10/19/23, 8:47 AM Blueventure: Blockchain Lab

Blueventure: Blockchain Lab

X

Track-and-Trace Blockchain Workshop for Hyperledger Fabric 2.2 (BETA)

- Create a Hyperledger Fabric Network
 - Create Network & Member
 - Accept invite and create
 Supplier member
 - Congratulations
- Setup Development Environment
- ▶ Set up a Fabric client
- ▼ Write and deploy chaincode

Chaincode development environment

Write chaincode

Create sharing policy

AWS account access

Open AWS console (us-east-1)

Get AWS CLI credentials

Exit event

Event dashboard > Write and deploy chaincode > Approve and commit the chaincode

Approve and commit the chaincode

(a) Both the **Retailer** and **Supplier** should run the approval command in its Cloud9 terminal to approve the chaincode. This step needs to be performed by all channel members.

```
1 peer lifecycle chaincode approveformyorg -o $ORDERER --channelID mainchannel --name supplychaincc --version 1.0 --sequence 1 --initial equ
```

The output should look like:

```
1 2022-02-22 03:43:32.149 UTC [chaincodeCmd] ClientWait -> INFO 001 txid [c758022ca70255d1140699159636495fb08a55a78dd1331221d1449740 35343
```

To verify that the chaincode has been approved by all members and is ready for its final commit to the channel, run the following command from either member's terminal:

```
1 peer lifecycle chaincode checkcommitreadiness -o $ORDERER --channelID mainchannel --name supplychaincc --version 1.0 --init-required --s
```

The output should look like:

```
1 {
2     "approvals": {
3         "m-ILUWOSUGQZEBHGRMZCKABK73M4": true,
4         "m-YZEU53CKNNFINNKAQGECHNHLRE": true
5     }
6 }
```

(a) After all members have approved the chaincode definition, *Either* **Retailer** *or* **Supplier** can run the commit command in its Cloud9 terminal to officially deploy the chaincode for use in the channel. This step only needs to be performed by one channel member.

Next, to make the chaincode operable for each member who installed and approved the chaincode in the channel, one member should commit the chaincode using the following command:

(1)



Track-and-Trace Blockchain Workshop for Hyperledger Fabric 2.2 (BETA) X

- Create a Hyperledger Fabric Network
 - Create Network & Member
 - Accept invite and create Supplier member
 - Congratulations
- Setup Development Environment
- ▶ Set up a Fabric client
- ▼ Write and deploy chaincode

Chaincode development environment

Write chaincode

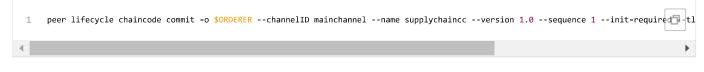
Create sharing policy

AWS account access

Open AWS console (us-east-1)

Get AWS CLI credentials

Exit event



The output should look like:

```
1 2022-02-22 03:44:59.223 UTC [chaincodeCmd] ClientWait -> INFO 001 txid [cdb77e0e14741ec052075e99ee06ce5c82699ce24c6c907cc1c130ebe0atpeb7
```

Finally, because this chaincode uses the low-level Fabric Chaincode Shim API and initatializes a product ID list in state during the Init() process, we must *initalize* the chaincode before using it. Note that this is not required for every chaincode package, particularly those that use the higher-level Fabric Contract API. Notice that the use of --init-required in the prior commands was used to remind us later that initalization is required, and will prevent usage of the chaincode with invoke() until init() is complete. It is a best practice to use the --init-required flag during the approve and commit steps whenever initialization is required to use the chaincode for the first time.

Our chaincode requires initiatlization, which can be achieved by invoking the chaincode with the flag --isInit for the first time, after which other methods in the chaincode can be invoked in the next section:

