**AIQX Microservice Application**

# Architecture Components

1. **Application Gateway**
   * Global routing and web application firewall.
   * For low-latency and failover across multiple regions
2. **Azure Kubernetes Service (AKS)**
   * Hosts all microservices:
     + **Auth Service** (Login/Register)
     + **User Service**
     + **Data API Service**
     + **API Validator Service**
     + **API Gateway - NGINX Ingress**
3. **Azure Container Registry (ACR)**
   * Stores container images
4. **Azure SQL Database / Cosmos DB**
   * Used for transactional and data storage.
5. **Azure Key Vault**
   * Securely stores secrets and tokens for external API access.
6. **Azure Monitor + Log Analytics**
   * Observability for AKS
7. **Virtual Network + Network Policies**
8. **External API Providers**

# High Availability

1. **AKS with multiple node pools and zones**:
   * We are going to use Availability Zones to deploy nodes across different data centers, this will ensure that services remain up if one zone.
2. **Application Gateway**:
   * Detects unhealthy endpoints and reroutes traffic.
   * As well as providing SSL termination
3. **Liveness and Readiness Probes –** included in Pods:
   * To detect unhealthy pods.

# Fault Tolerance

1. **Enable Pod Replication in AKS**:
   * Services are deployed with multiple replicas.
2. **Horizontal Pod Autoscaling (HPA)**:
   * Scale pods horizontally to prevents app from crashing due to traffic overload.

# Cost Optimization

1. Use Spot Nodes in AKS for non-critical workloads
2. Auto-scaling at node and pod levels
3. Serverless Azure SQL which helps with paying only for what you use.
4. Azure Reserved Instance
5. Container Optimization:
   * Use distroless images, minimize image size, and remove unused libraries to reduce resource usage.
6. Monitoring with Azure Monitor to Identify and remove underutilized resources.