

Blueventure: Blockchain Lab

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)

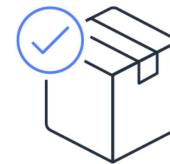
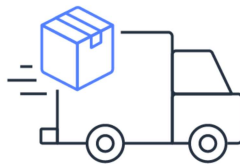
[Get AWS CLI credentials](#)

Exit event

[Event dashboard](#) > [Write and deploy chaincode](#) > **Create member identities**

Create member identities

One additional step will be to create certificates for each member's staff. These certificates will have custom attributes on them containing a list of operations that the staff member is allowed to perform. In this section, each member will create certificates used to verify the identities of each of its personnel and track the operations they perform.



Supplier

worker - manufactures and ships products to retailer

inspector - inspects products after fabrication



Retailer

worker - receives and labels products from supplier

seller - sells labeled products to customers



Both the **Retailer** and **Supplier** should copy the following commands into the Cloud9 terminal *before* moving on to subsequent steps.

The following commands will create a unique randomly-generated password for each user and store that password in AWS Secrets Manager for future use:

```
1 WORKER1_PASSWORD=$(aws secretsmanager get-random-password --exclude-punctuation | jq -r ".RandomPassword")
2
3 WORKER2_PASSWORD=$(aws secretsmanager get-random-password --exclude-punctuation | jq -r ".RandomPassword")
4
5
6 aws secretsmanager create-secret --name="HLF-MEMBER-PW-NETWORK-${NETWORKID}-ACCOUNT-${WORKER1_NAME}" --secret-string=$WORKER1_PASSWORD
7
8
```

Blueventure: Blockchain Lab

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

```
aws secretsmanager create-secret --name="HLF-MEMBER-PW-NETWORK-${NETWORKID}-ACCOUNT-${WORKER2_NAME}" --secret-string=$WORKER2_PASSWORD
```

The following commands create certificates for various personnel / roles, and then copies the certificate public keys for these identities to an S3 bucket where the other members can download them.

Create the first worker role:

```
1 # create worker 1 cert
2 cd
3 fabric-ca-client register -u https://$CASERVICEENDPOINT --id.name $WORKER1_NAME --id.affiliation $MEMBER_NAME --tls.certfiles $HOME/managedblockchain-tls-chain.
4 fabric-ca-client enroll -u https://$WORKER1_NAME:$WORKER1_PASSWORD@$CASERVICEENDPOINT --tls.certfiles $HOME/managedblockchain-tls-chain.
5 cp -r admin-msp/admincerts/ $WORKER1_NAME-msp
```

Create the second worker role:

```
1 # create worker 2 cert
2 fabric-ca-client register -u https://$CASERVICEENDPOINT --id.name $WORKER2_NAME --id.affiliation $MEMBER_NAME --tls.certfiles $HOME/managedblockchain-tls-chain.
3 fabric-ca-client enroll -u https://$WORKER2_NAME:$WORKER2_PASSWORD@$CASERVICEENDPOINT --tls.certfiles $HOME/managedblockchain-tls-chain.
4 cp -r admin-msp/admincerts/ $WORKER2_NAME-msp
```

Upload the certificates to S3 where they will be accessible via the sharing policy:

```
1 # upload admin certs to S3 bucket
2 export cacert=$(ls $HOME/admin-msp/cacerts/ca-*.pem)
3 aws s3api put-object --bucket $BUCKET_NAME --key ${MEMBER_ABBREVIATION}cacert.pem --body $cacert --acl bucket-owner-full-control
4 aws s3api put-object --bucket $BUCKET_NAME --key ${MEMBER_ABBREVIATION}admincert.pem --body $HOME/admin-msp/admincerts/cert.pem --acl bucket-owner-full-control
```

Previous

Next