

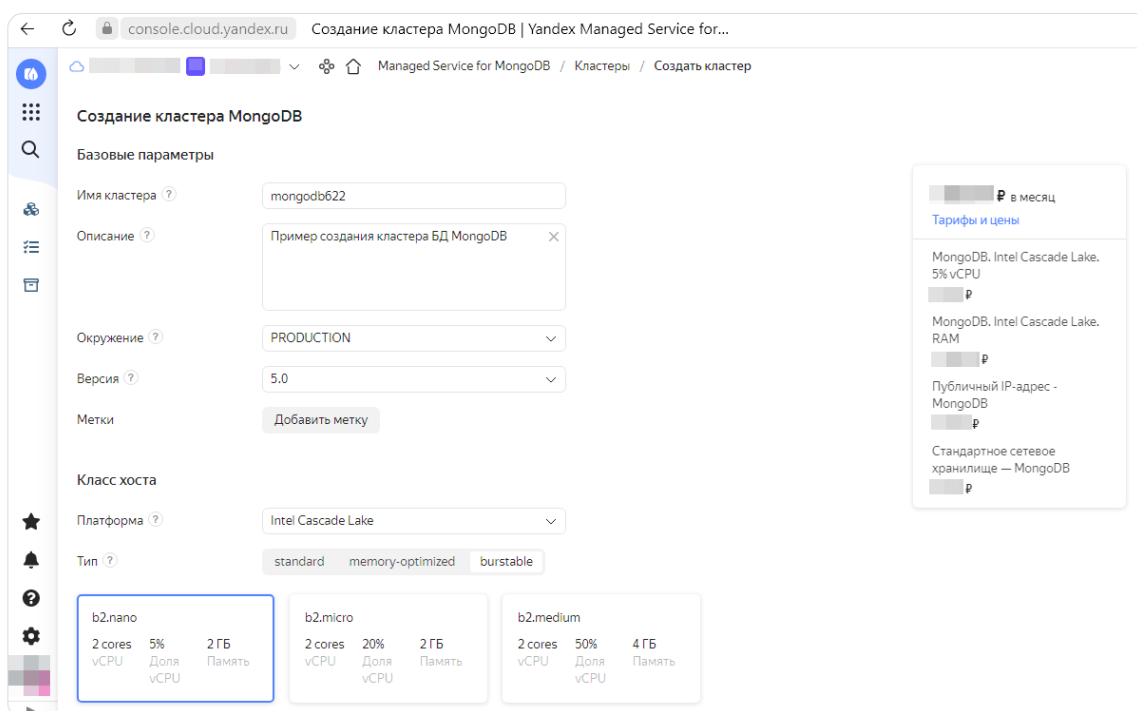
Практическая работа. Создание кластера MongoDB

На этом уроке вы создадите кластер MongoDB, подключитесь к нему и загрузите в него данные. Раньше вы работали только с реляционными БД, но использование кластера MongoDB принципиально не отличается от работы с кластером MySQL или PostgreSQL, так что многое будет вам знакомо.

Создание кластера базы данных

Выберите в консоли управления Yandex Cloud каталог для кластера БД. На дашборде каталога откройте раздел **Managed Service for MongoDB**. В открывшемся окне нажмите кнопку **Создать кластер**.

Установите основные настройки кластера. Для этого урока создайте кластер с минимальной конфигурацией: тип хоста `burstable`, класс `b2.nano`, стандартное сетевое хранилище размером 10 ГБ. Откройте публичный доступ к хосту и задайте пароль пользователя БД. Остальные значения оставьте по умолчанию.



Подключение к базе данных

В сервисе управляемых БД MongoDB к хостам можно подключаться через интернет или с виртуальных машин в той же сети. Порт для подключения — 27018.

Для подключения через интернет хосты кластера должны находиться в публичном доступе. Подключаться можно только через зашифрованное соединение.

Обратите внимание: если публичный доступ настроен только для некоторых хостов в кластере, то при автоматической смене основной реплики она может оказаться недоступной из интернета.

Если к хосту нет публичного доступа и вы подключаетесь к нему с виртуальных машин Yandex Cloud, то зашифрованное соединение необязательно.

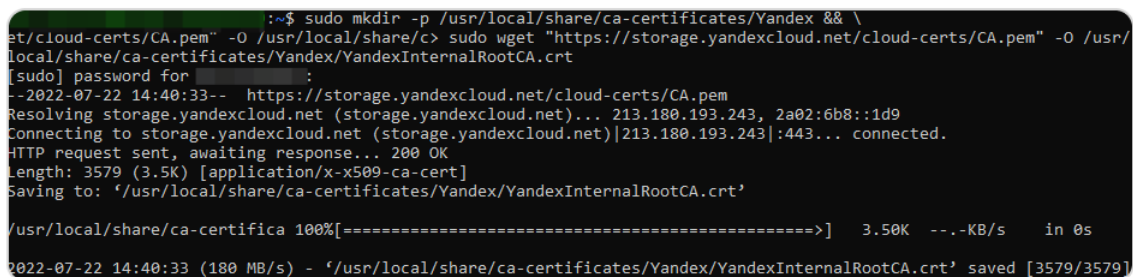
Подключитесь к созданной БД из интернета. Используйте SSL-сертификат, который вы подготовили на одной из предыдущих практических работ, или команду (для Ubuntu):

Скопировать код

```
sudo mkdir -p /usr/local/share/ca-certificates/Yandex && \
sudo wget "https://storage.yandexcloud.net/cloud-certs/CA.pem" -O
/usr/local/share/ca-certificates/Yandex/YandexInternalRootCA.crt
```

Если всё пройдет успешно — вы получите сообщение операционной системы о том, что сертификат сохранён.

Проверьте себя



```
~$ sudo mkdir -p /usr/local/share/ca-certificates/Yandex && \
et/cloud-certs/CA.pem" -O /usr/local/share/c> sudo wget "https://storage.yandexcloud.net/cloud-certs/CA.pem" -O /usr/
local/share/ca-certificates/Yandex/YandexInternalRootCA.crt
[sudo] password for :
--2022-07-22 14:40:33-- https://storage.yandexcloud.net/cloud-certs/CA.pem
Resolving storage.yandexcloud.net (storage.yandexcloud.net)... 213.180.193.243, 2a02:6b8::1d9
Connecting to storage.yandexcloud.net (storage.yandexcloud.net)|213.180.193.243|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3579 (3.5K) [application/x-x509-ca-cert]
Saving to: '/usr/local/share/ca-certificates/Yandex/YandexInternalRootCA.crt'

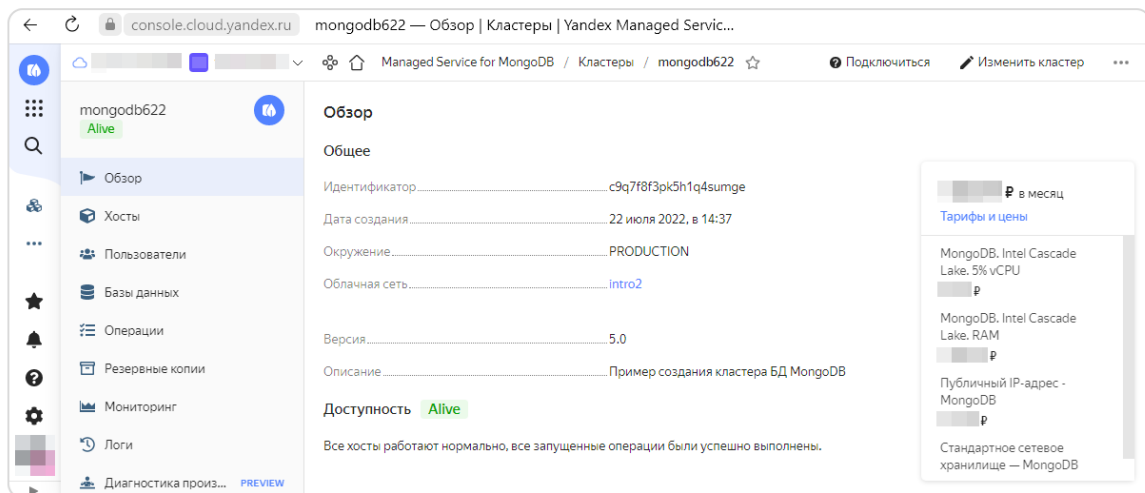
/usr/local/share/ca-certifica 100%[=====>] 3.50K --KB/s in 0s
2022-07-22 14:40:33 (180 MB/s) - '/usr/local/share/ca-certificates/Yandex/YandexInternalRootCA.crt' saved [3579/3579]
```

Установите утилиту [MongoDB Shell](#):

Скопировать код

```
sudo apt-get install -y mongodb-mongosh
```

Подключитесь к БД с помощью команды `mongosh`. Чтобы получить строку подключения, на основной странице сервиса в консоли управления выберите кластер, на вкладке **Обзор** нажмите кнопку **Подключиться**.



Сервис сформирует пример строки подключения для кластера. Там же вы можете посмотреть примеры кода на Python, PHP, Java, Node.js, Go для подключения из приложений.

Подключитесь к кластеру из командной строки.

```
mongosh --norc \
  --tls \
  --tlsCAFile /home/<домашняя директория>/.mongodb/root.crt \
  --host '<FQDN хоста MongoDB>:27018' \
  --username <имя пользователя БД> \
  --password <пароль пользователя БД> \
  <имя БД>
```

При успешном подключении вы получите сообщение:

```

$ mongosh --norc \
  --tls \
  --tlsCAFile /home/<домашняя директория>/.mongodb/root.crt \
  --host 'rs01/rc1b-skmd4u87nappuwer.mdb.yandexcloud.net:27018' \
  --username user1 \
  --password user1user1 \
  db1
Current MongoDB Log ID: 637bc28c7374824a7c9e386f
Connecting to: mongodb://<credentials>@rc1b-skmd4u87nappuwer.mdb.yandexcloud.net:27018/db1?replicaSet=rs01&tls=true&tlsCAFile=K2FHome%2F%2F.mongodb%2Froot.crt&appName=mongosh+1.6.0
Using MongoDB: 5.0.13-mdb-26cf0d36
Using Mongosh: 1.6.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2022-11-21T20:48:48.451+03:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
-----

rs01 [primary] db1>
```

Создадим в БД коллекцию `users`. Предположим, в ней содержится информация о пользователях вашего приложения.

Код приведен на картинке ниже!!!!!!

```
rs01:PRIMARY> db.createCollection("users")
{
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1658492188, 1),
    "signature" : {
      "hash" : BinData(0,"gR+2lJr06gCWc8jpv/tfFm1Qcx4="),
      "keyId" : NumberLong("7123161169736499206")
    }
  },
  "operationTime" : Timestamp(1658492188, 1)
}
rs01:PRIMARY>
```

Загрузим в коллекцию тестовые данные с помощью методов добавления одного документа

`db.insertOne(...)` и сразу нескольких `db.insertMany(...)`.

Сначала добавим один документ (данные одного пользователя).

Скопировать код

```
db.users.insertOne({firstName: "Adam", lastName: "Smith", age: 37,
email: "adam.smith@test.com"});
```

Ответ должен выглядеть примерно так:

```
rs01:PRIMARY> db.users.insertOne({firstName: "Adam", lastName: "Smith", age: 37, email: "adam.smith@test.com"});
{
  "acknowledged" : true,
  "insertedId" : ObjectId("62da956c3baedf297a0da74c")
}
rs01:PRIMARY>
```

Теперь посмотрим на содержимое коллекции с помощью команды `db.users.find()`

```
rs01 [primary] db1> db.users1.find()
[
  {
    _id: ObjectId("63e69a371705bac528677982"),
    firstName: 'Adam',
    lastName: 'Smith',
    age: 37,
    email: 'adam.smith@test.com'
  }
]
rs01 [primary] db1>
```