

**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

▼ **Invoking chaincode via API**

Deploy CDK application

Store user secrets

Create Coanito users

▼ **AWS account access**

[Open AWS console](#)

(us-east-1)

[Get AWS CLI credentials](#)

Exit event

[Event dashboard](#) > Invoking chaincode via API

Invoking chaincode via API

You must have completed the prior modules for the commands in this one to work properly, or your account must have been provisioned using our project automation scripts.

Many newcomers to blockchain technology experience a learning curve. Smart contracts and the constraints they place on development can take some getting used to. While most business logic that is shared by the various consortium members should be handled at the smart contract (chaincode) level, other layers of abstraction are usually needed to support modern applications. These layers can also ease the learning curve by allowing developers to spend most of their time interacting with systems they are more familiar with.

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

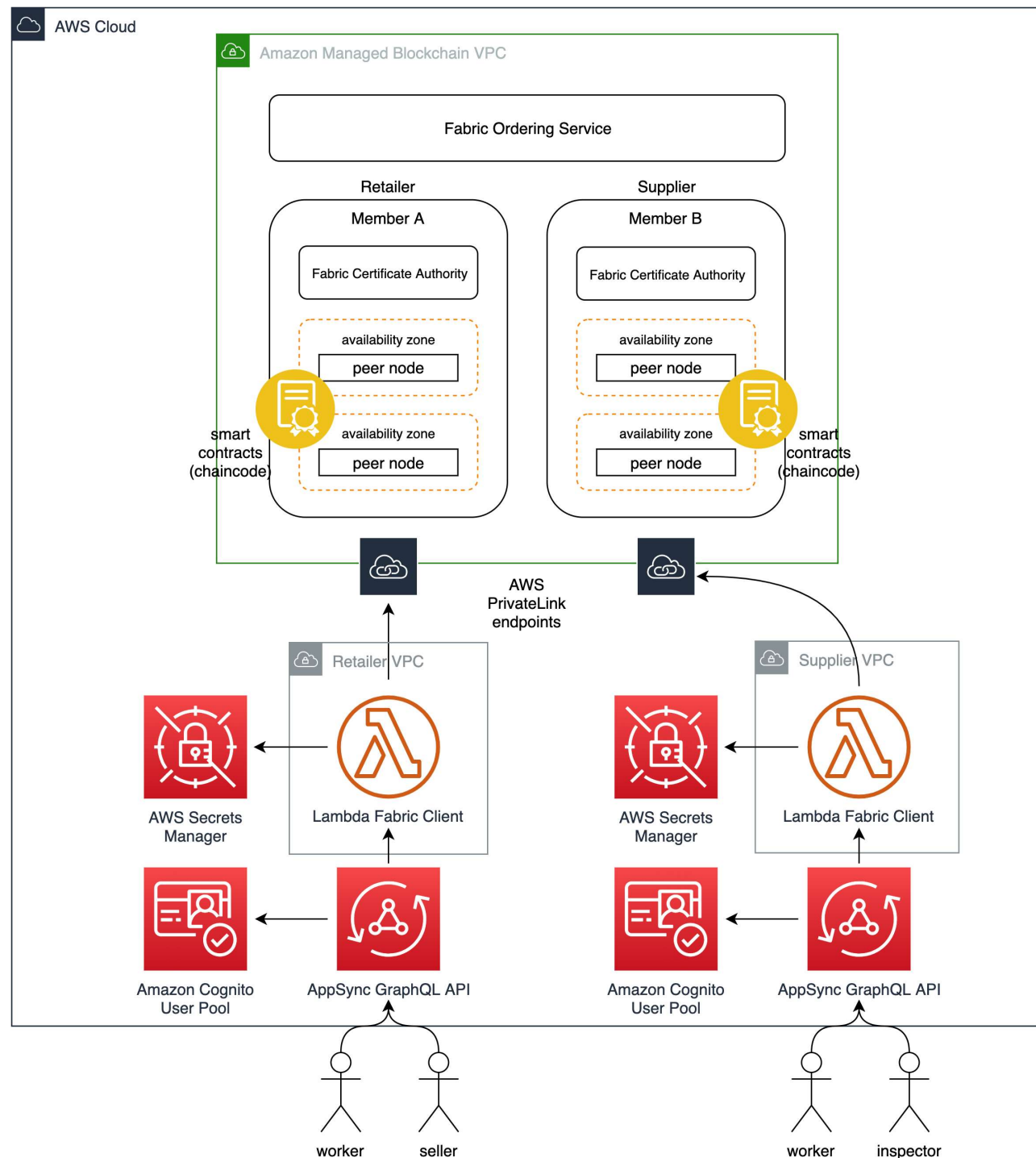
▼ Invoking chaincode via API

- Deploy CDK application
- Store user secrets
- Create Coanito users

▼ AWS account access

- [Open AWS console \(us-east-1\)](#)
- [Get AWS CLI credentials](#)

Exit event



**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

▼ **Invoking chaincode via API**

Deploy CDK application

Store user secrets

Create Coanito users

▼ **AWS account access**

[Open AWS console
\(us-east-1\)](#)

[Get AWS CLI credentials](#)

Exit event

In this module, you will begin interacting with your chaincode from an [AWS Lambda](#) function. This will reduce the amount of infrastructure that needs to be maintained to support your decentralized application, and will also help it to scale to handle heavier transaction volumes.

In front of the Lambda function is an [AWS AppSync](#) GraphQL API. This API authenticates users calling it against an [Amazon Cognito](#) User Pool. The Lambda function trusts that the usernames it receives from AppSync have been properly authenticated, and is authorized to look up secrets belonging to the user in [AWS Secrets Manager](#). These secrets contain the keys used to sign Fabric transactions on the user's behalf. The Lambda function uses these secrets to query and invoke chaincode in the blockchain network on behalf of the calling user.

AppSync authentication against Amazon Cognito is used to translate application-level user identities to chaincode-level user identities, ensuring that the trust model is unbroken from the application to the base layer. An additional benefit of AppSync integration is that developers can use a familiar paradigm (GraphQL queries) to interact with and be notified of updates in the blockchain.

Each consortium member should execute all of the commands in this module in their entirety.

[Previous](#)[Next](#)