ESCOLA SUPERIOR DE REDES

Módulo 6
Hyperledger Fabric
(finalização)

GoLedger

Queries no channel

As buscas são feitas dentro do peer através de operações de pesquisa.

As pesquisas são realizadas nas bases de *state* escolhida (***LevelDb*** ou ***CouchDb***)

LevelDb é a base default e roda dentro do container peer

Pesquisas em ***LevelDb*** são realizadas para as chaves (primárias ou compostas)

CouchDb é um modelo mais poderoso e roda em um container separado do *peer*.

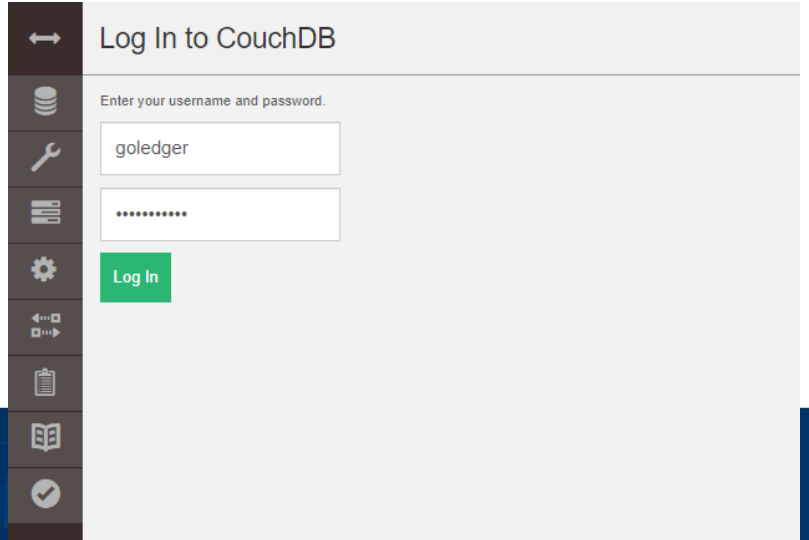
Rich queries e indexação são permitidas para ***CouchDb***

Trabalhando com CouchDb

O uso do CouchDb pode ser realizado com a configuração dos endorsing peer

Váriáveis de ambiente:
CORE_LEDGER_STATE_STATEDATABASE=CouchDB
CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couch.peer0.org.example.com:5984
CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=admin
CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=adminpw

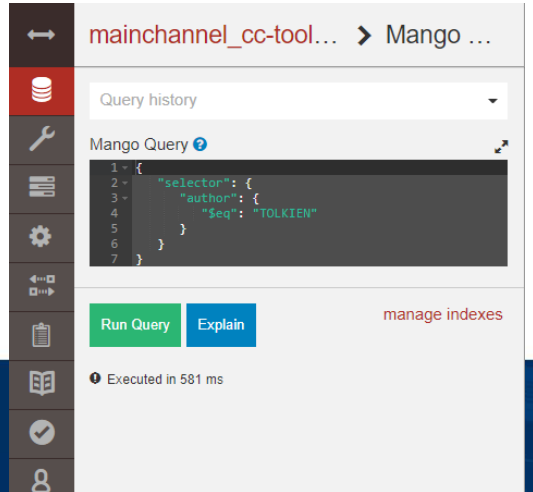
Exemplo de acesso ao couchdb via interface Web (*Fauxion*)
http://localhost:5984/_utils/#login



Usando a interface do *CouchDB*

O base do *state* é identificada pelo nome do *channel* junto com o nome do *chaincode*.

Queries podem ser executadas da interface através do item *Run Query with Mango*



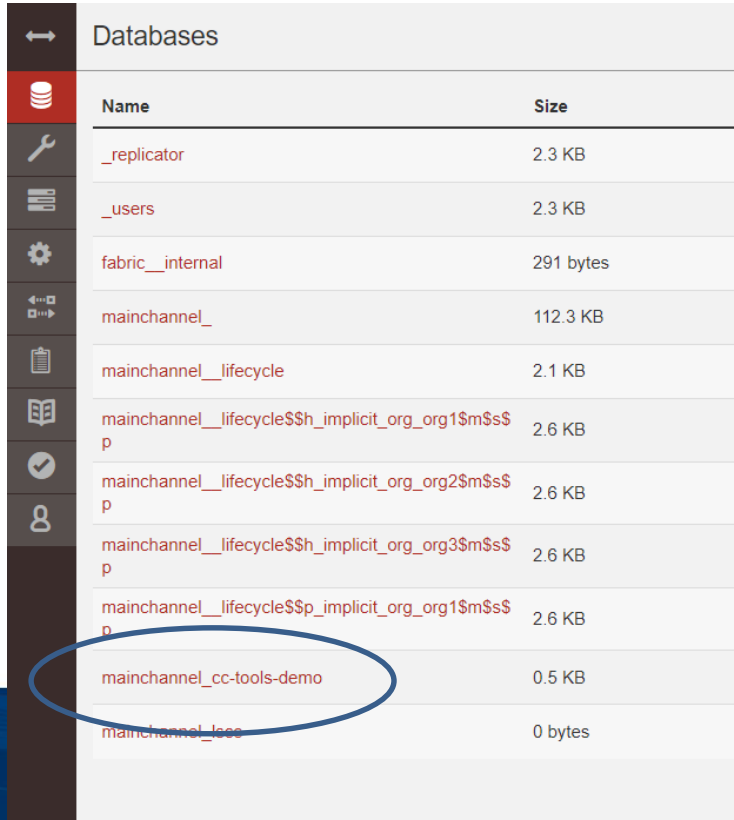


Tabela de Query CouchDb

Queries podem ser realizadas através de uma sintaxe. Maiores detalhes na documentação oficial do CouchDb

https://docs.couchdb.org/en/3.2.2-docs/api/database/find.html

Exemplo:
Busca documentos com status “draft”

```
"selector": {
   "status": { "$eq": "draft" }
}
```

Busca asset com year maior ou igual a 2020

```
“selector”: {
    "year": { "$gte": 1900 }
 }
```

Função *Search*

Função CC-Tools utiliza a função *Search* do pacote assets para realizar busca no *CouchDb*

func Search(stub *sw.StubWrapper, request map[string]interface{}, privateCollection string, resolve bool) (*SearchResponse, errors.ICCError)
request – selector da query
privateDataCollection – pequisa na base do PVC
resolve – resolve as referências dentro do asset.

Exemplo:

```
query := map[string]interface{}{
                "selector": map[string]interface{}{
                "@assetType": "book",
                "author":     authorName,
                },
}

response, err := assets.Search(stub, query, "", true)
```

Utilizando funções de pesquisa

O Hyperleger Fabric possui diversas funções para realizar queries dentro do chaincode. Essas funções trabalham com iterators dentro de loops.

Exemplos:

GetQueryResult
GetQueryResultWithPagination
GetStateByPartialCompositeKey
GetHistoryForKey

Exemplo:

```
resultsIterator, _ = stub.GetQueryResult(queryString)
for resultsIterator.HasNext() {
        queryResponse, err := resultsIterator.Next()
...
```

Private Data Collections

A biblioteca CC-Tools possui mecanismos para trabalhar com PDCs.

Assets que terão as informações gravadas em PDCs devem ter o operador *Readers*.

Exemplo de um asset em PDC

```
var Secret = assets.AssetType{
    Tag:         "secret",
    Label:       "Secret",
    Description: "Secret between Org2 and Org3",
    Readers: []string{"org2MSP", "org3MSP"},
    Props: []assets.AssetProp{
        {
            IsKey:    true,
            Tag:      "secretName",
            Label:    "Secret Name",
            DataType: "string",
            Writers:  []string{`org2MSP"}, // This means only org2 can create the asset (org3 can edit)
        },
        {
Tag:      "secret",
            Label:    "Secret",
            DataType: "string",
        },
    },
}
```

Arquivo collections2.json

Cada *asset PDC (Readers)* deve ter um elemento correspondente dentro do arquivo *collections2.json*

```
[
    {
        "name": "secret",
        "requiredPeerCount": 0,
        "maxPeerCount": 3,
        "blockToLive": 1000000,
        "memberOnlyRead": true,
        "policy": "OR('org2MSP.member', 'org3MSP.member')"
    }
]
```

Escalabilidade Hyperledger Fabric

Operações de escalabilidade

Um rede Hyperledger Fabric é uma rede viva.

Diversas operações de rede podem ser realizadas, por exemplo:

- Adicionar novo peer em uma organização
- Adicionar nova organização em um channel
- Adicionar/atualizar um novo chaincode em um channel.
- Adicionar um novo orderer em uma rede
- Remover um peer de uma organização
- Adicionar novo cliente (web service) de uma organização.

Adicionando uma organização no fabric-samples

Dentro da pasta *fabric-samples/test-network* subir uma rede com channel *channel1*

./network.sh up

./network.sh createChannel –c channel1

Após a criação da rede ir para a pasta addOrg3

O script de adicionar org

/addOrg3.sh up -c channel1

Adicionar nova org

Operação AddOrg

1. Criação dos artefatos criptográficos (MSP) na nova organização
2. Configuração do arquivo configtx.yaml para definição da nova org
3. Geração da transação de entrada de nova org no channel
4. Atualização do ultimo bloco de configuração para entrada da nova org no channel
5. Assinatura da transação pelas organizações administradoras do channel de acordo com a Policy de Admin
6. Aplicação do novo bloco de configuração no channel
7. Subir os containers dos peers da nova org
8. Pedido de entrada da nova org no channel.
9. Channel atualizado.

configtx.yaml da nova org

```
Organizations:
    - &Org3
        Name: Org3MSP
        ID: Org3MSP
        MSPDir: ../organizations/peerOrganizations/org3.example.com/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('Org3MSP.admin', 'Org3MSP.peer', 'Org3MSP.client')"
            Writers:
                Type: Signature
                Rule: "OR('Org3MSP.admin', 'Org3MSP.client')"
            Admins:
                Type: Signature
                Rule: "OR('Org3MSP.admin')"
            Endorsement:
                Type: Signature
                Rule: "OR('Org3MSP.peer')"
```

Geração de MSP e definição da nova organização

Geração dos artefatos criptográficos da nova organização

cryptogen generate --config=org3-crypto.yaml --output="../organizations“

Geração da definição da nova organização

configtxgen -printOrg Org3MSP > ../organizations/peerOrganizations/org3.example.com/org3.json

Subindo o novo peer da nova organização

docker-compose -f docker/docker-compose-org3.yaml up -d

Atualizando o bloco de configuração

Recuperar o ultimo bloco de configuração do channel atráves de um peer de uma org que já esteja no peer.

peer channel fetch config channel-artifacts/config_block.pb -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c channel1 --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem“

O bloco de configuração é gravado como protobuf (binário)

Para atualizar o bloco de configuração deve-se usar a comando configtxlator transformar os dados para JSON e de volta para protobuf

O novo bloco é trabalhado externamente para poder conceder a entrada da nova org no channel.

Aplicando o novo bloco de configuração no channel

O novo bloco gerado precisa ser assinado por todas as organizações definidas na configuração do channel.

setGlobals $ORG

peer channel signconfigtx -f channel-artifacts/org3_update_in_envelope.pb

A atualização é aplicada ao *channel*

peer channel update -f channel-artifacts/org3_update_in_envelope.pb -c channel1 -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"

Pedido de entrada no channel pela nova org

Nova org deve obter o *genesis block* para pedir entrada

peer channel fetch 0 channel-artifacts/channel1.block -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com -c channel1 --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem"

Realizar operação de *joinPeer*

peer channel join -b channel-artifacts/channel1.block

Após a operação bem sucedida a nova org se encontra definitivamente no *channel*

Orquestradores Blockchain

Orquestradores Hyperledger Fabric

Estão disponíveis diversos orquestrados para facilitar a operação em redes Hyperledger Fabric

- Orquestradores open source
- Orquestradores em nuvem
- Orquestradores licenciados

Orquestradores open-source

Disponibilizados pela Hyperledger Foundation

Hyperledger Cello
https://www.hyperledger.org/use/cello

Hyperledger Bevel
https://www.hyperledger.org/use/bevel

Fabric Operator (Hyperledger Labs)
https://labs.hyperledger.org/labs/hlf-operator.html

Orquestradores em nuvem e licenciados

Disponíveis na nuvens públicas

Amazon Managed Blockchain

https://aws.amazon.com/pt/managed-blockchain/

IBM Blockchain Platform

https://www.ibm.com/products/blockchain-platform-hyperledger-fabric

Oracle Blockchain Platform Service

https://www.oracle.com/br/blockchain/cloud-platform/

Binario Cloud Blockchain (GoFabric engine)

https://binario.cloud/produtos/blockchain

GoFabric

Orquestrador disponibilizado pela empresa GoLedger

https://gofabric.io/

- Lista de redes Hyperledger Fabric
- Dashboard de rede
- Gestão multichannel
- Operações de escalabilidade da rede (*addPeer*, *AddOrg*, etc)
- Integração automática com chaincodes desenvolvidos com a biblioteca CC-Tools
- Atualização de chaincodes
- Atualização de APIs
- Mapamento de dados dos chaincodes utilizando *Templates* de dados

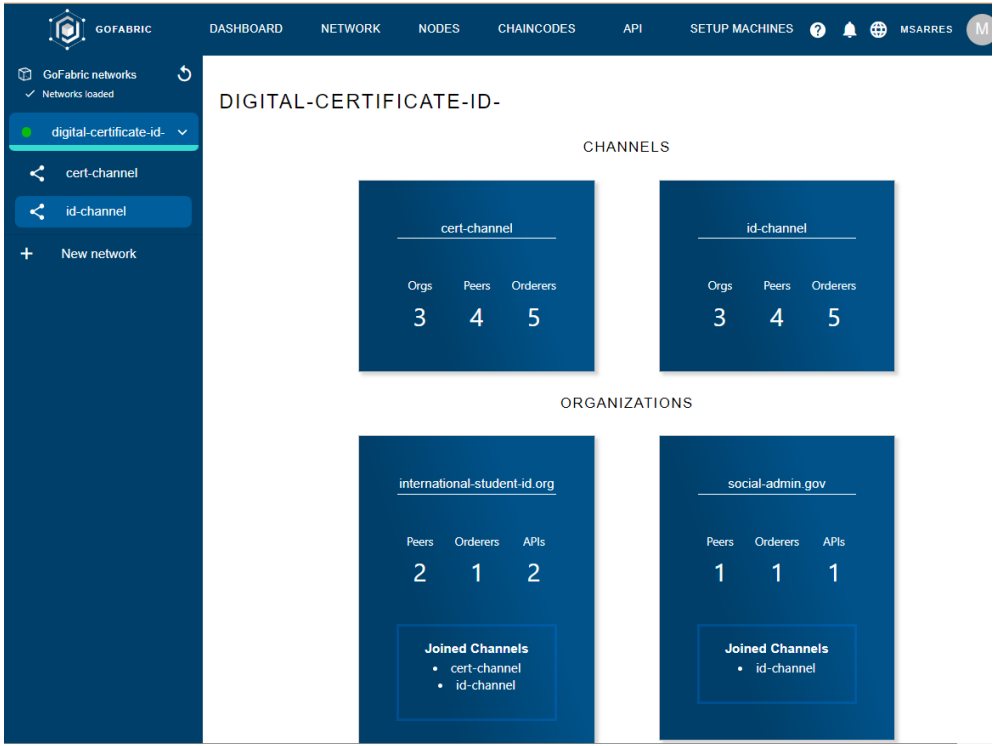
GoFabric

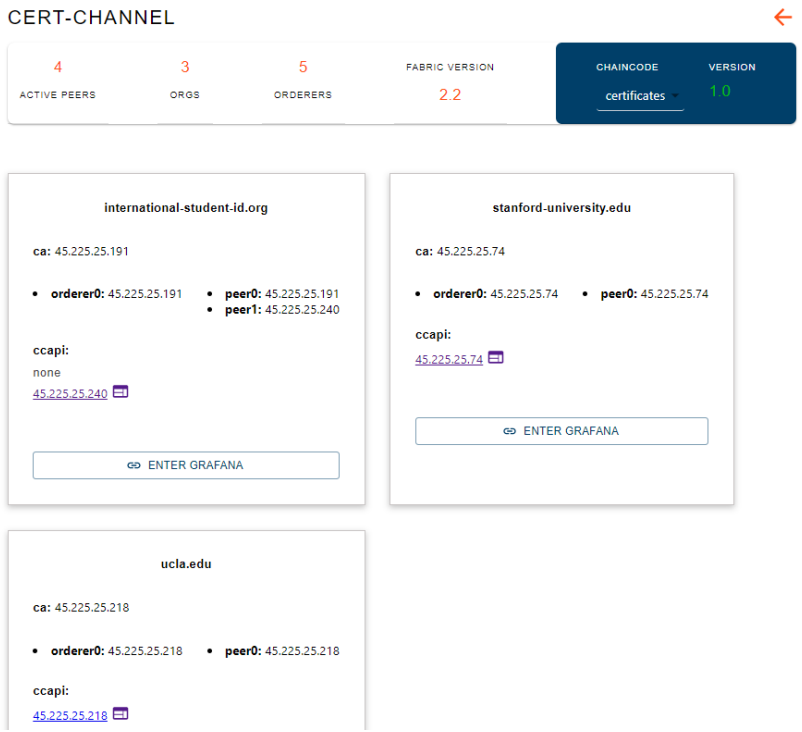
Orquestrador disponibilizado pela empresa GoLedger

https://gofabric.io/

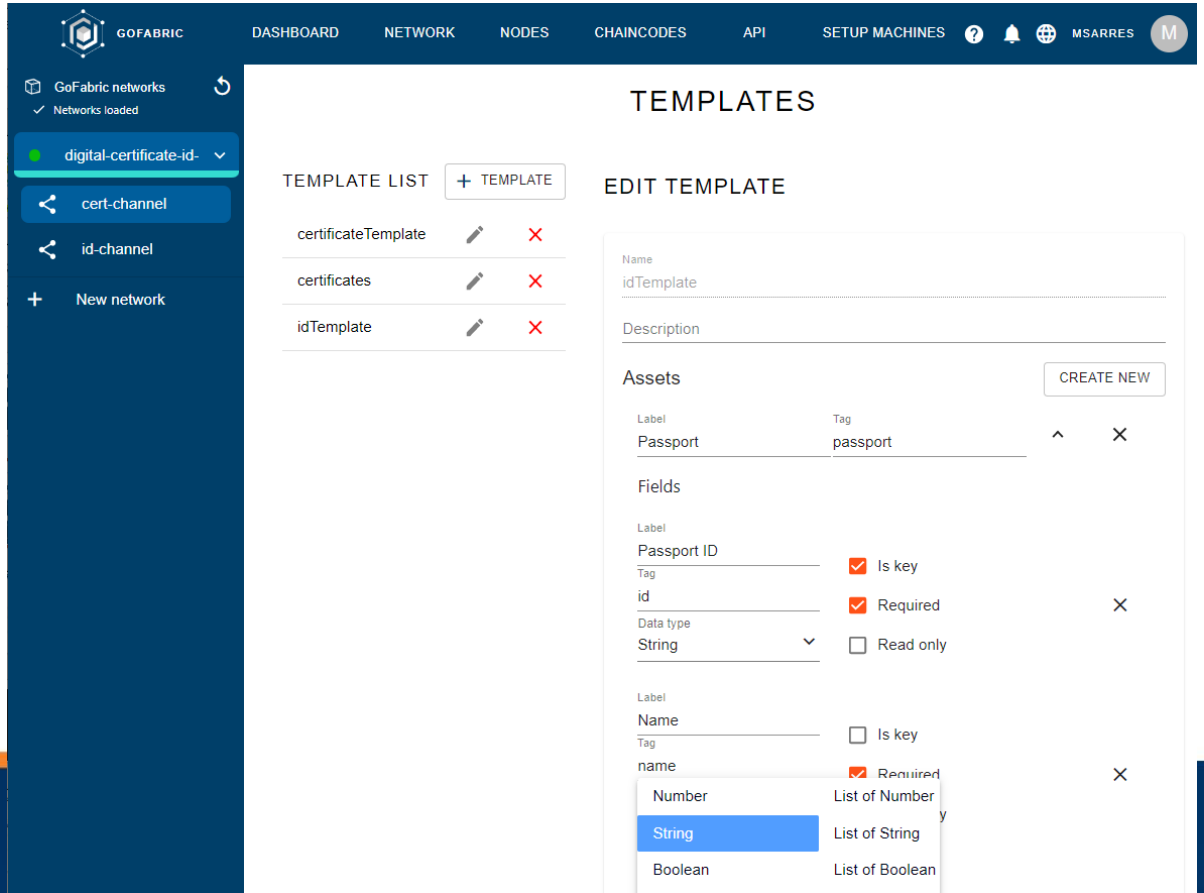
- Lista de redes Hyperledger Fabric
- Dashboard de rede
- Gestão multichannel
- Operações de escalabilidade da rede (*addPeer*, *AddOrg*, etc)
- Integração automática com chaincodes desenvolvidos com a biblioteca CC-Tools
- Atualização de chaincodes
- Atualização de APIs
- Mapamento de dados dos chaincodes utilizando *Templates* de dados

GoFabric – Dashboard



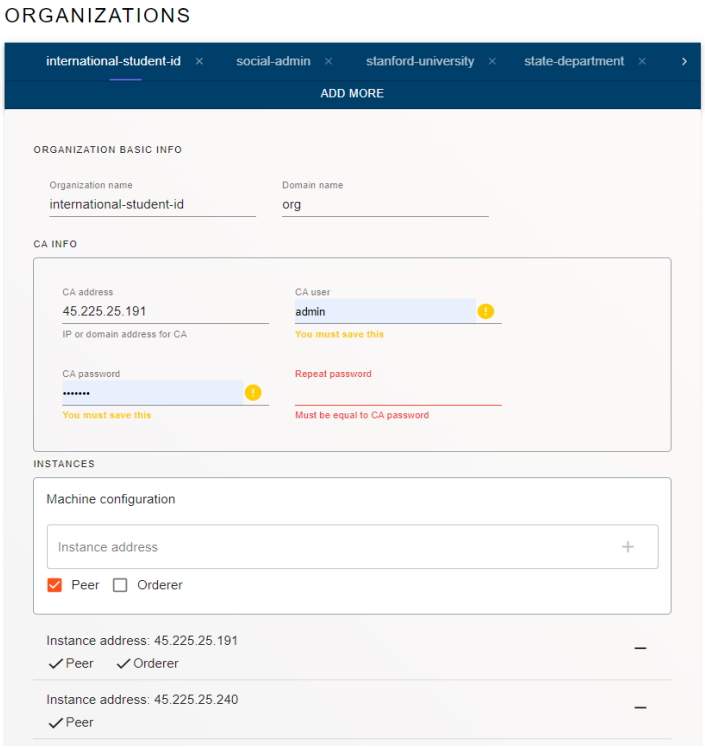


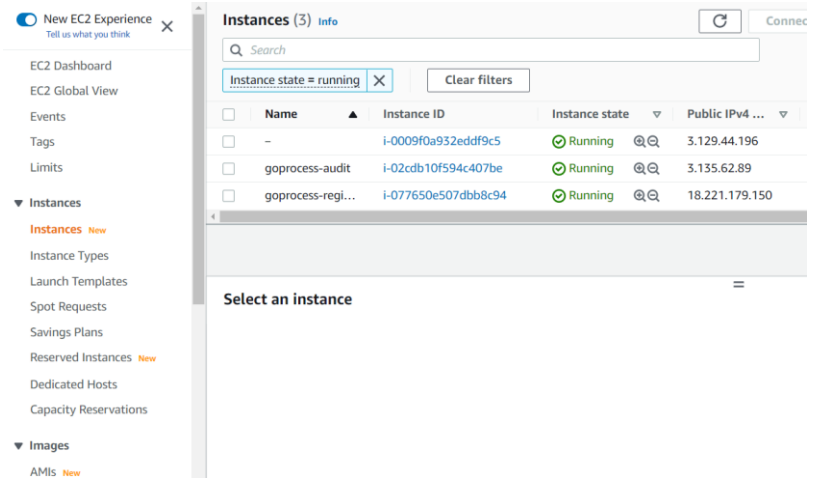
GoFabric – Data Mapping Templates



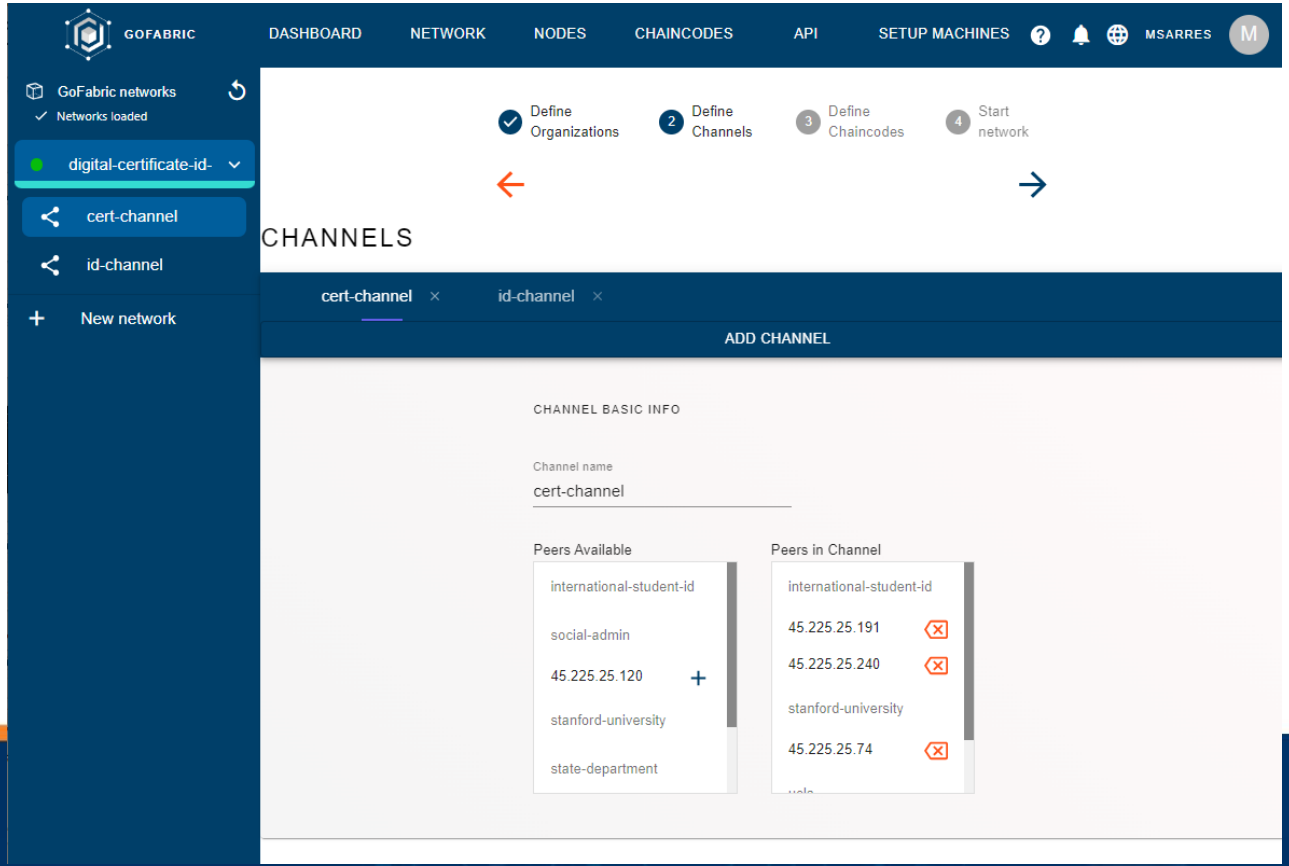
ESCOLA SUPERIOR DE REDES

GoFabric – deployment na nuvem



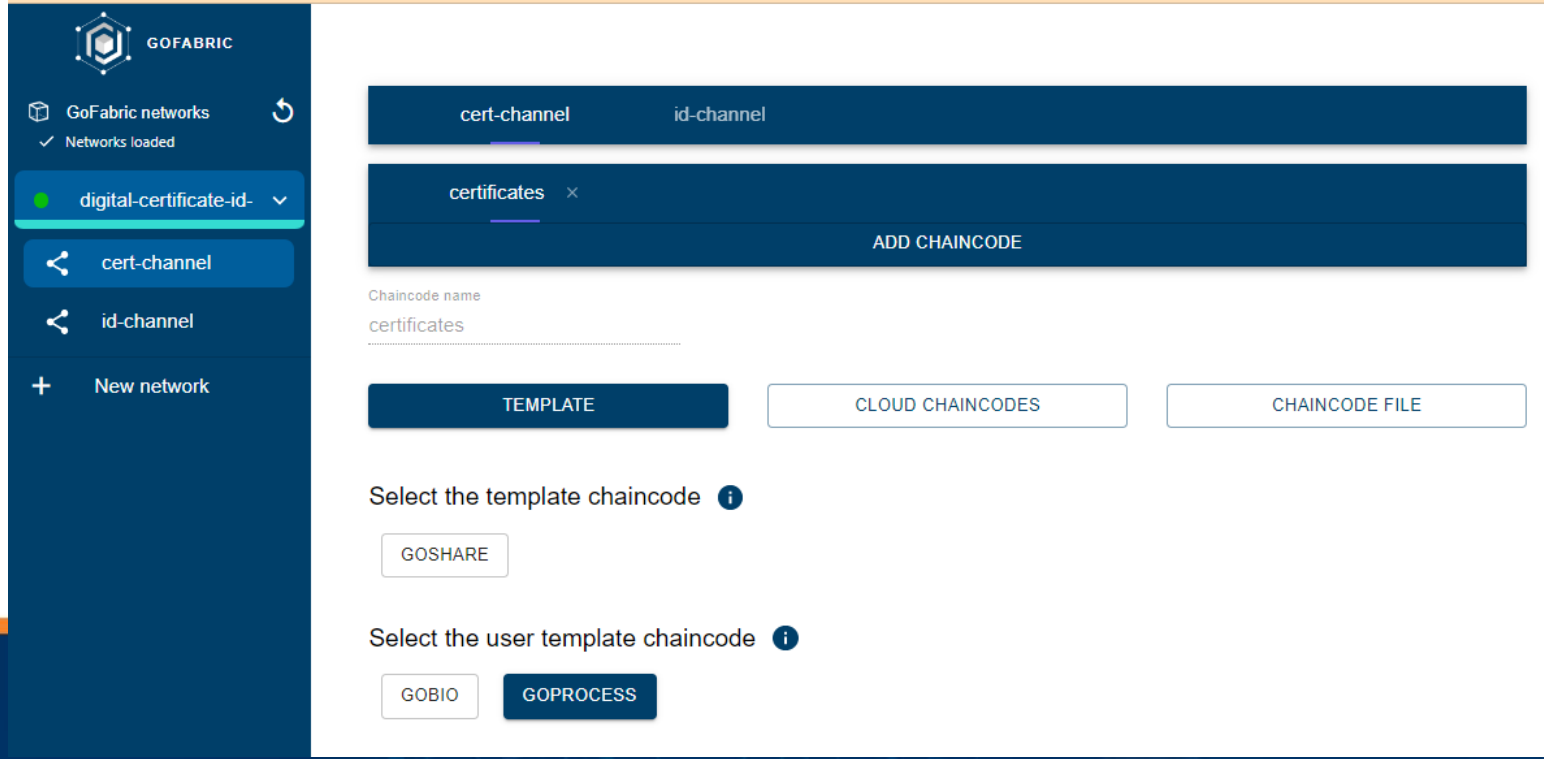


GoFabric – multichannel



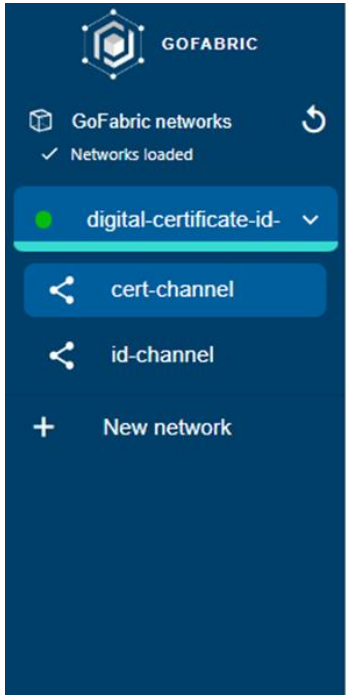
ESCOLA SUPERIOR DE REDES

GoFabric – Chaincode marketplace



ESCOLA SUPERIOR DE REDES

GoFabric – Private Chaincode



PERMISSIONS

If no permissions are set, every organization in the network will be able to read and write to every asset

Passport

Asset

international-student-id, stanford-university

Private data

- international-student-id
- social-admin
- stanford-university
- state-department
- ucla

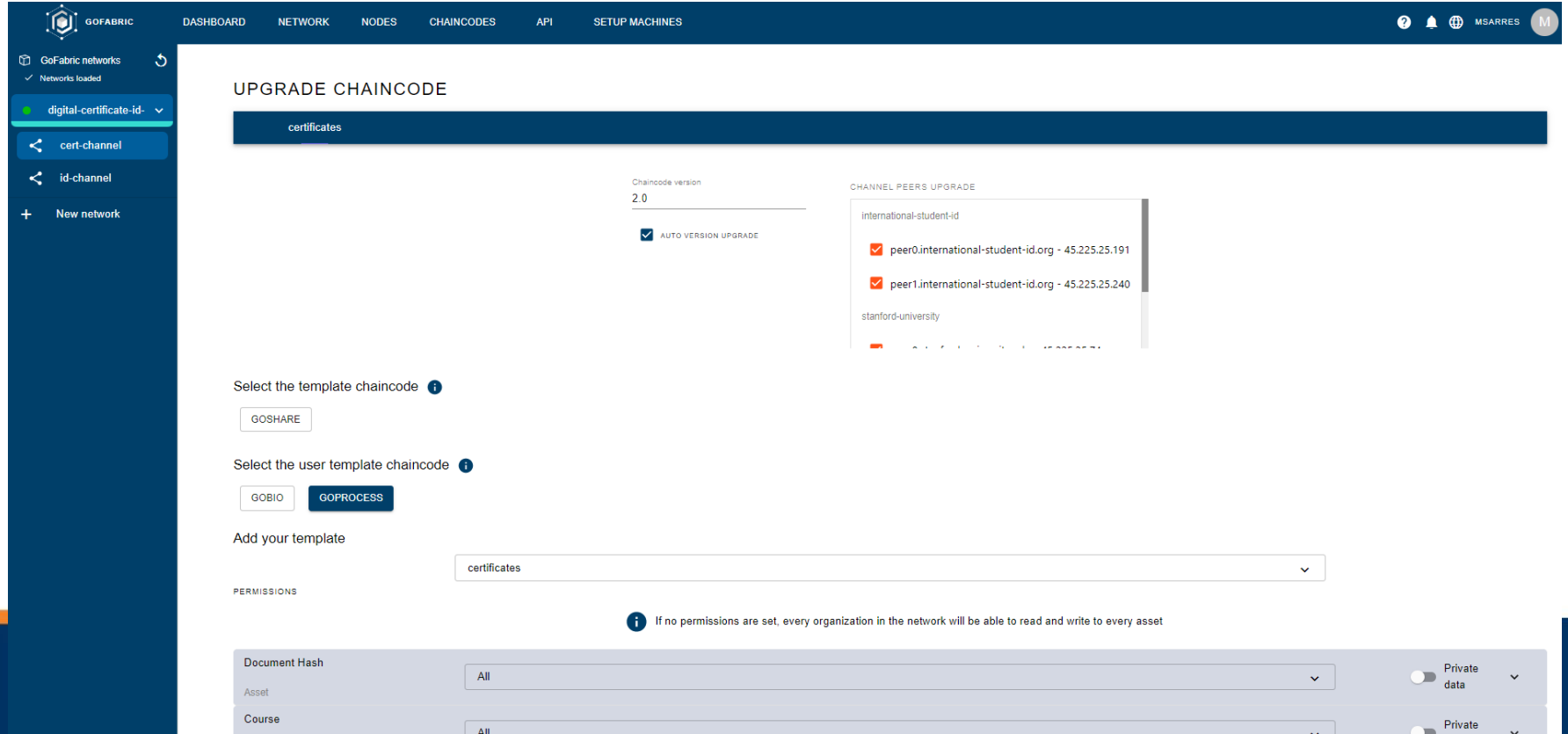
Private data

Passport ID string

Prop

All enabled

GoFabric – Atualizar Chaincode



ESCOLA SUPERIOR DE REDES

Tarefa

GoLedger

RNP ORGANIZAÇÃO SOCIAL DO MCTI

MINISTÉRIO DO TURISMO
MINISTÉRIO DA DEFESA
MINISTÉRIO DA SAÚDE
MINISTÉRIO DAS COMUNICAÇÕES
MINISTÉRIO DA EDUCAÇÃO
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES