Cloud Computing Applications and Services

University of Minho

Guide 1
Case-study application

2024

This guide assumes you have completed *Guide 0*. We now focus on the *Swap* application, which will be used as our case-study along the semester. The end goal is to understand how applications can be deployed in Cloud Infrastructures and what kind of concerns, decisions, and compromises need to be considered. Remember that we will be taking two roles: *Cloud Provider* and *Application Developer*.

- [☁] Tasks performed by the *cloud provider*;
- [</>] Tasks performed by the *application developer*;

Swap

The Swap application is used at the University of Minho to handle class enrollment and shift exchanges. In this guide, you will install its web application server and database server in two separate VMs. Swap's source code is available at: `https://github.com/Hackathonners/swap`

1 Setup [☁]

One of the most used services that Cloud Computing providers offer to clients are remote *Virtual Machines*, which customers can then configure and use as they see fit.
Assuming the role of a cloud provider, we will start by setting up two virtual machines for our application developers. Use the Vagrantfile from *Guide 0* to create them (*node1* and *node2*):

```
vagrant up node1 node2
```

2 Tasks [</>]

Next we assume the role of the developer and install the *SWAP* application in the two VMs.

2.1 Testing the infrastructure

Let us begin by testing the infrastructure we were given by the Cloud Provider. Using the SSH protocol, try to access the two Virtual Machines (*node 1* and *node 2*).

```
ssh vagrant@<NODE_IP>
```

2.2 Database server

Install and configure MySQL on ***node1***.

1. Install MySQL:

   ```
   sudo apt install mysql-server
   ```

2. Launch the MySQL client command line tool

   ```
   sudo mysql
   ```

3. Use the prompt to:

   (a) Create a database:

   ```
   CREATE DATABASE swap;
   ```

   (b) Create/grant privileges to a user on *node2* to access the database:

   ```
   CREATE USER '<USER>'@'<IP>' IDENTIFIED BY '<PASSWORD>';
   GRANT ALL PRIVILEGES ON swap.* TO '<USER>'@'<IP>' WITH GRANT OPTION;
   ```

   Note: the `<USER>`, `<PASSWORD>` and `<IP>` fields should be replaced. The `<IP>` field should correspond to the IP address from where the DB client will connect from (*e.g., 192.168.56.102 - node2* private IP in our setup). The `<USER>` and `<PASSWORD>` fields can be chosen as desired. Make sure to remember these, as they will be used later in this guide.

   (c) Exit the MySQL client console and edit (with *root* permissions) the `bind-address` configuration at:

   ```
   /etc/mysql/mysql.conf.d/mysqld.cnf
   ```

   Note: the `bind-address` is the IP address of the VM where the MySQL server is deployed and listening to requests (e.g., *node1* - 192.168.56.101).

   (d) Restart the MySQL service:

   ```
   sudo /etc/init.d/mysql restart
   ```

4. Verify if the database and the user were correctly create. Connect to ***node2***, install the `mysql-client` package and run the following command to access MySQL server running on *node1*:

   ```
   sudo mysql -u<USER> -p<PASSWORD> -h <NODE1_IP> <<< "SHOW DATABASES;"
   ```

   Note: If everything was correctly configured, you should be able to see *"swap"* on the list of databases outputted by the previous command.

2.3 Web application server

Install and configure Swap (and its dependencies) on ***node2***.

1. Start by installing PHP (v7.4) along with the required packages, as required by the application, by using the following commands:

   (a)
   ```
   sudo add-apt-repository ppa:ondrej/php
   ```

   (b)
   ```
   sudo apt update
   ```

   (c)
   ```
   sudo apt install php7.4 \
     php7.4-{fpm,zip,mbstring,tokenizer,mysql,gd,xml,bcmath,intl,curl}
   ```

2. Install remaining dependencies (NodeJS, npm, and Composer–v1):
   (a) `sudo apt install nodejs npm`
   (b) `curl -sS https://getcomposer.org/installer | php -- --1`

   `sudo mv composer.phar /usr/bin/composer`
3. Now, let us install and configure the Swap application.
   (a) Clone the Swap's Git repository:
   `git clone <SWAP_REPO_URL>`
   (b) On the Swap's directory, install the required packages with composer:
   `composer install`
   (c) Use npm to install Swap:
   `npm install`
   (d) Update the database configurations (`DB_HOST` – DB server's IP address, `DB_DATABASE` – DB's name, `DB_USERNAME` and `DB_PASSWORD`) at the '`.env.example`' and rename the file to '`.env`'
   Note: These variables should match the configurations from the database deployment.
   (e) Generate the application's key with:
   `php artisan key:generate`
   (f) Run database migrations with:
   `php artisan migrate`
   (g) Seed the database (*i.e.,* a script to create some entries at the DB) with:
   `php artisan db:seed`
4. Now that Swap is installed and configured, let us start it with:
   `php artisan serve --host=0.0.0.0`

   Question: What is the difference between using a specific IP address versus the `0.0.0.0` address?
5. Try it out!
   (a) Access Swap from your browser. The URL should contain the private IP address of *node2* and port 8000 (*e.g.*, `http://192.168.56.102:8000`)
   (b) Log in as administrator. The username is "`contact@hackathonners.org`" and the password is "`123456`".

Extras

1. Setup an external mail server account (*e.g.*, using Mailtrap).
2. Use Redis for session management.

Learning outcomes

Experiment with the manual distributed deployment and configuration of multi-tier applications.