IT 610 PROJECT

Name: Keyur Patel

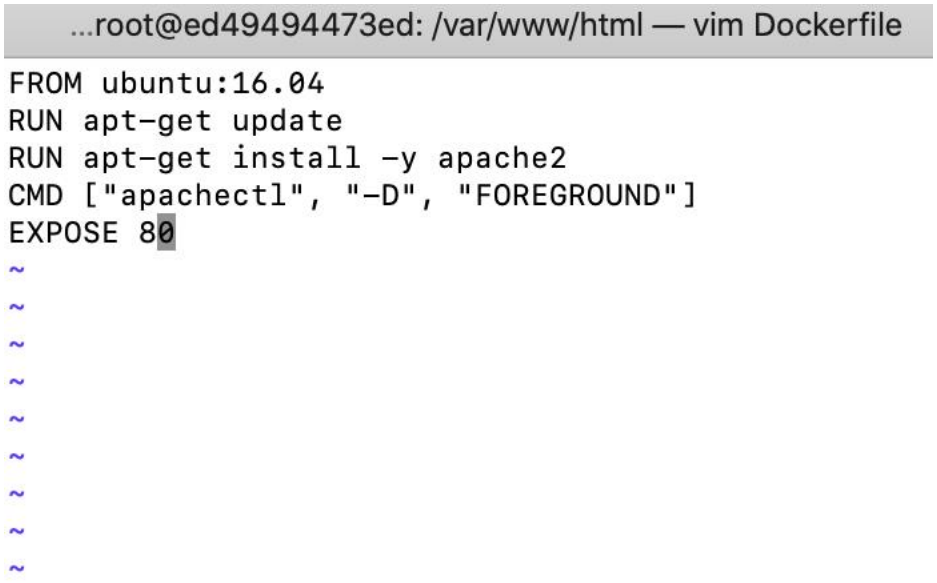
UCID: kp45

Objective: Building a functional LAMP Stack Environment with Docker. Where running two server in two container and then run a single container where link two servers and a run a web application.

Installed a docker engine. Run few docker commands i.e. **docker ps, docker images** etc. to verify docker daemon running perfectly or not. Although you can also verify by using **docker --version**.

To make my project separate from other directories and files, made a new directory using **'mkdir docker'.**

Now to run an apache server in a container, need an image for apache container, so wrote a docker file that use a base image of Ubuntu from docker registry. A docker registry is a repository for various docker images. Build a new image called a webserver2.



...root@ed49494473ed: /var/www/html — vim Dockerfile

```
FROM ubuntu:16.04
RUN apt-get update
RUN apt-get install -y apache2
CMD ["apachectl", "-D", "FOREGROUND"]
EXPOSE 80
~
~
~
~
~
~
~
~
~
```

FROM command use a base image if it is available locally otherwise it grab from the docker registry.

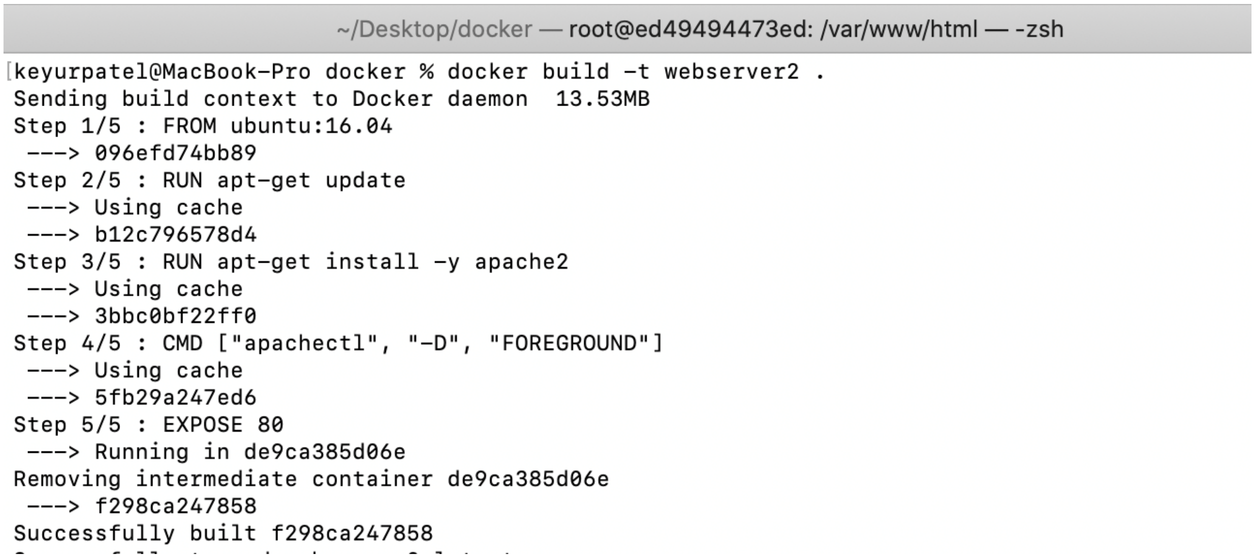
RUN command is execute in a container while building an image, so it creates a temporary container and run the command that we want run while building an image.

Here I have run two commands **'apt-get update'** and **'apt-get install –y apache2',** y is parameter is automate an installation when it ask for y/n to install an application it applies y.

CMD basically like a command prompt to execute the commands while building an image.

EXPOSE use to opens up port for the container.

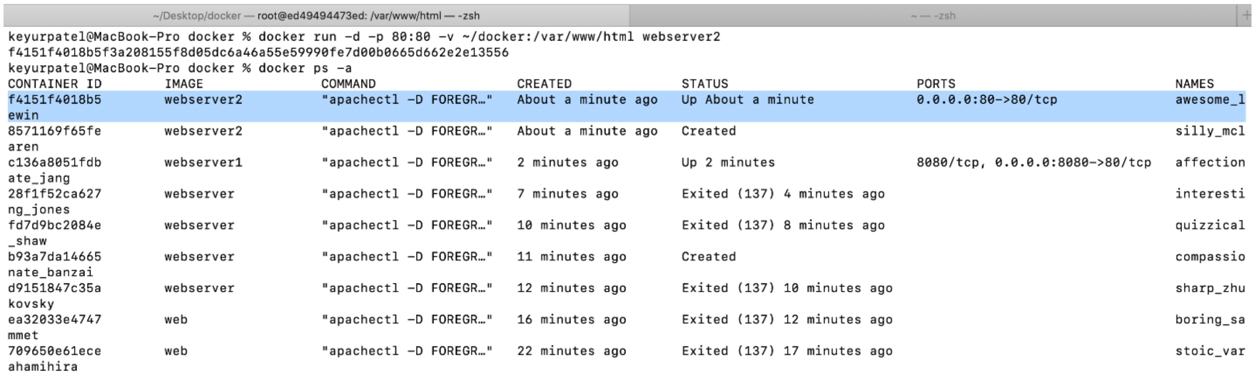
Now using this dockerfile I created a docker image using '**docker build –t webserver2' .** As seen in below image successfully built.



```
[keyurpatel@MacBook-Pro docker % docker build -t webserver2 .
 Sending build context to Docker daemon  13.53MB
 Step 1/5 : FROM ubuntu:16.04
  ---> 096efd74bb89
 Step 2/5 : RUN apt-get update
  ---> Using cache
  ---> b12c796578d4
 Step 3/5 : RUN apt-get install -y apache2
  ---> Using cache
  ---> 3bbc0bf22ff0
 Step 4/5 : CMD ["apachectl", "-D", "FOREGROUND"]
  ---> Using cache
  ---> 5fb29a247ed6
 Step 5/5 : EXPOSE 80
  ---> Running in de9ca385d06e
 Removing intermediate container de9ca385d06e
  ---> f298ca247858
 Successfully built f298ca247858
```

Now to run a container using this image, I use ‘**docker run –d –p 80:80 –v ~/docker:var/www/html webserver2 ’** command.

You can see that my container is running on port 80 using webserver2 image.



```
keyurpatel@MacBook-Pro docker % docker run -d -p 80:80 -v ~/docker:/var/www/html webserver2
f4151f4018b5f3a208155f8d05dc6a46a55e59990fe7d00b0665d662e2e13556
keyurpatel@MacBook-Pro docker % docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS                        PORTS                            NAMES
f4151f4018b5        webserver2          "apachectl -D FOREGR…"   About a minute ago   Up About a minute             0.0.0.0:80->80/tcp               awesome_l
ewin
8571169f65fe        webserver2          "apachectl -D FOREGR…"   About a minute ago   Created                                                        silly_mcl
aren
c136a8051fdb        webserver1          "apachectl -D FOREGR…"   2 minutes ago        Up 2 minutes                  8080/tcp, 0.0.0.0:8080->80/tcp   affection
ate_jang
28f1f52ca627        webserver           "apachectl -D FOREGR…"   7 minutes ago        Exited (137) 4 minutes ago                                     interesti
ng_jones
fd7d9bc2084e        webserver           "apachectl -D FOREGR…"   10 minutes ago       Exited (137) 8 minutes ago                                     quizzical
_shaw
b93a7da14665        webserver           "apachectl -D FOREGR…"   11 minutes ago       Created                                                        compassio
nate_banzai
d9151847c35a        webserver           "apachectl -D FOREGR…"   12 minutes ago       Exited (137) 10 minutes ago                                    sharp_zhu
kovsky
ea32033e4747        web                 "apachectl -D FOREGR…"   16 minutes ago       Exited (137) 12 minutes ago                                    boring_sa
mmet
709650e61ece        web                 "apachectl -D FOREGR…"   22 minutes ago       Exited (137) 17 minutes ago                                    stoic_var
ahamihira
```

-d parameter is to run container in detached/daemon mode means it runs in background.

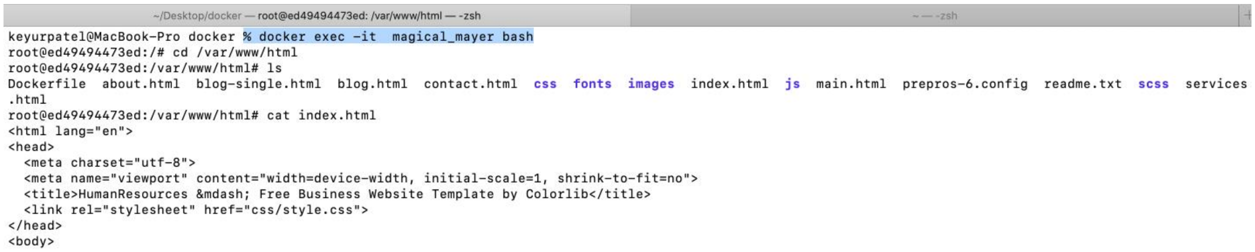
-p parameter for port forwarding means it will run my container on port 80, you can use any port, but make sure that port is free.

-v parameter is used to mount my local host directory of my local machine to apache container’s directory that is **/var/www/html**, now why I used that specific directory only because by default the html file is exits on this directory in apache container so to replace that file with my web applications files I used that directory.

Webserver2 is the image that I want to pass.

To verify if the directory is mounted in container or not I used **docker exec –it 'container_name/id' bash** command.

And you can see ths directory has been mounted on to the container.

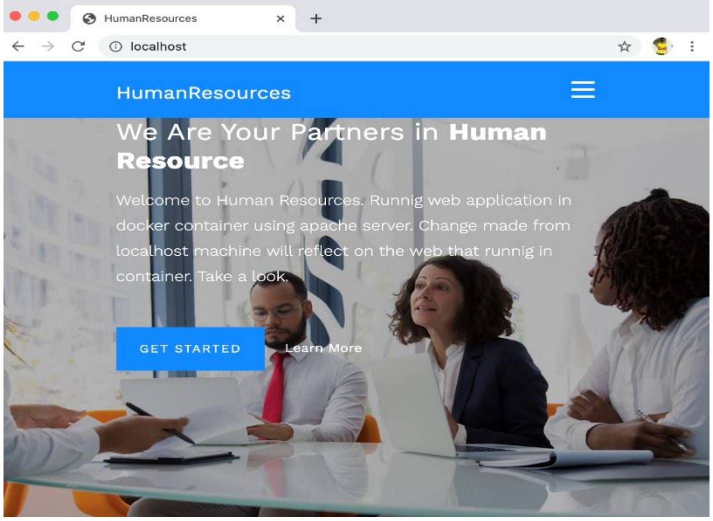


```
keyurpatel@MacBook-Pro docker % docker exec -it  magical_mayer bash
root@ed49494473ed:/# cd /var/www/html
root@ed49494473ed:/var/www/html# ls
Dockerfile  about.html  blog-single.html  blog.html  contact.html  css  fonts  images  index.html  js  main.html  prepros-6.config  readme.txt  scss  services
.html
root@ed49494473ed:/var/www/html# cat index.html
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>HumanResources &mdash; Free Business Website Template by Colorlib</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
```

Note: you can also copy your index.html while building an image but by mounting it on to the container it will gives you a flexibility to make a change in your index.html from local host machine and that will immediately effects on an application that running in container, you don't need to go to the container and make the changes every time from there. You can see that in action by clicking this animation file.

Apache Running in Container.mov

Running a web application in a container.

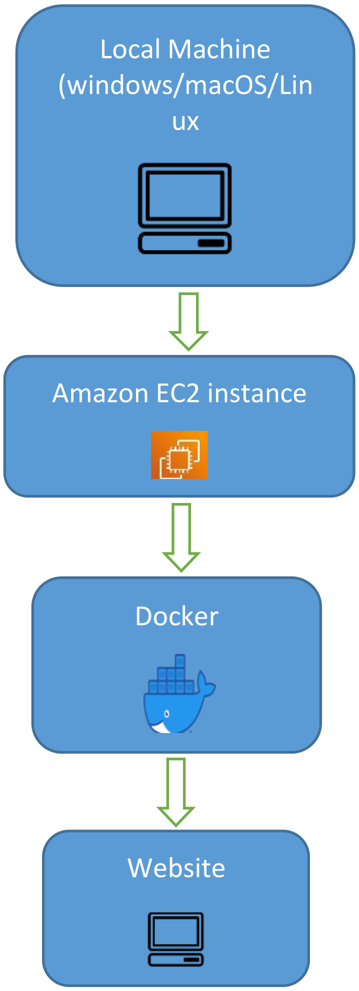


We Are Your Partners in **Human Resource**

Welcome to Human Resources. Runnig web application in docker container using apache server. Change made from localhost machine will reflect on the web that runnig in container. Take a look.

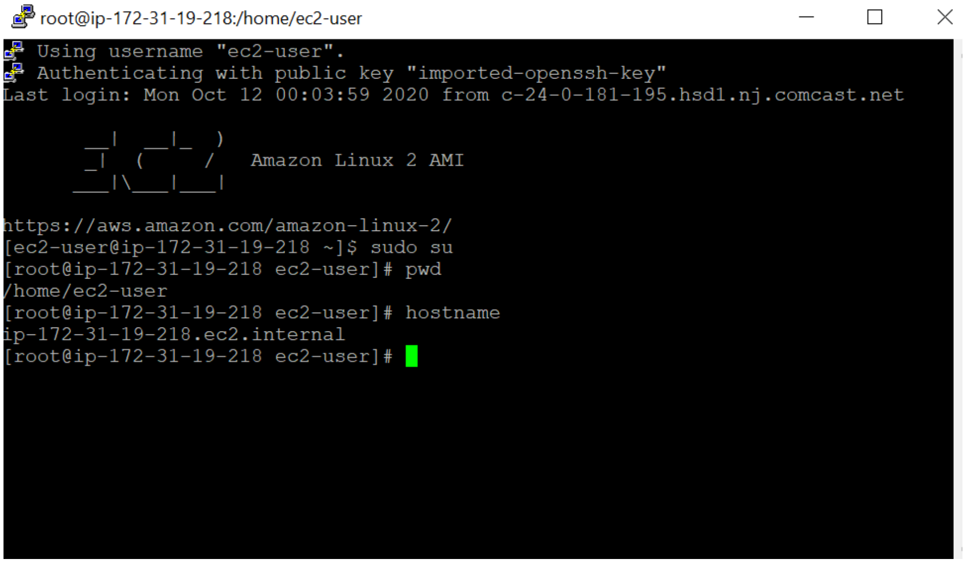
Now, I have web server running in a docker container and that hosting a website on localhost port 80 and I am running a docker from my local machine. But, I fill that I need to add something more in it so I decide to integrate with the AWS.

So, what I did I am running a AWS EC2 instance (Amazon Linux), running a docker in EC2, running a container inside a docker and in that I am running a apache server and now I have my website hosted on AWS with public DNS so, now anyone can visit that website I do not have to pay any fees or no need to buy any private DNS to deploy a website which is the purpose of AWS.



Different layers that involved in a multi-tier web application

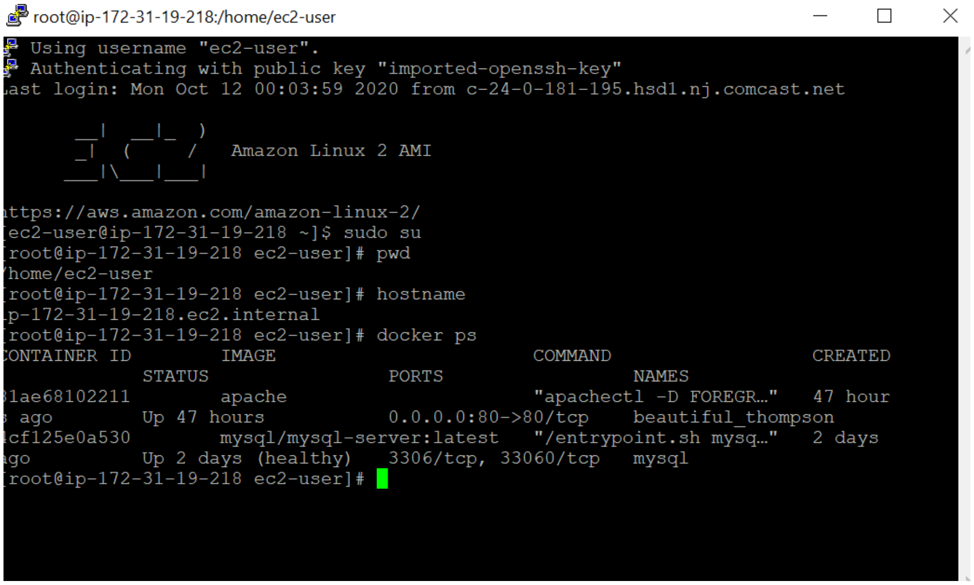
You can see here I am running a EC2 instance.



root@ip-172-31-19-218:/home/ec2-user

```
 Using username "ec2-user".
 Authenticating with public key "imported-openssh-key"
Last login: Mon Oct 12 00:03:59 2020 from c-24-0-181-195.hsd1.nj.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-19-218 ~]$ sudo su
[root@ip-172-31-19-218 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-19-218 ec2-user]# hostname
ip-172-31-19-218.ec2.internal
[root@ip-172-31-19-218 ec2-user]#
```

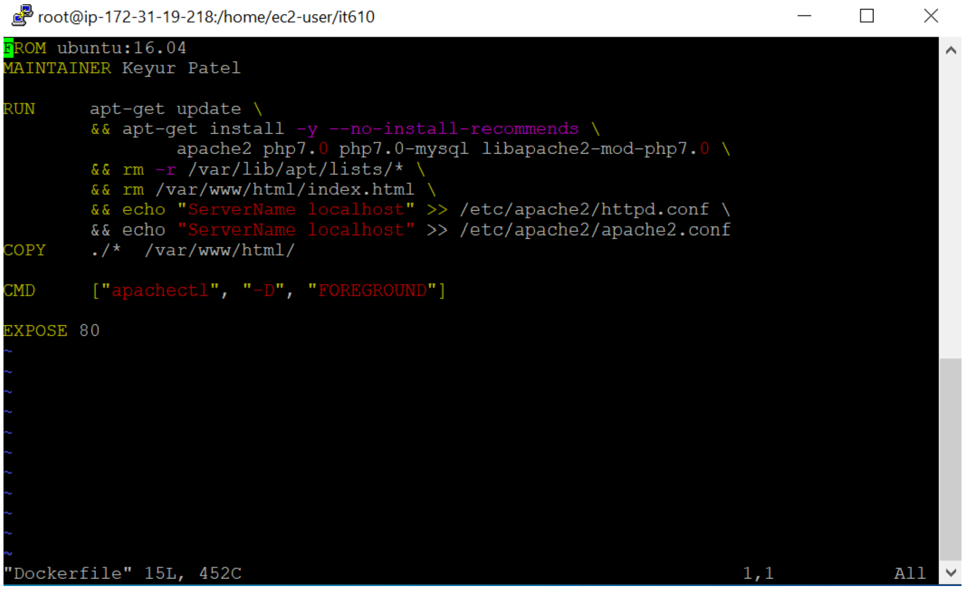
Running a docker in EC2.



root@ip-172-31-19-218:/home/ec2-user

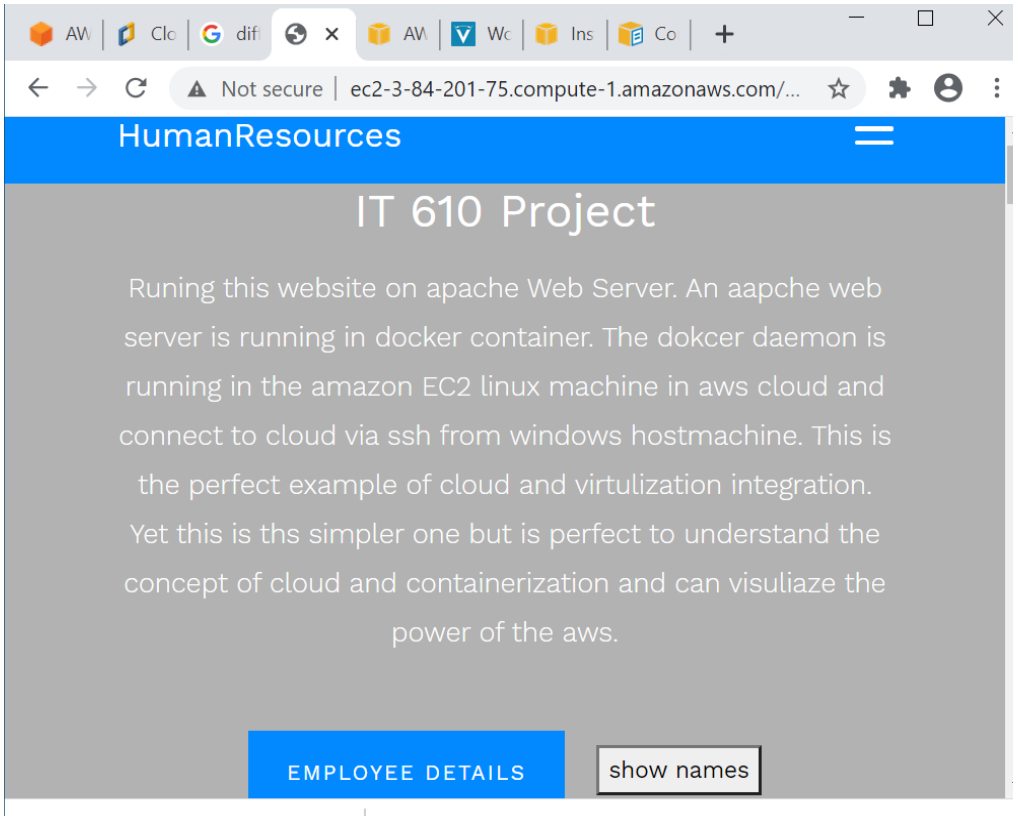
```
 Using username "ec2-user".
 Authenticating with public key "imported-openssh-key"
ast login: Mon Oct 12 00:03:59 2020 from c-24-0-181-195.hsd1.nj.comcast.net

      __|  __|_  )
      _|  (     /   Amazon Linux 2 AMI
     ___|\___|___|

ttps://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-19-218 ~]$ sudo su
[root@ip-172-31-19-218 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-19-218 ec2-user]# hostname
p-172-31-19-218.ec2.internal
[root@ip-172-31-19-218 ec2-user]# docker ps
ONTAINER ID        IMAGE                        COMMAND                  CREATED
            STATUS                  PORTS                 NAMES
1ae68102211        apache                       "apachectl -D FOREGR…"   47 hour
s ago       Up 47 hours             0.0.0.0:80->80/tcp    beautiful_thompson
cf125e0a530        mysql/mysql-server:latest    "/entrypoint.sh mysq…"   2 days
ago         Up 2 days (healthy)     3306/tcp, 33060/tcp   mysql
[root@ip-172-31-19-218 ec2-user]#
```

Docker image for apache server.



```
FROM ubuntu:16.04
MAINTAINER Keyur Patel

RUN     apt-get update \
        && apt-get install -y --no-install-recommends \
                apache2 php7.0 php7.0-mysql libapache2-mod-php7.0 \
        && rm -r /var/lib/apt/lists/* \
        && rm /var/www/html/index.html \
        && echo "ServerName localhost" >> /etc/apache2/httpd.conf \
        && echo "ServerName localhost" >> /etc/apache2/apache2.conf
COPY    ./*  /var/www/html/

CMD     ["apachectl", "-D", "FOREGROUND"]

EXPOSE 80
~
~
~
~
~
~
~
~
~
~
~
"Dockerfile" 15L, 452C                                      1,1           All
```

This is my website that deploy/hosted via AWS. You can visit on http://ec2-54-221-184-226.compute-1.amazonaws.com



HumanResources

IT 610 Project

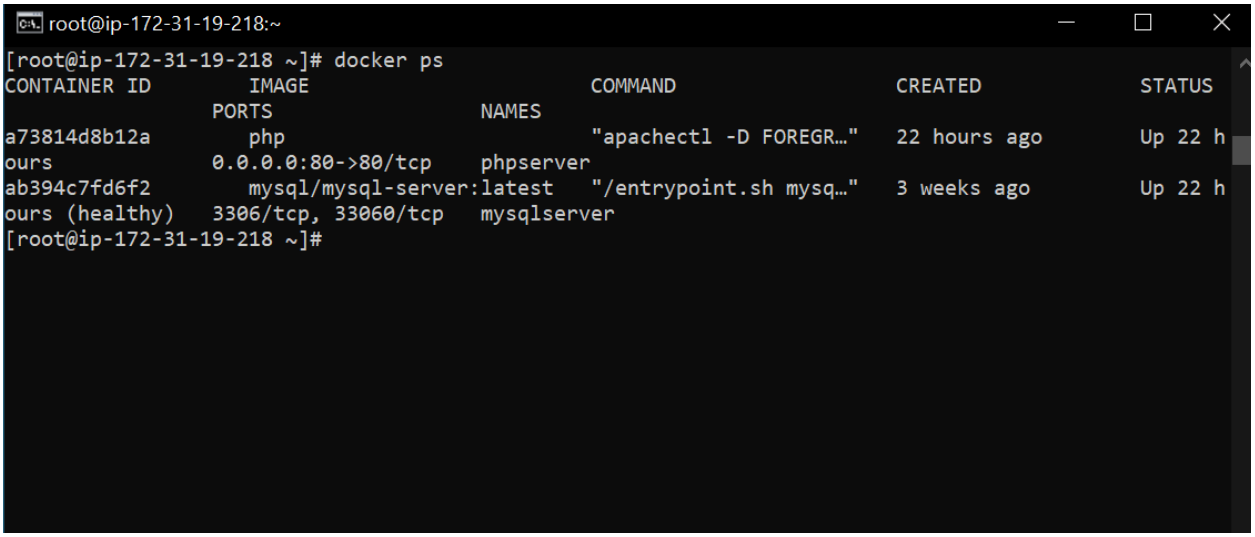
Runing this website on apache Web Server. An aapche web server is running in docker container. The dokcer daemon is running in the amazon EC2 linux machine in aws cloud and connect to cloud via ssh from windows hostmachine. This is the perfect example of cloud and virtulization integration. Yet this is ths simpler one but is perfect to understand the concept of cloud and containerization and can visuliaze the power of the aws.

So, this is a perfect example of integration of cloud and container. You can visualize the power of cloud and containerization technology. My, next step would be running a database server in a separate container and link it to web server container and fetch the data on a web site and displayed them on website. I am trying to implement this on aws.

Here is I have to keep running an EC2 instance in order to host my website on EC2 instance. If I will stop it and start it again then the public DNS given by aws is changed so in order to register the DNS name, I have to integrate Route 53 with my EC2 so I can start and stop my instance anytime and every time I can host my website with same DNS.

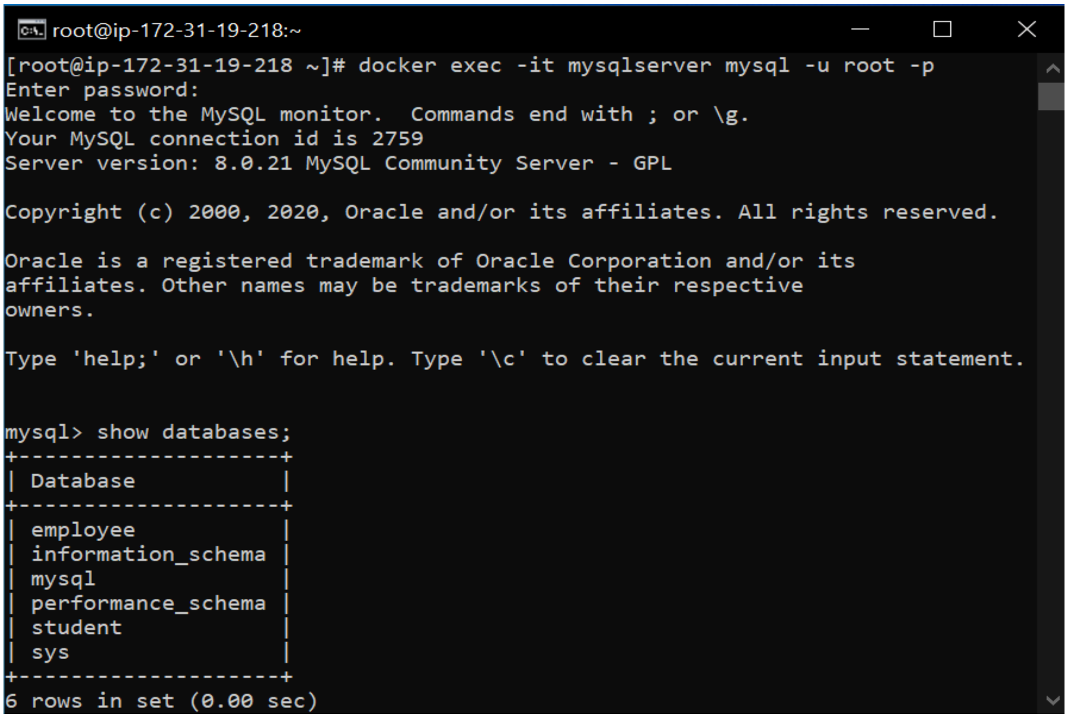
Integrating MySQL database with web server.

I am running a MySQL database in a separate container by simple pulling a mysql image from hub.docker.com

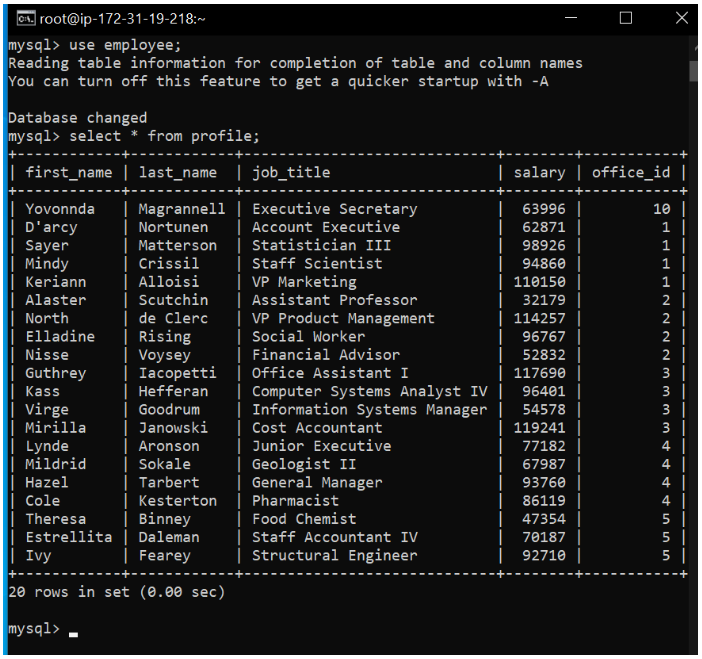


```
[root@ip-172-31-19-218 ~]# docker ps
CONTAINER ID        IMAGE                        COMMAND                  CREATED             STATUS
                 PORTS                  NAMES
a73814d8b12a        php                          "apachectl -D FOREGR…"   22 hours ago        Up 22 h
ours             0.0.0.0:80->80/tcp     phpserver
ab394c7fd6f2        mysql/mysql-server:latest    "/entrypoint.sh mysq…"   3 weeks ago         Up 22 h
ours (healthy)   3306/tcp, 33060/tcp    mysqlserver
[root@ip-172-31-19-218 ~]#
```

Now I have change the authentication in MySQL database in order to access the it and created a database named “employee”. Which has table named “profile” which will be displayed on the website.



```
[root@ip-172-31-19-218 ~]# docker exec -it mysqlserver mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2759
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> show databases;
+--------------------+
| Database           |
+--------------------+
| employee           |
| information_schema |
| mysql              |
| performance_schema |
| student            |
| sys                |
+--------------------+
6 rows in set (0.00 sec)
```

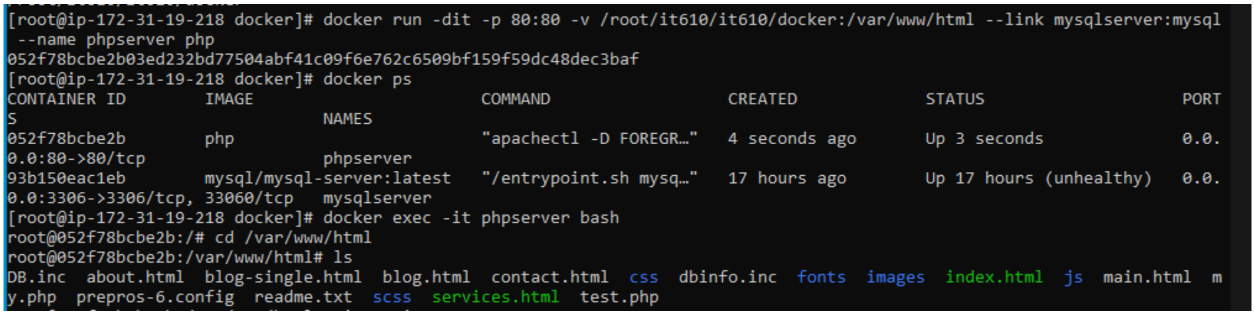


```
mysql> use employee;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from profile;
+------------+------------+------------------------------+--------+-----------+
| first_name | last_name  | job_title                    | salary | office_id |
+------------+------------+------------------------------+--------+-----------+
| Yovonnda   | Magrannell | Executive Secretary          |  63996 |        10 |
| D'arcy     | Nortunen   | Account Executive            |  62871 |         1 |
| Sayer      | Matterson  | Statistician III             |  98926 |         1 |
| Mindy      | Crissil    | Staff Scientist              |  94860 |         1 |
| Keriann    | Alloisi    | VP Marketing                 | 110150 |         1 |
| Alaster    | Scutchin   | Assistant Professor          |  32179 |         2 |
| North      | de Clerc   | VP Product Management        | 114257 |         2 |
| Elladine   | Rising     | Social Worker                |  96767 |         2 |
| Nisse      | Voysey     | Financial Advisor            |  52832 |         2 |
| Guthrey    | Iacopetti  | Office Assistant I           | 117690 |         3 |
| Kass       | Hefferan   | Computer Systems Analyst IV  |  96401 |         3 |
| Virge      | Goodrum    | Information Systems Manager  |  54578 |         3 |
| Mirilla    | Janowski   | Cost Accountant              | 119241 |         3 |
| Lynde      | Aronson    | Junior Executive             |  77182 |         4 |
| Mildrid    | Sokale     | Geologist II                 |  67987 |         4 |
| Hazel      | Tarbert    | General Manager              |  93760 |         4 |
| Cole       | Kesterton  | Pharmacist                   |  86119 |         4 |
| Theresa    | Binney     | Food Chemist                 |  47354 |         5 |
| Estrellita | Daleman    | Staff Accountant IV          |  70187 |         5 |
| Ivy        | Fearey     | Structural Engineer          |  92710 |         5 |
+------------+------------+------------------------------+--------+-----------+
20 rows in set (0.00 sec)

mysql>
```

Now, in order to make two container talk to each other, I am using docker link command.

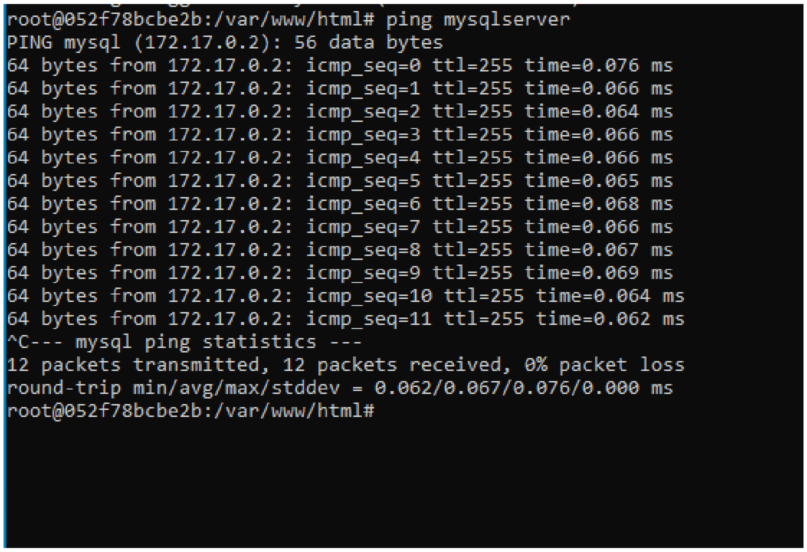
docker –link mysqlserver:mysql –name phpserver php

where mysqlserver is my container and mysql is alias from where the container is running.



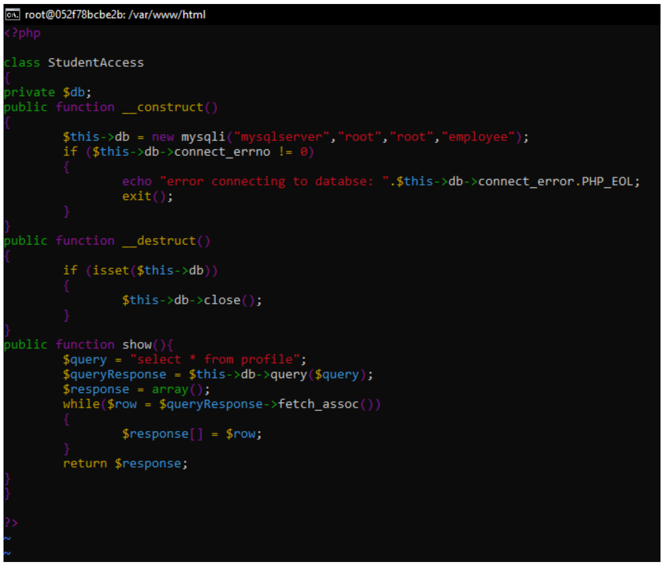
```
[root@ip-172-31-19-218 docker]# docker run -dit -p 80:80 -v /root/it610/it610/docker:/var/www/html --link mysqlserver:mysql
 --name phpserver php
052f78bcbe2b03ed232bd77504abf41c09f6e762c6509bf159f59dc48dec3baf
[root@ip-172-31-19-218 docker]# docker ps
CONTAINER ID        IMAGE                        COMMAND                  CREATED             STATUS                     PORT
S                         NAMES
052f78bcbe2b        php                          "apachectl -D FOREGR…"   4 seconds ago       Up 3 seconds               0.0.
0.0:80->80/tcp            phpserver
93b150eac1eb        mysql/mysql-server:latest    "/entrypoint.sh mysq…"   17 hours ago        Up 17 hours (unhealthy)    0.0.
0.0:3306->3306/tcp, 33060/tcp   mysqlserver
[root@ip-172-31-19-218 docker]# docker exec -it phpserver bash
root@052f78bcbe2b:/# cd /var/www/html
root@052f78bcbe2b:/var/www/html# ls
DB.inc  about.html  blog-single.html  blog.html  contact.html  css  dbinfo.inc  fonts  images  index.html  js  main.html  m
y.php  prepros-6.config  readme.txt  scss  services.html  test.php
```

Verify the connection between two containers by pinging database server from phpserver.



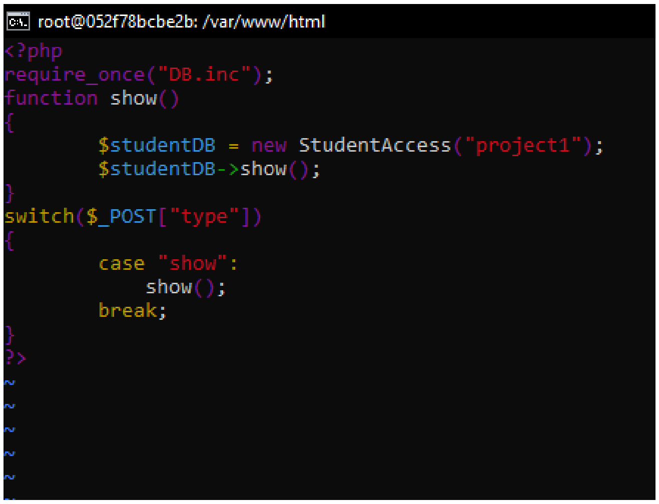
```
root@052f78bcbe2b:/var/www/html# ping mysqlserver
PING mysql (172.17.0.2): 56 data bytes
64 bytes from 172.17.0.2: icmp_seq=0 ttl=255 time=0.076 ms
64 bytes from 172.17.0.2: icmp_seq=1 ttl=255 time=0.066 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=255 time=0.064 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=255 time=0.066 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=255 time=0.066 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=255 time=0.065 ms
64 bytes from 172.17.0.2: icmp_seq=6 ttl=255 time=0.068 ms
64 bytes from 172.17.0.2: icmp_seq=7 ttl=255 time=0.066 ms
64 bytes from 172.17.0.2: icmp_seq=8 ttl=255 time=0.067 ms
64 bytes from 172.17.0.2: icmp_seq=9 ttl=255 time=0.069 ms
64 bytes from 172.17.0.2: icmp_seq=10 ttl=255 time=0.064 ms
64 bytes from 172.17.0.2: icmp_seq=11 ttl=255 time=0.062 ms
^C--- mysql ping statistics ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.062/0.067/0.076/0.000 ms
root@052f78bcbe2b:/var/www/html#
```

Here, I am integrating the data using DB.inc and my.php and displaying it on website using javascript function which will be called by sending a POST request to database server.



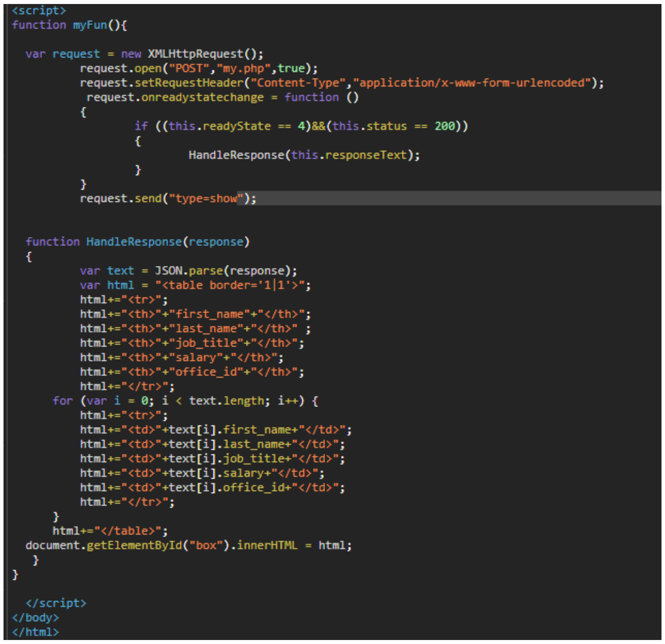
```
root@052f78bcbe2b: /var/www/html
<?php

class StudentAccess
{
private $db;
public function __construct()
{
        $this->db = new mysqli("mysqlserver","root","root","employee");
        if ($this->db->connect_errno != 0)
        {
                echo "error connecting to databse: ".$this->db->connect_error.PHP_EOL;
                exit();
        }
}
public function __destruct()
{
        if (isset($this->db))
        {
                $this->db->close();
        }
}
public function show(){
        $query = "select * from profile";
        $queryResponse = $this->db->query($query);
        $response = array();
        while($row = $queryResponse->fetch_assoc())
        {
                $response[] = $row;
        }
        return $response;
}
}

?>
~
~
```

DB.inc



```
root@052f78bcbe2b: /var/www/html
<?php
require_once("DB.inc");
function show()
{
        $studentDB = new StudentAccess("project1");
        $studentDB->show();
}
switch($_POST["type"])
{
        case "show":
            show();
        break;
}
?>
~
~
~
~
~
```

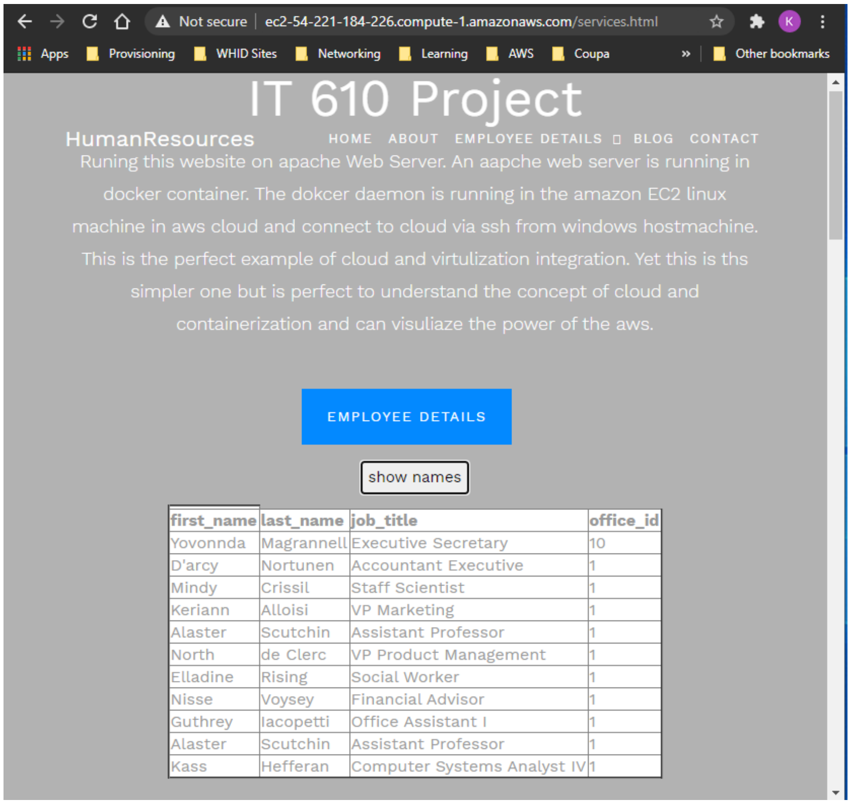
my.php



```
<script>
function myFun(){

  var request = new XMLHttpRequest();
          request.open("POST","my.php",true);
          request.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
           request.onreadystatechange = function ()
          {
                  if ((this.readyState == 4)&&(this.status == 200))
                  {
                          HandleResponse(this.responseText);
                  }
          }
          request.send("type=show");


  function HandleResponse(response)
  {
          var text = JSON.parse(response);
          var html = "<table border='1|1'>";
          html+="<tr>";
          html+="<th>"+"first_name"+"</th>";
          html+="<th>"+"last_name"+"</th>" ;
          html+="<th>"+"job_title"+"</th>";
          html+="<th>"+"salary"+"</th>";
          html+="<th>"+"office_id"+"</th>";
          html+="</tr>";
      for (var i = 0; i < text.length; i++) {
          html+="<tr>";
          html+="<td>"+text[i].first_name+"</td>";
          html+="<td>"+text[i].last_name+"</td>";
          html+="<td>"+text[i].job_title+"</td>";
          html+="<td>"+text[i].salary+"</td>";
          html+="<td>"+text[i].office_id+"</td>";
          html+="</tr>";
      }
      html+="</table>";
  document.getElementById("box").innerHTML = html;
   }
}

  </script>
</body>
</html>
```

Javascript

Here is the result:



IT 610 Project

HumanResources HOME ABOUT EMPLOYEE DETAILS ▯ BLOG CONTACT

Runing this website on apache Web Server. An aapche web server is running in docker container. The dokcer daemon is running in the amazon EC2 linux machine in aws cloud and connect to cloud via ssh from windows hostmachine. This is the perfect example of cloud and virtulization integration. Yet this is ths simpler one but is perfect to understand the concept of cloud and containerization and can visuliaze the power of the aws.

EMPLOYEE DETAILS

show names

first_name	last_name	job_title	office_id
Yovonnda	Magrannell	Executive Secretary	10
D'arcy	Nortunen	Accountant Executive	1
Mindy	Crissil	Staff Scientist	1
Keriann	Alloisi	VP Marketing	1
Alaster	Scutchin	Assistant Professor	1
North	de Clerc	VP Product Management	1
Elladine	Rising	Social Worker	1
Nisse	Voysey	Financial Advisor	1
Guthrey	Iacopetti	Office Assistant I	1
Alaster	Scutchin	Assistant Professor	1
Kass	Hefferan	Computer Systems Analyst IV	1

If you are trying to access the website using public DNS mention in this report then you will not be able to see any results because once I stop the EC2 instance and start it again it will change the public DNS and in order to use the private IP to host the website I have to registered my DNS record using Route 53.So here I have used a free tier account to execute results, therefore you will not able to see my web application in real time.