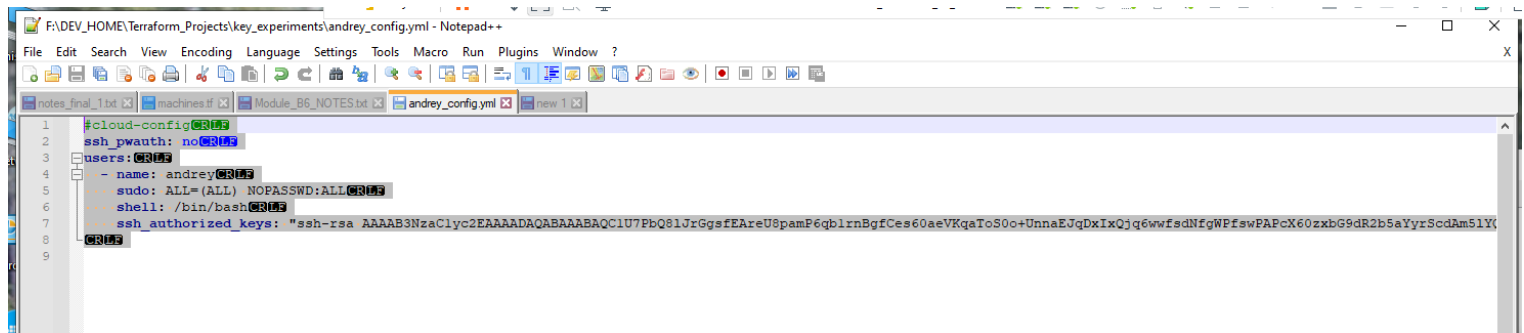


# Report B6 Project

## andrey\_config.yml



## machines.tf

```
# ----- VARIABLES
variable "zone" {
  description = "Use specific availability zone" # Опционально описание переменной
  type        = string                        # Опционально тип переменной
  default      = "ru-central1-a"              # Опционально значение по умолчанию для переменной
}
variable "cloud_id" {
  type        = string                        # Опционально тип переменной
  default      = "b1gfdopk51c4d5reva85"      # Опционально значение по умолчанию для переменной
}
variable "folder_id" {
  type        = string                        # Опционально тип переменной
  default      = "b1gug0h1o834u3niipmr"      # Опционально значение по умолчанию для переменной
}
variable "cloud_key_file" {
  type        = string                        # Опционально тип переменной
  default      = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key.json"      # Опционально значение по умолчанию для переменной
}
variable "ssh_key_file" {
  type        = string                        # Опционально тип переменной
  default      = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_key.pub"
}
variable "config_file" {
  type        = string                        # Опционально тип переменной
  default      = "F:/DEV_HOME/Terraform_Projects/key_experiments/andrey_config.yml"
}

# ----- PROVIDER
terraform {
  required_providers {
    yandex = {
      source = "yandex-cloud/yandex"
      version = "0.70.0" # Фиксируем версию провайдера
    }
  }
}

# Документация к провайдеру тут https://registry.terraform.io/providers/yandex-cloud/yandex/latest/docs#configuration-reference
# Настраиваем the Yandex.Cloud provider
provider "yandex" {
  service_account_key_file = var.cloud_key_file
}
```

```
cloud_id = var.cloud_id
folder_id = var.folder_id
zone     = var.zone # зона, в которая будет использована по умолчанию
}

# ----- WORKING CODE

data "yandex_compute_image" "ubuntu_2004" {
  family = "ubuntu-2004-lts-gpu"
}

data "yandex_compute_image" "centos_8" {
  family = "centos-8"
}

resource "yandex_compute_instance" "vm1" {
  name      = "vm1"

  resources {
    cores = 2
    memory = 2
  }

  boot_disk {
    initialize_params {
      image_id = data.yandex_compute_image.ubuntu_2004.id
    }
  }

  network_interface {
    subnet_id = yandex_vpc_subnet.subnet-1.id
    nat       = true
  }

  metadata = {
    ssh-keys = "${file(var.ssh_key_file)}"
    user-data = file(var.config_file)
  }
}

resource "yandex_compute_instance" "vm2" {
  name      = "vm2"

  resources {
    cores = 2
    memory = 2
  }

  boot_disk {
    initialize_params {
      image_id = data.yandex_compute_image.ubuntu_2004.id
    }
  }

  network_interface {
    subnet_id = yandex_vpc_subnet.subnet-1.id
    nat       = true
  }

  metadata = {
    ssh-keys = "${file(var.ssh_key_file)}"
```

```

        user-data = file(var.config_file)
    }
}

resource "yandex_compute_instance" "vm3" {
    name      = "vm3"

    resources {
        cores = 2
        memory = 2
    }

    boot_disk {
        initialize_params {
            image_id = data.yandex_compute_image.centos_8.id
        }
    }

    network_interface {
        subnet_id = yandex_vpc_subnet.subnet-1.id
        nat       = true
    }

    metadata = {
        ssh-keys = "${file(var.ssh_key_file)}"
        user-data = file(var.config_file)
    }
}

resource "yandex_vpc_network" "network-1" {
    name = "network1"
}

resource "yandex_vpc_subnet" "subnet-1" {
    name      = "subnet1"
    zone      = "ru-central1-a"
    network_id = yandex_vpc_network.network-1.id
    v4_cidr_blocks = ["192.168.10.0/24"]
}

output "external_ip_address_vm1" {
    value = yandex_compute_instance.vm1.network_interface.0.nat_ip_address
}
output "external_ip_address_vm2" {
    value = yandex_compute_instance.vm2.network_interface.0.nat_ip_address
}
output "external_ip_address_vm3" {
    value = yandex_compute_instance.vm3.network_interface.0.nat_ip_address
}

output "internal_ip_address_vm1" {
    value = yandex_compute_instance.vm1.network_interface.0.ip_address
}
output "internal_ip_address_vm2" {
    value = yandex_compute_instance.vm2.network_interface.0.ip_address
}
output "internal_ip_address_vm3" {
    value = yandex_compute_instance.vm3.network_interface.0.ip_address
}

```

```
}
```

Облако

default

Каталог

Дашборд каталога

Сервисные аккаунты

Федерации

Уведомления об инцидентах

Операции

Дашборд каталога

Создать ресурс

Compute Cloud

3 BM

3 Диска

Object Storage

1 Бакет

17.74 КБ Занятое место

Virtual Private Cloud

1 Сеть

1 Подсеть

3 Адреса

Cloud DNS NEW

2 Зоны

11 Записей

Каталог

Compute Cloud

Сервис

Виртуальные машины

Диски

Файловые хранилища

Снимки дисков

Образы

Группы виртуальных машин

Группы подкапиталов

Виртуальные машины

Создать ВМ

Фильтр по имени

Все статусы

Все зоны доступности

<input type="checkbox"/>	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания	
<input type="checkbox"/>	vm2	Running		Intel Broadwell	2	100 %	2 Гб	нет	30 Гб	ru-central1-a	192.168.10.28	130.193.38.205	03 февраля 2022, в 16:22	...
<input type="checkbox"/>	vm1	Running		Intel Broadwell	2	100 %	2 Гб	нет	30 Гб	ru-central1-a	192.168.10.17	130.193.48.55	03 февраля 2022, в 16:00	...
<input type="checkbox"/>	vm3	Running		Intel Broadwell	2	100 %	2 Гб	нет	10 Гб	ru-central1-a	192.168.10.21	130.193.39.35	03 февраля 2022, в 16:22	...

external\_ip\_address\_vm1 = "130.193.48.55"

external\_ip\_address\_vm2 = "130.193.38.205"

external\_ip\_address\_vm3 = "130.193.39.35"

internal\_ip\_address\_vm1 = "192.168.10.17"

internal\_ip\_address\_vm2 = "192.168.10.28"

internal\_ip\_address\_vm3 = "192.168.10.21"

cyukuh: ~/1

/home/infodba/1/inventory - infodba@130.193.48.55 - Editor - WinSCP

Encoding ▾ Color ▾ ?

```
[app]
192.168.10.28  ansible_user=andrey  ansible_ssh_private_key_file=/home/infodba/keys/andrey_key
192.168.10.21  ansible_user=andrey  ansible_ssh_private_key_file=/home/infodba/keys/andrey_key

[database]
192.168.10.17  ansible_user=andrey  ansible_ssh_private_key_file=/home/infodba/keys/andrey_key

[web]
192.168.10.17  ansible_user=andrey  ansible_ssh_private_key_file=/home/infodba/keys/andrey_key

[vm1]
192.168.10.17

[vm2]
192.168.10.28

[vm3]
192.168.10.21
```

```
/etc/ansible/hosts - infodba@130.193.48.55 - Editor - WinSCP
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
# Ex 1: Ungrouped hosts, specify before any group headers.
#green.example.com
#blue.example.com
#192.168.100.1
#192.168.100.10
# Ex 2: A collection of hosts belonging to the 'webservers' group
#[webservers]
#alpha.example.org
#beta.example.org
#192.168.1.100
#192.168.1.110
# If you have multiple hosts following a pattern you can specify
# them like this:
#www[001:006].example.com
# Ex 3: A collection of database servers in the 'dbservers' group
#[dbservers]
#
#db01.intranet.mydomain.net
#db02.intranet.mydomain.net
#10.25.1.56
#10.25.1.57
# Here's another example of host ranges, this time there are no
# leading 0s:
#db-[99:101]-node.example.com
[vm2]
192.168.10.28  ansible_user=andrey  ansible_ssh_private_key_file=/home/infodba/keys/andrey_key
[vm3]
192.168.10.21  ansible_user=andrey  ansible_ssh_private_key_file=/home/infodba/keys/andrey_key
Line: 1/49      Column: 1      Character: 35 (0x23)      Encoding: 1252 (ANSI - La
```

## Playbook.yml

```
---
- hosts: app
  user: infodba
  become: yes
  tasks:
    - name: " Installing Docker latest version"
      yum:
        name: docker-ce
        state: present
    - name: docker
      service:
        name: "docker"
        state: started
        enabled: yes
- hosts: database
```

```
user: infodba
become: yes
tasks:

- name: " Installing postgresql latest version"
  yum:
    name: postgresql
    state: present

- name: postgresql
  service:
    name: "postgresql"
    state: started
    enabled: yes
```

