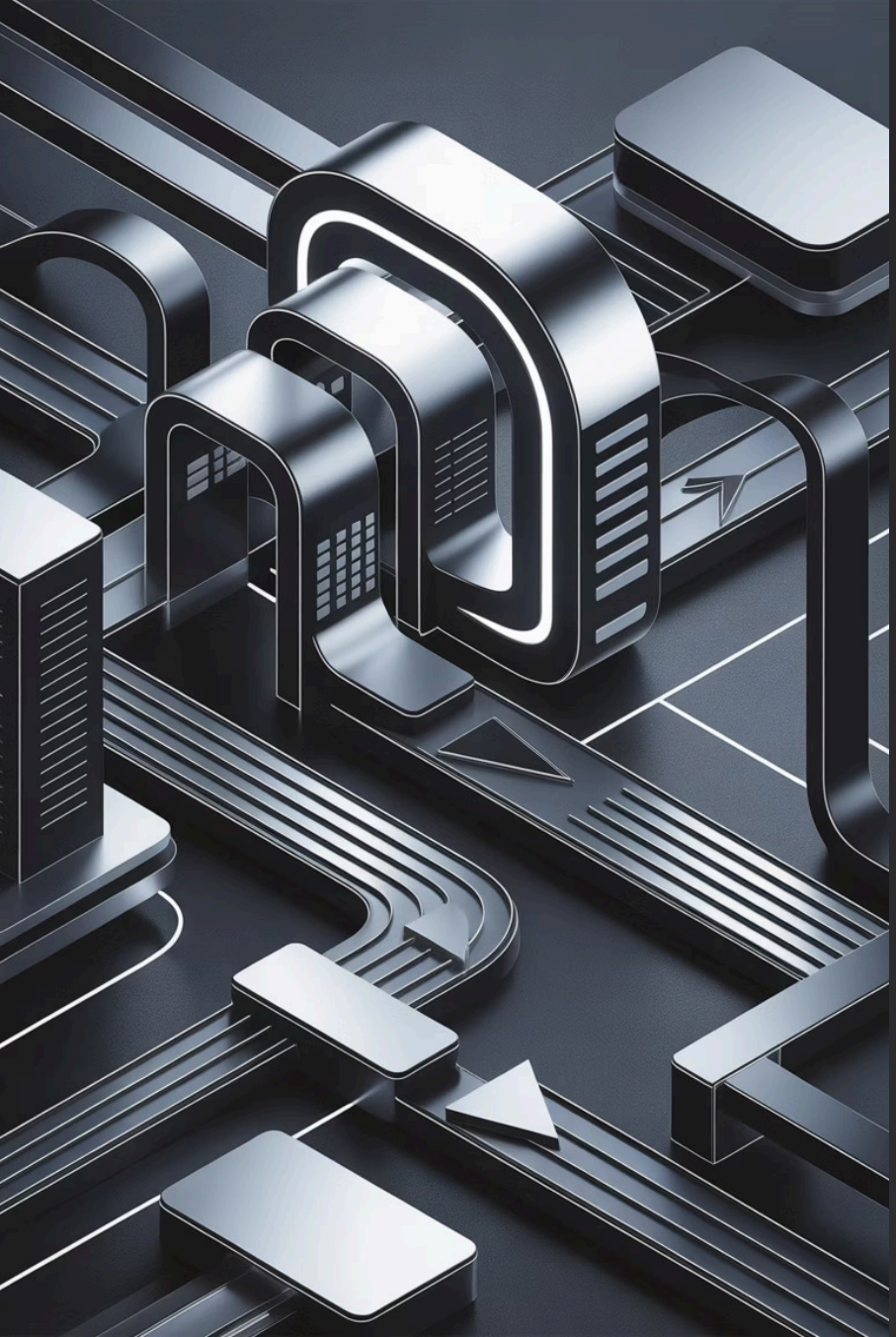
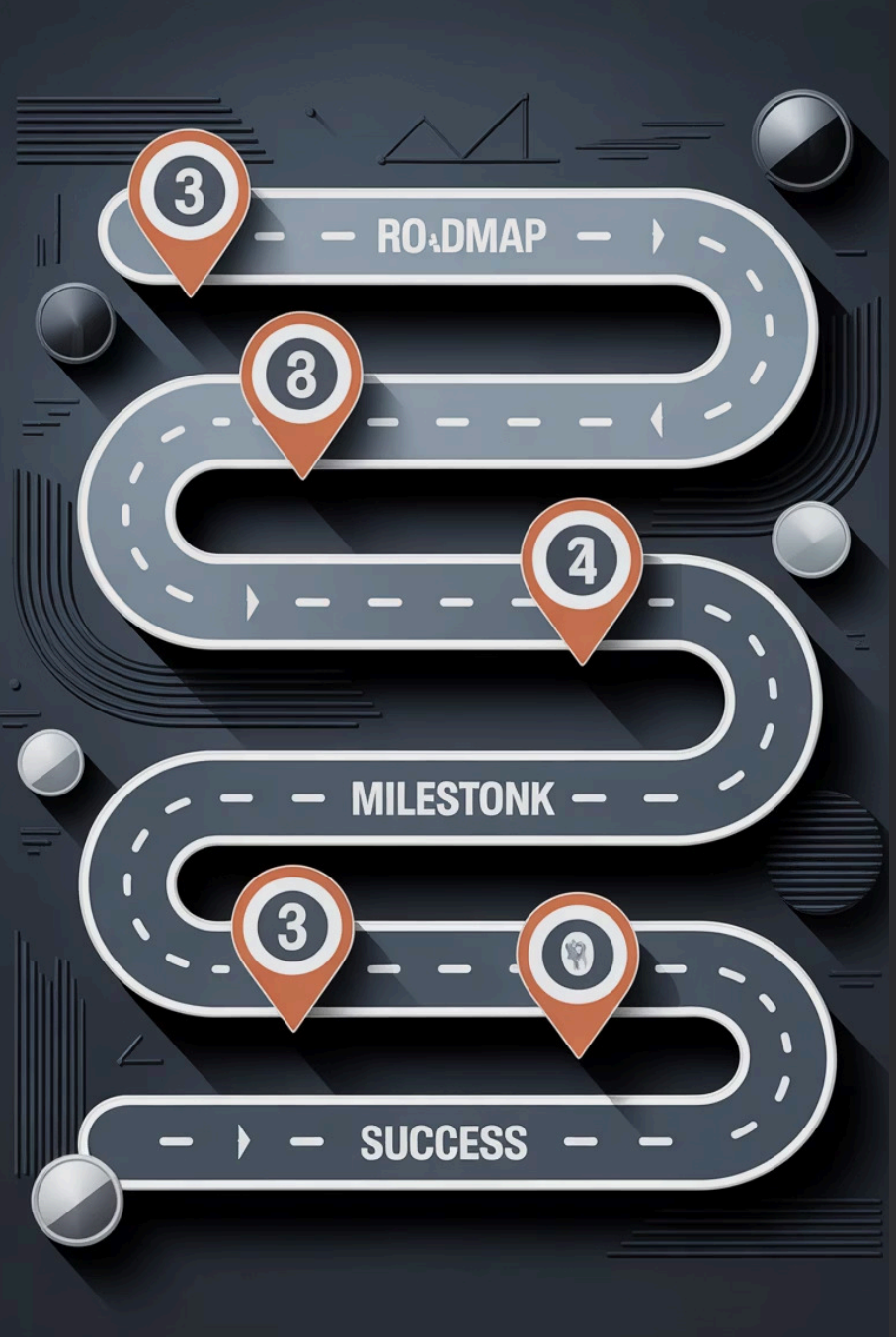


Azure API Management Deep Dive

A comprehensive 12-hour course designed to transform you from an Azure-aware developer into an APIM expert. Master configuration, security, monitoring, and enterprise-grade API platform management across three intensive sessions.



Course Navigation

Your Learning Journey

This course spans three 4-hour sessions covering everything from foundational concepts to advanced enterprise patterns. Today's session focuses on APIM fundamentals, architecture, and how it compares to alternative solutions. You'll gain hands-on experience with the core concepts that underpin every APIM deployment.

By the end of this course, you'll have been exposed to the knowledge required to confidently design, deploy, and manage production-grade API platforms in Azure, complete with security, monitoring, and disaster recovery capabilities.

What You'll Master

Architecture & Configuration

Understand APIM components, SKUs, deployment models, and how to architect solutions that scale from development to enterprise production environments.

Security & Access Control

Implement OAuth 2.0, JWT validation, certificate-based authentication, and network-level security with vNET integration and private endpoints.

APIOps & Automation

Deploy infrastructure as code, automate policy updates, implement CI/CD pipelines, and manage APIM configurations across multiple environments efficiently.

Observability & Troubleshooting

Configure comprehensive logging with Application Insights, set up alerts, trace requests across distributed systems, and diagnose production issues quickly.

Three-Day Agenda Overview

Day 1: Foundations & Core Concepts

1

APIM architecture, policy configuration, backends, API importing, debugging, caching, versioning, and exports. Build a solid foundation for everything that follows.

2

Day 2: Security & Operations

API security patterns, developer portal configuration, networking considerations, custom domains, SSL certificates, and comprehensive logging and monitoring strategies.

Day 3: Enterprise Patterns

3

APIOps automation, CI/CD integration, resilience patterns, disaster recovery planning, Traffic Manager integration, and Azure Front Door configuration for global distribution.

Prerequisites

What You Should Know

Technical Foundation

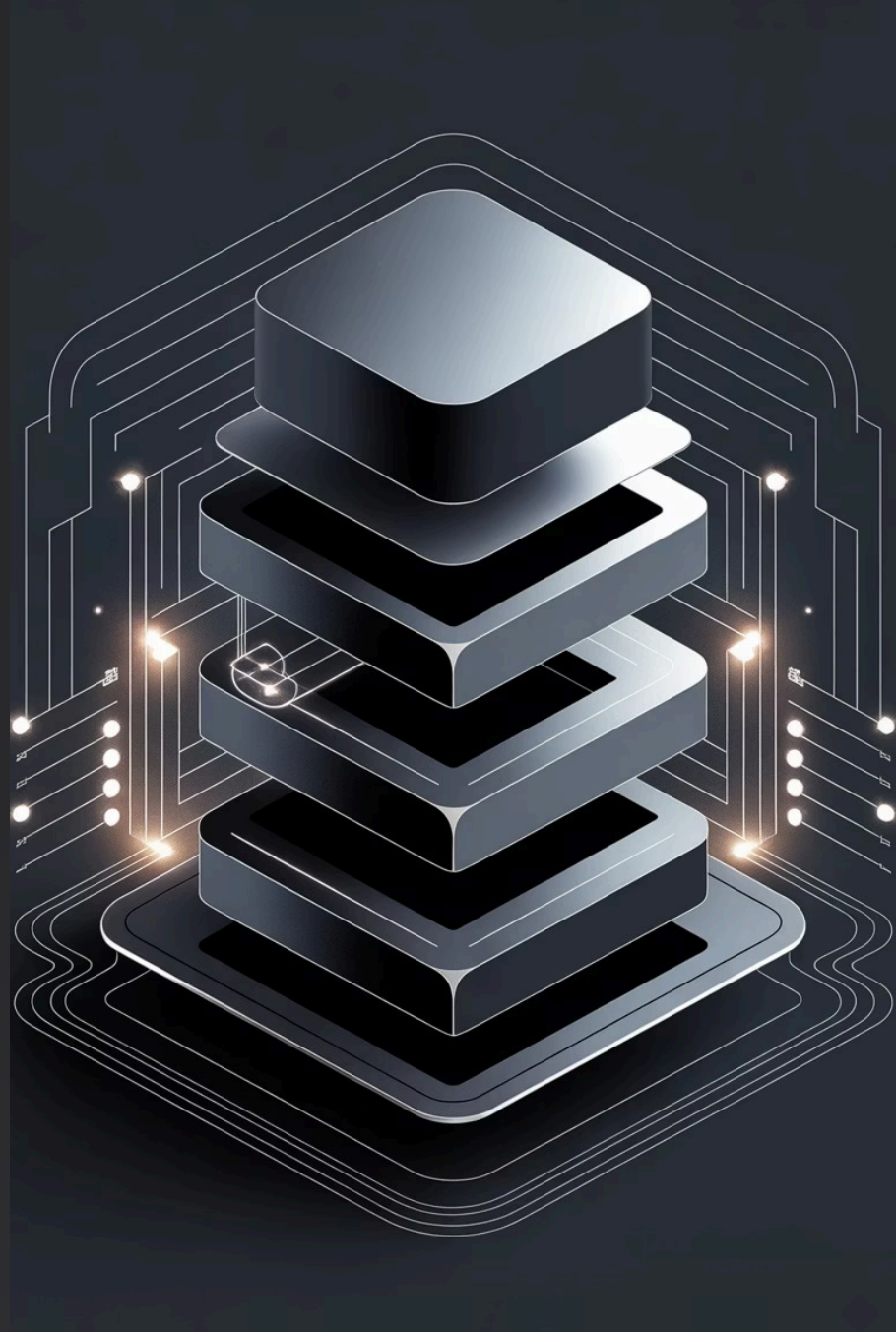
- Experience developing and consuming RESTful APIs
- Familiarity with Azure portal and basic resource management
- Understanding of HTTP methods, status codes, and headers
- Basic knowledge of JSON and XML data formats
- Exposure to authentication concepts like OAuth 2.0 and JWT

Azure Experience

- Comfort navigating Azure subscriptions and resource groups
- Basic understanding of Azure networking concepts
- Familiarity with Azure Active Directory fundamentals
- Experience with at least one Azure compute service (App Service, Functions, or Container Apps)

Module 1.1: Diving into APIM Fundamentals

We'll start by building your foundational understanding of Azure API Management, exploring its core components and where it fits in a modern cloud architecture.



Why API Gateways Matter

API gateways serve as the central entry point for all API traffic, abstracting backend complexity from consumers. They provide a unified interface that enforces consistent security, routing, transformation, and monitoring across your entire API estate.

Without a gateway, each backend service must implement its own authentication, rate limiting, logging, and error handling. This creates duplication, inconsistency, and increased maintenance burden. An API gateway centralizes these cross-cutting concerns, allowing backend teams to focus on business logic while the gateway handles infrastructure concerns.

Common API Gateway Use Cases

Legacy Modernization

Expose legacy SOAP services as modern REST APIs without modifying backend systems. Transform XML to JSON, add OpenAPI specifications, and gradually migrate consumers to new contracts.

Microservices Facade

Present a unified API surface over dozens of microservices. Aggregate responses, orchestrate multi-service calls, and shield consumers from internal service boundaries and versioning complexity.

Partner Integration

Create secure, rate-limited API products for external partners. Manage multiple subscription tiers, enforce quotas, provide self-service developer portals, and monitor usage for billing purposes.

Mobile Backend

Optimize APIs for mobile clients with response compression, caching, and payload transformation. Implement device-specific routing, version negotiation, and specialized authentication flows for iOS and Android apps.

Platform Architecture

Azure API Management Components

Azure API Management consists of four primary components working together to provide a complete API platform. Each component serves a distinct purpose in the API lifecycle, from design and deployment through operation and consumption.

Core APIM Components

API Gateway

The runtime proxy that accepts API calls and routes them to backends. Enforces policies, transforms requests and responses, caches results, and collects telemetry. Available as managed gateway or self-hosted for hybrid deployments.

Management Plane

Azure portal interface and REST API for configuring APIs, policies, products, and users. Provides role-based access control, audit logging, and integration with Azure Resource Manager for infrastructure automation.

Developer Portal

Customizable self-service website where API consumers discover APIs, read documentation, test endpoints interactively, and manage their subscription keys. Fully managed with built-in CMS capabilities.

Self-Hosted Gateway

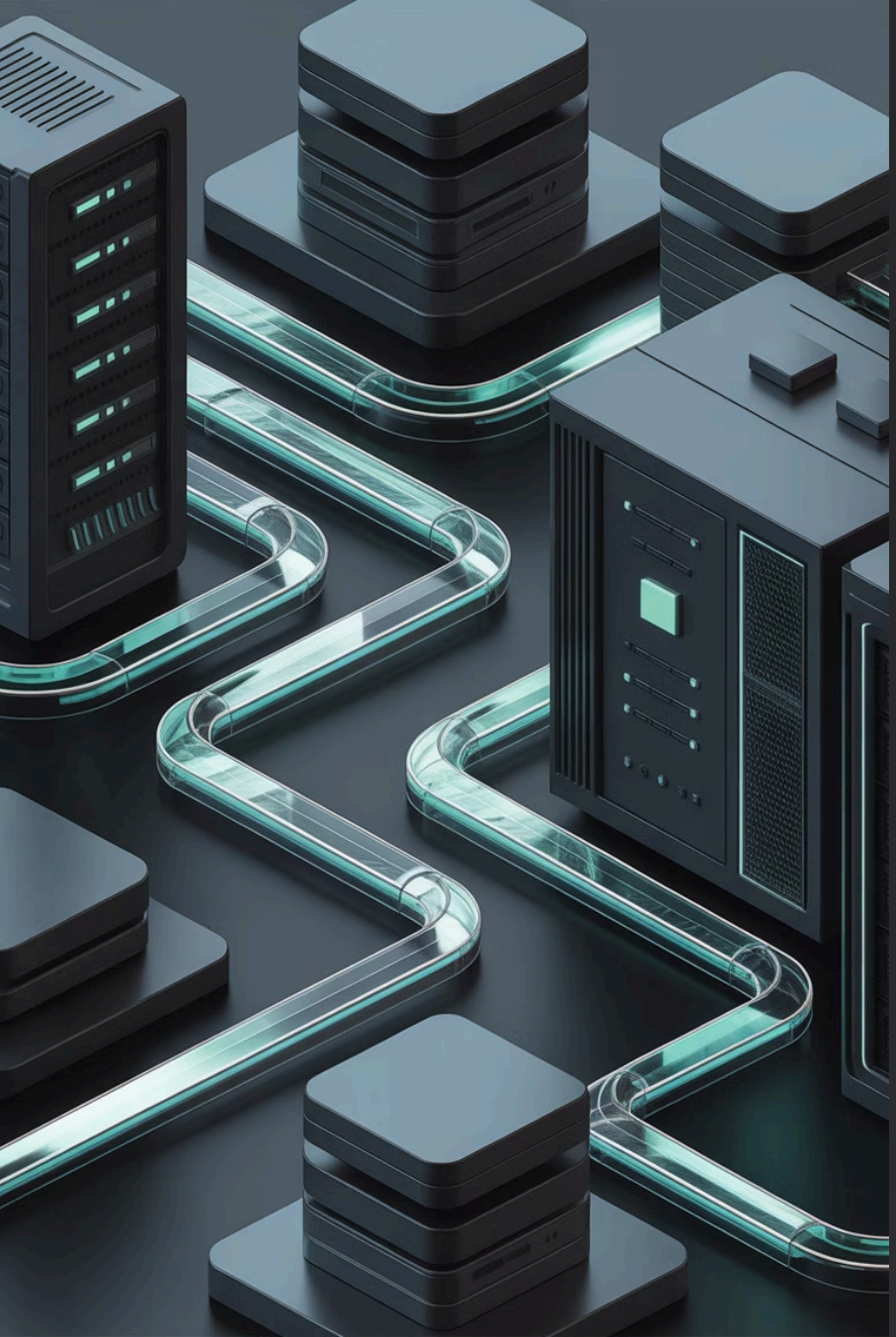
Containerized gateway runtime deployable to any Kubernetes cluster, on-premises data center, or other cloud. Maintains connectivity to Azure management plane while processing requests locally for latency-sensitive or compliance-driven scenarios.

Understanding APIM SKUs

Azure API Management offers multiple service tiers, each designed for different workload characteristics and organizational maturity levels. Choosing the right SKU impacts performance, availability, networking capabilities, and cost structure.

Consumption tier provides serverless scaling with pay-per-execution pricing, ideal for development and low-traffic scenarios. **Basic and Standard tiers** offer dedicated infrastructure with SLA guarantees suitable for production workloads. **Premium tier** adds multi-region deployment, vNET integration, and higher throughput for enterprise scenarios.

The **Developer tier** includes all features for non-production use at a reduced cost, without SLA guarantees. It's perfect for testing advanced features like vNET integration before committing to premium pricing.



Deployment Models

vNET Integration Scenarios

Azure API Management supports three networking modes: **External** mode exposes both gateway and management endpoints to the public internet. **Internal** mode makes all endpoints accessible only within the vNET, requiring private connectivity. **External with internal backend** exposes the gateway publicly while keeping backend APIs in private networks.

vNET integration enables secure connectivity to private backends, integration with on-premises systems via ExpressRoute or VPN, and compliance with data residency requirements. Available in Developer and Premium tiers only.

When to Choose Each Model

Public (No vNET)

APIs consumed by public internet users. SaaS products. Partner integrations. No private backend requirements.

External vNET

Public API exposure with private backends. On-prem integration. Firewall requirements. Hybrid cloud scenarios.

Internal vNET

Internal-only APIs. Regulatory compliance. Zero trust architecture. Private endpoint requirements.

Competitive Landscape

Azure APIM vs AWS API Gateway

Both platforms provide API gateway capabilities, but with different architectural philosophies and operational models. Understanding these differences helps you architect solutions that leverage each platform's strengths.

Azure APIM emphasizes policy-based configuration, rich transformation capabilities, and enterprise-grade features like self-hosted gateways and built-in developer portals. AWS API Gateway focuses on serverless integration, per-request pricing, and tight coupling with Lambda functions.

Ecosystem Positioning

Azure API Management

Integrated deeply with Azure Active Directory, Application Insights, and Event Hubs. First-class support for Azure Functions, App Service, and Logic Apps as backends. Strong focus on hybrid scenarios with self-hosted gateways connecting Azure to on-premises and multi-cloud environments.

AWS API Gateway

Tightly coupled with AWS Lambda for serverless architectures. Native integration with Cognito for authentication, CloudWatch for logging, and X-Ray for tracing. Emphasizes event-driven patterns and pay-per-use economics typical of AWS services.

Pricing & Scaling Models

Azure APIM

- **Consumption:** Pay per execution (first 1M calls free monthly)
- **Basic/Standard/Premium:** Fixed monthly cost based on units deployed
- **Scaling:** Manual scale-out by adding units, or autoscale in Premium
- **Predictability:** Fixed costs for sustained traffic patterns
- **Economics:** More cost-effective at high, consistent volumes

AWS API Gateway

- **REST API:** $3.50 per million requests (decreases with volume)
- **HTTP API:** $1.00 per million requests (simplified feature set)
- **Scaling:** Automatic, transparent scaling to any request volume
- **Predictability:** Variable costs directly tied to usage
- **Economics:** More cost-effective for unpredictable or bursty traffic

Configuration Philosophy

Azure APIM uses a **policy-based configuration model** where XML policies define behavior at various scopes (global, product, API, operation). Policies execute in a pipeline with inbound, backend, outbound, and on-error sections. This declarative approach enables powerful transformations and orchestrations without code.

AWS API Gateway uses a **stage-based configuration model** where each API has multiple stages (dev, test, prod) with stage-specific settings. Request/response transformations use Velocity Template Language (VTL). Integration requests connect to backends via HTTP, Lambda, AWS services, or VPC links.

Policy vs Stage Configuration

APIM Policies

Scope hierarchy: Global → Product → API → Operation, with inheritance and override capabilities

Pipeline execution: Sequential processing through inbound, backend, outbound, on-error sections

Transformation power: Advanced XML/JSON manipulation, C# expressions, built-in policy library

Reusability: Policy fragments shareable across APIs

API Gateway Stages

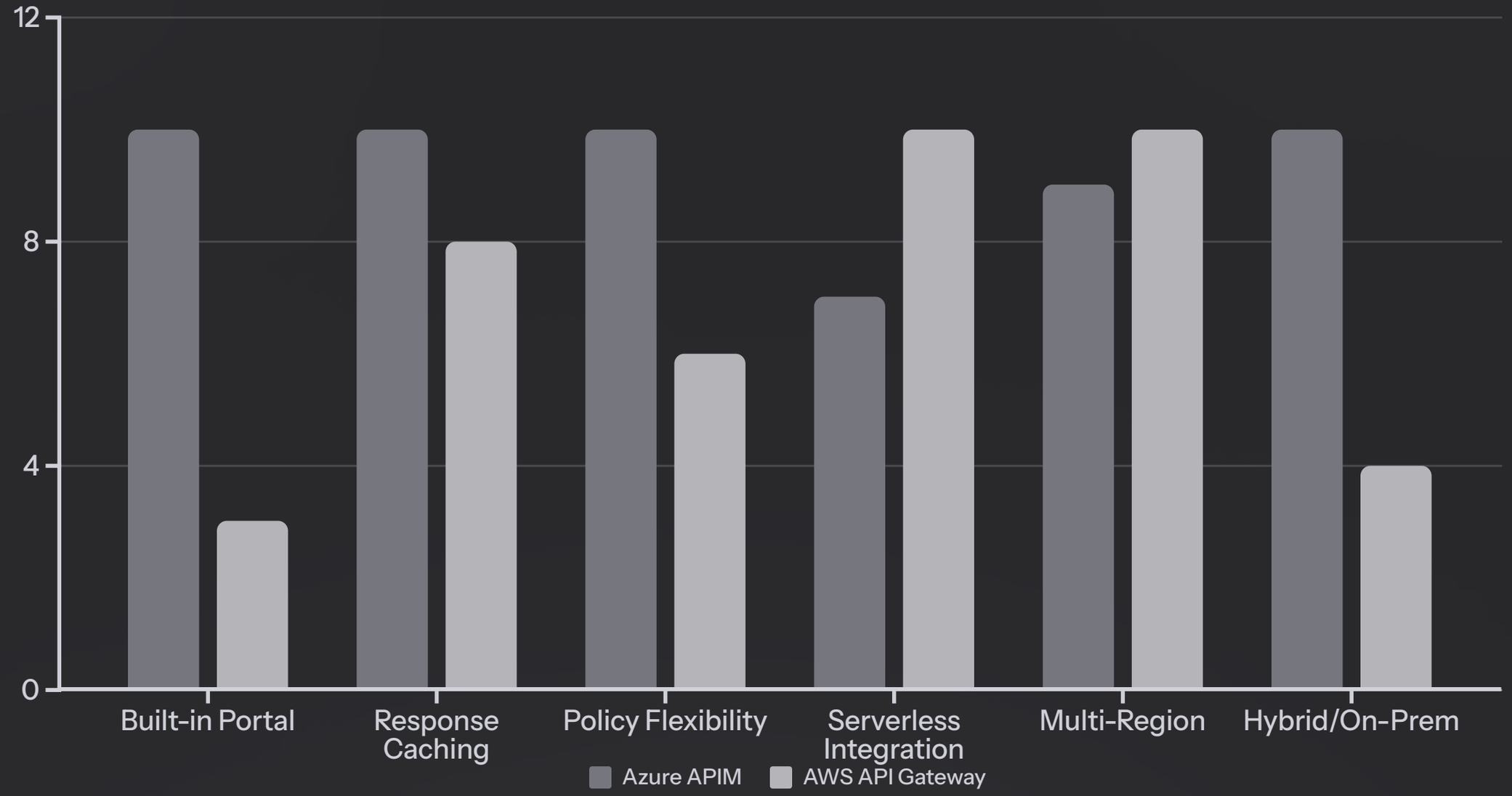
Environment isolation: Separate stages with independent configurations, throttling, caching

Integration types: Lambda, HTTP, Mock, AWS Service, VPC Link with method-level mappings

Mapping templates: VTL for request/response transformations at integration and method levels

Deployment model: Immutable deployments promoted through stages

Feature Comparison Highlights



Ratings are relative (1-10 scale) comparing platform strengths. APIM excels at enterprise features and hybrid scenarios, while API Gateway leads in serverless integration and global edge deployment.

Key Differentiators

What Makes APIM Unique

[ ]

Fully Managed Developer Portal

Out-of-the-box portal with interactive documentation, API testing console, and subscription management. No additional development required.

Rich Transformation Engine

Advanced XML and JSON manipulation, C# expression evaluation, built-in policy library for common patterns like rate limiting, JWT validation, and CORS.

Hybrid Gateway Deployment

Self-hosted gateway containers deployable anywhere maintain full policy enforcement while processing traffic locally for compliance and latency requirements.

API Versioning & Revisions

First-class support for versioning schemes (path, header, query) and safe revisions for testing changes before making them current without impacting consumers.

Module Summary

What We Covered

Course Framework

12-hour journey through APIM from basics to enterprise patterns

APIM Architecture

Four core components and how they work together

Platform Comparison

How APIM differs from AWS API Gateway

SKUs & Deployment

Choosing the right tier and networking model

Key Takeaways

APIM is a comprehensive API platform

Not just a gateway—includes management plane, developer portal, and self-hosted options for complete API lifecycle management from design through retirement.

SKU selection impacts capabilities

Consumption offers serverless economics, Premium enables enterprise features like vNET and multi-region. Match tier to your performance, networking, and SLA requirements.

Policy-based configuration is powerful

Azure's policy approach provides more flexibility than stage-based models. Declarative policies enable sophisticated transformations and orchestrations without custom code.

Platform choice depends on ecosystem

Choose APIM for Azure-centric and hybrid scenarios with complex policies. Choose API Gateway for AWS Lambda-heavy serverless architectures with simpler requirements.

Demo

Thank You for Attending!