

## E. Queda 1 ou 2 nós secundários

Mostrar todas as Tabelas

```
db.cliente.find()
```

---

**\*\*Derrubar nós**

Derrubar apenas 2, secundários

**\*\*Cuidado pra não derrubar o primário sem querer agora**

```
docker stop node1
```

```
docker stop node2
```

```
docker stop node3
```

```
docker stop node4
```

---

Verificar status dos nós caídos

```
docker exec -it node1 mongosh --eval "rs.status()"
```

nas conexões que derruba-mos

```
db.cliente.find()
```

-> é pra dar erro mesmo

no node primario

```
db.cliente.insertOne({codigo:6, nome: "Saito"});
```

-> Inserir para mostrar nos nós que vamos recuperar

---

Voltar nós derrubados

```
docker run -d --rm -p 27017:27017 --name node1 --network cluster
mongodb/mongodb-community-server:latest --replSet replicaSet --bind_ip
localhost,node1
```

```
docker run -d --rm -p 27018:27017 --name node2 --network cluster
mongodb/mongodb-community-server:latest --replSet replicaSet --bind_ip
localhost,node2
```

```
docker run -d --rm -p 27019:27017 --name node3 --network cluster
mongodb/mongodb-community-server:latest --replSet replicaSet --bind_ip
localhost,node3
```

```
docker run -d --rm -p 27020:27017 --name node4 --network cluster
mongodb/mongodb-community-server:latest --replSet replicaSet --bind_ip
localhost,node4
```

---

nos nós que voltaram

```
use sistemaCorp
```

Verificar a nova tabela inserida a pouco

```
db.cliente.find()
```