

Linux服务器构建与运维管理

第06章：使用MongoDB实现数据库服务

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

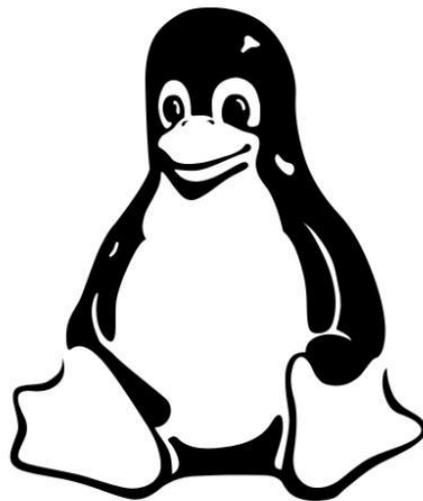
<http://linux.xg.hactcm.edu.cn>
<http://www.51xueweb.cn>

河南中医药大学信息管理与信息系统教研室
信息技术学院网络与信息系统科研工作室
河南中医药大学医疗健康信息工程技术研究所

2022.9

提纲

- 非关系型数据库
- mongoDB简介
 - mongoDB、特性、副本集
- mongoDB数据库服务
 - 实现mongoDB数据库服务
 - mongoDB服务进程
 - 使用mongoDB Compass管理MongoDB数据库服务
- 高可用
 - 实现mongoDB高可用
 - 监控mongoDB



1.非关系型数据库

1.1 NoSQL

- ❑ 非关系型数据库简称NoSQL，最初是为了满足互联网的业务需求而诞生的。互联网数据规模庞大，数据结构动态化，关系型数据库在处理此类问题时不仅十分麻烦，而且性能也达不到要求。
- ❑ NoSQL数据库是非关系型数据存储的广义定义，它不同于符合ACID理论的关系型数据库，数据存储不需要固定的表结构，通常也不存在连接操作。
- ❑ NoSQL数据库不使用传统的关系数据库模型，而是使用如键值存储数据库、列存储数据库、文档型数据库、图形数据库等方式存储数据模型。
- ❑ NoSQL在抛弃了关系型数据库的强制一致性和事务等特性后，可满足业务需求。



ACID，是指在数据库管理系统（DBMS）中事务所具有的四个特性：

- **原子性（Atomicity）**
- **一致性（Consistency）**
- **隔离性（Isolation，又称独立性）**
- **持久性（Durability）**

原子性：组成事务的一组操作，要么全部成功，要么全部失败，不会在中间的某个环节结束。如果在事务的执行过程中，某个操作失败了，数据库会回滚到事务开始前的状态，就像事务从来没有执行过。

一致性：事务执行前后，数据库的完整性没有被破坏，事务执行的前后都是合法的数据状态。

隔离性：数据库允许多个事务并发地对数据进行读写。多个事务并发执行会造成脏读，不可重复读，幻读，而隔离性可以防止多个事务交叉执行导致的数据不一致问题。事务的隔离级别有读未提交，读已提交，可重复读，串行化。

持久性：事务提交后，对数据的修改是持久的，不会因为外部原因丢失。



1.非关系型数据库

1.1 NoSQL

- ❑ 关系型数据库支持事务的ACID特性，保证在事务过程中数据的正确性。
- ❑ 从事务的四个特性可以看出，关系型数据库要求强一致性，但是这一点在非关系型数据库中是重点弱化的机制。
- ❑ 这是因为数据库保持强一致性时，很难保证系统具有横向扩展和可用性的优势，因此针对分布式数据存储管理只提供了弱一致性的保障。



1.非关系型数据库

1.1 NoSQL

- 非关系型数据库是相对于关系型数据库来讲的，不遵循二维数据模型。不同厂商针对应用不同，其非关系型数据库的数据模型也不同。
- 非关系型数据库具备的通用特点如下。
 - 高性能
 - 分布式
 - 易扩展
 - 不支持事务



1.非关系型数据库

1.1 NoSQL

- 非关系型数据库是相对于关系型数据库来讲的，不遵循二维数据模型。不同厂商针对应用不同，其非关系型数据库的数据模型也不同。
- 非关系型数据库具备的**适用场景**如下。
 - 数据模型比较简单
 - 需要灵活性更强的IT系统
 - 对数据库性能要求较高
 - 不需要高度的数据一致性



1.非关系型数据库

1.2 NoSQL的分类与特性

- 与关系型数据库不同，非关系型数据库并没有一个统一的架构，两种非关系型数据库之间的差异程度远远超过两种关系型数据库之间的差异。
 - 非关系型数据库通常具有较强应用场景适应性，不同应用场景下应选用不同产品。
 - 常见的非关系型数据库包括键值数据库、列族数据库、文档数据库和图形数据库。

键值
数据库

列组
数据库

文档
数据库

图性
数据库



非关系型数据库分类和特性

分类	相关产品	应用场景	数据模型	优点	缺点
键值数据库	Redis、Memcached	内容缓存、频繁读写	<key,value>键值对，通过散列表实现	大量操作时性能高	数据无结构化
列族数据库	HBase、Cassandra	分布式数据存储与管理	以列族式存储，将同一列数据存储在一起	查找速度快，复杂性低	功能局限，不支持事务的强一致性
文档数据库	MongoDB、Elasticsearch	Web应用、面向文档或半结构化的数据	<key,value>，value是JSON结构的文档	数据结构灵活	缺乏统一查询语法
图形数据库	Neo4j、AllegroGraph	推荐系统、构建关系图谱	图结构	支持复杂的图形算法	复杂性高，只能支持一定的数据规模

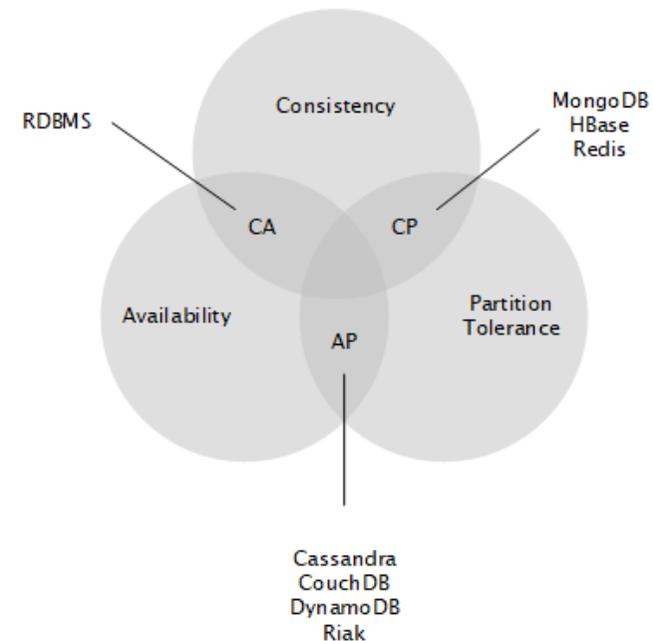


1.非关系型数据库

1.3 CAP

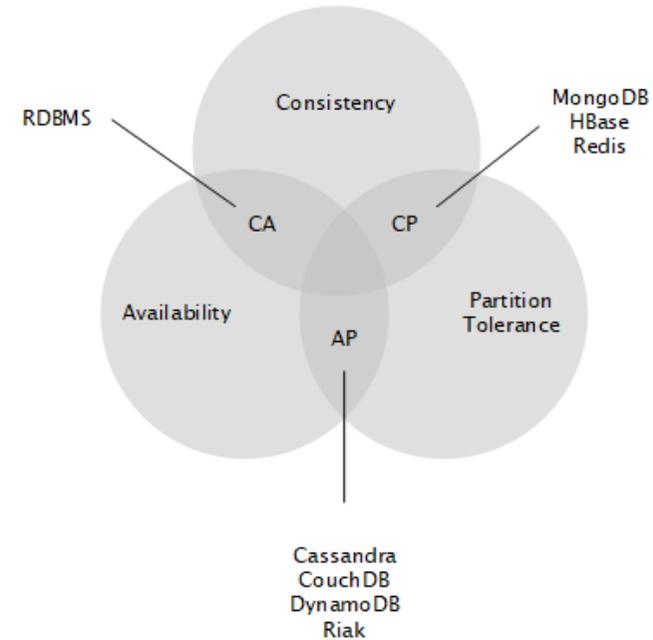
- CAP理论是由Eric Brewer在2001年提出的，他指出对于一个分布式计算系统来说，不可能同时满足以下三点。
 - 一致性 (Consistency)
 - 一致性是指更新操作成功后，所有节点在同一时间的数据完全一致。
 - 可用性 (Availability)
 - 可用性是指用户访问数据时，系统是否能在正常响应时间返回结果。
 - 分区容错性 (Partition Tolerance)
 - 分区容错性是指分布式系统在遇到某节点或网络分区故障的时候，仍然能够对外提供满足一致性和可用性的服务。





选择	特征	实例
一致性+可用性	2PC缓存验证	单机数据库、集群数据库、LDAP、xFS
一致性+分区容忍性	乐观锁	分布式系统、分布式锁、大部分的协议
可用性+分区容忍性	冲突解决, 乐观化	Coda (分布式档案系统)、网络缓存、DNS





选择	例子	对应功能分类
一致性+可用性	MySQL	关系型数据库
	Vertica	列存储数据库
一致性+分区容忍性	BigTable	列存储数据库
	HBase	列存储数据库
	MongoDB	文档型数据库
可用性+分区容忍性	Dynamo	键值存储数据库
	Cassandra	列存储数据库



1.非关系型数据库

1.5 BASE

- BASE是对CAP理论中一致性（C）和可用性（A）进行权衡的结果，其核心思想是无法做到强一致性，但每个应用都可以根据自身的特点，采用适当方式达到最终一致性。
- 一般来说，非关系型数据库都支持BASE原理。
 - 基本可用（Basically Available）：基本可用指分布式系统在出现故障时，系统允许损失部分可用性，即保证核心功能或者当前最重要功能可用。
 - 软状态（Soft-state）：软状态允许数据存在中间状态，但不会影响系统的整体可用性，即允许不同节点的副本之间存在暂时的不一致情况。
 - 最终一致性（Eventually Consistent）：最终一致性要求系统中数据副本最终能够一致，而不需要实时保证数据副本一致。最终一致性是BASE原理的核心，也是非关系型数据库的主要特点，通过弱化一致性，提高系统的可伸缩性、可靠性和可用性。



关系型与非关系型数据库分类和特性对比

数据库类型	特征	优点	缺点
关系型数据库	<ol style="list-style-type: none"> 1、关系型数据库的最大特点就是事务的一致性 2、简单来说，关系模型指的就是二维表格模型 	<ol style="list-style-type: none"> 1、容易理解：二维表结构是非常容易理解 2、使用方便：支持SQL，可用于复杂查询 3、易于维护：丰富的完整性大大降低了数据冗余和数据不一致的概率 	<ol style="list-style-type: none"> 1、为了维护一致性所付出的巨大代价就是其读写性能比较差 2、固定的表结构 3、高并发读写需求 4、海量数据的高效率读写
非关系型数据库	<ol style="list-style-type: none"> 1、使用键值对存储数据 2、分布式 3、非关系型数据库严格上不是一种数据库，应该是一种数据结构化存储方法的集合 	<ol style="list-style-type: none"> 1、无需经过sql层的解析，读写性能很高 2、基于键值对，数据没有耦合性，易扩展 	<ol style="list-style-type: none"> 1、不提供sql支持，学习和使用成本较高 2、无事务处理，附加功能bi和报表等支持也不好

关系型与非关系型数据库各有特点，对于数据库的选型应与自业务结合，充分考量。



2.mongoDB简介

2.1 mongoDB

- mongoDB是全球应用最为广泛的非关系型数据库之一，具备开源、基于文档、功能强大、应用简单等特点。
- mongoDB是一个表结构自由、开源、可扩展、面向文档的数据库，旨在为Web应用程序提供高性能、高可用且易扩展的数据存储解决方案。
- mongoDB支持多文档事务、连接查询，是较为接近关系型数据库的非关系型数据库。

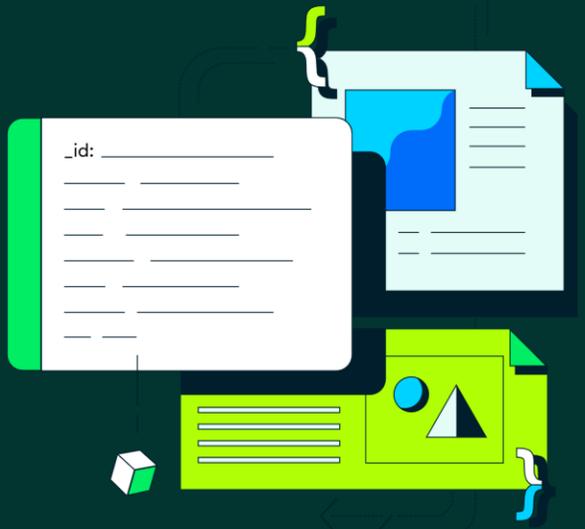


NEW Get the latest and greatest with MongoDB 6.0 - Learn more >

Build faster. Build smarter.

Get your ideas to market faster with a developer data platform built on the leading modern database. Support transactional, search, analytics, and mobile use cases while using a common query interface and the data model developers love.

Start Free Questions? Talk to us →



TRUSTED BY THOUSANDS OF ORGANIZATIONS INCLUDING



Atlas →

Developer Data Platform

Database

Search

Data Lake (Preview)

Charts

Device Sync

APIs, Triggers, Functions

Enterprise Advanced →

Enterprise software and support

Enterprise Server

Ops Manager

Enterprise Kubernetes

Operator

Community Edition →

Free software used by millions

Community Server

Cloud Manager

Community Kubernetes

Operator

Tools →

Build faster

Compass

Shell

VS Code Plugin

Atlas CLI

Database Connectors

Cluster to Cluster Sync

Mongoose ODM Support

transactional, search, analytics, and mobile use cases while using a common query interface and the data model developers love.

Start Free

Questions? Talk to us →

_id: _____



MongoDB Atlas

MongoDB Enterprise Advanced

MongoDB Community Edition

MongoDB Community Server

MongoDB Community
Kubernetes Operator

Tools

Mobile & Edge



MONGODB COMMUNITY SERVER

MongoDB Community Server Download

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

Give it a try with a free, highly-available 512 MB cluster.

Version

6.0.1 (current)

Platform

Windows

Package

msi

Download

More Options

2.MongoDB简介

2.1 mongoDB

- mongoDB的主要特性。
 - 灵活的数据模型
 - 强大的查询语言
 - 提供多种编程语言的API
 - 易于扩展
 - 支持复制和故障自动转移
- mongoDB分为社区版和企业版。
 - 社区版是开源免费版本。
 - 企业版是基于社区版订阅收费的，提供了功能更强大的操作工具、高级数据分析、数据可视化、平台集成和认证等高级功能。
 - 本项目使用社区版，在Linux平台下实现mongoDB的安装、管理、监控和高可用。



2.MongoDB简介

2.2 副本集

□ 什么是副本集

- 副本集是一组维护相同数据集的mongod实例。
- 一个副本集包含多个数据承载节点和一个仲裁器（Arbiter，可选）。
- 在数据承载节点中，只有一个成员被当作主节点，其他成员皆为从节点。
- 副本集中的节点数最好为奇数（为了选举顺利进行），成员个数最少为3个，不超过50个（最多有七个投票成员）。

□ 数据同步机制

- MongoDB在主节点上应用数据库操作,并在OPLOG（操作日志）记录操作，然后从节点通过异步进程请求操作日志并应用在自己的数据副本上。



2.MongoDB简介

2.2 副本集

□ 副本集中的成员

- MongoDB 副本集中的成员可分为三种：主节点（Primary）、次节点（Secondaries）、仲裁器，每种成员都在副本集上起着不同的作用。
- 主节点
 - 主节点是副本集中唯一能够接收写操作的成员。副本集只能有一个主节点，如果当前的主节点不可用，则通过选举确定新的主节点。
- 从节点
 - 从节点作为主节点数据集的副本，在副本集中起着数据备份、主节点候选人和负载均衡的作用。尽管客户端无法通过从节点写入数据，但是客户端可以选择从节点读取数据。
- 仲裁器
 - 在某些情况下（例如现在有一个主服务器和一个从服务器，由于成本限制，无法添加另一个从服务器），可以选择将仲裁器添加到副本集中。
 - 仲裁器既不保存数据也不能成为主节点，但是拥有投票权。



2.MongoDB简介

2.2 副本集

- 从节点从功能上又可以细分为三种类型：0优先级副本集成员、隐藏副本集成员、延迟副本集成员。
 - 0优先级副本集成员：
 - 该成员的优先级为0，即不能被选举为主节点，除此之外与其他节点没有什么区别，一般当作备用节点，以便可随时替换掉副本集中不可用的节点。
 - 隐藏副本集成员：
 - 该成员首先是0优先级副本集成员。该节点维护副本集的数据集且拥有选举投票权，但是对客户端不可见，通常作为数据备份节点。
 - 延迟副本集成员：
 - 该成员首先是0优先级副本集成员，也应该是隐藏副本集成员，因为该节点所维护的数据集相对于正常成员总是有一段时间延迟。
 - 例如当前时间是10:00，并且有一个小时的延迟，则该成员中没有早于09:00的数据，通常作为副本集的回滚备份或历史快照。



2.MongoDB简介

2.2 副本集

□ 选举

- 副本集通过选举来决定哪个节点为主节点。以下事件可以触发副本集选举。
- 向副本集添加新节点
- 副本集初始化
- 指定主节点为从节点或副本集重新配置
- 主节点响应超时（默认10秒）。

□ 以下因素影响选举。

- Heartbeats，副本集成员每两秒钟都会向彼此发送一次Heartbeats（类似ping）。如果某个成员在10秒钟内未响应，则其他成员将其标记为不可访问，该成员将不能成为主节点或被降低优先级。
- 优先级，优先级高的成员将优先获取投票权。
- 票数，得票数最多的成员将成为主节点。



3.mongodb数据库服务

3.1 任务1

任务1：实现mongodb数据库服务

步骤1：创建虚拟机并完成CentOS的安装

步骤2：完成虚拟机的主机配置、网络配置及通信测试

步骤3：通过在线方式安装mongodb

步骤4：mongodb服务管理

步骤5：查看mongodb运行信息

步骤6：配置mongodb服务为开机自启动





操作视频 / 现场演示

✓ 任务1: 实现mongoDB数据库服务

■ 任务目标

- 完成mongoDB的安装与配置
- 实现数据库、文档管理

Install MongoDB Community Edition on Red Hat or CentOS

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/#install-mongodb-community-edition>





任务总结 / 任务扩展

- ✓ MongoDB的数据逻辑结构与关系型数据库结构比较相似，都是三级存储结构，最大区别就是MongoDB中的集合是动态模式。
 - 文档
 - 文档是MongoDB存储的元数据，它是由键值对组成的数据结构，其结构类似JSON对象，字段值可以包括其他文档、数组和文档数组
 - 集合
 - MongoDB将文档存储在集合中，集合类似于关系数据库中的表。集合中的文档结构不需要相同，但为了管理方便和数据库的性能，应将相同类型的文档放在统一集合中。
 - 上限集合
 - 集合的大小固定，当其达到最大时会自动覆盖最早插入的数据。
 - 数据库
 - 多个集合组织在一起就是数据库。表6-1-2展示了MongoDB与关系型数据库的逻辑结构对比。



3.mongodb数据库服务

3.2 mongoDB服务进程

- mongod: MongoDB的守护进程, 负责处理数据请求、管理数据访问、执行后台管理。

命令详解: mongod

【语法】

mongod [选项]

【选项详解】

--config <filename>, -f <filename>	指定运行时配置选项的配置文件
--port <port>	MongoDB 实例侦听客户端连接的端口号, 默认为 27017
--bind_ip <hostnames ipaddresses Unix domain socket paths>	MongoDB 实例侦听客户端连接主机名或 IP 地址或完整的 Unix 域套接字路径, 可使用半角逗号隔开指定多个
--ipv6	启用 ipv6 支持, 默认禁用
--maxConns <number>	接受的最大连接数
--logpath <path>	日志文件路径
--syslog	将日志信息发送到主机的 syslog 系统, Windows 平台下不支持
--keyFile <file>	指定密钥文件的路径, 该密钥存储在分片群集或副本集成员相互认证的共享密钥
--auth	启用访问控制

操作命令+配置文件+脚本程序+结束



3.mongodb数据库服务

3.2 mongoDB服务进程

- mongo: mongoDB的交互式JavaScript Shell (mongo shell) 接口, 它提供了一些接口函数用于管理员对数据库系统进行管理。

命令详解: mongo

【语法】

mongo [选项]

【选项详解】

--port <port>	MongoDB 实例监听的端口号, 默认为 27017
--host <hostname>	运行 MongoDB 实例的主机名, 可使用半角逗号隔开指定多个, 默认为 localhost
--username <username>, -u <username>	进行身份验证的用户名
--password <password>, -p <password>	进行身份验证的密码
--networkMessageCompressors <string>	mongo shell 与 mongod 之间的通信启用网络压缩, 有三种压缩方式: snappy、zlib、zstd
--ipv6	启用 ipv6
<db name>	要连接的数据库名称
--authenticationDatabase <dbname>	指定身份验证的数据库



3.mongodb数据库服务

3.2 mongoDB服务进程

- mongodump: mongoDB的数据备份工具, 可将数据导出为二进制文件。

命令详解: mongodump

【语法】

mongodump [选项]

【选项详解】

--uri=<connectionString>	连接字符串, 用于指定要连接的主机地址以及连接选项
--host=<hostname> <:port>, -h=<hostname> <:port>	运行 MongoDB 实例的主机名, 可使用半角逗号隔开指定多个, 默认为 localhost
--port=<port>	MongoDB 实例监听的端口号, 默认为 27017
--username <username>, -u <username>	指定身份验证的用户名
--password <password>, -p <password>	指定身份验证的密码
--authenticationDatabase=<dbname>	指定身份验证的数据库
--db=<database>, -d=<database>	指定导出的数据库名称
--collection=<collection>, -c=<collection>	指定导出的集合名称
--query=<json>, -q=<json>	指定查询语句, 筛选数据
--out=<path>, -o=<path>	



工具

Compass。MongoDB 的图形用户界面。

Compass 是一款用于查询、优化和分析 MongoDB 数据的交互式工具。可供获取关键见解，拖放构建管道等。

立即下载 阅读文档 →



Compass 是由 MongoDB 构建并用于 MongoDB 的图形用户界面，可供轻松处理您的数据。Compass 在单个集中式界面中提供模式分析、索引优化和聚合管道等各种功能。

- Compass 可免费下载和使用
- 适用于 Linux、Mac 和 Windows

功能概览

[查看 Compass 文档](#) →



使用我们多功能的图形用户界面与数据交互

从单一的用户界面实施更新、查找数据并完成其他关键操作。



快速获取您需要的数据

直接从 Compass 查询栏中搜索您的文档和集合。



拖放聚合管道构建器

通过 200 多个运算符创建管道，然后以您的首选语言导出，以便在应用程序中使用。



在 Compass 中直接访问 MongoDB Shell

通过在嵌入式 MongoDB Shell 中运行命令来掌控您的部署。



根据需要导入和导出数据

Compass 使您能够轻松地在集中导入和导出 JSON 和 CSV 文件。



对架构进行可视化、验证和分析

了解文档的结构。跨各个集合创建、更新和实施模式。



详细评估查询性能

利用可视化解释计划检查单个查询并分解多阶段查询。



使用索引分析来优化查询

查看索引利用率，然后添加或删除索引以提高查询性能。

3.mongodb数据库服务

3.3 任务2

任务2：使用mongoDB Compass管理MongoDB数据库服务

步骤1：创建管理账户

步骤2：开启授权访问

步骤3：配置MongoDB开启远程管理并暂时管理安全措施

步骤4：在本地主机安装MongoDB Compass管理工具

步骤5：使用MongoDB Compass连接MongoDB

步骤6：使用MongoDB Compass监控MongoDB服务器





操作视频 / 现场演示



✓ 任务2：使用mongoDB Compass管理MongoDB数据库服务

■ 任务目标

- 实现MongoDB远程访问
- 完成MongoDB Compass的安装
- 完成使用MongoDB Compass管理MongoDB





命令指南 / 操作引导

1. #重启出错后, 执行清理工作并重新启动
2. `rm /var/lib/mongodb/mongod.lock`
3. `rm /tmp/mongodb-27017.sock`
4. #mongoDB管理员账号的创建
5. `[root@Project-06-Task-01 ~]# mongo`
6. 切换到MongoDB内置的admin数据库
7. `> use admin`
8. `> db.createUser(`
9. `... {`
10. `... user: "admin",`
11. `... pwd: "centos@mongodb#123",`
12. `... roles: [{ role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase"]`
13. `... }`
14. `...)`
15. #配置文件 /etc/mongod.conf
16. #开启身份验证
17. `security:`
18. `... authorization: enabled`
19. `net:`
20. `... port: 27017`
21. `... bindIp: 0.0.0.0`



4.高可用

4.1 任务3

任务3：实现mongoDB高可用

步骤1：创建服务器并完成操作系统的安装和网络配置

步骤2：在服务器-1上操作，配置副本集

步骤3：在服务器-2上操作，配置副本集

步骤4：在服务器-3上操作，配置副本集

步骤5：在服务器-1上操作，初始化副本集

步骤6：副本集的应用测试



4.高可用

4.1 任务3

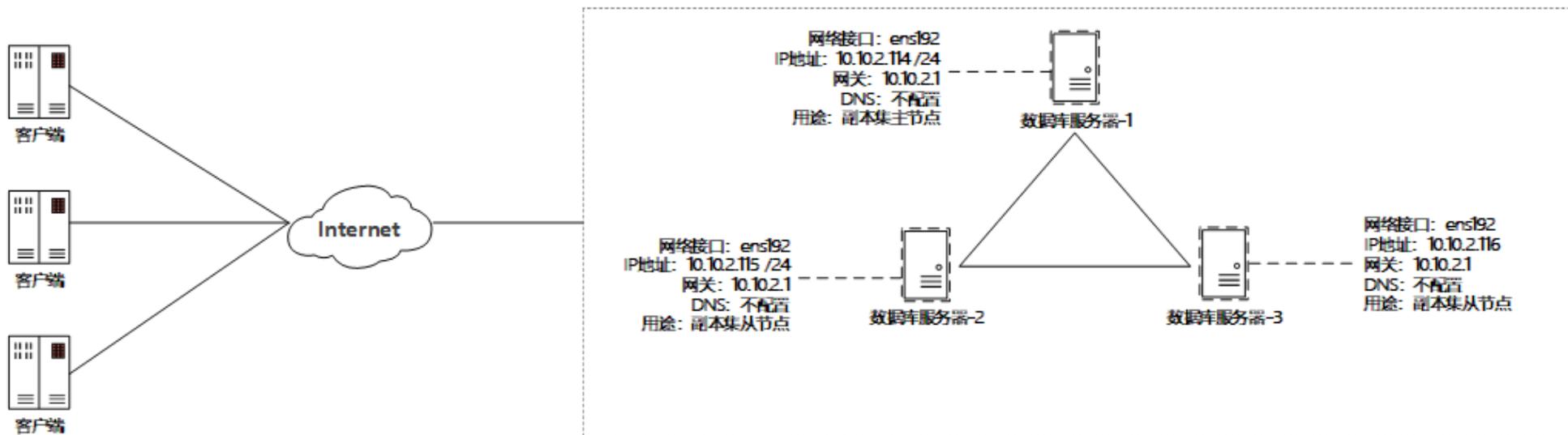
表 6-3-1 MongoDB 用户列表

用户名	角色	描述
admin	userAdminAnyDatabase、readWriteAnyDatabase	用于用户管理、数据库管理
repAdmin	clusterAdmin、readWriteAnyDatabase	用于集群管理、数据库管理



4.高可用

4.1 任务3



4.高可用

4.1 任务3

表 6-3-2 服务器规划表

序号	虚拟机名称	业务名称	作用
1	VM-Project-06-Task-02-10.10.2.114	服务器-1	作为副本集主节点
2	VM-Project-06-Task-03-10.10.2.115	服务器-2	作为副本集从节点
3	VM-Project-06-Task-04-10.10.2.116	服务器-3	作为副本集从节点





操作视频 / 现场演示

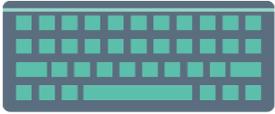


- ✓ 任务3：实现mongoDB高可用
 - 任务目标
 - 实现mongoDB副本集
 - 测试mongoDB副本集



在服务器-1上操作，配置副本集





命令指南 / 操作引导

1. #使用openssl命令生成密钥文件并选择存放在/var/lib/mongo/keyFile.file中
2. [root@Project-06-Task-02 ~]# openssl rand -base64 756 > /var/lib/mongo/keyFile.file
3. [root@Project-06-Task-02 ~]# chmod 400 /var/lib/mongo/keyFile.file

4. #修改密钥文件拥有者和用户组为mongod
5. [root@Project-06-Task-02 ~]# chown mongod:mongod /var/lib/mongo/keyFile.file

6. #使用scp命令将密钥文件分发至服务器-2
7. [root@Project-06-Task-02 ~]# scp /var/lib/mongo/keyFile.file root@10.10.2.115:/var/lib/mongo/

8. #使用SSH协议首次连接主机时的提示, 输入yes, 按【Enter】键继续
9. The authenticity of host '10.10.2.115 (10.10.2.115)' can't be established.
10. ECDSA key fingerprint is SHA256:o8PXGC1g4S2wbxS6lbGcLh/f+xSGveavweSgIKlogEA.
11. Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
12. Warning: Permanently added '10.10.2.115' (ECDSA) to the list of known hosts.

13. #输入要分发至主机的root账户密码, 按【Enter】键继续
14. root@10.10.2.115's password:
15. keyFile.file 100% 1024 969.8KB/s 00:00

16. #使用scp命令将密钥文件分发至服务器-3
17. [root@Project-06-Task-02 ~]# scp /var/lib/mongo/keyFile.file root@10.10.2.116:/var/lib/mongo/

18. #使用SSH协议首次连接主机时的提示, 输入yes, 按【Enter】键继续
19. The authenticity of host '10.10.2.116 (10.10.2.116)' can't be established.
20. ECDSA key fingerprint is SHA256:o8PXGC1g4S2wbxS6lbGcLh/f+xSGveavweSgIKlogEA.
21. Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
22. Warning: Permanently added '10.10.2.116' (ECDSA) to the list of known hosts.

23. #输入要分发至主机的root账户密码, 按【Enter】键继续
24. root@10.10.2.116's password:
25. keyFile.file 100% 1024 895.9KB/s 00:00

26. #配置文件/etc/mongod.conf
27. replication:
28. replSetName: "db-cluster-mongodb"





命令指南 / 操作引导

1. #使用chmod命令赋予密钥文件400权限
2. [root@Project-06-Task-02 ~]# chmod 400 /var/lib/mongo/keyFile.file
3. #修改密钥文件拥有者和用户组为mongod
4. [root@Project-06-Task-02 ~]# chown mongod:mongod /var/lib/mongo/keyFile.file
5. #配置文件 /etc/mongod.conf
6. # mongod.conf
1. # network interfaces
2. net:
3. port: 27017
4. bindIp: 0.0.0.0 # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.bindIpAll setting.
5. security:
6. authorization: enabled
7. clusterAuthMode: keyFile
8. keyFile: /var/lib/mongo/keyFile.file
9. replication:
10. replSetName: db-cluster-mongodb



在服务器-2上操作，配置副本集





命令指南 / 操作引导

1. #使用chmod命令赋予密钥文件400权限
2. [root@Project-06-Task-03 ~]# chmod 400 /var/lib/mongo/keyFile.file
3. #修改密钥文件拥有者和用户组为mongod
4. [root@Project-06-Task-03 ~]# chown mongod:mongod /var/lib/mongo/keyFile.file
5. #配置文件 /etc/mongod.conf
6. # mongod.conf
1. # network interfaces
2. net:
3. port: 27017
4. bindIp: 0.0.0.0 # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.bindIpAll setting.
5. security:
6. authorization: enabled
7. clusterAuthMode: keyFile
8. keyFile: /var/lib/mongo/keyFile.file
9. replication:
10. replSetName: db-cluster-mongodb



在服务器-3上操作，配置副本集





命令指南 / 操作引导

1. #使用chmod命令赋予密钥文件400权限
2. [root@Project-06-Task-04 ~]# chmod 400 /var/lib/mongo/keyFile.file
3. #修改密钥文件拥有者和用户组为mongod
4. [root@Project-06-Task-04 ~]# chown mongod:mongod /var/lib/mongo/keyFile.file
5. #配置文件 /etc/mongod.conf
6. # network interfaces
7. net:
8. port: 27017
9. bindIp: 0.0.0.0 # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, alternatively, use the net.bindIpAll setting.
10. security:
11. authorization: enabled
12. clusterAuthMode: keyFile
13. keyFile: /var/lib/mongo/keyFile.file
14. replication:
15. replSetName: db-cluster-mongodb



在服务器-1上操作，初始化副本集





命令指南 / 操作引导

```

1. #使用mongo命令连接MongoDB客户端，使用rs.initiate()方法初始化副本集。
2. [root@Project-06-Task-02 ~]# mongo
3. > rs.initiate({
4.     _id: "db-cluster-mongodb",
5.     members: [
6.         { _id: 0, priority:2,host: "10.10.2.114:27017" },
7.         { _id: 1, host: "10.10.2.115:27017" },
8.         { _id: 2, host: "10.10.2.116:27017" }
9.     ]
10. })
11. db-cluster-mongodb:STARTUP> quit()

12. #使用rs.status()方法查看副本集状态，验证服务器-1是否为主节点。
13. [root@Project-06-Task-02 ~]# mongo
14. Enter password:
15. db-cluster-mongodb:STARTUP> rs.status()
16. {
17.     #副本集名称
18.     "set" : "db-cluster-mongodb",
19.     "members" : [
20.         {
21.             #成员的唯一标识符
22.             "_id" : 0,
23.             "name" : "10.10.2.114:27017",
24.             #成员健康状态：1-正常，0-不可用
25.             "health" : 1,
26.             #副本集成员的状态，1表示主节点，2表示从节点
27.             "state" : 1,
28.             #与state相对应，具体如表6-3-6所示
29.             "stateStr" : "PRIMARY"
30.         },
31.         { "_id" : 1, ... },
32.         { "_id" : 2, ... } ],
33.     }
34. db-cluster-mongodb:PRIMARY> quit()

```





命令指南 / 操作引导

```
1. #为副本集创建账号
2. [root@Project-06-Task-02 ~]# mongo
3. Enter password:
4. db-cluster-mongodb:PRIMARY> use admin
5. switch to admin
6. db-cluster-mongodb:PRIMARY> db.createUser(
7.   ... {
8.   ...   user: "admin",
9.   ...   pwd: "centos@mongodb#123",
10.  ...   roles: [ { role: "userAdminAnyDatabase", db: "admin" } , "readWriteAnyDatabase" ]
11.  ... }
12.  ... )
13. db-cluster-mongodb:PRIMARY> db.getSiblingDB("admin").auth("admin", passwordPrompt())
14. Enter password:
15. db-cluster-mongodb:PRIMARY> db.getSiblingDB("admin").createUser(
16.  ... {
17.  ...   user : "repAdmin",
18.  ...   pwd : "centos@mongodb#123",
19.  ...   roles : [ { role : "clusterAdmin", db: "admin" } , "readWriteAnyDatabase" ]
20.  ... }
21.  ... )
22. db-cluster-mongodb:PRIMARY> quit()
```



4.高可用

4.2 任务4

任务4：监控mongoDB

步骤1：开启MongoDB云监控服务

步骤2：使用云监控查看MongoDB状态

步骤3：使用mongostat工具监控MongoDB

步骤4：使用MongoDB命令监控MongoDB





操作视频 / 现场演示



✓ 任务4: 监控mongoDB

■ 任务目标

- 实现使用云服务监控MongoDB
- 实现使用mongostat监控MongoDB





命令指南 / 操作引导

1. [root@Project-06-Task-03 ~]# mongo -u repAdmin -p
2. MongoDB shell version v4.2.6
3. Enter password:
4. db-cluster-mongodb:PRIMARY> db.enableFreeMonitoring()
5. {
6. #enabled表示已经开启
7. "state" : "enabled",
8. "message": "To see your monitoring data db.disableFreeMonitoring().",
9. #查看监控的url
10. "url" : "https://cloud.mongodb.com/freemonitoring/cluster/IYTCE5P3UUWOZ2XT7DM7DQDXI6BNTXC",
11. "userReminder" : "",
12. "ok" : 1
13. }
14. db-cluster-mongodb:PRIMARY> quit()





命令指南 / 操作引导

1. #使用mongostat监控单个mongod示例
2. [root@Project-06-Task-04 ~]# mongostat -u repAdmin -p centos@mongodb#123
3. --authenticationDatabase=admin
4. -o=insert,query,update,delete,net_in,net_out,conn,repr

5. #使用db.serverStatus()返回数据库状态的概述信息
6. [root@Project-06-Task-03 ~]# mongo -u repAdmin -p
7. MongoDB shell version v4.2.6
8. Enter password:
9. db-cluster-mongodb:PRIMARY> db.serverStatus()

10. #使用db.stats()返回单个数据库的状态信息
11. [root@Project-06-Task-03 ~]# mongo -u repAdmin -p
12. MongoDB shell version v4.2.6
13. Enter password:
14. db-cluster-mongodb:PRIMARY> use seconddb
15. db-cluster-mongodb:PRIMARY> db.stats()

16. #使用db.collection.collStats()返回指定集合的各种存储统计信息
17. [root@Project-06-Task-03 ~]# mongo -u repAdmin -p
18. MongoDB shell version v4.2.6
19. Enter password:
20. db-cluster-mongodb:PRIMARY> use seconddb
21. db-cluster-mongodb:PRIMARY> use seconddb
22. db-cluster-mongodb:PRIMARY> db.test_collection.stats()



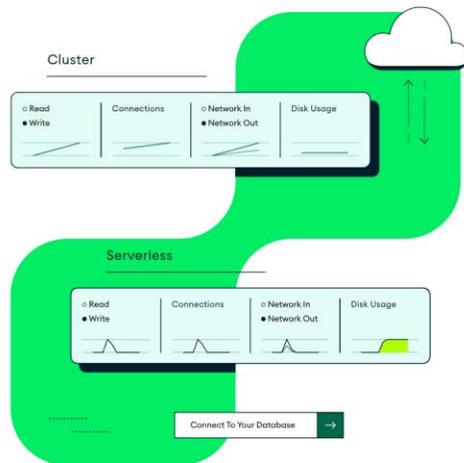
MONGODB ATLAS

MongoDB Atlas。 多云应用程序数据平台。

一款云数据库和数据服务的集成套件，可以加快使用数据进行构建的速度，还能简化相关构建方式。

免费试用

联系销售团队 →



深受数以千计组织的信赖，包括



