# Property Registration

## Smart Contract Solution Document

**Ankit Kumar**

**4/20/2023**

Document contains step by step guide to user who would like to understand different methods available implemented in the smart contracts and execute to understand the output.

## Introduction

There are two smart contracts

1) for User
2) Registrar. NodeJS is used to write these smart contracts.

## Pre-requisites

These smart contracts were developed using the VM provided by UpGrad which contains necessary software for deploying and executing smart contracts.

## Users

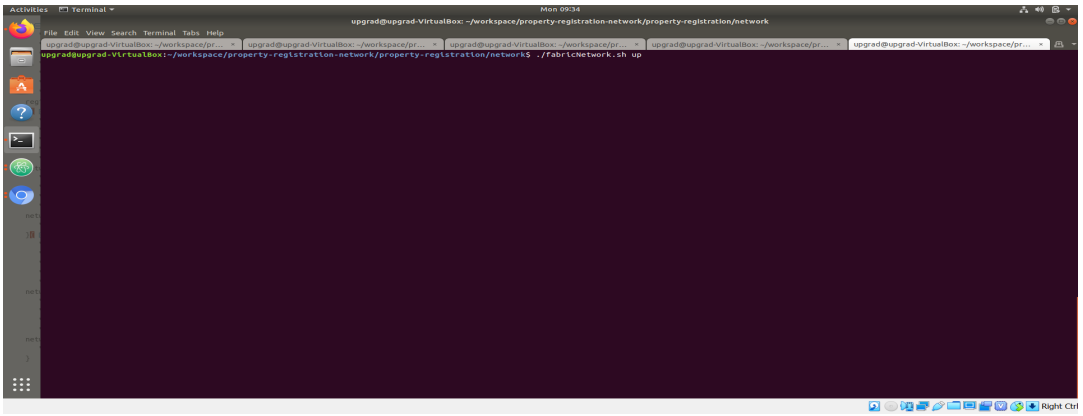
- Seller – AnkitKumar1, ankit.kkumar@gmail.com, 1234123412341231
- Buyer – Ankit Kumar Dev, dev.ankit.kkumar@gmail.com, 1234123412341232

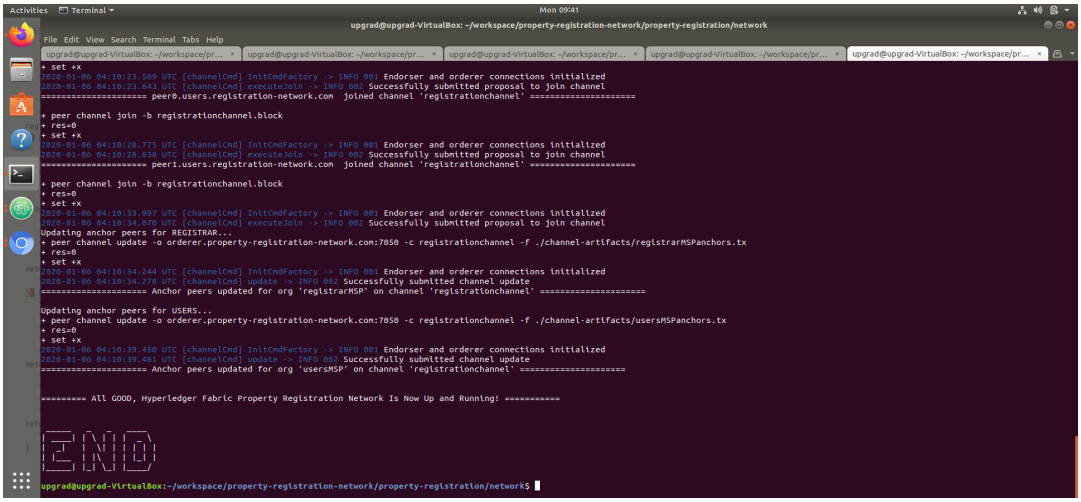
## Step 1: Bootstrap the network

### Command

1. Go to the 'network' folder of the project
2. Execute the command ./fabricNetwork.sh up, which will prompt for the confirmation to boot the network. Enter 'Y' in the terminal to bring up the network.

### Terminal Screenshot:



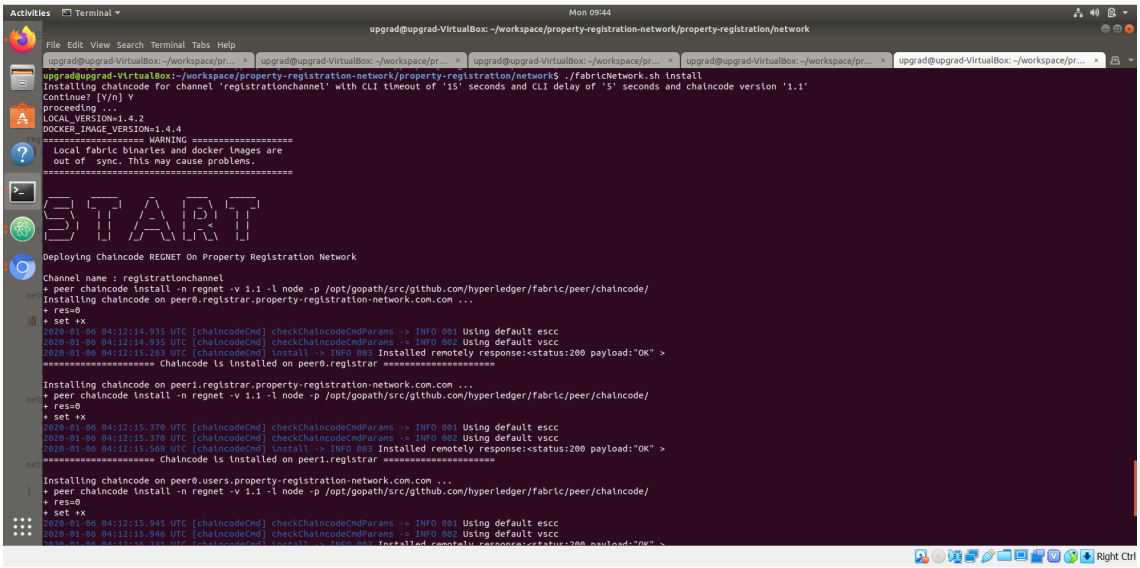

Activities
Terminal
Mon 09:41
upgrad@upgrad-VirtualBox: ~/workspace/property-registration-network/property-registration/network
File Edit View Search Terminal Tabs Help
upgrad@upgrad-VirtualBox: ~/workspace/pr...
upgrad@upgrad-VirtualBox: ~/workspace/pr...
upgrad@upgrad-VirtualBox: ~/workspace/pr...
upgrad@upgrad-VirtualBox: ~/workspace/pr...
upgrad@upgrad-VirtualBox: ~/workspace/pr...
+ set +x
2020-01-06 04:10:23.569 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-01-06 04:10:23.643 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
==================== peer0.users.registration-network.com joined channel 'registrationchannel' ====================
+ peer channel join -b registrationchannel.block
+ res=0
+ set +x
2020-01-06 04:10:28.775 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-01-06 04:10:28.838 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
==================== peer1.users.registration-network.com joined channel 'registrationchannel' ====================
+ peer channel join -b registrationchannel.block
+ res=0
+ set +x
2020-01-06 04:10:33.997 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-01-06 04:10:34.070 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
Updating anchor peers for REGISTRAR...
+ peer channel update -o orderer.property-registration-network.com:7050 -c registrationchannel -f ./channel-artifacts/registrarMSPanchors.tx
+ res=0
+ set +x
2020-01-06 04:10:34.244 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-01-06 04:10:34.278 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
==================== Anchor peers updated for org 'registrarMSP' on channel 'registrationchannel' ====================
Updating anchor peers for USERS...
+ peer channel update -o orderer.property-registration-network.com:7050 -c registrationchannel -f ./channel-artifacts/usersMSPanchors.tx
+ res=0
+ set +x
2020-01-06 04:10:39.450 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2020-01-06 04:10:39.481 UTC [channelCmd] update -> INFO 002 Successfully submitted channel update
==================== Anchor peers updated for org 'usersMSP' on channel 'registrationchannel' ====================
========= All GOOD, Hyperledger Fabric Property Registration Network Is Now Up and Running! ==========
upgrad@upgrad-VirtualBox:~/workspace/property-registration-network/property-registration/network$

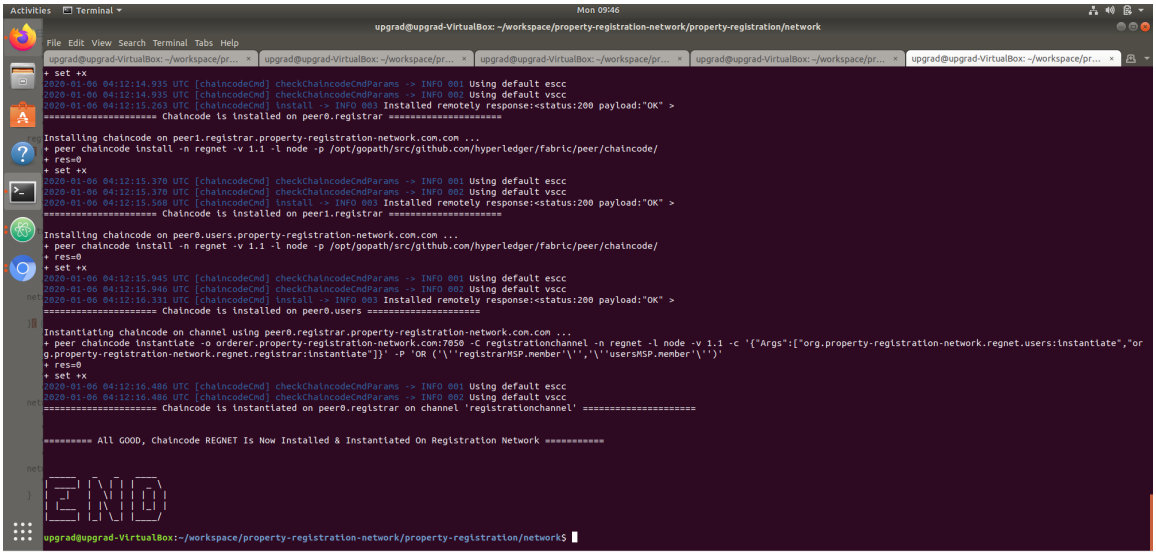
# Step 2: Chaincode Installation and Instantiation

## Command:

1. Once Step1 is complete, execute below command which will install & instantiate the smart contracts in chaincode container and allow us to invoke different methods listed below.
2. ./fabricNetwork.sh install

## Terminal



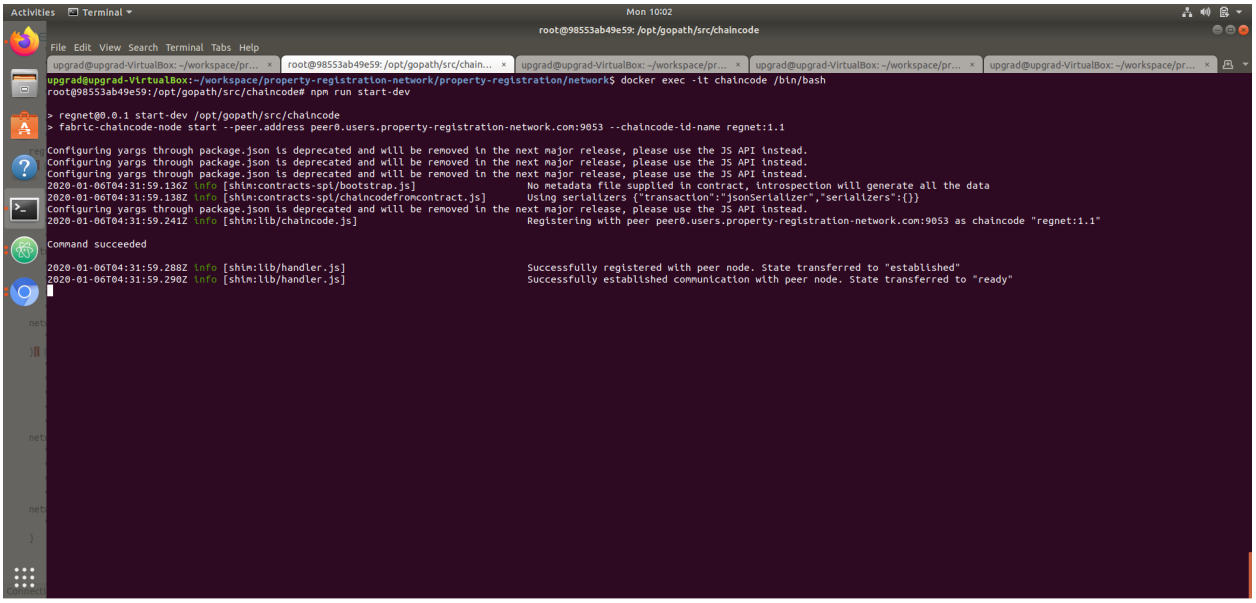
## Step 3: Start the chaincode node application

This is a must have step before proceeding to Step 4

### Command:

1. Enter into docker container for chaincode using command docker exec -it chaincode /bin/bash
2. Install node modules using 'npm install' command which will install all necessary Node modules inside chaincode container.
3. Start the node application: npm run start-dev

### Terminal:



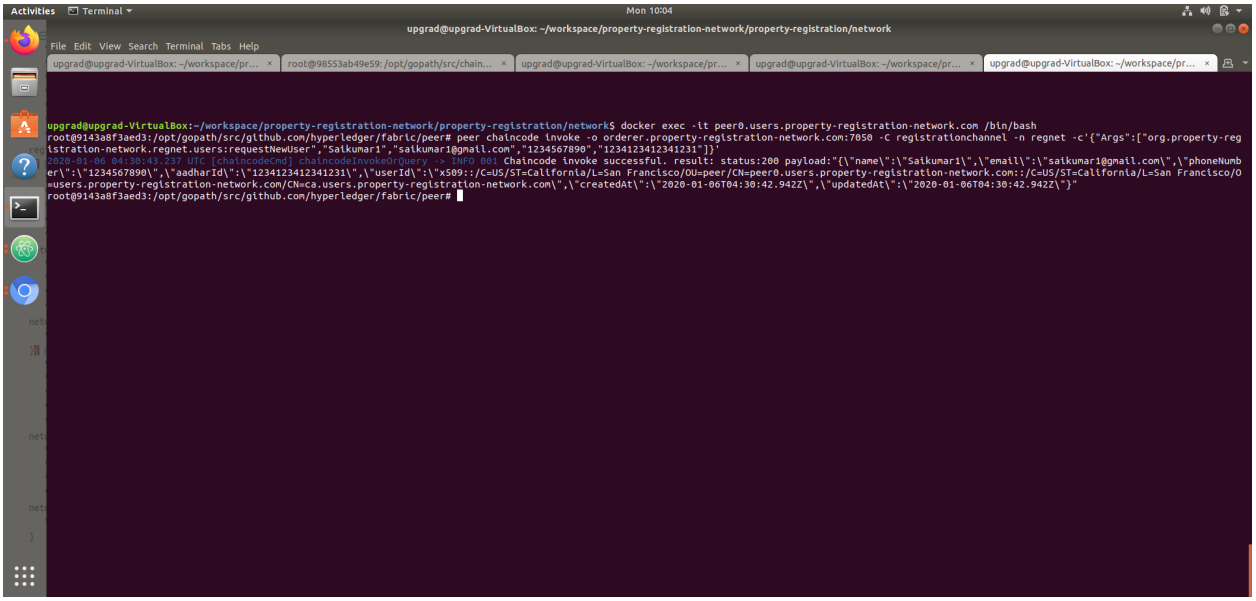
# Step 4: Invoke Smart Contract Methods

## Method 1: requestNewUser

### *Commands:*

1. Enter into Peer0 of users org: docker exec -it peer0.users.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:requestNewUser","Ankit Kumar","ankit.kkumar@gmail.com","1234567890","1234123412341231"]}'

### *Terminal:*

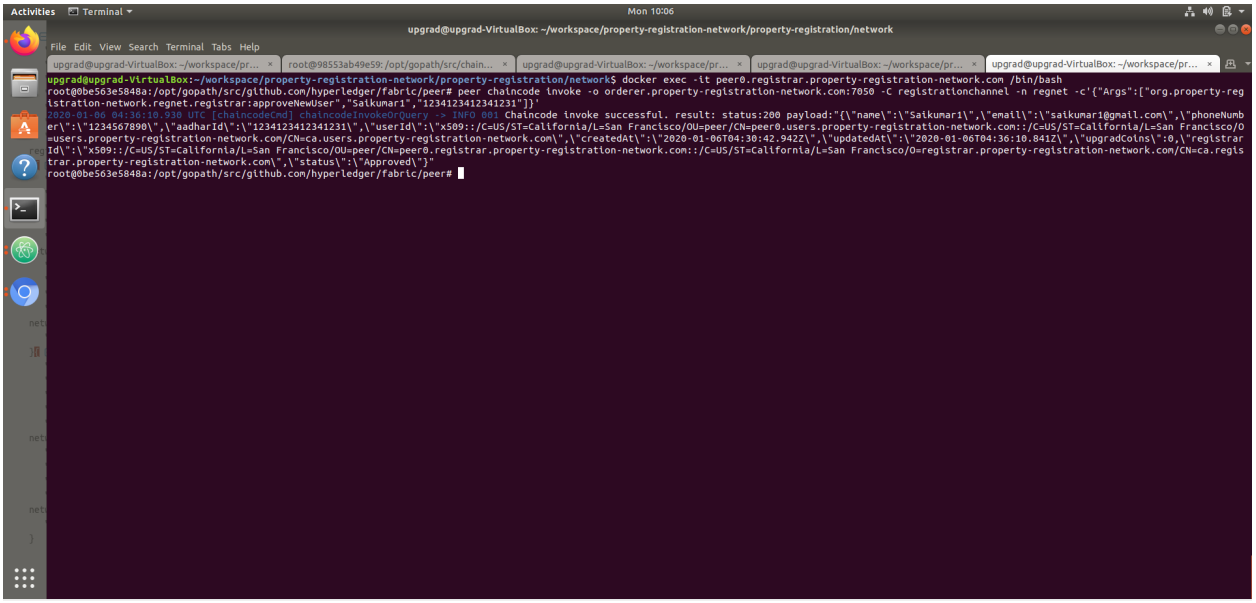


## Method 2: approveNewUser

### *Commands:*

1. Enter into Peer0 of registrar org: docker exec -it peer0.registrar.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.registrar:approveNewUser","Ankit Kumar","1234123412341231"]}'

### *Terminal:*

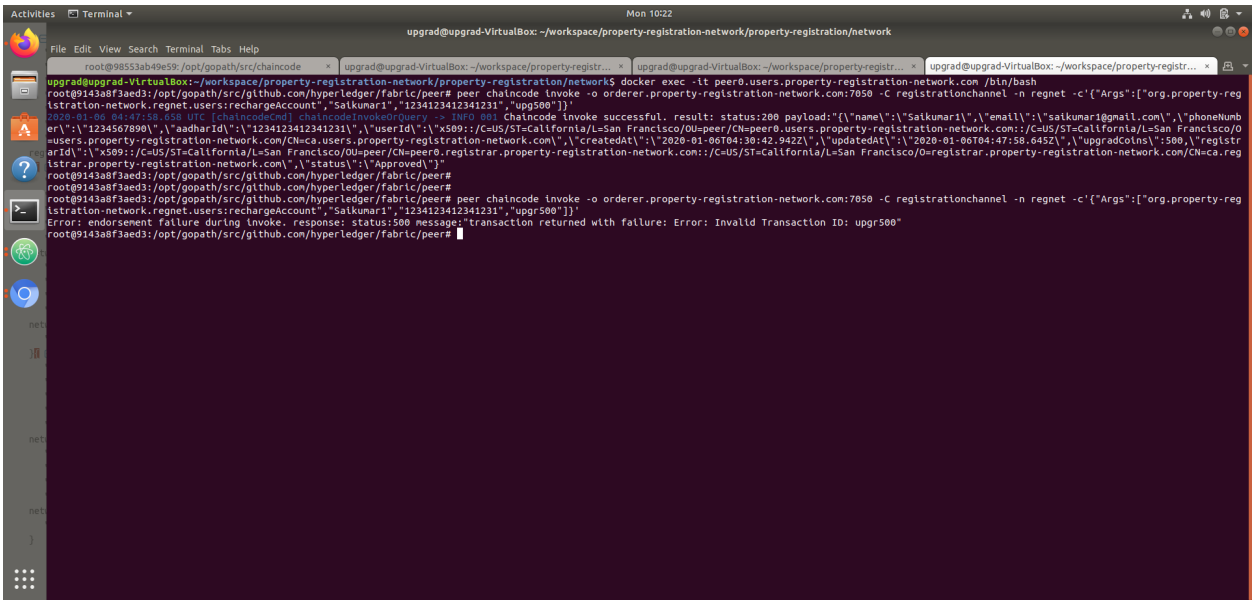


## Method 3: rechargeAccount

### *Commands:*

1. Enter into Peer0 of users org: docker exec -it peer0.users.property-registration-network.com /bin/bash
2. Execute command:
    a. Success: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:rechargeAccount","Ankit Kumar","1234123412341231","upg500"]}'
    b. Failure: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:rechargeAccount","Ankit Kumar","1234123412341231","upgr500"]}'

### *Terminal for Success & Failure:*



### Method 4: viewUser

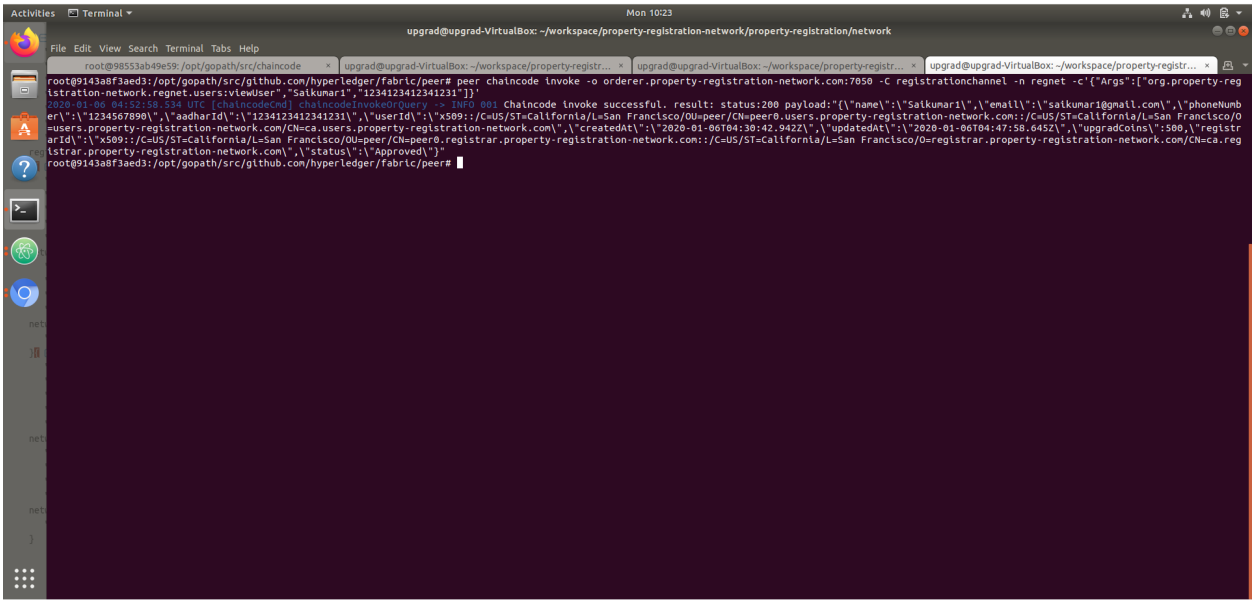
#### *Commands for Users Org:*

1. Enter into Peer0 of users org: docker exec -it peer0.users.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:viewUser","Ankit Kumar","1234123412341231"]}'

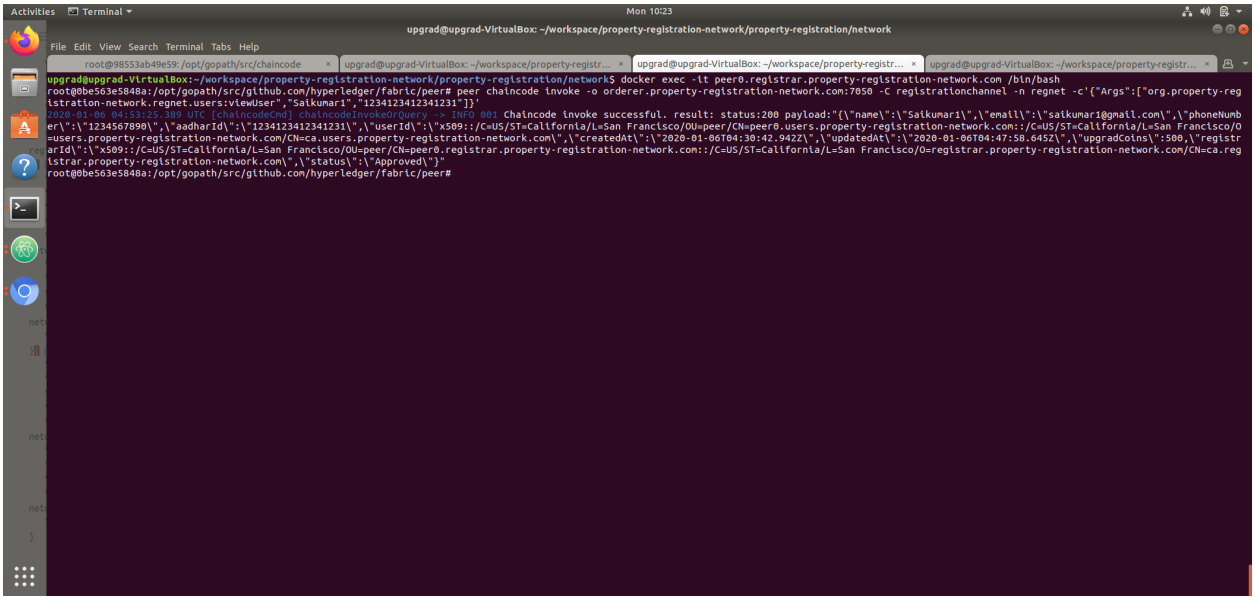
#### *Commands for Registrar Org:*

1. Enter into Peer0 of registrar org: docker exec -it peer0.registrar.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.registrar:viewUser","Ankit Kumar","1234123412341231"]}'

## *Terminal for User:*




## *Terminal for Registrar:*

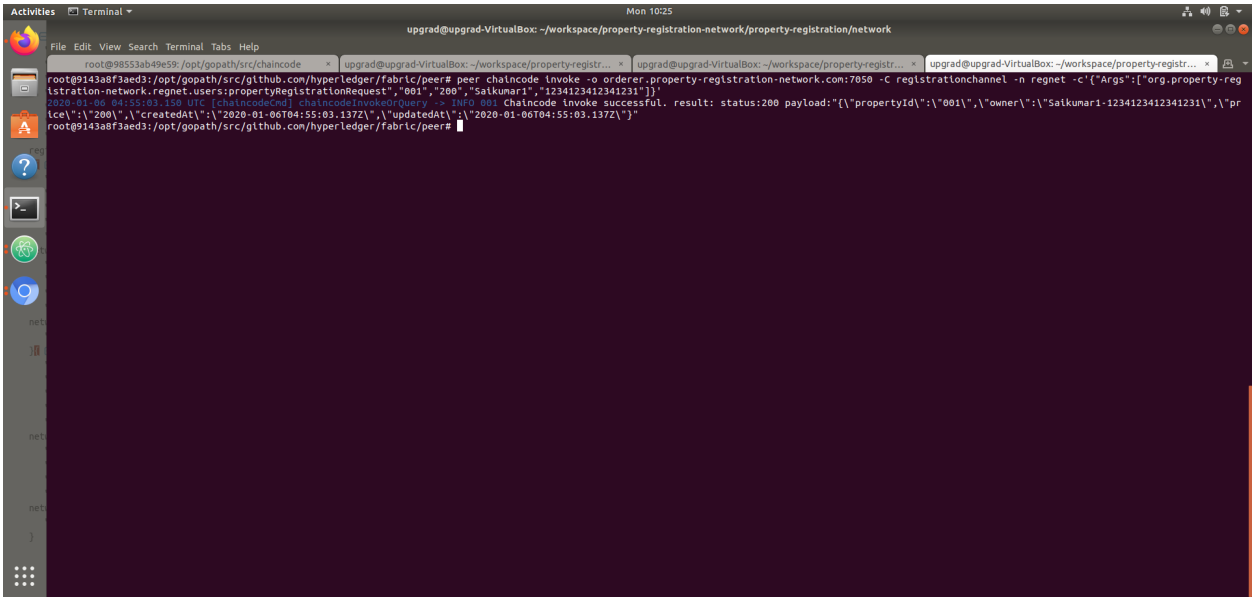


## Method 5: propertyRegistrationRequest

### *Commands:*

1. Enter into Peer0 of users org: docker exec -it peer0.users.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:propertyRegistrationRequest","001","200","Ankit Kumar","1234123412341231"]}'

### *Terminal:*

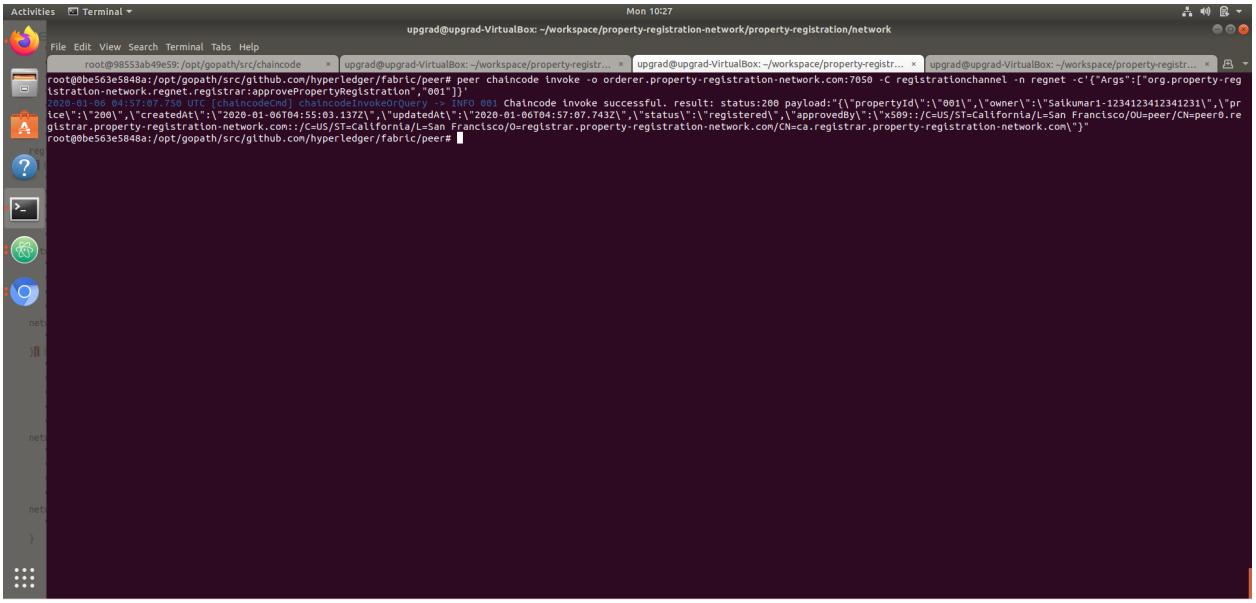


## Method 6: approvePropertyRegistration

### *Commands:*

1. Enter into Peer0 of registrar org: docker exec -it peer0.registrar.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.registrar:approvePropertyRegistration","001"]}'

### *Terminal:*



### Method 7: viewProperty

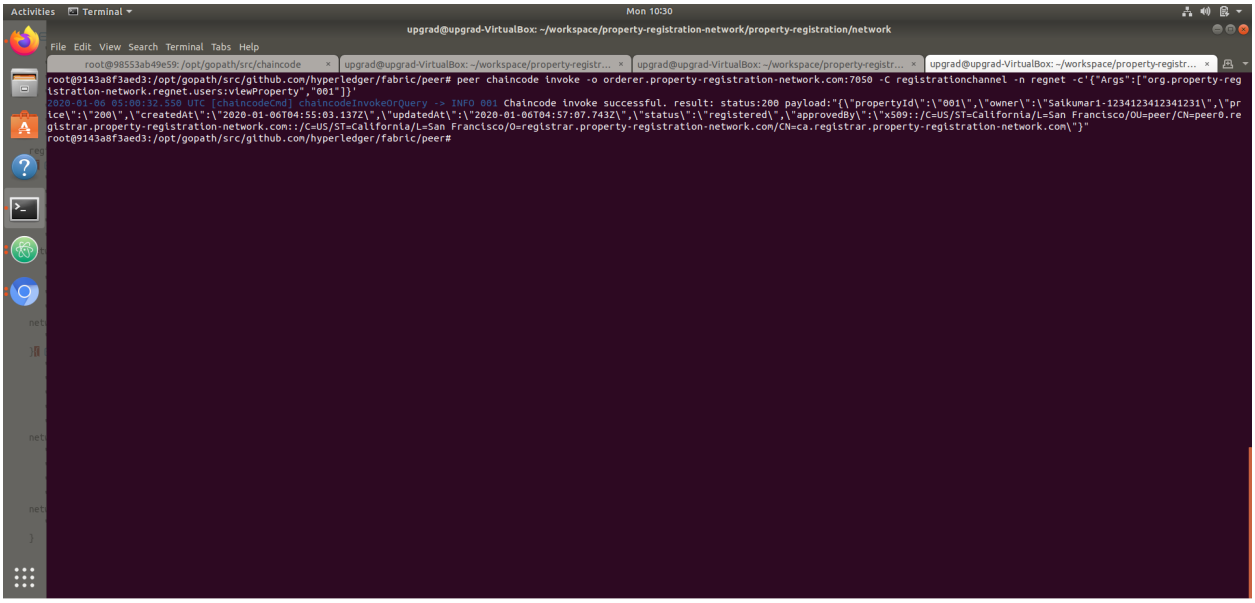
#### *Commands for Users Org:*

1. Enter into Peer0 of users org: docker exec -it peer0.users.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:viewProperty","001"]}'

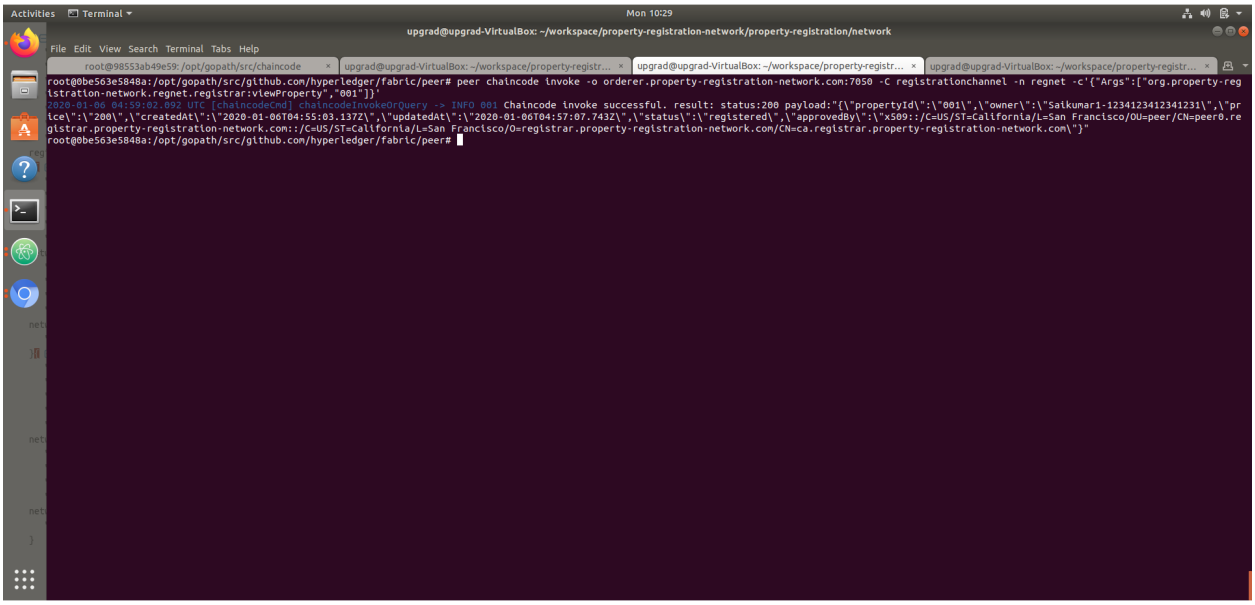
#### *Commands for Registrar Org:*

1. Enter into Peer0 of registrar org: docker exec -it peer0.registrar.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.registrar:viewProperty","001"]}'

## *Terminal for Users:*




## *Terminal for Registrar:*

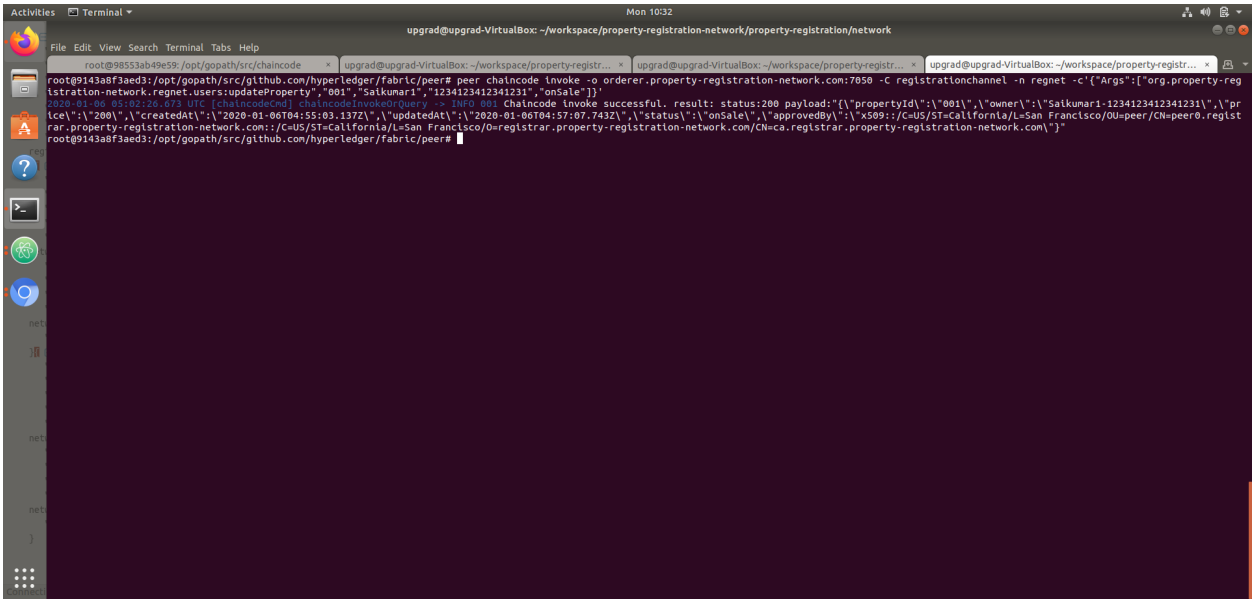


## Method 8: updateProperty

### *Commands:*

1. Enter into Peer0 of users org: docker exec -it peer0.users.property-registration-network.com /bin/bash
2. Execute command: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:updateProperty","001","Ankit Kumar","1234123412341231","onSale"]}'

### *Terminal:*



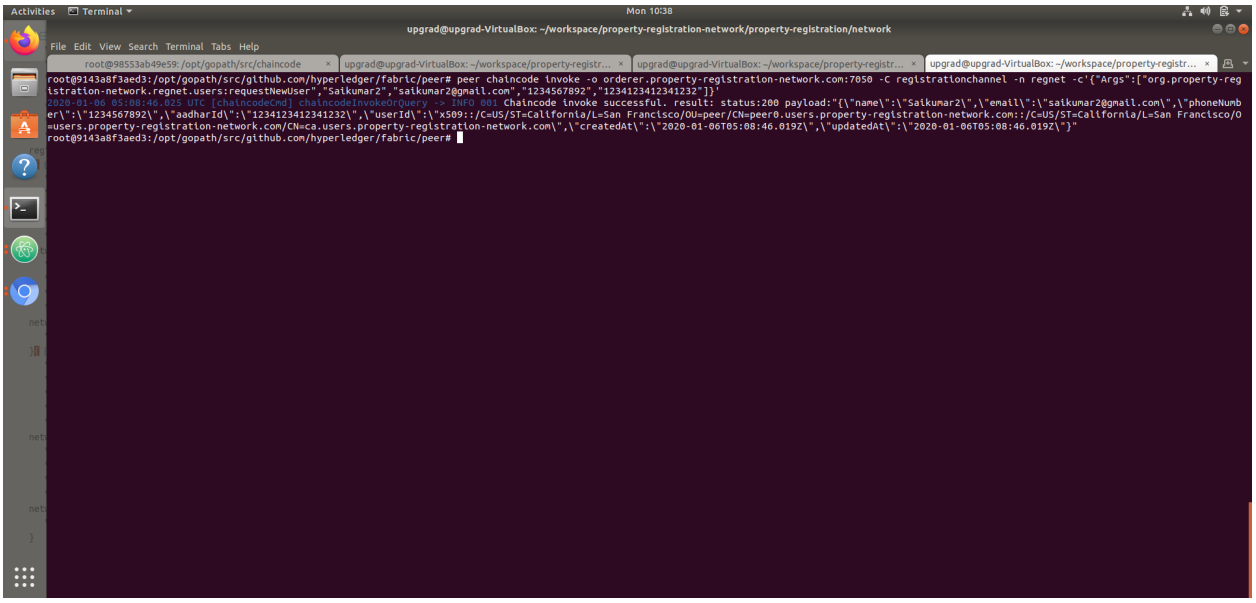
## Method 9: purchaseProperty

### *Pre-requisite:*

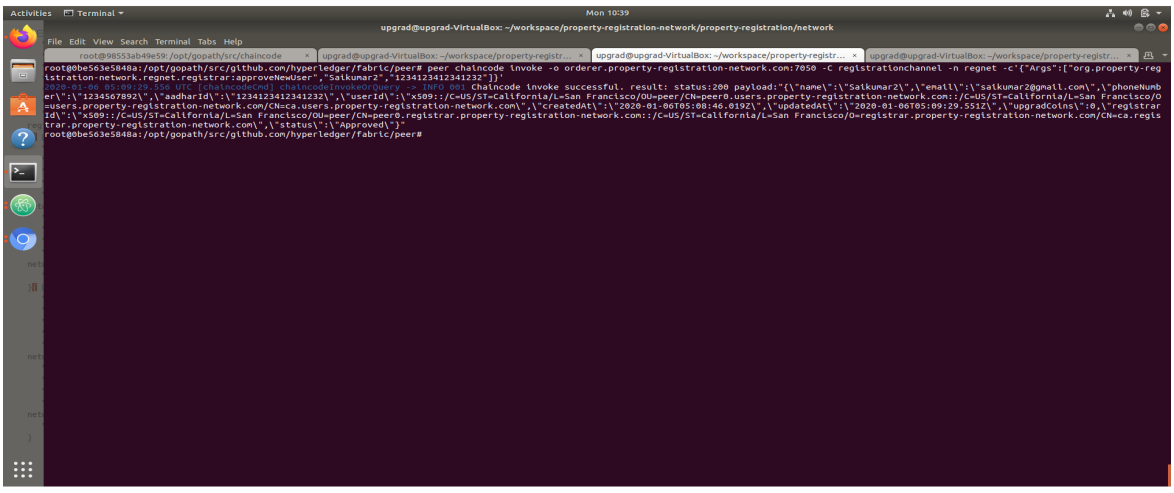
In order to purchase property, we need to register new user as mentioned below before proceeding with purchase of the property.

### *Steps to register new user (Ankit Kumar Dev) in the network and request property purchase*

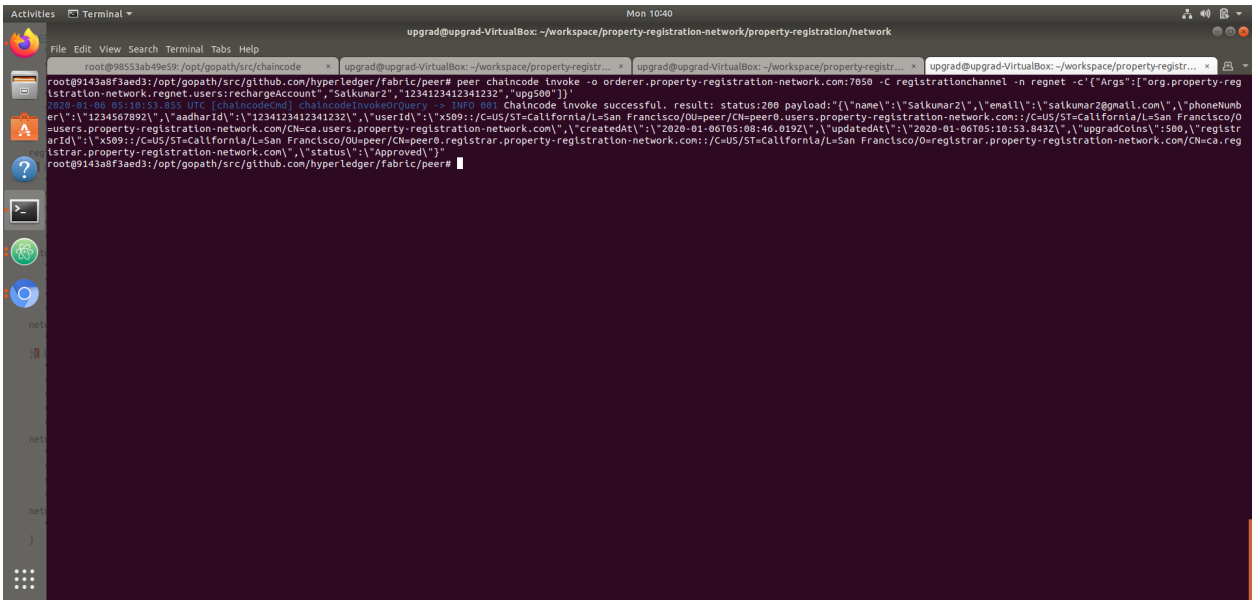
1. **requestNewUser**: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:requestNewUser","Ankit Kumar Dev","dev.ankit.kkumar@gmail.com","1234567892","1234123412341232"]}'

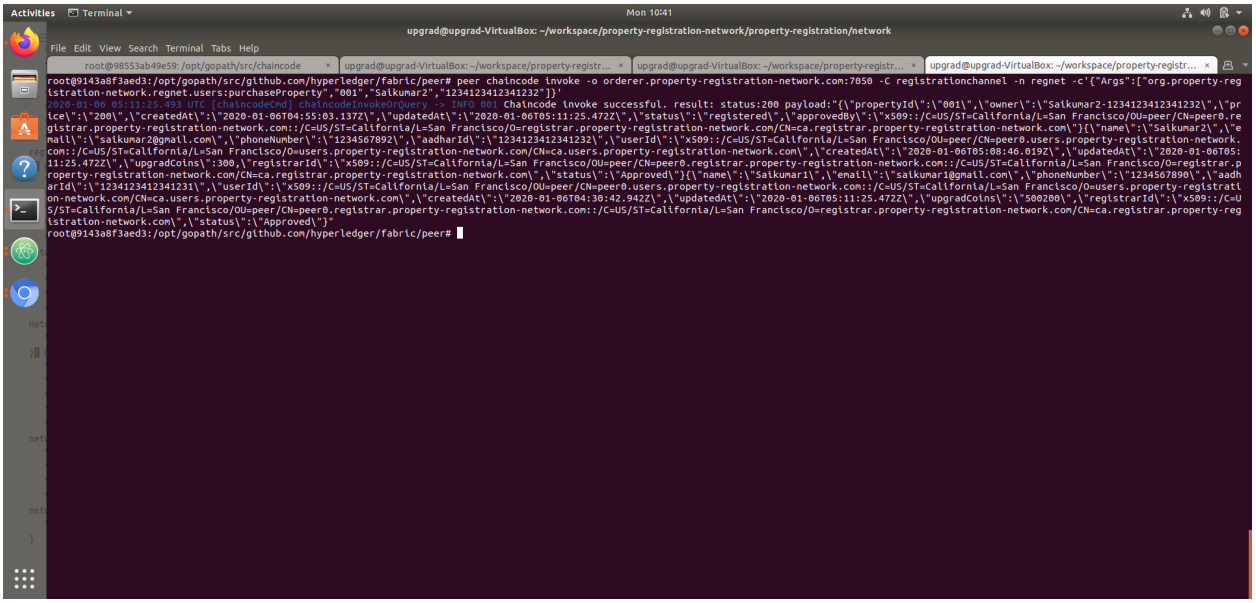



2. **approveNewUser**: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.registrar:approveNewUser","Ankit Kumar Dev","1234123412341232"]}'



3. **rechargeAccount**: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:rechargeAccount","Ankit Kumar Dev","1234123412341232","upg500"]}'




4. **purchaseProperty**: peer chaincode invoke -o orderer.property-registration-network.com:7050 -C registrationchannel -n regnet -c'{"Args":["org.property-registration-network.regnet.users:purchaseProperty","001","Ankit Kumar Dev","1234123412341232"]}'