з

Министерство образования и науки
Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ
Департамент информационных технологий и автоматики
Школа бакалавриата

Отчёт по лабораторной работе №6 «Работа с Terraform. Как управлять облачной инфраструктурой»

Выполнил:
Каюмова Юлия РИ-300001

г. Екатеринбург
2023 г.

Оглавление

Работа с Terraform_ как управлять облачной инфраструктурой

Цель работы:

Познакомиться с Terraform провайдером DECORT. Научиться его устанавливать, собирать, инициализировать и настраивать. Научиться создавать виртуальный сервер в облачной платформе DECORT с помощью соответствующего Terraform провайдера.

.

Задачи:

1. Установить Terraform провайдер DECORT.
2. Поместить содержимое архива, скачанного с гитхаба, по пути %userprofile%\AppData\Roaming\terraform.d\plugins\de.de-rtf.urfu.ru\decort\decort\3.4.3\windows_amd64
3. Создать файл main.tf
4. Провести инициализацию Terraform провайдера DECORT
5. Создать виртуальный сервер в облачной платформе DECORT с помощью соответствующего Terraform провайдера.
6.

Ход работы:

1 Установка Terraform провайдера DECORT

Переходим по ссылке https://github.com/rudecs/terraform-provider-decort/releases и устанавливаем нужную версию terraform. В данной ситуации ставим версию для 64-битной Windows

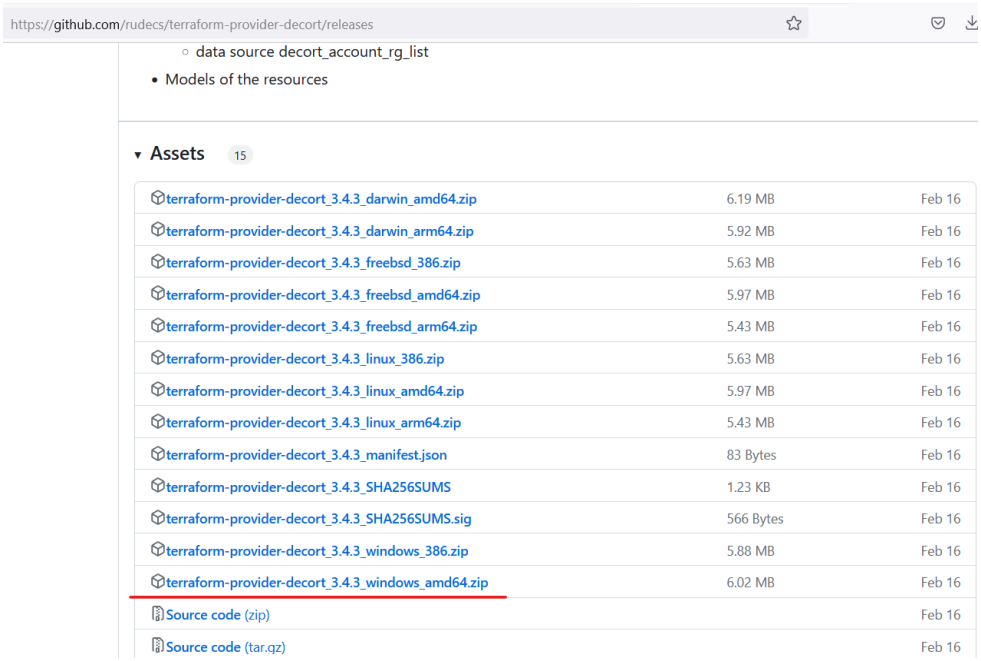

Рисунок 1 - Необходимая версия на гитхабе

Переходим по пути `%userprofile%\AppData\Roaming\terraform.d`

В этой папке создаём ещё папки, чтобы конечный путь выглядел следующим образом:

`%userprofile%\AppData\Roaming\terraform.d\plugins\de.de-rtf.urfu.ru\decort\decort\3.4.3\windows_amd64` и помещаем по этому пути содержимое скачанного с гитхаба архива

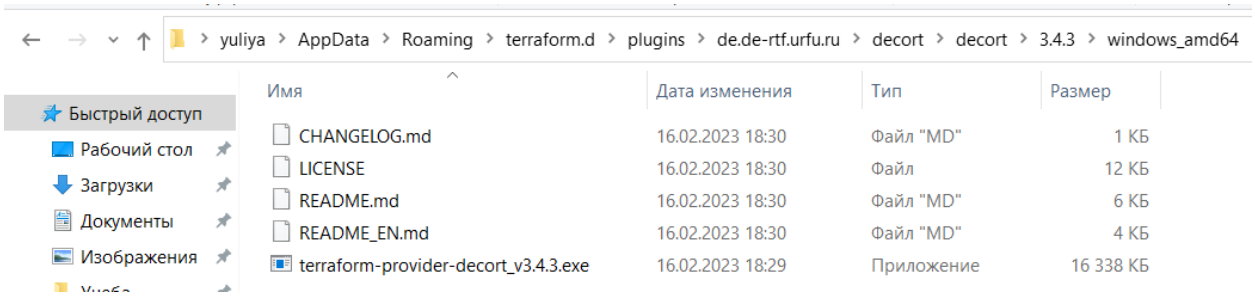

Рисунок 2 - Вставили содержимое архива по нужному пути

Создаём папку со своим именем в папке пользователя

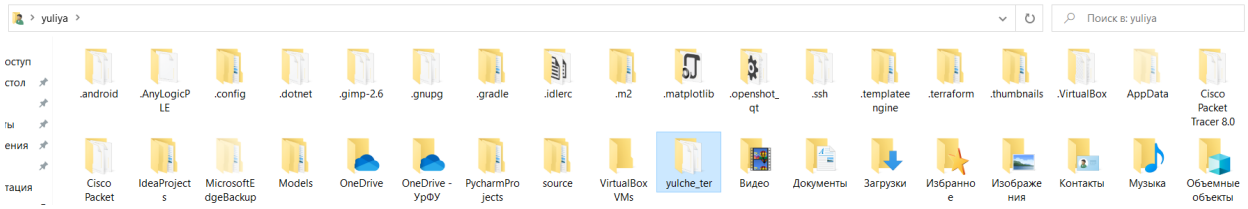

Рисунок 3 - Создана папка yulche_ter

В созданной папке создаём файл `main.tf`, который можно открывать при помощи notepad++

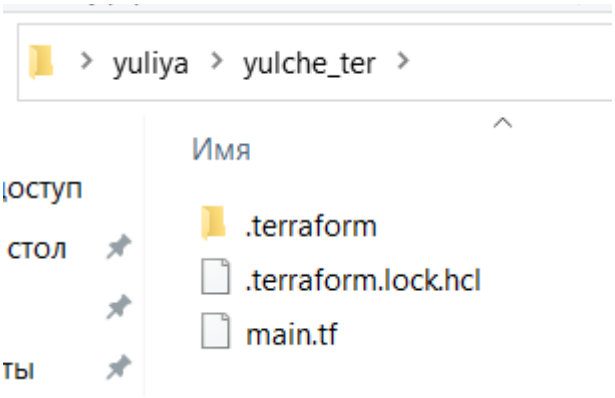

Рисунок 4 - Создан файл main.tf

В созданный файл помещаем следующий блок

```
terraform {
     required_providers {
          decort = {
               version = "3.4.3"
               source = "de.de-rtf.urfu.ru/decort/decort"
          }
     }
}
```

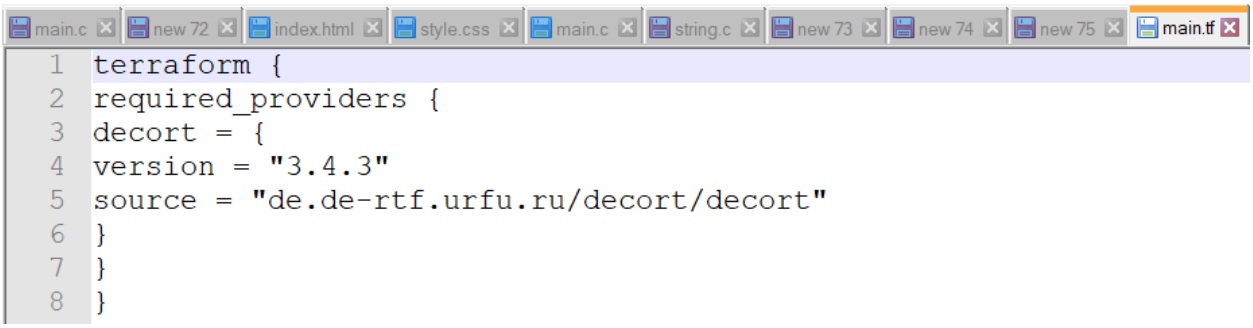

Рисунок 5 - Содержимое файла main.tf

Производим инициализацию terraform командой `terraform init`

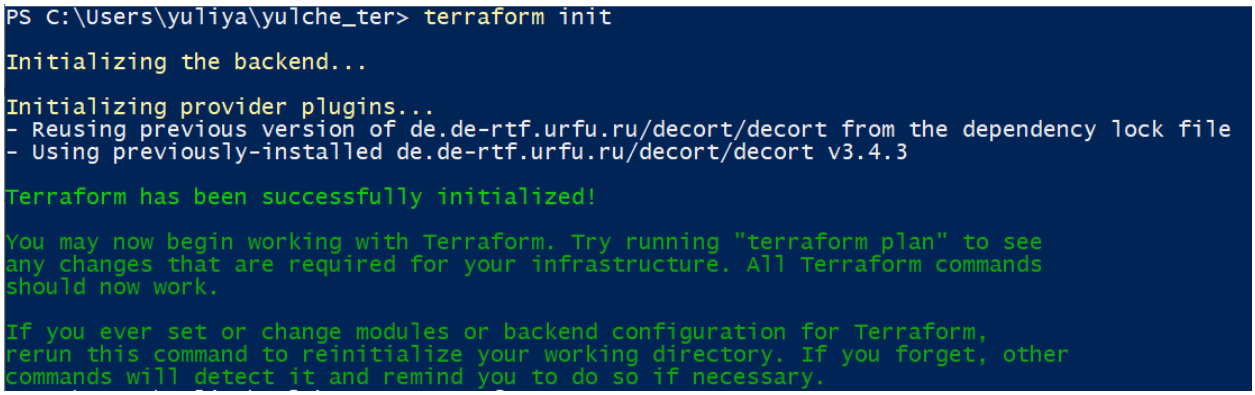

Рисунок 6 - Команда terraform init

Ошибок нет, а значит всё прошло хорошо

2 Инициализация Terraform провайдера DECORT

При инициализации Terraform провайдера DECORT используются следующие аргументы:

Аргумент	**Переменная окружения**	**Описание**
allow_unverified_ssl	-	Если данный аргумент явно установлен в true, то провайдер **не будет** проверять SSL сертификаты при взаимодействии с авторизационным сервисом OAuth2 и контроллером облачной платформы. Отключение проверок может быть полезным при работе в доверенной среде, использующей самоподписанные SSL сертификаты. Однако, так как отключение проверок несёт потенциальные риски безопасности, данную настройку следует использовать с осторожностью. Разрешённые значения: false (значение по умолчанию) и true.
app_id	DECORT_APP_ID	Идентификатор приложения (клиента) для авторизации в контроллере облачной платформы в режиме oauth2. Аргументы app_id и app_secret являются обязательными для режима авторизации authenticator=oauth2. Если app_id не задан в tf-файле, то провайдер будет использовать значение из переменной окружения DECORT_APP_ID.
app_secret	DECORT_APP_SECRET	Секретный код приложения (клиента) для авторизации в контроллере облачной платформы в режиме oauth2. Аргументы app_id и app_secret являются обязательными для режима

		авторизации authenticator=oauth2. Если app_secret не задан в tf-файле, то провайдер будет использовать значение из переменной окружения DECORT_APP_SECRET.
authenticator	-	Режим авторизации при подключении к контроллеру облачной платформы. Доступные режимы: oauth2, legacy или jwt. Данный аргумент является обязательным.
controller_url	-	URL контроллера облачной платформы, через который будет осуществляться управление облачными ресурсами. Данный аргумент является обязательным.
jwt	DECORT_JWT	JSON Web Token (JWT), который используется для авторизации в контроллере облачной платформы в режиме jwt. Данный аргумент является обязательным для режима авторизации authenticator=jwt. Если jwt не задан в tf-файле, то провайдер будет использовать значение из переменной окружения DECORT_JWT
oauth2_url	DECORT_OAUTH2_URL	URL авторизационного сервиса OAuth2, который используется для управления доступом пользователей (или программных клиентов) к контроллеру облачной платформы. Данный аргумент является обязательным для режимов авторизации authenticator=oauth2 или authenti cator=jwt. Если oauth2_url не задан в tf-файле, то провайдер будет использовать значение из переменной окружения DECORT_OAUTH2_URL
password	DECORT_PASSWOR D	Пароль для авторизации в контроллере облачной платформы в режиме legacy. Аргументы password и user являются обязательными для режима авторизации authenticator=legacy. Если password не задан в tf-файле, то провайдер будет использовать значение из переменной окружения DECORT_PASSWORD.
user	DECORT_USER	Имя пользователя для авторизации в контроллере облачной платформы в режиме legacy. Аргументы user и password являются обязательными для режима

		авторизации authenticator=legacy. Если user не задан в tf-файле, то провайдер будет использовать значение из переменной окружения DECORT_USER.

Сначала было необходимо разлогиниться на платформе https://de.de-rtf.urfu.ru/ , затем был произведен переход по ссылке https://de.de-rtf.urfu.ru/ , затем был произведен переход по ссылке https://sso-de.de-rtf.urfu.ru/ . На данной станице сайта был создан API ключ с названием terraform

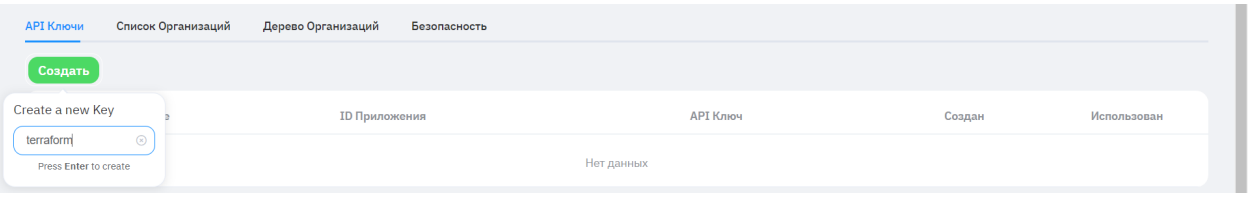

Рисунок 7 - Создание API ключа

После нажатия Enter был создан API ключ

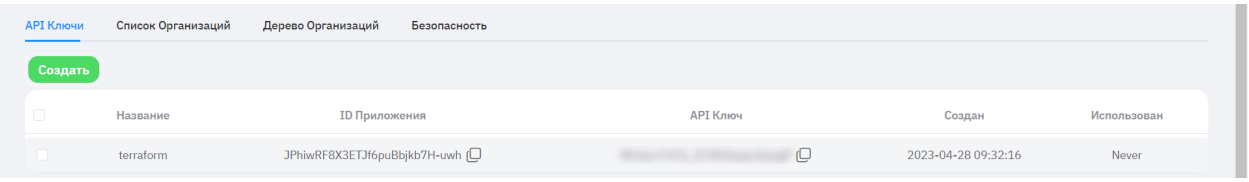

Рисунок 8 - Созданный API ключ

Затем в файле main.tf вставляем app_id и app_secret (строки 16, 17)

```
12 provider "decort" {
13     authenticator = "oauth2"
14     controller_url = "https://de.de-rtf.urfu.ru/" # specify correct DECORT controller URL, e.g. "https://dsl.digitalenergy.online"
15     oauth2_url = "https://sso-de.de-rtf.urfu.ru/" # specify corresponding DECORT SSO URL, e.g. "https://sso.digitalenergy.online"
16     app_id = "JPhiwRF8X3ETJf6puBbjkb7H-uwh"
17     app_secret = "MvUyv-FcE3i_3COEdiaopufqzegP"
18     allow_unverified_ssl = true
19 }
```

Рисунок 9 - Изменение main.tf

Используем задание аккаунта, под которым будем работать

```
21 # 2. Load account to use - new VM will belong to this account
22 data "decort_account" "my_account" {
23     account_id = 11
24     #name = "" # Specify the name of one of your accounts
25 }
```

Рисунок 10 - Настройка рабочего аккаунта

Прописываем id образа. В нашем случае это id = 4 (Ubuntu Linux 2004)

```
# 3. Load OS image to use for VM deployment
data "decort_image" "os_image" {
  image_id = 4 # Specify OS image id, e.g. 1234. You can get accessible image id from data "decort_image_list"
}
```

Рисунок 11 - Id образа ОС

Создаем новую ресурсную группу

```
# 4. Create new Resource Group in the selected account, new VM will be created in this RG
resource "decort_resgroup" "my_rg" {
  name = "Test-with-Terraform"
  account_id = 11
  gid = 249
  def_net_type = "NONE"
  # if you want to set resource quota on this Resource Group, uncomment
  # the following code fragment
    quota {
      cpu = 2     # CPU limit
      ram = 2048  # RAM limit in MB
      disk = 10   # disk volume limit in GB
  }
}
```

Рисунок 12 - Создание ресурсной группы

Прописываем в консоли команду terraform plan

```
PS C:\Users\yuliya\yulche_ter> terraform plan
data.decort_account.my_account: Reading...
data.decort_image.os_image: Reading...
data.decort_account.my_account: Read complete after 2s [id=d9193304-069c-4960-9831-a1df7df8a26b]
data.decort_image.os_image: Read complete after 2s [id=f052d48c-a4b3-40c8-9b1e-fa476809ed83]

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:

  # decort_kvmvm.my_new_vm will be created
  + resource "decort_kvmvm" "my_new_vm" {
      + account_id     = (known after apply)
      + account_name   = (known after apply)
      + boot_disk_id   = (known after apply)
      + boot_disk_size = 3
      + cpu            = 1
      + description    = "Test KVM VM Compute managed by Terraform"
      + detach_disks   = true
      + driver         = "KVM_X86"
      + enabled        = (known after apply)
      + id             = (known after apply)
      + image_id       = 4
      + name           = "tf-managed-vm"
      + os_users       = (known after apply)
      + permanently    = true
      + pool           = (known after apply)
      + ram            = 1024
      + rg_id          = (known after apply)
      + rg_name        = (known after apply)
      + sep_id         = (known after apply)
      + started        = (known after apply)

      + network {
          + mac      = (known after apply)
          + net_id   = 5
          + net_type = "EXTNET"
        }
    }

  # decort_resgroup.my_rg will be created
  + resource "decort_resgroup" "my_rg" {
      + account_id   = 11
      + account_name = (known after apply)
      + computes     = (known after apply)
      + def_net_id   = (known after apply)
      + def_net_type = "NONE"
      + ext_net_id   = 0
      + force        = false
      + gid          = 249
      + id           = (known after apply)
      + name         = "Test-with-Terraform"
      + permanently  = false
      + resources    = (known after apply)
      + status       = (known after apply)
      + vins         = (known after apply)
      + vms          = (known after apply)

      + quota {
          + cpu        = 2
          + disk       = 10
          + ext_ips    = -1
          + ext_traffic = -1
          + gpu_units  = -1
          + ram        = 2048
        }
    }

Plan: 2 to add, 0 to change, 0 to destroy.
```

Рисунок 13 - Результат работы команды terraform plan

3 Задание со звездочкой (разделение на разные файлы)

В папке, где находится main.tf, создаём файл provider.tf

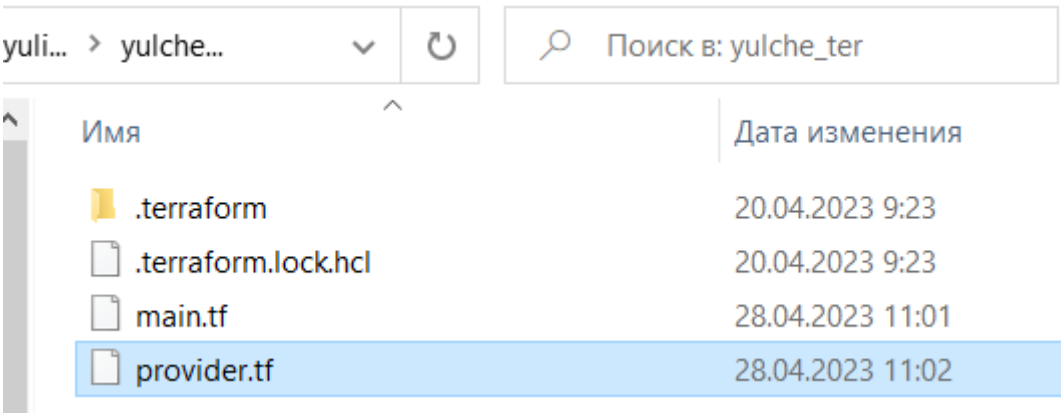

Рисунок 14 - Создали файл provider.tf

В файл provider.tf помещаем верхушку содержимого файла main.tf (из main.tf это дело вырезаем)

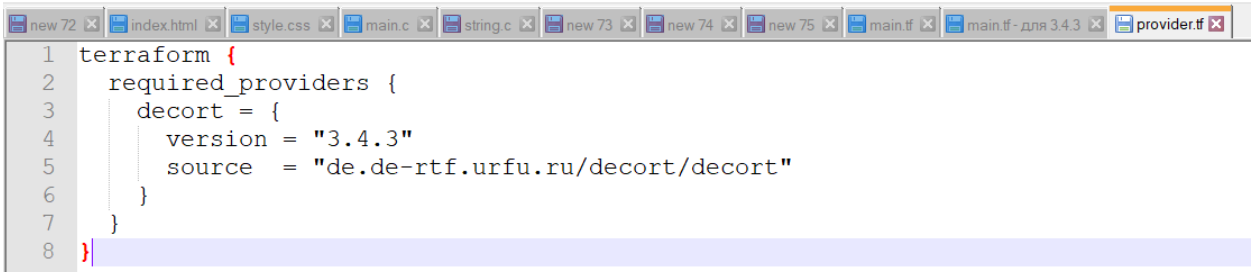

Рисунок 15 - Содержимое файла provider.tf

Создали в той же директории файл variables.tf. В кавычках в default указать значения app_id и app_secret

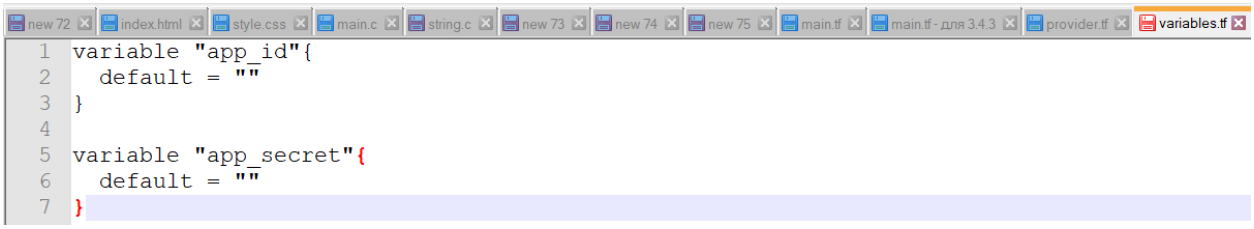

Рисунок 16 - Содержимое файла variables.tf

Произведём изменения в main.tf

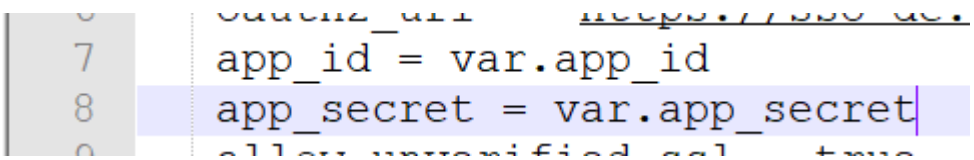

Рисунок 17 - Изменения в main.tf

Вывод

В ходе данной лабораторной работы я познакомилась с Terraform провайдером DECORT. Научилась его устанавливать, собирать, инициализировать и настраивать. Также был создан виртуальный сервер в облачной платформе DECORT с помощью соответствующего Terraform провайдера.