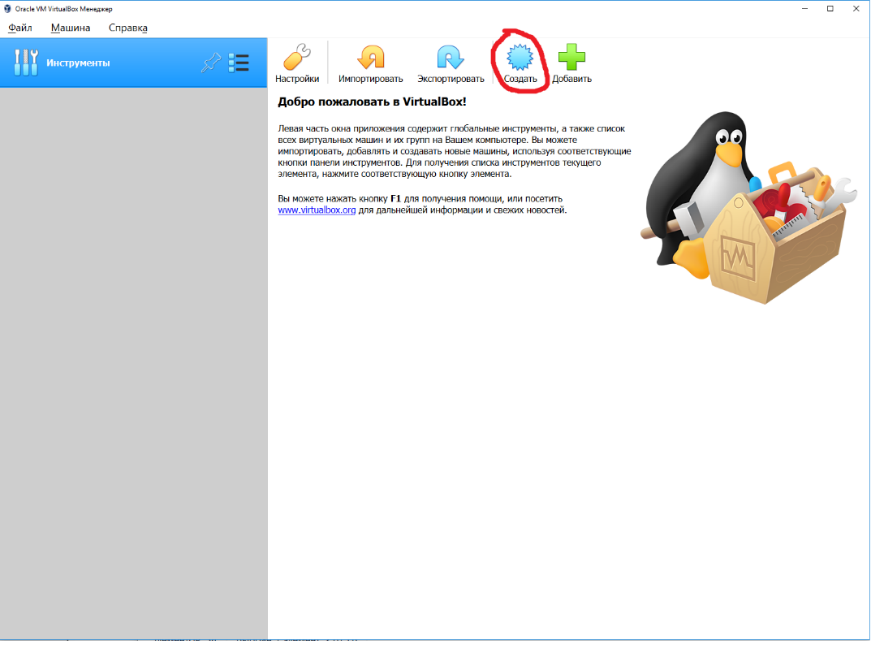
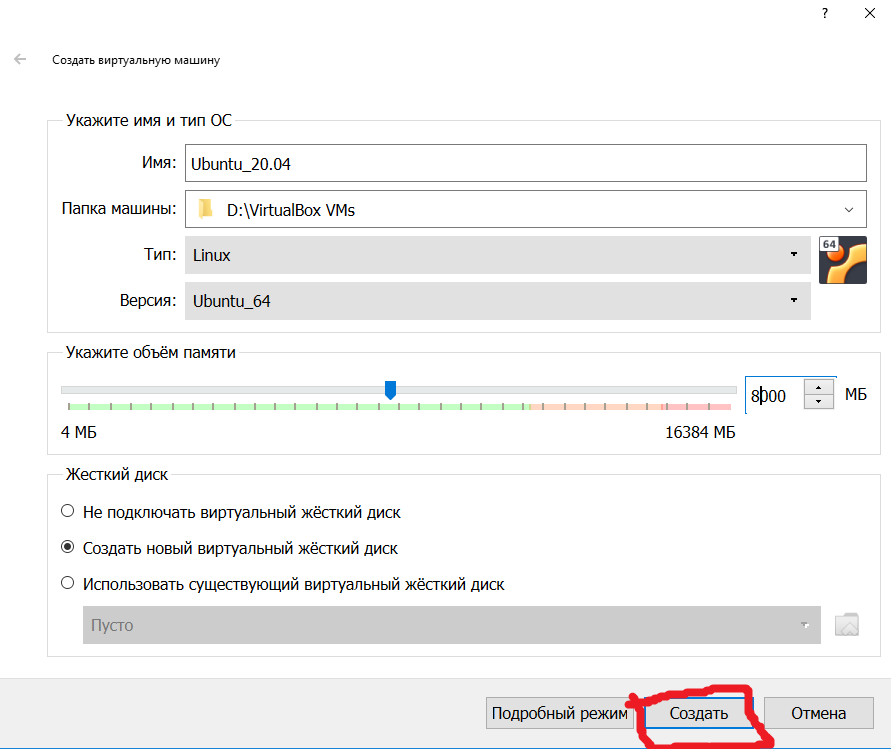
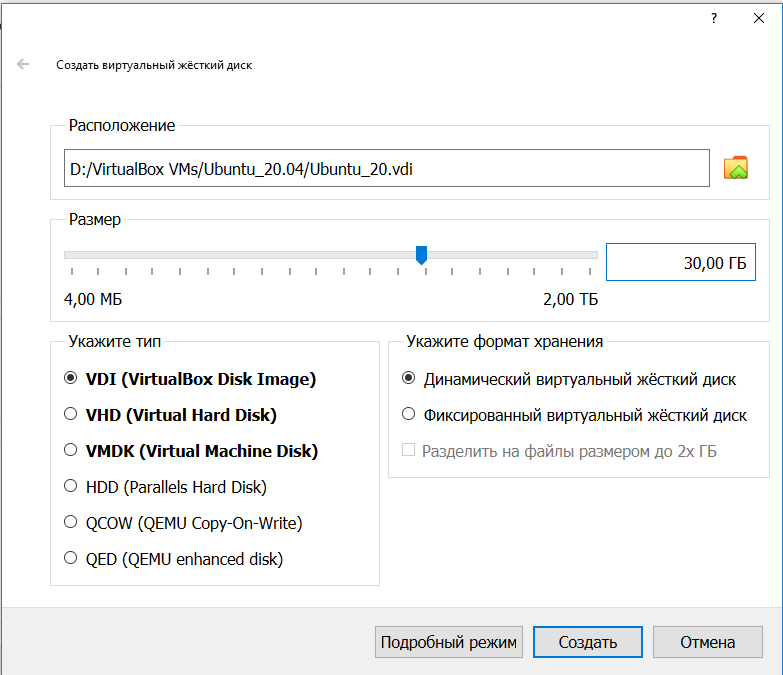
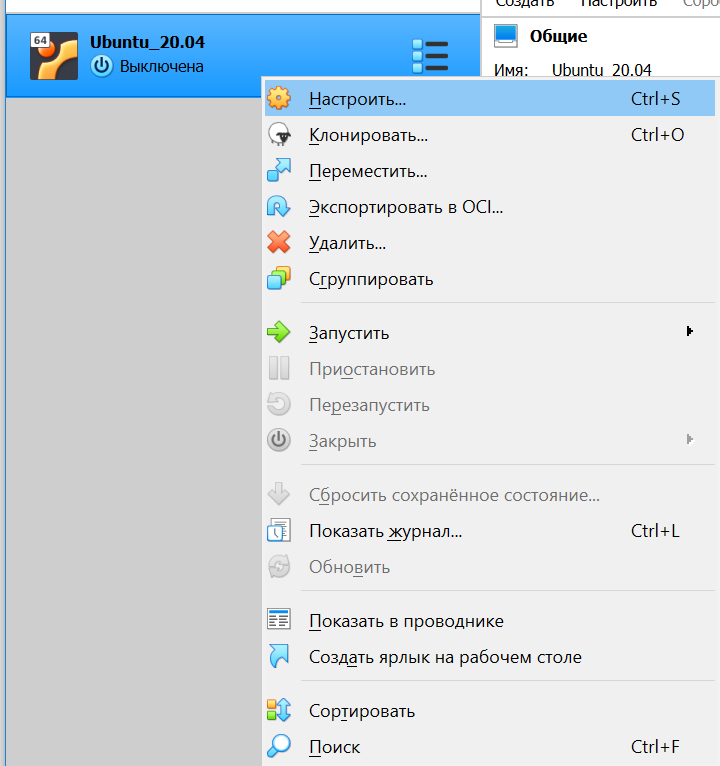
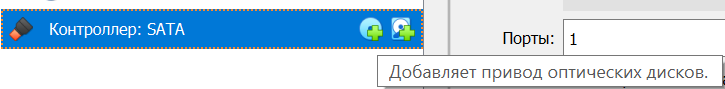
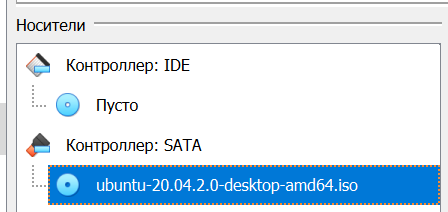
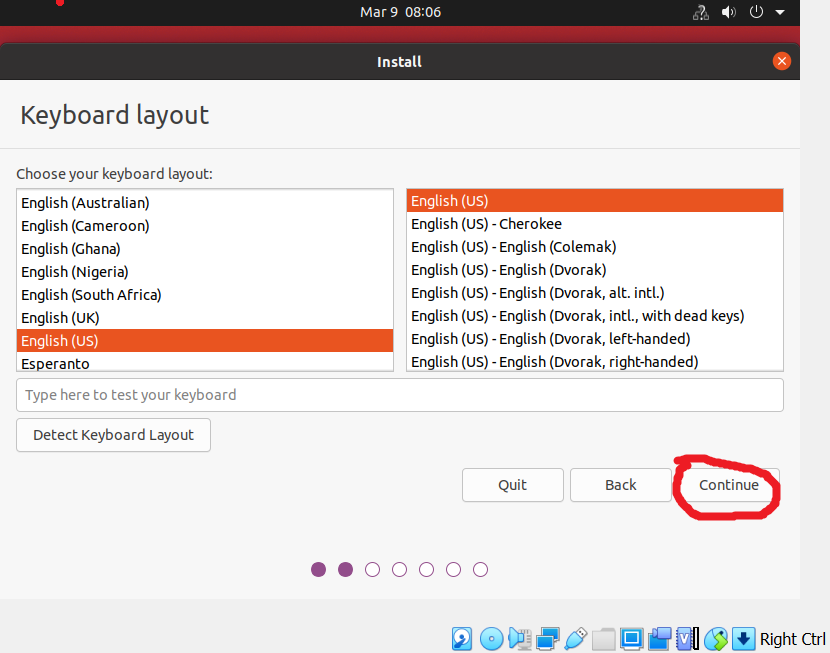
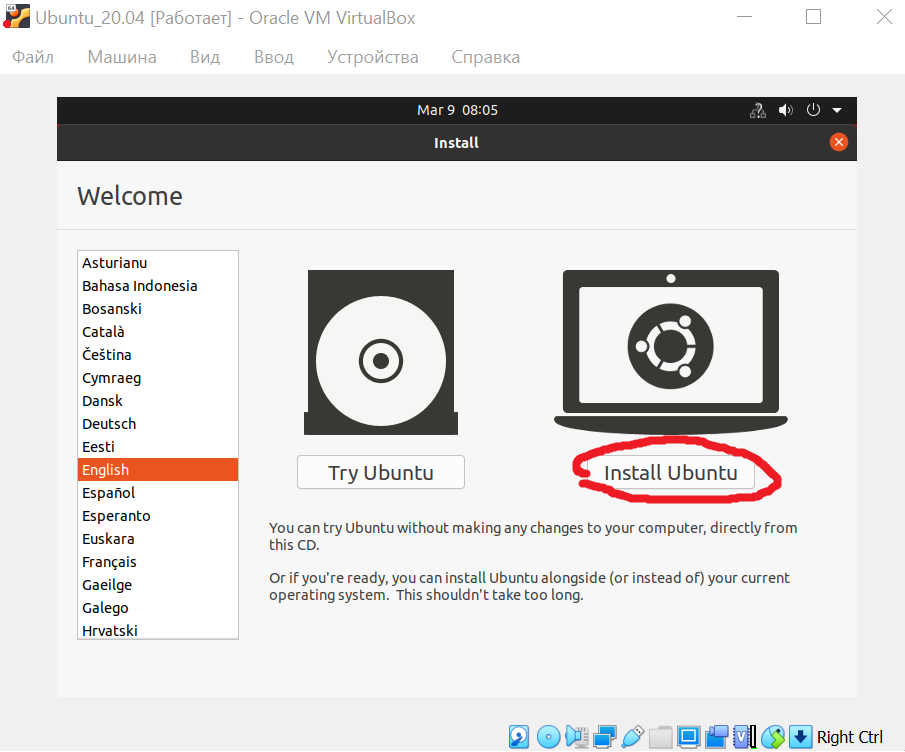
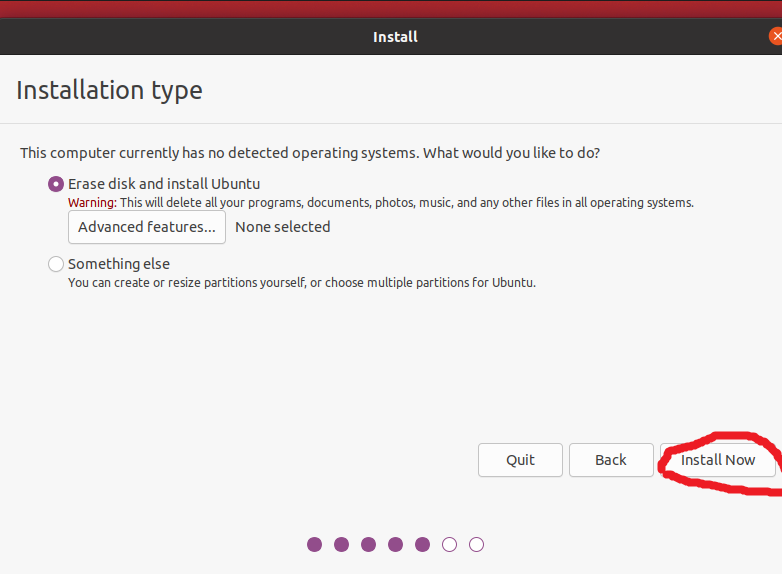
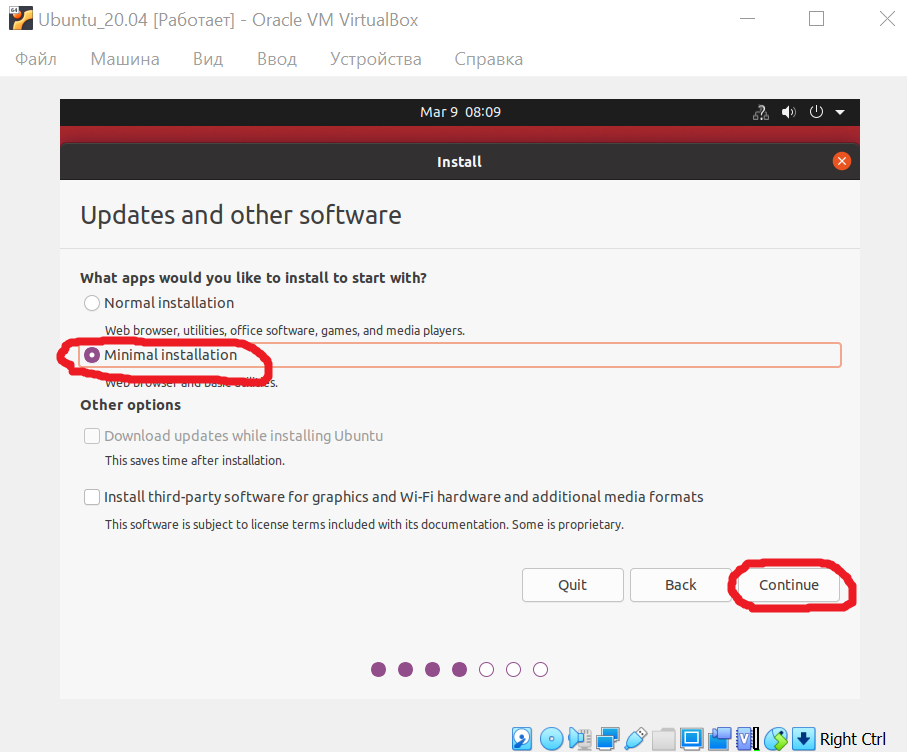
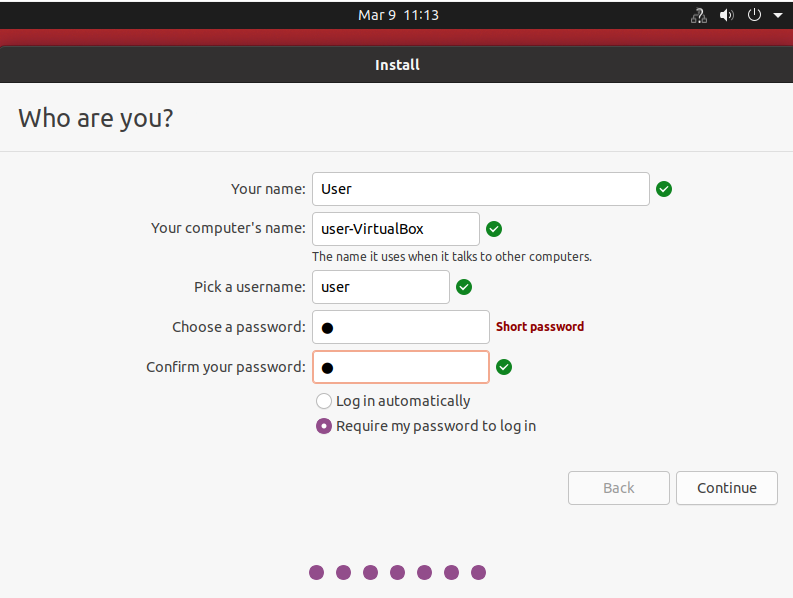
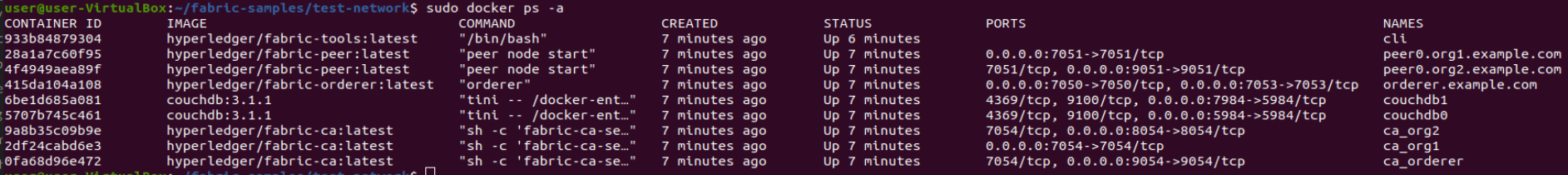
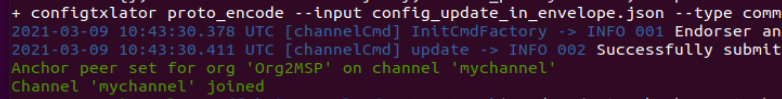
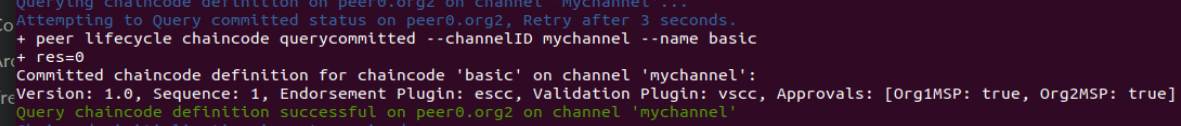
Установка VB для пользователей Windows

1. **Скачать и установить VirtualBox:**  
   <https://download.virtualbox.org/virtualbox/6.1.18/VirtualBox-6.1.18-142142-Win.exe>
2. **Скачать последний LTS дистрибутив Ubuntu (Например, Ubuntu 20.04.2.0 LTS)**<https://ubuntu.com/download/desktop>
3. **Запустить VirtualBox и создать виртуальную ОС**  
     
     
   

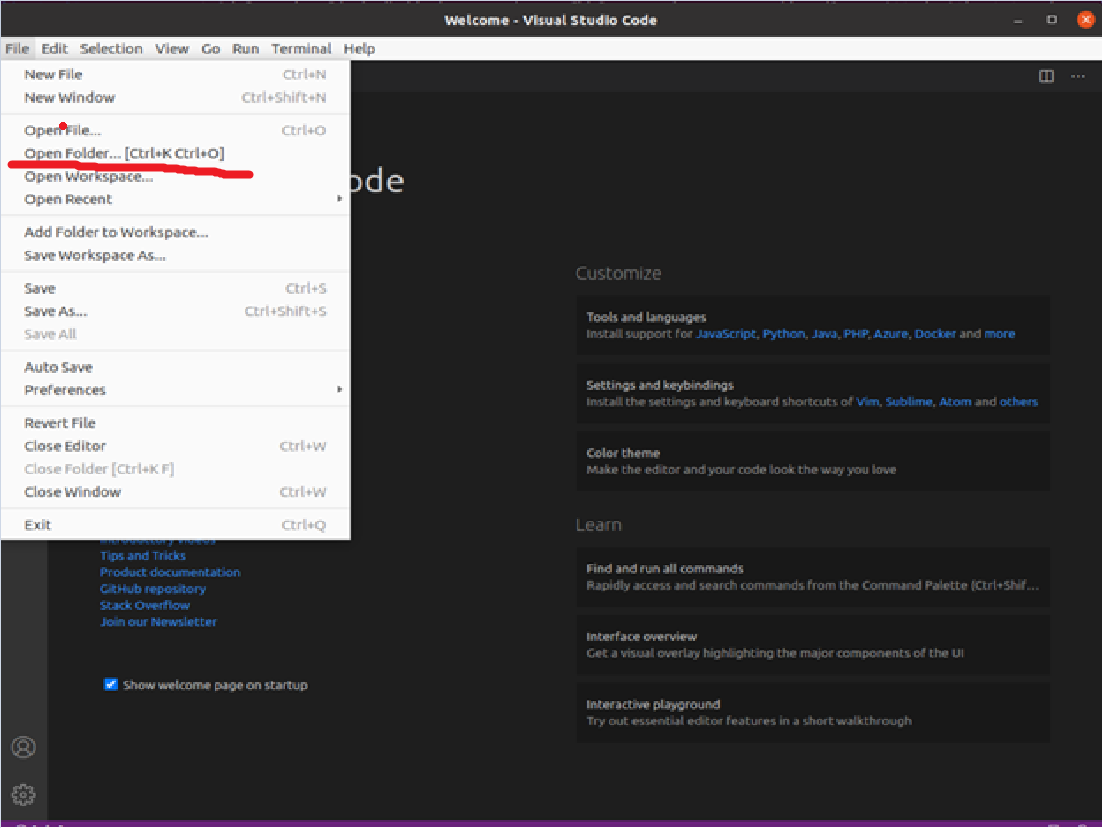
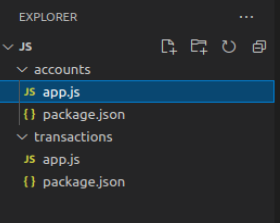
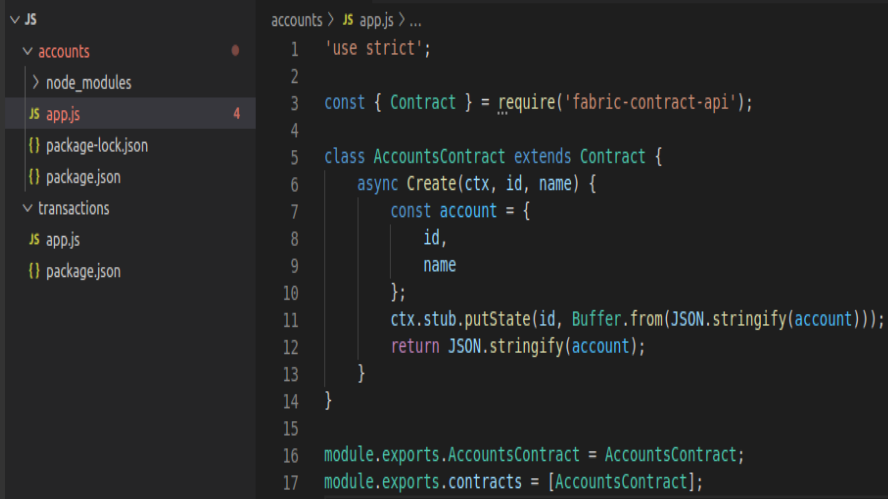
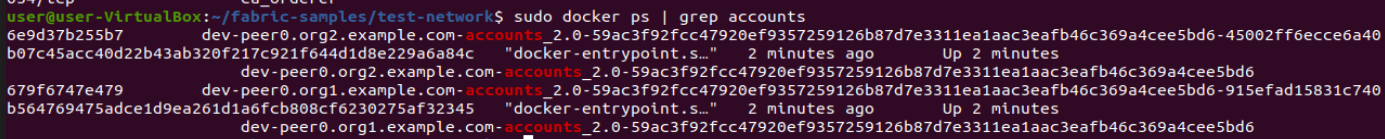
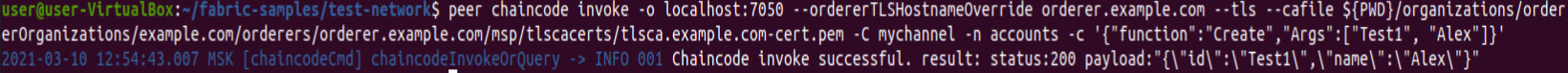
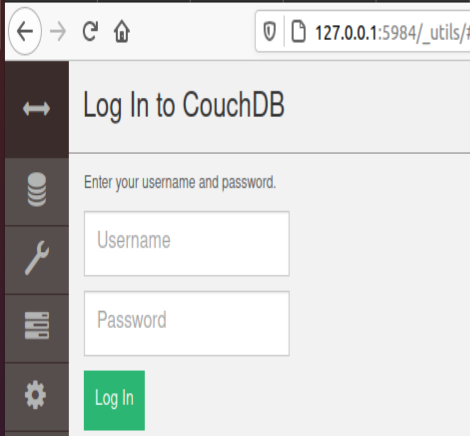
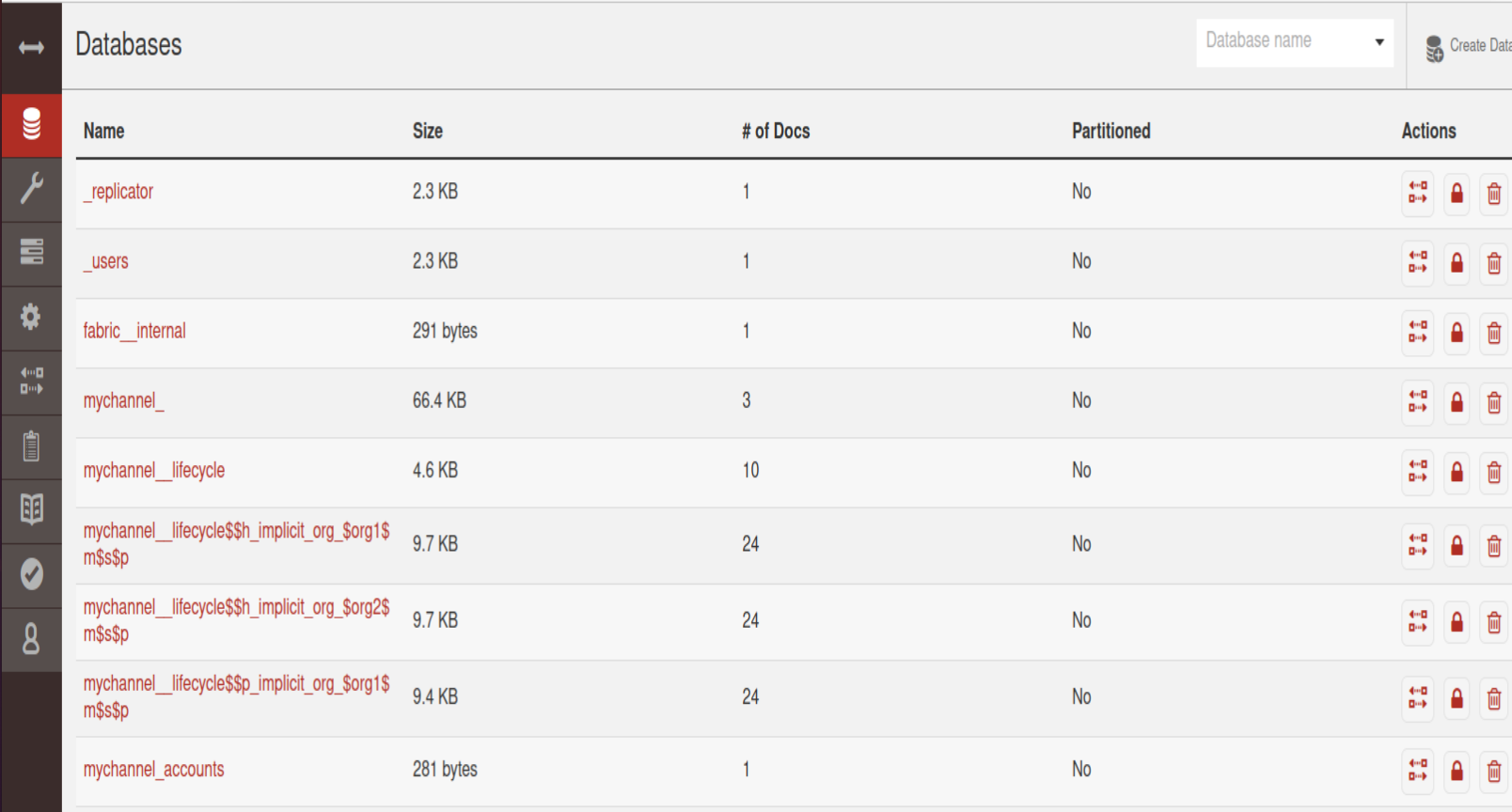
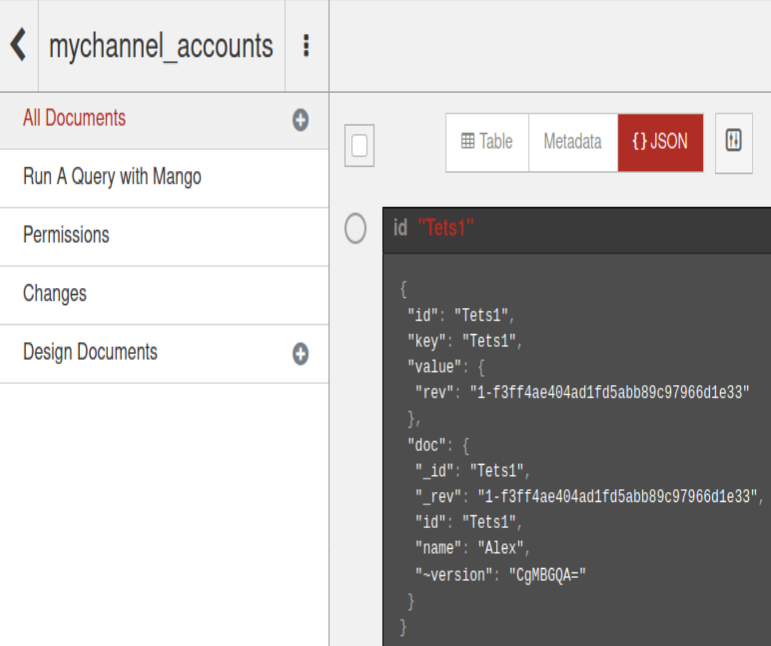
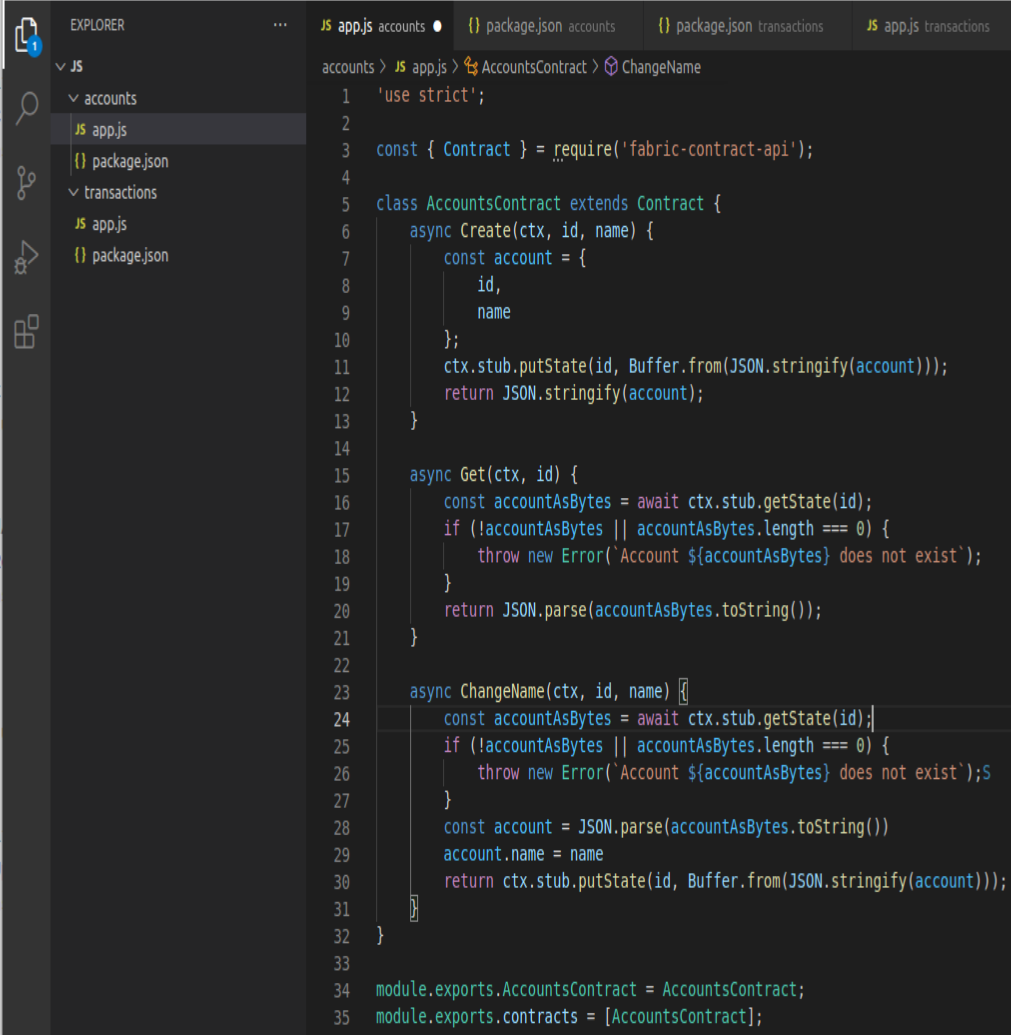
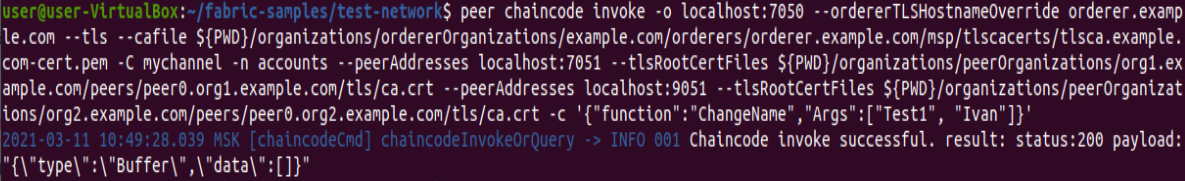


1. **Настроить виртуальную машину для установки ОС.**   
     
   Добавить скаченный образ   
   
2. **Запустить виртуальную машину и установить Ubuntu**  
     
     
     
     
   

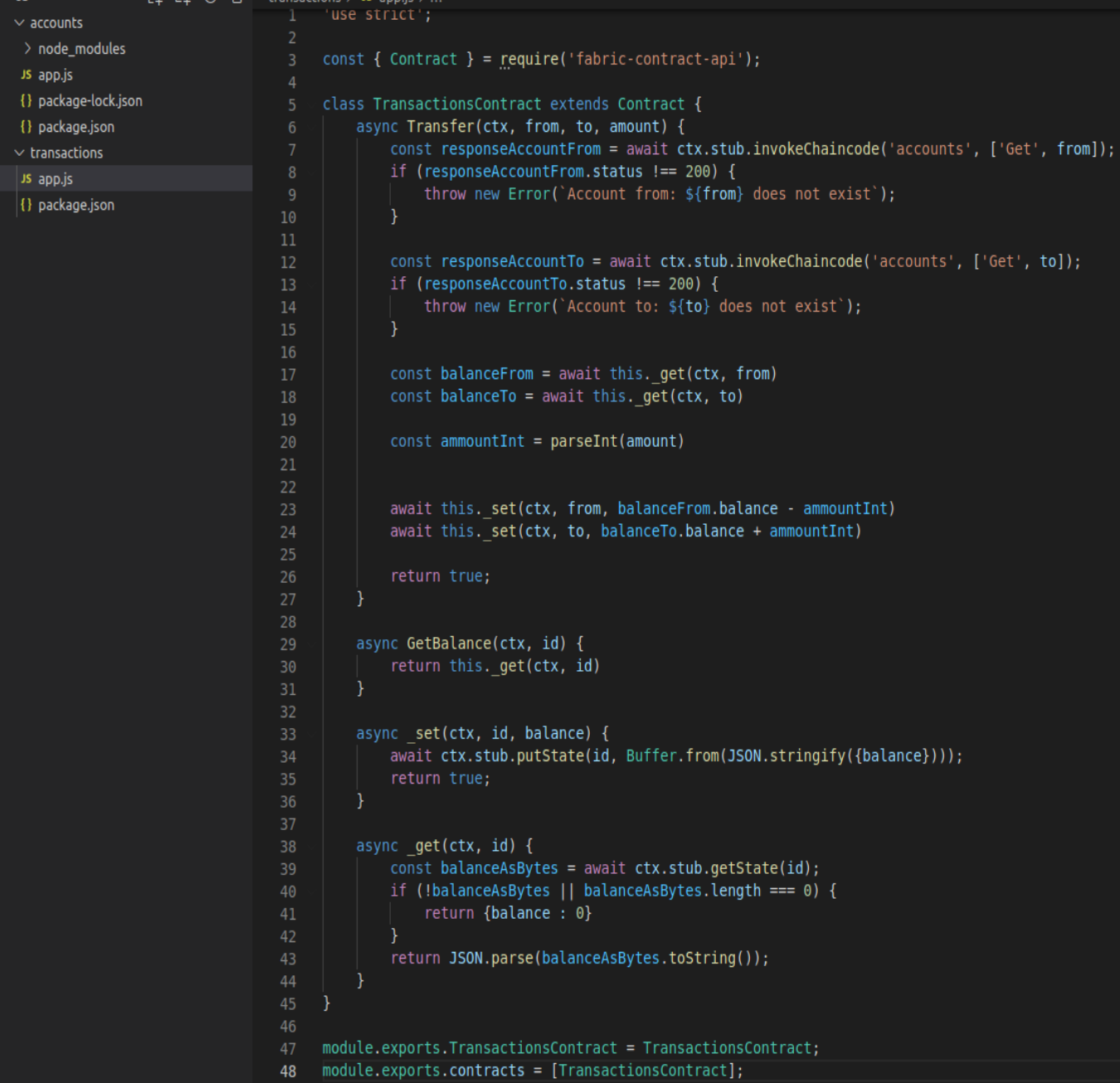
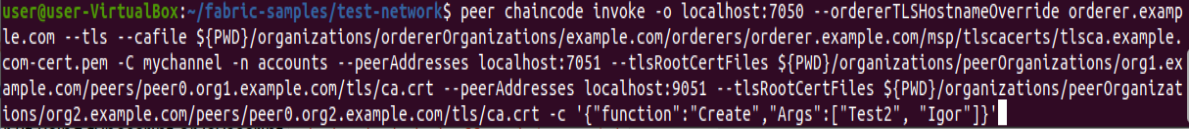
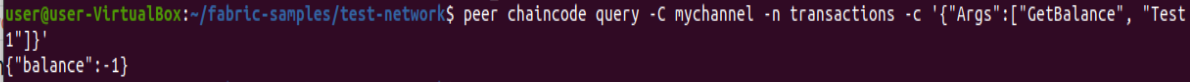
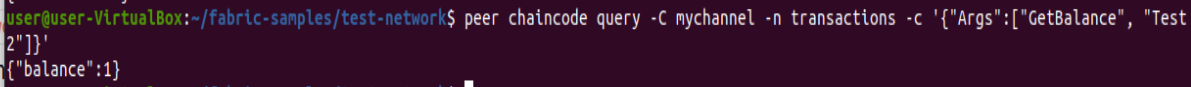
Развертывание и настройка сети Hyperledger Fabric

1. **Скачать и установить Visual Studio Code:**  
   <https://code.visualstudio.com/#alt-downloads>
2. **Установить последнюю версию git, curl, jq**  
   sudo apt-get install git  
   sudo apt-get install curl  
   sudo apt-get install jq
3. **Установить Docker**  
   sudo apt-get -y install docker-compose  
   sudo systemctl start docker  
   sudo systemctl enable docker  
   sudo usermod -a -G docker user
4. **Установить GO**  
   sudo add-apt-repository ppa:longsleep/golang-backports  
   sudo apt install golang
5. **Скачать Fabric Samples, докер образцы hyperledger и некоторые бинарные утилиты**  
   cd /home/user/  
   curl -sSL https://bit.ly/2ysbOFE | sudo bash -s
6. **Запустить сеть Hyperledger (Вместе с CA и базой данных couchDB)**  
   cd /fabric-samples/test-network/  
   sudo ./network.sh up -ca -s couchdb  
   **Убедиться что сеть поднялась**  
   sudo docker ps –a  
   
7. **Создать канал с именем *mychannel***  
   sudo ./network.sh createChannel -c mychannel  
   После выполнения команды, должны получить сообщение об успешном подключении к каналу  
   
8. **Изменить права на созданную папку organization для текущего пользователя:**sudo chown $USER:$USER /home/user/fabric-samples/test-network/organizations/ -R
9. **Установить chaincode-пример с именем basic**   
   sudo ./network.sh deployCC -cnn basic -ccp ../asset-transfer-basic/chaincode -ccl go  
   После установки должны увидеть такое сообщение  
   

Разработка чейнкода(смарт контракта)

1. **Установить NodeJS 14**  
   curl -sL https://deb.nodesource.com/setup\_14.x | sudo -E bash -  
   sudo apt install nodejs
2. **Создадим директорию с чейнкодами**  
    mkdir /home/user/chaincodes/js –p
3. **Откроем созданную директорию VS Code:**  
   
4. **Создать структуру директориев и файлов как показано на рисунке:**  
   
5. **Добавить для файлов package.json следующие содержимое**  
   
6. **Выполнить установку npm пакетов**  
   cd /home/user/chaincodes/js/accounts && npm i  
   cd /home/user/chaincodes/js/transactions && npm i
7. **Создадим базовую структуру чейнкода accounts(Создадим класс AccountsContracts с методом создания аккаунта Create), для это отредактируем файл и добавим следующие строки кода:**  
   
8. **Из директории /home/user/fabric-samples/test-network/ выполнить команду для деплоя чейнкода:**  
   sudo ./network.sh deployCC -ccn accounts -ccp /home/user/chaincodes/accounts -ccl javascript
9. **Проверить что чейнкод установился верно, для этого посмотреть создался ли контейнер с чейнкодом аккаунтов:**  
   sudo docker ps | grep accounts  
   
10. **Теперь необходимо проверить вызов метода этого чейнкода,** 
    1. **Перейти в директорию**: cd /home/user/fabric-samples/test-network/
    2. **Добавить в переменную среды PATH путь к бинарникам:**   
       export PATH=${PWD}/../bin:$PATH  
       
    3. **Еще понадобится уcтановить переменную FABRIC\_CFG\_PATH:**  
       export FABRIC\_CFG\_PATH=$PWD/../config/
    4. **Установить переменные для PEER для обращения к пиру организации Org1:**  
       1. export CORE\_PEER\_TLS\_ENABLED=true  
       2. export CORE\_PEER\_LOCALMSPID="Org1MSP"  
       3. export CORE\_PEER\_TLS\_ROOTCERT\_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt  
       4. export CORE\_PEER\_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp  
       5. export CORE\_PEER\_ADDRESS=localhost:7051
    5. **Путем вызова чейнкода создадим аккаунт с идентификатором Test1 и именем Alex для этого выполним команду:**  
       peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n accounts --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Create","Args":["Test1", "Alex"]}'  
       
    6. **Зайти в интерфейс DB и убедится, что запись создалась. Для этого перейти по адресу в браузере 127.0.0.1:5984/\_utils/ ввести логин:** *admin***, пароль:** *adminpw*
    7. **Открыть mychannel\_accounts:**  
         
       
11. **Улучшим код и добавим 2 метода для получения аккаунта по его идентификатору и метод изменения имени по идентификатору аккаунта:**  
    
12. **Обновим чейнкод аккаунтов для этого добавим параметры к команде из пункта 8 -ccv 2.0 –ccs 2:**  
    sudo ./network.sh deployCC -ccn accounts -ccp /home/user/chaincodes/accounts -ccl javascript -ccv 2.0 -css 2
13. **Вызвать метод чейнкода ChangeName и изменить имя Alex на Ivan по идентификатору Test1:**  
     peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n accounts --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"ChangeName","Args":["Test1", "Ivan"]}'  
    
14. **Теперь выполним команду peer chaincode query для получения информации по изменённому аккаунту:**  
    peer chaincode query -C mychannel -n accounts -c '{"Args":["Get","Test1"]}'  
    

Добавление чейнкода transactions

1. **Создадим новый чейнкод transactions**, который будет выполнять денежный перевод с одного аккаунта на другой по идентификаторам аккаунтов. В этом чейнкоде реализуем метод transfer, который будет получать на вход идентификаторы from, to и amount - сумму перевода. Метод будет обращаться к чейнкоду accounts для того что бы проверить существуют ли аккаунты с данными идентификаторами, получать балансы отправителя и получателя, проводит расчет новых балансов и сохранят их в БД. Так же добавим функцию получения баланса по идентификатору аккаунта GetBalance.  
   
2. **Задеплоить чейнкод transactions:**  
   sudo ./network.sh deployCC -ccn transactions -ccp /home/user/chaincodes/transactions -ccl javascript
3. **Создать аккаунт №2 с идентификатором test2 и именем Igor:**  
   peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n accounts --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Create","Args":["Test2", "Igor"]}'  
   
4. **Произвести перевод 1 денежной единицы с аккаунта Test1 на аккаунт Test2(Перевод 1 ДЕ от Ивана к Игорю)**  
   peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n transactions --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Transfer","Args":["Test1", " Test2","1"]}'
5. **Получить баланс аккаунта Test1:**peer chaincode query -C mychannel -n transactions -c '{"Args":["GetBalance","Test1"]}'  
   
6. **Получить баланс аккаунта Test2:**  
   peer chaincode query -C mychannel -n transactions -c '{"Args":["GetBalance","Test2"]}'  
   

Самостоятельное задание

1. Добавить метод topup, который будет пополнять аккаунт на переданную сумму по идентификатору аккаунта
2. Добавить проверку баланса аккаунта from на достаточность средств для перевода в метод transfer.
3. Обновить чейнкод transactions(-ccv 2.0 –ccs 2)
4. Пополнить баланс аккаунта Test1 на сумму 100
5. Перевести перевод с аккаунта Test1 на аккаунт Test2 80 единиц
6. Проверить балансы аккаунтов Test1 и Test2