

1. Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.network "public_network", bridge: "Intel"
  onfig.vm.box = "ubuntu/trusty64"
end
```

vagrant up:

vm1: port number 2200

vm2: port number 2222

vm3: port number 2201

2. Create three directories such as vm1, vm2, and vm3 and save the above Vagrantfile in each directory.

VM1: ssh -p 2200 vagrant@127.0.0.1

VM2: ssh -p 2222 vagrant@127.0.0.1

VM3: ssh -p 2201 vagrant@127.0.0.1

Check their IP address through ifconfig

VM1: 192.168.1.73

VM2: 192.168.1.72

VM3: 192.168.1.74

Change their host name:

VM1: sudo hostname active1

VM2: sudo hostname active2

VM3: sudo hostname passive1

Add IP address with hostnames into file /etc/hosts.

Sudo vim /etc/hosts

127.0.0.1 localhost

192.168.1.73 active1

192.168.1.72 active2

192.168.1.74 passive1

Modify bind_ip in /etc/mongod.conf in all three virtual machines:

Sudo vim /etc/mongod.conf

net:

port: 27017

bind_ip: 192.168.1.73

net:

port: 27018

bind_ip: 192.168.1.72

net:

port: 27019

bind_ip: 192.168.1.74

3. On three cmd window:

Create directory in each of them:

sudo mkdir -p /db/active1/data

```
sudo mkdir -p /db/active2/data
sudo mkdir -p /db/passive1/data
vagrant@active1:~$ sudo mongod --port 27017 -dbpath /db/active1/data --replSet
myReplicaSet --smallfiles --oplogSize 128 --bind_ip 192.168.1.73
vagrant@active2:~$ sudo mongod --port 27018 -dbpath /db/active2/data --replSet
myReplicaSet --smallfiles --oplogSize 128 --bind_ip 192.168.1.72
vagrant@passive1:~$ sudo mongod --port 27019 -dbpath /db/passive1/data --replSet
myReplicaSet --smallfiles --oplogSize 128 --bind_ip 192.168.1.74
```

4. On the fourth cmd window:

```
mongo 192.168.1.73:27017
```

```
rs.initiate()
```

```
rs.status()
```

```

{
  "set" : "myReplicaSet",
  "date" : ISODate("2019-03-18T03:47:04.725Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "syncingTo" : "",
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1552880818, 2),
      "t" : NumberLong(1)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1552880818, 2),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1552880818, 2),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1552880818, 2),
      "t" : NumberLong(1)
    }
  },
  "lastStableCheckpointTimestamp" : Timestamp(1552880816, 1),
  "members" : [
    {
      "_id" : 0,
      "name" : "192.168.1.73:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 880,
      "optime" : {
        "ts" : Timestamp(1552880818, 2),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2019-03-18T03:46:58Z"),
      "syncingTo" : "",
      "syncSourceHost" : "",
      "syncSourceId" : -1,
      "infoMessage" : "could not find member to sync from",
      "electionTime" : Timestamp(1552880815, 1),
      "electionDate" : ISODate("2019-03-18T03:46:55Z"),
      "configVersion" : 1,
      "self" : true,
      "lastHeartbeatMessage" : ""
    }
  ]
}

```

```

myReplicaSet:PRIMARY> rs.add("192.168.1.72:27018")
{
  "ok" : 1,
  "operationTime" : Timestamp(1552880844, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552880844, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}

myReplicaSet:PRIMARY> rs.add("192.168.1.74:27019")
{
  "ok" : 1,
  "operationTime" : Timestamp(1552881006, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552881006, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}

```

```

myReplicaSet:PRIMARY> rs.status()
{
  "set" : "myReplicaSet",
  "date" : ISODate("2019-03-18T03:50:12.509Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "syncingTo" : "",
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1552881006, 1),
      "t" : NumberLong(1)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1552881006, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1552881006, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1552881006, 1),
      "t" : NumberLong(1)
    }
  },
  "lastStableCheckpointTimestamp" : Timestamp(1552880997, 1),
  "members" : [
    {
      "id" : 0,
      "name" : "192.168.1.73:27017",
      "health" : 1,
      "state" : 1,
      "stateStr" : "PRIMARY",
      "uptime" : 1068,
      "optime" : {
        "ts" : Timestamp(1552881006, 1),
        "t" : NumberLong(1)
      },
      "optimeDate" : ISODate("2019-03-18T03:50:06Z"),
      "syncingTo" : "",
      "syncSourceHost" : "",
      "syncSourceId" : -1,
      "infoMessage" : "",
      "electionTime" : Timestamp(1552880815, 1),
      "electionDate" : ISODate("2019-03-18T03:46:55Z"),
      "configVersion" : 3,
      "self" : true,
      "lastHeartbeatMessage" : ""
    }
  ]
}

```

```

    "_id" : 1,
    "name" : "192.168.1.72:27018",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 177,
    "optime" : {
      "ts" : Timestamp(1552881017, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1552881017, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2019-03-18T03:50:17Z"),
    "optimeDurableDate" : ISODate("2019-03-18T03:50:17Z"),
    "lastHeartbeat" : ISODate("2019-03-18T03:50:20.484Z"),
    "lastHeartbeatRecv" : ISODate("2019-03-18T03:50:22.399Z"),
    "pingMs" : NumberLong(4),
    "lastHeartbeatMessage" : "",
    "syncingTo" : "192.168.1.73:27017",
    "syncSourceHost" : "192.168.1.73:27017",
    "syncSourceId" : 0,
    "infoMessage" : "",
    "configVersion" : 3
  },
  {
    "_id" : 2,
    "name" : "192.168.1.74:27019",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 15,
    "optime" : {
      "ts" : Timestamp(1552881017, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1552881017, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2019-03-18T03:50:17Z"),
    "optimeDurableDate" : ISODate("2019-03-18T03:50:17Z"),
    "lastHeartbeat" : ISODate("2019-03-18T03:50:20.495Z"),
    "lastHeartbeatRecv" : ISODate("2019-03-18T03:50:22.009Z"),
    "pingMs" : NumberLong(0),
    "lastHeartbeatMessage" : "",
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "infoMessage" : "",
    "configVersion" : 3
  }
]

```

5. Use example to check if replica set has been implemented correctly.

```

myReplicaSet:PRIMARY> for(var i = 0; i < 1000; i++) db.exampleCollection.insert({x: i})
WriteResult({"nInserted" : 1 })
myReplicaSet:PRIMARY> db.exampleCollection.find()
{"_id" : ObjectId("5c8f1a15dcb99fe6e3879b9c"), "x" : 0 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879b9d"), "x" : 1 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879b9e"), "x" : 2 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879b9f"), "x" : 3 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba0"), "x" : 4 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba1"), "x" : 5 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba2"), "x" : 6 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba3"), "x" : 7 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba4"), "x" : 8 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba5"), "x" : 9 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba6"), "x" : 10 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba7"), "x" : 11 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba8"), "x" : 12 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba9"), "x" : 13 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879baa"), "x" : 14 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bab"), "x" : 15 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bac"), "x" : 16 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bad"), "x" : 17 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bae"), "x" : 18 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879baf"), "x" : 19 }
Type "it" for more
myReplicaSet:PRIMARY> db.exampleCollection.count()
1000
myReplicaSet:PRIMARY>

```

```

myReplicaSet:SECONDARY> db.getMongo().setSlaveOk()
myReplicaSet:SECONDARY> use exampleDB
switched to db exampleDB
myReplicaSet:SECONDARY> db.exampleCollection.find()
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879baa"), "x" : 14 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bab"), "x" : 15 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bbf"), "x" : 35 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bbc"), "x" : 32 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bb2"), "x" : 22 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bb3"), "x" : 23 }
{"_id" : ObjectId("5c8f1a15dcb99fe6e3879b9c"), "x" : 0 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba2"), "x" : 6 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bb4"), "x" : 24 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bb7"), "x" : 27 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bb6"), "x" : 26 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bbe"), "x" : 34 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bc0"), "x" : 36 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bae"), "x" : 18 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879baf"), "x" : 19 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bb9"), "x" : 29 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879ba0"), "x" : 4 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bbd"), "x" : 33 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879bac"), "x" : 16 }
{"_id" : ObjectId("5c8f1a16dcb99fe6e3879b9d"), "x" : 1 }
Type "it" for more
myReplicaSet:SECONDARY> db.exampleCollection.count()
1000
myReplicaSet:SECONDARY>

```