

TRILHA BLOCKCHAIN HYPERLEDGER FABRIC

Criando uma rede e instalando um chaincode em uma rede Hyperledger Fabric 2.2

Configuração da VM

A VM já está com as seguintes ferramentas configuradas (necessárias para desenvolvimento Hyperledger Fabric):

- Curl
- Git
- Docker
- Docker-compose
- GoLang

Limpar os containers da VM com o seguinte comando.

```
docker stop $(docker ps -a -q) && docker rm $(docker ps -a -q) && docker rmi -f $(docker images) && docker volume prune && docker system prune
```

Tarefa 1

Ir para o diretório */home/student*

Baixar as ferramentas Hyperledger Fabric versão 2.2

```
curl -sSL https://bit.ly/2ysb0FE | bash -s -- 2.2.9 1.5.5
```

```
export PATH=$PATH:$HOME/fabric-samples/bin
```

Acessar o diretório *fabric-samples/test-network*.

Subir a rede

```
./network.sh up
```

Desligar rede

```
./network.sh down
```

Gravar os resultados das duas operações

Tarefa 2

Subir a rede utilizando CA

```
./network.sh up -ca
```

Gravar os resultados das duas operações

Gravar a inspeção da transação de criação bloco gênese com o seguinte comando.

TRILHA BLOCKCHAIN HYPERLEDGER FABRIC

configtxgen -inspectBlock ./system-genesis-block/genesis.block

Tarefa 3

Criar um channel *mychannel* e adicionar as orgs e configurar os anchor peers.

./network.sh createChannel

Gravar o resultado da operação.

Gravar a inspeção da transação de criação do channel *mychannel*.

Tarefa 4

Apagar o diretório *vendor* do diretório *chaincode/fabcar/go*

```
sudo rm -rf ~/go/pkg/  
rm -rf vendor
```

Vendorar o chaincode

go mod vendor

Gravar o resultado.

Tarefa 5

Instalar e instanciar um chaincode *fabcar* no channel *mychannel*.

./network.sh deployCC -ccn fabcar -ccp ../chaincode/fabcar/go -ccl go

Gravar o resultado da operação

Tarefa 6

Chamar a função *initLedger* do chaincode *fabcar*

```
export FABRIC_CFG_PATH=$PWD/../../config/
```

```
export CORE_PEER_TLS_ENABLED=true
```

```
export CORE_PEER_LOCALMSPID="Org1MSP"
```

```
export  
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
```

```
export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
```

```
export CORE_PEER_ADDRESS=localhost:7051
```

TRILHA BLOCKCHAIN HYPERLEDGER FABRIC

```
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile
"${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C
mychannel -n fabcar --peerAddresses localhost:7051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --
tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c
'{"function": "InitLedger", "Args": []}'
```

Mostrar o resultado da operação

Tarefa 7

Chamar a função `queryAllCars` chaincode `fabcar`

```
peer chaincode query -C mychannel -n fabcar -c '{"Args":["QueryAllCars",""]}'
```

Mostrar o resultado da operação.