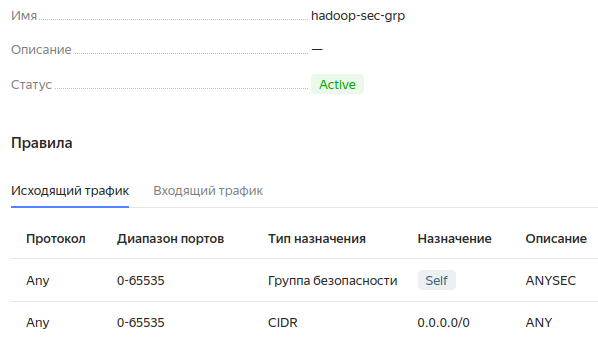
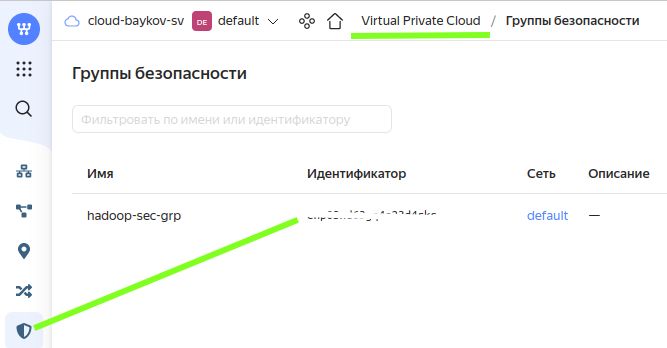
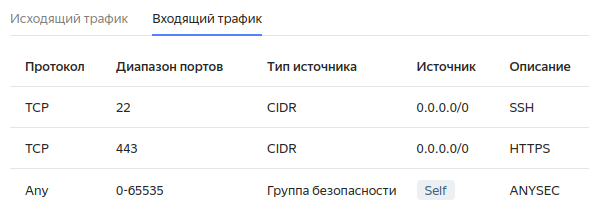
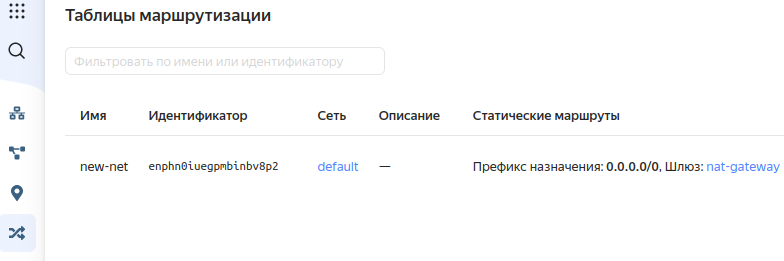
Создать Spark-кластер в Data Proc с двумя подкластерами со следующими характеристиками:

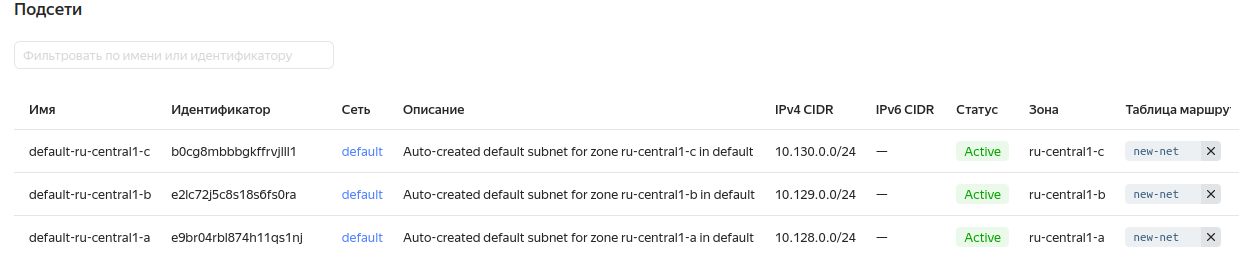
а) Мастер-подкластер: класс хоста s3-c2-m8, размер хранилища 40 ГБ.

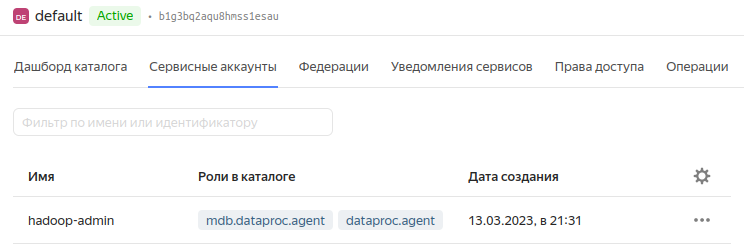
б) Data-подкластер: класс хоста s3-c4-m16, 3 хоста, размер хранилища 128 ГБ.











<https://cloud.yandex.ru/docs/cli/operations/install-cli>

curl -sSL https://storage.yandexcloud.net/yandexcloud-yc/install.sh | bash

yc dataproc cluster create HadoopSparkCluster \

--zone=ru-central1-b \

--service-account-name=hadoop-admin \

--version=2.0 \

--services=HDFS,SPARK,YARN \

--ssh-public-keys-file=/home/dev/.ssh/yac\_key.pub \

--subcluster name="HadoopSparkMasterSubcluster",`

`role=masternode,`

`resource-preset=s3-c2-m8,`

`disk-type=network-hdd,`

`disk-size=40,`

`subnet-name=default-ru-central1-b,`

`assign-public-ip=true \

--subcluster name="HadoopSparkDataSubcluster",`

`hosts-count=3,`

`role=datanode,`

`resource-preset=s3-c4-m16,`

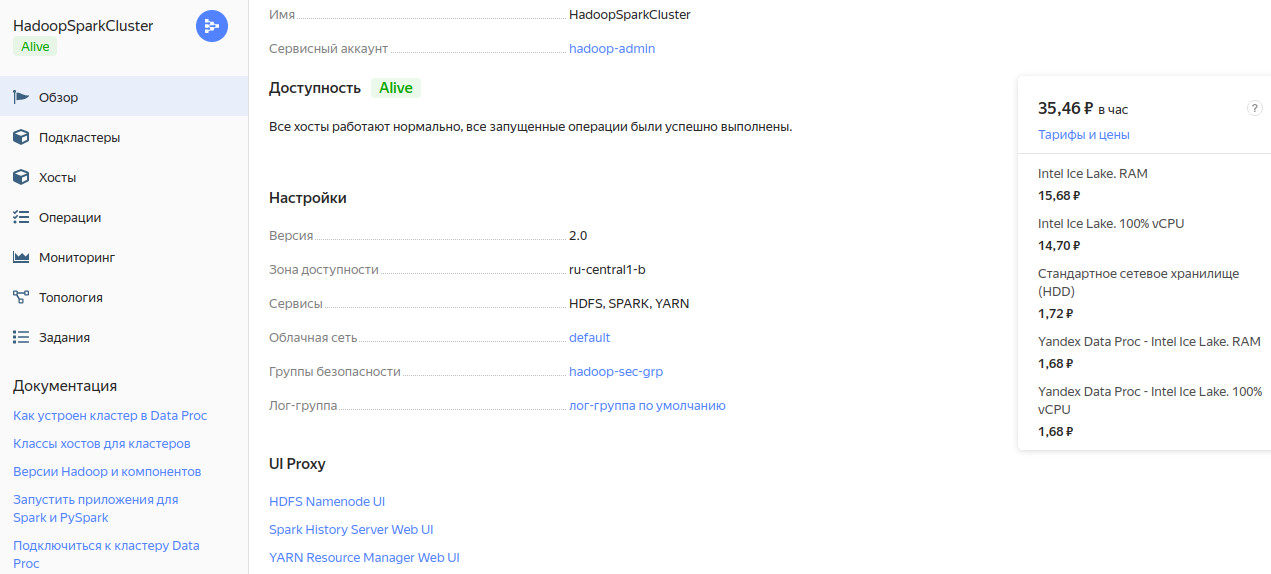
`disk-type=network-hdd,`

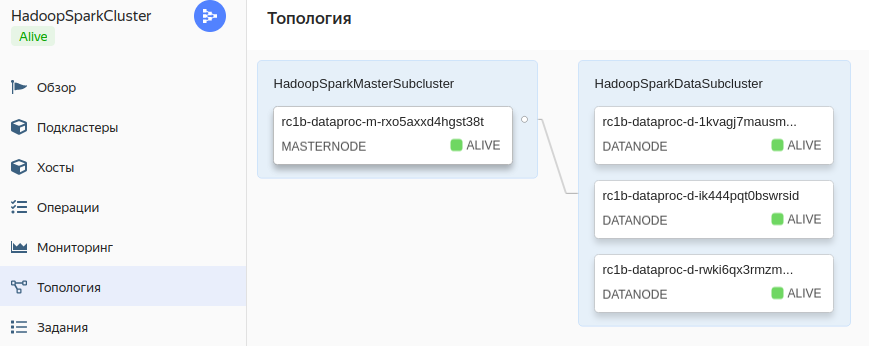
`disk-size=128,`

`subnet-name=default-ru-central1-b \

--security-group-ids=Идентификатор\_of\_hadoop-sec-grp \

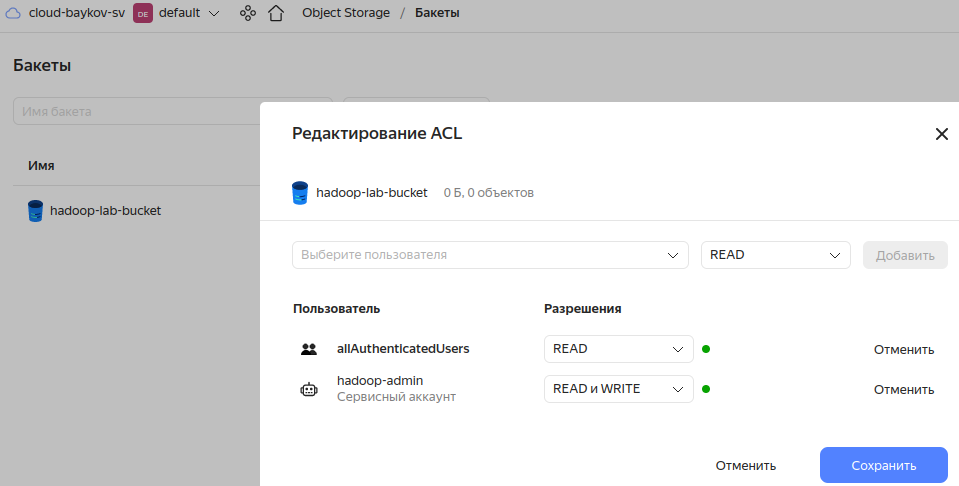
--ui-proxy=true

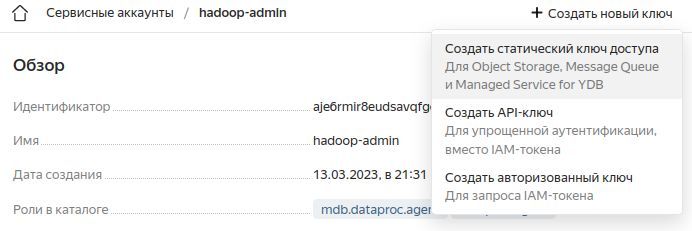




Создать новый backet в Yandex Cloud Object Storage и скопировать в него содержимое предоставленного Вам хранилища с использованием инструмента s3cmd.

Для проверки преподавателем данный basket необходимо сделать общедоступным, а точку доступа к нему привести в README-файле Вашего GitHub-репозитория.





sudo apt install s3cmd

(pip install s3cmd)

s3cmd --configure

ubuntu@rc1b-dataproc-m-rxo5axxd4hgst38t:~$ s3cmd --configure

Access Key: \*\*\*

Secret Key: \*\*\*

Default Region [US]: **ru-central1**

S3 Endpoint [s3.amazonaws.com]: **storage.yandexcloud.net**

DNS-style bucket+hostname:…...: **%(bucket)s.storage.yandexcloud.net**

s3cmd get s3://otus-bigdata/train.csv

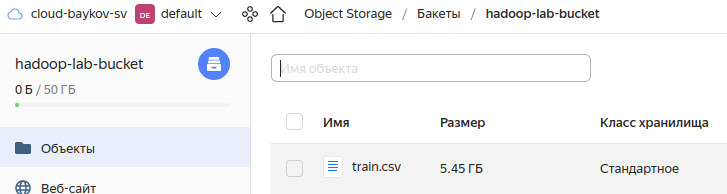
ubuntu@rc1b-dataproc-m-rxo5axxd4hgst38t:~$ **s3cmd get s3://otus-bigdata/train.csv**

download: 's3://otus-bigdata/train.csv' -> './train.csv' [1 of 1]

5846760913 of 5846760913 100% in 62s 88.74 MB/s done

s3://mlops-data/fraud-data/

ubuntu@rc1b-dataproc-m-rxo5axxd4hgst38t:~$ **s3cmd put train.csv s3://hadoop-lab-bucket/**

****

Соединиться по SSH с мастер-узлом и выполнить на нём команду копирования содержимого хранилища в файловую систему HDFS с использованием инструмента hadoop distcp.

Для проверки преподавателем необходимо вывести содержимое HDFS-директории в консоль, а снимок экрана с этой информацией привести в README-файле Вашего GitHub-репозитория.

hdfs dfs -mkdir /user/ubuntu/

hadoop distcp \

-D fs.s3a.bucket.dataproc-examples.endpoint=storage.yandexcloud.net \

-D fs.s3a.bucket.dataproc-examples.access.key=\*\*\* \

-D fs.s3a.bucket.dataproc-examples.secret.key=\*\*\* \

-update \

-skipcrccheck \

-numListstatusThreads 10 \

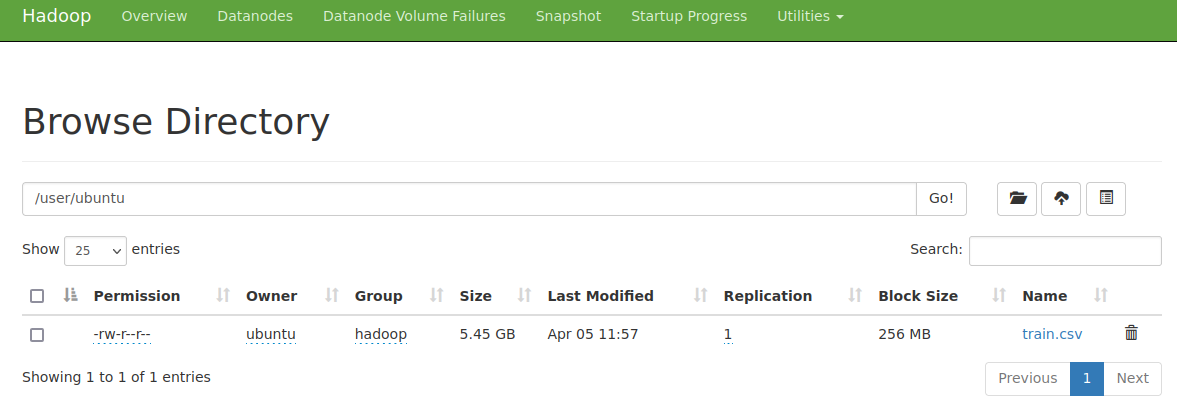
s3a://hadoop-lab-bucket/train.csv \

/user/ubuntu/

**hdfs dfs -ls /user/ubuntu/**

Found 1 items

-rw-r--r-- 1 ubuntu hadoop 5846760913 2023-04-05 08:57 /user/ubuntu/train.csv



hadoop fs -stat %r /user/ubuntu/train.csv

1

sudo nano /etc/hadoop/conf/hdfs-site.xml

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

hdfs dfs -setrep -R 3 /

( hdfs dfs -put train.csv /user/ubuntu/ )

pyspark

df = (spark

.read

.option("header", True)

.csv("train.csv", inferSchema=True)

)

df.printSchema()

df.count()

(df

.select('content\_id', 'answered\_correctly')

.groupby('content\_id')

.mean('answered\_correctly')

.show()

)

>>> df.printSchema()

root

|-- row\_id: integer (nullable = true)

|-- timestamp: long (nullable = true)

|-- user\_id: integer (nullable = true)

|-- content\_id: integer (nullable = true)

|-- content\_type\_id: integer (nullable = true)

|-- task\_container\_id: integer (nullable = true)

|-- user\_answer: integer (nullable = true)

|-- answered\_correctly: integer (nullable = true)

|-- prior\_question\_elapsed\_time: double (nullable = true)

|-- prior\_question\_had\_explanation: boolean (nullable = true)

>>> df.count()

101230332

>>> (df

... .select('content\_id', 'answered\_correctly')

... .groupby('content\_id')

... .mean('answered\_correctly')

... .show()

... )

+----------+-----------------------+

|content\_id|avg(answered\_correctly)|

+----------+-----------------------+

| 6620| 0.7847812401636765|

| 1238| 0.597189012655275|

| 4818| 0.4098718947459534|

| 13285| 0.9481481481481482|

| 11858| 0.6889063317634746|

| 4519| 0.4880797853928695|

| 3175| 0.8605798889574336|

| 496| 0.7542558870632896|

| 1959| 0.559052333804809|

| 8638| 0.4459175781681582|

| 23336| -1.0|

| 6336| 0.777745995423341|

| 3749| 0.6274719401389631|

| 8592| 0.5397205588822356|

| 7993| 0.438069594034797|

| 10362| 0.9089912280701754|

| 3918| 0.5936385255648038|

| 6654| 0.6702702702702703|

| 9376| 0.8307933662034962|

| 3997| 0.6283098243009156|

+----------+-----------------------+

only showing top 20 rows

