Secure Cloud & AI Architecture for a Research Website

Background

Social Research Organization is a (fictitious) nonprofit organization that provides a website for social science researchers to obtain global development statistics. Visitors can look up data points such as the life expectancy of any country in the world over the past 10 years.

Ahmed Idrissi, a researcher at the nonprofit, developed the website. He wanted to share the valuable data he had gathered with other researchers. The dataset is stored in a MySQL database, and a PHP-based web application serves the content. Initially, Ahmed deployed the application on a commercial hosting platform with limited technical and security support.

As the site grew in popularity, two problems emerged:

- Performance degradation due to higher traffic.
- An attempted ransomware attack (unsuccessful, but a warning).

Ahmed migrated the application to AWS, running both the PHP site and MySQL database on a single EC2 instance in a public subnet. This design is insecure and not scalable.

Your team is tasked with redesigning the system using secure, scalable AWS architecture, applying ISO/IEC 27001 risk management, and preparing for the addition of AI functionality.

Current Challenges

1. Database is publicly exposed.
2. No separation of concerns (app + DB on same EC2).
3. Single point of failure.
4. No auto scaling.
5. Limited security controls (IAM, logging, encryption not fully applied).

Solution Requirements

1. Secure Database Hosting

   - ✓ Move MySQL to Amazon RDS in private subnets.
   - ✓ Ensure DB is not publicly accessible.

2. Secure Access to Database

   - ✓ Use security groups so only application servers can connect.
   - ✓ Use IAM roles for DB authentication where possible.

3. Anonymous Access to Website

   - ✓ Researchers access via Application Load Balancer (ALB).
   - ✓ No direct database exposure.

4. Secure Hosting of Web Servers

   - ✓ Website runs on t2.micro EC2 instances in private subnets.
   - ✓ Admins access via bastion host or AWS Systems Manager Session Manager.

5. High Availability

   - ✓ Deploy across multiple Availability Zones.
   - ✓ Use ALB for traffic distribution.

6. Scalability

   - ✓ Implement Auto Scaling Group (ASG) with a launch template.

AWS Well-Architected Framework Principles

- Security

  - ✓ Encrypt data at rest (KMS) and in transit (TLS).
  - ✓ Enable AWS WAF for protection against SQL injection/XSS.
  - ✓ Enable CloudTrail, CloudWatch, and VPC Flow Logs.
  - ✓ Apply least privilege IAM roles.

- Reliability

  - ✓ Multi-AZ deployment.
  - ✓ Enable RDS automated backups & point-in-time recovery.
  - ✓ Auto Scaling policies for demand changes.

- Operational Excellence & Performance

  - ✓ Centralized monitoring in CloudWatch.
  - ✓ Health checks via ALB.
  - ✓ Optionally, caching with Amazon ElastiCache.

Secure Design Principles

- Defense in Depth: VPC isolation, subnets, SGs, IAM, encryption.
- Least Privilege Access: Only web tier → DB communication.
- Separation of Duties: Different roles for admins vs. researchers.
- Monitoring & Logging: Detect anomalies, track DB queries.

DevSecOps Integration

- CI/CD Pipeline (AWS CodePipeline or GitHub/GitLab CI):

  - ✓ Automate builds, tests, deployments.
  - ✓ Integrate SAST/DAST and dependency scanning.
  - ✓ Vulnerability scanning for Docker images if containers are used.

- Infrastructure as Code (IaC):

  - ✓ Use Terraform or CloudFormation for repeatable, auditable deployments.

- Security as Code:

  - ✓ Automated compliance checks (AWS Config, OPA policies).
  - ✓ Penetration test stages in pipeline.

Extension: AI Functionality Scenario

The organization plans to add an AI-powered assistant:

- Researchers can ask natural language questions about datasets.
- AI parses the query, fetches results, and provides insights.

Later, a recommendation engine will suggest trends and allow researchers to upload their own datasets for analysis.

ISO/IEC 27001 Risk Management for the research system

1 Asset Identification

- Identify and list all critical assets in the system, including web application, database, AI components, user data, and DevSecOps pipeline.
- Question: *Which assets are most critical for the confidentiality, integrity, and availability of the system? Justify your selection.*

2. Threat Identification

- Analyze potential threats affecting the entire system: infrastructure, web application, database, AI models, user data, and operations.
- Questions:
    1. *What are the top 10 threats to the system (for each layer) and why are they critical?*
    2. *How could these threats affect the AI functionality specifically?*

3. Risk Assessment

- Assess likelihood and impact of each threat.
- Questions:
    1. *For a given threat, how would you calculate its risk score?*
    2. *Which threats require immediate treatment based on their risk level?*

4. Control Mapping (Annex A)

- Map ISO 27001 Annex A controls to mitigate the identified risks.
- Questions:
    1. *Which Annex A controls would you apply to protect the database? The web application? The AI models?*
    2. *How would you enforce least privilege access across all layers?*
    3. *Which controls ensure compliance for DevSecOps pipelines?*

5. Risk Treatment

- Propose mitigation strategies for each threat.
- Questions:
    1. *How would you secure database and web application access?*

2. *What measures would you implement to prevent AI-specific risks, such as prompt injection, model theft, or hallucinations?*
3. *How would you ensure continuous monitoring and timely incident response for all components?*

6. Monitoring and Review

- Design monitoring and auditing processes for the entire system.
- Questions:
    1. *Which AWS services or tools would you use to monitor infrastructure, application, and AI models?*
    2. *How would you update the risk assessment when new AI features or datasets are added?*

Deliverables

1. AWS architecture diagram (before vs. after redesign).
2. Well-Architected Framework mapping (security, reliability, operations).
3. Security controls applied to web + DB + AI assistant.
4. DevSecOps pipeline design (CI/CD + security testing + IaC) + implementation
5. ISO 27001 risk register including AI risks.
6. Implementation on the AWS Academy plateform.
7. Optional / Recommended: Apply relevant controls from ISO/IEC 27017 (cloud security) and ISO/IEC 27034 (application security) to complement ISO 27001 risk management.