Experiment-VI: Use native MySQL connections from Google Compute Engine

HARDWARE REQUIREMENTS: Core I5 Processor, 4 GB RAM, 40GB HDD
SOFTWARE REQUIREMENTS: Google Cloud Platform, Compute Engine, Google cloud shell, nano editor, PhP, MySQL, PhPMyAdmin

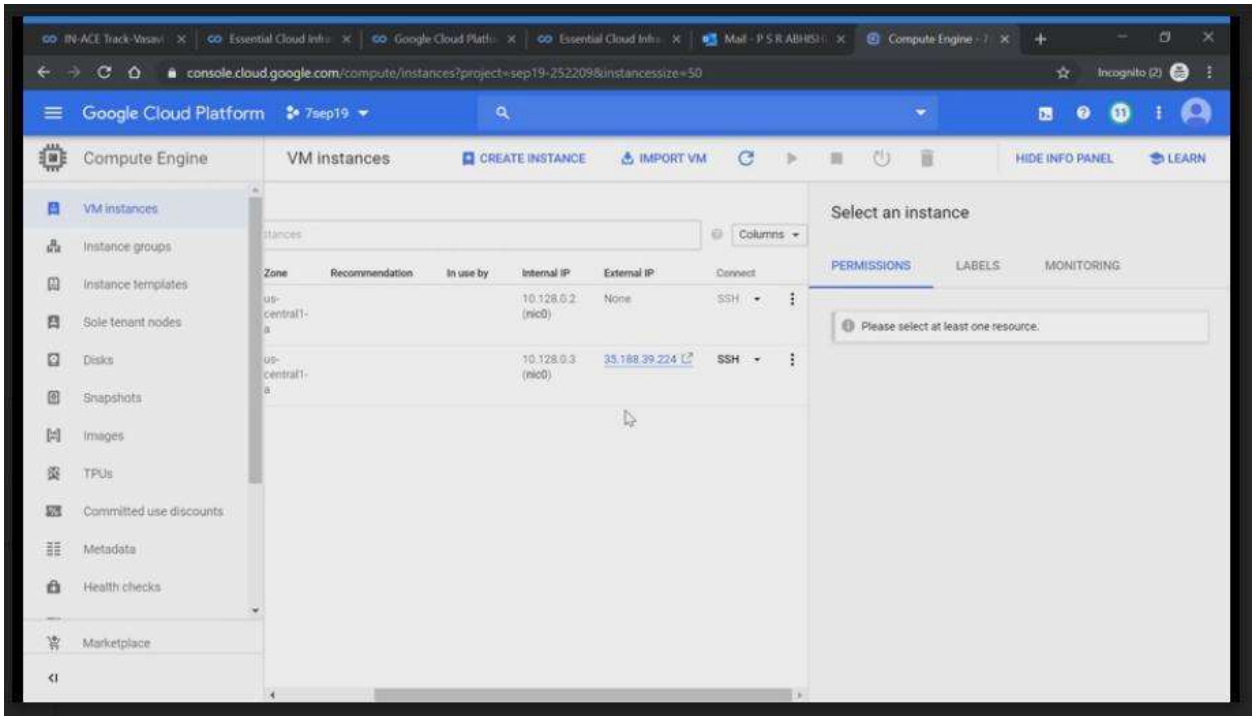
Description: This experiment will need access to Google Cloud platform. We will create a Google Cloud Ubuntu Instance and create a PhP application Framework on it. Next we clone a PhP application on Ubuntu instance and expose the deployed PhP application. Configure MySql & PhPMyadmin to make a full stack web application exposed by using Google Compute Instances

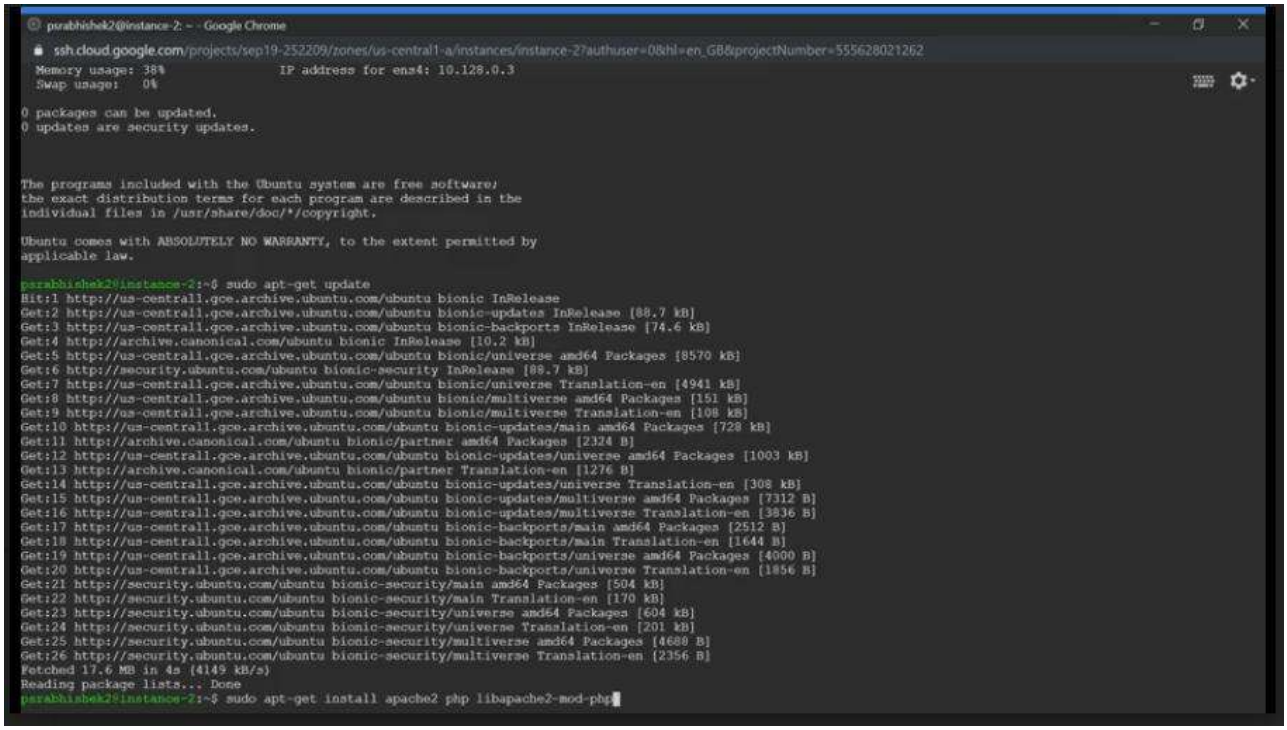
Steps to install and run PHP application with MYSQL on google cloud platform, using Compute Engine

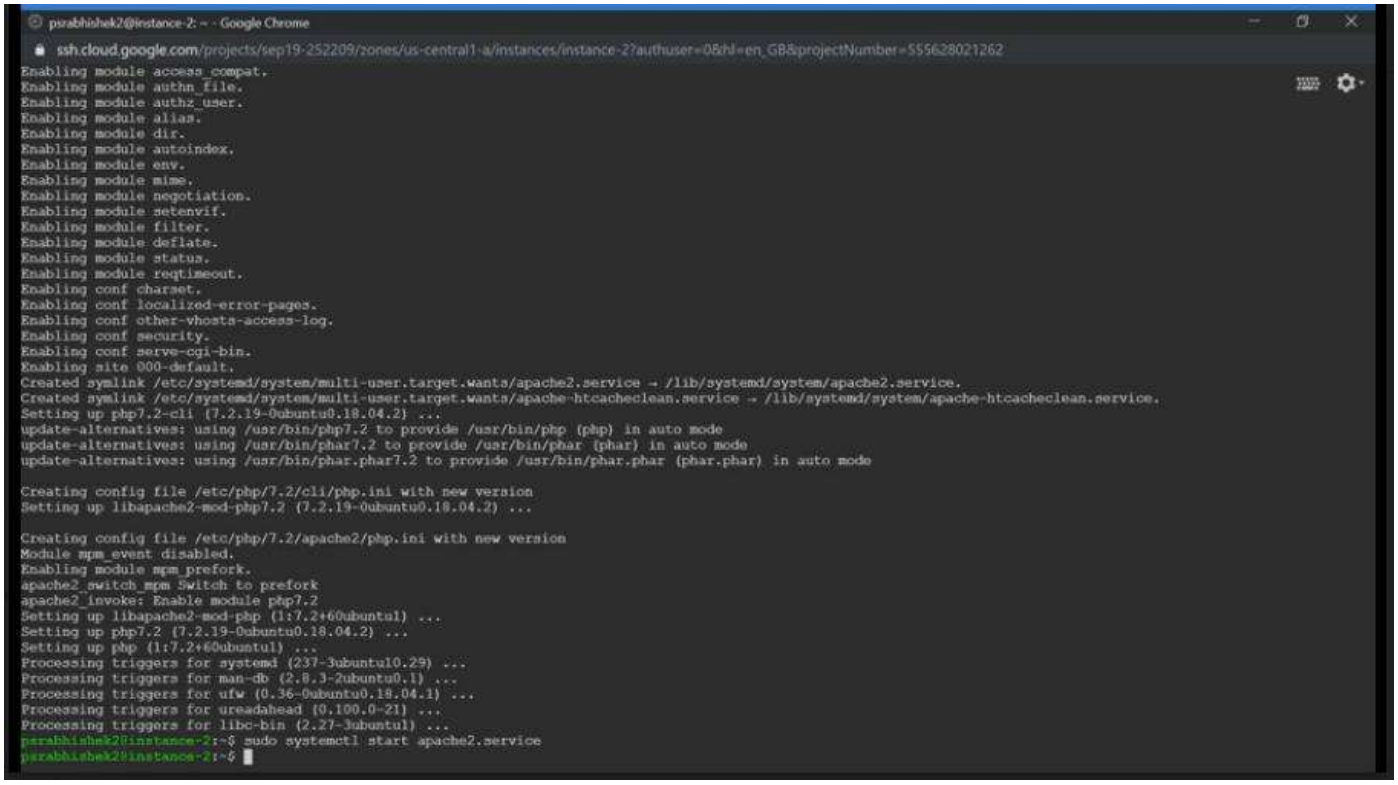
1. Create Ubuntu VM in Google Cloud.
   sudo apt-get update
   sudo apt-get install apache2 php libapache2-mod-php
2. Run Apache Service
   sudo systemctl start apache2.service #start apache
3. Check if server is running
4. Create a file and ping that on the server
   sudo sh -c 'echo "Welcome to my Home page" > /var/www/html/phpinfo.php'
5. Install MySql
   sudo apt-get -y install mysql-server
6. Improve MySql Installation security
   sudo mysql_secure_installation
7. Connect to MySql
   sudo mysql -u root –p
8. Install phpmyadmin
   sudo apt-get install phpmyadmin
   sudo dpkg-reconfigure phpmyadmin //incase of wrong config of phpmyadmin installation

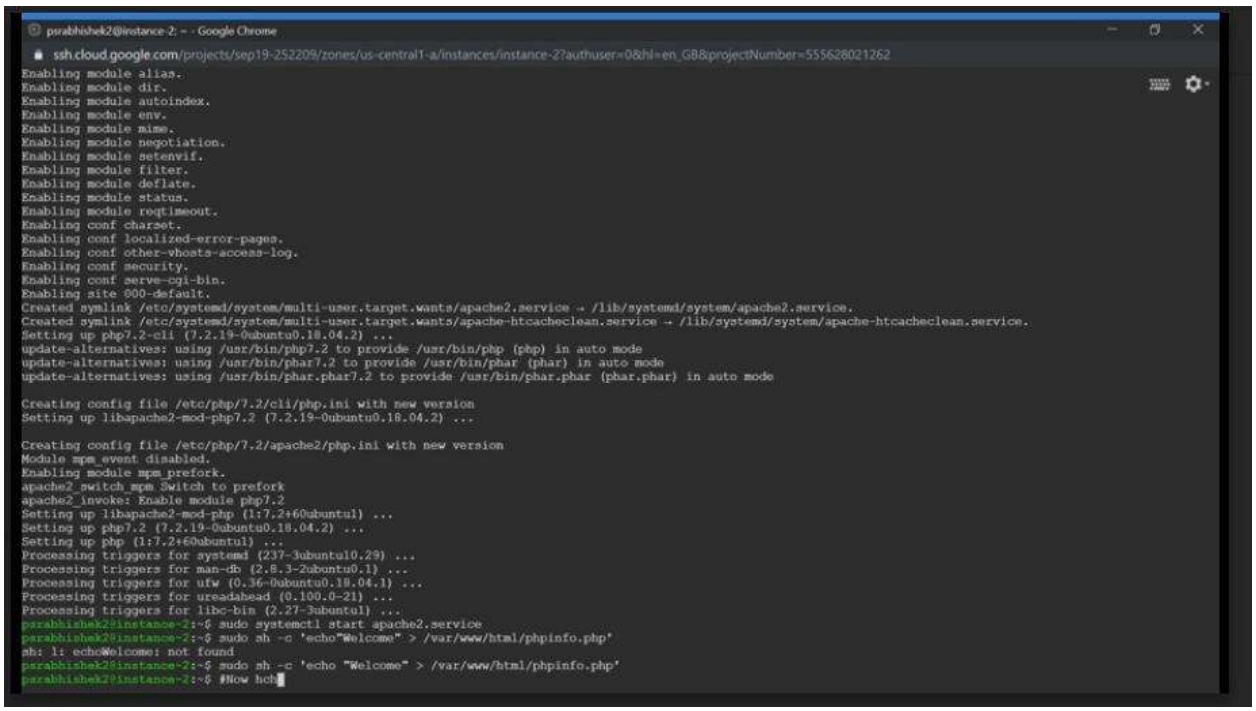
9. Configure phpMyAdmin
    a. Select apache2.
    b. Select yes to use dbconfig-common for database setup.
    c. Enter the database administrator's password that you chose during MySQL configuration.
    d. Enter a password for the phpMyAdmin application.
10. Incase of wrong config of phpmyadmin installation
    sudo dpkg-reconfigure phpmyadmin
11. Incase user doesn’t have privileges then lookup for the users using
    a. Select * from mysql.users
12. Login to phpmyadmin using the user as phpmyadmin
13. Upload file using Google shell.
14. Copy the files from the local folder to /var/www/html
15. Transferring files
    gcloud compute scp [LOCAL_FILE_PATH] lamp-tutorial:/var/www/html
    Ping PublicIP/index.php

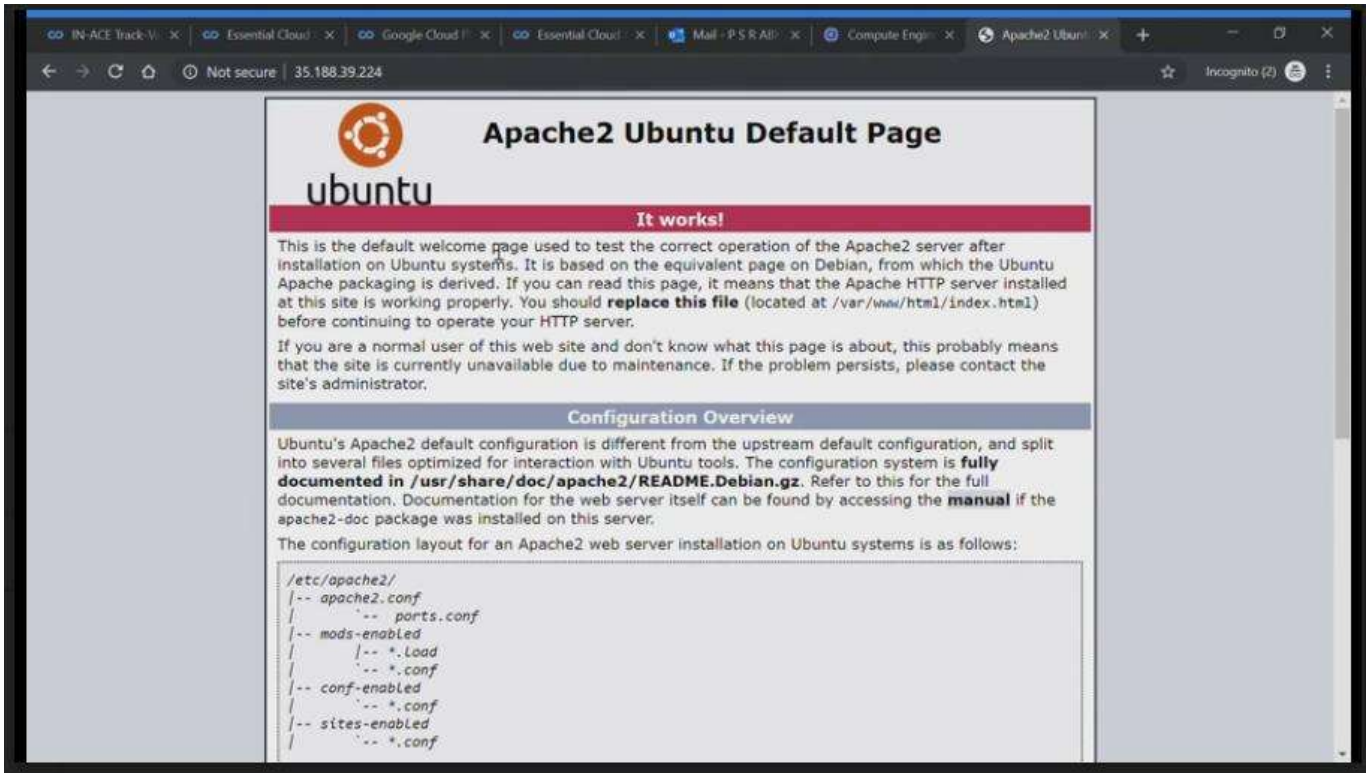
Input / Output:

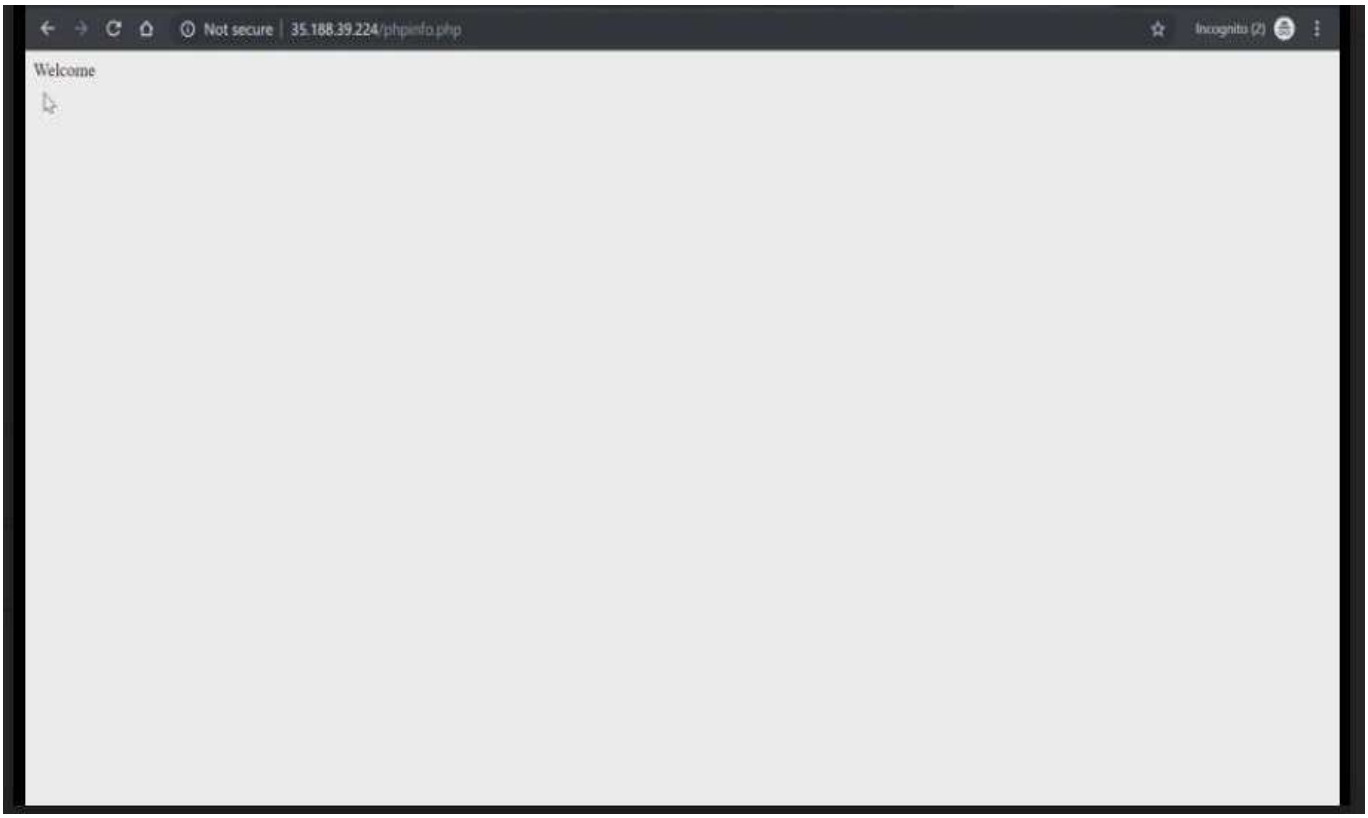

Google Cloud Platform
7sep19
Compute Engine
VM instances
CREATE INSTANCE
IMPORT VM
HIDE INFO PANEL
LEARN
VM instances
Instance groups
Instance templates
Sole tenant nodes
Disks
Snapshots
Images
TPUs
Committed use discounts
Metadata
Health checks
Marketplace
Zone
Recommendation
In use by
Internal IP
External IP
Connect
us-central1-a
10.128.0.2 (nic0)
None
SSH
us-central1-a
10.128.0.3 (nic0)
35.188.39.224
SSH
Select an instance
PERMISSIONS
LABELS
MONITORING
Please select at least one resource.

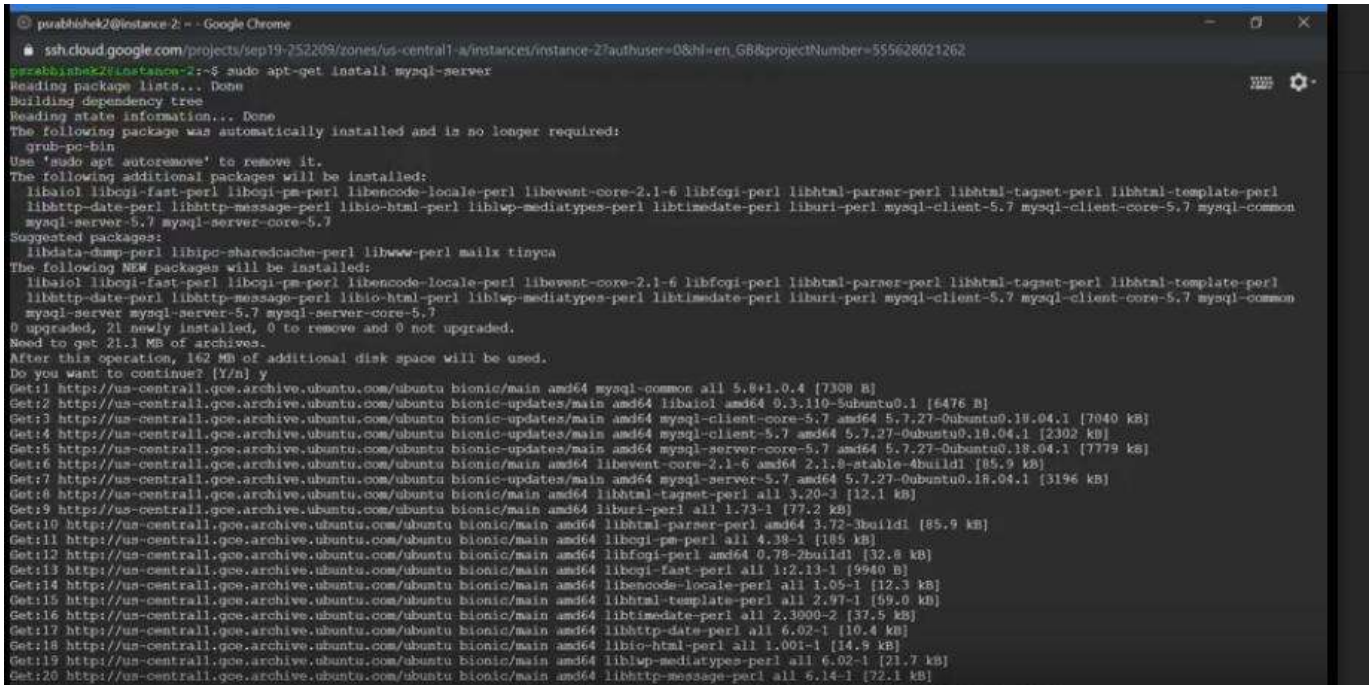

purabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
Memory usage: 38%
IP address for ens4: 10.128.0.3
Swap usage: 0%
0 packages can be updated.
0 updates are security updates.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
purabhishek2@instance-2:~$ sudo apt-get update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
Get:5 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:7 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:8 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:9 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:10 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [728 kB]
Get:11 http://archive.canonical.com/ubuntu bionic/partner amd64 Packages [2324 B]
Get:12 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1003 kB]
Get:13 http://archive.canonical.com/ubuntu bionic/partner Translation-en [1276 B]
Get:14 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [308 kB]
Get:15 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [7312 B]
Get:16 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [3836 B]
Get:17 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [2512 B]
Get:18 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [1644 B]
Get:19 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [4000 B]
Get:20 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [1856 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [504 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [170 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [604 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [201 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [4688 B]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2356 B]
Fetched 17.6 MB in 4s (4149 kB/s)
Reading package lists... Done
purabhishek2@instance-2:~$ sudo apt-get install apache2 php libapache2-mod-php

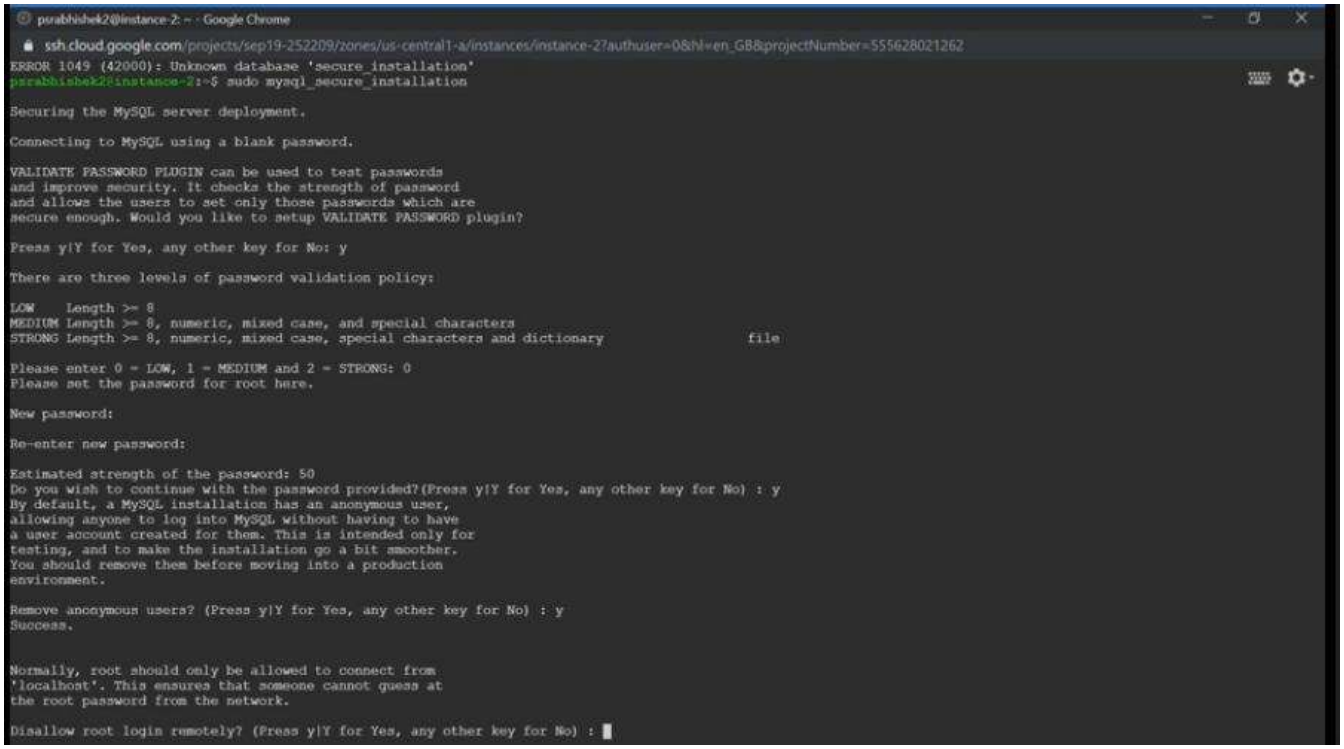

psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Setting up php7.2-cli (7.2.19-0ubuntu0.18.04.2) ...
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (phar.phar) in auto mode
Creating config file /etc/php/7.2/cli/php.ini with new version
Setting up libapache2-mod-php7.2 (7.2.19-0ubuntu0.18.04.2) ...
Creating config file /etc/php/7.2/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php7.2
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...
Setting up php7.2 (7.2.19-0ubuntu0.18.04.2) ...
Setting up php (1:7.2+60ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.29) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
psrabhishek2@instance-2:~$ sudo systemctl start apache2.service
psrabhishek2@instance-2:~$

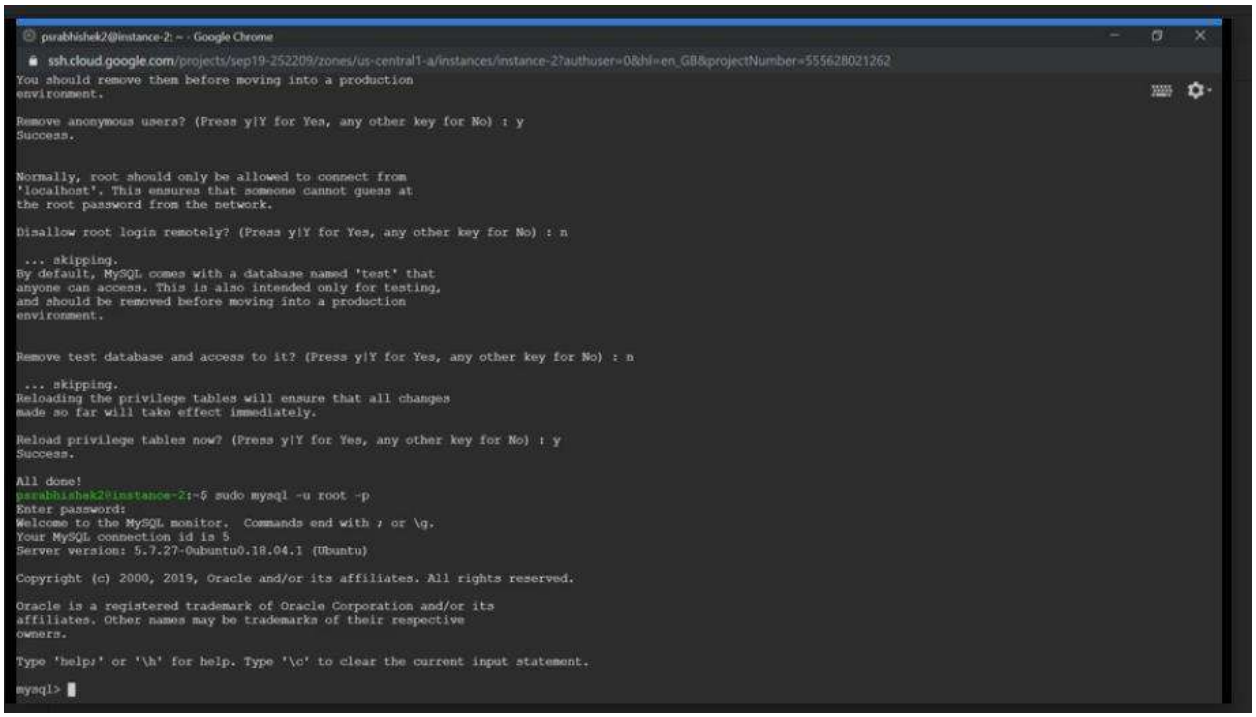

psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Setting up php7.2-cli (7.2.19-0ubuntu0.18.04.2) ...
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (phar.phar) in auto mode
Creating config file /etc/php/7.2/cli/php.ini with new version
Setting up libapache2-mod-php7.2 (7.2.19-0ubuntu0.18.04.2) ...
Creating config file /etc/php/7.2/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php7.2
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...
Setting up php7.2 (7.2.19-0ubuntu0.18.04.2) ...
Setting up php (1:7.2+60ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.29) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
psrabhishek2@instance-2:~$ sudo systemctl start apache2.service
psrabhishek2@instance-2:~$ sudo sh -c 'echo"Welcome" > /var/www/html/phpinfo.php'
sh: 1: echoWelcome: not found
psrabhishek2@instance-2:~$ sudo sh -c 'echo "Welcome" > /var/www/html/phpinfo.php'
psrabhishek2@instance-2:~$ #Now hch

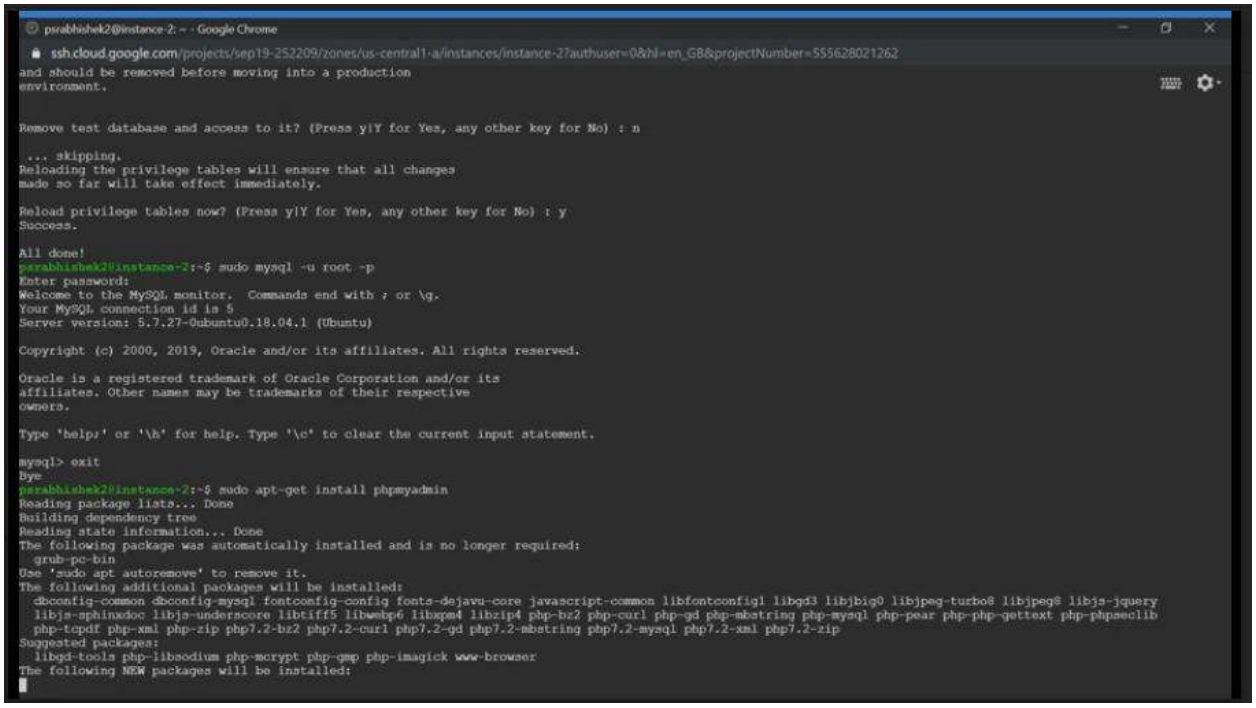

Not secure
35.188.39.224
Incognito (2)
ubuntu
Apache2 Ubuntu Default Page
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf

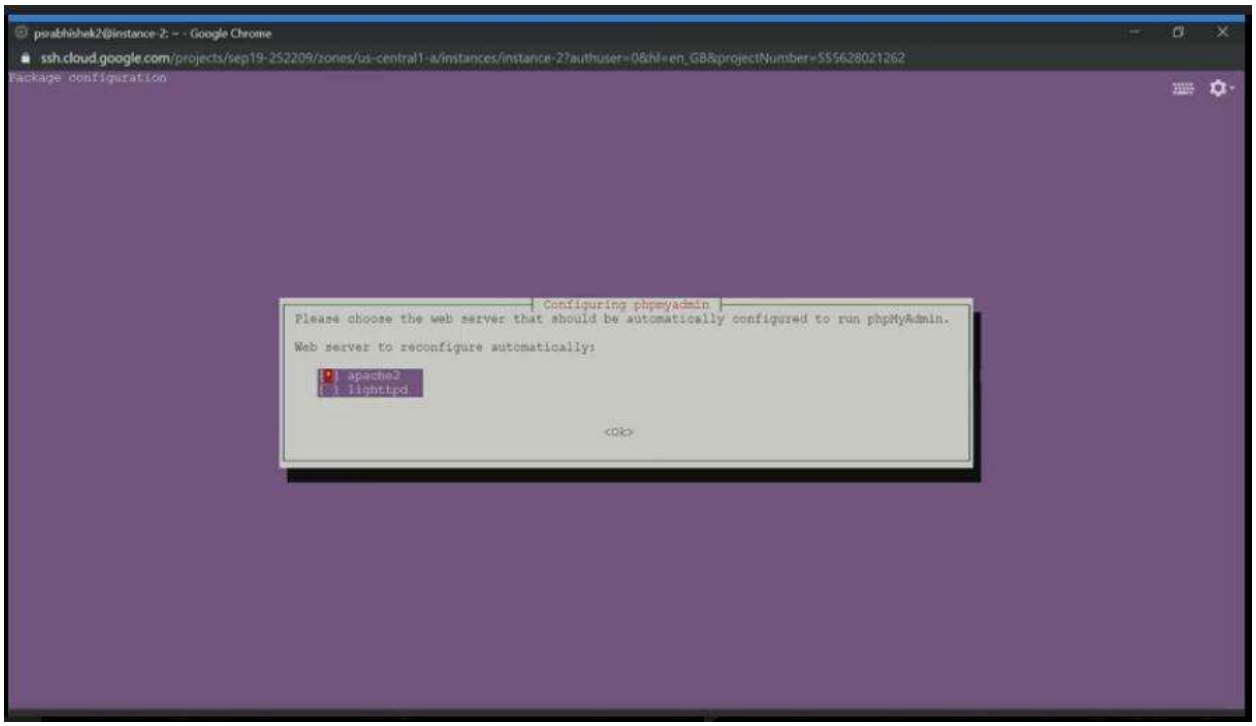

Not secure
35.188.39.224/phpinfo.php
Incognito (2)
Welcome

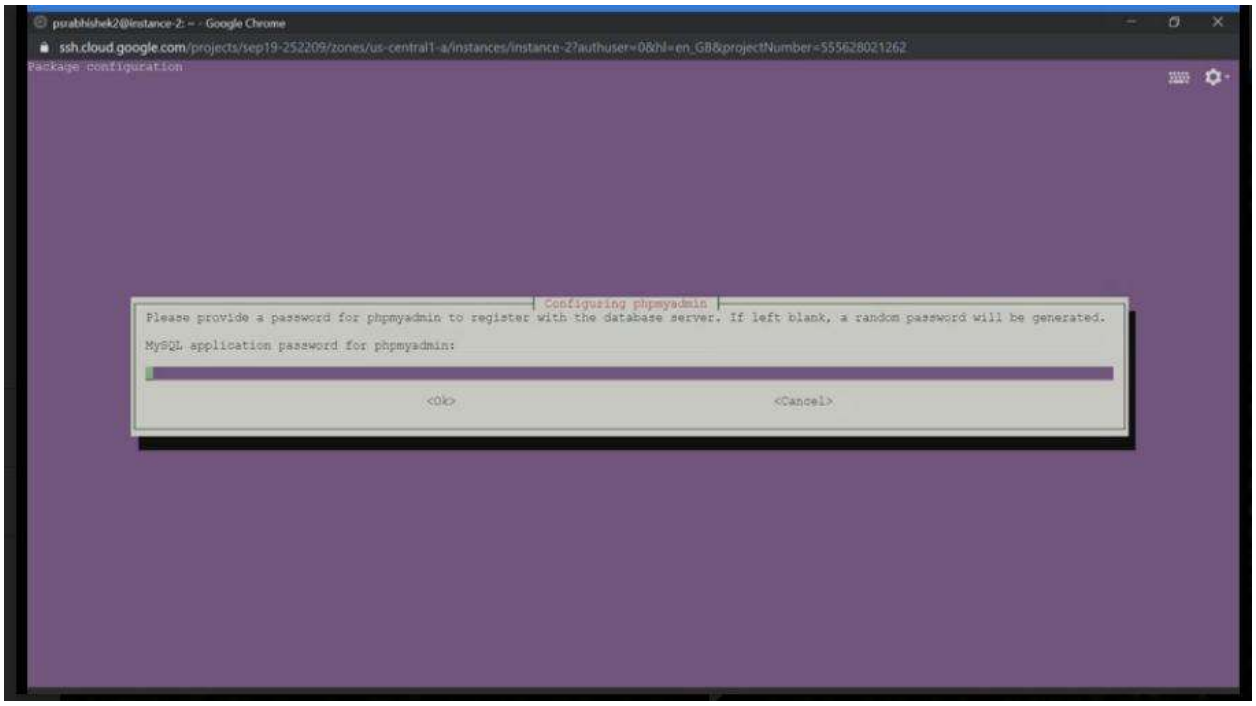

psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
psrabhishek2@instance-2:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7 mysql-common
mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7 mysql-client-core-5.7 mysql-common
mysql-server mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 0 not upgraded.
Need to get 21.1 MB of archives.
After this operation, 162 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 mysql-common all 5.8+1.0.4 [7308 B]
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libaio1 amd64 0.3.110-5ubuntu0.1 [6476 B]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.27-0ubuntu0.18.04.1 [7040 kB]
Get:4 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-5.7 amd64 5.7.27-0ubuntu0.18.04.1 [2302 kB]
Get:5 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-server-core-5.7 amd64 5.7.27-0ubuntu0.18.04.1 [7779 kB]
Get:6 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libevent-core-2.1-6 amd64 2.1.8-stable-4build1 [85.9 kB]
Get:7 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-server-5.7 amd64 5.7.27-0ubuntu0.18.04.1 [3196 kB]
Get:8 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libhtml-tagset-perl all 3.20-3 [12.1 kB]
Get:9 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 liburi-perl all 1.73-1 [77.2 kB]
Get:10 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libhtml-parser-perl amd64 3.72-3build1 [85.9 kB]
Get:11 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libcgi-pm-perl all 4.38-1 [185 kB]
Get:12 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libfcgi-perl amd64 0.78-2build1 [32.8 kB]
Get:13 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libcgi-fast-perl all 1:2.13-1 [9940 B]
Get:14 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libencode-locale-perl all 1.05-1 [12.3 kB]
Get:15 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libhtml-template-perl all 2.97-1 [59.0 kB]
Get:16 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libtimedate-perl all 2.3000-2 [37.5 kB]
Get:17 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libhttp-date-perl all 6.02-1 [10.4 kB]
Get:18 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libio-html-perl all 1.001-1 [14.9 kB]
Get:19 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 liblwp-mediatypes-perl all 6.02-1 [21.7 kB]
Get:20 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/main amd64 libhttp-message-perl all 6.14-1 [72.1 kB]

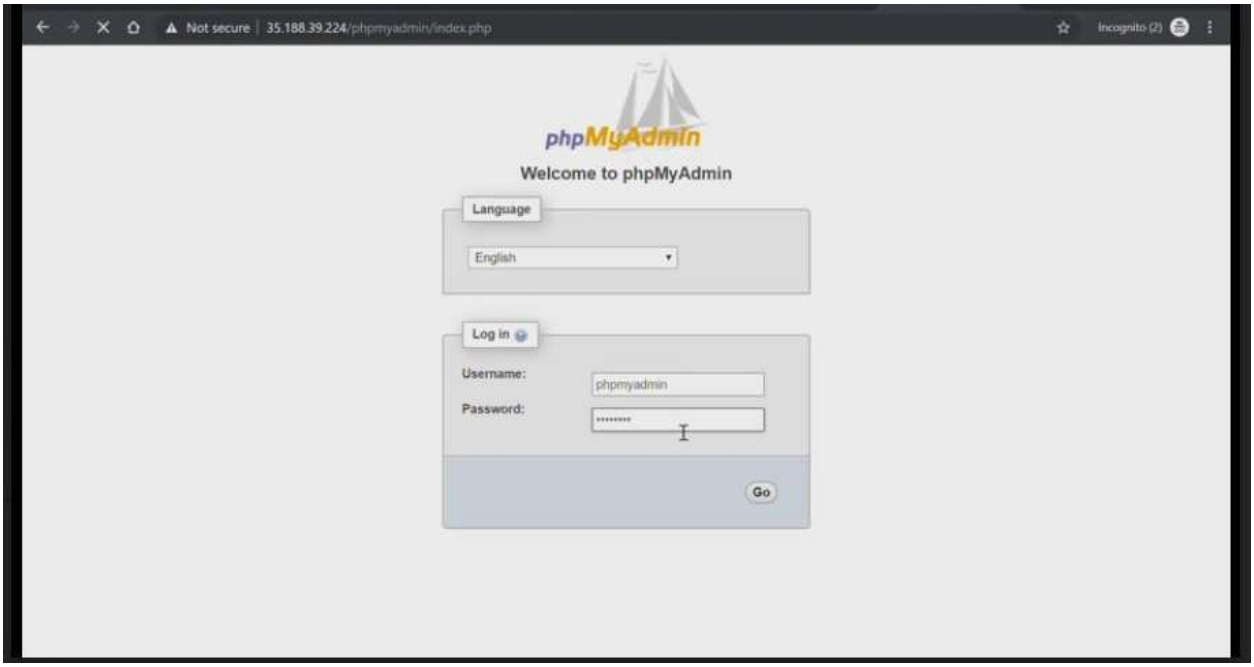

psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
ERROR 1049 (42000): Unknown database 'secure_installation'
psrabhishek2@instance-2:~$ sudo mysql_secure_installation
Securing the MySQL server deployment.
Connecting to MySQL using a blank password.
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?
Press y|Y for Yes, any other key for No: y
There are three levels of password validation policy:
LOW Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary file
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Please set the password for root here.
New password:
Re-enter new password:
Estimated strength of the password: 50
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.
Disallow root login remotely? (Press y|Y for Yes, any other key for No) :


psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
You should remove them before moving into a production
environment.
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n
... skipping.
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n
... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.
All done!
psrabhishek2@instance-2:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.27-0ubuntu0.18.04.1 (Ubuntu)
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>


psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
and should be removed before moving into a production
environment.
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n
... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.
All done!
psrabhishek2@instance-2:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.27-0ubuntu0.18.04.1 (Ubuntu)
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> exit
Bye
psrabhishek2@instance-2:~$ sudo apt-get install phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core javascript-common libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery
libjs-sphinxdoc libjs-underscore libtiff5 libwebp6 libxpm4 libzip4 php-bz2 php-curl php-gd php-mbstring php-mysql php-pear php-php-gettext php-phpseclib
php-tcpdf php-xml php-zip php7.2-bz2 php7.2-curl php7.2-gd php7.2-mbstring php7.2-mysql php7.2-xml php7.2-zip
Suggested packages:
libgd-tools php-libsodium php-mcrypt php-gmp php-imagick www-browser
The following NEW packages will be installed:


psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
Package configuration
Configuring phpmyadmin
Please choose the web server that should be automatically configured to run phpMyAdmin.
Web server to reconfigure automatically:
apache2
lighttpd
<Ok>


psrabhishek2@instance-2: ~ - Google Chrome
ssh.cloud.google.com/projects/sep19-252209/zones/us-central1-a/instances/instance-2?authuser=0&hl=en_GB&projectNumber=555628021262
Package configuration
Configuring phpmyadmin
Please provide a password for phpmyadmin to register with the database server. If left blank, a random password will be generated.
MySQL application password for phpmyadmin:
<Ok>
<Cancel>


Not secure
35.188.39.224/phpmyadmin/index.php
Incognito (2)
phpMyAdmin
Welcome to phpMyAdmin
Language
English
Log in
Username:
phpmyadmin
Password:
Go

Experiment-VII: Deploy a PHP application using Native MySQL connections from Google Cloud SQL to perform CRUD operations.

HARDWARE REQUIREMENTS: Core I5 Processor, 4 GB RAM, 40GB HDD
SOFTWARE REQUIREMENTS: Google Cloud Platform, Compute Engine, Google cloud shell, GIT, Composer, Google Cloud SDK for windows.

Description:

We will clone a sample PhP application from GitHub that allows us to perform CRUD operations on a Bookshelf application, allowing us to insert, retrieve, update and delete the list of books. This application is deployed as a Google App engine application and the CRUD operations are performed over a Google Cloud SQL instance.

Setup Google Cloud SDK

1. To deploy your app with the gcloud tool, you must download, install, and initialize the Cloud SDK, for the appropriate OS
    a. Windows:
    https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe
    b. Start Cloud Shell and enter gcloud init command to login into GCloud environment using GCloud SDK
    c. Enter the project ID taken from Google cloud
    d. Select the zone where the project is deployed

Downloading Sample App

2. Download or clone the app and Navigate to the getting-started directory
    https://github.com/GoogleCloudPlatform/php-docs-samples.git
    cd appengine/php72/getting-started

Informing dependencies using composer.json file

3. Open the composer.json file to review all direct dependencies

```
{
  "require": {
    "google/cloud-storage": "^1.6",
    "slim/slim": "^4.0",
```

```
    "slim/twig-view": "^3.0",
    "slim/http": "^1.0",
    "slim/psr7": "^1.0"
  },
  "autoload": {
    "psr-4": {
      "Google\\Cloud\\Samples\\AppEngine\\GettingStarted\\": "src"
    }
  },
  "require-dev": {
  }
}
```

4. Run ***composer install*** to download dependencies and produce a composer.lock file. The composer.lock file is used to ensure your app will retrieve the same versions of the packages you use across multiple builds and environments.

Initializing the app and defining front controllers

5. The index.php file initializes the app and forwards all requests to controllers defined in the ./src/controllers.php file.

```
// Use the composer autoloader to load dependencies.
require_once __DIR__ . '/vendor/autoload.php';

//  Load the application code.
/** @var Slim\App $app */
$app = require __DIR__ . '/src/app.php';
require __DIR__ . '/src/controllers.php';

// Bootstrap the slim framework to handle the request.
$app->run();
```

Configuring the Cloud SQL instance

6. To create and configure a Cloud SQL instance:
    a. Create a Cloud SQL Second Generation instance.

    gcloud sql instances create INSTANCE_NAME --tier=MACHINE_TYPE --region=REGION

    gcloud sql instances create bookshelf --tier=db-n1-standard-2 --region=us-central1

    Or using the Google Cloud SQL console

b. If you haven't already, set the password for the default user on your Cloud SQL instance:

   gcloud sql users set-password root --host=% --instance [INSTANCE_NAME] --password [PASSWORD]

   gcloud sql users set-password root --host=% --instance=bookshelf --password=root

c. If you don't want to use the default user to connect, create a user.

d. Record the connection name for the instance:

   gcloud sql instances describe [INSTANCE_NAME]

e. For example:

   connectionName: angularjs-crud-project:us-central1:instance1

f. You can also find this value in the Instance details page of the Google Cloud Platform Console.

g. Create a database on your Cloud SQL instance.

   gcloud sql databases create [DATABASE_NAME] --instance=[INSTANCE_NAME]

   gcloud sql databases create book-data --instance=bookshelf

Connecting to Cloud SQL database

7. This sample app uses PHP's PDO to interact with the MySQL database.

   appengine/php72/getting-started/src/app.php

```
43  // Cloud Storage bucket
44  $container['bucket'] = function ($container) {
45      $bucketName = getenv('GOOGLE_STORAGE_BUCKET');
46      // [START gae_php_app_storage_client_setup]
47      // Your Google Cloud Storage bucket name and Project ID can be configured
48      // however fits your application best.
49      // $projectId = 'YOUR_PROJECT_ID';
50      // $bucketName = 'YOUR_BUCKET_NAME';
51      $storage = new StorageClient([
52          'projectId' => $projectId,
53      ]);
```

```
65        $dbPass = getenv('CLOUDSQL_PASSWORD');
66        // [START gae_php_app_cloudsql_client_setup]
67        // Fill the variables below to match your Cloud SQL configuration.
68        // $dbConn = 'YOUR_CLOUDSQL_CONNECTION_NAME';
69        // $dbName = 'YOUR_CLOUDSQL_DATABASE_NAME';
70        // $dbUser = 'YOUR_CLOUDSQL_USER';
71        // $dbPass = 'YOUR_CLOUDSQL_PASSWORD';
72        $dsn = "mysql:unix_socket=/cloudsql/${dbConn};dbname=${dbName}";
73        $pdo = new PDO($dsn, $dbUser, $dbPass);
74        // [END gae_php_app_cloudsql_client_setup]
```

Query a Single Row

8. When the user clicks on a book, the app queries the database and returns a single row that includes the title, author, publication date, and description of the book.

```
$statement = $pdo->prepare('SELECT * FROM books WHERE id = :id');
$statement->bindValue('id', $id, PDO::PARAM_INT);
$statement->execute();
$result = $statement->fetch(PDO::FETCH_ASSOC);
```

Using Cloud Storage

9. Cloud Storage uses buckets to organize and control access to your data.

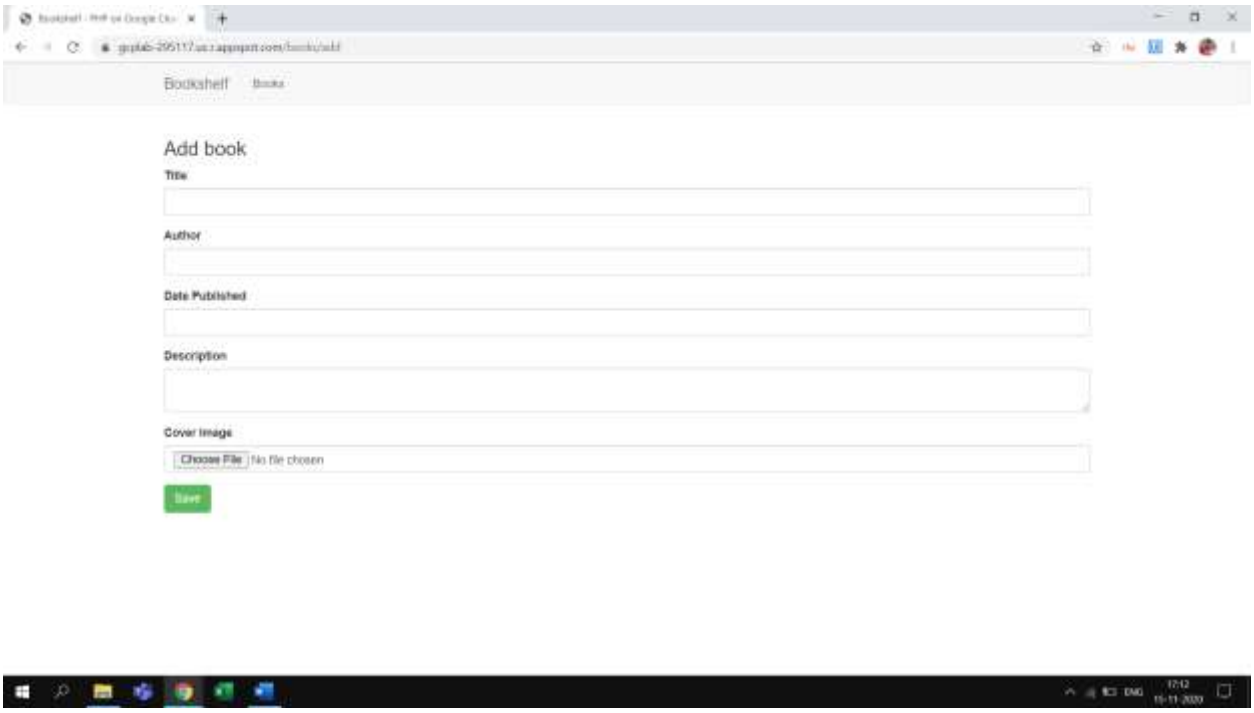
    a. Use Cloud SDK to create a Cloud Storage bucket:
    gsutil mb -l BUCKET_REGION gs://BUCKET_NAME/
    gsutil mb -l us-central1 gs://picture-storage/
    If the storage bucket creation is successful, then you get
    Creating gs://picture-storage/
    Else
    ServiceException: 409 Bucket picture-storage already exists

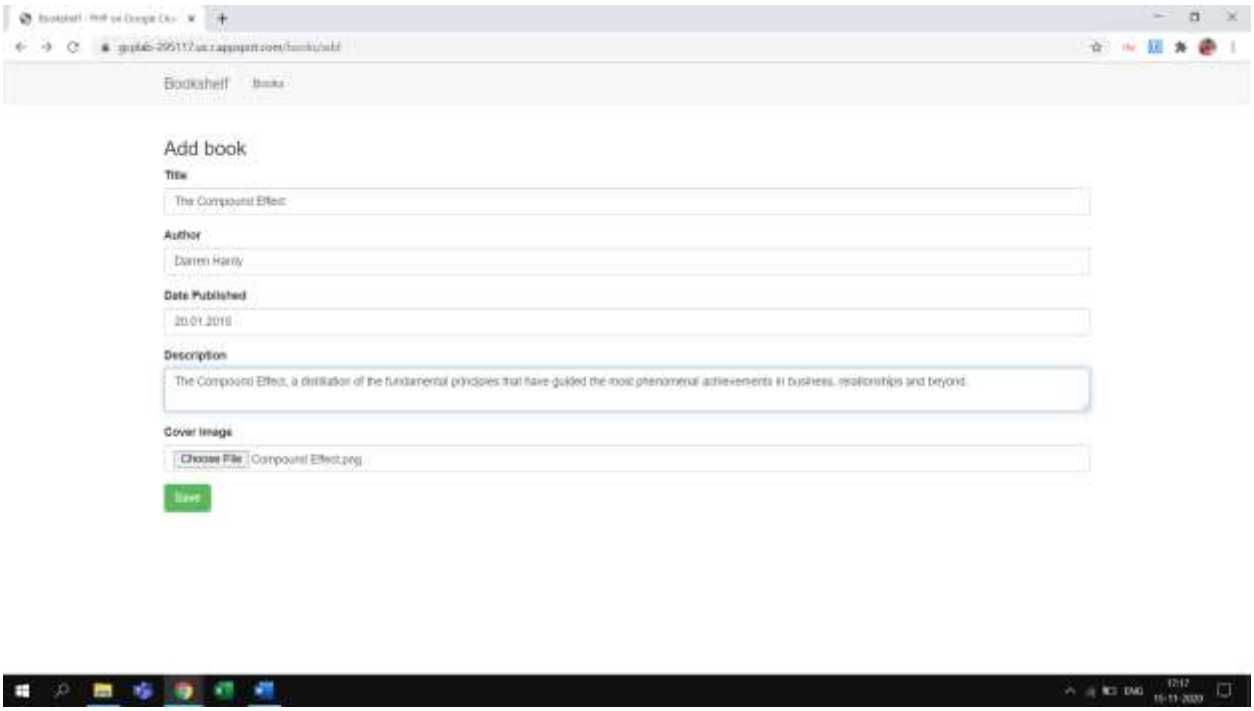
App Deploy

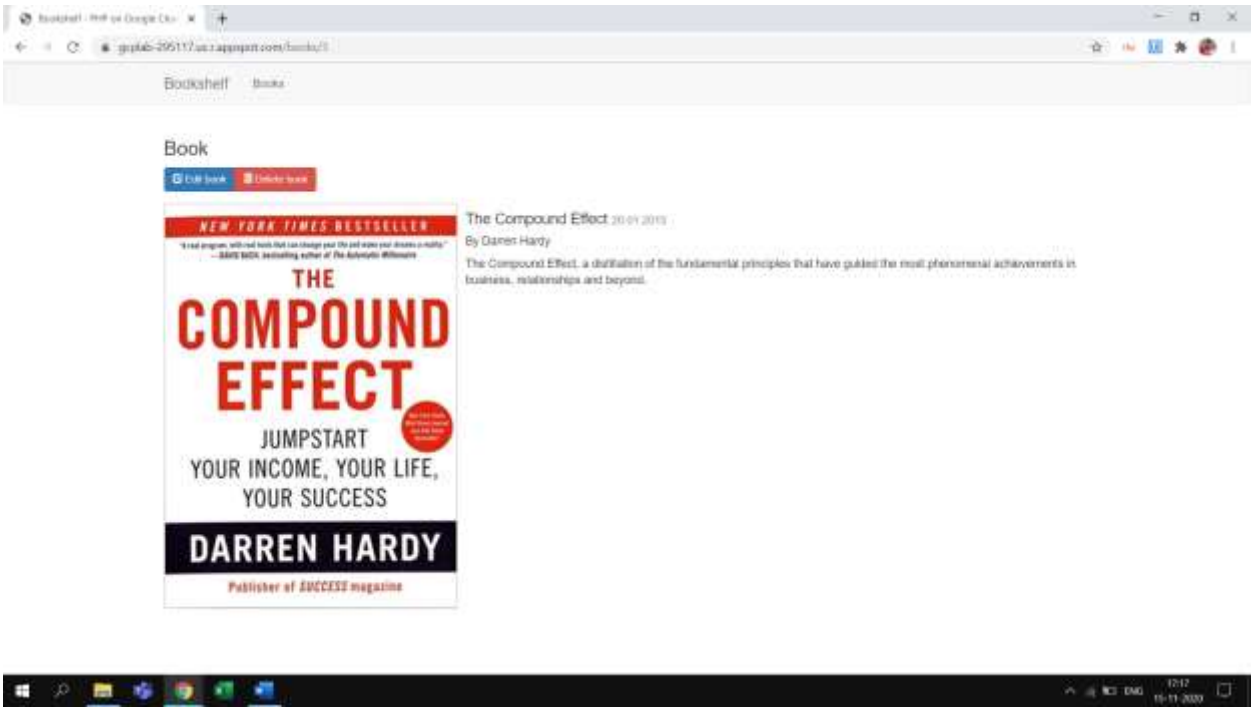
10. Go back to the ***getting-started*** folder and run the command

    gcloud app deploy

Input/Output:




Bookshelf
Books
Add book
Title
The Compound Effect
Author
Darren Hardy
Date Published
20.01.2010
Description
The Compound Effect, a distillation of the fundamental principles that have guided the most phenomenal achievements in business, relationships and beyond.
Cover Image
Choose File
Compound Effect.png
Save


Bookshelf
Books
Book
NEW YORK TIMES BESTSELLER
THE
COMPOUND
EFFECT
JUMPSTART
YOUR INCOME, YOUR LIFE,
YOUR SUCCESS
DARREN HARDY
Publisher of SUCCESS magazine
The Compound Effect
By Darren Hardy
The Compound Effect, a distillation of the fundamental principles that have guided the most phenomenal achievements in business, relationships and beyond.

Experiment-VIII: Deploy and develop scalable compute model using Distributed Storage

Problem Statement: Installation and configuration of Hadoop using Docker Container.

Description:

Create a Machine Image of Ubuntu Bionic 18.04LTS or Xenial 16.04.

To install **Docker CE**, first, you need to remove older versions of **Docker** were called **docker**, **docker.io**, or **docker-engine** from the system using the following command.

$ sudo apt-get remove docker docker-engine docker.io containerd runc

Next, you need to set up the Docker repository to install and update Docker from the repository using following commands.

1. Update the apt package index
   $ sudo apt-get update
2. Install packages to allow apt to use a repository over HTTPS
   $ sudo apt-get install \
   apt-transport-https \
   ca-certificates \
   curl \
   gnupg-agent \
   software-properties-common
3. Add Docker's official GPG key
   $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
4. Verify that you now have the key with the fingerprint 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, by searching for the last 8 characters of the fingerprint
   sudo apt-key fingerprint 0EBFCD88

   pub rsa4096 2017-02-22 [SCEA]

9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

uid [unknown] Docker Release (CE deb) <docker@docker.com>

sub rsa4096 2017-02-22 [S]

5. Use the following command to set up the stable repository

```
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

The lsb_release -cs sub-command below returns the name of your Ubuntu distribution, such as xenial. Sometimes, in a distribution like Linux Mint, you might need to change $(lsb_release -cs) to your parent Ubuntu distribution. For example, if you are using Linux Mint Tessa, you could use bionic. Docker does not offer any guarantees on untested and unsupported Ubuntu distributions.

6. Update the apt package index and install the latest version of **Docker CE** using following commands.

```
$ sudo apt-get update
```

7. Install the latest version of Docker Engine - Community and containerd, or go to the next step 8 to install a specific version

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

8. To install a specific version of Docker Engine - Community, list the available versions in the repo, then select and install: List the versions available in your repo:

```
$ apt-cache madison docker-ce
```

9. Install a specific version using the version string from the second column, for example, 5:18.09.1~3-0~ubuntu-xenial

```
$ sudo apt-get install docker-ce=<VERSION_STRING> docker-ce-
    cli=<VERSION_STRING> containerd.io
```

10. After successfully installing the **Docker CE** package, the service should be auto-started and auto-enabled to start at system boot, you can check its status using the following command.

```
$ sudo systemctl status docker
```

11. Press CTRL C to exit
12. Verify that Docker Engine - Community is installed correctly by running the hello-world image

    $ sudo docker run hello-world
13. This command downloads a test image and runs it in a container. When the container runs, it prints the below informational message

    Unable to find image 'hello-world:latest' locally
    latest: Pulling from library/hello-world
    1b930d010525: Pull complete
    Digest:
        sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
    Status: Downloaded newer image for hello-world:latest
    Hello from Docker!
    This message shows that your installation appears to be working correctly.
    To generate this message, Docker took the following steps:
    1. The Docker client contacted the Docker daemon.
    2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
    3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
    4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.
    To try something more ambitious, you can run an Ubuntu container with:
    $ docker run -it ubuntu bash
    Share images, automate workflows, and more with a free Docker ID:
    https://hub.docker.com/
    For more examples and ideas, visit:
    https://docs.docker.com/get-started/

Dockerizing a Node.js web application

14. Create a new folder namely nodejsapp
15. Make a package.json file as follows

```
{
  "name": "docker_web_app",
  "version": "1.0.0",
  "description": "Node.js on Docker",
  "author": "Sashi's First Nodejs Application on Container
        <sashi.mamidanna@gmail.com>",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "^4.16.1"
  }
}
```

16. Then create a file server.js to create a program that runs on the node. The idea is to enable the server.js file to run on the container at port no 8081

```
'use strict';
const express = require('express');

// Constants
const PORT = 8081;
const HOST = '0.0.0.0';

// App
const app = express();
app.get('/', (req, res) => {
```

```
  res.send('Hello world\n');
});

app.listen(PORT, HOST);
console.log(`Running on http://${HOST}:${PORT}`);
```

17. Create a dockerfile now namely dockerfile in the same directory
    $sudo nano dockerfile

18. Copy the source code into the dockerfile

```
FROM node:10

# Create app directory
WORKDIR /app
COPY . /app
RUN npm install

COPY . .

EXPOSE 8082
CMD [ "node", "server.js" ]
```

19. Now build the docker image with the node application on it
    $sudo docker build -t nodejsapp .

20. Run the application by executing run command on docker
    $sudo docker run -p 8082:8081 nodejsapp

21. The container engine will run the command node server.js that was initialized through the dockerfile. Now the server.js is listening to incoming requests on http://localhost:8081 on

the host operating system. But the application is running on port number 8082 on the docker engine.

22. Open a new ssh connection on the same VM and run the command to send an outgoing request to the application running on docker
    $sudo curl http://localhost:8082
    Hello World

23. This response is a result of the application running on node, devoted on the docker container, that's running on Docker engine available on the Ubuntu OS.
24. Run the bow command to check if the docker image is present in the list of images on Docker C
    $sudo docker ps
25. To stop the docker container image
    $sudo docker stop <docker image ID>
26. To remove the docker image
27. $sudo docker rmi <docker image ID>

Experiment-IX: Analyzing Big Data on Apache Spark Cluster setup using Amazon EMR

HARDWARE REQUIREMENTS: Core I5 Processor, 4 GB RAM, 40GB HDD
SOFTWARE REQUIREMENTS: Amazon AWS, Amazon EMR, Amazon S3, GIT, Gradle 6.x, Java 1.8

Description: Analyzing Big Data on Apache Spark Cluster setup using Amazon EMR.

Amazon EMR cluster provides a managed Hadoop framework that makes it easy, fast and cost-effective to process vast amounts of data dynamically scalable on Amazon EC2 instances.

It is possible to run popular distributed frameworks such as Apache Spark and HBase on Amazon EMR and interact with data in other AWS data stores such as Amazon S3 and Amazon DynamoDB.

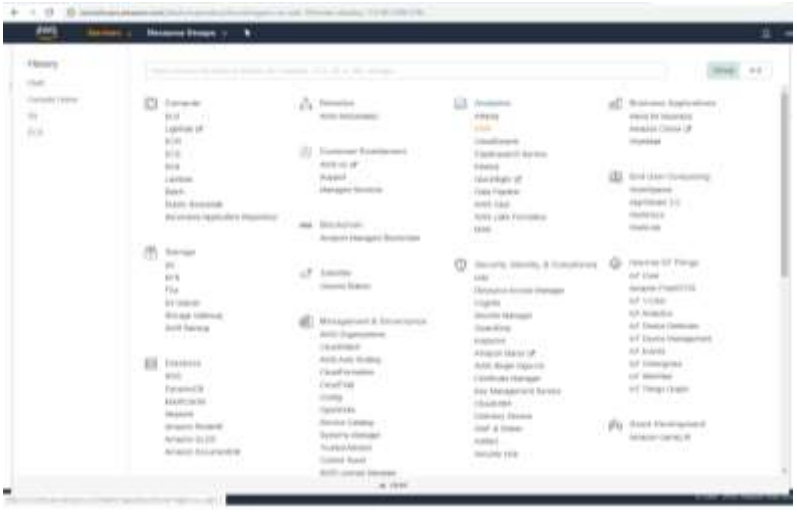
In this experiment we will run out Spark application on top of Hadoop cluster by processing the input data source into Amazon S3.

Amazon S3 is a distributed storage system & AWS's equivalent to HDFS.

By saving our input data source into S3, each spark node deployed on the EMR cluster can read the input data source from S3.

Source Code:

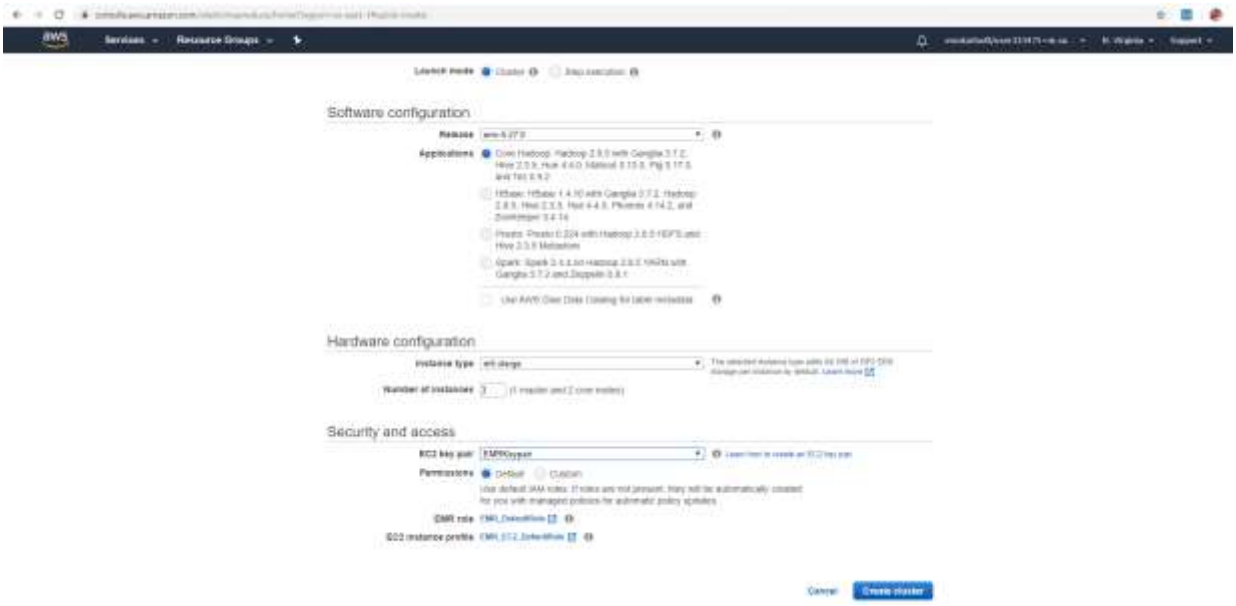
1. Login to Amazon Educate account and navigate to the AWS dashboard.
2. Click on Amazon EMR service



3. Click on create an EMR Cluster.

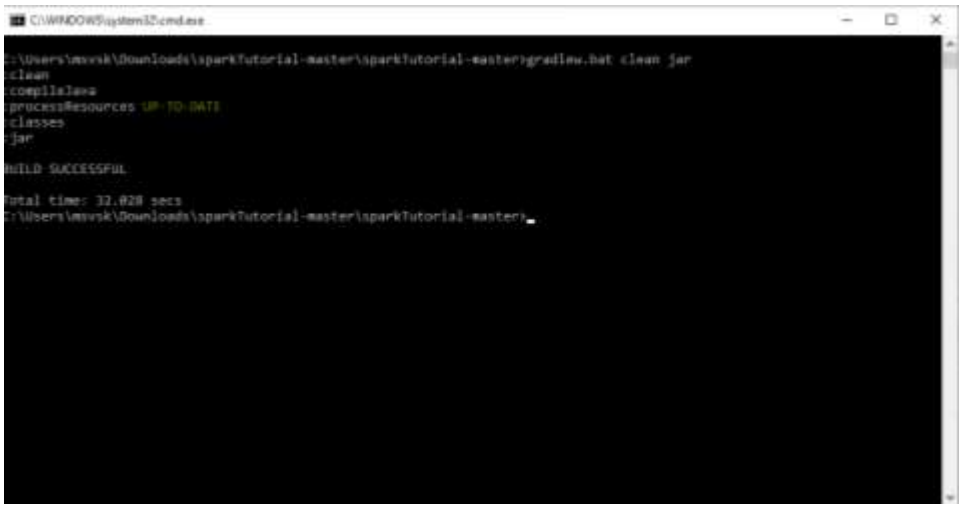
4. Give a name and select the type of applications that run on the cluster.
5. Select “Spark: Spark 2.4.4 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1”
6. Select the hardware configuration of the instance to be m5.xlarge and create one master and a slave node for the experiment purpose
7. Create a EC2 key pair, if not there already, else choose the keypair and continue.
8. To create a EC2 keypair, head back to EC2 dashboard and select Key pairs, to create and download the EC2 keypair.

   EC2 Keypair is necessary to login to the Master node, for submitting big data jobs onto the Spark environment.
9. Leave the rest of the security permissions at default options and click on create cluster

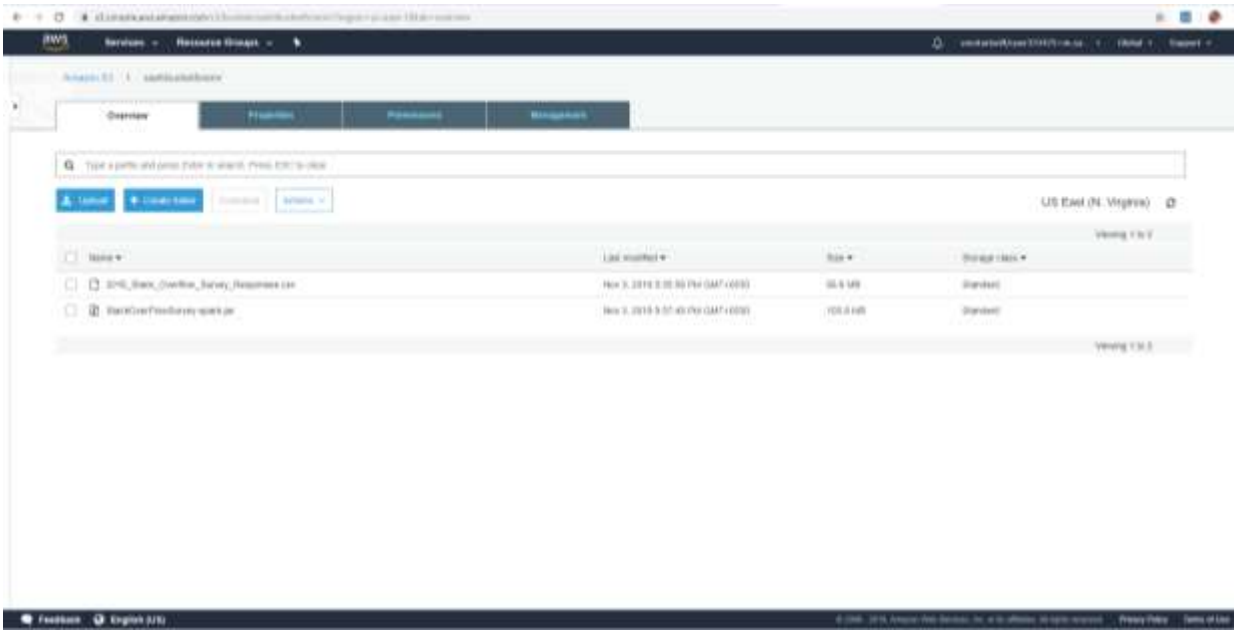


10. The cluster creation will take around 10-15 minutes and in the process will create one master and slave node AMI’s.
11. Meanwhile, navigate to the website stackoverflow.com/research to download a sample dataset for processing data. Download the 2016 Full Data Set
12. Create a S3 storage bucket for uploading the “2016 Stack Overflow Survey Responses.csv”. Ensure that the filename doesn’t have spaces.
13. Download the GIT repository “https://github.com/jleetutorial/sparkTutorial”

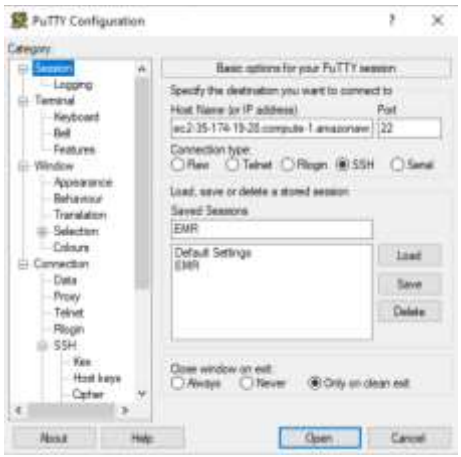
14. Open the source code from sparkSQL file “StackOverFlowSurvey.java”
15. When we run the Apache Spark application on the Amazon EMR cluster, Spark cluster will know how many nodes are available and where is the Master machine. So the cluster will provide a sensible setting at runtime.
16. The source code has already mentioned the settings for Master node, which will override the runtime configuration. But since we are running the application on a Local mode, Spark application will only run on one machine.
17. Remove the “Master option” when creating the Spark Session from the source code.
    SparkSession session = SparkSession.builder().appName("StackOverFlowSurvey").
    getOrCreate();
18. Update the name of the S3 bucket in the source code by using the name of the path as “s3n://bucket_name/input_file.csv”
    Dataset<Row> responses = dataFrameReader.option("header","true").
    csv("*s3n://sashibucketforemr/2016_Stack_Overflow_Survey_Responses.csv*");
19. Save the file and create the jar file out of the source code using Gradle Build batch file.
20. Open command prompt and navigate to the project folder “sparkTutorial-Master”
21. Type the below command to create a jar file in the ./build/libs folder
    gradlew.bat clear jar



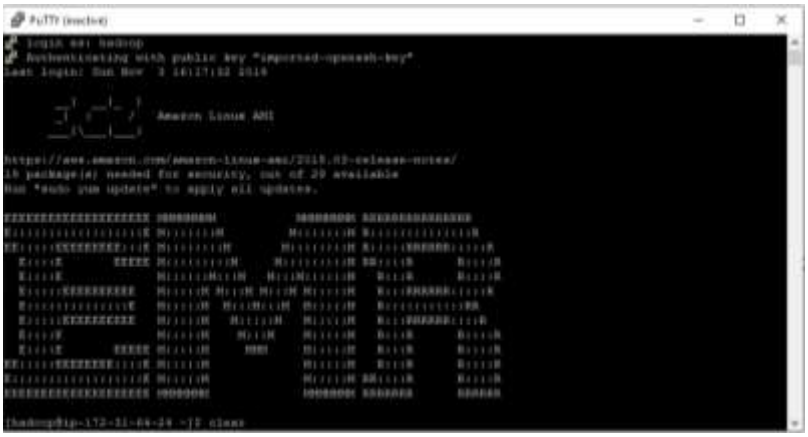
22. Now upload this jar file into the S3 bucket that was created to upload the input CSV.



23. Login to the Spark Cluster using SSH. Use PuttyGen to convert the .pew keypair into putty compatible key (.ppk)
24. Create a session to the Spark Cluster on putty using the connection URL and the Auth key on the session

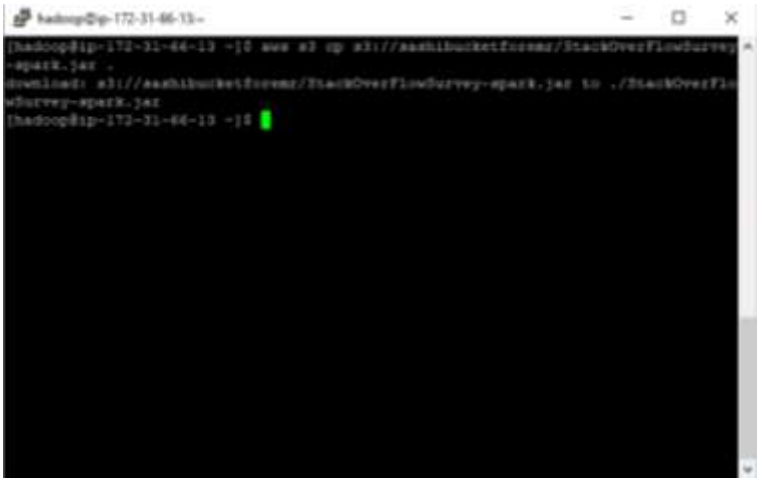


25. Login as the default user “Hadoop”, and you have now entered the EMR cluster console.



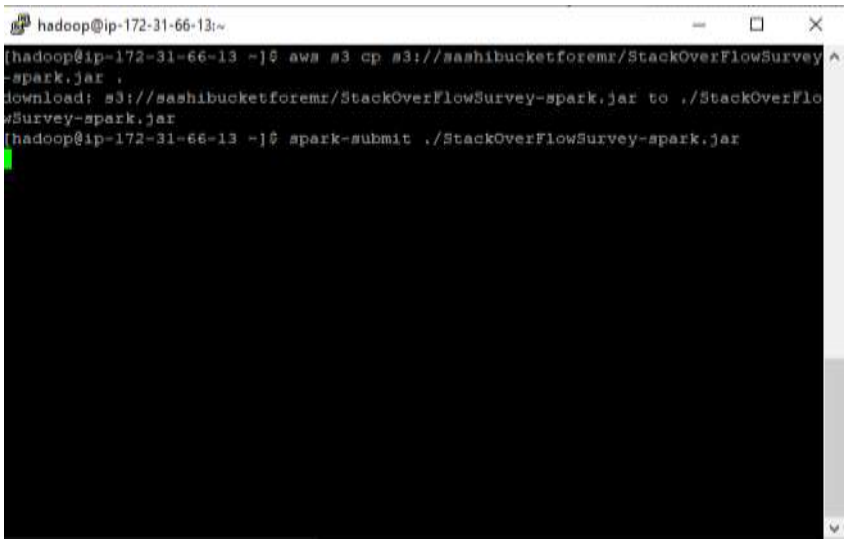
26. Now copy the jar from S3 into the cluster for running the application using the following command

aws s3 cp s3:/bucket_name/file_name.jar .



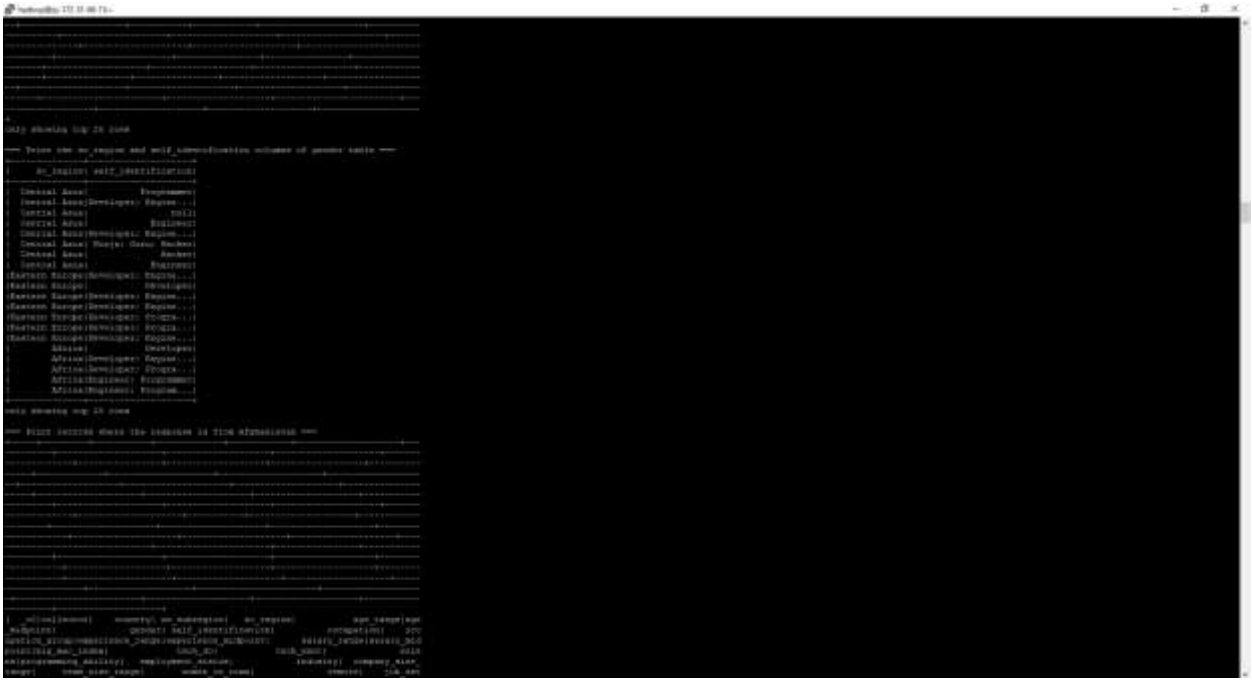
27. Now the jar is copied into the cluster. Check by typing *ls command*
28. Now run the application using the *spark-submit command*

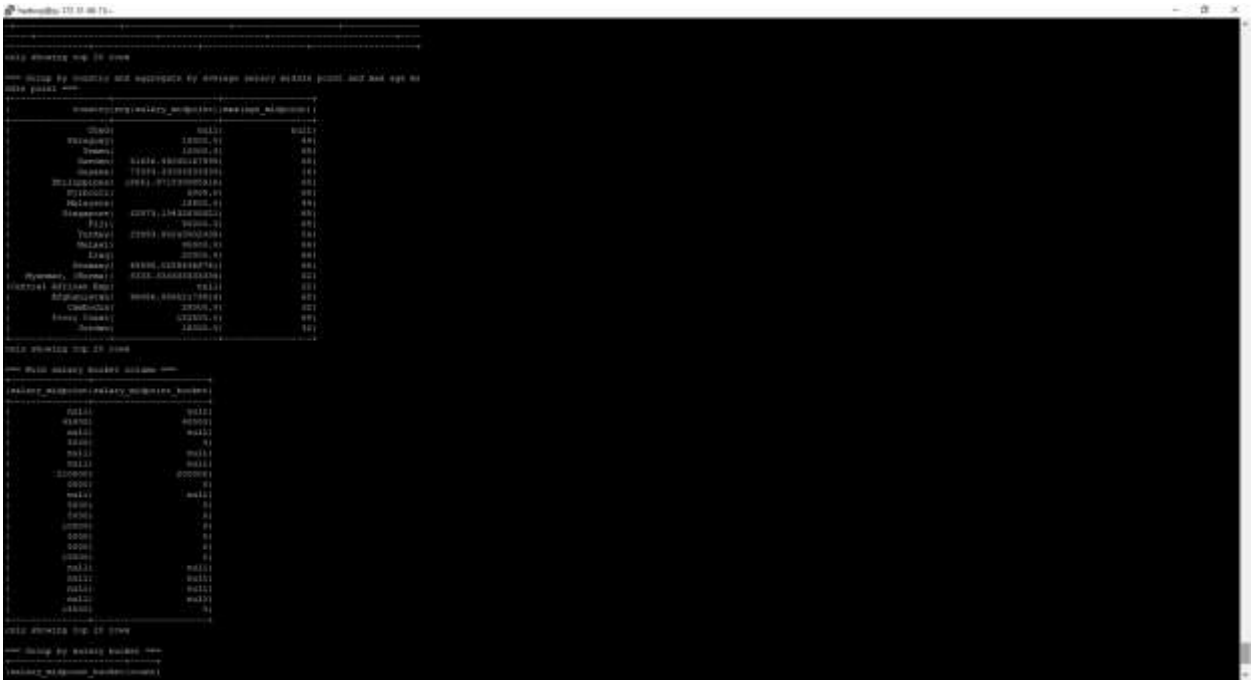
spark-submit ./file_name.jar



Input / Output:

The spark job executes on the EMR cluster and generates all the reports from the input file that was loaded from Amazon S3





Experiment-X: Deploy and develop scalable compute model using Distributed Storage.

Problem Statement: Implement a distributed application on Hadoop framework to count word frequency with Map Reduce**.**

Description:

In Hadoop, MapReduce is a computation that decomposes large manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers. The results of tasks can be joined together to compute final results.

MapReduce consists of 2 steps:

- **Map Function –** It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair).

Example – (Map function in Word Count)

Input	Set of data	Bus, Car, bus, car, train, car, bus, car, train, bus, TRAIN,BUS, bus, car, CAR, car, BUs, TrAin
Output	Convert into another set of data (Key,Value)	(Bus,1), (Car,1), (bus,1), (car,1), (train,1), (car,1), (bus,1), (car,1), (train,1), (bus,1), (TRAIN,1),(BUS,1), (BUS,1), (car,1), (CAR,1), (car,1), (BUs,1), (TrAin,1)

- **Reduce Function –** Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.

Example – (Reduce function in Word Count)

Input **(output of Map function)**	Set of Tuples	(Bus,1), (Car,1), (bus,1), (car,1), (train,1), (car,1), (bus,1), (car,1), (train,1), (bus,1), (TRAIN,1),(BUS,1), (buS,1), (caR,1), (CAR,1), (car,1), (BUS,1), (TRAIN,1)
Output	Converts into smaller set of tuples	(BUS,7), (CAR,7), (TRAIN,4)

Work Flow of the Program

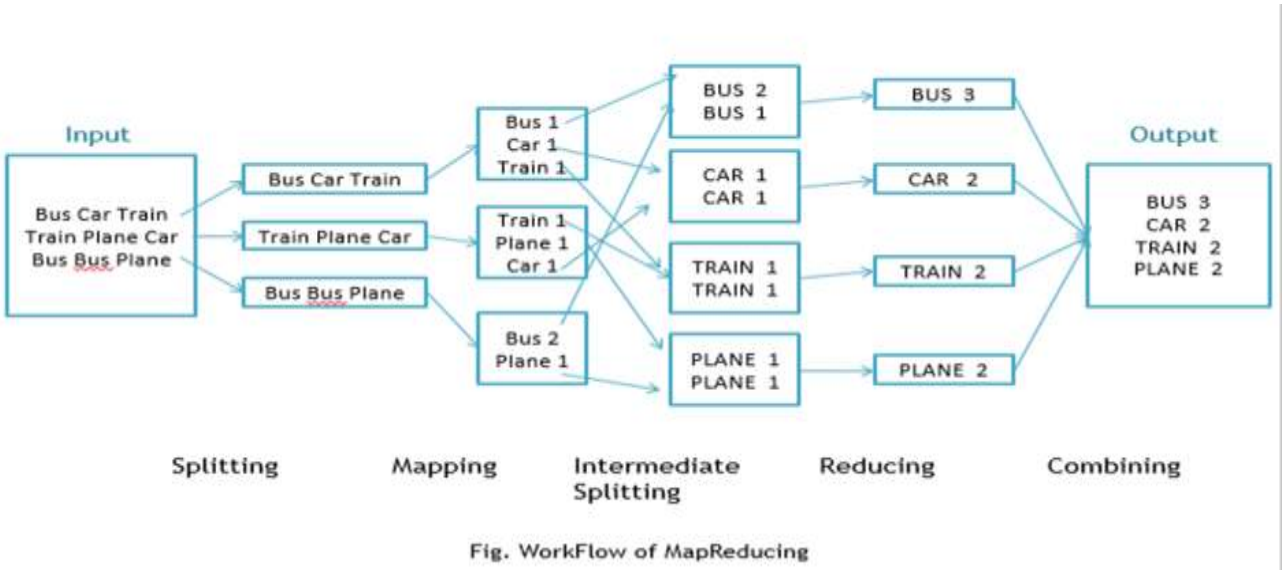


Fig. WorkFlow of MapReducing

Workflow of MapReduce consists of 5 steps:

1. Splitting – The splitting parameter can be anything, e.g. splitting by space, comma, semicolon, or even by a new line ('\n').
2. Mapping – as explained above.

3. Intermediate splitting – the entire process in parallel on different clusters. In order to group them in “Reduce Phase” the similar KEY data should be on the same cluster.

4. Reduce – it is nothing but mostly group by phase.

5. Combining – The last phase where all the data (individual result set from each cluster) is combined together to form a result.

1. Steps

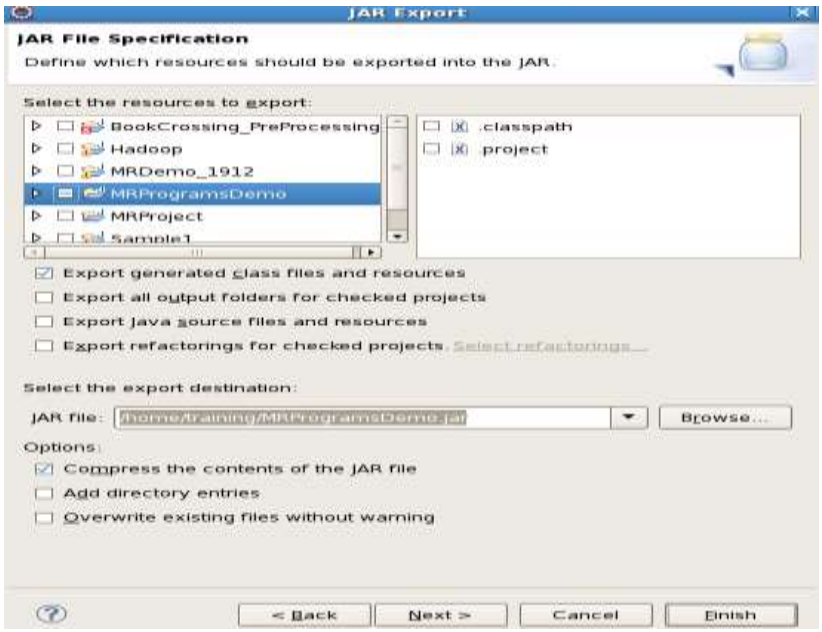
6. Open Eclipse> File > New > Java Project >(Name it – MRProgramsDemo) > Finish.

7. Right Click > New > Package (Name it - PackageDemo) > Finish.

8. Right Click on Package > New > Class (Name it - WordCount).

9. Add Following Reference Libraries:

1. Right Click on Project > Build Path> Add External
    1. /usr/lib/hadoop-0.20/hadoop-core.jar
    2. Usr/lib/hadoop-0.20/lib/Commons-cli-1.2.jar



Source Code:

```
package PackageDemo;
```

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class WordCount {
public static void main(String [] args) throws Exception
{
Configuration c=new Configuration();
String[] files=new GenericOptionsParser(c,args).getRemainingArgs();
Path input=new Path(files[0]);
Path output=new Path(files[1]);
Job j=new Job(c,"wordcount");
j.setJarByClass(WordCount.class);
j.setMapperClass(MapForWordCount.class);
j.setReducerClass(ReduceForWordCount.class);
j.setOutputKeyClass(Text.class);
j.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(j, input);
FileOutputFormat.setOutputPath(j, output);
System.exit(j.waitForCompletion(true)?0:1);
}
public static class MapForWordCount extends Mapper<LongWritable, Text, Text, IntWritable>{
```

```
public void map(LongWritable key, Text value, Context con) throws IOException,
InterruptedException
{
String line = value.toString();
String[] words=line.split(",");
for(String word: words )
{
    Text outputKey = new Text(word.toUpperCase().trim());
IntWritableoutputValue = new IntWritable(1);
con.write(outputKey, outputValue);
}
}
}
public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text, IntWritable>
{
public void reduce(Text word, Iterable<IntWritable> values, Context con) throws IOException,
InterruptedException
{
int sum = 0;
for(IntWritable value : values)
  {
  sum += value.get();
  }
con.write(word, new IntWritable(sum));
}
}
}
```

The above program consists of three classes:

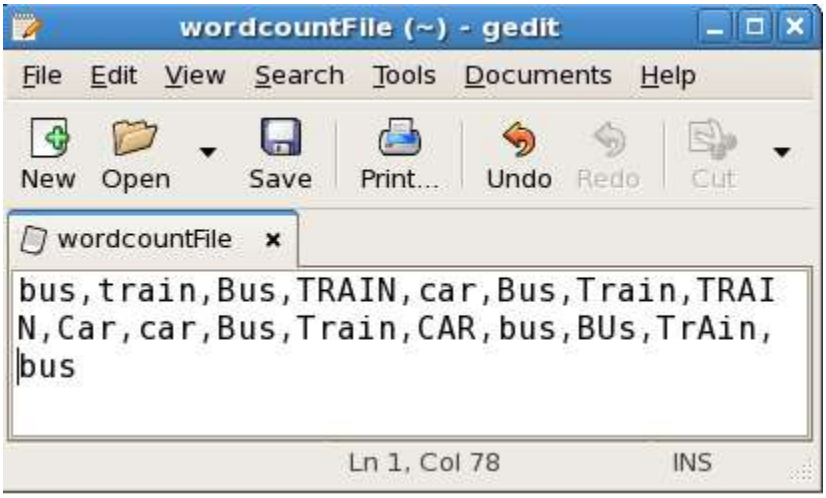
- Driver class (Public, void, static, or main; this is the entry point).

- The Map class which **extends** the public class Mapper<KEYIN,VALUEIN,KEYOUT,VALUEOUT> and implements the Map function.
- The Reduce class which extends the public class Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT> and implements the Reduce function.

10. Make a jar file

    Right Click on Project> Export> Select export destination as **Jar File** > next> Finish.

11. Take a text file and move it into HDFS format:



move this into Hadoop directly, open the terminal and enter the following commands:

[training@localhost~]$hadoop fs -putwordcountFilewordCountFile

Run the jar file:

[training@localhost~]$hadoop jar MRProgramsDemo.jar PackageDemo.WordCountwordCountFile MRDir1

Input / Output:

```
[training@localhost~]$hadoop fs -ls MRDir1
Found 3 items
-rw-r--r--   1 training supergroup          0 2016-02-23 03:36 /user/training/MRDir1/_SUCCESS
drwxr-xr-x   - training supergroup          0 2016-02-23 03:36 /user/training/MRDir1/_logs
-rw-r--r--   1 training supergroup         20 2016-02-23 03:36 /user/training/MRDir1/part-r-00000
[training@localhost~]$hadoop fs -cat MRDir1/part-r-00000
BUS     7
CAR     4
TRAIN   6
```

Additional Program

Experiment-II: AWS Application Load Balancer with Target Groups on EC2 Instances

Problem Statement: AWS Application Load Balancer with Target Groups on EC2 Instances

Description: Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, Lambda functions, and virtual appliances. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers four types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant

Step 1: Select a load balancer type

Elastic Load Balancing supports different types of load balancers. For this tutorial, you create a Classic Load Balancer.

To create a Classic Load Balancer

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the navigation bar, choose a Region for your load balancer. Be sure to select the same Region that you selected for your EC2 instances.
3. On the navigation pane, under LOAD BALANCING, choose Load Balancers.
4. Choose Create Load Balancer.
5. For Classic Load Balancer, choose Create.

Step 2: Define your load balancer
You must provide a basic configuration for your load balancer, such as a name, a network, and a listener.

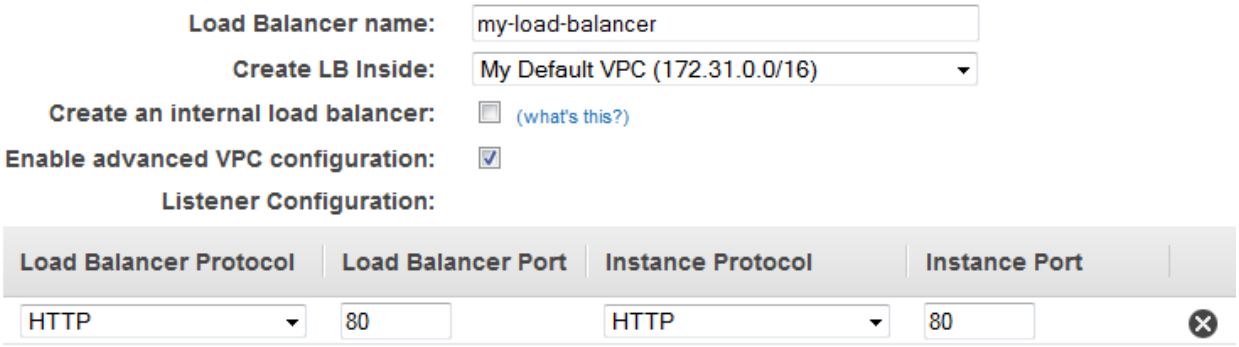
A listener is a process that checks for connection requests. It is configured with a protocol and a port for front-end (client to load balancer) connections and a protocol and a port for back-end (load balancer to instance) connections. In this tutorial, you configure a listener that accepts HTTP requests on port 80 and sends them to your instances on port 80 using HTTP.

To define your load balancer and listener

1. For Load Balancer name, type a name for your load balancer.

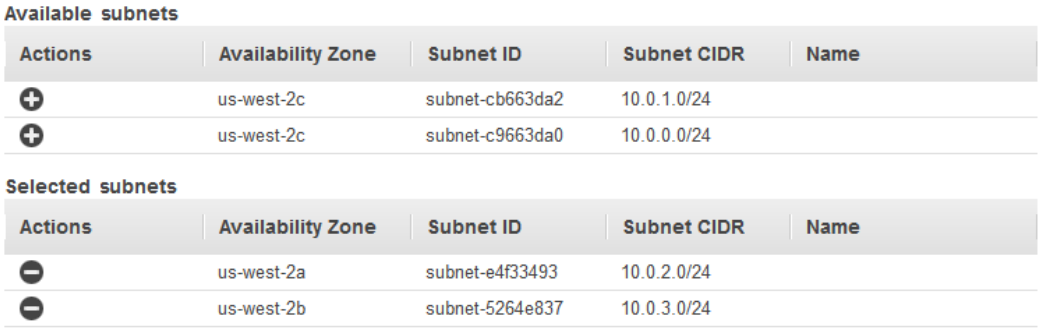
   The name of your Classic Load Balancer must be unique within your set of Classic Load Balancers for the region, can have a maximum of 32 characters, can contain only alphanumeric characters and hyphens, and must not begin or end with a hyphen.

2. For Create LB inside, select the same network that you selected for your instances: EC2-Classic or a specific VPC.
3. [Default VPC] If you selected a default VPC and would like to choose the subnets for your load balancer, select Enable advanced VPC configuration.
4. Leave the default listener configuration.



5. [EC2-VPC] For Available subnets, select at least one available public subnet using its add icon. The subnet is moved under Selected subnets. To improve the availability of your load balancer, select more than one public subnet.

   You can add at most one subnet per Availability Zone. If you select a subnet from an Availability Zone where there is already an selected subnet, this subnet replaces the currently selected subnet for the Availability Zone.



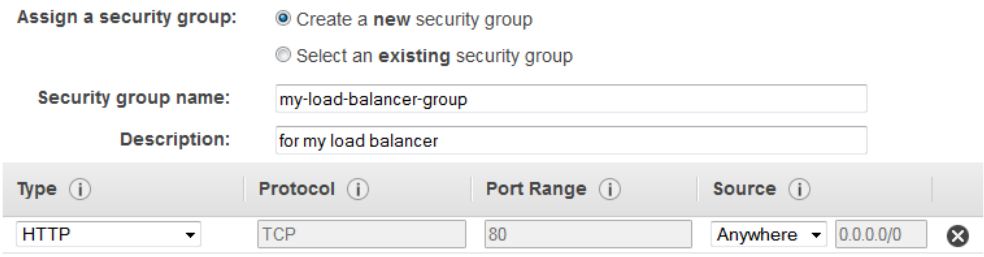
6. Choose Next: Assign Security Groups.

Step 3: Assign security groups to your load balancer in a VPC

If you selected a VPC as your network, you must assign your load balancer a security group that allows inbound traffic to the ports that you specified for your load balancer and the health checks for your load balancer.

To assign security group to your load balancer

1. On the Assign Security Groups page, select Create a new security group.
2. Type a name and description for your security group, or leave the default name and description. This new security group contains a rule that allows traffic to the port that you configured your load balancer to use.



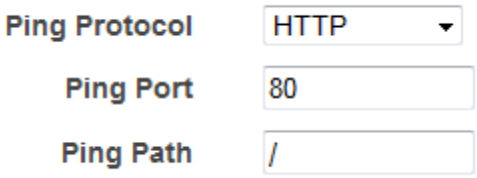
3. For this tutorial, you are not using a secure listener. Choose Next: Configure Health Check to continue to the next step.

Step 4: Configure health checks for your EC2 instances

Elastic Load Balancing automatically checks the health of the EC2 instances for your load balancer. If Elastic Load Balancing finds an unhealthy instance, it stops sending traffic to the instance and reroutes traffic to healthy instances. In this step, you customize the health checks for your load balancer.

To configure health checks for your instances

1. On the Configure Health Check page, leave Ping Protocol set to HTTP and Ping Port set to 80.
2. For Ping Path, replace the default value with a single forward slash ("/"). This tells Elastic Load Balancing to send health check queries to the default home page for your web server, such as index.html.



3. For Advanced Details, leave the default values.
4. Choose Next: Add EC2 Instances.

Step 5: Register EC2 instances with your load balancer

Your load balancer distributes traffic between the instances that are registered to it.

To register EC2 instances with your load balancer

1. On the Add EC2 Instances page, select the instances to register with your load balancer. Refer to step 9 for EC 2 instance creation
2. Leave cross-zone load balancing and connection draining enabled.
3. Choose Next: Add Tags.

Step 6: Tag your load balancer (optional)

You can tag your load balancer, or continue to the next step. Note that you can tag your load balancer later on; for more information, see Tag your Classic Load Balancer.

To add tags to your load balancer

1. On the Add Tags page, specify a key and a value for the tag.
2. To add another tag, choose Create Tag and specify a key and a value for the tag.
3. After you are finished adding tags, choose Review and Create.

Step 7: Create and verify your load balancer
Before you create the load balancer, review the settings that you selected. After creating the load balancer, you can verify that it's sending traffic to your EC2 instances.

To create and test your load balancer

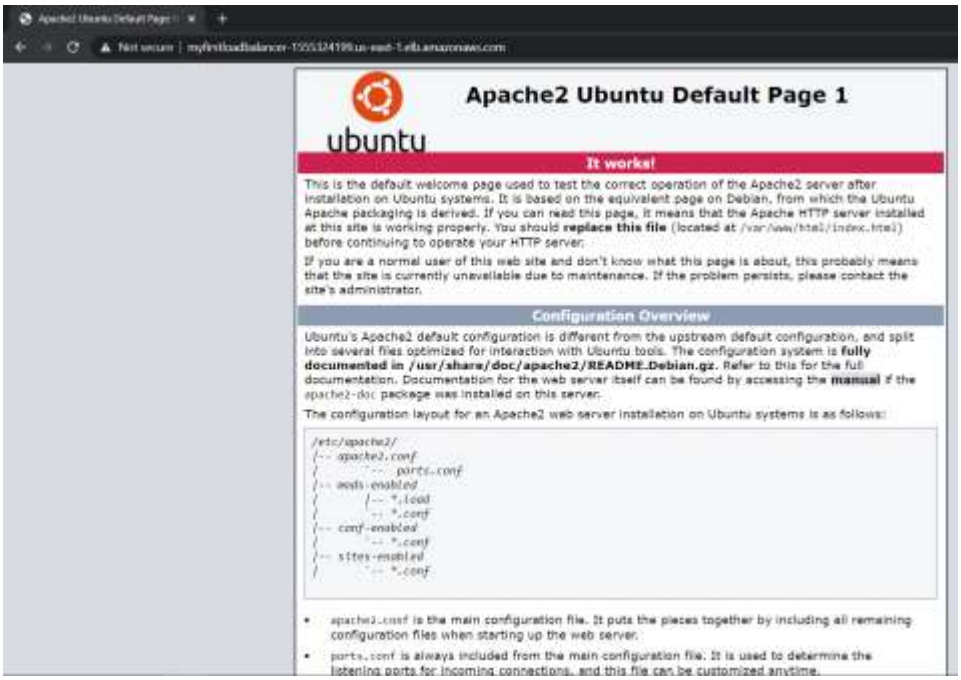
1. On the Review page, choose Create.

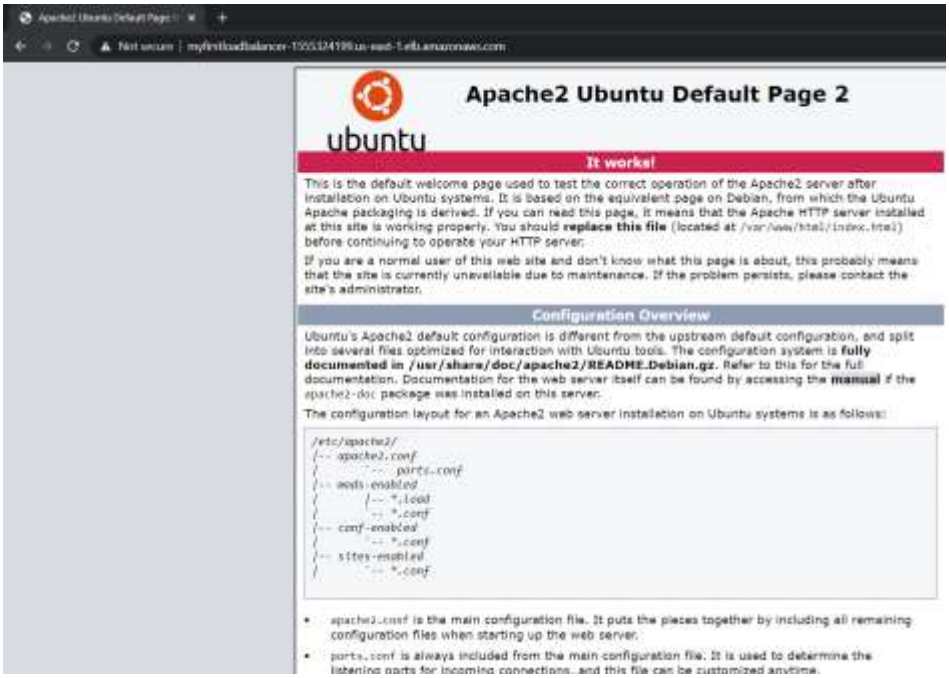
2. After you are notified that your load balancer was created, choose Close.

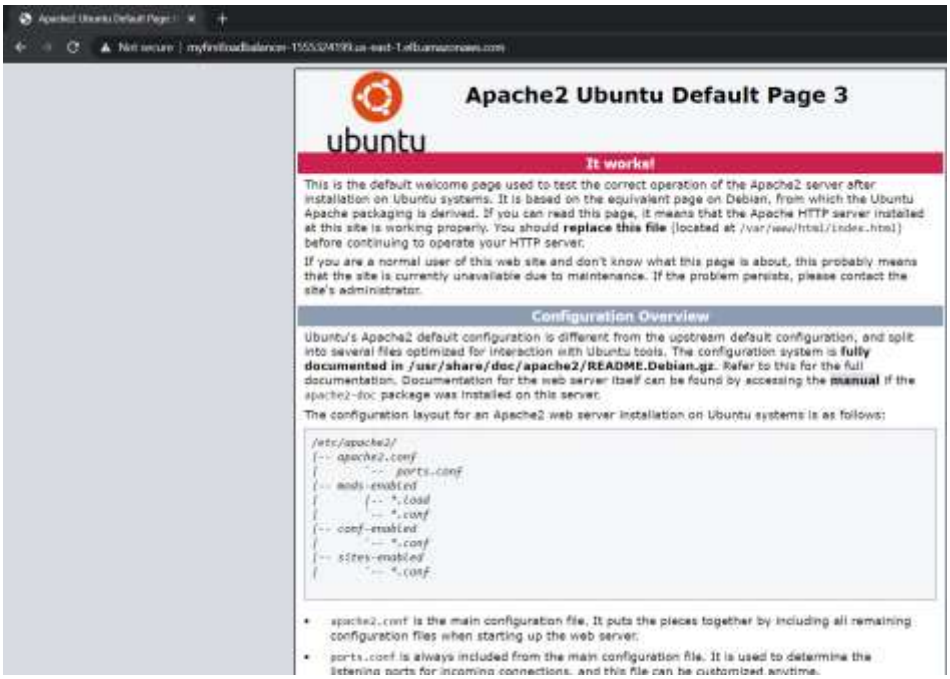
3. Select your new load balancer.

4. On the Description tab, check the Status row. If it indicates that some of your instances are not in service, its probably because they are still in the registration process. For more information, see Troubleshoot a Classic Load Balancer: Instance registration.

5. After at least one of your EC2 instances is in service, you can test your load balancer. Copy the string from DNS name (for example, my-load-balancer-1234567890.us-west-2.elb.amazonaws.com) and paste it into the address field of an internet-connected web browser. If your load balancer is working, you see the default page of your server.


Apache2 Ubuntu Default Page 1
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.


Apache2 Ubuntu Default Page 2
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.



Step 8: Delete your load balancer (optional)

As soon as your load balancer becomes available, you are billed for each hour or partial hour that you keep it running. When you no longer need a load balancer, you can delete it. As soon as the load balancer is deleted, you stop incurring charges for it. Note that deleting a load balancer does not affect the instances registered with the load balancer.

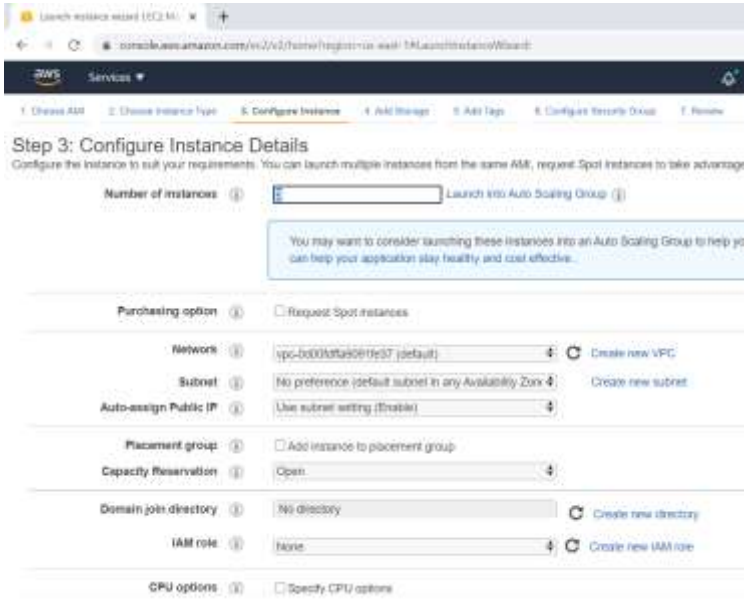
To delete your load balancer

1. If you have a CNAME record for your domain that points to your load balancer, point it to a new location and wait for the DNS change to take effect before deleting your load balancer.
2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
3. On the navigation pane, under LOAD BALANCING, choose Load Balancers.
4. Select the load balancer.
5. Choose Actions, Delete.
6. When prompted for confirmation, choose Yes, Delete.
7. (Optional) After you delete a load balancer, the EC2 instances associated with the load balancer continue to run, and you are billed for each hour or partial hour that you keep them running. For information about stopping or terminating your instances, see Stop and start your instance or Terminate your instance in the Amazon EC2 User Guide for Linux Instances.

Step 9: Creation of EC2 Instance

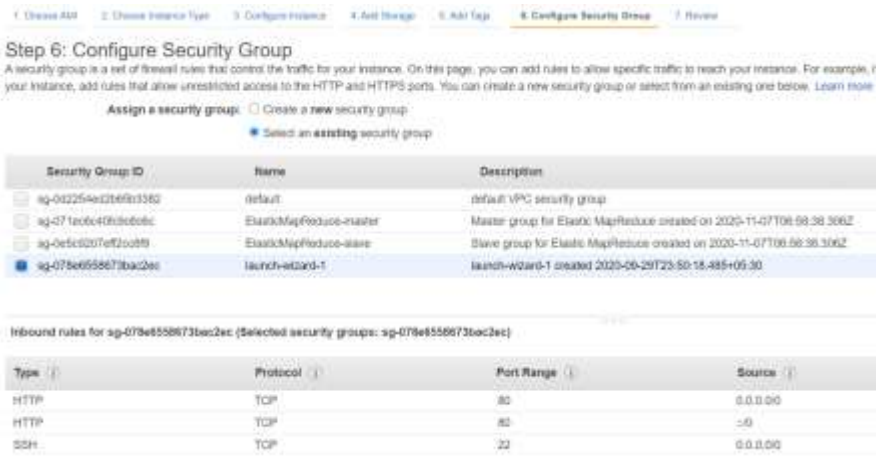
We have earlier created EC2 instances, and in the same way create few EC2 instances, atleast 2.

1. Navigate to EC2 dashboard and click on Create Instance
2. Select Ubuntu 18.04 LTS instance and click select
3. Select the free tier compute instance and click Configure Instance Details
4. Choose the number of instance as 3

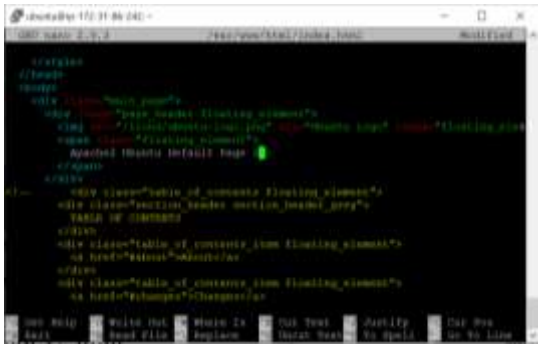


5. Scroll down and insert user data in the form of text with the command below. These commands are automatically executed during the preparation of the AMI. This will update the OS & install the apache2 server

6. Leave the storage, tags as default and now choose the security group that has SSH & HTTP ports open. If not create one with these rules and then click on review and launch



7. Choose a key pair to enable secure connection to the AMI's if needed using Putty. If there is no key available create one, and launch the AMI's.
8. Login to each of these AMI's and run the commands to install apache2 service
9. Now edit the index.html file to differentiate between the applications that's invoked by the Load balancer



10. Repeat the same for the other three AMI's as well