




ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы
заодно проверяем, включена ли запись занятия

Включил Юджин запись ли ты





Утилиты MongoDB



Аристов Евгений

telegram @AEugene

<https://aristov.tech>

Правила вебинара



Активно участвуем



Задаем вопрос в чат



Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара



Цели вебинара | После занятия вы сможете

1

Настроить безопасность кластера

2

Создавать бэкапы и восстанавливаться из них

3

Настраивать мониторинг

4

Использовать дополнительные утилиты

Не `db.find()`'ом
единым



The background of the slide is a blue-tinted aerial photograph of a dense city skyline, likely New York City. Overlaid on this image is a network diagram consisting of numerous small dots connected by thin lines, creating a web-like pattern across the center of the slide. The text is centered within this network area.

Безопасность кластера

Безопасность кластера. Worst practices

Админы-новички:

- используют порты, указанные по умолчанию;
- не включают аутентификацию сразу же (самая серьёзная проблема!);
- при использовании аутентификации дают широкий доступ всем и каждому;
- не используют LDAP/аналоги для принудительной смены паролей;
- не настаивают на использовании SSL в базе данных;
- не ограничивают доступ к базе данных для известных устройств сети (хостов приложения, балансировщиков нагрузки и т.д.);

Безопасность кластера

У MongoDB есть пять основных зон безопасности:

- **Аутентификация.** Аутентификация LDAP централизует элементы в каталоге вашей компании.
- **Авторизация.** Авторизация определяет, какие права доступа предоставляет база данных в зависимости от роли пользователя.
- **Шифрование.** Шифрование можно разделить на At-Rest и In-Transit. Оно имеет решающее значение для обеспечения безопасности MongoDB.
- **Аудит.** Аудит подразумевает возможность видеть, кто что сделал в базе данных.
- **Управление.** Управление подразумевает проверку документов и конфиденциальных данных (таких как номер счета, пароль, номер социального страхования или дата рождения). Это касается как знания того, где хранятся конфиденциальные данные, так и предотвращения их ввода в систему.

Безопасность кластера

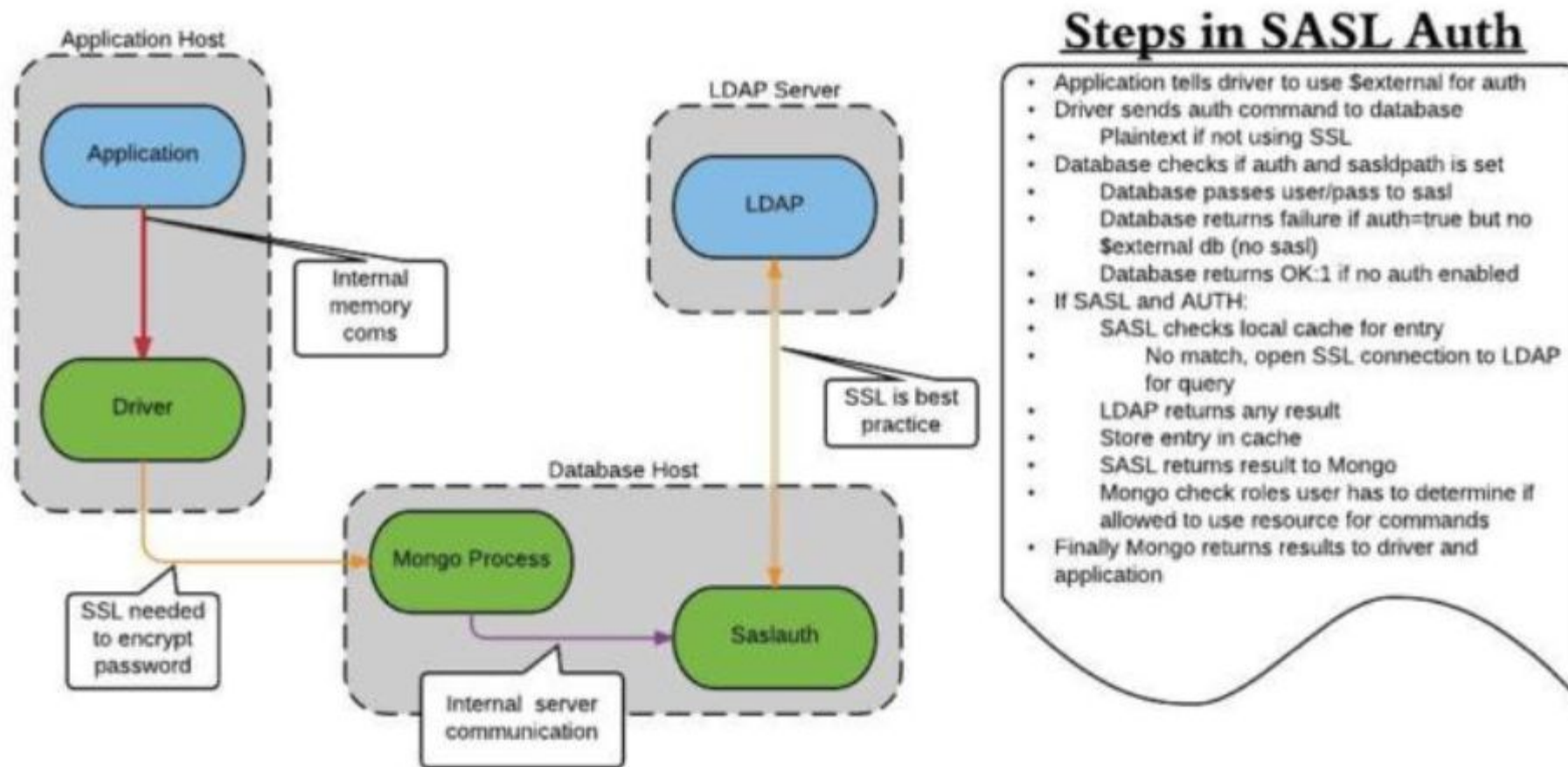


Рис. 1. Шаги в аутентификации SASL (Simple Authentication and Security Layer).

Безопасность кластера

минусы встроенных ролей:

- отсутствие политики по сложности пароля
- отсутствие ротации пароля по истечению срока действия
- отсутствие централизации и идентификации пользователей с разными наборами ролей

LDAP (AD) позволяет решить эти проблемы

поддержка LDAP доступна только в MongoDB Enterprise. Её нет в версии Community. Но она имеется в других open source версиях MongoDB, таких как Percona Server для MongoDB.

Контроль доступа на основе ролей (RBAC) лежит в основе MongoDB и заложена прямо в ядро.

Можно создать свои собственные роли, записав, какие именно действия разрешено выполнять конкретному пользователю.

<https://www.mongodb.com/blog/post/how-to-configure-LDAP-authentication-for-mongodb>

Безопасность кластера. Основные встроенные роли

- **read**
- **readWrite**
- **dbAdmin** - provides the ability to perform administrative tasks such as schema-related tasks, indexing, and gathering statistics. This role does not grant privileges for user and role management
- **userAdmin** - Provides the ability to create and modify roles and users on the current database. Since the [userAdmin](#) role allows users to grant any privilege to any user, including themselves, the role also indirectly provides [superuser](#) access to either the database or, if scoped to the admin database, the cluster
- **dbOwner** - combines the privileges granted by the [readWrite](#), [dbAdmin](#) and [userAdmin](#) roles.
- **root**
- **wildcard** (можно указать все БД и избежать роли dbAdminAnyDatabase)

Посмотрим подробнее <https://docs.mongodb.com/manual/reference/built-in-roles/>

Безопасность кластера

Backup and Restoration Roles

The admin database includes the following roles for backing up and restoring data:

backup - Provides minimal privileges needed for backing up data. This role provides sufficient privileges to use the [MongoDB Cloud Manager](#) backup agent, [Ops Manager](#) backup agent, or to use [mongodump](#) to backup an entire [mongod](#) instance.

restore - Provides the necessary privileges to restore data from backups *if* the data does not include [system.profile](#) collection data, соответственно нужны права:

- [createCollection](#)
- [createIndex](#)
- [createRole](#)
- [createUser](#)
- [etc...](#)

Безопасность кластера

Роли для кластера:

clusterAdmin - This role combines the privileges granted by the [clusterManager](#), [clusterMonitor](#), and [hostManager](#) roles. Additionally, the role provides the [dropDatabase](#) action.

clusterManager - Provides management and monitoring actions on the cluster. A user with this role can access the config and local databases (addShard, dbStats ...)

clusterMonitor - Provides read-only access to monitoring tools, such as the [MongoDB Cloud Manager](#) and [Ops Manager](#) monitoring agent. (dbStats, listIndexes...)

hostManager - Provides the ability to monitor and manage servers (shutdown, setParameter)

Безопасность кластера

All-Database Roles

- readAnyDatabase
- readWriteAnyDatabase
- userAdminAnyDatabase
- dbAdminAnyDatabase

Безопасность кластера

Виды ресурсов в MongoDB

- db
- collection (строка, например "any")
- cluster (true/false)
- anyResource (true/false)

Безопасность кластера

Виды действий в MongoDB

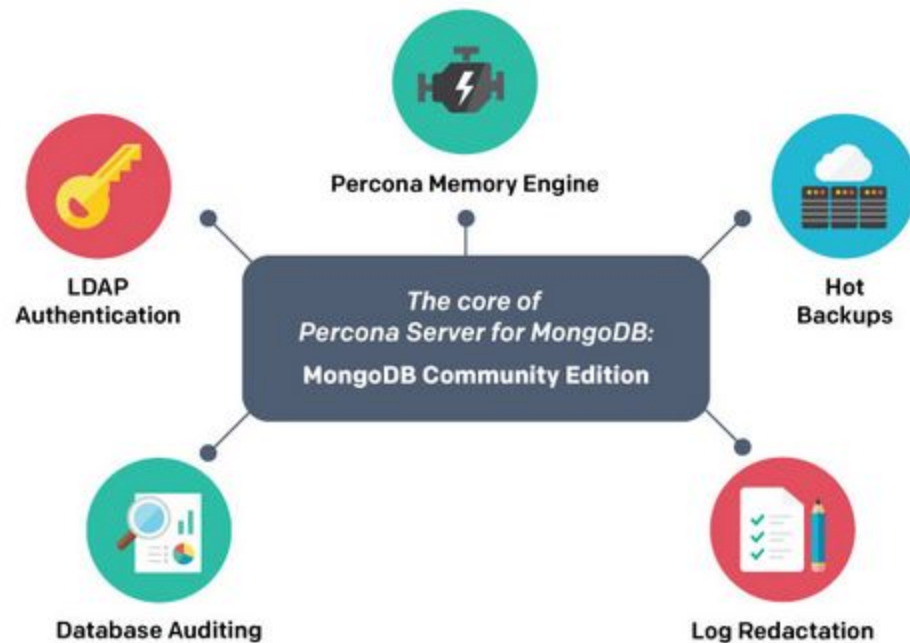
- find
- insert
- update
- remove
- addShard
- ...
- anyAction

<https://docs.mongodb.com/manual/reference/privilege-actions/>

Безопасность кластера

практика

Безопасность кластера



MongoDB Community Edition is at the core of Percona Server for MongoDB, so it automatically includes features such as native high availability, distributed transactions, a flexible data schema, and the familiarity of JSON documents.

But Percona Server for MongoDB doesn't stop there: with the Percona Memory Engine in-memory storage engine, HashiCorp Vault integration, Data-at-rest Encryption, audit logging, External LDAP Authentication with SASL, and hot backups its a complete package that maximizes performance and streamlines database efficiencies.

Безопасность кластера. SSL

Причины использовать SSL для защиты MongoDB:

- конфиденциальность
- аутентификация

Инфраструктура закрытого ключа SSL гарантирует, что только пользователи с правильным сертификатом CA могут получить доступ к MongoDB.

Ключевые функции:

- проверка корректности хоста;
- возможность указать конкретный `setup .key` для использования;
- Custom Certificate Authority (CA) для самоподписанных сертификатов и альтернативных подписчиков;
- режимы *allowSSL*, *preferSSL*, *requireSSL*, которые позволяют выбирать детализацию для вашего использования SSL (от менее безопасной до более безопасной).

<https://docs.mongodb.com/manual/tutorial/configure-ssl/>

<https://medium.com/mongoaudit/how-to-enable-tls-ssl-on-mongodb-d973a92cefa6>

Безопасность кластера. Шифрование

Данные **at-rest** — это данные, хранящиеся на диске.

Шифрование таких данных обычно подразумевает их сохранение в зашифрованное хранилище.

Зачем?

Безопасность кластера. Шифрование

Данные **at-rest** — это данные, хранящиеся на диске.

Шифрование таких данных обычно подразумевает их сохранение в зашифрованное хранилище.

Зачем?

- для предотвращения физической кражи
- для создания резервных копий, хранящихся таким образом, что их непросто прочесть третьей стороне
- если не своя инфраструктура

Безопасность кластера. Шифрование

Шифрование данных at-rest может быть реализовано одним из следующих способов или всеми сразу:

- зашифровать весь диск (LUKS, dm-crypt, etc..)
- зашифровать только файлы базы данных (встроено в ядро, выполняется прозрачным шифрованием данных TDE алгоритмом AES 256)
- зашифровать в приложении
- MongoDB 4.2 получила поддержку шифрования данных на уровне полей (field-level encryption, FLE)

<https://www.osp.ru/winitpro/2018/12/13054727/>

Безопасность кластера. Аудит

Аудит:

- возможность отслеживать, какой пользователь какие действия выполнил в базе данных
- позволяет фильтровать выходные данные конкретного пользователя, базы данных, коллекции
- создает журнал для проверки любых инцидентов в сфере безопасности
- *отсутствует в community edition*

<https://docs.mongodb.com/manual/tutorial/configure-auditing/>

<https://www.percona.com/blog/2017/03/03/mongodb-audit-log-why-and-how/>

Безопасность кластера. Управление

Управление связано отвечает за вставку и обновление данных, для проверки того, определено ли имя поля, вроде «birthday», «inn» или «creditCard». Внедряет сложные стандарты в систему MongoDB с помощью проверки документов (Document Validation new in 3.6).

Основными командами проверки документов являются:

- ***collMod***. Определяет ключ для проверки.
- ***validator***. Определяет параметры для проверки.
- ***validationLevel***. Устанавливает строгость валидатора (как часто он включается и насколько серьезные действия выполняет в зависимости от validationAction).
- ***validationAction***. Устанавливает порядок действий, когда что-либо не проходит проверку.

<https://www.mongodb.com/blog/post/document-validation-part-1-adding-just-the-right-amount-of-control-over-your-documents>

<https://docs.mongodb.com/manual/core/schema-validation/index.html>

Безопасность кластера. Управление

validationLevel can have the following values:

- “off” : validation is not applied
- “strict”: it’s the default value. Validation applies to all inserts and updates
- “**moderated**”: validation applies to all the **valid** existing documents. **Not valid** documents are ignored.

When **creating validation rules on existing collections**, the “moderated” value is the safest option.

validationAction can have the following values:

- “error”: it’s the default value. The document must pass the validation in order to be written
- “**warn**”: a document that doesn’t pass the validation is written but a warning message is logged

When adding validation rules to an existing collection the safest option is “warn”

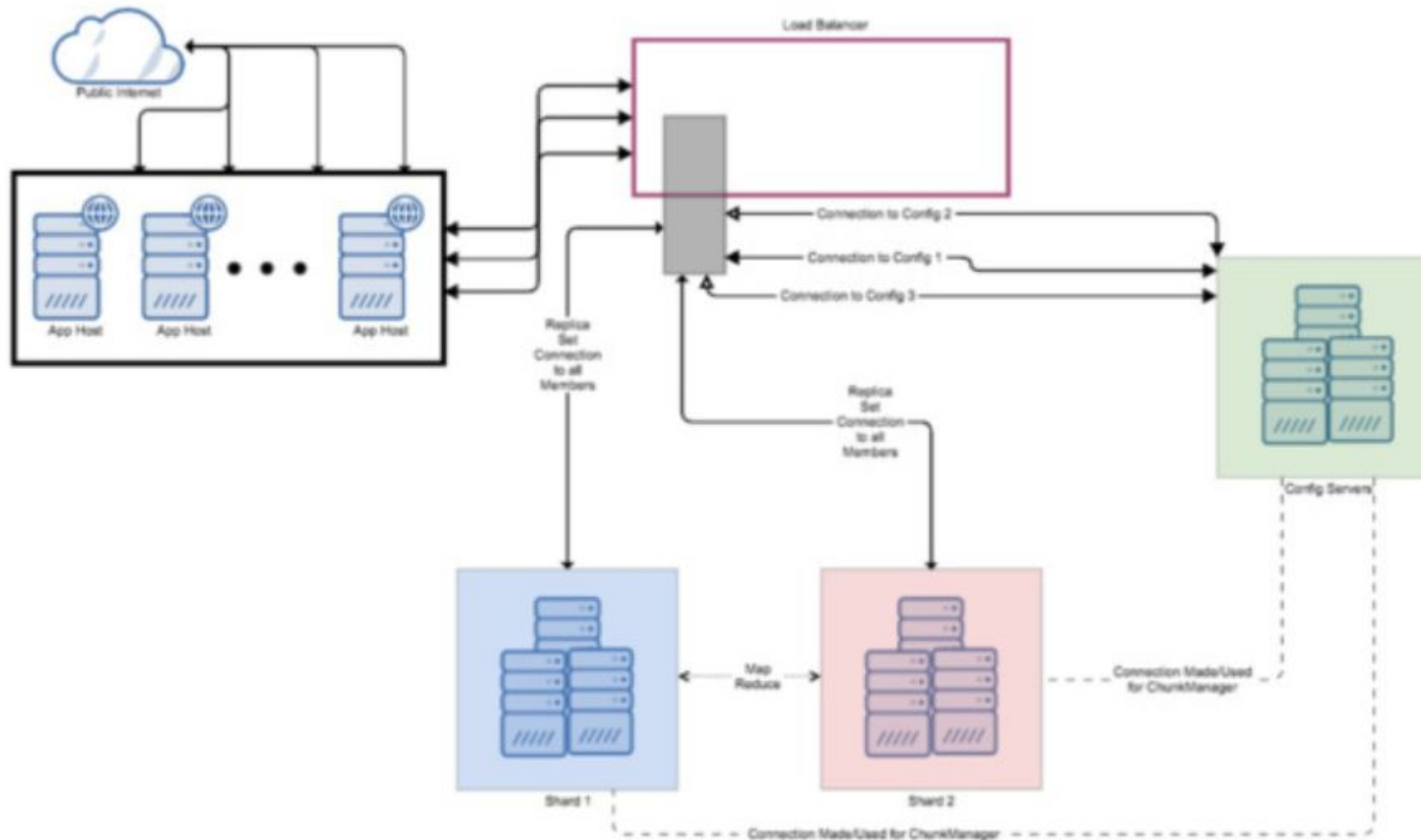
Безопасность кластера. Управление

практика по валидации документа на основе шаблона json

Безопасность кластера. Управление

В современных реалиях принято проводить проверки на бэке

Безопасность кластера. Управление



Безопасность кластера. Чеклист

- **Enable Access Control and Enforce Authentication**
- **Configure Role-Based Access Control**
- **Encrypt Communication (TLS/SSL)**
- **Encrypt and Protect Data**
- **Limit Network Exposure**
- **Audit System Activity**
- **Run MongoDB with a Dedicated User**
- **Run MongoDB with Secure Configuration Options**
- **Periodic/Ongoing Production Checks**

<https://docs.mongodb.com/manual/administration/security-checklist/>

The image features a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. A semi-transparent blue band with a white geometric network pattern of dots and lines stretches horizontally across the middle of the image. The word "Бэкап" is centered within this band in a white, bold, sans-serif font.

Бэкап

Бэкап

Варианты бэкапа

- [Back Up with Atlas](#)
- [Back Up with MongoDB Cloud Manager or Ops Manager](#)
- [Back Up by Copying Underlying Data Files](#)
- [Back Up with mongodump](#)

Бэкап

Back Up with Atlas

- [Continuous Backups](#), which take incremental backups of data in your cluster, ensuring your backups are typically just a few seconds behind the operational system. Atlas continuous backups allow you to restore from stored snapshots or from a selected point in time within the last 24 hours. You can also query a continuous backup snapshot.
- [Cloud Provider Snapshots](#), which provide localized backup storage using the native snapshot functionality of the cluster's cloud service provider.

Back Up with MongoDB Cloud Manager or Ops Manager

рекомендованный путь для шардированных кластеров

Бэкап

Back Up by Copying Underlying Data Files

- полностью файловую систему средствами ОС
- файлы командой `rsync` (остановить запись, иначе БД может стать invalid state)
- холодный и горячий

Tip

- In general, if using filesystem based backups for MongoDB Enterprise 4.2+, use the “hot” backup feature, if possible.
- For MongoDB Enterprise versions 4.0 and earlier, if you use AES256-GCM encryption mode, do **not** make copies of your data files or restore from filesystem snapshots (“hot” or “cold”).

Бэкап

[Back Up with mongodump](#)

Applications can continue to modify data while [mongodump](#) captures the output. For replica sets, [mongodump](#) provides the [--oplog](#) option to include in its output [oplog](#) entries that occur during the [mongodump](#) operation. This allows the corresponding [mongorestore](#) operation to replay the captured oplog. To restore a backup created with [--oplog](#), use [mongorestore](#) with the [--oplogReplay](#) option.

Чтобы сделать бэкап ролей - делаем бэкап бд admin

указываем реплику сет и параметр `--readPreference=secondary` для созжания бэкапа с SECONDARY

Бэкап

Варианты бесплатного бэкапа шардированного кластера
разберем на следующей лекции)

Бэкап

практика

The background of the slide is a high-angle, aerial photograph of a dense urban skyline, likely New York City, featuring numerous skyscrapers and buildings. The image is overlaid with a semi-transparent blue and green gradient. A network of thin, light blue lines connects various points across the gradient, creating a digital or technological aesthetic. In the center of the image, the word "Вопросы?" is written in a large, bold, white sans-serif font.

Вопросы?



МИНИТЕСТ

<https://forms.gle/Lfijz5LZEWBhsvdN8>

3-5 минут

The background of the image is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. The image is divided into three horizontal sections. The top and bottom sections show the city buildings. The middle section is a solid blue band with a white, glowing network pattern of lines and dots. The word "Мониторинг" is centered in this band.

Мониторинг

Мониторинг

Monitoring Strategies

MongoDB provides various methods for collecting data about the state of a running MongoDB instance:

- Starting in version 4.0, MongoDB offers [free Cloud monitoring](#) for standalones and replica sets.
- MongoDB distributes a set of utilities that provides real-time reporting of database activities.
- MongoDB provides various [database commands](#) that return statistics regarding the current database state with greater fidelity.
- [MongoDB Atlas](#) is a cloud-hosted database-as-a-service for running, monitoring, and maintaining MongoDB deployments.
- [MongoDB Cloud Manager](#) is a hosted service that monitors running MongoDB deployments to collect data and provide visualization and alerts based on that data.
- MongoDB Ops Manager is [an on-premise solution available in MongoDB Enterprise Advanced](#) that monitors running MongoDB deployments to collect data and provide visualization and alerts based on that data.

Мониторинг

Starting in version 4.0, MongoDB (Community Edition) offers free Cloud monitoring for standalones and replica sets.

By default, you can enable/disable free monitoring during runtime using [db.enableFreeMonitoring\(\)](#) and [db.disableFreeMonitoring\(\)](#).

You can also enable or disable free monitoring during [mongod](#) startup using either:

- the configuration file setting [cloud.monitoring.free.state](#) or
- the command-line option [--enableFreeMonitoring](#)

Free monitoring provides information about your deployment, including:

- Operation Execution Times
- Memory Usage
- CPU Usage
- Operation Counts

The data expires 24 hours after being uploaded

Мониторинг

Утилиты для мониторинга:

- **mongostat** — инструмент командной строки, который суммирует список статистических данных для исполняемого экземпляра MongoDB, это позволяет визуализировать количество вставок, обновлений, удалений, запросов и команд, а также ресурсопотребление экземпляра;

```
$ mongostat --port 27001
```

<https://docs.mongodb.com/manual/reference/program/mongostat/>

Мониторинг

inserts The number of objects inserted into the database per second. If followed by an asterisk (e.g. *), the datum refers to a replicated operation.

query The number of query operations per second.

update The number of update operations per second.

delete The number of delete operations per second.

getmore The number of get more (i.e. cursor batch) operations per second.

command The number of commands per second. On [secondary](#) systems, [mongostat](#) presents two values separated by a pipe character (e.g. |), in the form of local|replicated commands.

flushes For the [WiredTiger Storage Engine](#), flushes refers to the number of WiredTiger checkpoints triggered between each polling interval.

dirty Only for [WiredTiger Storage Engine](#). The percentage of the WiredTiger cache with dirty bytes, calculated by [wiredTiger.cache.tracked dirty bytes in the cache](#) / [wiredTiger.cache.maximum bytes configured](#).

used Only for [WiredTiger Storage Engine](#). The percentage of the WiredTiger cache that is in use, calculated by [wiredTiger.cache.bytes currently in the cache](#) / [wiredTiger.cache.maximum bytes configured](#).

Мониторинг

vsize The amount of virtual memory in megabytes used by the process at the time of the last [mongostat](#) call.

res The amount of resident memory in megabytes used by the process at the time of the last [mongostat](#) call.

locked Changed in version 3.0: Only appears when [mongostat](#) runs against pre-3.0 versions of MongoDB instances. The percent of time in a global write lock.

qr The length of the queue of clients waiting to read data from the MongoDB instance.

qw The length of the queue of clients waiting to write data from the MongoDB instance.

ar The number of active clients performing read operations.

aw The number of active clients performing write operations.

netIn The amount of network traffic, in *bytes*, received by the MongoDB instance. This includes traffic from [mongostat](#) itself.

netOut The amount of network traffic, in *bytes*, sent by the MongoDB instance. This includes traffic from [mongostat](#) itself.

conn The total number of open connections.

set The name, if applicable, of the replica set.

repl The replication status of the member.

Мониторинг

Утилиты для мониторинга:

- **mongotop** — инструмент, который предоставляет метод для отслеживания времени, которое считывает или записывает данные в экземпляре. Он также обеспечивает статистику на уровне каждой коллекции;

<https://docs.mongodb.com/manual/reference/program/mongotop/>

HTTP Console

Changed in version 3.6: MongoDB 3.6 removes the deprecated HTTP interface and REST API to MongoDB.

Мониторинг

Команды:

[db.serverStatus\(\)](#)

[текущее количество подключений к MongoDB](#)

[db.stats\(\)](#)

[db.collection.stats\(\)](#)

[rs.status\(\)](#)

[sh.status\(\)](#)

By default, the **Storage Node Watchdog** is disabled. You can only enable the Storage Node Watchdog on a [mongod](#) at startup time by setting the [watchdogPeriodSeconds](#) parameter to an integer greater than or equal to 60. However, once enabled, you can pause the Storage Node Watchdog and restart during runtime. See [watchdogPeriodSeconds](#) parameter for details.

If any of the filesystems containing the monitored directories become unresponsive, the Storage Node Watchdog terminates the [mongod](#) and exits with a status code of 61. If the [mongod](#) is the [primary](#) of a replica set, the termination initiates a [failover](#), allowing another member to become primary.

Мониторинг

[MongoDB Cloud Manager](#)

MongoDB Cloud Manager is a cloud-based suite of services for managing MongoDB deployments. MongoDB Cloud Manager provides monitoring, backup, and automation functionality. For an on-premise solution, see also [Ops Manager, available in MongoDB Enterprise Advanced](#).

[VividCortex](#)

VividCortex provides deep insights into MongoDB [production database workload and query performance](#) – in one-second resolution. Track latency, throughput, errors, and more to ensure scalability and exceptional performance of your application on MongoDB.

[Scout](#)

Several plugins, including [MongoDB Monitoring](#), [MongoDB Slow Queries](#), and [MongoDB Replica Set Monitoring](#).

[Server Density](#)

[Dashboard for MongoDB](#), MongoDB specific alerts, replication failover timeline and iPhone, iPad and Android mobile apps.

[Application Performance Management](#)

IBM has an Application Performance Management SaaS offering that includes monitor for MongoDB and other applications and middleware.

[New Relic](#)

New Relic offers full support for application performance management. In addition, New Relic Plugins and Insights enable you to view monitoring metrics from Cloud Manager in New Relic.

[Datadog](#)

[Infrastructure monitoring](#) to visualize the performance of your MongoDB deployments.

[SPM Performance Monitoring](#)

[Monitoring, Anomaly Detection and Alerting](#) SPM monitors all key MongoDB metrics together with infrastructure incl. Docker and other application metrics, e.g. Node.js, Java, NGINX, Apache, HAProxy or Elasticsearch. SPM provides correlation of metrics and logs.



Настройки

Настройки

mongodb.conf

Where to store the data.

dbpath=/var/lib/mongodb

#where to log

logpath=/var/log/mongodb/mongodb.log

logappend=true

bind_ip = 127.0.0.1

#port = 27017

..

mongod --config /etc/mongod.conf

mongos --config /etc/mongos.conf

<https://docs.mongodb.com/manual/reference/configuration-options/>

Настройки

Утечка 800 млн email через СУБД MongoDB сервиса подтверждения адресов

The background of the slide is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this image is a semi-transparent blue band across the middle, which contains a white network pattern of dots and lines. The word "Вопросы?" is centered within this band in a large, white, sans-serif font.

Вопросы?

The background of the entire image is an aerial photograph of a dense urban skyline, likely New York City, featuring numerous skyscrapers. The image is overlaid with a semi-transparent blue and green gradient. A network of thin, light blue lines connects various points across the central area, creating a digital or data network aesthetic. The word "УТИЛИТЫ" is centered in this network area.

УТИЛИТЫ

УТИЛИТЫ

mongoperf - проверка производительности I/O файловой системы (3.6)

<https://docs.mongodb.com/v3.6/reference/program/mongoperf/>

```
$ echo "{nThreads:32,fileSizeMB:1000,r:true}" | mongoperf
mongoperf
use -h for help
parsed options:
{ nThreads: 32, fileSizeMB: 1000, r: true }
creating test file size:1000MB ...
testing...
options:{ nThreads: 32, fileSizeMB: 1000, r: true }
wthr 32
new thread, total running : 1
read:1 write:0
4759 ops/sec 18 MB/sec
4752 ops/sec 18 MB/sec
4760 ops/sec 18 MB/sec
4758 ops/sec 18 MB/sec
4752 ops/sec 18 MB/sec
4754 ops/sec 18 MB/sec
4758 ops/sec 18 MB/sec
```

УТИЛИТЫ

mongoreplay — инструмент, обеспечивающий перехват, сохранение и последующее воспроизведение команд, посылаемых в MongoDB;

```
sudo mongoreplay record -i=lo -e="port 27001" -p=/home/mongo/playback
```

```
sudo mongoreplay play --host mongoddb://127.0.0.1:27001 -p /home/mongo/playback  
--report /home/mongo/report --collect json
```

<https://docs.mongodb.com/v3.6/reference/program/mongoreplay/>

УТИЛИТЫ

mongoexport

```
sudo mongoexport --port 27001 -d test -c people3 -o export.json
```

```
sudo mongoexport --port 27001 -d test -c people3 -f "name,surname" -o export.json
```

```
sudo mongoexport --port 27001 -d test -c people3 -f "name,surname" -q  
'{year_of_birth:{$gt:1963}}' -o export.json
```

<https://docs.mongodb.com/v3.6/reference/program/mongoexport/>

mongoimport

```
mongoimport --port 27001 -d test -c people5 --file export.json
```

```
mongoimport --port 27001 -d test -c people5 --file export.json --upsert
```

<https://docs.mongodb.com/v3.6/reference/program/mongoimport/>

The background of the image is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this is a semi-transparent blue band that contains a white network or mesh pattern of interconnected lines and dots. Centered within this band is the text "Percona MongoDB" in a large, white, sans-serif font.

Percona MongoDB

Percona MongoDB

Установка

<https://www.percona.com/software/mongodb/percona-server-for-mongodb>

Доки

<https://www.percona.com/doc/percona-server-for-mongodb/LATEST/index.html>

The background of the entire image is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue overlay covers the entire image. In the center, there is a network pattern of white lines connecting various points, resembling a web or data network. The text 'FAQ' is centered within this network pattern.

FAQ

FAQ


<https://docs.mongodb.com/manual/faq/diagnostics/>

The background of the slide features an aerial view of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that contains a network of white lines and dots, resembling a data or communication network. The text "Java Spring ORM" is centered in the middle of the slide in a large, white, sans-serif font.

Java Spring ORM

Java Spring ORM

<https://github.com/aeuge/otus-spring-library>

The image features a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. A semi-transparent blue band with a white network pattern of dots and lines runs horizontally across the center of the image. The Cyrillic text 'ДЗ' is centered within this band.

ДЗ

ДЗ


нет его

The image features a blue-tinted aerial view of a dense city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white network pattern of dots and lines runs horizontally across the middle of the image. The Russian word "Порефлексируем" is written in white, bold, sans-serif font across this band.


Порефлексируем

Вопросы?

- Кто что запомнил за сегодня?
- Какие вы помните варианты аутентификации?

The background of the image is an aerial view of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a white network pattern of interconnected dots and lines. The text is centered within this blue layer.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате
<https://otus.ru/polls/34473/>



Спасибо за внимание!
Приходите на следующие вебинары

Аристов Евгений