

MongoDB Replica Set Deployment



MongoDB副本集群单机部署方案
Yongjie.L@outlook.com


document author: Yongjie Lyu

本文参考官方文档:

Deploy a Replica Set - MongoDB Manual

To deploy a replica set with enabled access control, see Deploy New Replica Set With Keyfile Access Control. If you wish to deploy a replica set from a single MongoDB instance, see Convert a Standalone to a Replica Set. For more information on replica set


<https://www.mongodb.com/docs/v4.0/tutorial/deploy-replica-set/>

 **mongoDB** | FOR GIANT IDEAS

Install MongoDB Community on Red Hat or CentOS using .tgz Tarball - MongoDB Manual

Use this tutorial to manually install MongoDB 4.0 Community Edition on Red Hat Enterprise Linux, CentOS Linux, or Oracle Linux [1] using a downloaded tarball. This tutorial installs MongoDB 4.0 Community Edition. To install a different version of MongoDB Community, use the version drop-down

<https://www.mongodb.com/docs/v4.0/tutorial/install-mongodb-on-red-hat-tarball/>

 **mongoDB** | FOR GIANT IDEAS

MongoDB支持平台版本

Aa Name	4.0社区&企业版	3.6社区&企业版	3.4社区&企业版
<u>CentOS/Oracle Linux 8.0+</u>	4.0.14+	3.6.17+	不支持
<u>CentOS/Oracle Linux 7.0+</u>	✓	✓	✓
<u>CentOS/Oracle Linux 6.2+</u>	✓	✓	✓

本部署文档在CentOS Linux环境下使用**MongoDB Community 4.0**版本进行部署

下载MongoDB数据库

手动安装数据库有两种方式，由于yum模式安装不容易进行自由的自定义安装，且需要机器连接外网，本文档部署使用.tgz格式包下载进行安装

mongodb官方下载包地址：<https://www.mongodb.com/download-center/community/releases>

- RedHat / CentOS 8.0 x64
 - Archive: [mongodb-linux-x86_64-rhel80-4.0.28.tgz](#)
 - Server Package: [mongodb-org-server-4.0.28-1.el8.x86_64.rpm](#)
 - Mongos Package: [mongodb-org-mongos-4.0.28-1.el8.x86_64.rpm](#)
 - Utilities Package: [mongodb-org-tools-4.0.28-1.el8.x86_64.rpm](#)
 - Mongo Shell Package: [mongodb-org-shell-4.0.28-1.el8.x86_64.rpm](#)

安装MongoDB数据库（单节点）

libcurl with openssl网络协议库

安装MongoDB之前需要安装平台依赖包，根据具体使用平台安装，CentOS平台下需要安装libcurl库

官方下载地址：<https://curl.se/download.html>

[curl](#) / [Download](#)

Releases and Downloads

The curl project mostly provides source packages. Other packages are kindly provided by external persons and organizations.

Source Archives

curl 7.83.0, Released on the 27th of April 2022. [Changelog for 7.83.0.](#)

[curl-7.83.0.tar.gz](#) [gpg](#)
[curl-7.83.0.tar.bz2](#) [gpg](#)
[curl-7.83.0.zip](#) [gpg](#)
[curl-7.83.0.tar.xz](#) [gpg](#)

Download Wizard

Need help to select what to download? Use the [curl Download Wizard!](#)

Related:
[Changelog](#)
[Old Releases](#)
[Source code repository](#)
[Daily Snapshots](#)
[GPG Key](#)
[Releaselogs](#)

yum安装

```
yum install curl -y
```

验证是否正确安装，命令行输入 curl -v查看版本命令，系统会反馈如下：

```
libcurl/7.83.0 OpenSSL/1.0.2k-fips zlib/1.3.1
Release-Date: 2022-04-27
Protocols: dict file ftp ftps gopher http https imap imaps pop3 pop3s rtsp smb smbs smtp smtps telnet tftp
Features: AsynchDNS HTTPS-proxy IPv6 Largefile libz NTLM NTLM_WB SSL UnixSockets
```

继续安装MongoDB

使用root账户在执行如下命令

- 解压文件

```
tar -zxvf mongodb-linux-*4.0.27.tgz
```

- 确保文件在系统PATH环境变量下

这里有两种方法，如果将文件安装在已有的PATH环境变量下的话（比如：`/usr/local/bin/`），无需其他配置即可（`/path/to/the/mongodb-directory/bin/*`路径需要修改为实际数据库的安装文件夹）。

```
sudo cp /path/to/the/mongodb-directory/bin/* /usr/local/bin/
```

或者可以通过在PATH环境变量下用ln命令创建同步链接，（下面代码中的`/path/to/the/mongodb-directory/bin/*`路径需要修改为实际数据库的安装文件夹（例如：`/root/mongodb-linux-x86_64-ubuntu2004-5.0.0`））

```
sudo ln -s /path/to/the/mongodb-directory/bin/* /usr/local/bin/
```

- 安装MongoDB Shell

通过官方shell来连接刚刚部署的数据库

MongoDB数据库的前期配置

自定义数据文件夹及权限

- 使用官方自带默认仓库地址

创建MongoDB数据存放和log存放文件夹

```
sudo mkdir -p /var/lib/mongo
```

```
sudo mkdir -p /var/log/mongodb
```

默认情况下，MongoDB使用mongod用户帐户运行，软件包管理器会在安装过程中创建默认目录，所有者和组名为mongod。也可自定义创建，创建完毕要赋予对应用户和目录的权限 `chown -R 用户名:组名 xx/xxx`。

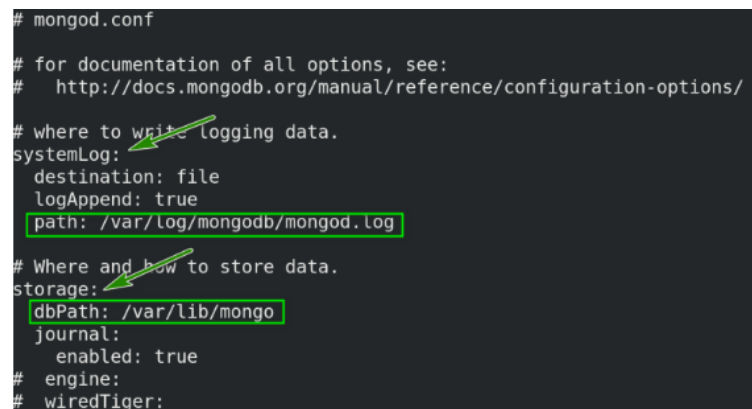
使用如下代码设置mongod的文件所有者（/var/lib/mongo文件的拥有者设定为mongod，群体的使用者为mongodb）

```
sudo chown -R mongod:mongod /var/lib/mongo
```

```
sudo chown -R mongod:mongod /var/log/mongodb
```

- 不使用官方默认仓库地址，需要自行创建好新的文件夹（命令同上，修改文件安装路径）

按照如下措施编辑conf文件（etc/mongod.conf）



```
# mongod.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true
# engine:
#   wiredTiger:
```

1. storage.dbPath 用来确定新的data数据仓库存放位置
2. systemLog.path 用来确定新的Log日志存放位置
3. 确保用户对新的仓库地址有读写权限，directory位置填写数据文件路径

```
sudo chown -R mongod:mongod <directory>
```

特别注意：如果操作用户发生变化，需要重新给新用户添加仓库访问权限！

自定义SELinux策略

如果SELinux处于enforcing模式下，则必须手动为MongoDB自定义SELinux策略。

查看SELinux模式：执行 `sestatus` 命令可查看当前系统模式

```

1 [root@localhost ~]# sestatus
2 SELinux status:                enabled
3 SELinuxfs mount:                /sys/fs/selinux
4 SELinux root directory:         /etc/selinux
5 Loaded policy name:              targeted
6 Current mode:                   enforcing
7 Mode from config file:          enforcing
8 Policy MLS status:              enabled
9 Policy deny_unknown status:     allowed
10 Max kernel policy version:     31

```

手动添加cgroup权限

1. 安装checkpolicy组件

```
sudo yum install checkpolicy
```

2. 手动创建mongodb_cgroup_memory.te自定义策略文件

```

cat > mongodb_cgroup_memory.te <<EOF
module mongodb_cgroup_memory 1.0;

require {
    type cgroup_t;
    type mongod_t;
    class dir search;
    class file { getattr open read };
}

#===== mongod_t =====
allow mongod_t cgroup_t:dir search;
allow mongod_t cgroup_t:file { getattr open read };
EOF

```

3. 配置文件创建完成后，对该文件进行编译加载

完成之后在SELinux在enforcing模式下，MongoDB进程就有权限对文件进行访问了

```

checkmodule -M -m -o mongodb_cgroup_memory.mod mongodb_cgroup_memory.te
semodule_package -o mongodb_cgroup_memory.pp -m mongodb_cgroup_memory.mod
sudo semodule -i mongodb_cgroup_memory.pp

```

给FTDC手动添加netstat权限

由于SELinux策略，目前MongoDB进程是不能够访问读写/proc/net/netstat文件的，需要调整策略

本项工作也需要checkpolicy组件才能够继续进行，上文中应该已经安装该组件

1. 手动创建mongodb_proc_net.te文件

```

cat > mongodb_proc_net.te <<EOF
module mongodb_proc_net 1.0;

require {
    type proc_net_t;
    type mongod_t;
    class file { open read };
}

#===== mongod_t =====
allow mongod_t proc_net_t:file { open read };
EOF

```

2. 编译加载该文件

```
checkmodule -M -m -o mongodb_proc_net.mod mongodb_proc_net.te
semodule_package -o mongodb_proc_net.pp -m mongodb_proc_net.mod
sudo semodule -i mongodb_proc_net.pp
```

若MongoDB使用自定文件路径的情况

1. 需要更新SELinux策略使Mongod服务有权限去使用新的文件夹

```
sudo semanage fcontext -a -t <type> </some/MongoDB/directory.*>
```

！注意：需要添加代码最后的“.*”

代码中的types更换为以下选项，/some/MongoDB/directory需要变为实际目录

- `mongod_var_lib_t` 数据仓库
- `mongod_log_t` log日志仓库
- `mongod_var_run_t` pid文件仓库

2. 对新的文件夹需要手动更新SELinux策略

```
sudo chcon -Rv -u system_u -t <type> </some/MongoDB/directory>
```

代码中的types更换为以下选项（同上）

- `mongod_var_lib_t` 数据仓库
- `mongod_log_t` log日志仓库
- `mongod_var_run_t` pid文件仓库

3. 将更新后的SELinux策略应用到文件夹

```
sudo restorecon -R -v </some/MongoDB/directory>
```

- 使用默认的MongoDB数据仓库地址（/mongodb/data）

```
sudo semanage fcontext -a -t mongod_var_lib_t '/mongodb/data.*'
sudo chcon -Rv -u system_u -t mongod_var_lib_t '/mongodb/data'
sudo restorecon -R -v '/mongodb/data'
```

- 不使用默认的MongoDB数据仓库地址

```
sudo semanage fcontext -a -t mongod_log_t '/mongodb/log.*'
sudo chcon -Rv -u system_u -t mongod_log_t '/mongodb/log'
sudo restorecon -R -v '/mongodb/log'
```

自定义MongoDB端口

<portnumber>填写端口地址

```
sudo semanage port -a -t mongod_port_t -p tcp <portnumber>
```

MongoDB数据库运行

1. 运行MongoDB

系统

```
mongod --dbpath /var/lib/mongo --logpath /var/log/mongodb/mongod.log --fork
```

dbpath：mongodb存放数据的文件路径

logpath：MongoDB存放Log日志数据的路径

2. 确认MongoDB是否已正常启动

通过查看位于/var/log/mongodb/mongod.log（默认路径）路径下的日志文件，查看是否有如下代码生成输出

```
[initandlisten] waiting for connections on port 27017
```

注意：日志文件中可能会记录有部分非重要错误提醒（non-critical warnings）

可使用status命令查看数据库运行状态 `systemctl status mongod.service`

3. 通过shell连接数据库

在主机上可以直接运行 `mongo` 命令来连接本地的mongod（默认端口为27017时）

其他配置

绑定本地监听端口bindip

本文档安装的4.0版本数据库在启动MongoDB时默认将bindip设置为127.0.0.1，在这种情况下mongod只能在安装机器上的客户端进行访问，在不修改默认值的情况下远程客户端无法进行访问且无法初始化成副本集架构。

修改/etc/mongod.conf文件：在conf配置文件中可以调整bind_ip参数，如果想要同时使用localhost和Server ip登录的话，可以将参数赋值调整为 `bind_ip=localhost, 172.XXX.XXX.XXX`

```
systemLog:
  destination: file
  path: "/var/log/mongodb/mongod.log"
  logAppend: true
storage:
  journal:
    enabled: true
processManagement:
  fork: true
net:
  bindIp: 127.0.0.1
  port: 27017
setParameter:
  enableLocalhostAuthBypass: false
...
```

配置用户角色

修改/etc/mongod.conf文件：使用客户端连接MongoDB之后使用如下命令添加角色和用户，以下为示例，具体db.createUser使用方法请参照官方文档

```
use admin;
db.createUser(
{
  user:"admin",
  pwd:"xxxxxxx",
  roles:[
    {"role":"userAdminAnyDatabase","db":"admin"},
    {"role":"readWriteAnyDatabase","db":"admin"}
  ]
})
```

副本集架构的部署



MongoDB副本集群部署方案(三成员)

介绍

以下教程演示如何将三个已有的mongod实例转换为有三个成员的副本集架构集群，在开始部署副本集架构前应该在每个系统中安装好mongoDB。在实际生产环境中，不同副本集成员应该尽量部署在不同的机器上或不同的主机服务器上。

注意：本文档是在访问控制(Access Control)关闭的情况下进行操作！

注意：生产环境中应该尽可能将副本集的成员都分不在不同的机器上

本地监听端口bindip

- bindip

从MongoDB3.6版本之后，mongo会默认设置localhost为bindip。如果有net.ipv6配置文件或是命令行加上了“—ipv6”配置的话，也会额外将本地ipv6localhost设置为bindip

可以使用net.bindIp配置文件或命令行参数来定义bindip列表，以下采用修改配置文件的方式：

```
mongod --bind_ip localhost,Example-HostName
```

mongod同时绑定了localhost和名称Example-HostName

副本集配置

将每个实例配置成副本集

需要将replication.replSetName设置为副本集的名称，注意给每个副本集成员设置不同的名称
将net.bindIp设置为hostname/ip的选项

如以下文档示例所示：

副本集成员	hostName
Member 0	mongodb0.example.net
Member 1	mongodb1.example.net
Member 2	mongodb2.example.net

可以通过命令行代码或者配置文件修改的方式来设置了副本成员名称和bindIp，文档使用修改配置文件的方式：

```
replication:
  replSetName: "rs0"
```

```
net:
  bindIp: localhost,<hostname(s)|ip address(es)>
```

注意在配置文件中的<主机名称hostname | ip地址>的组合设置中，需要确保远程客户端以及副本集成员之间能够连接。

命令行使用配置文件参数来启动mongo实例：

```
mongod --config <配置文件路径>
```

初始化mongod实例

在其中一个副本集成员上运行 `rs.initiate()` 可以让MongoDB使用默认的副本集配置文件来初始化一个副本集集群

配置文件：

```
rs.initiate( {
  _id : "rs0",
  members: [
    { _id: 0, host: "mongodb0.example.net:27017" },
    { _id: 1, host: "mongodb1.example.net:27017" },
    { _id: 2, host: "mongodb2.example.net:27017" }
  ]
})
```

查看副本成员的配置

`rs.conf()` 可以用来展示副本集群的配置信息，以下为三节点副本集群的配置参考文件：

```
{
  "_id" : "rs0",
  "version" : 1,
  "protocolVersion" : NumberLong(1),
  "members" : [
    {
      "_id" : 0,
      "host" : "mongodb0.example.net:27017",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 1,
      "tags" : {

      },
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    },
    {
      "_id" : 1,
      "host" : "mongodb1.example.net:27017",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 1,
      "tags" : {

      },
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    },
    {
      "_id" : 2,
      "host" : "mongodb2.example.net:27017",
      "arbiterOnly" : false,
      "buildIndexes" : true,
```

- `_id`：副本集集群的名称，注意需要和命令行`replSet`所指定的mongod保持一致
- `version`：之前配置文件的修改次数数的一个递增数字
- `protocolVersion`：默认为1，具体规定mongoDB读写规则，从4.0版本之后MongoDB仅支持使用规则“1”
- `members | _id`：仅能设置为0-255，设置之后不能修改
- `members | host`：主机名称及端口号
- `members | arbiterOnly`：该项为选填项，默认为False，表示该副本集成员是否为仲裁员
- `members | buildIndexes`：Mongod是否在该成员上建立索引，该选项仅能在刚加入新成员的时候设置
- `members | hidden`：该项为true的时候该实例隐藏，无法进行读操作，但该成员可以参与选举
- `members | priority`：该项表明该成员成为Primary节点的可能性，越高的数值表示可能性


```

        "hidden" : false,
        "priority" : 1,
        "tags" : {

        },
        "slaveDelay" : NumberLong(0),
        "votes" : 1
    }
],
"settings" : {
    "chainingAllowed" : true,
    "heartbeatIntervalMillis" : 2000,
    "heartbeatTimeoutSecs" : 10,
    "electionTimeoutMillis" : 10000,
    "catchUpTimeoutMillis" : -1,
    "getLastErrorModes" : {

    },
    "getLastErrorDefaults" : {
        "w" : 1,
        "wtimeout" : 0
    },
    "replicaSetId" : "bu

```

越高。如果数字为0的时候表示该成员不能参与选举成为Primary节点。

0-1000的数字表示primary和secondary节点，仲裁员为0

- members | slaveDelay：在主节点之后的成员应该延迟的时间，单位为秒，设置该项来创建延迟副本集成员
- members | votes：在副本集成员进行选举的时候，该成员所持有的票数，该选项为1或0，仲裁员永远为1