COSC349 ASSIGNMENT 2 REPORT

KIMBERLENE SHARMA

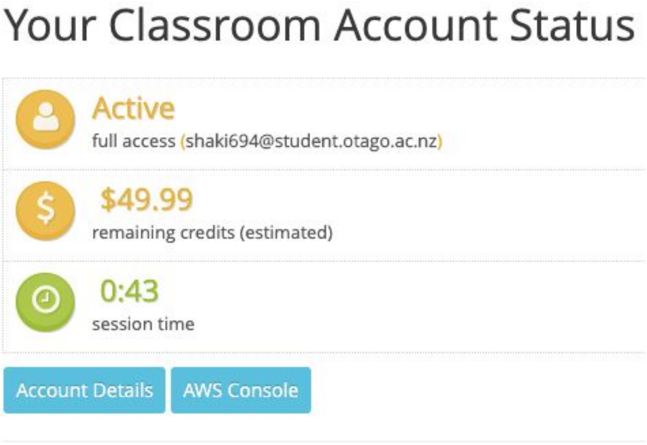
9559672

Introduction:

For the second assignment, I chose to extend my assignment 1 project. The one change I made in the interface was to make it a lot simpler since the assignment was based on the deployment and VMs and not the application itself. It is essentially a form where the user inputs notes and can see all the submitted forms in the second page which can be accessed using the link at the bottom. I had to fix my assignment 1 project before I could start this over but I am glad I was able to finish it. I chose to use AWS RDS for storage instead of the dbserver.

How I deployed my application:

To deploy my application, I had to make the credentials file in the path '~/.aws' by first using cd ~/.aws and then vim credentials commands.The vim command opens the vim editor and to edit the document you press I which puts it on insert mode. I pasted my credential information from the AWS workbench webpage. To save and exit , I simply had to press the esq button and typed :wq on the vim command line. The account credential information can be found if you click the Account Details button. Once pressed, a dialog opens up with credential information. I have provided some screenshots below for this.

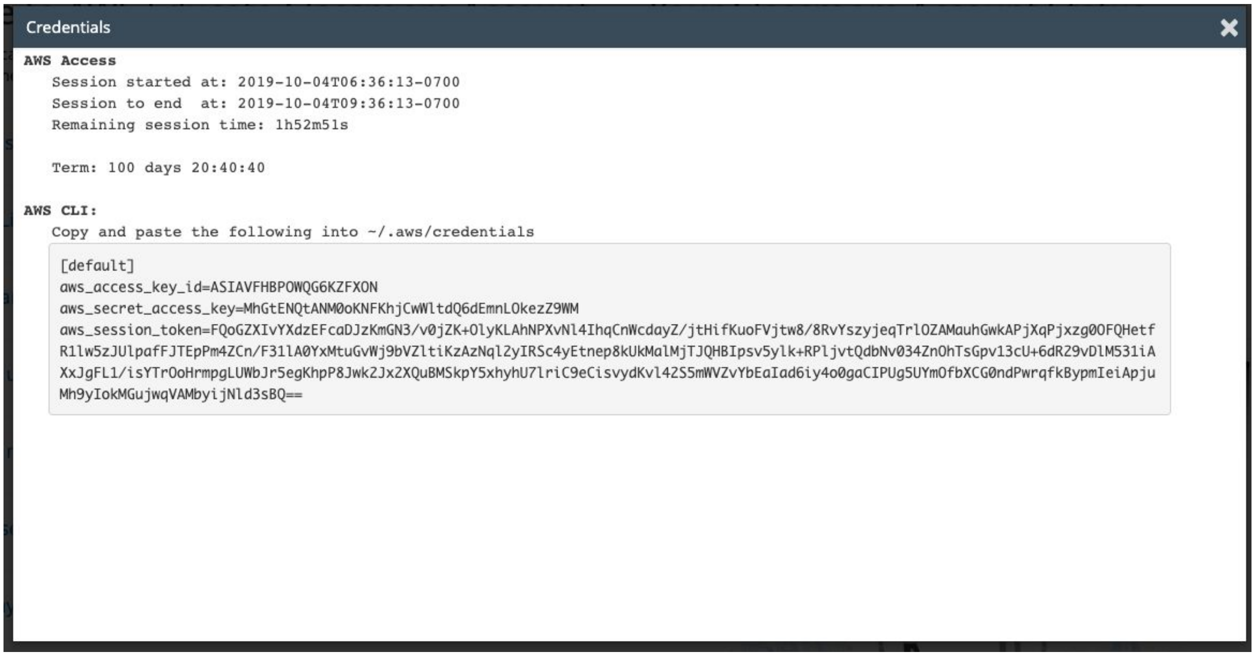


Your Classroom Account Status

Active
full access (shaki694@student.otago.ac.nz)

$49.99
remaining credits (estimated)

0:43
session time



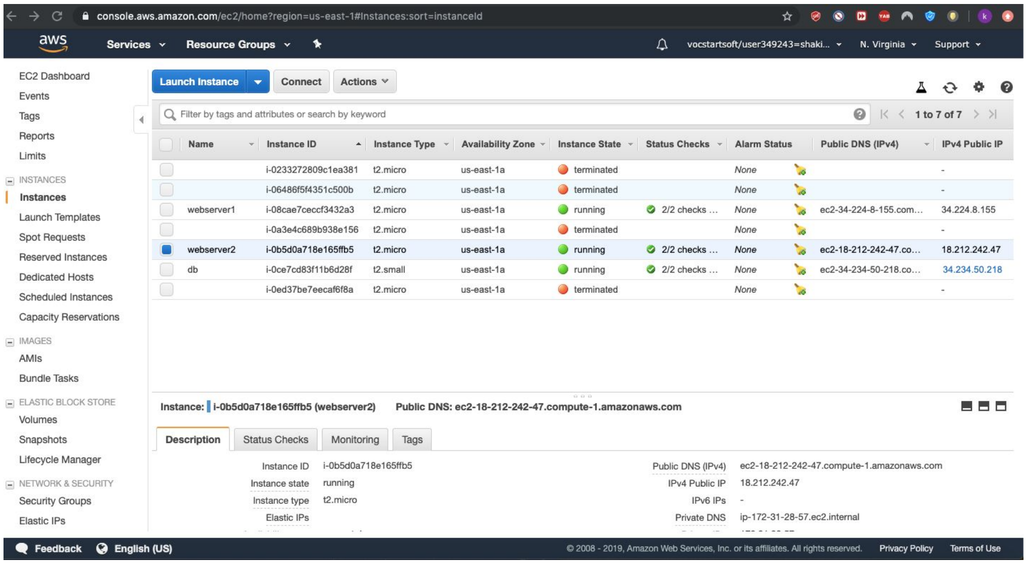
```
AWS Access
    Session started at: 2019-10-04T06:36:13-0700
    Session to end  at: 2019-10-04T09:36:13-0700
    Remaining session time: 1h52m51s

    Term: 100 days 20:40:40

AWS CLI:
    Copy and paste the following into ~/.aws/credentials
```

```
[default]
aws_access_key_id=ASIAVFHBPOWQG6KZFXON
aws_secret_access_key=MhGtENQtANM0oKNFKhjCwWltdQ6dEmnLOkezZ9WM
aws_session_token=FQoGZXIvYXdzEFcaDJzKmGN3/v0jZK+OlyKLAhNPXvNl4IhqCnWcdayZ/jtHifKuoFVjtw8/8RvYszyjeqTrlOZAMauhGwkAPjXqPjxzg0OFQHetf
R1lw5zJUlpafFJTEpPm4ZCn/F31lA0YxMtuGvWj9bVZltiKzAzNql2yIRSc4yEtnep8kUkMalMjTJQHBIpsv5ylk+RPljvtQdbNv034ZnOhTsGpv13cU+6dRZ9vDlM531iA
XxJgFL1/isYTrOoHrmpgLUWbJr5egKhpP8Jwk2Jx2XQuBMSkpY5xhyhU7lriC9eCisvydKvl42S5mWVZvYbEaIad6iy4o0gaCIPUg5UYmOfbXCG0ndPwrqfkBypmIeiApju
Mh9yIokMGujwqVAMbyijNld3sBQ==
```

I then changed my directory to point to where my Vagrantfile is and typed the following commands on the command line with the values from the credentials page

```
export AWS_ACCESS_KEY_ID=
export AWS_SECRET_ACCESS_KEY=
export AWS_SESSION_TOKEN=
```

With this being set, I hit vagrant up --provider=aws. This command creates two instances of my Webserver1 and Webserver2 virtual machines which can be seen here in the EC2 AWS console



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
□		i-0233272809c1ea381	t2.micro	us-east-1a	● terminated		None		-
□		i-06486f5f4351c500b	t2.micro	us-east-1a	● terminated		None		-
□	webserver1	i-08cae7ceccf3432a3	t2.micro	us-east-1a	● running	✓ 2/2 checks ...	None	ec2-34-224-8-155.com...	34.224.8.155
□		i-0a3e4c689b938e156	t2.micro	us-east-1a	● terminated		None		-
■	webserver2	i-0b5d0a718e165ffb5	t2.micro	us-east-1a	● running	✓ 2/2 checks ...	None	ec2-18-212-242-47.co...	18.212.242.47
□	db	i-0ce7cd83f11b6d28f	t2.small	us-east-1a	● running	✓ 2/2 checks ...	None	ec2-34-234-50-218.co...	34.234.50.218
□		i-0ed37be7eecaf6f8a	t2.micro	us-east-1a	● terminated		None		-

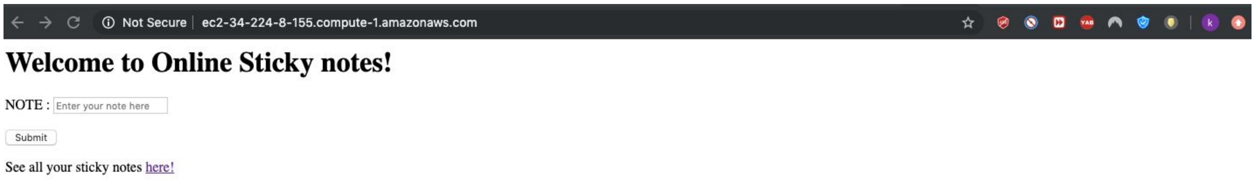
Instance: i-0b5d0a718e165ffb5 (webserver2) Public DNS: ec2-18-212-242-47.compute-1.amazonaws.com

Instance ID	i-0b5d0a718e165ffb5	Public DNS (IPv4)	ec2-18-212-242-47.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.212.242.47
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-28-57.ec2.internal

The db server instance is the RDS instance running in background. This application uses the RDS cloud service instead of the dbserver VM used in the last assignment.

The pages hosted by the webservers can be accessed by going on to the public DNS provided at the bottom right part of the page.

Go to http://ec2-34-224-8-155.compute-1.amazonaws.com/ to access the page hosted by webserver1 instance. See a screenshot of the first page with the public DNS on the URL address bar :

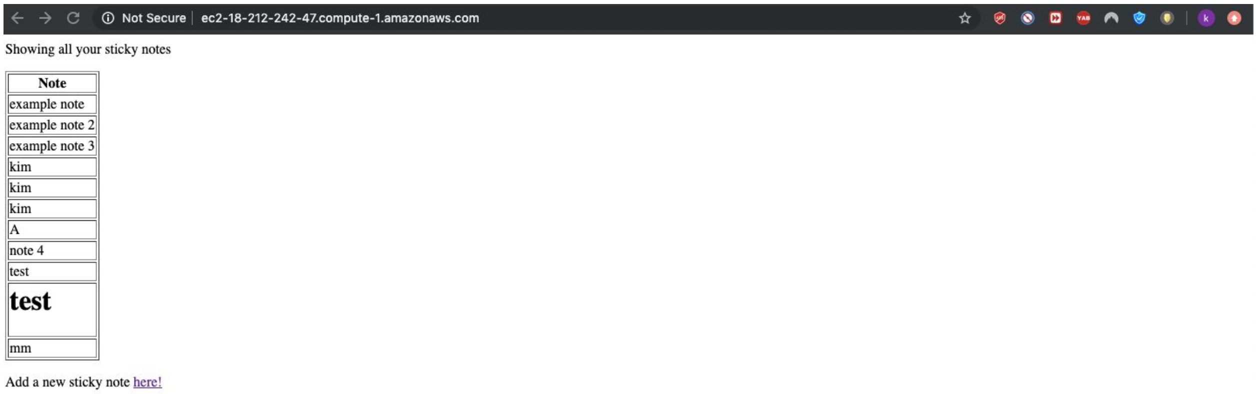


Welcome to Online Sticky notes!

NOTE : Enter your note here

See all your sticky notes here!

Go to http://ec2-18-212-242-47.compute-1.amazonaws.com/ to access the page hosted by the webserver2 instance. See a screenshot of the second page with the public DNS on the URL address bar:



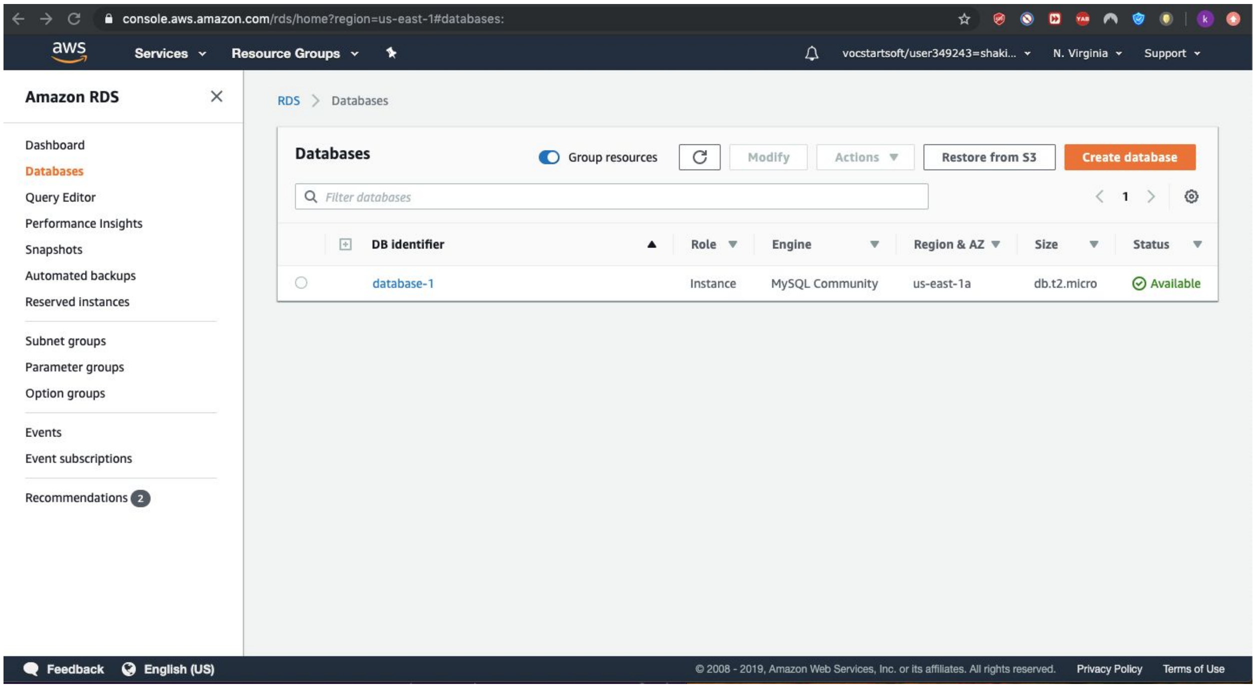
Not Secure | ec2-18-212-242-47.compute-1.amazonaws.com

Showing all your sticky notes

Note
example note
example note 2
example note 3
kim
kim
kim
A
note 4
test
test
mm

Add a new sticky note here!

Here is a screenshot of my RDS instance from the AWS console. It is important that it is running all the time so that the application functions properly

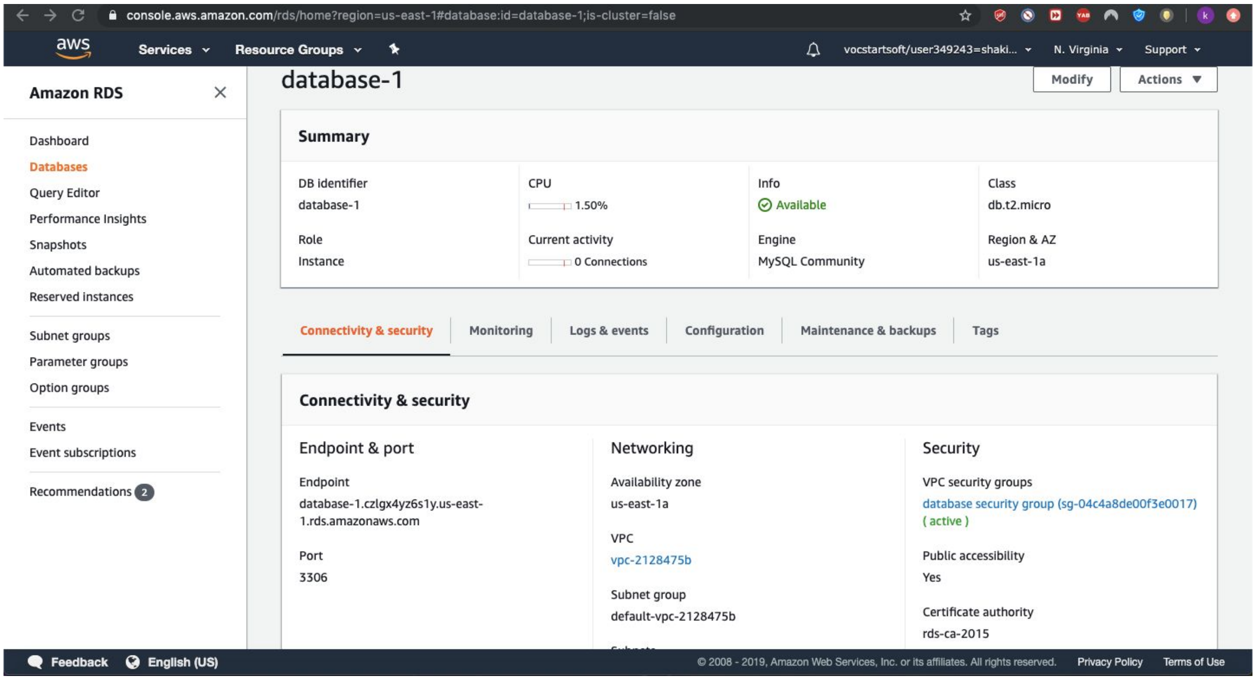


console.aws.amazon.com/rds/home?region=us-east-1#databases:

Databases

DB identifier	Role	Engine	Region & AZ	Size	Status
database-1	Instance	MySQL Community	us-east-1a	db.t2.micro	Available

When you click on the database name, it shows additional information about the instance Like the endpoint which I used to connect the database to the index.php pages and the port.



database-1

Summary

DB identifier: database-1
CPU: 1.50%
Info: Available
Class: db.t2.micro
Role: Instance
Current activity: 0 Connections
Engine: MySQL Community
Region & AZ: us-east-1a

Connectivity & security

Endpoint & port

Endpoint: database-1.czlgx4yz6s1y.us-east-1.rds.amazonaws.com

Port: 3306

Networking

Availability zone: us-east-1a

VPC: vpc-2128475b

Subnet group: default-vpc-2128475b

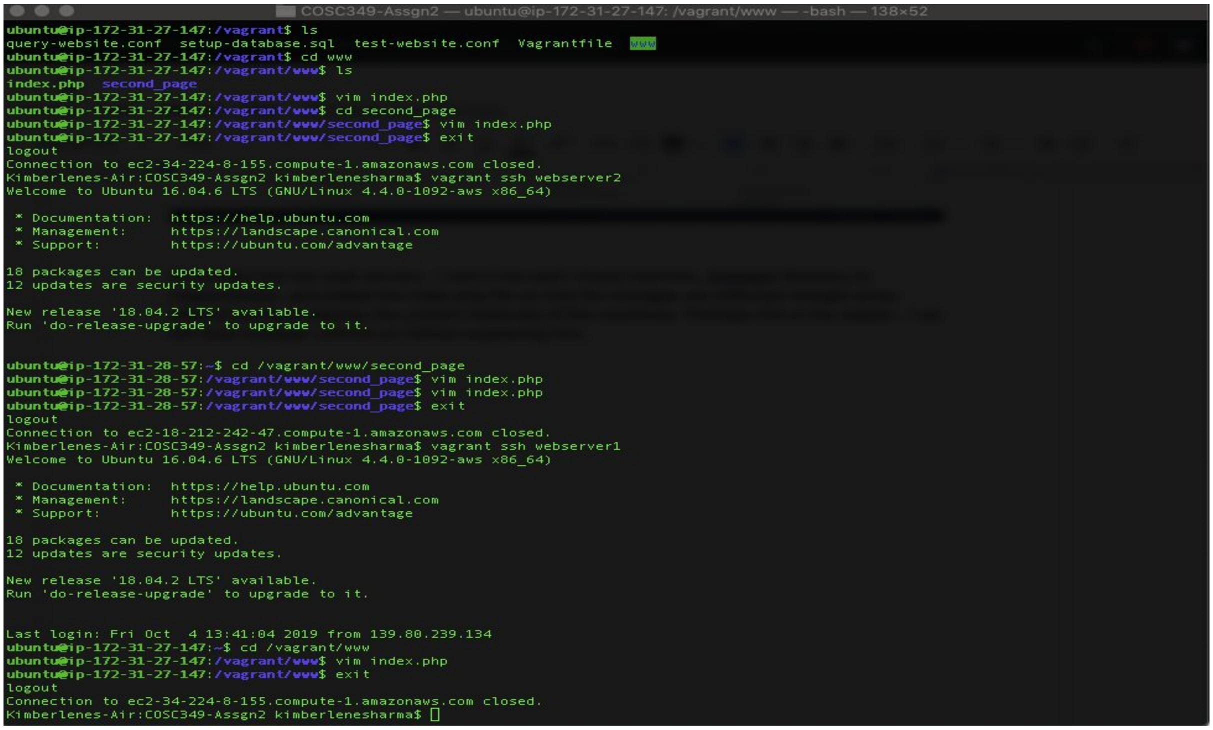
Security

VPC security groups: database security group (sg-04c4a8de00f3e0017) (active)

Public accessibility: Yes

Certificate authority: rds-ca-2015

To connect the two web servers , I ssh'd into each virtual machine, changed directory to /vagrant/www and edited the index.php file so that the changes are reflected straight away without having to destroy the current instances of the machines. Perhaps this is the reason , I do not have a proper commit on GitHub explaining this.


COSC349-Assgn2 — ubuntu@ip-172-31-27-147: /vagrant/www — -bash — 138×52

```
ubuntu@ip-172-31-27-147:/vagrant$ ls
query-website.conf  setup-database.sql  test-website.conf  Vagrantfile  www
ubuntu@ip-172-31-27-147:/vagrant$ cd www
ubuntu@ip-172-31-27-147:/vagrant/www$ ls
index.php  second_page
ubuntu@ip-172-31-27-147:/vagrant/www$ vim index.php
ubuntu@ip-172-31-27-147:/vagrant/www$ cd second_page
ubuntu@ip-172-31-27-147:/vagrant/www/second_page$ vim index.php
ubuntu@ip-172-31-27-147:/vagrant/www/second_page$ exit
logout
Connection to ec2-34-224-8-155.compute-1.amazonaws.com closed.
Kimberlenes-Air:COSC349-Assgn2 kimberlenesharma$ vagrant ssh webserver2
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1092-aws x86_64)

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

18 packages can be updated.
12 updates are security updates.

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


ubuntu@ip-172-31-28-57:~$ cd /vagrant/www/second_page
ubuntu@ip-172-31-28-57:/vagrant/www/second_page$ vim index.php
ubuntu@ip-172-31-28-57:/vagrant/www/second_page$ vim index.php
ubuntu@ip-172-31-28-57:/vagrant/www/second_page$ exit
logout
Connection to ec2-18-212-242-47.compute-1.amazonaws.com closed.
Kimberlenes-Air:COSC349-Assgn2 kimberlenesharma$ vagrant ssh webserver1
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1092-aws x86_64)

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

18 packages can be updated.
12 updates are security updates.

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


Last login: Fri Oct  4 13:41:04 2019 from 139.80.239.134
ubuntu@ip-172-31-27-147:~$ cd /vagrant/www
ubuntu@ip-172-31-27-147:/vagrant/www$ vim index.php
ubuntu@ip-172-31-27-147:/vagrant/www$ exit
logout
Connection to ec2-34-224-8-155.compute-1.amazonaws.com closed.
Kimberlenes-Air:COSC349-Assgn2 kimberlenesharma$ 
```

How to reach my application and interact with it:

To open my application simply type/paste this URL on the address bar of your favourite web browser: http://ec2-34-224-8-155.compute-1.amazonaws.com/
This opens up the Online Sticky Notes maker application. Type in a note in the box and press submit. The browser javascript dialog will let you know that your note has been added. Once you click ok, you can view the note you just saved by clicking the link below. This will take you to the page that shows all the saved notes.

Why and how I used RDS:

I found it easy to setup but had to counter a lot of problems to make the database work. RDS uses MySql which I used in the first assignment , thus implementing it was relatively easy. I followed the tutorial on the aws documentation. Before creating the database itself, I had to make a VPC security group for the RDS. I also made the RDS publicly accessible because after testing it with no public access, it did not run properly. I then accessed my webserver using ssh and then typed the commands sudo apt install mysql-server and sudo apt install mysql-client to set up the database. After this I accessed the database through the command:
mysql -u admin -h database-1.czlgx4yz6s1y.us-east-1.rds.amazonaws.com -P 3306 -p

I then Created the database using CREATE test and used it using USE test.
I proceeded to enter the contents of my setup-database.sql file where I create a table called note and enter values in it.

```
CREATE TABLE notes (
 note varchar(20) NOT NULL
);

INSERT INTO notes VALUES ('example note');
INSERT INTO notes VALUES ('example note 2');
INSERT INTO notes VALUES ('example note 3');
```

This then allowed me to add data to the database from my web servers which ultimately made my application work.

Conclusion:

For the most part of this assignment I used lab 9 as a reference along with the aws documentation tutorials. I encountered 403 permission not found errors which I solved by renaming all the web pages to index.php. For the 500 Internal server errors I used chmod 777 to resolve permission issues. Overall I am happy with the outcome of this assignment as I was able to successfully deploy a working application on cloud which uses one of the cloud services(RDS in our case).