

# COSC349 - Assignment 2

*Jack Royal - 6974461*

September/October 2019

This is the report section for the COSC349 Assignment 2. All work can be found on my Github

[https://github.com/jackroyalgit/COSC349\\_Assignment2.git](https://github.com/jackroyalgit/COSC349_Assignment2.git)

# 1 Your Application is running in the cloud

The application I decided to use was effectively identical to that of Assignment 1 apart from obviously the configuration of the Vagrantfile and the fact that it was being hosted in the cloud rather than locally. As you can see in my repo I did decide to continue using Vagrant for this assignment.

## 1.1 Your report should describe how you deployed your application

The deployment of my application involved a number of steps that included both adjusting my Vagrantfile and the configuration of the AWS cloud resources I decided to use (EC2 and RDS) along with slight updates to my PHP code to connect to the DB. The development of my application really began in Lab09 and this assignment was really just continuing on from that and putting all the pieces together. I will still however, lay out the process of deployment in a step by step manner below.

1. I used the process followed in Lab09 as a starting point as I had planned to use an EC2 instance previously and the Vagrantfile that came with Lab09 was a good basis on which to launch my application. Furthermore I already had a decent plan in my head having completed Lab09 previously so just had to stick all the pieces together effectively (already created key pair, security group etc).
2. I began by creating my key pair within the resources section of the AWS console and updated my Vagrantfile with the path of my keypair locally. I also added the instance type that I was using (t2.micro) to the Vagrantfile.
3. I then created an ssh security group which would allow me access into the VM. Following this I chose my availability zone and subnet ID. For my Amazon Machine Image I just chose Xenial Ubuntu image as suggested in the Lab.
4. From this point I could launch my VM instance and ran a few tests such as installing a web server and creating a security group for the web which all went smoothly, however when I tried to create a new directory and create my index.php file the web server wouldn't recognise it and

Apache would give me a 503 error. To fix this I effectively brute forced it which if I was to redo this assignment would be one of the things I changed and done it in a more elegant fashion in terms of permissions.

5. At this point I was ready to set up my database instance, for this I chose to use an AWS RDS instance specifically a MySQL instance in order to maintain consistency with Assignment 1. To create it I loosely followed the standard tutorial but overall it was a pretty straightforward process to get it set up. I again chose a t2.micro instance. I also added a database security group. From this I was able to connect into the DB through my VM and MySQL and loaded the SQL script for my application.
6. Now that I knew I could connect to my database instance all I had to do was modify my PHP script from Assignment 1 to use the credentials for my cloud based database and everything was almost identical to Assignment 1 except that everything was in the cloud
7. I then added my second VM (queryserver) which was straightforward and updated the credentials in the PHP file and my application was effectively finished.

## **1.2 Your report should describe how to reach your application in the cloud, and what a user can do easily to interact with it.**

My application is identical to that of Assignment 1 and is extremely easy to use. To get to the application use the link <http://ec2-100-26-142-112.compute-1.amazonaws.com/> and you can fill out the survey or view the results (as of writing this report my AWS Educate account only has 15 dollars left so could potentially run out before it is marked).

**1.3** Your report should include brief evidence—such as screen-captures with explanatory text—that show your application being used in the cloud.

[illegible]

Figure 1: Initial survey page

## Company X - User Survey

### Company Employees

First Name

Last

First Name

Gender

Job position

Age

Residence (Full)

City

Residence (Phone)

Address

Residence (State)

Residence (Country)

Submit

NOTE: On your mobile phone, the results are shown in the app.

Figure 2: Filling out the survey

[illegible]

Figure 3: Survey submitted

Recent Databases

Database "C"

1

Click to view the list of all databases managed by Amazon

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public IP Address	Private IP Address	Rep Name	Modification
db-1	i-1234567890123456	m4.xlarge	us-east-1a	available	2 checks passed	OK	10.0.0.100	10.0.0.100	db-1	created
db-2	i-0987654321098765	m4.xlarge	us-east-1b	available	2 checks passed	OK	10.0.0.200	10.0.0.200	db-2	created
db-3	i-1111111111111111	m4.xlarge	us-east-1c	available	2 checks passed	OK	10.0.0.300	10.0.0.300	db-3	created

Figure 5: EC2 Instances

[View](#)
[Edit](#)
[Delete](#)
[Add new](#)

## database-349

---

Summary

OS number database-349	CPU <div><div></div><div>100%</div></div> 1.2GHz	info <div><div></div><div>available</div></div>	Date @12 noon
Role Instance	Current activity <div><div></div><div>100%</div></div> Idle	Engine MySQL Community	Region & AZ us-east-1

---

[Connectivity & security](#)
[Monitoring](#)
[Logs & events](#)
[Configuration](#)
[Maintenance & backup](#)
[Tags](#)

---

Connectivity & security

Endpoint & port	Networking	Security
Endpoint database-349.mysql.amazonaws.com	Availability zones us-east-1c	VPC security groups [id=sg-047f0e37f16331633] (active)
Port 3306	VPC vpc-fd4af262	Public accessibility No
	Subnet group db-subnet-group	Certificate authority Amazon
	Subnets subnet-fd4af262 subnet-a0cd49f1 subnet-e0cd49f1	ssl-ca-2015 certificate-authority-date Mar 01, 2020

Figure 6: DB Instance

## **2 A cloud service is used in your application (beyond using the cloud to create your VMs)**

### **2.1 Your report describes your choice of service and how it is used. (Example services include Amazon S3 or a cloud database API.)**

The cloud service is used (beyond using EC2 to launch my VMs) was the AWS RDS solution for my database. I used this as it seemed like the most straightforward database solution that Amazon provided and there was lots of good documentation provided not only by Amazon but also from external sources. The database was a MySQL database which is what I used in my first assignment. This made it extremely simple to connect up to my PHP scripts as the only thing I had to change was the credentials in terms of the host and user of the database.

### **2.2 Your report explains the design of your application and how the VMs and other cloud APIs are used**

I have explained above how I used a cloud based database system. For my VMs I used EC2 instances as it allowed me to launch the VMs from Vagrant which I was very comfortable with as I had used it in the first assignment. The EC2 instances were also convenient as I had used them in Lab09 and it was just a natural progression to continue that work in the assignment.

### **2.3 References**

- David Eysers, Amazon EC2 via Vagrant, (2019), Lab Document, <https://hackmd.io/@dme26/Sknc7cXIS>
- AWS, Create an RDS Instance, (2019), Tutorial, [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_Tutorials.WebServerDB.CreateDBInstance.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html)