

MongoDB Sharding

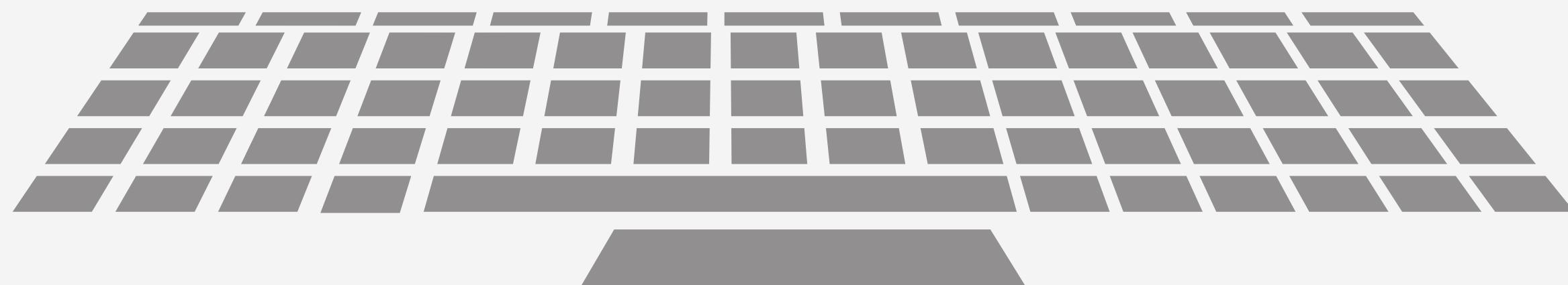
Lab 3

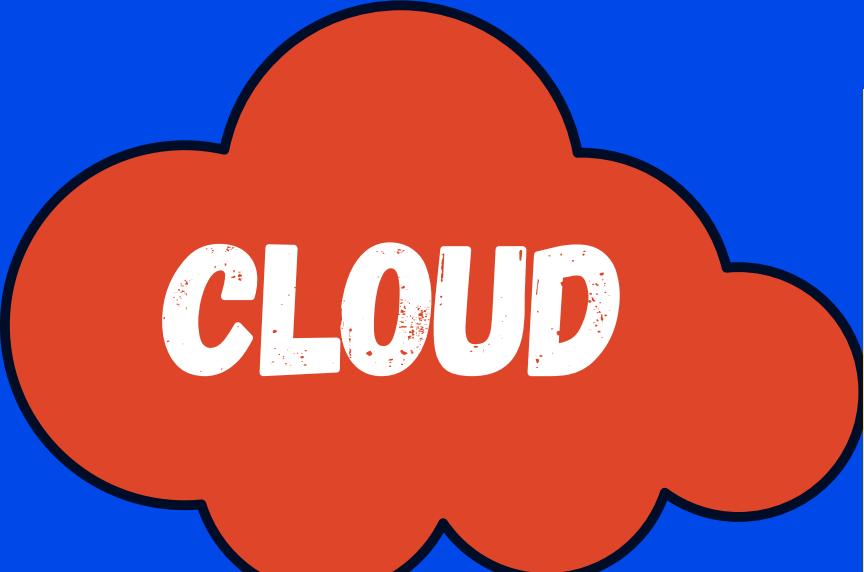
Miguel Glorioso

Dane Rosario

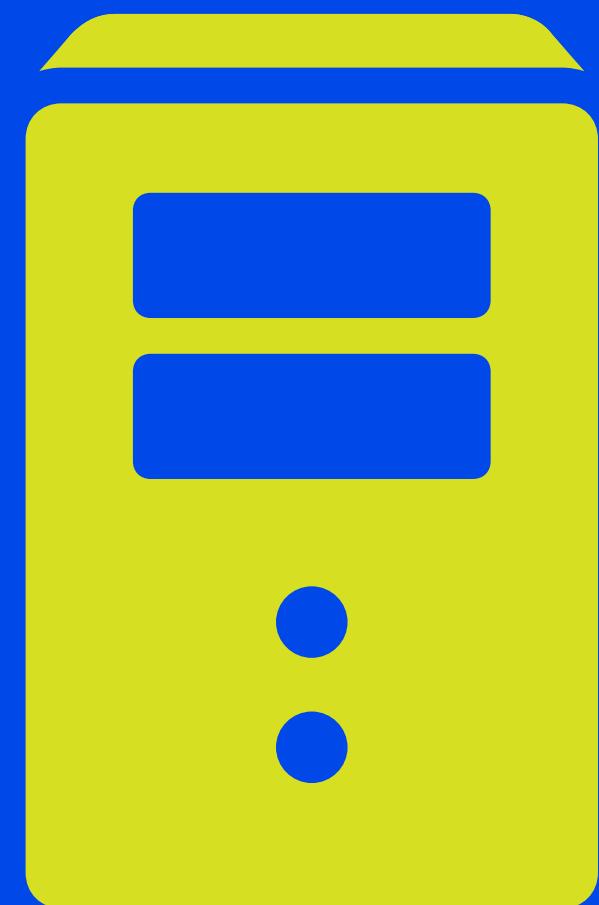
Alex Pino

Jeremy Tan

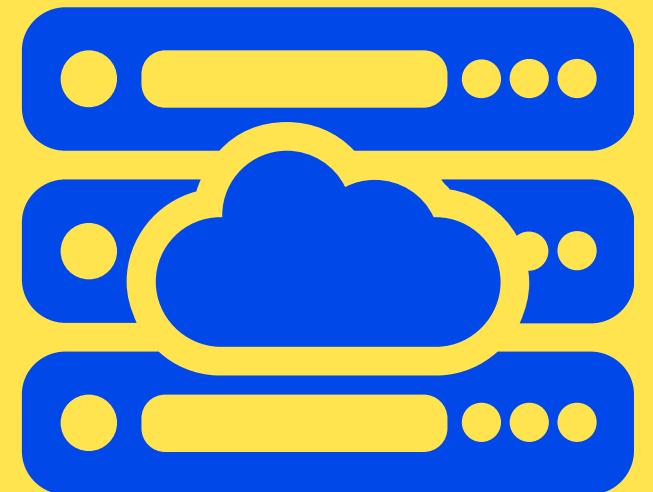




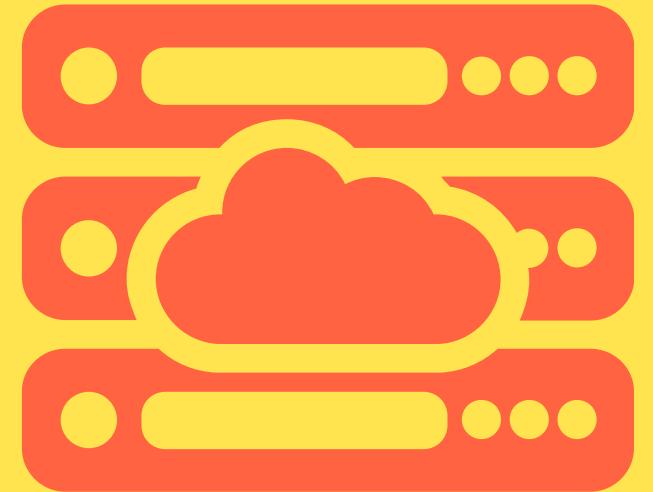
CLOUD



Config



Primary



Secondary

Shards

Primary Secondary



Primary Secondary



Primary Secondary



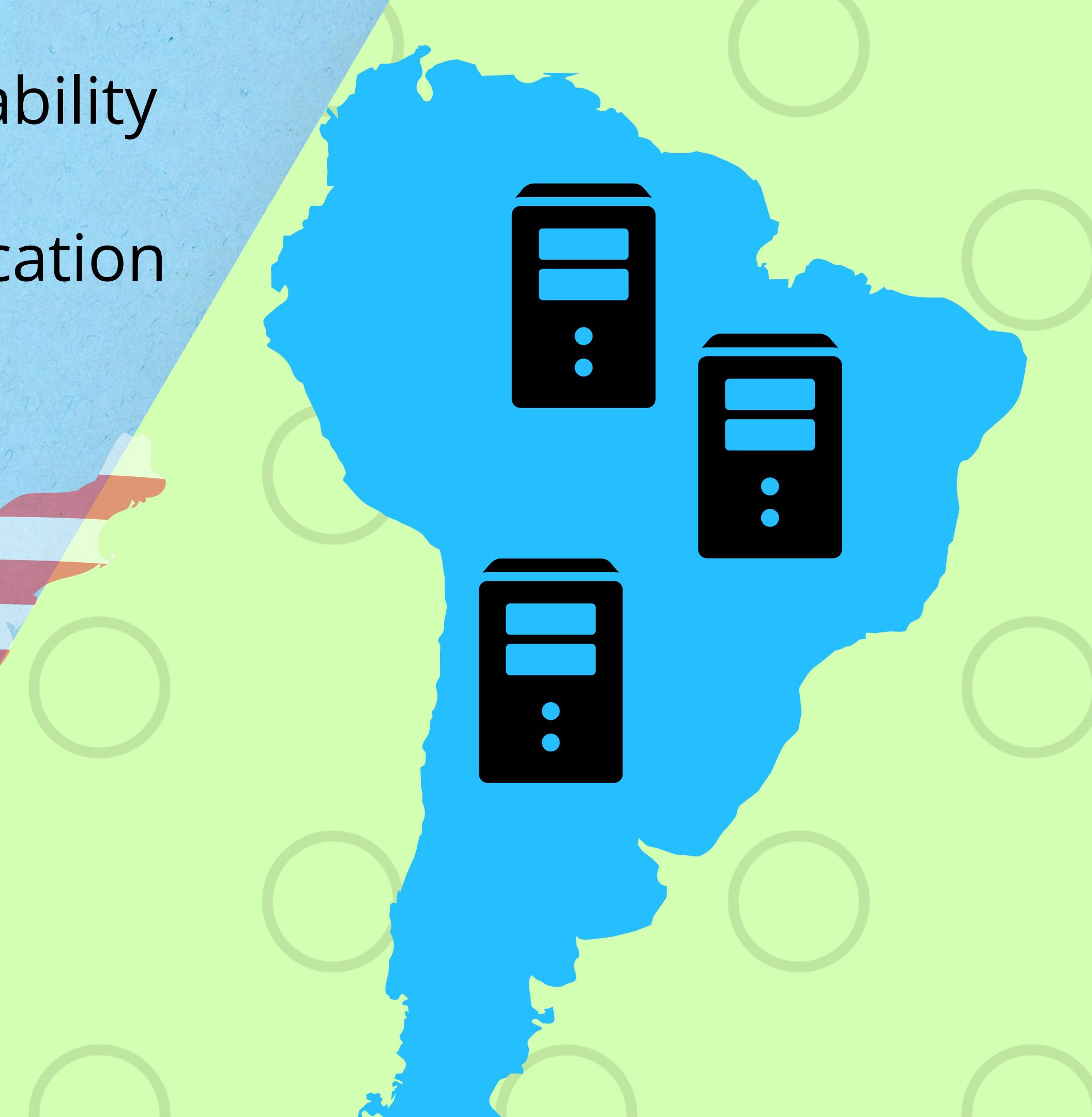
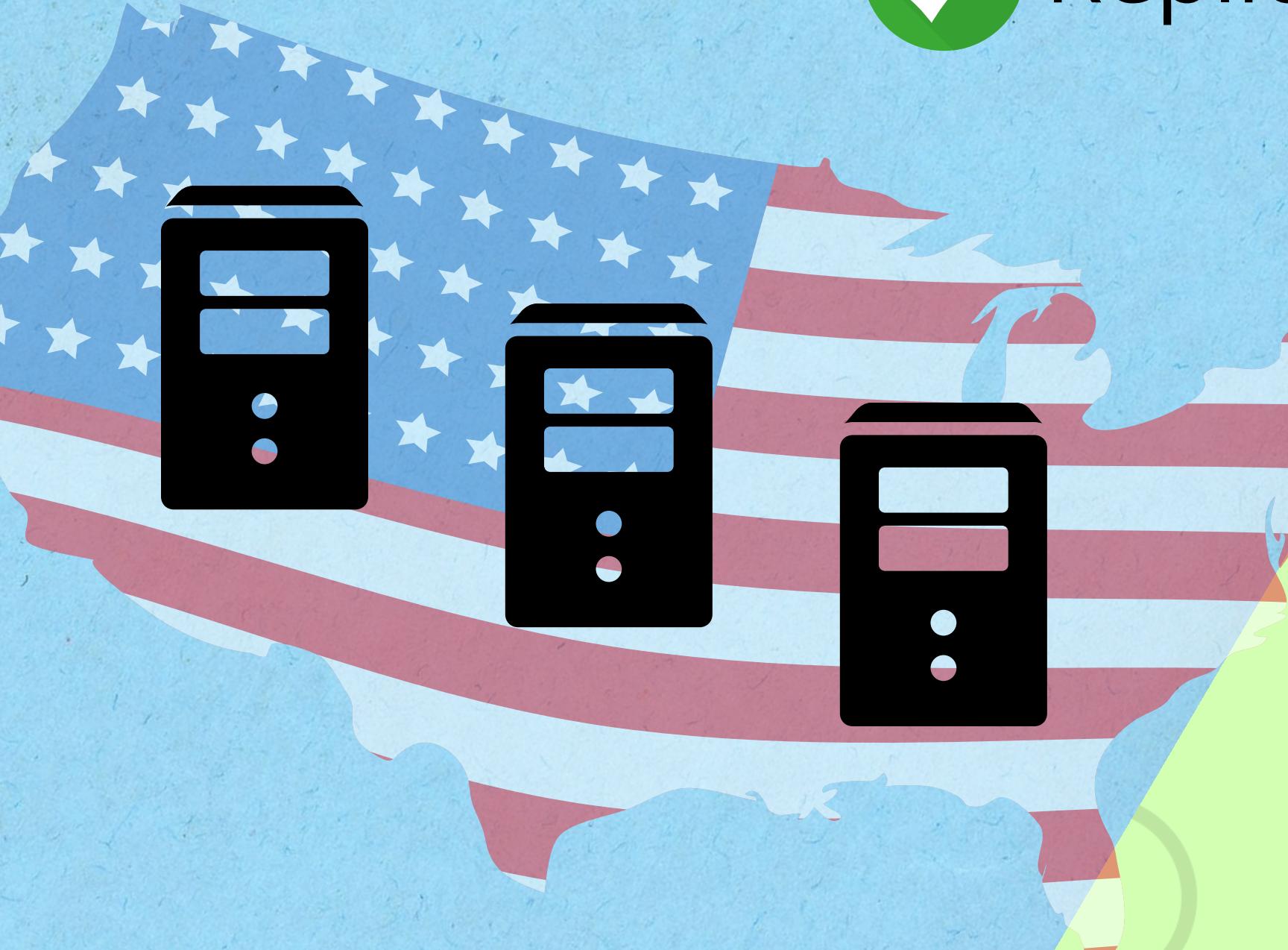
Subnets



Availability



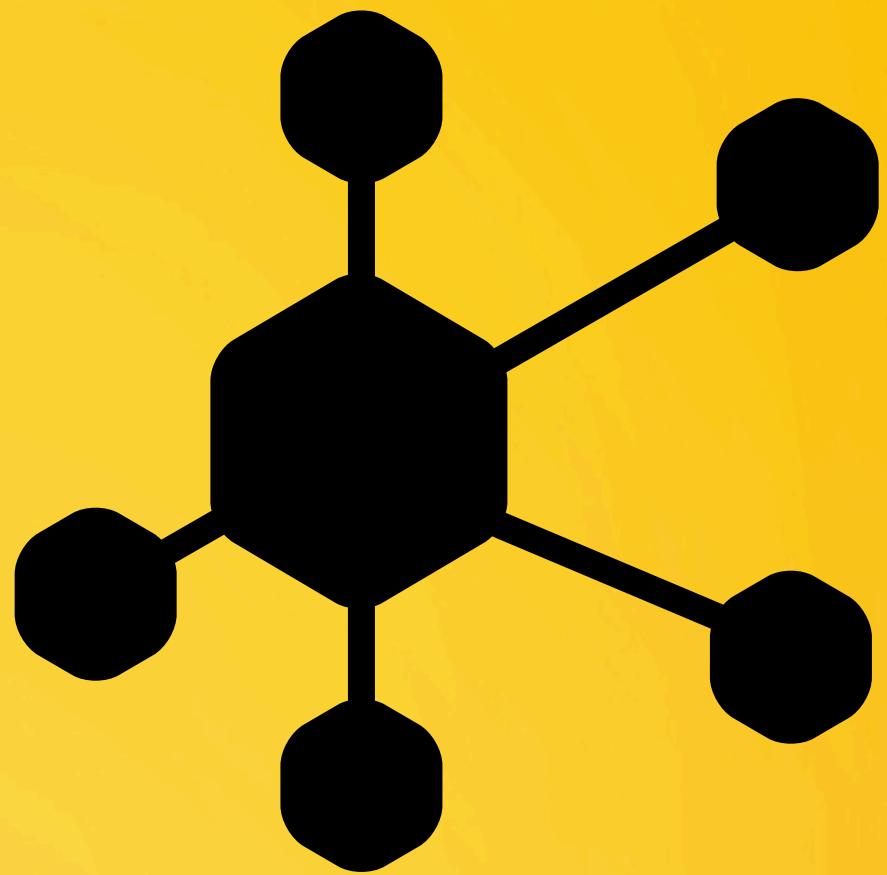
Replication



IP Address

<172.58.216.164>

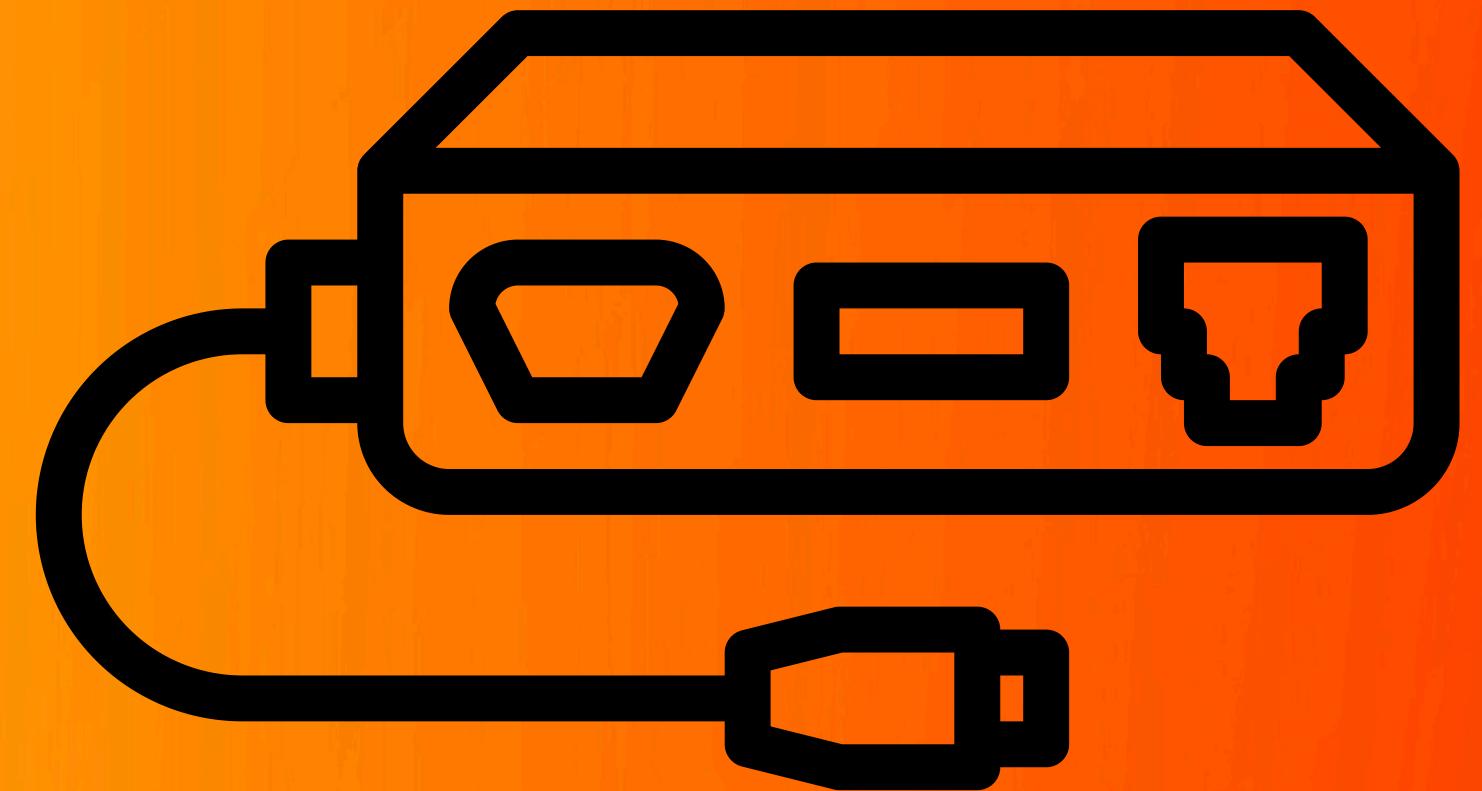
unique address that
identifies a device on
the internet



Port

< 27017 >

a way to identify a specific
process to which an internet is
to be forwarded when it arrives
at a server





*Screenshots of
Outputs from the Lab*



Shard 1

rs.initiate

Go to Anything (Ctrl-P)

ec2-user@ip-172-31-75-8 x +

```
opTime: ISODate("2021-10-13T13:55:20Z"),
    "ts" : Timestamp(1634133320, 4),
    "t" : NumberLong(1)
},
"optimeDate" : ISODate("2021-10-13T13:55:20Z"),
"optimeDurableDate" : ISODate("2021-10-13T13:55:20Z"),
"lastHeartbeat" : ISODate("2021-10-13T13:55:22.445Z"),
"lastHeartbeatRecv" : ISODate("2021-10-13T13:55:23.953Z"),
"pingMs" : NumberLong(1),
"lastHeartbeatMessage" : "",
"syncSourceHost" : "172.31.75.83:27018",
"syncSourceId" : 0,
"infoMessage" : "",
"configVersion" : 1,
"configTerm" : 1
},
],
"ok" : 1,
"$clusterTime" : {
    "clusterTime" : Timestamp(1634133320, 4),
    "signature" : {
        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAA="),
        "keyId" : NumberLong(0)
    }
},
"operationTime" : Timestamp(1634133320, 4)
}
MongoDB Enterprise shardSet1:PRIMARY> +
```

Untitled1 +

```
25
26 replication:
27     replSetName: shardSet1
28 sharding:
29         clusterRole: shardsvr
30
31 sudo systemctl restart mongod.service
32
33
34 replication
35
36 mongo --host 172.31.75.83 --port 27018
37
38 rs.initiate(
39     {
40         _id: "shardSet1",
41         members: [
42             {"_id" : 0, "host" : "172.31.75.83:27018"},
43             {"_id" : 1, "host" : "172.31.61.33:27018"}
44         ]
45     }
46 )
47
48
49 rs.status()
50
51
52
53
54
55
```

(11 Bytes) 49:1 JSON Spaces: 4



Shard 2

rs.initiate

The screenshot shows a terminal window with two tabs. The left tab, titled 'ec2-user@ip-172-31-70-2x', displays a MongoDB command and its JSON response. The right tab, titled 'Untitled1', contains a script for initiating a replication set.

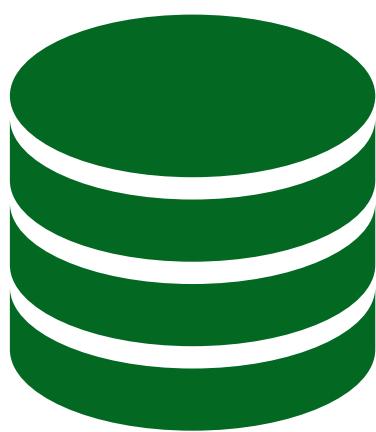
Left Tab (MongoDB Shell Output):

```
ec2-user@ip-172-31-70-2: ~ % mongo
MongoDB Enterprise shardSet2:PRIMARY> rs.initiate(
  {
    "version": 1,
    "shards": [
      {
        "shard": "shard0000",
        "host": "172.31.52.251:27018"
      }
    ],
    "ok": 1,
    "$clusterTime": {
      "clusterTime": Timestamp(1634134033, 1),
      "signature": {
        "hash": BinData(0, "AAAAAAAAAAAAAAAAAAAAAAA=")
      }
    },
    "operationTime": Timestamp(1634134033, 1)
  }
)
MongoDB Enterprise shardSet2:PRIMARY>
```

Right Tab (Script Content):

```
25
26   replication:
27     replSetName: shardSet2
28   sharding:
29     clusterRole: shardsvr
30
31   sudo systemctl restart mongod.service
32
33
34   replication
35
36   mongo --host 172.31.70.23 --port 27018
37
38   rs.initiate(
39     {
40       _id: "shardSet2",
41       members: [
42         {"_id": 0, "host": "172.31.52.251:27018"},
43         {"_id": 1, "host": "172.31.70.23:27018"}
44       ]
45     }
46   )
47
48
49   rs.status()
50
51
52
53
54
55
```

(11 Bytes) 49:1 JSON Spaces: 4

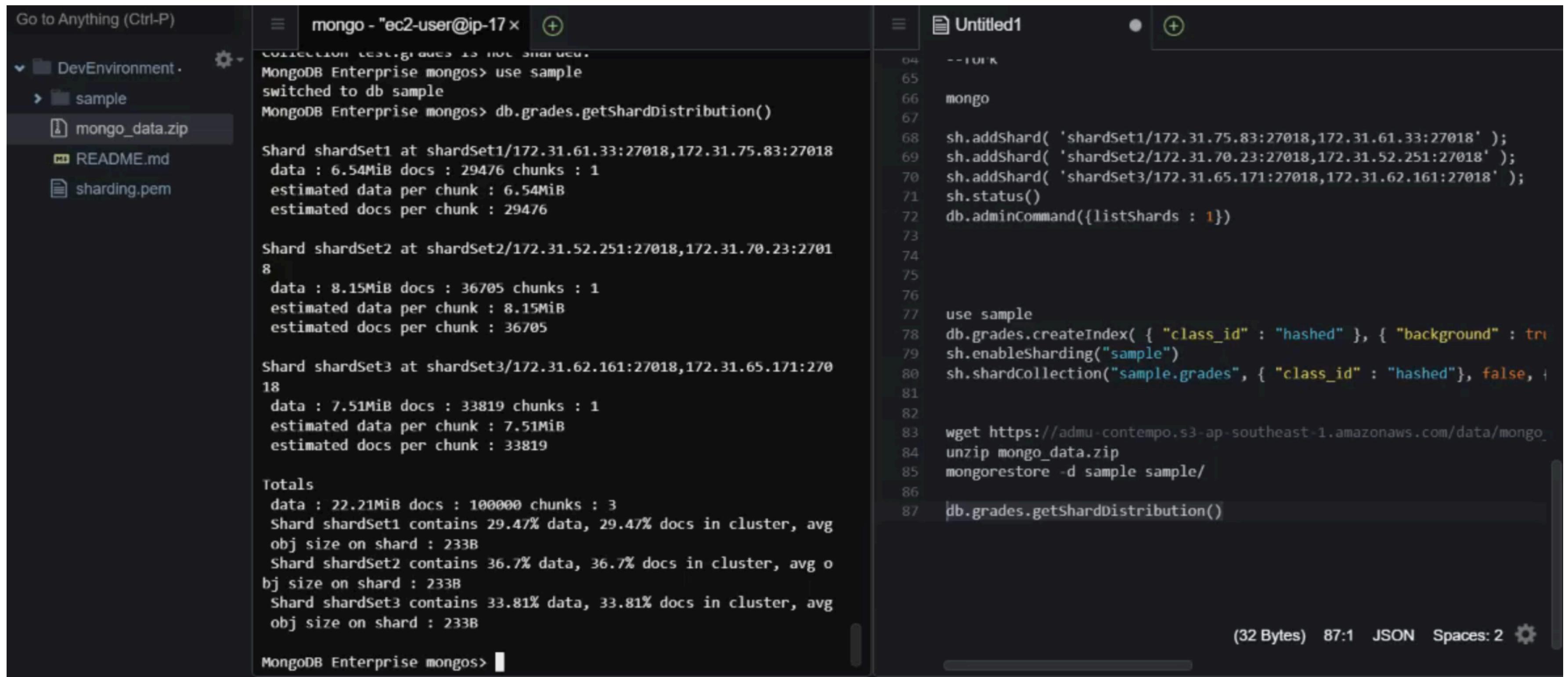


Shard 3

rs.initiate

```
Go to Anything (Ctrl-P)      ec2-user@ip-172-31-65-1x      Untitled1
                               +                                +
                               +-----+
                               | state : 1,
                               | "stateStr" : "PRIMARY",
                               | "uptime" : 49,
                               | "optime" : {
                               |   "ts" : Timestamp(1634134720, 4),
                               |   "t" : NumberLong(1)
                               },
                               | "optimeDate" : ISODate("2021-10-13T14:18:40Z"),
                               | "syncSourceHost" : "",
                               | "syncSourceId" : -1,
                               | "infoMessage" : "Could not find member to sync fr
om",
                               | "electionTime" : Timestamp(1634134720, 1),
                               | "electionDate" : ISODate("2021-10-13T14:18:40Z"),
                               | "configVersion" : 1,
                               | "configTerm" : 1,
                               | "self" : true,
                               | "lastHeartbeatMessage" : ""
                               }
                               ],
                               "ok" : 1,
                               "$clusterTime" : {
                               "clusterTime" : Timestamp(1634134720, 4),
                               "signature" : {
                               "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAA="),
                               "keyId" : NumberLong(0)
                               }
                               },
                               "operationTime" : Timestamp(1634134720, 4)
                           }
MongoDB Enterprise shardSet3:PRIMARY> 25
                               replication:
                               26   replSetName: shardSet3
                               27 sharding:
                               28   clusterRole: shardsvr
                               29
                               30 sudo systemctl restart mongod.service
                               31
                               32
                               33
                               34 replication
                               35
                               36 mongo --host 172.31.65.171 --port 27018
                               37
                               38 rs.initiate(
                               39   {
                               40     _id: "shardSet3",
                               41     members: [
                               42       {"_id" : 0, "host" : "172.31.62.161:27018"},
                               43       {"_id" : 1, "host" : "172.31.65.171:27018"}
                               44     ]
                               45   }
                               46 )
                               47
                               48
                               49 rs.status()
                               50
                               51
                               52
                               53
                               54
                               55
(11 Bytes) 49:1 JSON Spaces: 4 
```

db.grades.getShardDistribution()



The screenshot shows a terminal window with two panes. The left pane displays the output of a MongoDB command, and the right pane shows a shell script.

Left Pane (MongoDB Output):

```
mongo - "ec2-user@ip-17x" +  
COLLECTION TEST.GRADES IS NOT SHARDED.  
MongoDB Enterprise mongos> use sample  
switched to db sample  
MongoDB Enterprise mongos> db.grades.getShardDistribution()  
  
Shard shardSet1 at shardSet1/172.31.61.33:27018,172.31.75.83:27018  
  data : 6.54MiB docs : 29476 chunks : 1  
  estimated data per chunk : 6.54MiB  
  estimated docs per chunk : 29476  
  
Shard shardSet2 at shardSet2/172.31.52.251:27018,172.31.70.23:2701  
 8  
  data : 8.15MiB docs : 36705 chunks : 1  
  estimated data per chunk : 8.15MiB  
  estimated docs per chunk : 36705  
  
Shard shardSet3 at shardSet3/172.31.62.161:27018,172.31.65.171:270  
18  
  data : 7.51MiB docs : 33819 chunks : 1  
  estimated data per chunk : 7.51MiB  
  estimated docs per chunk : 33819  
  
Totals  
  data : 22.21MiB docs : 100000 chunks : 3  
  Shard shardSet1 contains 29.47% data, 29.47% docs in cluster, avg  
  obj size on shard : 233B  
  Shard shardSet2 contains 36.7% data, 36.7% docs in cluster, avg o  
bj size on shard : 233B  
  Shard shardSet3 contains 33.81% data, 33.81% docs in cluster, avg  
  obj size on shard : 233B  
  
MongoDB Enterprise mongos>
```

Right Pane (Shell Script):

```
64 -- TURK  
65  
66 mongo  
67  
68 sh.addShard( 'shardSet1/172.31.75.83:27018,172.31.61.33:27018' );  
69 sh.addShard( 'shardSet2/172.31.70.23:27018,172.31.52.251:27018' );  
70 sh.addShard( 'shardSet3/172.31.65.171:27018,172.31.62.161:27018' );  
71 sh.status()  
72 db.adminCommand({listShards : 1})  
73  
74  
75  
76  
77 use sample  
78 db.grades.createIndex( { "class_id" : "hashed" }, { "background" : true } )  
79 sh.enableSharding("sample")  
80 sh.shardCollection("sample.grades", { "class_id" : "hashed" }, false, {  
81  
82  
83 wget https://admu-contempo.s3-ap-southeast-1.amazonaws.com/data/mongo_  
84 unzip mongo_data.zip  
85 mongorestore -d sample sample/  
86  
87 db.grades.getShardDistribution()
```

(32 Bytes) 87:1 JSON Spaces: 2 

AWS: (not connected)

Full Details of the instances in our Lab

IP Addresses & Subnets

Instance	IP Address	Subnet	Port
Cloud9	172.31.89.118	us-east-1a	27017
Config F	172.31.66.136	us-east-1f	27019
Config E	172.31.57.234	us-east-1e	27019
Shard 1F	172.31.75.83	us-east-1f	27018
Shard 1E	172.31.61.33	us-east-1e	27018
Shard 2F	172.31.70.23	us-east-1f	27018
Shard 2E	172.31.52.251	us-east-1e	27018
Shard 3F	172.31.65.171	us-east-1f	27018
Shard 3E	172.31.62.161	us-east-1e	27018

Miguel Glorioso

Video editing,
voiceovers

Alexander Pino

Wrote the lab
instructions, script

Dane Rosario

Video Recording

Jeremy Tan

Powerpoint,
voiceovers