

### Задание номер 3: «Настройка балансировки веб-приложения»

Выполнил:

Слушатель курса «Linux Administrator. Advanced»

Бурнейка Артурас Витаутасович

-----  
Для целей настоящей работы в Yandex Cloud создаются 4 (четыре) виртуальные машины:

- loadbalancer
- backend-first
- backend-second
- database

В качестве метода балансировки нагрузки будет применена балансировка Least Connected.

В качестве бэкэндов будет разворачиваться сервис, реализованный мной в стеке Microsoft – Asp Net Core (версия 6), язык C#.

Данное решение и стек выбраны мной по причине того, что я являюсь web-разработчиком по данному направлению.

Сервис предоставляет ряд эндпоинтов, которые возвращают культурные объекты, внесённые в реестр объектов культурного наследия Министерством культуры Российской Федерации и Юнеско.

Основной эндпоинт:

- /CultObjects/Region/{Id} (Get)

Все виртуальные машины имеют следующие характеристики:

- Ubuntu 22.04 LTS в качестве операционной системы
- 2 ядра процессора семейства ... с долей процессора 5%
- 1 Гб оперативной память
- 20 Гб жёсткий диск

Проверка доступности группы серверов в Yandex Cloud. Все сервера в работе:

```
arturas@glznz:~/Documents/otus/task-03/ansible$ ansible all --module-name ping --user ubuntu --inventory inventory.yaml
150.160.180.111 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
150.160.122.122 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
150.160.61.231 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
04.201.159.221 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
arturas@glznz:~/Documents/otus/task-03/ansible$
```

После создания виртуальные машины будут иметь следующие DNS-имена и внутренние адреса:

- loadbalancer.ru-central1.internal (10.128.0.31)
- backend-first.ru-central1.internal (10.128.0.33)
- backend-second.ru-central1.internal (10.128.0.3)
- database.ru-central1.internal (10.128.0.17)

С помощью ansible-роли mongodb:

- устанавливается пакет docker.io из официального репозитория Ubuntu
- запускается сервис docker.service
- скачивается docker образ базы данных MongoDB (7.0.4)
- запускается контейнер с базой данных и устанавливаются credentials для работы с базой

С помощью инструментов MongoDB Database Tools в базу данных на сервере database загружается коллекция предварительно подготовленных данных:

```
Database=cultobjects
TargetCollection=cultobjects
Address=62.84.119.79:27017
```

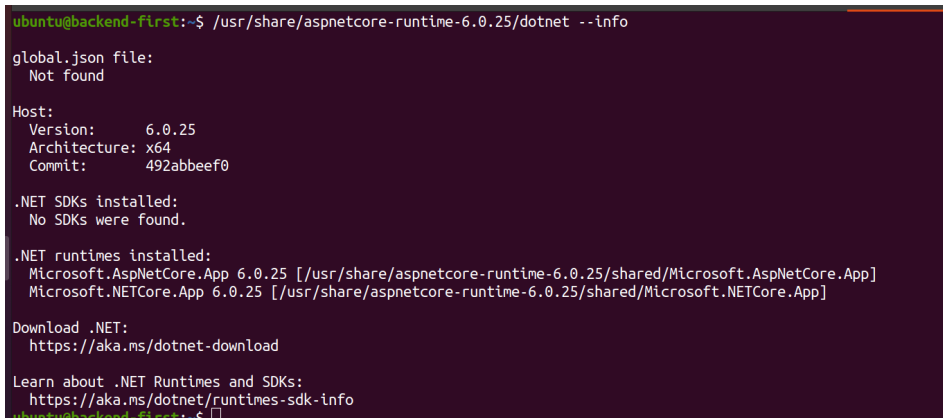
```
mongoimport --db=$Database --collection=$TargetCollection --file=$file
--jsonArray mongodb://<credentials>@$Address/?authSource=admin
```

С помощью ansible-роли aspnetcorert:

- скачивается среда выполнения приложений ASP.NET Core Runtime (текущая версия 6.0.25)
- создаётся каталог /usr/share/aspnetcore-runtime-6.0.25, в который будут распаковано содержимое среды выполнения из архива
- архив со средой выполнения приложений ASP.NET Core Runtime распаковывается в целевой каталог

По завершении работы роли среда выполнения успешно развёрнута на двух бэкэнд серверах (backend-first, backend-second).

Результат работы роли приведен на рисунке:



```
ubuntu@backend-first:~$ /usr/share/aspnetcore-runtime-6.0.25/dotnet --info
global.json file:
  Not found

Host:
  Version:      6.0.25
  Architecture: x64
  Commit:      492abbef0

.NET SDKs installed:
  No SDKs were found.

.NET runtimes installed:
  Microsoft.AspNetCore.App 6.0.25 [/usr/share/aspnetcore-runtime-6.0.25/shared/Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 6.0.25 [/usr/share/aspnetcore-runtime-6.0.25/shared/Microsoft.NETCore.App]

Download .NET:
  https://aka.ms/dotnet-download

Learn about .NET Runtimes and SDKs:
  https://aka.ms/dotnet/runtimes-sdk-info
ubuntu@backend-first:~$
```

С помощью ansible-роли service:

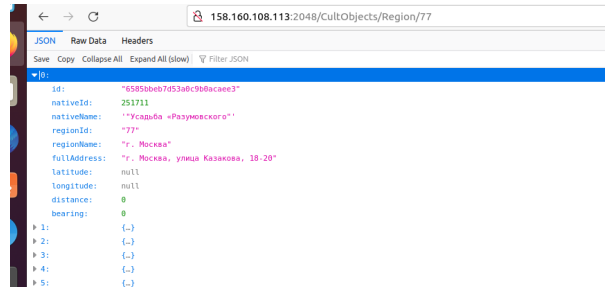
- останавливается systemd unit с сервисом (если он ранее был запущен, например, в случае обновления версии)
- удаляется каталог со старой версией сервиса
- создаётся каталог и в него распаковываются все необходимые для работы файлы (библиотеки, конфигурации)
- копируется systemd файл
- перезагружается системный демон systemd
- запускается служба с сервисом (CultureObjects.service) на порту 2048

Результаты работы роли приведены на рисунке. Сервис запущен:

```
ubuntu@backend-first:~$ sudo systemctl status CultureObjects.service
● CultureObjects.service - Culture objects service
   Loaded: loaded (/etc/systemd/system/CultureObjects.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-12-26 19:23:40 UTC; 16s ago
     Main PID: 5062 (dotnet)
        Tasks: 17 (limit: 1012)
      Memory: 43.4M
         CPU: 1.388s
    CGroup: /system.slice/CultureObjects.service
           └─5062 /usr/share/aspnetcore-runtime-6.0.25/dotnet CultureObjectsService.dll --urls http://0.0.0.0:2048

Dec 26 19:23:43 backend-first CultureObjects[5062]: warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
Dec 26 19:23:43 backend-first CultureObjects[5062]: No XML encryptor configured. Key {d8b44794-4d6b-481c-a0eb-f775bd5d5f24} may
Dec 26 19:23:43 backend-first CultureObjects[5062]: info: Microsoft.Hosting.Lifetime[14]
Dec 26 19:23:43 backend-first CultureObjects[5062]: Now listening on: http://0.0.0.0:2048
Dec 26 19:23:43 backend-first CultureObjects[5062]: info: Microsoft.Hosting.Lifetime[0]
Dec 26 19:23:43 backend-first CultureObjects[5062]: Application started. Press Ctrl+C to shut down.
Dec 26 19:23:43 backend-first CultureObjects[5062]: info: Microsoft.Hosting.Lifetime[0]
Dec 26 19:23:43 backend-first CultureObjects[5062]: Hosting environment: Production
Dec 26 19:23:43 backend-first CultureObjects[5062]: info: Microsoft.Hosting.Lifetime[0]
Dec 26 19:23:43 backend-first CultureObjects[5062]: Content root path: /usr/share/cultureobjects/
ubuntu@backend-first:~$
```

Сервис доступен на двух серверах (backend-first, backend-second) по публичному и приватному адресам на порту номер 2048:



С помощью ansible-роли nginx:

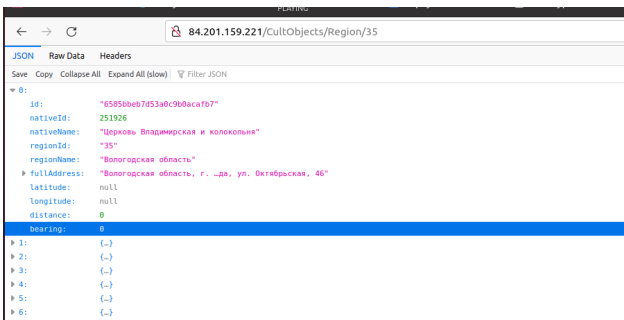
- устанавливается пакет с веб-сервером nginx
- копируется конфигурационный файл с настройками балансировки на бэкэнды, развернутые на машинах backend-first.ru-central1.internal и backend-second.ru-central1.internal
- удаляется серверный блок default
- сервис перезапускается и готов к проксированию и балансировке на развёрнутые бэкэнды

Вэб-сервер Nginx установлен и находится в работе на машине loadbalancer:

```
ubuntu@loadbalancer:~$ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-12-28 18:13:27 UTC; 3min 38s ago
     Docs: man:nginx(8)
  Process: 1724 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 1725 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 1820 (nginx)
    Tasks: 3 (limit: 1012)
   Memory: 7.1M
      CPU: 52ms
   CGroup: /system.slice/nginx.service
           └─1820 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─1822 "nginx: worker process"
               └─1823 "nginx: worker process"

Dec 28 18:13:27 loadbalancer systemd[1]: Starting A high performance web server and a reverse proxy server...
Dec 28 18:13:27 loadbalancer systemd[1]: Started A high performance web server and a reverse proxy server.
```

Результат обращения на сервер loadbalancer по публичному адресу:



Как видно на приведённом рисунке один из бэкэндов вернул запрошенную коллекцию объектов культуры по региону Вологодская область (код региона 35).

Произведена проверка работы системы в случае выхода из строя одного из серверов с бэкэндом (имитация).  
Выключен сервер с доменным именем `backend-first.ru-central1.internal`.

Виртуальные машины											
Фильтры: <input type="text" value="Виртуальные машины"/> Все статусы Все платформы Все зоны доступности											
<input type="checkbox"/>	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Дата создания
<input type="checkbox"/>	backend-first	Stopped	Linux	Intel Cascade Lake	2	5%	1 GB	Нет	20 GB	ru-central1-a	10.128.0.33
<input type="checkbox"/>	loadbalancer	Running	Linux	Intel Cascade Lake	2	5%	1 GB	Нет	20 GB	ru-central1-a	10.128.0.31
<input type="checkbox"/>	backend-second	Running	Linux	Intel Cascade Lake	2	5%	1 GB	Нет	20 GB	ru-central1-a	10.128.0.3
<input type="checkbox"/>	database	Running	Linux	Intel Cascade Lake	2	5%	1 GB	Нет	20 GB	ru-central1-a	10.128.0.17

Сервер loadbalancer продолжает отдавать ответ на запрос за счёт работы сервера `backend-second.ru-central1.internal`.

Проверка завершена.

Спасибо OTUS за новый навык!