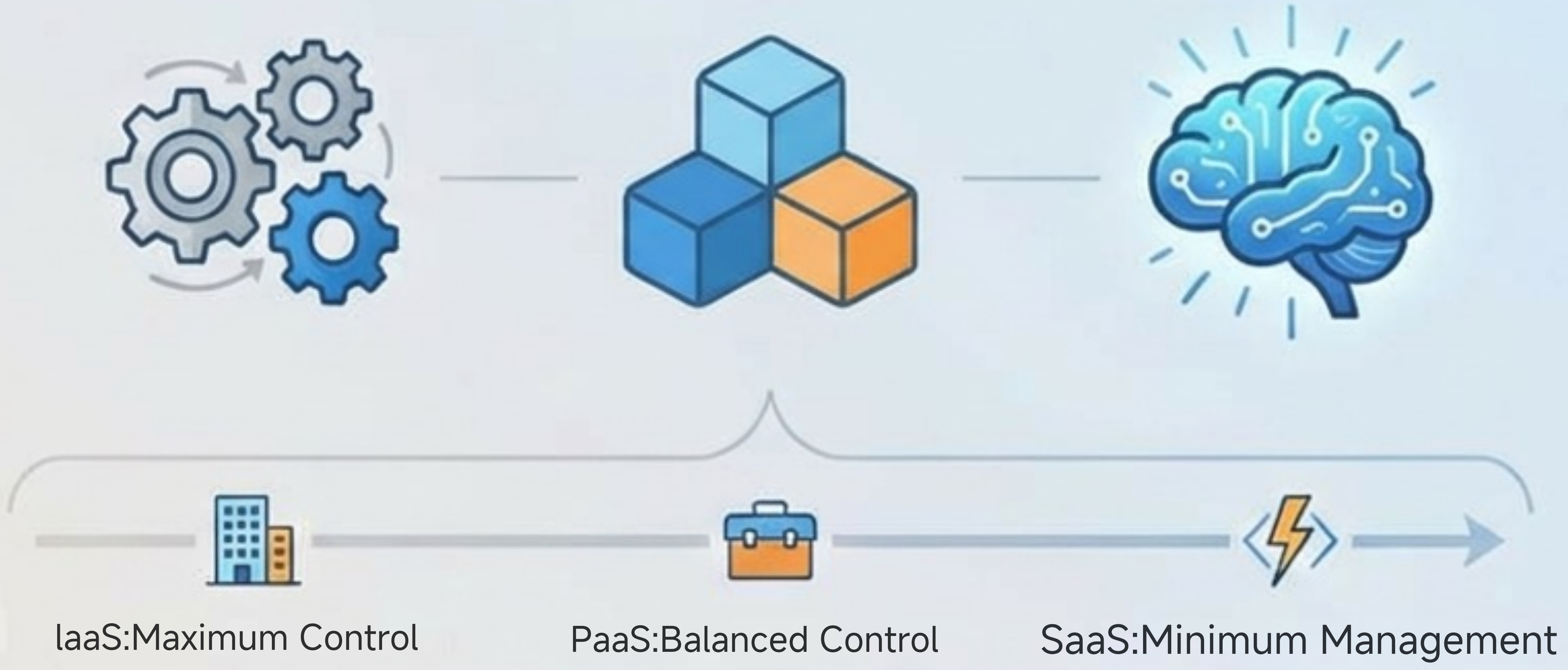
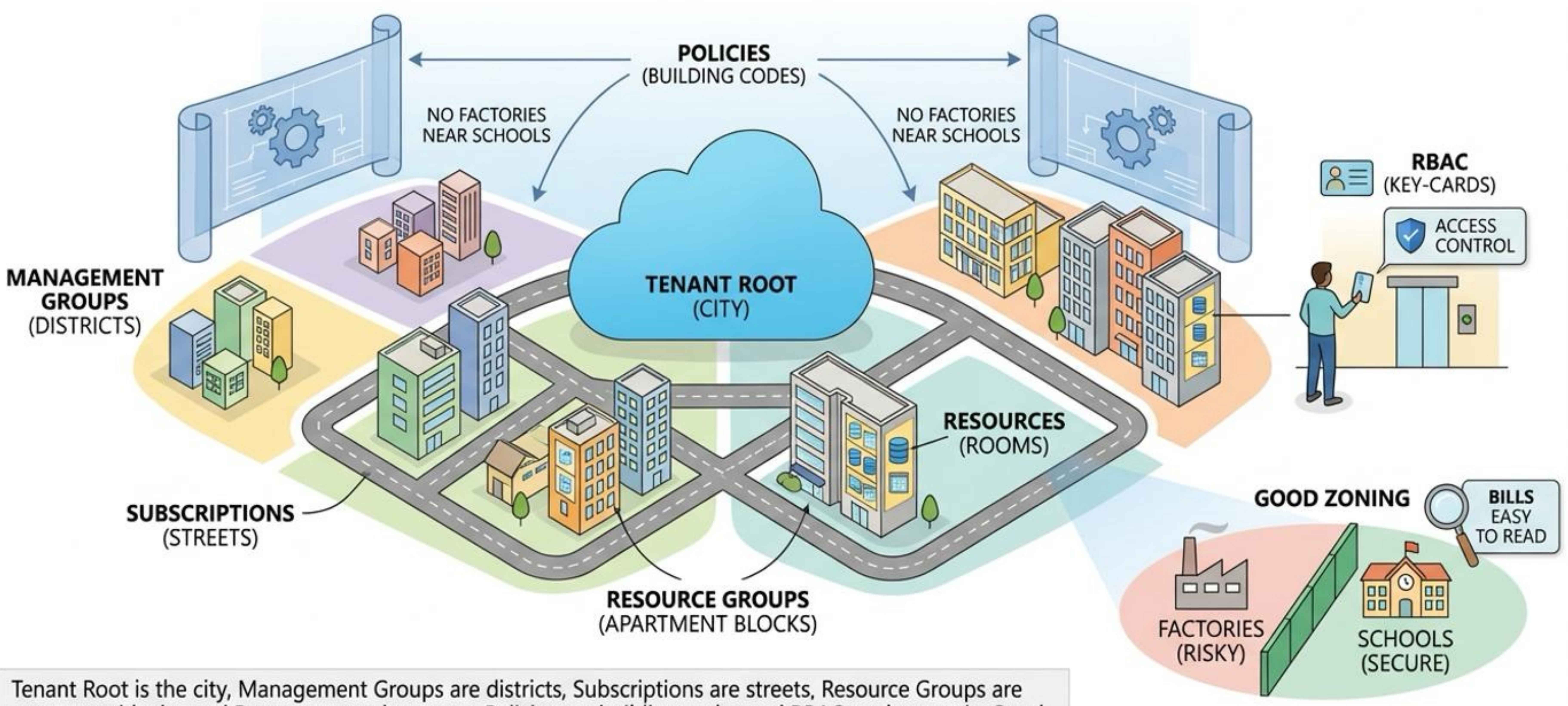
Azure Compute Solutions

Choosing the Right Compute Service for Your Workload



Azure Governance Analogy



Tenant Root is the city, Management Groups are districts, Subscriptions are streets, Resource Groups are apartment blocks, and Resources are the rooms. Policies are building codes and RBAC are key-cards. Good zoning keeps factories away from schools and bills easy to read.

Azure Compute Solutions

Choosing the Right Compute Service for Your Workload

IaaS: Infrastructure as a Service

Like building your own house.

- You manage VMs, OS, & runtime.
- Full flexibility.
- More maintenance.

IaaS: Maximum Control

PaaS: Platform as a Service

Like renting a furnished apartment.

- Microsoft manages infrastructure.
- You focus on your app & data.
- Faster deployment.

PaaS: Balanced Control

FaaS: Functions as a Service

Like booking a hotel room for a night.

- Fully managed, serverless.
- Runs only when triggered.
- Pay per execution.

FaaS: Minimum Management

MORE CONTROL / MORE MANAGEMENT ⟷ LESS MANAGEMENT / MORE AUTOMATION

Azure Compute Services: Real-World Analogies

Understanding Services through Everyday Concepts

Virtual Machines (VMs)

A bare server to decorate. You manage everything from the OS up.

App Service

A furnished flat. Ready to move in, utilities included, Microsoft manages the building.

Container Instances (ACI)

Pop-up kiosks. Quick setup for short-term, specific tasks.

Azure Kubernetes Service (AKS)

A shopping mall. Manages multiple stores (containers) and resources at scale.

Azure Functions

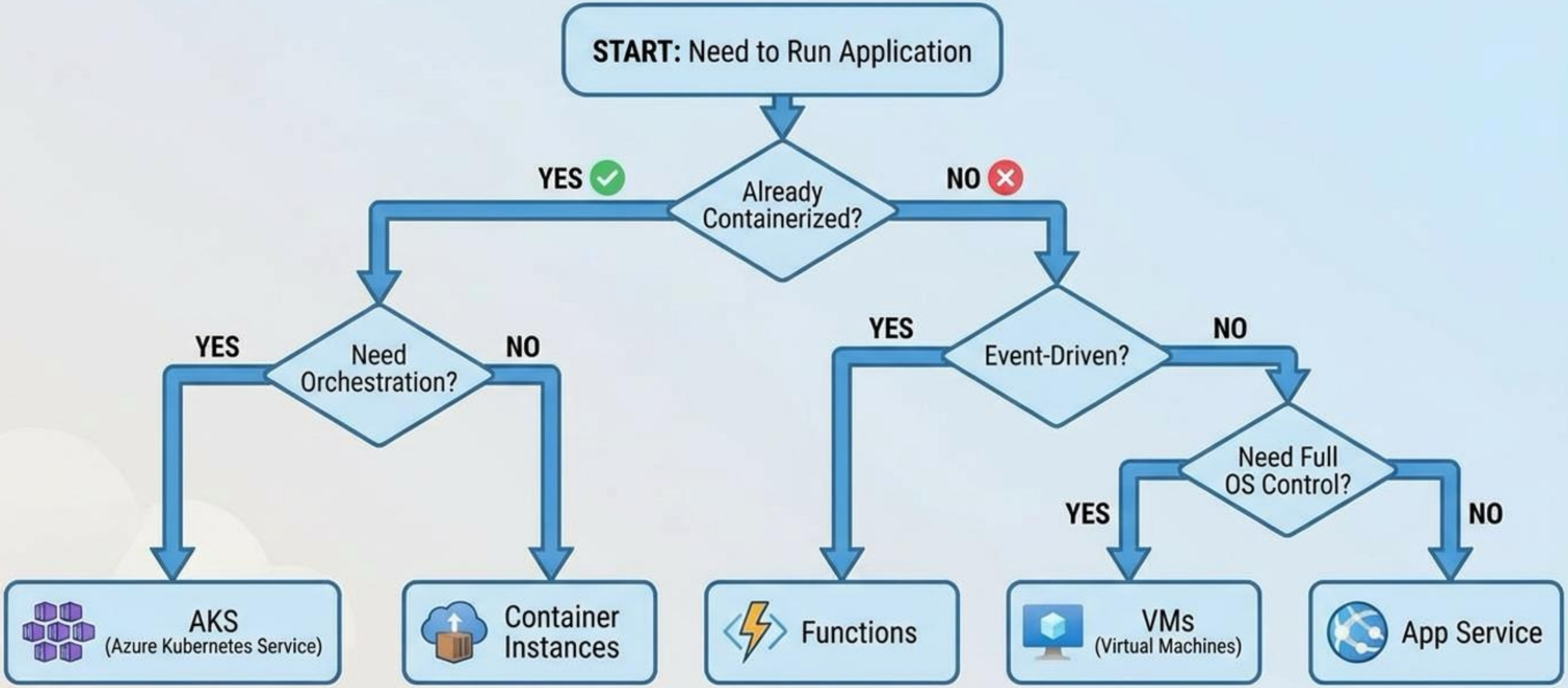
Vending machines. Event-driven, pay only for what you use, when you need it.

Logic Apps

Conveyor belts. Automates workflows and connects different services smoothly.

💡 Start simple, add complexity when needed. Choose the service that best fits your workload requirements.

How to Choose Your Compute Service



Azure Virtual Machines: Full Control IaaS

Planning Your VM Deployment: A Practical Checklist

1. Network & Naming

Begin with VNet, subnet, and public/private IP strategy. Define a consistent naming convention.

2. Region Selection

Pick a region close to your users for low latency. Consider data residency requirements.

3. Managed Disks

Choose Premium SSD for production. Select OS and Data disk based on IOPS needs.

4. VM Size Family

Select size: B (dev), D (general), E (memory), N (GPU). Match to workload requirements.

5. Updates & Monitoring

Turn on Windows Update schedule. Set up Azure Monitor, Boot diagnostics, and alerts.

6. Right-Size First

Start smaller, scale up later. Use B-series for dev/test to save costs (like running a smaller car).

💡 **Cost Tip:** Right-sizing saves money! A smaller car costs less to run. Start small, then scale up as needed.

Azure VM Sizes: The Car Analogy & Right-Sizing

B-series

Economy Hatch-back

For burstable, low-cost workloads (dev/test). Small cargo.

D-series

Family Sedan

General purpose (web servers, small databases). Balanced cargo.

F-series

Sports Coupe

Compute optimized (gaming, analytics). Speed cargo.

E-series

Moving Van

Memory optimized (large databases, in-memory caching). Heavy memory cargo.

L-series

Delivery Lorry

Storage optimized (big data, data warehousing). High disk I/O cargo.

N-series

Gaming Rig

GPU optimized (graphics, AI/ML). Graphics/ML cargo.

H-series

F1 Racer

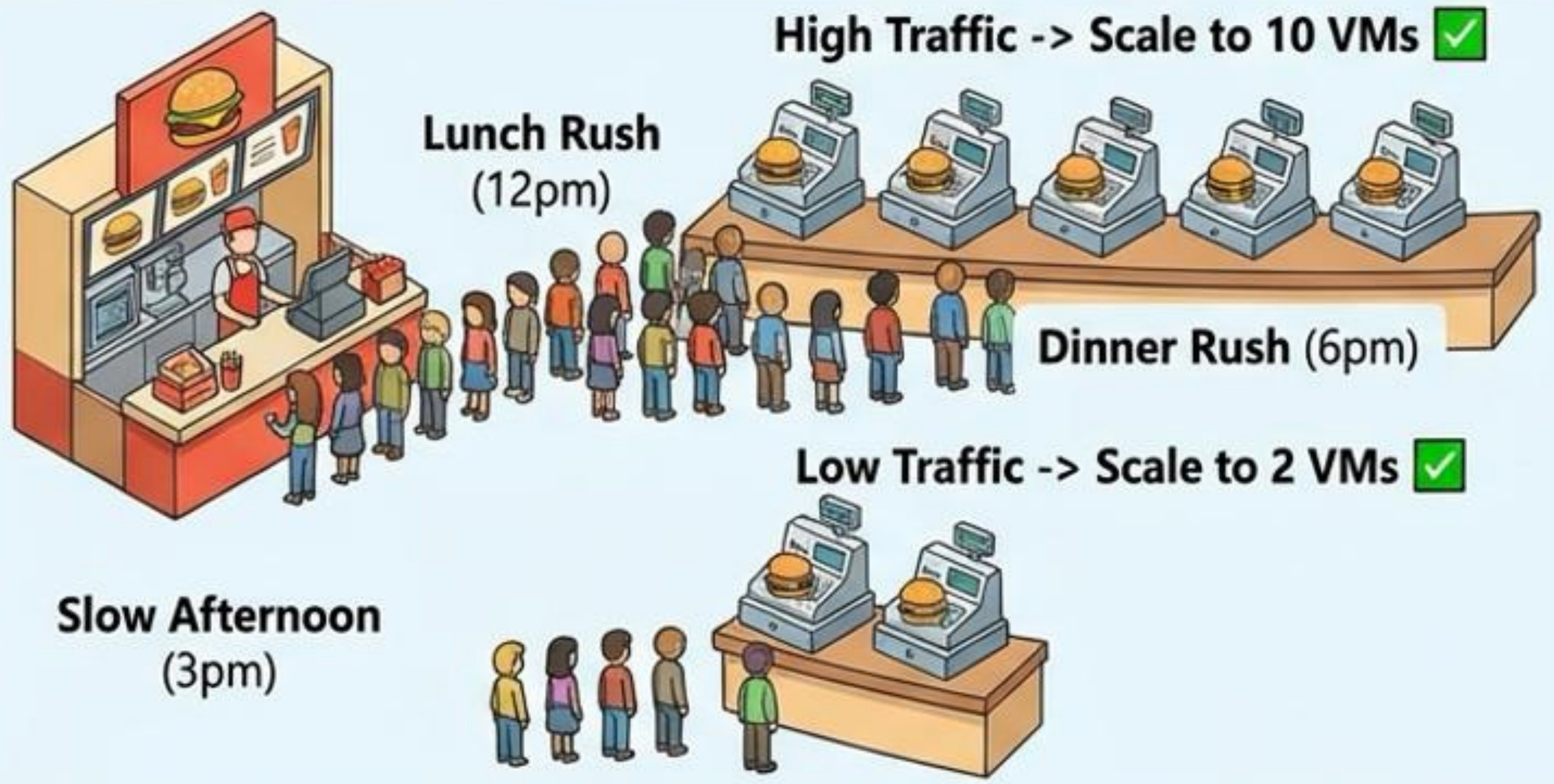
High Performance Compute (supercomputing). Extreme speed cargo.

Key Takeaway: Match the car to the cargo and you never pay for horsepower you don't use! Right-sizing saves money.

Virtual Machine Scale Sets (VMSS)

Auto-Scaling Made Easy

Real-World Analogy: Fast Food Restaurant


High Traffic -> Scale to 10 VMs ✅
Lunch Rush
(12pm)
Dinner Rush (6pm)
Low Traffic -> Scale to 2 VMs ✅
Slow Afternoon
(3pm)

Health Probe

Health checks remove unhealthy VMs & replace automatically

💰 **Saves money when not needed!**

When to Use VMSS ✅

- ✓ Variable traffic patterns
- ✓ Need automatic scaling
- ✓ Stateless applications
- ✓ 10+ VMs of same type

When NOT to Use ❌

- ⊗ Need unique VMs with different configs
- ⊗ Stateful applications requiring data persistence
- ⊗ Only need 1-2 VMs

Key Features

Auto-Scaling
Rule-based scaling, e.g., CPU > 80%

Load Balancer Integration
Traffic distributed, health checks

All VMs Identical
Same configuration, software, updates

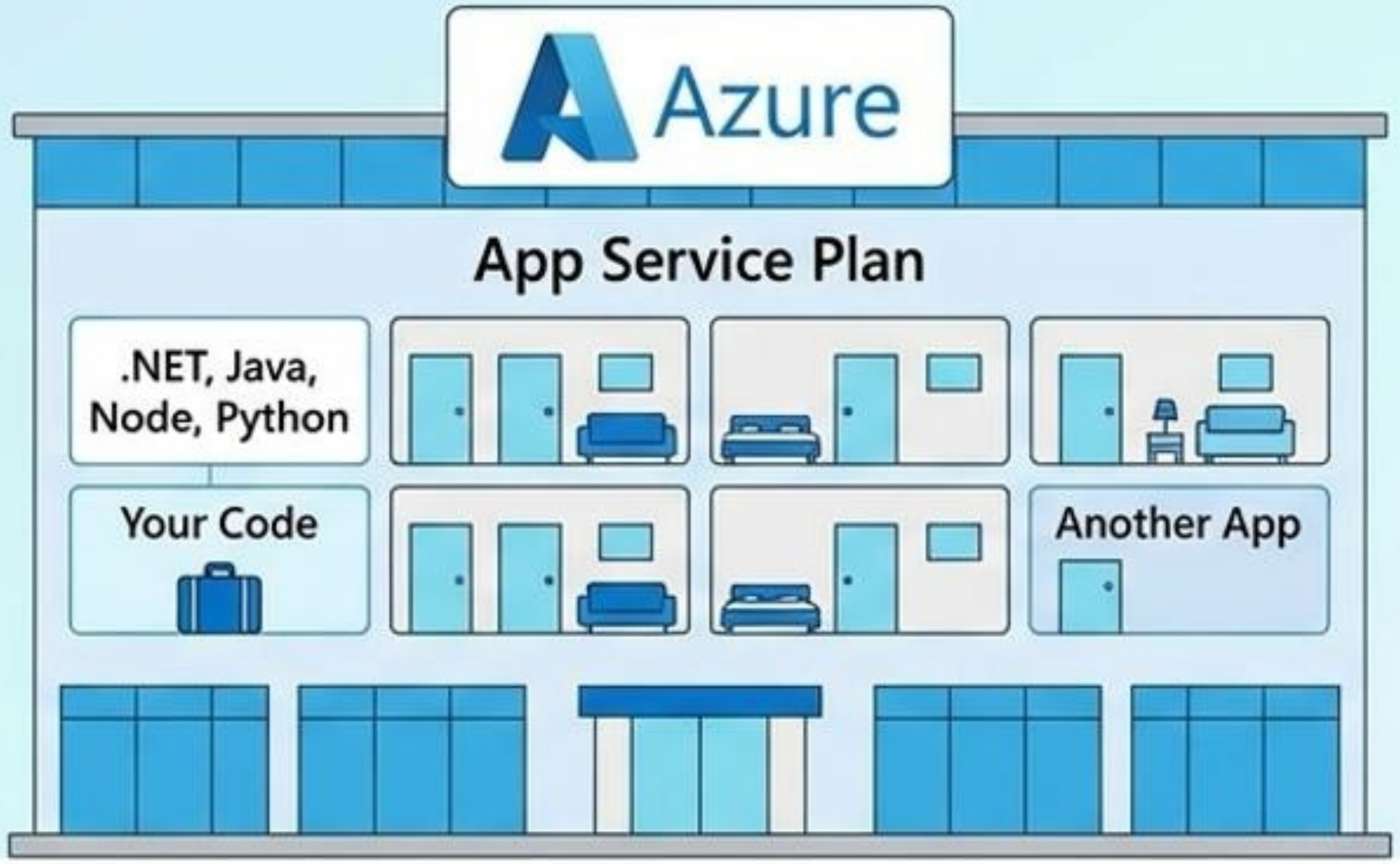
High Availability
Spread across Availability Zones, 99.99% SLA

Cost Benefit 💰

Traditional (Always 10 VMs): \$500/month × 10 = \$5,000/month

With VMSS Auto-Scale: Average 4 VMs, \$500/month × 4 = \$2,000/month.
💰 **Savings: \$3,000/month (60%!)**

Azure App Service: Your Fully Managed Hotel



Move-in ready: plumbing, power, security included.

Bring your code suitcase and you're online in minutes.

Swap staging & production doors for zero-downtime updates.

Concierge scales the floors when the party gets busy.

💡 You focus on code, Azure handles the rest.

Understanding Containers: The Shipping Container Analogy

Physical Shipping Container 🚢

Standardized Size, Fits Any Ship

Fits Diverse Contents (Furniture, Electronics, Food)

Unload at Any Port (Standardized Infrastructure)

Analogy: Standardization & Portability

Software Container 💻

Standardized Format, Runs Anywhere

Packages Your App + Just Needed Libraries

Runs on Any Platform (Azure, AWS, Laptop)

Key Takeaway: Containers create a consistent, portable environment for applications, solving the "it works on my machine" problem by packaging everything the app needs to run.

Azure Container Instances (ACI)

Fast, Simple, Serverless Containers

- Fast Startup
- Per-Second Billing
- Temporary Workloads

WEEKEND COFFEE STAND
OPEN: 1 MINUTE
00:00:58
start
RUN FOR WEEKEND
START: 1 MINUTE
PAY PER SECOND
END: TEAR DOWN

Auto-Scaling: The Shopping Mall Analogy

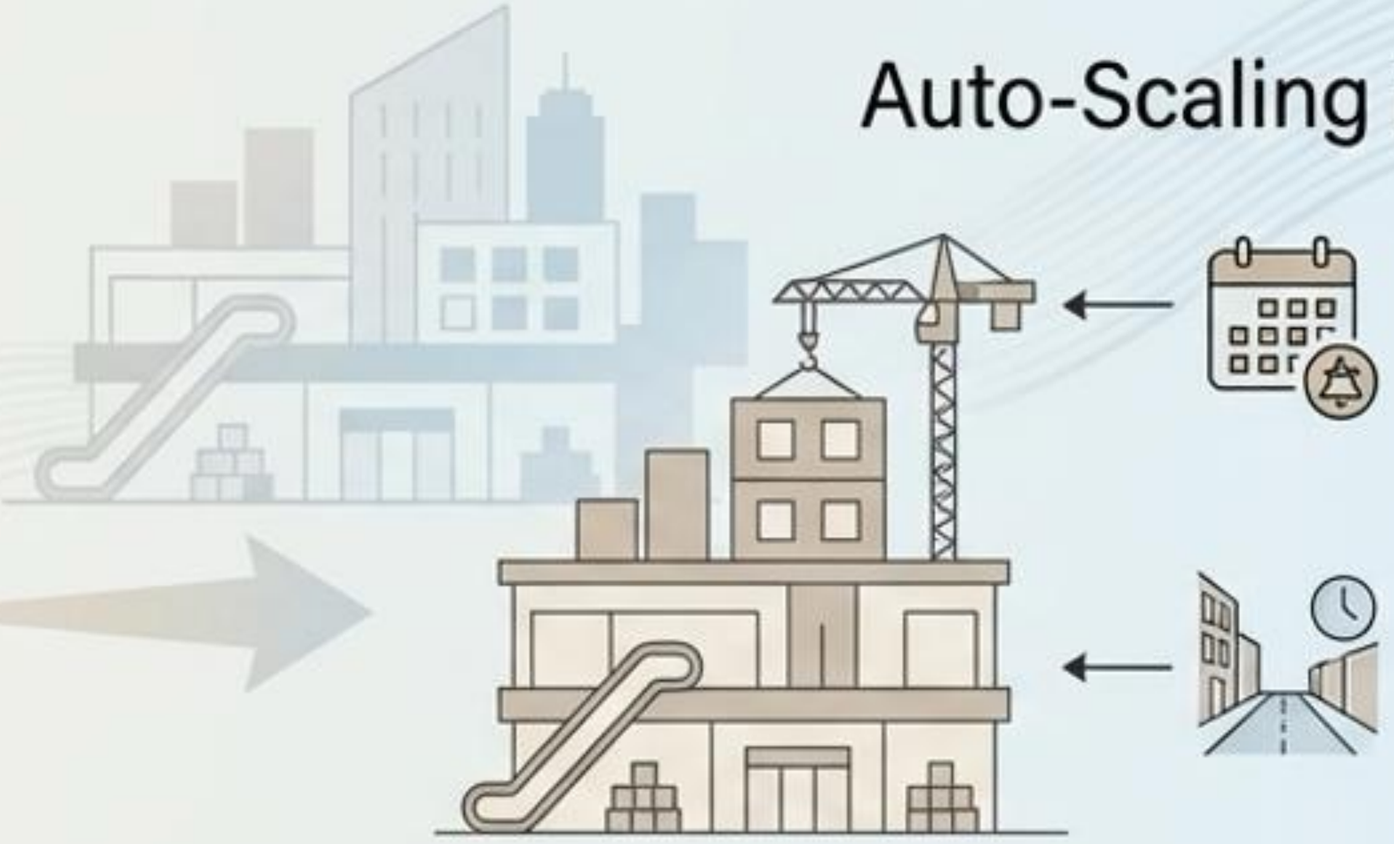
Dynamic Capacity Management for Cloud Workloads

The Platform (Microsoft)

Keeps escalators & fire alarms working.

- You decide which shops open.
- How many staff each needs.
- When to extend opening hours.

Auto-Scaling in Action



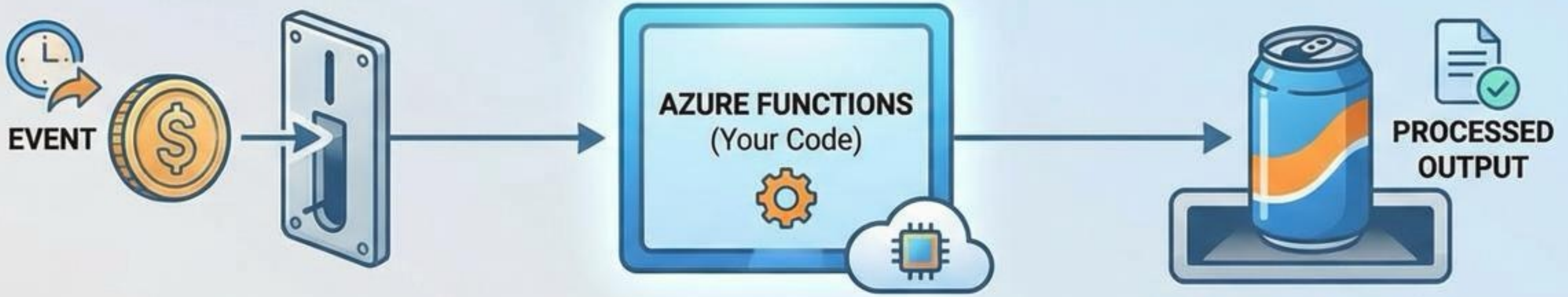
🎄 Adds more floors during the holidays (high demand).

▽ Shrinks back when it's quiet (low demand).

🛍 Shoppers keep browsing seamlessly. No interruptions.

Azure Functions

Serverless Computing



Runs for milliseconds.

Soda pops out—no clerk, no store, no rent.

An event drops into Azure and your code runs for milliseconds; you pay only for the sip, not the shop.

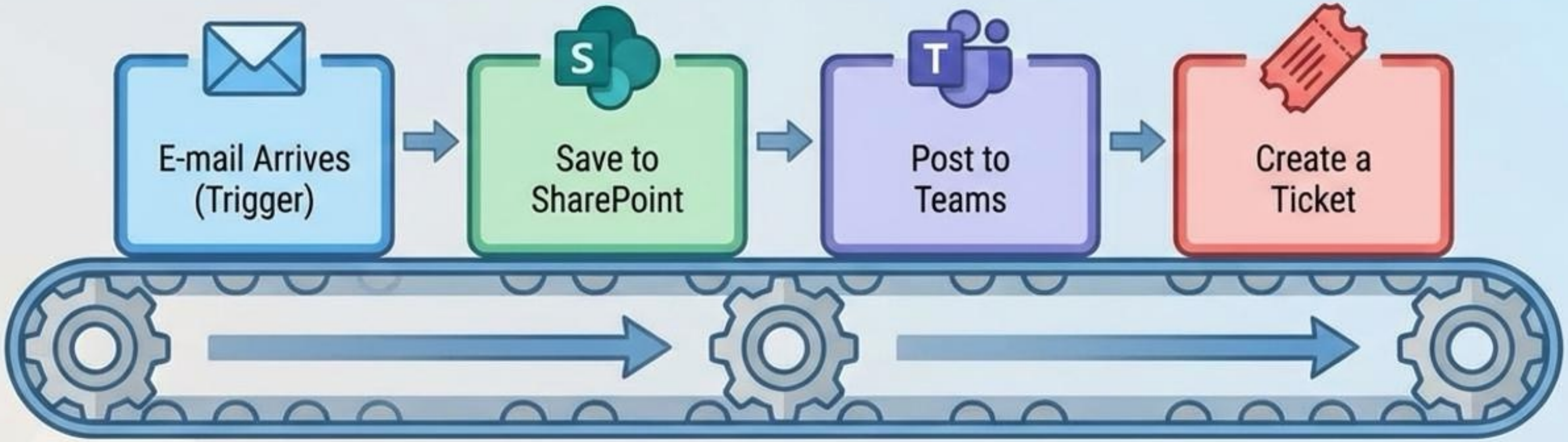
Perfect Use Cases (Short Tasks < 5 min)

- Thumbnail Creation
- API Glue
- Nightly Tidy-up Jobs

Pay-Per-Use | Event-Driven | Instant Scale.

Azure Logic Apps: Workflows Without Code

Drag, Drop, and Connect Your Business Processes



- ✓ No Coding Required – Visual Designer
- ✓ 400+ Built-in Connectors
- ✓ Ideal for Business Analysts & Integration

Choosing Your Compute Service - Quick Reference

One glance table ranks each service on control, ease, scale and cost so you can defend your choice in the meeting. VMs win on control, Functions win on cost, App Service sits in the sweet middle. Keep the card in your notebook and the debate ends quickly.

Criteria	VMs	App Svc	ACI	AKS	Functions
Control	★★★★★	★☆	★★	★★★★☆	★
Ease of Use	★	★★★★☆	★★★★★	★☆	★★★★☆
Scalability	★★☆	★★★★★	★☆	★★★★★	★★★★★
Cost Effective	★☆	★★★	★★★★☆	★★☆	★★★★★
Startup Speed	★	★☆	★★★★★	★★☆	★★★★☆

Real-World Scenarios - Matching Services

Find the Right Compute Solution for Your Needs

🛒 Scenario 1: E-Commerce Website 🛒

Requirements:
- Web application (ASP.NET)
- Traffic varies (100-5k users)
- Need auto-scaling
- Regular deployments

✅ BEST CHOICE: App Service

Why: Perfect for web apps
- Auto-scaling built-in
- CI/CD integration
- Managed platform

🖼️ Scenario 2: Image Processing 🖼️

Requirements:
- Process 10,000 images nightly
- Variable upload volume
- Run during off-hours

✅ BEST CHOICE: Azure Functions

Why: Event-driven (blob trigger)
- Pay only when processing
- Auto-scales to volume
- No idle costs

🏢 Scenario 3: Legacy ERP System 🏢

Requirements:
- 15-year-old app
- Windows Server 2012
- Custom COM components
- Cannot modify code

✅ BEST CHOICE: Virtual Machines

Why: Full OS control
- Replicate exact environment
- Lift-and-shift migration

🧱 Scenario 4: Microservices Platform 🏗️

Requirements:
- 50+ microservices
- Independent scaling
- Service mesh needed
- Container-based

✅ BEST CHOICE: AKS (Kubernetes)

Why: Designed for microservices
- Advanced orchestration
- Production-grade
- Industry standard

📈 Scenario 5: Daily Report Generation 📊

Requirements:
- Run every day at 6 AM
- Query database
- Generate Excel file
- Takes 5 minutes

✅ BEST CHOICE: Azure Functions

Why: Timer trigger (scheduled)
- Short duration
- Cost-effective
- Easy to maintain

📦 Scenario 6: Order Processing Workflow 📦

Requirements:
- Integrate Shopify, Stripe
- FedEx
- Approval workflow
- Business logic changes often

✅ BEST CHOICE: Logic Apps

Why: Visual workflow designer
- 400+ connectors
- No-code solution
- Easy for business users

🔬 Scenario 7: Scientific Simulation 🔬

Requirements:
- Run 5,000 simulations
- Needs HPC capabilities
- Parallel processing essential

✅ BEST CHOICE: Azure Batch

Why: Designed for HPC
- Massive parallel processing
- Auto-scales to 1000s of nodes
- Job scheduling

📱 Scenario 8: Mobile App Backend 📱

Requirements:
- RESTful APIs
- Push notifications
- User authentication
- Data sync

✅ BEST CHOICE: App Service (Mobile Apps)

Why: Mobile-specific features
- Built-in authentication
- Offline data sync
- Easy scaling

Azure Compute Best Practices: Five Habits for a Friendly Cloud Bill & Quiet Pager

Right-Size & Scale: Start small, optimize resource allocation, and configure auto-scaling for efficiency.

Managed Identity: Eliminate passwords; use Azure AD Managed Identities for secure, passwordless access.

Wire to Azure Monitor: Integrate every service with Azure Monitor for comprehensive visibility, diagnostics, and alerting.

Set Budgets & Alerts: Proactively manage costs with budget caps and spending alerts to avoid billing surprises.

Patch & Update on Schedule: Maintain security and performance with automated, scheduled updates and patches.

Follow these five habits and your cloud bill stays friendly and your pager stays quiet.

Common Anti-Patterns (What NOT to Do)

Avoid These Common Mistakes!

❌ Anti-Pattern 1: Using VMs for Everything

- Problem: "We know VMs, so everything goes on VMs"
- Impact:
  - Higher costs
  - Slower deployments
  - More maintenance
  - Less efficient scaling

✅ **Better:** Evaluate PaaS options first

❌ Anti-Pattern 2: No Auto-Scaling

- Problem: "We'll manually add capacity when needed"
- Impact:
  - Over-provisioned (wasting money) Or under-provisioned (poor performance)
  - Manual intervention required

✅ **Better:** Configure auto-scaling from day 1

❌ Anti-Pattern 3: Running Everything in Production

- Problem: "One environment for everything"
- Impact:
  - Can't test safely
  - Deployments risky
  - No staging environment

✅ **Better:** Separate Dev, Test, Production

❌ Anti-Pattern 4: Ignoring Monitoring

- Problem: "We'll add monitoring later"
- Impact:
  - Problems discovered by users first
  - No data for troubleshooting
  - Blind to performance issues

✅ **Better:** Enable monitoring from start

❌ Anti-Pattern 5: Over-Engineering

- Problem: "Let's use Kubernetes for this 2-container app"
- Impact:
  - Unnecessary complexity
  - Higher learning curve
  - More operational overhead

✅ **Better:** Start simple, scale complexity when needed

❌ Anti-Pattern 6: No Cost Monitoring

- Problem: "We'll worry about costs later"
- Impact:
  - Surprise bills
  - Wasted resources
  - No accountability

✅ **Better:** Set budgets and alerts immediately

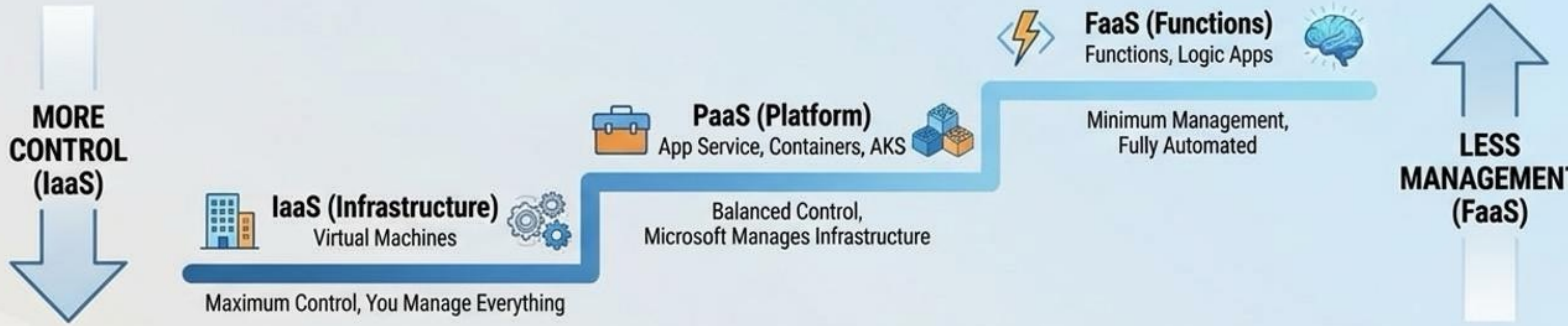
❌ Anti-Pattern 7: Shared Production Accounts

- Problem: "Everyone uses the admin password"
- Impact:
  - Security risk
  - No audit trail
  - Can't track who did what

✅ **Better:** Azure AD + RBAC for everyone

Balancing Control & Management: The Azure Compute Spectrum

More control means more chores; pick the highest step you can comfortably reach.


MORE CONTROL (IaaS)
IaaS (Infrastructure)
Virtual Machines
Maximum Control, You Manage Everything
PaaS (Platform)
App Service, Containers, AKS
Balanced Control, Microsoft Manages Infrastructure
FaaS (Functions)
Functions, Logic Apps
Minimum Management, Fully Automated
LESS MANAGEMENT (FaaS)

Governance, cost and **security** are not after-thoughts—they are the first three clicks in the portal.

Governance

Cost

Security

Start simple, add complexity only when the workload proves it needs it, and let the icons remind the team what lives where.