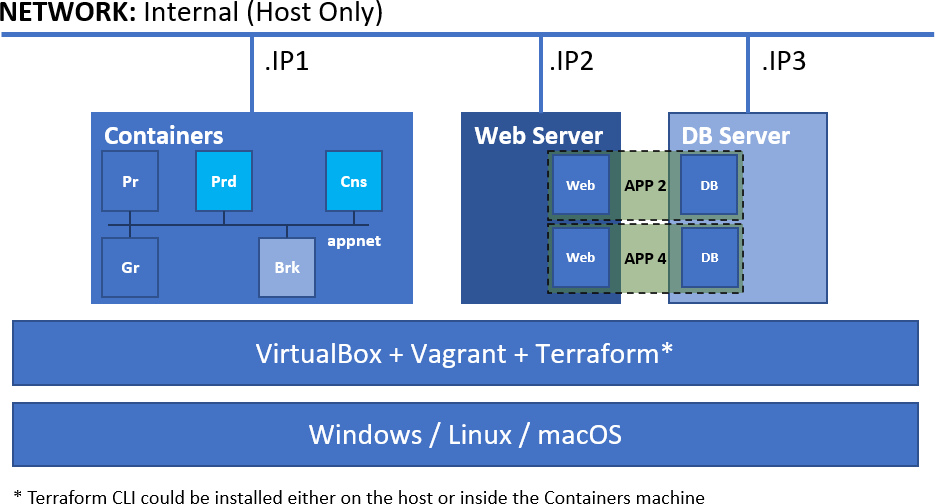
Exam: DevOps #2 2023.04 (2023.06.04)

# Main Goal

You are expected to utilize all or most of the studied products and technologies and create an infrastructure with **three** hosts. Their parameters and distributions are up to you to decide *(considering your free resources and the actual distribution of components)*

The **emphasis** should be on **features** usage **demonstration** versus optimal solution The goal is to have the whole **infrastructure** as a **file** or **set** of **files**

Your solution should look and follow this structure:



# Rules and Guidelines

Be sure to **follow** the **naming conventions** specified in the checklist and in project source files

The tasks execution order should not be derived from the order in which they are listed below. Please note that there are tasks that depend on the successful completion of one or more other tasks

# Proof

You are expected to prepare a short document that outlines the steps you did. It should include all major milestones

If there are any manual steps, you **must** describe them in a free form *(including commands if any)* in the document

There **must be pictures** (one or more) for **all tasks** that are shown like this **(T301, 2 pts)** at least. Any additional pictures are more than welcome. The **current date and time must be visible** on all the pictures

The document, together with all supporting files *(outputs, pictures, configuration files and scripts)*, should be stored in a compressed archive *(32 MB max)* and uploaded not later than **13:30** to:

<https://softuni.bg/trainings/4037/devops-infrastructure-configuration-management-april-2023#lesson-51102>

# Tasks

## Infrastructure as Code (19 pts)

*You are expected to demonstrate knowledge working with* ***Terraform****,* ***Vagrant*** *and* ***VirtualBox***

### Level #1

*Depending on the platform you use you are expected to create the following:*

* (T101, 3 pts) Create a set of **three** machines *(the distribution is up to you)*. Most of the provisioning is expected to be done with the help of configuration management tools *(there is a separate set of tasks)*

### Level #2

*Using* ***Terraform*** *(either on the host or inside the* ***Containers*** *machine) you are expected to implement the following:*

* (T102, 3 pts) Spin up an **Apache Kafka** or **RabbitMQ** *(it is up to you to decide)* single-node cluster (**Brk**)
* (T103, 2 pts) Enable the monitoring of the single-node cluster *(either by enabling a plugin or by running additional container)*
* (T104, 2 pts) Spin up a **discoverer container** (**Prd**) for the **animal-facts** topic/exchange by using the appropriate repository
  + for Apache Kafka – <https://hub.docker.com/repository/docker/shekeriev/kafka-discoverer>
  + for RabbitMQ – <https://hub.docker.com/repository/docker/shekeriev/rabbit-discoverer>
* (T105, 2 pts) Spin up an **observer container** (**Cns**) for the **animal-facts** topic/exchange by using the appropriate repository
  + for Apache Kafka – <https://hub.docker.com/repository/docker/shekeriev/kafka-observer>
  + for RabbitMQ – <https://hub.docker.com/repository/docker/shekeriev/rabbit-observer>
* (T106, 1 pts) Spin up a **Prometheus** instance (**Pr**) and
  + (T107, 2 pts) Set it to collect data from the ***single-node cluster*** *(not the node, but the middleware)*
  + (T108, 2 pts) And to collect data from the ***discoverer application***
* (T109, 2 pts) Spin up a **Grafana** instance (**Gr**) and set it to use the **Prometheus** instance as a data source

*The number and structure of the configurations to spin up the above is up to you to determine*

## Configuration Management (27 pts)

*You are expected to demonstrate knowledge working with two of the studied configuration management solutions. It is up to you to select which two*

### Configuration Management #1

* (T201, 3 pts) Do a basic (installed and running) installation of **Docker** on **VM1**
* (T202, 1 pts) The **user in use** (**vagrant** or another one) must be a member of the **docker** group

### Configuration Management #2

* (T203, 4 pts) Do a basic (installed and running) installation of **Apache** (+PHP +libraries) on **VM2**
* (T204, 3 pts) Add two virtual hosts by port – **8001** and **8002**
* (T205, 4 pts) Deploy both applications (**app2** and **app4**) files to the corresponding folders of the virtual hosts
* (T206, 3 pts) Do a basic (installed and running) installation of **MariaDB** on **VM3**
* (T208, 3 pts) Make sure the service is listening on all interfaces (should be accessible from **VM2**)
* (T207, 4 pts) Deploy applications’ databases
* (T209, 2 pts) Make sure that **VM2** and **VM3** can reach each other by name *Applications (****app2*** *and* ***app4****) can be found here:* [*https://github.com/shekeriev/do2-app-pack*](https://github.com/shekeriev/do2-app-pack) *Deploy the not as containers but following the classical approach*

## Monitoring (4 pts)

*You are expected to demonstrate basic knowledge working with both* ***Prometheus*** *and* ***Grafana***

* **(T301, 2 pts)** Create a simple visualization of a metric of the ***selected middleware***
* **(T302, 2 pts)** Create a simple visualization of one of the metrics (**discovered\_facts\_total** or

**time\_spent\_total**) of the ***discoverer application***

## Applications (10 pts)

*You are expected to manage to do a successful deployment of the three applications*

* **(T401, 4 pts)** Working pair of ***discoverer*** and ***observer***
* Working ***web application #2***

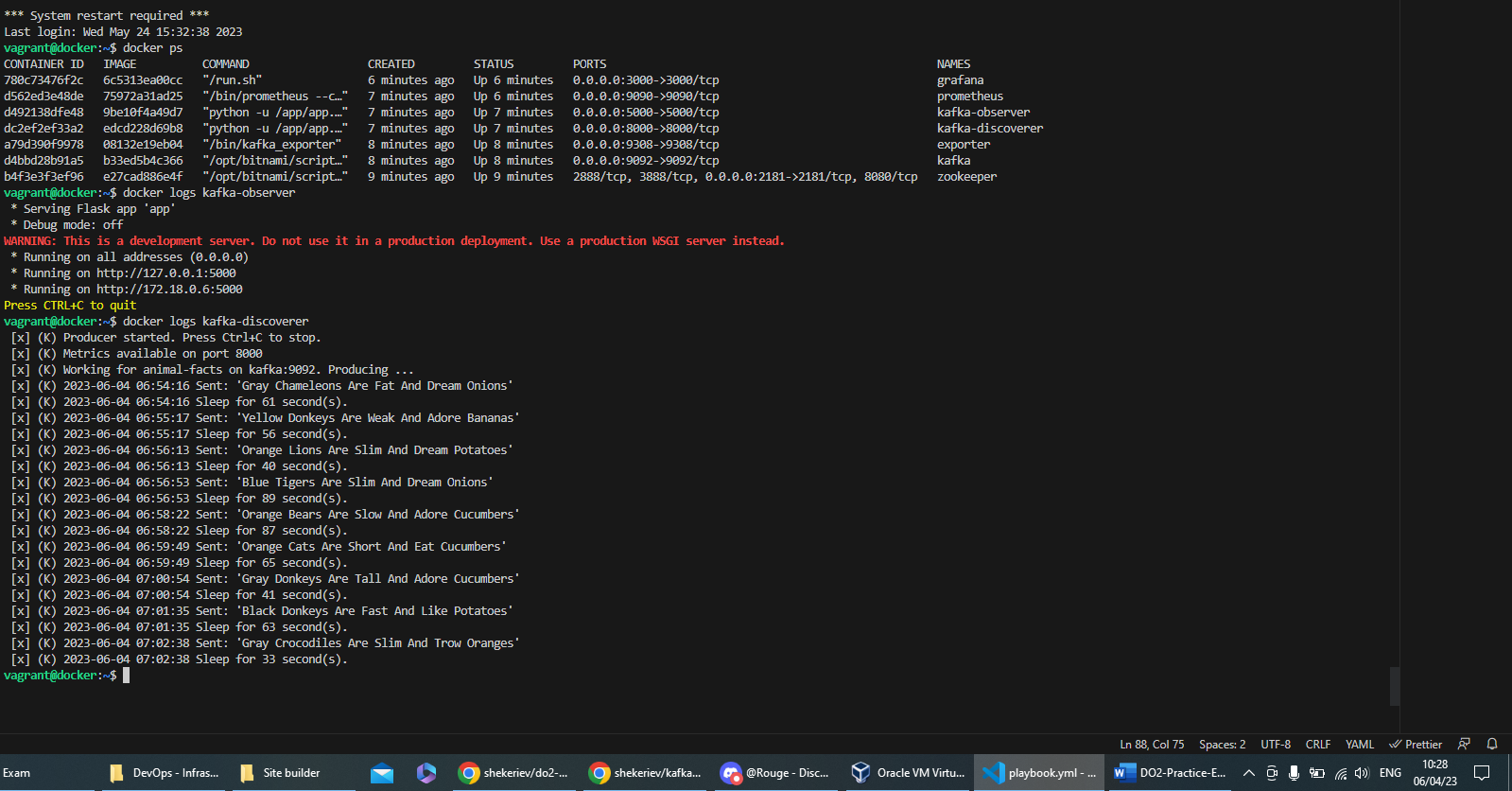
**(T402, 3 pts)**

* Working ***web application #4***

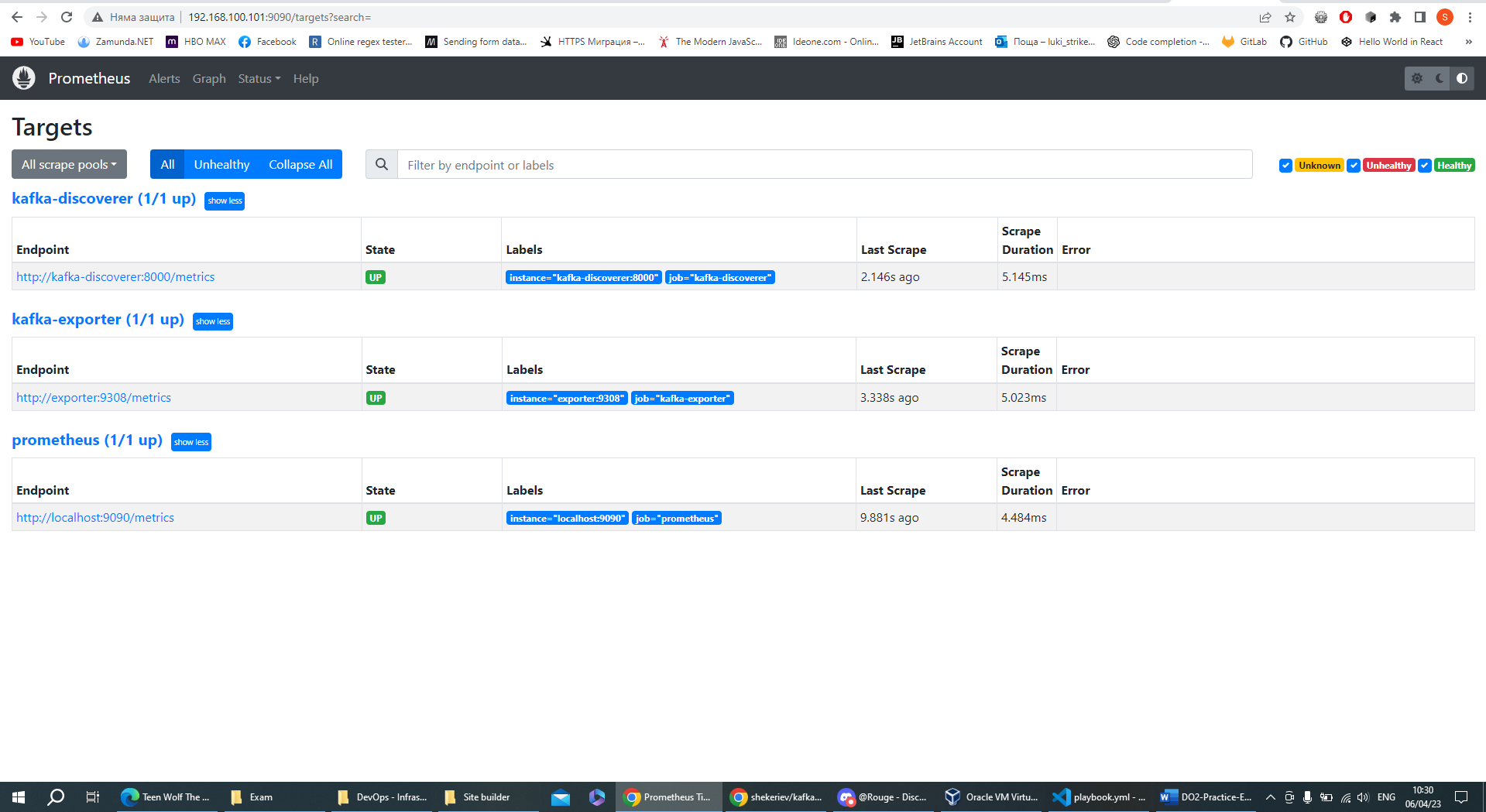
**(T403, 3 pts)**

# Solution

1. Vagrantfile
2. hosts.sh
3. docker.sh
4. terraform.sh
5. .\terraform\kafka\main.tf
6. .\terraform\exporter\main.tf
7. .\terraform\apps\main.tf
8. .\terraform\monitoring\main.tf
9. .\terraform\monitoring\prometheus.yml
10. .\terraform\monitoring\datasource.yml
11. .\manifests\docker.pp
12. .\playbooks\playbook.yml
13. .\playbooks\ports.conf
14. .\playbooks\app1.conf
15. .\playbooks\app2.conf
16. vagrant up



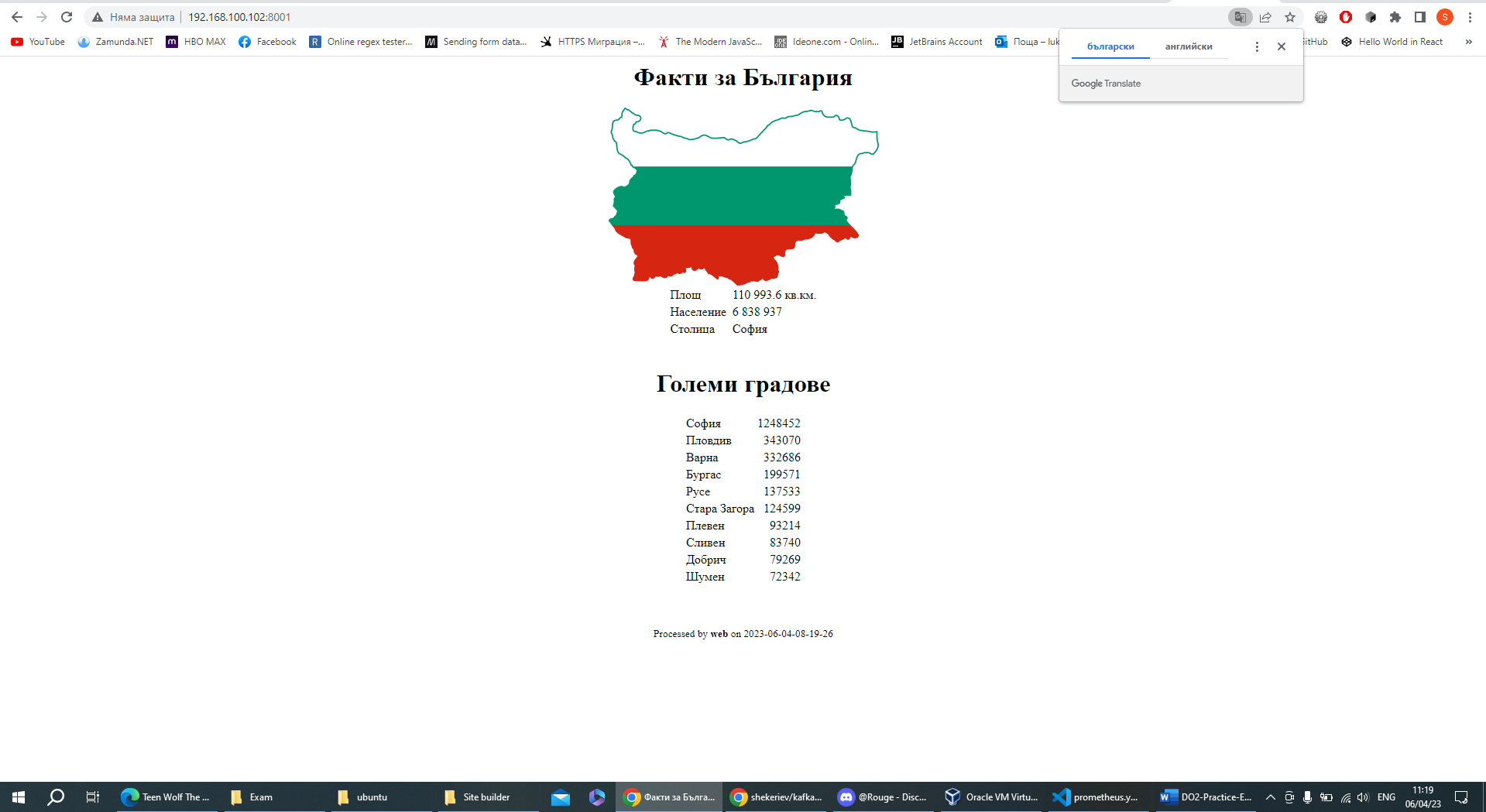
1. Open in your browser <http://192.168.100.101:9090/> then click Status->Targets



1. Open in your browser <http://192.168.100.101:3000/> with username **admin** and password **admin**
2. Dashboard->New->New Dashboard->Add visualization->Metric = **discovered\_facts\_total**, Last = 15 min, and what else you want to have.



1. Open in your browser <http://192.168.100.102:8001/>



1. Open in your browser <http://192.168.100.102:8002/>

