



LAB 02

Platform-as-a-Service with Heroku

Fullname: Trương Đăng Trúc Lâm

Student ID: B2111933

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

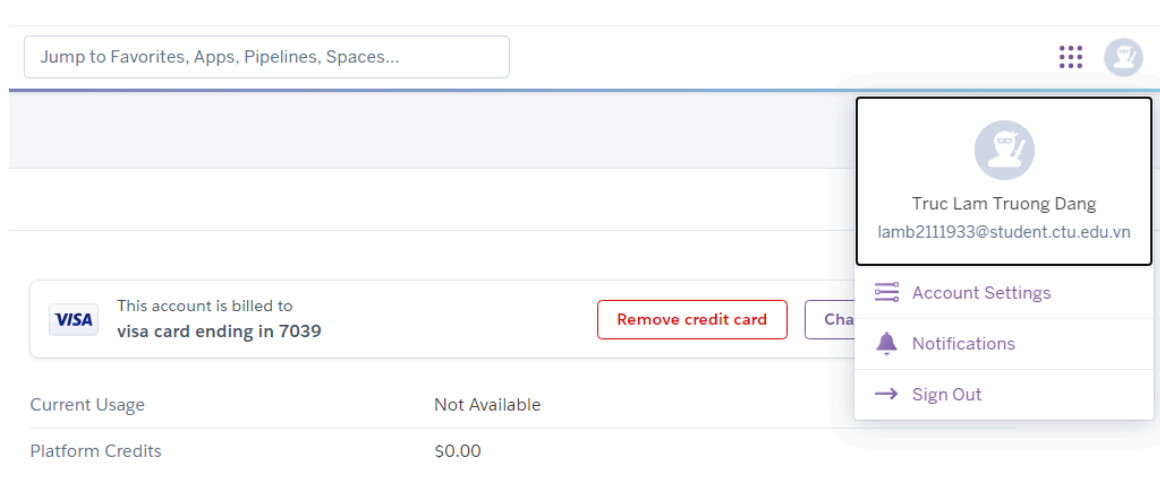
Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. In this lab, you will learn about building, deploying, and managing your apps on Heroku by finishing the exercises below.

1. Deploying a sample PHP application with Heroku

<https://devcenter.heroku.com/articles/getting-started-with-php>

(Take 2 screenshots of the sample PHP application, one after “Push local changes” step and another after “Provision a database” step)

- Prepare a verified Heroku Account



Verified Heroku Account

- Install apache, php and heroku
`$sudo apt update`

```
lamb2111933@lamb2111933-VirtualBox: ~  
lamb2111933@lamb2111933-VirtualBox:~$ sudo apt update  
[sudo] password for lamb2111933:  
Hit:1 http://vn.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://vn.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://vn.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 https://download.docker.com/linux/ubuntu jammy InRelease  
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Hit:6 https://ppa.launchpadcontent.net/katharaframework/kathara/ubuntu jammy InRelease  
Fetched 110 kB in 2s (63,2 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
2 packages can be upgraded. Run 'apt list --upgradable' to see them.  
lamb2111933@lamb2111933-VirtualBox:~$
```

Update apt packages

```
$sudo apt install apache2 mysql-server mysql-client php  
libapache2-mod-php php-mysql php-json php-xml php-mbstring  
php-curl php-zip php-memcached php-redis php-apcu curl  
php-cli php-mbstring unzip -y
```

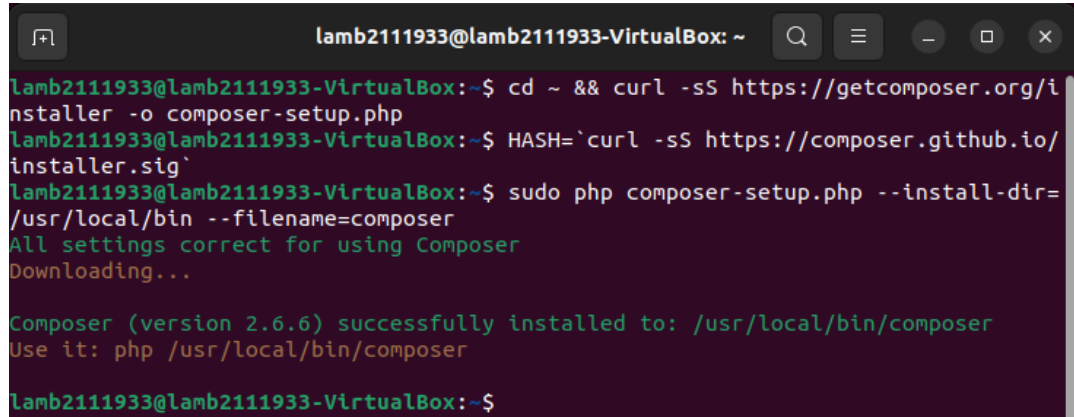
```
lamb2111933@lamb2111933-VirtualBox: ~  
lamb2111933@lamb2111933-VirtualBox:~$ sudo apt install apache2 mysql-server mysql-client php libapache2-mod-php php-mysql php-json php-xml php-mbstring php-curl php-zip php-memcached php-redis php-apcu curl php-cli php-mbstring unzip -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
curl is already the newest version (7.81.0-1ubuntu1.15).  
unzip is already the newest version (6.0-26ubuntu3.1).  
unzip set to manually installed.  
  
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...  
Processing triggers for php8.1-cli (8.1.2-1ubuntu2.14) ...  
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...  
lamb2111933@lamb2111933-VirtualBox:~$
```

Install apache, php and heroku

- Install composer

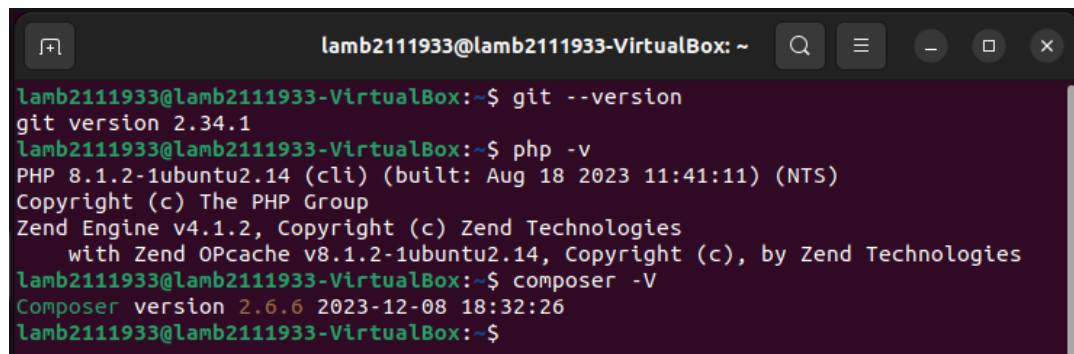
```
$cd ~ && curl -sS https://getcomposer.org/installer -o  
composer-setup.php  
$HASH=`curl -sS https://composer.github.io/installer.sig`
```

```
$sudo php composer-setup.php --install-dir=/usr/local/bin
--filename=composer
```

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: ~' showing the installation of Composer. The user runs 'cd ~ && curl -sS https://getcomposer.org/installer -o composer-setup.php', then 'HASH=`curl -sS https://composer.github.io/installer.sig`', and finally 'sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer'. The output shows 'All settings correct for using Composer', 'Downloading...', and 'Composer (version 2.6.6) successfully installed to: /usr/local/bin/composer'. It also provides the instruction 'Use it: php /usr/local/bin/composer'.

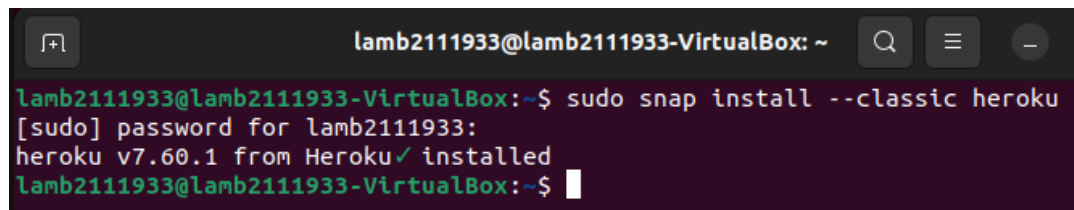
Install **composer**

- Check that you have the prerequisites installed properly.

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: ~' showing checks for prerequisites. The user runs 'git --version' (output: 'git version 2.34.1'), 'php -v' (output: 'PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS) Copyright (c) The PHP Group Zend Engine v4.1.2, Copyright (c) Zend Technologies with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies'), and 'composer -V' (output: 'Composer version 2.6.6 2023-12-08 18:32:26').

- Create heroku application (have done in exercise 1.)

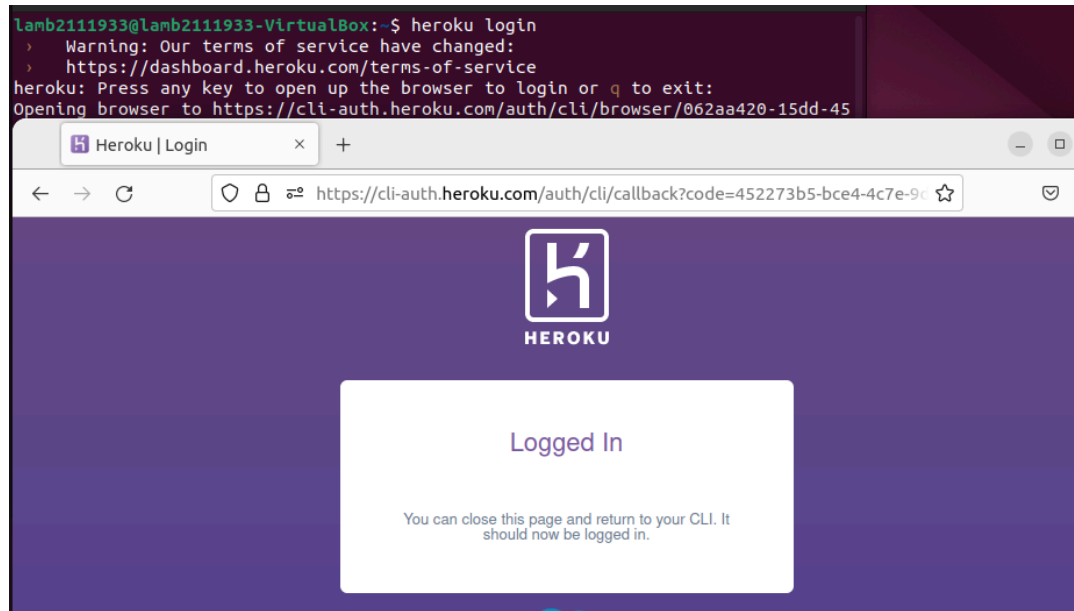
```
$sudo snap install --classic heroku
```

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: ~' showing the installation of Heroku. The user runs 'sudo snap install --classic heroku'. The prompt '[sudo] password for lamb2111933:' is shown, followed by the output 'heroku v7.60.1 from Heroku ✓ installed'.

Install **heroku**

```
$heroku login
```

if the web browser cannot open automatically, please run it manually; then copy the log in link to the address bar.



Log in heroku

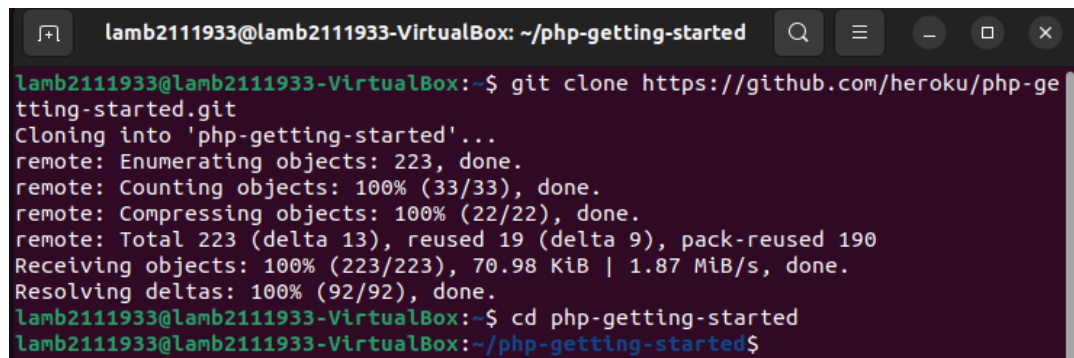
- Clone the **php-getting-started.git** to your local machine

```
$git
```

clone

<https://github.com/heroku/php-getting-started.git>

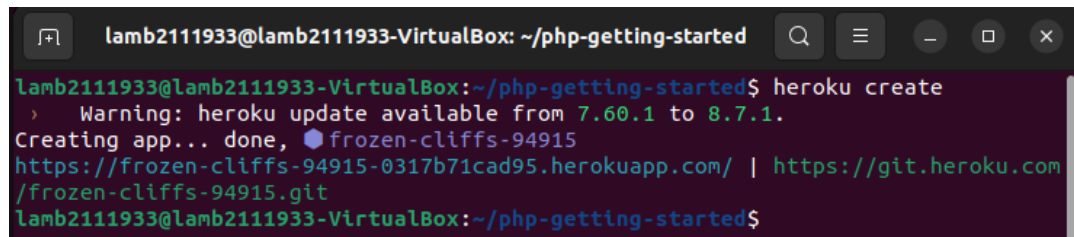
```
$cd php-getting-started
```



Get the local version of **php-getting-started.git**

- Create an app on Heroku, which prepares Heroku to receive your source code

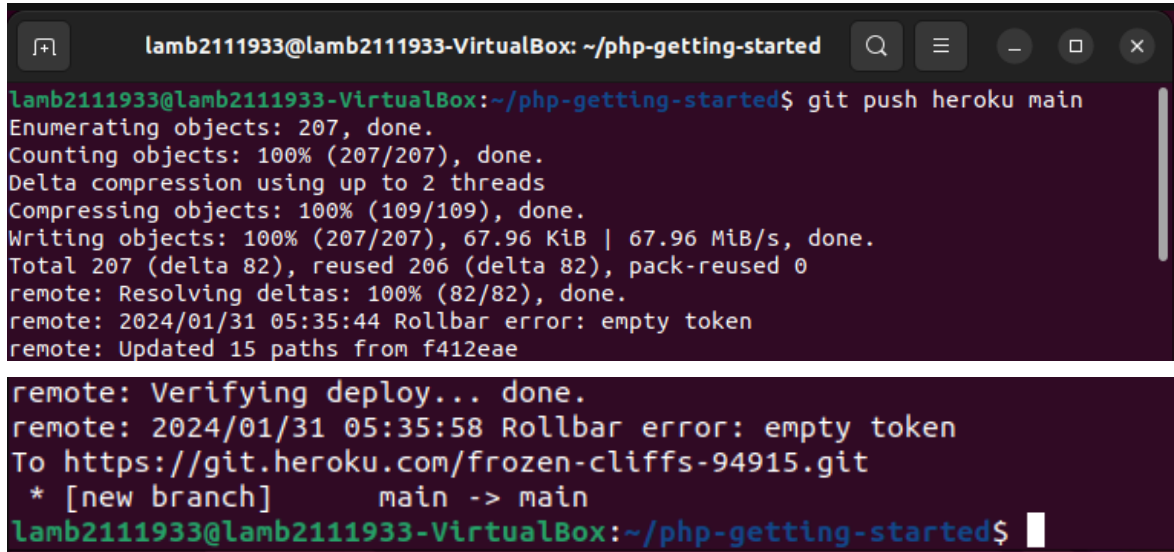
```
$heroku create
```



Create an **application** on **Heroku**

- Now deploy your code

```
$git push heroku main
```



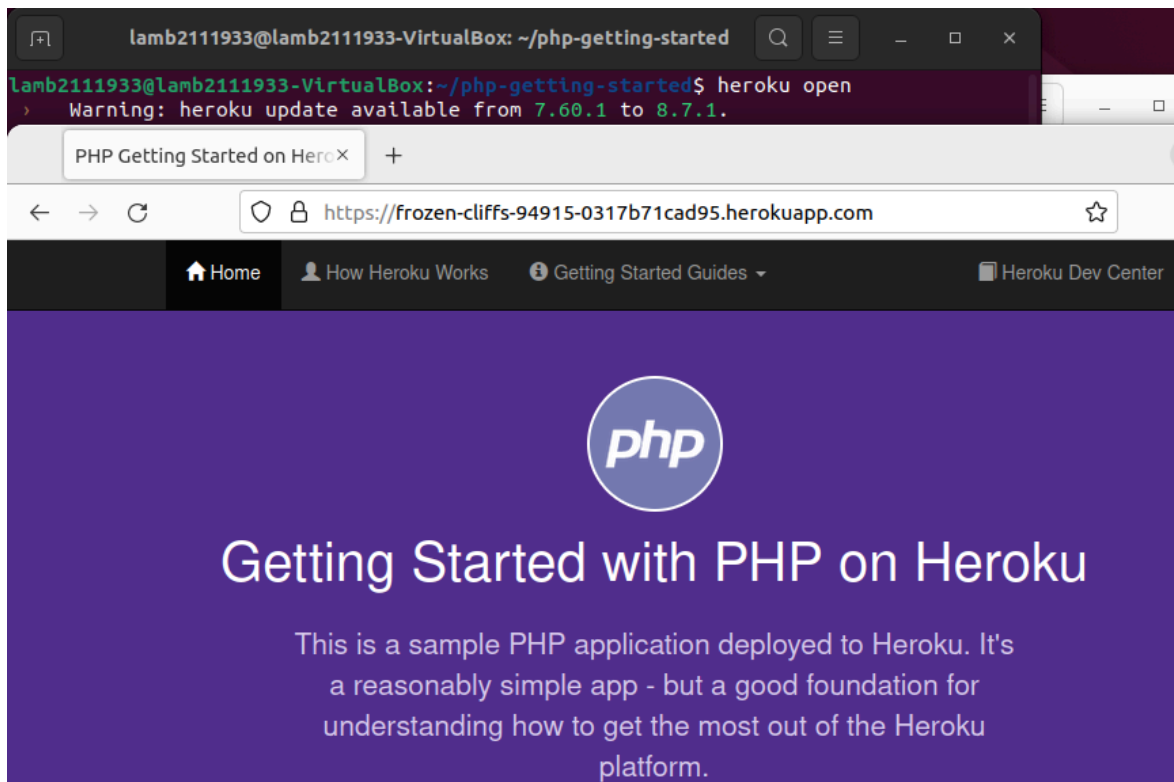
```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ git push heroku main
Enumerating objects: 207, done.
Counting objects: 100% (207/207), done.
Delta compression using up to 2 threads
Compressing objects: 100% (109/109), done.
Writing objects: 100% (207/207), 67.96 KiB | 67.96 MiB/s, done.
Total 207 (delta 82), reused 206 (delta 82), pack-reused 0
remote: Resolving deltas: 100% (82/82), done.
remote: 2024/01/31 05:35:44 Rollbar error: empty token
remote: Updated 15 paths from f412eae

remote: Verifying deploy... done.
remote: 2024/01/31 05:35:58 Rollbar error: empty token
To https://git.heroku.com/frozen-cliffs-94915.git
 * [new branch]      main -> main
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$
```

Deploy the code of **php-getting-started.git**

- Launch your application

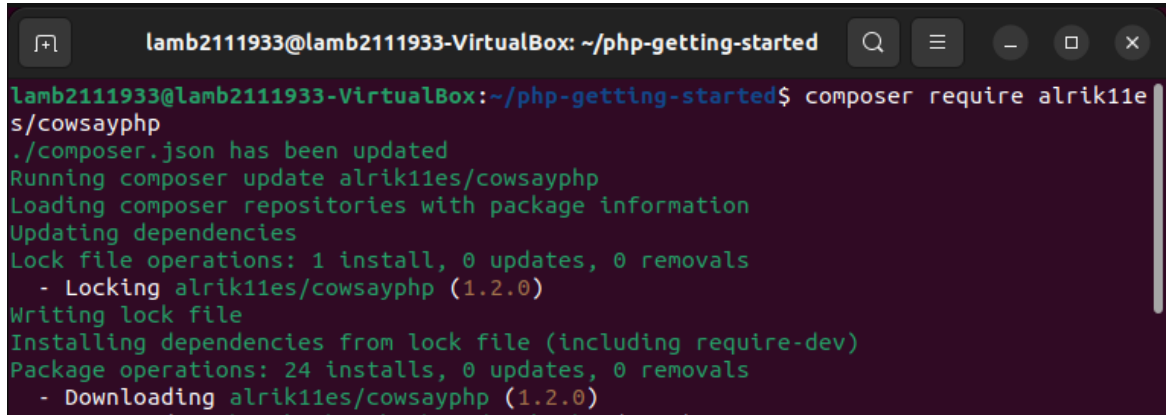
```
$heroku open
```



This is your **PHP** application

Push local changes:

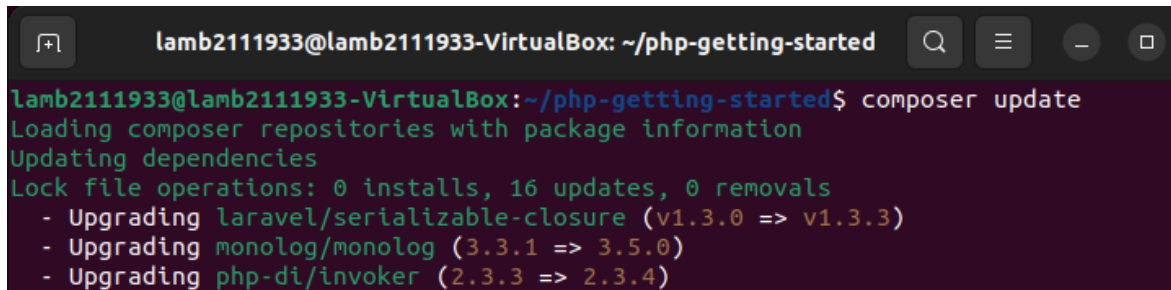
```
$composer require alrik11es/cowsayphp
```



```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ composer require alrik11es/cowsayphp
./composer.json has been updated
Running composer update alrik11es/cowsayphp
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking alrik11es/cowsayphp (1.2.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 24 installs, 0 updates, 0 removals
- Downloading alrik11es/cowsayphp (1.2.0)
```

Use **Composer** to require the new dependency

```
$composer update
```



```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ composer update
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 16 updates, 0 removals
- Upgrading laravel/serializable-closure (v1.3.0 => v1.3.3)
- Upgrading monolog/monolog (3.3.1 => 3.5.0)
- Upgrading php-di/invoke (2.3.3 => 2.3.4)
```

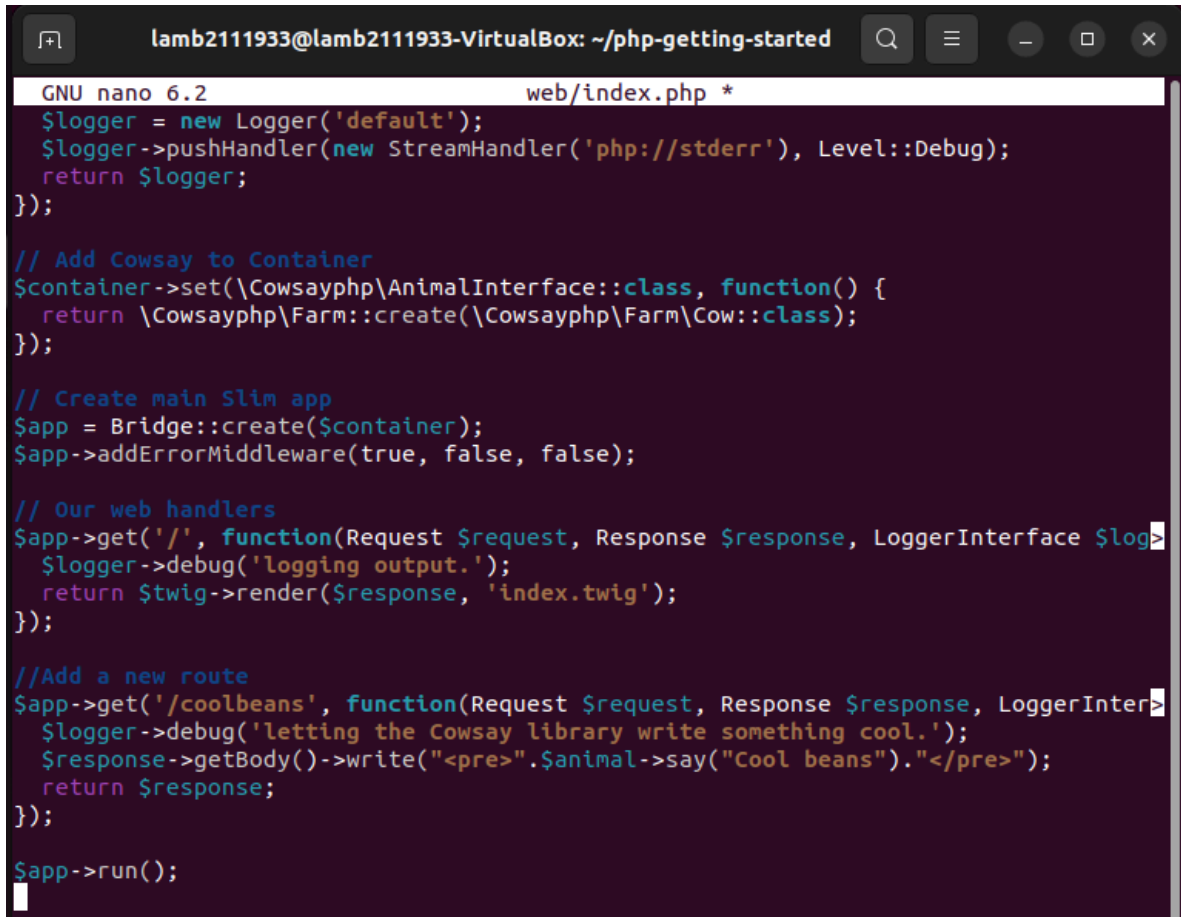
Be sure to update the dependencies

```
$nano web/index.php
// Add Cowsay to Container
$container->set(\Cowsayphp\AnimalInterface::class, function() {
    return \Cowsayphp\Farm::create(\Cowsayphp\Farm\Cow::class);
});

$app->get('/coolbeans', function(Request $request, Response $response, LoggerInterface $logger, \Cowsayphp\AnimalInterface $animal) {
    $logger->debug('letting the Cowsay library write something cool.');
```

```
$response->getBody()->write("<pre>".$animal->say("Cool beans")."</pre>");
    return $response;
```

```
});
```



The screenshot shows a terminal window with the nano 6.2 editor open at the file web/index.php. The code is a PHP script for a Slim web application. It includes a logger, sets up a container with a Cowsayphp interface and class, creates a main Slim app with error middleware, and defines two web handlers. The first handler logs and renders the index.twig template. The second handler, for the /coolbeans route, logs a message and returns a response with a preformatted string from the Cowsay library.

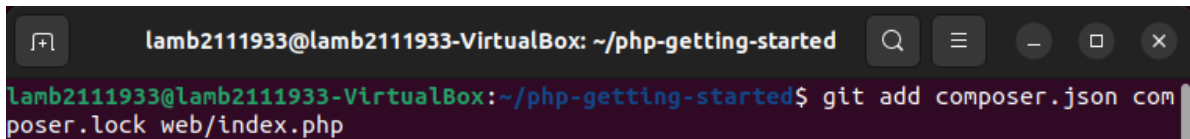
```
GNU nano 6.2 web/index.php *
$logger = new Logger('default');
$logger->pushHandler(new StreamHandler('php://stderr'), Level::Debug);
return $logger;
});

// Add Cowsay to Container
$container->set(\Cowsayphp\AnimalInterface::class, function() {
    return \Cowsayphp\Farm::create(\Cowsayphp\Farm\Cow::class);
});

// Create main Slim app
$app = Bridge::create($container);
$app->addErrorMiddleware(true, false, false);

// Our web handlers
$app->get('/', function(Request $request, Response $response, LoggerInterface $logger) {
    $logger->debug('logging output.');
```

```
$ git add composer.json composer.lock web/index.php
```

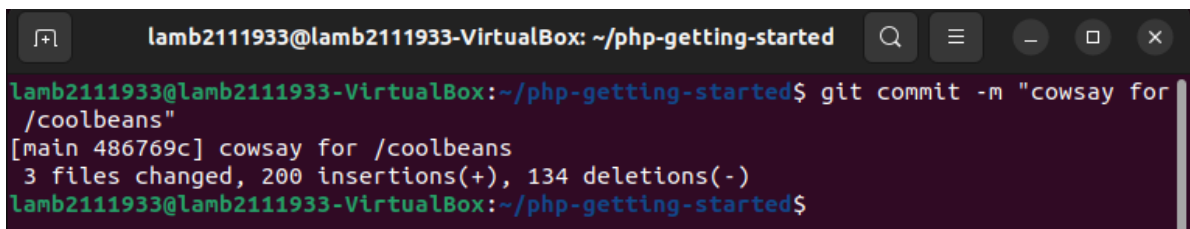


The screenshot shows a terminal window with the command `git add composer.json composer.lock web/index.php` being executed. The output is not visible, but the command is shown in the terminal history.

```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ git add composer.json com
poser.lock web/index.php
```

Add the modified files to your local repository

```
$ git commit -m "cowsay for /coolbeans"
```

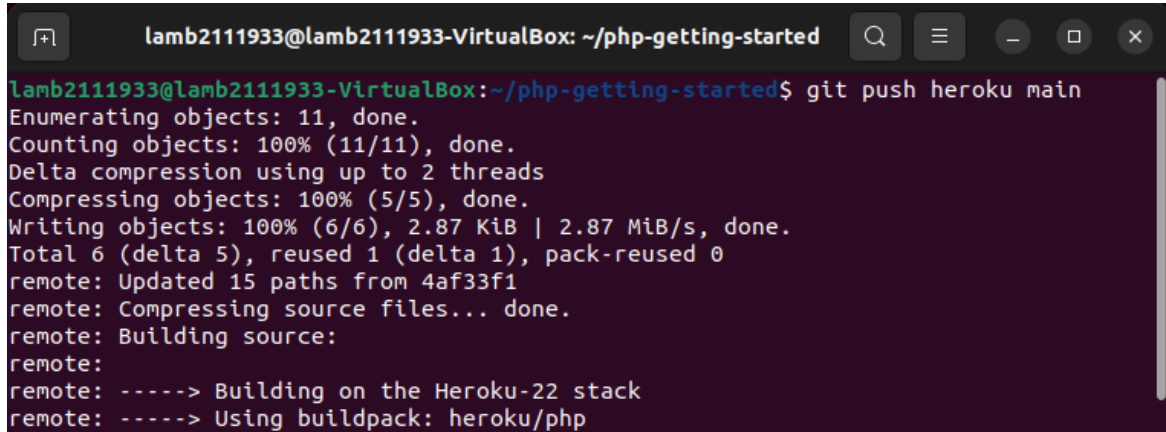


The screenshot shows a terminal window with the command `git commit -m "cowsay for /coolbeans"` being executed. The output shows the commit message, the commit hash, and the number of files changed, insertions, and deletions.

```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ git commit -m "cowsay for
/coolbeans"
[main 486769c] cowsay for /coolbeans
3 files changed, 200 insertions(+), 134 deletions(-)
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$
```


Commit these changes

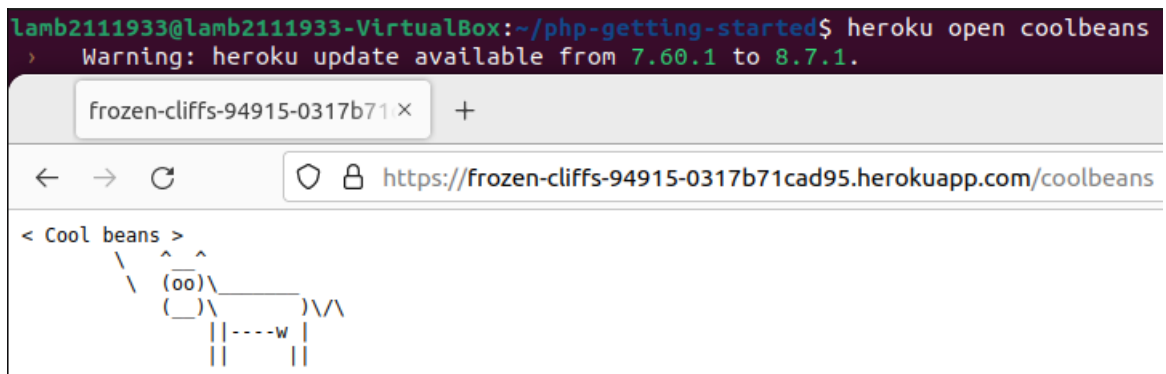
```
$ git push heroku main
```



```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ git push heroku main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 2.87 KiB | 2.87 MiB/s, done.
Total 6 (delta 5), reused 1 (delta 1), pack-reused 0
remote: Updated 15 paths from 4af33f1
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-22 stack
remote: -----> Using buildpack: heroku/php
```

Deploy just as you did previously

```
$ heroku open coolbeans
```



```
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ heroku open coolbeans
> Warning: heroku update available from 7.60.1 to 8.7.1.

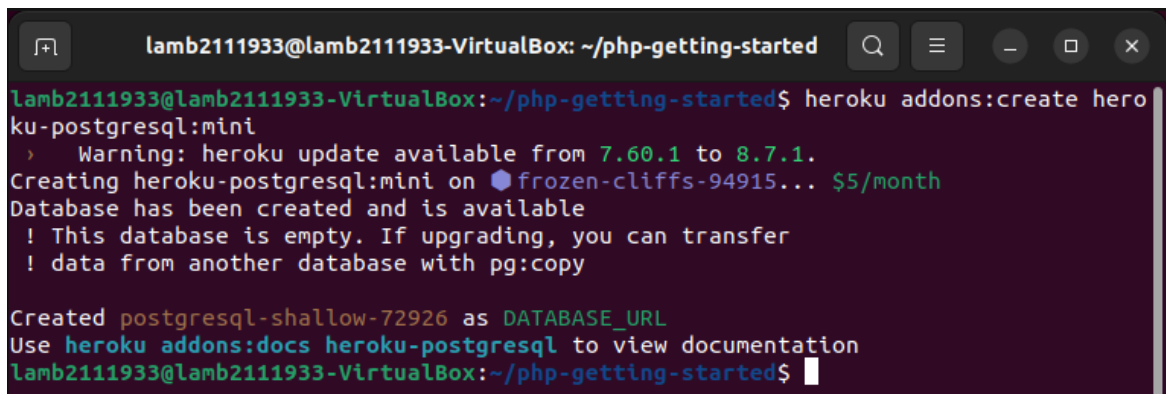
frozen-cliffs-94915-0317b71 x +
https://frozen-cliffs-94915-0317b71cad95.herokuapp.com/coolbeans

< Cool beans >
  \      ^ ^
   (oo)\_____)
    (_)\       )\/\
     ||----w |
     ||     ||
```

This is the result of the **coolbeans** page

Provision a database:

```
$ heroku addons:create heroku-postgresql:mini
```



```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ heroku addons:create heroku-postgresql:mini
> Warning: heroku update available from 7.60.1 to 8.7.1.
Creating heroku-postgresql:mini on frozen-cliffs-94915... $5/month
Database has been created and is available
! This database is empty. If upgrading, you can transfer
! data from another database with pg:copy

Created postgresql-shallow-72926 as DATABASE_URL
Use heroku addons:docs heroku-postgresql to view documentation
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$
```


Add the database

```
$nano web/index.php
```

```
// Add Database connection to Container
$container->set(PDO::class, function() {
    $dburl = parse_url(getenv('DATABASE_URL')) ?: throw new
Exception('no DATABASE_URL');
    return new PDO(sprintf(
        "pgsql:host=%s;port=%s;dbname=%s;user=%s;password=%s",
        $dburl['host'],
        $dburl['port'],
        ltrim($dburl['path'], '/'), // URL path is the DB name,
must remove leading slash
        $dburl['user'],
        $dburl['pass'],
    ));
});
```



```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
GNU nano 6.2 web/index.php *

// Add Cowsay to Container
$container->set(\Cowsayphp\AnimalInterface::class, function() {
    return \Cowsayphp\Farm::create(\Cowsayphp\Farm\Cow::class);
});

// Add Database connection to Container
$container->set(PDO::class, function() {
    $dburl = parse_url(getenv('DATABASE_URL')) ?: throw new Exception('no DATABASE_UR
    return new PDO(sprintf(
        "pgsql:host=%s;port=%s;dbname=%s;user=%s;password=%s",
        $dburl['host'],
        $dburl['port'],
        ltrim($dburl['path'], '/'), // URL path is the DB name, must remove leading sl
        $dburl['user'],
        $dburl['pass'],
    ));
});
```

Add Database connection to Container

```
//add a new route handler for URL /db
$app->get('/db', function(Request $request, Response $response,
LoggerInterface $logger, Twig $twig, PDO $pdo) {
    $st = $pdo->prepare('SELECT name FROM test_table');
    $st->execute();
```

```
$names = array();
while($row = $st->fetch(PDO::FETCH_ASSOC)) {
    $logger->debug('Row ' . $row['name']);
    $names[] = $row;
}
return $twig->render($response, 'database.twig', [
    'names' => $names,
]);
});
```

```
//Add a new route handler for URL /db
$app->get('/db', function(Request $request, Response $response, LoggerInterface $l>
    $st = $pdo->prepare('SELECT name FROM test_table');
    $st->execute();
    $names = array();
    while($row = $st->fetch(PDO::FETCH_ASSOC)) {
        $logger->debug('Row ' . $row['name']);
        $names[] = $row;
    }
    return $twig->render($response, 'database.twig', [
        'names' => $names,
    ]);
});

$app->run();
```

Add a new route handler for URL /db

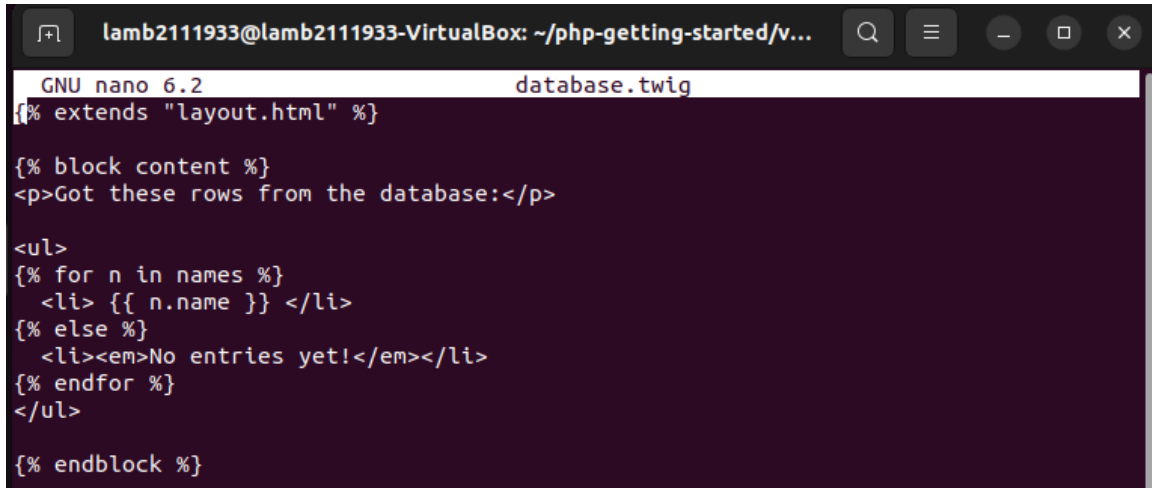
```
$cd views
$nano database.twig

{% extends "layout.html" %}

{% block content %}
<p>Got these rows from the database:</p>

<ul>
{% for n in names %}
    <li> {{ n.name }} </li>
{% else %}
    <li><em>No entries yet!</em></li>
{% endfor %}
</ul>

{% endblock %}
```



```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started/v...
GNU nano 6.2 database.twig
{% extends "layout.html" %}

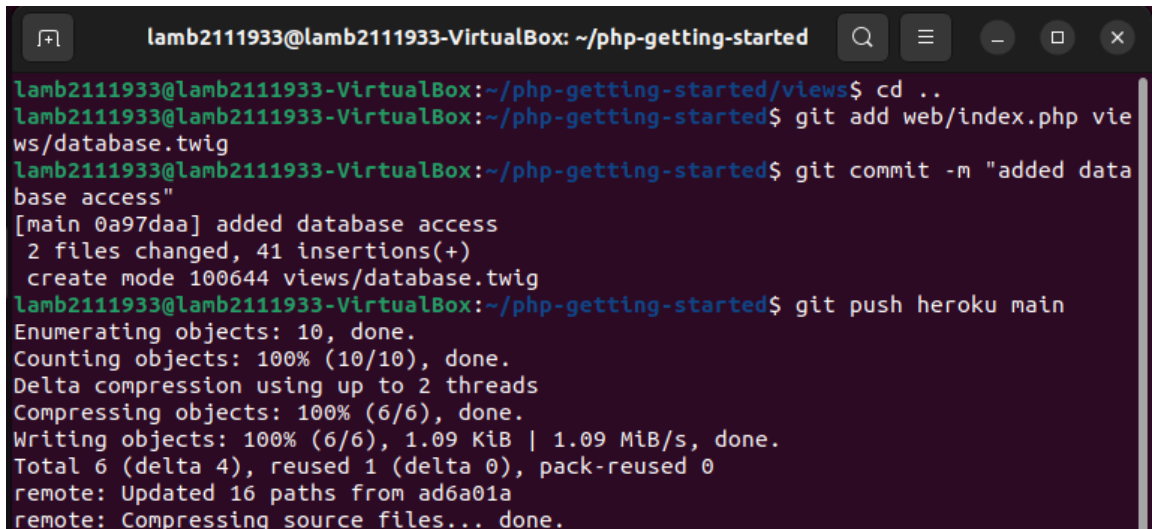
{% block content %}
<p>Got these rows from the database:</p>

<ul>
{% for n in names %}
  <li> {{ n.name }} </li>
{% else %}
  <li><em>No entries yet!</em></li>
{% endfor %}
</ul>

{% endblock %}
```

Create this **database.twig** template file inside the app's **views/** directory

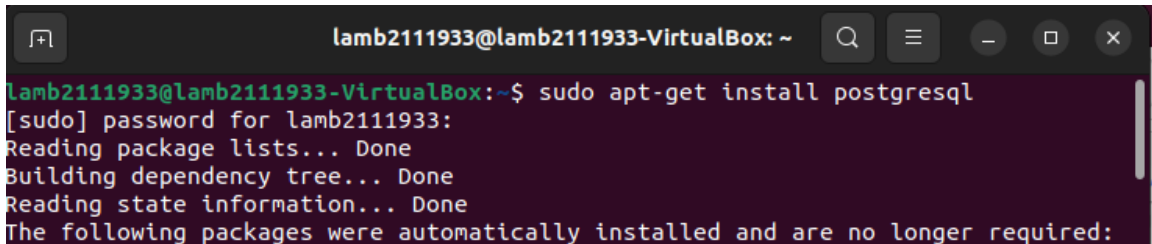
```
$ cd ..
$ git add web/index.php views/database.twig
$ git commit -m "added database access"
$ git push heroku main
```



```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started/views$ cd ..
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ git add web/index.php views/database.twig
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ git commit -m "added database access"
[main 0a97daa] added database access
2 files changed, 41 insertions(+)
create mode 100644 views/database.twig
lamb2111933@lamb2111933-VirtualBox:~/php-getting-started$ git push heroku main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.09 KiB | 1.09 MiB/s, done.
Total 6 (delta 4), reused 1 (delta 0), pack-reused 0
remote: Updated 16 paths from ad6a01a
remote: Compressing source files... done.
```

Deploy this app once more time

```
$ sudo apt-get install postgresql
```

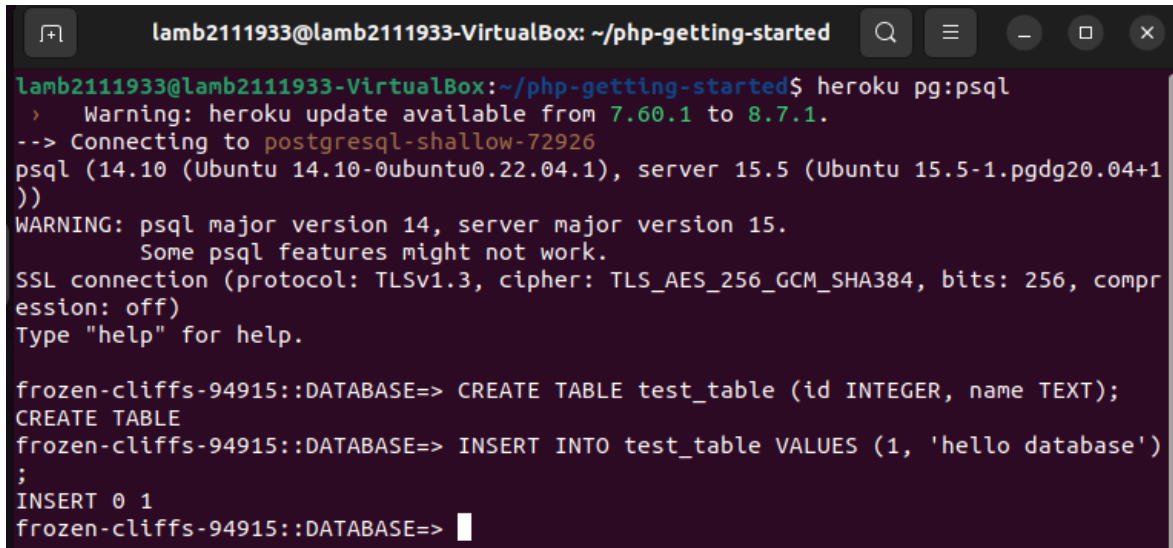


```
lamb2111933@lamb2111933-VirtualBox: ~
lamb2111933@lamb2111933-VirtualBox:~$ sudo apt-get install postgresql
[sudo] password for lamb2111933:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

Install **Postgresql**

Connect to the database and use some commands:

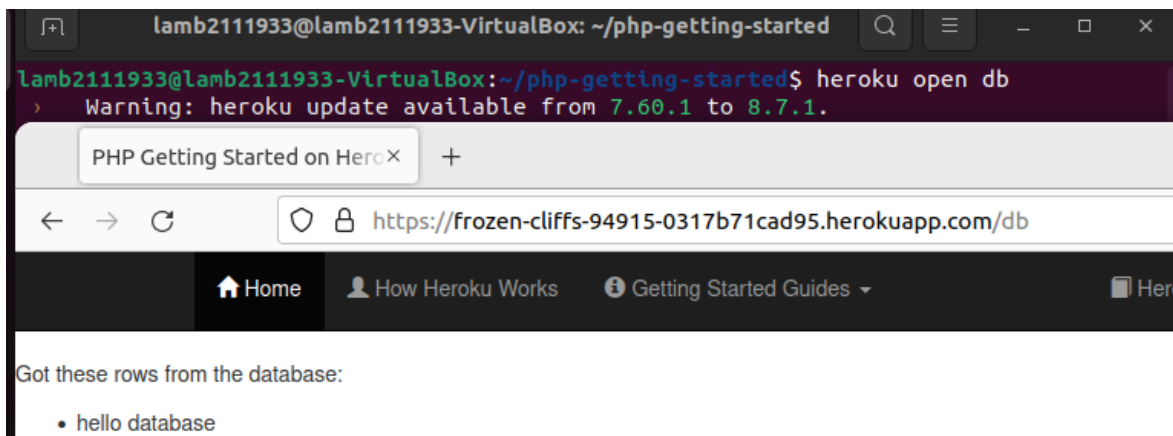
```
$ heroku pg:psql
```

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started'. The user runs 'heroku pg:psql'. The terminal shows a warning about a Heroku update, then connects to a PostgreSQL database. It displays the version (14.10) and server version (15.5). A warning indicates that some features might not work due to the version mismatch. The user then runs 'CREATE TABLE test_table (id INTEGER, name TEXT);' and 'INSERT INTO test_table VALUES (1, 'hello database');'. The terminal shows the successful execution of these commands.

```
lamb2111933@lamb2111933-VirtualBox: ~/php-getting-started$ heroku pg:psql
> Warning: heroku update available from 7.60.1 to 8.7.1.
--> Connecting to postgresql-shallow-72926
psql (14.10 (Ubuntu 14.10-0ubuntu0.22.04.1), server 15.5 (Ubuntu 15.5-1.pgdg20.04+1))
WARNING: psql major version 14, server major version 15.
        Some psql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

frozen-cliffs-94915::DATABASE=> CREATE TABLE test_table (id INTEGER, name TEXT);
CREATE TABLE
frozen-cliffs-94915::DATABASE=> INSERT INTO test_table VALUES (1, 'hello database')
;
INSERT 0 1
frozen-cliffs-94915::DATABASE=> 
```

Use some commands like **CREATE** or **Insert** on database



This is the result of the **database page**

2. Deploying a online shopping website to Heroku

2.1. Deploying an online-shopping website to run locally on your computer

- Install apache, php and heroku

```
$sudo apt update
```

```
$sudo apt install apache2 mysql-server mysql-client php
libapache2-mod-php php-mysql php-json php-xml php-mbstring
php-curl php-zip php-memcached php-redis php-apcu curl
php-cli php-mbstring unzip -y
```

(have done in exercise 1.)

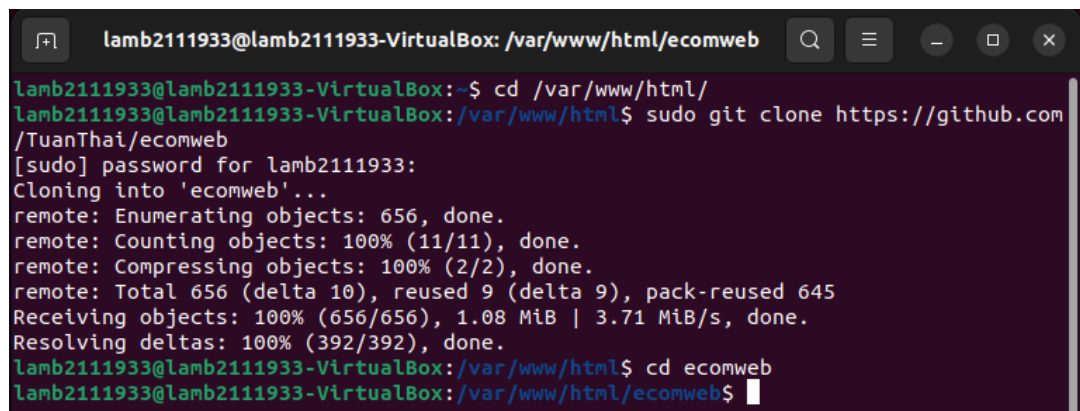
- Install composer (have done in exercise 1.)

```
$cd ~ && curl -sS https://getcomposer.org/installer -o
composer-setup.php
$HASH=`curl -sS https://composer.github.io/installer.sig`
$sudo php composer-setup.php --install-dir=/usr/local/bin
--filename=composer
```

(have done in exercise 1.)

- Clone the source of the website

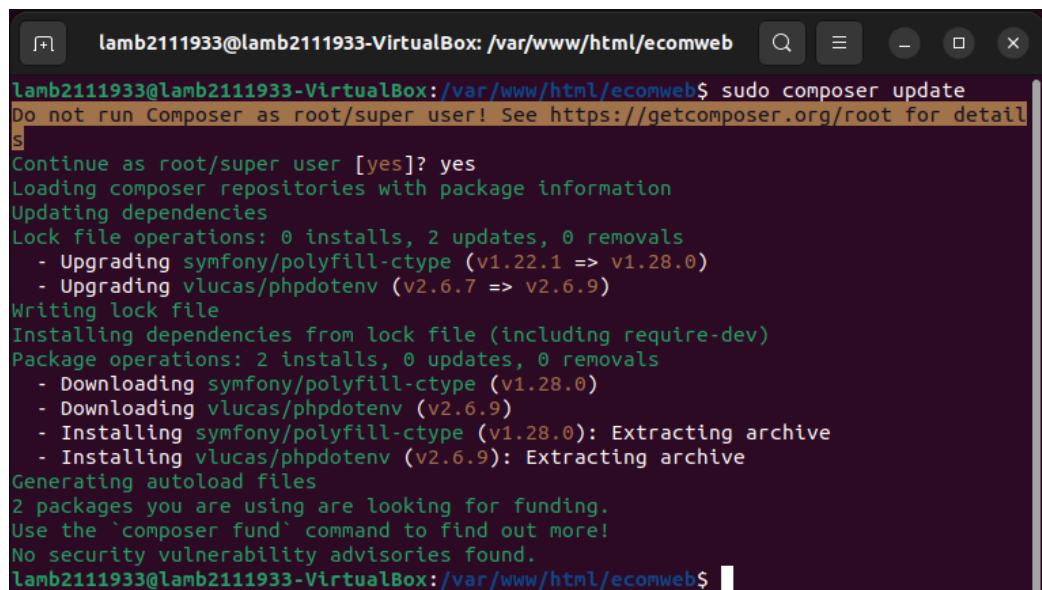
```
$cd /var/www/html/
$sudo git clone https://github.com/TuanThai/ecomweb
$cd ecomweb
```

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: /var/www/html/ecomweb'. The user runs 'cd /var/www/html/' and then 'sudo git clone https://github.com/TuanThai/ecomweb'. The terminal shows the progress of cloning the repository, including enumerating objects, counting objects, compressing objects, and receiving objects. The process completes successfully, and the user is prompted to enter a password for 'lamb2111933'. After cloning, the user runs 'cd ecomweb' and the prompt changes to '/var/www/html/ecomweb\$'.

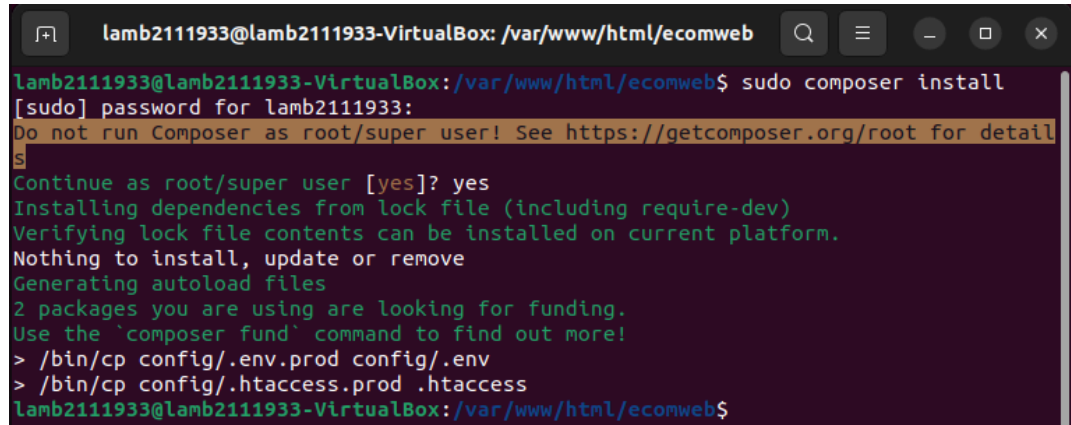
Create a local repository

- Install required modules using composer

```
$sudo composer update
```

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: /var/www/html/ecomweb'. The user runs 'sudo composer update'. The terminal displays a warning: 'Do not run Composer as root/super user! See https://getcomposer.org/root for details'. The user responds 'yes' to 'Continue as root/super user [yes]?'. The terminal then shows the process of loading composer repositories, updating dependencies, and installing dependencies from the lock file. It lists the packages being updated: 'symfony/polyfill-ctype' (v1.22.1 => v1.28.0) and 'vlucas/phpdotenv' (v2.6.7 => v2.6.9). The process completes successfully, and the user is prompted to enter a password for 'lamb2111933'. The prompt changes to '/var/www/html/ecomweb\$'.

```
$sudo composer install
```

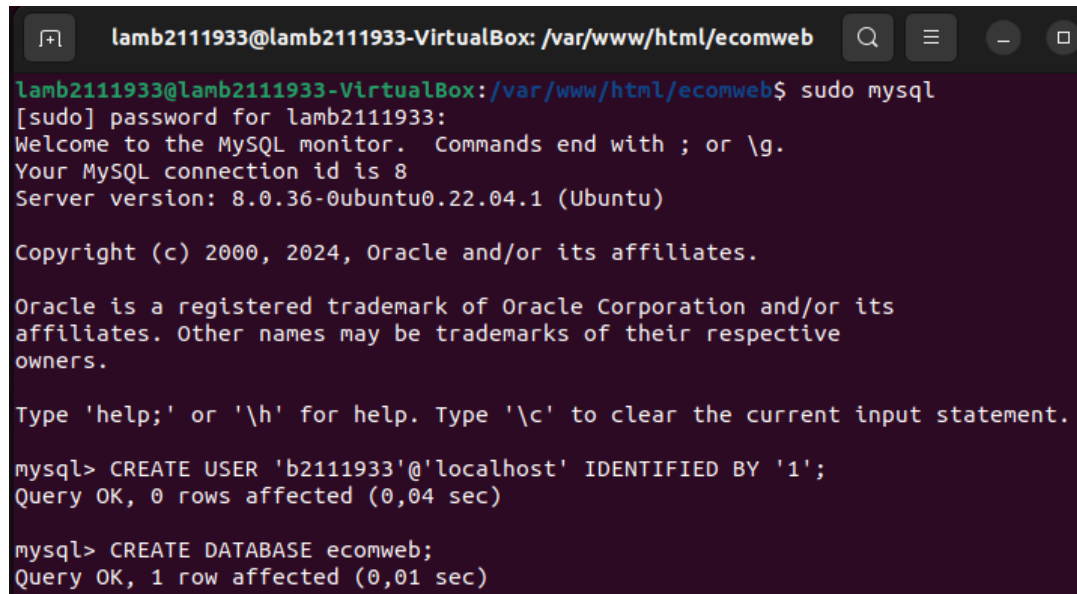


```
lamb2111933@lamb2111933-VirtualBox: /var/www/html/ecomweb$ sudo composer install
[sudo] password for lamb2111933:
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Continue as root/super user [yes]? yes
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Nothing to install, update or remove
Generating autoload files
2 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> /bin/cp config/.env.prod config/.env
> /bin/cp config/.htaccess.prod .htaccess
lamb2111933@lamb2111933-VirtualBox: /var/www/html/ecomweb$
```

Install required modules using **composer**

- Create a mysql user and import database

```
$sudo mysql
mysql>CREATE USER 'tmtuan'@'localhost' IDENTIFIED BY 'password';
mysql>CREATE DATABASE ecomweb;
mysql>GRANT ALL PRIVILEGES ON ecomweb.* TO 'tmtuan'@'localhost';
mysql>use ecomweb;
mysql>source ./config/db_dump.sql;
mysql>exit;
```



```
lamb2111933@lamb2111933-VirtualBox: /var/www/html/ecomweb$ sudo mysql
[sudo] password for lamb2111933:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'b2111933'@'localhost' IDENTIFIED BY '1';
Query OK, 0 rows affected (0,04 sec)

mysql> CREATE DATABASE ecomweb;
Query OK, 1 row affected (0,01 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON ecomweb.* TO 'b2111933'@'localhost';
Query OK, 0 rows affected (0,01 sec)

mysql> use ecomweb;
Database changed
mysql> source ./config/db_dump.sql;
Query OK, 0 rows affected (0,00 sec)

Query OK, 0 rows affected (0,00 sec)

Query OK, 0 rows affected (0,00 sec)

Query OK, 0 rows affected, 1 warning (0,00 sec)

Query OK, 0 rows affected, 2 warnings (0,03 sec)

Query OK, 0 rows affected, 1 warning (0,00 sec)

Query OK, 0 rows affected, 2 warnings (0,03 sec)
```

```
Query OK, 1 row affected (0,00 sec)

Query OK, 1 row affected (0,00 sec)

Query OK, 1 row affected (0,00 sec)

Query OK, 0 rows affected (0,01 sec)

mysql> exit
Bye
lamb2111933@lamb2111933-VirtualBox: /var/www/html/ecomweb$
```

Create a **mysql user** and **import database** from the repository above

- Modify database configuration

```
$sudo nano ./config/.env
#Modify the content of .env file
export STORE_URL='http://localhost/ecomweb'
...
export DB_HOSTNAME='localhost'
export DB_USERNAME='tmtuan'
export DB_PASSWORD='password'
export DB_DATABASE='ecomweb'
export DB_PORT='3306'
```



```
lamb2111933@lamb2111933-VirtualBox: /var/www/html/ecomweb
GNU nano 6.2                               ./config/.env

# Database configuration
#####
export DB_HOSTNAME='localhost'
export DB_USERNAME='b2111933'
export DB_PASSWORD='1'
export DB_DATABASE='ecomweb'
export DB_PORT='3306'
export DB_SOCKET=''
```

Modify database configuration

- Access the website using a web browser: <http://localhost/ecomweb>
(Take a screenshot of the website homepage)



Menu

- [Home](#)
- [Shop](#)
 - [Mobiles](#)
 - [Cameras](#)
 - [Books](#)
- [My Account](#)
 - [My Orders](#)
 - [My Wishlist](#)
 - [Update Address](#)
 - [Sign In](#)
- [Contact](#)

Your shopping cart is empty. It's pretty lonely over here, why not add something to it?

Type Search text here...

Shop

Browse our collection of Products to find just what you are looking for.

[Canon EOS 1300D 18MP Digital SLR Camera \(Black\) with 18-55mm ISII Lens, 16GB Card and Carry Case](#)

This is the **online-shopping website** that run locally

2.2. Deploying the website to Heroku

- Clone the source of the website

```
$cd ~ && git clone https://github.com/TuanThai/ecomweb &&
cd ecomweb
```

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb
lamb2111933@lamb2111933-VirtualBox:/var/www/html/ecomweb$ cd ~ && git clone https://github.com/TuanThai/ecomweb && cd ecomweb
Cloning into 'ecomweb'...
remote: Enumerating objects: 656, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 656 (delta 10), reused 9 (delta 9), pack-reused 645
Receiving objects: 100% (656/656), 1.08 MiB | 3.51 MiB/s, done.
Resolving deltas: 100% (392/392), done.
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$
```

Clone the source code to your local machine

- Create heroku application

```
$sudo snap install --classic heroku
```

```
$heroku login
```

if the web browser cannot open automatically, please run it manually; then copy the log in link to the address bar.

(have done in exercise 1.)

```
$heroku create <student_ID>
```

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$ heroku create b2111933
> Warning: heroku update available from 7.60.1 to 8.7.1.
Creating b2111933... done
https://b2111933-6429ef4d4802.herokuapp.com/ | https://git.heroku.com/b2111933.git
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$
```

Create an application with heroku

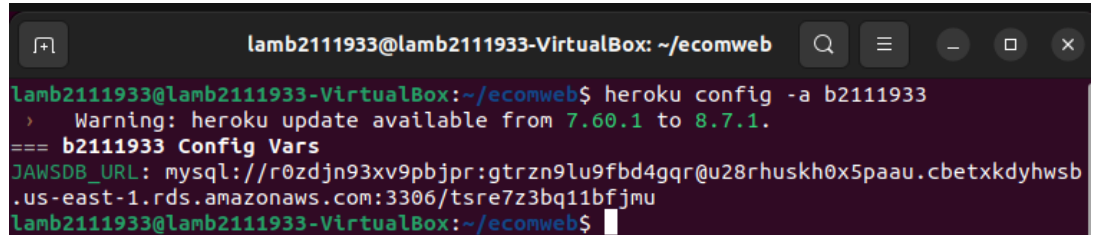
- Create database and import website data

```
$heroku addons:create -a <student_ID> jawsdb:kitefin
```

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$ heroku addons:create -a b2111933 jawsdb:kitefin
> Warning: heroku update available from 7.60.1 to 8.7.1.
Creating jawsdb:kitefin on b2111933... free
Resource is being provisioned.
jawsdb-clean-67282 is being created in the background. The app will restart when complete...
Use heroku addons:info jawsdb-clean-67282 to check creation progress
Use heroku addons:docs jawsdb to view documentation
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$
```

Create database

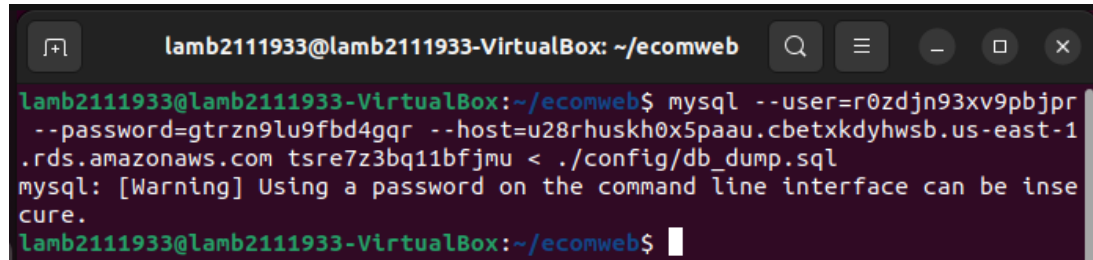
```
$heroku config -a <student_ID>
#You will get the database information as below:
#JAWSDB_URL:mysql://<username>:<password>@<server_name>/<database_name>?reconnect=true
```

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: ~/ecomweb'. The command 'heroku config -a b2111933' has been executed. The output shows a warning about a heroku update and then displays the configuration variables for 'b2111933'. The JAWSDB_URL is shown as 'mysql://r0zdn93xv9pbjpr:gtrzn9lu9fbd4gqr@u28rhuskh0x5paau.cbetxkdyhwsb.us-east-1.rds.amazonaws.com:3306/tsre7z3bq11bfjmu'.

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb$ heroku config -a b2111933
> Warning: heroku update available from 7.60.1 to 8.7.1.
=== b2111933 Config Vars
JAWSDB_URL: mysql://r0zdn93xv9pbjpr:gtrzn9lu9fbd4gqr@u28rhuskh0x5paau.cbetxkdyhwsb.us-east-1.rds.amazonaws.com:3306/tsre7z3bq11bfjmu
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb$
```

Get the database information

```
$mysql --user=username --password=password
--host=server_name database_name < ./config/db_dump.sql
```

A terminal window titled 'lamb2111933@lamb2111933-VirtualBox: ~/ecomweb'. The command 'mysql --user=r0zdn93xv9pbjpr --password=gtrzn9lu9fbd4gqr --host=u28rhuskh0x5paau.cbetxkdyhwsb.us-east-1.rds.amazonaws.com tsre7z3bq11bfjmu < ./config/db_dump.sql' has been executed. The output shows a warning about using a password on the command line interface.

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb$ mysql --user=r0zdn93xv9pbjpr --password=gtrzn9lu9fbd4gqr --host=u28rhuskh0x5paau.cbetxkdyhwsb.us-east-1.rds.amazonaws.com tsre7z3bq11bfjmu < ./config/db_dump.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb$
```

Import data of the website

- Modify database configuration

```
$nano ./config/.env.prod

#Modify the content of .env file
export STORE_URL='<Your Heroku APP URL>'
...
export DB_HOSTNAME='server_name'
export DB_USERNAME='username'
export DB_PASSWORD='password'
export DB_DATABASE='database_name'
export DB_PORT='3306'
```

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb
GNU nano 6.2                               ./config/.env.prod *
# export ADDRESS_STATE=''
# export ADDRESS_COUNTRY=''
# export ADDRESS_POSTAL_CODE=''

# Database configuration
#####
export DB_HOSTNAME='u28rhuskh0x5paau.cbetxkdyhwsb.us-east-1.rds.amazonaws'
export DB_USERNAME='r0zdjn93xv9pbjpr'
export DB_PASSWORD='gtrzn9lu9fbd4gqr'
export DB_DATABASE='tsre7z3bq11bfjmu'
export DB_PORT='3306'
export DB_SOCKET=''
```

Modify database **configuration**

- Push the source code to heroku

```
$git add .
```

```
$git commit -m "Modify file .env.prod" # Cẩn thận lỗi dấu  
nháy
```

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$ git add .
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$ git commit -m "Modify file .e
nv.prod"
[master 0cc7121] Modify file .env.prod
1 file changed, 6 insertions(+), 6 deletions(-)
```

Add **modified files** to **init**

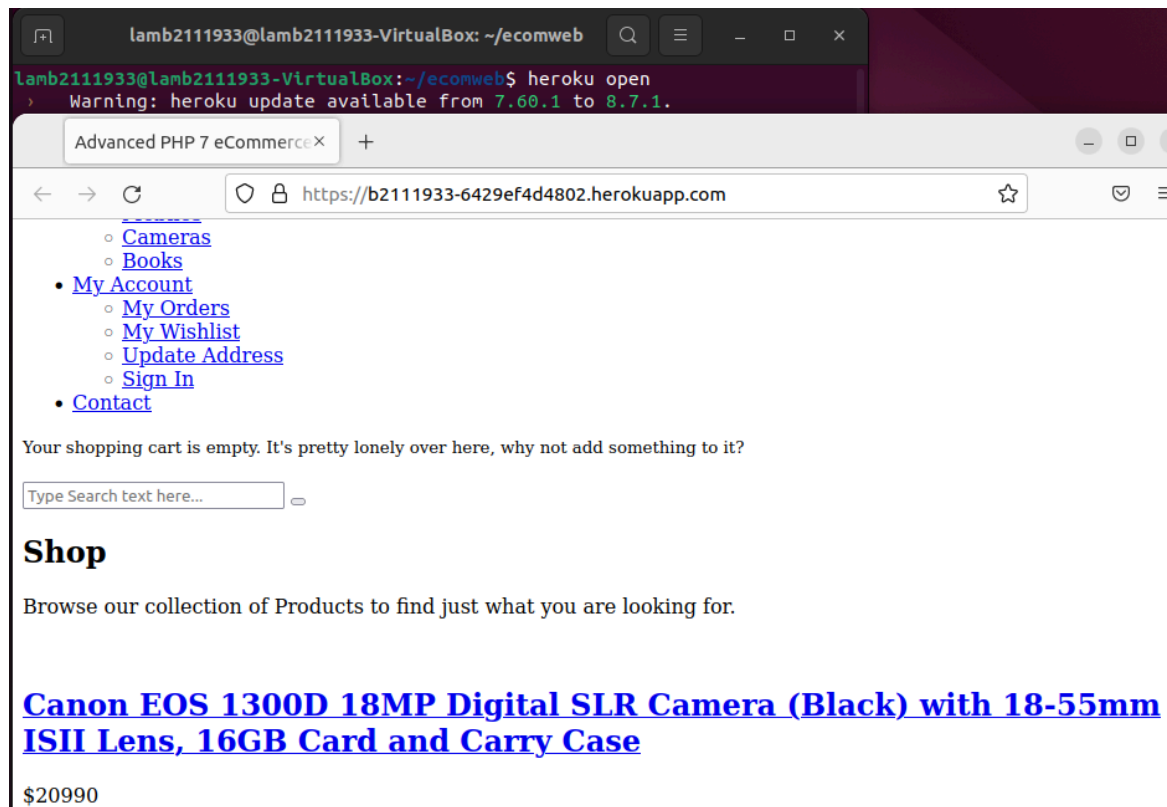
```
$git push heroku main
```

```
lamb2111933@lamb2111933-VirtualBox: ~/ecomweb
lamb2111933@lamb2111933-VirtualBox:~/ecomweb$ git push heroku master
Enumerating objects: 651, done.
Counting objects: 100% (651/651), done.
Delta compression using up to 2 threads
Compressing objects: 100% (246/246), done.
Writing objects: 100% (651/651), 1.08 MiB | 782.00 KiB/s, done.
Total 651 (delta 394), reused 645 (delta 392), pack-reused 0
remote: Resolving deltas: 100% (394/394), done.
remote: 2024/02/03 05:45:30 Rollbar error: empty token
remote: Updated 123 paths from 2fe78b8
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-22 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> PHP app detected
```

Push the source code to **heroku** (I used **master** branch instead)

- Launch your application

\$heroku open



The result

This is my website URL: <https://b2111933-6429ef4d4802.herokuapp.com/>

(Take a screenshot of your website homepage on Heroku + and add your website URL here)

---END---