# Exam: DevOps #2 2022.11 (2023.02.05)

### 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:

Graphical user interface, application, Teams

Description automatically generated

### 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

Don’t forget to check the **Proof** section for hints and guidelines on how to prepare and present your solution

### 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***
  + (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*

*Note that (Brk), (Prd), (Cns), (Pr), and (Gr) are used just as a reference to the picture and are not the actual/required names. Of course, you can use them, but not required to. The same applies to the name of the container network (appnet), it could be just anything*

#### 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 – **8081** and **8082**
* (T205, 4 pts) Deploy both applications (**app1** and **app4**) files to the corresponding folders of the virtual hosts
* (T206, 3 pts) Do a basic (installed and running) installation of **MariaDB/MySQL** 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 (****app1*** *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***. This would mean that messages/events are being generated by the discoverer (shown on the console) and consumed by the observer (shown on its web interface)
* (T402, 3 pts) Working ***web application #1***. This would mean that when visiting the web interface there is a connection to the database and the application works as expected (no missing parts)
* (T403, 3 pts) Working ***web application #4***. This would mean that when visiting the web interface there is a connection to the database and the application works as expected (no missing parts)

### Proof

Prepare a compressed archive with the files of your solution and any supporting files and upload it on the site

Make sure that you include at least all configuration files, a brief description of the workflow and pictures of important moments/achievements. This should include at least the state of the applications and any monitoring outcomes. Of course, pictures of other important moments and milestones are more than welcome

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

Make sure that all temporary and hidden (created by applications like vagrant, terraform, etc.) files are not included

In general, any hint (in written and/or with pictures) on what you do and why will be more than welcome