

Задание номер 5: «Установка и настройка кластера MySQL»

Выполнил:

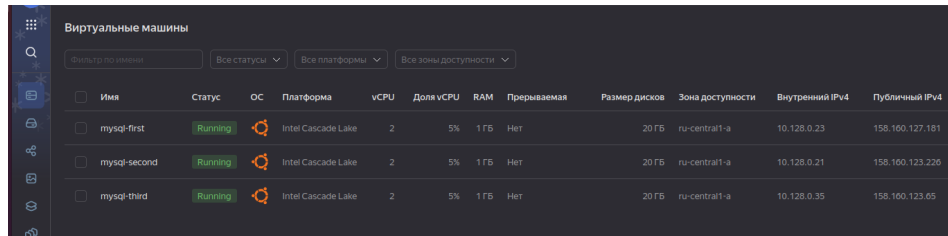
Слушатель курса «Linux Administrator. Advanced»

Бурнейка Артурас Витаутасович

Для целей настоящей работы в Yandex Cloud создаются 3 (три) виртуальные машины:

- mysql-first
- mysql-second
- mysql-third

Результат отработки terraform кода:



Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4
mysql-first	Running		Intel Cascade Lake	2	5%	1 GB	Нет	20 GB	ru-central1-a	10.128.0.23	158.160.127.181
mysql-second	Running		Intel Cascade Lake	2	5%	1 GB	Нет	20 GB	ru-central1-a	10.128.0.21	158.160.123.226
mysql-third	Running		Intel Cascade Lake	2	5%	1 GB	Нет	20 GB	ru-central1-a	10.128.0.35	158.160.123.65

Проверка доступности виртуальных машин в Яндекс-облаке:

```
arturas@arturas:~/Documents/ocds/task-05/ansible$ ansible all --module-name ping --user ubuntu --inventory inventory.yaml
158.160.127.181 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
158.160.123.65 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
158.160.123.226 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Кластер MySQL будет развёрнут на основе технологии InnoDB.

После установки сервера MySQL настроены пользователи root и croot для локального и кластерного администрирования соответственно:

```
1 row in set (0.0003 sec)
MySQL mysql-first:3306 ssl SQL> select User, Host, plugin from mysql.user where User='root' or User='croot';
+-----+-----+-----+
| User | Host | plugin |
+-----+-----+-----+
| croot | %    | caching_sha2_password |
| root  | localhost | mysql_native_password |
+-----+-----+-----+
2 rows in set (0.0012 sec)
MySQL mysql-first:3306 ssl SQL>
```

Все сущности MySQL успешно сконфигурированы и готовы к использованию в кластере:

```
mysql: [Warning] Can't connect to MySQL server on 'mysql-first:3306' (111) (MySQL Error 2003)
MySQL [localhost:33060+ ssl JS]: dba.configureLocalInstance('root@localhost:3306');
Please provide the password for 'root@localhost:3306': ****
Save password for 'root@localhost:3306'? [Y]es/[N]o/[e]x (default No): n
Configuring local MySQL instance listening at port 3306 for use in an InnoDB cluster...

This instance reports its own address as mysql-first:3306
Clients and other cluster members will communicate with it through this address by default. If this is not correct, the report_host MySQL system variable should be changed.

ERROR: User 'root' can only connect from 'localhost'. New account(s) with proper source address specification to allow remote connection from all instances must be created to manage the cluster.

1) Create remotely usable account for 'root' with same grants and password
2) Create a new admin account for InnoDB cluster with minimal required grants
3) Ignore and continue
4) Cancel

Please select an option [1]: 2
Please provide an account name (e.g. icroot@%) to have it created with the necessary
privileges or leave empty and press Enter to cancel.
Account Name: icroot@%
Password for new account: ****
Confirm password: ****

applierWorkerThreads will be set to the default value of 4.

NOTE: Some configuration options need to be fixed:
+-----+-----+-----+-----+
| Variable | Current Value | Required Value | Note |
+-----+-----+-----+-----+
| binlog_transaction_dependency_tracking | COMMIT_ORDER | WRITESET | Update the server variable |
| enforce_gtid_consistency | OFF | ON | Update read-only variable and restart the server |
| gtid_mode | OFF | ON | Update read-only variable and restart the server |
| server_id | 1 | <unique ID> | Update read-only variable and restart the server |
+-----+-----+-----+-----+

Some variables need to be changed, but cannot be done dynamically on the server.
Do you want to perform the required configuration changes? [y/n]: y
Do you want to restart the instance after configuring it? [y/n]: y

Creating user icroot@%.
Account icroot@% was successfully created.

Configuring instance...

WARNING: '@@binlog_transaction_dependency_tracking' is deprecated and will be removed in a future release. (Code 1287).
The instance 'mysql-first:3306' was configured to be used in an InnoDB cluster.
Restarting MySQL...
NOTE: MySQL server at mysql-first:3306 was restarted.
```

Просмотр значения переменной group_replication_consistency:

```
2 rows in set (0.0026 sec)
MySQL [mysql-first:33060+ ssl SQL] > show variables like 'group_replication%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| group_replication_consistency | EVENTUAL |
+-----+-----+
1 row in set (0.0026 sec)
mysql-first:33060+ ssl SQL >
```

Создание кластера с именем arturas:

```
Switching to Javascript mode...
MySQL [mysql-first:33060+ ssl JS]: cluster = dba.createCluster('arturas');
A new InnoDB Cluster will be created on instance 'mysql-first:3306'.

Validating instance configuration at mysql-first:3306...

This instance reports its own address as mysql-first:3306
Instance configuration is suitable.
NOTE: Group Replication will communicate with other members using 'mysql-first:3306'. Use the localAddress option to override.

* Checking connectivity and SSL configuration...

Creating InnoDB Cluster 'arturas' on 'mysql-first:3306'...

Adding Seed Instance...
Cluster successfully created. Use Cluster.addInstance() to add MySQL instances.
At least 3 instances are needed for the cluster to be able to withstand up to
one server failure.

<Cluster:arturas>
MySQL [mysql-first:33060+ ssl JS]:
```

Статус кластера сразу после создания:

```
<Cluster:arturas>
MySQL [mysql-first:3306 ssl JS] > cluster.status();
{
  "clusterName": "arturas",
  "defaultReplicaSet": {
    "name": "default",
    "primary": "mysql-first:3306",
    "ssl": "REQUIRED",
    "status": "OK_NO_TOLERANCE",
    "statusText": "Cluster is NOT tolerant to any failures.",
    "topology": {
      "mysql-first:3306": {
        "address": "mysql-first:3306",
        "memberRole": "PRIMARY",
        "node": "R/W",
        "readReplicas": {},
        "replicationLag": "applier_queue_applied",
        "role": "HA",
        "status": "ONLINE",
        "version": "8.0.35"
      }
    }
  },
  "topologyMode": "Single-Primary"
},
"groupInformationSourceMember": "mysql-first:3306"
}
```

Произведено добавление mysql-second и mysql-third в кластер:

```
Successfully set the value of replica_parallel_workers.
MySQL mysql-first:3306+ ssl JS> cluster.addInstance('croot@mysql-second:3306');

NOTE: The target instance 'mysql-second:3306' has not been pre-provisioned (GTID set is empty). The Shell is unable to decide whether incremental state recovery can correctly provision it.
The safest and most convenient way to provision a new instance is through automatic clone provisioning, which will completely overwrite the state of 'mysql-second:3306' with a physical snapshot from an existing cluster member. To use this method by default, set the 'recoveryMethod' option to 'clone'.

The incremental state recovery may be safely used if you are sure all updates ever executed in the cluster were done with GTIDs enabled, there are no purged transactions and the new instance contains the same GTID set as the cluster or a subset of it. To use this method by default, set the 'recoveryMethod' option to 'incremental'.

Please select a recovery method [C]lone/[I]ncremental recovery/[A]bort (default Clone): c
Validating instance configuration at mysql-second:3306...

This instance reports its own address as mysql-second:3306

Instance configuration is suitable.
NOTE: Group Replication will communicate with other members using 'mysql-second:3306'. Use the localAddress option to override.

* Checking connectivity and SSL configuration...
A new instance will be added to the InnoDB Cluster. Depending on the amount of data on the cluster this might take from a few seconds to several hours.

Adding instance to the cluster...

Monitoring recovery process of the new cluster member. Press ^C to stop monitoring and let it continue in background.
Clone based state recovery is now in progress.

NOTE: A server restart is expected to happen as part of the clone process. If the server does not support the RESTART command or does not come back after a while, you may need to manually start it back.

* Waiting for clone to finish...
NOTE: mysql-second:3306 is being cloned from mysql-first:3306
** Stage DROP DATA: Completed
** Clone Transfer
FILE COPY ##### 100% Completed
PAGE COPY ##### 100% Completed
REDO COPY ##### 100% Completed

NOTE: mysql-second:3306 is shutting down...

* Waiting for server restart... ready
* mysql-second:3306 has restarted, waiting for clone to finish...
** Stage RESTART: Completed
* Clone process has finished: 74.70 MB transferred in about 1 second (~74.70 MB/s)

State recovery already finished for 'mysql-second:3306'

The instance 'mysql-second:3306' was successfully added to the cluster.

MySQL mysql-first:3306+ ssl JS>
```

Привожу статус кластера после добавления все трех сущностей MySQL:

```
MySQL mysql-first:3306 ssl JS> cluster.status();
{
  "clusterName": "arturas",
  "defaultReplicaSet": {
    "name": "default",
    "primary": "mysql-first:3306",
    "ssl": "REQUIRED",
    "status": "OK",
    "statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",
    "topology": {
      "mysql-first:3306": {
        "address": "mysql-first:3306",
        "memberRole": "PRIMARY",
        "mode": "R/W",
        "readReplicas": {},
        "replicationLag": "applier_queue_applied",
        "role": "HA",
        "status": "ONLINE",
        "version": "8.0.35"
      },
      "mysql-second:3306": {
        "address": "mysql-second:3306",
        "memberRole": "SECONDARY",
        "mode": "R/O",
        "readReplicas": {},
        "replicationLag": "applier_queue_applied",
        "role": "HA",
        "status": "ONLINE",
        "version": "8.0.35"
      },
      "mysql-third:3306": {
        "address": "mysql-third:3306",
        "memberRole": "SECONDARY",
        "mode": "R/O",
        "readReplicas": {},
        "replicationLag": "applier_queue_applied",
        "role": "HA",
        "status": "ONLINE",
        "version": "8.0.35"
      }
    }
  },
  "topologyMode": "Single-Primary"
},
"groupInformationSourceMember": "mysql-first:3306"
```

Как видно на рисунке роль PRIMARY закреплена за сущностью mysql-first, с которой производились инициализация кластера и добавление двух других сущностей.

На PRIMARY-сервере создана база данных (cultobjects) и таблица с данными (cultobjects):

```
Query OK, 0 rows affected (0.0570 sec)
MySQL [mysql-second:33060+ ssl] cultobjects SQL> create table cultobjects (
-> id serial,
-> nativeId int not null,
-> nativeName nvarchar(1024) not null,
-> regionId int not null,
-> regionName nvarchar(128) not null,
-> primary key (id)
-> );
Query OK, 0 rows affected, 2 warnings (0.0813 sec)
Warning (code 3720): NATIONAL/NCHAR/NVARCHAR implies the character set UTF8MB3, which will be replaced by UTF8MB4 in a future release
```

С помощью SQL-команды LOAD DATA в таблицу загружены предварительно подготовленные данные из файла формата CSV в количестве 100 объектов:

```
MB4 in order to be unambiguous.
MySQL [mysql-second:33060+ ssl] cultobjects SQL> load data infile "/var/lib/mysql-files/cultobjects.csv"
-> into table cultobjects
-> fields terminated by ','
-> lines terminated by '\n'
-> ignore 1 rows
-> (@nativeId, @nativeName, @regionId, @regionName, @fullAddress, @latitude, @longitude)
-> set nativeId=@nativeId, nativeName=@nativeName, regionId=@regionId, regionName=@regionName;
Query OK, 100 rows affected (0.0144 sec)
Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
```

Содержимое таблицы cultobjects после загрузки данных (первые 10 объектов):

```
Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
MySQL [mysql-second:33060+ ssl] cultobjects SQL> select * from cultobjects limit 10;
```

id	nativeId	nativeName	regionId	regionName
1	147449	Дом, в котором в 1917 г. находился Совет рабочих и солдатских депутатов и Военно-революционный комитет. Здесь в 1925 г. выступал вождь немецких коммунистов Эрнст Тельман	67	Смоленская область
2	147450	Дом Ф.Д. Нагайкова	54	Новосибирская область
3	147454	Дом жилой	67	Смоленская область
4	147455	Памятник Александру Невскому	76	Ярославская область
5	147457	Здание городского торгового корпуса, где 14 декабря 1917 года была провозглашена Советская власть	54	Новосибирская область
6	147467	Военное собрание	56	Оренбургская область
7	147468	Дом со службами А.Я.Балашова	56	Оренбургская область
8	147488	Гостиница «Грандотель» Н.Ф.Мальнева	56	Оренбургская область
9	147508	Здание музея ин. С.Т. Коненкова	67	Смоленская область
10	147509	Дом жилой	67	Смоленская область

```
10 rows in set (0.0000 sec)
```

После создания базы, таблицы и пакетной загрузки данных проведена проверка доступности данных на других серверах кластера.

Данные доступны для чтения.

Также проведена проверка возможности операции WRITE на сервере, являющемся SECONDARY.

Результат – отказ в выполнении операции:

```
6 rows in set (0.0094 sec)
MySQL [localhost:3306+ ssl] SQL> select * from cultobjects.cultobjects limit 10;
```

id	nativeId	nativeName	regionId	regionName
1	147449	Дом, в котором в 1917 г. находился Совет рабочих и солдатских депутатов и Военно-революционный комитет. Здесь в 1925 г. выступал вождь немецких коммунистов Эрнст Тельман	67	Смоленская область
2	147450	Дом Ф.Д. Нагайкова	54	Новосибирская область
3	147454	Дом жилой	67	Смоленская область
4	147455	Памятник Александру Невскому	76	Ярославская область
5	147457	Здание городского торгового корпуса, где 14 декабря 1917 года была провозглашена Советская власть	54	Новосибирская область
6	147467	Военное собрание	56	Оренбургская область
7	147468	Дом со службами А.Я.Балашова	56	Оренбургская область
8	147488	Гостиница «Грандотель» Н.Ф.Мальнева	56	Оренбургская область
9	147508	Здание музея ин. С.Т. Коненкова	67	Смоленская область
10	147509	Дом жилой	67	Смоленская область

```
10 rows in set (0.0131 sec)
MySQL [localhost:3306+ ssl] SQL> truncate table cultobjects.cultobjects;
ERROR: 1290 (HY000): The MySQL server is running with the --super-read-only option so it cannot execute this statement
```

Таким образом, база данных проекта в кластере MySQL с объектами культурного наследия готова к работе. Кластер проверен.

Спасибо OTUS за новый навык и интересное задание!