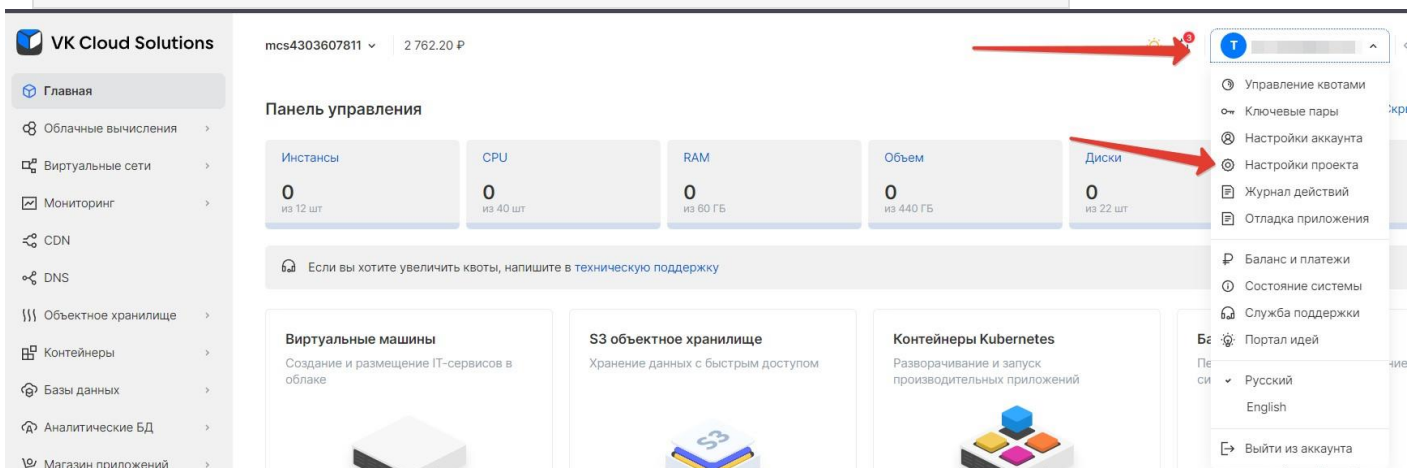


#Для начала нужно сделать папку, в которой будет расположен наш проект.
mkdir Cloud_solution #Скачать файл конфигурации terraform и следовать инструкциям:



mcs4303607811 2 762.20 P

Настройки проекта

Общая информация Квоты Цены API ключи API Endpoints **Terraform**

Terraform

На данный момент VK Cloud Solutions поддерживает работу с двумя Terraform провайдерами: OpenStack для управления IaaS-сервисами и собственный Terraform Provider для управления Kubernetes и базами данных. Они оба включены в файл конфигурации ниже.

Для использования Terraform вам нужно скачать этот файл, указать пароль от вашей учетной записи и разместить его в папке с другими файлами Terraform. После этого использовать команду `terraform init` для инициализации. Подробнее об использовании Terraform в [этой статье](#). Также вы можете обратиться к документации провайдера на сайте [Terraform](#).

Скачать файл конфигурации Terraform

Параметр	Значение
Project ID / tenant_id	73c38e3fc3d04e6cb5949b222a673c8f
Username	
User Domain ID	users
Region Name	RegionOne

Будет скачан файл mcs_provider.tf. Внутри будет следующее содержание:

```
terraform {
  required_providers {
    openstack = {
      source = "registry.tfpla.net/terraform-provider-openstack/openstack"
```

```

    }

    mcs = {
      source =
"registry.tfpla.net/MailRuCloudSolutions/mcs"
    }
  }

provider "openstack" {
  # Your
user account.    user_name = "тут
будет почта "

  # The password of the account
password = "YOUR_PASSWORD"

  # The tenant token can be taken from the project Settings tab - > API
keys.    # Project ID will be our token.    tenant_id =
"73c38e3fc3d04e6cb5949b222a673c8f"

  # The indicator of the location of users.
user_domain_id = "users"

  # API endpoint    # Terraform will use this address to access the VK
Cloud Solutions api.    auth_url = "https://infra.mail.ru:35357/v3/"

  # use octavia to manage load balancers
use_octavia = true

  # Region name
region = "RegionOne" }

provider "mcs" {
  # Your user
account.    username = "тут
будет почта"

  # The password of the account
password = "YOUR_PASSWORD"

```

```

# The tenant token can be taken from the project Settings tab - > API
keys.      # Project ID will be our token.      project_id =
"73c38e3fc3d04e6cb5949b222a673c8f"

# Region name
region = "RegionOne"
}

```

1. Подготовьте сценарий автоматизации развёртывания облачной инфраструктуры.

Статьи из документации в помощь:

<https://mcs.mail.ru/docs/networks/vnet/networks/tf-quick-start> - пример конфига

<https://mcs.mail.ru/docs/networks/vnet/networks/create-vm-tf> - пример конфига

Инструкции по модулям:

https://registry.tfpla.net/providers/vkcs/vkcs/latest/docs/resources/vkcs_compute_instance Инструкции по Openstack CLI:

<https://mcs.mail.ru/docs/additional/account/project/cli/setup>

<https://mcs.mail.ru/docs/additional/account/project/cli/authorization> Типы и размеры диска:

<https://mcs.mail.ru/docs/ru/base/iaas/vm-volumes/volume-sla>

1. Виртуальные машины и сети должны быть созданы согласно Топологии.
2. Имена виртуальных машин и сетей должны соответствовать Топологии.
3. Обеспечьте подключение виртуальных машин к соответствующим сетям.
4. Внешний доступ к устройствам должен быть разрешён только по протоколам ssh, http и https.

```

# Предварительно создать ключи на локальной машине

ssh-keygen # Проверить имя публичного ключа и
использовать ниже #Текст основного файла
конфигурации.

resource "openstack_compute_keypair_v2" "ssh" {
# Имя ssh ключа    name = "terraform_ssh_key"

# Путь до публичного ключа
# Тут указывается содержимое terraform_ssh_key
# В примере он находится в одной директории с main.tf

```

```

    public_key = file("./id_rsa.pub")
}

resource "openstack_compute_secgroup_v2" "rules" {
  name = "${var.instance_name}__security_group"
  description = "security group for terraform instance"
  rule {
    from_port = 22    to_port = 22
    ip_protocol = "tcp"    cidr = "0.0.0.0/0"  } rule {
    from_port = 80    to_port = 80    ip_protocol = "tcp"
    cidr = "0.0.0.0/0"  } rule {
    from_port = 443
    to_port = 443    ip_protocol = "tcp"    cidr =
    "0.0.0.0/0"  } rule {
    from_port = -1    to_port
    = -1    ip_protocol = "icmp"    cidr = "0.0.0.0/0"
  }
}

```

```

!!! #На этом этапе должен быть установлен и инициализирован
openstackcli #Для вывода списка с image_name, image_id openstack
image list > image.txt #В файле найти нужные по заданию ОС
#Для вывода списка flavor_name

```

!!!

```
resource "openstack_blockstorage_volume_v2" "<имя ресурса-тома>" {  #
Название диска    name = "storage"

    # Тип создаваемого диска    volume_type = "dp1"

    # Размер диска    size = "10"

    # uuid индикатор образа, в примере используется CentOS-8.4-202107    image_id
= "c9b7a469-a7ed-4119-b840-fd5169ee4348" }

resource "openstack_compute_instance_v2" "<имя инстанса>" {  #
Название создаваемой VM    name = "<имя по заданию>"

    # Имя и uuid образа с ОС    image_name = "CentOS-8.4-202107"
image_id = "c9b7a469-a7ed-4119-b840-fd5169ee4348"

    # Конфигурация инстанса    flavor_name =
"Basic-1-2-40"

    # Публичный ключ для доступа    key_pair =
openstack_compute_keypair_v2.ssh.name

    # Указываем, что при создании использовать config drive    # Без этой опции
VM не будет создана корректно в сетях без DHCP    config_drive = true

    # Присваивается security group для VM    security_groups = [
openstack_compute_secgroup_v2.rules.name
    ]
```

```

# В данном примере используется сеть ext-net - публичная сеть
network {      name = "<имя сети>"    }

# Блочное устройство  block_device {      uuid =
openstack_blockstorage_volume_v2.volume.id
boot_index = 0      source_type = "<имя ресурса-тома>"
destination_type = "<имя ресурса-тома>"
delete_on_termination = true    }
}

#В этой же директории создать файл output.tf, в котором напомним команду для
#вывода ip-адресов созданных VM

output "instance_ip_addr" {  value =
openstack_compute_instance_v2.<имя инстанса>.access_ip_v4
}

```

5. Создайте балансировщик нагрузки.
2. На локальной машине **PC** создайте скрипт запуска сценария автоматизации развёртывания облачной инфраструктуры.
 1. В качестве рабочей директории используйте путь **/home/skill39/cloud/**.

```

cd /home/skill39/cloud
terraform init

```

Используйте файл **cloud.conf** для указания настроек для подключения к облачному провайдеру.

```

mkdir -p /home/skill39/cloud mv <откуда>/cloud.conf
/home/skill39/cloud/cloud.conf

```

2. Скрипт должен выполняться из любой директории без явного указания пути к исполняемому файлу.

```

#Перенести cloud.conf в /usr/bin

```

3. Выполнение задания ожидается с использованием инструментов Terraform и/или OpenStack CLI. Однако, вы вправе выбрать другие инструменты, не противоречащие условиям задания и правилам соревнования.