QRC Lift & Shift Playbook

Playbook PDF: QRC Lift & Shift Playbook (Latest).pdf

Problem Statement: QRC currently operates a critical analytic framework on-premises, consisting of three core components: a **services component** that handles data ingestion and APIs, a **route component** that manages communication between services and processes, and an **engine component** responsible for heavy computation and data processing. While this system has served the business well, increasing data volumes, higher computational demands, and growing operational complexity are pushing the limits of the current infrastructure. As the business scales, the on-premises infrastructure is proving inflexible, costly to maintain, and unable to meet the dynamic needs of the organization.

The company recognizes the need for a scalable, reliable, and cost-efficient solution and is exploring a **lift and shift migration** strategy to Amazon Web Services (AWS) as a way to achieve operational parity while reducing infrastructure overhead and unlocking greater scalability.

Key Challenges:

1. **Scalability Limitations:** The current on-premises setup struggles to dynamically scale resources to accommodate fluctuating workloads, leading to either resource over-provisioning or under-provisioning, which directly impacts performance and cost efficiency.
2. **High Maintenance Costs:** The company faces increasing costs associated with maintaining and upgrading physical infrastructure, including hardware maintenance, cooling, energy consumption, and data center space.
3. **Disaster Recovery and Resilience:** Ensuring high availability, disaster recovery, and data redundancy is becoming increasingly complex and expensive on-premises, putting the company's business continuity at risk.
4. **Time-consuming Deployments:** Provisioning new resources and rolling out updates within the existing framework involves manual processes that are time-consuming, slowing down the company's ability to react quickly to business needs.
5. **Infrastructure Inflexibility:** The on-premises architecture lacks the flexibility to easily integrate with modern technologies like artificial intelligence, big data analytics, and machine learning, which limits the company's ability to innovate and derive greater value from its data.
6. **Security and Compliance Risks:** Managing data security and meeting compliance requirements for data privacy and industry regulations in the on-premises environment is resource-intensive and error-prone, with limited monitoring and automation tools.

Business Impact:

- **Decreased Agility:** Slow resource provisioning and scalability challenges hinder the company’s ability to rapidly innovate or respond to changes in business demands.
- **Increased Costs:** The capital and operational expenses of maintaining the on-premises environment are diverting resources from strategic business initiatives.
- **Risk of Downtime:** The company is exposed to potential downtime due to inadequate disaster recovery and failover mechanisms, impacting customer satisfaction and revenue.
- **Lack of Advanced Capabilities:** The inability to easily adopt modern analytics technologies limits the company’s competitive edge and ability to leverage data for strategic decision-making.

Proposed Solution: To address these challenges, [Insert Company Name] should pursue a **lift and shift** strategy to migrate its analytic framework to AWS. This migration will:

- **Increase Scalability:** AWS’s elastic compute and storage services, such as EC2 and S3, can dynamically scale to handle workloads as they fluctuate, eliminating resource constraints.
- **Reduce Operational Costs:** By migrating to AWS, the company can move from a capital expenditure (CapEx) model to a pay-as-you-go operational expenditure (OpEx) model, reducing infrastructure costs.

- **Enhance Resilience and Recovery:** AWS provides built-in disaster recovery solutions, automated backups, and multi-region redundancy to ensure high availability and business continuity.
- **Improve Time-to-Market:** AWS enables faster resource provisioning and automation, improving the company's ability to roll out new services and updates.
- **Leverage Modern Technologies:** Migrating to AWS allows the company to more easily integrate with advanced analytics tools, AI, and machine learning services offered by AWS.
- **Strengthen Security:** AWS's extensive security features, including encryption, identity management, and compliance certifications, will help the company meet security and regulatory requirements more effectively.

By implementing a lift and shift strategy, QR will be able to improve its operational efficiency, reduce costs, and position itself for future growth and innovation on AWS.

Solution Strategy:

Lift and Shift Strategy for Migrating On-Premises Analytic Framework to AWS

Objective:
To migrate the on-premises analytic framework, which consists of a **services component**, **route component**, and **engine component**, to AWS using a "lift-and-shift" strategy. This approach minimizes code changes and focuses on replicating the current environment in the cloud to achieve operational parity, enhanced scalability, and cost-efficiency.

---

1. Pre-Migration Assessment

- **Inventory Existing Infrastructure:**
  - Document the current on-premises environment, including:
    - Service dependencies and versions of the services component.
    - Communication protocols and pathways within the route component.
    - Compute, memory, storage, and performance requirements of the engine component.
  - Evaluate all software dependencies and third-party tools to ensure compatibility with AWS.
  - Identify any potential licensing issues or migration blockers.
- **Assess Security & Compliance:**
  - Review the security and compliance requirements of the existing framework to ensure AWS services meet or exceed current standards.
  - Identify any data residency or encryption policies that must be adhered to.
- **Cost and ROI Analysis:**
  - Estimate AWS costs (EC2 instances, EBS volumes, S3 storage, data transfer, etc.).
  - Compare these costs against current on-premises operational expenses.

---

2. Choose AWS Services for Migration

- **Services Component:**
  - **EC2 Instances:** Lift and shift the services component directly to Amazon EC2 instances that match the current on-premises configuration (compute, storage, and memory).
  - **Auto Scaling:** Implement EC2 Auto Scaling to dynamically scale the services based on demand.
- **Route Component:**
  - **Elastic Load Balancer (ELB):** Use AWS Elastic Load Balancer to replace any existing on-prem load balancing between the services and engine components.

- **Amazon Route 53:** If DNS-based routing is used, migrate to Route 53 for highly available and scalable DNS routing.

- **Engine Component:**
  - **EC2 or AWS Lambda:**
    - For heavy, persistent compute workloads, migrate the engine component to EC2 instances.
    - For event-driven, short-duration workloads, explore using AWS Lambda for serverless execution.
  - **Amazon EBS & EFS:** Migrate the on-premises storage used by the engine to Amazon Elastic Block Store (EBS) or Amazon Elastic File System (EFS) depending on performance requirements.
  - **Amazon S3:** Offload data storage, backups, and analytic outputs to Amazon S3 for scalable, durable object storage.

---

3. Migration Plan by Component

A. Services Component:

- **Migration Process:**
  a. Provision Amazon EC2 instances equivalent to on-prem servers.
  b. Deploy services on EC2 instances using the same configurations as on-prem.
  c. Migrate related databases to AWS RDS if applicable, or maintain in EC2 if using custom DB engines.
  d. Enable AWS IAM roles to manage access control and permissions securely.
- **Key Considerations:**
  - If microservices are part of the architecture, explore the use of **Amazon ECS** or **EKS** for better container orchestration.

B. Route Component:

- **Migration Process:**
  a. Set up Amazon ELB in front of the services and engine components to manage load distribution.
  b. Migrate routing logic to AWS Route 53 if using DNS-based routing, setting up health checks and failover mechanisms.
  c. Ensure appropriate network security groups and firewalls are in place for traffic routing.
- **Key Considerations:**
  - AWS Direct Connect or VPN might be used if certain on-prem systems must still connect to AWS resources.

C. Engine Component:

- **Migration Process:**
  a. Migrate compute resources for the engine component to EC2 instances, ensuring similar performance and resource configurations (CPU, RAM, etc.).
  b. Configure autoscaling groups if the engine component has variable demand.
  c. Migrate any existing data repositories to Amazon S3 or Amazon EBS depending on the data structure and performance needs.
  d. Set up CloudWatch for monitoring performance and alerting on key engine metrics.
- **Key Considerations:**
  - Explore **Amazon EMR** if the engine is based on Hadoop/Spark for optimized data processing.

---

4. Security and Compliance

- **VPC and Network Security:**
  - Set up a Virtual Private Cloud (VPC) to segment network traffic securely.
  - Use Security Groups and Network Access Control Lists (NACLs) to restrict access.
- **Encryption:**
  - Use **AWS KMS** to manage encryption for data at rest and in transit.

- **IAM Policies:**
  - Implement least-privilege access control using AWS Identity and Access Management (IAM).
- **Compliance Checks:**
  - Leverage **AWS Artifact** for compliance certifications and ensure adherence to industry regulations such as GDPR, HIPAA, etc.

---

5. Testing and Validation

- **Environment Testing:**
  - Run pilot tests of the services, route, and engine components in AWS before cutting over to the cloud environment.
  - Use **AWS CloudWatch** and **X-Ray** to monitor and trace the performance and ensure that all components communicate as expected.
- **Data Validation:**
  - Ensure data integrity by comparing pre- and post-migration data between on-premises and AWS.
  - Use **AWS DataSync** or **AWS Snowball** if bulk data transfers are needed.
- **Failover Strategy:**
  - Use **Amazon CloudFront** and **ELB** failover mechanisms to handle potential service disruptions.
  - Implement a backup and disaster recovery plan using **AWS Backup**.

---

6. Cutover and Go-Live

- Plan a **staged cutover** to gradually migrate critical workloads to AWS, ensuring no disruption to live services.
- Leverage **DNS redirection** to smoothly redirect traffic from on-premises to AWS after successful testing.
- Monitor the environment closely during the cutover period using CloudWatch and **AWS Trusted Advisor** to optimize performance and cost.

---

7. Post-Migration Optimization

- **Performance Tuning:**
  - Adjust instance types, storage configurations, and scaling rules based on post-migration performance metrics.
- **Cost Optimization:**
  - Use **AWS Cost Explorer** and **Reserved Instances** to optimize long-term operational costs.
- **Automation:**
  - Explore opportunities for automation using **AWS Lambda**, **Elastic Beanstalk**, and other AWS tools for continuous improvement.

By following this lift-and-shift strategy, QRC can seamlessly migrate its analytic framework to AWS, leveraging AWS's scalable infrastructure, cost efficiencies, and modern security tools while maintaining operational continuity.

Addendum: Handover Strategy

Optimal Method for Handover:

1. **Early Engagement and Collaboration:**
   - **Involve the Customer's Team from the Start:** Engage the customer's IT and operations teams early in the migration process to foster ownership and familiarity with the new environment.

- **Regular Communication:** Maintain open lines of communication through regular meetings and updates to keep the customer informed of progress and upcoming tasks.

2. **Comprehensive Documentation:**
   - **Detailed Architecture Diagrams:** Provide clear diagrams of the new AWS architecture, including network layouts, security groups, and service interactions.
   - **Configuration and Deployment Guides:** Offer step-by-step guides on how resources were configured and deployed, including any Infrastructure as Code (IaC) templates used (e.g., AWS CloudFormation, Terraform).
   - **Operational Runbooks:** Develop runbooks for routine operations, incident management, and disaster recovery procedures.
3. **Knowledge Transfer and Training:**
   - **Training Sessions:** Conduct formal training sessions covering AWS services, best practices, and specific configurations relevant to the customer's environment.
   - **Hands-On Workshops:** Provide practical workshops where the customer's team can gain experience by managing the environment under guidance.
   - **Shadowing Opportunities:** Allow the customer's staff to shadow the migration team during key activities to build confidence and understanding.
4. **Establish Governance and Best Practices:**
   - **Security and Compliance Frameworks:** Ensure the customer understands AWS security best practices, identity and access management, and compliance requirements.
   - **Cost Management Strategies:** Educate the customer on cost optimization techniques and tools like AWS Cost Explorer and AWS Budgets.
   - **Operational Excellence:** Share insights on monitoring, logging, and alerting using services like Amazon CloudWatch, AWS CloudTrail, and AWS Config.
5. **Support Structures:**
   - **Define Support Escalation Paths:** Establish clear support channels and escalation procedures for post-handover assistance.
   - **Service Level Agreements (SLAs):** Set expectations for response times and support availability.
6. **Pilot and Validation:**
   - **Pilot Migrations:** Begin with migrating non-critical workloads to validate the process and allow the customer to gain experience.
   - **Performance Testing:** Conduct performance and stress tests to ensure the environment meets required standards.

Optimal Cutoff Point for Handover:

The ideal cutoff point for handing over the migration is after the following milestones have been achieved:

1. **Completion of Migration Phases:**
   - **Data Migration:** All necessary data has been securely and accurately migrated to AWS.
   - **Application Migration:** Applications are running successfully in the AWS environment with validated functionality.
2. **Stabilization Period:**
   - **Monitoring and Adjustments:** Post-migration, the environment has been monitored, and any performance tuning or issue resolutions have been completed.
   - **Validation Testing:** Successful completion of user acceptance testing (UAT) and sign-off from key stakeholders.
3. **Operational Readiness:**
   - **Staff Competency:** The customer's team demonstrates the ability to manage and operate the AWS environment independently.
   - **Process Implementation:** Necessary operational processes, including backup, patch management, and incident response, are in place.
4. **Formal Handover Documentation:**
   - **Final Documentation Delivered:** All documentation is updated and handed over, including any changes made during the migration.
   - **Acceptance Sign-Off:** Formal acceptance of the environment and operational responsibility by the customer.
5. **Post-Handover Support Plan:**

- **Transition Support Period:** An agreed-upon period during which the migration team provides support to address any issues that arise.
- **Feedback Mechanisms:** Establish channels for ongoing feedback and continuous improvement.

Additional Considerations:

- **Risk Management:** Identify and mitigate any risks associated with the handover, ensuring contingency plans are in place.
- **Compliance and Security Audits:** Complete any necessary audits to confirm compliance with industry regulations and security standards.
- **Change Management:** Ensure all stakeholders are informed and prepared for the transition to minimize disruption.

By carefully planning the handover process and selecting a cutoff point that aligns with the customer's operational readiness, you facilitate a smooth transition. This approach ensures the customer is confident in managing their new AWS environment, reducing the risk of issues post-migration and setting the foundation for successful cloud operations.

Playbook Roles & Responsibilities:

1. **Project Manager:**
   - **Responsibilities:** Oversees the project development, schedule and handover process, coordinate activities between teams, and ensure that timelines and deliverables are met.
   - **Importance:** Provides leadership and maintains the overall direction and momentum of the project and handover.
2. **AWS Solutions Architect:**
   - **Responsibilities:** Share detailed knowledge of the AWS infrastructure, architecture design, and services used.
   - **Importance:** Ensures the customer understands the technical aspects of the AWS environment for effective management and future scaling.
3. **Cloud Engineers/DevOps Engineers:**
   - **Responsibilities:** Provide insights into infrastructure deployment, automation scripts, CI/CD pipelines, and operational best practices.
   - **Importance:** Equip the customer's technical team with the skills needed to maintain and evolve the cloud environment.
4. **Security Engineer:**
   - **Responsibilities:** Explain security configurations, identity and access management (IAM) roles, encryption methods, and compliance measures.
   - **Importance:** Ensures the customer can uphold security standards and comply with regulations.