLIENT_HOME/msp/admincerts

# Copy
cp $ADMIN_CLIENT_HOME/msp/signcerts/*
$FABRIC_CA_CLIENT_HOME/msp/admincerts
```

- This cmd copies the admin signcertificate to admincerts of msp of orderer
- **Note:**
  The genesis block created is in automobile folder.

Step 7:Launching the orderer

Note:

- Vagrant is up
- All commands will run in **Automobile** directory so use ***cd [folder_name]*** to move to a directory

```
# Set environment
export ORDERER_FILELEDGER_LOCATION=$pwd/Automobile/orderer/ledger

# Change this to control logs verbosity
export FABRIC_LOGGING_SPEC=INFO

export FABRIC_CFG_PATH=$PWD

# Launch orderer
orderer
```

- Executing these command will launch the orderer.

Error

- You may encounter the following error

```
        Metrics.Statsd.Prefix = ""
panic: Error creating dir if missing: error creating dir [/Automobile/orderer/ledger/index/]: mkdir /Automob
ile: permission denied
```

Solution:

The solution is you have to make manual directories with sudo permission.

```
sudo mkdir -p $pwd/Automobile/orderer/ledger/index
sudo chown -R $(whoami):$(whoami)
$pwd/Automobile/orderer/ledger/index
```

Error:

- If you get this error:

```
in opening ledger factory: failed to create ledger directory: /Automobile/orderer/ledger/chains: mkdir /Auto
mobile/orderer/ledger/chains: permission denied
```

Solution:

- Make manual directories with sudo permission for this directory also

```
sudo mkdir -p $pwd/Automobile/orderer/ledger/chains
sudo chown -R $(whoami):$(whoami)
$pwd/Automobile/orderer/ledger/chains
```

- Try again to launch orderer and it'll work. Use command:

```
orderer
```

Step 7:Application channel creation

- One of all organization creates channel transaction file using configtxgen tool. This channel transaction file must be signed from other admins as per our policy.
- Once every org admin sign this channel tx file it is then submitted to orderer by any admin using configtxgen tool.

Commands:

- Open new terminal and run the following commands
- **Note:**
  - *Vagrant is up*
  - *All commands will run in **Automobile** directory so use **cd [folder_name]** to move to a directory*
  - *Make sure your orderer is up*

```
vagrant ssh
cd Automobile
```

```
# Set environment variables
export FABRIC_LOGGING_SPEC=INFO
export FABRIC_CFG_PATH=$PWD

# Create channel
configtxgen -profile AutomobileChannel -outputCreateChannelTx
./automobile-channel.tx -channelID automobilechannel
```

- Channel is now created, and for submission, all admins need to sign that channel

Step 8:Signing the channeltx as a admin

- In order to sign the channel tx file as a admin copy **core.yaml** file from **file** named folder
- Paste the copied file in **Automobile** folder.
- **Note:**
  - *Vagrant is up*
  - *All commands will run in **Automobile** directory so use **cd [folder_name]** to move to a directory*

```
# Set environment variables
ORG_NAME=excise
CRYPTO_CONFIG_ROOT_FOLDER=`pwd`/client
export
CORE_PEER_MSPCONFIGPATH=$CRYPTO_CONFIG_ROOT_FOLDER/$ORG_NAME/admin/m
sp
MSP_ID="$(tr '[:lower:]' '[:upper:]' <<<
${ORG_NAME:0:1})${ORG_NAME:1}"
export CORE_PEER_LOCALMSPID=$MSP_ID"MSP”

# Sign the channel as excise admin
CHANNEL_TX_FILE=$PWD/automobile-channel.tx
peer channel signconfigtx -f $CHANNEL_TX_FILE
```

- Excise admin has signed the channel file

- Signing the channel file for either ***fbr*** or ***manufacturer*** because as per our policy ***two out of 3 admins need to sign the channel before submission.***
- Let's do it for fbr.

```
# Set environment variables
ORG_NAME=fbr
CRYPTO_CONFIG_ROOT_FOLDER=`pwd`/client
export
CORE_PEER_MSPCONFIGPATH=$CRYPTO_CONFIG_ROOT_FOLDER/$ORG_NAME/admin/m
sp
MSP_ID="$(tr '[:lower:]' '[:upper:]' <<<
${ORG_NAME:0:1})${ORG_NAME:1}"
export CORE_PEER_LOCALMSPID=$MSP_ID"MSP”

# Sign the channel as excise admin
CHANNEL_TX_FILE=$PWD/automobile-channel.tx
peer channel signconfigtx -f $CHANNEL_TX_FILE
```

- In order to sign as manufacturer just change ***ORG_NAME*** to ***manufacturer***

STEP 9:Submitting the channel tx to orderer

- **Note:**
  - *Vagrant is up*
  - *All commands will run in* ***Automobile*** *directory so use* ***cd [folder_name]*** *to move to a directory*
  - *Make sure your orderer is up (Step 7)*
- Open another terminal and execute the following commands to submit the channel transaction to orderer

```
# Set environment variables
ORDERER_ADDRESS="localhost:7050"
CHANNEL_TX_FILE="$PWD/automobile-channel.tx”
CRYPTO_CONFIG_ROOT_FOLDER=`pwd`/client
   IDENTITY="admin"
ORG_NAME=excise
```

```
export CORE_PEER_MSPCONFIGPATH=$CRYPTO_CONFIG_ROOT_FOLDER/$ORG_NAME/$IDENTITY/msp

# Setup the MSP ID
MSP_ID="$(tr '[:lower:]' '[:upper:]' <<< ${ORG_NAME:0:1})${ORG_NAME:1}"
export CORE_PEER_LOCALMSPID=$MSP_ID"MSP"

# Submission
peer channel create -o $ORDERER_ADDRESS -c automobilechannel -f $CHANNEL_TX_FILE
```

- As a result of execution of above command you would receive block 0.

STEP 9:Peer Setup

- Organization admin will create peer identities to run needed peer msp and **core.yaml** file.
- In order to let the peer join the channel it is first launched then admin of the organization use ***join channel command*** to join the application channel.
- Pre-requisites for peer setup are:
    - *Vagrant is up*
    - *All commands will run in **Automobile** directory so use **cd [folder_name]** to move to a directory*
    - *CA Server is up (Step 2)*
    - *Orderer is up (Step 7)*
    - *Channel is created successfully*
- Make a subfolder **peer1** in **excise** folder
- Make a subfolder **peer1** in **fbr** folder
- Make a subfolder **peer1** in **manufacturer** folder
- Copy the **fabric-ca-client-config.yaml** file in ***file/peer/[organization_name]***
- Paste the copied file to ***client/[organization_name]/peer1***.
- Repeat the process for all organizations **(excise, fbr, manufacturer)**

Commands:

- *Vagrant is up*
- *All commands will run in **Automobile** directory so use **cd [folder_name]** to move to a directory*
- *CA Server is up (Step 2)*
- *Orderer is up (Step 7)*

- *Channel is created successfully*

```
# Set environment variables
IDENTITY="admin"
ORG_NAME=excise
CA_CLIENT_FOLDER="client/$ORG_NAME"
export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"
ADMIN_CLIENT_HOME=$FABRIC_CA_CLIENT_HOME
PEER=peer1

# Register peer
fabric-ca-client register --id.type peer --id.name $PEER --id.secret
pw --id.affiliation $ORG_NAME

# Set environment variables
IDENTITY=peer1
 CA_CLIENT_FOLDER="client/$ORG_NAME"
 export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"

# Enroll peer
fabric-ca-client enroll -u http://$IDENTITY:pw@localhost:7054
mkdir -p $FABRIC_CA_CLIENT_HOME/msp/admincerts
CA_CLIENT_FOLDER="client/$ORG_NAME"
```

- Copy **cert.pem** file in **excise/msp/admincerts**
- Paste the copied file to **excise/peer1/msp/admincerts**

- Repeat the above process for other organizations ***manufacturer & fbr***
- For **fbr:**

```
# Set environment variables
IDENTITY="admin"
ORG_NAME=fbr
CA_CLIENT_FOLDER="client/$ORG_NAME"
export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"
ADMIN_CLIENT_HOME=$FABRIC_CA_CLIENT_HOME
PEER=peer1
```

```
# Register peer
fabric-ca-client register --id.type peer --id.name $PEER --id.secret
pw --id.affiliation $ORG_NAME

# Set environment variables
IDENTITY=peer1
 CA_CLIENT_FOLDER="client/$ORG_NAME"
 export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"

# Enroll peer
fabric-ca-client enroll -u http://$IDENTITY:pw@localhost:7054
mkdir -p $FABRIC_CA_CLIENT_HOME/msp/admincerts
CA_CLIENT_FOLDER="client/$ORG_NAME"
```

- Copy **cert.pem** file in **fbr/msp/admincerts**
- Paste the copied file to **fbr/peer1/msp/admincerts**

- For **manufacturer:**

```
# Set environment variables
IDENTITY="admin"
ORG_NAME=manufacturer
CA_CLIENT_FOLDER="client/$ORG_NAME"
export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"
ADMIN_CLIENT_HOME=$FABRIC_CA_CLIENT_HOME
PEER=peer1

# Register peer
fabric-ca-client register --id.type peer --id.name $PEER --id.secret
pw --id.affiliation $ORG_NAME

# Set environment variables
IDENTITY=peer1
 CA_CLIENT_FOLDER="client/$ORG_NAME"
 export FABRIC_CA_CLIENT_HOME="$CA_CLIENT_FOLDER/$IDENTITY"
```

```
# Enroll peer
fabric-ca-client enroll -u http://$IDENTITY:pw@localhost:7054
mkdir -p $FABRIC_CA_CLIENT_HOME/msp/admincerts
CA_CLIENT_FOLDER="client/$ORG_NAME"
```

- Copy **cert.pem** file in **manufacturer/msp/admincerts**
- Paste the copied file to **manufacturer/peer1/msp/admincerts**

Step 10:Launching the peer and join the channel

- Copy **core.yaml** file from **file/peer/[organization_name]**
- Paste the copied file to respective organization folder.
- Eg:
    - Copy **core.yaml** file from **file/peer/excise**
    - Paste the file in **Automobile/client/excise** folder
    - Repeat the above process for all organizations.

Running Excise peer

- Let say we are trying to run ***excise-peer***

```
# Set environment variables
export FABRIC_LOGGING_SPEC=INFO
export CORE_PEER_ID=excise-peer

#peer1 for all organizations
IDENTITY=peer1
export CORE_PEER_MSPCONFIGPATH=$PWD/client/excise/peer1/msp
export FABRIC_CFG_PATH="$PWD/client/excise"
export CORE_PEER_LOCALMSPID="ExciseMSP"

#ManufacturerMSP ,FbrMSP for other organizations
export GOPATH="$PWD/client/excise/gopath"
export NODECHAINCODE="$PWD/client/excise/nodechaincode"
VAR=$((PORT_NUMBER_BASE))
export CORE_PEER_FILESYSTEMPATH="$PWD/client/excise"
export PEER_LOGS=$PWD/excise/excise-peer
sudo -E mkdir -p $CORE_PEER_FILESYSTEMPATH
```

```
# Create the folder for the logs
mkdir -p $PEER_LOGS

# Start the peer
sudo -E peer node start 2> $PEER_LOGS/peer.log
```

- Now open **excise/excise-peer/peer1.log** file and see if there are any errors
- Any errors regarding this peer will be displayed here.
- Try same for fbr and manufacturer with differenct variables names will be changed.

Trying fbr peer

- Trying to run ***fbr-peer***

```
# Set environment variables
export FABRIC_LOGGING_SPEC=INFO
export CORE_PEER_ID=fbr-peer

#peer1 for all organizations
IDENTITY=peer1
export CORE_PEER_MSPCONFIGPATH=$PWD/client/fbr/peer1/msp
export FABRIC_CFG_PATH="$PWD/client/fbr"
export CORE_PEER_LOCALMSPID="FbrMSP"

#ManufacturerMSP ,FbrMSP for other organizations
export GOPATH="$PWD/client/fbr/gopath"
export NODECHAINCODE="$PWD/client/fbr/nodechaincode"
VAR=$((PORT_NUMBER_BASE))
export CORE_PEER_FILESYSTEMPATH="$PWD/client/fbr"
export PEER_LOGS=$PWD/fbr/fbr-peer
sudo -E mkdir -p $CORE_PEER_FILESYSTEMPATH

# Create the folder for the logs
mkdir -p $PEER_LOGS

# Start the peer
sudo -E peer node start 2> $PEER_LOGS/peer.log
```

- Now open **fbr/fbr-peer/peer1.log** file and see if there are any errors
- Any errors regarding this peer will be displayed here.

Running manufacturer peer

- Trying to run ***fbr-peer***

```
# Set environment variables
export FABRIC_LOGGING_SPEC=INFO
export CORE_PEER_ID=manufacturer-peer

#peer1 for all organizations
IDENTITY=peer1
export CORE_PEER_MSPCONFIGPATH=$PWD/client/manufacturer/peer1/msp
export FABRIC_CFG_PATH="$PWD/client/manufacturer"
export CORE_PEER_LOCALMSPID="ManufacturerMSP"

#ManufacturerMSP ,FbrMSP for other organizations
export GOPATH="$PWD/client/manufacturer/gopath"
export NODECHAINCODE="$PWD/client/manufacturer/nodechaincode"
VAR=$((PORT_NUMBER_BASE))
export CORE_PEER_FILESYSTEMPATH="$PWD/client/manufacturer"
export PEER_LOGS=$PWD/manufacturer/manufacturer-peer
sudo -E mkdir -p $CORE_PEER_FILESYSTEMPATH

# Create the folder for the logs
mkdir -p $PEER_LOGS

# Start the peer
sudo -E peer node start 2> $PEER_LOGS/peer.log
```

- Now open **manufacturer/manufacturer-peer/peer1.log** file and see if there are any errors
- Any errors regarding this peer will be displayed here.

Step 11: Set Administrator

- Login from respective organization admin and join that organization's peer to channel
- In order to set the variables for admin use following commands

```
#Set environment variables
```

```
export CORE_PEER_MSPCONFIGPATH=$PWD/client/excise/admin/msp
export FABRIC_CFG_PATH="$PWD/client/excise"
```

- Now run ***peer channel list*** command. This will list all channels that ***excise*** peer has joined.

Note

- *All commands will run in **Automobile** directory so use **cd [folder_name]** to move to a directory*
- *Orderer is up (Step 7)*
-
```
vagrant@vagrant:/vagrant/Automobile$ peer channel list
2022-12-25 10:24:23.872 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initial
ized
Channels peers has joined:
```
- You'll notice it will display nothing as no channel has joined yet by peer.
- Now let the anchor peer to join the automobile channel.
- As you have launched the peer, now you have to set the environment for admin as only admin can run the join peer command.

```
# Set environment variables
CORE_PEER_MSPCONFIGPATH=$PWD/client/excise/admin/msp
export FABRIC_CFG_PATH="$PWD/client/excise"
# Fetch channel configuration
# peer channel fetch config $AUTOMOBILE_CHANNEL_BLOCK -o
$ORDERER_ADDRESS -c airlinechannel
peer channel fetch 0 ./automobilechannel.block -o localhost:7050 -c
automobilechannel

# Join the channel
peer channel join -o localhost:7050 -b ./automobilechannel.block
```

-
```
2022-12-25 13:19:26.561 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initial
ized
2022-12-25 13:19:26.568 UTC [cli.common] readBlock -> INFO 002 Received block: 0
2022-12-25 13:19:26.649 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initial
ized
2022-12-25 13:19:26.707 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join cha
nnel
```

  If you have received block 0 it means you have successfully got the configuration block which is zero block and then joined channel

-
```
vagrant@vagrant:/vagrant/Automobile$ peer channel list
2022-12-25 13:26:28.294 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initial
ized
Channels peers has joined:
automobilechannel
```

- Now run command ***peer channel list*** it will display the channel which this organization's anchor peer has joined.

Doing same for *fbr:*

```
#Set environment variables
export CORE_PEER_MSPCONFIGPATH=$PWD/client/fbr/admin/msp
export FABRIC_CFG_PATH="$PWD/client/fbr"

# Set environment variables
CORE_PEER_MSPCONFIGPATH=$PWD/client/fbr/admin/msp
export FABRIC_CFG_PATH="$PWD/client/fbr"
# Fetch channel configuration
peer channel fetch 0 ./automobilechannel.block -o localhost:7050 -c
automobilechannel

# Join the channel
peer channel join -o localhost:7050 -b ./automobilechannel.block
```

- ***fbr*** peer has joined the channel

Doing same for manufacturerer*:*

```
#Set environment variables
export CORE_PEER_MSPCONFIGPATH=$PWD/client/manufacturerer/admin/msp
export FABRIC_CFG_PATH="$PWD/client/manufacturerer"

# Set environment variables
CORE_PEER_MSPCONFIGPATH=$PWD/client/manufacturerer/admin/msp
export FABRIC_CFG_PATH="$PWD/client/manufacturerer"
# Fetch channel configuration
peer channel fetch 0 ./automobilechannel.block -o localhost:7050 -c
automobilechannel

# Join the channel
peer channel join -o localhost:7050 -b ./automobilechannel.block
```

- ***manufaturer*** peer has joined the channel