How to Setup Ethereum Blockchain in AWS Cloud using Ubuntu EC2 Box

Step 0

Launch at least two EC2 Ubuntu 16.04 Servers on AWS

You need to launch at least two EC2 Ubuntu 16.04 LTS or 14.04 LTS servers in AWS. Please open the inbound ports 30301-30399 in the security group.

Step 1

SSH

SSH into the All Nodes (ie. Node 1 & Node 2)

```
ssh –i ethereum.pem ubuntu@<Server’s public IP>
```

Step 2

Install Geth client in all nodes

On all servers, Install Geth (ie. Node 1 & Node 2)

```
sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install ethereum
```

After installation make sure all the servers have geth properly install by running this commans.

```
geth –help
```

you must see a help page of geth command.

Step 3

Initialize Chain Data in Node 1 using Genesis Block

On Node 1, Initialize Genesis Block in genesis.json inside /home/ubuntu/eth-dev/ directory.

```
cd ~
mkdir eth-dev
cd eth-dev
nano genesis.json
```

Paste the below line into the nano's window.

```
{
"config": {
        "chainId": 45634,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
"coinbase" : "0x0000000000000000000000000000000000000000",
"difficulty" : "0x20000",
"extraData" : "",
"gasLimit" : "0x2fefd8",
"nonce" : "0x0000000000000042",
"mixhash" : "0x0000000000000000000000000000000000000000000000000000000000000000",
"parentHash" : "0x0000000000000000000000000000000000000000000000000000000000000000",
"timestamp" : "0x00",
"alloc": {}
}
```

Initialize the chain data with below command:

```
geth --datadir ~/eth-dev/ init genesis.json
```

Start the Ethereum Node 1 with the below command

```
geth --datadir ~/eth-dev/ --networkid 45634 --verbosity 3 --
ipcdisable --rpc --port 30301 --rpcport 8545 --rpcaddr <Public
DNS address of your Node 1> console 2>> ~/eth-dev/eth.log
```

Here the **<Public DNS address of your Node 1>** must be something like **ec2-54-145-45-122.compute-1.amazonaws.com** without quotes. This will become your JSON-RPC end point address to send any JSON-RPC Request to perform any action on Ethereum Blockchain.

You will see the geth command line interface where you can execute the commands.

If you want to see the logs created by this command, then open a new terminal SSH into the Node-1 Server & execute the below command:

```
tail -F ~/eth-dev/eth.log
```

You will see the logs created by the geth client.

Step 4

Initialize Chain Data in node 2

On Node 2, Initialize Genesis Block in genesis.json inside /home/ubuntu/eth-dev/ directory.

```
cd ~
```

```
mkdir eth-dev
cd eth-dev
nano genesis.json
```

Paste the below line into the nano's window.

```
{
"config": {
        "chainId": 45634,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
"coinbase" : "0x0000000000000000000000000000000000000000",
"difficulty" : "0x20000",
"extraData" : "",
"gasLimit" : "0x2fefd8",
"nonce" : "0x0000000000000042",
"mixhash" : "0x0000000000000000000000000000000000000000000000000000000000000000",
"parentHash" : "0x0000000000000000000000000000000000000000000000000000000000000000",
"timestamp" : "0x00",
"alloc": {}
}
```

Initialize the chain data with below command:

```
geth --datadir ~/eth-dev/ init genesis.json
```

Start the Ethereum **Node 2** with the below command

```
geth --datadir ~/eth-dev/ --networkid 45634 --verbosity 3 --
ipcdisable --rpc --port 30302 --rpcport 8546 console 2>> ~/eth-
dev/eth.log
```

Step 5

Mine Blocks in Node 1 & Check balance

On Node 1, start the miner using the below command:

```
miner.start()
```

if it says you don't have any coinbase or primary address to start the mining then you need to create a new account/address which will work as a coinbase (default account where mined ether will be collected). To create a new account, follow the below instructions:

Enter the below command in the geth console. It will ask for the password. DO NOT LOSE THIS PASSWORD IN ANY CASE.:

```
personal.newAccount()
Passphrase:
Repeat passphrase:
```

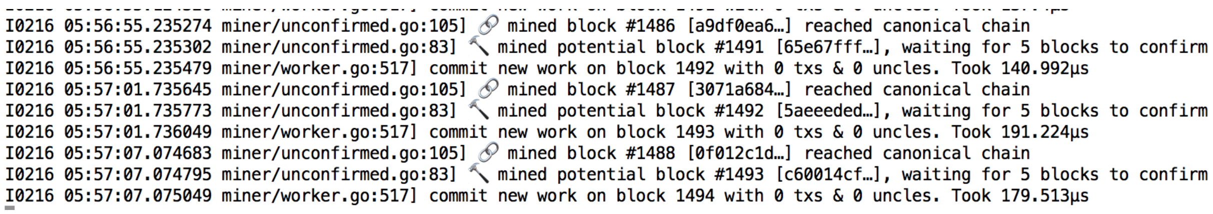
```
"0xae0a0defea5a3f9b94c671dac07389439c3f17dc"
```

This green color string (without quote) will be the Coinbase address (if it is the first account). Copy this Coinbase address and keep it safe you will need it multiple times.

Now you can start the mining in node 1. using the below command:

```
miner.start()
```

if you are monitoring the eth.log file on this node you will see the below text in the terminal. The block number will be from #1, #2, #3 etc…



```
I0216 05:56:55.235274 miner/unconfirmed.go:105] 🔗 mined block #1486 [a9df0ea6…] reached canonical chain
I0216 05:56:55.235302 miner/unconfirmed.go:83] ⛏ mined potential block #1491 [65e67fff…], waiting for 5 blocks to confirm
I0216 05:56:55.235479 miner/worker.go:517] commit new work on block 1492 with 0 txs & 0 uncles. Took 140.992µs
I0216 05:57:01.735645 miner/unconfirmed.go:105] 🔗 mined block #1487 [3071a684…] reached canonical chain
I0216 05:57:01.735773 miner/unconfirmed.go:83] ⛏ mined potential block #1492 [5aeeeded…], waiting for 5 blocks to confirm
I0216 05:57:01.736049 miner/worker.go:517] commit new work on block 1493 with 0 txs & 0 uncles. Took 191.224µs
I0216 05:57:07.074683 miner/unconfirmed.go:105] 🔗 mined block #1488 [0f012c1d…] reached canonical chain
I0216 05:57:07.074795 miner/unconfirmed.go:83] ⛏ mined potential block #1493 [c60014cf…], waiting for 5 blocks to confirm
I0216 05:57:07.075049 miner/worker.go:517] commit new work on block 1494 with 0 txs & 0 uncles. Took 179.513µs
```

you can stop the mining using the below command anytime you want:

```
miner.stop()
```

since we have the difficulty level low, If you mine for few seconds or minutes you will get some ether mined in your coinbase account.

To check the balance of your Coinbase account address use the below command. This command will show the 0th account balance in ether. 0th account is the Coinbase account.

```
web3.fromWei(web3.eth.getBalance(web3.eth.accounts[0]), 'ether')
>8756.7890
```

Step 6

Connect Node 2 with Node 1 in Private Ethereum Blockchain Network

On Node 2, get the enode address of the node so that we can connect it with node 1.

```
admin.nodeInfo.enode
```

this will give the long address something like given below:

```
"enode://e89e2457e4c8349919a013c4daab3e93660e3bf3782f83ed6445c52
2423b287c002500cc5641b5b50aeff63fa3e685cba95156a3de29f7ba8c01fe7
ecd57a100@[::]:30302"
```

[Please use your won address] this address is structured as enode://public_key@IP-Address:Port. Here you need to replace [::] with the private IP of the node 2 (if it is on the private node otherwise public address). Finally you enode address will look like this below

```
"enode://e89e2457e4c8349919a013c4daab3e93660e3bf3782f83ed6445c52
2423b287c002500cc5641b5b50aeff63fa3e685cba95156a3de29f7ba8c01fe7
ecd57a100@172.31.49.166:30302"
```

Important: Please use your own enode address.

Now, go to node 1 & execute this below command:

```
admin.addPeer("enode://e89e2457e4c8349919a013c4daab3e93660e3bf37
82f83ed6445c522423b287c002500cc5641b5b50aeff63fa3e685cba95156a3d
e29f7ba8c01fe7ecd57a100@172.31.49.166:30302")
```

If you see "true" after executing this command, then it means you have added the node 2 as a peer to node 1.

You can now check the peers details like this:

```
net.peerCount
1
```

or this below command:

```
admin.peers
[{
    caps: ["eth/62", "eth/63"],
    id:
"98ce62608390dededd9ae08f6c97ac13bc4850cc96d3ed5946e6a9ebb4ab61a
5e684b8d5c485898f87c07d596674225e30f5a49e87a710494945bb437824333
f",
    name: "Geth/v1.5.9-stable-a07539fb/linux/go1.7.3",
    network: {
      localAddress: "172.31.49.166:30302",
      remoteAddress: "172.31.61.150:35897"
    },
    protocols: {
      eth: {
        difficulty: 411258110,
        head:
"0xee2eb615c9ad394c3cee6edc55b27402f4e49484ff9447a44ba14aad1bf19
5ec",
        version: 63
      }
    }
}]
```

Step 7

Start Mining on Node 2

On Node 2, you can enable the mining like this below and start the mining.

```
miner.start()
```

it will show the error saying that coinbase need to be specified explicitly.

```
miner.setEtherbase("0x9690c9ceb9a900a3a55956c5315580ad615358d2")
```

Please replace "0x9690c9ceb9a900a3a55956c5315580ad615358d2" with your own Coinbase address.

now you can start the mining easily.

```
miner.start()
```

and then you can stop the mining with the below command

```
miner.stop()
```

and can also check the balance on **Node 1** to ensure that it's balance is increasing even though **Node 1** is not mining.

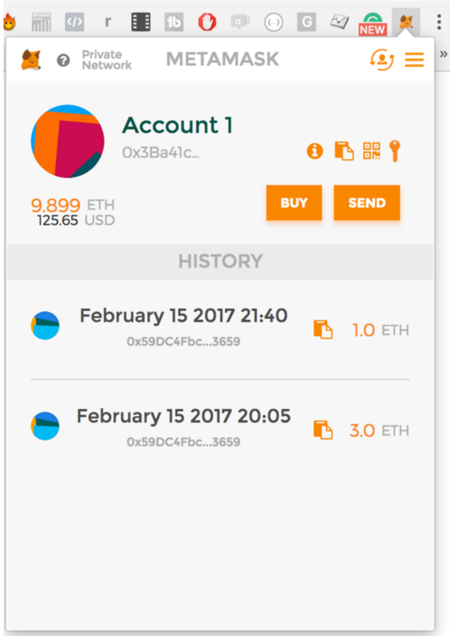
to check the balance:

```
web3.fromWei(web3.eth.getBalance(web3.eth.accounts[0]), 'ether')
```

Step 8

Transfer ether to an address

In your chrome browser, Install MetaMask extension which let you connect to Ethereum network and manage the accounts & balance right inside your chrome extension.



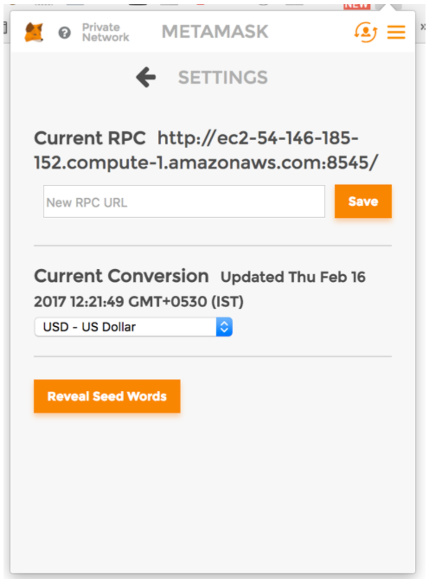
Account 1
0x3Ba41c...

9.899 ETH
125.65 USD

February 15 2017 21:40
0x59DC4Fbc...3659
1.0 ETH

February 15 2017 20:05
0x59DC4Fbc...3659
3.0 ETH

Click on top left corner where it says Private Network & paste the JSON-RPC URL of Node 1 which we copied in step 3 and looks something like this **ec2-54-145-45-122.compute-1.amazonaws.com.**



Current RPC http://ec2-54-146-185-152.compute-1.amazonaws.com:8545/

Current Conversion Updated Thu Feb 16 2017 12:21:49 GMT+0530 (IST)
USD - US Dollar

This will create a new wallet & add one account inside it. you can create as many account as you want within this window. Copy the Account 1’s address because we are going to transfer 100 Ether into this account from our Coinbase account from Node 1.

Now go to Node 1’s geth console and type this command

```
eth.sendTransaction({from:eth.coinbase, to:”<MetaMask account
address you just copied>”, value: web3.toWei(100, "ether")})
```

above command will return an error saying that account need to be unlocked and to do this you need to execute this below command.

```
personal.unlockAccount(eth.coinbase, "<whatever your Coinbase
password is>")
```

Now try to send 100 ether again to MetaMask account using the same command.

```
eth.sendTransaction({from:eth.coinbase, to:"<MetaMask account
address you just copied>", value: web3.toWei(100, "ether")})
```

Now this will return a transaction ID.

Important: To see the balance toped up to 100 ether in MetaMask at least one of the node must be mining the floating transaction. Here we are considering that Node 2 is mining the Block. If all is set, then MetaMask will show the 100 Ether in the copied account.

You can now transfer any amount of the available ether into any other account using MetaMask interface.

The End