

Welcome to Week 1

Virtual Mentored Academy

Azure Fundamentals



PLURALSIGHT

Hello

HELLO
my name is

Allen Sanders
Senior Technology Instructor
Pluralsight ELS

About me...



- 27+ years in the industry
- 23+ years in teaching
- Certified Cloud architect
- Passionate about learning
- Also, passionate about Reese's Cups!



Agenda

- Speaking the language of Cloud
- Compute, Networking, & Storage
- Security
- Monitoring & Managing



How we're going to work together

- Slides and words to highlight key concepts
- Demos to bring those concepts “to life”
- Lab work (which will take place in sandboxes provided by “A Cloud Guru”) for hands-on reinforcement
- NOTE: I welcome being interrupted – if you need more info, or clarification, or anything else, just break in and ask. I am here to help you.

A decorative graphic consisting of a thick orange line that runs horizontally across the top and then turns 90 degrees downward on the right side. A thick pink line runs horizontally below the orange line, starting from the left edge and ending where the orange line turns. The background is black with a grid of small white dots on the right side.

Speaking the Language of Cloud



Application Hosting

By Application Hosting, we mean the target infrastructure and runtime platform used for deployment and execution of an application or system; can include compute (CPU and server resources), storage, network, data and operating system



Application Hosting – An “Interesting” Example?

Here's an example of someone thinking “outside-of-the-box” when it comes to application hosting!

<https://mashable.com/article/pregnancy-test-doom/>

What Are the Hosting Options with Cloud?

- ☐ IaaS
- ☐ PaaS
- ☐ Serverless / FaaS
- ☐ SaaS
- ☐ Containers



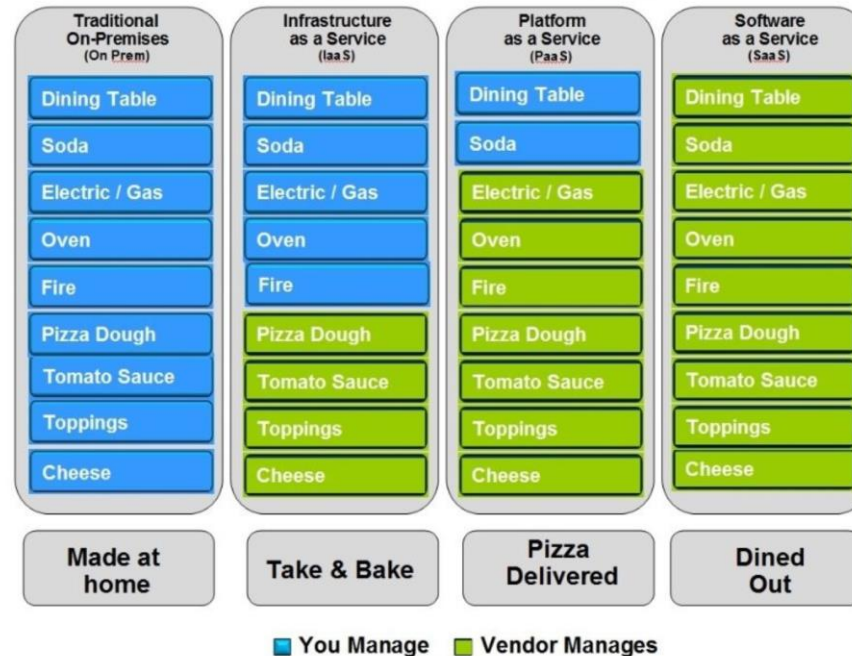
What do they all mean?

Pizza-as-a-Service

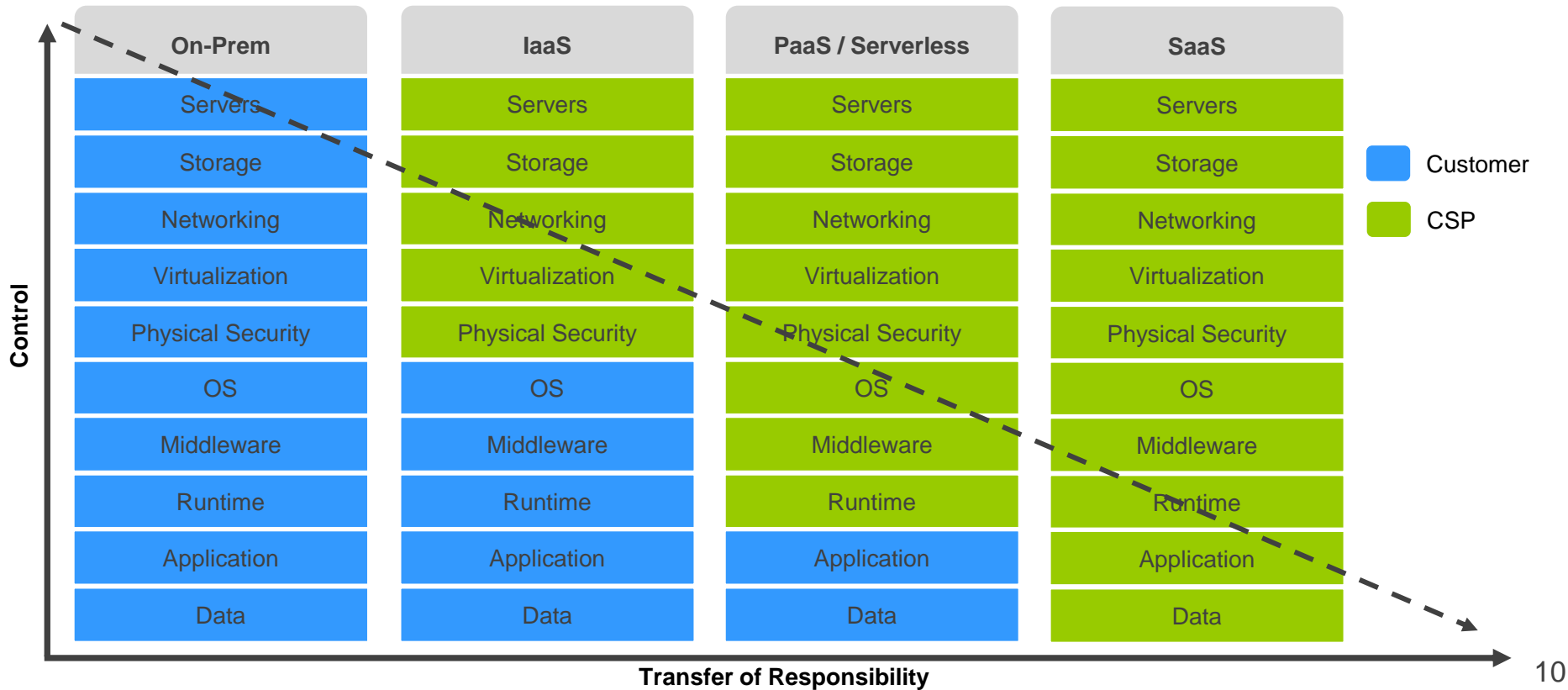
From a LinkedIn post by Albert Barron from IBM (<https://www.linkedin.com/pulse/20140730172610-9679881-pizza-as-a-service/>)



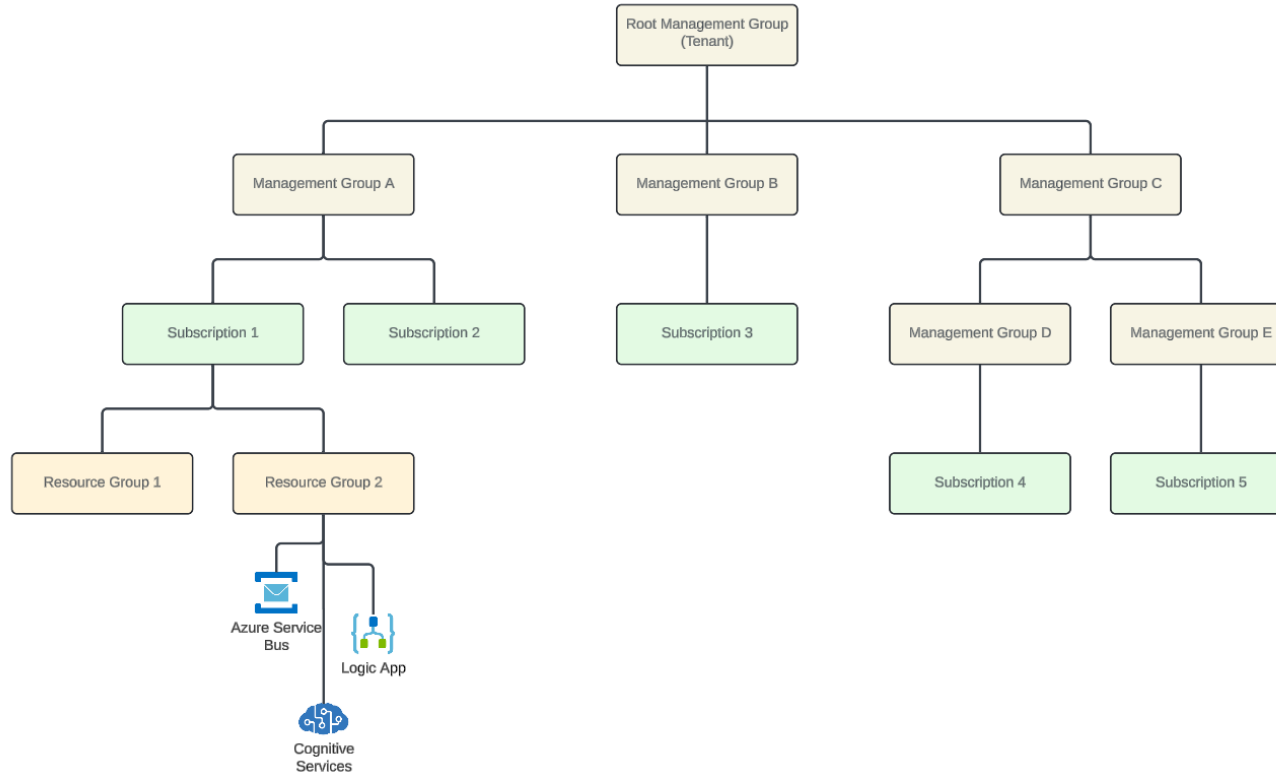
Pizza as a Service



Side-by-Side Comparison



Azure Resource Architecture



Each “level” (Management Group, Subscription, and Resource Group) provides a boundary for enforcing security and policy



Compute, Networking, & Storage

A thick yellow L-shaped line is positioned on the left side of the slide, starting from the top and extending downwards. The background of the slide is a solid pink color with a subtle grid of small white dots.

Compute

Compute (IaaS)



All services | Compute

Filter services Service providers : All Release Status : All

Infrastructure as a Service (IaaS)

Availability sets	Community images
Compute Fleet	Azure compute galleries
Disk Pools PREVIEW	Host groups
Image templates	Images
Lab accounts	Proximity placement groups
Restore Point Collections PREVIEW	SSH keys
Azure Virtual Desktop	Virtual machine scale sets
Virtual machines ★	VM application definitions
VM application versions	VM image definitions
VM image versions	

Compute (PaaS)



All services | Compute

All
Favorites
Recents
Recommended for you
Categories
AI + machine learning
Analytics
Compute

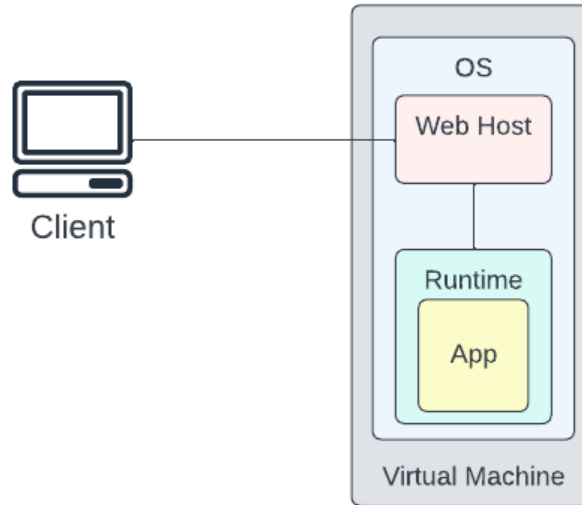
VM image versions

Platform as a Service (PaaS)

- App Services ★
- Cloud services (extended support)
- Azure Spring Apps
- Virtual Instances for SAP solutions
- Azure VMware Solution

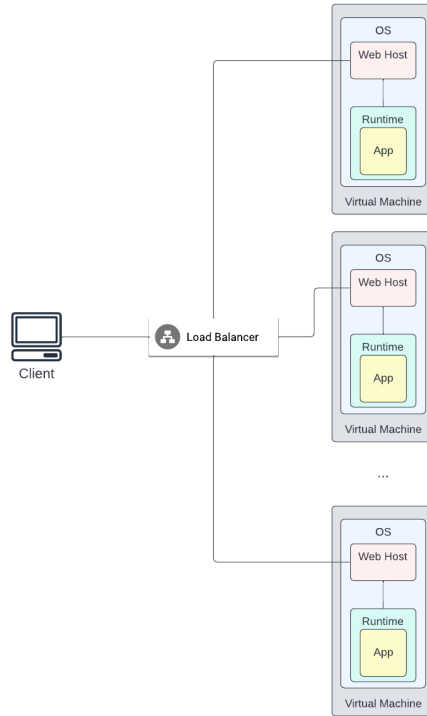
The Value Proposition

Deploying Web Apps – the “Hard(er)” Way



- Responsible for management of OS
- Responsible for installation & configuration of a web hosting framework
- Responsible for installation & configuration of a runtime
- Responsible for installation & configuration of the app itself

What If We Need More Instances?



- Now our responsibilities (and associated effort) gets multiplied by N
- Also, need new components (like a load balancer)
- Need custom domain management
- Likely need an SSL/TLS framework to support encryption
- Increases complexity

Maybe there's a better way?

Azure App Service + Plan



Azure App Service

- Provides an HTTP-based managed service for hosting web apps and APIs
- Supports multiple development languages
- Relieves operational burden when hosting & exposing a web-enabled workload
- Configuration managed via App Service Plan



Azure App Service Plan

- App Service always runs in App Service Plan
- Defines the compute resources used to host and service your web app or API
- Dictates level of isolation for your workload
- Also, used to expose key capabilities like scaling, deployment, encryption, etc.



Azure App Service Plan – Pricing Tiers

Shared Compute

Dedicated Compute

Isolated

LAB:

Azure App Service

Execute the “Hands-On” lab available at
https://github.com/KernelGamut32/azure_docker_microservices-public/tree/main/week01/labs/lab01

Compute (Serverless)



All services | Compute

All

Favorites

Recents

Recommended for you


Categories


AI + machine learning

Analytics


Compute

Serverless and microservices


 Container Apps



 Container Apps Environments



 Function App



 Kubernetes services

High performance computing

Azure Function Apps



Azure Functions

- One of Microsoft's serverless offerings on Azure
- As with other serverless offerings, enables focus on application and data (not infrastructure)
- Can implement logic (in multiple languages) in code blocks called “functions”
- Supports multiple types of “triggers”
- Scales automatically to address increases in demand

Azure Functions

If you want to...	then...
Process file uploads	Run code when a file is uploaded or changed in blob storage.
Process data in real time	Capture and transform data from event and IoT source streams on the way to storage.
Infer on data models	Pull text from a queue and present it to various AI services for analysis and classification.
Run scheduled task	Execute data clean-up code on pre-defined timed intervals.
Build a scalable web API	Implement a set of REST endpoints for your web applications using HTTP triggers.
Build a serverless workflow	Create an event-driven workflow from a series of functions using Durable Functions.
Respond to database changes	Run custom logic when a document is created or updated in Azure Cosmos DB .
Create reliable message systems	Process message queues using Queue Storage, Service Bus, or Event Hubs.



Durable Functions

- Extension of Azure Functions
- Supports stateful functions that can be integrated into orchestrated workflows or entity management algorithms
- As with Azure Functions, supports multiple programming languages
- Also supports multiple patterns
- See <https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview> for more info



Hosting Plans

- Azure Functions offers multiple types of hosting plans
- Hosting plan drives other features of function app (like scalability, available timeouts, service limits, etc.)
- See <https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale> for more info

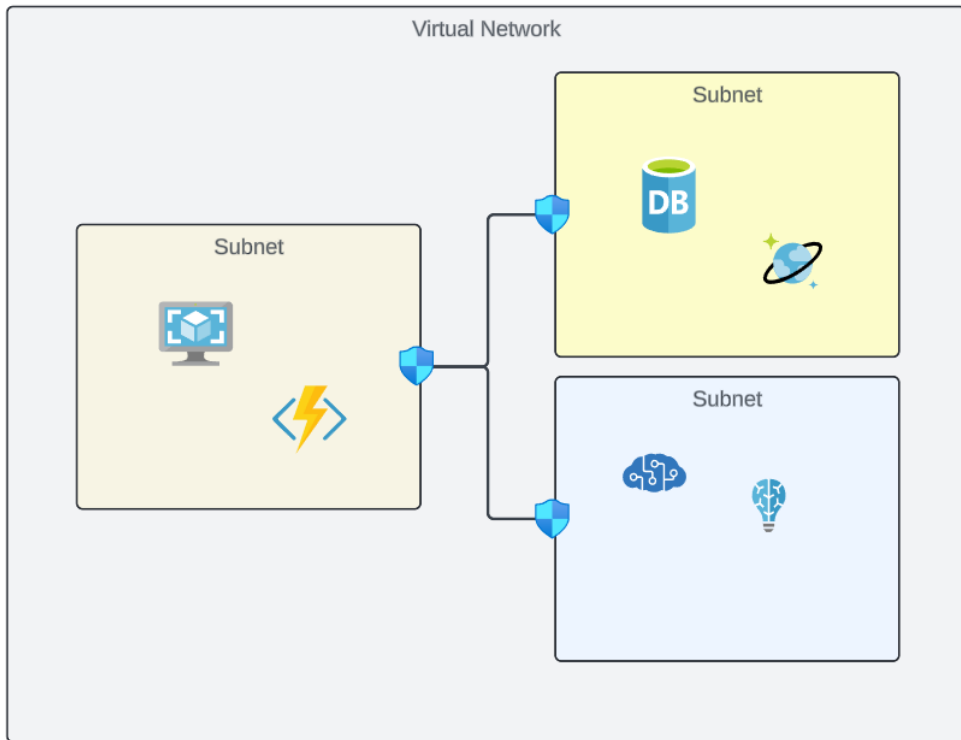
LAB:

Azure Function App

Execute the “Hands-On” lab available at
https://github.com/KernelGamut32/azure_docker_microservices-public/tree/main/week01/labs/lab02

Networking

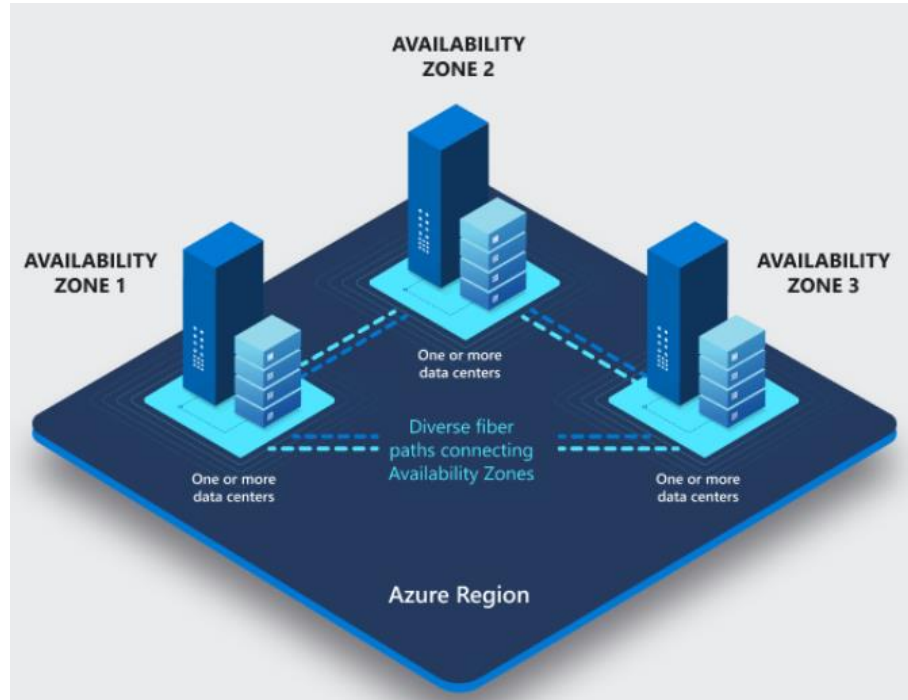
Virtual Networks in Azure



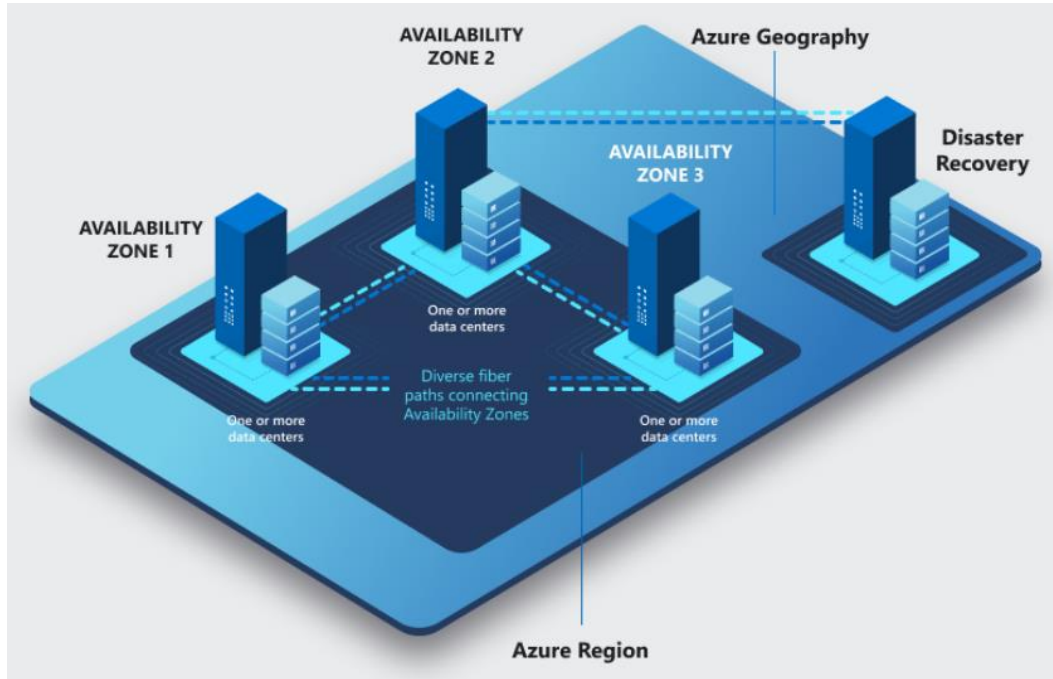
- Provides isolated area of network space in the Cloud for resource deployment
- Can be further subdivided into subnets
- Resources can be deployed into subnets and standard rules in Azure dictate how resources across the network space can integrate
- Additionally, you can define a set of rules (in a Network Security Group) to further control what a resource can connect to or what can be connected to it

Regions and Availability Zones

- Azure provides regions across the globe for deploying resources closer (i.e., lower latency) to consumers of those resources
- Availability Zones within a region help protect against datacenter failure



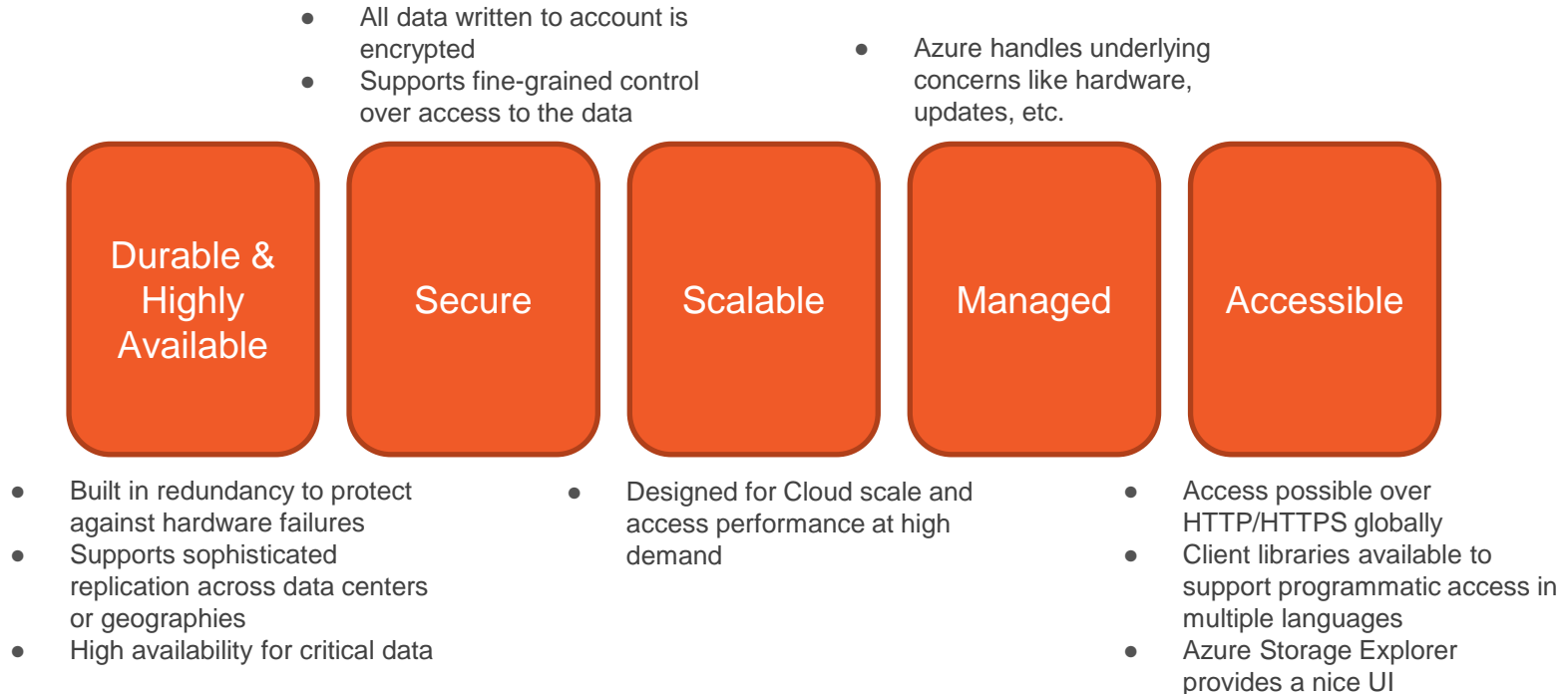
Geographic Redundancy and BC/DR



- Supports BC/DR (Business Continuity/Disaster Recovery)
- To provide redundancy, replicas of resources can be deployed to multiple regions
- Regions are paired – maintenance on pairs is coordinated to help minimize downtime due to resource failure

Azure Storage

Azure Storage – Benefits Provided





First Step: Creating an Azure Storage Account

- Multiple methods for creation supported, including Azure PowerShell, Azure Resource Manager (ARM) templates – a form of Infrastructure-as-Code, Azure CLI, and the Azure Portal
- Requires some configuration details to define where the Storage Account will “live” in Azure – the subscription, resource group, and region
- Also, needs a name – something globally unique

First Step: Creating an Azure Storage Account

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *

Resource group *
[Create new](#)

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name ⓘ *

Region ⓘ *

Performance ⓘ *

☒ **Standard:** Recommended for most scenarios (general-purpose v2 account)

☐ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ *

☒ Make read access to data available in the event of regional unavailability.

- Performance options include “Standard” and “Premium”
- Standard will be utilized for most workloads
- Premium is reserved for creation of disks used by virtual machines

First Step: Creating an Azure Storage Account

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *

Resource group * [Create new](#)

Instance details

If you need to create a legacy storage account, select a legacy storage account type.

Storage account name ⓘ *

Region ⓘ *

Performance ⓘ *

Redundancy ⓘ *

Locally-redundant storage (LRS):
Lowest-cost option with basic protection against server rack and drive failures. Recommended for non-critical scenarios.

Geo-redundant storage (GRS):
Intermediate option with failover capabilities in a secondary region. Recommended for backup scenarios.

Zone-redundant storage (ZRS):
Intermediate option with protection against datacenter-level failures. Recommended for high availability scenarios.

Geo-zone-redundant storage (GZRS):
Optimal data protection solution that includes the offerings of both GRS and ZRS. Recommended for critical data scenarios.

☒ Make read access to data available in the event of regional unavailability.

- Multiple redundancy options available
- Protects against data loss with different types of infrastructure failure
- More sophisticated levels of redundancy carry a higher cost

Redundancy Options

Locally Redundant Storage (LRS)

- Replicates Storage Account 3 times within a single data center in primary region
- Offers least durability – does not protect against data center failure
- Lowest-cost option
- Write requests considered complete only after data successfully written to all 3 replicas

Zone Redundant Storage (ZRS)

- Replicates Storage Account across 3 availability zones in primary region
- Provides improved durability with protection against a single data center failure
- Write requests considered complete only after data successfully written to all 3 replicas across the availability zones
- Good option for Azure Files workloads

Geo-Redundant Storage (GRS)

- Copies data synchronously 3 times within a single physical location in the primary region (using LRS)
- Then copies data asynchronously to single physical location in a secondary region with substantial distance from the primary region
- Provides improved durability with protection against failure in a region

Geo-Zone-Redundant Storage (GZRS)

- Copies data across 3 availability zones in the primary region
- Also, replicates data using LRS to a data center in a secondary geographic region
- Can provide protection against both data center and region failure
- Most durable but also most expensive

Azure Storage – Types

Blob

Disk

File

Archive



Azure Storage – Blob

- Supports storage and access to unstructured data
- Can be used to host static resources for serving a simple website
- Can be used to store files, documents, bits of JSON, task definitions, etc.



Azure Storage – Blob

- Can be used in support of audio or video streaming
- Can be used to store backups for Business Continuity/Disaster Recovery (BC/DR)
- Blob data lives in a container you create in the Storage Account represented as a flat-file hierarchy



Azure Storage – Disk

- Provides virtualized backing storage for a virtual machine
- Like a physical disk you'd find in a physical piece of infrastructure
- But implemented as a separate, virtualized storage facility that enables reassociation to different VMs (if required)
- Supports multiple options depending on performance requirements (premium solid-state, standard solid-state, and standard hard disk drives)



Azure Storage – File

- Provides fully managed file shares in the Cloud for use in multiple other service offerings
- Can be mounted in Cloud workloads as well as on-premise workloads
- Offers very robust option for replacing on-premise file servers with a fully managed Cloud service that supports multiple file protocols and access to data across the globe



Azure Storage – Archive

- Tiering is available for Blob containers in an Azure Storage Account
- Offers different combinations of data retrieval performance and cost
- Includes Hot, Cool, and Archive

Azure Storage – Archive

Hot	Cool	Archive
<ul style="list-style-type: none">• Provides fastest retrieval• Highest storage cost• Lowest access cost• No minimum period for data storage	<ul style="list-style-type: none">• Longer retrieval times• Medium storage costs• Medium access costs• 30-day minimum period for data storage	<ul style="list-style-type: none">• Longest retrieval times (on the order of hours)• Lowest storage cost• Highest access cost• 180-day minimum period for data storage



Securing Access to Azure Storage Account

- Multiple levels of security – security at the account level, at the container level, and at the individual file level for Blobs
- Uses Microsoft Entra ID Role-Based Access Control (RBAC) to secure at the account level
- Uses access keys to secure at the account level or for individual containers
- Uses something called Shared Access Signature (SAS) to secure at the individual Blob level



Storage Account Access Keys

- Storage Accounts have a primary and secondary access key
- Both a key and a related connection string are available for enabling scripted or programmatic access to the account
- Should be protected and regularly rotated as a best practice for securing data in the account



Shared Access Signature (SAS)

- SAS token can be generated by an authorized user or application for distribution to an end user
- Represents a URI-friendly hashed string derived from the Storage Account access key
- Provides more granular levels of access control on the account



Shared Access Signature (SAS)

- Includes start and end times for limiting access to a specific time window (safer)
- Allows specification of the types of access to allow (read, write, create, list, delete)
- Can be generated for the entire Storage account, for a specific container, or individual files



Stored Access Policy

- Provides a common reference for creating multiple SAS tokens
- The policy defines permissions, time frame, etc. – SAS tokens generated from the policy use the common configuration
- Management of the policy can be used to manage all tokens generated from it
- Only applies at the container level, not individual files

LAB:

Azure Storage

Execute the “Hands-On” lab available at
https://github.com/KernelGamut32/azure_docker_microservices-public/tree/main/week01/labs/lab03

SQL Server in Azure

SQL Server in Azure - Options

SQL Server on Azure VM

- Uses a virtual machine created in Azure (IaaS)
- Customer downloads, installs, and manages SQL Server software & instance on the VM

Azure SQL

- Fully managed service
- Does not provide access to all features available with SQL Server installed on a VM

Azure SQL Managed Instance

- Fully managed service
- Has more compatibility with on-premise instances and IaaS instances installed on a VM
- Better supports common tools used with SQL Server-based workloads

Azure Cosmos DB



Azure Cosmos DB

- Provides a fully managed NoSQL database platform
- NoSQL is the model used when the data to be stored does not fit a strict schema (or structure)
- Built around the concept of storing data as “documents” – hierarchical sets of key/value data that describe an entity
- JSON (or JavaScript Object Notation) is a common document format used for storage

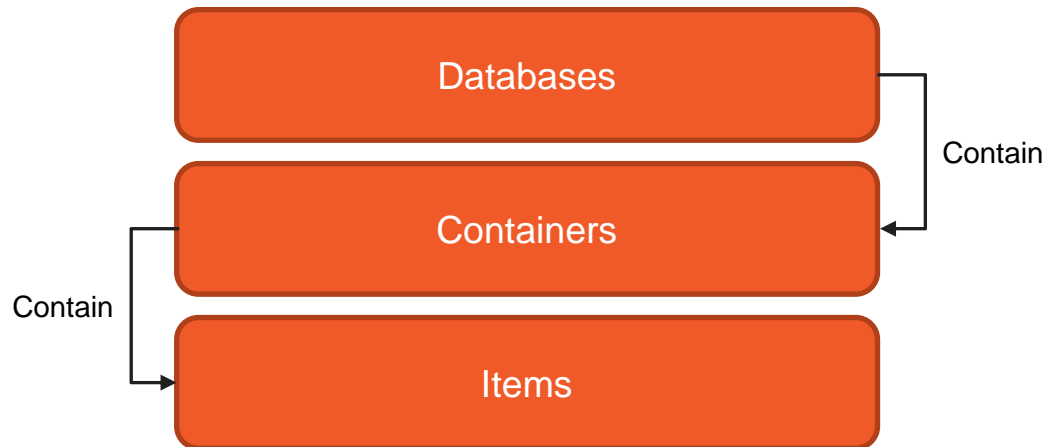


Azure Cosmos DB

- Intended for all types of application use cases
- Provides very fast data access, elastic scalability, and global availability
- Built into the service offering is support for deployment to multiple regions across the world, enabling low latency access for users to locations closest in proximity

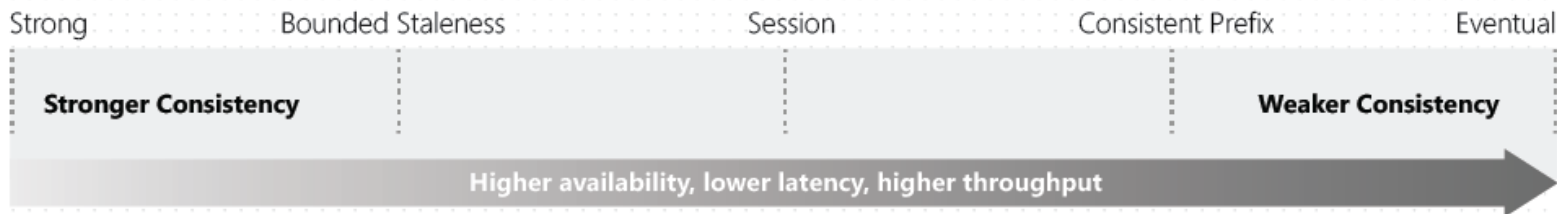
Azure Cosmos DB – Resource Model

- Within an Azure subscription, a new Cosmos DB account can be created
- Within the Cosmos DB account, data is stored and organized using a defined hierarchy



Consistency Levels

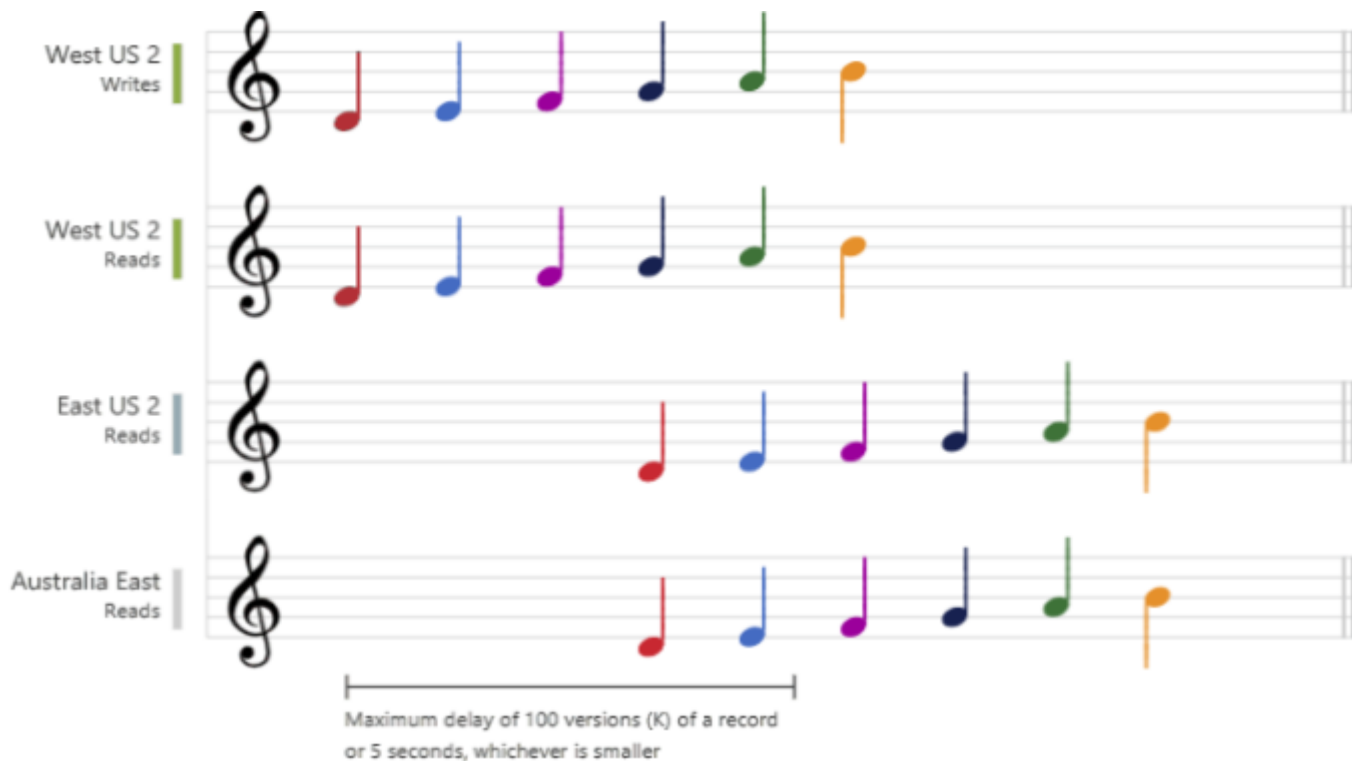
- With multiple replicas of your data distributed globally, consistency defines how in sync the replicas remain and how quickly out-of-sync issues are corrected
- There are 5 levels provided by Cosmos DB
- Each level provides tradeoffs in terms of data sync and performance



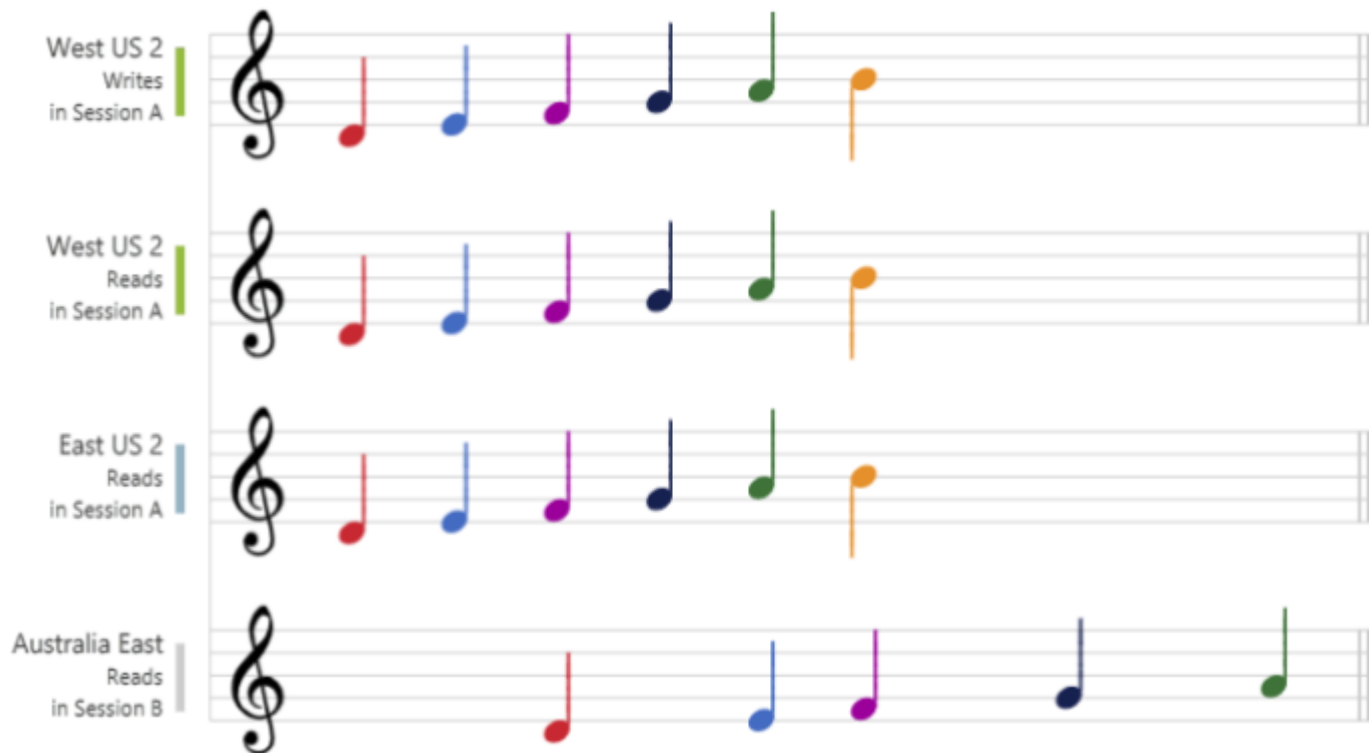
Strong Consistency



Bounded Staleness



Session



Consistent Prefix



Eventual



Azure Cosmos DB – APIs Supported

Core/SQL API

API for MongoDB

Cassandra API

Gremlin API

Table API



Azure Database for PostgreSQL

- PostgreSQL is an open-source database product
- Uses tables, columns, rows, and relationships to create a hierarchy of application entities (relational) but also supports non-relational (NoSQL) data
- As with the other types, supports network connectivity for connecting and operating on data remotely
- Azure's service provides a fully managed instance of the DB platform

LAB:

Azure Cosmos DB

Execute the “Hands-On” lab available at
https://github.com/KernelGamut32/azure_docker_microservices-public/tree/main/week01/labs/lab04



Security

Microsoft Entra ID



What is it?

- Formerly Azure Active Directory
- Provides cloud-based IAM (Identity & Access Management)
- Enables integrated access to several Microsoft platforms (including Azure, Microsoft 365, etc.) as well as linking with multiple SaaS applications
- Organized by tenant – represents an organization and helps with managing internal & external user access to resources
- Supports several license types defining levels of access to more advanced capabilities



What is it?

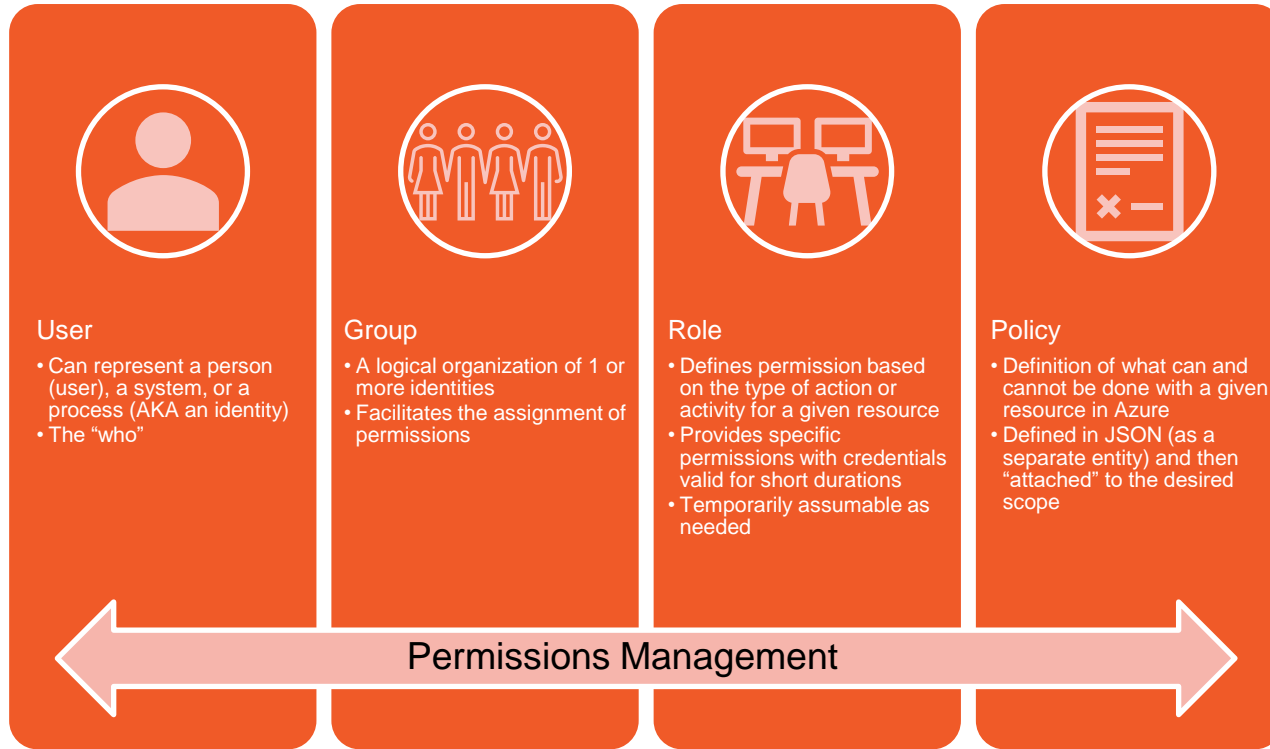
- Built around industry standards (like OAuth 2.0, OpenID Connect, and SAML v2.0)
- Supports multiple identity types – work or school accounts, personal Microsoft accounts, and social accounts (Facebook, Google, Twitter, etc.)
- Provides open-source libraries to facilitate programmatic integration
- Portal for managing application registrations (more to come on that)
- Provides APIs to support automatic registration of applications as part of CI/CD pipelines



What is it?

- License types supported include Free, Office 365, Premium P1, and Premium P2
- As expected, the Premium license versions offer more IAM capability but also cost more
- Also provides B2B (Business-to-Business) and B2C (Business-to-Customer) offerings to support access for internal/external users and SSO (Single Sign-On)

IAM – What is it?



Azure Key Vault

Azure Key Vault

- Provides secure storage, management, and access for keys used to encrypt (and decrypt) data
- Key unlocks sensitive information so protection is critical

Secrets Management

Key Management

Certificate Management

- Securely store and tightly control tokens, passwords, API keys, etc.
- Can also be used to store/manage sensitive application config detail (like Connection Strings)
- Supports provision, management, and deployment for public/private certs used to secure workloads



Azure Key Vault

- Comes in two “flavors”
 - Standard – encrypts with a software key
 - Premium – secures data using Hardware Security Module (HSM)-protected keys; more secure but also costlier
- Supports API integration for pulling a secret into a runtime workflow
- Also, multiple other Azure services have direct integration with Key Vault built in so that you can easily apply as part of service execution/operation
- Also, can be easily integrated with CI/CD pipelines in Jenkins, GitHub Actions, and Azure DevOps



Azure Key Vault

- Some of the high-level benefits include:
 - Centralization of sensitive detail needed to configure and operate your workloads
 - Granular security, allowing admin to control access through Microsoft Entra ID and Role-Based Access Control (RBAC)
 - Provides robust monitoring capabilities so an organization can see and know who is accessing and how
 - As previously mentioned, integrates seamlessly with other Azure services (e.g., encryption for Azure Disks, sophisticated encryption support in Azure SQL, and secure configuration for Azure App Service)

LAB:

Azure Key Vault

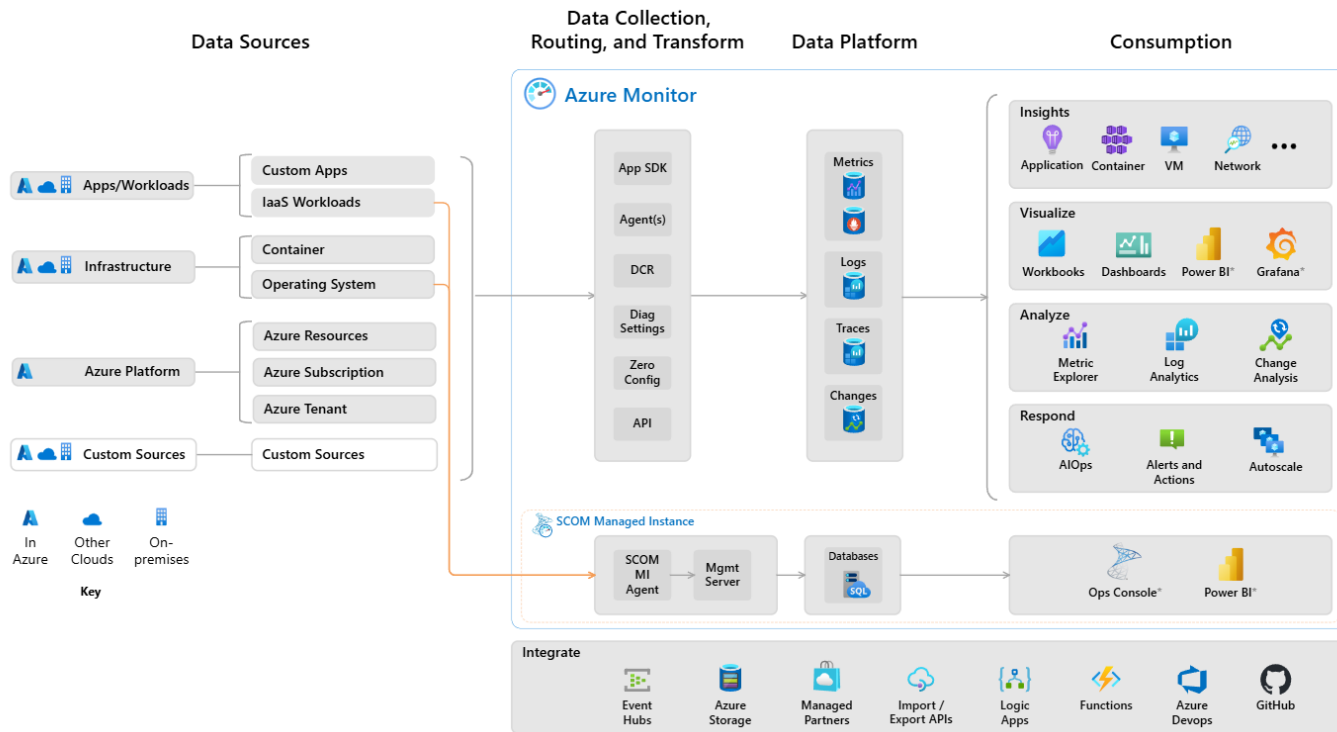
Execute the “Hands-On” lab available at
https://github.com/KernelGamut32/azure_docker_microservices-public/tree/main/week01/labs/lab05



Monitoring & Managing

Azure Monitor

Azure Monitor

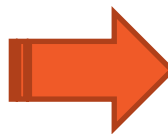


Infrastructure-as-Code (IaC)



IaC – What is it?

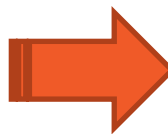
- As the name implies, the definition & configuration of our infrastructure IN code
- Instead of manually creating (inefficient) → automated in scripts that run “at the push of a button”





IaC – Why is it valuable?

- If only creating a handful of resources, manual is (probably) fine
- Creating hundreds (or even thousands), not so much!
- Modern DevOps is built around automation – quickly tearing down and rebuilding entire sets of infrastructure as and when required



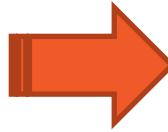
laC – Advantages?



Testable

Repeatable

Auditable



ARM (Azure Resource Manager)

Azure Resource Manager (ARM) Templates

Microsoft's JSON-based IaC solution

Supports definition of Cloud resources for Azure in code that adheres to a predefined schema

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "languageVersion": "",
  "contentVersion": "",
  "apiProfile": "",
  "definitions": { },
  "parameters": { },
  "variables": { },
  "functions": [ ],
  "resources": [ ], /* or "resources": { } with languageVersion 2.0 */
  "outputs": { }
}
```

Going to briefly look at details for just a few of the available elements

Source: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/syntax>



\$schema Element

Location of schema file that defines target version of template language
Used to enforce rules around properties, hierarchy, and values applied to each
Can be used to automate validation of a given ARM template instance

parameters Element

Optionally allows specification of values that can be inputs at deployment time
Limited to 256 parameters in a given template

```
"definitions": {  
  "<definition-name>": {  
    "type": "<data-type-of-definition>",  
    "allowedValues": [ "<array-of-allowed-values>" ],  
    "minValue": <minimum-value-for-int>,  
    "maxValue": <maximum-value-for-int>,  
    "minLength": <minimum-length-for-string-or-array>,  
    "maxLength": <maximum-length-for-string-or-array>,  
    "prefixItems": <schema-for-validating-array>,  
    "items": <schema-for-validating-array-or-boolean>,  
    "properties": <schema-for-validating-object>,  
    "additionalProperties": <schema-for-validating-object-or-boolean>,  
    "discriminator": <schema-to-apply>,  
    "nullable": <boolean>,  
    "metadata": {  
      "description": "<description-of-the-type-definition>"  
    }  
  }  
}
```

Source: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/syntax>

variables Element

Optionally allows definition of variables that can be used throughout template – similar to variables used in application code

Can help reduce complex expressions through reusability

```
"variables": {  
  "<variable-name>": "<variable-value>",  
  "<variable-name>": {  
    <variable-complex-type-value>  
  },  
  "<variable-object-name>": {  
    "copy": [  
      {  
        "name": "<name-of-array-property>",  
        "count": <number-of-iterations>,  
        "input": <object-or-value-to-repeat>  
      }  
    ],  
    "copy": [  
      {  
        "name": "<variable-array-name>",  
        "count": <number-of-iterations>,  
        "input": <object-or-value-to-repeat>  
      }  
    ]  
  }  
}
```

functions Element

Optionally allows definition of user-defined, custom functions

Allows encapsulation of complex expressions/instructions that are callable by name

```
"functions": [  
  {  
    "namespace": "<namespace-for-functions>",  
    "members": {  
      "<function-name>": {  
        "parameters": [  
          {  
            "name": "<parameter-name>",  
            "type": "<type-of-parameter-value>"  
          }  
        ],  
        "output": {  
          "type": "<type-of-output-value>",  
          "value": "<function-return-value>"  
        }  
      }  
    }  
  }  
],
```

Source: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/syntax>



resources Element

Required section of template that enables the definition of the actual resources to be deployed
See source URL for additional info

outputs Element

Defines values returned from a deployment (i.e., outputs from execution of the deployment)
Can be used to return details from deployed resources (e.g., public DNS value)

```
"outputs": {  
  "<output-name>": {  
    "condition": "<boolean-value-whether-to-output-value>",  
    "type": "<type-of-output-value>",  
    "value": "<output-value-expression>",  
    "copy": {  
      "count": <number-of-iterations>,  
      "input": <values-for-the-variable>  
    }  
  }  
}
```

Source: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/syntax>

LAB:

ARM Template

Execute the “Hands-On” lab available at
[https://github.com/KernelGamut32/azure_docker_microservices-
public/tree/main/week01/labs/lab06](https://github.com/KernelGamut32/azure_docker_microservices-public/tree/main/week01/labs/lab06)



Thank you!

If you have additional questions,
please reach out to me at:
asanders@gamuttechnologysvcs.com



PLURALSIGHT