Instrucciones para montar una Red Hyperledger Fabric con contenedores Docker (Docker-Compose)

Lo primero de todo es crear un archivo docker-compose.yml para definir los contenedores y dentro del archivo.

Puesto que los contenedores no van a ir todos dentro de una maquina (en el caso más simple si, pero en una implementación real, no es conveniente) hay que ir dividiendo el archivo “genérico” (asi se llamará ahora en adelante) en diferentes archivos “yml según necesidades

Este es de internet 🡪

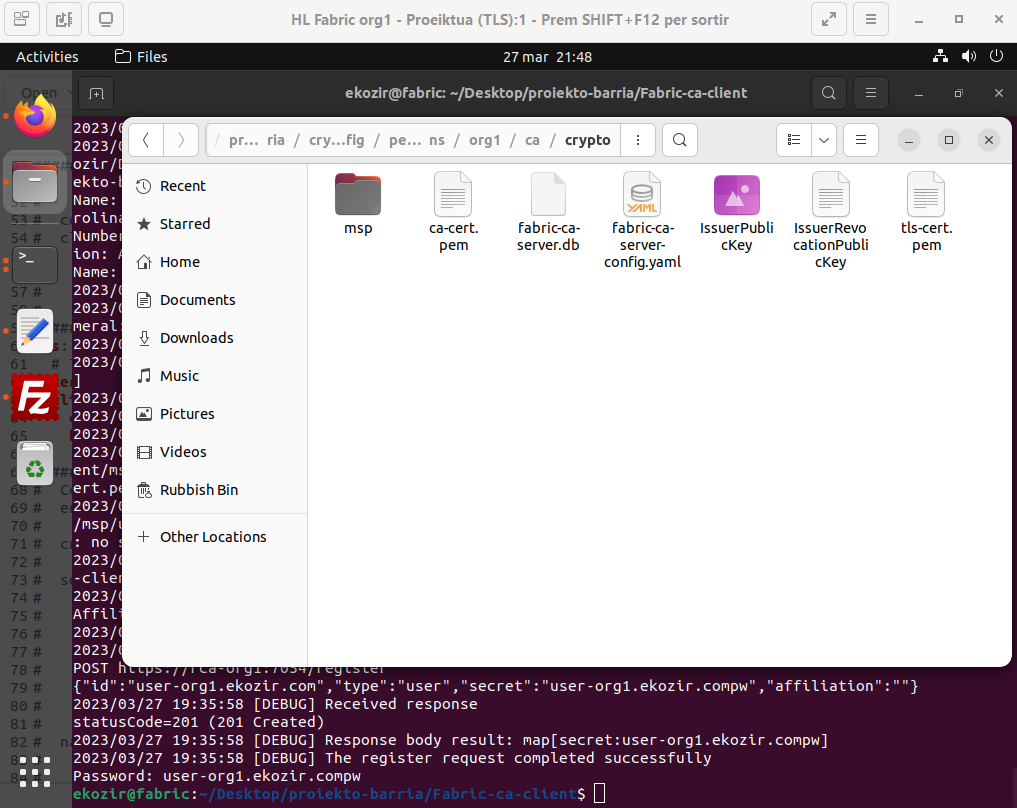
Y este es el mio 🡪

GENERACIÓN DE CERTIFICADOS

Primeramente hay que descargarse el archivo binario del cliente Fabric-ca (llamado como “Fabric-ca-client” 🡪 <https://github.com/hyperledger/fabric-ca/releases> )

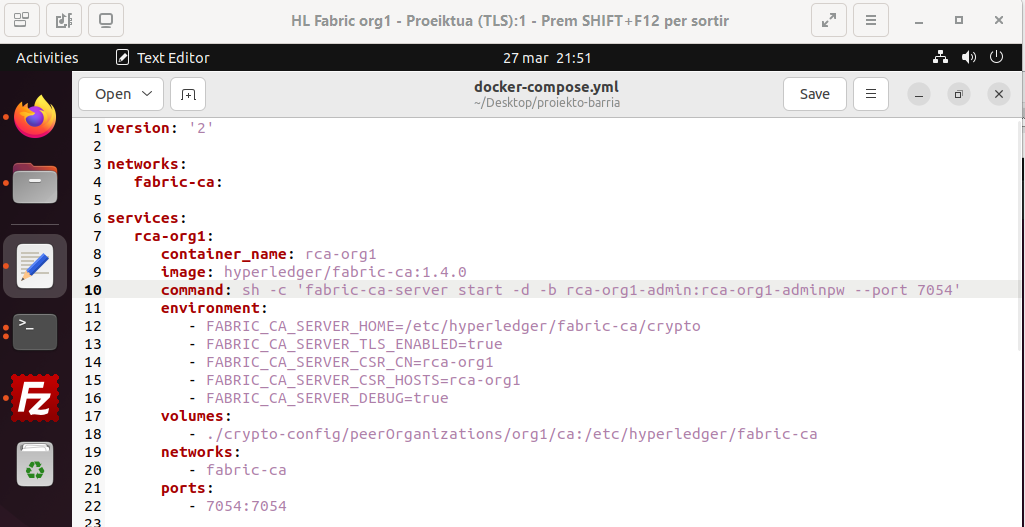
El archivo de Fabric-ca-client lo dejaremos en una carpeta dedicada para ello

En esta carpeta se pondrá SIEMPRE el archivo que habrá generado el servidor ca (Docker ca que habremos emezado en el contenedor), ese archivo se llama ca-cert.pem y se esta junto con el archivo Fabric-ca-server-config.yaml que genera el servidor como aparece en la imagen



Además, se ha generado una carpeta msp y el archivo tls-cert.pem (que es el que presenta el servidor en la comunicación segura). La carpeta de arriba es la carpeta que se define como volumen en el Docker como se verá a continuación en FABRIC\_CA\_SERVER\_HOME.

Para que las líneas se referencien y los certificados se puedan generar correctamente, hay que cambiar el apartado FABRIC\_CA\_SERVER\_CSR\_HOSTS al nombre que queramos o que se llama el host.



Posteriormente, una vez que definamos

export FABRIC\_CA\_CLIENT\_TLS\_CERTFILES **al archivo copiado del servidor ca-cert.pem que hay que poner con el binario de fabric-ca-client**

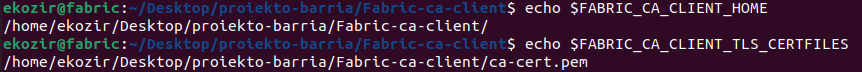
export FABRIC\_CA\_CLIENT\_HOME y la dirección qu ese encuentra el binario

Se puede enrollear al administrador del CA con el siguiente comando

fabric**-**ca**-**client enroll **-**d **-**u https:**//**rca**-**org1**-**admin:rca**-**org1**-**adminpw@rca-org1:7054

o estipulando la salida del msp con el flag –mspdir

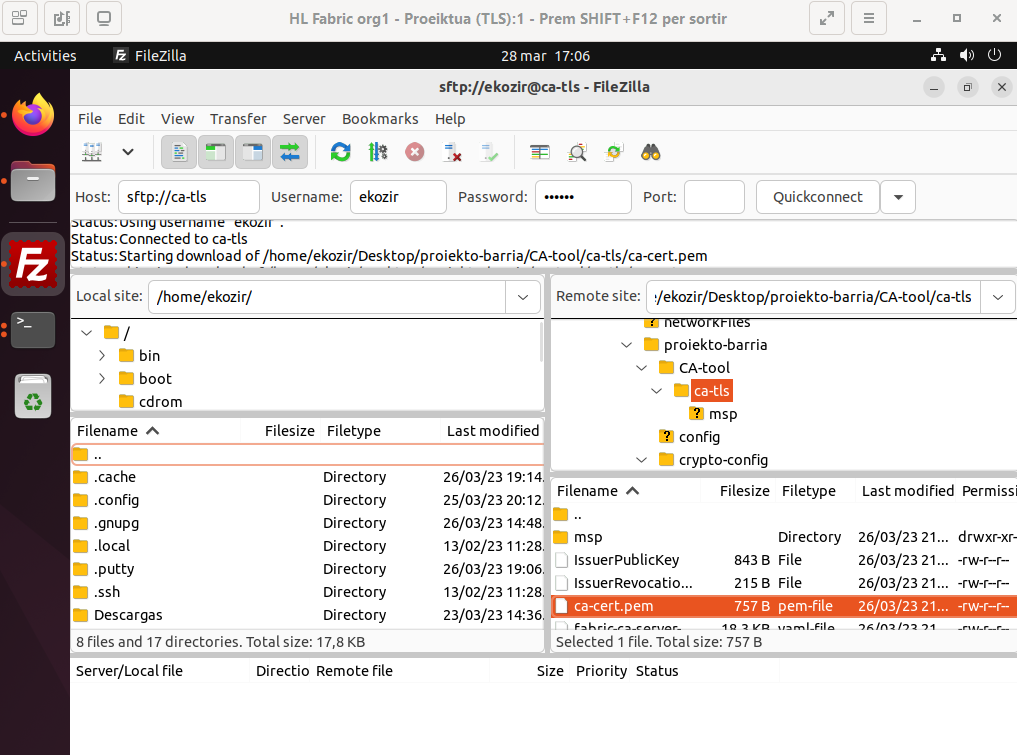
Si no se pone nada, se genera la carpeta junto al binario y si se pone se estipula el sitio donde queremos dejar el msp





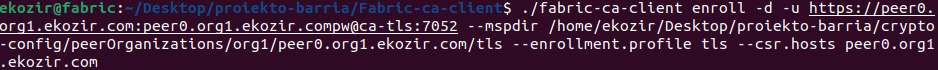
TLS DE PEER0.ORG1.EKOZIR.COM

Se ha copiado mediante Filezilla el archivo ca-cert.pem de TLS-CA



Y se ha copiado en la ruta …../peerOrganizations/org1/peer0.org1.ekozir.com/tls-ca-root-cert.pem

Y se ha usado como archivo en la ENV VARIABLE de export FABRIC\_CA\_CLIENT\_TLS\_CERTFILES para hacer lo siguientes:



Devolviendo todo correctamente a la carpeta puesat en –mspdir

Ahora se hace:

Go to path  /home/ekozir/Desktop/proiekto-barria/crypto-config/peerOrganizations/org1/peer0.org1.ekozir.com/tls/keystore and change the name of the key to key.pem. This will make it easy to be able to refer to in later steps /

ENROLL ORG1 ADMIN

Despues e va a enrollear al administrador del org1 y para eso se hará lo siguiente:

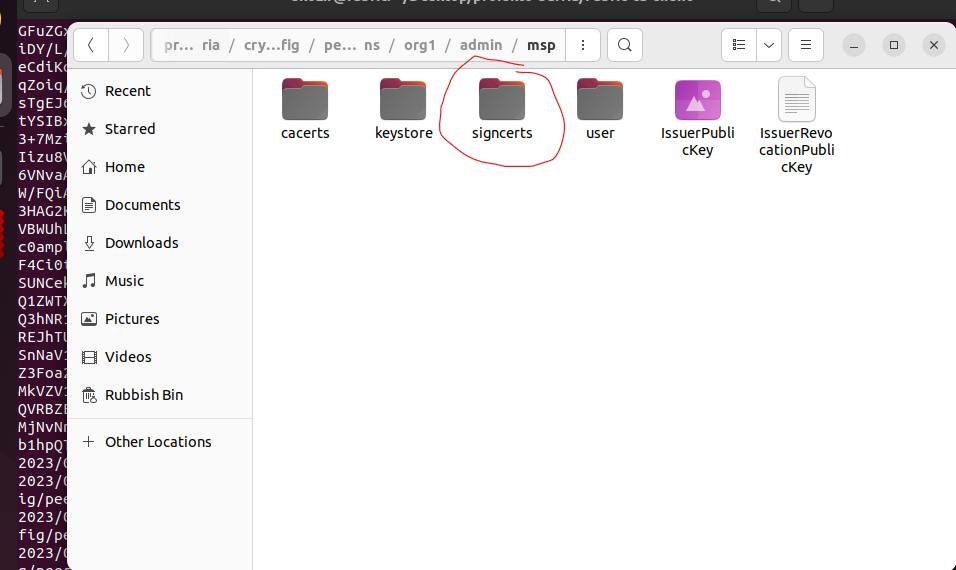
Now, you will enroll Org1’s admin identity. The admin identity is responsible for activities such as installing and instantiating chaincode. The steps below will enroll the admin. In the commands below, we will assume that they are being executed on Peer1’s host machine.

export FABRIC\_CA\_CLIENT\_HOME**=/**home/ekozir/Desktop/proiekto-barria/Fabric-ca-client/

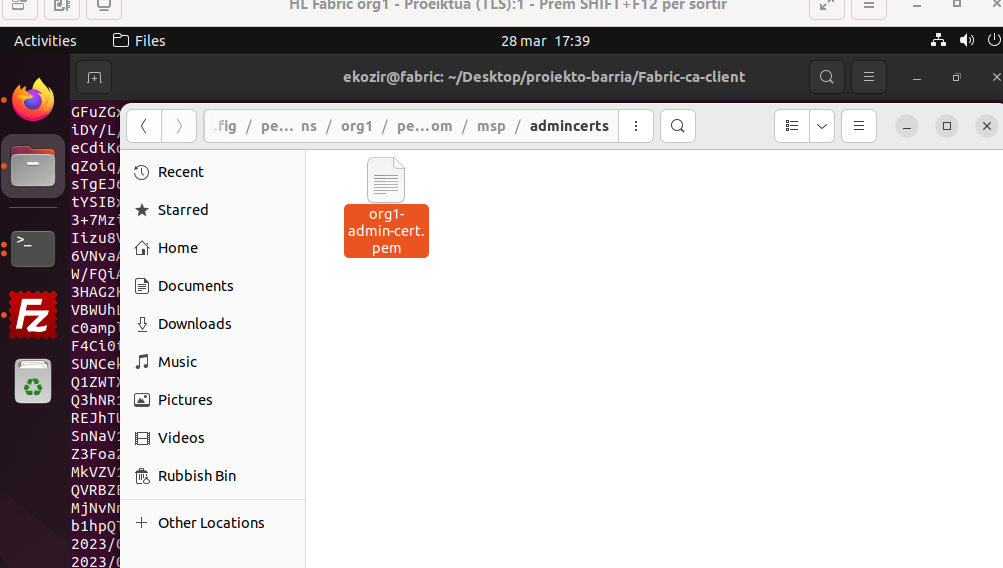
export FABRIC\_CA\_CLIENT\_TLS\_CERTFILES**=**/**home/ekozir/Desktop/proiekto-barria/Fabric-ca-client/ca-cert.pem**

./fabric-ca-client enroll -d -u https://admin-org1.ekozir.com:admin-org1.ekozir.compw@rca-org1:7054 --mspdir /home/ekozir/Desktop/proiekto-barria/crypto-config/peerOrganizations/org1/admin/msp

Luego hay que coger el signcerts/cert.pem del msp de administrador creado y hay que copiar en todas las carpetas admincerts de los peer



Y copiarlo aquí

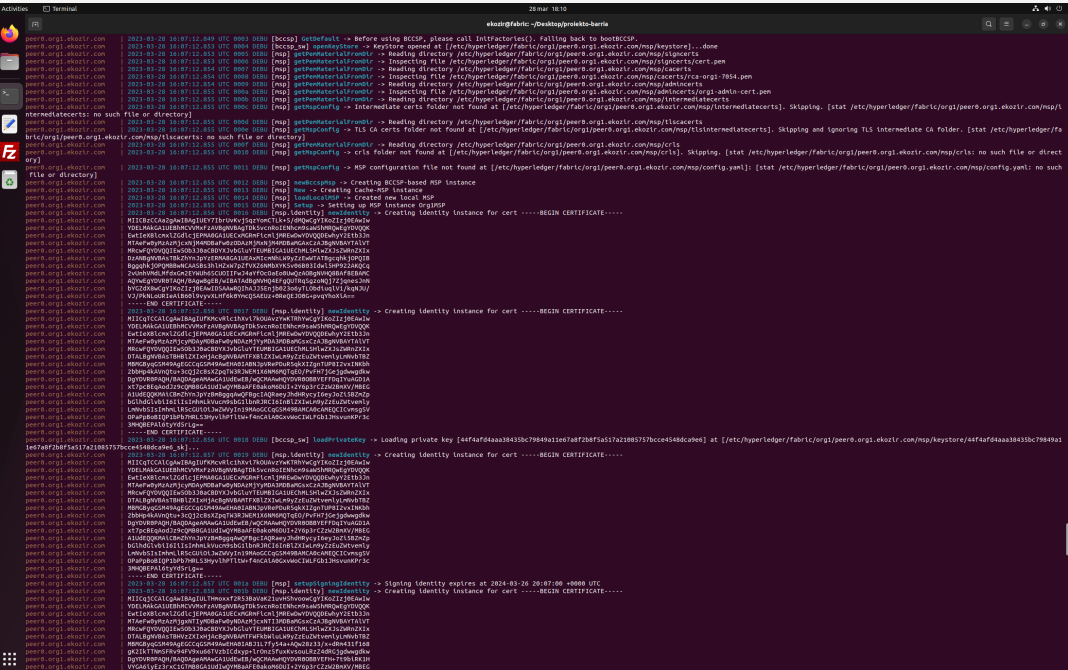


LANZAR PEER0.ORG1

Se lanza con estos códigos de Docker



Cuando arrancamos el peer0 con sus certifiados de msp y tls arranca bien y coherentemente

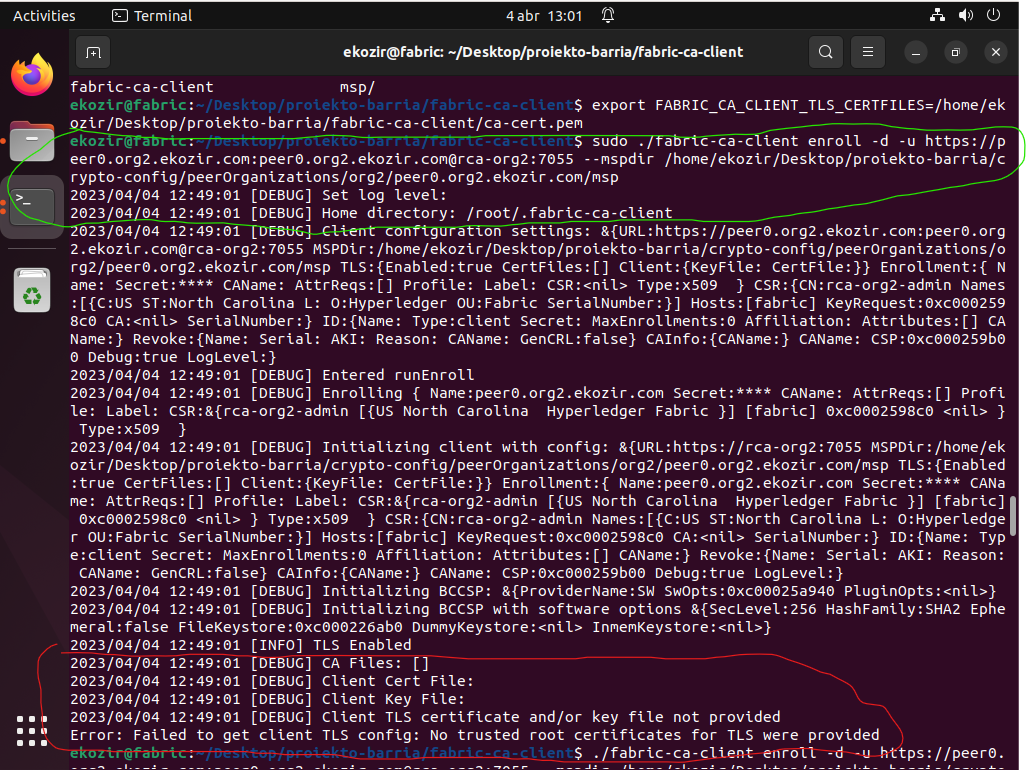


SET UP ORG2’S PEER

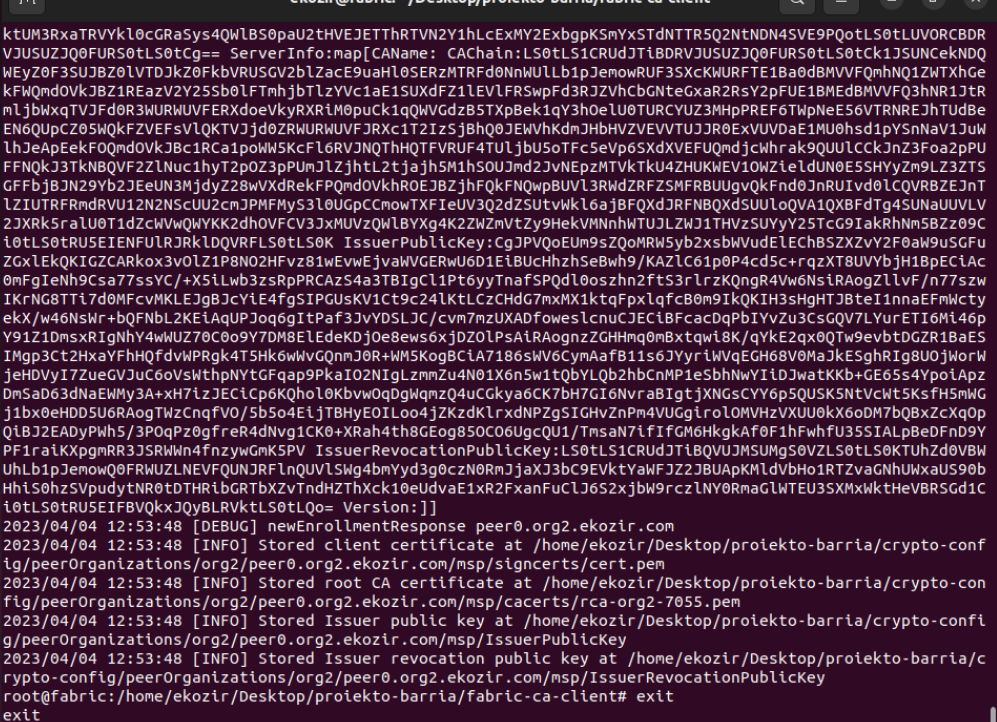
Aquí daba un fallo raro, porque cuando ponía los parámetros de export FABRIC\_CA\_CLIENT\_HOME=/home/ekozir/Desktop/proiekto-barria/fabric-ca-client/

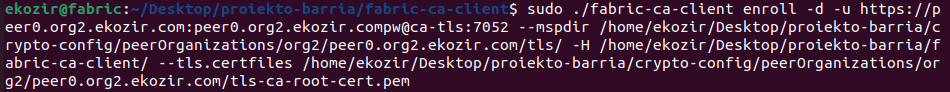
Y export FABRIC\_CA\_CLIENT\_TLS\_CERTFILES=/home/ekozir/Desktop/proiekto-barria/fabric-ca-client/ca-cert.pem

Me daban este error (en rojo) y creo porque era razon del home directory, que ponai algo raro (en verde):



Para subsanar eso, se han hecho todas las operaciones como root y ha salido bien:



Esto sucede cuando utilizamos el comando “sudo” (SUperuser DO), de ese modo los ENVIROMENT VARIABLES no los tiene en cuenta y es como si no existen. Para subsanar eso, se pone como root y se definen los ENVIROMENT VARIABLES o sino, se pasan los enviroment variables en el comando, cómo se ha hecho en la siguiente 

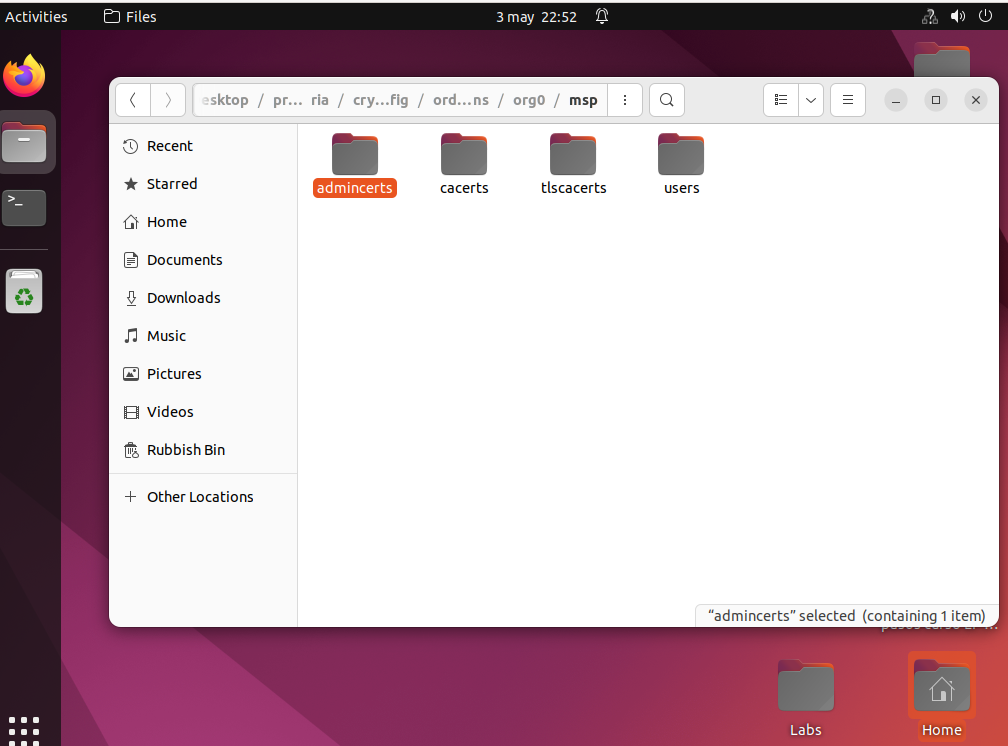
En esta imagen se han generado los certificados TLS para el peero0 del Org2 y se han guardado donde aparece –mspdir.

Este directorio hay que definirlo en el archivo “Docker-compose” para que se utilice para inicializar nuestro peer0.org2.ekozir.com

Posteriormente, hay que generar una carpeta que es “admincerts” que contendrá los certificados de administrador de la organización. Pero para ello hay que enrollear al administrador de la organización y posteriormente copiar esos archivos.

“LAS HOJAS QUE TENGO TODO ANOTADO”

Cuando la carpeta de admincerts esta el certificado de administrador:



Y aquí se copia d ela siguiente manera: (lo siguiente cambiar como lo tengo anotado)

mkdir **/**tmp**/**hyperledger**/**org2**/**peer1**/**msp**/**admincerts

cp **/**tmp**/**hyperledger**/**org2**/**admin**/**msp**/**signcerts**/**cert**.**pem **/**tmp**/**hyperledger**/**org2**/**peer1**/**msp**/**admincerts**/**org2**-**admin**-**cert**.**pem

[**Create Genesis Block and Channel Transaction**](https://hyperledger-fabric-ca.readthedocs.io/en/latest/operations_guide.html#create-genesis-block-and-channel-transaction)

Una vez que todos los MSP-s están creados y definidos, hay que juntarlos en el host del ORG0 (orderer). Para ello se utiliza Filezilla, salvo org0 que tiene todos los certificados en su mismo HOST:

Org0 PONER EL DE TODOS ABAJO BIEN COMO TENGO EN LAS HOJAS

/tmp/hyperledger/org0/msp

├── admincerts

│ └── admin-org0-cert.pem

├── cacerts

│ └── org0-ca-cert.pem

├── tlscacerts

│ └── tls-ca-cert.pem

└── users

The pattern is the same for all organization. The MSP folder structure for Org1 would like:

/tmp/hyperledger/org1/msp

├── admincerts

│ └── admin-org1-cert.pem

├── cacerts

│ └── org1-ca-cert.pem

├── tlscacerts

│ └── tls-ca-cert.pem

└── users

The MSP folder structure for Org2 would like:

/tmp/hyperledger/org2/msp

├── admincerts

│ └── admin-org2-cert.pem

├── cacerts

│ └── org2-ca-cert.pem

├── tlscacerts

│ └── tls-ca-cert.pem

└── users

Se pone la carpeta de “bin” en proiektu barria y se actualiza $PATH con el siguiente comando.

Estando en la carpeta de “proiektu-barria”

configtxgen **-**profile OrgsOrdererGenesis **-**outputBlock config/genesis**.**block **-**channelID syschannel

configtxgen **-**profile OrgsChannel **-**outputCreateChannelTx config**/**channel**.**tx **-**channelID mychannel

