––Project Assessment

Unit code, name and release number

ITNET313A

Qualification/Course code, name and release number

Cloud Computing (Bachelor of Cybersecurity)

Student details

Student number

801301053

Student name

Murilo de Grandi

Assessment Declaration

- This assessment is my original work and no part of it has been copied from any other source except where due acknowledgement is made.
- No part of this assessment has been written for me by any other person except where such collaboration has been authorised by the assessor concerned.
- I understand that plagiarism is the presentation of the work, idea or creation of another person as though it is your own. Plagiarism occurs when the origin of the material used is not appropriately cited. No part of this assessment is plagiarised.

Student signature and Date

[illegible] 22/05/2022

A5 - Project Implementation

ITNET313A – Cloud Computing

Murilo de Grandi - 801301053

Date: 22 May 2022

Table of Contents

EXECUTIVE SUMMARY

The new design of the Example Social Research website has been approved by Shirley Rodriguez and the organisation. This document explains the new website implementation process and how the new solution will deliver a more robust and cost-effective website.

The new website infrastructure was upgraded from a monolithic web application to a two-tier architecture that leverages AWS services such as Elastic Load Balancer (ELB), Application Auto Scaling Group and Multi-AZ RDS database.

The new website solution was successfully implemented and met all the project requirements

1. Introduction

The Example Social Research website is an important source of information for social science researchers. This document aims at presenting and explaining the implementation process that was carried out for the creation of the new website according to the new design that was presented and approved by Shirley Rodriguez and the organisation.

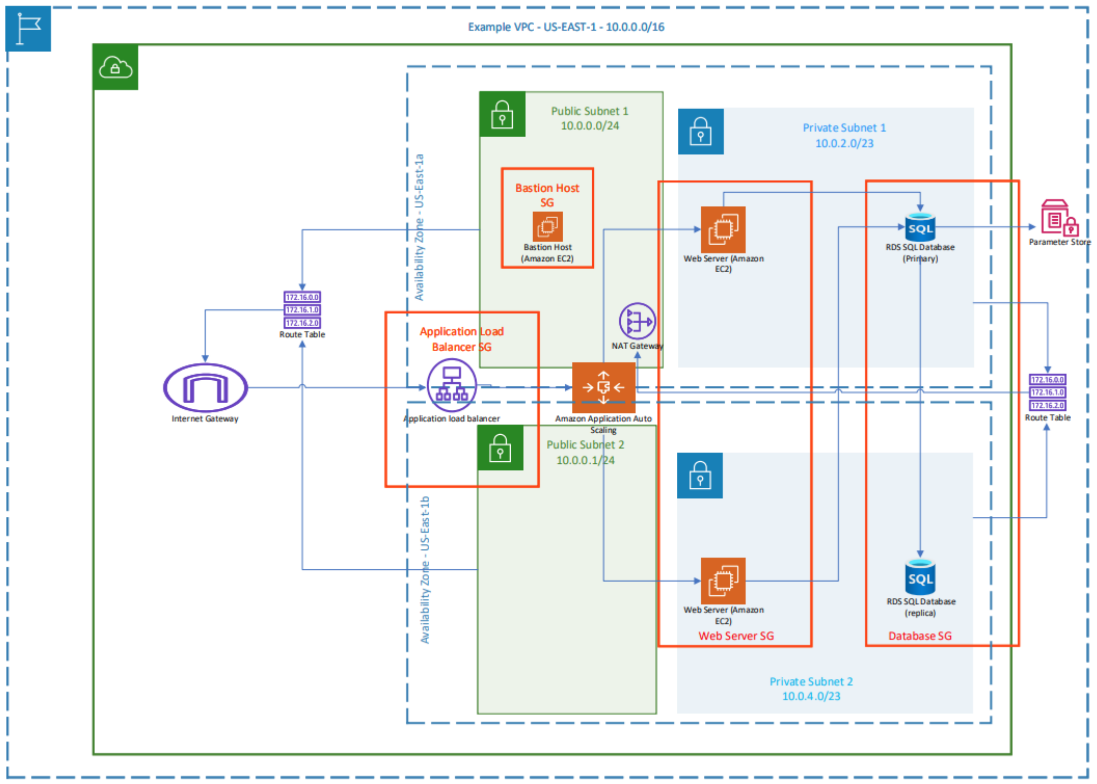
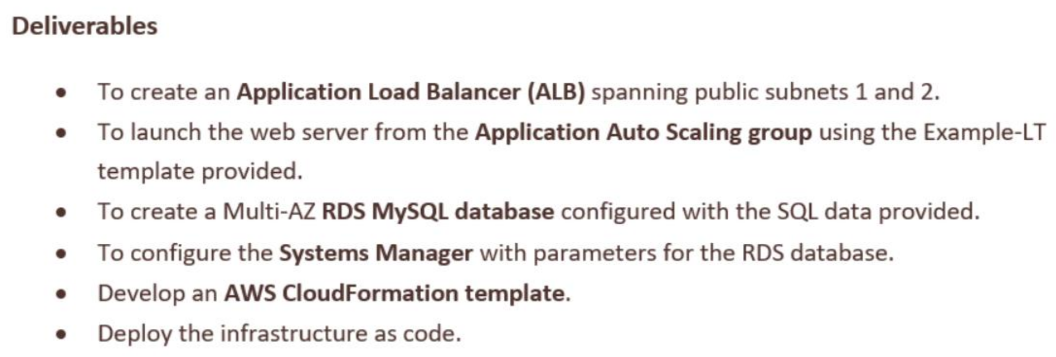


Figure 1: New website architecture design.

The implementation stage of the project was carried out following the project requirements and deliverables as defined in the project design document.



Deliverables

- To create an **Application Load Balancer (ALB)** spanning public subnets 1 and 2.
- To launch the web server from the **Application Auto Scaling group** using the Example-LT template provided.
- To create a Multi-AZ **RDS MySQL database** configured with the SQL data provided.
- To configure the **Systems Manager** with parameters for the RDS database.
- Develop an **AWS CloudFormation template.**
- Deploy the infrastructure as code.

Figure 2: Project implementation deliverables (A4 Project Design 2022).

2. Implementation

ELASTIC LOAD BALANCER (ELB) AND APPLICATION AUTO SCALING GROUP

Elastic load balancers are key components of the AWS web hosting architecture as they provide automatic distribution of incoming traffic across multiple application targets either from a single Availability Zone (AZs) or situated across multiple Availability Zones (AWS 2022a).

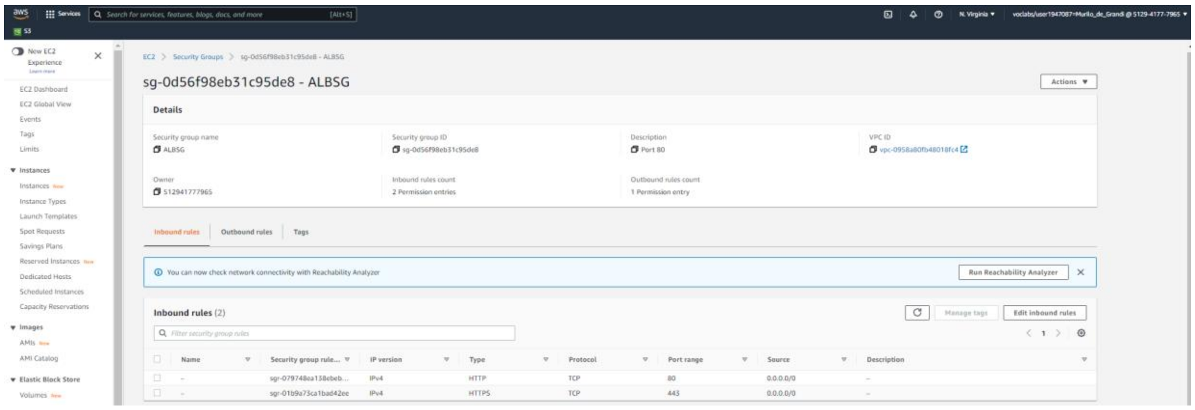
There are currently four types of load balancers that can be used to balance traffic across different targets:

- Network load balancer - works on layer 4 and listens for TCP, UDP and TLS traffics;
- Application load balancer - works on layer 7 and listens to HTTP/HTTPs and gRPC traffics;
- Gateway load balancer – works on layer 3 and 4 and listens to IP traffic;
- Classic load balancer (legacy) – works on layers 4/7 and listens to TCP, SSL/TLS,HTTP and HTTPS (AWS 2022b).

The Example website is expected to be highly available for internet users over the World Wide Web (www) using HTTP protocol, which is not recommended for production deployment as it does not encrypts the communications.

A multi-AZ Application load balancer can be deployed to provide high availability and add security to the website solution by preventing internet users from directly accessing the web servers (Virtanen 2021).

An internet-facing Application load balancer named *Example-ELB* was deployed and spans the two public subnets of the Example VPC. Access control was implemented using the existing security group (*ALBSG*) to allow only HTTP (port 80) and HTTPS (port 443) inbound traffic from the internet.



sg-0d56f98eb31c95de8 - ALBSG

Details

Security group name: ALBSG
Security group ID: sg-0d56f98eb31c95de8
Description: Port 80
VPC ID: vpc-0958a80fb48018fc4
Owner: 512941777965
Inbound rules count: 2 Permission entries
Outbound rules count: 1 Permission entry

You can now check network connectivity with Reachability Analyzer

Inbound rules (2)

Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
–	sgr-079748ea138ebeb...	IPv4	HTTP	TCP	80	0.0.0.0/0	–
–	sgr-01b9a73ca1bad42ee	IPv4	HTTPS	TCP	443	0.0.0.0/0	–

Figure 3: ALBSG Security Group.

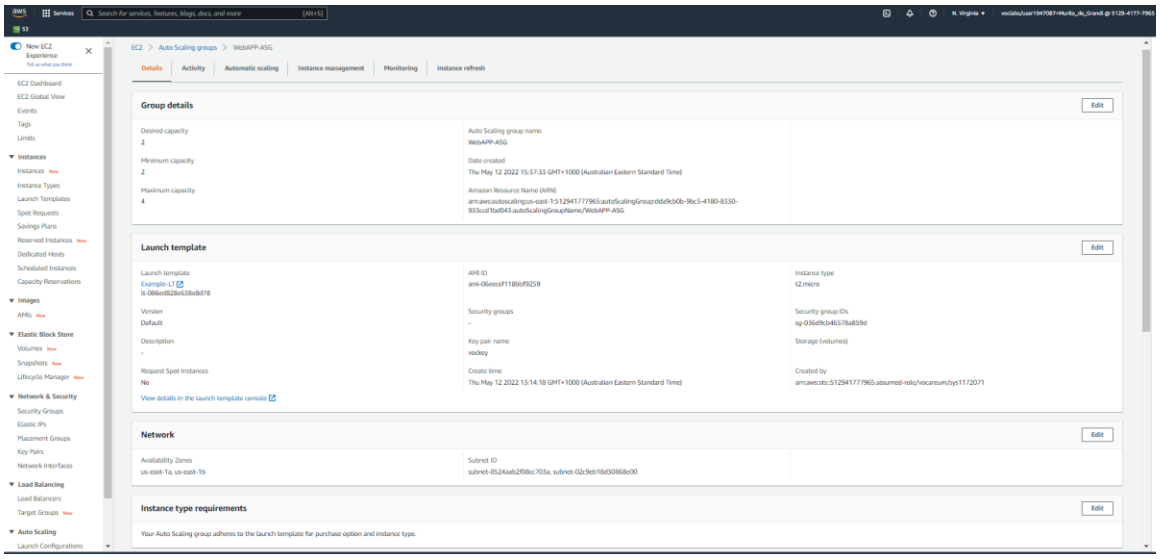
Due to the availability and scalability requirement of the new website design, the PHP application was deployed by an Auto Scaling Group (ASG) from the existing launch template *Example-LT*. The ASG is configured to provide elastic capability to the website, scaling hosts in and out according to changes in access demand and capacity needs.

The desired capacity (number of VMs) and minimum capacity were set to 2, and the maximum capacity was set to 4.

The Auto Scaling Group was launched with the following configuration:

Auto Scaling group name:	WebAPP-ASG
Launch Template:	Example-LT
Availability Zones and subnets:	Private Subnet 1/Private Subnet 2
Health-Check:	300
Group size:	Desired Capacity (2) Minimum Capacity (2) Maximum Capacity (4)
Tag:	Name: WebAPP-ASG

Table 1: Auto Scaling Group settings.



Group details

Desired capacity: 2
Minimum capacity: 2
Maximum capacity: 4
Auto Scaling group name: WebAPP-ASG
Date created: Thu May 12 2022 15:37:33 GMT+1000 (Australian Eastern Standard Time)
Amazon Resource Name (ARN): arn:aws:autoscaling:us-east-1:512941777965:autoScalingGroup:dda9cb0b-9bc3-4180-8330-933ccd1bd043:autoScalingGroupName/WebAPP-ASG

Launch template

Launch template: Example-LT lt-086ed828e638e8d78
Version: Default
Description: -
Request Spot Instances: No
AMI ID: ami-06eecef118bbf9259
Security groups: -
Key pair name: vockey
Create time: Thu May 12 2022 13:14:18 GMT+1000 (Australian Eastern Standard Time)
Instance type: t2.micro
Security group IDs: sg-036d9cb46578a859d
Storage (volumes)
Created by: arn:aws:sts::512941777965:assumed-role/vocareum/sys1172071

Network

Availability Zones: us-east-1a, us-east-1b
Subnet ID: subnet-0524aab2f08cc703a, subnet-02c9eb18d30868e00

Instance type requirements

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

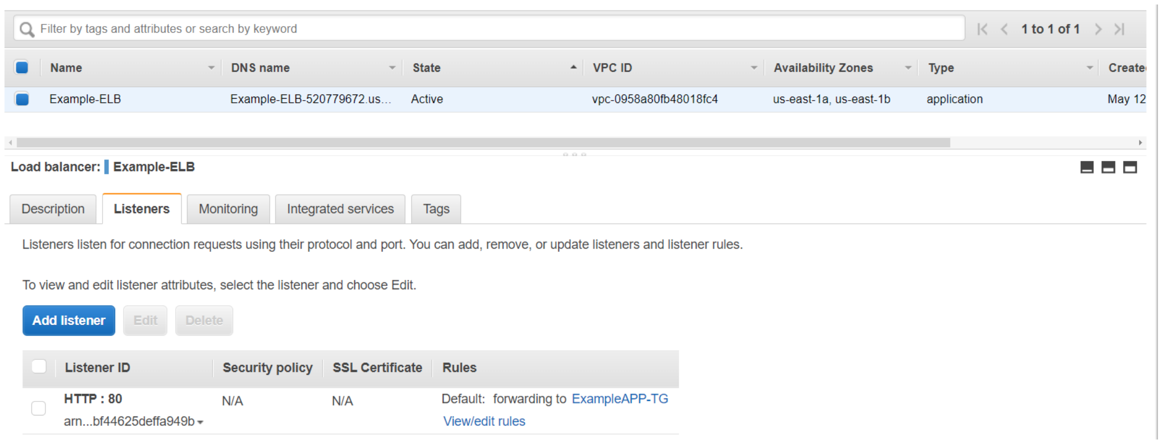
Figure 4: Auto Scaling Group

Then, a Target Group (TG) pointing to the PHP application (*WebAPP-ASG* instances) was created. And finally, the Application Load Balancer was deployed to route the HTTP and HTTPS traffic to the target group and provide access to the website for internet users.

The load balancer was launched with the following configuration:

Load balancer name:	Example-ELB
Scheme:	Internet-facing
VPC:	Example VPC
Subnets:	Public Subnet 1/Public Subnet 2
Security group:	ALBSG
Target group:	WebAPP-TG

Table 2: Load balancer settings.



Filter by tags and attributes or search by keyword — 1 to 1 of 1

Name	DNS name	State	VPC ID	Availability Zones	Type	Create
Example-ELB	Example-ELB-520779672.us...	Active	vpc-0958a80fb48018fc4	us-east-1a, us-east-1b	application	May 12

Load balancer: Example-ELB

Description | Listeners | Monitoring | Integrated services | Tags

Listeners listen for connection requests using their protocol and port. You can add, remove, or update listeners and listener rules.

To view and edit listener attributes, select the listener and choose Edit.

Add listener | Edit | Delete

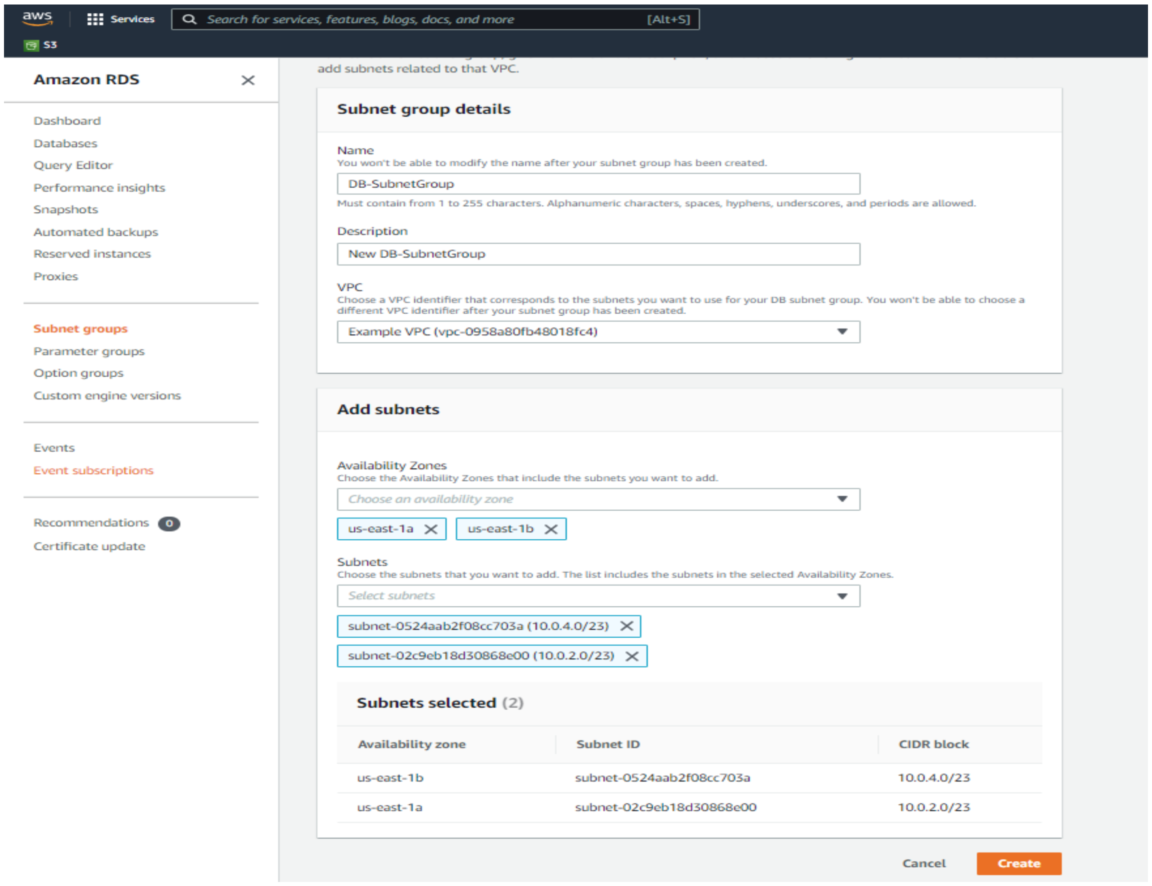
Listener ID	Security policy	SSL Certificate	Rules
HTTP : 80 arn...bf44625deffa949b	N/A	N/A	Default: forwarding to ExampleAPP-TG View/edit rules

Figure 5: Elastic load balancer.

RDS DATABASE

The PHP application of the website requires an RDS database solution with MySQL engine to integrate the existing website data from the MySQL dump file. Amazon Multi-AZ RDS with one standby is a cost-effective solution that consists of one primary and one replica database to provide fault tolerance in the event of failures (AWS 2022a).

Prior to creating the database, a new subnet group containing the two private subnets of Example VPC was created and attached to the RDS database. This multi-AZ implementation with automatic failover will guarantee that the replica assumes control in under 60 seconds and without any data loss in case the primary database becomes unavailable.



add subnets related to that VPC.

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.
DB-SubnetGroup
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description
New DB-SubnetGroup

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
Example VPC (vpc-0958a80fb48018fc4)

Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a
us-east-1b

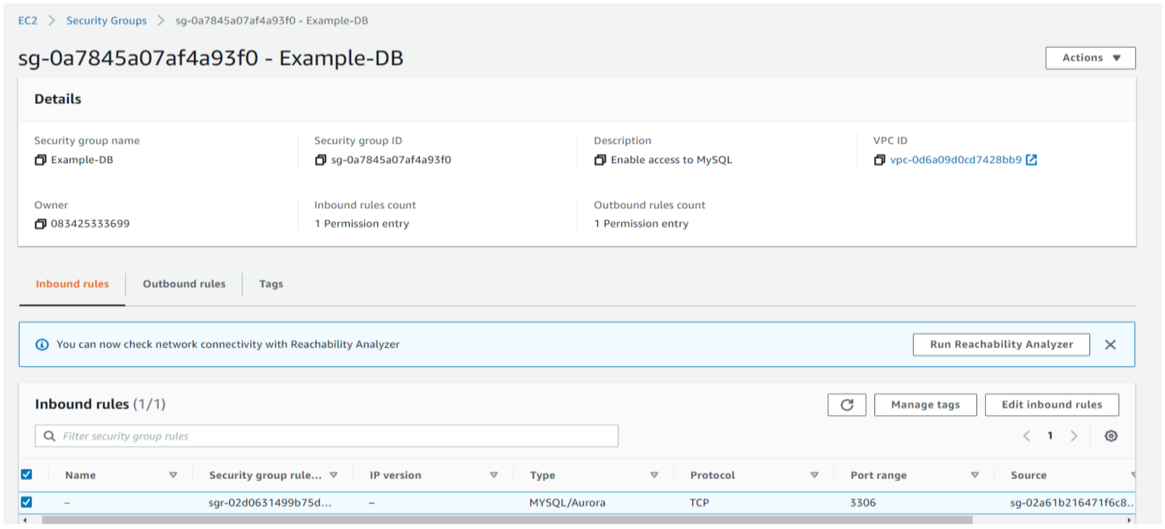
Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-0524aab2f08cc703a (10.0.4.0/23)
subnet-02c9eb18d30868e00 (10.0.2.0/23)

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-0524aab2f08cc703a	10.0.4.0/23
us-east-1a	subnet-02c9eb18d30868e00	10.0.2.0/23

Figure 6: Creating the subnet group.

The existing security group *Example-DBSG* was assigned to the RDS database and will exclusively allow traffic from the Inventory-App security group that is assigned to the web servers.


EC2 > Security Groups > sg-0a7845a07af4a93f0 - Example-DB
Actions
Inbound rules
Outbound rules
Tags
Run Reachability Analyzer
Manage tags
Edit inbound rules
Filter security group rules
1

sg-0a7845a07af4a93f0 - Example-DB

Details

Security group name	Security group ID	Description	VPC ID
Example-DB	sg-0a7845a07af4a93f0	Enable access to MySQL	vpc-0d6a09d0cd7428bb9
Owner	Inbound rules count	Outbound rules count	
083425333699	1 Permission entry	1 Permission entry	

You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

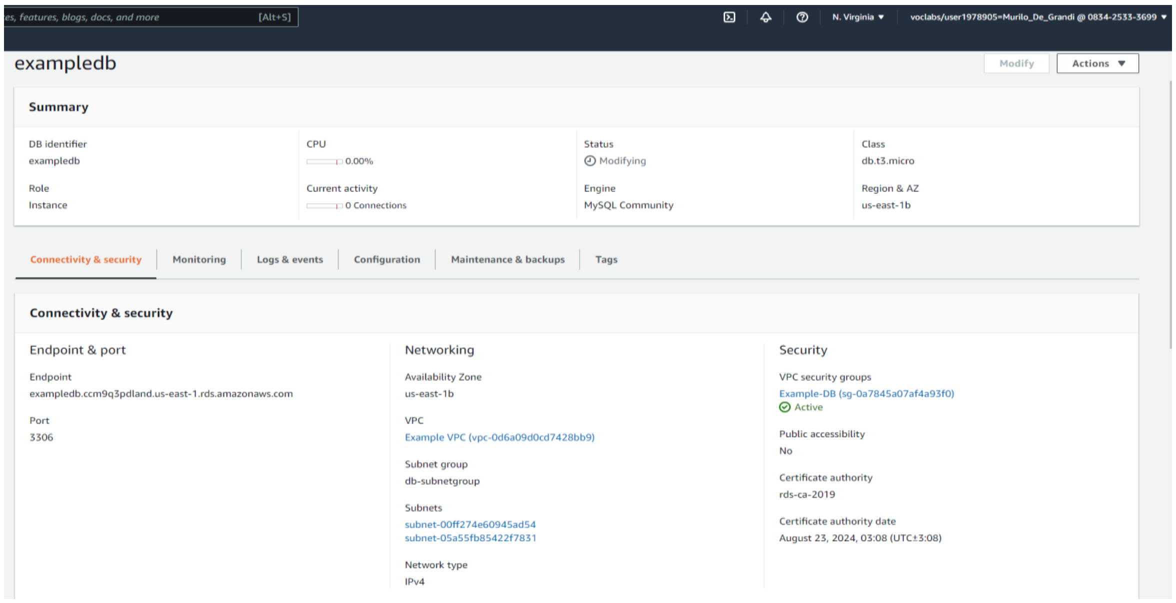
Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-02d0631499b75d...	–	MYSQL/Aurora	TCP	3306	sg-02a61b216471f6c8..

Figure 7: Example-DB security group.

Finally, the multi-AZ RDS database was deployed with the following configuration:

Database Engine: MySQL version 8.0.23
Template: Dev/Test
Availability and durability: Multi-AZ DB Instance
Db Instance Identifier: exampledb
DB Instance class: Burstable classes (Db.t3.micro)
Storage: General Purpose SSD (20gb) with storage autoscaling
Public Access: No

Table 3: RDS database settings.



exampledb

Summary

DB identifier	CPU	Status	Class
exampledb	0.00%	Modifying	db.t3.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-1b

Connectivity & security

Endpoint & port

Endpoint
exampledb.ccm9q3pdland.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1b

VPC
Example VPC (vpc-0d6a09d0cd7428bb9)

Subnet group
db-subnetgroup

Subnets
subnet-00ff274e60945ad54
subnet-05a55fb85422f7831

Network type
IPv4

Security

VPC security groups
Example-DB (sg-0a7845a07af4a93f0)
Active

Public accessibility
No

Certificate authority
rds-ca-2019

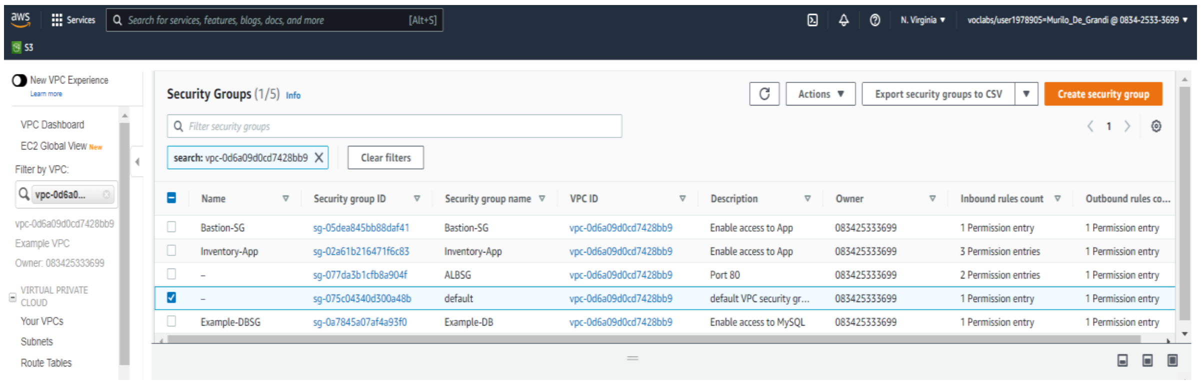
Certificate authority date
August 23, 2024, 03:08 (UTC±3:08)

Figure 8: New RDS Database.

NETWORKING AND SECURITY FEATURES

Security Groups (SGs) work similarly to stateful firewalls and are used to allow access to hosts. Network Access Control Lists (NACLs) provide stateless security management of IP traffic at the subnet level and can be used to allow or deny traffic before it reaches the target host (Todorov and Ozkan 2013).

Example VPC comes with a default NACL that allows all types of inbound or outbound traffic to reach both public and private subnets. The security of the web servers and databases of the Example website is delivered at the host level by using Security Groups (SGs). The security groups permitted the implementation of different security policies according to the particular security needs of each host.



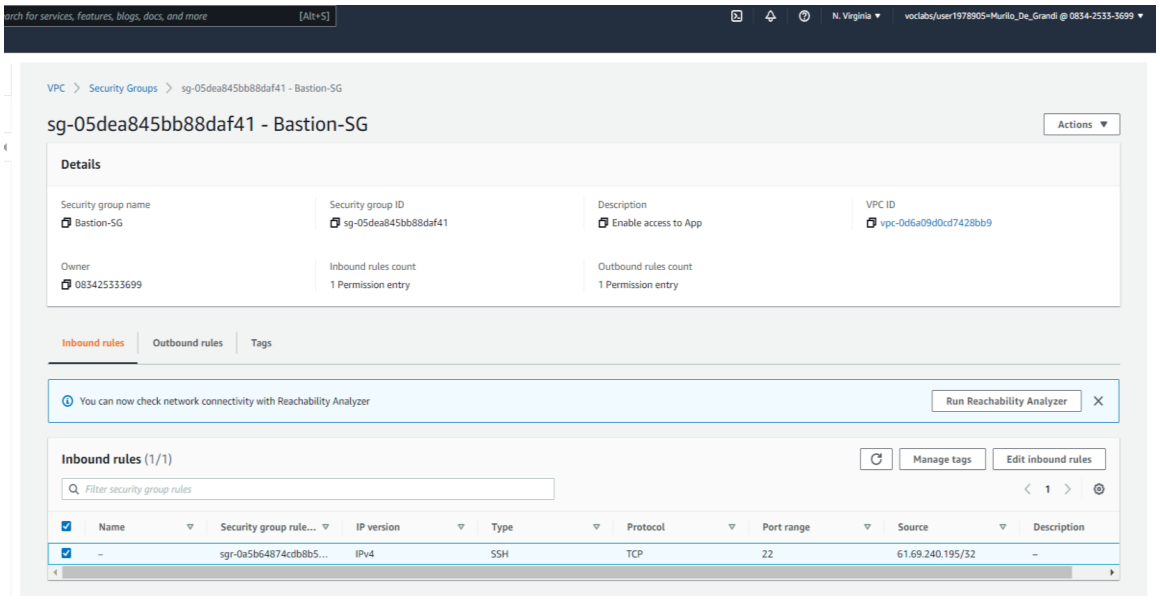
Security Groups (1/5) Info

search: vpc-0d6a09d0cd7428bb9

	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules co...
☐	Bastion-SG	sg-05dea845bb88daf41	Bastion-SG	vpc-0d6a09d0cd7428bb9	Enable access to App	083425333699	1 Permission entry	1 Permission entry
☐	Inventory-App	sg-02a61b216471f6c83	Inventory-App	vpc-0d6a09d0cd7428bb9	Enable access to App	083425333699	3 Permission entries	1 Permission entry
☐	–	sg-077da3b1cfb8a904f	ALBSG	vpc-0d6a09d0cd7428bb9	Port 80	083425333699	2 Permission entries	1 Permission entry
✓	–	sg-075c04340d300a48b	default	vpc-0d6a09d0cd7428bb9	default VPC security gr...	083425333699	1 Permission entry	1 Permission entry
☐	Example-DBSG	sg-0a7845a07af4a93f0	Example-DB	vpc-0d6a09d0cd7428bb9	Enable access to MySQL	083425333699	1 Permission entry	1 Permission entry

Figure 8: Example VPC Security groups.

Bastion-SG security group: Permits SSH traffic to the Bastion Host. This existing SG originally contained a rule that permits connections from any IP sources, introducing unnecessary risks for the host and the environment. To solve that issue, this rule has been modified and now it only allows access for the administrator (my IP address).


arch for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
voclabs/user1978905=Murilo_De_Grandi @ 0834-2533-3699
VPC
Security Groups
sg-05dea845bb88daf41 - Bastion-SG
Actions
Inbound rules
Outbound rules
Tags
Run Reachability Analyzer
Manage tags
Edit inbound rules
Filter security group rules

sg-05dea845bb88daf41 - Bastion-SG

Details

Security group name	Security group ID	Description	VPC ID
Bastion-SG	sg-05dea845bb88daf41	Enable access to App	vpc-0d6a09d0cd7428bb9
Owner	Inbound rules count	Outbound rules count	
083425333699	1 Permission entry	1 Permission entry	

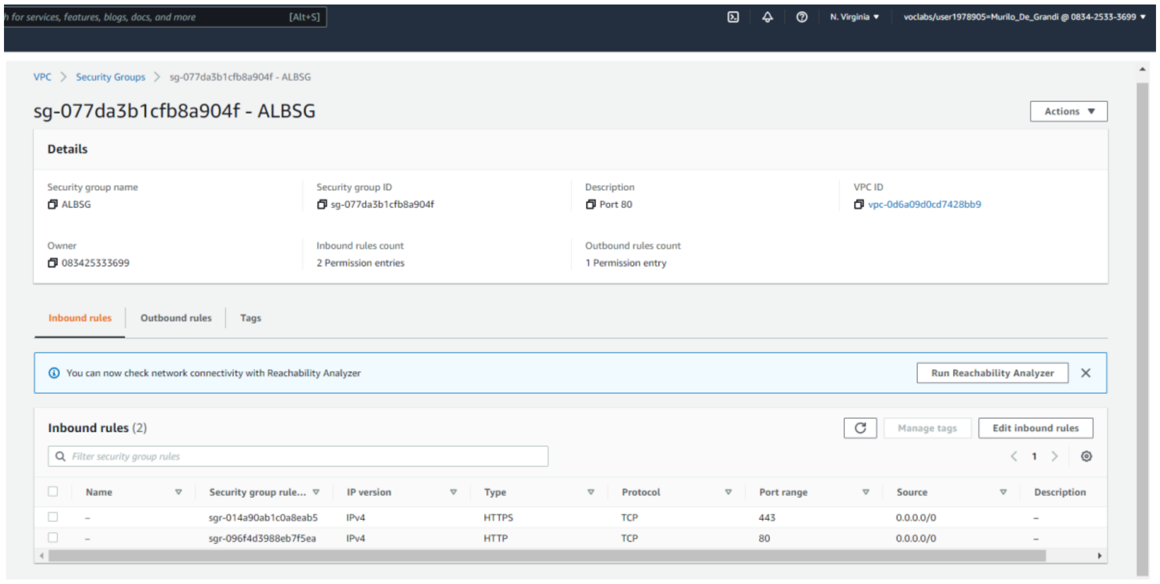
You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
✓	–	sgr-0a5b64874cdb8b5...	IPv4	SSH	TCP	22	61.69.240.195/32	–

Figure 9: Bastion-SG security group.

ALBSG security group: The Application Load Balancer security group. Allows HTTP (80) and HTTPS (443) traffic from any source.



VPC > Security Groups > sg-077da3b1cfb8a904f - ALBSG

sg-077da3b1cfb8a904f - ALBSG

Details

Security group name	Security group ID	Description	VPC ID
ALBSG	sg-077da3b1cfb8a904f	Port 80	vpc-0d6a09d0cd7428bb9
Owner	Inbound rules count	Outbound rules count	
083425333699	2 Permission entries	1 Permission entry	

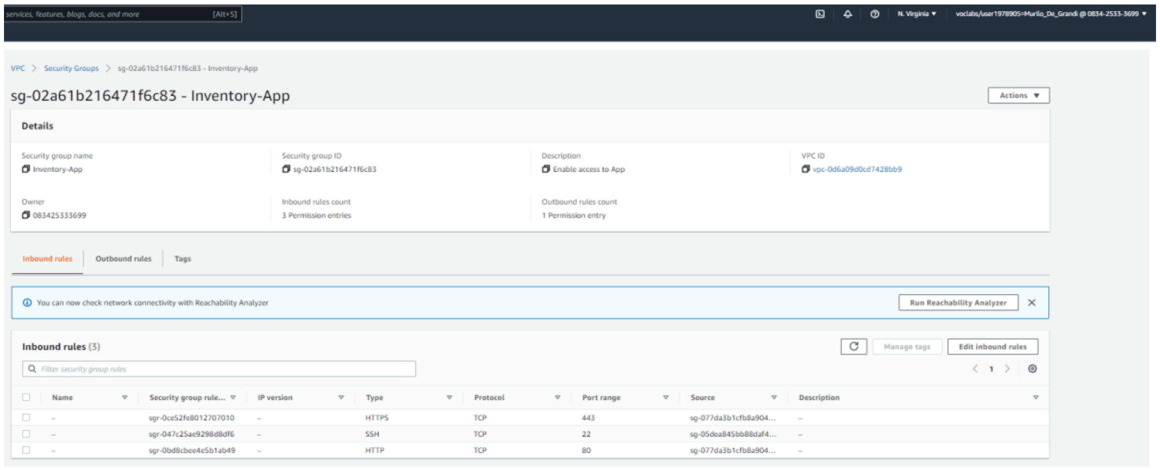
You can now check network connectivity with Reachability Analyzer

Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
–	sgr-014a90ab1c0a8eab5	IPv4	HTTPS	TCP	443	0.0.0.0/0	–
–	sgr-096f4d3988eb7f5ea	IPv4	HTTP	TCP	80	0.0.0.0/0	–

Figure 10: ALBSG security group.

Inventory-App security group: Defines access rules for the *WebAPP-ASG* instances. The purpose of this SG is to allow HTTP(80) and HTTPS(443) traffic from the Application Load Balancer (*ALBSG)*. In order to permit remote management of the instances, a new rule was added to this SG to allow SSH traffic from the *Bastion*-*SG*.



VPC > Security Groups > sg-02a61b216471f6c83 - Inventory-App

sg-02a61b216471f6c83 - Inventory-App

Details

Security group name	Security group ID	Description	VPC ID
Inventory-App	sg-02a61b216471f6c83	Enable access to App	vpc-0d6a09d0cd7428bb9
Owner	Inbound rules count	Outbound rules count	
083425333699	3 Permission entries	1 Permission entry	

You can now check network connectivity with Reachability Analyzer

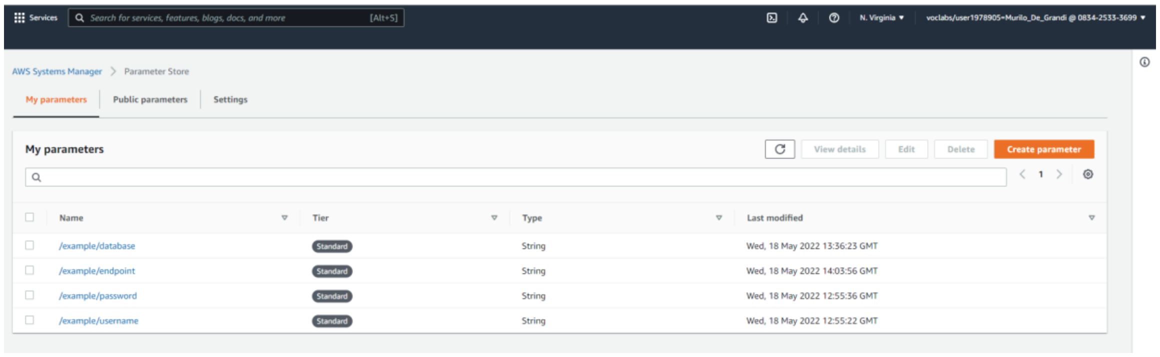
Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
–	sgr-0ce52fe8012707010	–	HTTPS	TCP	443	sg-077da3b1cfb8a904...	–
–	sgr-047c25ae9298d8df6	–	SSH	TCP	22	sg-05dea845bb88daf4...	–
–	sgr-0bd8cbee4e5b1ab49	–	HTTP	TCP	80	sg-077da3b1cfb8a904...	–

Figure 11: Inventory-App security group.

The security of the AWS environment can also be enhanced by using the AWS Systems Manager Parameter Store. According to (Todorov and Ozkan 2013) the Parameter Store provides a secure, reliable and convenient way of storing sensitive information such user credentials and access keys from applications within the AWS Systems Manager rather than including that in the code.

In this implementation, AWS Parameter Store was used to store the sensitive information related to the database.


Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1978905=Murilo_De_Grandi @ 0834-2533-3699
AWS Systems Manager > Parameter Store
My parameters
Public parameters
Settings
View details
Edit
Delete
Create parameter

My parameters

Name	Tier	Type	Last modified
/example/database	Standard	String	Wed, 18 May 2022 13:36:23 GMT
/example/endpoint	Standard	String	Wed, 18 May 2022 14:03:56 GMT
/example/password	Standard	String	Wed, 18 May 2022 12:55:36 GMT
/example/username	Standard	String	Wed, 18 May 2022 12:55:22 GMT

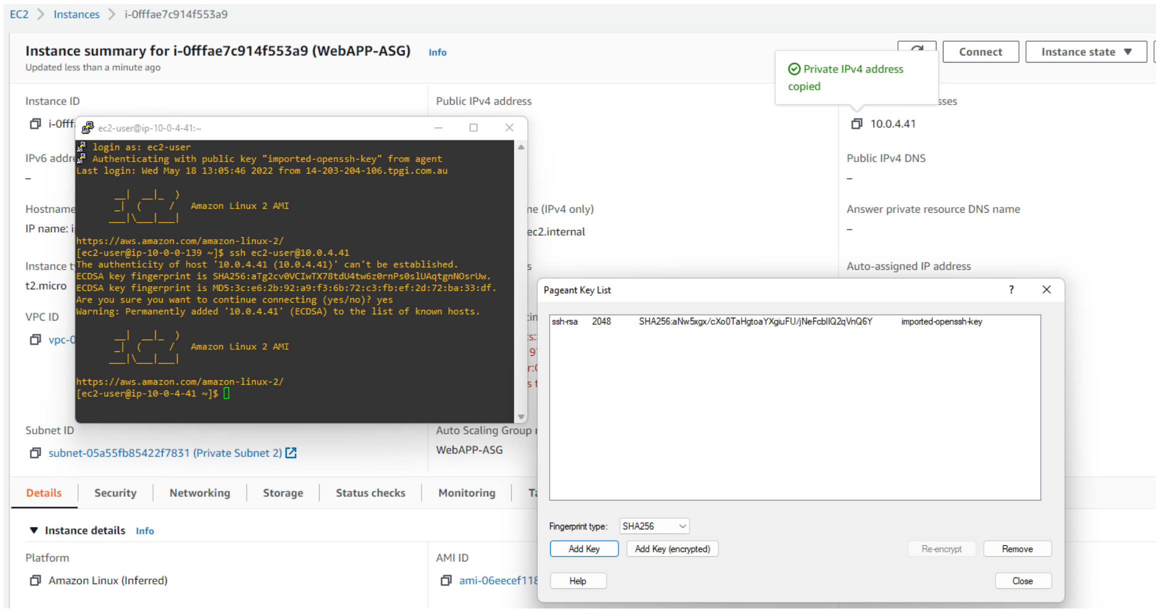
Figure 12: Parameter Store.

WEB SERVER AND DATABASE MANAGEMENT

According to Thalagatti et al. (2015), a Bastion Host located in a public subnet can be used to perform administrative and maintenance tasks in hosts located in private subnets using SSH (TCP/22) via Putty (on Windows) with agent forwarding enabled. This technique is known as SSH passthrough.

Considering that the PHP application is launched by the Auto Scaling Group *WebAPP-*ASG using the launch template *Example-LT which* comes with all necessary configuration and user data out of the box, the only task remaining to enable the website is to populate the database with data. That could be easily achieved by transferring the SQL dump file from one of the web server instances to the RDS database.

Using Putty and Pageant from a Windows machine it was possible to SSH into the Bastion Host instance, and from there to SSH again into the web server instance to transfer the SQL dump file. This process was only possible because an inbound rule was added to the existing *Inventory-App* security group permitting SSH connections from the *Bastion Host-SG*.


EC2 > Instances > i-0fffae7c914f553a9

Instance summary for i-0fffae7c914f553a9 (WebAPP-ASG)

Updated less than a minute ago

Private IPv4 address copied

Instance ID: i-0fff

Public IPv4 address

Private IPv4 addresses: 10.0.4.41

Public IPv4 DNS: –

Answer private resource DNS name: –

Instance type: t2.micro

Subnet ID: subnet-05a55fb85422f7831 (Private Subnet 2)

Auto Scaling Group name: WebAPP-ASG

Platform: Amazon Linux (Inferred)

AMI ID: ami-06eecef118

```
login as: ec2-user
Authenticating with public key "imported-openssh-key" from agent
Last login: Wed May 18 13:05:46 2022 from 14-203-204-106.tpgi.com.au

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-139 ~]$ ssh ec2-user@10.0.4.41
The authenticity of host '10.0.4.41 (10.0.4.41)' can't be established.
ECDSA key fingerprint is SHA256:aTg2cv0VCIwTX78tdU4tw6z0rnPs0slUAqtgnNOsrUw.
ECDSA key fingerprint is MD5:3c:e6:2b:92:a9:f3:6b:72:c3:fb:ef:2d:72:ba:33:df.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.4.41' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-4-41 ~]$
```

Pageant Key List

```
ssh-rsa   2048   SHA256:aNw5xgx/cXo0TaHgtoaYXgiuFU/jNeFcblIQ2qVnQ6Y   imported-openssh-key
```

Fingerprint type: SHA256

Figure 13: SSH passthrough.

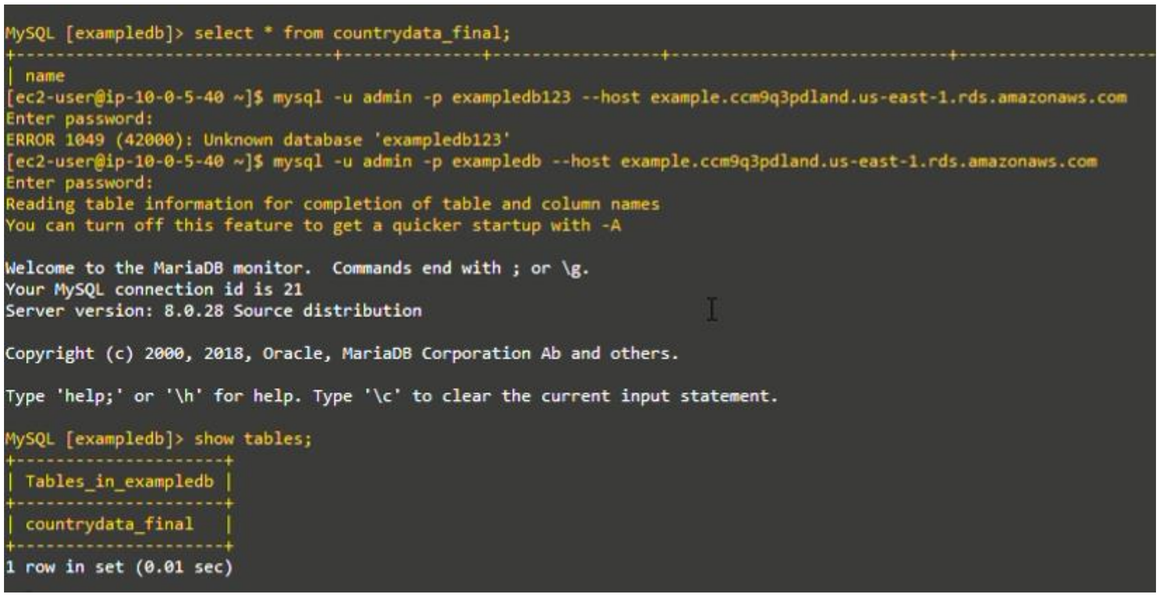
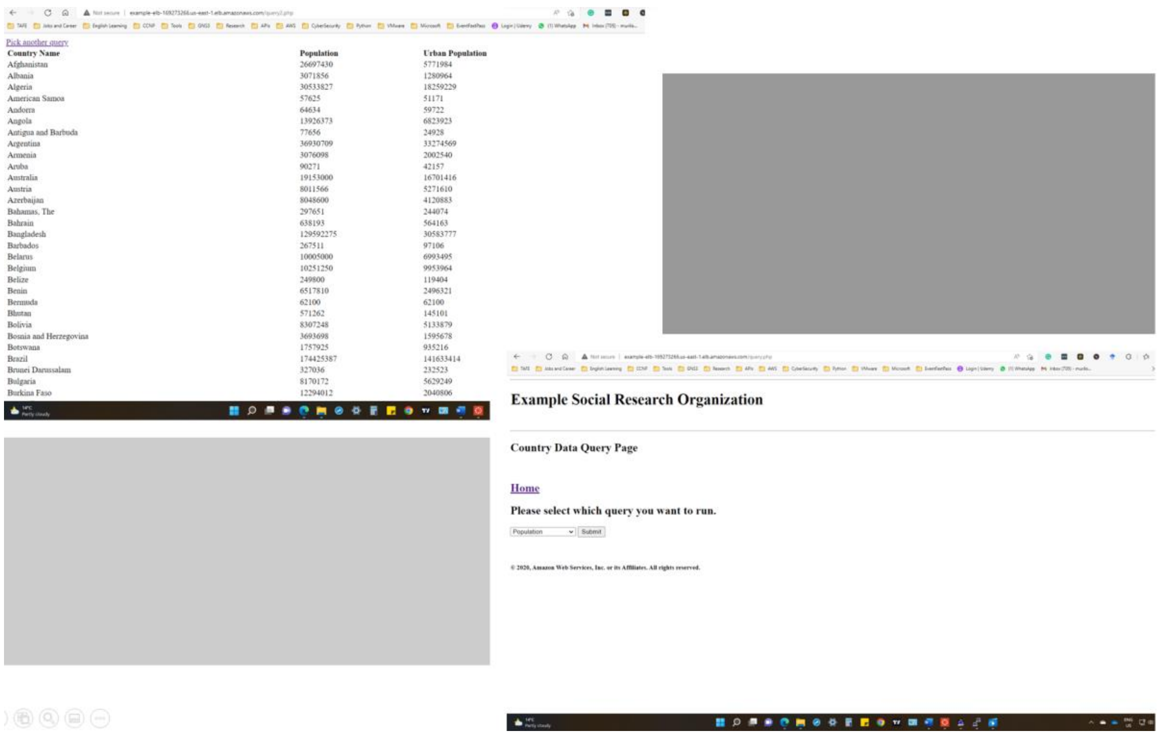

```
MySQL [exampledb]> select * from countrydata_final;
+--------------------------------+--------------+-----------------+----------------------------+------------
| name
[ec2-user@ip-10-0-5-40 ~]$ mysql -u admin -p exampledb123 --host example.ccm9q3pdland.us-east-1.rds.amazonaws.com
Enter password:
ERROR 1049 (42000): Unknown database 'exampledb123'
[ec2-user@ip-10-0-5-40 ~]$ mysql -u admin -p exampledb --host example.ccm9q3pdland.us-east-1.rds.amazonaws.com
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [exampledb]> show tables;
+---------------------+
| Tables_in_exampledb |
+---------------------+
| countrydata_final   |
+---------------------+
1 row in set (0.01 sec)
```

Figure 14: Transfering the SQL dump file to the RDS database.

Once the file SQL dump file was imported to the database, the new website solution became fully ready and available for production.


Pick another query

Country Name	Population	Urban Population
Afghanistan	26697430	5771984
Albania	3071856	1280964
Algeria	30533827	18259229
American Samoa	57625	51171
Andorra	64634	59722
Angola	13926373	6823923
Antigua and Barbuda	77656	24928
Argentina	36930709	33274569
Armenia	3076098	2002540
Aruba	90271	42157
Australia	19153000	16701416
Austria	8011566	5271610
Azerbaijan	8048600	4120883
Bahamas, The	297651	244074
Bahrain	638193	564163
Bangladesh	129592275	30583777
Barbados	267511	97106
Belarus	10005000	6993495
Belgium	10251250	9953964
Belize	249800	119404
Benin	6517810	2496321
Bermuda	62100	62100
Bhutan	571262	145101
Bolivia	8307248	5133879
Bosnia and Herzegovina	3693698	1595678
Botswana	1757925	935216
Brazil	174425387	141633414
Brunei Darussalam	327036	232523
Bulgaria	8170172	5629249
Burkina Faso	12294012	2040806

Example Social Research Organization

Country Data Query Page

Home

Please select which query you want to run.

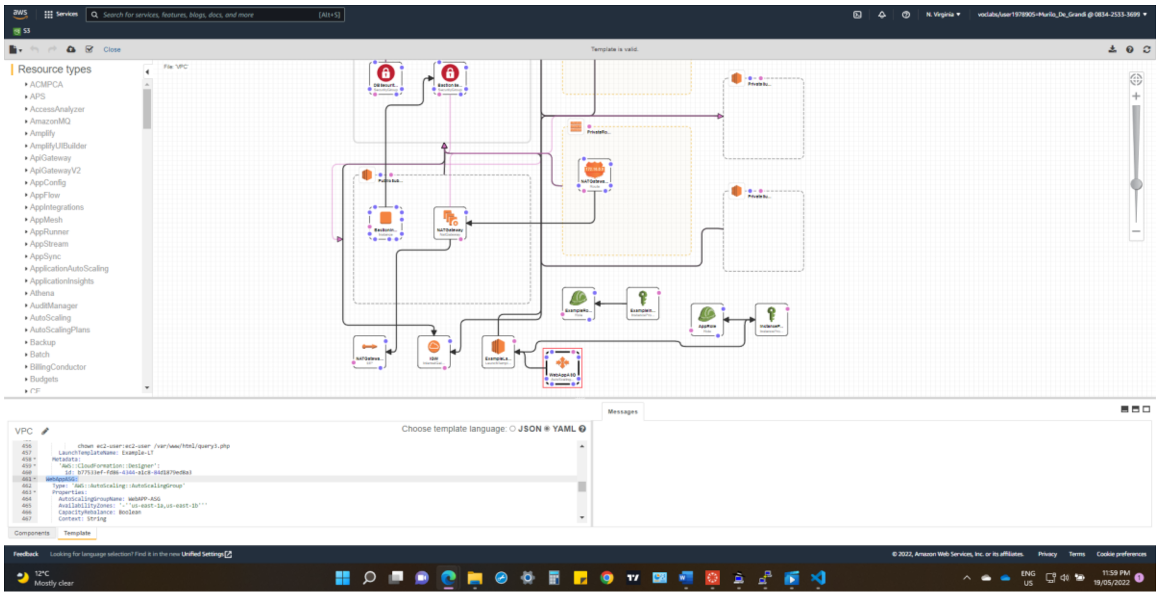
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Figure 15: New Example website.

AUTOMATION

AWS enables Infrastructure as Code (IaC) by using the native service called AWS CloudFormation. Alternatively, automation can also be achieved by using the other popular application deployment and configuration tools such as Puppet, Chef and Ansible.

AWS CloudFormation templates provide a convenient way of automatically provisioning large collections of AWS and third-party resources that can even span across different accounts and regions (Kaushik 2021). The Auto Scaling Group WebApp-ASG was created in code and attached to the existing Capstone CloudFormation template.



```
        chown ec2-user:ec2-user /var/www/html/query3.php
    LaunchTemplateName: Example-LT
  Metadata:
    'AWS::CloudFormation::Designer':
      id: b77533ef-fd86-4344-a1c8-84d1879ed8a3
WebAppASG:
  Type: 'AWS::AutoScaling::AutoScalingGroup'
  Properties:
    AutoScalingGroupName: WebAPP-ASG
    AvailabilityZones: '-''us-east-1a,us-east-1b'''
    CapacityRebalance: Boolean
    Context: String
```

Figure 16: Auto Scaling Group Resource

Below is a sample of the WebAppASG YAML code that was used.

```
WebAppASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
      AutoScalingGroupName: 'WebAPP-ASG'
      AvailabilityZones:
          -'us-east-1a,us-east-1b'
      CapacityRebalance: Boolean
      Context: String
      Cooldown: String
      DefaultInstanceWarmup: Integer
      DesiredCapacity: '2'
      DesiredCapacityType: String
      HealthCheckGracePeriod: Integer
      HealthCheckType: String
      InstanceId: String
      LaunchConfigurationName: String
      LaunchTemplate:
          LaunchTemplateId: String
          LaunchTemplateName: 'Example-LT'
          Version: !GetAtt ExampleLaunchTemplate.LatestVersionNumber
      LifecycleHookSpecificationList:
          - LifecycleHookSpecification
      LoadBalancerNames:
          - 'arn:aws:elasticloadbalancing:us-east-
1:512941777965:loadbalancer/app/Example-ELB/ebe8c769382350f2'
```

```
      MaxInstanceLifetime: Integer
      MaxSize: '4'
      MetricsCollection:
        - MetricsCollection
      MinSize: '2'
      MixedInstancesPolicy:
        MixedInstancesPolicy
      NewInstancesProtectedFromScaleIn: Boolean
      NotificationConfigurations:
        - NotificationConfiguration
      PlacementGroup: String
      ServiceLinkedRoleARN: 'arn:aws:iam::512941777965:role/aws-service-
role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling'
      Tags:
        - Key: 'Name'
      PropagateAtLaunch: True
      Value: 'WebAPP-ASG'
      TargetGroupARNs:
        - 'arn:aws:elasticloadbalancing:us-east-
1:512941777965:targetgroup/ExampleAPP-TG/89f69da556cf98d2'
      TerminationPolicies:
        - String
      VPCZoneIdentifier:
        - subnet-0524aab2f08cc703a
        - subnet-02c9eb18d30868e00
```

3. Proposed Design VS. Actual Implementation

This project has been delivered within the allocated budget and timeframe. The new website solution is up and running, meeting all the solution requirements and was implemented according to the proposed design.

The implementation was based on a two-tier architecture design. The new solution enhanced availability, reliability, scalability and security aspects through the Application Load Balancer and the Auto Scaling Group.

By comparing the implementation results obtained against the objectives of the new design, it is noticeable that most objectives were successfully achieved except for the Infrastructure as Code (IaC) deployment, which was partially delivered.

In terms of resources, the addition of an extra Nat Gateway to create a multi-az connection was the only change to the proposed design. This implementation should improve the security posture of the solution by ensuring that the web servers will be always fed with the latest patches and OS updates, even if the public subnet that serves one of the NAT Gateways is unavailable.

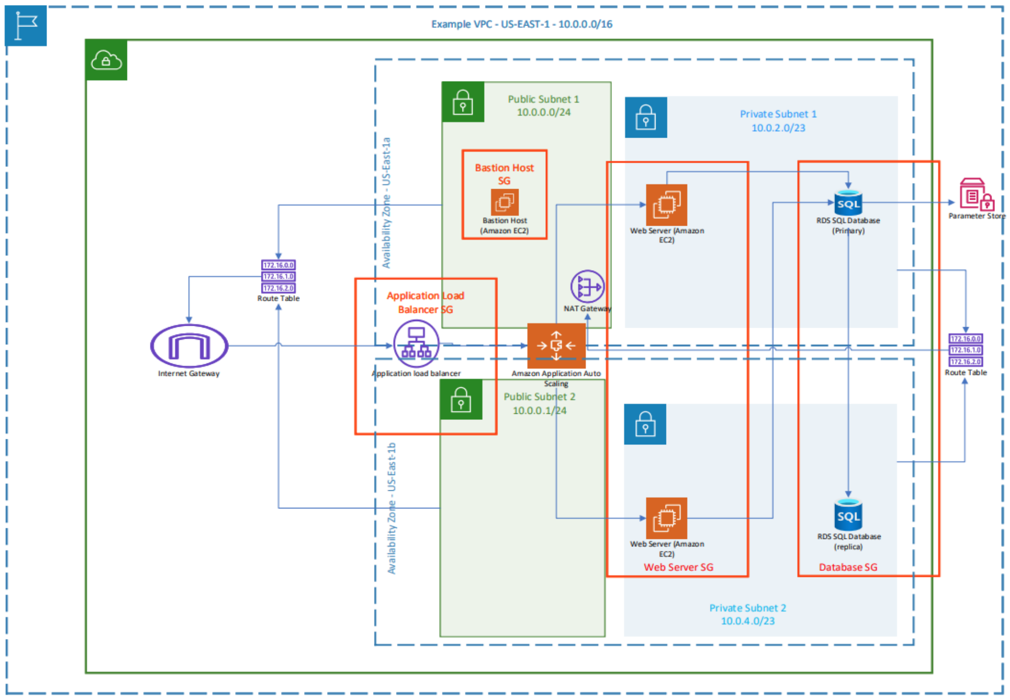
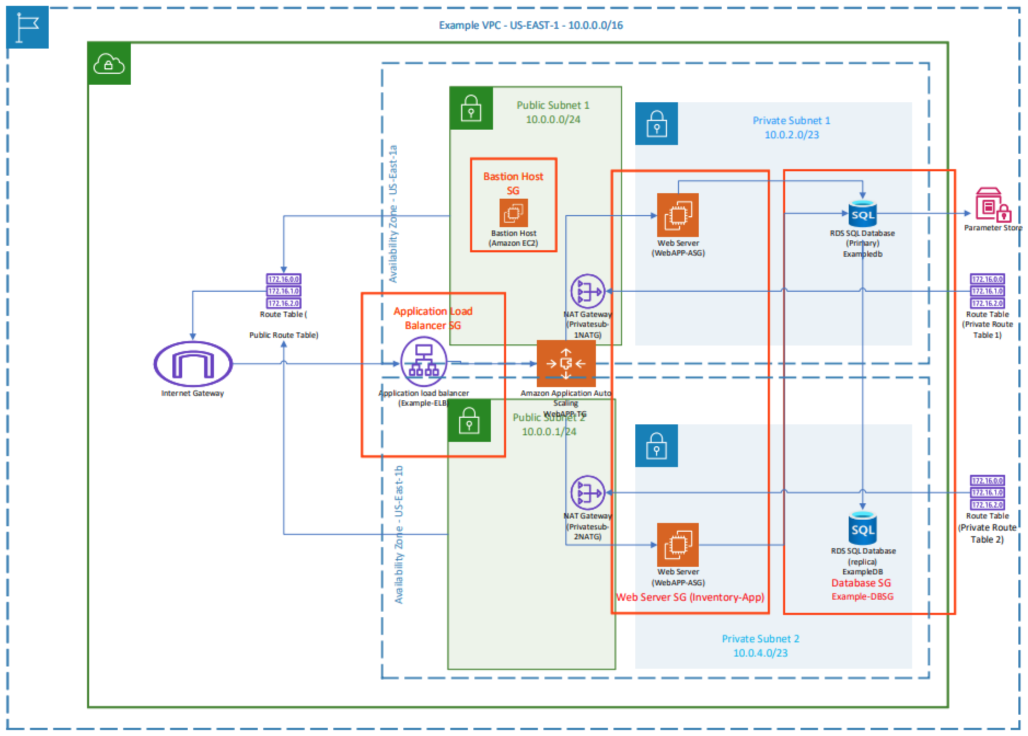
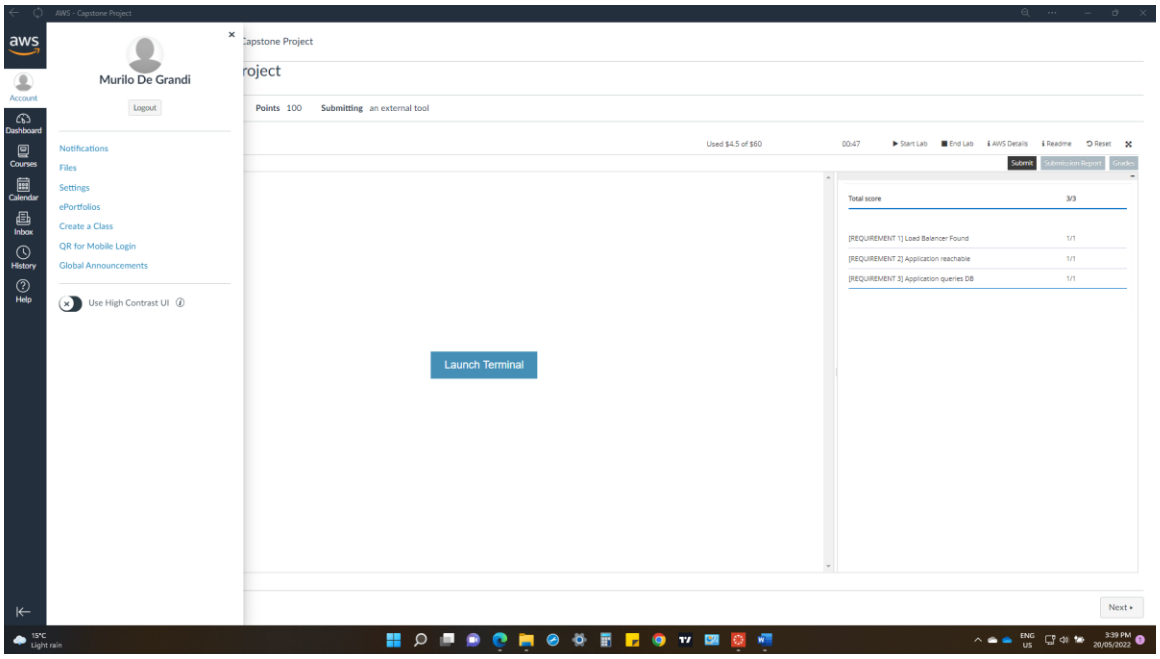


Figure 17: Proposed design diagram.



Example VPC - US-EAST-1 - 10.0.0.0/16
Availability Zone - US-East-1a
Public Subnet 1
10.0.0.0/24
Private Subnet 1
10.0.2.0/23
Bastion Host SG
Bastion Host (Amazon EC2)
Web Server (WebAPP-ASG)
RDS SQL Database (Primary) Exampledb
Parameter Store
Route Table (Public Route Table)
172.16.0.0
172.16.1.0
172.16.2.0
Route Table (Private Route Table 1)
Internet Gateway
Application Load Balancer SG
Application load balancer (Example-ELB)
NAT Gateway (Privatesub-1NATG)
Amazon Application Auto Scaling
WebAPP-TG
Public Subnet 2
10.0.0.1/24
Availability Zone - US-East-1b
NAT Gateway (Privatesub-2NATG)
Web Server (WebAPP-ASG)
Web Server SG (Inventory-App)
RDS SQL Database (replica) ExampleDB
Database SG
Example-DBSG
Route Table (Private Route Table 2)
Private Subnet 2
10.0.4.0/23

Figure 18: Implemented design diagram.



Points 100 Submitting an external tool

Used $4.5 of $60 00:47

Total score	3/3
[REQUIREMENT 1] Load Balancer Found	1/1
[REQUIREMENT 2] Application reachable	1/1
[REQUIREMENT 3] Application queries DB	1/1

Figure 19: Capstone Project lab results.

4. Conclusion

Designing and implementing the Capstone project was a fantastic opportunity to discover and practice with multiple AWS services at once, as well as to gain hands-on experience while hosting a website solution according to AWS best practices.

Unfortunately, automation through CloudFormation could not be delivered in full due to time constraints. Nevertheless, the benefits introduced by IaC for developers and organisations cannot be denied and will certainly be included in my future works.

Finally, all the major objectives could be achieved and a more robust, reliable and cost-effective website implementation that meets the design requirements was delivered as planned.

References

AWS 2022a, *Key components of an AWS web hosting architecture – AWS Whitepaper*, Web Application Hosting in the AWS Cloud, viewed 20 April 2022, <https://docs.aws.amazon.com/whitepapers/latest/web-application-hosting-best-practices/key-components-of-an-aws-web-hosting-architecture.html>.

Virtanen, J Hämeenlinna University Center 2021, 'Comparing Different CI/CD Pipelines', *Degree Programme in Business Information Technology*, viewed 20 May 2022, <https://www.theseus.fi/bitstream/handle/10024/511026/Opinnaytetyo_Joni_Virtanen.pdf?sequence=2&isAllowed=y>.

Thalagatti, MP, Asrar Naveed, M & Chaitra, B Acharya Institute of Technology 2015, 'Reliable Data Tier Architecture for Job Portal using AWS', *(IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 6, viewed 26 April 2022, <http://13.232.72.61:8080/jspui/bitstream/123456789/353/1/Reliable%20Data%20Tier%20Architecture%20for%20Job%20Portal%20using%20AWS.pdf>.

Todorov, D & Ozkan, Y ACADEMIA 2013, 'Amazon Web Services – AWS Security Best Practices AWS Security Best Practices Amazon Web Services', *AWS Best Practices*, viewed 20 May 2022, <https://bit.ly/3G1lwOA>.

Kaushik, A National College of Ireland 2021, 'A CloudFormation template using AWS golden standards to avoid cascading failures in hardware', *MSc in Cloud Computing Research Project*, viewed 20 May 2022, <http://norma.ncirl.ie/5088/1/arnauvkaushik.pdf>.

Elastic Load Balancing features 2022, AWS, viewed 20 May 2022, <https://aws.amazon.com/elasticloadbalancing/features/?nc=sn&loc=2&dn=1>.