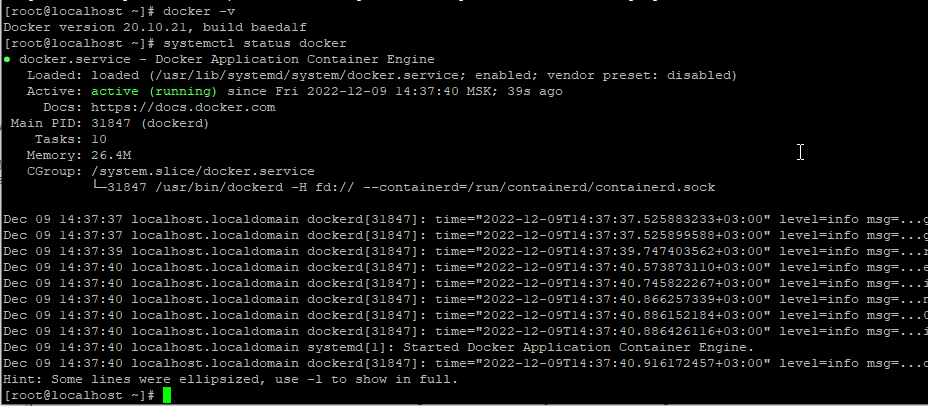
**Docker**

Task 1 Lecture 1

**Install docker and check**



**dockerfile**

FROM ubuntu:20.04

ENV TZ=Europe/Kiev

RUN apt-get update -y

RUN DEBIAN\_FRONTEND="noninteractive" apt-get install -y apache2

RUN echo 'Hi there, what is it?' > /var/www/html/index.html

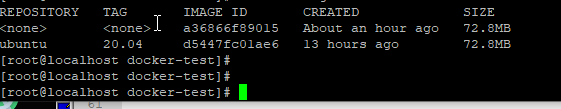
RUN echo 'It is a sone' >> /var/www/html/index.html

CMD ["/usr/sbin/apache2ctl", "-DFOREGROUND"]

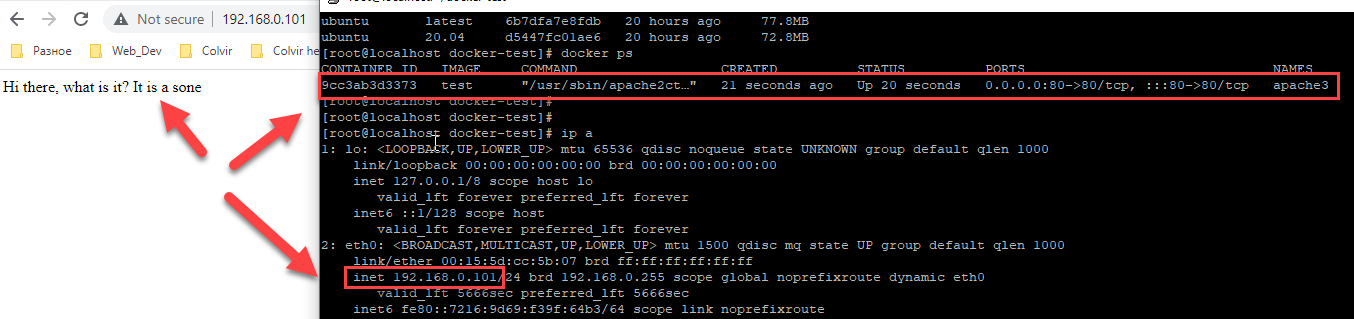
EXPOSE 80

**create image**

docker build -t test /root/docker-test/



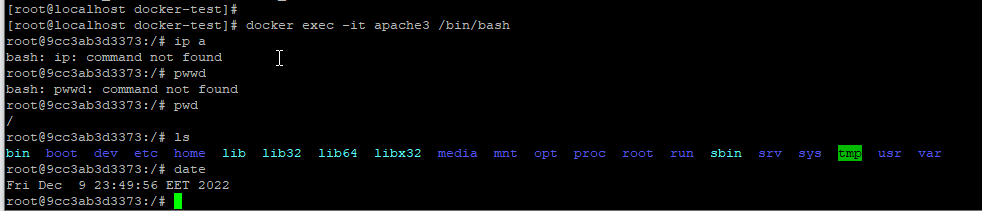
**docker run --name apache3 -p 80:80 -d test**



**docker exec -it apache3 /bin/bash**

**date**

Fri Dec 9 23:49:56 EET 2022



Task 2 Lecture 2

**help link:**

<https://www.digitalocean.com/community/tutorials/how-to-build-and-deploy-a-flask-application-using-docker-on-ubuntu-18-04>

https://www.digitalocean.com/community/tutorials/how-to-use-templates-in-a-flask-application

**nano Dockerfile**

FROM tiangolo/uwsgi-nginx-flask:python3.6-alpine3.7

RUN apk --update add bash nano

ENV STATIC\_URL /static

ENV STATIC\_PATH /var/www/app/static

COPY ./requirements.txt /var/www/requirements.txt

RUN pip install -r /var/www/requirements.txt

**nano main.py**

from app import app

**nano requirements.txt**

Flask==1.0.2

#################################

**nano start.sh**

#!/bin/bash

app="docker.test"

docker build -t ${app} .

docker run -d -p 56733:80 \

--name=${app} \

-v $PWD:/app ${app}

###################################

**nano uwsgi.ini**

[uwsgi]

module = main

callable = app

master = true

#################################

**mkdir app**

**cd app**

##################################

**nano \_\_init\_\_.py**

from flask import Flask

app = Flask(\_\_name\_\_)

from app import views

#######################################

**nano views.py**

from app import app

from flask import render\_template

import random

# list of cat images

images = [

"https://www.deviantart.com/favorisxp/art/Neon-Rider-x-SteelSeries-Live-Wallpaper-884137969",

"https://wallpaper.mob.org/image/nature-doroga-derevya-sneg-gora-vershina-150657.html",

"https://wallpaper.mob.org/image/nature-gora-vershina-zasnezhennii-oblaka-dolina\_kkhumbu-namche-nepal-109549.html"

]

@app.route('/')

def home():

return "Hello world!"

@app.route('/template')

def template():

url = random.choice(images)

return render\_template('home.html', url=url)

###############################

**cd templates**

**nano home.html**

<html>

<head>

<style type="text/css">

body {

background: black;

color: green;

}

div.container {

max-width: 500px;

margin: 100px auto;

border: 20px solid white;

padding: 10px;

text-align: center;

}

h4 {

text-transform: uppercase;

}

</style>

</head>

<body>

<div class="container">

<h4>Welcome to Python App running on K8S cluster in EKS!!</h4>

<h4>Automating using Jenkins, Kubernetes, Docker, GitHub and Pipeline!!!!!</h4>

<br>

<h4>Anything is Possible!!!!!</h4>

<a href={{ url }}>Pictures</a>

</div>

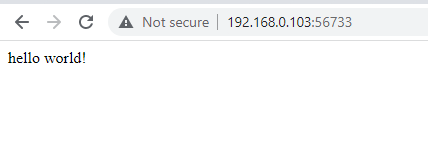
</body

#########################################################

**docker start docker.test**

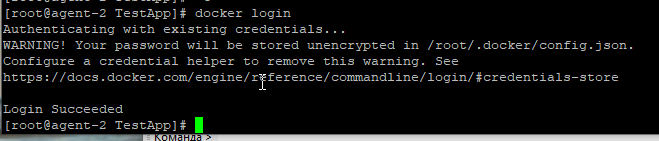
**docker stop docker.test**

**sudo docker stop docker.test && sudo docker start docker.test**

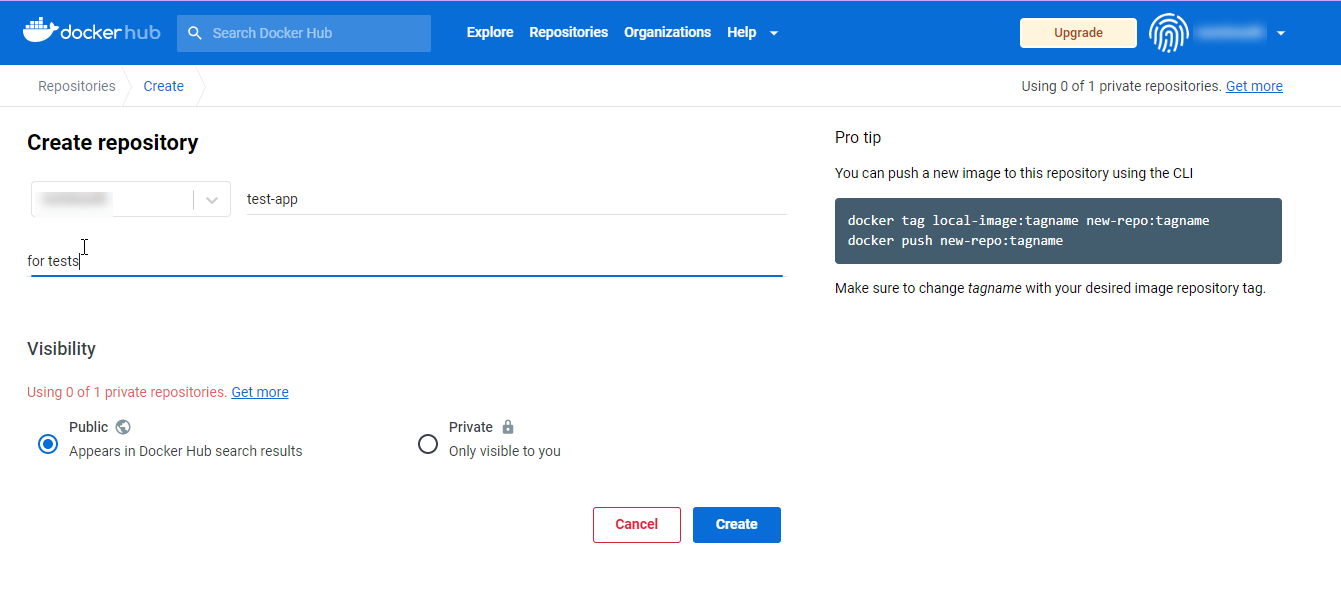




**docker login**



**https://hub.docker.com/**



**docker tag docker.test nominonik/test**

**docker.test** - existing image

**nominonik** - docker username

**test** - name image in docker hub

**docker tag docker.test nominonik/test:latest**

latest - default

[root@agent-2 TestApp]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

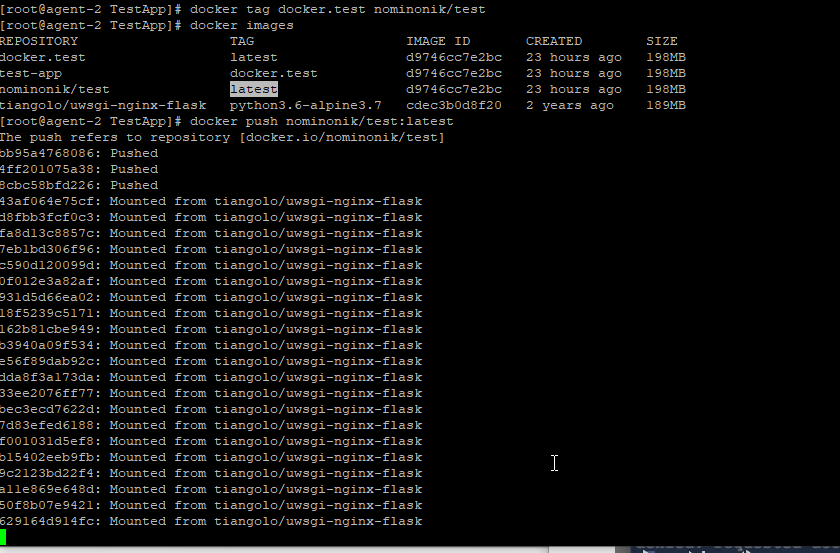
docker.test latest d9746cc7e2bc 23 hours ago 198MB

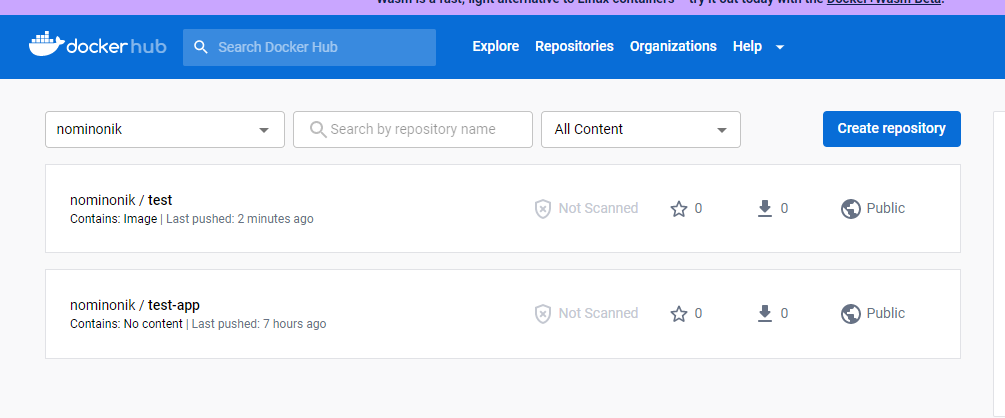
test-app docker.test d9746cc7e2bc 23 hours ago 198MB

nominonik/test latest d9746cc7e2bc 23 hours ago 198MB

tiangolo/uwsgi-nginx-flask python3.6-alpine3.7 cdec3b0d8f20 2 years ago 189MB

**docker push nominonik/test:latest**



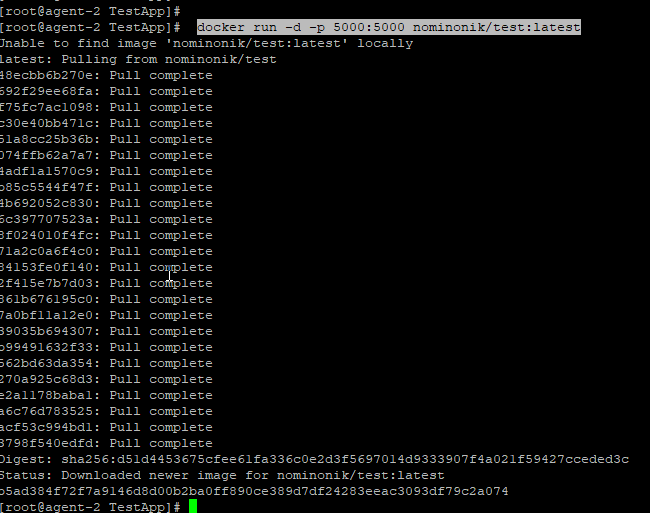


docker rm -f $(docker ps -a -q)

docker rmi -f $(docker images -q)

**Pull image from docker Hub**

docker run -d -p 5000:5000 nominonik/test:latest



Task 3 Lecture 3

curl -SL https://github.com/docker/compose/releases/download/v2.6.1/docker-compose-linux-x86\_64 -o /usr/local/bin/docker-compose

ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

chmod +x /usr/local/bin/docker-compose

docker-compose -v

Docker Compose version v2.6.1

**nano docker-compose.yml:**

version: '3'

services:

# Database

db:

image: mysql:5.7

volumes:

- db\_data:/var/lib/mysql

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: password

MYSQL\_DATABASE: wordpress

MYSQL\_USER: wordpress

MYSQL\_PASSWORD: wordpress

# Wordpress

wordpress:

depends\_on:

- db

image: wordpress:latest

ports:

- '8000:80'

restart: always

volumes: ['./:/var/www/html']

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wordpress

WORDPRESS\_DB\_PASSWORD: wordpress

volumes:

db\_data:

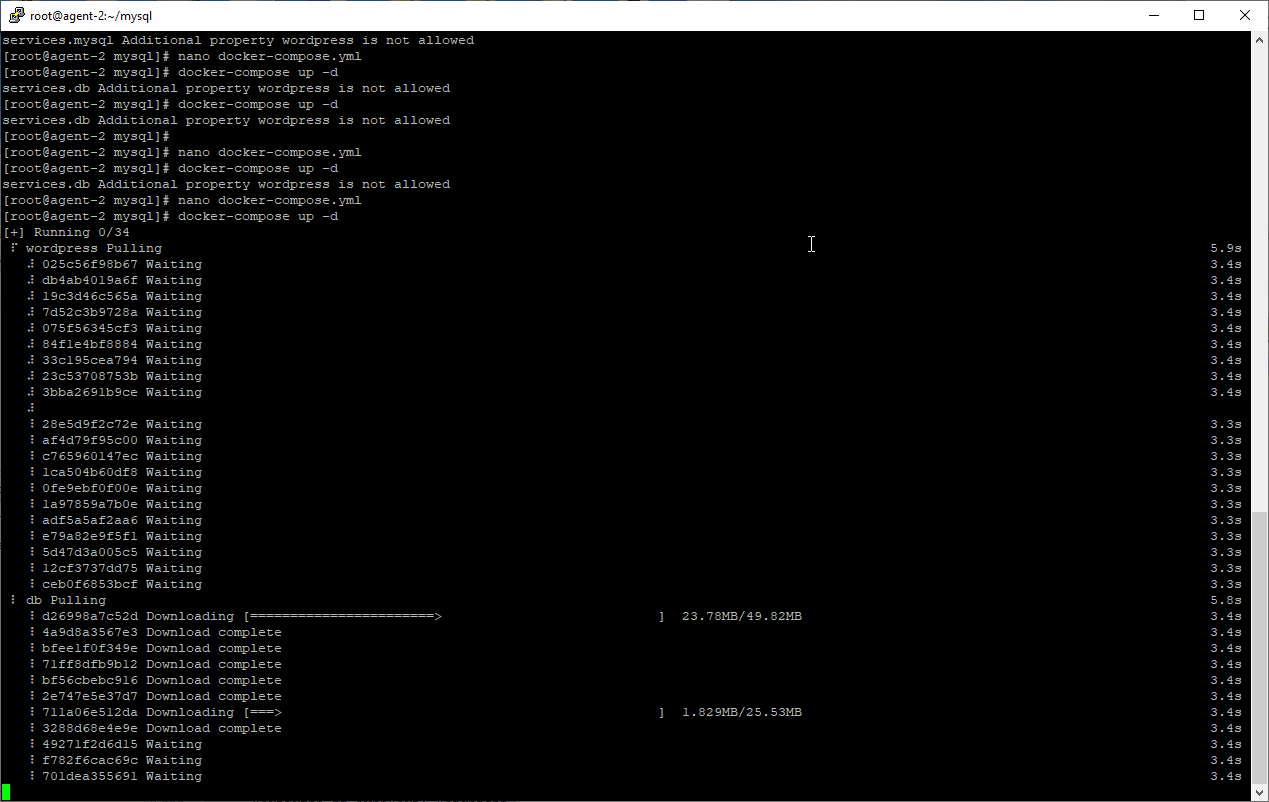
**docker-compose up -d**

ports:

- '8000:80'

8000 - port VM

80 - port in docker

****

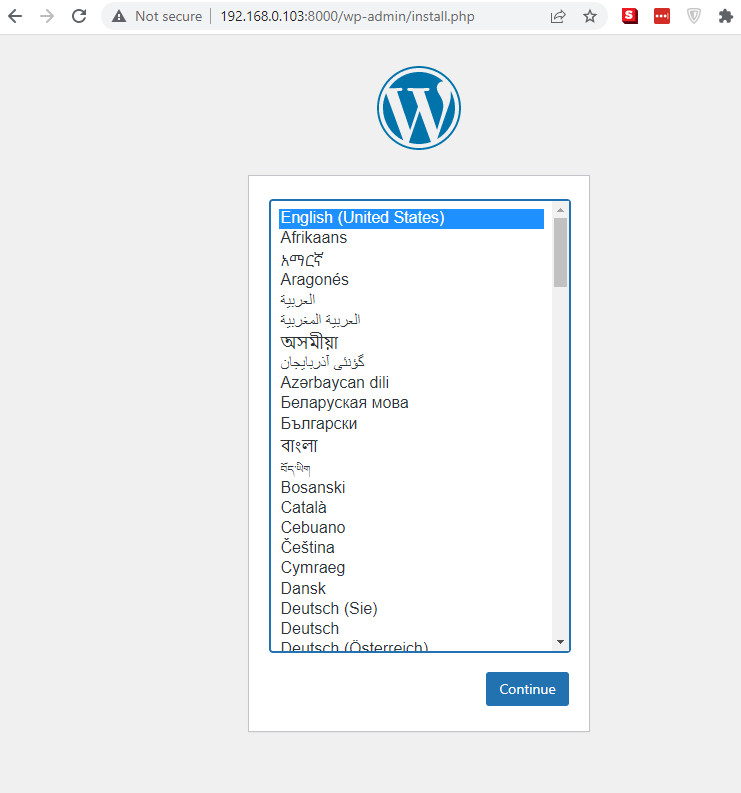
[root@agent-2 mysql]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

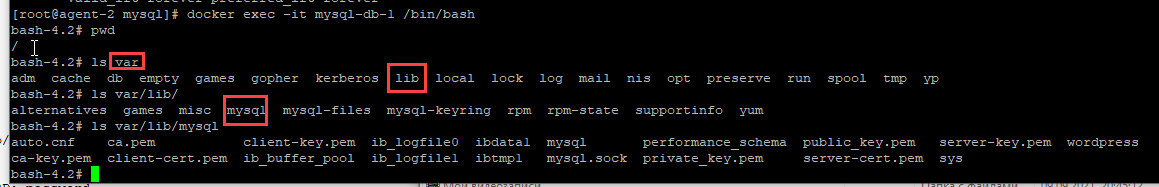
2a42150d4284 wordpress:latest "docker-entrypoint.s…" 3 minutes ago Up 2 minutes 0.0.0.0:8000->80/tcp, :::8000->80/tcp mysql-wordpress-1

2da220dd38ff mysql:5.7 "docker-entrypoint.s…" 3 minutes ago Up 2 minutes 3306/tcp, 33060/tcp mysql-db-1

[root@agent-2 mysql]#

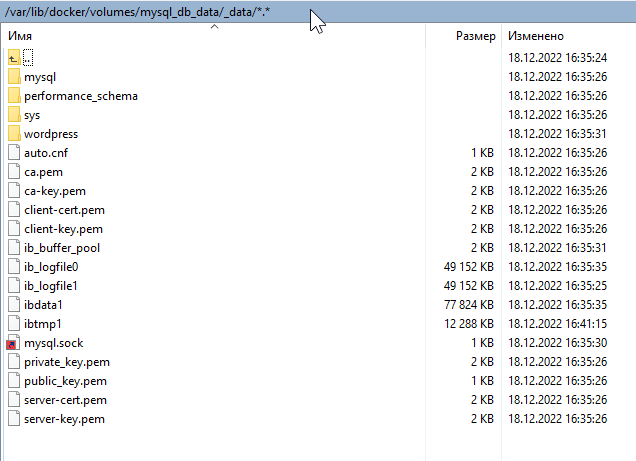
****

**/var/lib/mysql - inside docker vm**

****

**volume on host VM**

**/var/lib/docker/volumes/mysql\_db\_data/\_data**

****

**install pgAdmin**

**nano docker-compose-pg.yml:**

version: "3.7"

services:

pgadmin:

image: dpage/pgadmin4

container\_name: pgadmin

environment:

PGADMIN\_DEFAULT\_EMAIL: sys@colvir.ru

PGADMIN\_DEFAULT\_PASSWORD: sys

PGADMIN\_LISTEN\_PORT: 8080

restart: always

ports:

- "8080:8080"

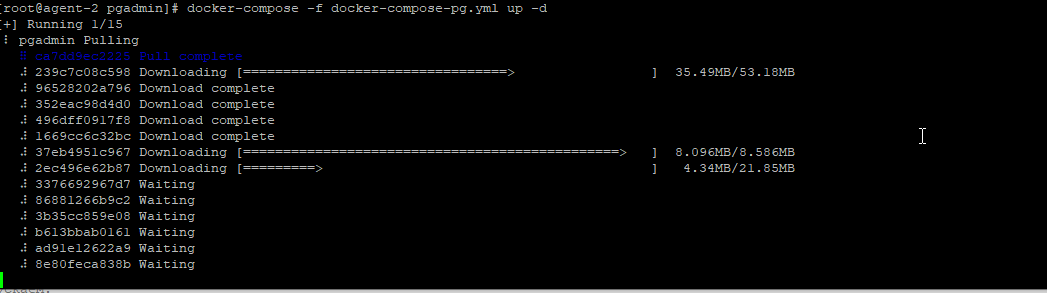
volumes:

- pgadmin:/var/lib/pgadmin

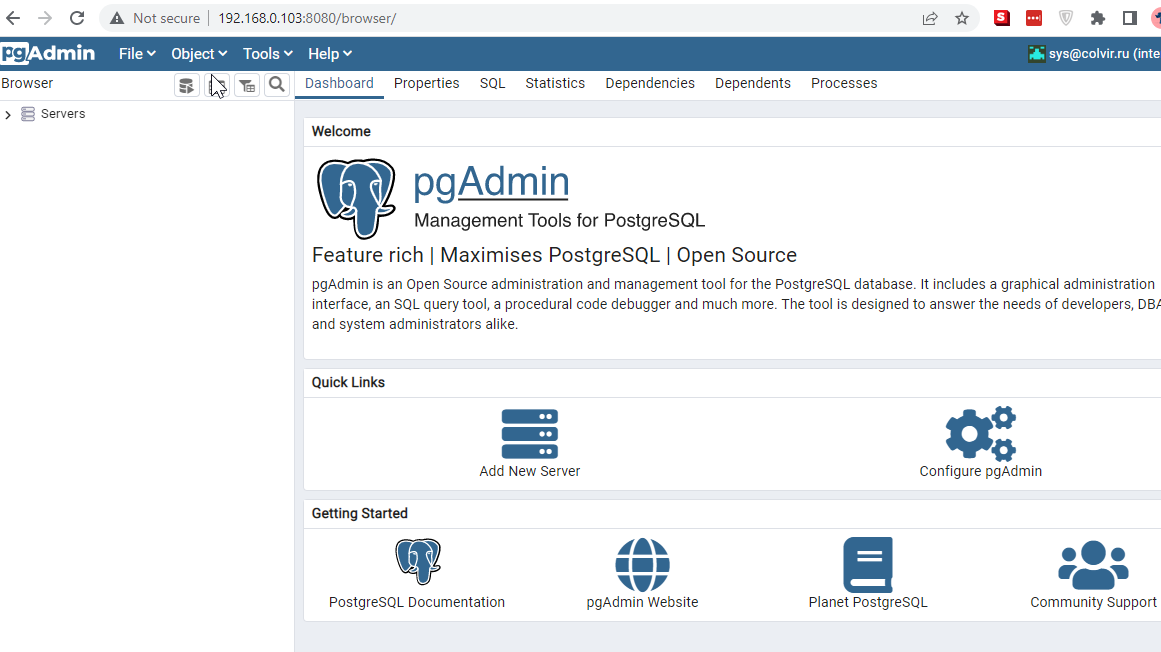
volumes:

pgdata:

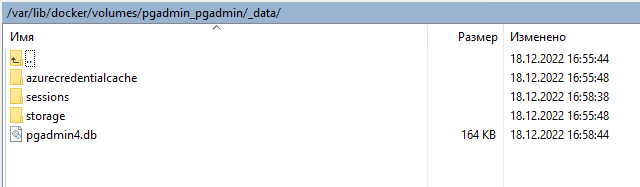
pgadmin:

****

**docker-compose -f docker-compose-pg.yml up -d**

****

**/var/lib/docker/volumes/pgadmin\_pgadmin/\_data**

****