

ICS 432 - 01 – Distributed and Cloud Computing Fall 2021

- Assignment #1 : Working with Cloud Virtual Machines
- Assignment #2 : Load Balancing and Auto Scaling
- Assignment #3 : MapReduce
- Docker Lab : Docker in AWS and GCP

Assignment #1 : Working with Cloud Virtual Machines

Exercise 1: Hosting a web site on an AWS EC2 instance.

Homework report screen-shot #1:

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with various services like EC2 Dashboard, Events, Tags, Limits, Instances (selected), Images, Elastic Block Store, Network & Security, and more. The main content area displays a table of instances. One instance is selected: "Danddank We...", with the instance ID "i-05a38511178bf90a0". The instance is shown as "Running" in the "Instance state" column. Below the table, a detailed view of the selected instance is shown in a modal window. The modal has tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The Details tab is active, showing the instance summary. It includes fields for Instance ID (i-05a38511178bf90a0), Public IPv4 address (18.212.243.94), Private IPv4 addresses (172.31.16.117), Public IPv4 DNS (ec2-18-212-243-94.compute-1.amazonaws.com), and other details like VPC ID (vpc-64fa8419) and Subnet ID (subnet-9bf20fd7). A note in the modal states: "AWS Compute Optimizer finding: User: arn:aws:sts::196580907486:assumed-role/vocstartsoftuser1564948#Nalongsone_Danddank is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resources * with an explicit deny. Retry".

Homework report screen-shot #2:

```

ec2-user@ip-172-31-16-117:~ Last login: Sat Sep 18 18:26:21 on ttys001
(base) ping58972@Nalongsones-MacBook-Air Assignment-01 % chmod 400 ics432homework1keypair.pem
(base) ping58972@Nalongsones-MacBook-Air Assignment-01 % ssh -i "ics432homework1keypair.pem" ec2-user@ec2-18-212-243-94.compute-1.amazonaws.com
The authenticity of host 'ec2-18-212-243-94.compute-1.amazonaws.com (18.212.243.94)' can't be established.
ECDSA key fingerprint is SHA256:EeRPzAW0v8+5XD4Ysgz9Yjqlid2qDExzAzGs9RCxiqA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-212-243-94.compute-1.amazonaws.com,18.212.243.94' (ECDSA) to the list of known hosts.

--| --|_
_| (   /   Amazon Linux 2 AMI
---| \___|_____|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-16-117 ~]$ █

```

Homework report screen-shot #3:

```

ec2-user@ip-172-31-16-117:~ Installing : gperftools-libs-2.6.1-1.amzn2.x86_64          3/5
Installing : 1:nginx-filesystem-1.20.0-2.amzn2.0.noarch      4/5
Installing : 1:nginx-1.20.0-2.amzn2.0.4.x86_64           5/5
Verifying   : openssl11-pkcs11-0.4.10-6.amzn2.0.1.x86_64    1/5
Verifying   : 1:nginx-1.20.0-2.amzn2.0.4.x86_64           2/5
Verifying   : 1:nginx-filesystem-1.20.0-2.amzn2.0.4.noarch  3/5
Verifying   : 1:openssl11-libs-1.1.1g-12.amzn2.0.3.x86_64    4/5
Verifying   : gperftools-libs-2.6.1-1.amzn2.x86_64          5/5

Installed:
  nginx.x86_64 1:1.20.0-2.amzn2.0.4

Dependency Installed:
  gperftools-libs.x86_64 0:2.6.1-1.amzn2.0.3           nginx-filesystem.noarch 1:1.20.0-2.amzn2.0.4
  openssl11-libs.x86_64 1:1.1.1g-12.amzn2.0.3          openssl11-pkcs11.x86_64 0:0.4.10-6.amzn2.0.1

Complete!
  0 ansible2           available  \
    [ =2.4.2  =2.4.6  =2.8  =stable ]
  2 httpd_modules       available  [ =1.0  =stable ]
  3 memcached1.5        available  \
    [ =1.5.1  =1.5.16  =1.5.17 ]
  5 postgresql9.6        available  \
    [ =9.6.6  =9.6.8  =stable ]
  6 postgresql10         available  [ =10  =stable ]
  9 R3.4                 available  [ =3.4.3  =stable ]
  10 rust1               available  \
    [ =1.22.1  =1.26.0  =1.26.1  =1.27.2  =1.31.0  =1.38.0
      =stable ]
  11 vim                 available  [ =8.0  =stable ]
  15 php7.2              available  \
    [ =7.2.0  =7.2.4  =7.2.5  =7.2.8  =7.2.11  =7.2.13  =7.2.14
      =7.2.16  =7.2.17  =7.2.19  =7.2.21  =7.2.22  =7.2.23
      =7.2.24  =7.2.26  =stable ]
  17 lamp-mariadb10.2-php7.2 available  \
    [ =10.2.10_7.2.0  =10.2.10_7.2.4  =10.2.10_7.2.5
      =10.2.10_7.2.8  =10.2.10_7.2.11  =10.2.10_7.2.13
      =10.2.10_7.2.14 =10.2.10_7.2.16  =10.2.10_7.2.17
      =10.2.10_7.2.19 =10.2.10_7.2.22  =10.2.10_7.2.23
      =10.2.10_7.2.24 =stable ]
  18 libreoffice          available  \
    [ =5.0.6.2_15  =5.3.6.1  =stable ]
  19 gimp                available  [ =2.8.22 ]
  20 docker=latest        enabled   \
    [ =17.12.1  =18.03.1  =18.06.1  =18.09.9  =stable ]
  21 mate-desktop1.x       available  \
    [ =1.19.0  =1.20.0  =stable ]
  22 GraphicsMagick1.3     available  \
    [ =1.3.29  =1.3.32  =1.3.34  =stable ]
  23 tomcat8.5            available  \
    [ =8.5.31  =8.5.32  =8.5.38  =8.5.40  =8.5.42  =8.5.50
      =stable ]

```

Homework report screen-shot #4:

```
ec2-user@ip-172-31-16-117:~/data/www/BallotOnline
[ec2-user@ip-172-31-16-117 www]$ cd BallotOnline
[ec2-user@ip-172-31-16-117 BallotOnline]$ ls
about-us.html assets contact-us.html index.html nginx.conf services.html
[ec2-user@ip-172-31-16-117 BallotOnline]$
```

Homework report screen-shot #5:

```
ec2-user@ip-172-31-16-117:/data/www/BallotOnline
about-us.html assets contact-us.html index.html nginx.conf services.html
[ec2-user@ip-172-31-16-117 BallotOnline]$ cat index.html
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8" />
    <title>Home - BallotOnline</title>
  </head>
  <body>
    <div style="text-align: left; text-indent: 0px; padding: 0px 0px 0px; margin: 0px 0px 0px;"><table width="100%" border="0" cellpadding="0" cellspacing="2" style="border-width: 0px; background-color: #ffffff;">
      <tr valign="top">
        <td colspan="2" width="523">
          <center>
            <a href="index.html">
              
            </a><br />
          </center>
        </td>
      </tr>
      <tr valign="top">
        <td width="164">
          <ul>
            <li>
              <a href="index.html">Home</a>
            </li>
            <li>
              <a href="about-us.html">About Us</a>
            </li>
            <li>
              <a href="services.html">Services</a>
            </li>
            <li>
              <a href="contact-us.html">Contact Us</a>
            </li>
          </ul><br />
        </td>
        <td width="359">
          <p>BallotOnline, a growing company providing online voting solutions to a global client base in North America, Europe, and Asia.</p>
          <br />
        </td>
      </tr>
    </table>
  </body>

```

Homework report screen-shot #6:

```
ec2-user@ip-172-31-16-117:/data/www/BallotOnline
<li>
  <a href="contact-us.html">Contact Us</a>
</li>
</ul><br />
</td>
<td width="359">
  <p>BallotOnline, a growing company providing online voting solutions to a global client base in North America, Europe, and Asia.</p>
  <br />
</td>
</tr>
<tr valign="top">
  <td colspan="2" width="523">
    <p>UMUC Cloud Computing Architecture Master's Degree program</p>
    
    <p>For educational purposes only. © Learning Design & Solutions, UMUC. All Rights Reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from UMUC.</p>
  </td>
</tr>
</table>
</div>
</body> <!-- /body -->
[ec2-user@ip-172-31-16-117 BallotOnline]$ sudo chmod 755 index.html
[ec2-user@ip-172-31-16-117 BallotOnline]$ sudo mv /etc/nginx/nginx.conf /etc/nginx.conf.old
[ec2-user@ip-172-31-16-117 BallotOnline]$ cat /ect/nginx/nginx.conf
cat: /ect/nginx/nginx.conf: No such file or directory
[ec2-user@ip-172-31-16-117 BallotOnline]$
```

Homework report screen-shot #7:

```
ec2-user@ip-172-31-16-117:/data/www/BallotOnline
```

```
#  
# server {  
#     listen      443 ssl http2 default_server;  
#     listen      [::]:443 ssl http2 default_server;  
#     server_name _;  
#     root        /usr/share/nginx/html;  
#  
#     ssl_certificate "/etc/pki/nginx/server.crt";  
#     ssl_certificate_key "/etc/pki/nginx/private/server.key";  
#     # It is *strongly* recommended to generate unique DH parameters  
#     # Generate them with: openssl dhparam -out /etc/pki/nginx/dhparams.pem 2048  
#     #ssl_dhparam "/etc/pki/nginx/dhparams.pem";  
#     ssl_session_cache shared:SSL:1m;  
#     ssl_session_timeout 10m;  
#     ssl_protocols TLSv1 TLSv1.2;  
#     ssl_ciphers HIGH:SEED:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!aDH:!aECDH:!EDH-DSS-DES-CBC3-S  
HA:!KRB5-DES-CBC3-SHA:!SRP;  
#     ssl_prefer_server_ciphers on;  
#  
#     # Load configuration files for the default server block.  
#     include /etc/nginx/default.d/*.conf;  
#  
#     location / {  
#  
#         error_page 404 /404.html;  
#         location = /40x.html {  
#             }  
#  
#         error_page 500 502 503 504 /50x.html;  
#         location = /50x.html {  
#             }  
#     }  
# }
```

[ec2-user@ip-172-31-16-117 BallotOnline]\$ █

Homework report screen-shot #8:

```
ec2-user@ip-172-31-16-117:/data/www/BallotOnline
```

```
[ec2-user@ip-172-31-16-117 BallotOnline]$ sudo nginx  
[ec2-user@ip-172-31-16-117 BallotOnline]$ curl http://localhost  
<!DOCTYPE html>  
  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <title>Home - BallotOnline</title>  
  </head>  
  
  <body>  
    <div style="text-align: left; text-indent: 0px; padding: 0px 0px 0px 0px; margin: 0px 0px 0px 0px;"><table width="100%" border="0" cellpadding="2" cellspacing="2" style="border-width: 0px; background-color: #ffffff;">  
      <tr valign="top">  
        <td colspan=2 width="523">  
          <center>  
            <a href="index.html">  
                
            </a><br />  
          </center>  
        </td>  
      </tr>  
      <tr valign="top">  
        <td width="164">  
          <ul>  
            <li>  
              <a href="index.html">Home</a>  
            </li>  
            <li>  
              <a href="about-us.html">About Us</a>  
            </li>  
            <li>  
              <a href="services.html">Services</a>  
            </li>
```

Homework report screen-shot #9:

The screenshot shows the AWS EC2 Management Console with the URL <https://console.aws.amazon.com/ec2/>. The page is titled "Edit inbound rules" for a security group named "sg-0944d536063035dc5 - launch-wizard-1". The interface displays two inbound rules:

Inbound rules	Type	Protocol	Port range	Source	Description - optional
sgr-0cc0730b99c66f6bd	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0dd2ae3026c9f79b	SSH	TCP	22	Custom	0.0.0.0/0

Buttons at the bottom include "Add rule", "Cancel", "Preview changes", and "Save rules".

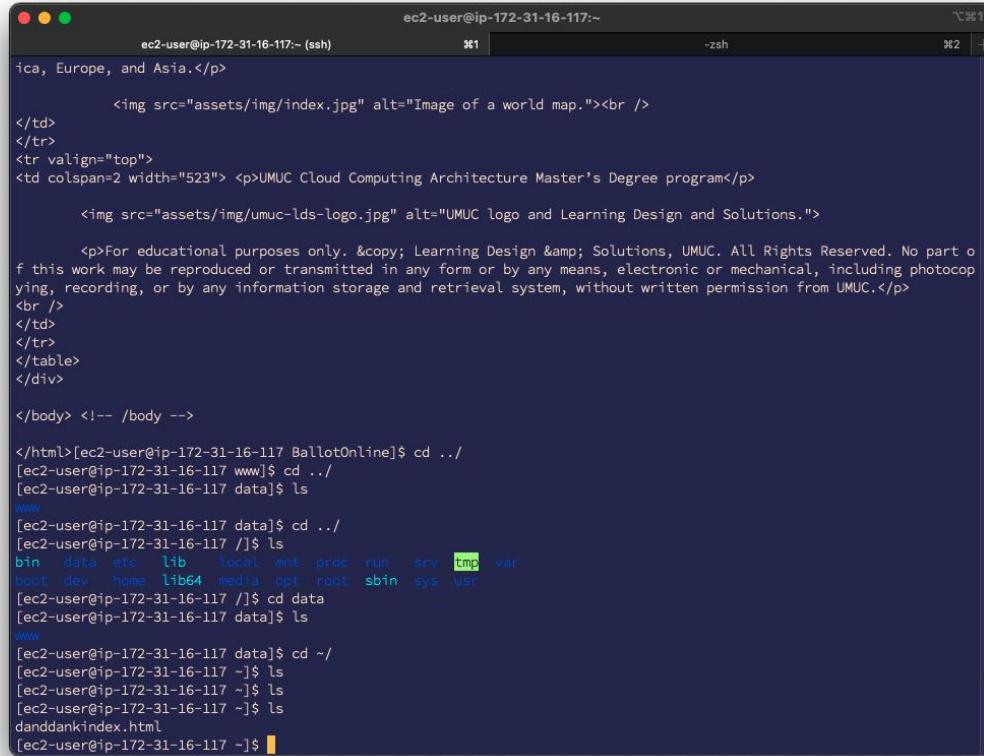
Homework report screen-shot #10: <http://18.212.243.94/index.html>

The screenshot shows a web browser window with the URL <http://18.212.243.94/index.html>. The page title is "Not Secure – 18.212.243.94". The header includes the "Workbench" tab and the "EC2 Management Console". The main content features a logo with a ballot box icon and the text "BallotOnline". Below the logo is a world map. A navigation menu on the left lists "Home", "About Us", "Services", and "Contact Us". At the bottom, there is footer information about UMUC and a copyright notice.

UMUC Cloud Computing Architecture Master's Degree program
UMUC | LEARNING DESIGN & SOLUTIONS
University of Maryland University College

For educational purposes only. © Learning Design & Solutions, UMUC. All Rights Reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from UMUC.

Homework report screen-shot #11:



```
ec2-user@ip-172-31-16-117:~
```

```
ica, Europe, and Asia.</p>
```

```
</td>
```

```
</tr>
```

```
<tr valign="top">
```

```
<td colspan=2 width="523"> <p>UMUC Cloud Computing Architecture Master's Degree program</p>
```

```
    
```

```
    <p>For educational purposes only. © Learning Design & Solutions, UMUC. All Rights Reserved. No part o
```

```
f this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocop
```

```
ying, recording, or by any information storage and retrieval system, without written permission from UMUC.</p>
```

```
<br />
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</body> <!-- /body -->
```

```
</html>[ec2-user@ip-172-31-16-117 BallotOnline]$ cd ..
```

```
[ec2-user@ip-172-31-16-117 www]$ cd ..
```

```
[ec2-user@ip-172-31-16-117 data]$ ls
```

```
www
```

```
[ec2-user@ip-172-31-16-117 data]$ cd ..
```

```
[ec2-user@ip-172-31-16-117 /]$ ls
```

```
bin data etc lib local mnt proc run srv tmp var
```

```
host dev home lib64 media opt root sbin sys usr
```

```
[ec2-user@ip-172-31-16-117 /]$ cd data
```

```
[ec2-user@ip-172-31-16-117 data]$ ls
```

```
www
```

```
[ec2-user@ip-172-31-16-117 data]$ cd ~
```

```
[ec2-user@ip-172-31-16-117 ~]$ ls
```

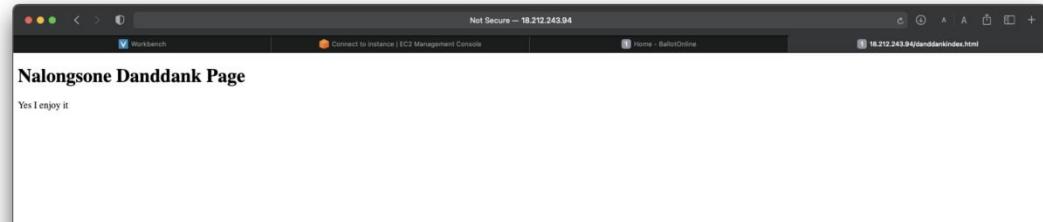
```
[ec2-user@ip-172-31-16-117 ~]$ ls
```

```
[ec2-user@ip-172-31-16-117 ~]$ ls
```

```
danddankindex.html
```

```
[ec2-user@ip-172-31-16-117 ~]$
```

Homework report screen-shot #12: <http://18.212.243.94/danddankindex.html>



Exercise 2: Data Analysis using SQL on a GCP Virtual Machine.

Homework report screen-shot #13:

The screenshot shows the Google Cloud Platform dashboard for the project 'dandankics432homework1'. The left sidebar includes sections for Marketplace, Billing, APIs & Services, Support, IAM & Admin, Getting started, Compliance, Security, Anthos, Compute (Compute Engine, Kubernetes Engine, VMware Engine), Serverless (Cloud Run, Cloud Functions, App Engine), and Release Notes. The main area displays 'Project info' (Project name: dandankics432homework1, Project ID: dandankics432homework1, Project number: 354416418761), 'APIs' (Requests (requests/sec) chart from 2 AM to 2:45 PM), 'Google Cloud Platform status' (All services normal), 'Monitoring' (Create my dashboard, Set up alerting policies, Create uptime checks, View all dashboards, Go to Monitoring), and 'Error Reporting' (No sign of any errors). A notification at the top encourages users to join Google Cloud Next on October 12-14.

Homework report screen-shot #14:

The screenshot shows the 'Overview' page for the 'Billing' section. The left sidebar lists options like Reports, Cost table, Cost breakdown, Commitments, Commitment analysis, Budgets & alerts, Billing export, Pricing, Account management, and Release Notes. The main content area shows 'Current month' (September 1 – 19, 2021) with 'Month-to-date total cost' at \$0.00 and 'End-of-month total cost (forecasted)' at \$0.00. It also features a 'Cost trend' chart for September 1, 2020 – September 30, 2021, showing average monthly total cost at \$0.00. On the right, there are sections for 'Billing account' (Manage, Billing Account for Education, D101B0-2A5BC3-BE6EE1), 'Organization' (No organization), 'Billing health checks' (Credits: \$50.00, Remaining credits: Out of \$50.00), and 'Remaining credits' (Distributed and Cloud Computing: \$50.00, pt2). A search bar at the top is used to find specific billing pages.

Homework report screen-shot #15:

The screenshot shows the Google Cloud Platform VM Instances page. On the left, there's a sidebar with sections like Compute Engine, Storage, Instance groups, and VM Manager. The main area displays a table of VM instances. One instance, 'danddarkhomework1mysql', is listed with the following details:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
up	danddarkhomework1mysql	us-central-1a			10.128.0.2	34.133.21.116 (nicD)	SSH

Below the table, there are related actions: View billing report, Monitor VMs, Explore VM logs, Set up firewall rules, and Patch management.

A modal window titled 'Select an instance' is open on the right, displaying the message 'Please select at least one resource.'

Homework report screen-shot #16:

The screenshot shows an SSH session connected to a Google Cloud VM instance. The terminal window displays the following text:

```
Connected to host [REDACTED] 0. A233T00BZ17017818C1DC1C1CF651729C00
Last login: Mon Jul 16 14:45:18 UTC 2018 from [REDACTED]
Linux danddarkhomework1mysql 4.19.0-17-cloud-amd64 #1 SMP Debian 4.19.19-3 (202
1-07-18) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual copyright headers in /usr/share/doc/*copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[REDACTED]@danddarkhomework1mysql:~$
```

Homework report screen-shot #17:

Homework report screen-shot #18:

```
● Qiklabs - Hands-On Cloud Training                               ssh.cloud.google.com
● VM instances - Compute Engine - dandankhomework... - Google Cloud Platform
● nalongzone_dandankhomework@mysql:~$ /tmp

nalongzone_dandankhomework@mysql:~$ /tmp$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2021-09-19 08:17:44 UTC; 2hs ago
       Docs: man:mysqld(8)
     Main PID: 2978 (mysqld)
        Status: "Server is operational"
           CPU: 0.000 CPU(s) total (453)
          Memory: 360.9M
        CGroup: /system.slice/mysql.service
                 └─mysqld --basedir=/var/lib/mysql --log-error=/var/log/mysqld.log

Sep 19 08:17:43 dandankhomework.mysql systemd[1]: Starting MySQL Community Server...
Sep 19 08:17:44 dandankhomework.mysql systemd[1]: Started MySQL Community Server.
nalongzone_dandankhomework@mysql:~$ /tmp$ mysqladmin -u root -p version
mysqladmin Ver 8.0.26 for Linux on x86_64 MySQL Community Server - GTG
Copyright (c) 2000, 2021, 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.

Protocol version         8
Server version          8.0.26
Protocol version        10
Connection               Localhost via UNIX socket
UNIX socket              /var/run/mysqld/mysqld.sock
Uptime:                  1 min 50 sec

Threads: 2 Questions: 2 Slow queries: 0 Opens: 117 Flush tables: 3 Open tables: 36 Queries per second avg: 0.018
nalongzone_dandankhomework@mysql:~$ /tmp$ mysql -u root -p
Welcome to the MySQL monitor. Commands end with ; or \q.
Your MySQL connection id is 9
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, 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> /
```

Homework report screen-shot #19:

```
● Qiklabs - Hands-On Cloud Training
Status: "Passed" is operational*
State: 38 (INIT)
Memory: 360.9M
CGroup: /system.slice/mysql.service
└─:3306 /usr/sbin/mysqld

Sep 19 08:17:44 valongone_danddankdanddankhomeworkmysqlg systemd[1]: Starting MySQL Community Server...
Sep 19 08:17:44 valongone_danddankdanddankhomeworkmysqlg systemd[1]: Started MySQL Community Server.
valongone_danddankdanddankhomeworkmysqlg:/tmp$ mysql -u root -p version
mysqladmin Ver 8.0.26 for Linux on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2021, 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.

server version      8.0.26
Protocol version   10
Connection          Localhost via UNIX socket
UNIX socket        /var/run/mysqld/mysqld.sock
Uptime              1 min 50 sec

Threads: 2 Questions: 0 Opens: 117 Flush tables: 3 Open tables: 36 Queries per second avg: 0.018

valongone_danddankdanddankhomeworkmysqlg:/tmp$ mysql -u root -p
Welcome to the MySQL monitor. Commands end with ; or \q.
Your MySQL connection id is 9
server version: 8.0.26 MySQL Community Server - GPL
Copyright (c) 2000, 2021, 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 database movie_ratings_database;
Query OK, 1 row affected (0.01 sec)

mysql> exit
Bye
valongone_danddankdanddankhomeworkmysqlg:/tmp$ cd ~/
valongone_danddankdanddankhomeworkmysqlg:~$ ls
item.csv  ratings.csv  users.csv
valongone_danddankdanddankhomeworkmysqlg:~$
```

Homework report screen-shot #20: *943 rows in the users table*

```
OnMaster : HandsonCloudTraining
natingame_dandtark@handsoncloudtraining:~$ sudo apt update
Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://deb.debian.org/debian buster InRelease
Hit:3 http://deb.debian.org/debian buster-updates InRelease
Hit:4 http://deb.debian.org/debian buster-backports InRelease
Get:5 https://packages.cloud.google.com/apt google-cloud-packages-archive-keyring-buster InRelease [9553 B]
Hit:6 https://packages.cloud.google.com/apt google-compute-engine-buster InRelease
Get:7 https://packages.cloud.google.com/apt google-compute-engine-buster Sources [944 B]
Fetched 944 B in 0s (1.0 kB/s)
901 | 28 M | executive | LIVW
902 | 28 M | student | LIVW
903 | 28 M | educator | LIVW
904 | 27 M | student | LIVW
905 | 27 M | librarian | LIVW
906 | 45 M | librarian | 70124
907 | 26 F | other | 8596
908 | 44 F | librarian | 68004
909 | 50 F | educator | 33171
910 | 50 F | librarian | 12775
911 | 37 F | writer | 53210
912 | 51 M | other | 6513
913 | 51 M | student | 143001
914 | 44 F | other | 8105
915 | 50 M | entertainment | 69414
916 | 50 M | other | 20008
917 | 22 F | student | 20006
918 | 25 M | scientist | 1165
919 | 25 M | other | 14215
920 | 30 F | actress | 90008
921 | 29 F | student | 89101
922 | 29 F | administrator | 21114
923 | 21 M | student | 8238R
924 | 21 M | other | 1123
925 | 18 F | salesman | 49036
926 | 49 M | entertainment | 1701
927 | 23 M | programmer | 10088
928 | 21 M | student | 55408
929 | 44 M | scientist | 53711
930 | 33 M | teacher | 53173
931 | 60 M | educator | 33556
932 | 59 M | educator | 4747
933 | 28 M | teacher | 48105
934 | 61 M | engineer | 22902
935 | 24 M | student | 11121
936 | 24 M | other | 32789
937 | 48 M | educator | 98672
```

```
VM instances - Compute Engine - danddankics432homew... - Goo...
nalongsonen_danddank@danddankhomework1mysql:~$ mysql --local-infile=1 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, 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> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| movie_ratings_database |
| mysql          |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.00 sec)

mysql> use movie_ratings_database;
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> show tables;
+-----+
| Tables_in_movie_ratings_database |
+-----+
| items        |
| ratings      |
| users        |
+-----+
3 rows in set (0.00 sec)

mysql> CREATE TABLE items(movieid int primary key, movietitle varchar(1000), releasedate varchar(50)[], video
rel_date varchar(50), IMDb_URL varchar(500), unknown tinyint(1), action tinyint(1), adventure tinyint(1), drama
tinyint(1), fantasy tinyint(1), film_noir tinyint(1), horror tinyint(1), musical tinyint(1), mystery tinyint(1),
romance tinyint(1), sci fi tinyint(1), thriller tinyint(1), war tinyint(1), western tinyint(1);
```

```
VM instances - Compute Engine - danddankics432homew... - Goo... ssh.cloud.google.com nalongsone_danddank@danddankhomework1mysql:~$ mysql --local-infile=1 -u root -p Enter password: Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 16 Server version: 8.0.26 MySQL Community Server - GPL Copyright (c) 2000, 2021, 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> show databases; +-----+ | Database | +-----+ | information_schema | | movie_ratings_database | | mysql | | performance_schema | | sys | +-----+ 5 rows in set (0.00 sec) mysql> use movie_ratings_database; 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> show tables; +-----+ | Tables_in_movie_ratings_database | +-----+ | items | | ratings | | users | +-----+ 3 rows in set (0.00 sec) mysql> CREATE TABLE ratings(userid int not null, movieid int not null, rating int, timestamp varchar(100), C ONSTRAINT PK_rating (userid, movieid), FOREIGN KEY(userid) REFERENCES users(userid), FOREIGN KEY(movieid) RE FERENCES items(movieid));
```

```
VM instances - Compute Engine - danddankics432homew... - Goo... ssh.cloud.google.com nalongsone_danddank@danddankhomework1mysql:~$ mysql --local-infile=1 -u root -p Enter password: Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 16 Server version: 8.0.26 MySQL Community Server - GPL Copyright (c) 2000, 2021, 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> show databases; +-----+ | Database | +-----+ | performance_schema | | sys | +-----+ 5 rows in set (0.00 sec) mysql> use movie_ratings_database; 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> show tables; +-----+ | Tables_in_movie_ratings_database | +-----+ | items | | ratings | | users | +-----+ 3 rows in set (0.00 sec) mysql> DESC items; +-----+ | Field | | Type | | Null | | Key | | Default | | Extra | +-----+ | movieid | | int | | NO | | PRI | | NULL | | | | movieitle | | varchar(1000) | | YES | | NULL | | NULL | | | | releasedate | | varchar(50) | | YES | | NULL | | NULL | | | | video_rel_date | | varchar(50) | | YES | | NULL | | NULL | | | | IMDb_URL | | varchar(500) | | YES | | NULL | | NULL | | | | unknown | | tinyint(1) | | YES | | NULL | | NULL | | | | action | | tinyint(1) | | YES | | NULL | | NULL | | | | adventure | | tinyint(1) | | YES | | NULL | | NULL | | | | animation | | tinyint(1) | | YES | | NULL | | NULL | | | | children | | tinyint(1) | | YES | | NULL | | NULL | | | | comedy | | tinyint(1) | | YES | | NULL | | NULL | | | | crime | | tinyint(1) | | YES | | NULL | | NULL | | | | documentary | | tinyint(1) | | YES | | NULL | | NULL | | | | drama | | tinyint(1) | | YES | | NULL | | NULL | | | | fantasy | | tinyint(1) | | YES | | NULL | | NULL | | | | film_noir | | tinyint(1) | | YES | | NULL | | NULL | | | | horror | | tinyint(1) | | YES | | NULL | | NULL | | | | musical | | tinyint(1) | | YES | | NULL | | NULL | | | | mystery | | tinyint(1) | | YES | | NULL | | NULL | | | | romance | | tinyint(1) | | YES | | NULL | | NULL | | | | scifi | | tinyint(1) | | YES | | NULL | | NULL | | | | thriller | | tinyint(1) | | YES | | NULL | | NULL | | | | war | | tinyint(1) | | YES | | NULL | | NULL | | | | western | | tinyint(1) | | YES | | NULL | | NULL | | +-----+ 24 rows in set (0.00 sec) mysql>
```

```

ssh.cloud.google.com
VM instances - Compute Engine - danddankics432homew... - Goo...
nalongsonे_danddank@danddankhomework1mysql: ~

+-----+-----+-----+-----+-----+
| movieid | int   | NO    | PRI  | NULL |
| movietitle | varchar(1000) | YES   |       | NULL |
| releasedate | varchar(50) | YES   |       | NULL |
| video_rel_date | varchar(50) | YES   |       | NULL |
| IMDb_URL | varchar(500) | YES   |       | NULL |
| unknown | tinyint(1) | YES   |       | NULL |
| action | tinyint(1) | YES   |       | NULL |
| adventure | tinyint(1) | YES   |       | NULL |
| animation | tinyint(1) | YES   |       | NULL |
| children | tinyint(1) | YES   |       | NULL |
| comedy | tinyint(1) | YES   |       | NULL |
| crime | tinyint(1) | YES   |       | NULL |
| documentary | tinyint(1) | YES   |       | NULL |
| drama | tinyint(1) | YES   |       | NULL |
| fantasy | tinyint(1) | YES   |       | NULL |
| film_noir | tinyint(1) | YES   |       | NULL |
| horror | tinyint(1) | YES   |       | NULL |
| musical | tinyint(1) | YES   |       | NULL |
| mystery | tinyint(1) | YES   |       | NULL |
| romance | tinyint(1) | YES   |       | NULL |
| scifi | tinyint(1) | YES   |       | NULL |
| thriller | tinyint(1) | YES   |       | NULL |
| war | tinyint(1) | YES   |       | NULL |
| western | tinyint(1) | YES   |       | NULL |
+-----+-----+-----+-----+-----+
24 rows in set (0.00 sec)

mysql> desc ratings;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| userid | int   | NO    | PRI  | NULL |
| movieid | int   | NO    | PRI  | NULL |
| rating | int   | YES   |       | NULL |
| timestamp | varchar(100) | YES   |       | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc users;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| userid | int   | NO    | PRI  | NULL |
| age   | int   | YES   |       | NULL |
| gender | varchar(5) | YES   |       | NULL |
| occupation | varchar(100) | YES   |       | NULL |
| zipcode | varchar(10) | YES   |       | NULL |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> 

```

Homework report screen-shot #21: *99548 rows in the rating table*

```

ssh.cloud.google.com
VM instances - Compute Engine - danddankics432homew... - Goo...
nalongsonе_данддакн@данддакнhomework1mysql: ~
nalongsonе_данддакн@данддакнhomework1mysql: ~$ mysql --local-infile=1 -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, 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> show databases;
+-----+
| Database |
+-----+
| information schema |
| movie_ratings_database |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use movie_ratings_database;
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> show tables;
+-----+
| Tables_in_movie_ratings_database |
+-----+
| items |
| ratings |
| users |
+-----+
3 rows in set (0.00 sec)

mysql> CREATE TABLE ratings(userid int not null, movieid int not null, rating int, timestamp varchar(100), PRIMARY KEY(userid, movieid), FOREIGN KEY(userid) REFERENCES users(userid), FOREIGN KEY(movieid) REFERENCES items(movieid));

```

Homework report screen-shot #22: *1682 rows in the items table*

```
 ⑤ VM Instances - Compute Engine · danddark@dashdark:home · Google Cloud Platform
 ⑥ ssh.cloud.google.com
 ⑦ walongsonc_danddark@dashdark/home/walysql:~$ 

| movieid | movieTitle | releaseDate | video_rel_date | IMDB_URL |
+-----+-----+-----+-----+-----+
| 1 | Toy Story (1995) | 1-Jan-95 | 1-Jan-95 | http://us.imdb.com/M/title-exact/ToyStory20(1995) |
| 2 | Goldengate (1995) | 0 | 0 | http://us.imdb.com/M/title-exact/GoldenEye20(1995) |
| 3 | Four Rooms (1995) | 0 | 0 | http://us.imdb.com/M/title-exact/FourRooms20(1995) |
| 0 | 0 | 0 | 0 | http://us.imdb.com/M/title-exact/GetShorty20(1995) |
| 1 | 0 | 0 | 0 | http://us.imdb.com/M/title-exact/GetShorty20(1995) |
| 5 | Copycat (1995) | 1 | 0 | http://us.imdb.com/M/title-exact/CopyCat20(1995) |
| 6 | Shanghai Triad (Tao ya yan da waiqiao) (1995) | 1-Jan-95 | 0 | http://us.imdb.com/M/title-exact/TaoYanDaWaiQiao20(1995) |
| 7 | Twelve Monkeys (1995) | 0 | 0 | http://us.imdb.com/M/title-exact/TwelveMonkeys20(1995) |
| 8 | Babe (1995) | 0 | 0 | http://us.imdb.com/M/title-exact/Babe20(1995) |
| 0 | 0 | Dead Man Walking (1995) | 0 | http://us.imdb.com/M/title-exact/Dead20Man20Walking20(1995) |
| 10 | Richard III (1995) | 1 | 0 | http://us.imdb.com/M/title-exact/Richard20III20(1995) |

10 rows in set (0.00 sec)

mysql> select count(*) from ratings;
+-----+
| count(*) |
+-----+
| 99548 |
+-----+
1 row in set (0.01 sec)

mysql> select * from ratings limit 10;
+-----+-----+-----+-----+
| userid | movieid | rating | timestamp |
+-----+-----+-----+-----+
| 1 | 1 | 2 | 3 | 874693371 |
| 1 | 1 | 2 | 3 | 874693260 |
| 1 | 1 | 4 | 3 | 876893119 |
| 1 | 1 | 5 | 3 | 876893222 |
| 1 | 1 | 6 | 5 | 887431973 |
| 1 | 1 | 7 | 4 | 875071561 |
| 1 | 1 | 8 | 3 | 878543563 |
| 1 | 1 | 9 | 5 | 878543561 |
| 1 | 1 | 10 | 3 | 875693209 |
| 1 | 1 | 11 | 2 | 875693262 |

10 rows in set (0.00 sec)

mysql> exit
Bye
walongsonc_danddark@dashdark/home/walysql:~$ 
```

Homework report screen-shot #23:

1- Find how many ratings are there for each of the following values (5,4,3,2, and 1). Draw a histogram to show these values

The screenshot shows a terminal window titled 'VM instances - Compute Engine - danddankics432homew... - Google Cloud Platform' connected via ssh to 'nalongsone_danddankhomework!mysql'. The MySQL prompt is visible at the bottom. The terminal displays several SQL queries using the 'count(*)' function to determine the number of rows for each rating value (1, 2, 3, 4, 5) in the 'ratings' table of the 'movie_ratings_database'. The results are as follows:

```
mysql> select count(*) from ratings where rating=1;
+-----+
| count(*) |
+-----+
| 6102 |
+-----+
1 row in set (0.02 sec)

mysql> select count(*) from ratings where rating=2;
+-----+
| count(*) |
+-----+
| 11343 |
+-----+
1 row in set (0.02 sec)

mysql> select count(*) from ratings where rating=3;
+-----+
| count(*) |
+-----+
| 27049 |
+-----+
1 row in set (0.02 sec)

mysql> select count(*) from ratings where rating=4;
+-----+
| count(*) |
+-----+
| 33972 |
+-----+
1 row in set (0.02 sec)

mysql> select count(*) from ratings where rating=5;
+-----+
| count(*) |
+-----+
| 21082 |
+-----+
1 row in set (0.02 sec)

mysql>
```

Rating	Number of rating
1	6102
2	11343
3	27049
4	33972
5	21082

Homework report screen-shot #24:

2- Find how many distinct zip codes are there: 795

Homework report screen-shot #25:

3- Find how many users are there from each zip code.

```
ssh.cloud.google.com
VM instances - Compute Engine - danddankics432homew... - Google Cloud Platform
nalongsonen_danddank@danndankhomework1mysql: ~

1 row in set (0.00 sec)

mysql> select zipcode, count(userid) from users group by zipcode order by zipcode limit 20;
+-----+-----+
| zipcode | count(userid) |
+-----+-----+
| 0       |      2 |
| 10003   |      5 |
| 10010   |      1 |
| 10011   |      1 |
| 10016   |      2 |
| 10018   |      1 |
| 10019   |      2 |
| 1002    |      1 |
| 10021   |      3 |
| 10022   |      2 |
| 10025   |      2 |
| 10309   |      1 |
| 10314   |      1 |
| 1040    |      1 |
| 10522   |      1 |
| 10707   |      1 |
| 1080    |      1 |
| 10960   |      2 |
| 11101   |      1 |
| 11201   |      1 |
+-----+
20 rows in set (0.00 sec)

mysql> [REDACTED]
```

```

+-----+-----+
| 29301 |      1 |
| 53210 |      1 |
| 6512  |      1 |
| 76201 |      1 |
| 8105  |      1 |
| 60614 |      1 |
| N2L5N |      1 |
| 20006 |      1 |
| 70116 |      1 |
| 90008 |      1 |
| 98801 |      1 |
| E2E3R |      1 |
| 11753  |      1 |
| 49036 |      1 |
| 1701   |      1 |
| 55428  |      1 |
| 7310   |      1 |
| 33556  |      1 |
| 6437   |      1 |
| 48105 |      1 |
| 66221 |      1 |
| 32789 |      1 |
| 55038 |      1 |
| 33319 |      1 |
| 97229 |      1 |
| 78209 |      1 |
| 77841 |      1 |
+-----+
795 rows in set (0.00 sec)

mysql> 
```

Homework report screen-shot #26:

4- What is the age and occupation of the user who gave the maximum number of reviews?

`select userid, age, occupation from users where users.userid = (select userid from ratings group by userid order by count(userid) desc limit 1);`

```

+-----+-----+
| userid | age   | occupation |
+-----+-----+
| 405   | 22    | healthcare |
+-----+
1 row in set (0.02 sec)

mysql> select userid, count(userid)as num from ratings group by userid order by num desc limit 10;
+-----+-----+
| userid | num  |
+-----+-----+
| 405   | 737  |
| 655   | 684  |
| 13    | 635  |
| 450   | 539  |
| 276   | 517  |
| 416   | 492  |
| 537   | 489  |
| 303   | 483  |
| 234   | 479  |
| 393   | 447  |
+-----+-----+
10 rows in set (0.02 sec)

mysql> select userid, age, occupation from users where users.userid = (select userid from ratings group by userid order by count(userid) desc limit 1);
+-----+-----+
| userid | age   | occupation |
+-----+-----+
| 405   | 22    | healthcare |
+-----+
1 row in set (0.02 sec)

mysql> 
```

Homework report screen-shot #27:

5- What are the top five zip codes in terms of number of users? How many users are there in each one of these zip codes.

`select zipcode, count(zipcode) as num_user from users group by zipcode order by num_user desc limit 5;`

```
ssh.cloud.google.com
VM instances - Compute Engine - danddankics432...
nalongsonen_danddank@danddankhomework1mysql: ~
SSH from the browser | Compute Engine Document...
20009 | 5 |
55337 | 5 |
10003 | 5 |
55408 | 4 |
27514 | 4 |
55454 | 4 |
2215 | 3 |
55108 | 3 |
55113 | 3 |
11217 | 3 |
97301 | 3 |
55106 | 3 |
80525 | 3 |
60201 | 3 |
22903 | 3 |
22902 | 3 |
48103 | 3 |
94043 | 3 |
+-----+
20 rows in set (0.00 sec)

mysql> select zipcode, count(zipcode) as num_user from users group by zipcode order by num_user desc limit 5;
+-----+-----+
| zipcode | num_user |
+-----+-----+
| 55414 | 9 |
| 55105 | 6 |
| 20009 | 5 |
| 55337 | 5 |
| 10003 | 5 |
+-----+
5 rows in set (0.00 sec)

mysql> ■
```

Homework report screen-shot #28:

6- Find how many different occupations are there. -> 21

[select count\(distinct\(occupation\)\) from users;](#)

```
ssh.cloud.google.com
VM instances - Compute Engine - danddankics432...
nalongsonen_danddank@danddankhomework1mysql: ~
SSH from the browser | Compute Engine Document...
mysql> select distinct(occupation) from users;
+-----+
| occupation |
+-----+
| technician |
| other |
| writer |
| executive |
| administrator |
| student |
| lawyer |
| educator |
| scientist |
| entertainment |
| programmer |
| librarian |
| homemaker |
| artist |
| engineer |
| marketing |
| none |
| healthcare |
| retired |
| salesman |
| doctor |
+-----+
21 rows in set (0.00 sec)

mysql> select count(distinct(occupation)) from users;
+-----+
| count(distinct(occupation)) |
+-----+
| 21 |
+-----+
1 row in set (0.00 sec)

mysql> ■
```

Homework report screen-shot #29:

7- What are the five top occupations in terms of the number of users? How many users are there in each one of these top five occupations?

```
select occupation, count(occupation) as num_user from users group by occupation order by num_user desc limit 5;
```

The screenshot shows an SSH session on a Compute Engine VM. The user has run several MySQL commands:

- A list of all distinct occupations in the 'users' table.
- A query to count the number of distinct occupations.
- A query to find the top 5 occupations by user count, ordered by user count in descending order.

```
| scientist      |
| entertainment |
| programmer    |
| librarian     |
| homemaker     |
| artist         |
| engineer       |
| marketing     |
| none           |
| healthcare    |
| retired        |
| salesman       |
| doctor          |
+-----+
21 rows in set (0.00 sec)

mysql> select count(distinct(occupation)) from users;
+-----+
| count(distinct(occupation)) |
+-----+
|                      21 |
+-----+
1 row in set (0.00 sec)

mysql> select occupation, count(occupation) as num_user from users group by occupation
   order by num_user desc limit 5;
+-----+-----+
| occupation | num_user |
+-----+-----+
| student    |     196 |
| other      |     105 |
| educator   |      95 |
| administrator |     79 |
| engineer   |      67 |
+-----+-----+
5 rows in set (0.00 sec)

mysql> ■
```

Homework report screen-shot #30:

8- What are the top five movie ids in terms of the number of ratings.

```
select movieid, count(movieid) as num_rating from ratings group by movieid order by num_rating desc limit 5;
```

The screenshot shows an SSH session on a Compute Engine VM. The user has run several MySQL commands:

- A query to find the top 5 movie IDs based on the number of ratings.
- A query to display the first 5 rows of the 'ratings' table.
- A query to find the top 5 movie IDs based on the number of ratings, but this query fails due to a syntax error in the GROUP BY clause.
- A corrected query to find the top 5 movie IDs based on the number of ratings.

```
| educator | 95 |
| administrator | 79 |
| engineer | 67 |
+-----+
5 rows in set (0.00 sec)

mysql> select * from ratings limit 5;
+-----+-----+-----+-----+
| userid | movieid | rating | timestamp |
+-----+-----+-----+-----+
| 1 | 2 | 3 | 876893171 |
| 1 | 3 | 4 | 878542960 |
| 1 | 4 | 3 | 876893119 |
| 1 | 5 | 3 | 889751712 |
| 1 | 6 | 5 | 887431973 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select movieid, count(userid) as num_rating from ratings group by userid order by num_rating desc limit 5;
ERROR 1055 (42000): Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'movie_ratings.database.ratings.movieid' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by
mysql> select movieid, count(movieid) as num_rating from ratings group by movieid order by num_rating desc limit 5;
+-----+-----+
| movieid | num_rating |
+-----+-----+
| 50 | 583 |
| 258 | 509 |
| 100 | 508 |
| 181 | 507 |
| 294 | 485 |
+-----+-----+
5 rows in set (0.02 sec)

mysql> ■
```

Homework report screen-shot #31:

9- Find the top ten movie ids in terms of average rating? What are the titles for these movies. You can use more than one query.

```
create view mid_tnr as select movieid, count(movieid) as num_rating from ratings group by movieid order by movieid;
```

```
create view mid_rxn as select movieid, rating, count(movieid) as num_rating,(rating * count(movieid)) as rxn from ratings group by movieid, rating order by movieid;
```

```
create view mid_rxn_sum as select movieid, sum(rxn) as sum_rxn from mid_rxn group by movieid;
```

```
create view tnr_join_rxns as select mid_tnr.movieid, sum_rxn, num_rating from mid_rxn_sum join mid_tnr on mid_rxn_sum.movieid = mid_tnr.movieid;
```

```
create view mid_avr as select movieid, sum_rxn, num_rating, (sum_rxn/num_rating) as average_rating from tnr_join_rxns order by average_rating desc;
```

```
create view mid_avr_title as (select mid_avr.movieid, movietitle, sum_rxn, num_rating, average_rating from mid_avr join items on mid_avr.movieid = items.movieid order by average_rating desc, num_rating desc);
```

```
select movieid, movietitle, average_rating from mid_avr_title limit 10;
```

movieid	movietitle	average_rating
1189	Prefontaine (1997)	5.0000
1293	Star Kid (1997)	5.0000
1467	"Saint of Fort Washington	5.0000
1500	Santa with Muscles (1996)	5.0000
814	"Great Day in Harlem	5.0000
1201	Marlene Dietrich: Shadow and Light (1996)	5.0000
1653	Entertaining Angels: The Dorothy Day Story (1996)	5.0000
1599	Someone Else's America (1995)	5.0000
1122	They Made Me a Criminal (1939)	5.0000
1536	Aiqing wansui (1994)	5.0000

```

ssh.cloud.google.com
VM instances - Compute Engine - danddankics432homew... - Google Cloud Platform
nalongsonc_danddank@danddankhomework1:mysql> id_avr join items on mid_avr.movieid = items.movieid order by average_rating desc, num_rating desc);
Query OK, 0 rows affected (0.01 sec)

mysql> select * from mid_avr_title limit 10;
+-----+-----+-----+-----+
| movieid | movietitle | sum_rxn | num_rating | average_rating |
+-----+-----+-----+-----+
| 1189 | Prefontaine (1997) | 15 | 3 | 5.0000 |
| 1293 | Star Kid (1997) | 15 | 3 | 5.0000 |
| 1467 | "Saint of Fort Washington | 10 | 2 | 5.0000 |
| 1500 | Santa with Muscles (1996) | 10 | 2 | 5.0000 |
| 814 | "Great Day in Harlem | 5 | 1 | 5.0000 |
| 1201 | Marlene Dietrich: Shadow and Light (1996) | 5 | 1 | 5.0000 |
| 1653 | Entertaining Angels: The Dorothy Day Story (1996) | 5 | 1 | 5.0000 |
| 1599 | Someone Else's America (1995) | 5 | 1 | 5.0000 |
| 1122 | They Made Me a Criminal (1939) | 5 | 1 | 5.0000 |
| 1536 | Aiqing wansui (1994) | 5 | 1 | 5.0000 |
+-----+-----+-----+-----+
10 rows in set (0.10 sec)

mysql> select movieid, movietitle, average_rating from mid_avr_title limit 10;
+-----+-----+-----+
| movieid | movietitle | average_rating |
+-----+-----+-----+
| 1189 | Prefontaine (1997) | 5.0000 |
| 1293 | Star Kid (1997) | 5.0000 |
| 1467 | "Saint of Fort Washington | 5.0000 |
| 1500 | Santa with Muscles (1996) | 5.0000 |
| 814 | "Great Day in Harlem | 5.0000 |
| 1201 | Marlene Dietrich: Shadow and Light (1996) | 5.0000 |
| 1653 | Entertaining Angels: The Dorothy Day Story (1996) | 5.0000 |
| 1599 | Someone Else's America (1995) | 5.0000 |
| 1122 | They Made Me a Criminal (1939) | 5.0000 |
| 1536 | Aiqing wansui (1994) | 5.0000 |
+-----+-----+-----+
10 rows in set (0.09 sec)

mysql>

```

Homework report screen-shot #32:

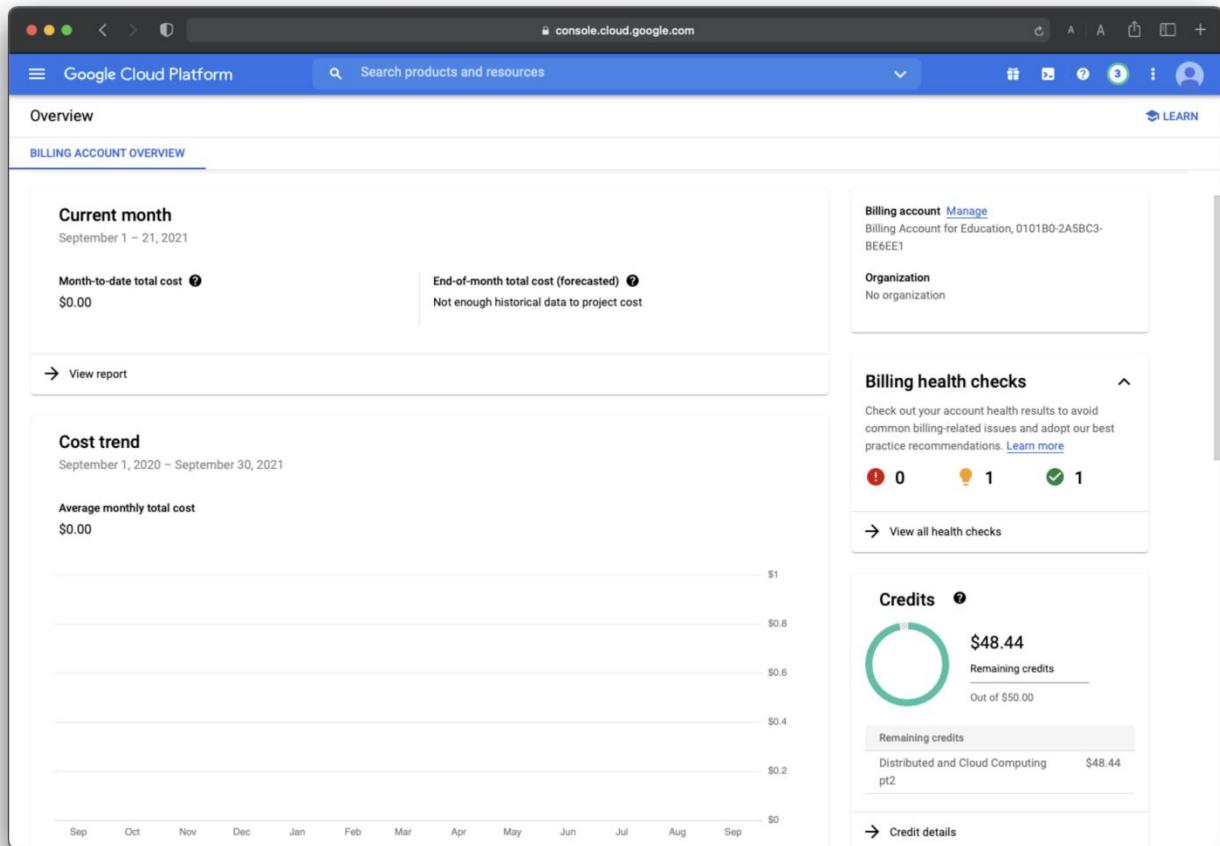
The screenshot shows the Google Cloud Platform Compute Engine interface. The left sidebar navigation includes:

- Compute Engine** (selected)
- Virtual machines** (selected)
 - VM instances** (selected)
 - Instance templates
 - Sole-tenant nodes
 - Machine images
 - TPUs
 - Committed use discounts
 - Migrate for Compute Engi...
- Storage
- Instance groups
- VM Manager

The main content area displays the details for the VM instance **danddankhomework1mysql**. Key sections include:

- Details** tab (selected): Shows the instance ID (445321256354140480), machine type (e2-medium (2 vCPUs, 4 GB memory)), and creation time (Sep 19, 2021, 3:07:11 AM).
- Network interfaces**: Shows a single interface named **nic0** with settings: Network (default), Subnetwork (default), Primary Internal IP (10.128.0.2), External IP (Ephemeral), Network Tier (Premium), and IP forwarding (Off). A link to **View details** is present.
- Public DNS PTR Record**: Shows the PTR record for the instance.

Homework report screen-shot #33:



The screenshot shows the Google Cloud Platform Billing Account Overview page. At the top, there's a navigation bar with the Google Cloud logo, a search bar, and various icons. Below the header, the main content area is divided into several sections:

- Current month**: Shows the period from September 1 – 21, 2021. It displays the Month-to-date total cost (\$0.00) and the End-of-month total cost (forecasted), which is noted as "Not enough historical data to project cost". A link to "View report" is provided.
- Billing account**: Shows the Billing Account for Education, ID 0101B0-2A5BC3-BE6EE1. It also indicates that there is no organization.
- Organization**: Shows that there is no organization.
- Billing health checks**: Provides account health results with 0 critical issues, 1 warning, and 1 green checkmark. A link to "View all health checks" is available.
- Credits**: Displays a remaining credit balance of \$48.44 out of a \$50.00 limit. It lists "Distributed and Cloud Computing pt2" as the source of the credit.
- Cost trend**: A chart showing average monthly total cost over time, with data points from September 2020 to September 2021. The y-axis ranges from \$0 to \$1.

Assignment #2 : Load Balancing and Auto Scaling

Exercise 1: Implementing a scalable AWS infrastructure to host a web site.

Homework report screen-shot #1:

The screenshot shows the AWS EC2 Management Console with the URL <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroup:groupid=sg-01526ecc2e712af5a>. The left sidebar shows the navigation menu for EC2 services. The main content area displays the details of a security group named "danddank-ics432website-securitygroup". The "Details" section includes fields for Security group name (danddank-ics432website-securitygroup), Security group ID (sg-01526ecc2e712af5a), Description ("This security group is used by the instance group that hosts ICS 432 web site"), and VPC ID (vpc-64fa8419). Below this, there are sections for Owner (196580907486), Inbound rules count (2 Permission entries), and Outbound rules count (1 Permission entry). A tab for "Inbound rules" is selected, showing two rules: one for port 80 (HTTP) and another for port 22 (SSH). A button to "Run Reachability Analyzer" is also present.

Homework report screen-shot #2:

The screenshot shows an SSH terminal window titled "ec2-user@ip-172-31-16-135:/data/www/ICS432Fall2021 (ssh)". The terminal displays the content of an HTML file. The file starts with a DOCTYPE declaration and contains a simple header and body structure. The header includes meta charset="utf-8" and title="Home - ICS 432 testing Web site". The body features a large centered image of the Metro State logo and a navigation menu with links to index.html, about-us.html, and services.html.

```
ec2-user@ip-172-31-16-135:/data/www/ICS432Fall2021 (ssh) ~% curl http://localhost<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Home - ICS 432 testing Web site</title>
<style>
p{font-size: 30px;}
```

Homework report screen-shot #3:

My Classrooms Workbench Instance details | EC2 Manager Home - ICS 432 testing Web ...

Not Secure | 54.144.252.25/index.html

Metropolitan State University

ICS 432 Distributed and Cloud Computing Fall 2021

This is a very simple web site for testing purposes.

Cloud Computing VS *Distributed Computing*

Homework report screen-shot #4:

My Classrooms Workbench AMIs | EC2 Management Console

console.aws.amazon.com/ec2/v2/home?region=north-east-1#images:visibility=owned-by-me;sort=name

aws Services ▾

Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations

Images **AMIs**

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Load Balancing Load Balancers Target Groups

Auto Scaling Launch Configurations Auto Scaling Groups

Feedback English (US) ▾

Launch EC2 Image Builder Actions

Owned by me Filter by tags and attributes or search by keyword

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform	Root Device Type	Virtualization	Deprecation Time
Danddark-ICS432-Image	ami-0358b99a865b73bf5	196580907486...	196580907486...	Private	available	October 19, 2021 at 11:55:2...	Other Linux	ebs	hvm	-	

Image: ami-0358b99a865b73bf5

Details Permissions Tags

AMI ID: ami-0358b99a865b73bf5	AMI Name: Danddark-ICS432-Image
Owner: 196580907486	Source: 196580907486/Danddark-ICS432-Image
Status: available	State Reason: -
Creation date: October 19, 2021 at 11:55:27 PM UTC-5	Platform details: Linux/UNIX
Architecture: x86_64	Usage operation: RunInstances
Image Type: machine	Virtualization type: hvm

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Homework report screen-shot #5:

Screenshot of the AWS EC2 Management Console showing the 'CreateLaunchConfiguration' page. The page is for creating a launch configuration for an Auto Scaling group. It includes fields for RAM disk ID, Metadata accessible, Metadata version, Metadata response hop limit, User data (containing a shell script to print the public IP), and IP address type (set to 'Only assign a public IP address to instances launched in a subnet with auto-assign public IP enabled (default)'). A note at the bottom states: 'Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.'

Homework report screen-shot #6:

Screenshot of a web browser displaying a simple website for 'Metropolitan State University'. The page title is 'ICS 432 Distributed and Cloud Computing Fall 2021'. Below the title is a sub-header: 'This is a very simple web site for testing purposes.' To the left is a navigation menu with links: Home, About Us, Topics Covered, Contact Us, and Server IP. To the right is a diagram comparing 'Cloud Computing' and 'Distributed Computing'. The 'Cloud Computing' section shows a central cloud icon containing 'Storage', 'Database', 'Compute', and 'Applications', with arrows pointing from various devices (laptop, smartphone, tablet) to the cloud. The 'Distributed Computing' section shows a central node connected to multiple other nodes, with arrows labeled 'Task' and 'Result' indicating data flow between them.

My Classrooms Workbench Instances | EC2 Management Home - ICS 432 testing Web Home - ICS 432 testing Web

Not Secure | 54.196.57.121/index.html

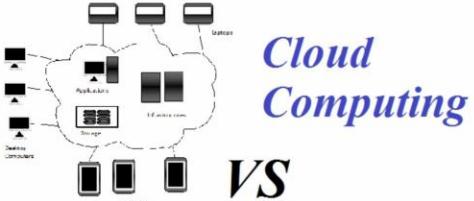
Metropolitan State University



ICS 432 Distributed and Cloud Computing Fall 2021

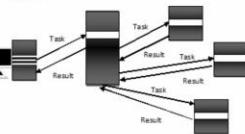
This is a very simple web site for testing purposes.

Cloud Computing



VS

Distributed Computing



Homework report screen-shot #7:

My Classrooms Workbench Instances | EC2 Management About Us - Very Simple Web Site About Us - Very Simple Web Site

Not Secure | 54.196.57.121/server-ip.html

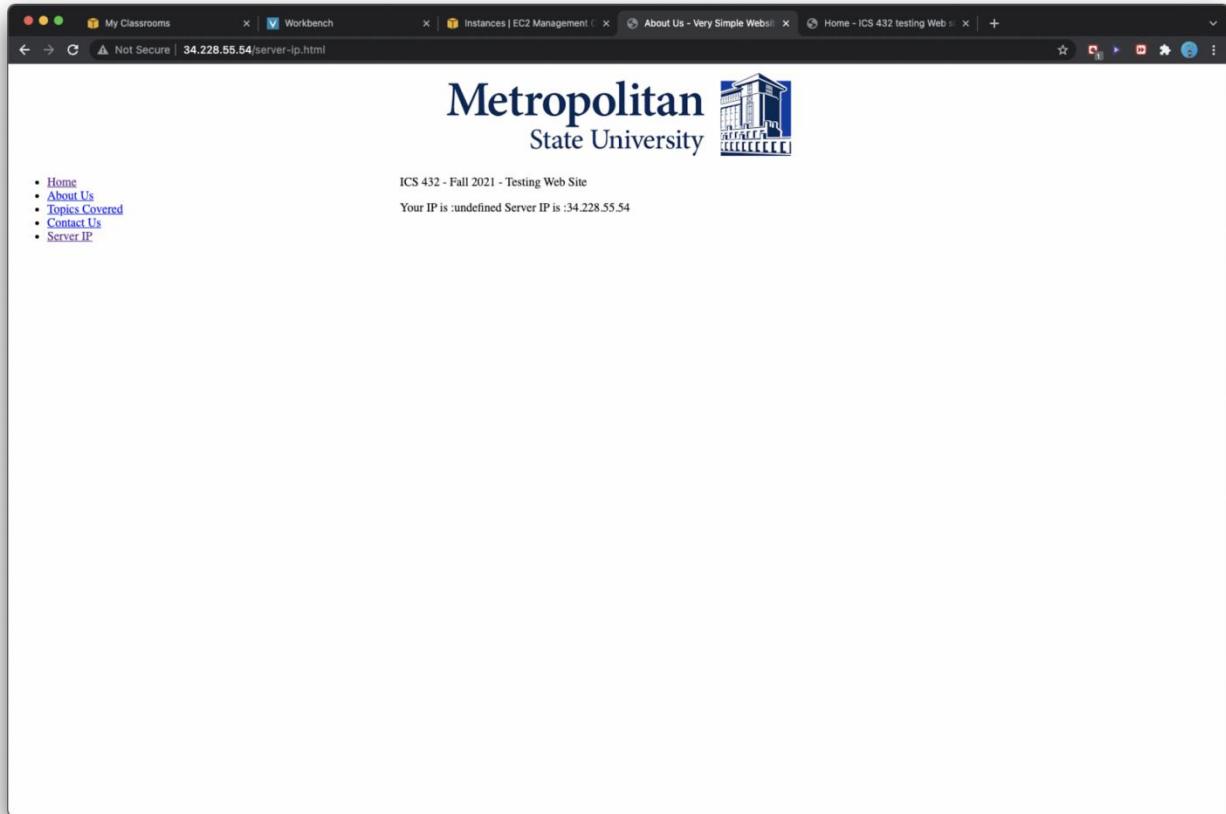
Metropolitan State University



ICS 432 - Fall 2021 - Testing Web Site

Your IP is :undefined Server IP is :54.196.57.121

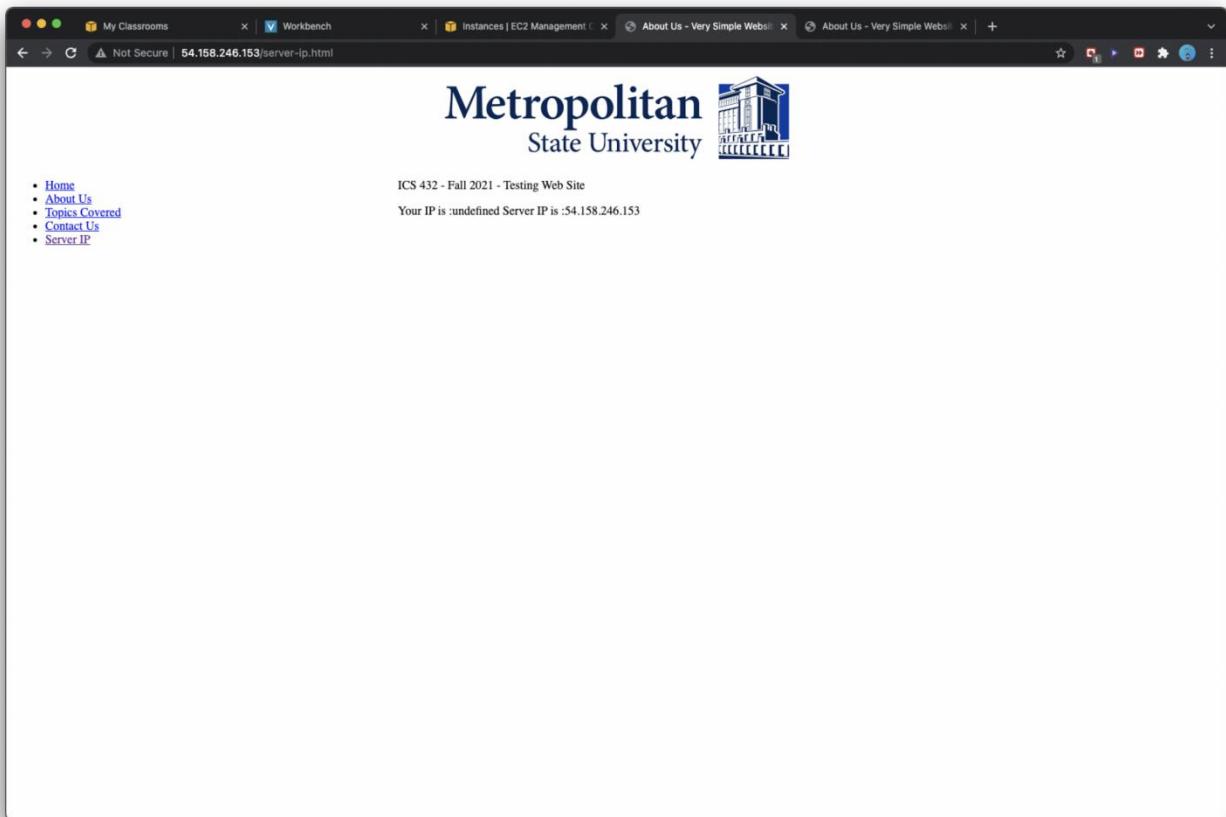
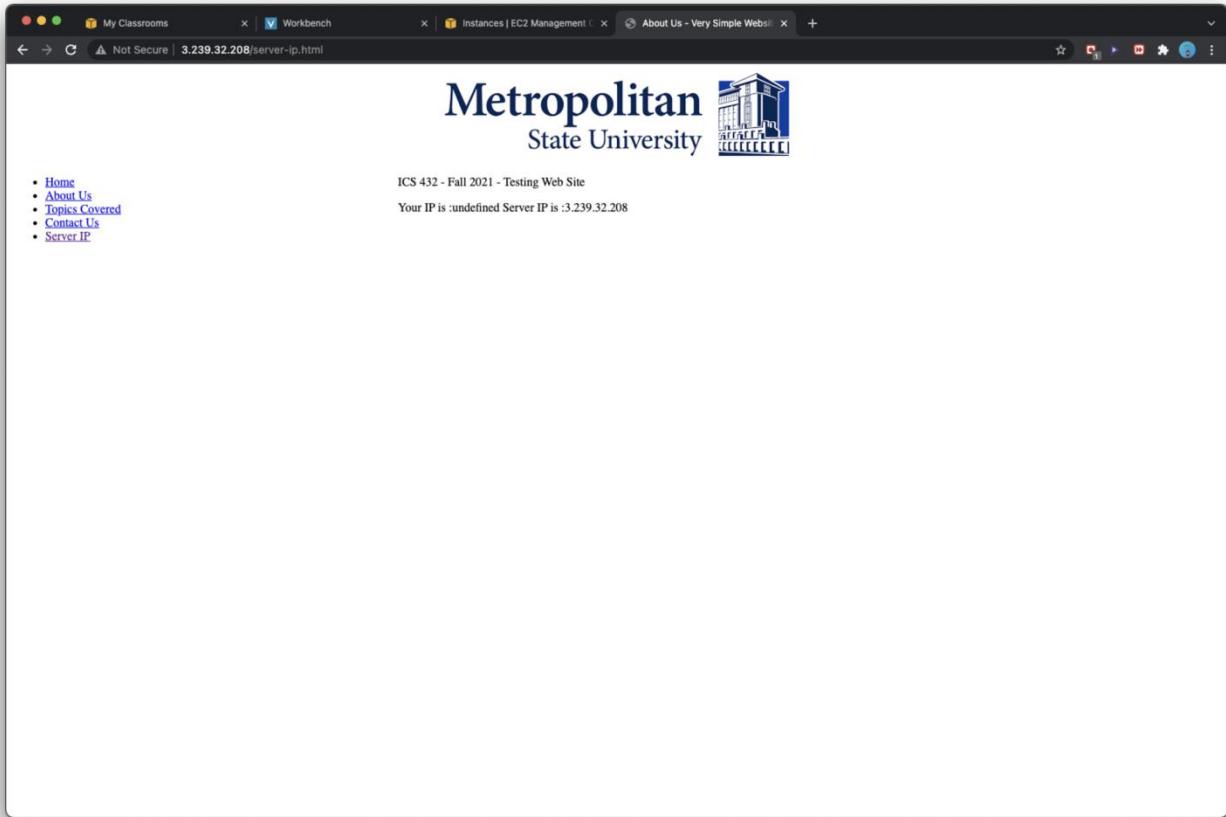
• Home
• About Us
• Topics Covered
• Contact Us
• Server IP



Homework report screen-shot #8:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
run-by-auto-scaling-group2	i-0fbaf4b8b8774f270	Running	t3.micro	Initializing	No alarms	us-east-1c	ec2-5-23
Dandank Web Server	i-05a38511178bf90a0	Stopped	t2.micro	-	No alarms	us-east-1a	-
DandankLab4Instance3	i-022df32823332594e	Stopped	t2.micro	-	No alarms	us-east-1a	-
dandank-ics432website-instance	i-0a4af20a30b1214a	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-3-89
nalongson-ics432website-instance	i-04b45fe4b62fdfa45	Stopped	t2.micro	-	No alarms	us-east-1a	-
run-by-auto-scaling-group1	i-0460556cc3c152b65	Running	t3.micro	Initializing	No alarms	us-east-1a	ec2-54-1

Homework report screen-shot #9:



Homework report screen-shot #10:

Screenshot of the AWS EC2 Management Console showing the Load Balancers page.

Left Sidebar:

- Instances [New](#)
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances [New](#)
- Dedicated Hosts
- Scheduled Instances
- Capacity Reservations
- Images
- AMIs
- Elastic Block Store
 - Volumes
 - Snapshots
 - Lifecycle Manager [New](#)
- Network & Security
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs
 - Network Interfaces
- Load Balancing
 - Load Balancers**
 - Target Groups [New](#)
- Auto Scaling
 - Launch Configurations
 - Auto Scaling Groups

Top Bar: My Classrooms | Workbench | EC2 Management Console | Search for services, features, marketplace products, and docs [Option+5] | vocstartsoft/user1564948=Nalongsone_Danddank @ 1965-8090-7486 | N. Virginia | Support

Table: List of Load Balancers

Name	DNS name	State	VPC ID	Availability Zones	Type
danddank-ics432web-load-b...	danddank-ics432web-load-balancer-572798612.us-east-1.elb.amazonaws.com	InService	vpc-64fa8419	us-east-1f, us-east-1e, ...	clas

Section: Load balancer: danddank-ics432web-load-balancer

Tabs: Description, **Instances**, Health check, Listeners, Monitoring, Tags, Migration

Text: Connection Draining: Enabled, 300 seconds ([Edit](#))

Tables:

- Edit Instances:**

Instance ID	Name	Availability Zone	Status	Actions
i-0fbaf4b8b8774f270	run-by-auto-scaling-group2	us-east-1c	InService	Remove from Load Balancer
i-0460556cc3c152b65	run-by-auto-scaling-group1	us-east-1a	InService	Remove from Load Balancer
- Edit Availability Zones:**

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
us-east-1f	subnet-399b7c38	172.31.64.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1e	subnet-7c54254d	172.31.48.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1d	subnet-86dc4a7	172.31.80.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1c	subnet-15466b73	172.31.0.0/20	1	Yes	Remove from Load Balancer
us-east-1b	subnet-310b256e	172.31.32.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer
us-east-1a	subnet-9bf20fd7	172.31.16.0/20	1	Yes	Remove from Load Balancer

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Homework report screen-shot #11:

Screenshot of a web browser displaying a simple website for Metropolitan State University.

Page Title: About Us - Very Simple Website

Header:

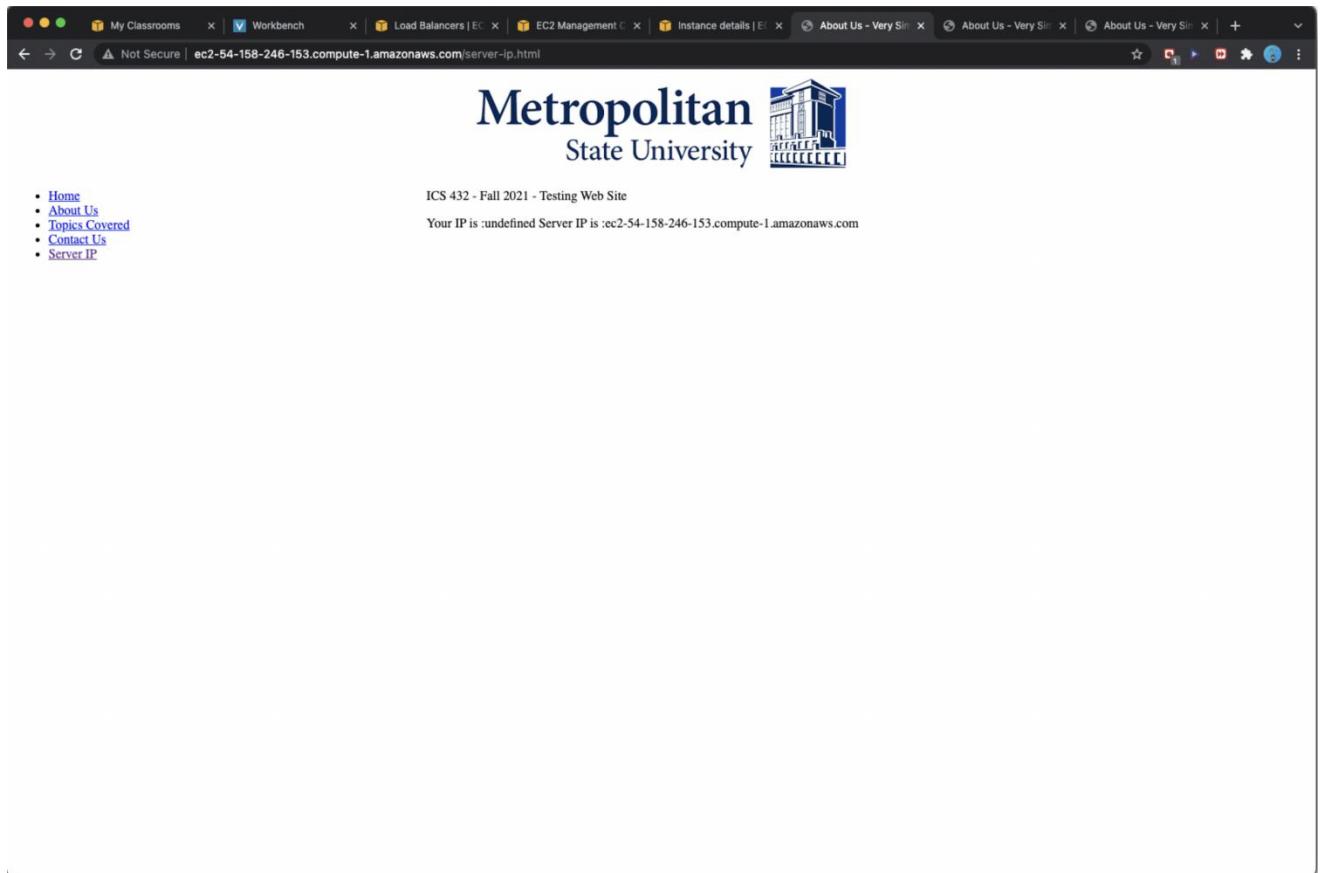
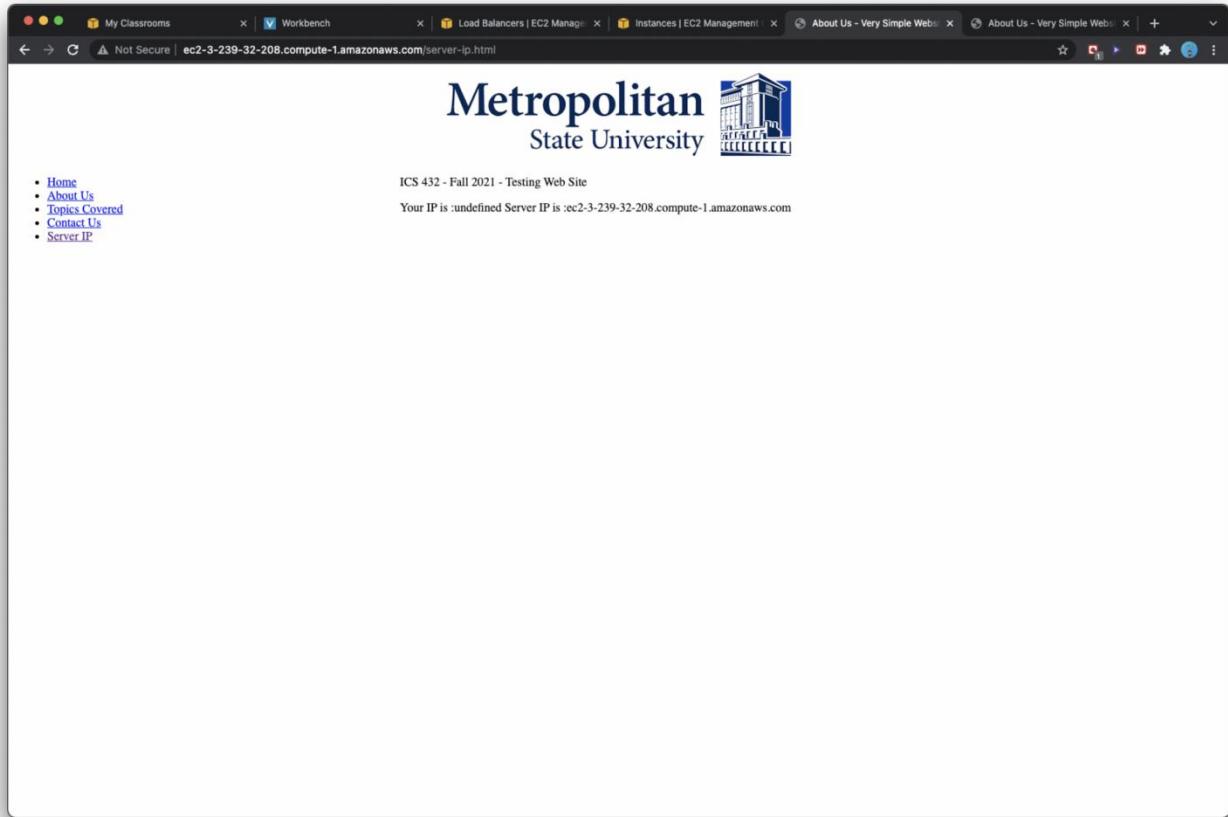
Metropolitan State University 

Navigation:

- Home
- About Us
- Topics Covered
- Contact Us
- Server IP

Content:

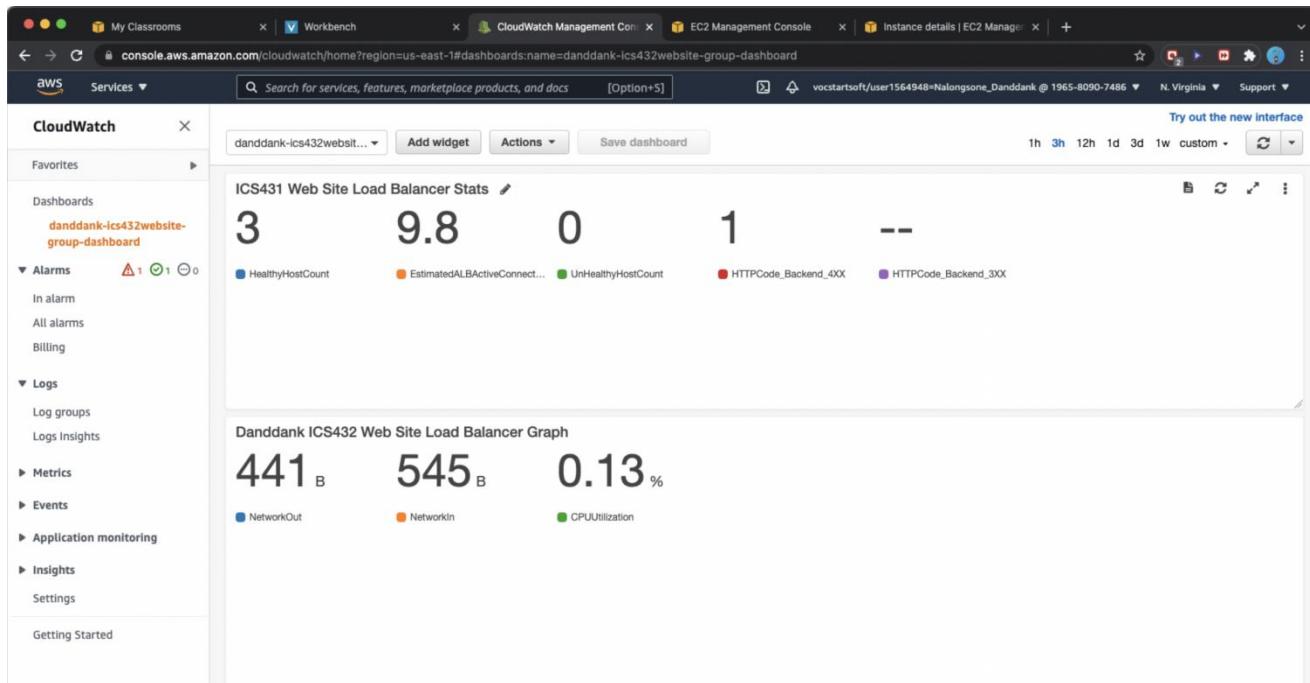
ICS 432 - Fall 2021 - Testing Web Site
Your IP is :undefined Server IP is :danddank-ics432web-load-balancer-572798612.us-east-1.elb.amazonaws.com



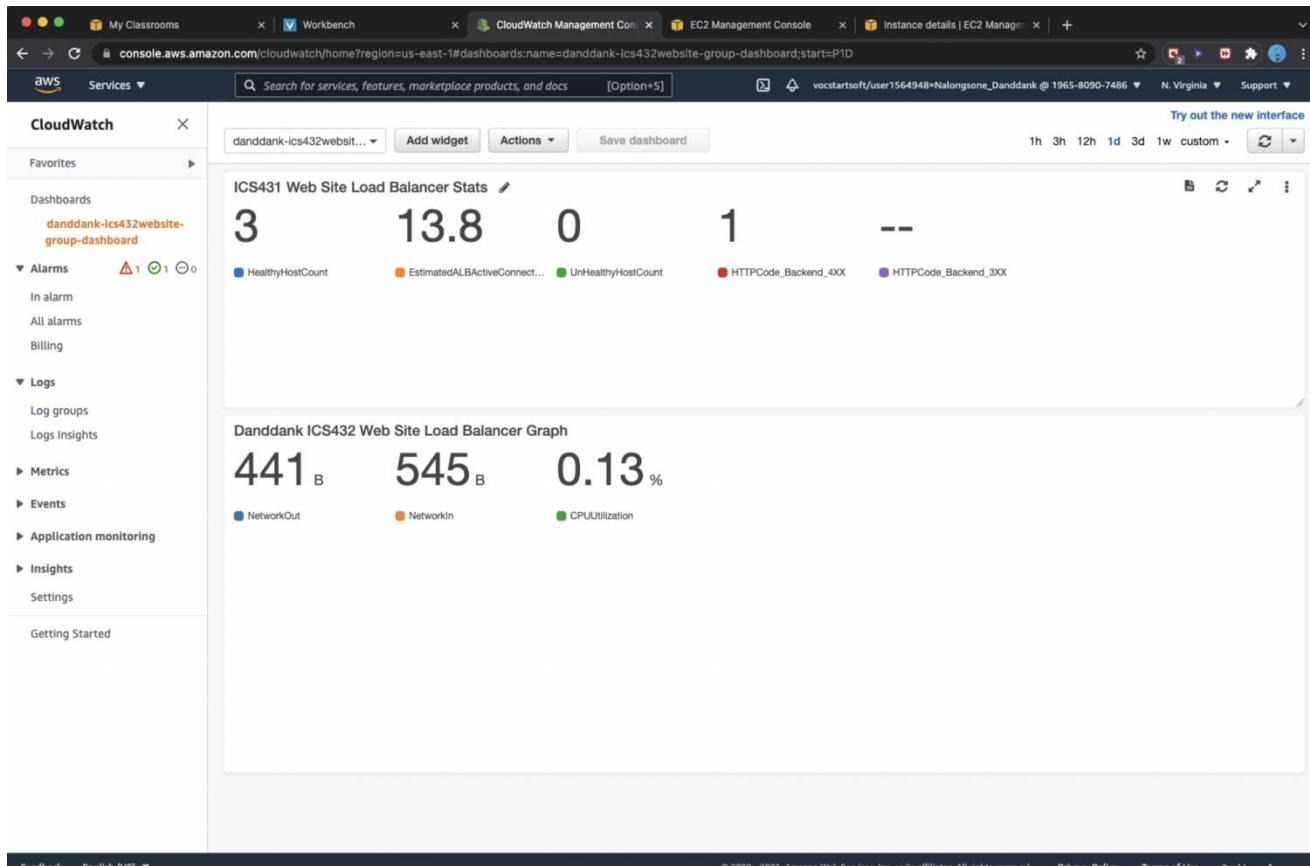
Homework report screen-shot #12:

Homework report screen-shot #13:

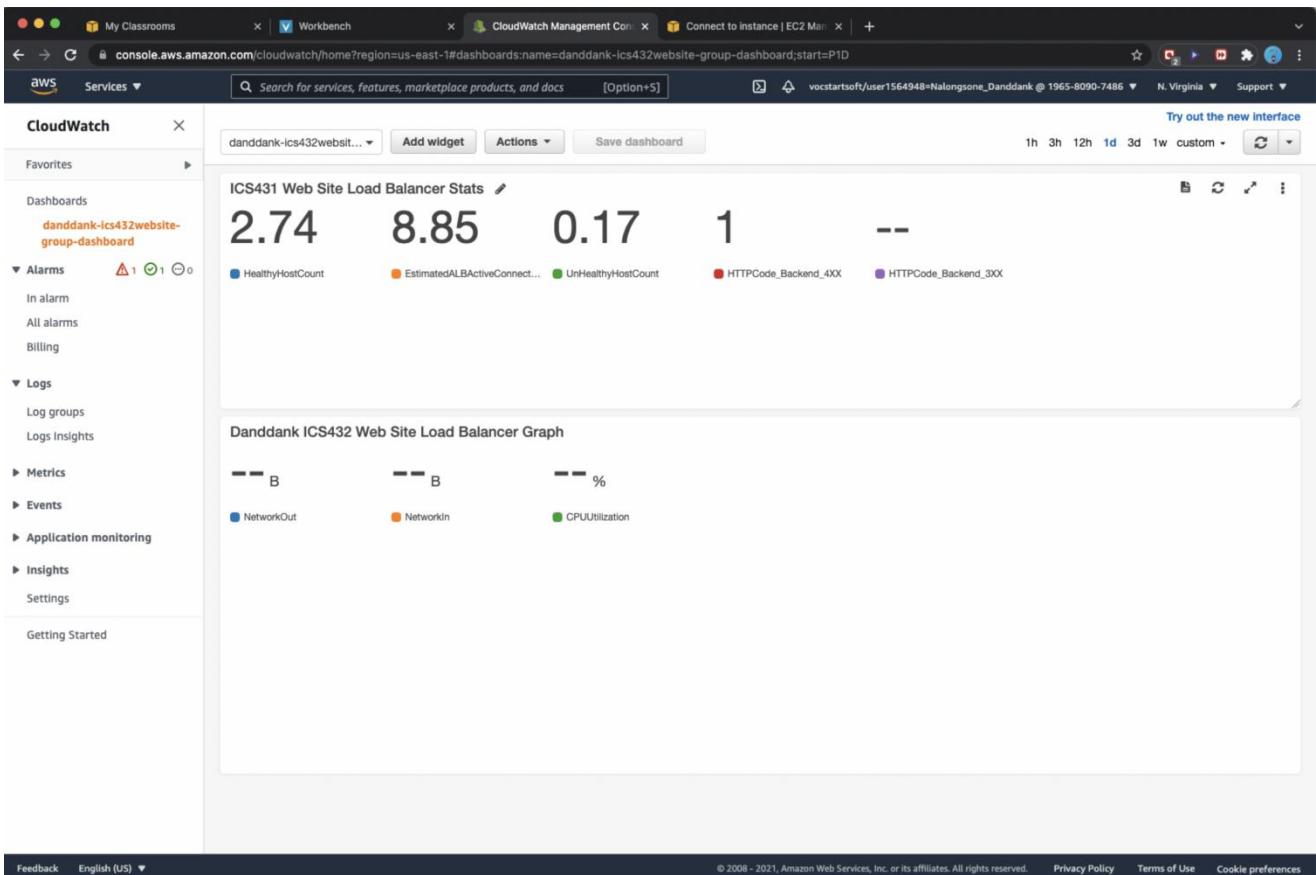
Homework report screen-shot #14:



Homework report screen-shot #15:



Homework report screen-shot #16:

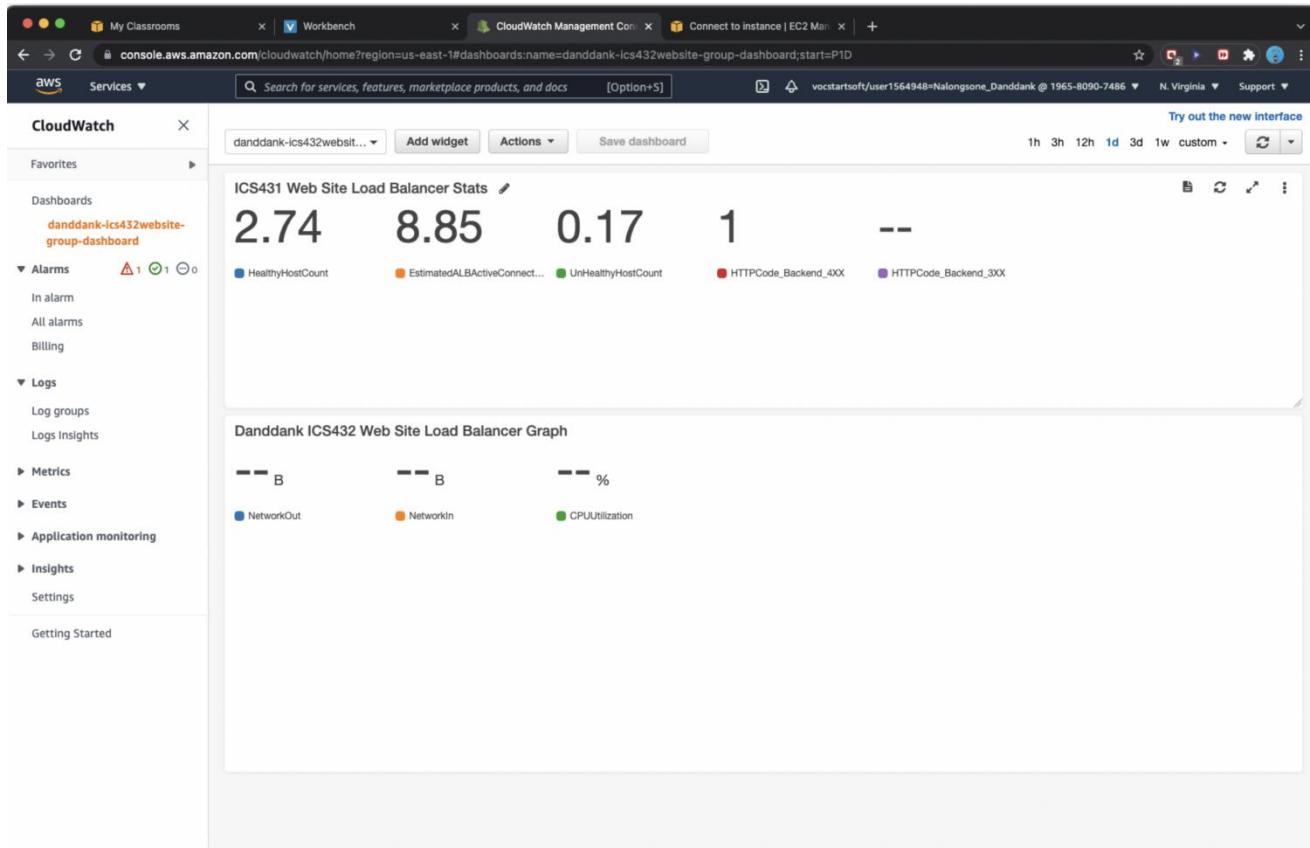


Homework report screen-shot #17:

Instances (10) <small>Info</small>								
		Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	Name	i-0fbaf4fb8b8774f270	Terminated	t3.micro	-	No alarms	+ us-east-1c	-
<input type="checkbox"/>	-	i-08a7450946567f46	Terminated	t3.micro	-	No alarms	+ us-east-1c	-
<input type="checkbox"/>	-	i-0f25f3559e1825aad	Running	t3.micro	Initializing	No alarms	+ us-east-1c	ec2-18-2
<input type="checkbox"/>	-	i-00962b3b2d82da2ce	Running	t3.micro	Initializing	No alarms	+ us-east-1c	ec2-44-1
<input type="checkbox"/>	Danddank Web Server	i-05a38511178bf90a0	Stopped	t2.micro	-	No alarms	+ us-east-1a	-
<input type="checkbox"/>	DanddankLab4Instance3	i-022df32823332594e	Stopped	t2.micro	-	No alarms	+ us-east-1a	-
<input type="checkbox"/>	danddank-ics432website-instance	i-0a4aff20a30b1214a	Stopped	t2.micro	-	No alarms	+ us-east-1a	-
<input type="checkbox"/>	nalongstone-ics432website-instance	i-04b45fe4b62fdfa45	Stopped	t2.micro	-	No alarms	+ us-east-1a	-
<input type="checkbox"/>	run-by-auto-scaling-group1	i-0460556cc3c152b65	Terminated	t3.micro	-	No alarms	+ us-east-1a	-
<input type="checkbox"/>	run-by-auto-scaling-group3	i-075fb8bf5cc6c569e	Running	t3.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-54-9

Select an instance above

Homework report screen-shot #18:



Summarize your learning:

I have learned to create load balance and auto scale to launch a EC2 instance of virtual machine in AWS, and learned how to deploy a web server to run in AWS in the Linux instance which I have created by auto scale group and load balance from AMIs and Instance template. then install a nginx web server software on the EC2 instance. Then, modified the security setting of the instance to allow public access for other people can access the website that I created. Finally, stop all the EC2 instance to terminate.

The most important thing is I know how to create the load balance and auto scale increase and decrease the number of instances following the requirement of traffic internet that is every significant concept of current technology and most popular use to deploy our modern website and other applications these day.

I think the most challenge me about working with the concept of how to balance the instances that created by auto scale and how to set up the parameters for them to let them to be better perform and effective. And another thing challenge me is working with the environment system, Linux, command line and system file of Linux, that is every challenge for me in the beginning to work quickly.

Exercise 2: Implementing a scalable GCP infrastructure to host a web site.

Homework report screen-shot #19:

Setting up a network load bal... | VM Instances – Compute Eng... | Home - ICS 432 testing Web... | +

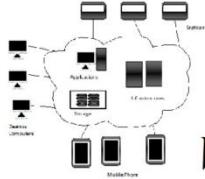
Not Secure | 34.71.199.139

Metropolitan
State University 

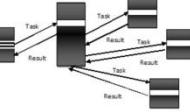
ICS 432 Distributed and Cloud Computing Fall 2021

This is a very simple web site for testing purposes.

Cloud Computing



Distributed Computing



VS

• Home
• About Us
• Topics Covered
• Contact Us
• Server IP

Setting up a network load bal... | VM Instances – Compute Eng... | About Us - Very Simple Web... | +

Not Secure | 34.71.199.139/server-ip.html

Metropolitan
State University 

ICS 432 - Fall 2021 - Testing Web Site

Your IP is : undefined Server IP is : 34.71.199.139

• Home
• About Us
• Topics Covered
• Contact Us
• Server IP

Homework report screen-shot #20:

Setting up a network load balancer

Images – Compute Engine

Google Cloud Platform dandank-hw2-loadbalancing

Search products and resources

Compute Engine

Images [CREATE IMAGE] REFRESH DELETE

An image is a replica of a disk that contains the applications and operating system needed to start a VM. You can create custom images or use public images pre-configured with Linux or Windows OSes. Learn more

IMAGES IMAGE IMPORT HISTORY IMAGE EXPORT HISTORY

Filter Enter property name or value

Status	Name	Location	Archive size	Disk size	Created by	Family	Creation time	Actions
<input type="checkbox"/>	dandank-ics432website-image	us-central1	1.08 GB	20 GB	dandank-hw2-loadbalancing		Oct 20, 2021, 10:18:54 PM UTC-05:00	⋮
<input type="checkbox"/>	c0-deeplearning-common-cpu-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	common-cpu-debian-10	Oct 11, 2021, 3:21:36 PM UTC-05:00	⋮
<input type="checkbox"/>	c0-deeplearning-common-cu110-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	common-di-gpu-debian-10	Oct 11, 2021, 3:52:39 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-1-15-cu110-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	tf-1-15-gpu-debian-10	Oct 11, 2021, 4:52:58 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-1-15-tpu-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	tf-1-15-tpu-debian-10	Oct 11, 2021, 4:33:33 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-2-1-cu110-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	tf2-2-1-gpu-debian-10	Oct 11, 2021, 5:11:59 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-2-1-tpu-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	tf2-2-1-tpu-debian-10	Oct 11, 2021, 4:57:35 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-2-3-cu110-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	tf2-2-3-gpu-debian-10	Oct 11, 2021, 5:36:22 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-2-3-tpu-v20211011-debian-10	asia, eu, us	—	50 GB	Debian	tf2-2-3-tpu-debian-10	Oct 11, 2021, 5:11:41 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-2-4-cu110-v20210512-debian-10	asia, eu, us	—	50 GB	Debian	tf2-2-4-gpu-debian-10	May 12, 2021, 6:54:01 PM UTC-05:00	⋮
<input type="checkbox"/>	c1-deeplearning-tf-2-4	asia, eu, us	—	50 GB	Debian	tf-2-4-tpu	May 12, 2021, 5:04:25	⋮

Show deprecated images

Select an image

PERMISSIONS LABELS

Please select at least one resource.

Homework report screen-shot #21:

Setting up a network load balancer

VM instances – Compute Engine

Home - ICS 432 testing Web

Not Secure | 35.222.179.200

Metropolitan State University

ICS 432 Distributed and Cloud Computing Fall 2021

This is a very simple web site for testing purposes.

Cloud Computing

VS

Distributed Computing

Setting up a network load bal... | VM Instances – Compute Engine | Home - ICS 432 testing Web | 34.135.64.188/index.html

Not Secure | 34.135.64.188/index.html

Metropolitan State University



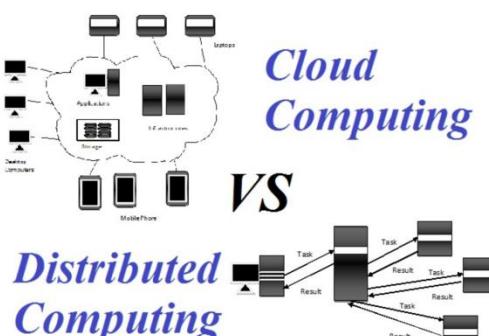
ICS 432 Distributed and Cloud Computing Fall 2021

This is a very simple web site for testing purposes.

Cloud Computing

VS

Distributed Computing

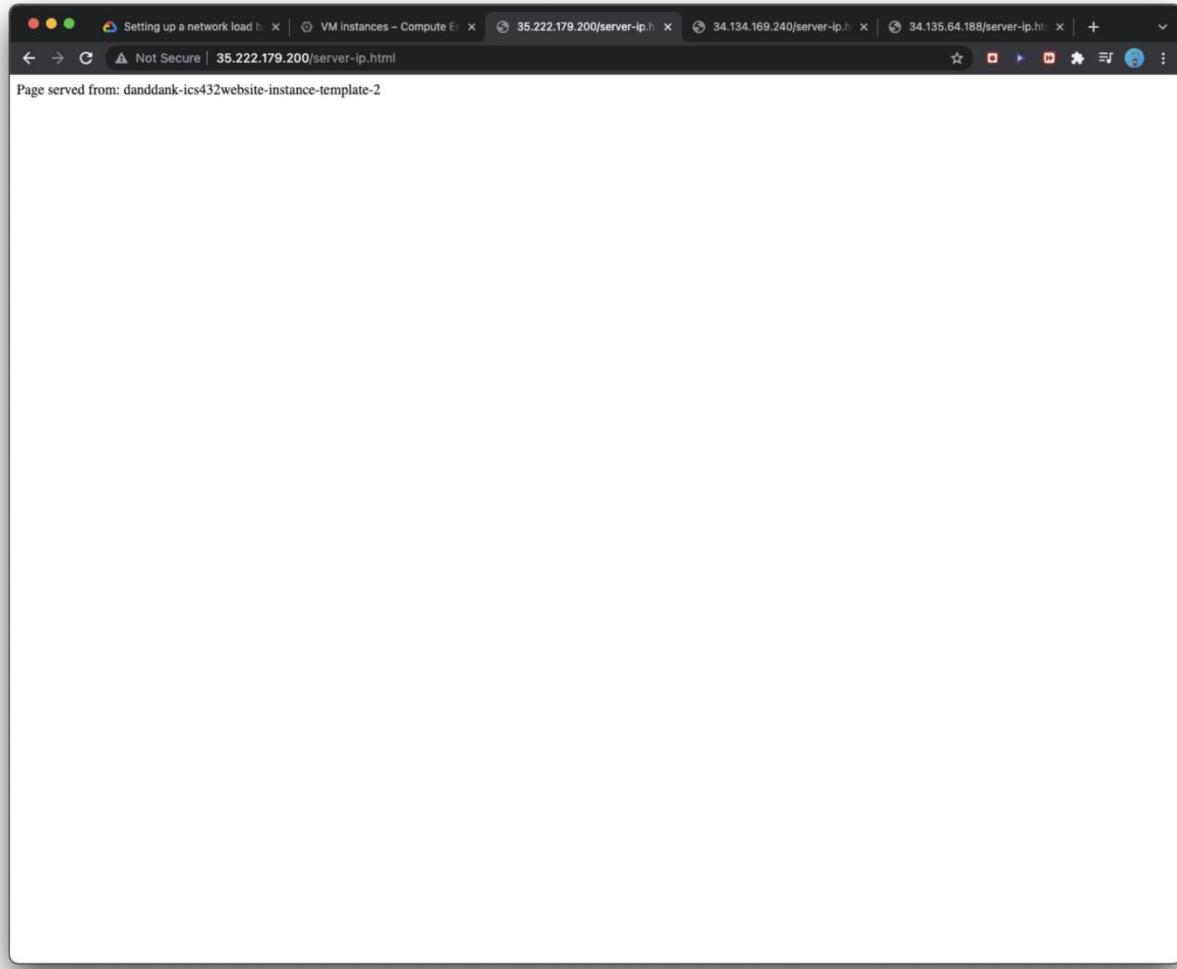


Homework report screen-shot #22:

Setting up a network load bal... | VM Instances – Compute Engine | 34.135.64.188/server-ip.html | 34.135.64.188/server-ip.html

Not Secure | 34.135.64.188/server-ip.html

Page served from: dandank-ics432website-instance-template-3



Homework report screen-shot #23:

External IP address '34.135.64.188' copied.

Status	Name	Creation Time	Template	Per instance config	Internal IP	External IP	Health	Connect
Ready	danddank-ics432website-instance-group-dkbr	Oct 21, 2021, 2:22:02 AM UTC-05:00	danddank-ics432website-instance-template		10.128.0.9 (nic0)	34.135.64.188	SSH	⋮
Ready	danddank-ics432website-instance-group-z0s4	Oct 21, 2021, 2:22:02 AM UTC-05:00	danddank-ics432website-instance-template		10.128.0.8 (nic0)	34.136.60.170	SSH	⋮

Homework report screen-shot #24:

The screenshot shows the Google Cloud Platform Compute Engine Instance Groups page. The left sidebar is collapsed. The main area displays the 'danddank-ics432website-instrnace-group' instance group. The 'OVERVIEW' tab is selected. Key details shown include:

- Instances by status:** 3 instances (all Ready)
- Instance health:** Not configured
- Autoscaling:** On (min 3, max 4)
- Status:** Ready
- Creation Time:** Oct 21, 2021, 2:21:56 AM UTC-05:00
- Description:** (empty)
- Number of instances:** 3
- Template:** danddank-ics432website-instance-template
- Location:** us-central1-a

The 'Instance Group Members' section lists three instances:

Status	Name	Creation Time	Template	Per instance config	Internal IP	External IP	Health	Connect
Ready	danddank-ics432website-instrnace-group-dkbr	Oct 21, 2021, 2:22:02 AM UTC-05:00	danddank-ics432website-instance-template		10.128.0.9 (nic0)	34.135.64.188	SSH	▼
Ready	danddank-ics432website-instrnace-group-sr6x	Oct 21, 2021, 2:26:16 AM UTC-05:00	danddank-ics432website-instance-template		10.128.0.10 (nic0)	34.68.156.134	SSH	▼
Ready	danddank-ics432website-instrnace-group-z0s4	Oct 21, 2021, 2:22:02 AM UTC-05:00	danddank-ics432website-instance-template		10.128.0.8 (nic0)	34.136.60.170	SSH	▼

Homework report screen-shot #25:

The screenshot shows the Google Cloud Platform Firewall rules page. The left sidebar is collapsed. A modal window titled 'Get real-time analytics with Network Intelligence Center' is open, providing instructions on how to use the feature.

The main table lists firewall rules:

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs	Hit count	Last hit	Insights
default-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80	Allow	1000	default	Off	—	—	
default-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	default	Off	—	—	
fw-allow-health-check	Ingress	allow-health-check	IP ranges: 130.211.0.0/23, 35.191.0.0/16	tcp:80	Allow	1000	default	Off	—	—	
default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default	Off	—	—	
default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default	Off	—	—	
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default	Off	—	—	
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default	Off	—	—	

A success message at the bottom right states: 'Successfully created firewall rule "fw-allow-health-check".'

Homework report screen-shot #26:

The screenshot shows the Google Cloud Platform VPC network External IP addresses page. A success message at the bottom states "Successfully created address 'lb-ipv4-ics432website'".

Name	External Address	Region	Type	Version	In use by	Network Tier	Labels
lb-ipv4-ics432website	34.149.58.133		Static	IPv4	VM instance danddank-ics432website-instance-group-8tq (Zone us-central1-a)	Premium	CHANGE
-	35.184.239.81	us-central1	Ephemeral	IPv4	VM instance danddank-ics432website-instance-group-6b6q (Zone us-central1-a)	RESERVE	
-	35.202.19.160	us-central1	Ephemeral	IPv4	VM instance danddank-ics432website-instance-group-7qlc (Zone us-central1-a)	RESERVE	
-	35.232.57.174	us-central1	Ephemeral	IPv4	VM instance danddank-ics432website-instance-group-8tg5 (Zone us-central1-a)	RESERVE	
-	35.238.104.95	us-central1	Ephemeral	IPv4	VM instance danddank-ics432website-instance-group-8tg5 (Zone us-central1-a)	RESERVE	

Homework report screen-shot #27:

The screenshot shows the Google Cloud Platform Network services Load balancer details page for the load balancer "danddank-ics432website-load-balancer".

Frontend

Protocol	IP:Port	Certificate	SSL Policy	Network Tier
HTTP	34.120.169.146:80	-	-	Premium

Host and path rules

Hosts	Paths	Backend
All unmatched (default)	All unmatched (default)	danddank-ics432website-backend-service

Backend

Backend services

1. **danddank-ics432website-backend-service**

Endpoint protocol	Named port	Timeout	Health check	Cloud CDN
HTTP	http	30 seconds	http-basic-check	Disabled

ADVANCED CONFIGURATIONS

Name	Type	Zone	Healthy	Autoscaling	Balancing mode	Selected ports	Capacity
danddank-ics432website-instance-group	Instance group	us-central1-a	3 of 3	On: Target CPU utilization 60%	Max backend utilization: 80%	80	100%

Homework report screen-shot #28:

Setting up a network load bal... | Load balancer details – Network... | Home - ICS 432 testing Web site | +

Not Secure | 34.120.169.146

Metropolitan State University

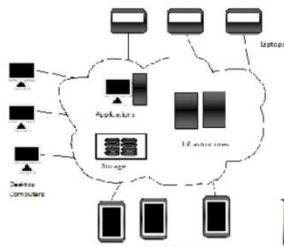


ICS 432 Distributed and Cloud Computing Fall 2021

This is a very simple web site for testing purposes.

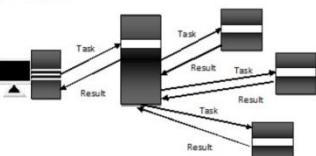
- [Home](#)
- [About Us](#)
- [Topics Covered](#)
- [Contact Us](#)
- [Server IP](#)

Cloud Computing



VS

Distributed Computing



Homework report screen-shot #29:

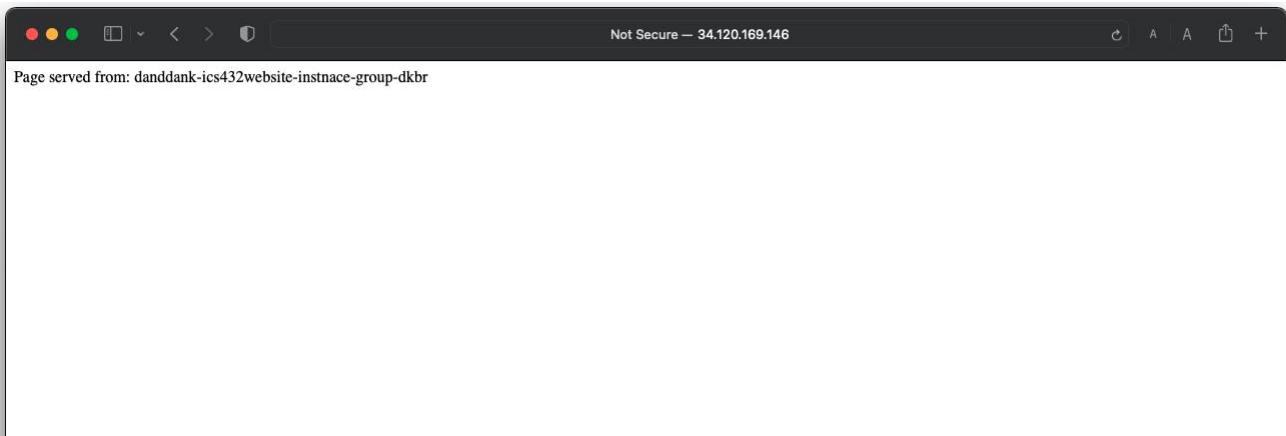
Setting up a network load bal... | Load balancer details – Network... | 34.120.169.146/server-ip.html | +

Not Secure | 34.120.169.146/server-ip.html

Page served from: danddank-ics432website-instance-group-z0s4

Not Secure — 34.120.169.146

Page served from: danddank-ics432website-instance-group-dkbr



Summarize your learning:

Purpose	AWS	GCP
Running compute resources on the cloud	Compute -> Amazon EC2	Compute -> Compute Engine -> VM instances
Creating Instance template	Compute -> Amazon EC2 -> Auto Scaling -> Launch Configurations	Compute -> Compute Engine -> Instances templates
Create Load Balancers	Compute -> Amazon EC2 -> Load Balancing -> Load Balancers	Networking -> Network services -> Load Balancing
Create Security Groups	Compute -> Amazon EC2 -> Network & Security -> Security Groups	IAM & Admin ->
Create Instance Images	Compute -> Amazon EC2 -> Images -> AMIs	Compute -> Compute Engine -> Storage -> Images
Stop compute resources on the cloud	Compute -> Amazon EC2	Compute -> Compute Engine -> VM instances
Building Monitoring Application Checking	Management & Governance -> CloudWatch	Operations -> Monitoring

Assignment #3 : MapReduce

Exercise 1: Analyzing Stock Data.

Task 1.1:

1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some duplicate of the key that should be the value. Then, implement python code mapper to print out all of key, stock_symbol and follow by 1. finally, implement python code for reduce that combine all value that are same key, and print out the key with total counting number.

For example: output key and value of reduce, GA 12, GAB 75, GAH 18, GAI 34, GAJ 27...

2-

```
❸ mapper1.py > ...
1  #!/usr/bin/env python
2
3  import sys
4
5  for line in sys.stdin:
6      line = line.strip()
7      words = line.split(',')
8      print("%s\t%s" % (words[1], 1))
9
```

3-

```
❸ reducer1.py > ...
1  #!/usr/bin/env python
2
3  import sys
4
5  current_word = None
6  current_count = 0
7  word = None
8
9  for line in sys.stdin:
10     line = line.strip()
11     (word, count) = line.split('\t')
12     count = int(count)
13     if current_word == word:
14         current_count += 1
15     else:
16         if current_word:
17             print(current_word, current_count)
18         current_count = count
19         current_word = word
20
21     if current_word == word:
22         print(current_word, current_count)
23
```

4-

```

GYC 1
GYC 1
GYC 1
GYC 1
GYC 1
GYC 1
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper1.py | sort
| python3 reducer1.py
GA 12
GAB 75
GAH 18
GAI 34
GAJ 27
GAM 88
GAP 76
GAR 1
GAS 70
GAT 6
GB 28
GBB 9
GBE 20
GBF 10
GBL 41
GBX 55
GCA 12
GCF 8
GCH 54
GCI 54
GCO 88
GCS 14
GCV 43
GD 100
GDF 48
GDI 55
GDL 8
GDO 1
GDP 71
GDV 15
GE 155
GEA 5
GEC 22
GED 22
GEF 46
GEG 12
GEJ 5
GEO 55
GEP 14
GER 11
GES 45
GET 61
GEX 7
GF 64
GFA 8
GFF 54
GFI 52
GFW 25
GFY 16
GEZ 25

```

Task 1.2:

1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some the same key that should be the value. Then, implement python code mapper to print out all of key, stock_symbol and follow by its value, stock_price_high. finally, implement python code for reduce that compare all value that are same key to get the highest, and print out the key and the highest stock price.

For example: output key and value of reduce, GA 13.35, GAB 11.99, GAH 25.3, GAI 14.62, GAJ 25.79...

2-

```

...
  ↗ mapper2.py ✘ × ↗ reducer2.py ✘
  ↗ mapper2.py > ...
U 1  #!/usr/bin/env python
U 2
U 3  import sys
U 4
U 5  for line in sys.stdin:
U 6  |   line = line.strip()
U 7  |   words = line.split(',')
U 8  |   print("%s\t%s" % (words[1], words[4]))
U 9

```

3-

```
... mapper2.py U reducer2.py U X
    reducer2.py > ...
U 1 #!/usr/bin/env python
U 2
U 3 import sys
U 4
U 5 current_word = None
U 6 current_max = float('-inf')
U 7 word = None
U 8
U 9 for line in sys.stdin:
U 10     line = line.strip()
U 11     (word, price) = line.split('\t')
U 12     price = float(price)
U 13     if current_word == word:
U 14         current_max = max(current_max, price)
U 15     else:
U 16         if current_word:
U 17             print(current_word, current_max)
U 18         current_max = price
U 19         current_word = word
U 20
U 21     if current_word == word:
U 22         print([current_word, current_max])
U 23
```

4-

```
... mapper2.py U reducer2.py U X
    reducer2.py > ...
U 1 #!/usr/bin/env python
U 2
U 3 import sys
U 4
U 5 GYC 22.07
U 6 GYC 22.39
U 7 GYC 22.72
U 8 GYC 23.4
U 9 GYC 23.74
U 10 (base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper2.py | sort | p
U 11 ython3 reducer2.py
GA 13.35
GAB 11.99
GAH 25.3
GAI 14.62
GAJ 25.79
GAM 43.36
GAP 44.48
GAR 26.11
GAS 51.54
GAT 29.53
GB 44.6
GBB 52.36
GBE 14.05
GBF 187.06
GBL 60.73
GBX 40.45
CCA 18.0
GCF 19.23
GCH 25.97
GCI 87.6
GOO 53.26
GCS 17.57
GOV 11.2
GD 116.47
GDF 15.48
GDI 56.55
GDL 19.99
GDO 20.02
GDP 62.35
GDV 22.27
GE 160.0
GEA 25.3
GEC 26.5
GED 25.76
GEF 125.49
GEG 25.54
GEJ 24.7
GEO 45.6
GEP 24.2
GER 26.21
GES 81.82
GET 56.72
GEX 56.73
GF 17.65
GFA 37.17
GFF 26.78
GFI 21.58
GFW 25.89
GFY 19.16
GFZ 25.8
GG 44.16
GGB 51.05
GGC 50.41
GGG 48.84
```

Task 1.3:

1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some the same key and its price and its volume . Then, implement python code mapper to print out all of key, stock_symbol and follow by its values, stock_price_high and its volume greater than 250,000. finally, implement python code for reduce that compare all value that are same key to get the highest, and print out the key and the highest stock price and volume>250000.

For example: output key and value of reduce, GA 13.35 299100, GAB 10.6 284400, GAI 4.75 256400, GAP 32.9 371200, GAS 51.54 421800...

2-

```
... mapper3.py U × reducer3.py U NYSE.csv U
  ↗ mapper3.py > ...
U 1 #!/usr/bin/env python
U 2
U 3 import sys
U 4
U 5 for line in sys.stdin:
U 6     line = line.strip()
U 7     words = line.split(',')
U 8     if int(words[7]) > 250000:
U 9         print("%s\t%s\t%s" % (words[1], words[4], words[7]))
U 10
```

3-

```
... mapper3.py U reducer3.py U NYSE.csv U
  ↗ reducer3.py > ...
U 1 #!/usr/bin/env python
U 2
U 3 import sys
U 4
U 5 current_word = None
U 6 current_max = float('-inf')
U 7 word = None
U 8
U 9 for line in sys.stdin:
U 10    line = line.strip()
U 11    (word, price, volume) = line.split('\t')
U 12    price = float(price)
U 13    if current_word == word:
U 14        current_max = max(current_max, price)
U 15    else:
U 16        if current_word:
U 17            print(current_word, current_max, volume)
U 18        current_max = price
U 19        current_word = word
U 20
U 21    if current_word == word:
U 22        print(current_word, current_max, volume)
U 23
```

4-

```

... mapper3.py U reducer3.py U NYSE.csv U
U reducer3.py > ...
U 12     price = float(price)
U
U PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
U
U GY    19.09  576000
U GY    19.89  352900
U GY    2.16   599400
U GY    20.1   553500
U GY    36.43  658200
U GY    7.33   410500
U GY    7.9    337400
U GY    8.43   857000
U GY    8.64   498700
U (base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper3.py | sort | python3 reducer3.py
U GA 13.35 299100
U GAB 10.6 284400
U GAI 4.75 256400
U GAP 32.9 371200
U GAS 51.54 421800
U GB 44.6 324000
U GBE 13.07 273400
U GBX 37.4 395000
U GCA 18.0 270500
U GCH 25.97 4030100
U GCI 87.6 251100
U GCO 51.04 404900
U GCS 16.34 922400
U GD 116.47 292800
U GDI 56.55 289300
U GDP 62.35 627300
U GDV 21.58 3840000
U GE 160.0 252100
U GEC 25.6 336600
U GEF 125.49 376400
U GEO 45.6 2618900
U GES 81.82 629100
U GET 55.07 331400
U GEX 31.48 1467700
U GF 11.56 261800
U GFA 37.17 370400
U GFF 23.92 4692100
U GFI 21.58 1239200
U GG 44.16 11267200
U GGG 51.05 905400
U GGG 45.69 330700
U GHI 8.0 740700
U GHL 81.0 324600
U GIB 45.25 1606000
U GIL 60.65 272000
U GIM 9.98 3038400
U GIS 83.25 520200
U GKK 29.0 2464500
U GLD 90.96 270400
U GLF 64.58 619700
U GLG 11.26 341900
U GLS 17.34 360500
U GLT 14.49 2418500
U GLW 253.0 1348600
U GME 58.41 488200
U GMR 47.74 366800
U GMT 62.81 496000
U GMXR 47.48 254500
U GNA 14.94 271500
U GNK 83.37 361800
U GNV 1.48 13621500
U GNW 35.15 2671800
U GOL 33.86 365300
U GOM 23.1 291400
U GOV 23.8 255500

```

Task 1.4:

1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some the same keys, stock and date, and its price. Then, implement python code mapper to print out all of keys, stock_symbol with year which extract by using datetime, and follow by its values, stock_price_high. finally, implement python code for reduce that compare all price that are same keys, stock and year to get the highest price, and print out each stock and each year with the highest stock price.

For example: output key and value of reduce, GYC 2019 18.15, GYC 2018 20.6, GYB 2020 21.41, GY 2019 8.43, GY 2018 8.64...

2-

```
... mapper4.py U X reducer4.py U NYSE.csv U
.mapper4.py > ...
1 #!/usr/bin/env python
2
3 import sys
4 from datetime import datetime
5
6 for line in sys.stdin:
7     line = line.strip()
8     words = line.split(',')
9     date = datetime.strptime(words[2], "%m/%d/%Y")
10    year = date.strftime('%Y')
11    print("%s\t%s\t%s" % (words[1], year, words[4]))
12
```

3-

```
... mapper4.py U reducer4.py U NYSE.csv U
.reducer4.py > ...
1 #!/usr/bin/env python
2
3 import sys
4
5 current_word = None
6 current_year = None
7 price_max = float('-inf')
8 word = None
9 year = None
10
11 for line in sys.stdin:
12     line = line.strip()
13     (word, year, price) = line.split('\t')
14     price = float(price)
15     if current_word == word and current_year == year:
16         current_max = max(current_max, price)
17     else:
18         if current_word:
19             print(current_word, current_year, current_max)
20         current_max = price
21         current_word = word
22         current_year = year
23
24 if current_word == word and current_year == year:
25     print(current_word, current_year, current_max)
26
```

4-

```

... mapper4.py U reducer4.py U NYSE.csv U
U reducer4.py > ...
U 1 #!/usr/bin/env python
U 2
U PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
U zsh + □
U GWW 2013 45.01
U GWW 2014 55.43
U GWW 2015 62.48
U GWW 2016 75.25
U GWW 2017 87.65
U GWW 2018 91.97
U GWW 2019 72.01
U GXP 1998 29.37
U GXP 1999 35.88
U GXP 2000 34.5
U GXP 2001 37.38
U GXP 2002 24.37
U GXP 2003 26.0
U GXP 2004 22.37
U GXP 2005 25.5
U GXP 2006 27.87
U GXP 2007 29.12
U GXP 2009 28.19
U GXP 2010 26.69
U GXP 2011 24.81
U GXP 2012 22.0
U GXP 2014 34.19
U GXP 2015 29.98
U GXP 2016 32.35
U GXP 2017 32.05
U GXP 2018 23.22
U GXP 2019 16.04
U GY 1997 36.43
U GY 1999 6.06
U GY 2000 4.39
U GY 2001 3.62
U GY 2002 6.03
U GY 2003 7.09
U GY 2004 6.21
U GY 2005 6.29
U GY 2006 8.25
U GY 2007 14.83
U GY 2008 15.31
U GY 2009 12.97
U GY 2010 9.88
U GY 2011 13.87
U GY 2012 10.23
U GY 2013 10.09
U GY 2014 17.34
U GY 2015 20.1
U GY 2016 19.89
U GY 2017 13.72
U GY 2018 8.64
U GY 2019 8.43
U GYA 2014 22.85
U GYA 2015 24.78
U GYA 2016 24.5
U GYA 2017 23.31
U GYA 2020 22.2
U GYB 2014 25.0
U GYB 2017 24.08
U GYB 2018 19.66
U GYB 2019 16.4
U GYB 2020 21.41
U GYC 2016 22.39
U GYC 2017 23.74
U GYC 2018 20.6
U GYC 2019 18.15
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper4.py | sort | python3 reducer4.py

```

Task 1.5:

1- First do the Job 1 by run the CSV file with mapper1.py and reducer1.py, then go to do Job 2 by taking the output to run the mapper5.py and reducer5.py which implement to count all number that output from Job 1 to find total number distinct stock symbols.

Output: 1 154

2-

mapper1.py

```

... reducer5.py U mapper5.py U mapper1.py U
U reducer5.py > ...
U 1 #!/usr/bin/env python
U 2
U 3 import sys
U 4
U 5 for line in sys.stdin:
U 6     line = line.strip()
U 7     words = line.split('\t')
U 8     print("%s\t%s" % (words[1], 1))
U 9

```

mapper5.py

```

...
 reducer5.py U  mapper5.py U  mappert1.py U
 reducer5.py > ...
 1 #!/usr/bin/env python
 2
 3 import sys
 4
 5 for line in sys.stdin:
 6     print("%s\t%s" % (1, 1))
 7

```

3-

reducer1.py

```

...
 reducer5.py U  mapper5.py U  reducer1.py U
 reducer1.py > ...
 1 #!/usr/bin/env python
 2
 3 import sys
 4
 5 current_word = None
 6 current_count = 0
 7 word = None
 8
 9 for line in sys.stdin:
10     line = line.strip()
11     (word, count) = line.split('\t')
12     count = int(count)
13     if current_word == word:
14         current_count += 1
15     else:
16         if current_word:
17             print(current_word, current_count)
18         current_count = count
19         current_word = word
20
21 if current_word == word:
22     print(current_word, current_count)
23

```

reducer5.py

```

...
 reducer5.py U  mapper5.py U  reducer1.py U
 reducer5.py > ...
 1 #!/usr/bin/env python
 2
 3 import sys
 4
 5 key_ = "1"
 6 count_ = 0
 7
 8 for line in sys.stdin:
 9     line = line.strip()
10     (key, value) = line.split('\t')
11     count = int(value)
12     count_ += count
13
14 print(key_, count_)
15

```

4-

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
zsh + × ^ ×
1 1
1 1
1 1
1 1
1 1
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper1.py | sort | python3 reducer1.py | python3 mapper5.py | python3 reducer5.py
3 154
(base) ping58972@Nalongsones-MacBook-Air assignment3-files %

```

Exercise 2: Implementing Relational Union and Intersection using mapreduce.

Task 2.1:

1- Print out some line for looking for some product ID in both the txt file and find out some duplicate of the ID. Then, implement python code mapper to print out and sort all of both file each line. finally, implement python code for reduce ignore the same product ID, and print out the unique product ID line with their describe and price.

For example output:

```

1001 Zip Bag 100
1002 Harness 150
1003 Full Charger 125
1004 Big Helmet 40
1009 Small Helmet 40

```

2001 Stove 80
2002 Soft Boot 70
2003 Soft-L Jacket 35
2004 Strongster Harness 20
2008 Boot 70
2009 Umbrella 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
3004 Big Tire 30
3008 Small Tire 30
4004 Hard Boot 90
4009 Stand 90
5005 Tent 150
6006 Hi-Tent 250
7007 Tech GPS 300
8008 Pedals 20
9009 Reli-Rope 302-

The screenshot shows a code editor with two tabs open: 'mapper2_1.py' and 'reducer2_1.py'. The 'mapper2_1.py' tab is active, displaying the following Python code:

```
#!/usr/bin/env python
import sys
for line in sys.stdin:
    line = line.strip()
    words = line.split(',')
    print("%s\t%s\t%s" % (words[0], words[1], words[2]))
```

3-

The screenshot shows a code editor with the 'reducer2_1.py' tab active, displaying the following Python code:

```
#!/usr/bin/env python
import sys
current_id = None
id = None
for line in sys.stdin:
    line = line.strip()
    (id, desc, price) = line.split('\t')
    if current_id == id:
        continue
    else:
        current_id = id
        print(id, desc, price)
```

4-

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
9009,Reli-Rope,30
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat store1.txt store2.txt | python3 mapper2_1.py | sort
1001 Zip Bag 100
1001 Zip Bag 100
1002 Harness 150
1002 Harness 150
1003 Full Charger 125
1003 Full Charger 125
1004 Big Helmet 40
1004 Big Helmet 40
1009 Small Helmet 40
2001 Stove 80
2001 Stove 80
2002 Soft Boot 70
2002 Soft Boot 70
2003 Soft-L Jacket 35
2004 Strongster Harness 20
2008 Boot 70
2008 Boot 70
2009 Umbrella 70
3001 Pad 25
3001 Pad 25
3002 Knife 60
3002 Knife 60
3003 Soft Sock 15
3003 Soft Sock 15
3004 Big Tire 30
3008 Small Tire 30
4004 Hard Boot 90
4004 Hard Boot 90
4009 Stand 90
5005 Tent 150
6006 Hi-Tent 250
6006 Hi-Tent 250
7007 Tech GPS 300
8008 Pedals 20
9009 Reli-Rope 30
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat store1.txt store2.txt | python3 mapper2_1.py | sort | python3 reducer2_1.py
1001 Zip Bag 100
1002 Harness 150
1003 Full Charger 125
1004 Big Helmet 40
1009 Small Helmet 40
2001 Stove 80
2002 Soft Boot 70
2003 Soft-L Jacket 35
2004 Strongster Harness 20
2008 Boot 70
2009 Umbrella 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
3004 Big Tire 30
3008 Small Tire 30
4004 Hard Boot 90
4009 Stand 90
5005 Tent 150
6006 Hi-Tent 250
7007 Tech GPS 300
8008 Pedals 20
9009 Reli-Rope 30
(base) ping58972@Nalongsones-MacBook-Air assignment3-files %

```

Task 2.2:

1- Print out some line for looking for some product ID in both the txt file and find out some duplicate of the ID. Then, implement python code mapper to print out and sort all of both file each line. finally, implement python code for reduce ignore the product ID which only belong in just one file, and print out the unique product ID which belong to both files with their describe and price.

For example output:

```

1001 Zip Bag 100
1002 Harness 150
1003 Full Charger 125
1004 Big Helmet 40
2001 Stove 80
2002 Soft Boot 70
2008 Boot 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
4004 Hard Boot 90
5005 Tent 150
6006 Hi-Tent 250
7007 Tech GPS 300
8008 Pedals 20
9009 Reli-Rope 30

```

2-

A screenshot of a code editor showing two files: mapper2_2.py and reducer2_2.py. The mapper2_2.py file contains the following code:

```
1 #!/usr/bin/env python
2
3 import sys
4
5 for line in sys.stdin:
6     line = line.strip()
7     words = line.split(',')
8     print("%s\t%s\t%s" % (words[0], words[1], words[2]))
```

3-

A screenshot of a code editor showing the reducer2_2.py file. The code is as follows:

```
1 #!/usr/bin/env python
2
3 import sys
4
5 current_id = None
6 id = None
7
8 for line in sys.stdin:
9     line = line.strip()
10    (id, desc, price) = line.split('\t')
11    if current_id == id:
12        print(id, desc, price)
13    else:
14        current_id = id
```

4-

A screenshot of a terminal window showing the execution of the scripts. The output is as follows:

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
3008 Small Tire 30
4004 Hard Boot 90
4004 Hard Boot 90
4009 Stand 90
5005 Tent 150
6006 Hi-Tent 250
6006 Hi-Tent 250
7007 Tech GPS 300
8008 Pedals 20
9009 Reli-Rope 30
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat store1.txt store2.txt | python3 mapper2_2.py | sort | python3 reducer2_2.py
1001 Zip Bag 100
1002 Harness 150
1003 Full Charger 125
1004 Big Helmet 40
2001 Stove 80
2002 Soft Boot 70
2008 Boot 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
4004 Hard Boot 90
6006 Hi-Tent 250
(base) ping58972@Nalongsones-MacBook-Air assignment3-files %
```

Docker Lab : Docker in AWS and GCP

Exercise 1: Docker in AWS with Python Flask

Lab report screen-shot #1:

The screenshot shows the AWS Cloud9 IDE interface. On the left, there's a file tree for a project named 'DockerLab - Home'. The 'simple-webapp' directory contains an 'app.py' file and a 'Dockerfile'. The 'app.py' file contains the following code:

```
1 import os
2 from flask import Flask
3 app = Flask(__name__)
4
5 @app.route("/")
6 def main():
7     return "Welcome!"
8
9 @app.route("/how are you")
10 def hello():
11     return "I am good, how about you?"
12
13 if __name__ == "__main__":
14     app.run(host="0.0.0.0", port=8080)
15
```

The 'Dockerfile' contains:

```
1 FROM python:3.8-slim
2 WORKDIR /app
3 COPY . /app
4 EXPOSE 8080
5 ENTRYPOINT ["python", "app.py"]
```

In the bottom terminal window, the user runs commands to build and run a Docker container:

```
vocstartsoft:~/environment $ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
618ab61ca953 simple-app-image "/bin/sh -c 'FLASK_A..." 7 minutes ago Up 7 minutes 0.0.0.0:80->80/tcp, :::80->80/tcp simpleweb
vocstartsoft:~/environment $ docker stop 618
618
vocstartsoft:~/environment $ docker run --name simpleweb2 -d -p 80:5000 simple-app-image
a6656d0fbccad4317ca91f9bd032b96ceae93f807ec7f4e794d788f90ae5f52
vocstartsoft:~/environment $ docker inspect simpleweb2
bash: docker: command not found
vocstartsoft:~/environment $ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a6656d0fbccad4317ca91f9bd032b96ceae93f807ec7f4e794d788f90ae5f52
vocstartsoft:~/environment $ docker inspect simpleweb2
[{"Id": "a6656d0fbccad4317ca91f9bd032b96ceae93f807ec7f4e794d788f90ae5f52", "Created": "2021-11-30T23:20:25.946895781Z", "Path": "/bin/sh", "Args": []}
```

This screenshot shows the same AWS Cloud9 IDE setup, but with a modified Dockerfile. The 'Dockerfile' now specifies 'FROM ubuntu' instead of 'python:3.8-slim':

```
1 FROM ubuntu
2 WORKDIR /app
3 RUN apt-get update
4 RUN apt-get install -y python3 python3-pip
5 RUN pip install flask
6
7 COPY app.py /opt/app.py
8
9 # EXPOSE 80
10
11 ENTRYPOINT FLASK_APP=/opt/app.py flask run --host=0.0.0.0
```

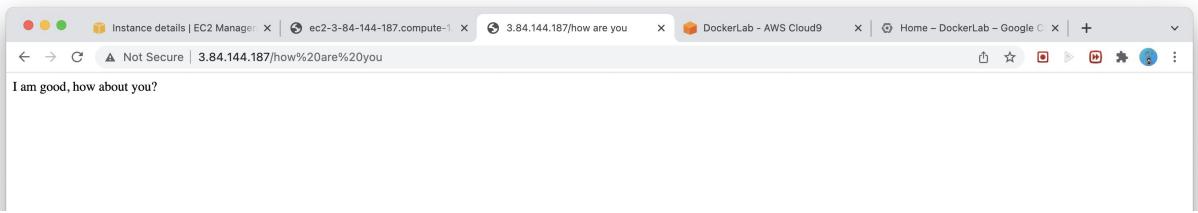
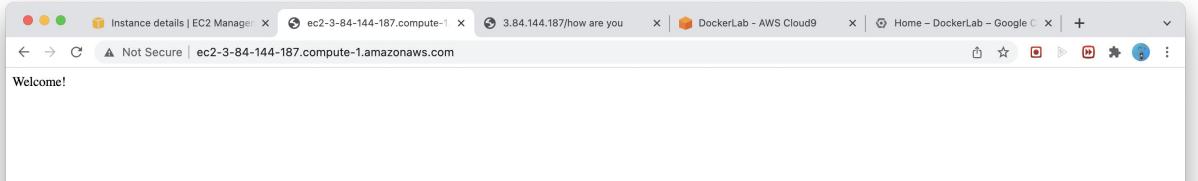
The terminal output remains identical to the first screenshot, showing the creation of a Docker container and its inspection.

```

12:37 Dockerfile Spaces: 4
bash - root@618ab61ca9t x Immediate bash - ip-172-31-47-37.e.x + 
curl: (7) Failed to connect to localhost port 80: Connection refused
root@618ab61ca9t:/# curl http://localhost:8000
curl: (7) Failed to connect to localhost port 8000: Connection refused
root@618ab61ca9t:/# curl http://localhost:5000
vocstartsoft:/environment $ docker images
docker: 'images' is not a docker command.
See 'docker --help'.
vocstartsoft:/environment $ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
icsweb-image latest 2903c5fcab8 39 minutes ago 234MB
simple-app-image latest a8fffc486c2c 50 minutes ago 417MB
<none> <none> f4a89bcraf0d6 About an hour ago 417MB
ping58977/simple-web latest 6f274ba349a7 16 hours ago 313MB
ubuntu latest ba6accfed29 6 weeks ago 72.8MB
ubuntu 16.04 b6f597652425 3 months ago 135MB
lambda/lambda python3.8 094248252696 10 months ago 524MB
lambda/lambda nodejs12.x 2244ada8399c 10 months ago 390MB
lambda/lambda nodejs10.x db930e728e7b 10 months ago 385MB
lambda/lambda python3.x 22d456fd9268 10 months ago 946MB
lambda/lambda python2.7 177c85a10179 10 months ago 894MB
lambda/lambda python2.7 d96a91fe4c80 10 months ago 763MB
lambda/lambda nodejs18.10 5754fe26e6e 10 months ago 813MB
vocstartsoft:/environment $ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a6656d0fbca simple-app-image "/bin/sh -c 'FLASK_A..." 13 minutes ago Up 13 minutes 0.0.0.0:80->5000/tcp, :::80->5000/tcp simpleweb2
vocstartsoft:/environment $ 

```

AWS



Exercise 2: Docker in GCP with Python Flask

Lab report screen-shot:

Google Cloud Platform DockerLab

CLOUD SHELL EDITOR

Dockerfile

```

FROM ubuntu
RUN apt-get update
RUN apt-get install -y python3 python3-pip
RUN pip install flask
COPY app.py /opt/app.py
ENTRYPOINT FLASK_APP=/opt/app.py flask run --host=0.0.0.0

```

PINNED

Marketplace

Billing

APIs & Services

Support

IAM & Admin

Getting started

Compliance

Security

Anthos

Compute

Compute Engine

Kubernetes Engine

Cloud Code minikube

Ln 10, Col 1 LF UTF-8 Spaces: 4 Dockerfile

The screenshot shows the Google Cloud Platform Cloud Shell interface. On the left, the sidebar lists various services like Marketplace, Billing, APIs & Services, Support, IAM & Admin, Getting started, Compliance, Security, and Anthos. The main area is titled 'CLOUD SHELL' and shows an 'Editor' window with the file 'app.py' open. The code in 'app.py' is:

```
1 import os
2 from flask import Flask
3 app = Flask(__name__)
4
5 @app.route("/")
6 def main():
7     return "Welcome!"
8
9 @app.route("/how are you")
10 def hello():
11     return "I am good, how about you?"
12
13 if __name__ == "__main__":
14     app.run(host="0.0.0.0", port=8080)
```

The screenshot shows the Google Cloud Platform Cloud Shell interface with a terminal tab active. The terminal output shows the steps of a Docker build:

```
Processing triggers for liblc-bin (2.31-0ubuntu9.2) ...
Processing triggers for ca-certificates (20210119-20.04.2) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Removing intermediate container 184ac1675b71
-> 5e1594f892c8
Step 4/6 : RUN pip install flask
--> Running in 9a9f2f24d4e4
Collecting flask
  Downloading Flask-2.0.2-py3-none-any.whl (95 kB)
Collecting Werkzeug==2.0.0
  Downloading Werkzeug-2.0.0-py3-none-any.whl (208 kB)
Collecting click==7.1.2
  Downloading click-7.1.2-py3-none-any.whl (97 kB)
Collecting Jinja2>=3.0
  Downloading Jinja2-3.0.3-py3-none-any.whl (133 kB)
Collecting itsdangerous<2.0
  Downloading itsdangerous-2.0.1-py3-none-any.whl (18 kB)
Collecting MarkupSafe<2.0
  Downloading MarkupSafe-2.0.1-py3-none-any.whl (30 kB)
Installing collected packages: Werkzeug, click, MarkupSafe, Jinja2, itsdangerous, flask
Successfully installed Jinja2-3.0.3 MarkupSafe-2.0.1 Werkzeug-2.0.2 click-7.1.2 flask-2.0.2 itsdangerous-2.0.1
Removing intermediate container 9a9f2f24d4e4
-> 259fc47f2d70
Step 5/6 : COPY app.py /opt/app.py
--> Running in 5072756976bb
Step 6/6 : EXPOSE 5000
--> Running in 5072756976bb
Removing intermediate container 5072756976bb
-> 21fa39799459
Successfully built 21fa39799459
Successfully tagged simpleweb:latest
pink5972@cloudshell:~ (dockerlab-333723)$ docker images
docker: command not found
Simple help: 'docker --help'
pink5972@cloudshell:~ (dockerlab-333723)$ docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
simpleweb           latest        21fa39799459   43 seconds ago   417MB
ubuntu              latest        bafacceddd29    6 weeks ago       72.8MB
pink5972@cloudshell:~ (dockerlab-333723)$ docker run --name simple -d -p 5000:5000 simpleweb
7e66912d20a13edc046ff4a70cb2d6a2e8a7c274f018f5a5de6a7f79502le
pink5972@cloudshell:~ (dockerlab-333723)$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
7e66912d20a1        simpleweb          "/bin/sh -c 'FLASK_A..."   5 seconds ago      Up 4 seconds         0.0.0.0:5000->5000/tcp   simple
```

The screenshot shows a web browser window with the URL <https://5000-cs-222009696386-default.cs-us-central1-pits.cloudshell.dev/>. The page displays the message "Welcome!".

