AnyCompany AWS Architecture Evaluation Report

Team Members: Yasmeen Shakarnh – Salsabeel Al-Maghalsa – Ruaa Abumqadam Safa Ghith

Project Introduction: AnyCompany

AnyCompany is an innovative business founded in 2008 by **John Doe**, specializing in the creation and sale of high-resolution 3D-printed cityscapes under the slogan: **"Cityscapes you can stand over."** The company has developed proprietary technology to reconstruct detailed 3D models using a combination of photographs and video footage through structure-from-motion techniques.

The platform is built on a **cloud-native architecture using AWS services**, and it operates across three main stages:

1. **Fly and Snap:**
   Multiple camera devices are mounted on lightweight aircraft that capture aerial imagery of cities. The captured data—synchronized with navigation systems—is stored on external storage arrays, which are later transferred to an ingest system for compression, backup, and upload to the cloud.
2. **Show and Sell:**
   Through AnyCompany's website, customers can preview and customize cityscapes. The site integrates with backend services to confirm image availability and process orders via a PCI-compliant payment gateway. Once payment is confirmed, the order flows into the production system and is recorded in a central database.
3. **Make and Ship:**
   Rendering services generate 3D models and flyby preview videos using EC2 GPU instances. Orders are sent to on-premises 3D printers via a print management system. Status updates are pushed to customers in real time, and completed orders are handed over to a third-party dispatch service.

Now preparing for potential **external investment**, AnyCompany is undergoing a **technical review based on the AWS Well-Architected Framework**, which includes the following five pillars:

- **Operational Excellence**
- **Security**
- **Reliability**
- **Performance Efficiency**
- **Cost Optimization**

This review aims to assess the **current state** of AnyCompany's cloud architecture, define the **future state**, and identify **key improvements** that will help the platform scale, remain secure, and become more efficient—positioning it for long-term success and investor confidence.

❖ Operational Excellence pillar

A review of the current operational and technical landscape highlights several challenges affecting system efficiency and team collaboration. In the area of **Monitoring**, there is no centralized system, which hinders effective performance tracking. It is recommended to implement tools like AWS CloudWatch and X-Ray, along with integrated dashboards and service health checks to enhance reliability.

For **Priority Setting**, the absence of a clear framework leads to scattered efforts. Weekly review meetings focused on customer and business needs are proposed to better define focus areas.

The **Organizational Structure** suffers from disconnected departments, limiting collaboration. A shift to integrated, cross-functional teams, supported by shared communication channels and documented workflows, is advised to improve cohesion.

In terms of **Work Culture**, there is limited emphasis on innovation. Promoting a culture of continuous improvement through knowledge-sharing sessions and encouraging open feedback can foster collaboration and creativity.

Regarding **Quality Testing**, the current state lacks pre-deployment checks. Establishing a Staging environment and implementing automated testing through Continuous Integration (CI) pipelines is suggested to enhance deployment quality.

For **Deployments**, the current manual approach is error-prone. Moving to CI/CD pipelines using AWS CodePipeline and adopting a Blue/Green deployment strategy will enable safer, more reliable releases.

Incident Response currently lacks a structured plan, delaying resolution times. Developing a comprehensive response strategy and running simulation exercises will help ensure readiness and minimize downtime.

In the area of **Workload Readiness**, there is uncertainty around capacity limits. Load testing, enabling Auto Scaling, and conducting Game Days can validate system resilience under high traffic.

Performance Tracking is currently lacking. Introducing metrics collection and weekly KPI reporting is essential for performance visibility and data-driven decision-making.

Finally, **Ops Evolution** is hindered by outdated procedures. Monthly reviews and iterative upgrades are recommended to support continuous operational excellence.

❖ Security Pillar

The current security landscape reveals critical gaps in control, visibility, and preparedness across workloads. In the area of **Workload Security**, there is no centralized control or auditing for access, exposing systems to potential misuse. To achieve a secure, least-privilege model, it's recommended to implement IAM roles and policies, enforce least-privilege access, and activate AWS CloudTrail for comprehensive logging.

Security Event Detection is currently limited, reducing the ability to identify breaches early. A more proactive approach involves enabling Amazon GuardDuty, configuring AWS Config rules, and integrating findings into AWS Security Hub for centralized threat detection and response.

In terms of **Compute Protection**, basic security configurations leave systems vulnerable. To strengthen this area, it is proposed to enforce strict security group rules, use EC2 Image Builder for hardened images, and automate patching through AWS Systems Manager.

For **Identity and Access Management**, user provisioning is manual and permissions are inconsistent. The ideal state involves automated provisioning with fine-grained control. Using AWS IAM Identity Center (SSO), applying permission boundaries, and enforcing multi-factor authentication (MFA) are key steps toward achieving secure and scalable access management.

Infrastructure Protection currently relies on a flat network with limited segmentation, increasing risk exposure. Transitioning to a segmented architecture using VPCs, subnets, route tables, and security groups, alongside the implementation of network ACLs and AWS Firewall Manager, will significantly enhance network security.

In the realm of **Data Protection**, sensitive data is often unencrypted and poorly classified. To align with best practices, it's recommended to enable KMS encryption for data in transit and at rest, apply S3 bucket policies, and utilize Amazon Macie for sensitive data discovery and classification.

Finally, **Incident Response** lacks formal plans or playbooks, making recovery efforts inconsistent. Establishing well-defined runbooks, automating workflows with AWS Systems Manager, and conducting regular incident simulations will ensure the organization is prepared to respond effectively to security events.

❖ Reliability Pillar

The current system architecture reveals several weaknesses that threaten overall reliability and resilience. **Service Quotas** are not actively monitored, creating a risk of hitting usage limits without warning. To address this, it's recommended to integrate AWS Service Quotas with CloudWatch and configure proactive alarms. **Network Topology** is overly simple, likely

constrained to a single Availability Zone, which limits fault isolation. Redesigning the network to use multi-AZ/multi-region VPCs with public and private subnets will improve fault tolerance and scalability.

In terms of **Service Architecture**, there's a heavy reliance on EC2 with limited use of managed AWS services. Transitioning to scalable and decoupled services like ECS, Lambda, and RDS can enhance reliability and reduce operational overhead. The current approach lacks **Interaction Failure Prevention** mechanisms, such as retries or circuit breakers. Incorporating idempotent operations, timeout handling, and retry/backoff logic with circuit breakers is essential for improving service robustness.

Failures are also not mitigated properly—there are no **dead-letter queues** or fallback processes. Implementing DLQs for all SQS queues, along with alerting and fallback flows, will improve resilience in messaging systems. **Monitoring Resources** are limited to basic metrics. A unified observability strategy using CloudWatch, X-Ray, and custom dashboards will provide better visibility and faster issue resolution. While the web front-end supports autoscaling, **Demand Adaptability** is lacking on the backend. Enabling Auto Scaling across all service tiers and tuning based on usage trends is key to achieving elastic scalability.

Further gaps exist in **Change Implementation**, where manual deployments increase the risk of errors. The solution is to adopt a CI/CD pipeline using AWS CodePipeline, CodeBuild, and Infrastructure as Code (IaC) with approval and rollback stages. **Data Backup** practices are outdated, relying on tapes without coverage for live systems. It's critical to automate backups using RDS snapshots, S3 versioning, and regularly conduct DR drills.

There is minimal **Fault Isolation** due to operating within a single AWS account with loosely separated environments. Segregating workloads into different accounts or VPCs with strict IAM controls can significantly limit blast radius. The infrastructure also lacks **Component Failure Resilience**, with no auto-healing or failover in place. Implementing EC2 Auto Recovery, load balancing, and multi-AZ/multi-region failover configurations is necessary.

Finally, **Reliability Testing** and **Disaster Recovery** capabilities are nearly nonexistent. Scheduling game days, introducing chaos engineering practices, and developing a documented, tested disaster recovery strategy—including cross-region replication and clearly defined RTO/RPO goals—are essential steps toward a resilient, enterprise-grade infrastructure.

❖ Performance Efficiency Pillar

A review of the current cloud infrastructure highlights multiple opportunities to modernize and improve operational efficiency. In terms of **Architecture**, the system currently depends on an EC2-based setup with manual processes, which limits scalability and agility. Transitioning to managed and serverless AWS services such as Lambda, RDS, and S3 is recommended to reduce operational burden and enhance scalability.

For **Compute Resources**, the current use of EC2 instances like g2.2xlarge for rendering presents limitations in flexibility and performance. It is advised to adopt containerized solutions with ECS and Fargate or upgrade to G5 GPU instances, enabling auto scaling and better resource utilization.

With regard to **Storage**, imagery is stored in S3 while archives rely on tape-based storage, which is slow and difficult to manage. Replacing tape with S3 Glacier Deep Archive and applying S3 Lifecycle policies will optimize storage costs and simplify archive management.

In the area of **Database Management**, the use of a self-managed RDBMS on EC2 results in increased administrative overhead and potential availability issues. Migrating to Amazon RDS or Aurora with Multi-AZ deployment and automated backups will enhance reliability, availability, and ease of management.

For **Networking**, the current basic VPC setup lacks secure segmentation and defined access controls. Implementing VPC best practices, such as using PrivateLink, service endpoints, and IAM roles, will improve security and connectivity.

When considering **Architecture Evolution**, there is no structured process in place. Establishing a regular review cycle, such as semi-annual architecture assessments using the AWS Well-Architected Tool, is essential to ensure continuous improvement and alignment with evolving best practices.

In terms of **Monitoring**, the absence of centralized tools leads to blind spots in system health and performance. Introducing AWS CloudWatch, CloudTrail, and X-Ray, along with real-time dashboards and alerts, will provide comprehensive observability and faster incident response.

Finally, to **Optimize Performance**, reliance on EC2 for all workloads and tape storage increases overhead and reduces efficiency. Shifting to serverless compute options and using S3 Glacier for long-term storage will minimize management effort while improving system scalability and cost-effectiveness.

❖ Cost Optimization Pillar

Currently, AnyCompany lacks a centralized approach to **cloud financial management**, leading to limited visibility into usage, costs, and optimization opportunities. There is no structured governance or active monitoring of resource utilization, which raises the risk of over-provisioning and unused resources—especially across EC2 instances used for preprocessing, rendering, and database workloads. To improve, AnyCompany should adopt a Cloud Financial Management (CFM) strategy that includes tagging resources, setting budgets, and using AWS Cost Explorer and AWS Budgets for real-time cost monitoring and forecasting.

In terms of **expenditure and usage awareness**, many workloads continue running without automation for shutdown, right-sizing, or decommissioning. FTP-based transfers and infrequent deletion of old data (such as preview videos) suggest inefficient storage practices. Implementing automation scripts to decommission idle resources, using S3 lifecycle policies, and enabling storage class transitions (e.g., to S3 Infrequent Access or S3 Glacier) will help optimize costs.

Regarding **cost-effective resources**, current reliance on general-purpose EC2 instances (e.g., g2.2xlarge) misses opportunities to leverage more cost-efficient alternatives like Graviton processors or spot instances. Services such as Lambda, Fargate, or Amazon Batch can reduce costs in the preprocessing and rendering pipelines. Evaluating service selection using pricing calculators and performance benchmarks will allow for better alignment with cost targets.

Demand and supply management is mostly manual—front-end services scale automatically, but backend components remain static. By enabling Auto Scaling across backend workloads and applying demand-based provisioning for render and ingest tasks, AnyCompany can reduce waste during low usage periods.

Finally, the company lacks a strategy to **optimize over time** by evaluating new AWS services or pricing models. Regular reviews of new offerings—such as Savings Plans, Compute Optimizer insights, or Graviton migration assessments—should be embedded in the development lifecycle to ensure long-term cost efficiency.

❖ Conclusion

In summary, AnyCompany has built a unique and innovative platform that combines aerial data capture, 3D rendering, and custom product manufacturing to deliver high-quality cityscape models. While the current architecture provides a solid foundation, the review based on the AWS Well-Architected Framework has revealed key areas for improvement across all five pillars—Operational Excellence, Security, Reliability, Performance Efficiency, and Cost Optimization.

By implementing best practices such as automation, centralized monitoring, security hardening, and scalable infrastructure, AnyCompany can enhance its cloud environment to be more robust, efficient, and secure. These improvements will not only support future growth and investor readiness but also ensure a better experience for customers and more resilient operations.

With the right strategic changes, AnyCompany is well-positioned to scale its business and continue delivering its vision of "Cityscapes you can stand over" on a global level.