COMP5349 Cloud Computing

Reenal Raina Pereira(SID 520112117)

May 18 2023

Contents

1 Summary of the solution

1.1 Brief overview of the existing design and architecture:

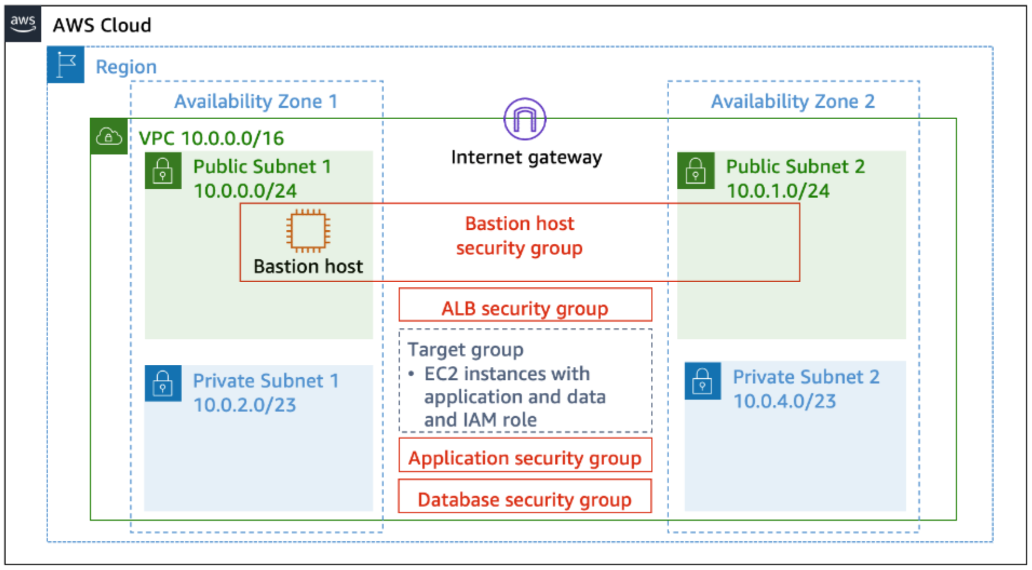


Figure 1: Existing environment diagram

After an initial analysis of the existing environment and diagram, here's the list of things provided. A Virtual Private Network (VPC) namely 'Example VPC' along with 4 subnets. Of which 2 are private and 2 are public subnets 'Private subnet 1', 'Private subnet 2' and 'Public subnet 1', 'Public subnet 2'. Among these 'Private subnet 1' and 'Public subnet 1' are hosted in Availability Zone 1 and 'Private subnet 2' and 'Public subnet 2' are hosted in Availability Zone 2. An EC2 instance 'Bastion host' is located in Availability Zone 1. Through Bastion Host Security Group it is available in both public subnets. An internet gateway namely 'Example IGW' helps connect 'Bastion Host' to the local system. The route table is connected to 'Example VPC' as the Public Route table and Private Route table with 2 subnets. 'Example VPC' is available in both availability zones. There are 4 security groups namely Inventory-App, Bastion-SG, Example-DBSG, and ALBSG.

1.2 Sequential design process:

In the beginning, an Application Load Balancer 'RPELB' was created for the high availability of the website and this was mapped to the 'Example VPC' and connected to public subnets with the existing security group 'ALBSG' which is connected to a listener. A listener is present to check incoming traffic on the load balancer's IP address. To ensure the website was accessible all the time, it was configured with both Availability zones in Public Subnet 1 and Public Subnet 2.

For monitoring the application and automatically adjusting the capacity to maintain steady, predictable performance at the lowest possible cost, Auto Scaling Group was created namely 'MyAutoScalingGroup' using the existing launch template. In the network configuration, this Auto Scaling

Group was connected to Private Subnet 1 and Private Subnet 2 to an existing load balancer 'RPELB' for high availability, scalability and efficient distribution of incoming traffic.

In order to have a minimum capacity of 2 EC2 web servers over both availability zones, the minimum capacity, desired capacity, and maximum capacity are set to 2, 2, and 4, respectively. This configuration ensures that there will always be 2 EC2 web servers all the time and this will allow for auto-scaling as well as handling any incoming traffic load.

Later, two NAT gateways were created in each of the Public Subnets. A route table was also created 'private route table 2' with subnet association configured to Private Subnet 2. This setup will help to direct traffic to one of the NAT gateways. Furthermore, in the existing 'Private Route Table 1', the subnet association was modified to Private Subnet 1. A new route was added for the second NAT gateway. Thus, both EC2 servers gained access to the internet. This way both EC2 servers placed in private subnets gained internet access through their own NAT gateways.

In order to query through the DB, databases have to be connected to the website. First, a DB subnet group 'dbsubnetgroup' was created on 'Example VPC' and connected to both private subnets and used 'us-east-1a' and 'us-east-1b' availability zones. After the DB subnet group, MySQL engine was used in Dev/Test environment and deployed in the 'Multi-AZ DB' instance as this will create a primary DB instance in private subnet 1 and a standby DB instance in private subnet 2. This helps in providing high availability and data redundancy. This instance is configured in t3.micro with 20GB of General purpose SSD (gp2) storage. To have secure administrative access to the DB, the 'admin' value was assigned to the Master username with a random password. Also, for the 'database authentication' section 'password authentication' was selected for secure administrative access. For the appropriate incoming traffic, 'Example-DB' was selected as the security group. After the successful creation of the database, the RDS endpoint can be retrieved.

As per the solution requirement, the below parameters were added in Parameter Store in the AWS Systems Manager in order for the PHP application to connect to the database,:

- /example/endpoint
- /example/username
- /example/password
- /example/database

In order to provide access to the SSH connection, inbound rules for the 'Inventory-App' security group was modified. These changes allowed secure SSH access for administrators with the 'Bastion' host.

Through the Bastion instance, after connecting to Amazon Linux 2 AMI, the private key was added which was provided in the project. Then the 'Countrydatadump.sql' file was imported to 'exampledb'. For a successful connection test, a query was executed on the website and table data was displayed. Thus the system was working as per expectation.

After following all the solution requirements, the below environment was developed.

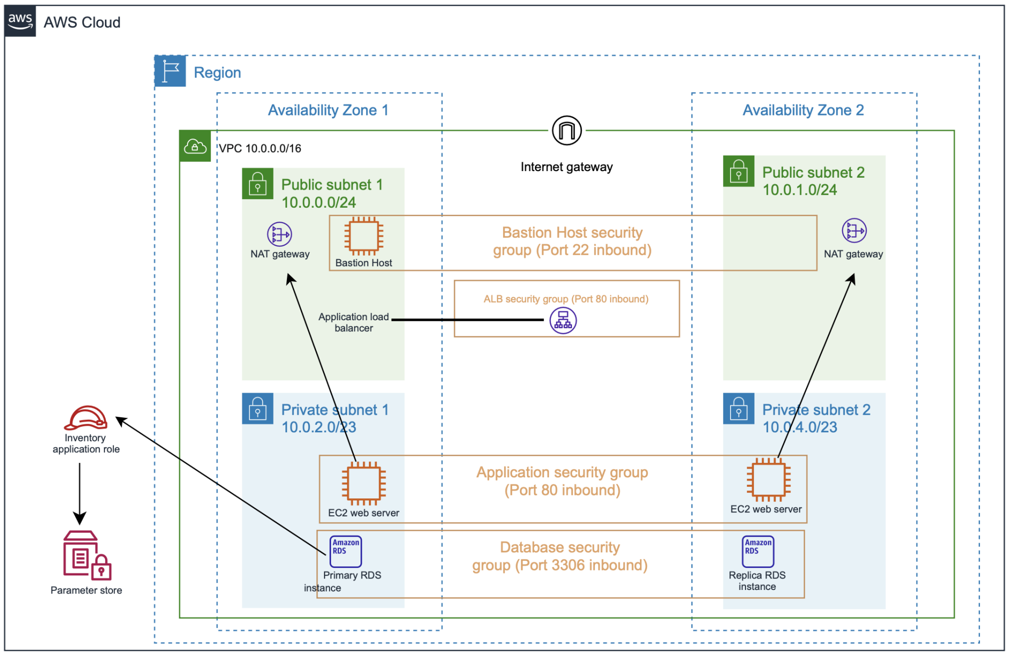
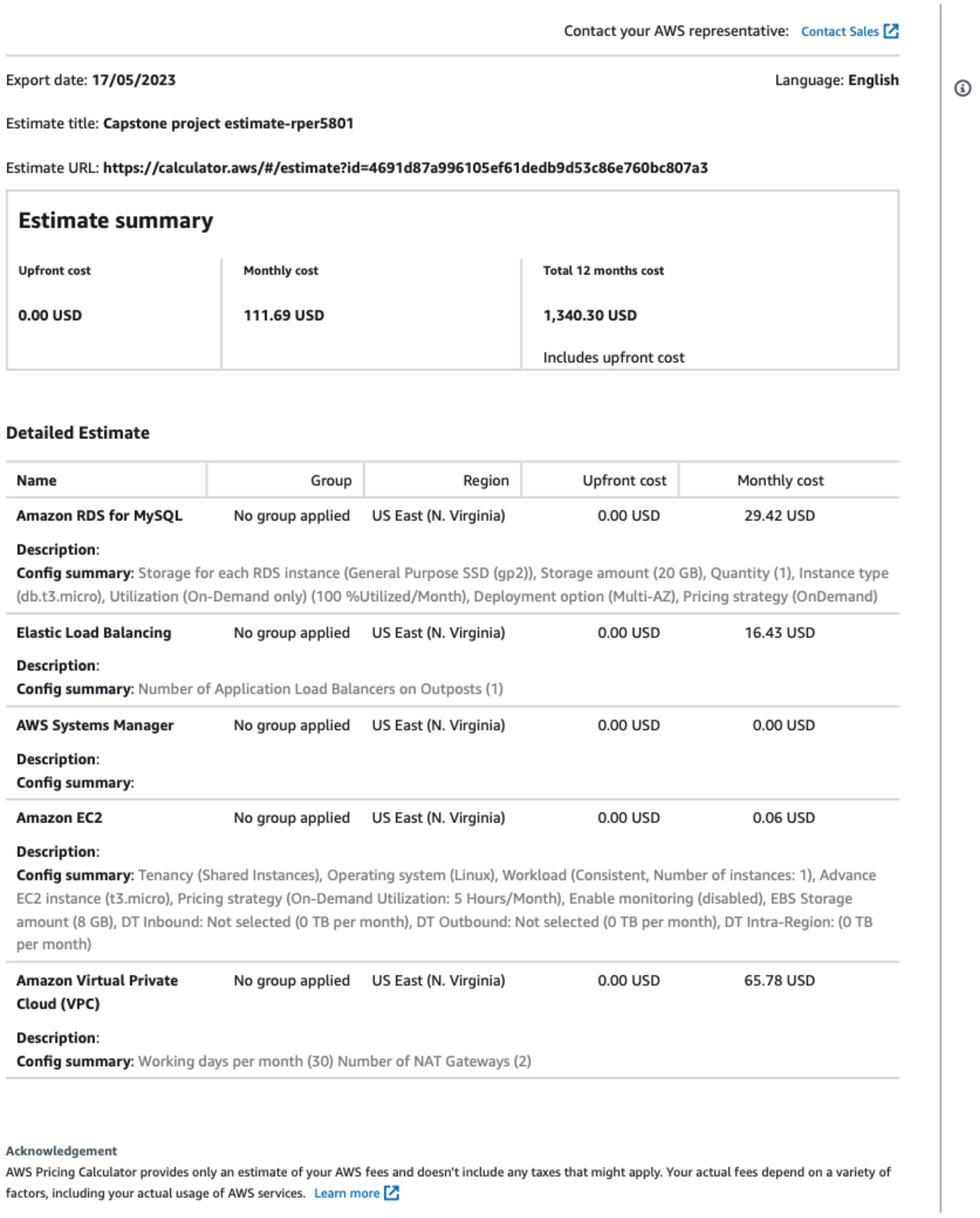


Figure 2: Solution environment diagram

2 Cost estimation

The below image is the estimated average cost of the overall architecture. This cost could be changed as it depends on various factors like traffic, the inclusion of additional services, etc. As per the AWS pricing calculator, the monthly estimated cost is $111.64 and the total 12 months cost is $1339.68



Contact your AWS representative: Contact Sales

Export date: **17/05/2023**

Language: **English**

Estimate title: **Capstone project estimate-rper5801**

Estimate URL: **https://calculator.aws/#/estimate?id=4691d87a996105ef61dedb9d53c86e760bc807a3**

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	111.69 USD	1,340.30 USD Includes upfront cost

Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon RDS for MySQL	No group applied	US East (N. Virginia)	0.00 USD	29.42 USD
Elastic Load Balancing	No group applied	US East (N. Virginia)	0.00 USD	16.43 USD
AWS Systems Manager	No group applied	US East (N. Virginia)	0.00 USD	0.00 USD
Amazon EC2	No group applied	US East (N. Virginia)	0.00 USD	0.06 USD
Amazon Virtual Private Cloud (VPC)	No group applied	US East (N. Virginia)	0.00 USD	65.78 USD

Amazon RDS for MySQL
Description:
Config summary: Storage for each RDS instance (General Purpose SSD (gp2)), Storage amount (20 GB), Quantity (1), Instance type (db.t3.micro), Utilization (On-Demand only) (100 %Utilized/Month), Deployment option (Multi-AZ), Pricing strategy (OnDemand)

Elastic Load Balancing
Description:
Config summary: Number of Application Load Balancers on Outposts (1)

AWS Systems Manager
Description:
Config summary:

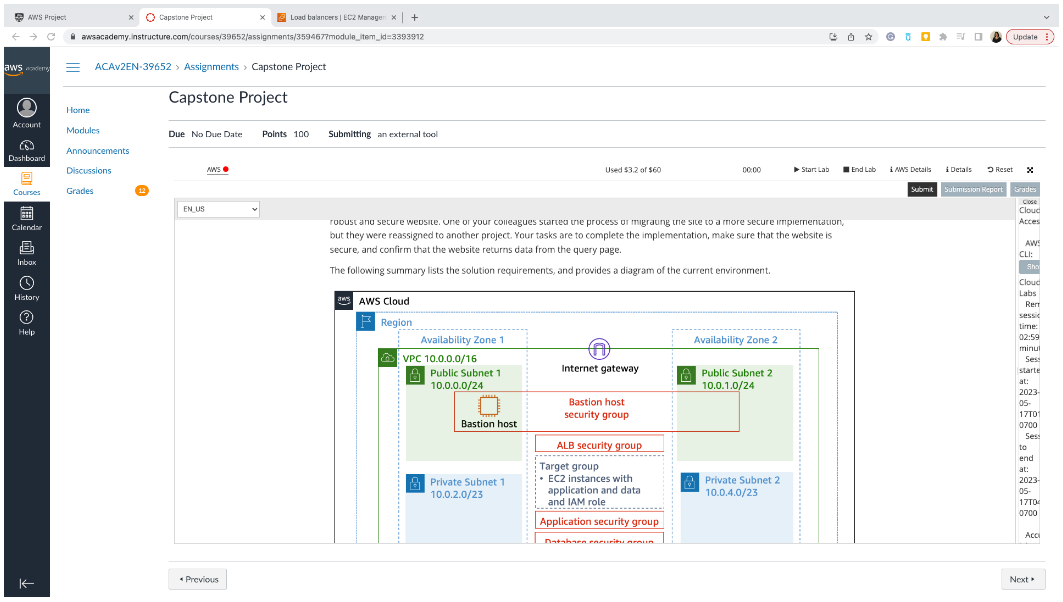
Amazon EC2
Description:
Config summary: Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 1), Advance EC2 instance (t3.micro), Pricing strategy (On-Demand Utilization: 5 Hours/Month), Enable monitoring (disabled), EBS Storage amount (8 GB), DT Inbound: Not selected (0 TB per month), DT Outbound: Not selected (0 TB per month), DT Intra-Region: (0 TB per month)

Amazon Virtual Private Cloud (VPC)
Description:
Config summary: Working days per month (30) Number of NAT Gateways (2)

Acknowledgement
AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services. Learn more

Figure 3: Estimated average cost

Before implementation of the solution the initial cost was \$0.4. After the implementation of the solution, \$3.2 was the estimated cost generated after approximately 24 hrs. So if we deduct the initial cost from implementation cost, the solution cost will be = \$2.8 To calculate the approximate cost for


AWS Project | Capstone Project | Load balancers | EC2 Managem

awsacademy.instructure.com/courses/39652/assignments/359467?module_item_id=3393912

ACAv2EN-39652 › Assignments › Capstone Project

Capstone Project

Due No Due Date **Points** 100 **Submitting** an external tool

AWS Used \$3.2 of \$60 00:00 Start Lab End Lab AWS Details Details Reset

robust and secure website. One of your colleagues started the process of migrating the site to a more secure implementation, but they were reassigned to another project. Your tasks are to complete the implementation, make sure that the website is secure, and confirm that the website returns data from the query page.

The following summary lists the solution requirements, and provides a diagram of the current environment.

Figure 4: actual solution cost

a month (considering 30 days in a month), \$2.8 X 30 = \$84. This means the cost of the solution deployed is \$84/month and \$1008/year. We see that there is a difference in the actual and estimated cost of \$27.69/month and \$332.3 over a period of a year and this difference is under the acceptable range for a website. The potential reason for this could be: Certain assumptions made as a solution about expected usage patterns could be changed and frequent unexpected spikes in the traffic could possibly lead to higher costs than estimated.

*Here \$ indicates USD.

3 Demonstration Plan

- **Part 1 – Introduction**
  Start: 0:00:00 End: 0:00:08 Duration: 08 seconds
- **Part 2 - Demonstrating VPC**
  Start: 0:00:09 End: 0:00:29 Duration: 20 seconds
- **Part 3 – Inspecting Security Groups and Inbound/Outbound Rules**
  Start: 0:00:30 End: 0:02:53 Duration: 2 minutes 23 seconds
- **Part 4 – Inspecting EC2 Instances**
  Start: 0:02:54 End: 0:04:27 Duration: 1 minute 33 seconds
- **Part 5 – Inspecting RDS Database**
  Start: 0:04:28 End: 0:04:46 Duration: 1 minute 33 seconds
- **Part 6 – Displaying basic website functioning**
  Start: 0:04:47 End: 0:05:25 Duration: 38 seconds
- **Part 7 – Accessing one of the EC2 instances via the bastion host using SSH and updating the database content**
  Start: 0:05:26 End: 0:08:05 Duration: 2 minutes 39 seconds
- **Part 8 – Demonstrating auto-scaling functionality which will start new instances in availability zones.**
  Start: 0:08:06 End: 0:12:07 Duration: 4 minutes 01 seconds
  (this is a lengthy section as this includes waiting time to start the instances in 2 availability zones.)