This is a demo to demonstrate how to run a private ethereum blockchain on OCP 3.7.
The use case is based on the ‘Rewards Partner.’

Pre-req

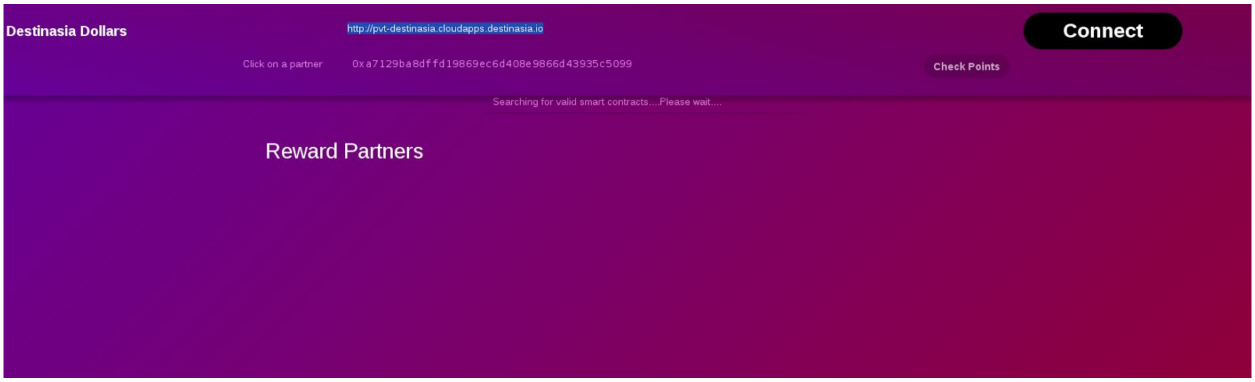
- OCP platform 3.7
- Nodejs v6 image

Install via the provided [Red Hat Demo Central project](), following readme instructions.

Running the demo:

Access the url of the nodejs (dapp) . e.g. http://dapp-ethe-demo.192.168.99.100.nip.io

You will see this :



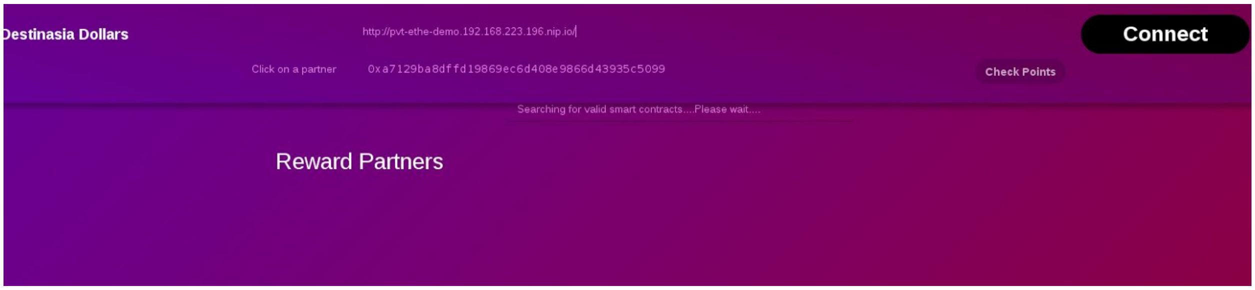
http://pvt-destinasia.cloudapps.destinasia.io

0xa7129ba8dffd19869ec6d408e9866d43935c5099

Searching for valid smart contracts....Please wait....

You will notice the url at the top points to a http://dapp-ethe-demo.192.168.99.100.nip.io this is supposed to be the endpoint of the blockchain,so if yours differs (see console output on install) replace it with yours.

I.e.:

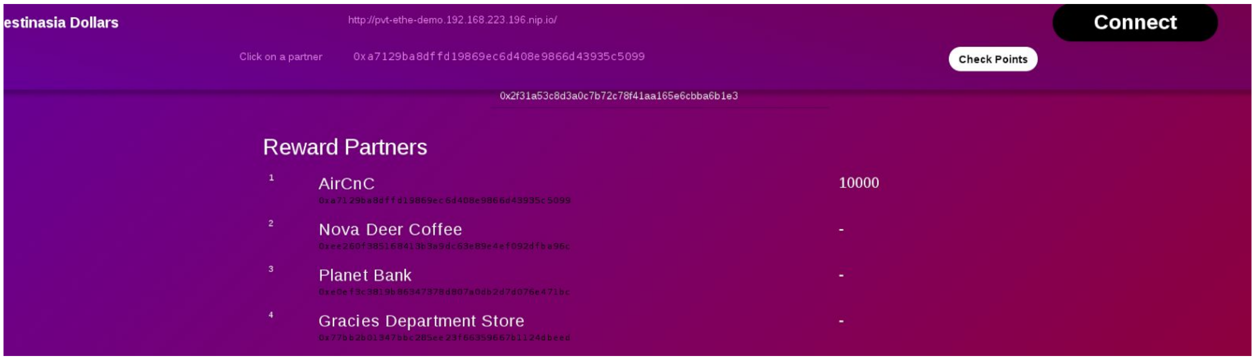


http://pvt-ethe-demo.192.168.223.196.nip.io/

0xa7129ba8dffd19869ec6d408e9866d43935c5099

Searching for valid smart contracts....Please wait....

Once you click connect,



0xa7129ba8dffd19869ec6d408e9866d43935c5099

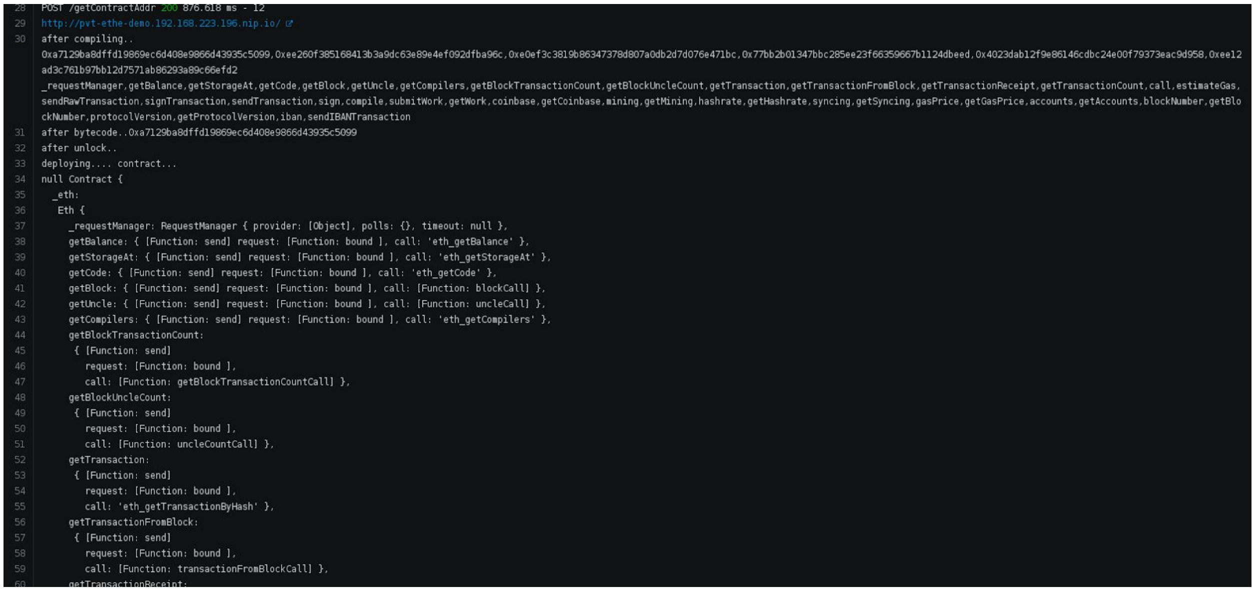
0x2f31a53c8d3a0c7b72c78f41aa165e6cbba6b1e3

Reward Partners

1	AirCnC 0xa7129ba8dffd19869ec6d408e9866d43935c5099	10000
2	Nova Deer Coffee 0xee260f385168413b3a9dc63e89e4ef092dfba96c	-
3	Planet Bank 0xe0ef3c3819b86347378d807a0db2d7d076e471bc	-
4	Gracies Department Store 0x77bb2b01347bbc285ee23f66359667b1124dbeed	-

You will notice:

- A list of ‘rewards partners’
- There will be a flash message to show that the app is trying to auto deploy a contract into the blockchain, and once that is done, the contract address will be displayed in the textbox above the “Reward Partners” label
- You can verified with the logs in the ethereum private chain pod (pvt) that a contract is created. Do note the logs are verbose so you may have to scroll up….



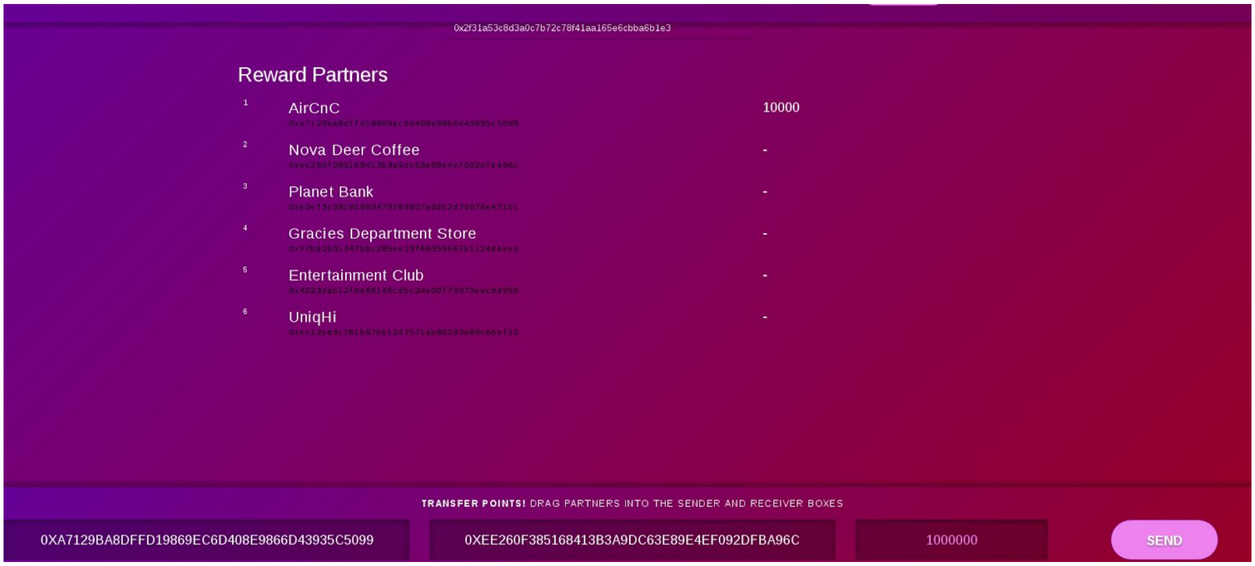
```
POST /getContractAddr 200 876.618 ms - 12
http://pvt-ethe-demo.192.168.223.196.nip.io/
after compiling..
0xa7129ba8dffd19869ec6d408e9866d43935c5099,0xee260f385168413b3a9dc63e89e4ef092dfba96c,0xe0ef3c3819b86347378d807a0db2d7d076e471bc,0x77bb2b01347bbc285ee23f66359667b1124dbeed,0x4023dab12f9e86146cdbc24e00f79373eac9d958,0xee12
ad3c761b97bb12d7571ab86293a89c66efd2
_requestManager,getBalance,getStorageAt,getCode,getBlock,getUncle,getCompilers,getBlockTransactionCount,getBlockUncleCount,getTransaction,getTransactionFromBlock,getTransactionReceipt,getTransactionCount,call,estimateGas,
sendRawTransaction,signTransaction,sendTransaction,sign,compile,submitWork,getWork,coinbase,getCoinbase,mining,getMining,hashrate,getHashrate,syncing,getSyncing,gasPrice,getGasPrice,accounts,getAccounts,blockNumber,getBlo
ckNumber,protocolVersion,getProtocolVersion,iban,sendIBANTransaction
after bytecode..0xa7129ba8dffd19869ec6d408e9866d43935c5099
after unlock..
deploying.... contract...
null Contract {
  _eth:
   Eth {
     _requestManager: RequestManager { provider: [Object], polls: {}, timeout: null },
     getBalance: { [Function: send] request: [Function: bound ], call: 'eth_getBalance' },
     getStorageAt: { [Function: send] request: [Function: bound ], call: 'eth_getStorageAt' },
     getCode: { [Function: send] request: [Function: bound ], call: 'eth_getCode' },
     getBlock: { [Function: send] request: [Function: bound ], call: [Function: blockCall] },
     getUncle: { [Function: send] request: [Function: bound ], call: [Function: uncleCall] },
     getCompilers: { [Function: send] request: [Function: bound ], call: 'eth_getCompilers' },
     getBlockTransactionCount:
      { [Function: send]
        request: [Function: bound ],
        call: [Function: getBlockTransactionCountCall] },
     getBlockUncleCount:
      { [Function: send]
        request: [Function: bound ],
        call: [Function: uncleCountCall] },
     getTransaction:
      { [Function: send]
        request: [Function: bound ],
        call: 'eth_getTransactionByHash' },
     getTransactionFromBlock:
      { [Function: send]
        request: [Function: bound ],
        call: [Function: transactionFromBlockCall] },
     getTransactionReceipt:
```

Click on ‘Check Point”, the app will return the points allocated to the account number ,e.g. “0xa7129ba8dffd19869ec6d408e9866d43935c5099” which is to the left of “check point”

By default , 10000 points is allocated to the first partner ‘AirCnC’

To check the points of another partner, click on the name, it’s account number will be updated in the “check point” text box

You can drag and drop the partner account numbers into the ‘Sender’ and ‘Receiver’ boxes at the bottom, put in a number and click send



0x2f31a53c8d3a0c7b72c78f41aa165e6cbba6b1e3

Reward Partners

1	AirCnC 0xa7129ba8dffd19869ec6d408e9866d43935c5099	10000
2	Nova Deer Coffee 0xee260f385168413b3a9dc63e89e4ef092dfba96c	-
3	Planet Bank 0xe0ef3c3819b86347378d807e0db2d7d076e471bc	-
4	Gracies Department Store 0x77bb2b01347bbc285ee23f66359667b1124dbeed	-
5	Entertainment Club 0x4023dab12f9e86146cdbc24e00f79373eac9d958	-
6	UniqHi 0xee12ad3c761b97bb12d7571ab86293a89c66efd2	-

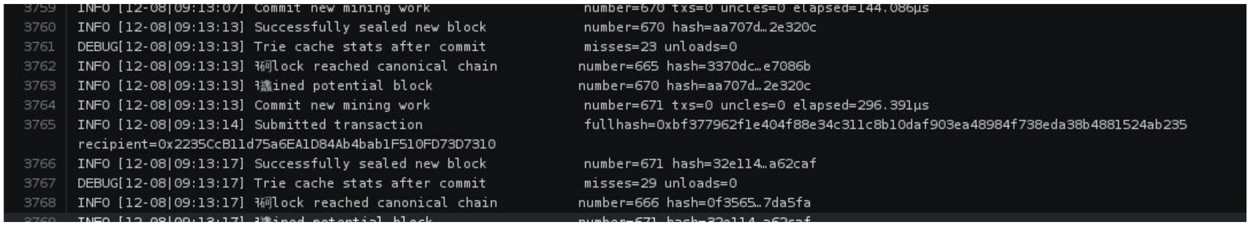
TRANSFER POINTS! DRAG PARTNERS INTO THE SENDER AND RECEIVER BOXES

0XA7129BA8DFFD19869EC6D408E9866D43935C5099

0XEE260F385168413B3A9DC63E89E4EF092DFBA96C

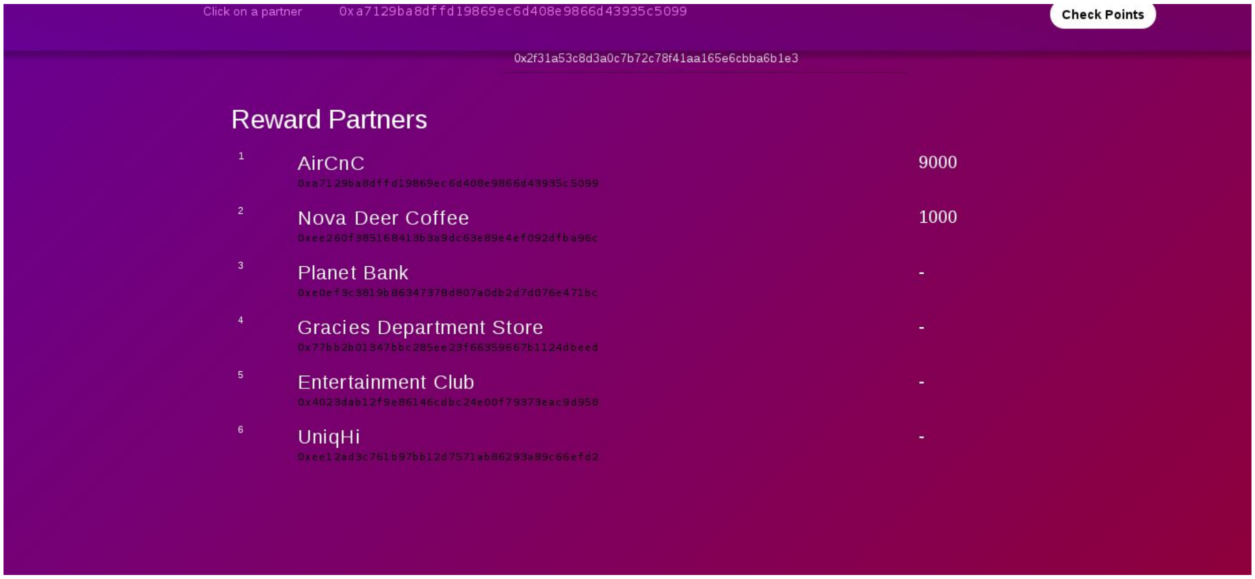
1000000

You should be able to catch the logs indicating a transaction being processed.



```
INFO [12-08|09:13:07] Commit new mining work                number=670 txs=0 uncles=0 elapsed=144.086µs
INFO [12-08|09:13:13] Successfully sealed new block         number=670 hash=aa707d…2e320c
DEBUG[12-08|09:13:13] Trie cache stats after commit         misses=23 unloads=0
INFO [12-08|09:13:13] 衲lock reached canonical chain       number=665 hash=3370dc…e7086b
INFO [12-08|09:13:13] 譴ined potential block               number=670 hash=aa707d…2e320c
INFO [12-08|09:13:13] Commit new mining work                number=671 txs=0 uncles=0 elapsed=296.391µs
INFO [12-08|09:13:14] Submitted transaction                 fullhash=0xbf377962f1e404f88e34c311c8b10daf903ea48984f738eda38b4881524ab235
recipient=0x2235CcB11d75a6EA1D84Ab4bab1F510FD73D7310
INFO [12-08|09:13:17] Successfully sealed new block         number=671 hash=32e114…a62caf
DEBUG[12-08|09:13:17] Trie cache stats after commit         misses=29 unloads=0
INFO [12-08|09:13:17] 衲lock reached canonical chain       number=666 hash=0f3565…7da5fa
INFO [12-08|09:13:17] 譴ined potential block               number=671 hash=32e114…a62caf
```

Depending on the time taken to process the transaction, after a brief moment, the points will be updated in the respective accounts (you need to manually click check points for every account) to get the updated points.


Click on a partner
0xa7129ba8dffd19869ec6d408e9866d43935c5099
Check Points
0x2f31a53c8d3a0c7b72c78f41aa165e6cbba6b1e3

Reward Partners

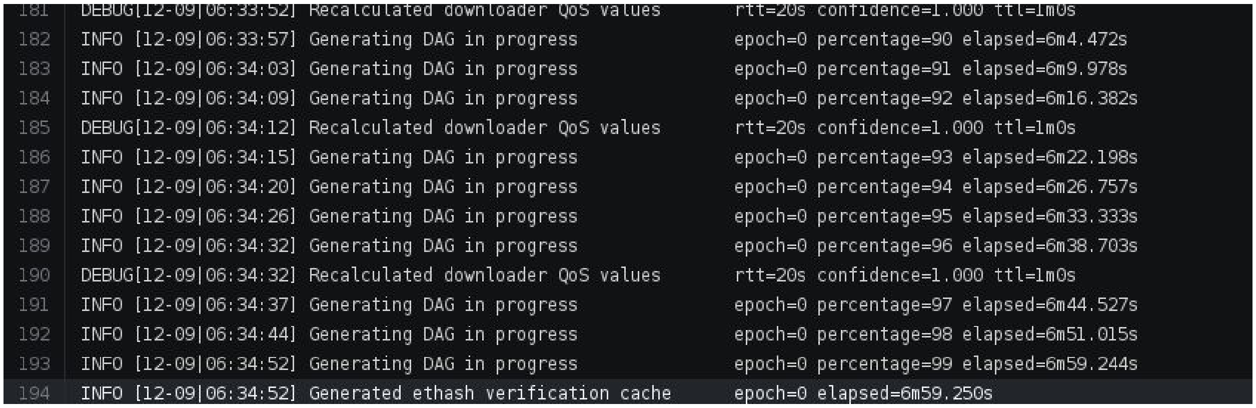
1	AirCnC 0xa7129ba8dffd19869ec6d408e9866d43935c5099	9000
2	Nova Deer Coffee 0xee260f385168413b3a9dc63e89e4ef092dfba96c	1000
3	Planet Bank 0xe0ef3c3819b86347378d807a0db2d7d076e471bc	-
4	Gracies Department Store 0x77bb2b01347bbc285ee23f66359667b1124dbeed	-
5	Entertainment Club 0x4023dab12f9e86146cdbc24e00f79373eac9d958	-
6	UniqHi 0xee12ad3c761b97bb12d7571ab86293a89c66efd2	-

This concludes the Demo

======Gotchas===============

1. DAG generation

If the blockchain is restarted (e.g. OCP instance restarts), you will need to wait a while for the blockchain to stabilized, one way is to take note of the log files
“Generating DAG in progress” with a “epoch=0 **percentage=...**”
You have to wait till the progress is completed, reaching 100 percent before



```
DEBUG[12-09|06:33:52] Recalculated downloader QoS values       rtt=20s confidence=1.000 ttl=1m0s
INFO [12-09|06:33:57] Generating DAG in progress               epoch=0 percentage=90 elapsed=6m4.472s
INFO [12-09|06:34:03] Generating DAG in progress               epoch=0 percentage=91 elapsed=6m9.978s
INFO [12-09|06:34:09] Generating DAG in progress               epoch=0 percentage=92 elapsed=6m16.382s
DEBUG[12-09|06:34:12] Recalculated downloader QoS values       rtt=20s confidence=1.000 ttl=1m0s
INFO [12-09|06:34:15] Generating DAG in progress               epoch=0 percentage=93 elapsed=6m22.198s
INFO [12-09|06:34:20] Generating DAG in progress               epoch=0 percentage=94 elapsed=6m26.757s
INFO [12-09|06:34:26] Generating DAG in progress               epoch=0 percentage=95 elapsed=6m33.333s
INFO [12-09|06:34:32] Generating DAG in progress               epoch=0 percentage=96 elapsed=6m38.703s
DEBUG[12-09|06:34:32] Recalculated downloader QoS values       rtt=20s confidence=1.000 ttl=1m0s
INFO [12-09|06:34:37] Generating DAG in progress               epoch=0 percentage=97 elapsed=6m44.527s
INFO [12-09|06:34:44] Generating DAG in progress               epoch=0 percentage=98 elapsed=6m51.015s
INFO [12-09|06:34:52] Generating DAG in progress               epoch=0 percentage=99 elapsed=6m59.244s
INFO [12-09|06:34:52] Generated ethash verification cache      epoch=0 elapsed=6m59.250s
```

2. Scaling of client app

In this demo, the contract only exist on the client side, scaling up or refreshing the browser will generate a new contract key. This is a limitation of the demo app.
Scaling up of private chain is ok

3. Administrating the blockchain

The blockchain is configured to generate ether at a very high frequency (for demo purposes) so if you want to pause the mining process, you can rsh into the pod
E.g. to stop / start the mining

```
oc rsh <private chain pod>
/ #  geth attach rpc:http://127.0.0.1:8545
Welcome to the Geth JavaScript console!

instance: Geth/v1.7.0-unstable/linux-amd64/go1.8.3
coinbase: 0xa7129ba8dffd19869ec6d408e9866d43935c5099
at block: 388 (Sat, 09 Dec 2017 07:56:35 UTC)
 datadir: /root/.ethereum
 modules: admin:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0
txpool:1.0 web3:1.0

> miner.stop()
true
> miner.start()
null
```

If you have a geth library on your machines, you can also geth attach via the route