

## Flask 4: Deployment on OpenShift







### Lab Objectives

The objectives of this lab are to learn:

- how to change from SQLite to MySQL db, and use it;
- how to deploy your website on OpenShift.

### PRELIMINARIES

- As before, this exercise is not assessed, but you should complete all tasks.
- **VPN connection is required to connect to MySQL, OpenShift, and phpMyAdmin.**
- The School's **MySQL server** is available at [csmysql.cs.cf.ac.uk](https://csmysql.cs.cf.ac.uk). You should use this for all work in this module<sup>(1)</sup>.
- As for all labs, you should do all the work with your virtual environment **activated**.
- Review the **RELIMINARIES** section in Flask 1 instructions - these are relevant to this lab too.

Useful addresses and resources	
Snapshots demonstrating code at various points of lab tasks completion:	<a href="https://git.cardiff.ac.uk/scmne/flask-labs">https://git.cardiff.ac.uk/scmne/flask-labs</a> 
COMSC <b>MySQL</b> db server:	<a href="https://csmysql.cs.cf.ac.uk">csmysql.cs.cf.ac.uk</a>
COMSC <b>phpMyAdmin</b> :	<a href="https://www.cs.cf.ac.uk/phpMyAdmin">https://www.cs.cf.ac.uk/phpMyAdmin</a> 
COSMC <b>OpenShift</b> server	<a href="https://console.openshift.cs.cf.ac.uk/">https://console.openshift.cs.cf.ac.uk/</a> 
MySQL manual:	<a href="https://dev.mysql.com/doc/refman/8.0/en/">https://dev.mysql.com/doc/refman/8.0/en/</a>  *
SQL statement/syntax index:	<a href="https://dev.mysql.com/doc/refman/8.0/en/dynindex-statement.html">https://dev.mysql.com/doc/refman/8.0/en/dynindex-statement.html</a>  *
* You can select documentation for other versions as well on that page.	
SQL tutorial and reference:	<a href="http://www.w3schools.com/sql/">http://www.w3schools.com/sql/</a> 
SQL and MySQL cheat sheets ( <i>in no particular order</i> )	<a href="#">[1]</a> , <a href="#">[2]</a>
Also see the 'Useful Resources' section in Flask 1 lab.	

(1) Requirements for other modules might differ - always check with your lecturer(s).

## MySQL: Prerequisites

✓ **Make sure you have your MySQL account already set up.**

- If not, you need to go to <https://dbmanager.cs.cf.ac.uk/mysql/> to create a new account.
- On that page, you can also create **up to 5** databases on your account.
- Your MySQL account is **not** the same as your normal Cardiff university one, and you should set up a **different password** for this account. You can always reset your MySQL password by going to <https://docs.cs.cf.ac.uk/services/passwords/>.
- Further information on MySQL in the School is found at <http://docs.cs.cf.ac.uk/notes/mysql-in-the-school/>.

You can **interact with the MySQL server** in several ways, using:

- web-based front end **phpMyAdmin**, available at <https://phpmyadmin.cs.cf.ac.uk>;
- terminal-based SQL command interface to MySQL (see quick guide: <https://docs.cs.cf.ac.uk/notes/accessing-mysql-from-windows/>);
- GUI front end called “MySQL Workbench” (see: <https://docs.cs.cf.ac.uk/notes/accessing-mysql-from-windows/>).

**NB:** A worksheet, containing exercises on how to create, use and manage MySQL databases along with accompanying video workthrough is available. These are **optional**, but useful to get a better understanding of how a MySQL db can be managed.

## CHANGING DB to MySQL

So far, we have used SQLite db. This was good enough for testing development of the website locally. To enable better control whilst deploying our blogging website on the remote server, we will change our database to MySQL. Similar to Flask 2 Lab, we will be using **Flask-SQLAlchemy**<sup>(2)</sup> to manage connection to our db together with PyMySQL driver<sup>(3)</sup>.

1. If you haven't done it yet, create a MySQL db to be used for your blogging website , e.g. `flask_lab_db`
2. Install `pymysql`.
3. In `__init__.py`, change the setting for the db connection by **replacing**:

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:/// ' + os.path.join(basedir,
↪ 'blog.db')
```

**with:**

```
app.config['SQLALCHEMY_DATABASE_URI'] =
↪ 'mysql+pymysql://USERNAME:PASSWORD@csmysql.cs.cf.ac.uk:3306/USERNAME_DATABASE_NAME'
```


### To Note:

- **NB:** `USERNAME` is the one that starts with the lowercase 'c'; `PASSWORD` is your MySQL password (**not** your Cardiff university network one!); and `USERNAME_DATABASE_NAME` is the name of your db prefixed with your username, e.g. `c123456_flask_lab_db`. **All are case sensitive!**
- The following lines can now be deleted as they are no longer needed:

```
...
import os
...
basedir = os.path.abspath(os.path.dirname(__file__))
...
```


4. Using the db schema specified in `models.py`, create the two db tables, using python shell commands - see *Flask 2 Task 6*.
5. Populate the db with some suitable data, using `sql` commands or **phpMyAdmin** GUI. page

---

(2) <https://flask-sqlalchemy.palletsprojects.com/en/2.x/> 

(3) <https://pypi.org/project/PyMySQL/> 

## DEPLYOMENT ON OPENSIFT



To deploy our Flask website we will be using our School's OpenShift server (<https://console.openshift.cs.cf.ac.uk/> ).<sup>(4)</sup>

### Initial Setup and Checks

6. Check you have `requirements.txt` file in the root dir of your project, and all the necessary libraries are listed in this file. If unsure, confer with the latest snapshot in the Flask Labs repository (<https://git.cardiff.ac.uk/scmne/flask-labs> )
7. We will be using Gunicorn<sup>(5)</sup> to deploy our flask website on OpenShift, so you need to make sure `gunicorn` is added to your `requirements.txt`.
8. Edit `wsgi.py` file so the the first line reads as:  

```
from blog import app as application
```
9. Make sure you already have all the project files in the appropriate directories.
10. Commit changes locally, push to the remote repo on GitLab, and check that all the files from the local directory have been successfully pushed.

### Project on OpenShift

11. Connect to VPN.
12. Go to our OpenShift server, <https://console.openshift.cs.cf.ac.uk/> <sup>(6)</sup>, and log in with your University network credentials.
13. **Create a project**, by following the instructions in "**Creating a Project**" and "**Adding Your SSH Key**" sections on <https://docs.cs.cf.ac.uk/notes/openshift-getting-started/>  page.
14. Click on **Topology** menu item on the left, and select **From Git** tile.
  - (a) Go to your remote repo's home page on Gitlab. Copy your repo's URL by clicking on **Clone** and then **Clone with SSH**. The URL should be in the format:  
`git@git.cardiff.ac.uk:USERNAME//REPO\NAME.git`
  - (b) Paste the copied URL into **Git Repo URL** textfield. (You will get a warning message *"URL is valid but cannot be reached. If this is a private repository, enter a source Secret in advanced Git options"*.)
  - (c) Click on **Show advanced Git options**.
  - (d) Expand **Select Secret name** dropdown menu, and click on the *Secret* you created as part of creating your project in the previous task (No. 13).

---

(4) A short video, demonstrating steps on how to deploy our website on OpenShift, which are covered in this lab, is available on Learning Central.

(5) A Python Web Server Gateway Interface HTTP server - <https://gunicorn.org> 

(6) If you are on a Mac, Safari might not work with our OpenShift. Use another browser, e.g. Chrome or Firefox.

- (e) Select **Python** for the **Builder Image**, and then **Create** button.
- (f) You will be taken to **Topology** page, from which you can monitor your project build and deployment progress, and access your git repo and your deployed website's URL, which will be something like: <http://<PROJECTNAME>.apps.openshift.cs.cf.ac.uk>.

### Project Rebuild and Redeployment

15. If you modify the code, you need to rebuild your code on OpenShift. To trigger a rebuild manually:
- (a) From **Topology** page: click the round green tick icon (*'Built Complete'*), and then **Actions** dropdown menu and then **Rebuild**.
  - (b) You can monitor the progress of the build through clicking on **Logs** item .
  - (c) Your website should deploy automatically, and the changes will be shown on your website.<sup>(7)</sup>



**If you modify your code locally, you need to push your modified code to GitLab, and then re-build and re-deploy your website on OpenShift.**



**It is strongly advised to work on your code locally and not make any code modifications directly on GitLab, otherwise you might end up with having to deal with rectifying conflicts.**

---

(7) If the code changes are not reflected on your website try triggering another rebuild - sometimes there might be a delay between the GitLab and OpenShift.

## CONCLUDING REMARKS

This completes the series of 'essential' labs on Flask, and you should now have a basic blogging site. You could use the lab work for your coursework, but you will need to carry on and implement additional functionality, required for the coursework - see the coursework brief on Learning Central. You should also style your website to create a '*look and feel*' you want your website to have.

## Limitations and Further Work

Due to the time and scope constraints, the work we produced in these labs has **several limitations**, e.g. with regard to enhanced security and functionality. This constitutes **further work**, e.g. in your coursework and other independent work, other modules and beyond.

Other suggestion for independent work include (but not limited to):

- Completion of optional exercises on how to implement the 'Admin Panel'. This is not essential for the coursework, but you might find it useful for managing your website contents using a GUI interface.
- Optional exercise on MySQL are also available.
- Learning how to create Flask modular applications using **blueprints** <sup>(8)</sup>, how to test Flask applicators <sup>(9)</sup>; and how to move to Django framework <sup>(10)</sup>.

---

(8) <https://flask.palletsprojects.com/en/2.0.x/blueprints/> 

(9) <https://flask.palletsprojects.com/en/2.0.x/testing/> 

(10) <https://www.djangoproject.com> 