# **Mongo 3.4 Installation**

**Linux-Ubuntu Distro:**

Follow the below steps to install the Mongo 3.4.

1. By default Ubuntu comes with a mongo package which we can use to install but no guarantee that we would always get the latest version.

|  |
| --- |
| sudo apt-cache policy mongodb-org |

1. So use the below command to Download the binary and package from the mongodb site and install it with the required keys on your system.

|  |
| --- |
| sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC711F9BA15703C6 |

|  |
| --- |
| echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list |

1. You must need to inform the Ubuntu package manager to update the downloaded repository packages via below command.

|  |
| --- |
| sudo apt-get update |

1. Once the packages updated, now install the mongodb latest version using the below command.

|  |
| --- |
| sudo apt install mongodb-org=3.4.21 mongodb-org-server=3.4.21 mongodb-org-shell=3.4.21 mongodb-org-mongos=3.4.21 mongodb-org-tools=3.4.21 |

1. After Installation start and check the status of the mongod service.

|  |
| --- |
| sudo systemctl start mongod |

|  |
| --- |
| sudo systemctl status mongod |

1. Enable the mongod service to start at system bootup using below command to avoid manually starting it up.

|  |
| --- |
| sudo systemctl enable mongod |

**Windows:**

Follow the below steps:

1. Download the mongodb installer from this URL <http://www.mongodb.org/downloads>
2. Select the required version of Mongodb for your system either 32 or 64 bit.
3. Once downloaded, double click the .msi file from the folder where you have downloaded and follow the options to complete the installation.
4. By default if you don’t change the path, mongo will installed on this path only **C:\Program Files\MongoDB\Server\3.4\**
5. Now, setup the mongo db environment as it requires a data directory to store all data.

|  |
| --- |
| md \data\db  "C:\Program Files\MongoDB\Server\3.4\bin\mongod.exe" --dbpath d:\test\mongodb\data  "C:\Program Files\MongoDB\Server\3.4\bin\mongod.exe" --dbpath "d:\test\mongo db data" |

1. After above all steps, execute the mongod.exe service to start the Mongodb via below command on your command prompt.

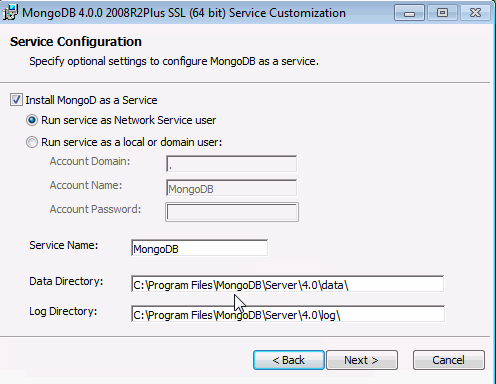
|  |
| --- |
| "C:\Program Files\MongoDB\Server\3.4\bin\mongod.exe" |

# **Mongo 4.2 Installation**

Follow the below steps to Install Mongo 4.2

**Windows:**

1. Download the mongodb installer from this URL <https://www.mongodb.com/try/download/community?tck=docs_server>
2. Select the appropriate version which you require and download it to any directory which you want.
3. After download go to the installed directory and click the .msi file to start the installation.
4. Choose the complete setup option and select the option “Install mongodb as a service”.
5. Refer the below snap details and you can modify the details as per your need.



1. Once Installation done. Open a command prompt and run a below command to start the mongodb service

|  |
| --- |
| "C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe" |

**Linux-Ubuntu Distro:**

1. Install the required Mongo public key for package update to get the latest version.

Note: After executing the below command, the response should be Ok.

|  |
| --- |
| sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 4B7C549A058F8B6B |

1. If you receive any error indicating gnupg while importing the key, then install the gnupg using the below command and execute the above command once again.

Note: Ignore the step 2 if you successfully get Ok at your first step.

|  |
| --- |
| sudo apt-get install gnupg |

1. Create a list file and import the repository to get the 4.2 version.

|  |
| --- |
| echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb.list |

1. You must need to inform the Ubuntu package manager to update the downloaded repository packages via below command.

|  |
| --- |
| sudo apt-get update |

1. Once the packages updated, now install the mongodb latest version using the below command.

|  |
| --- |
| sudo apt install mongodb-org=4.2.8 mongodb-org-server=4.2.8 mongodb-org-shell=4.2.8 mongodb-org-mongos=4.2.8 mongodb-org-tools=4.2.8 |

1. After Installation start and check the status of the mongod service.

|  |
| --- |
| sudo systemctl start mongod |

|  |
| --- |
| sudo systemctl status mongod |

1. Enable the mongod service to start at system bootup using below command to avoid manually starting it up.

|  |
| --- |
| sudo systemctl enable mongod |

# 

# **Authentication**

Follow the steps below to authenticate

**Set up Mongodb Authentication:**

**Generate a key File:**

1. Issue this command to generate your key file

|  |
| --- |
| openssl rand -base64 756 > mongo-keyfile |

Once you’ve generated the key, copy it to each member of your replica set.

1. The rest of the steps in this section should be performed on each member of the replica set, so that they all have the key file located in the same directory, with identical permissions. Create the /opt/mongo directory to store your key file.

|  |
| --- |
| sudo mkdir /opt/mongo |

1. Assuming that your key file is under the home directory for your user, move it to /opt/mongo, and assign it the correct permissions:

|  |
| --- |
| sudo mv ~/mongo-keyfile /opt/mongo |

|  |
| --- |
| sudo chmod 400 /opt/mongo/mongo-keyfile |

1. Update the ownership of your key file, so that it belongs to the MongoDB user. Use the appropriate command for your distribution:

|  |
| --- |
| sudo chown mongodb:mongodb /opt/mongo/mongo-keyfile |

**Create an Admin user:**

1. Inside Mongo shell run the below commands.
2. Connect to the admin database

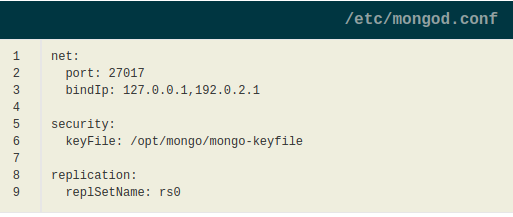
|  |
| --- |
| use admin |

1. Create an administrative user with root privileges. Replace “password” with a strong password of your choice:

|  |
| --- |
| db.createUser({user: "mongo-admin", pwd: "password", roles:[{role: "root", db: "admin"}]}) |

**Configure MongoDB**

1. Make the below following changes to your /etc/mongod.conf file



1. Save the changes and restart the mongod service

|  |
| --- |
| sudo systemctl restart mongod |

1. Run the below command to authenticate the database

|  |
| --- |
| db.auth( "username", passwordPrompt() ) |

1. Use the below command for logout.

|  |
| --- |
| Logout: { logout: 1 } |

# Replica Set Configuration

Follow the below steps to perform Replica set connection with mongo nodes.

Pre-requisites:

1. 3 Nodes(Instance,VM’s)
2. 2 Nodes and 1 Arbiter (Arbiter requires if you choose even no cluster)

Steps:

1. Install MongoDB on each node using the above Installation steps.
2. Check whether the mongod service is active or not, if not enable it using below command.

|  |
| --- |
| sudo systemctl enable mongod.service |

1. If any error happens and service is still showing as failed, follow the below troubleshooting steps.

|  |
| --- |
| **Troubleshooting-1**  sudo systemctl unmask mongod  sudo systemctl restart mongod  **Troubleshooting-2**  sudo rm /var/lib/mongodb/mongod.lock  mongod --repair  sudo systemctl restart mongod  **Troubleshooting-3**  sudo systemctl stop mongod  sudo apt-get purge mongodb-org\*  sudo rm -r /var/log/mongodb  sudo rm -r /var/lib/mongodb  Install Mongodb once again |

1. Once Mongodb installed and the service is active/running, run the below command once to see its working fine.

|  |
| --- |
| mongod |

1. Mongodb by default stores the data at /var/lib/mongodb, so if you want to change the db path then create a new directory and allow permissions as an ownership with mongodb.

|  |
| --- |
| sudo mkdir -p /data/db  sudo chmod 755 /data/db  sudo chown -R mongodb:mongodb /data/db |

1. Also it stores the log by default at /var/log/mongodb/mongo.log file, just check the file permission once as it doesn’t come with required permissions. Allow the permissions using the below command.

|  |
| --- |
| sudo chmod -R 755 /var/log/mongodb  sudo chmod 666 /var/log/mongodb/mongo.log |

1. Once done, edit the mongo configuration file which is located at /etc/mongod.conf and add the primary domain name or IP address along with replication name as per below.

|  |
| --- |
| sudo vi /etc/mongod.conf  # mongod.conf  # for documentation of all options, see:  # http://docs.mongodb.org/manual/reference/configuration-options/  # Where and how to store data.  storage:  **dbPath: /var/lib/mongodb (Here is the place you need to define your storage path if you want to change)**  journal:  enabled: true  # engine:  # mmapv1:  # wiredTiger:  # where to write logging data.  systemLog:  destination: file  logAppend: true  path: /var/log/mongodb/mongod.log  # network interfaces  **net:**  **port: 27017**  **bindIp: <domain-name or IP of your primary>,127.0.0.1 (Must keep localhost here)**  #processManagement:  #security:  #operationProfiling:  **replication:**  **replSetName: "<Any replica name you wish>"**  #sharding:  ## Enterprise-Only Options:  #auditLog:  #snmp: |

1. Restart the service after the above changes are done.
2. Follow the same in all other replica set nodes, but you must need to mention the same replica name for all the nodes.
3. Also mention all the nodes private ip with any name on /etc/hosts file, mongo primary will refer to the hosts file to communicate with other nodes.
4. Go to the mongo shell now, and run the below commands to make a replica set connection with the primary node.

|  |
| --- |
| mongo  > rs.initiate()  If you get ok as a output, then follow the next  > rs.add(“<domain-name or IP or 2nd node:27017>”)  If you get ok as an output, then the node has been connected.  >rs.conf()  Confirm the connected nodes via this command  >rs.status() or rs.isMaster()  Confirm the node which your in  >rs.slaveOk()  Run this command on secondary nodes to execute the db commands there with master rights.  Or  rs.initiate( {  \_id : "rs0",  members: [  { \_id: 0, host: "mongodb-primary:27017" }]})  rs.add(“mongodb-replica-1:27017”)  rs.add(“mongodb-replica-2:27017”) |

1. You can also check the connection between other nodes from your primary node using below commands.

|  |
| --- |
| mongo -host <domain-name or IP of other node> -port 27017 |

1. Create some records on primary db and check the same in secondary dbs to confirm whether it's synchronized or not.

# Backup & Restore DB

Follow the below steps to back up and restore the mongodb data.

**Backup:**

1. Create a directory on your host for your backup.

|  |
| --- |
| sudo mkdir /var/backups/mongobackups |

1. Use the mongodump command to take the back up per below

|  |
| --- |
| sudo mongodump --db <db-name> --connection <connection-name> --host <host-name> --port <any-port> --password []--out /var/backups/mongobackups/backup.db |

**Restore:**

1. Restore the backup back to the database as per below command.

|  |
| --- |
| sudo mongorestore --db <db-name> --connection <connection-name> --host <host-name> --port <any-port> --password [] --drop /var/backups/mongobackups/backup.db |

# 