智能合约部署

智能合约是用程序实现合约的内容，并且这个程序是事件驱动、有状态的。智能合约是早就出现的概念，早于区块链提出。但是，区块链的出现为智能合约的实现提供了一个非常理想的环境。而在Fabric中，Chaincode就是开发者实现智能合约的方式。

我们知道blockchain系统是一个网络，由若干结点构成。Fabric区块链系统也不例外，而要运行Chaincode程序，就要把它首先部署到fabric系统的结点上。也就是说，Chaincode程序是依赖于fabric系统结点的。在区块链部署这一板块中已经详细的介绍了fabric系统部署，在此就不再赘述。

1.进入peer节点

docker exec -it peer0.org1.example.com bash

2.进入需要编译的智能合约的目录，如下示例：

cd /opt/gopath/src/github.com/hyperledger/fabric/examples/chaincode/go/marbles02

3.进行编译

go build //编译代码，生成marbles\_chaincode可执行程序。编译成功将不显示任何信息，失败会显示失败信息

4.发布智能合约

CORE\_CHAINCODE\_ID\_NAME=mycc CORE\_PEER\_ADDRESS=10.139.12.43:7051 ./marbles\_chaincode

//发布chaincode，并为其注册一个名称CORE\_CHAINCODE\_ID\_NAME。如果发布成功会在打印信息中显示[shim] DEBU : Received REGISTERED, ready for invocations

5.调用智能合约

peer chaincode instantiate -n mycc -C mychannel -c '{"Args": []}' -o orderer.example.com:7050 -P "OR ('Org1MSP.member','Org2MSP.member')"

//实例化智能合约

peer chaincode invoke -o orderer.example.com:7050 -C mychannel -n mycc -c '{"Args":["initMarble","marble1","blue","35","tom"]}'

//调用智能合约

peer chaincode query -o orderer.example.com:7050 -C mychannel -n mycc -c '{"Args":["readMarble","marble1"]}'

//查询智能合约