# Practice M8: Exam Preparation (Automated Approach)

## 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

### Preparation

#### Host

Extract the **M8-Practice-Exam-Prep-3-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/>

Use credentials **admin** with password **admin** if asked

Click on the **New Item** option in the left menu

Enter or select the following:

* Name: **Docker-GitHub-Final**
* Type: **Freestyle Project**

Confirm with the **OK** button

Do the following adjustments:

* Set the project type to **GitHub project**
* Enter the following URL: <https://github.com/shekeriev/two-docker-images.git>
* Click on the second button named **Advanced**
* Select the **Use custom workspace** option and enter: **/vagrant/www-dynamic**
* Turn on the **Restrict where this project can be run** option and for label set **docker**

Scroll down to the **Source Code Management** section, select **Git**, and enter:

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

Scroll down to the **Build** section and select **Execute shell**

In the **Command** text box enter:

* **cd /vagrant/www-dynamic/php**
* **docker image build -t img-php .**

While still in the **Build** section, click on the **Add build step** button, and select **Execute shell**

Enter the following:

* **cd /vagrant/www-dynamic/mysql**
* **docker image build -t img-mysql .**

While still in the **Build** section, click on the **Add build step** button, and select **Execute shell**

Enter the following:

* **docker container rm -f dob-http || true**
* **docker container run -d --net dob-network -p 80:80 --name dob-http -v /vagrant/www-dynamic/site:/var/www/html img-php**

Once more, click on the **Add build step** button and select **Execute shell**

Then enter the following:

* **docker container rm -f dob-mysql || true**
* **docker container run -d --net dob-network --name dob-mysql -e MYSQL\_ROOT\_PASSWORD=12345 img-mysql**

Confirm with the **Save** button

Start the building process with the **Build Now** option in the left menu

Monitor the execution process

There is one final touch missing - we have to check periodically for changes

We can click on the **Configure** option in the left menu to go to the project settings

Scroll down to the **Build Triggers** section

Turn on the **Poll SCM** option

In the **Schedule** text field enter **H/2 \* \* \* \*** to check every two minutes

Confirm with the **Save** button

*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