

实验九：使用 MongoDB 建设数据库集群

一、实验目的

- 1、了解 MongoDB 数据库；
- 2、掌握 MongoDB 的安装与基本配置；
- 3、掌握 MongoDB 数据库集群的实现；
- 4、掌握使用 MongoDB Compass 管理 MongoDB 数据库集群。

二、实验学时

2 学时

三、实验类型

综合性



四、实验需求

1、硬件

每人配备计算机 1 台。

2、软件

Windows 操作系统，安装 Oracle VM VirtualBox 软件，安装 MobaXterm 软件。
安装 MongoDB Compass 软件。

3、网络

本地主机与虚拟机能够访问互联网，不使用 DHCP 服务。

4、工具

无。

五、实验任务

- 1、完成 MongoDB 的安装；
- 2、完成使用 MongoDB 实现副本集；
- 3、完成使用 MongoDB Compass 管理 MongoDB 数据库集群，并进行副本集测试。

六、实验环境

虚拟机配置	操作系统配置
虚拟机名称: VM-Lab-09-Task-01-172.20.1.20 内存: 1GB CPU: 1 颗, 1 核心	主机名: Lab-09-Task-01 IP 地址: 172.20.1.20 子网掩码: 255.255.255.0 网关: 172.20.1.1

虚拟磁盘：20GB 网卡：1 块，桥接	DNS：8.8.8.8
虚拟机名称：VM-Lab-09-Task-02-172.20.1.21 内存：1GB CPU：1 颗，1 核心 虚拟磁盘：20GB 网卡：1 块，桥接	主机名：Lab-09-Task-02 IP 地址：172.20.1.21 子网掩码：255.255.255.0 网关：172.20.1.1 DNS：8.8.8.8
虚拟机名称：VM-Lab-09-Task-03-172.20.1.22 内存：1GB CPU：1 颗，1 核心 虚拟磁盘：20GB 网卡：1 块，桥接	主机名：Lab-09-Task-03 IP 地址：172.20.1.22 子网掩码：255.255.255.0 网关：172.20.1.1 DNS：8.8.8.8

3、本实验拓扑图，如图 9-1 所示。

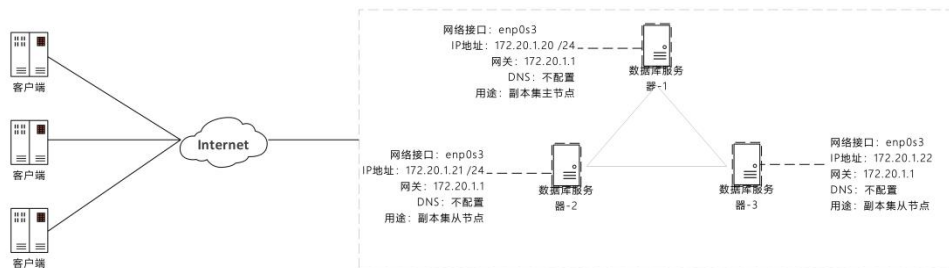


图 9-1 实验拓扑图

4、本实验操作演示视频。

本实验为视频集的第 6 集：<https://www.bilibili.com/video/BV1Vh4y1T7EP?p=6>

七、实验内容步骤

1、完成 MongoDB 的安装

1.1 在主机 Lab-09-Task-01 上操作

(1) 查看防火墙 Firewalld 服务状态（CentOS 操作系统默认安装 Firewalld 防火墙，并创建 firewalld 服务，该服务已开启且已配置为开机自启动）。

(2) 使用 firewall-cmd 命令添加本地客户端允许远程连接 MongoDB 数据库，以及其余两台主机允许联通该主机的 MongoDB 数据库，并重新载入防火墙配置使其生效。

(3) 使用配置 yum 源的方式完成 MongoDB 的安装，安装前需先更新软件包缓存。

(4) 启动 mongod 服务，设置 mongod 服务为开机自启动并查看 mongod 服务运行状态。

查看防火墙 Firewalld 服务状态

```
[root@Lab-09-Task-01 ~]# systemctl status firewalld
```

```
# 添加本地客户端允许远程连接 MongoDB 数据库防火墙规则
[root@Lab-09-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ip
v4 source address=172.20.1.134 port port=27017 protocol=tcp accept'
# 添加其余两台主机允许联通当前主机的 MongoDB 数据库防火墙规则
[root@Lab-09-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ip
v4 source address=172.20.1.21 port port=27017 protocol=tcp accept'
[root@Lab-09-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ip
v4 source address=172.20.1.22 port port=27017 protocol=tcp accept'

# 重新载入防火墙配置使其生效
[root@Lab-09-Task-01 ~]# firewall-cmd --reload

# 使用 vi 命令编辑/etc/yum.repos.d/mongodb.repo 文件
[root@Lab-09-Task-01 ~]# vi /etc/yum.repos.d/mongodb.repo
# -----/etc/yum.repos.d/mongodb.repo 文件-----
[mongodb-org-4.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/8/mongodb-org/4.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-4.4.asc
# -----/etc/yum.repos.d/mongodb.repo 文件-----

# 更新软件包缓存
[root@Lab-09-Task-01 ~]# dnf update

# 使用 dnf 命令安装 MongoDB
[root@Lab-09-Task-01 ~]# dnf install -y mongodb-org

# 启动 mongod 服务
[root@Lab-09-Task-01 ~]# systemctl start mongod
# 设置 mongod 服务为开机自启动
[root@Lab-09-Task-01 ~]# systemctl enable mongod
# 查看 mongod 服务运行状态
[root@Lab-09-Task-01 ~]# systemctl status mongod
```

提醒:

- 1、如果 FirewallD 防火墙未启动, 请使用命令 `systemctl start firewalld` 启动防火墙; 如果 FirewallD 防火墙未设置为开机自启动, 请使用命令 `systemctl enable firewalld` 设置为开机启动。
 - 2、上述命令中“172.20.1.134”为本机 IP 地址, 请根据实际情况替换为 IP 地址。
 - 3、当前主机 IP 地址为 172.20.1.20, 那么需要添加其余主机 (172.20.1.21、172.20.1.22) 允许联通当前主机的 MongoDB 数据库防火墙规则; 同理, 当前主机 IP 地址为 172.20.1.21, 那么需要添加其余主机 (172.20.1.20、172.20.1.22) 允许联通当前主机的 MongoDB 数据库防火墙规则; 当前主机 IP 地址为 172.20.1.22, 那么需要添加其余主机 (172.20.1.20、172.20.1.21) 允许联通当前主机的 MongoDB 数据库防火墙规则。上述命令中仅为主机 172.20.1.20 的防火墙配置规则, 当配置其他主机时, 请修改配置防火墙规则中的 IP 地址为其余两台主机, 保证 3 台主机的 MongoDB 数据库可以相互联通。
-

1.2 在主机 Lab-09-Task-02 上操作

主机 Lab-09-Task-02 安装 MongoDB 的方法和过程，请参考步骤 1.1 完成。

1.3 在主机 Lab-09-Task-03 上操作

主机 Lab-09-Task-03 安装 MongoDB 的方法和过程，请参考步骤 1.1 完成。

2、配置副本集

2.1 在主机 Lab-09-Task-01 上操作

具体过程如下。

- (1) 生成 MongoDB 的副本集密钥。
- (2) 设置上述创建的密钥文件 (keyfile.key) 的读写权限为 400，所有者和所属组均为 mongod。
- (3) 使用 scp 命令将主机 Lab-09-Task-01 上的副本集密钥文件分发到主机 Lab-09-Task-02，并按照操作提示完成文件的传输。
- (4) 使用 scp 命令将主机 Lab-09-Task-01 上的副本集密钥文件分发到主机 Lab-09-Task-03，并按照操作提示完成文件的传输。
- (5) 修改/etc/mongod.conf 文件配置 MongoDB 支持副本集，修改完成后重启 mongod 服务，确保配置生效。

```
# 生成 MongoDB 的副本集密钥
[root@Lab-09-Task-01 ~]# openssl rand -base64 756 > /var/lib/mongo/keyfile.key

# 设置副本集密钥文件 (keyfile.key) 的读写权限为 400，所有者和所属组均为 mongod
[root@Lab-09-Task-01 ~]# chmod 400 /var/lib/mongo/keyfile.key
[root@Lab-09-Task-01 ~]# chown mongod:mongod /var/lib/mongo/keyfile.key

# 将主机 Lab-09-Task-01 上的副本集密钥文件分发到主机 Lab-09-Task-02
[root@Lab-09-Task-01 ~]# scp /var/lib/mongo/keyfile.key root@172.20.1.21:/var/lib/
mongo/
# 将主机 Lab-09-Task-01 上的副本集密钥文件分发到主机 Lab-09-Task-03
[root@Lab-09-Task-01 ~]# scp /var/lib/mongo/keyfile.key root@172.20.1.22:/var/lib/
mongo/

# 使用 vi 命令编辑/etc/mongod.conf 数据库配置文件
[root@Lab-09-Task-01 ~]# vi /etc/mongod.conf
# -----/etc/mongod.conf 文件-----
net:
  port: 27017
  bindIp: 0.0.0.0
security:
  keyFile: /var/lib/mongo/keyfile.key
replication:
  replSetName: "mongodb-cluster"
# -----/etc/mongod.conf 文件-----
# 重启 mongod 服务
[root@Lab-09-Task-01 ~]# systemctl restart mongod
```

2.2 在主机 Lab-09-Task-02 上操作

(1) 将从主机 Lab-09-Task-01 传输过来的副本集密钥文件 (keyfile.key) 的读写权限设置为 400, 所有者和所属组均设置为 mongod。

(2) 修改/etc/mongod.conf 文件配置 MongoDB 支持副本集, 修改完成后重启 mongod 服务, 确保配置生效。

```
# 设置副本集密钥文件 (keyfile.key) 的读写权限为 400, 所有者和所属组均为 mongod
[root@Lab-09-Task-02 ~]# chmod 400 /var/lib/mongo/keyfile.key
[root@Lab-09-Task-02 ~]# chown mongod:mongod /var/lib/mongo/keyfile.key

# 使用 vi 命令编辑/etc/mongod.conf 数据库配置文件
[root@Lab-09-Task-02 ~]# vi /etc/mongod.conf
# -----/etc/mongod.conf 文件-----
net:
  port: 27017
  bindIp: 0.0.0.0
security:
  keyFile: /var/lib/mongo/keyfile.key
replication:
  replSetName: "mongodb-cluster"
# -----/etc/mongod.conf 文件-----
# 重启 mongod 服务
[root@Lab-09-Task-02 ~]# systemctl restart mongod
```

2.3 在主机 Lab-09-Task-03 上操作

主机 Lab-09-Task-03 配置副本集的方法和过程, 请参考步骤 2.2 完成。

3、初始化副本集

在主机 Lab-09-Task-01 上进行操作, 进行副本集的初始化, 具体步骤如下。

(1) 使用 mongo 命令连接 MongoDB 客户端, 初始化副本集, 并退出当前连接。

(2) 使用 mongo 命令连接 MongoDB 客户端, 查看副本集状态, 并退出当前连接。

(3) 使用 mongo 命令连接 MongoDB 客户端, 为副本集创建用户, 并退出当前连接。

```
# 使用 mongo 命令连接 MongoDB 客户端
[root@Lab-09-Task-01 ~]# mongo
# 初始化副本集
> rs.initiate( {
  _id : "mongodb-cluster",
  members: [
    { _id:0, priority:2, host:"172.20.1.20:27017"},
    { _id:1, host:"172.20.1.21:27017"},
    { _id:2, host:"172.20.1.22:27017"}
  ]
})
# 退出当前连接
```

```
mongodb-cluster:PRIMARY> quit()

# 使用 mongo 命令连接 MongoDB 客户端
[root@Lab-09-Task-01 ~]# mongo
#查看副本集状态
mongodb-cluster:PRIMARY> rs.status()
# 退出当前连接
mongodb-cluster:PRIMARY> quit()

# 使用 mongo 命令连接 MongoDB 客户端
[root@Lab-09-Task-01 ~]# mongo
# 为副本集创建用户
mongodb-cluster:PRIMARY> use admin
mongodb-cluster:PRIMARY> db.createUser(
  {
    user: "mongodblab1",
    pwd: "mongodblab1#PWD",
    roles: [{ role: "userAdminAnyDatabase",db: "admin"} ,"readWriteAnyDatabase"]
  }
)
mongodb-cluster:PRIMARY> db.getSiblingDB("admin").auth("mongodblab1",passwordPrompt())
Enter password:这里输入 mongodblab1 用户密码(mongodblab1#PWD)

mongodb-cluster:PRIMARY> db.getSiblingDB("admin").createUser(
  {
    user: "mongodblab2",
    pwd: "mongodblab2#PWD",
    roles: [{role: "clusterAdmin",db: "admin"},"readWriteAnyDatabase"]
  }
)
# 退出当前连接
mongodb-cluster:PRIMARY> quit()
```

提醒:

通过上述 `rs.status()` 命令查看副本集状态, 当前主机 Lab-09-Task-01 为主节点, 主机 Lab-09-Task-02、主机 Lab-09-Task-03 为副本节点。

4、使用 MongoDB Compass 管理 MongoDB 数据库集群

(1) 从 MongoDB Compass 的官方网站 (<https://www.mongodb.com>) 获取可执行程序, 并进行安装。

(2) 运行 MongoDB Compass, 依据向导添加 3 台 MongoDB 服务器, 并进行连接。

点击 “New connection” 按钮, 在右侧的 URI 输入框中通过标准连接字符串格式连接 MongoDB 数据库, 如图 9-2 所示。

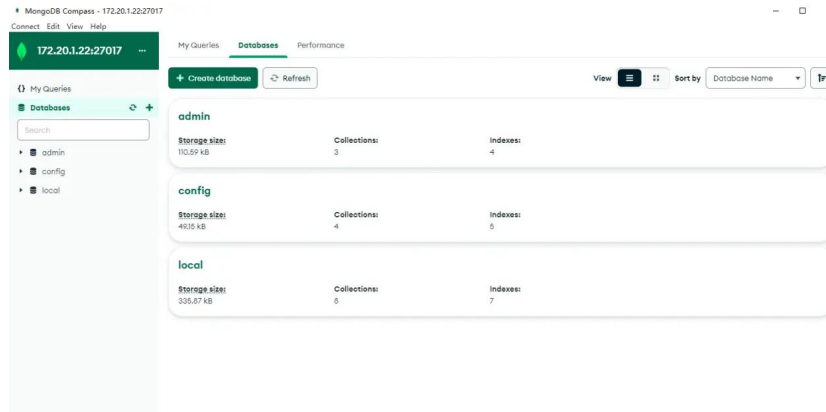


图 9-2 MongoDB Compass 的主界面

连接格式如下。

URI 连接格式:

mongodb://数据库用户名:数据库用户密码@IP 地址:端口号

以连接主机 Lab-09-Task-01 中的 MongoDB 数据库为例:

mongodb://mongodblab2:mongodblab2%23PWD@172.20.1.20:27017

首次连接成功后, 点击左上角 “connect” - “New Window” 新建连接窗口, 并依次连接剩余两个 MongoDB 数据库, 连接格式如下:

连接主机 Lab-09-Task-02 中的 MongoDB 数据库

mongodb://mongodblab2:mongodblab2%23PWD@172.20.1.21:27017

连接主机 Lab-09-Task-03 中的 MongoDB 数据库:

mongodb://mongodblab2:mongodblab2%23PWD@172.20.1.22:27017

(3) 连接到数据库服务器后, 可点击 “Databases” 查看 MongoDB 数据库列表, 如图 9-3 所示。

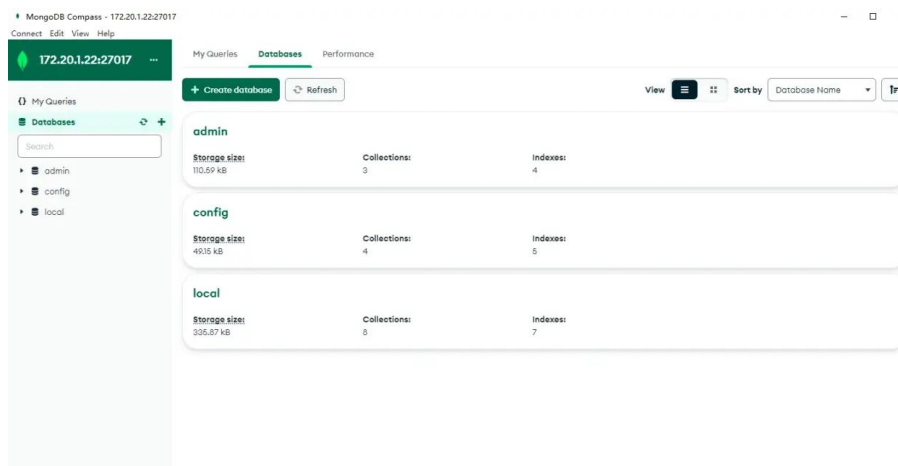


图 9-3 MongoDB Compass 的主界面

5、测试 MongoDB 集群的可用性

场景 1：主节点增加数据，从节点同步增加

(1) 使用 MongoDB Compass 通过用户“mongodblab2”连接到副本集主节点 Lab-09-Task-01，创建数据库、集合，并添加数据。

(2) 使用 MongoDB Compass 通过用户“mongodblab2”连接到副本集从节点-1 (Lab-09-Task-02)，查看在主节点上创建的数据库、集合以及添加的数据，是否存在。

(3) 使用 MongoDB Compass 通过用户“mongodblab2”连接到副本集从节点-2 (Lab-09-Task-03)，查看在主节点上创建的数据库、集合以及添加的数据，是否存在。

场景 2：主节点删除数据，从节点同步删除

(1) 使用 MongoDB Compass 通过用户“mongodblab2”连接到副本集主节点 Lab-09-Task-01，删除数据。

(2) 使用 MongoDB Compass 通过用户“mongodblab2”连接到副本集从节点-1 (Lab-09-Task-02)，查看在主节点上删除的数据，是否已不存在。

(3) 使用 MongoDB Compass 通过用户“mongodblab2”连接到副本集从节点-2 (Lab-09-Task-03)，查看在主节点上删除的数据，是否已不存在。

场景 3：主节点宕机，业务不受影响

(1) 在主节点 Lab-09-Task-01 上执行 `systemctl stop mongod` 命令，关闭 mongod 服务，以模拟主节点宕机故障。

(2) 在从节点 Lab-09-Task-02 上使用 `mongo -umongodblab2 -pmongodblab2#PWD` 命令连接 MongoDB 客户端，查看当前主节点所在的主机。

```
# 停止 mongod 服务
[root@Lab-09-Task-01 ~]# systemctl stop mongod

# 通过用户 mongodblab2 连接 MongoDB 客户端
[root@Lab-09-Task-02 ~]# mongo -umongodblab2 -pmongodblab2#PWD
# 查看当前主节点所在的主机
mongodb-cluster:PRIMARY> rs.status()
```

场景 4：原主节点恢复正常，业务不受影响

(1) 在原主节点 Lab-09-Task-01 上执行 `systemctl start mongod` 命令，启动 mongod 服务，以模拟原主节点恢复正常，连接 MongoDB 客户端并查看副本集状态。

(2) 使用 MongoDB Compass 通过用户“mongodblab2”连接到副本集原主节点 Lab-09-Task-01，查看宕机期间未同步的数据是否已同步。

```
# 启动 mongod 服务
[root@Lab-09-Task-01 ~]# systemctl start mongod

[root@Lab-09-Task-01 ~]# mongo -umongodblab2 -pmongodblab2#PWD
# 查看副本集状态
mongodb-cluster:SECONDARY> rs.status()
```
