ESTABLISHING AN AWS LIGHTSAIL DJANGO INSTANCE

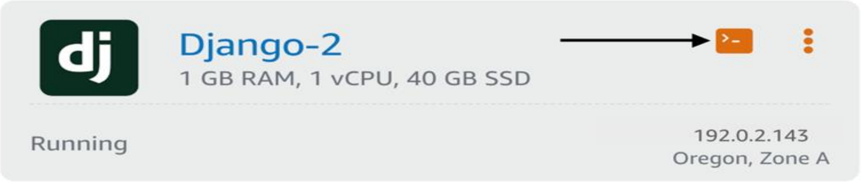
Get started!

1. Establish a free AWS account. Record UserID and PW.
2. Navigate to: https://console.aws.amazon.com/iam/home#/security_credentials$access_key and set up and download a set of access keys (access key ID and secret access key).

From this tutorial: https://aws.amazon.com/getting-started/hands-on/deploy-python-application/

3. If you insist, open the VPS instance (Django-1 or whatever you named it) click the orange

   >_ icon.



Once the web CLI or PuTTY is open, run the green commands:

4. Create the projects directory and give the current system user write permissions to the new directory.

```
sudo mkdir /opt/bitnami/projects && sudo chown $USER /opt/bitnami/projects
```

5. Create the projects directory and give the current system user write permissions to the new directory.

```
cd /opt/bitnami/projects && django-admin startproject tutorial
```

6. Change into the tutorial project directory, and create a new hello_world app.

```
cd tutorial && python3 manage.py startapp hello_world
```

7. Open the views.py file using the Vim text editor:

```
sudo vim /opt/bitnami/projects/tutorial/hello_world/views.py
```

8. Type `:1,$d` and press `Enter` to select all, and delete the existing contents of the views.py file.

9. Press I to edit the file, and add the following code to the views.py file:

```
from django.http import HttpResponse

def index(request):
      return HttpResponse("Hello, world")
```

10. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.

11. Create a urls.py file for your application in the same directory as the views.py file:

```
sudo vim /opt/bitnami/projects/tutorial/hello_world/urls.py
```

Press I to edit the file, and add the following code:

```
from django.urls import path

from . import views


urlpatterns = [

     path('', views.index, name='index'),

]
```

12. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.

13. Enter the following command to open the urls.py file:

```
sudo vim /opt/bitnami/projects/tutorial/tutorial/urls.py
```

14. Type `:1,$d` and press `Enter` to select all, and delete the existing contents of the urls.py file.

15. Press I to edit the file, and add the following code to the urls.py file:

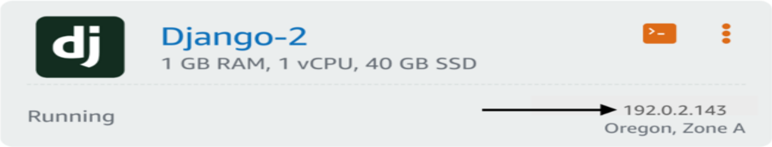
```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('', include('hello_world.urls')),
    path('admin/', admin.site.urls),
]
```

16. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.

Testing the new Django instance:

17. Return to the Lightsail home page and click on the name of your Django instance.
18. Click on the Networking tab.
19. Under Firewall click Add Rule.
20. Under Port Range enter `8000` and click Create.
21. Return to the Lightsail home page and make a note of the instance IP address.



22. Return to the terminal session, and enter the following command to open the settings.py file:

```
sudo vim /opt/bitnami/projects/tutorial/tutorial/settings.py
```

22. Press I and change the ALLOWED_HOSTS line to include the IP Address of your Lightsail instance:

Note: When editing your settings.py file, be sure to replace 192.0.2.143 with the public IP address of your instance.

```
ALLOWED_HOSTS = ['192.0.2.143']
```

23. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.

24. Start Django's web server on port 8000 with:

```
cd /opt/bitnami/projects/tutorial/ && python3 manage.py runserver 0.0.0.0:8000
```

25. Point your browser at 192.0.2.143:8000 and observe "Hello world!".

26. In SSH window, `CTRL C` to stop server.

Host the Application Using Apache

It's not recommended to run production applications using Django's built-in server. So, in this final section you'll configure Apache to serve your application.

You'll do this by configuring the application to use Web Services Gateway Interface (WSGI), and then create an Apache virtual host (vHost).

You should still be executing these commands in the Lightsail browser-based SSH client window.

27. Type the following command to edit the wsgi.py file:

```
sudo vim /opt/bitnami/projects/tutorial/tutorial/wsgi.py
```

28. Type `:1,$d` and press `Enter` to select all, and delete the existing contents of the wsgi.py file.

29. Press I to edit, and add the following code to the wsgi.py file:

```
import os
import sys

sys.path.append('/opt/bitnami/projects/tutorial')

os.environ.setdefault("PYTHON_EGG_CACHE",
“/opt/bitnami/projects/tutorial/egg_cache")

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "tutorial.settings")

from django.core.wsgi import get_wsgi_application

application = get_wsgi_application()
```

30. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.
31. Enter the following command to create a conf directory for your application

```
mkdir /opt/bitnami/projects/tutorial/conf
```

32. Enter the following command to create a httpd-app.conf file in the conf directory you just created:

```
sudo vim /opt/bitnami/projects/tutorial/conf/httpd-app.conf
```

33. Press I to edit, and add the following code:

```
<IfDefine !IS_DJANGOSTACK_LOADED>
      Define IS_DJANGOSTACK_LOADED
      WSGIDaemonProcess wsgi-djangostack   processes=2 threads=15    display-
name=%{GROUP}
</IfDefine>

<Directory "/opt/bitnami/tutorial/tutorial">
    Options +MultiViews
    AllowOverride All
    <IfVersion >= 2.3>
        Require all granted
    </IfVersion>

    WSGIProcessGroup wsgi-djangostack

    WSGIApplicationGroup %{GLOBAL}
</Directory>

Alias /tutorial/static "/opt/bitnami/python/lib/python3.8/site-
packages/django/contrib/admin/static/"
```

```
WSGIScriptAlias /tutorial '/opt/bitnami/projects/tutorial/tutorial/wsgi.py'
```

34. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.

35. Enter the following command to create a httpd-prefix.conf file in the conf directory for your application:

```
sudo vim /opt/bitnami/projects/tutorial/conf/httpd-prefix.conf
```

36. Press I to edit, and add the following code:

```
Include "/opt/bitnami/projects/tutorial/conf/httpd-app.conf"
```

37. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.

38. Enter the following command to open the bitnami.conf file:

```
sudo vim /opt/bitnami/apache2/conf/bitnami/bitnami.conf
```

39. Type `:1,$d` and press `Enter` to select all, and delete the existing contents of the bitnami.conf file.

40. Press I to edit, and add the following code to the file:

```
<VirtualHost _default_:80>
    WSGIScriptAlias / /opt/bitnami/projects/tutorial/tutorial/wsgi.py
    <Directory /opt/bitnami/projects/tutorial>
        AllowOverride all
        Require all granted
        Options FollowSymlinks
    </Directory>
    DocumentRoot /opt/bitnami/projects/tutorial
</VirtualHost>
Include "/opt/bitnami/apache/conf/bitnami/bitnami-ssl.conf"
```

41. Press `Esc` then `:wq!` then press `Enter` to save and quit Vim.

42. Create a static IP address and assign it to the Django Instance.

Follow this explainer:

https://lightsail.aws.amazon.com/ls/docs/en_us/articles/lightsail-create-static-ip

43. Enter the following command to restart Apache:

    ```
    sudo /opt/bitnami/ctlscript.sh restart apache
    ```

44. Return to your browser and enter ht tp://your.static.IP.address (without the space in http)
    If you get the Bitnami congratulations screen, go into your browser's setting and temporarily allow insecure content. Then you should see "Hello World" text. If not, you need to go back and check every last entry in the Apache Section. It has to be perfect.

45. You're not done! Htaccess changes, 443 ssl certs… gahh!