System Deployment and Benchmarking

Case-study application: Swap

September 25, 2019

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. Install and configure MySQL (package mysql-server) in a virtual machine (VM1).
2. Use the mysql client command line to:
   (a) create a database.
   ```
   (sudo) mysql -p
   ```
   ```
   CREATE DATABASE swap;
   ```
   (b) create/grant privileges to a user on the other VM to access the database.
   ```
   CREATE USER ’user’@’ip’ IDENTIFIED BY ’password’;
   ```
   ```
   GRANT ALL PRIVILEGES ON swap.* TO ’user’@’ip’ WITH GRANT OPTION;
   ```
   (c) edit bind-address configuration at:
   ```
   /etc/mysql/mysql.conf.d/mysqld.cnf
   ```
   (d) restart mysql service.
3. In the other virtual machine (VM2) install the Swap platform and dependencies.
4. Install PHP, as required by the application, by using the following commands:
   (a) `sudo add-apt-repository ppa:ondrej/php`
   (b) `sudo apt-get update`
   (c) install php extensions with apt-get
   ```
   php7.2 php7.2-{fpm,zip,mbstring,tokenizer,mysql,gd,xml,bcmath,intl,curl}
   ```
5. Install remaining dependencies (NodeJS v8)
   (a) `curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -`
   (b) `sudo apt-get install nodejs`
6. Clone Swap’s git repository and cd to Swap directory
7. Install Composer and Swap
   (a) instructions at `https://getcomposer.org/download/`

   (b) install composer with php:

   ```
   php composer.phar install
   ```

   (c) follow Swap's readme file

   (d) do not forget to change the database configurations!

   (e) use npm instead of yarn to install Swap:

   ```
   npm install
   ```

8. Start swap with:

   ```
   php artisan serve --host=0.0.0.0
   ```

9. Try it out!

Extras

1. Setup an external mail server account (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?
4. How would you benchmark this application?

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