Enterprise Azure Design: Principles & Quick Reference

Accelerate Your Solutions with Strategic Azure Design Choices

Limcify.com

Table of Contents

1. Introduction

This document serves as a concise guide to the fundamental design principles required for architecting enterprise-grade solutions on Microsoft Azure. Each entry presents a core architectural concept tested in the AZ-305 exam and offers a validated strategic recommendation. The focus is on balancing security, cost, scalability, and compliance—the four pillars of effective solution architecture.

2. Identity, Governance, and Security

2.1. Governance Fundamentals

Concept	Design Recommendation

Management Groups & Policy Inheritance	Organize subscriptions under **Management Groups (MGs)**. Assign **Azure Policies** at the MG level to ensure consistent governance across all child subscriptions and resources.
Azure Policy vs Initiative vs Blueprint	**Policy** is an individual rule. **Initiative** is a group of policies. **Blueprint** is a package of policies, **RBAC**, and resource templates for repeatable, compliant deployments.
RBAC (Role-Based Access Control)	Apply the **principle of least privilege** at the lowest required scope (Resource Group or Resource). Only use custom roles when built-in roles are truly insufficient.
Resource Locks	Apply CanNotDelete or ReadOnly locks to **critical production resources** to prevent accidental modification or deletion.
Tagging Strategy	Enforce standardized tags (**Owner, Environment, CostCenter**) consistently across all resources. Use **Azure Policy** to enforce automatic tag compliance.

2.2. Identity Management

Concept	Design Recommendation
Privileged Identity Management (PIM)	Enable **Just-in-Time (JIT)** access for administrators. Require approval and **MFA** for all high-privilege roles to minimize the attack surface.
Conditional Access (CA)	Enforce **MFA** and risk-based access rules. Combine CA with device compliance checks or specific location filters for sensitive applications.
Managed Identities	Use **Managed Identities** to replace service principals or credentials in application code. This is the preferred

	method for secure resource-to-resource access within Azure.
Zero Trust Architecture	Apply the "never trust, always verify" model. Enforce granular identity, device, and network-based access controls for every transaction.

2.3. Security & Key Management

Concept	Design Recommendation
Key Vault + CMK + Always Encrypted	Store and manage encryption keys centrally in **Key Vault**. Use **Customer-Managed Keys (CMK)** for regulatory compliance. Use **Always Encrypted** to protect sensitive database columns from administrators.
Data Encryption (TDE, Always Encrypted)	**Transparent Data Encryption (TDE)** protects data at rest at the database file level. **Always Encrypted** protects data in use, even from database administrators (DBAs).

3. Infrastructure and Networking

3.1. Infrastructure Design Choices

Concept	Design Recommendation
VM Scale Sets & Availability Sets/Zones	**Scale Sets** enable automated horizontal scaling. **Availability Sets** provide rack-level fault isolation. **Availability Zones** provide datacenter-level fault tolerance within a region.
Azure Functions / App Service / AKS	Use **Functions** for event-driven, serverless compute. Use **App Service** for managed web/API hosting. Use **AKS** for container

	orchestration requiring fine-grained control and complex scaling.
Storage Redundancy (LRS, ZRS, GRS, RA-GRS)	Select redundancy based on required SLA and geo-availability. **LRS** (local) is lowest cost/risk. **ZRS** (zone) protects against a single AZ failure. **GRS** (geo) protects against regional failure.
Database Choice (SQL, MI, Cosmos DB)	Use **Azure SQL/Managed Instance (MI)** for relational workloads. Use **Cosmos DB** for globally distributed NoSQL with tunable consistency and high-throughput needs.

3.2. Networking Strategy

Concept	Design Recommendation
Private Endpoints vs Service Endpoints	**Private Endpoint** provides private IP access to PaaS services via the Azure backbone. **Service Endpoint** extends VNet identity to the PaaS service over the public IP.
VNet Peering vs VPN Gateway	Use **VNet Peering** for low-latency, high-bandwidth communication between VNets within Azure. Use **VPN Gateway** for encrypted site-to-site or cross-region connectivity (IPsec/SSL).
ExpressRoute	Use **ExpressRoute** for dedicated private connectivity between on-premises environments and Azure with guaranteed higher bandwidth and reliability (avoids the public internet).
Azure Firewall vs NSG	**Azure Firewall** performs stateful Layer 3–L7 inspection across multiple VNets. **NSG (Network Security Group)** performs stateless L3/L4 filtering on individual VNICs or subnets. They are often used together.

3.3. Load Balancing & Traffic Routing

Concept	Design Recommendation
LB vs App Gateway vs Front Door vs Traffic Manager	**Load Balancer (L4)** for internal network traffic. **Application Gateway (L7 + WAF)** for public web apps. **Front Door** for global web acceleration and site failover. **Traffic Manager** for DNS-based geo-routing and basic failover.

4. Business Continuity and Disaster Recovery (BCDR)

Concept	Design Recommendation
Disaster Recovery (ASR)	Use **Azure Site Recovery (ASR)** for replicating Virtual Machines (VMs) across regions. Always define clear **RTO (Recovery Time Objective)** and **RPO (Recovery Point Objective)** targets.
Backup Strategy	Configure **soft delete** for accidental deletion protection. Use **GRS (Geo-Redundant Storage)** for backups. Test restore processes regularly to validate your RTO/RPO targets.
Business Continuity Design	Pair a primary region with a synchronized **secondary region**. Use **Traffic Manager** or **Front Door** to automatically manage region failover and redirect traffic based on availability.

5. Monitoring and Observability

Concept	Design Recommendation
Azure Monitor (Metrics, Logs, Alerts)	Collect performance and activity telemetry

	centrally. Define **actionable alerts** and route notifications via **Action Groups**.
Log Analytics & KQL	Centralize diagnostic logs from all Azure resources into a **Log Analytics Workspace**. Run **Kusto Query Language (KQL)** queries for in-depth insight and troubleshooting.
Application Insights	Implement **distributed tracing** and dependency tracking within your applications. Integrate with Log Analytics to correlate application performance with infrastructure metrics.
Defender for Cloud & Sentinel	**Defender for Cloud** provides Cloud Security Posture Management (CSPM) and Cloud Workload Protection (CWPP). **Sentinel** is the **SIEM/SOAR** platform for threat correlation and automated response.
Hybrid Monitoring (Azure Arc)	Use **Azure Arc** to onboard on-premises servers and other-cloud resources to Azure Monitor and Defender for unified observability and governance.
Monitoring Retention & Cost Optimization	Configure log retention periods strictly based on **compliance needs**. Archive older logs to cheap **Azure Storage** for cost control.

6. Cost Management and Compliance

Concept	**Design Recommendation**
Cost Management & Budgets	Use **Azure Budgets** to set spending thresholds and trigger alerts via **Action Groups**. Monitor cost trends using the Azure Cost Management portal.

Autoscale & Performance Tuning	Utilize **VM Scale Sets (VMSS)** or App Service autoscale rules to dynamically adjust capacity. Regularly review performance metrics to **right-size resources** and ensure resources aren't over-provisioned.
Compliance Frameworks	Map business requirements (**ISO, HIPAA, PCI, GDPR, NIST**) to specific **Azure Policy Initiatives** and monitor compliance using **Defender regulatory dashboards**.

7. Key Takeaways & Design Strategy

1. **Prioritize Governance First:** Before deployment, establish your Management Group hierarchy, deploy essential **Azure Policies**, and enforce a robust **tagging strategy** for cost allocation.
2. **Identity is the Perimeter:** Adopt a **Zero Trust** mindset. Use **Managed Identities** instead of service credentials and enforce **PIM** for all privileged access.
3. **Balance Trade-offs:** The AZ-305 exam (and real architecture) requires balancing **Security, Cost, Scalability, and Manageability.** Never default to the most expensive or complex solution if a simpler, more cost-effective one meets the requirements (e.g., App Service over AKS for simple web apps).
4. **Security in Layers:** Implement a defense-in-depth approach, combining **Azure Firewall** (perimeter) with **NSGs** (segmentation) and **Key Vault** (secrets/keys).
5. **Design for RTO/RPO:** Always anchor your BCDR and backup decisions to quantifiable business requirements (**RTO/RPO**) rather than just choosing the highest redundancy level.

8. About the Author

Limcify.com

Limcify is a dedicated Azure and Architecture learning brand founded by a **23-year veteran Enterprise Architect**. Our mission is to move IT professionals beyond simple certification knowledge and equip them with the strategic frameworks and real-world methodologies needed to succeed in high-stakes cloud architecture roles. We believe that true Solution Architects master not only the technology (Azure) but also the discipline of governance, risk management, and business value alignment (TOGAF, BizzDesign). Limcify provides the premium content, practical Architecture Kits, and expert insights required to close the gap between passing the exam and performing at an executive level.

Limcify.com: Smart Azure Learning.