

Практическое задание номер 01: «Первый Terraform-скрипт».

Выполнил: слушатель курса «Linux. Advanced», Бурнейка Артурас Витаутасович.

Установка Ansible.

Выполнена из официального репозитория Ubuntu (20.04 Focal).

Привожу результаты установки:

```
arturas@ginzzu: ~/Documents/otus/task-01/terraform$ sudo apt-get update
[sudo] password for arturas:
Hit:1 http://packages.microsoft.com/repos/code stable InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:3 http://ca.archive.ubuntu.com/ubuntu focal InRelease
Get:4 http://ca.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [23.6 kB]
Hit:6 http://ca.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:7 http://ca.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [2,960 kB]
Get:8 http://ca.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2,960 kB]
Get:9 http://ca.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [755 kB]
Get:10 http://ca.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1,130 kB]
Get:11 http://ca.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [25.8 kB]
Fetched 6,639 kB in 8s (789 kB/s)
Reading package lists... Done
arturas@ginzzu:~/Documents/otus/task-01/terraform$ sudo apt-get install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
ansible is already the newest version (2.9.6+dfsg-1).
0 upgraded, 0 newly installed, 0 to remove and 299 not upgraded.
arturas@ginzzu:~/Documents/otus/task-01/terraform$ sudo apt list --installed | grep ansible
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
ansible/focal,focal,now 2.9.6+dfsg-1 all [installed]
```

Установка Terraform.

Выполнена вручную после скачивания zip-архива с бинарными файлами с репозитория Yandex (<https://hashicorp-releases.yandexcloud.net/terraform/1.6.3/>).

Установка Yandex Cloud CLI.

Выполнена в соответствии с инструкциями, изложенными в официальной документации Yandex Cloud (<https://cloud.yandex.com/en-ru/docs/cli/quickstart>):

Привожу результаты успешной установки Terraform и yc:

```
arturas@ginzzu: ~/Documents/otus$ terraform --version
Terraform v1.6.3
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.6.4. You can update by downloading from https://www.terraform.io/downloads.html
arturas@ginzzu:~/Documents/otus$ yc --version
Yandex Cloud CLI 0.113.0 linux/amd64
```

Инициализация Yandex Cloud CLI.

Установка OAuth-токена, Cloud-Id, Folder-Id, Zone.

Привожу данные результатов инициализации:

```
arturas@ginzzu:~/Documents/otus$ cat ./Documents/otus/task-01/yc_config_list.txt
token: y0_AgAA...Kd1T_aMnFLW
cloud-id: big9...hvt2
folder-id: big9...87ep
compute-default-zone: ru-central1-d
```

Настройка установщика провайдера Terraform для Yandex Cloud.
Привожу содержимое файла ~/.terraformrc:

```
arturas@ginzzu: ~/Documents/otus
arturas@ginzzu:~$ cat ~/.terraformrc
provider_installation {
  network_mirror {
    url = "https://terraform-mirror.yandexcloud.net/"
    include = ["registry.terraform.io/*/*"]
  }
  direct {
    exclude = ["registry.terraform.io/*/*"]
  }
}
arturas@ginzzu:~$
```

Валидация исходного кода Теггаформ-манифеста.
Привожу результаты валидации:

```
arturas@ginzzu: ~/Documents/otus/task-01
arturas@ginzzu:~/Documents/otus/task-01$ terraform validate
Success! The configuration is valid.
arturas@ginzzu:~/Documents/otus/task-01$
```

Инициализация рабочего каталога с Теггаформ-кодом.
Привожу результаты инициализации:

```
arturas@ginzzu: ~/Documents/otus/task-01/terraform
arturas@ginzzu:~/Documents/otus/task-01/terraform$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of yandex-cloud/yandex from the dependency lock file
- Using previously-installed yandex-cloud/yandex v0.102.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
arturas@ginzzu:~/Documents/otus/task-01/terraform$
```

Просмотр плана выполнения.
Привожу содержимое плана выполнения:

```
arturas@ginzzu: ~/Documents/otus/task-01/terraform
arturas@ginzzu:~/Documents/otus/task-01/terraform$ terraform plan

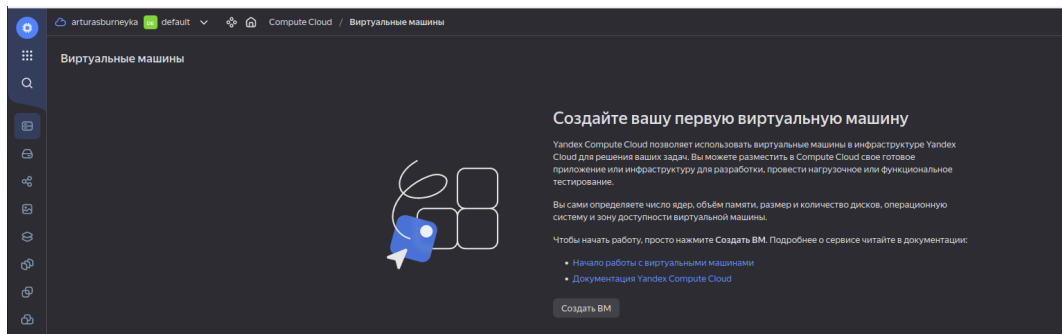
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# yandex_compute_instance.firstserver will be created
+ resource "yandex_compute_instance" "firstserver" {
+   created_at           = (known after apply)
+   folder_id            = (known after apply)
+   fqdn                 = (known after apply)
+   gpu_cluster_id       = (known after apply)
+   hostname             = "arturas-server-01"
+   id                   = (known after apply)
+   metadata              = {
+     "ssh-keys" = "ubuntu:ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGBnK0Q5tUI/2kxvVmq0XcJAF1Rt/hpDT5v8cukKufvWJ2BFNthGdpCett1XQ1QYX+1JRTavXIu
+   }
+   name                 = "arturas-server-01"
+   network_acceleration_type = "standard"
+   platform_id          = "standard-v1"
+   service_account_id    = (known after apply)
+   status                = (known after apply)
+   zone                 = "ru-central1-a"
+   boot_disk {
+     auto_delete = true
+     device_name = (known after apply)
+     disk_id     = (known after apply)
+     mode        = (known after apply)
+   }
+   initialize_params {
+     block_size = (known after apply)
+   }
+ }

Plan: 1 to add, 0 to change, 0 to destroy.
```

Yandex Cloud. Раздел виртуальных машин до применения Terraform-кода:



Применение Terraform-кода для создания виртуальной машины:

```
yandex_compute_instance.firstserver: Creation complete after 2m12s [id=fhmbpsb6f4buj226v8d]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
Outputs:
address = "51.250.76.15"
name = "arturas-server-01"
arturas@ginzzu:~/Documents/otus/task-01/terraform$
```

Yandex Cloud. Раздел виртуальных машин после применения Terraform-кода:

Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания	Идентификатор
arturas-server-01	Running		Intel Broadwell	2	5 %	1 GB	Нет	10 GB	ru-central1-a	10.128.0.16	51.250.76.15	19.11.2023, в 22:53	fhmbpsb6f4buj226v8d

Создание виртуальной машины заняло 2 минуты 12 секунд.

Характеристики (ОС, процессор, память, ЖД) представлены на рисунке.

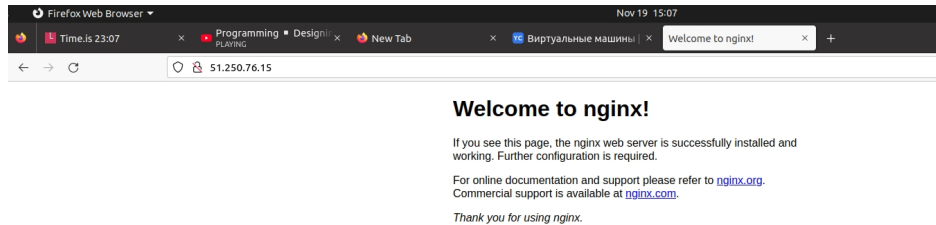
Также привожу информацию о созданной виртуальной машине, полученную через Yandex Cloud CLI:

```
arturas@ginzzu: ~/Documents/otus/task-01/terraform
arturas@ginzzu:~/Documents/otus/task-01/terraform$ yc compute instance list
+-----+-----+-----+-----+-----+-----+
| ID | NAME | ZONE ID | STATUS | EXTERNAL IP | INTERNAL IP |
+-----+-----+-----+-----+-----+-----+
| fhmbpsb6f4buj226v8d | arturas-server-01 | ru-central1-a | RUNNING | 51.250.76.15 | 10.128.0.16 |
+-----+-----+-----+-----+-----+-----+

arturas@ginzzu:~/Documents/otus/task-01/terraform$ yc compute instance get --name arturas-server-01
id: fhmbpsb6f4buj226v8d
folder_id: b1grqjdnjfo2vldj87ep
created_at: "2023-11-19T19:53:22Z"
name: arturas-server-01
zone_id: ru-central1-a
platform_id: standard-v1
resources:
  memory: "1073741824"
  cores: "2"
  core_fraction: "5"
status: RUNNING
metadata_options:
  gce_http_endpoint: ENABLED
  aws_v1_http_endpoint: ENABLED
  gce_http_token: ENABLED
  aws_v1_http_token: DISABLED
boot_disk:
  mode: READ_WRITE
  device_name: fhmb72to49onj3kc3as9
  auto_delete: true
  disk_id: fhmb72to49onj3kc3as9
network_interfaces:
  - index: "0"
    mac_address: d0:0d:bc:df:8b:33
    subnet_id: e9b0d8tdv6faffd7frs
    primary_v4_address:
      address: 10.128.0.16
      one_to_one_nat:
        address: 51.250.76.15
```

После применения провиженера типа local-ехес для запуска ansible-playbook, на целевой виртуальной машине успешно установлен и запущен веб-сервер Nginx.

Прилагаю результат запроса из браузера на адрес созданной виртуальной машины:



Удаление ресурса (виртуальной машины) после проведения проверки корректности работы.

Удаление заняло 58 секунд.

Привожу результаты удаления ресурса:

```
arturas@ginzzu: ~/Documents/otus/task-01/terraform
arturas@ginzzu: ~/Documents/otus/task-01/terraform
}
host_affinity_rules = [] -> null
}
resources {
  core_fraction = 5 -> null
  cores         = 2 -> null
  gpus          = 0 -> null
  memory        = 1 -> null
}
scheduling_policy {
  preemptible = false -> null
}
}
Plan: 0 to add, 0 to change, 1 to destroy.
Changes to Outputs:
  address = "51.250.76.15" -> null
  name    = "arturas-server-01" -> null
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.
Enter a value: yes
yandex_compute_instance.firstserver: Destroying... [id=fhmbpnsb6f4buj226v8d]
yandex_compute_instance.firstserver: Still destroying... [id=fhmbpnsb6f4buj226v8d, 10s elapsed]
yandex_compute_instance.firstserver: Still destroying... [id=fhmbpnsb6f4buj226v8d, 20s elapsed]
yandex_compute_instance.firstserver: Still destroying... [id=fhmbpnsb6f4buj226v8d, 30s elapsed]
yandex_compute_instance.firstserver: Still destroying... [id=fhmbpnsb6f4buj226v8d, 40s elapsed]
yandex_compute_instance.firstserver: Still destroying... [id=fhmbpnsb6f4buj226v8d, 50s elapsed]
yandex_compute_instance.firstserver: Destruction complete after 58s
Destroy complete! Resources: 1 destroyed.
arturas@ginzzu:~/Documents/otus/task-01/terraform$
```