

Linux 操作系统

(基于 openEuler)

第7章：使用MongoDB实现数据库服务

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

<https://internet.hactcm.edu.cn>
<http://www.51xueweb.cn>

河南中医药大学信息技术学院智能医疗教研室
河南中医药大学医疗健康信息工程技术研究所

2025.8

提纲

- 了解MongoDB
 - 非关系型数据库
 - CAP、ACID、BASE
 - MongoDB
- 实现MongoDB数据库服务
 - 单机模式的数据库服务
 - 集群模式的数据库服务
- 管理与监控
 - MongoDB Database Tools
 - MongoDB Shell
 - MongoDB Compass
 - Percona Monitoring and Management



1. 了解MongoDB

1.1 非关系型数据库 (No-SQL DBMS)

- 互联网具有数据规模庞大和数据结构动态化等特点：
 - 关系型数据库在处理此类问题时不仅麻烦而且性能方面也达不到要求。
 - 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，其抛弃了关系型数据库的强制一致性和事务等特性。
 - 关系型数据库的特性：
 - 支持复杂查询
 - 支持标准的SQL语言
 - 数据完整性高
 - 非关系型数据库的特性：
 - 存储的伸缩性更强
 - 数据操作的并发性能更强
 - 更容易通过多节点部署提高可用性
 - 数据模型更加灵活



网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

5分钟解释关系型和非关系型数据库

5分钟解释关系型 和非关系型数据库

https://www.bilibili.com/video/BV1Bx4y1j7Js/?vd_source=50888f0a82ef07bed999f62cf37946c1

Problem: IoT connectivity disruptions.
Solution: xtrameDB [Learn how](#)

Knowledge Base of Relational and NoSQL Database Management Systems provided by Redote Software

Home DB-Engines Ranking Systems Encyclopedy Blog Sponsors Search Vendor Login

Featured Products: [RaimaDB](#) [Synology](#) [Nesdi](#) [Mux](#) [DelmariaDB](#)

Select a ranking

- Complete ranking
- Relational DBMS
- Key-value stores
- Document stores
- Time Series DBMS
- Graph DBMS
- Search engines
- Object oriented DBMS
- IoT
- Vector DBMS
- Wide column stores
- Multivariate DBMS
- Spatial DBMS
- Native SQL DBMS
- Event Stores
- Correlation stores
- Complex Event Processing
- Neurological DBMS
- Columnar

Special reports

- Ranking by database model
- Open source vs. commercial

Featured Products

Vector database that provides fast, flexible, and scalable solutions for enterprise implementations. [See Detailed Story For Free](#)

Bring all your data to Databricks or Astra and make it available to all your applications by the most complete vector database available. [See Us Here](#)

Relational database for mission-critical applications.

Problem: IoT connectivity disruptions.
Solution: xtrameDB [Learn how](#)

Knowledge Base of Relational and NoSQL Database Management Systems provided by Redote Software

Home DB-Engines Ranking Systems Encyclopedy Blog Sponsors Search Vendor Login

Featured Products: [RaimaDB](#) [Synology](#) [Nesdi](#) [Mux](#) [DelmariaDB](#)

Select a ranking

- Complete ranking
- Relational DBMS
- Key-value stores
- Document stores
- Time Series DBMS
- Graph DBMS
- Search engines
- Object oriented DBMS
- IoT
- Vector DBMS
- Wide column stores
- Multivariate DBMS
- Spatial DBMS
- Native SQL DBMS
- Event Stores
- Correlation stores
- Complex Event Processing
- Neurological DBMS
- Columnar

Special reports

- Ranking by database model
- Open source vs. commercial

Featured Products

Vector database that provides fast, flexible, and scalable solutions for enterprise implementations. [See Detailed Story For Free](#)

Bring all your data to Databricks or Astra and make it available to all your applications by the most complete vector database available. [See Us Here](#)

Relational database for mission-critical applications.

Ranking - Complete Ranking

DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.

423 systems in ranking, October 2024

Database Model

Rank	DBMS	Model	Score
1	Oracle	Relational, Multi-model	1309.45 +22.89 +48.03
2	MySQL	Relational, Multi-model	1022.78 +47.19 +10.16
3	Microsoft SQL Server	Relational, Multi-model	802.09 +4.47 +46.78
4	PostgreSQL	Relational, Multi-model	652.16 +7.86 +13.34
5	MongoDB	Document, Multi-model	405.21 +5.92 +26.21
6	RainDB	Relational, Multi-model	149.63 +2.50 +12.33
7	SQLite	Relational	140.60 +6.88 +17.36
8	Elasticsearch	Multi-model	133.85 +3.06 +3.20
9	IBM DB2	Relational, Multi-model	122.77 +0.44 +10.10
10	SQLite	Relational	101.91 +4.22 +23.22
11	Apache Cassandra	Wide column, Multi-model	97.95 +1.94 +12.11
12	Microsoft Access	Relational	92.15 +1.44 +24.46
13	Spilunk	Search engine	91.27 +1.79 +1.05
14	DelmariaDB	Multi-model	85.60 +1.35 +9.78
15	Oracle RDB	Relational, Multi-model	84.89 +1.45 +47.77
16	Google BigQuery	Relational, Multi-model	78.53 +1.58 +4.40
17	Amazon DynamoDB	Multi-model	71.85 +1.78 +9.07
18	Apache Hive	Relational	52.57 +0.50 +16.61
19	Google BigQuery	Relational	51.18 +1.48 +3.50
20	FlixFaker	Relational	44.40 +0.82 +0.00
21	Neo4j	Graph	42.51 +1.17 +2.87
22	SAP HANA	Relational, Multi-model	41.21 +0.11 +0.21
23	Teradata	Relational, Multi-model	40.69 +1.49 +0.00
24	Apache Solr	Search engine, Multi-model	32.98 +1.17 +12.36
25	SAP Adaptive Server	Relational	32.40 +0.27 +0.12
26	Apache Iceberg	Wide column	27.19 +0.21 +3.50
27	Microsoft Azure Cosmos DB	Multi-model	24.50 +0.47 +4.80
28	InfluxDB	Time Series, Multi-model	22.39 +0.27 +2.01
29	PostGIS	Spatial, Multi-model	20.58 +0.40 +4.70
30	Firebird	Relational	19.02 +0.44 +0.00
31	Microsoft Azure Synapse Analytics	Relational	18.72 +0.19 +4.41
32	Mechanich	Key-value	17.79 +0.55 +3.30

Ranking - Trend

DB-Engines Ranking - Trend Popularity

The DB-Engines Ranking ranks database management systems according to their popularity. Read more about the [method](#) of calculating the scores.

DB-Engines Ranking

Click on a system in the legend to see the year to year trend

DB-Engines Ranking

- Oracle
- MySQL
- Microsoft SQL Server
- PostgreSQL
- IBM DB2
- SQLite
- Apache Cassandra
- Microsoft Access
- Elasticsearch
- IBM DB2
- Oracle RDB
- Google BigQuery
- Amazon DynamoDB
- Apache Hive
- Google BigQuery
- FlixFaker
- Neo4j
- SAP HANA
- Teradata
- Apache Solr
- SAP Adaptive Server
- Apache Iceberg
- Microsoft Azure Cosmos DB
- InfluxDB
- PostGIS
- Firebird
- Microsoft Azure Synapse Analytics
- Mechanich

Ranking - Trend

DB-Engines Ranking - Trend Popularity

The DB-Engines Ranking ranks database management systems according to their popularity. Read more about the [method](#) of calculating the scores.

DB-Engines Ranking

Click on a system in the legend to see the year to year trend

DB-Engines Ranking

- Oracle
- MySQL
- Microsoft SQL Server
- PostgreSQL
- IBM DB2
- SQLite
- Apache Cassandra
- Microsoft Access
- Elasticsearch
- IBM DB2
- Oracle RDB
- Google BigQuery
- Amazon DynamoDB
- Apache Hive
- Google BigQuery
- FlixFaker
- Neo4j
- SAP HANA
- Teradata
- Apache Solr
- SAP Adaptive Server
- Apache Iceberg
- Microsoft Azure Cosmos DB
- InfluxDB
- PostGIS
- Firebird
- Microsoft Azure Synapse Analytics
- Mechanich

SingleStore.

The platform to build your intelligent applications. [See Us Here](#)

Bring all your data to Databricks or Astra and make it available to all your applications by the most complete vector database available. [See Us Here](#)

Relational database for mission-critical applications.

Upcoming events

PaaS Data Community Summit
1-6 November 2024
8 November 2024

Microsoft SQL Server event
PaaS Data Community Summit
1-6 November 2024
8 November 2024

[PaaS Data Community Summit](#)

Share this page

About Us Advertising and Services Privacy Policy Contact

1. 了解MongoDB

1.1 非关系型数据库 (No-SQL DBMS)

- 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。
- 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
- 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

网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

1. 了解MongoDB

1.1 非关系型数据库 (No-SQL DBMS)

- 非关系型数据库没有统一架构，通常具有较强的应用场景适应性，不同应用场景下应选用不同的产品。
- 非关系型数据库常见的有键值数据库、列族数据库、文档数据库和图形数据库。

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、GaussDB(for Redis)、TcaplusDB	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、GaussDB(for Mongo)、GaussDB(for Influx)	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

网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

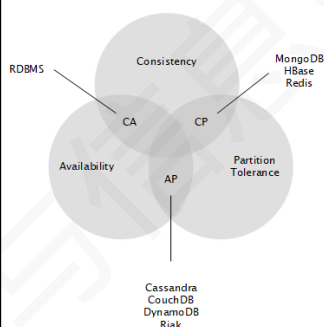
1. 了解MongoDB

1.2 CAP、ACID、BASE

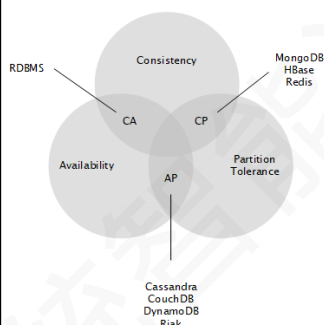
- CAP理论是由Eric Brewer在2001年提出的，指出对于一个分布式计算系统来说，不可能同时满足以下三点。
 - 一致性（Consistency）
 - 一致性是指更新操作成功后，所有节点在同一时间的数据完全一致。
 - 可用性（Availability）
 - 可用性是指用户访问数据时，系统是否能在正常响应时间返回结果。
 - 分区容错性（Partition Tolerance）
 - 分区容错性是指分布式系统在遇到某节点或网络分区故障的时候，仍然能够对外提供满足一致性和可用性的服务。

网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>



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. 了解MongoDB

1.2 CAP、ACID、BASE

- 关系型数据库支持事务的ACID特性，即原子性、一致性、隔离性和持久性，这4种特性保证在事务过程中数据的正确性，具体描述如下。
 - 原子性 (Atomicity) 。
 - 一个事务的所有操作步骤被看成一个动作，所有的步骤要么全部完成，要么一个也不会完成。如果在事务过程中发生错误，则会回滚到事务开始前的状态，将要被改变的数据库记录不会被改变。
 - 一致性 (Consistency) 。
 - 一致性是指在事务开始之前和事务结束以后，数据库的完整性约束没有被破坏，即数据库事务不能破坏关系数据的完整性及业务逻辑上的一致性。
 - 隔离性 (Isolation) 。
 - 主要用于实现并发控制，隔离能够确保并发执行的事务按顺序一个接一个地执行。通过隔离，一个未完成事务不会影响另外一个未完成事务。
 - 持久性 (Durability) 。
 - 一旦一个事务被提交，它应该持久保存，不会因为与其他操作冲突而取消这个事务。
 - 从事务的4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. 了解MongoDB

1.2 CAP、ACID、BASE

- BASE是对CAP理论中一致性和可用性进行权衡的结果，核心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。
- 从以上可以看出，关系型数据库与非关系型数据库各有特点，对于数据库的选型应与自己的业务结合，充分考量。

网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

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	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	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	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	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	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	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。

1. 了解MongoDB

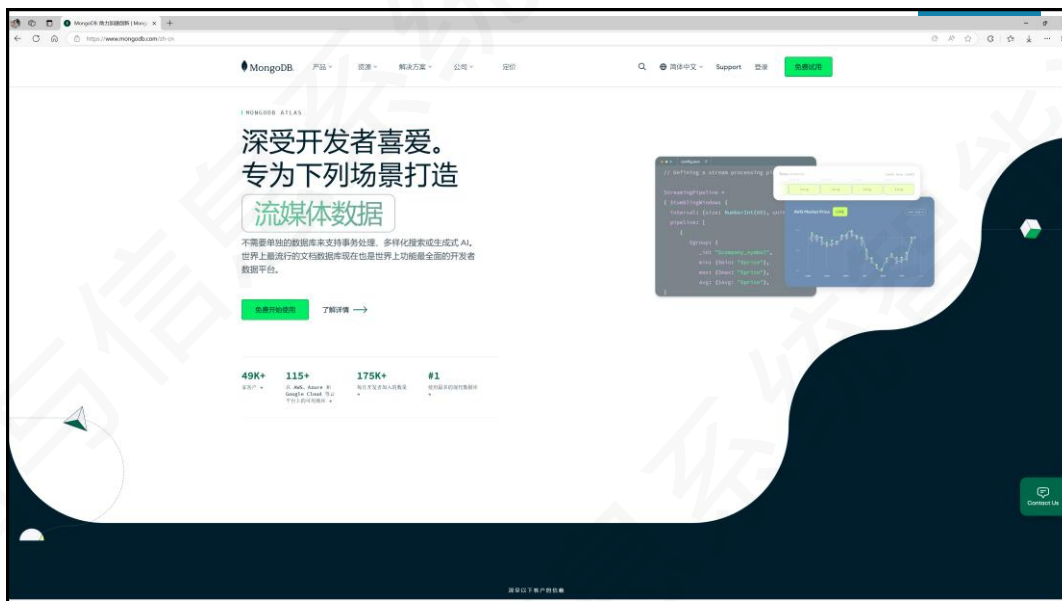
1.3 MongoDB

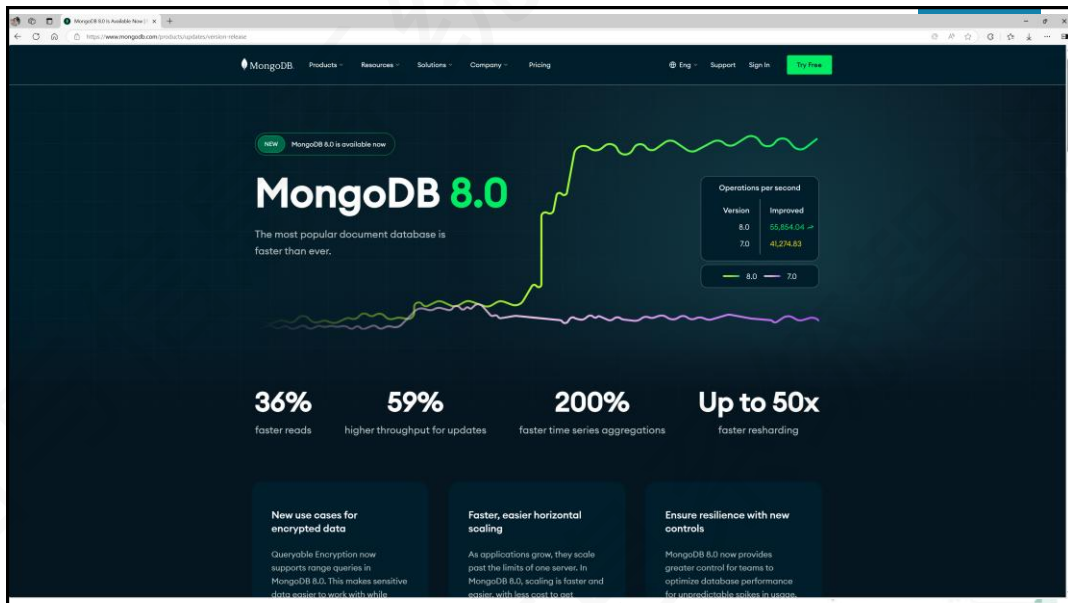
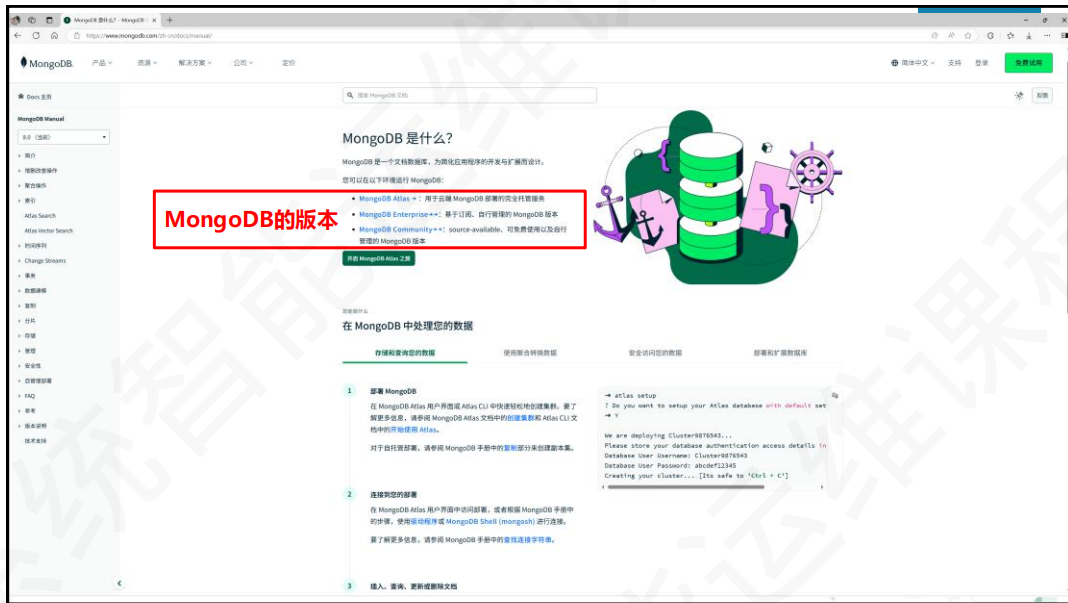
- MongoDB是一个表结构自由、开源、可扩展、面向文档的数据库，旨在为Web应用程序提供高性能、高可用且易扩展的数据存储解决方案。
- MongoDB支持多文档事务、连接查询，是较为接近关系型数据库的非关系型数据库。
- MongoDB的主要特性：
 - 灵活的数据模型
 - 强大的查询语言
 - 提供多种编程语言的API
 - 易于扩展
 - 支持复制和故障自动转移

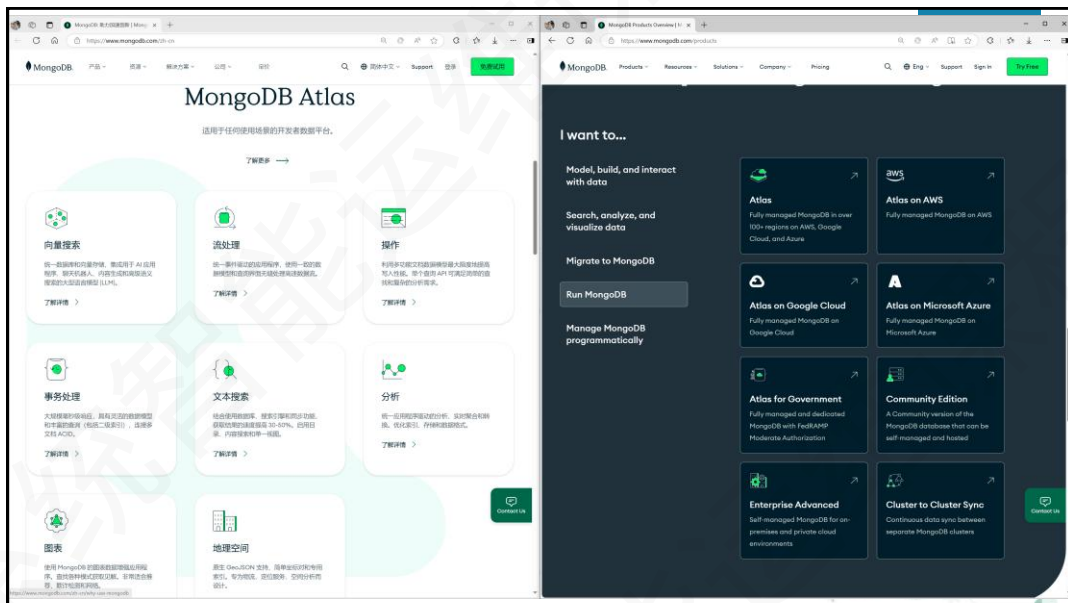


网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>







20

1. 了解MongoDB

1.4 MongoDB的数据逻辑结构

- 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，但它们最大的区别就是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，例如：

```
{
  name:'Su'
  age:'26',
  status:'A',
  groups:['news','sports' ]
}
```

- **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。如表7-1-2所示展示了MongoDB与关系型数据库的逻辑结构对比。

MongoDB与关系型数据库的逻辑结构对比

MongoDB	关系型数据库
文档 (document)	行 (row)
集合 (collection)	表 (table)
数据库 (database)	数据库 (database)



2. 实现MongoDB数据库服务

2.1 单机模式的数据库服务



单机模式的数据库服务

任务目标：

- 安装MongoDB。
- 实现MongoDB服务管理。
- 实现MongoDB服务状态查看。

操作步骤：

- 通过RPM安装
- 配置存储库
- 通过在线方式安装
- 配置服务并测试

操作演示：



2. 实现MongoDB数据库服务

2.2 MongoDB副本集

□ MongoDB集群

- 在MongoDB官方描述中，通常指副本集（Replica Set）或分片集群（Sharded Cluster）。
- 副本集是存储相同数据副本实例，是一组维护相同数据集的实例。
 - 节点（服务器）之间通过自动复制数据，以提供冗余和防止系统故障或计划内维护时停机的保护，确保高可用性。
 - 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。
 - 副本集节点数最好为奇数，成员数最少为3个，不超过50个，最多7个投票成员。
- 分片集群是分布在许多分片（服务器）上的数据集的集合。
 - 在集合级别对数据进行分片，将集中的文档分布在集群中的分片之间，以实现处理海量数据增长的水平可扩展性和更好的读写操作性能。



2. 实现MongoDB数据库服务

2.2 MongoDB副本集

□ 副本集中的成员

- MongoDB 副本集中的成员可分为3种：主节点（Primary）、从节点（Secondaries）和仲裁器（Arbiter），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。



2. 实现MongoDB数据库服务

2.2 MongoDB副本集

- **副本集的从节点**从功能上可细分为3种属性：
 - 优先级为0的副本集成员（Priority 0 Replica Set Members）
 - 该成员的优先级为0（也作为备用成员），不可选举为主节点，但可投票，可驻留在辅助数据中心或充当冷备用数据库。
 - 隐藏副本集成员（Hidden Replica Set Members）
 - 该成员维护副本集的数据集且拥有选举投票权，但对客户端不可见，通常作数据备份节点。
 - 可阻止应用程序从中读取数据，这使它可以运行需要与正常流量隔离的应用程序。
 - 该成员的优先级为0，具有优先级为0的副本集成员的属性。
 - 延迟副本集成员（Delayed Replica Set Members）
 - 该成员所维护的数据集相对于正常成员总是有一段时间的延迟，保留运行中的“历史”快照，以用于从某些错误（例如意外删除的数据库）中恢复。
 - 该成员既是优先级为0的成员，以防止延迟成员成为主要成员；也是隐藏成员，始终阻止应用程序查看和查询延迟的成员。

2. 实现MongoDB数据库服务

2.2 MongoDB副本集

- **副本集的仲裁器**
 - 该成员没有数据副本，也不会成为主节点，主要用来选举投票。
 - 当副本集的节点数据为偶数时，需要添加一个仲裁节点。
 - 仲裁节点因为没有数据，只参与投票，所以仲裁节点需要的资源很少，但官方不建议将仲裁节点部署在副本集的其他节点上。

2. 实现MongoDB数据库服务

2.2 MongoDB副本集

- **副本集的选举。**
 - 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10s）
 - 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10s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。

网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>



2. 实现MongoDB数据库服务

2.3 集群模式的数据库服务



集群模式的数据库服务

任务目标：

- 实现MongoDB replica set的部署。
- 完成MongoDB replica set的测试。

操作步骤：

- 设计方案
- 完成节点部署
- 初始化副本集
- 应用测试

操作演示：

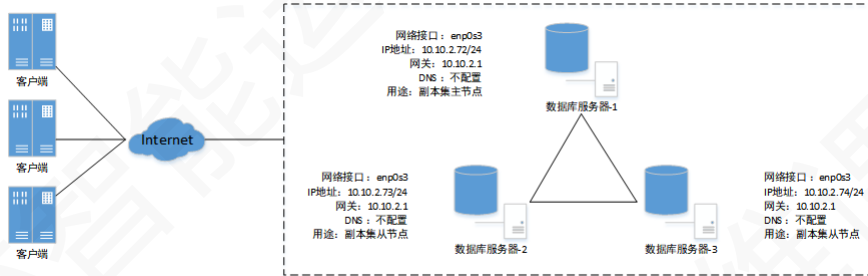


网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>



集群模式的数据库服务拓扑结构

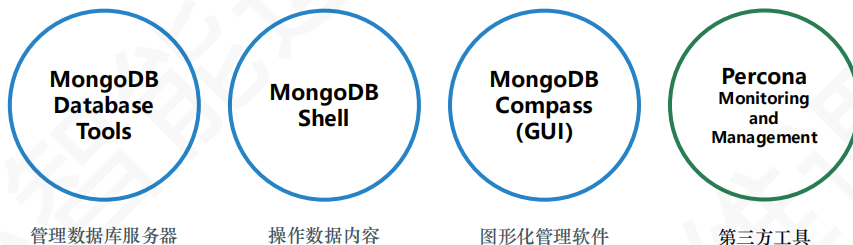


本任务配置MongoDB 副本集的流程



3. 管理和监控

command-line utilities



网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn> 棒棒堂智能运维 <http://www.51xueweb.cn>

3. 管理和监控

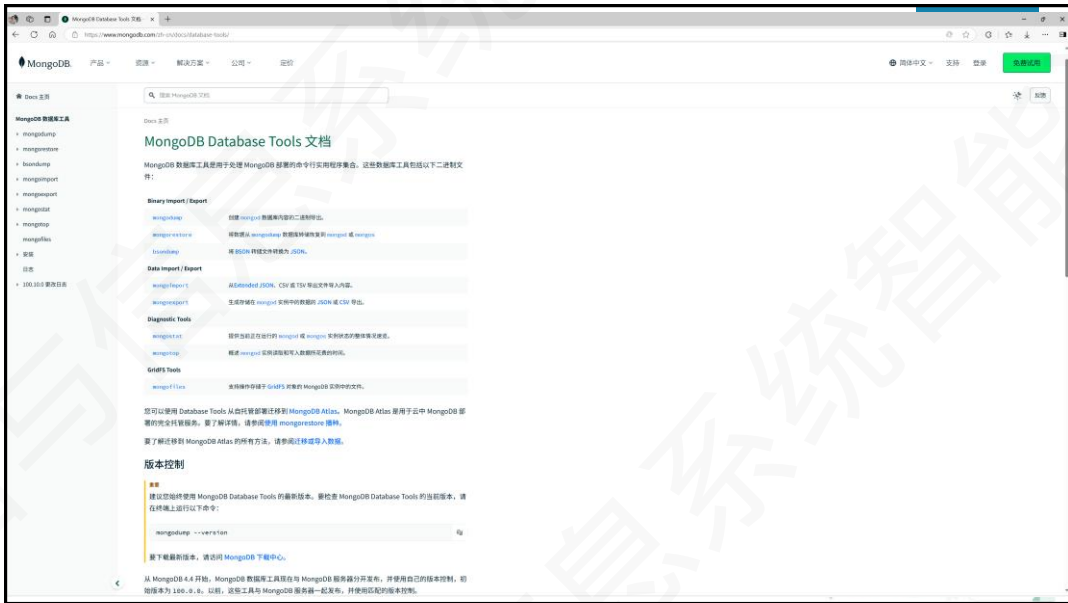
3.1 MongoDB Database Tools

- MongoDB Database Tools 是用于 MongoDB 部署和管理的命令行工具集。
 - 从 MongoDB 4.4 开始, MongoDB Database Tools 与 MongoDB 服务器分开发布, 并使用独立的版本控制, 初始版本号为 100.0.0。(推荐使用最新版本)
 - MongoDB Database Tools 与 MongoDB 同时发布, 有版本兼容性。(mongodump --version 查看版本)

MongoDB Database Tools 工具集列表

名称	类型	描述
mongodump	Binary Import / Export	以二进制格式导出数据
mongorestore	Binary Import / Export	将 mongodump 导出的数据进行转储还原
bsondump	Binary Import / Export	将 BSON 格式转换为 JSON
mongoimport	Data Import / Export	导入数据, 支持 JSON、CSV 或 TSV 格式
mongoexport	Data Import / Export	导出数据, 支持 JSON、CSV 格式
mongostat	Diagnostic Tools	查看单实例或集群的状态
mongotop	Diagnostic Tools	查看实例的读写耗时
mongofiles	GridFS Tools	从其他成员来看, 该成员不可访问

网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn> 棒棒堂智能运维 <http://www.51xueweb.cn>



3. 管理和监控

3.1 MongoDB Database Tools



mongostat <options> <connection-string> <polling interval in seconds>

功能:

- 根据数据库的操作类型（例如插入、查询、更新、删除等）捕获并返回服务器负载情况。

参数/命令:

- 详细信息查询官方技术手册:
 - <https://www.mongodb.com/zh-cn/docs/database-tools/mongostat/>

主要选项:

- host=<hostname><:port>:
 - 数据库服务器主机地址，可使用半角逗号隔开指定多个
- port=<port>: 数据库服务器主端口号
- username=<username>, -u=<username>:
 - 数据库连接用户名
- password=<password>, -p=<password>:
 - 数据库连接密码
- authenticationDatabase=<dbname>: 认证数据库
- uri=<connectionString>:
 - 使用连接字符串格式连接，但不可与认证字段同时使用
- humanReadable=<boolean>: 格式化输出日期等字段
- O=<field list>:
 - 指定输出字段和字段名称，可使用半角逗号隔开指定多个
- json: 以JSON格式输出信息
- <sleeptime>: 执行mongostat命令执行的间隔时间

网络与信息系统智能运维课程体系 <https://www.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

mongostat命令结果的字段说明如下。

- insert: 检测磁盘设备名称
- query: 每秒的执行查询操作数
- update: 每秒的执行更新操作数
- delete: 每秒的执行删除操作数
- getmore: 每秒获取更多（即游标批处理）操作数
- command: 每秒的执行操作数
- vsz: 进程使用的虚拟内存量（以MB为单位）
- res: 进程使用的驻留内存量（以MB为单位）
- locked: 全局写锁定中的时间百分比
- qr: 等待读数据的客户端数
- qw: 等待写数据的客户端数
- ar: 执行读数据的客户端数
- aw: 执行写数据的客户端数
- net in: 实例接收的网络流量（以字节为单位）
- net out: 实例发送的网络流量（以字节为单位）
- conn: 数据库连接数
- set: 副本集的名称
- repl: 副本集成员状态

3. 管理和监控

3.1 MongoDB Database Tools



mongotop <options> <connection-string> <polling interval in seconds>

功能:

- 根据数据库的操作类型（例如插入、查询、更新、删除等）捕获并返回服务器负载情况。

参数/命令:

- 详细信息查询官方技术手册:
 - <https://www.mongodb.com/zh-cn/docs/database-tools/mongotop/>
- 应用举例:
 - mongotop
 - 没有任何命令行选项运行 mongotop, 将连接到在本主机上运行的 27017 实例, 且每秒返回一次值。
 - mongotop 30
 - 为使 mongotop 每 30 秒报告一次, 请指定轮询间隔:

主要选项:

- host=<hostname>[:<port>]:
 - 数据库服务器主机地址, 可使用半角逗号隔开指定多个
- port=<port>: 数据库服务器主端口号
- username=<username>, -u=<username>:
 - 数据库连接用户名
- password=<password>, -p=<password>:
 - 数据库连接密码
- authenticationDatabase=<dbname>: 认证数据库
- uri=<connectionString>:
 - 使用连接字符串格式连接, 但不可与认证字段同时使用
- humanReadable=<boolean>: 格式化输出日期等字段
- O=<field list>:
 - 指定输出字段和字段名称, 可使用半角逗号隔开指定多个
- json: 以JSON格式输出信息
- <sleeptime>: 执行mongostat命令执行的间隔时间

网络与信息系统智能运维课程体系 <https://www.it-ebooks.info>

棒棒堂智能运维 <http://www.51xueweb.cn>

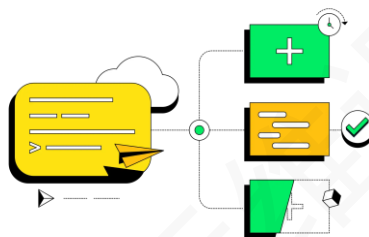
mongotop命令结果的字段说明如下。

- ns: 数据库命名空间
- db: 数据库的名称
- total: 在此命名空间上操作所消耗的总时长
- read: 在此命名空间上执行读操作所消耗的时间
- write: 在此命名空间上执行写操作所消耗的时间
- <timestamp>: 检测的有效时间

3. 管理和监控

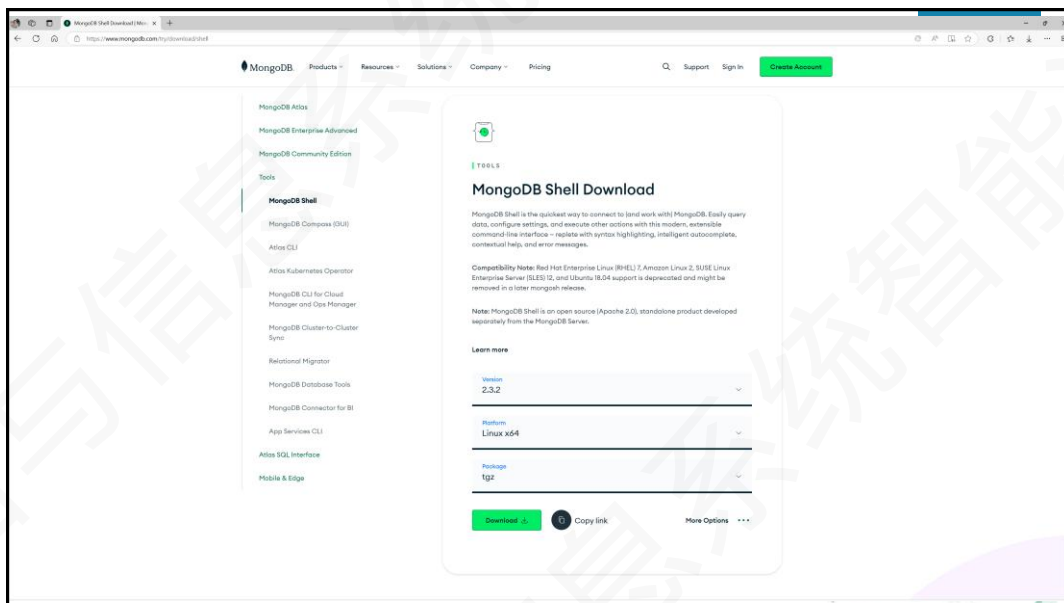
3.2 MongoDB Shell

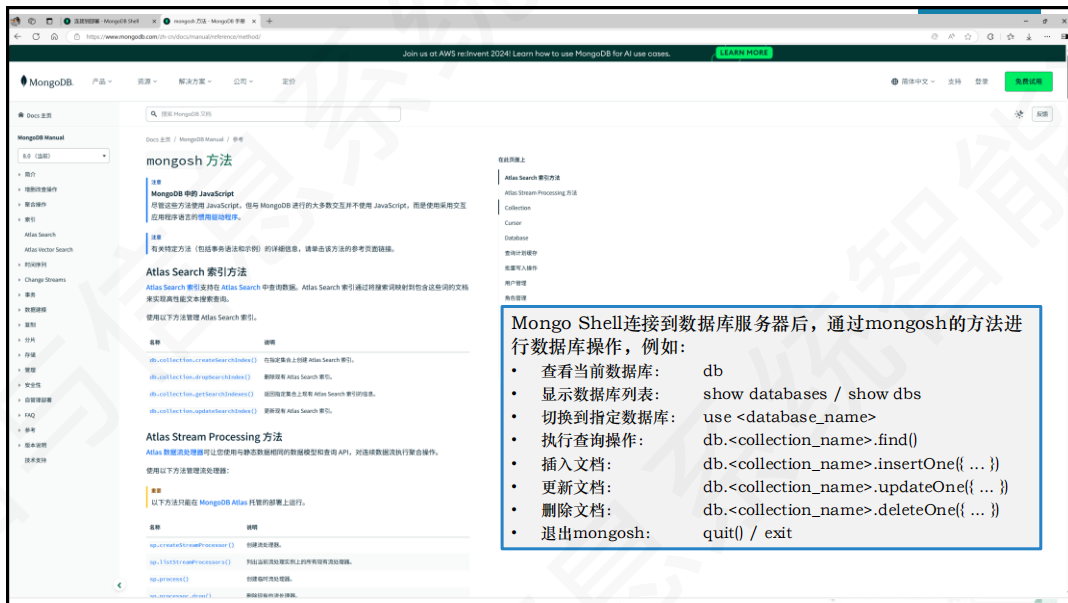
- MongoDB Shell (mongosh)
 - 是一个JavaScript 和 Node.js REPL环境。
 - 用于与Atlas、本地或其他远程主机上的MongoDB实例进行交互操作。
- 使用MongoDB Shell可以：
 - 连接到MongoDB
 - 执行增删改查和聚合操作
 - 管理MongoDB
 - 管理数据库和集合
 - 管理服务器
 - 管理用户和角色
 - 使用脚本实现自动化运维

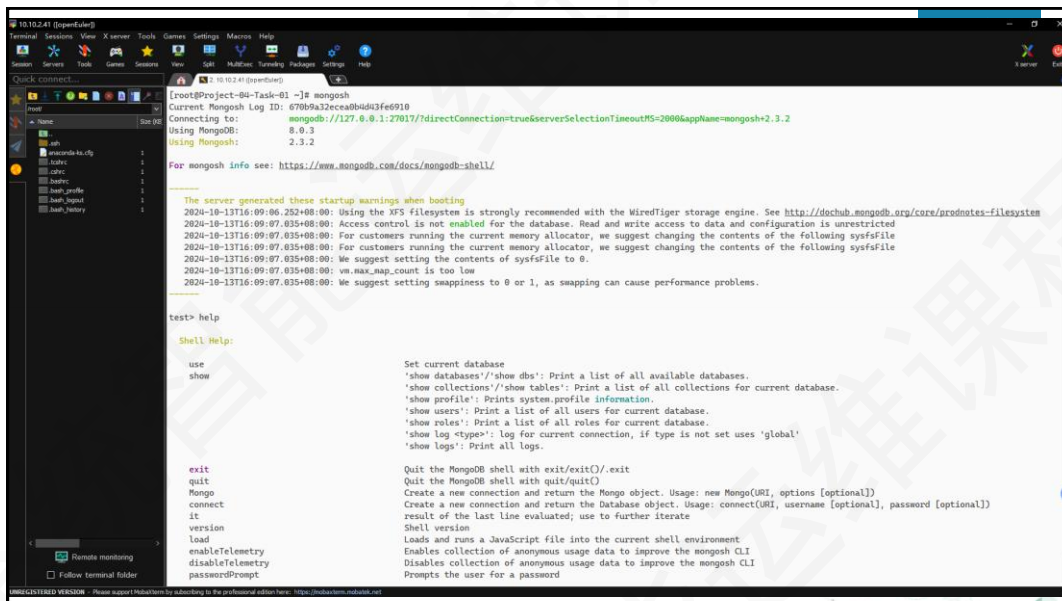


网络与信息系统智能运维课程体系 <https://intemet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>







3. 管理和监控

3.3 MongoDB Compass(GUI)

- MongoDB Compass是MongoDB官方提供的一款可视化数据库管理工具。
 - 支持多种操作系统，包括Windows、macOS和Linux。
- MongoDB Compass的主要功能：

功能	详情
数据管理	创建、查看、修改和删除数据库、数据表、视图、列、索引等
索引管理	创建、删除索引
数据聚合	创建和执行聚合管道
数据导入导出	支持从 SQL、CSV、XML 等格式文件导入数据 支持以 CSV、XML、PDF、SQL 等格式导出数据
监控	监控数据库服务器的流量、连接、进程、查询统计、数据库变量状态、主机状态等
集群管理	查看集群状态、查看集群成员、添加集群成员
统一身份验证	支持 Kerberos、LDAP 和 x.509 身份验证
文档模型分析	提供对指定集合中文档的字段和值的分析



3. 管理和监控

3.3 MongoDB Compass(GUI)

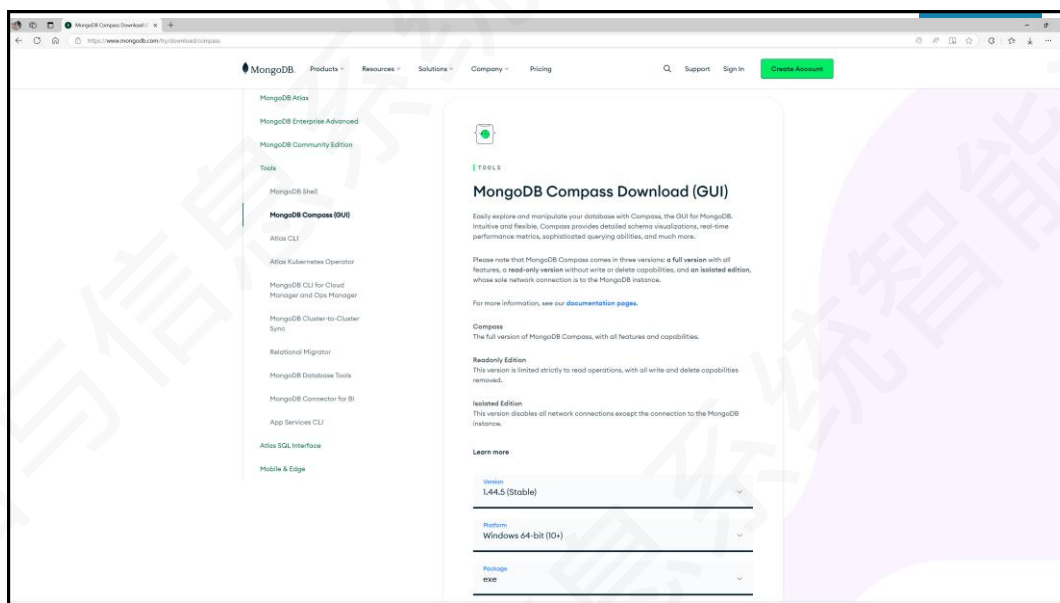
□ MongoDB Compass的版本:

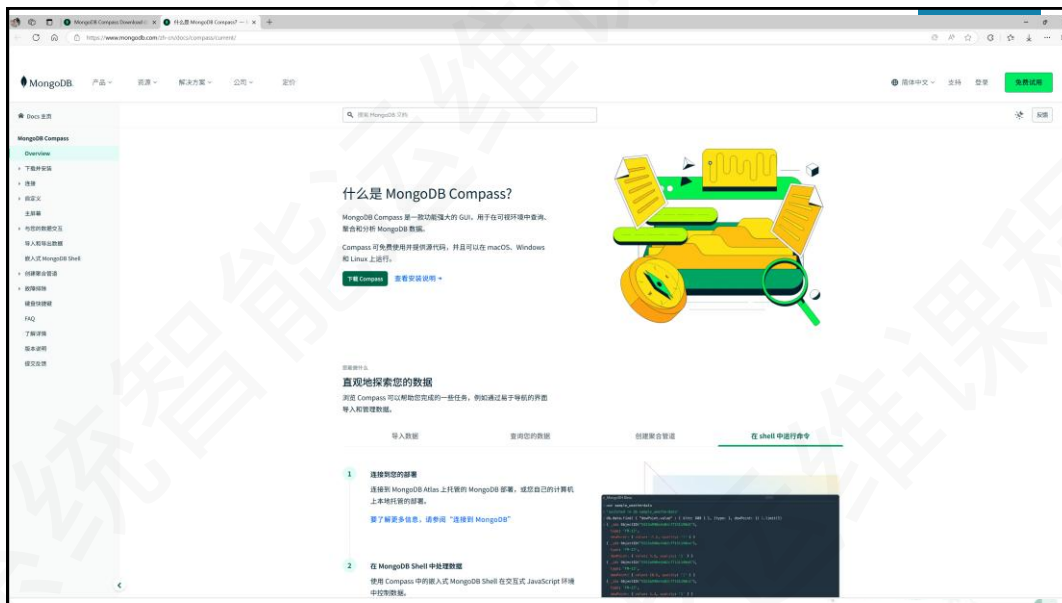
- Compass: 完整版的MongoDB Compass, 具有所有功能
- Compass Readonly: 只允许读取操作
- Compass Isolated: 除了连接MongoDB服务器外, 不发起任何网络请求

	Compass	Compass Readonly	Compass 隔离
通过完整的 CRUD 功能与文档、collection和数据库进行交互	✓		✓
创建并执行查询和聚合管道	✓	✓	✓
创建和删除索引	✓		✓
使用可视化解释计划查看和优化查询性能	✓	✓	✓
Kerberos, LDAP 和 x.509 身份验证	✓	✓	✓
模式分析	✓	✓	✓
实时服务器统计信息	✓	✓	✓
创建、删除和编辑文档验证规则	✓		✓
错误报告和数据库使用情况collection	✓	✓	
Automatic updates	✓	✓	
嵌入式 Shell 支持	✓		✓

网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>





3. 管理和监控

48

3.3 MongoDB Compass(GUI)



使用MongoDB Compass管理数据库服务器

任务目标:

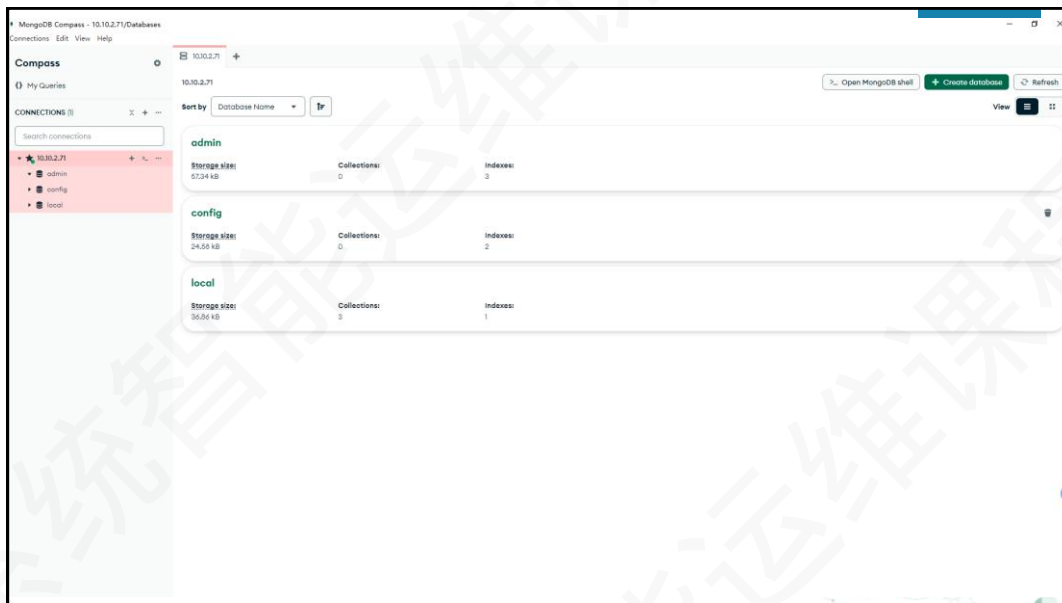
- 完成MongoDB Compass的安装, 并对MongoDB数据库服务器进行管理。

操作步骤:

- 获取MongoDB Compass
- 安装并配置
- 连接到MongoDB
- 数据管理和操作
- 监控数据库服务器

操作演示:





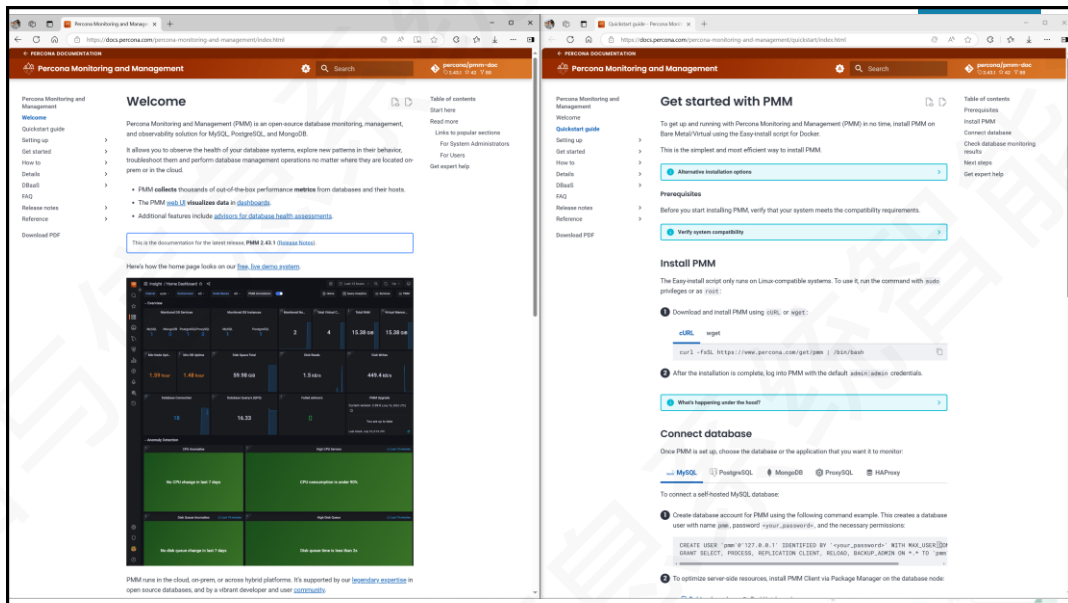
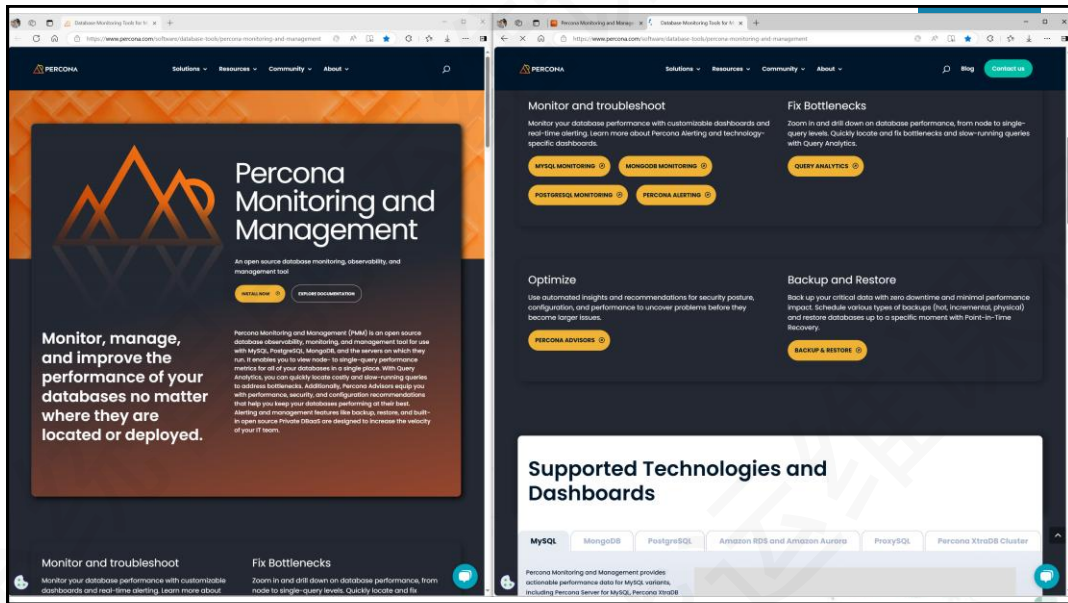
50

3. 管理和监控

3.4 Percona Monitoring and Management

- Percona Monitoring and Management（简称“PMM”）是一款开源数据库监控和管理工具软件。
 - 支持MongoDB、MySQL、MariaDB、PostgreSQL等常用数据库。
 - PMM内置监控指标和性能分析模型，并具有可视化监控分析功能。
 - PMM依据监控数据可提供性能、安全性和配置建议。
- 官方的描述：
 - An open source database monitoring, observability, and management tool.
 - Monitor, manage, and improve the performance of your databases no matter where they are located or deployed.





3. 管理和监控

3.4 Percona Monitoring and Management



使用Percona Monitoring and Management管理数据库服务器

任务目标:

- 部署PMM，并实现对MongoDB、MySQL Server的监控和运维管理。

操作步骤:

- 部署PMM
- 监控和管理MongoDB
- 监控和管理MySQL Server
- 理解监控数据
- 理解优化建议

操作演示:



网络与信息系统智能运维课程体系 <https://internet.hactcm.edu.cn>

棒棒堂智能运维 <http://www.51xueweb.cn>

网络与信息系统智能运维 课程体系学习平台

本课程体系由
河南中医药大学信息技术学院建设
课程体系学习平台由河南中医药大学医疗健康信息
工程技术研究所开发与技术保障
网络与信息系统智能运维课程体系学习平台
<https://internet.hactcm.edu.cn>
互联网运维管理工程应用丛书
<http://www.51xueweb.cn>



扫码学习
并获取课程资源

