Private network

- Private Network Setup

Discount Coupon Links to UDEMY courses:

https://www.udemy.com/hyperledger/?couponCode=DKHLF1099

https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099

https://www.udemy.com/rest-api/?couponCode=DKRST1099

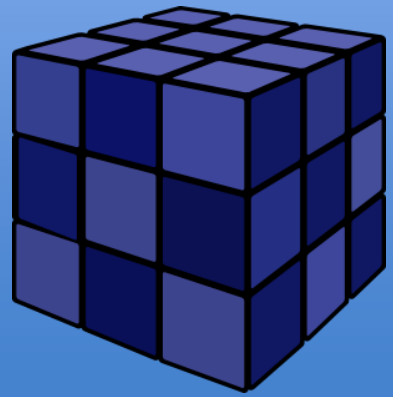
mentoring, seeking Blockchain part time work, project guidance, advice … …
http://www.bcmentors.com

This deck is part of a online course on "Ethereum: Design and Development of Decentralized Apps.

raj@acloudfan.com

@acloudfan

http://ACloudFan.com



Motivation

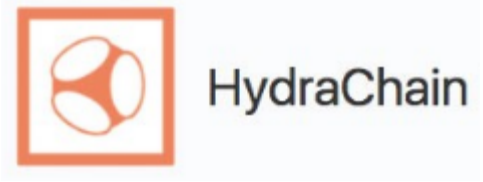
- Business Use Case
- Development of contracts (DevOps)
- Consortium
- Experimentation

Considerations

- No public access to the chain
- Peers are restricted to known entities (nodes)
- Chain need to be Permissioned
- Proof of Work is NOT the preferred consensus model
- Transaction Speed & Fees

Examples

MONAX



Creating a Private Network

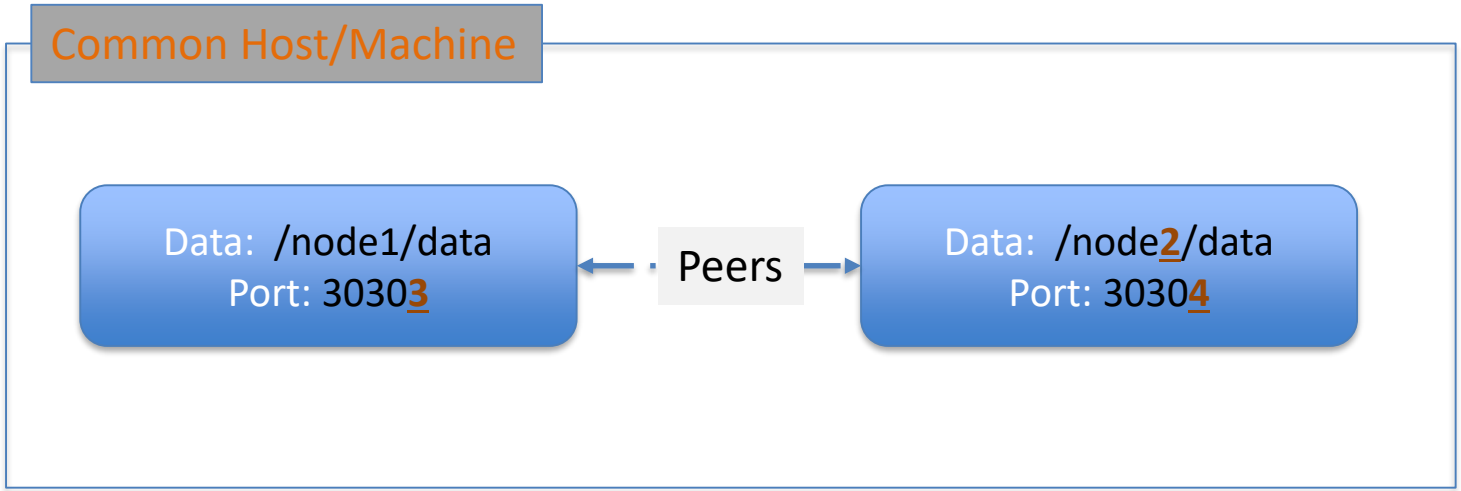
1. Single Node for experimentation 

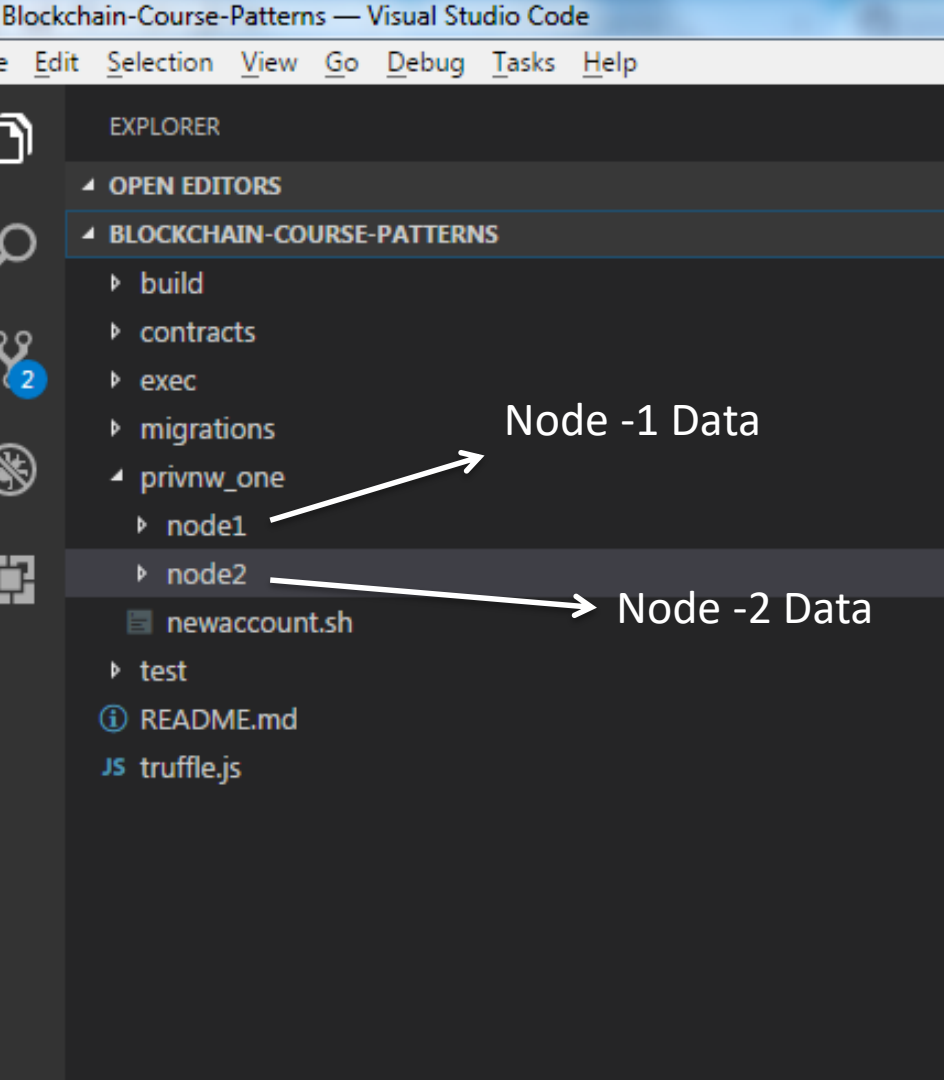
   > geth --dev

2. Multi Node on single Host (machine/server)

3. Multi Node on multiple hosts (Amazon EC2)

Demo Setup


Common Host/Machine
Data: /node1/data
Port: 30303
Peers
Data: /node2/data
Port: 30304



/acloudfan/Blockchain-Course-Patterns

Private Chain

1. Create an account : node1

2. Setup *genesis.json*

3. Initialize the chain on 2 nodes

4. Add as peer

Private Network

- Setting up a private node is a 4 step process

1. Create an account : node1

> geth --datadir ".." account new

2. Setup *genesis.json*

- Set chain ID
- Allocate to account

3. Initialize the chain on 2 nodes

> geth --datadir ".." init *genesis.json*

4. Add as peer

> admin.nodeInfo.enode

> admin.addPeer(…enode url….)

- Setup <datadir>/static-nodes.json

Bootnode tool

raj@acloudfan.com

@acloudfan

http://ACloudFan.com

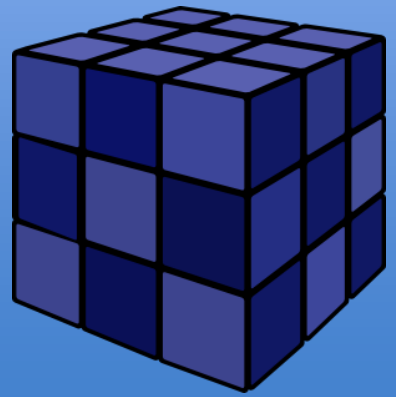
Discount Coupon Links to UDEMY courses:

https://www.udemy.com/hyperledger/?couponCode=DKHLF1099

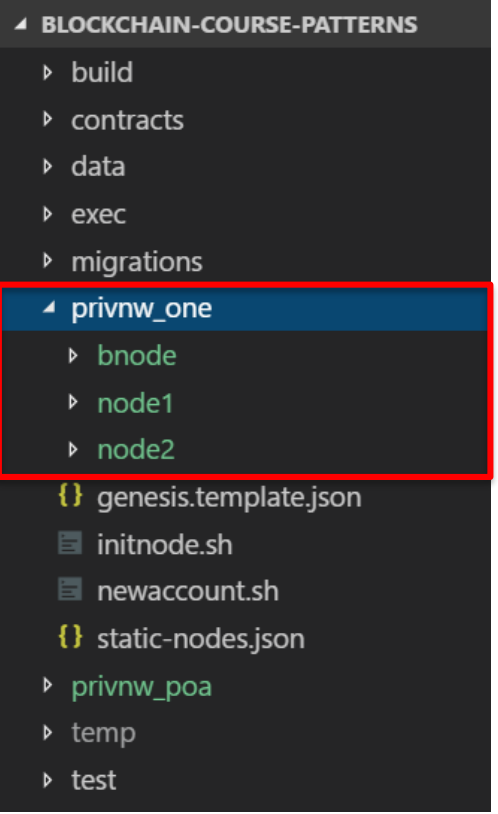
https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099

https://www.udemy.com/rest-api/?couponCode=DKRST1099

mentoring, seeking Blockchain part time work, project guidance, advice … …
http://www.bcmentors.com



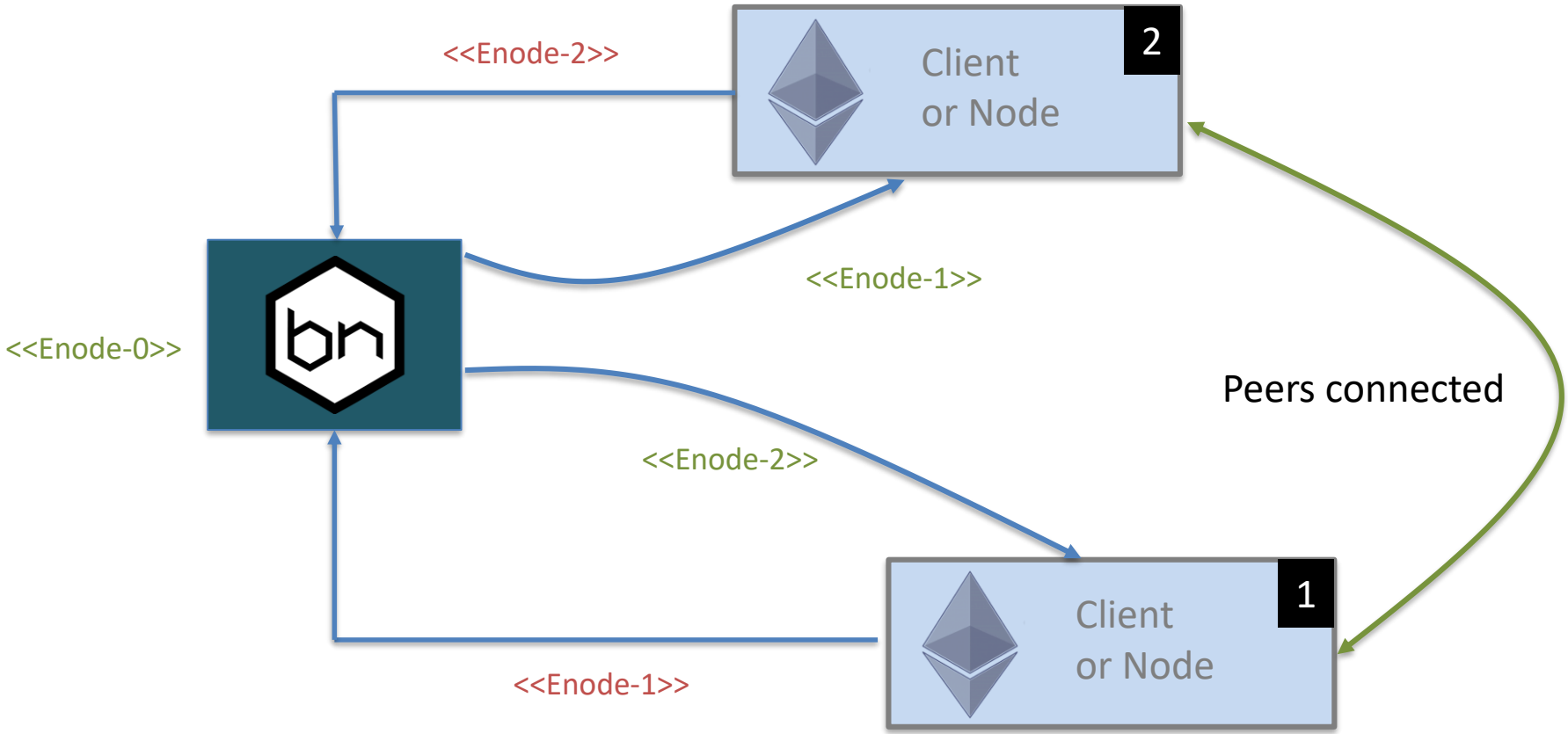
This deck is part of a online course on "Ethereum: Design and Development of Decentralized Apps.



https://github.com/acloudfan/**Blockchain-Course-Patterns**

Geth Tool : Available with v1.6 and above

- Stripped down version of *Geth*
  - Has only the discovery protocol implemented
  - Sole responsibility is to assist nodes to discover peers
  - Publicly available IP Address


<<Enode-2>>
Client
or Node
2
<<Enode-1>>
<<Enode-0>>
bn
Peers connected
<<Enode-2>>
Client
or Node
1
<<Enode-1>>

Geth Tool : Available with v1.6 and above

Best Practices:

- For private network use Bootnode instead of manually adding the peers
- Setup multiple Bootnode processes to avoid *Single Point of Failure*
- *Bootstrapping:* Use Bootnode instead of full blown geth client

bootnode --help

Address on which Boonode is listens

Key identifies the node

```
-addr string
      listen address (default ":30301")
-genkey string
      generate a node key
-nat string
      port mapping mechanism (any|none|upnp|pmp|extip:<IP>) (default "none")
-netrestrict string
      restrict network communication to the given IP networks (CIDR masks)
-nodekey string
      private key filename
-nodekeyhex string
      private key as hex (for testing)
-v5
      run a v5 topic discovery bootnode
-verbosity int
      log verbosity (0-9) (default 3)
-vmodule string
      log verbosity pattern
-writeaddress
      write out the node's pubkey hash and quit
```

Specifies the key used by the Bootnode

Versbosity

Bootnode setup walk through

Setup the *Bootnode* key

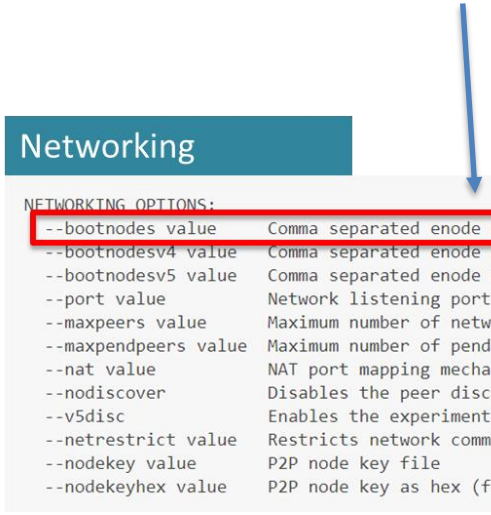
```
>_  bootnode -genkey boot.key
```

Launch using the key generated

```
>_  bootnode -nodekey boot.key
```

Take note of the <<enode>> info

Setup the *peers* to use Bootnode



Proof of Authority

- Setup private network

Discount Coupon Links to UDEMY courses:

https://www.udemy.com/hyperledger/?couponCode=DKHLF1099

https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099

https://www.udemy.com/rest-api/?couponCode=DKRST1099

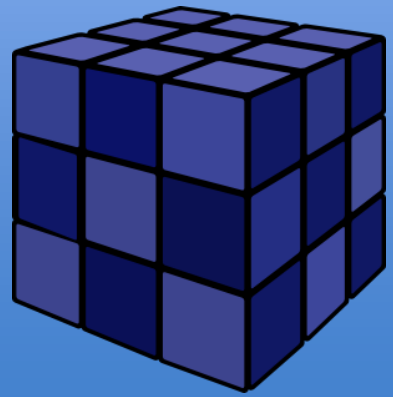
mentoring, seeking Blockchain part time work, project guidance, advice … …
http://www.bcmentors.com

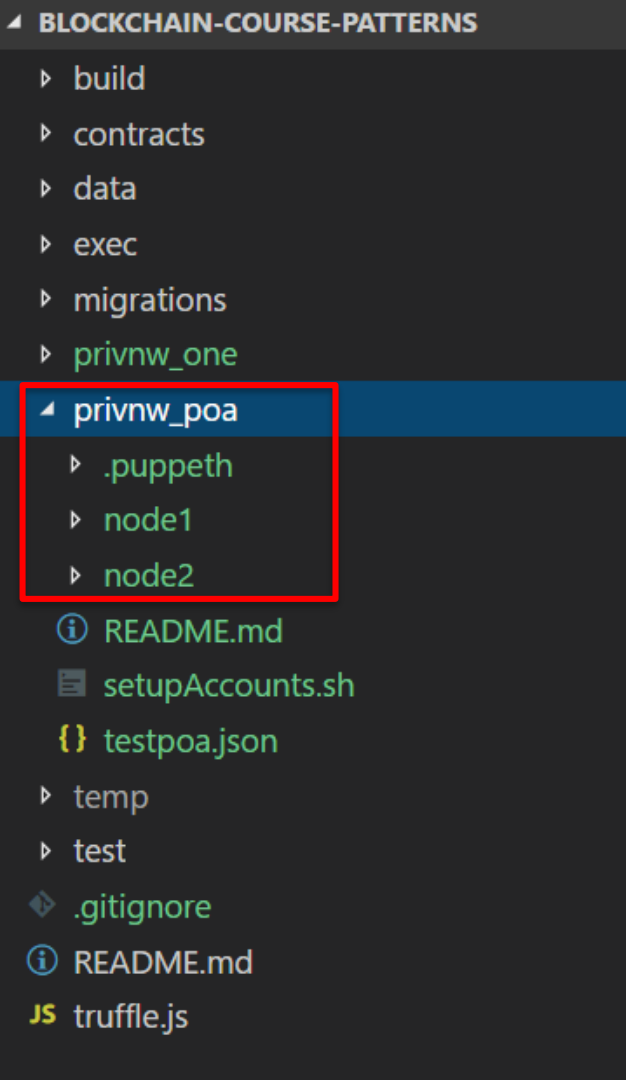
This deck is part of a online course on "Ethereum: Design and Development of Decentralized Apps.

raj@acloudfan.com

@acloudfan

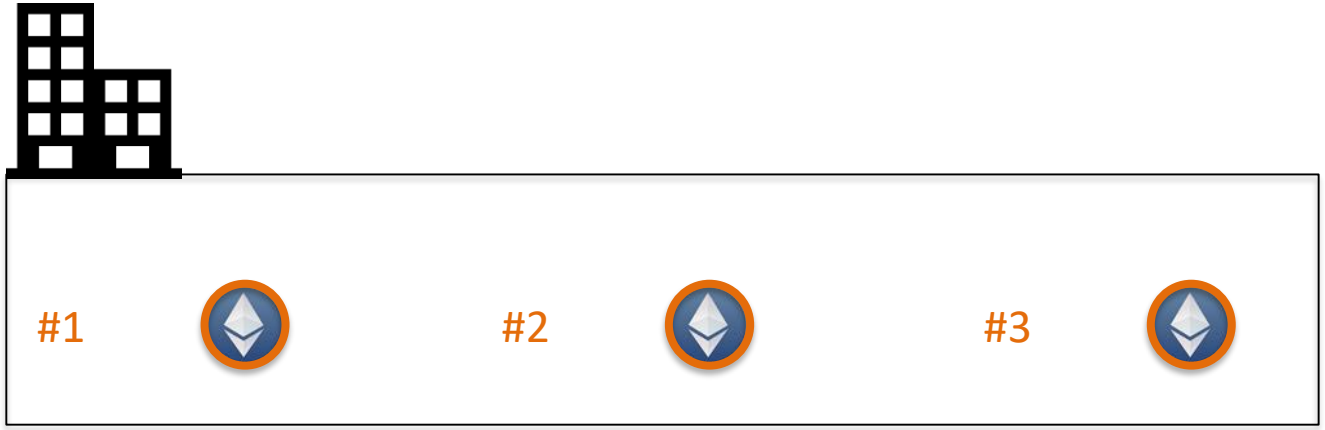
http://ACloudFan.com



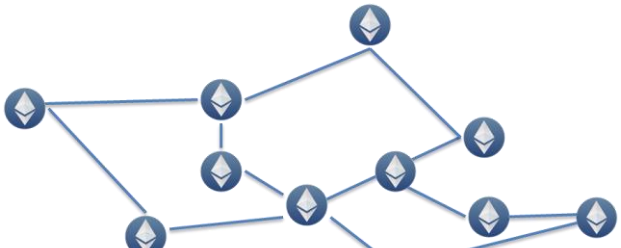


https://github.com/acloudfan/**Blockchain-Course-Patterns**

CLIQUE



- No Block rewards
- Block time is configurable
- Sealers can be distributed across multiple orgs



CLIQUE

- Sealers added | removed by way of voting

```
clique.propose("0x...", true | false)
```

  - More than half votes needed

Walkthrough

STEP 1 Setup accounts on node-1 & node-2

STEP 2 Setup genesis block using Puppeth 

STEP 3 Initialize geth node 1 & 2

STEP 4 Setup launch commands for Node 1 & Node 2

Puppeth

Command line tool for managing private network

- Available in Geth v1.6 and above

```
>_ puppeth

What would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
```


```
What would you like to deploy? (recommended order)
 1. Ethstats  - Network monitoring tool
 2. Bootnode  - Entry point of the network
 3. Sealer    - Full node minting new blocks
 4. Explorer  - Chain analysis webservice (ethash only)
 5. Wallet    - Browser wallet for quick sends
 6. Faucet    - Crypto faucet to give away funds
 7. Dashboard - Website listing above web-services
```

Puppeth

- Recommendation from developers
  - OK for small scale setup
  - *Enterprise setup:* Use existing orchestration tools

Walkthrough

Initialize geth node 1 & 2

Node -1 Subfolder

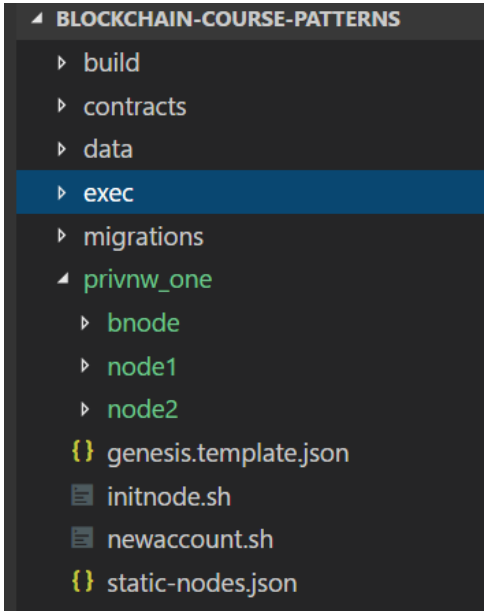
```
>_ geth --datadir ./data init  ../testpoa.json
```

Node -2 Subfolder

```
>_ geth --datadir ./data init  ../testpoa.json
```

Walkthrough

Launch Bootnode & setup <<enode>> on node 1 & 2



- Use the bootnode setup discussed in last lecture

Walkthrough

Setup launch commands for Node 1 & Node 2

```
geth --networkid 1015 --datadir "./data“ --port 30303
--bootnodes '<<enode>>’
--rpc
--rpcport 8545   ----rpccorsdomain "*“
--rpcapi "web3,eth,net,personal“
console
```