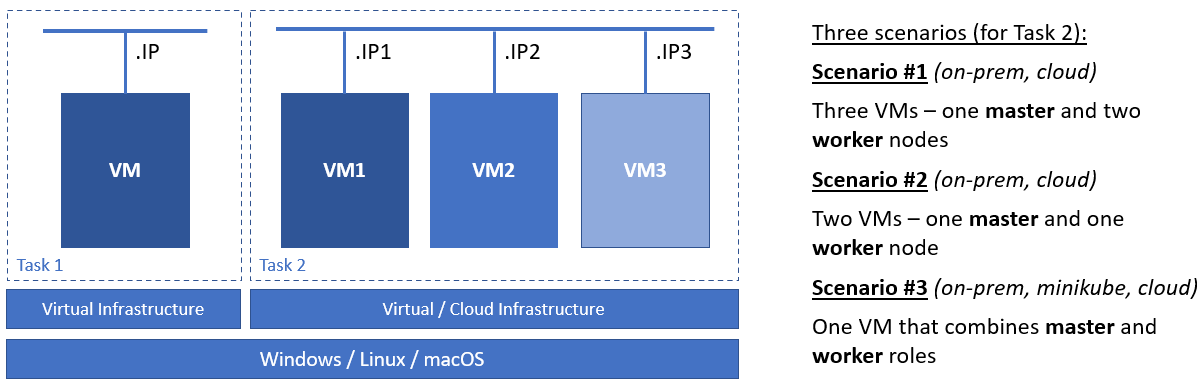
# Exam: DOF (2021.07) / 2021.09.18

## Main goal

You are expected to utilize all or most of the studied products and technologies and create an infrastructure with a few hosts. The **emphasis** should be on **features** usage **demonstration** and not on finding an optimal solution

Your solution may look like:



Please note, that it is not mandatory to have both set of machines up at the same time. Instead, finish with one of the tasks, then power off the machines, and continue with the other task. Don’t forget to prepare a proof of what you did (check the last paragraph of this document)

**Project URL**: [**https://github.com/shekeriev/dof-exam-2021.git**](https://github.com/shekeriev/dof-exam-2021.git)

## Prerequisites, Rules, and Guidelines

You must have access to:

* Source control or versioning system. You can create a free account in **github.com** if you do not have one
* Docker registry. You can create a free account in **hub.docker.com** if you do not have one

You must clone the project in **your own repository** either by forking it or by downloading and uploading the files

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

## Tasks

**Project files are available here:** [**https://github.com/shekeriev/dof-exam-2021.git**](https://github.com/shekeriev/dof-exam-2021.git)

### Task 1: Single Host (15 pts.)

The ultimate goal is to have the whole **infrastructure** as a **file** or **set** of **files**. Then on top of it to spin up a simple containerized application using **Terraform**

This task must be executed on-premise

#### Infrastructure and Configuration Management (12 pts)

Spin up a single Docker host

* (T1-101 / 3 pts) Single machine prepared for **Docker** host
* (T1-102 / 6 pts) Use a configuration management tool (any of **Puppet**, **Chef**, or **Salt**) to install and configure **Docker** and any other required packages on the machine. You may combine it with **BASH** if you like
* (T1-103 / 3 pts) Use **Terraform** to spin up the containerized application

#### Containerized Application (3 pts)

Manage to have a working application

* (T1-201 / 3 pts) Working application

### Task 2: Cluster (45 pts.)

The ultimate goal is to have the whole **infrastructure** as a **file** or **set** of **files**. Then on top of it to create an **automated** **build** **process** which upon a change in the project’s files will trigger a build and application update. The change notification can be either triggered by the **source control system** or as a result from a regular check on **every five minutes** initiated by the CD/CI system

**All** **hosts** should be **provisioned** and **configured** either on-premise (set of virtual machines or minikube) or in a cloud platform of your choice in an automated fashion by utilizing **any combination** of the **appropriate tools**. There should be monitoring on the hosts (master + worker nodes) **with Elastic Stack**

#### Infrastructure (8 pts)

Create a small **Kubernetes** or **Nomad** cluster

* (T2-101 / 5 pts) At least one machine prepared for master
* (T2-102 / 3 pts) One or more additional machines prepared for workers (or master configured to execute pods as well)

#### Monitoring (15 pts)

Deploy **Elastic Stack** components to serve as a monitoring solution

* (T2-201 / 6 pts) At lest **Elasticsearch** and **Kibana** to be installed, configured, and running (you may add **Logstash** as well, or skip it, it is up to you)
* (T2-202 / 3 pts) Installed and configured at least one **Beat** that provides data (CPU, RAM, etc.) about the nodes (even though it is not necessary, you may include the master as well)
* (T2-203 / 2 pts) Defined index pattern in **Kibana** for the data coming from the installed **Beat**(s)
* (T2-204 / 2 pts) Created at least one visualization component that utilizes some of the collected data
* (T2-205 / 2 pts) **Kibana** dashboard that includes the visualization(s) created earlier

#### CD/CI (15 pts)

Deploy a simple CD/CI solution and utilize it to build and deploy a simple application

* (T2-301 / 3 pts) Working base installation of a CD/CI solution, for example **Jenkins**
* (T2-302 / 1 pts) Additional components and configuration as per the requirements (plugins, settings, etc.)
* (T2-303 / 3 pts) Created one job or pipeline to facilitate the requirements
* (T2-304 / 2 pts) Trigger a build either on a regular basis (**every 5 min.**) or on source code change
* (T2-305 / 3 pts) Rebuild and push images to a registry (your local or public)
* (T2-306 / 3 pts) Propagate the changes to the running application

#### Application (7 pts)

Manage to have a working application

* (T2-401 / 7 pts) Working application

## Proof

Prepare a compressed archive containing a set of files proving that you managed to solve the assignment which may include but is not limited to:

* Configuration files
* Pictures showing important achievements during the process including the result (working application, successful execution, dashboard or visualization, etc.)
* Logs or command outputs
* Description of any manual steps that you may have executed
* Any other comments that you want to share

Files for each of the two tasks, must go in separate folders – **task1** and **task2**

Don’t forget to remove any extra folders and files (like hidden or temporary state files)

**Upload the compressed archive in the Exam section of the site**