Mongodb replica setup for production Environment on Red hat linux

====================================================

**Note:**This installation only supports 64-bit systems.

**Before installing mongodb** we have to make the changes at OS level.

**i**.Transparent Huge Pages

**ii.**.Linux Ulimit

**iii.**Virtual Memory

**iv**.Swappiness

**v**.Network Stack

**vi**.Security Enhanced Linux

From **Root user** we need to make the changes in OS level.

1. **Disable THP** by saving the following file to **/etc/init.d/disable-transparent-hugepages**.

**Transparent Huge Pages (THP)** is a Linux memory management system that reduces the overhead of Translation Lookaside Buffer (TLB) lookups on machines with large amounts of memory by using larger memory pages.

**#vim /etc/init.d/disable-transparent-hugepages**

#!/bin/bash

### BEGIN INIT INFO

# Provides: disable-transparent-hugepages

# Required-Start: $local\_fs

# Required-Stop:

# X-Start-Before: mongod mongodb-mms-automation-agent

# Default-Start: 2 3 4 5

# Default-Stop: 0 1 6

# Short-Description: Disable Linux transparent huge pages

# Description: Disable Linux transparent huge pages, to improve

# database performance.

### END INIT INFO

case $1 in

start)

if [ -d /sys/kernel/mm/transparent\_hugepage ]; then

thp\_path=/sys/kernel/mm/transparent\_hugepage

elif [ -d /sys/kernel/mm/redhat\_transparent\_hugepage ]; then

thp\_path=/sys/kernel/mm/redhat\_transparent\_hugepage

else

return 0

fi

echo 'never' > ${thp\_path}/enabled

echo 'never' > ${thp\_path}/defrag

re='^[0-1]+$'

if [[ $(cat ${thp\_path}/khugepaged/defrag) =~ $re ]]

then

# RHEL 7

echo 0 > ${thp\_path}/khugepaged/defrag

else

# RHEL 6

echo 'no' > ${thp\_path}/khugepaged/defrag

fi

unset re

unset thp\_path

;;

esac

Copy and paste it in the above path “**/etc/init.d/disable-transparent-hugepages**”.

**#chmod 755 /etc/init.d/disable-transparent-hugepages**

**#update-rc.d disable-transparent-hugepages defaults**

**Note:**Repeat on each node with above steps and reboot to enable the changes.

**2.Linux Ulimit**

The value of ulimit is one of the mechanisms used by Unix OSs such as Linux to prevent a single user from using too many system resources, such as files, threads, network connections, etc.

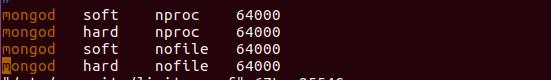
**#vim /etc/security/limits.conf**

mongod soft nproc 64000

mongod hard nproc 64000

mongod soft nofile 64000

mongod hard nofile 64000



**Note:** This change only applies to new shells, meaning you must restart mongod.

**Recommended values for mongodb:**

-f (file size): unlimited

-t (cpu time): unlimited

-v (virtual memory): unlimited [1]

-l (locked-in-memory size): unlimited

-n (open files): 64000

-m (memory size): unlimited [1] [2]

-u (processes/threads): 64000

**3.**Make the changes in sysctl configuration file for **Virtual memory**

**#vim /etc/sysctl.conf**

vm.dirty\_ratio=15

vm.dirty\_background\_ratio=5

**4.Swapping memory** in the below configuration file with valid parameter values .

#**vim /etc/sysctl.conf**

vm.swappiness=1

**Note:** you must run the command **“/sbin/sysctl -p**” as root/sudo (or reboot) to apply the changes.

**5.**Please make sure do the network changes in the below configuration file.

**#vim /etc/sysctl.conf**

net.core.somaxconn=4096

net.ipv4.tcp\_fin\_timeout=30

net.ipv4.tcp\_keepalive\_intvl=30

net.ipv4.tcp\_keepalive\_time=120

net.ipv4.tcp\_max\_syn\_backlog=4096

After making the changes please save it and run the command from root user:

# /sbin/sysctl -p

6.Security Enhanced Linux

**#vim /etc/sysconfig/selinux**

SELINUX=disabled

**Step 1:Configure the package management system (yum).**

Create a **/etc/yum.repos.d/mongodb-org-3.6.repo** file using the below repository file.

Open file with vim editor and paste into the file as shown below:

Ex:**#vim /etc/yum.repos.d/mongodb-org-3.6.repo**

[mongodb-org-3.6]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.6/x86\_64/

gpgcheck=1

enabled=1

gpgkey=<https://www.mongodb.org/static/pgp/server-3.6.asc>

**Step2:Install Mongodb packages.**

**#sudo yum install -y mongodb-org** (Install latest)

**Or**

**#sudo yum install -y mongodb-org-3.6.6 mongodb-org-server-3.6.6 mongodb-org-shell-3.6.6 mongodb-org-mongos-3.6.6 mongodb-org-tools-3.6.6** (Install specific release)

After installation please check with below commands:

**#sudo service mongod star**t (To start the mongodb)

**#sudo chkconfig mongod on** (Startup up at boot time)

**#sudo service mongod stop** (Stop the services)

**#sudo service mongod restart** (Restart the services)

**#sudo service mongod status** (To check the status)

**Step3:Next, We will disable SELinux by editing the configuration file with vim.**

#vim /etc/sysconfig/selinux

SELINUX=disabled

Change value **'enforcing' to 'disabled',**then save and reboot the server.

**Step 4:Configure firewalld**

In this step, we already disabled SELinux.For security reasons, we will now enable firewalld on all nodes and open only the ports that are used by MongoDB and SSH.

**#yum -y install firewalld**

Start firewalld and enable it to start at boot time.

**#systemctl start firewalld**

**#systemctl enable firewalld**

Next, open your ssh port and the MongoDB default port 27017.

**#firewall-cmd --permanent --add-port=22/tcp**

**#firewall-cmd --permanent --add-port=27017/tcp**

Reload the firewall to apply the changes

**#firewall-cmd --reload**

**Step5:Configure MongoDB Replica Set**

Edit the MongoDB configuration file to enable authentication and change the security setting then save and quit.

**#sudo vim /etc/mongod.conf**

security:

authorization: enabled

**Step 6:Connect to MongoDB Shell and create a admin user for use in the database:**

**#sudo service mongod star**t

**#mongo**

**>db.createUser({user: "admin", pwd: "admin123", roles:[{role: "root", db: "admin"}]})**

**>quit()**

**Step7: Connect with admin user was created successfully**

#mongo -u admin -p --authenticationDatabase admin

**Step 8:Creating the cluster**

1.First step is to tell mongodb the name of the replica set and stop the services.and then update the replication section of the config file with below command:

#**sudo service mongod stop**

#vim /etc/mongod.conf

replication:

replSetName: rs0

2.Start mongodb services

#**sudo service mongod star**t

**Note:**Repeat the above steps for other nodes.

3.Next step is to create a shared keyfile the servers can use to authenticate the connection between them.

#sudo openssl rand -base64 756 > /etc/ssl/mongodb-internal.key

#sudo chown mongod:mongod /etc/ssl/mongodb-internal.key

#chmod 400 /etc/ssl/mongodb-internal.key

Then copy this file to the other nodes, into **/etc/ssl/mongodb-internal.key**

Update the MongoDB config to use this key file, change the security section as follows:

4**.#vim /etc/mongod.conf**

security:

authorization: enabled

keyFile: /etc/ssl/mongointernal.key

**Step9:** Next create replica set and run the mongodb from command line and register the nodes.

#mongo -u admin -p --authenticationDatabase admin

>rs.initiate(

{

\_id : rs0,

members: [

{ \_id : 0, host : "<node1>:27017" },

{ \_id : 1, host : "<node2>:27017" },

{ \_id : 2, host : "<node3>:27017" }

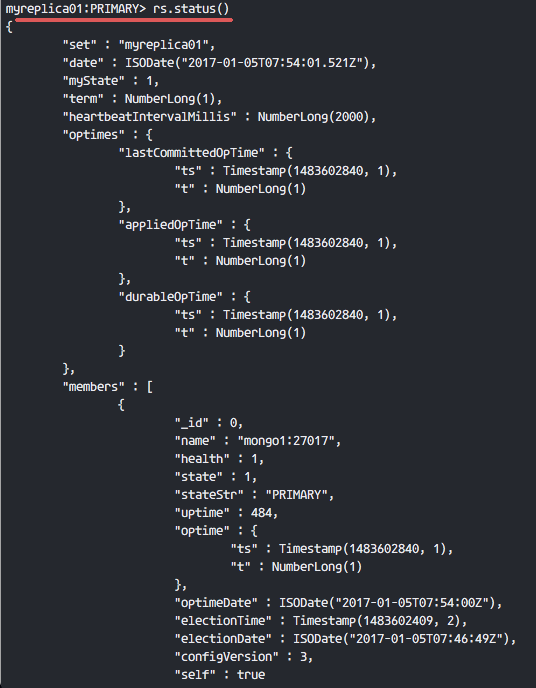
]

}

)

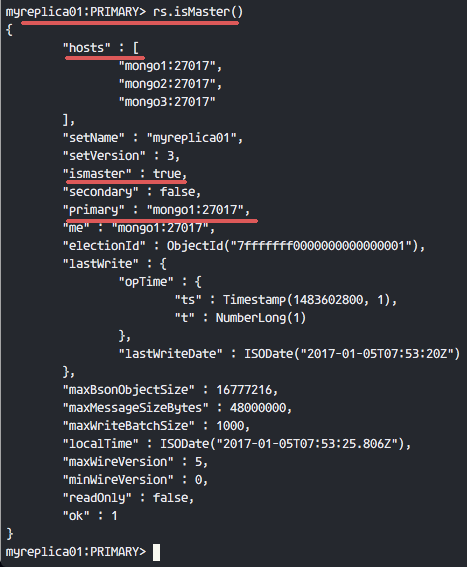
**>rs.status()**

**Output:**

****

**>rs.isMaster()**

**Output:**

****

**Step 10:Test the Replication:**

In this example,taking as **node1** as primary and **node2,node3** as secondary nodes.

1.Login to the 'node1' server and open mongo shell.

Then create a new database '**lemp**' and new '**stack**' collection for the database.

#ssh root@node1

#mongo

Then create a new database '**lemp**' and new '**stack**' collection for the database.

>use lemp

>db.stack.save(

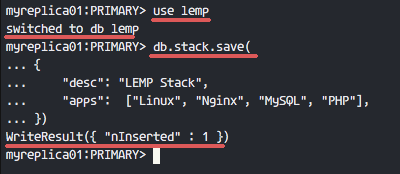
{

"desc": "LEMP Stack",

"apps": ["Linux", "Nginx", "MySQL", "PHP"],

})

**Output:**



2.Next, go to the 'SECONDARY' node 'node2' and open the mongo shell.

#ssh root@node2

#mongo

Enable reading from the 'SECONDARY' node with the query 'rs.slaveOk()', and then check if the 'lemp' database exists on the 'SECONDARY' nodes.

>rs.slaveOk()

>show dbs

>use lemp

>show collections

>db.stack.find()

**Output:**

