#Milano

Cinquanta sfumature di Azure Compute!

Massimo Bonanni

Microsoft Technical Trainer @ Microsoft

#GLOBALAZURE
BY COMMUNITY • FOR COMMUNITY
2026
#Milano

improove

TD SYNNEX

Grazie ai nostri sponsor 🙏

Your mission today…

is to learn how to choose the right service for the job.

But there's a problem.

Azure doesn't give you one way to do it...

It gives you many.

Too many services.

Too many choices.

Choose the wrong one…

and nothing explodes.

…but your architecture might.

Your complexity will.

And your bill definitely will.

Your mission today…

is to learn how to choose the right service for the job.

This session will not self-destruct…

but your next deployment might,

if you choose poorly.

The A-Team


AGENT VM
CODENAME: THE VETERAN
AGENT APP SERVICE
CODENAME: THE PROFESSIONAL
AGENT AZURE FUNCTIONS
CODENAME: THE GHOST
AGENT CONTAINER INSTANCES
CODENAME: THE HITMAN
AGENT CONTAINER APPS
CODENAME: THE COORDINATOR
AGENT AKS
CODENAME: THE COMMANDER

AGENT VM

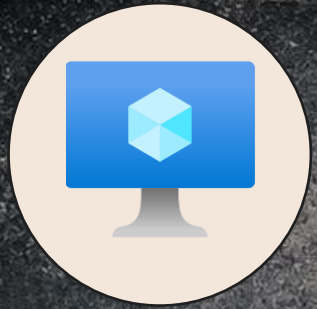
CODENAME: THE VETERAN

IaaS

A virtual server in Azure with strong control over the environment.

Ops effort: **higher**

Azure manages: **the infrastructure, not your guest OS**



IaaS

A virtual server in Azure with strong control over the environment.

Ops effort: **higher**

Azure manages: **the infrastructure, not your guest OS**

Full machine control

Choose the OS, install software, configure the runtime, and access the VM through SSH or RDP.

Flexible sizing

Pick the VM size, region, network, disks, and related resources that match the workload.

Good for migration

A strong fit when you want to move an existing server or app with minimal redesign.

You still operate it

You remain responsible for configuration, patching, and software maintenance inside the VM.

Virtual Machines - common scenarios

Lift-and-shift servers

Move existing Windows or Linux workloads to Azure when redesign is not the first step.

Dev and test labs

Spin up machines with specific configurations for demos, labs, training, and temporary environments.

Custom or legacy apps

Run software that needs admin rights, special OS settings, custom agents, or unusual dependencies.

Choose it when: you need strong OS-level control, custom software setup, or a like-for-like migration path.

AGENT APP SERVICE

CODENAME: THE PROFESSIONAL

PaaS

Think “managed web hosting for developers.”

Ops effort: **low to medium**

Azure manages: **the web hosting platform and much of the runtime plumbing**

Managed web platform

Run web applications, REST APIs, and mobile back ends without managing the underlying infrastructure.

Many runtimes supported

Supports .NET, Java, Node.js, Python, PHP, and also custom containers on Windows or Linux.

CI/CD friendly

Works well with GitHub Actions, Azure Pipelines, and staging environments for safer releases.

Built-in platform extras

Includes useful capabilities such as authentication, custom domains, certificates, and VNet integration.

App Service - common scenarios

Company website or portal

A great fit for public websites, internal portals, and business web apps with predictable hosting needs.

REST APIs

Host APIs quickly with built-in HTTPS, deployment tooling, and integration with other Azure services.

Line-of-business web apps

Useful when a team wants managed hosting plus enterprise features like auth, domains, and networking.

Choose it when: your workload is primarily a web app or API, and you want a developer-friendly managed platform.

AGENT AZURE FUNCTIONS

CODENAME: THE GHOST

PaaS (FaaS)

You focus on the function.
Azure handles the hosting.

Ops effort: **low for simple tasks**

Azure manages: **the serverless runtime, scaling, and platform services**

Event-driven by design

Functions reacts to triggers such as HTTP calls, timers, storage events, queues, and message systems.

Triggers and bindings

Built-in integrations connect your function to other services without lots of plumbing code.

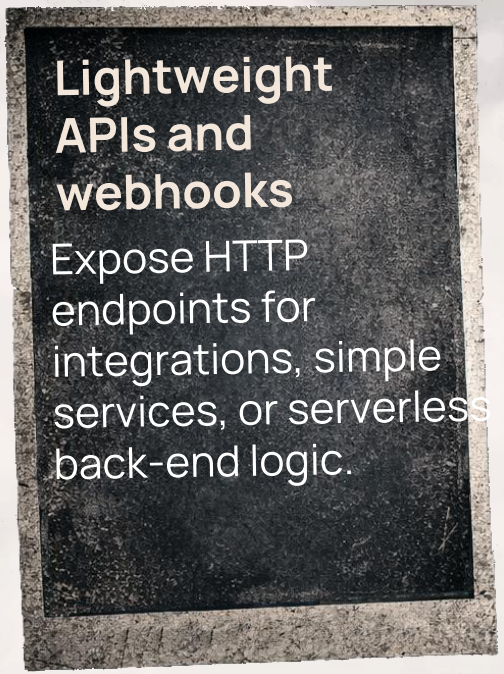
Great for workflows and APIs

You can build lightweight APIs, background tasks, and longer workflows with Durable Functions.

Developer friendly

Develop locally, deploy through CI/CD, and monitor with Azure Monitor and Application Insights.

Azure Functions - common scenarios



Choose it when: the workload is event-driven, bursty, or naturally broken into small units of code.

AGENT CONTAINER INSTANCES

CODENAME: **THE HITMAN**

PaaS (CaaS)

A quick way to run one container or one small container group on demand.

Ops effort: **low**

Azure manages: **the container host and infrastructure runtime**

PaaS (CaaS)

A quick way to run one container or one small container group on demand.

Ops effort: **low**

Azure manages: **the container host and infrastructure runtime**

Fast to start

ACI is built for getting containers running quickly, often in seconds, with minimal setup.

No orchestration required

You do not run Kubernetes or manage virtual machines just to launch a container workload.

Container groups

You can run multi-container groups for tightly coupled pieces that should start together.

Azure integrations

Connect to virtual networks and use Azure Container Registry, managed identity, and file shares.

ACI - common scenarios

Batch and build jobs

Run short-lived jobs such as data processing, build tasks, or one-off scripts in containers.

Simple containerized apps

Deploy a basic tool, demo, utility, or proof of concept when orchestration would be overkill.

Burst or temporary capacity

Use isolated containers on demand when traffic or work arrives in spikes and does not justify a full cluster.

Choose it when: you need containers fast, but you do not need Kubernetes-style orchestration.

AGENT CONTAINER APPS

CODENAME: THE COORDINATOR

PaaS (CaaS)

More app platform, less infrastructure.

Ops effort: **medium**

Azure manages: **the container platform, scaling, ingress, and much of the service plumbing**

PaaS (CaaS)

More app platform, less infrastructure.

Ops effort: **medium**

Azure manages: **the container platform, scaling, ingress, and much of the service plumbing**

Serverless containers

Run containerized applications without managing server configuration or a Kubernetes control plane.

Autoscaling

Scale based on HTTP traffic, events, CPU, memory, and other KEDA-supported triggers. Many apps can scale to zero.

Built for microservices

Supports ingress, internal service discovery, DAPR integration, and jobs for background work.

Safe releases

Use revisions and traffic splitting for blue/green deployments, testing, and gradual rollouts.

Container Apps - common scenarios

Microservices

Run several small services that need autoscaling, internal communication, and modern deployment patterns.

Containerized APIs

Host web APIs in containers with managed ingress and scale rules, without building a cluster yourself.

Background workers and jobs

Handle event-driven processing, scheduled jobs, and asynchronous workers in a container-first model.

Choose it when: you want a modern container platform with autoscaling and microservice features, but not full Kubernetes administration.

AGENT AKS

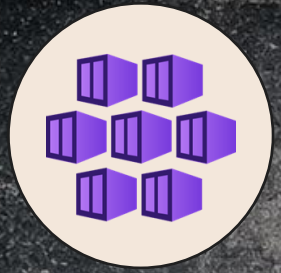
CODENAME: THE COMMANDER

Managed Kubernetes

The most capable container option - and usually the most operationally demanding.

Ops effort: **higher**

Azure manages: **the Kubernetes control plane and many Azure integrations**



Managed Kubernetes

The most capable container option - and usually the most operationally demanding.

Ops effort: **higher**

Azure manages: **the Kubernetes control plane and many Azure integrations**

Managed Kubernetes

Azure operates the control plane so teams can focus on running workloads on Kubernetes.

Scale and structure

Use node pools and autoscaling features to support different workload shapes and growth patterns.

Platform integrations

Connect AKS with identity, policy, monitoring, networking, storage, and developer tooling.

Advanced orchestration

A strong fit for teams that need Kubernetes patterns such as self-healing, rolling deployments, and fine control.

AKS - common scenarios

Large microservice platforms

Run many containerized services with policy, scaling, secrets, observability, and controlled rollout patterns.

Platform engineering standardization

Use Kubernetes when teams want a common orchestration layer, GitOps/Helm practices, or portable platform skills.

Specialized or demanding workloads

Choose AKS when applications need advanced networking, storage, mixed node pools, or more fine-grained control.

Choose it when: you specifically need Kubernetes features, and your team is ready for the extra platform complexity.

Start with the simplest service that matches the workload.

Then move toward more control or more orchestration only when the application truly needs it.

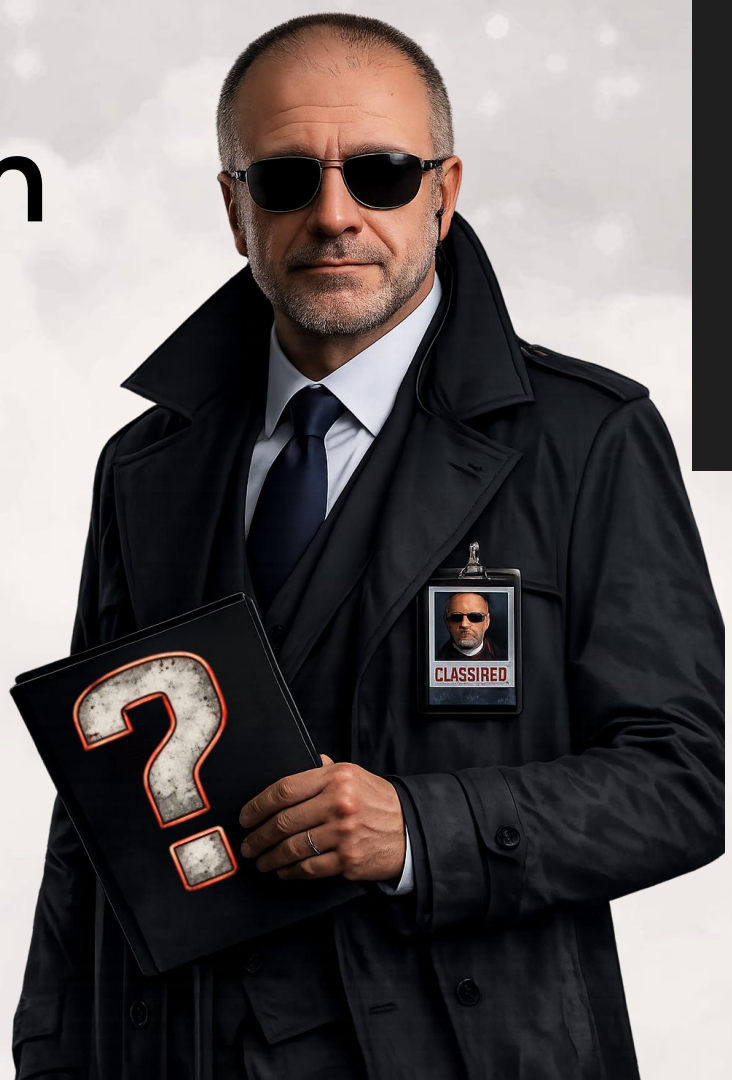
I love it when a plan comes together!

Massimo Bonanni

Sr Technical Trainer @ Microsoft

massimo.bonanni@microsoft.com
github.com/massimobonanni

aka.ms/maxlinkedin



#GLOBALAZURE 2026

#Milano

Slide e video:
https://www.globalazuremilano.it