Hands-on Minifabric

3 Orgs, chaincode e Web API em Node.js e APP Web com Vue.js

Link com todas as informações do Minifabric:
https://github.com/hyperledger-labs/minifabric

Pré-requisitos para o Hands-on
- Sistema Operacional Linux (Sugestão o Ubuntu 22.04)
- Docker
  - Link bem explicativo para instalar o docker corretamente:
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-pt

Dicas Importantes:

- **Se for utilizar o VirtualBox para testar a conexão com outros nós localmente:**
  - Desabilitar o Boot de segurança **na BIOS.**
  - Se tiver problema de conexão de rede com a VM em modo **NAT** colocar a placa de rede da VM em modo **"bridge"** para seguir com a implementação, isso fará com que a placa de rede do host atue como uma "ponte" entre as redes do host 1 e host 2(VM).
- **Faça backup do chaincode** criado e deixe separado, pois quando a rede é reiniciada, **a pasta VARS tem todos os seus arquivos deletados.**
- **Para realizar os comandos do minifabric** executar dentro da pasta **mywork** onde contém o binário do **minifab** e o arquivo **spec.yaml**
- Caso tenha algum problema ao executar os comandos do **minifab** antes de tentar resolver de outra forma tenta executar com **sudo.**
- Monitorar os logs dos containers docker sugestão utilizar o **Portainer.IO** ajudará a entender o que está acontecendo num nível mais baixo ("Debug da Rede").
  - **LInk para instalação do Portainer.IO:** https://docs.portainer.io/start/install-ce/server/docker/linux

Dowload do Minifabric

Comando para fazer o download da versão atual do Minifabric, irá criar a pasta mywork com o binário do minifabric:

mkdir -p ~/mywork && cd ~/mywork && curl -o minifab -sL https://tinyurl.com/yxa2q6yr && chmod +x minifab

Criando a Rede

Passo 1: Configurar o arquivo spec.yaml e deixar na raiz da pasta mywork das organizações (cada uma tem o seu arquivo específico). (Ver arquivos disponibilizados)

Passo 3 - Colocar a pasta do chaincode na seguinte estrutura de pastas das organizações:

mywork/vars/**chaincode**/**chaincodenerdstore**/node/[ARQUIVOS]

(Ver arquivos disponibilizados)

Passo 4 - Iniciando a Rede nas Organizações:

Host 1 - Máquina hospedeira (Dois peers, um representando a fábrica e outro representando o distribuidor).

Comando: *./minifab up -o fabrica.nerdstore.com.br -e 7050 -s couchdb -n chaincodenerdstore -l node -p '"IniciarLedger"'*

Host 2 - Virtual Machine (Um peer representando o vendedor)

Comando: *./minifab netup -o vendedor.nerdstore.com.br*

Passo 5 - Troca de arquivos entre os hosts para a conexão

5.1 - Para o host 1 reconhecer a entrada do host 2 na rede e permitir a confirmação de transações:

- Copiar o arquivo **JoinRequest_vendedor-nerdstore-com-br.json** do **host 2** presente na pasta **mywork/vars**,
- Renomear para **NewOrgJoinRequest.json**,
- Colocar no **host 1** na pasta **mywork/vars** e executar o comando no **host 1**: *./minifab orgjoin*
- No **host 1** gerar arquivo de configuração do canal: ./minifab channelquery
- Com o arquivo gerado (mychannel_config.json) ajustar rule na seção **Endorsement** e **LifecycleEndorsement** de **"MAJORITY"** para **"ANY"** para permitir qualquer usuário "Admin" das organizações consiga confirmar as transações

- Atualizar as configurações do Canal no host 1: *./minifab channelsign, channelupdate, discover*

5.2 - Para o host 2 "aprender" a se conectar na rede onde ele já foi admitido

- Copiar o arquivo **endpoints.yaml**, presente em **mywork/vars/profiles** do **host 1.**
- Colocar o arquivo no **host 2**, na pasta **mywork/vars**
- Realizar o comando no **host 2**: *./minifab nodeimport,join*
- Realizar o comando no **host 2**: *./minifab discover*
- Realizar o comando no **host 1**: *./minifab approve,discover,commit*
- **Iniciar o Ledger no host 2:** *./minifab install,approve -n chaincodenerdstore -l node -p '"IniciarLedger"'*
- Realizar novamente o comando no **host 1**: *./minifab approve,discover,commit*

Se tudo ocorrer bem, as organizações já estarão na mesma rede blockchain e atuando sobre o mesmo ledger do canal através do chaincode.

Atualizando o Chaincode

1 - Alterar o chaincode e colocar na pasta e alterar a versão no arquivo package.json
2 - Instalar a nova versão: *./minifab install -n chaincodenerdstore -v 1.1.0*
3 - Aprovar e efetivar a instalação *./minifab approve,commit*
4 - Inicializar o chaincode: *./minifab initialize -p '"IniciarLedger"'*
5 - *./minifab discover*

6 - Após ter sido feita a atualização do chaincode no **host 1**, precisamos dar o comando *./minifab discover* nas outras organizações para atualizar o chaincode.

Parando, reiniciando ou excluindo a rede

Parando a rede*: ./minifab down*
Reiniciando a rede: *./minifab netup*
Excluindo a rede em cada host: *./minifab cleanup*
****ATENÇÃO QUANDO EXCLUIR A REDE O CHAINCODE FEITO TAMBÉM SERÁ EXCLUIDO, NÃO ESQUEÇA DE FAZER O BACKUP DA PASTA CHAINCODE ANTES****
Checando status da rede*: ./minifab stats*

Outras Ferramentas

Acompanhar visualmente a rede com Hyperledger Explorer:
- Iniciar a aplicação: *./minifab explorerup*
- Parar a aplicação: *./minifab explorerdown*

****QUANDO A APLICAÇÃO FOR INICIADA SERÁ FORNECIDA O USUÁRIO E A SENHA DE ACESSO NO CONSOLE****

Acessando o banco de dados CouchDB:
Endereço: *http://0.0.0.0:7054/_utils/*
Usuário: *admin*
Senha: *adminpw*

API Web

LInk Base:
https://kctheservant.medium.com/an-implementation-of-api-server-for-hyperledger-fabric-network-8764c79f1a87

0 - Criar a pasta wallets **(API) e colocar o arquivo apiserver.js na pasta**
(Ver arquivos disponibilizados)

1 - Colocar as carteiras na pasta

Origem:/mywork/vars/profiles/vscode/wallets
Destino: API/wallets

2 - Colocar os arquivos de conexão na pasta (API)

Origem: mywork/vars/profiles
mychannel_connection_for_nodesdk.json
mychannel_connection_for_nodesdk.yaml

3 - Iniciar a pasta como um projeto node: **npm init**

4 - instalar módulos:

```
npm install
npm install express body-parser --save
npm install cors
npm install fabric-network
```

5 - Executando Server Web API: **sudo node apiserver.js**

6 - Importar Collection no Postman e fazer chamadas

App Web Vue

(Ver arquivos disponibilizados)

Links
Base para a aplicação:
https://www.creative-tim.com/product/vue-material-dashboard?ref=vuematerial.io

Fontes e Ícones:
https://fonts.google.com

```
Instalação
npm install
npm install vuelidate --save

npm install eslint -g -D
eslint --init
```

Rodando App Web Vue: npm run serve