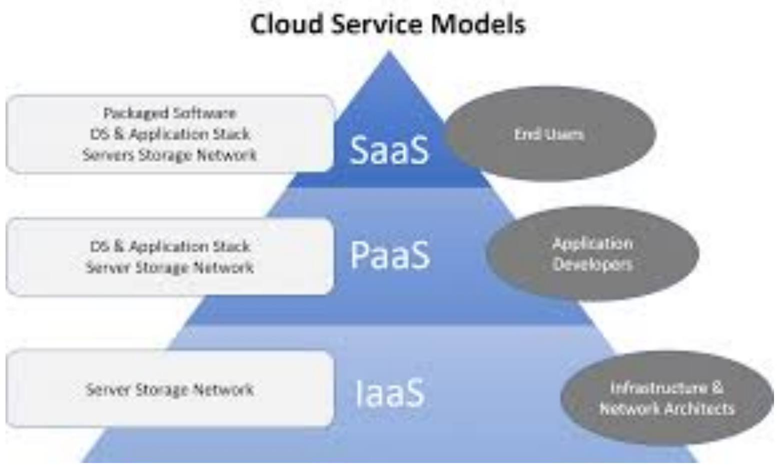


CS352: Cloud Computing Jan-May 2019

SelfieLessActs App

A photo sharing web app being hosted on AWS Cloud.

CS352 | 30-04-2019

SNo	Name	USN	Class/Section
1	Rohan Sharma	01FB16ECS311	F
2	Royal Mehta	01FB16ECS315	F
3	Shashank Shekhar Pathak	01FB16ECS359	F
4	Shikhar Shrestha	01FB16ECS362	F

INTRODUCTION

SelfieLessActs app is a photo sharing web app being hosted on AWS Cloud. It is used to store information about anything that is good for the society that we observe.

Examples of such acts could be :

- Picking up a piece of garbage and dumping it in a garbage can
- Road getting laid in your area
- Someone helping a blind man cross the road.
- Helping your mother at home in the kitchen.

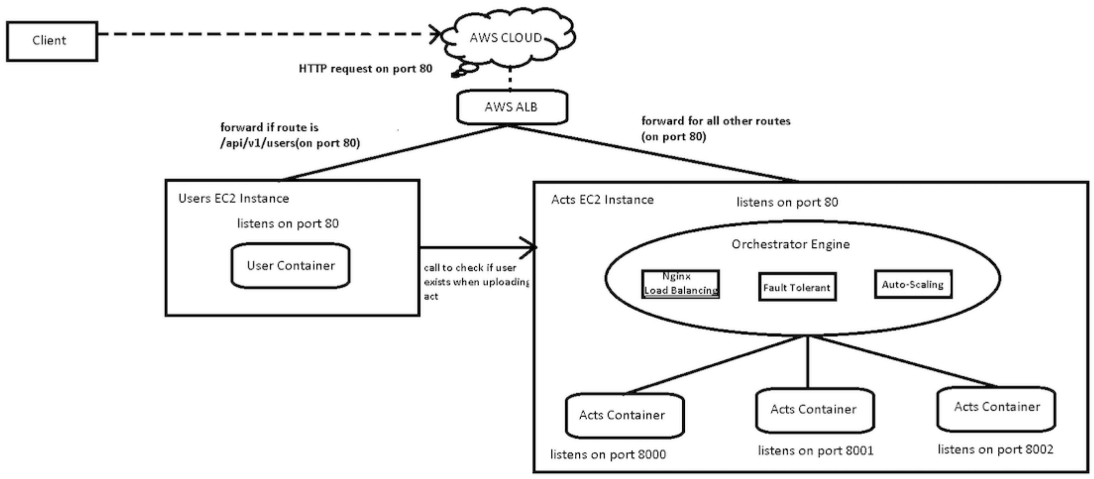
The SelfLessActs application allows the users of the application to upload image of the act with a small caption and categories. They can also view the acts which they have uploaded as well as on selecting a category, all the acts in that category can be viewed in reverse chronological order. Acts can also be liked by the users.

RELATED WORK

- **Flask** - To implement backend server for Acts/Users (https://www.tutorialspoint.com/flask)
- **SQLite3** - DataBase system to store information about Acts. (https://www.tutorialspoint.com/sqlite/sqlite_python.htm)
- **JavaScript** - To implement Client's Web App dynamically. (https://www.w3schools.com/js/)
- **XMLHttpRequest** - To make a request to server to get or post the data. (https://www.w3schools.com/xml/xml_http.asp)
- **Docker/Container** - To implement microservices. (https://www.docker.com/resources/what-container)
- **Nginx** - To implement load balancing among the containers. (https://www.docker.com/resources/what-container)

EXPERIMENTS / DESIGN

We have created two instances on AWS, one for users and one for acts. On each instance, we have created docker container on which acts and users api file runs. Both these instances are accessible from under the same public IP address and also the same port (80).



Img 1. Design for our Application

We have used AWS application load balancer for path based routing which will distribute incoming HTTP requests to one of the two EC2 instances based on the URL route of the request.

In the Acts instance, we have created an orchestration engine which performs load balancing, fault tolerance and auto-scaling. The orchestrator engine runs inside the Acts EC2 instance, listen on port 80 of the public IP, and load balance all HTTP incoming requests equally to every Acts container.

We have used Nginx server for load balancing which runs on port 80 of localhost. We have used round-robin algorithm in Nginx server for distributing the traffic among all the active acts container.

For fault tolerance and auto-scaling, we have written two separate python scripts. We are continuously monitoring the health of the active containers. Whenever we find an unhealthy container, we stop and remove that container and replace it with a new container which runs on the same port as the previous unhealthy container.

For auto-scaling, we use a threshold value to determine when to scale-up and when to scale-down the containers. For this we check the log file of the nginx server which has details of all the requests made. We have used a python timer library to create the timer for 2 minutes. Whenever the first request is made, the timer is started. At every 2 minute interval, we check the no. of requests made and the no. of containers which are active and the orchestrator must increase or decrease the number of Acts containers according to the following rules:

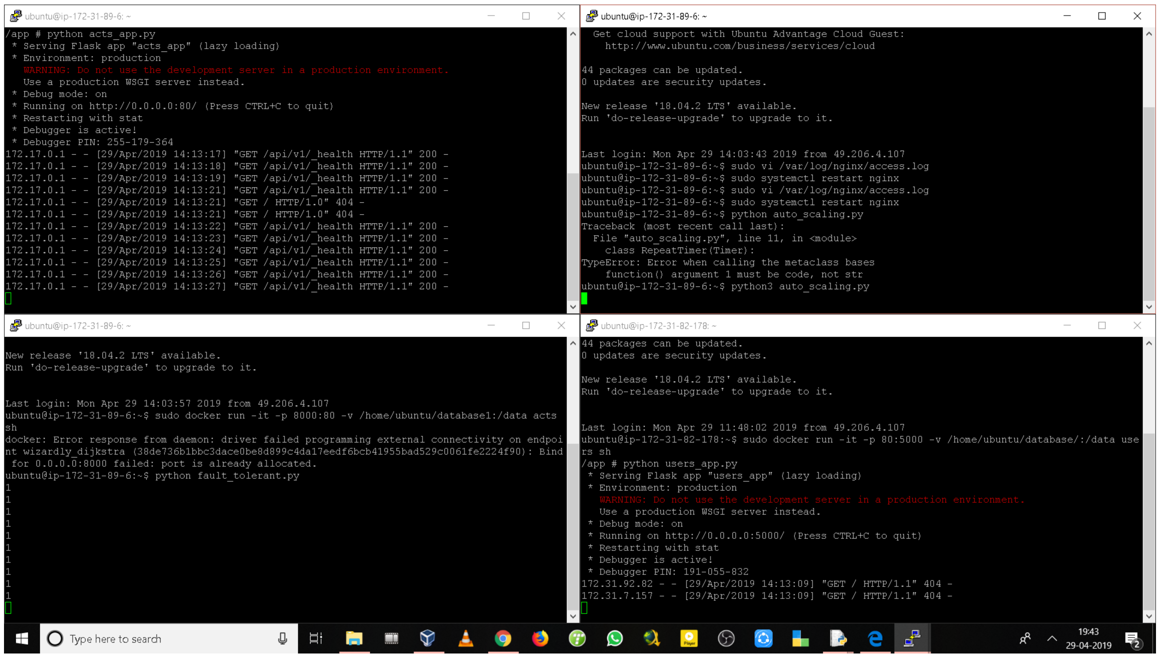
- If the number of requests is less than 20, then only 1 Acts container must be running.
- If the number is >= 20 and < 40, then 2 Acts containers must be running.
- If the number is >= 40 and < 60, then 3 Acts containers must be running.
- and so on…

Using all this information, we update the /etc/nginx/conf.d/default.conf file with the port numbers for the new acts containers, whenever we increase the containers. And when we want to decrease the number of acts containers, we remove the required lines from the configuration file.

In this way, we have implemented load balancing, fault tolerance and auto-scaling

TESTING/RESULTS

We have tested our code for load balancing, fault tolerance and auto-scaling.



```
/app # python acts_app.py
 * Serving Flask app "acts_app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 255-179-364
172.17.0.1 - - [29/Apr/2019 14:13:17] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:18] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:19] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:21] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:21] "GET / HTTP/1.0" 404 -
172.17.0.1 - - [29/Apr/2019 14:13:21] "GET / HTTP/1.0" 404 -
172.17.0.1 - - [29/Apr/2019 14:13:22] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:23] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:24] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:25] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:26] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:13:27] "GET /api/v1/_health HTTP/1.1" 200 -
```

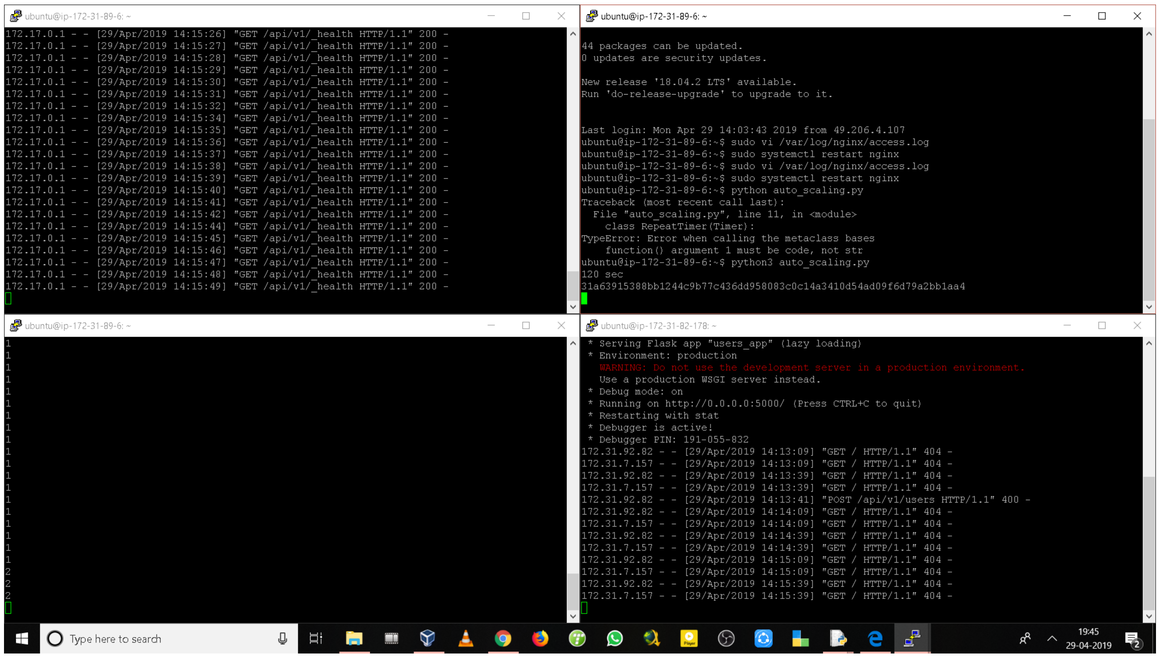
```
  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

44 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Apr 29 14:03:43 2019 from 49.206.4.107
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ python auto_scaling.py
Traceback (most recent call last):
  File "auto_scaling.py", line 11, in <module>
    class RepeatTimer(Timer):
TypeError: Error when calling the metaclass bases
    function() argument 1 must be code, not str
ubuntu@ip-172-31-89-6:~$ python3 auto_scaling.py
```

```

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Apr 29 14:03:57 2019 from 49.206.4.107
ubuntu@ip-172-31-89-6:~$ sudo docker run -it -p 8000:80 -v /home/ubuntu/database1:/data acts
sh
docker: Error response from daemon: driver failed programming external connectivity on endpoi
nt wizardly_dijkstra (38de736b1bbc3dace0be8d899c4da17eedf6bcb41955bad529c0061fe2224f90): Bind
 for 0.0.0.0:8000 failed: port is already allocated.
ubuntu@ip-172-31-89-6:~$ python fault_tolerant.py
1
1
1
1
1
1
1
1
1
1
```

```
44 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Apr 29 11:48:02 2019 from 49.206.4.107
ubuntu@ip-172-31-82-178:~$ sudo docker run -it -p 80:5000 -v /home/ubuntu/database/:/data use
rs sh
/app # python users_app.py
 * Serving Flask app "users_app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 191-055-832
172.31.92.82 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
```

Img 2.

In the beginning, we had only one Acts container which was active.



```
172.17.0.1 - - [29/Apr/2019 14:15:26] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:27] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:28] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:29] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:30] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:31] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:32] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:34] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:35] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:36] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:37] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:38] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:39] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:40] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:41] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:42] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:44] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:45] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:46] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:47] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:48] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:15:49] "GET /api/v1/_health HTTP/1.1" 200 -
```

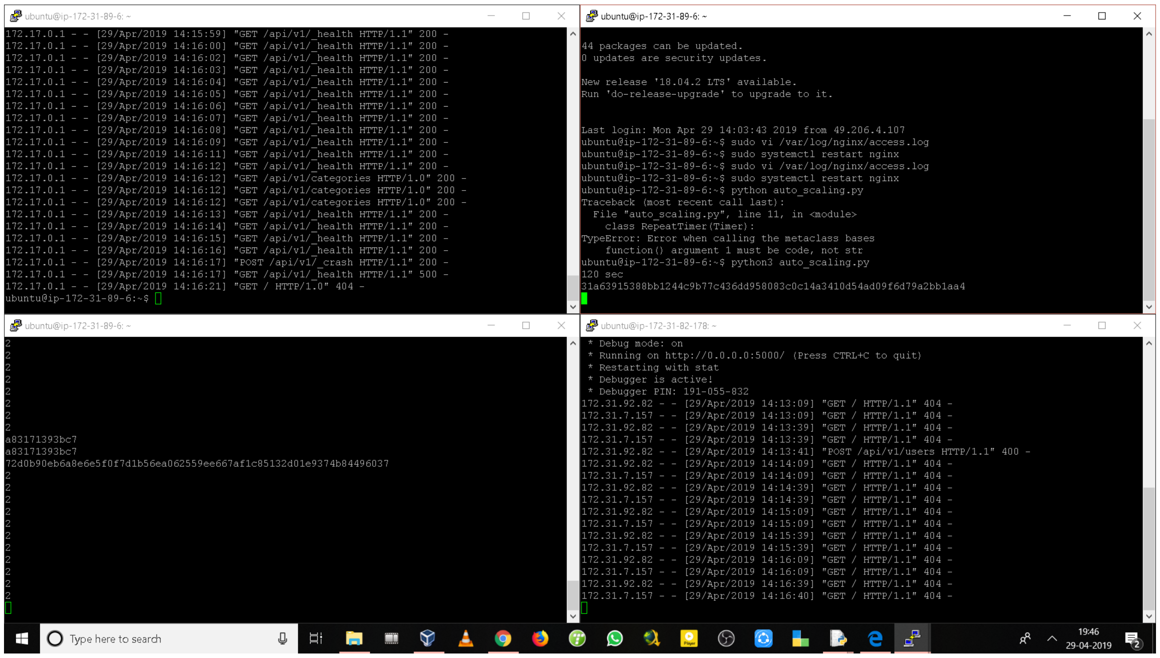
```

44 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Apr 29 14:03:43 2019 from 49.206.4.107
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ python auto_scaling.py
Traceback (most recent call last):
  File "auto_scaling.py", line 11, in <module>
    class RepeatTimer(Timer):
TypeError: Error when calling the metaclass bases
    function() argument 1 must be code, not str
ubuntu@ip-172-31-89-6:~$ python3 auto_scaling.py
120 sec
31a63915388bb1244c9b77c436dd958083c0c14a3410d54ad09f6d79a2bb1aa4
```

```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
2
2
2
```

```
 * Serving Flask app "users_app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 191-055-832
172.31.92.82 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:13:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:13:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:13:41] "POST /api/v1/users HTTP/1.1" 400 -
172.31.92.82 - - [29/Apr/2019 14:14:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:14:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:14:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:14:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:15:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:15:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:15:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:15:39] "GET / HTTP/1.1" 404 -
```

Img 3.

After that 20 requests were made, as a result of which the containers were scaled up and now we have 2 active containers as is visible from Img 3., between which load balancing takes place.



```
ubuntu@ip-172-31-89-6: ~
172.17.0.1 - - [29/Apr/2019 14:15:59] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:00] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:02] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:03] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:04] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:05] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:06] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:07] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:08] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:09] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:11] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:13] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:14] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:15] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:16] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:17] "POST /api/v1/_crash HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:17] "GET /api/v1/_health HTTP/1.1" 500 -
172.17.0.1 - - [29/Apr/2019 14:16:21] "GET / HTTP/1.0" 404 -
ubuntu@ip-172-31-89-6:~$
```

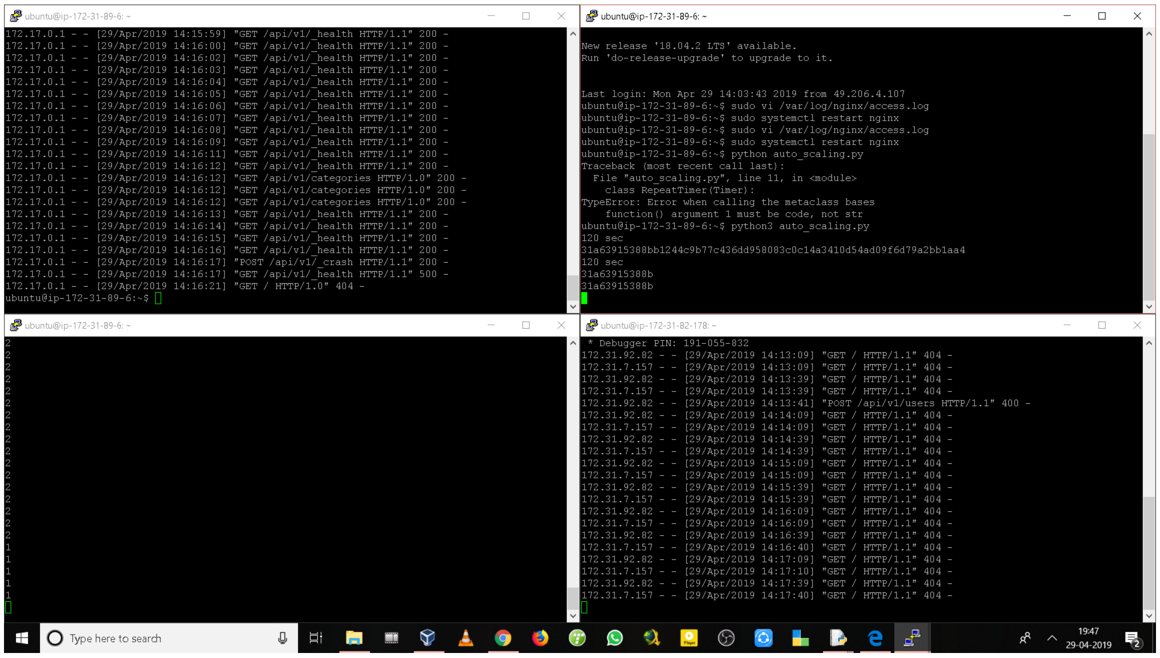
```
ubuntu@ip-172-31-89-6: ~

44 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Apr 29 14:03:43 2019 from 49.206.4.107
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ python auto_scaling.py
Traceback (most recent call last):
  File "auto_scaling.py", line 11, in <module>
    class RepeatTimer(Timer):
TypeError: Error when calling the metaclass bases
    function() argument 1 must be code, not str
ubuntu@ip-172-31-89-6:~$ python3 auto_scaling.py
120 sec
31a63915388bb1244c9b77c436dd958083c0c14a3410d54ad09f6d79a2bb1aa4
```

```
ubuntu@ip-172-31-89-6: ~
2
2
2
2
2
2
2
2
a83171393bc7
a83171393bc7
72d0b90eb6a8e6e5f0f7d1b56ea062559ee667af1c85132d01e9374b84496037
2
2
2
2
2
2
2
2
2
2
2
```

```
ubuntu@ip-172-31-82-178: ~
 * Debug mode: on
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 191-055-832
172.31.92.82 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:13:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:13:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:13:41] "POST /api/v1/users HTTP/1.1" 400 -
172.31.92.82 - - [29/Apr/2019 14:14:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:14:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:14:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:14:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:15:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:15:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:15:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:15:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:16:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:16:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:16:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:16:40] "GET / HTTP/1.1" 404 -
```

Img 4.

After this, the crash api request was sent as a result of which 2nd acts container running on port 8001 was crashed and it was detected by the fault tolerant script which stops and removes this container and in place of it starts a new container running on the same port and we can see the container id of with these containers from the Img 4.



```
ubuntu@ip-172-31-89-6: ~
172.17.0.1 - - [29/Apr/2019 14:15:59] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:00] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:02] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:03] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:04] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:05] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:06] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:07] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:08] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:09] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:11] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:12] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:13] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:14] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:15] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:16] "GET /api/v1/_health HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:17] "POST /api/v1/_crash HTTP/1.1" 200 -
172.17.0.1 - - [29/Apr/2019 14:16:17] "GET /api/v1/_health HTTP/1.1" 500 -
172.17.0.1 - - [29/Apr/2019 14:16:21] "GET / HTTP/1.0" 404 -
ubuntu@ip-172-31-89-6:~$
```

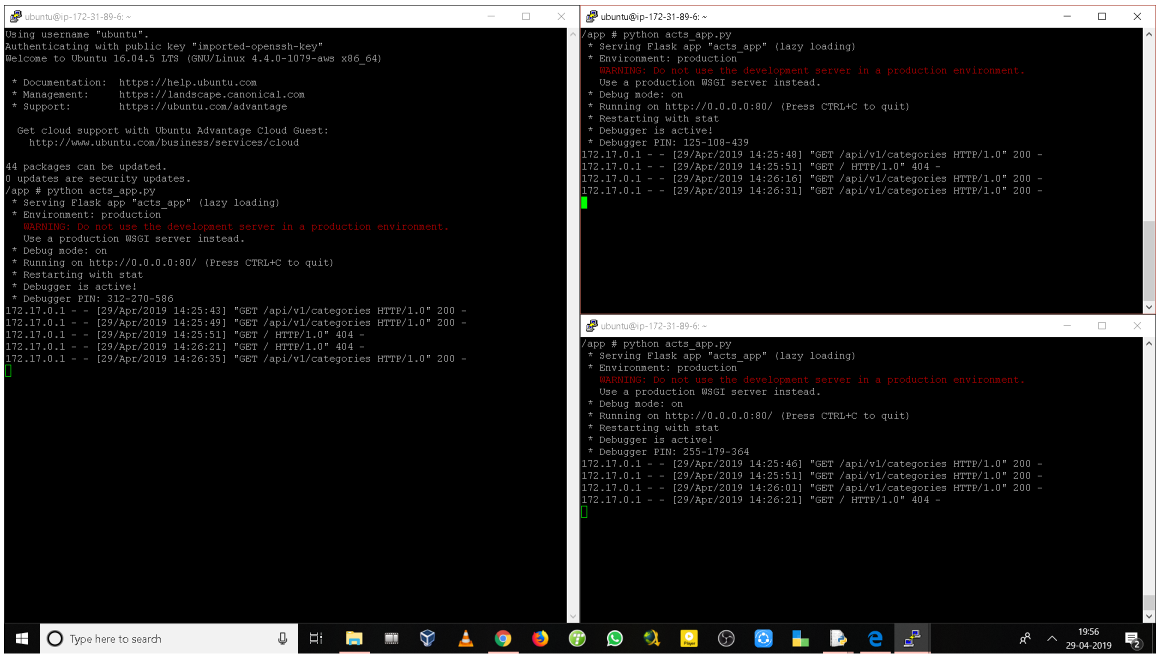
```
ubuntu@ip-172-31-89-6: ~

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Apr 29 14:03:43 2019 from 49.206.4.107
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ sudo vi /var/log/nginx/access.log
ubuntu@ip-172-31-89-6:~$ sudo systemctl restart nginx
ubuntu@ip-172-31-89-6:~$ python auto_scaling.py
Traceback (most recent call last):
  File "auto_scaling.py", line 11, in <module>
    class RepeatTimer(Timer):
TypeError: Error when calling the metaclass bases
    function() argument 1 must be code, not str
ubuntu@ip-172-31-89-6:~$ python3 auto_scaling.py
120 sec
31a63915388bb1244c9b77c436dd958083c0c14a3410d54ad09f6d79a2bb1aa4
120 sec
31a63915388b
31a63915388b
```

```
ubuntu@ip-172-31-89-6: ~
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
1
1
1
1
1
```

```
ubuntu@ip-172-31-82-178: ~
 * Debugger PIN: 191-055-832
172.31.92.82 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:13:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:13:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:13:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:13:41] "POST /api/v1/users HTTP/1.1" 400 -
172.31.92.82 - - [29/Apr/2019 14:14:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:14:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:14:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:14:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:15:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:15:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:15:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:15:39] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:16:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:16:09] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:16:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:16:40] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:17:09] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:17:10] "GET / HTTP/1.1" 404 -
172.31.92.82 - - [29/Apr/2019 14:17:39] "GET / HTTP/1.1" 404 -
172.31.7.157 - - [29/Apr/2019 14:17:40] "GET / HTTP/1.1" 404 -
```

Img 5.

Now since only 6 requests were made in this next two minutes, so the containers are scaled-down and the 2nd container is removed. As we can see from the Img 5. there is only one active container now.



```
ubuntu@ip-172-31-89-6: ~
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1079-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

44 packages can be updated.
0 updates are security updates.
/app # python acts_app.py
 * Serving Flask app "acts_app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 312-270-586
172.17.0.1 - - [29/Apr/2019 14:25:43] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:25:49] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:25:51] "GET / HTTP/1.0" 404 -
172.17.0.1 - - [29/Apr/2019 14:26:21] "GET / HTTP/1.0" 404 -
172.17.0.1 - - [29/Apr/2019 14:26:35] "GET /api/v1/categories HTTP/1.0" 200 -
```

```
ubuntu@ip-172-31-89-6: ~
/app # python acts_app.py
 * Serving Flask app "acts_app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 125-108-439
172.17.0.1 - - [29/Apr/2019 14:25:48] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:25:51] "GET / HTTP/1.0" 404 -
172.17.0.1 - - [29/Apr/2019 14:26:16] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:26:31] "GET /api/v1/categories HTTP/1.0" 200 -
```

```
ubuntu@ip-172-31-89-6: ~
/app # python acts_app.py
 * Serving Flask app "acts_app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 255-179-364
172.17.0.1 - - [29/Apr/2019 14:25:46] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:25:51] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:26:01] "GET /api/v1/categories HTTP/1.0" 200 -
172.17.0.1 - - [29/Apr/2019 14:26:21] "GET / HTTP/1.0" 404 -
```

Img 6.

When we send 9 requests, the requests are evenly distributed among the 3 active containers in round robin manner.

REFERENCES

- https://www.tutorialspoint.com/flask
- https://www.tutorialspoint.com/sqlite/sqlite_python.htm
- https://www.w3schools.com/js/
- https://www.w3schools.com/xml/xml_http.asp
- https://www.docker.com/resources/what-container
- https://www.docker.com/resources/what-container

EVALUATIONS (Leave this for the faculty)

Date	Evaluator	Comments	Score

CHECKLIST

SNo	Item	Status
1	Source code documented	
2	Source code uploaded to CCBD server	
3	Source code in GitLab. Please do not upload your source code to github where it can be seen by everyone.	