Securing A Cloud-Based E-Commerce Platform

Assessment Report

Version 1.0

Prepared by Group 12

Bharath Dibbadahalli Hanumanthappa

Brandan Deguzman

Ty Newkirk

Siddharth Muragundla

Table of Contents

1.0 Executive Summary:

1.1 Scope:

The purpose of this security assessment was to assess the security posture of an E-commerce applications cloud infrastructure. By doing this we can identify key areas of weakness that the organization can improve upon and ultimately help ensure the confidentiality, integrity, and availability of the cloud infrastructure.

We were tasked with assessing the cloud infrastructure for an E-commerce application. This included analyzing the CloudFormation template and the EC2 instance running the Juice Shop application after deploying the template. Additionally, the key areas that were examined included access control and IAM, data security, network security, disaster recovery, vulnerability management, and logging and monitoring.

1.2 Objective:

The objective of this security assessment was to identify key risks and vulnerabilities in several areas for an E-commerce application cloud infrastructure. These areas included data security, disaster recovery, network security, and more. By identifying the risks and vulnerabilities present within the infrastructure, we can help ensure that the organization is ensuring the confidentiality, integrity, and availability of the application and its data.

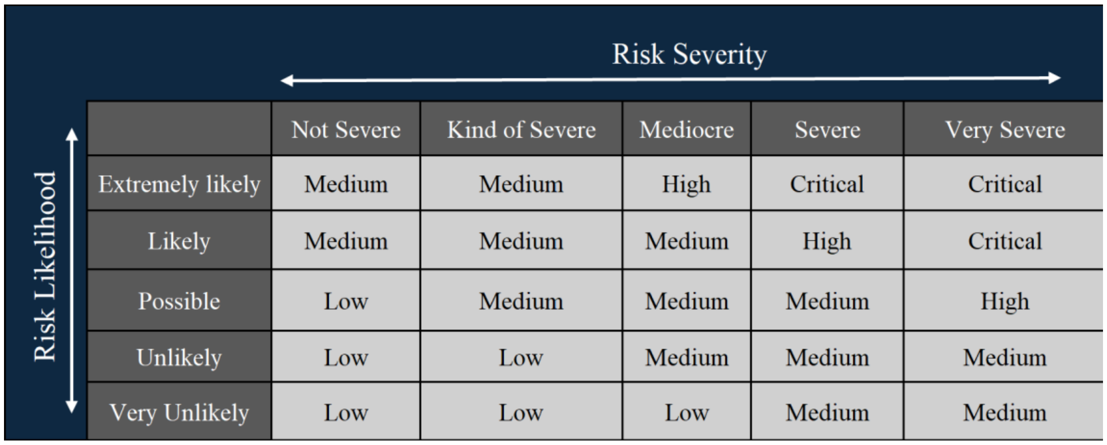
1.3 Methodology:

To conduct this assessment there were several key methods that were used. One key method was by using the OWASP Top 10 Framework to identify vulnerabilities and risks found present within the CloudFormation template and within the Juice Shop application that was being run on the EC2 instance. It allowed us to quickly identify the most common web application security issues that could affect the confidentiality, integrity, and availability of the organization. By using the framework we were also able to offer effective remediation strategies and steps that the organization can take to help improve the overall security posture of the cloud infrastructure. Another key method was by comparing AWS CloudFormation and EC2 best practices to the current E-commerce application infrastructure. AWS has lots of online documentation on best practices for ensuring confidentiality, integrity, and availability for all their services including CloudFormation, EC2, and more. They list and describe features or configurations that you should implement and exact steps for doing so. After reading the applicable documentation, we were able to quickly identify key areas that were present but were lacking the correct configurations or areas that the cloud infrastructure was completely missing. There were several other methods used but these were the two main ones that allowed us to identify key risks and vulnerabilities present within the E-commerce cloud infrastructure.

1.4 Risk Matrix:

Throughout the security assessment, we have assessed the severity of discovered risks by using the following risk matrix table below. It combines the severity of the risk and the likelihood of the risk occurring to calculate an overall risk value of either low, medium, high, or critical. The calculations for the overall risk value are broken down in the table below.

Table 1.1



Risk Severity

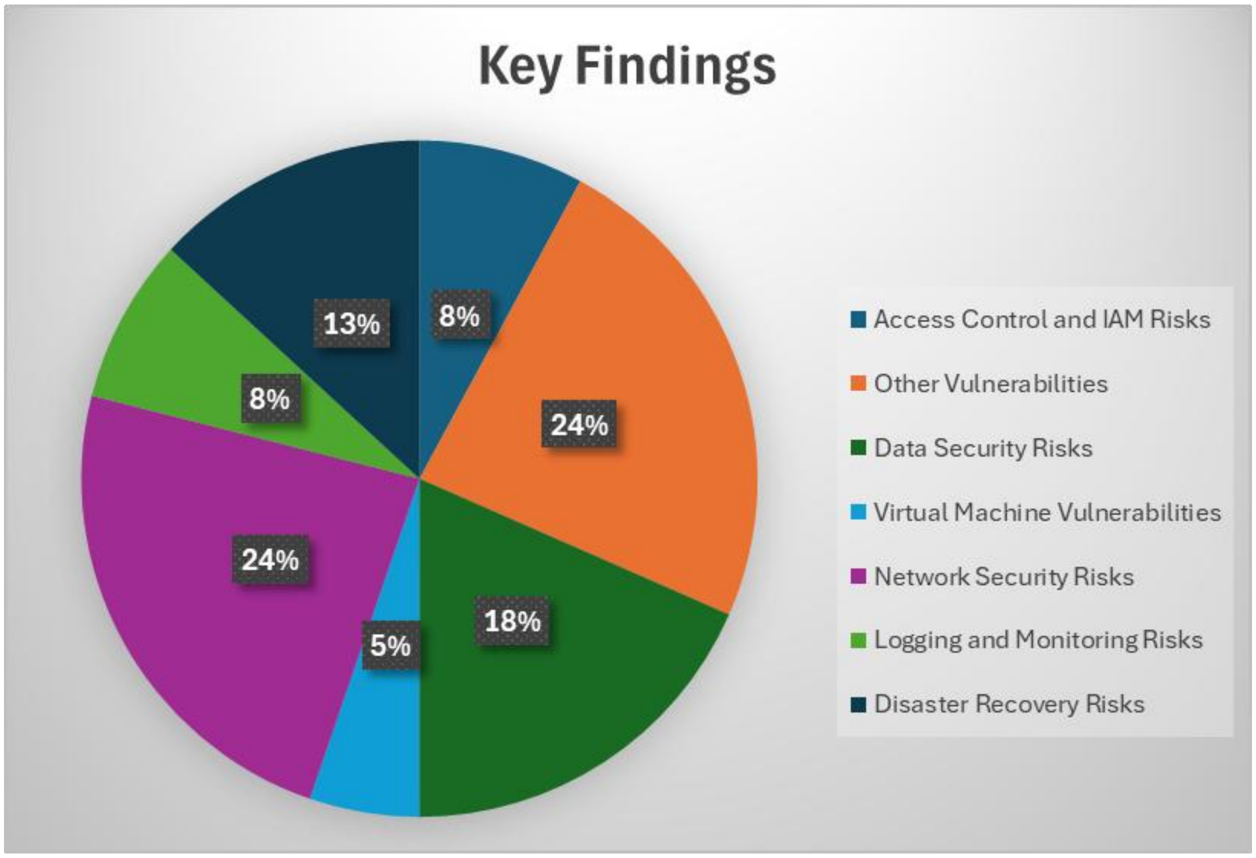
Risk Likelihood

	Not Severe	Kind of Severe	Mediocre	Severe	Very Severe
Extremely likely	Medium	Medium	High	Critical	Critical
Likely	Medium	Medium	Medium	High	Critical
Possible	Low	Medium	Medium	Medium	High
Unlikely	Low	Low	Medium	Medium	Medium
Very Unlikely	Low	Low	Low	Medium	Medium

1.5 Key Findings:

An overall view of the common security risks found within the E-commerce application across multiple domains can be seen in Table 1.1 below. The domains that we evaluated included access control and IAM risks, data security risks, virtual machine vulnerabilities, network security risks, logging and monitoring risks, disaster recovery risks, and other vulnerabilities. The most common risks found were data security and disaster recovery risks. These risks included unencrypted databases and storage volumes, lack of backups for databases and storage services, nonexistent disaster recovery plans, and more. The most common vulnerabilities found included application and database vulnerabilities. These are vulnerabilities that could be exploited by attacks like SQL injection (SQLi), Cross-Site Scripting (XSS), brute force, and more. Most of the vulnerabilities and risks found are severe in nature and should be handled immediately to avoid compromising the confidentiality, integrity, and availability of your organization.

Table 1.2:



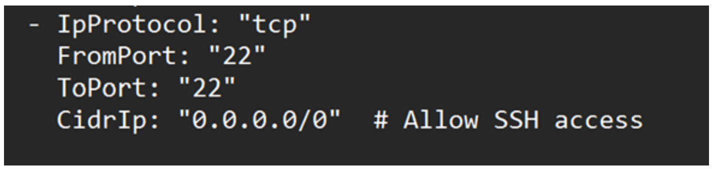
1.6 Recommendations:

Based on our findings, we have identified several key recommendations to improve the E-commerce application overall security posture: One key recommendation is implementing better data backup and retention strategies. All of the data being stored for the E-commerce application is currently unencrypted, which could allow unauthorized individuals to view sensitive data. Additionally, the data being stored is not being backed up either, which presents serious risk if a disaster were to occur because it would make data recovery lengthy and challenging and would lead to extended downtime. Another key recommendation is implementing data loss prevention strategies. Currently, there are many risks related to data loss, so it is important to implement DLP solutions that will help you monitor and detect unauthorized data transfer or access to ensure the confidentiality of your organization's data. AWS offers DLP solutions like Amazon Macie that you could implement to help discover and protect your organization's critical data from being exfiltrated. Another key recommendation is implementing logging and monitoring services or tools. We have identified zero logging and monitoring services or tools in place making it harder for the organization to monitor and protect the E-commerce application. Security incidents are currently going unnoticed, and you are not able to monitor the state of the application in real time, which presents serious risk. The last key recommendation is implementing a disaster recovery plan. Currently the organization does not have one, so it is crucial that you implement one because implementing a disaster recovery plan will help you prepare for potential disasters that may occur in the future. The purpose of the plan is to aid in getting the organization's operations up and running as quickly as possible after a disaster occurred. These plans typically outline procedures the organization can take in order to restore data if any was lost, restore systems, and more.

2.0 Access Control and IAM Assessment:

2.1 Finding 1:

- **Description:** The WebServerSecurityGroup for the EC2 Instance, allows SSH access from any IP address “0.0.0.0/0”.
- **Risk:** Medium Risk of Unauthorized Access due to Access Control Misconfigurations
- **Impact:** Allowing any IP to use SSH would lead to individuals gaining unauthorized access and would ultimately make it difficult to track and manage who is accessing the EC2 instance via SSH. SSH access should be restricted to trusted and authenticated personnel only. SSH would be used by personnel to connect to and from the EC2 instance and to perform administrative tasks.
- **Remediation:** The easiest way to remediate this risk is by restricting SSH access to trusted IP addresses or ranges. For example, your organization may have an IP address range for devices on the network. Once this is in place you can ensure that only valid and trusted devices are using SSH to connect to the EC2 instance and not all public IP addresses.
- **Screenshot of Finding:**

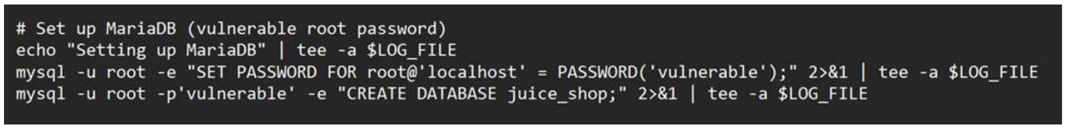


```
- IpProtocol: "tcp"
  FromPort: "22"
  ToPort: "22"
  CidrIp: "0.0.0.0/0"  # Allow SSH access
```

2.2 Finding 2:

- **Description:** The root account password used to set up the MySQL database within the CloudFormation Template is weak and is susceptible to brute force attacks. If compromised an attacker could have root access to all database data for the E-commerce platform.

- ➢ **Risk:** Critical Risk due to Weak Credentials leading to Broken Access Control
- ➢ **Impact:** If an attacker was able to brute force the weak root account database password, they could bypass the access controls put in place by gaining unauthorized root access to sensitive E-commerce data that they could potentially modify, export, delete, and more.
- ➢ **Remediation:** The easiest way to remediate this risk is by changing the root database password to something more secure. Your organization may have a password policy in place, but if it does not, the industry standard is twelve characters, with at least one uppercase letter, one lowercase letter, numbers, symbols, and does not contain dictionary words. If you are having trouble coming up with a password, there are password credential applications that can suggest safe passwords as well as save them, so you do not forget your passwords.
- ➢ **Screenshot of Finding:**



```
# Set up MariaDB (vulnerable root password)
echo "Setting up MariaDB" | tee -a $LOG_FILE
mysql -u root -e "SET PASSWORD FOR root@'localhost' = PASSWORD('vulnerable');" 2>&1 | tee -a $LOG_FILE
mysql -u root -p'vulnerable' -e "CREATE DATABASE juice_shop;" 2>&1 | tee -a $LOG_FILE
```

2.3 Finding 3:

- ➢ **Description:** Nonexistent or lack of IAM roles or policies for managing the EC2 Instance found within the CloudFormation template for the E-commerce platform.
- ➢ **Risk:** High Risk due to a Lack of IAM Roles or Policies Potentially Leading to Unauthorized Access and Potential Data Breaches.
- ➢ **Impact:** Not being able to properly manage the identities and access to the EC2 instance could result in unauthorized users gaining access to the instance and its data. It would also be harder to track and audit who is accessing the system, making it more difficult to detect and respond to malicious activities.
- ➢ **Remediation:** The easiest way to remediate this risk is to add and create applicable IAM roles and policies within the CloudFormation template. This would allow you to ensure proper access and manage identities for your EC2 instance. Applicable roles include roles for an administrator, developers, low privileged users, etc. Typically an administrator would have full access to all resources and would be able to perform all actions on these resources, low privileged users would typically have limited privileges like only having read access to particular resources, and developers would typically have access to development resources and would be able to perform far more actions than a lower privileged user, but less actions than an administrator. By applying these roles within the CloudFormation template, you can ensure the principle of least

privilege is being applied and that each user has the minimum permissions to do their job and nothing more. Furthermore, you can prevent or reduce the likelihood of an attacker gaining unauthorized access to your EC2 instance and its data.

3.0 Vulnerability Assessment Report:

3.1 Hardcoded Database Credentials:

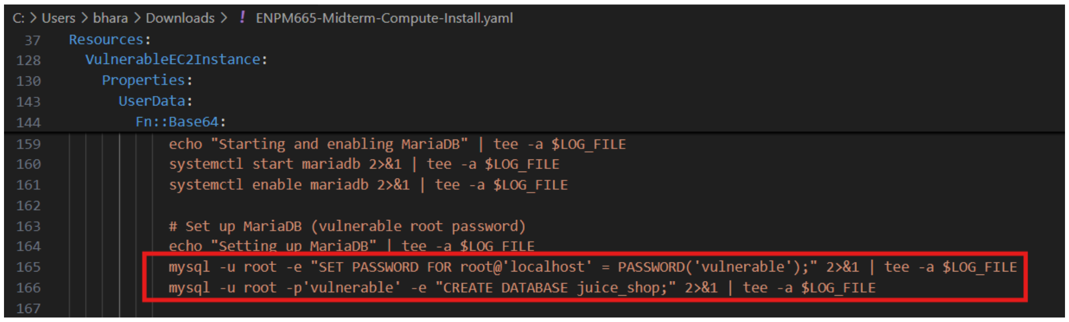
➢ **Description:** Credentials for the MariaDB database are present in plaintext in the UserData script of the CloudFormation template.

➢ **Risk:** Exposure of credentials can lead to unauthorized access and data breaches. Once the credentials are leaked, attackers can manipulate or exfiltrate sensitive data. Hence the risk is critical.

➢ **Impact:** Hardcoding sensitive information like credentials in the EC2 UserData script can lead to unintentional exposure of critical application secrets. If an attacker gains access to the EC2 instance or metadata, they can easily retrieve sensitive information such as database credentials. This leads to unauthorized access to the database, attackers can modify or delete data, compromise the application or move laterally. Lack of secrets management makes it harder to update or rotate credentials securely.

➢ **Remediation:** AWS Secrets manager should be used to securely store and manage sensitive data like database credentials. Using this service ensures data is encrypted and not stored in plain sight and allows only authorized users to access them. By using AWS Secrets manager, credentials can be retrieved dynamically at runtime, without exposing it in logs or instance configurations.

➢ **Evidence:**



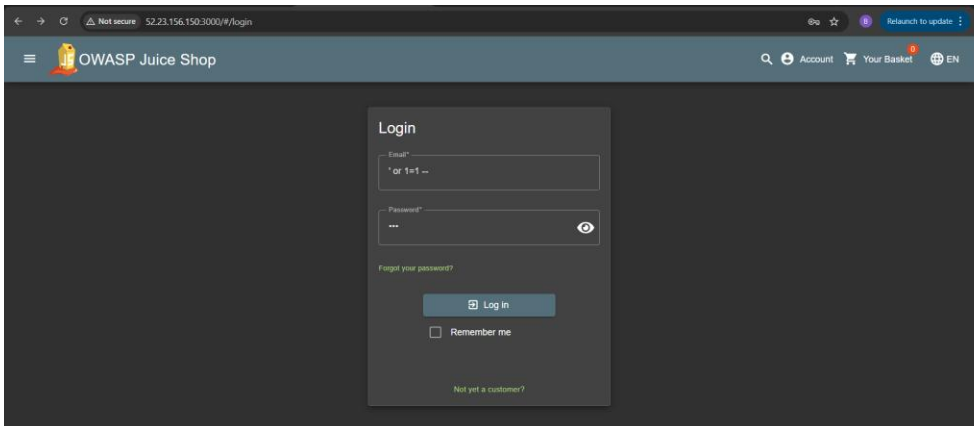
C: > Users > bhara > Downloads > ! ENPM665-Midterm-Compute-Install.yaml

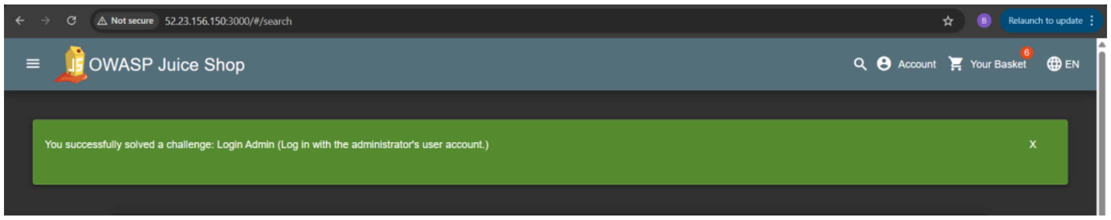
```
Resources:
  VulnerableEC2Instance:
    Properties:
      UserData:
        Fn::Base64:
            echo "Starting and enabling MariaDB" | tee -a $LOG_FILE
            systemctl start mariadb 2>&1 | tee -a $LOG_FILE
            systemctl enable mariadb 2>&1 | tee -a $LOG_FILE

            # Set up MariaDB (vulnerable root password)
            echo "Setting up MariaDB" | tee -a $LOG_FILE
            mysql -u root -e "SET PASSWORD FOR root@'localhost' = PASSWORD('vulnerable');" 2>&1 | tee -a $LOG_FILE
            mysql -u root -p'vulnerable' -e "CREATE DATABASE juice_shop;" 2>&1 | tee -a $LOG_FILE
```

3.2 SQL Injection Vulnerability:

➢ **Description:** The E-Commerce website is vulnerable to SQL injection attacks.

- ➢ **<u>Risk:</u>** This risk is critical as SQL injection allows attackers to manipulate database queries, bypass authentication and gain unauthorized access to sensitive data.
- ➢ **<u>Impact:</u>** SQL injection vulnerabilities are one of the dangerous vulnerabilities as it allows malicious users to manipulate database queries. User input sanitization is not performed hence by using a simple payload “ ' or 1=1 – ”, we were able to bypass the authentication mechanism and login as administrator on the E-commerce website. Attackers can gain unauthorized access to sensitive data such as credit card details stored in the database, they can execute arbitrary commands, exfiltrate, modify or even delete the data. The system can also be compromised by the attackers.
- ➢ **<u>Remediation:</u>** Secure coding practices such as using prepared statements and parameterized queries should be enforced so that user input is not treated as executable code. Input validation and sanitization should be enforced so that only valid data is passed. Regular security audits and web application firewalls should be used to enhance security. By adopting these measures, we can significantly reduce the risk of SQL injections.
- ➢ **<u>Evidence:</u>**





You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)

3.3 XSS Vulnerability:

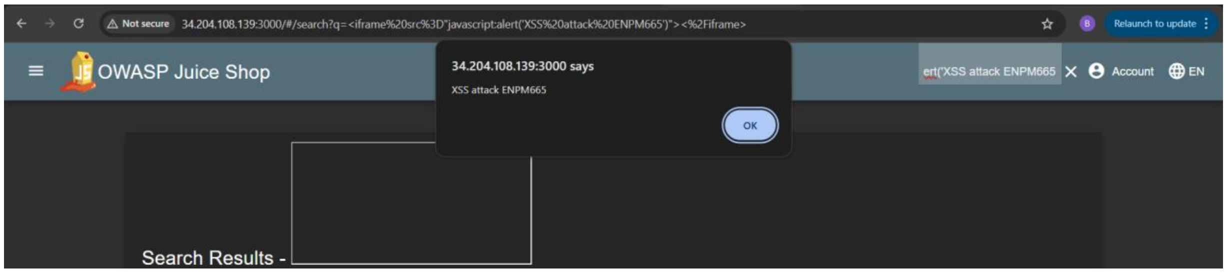
- ➢ **<u>Description:</u>** The E-Commerce website is vulnerable to Cross-Site Scripting attack.

➢ **Risk:** Cross-site scripting allows attackers to inject malicious JavaScript into webpages viewed by users that leads to session hijacking, data theft or phishing attacks. The risk is high.

➢ **Impact:** DOM based XSS vulnerability occurs when an attacker can inject malicious JavaScript code into the client-side DOM of a website. The payload "<iframe src="javascript:alert('XSS')"></iframe>" successfully executed when placed in the search bar of the E-Commerce website. This indicates that the website fails to validate and sanitize user inputs before incorporating them into DOM. This vulnerability can be exploited by stealing session cookies, authentication tokens or sensitive data from the user. Attackers can also inject fake forms or phishing pages to trick users into providing their credentials and credit card details.

➢ **Remediation:** User input validation and sanitization are really important. This ensures that malicious code injections do not take place. Content Security Policy should be adopted to limit the execution of untrusted scripts which reduces the potential impact of any injected malicious script. Inline JavaScript and event handlers should be avoided. Events should be handled in a secure manner. Regular security testing should be done to identify and remediate vulnerabilities.

➢ **Evidence:**



34.204.108.139:3000/#/search?q=<iframe%20src%3D"javascript:alert('XSS%20attack%20ENPM665')"><%2Fiframe>

34.204.108.139:3000 says

XSS attack ENPM665

ert('XSS attack ENPM665

3.4 Unencrypted Database:

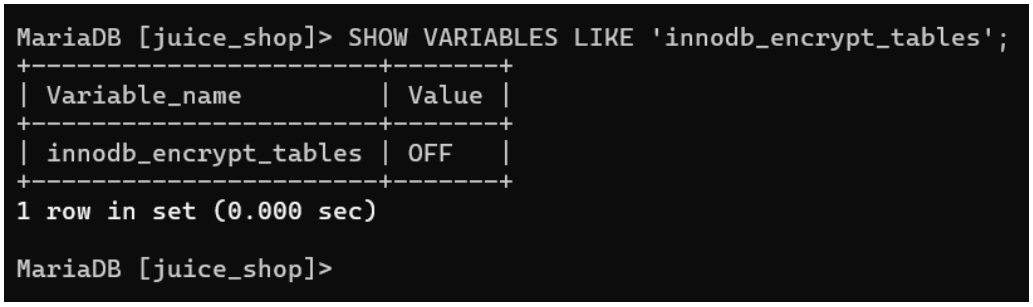
➢ **Description:** The MariaDB Database and the EBS Volume are not encrypted.

➢ **Risk:** Storing data in an unencrypted format exposes sensitive information to unauthorized access. In the event of unauthorized access to the database, attackers can read data in plaintext, hence the risk is critical.

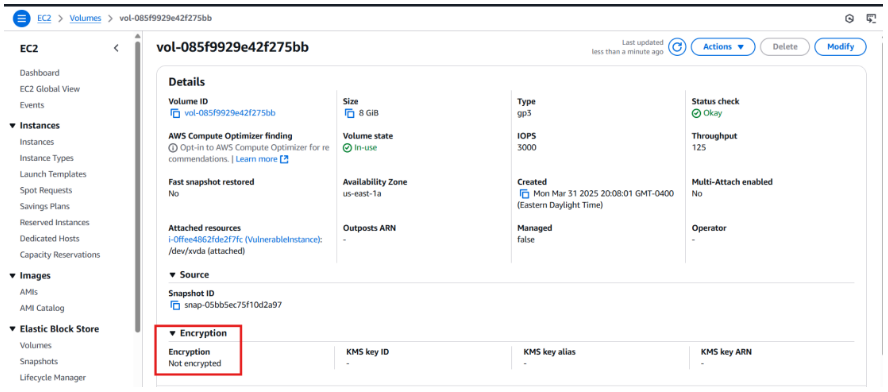
➢ **Impact:** Usage of unencrypted MariaDB database and EBS volume in a cloud environment significantly increase the risk of unauthorized access to sensitive data. When data at rest is not encrypted, data is not protected if an attacker gains access to the database. Attackers will be able to read sensitive data like credit card details in plain text and can use it for malicious purposes. In the event of physical theft of the EBS volume, lack of encryption poses more risk. Unencrypted data also leads to compliance violations.

➢ **Remediation:** Data should always be encrypted, whether at rest or in transit. MariaDB database should be configured to use encryption at rest. Data transferred during communication should also be encrypted by enabling SSL/TLS for database connections. EBS Volumes can be encrypted at the time of creation itself. For existing volumes, encrypted copies can be created and migrate data to the encrypted volume. Automation is available to encrypt new instances as and when they are created.

➢ **Evidence:**



```
MariaDB [juice_shop]> SHOW VARIABLES LIKE 'innodb_encrypt_tables';
+-----------------------+-------+
| Variable_name         | Value |
+-----------------------+-------+
| innodb_encrypt_tables | OFF   |
+-----------------------+-------+
1 row in set (0.000 sec)

MariaDB [juice_shop]>
```



vol-085f9929e42f275bb

Details

Volume ID vol-085f9929e42f275bb	Size 8 GiB	Type gp3	Status check Okay
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more	Volume state In-use	IOPS 3000	Throughput 125
Fast snapshot restored No	Availability Zone us-east-1a	Created Mon Mar 31 2025 20:08:01 GMT-0400 (Eastern Daylight Time)	Multi-Attach enabled No
Attached resources i-0ffee4862fde2f7fc (VulnerableInstance): /dev/xvda (attached)	Outposts ARN -	Managed false	Operator -

Source

Snapshot ID
snap-05bb5ec75f10d2a97

Encryption

Encryption Not encrypted	KMS key ID -	KMS key alias -	KMS key ARN -

3.5 Directory Traversal Vulnerability:

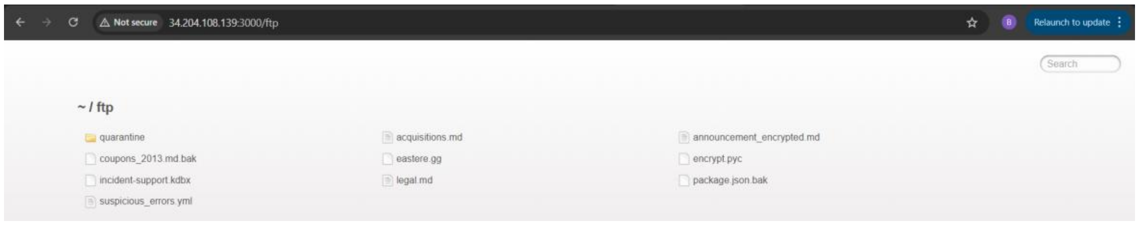
➢ **Description:** Directory traversal vulnerability allows attackers to access unauthorized files and directories on a server by manipulating file paths.

➢ **Risk:** Improper access controls allow attackers to navigate through server directories and access sensitive files. This risk is high.

➢ **Impact:** Navigating to 'http://<public IP address>:3000/ftp' exposed folders and files which indicates improper access controls. Attackers can access sensitive configuration files or credentials stored on the server which leads to data leaks and further exploitation. With this information, attackers can plan their next attack steps.

➢ **Remediation:** Proper file system permissions should be used to prevent unauthorized access to sensitive data. Only necessary files should be exposed to the public. User input should be sanitized and validated to prevent directory traversal attempts. Web application firewall should be deployed to detect and block directory traversal attempts. If a certain directory is not required for the functioning of an application, it should be removed, or access should be restricted.

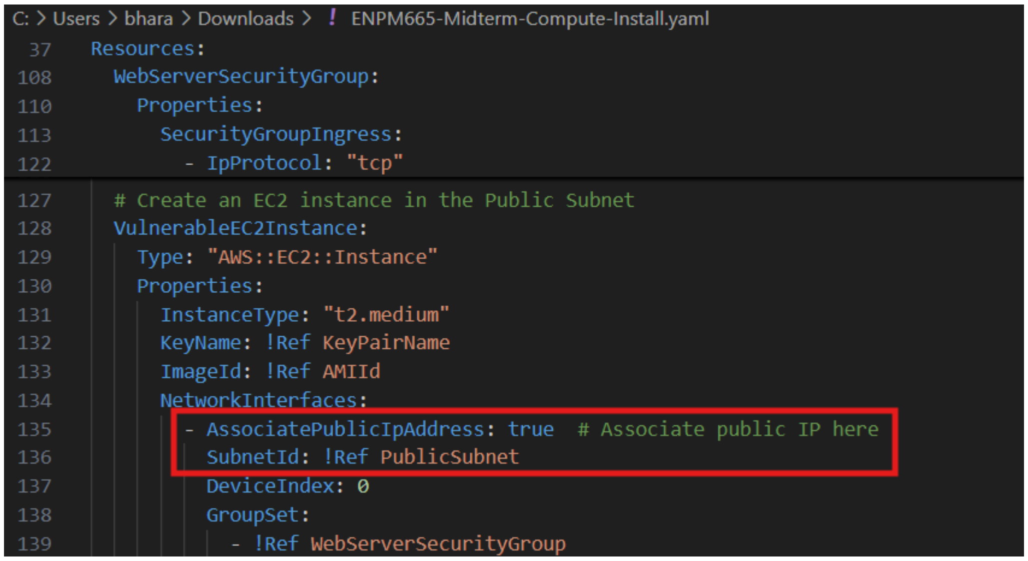
- **Evidence:**



```
Not secure  34.204.108.139:3000/ftp

~ / ftp

quarantine              acquisitions.md     announcement_encrypted.md
coupons_2013.md.bak     eastere.gg          encrypt.pyc
incident-support.kdbx   legal.md            package.json.bak
suspicious_errors.yml
```

3.6 Instance in public subnet:

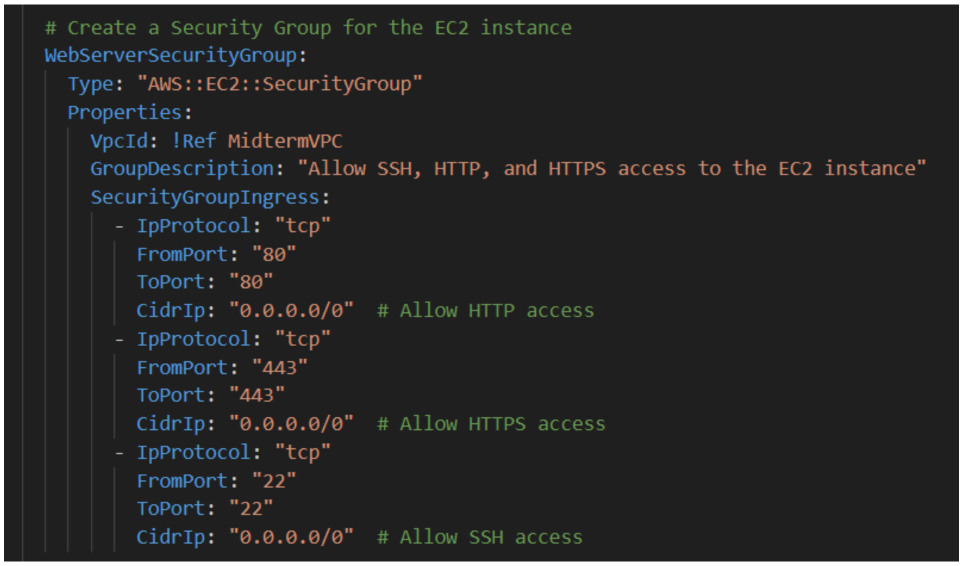
- **Description:** The EC2 instance is deployed in the public subnet, providing access to anyone on the internet.
- **Risk:** Deploying EC2 instance in the public subnet exposes it to direct internet access. Attackers can pivot into private resources and escalate their attack. Hence the risk is high.
- **Impact:** Services deployed in the public subnet are vulnerable to a wide range of cyber-attacks such as brute force attacks, Denial of service attacks, automated scans by attackers. Services can be easily compromised. Absence of proper access controls makes the service susceptible to unauthorized access. Public subnet instances can become the entry point for attacks on the internal network. Attackers can first compromise the services facing the public and then gain access to the private subnet resources and exfiltrate sensitive data or compromise critical resources.
- **Remediation:** The EC2 instance should be deployed in a private subnet, the service can be connected to the internet via NAT Gateway allowing private instances to access the internet without exposing them directly to the internet. A bastion host can be used to allow access to instances instead of SSH or RDP access. VPN can be used to provide secure access to the private network. Logging and monitoring should be enabled using CloudTrail and CloudWatch to detect and respond to suspicious activities.
- **Evidence:**



```
C: > Users > bhara > Downloads > ! ENPM665-Midterm-Compute-Install.yaml
  Resources:
    WebServerSecurityGroup:
      Properties:
        SecurityGroupIngress:
          - IpProtocol: "tcp"
    # Create an EC2 instance in the Public Subnet
    VulnerableEC2Instance:
      Type: "AWS::EC2::Instance"
      Properties:
        InstanceType: "t2.medium"
        KeyName: !Ref KeyPairName
        ImageId: !Ref AMIId
        NetworkInterfaces:
          - AssociatePublicIpAddress: true  # Associate public IP here
            SubnetId: !Ref PublicSubnet
            DeviceIndex: 0
            GroupSet:
              - !Ref WebServerSecurityGroup
```

3.7 Inadequate Security Group Restrictions:

- **Description:** Unrestricted inbound access to the EC2 instance is provided.
- **Risk:** Unrestricted access, especially on SSH, HTTP and HTTPS increase the likelihood of brute force attacks and web-based attacks. Attackers can compromise the instance and takeover the cloud resources. The risk is critical.
- **Impact:** Allowing SSH access from anyone present on the internet provides attackers with the opportunity to brute force or exploit weak credentials to gain unauthorized access to the instance. As HTTP and HTTPS ports are open, EC2 instance is vulnerable to variable web-based attacks. These attacks can lead to data breaches or system compromise. Open access to critical resources from the public internet results in non-compliance with security standards or regulations.
- **Remediation:** Access to SSH port should be limited, only trusted IP ranges should be allowed access. HTTP and HTTPS access should be limited only to necessary external traffic. Web application firewall should be used to protect the instance from common vulnerabilities. Enabling logging through CloudTrail and CloudWatch helps to detect and respond to suspicious activities quickly.
- **Evidence:**



```yaml
# Create a Security Group for the EC2 instance
WebServerSecurityGroup:
  Type: "AWS::EC2::SecurityGroup"
  Properties:
    VpcId: !Ref MidtermVPC
    GroupDescription: "Allow SSH, HTTP, and HTTPS access to the EC2 instance"
    SecurityGroupIngress:
      - IpProtocol: "tcp"
        FromPort: "80"
        ToPort: "80"
        CidrIp: "0.0.0.0/0"  # Allow HTTP access
      - IpProtocol: "tcp"
        FromPort: "443"
        ToPort: "443"
        CidrIp: "0.0.0.0/0"  # Allow HTTPS access
      - IpProtocol: "tcp"
        FromPort: "22"
        ToPort: "22"
        CidrIp: "0.0.0.0/0"  # Allow SSH access
```

3.8 Lack of IAM Roles and Policies

➢ **Description:** Absence of properly defined IAM roles and policies leads to excessive or mismatched permissions which in turn leads to an increase in risk of unauthorized access.

➢ **Risk:** The risk is critical, as users and services may get excessive privileges which violates the principle of least privilege.

➢ **Impact:** Absence of IAM roles and policies leads to users getting excessive permissions to critical resources. This violates the principle of least privilege. Attackers can just compromise any one user and then can perform actions which are generally reserved for root users. They can exfiltrate data or disrupt services. Without IAM roles and policies it becomes difficult to track which user has access to what resources. This can prevent security incidents, hinder incident response and it will be challenging to conduct audits for compliance.

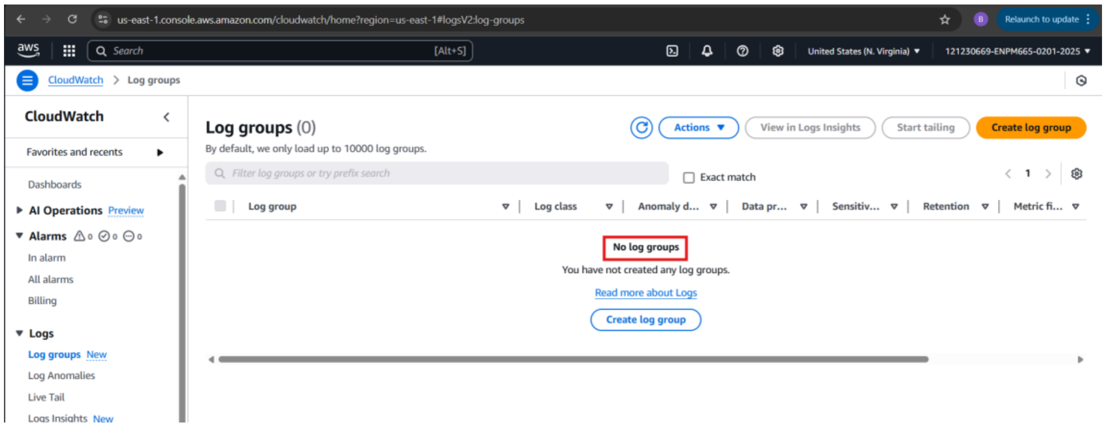
➢ **Remediation:** Granular IAM policies that enforce the principle of least privilege should be defined. Each user, group and service should have only the necessary permissions to perform their tasks. IAM roles should be used to grant EC2 instances or other services access to other AWS resources. Multi-factor authentication should be enforced for access to all sensitive resources. MFA adds an additional layer of security.

➢ **Evidence:** There is no evidence of IAM rules and policies being present.

3.9 Absence of Logging and Monitoring:

➢ **Description:** Logging and monitoring are not enabled.

- ➢ **Risk:** Without logging and monitoring, security incidents such as unauthorized access cannot be traced. This increases the risk of prolonged attacks and makes it difficult to detect and respond quickly. Hence the risk is high.
- ➢ **Impact:** Absence of proper logging and monitoring will lead to unauthorized access attempts, privilege escalation or any kind of security incident going unnoticed. This increases the window of opportunity for the attackers to exploit vulnerabilities. An effective response cannot be performed if proper logging of the incident is not done. Logs are essential to track configurations and operational status of resources. Without logging and monitoring, misconfigurations and mistakes in infrastructure will go unnoticed and attackers can take advantage of this to exploit vulnerabilities. Suspicious activities performed by internal threat actors also go untraced if logging is not enabled.
- ➢ **Remediation:** CloudTrail should be used as it provides comprehensive logs of API activity. CloudTrail should be enabled for all regions and logs should be stored in a secure S3 bucket with versioning. Enable CloudWatch logs for EC2 instances to capture application and system logs. CloudWatch alarms can be used to monitor key metrics and set up thresholds. AWS GuardDuty should be implemented as it continuously monitors for malicious activity in the AWS environment. Integrating CloudWatch with GuardDuty enables us to receive actionable alerts and take prompt corrective actions.
- ➢ **Evidence:**


us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups

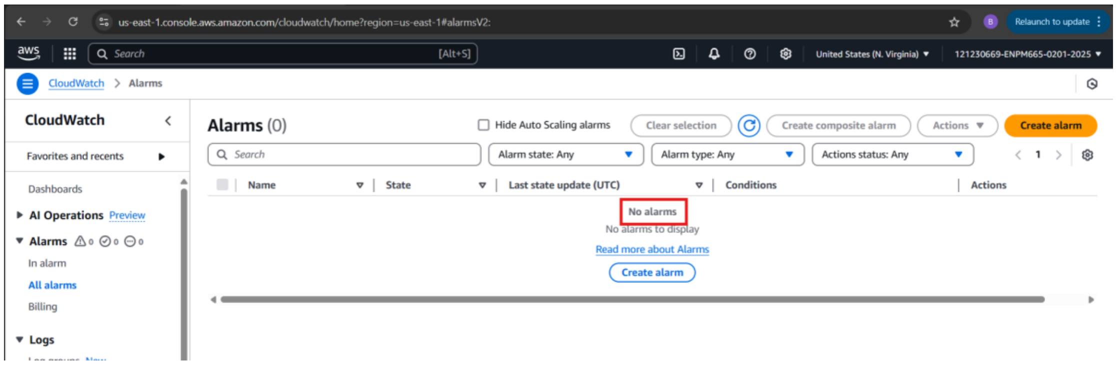
United States (N. Virginia) 121230669-ENPM665-0201-2025

Log groups (0)

By default, we only load up to 10000 log groups.

No log groups

You have not created any log groups.


us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:

United States (N. Virginia) 121230669-ENPM665-0201-2025

Alarms (0)

No alarms

No alarms to display

4.0 Data Security Assessment:

4.1 Unencrypted Database:

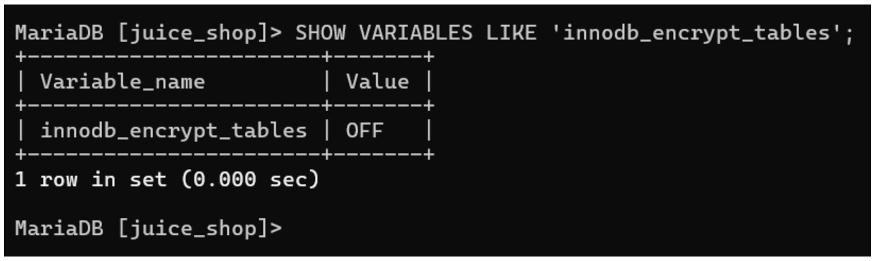
➢ **<u>Description:</u>** MariaDB database is being used without encryption.

➢ **<u>Risk:</u>** Database exposed to unauthorized access due to lack of encryption. The risk is critical because if breached, attackers can exfiltrate customer data such as credit card details.

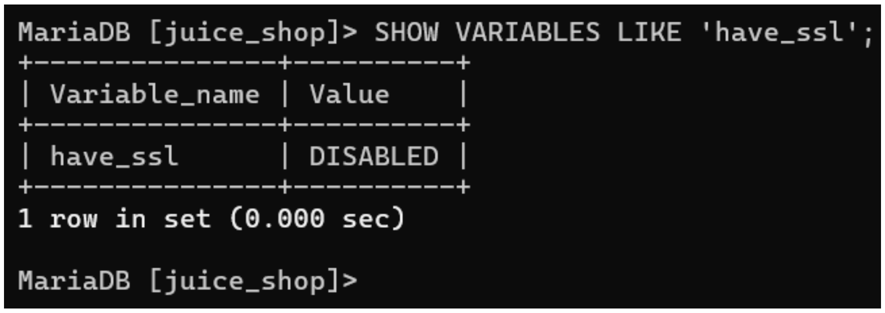
➢ **<u>Impact:</u>** In the cloud, unencrypted database poses significant risks to both data security and compliance. Without encryption any sensitive information stored in the database is vulnerable to unauthorized access, whether it is an internal breach, cyberattack or accidental exposure. Regulations like PCI-DSS and GDPR mandate specific measures to protect data, especially sensitive data like credit card details. The web application has the option to store credit card details, if not stored securely, attackers can use them for malicious purposes. Failure to comply with the regulations can lead to hefty fines, reputational damage and legal consequences.

➢ **<u>Remediation:</u>** MariaDB configuration should be modified to turn on encryption for both tables and logs, this ensures that all data stored in the database is encrypted automatically. AWS Key Management Service can be used to store and manage encryption keys securely. SSL/TLS encryption can be enabled for the data in transit as well by configuring the database to use SSL certificates.

➢ **<u>Evidence:</u>**



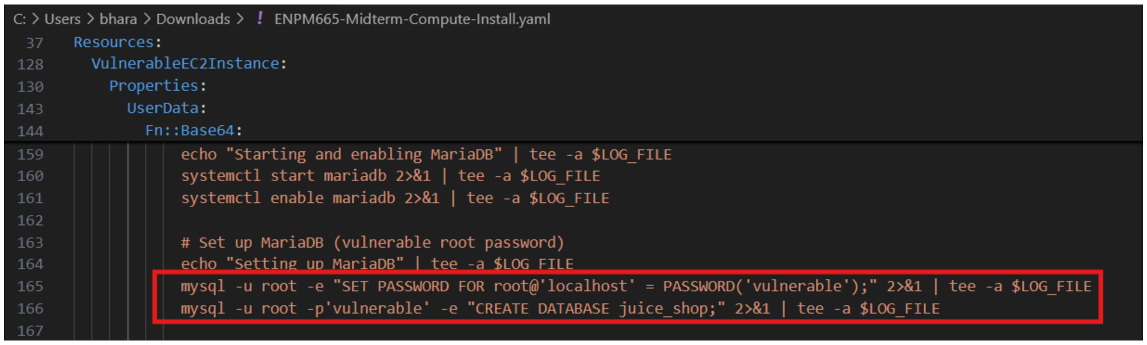
```
MariaDB [juice_shop]> SHOW VARIABLES LIKE 'innodb_encrypt_tables';
+-----------------------+-------+
| Variable_name         | Value |
+-----------------------+-------+
| innodb_encrypt_tables | OFF   |
+-----------------------+-------+
1 row in set (0.000 sec)

MariaDB [juice_shop]>
```



```
MariaDB [juice_shop]> SHOW VARIABLES LIKE 'have_ssl';
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_ssl      | DISABLED |
+---------------+----------+
1 row in set (0.000 sec)

MariaDB [juice_shop]>
```

4.2 Hardcoded Credentials:

- **Description:** Database credentials have been hardcoded in the UserData script present in the template.
- **Risk:** Hardcoded credentials increase the risk of unauthorized access. The risk is critical as hardcoded credentials result in data theft, privilege escalation and long-term security risks.
- **Impact:** Hardcoded credentials such as the username and password for the MariaDB database pose a severe security risk. Anyone with access to the instance can retrieve the credentials by inspecting CloudFormation Stack, EC2 UserData logs and bash history or any environment variable storing the credentials. This makes it extremely easy for an attacker to gain access to the database as root, modify or extract sensitive information.
- **Remediation:** Instead of hard coding credentials, AWS Secrets Manager can be used to store credentials and retrieve them at runtime. An IAM role with necessary permissions should be assigned to the EC2 instance or application accessing the secret. Through this, applications can dynamically retrieve credentials instead of storing them in plaintext.
- **Evidence:**



C: > Users > bhara > Downloads > ! ENPM665-Midterm-Compute-Install.yaml

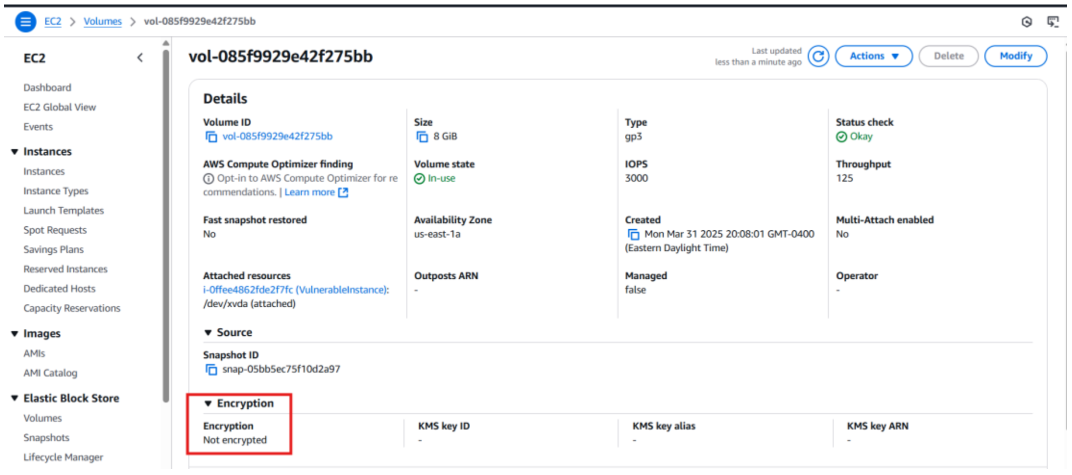
```
Resources:
  VulnerableEC2Instance:
    Properties:
      UserData:
        Fn::Base64:
            echo "Starting and enabling MariaDB" | tee -a $LOG_FILE
            systemctl start mariadb 2>&1 | tee -a $LOG_FILE
            systemctl enable mariadb 2>&1 | tee -a $LOG_FILE

            # Set up MariaDB (vulnerable root password)
            echo "Setting up MariaDB" | tee -a $LOG_FILE
            mysql -u root -e "SET PASSWORD FOR root@'localhost' = PASSWORD('vulnerable');" 2>&1 | tee -a $LOG_FILE
            mysql -u root -p'vulnerable' -e "CREATE DATABASE juice_shop;" 2>&1 | tee -a $LOG_FILE

```

4.3 Unencrypted EBS Volume:

- **Description:** Unencrypted EBS volumes expose data to unauthorized access, increasing the risk of breaches and compliance violations.
- **Risk:** The risk is critical as storing data on unencrypted volume leaves it vulnerable to unauthorized access, data breaches and compliance violations.
- **Impact:** Unencrypted EBS volumes pose severe security risk as data is stored in plaintext. If a security breach occurs, an attacker can extract sensitive data and use it for malicious purposes. Although unlikely, if an attacker gains unauthorized access to AWS hardware, they can easily extract the data. If snapshots of the unencrypted volumes are taken, the risk extends further as these snapshots remain unprotected. Without encryption, there is no guarantee that the data remains

secure if AWS hardware has decommissioned or repurposed. Unencrypted data also leads to compliance failure.

- ➢ **Remediation:** To mitigate this risk, the feature of built-in encryption provided by AWS should be enabled to protect data at rest. For existing unencrypted volume, snapshot of the volume can be created on which the encryption can be applied, and this can be restored to a new encrypted volume. When creating EBS volumes, users should enable encryption using AWS Key Management Service to ensure that all data stored gets encrypted automatically. Access controls should be implemented using IAM roles and policies to prevent unauthorized and unnecessary access to EBS volumes.
- ➢ **Evidence:**



vol-085f9929e42f275bb

Details

Volume ID vol-085f9929e42f275bb	Size 8 GiB	Type gp3	Status check Okay
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more	Volume state In-use	IOPS 3000	Throughput 125
Fast snapshot restored No	Availability Zone us-east-1a	Created Mon Mar 31 2025 20:08:01 GMT-0400 (Eastern Daylight Time)	Multi-Attach enabled No
Attached resources i-0ffee4862fde2f7fc (VulnerableInstance): /dev/xvda (attached)	Outposts ARN -	Managed false	Operator -

Source

Snapshot ID
snap-05bb5ec75f10d2a97

Encryption

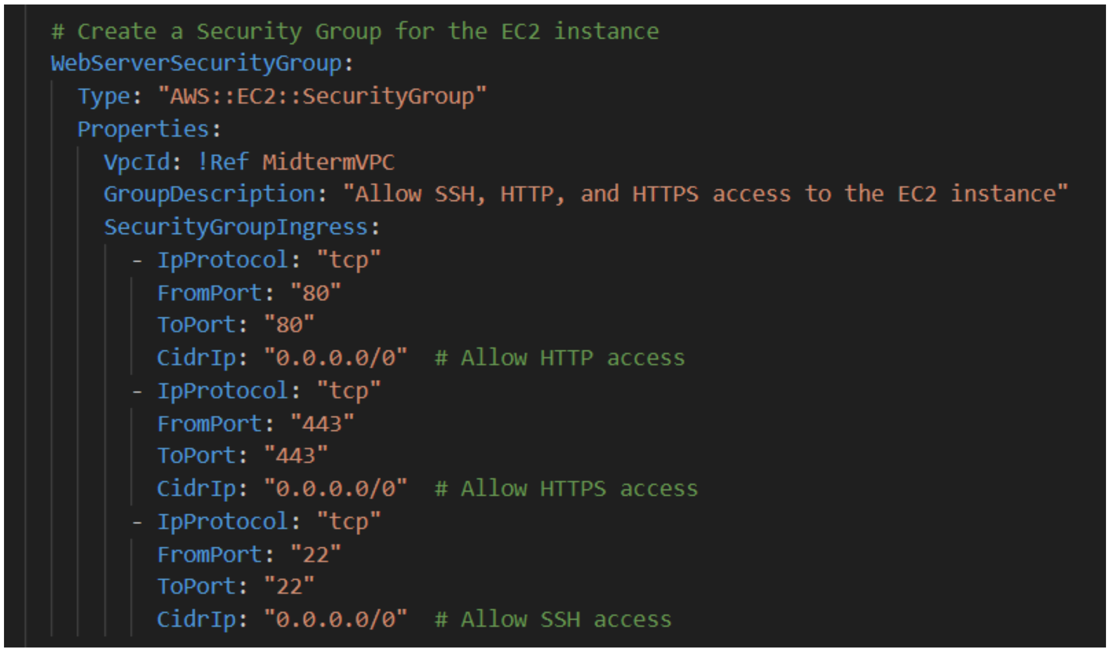
Encryption	KMS key ID	KMS key alias	KMS key ARN
Not encrypted	-	-	-

4.4 Overly Permissive Security Group:

- ➢ **Description:** The security group allows unrestricted inbound access on SSH, HTTP and HTTPS that increases the risk of unauthorized access.
- ➢ **Risk:** Allowing unrestricted inbound access, especially SSH exposes the EC2 instance to brute force attacks and unauthorized access attempts. Hence the risk is critical.
- ➢ **Impact:** Allowing unrestricted access to the EC2 instance exposes it to a range of cyber-attacks. Publicly accessible SSH increases the risk of brute force attacks. If compromised, attackers can exfiltrate sensitive data, manipulate system files or also be used as a foothold for other attacks. By exposing HTTP and HTTPS to the entire internet, a chance is given for the attacker to scan the application for vulnerabilities and even exploit them. Publicly accessible instances are more vulnerable to Denial-of-Service attacks.
- ➢ **Remediation:** SSH access should only be allowed from trusted IP addresses, or better yet, AWS Systems Manager Session Manager should be used instead of direct SSH to improve security. To protect web traffic, AWS Web Application

Firewall (WAF) can help block malicious requests. Security groups should follow the least privileged rule, meaning only necessary connections should be allowed. Enabling multi-factor authentication (MFA) for SSH access adds another layer of security. Finally, keeping an eye on access logs using AWS CloudTrail and VPC Flow Logs can help detect any suspicious activity.

➢ **Evidence:**



```
# Create a Security Group for the EC2 instance
WebServerSecurityGroup:
  Type: "AWS::EC2::SecurityGroup"
  Properties:
    VpcId: !Ref MidtermVPC
    GroupDescription: "Allow SSH, HTTP, and HTTPS access to the EC2 instance"
    SecurityGroupIngress:
      - IpProtocol: "tcp"
        FromPort: "80"
        ToPort: "80"
        CidrIp: "0.0.0.0/0"  # Allow HTTP access
      - IpProtocol: "tcp"
        FromPort: "443"
        ToPort: "443"
        CidrIp: "0.0.0.0/0"  # Allow HTTPS access
      - IpProtocol: "tcp"
        FromPort: "22"
        ToPort: "22"
        CidrIp: "0.0.0.0/0"  # Allow SSH access
```

4.5 Public IP Exposure:

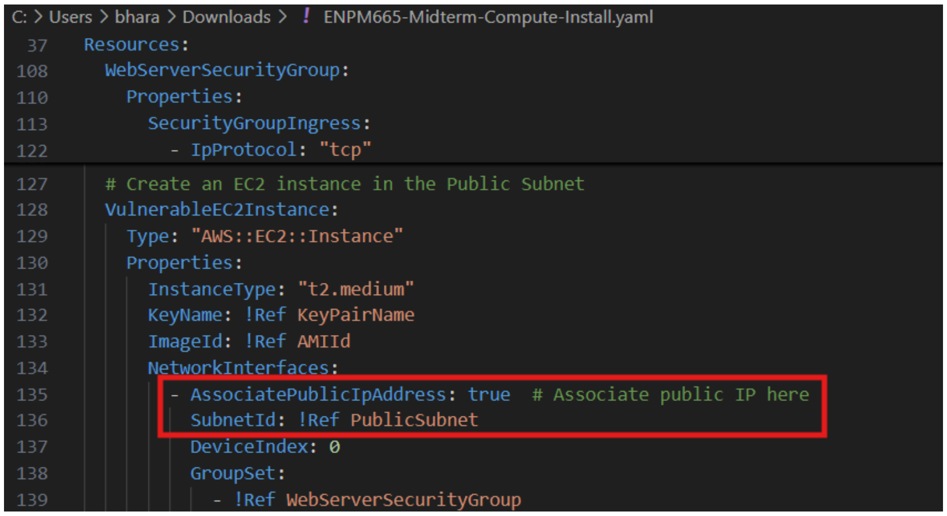
➢ **Description:** The EC2 instance is deployed in a public subnet with direct internet access, increasing exposure to potential attacks.

➢ **Risk:** Deploying a service in the public subnet directly exposes it to the internet and makes it an easy target for cyberattacks. The risk is high as public-facing instances increase the risk of unauthorized access attempts.

➢ **Impact:** EC2 instance deployed in a public subnet with direct internet access not only impacts on the security of data but can also affect the overall reliability and performance of the EC2 service. By giving the instance a public IP address, sensitive services become open to the internet which makes it vulnerable to brute force attacks, unauthorized access and other cyber-attacks. If the instance is compromised, attackers can disrupt service availability or cause system instability.

➢ **Remediation:** EC2 instance should be deployed in a private subnet which ensures that the instance is not directly exposed to the internet. NAT Gateway or VPN should be used to facilitate secure, indirect internet access. This ensures that the service can still access the required internet resources securely. To enhance security, a bastion host should be deployed in the public subnet for administrative access to instances in the private subnet. Through these sensitive resources remain isolated from the internet, with all access routed through the bastion host.

➢ <u>**Evidence:**</u>



C: > Users > bhara > Downloads > ! ENPM665-Midterm-Compute-Install.yaml

```
Resources:
  WebServerSecurityGroup:
    Properties:
      SecurityGroupIngress:
        - IpProtocol: "tcp"
```

```
  # Create an EC2 instance in the Public Subnet
  VulnerableEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      InstanceType: "t2.medium"
      KeyName: !Ref KeyPairName
      ImageId: !Ref AMIId
      NetworkInterfaces:
        - AssociatePublicIpAddress: true  # Associate public IP here
          SubnetId: !Ref PublicSubnet
          DeviceIndex: 0
          GroupSet:
            - !Ref WebServerSecurityGroup
```

4.6 Overly permissive Root Access and File Permissions:

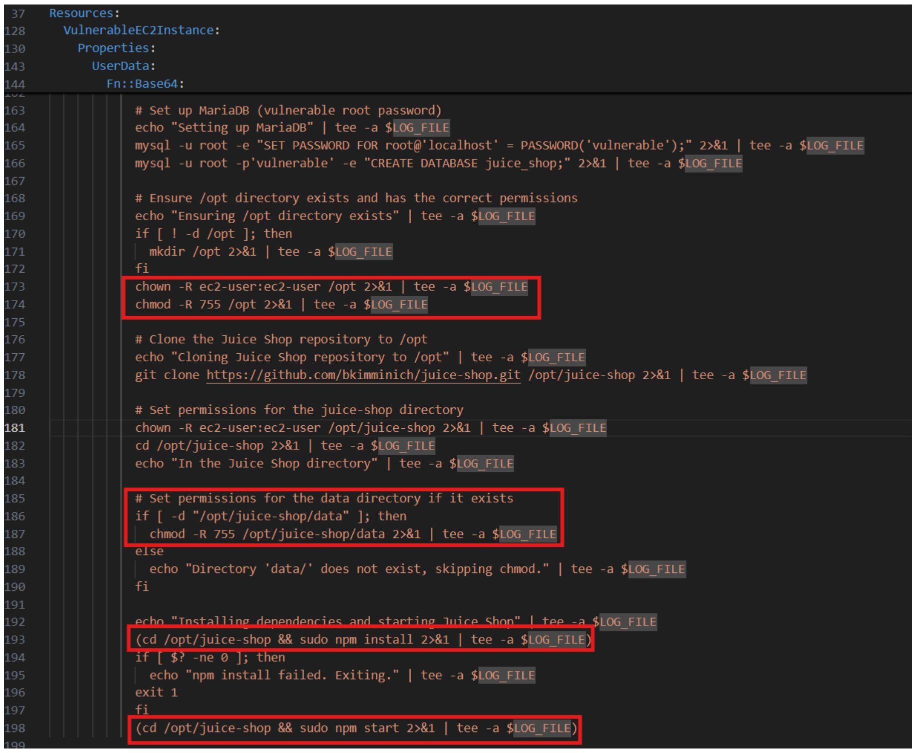
➢ <u>**Description:**</u> Excessive root access and weak file permissions increase the risk of unauthorized access and privilege escalation.

➢ <u>**Risk:**</u> This risk is critical as granting executive root access and weak file permissions increases the likelihood of privilege escalation.

➢ <u>**Impact:**</u> Allowing unrestricted root access in the UserData script and setting overly permissive file permissions introduces multiple vulnerabilities. Weak access controls lead to unauthorized access, data breaches and privilege escalation. If the service is compromised, attackers can install malicious software, exfiltrate sensitive data or modify system configurations with the help of root privileges. With overly permissive file permissions, any system user can read and execute sensitive files.

➢ <u>**Remediation:**</u> Principle of least privilege should be followed by restricting root access and using a dedicated non-root user to run services. Instead of hardcoding permissions in the Userdata script, permissions should be configured so that only essential users have access to critical resources. Access to sensitive files should be limited. System directories and configuration files should be accessible only to authorized users. AWS CloudTrail and Amazon CloudWatch can be used for logging and monitoring.

➢ <u>**Evidence:**</u>



```
Resources:
  VulnerableEC2Instance:
    Properties:
      UserData:
        Fn::Base64:
            # Set up MariaDB (vulnerable root password)
            echo "Setting up MariaDB" | tee -a $LOG_FILE
            mysql -u root -e "SET PASSWORD FOR root@'localhost' = PASSWORD('vulnerable');" 2>&1 | tee -a $LOG_FILE
            mysql -u root -p'vulnerable' -e "CREATE DATABASE juice_shop;" 2>&1 | tee -a $LOG_FILE

            # Ensure /opt directory exists and has the correct permissions
            echo "Ensuring /opt directory exists" | tee -a $LOG_FILE
            if [ ! -d /opt ]; then
              mkdir /opt 2>&1 | tee -a $LOG_FILE
            fi
            chown -R ec2-user:ec2-user /opt 2>&1 | tee -a $LOG_FILE
            chmod -R 755 /opt 2>&1 | tee -a $LOG_FILE

            # Clone the Juice Shop repository to /opt
            echo "Cloning Juice Shop repository to /opt" | tee -a $LOG_FILE
            git clone https://github.com/bkimminich/juice-shop.git /opt/juice-shop 2>&1 | tee -a $LOG_FILE

            # Set permissions for the juice-shop directory
            chown -R ec2-user:ec2-user /opt/juice-shop 2>&1 | tee -a $LOG_FILE
            cd /opt/juice-shop 2>&1 | tee -a $LOG_FILE
            echo "In the Juice Shop directory" | tee -a $LOG_FILE

            # Set permissions for the data directory if it exists
            if [ -d "/opt/juice-shop/data" ]; then
              chmod -R 755 /opt/juice-shop/data 2>&1 | tee -a $LOG_FILE
            else
              echo "Directory 'data/' does not exist, skipping chmod." | tee -a $LOG_FILE
            fi

            echo "Installing dependencies and starting Juice Shop" | tee -a $LOG_FILE
            (cd /opt/juice-shop && sudo npm install 2>&1 | tee -a $LOG_FILE)
            if [ $? -ne 0 ]; then
              echo "npm install failed. Exiting." | tee -a $LOG_FILE
            exit 1
            fi
            (cd /opt/juice-shop && sudo npm start 2>&1 | tee -a $LOG_FILE)
```

4.7 Lack of Backup Strategy:

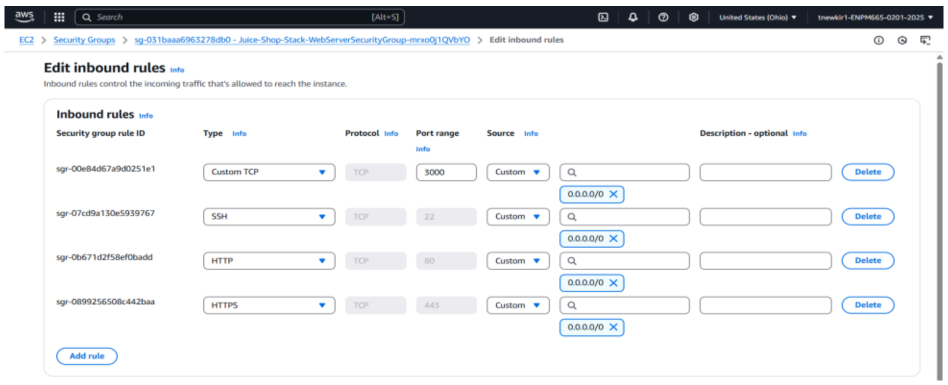
- **Description:** The EC2 instance lacks a backup mechanism that increases the risk of data loss due to security incidents or accidental deletion.
- **Risk:** Lack of a backup strategy exposes sensitive data to permanent loss and disrupts business continuity if a security incident occurs. Hence this risk is high.
- **Impact:** As there is no backup strategy, critical data stored on the EC2 instance is at risk of permanent loss due to cyberattacks, system failures or accidental deletion. The absence of backup increases the downtime, financial losses and disrupts operations if a security incident occurs. Without a proper backup and recovery plan, compliance requirements may not be met, which further exposes the company to regulatory risks and penalties.
- **Remediation:** AWS Backup should be used to automate backup scheduling, ensuring regular and consistent backups. Amazon EBS snapshots should be enabled to take periodic backups of the instance's storage. Backups can be stored in S3 buckets with versioning and lifecycle policies to prevent accidental deletions. Backups should be encrypted using AWS Key Management Service. A disaster recovery plan should be implemented, it should include backup testing to verify the integrity of data and quick restoration when needed.
- **Evidence:** Backup configuration is not present in the template and no snapshot policies are defined as well.

5.0 Virtual Machine Vulnerability Assessment:

5.1 Misconfigured Security Groups: Overly Permissive Inbound Rules:

- **Description:** Misconfigured security groups with overly permissive inbound rules pose a significant security risk by allowing excessive or unnecessary access to networked resources. Security groups function as virtual firewalls in cloud environments, controlling inbound and outbound traffic to instances based on defined rules.
- **Impact:** Allowing SSH from 0.0.0.0/0 means anyone on the internet can attempt to access your EC2 instance. This exposes the instance to brute-force attacks, credential stuffing, and exploitation of SSH vulnerabilities. Allowing HTTP from 0.0.0.0/0 is common for public-facing websites but can expose unsecured traffic to the internet. If HTTP is enabled without HTTPS**,** attackers can intercept and manipulate traffic via Man-in-the-Middle (MITM) attacks. Attackers can capture unencrypted traffic containing sensitive information and since the instance hosts a web app, it may be vulnerable to web application attacks such as SQL injection, XSS, and RCE.
- **Risk:** The risk level For SSH Brute Forcing is high because it provides direct access to a server's operating system if compromised. An attacker can have full control of the system, install malware, exfiltrate data, or create backdoors. The risk level for using HTTP is medium because although it does not provide full access to the system, it can still be exploited if it transmits sensitive encrypted data, has misconfigurations that can lead to unauthorized access, or has web application vulnerabilities (e.g. SQL injection, XSS).
- **Remediation:** Restrict SSH access to trusted IPs or use AWS Systems Manager Session Manager instead. Use key-based authentication such as SSH key pairs and restrict key distribution. To remediate HTTP vulnerability, Use AWS Certificate Manager (ACM) to set up an SSL/TLS certificate for HTTPS, deploy a Web Application Firewall to filter out malicious requests, and regularly patch and harden web servers.

- **Evidence:**



EC2 > Security Groups > sg-031baaa6963278db0 - Juice-Shop-Stack-WebServerSecurityGroup-mrxo0j1QVbYO > Edit inbound rules

Edit inbound rules Info

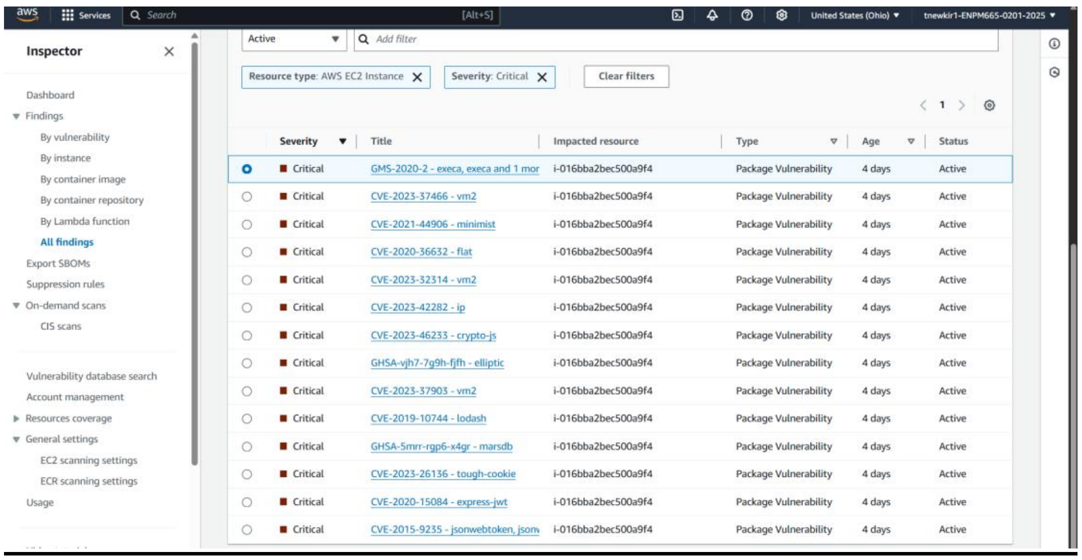
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-00e84d67a9d0251e1	Custom TCP	TCP	3000	Custom	0.0.0.0/0	
sgr-07cd9a130e5939767	SSH	TCP	22	Custom	0.0.0.0/0	
sgr-0b671d2f58ef0badd	HTTP	TCP	80	Custom	0.0.0.0/0	
sgr-0899256508c442baa	HTTPS	TCP	443	Custom	0.0.0.0/0	

5.2 Outdated Node Package Manager (NPM) version and Lack of Security Features:

- **Description:** An outdated NPM version refers to the use of an older release of the Node Package Manager that may contain known security vulnerabilities, deprecated functionalities, or lack critical performance and security updates. Running an outdated NPM version increases the risk of supply chain attacks, dependency resolution issues, and exposure to unpatched security flaws.
- **Impact:** Using an outdated npm version creates serious security risks and operational issues. Some notable risks include remote code execution (RCE), credential leaks and token exposure, path traversal and arbitrary file overwrite, supply chain attacks.
- **Risk:** The risk level is critical as older versions are vulnerable to RCE, credential leaks, and supply chain attacks. Broken dependencies have a high-risk level as older versions may not support modern packages, causing installation failures. Lastly, missing security features have a high-risk level as it increases the attack surface for hackers.
- **Remediation:** Update npm to the latest version with the npm install -g npm@latest command and enable npm audit and fix vulnerabilities
- **Evidence:**



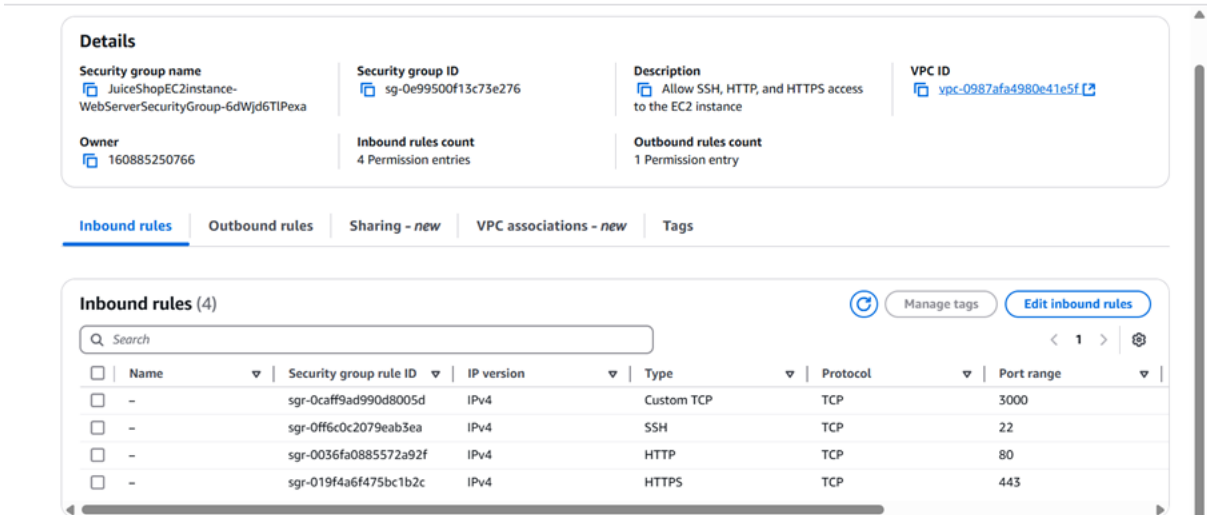
Severity	Title	Impacted resource	Type	Age	Status
Critical	GMS-2020-2 - execa, execa and 1 mor	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2023-37466 - vm2	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2021-44906 - minimist	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2020-36632 - flat	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2023-32314 - vm2	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2023-42282 - ip	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2023-46233 - crypto-js	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	GHSA-vjh7-7g9h-fjfh - elliptic	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2023-37903 - vm2	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2019-10744 - lodash	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	GHSA-5mrr-rgp6-x4gr - marsdb	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2023-26136 - tough-cookie	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2020-15084 - express-jwt	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active
Critical	CVE-2015-9235 - jsonwebtoken, jsonw	i-016bba2bec500a9f4	Package Vulnerability	4 days	Active

6.0 Network Security Assessment:

Target Infrastructure: OWASP Juice Shop deployed on AWS via CloudFormation.
Scope: Analysis of network configurations, security groups, subnets, and traffic rules.

6.1 Finding One: SSH (Port 22) Open to the Internet:

- **Description:**
- **AWS Console**: Reviewed security group inbound rules and identified unrestricted access (0.0.0.0/0) to port 22.
- **Evidence**: Security group rule sgr-0ff6c0c2079eab3ea
- **Impact**: The security group JuiceShopEC2InstanceWebServerSecurityGroup allows unrestricted SSH access from 0.0.0.0/0. Attackers can brute-force credentials to gain control of the EC2 instance (i-0186aa6de92d0b4b1.



Details

Security group name	Security group ID	Description	VPC ID
JuiceShopEC2instance-WebServerSecurityGroup-6dWjd6TlPexa	sg-0e99500f13c73e276	Allow SSH, HTTP, and HTTPS access to the EC2 instance	vpc-0987afa4980e41e5f

Owner	Inbound rules count	Outbound rules count
160885250766	4 Permission entries	1 Permission entry

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0caff9ad990d8005d	IPv4	Custom TCP	TCP	3000
-	sgr-0ff6c0c2079eab3ea	IPv4	SSH	TCP	22
-	sgr-0036fa0885572a92f	IPv4	HTTP	TCP	80
-	sgr-019f4a6f475bc1b2c	IPv4	HTTPS	TCP	443

- **Risk**: Critical Risk. Public SSH access violates the principle of least privilege and exposes the instance to credential-based attacks. SSH is a prime target for attackers due to its ability to grant full system control. If compromised, attackers can:
  - Install malware/ransomware.
  - Steal sensitive data (e.g., databases, credentials).
  - Use the instance as a pivot point for lateral movement.
- **Consequence:** A single brute-forced SSH login could lead to complete infrastructure compromise.
- **Remediation:**
  - Restrict SSH to a bastion host in a private subnet.
  - Use AWS Systems Manager Session Manager for secure access.

- Update the security group to allow SSH only from trusted IP ranges (e.g., corporate VPN).

➢ **Reference:**

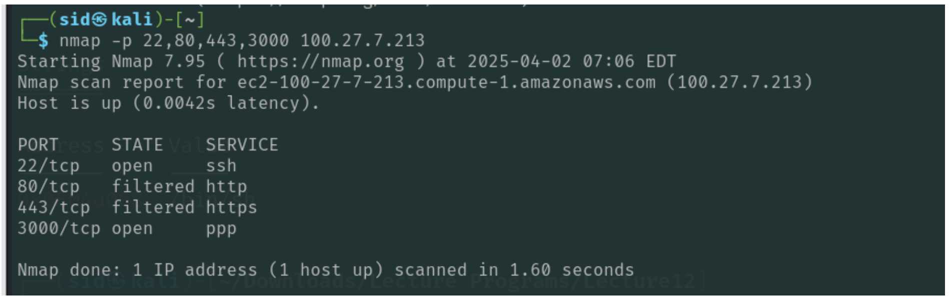
- Inbound rule sgr-0ff6c0c2079eab3ea (SSH from 0.0.0.0/0).
- Public IP 100.27.7.213 confirms exposure.

6.2 Finding Two: Juice Shop Port (3000) Publicly Exposed:

➢ **Description:**

➢ **AWS Console**: Noted a custom TCP rule (sgr-0caff9ad990d80054) allowing global access to port 3000.

➢ **Nmap Scan**: Verified port 3000 was open on the EC2 instance's public IP.



```
┌──(sid㉿kali)-[~]
└─$ nmap -p 22,80,443,3000 100.27.7.213
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-02 07:06 EDT
Nmap scan report for ec2-100-27-7-213.compute-1.amazonaws.com (100.27.7.213)
Host is up (0.0042s latency).

PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   filtered http
443/tcp  filtered https
3000/tcp open     ppp

Nmap done: 1 IP address (1 host up) scanned in 1.60 seconds
```

➢ **Evidence**: Screenshot of Juice Shop interface.

➢ **Impact**: Port 3000 is open to 0.0.0.0/0, exposing the application to attacks like SQL injection or DDoS.

➢ **Risk**: High Risk. Public exposure of unhardened applications can lead to data breaches or service disruption. Port 3000 hosts the Juice Shop application, which is vulnerable to:

- **SQL Injection**: Attackers could dump customer databases.
- **Cross-Site Scripting (XSS)**: Steal user sessions or credentials.
- **DDoS Attacks**: Overload the application, causing downtime.

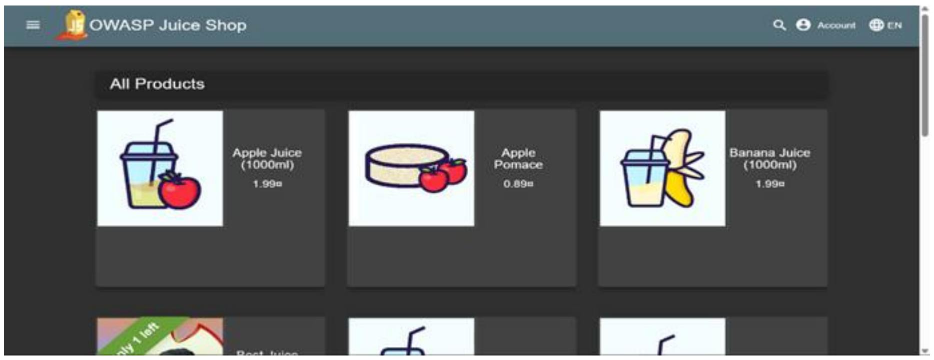
➢ **Consequence**: Exploitation could lead to data breaches, financial loss, or reputational damage.

➢ **Remediation**:

- Restrict port 3000 to trusted IP ranges.
- Deploy behind an AWS Application Load Balancer (ALB) with WAF.

➢ **Reference**:

- Inbound rule sgr-0caff9ad990d80054 (TCP port 3000).



- ➢ Juice Shop interface confirms accessibility.

6.3 Finding Three: HTTP/HTTPS Open to the World:

- ➢ **Description:**
- ➢ **AWS Console:** Observed security group rules (sgr-0036fa0885572a92f and sgr-0194a6475bc1b2c) allowing HTTP/HTTPS from 0.0.0.0/0.
- ➢ **Evidence**: Security group details
- ➢ **Impact**: Security group allows HTTP (port 80) and HTTPS (port 443) from 0.0.0.0/0
- ➢ **Risk**: High Risk. Attackers can exploit vulnerabilities in the Juice Shop application. Open ports 80/443 expose the web server to:
  - ○ **Exploitable Vulnerabilities**: Outdated libraries (e.g., Log4j) could lead to remote code execution.
  - ○ **Phishing Attacks**: Host fake login pages if the server is compromised.
- ➢ **Consequence**: Unauthorized access to sensitive user data or server takeover.
- ➢ **Remediation**:
  - o Restrict HTTP/HTTPS to specific IP ranges.
  - o Enable AWS Shield for DDoS protection.
- ➢ **Reference**:
  - o Rules sgr-0036fa0885572a92f (HTTP) and sgr-0194a6f475bc1b2c (HTTPS).

6.4 Finding Four: Unrestricted Outbound Traffic:

- ➢ **Description**:
- ➢ **AWS Console**: Noted the default outbound rule (0.0.0.0/0) in the security group.
- ➢ **Evidence**: Security group outbound rules.
- ➢ **Impact**: Default outbound rule allows all traffic and Risk of data exfiltration or command-and-control (C2) communication.
- ➢ **Risk**: Medium Risk. Uncontrolled outbound traffic complicates incident response and data loss prevention. Uncontrolled address allows:

- Data Exfiltration: Stolen data sent to attacker-controlled servers.
- C2 Communication: Malware "phoning home" for instructions.

➢ **<u>Consequence</u>**: Prolonged undetected breaches and regulatory penalties.

➢ **<u>Remediation</u>**:
- Define explicit outbound rules.
- Use VPC endpoints for AWS services.

➢ **<u>Reference</u>**:
- Outbound rules are not explicitly restricted.

6.5 Finding Five: EC2 Instance Deployed in Public Subnet:

➢ **<u>Description:</u>**

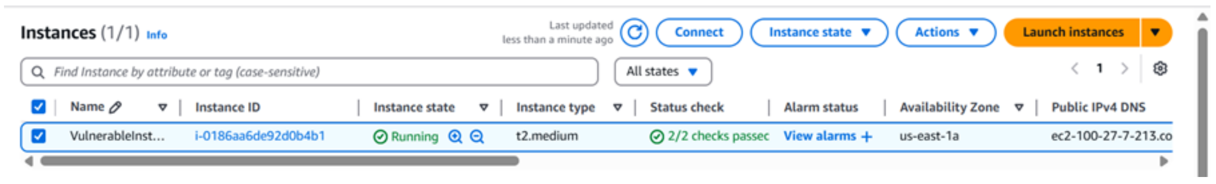
➢ **<u>CloudFormation Template</u>**: Identified AssociatePublicIpAddress: true in the EC2 instance configuration.

➢ **<u>AWS Console</u>**: Confirmed public IP (100.27.7.213) and subnet placement.

➢ **<u>Impact</u>**: EC2 instance is in a public subnet with AssociatePublicIpAddress: true. Direct internet exposure increases the risk of lateral movement.

➢ **<u>Risk</u>**: Critical Risk. Public subnets lack layered defenses, increasing the risk of lateral movement. Public subnets lack layered defenses, allowing attackers to:
- Directly target the instance via its public IP.
- Bypass perimeter controls (e.g., firewalls).

➢ **<u>Consequence</u>**: Increased risk of ransomware, data leaks, or cryptojacking.


Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
VulnerableInst...	i-0186aa6de92d0b4b1	Running	t2.medium	2/2 checks passec	View alarms +	us-east-1a	ec2-100-27-7-213.co

➢ **<u>Remediation</u>**:
- Migrate instance to a private subnet.
- Use NAT gateways for outbound traffic.

➢ **<u>Reference</u>**:
- CloudFormation stack deploys EC2 in a public subnet.
- Public IP 100.27.7.213 confirms exposure.

6.6 Finding Six: Insecure Network Segmentation:

- **Description:**
- **CloudFormation Template**: Noted all resources (web/database) were deployed in the same subnet.
- **AWS Console**: Verified subnet design lacked isolation.
- **Impact:** Web and database tiers share the same subnet, allowing attackers to pivot laterally. Route tables permit unrestricted outbound traffic to 0.0.0.0/0.
- **Risk:** High (Likelihood: Medium | Impact: High). Shared subnets allow attackers to:
  - Pivot from the web tier (Juice Shop) to databases.
  - Exploit internal services (e.g., MariaDB on port 3306).

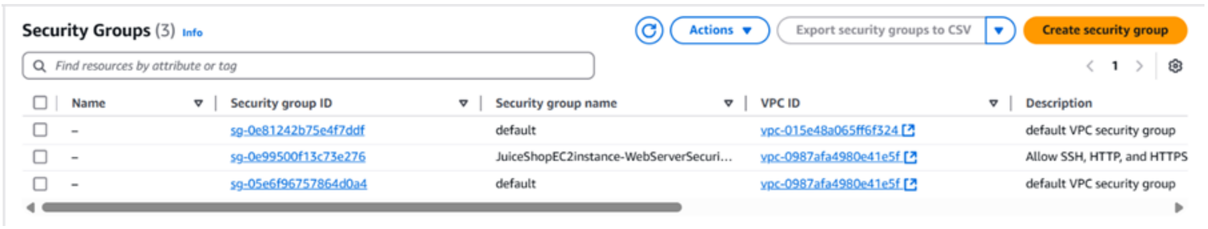
- **Consequence**: A single compromised component can lead to full infrastructure takeover.
- **Remediation:**
  - Deploy databases in private subnets.
  - Add NACLs to block inbound traffic except from the web tier (port 3306 for MariaDB).
  - Restrict route tables to specific AWS services.

- **Evidence:**
  - CloudFormation stack lacks private subnet definitions.
  - No NACLs configured.

6.7 Finding Seven: Missing Network ACLs (NACLs):

- **Description**:
- **AWS Console**: Checked the VPC’s Network ACLs and found no custom rules.
- **Evidence**: Security group list showing NACLs missing.
- **Impact**: No NACLs configured, relying solely on security groups.
- **Risk**: Medium Risk. Without NACLs, unauthorized traffic could bypass security group rules. NACLs provide stateless traffic filtering, complementing stateful security groups. Without NACLs:
  - Attackers can bypass security group rules via spoofed IPs.
  - No subnet-level defense against port scanning.
- **Consequence**: Reduced defense-in-depth against advanced attacks.
- **Remediation**:
  - Implement NACLs with default-deny rules.
  - Segment web and database tiers.

- ➢ **Reference:**
  - ○ Only security groups listed; NACLs missing.


Security Groups (3) Info
Actions
Export security groups to CSV
Create security group
Find resources by attribute or tag
1

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0e81242b75e4f7ddf	default	vpc-015e48a065ff6f324	default VPC security group
-	sg-0e99500f13c73e276	JuiceShopEC2instance-WebServerSecuri...	vpc-0987afa4980e41e5f	Allow SSH, HTTP, and HTTPS
-	sg-05e6f96757864d0a4	default	vpc-0987afa4980e41e5f	default VPC security group

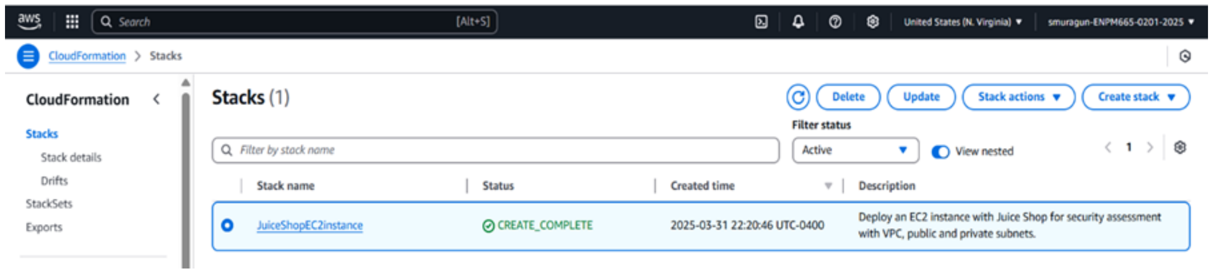
6.8 Finding Eight: Weak MariaDB Password:

- ➢ **Description:**
- ➢ **CloudFormation Template:** Analyzed the UserData script and found hardcoded credentials (SET PASSWORD ... 'vulnerable').
- ➢ **Evidence:** UserData script snippet.
- ➢ **Impact:** MariaDB root password is set to vulnerable in the UserData script. If the database is network-accessible, attackers can exploit weak credentials.
- ➢ **Risk:** High Risk. Default/weak credentials are the #1 cause of database breaches. Attackers can:
  - ○ Dump/delete sensitive data (e.g., customer PII).
  - ○ Escalate privileges to AWS services via the database.
- ➢ **Consequence:** Regulatory fines (e.g., GDPR) and loss of customer trust.
- ➢ **Remediation:**
  - ○ Use a strong, randomly generated password stored in AWS Secrets Manager.
  - ○ Restrict database access to the EC2 instance via security groups.
- ➢ **Reference:**
  - ○ UserData script in CloudFormation template.

6.9 Finding Nine: No Logging/Monitoring:

- ➢ **Description:**

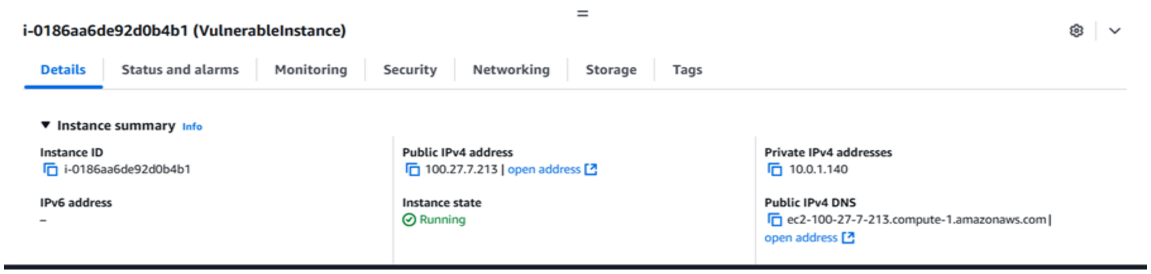
- ➢ **AWS Console:** Checked CloudWatch Logs and VPC Flow Logs, which were not enabled.
- ➢ **Evidence:** Instance monitoring section showed no alarms.



Stacks (1)

Stack name	Status	Created time	Description
JuiceShopEC2instance	CREATE_COMPLETE	2025-03-31 22:20:46 UTC-0400	Deploy an EC2 instance with Juice Shop for security assessment with VPC, public and private subnets.

- ➢ **Impact**: Lack of VPC Flow Logs or CloudWatch monitoring. Inability to detect network intrusions or suspicious traffic.



i-0186aa6de92d0b4b1 (VulnerableInstance)

Instance summary

Instance ID: i-0186aa6de92d0b4b1
IPv6 address: –
Public IPv4 address: 100.27.7.213
Instance state: Running
Private IPv4 addresses: 10.0.1.140
Public IPv4 DNS: ec2-100-27-7-213.compute-1.amazonaws.com

- ➢ **Risk**: Medium Risk. Attacks go undetected for months (average breach detection time: 207 days). No forensic data for incident response.
- ➢ **Consequence**: Prolonged attacker dwell time and irreversible damage.
- ➢ **Remediation**:
  - o Enable VPC Flow Logs and CloudWatch alarms for network traffic.
- ➢ **Reference**:
  - o No monitoring alarms listed for the instance.

Risk Prioritization Matrix:

Risk	**Likelihood (1-5)**	**Impact (1-5)**	**Severity**	**Justification**
SSH Open to the Internet	5 (High)	5 (Critical)	**Critical**	Port 22 is a common target; full system compromise is likely.
Juice Shop Port (3000) Exposed	4 (Medium-High)	4 (High)	**High**	Direct app exposure invites SQLi/DDoS attacks.

HTTP/HTTPS Open to the World	4 (Medium-High)	4 (High)	**High**	Ports 80/443 are widely scanned; web app flaws are exploitable.
Public Subnet Exposure	5 (High)	5 (Critical)	**Critical**	Public IPs are easily discoverable; lateral movement is likely.
Unrestricted Outbound Traffic	3 (Medium)	3 (Medium)	**Medium**	Data exfiltration is possible but requires prior compromise.
Missing NACLs	3 (Medium)	3 (Medium)	**Medium**	Security groups block most traffic, but NACLs add defense-in-depth.
Weak MariaDB Password	4 (Medium-High)	4 (High)	**High**	Credential brute-forcing is trivial if DB is network-accessible.
No Logging/Monitoring	2 (Low-Medium)	3 (Medium)	**Medium**	Delayed detection increases breach impact, but no direct exploitation path.
Insecure Network Segmentation	3 (Medium)	4 (High)	**High**	Shared subnets enable lateral movement; critical if web tier is compromised.

➢ **<u>Recommendations:</u>**

1. **Immediate Fixes**:

➢ Restrict SSH (22) and Juice Shop (3000) to trusted IPs.

➢ Encrypt EBS volumes and MariaDB storage.

2. **Long-Term Enhancements**:

➢ Migrate EC2 to a private subnet.

➢ Enable VPC Flow Logs and AWS WAF.

➢ Replace hardcoded credentials with Secrets Manager.

7. Logging and Monitoring Analysis:

7.1 Finding One: Insufficient Log Retention Policy:

- **Impact:** Without a sufficient log retention policy, critical security events may not be available for forensic analysis, delaying incident response and potentially allowing attackers to cover their tracks.
- **Risk:** Limited log retention reduces an organization's ability to investigate security breaches, comply with regulatory requirements, and identify persistent threats.
- **Remediation:** Implement a log retention policy that aligns with compliance requirements (e.g., NIST, PCI-DSS) and business needs. Store logs securely in a centralized location with appropriate access controls and regular backups.
- **Reference:**
  - NIST Special Publication 800-92: Guide to Computer Security Log Management**.**

7.2 Finding Two: Lack of Real-Time Alerting for Security Events:

Traditional Logging	**SIEM with Real-Time Alerting**
Logs are stored for later analysis	Events are monitored in real-time
Manual review required for threat detection	Automated correlation of events to detect anomalies
High risk of delayed detection	Immediate alerts help in quick response
No automated incident response	Supports automated threat response workflows

- **Impact:** Without real-time alerts, security teams may fail to detect and respond to threats in a timely manner, increasing the risk of prolonged breaches and data exposure.
- **Risk:** Delayed detection of malicious activities, such as unauthorized access or data exfiltration, can lead to significant financial and reputational damage.
- **Remediation:** Implement a Security Information and Event Management (SIEM) solution with real-time alerting and automated responses for critical security events. Regularly test and fine-tune alerting rules to reduce false positives.
- **Reference:**
  - MITRE ATT&CK Framework – Detection and Response

7.3 Finding Three: Inadequate Log Integrity Protections:

Logging Without Integrity Protections	Logging With Integrity Protections
Logs can be modified or deleted	Logs are immutable and protected
No evidence preservation	Cryptographic hashing ensures integrity
Risk of tampering	Logs are stored in a tamper-proof system
Low forensic reliability	Logs provide trustworthy forensic evidence

- **Impact:** If log files can be modified or deleted without detection, attackers can erase evidence of their activities, making forensic investigations impossible.
- **Risk:** Compromised log integrity undermines the reliability of security monitoring and incident response, increasing the risk of undetected and untraceable attacks.
- **Remediation:** Enable cryptographic hashing and integrity checks for logs. Store logs in a tamper-resistant system, such as an immutable storage solution or a dedicated logging service with restricted write permissions.
- **Reference:**
  - CIS Controls v8 – Control 8: Audit Log Management

7.4 Finding Four: Logging of Sensitive Data Without Masking:

Risk Level	**Examples of Sensitive Data Logged**	**Impact**
High	User passwords, API keys, session tokens	Can lead to credential leaks and account takeover
Medium	Personal Identifiable Information (PII) (e.g., SSN, address, phone numbers)	Data privacy violations (GDPR, HIPAA fines)
Low	Internal system messages, error logs with metadata	Can provide attackers with reconnaissance data

- **Impact:** Logging sensitive data, such as passwords, API keys, or personal information, can expose it to unauthorized access and increase compliance risks.
- **Risk:** Unprotected sensitive data in logs may lead to regulatory violations (e.g., GDPR, HIPAA) and potential exploitation by attackers who gain access to logs.
- **Remediation:** Configure logging mechanisms to mask or redact sensitive data. Use environment variables or secure vault solutions to handle secrets instead of logging them. Regularly audit logs for compliance with data protection policies.
- **Reference:**
    - OWASP Logging Cheat Sheet

8.0 Disaster Recovery Assessment:

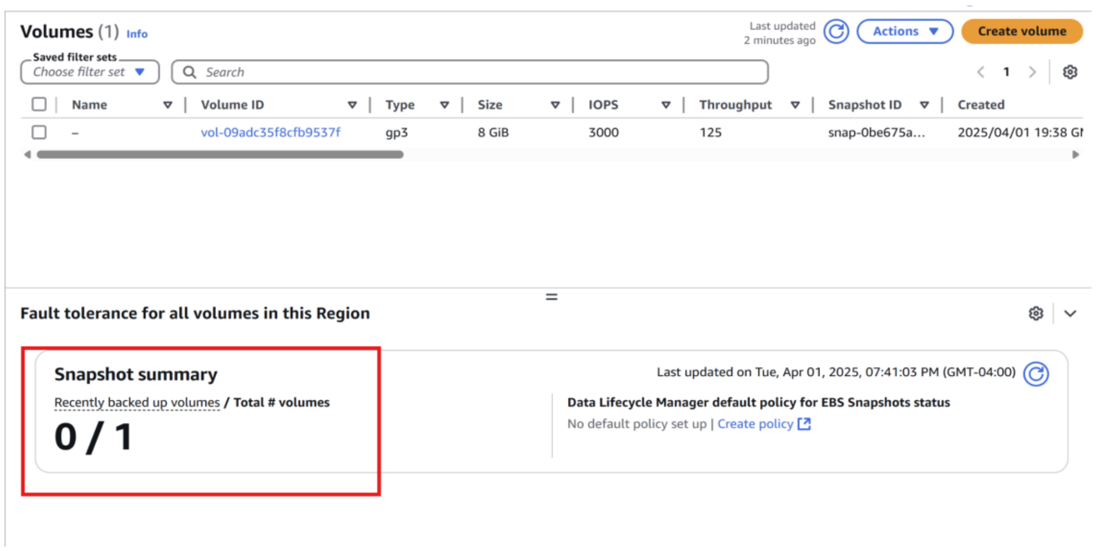
8.1 Finding 1:

- **Description:** Nonexistent EBS volume snapshots are found present in the CloudFormation template or AWS Console, meaning the data is not being properly backed up.
- **Risk:** High Risk of Data Loss from Lack of EBS Volume Snapshots
- **Impact:** By not creating EBS volume snapshots, the data stored in the EBS volumes for the E-commerce application that is hosted on the EC2 instance, is at

risk for data loss and extended application downtime if a disaster were to occur. If a disaster does occur and something happens to the EBS volume, whether from human error or some other event, data recovery efforts would be difficult, and it would ultimately lead to extended downtime.

- ➢ **Remediation:** The easiest way to remediate the risk of data loss from lack of EBS volume snapshots is by manually creating an EBS snapshot using the AWS Console. You can do this by clicking on the EBS volume, then clicking on actions, and then "Create snapshot". If you do not want to manually do this, there are automated technologies that will do it for you. For example, you can also use Amazon Data Lifecycle Manager to automate the creation, retention, and deletion of your EBS snapshots.
- ➢ **Reference:**
https://docs.aws.amazon.com/ebs/latest/userguide/ebs-creating-snapshot.html
https://docs.aws.amazon.com/ebs/latest/userguide/snapshot-lifecycle.html

- ➢ **Screenshot of Finding:**



Volumes (1) Info

Last updated 2 minutes ago

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created
–	vol-09adc35f8cfb9537f	gp3	8 GiB	3000	125	snap-0be675a...	2025/04/01 19:38 GI

Fault tolerance for all volumes in this Region

Snapshot summary

Recently backed up volumes / Total # volumes

0 / 1

Last updated on Tue, Apr 01, 2025, 07:41:03 PM (GMT-04:00)

Data Lifecycle Manager default policy for EBS Snapshots status

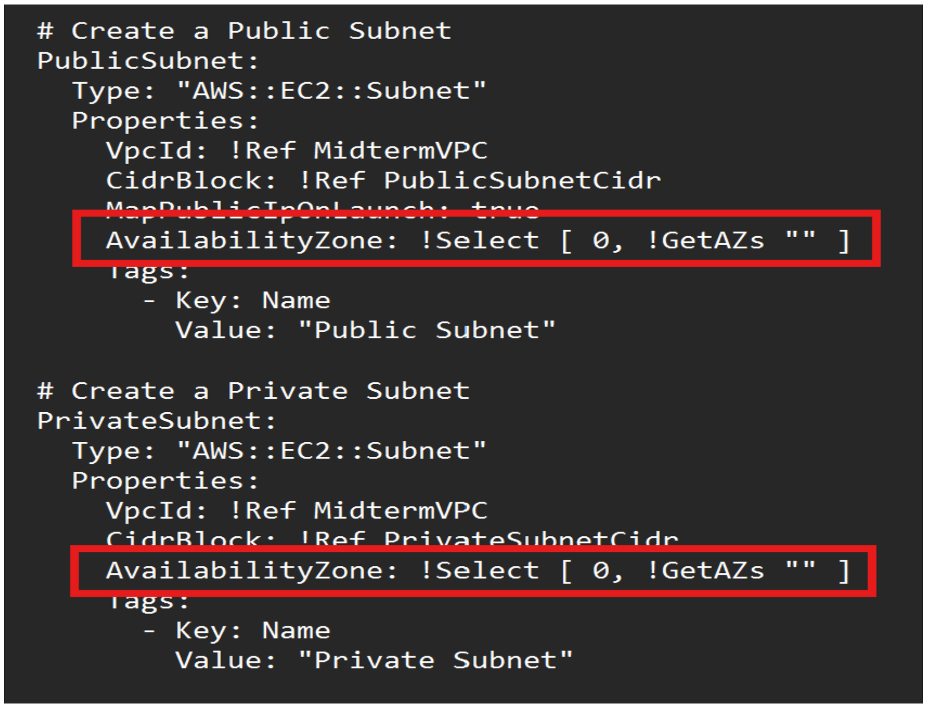
No default policy set up | Create policy

8.2 Finding 2:

- ➢ **Description:** Both Public and Private Subnets are created in the same availability zone (AZ) within the CloudFormation template, which creates a single point of failure for the subnets.
- ➢ **Risk:** High Risk of Extended Downtime due to a Single Point of Failure in Availability Zones (AZ)
- ➢ **Impact:** If a disaster were to occur and the availability zones that the public and private subnets were using went down, the EC2 instance that is hosted in the public subnet would become unavailable and users would not be able to access

the E-commerce application. Similarly, the private subnet would also be affected as it was in the same availability zone as the public subnet.

- **Remediation:** Per AWS documentation, when configuring availability zones, it is best practice to configure one subnet per unique availability zone. With this in mind, to remediate the current configuration the public subnet should be in its own unique availability zone, and the private subnet should be in a different unique availability zone. This way you can maintain availability even if a disaster were to occur. If a disaster were to knock out one availability zone, only one of your subnets would be affected instead of both of them.
- **Reference:** https://docs.aws.amazon.com/whitepapers/latest/best-practices-for-deploying-amazon-appstream-2/availability-zones.html
- **Screenshot of Finding:**



```
# Create a Public Subnet
PublicSubnet:
  Type: "AWS::EC2::Subnet"
  Properties:
    VpcId: !Ref MidtermVPC
    CidrBlock: !Ref PublicSubnetCidr
    MapPublicIpOnLaunch: true
    AvailabilityZone: !Select [ 0, !GetAZs "" ]
    Tags:
      - Key: Name
        Value: "Public Subnet"

# Create a Private Subnet
PrivateSubnet:
  Type: "AWS::EC2::Subnet"
  Properties:
    VpcId: !Ref MidtermVPC
    CidrBlock: !Ref PrivateSubnetCidr
    AvailabilityZone: !Select [ 0, !GetAZs "" ]
    Tags:
      - Key: Name
        Value: "Private Subnet"
```

8.3 Finding 3:

- **Description:** The CloudFormation template does not have an AWS Backup Plan in place for the E-commerce platform, meaning that AWS resources are not being backed up properly.
- **Risk:** High Risk of Data Loss from Lack of AWS Backup Strategies/Configurations
- **Impact:** If a disaster were to occur, the risk of data loss and application availability being impacted is very high, due to the lack of backup configurations within the CloudFormation template. As a result of a lack of AWS backup

configurations, AWS resources would be greatly affected if a disaster were to occur because there were not any backup configurations in place.

- **Remediation:** To help remediate the risk of data loss from disasters, you can use AWS Backup. AWS Backup will help you centralize and automate the data protection process for services offered by AWS. There are several things you can do with it including automating backups, auditing and reporting, implementing backup types like incremental backups, encrypt your backups, and more. As an added bonus you can even configure this in your CloudFormation template so that when you deploy an instance you will already have a reliable service managing your backups and associated backup tasks. Ultimately, if you implement AWS Backup into your CloudFormation template you can help mitigate the risk of data loss during a disaster.
- **Reference:**
  https://docs.aws.amazon.com/aws-backup/latest/devguide/creating-a-backup-plan.html
  https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-backup-backupplan.html

8.4 Finding 4:

- **Description:** The CloudFormation template does not include configurations or methods for backing up the E-commerce data stored in MariaDB, which means database data is not being properly backed up.
- **Risk:** High Risk of Database Data loss due to Lack of Database Backups
- **Impact:** If a disaster were to occur or if something happened to the server hosting the MariaDB database, the E-commerce application data that was being stored within the database would be extremely hard to recover and would lead to extended periods of downtime. If a disaster did occur, the difficulties in data recovery would ultimately stem from the lack of backups for the database.
- **Remediation:** One way to remediate the risk of data loss from a disaster occurring due to a lack of database backups, is to implement Amazon RDS. Implementing Amazon RDS will allow you to automate the setup, management, backup, and operation tasks for MariaDB. By implementing this using this useful service, you can ensure that you are implementing proper redundancy and availability techniques that would aid in data restoration if a disaster affected your EC2 instance. You can implement Amazon RDS directly into your CloudFormation template and when you deploy it, it will automatically setup your MariaDB database and handle other operational setup tasks.
- **Reference:**
  https://aws.amazon.com/rds/mariadb/

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-rds-dbinstance.html

8.5 Finding 5:

- ➢ **Description:** No Disaster Recovery plan or strategy documents are in place for the E-commerce platform.
- ➢ **Risk:** High Risk of Data Loss and Extended Downtime due to Lack of a Disaster Recovery Plan
- ➢ **Impact:** If a disaster occurred, the impact to the E-commerce platform, its data, and availability, would be detrimental due to not having a documented disaster recovery plan or strategy in place. Data recovery times would be lengthy and exhausting, availability would be greatly affected, and more.
- ➢ **Remediation:** To remediate the risk of data loss and extended downtime, you can implement a disaster recovery plan. Implementing a disaster recovery plan will help you prepare for potential disasters that may occur in the future. They will outline certain procedures the organization can take in order to restore data if any was lost, restore systems, and other operations. Ultimately, the purpose of a disaster recovery plan is to minimize downtime and restore operations as quickly as possible.
- ➢ **Reference:** https://www.vmware.com/topics/disaster-recovery-plan

8.6 Proposed Backup and Disaster Recovery Strategy:

Before you can implement a backup and disaster recovery strategy and documentation, you must take several factors into consideration. Factors such as backup frequency, cost, type of backup (e.g. full, incremental, differential, etc), data retention and testing processes, geographical locations for storing your backups, recovery time objective (RTO), recovery point objective (RPO), etc. Additionally, there are other factors to consider. For example, if you are using AWS services, AWS offers backup and data retention services that you can manually configure or you can configure them to automatically do it for you. One thing that you need to keep in mind is the cost of some of these services. Once you determine direction and factors that best suit your organization, you can start developing your disaster recovery plan. It will typically include the factors discussed above and then it will also outline clear instructions for accessing the backups, restoring systems and data, the designated roles and responsibilities for employees within the organization, and more.