Name Yvonne Otuoniyo

Subject seidea cloud bootcamp project

Design a scalable cloud infrastructure using AWS or Azure services taking into account sage's requirements for data security and operational efficiency

Detailed design

Implement IAM roles and policies for access control

Use KMS/ key Vault for data encryption

Configure security groups/ network security groups for network segmentation

Enable CloudWatch / monitor logging and monitoring

Compute

Use EC2 / virtual machine for scalable computing

Implement containerization using ECS / Kubernetes service

Utilize serverless computing with lambda/functions

Storage

Store static assets in s3 /blob storage

Use EBS / DISK storage for compute instances

Implement RDS/SQL data base for relational databases

Networking

Create a VPC/ Virtual Network for secure networking

Segment VPC into public and private subnets

Configure ELB/load balancer for traffic distribution

Operational efficiency measures

Automation. use cloud formation resources manager

Scaling. use auto scaling /scale sets

Monitoring. use CloudFormation / resources manager

Scaling . use auto scaling / scale sets

Monitoring use cloud watch / monitor

Cost optimization . use AWS cost explorer /Azure cost estimator

Data security measures

Encryption . use KMS/ key vault

Access control. Use IAM roles and policies

Network segmentation . use security groups / network security groups

Monitoring . use CloudWatch / monitor

Provide a network diagram detailing service like virtual machines, storage, networking and data solution.

Answer

Internet

+----------------------- --

+-------------------------

Firewall

+-----------------------

+------------------------

Balancer Load

ALB) (ELB/

```
+---------------------------

+-----------------------------


Network                                                 Virtual

VNet                                                    (VPC

+--------------------------------




+-----------------------------------

                                                        Subnets

(public /Private)

+-------------------------------




+--------------------------------

Machines                                                Virtual

VMs                                                     (EC2/


+-----------------------------------

                                                        Storage

                                                        (S3/

Blob)

+------------------------------------


+---------------------------------------


Database   Storage                                      (RDS/

SQL)
```

+---------------------------------------

+--------------------------------------

Database storage

DISK) (EBS

+-----------------------------

Explain why your chosen architecture supports sage cloud based financial operational and ensure compliance

Scalability provides scalable infrastructure to handle increasing financial transaction volumes

Reliability high availability and redundancy ensure uninterrupted financial operations

Security robust security features protect sensitive financial data

Performance optimized infrastructure ensure fast transaction processing

Collaboration . integrated tools enable seamless team collaboration

Ensure compliance

Data sovereignty. Data residency and compliance with regional regulations (example GDPR, HIPAA)

Financial regulations. Compliance with financial regulations (example , PCI -DSS, SOX

Security standards adherence to industry recognized security standards exampleISO27001

Data encryption. End to end encryption for sensitive data

2. implement an IMA solution using RBAC and MFA to manage users' permission

IAM solution components

Steup identity provider (Idp)

Choose an Idp example AWS IAM, Azure Active Directory)

Configure Idp settings example user authentication, group management

Integrate Idp with user directory example LDAP, Active Directory

Implement RBAC System

Define roles and permissions example admin, developer, read only

Create role hierarchies' examples admin developer

Assign users to roles

Configure role-based access control policies

Implement MFA solution

Choose an MFA solution example AWS MFA, google authenticator

Configure MFA settings examples authentication method, token expiration

Integrate MFA with Idp

Integrate RBAC and MFA

Configure RBAC to require MFA for role assignment

Ensure MFA is enforced for all role-based access

Test and validate

Test RBAC and MFA configuration

Validate user access and permissions

Conduct security audits and penetration testing

RBAC Configuration

Roles

Admin full access

Developer read write access

Read only read only access

Permissions

Admin create, read, update, delete

Develop read update

Read only read

Role Hierarchies

Admin >Developer

Developer >read only

MFA Configuration

Authentication method

Time based onetime password (TOTP)

HMAC based onetime password (HoTP)

Token expiration :30 seconds

MFA enforcement: required for all roles

Benefits

Improved security through MFA

Simplified user management with RBAC

Fine -grained access control

Compliance with regulatory requirement

Provide a sample IAM Policy for managing access to sensitive data, ensuring that only authorized personnel can access confidential financial data

```
Version 2012- 10-17

‘statement “[

        “sid”

“Allow Financial Team Access|”,

        “Effect” : “Allow”,

        “Action” :{

        “s3: GetObject”,

        “s3: Put Object”,

        “s3:DeleteObject”

},

“Resource”:
```

```
“arn:  aws: s3::::financial -data/*”,
        “Condition”: {
        “String Equals” :[
        aws: group”: {
        “Financial Team”
}
}
}
        “sid”:
“DenyUnauthorizedAccess”,
        “Effect”: “Deny”,
        “Action”: {
“s3:GetObject”,
“s3:PutObject”,
“s3DeleteObject”
}
“Resource”:
“arn: aws: s3::::financial -data/*”,
        “condition” :{
        “StringNotEquals”: {
        “aws: group”:{
        “Financial Team”
}
}
}
}
        “sid”
“AllowAuditorsReadAccess”
        “Effect” : “Allow”,
        “Action”:    {
        “s3: GetObject”
```

```
          “Resource”
“arn: aws” :::::::::: financial- data/ *”,
          “condition” : {
          “String Equals” : {
          “aws: groups” : {
          “Auditors”
```

3 a) Outline a plan for encrypting data both at rest and in transit using cloud-native tools such as AWS Key Management Service.

Encryption of Data at Rest

Step 1: Create a KMS Key

- Create a **customer-managed key (CMK)** in AWS KMS for encrypting your data.
- Consider using **symmetric encryption keys** for data encryption.

Step 2: Use KMS to Encrypt Data at Rest

- AWS services such as **S3**, **EBS**, **RDS**, **DynamoDB**, and **EFS** support encryption at rest using KMS.
- Enable encryption for S3 buckets, EBS volumes, RDS databases, or any other storage service you use by specifying the CMK to be used for encryption.

Step 3: Enable Automatic Key Rotation

Step 4: Implement Access Control

- Use IAM roles and policies to control access to the KMS keys. Only authorized users and services should have permission to use the encryption keys.

Encryption of Data in Transit

Step 1: Request an SSL/TLS certificate from ACM.

Step 2: Choose the encryption method (either symmetric or asymmetric keys) for the certificate.

Step 3: Enable automatic renewal of the certificate.

Step 4: Configure your load balancer or application to use the ACM certificate.

b) Highlight the importance of encryption for protecting financial data, referencing relevant compliance requirements like GDPR.

1. Confidentiality and Protection against Unauthorized Access
2. Compliance with regulatory standards
3. Miitigating data breach risk and financial loss
4. Customer trust and confidence

v4 a) Create a security monitoring and incident response strategy, using AWS cloudwatch.

1. Security Monitoring with AWS CloudWatch

Start by gathering logs and events from various AWS services to monitor the security posture of your environment.

Key AWS Services to Monitor:

- **CloudTrail**: Logs all API activity across your AWS environment.
- **VPC Flow Logs**: Captures information about the IP traffic going to and from network interfaces in your VPC.
- **GuardDuty**: A threat detection service that continuously monitors for malicious activity.
- **AWS Config**: Tracks configuration changes to your resources.
- **Amazon S3**: Logs access to sensitive objects stored in S3.
- **IAM**: Monitor for changes to IAM policies, users, roles, and permissions.

Steps:

1. **Enable CloudTrail**: Ensure that AWS CloudTrail is enabled across all regions to capture API activity. Integrate CloudTrail with CloudWatch Logs for real-time monitoring.
2. **Enable VPC Flow Logs**: Capture traffic logs from all VPC interfaces to detect any suspicious traffic patterns.
3. **Activate GuardDuty**: Enable Amazon GuardDuty in all regions to monitor for suspicious activity like unusual API calls, unauthorized access attempts, or anomalous traffic.
4. **Enable S3 Logging**: Ensure S3 server access logging is enabled for monitoring access to sensitive data in S3 buckets.
5. **Integrate AWS Config**: Use AWS Config to continuously monitor the configuration of your AWS resources to ensure they comply with security best practices.

2. Setting Up CloudWatch Metrics and Alarms

Create custom CloudWatch metrics and set up alarms to monitor specific security-related behaviors.

Steps:

1. **Create Custom CloudWatch Metrics**
2. **Set Alarms**: Define CloudWatch Alarms based on critical thresholds.

3. Set Up CloudWatch Logs Insights for Real-Time Querying

CloudWatch Logs Insights enables you to query, analyze, and visualize log data in real-time. Use this for in-depth analysis and ad-hoc queries to detect unusual activity.

Steps:

1. Create queries to detect specific security-related events, such as:
   - Unusual access patterns to critical resources (e.g., repeated access attempts to EC2 instances).
   - Access to S3 buckets from unauthorized IP ranges.
2. Store and automate queries for continuous security monitoring.

Incident Response Strategy

The incident response process involves detecting, responding to, and recovering from security incidents. With AWS CloudWatch, this process can be automated and improved for faster resolution.

Detection

Key Indicators:

- **Anomalous Behavior**: Using GuardDuty findings, CloudTrail logs, and VPC flow logs, you can identify suspicious activity such as unusual logins, data exfiltration attempts, or unauthorized changes to critical resources.
- **Failed Logins/Privilege Escalation**: Track failed login attempts and changes to IAM roles or policies.
- **Traffic Anomalies**: Use VPC Flow Logs to detect anomalous traffic, such as spikes in traffic to sensitive resources or traffic from unexpected sources.

Response

Key Actions:

- **Automated Actions via AWS Lambda**: Automatically respond to incidents by using Lambda functions triggered by CloudWatch Alarms. For example:
    - **Disable compromised IAM user**: Automatically disable an IAM user account after a suspicious login attempt is detected.
    - **Block IP address**: Use Lambda to modify security group rules to block a suspicious IP address.
    - **Revoke access**: Automatically revoke access to S3 buckets or EC2 instances after detecting unauthorized access.
- **Escalation and Notification**: Use Amazon SNS to notify security teams or administrators. Set up CloudWatch Alarms to send notifications to a Slack channel, email, or SMS for immediate attention.

3. Investigation

Steps:

1. Use CloudWatch Logs Insights queries to narrow down suspicious events, such as unusual logins, access patterns, or changes to critical infrastructure.
2. Correlate events from CloudTrail, GuardDuty, and VPC Flow Logs to understand the full scope of the incident.
3. Review any changes made to IAM roles or permissions via AWS Config to check if privilege escalation occurred.

4. Remediation and Recovery

Steps:

1. **Containment**: Use automated workflows (via Lambda or manual interventions) to isolate affected resources, block malicious IPs, or disable compromised accounts.
2. **Forensics and Analysis**: Perform forensic analysis by reviewing logs, CloudTrail data, and other monitoring information to understand the attack's origin and method.

3. **Restoration**: Restore any affected resources, such as EC2 instances, back to a known good state using automated snapshots or backups.

Automated Recovery:

- Use CloudWatch Events to trigger recovery actions like restoring EC2 instances from a snapshot or rolling back a security group change.

3. Continuous Improvement

Security is an ongoing process. After an incident, always evaluate your monitoring strategy to identify potential gaps.

Key Steps:

1. **Post-Incident Review**: After resolving an incident, conduct a post-mortem to understand what happened, what was done well, and where improvements can be made.
2. **Improve Logging and Alarming**: Based on the incident, adjust your CloudWatch alarms and logging configurations. For example, increase the sensitivity of certain alarms or add additional metrics to monitor.
3. **Test and Update**: Regularly test incident response workflows, including Lambda functions, CloudWatch alarms, and recovery procedures, to ensure they work efficiently when needed.

b) Explain how alerts will be triggered and provide a workflow for responding to security breaches, ensuring quick resolution.

CloudWatch Alarms

- **Condition-based Alerts**: CloudWatch alarms are set to trigger based on specific conditions defined in the metrics or logs. These conditions can range from high error rates to specific thresholds for security events.
- **Event-Based Alerts**: CloudWatch can also trigger alarms based on AWS CloudTrail, GuardDuty findings, or other AWS service events (e.g., suspicious API calls, unauthorized access attempts).

Log-Based Alerts

- **CloudWatch Logs Insights**: Custom queries on logs such as AWS CloudTrail logs or VPC Flow Logs are used to detect abnormal patterns, such as multiple failed login attempts or traffic anomalies. Alerts are generated when specific queries return suspicious activity.

GuardDuty Findings

- **Threat Detection Alerts**: Amazon GuardDuty continuously monitors for threats like unusual API calls, potential compromised instances, or other anomalous activity. When GuardDuty detects suspicious activity, it generates findings which are forwarded to CloudWatch Events, triggering alarms.

CloudTrail and AWS Config

- **API Activity Alerts**: CloudTrail logs API calls, which can be analyzed for unauthorized or suspicious actions (e.g., a new IAM policy being created, changes to VPC security groups). CloudWatch alarms are set to trigger based on these event patterns.
- **Config Compliance Alerts**: AWS Config tracks configuration changes, such as changes to IAM roles or security settings. Alerts can be triggered when resources drift from desired security configurations.

Incident Response Workflow for Security Breaches

Once an alert is triggered, it's important to have a clear, structured workflow for responding to security incidents to ensure a fast and effective resolution. The workflow includes the following key stages: **Detection, Triage, Containment, Eradication, Recovery, and Post-Incident Review**.

1. **Detection and Alerting**
    - **Triggering Event**: CloudWatch Alarm is triggered based on pre-defined metrics or log events. For example, a high number of failed logins or a GuardDuty finding indicating compromised credentials.
    - **Notification**: An alert is sent to relevant stakeholders via **Amazon SNS** (email, SMS, or integration with communication tools like Slack) or other monitoring platforms.
2. **Triage**
    - **Initial Investigation**: The security team investigates the alarm. This includes reviewing:
        - **CloudWatch Logs Insights**: Query logs to check for specific details about failed login attempts (e.g., timestamp, IP address).
        - **CloudTrail Logs**: Look for suspicious API activity related to the affected resource.

    - **GuardDuty Findings**: Review threat intelligence on the suspected malicious activity.

- **Assess Severity**: Determine whether the activity is a false positive or a potential security breach. This involves validating the identity of the user, reviewing associated IAM permissions, and assessing any other anomalous behavior.

3. **Containment**

- **Immediate Action**: If the incident is deemed legitimate, take immediate containment actions to limit the scope of the breach:
    - **Disable Compromised Accounts**: Use AWS IAM to disable the compromised IAM user, or use AWS Lambda to automate this action when suspicious activity is detected.
    - **Update Security Groups/Firewall Rules**: Automatically block the suspicious IP by modifying VPC security groups or Network ACLs using Lambda.
    - **Revoke API Keys/Secrets**: Invalidate compromised credentials by rotating API keys or using the AWS Secrets Manager to rotate secrets.
    - **Isolate Compromised Resources**: If an EC2 instance is compromised, consider stopping the instance to prevent further damage.

4. **Eradication**

- **Remove Root Cause**: Identify the root cause of the security incident and take steps to remove it:
    - **Review Logs for Further Compromise**: Analyze logs from CloudTrail, GuardDuty, and VPC Flow Logs to ensure no additional malicious activity is ongoing.
    - **Patch Vulnerabilities**: Apply patches or security updates to any affected resources (e.g., EC2 instances, containers).
    - **Reinforce IAM Policies**: Ensure that IAM roles and policies are correctly configured and follow the principle of least privilege.

5. **Recovery**

- **Restore Operations**: After the root cause is eradicated, begin restoring affected systems and services to normal operations:

  - **Restore EC2 Instances**: If EC2 instances were compromised, restore them from backups or snapshots.
  - **Re-enable Access**: Restore IAM access to users after confirming that their credentials are secure.
  - **Monitor Post-Recovery**: Keep a heightened monitoring posture for a period after recovery to detect any signs of re-compromise.

6. **Post-Incident Review and Improvement**

   - **Root Cause Analysis**: Conduct a post-incident review to determine the cause of the breach and identify any gaps in security measures.
   - **Lessons Learned**: Document findings and make necessary improvements to the security posture.
   - **Update Security Controls**: Based on the findings, enhance CloudWatch monitoring, IAM policies, VPC security, and any other relevant security mechanisms.

4 a) Create a security monitoring and incident response strategy, using AWS cloudwatch.

2. **Security Monitoring with AWS CloudWatch**

Start by gathering logs and events from various AWS services to monitor the security posture of your environment.

Key AWS Services to Monitor:

- **CloudTrail**: Logs all API activity across your AWS environment.
- **VPC Flow Logs**: Captures information about the IP traffic going to and from network interfaces in your VPC.
- **GuardDuty**: A threat detection service that continuously monitors for malicious activity.
- **AWS Config**: Tracks configuration changes to your resources.
- **Amazon S3**: Logs access to sensitive objects stored in S3.
- **IAM**: Monitor for changes to IAM policies, users, roles, and permissions.

Steps:

6. **Enable CloudTrail**: Ensure that AWS CloudTrail is enabled across all regions to capture API activity. Integrate CloudTrail with CloudWatch Logs for real-time monitoring.
7. **Enable VPC Flow Logs**: Capture traffic logs from all VPC interfaces to detect any suspicious traffic patterns.

8. **Activate GuardDuty**: Enable Amazon GuardDuty in all regions to monitor for suspicious activity like unusual API calls, unauthorized access attempts, or anomalous traffic.
9. **Enable S3 Logging**: Ensure S3 server access logging is enabled for monitoring access to sensitive data in S3 buckets.
10. **Integrate AWS Config**: Use AWS Config to continuously monitor the configuration of your AWS resources to ensure they comply with security best practices.

2. Setting Up CloudWatch Metrics and Alarms

Create custom CloudWatch metrics and set up alarms to monitor specific security-related behaviors.

Steps:

3. **Create Custom CloudWatch Metrics**
4. **Set Alarms**: Define CloudWatch Alarms based on critical thresholds.

3. Set Up CloudWatch Logs Insights for Real-Time Querying

CloudWatch Logs Insights enables you to query, analyze, and visualize log data in real-time. Use this for in-depth analysis and ad-hoc queries to detect unusual activity.

Steps:

3. Create queries to detect specific security-related events, such as:
   - Unusual access patterns to critical resources (e.g., repeated access attempts to EC2 instances).
   - Access to S3 buckets from unauthorized IP ranges.
4. Store and automate queries for continuous security monitoring.

Incident Response Strategy

The incident response process involves detecting, responding to, and recovering from security incidents. With AWS CloudWatch, this process can be automated and improved for faster resolution.

Detection

Key Indicators:

- **Anomalous Behavior**: Using GuardDuty findings, CloudTrail logs, and VPC flow logs, you can identify suspicious activity such as unusual logins, data exfiltration attempts, or unauthorized changes to critical resources.
- **Failed Logins/Privilege Escalation**: Track failed login attempts and changes to IAM roles or policies.
- **Traffic Anomalies**: Use VPC Flow Logs to detect anomalous traffic, such as spikes in traffic to sensitive resources or traffic from unexpected sources.

Response

Key Actions:

- **Automated Actions via AWS Lambda**: Automatically respond to incidents by using Lambda functions triggered by CloudWatch Alarms. For example:
  - **Disable compromised IAM user**: Automatically disable an IAM user account after a suspicious login attempt is detected.
  - **Block IP address**: Use Lambda to modify security group rules to block a suspicious IP address.
  - **Revoke access**: Automatically revoke access to S3 buckets or EC2 instances after detecting unauthorized access.
- **Escalation and Notification**: Use Amazon SNS to notify security teams or administrators. Set up CloudWatch Alarms to send notifications to a Slack channel, email, or SMS for immediate attention.

3. Investigation

Steps:

4. Use CloudWatch Logs Insights queries to narrow down suspicious events, such as unusual logins, access patterns, or changes to critical infrastructure.
5. Correlate events from CloudTrail, GuardDuty, and VPC Flow Logs to understand the full scope of the incident.
6. Review any changes made to IAM roles or permissions via AWS Config to check if privilege escalation occurred.

4. Remediation and Recovery

Steps:

4. **Containment**: Use automated workflows (via Lambda or manual interventions) to isolate affected resources, block malicious IPs, or disable compromised accounts.
5. **Forensics and Analysis**: Perform forensic analysis by reviewing logs, CloudTrail data, and other monitoring information to understand the attack's origin and method.
6. **Restoration**: Restore any affected resources, such as EC2 instances, back to a known good state using automated snapshots or backups.

Automated Recovery:

- Use CloudWatch Events to trigger recovery actions like restoring EC2 instances from a snapshot or rolling back a security group change.

3. Continuous Improvement

Security is an ongoing process. After an incident, always evaluate your monitoring strategy to identify potential gaps.

Key Steps:

7. **Post-Incident Review**: After resolving an incident, conduct a post-mortem to understand what happened, what was done well, and where improvements can be made.
8. **Improve Logging and Alarming**: Based on the incident, adjust your CloudWatch alarms and logging configurations. For example, increase the sensitivity of certain alarms or add additional metrics to monitor.
9. **Test and Update**: Regularly test incident response workflows, including Lambda functions, CloudWatch alarms, and recovery procedures, to ensure they work efficiently when needed.

b) Explain how alerts will be triggered and provide a workflow for responding to security breaches, ensuring quick resolution.

CloudWatch Alarms

- **Condition-based Alerts**: CloudWatch alarms are set to trigger based on specific conditions defined in the metrics or logs. These conditions can range from high error rates to specific thresholds for security events.
- **Event-Based Alerts**: CloudWatch can also trigger alarms based on AWS CloudTrail, GuardDuty findings, or other AWS service events (e.g., suspicious API calls, unauthorized access attempts).

Log-Based Alerts

- **CloudWatch Logs Insights**: Custom queries on logs such as AWS CloudTrail logs or VPC Flow Logs are used to detect abnormal patterns, such as multiple failed login attempts or traffic anomalies. Alerts are generated when specific queries return suspicious activity.

GuardDuty Findings

- **Threat Detection Alerts**: Amazon GuardDuty continuously monitors for threats like unusual API calls, potential compromised instances, or other anomalous activity. When GuardDuty detects suspicious activity, it generates findings which are forwarded to CloudWatch Events, triggering alarms.

CloudTrail and AWS Config

- **API Activity Alerts**: CloudTrail logs API calls, which can be analyzed for unauthorized or suspicious actions (e.g., a new IAM policy being created, changes to VPC security groups). CloudWatch alarms are set to trigger based on these event patterns.
- **Config Compliance Alerts**: AWS Config tracks configuration changes, such as changes to IAM roles or security settings. Alerts can be triggered when resources drift from desired security configurations.

Incident Response Workflow for Security Breaches

Once an alert is triggered, it’s important to have a clear, structured workflow for responding to security incidents to ensure a fast and effective resolution. The workflow includes the following key stages: **Detection, Triage, Containment, Eradication, Recovery, and Post-Incident Review**.

1. Detection and Alerting

- **Triggering Event**: CloudWatch Alarm is triggered based on pre-defined metrics or log events. For example, a high number of failed logins or a GuardDuty finding indicating compromised credentials.
- **Notification**: An alert is sent to relevant stakeholders via **Amazon SNS** (email, SMS, or integration with communication tools like Slack) or other monitoring platforms.

2. Triage

- **Initial Investigation**: The security team investigates the alarm. This includes reviewing:
    - **CloudWatch Logs Insights**: Query logs to check for specific details about failed login attempts (e.g., timestamp, IP address).
    - **CloudTrail Logs**: Look for suspicious API activity related to the affected resource.
    - **GuardDuty Findings**: Review threat intelligence on the suspected malicious activity.
- **Assess Severity**: Determine whether the activity is a false positive or a potential security breach. This involves validating the identity of the user, reviewing associated IAM permissions, and assessing any other anomalous behavior.

3. Containment

- **Immediate Action**: If the incident is deemed legitimate, take immediate containment actions to limit the scope of the breach:
    - **Disable Compromised Accounts**: Use AWS IAM to disable the compromised IAM user, or use AWS Lambda to automate this action when suspicious activity is detected.
    - **Update Security Groups/Firewall Rules**: Automatically block the suspicious IP by modifying VPC security groups or Network ACLs using Lambda.
    - **Revoke API Keys/Secrets**: Invalidate compromised credentials by rotating API keys or using the AWS Secrets Manager to rotate secrets.
    - **Isolate Compromised Resources**: If an EC2 instance is compromised, consider stopping the instance to prevent further damage.

10. Eradication

- **Remove Root Cause**: Identify the root cause of the security incident and take steps to remove it:
    - **Review Logs for Further Compromise**: Analyze logs from CloudTrail, GuardDuty, and VPC Flow Logs to ensure no additional malicious activity is ongoing.
    - **Patch Vulnerabilities**: Apply patches or security updates to any affected resources (e.g., EC2 instances, containers).

    - **Reinforce IAM Policies**: Ensure that IAM roles and policies are correctly configured and follow the principle of least privilege.

11. Recovery

- **Restore Operations**: After the root cause is eradicated, begin restoring affected systems and services to normal operations:
    - **Restore EC2 Instances**: If EC2 instances were compromised, restore them from backups or snapshots.
    - **Re-enable Access**: Restore IAM access to users after confirming that their credentials are secure.
    - **Monitor Post-Recovery**: Keep a heightened monitoring posture for a period after recovery to detect any signs of re-compromise.

12. Post-Incident Review and Improvement

- **Root Cause Analysis**: Conduct a post-incident review to determine the cause of the breach and identify any gaps in security measures.
- **Lessons Learned**: Document findings and make necessary improvements to the security posture.
- **Update Security Controls**: Based on the findings, enhance CloudWatch monitoring, IAM policies, VPC security, and any other relevant security mechanisms.

5. Design a disaster recovery plan, detailing back up schedule recovery point objectives (RPO) and recovery time objectives (RTO) To minimize downtime

Objective. Minimize downtime and ensure business continuity in the event of a disaster.

Scope. All critical systems, applications, and data.

Recovery point objective (RPO):4hours

Recovery time objective (RTO) 2hours

Backup schedule

Data type/backup frequency/backup windows

Critical system/daily/12:00AM -2:00AM

Database /hourly /1:00AM-1:59A

File servers/daily /10:00pm- 12:00AM

Virtual machine /weekly /Saturday, 12:00AM – 2.00AM

Backup method

Full backups. Weekly

Incremental backups. Daily

Differential backups. Hourly for data base

Back up storage

Onsite storage. NAS devices

Off-site storage. cloud storage. AWS s3, Azure blob storage

Recovery procedure

1 Critical system.

Restore from backups within 2hours

Manual intervention required

2. Database

Restore from backups within 1 hour

Automated restore process

3. file servers

Restore from backups within 4hours

Manual intervention required

Disaster recovery steps

Declaration of disaster

Activation of disaster recovery team

Assessment of damage

Recovery of critical systems

Recovery of database

Recovery of file servers

Testing and validation

Return to normal operations

RTO and RPO matrix

System /RTO/RPO

Critical system /2hours/4hours

Database /1hour/1hour

File servers /4hours/24hours

Disaster recovery team

IT manager

System administrators

Database administrators

Network engineers

Training and testing

Quarterly training sessions

Bi-annual disaster recovery drills

Annual review and update of disaster recovery plan

Disaster classification

Disaster level / description /ROT/ROP/

Level1 / minor outage/2hours/4hours

Level2 /major outage/4hours/24hours

Level3/ catastrophic failure /24hours/72hours

Communication plan

Internal communication inform stakeholders of disaster

External communication notify customers, partners, and regulatory bodies as needed

Communication templates. Establish disaster communication templates

Continuous improvement

Regular review and update of disaster recovery plan

Identification and mitigation of single points of failure

Implementation of new technologies and processes to improve recovery times

5b. ensure the plan complies with industry standards and supports business continuity for sages cloud-based services

Industry standards

ISO 27001. Information security management

1SO 22301 business continuity management

NIST cybersecurity framework

Cloud security alliance (CSA) cloud controls matrix

Regulatory requirements

GDPR (general data protection regulations

Business continuity plans

Risk assessment

Identify potential risks

Assess like hood and impact

6.reflect on how the seidea cloud bootcamp has prepared you for this assignment. Identity specific tools, concepts or lesson from the bootcamp that were crucial in completing the project

Answer

Reflecting on the cloud bootcamp experience, I can confidently say that it provided invaluable knowledge and hands on experience that enabled me to excel in this assignment. Here are specific tools, concepts, and lessons from seidea bootcamp that where crucial in completing the project

Concepts. Cloud security and compliance, disaster recovery and business continuity, scalability and high availability, cloud architecture design principles

Essential lessons

Designing scalable cloud architectures

Securing cloud resources using IAM and access controls

Implementing disaster recovery and business continuity plans

Monitoring and logging cloud resources

Developed skills

Cloud architecture design

Cloud security and compliance

Infrastructure as code

Cloud migration and deployment

The bootcamp comprehensive curriculum, hands on experience and real-world project prepared me to design and implement secure cloud architectures

Ensure cloud security and compliance

Implement disaster recovery and business continuity plans

By leveraging these skills and knowledge, I successfully completed the assignment demonstrating my proficiency in cloud computing