# Hosting on Digital Ocean

1. Create an account at https://www.digitalocean.com/

2. Go to https://education.github.com/pack/offers

   If you are currently logged into your GitHub account, it should recognize that you already have a Student Developer Pack. Scroll down to the Digital Ocean section in the list and get an offer code.

3. Login to your Digital Ocean account. You will have to enter a valid payment method, but at the bottom select "Have a Promo Code?" and enter in the code you received in Step 2.

4. Once you've applied the promo code and completed sign-up, select "Create a New Droplet".

   Droplet Options -
   Choose an image: Ubuntu 16.04.2 x64 (should be selected by default)
   Choose a size: $5/mo (just choose the smallest, it should be fine)
   Add block storage: (skip)
   Choose a datacenter region: New York (should be selected by default)
   Select additional options: (skip)
   Add your SSH keys: (skip)
   Finalize and Create: 1 Droplet, Enter whatever name you want

   Select "Create"

   You should receive an email listing your Droplet Name, IP Address, Username, and Default Password.

5. Use ssh to login to your Droplet on Digital Ocean: ssh root@<your_ip_address>
   You will be prompted to change your password.

6. Update and install apache:
   sudo apt-get update
   sudo apt-get install apache2

7. Install Web Server Gateway Interface (WSGI):
   sudo apt-get install libapache2-mod-wsgi python-dev

   Enable mod_wsgi with:
   sudo a2enmod wsgi

8. Create/Download a Flask Application:

   Move to the /var/www/ directory: cd /var/www
   Clone your project from GitHub: git clone https://github.com/yourgithub/cs329e-idb.git
   Move into your project directory: cd cs329e-idb

   Set up a virtual environment:
   sudo apt-get install python-pip
   sudo pip install virtualenv
   sudo virtualenv –p python3 venv

   Now you can use your virtual environment as described in the Flask tutorial. At this point you'll either want to install Flask or install your needed python packages from requirements.txt. Use either pip install Flask or pip install –r requirements.txt (with the virtual environment activated).

   Test your installation by running python app.py within the virtual environment. If you get a message saying "Running on http://localhost:5000/" or "Running on http://127.0.0.1:5000/" your installation was successful.

   The setup will later require that your main project folder be a module. To do this in Python you will need to add a file called __init__.py, even if it is an empty file. Do this with: touch /var/www/cs329e-idb/__init__.py.

9. Configure and Enable a New Virtual Host:

   In your droplet terminal enter: sudo nano /etc/apache2/sites-available/cs329e-idb.conf

   Add the following lines of code to the file to configure the virtual host. Be sure to change the ServerName to your domain or cloud server's IP address.

```
<VirtualHost *:80>
        ServerName mywebsite.com
        ServerAdmin admin@mywebsite.com
        WSGIScriptAlias / /var/www/cs329e-idb/app.wsgi
        <Directory /var/www/cs329e-idb/>
           Order allow,deny
           Allow from all
        </Directory>
        Alias /static /var/www/cs329e-idb/static
        <Directory /var/www/cs329e-idb/static/>
           Order allow,deny
           Allow from all
        </Directory>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

   Save and close the file. Enable the virtual host with: sudo a2ensite cs329e-idb

10. Create the .wsgi File:

    Apache uses the .wsgi file to serve the Flask app. Move to the /var/www/cs329e-idb directory and create a file named app.wsgi with the following commands:

```
cd /var/www/cs329e-idb
sudo nano app.wsgi
```

    Add the following lines to the file:

```
#!/usr/bin/python
import sys
import logging
logging.basicConfig(stream=sys.stderr)
sys.path.insert(0,"/var/www/cs329e-idb/")

from app import app as application
application.secret_key = 'Add your secret key'
```

11. Restart Apache: sudo service apache2 restart
12. Update app.run( ) in app.py:

    At the bottom of app.py change the call to app.run( ) to app.run('your_ip_address', '80')

To check if everything is working properly, activate your virtual environment and run python app.py & (the '&' will run your program in the background so you can still interact in the terminal). Navigate in a web browser to the domain name or IP address that you entered in your virtual host configuration.

If you get a message saying "Address already in use" or you want to stop and restart your application, use the following steps:

- Enter the command: lsof –i:80
  This will display all currently running processes. If your website is running properly it will be listed with the command of "python" and <your_server's_IP>:http (LISTEN) as its name. When initially setting up your server, if you received an "address already in use" message you will probably see several "apache2" commands listed.
- Kill currently running processes: kill <PID>
  Each running process in the list shown after the lsof command will have a PID number. Use that number to stop that process with the kill command. You want to stop any processes associated with python or apache2.
- Restart web application: python app.py & (within the virtual environment)
  Move back into your main project directory, activate the virtual environment, and start the web application. You should now be able to access your website in a web browser using its IP address or domain name.

Links:

- https://github.com/brpowell/flask-example/wiki/Hosting-on-Digital-Ocean
- https://www.digitalocean.com/community/tutorials/how-to-deploy-a-flask-application-on-an-ubuntu-vps