# Udacity Blockchain Nanodegree Project 6 Write-Up

## 1. Contract Address

Steps followed to deploy the smart contracts included:

* Added Rinkeby network configuration to truffle.js (Infura hosted and provided by HDWalletProvider)
* To compile the contracts: `*truffle run compile*`
* To migrate the contracts to the Rinkey network: `*truffle migrate --reset --network rinkeby*`
* Terminal output from the above command is as follows (includes highlighted Address of each Contract)

Starting migrations...

======================

> Network name: 'rinkeby'

> Network id: 4

> Block gas limit: 7005292

1\_initial\_migration.js

======================

Deploying 'Migrations'

----------------------

> transaction hash: 0xef1d41dcc3affc20ec14c290bab629ef3b995ea6f08874598dcf8d1bc9193e65

> Blocks: 1 Seconds: 16

> contract address: 0xb59adAaf3c524ffe92984c636025DF867B3cbb3b

> account: 0x4EA55452B992b81C95f2077c15C09aCa1c6Fdaed

> balance: 5.9640155699999997

> gas used: 284908

> gas price: 10 gwei

> value sent: 0 ETH

> total cost: 0.00284908 ETH

> Saving migration to chain.

> Saving artifacts

-------------------------------------

> Total cost: 0.00284908 ETH

2\_deploy\_contracts.js

=====================

Deploying 'FarmerRole'

----------------------

> transaction hash: 0x83fd030c06ce122baafd5e77f9156bf2dc5ee78e5130ab20ce5ec68f6a37a2d9

> Blocks: 0 Seconds: 12

> contract address: **0xbc5818522e0225ce91547E463a47703c5E959738**

> account: 0x4EA55452B992b81C95f2077c15C09aCa1c6Fdaed

> balance: 5.9596454799999997

> gas used: 394975

> gas price: 10 gwei

> value sent: 0 ETH

> total cost: 0.00394975 ETH

Deploying 'DistributorRole'

---------------------------

> transaction hash: 0xf843c7d24ba1a70484fa107939ebf26e8f1f29f8f110cc2d43afa6027353f3d4

> Blocks: 1 Seconds: 24

> contract address: **0x75866d0E1057Be96b03a520C532692FED8614F01**

> account: 0x4EA55452B992b81C95f2077c15C09aCa1c6Fdaed

> balance: 5.9556950899999997

> gas used: 395039

> gas price: 10 gwei

> value sent: 0 ETH

> total cost: 0.00395039 ETH

Deploying 'RetailerRole'

------------------------

> transaction hash: 0xb68e032cd8d3f4deb6a32cde3e63fccd4934aadf124963420dd909c3e58dba9f

> Blocks: 1 Seconds: 12

> contract address: **0x18453cfd2C0F813A07c74C7C5f7B07Bbc0fa88Cd**

> account: 0x4EA55452B992b81C95f2077c15C09aCa1c6Fdaed

> balance: 5.9517421399999997

> gas used: 395295

> gas price: 10 gwei

> value sent: 0 ETH

> total cost: 0.00395295 ETH

Deploying 'ConsumerRole'

------------------------

> transaction hash: 0xce2d7b68289ab2f39c953a4262f137d0d4230d1fd92f6638c175383060cd3ee2

> Blocks: 1 Seconds: 12

> contract address: **0x25800A361Fc36B39ee73CDa53c7ca3c7AD8e28B4**

> account: 0x4EA55452B992b81C95f2077c15C09aCa1c6Fdaed

> balance: 5.9477911099999997

> gas used: 395103

> gas price: 10 gwei

> value sent: 0 ETH

> total cost: 0.00395103 ETH

Deploying 'SupplyChain'

-----------------------

> transaction hash: 0xc1ffaba42fb8860c7955a116f50a3d841a0620f59f5fc52732fec25859e1b296

> Blocks: 1 Seconds: 12

> contract address: **0xB05E693d273E6BE70078E64581Ed371117bd6A11**

> account: 0x4EA55452B992b81C95f2077c15C09aCa1c6Fdaed

> balance: 5.9127371699999997

> gas used: 3505394

> gas price: 10 gwei

> value sent: 0 ETH

> total cost: 0.03505394 ETH

> Saving migration to chain.

> Saving artifacts

-------------------------------------

> Total cost: 0.05085806 ETH

Summary

=======

> Total deployments: 6

> Final cost: 0.05370714 ETH

## 2. Libraries Used

1. truffle-contract

* Ethereum contract abstraction for Node and the Browser
* In app.js truffle-contract method TruffleContract is used to create an instance of the SupplyChain contract. The input to this function is the JSON blob created in the build folder upon running `truffle compile`.
* By setting the provider of the SupplyChain contract as the newly created web3 instance, we now have access to all the contract methods at(), deployed(), and new()
* Calling the deployed() method and awaiting the returned instance of the SupplyChain contract, we are now able to leverage the contract methods including harvestItem, processItem, packItem etc.

1. truffle-hdwallet-provider
   * In order to deploy via Infura, a wallet provider is required to sign transactions before they’re sent to a remote node
   * In addition to the Infrua Rinkeby Testnet API key, the wallet provider requires a mnemonic (a 12 word phrase the wallet uses to generate public/private key pairs)

## 3. IPFS

* IPFS was not used in this assignment

## 4. Version Numbers

* Node: v10.13.0
* Truffle: v 5.0.3
* Web3: 1.2.2

## 5. General Write-Up

This project leverages the existing coffee supply chain starter code. The actors in this supply chain include:

* *Farmer*: The Farmer can harvest coffee beans, process coffee beans, pack coffee palettes, add coffee palettes, ship coffee palettes, and track authenticity.
* *Distributor*: The Distributor can buy coffee palettes and track authenticity.
* *Retailer*: The Retailer can receive coffee palettes and track authenticity.
* *Consumer*: The consumer can buy coffee palettes and track authenticity.

The data points tracked as the coffee beans move through the supply chain include:

* Product ID
* Product UPC
* Origination Information
* Origin Actor (e.g. Farmer ID, Farmer Name)
* Misc. organization information (e.g. Farmer Information)
* Longitude and Latitude of Origin Coordinates (e.g. Farm’s Longitude and Latitude)
* Product notes
* Product price

The functions required to be invoked for a coffee beans to move through the supply chain (from Farmer to Consumer) are illustrated in the following UML sequence diagram:

