Kroki do uruchomienia sieci:

C:\Users\filip\Documents\GitHub\blockchain-biz-secure\ca

Uruchamia serwery certyfikujące TLS i tożsamości

docker-compose up

ca/\_config\_files/shared/spread-config.sh

Rozprzestrzenienie pliku config.yaml odpowiedzialnego za NodeOUS

Następnie uruchamiamy orderery

docker-compose -f docker-compose.orderer.all.yaml

Później węzły z tożsamościami administratorów do stworzenia kanału i dodania ordererów do tych kanałów.

docker-compose -f docker-compose.configtx.all.yaml up

I peery o roli Anchor

C:\Users\filip\Documents\GitHub\blockchain-biz-secure\ca\\_scripts

Uruchamiac należy z ubuntu wsl ponieważ inaczej nie zadziała mapowanie na socket dockera, do tworzenia kontenera chaincodu

./run-anchor-peers.sh

Następnym krokiem jest uruchomienie pozostałych peerów czyli peer1 i peer2 dla każdej z sieci – skrypt pobierze blok kanału od węzła typu orderer i tym samym rozpocznie komunikację.

Ważne jest aby dla ich ustawień zmienić ta linijkę:

*# This is an endpoint that is published to peers outside of the organization.  
# If this isn't set, the peer will not be known to other organizations.  
#externalEndpoint: "peer0.furnituresmakers.com"*

W przykładowej sieci korzystam ze statycznego wyboru anchor peera, więc ustawiam:

*# Defines whenever peer will initialize dynamic algorithm for  
# "leader" selection, where leader is the peer to establish  
# connection with ordering service and use delivery protocol  
# to pull ledger blocks from ordering service. It is recommended to  
# use leader election for large networks of peers.*useLeaderElection: false  
*# Statically defines peer to be an organization "leader",  
# where this means that current peer will maintain connection  
# with ordering service and disseminate block across peers in  
# its own organization*orgLeader: true *#There is need for at least one leader since useLeaderElections is false*

*natomiast, dla peerów wewnątrz organizacyjnych ustawienie orgLeader=false*

Pojedyńczego peera uruchamiam za pomocą skryptu on tak samo pobiera pierwszy blok kanału.

./run-peer.sh peer1 Furnitures\_Makers 7051

Przykład aliveMessage:

2025-01-17 00:52:27.537 UTC 010c DEBU [gossip.discovery] handleAliveMessage -> Entering GossipMessage: Channel: , nonce: 0, tag: EMPTY Alive Message:Membership: Endpoint:peer0.furnituresmakers.com:7051 PKI-id:772b06de0d93863e78750a73bc0cb8028ab44bef02c779b0d3a0d1281f7a7eb4Identity:Timestamp:inc\_num:1737074851082847553 seq\_num:154 , Envelope: 88 bytes, Signature: 70 bytes Secret payload: 33 bytes, Secret Signature: 71 bytes

Dodatkowo istnieje skrypt uruchamiający wszystkie przewidziane peery:

./run-all-inner-peers.sh

peer chaincode query \

-C yfw-channel \

-n basic \

-c '{"Args":["TokenSymbol"]}' \

--tls \

--cafile $CA\_FILE

peer chaincode query -C yfw-channel -n basic -c '{"Args":["TokenSymbol"]}' --tls --cafile $CA\_FILE



2025-01-28 12:23:13.167 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200

