

Secure-by-Design Architecture Review of a Three-Tier Web Application on AWS

Prepared by:

Tayyab Hameed

tayyab.tavv@gmail.com

Date: March 2026

Table of Contents

EXECUTIVE SUMMARY	3
ARCHITECTURE OVERVIEW.....	3
SECURITY REVIEW SCOPE	4
THREAT MODELING	4
SECURITY FINDINGS	5
NIST CYBERSECURITY FRAMEWORK MAPPING	6
RECOMMENDATIONS	7
BUSINESS IMPACT (COST OF DOING NOTHING)	7

Executive Summary

This project presents a secure-by-design architecture review of a typical three-tier web application deployed on Amazon Web Services (AWS). The objective of the assessment is to identify potential security risks within the architecture and recommend improvements aligned with secure infrastructure practices.

The review evaluates the system across multiple security domains including network security, access control, data protection, and monitoring capabilities. Several potential risks were identified, including overly permissive IAM roles, storage exposure risks, and insufficient monitoring controls.

By addressing these issues early in the design phase, organizations can significantly reduce their attack surface and strengthen the overall security posture of the system.

Architecture Overview

The architecture shown above represents a typical three-tier web application deployed on Amazon Web Services (AWS). The design separates the system into different layers to improve scalability, performance, and security.

User traffic first enters through the edge layer where requests are delivered through Amazon CloudFront. AWS WAF is placed in front of the application to filter potentially malicious requests and provide protection against common web attacks.

The application is deployed within an Amazon Virtual Private Cloud (VPC). The Application Load Balancer is placed in a public subnet and is responsible for distributing incoming traffic to the backend compute resources.

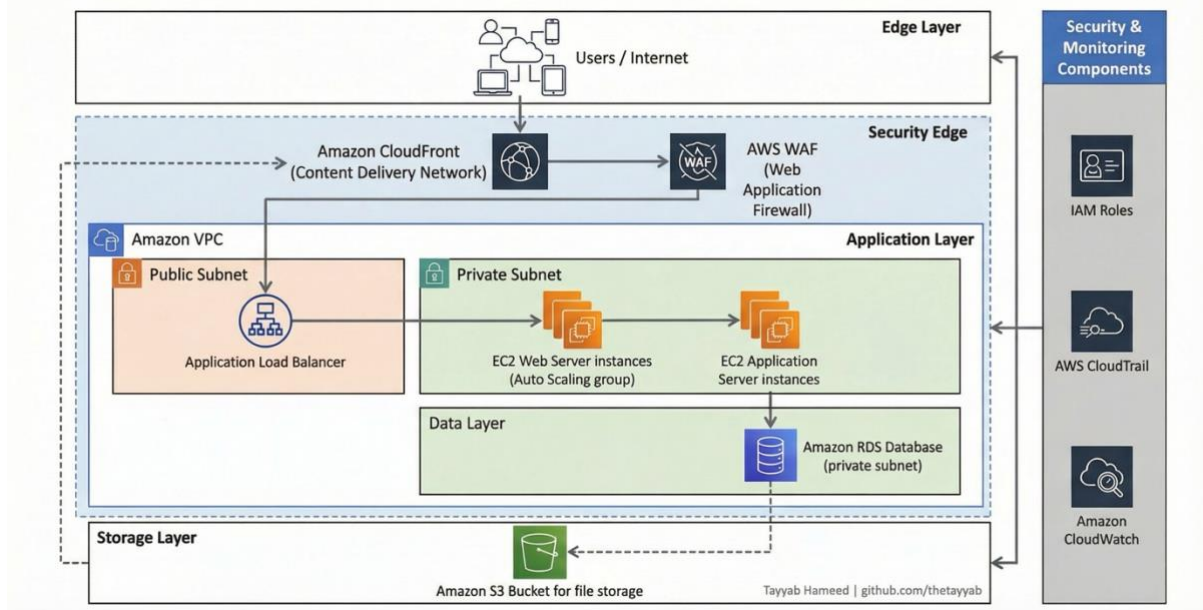
The compute layer consists of EC2 web server instances and EC2 application server instances deployed inside a private subnet. These servers handle application logic and process requests coming from the load balancer.

The data layer is implemented using Amazon RDS, which stores the application's persistent data and is also deployed within the private subnet to reduce direct exposure to the internet.

In addition, Amazon S3 is used as a storage layer for files and application assets. Supporting services such as AWS IAM, AWS CloudTrail, and Amazon CloudWatch provide identity management, auditing, and monitoring capabilities for the environment.

Overall, this layered architecture helps isolate different system components and supports better control over network access, data protection, and system monitoring.

Secure Architecture Review – Three Tier Web Application on AWS



Security Review Scope

This review focuses on evaluating the security of the three-tier web application architecture deployed on AWS. The goal is to identify potential security risks in the system design and highlight areas where additional security controls may be required.

The assessment mainly looks at the infrastructure components shown in the architecture diagram, including the network configuration, edge protection mechanisms, compute resources, data storage, and monitoring services.

The review covers key security areas such as network exposure, access control through IAM roles, protection of application entry points using AWS WAF, data storage within Amazon RDS and S3, and logging through AWS CloudTrail and CloudWatch.

This review focuses on the infrastructure architecture only. It does not include analysis of the application source code, internal development processes, or penetration testing of the deployed system.

Threat Modeling

Threat modeling was performed to identify potential attack surfaces within the architecture and determine where an attacker might attempt to compromise the system.

The main attack surfaces identified from the architecture include:

- **Internet Entry Point**
The application is publicly accessible through the internet. Attackers may attempt web-based attacks such as SQL injection, cross-site scripting (XSS), or denial-of-service attacks targeting the application.
- **Edge Security Layer (CloudFront and AWS WAF)**
CloudFront and AWS WAF provide the first layer of protection for incoming traffic. If WAF rules are not properly configured, malicious requests may bypass filtering controls and reach the backend application servers.
- **Compute Layer (EC2 Instances)**
The EC2 web and application servers handle incoming requests and application logic. Misconfigured instances, outdated software, or overly permissive IAM roles could allow attackers to gain unauthorized access or escalate privileges.
- **Data Layer (Amazon RDS)**
The Amazon RDS database stores application data, which may include sensitive information. Weak access controls or network restrictions could allow unauthorized access to the database.
- **Storage Layer (Amazon S3)**
Amazon S3 is used for file storage and application assets. Misconfigured bucket permissions could expose stored files publicly and lead to unintended data disclosure.
- **Identity and Access Management (IAM Roles)**
Improperly configured IAM roles may grant excessive permissions to services or users, increasing the risk of privilege escalation if a resource is compromised.
- **Logging and Monitoring Services**
CloudTrail and CloudWatch are responsible for logging and monitoring activity within the environment. If logging is not properly configured or monitored, malicious activity may go undetected.

Security Findings

Based on the threat modeling analysis, several potential security risks were identified within the architecture. These findings highlight areas where misconfiguration or weak controls could increase the risk of unauthorized access, data exposure, or service disruption.

The key security findings are outlined below.

- **Overly Permissive IAM Roles**
If IAM roles assigned to EC2 instances are not properly restricted, they may grant excessive permissions to AWS resources. This could allow an attacker who compromises an instance to access other services such as S3 buckets or databases.
- **Potential Misconfiguration of AWS WAF**
AWS WAF is responsible for filtering malicious traffic before it reaches the application. If WAF rules are not properly configured or maintained, attackers may bypass filtering controls and exploit vulnerabilities in the application.

- Public Exposure Risk for Amazon S3**
 Amazon S3 is used to store application files and assets. Incorrect bucket permissions may accidentally expose files to the public, potentially leading to sensitive data leakage.
- Weak Network Segmentation**
 Although the architecture separates resources into public and private subnets, poorly configured security groups could allow unnecessary communication between components. This may increase the risk of lateral movement if one server becomes compromised.
- Insufficient Logging and Monitoring**
 While AWS CloudTrail and CloudWatch are present in the architecture, improper configuration or lack of active monitoring may prevent timely detection of suspicious activity.
- Lack of Encryption for Stored Data**
 If encryption is not enabled for services such as Amazon RDS or S3, stored data could be exposed if storage resources are accessed by unauthorized entities.
- Unpatched EC2 Instances**
 EC2 instances running outdated operating systems or software may contain known vulnerabilities that attackers can exploit to gain access to the system.

NIST Cybersecurity Framework Mapping

The identified security findings were mapped to the NIST Cybersecurity Framework (NIST CSF) to understand how the risks relate to recognized cybersecurity practices. This mapping helps demonstrate how security controls can be applied to reduce the likelihood and impact of potential threats.

Security Finding	NIST CSF Function	Description
Overly Permissive IAM Roles	Protect (PR.AC – Access Control)	Proper identity and access management is required to ensure that users and services only have the permissions necessary to perform their functions.
Misconfigured AWS WAF Rules	Protect (PR.IP – Protective Technology)	Web application firewalls should be configured to filter malicious traffic and protect the application from common web attacks.
Public Exposure of Amazon S3	Protect (PR.DS – Data Security)	Data stored in cloud storage should be properly protected through access controls and secure configurations.
Weak Network Segmentation	Protect (PR.AC – Access Control)	Network segmentation and security group configurations should restrict communication between system components.
Insufficient Logging and Monitoring	Detect (DE.CM – Security Monitoring)	Continuous monitoring and logging are necessary to detect suspicious activity and potential security incidents.

Lack of Data Encryption	Protect (PR.DS – Data Security)	Sensitive data should be encrypted both at rest and in transit to protect confidentiality.
Unpatched EC2 Instances	Protect (PR.IP – Protective Technology)	Systems should be regularly updated and patched to reduce exposure to known vulnerabilities.

Recommendations

Based on the identified security findings, several security improvements are recommended to strengthen the overall architecture and reduce potential attack risks.

- Apply Least Privilege to IAM Roles**
 IAM roles assigned to EC2 instances and services should follow the principle of least privilege. Permissions should only allow the actions required for the service to function, reducing the risk of privilege escalation if a resource is compromised.
- Strengthen AWS WAF Configuration**
 AWS WAF rules should be regularly updated to block common web application attacks such as SQL injection, cross-site scripting (XSS), and malicious bots. Managed rule sets can also be enabled to improve protection.
- Secure Amazon S3 Bucket Permissions**
 S3 buckets should be configured with private access by default. Public access settings should be restricted, and bucket policies should be reviewed to ensure sensitive files are not exposed.
- Improve Network Segmentation**
 Security groups and network access controls should restrict communication between components of the architecture. Only required ports and services should be allowed between the application servers and the database layer.
- Enable Comprehensive Logging and Monitoring**
 CloudTrail and CloudWatch should be fully configured to monitor system activity and detect suspicious behaviour. Alerts should be enabled for unusual login activity, permission changes, or abnormal traffic patterns.
- Enable Data Encryption**
 Encryption should be enabled for Amazon RDS and Amazon S3 to protect stored data. HTTPS should also be enforced to ensure secure communication between users and the application.
- Implement Patch Management for EC2 Instances**
 EC2 instances should be regularly updated and patched to ensure that known vulnerabilities are addressed. Automated patch management tools can be used to maintain system security.

Business Impact (Cost of Doing Nothing)

If the identified security risks are not addressed, the system may become vulnerable to unauthorized access, data exposure, and service disruption. Misconfigured access controls or publicly exposed storage resources could potentially lead to sensitive data being accessed by attackers.

In addition, insufficient monitoring and logging may delay the detection of security incidents, allowing malicious activity to remain unnoticed for extended periods. This could increase the impact of a potential breach and make incident response more difficult.

Addressing these security risks during the design stage significantly reduces the likelihood of security incidents and improves the overall security posture of the system. Implementing security controls early in the architecture helps minimize operational disruption, financial loss, and reputational damage in the future.