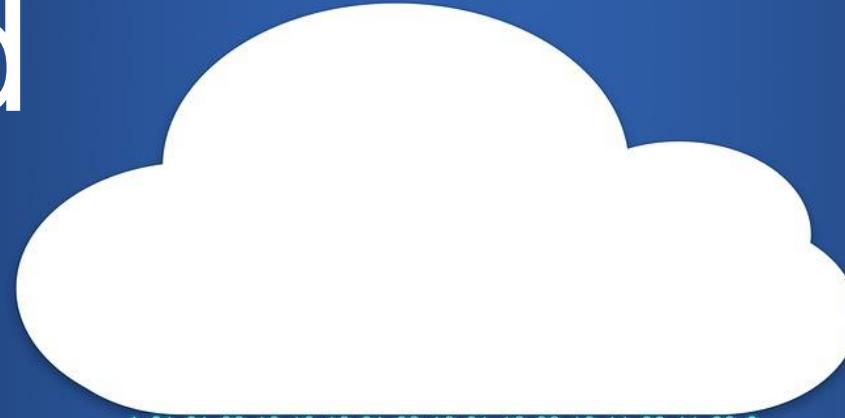


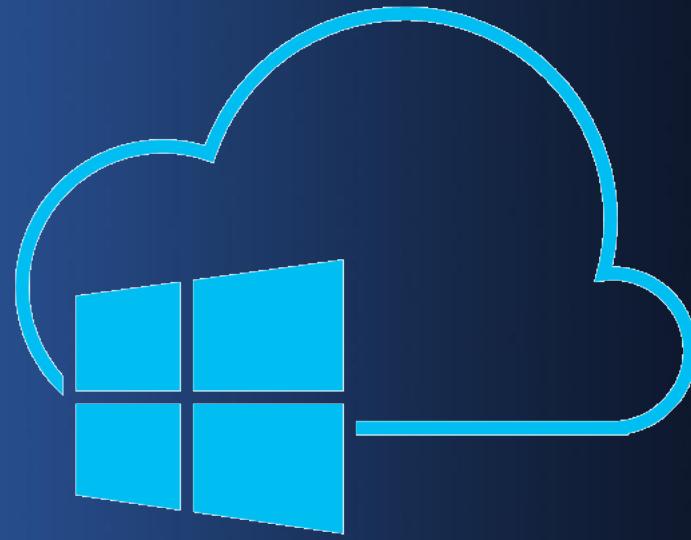
Advanced

Azure



```
1 01 01 00 10 10 10 01 00 10 01 10 00 10 11 00 11 00 0  
1 01 11 11 10 11 00 11 10 10 01 10 10 11 01 10 11 10 0  
0 10 10 10 10 10 10 00 11 00 00 00 00 01 01 11 10 10 0  
0 11 01 11 00 00 01 00 11 11 00 11 11 01 10 00 10 00 1  
0 00 11 00 01 11 01 10 10 10 01 00 11 11 11 01 01 11 0  
1 11 10 01 01 00 01 01 00 01 00 00 11 00 00 01 10 0  
1 10 01 01 11 01 00 10 00 00 10 00 01 10 11 00 10 1  
1 00 10 00 01 00 10 11 01 00 00 10 10 10 01 10 01 10 0  
1 10 11 00 01 11 00 10 01 11 11 11 11 10 11 01 01 1  
0 00 11 00 11 10 10 00 00 11 01 11 10 11 01 10 01 01 0  
1 01 00 01 01 00 01 10 01 00 00 11 11 00 11 00 10 01 0  
1 10 10 00 10 10 10 11 10 01 10 10 01 11 00 11 00 10 0  
0 00 01 10 01 00 00 11 00 11 10 00 10 10 00 11 00 10 1  
1 01 11 01 01 10 11 00 00 11 10 01 10 00 00 01 10 01 1  
0 11 00 11 01 00 01 11 11 10 10 10 11 11 11 00 00 10 0  
1 11 01 10 11 11 01 00 01 01 00 00 01 11 00 00 00 10 0  
1 11 01 01 10 11 01 11 11 11 10 11 00 11 10 01 01 1  
0 01 01 00 10 11 11 00 01 01 00 10 10 00 00 00 11 11 1  
0 10 11 01 00 01 11 10 11 10 01 01 11 10 00 01 00 00 0  
0 01 11 01 10 10 01 11 00 11 00 00 01 10 01 00 01 10 1  
1 00 01 00 11 11 10 10 11 11 01 11 10 11 10 11 11 1  
1 10 11 11 10 00 10 01 00 11 11 00 11 01 10 10 00 0  
1 11 10 11 10 10 00 10 10 00 10 01 11 01 11 10 00 11 0  
1 10 01 00 10 01 11 00 10 00 10 01 01 01 11 10 10 00 1  
0 10 00 11 10 10 11 01 00 10 10 11 00 10 01 11 11 01 1  
0 00 01 11 00 01 11 00 10 00 10 00 01 01 11 01 11 00 0  
1 11 11 01 00 10 00 11 10 10 00 11 00 00 10 11 10 10 0  
1 01 11 11 11 01 00 11 01 11 00 11 01 10 00 00 01 1  
0 11 00 00 11 01 01 11 01 00 11 10 11 01 10 10 1  
1 10 10 11 00 00 10 00 10 10 00 11 00 11 10 11 10 1  
1 10 01 10 11 00 01 01 01 10 10 10 10 11 11 10 01 01 0  
1 11 10 01 00 11 01 01 11 00 10 11 00 11 01 10 00 01 1  
0 01 10 00 01 10 11 10 01 00 00 11 00 11 01 00 01 00 1  
1 11 10 11 10 00 00 11 00 11 00 00 11 11 01 11 01 11 0  
0 00 00 01 10 01 01 10 11 11 10 10 01 01 00 11 11 00 0
```

Develop a passion for learning.





WORKFORCE DEVELOPMENT



PARTICIPANT GUIDE



Content Usage Parameters

Content refers to material including instructor guides, student guides, lab guides, lab or hands-on activities, computer programs, etc. designed for use in a training program

1

Content is subject to
copyright protection

2

Content may only be
leveraged by students
enrolled in the training
program

3

Students agree not to
reproduce, make
derivative works of,
distribute, publicly perform
and publicly display
content in any form or
medium outside of the
training program

4

Content is intended as
reference material only to
supplement the instructor-
led training

DAY 1 - COURSE REVIEW

Azure Core Infrastructure

Dashboard & Cloud Services: Navigated the portal, examined core components like VNets, compute, and storage.

Resource Management: Built understanding of resource groups, providers, ARM templates, and automation strategies.

VM Deployment & Networking

Explored **multi-region deployments, scaling strategies** (scale-up vs. scale-out), and high availability features like **Availability Sets and Scale Sets**.

Hands-on labs included setting up VMs across subnets and regions and testing connectivity via NSGs and peering.

Containers & Orchestration

Introduced **Azure Container Service**, especially **AKS**, and integrated it into DevOps workflows.

Demonstrated container deployment, registry setup, and CI/CD pipeline integration.

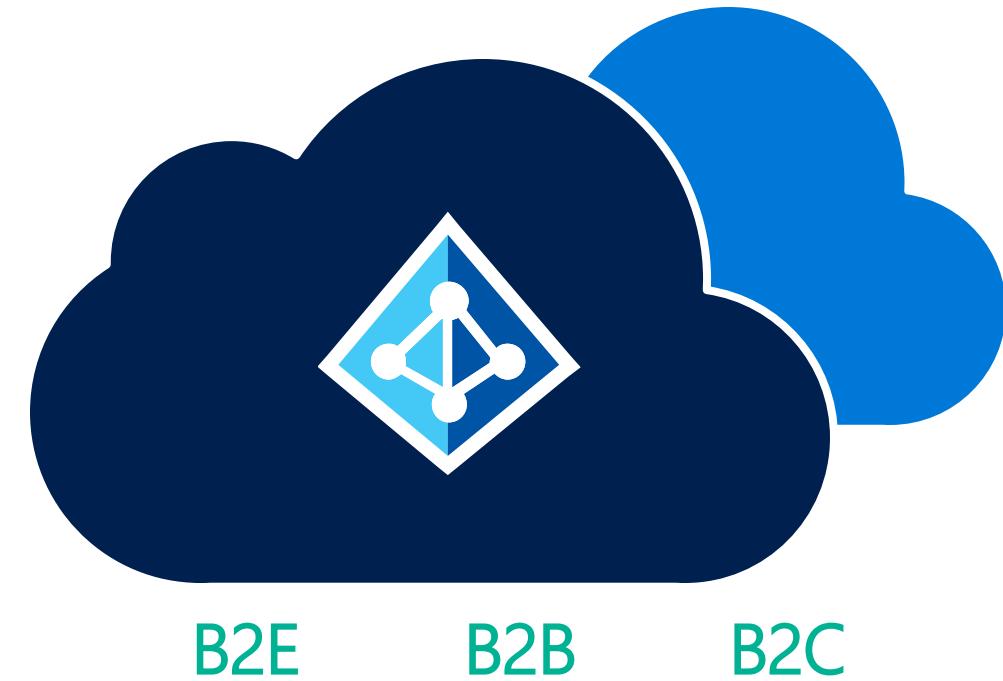


Microsoft Entra ID

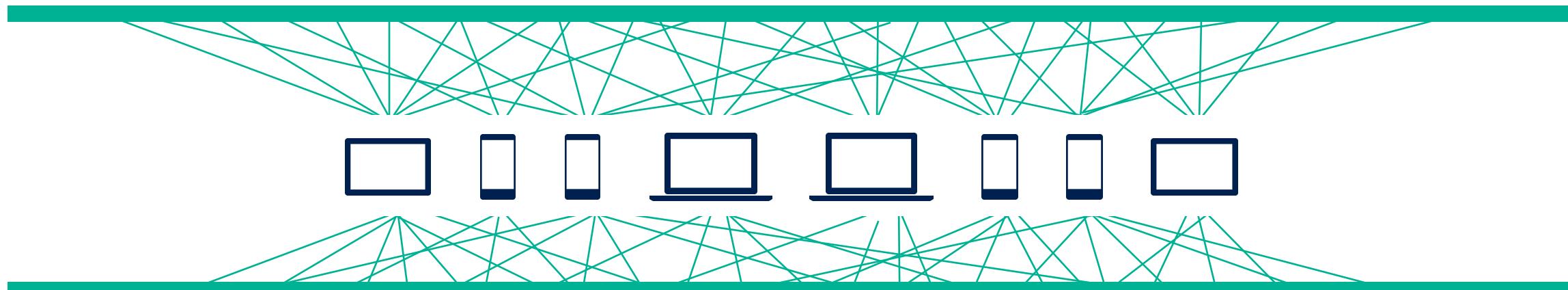
WHAT IS MICROSOFT ENTRA ID (FORMERLY MICROSOFT ACTIVE DIRECTORY)?

A comprehensive identity and access management cloud solution for your employees, partners, and customers.

It combines directory services, advanced identity governance, application access management, and a rich standards-based platform for developers.



CURRENT REALITY



On-premises



Managed devices

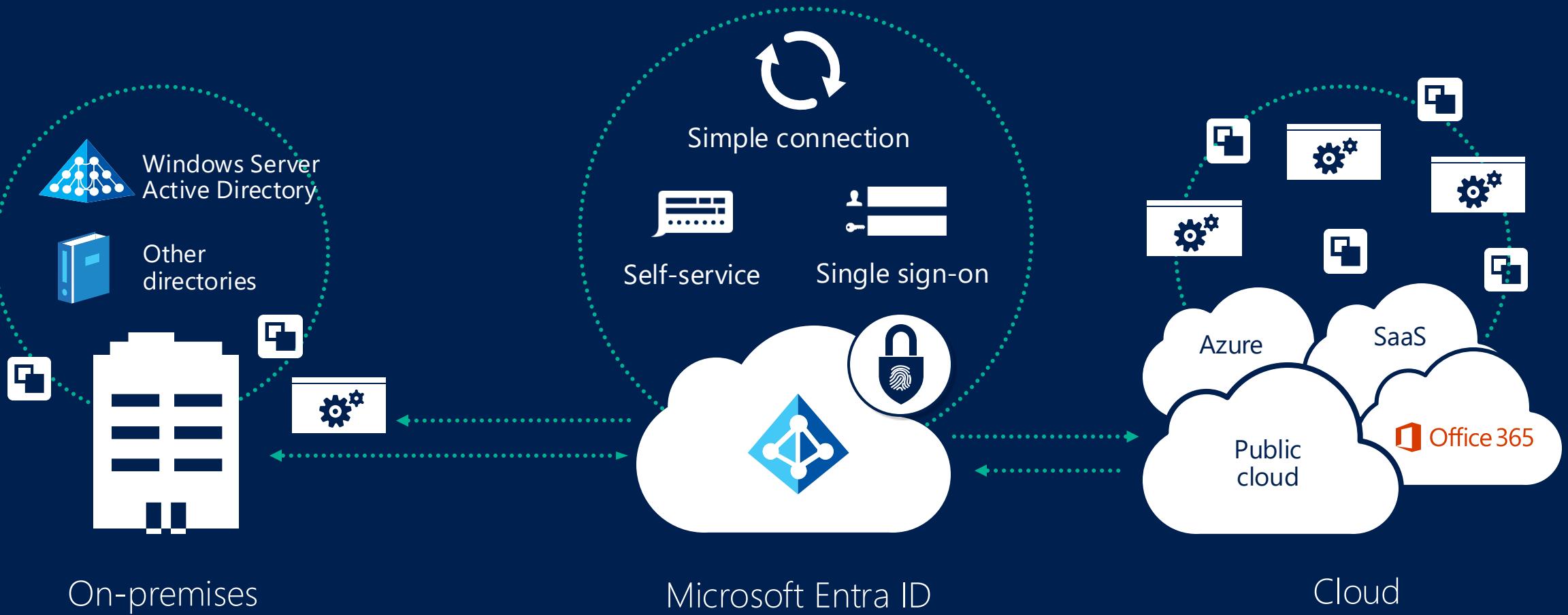


Entra ID



Windows Server

IDENTITY AS THE CORE OF ENTERPRISE MOBILITY



MICROSOFT ENTRA ID (AAD)

Microsoft's "Identity Management as a Service (IDaaS)" for organizations.

Millions of **independent** identity systems controlled by enterprise and government "tenants."

Information is **owned and used by the controlling organization**—not by Microsoft.

Born-as-a-cloud directory for Office 365. Extended to manage across many clouds.

Evolved to manage an organization's relationships with its customers/citizens and partners (B2C and B2B).

86%
of Fortune 500 companies use Microsoft Cloud (Azure, O365, CRM Online, and PowerBI)

Microsoft Entra ID Directories
>9 M

More than **600 M** user accounts on Microsoft Entra ID

1 trillion
Microsoft Entra ID authentications since the release of the service

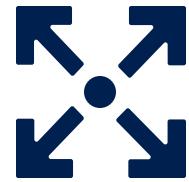
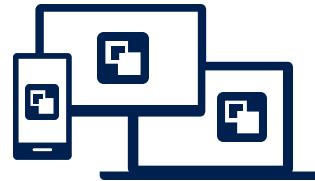
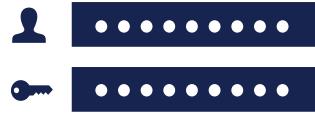
>80k
third-party applications used with Microsoft Entra ID each month

>1.3 billion
authentications every day on Microsoft Entra ID

Every Office 365 and Microsoft Azure customer uses Microsoft Entra ID

IDENTITY AND ACCESS MANAGEMENT IN CLOUD

Microsoft Entra ID. Identity at the core of your business



1000s of apps,
1 identity

Enable business
without borders

Manage access
at scale

Cloud-powered
protection

Provide one persona to the
workforce for SSO to 1000s of cloud
and on-premises apps

Stay productive with universal
access to every app and
collaboration capability

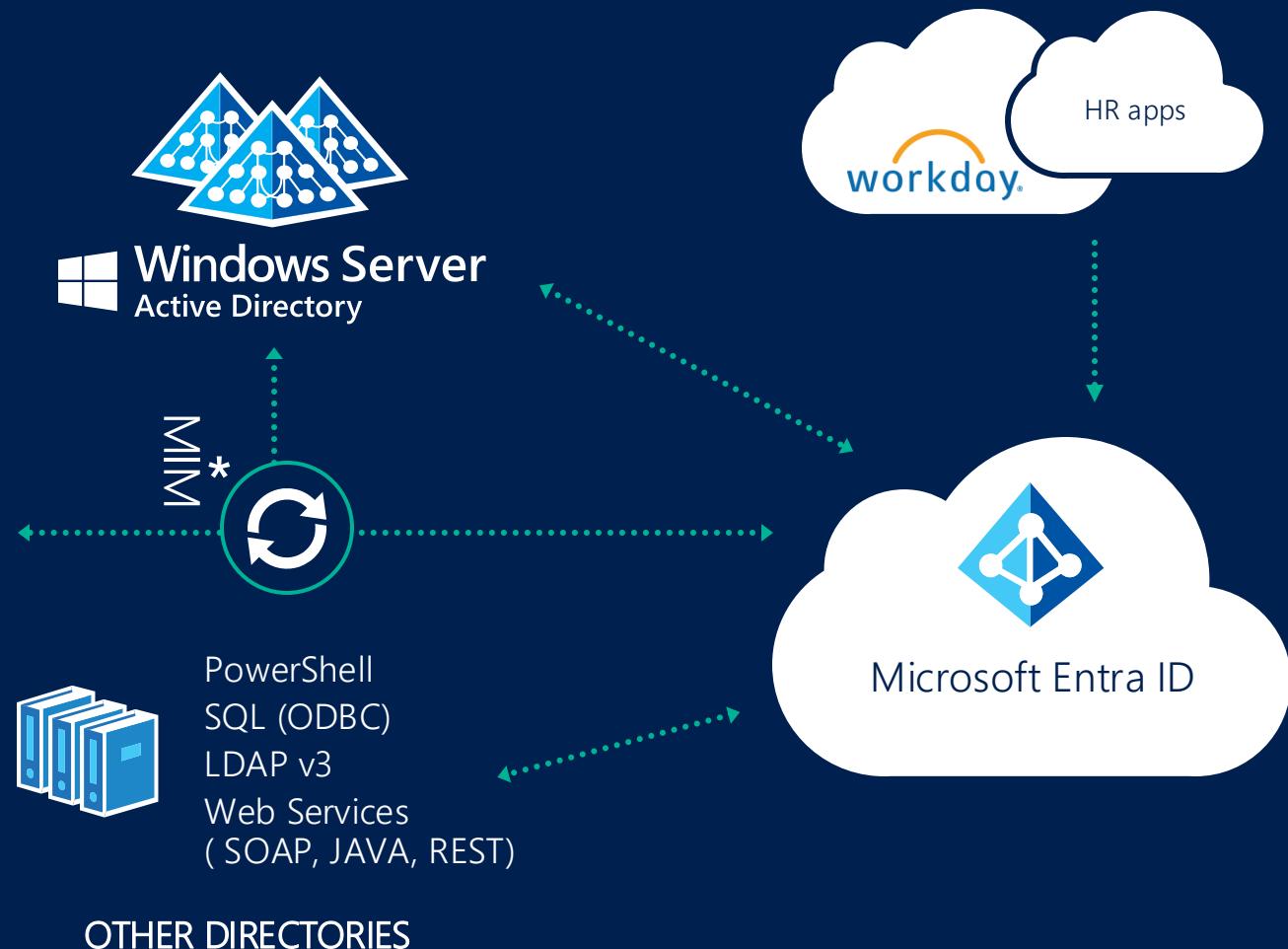
Manage identities and
access at scale in the cloud
and on-premises

Ensure user and admin
accountability with better
security and governance

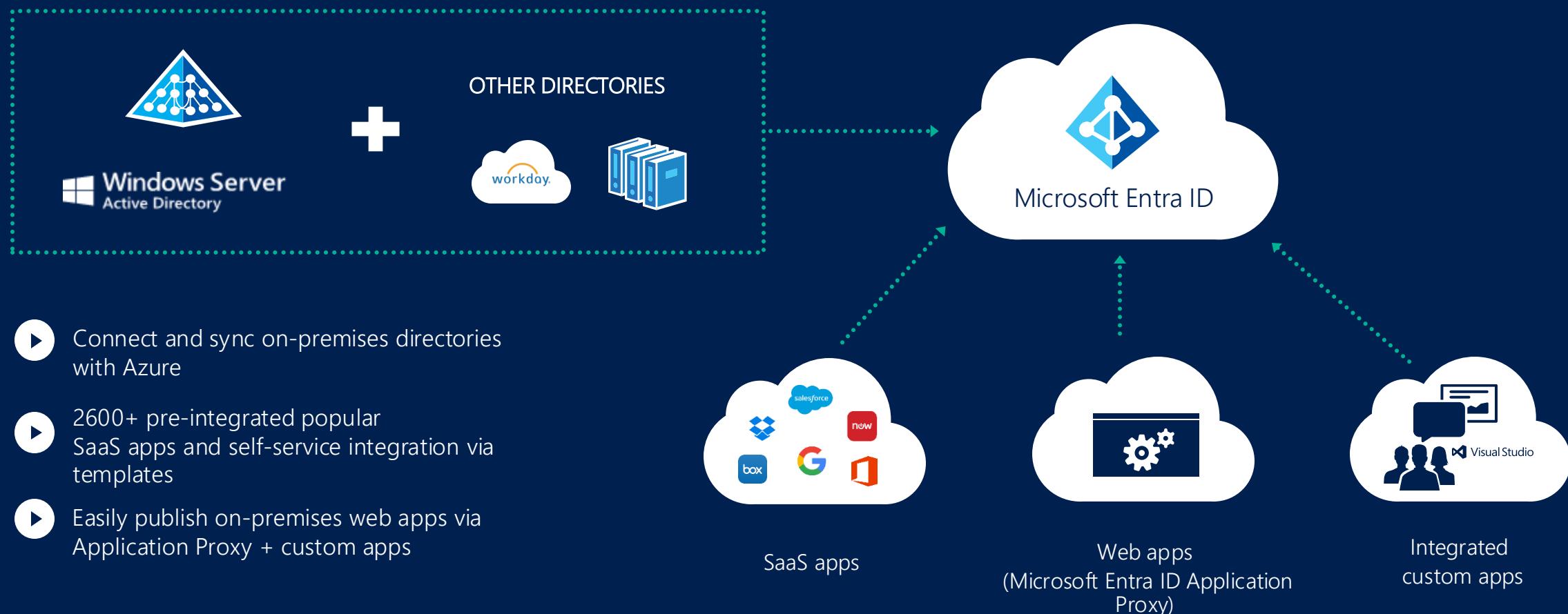
CLOUD CONNECTED, SEAMLESS AUTH EXPERIENCE

Connect and sync on-premises
directories with Microsoft Entra ID

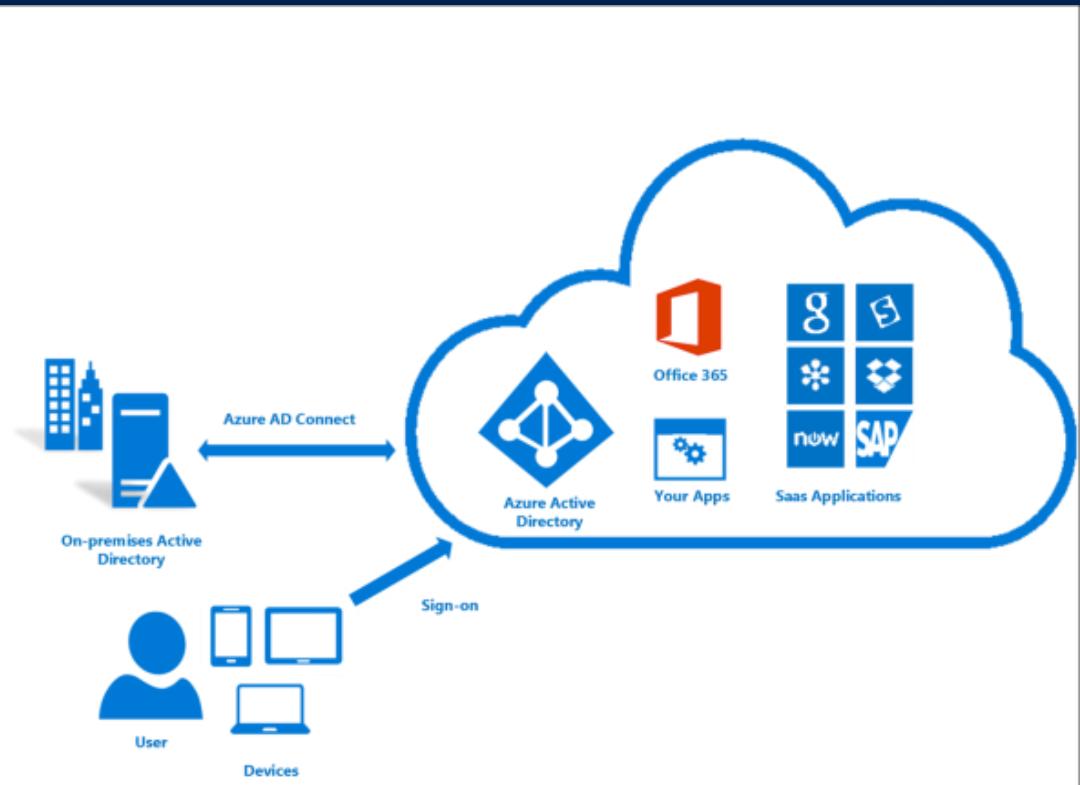
* Microsoft Entra ID Connect and
Connect Health



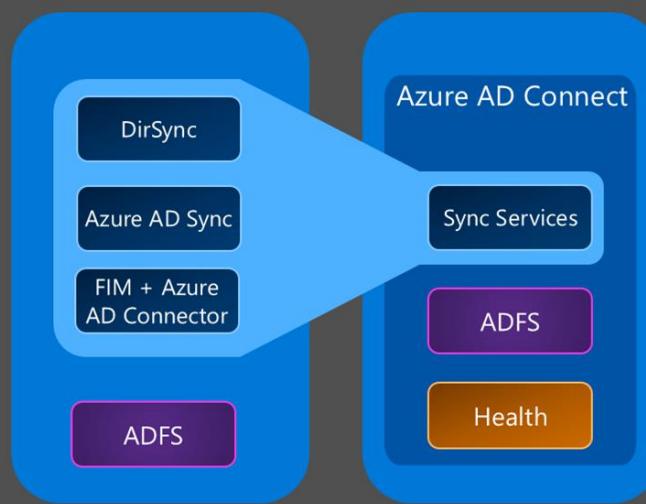
SINGLE SIGN-ON TO ANY APP



MICROSOFT ENTRA CONNECT V2



Making hybrid identity simple



Azure Active Directory Connect

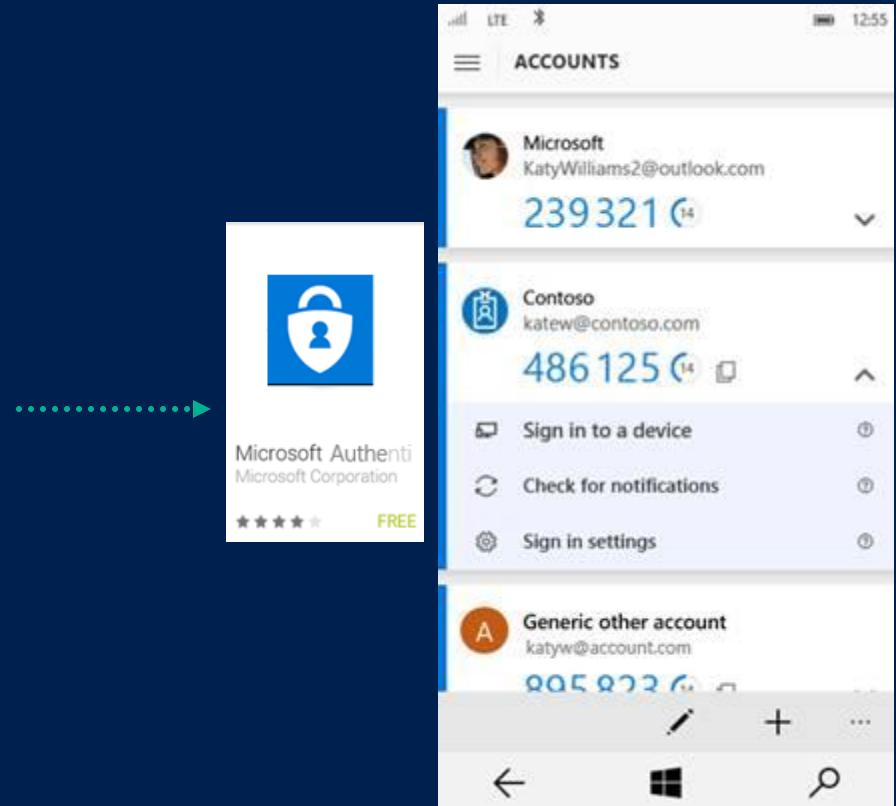
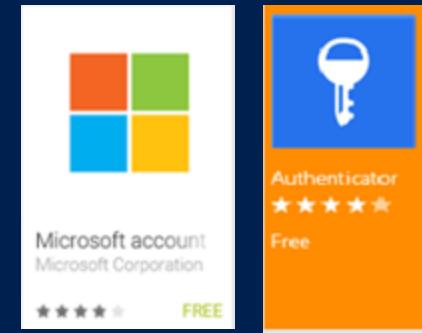
Consolidated deployment
assistant for your identity
bridge components.

MICROSOFT AUTHENTICATOR

A mobile authenticator application for all platforms

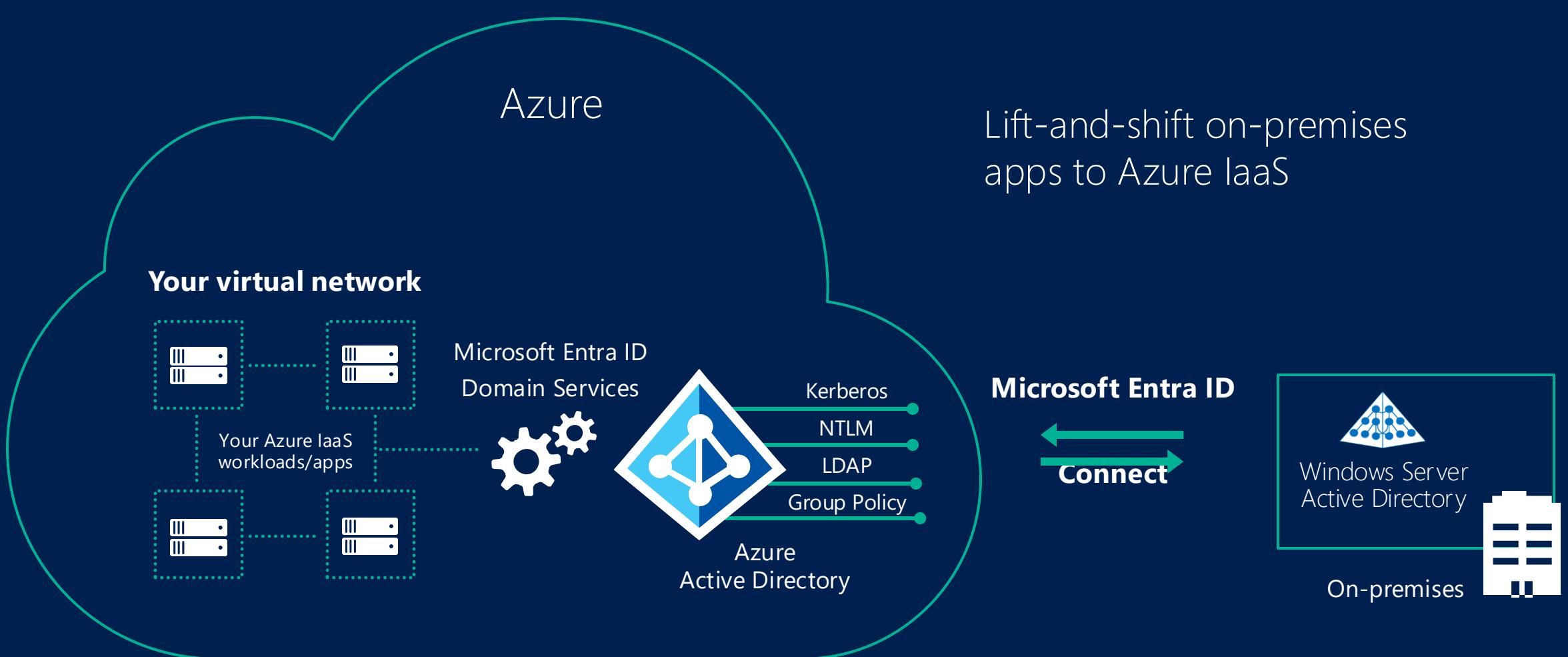
Converges the existing Azure Authenticator and all consumer Authenticator applications.

- ▶ MFA for any account, enterprise or consumer and 3rd party : Push Notifications/OTP
- ▶ Device Registration (workplace join)
- ▶ SSO to native mobile apps - Certificate-based SSO
- ▶ Future: Sign in to a device (Windows Hello), app, or website without a password



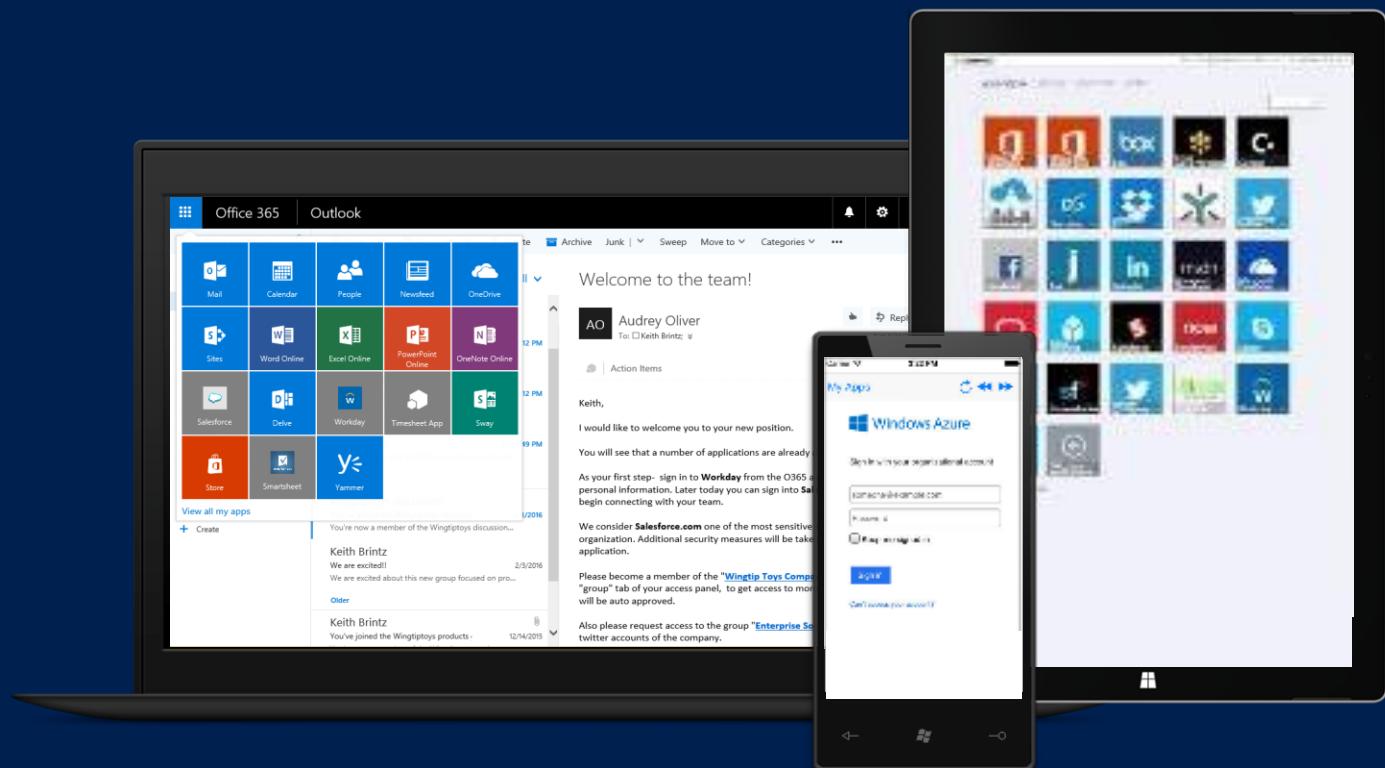
MICROSOFT ENTRA DOMAIN SERVICES

Your domain controller as a service for lift-and-shift scenarios

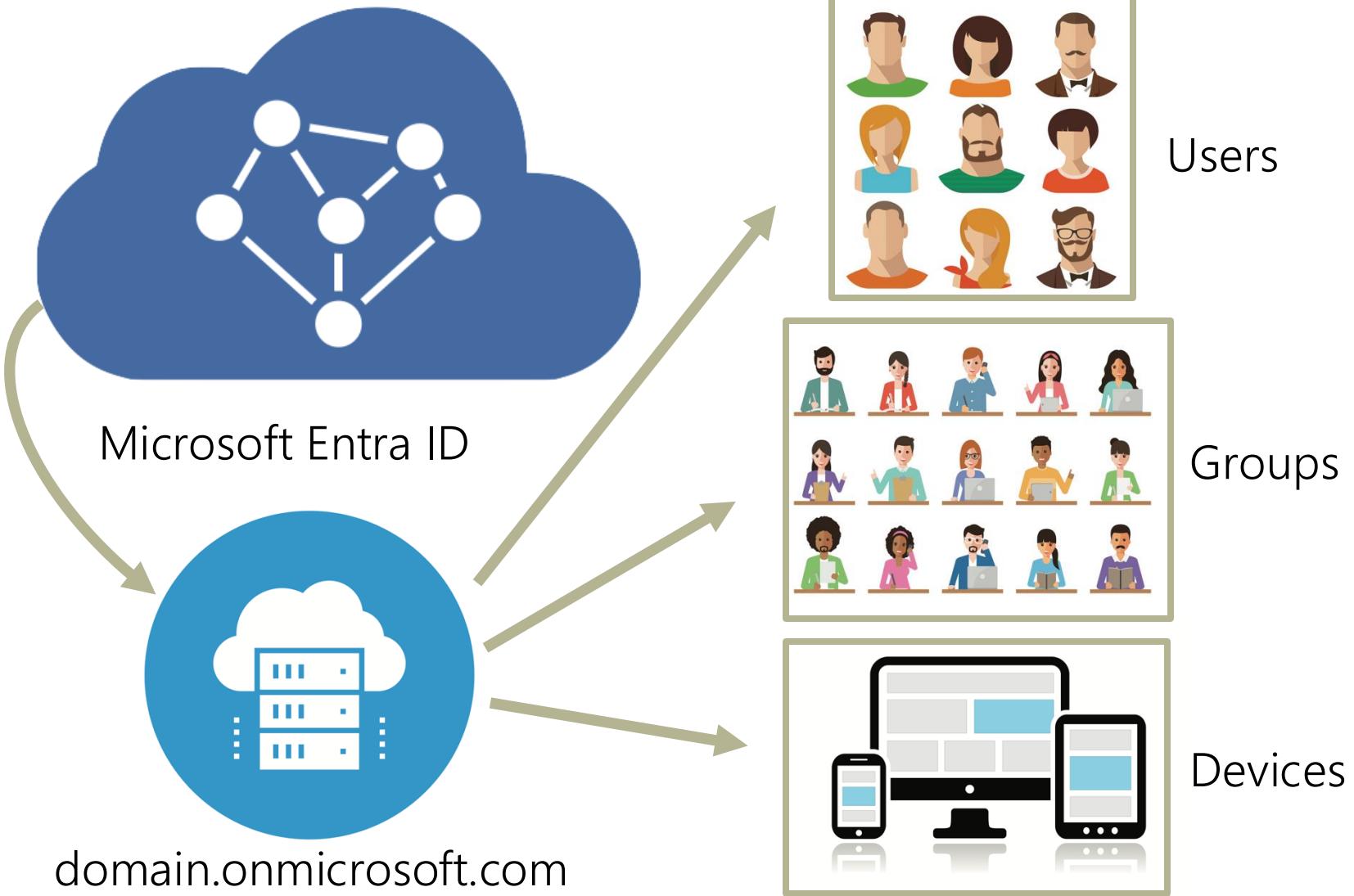


MAKING LIVES OF USERS (AND IT) EASIER

- ▶ Company-branded, personalized application Access Panel:
<http://myapps.microsoft.com>
+ iOS and Android Mobile Apps
- ▶ Integrated Office 365 app launching
- ▶ Manage your account, apps, and groups
- ▶ Self-service password reset
- ▶ Application access requests



USER MANAGEMENT



CUSTOM DOMAINS



- FQDN
- Must be verified
 - TXT record
- Local Active Directory
- Subdomains
- Primary domain per tenant

MICROSOFT ENTRA ID USERS



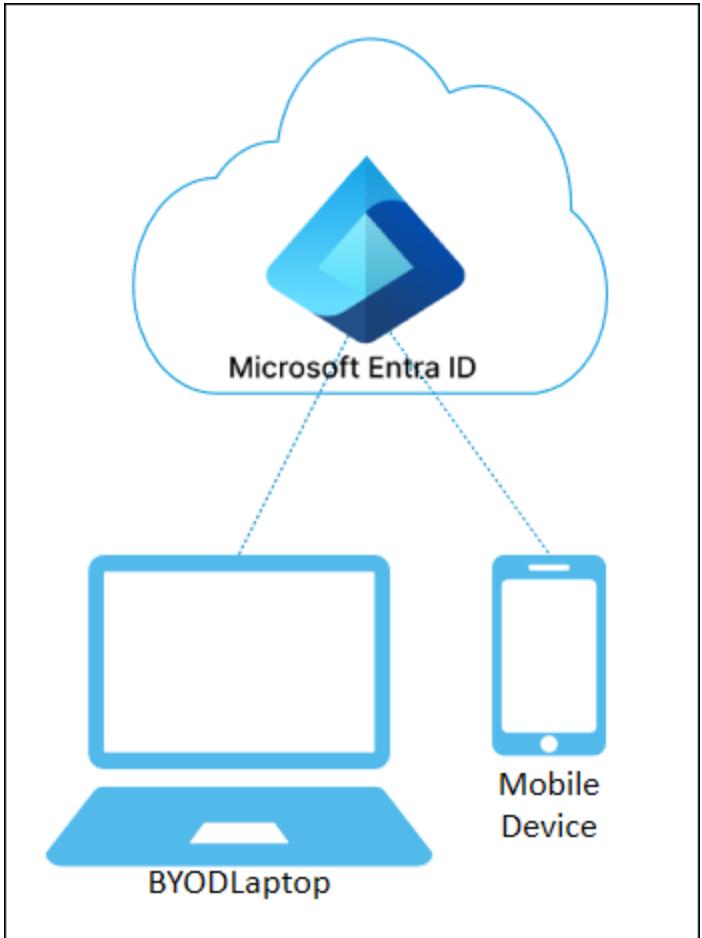
- Types of users
 - Cloud
 - Created in AAD/Flagged as cloud
 - Synced
 - From onsite AD/Flagged as local
 - Member or Guest
- Common settings
- Usage Location
- User Principal Name

MICROSOFT ENTRA ID GROUPS



- Types of groups
 - Assigned or Dynamic
 - Security or Office 365
- Owners and members
- Expiration of groups

MICROSOFT ENTRA ID DEVICES



Registration types

- Registered or joined
- Hybrid joined

MICROSOFT ENTRA ID APPS



- Microsoft Entra ID IDaaS
 - Application types
 - Third-party or internal
 - Pre-integrated or proxied
 - Automated user provisioning
 - Available on select apps

APP REGISTRATIONS

Application Object

The global definition of your app in the tenant

Service Principal

Tenant-specific identity representing the app at runtime

Redirect URIs

Allowed callback endpoints for auth responses

Supported Account Types

Single-tenant, multi-tenant, or personal Microsoft accounts

Certificates & Secrets

Credentials used for app authentication

The screenshot shows the Azure App Registrations portal for the application 'WebAppA2022'. The left sidebar includes links for Overview, Quickstart, Integration assistant, Diagnose and solve problems, Manage (with sub-links for Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions, Expose an API, App roles, Owners, Roles and administrators, Manifest, and Support + Troubleshooting), Get Started, and Documentation.

The main content area displays the application's details under the 'Essentials' section:

- Display name: WebAppA2022
- Application (client) ID: [REDACTED]
- Object ID: [REDACTED]
- Directory (tenant) ID: [REDACTED]
- Supported account types: All Microsoft account users
- Client credentials: 0 certificate, 1 secret
- Redirect URLs: 1 web, 0 spa, 0 public client
- Application ID URI: api/...
- Managed application in I...: WebAppA2022

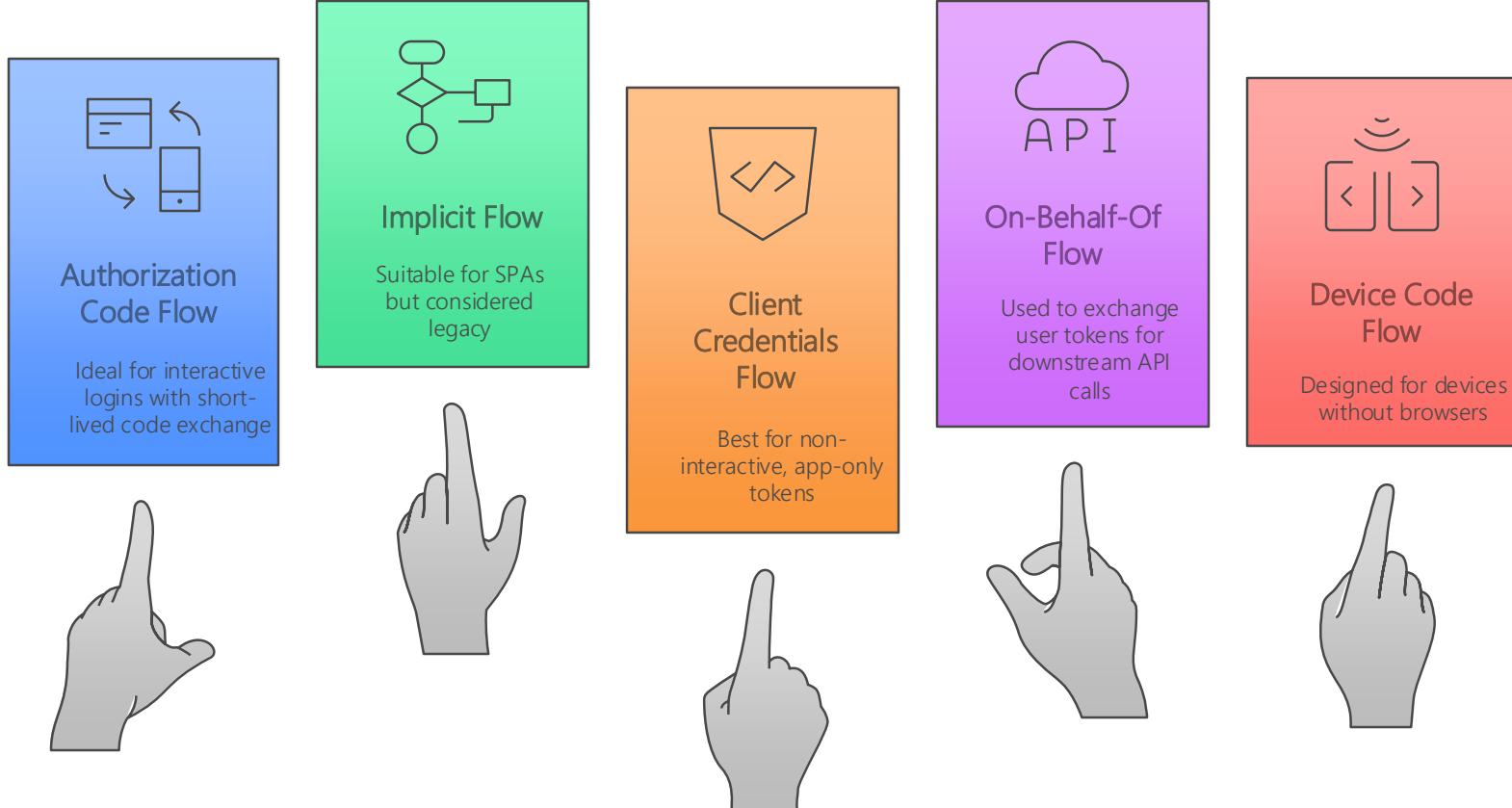
Below the essentials, there are two informational boxes:

- A blue box: Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)
- An orange box: Starting November 9th, 2020 end users will no longer be able to grant consent to newly registered multitenant apps without verified publishers. [Add MPN ID to verify publisher](#)

At the bottom, there are three quickstart sections:

- Call APIs**: Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources. [View API permissions](#)
- Sign in users in 5 minutes**: Use our SDKs to sign in users and call APIs in a few steps. Use the quickstarts to start a web app, mobile app, SPA, or daemon app. [View all quickstart guides](#)
- Configure for your organization**: Assign users and groups, apply conditional access policies, configure single sign-on, and more in Enterprise applications. [Go to Enterprise applications](#)

OAUTH 2.0 & OPENID CONNECT FLOWS



EXPOSING APIs & SCOPES

The screenshot shows the Azure portal interface for managing API scopes. The left sidebar navigation bar includes links for Overview, Quickstart, Integration assistant, Diagnose and solve problems, Manage (with sub-links for Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions, and Expose an API), App roles, Owners, Roles and administrators, Manifest, and Support + Troubleshooting. The 'Expose an API' link is highlighted with a grey background.

The main content area displays the Application ID URI: `api://d43ca0cf-8fee-4bde-b6ca-f81e5e8a1670`. Below this, it says "Scopes defined by this API" and provides instructions for defining custom scopes to restrict access to parts of the API. It notes that adding a scope creates delegated permissions and links to 'App roles' for application-only scopes.

A table lists the defined scope:

Scopes	Who can consent	Admin consent display name	User consent display name	State
<code>api://d43ca0cf-8fee-4bde-b6ca-f81e5e8a1670/user_i...</code>	Admins and users	Access WebAppA2022	Access WebAppA2022	Enabled

The section "Authorized client applications" indicates that no client applications have been authorized yet.

CLIENT CONFIGURATION & PERMISSIONS

WebAppA2022 | API permissions

Search Refresh Got feedback?

Overview Quickstart Integration assistant Diagnose and solve problems Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions Expose an API App roles Owners Roles and administrators Manifest Support + Troubleshooting

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions shows all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for Drimkoe Innovations

API / Permissions name	Type	Description	Admin consent requ...	Status
✓ Azure Cosmos DB (1)				
user_impersonation	Delegated	Access Azure Cosmos DB	No	
✓ Microsoft Graph (2)				
Calendars.Read	Delegated	Read user calendars	No	✓ Granted for
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

SECURITY BEST PRACTICES WITH ENTRA ID

Managed Identities for Azure Resources: Eliminate secrets by using platform-managed identities

Certificate Rotation: Automate renewal for app certificates via Key Vault

Conditional Access Policies: Enforce MFA, device compliance before token issuance

Application Proxy Integration: Publish on-prem apps securely via Entra ID

Monitoring & Alerting: Track sign-ins, risky apps, and expired credentials

POP QUIZ:

You have a multi-tenant web API registered in Microsoft Entra ID with scope api://1234/read. A client app in another tenant requests tokens for that scope but fails. Which step is **required** to enable cross-tenant access?

- A. Add a redirect URI in the client's registration
- B. Grant admin consent in the target tenant's Enterprise Applications
- C. Expose the API scope as "public" in the application manifest
- D. Assign the client app to an Azure AD security group in the issuing tenant

POP QUIZ:

You have a multi-tenant web API registered in Microsoft Entra ID with scope api://1234/read. A client app in another tenant requests tokens for that scope but fails. Which step is **required** to enable cross-tenant access?

- A. Add a redirect URI in the client's registration
- B. Grant admin consent in the target tenant's Enterprise Applications**
- C. Expose the API scope as "public" in the application manifest
- D. Assign the client app to an Azure AD security group in the issuing tenant

LAB: ENTRA ID

- Create an app registration
- Deploy nodejs app which uses Entra ID for authentication



Infrastructure



Platform



Software



AZURE WEB APPS

The screenshot shows the Azure Marketplace page for the 'Web App + Database' template. It includes the template icon, name, rating (3.2, 19 ratings), a 'Create' button, and tabs for Overview, Plans, Usage Information + Support, and Ratings + Reviews.

Overview Plans Usage Information + Support Ratings + Reviews

Create and deploy web apps in seconds, as powerful as you need them

Leverage your existing tools to create and deploy applications without the hassle of managing infrastructure. App Service deployment, and scaling options for any sized web application. Use frameworks and templates to create web apps in seconds. GitHub, and BitBucket. Use any tool or OS to develop your app with .NET, PHP, Node.js or Python.

Use this Azure template to create an App Service web app with your choice of database: Azure SQL, Postgres, MySQL, or MongoDB. The platform that scales automatically, where you can:

- Use any code editor or OS
- Develop with .NET, PHP, Node.js or Python
- Select source control options like GitHub, Azure pipelines, and more
- Monitor, alert, and auto scale
- Optional Azure Cache for Redis to boost app performance and save computing resources
- Use frameworks and templates to create web apps in seconds

Create Web App + Database ...

i Effective May 31, the dedicated Web App + Database create experience will be removed. All its functionality is now integrated into the

Basics Tags Review + create

This template will create a secure by default configuration where the only publicly accessible endpoint will be your app following the recommended security best practices. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * i Microsoft Azure Sponsorship

Resource Group * i demo-rg [Create new](#)

Region * Canada Central

Web App Details

Name Web App name [.azurewebsites.net](#)

Secure unique default hostname on. [More about this update](#)

Runtime stack * Select a runtime stack

Database

i Database access will be locked down and not exposed to the public internet. This is in compliance with recommended best practices for security.

Engine * i Select a database engine

Server name * Server name

Database name * Database name

[Review + create](#)

[< Previous](#)

[Next : Tags >](#)

WEB APPS - CONFIGURATION OPTIONS

Custom Domains & SSL: Bind custom domains and secure them with free or custom SSL certificates.

Deployment Slots: Use staging slots for zero-downtime deployments and A/B testing.

App Settings & Secrets: Manage environment variables and secrets securely via Azure Key Vault.

Backup & Restore: Automate backups and restore apps to previous states with minimal downtime.

Traffic Routing: Route traffic between slots for gradual rollouts or testing.

-  Diagnose and solve problems
-  Microsoft Defender for Cloud
-  Events (preview)
-  Recommended services (preview)
-  Resource visualizer
-  Deployment
 -  Deployment slots
 -  Deployment Center
-  Settings
 -  Environment variables
 -  Configuration
 -  Authentication
 -  Identity
 -  Backups
 -  Custom domains
 -  Certificates
 -  Networking
 -  Scale up (App Service plan)
 -  Scale out (App Service plan)
 -  WebJobs
 -  Service Connector
 -  Locks

WEB APPS - PERFORMANCE OPTIMIZATION

Auto-Scaling Rules

Automatically adjust resources based on demand using metrics.

Always On

Ensures applications remain active, minimizing delays from cold starts.

App Service Plans

Select the appropriate service tier to meet performance requirements.

VNet Integration

Securely link applications to backend services through virtual networks.

Caching Strategies

Implement caching with Azure Cache for Redis to optimize database performance.

WEB APPS – SECURITY BEST PRACTICES

Authentication & Authorization: Integrate with Microsoft Entra ID, OAuth, or social logins.

Managed Identity: Securely access Azure resources without storing credentials.

Private Endpoints: Restrict access to your app via private IP addresses.

DDoS Protection & WAF: Protect against common web threats and attacks.

TLS Enforcement: Enforce HTTPS and use latest TLS versions.

WEB APPS – MONITORING & DIAGNOSTICS



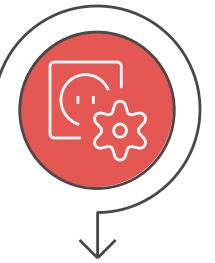
Application Insights

Provides real-time telemetry and performance metrics. Tracks user behavior for insights.



Log Streaming

Allows viewing logs in real time for debugging. Aids in quick diagnostics.



Diagnostic Settings

Routes logs to Log Analytics, Event Hub, or Storage. Offers flexible log management.



Health Checks

Monitors application health and triggers auto-healing processes. Ensures application stability.



Alerts & Metrics

Sets up alerts for performance, availability, and security. Notifies of potential anomalies.

WEB APPS – DEVOPS AUTOMATION

CI/CD Pipelines: Integrate with GitHub, Azure DevOps, or Bitbucket for automated deployments.

Infrastructure as Code: Use ARM, Bicep, or Terraform to manage app infrastructure.

Deployment Center: Visual interface for configuring deployment sources and workflows.

GitHub Actions: Automate testing, builds, and deployments with reusable workflows.

Azure CLI & PowerShell: Script and automate app management tasks.

APP SERVICES – FOR MOBILE APPS

Mobile App Backend: Use Azure App Service to host scalable RESTful APIs.

Authentication & Authorization: Integrate with Microsoft Entra ID, Facebook, Google, and more.

Push Notifications: Send targeted messages using Azure Notification Hubs.

Offline Sync: Enable data sync when devices reconnect to the internet.

Cross-Platform Support: Backend works with iOS, Android, MAUI, and React Native.



APP FUNCTIONS & MOBILE WORKLOADS

Serverless Architecture: Run backend logic without managing infrastructure.

Event-Driven Triggers: Respond to HTTP requests, queues, or database changes.

Scalable APIs: Build lightweight, scalable endpoints for mobile apps.

Cost Efficiency: Pay only for execution time, ideal for bursty workloads.

Integration with Azure Services: Easily connect to Cosmos DB, Blob Storage, and more.

DATABASES FOR APPS - COSMOS DB



APP FUNCTIONS & MOBILE WORKLOADS

Serverless Architecture: Run backend logic without managing infrastructure.

Event-Driven Triggers: Respond to HTTP requests, queues, or database changes.

Scalable APIs: Build lightweight, scalable endpoints for mobile apps.

Cost Efficiency: Pay only for execution time, ideal for bursty workloads.

Integration with Azure Services: Easily connect to Cosmos DB, Blob Storage, and more.

AZURE FUNCTIONS - ARCHITECTURE



Serverless Compute

Executes code without VM management

Event-Driven

Reacts to triggers rather than polling

Stateless Workers

Isolated executions with optional durable state

Binding Abstractions

Simplifies I/O via input/output bindings

Integration Framework

Connects seamlessly to Azure services

FUNCTION EXECUTION FLOW

Your function

```
def my_handler(event, context):
    message = 'Hello {} {}!'.format(event['first_name'],
                                    event['last_name'])
    return {
        'message' : message
    }
```

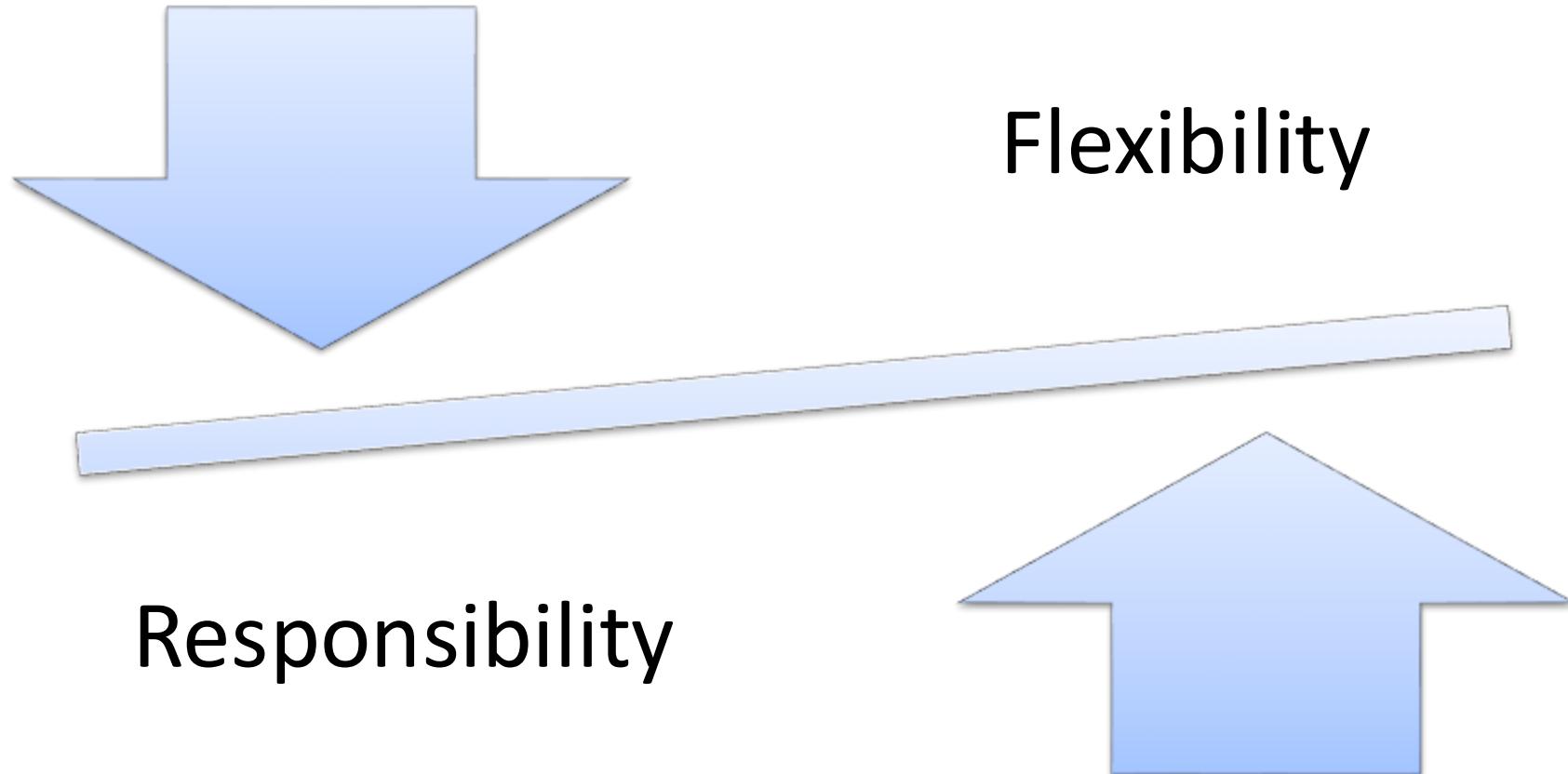
Container

- Your code is encapsulated in a prebuilt container from the provider that contains a dispatch agent. An ultra light HTTP endpoint that accepts requests, and executes your snippet of code.

Serverless

- When a request comes in, an API gateway looks for a container running your function, if none exist one is created and the request is routed.

SERVERLESS IS CONTAINERS?



AZURE FUNCTIONS – HOSTING MODELS & PLANS

Characteristic	Consumption	Premium	Dedicated (App Service)	Elastic Premium	Kubernetes & Container
Scaling	Auto-scale	Pre-warmed instances	Existing App Service resources	Controls min/max scale	Custom containers
Pricing	Pay-per-execution	Included	Included	Included	Varies
Infrastructure	Serverless	Serverless	App Service	Serverless	Custom containers
VNet Support	No	Yes	Yes	Yes	Yes
Cold Start	Yes	Controls cold-start	No	Controls cold-start	Varies

AZURE FUNCTIONS – TRIGGERS & BINDINGS



Trigger Types

HTTP, Timer, Queue, Event Grid, Cosmos DB, etc.

Input Bindings

Declarative data reads without client SDKs

Output Bindings

Push data to services with minimal code

Binding Expressions

Parameterize at runtime using metadata

Extension Model

Custom bindings via NuGet for new services

AZURE FUNCTIONS – SCALING & PERFORMANCE

Dynamic Scaling

Scale based on event metrics automatically

Cold-Start Impact

Mitigated by pre-warming or Premium plan

Concurrency Controls

Batch sizes, max concurrent function executions

Host Scaling Logic

Based on queue length, CPU, and custom metrics

Distributed Tracing

Monitor performance via Application Insights

AZURE FUNCTIONS – SECURITY, DEVOPS, BEST PRACTICES

Managed Identities

Securely access Azure AD-protected resources

Function Keys & AAD

Control HTTP access via keys or OAuth tokens

CI/CD Integration

Deploy via GitHub Actions, Azure DevOps pipelines

Configuration Management

Use App Settings & Key Vault references

Monitoring & Alerts

Leverage built-in logs, metrics, and alerts

The screenshot shows the 'System keys' section of the Azure Functions portal. On the left, there's a sidebar with icons for Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Recommended services (preview), Resource visualizer, Functions (selected), App keys (highlighted), App files, Proxies, Deployment, Settings, Environment variables, Configuration, and Authentication. The main area has a header with 'Search', 'New host key', 'Refresh', and 'Send us your feedback'. Below it, a sub-header says 'System keys' with a note: 'System keys are automatically managed by the Function runtime. System Keys provide granular access to functions runtime features.' A table lists 'Name' and 'Value' for two host keys: '_master' and 'default', each with 'Show value' and 'Renew key value' buttons.

Name	Value
_master	[REDACTED] Show value Renew key value
default	[REDACTED] Show value Renew key value

DURABLE FUNCTIONS

Orchestrator Functions

Coordinate workflows by calling activity functions

Activity Functions

Execute discrete, single-purpose tasks

Entity Functions

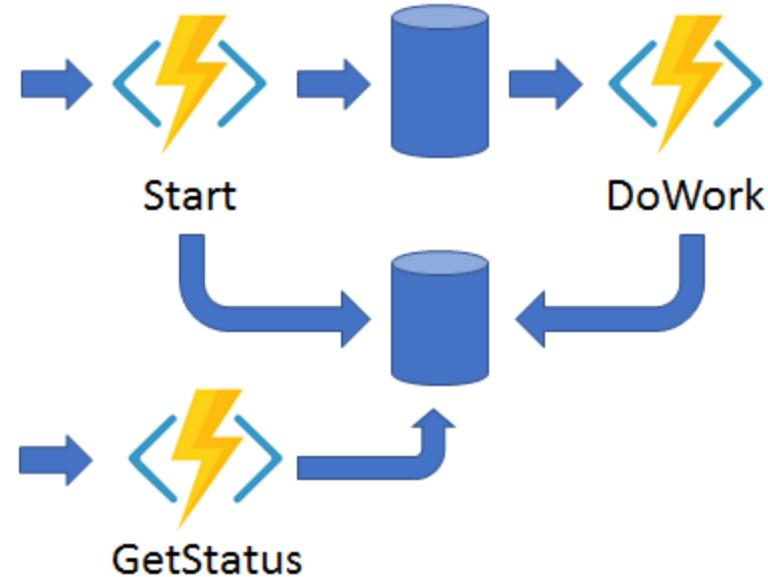
Provide durable, single-instance stateful objects

Client Functions

Serve as entry points to start or query orchestrations

Task Hubs

Underlying storage containers for state and history



DURABLE FUNCTIONS – ORCHESTRATION PATTERNS

Function Chaining

Sequential task execution, passing outputs forward

Fan-Out/Fan-In

Parallel activities with aggregated results

Async HTTP APIs

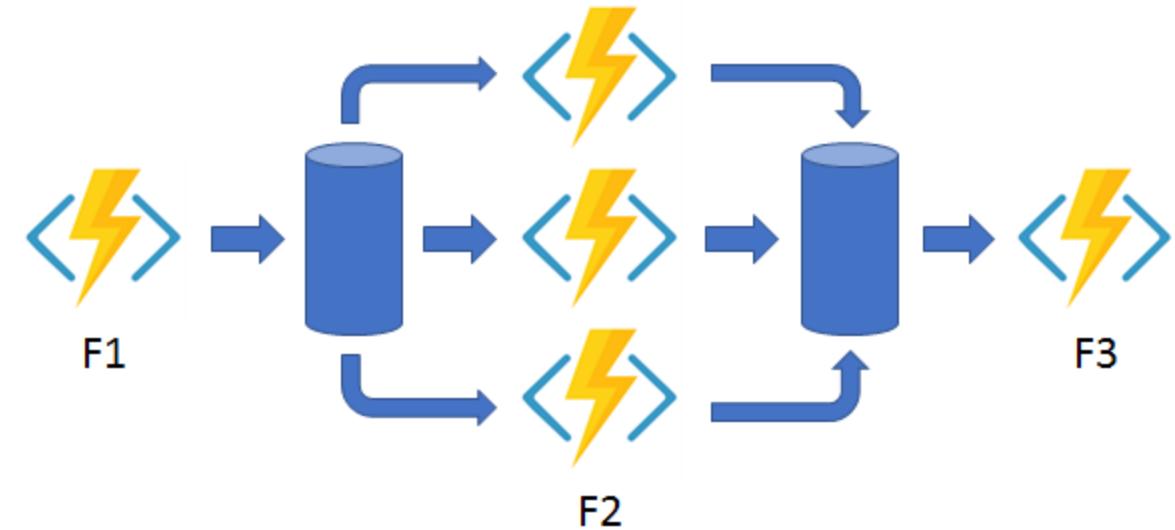
Long-running processes with status monitoring

Human Interaction

Pausing for external events like approvals

Monitoring & Compensation

Implement retries, compensating actions on failure



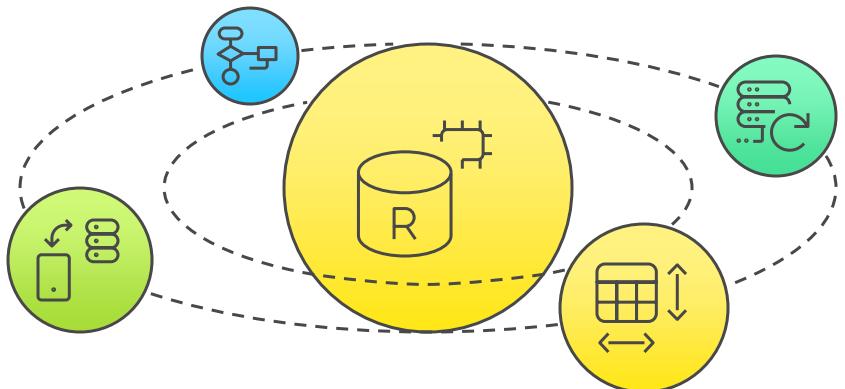
ENTITY FUNCTIONS & STATE MANAGEMENT

Operations & Signals

Invoke entity methods by name or pattern

Lock-Free Concurrency

Runtime serializes messages per entity



State Versioning

Automatic snapshotting and history replay

Scaling Entities

Entities scale horizontally via sharding

Durable Entities are the Durable Functions answer to stateful objects. Each entity represents a single logical instance identified by an ID. You define methods on the entity class; the runtime takes care of persisting its state in storage.

RELIABILITY & PERFORMANCE

Checkpointing & Replay

Ensures orchestrator resilience across failures

Activity Retries

Configurable retry policies on transient errors

Parallelism Controls

Throttle fan-out to protect downstream systems

Storage Throttling

Backpressure handling for Azure Storage limits

Metrics & Billing

Track orchestration count, execution time, storage costs



DURABLE FUNCTIONS – DEVOPS & BEST PRACTICES

Idempotent Activities: Design activities to handle replays safely

Versioning Orchestrations: Use version checks to manage code updates

Local Debugging: Leverage the Functions Core Tools emulator

Monitoring Dashboards: Visualize live orchestration states via App Insights

Cleanup & Retention: Configure history retention to control storage growth

LAB: WebApp

- Deploy web app with az cli
- Deploy second web app with ARM templates

MONITORING



MONITORING & OBSERVABILITY

 Monitor | Overview Microsoft

The Log Analytics agents, used by VM Insights, won't be supported as of August 31, 2024. Plan to migrate to VM Insights on Azure Monitor agent prior to this date. →

[Search](#) [X](#) [<](#)

[Overview](#) [Tutorials](#)

Insights
Use curated monitoring views for specific Azure resources. [View all insights](#)

 Application insights Monitor your app's availability, performance, errors, and usage. View More	 Container Insights Gain visibility into the performance and health of your controllers, nodes, and containers. View More	 VM Insights Monitor the health, performance, and dependencies of your VMs and VM scale sets. View More	 Network Insights View the health and metrics for all deployed network resources. View More
---	--	--	--

Detection, triage, and diagnosis
Visualize, analyze, and respond to monitoring data and events. [Learn more about monitoring](#)

 Metrics Create charts to monitor and investigate the usage and performance of your Azure resources. View More	 Alerts Get notified and respond using alerts and actions. View More	 Logs Analyze and diagnose issues with log queries. View More	 Workbooks View, create and share interactive reports. View More	 Change Analysis Investigate what changed to triage incidents. View More
 Diagnostic Settings Route monitoring metrics and logs to selected locations. View More	 Azure Monitor SCOM managed instance SCOM managed instance monitors workloads running on cloud and on-prem. View More	 Managed Prometheus Collect Prometheus metrics from your containerized workloads to monitor their health and performance. View More		

WHY WE NEED CLOUD MONITORING?

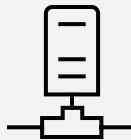
Infrastructure



Apps



Network



Visibility into Health: Detect outages and performance degradation

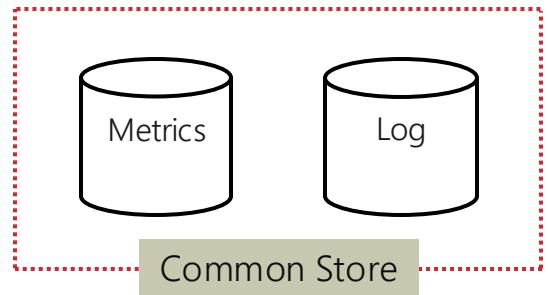
Proactive Issue Detection: Alert on anomalies before customer impact

Capacity Planning: Forecast growth and right-size resources

Compliance & Audit: Retain logs for security and regulatory needs

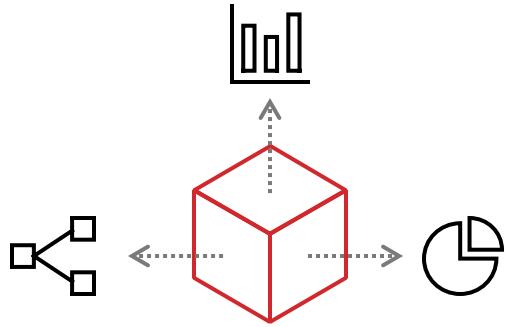
Cost Optimization: Identify under-utilized assets and idle spend

AZURE MONITOR



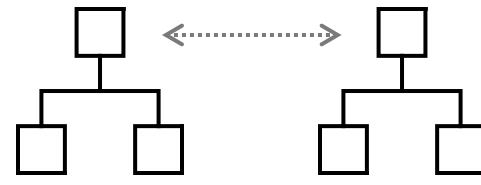
Unified Monitoring

A common platform for all metrics, logs and other monitoring telemetry



Data Driven Insights

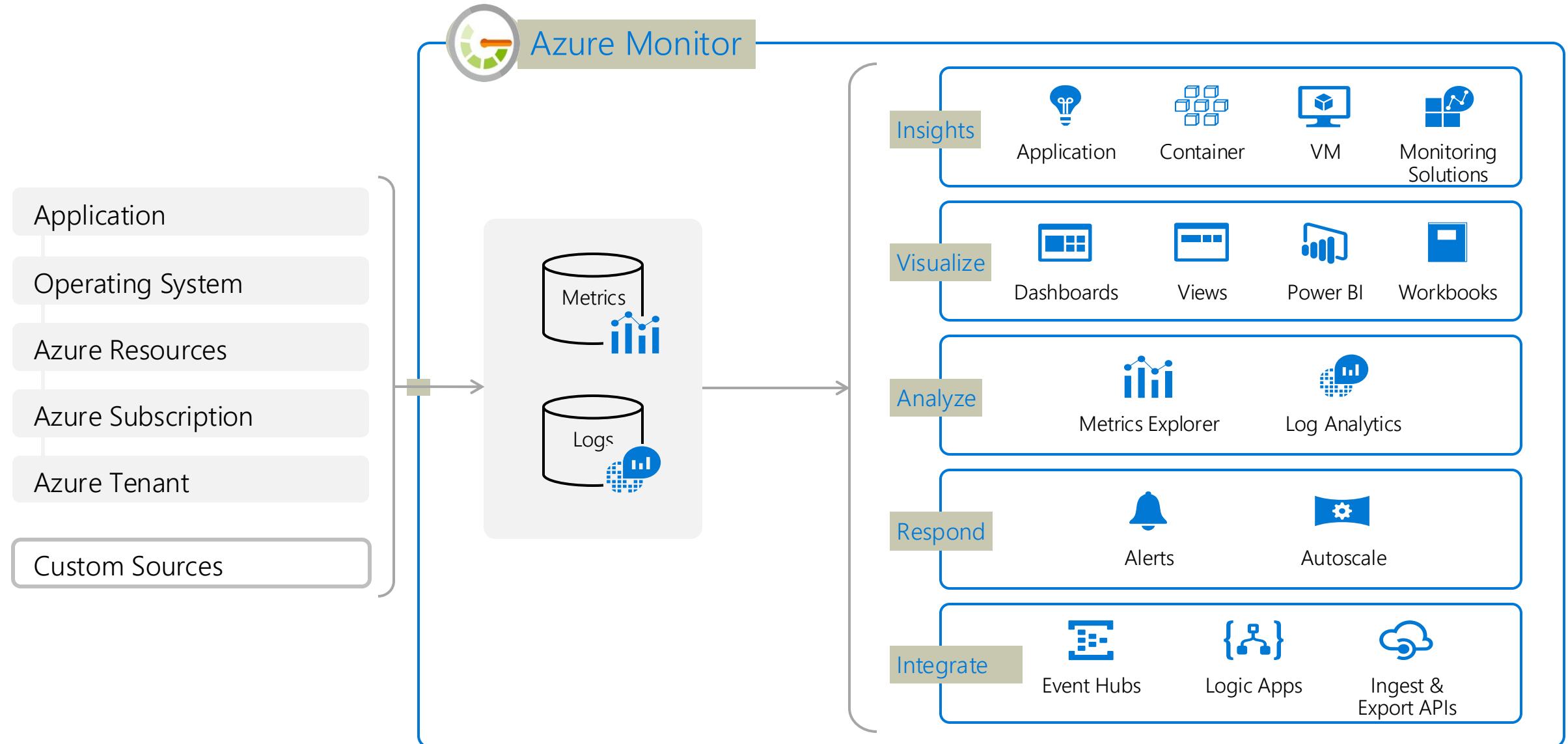
Advanced diagnostics and analytics powered by machine learning capabilities



Workflow Integrations

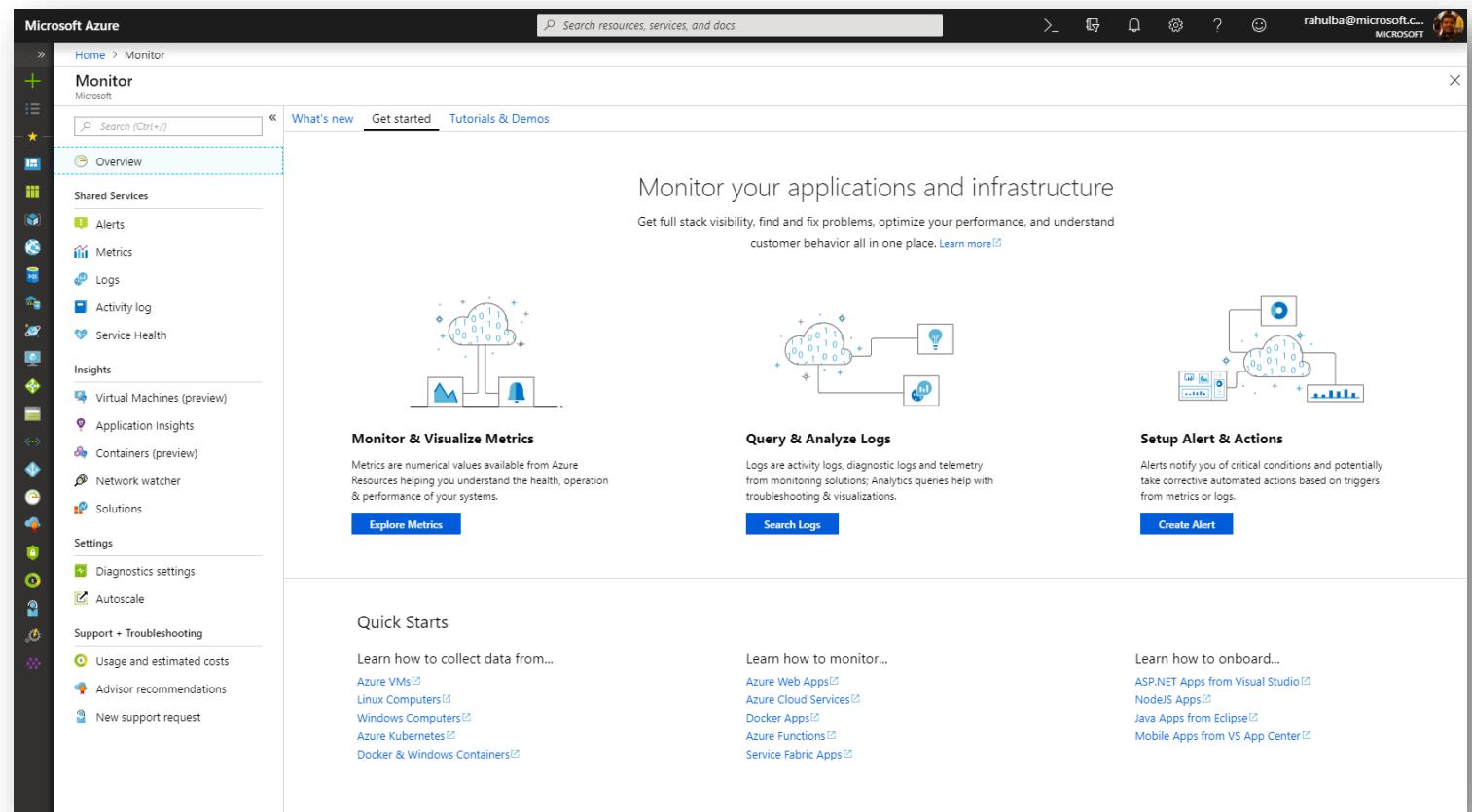
Rich ecosystem of popular DevOps, issue management, SIEM, and ITSM tools

AZURE MONITOR



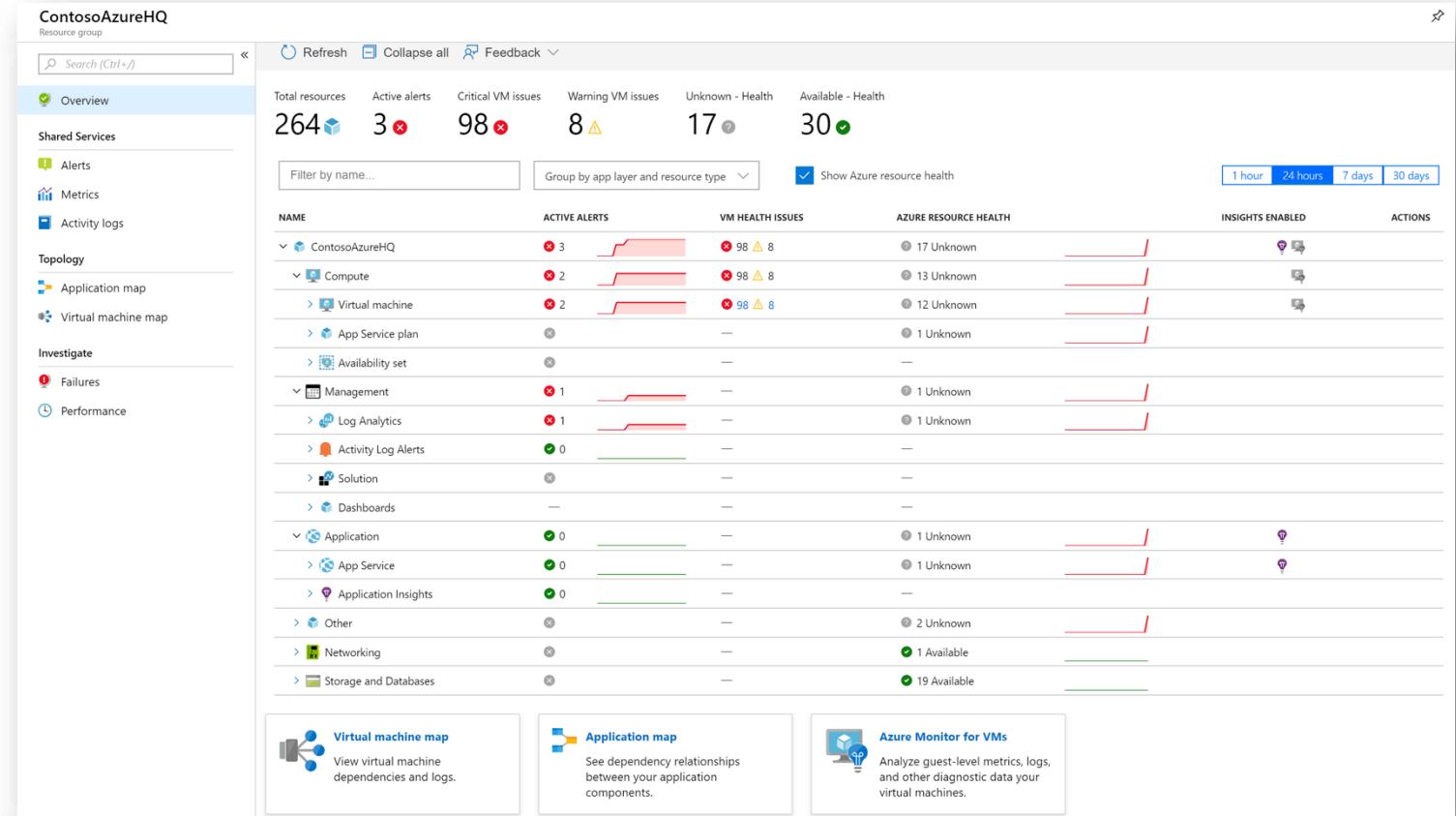
UNIFIED MONITORING DASHBOARD

- One Metrics, One Logs, One Alerts across Azure/on-prem resources
- Unified offering with App Insights & Log Analytics as integrated features
- Integration into native Azure resource blades
- Ability to send custom metrics & custom logs



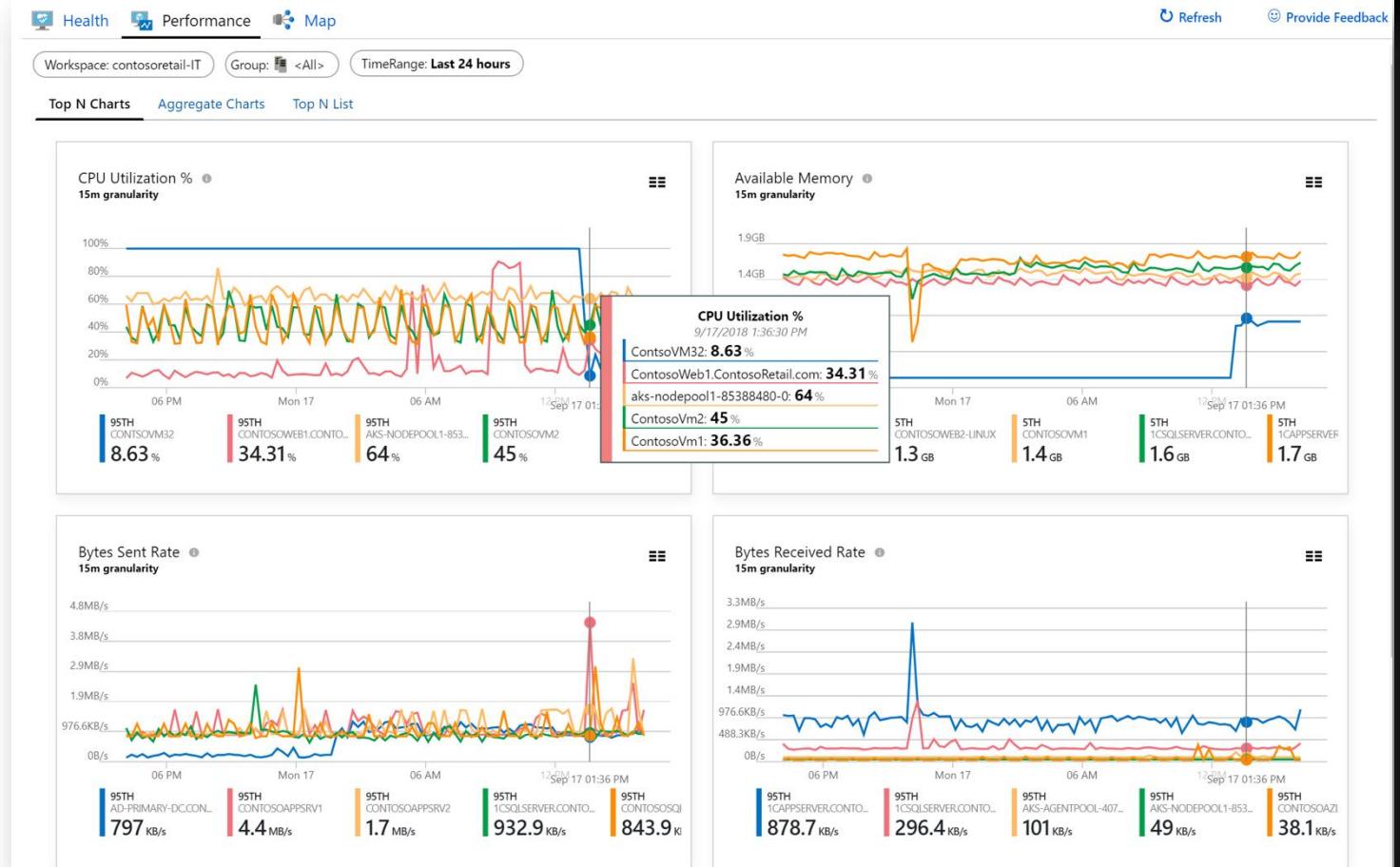
FULL STACK VISIBILITY IN RESOURCE GROUPS

- Monitor health state of all resources
- See alerts firing across app & infra
- Jump to Application Map or VM Map
- Drill down into failures or perf issues



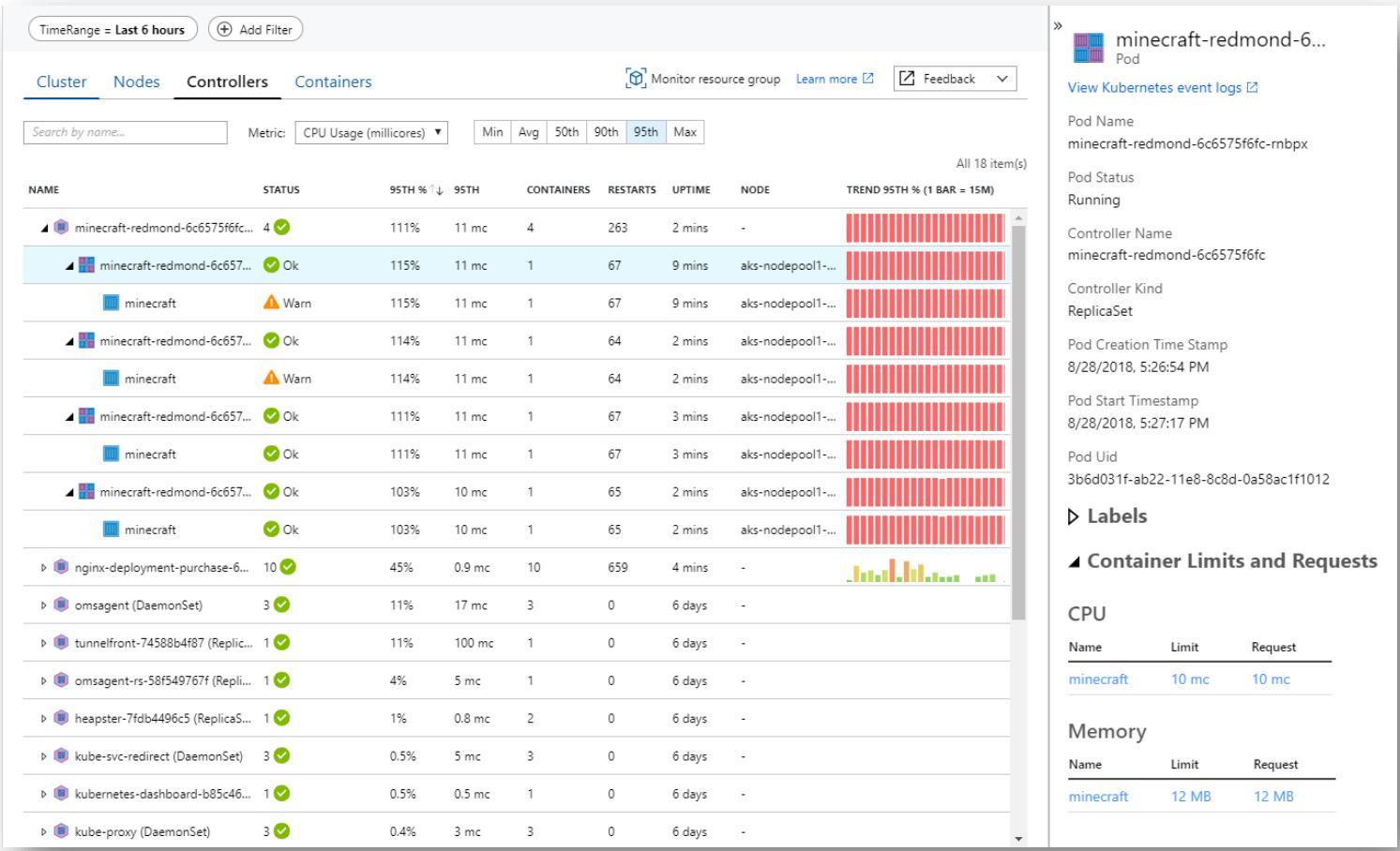
VIRTUAL MACHINES

- Monitor single VMs or at scale
- Identify & isolate host-level or guest-level health problems
- Troubleshoot perf issues like CPU, memory, disk, and network
- Visualize service dependencies & connection failures in Maps
- Onboard at scale using PowerShell or Azure Policy



AKS CONTAINERS

- Monitor multi-cluster health & node/pod status
- View overall perf across nodes, controllers and containers
- Analyze Kubernetes event & container logs for troubleshooting
- Understand cluster capacity needs under average or heaviest loads
- Monitor containers on demand in AKS with virtual nodes



MONITORING VMS, CONTAINERS & APPS

Characteristic	Infrastructure Metrics	Container & Kubernetes	Application Performance	Log Aggregation & Search	Distributed Tracing
Key Aspects	CPU, memory, disk I/O, network throughput	Pod health, node status, cluster events	Request latency, error rates, dependency	Centralized logs from various sources	Follow requests across microservices

MONITORING INTEGRATIONS

Application Insights

Deep APM for .NET, Java, Node, Python with SDKs



Azure Log Analytics

Kusto-powered query language for custom insights

Third-Party Integrations

Prometheus exporters, Grafana dashboards, Datadog, New Relic

ITSM & Notifications

Connect alerts to PagerDuty, ServiceNow, Teams, or OpsGenie

METRICS FOR AUTO-SCALING

Trigger Signals

CPU, memory, queue length, custom App Insights metrics

Scale-Out & Scale-In Rules

Define thresholds, cool-down periods, and instance limits

Predictive Autoscale

Schedule-based scaling aligned to known usage patterns

Custom Metrics & Alerts

Leverage log-based metrics (e.g., request rate per second)

Autoscale Profiles

Multiple rules per profile for different time windows or regions

POP QUIZ:

You need to autoscale a Service Bus-driven function app based on message backlog. Which metric is **best** to trigger scale-out?

- A. CPU Percentage
- B. Memory Working Set
- C. Service Bus Queue Length
- D. HTTP Request Latency

POP QUIZ:

You need to autoscale a Service Bus-driven function app based on message backlog. Which metric is **best** to trigger scale-out?

- A. CPU Percentage
- B. Memory Working Set
- C. Service Bus Queue Length**
- D. HTTP Request Latency

POP QUIZ:

Your Log Analytics workspace is growing 50 GB/day, and costs are soaring. Which action most **directly** reduces storage bill?

- A. Increase workspace pricing tier
- B. Archive older logs to a storage account
- C. Enable daily export to Power BI
- D. Turn on Azure Monitor autoscale

POP QUIZ:

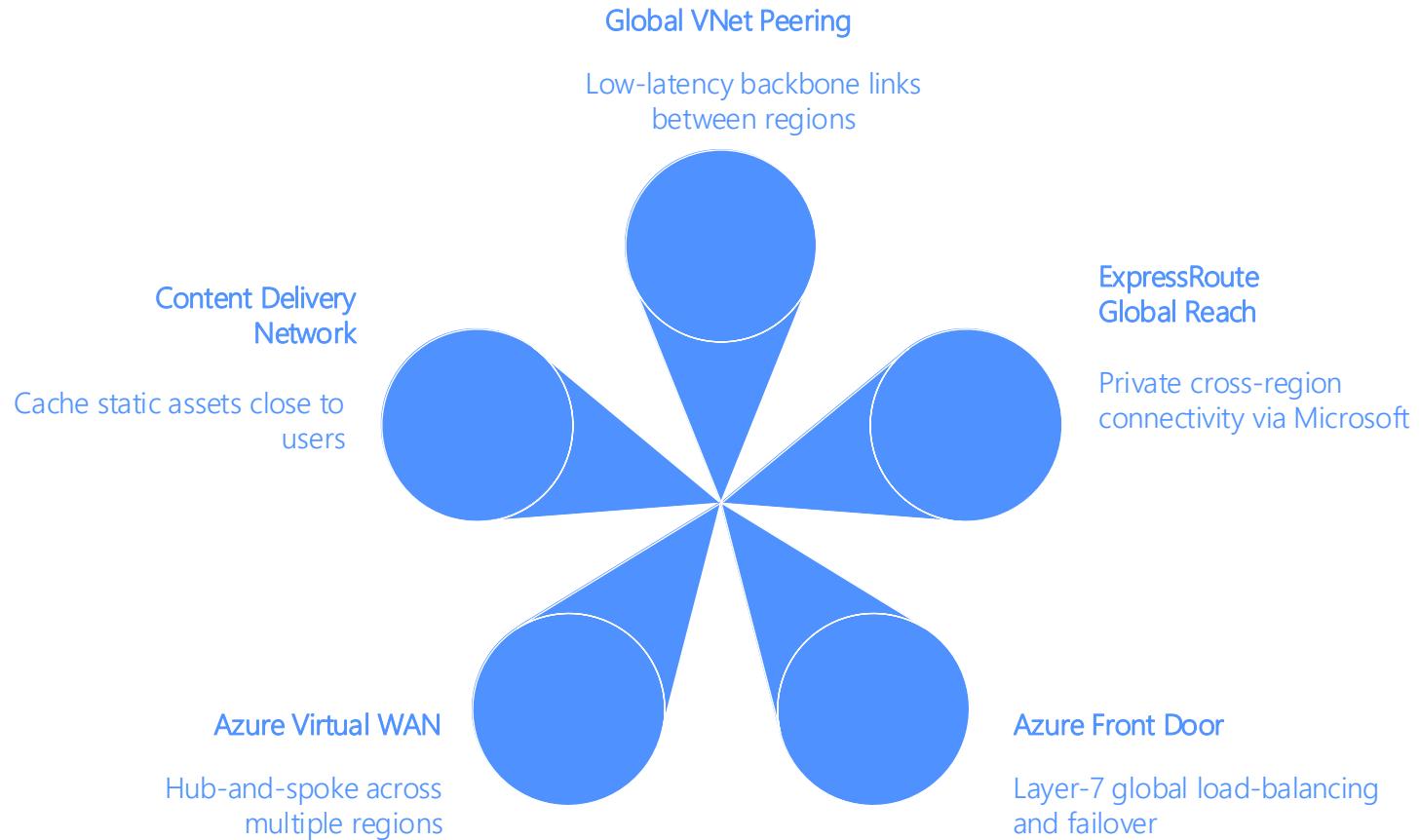
Your Log Analytics workspace is growing 50 GB/day, and costs are soaring. Which action most **directly** reduces storage bill?

- A. Increase workspace pricing tier
- B. Archive older logs to a storage account**
- C. Enable daily export to Power BI
- D. Turn on Azure Monitor autoscale

NETWORKING



NETWORKING PATTERNS FOR GEOGRAPHIC RESILIENCY



DNS STRATEGIES FOR MULTI-REGION FAILOVER

Azure Traffic Manager Profiles

Geographic, priority, performance, or weighted routing

Alias Records

Map CDN or Front Door endpoints transparently

TTL Tuning

Balance responsiveness against DNS query load

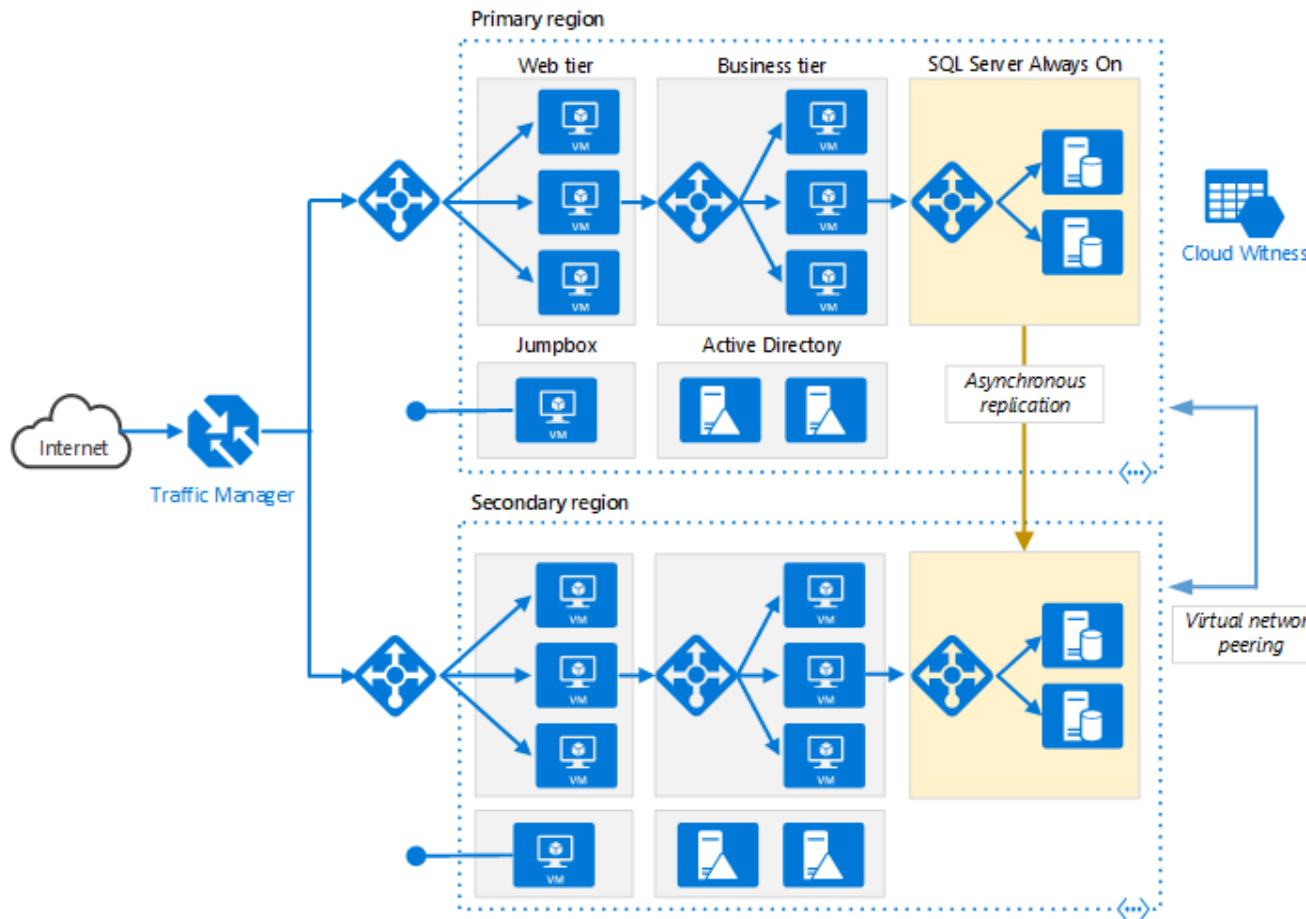
Health Probes

Monitor endpoint health to trigger DNS changes

Private DNS Zones

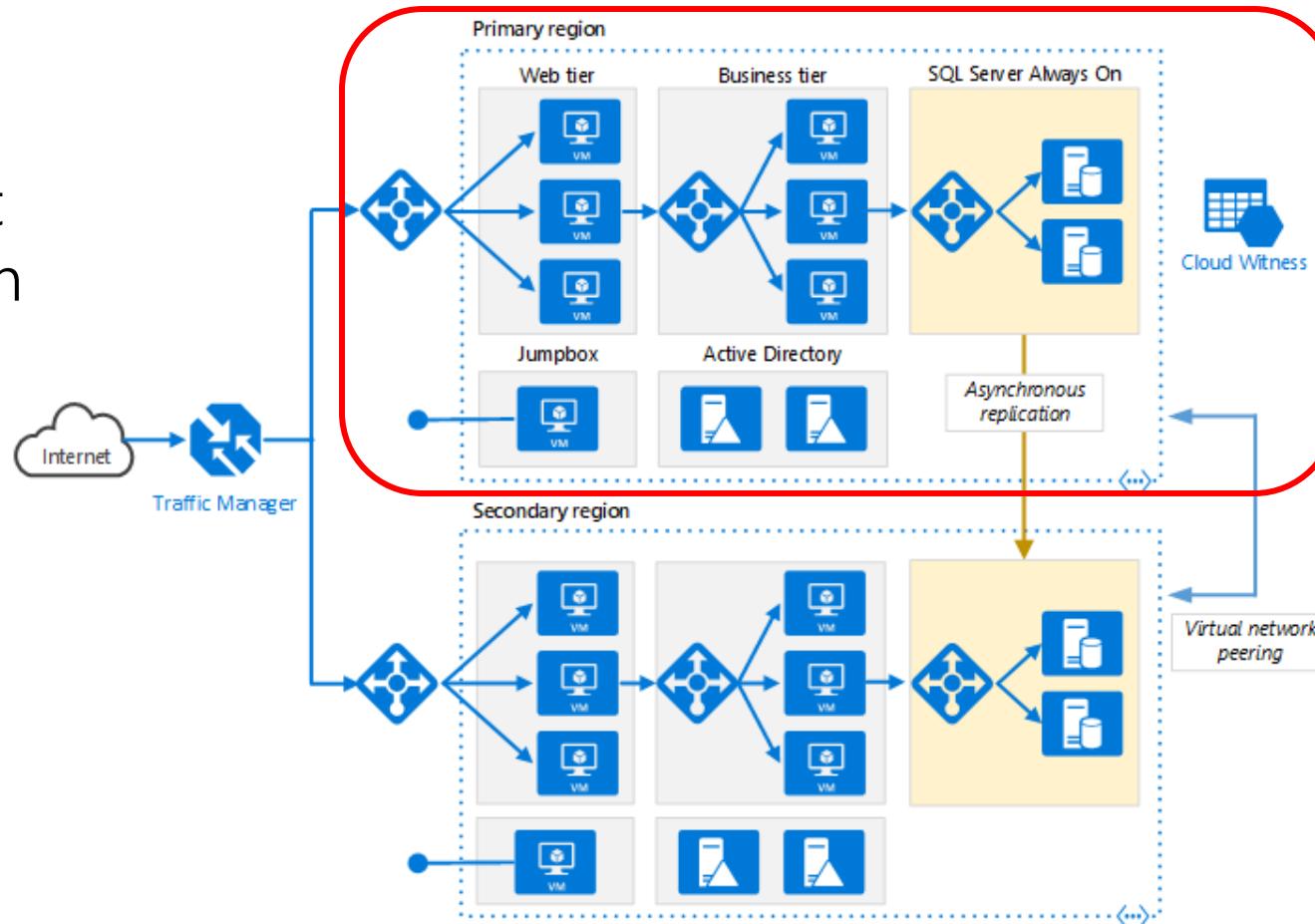
Regional name resolution for VNet-to-VNet scenarios

RESILIENT APPLICATION ARCHITECTURE

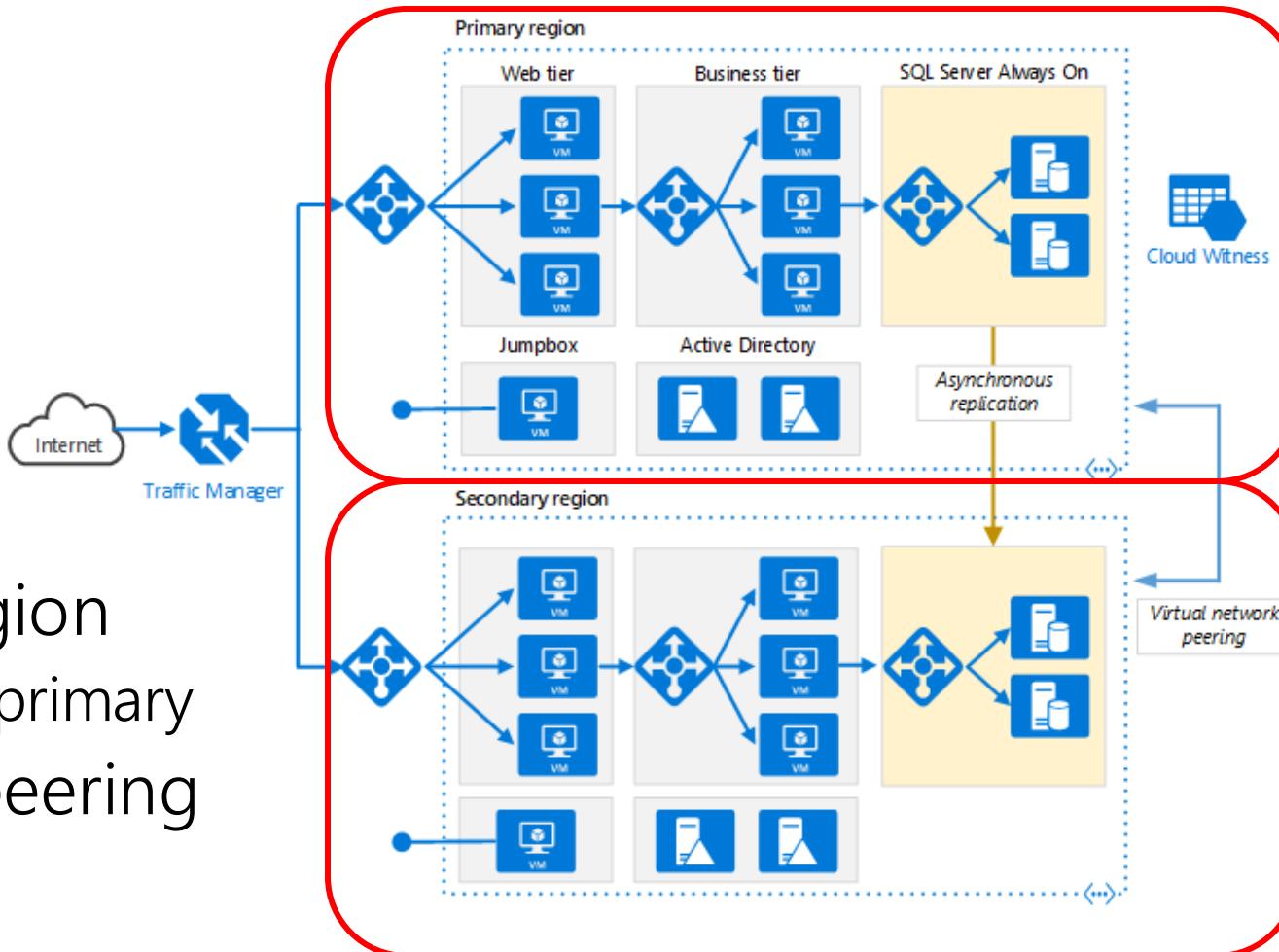


RESILIENT APPLICATION ARCHITECTURE

- Primary region
 - Separate VNet for each region
 - VMs in ScaleSet

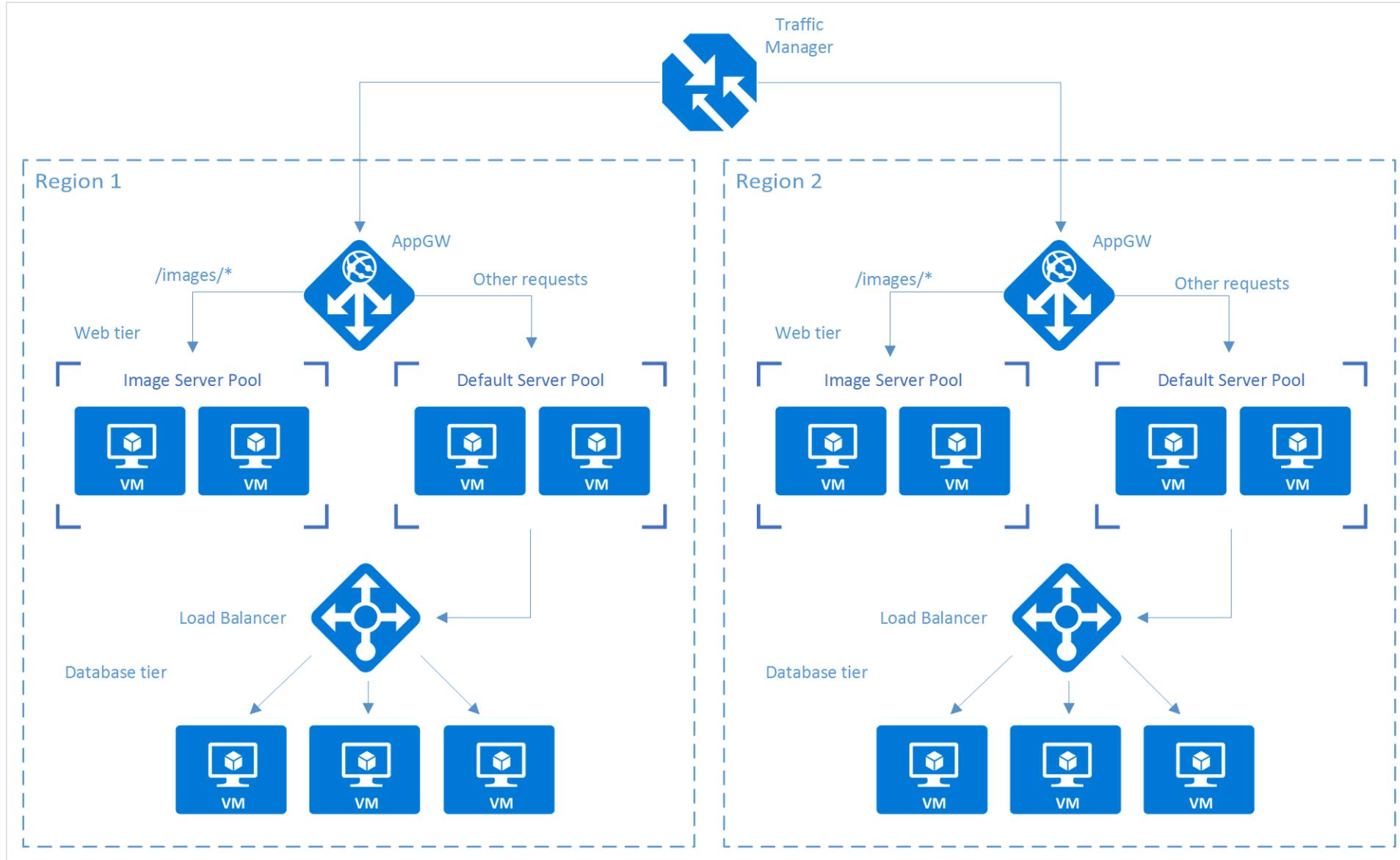


RESILIENT APPLICATION ARCHITECTURE



- Secondary region
 - duplicate of primary
- Global VNet peering

TRAFFIC MANAGER

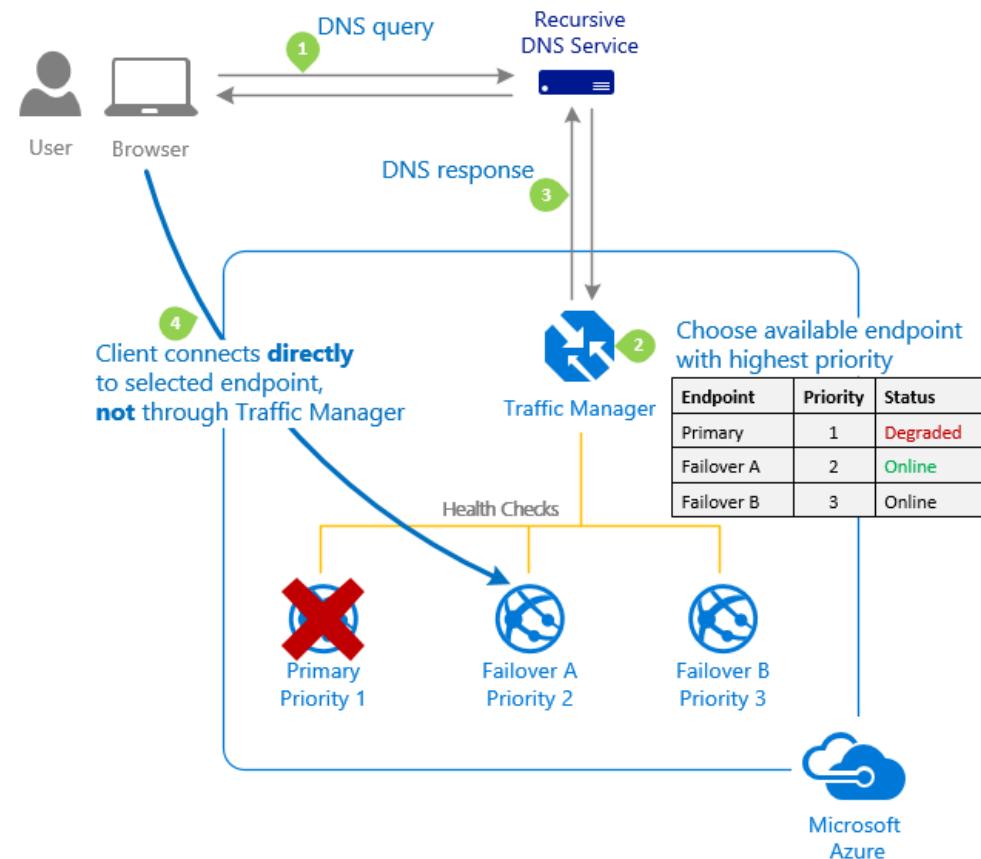


TRAFFIC MANAGER

- DNS-based load balancer
- Distribute traffic across global regions
- Six traffic-routing methods
- Endpoint health monitoring and automatic failover

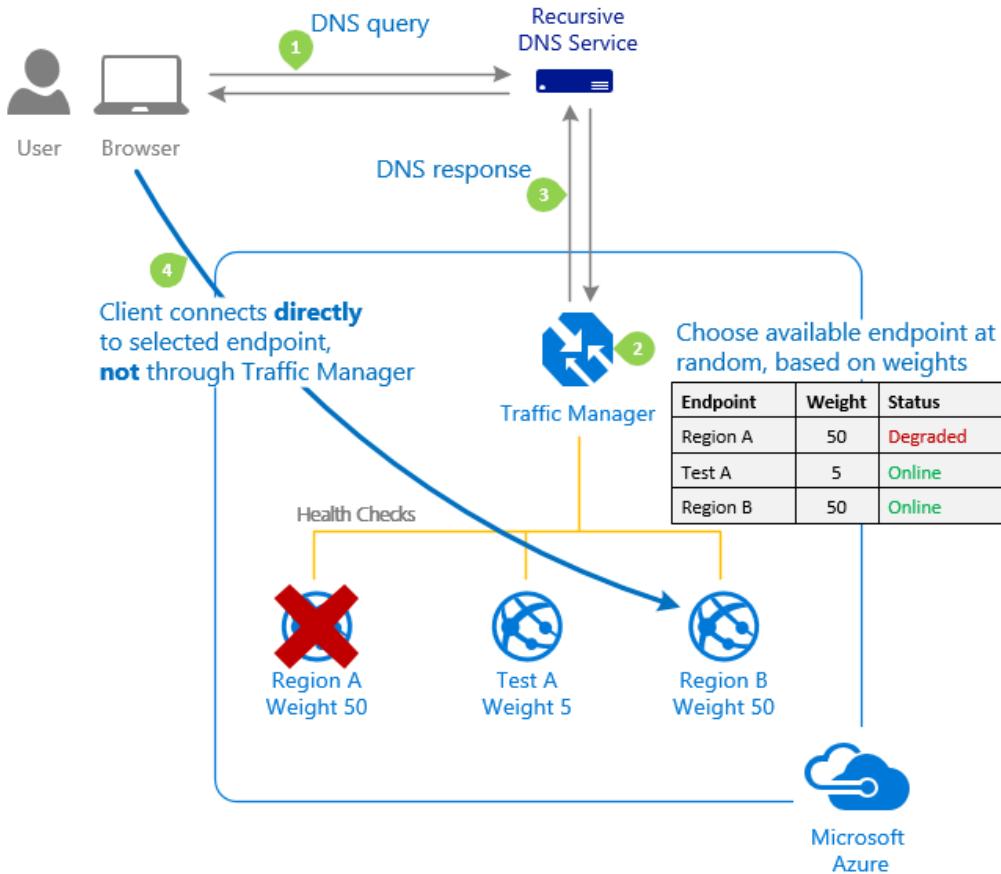
TRAFFIC MANAGER

- Priority method



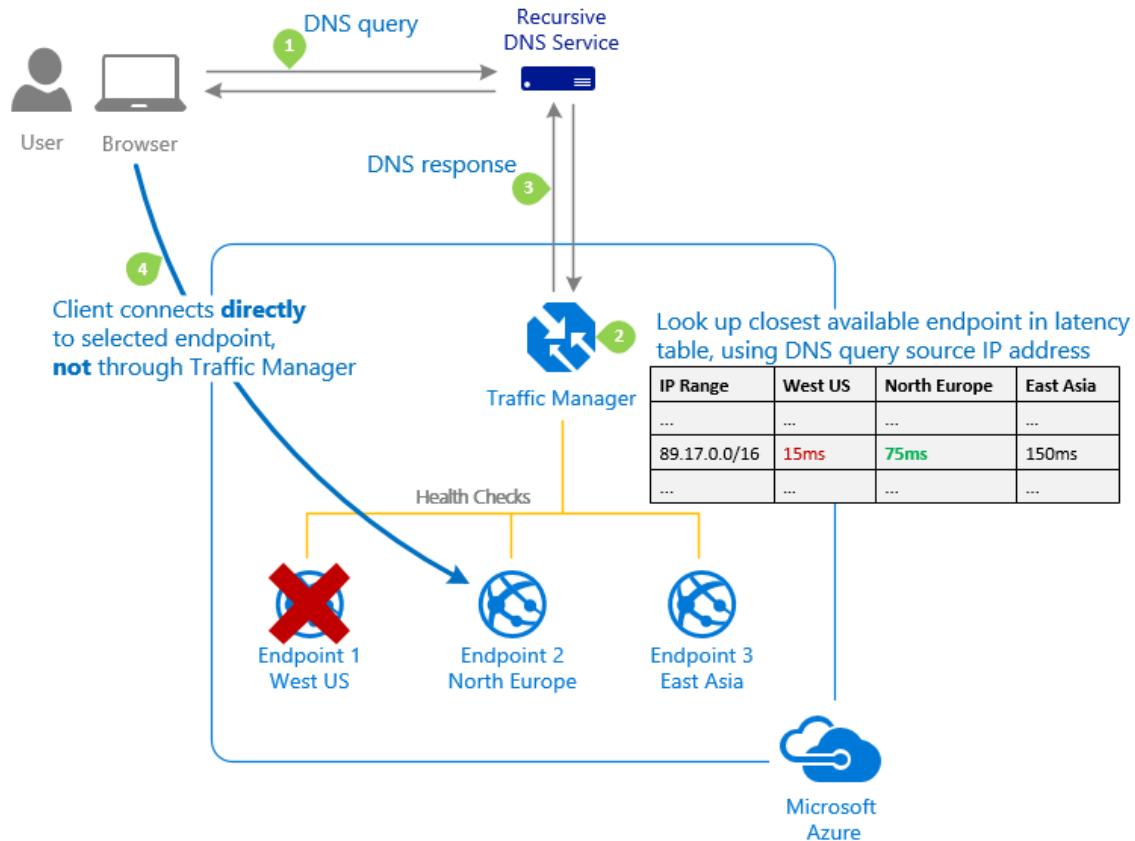
TRAFFIC MANAGER

- Weighted method



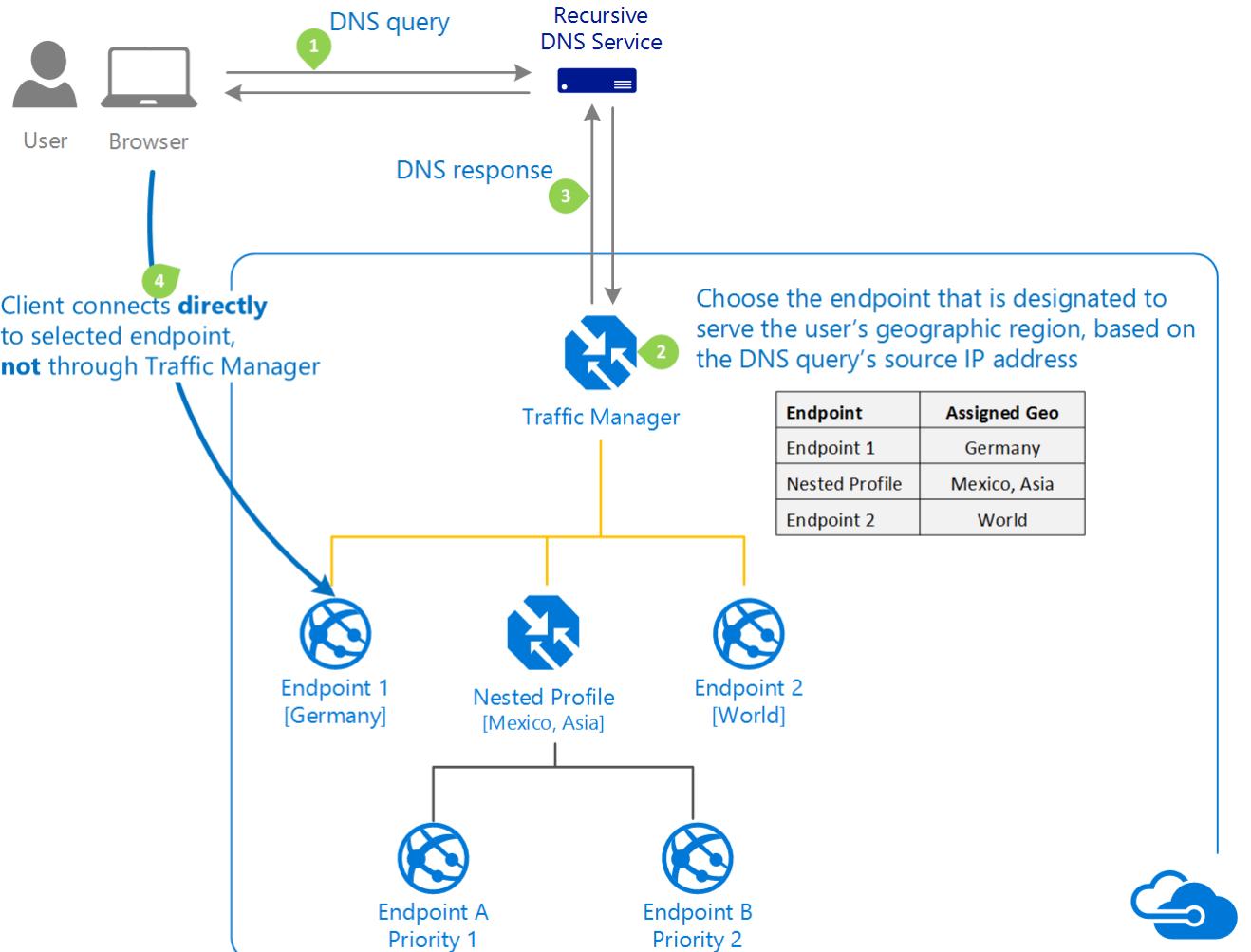
TRAFFIC MANAGER

- Performance method



TRAFFIC MANAGER

- Geographic method



FAILOVER ARCHITECTURES AND BEST PRACTICES

Active-Active vs. Active-Passive

Trade-offs in complexity, cost, and consistency

Health-Based Failover

Automate using probes and Route 53-style policies

Automated Runbooks

Use Automation or Functions to orchestrate failover tasks

Data Replication

Geo-redundant storage, Cosmos DB multi-master, or SQL failover groups

Chaos Engineering

Regularly test failover paths with controlled experiments

LAB: Load Balancer & Traffic Manager

- Deploy Azure VMs with ARM templates
 - Multi-region
- Implement Load Balancing
- Implement Traffic Manager balancing

POP QUIZ:

Your primary region goes down. You want DNS-based failover to kick in within 30 seconds, but avoid excessive DNS query load. What TTL and health-probe settings strike the best balance?

- A. TTL = 30 s, probe interval = 10 s
- B. TTL = 300 s, probe interval = 90 s
- C. TTL = 5 s, probe interval = 60 s
- D. TTL = 60 s, probe interval = 30 s

POP QUIZ:

Your primary region goes down. You want DNS-based failover to kick in within 30 seconds, but avoid excessive DNS query load. What TTL and health-probe settings strike the best balance?

- A. TTL = 30 s, probe interval = 10 s
- B. TTL = 300 s, probe interval = 90 s
- C. TTL = 5 s, probe interval = 60 s
- D. TTL = 60 s, probe interval = 30 s

PRODUCTION DEPLOYMENTS BEST PRACTICES

Infrastructure
as Code

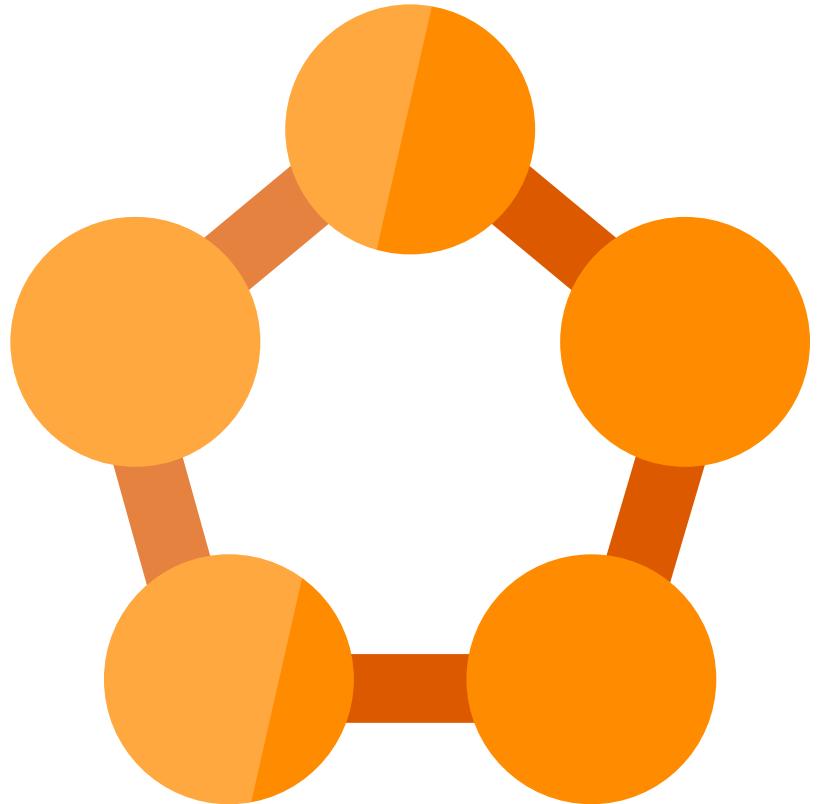
Immutable
Deployments

Blue/Green &
Canary

Health Checks

Automated
Rollback

SERVICE FABRIC



Microservices Platform

Hosts stateful and stateless services at scale

Reliable Actors & Services

Built-in patterns for state and lifecycle management

Cluster Resource Management

Automatic placement, scaling, and healing of services

Rolling Upgrades

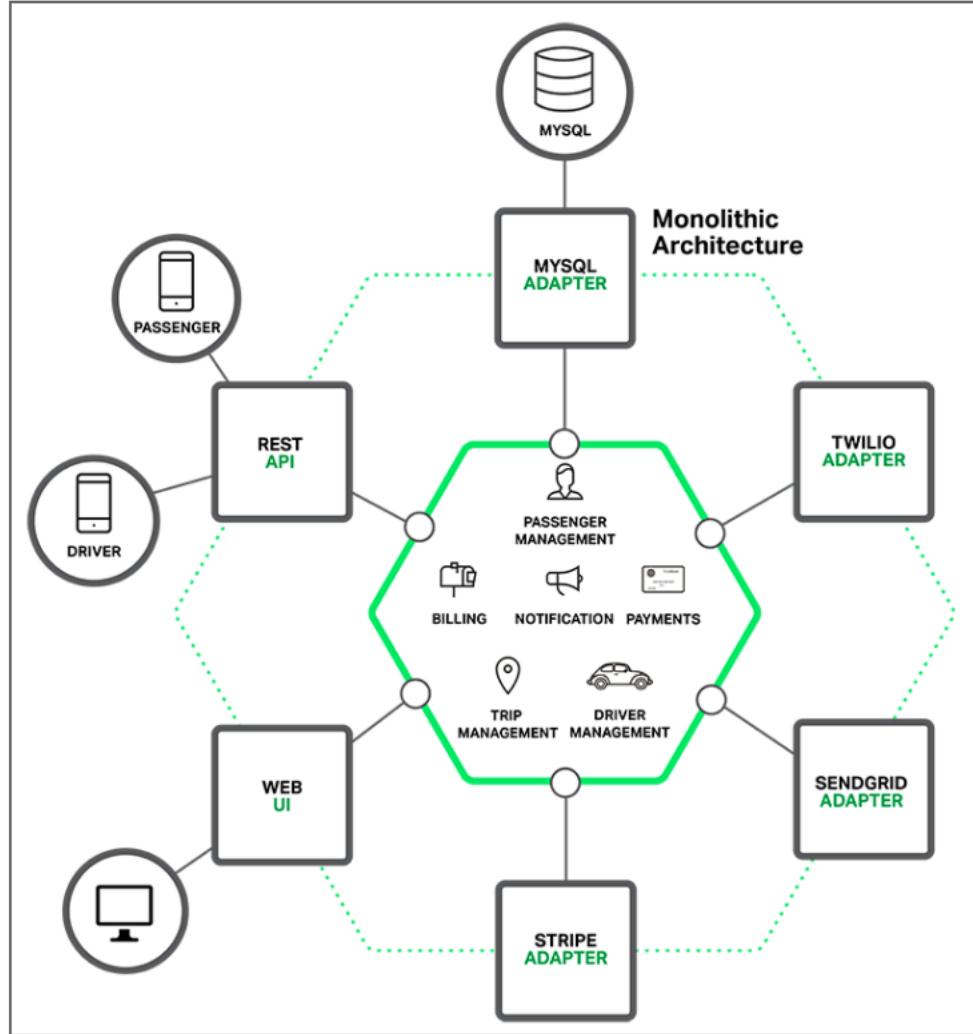
Zero-downtime application and OS patching

Health-Driven Failover

Monitored by Fabric to maintain SLA compliance

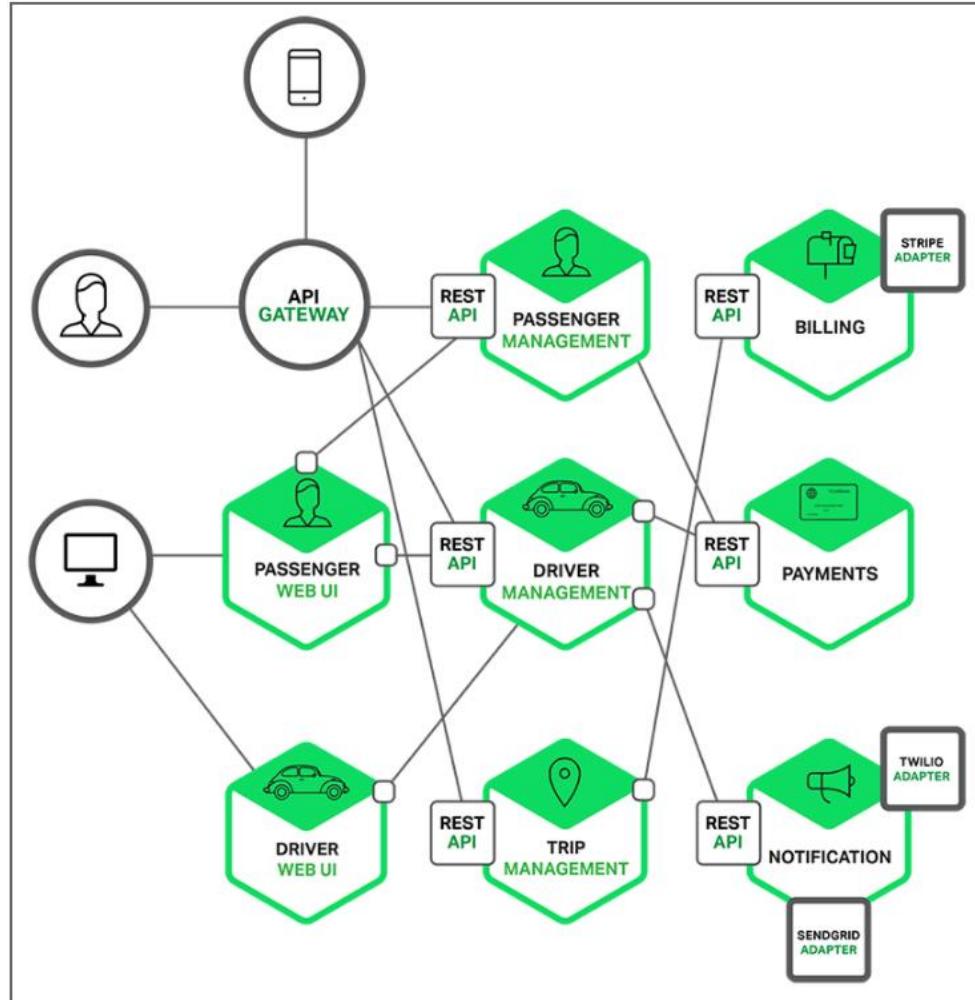
MONOLITHIC APP

- Single development stack
- Changes are difficult
- Updates can lead to downtime



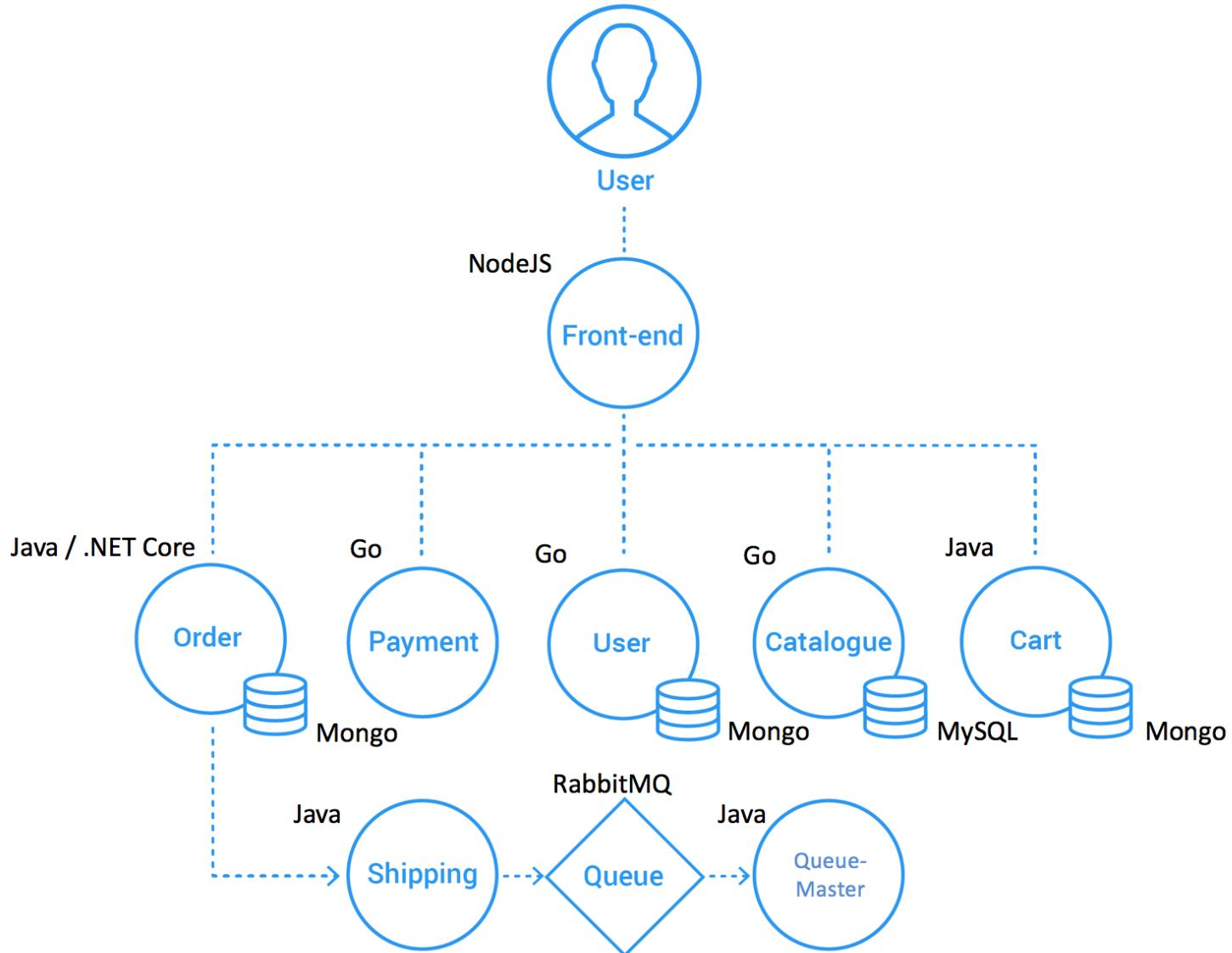
MICROSERVICE APP

- Services communicate with each other over the network.
- Update services independently.
 - No need to change other services.
- Self-contained.
 - You can update the code without knowing anything about internals of other microservices.



MICROSERVICE EXAMPLE APP

- Go
- Spring Boot
- Node.js
- RabbitMQ
- MySQL
- Mongo



AZURE SERVICE FABRIC

Service Fabric: A Microservices Platform

Build Applications with many Languages, Frameworks, & Runtimes

Service Fabric: Microservices Platform

Lifecycle
Mgmt

Independent
Scaling

Rolling
Upgrades

Always On
Availability

Resource
Efficient

Stateless/
Stateful



Public Clouds



On Premises
Private cloud



Developer

AZURE SERVICE FABRIC

- Provide a distributed systems platform that allows developers to build and deploy microservice applications without enduring typical complexities and problems usually associated with microservices application development.
- Rolling upgrades
 - Service Fabric upgrades apps at zero downtime
- Guaranteed availability
 - self-healing, autoscaling

AZURE SERVICE FABRIC

- Shrinking set of use cases
- Service Fabric was used prior to AKS
- Supports containers
- Start migrating to Kubernetes

LOAD BALANCERS & TRAFFIC MANAGER

Azure Load Balancer

Layer 4 regional LB for internal and public traffic

Application Gateway

Layer 7 routing, WAF, SSL offload, and path-based rules

Traffic Manager

DNS-based global traffic distribution with failover and performance routing

Health Probes

Enable granular endpoint health checks for each load-balancer

Hybrid Scenarios

Combine global DNS routing with local load balancing for end-to-end resiliency

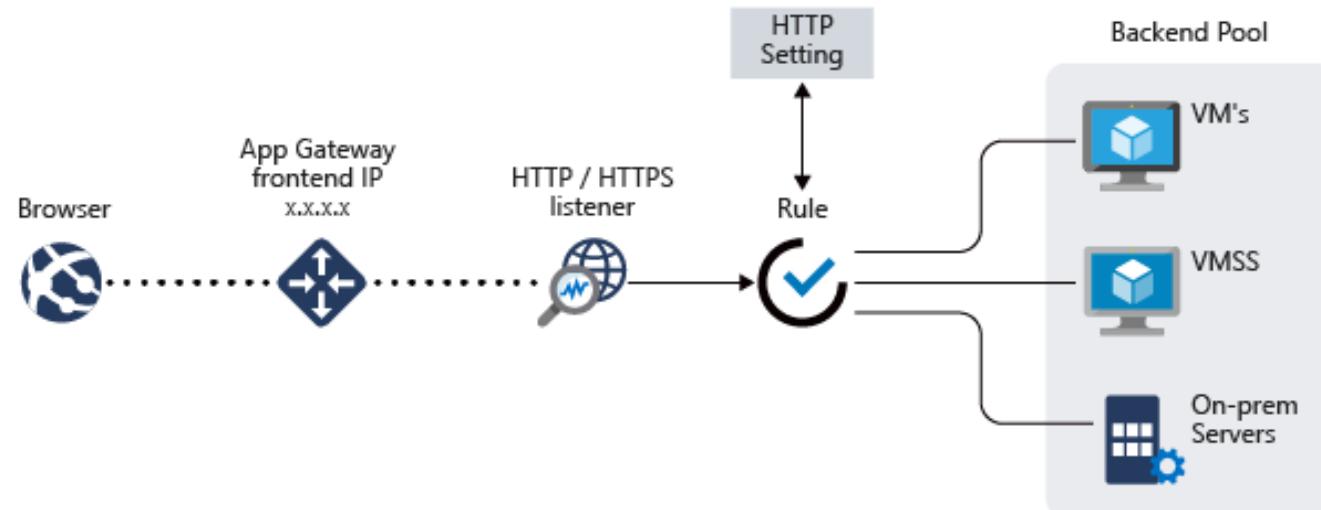
AZURE APPLICATION GATEWAY



Application Gateway's **WAF** offers managed rule sets that defend against common web vulnerabilities like SQL injection and cross-site scripting, as well as the ability to author custom rules for specialized threats—critical for production security.

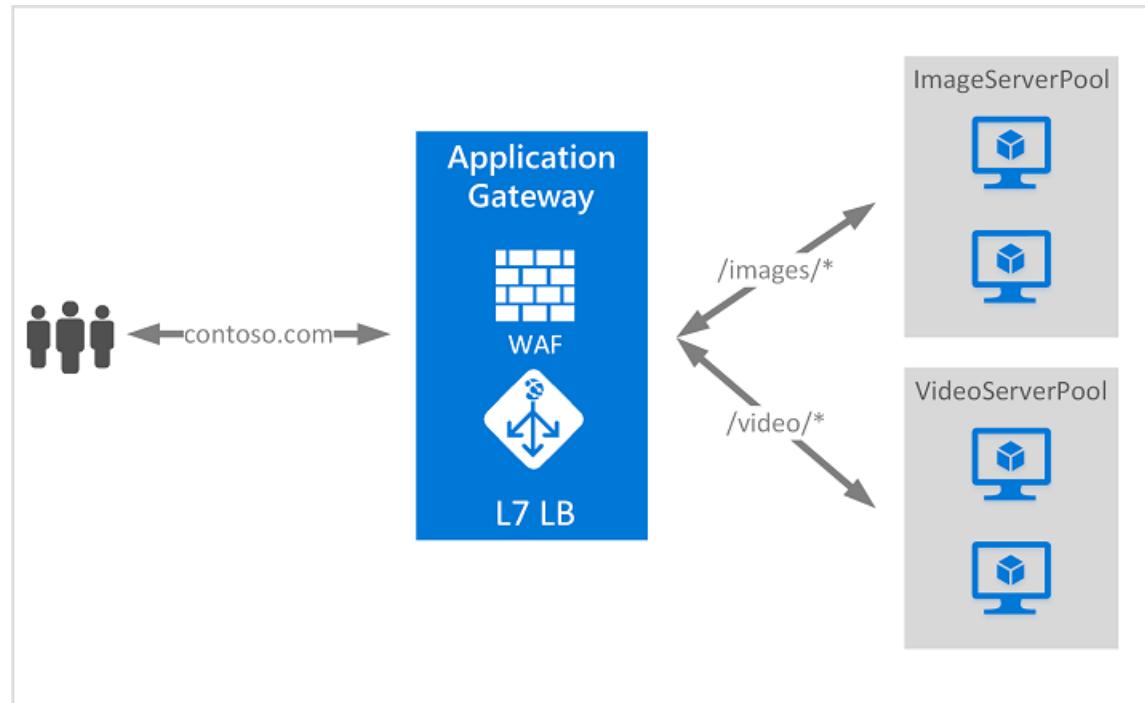
TRADITIONAL LOAD BALANCING

- Layer 4 (TCP & UDP)
- Limited to:
 - source IP, address, port
 - dest IP, address, port



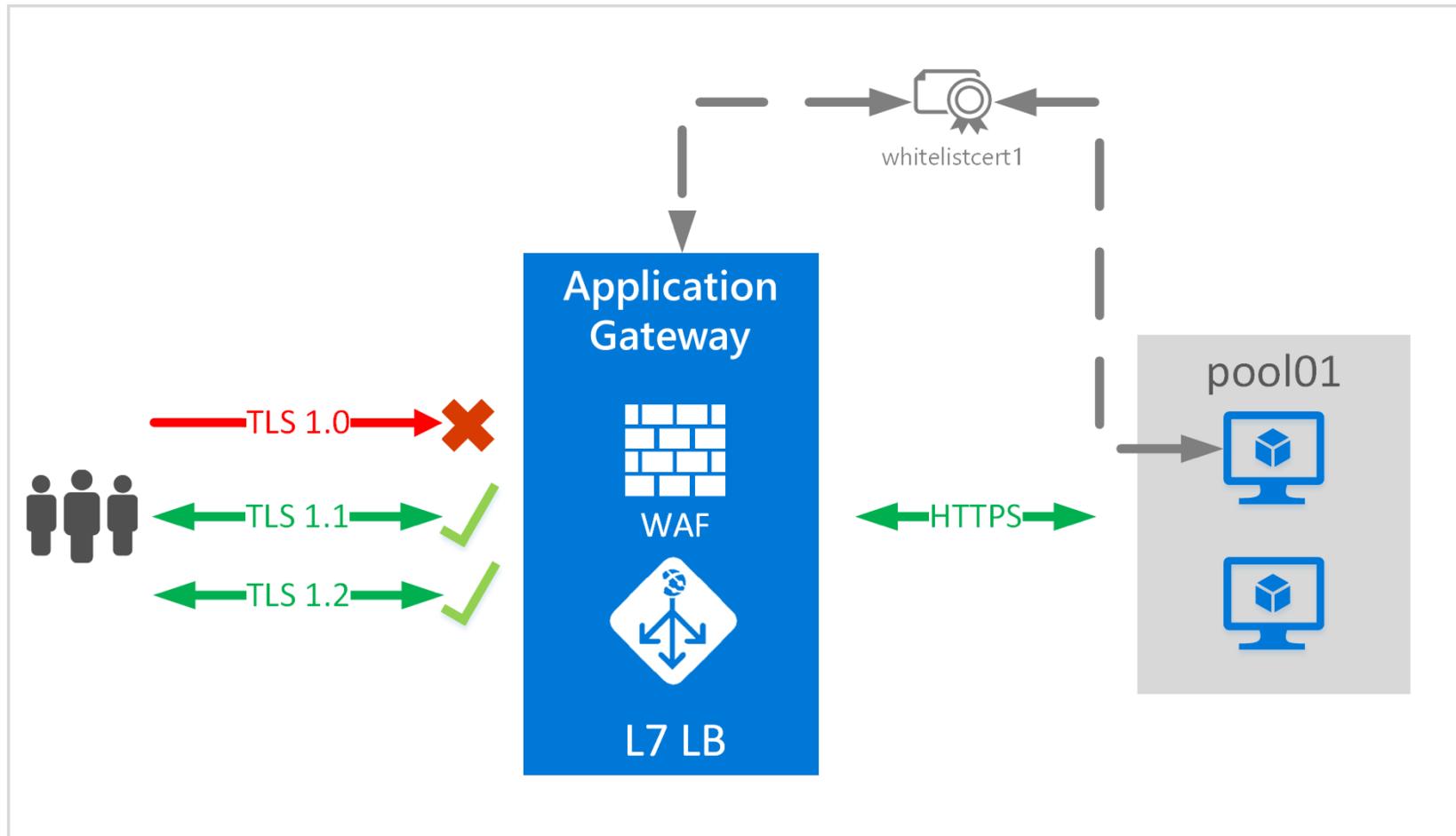
AZURE APPLICATION GATEWAY

- Proxy that can route traffic based on Layer 7 (HTTP)
 - URI path
 - host headers



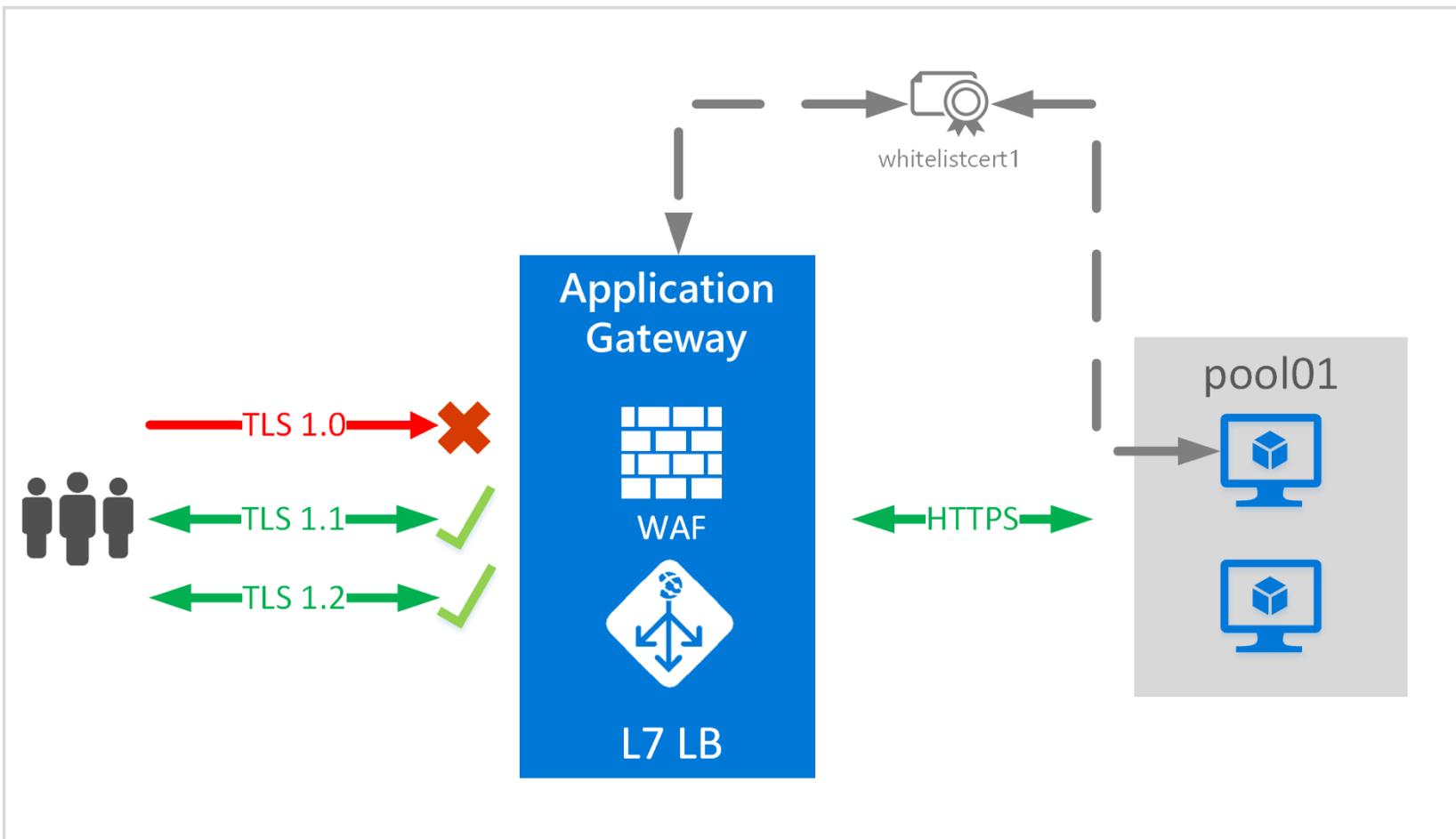
AZURE APPLICATION GATEWAY

- SSL Termination
- Web Application FW
 - up to 100 websites
- Monitoring
 - Azure Security Center



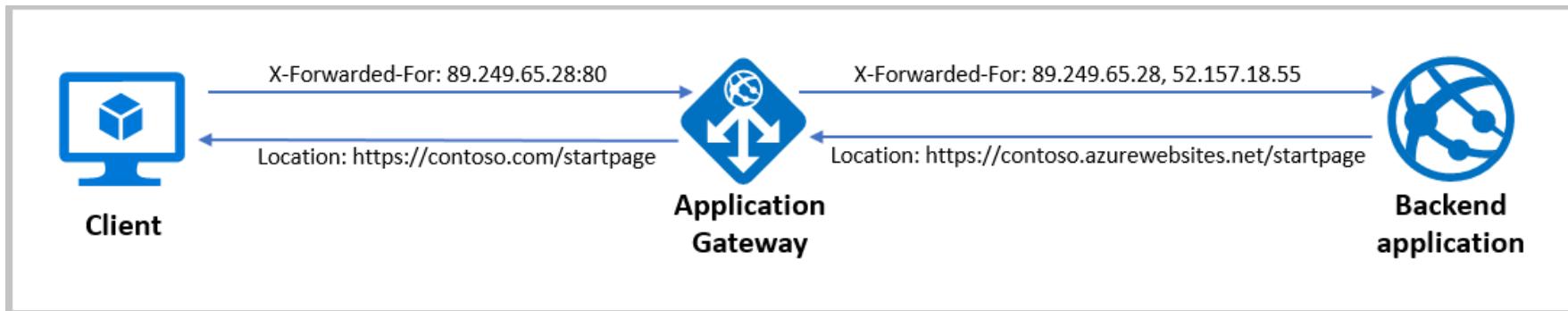
AZURE APPLICATION GATEWAY

- Redirect HTTP -> HTTPS
 - Global or by URL
- Rewrite HTTP headers



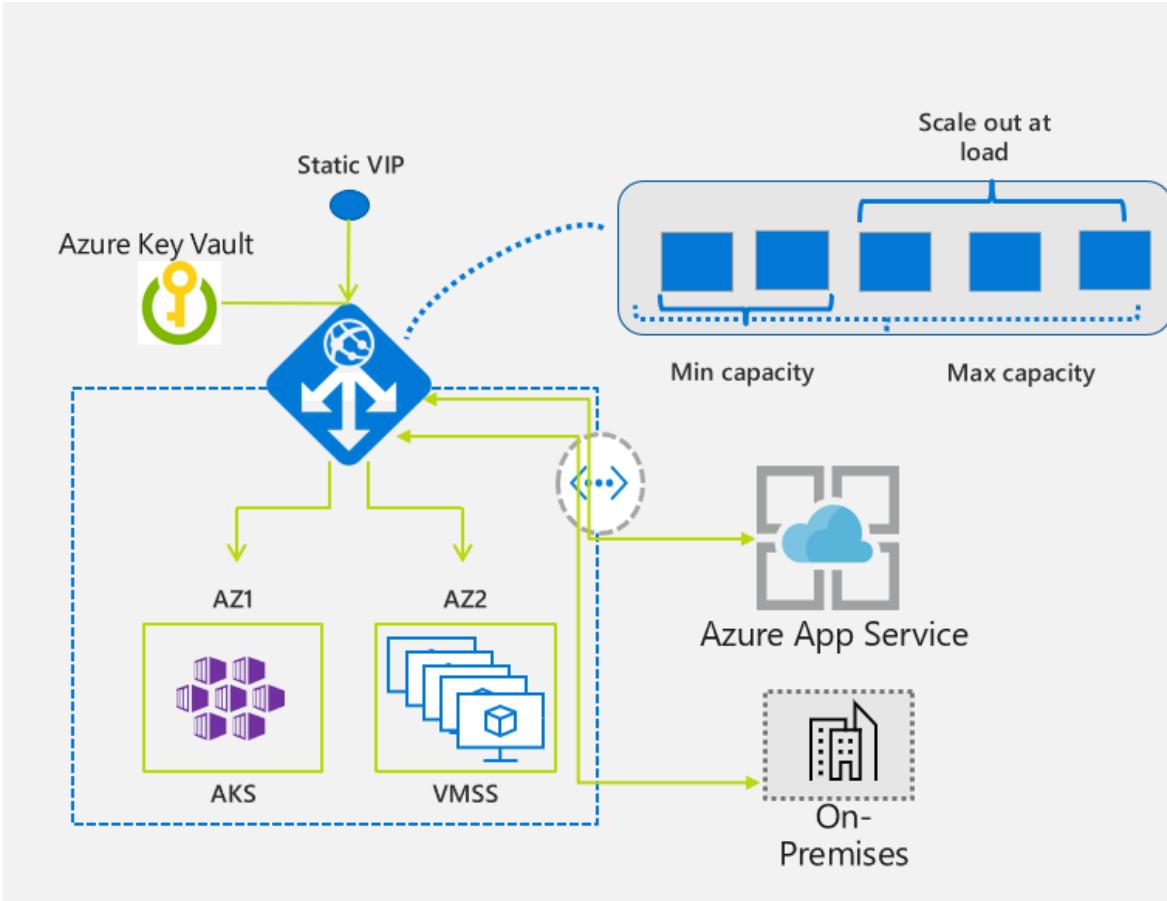
AZURE APPLICATION GATEWAY

- Rewrite HTTP headers

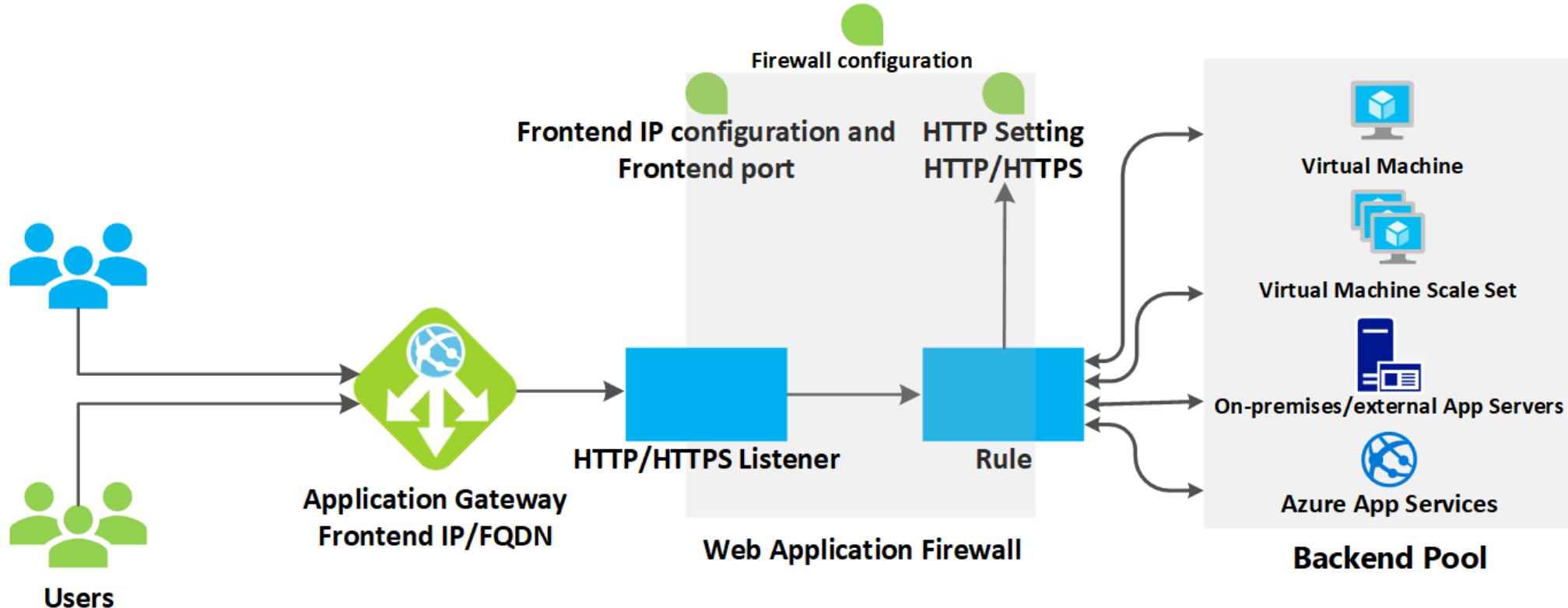


AZURE APPLICATION GATEWAY

- Supports autoscaling
- Spans multiple zones



AZURE APPLICATION ARCHITECTURE



POP QUIZ:

Which approach most directly prevents “configuration drift” in production environments?

- A. Periodic manual audits of server settings
- B. Storing VM images in a shared blob container
- C. Defining all resources in version-controlled ARM/Bicep or Terraform code
- D. Using Azure Policy to enforce tags on resources

POP QUIZ:

Which approach most directly prevents “configuration drift” in production environments?

- A. Periodic manual audits of server settings
- B. Storing VM images in a shared blob container
- C. Defining all resources in version-controlled ARM/Bicep or Terraform code**
- D. Using Azure Policy to enforce tags on resources

POP QUIZ:

In a Canary deployment you want to route 10% of traffic to a new version and monitor errors before full rollout. Which combo of Azure services is *best*?

- A. Traffic Manager + two App Service slots with weighted endpoints
- B. Azure Front Door + single App Service with staging slot swap
- C. Azure Load Balancer + VM Scale Set with health probes
- D. Application Gateway + Service Fabric upgrade domains

POP QUIZ:

In a Canary deployment you want to route 10% of traffic to a new version and monitor errors before full rollout. Which combo of Azure services is *best*?

- A. Traffic Manager + two App Service slots with weighted endpoints
- B. Azure Front Door + single App Service with staging slot swap
- C. Azure Load Balancer + VM Scale Set with health probes
- D. Application Gateway + Service Fabric upgrade domains

COURSE REVIEW

This course has equipped you with the end-to-end skills needed to architect, build, and operate production-grade solutions on Azure. Continue to refine these patterns in your own projects, keep exploring new Azure features, and leverage Azure's rich ecosystem to drive innovation in your organization.

- Azure Core Infrastructure & IaC
Mastered the portal, VNets, storage, and defined environments with ARM/Bicep
- Compute Scaling & Resiliency
Explored VMs (Availability Sets, Scale Sets), Containers (AKS), and serverless compute
- Identity & Access
Integrated on-premises AD with Entra ID, secured apps/APIs via app registrations
- Monitoring & Geo-Resiliency
Implemented Azure Monitor, App Insights, autoscale rules; designed multi-region failover
- Production Deployment Patterns
Applied best practices: IaC pipelines, blue/green & canary, Service Fabric, LB/Traffic Manager, App Gateway

INDIVIDUAL KEY TAKEAWAYS



Write down three key insights from today's session.

Highlight how these take aways influence your work.

Q&A AND OPEN DISCUSSION



