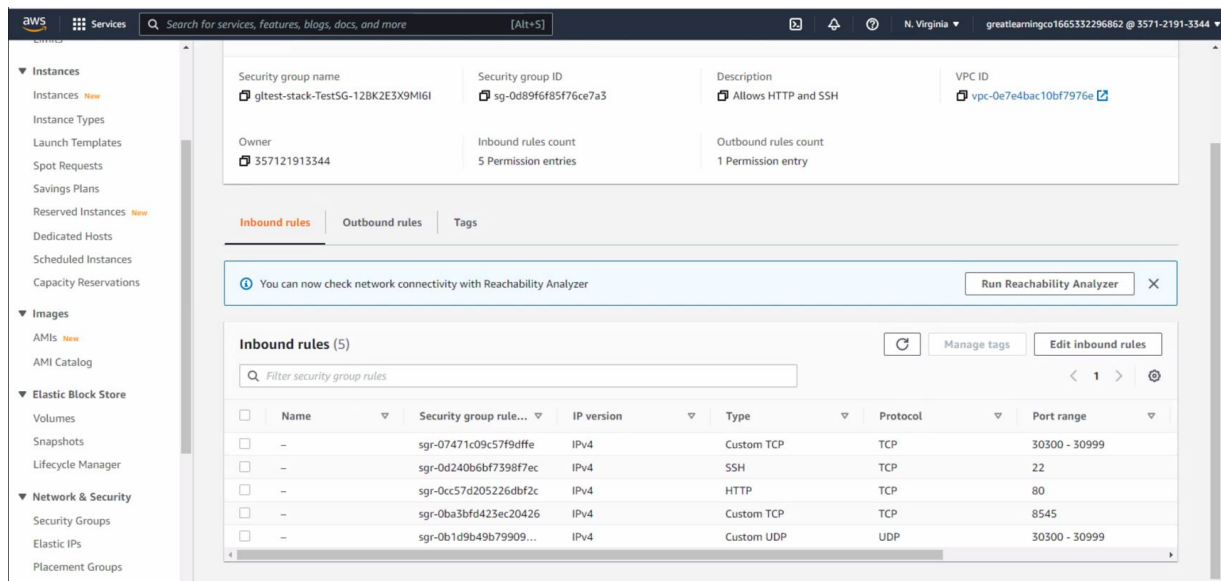


# Capstone project – Blockchain - Group 5

1. Created 3 ec2 machines (Ubuntu)
2. When all three ec2 machines are running SSH to all the three machines.
3. Updated security group which is common for all the three machines as given in the screenshot:



4. Installed Ethereum on all the three machines by using command given below:  

```
$ sudo apt-get install software-properties-common  
$ sudo add-apt-repository -y ppa:ethereum/ethereum  
$ sudo apt-get update  
$ sudo apt-get install ethereum
```
5. Our objective here is to create one bootnode and three nodes (node1, node2 & node3) on three different ec2 machines.
  - 1) on ec2(machine2)
    - a. Create a node2 folder using **mkdir node2** command
    - b. Go to the node2 folder using **cd node2** command
    - c. Run the command **geth --datadir ./data account new**
    - d. and save public address to your notepad

- 2) on ec2(machine3)
  - a. Create a node3 folder using **mkdir node3** command
  - b. Go to the node3 folder using **cd node3** command
  - c. Run the command **geth --datadir ./data account new**
  - d. and save public address to your notepad
- 3) on ec2(machine1)
  - a. Create a node1 folder using **mkdir node1** command
  - b. Go to the node1 folder using **cd node1** command
  - c. Run the command **geth --datadir ./data account new**
  - d. and save public address to your notepad
- 4) on ec2(machine2)
  - a. Create genesis block by running command **puppeth**
    - 1.Go back to the machine2 node2 folder and type **puppeth**
    - 2.Gave network name as '**Groupfive**'
    - 3.Select option 2 - 'Configure new genesis',
    - 4.Select option 1 - 'Create new genesis from scratch',
    - 5.Select option 2 - 'Clique - proof-of-authority'.
    6. 'How many seconds should blocks take? – Gave 10 Seconds
    - 7.'Which accounts are allowed to seal?'
      - Airline1 - 0x84973c48281158Fa6A4c7FAc9499577f9f4b9cA2
      - Airline2 - 0x255Ae4535cb9379777F26D3BF11e3243581Ba05D
    - 8.'Which accounts should be pre-funded?'
      - Airline1 - 0x84973c48281158Fa6A4c7FAc9499577f9f4b9cA2
      - Airline2 - 0x255Ae4535cb9379777F26D3BF11e3243581Ba05D
    - 9.'Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei?(advisable yes)' - Yes
    10. Specified chain id as **98777**
    11. Selected option 2 - 'Manage existing genesis',
    - 12.Selected option 2 - 'Export genesis configurations',
    - 13.Opened groupfive.json and observe the config file properties.
    - 14.mv groupfive.json genesis.json
- 5) Copied the created genesis block on machine2 and pasted on machine3 and Machine1.

## 6. Initialized all the nodes

- 1) on machine1
  - a. Go to the node1 folder - **cd node1**
  - b. Run the command - **geth --datadir ./data init ../ genesis.json**
- 2) on machine2
  - a. Go to the node2 folder - **cd node2**
  - b. Run the command - **geth --datadir ./data init ../ genesis.json**
- 3) on machine3
  - a. Go to the node3 folder - **cd node3**
  - b. Run the command - **geth --datadir ./data init ../ genesis.json**

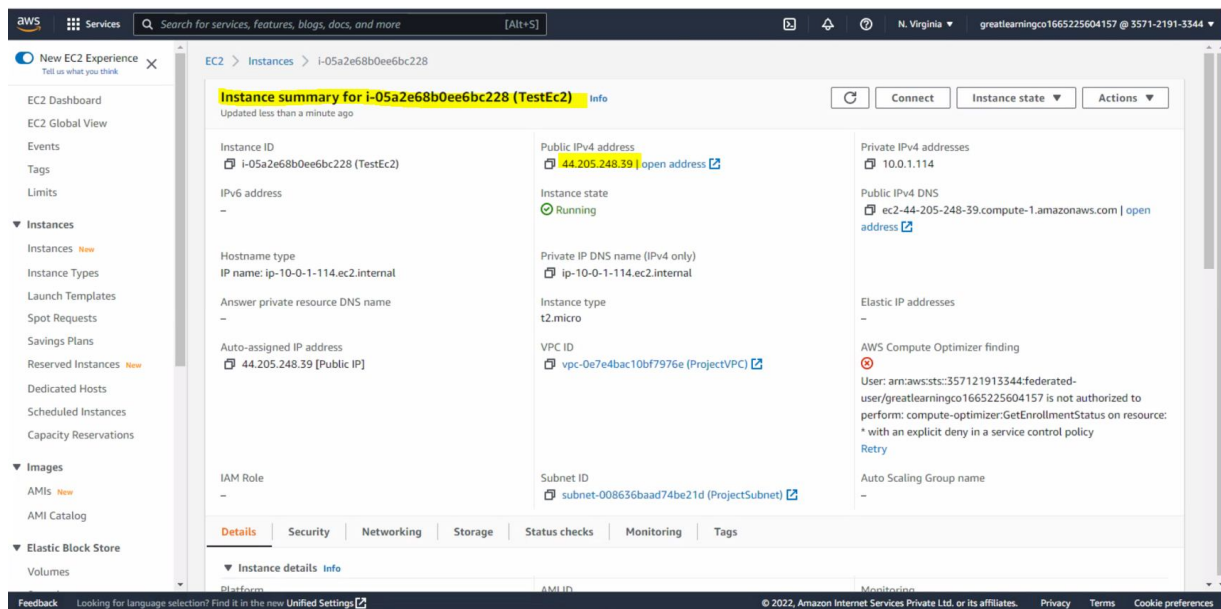
## 7. Created bootnode on machine1

- 1) Create a bnode folder using the **mkdir bnode** command
- 2) Go to the bnode folder using the **cd bnode** command
- 3) Run the command - **bootnode -genkey boot.key**
- 4) Run the command - **bootnode -nodekey ./boot.key -verbosity 7 -addr :30301**(kept it running)

## 8. We got enode:

enode://64692366a512335292b9ae42b3a34482167afb39f98a746c1afda05dee7343c581cdd204fb5aee98a3240303260bf61d1709edf2de5dc80658043e96707badaf@44.205.248.39:0?discport=30301

We changed public IPv4 address of the machine1 at the place of localhost



## 9. We created password.txt in node1, node2 and node3 directory and saved the password

## 10. Command on node 2 (machine2) by passing following parameters in the below command:

```
geth --networkid 98777 --datadir ./data --bootnodes  
"enode://64692366a512335292b9ae42b3a34482167afb39f98a746c1afda05dee73  
43c581cdd204fb5aee98a3240303260bf61d1709edf2de5dc80658043e96707badaf  
@44.205.248.39:0?discport=30301" --port 30303 --syncmode 'full' --http --allow-  
insecure-unlock --http.corsdomain "*" --http.port 8545 --http.addr ec2-18-212-  
198-154.compute-1.amazonaws.com --ipcdisable --unlock  
'84973c48281158Fa6A4c7FAc9499577f9f4b9cA2' --password ./password.txt --  
mine console
```

11. Ran command on node 3 (machine3) by passing following parameters in the below command:

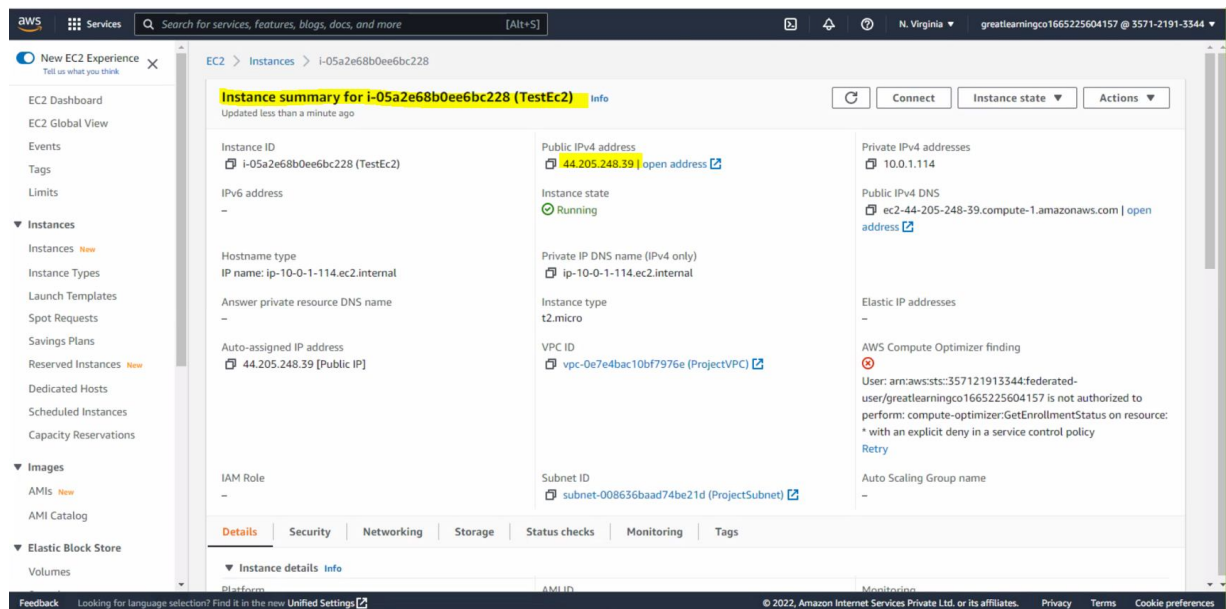
```
geth --networkid 98777 --datadir ./data --bootnodes
"enode://64692366a512335292b9ae42b3a34482167afb39f98a746c1afda05dee73
43c581cdd204fb5aee98a3240303260bf61d1709edf2de5dc80658043e96707badaf
@44.205.248.39:0?discport=30301" --port 30304 --syncmode 'full' --http --allow-
insecure-unlock --http.corsdomain "*" --http.port 8545 --ipcdisable --unlock
'0x255Ae4535cb9379777F26D3BF11e3243581Ba05D' --password ./password.txt -
-mine console
```

12. And in another terminal Run command on node1 (machine1) by passing below command:

```
geth --networkid 98777 --datadir ./data --bootnodes
"enode://64692366a512335292b9ae42b3a34482167afb39f98a746c1afda05dee73
43c581cdd204fb5aee98a3240303260bf61d1709edf2de5dc80658043e96707badaf
@44.205.248.39:0?discport=30301" --port 30305 --syncmode 'full' --http --allow-
insecure-unlock --http.corsdomain "*" --http.port 8545 --ipcdisable --unlock
'0xcc03AB6dA0fF4e20c761e52E4BB57D17646f09F6' --password ./password.txt
console
```

13. Below are the screenshots of the ec2 machines with configurations.

a. Machine1



## b. Machine2

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and a list of instances. The main content area shows the 'Instance summary for i-085573843dbbeb9b6 (TestEc3)'. The instance is in a 'Running' state. Key details include: Instance ID: i-085573843dbbeb9b6 (TestEc3), Public IPv4 address: 18.212.198.154, Private IPv4 address: 10.0.1.185, Hostname type: IP name: ip-10-0-1-185.ec2.internal, Instance type: t2.micro, VPC ID: vpc-0e7e4bac10bf7976e (ProjectVPC), and Subnet ID: subnet-008636baad74be21d (ProjectSubnet). A warning message from AWS Compute Optimizer is visible, indicating that the user is not authorized to perform certain actions on the resource.

Instance ID	Public IPv4 address	Private IPv4 addresses
i-085573843dbbeb9b6 (TestEc3)	18.212.198.154 (open address)	10.0.1.185

Instance state	Public IPv4 DNS
Running	ec2-18-212-198-154.compute-1.amazonaws.com (open address)

Hostname type	Private IP DNS name (IPv4 only)	Instance type
IP name: ip-10-0-1-185.ec2.internal	ip-10-0-1-185.ec2.internal	t2.micro

Auto-assigned IP address	VPC ID	Subnet ID
18.212.198.154 [Public IP]	vpc-0e7e4bac10bf7976e (ProjectVPC)	subnet-008636baad74be21d (ProjectSubnet)

Warning: AWS Compute Optimizer finding. User: aws:sts::357121913344:federated-user/greatlearningco1665225604157 is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: \* with an explicit deny in a service control policy. [Retry]

## c. Machine3

The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and a list of instances. The main content area shows the 'Instance summary for i-0b2b51116325d675f (TestEc4)'. The instance is in a 'Running' state. Key details include: Instance ID: i-0b2b51116325d675f (TestEc4), Public IPv4 address: 34.230.61.211, Private IPv4 address: 10.0.1.75, Hostname type: IP name: ip-10-0-1-75.ec2.internal, Instance type: t2.micro, VPC ID: vpc-0e7e4bac10bf7976e (ProjectVPC), and Subnet ID: subnet-008636baad74be21d (ProjectSubnet). A warning message from AWS Compute Optimizer is visible, indicating that the user is not authorized to perform certain actions on the resource.

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b2b51116325d675f (TestEc4)	34.230.61.211 (open address)	10.0.1.75

Instance state	Public IPv4 DNS
Running	ec2-34-230-61-211.compute-1.amazonaws.com (open address)

Hostname type	Private IP DNS name (IPv4 only)	Instance type
IP name: ip-10-0-1-75.ec2.internal	ip-10-0-1-75.ec2.internal	t2.micro

Auto-assigned IP address	VPC ID	Subnet ID
34.230.61.211 [Public IP]	vpc-0e7e4bac10bf7976e (ProjectVPC)	subnet-008636baad74be21d (ProjectSubnet)

Warning: AWS Compute Optimizer finding. User: aws:sts::357121913344:federated-user/greatlearningco1665225604157 is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: \* with an explicit deny in a service control policy. [Retry]

Screen shot where ec2 machines are running

```
4. ubuntu@ip-10-0-1-114: ~node1
2. ubuntu@ip-10-0-1-185: ~node2
3. ubuntu@ip-10-0-1-75: ~node3

3. Remove genesis configuration
> 2

Which folder to save the genesis spec into? (default = current)
Will create groupfive.json
>
INFO [10-08|10:50:59.267] Saved native genesis chain spec      path=groupfive.json

What would you like to do? (default = stats)
1. Show network stats
2. Manage existing genesis
3. Track new remote server
4. Deploy network components
>
INFO [10-08|10:51:02.236] No remote machines to gather stats from

What would you like to do? (default = stats)
1. Show network stats
2. Manage existing genesis
3. Track new remote server
4. Deploy network components
> ^C
> CRIT [10-08|10:51:04.566] Failed to read user input      err=E0F
ubuntu@ip-10-0-1-185:~/node2$ ls
data groupfive.json
ubuntu@ip-10-0-1-185:~/node2$ mv groupfive.json genesis.json
ubuntu@ip-10-0-1-185:~/node2$ ls
data genesis.json
ubuntu@ip-10-0-1-185:~/node2$ cat genesis.json
{
  "config": {
    "chainId": 98777,
    "homesteadBlock": 0,
    "eip158Block": 0,
    "eip150hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip158Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "clique": {
      "period": 10,
      "epoch": 30000
    }
  },
  "nonce": "0x0",
  "timestamp": 1514567890,
  "extraData": ""
}
```

Initiating the genesis on Machine 2 with chain id 98777

```
5. ubuntu@ip-10-0-1-114: ~node
4. ubuntu@ip-10-0-1-114: ~node1
2. ubuntu@ip-10-0-1-185: ~node2
3. ubuntu@ip-10-0-1-75: ~node3

--txpool.rejournal.value (default: 1h0m0s)
Time interval to regenerate the local transaction journal

VIRTUAL MACHINE
--vmdebug (default: false)
Record information useful for VM and contract debugging

COPYRIGHT:
Copyright 2013-2022 The go-ethereum Authors
flag provided but not defined: -dadadir
ubuntu@ip-10-0-1-114:~/node1$ geth --dadadir ./data init genesis.json
INFO [10-08|10:54:39.138] Maximum peer count ETH=50 LES=0 total=50
INFO [10-08|10:54:39.146] Smartcard socket not found, disabling err=stat /run/pcscd/pcscd.comm: no such file or directory
WARN [10-08|10:54:39.168] Sanitizing cache to Go's GC limits provided=1024 updated=322
INFO [10-08|10:54:39.169] Set global gas cap cap=50,000,000
INFO [10-08|10:54:39.173] Allocated cache and file handles database=/home/ubuntu/node1/data/eth/chaindata cache=16.00MiB handles=16
INFO [10-08|10:54:39.213] Opened ancient database database=/home/ubuntu/node1/data/eth/chaindata/ancient/chain readonly=false
INFO [10-08|10:54:39.213] Writing custom genesis block nodes=357 size=50.70KiB time=1.417854ms gcnodes=0 gcsz=0.00B gctime=0s livenodes=1 liveness=0.00B
INFO [10-08|10:54:39.229] Persisted trie from memory database database=chaindata hash=f17d30..e12ccf
INFO [10-08|10:54:39.238] Successfully wrote genesis state database=/home/ubuntu/node1/data/eth/lightchaindata cache=16.00MiB handles=16
INFO [10-08|10:54:39.238] Allocated cache and file handles database=/home/ubuntu/node1/data/eth/lightchaindata/ancient/chain readonly=false
INFO [10-08|10:54:39.264] Opened ancient database nodes=357 size=50.70KiB time="867.813µs" gcnodes=0 gcsz=0.00B gctime=0s livenodes=1 liveness=0.00B
INFO [10-08|10:54:39.264] Writing custom genesis block database=lightchaindata hash=f17d30..e12ccf
INFO [10-08|10:54:39.276] Persisted trie from memory database
B
INFO [10-08|10:54:39.277] Successfully wrote genesis state
ubuntu@ip-10-0-1-114:~/node1$
ubuntu@ip-10-0-1-114:~/node1$ echo '12345'>password.txt
ubuntu@ip-10-0-1-114:~/node1$ geth --networkid 98777 --dadadir ./data --bootnodes "enode://64692366a512335292b9ae42b3a34482167afb39f98a746c1afda05dee7343c581cdd204fb5aee98a3240303260bf61d1709edf2de5dc80658043e96707badaf@44.205.248.39:0?discport=30301" --port 30305 --syncmode 'full' --http --allow-insecure-unlock --http.corsdomain "*" --http.port 8545 --ipcdisable --unlock '0xccc03A86dA0fF4e20c761e52E48B57D17646f09F6' --password ./password.txt console
INFO [10-08|11:06:28.310] Maximum peer count ETH=50 LES=0 total=50
INFO [10-08|11:06:28.312] Smartcard socket not found, disabling err=stat /run/pcscd/pcscd.comm: no such file or directory
WARN [10-08|11:06:28.332] Sanitizing cache to Go's GC limits provided=1024 updated=322
INFO [10-08|11:06:28.332] Set global gas cap cap=50,000,000
INFO [10-08|11:06:28.336] Allocated trie memory caches clean=48.00MiB dirty=80.00MiB
INFO [10-08|11:06:28.354] Allocated cache and file handles database=/home/ubuntu/node1/data/eth/chaindata cache=160.00MiB handles=524,288
INFO [10-08|11:06:28.384] Opened ancient database database=/home/ubuntu/node1/data/eth/chaindata/ancient/chain readonly=false
INFO [10-08|11:06:28.385] -----
INFO [10-08|11:06:28.385] Chain ID: 98777 (unknown)
INFO [10-08|11:06:28.385] Consensus: Clique (proof-of-authority)
INFO [10-08|11:06:28.385]
```

lobaTerm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



## Machine 2 was started with genesis and started sealing

```
at block: 1 (Sat Oct 08 2022 11:03:41 GMT+0000 (UTC))
datadir: /home/ubuntu/node2/data
modules: admin:1.0 clique:1.0 debug:1.0 engine:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> INFO [10-08|11:09:02.397] Looking for peers
INFO [10-08|11:09:08.583] Imported new chain segment
INFO [10-08|11:09:08.583] Commit new sealing work
INFO [10-08|11:09:08.584] Commit new sealing work
INFO [10-08|11:09:12.421] Looking for peers
INFO [10-08|11:09:18.001] Successfully sealed new block
INFO [10-08|11:09:18.001] % mined potential block
INFO [10-08|11:09:18.002] Commit new sealing work
WARN [10-08|11:09:18.002] Block sealing failed
INFO [10-08|11:09:18.003] Commit new sealing work
INFO [10-08|11:09:22.448] Looking for peers
INFO [10-08|11:09:28.004] Imported new chain segment
INFO [10-08|11:09:28.005] Commit new sealing work
INFO [10-08|11:09:28.005] Commit new sealing work
INFO [10-08|11:09:32.474] Looking for peers
INFO [10-08|11:09:35.000] Successfully sealed new block
INFO [10-08|11:09:38.001] % mined potential block
INFO [10-08|11:09:38.002] Commit new sealing work
WARN [10-08|11:09:38.002] Block sealing failed
INFO [10-08|11:09:38.002] Commit new sealing work
INFO [10-08|11:09:42.502] Looking for peers
INFO [10-08|11:09:48.003] Imported new chain segment
INFO [10-08|11:09:48.004] Commit new sealing work
INFO [10-08|11:09:48.005] Commit new sealing work
INFO [10-08|11:09:52.525] Looking for peers
INFO [10-08|11:09:58.002] Successfully sealed new block
INFO [10-08|11:09:58.002] % mined potential block
INFO [10-08|11:09:58.003] Commit new sealing work
WARN [10-08|11:09:58.003] Block sealing failed
INFO [10-08|11:09:58.004] Commit new sealing work
INFO [10-08|11:10:02.551] Looking for peers
INFO [10-08|11:10:08.004] Imported new chain segment
INFO [10-08|11:10:08.005] Commit new sealing work
INFO [10-08|11:10:08.005] Commit new sealing work
INFO [10-08|11:10:12.578] Looking for peers
INFO [10-08|11:10:18.001] Successfully sealed new block
INFO [10-08|11:10:18.001] % mined potential block
INFO [10-08|11:10:18.002] Commit new sealing work
WARN [10-08|11:10:18.002] Block sealing failed
INFO [10-08|11:10:18.002] Commit new sealing work
INFO [10-08|11:10:22.604] Looking for peers

peercount=1 tried=1 static=0
blocks=1 txs=0 mgas=0.000 elapsed="347.326µs" mgasps=0.000 number=2 hash=17d864..dffcab dirty=0.008
number=3 sealhash=cab0b9..60fcd6 uncles=0 txs=0 gas=0 fees=0 elapsed="148.4µs"
number=3 sealhash=cab0b9..60fcd6 uncles=0 txs=0 gas=0 fees=0 elapsed="599.069µs"
peercount=1 tried=0 static=0
number=3 sealhash=cab0b9..60fcd6 hash=f8d1d2..ae1c09 elapsed=9.417s
number=3 hash=f8d1d2..ae1c09
number=4 sealhash=4e6af8..9c0c9a uncles=0 txs=0 gas=0 fees=0 elapsed="330.204µs"
error="signed recently, must wait for others"
number=4 sealhash=4e6af8..9c0c9a uncles=0 txs=0 gas=0 fees=0 elapsed="961.612µs"
peercount=2 tried=0 static=0
blocks=1 txs=0 mgas=0.000 elapsed="290.698µs" mgasps=0.000 number=4 hash=9adb80..91103b dirty=0.008
number=5 sealhash=202a79..754d11 uncles=0 txs=0 gas=0 fees=0 elapsed="143.755µs"
number=5 sealhash=202a79..754d11 uncles=0 txs=0 gas=0 fees=0 elapsed="587.376µs"
peercount=2 tried=0 static=0
number=5 sealhash=202a79..754d11 hash=d855f4..4756e1 elapsed=9.996s
number=5 hash=d855f4..4756e1
number=6 sealhash=b3fef2..70f2f7 uncles=0 txs=0 gas=0 fees=0 elapsed="326.849µs"
error="signed recently, must wait for others"
number=6 sealhash=b3fef2..70f2f7 uncles=0 txs=0 gas=0 fees=0 elapsed="766.977µs"
peercount=2 tried=0 static=0
blocks=1 txs=0 mgas=0.000 elapsed="288.973µs" mgasps=0.000 number=6 hash=937232..3c7979 dirty=0.008
number=7 sealhash=59264a..d9072c uncles=0 txs=0 gas=0 fees=0 elapsed="151.329µs"
number=7 sealhash=59264a..d9072c uncles=0 txs=0 gas=0 fees=0 elapsed="611.053µs"
peercount=2 tried=0 static=0
number=7 sealhash=59264a..d9072c hash=01c218..826494 elapsed=9.998s
number=7 hash=01c218..826494
number=8 sealhash=7a5548..86780d uncles=0 txs=0 gas=0 fees=0 elapsed="339.409µs"
error="signed recently, must wait for others"
number=8 sealhash=7a5548..86780d uncles=0 txs=0 gas=0 fees=0 elapsed="773.586µs"
peercount=2 tried=0 static=0
blocks=1 txs=0 mgas=0.000 elapsed="271.932µs" mgasps=0.000 number=8 hash=b562a8..3731c4 dirty=0.008
number=9 sealhash=b07bfa..5c7561 uncles=0 txs=0 gas=0 fees=0 elapsed="144.134µs"
number=9 sealhash=b07bfa..5c7561 uncles=0 txs=0 gas=0 fees=0 elapsed="580.494µs"
peercount=2 tried=0 static=0
number=9 sealhash=b07bfa..5c7561 hash=fd4cc4..78eba9 elapsed=9.996s
number=9 hash=fd4cc4..78eba9
number=10 sealhash=856223..67d81c uncles=0 txs=0 gas=0 fees=0 elapsed="325.198µs"
error="signed recently, must wait for others"
number=10 sealhash=856223..67d81c uncles=0 txs=0 gas=0 fees=0 elapsed="816.468µs"
peercount=2 tried=0 static=0
```

## Machine 3 also started sealing

```
INFO [10-08|11:09:08.580] % mined potential block
INFO [10-08|11:09:08.581] Commit new sealing work
WARN [10-08|11:09:08.582] Block sealing failed
INFO [10-08|11:09:08.582] Commit new sealing work
INFO [10-08|11:09:08.628] Looking for peers
INFO [10-08|11:09:18.004] Imported new chain segment
INFO [10-08|11:09:18.004] Commit new sealing work
INFO [10-08|11:09:18.005] Commit new sealing work
INFO [10-08|11:09:18.652] Looking for peers
INFO [10-08|11:09:28.001] Successfully sealed new block
INFO [10-08|11:09:28.002] % mined potential block
INFO [10-08|11:09:28.002] Commit new sealing work
WARN [10-08|11:09:28.002] Block sealing failed
INFO [10-08|11:09:28.002] Commit new sealing work
INFO [10-08|11:09:28.678] Looking for peers
INFO [10-08|11:09:38.004] Imported new chain segment
INFO [10-08|11:09:38.005] Commit new sealing work
INFO [10-08|11:09:38.006] Commit new sealing work
INFO [10-08|11:09:38.706] Looking for peers
INFO [10-08|11:09:48.000] Successfully sealed new block
INFO [10-08|11:09:48.001] % mined potential block
INFO [10-08|11:09:48.001] Commit new sealing work
WARN [10-08|11:09:48.001] Block sealing failed
INFO [10-08|11:09:48.002] Commit new sealing work
INFO [10-08|11:09:48.002] Commit new sealing work
INFO [10-08|11:09:48.731] Looking for peers
INFO [10-08|11:09:58.005] Imported new chain segment
INFO [10-08|11:09:58.006] Commit new sealing work
INFO [10-08|11:09:58.007] Commit new sealing work
INFO [10-08|11:09:58.760] Looking for peers
INFO [10-08|11:10:08.001] Successfully sealed new block
INFO [10-08|11:10:08.002] % mined potential block
INFO [10-08|11:10:08.002] Commit new sealing work
WARN [10-08|11:10:08.002] Block sealing failed
INFO [10-08|11:10:08.002] Commit new sealing work
INFO [10-08|11:10:08.786] Looking for peers
INFO [10-08|11:10:18.005] Imported new chain segment
INFO [10-08|11:10:18.006] % block reached canonical chain
INFO [10-08|11:10:18.006] Commit new sealing work
INFO [10-08|11:10:18.006] Commit new sealing work
INFO [10-08|11:10:18.814] Looking for peers
INFO [10-08|11:10:28.001] Successfully sealed new block
INFO [10-08|11:10:28.001] % mined potential block
INFO [10-08|11:10:28.002] Commit new sealing work
WARN [10-08|11:10:28.002] Block sealing failed
INFO [10-08|11:10:28.003] Commit new sealing work
INFO [10-08|11:10:28.844] Looking for peers

number=2 hash=17d864..dffcab
number=3 sealhash=419635..88aa14 uncles=1 txs=0 gas=0 fees=0 elapsed="307.565µs"
error="signed recently, must wait for others"
number=3 sealhash=419635..88aa14 uncles=1 txs=0 gas=0 fees=0 elapsed="638.596µs"
peercount=1 tried=0 static=0
blocks=1 txs=0 mgas=0.000 elapsed="288.477µs" mgasps=0.000 number=3 hash=f8d1d2..ae1c09 dirty=0.008
number=4 sealhash=6e4cfa..a0c438 uncles=1 txs=0 gas=0 fees=0 elapsed="144.134µs"
number=4 sealhash=6e4cfa..a0c438 uncles=1 txs=0 gas=0 fees=0 elapsed="631.264µs"
peercount=2 tried=0 static=0
number=4 sealhash=6e4cfa..a0c438 hash=9adb80..91103b elapsed=9.996s
number=4 hash=9adb80..91103b
number=5 sealhash=9900ac..7ddb07 uncles=1 txs=0 gas=0 fees=0 elapsed="350.273µs"
error="signed recently, must wait for others"
number=5 sealhash=9900ac..7ddb07 uncles=1 txs=0 gas=0 fees=0 elapsed="815.295µs"
peercount=2 tried=0 static=0
blocks=1 txs=0 mgas=0.000 elapsed=1.034ms mgasps=0.000 number=5 hash=d855f4..4756e1 dirty=0.008
number=6 sealhash=34d214..5b487a uncles=1 txs=0 gas=0 fees=0 elapsed="222.895µs"
number=6 sealhash=34d214..5b487a uncles=1 txs=0 gas=0 fees=0 elapsed="661.045µs"
peercount=2 tried=0 static=0
number=6 sealhash=34d214..5b487a hash=937232..3c7979 elapsed=9.995s
number=6 hash=937232..3c7979
number=7 sealhash=a6e32b..82a97f uncles=1 txs=0 gas=0 fees=0 elapsed="281.31µs"
error="signed recently, must wait for others"
number=7 sealhash=a6e32b..82a97f uncles=1 txs=0 gas=0 fees=0 elapsed=1.204ms
peercount=2 tried=0 static=0
blocks=1 txs=0 mgas=0.000 elapsed="292.076µs" mgasps=0.000 number=7 hash=01c218..826494 dirty=0.008
number=8 sealhash=c57d30..60a458 uncles=0 txs=0 gas=0 fees=0 elapsed="189.073µs"
number=8 sealhash=c57d30..60a458 uncles=0 txs=0 gas=0 fees=0 elapsed="593.541µs"
peercount=2 tried=0 static=0
number=8 sealhash=c57d30..60a458 hash=b562a8..3731c4 elapsed=9.994s
number=8 hash=b562a8..3731c4
number=9 sealhash=89c813..1f3dd3 uncles=0 txs=0 gas=0 fees=0 elapsed="388.084µs"
error="signed recently, must wait for others"
number=9 sealhash=89c813..1f3dd3 uncles=0 txs=0 gas=0 fees=0 elapsed="854.218µs"
peercount=2 tried=0 static=0
blocks=1 txs=0 mgas=0.000 elapsed="261.465µs" mgasps=0.000 number=9 hash=fd4cc4..78eba9 dirty=0.008
number=2 hash=17d864..dffcab
number=10 sealhash=b2b0f6..fbda73 uncles=0 txs=0 gas=0 fees=0 elapsed="352.817µs"
number=10 sealhash=b2b0f6..fbda73 uncles=0 txs=0 gas=0 fees=0 elapsed="781.026µs"
peercount=2 tried=0 static=0
number=10 sealhash=b2b0f6..fbda73 hash=def0eb..4b029e elapsed=9.995s
number=10 hash=def0eb..4b029e
number=11 sealhash=74cc6c..81177f uncles=0 txs=0 gas=0 fees=0 elapsed="292.07µs"
error="signed recently, must wait for others"
number=11 sealhash=74cc6c..81177f uncles=0 txs=0 gas=0 fees=0 elapsed="780.37µs"
peercount=2 tried=0 static=0
```

## Configuring network to MetaMask

General    Networks > Add a network > Add a network manually

**Network name**

groupfive

**New RPC URL**

http://18.212.198.154:8545

**Chain ID**

9

The RPC URL you have entered returned a different chain ID (98777). Please update the Chain ID to match the RPC URL of the network you are trying to add.

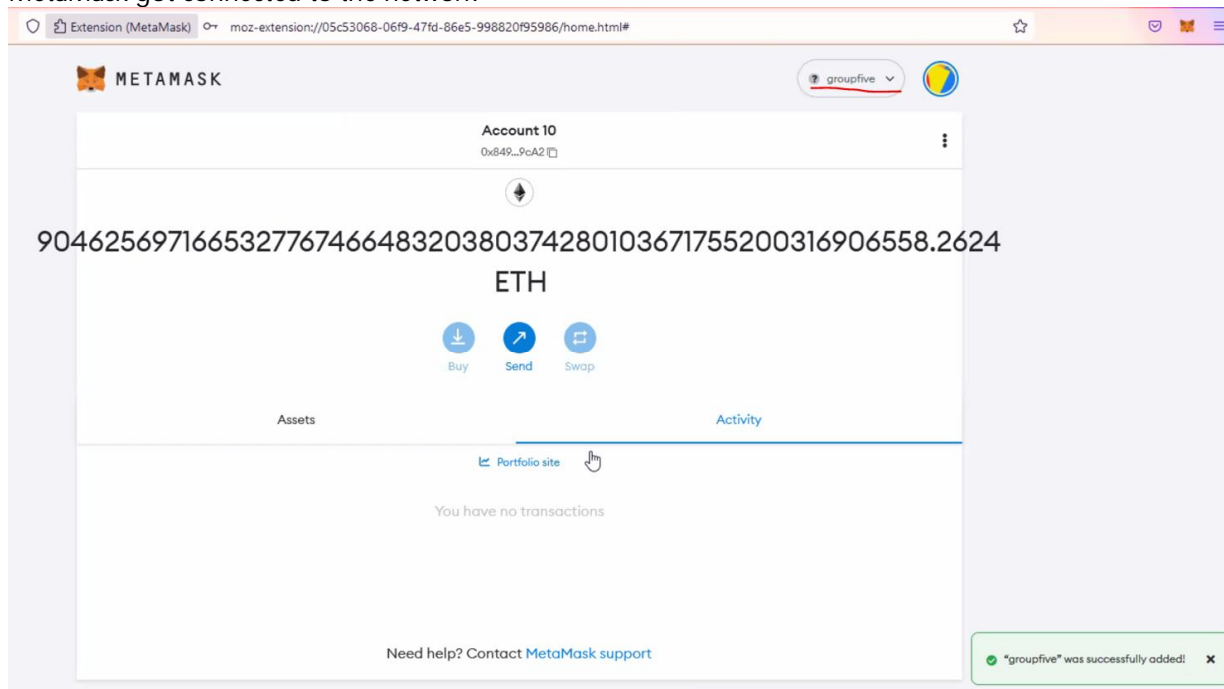
**Currency symbol**

ETH

The network with chain ID 9 may use a different currency symbol (TUBQ) than the one you have entered. Please verify before continuing.

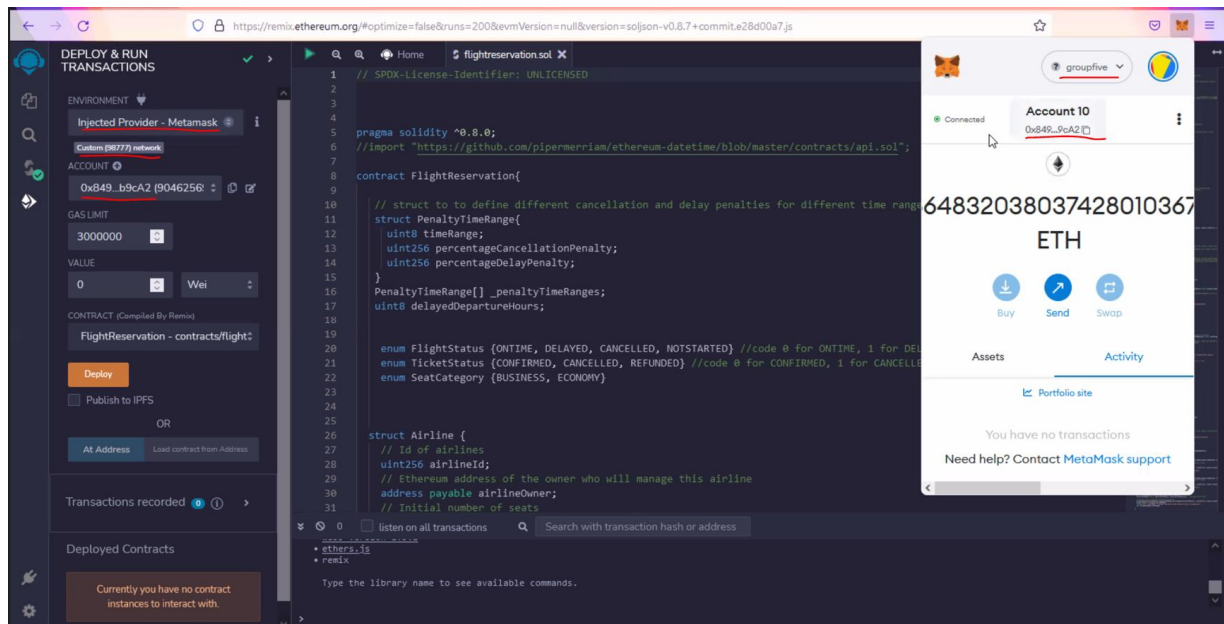
**Block explorer URL (Optional)**

## MetaMask got connected to the network

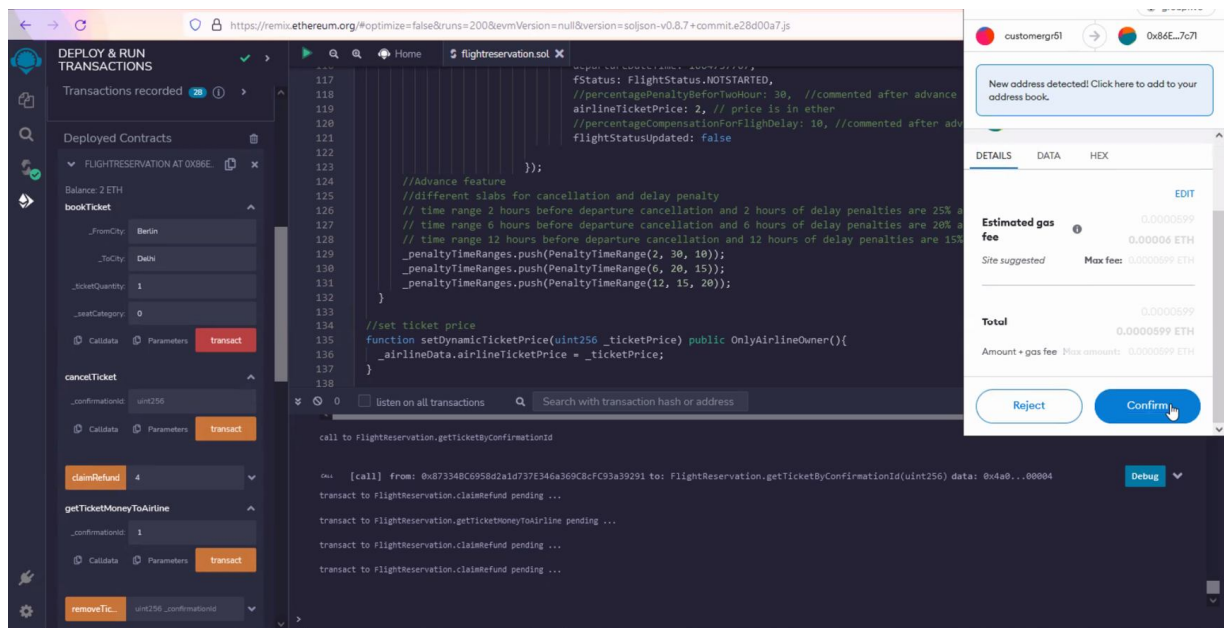




The Account also got injected to remix and ready for execution of the Smart Contract



Also added customer account in MetaMask and connected with remix and started booking the flight ticket and it's asking the transaction confirmation on MetaMask.



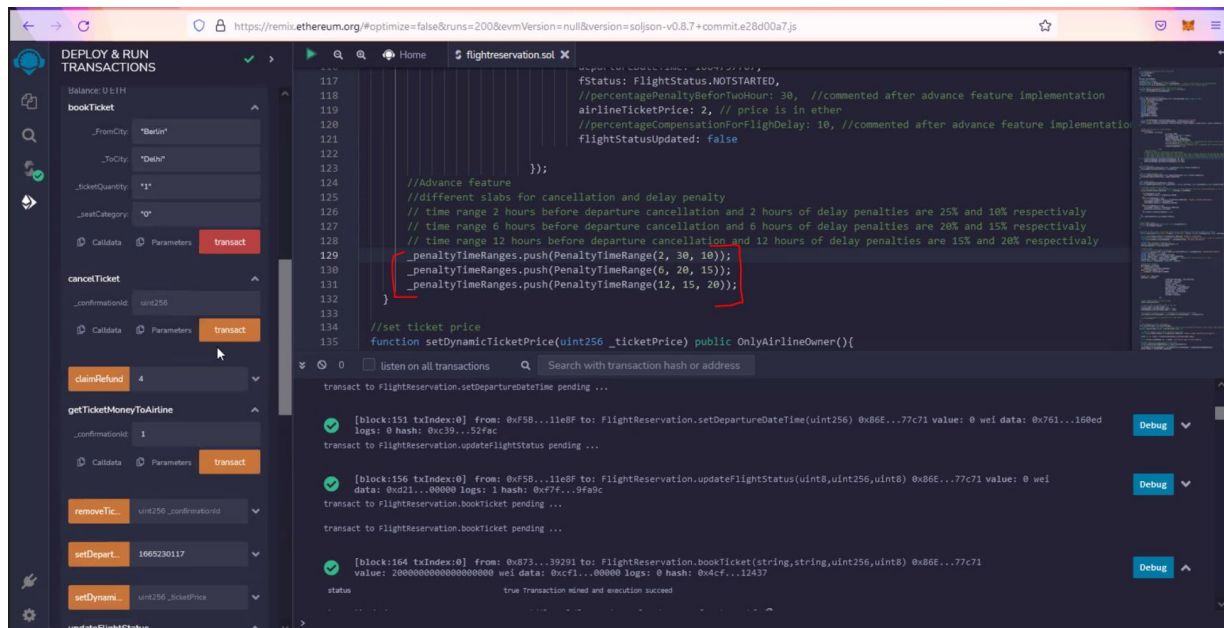
## Flight reservation completed and confirmation received on remix

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows a list of transactions. The 'updateFlightStatus' transaction is selected, with parameters: `_flightStatus: 1`, `_airlineId: 9988`, and `_delayedHours: 2`. The 'transaction' button is highlighted. The main editor shows the Solidity code for the `FlightReservation` contract, including functions like `setDepartureDateTime`, `updateFlightStatus`, and `bookTicket`. The right panel shows the transaction details, including the transaction hash `0x223a0d7f70272f70e012230e0f34c08ac180f` and the transaction status 'true Transaction mined and execution succeed'.

## Showing mining transaction successful information on the console

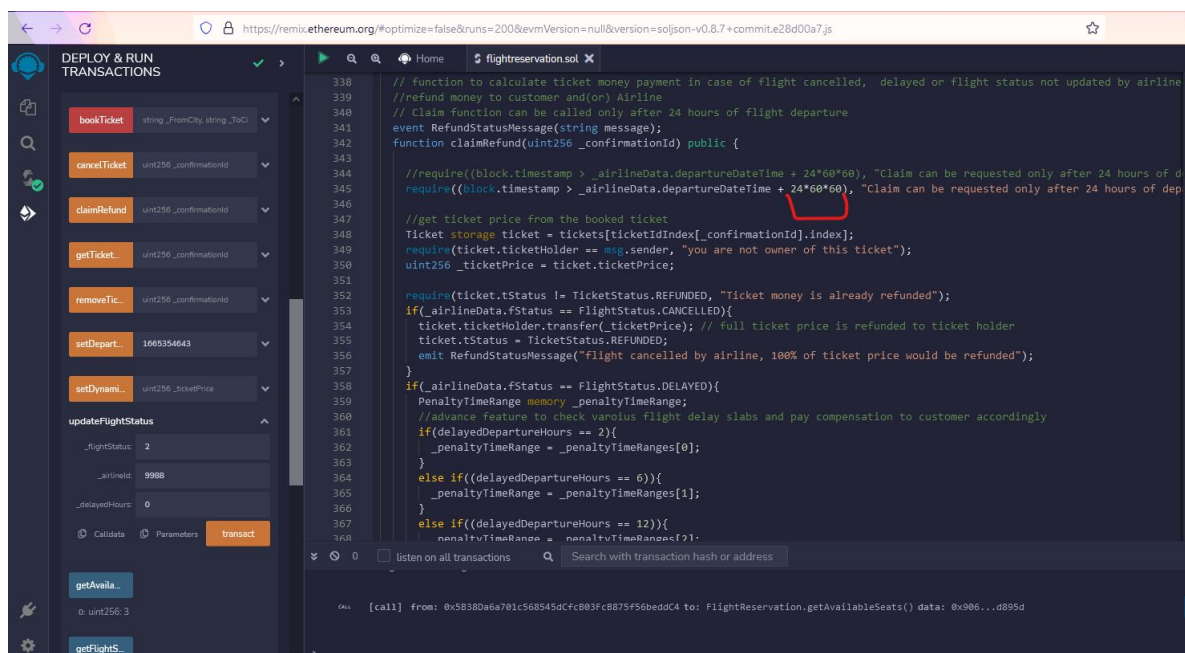
The screenshot shows the terminal output of a mining process. The output includes various log messages such as 'INFO [10-08]11:44:48.004 Imported new chain segment', 'WARN [10-08]11:44:49.508 Served eth\_call', and 'INFO [10-08]11:44:58.001 Successfully sealed new block'. The output also displays transaction details, including the transaction hash `0x223a0d7f70272f70e012230e0f34c08ac180f` and the transaction status 'true Transaction mined and execution succeed'.

Implemented the **advanced** feature in the contract



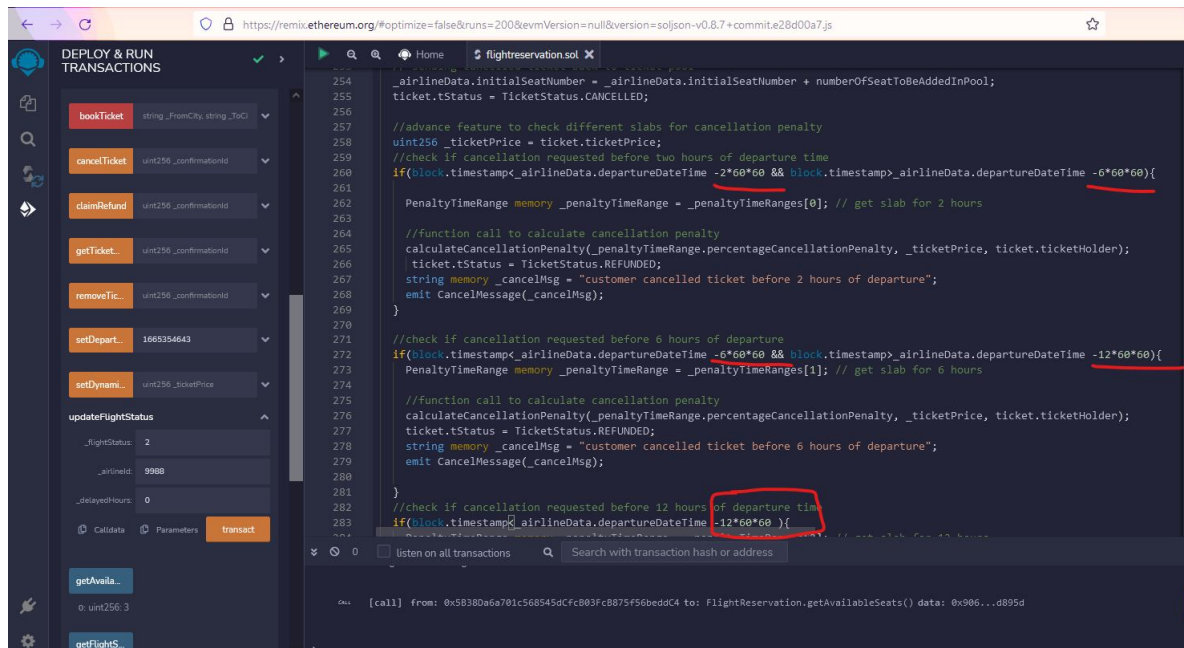
## Interaction with the system

Time stamp in the entire system has been used as epoch timestamp. In order to test system faster, time duration in the code can be reduced so that customer would not have to wait 24 hours after departure time to get refund for delay or cancellation or flight status not updated by airline, code can be modified for 2 minutes for faster execution, also for the cancellation by the customer time duration can be reduced to test the system faster. Please refer to screenshots





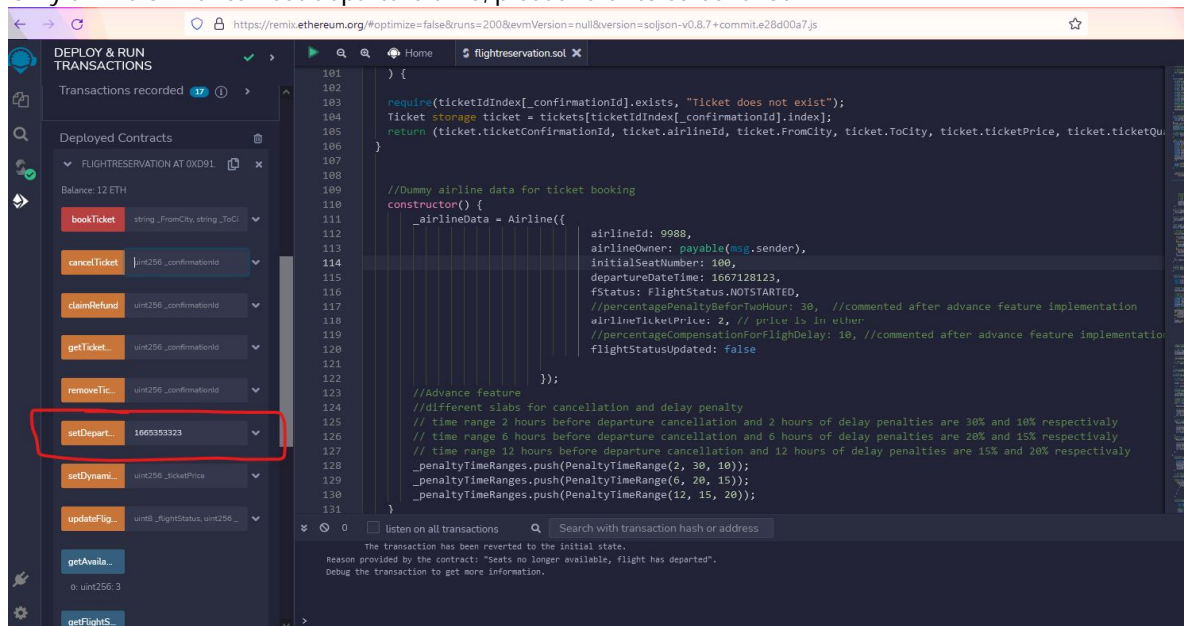
please change to  $2*60$ (2 minutes) to check faster



```
254 _airlineData.initialSeatNumber = _airlineData.initialSeatNumber + numberOfSeatToBeAddedInPool;
255 ticket.tStatus = TicketStatus.CANCELLED;
256
257 //advance feature to check different slabs for cancellation penalty
258 uint256 _ticketPrice = ticket.ticketPrice;
259 //check if cancellation requested before two hours of departure time
260 if(block.timestamp < _airlineData.departureDateTime - 2*60*60 && block.timestamp > _airlineData.departureDateTime - 6*60*60){
261
262     PenaltyTimeRange memory _penaltyTimeRange = _penaltyTimeRanges[0]; // get slab for 2 hours
263
264     //function call to calculate cancellation penalty
265     calculateCancellationPenalty(_penaltyTimeRange.percentageCancellationPenalty, _ticketPrice, ticket.ticketHolder);
266     ticket.tStatus = TicketStatus.REFUNDED;
267     string memory _cancelMsg = "customer cancelled ticket before 2 hours of departure";
268     emit CancelMessage(_cancelMsg);
269 }
270
271 //check if cancellation requested before 6 hours of departure
272 if(block.timestamp < _airlineData.departureDateTime - 6*60*60 && block.timestamp > _airlineData.departureDateTime - 12*60*60){
273     PenaltyTimeRange memory _penaltyTimeRange = _penaltyTimeRanges[1]; // get slab for 6 hours
274
275     //function call to calculate cancellation penalty
276     calculateCancellationPenalty(_penaltyTimeRange.percentageCancellationPenalty, _ticketPrice, ticket.ticketHolder);
277     ticket.tStatus = TicketStatus.REFUNDED;
278     string memory _cancelMsg = "customer cancelled ticket before 6 hours of departure";
279     emit CancelMessage(_cancelMsg);
280 }
281
282 //check if cancellation requested before 12 hours of departure time
283 if(block.timestamp < _airlineData.departureDateTime - 12*60*60 ){
284     //function call to calculate cancellation penalty
285     calculateCancellationPenalty(_penaltyTimeRange.percentageCancellationPenalty, _ticketPrice, ticket.ticketHolder);
286     ticket.tStatus = TicketStatus.REFUNDED;
287     string memory _cancelMsg = "customer cancelled ticket before 12 hours of departure";
288     emit CancelMessage(_cancelMsg);
289 }
```

After changing time in the code:

1. Deploy **flightreservation.sol** on Remix with Airline Account
2. At the time of evaluation of the project time flight departure time may be in the past. System will not allow to book the ticket. In order to test the system, Airline owner needs to set new departure time (Currently system supports **Epoch timestamp**) ex. 1665396523. Only airline owner can set departure time, please refer to screenshot.



```
101 } {
102
103     require(ticketIdIndex[confirmationId].exists, "Ticket does not exist");
104     Ticket storage ticket = tickets[ticketIdIndex[confirmationId].index];
105     return (ticket.ticketConfirmationId, ticket.airlineId, ticket.FromCity, ticket.ToCity, ticket.ticketPrice, ticket.ticketQu
106 }
107
108 //Dummy airline data for ticket booking
109 constructor() {
110     _airlineData = Airline({
111         airlineId: 9988,
112         airlineOwner: payable(msg.sender),
113         initialSeatNumber: 100,
114         departureDateTime: 1667128123,
115         fStatus: FlightStatus.NOTSTARTED,
116         //percentagePenaltyBeforeTwoHour: 30, //commented after advance feature implementation
117         airlineTicketPrice: 2, // price is in ether
118         //percentageCompensationForFlightDelay: 10, //commented after advance feature implementation
119         flightStatusUpdated: false
120     });
121 }
122
123 //Advance feature
124 //different slabs for cancellation and delay penalty
125 // time range 2 hours before departure cancellation and 2 hours of delay penalties are 30% and 10% respectively
126 // time range 6 hours before departure cancellation and 6 hours of delay penalties are 20% and 15% respectively
127 // time range 12 hours before departure cancellation and 12 hours of delay penalties are 15% and 20% respectively
128 _penaltyTimeRanges.push(PenaltyTimeRange(2, 30, 10));
129 _penaltyTimeRanges.push(PenaltyTimeRange(6, 20, 15));
130 _penaltyTimeRanges.push(PenaltyTimeRange(12, 15, 20));
131 }
```



3. Customer can book ticket by invoking bookTicket function and providing details like From city, To City, seat quantity and seat category, customer also needs to pass ticket money (in ether) in value fields.

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows the 'bookTicket' function being executed with the following parameters: FromCity: Berlin, ToCity: Delhi, TicketQuantity: 1, SeatCategory: 1, and a value of 2 ETH. The main editor shows the Solidity code for the FlightReservation contract, including the constructor and the bookTicket function. The console at the bottom shows the transaction hash and status.

4. Customer can get

- ticket details by invoking function **getTicketByConfirmationId**
- flight status by invoking function **getFlightStatus**
- available seats by invoking function **getAvailableSeats**

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows the 'getTicketByConfirmationId' function being executed with the following parameters: ConfirmationId: 2, TicketQuantity: 1, and SeatCategory: 1. The main editor shows the Solidity code for the FlightReservation contract, including the constructor and the getTicketByConfirmationId function. The console at the bottom shows the transaction hash and status.

5. Customer can cancel ticket by invoking function **cancelTicket** and providing confirmation id

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays a list of functions for the 'FLIGHTRESERVATION AT 0XD91' contract. The 'cancelTicket' function is highlighted with a red box. The main editor shows the Solidity code for 'flightreservation.sol'. The code includes functions for setting ticket price, departure time, and flight status, as well as an event for showing flight status. The 'cancelTicket' function is defined as follows:

```
function cancelTicket(uint256 _confirmationId) public OnlyAirlineOwner() {
    // ... (code for canceling the ticket)
}
```

6. Customer can claim refund by invoking function **claimRefund** and providing confirmation Id after 24 hours of departure time

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays a list of functions for the 'FLIGHTRESERVATION AT 0XD91' contract. The 'claimRefund' function is highlighted with a red box. The main editor shows the Solidity code for 'flightreservation.sol'. The code includes functions for setting ticket price, departure time, and flight status, as well as an event for showing flight status. The 'claimRefund' function is defined as follows:

```
function claimRefund(uint256 _confirmationId) public OnlyAirlineOwner() {
    // ... (code for claiming a refund)
}
```

7. Airline owner can get ticket money from smart contract by invoking function **getTicketMoneyToAirline** and passing confirmation Id after flight departure and in case of normal flight.

The screenshot shows the Remix IDE interface. On the left, under 'DEPLOY & RUN TRANSACTIONS', the 'getTicket' function is highlighted with a red box. The function parameters are set to 'uint256 \_confirmationId'. The main editor displays the Solidity code for the 'FlightReservation' contract. The code includes functions for setting ticket price, dynamic ticket price, departure time, and flight status. The 'getTicket' function is not explicitly shown in the code snippet, but the sidebar indicates it is the function being invoked.

8. Airline can set the status by invoking function **updateFlightStatus**, and passing parameters like status (0 for ONTIME, 1 for DELAYED and 2 for CANCELLED, By default flight status is NOTSTARTED), flight ID, and delay in hours, in case of ONTIME or CANCELLED, **delayedHours** should be passed as 0.

The screenshot shows the Remix IDE interface. On the left, under 'DEPLOY & RUN TRANSACTIONS', the 'updateFlightStatus' function is highlighted with a red box. The function parameters are set to '\_flightStatus: 2', '\_airlineId: 9988', and '\_delayedHours: 0'. The main editor displays the Solidity code for the 'FlightReservation' contract. The code includes functions for setting ticket price, dynamic ticket price, departure time, and flight status. The 'updateFlightStatus' function is not explicitly shown in the code snippet, but the sidebar indicates it is the function being invoked.

9. There are some extra features like removing ticket from the ticket storage for next flight booking and set dynamic ticket price

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel lists several functions: `bookTicket`, `cancelTicket`, `claimRefund`, `getTicket`, `removeTicket`, `setDepart`, `setDynamic`, `updateFlight`, `getAvailableSeats`, `getFlightStatus`, and `getTicketById`. The `removeTicket` and `setDynamic` functions are highlighted with red rectangles. The main editor shows the Solidity code for the `FlightReservation` contract, including functions for booking, cancellation, and refund processing. The bottom status bar shows a transaction call from `0x58380a6a701c568545dcfc803fc8875f56bedd4` to `FlightReservation.getAvailableSeats()` with data `0x906...d895d`.

```
254 _airlineData.initialSeatNumber = _airlineData.initialSeatNumber + numberOfSeatToBeAddedInPool;
255 ticket.tStatus = TicketStatus.CANCELLED;
256
257 //advance feature to check different slabs for cancellation penalty
258 uint256 _ticketPrice = ticket.ticketPrice;
259 //check if cancellation requested before two hours of departure time
260 if(block.timestamp < _airlineData.departureDateTime - 2*60*60 && block.timestamp > _airlineData.departureDateTime - 6*60*60){
261
262     PenaltyTimeRange memory _penaltyTimeRange = _penaltyTimeRanges[0]; // get slab for 2 hours
263
264     //function call to calculate cancellation penalty
265     calculateCancellationPenalty(_penaltyTimeRange.percentageCancellationPenalty, _ticketPrice, ticket.ticketHolder);
266     ticket.tStatus = TicketStatus.REFUNDED;
267     string memory _cancelMsg = "customer cancelled ticket before 2 hours of departure";
268     emit CancelMessage(_cancelMsg);
269 }
270
271 //check if cancellation requested before 6 hours of departure
272 if(block.timestamp < _airlineData.departureDateTime - 6*60*60 && block.timestamp > _airlineData.departureDateTime - 12*60*60){
273     PenaltyTimeRange memory _penaltyTimeRange = _penaltyTimeRanges[1]; // get slab for 6 hours
274
275     //function call to calculate cancellation penalty
276     calculateCancellationPenalty(_penaltyTimeRange.percentageCancellationPenalty, _ticketPrice, ticket.ticketHolder);
277     ticket.tStatus = TicketStatus.REFUNDED;
278     string memory _cancelMsg = "customer cancelled ticket before 6 hours of departure";
279     emit CancelMessage(_cancelMsg);
280 }
281
282 //check if cancellation requested before 12 hours of departure time
283 if(block.timestamp < _airlineData.departureDateTime - 12*60*60 ){
284     //function call to calculate cancellation penalty
285     calculateCancellationPenalty(_penaltyTimeRange.percentageCancellationPenalty, _ticketPrice, ticket.ticketHolder);
286     ticket.tStatus = TicketStatus.REFUNDED;
287     string memory _cancelMsg = "customer cancelled ticket before 12 hours of departure";
288     emit CancelMessage(_cancelMsg);
289 }
```