Подготовка облачной инфраструктуры (Подключение Terraform к провайдеру OpenStack)

Задача:

- Подключение Terraform к провайдеру OpenStack
- Подключение openstack-cli к Кибер Инфраструктура

Исходя из требований задания, реализуем в правильных директориях правильные файлы (наименование)

Имеем пользовательские данные для доступа к Кибер Инфраструктура (в данном конкретном случае):

- Кибер Инфраструктура доступна по доменному имени - **cyber-infra.local.prof**;
- Имя домена (в контексте Кибер Инфраструктура) - **Region2025**;
- Пользователь - **user01**;
- Пароль пользователя - **user01P@ssw0rd**;

Вариант реализации:

ControlVM:

- Создадим файл конфигурации зеркала Terraform
  - Файл должен иметь имя **.terraformrc**
  - Файл должен быть расположен в домашнем каталоге пользователя

```
vim ~/.terraformrc
```

  - Помещаем в данный файл следущее содержимое:

```
provider_installation {
    network_mirror {
        url = "https://terraform-mirror.mcs.mail.ru"
        include = ["registry.terraform.io/*/*"]
    }
    direct {
        exclude = ["registry.terraform.io/*/*"]
    }
}
```

- Из под пользователя **altlinux** создаём директорию **bin** в домашнем каталоге пользователя и переходим в неё:

```
mkdir ~/bin && cd ~/bin
```

- Создадим файл **cloud.conf** в котором опишем необходимые переменные (будут использоваться как переменные окружения) для работы **terraform** и **openstack-cli** с Кибер Инфраструктура:

```
vim cloud.conf
```

  - Помещаем в данный файл следущее содержимое:

```
# Terraform
export TF_VAR_OS_AUTH_URL=https://cyber-infra.local.prof:5000/v3
export TF_VAR_OS_PROJECT_NAME=Project1
export TF_VAR_OS_USERNAME=user01
export TF_VAR_OS_PASSWORD=user01P@ssw0rd

# openstacl-cli
export OS_AUTH_URL=https://cyber-infra.local.prof:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_AUTH_TYPE=password
export OS_PROJECT_DOMAIN_NAME=Region2025
export OS_USER_DOMAIN_NAME=Region2025
export OS_PROJECT_NAME=Project1
export OS_USERNAME=user01
export OS_PASSWORD=user01P@ssw0rd
```

- Применяем переменные окружения указанные в файле
  - Стоит учесть данное действие про конечном формирование финального скрипта **cloudinit.sh** (будет рассмотрено далее)

```
source cloud.conf
```

- Проверяем возможность взаимодействовать чере **openstack-cli** с Кибер Инфраструктура:
  - Например выведем список серверов (Виртуальных Машин)

?

```
[altlinux@controlvm bin]$ source cloud.conf
[altlinux@controlvm bin]$ openstack --insecure server list
+--------------------------------------+-----------+--------+--------------------------------------+--------------------------+--------+
| ID                                   | Name      | Status | Networks                             | Image                    | Flavor |
+--------------------------------------+-----------+--------+--------------------------------------+--------------------------+--------+
| 1ca98f06-e5a7-4a35-a80f-ead6c0ce4cf5 | ControlVM | ACTIVE | cloud=192.168.100.139, 192.168.15.177 | N/A (booted from volume) | medium |
+--------------------------------------+-----------+--------+--------------------------------------+--------------------------+--------+
[altlinux@controlvm bin]$
```

- Создадим файл **provider.tf** и опишем параметры для подключения к провайдеру **openstack** для работы с Кибер Инфраструктура используя данного провайдера:

```
vim provider.tf
```

  - Помещаем в данный файл следущее содержимое:

```
terraform {
  required_providers {
    openstack = {
      source  = "terraform-provider-openstack/openstack"
      version = "2.1.0"
    }
  }
}

provider "openstack" {
  auth_url    = var.OS_AUTH_URL
  tenant_name = var.OS_PROJECT_NAME
  user_name   = var.OS_USERNAME
  password    = var.OS_PASSWORD
  insecure    = true
}
```

- Создадим файл **variables.tf** и опишем переменные для подключения к провайдеру **openstack**:
  - При таком подходе переменные будут браться из значений определённых в переменных окружения, которые в свою очередь были указаны в файле **cloud.conf**, т.к. по условиям задания эксперты могут только отредактировать его (например для запуска в другом проекте):

```
vim variables.tf
```

  - Помещаем в данный файл следущее содержимое:

```
variable "OS_AUTH_URL" {
    type = string
    sensitive = true
}

variable "OS_PROJECT_NAME" {
    type = string
    sensitive = true
}

variable "OS_USERNAME" {
    type = string
    sensitive = true
}

variable "OS_PASSWORD" {
    type = string
    sensitive = true
}
```

- Инициализируем текущий каталог для работы с **terraform** и провайдером **openstack**:

```
terraform init
```

  - Результат:

```
altlinux@controlvm: /home/altlinux/bin
[altlinux@controlvm bin]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding terraform-provider-openstack/openstack versions matching "2.1.0"...
- Installing terraform-provider-openstack/openstack v2.1.0...
- Installed terraform-provider-openstack/openstack v2.1.0 (unauthenticated)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.


  Warning: Incomplete lock file information for providers

  Due to your customized provider installation methods, Terraform was forced to calculate lock file checksums locally for the following providers:
    - terraform-provider-openstack/openstack

  The current .terraform.lock.hcl file only includes checksums for linux_amd64, so Terraform running on another platform will fail to install these
  providers.

  To calculate additional checksums for another platform, run:
    terraform providers lock -platform=linux_amd64
  (where linux_amd64 is the platform to generate)

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.
[altlinux@controlvm bin]$
```

Последнее изменение: вторник, 14 января 2025, 07:52

Обратная связь

Подпишитесь

Вы используете гостевой доступ (Вход)

Сводка хранения данных

Тема оформления сайта разработана

conecti.me