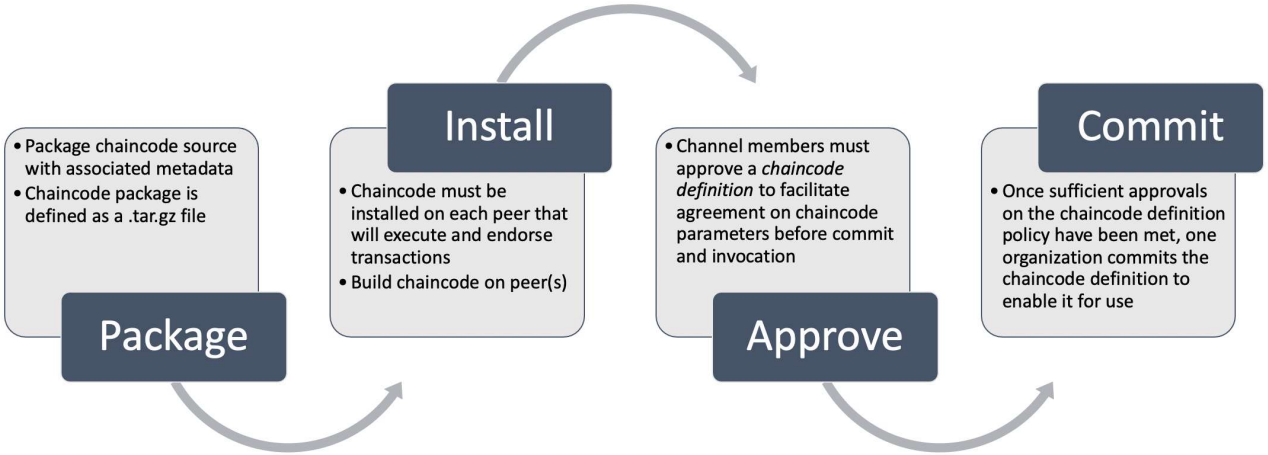
Congratulations

You've now successfully completed the second module, in which you set up your own Fabric client! In the next module, you will create a channel in which the Retailer and Supplier members will interact. Then, you will build and deploy Chaincode to your Hyperledger Fabric network.

Understanding the Fabric Chaincode Lifecycle
Before deploying chaincode to our test network, it's critical to understand *how* chaincode is managed on Hyperledger Fabric networks!

In Hyperledger Fabric, Chaincode is the mechanism by which business logic can be enforced on transactions on the network and those transactions' effect on ledger state. Chaincode can be described as an application, or bundle of source code, written in Go, JavaScript (Node.js), or Java, which is deployed by mutual agreement amongst participants in a Hyperledger Fabric network channel. Due to the decentralized nature of the blockchain network and its channels therein, the agreement between channel members on the definition of a given chaincode package and the subsequent deployment or upgrade of that chaincode requires a decentralized governance process: **The Fabric Chaincode Lifecycle**.

The Chaincode Lifecycle, as described in the Hyperledger Fabric Documentation, is "a process that allows multiple organizations to agree on how a chaincode will be operated before it can be used on a channel". The process is divided into four distinct phases; 1) package, 2) install, 3) approve, 4) commit,where upon commitment the chaincode is available to be used by members of a channel.



Now that you understand the process behind the Chaincode Lifecycle in Hyperledger Fabric 2.2, let's deploy a sample chaincode package written in *NodeJS* to our recently created channel!

To put this knowledge into practice, move on to the next section!

Previous Next