Before starigng anything run these commands::

**./**byfn**.**sh **-**m down

docker rm -f $(docker ps -aq)

docker rmi -f $(docker images -q)

docker network prune

**For Fabric use fabric-samples as base platform, for composer use fabric-tools as platform**

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.

1. To install composer-cli run the following command:

Copy

npm install -g composer-cli

The composer-cli contains all the command line operations for developing business networks.

1. To install generator-hyperledger-composer run the following command:

Copy

npm install -g generator-hyperledger-composer

The generator-hyperledger-composer is a Yeoman plugin that creates bespoke applications for your business network.

1. To install composer-rest-server run the following command:

Copy

npm install -g composer-rest-server

The composer-rest-server uses the Hyperledger Composer LoopBack Connector to connect to a business network, extract the models and then present a page containing the REST APIs that have been generated for the model.

1. To install Yeoman run the following command:

Copy

npm install -g yo

docker kill $(docker ps -q)

docker rm $(docker ps -aq)

docker rmi $(docker images dev-\* -q)

1. In a directory of your choice (will assume ~/fabric-tools) get the zip file that contains the tools to install Hyperledger Fabric v1.0.

Copy

mkdir ~/fabric-tools && cd ~/fabric-tools

curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.zip

unzip fabric-dev-servers.zip

Note:: fabric-tools already available at github.

In fabric-tools>>fabric-scripts>>hlfv1>>composer>>docker-composer.yml add **dns\_search: .** below peer0.org1.example.com

Copy

cd ~/fabric-tools

./downloadFabric.sh

./startFabric.sh

./createPeerAdminCard.sh

Then follow this guide to create a composer project <https://hyperledger.github.io/composer/tutorials/developer-tutorial.html>

This is already available at github with name of hyperledger-tutorial

You first create a PeerAdmin with a card which is used to install business network(chaincode) on hyperledger blockchain and then u create Admin with a card which is used to talk to blockchain and used by anyone who needs to talk to blockchain

Start Rest Server thorugh Copy

composer-rest-server

and then u can call the rest api to make transactions.

## Deploying for Single Organization:: (folder named as tutorial available on github)

## Step One: Starting a Hyperledger Fabric network

In order to follow this tutorial, you must start a Hyperledger Fabric network. You can use the simple Hyperledger Fabric network provided in the development environment, or you can use your own Hyperledger Fabric network that you have built by following the Hyperledger Fabric documentation.

The tutorial will assume that you use the simple Hyperledger Fabric network provided in the development environment. If you use your own Hyperledger Fabric network, then you must map between the configuration detailed below and your own configuration.

1. Start a clean Hyperledger Fabric by running the following commands:

Copy

cd ~/fabric-tools

./stopFabric.sh

./teardownFabric.sh

./downloadFabric.sh

./startFabric.sh

1. Delete any business network cards that may exist in your wallet. It is safe to ignore any errors that state that the business network cards cannot be found:

Copy

composer card delete -n PeerAdmin@fabric-network

composer card delete -n admin@tutorial-network

## Step Two: Exploring the Hyperledger Fabric network

This step will explore the Hyperledger Fabric network that you have just started, so that you can understand how it has been configured, and what components it consists of. You will use all of the information in this section to configure Hyperledger Composer in subsequent steps.

#### Configuration files

The simple Hyperledger Fabric network provided in the development environment has been configured using the Hyperledger Fabric configuration tools cryptogen and configtxgen.

The configuration for cryptogen is stored in the file:

Copy

~/fabric-tools/fabric-scripts/hlfv1/composer/crypto-config.yaml

The configuration for configtxgen is stored in the file:

Copy

~/fabric-tools/fabric-scripts/hlfv1/composer/configtx.yaml

You can find more information about these configuration tools, what they do, and how to use them by reading the Hyperledger Fabric documentation.

#### Organizations

The simple Hyperledger Fabric network is made up of a single organization called Org1. The organization uses the domain name org1.example.com. Additionally, the Membership Services Provider (MSP) for this organization is called Org1MSP. In this tutorial, you will deploy a blockchain business network that only the organization Org1 can interact with.

#### Network components

The Hyperledger Fabric network is made up of several components:

* A single peer node for Org1, named peer0.org1.example.com.
  + The request port is 7051.
  + The event hub port is 7053.
* A single Certificate Authority (CA) for Org1, named ca.org1.example.com.
  + The CA port is 7054.
* A single orderer node, named orderer.example.com.
  + The orderer port is 7050.

The Hyperledger Fabric network components are running inside Docker containers. When running Hyperledger Composer within a Docker container, the names above (for example, peer0.org1.example.com) can be used to interact with the Hyperledger Fabric network.

This tutorial will run Hyperledger Composer commands on the Docker host machine, rather than from inside the Docker network. This means that the Hyperledger Composer commands must interact with the Hyperledger Fabric network using localhost as the host name and the exposed container ports.

#### Users

The organization Org1 is configured with a user named Admin@org1.example.com. This user is an administrator. Administrators for an organization have the permission to install the code for a blockchain business network onto their organization's peers, and can also have the permission to start the blockchain business network, depending on configuration. In this tutorial, you will deploy a blockchain business network by acting as the user Admin@org1.example.com.

The user Admin@org1.example.com has a set of certificates and private key files stored in the directory:

Copy

~/fabric-tools/fabric-scripts/hlfv1/composer/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

You will use some of these files later on to interact with the Hyperledger Fabric network.

In addition to the administrator, the CA (Certificate Authority) for Org1 has been configured with a default user. This default user has an enrollment ID of admin and an enrollment secret of adminpw. However, this user does not have permission to deploy a blockchain business network.

#### Channel

Finally, a channel named composerchannel has been created. The peer node peer0.org1.example.com has been joined to this channel. You can only deploy Hyperledger Composer blockchain business networks into existing channels, but you can create additional channels by following the Hyperledger Fabric documentation.

## Step Three: Building a connection profile

A connection profile specifies all of the information required to locate and connect to the Hyperledger Fabric network, for example the host names and ports of all of the Hyperledger Fabric network components. In this step, you will create a connection profile for Hyperledger Composer to use to connect to the Hyperledger Fabric network.

1. Create a connection profile file called connection.json.
2. Give the connection profile name and type properties by adding the following three lines to the top of connection.json:

Copy

{

"name": "fabric-network",

"type": "hlfv1",

The name property in a connection profile gives a name to the Hyperledger Fabric network, so we can reference it later on. In the connection profile you have just created, the name is fabric-network. You can use any name you like for the Hyperledger Fabric network.

Hyperledger Composer is designed to be compatible with different types blockchain networks. Currently, only Hyperledger Fabric v1.0 is supported, but you must specify the type of blockchain network to use. The type for Hyperledger Fabric v1.0 is hlfv1.

1. The name of the MSP that is used to connect to the Hyperledger Fabric network must be specified:

Copy

"mspID": "Org1MSP",

We are connecting as Org1, and the MSP for Org1 is called Org1MSP.

1. We must specify the host names and ports of all of the peer nodes in the Hyperledger Fabric network that we want to connect to.

Copy

"peers": [

{

"requestURL": "grpc://localhost:7051",

"eventURL": "grpc://localhost:7053"

}

],

Here, we have specified our single peer node peer0.org1.example.com (using the host name localhost), the request port 7051, and the event hub port 7053.

The peers array can contain multiple peer nodes. If you have multiple peer nodes, you should add them all to the peers array so that Hyperledger Composer can interact with them.

The blockchain business network will be deployed to all of the specified peer nodes. Once the blockchain business network has been deployed, the specified peer nodes will be used for querying the blockchain business network, endorsing transactions, and subscribing to events.

1. We must specify the host name and port of the certificate authority (CA) in the Hyperledger Fabric network that we want to use for enrolling existing users and registering new users.

Copy

"ca": {

"url": "http://localhost:7054",

"name": "ca.org1.example.com"

},

Here we have specified our single CA ca.or1.example.com (using the hostname localhost) and the CA port 7054.

1. We must specify the host names and ports of all of the ordering nodes in the Hyperledger Fabric that we want to connect to.

Copy

"orderers": [

{

"url" : "grpc://localhost:7050"

}

],

Here, we have specified our single orderer node orderer.example.com (using the hostname localhost) and the orderer port 7050.

The orderers array can contain multiple orderer nodes. If you have multiple orderer nodes, you should add them all to the orderers array so that Hyperledger Composer can interact with them.

1. We must specify the name of an existing channel. We will deploy our blockchain business network into the channel composerchannel.

Copy

"channel": "composerchannel",

1. Finally, we can optionally specify a timeout for endorsing transactions when interacting with a blockchain business network.

Copy

"timeout": 300

}

Here, we have specified a timeout of 300 seconds. If a transaction takes longer than 300 seconds to endorse, then a timeout error will be thrown.

1. Save your changes to connection.json. The completed connection profile should look like the following:

Copy

{

"name": "fabric-network",

"type": "hlfv1",

"mspID": "Org1MSP",

"peers": [

{

"requestURL": "grpc://localhost:7051",

"eventURL": "grpc://localhost:7053"

}

],

"ca": {

"url": "http://localhost:7054",

"name": "ca.org1.example.com"

},

"orderers": [

{

"url" : "grpc://localhost:7050"

}

],

"channel": "composerchannel",

"timeout": 300

}

## Step Four: Locating the certificate and private key for the Hyperledger Fabric administrator

In order to deploy a blockchain business network to this Hyperledger Fabric network, we must identify ourselves as an administrator with the permissions to perform this operation. In this step, you locate the files required to identify yourself as an administrator.

The administrator for our Hyperledger Fabric network is a user called Admin@org1.example.com. The certificates and private key files for this user are stored in the directory:

Copy

~/fabric-tools/fabric-scripts/hlfv1/composer/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

You must first locate the certificate file for this user. The certificate is the public part of the identity. The certificate file can be found in the signcerts subdirectory and is named Admin@org1.example.com-cert.pem. If you look at the contents of this file, then you will find a PEM encoded certificate similar to the following:

Copy

-----BEGIN CERTIFICATE-----

MIICGjCCAcCgAwIBAgIRANuOnVN+yd/BGyoX7ioEklQwCgYIKoZIzj0EAwIwczEL

MAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNhbiBG

cmFuY2lzY28xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh

Lm9yZzEuZXhhbXBsZS5jb20wHhcNMTcwNjI2MTI0OTI2WhcNMjcwNjI0MTI0OTI2

WjBbMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEWMBQGA1UEBxMN

U2FuIEZyYW5jaXNjbzEfMB0GA1UEAwwWQWRtaW5Ab3JnMS5leGFtcGxlLmNvbTBZ

MBMGByqGSM49AgEGCCqGSM49AwEHA0IABGu8KxBQ1GkxSTMVoLv7NXiYKWj5t6Dh

WRTJBHnLkWV7lRUfYaKAKFadSii5M7Z7ZpwD8NS7IsMdPR6Z4EyGgwKjTTBLMA4G

A1UdDwEB/wQEAwIHgDAMBgNVHRMBAf8EAjAAMCsGA1UdIwQkMCKAIBmrZau7BIB9

rRLkwKmqpmSecIaOOr0CF6Mi2J5H4aauMAoGCCqGSM49BAMCA0gAMEUCIQC4sKQ6

CEgqbTYe48az95W9/hnZ+7DI5eSnWUwV9vCd/gIgS5K6omNJydoFoEpaEIwM97uS

XVMHPa0iyC497vdNURA=

-----END CERTIFICATE-----

Next, you must locate the private key file for this user. The private key is used to sign transactions as this identity. The private key file can be found in the keystore subdirectory. The name of the private key file is a long hexadecimal string, with a suffix of \_sk, for example 114aab0e76bf0c78308f89efc4b8c9423e31568da0c340ca187a9b17aa9a4457\_sk. The name will change every time the configuration is generated. If you look at the contents of this file, then you will find a PEM encoded private key similar to the following:

Copy

-----BEGIN PRIVATE KEY-----

MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQg00IwLLBKoi/9ikb6

ZOAV0S1XeNGWllvlFDeczRKQn2uhRANCAARrvCsQUNRpMUkzFaC7+zV4mClo+beg

4VkUyQR5y5Fle5UVH2GigChWnUoouTO2e2acA/DUuyLDHT0emeBMhoMC

-----END PRIVATE KEY-----

Remember the path to both of these files, or copy them into the same directory as the connection profile file connection.json that you created in the previous step. You will need these files in the next step.

## Step Five: Creating a business network card for the Hyperledger Fabric administrator

A business network card contains all of the information required to connect to a blockchain business network and the underlying Hyperledger Fabric network. This information includes the connection profile created in step three, and the certificate and private key for the administrator located in step four.

In this step you will create a business network card for the administrator to use to deploy the blockchain business network to the Hyperledger Fabric network.

Run the composer card create command to create a business network card. You must specify the path to all three files that you either created or located in the previous steps:

Copy

composer card create -p connection.json -u PeerAdmin -c Admin@org1.example.com-cert.pem -k 114aab0e76bf0c78308f89efc4b8c9423e31568da0c340ca187a9b17aa9a4457\_sk -r PeerAdmin -r ChannelAdmin

A business network card file called PeerAdmin@fabric-network.card will have been written to the current directory. Let's explore the options that we passed to the composer card create command.

Copy

-p connection.json

This is the path to the connection profile file that we created in step three.

Copy

-u PeerAdmin

This is a name that we use to refer to the administrator user. Instead of using Admin@org1.example.com everywhere, which is quite lengthy to type, we have given a name of PeerAdmin so we can easily refer to this user.

Copy

-c Admin@org1.example.com-cert.pem

This is the path to the certificate file for the user Admin@org1.example.com that we located in step four.

Copy

-k 114aab0e76bf0c78308f89efc4b8c9423e31568da0c340ca187a9b17aa9a4457\_sk

This is the path to the private key file for the user Admin@org1.example.com that we located in step four.

Copy

-r PeerAdmin -r ChannelAdmin

Here, we specify which roles the user has. This information is required so that Hyperledger Composer knows which users are able to perform which operations. The user Admin@org1.example.com is an administrator for the Hyperledger Fabric network, and has the roles PeerAdmin (ability to install chaincode) and ChannelAdmin (ability to instantiate chaincode).

## Step Six: Importing the business network card for the Hyperledger Fabric administrator

Hyperledger Composer can only use business network cards that are placed into a wallet. The wallet is a directory on the file system that contains business network cards. In this step, you will import the business network card created in step five into the wallet so that you can use the business network card in subsequent steps.

Run the composer card import command to import the business network card into the wallet:

Copy

composer card import -f PeerAdmin@fabric-network.card

Let's explore the options that we passed to the composer card import command.

Copy

-f PeerAdmin@fabric-network.card

This is the path to the business network card file that we created in step five.

You can now use this business network card by specifying the name PeerAdmin@fabric-network. You are now all set to deploy the blockchain business network to the Hyperledger Fabric network.

We are going to deploy the blockchain business network tutorial-network that is created by following the [Developer Tutorial](https://hyperledger.github.io/composer/tutorials/developer-tutorial.html).

## Step Seven: Installing the Hyperledger Composer runtime onto the Hyperledger Fabric peer nodes

Hyperledger Composer includes a component called the Hyperledger Composer runtime that provides all of the functionality to host and support a business network archive, for example data validation, error handling, transaction processor function execution, and access control. In Hyperledger Fabric terms, the Hyperledger Composer runtime is a standard chaincode.

In this step, you will install the Hyperledger Composer runtime onto all of the Hyperledger Fabric peer nodes. In Hyperledger Fabric terms, this is a chaincode install operation.

Run the composer runtime install command to install the Hyperledger Composer runtime onto all of the Hyperledger Fabric peer nodes that you specified in the connection profile file you created in step three:

Copy

composer runtime install -c PeerAdmin@fabric-network -n tutorial-network

Let's explore the options that we passed to the composer runtime install command.

Copy

-c PeerAdmin@fabric-network

This is the name of the business network card that we imported into the wallet in step six.

Copy

-n tutorial-network

You must install a copy of the Hyperledger Composer runtime for each blockchain business network, and specify the name of the blockchain business network. Here we specify the name of the blockchain business network that we are deploying, tutorial-network.

## Step Eight: Starting the blockchain business network

In this step, you will start the blockchain business network. In Hyperledger Fabric terms, this is a chaincode instantiate operation.

Run the composer network start command to start the blockchain business network:

Copy

composer network start -c PeerAdmin@fabric-network -a tutorial-network.bna -A admin -S adminpw

Let's explore the options that we passed to the composer network start command.

Copy

-c PeerAdmin@fabric-network

This is the name of the business network card that we imported into the wallet in step six.

Copy

-a tutorial-network.bna

This is the path to the business network archive that contains the business network definition for our blockchain business network called tutorial-network.

Copy

-A admin

When a blockchain business network is deployed, you must create at least one participant who will be a blockchain business network administrator. This participant is responsible for onboarding other participants into the blockchain business network. Here, we are specifying that we want to create a single blockchain business network administrator called admin.

Copy

-S adminpw

This specifies that our blockchain business network administrator admin will use an enrollment secret of adminpw to request a certificate and private key from the CA (Certificate Authority). When you specify this option, the name specified for the business network administrator must be an existing enrollment ID for a user that is already registered with the CA.

Now that our blockchain business network has been started, we can interact with it using the business network card file admin@tutorial-network.card that was created.

## Step Nine: Importing the business network card for the business network administrator

Run the composer card import command to import the business network card into the wallet:

Copy

composer card import -f admin@tutorial-network.card

You can now use this business network card by specifying the name admin@tutorial-network. You are now all set to interact with the running blockchain business network!

## Step Ten: Testing the connection to the blockchain business network

Run the composer network ping command to test the connection to the blockchain business network:

Copy

composer network ping -c admin@tutorial-network

-server

# Deploying a Hyperledger Composer blockchain business network to Hyperledger Fabric (multiple organizations) folders named as multipleorganizationdeployment and fabric-samples.

Note:: in fabric-samples>firstnetwork>base>docker-compose-base.yaml add **dns\_search: .** under everything.

This tutorial will demonstrate the steps that administrators in multiple organization scenarios must take to deploy a blockchain business network to an instance of Hyperledger Fabric, including how to generate the Hyperledger Composer configuration.

It is recommended that you first follow the previous tutorial that demonstrates how to deploy a blockchain business network to an instance of Hyperledger Fabric for a single organization, as it will explain some of the concepts in more detail.

This tutorial will cover how to deploy a blockchain business network to a Hyperledger Fabric network that spans two organizations, Org1 and Org2. The tutorial is presented with different types of steps depending on which organization should follow the step.

The first kind of step is for both organizations to follow:

## Example Step: A step for Org1 and Org2 to follow

The organization Org1 is represented by Alice, the Green Conga Block:

## Example Step: A step for Org1 to follow

The organization Org2 is represented by Bob, the Violet Conga Block:

## Example Step: A step for Org2 to follow

You can follow these steps by yourself, or pair with a friend or colleague and follow the steps together.

Let's get started!

## Prerequisites

If you have installed the development environment, you will need to first stop the Hyperledger Fabric provided by the development environment:

Copy

cd ~/fabric-tools

./stopFabric.sh

./teardownFabric.sh

Clone the following GitHub repository:

Copy

git clone -b issue-6978 https://github.com/sstone1/fabric-samples.git

Follow the [Building Your First Network tutorial](http://hyperledger-fabric.readthedocs.io/en/release/build_network.html), ensuring that you use the GitHub repository cloned in the previous step. You must not clone and use the Hyperledger Fabric version of the GitHub repository as it is currently missing changes that are required for this tutorial.

## Step One: Starting a Hyperledger Fabric network

In order to follow this tutorial, you must start a Hyperledger Fabric network.

This tutorial will assume that you use the Hyperledger Fabric network provided in the Hyperledger Fabric [Building Your First Network tutorial](http://hyperledger-fabric.readthedocs.io/en/release/build_network.html). We will refer to this Hyperledger Fabric network as the BYFN (Building Your First Network) network.

You can now start the BYFN network. You must specify additional flags that are not specified in the Building Your First Network tutorial. This is because we want to use CouchDB as the world state database, and we want to start a Certificate Authority (CA) for each organization.

Copy

./byfn.sh -m generate

./byfn.sh -m up -s couchdb -a

If the command works successfully, the BYFN network is started, and you will see the following output:

Copy

========= All GOOD, BYFN execution completed ===========

\_\_\_\_\_ \_ \_ \_\_\_\_

| \_\_\_\_| | \ | | | \_ \

| \_| | \| | | | | |

| |\_\_\_ | |\ | | |\_| |

|\_\_\_\_\_| |\_| \\_| |\_\_\_\_/

Next, delete any business network cards that may exist in your wallet. It is safe to ignore any errors that state that the business network cards cannot be found:

Copy

composer card delete -n PeerAdmin@byfn-network-org1-only

composer card delete -n PeerAdmin@byfn-network-org1

composer card delete -n PeerAdmin@byfn-network-org2-only

composer card delete -n PeerAdmin@byfn-network-org2

composer card delete -n alice@tutorial-network

composer card delete -n bob@tutorial-network

composer card delete -n admin@tutorial-network

composer card delete -n PeerAdmin@fabric-network

## Step Two: Exploring the Hyperledger Fabric network

This step will explore the BFYN network configuration and components. The configuration details are required to complete the subsequent steps.

#### Organizations

The BYFN network is made up of two organizations: Org1 and Org2. The organization Org1 uses the domain name org1.example.com. The Membership Services Provider (MSP) for Org1 is called Org1MSP. The organization Org2 uses the domain name org2.example.com. The MSP for Org2 is called Org2MSP. In this tutorial, you will deploy a blockchain business network that both of the organizations Org1 and Org2 can interact with.

#### Network components

The Hyperledger Fabric network is made up of several components:

* Two peer nodes for Org1, named peer0.org1.example.com and peer1.org1.example.com.
  + The request port for peer0 is 7051.
  + The event hub port for peer0 is 7053.
  + The request port for peer1 is 8051.
  + The event hub port for peer1 is 8053.
* A single CA (Certificate Authority) for Org1, named ca.org1.example.com.
  + The CA port is 7054.
* Two peer nodes for Org2, named peer0.org2.example.com and peer1.org2.example.com.
  + The request port for peer0 is 9051.
  + The event hub port for peer0 is 9053.
  + The request port for peer1 is 10051.
  + The event hub port for peer1 is 10053.
* A single CA (Certificate Authority) for Org2, named ca.org2.example.com.
  + The CA port is 8054.
* A single orderer node, named orderer.example.com.
  + The orderer port is 7050.

These components are running inside Docker containers. When running Hyperledger Composer within a Docker container, the names above (for example, peer0.org1.example.com) can be used to interact with the Hyperledger Fabric network.

This tutorial will run Hyperledger Composer commands on the Docker host machine, rather than from inside the Docker network. This means that the Hyperledger Composer commands must interact with the Hyperledger Fabric network using localhost as the host name and the exposed container ports.

All of the network components are secured using TLS to encrypt communications. You will need the Certificate Authority (CA) certificates for all of the network components in order to connect to those network components. The CA certificates can be found in the directory containing the byfn.sh script.

CA certificate for the orderer node:

Copy

crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/ca.crt

CA certificate for Org1:

Copy

crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

CA certificate for Org2:

Copy

crypto-config/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

You will use these files later on to interact with the Hyperledger Fabric network.

#### Users

The organization Org1 is configured with a user named Admin@org1.example.com. This user is an administrator.

The user Admin@org1.example.com has a set of certificates and private key files stored in the directory:

Copy

crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

The organization Org2 is configured with a user named Admin@org2.example.com. This user is an administrator.

The user Admin@org2.example.com has a set of certificates and private key files stored in the directory:

Copy

crypto-config/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

You will use some of these files later on to interact with the Hyperledger Fabric network.

In addition to the administrator, the CAs (Certificate Authorities) for Org1 and Org2 have been configured with a default user. This default user has an enrolment ID of admin and an enrolment secret of adminpw. However, this user does not have permission to deploy a blockchain business network.

#### Channel

A channel named mychannel has been created. All four peer nodes - peer0.org1.example.com, peer1.org1.example.com, peer0.org2.example.com, and peer1.org2.example.com have been joined to this channel.

## Step Three: Building connection profiles for Org1

Org1 requires two connection profiles. One connection profile will contain just the peer nodes that belong to Org1, and the other connection profile will contain the peer nodes that belong to Org1 and Org2.

Create a connection profile file called connection-org1-only.json with the following contents and save it to disk. This connection profile will contain just the peer nodes that belong to Org1. You will use this file in later steps, so remember where you place it!

Copy

{

"name": "byfn-network-org1-only",

"type": "hlfv1",

"mspID": "Org1MSP",

"peers": [

{

"requestURL": "grpcs://localhost:7051",

"eventURL": "grpcs://localhost:7053",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer0.org1.example.com"

},

{

"requestURL": "grpcs://localhost:8051",

"eventURL": "grpcs://localhost:8053",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer1.org1.example.com"

}

],

"ca": {

"url": "https://localhost:7054",

"name": "ca-org1",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "ca.org1.example.com"

},

"orderers": [

{

"url" : "grpcs://localhost:7050",

"cert": "INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "orderer.example.com"

}

],

"channel": "mychannel",

"timeout": 300

}

Replace all instances of the text INSERT\_ORG1\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the peer nodes for Org1:

Copy

crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

Replace all instances of the text INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the orderer node:

Copy

crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/ca.crt

Create another connection profile file called connection-org1.json with the following contents and save it to disk. This connection profile will contain the peer nodes that belong to Org1 and Org2. You will use this file in later steps, so remember where you place it!

Copy

{

"name": "byfn-network-org1",

"type": "hlfv1",

"mspID": "Org1MSP",

"peers": [

{

"requestURL": "grpcs://localhost:7051",

"eventURL": "grpcs://localhost:7053",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer0.org1.example.com"

},

{

"requestURL": "grpcs://localhost:8051",

"eventURL": "grpcs://localhost:8053",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer1.org1.example.com"

},

{

"requestURL": "grpcs://localhost:9051",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer0.org2.example.com"

},

{

"requestURL": "grpcs://localhost:10051",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer1.org2.example.com"

}

],

"ca": {

"url": "https://localhost:7054",

"name": "ca-org1",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "ca.org1.example.com"

},

"orderers": [

{

"url" : "grpcs://localhost:7050",

"cert": "INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "orderer.example.com"

}

],

"channel": "mychannel",

"timeout": 300

}

Replace all instances of the text INSERT\_ORG1\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the peer nodes for Org1:

Copy

crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

Replace all instances of the text INSERT\_ORG2\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the peer nodes for Org2:

Copy

crypto-config/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

Replace all instances of the text INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the orderer node:

Copy

crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/ca.crt

Note that where this connection profile contains details of the peer nodes for Org2, it only includes the request port and does not contain the event hub port. This is because one organization cannot access another organizations event hub port.

## Step Four: Building connection profiles for Org2

Org2 requires two connection profiles. One connection profile will contain just the peer nodes that belong to Org2, and the other connection profile will contain the peer nodes that belong to Org2 and Org1.

Create a connection profile file called connection-org2-only.json with the following contents and save it to disk. This connection profile will contain just the peer nodes that belong to Org2. You will use this file in later steps, so remember where you place it!

Copy

{

"name": "byfn-network-org2-only",

"type": "hlfv1",

"mspID": "Org2MSP",

"peers": [

{

"requestURL": "grpcs://localhost:9051",

"eventURL": "grpcs://localhost:9053",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer0.org2.example.com"

},

{

"requestURL": "grpcs://localhost:10051",

"eventURL": "grpcs://localhost:10053",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer1.org2.example.com"

}

],

"ca": {

"url": "https://localhost:8054",

"name": "ca-org2",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "ca.org2.example.com"

},

"orderers": [

{

"url" : "grpcs://localhost:7050",

"cert": "INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "orderer.example.com"

}

],

"channel": "mychannel",

"timeout": 300

}

Replace all instances of the text INSERT\_ORG2\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the peer nodes for Org2:

Copy

crypto-config/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

Replace all instances of the text INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the orderer node:

Copy

crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/ca.crt

Create another connection profile file called connection-org2.json with the following contents and save it to disk. This connection profile will contain the peer nodes that belong to Org2 and Org1. You will use this file in later steps, so remember where you place it!

Copy

{

"name": "byfn-network-org2",

"type": "hlfv1",

"mspID": "Org2MSP",

"peers": [

{

"requestURL": "grpcs://localhost:9051",

"eventURL": "grpcs://localhost:9053",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer0.org2.example.com"

},

{

"requestURL": "grpcs://localhost:10051",

"eventURL": "grpcs://localhost:10053",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer1.org2.example.com"

},

{

"requestURL": "grpcs://localhost:7051",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer0.org1.example.com"

},

{

"requestURL": "grpcs://localhost:8051",

"cert": "INSERT\_ORG1\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "peer1.org1.example.com"

}

],

"ca": {

"url": "https://localhost:8054",

"name": "ca-org2",

"cert": "INSERT\_ORG2\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "ca.org2.example.com"

},

"orderers": [

{

"url" : "grpcs://localhost:7050",

"cert": "INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH",

"hostnameOverride": "orderer.example.com"

}

],

"channel": "mychannel",

"timeout": 300

}

Replace all instances of the text INSERT\_ORG2\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the peer nodes for Org2:

Copy

crypto-config/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt

Replace all instances of the text INSERT\_ORG1\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the peer nodes for Org1:

Copy

crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

Replace all instances of the text INSERT\_ORDERER\_CA\_CERT\_FILE\_PATH with the fully qualified path to the file containing the CA certificate for the orderer node:

Copy

crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/ca.crt

Note that where this connection profile contains details of the peer nodes for Org1, it only includes the request port and does not contain the event hub port. This is because one organization cannot access another organizations event hub port.

## Step Five: Locating the certificate and private key for the Hyperledger Fabric administrator for Org1

The administrator for our Hyperledger Fabric network is a user called Admin@org1.example.com. The certificates and private key files for this user are stored in the directory:

Copy

crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

You must first locate the certificate file for this user. The certificate is the public part of the identity. The certificate file can be found in the signcerts subdirectory and is named Admin@org1.example.com-cert.pem.

Next, you must locate the private key file for this user. The private key is used to sign transactions as this identity. The private key file can be found in the keystore subdirectory. The name of the private key file is a long hexadecimal string, with a suffix of \_sk, for example 78f2139bfcfc0edc7ada0801650ed785a11cfcdef3f9c36f3c8ca2ebfa00a59c\_sk. The name will change every time the configuration is generated.

Remember the path to both of these files, or copy them into the same directory as the connection profile file connection-org1.json that you created in step three. You will need these files in the next steps.

## Step Six: Locating the certificate and private key for the Hyperledger Fabric administrator for Org2

The administrator for our Hyperledger Fabric network is a user called Admin@org2.example.com. The certificates and private key files for this user are stored in the directory:

Copy

crypto-config/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp

You must first locate the certificate file for this user. The certificate is the public part of the identity. The certificate file can be found in the signcerts subdirectory and is named Admin@org2.example.com-cert.pem.

Next, you must locate the private key file for this user. The private key is used to sign transactions as this identity. The private key file can be found in the keystore subdirectory. The name of the private key file is a long hexadecimal string, with a suffix of \_sk, for example d4889cb2a32e167bf7aeced872a214673ee5976b63a94a6a4e61c135ca2f2dbb\_sk. The name will change every time the configuration is generated.

Remember the path to both of these files, or copy them into the same directory as the connection profile file connection-org2.json that you created in step four. You will need these files in the next steps.

## Step Seven: Creating business network cards for the Hyperledger Fabric administrator for Org1

In this step you will create business network cards for the administrator to use to deploy the blockchain business network to the Hyperledger Fabric network.

Run the composer card create command to create a business network card using the connection profile that just contains the peers for Org1. You must specify the path to all three files that you either created or located in the previous steps: (note: the sk file will differ.)

Copy

composer card create -p connection-org1-only.json -u PeerAdmin -c Admin@org1.example.com-cert.pem -k 78f2139bfcfc0edc7ada0801650ed785a11cfcdef3f9c36f3c8ca2ebfa00a59c\_sk -r PeerAdmin -r ChannelAdmin

If the command works successfully, a business network card file called PeerAdmin@byfn-network-org1-only.card will have been written to the current directory.

Run the composer card create command to create a business network card using the connection profile that contains the peers for Org1 and Org2. You must specify the path to all three files that you either created or located in the previous steps:

Copy

composer card create -p connection-org1.json -u PeerAdmin -c Admin@org1.example.com-cert.pem -k 78f2139bfcfc0edc7ada0801650ed785a11cfcdef3f9c36f3c8ca2ebfa00a59c\_sk -r PeerAdmin -r ChannelAdmin

If the command works successfully, a business network card file called PeerAdmin@byfn-network-org1.card will have been written to the current directory.

## Step Eight: Creating business network cards for the Hyperledger Fabric administrator for Org2

In this step you will create business network cards for the administrator to use to deploy the blockchain business network to the Hyperledger Fabric network.

Run the composer card create command to create a business network card using the connection profile that just contains the peers for Org2. You must specify the path to all three files that you either created or located in the previous steps:

Copy

composer card create -p connection-org2-only.json -u PeerAdmin -c Admin@org2.example.com-cert.pem -k d4889cb2a32e167bf7aeced872a214673ee5976b63a94a6a4e61c135ca2f2dbb\_sk -r PeerAdmin -r ChannelAdmin

If the command works successfully, a business network card file called PeerAdmin@byfn-network-org2-only.card will have been written to the current directory.

Run the composer card create command to create a business network card using the connection profile that contains the peers for Org2 and Org1. You must specify the path to all three files that you either created or located in the previous steps:

Copy

composer card create -p connection-org2.json -u PeerAdmin -c Admin@org2.example.com-cert.pem -k d4889cb2a32e167bf7aeced872a214673ee5976b63a94a6a4e61c135ca2f2dbb\_sk -r PeerAdmin -r ChannelAdmin

If the command works successfully, a business network card file called PeerAdmin@byfn-network-org2.card will have been written to the current directory.

## Step Nine: Importing the business network cards for the Hyperledger Fabric administrator for Org1

Run the composer card import command to import the business network card that just contains the peers for Org1 into the wallet:

Copy

composer card import -f PeerAdmin@byfn-network-org1-only.card

If the command works successfully, a business network card called PeerAdmin@byfn-network-org1-only will have been imported into the wallet.

Run the composer card import command to import the business network card that contains the peers for Org1 and Org2 into the wallet:

Copy

composer card import -f PeerAdmin@byfn-network-org1.card

If the command works successfully, a business network card called PeerAdmin@byfn-network-org1 will have been imported into the wallet.

## Step Ten: Importing the business network cards for the Hyperledger Fabric administrator for Org2

Run the composer card import command to import the business network card that just contains the peers for Org2 into the wallet:

Copy

composer card import -f PeerAdmin@byfn-network-org2-only.card

If the command works successfully, a business network card called PeerAdmin@byfn-network-org2-only will have been imported into the wallet.

Run the composer card import command to import the business network card that contains the peers for Org2 and Org1 into the wallet:

Copy

composer card import -f PeerAdmin@byfn-network-org2.card

If the command works successfully, a business network card called PeerAdmin@byfn-network-org2 will have been imported into the wallet.

## Step Eleven: Installing the Hyperledger Composer runtime onto the Hyperledger Fabric peer nodes for Org1

Run the composer runtime install command to install the Hyperledger Composer runtime onto all of the Hyperledger Fabric peer nodes for Org1 that you specified in the connection profile file you created in step three:

Copy

composer runtime install -c PeerAdmin@byfn-network-org1-only -n tutorial-network

## Step Twelve: Installing the Hyperledger Composer runtime onto the Hyperledger Fabric peer nodes for Org2

Run the composer runtime install command to install the Hyperledger Composer runtime onto all of the Hyperledger Fabric peer nodes for Org2 that you specified in the connection profile file you created in step four:

Copy

composer runtime install -c PeerAdmin@byfn-network-org2-only -n tutorial-network

## Step Thirteen: Defining the endorsement policy for the business network

A running business network has an endorsement policy, which defines the rules around which organizations must endorse transactions before they can be committed to the blockchain. By default, a business network is deployed with an endorsement policy that states that only one organization has to endorse a transaction before it can be committed to the blockchain.

In real world blockchain business networks, multiple organizations will want to ensure that they endorse transactions before they can be committed to the blockchain, and so the default endorsement policy is not suitable. Instead, you can specify a custom endorsement policy when you start a business network.

You can find more information on endorsement policies in the Hyperledger Fabric documentation, in [Endorsement policies](https://hyperledger-fabric.readthedocs.io/en/release/endorsement-policies.html).

Please note that the endorsement policies used for a business network must be in the JSON format used by the Hyperledger Fabric Node.js SDK. This is a different format to the simple endorsement policy format used by the Hyperledger Fabric CLI, which you will see in the Hyperledger Fabric documentation.

Create an endorsement policy file called endorsement-policy.json with the following contents and save it to disk. You will use this file in later steps, so remember where you place it!

Copy

{

"identities": [

{

"role": {

"name": "member",

"mspId": "Org1MSP"

}

},

{

"role": {

"name": "member",

"mspId": "Org2MSP"

}

}

],

"policy": {

"2-of": [

{

"signed-by": 0

},

{

"signed-by": 1

}

]

}

}

The endorsement policy you have just created states that both Org1 and Org2 must endorse transactions in the business network before they can be committed to the blockchain. If Org1 or Org2 do not endorse transactions, or disagree on the result of a transaction, then the transaction will be rejected by the business network.

## Step Fourteen: Understanding and selecting the business network administrators

When a business network is started, the business network must be configured with a set of initial participants. These participants will be responsible for bootstrapping the business network and onboarding other participants into the business network. In Hyperledger Composer, we call these initial participants the business network administrators.

In our business network, the organizations Org1 and Org2 have equal rights. Each organization will provide a business network administrator for the business network, and those business network administrators will onboard the other participants in their organizations. The business network administrator for Org1 will be Alice, and the business network administrator for Org2 will be Bob.

When the business network is started, the certificates (the public part of the identity) for all of the business network administrators must be passed to the organization performing the commands to start the business network. After the business network has been started, all of the business network administrators can use their identities to interact with the business network.

You can find more information on business network administrators in [Deploying Business Networks](https://hyperledger.github.io/composer/business-network/bnd-deploy.html).

## Step Fifteen: Retrieving business network administrator certificates for Org1

Run the composer identity request command to retrieve certificates for Alice to use as the business network administrator for Org1:

Copy

composer identity request -c PeerAdmin@byfn-network-org1-only -u admin -s adminpw -d alice

The -u admin and the -s adminpw options to this command correspond to the default user registered with the Hyperledger Fabric CA (Certificate Authority).

The certficates will be placed into a directory called alice in the current working directory. There are three certificate files created, but only two are important. These are admin-pub.pem, the certificate (including the public key), and admin-priv.pem, the private key. Only the admin-pub.pem file is suitable for sharing with other organizations. The admin-priv.pem file must be kept secret as it can be used to sign transactions on behalf of the issuing organization.

## Step Sixteen: Retrieving business network administrator certificates for Org2

Run the composer identity request command to retrieve certificates for Bob to use as the business network administrator for Org2:

Copy

composer identity request -c PeerAdmin@byfn-network-org2-only -u admin -s adminpw -d bob

The -u admin and the -s adminpw options to this command correspond to the default user registered with the Hyperledger Fabric CA (Certificate Authority).

The certficates will be placed into a directory called bob in the current working directory. There are three certificate files created, but only two are important. These are admin-pub.pem, the certificate (including the public key), and admin-priv.pem, the private key. Only the admin-pub.pem file is suitable for sharing with other organizations. The admin-priv.pem file must be kept secret as it can be used to sign transactions on behalf of the issuing organization.

## Step Seventeen: Starting the business network

Run the composer network start command to start the business network. Only Org1 needs to perform this operation. This command uses the endorsement-policy.json file created in step thirteen, and the admin-pub.pem files created by both Alice and Bob in step fifteen and step sixteen, so you must ensure that all of these files are accessible to this command:

Copy

composer network start -c PeerAdmin@byfn-network-org1 -a tutorial-network@0.0.1.bna -o endorsementPolicyFile=endorsement-policy.json -A alice -C alice/admin-pub.pem -A bob -C bob/admin-pub.pem

Once this command completes, the business network will have been started. Both Alice and Bob will be able to access the business network, start to set up the business network, and onboard other participants from their respective organizations. However, both Alice and Bob must create new business network cards with the certificates that they created in the previous steps so that they can access the business network.

## Step Eighteen: Creating a business network card to access the business network as Org1

Run the composer card create command to create a business network card that Alice, the business network administrator for Org1, can use to access the business network:

Copy

composer card create -p connection-org1.json -u alice -n tutorial-network -c alice/admin-pub.pem -k alice/admin-priv.pem

Run the composer card import command to import the business network card that you just created:

Copy

composer card import -f alice@tutorial-network.card

Run the composer network ping command to test the connection to the blockchain business network:

Copy

composer network ping -c alice@tutorial-network

If the command completes successfully, then you should see the fully qualified participant identifier org.hyperledger.composer.system.NetworkAdmin#alice in the output from the command. You can now use this business network card to interact with the blockchain business network and onboard other participants in your organization.

## Step Nineteen: Creating a business network card to access the business network as Org2

Run the composer card create command to create a business network card that Bob, the business network administrator for Org2, can use to access the business network:

Copy

composer card create -p connection-org2.json -u bob -n tutorial-network -c bob/admin-pub.pem -k bob/admin-priv.pem

Run the composer card import command to import the business network card that you just created:

Copy

composer card import -f bob@tutorial-network.card

Run the composer network ping command to test the connection to the blockchain business network:

Copy

composer network ping -c bob@tutorial-network

If the command completes successfully, then you should see the fully qualified participant identifier org.hyperledger.composer.system.NetworkAdmin#bob in the output from the command. You can now use this business network card to interact with the blockchain business network and onboard other participants in your organization.

References:::

https://www.youtube.com/watch?v=t5wZvmZGuAY

https://www.youtube.com/watch?v=S2S5cEcmx7Q

https://www.youtube.com/watch?v=2\_RgCfjunEU

<https://www.youtube.com/watch?v=DKuGU5CYV_E>

https://www.udemy.com/ethereum-masterclass/?couponCode=B21BLOCK

https://www.udemy.com/ethereum-developer/?couponCode=B21BLOCK

https://www.edx.org/course/blockchain-business-introduction-linuxfoundationx-lfs171x

https://conferences.oreilly.com/oscon/oscon-tx/public/schedule/detail/57875