# **Задание 1**

Возьмите за основу [решение к заданию 1 из занятия «Подъём инфраструктуры в Яндекс Облаке»](https://github.com/netology-code/sdvps-homeworks/blob/main/7-03.md#%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-1).

1. Теперь вместо одной виртуальной машины сделайте terraform playbook, который:

* создаст 2 идентичные виртуальные машины. Используйте аргумент [count](https://www.terraform.io/docs/language/meta-arguments/count.html) для создания таких ресурсов;
* создаст [таргет-группу](https://registry.terraform.io/providers/yandex-cloud/yandex/latest/docs/resources/lb_target_group). Поместите в неё созданные на шаге 1 виртуальные машины;
* создаст [сетевой балансировщик нагрузки](https://registry.terraform.io/providers/yandex-cloud/yandex/latest/docs/resources/lb_network_load_balancer), который слушает на порту 80, отправляет трафик на порт 80 виртуальных машин и http healthcheck на порт 80 виртуальных машин.

Рекомендуем изучить [документацию сетевого балансировщика нагрузки](https://cloud.yandex.ru/docs/network-load-balancer/quickstart) для того, чтобы было понятно, что вы сделали.

1. Установите на созданные виртуальные машины пакет Nginx любым удобным способом и запустите Nginx веб-сервер на порту 80.
2. Перейдите в веб-консоль Yandex Cloud и убедитесь, что:

* созданный балансировщик находится в статусе Active,
* обе виртуальные машины в целевой группе находятся в состоянии healthy.

1. Сделайте запрос на 80 порт на внешний IP-адрес балансировщика и убедитесь, что вы получаете ответ в виде дефолтной страницы Nginx.

*В качестве результата пришлите:*

*1. Terraform Playbook.*

*2. Скриншот статуса балансировщика и целевой группы.*

*3. Скриншот страницы, которая открылась при запросе IP-адреса балансировщика.*

# **Решение 1**

Плейбук:

1. terraform {
2. required\_providers {
3. yandex = {
4. source = "yandex-cloud/yandex"
5. }
6. }
7. }
8. provider "yandex" {
9. cloud\_id = "b1grulihhlu8ubggpm9k"
10. folder\_id = "b1ghitn6n7m4ui0tnhfd"
11. zone = "ru-central1-a"
12. }
13. resource "yandex\_compute\_instance" "server" {
14. count = 2
15. name = "server-${count.index}"
16. resources {
17. cores = 2
18. memory = 4
19. }
20. boot\_disk {
21. initialize\_params {
22. image\_id = "fd84ocs2qmrnto64cl6m" # ubuntu 22 lts
23. size = 30
24. }
25. }
26. network\_interface {
27. subnet\_id = yandex\_vpc\_subnet.subnet.id
28. nat = true
29. }
31. metadata = {
32. user-data = "${file("./meta.yaml")}"
33. }
34. }
35. resource "yandex\_vpc\_network" "network" {
36. name = "network"
37. }
38. resource "yandex\_vpc\_subnet" "subnet" {
39. name = "subnet"
40. zone = "ru-central1-a"
41. network\_id = yandex\_vpc\_network.network.id
42. v4\_cidr\_blocks = ["192.168.10.0/24"]
43. }
44. resource "yandex\_lb\_target\_group" "target-group" {
45. name = "target-group"
46. region\_id = "ru-central1"
47. dynamic "target" {
48. for\_each = yandex\_compute\_instance.server
49. content {
50. subnet\_id = yandex\_vpc\_subnet.subnet.id
51. address = target.value.network\_interface[0].ip\_address
52. }
53. }
54. }
55. resource "yandex\_lb\_network\_load\_balancer" "load-balancer" {
56. name = "load-balancer"
57. listener {
58. name = "listener"
59. port = 80
60. external\_address\_spec {
61. ip\_version = "ipv4"
62. }
63. }
64. attached\_target\_group {
65. target\_group\_id = yandex\_lb\_target\_group.target-group.id
66. healthcheck {
67. name = "http"
68. http\_options {
69. port = 80
70. }
71. }
72. }
73. }

Устанавливал nginx через ансибл, запуск роли:

---

- name: Выполнение роли cloud-balance

hosts: server

become: true

roles:

- cloud-balance

таск main.tf:

---

- name: install and start nginx

import\_tasks: nginx.yml

таски:

---

- name: Установка пакета Nginx

apt:

name: nginx

state: present

- name: Запуск службы Nginx

service:

name: nginx

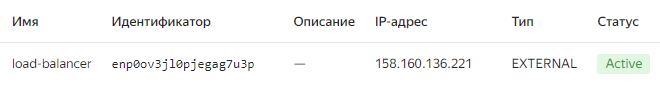
state: started

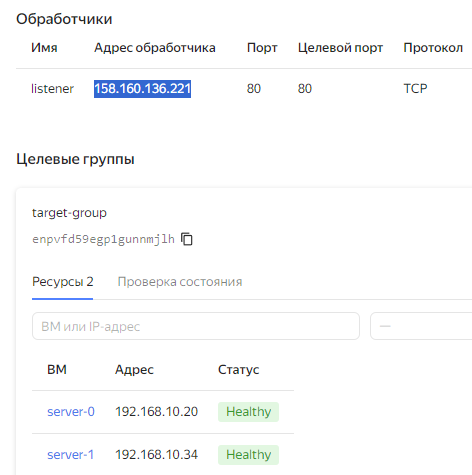
enabled: true

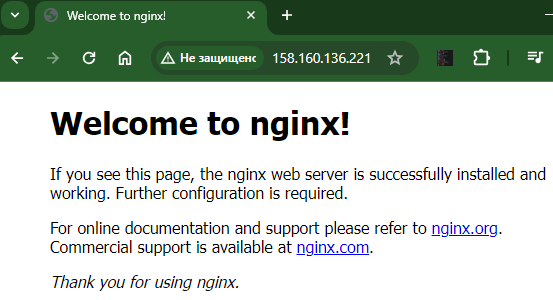
вывод:

1. output "external\_ips" {
2. value = [for instance in yandex\_compute\_instance.server : instance.network\_interface[0].nat\_ip\_address]
3. }
4. output "internal\_ips" {
5. value = [for instance in yandex\_compute\_instance.server : instance.network\_interface[0].ip\_address]
6. }

Сам балансир, статус таргетов, адрес балансировщика через браузер:







# **Задание 2**

1. Теперь вместо создания виртуальных машин создайте [группу виртуальных машин с балансировщиком нагрузки](https://cloud.yandex.ru/docs/compute/operations/instance-groups/create-with-balancer).
2. Nginx нужно будет поставить тоже автоматизированно. Для этого вам нужно будет подложить файл установки Nginx в user-data-ключ [метадаты](https://cloud.yandex.ru/docs/compute/concepts/vm-metadata) виртуальной машины.

* [Пример файла установки Nginx](https://github.com/nar3k/yc-public-tasks/blob/master/terraform/metadata.yaml).
* [Как подставлять файл в метадату виртуальной машины.](https://github.com/nar3k/yc-public-tasks/blob/a6c50a5e1d82f27e6d7f3897972adb872299f14a/terraform/main.tf#L38)

1. Перейдите в веб-консоль Yandex Cloud и убедитесь, что:

* созданный балансировщик находится в статусе Active,
* обе виртуальные машины в целевой группе находятся в состоянии healthy.

1. Сделайте запрос на 80 порт на внешний IP-адрес балансировщика и убедитесь, что вы получаете ответ в виде дефолтной страницы Nginx.

*В качестве результата пришлите*

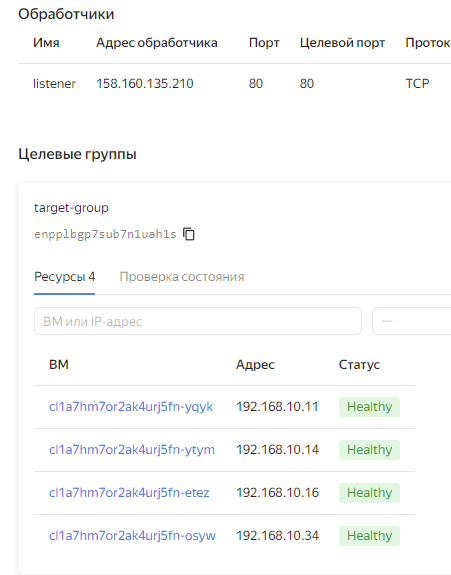
*1. Terraform Playbook.*

*2. Скриншот статуса балансировщика и целевой группы.*

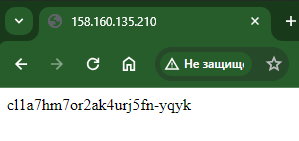
*3. Скриншот страницы, которая открылась при запросе IP-адреса балансировщика.*

# **Решение 2**

1. terraform {
2. required\_providers {
3. yandex = {
4. source = "yandex-cloud/yandex"
5. }
6. }
7. }
8. provider "yandex" {
9. cloud\_id = "b1grulihhlu8ubggpm9k"
10. folder\_id = "b1ghitn6n7m4ui0tnhfd"
11. zone = "ru-central1-a"
12. }
13. resource "yandex\_compute\_instance\_group" "ig" {
14. name = "fixed-ig-with-balancer"
15. folder\_id = "b1ghitn6n7m4ui0tnhfd"
16. service\_account\_id = "aje7mjkp33a5frd01v1t"
17. instance\_template {
18. platform\_id = "standard-v1"
19. resources {
20. memory = 2
21. cores = 2
22. }
23. boot\_disk {
24. mode = "READ\_WRITE"
25. initialize\_params {
26. image\_id = "fd84ocs2qmrnto64cl6m"
27. }
28. }
29. network\_interface {
30. network\_id = yandex\_vpc\_network.network.id
31. subnet\_ids = [yandex\_vpc\_subnet.subnet.id]
32. nat = true
33. }
34. metadata = {
35. user-data = file("./meta.yaml")
36. }
37. }
38. scale\_policy {
39. fixed\_scale {
40. size = 4
41. }
42. }
43. allocation\_policy {
44. zones = ["ru-central1-a"]
45. }
46. deploy\_policy {
47. max\_unavailable = 1
48. max\_expansion = 0
49. }
50. load\_balancer {
51. target\_group\_name = "target-group"
52. target\_group\_description = "load balancer target group"
53. }
54. }
55. resource "yandex\_vpc\_network" "network" {
56. name = "network"
57. }
58. resource "yandex\_vpc\_subnet" "subnet" {
59. name = "subnet"
60. zone = "ru-central1-a"
61. network\_id = yandex\_vpc\_network.network.id
62. v4\_cidr\_blocks = ["192.168.10.0/24"]
63. }
64. resource "yandex\_lb\_network\_load\_balancer" "load-balancer" {
65. name = "load-balancer"
66. listener {
67. name = "listener"
68. port = 80
69. external\_address\_spec {
70. ip\_version = "ipv4"
71. }
72. }
73. attached\_target\_group {
74. target\_group\_id = yandex\_compute\_instance\_group.ig.load\_balancer.0.target\_group\_id
75. healthcheck {
76. name = "http"
77. http\_options {
78. port = 80
79. }
80. }
81. }
82. }



Тут я использовал код [отсюда](https://github.com/nar3k/yc-public-tasks/blob/master/terraform/metadata.yaml), дабы было заметно когда балансир отработает:



Если дать много запросов - балансир сработает - переведет на другой сервер:

