

Module_B11_HWProject_Report

Что нужно сделать

1. Создать в Я.Облаке виртуальную машину со следующими характеристиками: *2vCPU, 2GB, RAM, 20GB, HDD.* DONE
2. Поднять на этой машине *CI*-сервер на ваш выбор. DONE
3. Создать репозиторий (*github/gitlab*/проч. на ваше усмотрение) и создать там файл *index.html*. DONE
4. Настроить *CI*: DONE

• Запускающий контейнер с *nginx* (версия на ваше усмотрение) с пробросом порта 80 в порт 9889 хостовой системы. При обращении к *nginx* в контейнере по *HTTP*, *nginx* должен выдавать измененный файл *index.html*. DONE

• Проверяющий код ответа запущенного контейнера при *HTTP*-запросе (код должен быть 200). NOT DONE

• Сравнивающий *md5*-сумму измененного файла с *md5*-суммой файла, отдаваемого *nginx* при *HTTP*-запросе (суммы должны совпадать). NOT DONE

• Триггер для старта *CI*: внесение изменений в созданный вами файл *index.html* из п.3. В случае выявления ошибки (в двух предыдущих пунктах), должно отправляться оповещение вам в удобный канал связи — *Telegram/Slack/email*. Текст оповещения — на ваше усмотрение. NOT DONE

• После выполнения *CI* созданный контейнер удаляется. NOT DONE
5. Прислать ментору написанный вами для *CI* код (или скрин с описанием джоба), а также ссылку на репозиторий и на развернутую *CI*-систему. DONE

Step 1

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_1804" {
  family = "ubuntu-1804-lts"
}

resource "yandex_compute_instance" "ci_server" {
  name      = "ci-server"

  resources {
    cores = 2
    memory = 2
  }

  boot_disk {
    initialize_params {
      image_id = data.yandex_compute_image.ubuntu_1804.id
      size = 20
      type = "network-hdd"
    }
  }

  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_ci_server" {
  value = yandex_compute_instance.ci_server.network_interface.0.nat_ip_address
}

output "internal_ip_address_ci_server" {
  value = yandex_compute_instance.ci_server.network_interface.0.ip_address
}
```

← → ↺ 🏠

console.cloud.yandex.ru/folders/b1gug0h1o834u3niipmr/compute/instance/fhmaadvor6snmm1ihkmd/overvi...

DE default silverstandart

🔍 🔔 ⚙️ 😊

Yandex Cloud

Поиск по облачным ресурсам

< Compute Cloud

ci-server
Виртуальная машина

Обзор

Диски

Файловые хранилища

Операции

Мониторинг

Серийная консоль

Последовательный порт

Документация

Создать виртуальную машину

Тарифы Compute Cloud

Настройка синхронизации

Обзор

Изменить VM

Остановить

Перезапустить

Удалить

Идентификатор.....fhmaadvor6snmm1ihkmd

Статус.....Running

Имя.....ci-server

Дата создания.....13 марта 2022, в 01:57

Внутренний FQDN.....fhmaadvor6snmm1ihkmd.auto.internal

Зона доступности.....ru-central1-a

Ресурсы

Платформа.....Intel Broadwell

Гарантированная доля vCPU.....100%

vCPU.....2

RAM.....2 ГБ

Объём дискового пространства.....20 ГБ

Сеть

Сетевой интерфейс

Внутренний IPv4.....192.168.10.23

Публичный IPv4.....84.252.131.29

Подсеть.....subnet1

Настройки DNS для внутренних адресов

Зона	FQDN	TTL
Нет данных		

Step 2

hub.docker.com

docker hub Search for great content (e.g., mysql) Explore Repositories Organizations Help Upgrade silverstandart

silverstandart Search by repository name Create Repository

silverstandart / wrapper_lower Last pushed: an hour ago	Not Scanned	0	13	Public
silverstandart / wrapper Last pushed: 3 days ago	Not Scanned	0	46	Public
silverstandart / simple_nginx Last pushed: 8 days ago	Not Scanned	0	13	Public
silverstandart / test-repo Last pushed: a month ago	Not Scanned	0	3	Public

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

Create an Organization
Manage Docker Hub repositories with your team

docker community
ALL HANDS #5

https://hub.docker.com/layers/198385645/silverstandart/simple_nginx/latest/images/sha256-1a763cbd30ef4dbc7f8e3fa2e6670fd726f4bddb0ef58868a243c0cb8b35cde1?context=repo&tab=layers

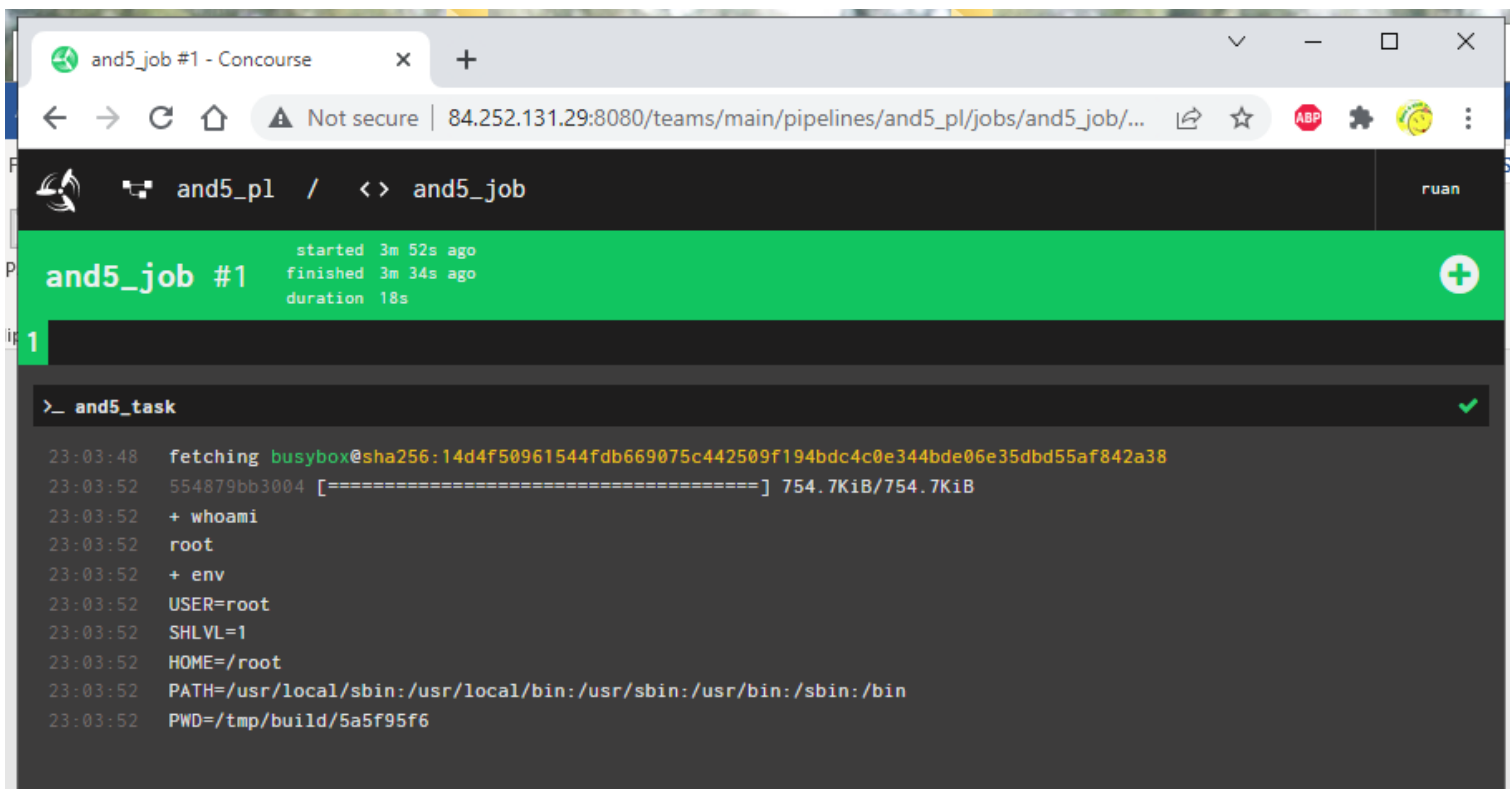
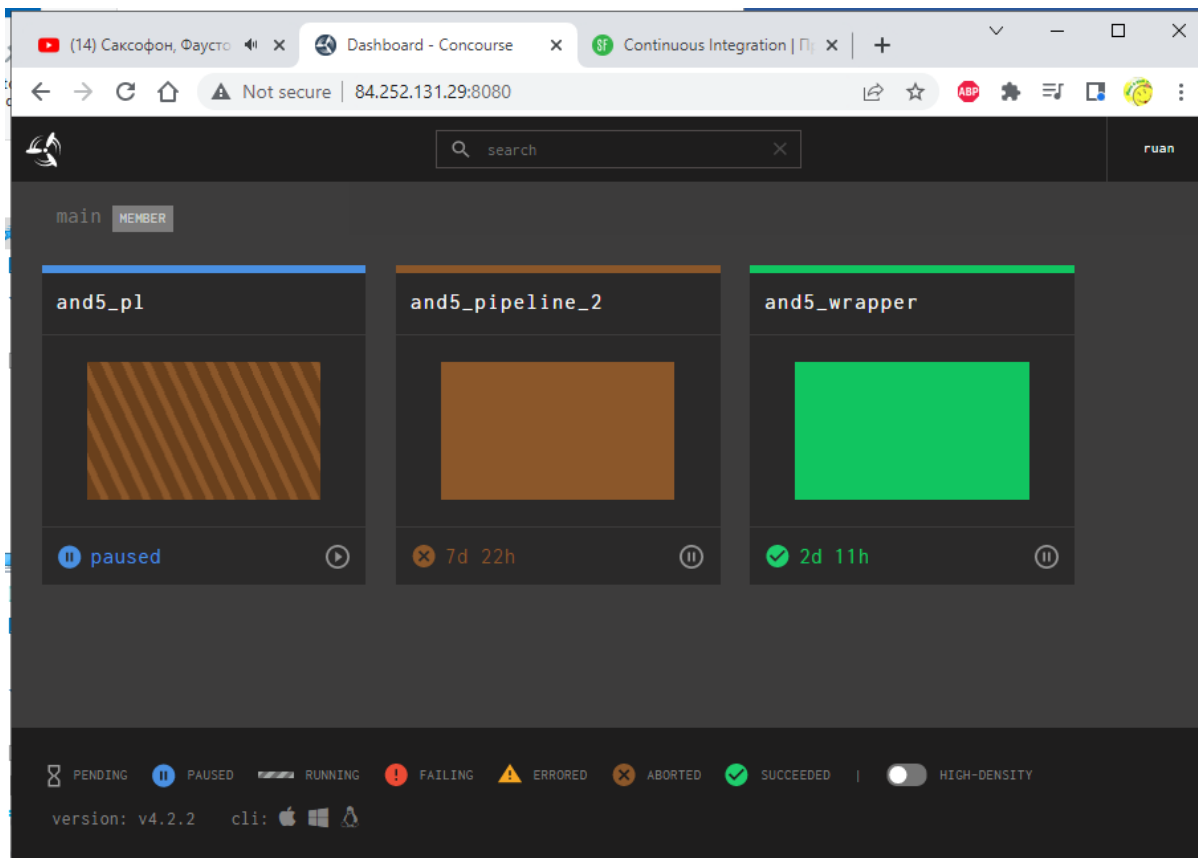
(14) Саксофон, Фаусто and5_pl - Concourse Continuous Integration | Not secure | 84.252.131.29:8080/teams/main/pipelines/and5_pl

and5_pl ruan

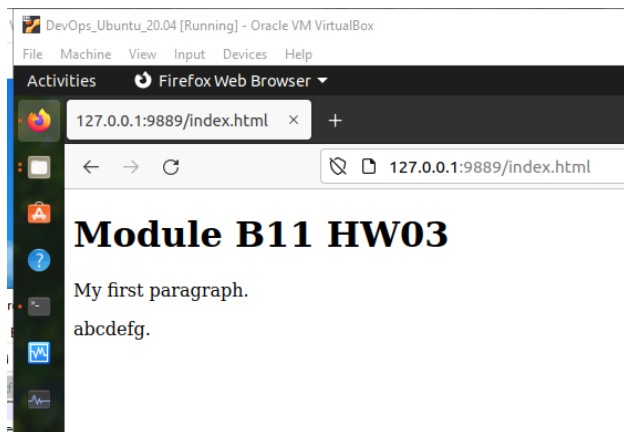
and5_resource_git — and5_job

- succeeded
- errored
- aborted
- paused
- failed
- pending
- started
- dependency
- dependency (trigger)

cli: version: v4.2.2



```
DevOps_Ubuntu_20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal maalis 19 01:23
and5_pipeline.yml
~/DEV_HOME/1_fly
1 ---
2 resources:
3 - name: and5_resource_git
4   type: git
5   source:
6     uri: https://github.com/silverstandart/module_11_HW03.git
7     branch: main
8
9 jobs:
10 - name: and5_job
11   public: true
12   plan:
13     - get: and5_resource_git
14       trigger: true
15     - task: and5_task
16       config:
17         inputs:
18           - name: and5_resource_git
19         platform: linux
20         image_resource:
21           type: registry-image
22           source:
23             repository: nginx
24       run:
25         path: sh
26         args:
27           - -exc
28           - whoami && env
29
30
31
andrey@andrey-VirtualBox: ~/DEV_HOME/1_fly$ fly -t and5 set-pipeline -p and5_pl -c ./and5_pipeline.yml --non-interactive
jobs:
  job and5_job has changed:
  name: and5_job
  public: true
  plan:
    - get: and5_resource_git
      trigger: true
    - task: and5_task
      config:
        platform: linux
        image_resource:
          type: registry-image
          source:
            repository: busybox
            repository: nginx
        run:
          path: sh
          args:
            - -exc
            - whoami && env
          inputs:
            - name: and5_resource_git
configuration updated
andrey@andrey-VirtualBox: ~/DEV_HOME/1_fly$ fly -t and5 trigger-job --job and5_pl/and5_job --watch
started and5_pl/and5_job #2
initializing
fetching nginx@sha256:1a763cbd30ef4dbc7f8e3fa2e6670fd726f4bddb0ef58868a243c0cb8b35cde1
ae13dd578326 [=====] 29.9MiB/29.9MiB
6c0ee9353e13 [=====] 24.2MiB/24.2MiB
dca7733b187e [=====] 601b/601b
352e5a6cac26 [=====] 892b/892b
9eaf108767c7 [=====] 664b/664b
be0c016df0be [=====] 1.4KiB/1.4KiB
running sh -exc whoami && env
+ whoami
root
+ env
USER=root
HOME=/root
PKG_RELEASE=1-bullseye
NGINX_VERSION=1.21.6
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
NJS_VERSION=0.7.2
PWD=/tmp/build/5a5f95f6
succeeded
andrey@andrey-VirtualBox: ~/DEV_HOME/1_fly$
```



docker-compose.yml

```
version: '3.9'

services:
  nginx:
    container_name: my_nginx
    image: nginx:latest
    ports:
      - "9889:80"
    volumes:
      - ./www:/var/www
      - ./nginx/dev.conf:/etc/nginx/conf.d/default.conf
    networks:
      my-network:
        ipv4_address: 10.5.0.5
```

```
networks:
  my-network:
    driver: bridge
    ipam:
      config:
        - subnet: 10.5.0.0/16
          gateway: 10.5.0.1
```

and5_pipeline.yml

```
---
resources:
- name: and5_resource_git
  type: git
  source:
    uri: https://github.com/silverstandart/module_11_HW03.git
    branch: main

jobs:
- name: and5_job
  public: true
  plan:
  - get: and5_resource_git
    trigger: true
  - task: and5_task
    config:
      inputs:
        - name: and5_resource_git
      platform: linux
      image_resource:
        type: registry-image
        source:
          repository: nginx
      run:
        path: sh
        args:
          - -cx
          - |
            nginx -v
            whoami
            env
            apt-get update
            apt-get upgrade
            apt-get install -y iproute2
            apt-get install -y net-tools
            apt-get update
            ip a
            nginx -t

            cat /etc/nginx/nginx.conf
            #sleep 600
            netstat -tulpn | grep LISTEN
            cd and5_resource_git
            ls -lah
            docker compose up -d
```


and5_pipeline.yml version 2

```
---
resources:
- name: and5_resource_git
  type: git
  source:
    uri: https://github.com/silverstandart/module_11_HW03.git
    branch: main

jobs:
- name: and5_job
  public: true
  plan:
  - get: and5_resource_git
    trigger: true
  - task: and5_task
    config:
      inputs:
      - name: and5_resource_git
      platform: linux
      image_resource:
        type: registry-image
        source:
          repository: nginx
      run:
        path: sh
        args:
        - -cx
        - |
          nginx -v
          whoami
        env
```

1_docker_run.sh (this is the problem what I had)

```
#!/bin/sh
# -----
#  Publish Wrapper Lower Image with Docker / 2022_03_25 / ANa
# -----

echo -e "\n\n----- Create temp folder wrapper"
mkdir ./wrapper
chmod 777 ./wrapper
cd wrapper

echo -e "\n\n----- Create and5_service.sh"
cat << EOF > ./and5_service.sh
#!/usr/bin/env bash
# ----- and5_service.sh START -----
#  This file defined to be executed as Service in running container
# -----

echo $HOME
```

```
echo $PWD
sleep 10s
```

```
for ((counter=0,i=0; counter==0; i+=1)); do
    echo $counter "-" $i
    sleep 5s
done
```

```
# ----- and5_service.sh END -----
```

```
EOF
```

```
chmod 777 ./and5_service.sh
```

```
cat ./and5_service.sh
```

```
echo -e "\n\n----- Create Dockerfile"
```

```
cat << EOF > ./Dockerfile
```

```
# ----- Dockerfile START -----
```

```
FROM ubuntu:latest
```

```
RUN apt-get update && apt-get install -y
```

```
#RUN apt install apt-utils
```

```
#RUN apt install net-tools
```

```
#RUN apt install -y iproute2
```

```
#RUN netstat -tulpn | grep LISTEN
```

```
RUN mkdir -p /srv/app
```

```
WORKDIR /srv/app/
```

```
COPY ./and5_service.sh /srv/app/
```

```
RUN cat /srv/app/and5_service.sh
```

```
ENTRYPOINT ["bash", "/srv/app/and5_service.sh"]
```

```
#CMD [ "sh", "-c", "echo $HOME" ]
```

```
#CMD bash -C '/srv/app/and5_service.sh';'bash'
```

```
#CMD /srv/app/and5_service.sh ; sleep infinity
```

```
# ----- Dockerfile END -----
```

```
EOF
```

```
cat ./Dockerfile
```

```
echo -e "\n\n----- Build Docker Image as wrapper_lower"
```

```
docker build . -t wrapper_lower --no-cache
```

```
echo -e "\n\n----- Run Docker Image as wrapper_lower"
```

```
docker run -it --name infodba_DEV -v vol:/infodba_VOL --network=host wrapper_lower
```

```
cd ..
```

```
echo -e "\n\n----- Publish Image as silverstandart/wrapper_lower  
into https://hub.docker.com/"
```

```
docker images -a
```

```
docker image ls -qa
```

```
docker tag $(docker image ls -qa wrapper_lower) silverstandart/wrapper_lower
```

```
docker login
```

```
docker push silverstandart/wrapper_lower
```

```
echo -e "\n\n----- Environment INFO"
```

```
echo ""
```

```
docker image ls
echo ""
docker ps
echo ""
docker compose ps
echo ""
docker volume ls
echo ""
docker network ls
echo ""
```

```
echo -e "\n\n----- Remove Docker Garbage as
silverstandart/wrapper_lower"
rm -rf wrapper
docker rm $(docker ps -qa)
sleep 1
docker image rm $(docker images -q)
docker volume rm $(docker volume ls -q)
```