

CSCI 5902 - Fall 23 - Azure Tutorial

Designed under guidance of Dr. Lu Yang

Harmit Narula
©2023, Faculty of Computer Science

Recap

- Serverless Architecture
- Serverless Compute - Azure Container Apps, Serverless K8s, Functions, App Environments.
- Functions - file uploads, Real time stream & event processing
- Durable Functions vs Logic Apps
- Serverless DBs - SQL, CosmosDB
- Reference Architecture for E-commerce platform

T10: Decoupled, Microservices, HA-DR Architecture in Azure

**Is Microservices same as decoupled
architecture?**

Decoupled vs Microservices

- With decoupled architecture we aim to enhance flexibility, scalability and maintainability by reducing inter-dependencies between components of software system.
- A Decoupled Architecture acts as a catalyst to speed up existing Application re-designing and migration as well as new Software Application creation in the public cloud.
- Key Characteristics:
 - Modularity
 - Loose Coupling
 - High Cohesion
 - Scalability
 - Technology Agnostic

Decoupled vs Microservices(Contd.)

- Ways to implement decoupled Architecture:
 - Microservices Architecture
 - Event Driven Architecture
 - Service Oriented Architecture
 - Plugin Based Architecture
- Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are:
 - Independently deployable
 - Losely Coupled
 - Organized around business capabilities
 - Owned by small team.[1]

2000's

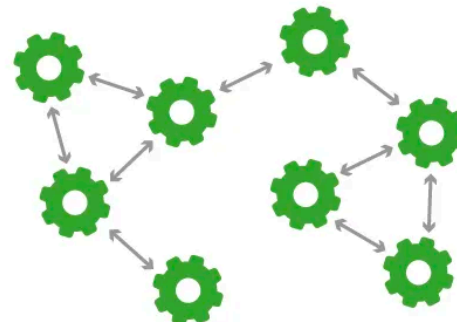
SERVICE ORIENTED ARCHITECTURE



SOA based applications are comprised of more loosely coupled components that use an Enterprise Services Bus messaging protocol to communicate between themselves.

2010's

MICROSERVICES ARCHITECTURE



Microservices are a number of independent application services delivering one single functionality in a loosely connected and self-contained fashion, communicating through light-weight messaging protocols such as HTTP, REST or Thrift API.

Imagine you have a business critical monolith App, you are struggling to meet business needs. Should you migrate to microservices?

It depends

Make the most of your monolith - adopt devops, reorganize into loosely coupled, small teams

Optional Reading for microservices migration:

<https://microservices.io/post/architecture/2022/11/29/yow-london.html>

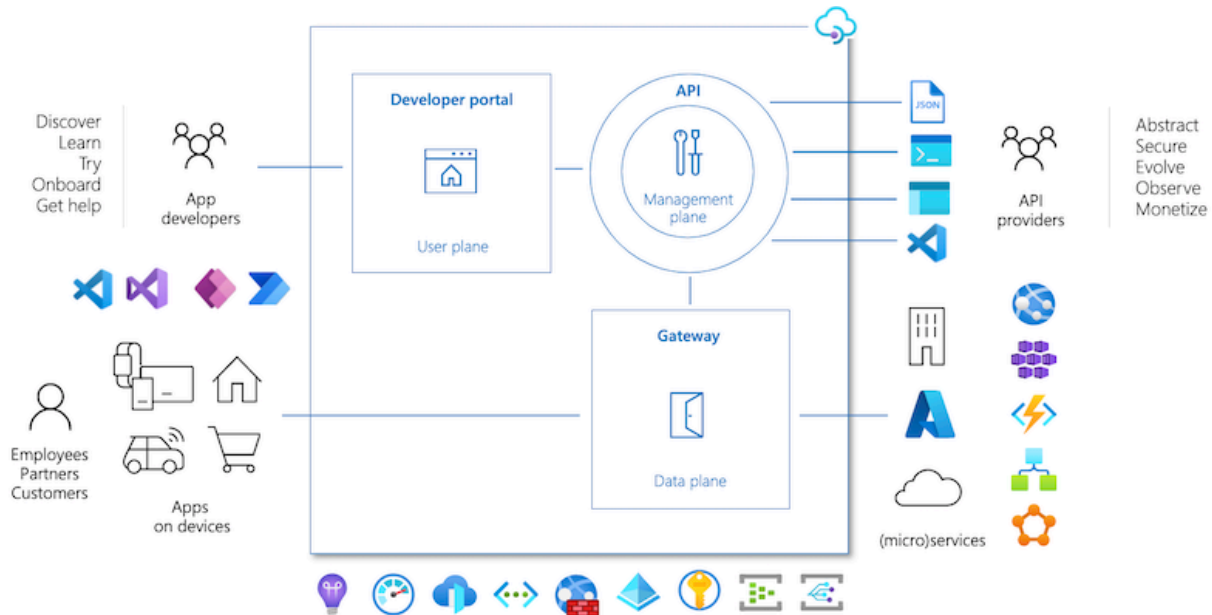
Microservices Architecture in Azure

API Gateway

- In a microservices Architecture, API gateway sits between clients and services.
- Provides reverse proxy, authentication, SSL termination, rate limiting.
- Provides layer 7 routing and aggregation services.
- Example tasks which can be offloaded to API gateway:
 - SSL Termination
 - Authentication
 - IP Allowlist/Blocklist
 - Rate limiting/throttling
 - Logging & Monitoring
 - Response caching
 - WAF
 - Servicing static content

Azure API Management(APIM)

- Azure API Management is made up of an *API gateway*, a *management plane*, and a *developer portal*. These components are Azure-hosted and fully managed by default. API Management is available in various tiers differing in capacity and features.



APIM(Contd.)

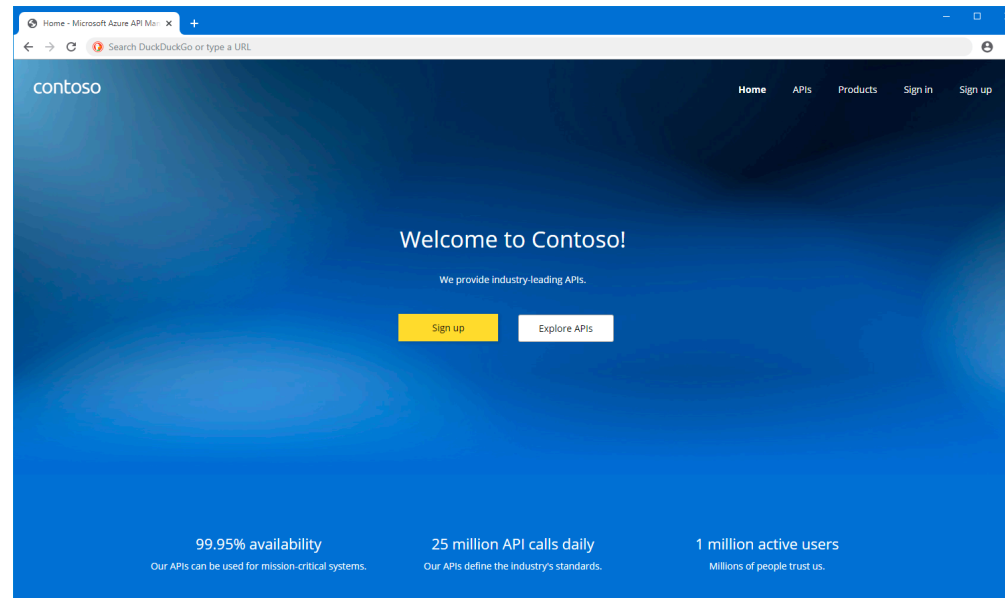
- **API Gateway:** All requests from client applications first reach the API gateway, which then forwards them to respective backend services.
- You can also configure APIM with self-hosted gateway. The self-hosted gateway enables customers with hybrid IT infrastructure to manage APIs hosted on-premises and across clouds from a single API Management service in Azure.
- **Management Plane:** API providers interact with the service through the management plane, which provides full access to the API Management service capabilities.

APIM(Contd.)

- A management plane can be used to:
 - A Provision and configure API Management service settings
 - Define or import API schemas from a wide range of sources, including OpenAPI specifications, Azure compute services, or WebSocket or GraphQL backends
 - Package APIs into products
 - Set up policies like quotas or transformations on the APIs
 - Get insights from analytics
 - Manage users[3]
- **Developer Portal:** The open-source developer portal is an automatically generated, fully customizable website with the documentation of your APIs.

APIM(Contd.)

- Using the developer portal, developers can:
 - Read API documentation
 - Call an API via the interactive console
 - Create an account and subscribe to get API keys
- Access analytics on their own usage
- Download API definitions
- Manage API keys



Source: [3]

Default Developer Portal

APIM Tier Comparison

- APIM is offered in 5 tiers - Consumption, Developer, Basic, Standard, Premium

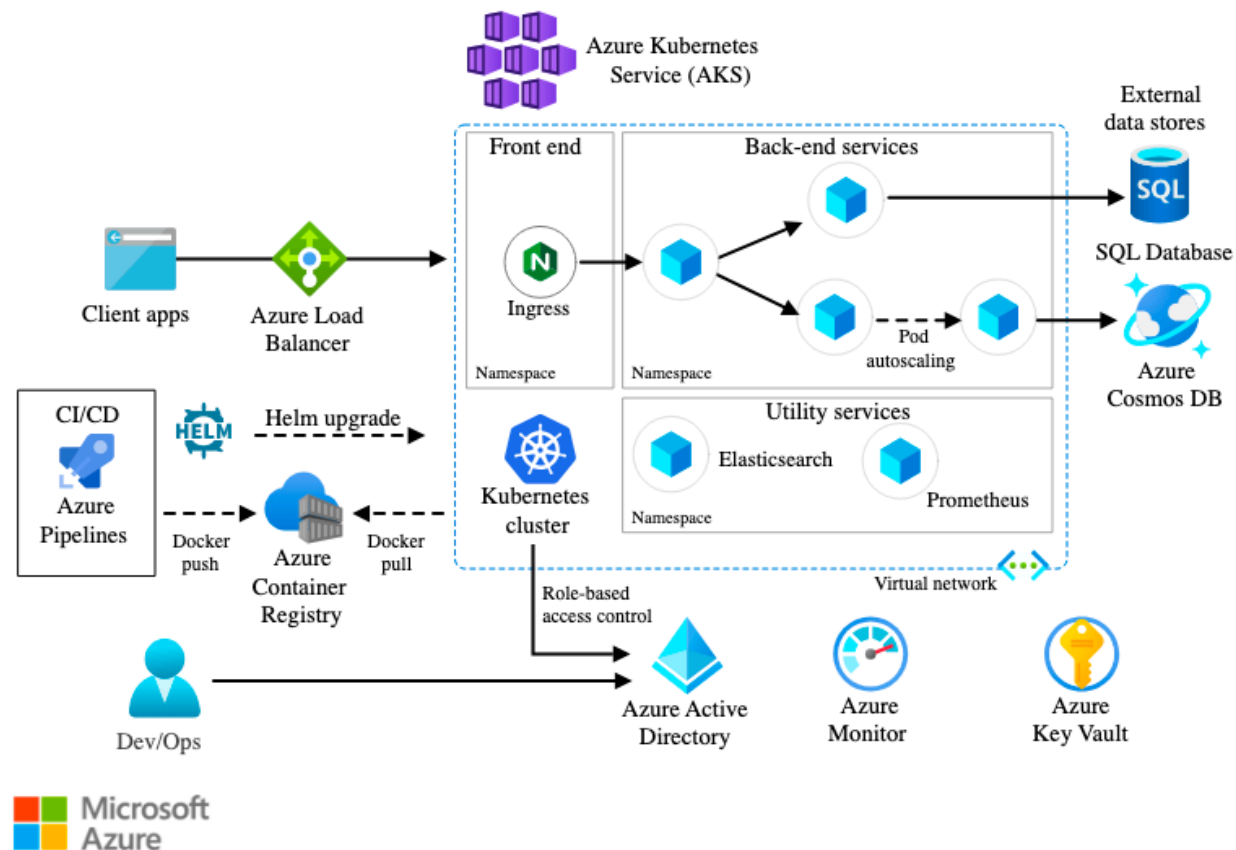
Self Reading:

<https://learn.microsoft.com/en-us/azure/api-management/api-management-features>

Challenges of Microservices Architecture

- Complexity
- Development and Testing
- Lack of governance
- Network Congestion and latency
- Data integrity
- Versioning
- Skill set(team)

Microservices Reference Architecture on AKS



HA-DR in Azure

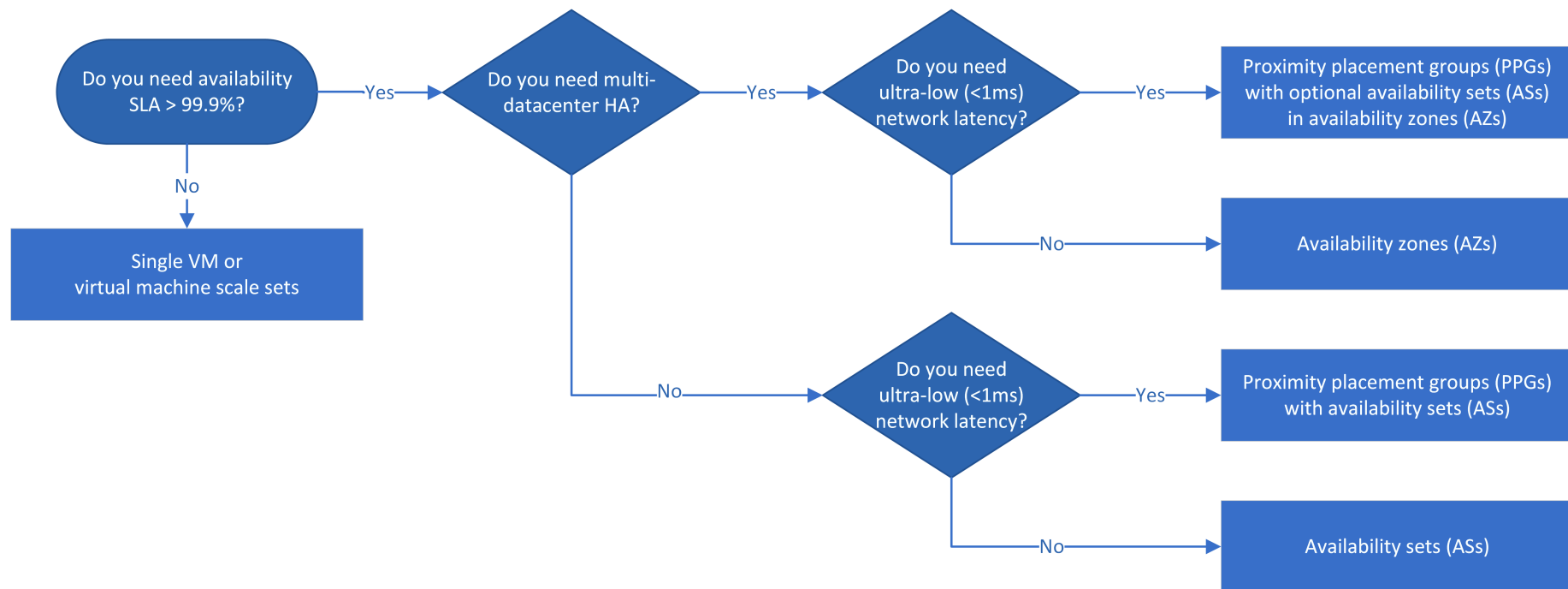
High Availability(HA)

- HA refers to the ability of a system or application to remain operational and accessible, even in the face of failures.
- How do we achieve high availability?
 - Eliminating single point of failure.
 - Redundancy
 - Rapid failover to backup systems
- Availability Zones, Load Balancers, VM Scale Sets, Availability Sets.

Best Practices for HA

- Design for fault tolerance
- Automated monitoring and alerting
- Test failover and disaster recovery process
- Use Availability Sets.
- Leverage Managed Services

HA Decision Tree for IaaS systems



Disaster Recovery(DR)

- DR is always considered along with availability. The ability to keep system running in healthy state and data durability.
- RPO vs RTO
- While designing HA and DR systems consider following factors:
 - Assess Criticality
 - Geographic dependencies
 - Cost Considerations
 - SLA requirements
 - Regular tests to check effectiveness

Azure Site Recovery(ASR)

- ASR provides Site Recovery and Backup Services.
- Site Recovery: helps ensure business continuity by keeping business apps and workloads running during outages. Site Recovery replicates workloads running on physical and virtual machines (VMs) from a primary site to a secondary location.
- Backup Service: Keeps data safe and recoverable.
- Can help in replicating VMs between regions, On-premise, across CSPs
- Provides continuous replication for Azure VMs.

**Can I use ASR to protect an app
which requires persistent MAC
address on VM?**

No

It's a wrap



References

- [1] <https://microservices.io/>
- [2] <https://medium.com/microtica/microservices-vs-soa-is-there-any-difference-at-all-2a1e3b66e1be>
- [3] <https://learn.microsoft.com/en-us/azure/api-management/api-management-key-concepts>
- [4] <https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-microservices/aks-microservices>
- [5] <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/infrastructure/iaas-high-availability-disaster-recovery>