Cloud Computing Applications and Services

Case-study application: Swap

October 1, 2022

Swap

Consider the Swap application, used to handle class enrolment and shift exchanges. It is available from: `https://github.com/Hackathonners/swap`. The goal is to install Swap along with its dependencies and a MySQL database in separate virtual machines.

Tasks

1. Use the VagrantFile (see warm-up guide 0) to create two VMs (*e.g.*, VM1 and VM2). Then connect to the VMs through the SSH protocol.
2. Install and configure MySQL in VM2:

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

3. Use the mysql client command line to:
   (a) create a database:

   ```
   sudo mysql

   CREATE DATABASE swap;
   ```

   (b) create/grant privileges to a user on VM1 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 by the appropriate values.
   For example, the ip field shoud correspond to the private ip address defined for VM1 (*e.g., 192.168.56.101*). The user and password fields can be chosen as desired.
   (c) exit the mysql console and edit 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 (e.g., 192.168.56.102).
   (d) restart mysql service (sudo /etc/init.d/mysql restart).
4. In the other virtual machine (VM1) let us install the Swap platform and dependencies. In more detail, start by installing PHP, as required by the application, by using the following commands:
   (a) `sudo add-apt-repository ppa:ondrej/php`
   (b) `sudo apt update`
   (c) install php extensions with *sudo apt install*

   ```
   php7.4 php7.4-{fpm,zip,mbstring,tokenizer,mysql,gd,xml,bcmath,intl,curl}
   ```

5. Install remaining dependencies (NodeJS, Composer and npm):
   (a) `sudo apt install nodejs`
   (b) `sudo apt install composer`
   (c) `sudo apt install npm`
6. Clone Swap's git repository and "cd" to Swap directory. Now let us install and configure the Swap application:
   (a) do not forget to change the database configurations (DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME and DB_PASSWORD) at the .env.example and rename the file to .env. These should match the configurations previously defined at the Step 3 of this guide.
   (b) install required packages with composer:
   ```
   composer install
   ```
   (c) use npm instead of yarn to install Swap:
   ```
   npm install
   ```
   (d) generate the appplication's key with:
   ```
   php artisan key:generate
   ```
   (e) run database migrations with:
   ```
   php artisan migrate
   ```
   (f) seed the database with:
   ```
   php artisan db:seed
   ```
7. Start swap with:
   ```
   php artisan serve --host=0.0.0.0
   ```
   Note: Understant the difference between using a specific ip address versus the 0.0.0.0 ip
8. Try it out!
   (a) access swap from your browser. The url should contain the private ip address of VM1 and port 8000 (e.g., 192.168.56.101:8000)
   (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., by using mailtrap).
2. Use Redis for session management.

Questions

1. What is this application's architecture and what pattern(s) are present?
2. What would you expect the bottleneck of this application to be? Why?
3. How would you scale this application? Which patterns would you use? Why?

Learning outcomes Experiment the distributed deployment and configuration of multi-tier applications.