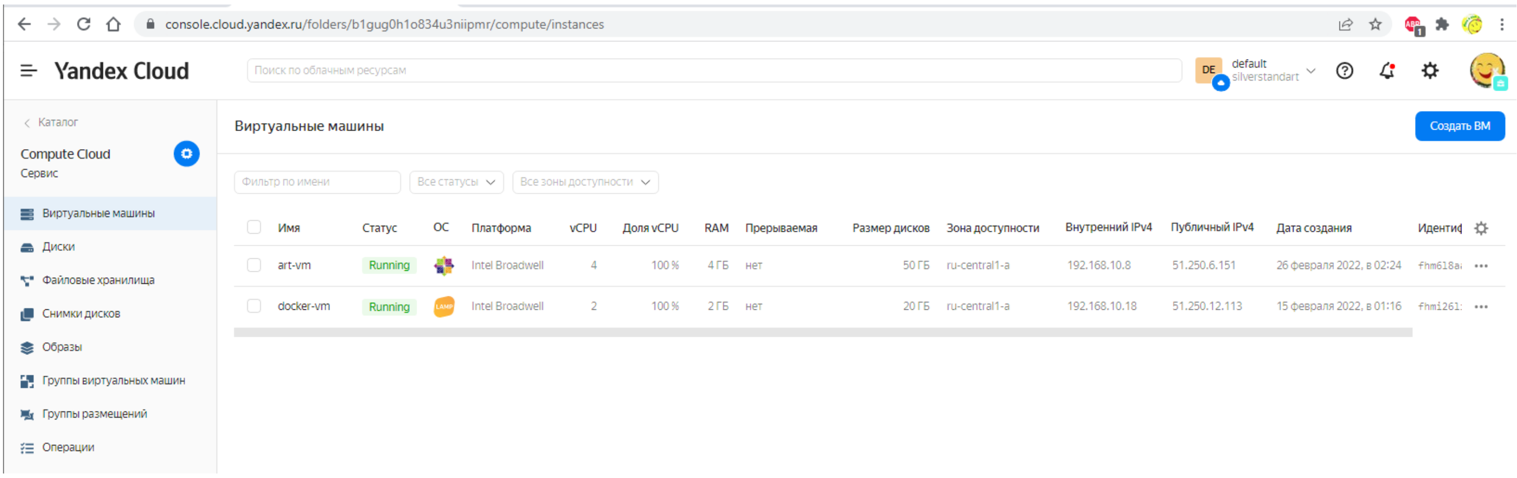
Module_B9_Project_Report

machines.tf

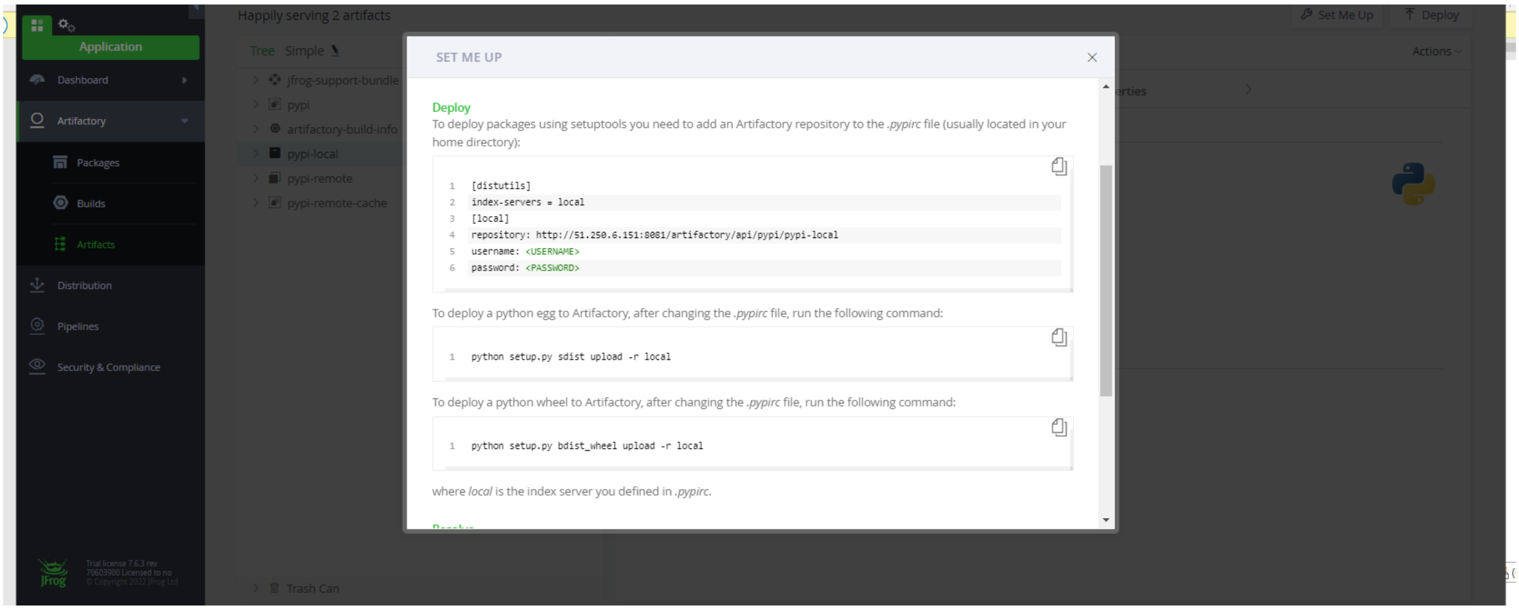
```
# -------------------------------------------------------- VARIABLES
variable "zone" {                              # Используем переменную для передачи в конфиг инфраструктуры
  description = "Use specific availability zone" # Опционально описание переменной
  type       = string                          # 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
}
```



Виртуальные машины

Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания	Идентиф
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	LAMP	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	fhmi261:

JFrog



SET ME UP

Deploy

To deploy packages using setuptools you need to add an Artifactory repository to the *.pypirc* file (usually located in your home directory):

```
[distutils]
index-servers = local
[local]
repository: http://51.250.6.151:8081/artifactory/api/pypi/pypi-local
username: <USERNAME>
password: <PASSWORD>
```

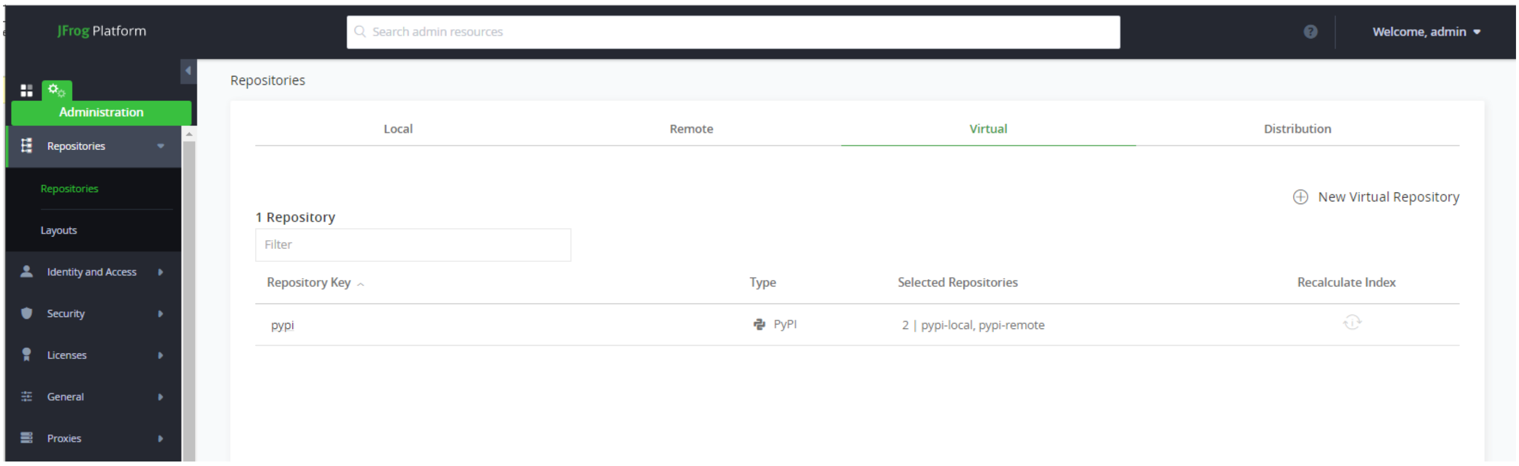
To deploy a python egg to Artifactory, after changing the *.pypirc* file, run the following command:

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

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

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

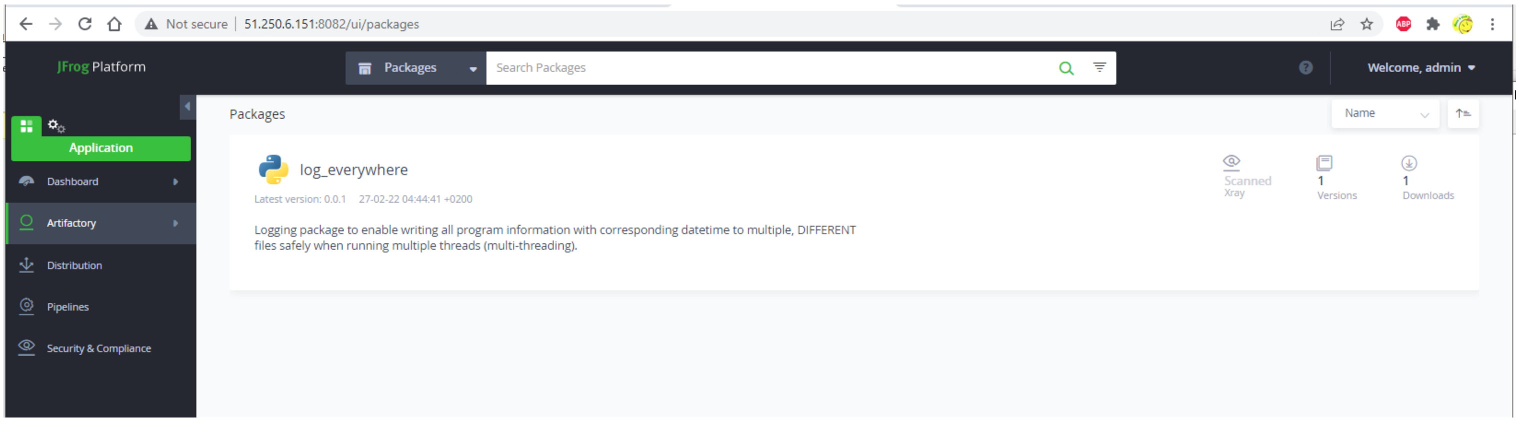
where *local* is the index server you defined in *.pypirc*.



Repositories

1 Repository

Repository Key	Type	Selected Repositories	Recalculate Index
pypi	PyPI	2 | pypi-local, pypi-remote	



Packages

log_everywhere

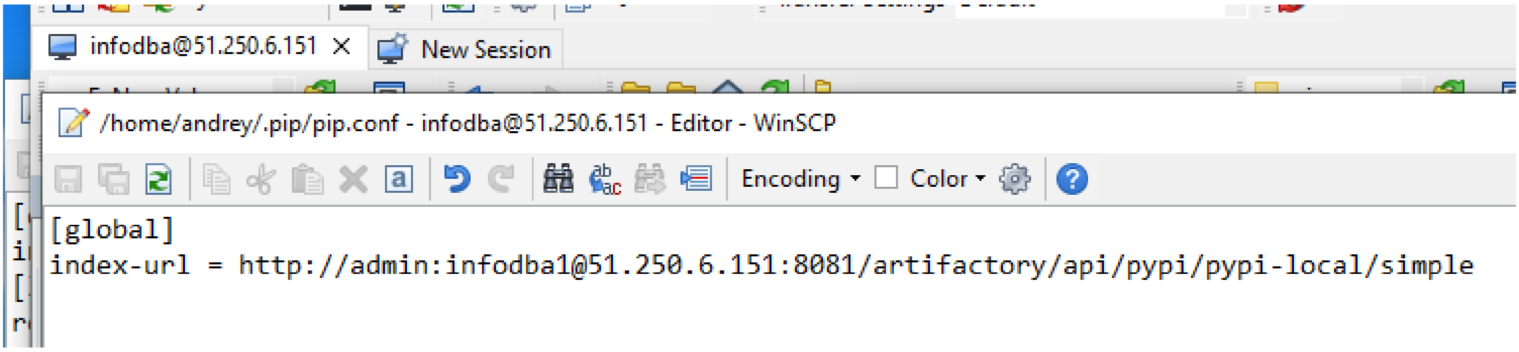
Latest version: 0.0.1 27-02-22 04:44:41 +0200

Logging package to enable writing all program information with corresponding datetime to multiple, DIFFERENT files safely when running multiple threads (multi-threading).

Scanned Xray 1 Versions 1 Downloads

pip.conf

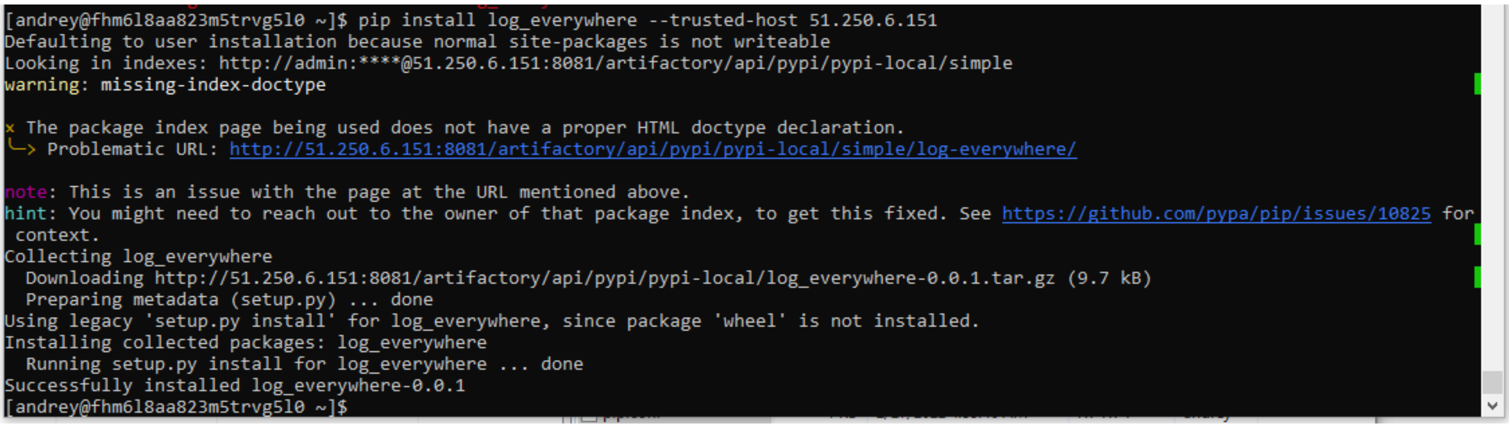
```
[global]
index-url = http://admin:infodba1@51.250.6.151:8081/artifactory/api/pypi/pypi-local/simple
```



/home/andrey/.pip/pip.conf - infodba@51.250.6.151 - Editor - WinSCP

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

Result



```
[andrey@fhm6l8aa823m5trvg5l0 ~]$ 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

× 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@fhm6l8aa823m5trvg5l0 ~]$
```