**Chaincode in Golang**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | GitHub URL | https://github.com/zs-papandas/serialization | | Username | zs-papandas | | Password | pd!19385 | |

# GET STATED WITH CHAINCODE

## Setup Project Enviroment

Navigate to this folder.

(NOTE: Fabric Sample could be found here in this link https://github.com/hyperledger/fabric-samples)

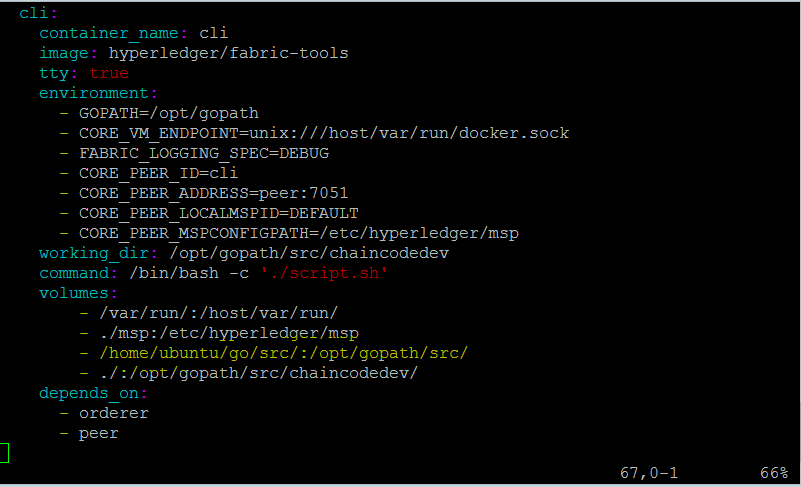
|  |
| --- |
| cd ~/fabric-tools/fabric-samples/chaincode-docker-devmode |

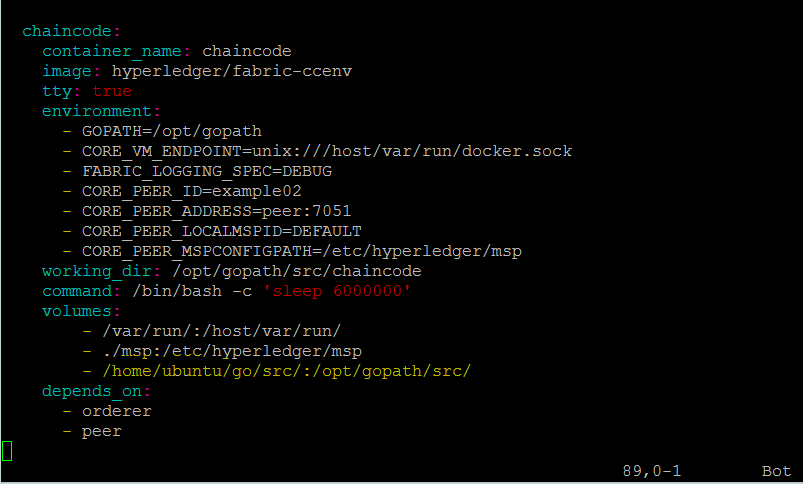
Make a copy of the yaml file

|  |
| --- |
| cp docker-compose-simple.yaml docker-compose-simple.yaml.bak |

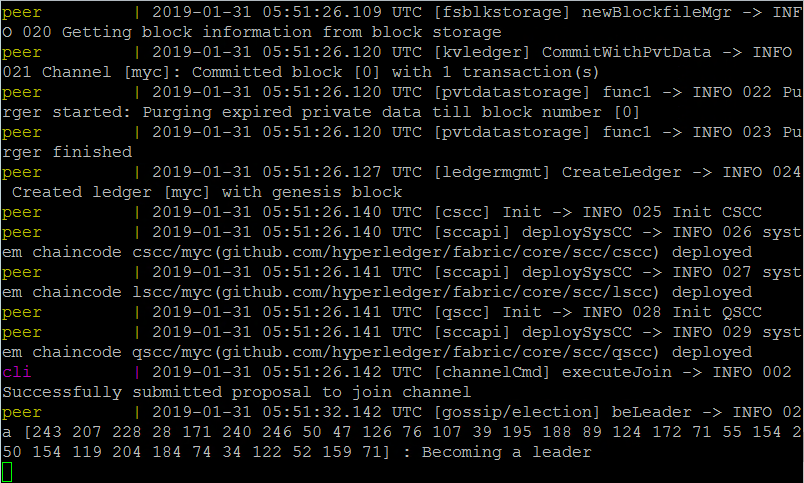
Edit the yaml file

|  |
| --- |
| vim docker-compose-simple.yaml |





|  |
| --- |
| docker-compose -f docker-compose-simple.yaml up |



## Set up Chaincode console window

Open a new terminal

|  |
| --- |
| cp ~/go/src/github.com/zs-papandas/serialization |



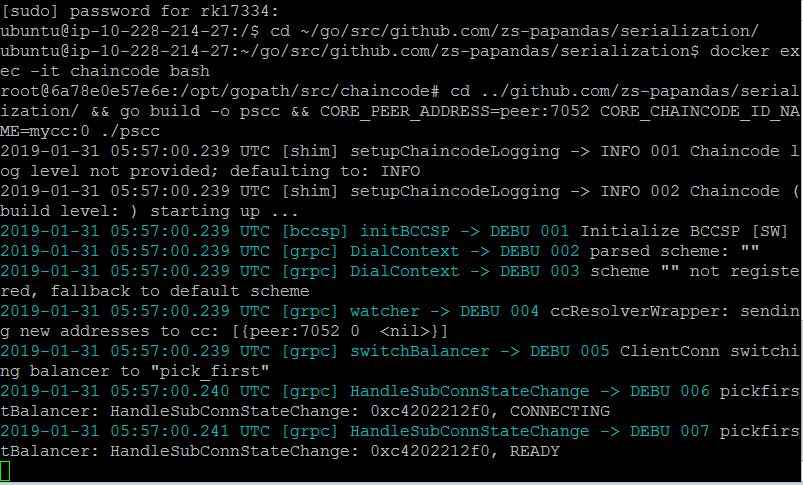
Update the Git repo

|  |
| --- |
| git pull |

Open the Chaincode Container in interactive mode.

|  |
| --- |
| docker exec -it chaincode bash |

|  |
| --- |
| cd ../github.com/zs-papandas/serialization/ && go build -o pscc && CORE\_PEER\_ADDRESS=peer:7052 CORE\_CHAINCODE\_ID\_NAME=mycc:0 ./pscc |



## Set up CLI window

Open a new terminal

|  |
| --- |
| docker exec -it cli bash |

|  |
| --- |
| peer chaincode install -p github.com/zs-papandas/serialization -n mycc -v 0  peer chaincode instantiate -n mycc -v 0 -c '{"Args":[]}' -C myc |

# API REFERANCE

|  |
| --- |
| Add new user |
| |  |  | | --- | --- | | method | createAccount | |  | invoke | | parameters | Firstname string  Lastname string  DOB string  Email string  Mobile string  Company string  UserType string (manufacturer, wholesaler, retailer, patient) | | example | peer chaincode invoke -n mycc -c '{"Args":["createAccount", "a", "Manufacturer","John","1987-11-14","manufacturer@zs.com","9641443962","ZS Associates India Pvt Ltd","manufacturer"]}' -C myc | |

|  |
| --- |
| Add new product |
| |  |  | | --- | --- | | method | createProduct | |  | invoke | | parameters | Owner string (Unique User ID)  Name string  Expire string  GTIN string  LotNumber string  Amount string  TotalQty string  ProductType string (pallet, box, packet, item)  ParentProduct string (empty if Pallet) | | example | peer chaincode invoke -n mycc -c '{"Args":["createProduct", "a", "CROSINpallet","expire","gtin-102030","lotnum/23/45/as","1000","2","pallet"," XXXXXXXXXXXXXXX"]}' -C myc | |

|  |
| --- |
| Change ownership |
| |  |  | | --- | --- | | method | changeOwner | |  | invoke | | parameters | SerialId string  Owner string (Unique User ID) | | example | peer chaincode invoke -n mycc -c '{"Args":["changeOwner", "XXXXXXXXXXXXXX", "c"]}' -C myc | |

|  |
| --- |
| Get a single product detail |
| |  |  | | --- | --- | | method | retrieveProduct | |  | invoke | | parameters | SerialId string | | example | peer chaincode invoke -n mycc -c '{"Args":["retrieveProduct", "XXXXXXXXXXXXXX"]}' -C myc | |

|  |
| --- |
| Get product history |
| |  |  | | --- | --- | | method | listProductHistory | |  | invoke | | parameters | SerialId string | | example | peer chaincode invoke -n mycc -c '{"Args":["listProductHistory", "XXXXXXXXXXXXXX"]}' -C myc | |

|  |
| --- |
| Auto generate product list |
| |  |  | | --- | --- | | method | retrieveProduct | |  | invoke | | parameters | Owner string (Unique User ID)  Name string  Expire string  GTIN string  LotNumber string  Amount string  Pallet string  Box string  Packet string  Item string | | example | peer chaincode invoke -n mycc -c '{"Args":["generateProduct", "a", "CROSINpallet","expire","gtin-102030","lotnum/23/45/as","1000","2","2","2","2"]}' -C myc | |