

Proposal for Acme Brick's Global Digital Expansion and AWS Migration

Prepared by: Steven Meyer - AWS Principal Solutions Architect

Executive Summary

As Acme Brick embarks on its global digital expansion, migrating to Amazon Web Services (AWS) presents a strategic opportunity to enhance scalability, security, and operational efficiency. Leveraging the AWS Well-Architected Framework, this proposal outlines a comprehensive migration strategy that integrates DevSecOps practices and addresses Acme Brick's unique business and technical requirements.

Table of Contents

1. Introduction
2. Assumptions
3. Objectives and Goals
4. Migration Strategy
5. Proposed AWS Architecture
6. Alignment with AWS Well-Architected Framework
7. Project Requirements
8. Level of Effort and Resource Planning
9. Detailed Project Plan
10. Specific Solution Components
11. Conclusion and Next Steps

1. Introduction

As an AWS Principal Solutions Architect with extensive experience in cloud migrations and global infrastructure design, I present this proposal to guide Acme Brick through a successful transition to AWS. The proposed approach ensures that the migration aligns with best practices, maximizes return on investment, and positions Acme Brick for sustained global growth.

2. Assumptions:

1. **Current Infrastructure:** Acme Brick currently operates a mix of monolithic and microservices-based applications hosted on-premises.
2. **Global Expansion:** The company plans to serve customers globally, requiring low-latency access and compliance with international regulations like GDPR.
3. **Security Requirements:** High emphasis on security and compliance; need to integrate security into all stages of development and operations (DevSecOps).
4. **Application Landscape:** Some applications are suitable for cloud-native transformation, while others are better suited for lift-and-shift due to legacy dependencies.
5. **Timeline:** The migration should be completed within six months to align with business expansion goals.
6. **Budget Constraints:** The solution should be cost-effective without compromising on performance or security.

3. Objectives and Goals

- **Global Scalability:** Deploy applications and services globally to provide low-latency access to customers worldwide.
- **Enhanced Security:** Integrate security at every layer, adopting DevSecOps practices to proactively manage vulnerabilities.
- **Operational Excellence:** Streamline operations through automation and adoption of cloud-native services.
- **Cost Optimization:** Utilize AWS services and pricing models to achieve cost-efficiency without compromising performance.
- **Compliance and Governance:** Ensure adherence to international regulations such as GDPR and industry-specific standards.

4. Migration Strategy

Hybrid Migration Approach

1. Assess and Prioritize Applications:

- **Category A (Lift-and-Shift):** Legacy applications that require minimal modification. Migrated to AWS using Amazon EC2.
- **Category B (Re-Platform):** Applications that can benefit from cloud optimization without significant code changes. Transitioned to managed services like Amazon RDS.
- **Category C (Re-Architect):** Applications suited for cloud-native transformation. Refactored to leverage services like AWS Lambda, Amazon ECS, and AWS Fargate.

2. Phased Migration Plan:

- **Phase 1:** Migrate non-critical applications to build AWS proficiency.
- **Phase 2:** Transition critical applications, ensuring minimal disruption.
- **Phase 3:** Optimize and refactor applications for cloud-native benefits.

3. DevSecOps Integration:

- Implement security practices throughout the CI/CD pipeline.
- Automate security testing and compliance checks.

5. Proposed AWS Architecture

Compute Layer

- **Amazon EC2:** For lift-and-shift workloads, providing full control over the operating environment.
- **AWS Lambda:** For serverless execution of event-driven applications.
- **Amazon ECS:** For containerized applications, offering scalability and reduced management overhead.

Storage and Data Management

- **Amazon S3:** Scalable object storage with lifecycle policies and cross-region replication.
- **Amazon EFS:** Scalable file storage accessible across multiple instances.
- **Amazon EBS:** High-performance block storage for EC2 instances.
- **Amazon RDS (Aurora, MySQL, PostgreSQL):** Managed relational databases with high availability and read replicas.

- **Amazon DynamoDB:** Fully managed NoSQL database for high-throughput workloads.
- **Amazon Redshift:** Data warehousing for analytics.

Networking and Content Delivery

- **Amazon VPC:** Isolated network environment with granular control.
- **AWS Transit Gateway:** Simplifies network architecture by connecting VPCs and on-premises networks.
- **AWS Direct Connect:** Dedicated network connection for consistent throughput and low latency.
- **Amazon CloudFront:** Global CDN for content delivery with edge caching.
- **AWS Global Accelerator:** Enhances performance and availability using the AWS global network.

Security and Compliance

- **AWS IAM:** Centralized identity management with fine-grained access control.
- **AWS Organizations:** Policy-based management of multiple AWS accounts.
- **AWS KMS:** Centralized key management and encryption.
- **AWS WAF and AWS Shield Advanced:** Protection against web exploits and DDoS attacks.
- **AWS Security Hub:** Unified view of security alerts and compliance status.
- **Amazon GuardDuty:** Intelligent threat detection and continuous monitoring.

Management and Governance

- **Terraform:** Infrastructure as Code for consistent and repeatable deployments.
- **AWS Config:** Continuous assessment of resource configurations.
- **AWS Systems Manager:** Unified interface for operational tasks across AWS resources.

Monitoring and Logging

- **Amazon CloudWatch:** Performance monitoring and operational data.
- **AWS CloudTrail:** Governance, compliance, and operational auditing.
- **AWS X-Ray:** Distributed tracing for debugging and analyzing applications.

CI/CD Pipeline

- **GitHub:** Secure, highly scalable source control service.
- **AWS CodeBuild:** Continuous integration service that compiles source code, runs tests, and produces packages.
- **AWS CodeDeploy:** Automated code deployments to any instance.
- **AWS CodePipeline:** Continuous delivery service for fast and reliable application updates.

6. Alignment with AWS Well-Architected Framework

Operational Excellence

- **Best Practices:**
 - Implement Infrastructure as Code using Terraform.
 - Automate change management and deployment pipelines.
 - Use AWS Systems Manager for operational tasks and patch management.

- **Outcomes:**
 - Improved deployment speed and consistency.
 - Reduced risk of human error.
 - Enhanced visibility into operational health.

Security

- **Best Practices:**
 - Apply the principle of least privilege with IAM policies.
 - Encrypt data at rest and in transit using AWS KMS and SSL/TLS.
 - Implement automated compliance checks using AWS Config Rules.
 - Use Amazon GuardDuty and AWS Security Hub for continuous monitoring.
- **Outcomes:**
 - Robust security posture with proactive threat detection.
 - Compliance with international regulations (e.g., GDPR).
 - Reduced vulnerability exposure through continuous monitoring.

Reliability

- **Best Practices:**
 - Design for failure with multi-AZ and multi-region deployments.
 - Use auto-scaling and load balancing for high availability.
 - Implement disaster recovery strategies with defined RTO/RPO.
- **Outcomes:**
 - Increased system uptime and resilience.
 - Ability to withstand and quickly recover from failures.
 - Consistent performance during traffic spikes.

Performance Efficiency

- **Best Practices:**
 - Right-size resources using AWS Compute Optimizer recommendations.
 - Leverage serverless architectures where appropriate.
 - Utilize caching layers with Amazon ElastiCache.
 - Monitor performance metrics and adjust as needed.
- **Outcomes:**
 - Optimized application performance.
 - Scalability to meet global demand.
 - Efficient use of computing resources.

Cost Optimization

- **Best Practices:**
 - Implement cost allocation tags for resource tracking.
 - Use AWS Cost Explorer and AWS Budgets to monitor expenses.
 - Opt for Reserved Instances and Savings Plans for predictable workloads.
 - Turn off or scale down non-production environments when not in use.
- **Outcomes:**
 - Controlled and predictable cloud spending.
 - Maximized ROI through efficient resource utilization.
 - Visibility into cost drivers and opportunities for savings.

7. Project Requirements

Technical Requirements

- **Application Compatibility:** Ensure applications are compatible with AWS services.
- **Data Migration:** Secure and efficient transfer of data with minimal downtime.
- **Network Configuration:** Design VPCs, subnets, and routing to meet performance and security needs.
- **Security Compliance:** Adherence to GDPR, ISO 27001, and other relevant standards.
- **Disaster Recovery:** Implement strategies to meet defined RTO and RPO objectives.
- **Monitoring and Alerting:** Set up comprehensive monitoring for all services.

Operational Requirements

- **Process Development:** Establish processes for cloud governance and operational management.
- **Documentation:** Maintain detailed documentation for all configurations and procedures.
- **Change Management:** Implement controls for managing changes to the environment.

8. Level of Effort and Resource Planning

Given the scope and complexity of the project, the following is an estimated level of effort and resource allocation:

- **Project Duration: Approximately 24 weeks (6 months).**

Resource Plan

- **AWS Principal Solutions Architect (1)**
 - Leads architecture design, migration strategy, and oversees implementation.
- **DevOps Engineers (2)**
 - Focus on CI/CD pipeline development, automation, and infrastructure setup.
- **Security Engineer (1)**
 - Specializes in security implementations and compliance.
- **Application Developer (2)**
 - Handles application refactoring and optimization.
- **Database Administrator (1)**
 - Manages data migration and database optimization.
- **Network Engineer (1)**
 - Designs and configures networking components.
- **QA/Test Engineer (1)**
 - Conducts testing and validation.
- **Training Specialist (1)**
 - Provides team training and knowledge transfer.

9. Detailed Project Plan

Phase 1: Discovery and Assessment (Weeks 1-4)

- **Activities:**
 - Conduct a comprehensive inventory of existing applications and infrastructure.
 - Assess application dependencies and cloud readiness.

- Identify regulatory compliance requirements.
- Perform a Well-Architected Review to establish a baseline.
- **Deliverables:**
 - Application and infrastructure inventory.
 - Cloud readiness report.
 - Risk assessment document.

Phase 2: Planning and Design (Weeks 5-7)

- **Activities:**
 - Develop a detailed migration plan and timeline.
 - Design AWS architecture adhering to the Well-Architected Framework.
 - Define security and compliance strategies.
 - Plan CI/CD pipeline architecture.
- **Deliverables:**
 - Migration strategy document.
 - AWS architectural diagrams.
 - Security and compliance plan.
 - CI/CD pipeline design.

Phase 3: Infrastructure Setup (Weeks 8-11)

- **Activities:**
 - Set up AWS accounts using AWS Organizations with appropriate SCPs (Service Control Policies).
 - Configure VPCs, subnets, routing tables, and security groups.
 - Implement IAM roles and policies.
 - Establish logging and monitoring infrastructure.
- **Deliverables:**
 - Configured AWS environments.
 - Network and security configurations.
 - Monitoring and logging setup documentation.

Phase 4: Application Migration and Refactoring (Weeks 12-19)

- **Activities:**
 - Migrate lift-and-shift applications to EC2 instances.
 - Refactor suitable applications to use AWS Lambda, ECS.
 - Migrate databases using AWS DMS with minimal downtime.
 - Implement caching and performance optimizations.
- **Deliverables:**
 - Migrated and refactored applications.
 - Updated application documentation.
 - Performance testing reports.

Phase 5: CI/CD Pipeline Implementation (Weeks 12-15)

- **Activities:**
 - Set up Github repositories for source control.
 - Configure CodeBuild projects for continuous integration.
 - Implement CodeDeploy for automated deployments.

- Integrate security testing tools into the pipeline.
- **Deliverables:**
 - Operational CI/CD pipelines.
 - Documentation on pipeline processes.
 - Automated testing and deployment scripts.

Phase 6: Security Implementation (Weeks 12-17)

- **Activities:**
 - Implement AWS WAF rules and Shield Advanced protections.
 - Configure AWS KMS for key management and encryption.
 - Set up AWS Security Hub and GuardDuty.
 - Automate compliance checks with AWS Config Rules.
- **Deliverables:**
 - Security configurations and policies.
 - Compliance reports.
 - Security testing results.

Phase 7: Testing and Validation (Weeks 18-21)

- **Activities:**
 - Perform functional and performance testing.
 - Conduct security assessments and penetration testing.
 - Execute disaster recovery drills.
 - Complete a Well-Architected Review for validation.
- **Deliverables:**
 - Test reports and remediation plans.
 - Updated Well-Architected Review.
 - User acceptance sign-off.

Phase 8: Training and Knowledge Transfer (Weeks 20-21)

- **Activities:**
 - Conduct training sessions on AWS services, DevSecOps, and operational processes.
 - Provide comprehensive documentation and runbooks.
- **Deliverables:**
 - Training materials.
 - Knowledge transfer sessions.
 - Updated documentation repository.

Phase 9: Go-Live and Post-Migration Support (Weeks 22-24)

- **Activities:**
 - Execute production cutover.
 - Monitor systems for stability and performance.
 - Provide hypercare support during the transition.
 - Identify opportunities for further optimization.
- **Deliverables:**
 - Successful production deployment.
 - Post-migration support reports.

- Optimization roadmap.

10. Specific Solution Components

CI/CD Pipeline Implementation

- **Version Control:**
 - Github repositories with branching strategies.
- **Continuous Integration:**
 - AWS CodeBuild projects with buildspec files defining build steps.
 - Integration of code quality tools like SonarQube or AWS CodeGuru Reviewer.
- **Security Integration:**
 - Use AWS CodePipeline to orchestrate steps, including security scans.
 - Integrate Amazon Inspector for vulnerability assessments.
 - Employ Snyk or Black Duck for dependency vulnerability scanning.
- **Continuous Deployment:**
 - AWS CodeDeploy with Blue/Green deployments for zero-downtime releases.
 - Implement deployment validation tests and automatic rollback mechanisms.
- **Automated Testing:**
 - Unit tests executed during the build phase.
 - Integration and end-to-end tests using frameworks like JUnit or pytest.
 - Load testing with AWS Distributed Load Testing solution.

Infrastructure as Code (IaC)

- **Terraform:**
 - Modules for provisioning all AWS resources.
 - .tfvars files for different environments (dev, test, prod).

Security Measures

- **Identity and Access Management:**
 - Implement fine-grained IAM policies.
 - Use AWS Single Sign-On (SSO) for centralized access.
- **Data Protection:**
 - Encrypt S3 buckets using SSE-S3 or SSE-KMS.
 - Enable encryption for EBS volumes and RDS instances.
- **Network Security:**
 - Configure security groups and network ACLs.
 - Use AWS Network Firewall for advanced network protections.
- **Compliance Automation:**
 - Use AWS Audit Manager to automate evidence collection.
 - Regularly run AWS Config Conformance Packs.

Monitoring and Alerting

- **Amazon CloudWatch:**
 - Custom dashboards for key metrics.
 - Set alarms for thresholds and anomalies.
- **AWS CloudTrail:**
 - Enable multi-region and global services logging.

- Integrate with Amazon S3 Glacier for long-term log storage.
- **AWS X-Ray:**
 - Instrument applications for tracing and debugging.
- **Incident Response:**
 - Set up AWS Systems Manager Incident Manager.

Global Scalability and Performance

- **Content Delivery:**
 - Amazon CloudFront with custom SSL certificates.
- **Latency Optimization:**
 - Use AWS Global Accelerator to route traffic to optimal endpoints.
- **Auto Scaling:**
 - Configure Auto Scaling Groups with predictive scaling policies.
- **Database Scaling:**
 - Use Amazon Aurora Global Database for cross-region replication.

Cost Management

- **AWS Cost Explorer and Budgets:**
 - Set up cost anomaly detection.
- **Resource Optimization:**
 - Regularly review resource utilization.
 - Implement lifecycle policies for S3 and EBS snapshots.
- **Reserved Instances and Savings Plans:**
 - Analyze usage patterns to commit to long-term discounts.

11. Conclusion and Next Steps

By adopting this comprehensive migration strategy, Acme Brick will achieve a secure, scalable, and cost-optimized infrastructure on AWS, fully equipped to support its global digital expansion. The integration of DevSecOps practices and alignment with the AWS Well-Architected Framework ensures that the solution is robust, efficient, and future-proof.

Immediate Next Steps:

1. **Project Kickoff:**
 - Schedule a meeting to align stakeholders and confirm project scope and timelines.
2. **Team Mobilization:**
 - Assemble the project team, including internal staff and any required external experts.
3. **AWS Account Setup:**
 - Establish AWS accounts and organizational structure.
4. **Discovery Phase Initiation:**
 - Begin detailed discovery and assessment activities.

Note: This proposal is designed to be flexible and may be adjusted based on further discussions and discoveries during the initial phases of the project. I am committed to working closely with Acme Brick's team to ensure the success of this migration and the long-term achievement of your business objectives.

Appendix A: Detailed Resource Breakdown

This appendix provides a comprehensive breakdown of the resources required for the migration project, detailing roles, responsibilities, estimated time commitments, and necessary skill sets, based on the optimized resource allocation.

1. AWS Principal Solutions Architect (1)

- **Role:** Project Lead
- **Responsibilities:**
 - Develop and validate the migration strategy.
 - Design AWS architecture aligned with the AWS Well-Architected Framework.
 - Provide technical leadership and mentorship to the team.
 - Conduct AWS Well-Architected Reviews at key project milestones.
 - Liaise with Acme Brick's stakeholders and executive leadership.
 - Oversee risk management and mitigation strategies.
- **Skills Required:**
 - Extensive experience with AWS services and cloud architecture.
 - Expertise in the AWS Well-Architected Framework.
 - Strong leadership and project management skills.
 - Excellent communication and stakeholder management abilities.
- **Estimated Time Commitment:** Approximately 30 hours/week throughout the project duration (24 weeks).
- **Total Hours: 720 hours**

2. DevOps Engineers (2)

- **Role:** Infrastructure Setup and CI/CD Pipeline Development
- **Responsibilities:**
 - Implement Infrastructure as Code (IaC) using Terraform.
 - Develop and maintain CI/CD pipelines with AWS CodePipeline, CodeBuild, and CodeDeploy.
 - Automate deployment processes and operational tasks.
 - Configure and manage AWS services (EC2, ECS, Lambda).
 - Monitor system performance and optimize resource utilization.
- **Skills Required:**
 - Proficiency in DevOps tools and practices.
 - Experience with AWS services and automation.
 - Scripting skills (Python, Bash).
 - Knowledge of containerization technologies (Docker, Kubernetes).
- **Estimated Time Commitment:** Full-time during Phases 3 to 9 (Weeks 8–24).
- **Total Hours per Engineer: 560 hours**
- **Combined Total Hours: 1,120 hours**

3. Security Engineer (1)

- **Role:** Security Implementation and Compliance
- **Responsibilities:**
 - Design and implement security measures across AWS environments.
 - Configure AWS security services (IAM, KMS, WAF, Shield, GuardDuty).
 - Ensure compliance with GDPR, ISO 27001, and other regulations.

- Integrate security testing into the CI/CD pipeline.
- Conduct security assessments and penetration testing.
- Monitor security alerts and respond to incidents.
- **Skills Required:**
 - In-depth knowledge of AWS security services and best practices.
 - Experience with security frameworks and compliance standards.
 - Proficiency in security tools and methodologies.
 - Analytical and problem-solving skills.
- **Estimated Time Commitment:** Approximately 20 hours/week during Phases 2 to 9 (Weeks 5–24).
- **Total Hours: 400 hours**

4. Application Developers (2)

- **Role:** Application Refactoring and Optimization
- **Responsibilities:**
 - Assess applications for cloud readiness and suitability for refactoring.
 - Refactor applications to leverage AWS managed services (Lambda, ECS).
 - Optimize code for performance, scalability, and cost-efficiency.
 - Collaborate with DevOps engineers to integrate applications into CI/CD pipelines.
 - Perform unit testing and code reviews.
- **Skills Required:**
 - Experience in application development and cloud-native architectures.
 - Proficiency in programming languages used in Acme Brick's applications.
 - Knowledge of AWS services and serverless computing.
 - Familiarity with DevOps practices.
- **Estimated Time Commitment:** Full-time during Phases 4 to 7 (Weeks 12–21).
- **Total Hours per Developer: 300 hours**
- **Combined Total Hours: 600 hours**

5. Database Administrator (1)

- **Role:** Data Migration and Database Optimization
- **Responsibilities:**
 - Plan and execute data migration using AWS Database Migration Service (DMS).
 - Optimize database configurations for AWS (RDS, DynamoDB).
 - Implement backup, recovery, and disaster recovery strategies.
 - Ensure data integrity and compliance with data protection regulations.
 - Monitor database performance and make necessary adjustments.
- **Skills Required:**
 - Expertise in database management systems (SQL, NoSQL).
 - Experience with AWS database services.
 - Knowledge of data migration techniques and tools.
 - Attention to detail and analytical skills.
- **Estimated Time Commitment:** Approximately 16 hours/week during Phases 4 to 7 (Weeks 12–21).
- **Total Hours: 160 hours**

6. Network Engineer (1)

- **Role:** Network Design and Configuration
- **Responsibilities:**
 - Design VPCs, subnets, routing tables, and security groups.
 - Configure AWS Direct Connect or VPN for secure connectivity.
 - Implement AWS Transit Gateway for network simplification.
 - Optimize network performance and latency for global access.
 - Monitor network health and troubleshoot issues.
- **Skills Required:**
 - Strong understanding of networking concepts and protocols.
 - Experience with AWS networking services.
 - Problem-solving skills and attention to detail.
- **Estimated Time Commitment:** Approximately 10 hours/week during Phases 3 and 4 (Weeks 8–19).
- **Total Hours: 120 hours**

7. QA/Test Engineer (1)

- **Role:** Testing and Validation
- **Responsibilities:**
 - Develop test plans and test cases for functional, performance, and security testing.
 - Execute tests and document results.
 - Identify defects and coordinate with developers for resolution.
 - Validate compliance with requirements and standards.
 - Automate testing processes where feasible.
- **Skills Required:**
 - Experience in software testing methodologies.
 - Familiarity with testing tools and frameworks.
 - Knowledge of AWS testing services (AWS Device Farm, AWS Fault Injection Simulator).
 - Analytical and documentation skills.
- **Estimated Time Commitment:** Approximately 16 hours/week during Phases 4, 5, and 7 (Weeks 12–21).
- **Total Hours: 160 hours**

8. Training Specialist (1)

- **Role:** Training and Knowledge Transfer
- **Responsibilities:**
 - Develop training materials and programs tailored to Acme Brick's team.
 - Conduct training sessions on AWS services, DevSecOps practices, and operational procedures.
 - Create user manuals, guides, and runbooks.
 - Assess training effectiveness and provide follow-up support.
- **Skills Required:**
 - Experience in instructional design and training delivery.
 - Knowledge of AWS services and cloud technologies.
 - Strong communication and presentation skills.

- **Estimated Time Commitment:** Full-time during Phase 8 (Weeks 20–21).
- **Total Hours: 60 hours**

9. Support Engineers (2)

- **Role:** Post-Migration Support
- **Responsibilities:**
 - Monitor systems post-deployment for stability and performance.
 - Respond to incidents and provide troubleshooting support.
 - Implement optimizations based on operational insights.
 - Assist with the handover to Acme Brick's operational teams.
- **Skills Required:**
 - Experience in system administration and support.
 - Knowledge of AWS operational best practices.
 - Problem-solving and customer service skills.
- **Estimated Time Commitment:** Approximately 16 hours/week during Phase 9 (Weeks 22–24).
- **Total Hours per Engineer: 80 hours**
- **Combined Total Hours: 160 hours**

Total Estimated Resource Hours

Role	Total Hours
AWS Principal Solutions Architect	720
DevOps Engineers (2)	1,120
Security Engineer	400
Application Developers (2)	600
Database Administrator	160
Network Engineer	120
QA/Test Engineer	160
Training Specialist	60
Support Engineers (2)	160
Total	3,500 Hours

This resource breakdown reflects the optimized allocation of hours, ensuring that each role contributes effectively to the project's success while maintaining efficiency. The adjustments aim to reduce overall project hours without compromising quality, security, or the achievement of Acme Brick's global expansion objectives.