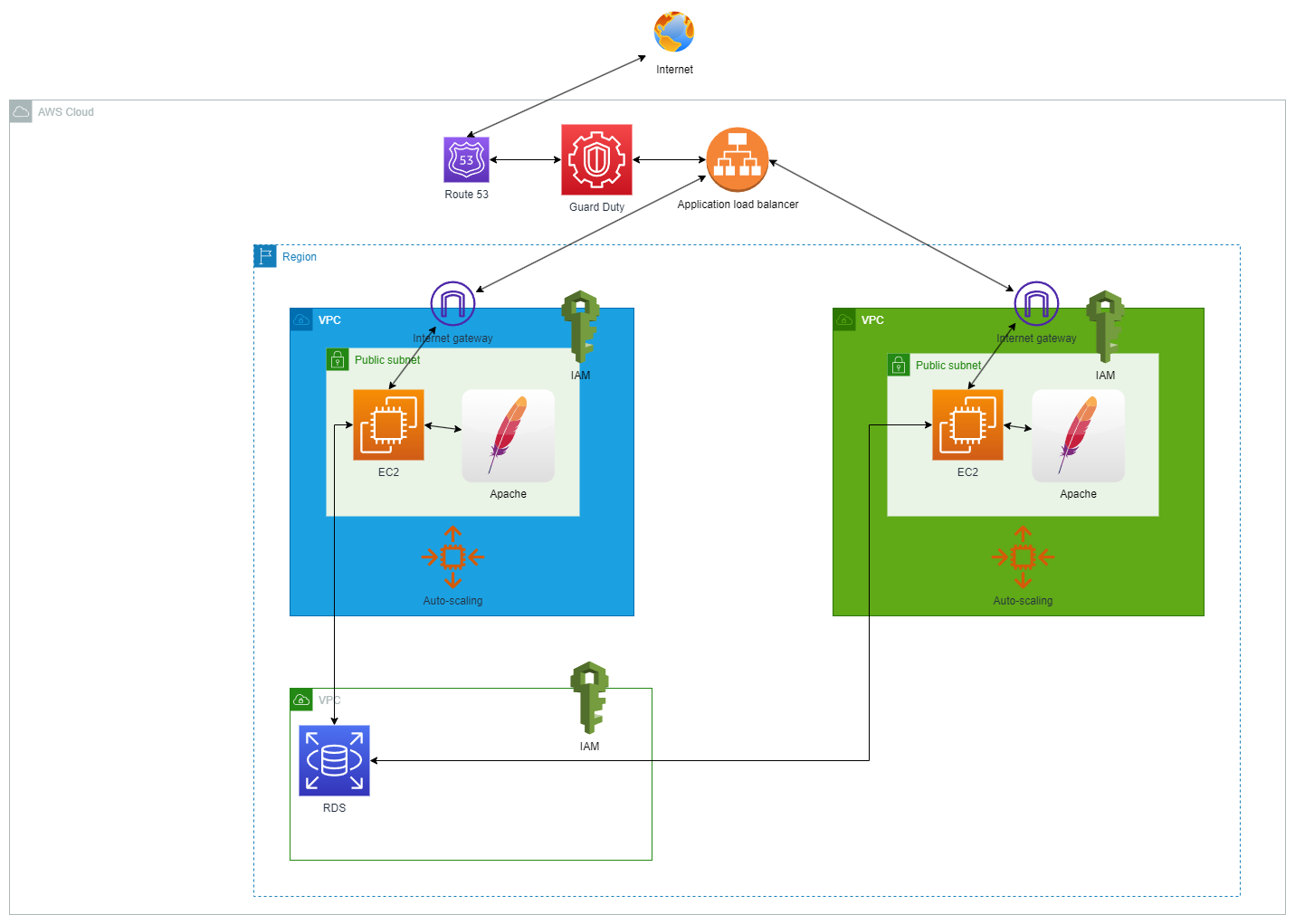
**Theoretical part**

Architectural diagram of the project on AWS Cloud with frontend and backend:



**Picture 1**

My project consists of two separate EC2 instances with public Apache servers(frontend) and private databases(RDS it can be MySql or something else; it`s backend). I`ve made internet gateways so the users outside the VPC have access. There are two VPC`s. Green and Blue. The green one gets updated first. It gets tested and if everything ok the application load balancer switches traffic over to green one. After that blue one gets the same update as a green one and then the application load balancer switches traffic over again to blue. If something wrong it rolls back to previous version and programmers change the code of update. Then i(as devops) deploy and test everything again. If some servers shut down, the auto-scaling group rewrites/remakes the EC2 instance with services. Also EC2 has auto-scaling.

To have secure access to VPC`s and everything inside them i created IAM role with MFA. And also GuardDuty works as protection from internet threats.

**Practical Part**

Solution for deploying a Docker application consisting of 2 components: NGINX + PHP.

Created a EC2 instance in AWS. A docker image was created from Dockerfile.

**Dockerfile code:**

FROM ubuntu:22.04

ENV DEBIAN\_FRONTEND noninteractive

RUN apt update && apt install -y tcl

LABEL maintainer="TarSyr"

RUN apt-get update && \

apt-get install -y -q curl gnupg2

RUN curl http://nginx.org/keys/nginx\_signing.key | apt-key add -

RUN apt-get purge nginx nginx-common nginx-full

RUN apt-get update && \

apt-get install -y -q nginx

RUN apt update

RUN apt install apache2 -y

RUN apt install apache2-utils -y

RUN apt clean

RUN apt-get install software-properties-common -y

RUN add-apt-repository ppa:ondrej/php

RUN apt-get update

RUN apt purge libapache2-mod-php7.1

RUN apt install libapache2-mod-php7.1 -y

RUN a2enmod php7.1

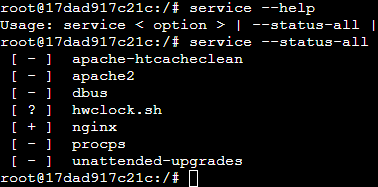
RUN service apache2 restart

EXPOSE 443 80

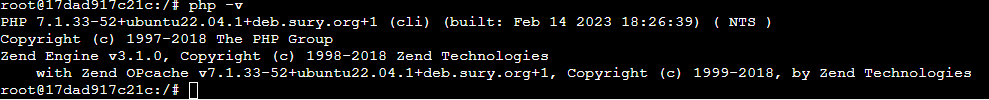
CMD [“apache2ctl”, “-D”, “FOREGROUND”]

CMD ["nginx", "-g", "daemon off;"]

Then i run it. Start the container. Checking services which has to be installed(NGINX+PHP):



**Picture 2** There is NGINX

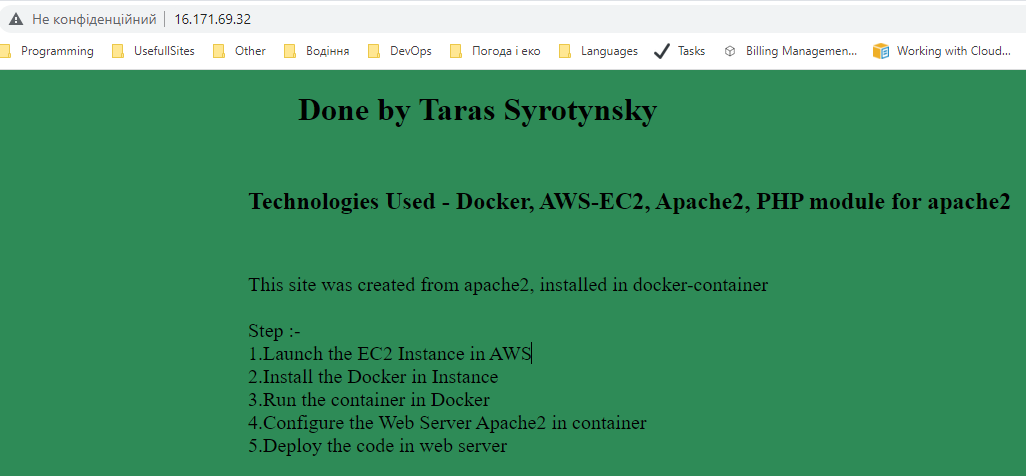


**Picture 3** PHP version 7.1.33-52

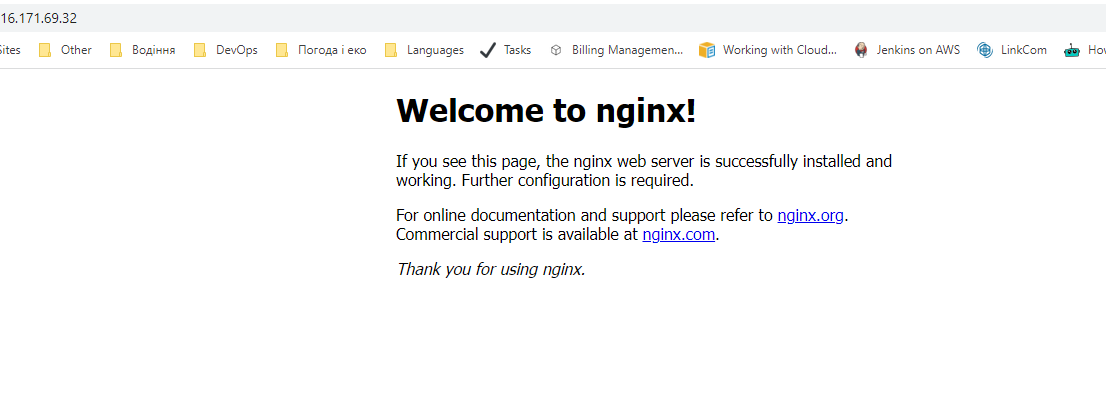
Site is accessible for everyone in the internet. But i didn`t buy and register a domain name yet.

Public IP: <http://16.171.69.32/>

Results:



**Picture 4** Apache



**Picture 5** NGINX