

Module_B9_Project_Report

machines.tf

```
# ----- VARIABLES
variable "zone" {
    # Используем переменную для передачи в конфиг инфраструктуры
    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" "centos" {
    family = "centos-7"
}

resource "yandex_compute_instance" "art_vm" {
    name      = "art-vm"

    resources {
        cores = 4
        memory = 4
    }

    boot_disk {
        initialize_params {
            image_id = data.yandex_compute_image.centos.id
            size     = 50
            type     = "network-hdd"
        }
    }
}

network_interface {
    subnet_id = "e9b625f03tp7kq77e9vo"
```

```

nat      = true
}

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

output "external_ip_address_docker_vm" {
  value = yandex_compute_instance.art_vm.network_interface.0.nat_ip_address
}

output "internal_ip_address_docker_vm" {
  value = yandex_compute_instance.art_vm.network_interface.0.ip_address
}

```

← → ↺ 🏠

console.cloud.yandex.ru/folders/b1gug0h1o834u3niipmr/compute/instances

☰

Yandex Cloud

Поиск по облачным ресурсам

DE default silverstandart

🔍 📢 ⚙️ 🧑

Каталог

Compute Cloud

Сервис

Виртуальные машины

Диски

Файловые хранилища

Снимки дисков

Образы

Группы виртуальных машин

Группы размещений

Операции

Виртуальные машины

Фильтр по имени

Все статусы

Все зоны доступности

☐

Имя

☐

Статус

☐

ОС

☐

Платформа

☐

vCPU

☐

Доля vCPU

☐

RAM

☐

Прерываемая

☐

Размер дисков

☐

Зона доступности

☐

Внутренний IPv4

☐

Публичный IPv4

☐

Дата создания

☐

Идентификатор

⚙️

☐	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания	Идентификатор	⚙️
☐	art-vm	Running	🍷	Intel Broadwell	4	100 %	4 Гб	нет	50 Гб	ru-central1-a	192.168.10.8	51.250.6.151	26 февраля 2022, в 02:24	fhm618a...	⋮
☐	docker-vm	Running	🍷	Intel Broadwell	2	100 %	2 Гб	нет	20 Гб	ru-central1-a	192.168.10.18	51.250.12.113	15 февраля 2022, в 01:16	fhm1261...	⋮

Создать VM

JFrog

Application

- Dashboard
- Artifactory
- Packages
- Builds
- Artifacts
- Distribution
- Pipelines
- Security & Compliance

Tree Simple

- jfrog-support-bundle
- pypl
- artifactory-build-info
- pypl-local
- pypl-remote
- pypl-remote-cache

SET ME UP

Deploy

To deploy packages using setuputils you need to add an Artifactory repository to the `.pyproc` file (usually located in your home directory):

```
1 [distutils]
2 index-servers = local
3 [local]
4 repository: http://51.250.6.151:8081/artifactory/api/pypl/pypl-local
5 username: <USERNAME>
6 password: <PASSWORD>
```

To deploy a python egg to Artifactory, after changing the `.pyproc` file, run the following command:

```
1 python setup.py sdist upload -r local
```

To deploy a python wheel to Artifactory, after changing the `.pyproc` file, run the following command:

```
1 python setup.py bdist_wheel upload -r local
```

where `local` is the index server you defined in `.pyproc`.

Deploy

Trash Can

Total license 7.6.3 rev 7003300 Licensed to me 6 Copyright 2013 jfrog Ltd

The top screenshot shows the JFrog Platform 'Repositories' page. It features a sidebar with navigation options like Administration, Repositories, Layouts, Identity and Access, Security, Licenses, General, and Proxies. The main content area has tabs for Local, Remote, Virtual, and Distribution. Under the 'Virtual' tab, there is a '1 Repository' section with a filter input. Below this is a table with columns: Repository Key, Type, Selected Repositories, and Recalculate Index. The table contains one entry: 'pypi' with type 'PyPI' and '2 | pypi-local, pypi-remote' selected repositories.

The bottom screenshot shows the 'Packages' page for the 'log_everywhere' package. It displays the package name, latest version (0.0.1), and a description: 'Logging package to enable writing all program information with corresponding datetime to multiple, DIFFERENT files safely when running multiple threads (multi-threading)'. It also shows statistics: Scanned Xray, 1 Versions, and 1 Downloads.

pip.conf

```
[global]
index-url = http://admin:infodba1@51.250.6.151:8081/artifactory/api/pypi/pypi-local/simple
```

The screenshot shows a WinSCP editor window with the file path `/home/andrey/.pip/pip.conf`. The file content is the same as the previous block:

```
[global]
index-url = http://admin:infodba1@51.250.6.151:8081/artifactory/api/pypi/pypi-local/simple
```

Result

The screenshot shows a terminal window with the following output:

```
[andrey@f6m618aa823m5trvg510 ~]$ pip install log_everywhere --trusted-host 51.250.6.151
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: http://admin:****@51.250.6.151:8081/artifactory/api/pypi/pypi-local/simple
warning: missing-index-doctype
x The package index page being used does not have a proper HTML doctype declaration.
  Problematic URL: http://51.250.6.151:8081/artifactory/api/pypi/pypi-local/simple/log-everywhere/
note: This is an issue with the page at the URL mentioned above.
hint: You might need to reach out to the owner of that package index, to get this fixed. See https://github.com/pypa/pip/issues/10825 for context.
Collecting log_everywhere
  Downloading http://51.250.6.151:8081/artifactory/api/pypi/pypi-local/log_everywhere-0.0.1.tar.gz (9.7 kB)
  Preparing metadata (setup.py) ... done
Using legacy 'setup.py install' for log_everywhere, since package 'wheel' is not installed.
Installing collected packages: log_everywhere
  Running setup.py install for log_everywhere ... done
Successfully installed log_everywhere-0.0.1
[andrey@f6m618aa823m5trvg510 ~]$
```