



NI6504 ASSIGNMENT 2

Cloud Computing

Dr Bryce Antony

Andrew Graff
2231290

Introduction

Kia ora,

In this report we perform some hands-on tasks of creating a new VPC (virtual private console), launching an EC2 (elastic computing 2) instance, creating an Amazon RDS (relational database) DB instance and installing a web server on the EC2 instance. These are completed in our AWS (amazon web services) sandbox environment supported with 20 screenshots of which will demonstrate the carrying out of these specified tasks.

We simply create a new virtual network, set up a virtual computer, create online database and install a web server on our computer to host websites.

Each image will show legibility, accuracy, completeness and correctness as per instructions. There are screenshots of VPC set up for the DB instance, security groups that were created for the public web server and private DB instance and attached to our VPC. A DB subnet group was created

The second part of this report is a discussion of what was involved to achieve the tasks required. We discuss troubleshooting processes and methods to obtain the screenshots within restraints which in this case was time and the familiarity with AWS and the various tools for set up.

1: Create VPC for use with DB instance (IPv4 only)

Use the following information:

- Name tag auto-generation – NI6504-2231290
- IPv4 CIDR block – 10.90.0.0/16
- IPv6 CIDR block – No IPv6 CIDR block
- Tenancy – Default
- Number of Availability Zones (AZs) – 2
- Customize AZs – Keep the default values.
- Number of public subnet – 2
- Number of private subnets – 2
- Customize subnets CIDR blocks – Keep the default values.
- NAT gateways (\$) – None
- VPC endpoints – None
- DNS options – Keep the default values.

The screenshot displays the AWS Management Console interface for a VPC. The top navigation bar shows the user is logged in as 'voclabs/user3388989=Andrew_Graff' in the 'N. Virginia' region. The breadcrumb trail indicates the path: VPC > Your VPCs > vpc-07f83ca94987eaf1d. The main heading is 'vpc-07f83ca94987eaf1d / NI6504-2231290-vpc'. Below this, there are tabs for 'Details' (selected) and 'Info'. The 'Details' tab shows a table with the following information:

Property	Value
VPC ID	vpc-07f83ca94987eaf1d
Tenancy	Default
Default VPC	No
Network Address Usage metrics	Disabled
State	Available
DHCP option set	dopt-0f05656c7caa02287
IPv4 CIDR	10.90.0.0/16
Route 53 Resolver DNS Firewall rule groups	-
DNS hostnames	Enabled
Main route table	rtb-0d7633574a971c555
IPv6 pool	-
Owner ID	565109282158
DNS resolution	Enabled
Main network ACL	acl-01b6040184924b88d
IPv6 CIDR (Network border group)	-

Below the details table, there are tabs for 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'. The 'Resource map' tab is selected, showing a diagram of the VPC resources. The diagram includes:

- VPC**: A box labeled 'NI6504-2231290-vpc'.
- Subnets (4)**: A box containing four subnets: 'us-east-1a' (two public and two private) and 'us-east-1b' (two public and two private).
- Route tables (4)**: A box containing four route tables: 'rtb-0d7633574a971c555' (public), 'NI6504-2231290-rtb-private2-us-eas...', 'NI6504-2231290-rtb-private1-us-eas...', and 'NI6504-2231290-rtb-private1-us-eas...'.
- Network connections (1)**: A box containing one connection: 'NI6504-2231290-igw'.

Lines connect the subnets to the route tables, and the route tables to the network connection.

2: Create VPC Security Group for Public Web Server and attach to the VPC you have just created

Use the following Information:

- Security group name: NI6504-2231290-securitygroup
- Description: NI6504 Security Group

Inbound Rules:

- Allow SSH access to ONLY your PC IP Address.
(Note, if you are behind a DNAT Wireless setup, you may have to use 0.0.0.0/0. **Remember that this is very unsafe and will only be used in for this assessment**)
- Allow HHTP from anywhere.

The screenshot shows the AWS Management Console interface for a security group. The breadcrumb navigation at the top indicates the path: VPC > Security Groups > sg-08c6cf474fc67af46 - NI6504-2231290-securitygroup. The main heading is "sg-08c6cf474fc67af46 - NI6504-2231290-securitygroup" with an "Actions" dropdown menu to its right.

The "Details" section contains the following information:

Security group name NI6504-2231290-securitygroup	Security group ID sg-08c6cf474fc67af46	Description NI6504 Security Group	VPC ID vpc-07f83ca94987eaf1d
Owner 565109282158	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Below the details are tabs for "Inbound rules", "Outbound rules", and "Tags". The "Inbound rules" tab is selected, showing "Inbound rules (2)". There are buttons for "Manage tags" and "Edit inbound rules". A search bar is present with the text "Search".

The inbound rules are listed in a table:

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0717aa2223d1a3e...	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0958a1359b60ae0...	IPv4	HTTP	TCP	80

3: Create VPC Security Group for Private DB Instance.

Use the following information:

- Security group name: NI6504-2231290-db-securitygroup
- Description: NI6504 DB Instance Security Group

Inbound Rules:

- Type: MySQL/Aurora
- Source: The identifier of the security group you created previously

The screenshot displays the AWS Management Console interface for a VPC Security Group. The breadcrumb navigation shows the path: VPC > Security Groups > sg-0d5f392646f054f76 - NI6504-2231290-db-securitygroup. The main title is "sg-0d5f392646f054f76 - NI6504-2231290-db-securitygroup" with an "Actions" dropdown menu.

Details

Security group name NI6504-2231290-db-securitygroup	Security group ID sg-0d5f392646f054f76	Description NI6504 DB Instance Security Group	VPC ID vpc-07f83ca94987eaf1d
Owner 565109282158	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

Inbound rules (1)

Search

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
<input type="checkbox"/>	-	sg-09dd4c2431250fbf9	-	MYSQL/Aurora	TCP	3306	sg-08c6cf474fc67af46...

4: Create a DB Subnet Group from the RDS Dashboard

Use the following information:

- The private Subnet ID Information for each of the 2 private subnets that have been created.
- Name: ni6504-2231290-db-subnetgroup
- Description: NI6504 DB Subnet Group
- Chose the correct VPC
- Select 2 Availability Zones and add one of the Private Subnets to each

[Alt+S] ⓘ 🔔 ⓘ ⚙️ N. Virginia ▼ voclabs/user3388989=Andrew_Graff @ 5651-0928-2158 ▼

RDS > Subnet groups > ni6504-2231290-db-subnetgroup

ni6504-2231290-db-subnetgroup

Subnet group details

VPC ID

vpc-07f83ca94987eaf1d ⓘ

ARN

arn:aws:rds:us-east-1:565109282158:subgrp:ni6504-2231290-db-subnetgroup

Supported network types

IPv4

Description

NI6504 DB Subnet Group

Subnets (2)

Availability zone	Subnet ID	CIDR block
us-east-1b	subnet-05ff45694c9ee9cda ⓘ	10.90.144.0/20
us-east-1a	subnet-04c358809dce8bf20 ⓘ	10.90.128.0/20

5: Create and Launch EC2 Instance

Use the following information:

- Name and Tag: NI6504-2231290-ec2-instance-web-server
- Application and OS Image: Amazon Linux / Amazon Linux 2023 AMI
- Type: t2.micro
- Create a new Key Pair (Login) called NI6504
- Allow SSH traffic from either: your IP or 0.0.0.0/0 if you are behind a wireless router (Only for this assignment. This is not a good security practice)
- Allow HTTPs Traffic
- Allow HTTP Traffic

The screenshot displays the AWS Management Console interface for an EC2 instance. The top navigation bar shows the user is logged in as 'voclabs/user3388989=Andrew_Graff' in the 'N. Virginia' region. The breadcrumb trail indicates the path: EC2 > Instances > i-0a20e481c0f340fa1.

The main section is titled 'Instance summary for i-0a20e481c0f340fa1 (NI6504-2231290-ec2-instance-web-server)'. It includes a 'Connect' button, 'Instance state' dropdown (set to 'Running'), and an 'Actions' dropdown. The summary is organized into three columns:

- Left Column:** Instance ID (i-0a20e481c0f340fa1), IPv6 address (none), Hostname type (IP name: ip-172-31-33-3.ec2.internal), Answer private resource DNS name (IPv4 (A)), Auto-assigned IP address (35.153.130.76 [Public IP]), IAM Role (none), and IMDSv2 (Required).
- Middle Column:** Public IPv4 address (35.153.130.76), Instance state (Running), Private IP DNS name (ip-172-31-33-3.ec2.internal), Instance type (t2.micro), VPC ID (vpc-02cdd89f8d2a1ce16), Subnet ID (subnet-09888155f03a9d4f0), and Instance ARN (arn:aws:ec2:us-east-1:565109282158:instance/i-0a20e481c0f340fa1).
- Right Column:** Private IPv4 addresses (172.31.33.3), Public IPv4 DNS (ec2-35-153-130-76.compute-1.amazonaws.com), Elastic IP addresses (none), AWS Compute Optimizer finding (Opt-in to AWS Compute Optimizer for recommendations), and Auto Scaling Group name (none).

Below the summary is a tabbed interface with 'Details' selected. The 'Instance details' section shows:

- Platform:** Amazon Linux (Inferred)
- Platform details:** Linux/UNIX
- Stop protection:** Disabled
- AMI ID:** ami-0ff1b9a61dec8a5f
- AMI name:** al2023-ami-2023.5.20241001.1-kernel-6.1-x86_64
- Launch time:** Wed Oct 09 2024 18:25:24 GMT+1300 (New Zealand Daylight Time)
- Monitoring:** disabled
- Termination protection:** Disabled
- AMI location:** amazon/al2023-ami-2023.5.20241001.1-kernel-6.1-x86_64

6: Create MariaDB database in RDS Console

Use the following Information:

- Template: Free Tier
- DB Instance Identifier : NI6504-Andrew
- User Name : Your_Andrew
- Password : NI6504-2231290
- Instance: Burstable
- db.t3.micro
- Connectivity: Connect to EC2 Compute Resource
- EC2 Instance: Your EC2 Instance you have created
- Initial DB Name: NI6504

The screenshot shows the AWS RDS console interface for a MariaDB instance named **ni6504-2231290**. The instance is in the **us-east-1** region, **us-east-1a** Availability Zone, and is a **db.t3.micro** class instance. The CPU usage is at 46.39%.

The **Connectivity & security** tab is active, showing the following details:

Endpoint & port	Networking	Security
Endpoint ni6504-2231290.czdv8hvfmkfl.us-east-1.rds.amazonaws.com	Availability Zone us-east-1a	VPC security groups rds-ec2-1 (sg-06e052aa14499a3ef) Active
Port 3306	VPC vpc-02cdd89f8d2a1ce16	Publicly accessible No
	Subnet group rds-ec2-db-subnet-group-1	Certificate authority Info rds-ca-rsa2048-g1
	Subnets subnet-0120f97fd3fafc4d1 subnet-0b0bdde3154172799 subnet-059dfcf15053c1bba subnet-0a2c01253109b519d subnet-0318897e5a914d920	Certificate authority date May 26, 2061, 11:34 (UTC+12:00)
	Network type IPv4	DB instance certificate expiration date October 09, 2025, 18:32 (UTC+13:00)

(Take note of your Endpoint and Port)

ni6504-2231290.czdv8hvfmkfl.us-east-1.rds.amazonaws.com port 3306

7: Install Apache Webserver with PHP and MariaDB

(use the 'systemctl status' command for apache2 here)

```
aws Services Search [Alt+S] N. Virginia voclabs/user3388989=Andrew_Graff @ 5651-0928-2158

Verifying : perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 21/22
Verifying : perl-base-2.27-477.amzn2023.0.6.noarch 22/22

ninstalled:
mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64 mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch
mariadb105-3:10.5.25-1.amzn2023.0.1.x86_64 mariadb105-backup-3:10.5.25-1.amzn2023.0.1.x86_64
mariadb105-common-3:10.5.25-1.amzn2023.0.1.x86_64 mariadb105-cracklib-password-check-3:10.5.25-1.amzn2023.0.1.x86_64
mariadb105-errmsg-3:10.5.25-1.amzn2023.0.1.x86_64 mariadb105-gssapi-server-3:10.5.25-1.amzn2023.0.1.x86_64
mariadb105-server-3:10.5.25-1.amzn2023.0.1.x86_64 mariadb105-server-utils-3:10.5.25-1.amzn2023.0.1.x86_64
mysql-selinux-1.0.4-2.amzn2023.0.3.noarch perl-B-1.80-477.amzn2023.0.6.x86_64
perl-DBD-MariaDB-1.22-1.amzn2023.0.4.x86_64 perl-DBI-1.643-7.amzn2023.0.3.x86_64
perl-Data-Dumper-2.174-460.amzn2023.0.2.x86_64 perl-File-Copy-2.34-477.amzn2023.0.6.noarch
perl-FileHandle-2.03-477.amzn2023.0.6.noarch perl-Math-BigInt-1:1.9998.39-2.amzn2023.0.2.noarch
perl-Math-BigRat-0.2614-458.amzn2023.0.2.noarch perl-Math-Complex-1.59-477.amzn2023.0.6.noarch
perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 perl-base-2.27-477.amzn2023.0.6.noarch

complete!
ec2-user@ip-172-31-33-3 ~]$ sudo systemctl start mariadb
ec2-user@ip-172-31-33-3 ~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
ec2-user@ip-172-31-33-3 ~]$ sudo systemctl status httpd
httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
Drop-In: /usr/lib/systemd/system/httpd.service.d
└─php-fpm.conf
Active: active (running) since Wed 2024-10-09 06:09:12 UTC; 3min 17s ago
Docs: man:httpd.service(8)
Main PID: 28000 (httpd)
Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
Tasks: 177 (limit: 1112)
Memory: 13.0M
CPU: 140ms
CGroup: /system.slice/httpd.service
├─28000 /usr/sbin/httpd -DFOREGROUND
├─28002 /usr/sbin/httpd -DFOREGROUND
├─28003 /usr/sbin/httpd -DFOREGROUND
├─28004 /usr/sbin/httpd -DFOREGROUND
└─28005 /usr/sbin/httpd -DFOREGROUND

Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal httpd[28000]: Server configured, listening on: port 80
ec2-user@ip-172-31-33-3 ~]$
```

8: PHP

```
aws Services Search [Alt+S] N. Virginia voclabs/user3388989=Andrew_Graff @ 5651-0928-2158

perl-FileHandle-2.03-477.amzn2023.0.6.noarch perl-Math-BigInt-1:1.9998.39-2.amzn2023.0.2.noarch
perl-Math-BigRat-0.2614-458.amzn2023.0.2.noarch perl-Math-Complex-1.59-477.amzn2023.0.6.noarch
perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 perl-base-2.27-477.amzn2023.0.6.noarch

Complete!
ec2-user@ip-172-31-33-3 ~]$ sudo systemctl start mariadb
ec2-user@ip-172-31-33-3 ~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
ec2-user@ip-172-31-33-3 ~]$ sudo systemctl status httpd
httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
Drop-In: /usr/lib/systemd/system/httpd.service.d
└─php-fpm.conf
Active: active (running) since Wed 2024-10-09 06:09:12 UTC; 3min 17s ago
Docs: man:httpd.service(8)
Main PID: 28000 (httpd)
Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
Tasks: 177 (limit: 1112)
Memory: 13.0M
CPU: 140ms
CGroup: /system.slice/httpd.service
├─28000 /usr/sbin/httpd -DFOREGROUND
├─28002 /usr/sbin/httpd -DFOREGROUND
├─28003 /usr/sbin/httpd -DFOREGROUND
├─28004 /usr/sbin/httpd -DFOREGROUND
└─28005 /usr/sbin/httpd -DFOREGROUND

Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal httpd[28000]: Server configured, listening on: port 80
ec2-user@ip-172-31-33-3 ~]$ sudo yum install -y php php-mysqld
Last metadata expiration check: 0:48:58 ago on Wed Oct 9 05:25:57 2024.
Package php8.3-8.3.10-1.amzn2023.0.1.x86_64 is already installed.
Package php8.3-mysqld-8.3.10-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
ec2-user@ip-172-31-33-3 ~]$ php -version
PHP 8.3.10 (cli) (built: Jul 30 2024 13:44:37) (NTS gcc x86_64)
Copyright (c) The PHP Group
Zend Engine v4.3.10, Copyright (c) Zend Technologies
with Zend OPcache v8.3.10, Copyright (c), by Zend Technologies
ec2-user@ip-172-31-33-3 ~]$
```

9: MariaDB

```
aws Services Search [Alt+S] N. Virginia voclabs/user3388989=Andrew_Graff @ 5651-0928-2158

└─28004 /usr/sbin/httpd -DFOREGROUND
└─28005 /usr/sbin/httpd -DFOREGROUND

Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal httpd[28000]: Server configured, listening on: port 80
[ec2-user@ip-172-31-33-3 ~]$ sudo yum install -y php php-mysqldb
Last metadata expiration check: 0:48:50 ago on Wed Oct 9 05:25:57 2024.
Package php8.3-8.3.10-1.amzn2023.0.1.x86_64 is already installed.
Package php8.3-mysqldb-8.3.10-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-33-3 ~]$ php -version
PHP 8.3.10 (cli) (built: Jul 30 2024 13:44:37) (NTS gcc x86_64)
Copyright (c) The PHP Group
Zend Engine v4.3.10, Copyright (c) Zend Technologies
with Zend OPcache v8.3.10, Copyright (c), by Zend Technologies
[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl enable mariadb
[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
   Active: active (running) since Wed 2024-10-09 06:11:52 UTC; 5min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 29657 (mariadb)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 1112)
    Memory: 66.0M
       CPU: 340ms
    CGroup: /system.slice/mariadb.service
            └─29657 /usr/libexec/mariadb --basedir=/usr

Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: The second is mysql@localhost, it has no pas
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: you need to be the system 'mysql' user to c
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: After connecting you can set the password,
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: able to connect as any of these users with
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: See the MariaDB Knowledgebase at https://ma
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: Please report any problems at https://mari
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: The latest information about MariaDB is ava
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: Consider joining MariaDB's strong and vibr
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: https://mariadb.org/get-involved/
Oct 09 06:11:52 ip-172-31-33-3.ec2.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.
```

(use a CLI command to confirm that MariaDB is functional here)

10: Start the Apache2 Service.

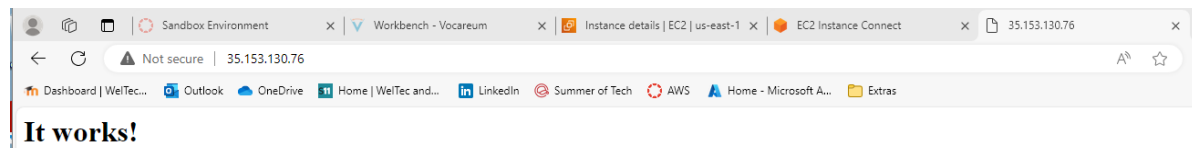
```
aws Services Search [Alt+S] N. Virginia voclabs/user3388989=Andrew_Graff @ 5651-0928-2158

[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl enable mariadb
[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
   Active: active (running) since Wed 2024-10-09 06:11:52 UTC; 5min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 29657 (mariadb)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 1112)
    Memory: 66.0M
       CPU: 340ms
    CGroup: /system.slice/mariadb.service
            └─29657 /usr/libexec/mariadb --basedir=/usr

Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: The second is mysql@localhost, it has no pas
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: you need to be the system 'mysql' user to c
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: After connecting you can set the password,
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: able to connect as any of these users with
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: See the MariaDB Knowledgebase at https://ma
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: Please report any problems at https://mari
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: The latest information about MariaDB is ava
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: Consider joining MariaDB's strong and vibr
Oct 09 06:11:51 ip-172-31-33-3.ec2.internal mariadb-prepare-db-dir[29614]: https://mariadb.org/get-involved/
Oct 09 06:11:52 ip-172-31-33-3.ec2.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.
lines 1-23/23 (END)
[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-33-3 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Wed 2024-10-09 06:09:12 UTC; 10min ago
     Docs: man:httpd.service(8)
   Main PID: 28000 (httpd)
    Status: "Total requests: 1; Idle/Busy workers 100/0;Requests/sec: 0.00159; Bytes served/sec: 0 B/sec"
     Tasks: 177 (limit: 1112)
    Memory: 13.1M
       CPU: 377ms
    CGroup: /system.slice/httpd.service
            └─28000 /usr/sbin/httpd -DFOREGROUND
              └─28002 /usr/sbin/httpd -DFOREGROUND
                └─28003 /usr/sbin/httpd -DFOREGROUND
                  └─28004 /usr/sbin/httpd -DFOREGROUND
                    └─28005 /usr/sbin/httpd -DFOREGROUND

Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal httpd[28000]: Server configured, listening on: port 80
[ec2-user@ip-172-31-33-3 ~]$
```

11. Access the Web Application and make sure the Web Server Instance we can be accessed from the internet.



(This will be the Apache2 Default Webpage)

12: Trouble Shoot CLI connection to google.com from your EC2 Instance

```
aws Services Search [Alt+S] N. Virginia voclabs/user3388989=Andrew_Graff @ 5651-0928-2158
cgroup: /system.slice/httpd.service
└─28000 /usr/sbin/httpd -DFOREGROUND
└─28002 /usr/sbin/httpd -DFOREGROUND
└─28003 /usr/sbin/httpd -DFOREGROUND
└─28004 /usr/sbin/httpd -DFOREGROUND
└─28005 /usr/sbin/httpd -DFOREGROUND

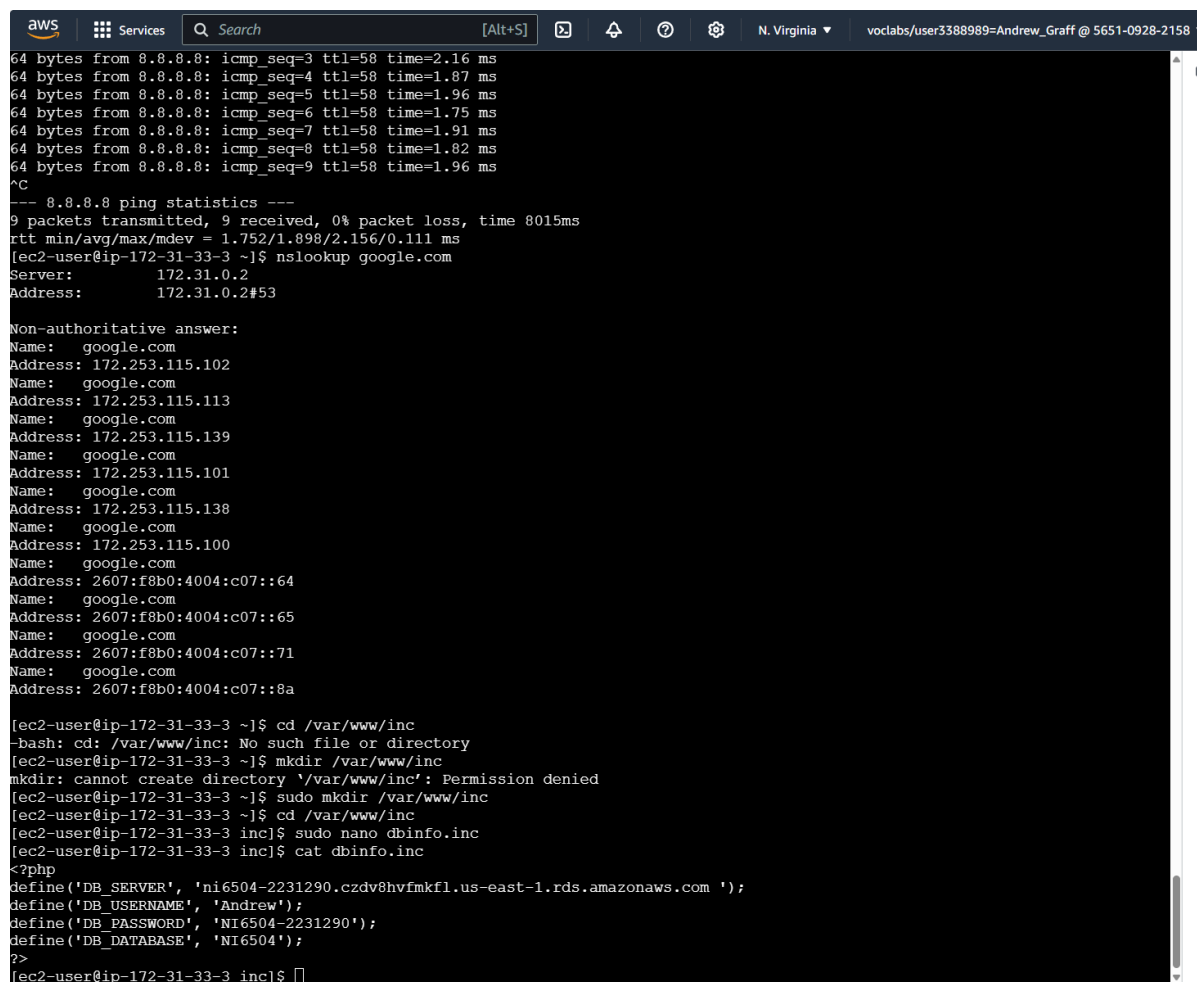
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 09 06:09:12 ip-172-31-33-3.ec2.internal httpd[28000]: Server configured, listening on: port 80
[ec2-user@ip-172-31-33-3 ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=1.81 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=2.16 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=1.87 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=58 time=1.96 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=58 time=1.75 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=58 time=1.91 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=58 time=1.82 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=58 time=1.96 ms
^C
-- 8.8.8.8 ping statistics --
9 packets transmitted, 9 received, 0% packet loss, time 8015ms
rtt min/avg/max/mdev = 1.752/1.898/2.156/0.111 ms
[ec2-user@ip-172-31-33-3 ~]$ nslookup google.com
Server:      172.31.0.2
Address:     172.31.0.2#53

Non-authoritative answer:
Name:   google.com
Address: 172.253.115.102
Name:   google.com
Address: 172.253.115.113
Name:   google.com
Address: 172.253.115.139
Name:   google.com
Address: 172.253.115.101
Name:   google.com
Address: 172.253.115.138
Name:   google.com
Address: 172.253.115.100
Name:   google.com
Address: 2607:f8b0:4004:c07::64
Name:   google.com
Address: 2607:f8b0:4004:c07::65
Name:   google.com
Address: 2607:f8b0:4004:c07::71
Name:   google.com
Address: 2607:f8b0:4004:c07::8a
```

13: Create new file in /var/www/inc called dbinfo.inc and add the following lines:

```
<?phpdefine('DB_SERVER', 'ni6504-2231290.czd8hvfmkfl.us-east-1.rds.amazonaws.com ');
define('DB_USERNAME', 'Andrew');
define('DB_PASSWORD', 'NI6504-2231290');
define('DB_DATABASE', 'NI6504');?>
```

Note: The items in RED are specific to your installation. You must use your configuration information.

A screenshot of an AWS CloudShell terminal window. The top bar shows the AWS logo, 'Services', a search bar, and the user's profile 'voclabs/user3388989=Andrew_Graff @ 5651-0928-2158'. The terminal output shows a series of ping commands to 8.8.8.8, followed by ping statistics. Then, a series of 'nslookup google.com' commands are shown, displaying IP addresses for google.com. Finally, the user navigates to /var/www/inc, creates the directory, and creates a file dbinfo.inc with the provided PHP code. The code defines database connection parameters for an Amazon RDS instance.

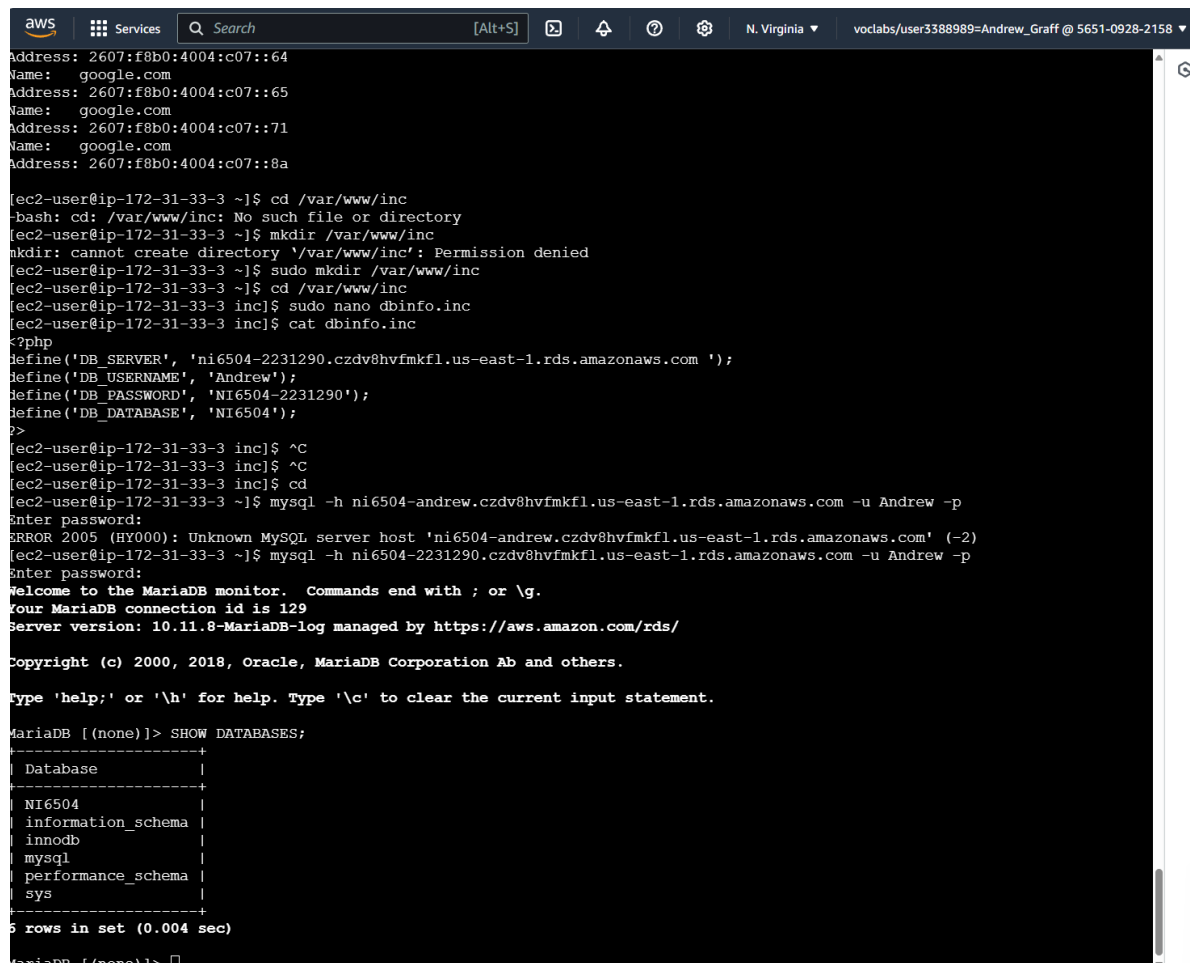
```
aws Services Search [Alt+S] N. Virginia voclabs/user3388989=Andrew_Graff @ 5651-0928-2158
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=2.16 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=1.87 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=58 time=1.96 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=58 time=1.75 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=58 time=1.91 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=58 time=1.82 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=58 time=1.96 ms
^C
--- 8.8.8.8 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8015ms
rtt min/avg/max/mdev = 1.752/1.898/2.156/0.111 ms
[ec2-user@ip-172-31-33-3 ~]$ nslookup google.com
Server:      172.31.0.2
Address:     172.31.0.2#53

Non-authoritative answer:
Name:   google.com
Address: 172.253.115.102
Name:   google.com
Address: 172.253.115.113
Name:   google.com
Address: 172.253.115.139
Name:   google.com
Address: 172.253.115.101
Name:   google.com
Address: 172.253.115.138
Name:   google.com
Address: 172.253.115.100
Name:   google.com
Address: 2607:f8b0:4004:c07::64
Name:   google.com
Address: 2607:f8b0:4004:c07::65
Name:   google.com
Address: 2607:f8b0:4004:c07::71
Name:   google.com
Address: 2607:f8b0:4004:c07::8a

[ec2-user@ip-172-31-33-3 ~]$ cd /var/www/inc
-bash: cd: /var/www/inc: No such file or directory
[ec2-user@ip-172-31-33-3 ~]$ mkdir /var/www/inc
mkdir: cannot create directory '/var/www/inc': Permission denied
[ec2-user@ip-172-31-33-3 ~]$ sudo mkdir /var/www/inc
[ec2-user@ip-172-31-33-3 ~]$ cd /var/www/inc
[ec2-user@ip-172-31-33-3 inc]$ sudo nano dbinfo.inc
[ec2-user@ip-172-31-33-3 inc]$ cat dbinfo.inc
<?php
define('DB_SERVER', 'ni6504-2231290.czd8hvfmkfl.us-east-1.rds.amazonaws.com ');
define('DB_USERNAME', 'Andrew');
define('DB_PASSWORD', 'NI6504-2231290');
define('DB_DATABASE', 'NI6504');
?>
[ec2-user@ip-172-31-33-3 inc]$
```

(Cat your dbinfo.inc file)

14. Troubleshoot the connection to MariaDB RDS and check existing database on RDS using a *mysql* CLI command:



```
aws Services Search [Alt+S] N. Virginia voclabs/user3388989=Andrew_Graff @ 5651-0928-2158
Address: 2607:f8b0:4004:c07::64
Name: google.com
Address: 2607:f8b0:4004:c07::65
Name: google.com
Address: 2607:f8b0:4004:c07::71
Name: google.com
Address: 2607:f8b0:4004:c07::8a

[ec2-user@ip-172-31-33-3 ~]$ cd /var/www/inc
bash: cd: /var/www/inc: No such file or directory
[ec2-user@ip-172-31-33-3 ~]$ mkdir /var/www/inc
mkdir: cannot create directory '/var/www/inc': Permission denied
[ec2-user@ip-172-31-33-3 ~]$ sudo mkdir /var/www/inc
[ec2-user@ip-172-31-33-3 ~]$ cd /var/www/inc
[ec2-user@ip-172-31-33-3 inc]$ sudo nano dbinfo.inc
[ec2-user@ip-172-31-33-3 inc]$ cat dbinfo.inc
<?php
define('DB_SERVER', 'ni6504-2231290.czdv8hvfmkfl.us-east-1.rds.amazonaws.com ');
define('DB_USERNAME', 'Andrew');
define('DB_PASSWORD', 'NI6504-2231290');
define('DB_DATABASE', 'NI6504');
?>
[ec2-user@ip-172-31-33-3 inc]$ ^C
[ec2-user@ip-172-31-33-3 inc]$ ^C
[ec2-user@ip-172-31-33-3 inc]$ cd
[ec2-user@ip-172-31-33-3 ~]$ mysql -h ni6504-andrew.czdv8hvfmkfl.us-east-1.rds.amazonaws.com -u Andrew -p
Enter password:
ERROR 2005 (HY000): Unknown MySQL server host 'ni6504-andrew.czdv8hvfmkfl.us-east-1.rds.amazonaws.com' (-2)
[ec2-user@ip-172-31-33-3 ~]$ mysql -h ni6504-2231290.czdv8hvfmkfl.us-east-1.rds.amazonaws.com -u Andrew -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 129
Server version: 10.11.8-MariaDB-log managed by https://aws.amazon.com/rds/

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

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

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| NI6504   |
| information_schema |
| innodb  |
| mysql   |
| performance_schema |
| sys     |
+-----+
5 rows in set (0.004 sec)

MariaDB [(none)]>
```

For the following either include a screenshot of successful completion of each task, or a brief description of what you have attempted to get each step ‘working’:

15. Create AMI from Web Application Instance

[Alt+S]

N. Virginia

voclabs/user3388989=Andrew_Graff @ 5651-0928-2158

EC2 > AMIs > ami-0b9c254e889e54cad

Image summary for ami-0b9c254e889e54cad

EC2 Image Builder

Actions

Launch instance from AMI

AMI ID

ami-0b9c254e889e54cad

Image type

machine

Platform details

Linux/UNIX

Root device type

EBS

AMI name

NI6504-AMI

Owner account ID

565109282158

Architecture

x86_64

Usage operation

RunInstances

Root device name

/dev/xvda

Status

Pending

Source

565109282158/NI6504-AMI

Virtualization type

hvm

Boot mode

uefi-preferred

State reason

-

Creation date

Wed Oct 09 2024 20:01:00 GMT+1300 (New Zealand Daylight Time)

Kernel ID

-

Description

-

Product codes

-

RAM disk ID

-

Deprecation time

-

Last launched time

-

Block devices

/dev/xvda=snap-04786da7a05fbed6b:8:true:gp3

Deregistration protection

Disabled

16. Create Target Group

[Alt+S]

N. Virginia

voclabs/user3388989=Andrew_Graff @ 5651-0928-2158

EC2 > Target groups > NI6504

NI6504

Actions

Details

arn:aws:elasticloadbalancing:us-east-1:565109282158:targetgroup/NI6504/ef2eb79ab5b172f7

Target type

Instance

Protocol : Port

HTTP: 80

Protocol version

HTTP1

VPC

vpc-07f83ca94987eaf1d

IP address type

IPv4

Load balancer

None associated

0

0

0

0

0

0

Total targets

Healthy

Unhealthy

Unused

Initial

Draining

0 Anomalous

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (0)

Info

Anomaly mitigation: Not applicable

Deregister

Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

17. Create Load Balancers

EC2 > Load balancers > NI6504

NI6504

Details

Load balancer type
Application

Scheme
Internet-facing

Status
Provisioning

Hosted zone
Z35SXDOTRQ7X7K

VPC
vpc-07f83ca94987eaf1d

Availability Zones
subnet-050cbc5e1e03cd12b us-east-1b (use1-az1)
subnet-069128587f2bbc234 us-east-1a (use1-az6)

Load balancer IP address type
IPv4

Date created
October 9, 2024, 20:28 (UTC+13:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:565109282158:loadbalancer/app/NI6504/aea97d862dfe5839

DNS name
NI6504-755546549.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules

Network mapping

Resource map - new

Security

Monitoring

Integrations

Attributes

Tags

Listeners and rules (1)

Manage rules

Manage listener

Add listener

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Filter listeners

Protocol:Port

Default action

Rules

ARN

Security policy

Default SSL/TLS certificate

mTLS

HTTP:80

Forward to target group

- NI6504 1 (100%)
- Target group stickiness: Off

1 rule

ARN

Not applicable

Not applicable

Not applicable

18. Create Launch Template

EC2 > ... > NI6504

NI6504 (lt-0f420127cff4798e9)

Actions

Delete template

Launch template details

Launch template ID
lt-0f420127cff4798e9

Launch template name
NI6504

Default version
1

Owner
arn:aws:sts::565109282158:assumed-role/voclabs/user3388989=Andrew_Graff

Details

Versions

Template tags

Launch template version details

Actions

Delete template version

Version
1 (Default)

Description
-

Date created
2024-10-09T07:31:53.000Z

Created by
arn:aws:sts::565109282158:assumed-role/voclabs/user3388989=Andrew_Graff

Instance details

Storage

Resource tags

Network interfaces

Advanced details

AMI ID
ami-0b9c254e889e54cad

Instance type
t2.micro

Availability Zone
-

Key pair name
-

Security groups
-

Security group IDs
sg-08c6cf474fc67af46

19. Create Auto Scaling Group

Services

Search

[Alt+S]

N. Virginia

voclabs/user:3388989=Andrew_Graff @ 5651-0928-2158

EC2 > Auto Scaling groups > NI6504

NI6504

DetailsActivityAutomatic scalingInstance managementMonitoringInstance refresh

Group details

Edit

Auto Scaling group name NI6504	Desired capacity 2	Desired capacity type Units (number of instances)	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:565109282158:autoScalingGroup:c2f4f0a5-2c96-4ef1-94dc-ae93dddfa26e:autoScalingGroupName/NI6504
Date created Wed Oct 09 2024 20:55:59 GMT+1300 (New Zealand Daylight Time)	Minimum capacity 2	Status -	
	Maximum capacity 6		

Launch template

Edit

Launch template lt-0a9af5f0e525b62d0 NI6504-LaunchTemplate	AMI ID ami-0fff1b9a61dec8a5f	Instance type t2.micro	Owner arn:aws:sts::565109282158:assumed-role/voclabs/user:3388989=Andrew_Graff
Version Default	Security groups -	Security group IDs sg-01c801377ffe6686e	Create time Wed Oct 09 2024 20:52:26 GMT+1300 (New Zealand Daylight Time)
Description -	Storage (volumes) -	Key pair name NI6504	Request Spot Instances No

View details in the launch template console

20. Test Web App access via Load Balancer URL

Sandbox EnvironmentWorkbench - VocareumModify template (Create new ver...Load balancer details | EC2 | us-...35.153.130.76503 Service Temporarily Unavailable

Not secure | ni6504-755546549.us-east-1.elb.amazonaws.com

Dashboard | WellTec...OutlookOneDriveHome | WellTec and...LinkedInSummer of TechAWSHome - Microsoft A...Extras

503 Service Temporarily Unavailable

Reflection

The most challenging task: Task 19

This was the most challenging task because after several attempts, you've carefully got everything up and running and the screenshots are all consistent with the steps. Time restrictions increased pressure of having to re-do the entire screenshots. This was most often the case. The auto scaling often led an unauthorised message disallowing the use the AMI image as the launch template.

It required several troubleshooting techniques. Multiple copies of the AMI launch templates were configured with efforts the auto scaling group would be created however not so the case. The resolution to this was changing the Launch Template version details and associating the template with a mirrored version of the AMI to eventually create the auto group scaling with the security groups .

Least challenging task: Task 1

This task become the most memorised after rehearsing the steps again and again. The instructions of this first part of report were straight forward which added to the ease. Creating a VPC for use with DB instance, was easy given (we selected VPC and more) to identify the extra settings necessary to fill. Task 1 and Task 2 were relatively the same. The web application is a nice GUI and ease of use by completing each area step by step as instructed.

- **Discussion**

Discussion of three troubleshooting techniques:

1. The first troubleshooting the connection of CLI to google.com. This involved ensuring the configuration set up of our Apache Webserver, with PHP and MariaDB was correct. We needed to test and confirm these services were running in our CLI. If these weren't running, then an update would need to be performed or correctly installing the packages needed.
2. The second troubleshooting technique was bringing up the console. This was difficult for me and often led to being timed out after three hours of trying to find resolution. What we did was ensure the security groups were configured correctly. A change occurred here where at first, we allowed the traffic from 0.0.0.0/0 and My IP and enabled us access to the CLI console. So having the two SSH ports implemented traffic to having the console up and running.
3. The third troubleshooting was importing and exporting database data. Early in the steps, it was advised to take note of our endpoint as this become important towards the end. An issue that arose from this was when re-doing the sandbox activities again, the end point address will change. So, an approach was to carefully do the entire practical - so at the stage of capturing the endpoint, we would insert the correct link into our config file that was specific to our data base installation.

Each Task requires an explanation of the Troubleshooting Technique, and the outcome of the action you have taken.

- Demonstrate that you can connect via CLI to google.com:

1. Required to set up EC2 instance to allow SSH traffic from My IP 202.2.13.130/32 instead of Anywhere during networking settings.

2. Now to create Maria DB in RDS console which took a little bit of time and not note the endpoint and port number

3. EC2 instance connect service IP addresses are not authorized as this is where our trouble shooting requirement begins



EC2 Instance Connect service IP addresses are not authorized

Port 22 (SSH) is authorized in [your security group](#). However, to use EC2 Instance Connect, it is recommended to also authorize port 22 for the EC2 Instance Connect service IP addresses in your Region: 18.206.107.24/29. [Learn more](#).

Failed to connect to your instance

EC2 Instance Connect is unable to connect to your instance. Ensure your instance network settings are configured correctly for EC2 Instance Connect. For more information, see EC2 Instance Connect Prerequisites at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-connect-prerequisites.html>.

4. I delete the HTTPS from the inbound rules in the security group and add another SSH 22 port that allows connection from anywhere. So, we have two inbound rules for port 22 (HTTP from anywhere and My IP).

Inbound rules [Info](#)

Inbound rule 1

Delete

Security group rule ID
sgr-0267eed792d9fb861

Type [Info](#)

SSH

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Custom

Source [Info](#)

202.2.13.130/32

Description - optional [Info](#)

Inbound rule 2

Delete

Security group rule ID
sgr-00ed406b438281076

Type [Info](#)

SSH

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Custom

Source [Info](#)

0.0.0.0/0

5. Finally, we enter mySQL CLI command script “SHOW DATABASES;” to list our databases. Once this is retrieved we can confirm RDS connectivity with our instance.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 129
Server version: 10.11.8-MariaDB-log managed by https://aws.amazon.com/rds/

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

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

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| NI6504   |
| information_schema |
| innodb   |
| mysql    |
| performance_schema |
| sys      |
+-----+
6 rows in set (0.004 sec)
```

- Demonstrate that you can import and export Database Data
 1. Export some data to our RDS database we can do this with the command below. We enter the below command to create a back up of the database “mysqldump” and connecting this to database using name and password. The NI6504 database will be backed up to the databasedata.sql

```
mysqldump -u Andrew -p NI6504 > databasedata.sql
```

2. We can double check the first 10 lines of the data we’ve backed up to double check we have some database data.

```
head -n 10 databasedata.sql
```

3. We create a new database and give it a name so we can input data. The mysql command will prompt the password set and execute the SQL statement from the command line. The new database will be called Backup

```
mysql -u Andrew -p -e “CREATE DATABASE Backup;”
```

4. Import some data using the SQL command, that is from the databasedata.sql

```
mysql -u Andrew -p Backup < databasedata.sql
```

5. Check database to see if the data has been imported

```
mysql -u username -p -e “SHOW TABLES IN Backup;
```