

# ECS417U Lab Sheet

## Week 1 - OpenShift and Git Repository

Before doing the lab exercises, you will need to set up OpenShift, which is where you will be deploying your weekly exercises. In addition to this, you will need to set up your Git repository for the work you do in this module.

### Create a Git repository with PHP and MySQL template code

1. Login into your GitHub account. If you do not have one, then you can create one via the following link:  
<https://github.com>

2. Access the following repository via the following link:  
<https://github.com/sclorg/cakephp-ex>

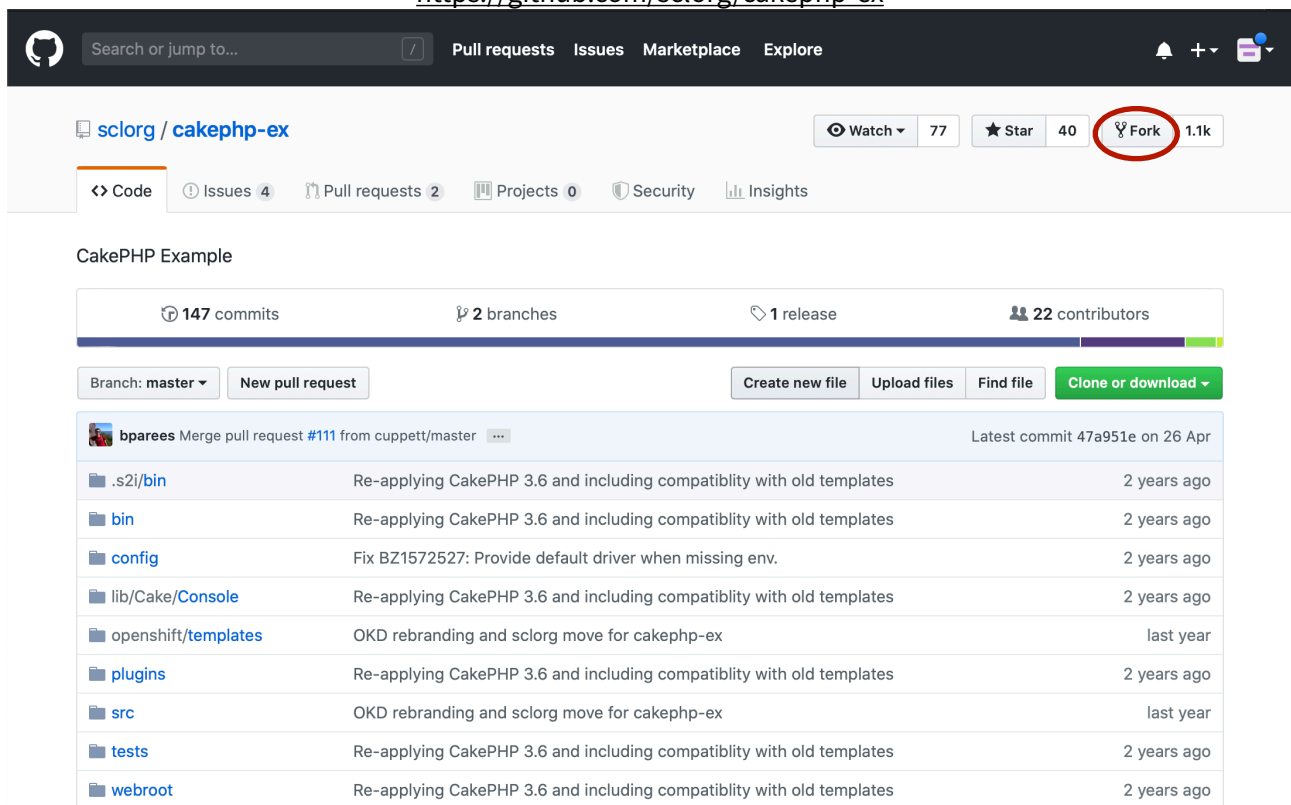


FIGURE 1

3. Fork a copy of this repository by clicking on the 'Fork' link (Figure 1).

4. Open Terminal (Mac and Linux) or Git Bash (Windows)

5. Create a bare clone of the repository by typing the following:

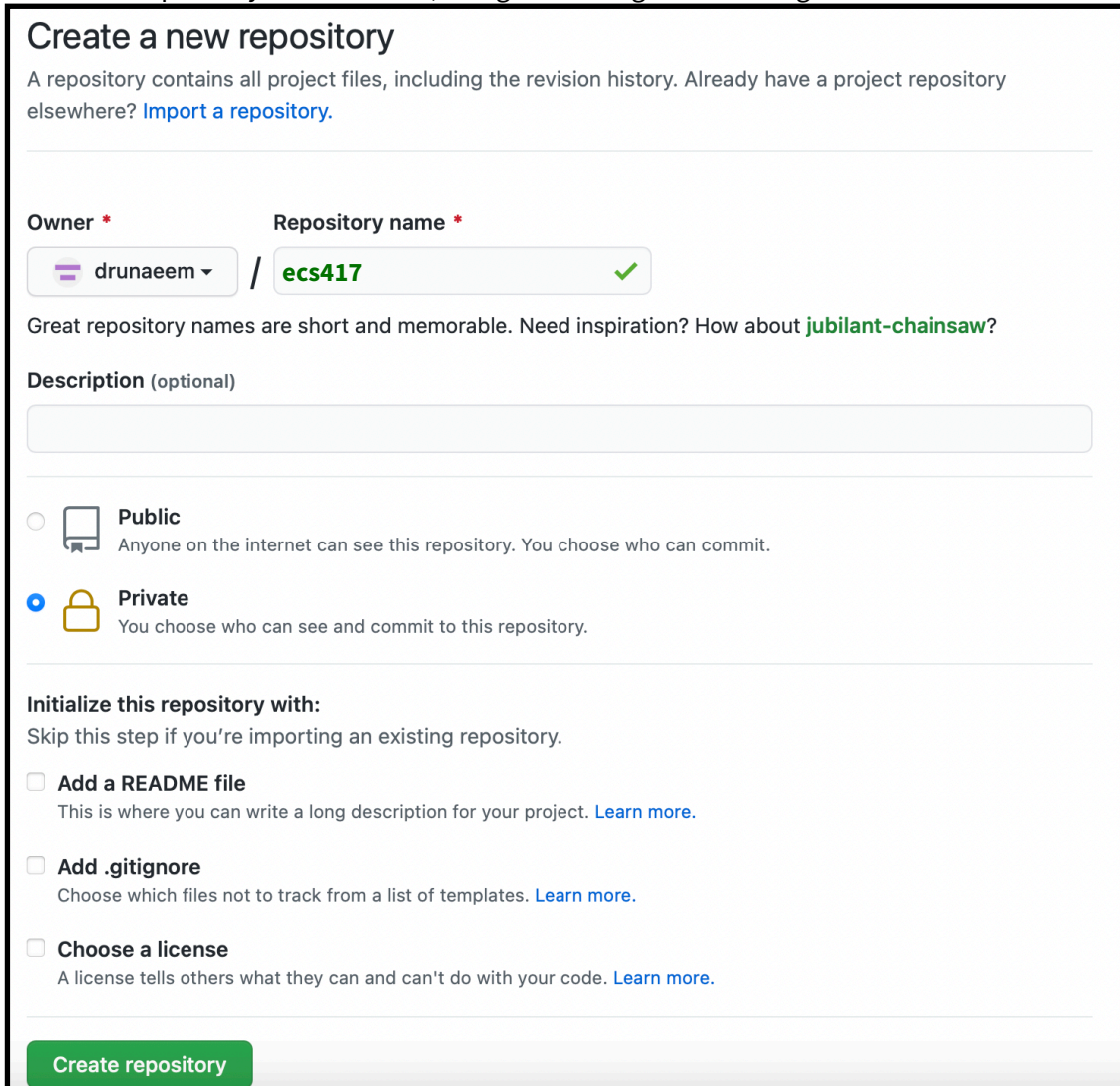
```
git clone --bare https://github.com/<username>/cakephp-ex.git
```

Note: You may need to install git if it's not installed following the standard process of installation via link below:

<https://gitforwindows.org/> - Windows

<https://git-scm.com/download/mac> - Mac

6. Create a new repository called ecs417, using the settings shown in figure 2.



**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \* drunaeem / Repository name \* ecs417 ✓

Great repository names are short and memorable. Need inspiration? How about [jubilant-chainsaw](#)?

Description (optional)

☐ Public  
Anyone on the internet can see this repository. You choose who can commit.

☒ Private  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

**Create repository**

**FIGURE 2**

7. Mirror-push the bare clone repository (cakephp-ex.git) to your new repository 'ecs417.git'

```
cd cakephp-ex.git/  
git push --mirror https://github.com/<username>/ecs417.git
```

Note: During this step you will be prompted to enter your GitHub login credentials.

8. Remove the temporary local repository you created in step 5.

```
cd ..  
rm -rf cakephp-ex.git
```

9. Clone your repository 'ecs417.git' onto your local machine

```
git clone https://github.com/<username>/ecs417.git
```

### Create a Git “personal access token”

10. In your GitHub account, navigate to Settings -> Developer settings -> Personal access tokens

11. Choose “Generate new token”

12. Enter a token description of your choice (e.g. ecs417) for the ‘Note’ field and select “repo” (Full control of private repositories). Then click on ‘Generate token’. You must ensure that you make a note of the generated token, as you will need it for OpenShift deployment.

### Create Openshift project on web console from on your Git repository

13. Create a CakePHP + MySQL project on OpenShift web console by accessing the following link:  
<https://console.okd.eecs.qmul.ac.uk>

Note: you will need to login with your EECS credentials.

14. Choose “CakePHP + MySQL” from the Service Catalogue. Just choose a name for your project.

15. In the Git Repository URL (figure 3), use the URL for your Git repository using your personal access token generated as:

<https://<generated token>@github.com/<GIT username>/ecs417.git>

\* Volume Capacity

1Gi

Volume space available for data, e.g. 512Mi, 2Gi

\* Git Repository URL

The URL of the repository with your application source code.  
Git Repository URL is required.

FIGURE 3

16. The database field entries should entered as shown in figure 4.

\* Database Engine

mysql

Database engine: postgresql, mysql or sqlite (default).

\* Database Name

ecs417

\* Database User

user

Database Password

password

FIGURE 4

17. Leave all other fields as they are.

### Deploying code changes

18. From the Web Console homepage, navigate to your project
19. Click on Builds -> Builds
20. Click the link with your BuildConfig name (e.g. cakephp-mysql-persistent)
21. Click the Configuration tab
22. Click the "Copy to clipboard" icon to the right of the "GitHub Webhook URL" field
23. Navigate to your repository on GitHub and click on repository Settings > Webhooks > Add webhook
24. Paste your webhook URL provided by OpenShift in “Payload URL”

25. Choose "Content type" as "application/json"
26. Disable SSL Verification.
27. Leave the defaults for the remaining fields.

### Deploy your webpage

28. Download the file helloWorld.html from QMPlus (the same place where you downloaded this lab sheet from), and save it in your local repository within the following folder 'webroot'.

29. Change your directory to local repository

```
cd ecs417
```

30. Type the following to get an overview of the files that have been staged, unstaged and untracked.

```
git status
```

The output should be as follows:

```
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    webroot/helloWorld.html

nothing added to commit but untracked files present (use "git add"
to track)
```

FIGURE 5

31. Type the following to stage all changes in your directory for the next commit. Note: The '.' Can be replaced with either a directory or specific file that needs to be staged.

```
git add .
```

32. You will now need to commit the staged snapshot, but instead of launching a test editor, use <message> as the commit message.

```
git commit -am "changed"
```

The output should be as follows:

```
[master b437e4b] changed
1 file changed, 13 insertions(+)
create mode 100644 webroot/helloWorld.html
```

FIGURE 6

33. Finally type the following:

```
git push
```

34. Your updated file will now be deployed onto the OpenShift platform. You will need to allow a few minutes for your updated files to appear, as the build process needs to complete before you can see the uploaded files. Once the build process has completed, Click on Applications -> Routes, as this is where you will get access to the URL of your webpage (figure 7). The link to access the helloWorld.html will be as follows:

<URL>/helloWorld.html

OPENSIFT DEDICATED

Application Console

un300

ecs417

Search Catalog

Add to Project

Routes

Learn More

Create Route

Filter by label

Add

Name	Hostname	Service	Target Port	TLS Termination
cakephp-mysql-persistent	<a href="http://cakephp-mysql-persistent-ecs417.bde1.qmul-eecs.openshiftapps.com">http://cakephp-mysql-persistent-ecs417.bde1.qmul-eecs.openshiftapps.com</a>	cakephp-mysql-persistent		

FIGURE 7

If you have successfully completed this, then you are strongly recommended to add/remove and edit files within your repository using the steps mentioned above. Also it will be a good idea to create folders for each lab topic within the webroot folder, as this will help you organise the work you conduct for this module. For example, 'Topic2' folder for Topic 2 exercises.