# Practice M8: Exam Preparation (Automated Approach #2)

## Task

We are presented with a two-component Docker-ized application - **php+apache** and **mysql**

We are expected to build a complete infrastructure that includes the following hosts:

* **Ansible**
* **Docker**
* **Jenkins**
* **Nagios**

Application code is hosted on **GutHub** and available on the following URL:

<https://github.com/shekeriev/two-docker-images.git>

Our solution should do periodic checks (every two minutes) for code changes. If a change is registered, a new pair of images must be produced and then two new containers must be run out of those images

**Nagios** must be used to monitor all hosts by **PING** and **SSH**. Additionally, it must track if the containers are working as expected

## Possible Solution

Automatization and host provisioning will be implemented with the help of **Ansible** for all hosts except the **ansible.dob.lab** host which will be provisioned with **Vagrant**

After the basic provisioning is done, the **ansible** host will execute all playbooks automatically to prepare the rest of the infrastructure including adding the **docker** host as a slave node and adding the build task in **Jenkins**

### Preparation

#### Host

Extract the **M8-Practice-Exam-Prep-4-Auto.zip** file in a folder of your choice and navigate there

Explore the provided files and adjust them if needed

Start the environment provisioning process with

**vagrant up**

#### Job for the Actual Task (Solution)

Open a browser tab on the host and navigate to

<http://localhost:8080/>

And explore the created configuration

*We can make a change in the* ***GitHub*** *project to see the effect*

#### Host (check the Nagios monitoring)

Open a browser tab on the host and navigate to the following URL:

<http://localhost:8081/nagios>

When asked for credentials use **nagiosadmin** and **Password1**

Explore the hosts, services, and the map

#### Host (check the Docker application)

Open a browser tab on the host and navigate to the following URL:

<http://localhost:8082>

We can see our working application