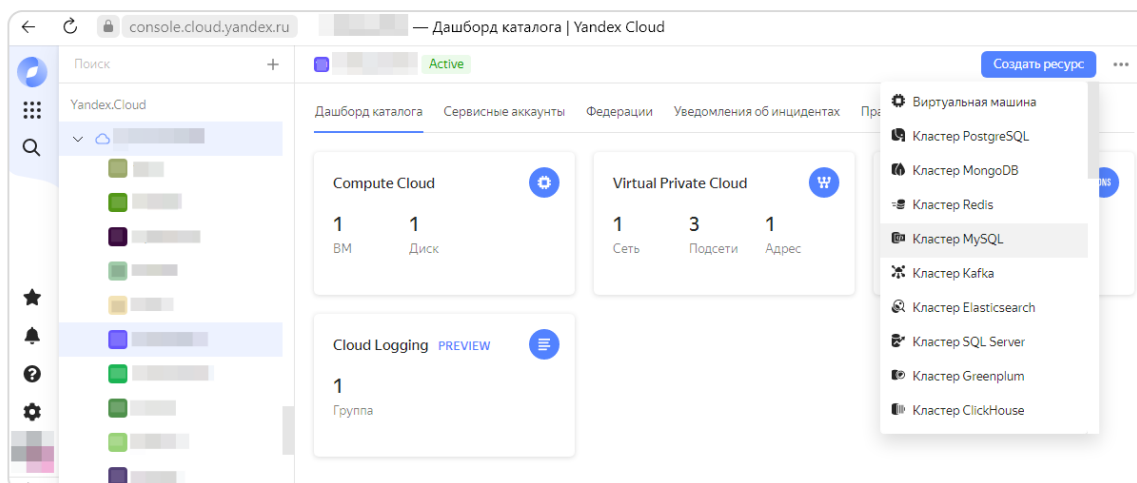


Практическая работа. Создание кластера базы данных MySQL

Object Storage — удобный и полезный инструмент для хранения данных в облаке. Но для решения практических задач важно не просто хранить данные, но и иметь возможность их изменять и выполнять с ними различные операции (сортировать, группировать, делать выборки и так далее). Для этого используются базы данных. В этой и следующих темах вы научитесь работать с несколькими управляемыми БД. И начнем мы с одной из самых популярных — MySQL.

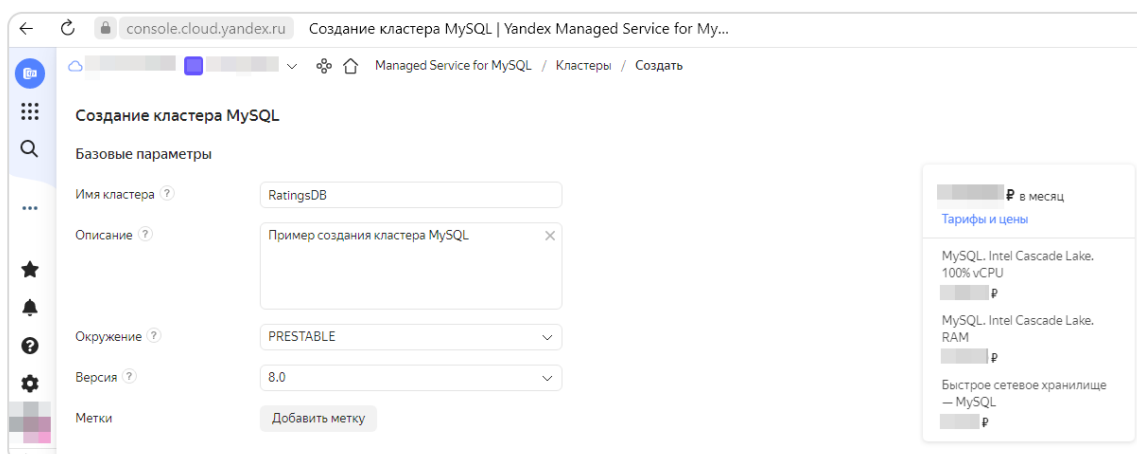
На этом уроке вы создадите и настроите кластер управляемой БД MySQL, подключитесь к нему, перенесёте данные в облако, познакомитесь с возможностями резервного копирования и мониторинга. Эти навыки пригодятся вам и в других сервисах управляемых БД, поскольку принципы работы в них очень похожи. Предположим, вы решили добавить в разрабатываемый вами мессенджер новую функциональность. Вы написали микросервис, который позволяет оценивать сообщения в групповых чатах и хранит оценки в БД MySQL. Давайте поместим эту БД в Yandex Cloud. Прежде всего понадобится создать кластер: набор виртуальных машин (ВМ, или хостов), на которых будет развёрнута БД. Это обязательный первый шаг при использовании любого сервиса управляемых БД.

Войдите в консоль управления Yandex Cloud и выберите каталог для кластера. Вверху справа нажмите кнопку **Создать ресурс** и выберите из выпадающего списка **Кластер MySQL**.



Откроется страница с основными настройками кластера. Рассмотрим их подробнее.

Базовые параметры



Имя кластера может включать только цифры, прописные и строчные латинские буквы, дефисы.

Поле **Описание** заполнять необязательно. Оно полезно, если вам нужно создать несколько кластеров для разных целей, чтобы в них было проще ориентироваться.

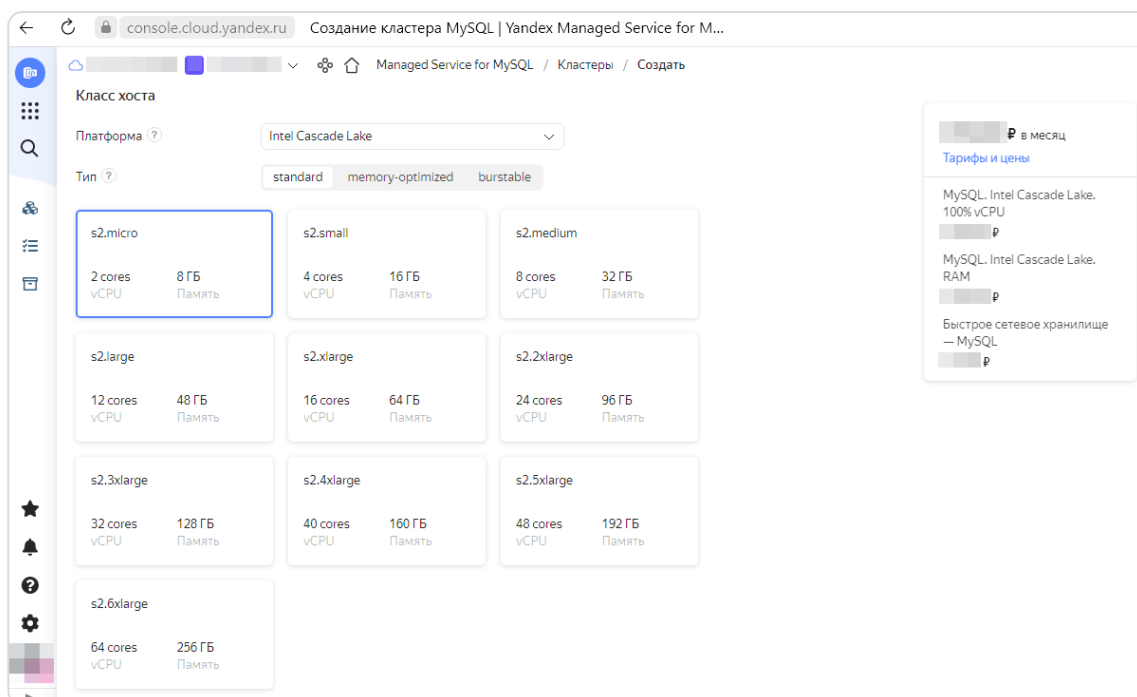
О том, какое бывает **Окружение** кластера и чем различаются PRESTABLE и PRODUCTION, мы говорили на одном из предыдущих [уроков](#).

Поскольку микросервис только разрабатывается, выберите окружение PRESTABLE.

Версия. В качестве сервера MySQL в Yandex Cloud используется Percona Server версии 5.7 или 8.0. У этих реализаций сервера улучшенная производительность на многоядерных машинах. Если для вас критична стабильность работы микросервиса, выбирайте проверенную временем 5.7. Для нашей задачи подойдёт 8.0: в ней много новых функций, но она ещё не полностью обкатана.

Класс хостов

Следующий шаг — выбор [класса хостов](#), или шаблона VM. Хосты кластера будут развёрнуты на базе VM Compute Cloud с использованием этого шаблона.



Платформа определяет тип физического процессора (Intel Broadwell или Intel Cascade Lake), а также конфигурации числа ядер виртуального процессора (vCPU) и размера оперативной памяти. Если тип процессора для вас неважен, выбирайте более современную платформу Intel Cascade Lake. Она предоставляет широкий выбор конфигураций вычислительных ресурсов.

Также на конфигурации влияет **тип BM**, на которой будет развёрнута БД.

Standard — это обычные VM с 4 ГБ RAM на ядро vCPU. Это оптимальный баланс между количеством запущенных процессов, быстродействием и потребляемой оперативной памятью.

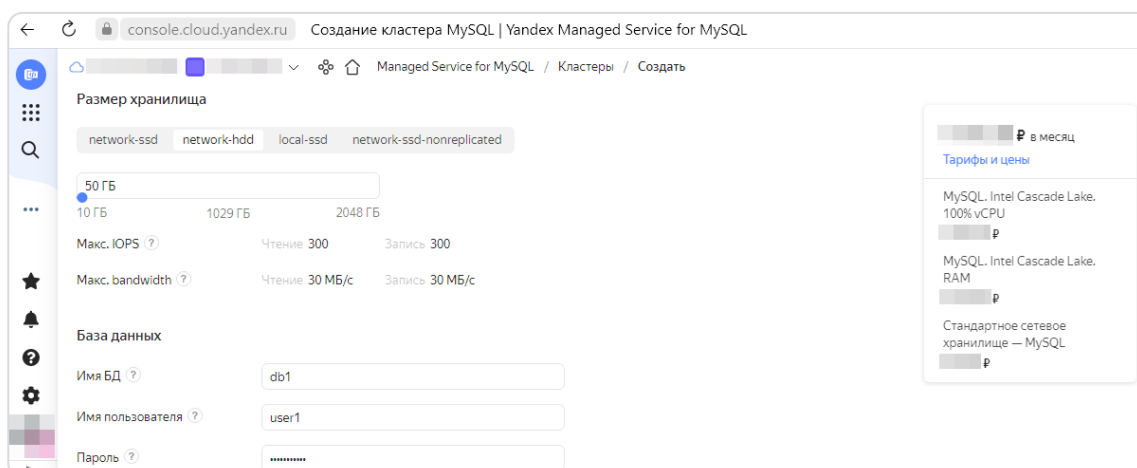
Memory-optimized — машины с вдвое увеличенным объёмом RAM на каждое ядро. Выбирайте их для высоконагруженных сервисов с повышенными требованиями к кешу.

Burstable — машины, для которых гарантируется использование лишь доли ядра vCPU (5, 20 или 50%) с вероятностью временного повышения вплоть до 100%. Они стоят дешевле и подходят для задач, где не нужен постоянный уровень производительности, т. е. для тестирования или разработки.

Выберем для микросервиса следующий класс хоста: платформа — Intel Cascade Lake; тип — standard; конфигурация вычислительных ресурсов — s2.micro (два ядра vCPU, 8 ГБ RAM).

Хранилище данных

Хранилище БД может быть сетевым или локальным. В первом случае данные находятся на виртуальных дисках в инфраструктуре Yandex Cloud. Локальное хранилище — это диски, которые физически размещаются в серверах хостов БД.



При создании кластера можно выбирать между следующими типами хранилища:

- Стандартное сетевое (**network-hdd**) — это наиболее экономичный вариант. Выбирайте его, если к скорости записи и чтения нет особых требований.
- Быстрое сетевое (**network-ssd**) стоит примерно в четыре раза дороже, но при размере хранилища от 100 ГБ работает быстрее стандартного в десять и более раз (чем больше размер, тем заметнее разница в скорости).
- Сетевое на нереплицируемых SSD-дисках (**network-ssd-nonreplicated**) — использует сетевые SSD-диски с повышенной производительностью, реализованной за счет устранения избыточности. Объем такого хранилища можно увеличивать только с шагом 93 ГБ.
- Быстрое локальное (**local-ssd**) — самое быстрое и дорогое. Если локальный диск откажет, все сохранённые на нём данные будут потеряны. Чтобы этого избежать, при выборе локального хранилища сервис автоматически создаст отказоустойчивый кластер минимум из трёх хостов.

При создании кластера внимательно выбирайте тип хранилища.

Размер хранилища можно будет позже изменить, а тип — нет.

Выберите для кластера стандартное сетевое хранилище `network-hdd` размером 50 ГБ.

База данных

В этом разделе настроек задаются атрибуты базы: **Имя БД**, уникальное в рамках кластера, **Имя пользователя** (владельца БД) и **Пароль** пользователя.

console.cloud.yandex.ru Создание кластера MySQL | Yandex Managed Service for MySQL

Managed Service for MySQL / Кластеры / Создать

База данных

Имя БД ? db1

Имя пользователя ? user1

Пароль ?

Сетевые настройки

Сеть ? intro2

Группы безопасности ? default-sg-enp45gigtd6e44dn1fj (default)

Тарифы и цены

MySQL. Intel Cascade Lake. 100% vCPU

MySQL. Intel Cascade Lake. RAM

Стандартное сетевое хранилище — MySQL

Сеть

Здесь можно выбрать облачную сеть для кластера и группы безопасности для его сетевого трафика.

Оставьте сеть по умолчанию (`default`) или выберите сеть, которую создали на предыдущем курсе. Кластер будет доступен для всех VM, которые подключены к вашей облачной сети.

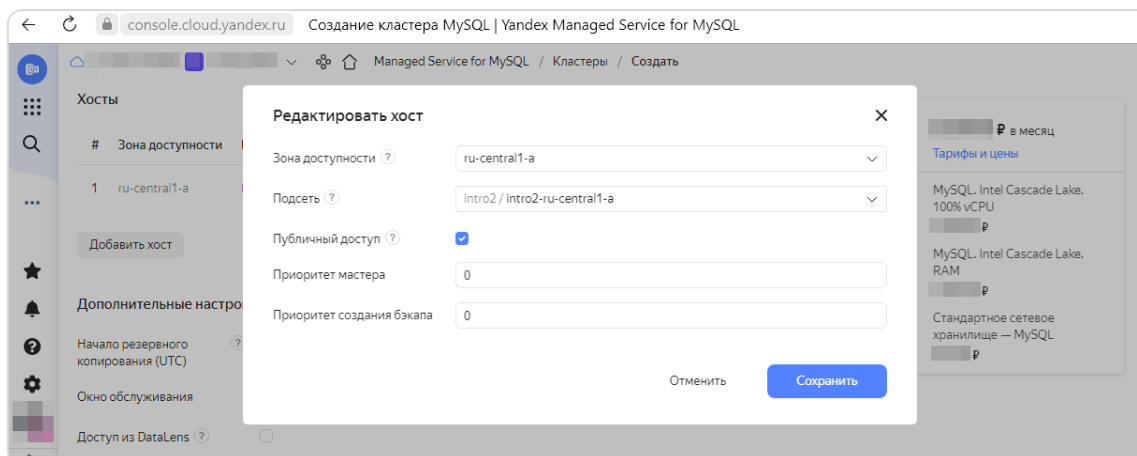
Параметры хостов

В этом блоке можно добавить количество хостов, которые будут созданы вместе с кластером, и изменить их параметры.

Дополнительные хосты могут понадобиться, например, для репликации БД или снижения нагрузки на хост-мастер.

Для наших целей достаточно кластера из одного хоста. Нажмите значок редактирования параметров хоста и в открывшемся окне выберите опцию **Публичный доступ**. Это означает, что к хосту можно будет подключиться из интернета, а не только из облачной сети.

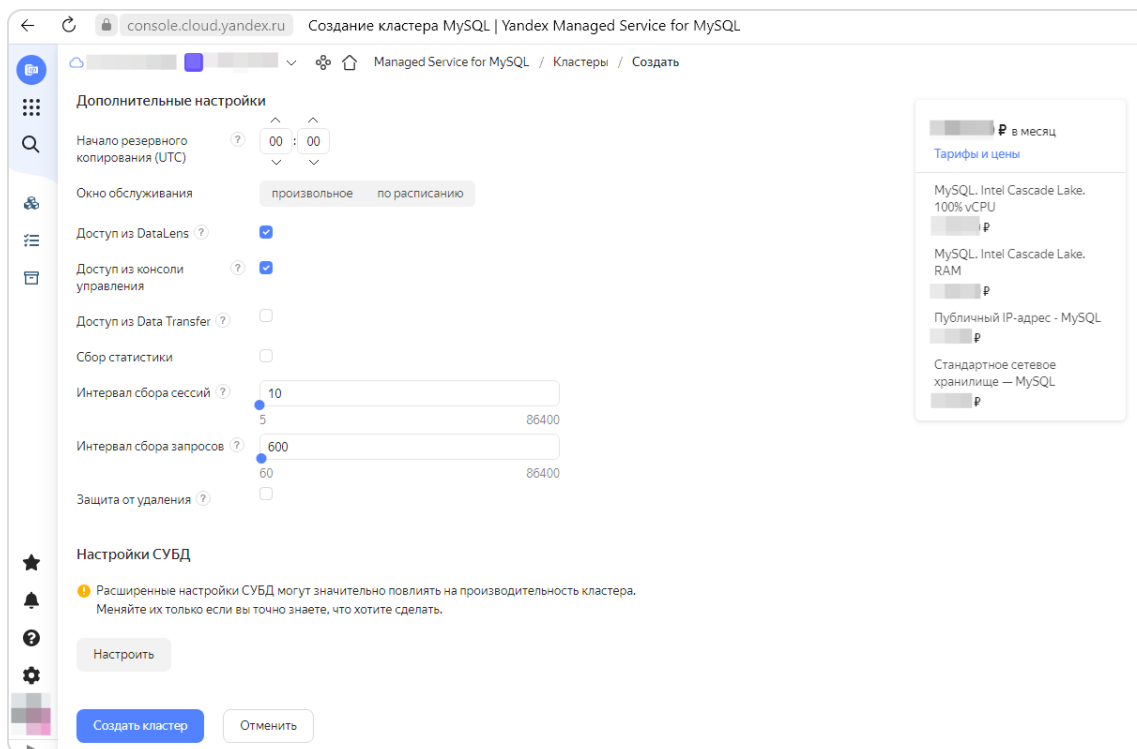
Остальные параметры оставьте без изменений.



Дополнительные настройки

Здесь можно:

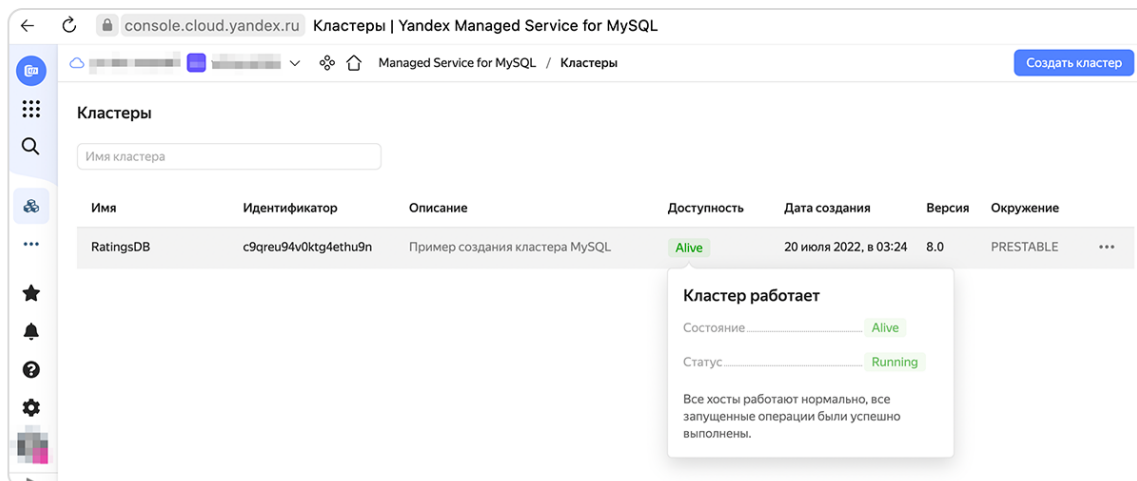
- указать время **Начала резервного копирования** и **Окна обслуживания**. Это пригодится, если вы хотите, чтобы резервное копирование и техобслуживание хостов кластера не совпадали с периодами пиковых нагрузок на БД;
- разрешить **Доступ из DataLens**, если вы планируете анализировать в DataLens данные из базы. Подробнее о DataLens вы узнаете на одном из следующих занятий;
- разрешить **Доступ из консоли управления**, чтобы выполнять SQL-запросы к БД из консоли управления Yandex Cloud. Отметьте этот пункт: доступ из консоли понадобится нам на следующих практических работах;
- разрешить **Доступ из Data Transfer**, чтобы разрешить доступ к кластеру из сервиса [Yandex Data Transfer](#) в Serverless-режиме;
- разрешить **Сбор статистики**, чтобы воспользоваться инструментом [Диагностика производительности](#) в кластере;
- установить **Защиту от удаления**, чтобы защитить кластер от непреднамеренного удаления пользователем.



В этом блоке также можно задать настройки БД (например используемую сервером MySQL кодировку при работе с данными и обмене информацией с клиентами). По умолчанию при создании кластера сервис выбирает оптимальные настройки. Изменяйте их, если уверены, что это необходимо.

Настройка завершена. Осталось только нажать кнопку **Создать кластер**.

Создание кластера займёт несколько минут. Когда он будет готов к работе, его статус на панели Managed Service for MySQL сменится с **Creating** на **Running**, а состояние — на **Alive**.



Статус показывает, что происходит с кластером:

- **Creating** — создаётся;
- **Running** — работает;
- **Error** — не отвечает, возникла проблема;
- **Updating** — обновляется;
- **Stopped** — остановлен;
- **Unknown** — статус неизвестен (так может быть, например, когда кластер не виден из интернета).

Состояние — это показатель доступности кластера:

- **Alive** — все хосты кластера работают;
- **Degraded** — часть хостов (один или больше) не работает;
- **Dead** — все хосты не работают.

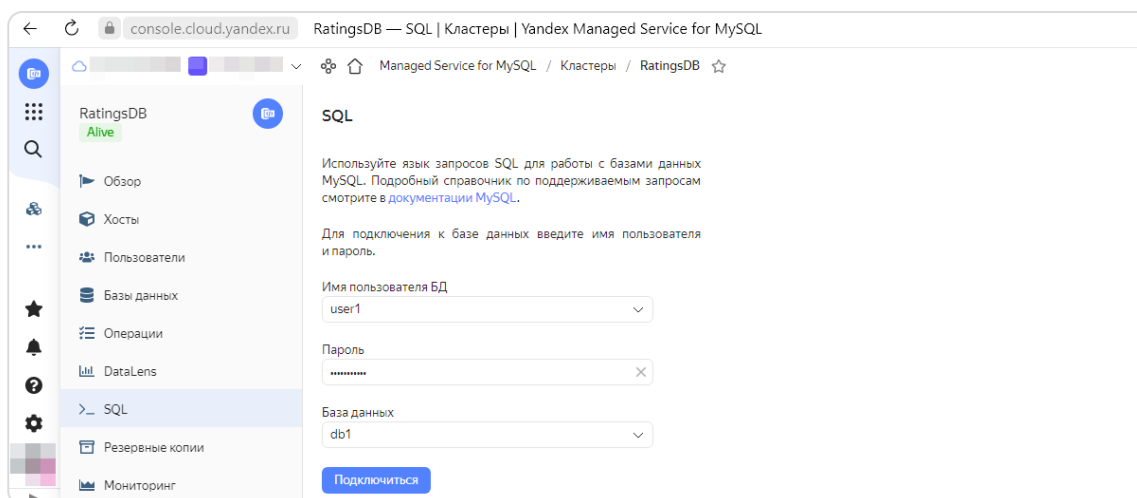
Практическая работа. Подключение к БД и добавление данных

Доступ из консоли управления

Продолжим практическую работу. В кластере, который вы создали, уже есть БД. Она пока пустая. Поскольку при создании кластера вы выбрали в настройках пункт **Доступ из консоли управления**, в консоли управления Yandex Cloud появилась вкладка с интерфейсом для выполнения SQL-запросов к БД.

Давайте зайдём туда и создадим в БД таблицу для нашего микросервиса.

На странице **Managed Service for MySQL** выберите строку с созданным вами кластером. В панели консоли управления перейдите на вкладку **SQL**. Вам будет предложено выбрать БД для SQL-запросов и имя пользователя, а также ввести пароль. Все эти атрибуты вы задавали при создании кластера.



Нажмите кнопку **Подключиться**. Откроется структура БД (сейчас там написано, что данных нет) и окно ввода для SQL-запросов.

Теперь создадим таблицу. Введите в окне ввода следующий запрос и нажмите кнопку **Выполнить**.

Скопировать код

```
CREATE TABLE IF NOT EXISTS ratings (  
    rating_id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,
```

```
message_id INT NOT NULL,  
rating INT NOT NULL  
) ENGINE=INNODB;
```

Обратите внимание, что в качестве движка в сервисе управляемых БД MySQL используется только InnoDB.

В таблицу можно добавить данные с помощью команды INSERT.

Скопировать код

```
INSERT INTO ratings (user_id,message_id,rating) VALUES (44,368,4);
```

Чтобы отобразить обновлённую структуру БД, нажмите на имя БД и выберите таблицу `ratings`.

The screenshot shows the Yandex Cloud console interface for a MySQL database named 'RatingsDB'. The left sidebar contains navigation links: Обзор, Хосты, Пользователи, Базы данных, Операции, DataLens, SQL (selected), and Резервные копии. The main panel displays the 'ratings' table with the following data:

#	rating_id	user_id	message_id	rating
0	1	44	368	4
1	2	44	368	4
2	3	100	944	10

Below the table, there is a SQL query editor with the following query:

```
1 INSERT INTO ratings (user_id,message_id,rating)  
VALUES (100,944,10);
```

The query has been executed, and the result is shown in a table below:

#	result
0	1

Наведите указатель на заголовок столбца, чтобы увидеть тип данных в нём.

SQL-запросы через консоль управления Yandex Cloud — нетипичный способ работы с БД. Используйте его для небольших, разовых задач, когда быстрее и проще открыть подключение в браузере. Этот способ не очень удобен: текст запроса и результат его выполнения доступны, только пока вы не закрыли или не перезагрузили страницу в браузере. Конечно, если запрос успешно запущен, то сервис обработает его независимо от состояния консоли управления.

В консоли выводятся только первые 1000 строк результата запроса, даже если данных больше. Чтобы увидеть строку, введите её номер в поле **Номер первой строки**.

Подключение к кластеру

В основном вы будете работать с БД из приложений или из командной строки. Однако для этого нужно подключиться к хосту, на котором развёрнута БД.

Есть два варианта подключения. Если публичный доступ к хосту открыт, подключитесь к нему через интернет с помощью защищённого SSL-соединения. Если публичного доступа нет, подключитесь к хосту с виртуальной машины, созданной в той же виртуальной сети.

SSL-соединение можно не использовать, но тогда трафик между виртуальной машиной и БД шифроваться не будет.

Давайте подключимся к БД через интернет и создадим в ней ещё одну таблицу. Для выполнения этого задания вы можете использовать виртуальную машину с Ubuntu.

Для создания таблицы сделаем в текстовом редакторе файл

`createTables.sql` с командами. Например, такой:

Скопировать код

```
CREATE TABLE IF NOT EXISTS users (  
    user_id INT AUTO_INCREMENT,  
    nickname VARCHAR(128) NOT NULL,  
    avatar VARCHAR(255),  
    mail VARCHAR(255),  
    PRIMARY KEY (user_id)  
) ENGINE=INNODB;
```

Чтобы выполнить этот запрос в БД, подключимся к хосту. Для этого понадобится SSL-сертификат. Команды для его получения в Ubuntu:

Чтобы получить команды для подключения к БД, в консоли управления перейдите на страницу кластера, на вкладке **Обзор** нажмите кнопку

Подключиться. В результате их выполнения в директории

`/home/<домашняя_директория>/.mysql/` сохранится SSL-сертификат `root.crt`.

Установите утилиту `mysql-client`.

Скопировать код

```
sudo apt update  
sudo apt install -y mysql-client
```

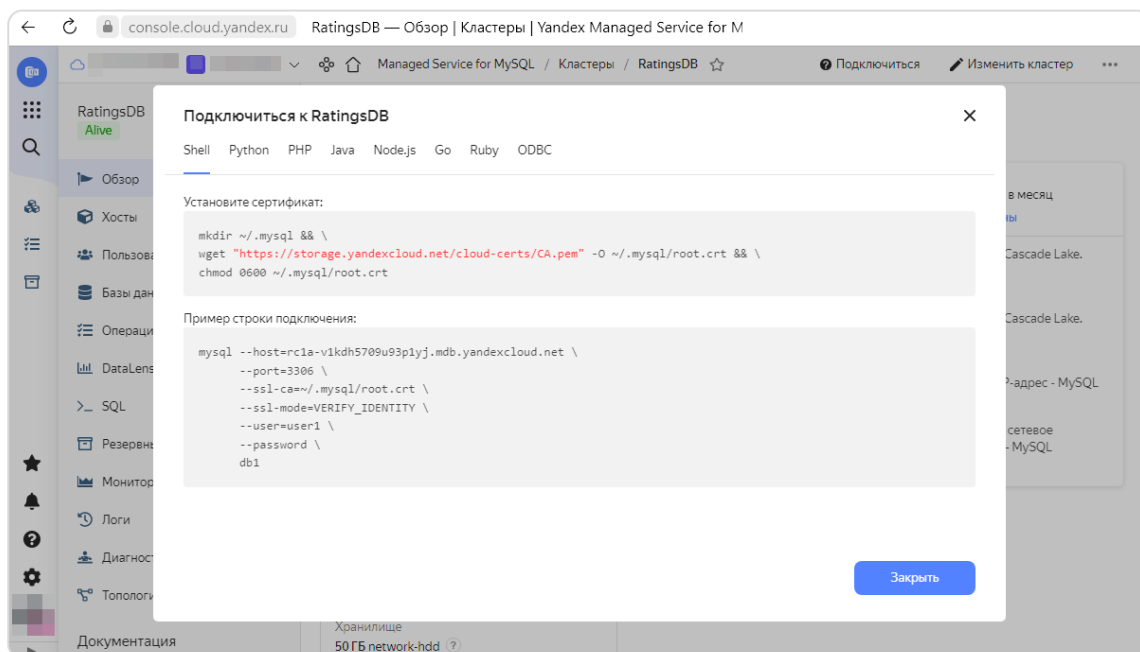
Чтобы подключиться к БД, введите команду `mysql`. Для запуска нашего скрипта она выглядит следующим образом:

Скопировать код

```
mysql --host=<адрес хоста> \  
    --port=3306 \  
    --ssl-ca=/home/alina/CA.pem \  
    --ssl-mode=VERIFY_IDENTITY \  
    --user=<имя пользователя> \  
    --ssl-cert=/home/alina/root.crt
```

```
--password \  
<имя_базы_данных> < createTables.sql
```

Сервис помогает заполнить параметры в команде. Чтобы посмотреть пример команды с адресом хоста, именами пользователя и БД, в консоли управления перейдите на страницу кластера, на вкладке **Обзор** нажмите кнопку **Подключиться**.



После запуска команды введите пароль к БД, после чего в ней будет создана таблица `users`.

Если при создании кластера вы **не включили** публичный доступ, то к БД можно подключиться с виртуальной машины из той же облачной сети без использования шифрования.

\\

Следовательно, в этом случае в команде для подключения опускается параметр `--ssl-ca`, а `--ssl-mode` передаётся со значением `DISABLED`:

Скопировать код

```
mysql --host=адрес_хоста \  
      --port=3306 \  
      --ssl-mode=DISABLED \  
      --user=<имя пользователя> \  
      --password \  
      <имя_базы_данных> < createTables.sql
```

Естественно, подключаться к БД можно не только из командной оболочки, но и из приложений. Нажмите уже знакомую вам кнопку **Подключиться** и посмотрите примеры кода для Python, PHP, Java, Node.js, Go, Ruby или настроек для драйвера ODBC.

Если вы хотите перенести БД в облако, то понадобится создать **дамп** и восстановить его в нужном кластере. Дамп — это копия БД или её части, представляющая собой текстовый файл с командами SQL (например, CREATE TABLE или INSERT). Его создают с помощью утилиты [mysqldump](#).

Давайте попробуем перенести данные в кластер с помощью дампа. Для этого воспользуемся [тестовой БД](#) с данными о сотрудниках компании (имя, дата рождения, дата найма, место работы, зарплата и т. д.). Размер БД — около 167 Мб.

Скачайте из репозитория и сохраните на компьютере файлы с расширениями .sql и .dump. В файле employees.sql содержатся SQL команды, необходимые для создания таблиц и добавления в них данных из dump-файлов. Для переноса тестовой БД в облако понадобится запустить этот файл. Но, прежде чем приступить к переносу БД, откройте этот файл и удалите или закомментируйте (допишите в начало строки --) в нём строку 110. В этой строке расположена команда FLUSH LOGS, которая закрывает и снова открывает файлы журналов, а они в этой тестовой БД отсутствуют. Создайте базу данных employees через консоль управления. Для этого на странице кластера перейдите на вкладку **Базы данных** и нажмите кнопку **Добавить**.

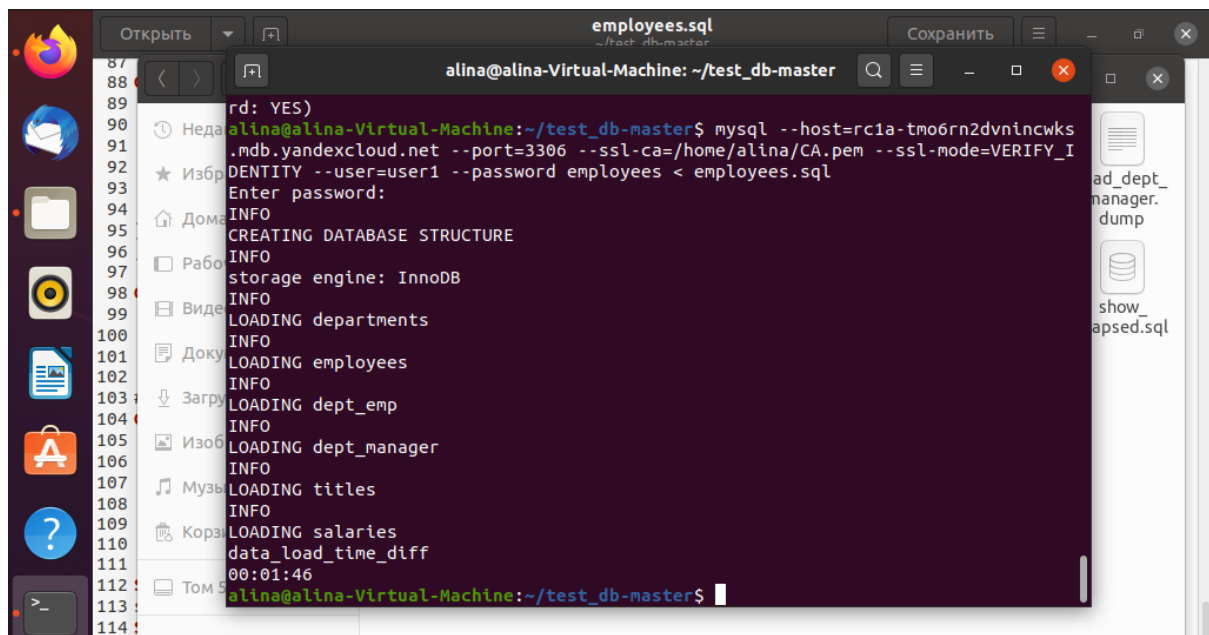
Добавьте пользователю, например user1, разрешение на доступ к БД employees. Для этого на странице кластера перейдите на вкладку **Пользователи**, напротив пользователя user1 нажмите кнопку ... и выберите **Настроить**. Во всплывающем окне нажмите **Добавить базу данных**, выберите employees, добавьте роль ALL_PRIVILEGES и нажмите **Сохранить**.

Затем в командной строке перейдите в папку сохраненными файлами .sql и .dump и восстановите данные из дампа с помощью команды:

Скопировать код

```
mysql --host=<адрес хоста> \  
      --port=3306 \  
      --ssl-ca=/home/alina/CA.pem \  
      --ssl-mode=VERIFY_IDENTITY \  
      --user=<имя_пользователя> \  
      --password \  
      <имя_базы_данных> <имя_таблицы> <имя_файла>
```

```
employees < employees.sql
```



The screenshot shows a terminal window titled "employees.sql" with the command prompt "alina@alina-Virtual-Machine: ~/test_db-master". The command executed is `mysql --host=rc1a-tmo6rn2dvnincwks.mdb.yandexcloud.net --port=3306 --ssl-ca=/home/alina/CA.pem --ssl-mode=VERIFY_IDENTITY --user=user1 --password employees < employees.sql`. The output shows the MySQL client loading the database structure and data from the file. The output includes: "Enter password:", "INFO", "CREATING DATABASE STRUCTURE", "INFO", "storage engine: InnoDB", "INFO", "LOADING departments", "INFO", "LOADING employees", "INFO", "LOADING dept_emp", "INFO", "LOADING dept_manager", "INFO", "LOADING titles", "INFO", "LOADING salaries", "data_load_time_diff", "00:01:46", and the prompt "alina@alina-Virtual-Machine: ~/test_db-master\$".

После того как данные скопируются, ваш кластер и БД будут готовы к работе. Подключитесь к БД в консоли управления и убедитесь, что данные перенесены.

