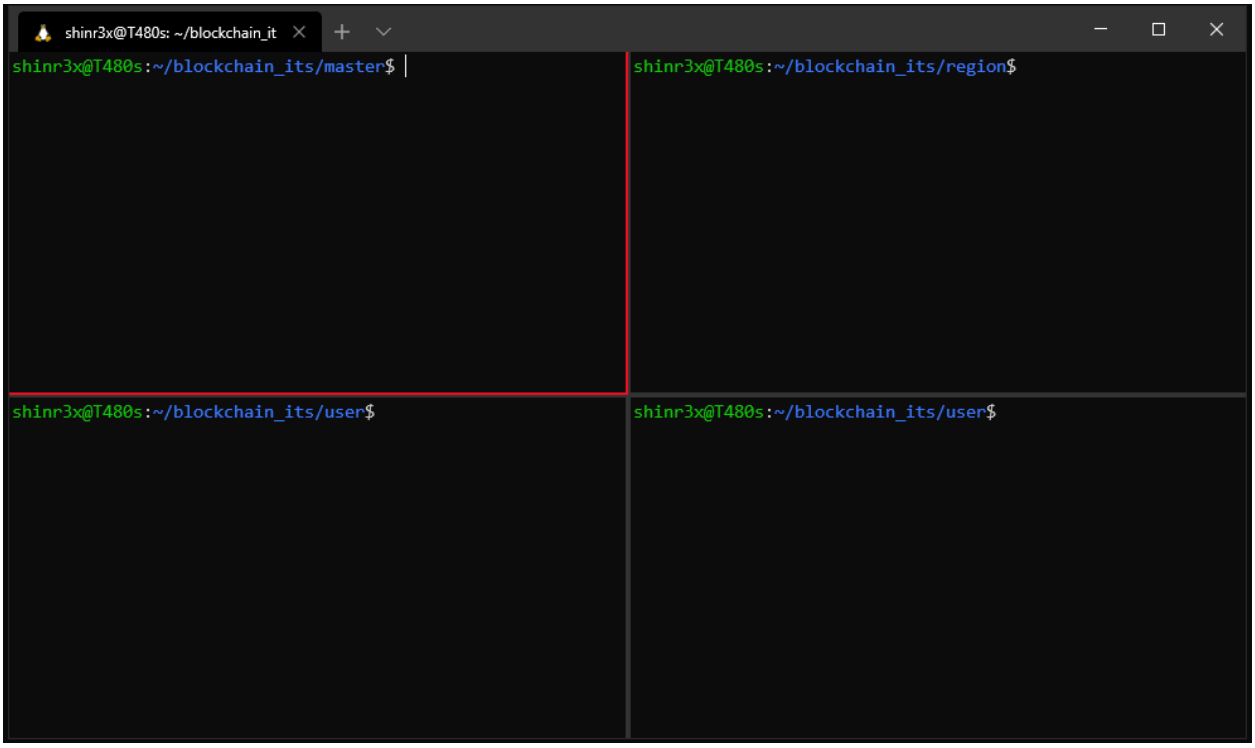
1. Install golang (https://golang.org/doc/install)
2. Download the file projects anywhere on your system (https://github.com/DragosCocirlea/blockchain_its)
3. Go to that path (another terminal for each node)



4. First run the master node using the following command: **go run *.go -port 10000**
Note: the port flag denotes which port the node binds to

When you first run this, the go mods file will make sure that all dependencies are downloaded

```
shinr3x@T480s:~/blockchain_its/master$ go run *.go -port 10000
go: downloading github.com/multiformats/go-multiaddr v0.2.1
go: downloading github.com/libp2p/go-libp2p-core v0.5.3
go: downloading github.com/libp2p/go-libp2p-protocol v0.1.0
go: downloading github.com/libp2p/go-libp2p v0.8.3
go: downloading github.com/libp2p/go-libp2p-net v0.1.0
go: downloading github.com/multiformats/go-varint v0.0.5
go: downloading github.com/multiformats/go-multihash v0.0.13
go: downloading github.com/mr-tron/base58 v1.1.3
go: downloading github.com/minio/sha256-simd v0.1.1
go: downloading github.com/spaolacci/murmur3 v1.1.0
go: downloading golang.org/x/crypto v0.0.0-20200221231518-2aa609cf4a9d
go: downloading github.com/gogo/protobuf v1.3.1
go: downloading github.com/minio/blake2b-simd v0.0.0-20160723061019-3f5f724cb5b1
go: downloading github.com/ipfs/go-cid v0.0.5
go: downloading github.com/jbenet/goprocess v0.1.4
go: downloading github.com/libp2p/go-tcp-transport v0.2.0
go: downloading github.com/libp2p/go-libp2p-peerstore v0.2.3
go: downloading github.com/libp2p/go-buffer-pool v0.0.2
go: downloading github.com/libp2p/go-flow-metrics v0.0.3
go: downloading github.com/libp2p/go-libp2p-mplex v0.2.3
go: downloading github.com/libp2p/go-ws-transport v0.3.1
go: downloading github.com/ipfs/go-log v1.0.4
go: downloading github.com/libp2p/go-libp2p-circuit v0.2.2
go: downloading github.com/libp2p/go-libp2p-yamux v0.2.7
go: downloading github.com/btcsuite/btcd v0.20.1-beta
go: downloading go.opencensus.io v0.22.3
```

5. After the master node is up, it will output the command needed to boot up a region node that connects to this master node.

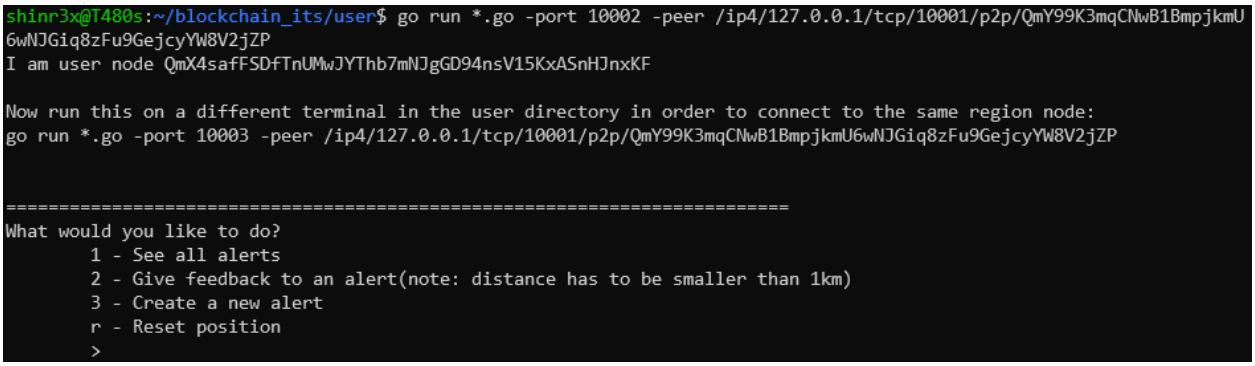
```
go: downloading github.com/libp2p/go-msgio v0.0.4
go: downloading go.uber.org/atomic v1.6.0
go: downloading github.com/google/gopacket v1.1.17
go: downloading github.com/multiformats/go-multiaddr-dns v0.2.0
go: downloading github.com/multiformats/go-base32 v0.0.3
go: downloading github.com/libp2p/go-libp2p-loggables v0.1.0
go: downloading github.com/whyrusleeping/multiaddr-filter v0.0.0-20160516205228-e903e4adabd7
go: downloading github.com/multiformats/go-multistream v0.1.1
go: downloading go.uber.org/multierr v1.5.0
go: downloading github.com/davidlazar/go-crypto v0.0.0-20170701192655-dcfb0a7ac018
go: downloading github.com/koron/go-ssdp v0.0.0-20191105050749-2e1c40ed0b5d
go: downloading github.com/libp2p/go-addr-util v0.0.1
go: downloading github.com/jackpal/go-nat-pmp v1.0.2
go: downloading github.com/huin/goupnp v1.0.0
go: downloading github.com/jbenet/go-temp-err-catcher v0.0.0-20150120210811-aac704a3f4f2
go: downloading github.com/ipfs/go-ipfs-util v0.0.1
go: downloading github.com/google/uuid v1.1.1
go: downloading golang.org/x/net v0.0.0-20190923162816-aa69164e4478
go: downloading github.com/whyrusleeping/mafmt v1.2.8
go: downloading golang.org/x/text v0.3.2
I am master node QmfXySVFjjkNcRtBcenvKj4bAp8wZXHXWx69skX1hJqpnz

Now run this on a different terminal in the region directory:
go run *.go -port 10001 -peer /ip4/127.0.0.1/tcp/10000/p2p/QmfXySVFjjkNcRtBcenvKj4bAp8wZXHXWx69skX1hJqpnz
```

6. Running that command in the region node terminal will first download some more needed dependencies, after which it will output the command needed to boot up a user node that wants to connect to this region node.

```
shinr3x@T480s:~/blockchain_its/region$ go run *.go -port 10001 -peer /ip4/127.0.0.1/tcp/10000/p2p/QmfXySVFjjkNcRtBcen
vKj4bAp8wZXHXWx69skX1hJqpnz
go: downloading github.com/libp2p/go-libp2p-peer v0.2.0
go: downloading github.com/libp2p/go-libp2p-crypto v0.1.0
I am region node QmRKDUkzqGDLDaZovCEk3yTxBVuZ2x83kZVseeGPfuj7wP

Now run this on a different terminal in the user directory after the blockchain parsing has finished:
go run *.go -port 10002 -peer /ip4/127.0.0.1/tcp/10001/p2p/QmRKDUkzqGDLDaZovCEk3yTxBVuZ2x83kZVseeGPfuj7wP

Waiting for blockchain from master node...


Blockchain received:
[
  {
    "Index": 0,
    "Timestamp": "2020-07-08 16:09:56.0179204 +0300 EEST m=+0.211438601",
    "Data": {
      "SpeedReports": null,
      "Alerts": null,
      "UsersReputation": null
    },
    "Hash": "5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9",
    "PrevHash": ""
  }
]
Parsing the blockchain...
```

It will also output the entire blockchain so far. In the case above, the blockchain only contains the genesis block.

7. Running the new command in the user node terminal spawns a new node that sends data to the region node. It will also offer a command needed to connect to the same region node.



8. In the previous cases there are three differences between the nodes:
    a. The port: each node needs to bind to a different port
    b. The name/id: every node has an id which is used to enable other nodes connect to it

9. The system is up and running. Other region nodes can be created, and other user nodes can be connected to the already existing region nodes. Interacting with the alert system is intuitive and other functionalities can be seen in the video.