

J CERCHIARO

PRACTICA HF CERT

1. Despliega la red. Parece que existen algunos contenedores que no se levantan correctamente. Identifícalos y resuelve los problemas asociados.

Se corrigen las líneas 42-91 palabras server y hyperledge

```
42 - ORDERER_ADMIN_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
91 - ./organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/etc/hyperledger/fabric
```

2. Despliega el chaincode “merca-chaincode”. Aunque el proceso de ciclo de vida del chaincode es satisfactorio, parece que este no se despliega correctamente. Identifica el/los problemas y resuélvelos. Documenta este proceso de investigación y resolución.

Para desplegar los chain codes

```
./network.sh up createChannel -ca ./network.sh deployCC -ccn merca-chaincode -ccl javascript -ccv 1.0.0 -ccp ../merca-chaincode
```

Se selecciona el archivo que solo está en RW y se cambia a XRW ejecutable xon el fin de desplegar el chaincode.

```
cerchiaro55@JC-JB208:~/hf-certification-practica/merca-chain$ ./network.sh deployCC -ccn merca-chaincode -ccl javascript -ccv 1.0.0 -ccp ../merca-chaincode
Using docker and docker compose
deploying chaincode on channel 'mychannel'
executing with the following
- CHANNEL_NAME: mychannel
- CC_NAME: merca-chaincode
- CC_SRC_PATH: ../merca-chaincode
- CC_SRC_LANGUAGE: javascript
- CC_VERSION: 1.0.0
- CC_SEQUENCE: auto
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
executing with the following
- CC_NAME: merca-chaincode
- CC_SRC_PATH: ../merca-chaincode
- CC_SRC_LANGUAGE: javascript
- CC_VERSION: 1.0.0
+ '[' false = true ']'
+ peer lifecycle chaincode package merca-chaincode.tar.gz --path ../merca-chaincode --lang node --label merca-chaincode_1.0.0
+ res=0
Chaincode is packaged
```

3. Desde Merca-link nos comentan que los logs de los peers no proporcionan la información deseada. Modifica el logging de la red para que se ajuste a los siguientes requerimientos: - Los logs de los orderers deben estar en formato JSON. - Los logs de

los peers deben de mostrar la fecha y la hora al final. - Dado que estamos en fase de pruebas, los logs de los peers deben de tener un nivel de logging de DEBUG por defecto. Además, los logs asociados al logger “gossip” tendrán un nivel por defecto de WARNING y los del logger de “chaincode” de INFO para no sobrecargar demasiado los logs.

Para realizar esto se toman los cod siguientes

```
# Muestra solo logs con severidad error, panic y fatal. FABRIC_LOGGING_SPEC=error
# Nivel por defecto es warning; logs de msp y gossip son info; logs de chaincode son
error. FABRIC_LOGGING_SPEC=warning:msp,gossip=info:chaincode=error # Nivel por
defecto es debug; logs de policies y orderer son info.
FABRIC_LOGGING_SPEC=debug:policies,orderer=info
```

```
# Muestra los logs en formato json FABRIC_LOGGING_FORMAT=json # Customización
del formato de logging FABRIC_LOGGING_FORMAT="%{color}%{time:2006-01-02
15:04:05.000 MST} [%{module}] %{shortfunc} -> %{level:.4s} %{id:03x}%{color:reset}
%{message}"
```

Al correr de nuevo se puede ver el nuevo formato incluyendo lo solicitado DEB, la fecha la final, etc.

```
[cauthdsl]func2 -> DEBU 1d76 0xc040802110 principal evaluation fails2024-04-20 15:36:11.395 UTC"
[cauthdsl]func1 -> DEBU 1d77 0xc040802110 gate 1713627371394222683 evaluation succeeds2024-04-20 15:36:11.395 UTC"
[policies]EvaluateSignedData -> DEBU 1d78 Signature set satisfies policy /Channel/Application/Org1MSP/Readers2024-04-20 15:36:11.395 UTC"
[policies]EvaluateSignedData -> DEBU 1d79 == Done Evaluating *cauthdsl.policy Policy /Channel/Application/Org1MSP/Readers2024-04-20 15:36:11.395 UTC"
[policies]EvaluateSignedData -> DEBU 1d7a Signature set satisfies policy /Channel/Application/Readers2024-04-20 15:36:11.395 UTC"
[policies]EvaluateSignedData -> DEBU 1d7b == Done Evaluating *policies.ImplicitMetaPolicy Policy /Channel/Application/Readers2024-04-20 15:36:11.395 UTC"
[msp]DeserializeIdentity -> DEBU 1d7c Obtaining identity2024-04-20 15:36:11.395 UTC"
[msp]DeserializeIdentity -> DEBU 1d7d Obtaining identity2024-04-20 15:36:11.396 UTC"
[msp.identity]Sign -> DEBU 1d7e Sign: plaintext: 180124f0a3f0a1b70656572302e6f72...67EC120C08A28AA1F6C5DA81E4171064 2024-04-20 15:36:11.499 UTC"
[msp.identity]Sign -> DEBU 1d7f Sign: digest: 2431c7a70609a9f8009ca128aa3000a390bf8045df253cd8360f52b14b5a2818 2024-04-20 15:36:11.499 UTC"
[msp.identity]Sign -> DEBU 1d80 Sign: plaintext: 0a1b70656572302e6f7267322e6578616d706c6552e636f603a39303531 2024-04-20 15:36:11.500 UTC"
[msp.identity]Sign -> DEBU 1d81 Sign: digest: 3ECDAC835A286E7CA759C102358209D7A1071384A591E6A373D0B850181CBDCF 2024-04-20 15:36:11.500 UTC"
[msp]DeserializeIdentity -> DEBU 1d82 Obtaining identity2024-04-20 15:36:11.501 UTC"
[msp.identity]Verify -> DEBU 1d83 Verify: signer identity (certificate subject=CN=peer0,OU=peer0,OU=Hyperledger,ST=North Carolina,C=US issuer=CN=fabric-ca-server,OU=Fabric,OU=Hyperledger,S
=North Carolina,C=US serialnumber=35024622930605249097309175637921707568046215127)2024-04-20 15:36:11.506 UTC"
[msp]DeserializeIdentity -> DEBU 1d84 Obtaining identity2024-04-20 15:36:12.361 UTC"
[peer.gossip.mcs]getValidatedIdentity -> DEBU 1d85 Validation succeeded {"CN":"peer0","Issuer-CN":"fabric-ca-server","Issuer-L-ST-C":["[]-[US]","Issuer-OU":["Fabric"],"L-ST-C":["[]-[US]","MSP":"Org1MSP","OU":["peer"]}] on [mychannel]2024-04-20 15:36:12.362 UTC"
```

4. Los técnicos de Merca-link nos muestran su preocupación acerca de las claves criptográficas, que se almacenan en texto plano dentro de los directorios de la red. Como una primera medida de securización de las claves y de las operaciones criptográficas sugieren el acoplamiento de un HSM a las CAs, utilizándose softshsm2 para esta fase de pruebas y configurando un token diferente para cada CA.

Descargas

```
sudo apt install softshsm2
```

Tokens

```
softhsm2-util --init-token --slot <slot_number> --label <token_label> --pin <pin> --so-pin <so_pin>
```

softhm2

```
! compose-test-net.yaml M X JS assetTransfer.js M
merca-chain > compose > ! compose-test-net.yaml

130 - ../organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/etc/hyperledger/fabric
131 - peer0.org2.example.com:/var/hyperledger/production
132 working_dir: /root
133 command: peer node start
134 ports:
135 - 9051:9051
136 - 9445:9445
137 networks:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS sudo - merca-chain + v

GNU nano 6.2 /etc/softhsm/softhsm2.conf
# SoftHSM v2 configuration file

directories.tokendir = /var/lib/softhsm/tokens/
objectstore.backend = file

# ERROR, WARNING, INFO, DEBUG
log.level = ERROR

# If CKF_REMOVABLE_DEVICE flag should be set
slots.removable = false

# Enable and disable PKCS#11 mechanisms using slots.mechanisms.
slots.mechanisms = ALL

# If the library should reset the state on fork
library.reset_on_fork = false

cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ pwd
/home/cerchiaro55/hf-certification-practica/merca-chain
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ ^C
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$

GNU nano 6.2 /etc/softhsm/softhsm2.conf
# SoftHSM v2 configuration file

directories.tokendir = /home/cerchiaro55/hf-certification-practica/merca-chain/tokens/
objectstore.backend = file

# ERROR, WARNING, INFO, DEBUG
log.level = ERROR

# If CKF_REMOVABLE_DEVICE flag should be set
slots.removable = false

# Enable and disable PKCS#11 mechanisms using slots.mechanisms.
slots.mechanisms = ALL

# If the library should reset the state on fork
library.reset_on_fork = false

cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ sudo nano /etc/softhsm/softhsm2.conf
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ sudo nano /etc/softhsm/softhsm2.conf
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ sudo nano /etc/softhsm/softhsm2.conf
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ export SOFTSM2_CONF=/etc/softhsm2.conf
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ softhsm2-util --init-token --label CA1_token --pin PIN_CA1 --so-pin SO_PIN_CA1
ERROR: Could not load the SoftHSM configuration.
ERROR: Please verify that the SoftHSM configuration is correct.
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ sudo !!
sudo softhsm2-util --init-token --label CA1_token --pin PIN_CA1 --so-pin SO_PIN_CA1
ERROR: Failed to enumerate object store in /home/cerchiaro55/hf-certification-practica/merca-chain/tokens/
ERROR: Please verify that the SoftHSM configuration is correct.
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ mkdir tokens
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ sudo softhsm2-util --init-token --label CA1_token --pin PIN_CA1 --so-pin SO_PIN_CA1
ERROR: A slot/token must be supplied. Use --slot <number>, --serial <serial>, --token <label>, or --free
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ sudo softhsm2-util --init-token --label CA1_token --pin PIN_CA1 --so-pin SO_PIN_CA1 --slot 0
The token has been initialized and is reassigned to slot 1610545195
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$
```

```
cerchiaro55@JC-JB20B:~/hf-certification-practica/merca-chain$ sudo !!
sudo softhsm2-util --show-slots
Available slots:
Slot 1610545195
  Slot info:
    Description:      SoftHSM slot ID 0x5ffef82b

    Manufacturer ID:  SoftHSM project
    Hardware version:  2.6
    Firmware version:  2.6
    Token present:     yes
  Token info:
    Manufacturer ID:  SoftHSM project
    Model:            SoftHSM v2
    Hardware version:  2.6
    Firmware version:  2.6
    Serial number:     cealcf6bdfef82b
    Initialized:       yes
    User PIN init.:    yes
    Label:             CA1_token
Slot 1711334517
  Slot info:
    Description:      SoftHSM slot ID 0x6600e475

    Manufacturer ID:  SoftHSM project
    Hardware version:  2.6
    Firmware version:  2.6
    Token present:     yes
  Token info:
    Manufacturer ID:  SoftHSM project
    Model:            SoftHSM v2
    Hardware version:  2.6
    Firmware version:  2.6
    Serial number:     230b9044e600e475
    Initialized:       yes
    User PIN init.:    yes
    Label:             CA2_token
```

```
Manufacturer ID: SoftHSM project
Hardware version: 2.6
Firmware version: 2.6
Token present: yes
Token info:
Manufacturer ID: SoftHSM project
Model: SoftHSM v2
Hardware version: 2.6
Firmware version: 2.6
Serial number: 230b9044e600e475
Initialized: yes
User PIN init.: yes
Label: CA2_token
Slot 1903152988
Slot info:
Description: SoftHSM slot ID 0x716fcf5c

Manufacturer ID: SoftHSM project
Hardware version: 2.6
Firmware version: 2.6
Token present: yes
Token info:
Manufacturer ID: SoftHSM project
Model: SoftHSM v2
Hardware version: 2.6
Firmware version: 2.6
Serial number: 85a7e767f16fcf5c
Initialized: yes
User PIN init.: yes
Label: CA_OR_token
Slot 3
Slot info:
Description: SoftHSM slot ID 0x3

Manufacturer ID: SoftHSM project
Hardware version: 2.6
Firmware version: 2.6
Token present: yes
Token info:
Manufacturer ID: SoftHSM project
Model: SoftHSM v2
Hardware version: 2.6
Firmware version: 2.6
Serial number:
Initialized: no
User PIN init.: no
Label:
```



```
bric-ca-server-config.yaml .../org1 ! fabric-ca-server-config.yaml .../org2 ! fabric-ca-server-config.yaml .../ordererOrg Dockerfile ! compose-ca.yaml .gitignore x JS assetTransfer.js ...

merca-chain > @ .gitignore
1 /channel-artifacts/*.tx
2 /channel-artifacts/*.block
3 /ledgers
4 /ledgers-backup
5 /channel-artifacts/*.json
6 /org3-artifacts/crypto-config/*
7 organizations/fabric-ca/ordererOrg/*
8 organizations/fabric-ca/org1/*
9 organizations/fabric-ca/org2/*
10 organizations/ordererOrganizations/*
11 organizations/peerOrganizations/*
12 system-genesis-block/*
13 *.tar.gz
14 log.txt
15 bin/

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash-merca-chain + v [] ... ^ x

2024-04-21 18:30:20.184 UTC 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
2024-04-21 18:30:20.208 UTC 0002 INFO [channelCmd] update -> Successfully submitted channel update
Anchor peer set for org 'Org2MSP' on channel 'mychannel'
Channel 'mychannel' joined
cerchiaro55@JC-JB208:~/hf-certification-practica/merca-chain$ sudo ../bin/fabric-ca-client enroll -u https://admin:adminpw@localhost:7054 --caname ca-org1 --tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
2024/04/21 13:31:22 [INFO] TLS Enabled
2024/04/21 13:31:22 [INFO] generating key: &{A:ecdsa S:256}
2024/04/21 13:31:22 [INFO] encoded CSR
2024/04/21 13:31:22 [INFO] Stored client certificate at /root/.fabric-ca-client/msp/signcerts/cert.pem
2024/04/21 13:31:22 [INFO] Stored root CA certificate at /root/.fabric-ca-client/msp/cacerts/localhost-7054-ca-org1.pem
2024/04/21 13:31:22 [INFO] Stored Issuer public key at /root/.fabric-ca-client/msp/IssuerPublicKey
2024/04/21 13:31:22 [INFO] Stored Issuer revocation public key at /root/.fabric-ca-client/msp/IssuerRevocationPublicKey
cerchiaro55@JC-JB208:~/hf-certification-practica/merca-chain$ sudo ../bin/fabric-ca-client register -d -u https://localhost:7054 --caname ca-org1 --id.name merca-admin --id.secret merka-12345 --id.type client --id.attrs "hf.Registrar.Roles=peer,client" --id.affiliation org1.department1 --tls.certfiles "${PWD}/organizations/fabric-ca/org1/ca-cert.pem"
2024/04/21 13:32:07 [DEBUG] Set log level:
2024/04/21 13:32:07 [DEBUG] Home directory: /root/.fabric-ca-client
2024/04/21 13:32:07 [INFO] Configuration file location: /root/.fabric-ca-client/fabric-ca-client-config.yaml
2024/04/21 13:32:07 [DEBUG] Checking for enrollment
2024/04/21 13:32:07 [DEBUG] Initializing client with config: &{URL:https://localhost:7054 MSPDir:msp TLS:{Enabled:true CertFiles:[/home/cerchiaro55/hf-certification-practica/merca-chain/organizations/fabric-ca/org1/ca-cert.pem] Client:{Keyfile: CertFiles:} Enrollment:{ Name: Secret:**** CAName: AttrReqs:[] Profile: Label: CSR:<nil> Type:x509 } CSR:{CN:admin Names:[([C:US ST:North Carolina L: 0:Hyperledger OU:Fabric SerialNumbers:]] Hosts:[JC-JB208] KeyRequest:0xc00053a400 CA:<nil> SerialNumber:} ID:(Name:merca-admin Type:client Secret:merka-12345 MaxEnrollments:0 Affiliation:org1.department1 Attributes:{Name:hf.Registrar.Roles Value:peer,client ECert:false} CAName:} Revoke:{Name: Serial: AKI: Reason: CAName: GenCRL:false} CAInfo:{CAName:} CAName:ca-org1 CSP:0xc00053a440 Debug:true LogLevel: Idemix:{Curve:amcl.Fp256bn}}
2024/04/21 13:32:07 [DEBUG] Initializing BCCSP: &{Default:SW SW:0xc00053a460 PKCS11:<nil>}
2024/04/21 13:32:07 [DEBUG] Initializing BCCSP with software options &{Security:256 Hash:SHA2 FileKeystore:0xc000225290}
2024/04/21 13:32:07 [INFO] TLS Enabled
2024/04/21 13:32:07 [DEBUG] CA Files: [/home/cerchiaro55/hf-certification-practica/merca-chain/organizations/fabric-ca/org1/ca-cert.pem]
2024/04/21 13:32:07 [DEBUG] Client Cert File:
2024/04/21 13:32:07 [DEBUG] Client Key File:
2024/04/21 13:32:07 [DEBUG] Client TLS certificate and/or key file not provided
2024/04/21 13:32:07 [DEBUG] Using curve amcl.Fp256bn for Idemix
2024/04/21 13:32:07 [DEBUG] CheckIdemixEnrollment - ipkFile: /root/.fabric-ca-client/msp/IssuerPublicKey, idemixCredFile: /root/.fabric-ca-client/msp/user/SignerConfig
2024/04/21 13:32:07 [DEBUG] Client configuration settings: &{URL:https://localhost:7054 MSPDir:/root/.fabric-ca-client/msp TLS:{Enabled:true CertFiles:[/home/cerchiaro55/hf-
```

6. Los merca-ingenieros de Merca-link consideran que tener herramientas de monitorización a su disposición agilizaría estas tareas de administración en futuras ocasiones. Despliega sendas instancias de Prometheus y Grafana para satisfacer estas necesidades.

A continuación se despliegan grafana y prometheus se entra al folder y se ejecuta el comando

```
cerchiaro55@JC-JB208:~/hf-certification-practica/merca-chain$ cd prometheus-grafana/
cerchiaro55@JC-JB208:~/hf-certification-practica/merca-chain/prometheus-grafana$ docker compose up -d
WARN[0000] /home/cerchiaro55/hf-certification-practica/merca-chain/prometheus-grafana/docker-compose.yaml: 'version' is obsolete
[+] Running 32/7
  ✓ prometheus Pulled
  ✓ grafana Pulled
  ✓ cadvisor Pulled
  ✓ node-exporter Pulled

[+] Running 6/6
  ✓ Volume "prometheus-grafana_grafana_storage" Created
  ✓ Volume "prometheus-grafana_prometheus_data" Created
  ✓ Container prometheus Started
  ✓ Container cadvisor Started
  ✓ Container node-exporter Started
  ✓ Container grafana Started
```


7. Investiga sobre los despliegues distribuidos (esto es, en distintos nodos) de Hyperledger Fabric, y resume en unas pocas líneas tus recomendaciones y los principales puntos a tener en cuenta para que Merka-link despliegue su red de forma distribuida. No olvides referencias los artículos/blogs/páginas web que has consultado.Despliegues distribuidos

El despliegue de nodos distribuidos implica la configuración y conexión de múltiples nodos de manera geográficamente dispersa. Esto se puede lograr mediante tecnologías como contenedores y orquestadores, como Kubernetes, que permiten la gestión centralizada de los nodos y facilitan la escalabilidad y la alta disponibilidad de las aplicaciones distribuidas. Además, se pueden emplear herramientas de automatización para simplificar el proceso de implementación y mantenimiento de la infraestructura distribuida.

```
# docker-compose.yml

version: '3'

services:
  node1:
    image: tu_imagen_de_aplicacion
    ports:
      - "8001:8000"
    networks:
      - app-network

  node2:
    image: tu_imagen_de_aplicacion
    ports:
      - "8002:8000"
    networks:
      - app-network

networks:
  app-network:
    driver: bridge
```

Este archivo de Docker Compose define dos servicios, `node1` y `node2`, cada uno utilizando la misma imagen de aplicación pero exponiendo puertos diferentes. Puedes ejecutar este archivo usando el comando `docker-compose up -d` para lanzar los nodos distribuidos.

Este es un ejemplo básico y puedes personalizarlo según tus necesidades, como agregar más nodos, configurar volúmenes compartidos, o incluir un orquestador como Kubernetes para una gestión más avanzada.

https://hyperledger-fabric.readthedocs.io/es/latest/cc_service.html?highlight=distributed%20deployments#chaincode-as-an-external-service

https://hyperledger-fabric.readthedocs.io/es/latest/cc_service.html?highlight=distributed%20deployments#chaincode-as-an-external-service