TASK 2: WAF ASSESSMENT TABLE

Pillar	Observation	Improvement	Recommendation	Supporting AWS Service
Operational Excellence	Two-tier architecture separates concerns between presentation and data layers, enabling independent updates	Lack of automation in operational procedures including deployment pipelines from an on-premises environment.	Implement IaC to establish CI/CD pipelines for automated testing and deployment of both application and infrastructure changes	AWS CloudFormation
Security	Application supports TLS/SSL encryption for data in transit between users and web tier	No encryption for data at rest in database and weak authentication module without MFA	Implement encryption at rest for database and implement MFA and principle of Least Privilege to users and groups.	AWS Secrets Manager
Reliability	Separation of web and database tiers provides some faulty isolation.	Single point of failure, no backup strategy for database and no automatic scaling of web application to meet demand spikes	Deploy across multiple Availability Zones. Implement Auto Scaling for web tiers to handle variable load and implement automated backups and failovers.	Amazon RDS multi-AZ
Performance Efficiency	Two-tier architecture allows independent scaling of presentation and data layers based on demand patterns	High network latency for users in geographically distant regions. Static provisioning of workloads leads to	Use Content Delivery Network for caching contents to global users and consider serverless or container options for web tier.	Amazon CloudFront

		over/under provisioning		
Cost Optimization	Predictable capital expenditure model with fixed costs for owned infrastructure, and no variable data transfer or API call charges.	Infrastructure sized for peak capacity runs 24/7, wasting resources during off-hours and low-traffic periods	Implement dynamic scaling to match resource provisioning to actual demand, eliminating waste during off-hours, and use Pay-Per-Use pricing to avoid overprovisioning	AWS Cost Explorer

TASK 3: CAF READINESS SUMMARY

1. **<u>Business perspective:</u>**
   The Business perspective is medium. The organization demonstrates basic readiness by recognizing the need for AWS migration aligned with best practices but lacks strategic cloud maturity. There's no evidence of defined business outcome metrics (revenue impact, customer experience gains) or innovation roadmap beyond technical migration. Cloud strategy appears reactive rather than proactive, with limited portfolio.
   <u>Key Actions:</u>
   - Define Business Outcomes: Establish measurable KPIs linking migration to business value (e.g., reduce customer latency by 40%, improve application availability to 99.9%, enable entry into new geographic markets
   - Develop Cloud Business Case: Quantify benefits including cost savings from eliminating on-premises infrastructure, revenue gains from improved performance, and risk reduction from enhanced reliability
   - Create Innovation Roadmap: Beyond migration, identify cloud-native capabilities to pursue (e.g., analytics on customer behavior data, ML-driven personalization, API economy opportunities).

- Establish Executive Sponsorship: Secure C-level champions like Chief Technology Officer to drive transformation and align cloud investments with strategic intent.
- Portfolio Management: Use the 7 R's migration strategy framework to rationalize the application portfolio and prioritize this two-tier application appropriately

2. **People perspective:**

   The People perspective is low. The organization faces significant people's challenges. Current staff likely possess traditional infrastructure skills but lack cloud fluency in AWS services and cloud-native patterns. There's no evidence of DevOps culture, agile methodologies, or change management framework. Teams may resist transformation, fearing job displacement or increased complexity without transformational leadership actively championing the migration

   Key Actions:

   - Skills Assessment and Training Plan: Conduct gap analysis of current vs. required cloud skills. Enroll teams in AWS Training and Certification programs (Solutions Architect, DevOps, SysOps). Leverage AWS Skill Builder for hands-on labs
   - Establish Cloud Center of Excellence: Create cross-functional team of cloud champions to develop best practices, provide internal consulting, and accelerate learning across the organization
   - Workforce Transformation Strategy: Define new roles (Cloud Architect, DevOps Engineer, SRE) and career paths. Use partners/managed services to augment capabilities during transition.
   - Foster Agile and DevOps Culture: Introduce agile ceremonies, continuous improvement mindset, and collaborative practices between development and operations teams to support cloud operating models.

3. **Governance perspective:**

   Governance readiness is low. The organization likely operates with traditional ITIL processes unsuited for cloud agility. There's no Cloud Financial Management capability to control variable spend, no tagging strategy for cost allocation, and risk management focuses on physical infrastructure rather than cloud-specific concerns (shared responsibility, data residency, API security).

   Key Actions:

- Establish Multi-Account Strategy: Implement AWS Organizations with separate accounts for production, development, and testing using AWS Control Tower to enforce governance guardrails and security baselines.
- Implement Cloud Financial Management: Deploy AWS Budgets with alerts, enable Cost Explorer, and establish cost allocation tags
- Define Governance Framework: Create cloud governance policies covering security baselines, compliance requirements, resource provisioning standards, and change approval of workflows adapted for cloud agility.
- Risk Management: Update risk register with cloud-specific risks (vendor lock-in, data sovereignty, skills shortage). Conduct AWS Well-Architected Review to identify architectural risks early
- Migration Program Structure: Assign program manager, define work streams (application, database, security, networking), and implement agile project tracking with regular sprint reviews and retrospectives

4. **Platform perspective:**

   Platform readiness is moderate. The two-tier architecture provides a foundation but lacks cloud-native characteristics. There's no Infrastructure as Code (IaC), CI/CD pipelines, or modern application development practices (containers, serverless). Data architecture is simplistic without caching or reading replica strategies.

   Key Actions:

   - Design Target Architecture**:** Create AWS reference architecture for two-tier application using VPC with public/private subnets across multiple AZs, Application Load Balancer, Auto Scaling Groups for web tier, and Amazon RDS Multi-AZ for database
   - Adopt Infrastructure as Code**:** Implement AWS CloudFormation or Terraform templates to define all infrastructure declaratively. Store templates in version control. This enables repeatability and disaster recovery
   - Build Landing Zone: Use AWS Control Tower or Landing Zone Accelerator to establish foundational platform with networking, security, and logging standards pre-configured
   - Database Migration Strategy: Use AWS Database Migration Service (DMS) for initial migration with minimal downtime. Evaluate Amazon Aurora for MySQL/PostgreSQL workloads to gain cloud-native database benefits
   - Establish Service Catalog**:** Create AWS Service Catalog with pre-approved, standardized infrastructure patterns that development teams can self-service provision

5. **Security perspective:**

   Security readiness is low for the Cloud. While basic on-premises security exists, the organization lacks cloud-specific security governance and understanding of AWS Shared Responsibility Model. Critical gaps include no encryption at rest, missing WAF/IDS/IPS, weak authentication without MFA, and no cloud-native threat detection or incident response capabilities.

   Key Actions:

   - Implement Defense in Depth: Deploy layered security: AWS WAF on ALB (application layer), Network ACLs and Security Groups (network layer), encryption at rest with AWS KMS (data layer). Enable VPC Flow Logs and AWS GuardDuty for threat detection
   - Identity and Access Management**:** Federate on-premises Active Directory with AWS IAM Identity Center (formerly SSO). Implement least-privilege IAM roles and policies. Enforce MFA for all human users. Use IAM roles for EC2 instances instead of access keys
   - Data Protection Strategy**:** Enable encryption at rest for RDS using KMS customer-managed keys and Use AWS Secrets Manager for database credentials and API keys. Enable automated backups with encryption.
   - Compliance and Auditing**:** Enable AWS CloudTrail for all API activity logging. Use AWS Config for continuous compliance monitoring against security baselines. Implement AWS Security Hub as centralized security dashboard

6. **Operations perspective:**

   Operations readiness is low. The organization likely uses traditional monitoring tools and reactive incident management unsuitable for distributed cloud. There's no observability platform, automated remediation, or AIOps capability. Issues like single-AZ deployment and no auto-scaling indicate poor availability management. Change processes probably involve lengthy approval boards incompatible with CI/CD velocity

   Key Actions:

   - Deploy Amazon CloudWatch for centralized metrics, logs, and dashboards with CloudWatch Application Insights for automated problem detection. Implement AWS X-Ray for distributed tracing across application tiers
   - Define Service Level Indicators (page load time, query latency), Service Level Objectives (95th percentile response times), and monitor against targets

- Automate incident response with CloudWatch alarms triggering Lambda remediation and Systems Manager Automation.
- Implement Auto Scaling with target tracking policies and schedule resources to shut down during non-business hours.
- Build disaster recovery plan defining RTO/RPO with automated backups using AWS Backup and quarterly failover testing

A BRIEF REFLECTION ON WHAT WAS LEARNED

Completing these tasks reinforced that successful cloud migration requires more than technical knowledge—it demands systematic evaluation across multiple dimensions. The AWS Well-Architected Framework provided a structured lens to assess the current architecture's deficiencies (single-AZ deployment, missing backups, weak security) and design improvements addressing operational excellence, security, reliability, performance, and cost optimization simultaneously.

The Cloud Adoption Framework revealed that technology transformation alone is insufficient. People's readiness (skills, culture, change management), governance mechanisms (cost controls, risk management), and organizational alignment are equally critical. The interconnection between frameworks became evident—for example, implementing CI/CD pipelines (Platform perspective) enables operational excellence (a WAF pillar), while Cloud Financial Management (Governance perspective) supports cost optimization (a WAF pillar).

This exercise demonstrated that AWS best practices aren't isolated checklists, but integrated approaches requiring cross-functional collaboration. The migration from on-premises to cloud represents organizational transformation, not just infrastructure relocation. Real success demands addressing technical architecture, people's capabilities, governance frameworks, and business outcomes holistically.