Application Landing Zone Design

Name	Application Landing Zone Design
ID	ID0XX
Version	v1.0
Category	Services / Architecture / Technical Architecture
Description	This service delivers comprehensive design and documentation for an implementation-ready, application-specific landing zone in public cloud environments (Azure, AWS, GCP). The service provides detailed technical blueprints, architectural diagrams, and specifications that enable organizations to host applications safely and consistently within their established cloud foundation. The comprehensive scope of architectural design translates assessment outcomes and target architecture principles into concrete, actionable designs covering networking, identity and access management, security controls, compute patterns, data layer architecture, DevOps integration, monitoring strategies, and operational aspects. This holistic approach ensures all critical infrastructure components are designed to work together seamlessly while adhering to enterprise standards and compliance requirements. This service enables organizations to accelerate application cloud adoption by providing a standardized, secure, and well-documented foundation for workload deployment. It covers single-cloud scenarios with design patterns that can be adapted for multi-cloud or hybrid environments when required. This is an implementation-ready design service only—it produces all artifacts required for deployment but does not include hands-on implementation or configuration activities. **Note:** This service assumes that an Enterprise-Scale Landing Zone for the organization is already prepared and that basic architecture and design principles for deploying applications in cloud environments are defined within the organization.

Usage scenarios	**1. Post-Assessment Implementation Planning** Organizations need a concrete design to translate infrastructure, platform, or application assessment findings into an actionable cloud deployment blueprint. **2. Enterprise Landing Zone Application Onboarding** Before onboarding new applications to an existing enterprise landing zone, organizations require standardized application-specific designs that align with corporate cloud governance. **3. Standardized Environment Creation** When building standardized application environments (production/non-production) that need consistent architecture patterns, security controls, and operational procedures. **4. Regulated Workload Deployment** When onboarding regulated or business-critical workloads that require detailed compliance mapping, security architecture, and audit-ready documentation. **5. Shared Foundation Multi-Team Deployment** When multiple development teams will deploy applications into a shared cloud foundation, requiring clear boundaries, access models, and resource isolation patterns. **6. Application Modernization Landing** Organizations migrating legacy applications to cloud need a dedicated landing zone design that accommodates specific technical requirements and integration dependencies. **7. Disaster Recovery Architecture** When applications require multi-region resilience and business continuity designs integrated into the landing zone architecture. **8. DevOps-Enabled Platform Foundation** Organizations requiring CI/CD-ready infrastructure designs with Infrastructure as Code integration points and automated deployment patterns.
Dependencies	Prerequisite services • Enterprise Scale Landing Zone Design (Required) • Cloud Network Architecture (Required) • Application Assessment (Recommended) • Infrastructure Assessment (Optional) • Application Architecture (Optional) Triggers for • Application Landing Zone Build (Required) • Application Migration (Required) • Application Modernization (Optional) • Container Platform Architecture (Recommended) Can run in parallel with • Application Architecture • Container Platform Architecture

In scope	**1. Platform Architecture** • Application landing zone architecture aligned with enterprise standards • Subscription / account structure for application workloads • Resource group / project organization model • Environment separation model (dev/test/staging/prod) **2. Network Architecture** • Network topology (VNets/VPCs, subnets, routing, connectivity) • Hub-spoke or mesh connectivity patterns • Private DNS zone design • Traffic flow and routing design **3. Security Architecture** • Security controls (NSGs, firewalls, private endpoints, encryption) • Network segmentation and micro-segmentation • Data encryption (at rest and in transit) • Key management architecture **4. Identity & Access Management** • Identity & access model (RBAC, managed identities, service roles) • Privileged access management design • Service principal and managed identity patterns • Federation and authentication flows **5. Compute Architecture** • Compute model (VMs, containers, PaaS services) • Scaling patterns and configurations • Container orchestration design (if applicable) • Serverless integration patterns **6. Data Architecture** • Data layer design (databases, storage, backups) • Data residency and sovereignty considerations • Backup and retention policies • Data lifecycle management **7. Integration Architecture** • CI/CD and IaC integration points • Application integration patterns • API management considerations • Event-driven architecture patterns **8. Observability Architecture** • Logging, monitoring, alerting design • Centralized logging strategy • Performance monitoring patterns • Dashboard and visualization specifications **9. Resilience & Continuity** • DR/BCP design considerations • High availability patterns • Failover and recovery procedures • RPO/RTO mapping

	10. Governance Framework • Naming conventions, tagging, and standards • Policy definitions and assignments • Cost allocation and chargeback tagging • Compliance control mapping **11. Documentation & Handover** • Bill of Materials (logical design, not pricing) • Handover-ready documentation • Design decisions and rationale log • Environment definition files
Out of scope	• Actual deployment or configuration of cloud resources • Application refactoring or code changes • Detailed CI/CD pipeline implementation (design only) • Security penetration testing or vulnerability assessments • Operational runbooks development (available as optional add-on) • Cost optimization beyond design assumptions • License procurement or management • Third-party tool implementation • Performance testing and benchmarking • End-user training • Ongoing operational support
Prerequisite	**Organizational Prerequisites** • Identified application owners and stakeholders • Designated project sponsor with decision authority • Clear organizational cloud strategy and governance model • Defined target operating model for cloud operations • Stakeholder availability for workshops (minimum 80% attendance) **Technical Prerequisites** • Completed relevant assessments (infrastructure/application/platform) • Enterprise landing zone principles or reference architecture established • Target cloud provider selected (Azure / AWS / GCP) • Application non-functional requirements (NFRs) documented • Existing network architecture documentation (if hybrid connectivity required) **Documentation Prerequisites** • Application portfolio documentation • Compliance and security requirements documentation • Current state architecture diagrams (if applicable) • Integration dependency mapping • Data classification policies

Required environment and tools

Cloud Platforms

Capability	AWS	Azure	GCP
Reference Architecture	AWS Well-Architected Framework	Azure Cloud Adoption Framework (CAF)	Google Cloud Architecture Framework
Landing Zone Accelerator	AWS Control Tower	Azure Landing Zone Accelerator	Google Cloud Foundation Toolkit
Policy Management	AWS Organizations SCPs	Azure Policy	GCP Organization Policies
Identity	AWS IAM Identity Center	Microsoft Entra ID	Cloud Identity

Design & Documentation Tools

- Diagramming tools (Visio, Lucidchart, Draw.io, Diagrams.net)
- Cloud-native diagram tools (AWS Architecture Icons, Azure Diagrams, GCP Architecture Diagramming)
- Documentation platforms (Confluence, SharePoint, Notion)
- Architecture decision record (ADR) templates

Infrastructure as Code Reference

Category	Column 2
IaC Frameworks	Terraform, Bicep, CloudFormation, Pulumi
Module Libraries	Terraform Registry, Azure Verified Modules, AWS Solutions Library
Policy as Code	OPA/Rego, Azure Policy, AWS Config Rules
Version Control	Git (GitHub, GitLab, Azure DevOps)

Assessment & Analysis Tools

- Cloud provider assessment tools
- Network topology analyzers
- Dependency mapping tools
- Cost estimation calculators (AWS Pricing Calculator, Azure Pricing Calculator, GCP Pricing Calculator)

Note: No production environment access required. Design work is based on assessment outputs and stakeholder workshops.

License	**Required by Customer**
	• Active cloud provider account(s) with appropriate permissions:
	  ◦ Azure: EA/CSP/MCA agreement with subscription creation rights
	  ◦ AWS: AWS Organizations with account creation rights
	  ◦ GCP: GCP Organization with project creation rights
	• Access to enterprise identity provider (Microsoft Entra ID, Okta, etc.)
	
	Recommended/Optional
	• Premium/Enterprise tier for enhanced security features:
	  ◦ Azure: Microsoft Entra ID P1/P2 for advanced identity features
	  ◦ AWS: AWS Security Hub, GuardDuty
	  ◦ GCP: Security Command Center Premium
	• Advanced networking licenses (if applicable)
	• Third-party security or monitoring tools (if part of enterprise standard)
	
	Provided by Service Provider
	• Design and diagramming tools
	• Assessment templates and frameworks
	• Reference architecture documentation
	• Landing zone design templates and patterns
	• Architecture decision record templates
	• Bill of Materials templates

Necessity of interaction

Level: HIGH

Customer Must Provide

- Designated project sponsor with decision authority
- Technical lead with deep application knowledge
- Access to existing assessment documentation and outputs
- Timely responses to design decisions (SLA: 3 business days)
- Workshop participation and review sessions
- Feedback on proposed architectures within defined review cycles

Workshop Participation Required From

Stakeholder	Involvement
CTO/CIO or delegate	Kick-off, key decision points, final sign-off
Application owners	All workshops, design validation
Platform / Cloud architects	All technical workshops
Security & Compliance teams	Security design workshops, compliance validation
Networking teams	Network design workshops
DevOps / Platform engineering	CI/CD and IaC integration workshops
Operations teams	Monitoring and operational design workshops
Database administrators	Data layer design sessions (if applicable)

Access Requirements

- Read access to assessment outputs and documentation
- Access to enterprise architecture artifacts and standards
- Documentation of current integration dependencies
- Application NFR documentation
- Compliance requirement documentation
- Network topology documentation (for hybrid scenarios)

Service inputs (parameters)

Parameter	Description	Required
Number of applications	Total count of applications to onboard to the landing zone	Yes
Application types	Classification of each application (web, API, batch, data processing, legacy)	Yes
Required environments	List of environments needed (dev/test/staging/prod)	Yes
Availability requirements	Uptime SLAs, RTO/RPO targets, HA requirements	Yes
Scalability requirements	Expected load patterns, scaling triggers, capacity projections	Yes
Performance requirements	Latency, throughput, and response time requirements	Yes
Data classification	Sensitivity levels and data handling requirements	Yes
Compliance requirements	Regulatory frameworks (GDPR, HIPAA, PCI-DSS, SOC2, etc.)	Yes
Integration dependencies	Systems, APIs, and services the application must integrate with	Yes
Preferred compute model	Target compute platform (VM / container / PaaS / serverless)	Yes

Target operating model	Desired operational responsibilities (central ops, DevOps, hybrid)	Recommended
Existing enterprise standards	Current naming conventions, tagging policies, design standards	Recommended
Network connectivity requirements	On-premises connectivity, ExpressRoute/Direct Connect needs	Recommended
Backup and retention requirements	Data protection policies and retention periods	Recommended
Cost constraints	Budget limitations or cost optimization priorities	Optional

Service outputs

1. Technical Architecture Design Document

- Executive summary and design overview
- Architecture principles and design decisions
- Assumptions and constraints
- Design standards and guidelines

2. Application Landing Zone Architecture Diagrams

- High-level architecture overview
- Detailed component architecture
- Network topology diagrams
- Data flow diagrams
- Integration architecture diagrams

3. Network and Security Design

- Network topology specification
- Subnet design and CIDR allocation
- Security group / firewall rules specification
- Private endpoint design
- Connectivity patterns (hybrid, internet, inter-VNet/VPC)

4. Identity and Access Model

- RBAC role definitions and assignments
- Managed identity / service account design
- Privileged access management design
- Authentication and authorization flows

5. Environment Separation Model

- Environment isolation strategy
- Resource naming and organization
- Environment promotion patterns
- Configuration management approach

6. Application Deployment Patterns

- Compute deployment specifications
- Scaling configuration design
- Container orchestration design (if applicable)
- PaaS service configurations

7. CI/CD and IaC Integration Design

- Infrastructure as Code module specifications
- Pipeline integration points
- Deployment automation patterns
- GitOps workflow design (if applicable)

8. Observability Design

- Logging architecture and log routing
- Monitoring strategy and metrics
- Alerting rules and thresholds
- Dashboard specifications

9. DR/BCP Design Summary

- Disaster recovery architecture
- Backup and restore procedures
- Failover and failback processes
- Business continuity considerations

10. Implementation Artifacts

- Design decisions & assumptions log

- Environment definition file (YAML/JSON)
- Detailed specification of all cloud resources (Bill of Materials)
- Detailed specification of all networks/subnets, CIDRs, sizing & topology
- Implementation roadmap and phasing recommendations

Expected Timeline

Size	Duration
Small (S)	2-3 weeks
Medium (M)	4-6 weeks
Large (L)	8-12 weeks

Breakdown by Phase

Phase	Small (S)	Medium (M)	Large (L)
Phase 1: Discovery & Requirements	2-3 days	4-5 days	7-10 days
Phase 2: Architecture Design	5-7 days	10-15 days	20-30 days
Phase 3: Security & Compliance Design	2-3 days	5-7 days	10-15 days
Phase 4: Documentation & Specifications	3-4 days	5-7 days	10-15 days
Phase 5: Review & Iteration	2-3 days	4-5 days	7-10 days
Phase 6: Presentation & Handover	1-2 days	2-3 days	3-5 days

Service Size Options

Size Determination Criteria

Criterion	Small (S)	Medium (M)	Large (L)
Number of applications	1	2-5	6+
Application components	Single tier	3-5 components	6+ tightly integrated
Compute model	PaaS only	Mix of containers and PaaS	Multi-model (VM, containers, PaaS)
Environment count	2 (Dev + Prod)	3 (Dev/Test/Prod)	4+ (Dev/Test/Staging/Prod/DR)
Network complexity	Standard VNet/VPC	Private networking, private endpoints	Hybrid connectivity, multi-region
Compliance requirements	Non-regulated	Moderate	High (finance, healthcare)
Availability requirements	Standard	Moderate HA	High HA, multi-region DR
Complexity	Low	Medium	High
Duration	2-3 weeks	4-6 weeks	8-12 weeks
Estimated Effort	40-60 hours	80-120 hours	160-240 hours
Team Size	1-2 resources	2-3 resources	3-4 resources

Summary Table

Size	Scope	Duration	Effort
S	Single application, PaaS-based, standard patterns, non-regulated	2-3 weeks	40-60 hours

M	Multi-component solution, mixed compute, private networking, moderate compliance	4-6 weeks	80-120 hours
L	Enterprise application portfolio, hybrid connectivity, multi-region HA/DR, regulated industry	8-12 weeks	160-240 hours

Sizing Parameters

Scale Parameters

Number of Applications

- 1 application: Size S
- 2-5 applications: Size M
- 6+ applications: Size L

Application Components

- Single tier: Size S
- 3-5 components: Size M
- 6+ tightly integrated components: Size L

Compute Model

- PaaS only: Size S
- Mix of containers and PaaS: Size M
- Multi-model (VM, containers, PaaS, serverless): Size L

Environment Count

- 2 environments (Dev + Prod): Size S
- 3 environments (Dev/Test/Prod): Size M
- 4+ environments (Dev/Test/Staging/Prod/DR): Size L

Network Complexity

- Standard VNet/VPC: Size S
- Private networking with private endpoints: Size M
- Hybrid connectivity, multi-region: Size L

Compliance Requirements

- Non-regulated: Size S
- Moderate compliance (SOC2, ISO 27001): Size M
- High compliance (HIPAA, PCI-DSS, financial regulations): Size L

Availability Requirements

- Standard (99.5% SLA): Size S
- Moderate HA (99.9% SLA): Size M
- High HA with multi-region DR (99.99% SLA): Size L

Technical Parameters

Hybrid Connectivity

- No hybrid connectivity (baseline): -
- ExpressRoute/Direct Connect required: +16 hours

Multi-Region Deployment

- Single region (baseline): -
- Multi-region with DR: +24 hours

Container Orchestration

- No containers (baseline): -
- Kubernetes (AKS/EKS/GKE) design: +16 hours

Service Mesh Integration

- No service mesh (baseline): -
- Service mesh design (Istio, Linkerd): +12 hours

Customer-Managed Keys (CMK)

- Platform-managed encryption (baseline): -

- CMK with Key Vault/KMS design: +8 hours

API Management

- No API gateway (baseline): -
- API Management design (APIM, API Gateway): +8 hours

GitOps Implementation

- Standard CI/CD integration (baseline): -
- Full GitOps workflow design: +12 hours

Scope Dependencies

Container Platform Design

- Requires: Compute Architecture, Network Architecture

DR/BCP Architecture

- Requires: Platform Architecture, Data Architecture, Network Architecture

CI/CD Integration Design

- Requires: Environment Separation Model, Compute Architecture

Security Architecture (Micro-segmentation)

- Requires: Network Architecture, Identity & Access Management

Observability Architecture

- Requires: Platform Architecture, Compute Architecture

Data Architecture (Cross-region replication)

- Requires: DR/BCP Design, Network Architecture

Effort Estimation

Scope Area Effort

Platform Architecture

- Base Hours: 8
- Notes: Subscription/account structure, resource organization, environment separation

Network Architecture

- Base Hours: 12
- Notes: VNet/VPC design, subnets, routing, DNS zones, traffic flow

Security Architecture

- Base Hours: 16
- Notes: Security controls, segmentation, encryption, key management

Identity & Access Management

- Base Hours: 12
- Notes: RBAC model, managed identities, privileged access, federation

Compute Architecture

- Base Hours: 10
- Notes: Compute model selection, scaling patterns, container design

Data Architecture

- Base Hours: 10
- Notes: Database design, storage, backup policies, data lifecycle

Integration Architecture

- Base Hours: 8
- Notes: CI/CD integration points, API patterns, event-driven design

Observability Architecture

- Base Hours: 8
- Notes: Logging, monitoring, alerting, dashboards

Resilience & Continuity

- Base Hours: 12
- Notes: DR/BCP design, HA patterns, failover procedures

Governance Framework

- Base Hours: 6
- Notes: Naming conventions, tagging, policies, compliance mapping

Documentation & Handover

- Base Hours: 10
- Notes: Bill of Materials, design decisions log, environment definitions

Technical Complexity Additions

Hybrid connectivity (ExpressRoute/Direct Connect)

- Condition: On-premises integration required
- Hours Added: +16

Multi-region deployment

- Condition: Active-active or active-passive cross-region
- Hours Added: +24

Container orchestration (AKS/EKS/GKE)

- Condition: Kubernetes-based workloads
- Hours Added: +16

Service mesh integration

- Condition: Istio, Linkerd, or similar required
- Hours Added: +12

Customer-managed keys (CMK)

- Condition: BYOK or CMK encryption required
- Hours Added: +8

API Management design

- Condition: API gateway/management layer required
- Hours Added: +8

GitOps workflow design

- Condition: Full GitOps with ArgoCD/Flux required
- Hours Added: +12

Legacy application integration

- Condition: Complex integration with legacy systems
- Hours Added: +12

Multi-cloud design patterns

- Condition: Cross-cloud consistency requirements
- Hours Added: +20

Additional compliance framework

- Condition: Per additional framework (HIPAA, PCI-DSS, etc.)
- Hours Added: +8 per framework

Complex data residency requirements

- Condition: Data sovereignty with multiple jurisdictions
- Hours Added: +10

Sizing Examples

S – Single PaaS Web Application Landing Zone

Scenario: A mid-size company needs to deploy a new customer-facing web application to Azure. The application uses Azure App Service with Azure SQL Database and requires only production and development environments. No regulatory compliance requirements apply.

Characteristics:

- Single web application with managed database
- Uses standard enterprise landing zone patterns
- PaaS services only (App Service, Azure SQL, Storage Account)
- Non-regulated workload
- Dev + Prod environments only
- Standard RBAC model
- No hybrid connectivity required
- Standard monitoring and logging
- Single-region deployment

Deliverables: Architecture design document, landing zone diagrams, network design, RBAC model, environment specifications, IaC integration points, monitoring design.

M – Multi-Component Container-Based Solution

Scenario: A financial services company is deploying a new payment processing solution consisting of a web frontend, multiple API services, and a data processing component. The solution requires containerized deployment with private networking, moderate compliance requirements, and three distinct environments.

Characteristics:

- 3-5 application components forming an integrated solution
- Mix of containers (AKS/EKS) and PaaS services
- Private networking with private endpoints
- Dev/Test/Prod environments
- Moderate availability requirements (99.9% SLA)
- PCI-DSS considerations for payment data
- Service mesh or API gateway integration
- Enhanced monitoring and alerting
- Backup and recovery design
- CI/CD pipeline integration points

Deliverables: Comprehensive architecture document, detailed component diagrams, network security design with private endpoints, container platform design, identity model with workload identities, compliance control mapping, environment promotion strategy, observability design, DR considerations.

L – Enterprise Regulated Application Portfolio

	Scenario: A large healthcare organization is migrating a suite of patient management applications to AWS. The solution includes 8 interconnected applications with hybrid connectivity to on-premises systems, strict HIPAA compliance requirements, multi-region disaster recovery, and complex integration patterns. **Characteristics:** • 6+ tightly integrated application components • Hybrid connectivity to on-premises data centers (Direct Connect/ExpressRoute) • Multi-region deployment for high availability • Cross-region disaster recovery architecture • HIPAA regulated with BAA requirements • Complex RBAC with fine-grained access control • Comprehensive auditing and logging requirements • Data encryption with customer-managed keys • PHI data handling and protection design • Multiple environment tiers (Dev/Test/Staging/Prod/DR) • Complex integration with existing enterprise systems • Advanced monitoring with compliance dashboards **Deliverables:** Enterprise architecture document, multi-region topology diagrams, comprehensive security architecture, HIPAA compliance mapping, detailed RBAC model, data protection design, encryption key management, hybrid connectivity design, advanced DR/BCP architecture, detailed specifications for all resources, complete IaC module specifications, operational handover package.

Responsible role(s)

Role	Responsibility
Cloud Architect (Primary Owner)	Overall architecture design, technical leadership, design decisions, stakeholder alignment, quality assurance of all deliverables, architecture pattern selection, cross-domain design integration
Security Architect	Security architecture design, compliance mapping, identity and access model, encryption design, security control specifications
Network Architect	Network topology design, connectivity patterns, firewall rules, DNS design, hybrid connectivity architecture
Project Manager	Timeline management, resource coordination, stakeholder communication, workshop scheduling, deliverable tracking, risk management

Team Allocation

Team Composition by Size

Small (S) - 1-2 resources, 40-60 hours

- Cloud Architect (Primary Owner): 0.8 FTE
- Security Architect: 0.2 FTE
- Network Architect: 0.1 FTE (as needed)
- Project Manager: 0.1 FTE

Medium (M) - 2-3 resources, 80-120 hours

- Cloud Architect (Primary Owner): 1.0 FTE
- Security Architect: 0.5 FTE
- Network Architect: 0.3 FTE
- Project Manager: 0.2 FTE

Large (L) - 3-4 resources, 160-240 hours

- Cloud Architect (Primary Owner): 1.0 FTE
- Security Architect: 0.8 FTE
- Network Architect: 0.6 FTE
- Project Manager: 0.4 FTE
- Additional Cloud Architect (Support): 0.5 FTE

Multi-Cloud Considerations	When designing application landing zones that may span multiple cloud providers or require consistency across different cloud implementations: • **Design Pattern Abstraction:** Create cloud-agnostic design patterns that can be implemented with provider-specific services • **Naming Convention Alignment:** Establish naming standards that work across all target cloud platforms • **Identity Federation:** Design for centralized identity with federation to multiple cloud providers • **Network Connectivity:** Consider multi-cloud network fabrics and consistent IP address management • **Monitoring Consolidation:** Design for centralized observability across cloud boundaries • **IaC Consistency:** Use Terraform or Pulumi for cross-cloud infrastructure code consistency • **Security Baseline:** Establish equivalent security controls mapped across cloud providers