Certainly! Azure is a cloud computing platform provided by Microsoft, offering a wide range of services that enable organizations to build, deploy, and manage applications and services through Microsoft's global network of data centers. Here are some basics of Azure that may be relevant for a Cloud DevOps interview:

1. **Azure Services:**

- **Compute Services**: Azure provides various options for computing resources, including Virtual Machines (VMs), Azure Kubernetes Service (AKS) for container orchestration, and Azure Functions for serverless computing.

-  **Storage Services**: Azure offers scalable and secure storage solutions such as Azure Blob Storage, Azure Table Storage, Azure Files, and Azure Queues.

-  **Networking Services**: Azure provides networking services like Virtual Network (VNet), Azure Load Balancer, Azure Application Gateway, and Azure VPN Gateway to enable connectivity and communication between different resources.

-  **Database Services**: Azure offers fully managed database services like Azure SQL Database, Azure Cosmos DB, Azure Database for MySQL, and more.

2.  **Azure Resource Manager (ARM):**

- ARM is the deployment and management service in Azure that allows you to define and manage the resources in your Azure solution. You use ARM templates to deploy and manage resources consistently.

3.  **Azure DevOps:**

- Azure DevOps is a set of development tools and services that facilitate the entire DevOps lifecycle. It includes services for version control (Azure Repos), continuous integration (Azure Pipelines), package management (Azure Artifacts), release management, and more.

4.  **Infrastructure as Code (IaC):**

- In Azure, Infrastructure as Code is often implemented using ARM templates or tools like Terraform. This allows you to define and manage infrastructure configurations in a code format, enabling automation and reproducibility.

**5. Azure DevOps Pipelines:**

- Azure Pipelines is a CI/CD (Continuous Integration/Continuous Deployment) service that automates the building, testing, and deployment of applications. It supports integration with various languages, platforms, and deployment targets.

**6. Azure Monitoring and Logging:**

- Azure Monitor provides comprehensive monitoring solutions, including metrics, logs, and application insights. Azure Log Analytics can be used for advanced querying and analysis of log data.

**7. Security and Compliance:**

- Azure provides a range of security features, including Azure Active Directory for identity and access management, Azure Key Vault for managing secrets and keys, and Azure Security Center for threat detection and management.

**8. Azure DevOps Repositories:**

- Azure Repos provides Git repositories or Team Foundation Version Control (TFVC) for version control. It supports collaborative development and integrates seamlessly with Azure Pipelines.

**9. Azure CLI and Azure PowerShell:**

- These are command-line tools that allow you to manage Azure resources using scripts and commands. Azure CLI is cross-platform and works on Windows, macOS, and Linux, while Azure PowerShell is specifically for Windows users.

**10. Azure Service Level Agreements (SLAs):**

- Understanding SLAs is crucial. SLAs define the guaranteed uptime and performance of Azure services, and it's essential to consider them when designing and deploying applications.

These are just a few fundamental aspects of Azure that might be relevant for a Cloud DevOps interview. Depending on the specific role and responsibilities, you may need to delve deeper into specific areas such as Azure DevOps Pipelines, security practices, and monitoring strategies. Always be prepared to discuss your hands-on experience with these tools and services.

Cloud Computing:

Cloud computing is a technology model that provides on-demand access to computing resources over the internet. Instead of owning and maintaining physical hardware, users can access computing power, storage, and other services as needed.

Key Characteristics:

1. On-Demand Self-Service: Users can provision and manage resources as needed without human intervention from the service provider.

2. Broad Network Access: Services are accessible over the network and can be accessed through standard mechanisms (e.g., web browsers, mobile devices).

3. Resource Pooling: Computing resources are pooled to serve multiple users, with different physical and virtual resources dynamically assigned and reassigned according to demand.

4. Rapid Elasticity: Resources can be quickly and easily scaled up or down to accommodate changing workloads.

5. Measured Service: Cloud resources are metered, and users pay for only what they consume. This pay-as-you-go model is often more cost-effective.

**Cloud Service Models:**

Infrastructure as a Service (IaaS):

- Provides virtualized computing resources over the internet. Users can deploy and run virtual machines, storage, and networking.

Platform as a Service (PaaS):

- Offers a platform allowing customers to develop, run, and manage applications without dealing with the complexities of infrastructure. It includes development tools, databases, and middleware.

Software as a Service (SaaS):

- Delivers software applications over the internet. Users access the application through a web browser without worrying about installation or maintenance.

**Cloud Deployment Models:**

1. Public Cloud:

- Resources are owned and operated by a third-party cloud service provider and made available to the general public. Azure is an example of a public cloud.

2. Private Cloud:

- Resources are used exclusively by one business or organization. They can be hosted on-premises or by a third-party provider.

3. Hybrid Cloud:

- Combines public and private cloud resources, allowing data and applications to be shared between them. This provides greater flexibility and more deployment options.

**Benefits of Cloud Computing:**

Cloud computing offers a multitude of benefits, transforming the way organizations manage, store, and process data. Here's a detailed explanation of the key advantages of cloud computing:

**Cost Savings:**

- Reduced Capital Expenditure: Cloud computing eliminates the need for organizations to invest heavily in physical hardware, data centers, and other infrastructure. Instead, they can use and pay for resources on a pay-as-you-go basis.

- Operational Efficiency:Cloud services automate many tasks, reducing the need for manual intervention, and optimizing resource utilization, leading to operational cost savings.

**2. Scalability:**

Scalability is a crucial concept in cloud computing, including Azure. It refers to the ability of a system or application to handle an increasing amount of work, or its potential to be enlarged to accommodate growth. Scalability can be achieved through various means, and it is important to understand related terms such as elasticity, autoscaling, and global scalability.

Scalability in Azure refers to the capability of Azure services and applications to handle growing workloads by efficiently adding or removing resources as needed.

Vertical Scaling: Increasing the capacity of existing resources, such as adding more CPU, memory, or storage to a virtual machine.

Horizontal Scaling: Adding more instances of resources, such as deploying additional virtual machines, to distribute the load.

**Elasticity**  is a subset of scalability and emphasizes the ability to automatically and dynamically provision and de-provision resources based on demand. It is about adapting to workload changes in real-time.

Automatic Resource Adjustment: Resources scale in and out based on predefined rules or dynamically changing conditions.

Cost Efficiency: Ensures that resources are only utilized when needed, leading to cost savings during periods of low demand.

**Autoscaling**  is a specific implementation of elasticity where the cloud environment automatically adjusts the number of resources allocated to an application based on predefined criteria.

Rule-Based Scaling: Policies and rules determine when to scale resources up or down.

Managed by Cloud Provider: The cloud platform (such as Azure) handles the process of adding or removing resources.

Global Scalability involves deploying applications or services across multiple geographic regions to provide improved performance, resilience, and redundancy.

Data Replication: Data and services are replicated across multiple regions to ensure availability and disaster recovery.

Low-Latency Access: Users can access resources from a nearby data center, reducing latency and improving user experience.

Summary of Differences:

1. Scalability:

- Refers to the ability to handle increased workloads by adding or removing resources.

- Can be achieved through vertical or horizontal scaling.

2. Elasticity:

- Focuses on the ability to dynamically adapt to workload changes by automatically provisioning or de-provisioning resources.

- Emphasizes real-time adjustments based on demand.

3. Autoscaling:

- A specific implementation of elasticity where the cloud platform automatically adjusts resources based on predefined rules or conditions.

- Managed by the cloud provider and often involves rule-based scaling.

4. Global Scalability:

- Involves deploying applications or services across multiple geographic regions.

- Enhances performance, resilience, and redundancy by replicating data and services globally.

In summary, scalability is a general concept related to handling workload changes, while elasticity and autoscaling specifically address the dynamic adjustment of resources. Global scalability extends this concept to a global scale, providing benefits such as low-latency access and improved reliability through geographic redundancy. In the context of Azure, these principles are fundamental for designing resilient and efficient cloud-based solutions.

**3. Flexibility and Agility:**

- Resource Variety: Cloud services offer a wide range of computing resources, storage options, and development tools. Organizations can choose the services that best fit their needs.

- Rapid Deployment: Cloud computing enables rapid provisioning and deployment of resources, reducing the time it takes to launch new applications or services.

**4. Reliability and Availability:**

- Data Redundancy: Cloud providers replicate data across multiple locations, ensuring data durability and availability even in the event of hardware failures.

- Service Level Agreements (SLAs): Cloud providers offer SLAs that guarantee a certain level of uptime and performance, providing assurance to organizations relying on their services.

**5. Security:**

- Specialized Security Teams: Cloud providers have dedicated security teams that implement and maintain robust security measures, often exceeding what individual organizations could achieve on their own.

- Compliance and Certifications: Cloud services adhere to industry-specific compliance standards and certifications, ensuring data protection and regulatory compliance.

**6. Collaboration Efficiency:**

- Remote Collaboration: Cloud computing facilitates remote collaboration by providing centralized access to data and applications. Teams can work together in real-time from different locations.

- Version Control: Cloud-based collaboration tools often include version control, ensuring that team members are working with the latest version of documents and code.

**7. Automatic Updates and Maintenance:**

- Vendor Responsibility: Cloud providers handle routine maintenance tasks, including software updates and security patches. This allows organizations to focus on their core business activities rather than infrastructure management.

**8. Disaster Recovery:**

- Data Backup and Recovery: Cloud providers offer automated backup and recovery solutions, reducing the risk of data loss due to unforeseen events.

- Geographic Redundancy: Multi-region data storage and disaster recovery options enhance data resilience.

**9. Environmental Impact:**

- Resource Optimization: Cloud providers optimize the use of resources, leading to better energy efficiency and reduced environmental impact compared to traditional on-premises infrastructure.

**10. Innovation Opportunities:**

- Access to Emerging Technologies: Cloud platforms provide access to cutting-edge technologies, such as machine learning, artificial intelligence, and Internet of Things (IoT), enabling organizations to innovate and stay competitive.

**11. Pay-as-You-Go Model:**

- Cost Control: Organizations only pay for the resources they consume. This pay-as-you-go model is cost-effective, particularly for startups and small businesses.

**12. Managed Services:**

- Outsourced Management: Cloud providers offer managed services, reducing the burden on organizations to manage and maintain certain aspects of their infrastructure. This includes services like managed databases and serverless computing.

Understanding these benefits helps organizations make informed decisions about adopting cloud computing, aligning their IT strategies with business goals, and leveraging the advantages of a scalable and flexible computing environment.

Azure Global Infrastructure :



**Azure region pairs**



Each Azure region is always paired with another region within the same geography

Data centers are usually 300+ miles apart

Automatic replication and failover for some azure services.

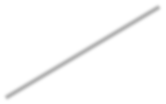
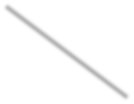
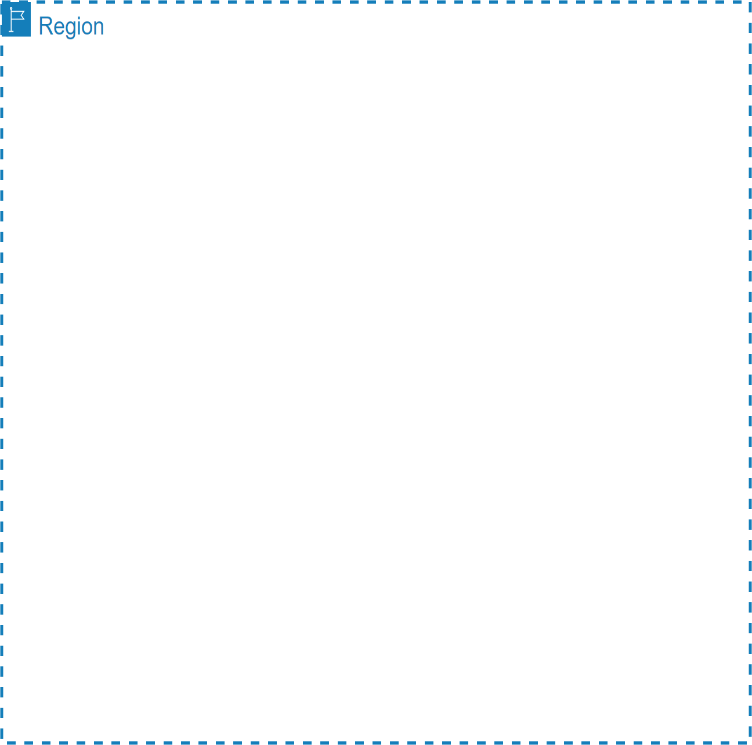
Additional advantages of region pairs:

If an extensive Azure outage occurs, one region out of every pair is prioritized to make sure at least one is restored as quickly as possible for applications hosted in that region pair.

Planned Azure updates are rolled out to paired regions one region at a time to minimize downtime and risk of application outage.

Data continues to reside within the same geography as its pair.

**Availability Zones**



High-speed, private

fiber-optic networks

Region represents a separate geographic area.

Availability zone is a set of discrete data centers.

Availability zone is set up to be an isolation boundary. If one zone goes down, the other continues working.

Each availability zone has independent power, cooling and networking.

Availability zones are connected via high bandwidth, ultra-low latency networking

AZs are physically separated by several kilometers, while being within 100 km (60 miles) of one each.

All AZ traffic is encrypted.

Not every region has support for availability zones.

**Azure Management Infrastructure:**

The management infrastructure includes Azure resources and resource groups, subscriptions, and accounts. Understanding the hierarchical organization will help you plan your projects and products within Azure.

**Azure resources and resource groups:**

A resource is the basic building block of Azure. Anything you create, provision, deploy, etc. is a resource. Virtual Machines (VMs), virtual networks, databases, cognitive services, etc. are all considered resources within Azure.

Resource groups are simply groupings of resources. When you create a resource, you’re required to place it into a resource group. While a resource group can contain many resources, a single resource can only be in one resource group at a time. Some resources may be moved between resource groups, but when you move a resource to a new group, it will no longer be associated with the former group. Additionally, resource groups can't be nested, meaning you can’t put resource group B inside of resource group A.

Resource groups provide a convenient way to group resources together. When you apply an action to a resource group, that action will apply to all the resources within the resource group. If you delete a resource group, all the resources will be deleted. If you grant or deny access to a resource group, you’ve granted or denied access to all the resources within the resource group.

When you’re provisioning resources, it’s good to think about the resource group structure that best suits your needs.

For example, if you’re setting up a temporary dev environment, grouping all the resources together means you can deprovision all of the associated resources at once by deleting the resource group. If you’re provisioning compute resources that will need three different access schemas, it may be best to group resources based on the access schema, and then assign access at the resource group level.

There aren’t hard rules about how you use resource groups, so consider how to set up your resource groups to maximize their usefulness for you.

Azure subscriptions

In Azure, subscriptions are a unit of management, billing, and scale. Similar to how resource groups are a way to logically organize resources, subscriptions allow you to logically organize your resource groups and facilitate billing.

Using Azure requires an Azure subscription. A subscription provides you with authenticated and authorized access to Azure products and services. It also allows you to provision resources. An Azure subscription links to an Azure account, which is an identity in Azure Active Directory (Azure AD) or in a directory that Azure AD trusts.

* An account can have multiple subscriptions, but it’s only required to have one. In a multi-subscription account, you can use the subscriptions to configure different billing models and apply different access-management policies. You can use Azure subscriptions to define boundaries around Azure products, services, and resources. There are two types of subscription boundaries that you can use:
* Billing boundary: This subscription type determines how an Azure account is billed for using Azure. You can create multiple subscriptions for different types of billing requirements. Azure generates separate billing reports and invoices for each subscription so that you can organize and manage costs.
* Access control boundary: Azure applies access-management policies at the subscription level, and you can create separate subscriptions to reflect different organizational structures. An example is that within a business, you have different departments to which you apply distinct Azure subscription policies. This billing model allows you to manage and control access to the resources that users provision with specific subscriptions.

**Create additional Azure subscriptions:**

Similar to using resource groups to separate resources by function or access, you might want to create additional subscriptions for resource or billing management purposes. For example, you might choose to create additional subscriptions to separate:

* **Environments**: You can choose to create subscriptions to set up separate environments for development and testing, security, or to isolate data for compliance reasons. This design is particularly useful because resource access control occurs at the subscription level.
* **Organizational structures**: You can create subscriptions to reflect different organizational structures. For example, you could limit one team to lower-cost resources, while allowing the IT department a full range. This design allows you to manage and control access to the resources that users provision within each subscription.
* **Billing**: You can create additional subscriptions for billing purposes. Because costs are first aggregated at the subscription level, you might want to create subscriptions to manage and track costs based on your needs. For instance, you might want to create one subscription for your production workloads and another subscription for your development and testing workloads.

**Azure management groups:**

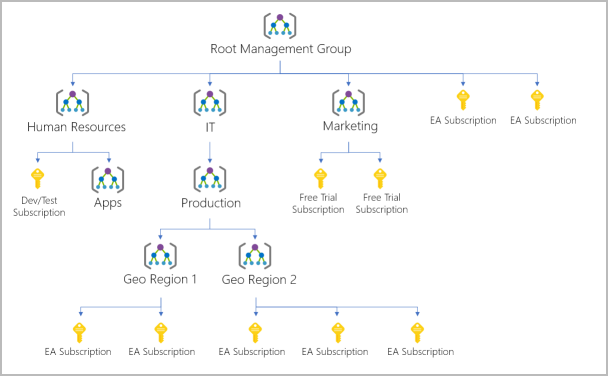
The final piece is the management group. Resources are gathered into resource groups, and resource groups are gathered into subscriptions. If you’re just starting in Azure that might seem like enough hierarchy to keep things organized. But imagine if you’re dealing with multiple applications, multiple development teams, in multiple geographies.

**If you have many subscriptions, you might need a way to efficiently manage access, policies, and compliance for those subscriptions**. Azure management groups provide a level of scope above subscriptions. You organize subscriptions into containers called management groups and apply governance conditions to the management groups. All subscriptions within a management group automatically inherit the conditions applied to the management group, the same way that resource groups inherit settings from subscriptions and resources inherit from resource groups.

Management groups give you enterprise-grade management at a large scale, no matter what type of subscriptions you might have. Management groups can be nested.

**Management group, subscriptions, and resource group hierarchy:**

You can build a flexible structure of management groups and subscriptions to organize your resources into a hierarchy for unified policy and access management. The following diagram shows an example of creating a hierarchy for governance by using management groups.



**How Management Groups Work:**

* Management groups form a hierarchy, with a root management group at the top.
* Each management group can have child management groups, and each management group can contain subscriptions.
* Policies, RBAC assignments, and other governance controls applied at a management group are inherited by its child management groups and subscriptions.
* Management groups provide a way to structure your Azure environment, aligning with your organization's structure or specific projects.

Example Hierarchy:

* Root Management Group
  + Department A
    - Subscriptions 1, 2
  + Department B
    - Subscriptions 3, 4

Considerations:

* The root management group is automatically created when an Azure AD directory is associated with an Azure subscription.
* Azure Policy and RBAC assignments at a management group are inherited by child management groups and subscriptions, but they can be overridden at lower levels.
* A subscription can only be a member of one management group.

Azure Identities and Governance:

Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service. It is designed to help your organization manage and secure user identities and access to applications and resources in the cloud and on-premises. Azure AD is a crucial component of Microsoft's Azure cloud platform and is widely used for authentication and authorization in various scenarios.

Key features and functionalities of Azure Active Directory include:

1. Identity Management:

- User Authentication: Azure AD provides authentication services, allowing users to sign in using their Azure AD credentials (username and password).

- Multi-Factor Authentication (MFA): Azure AD supports MFA to add an extra layer of security by requiring users to provide additional verification, such as a code sent to their mobile device.

2. Single Sign-On (SSO):

- Azure AD enables Single Sign-On, allowing users to access multiple applications and services with a single set of credentials after initial authentication.

3. Application Management:

- App Registration: Developers can register their applications with Azure AD, allowing them to integrate with Azure AD for authentication and authorization.

- Gallery Applications: Azure AD includes a gallery of pre-integrated applications, making it easier for administrators to configure and manage access to these applications.

4. Identity Protection:

- Azure AD Identity Protection uses advanced machine learning and security analytics to detect and respond to potential identity risks and threats.

5. B2B and B2C Collaboration:

- Azure AD B2B (Business-to-Business): Allows organizations to securely share their applications and resources with guest users from other organizations while maintaining control over their own corporate data.

- Azure AD B2C (Business-to-Consumer): Enables organizations to provide identity and access management solutions for customer-facing applications.

6. Conditional Access:

- Allows organizations to define policies that control access to applications and resources based on conditions such as user location, device compliance, and risk level.

7. Role-Based Access Control (RBAC):

- Azure AD supports RBAC, allowing administrators to assign roles to users, groups, and applications to control access to Azure resources.

8. Integration with On-Premises Directories:

- Azure AD can be integrated with on-premises Active Directory using Azure AD Connect, providing a seamless and unified identity solution for hybrid environments.

9. Self-Service Password Reset:

- Users can reset their passwords or unlock their accounts without the need for administrator intervention, improving user productivity and reducing helpdesk requests.

Azure Active Directory is a fundamental component for building secure and scalable identity solutions in the cloud. It plays a central role in enabling secure access to various Microsoft and third-party services within the Azure ecosystem.

Windows server AD (AD DS ) vs Azure AD :

Windows Server Active Directory (AD DS) and Azure Active Directory (Azure AD) are both identity and access management solutions from Microsoft, but they serve different purposes and are used in different scenarios. Here's a comparison of Windows Server Active Directory (AD DS) and Azure Active Directory (Azure AD):

|  |  |  |
| --- | --- | --- |
|  | Windows server Active directory (ADDS) | Azure Active directory |
| Authentication Protocols | Supports Kerberos, NTLM, LDAP, and others for on-premises authentication. | Supports OAuth 2.0, OpenID Connect, SAML, and others for cloud-based authentication. |
| Directory Objects | Manages objects such as users, groups, computers, and organizational units within the on-premises Active Directory. | Manages similar objects but in the cloud. It includes users, groups, and devices, and these objects can be synchronized with on-premises AD DS. |
| Hybrid Scenarios: | Windows Server AD DS: Can be integrated with Azure AD using tools like Azure AD Connect, allowing for hybrid scenarios where on-premises AD DS is extended to the cloud. | Supports hybrid identity scenarios, allowing organizations to synchronize on-premises AD DS with Azure AD for a unified identity experience across on-premises and cloud resources. |
| Authorization and Access Management | Manages authorization and access to resources within the on-premises network. | Manages access to cloud-based applications, services, and resources. It also provides Conditional Access policies to control access based on various conditions. |
| Authentication | Provides authentication services for on-premises resources. It supports Kerberos and NTLM authentication protocols | Provides authentication services for cloud-based resources. It supports modern authentication protocols like OAuth 2.0 and OpenID Connect. It also supports federation with on-premises AD DS for a single sign-on experience. |
| Use Cases | Windows Server AD DS: Primarily used for traditional on-premises scenarios. It manages authentication and authorization for users and devices within the corporate network. | Designed for cloud-based scenarios. It manages identities for cloud services, including Office 365, Azure, and other Software as a Service (SaaS) applications. It is also used for identity and access management in hybrid environments, connecting on-premises AD DS to the cloud. |
| Deployment Location | Windows Server AD DS: It is typically deployed on on-premises servers. Organizations set up their own physical or virtual servers to run Active Directory Domain Services for managing user identities, authentication, and authorization within their network. | It is a cloud-based service provided by Microsoft and is used for identity and access management in the cloud. Azure AD is not a direct replacement for Windows Server AD DS but complements it, especially in hybrid or cloud-centric scenarios. |

In summary, Windows Server AD DS is focused on traditional on-premises identity and access management, while Azure AD is designed for cloud-based identity services. Many organizations use both in a hybrid configuration to leverage the strengths of each in a unified identity solution. Azure AD Connect is a key tool for connecting and synchronizing identities between on-premises AD DS and Azure AD.

Azure AD Roles :

* When you create any user inside azure AD, it will will ask for group id and role.
* Once you mention the group , you have select some predefined azure roles.

In Azure Active Directory (Azure AD), roles are used to manage access to Azure AD and other Microsoft Online Services. Roles in Azure AD help define what users can and cannot do within the Azure AD environment, including tasks related to identity and access management. There are different types of roles, each serving a specific purpose. Here are the main types of roles in Azure AD:

1. Azure AD Roles:

- Global Administrator: A role that has full access to all administrative features in Azure AD. Global Administrators can assign administrator roles to others, manage user accounts, and configure all aspects of Azure AD.

- User Administrator: This role allows the management of user accounts, password resets, and user attributes, but it does not have the ability to assign administrator roles to other users.

- Billing Administrator: Limited to managing billing information for an Azure subscription. This role does not have access to other administrative features.

- Conditional Access Administrator: Manages Conditional Access policies, which control access to applications based on conditions such as user location, device compliance, and risk level.

- Application Administrator: Manages application registrations and settings in Azure AD. This role is often assigned to developers or administrators responsible for configuring and managing applications integrated with Azure AD.

- Cloud Application Administrator: Similar to the Application Administrator role but focused on managing applications specifically designed to run in the cloud.

- Security Administrator: Has full access to the Azure AD Security Center and is responsible for configuring security-related settings and responding to security incidents.

- Security Reader: Has read-only access to the Azure AD Security Center and can view security-related settings and reports.

- Authentication Administrator: Manages authentication methods, including Multi-Factor Authentication (MFA) settings.

- Privileged Role Administrator: Manages Azure AD roles that provide access to sensitive tasks, such as assigning administrator roles.

2. Directory Roles:

- Directory Readers: Has read-only access to Azure AD and can view information but cannot make any changes.

- Directory Writers: Has the ability to write (create, update, delete) information in Azure AD, excluding administrative roles and licenses.

- Directory Administrators: Has full administrative access to Azure AD and can manage all aspects of the directory.

3. Application Roles:

- Application roles are specific to individual applications integrated with Azure AD. These roles are defined by the application developer and control access within the application.

Roles in Azure AD are assigned to users or groups, and users can have multiple roles based on their responsibilities. Assigning the appropriate roles is crucial for ensuring that users have the necessary permissions to perform their tasks without providing unnecessary access. Role assignments can be managed through the Azure portal, Azure AD PowerShell, or the Microsoft Graph API.

Azure AD roles VS RBAC Roles :

* Assigned roles are Azure AD roles and Azure role assignment are RBAC roles.

Azure AD (Azure Active Directory) roles and Azure RBAC (Role-Based Access Control) roles serve different purposes within the Azure ecosystem. It's important to understand the distinctions between these two types of roles:

Azure AD Roles:

* roles are primarily related to managing identities within Azure AD. They define permissions for tasks related to user and group management, conditional access, and application registrations.
* Typically apply to Azure AD as a whole and are often associated with administrative tasks related to user identity, authentication, and security.
* Examples of Azure AD Roles:

- Global Administrator: Full access to all administrative features in Azure AD.

- User Administrator: Manages user accounts, password resets, and user attributes.

- Application Administrator: Manages application registrations and settings in Azure AD.

Azure RBAC Roles:

* These roles are focused on managing access to Azure resources, such as virtual machines, storage accounts, databases, and more. RBAC allows you to control who can perform actions on specific resources within your Azure subscription.
* Apply to specific Azure resources, resource groups, or subscriptions. They define permissions for actions such as reading, writing, deleting, and managing Azure resources.
* Examples of Azure RBAC Roles:

- Owner: Full access to all resources and can delegate access to others.

- Contributor: Can create and manage all types of Azure resources but cannot grant access to others.

- Reader: Can view existing Azure resources but cannot make any changes.

Distinctions:

* Azure AD roles are more focused on identity and access management within Azure AD itself.
* Azure RBAC roles are specifically designed for controlling access to Azure resources.
* Azure AD roles are often associated with administrative tasks related to user and group management.
* Azure RBAC roles are associated with managing and controlling access to Azure resources, infrastructure, and services.
* Azure AD roles are generally more focused on administrative responsibilities related to the Azure AD service.
* Azure RBAC roles provide fine-grained control over access to specific Azure resources and services.

In summary, while Azure AD roles focus on identity and access management within Azure AD, Azure RBAC roles are designed for controlling access to Azure resources. Both play crucial roles in securing and managing access within the Azure environment, and they often complement each other in a holistic access control strategy.

Role Based Access Control (RBAC) :

Secure access management for cloud resources is critical for businesses that operate in the cloud. Role-based access control (RBAC) is a mechanism that can help you manage who can access your Azure resources. RBAC lets you determine what operations specific users can do on specific resources, and control what areas of a resource each user can access.

Azure RBAC is an authorization system built on Azure Resource Manager. Azure RBAC provides fine-grained access management of resources in Azure.

### Things to know about Azure RBAC

Here are some things you can do with Azure RBAC:

* Allow an application to access all resources in a resource group.
* Allow one user to manage VMs in a subscription, and allow another user to manage virtual networks.
* Allow a database administrator (DBA) group to manage SQL databases in a subscription.
* Allow a user to manage all resources in a resource group, such as VMs, websites, and subnets.

### Azure RBAC concepts

The following table describes the core concepts of Azure RBAC.

| **Concept** | **Description** | **Examples** |
| --- | --- | --- |
| **Security principal** | An object that represents something that requests access to resources. | User, group, service principal, managed identity |
| **Role definition** | A set of permissions that lists the allowed operations. Azure RBAC comes with built-in role definitions, but you can also create your own custom role definitions. | Some built-in role definitions: Reader, Contributor, Owner, User Access Administrator |
| **Scope** | The boundary for the requested level of access, or "how much" access is granted. | Root, management group, subscription, resource group, resource |
| **Assignment** | An **assignment** attaches a **role definition** to a **security principal** at a particular **scope**. Users can grant the access described in a role definition by creating (attaching) an assignment for the role. | - Assign the User Access Administrator role to an admin group scoped to a management group - Assign the Contributor role to a user scoped to a subscription |

### Things to consider when using Azure RBAC

As you think about how you can implement roles and scope assignments within your organization, consider these points:

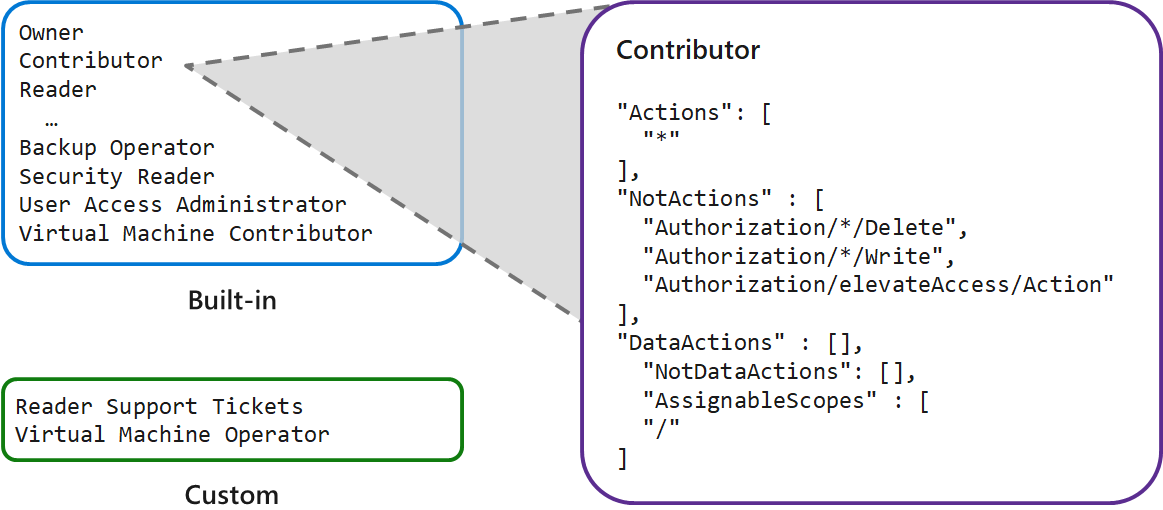
* **Consider your requestors**. Plan your strategy to accommodate for all types of access to your resources. Security principals are created for anything that requests access to your resources. Determine who are the requestors in your organization. Requestors can be internal or external users, groups of users, applications and services, resources, and so on.
* **Consider your roles**. Examine the types of job responsibilities and work scenarios in your organization. Roles are commonly built around the requirements to fulfill job tasks or complete work goals. Certain users like administrators, corporate controllers, and engineers can require a level of access beyond what most users need. Some roles can be defined to provide the same access for all members of a team or department for specific resources or applications.
* **Consider scope of permissions**. Think about how you can ensure security by controlling the scope of permissions for role assignments. Outline the types of permissions and levels of scope that you need to support. You can apply different scope levels for a single role to support requestors in different scenarios.
* **Consider built-in or custom definitions**. Review the built-in role definitions in Azure RBAC. Built-in roles can be used as-is, or adjusted to meet the specific requirements for your organization. You can also create custom role definitions from scratch.

# Create a role definition

A role definition consists of sets of permissions that are defined in a JSON file. Each permission set has a name, such as Actions or NotActions that describes the purpose of the permissions. Some examples of permission sets include:

* Actions permissions identify what actions are allowed.
* NotActions permissions specify what actions aren't allowed.
* DataActions permissions indicate how data can be changed or used.
* AssignableScopes permissions list the scopes where a role definition can be assigned.

The following diagram shows details for the Contributor role in Azure RBAC, which has three sets of permissions.



The Actions permissions show the Contributor role has all action privileges. The asterisk "\*" wildcard means "all." The NotActions permissions narrow the privileges provided by the Actions set, and deny three actions:

* Authorization/\*/Delete: Not authorized to delete or remove for "all."
* Authorization/\*/Write: Not authorized to write or change for "all."
* Authorization/elevateAccess/Action: Not authorized to increase the level or scope of access privileges.

The Contributor role also has two DataActions permissions to specify how data can be affected:

* "NotDataActions": []: No specific actions are listed. Therefore, all actions can affect the data.
* "AssignableScopes": ["/"]: The role can be assigned for all scopes that affect data.

Here's another example of a role definition in PowerShell:

PowerShellCopy

Name: Owner

ID: 01010101-2323-4545-6767-987453021523

IsCustom: False

Description: Manage everything, including access to resources

Actions: {\*} # All actions allowed

NotActions: {} # No actions denied

AssignableScopes: {/} # Role can be assigned to all scopes

Review the following characteristics of role definitions:

* Azure RBAC provides built-in roles and permissions sets. You can also create custom roles and permissions.
* The Owner built-in role has the highest level of access privilege in Azure.
* The system subtracts NotActions permissions from Actions permissions to determine the effective permissions for a role.
* The AssignableScopes permissions for a role can be management groups, subscriptions, resource groups, or resources.

### Role permissions

Use the Actions and NotActions permissions together to grant and deny the exact privileges for each role. The Actions permissions can provide the breadth of access and the NotActions permissions can narrow the access.

The following table shows how the Actions or NotActions permissions are used in the definitions for three built-in roles: Owner, Contributor, and Reader. The permissions are narrowed from the Owner role to the Contributor and Reader roles to limit access.

| **Role name** | **Description** | **Actions permissions** | **NotActions permissions** |
| --- | --- | --- | --- |
| Owner | Allow all actions | \* | n/a |
| Contributor | Allow all actions, except write or delete role assignment | \* | - Microsoft.Authorization/\*/Delete - Microsoft.Authorization/\*/Write - Microsoft.Authorization/elevateAccess/Action |
| Reader | Allow all read actions | /\*/read | n/a |
|  |  |  |  |

### Role scopes

After you define the role permissions, you use the AssignableScopes permissions to specify how the role can be assigned. Let's look at a few examples.

* Scope a role as available for assignment in two subscriptions:

"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e", "/subscriptions/e91d47c4-76f3-4271-a796-21b4ecfe3624"

* Scope a role as available for assignment only in the Network resource group:

"/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e/resourceGroups/Network"

* Scope a role as available for assignment for all requestors:

"/"

### Things to consider when creating roles

Consider the following points about creating role definitions in Azure RBAC:

* **Consider using built-in roles**. Review the list of [built-in role definitions](https://learn.microsoft.com/en-us/azure/role-based-access-control/built-in-roles) in Azure RBAC. There are over 100 pre-defined role definitions to choose from, such as Owner, Backup Operator, Website Contributor, and SQL Security Manager. Built-in roles are defined for several categories of services, tasks, and users, including General, Networking, Storage, Databases, and more.
* **Consider creating custom definitions**. Define your own [custom roles](https://learn.microsoft.com/en-us/azure/role-based-access-control/custom-roles) to meet specific business scenarios for your organization. You can modify the permissions for a built-in role to meet the specific requirements for your organization. You can also create custom role definitions from scratch.
* **Consider limiting access scope**. Assign your roles with the minimum level of scope required to perform the job duties. Some users like administrators require full access to corporate resources to maintain the infrastructure. Other users in the organization can require write access to personal or team resource, and read-only access to shared company resources.
* **Consider controlling changes to data**. Identify data or resources that should only be modified in specific scenarios and apply tight access control. Limit users to the least of amount of access they need to get their work done. A well-planned access management strategy helps to maintain your infrastructure and prevent security issues.
* **Consider applying deny assignments**. Determine if you need to implement the deny assignment feature. Similar to a role assignment, a deny assignment attaches a set of deny actions to a user, group, or service principal at a particular scope for the purpose of denying access. Deny assignments block users from performing specific Azure resource actions even if a role assignment grants them access.

# Create a role assignment

A role assignment is the process of scoping a role definition to limit permissions for a requestor, such as a user, group, service principal, or managed identity.

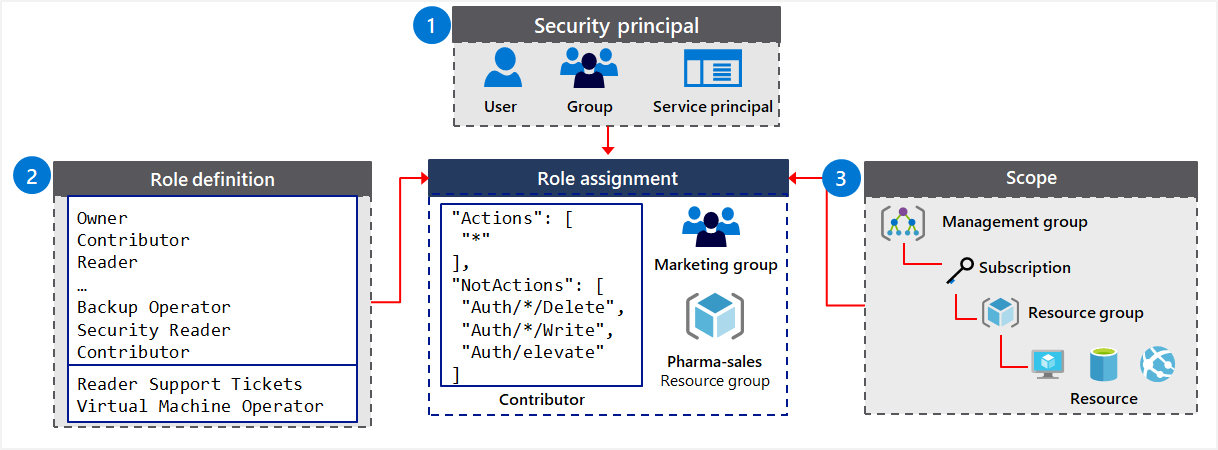
### Things to know about role assignments

Review the following characteristics of role assignments:

* The purpose of a role assignment is to control access.
* The scope limits which permissions defined for a role are available for the assigned requestor.
* Access is revoked by removing a role assignment.
* A resource inherits role assignments from its parent resource.
* The effective permissions for a requestor are a combination of the permissions for the requestor's assigned roles, and the permissions for the roles assigned to the requested resources.

### Things to consider when assigning scope levels for roles

The following diagram shows an example of how scopes can be applied for a role to grant varying levels of access for different users. Think about how you can implement scopes for your roles to create meaningful assignments for your organization.



This scenario has the following access management configuration:

* Three security principals are supported: user, group, service principal.
* Six built-in roles are implemented, and two custom roles are defined: Reader Support Tickets and Virtual Machine Operator.
* The built-in Contributor role has two sets of permissions: Actions and NotActions.
* The Contributor role is assigned at different scopes to the Marketing group and Pharma-sales resource group:
  + Users in the Marketing group are granted access to create or manage any Azure resource in the Pharma-sales resource group.
  + Marketing users aren't granted access to resources outside the Pharma-sales resource group, unless they have another role assignment that grants them access to the resource group.

# Compare Azure roles to Azure Active Directory roles

Three types of roles are available for access management in Azure:

* Classic subscription administrator roles
* Azure role-based access control (RBAC) roles
* Azure Active Directory (Azure AD) administrator roles

To better understand how these different roles are defined and implemented in Azure, it helps to know some of the history.

When Azure was initially released, access to resources was managed with just three administrator roles: Account Administrator, Service Administrator, and Co-Administrator. Access was controlled by assigning admin roles to subscriptions.

Later, role-based access control (RBAC) for Azure resources was added. Azure RBAC is a newer authorization system that provides fine-grained access management to Azure resources. RBAC includes many built-in roles that can be assigned at different scopes. The Azure RBAC model also lets you create your own custom roles.

In addition to Classic subscription admin roles and Azure RBAC roles, Azure AD provides built-in administrator roles to manage Azure AD resources like users, groups, and domains.

**Tip**

If you're considering using Classic administrator roles, use Azure Resource Manager roles instead. The following table highlights differences between Azure RBAC roles and Azure AD administrator roles.

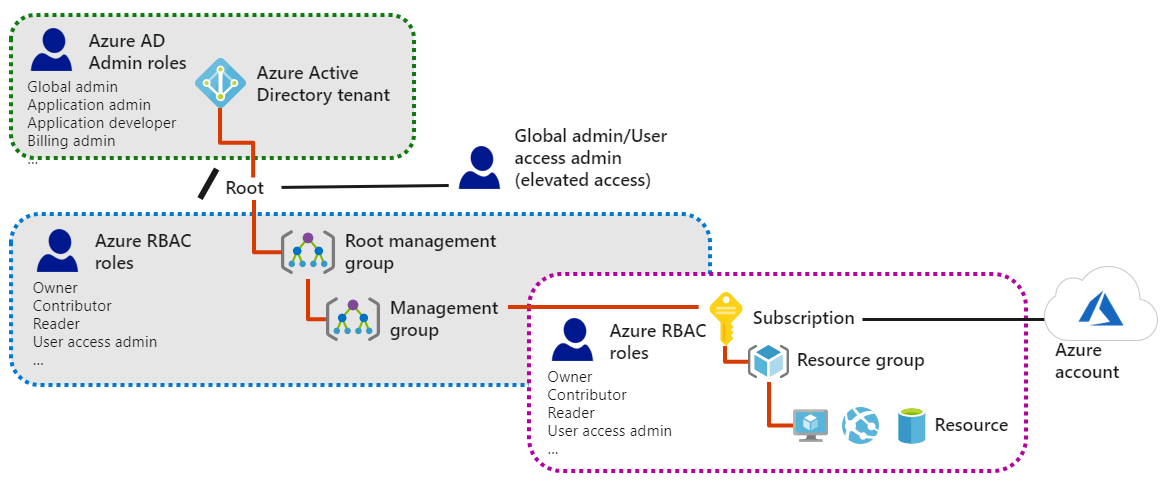
|  | **Azure RBAC roles** | **Azure AD admin roles** |
| --- | --- | --- |
| **Access management** | Manages access to Azure resources | Manages access to Azure AD resources |
| **Scope assignment** | Scope can be specified at multiple levels, including management groups, subscriptions, resource groups, and resources | Scope is specified at the tenant level |
| **Role definitions** | Roles can be defined via the Azure portal, the Azure CLI, Azure PowerShell, Azure Resource Manager templates, and the REST API | Roles can be defined via the Azure admin portal, Microsoft 365 admin portal, and Microsoft Graph Azure AD PowerShell |

# Apply role-based access control

Built-in role definitions in Azure RBAC are defined for several categories of services, tasks, and users. You can assign built-in roles at different scopes to support various scenarios, and build custom roles from the base definitions.

Azure Active Directory (Azure AD) also provides built-in roles to manage resources in Azure AD, including users, groups, and domains. Azure AD offers [administrator roles](https://learn.microsoft.com/en-us/azure/active-directory/roles/permissions-reference) that you can implement for your organization, such as Global admin, Application admin, and Application developer.

The following diagram illustrates how you can apply Azure AD administrator roles and Azure RBAC roles in your organization.



* **Azure AD admin roles** are used to manage resources in Azure AD, such as users, groups, and domains. These roles are defined for the Azure AD tenant at the root level of the configuration.
* **Azure RBAC roles** provide more granular access management for Azure resources. These roles are defined for a requestor or resource and can be applied at multiple levels: the root, management groups, subscriptions, resource groups, or resources.

# Review fundamental Azure RBAC roles

Azure RBAC provides over 100 pre-defined role definitions. Roles can grant access to data within an object. If a user has read data access to a storage account, then they can read the blobs or messages in the storage account.

The following table describes four built-in Azure RBAC role definitions that are considered fundamental.

| **Fundamental role** | **Description** |
| --- | --- |
| Owner | The Owner role has full access to all resources, including the right to delegate access to others. The Service Administrator and Co-Administrators roles are assigned the Owner role at the subscription scope. |
| Contributor | The Contributor role can create and manage all types of Azure resources. This role can't grant access to others. |
| Reader | The Reader role can view existing Azure resources. |
| User Access Administrator | The User Access Administrator role can manage user access to Azure resources. |

Azure role-based access control (RBAC) is a system that enables granular access management of Azure resources. Azure Administrators use Azure RBAC to segregate duties within a team, and grant users the specific access they need to perform their jobs.

In this module, you identified the features and use cases for RBAC. You discovered how to create role definitions and role assignments, and find and use built-in Azure RBAC roles. You explored how to use RBAC to manage access to subscriptions with RBAC. You reviewed the differences between Azure RBAC and Azure Active Directory (Azure AD) roles.

## Azure subscriptions

First, remember that each Azure subscription is associated with a single Azure AD directory. Users, groups, and applications in that directory can manage resources in the Azure subscription. The subscriptions use Azure AD for single sign-on (SSO) and access management. You can extend your on-premises Active Directory to the cloud by using **Azure AD Connect**. This feature allows your employees to manage their Azure subscriptions by using their existing work identities. When you disable an on-premises Active Directory account, it automatically loses access to all Azure subscriptions connected with Azure AD.

Using Azure requires an Azure subscription.

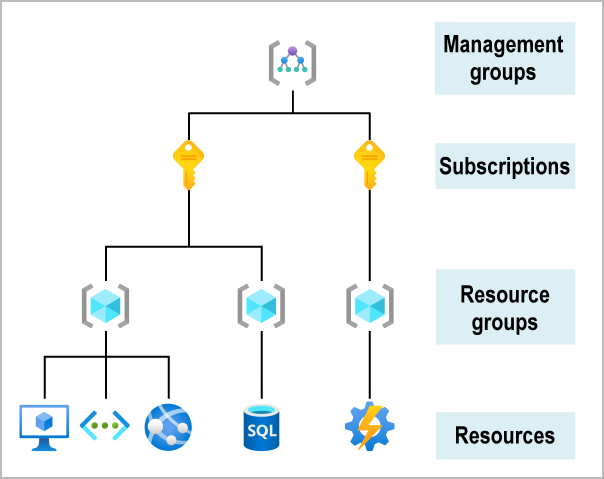
An Azure subscription is a logical unit of Azure services that links to an Azure account. It also allows you to provision resources.

subscription provides you with authenticated and authorized access to Azure products and services.

* Azure generates separate billing reports and invoices for each subscription
* Two types of subscription boundaries
  + Billing boundary
  + Access control boundary
* You can create separate subscription based on:
  + Environment: development and testing, security, or to isolate data for compliance reasons
  + Organizational structures: IT, HR, Admin and so on
  + Billing: manage and track costs based on your needs, for example –Production, Test and Dev
* Different types of Subscription:
  + FREE: An email address and a credit card are required to sign up for a free trial subscription that provides $200 credit for thefirst 30 days and 12 months of restricted access.
  + Pay-Per-Use: Charges monthly based on Cloud resource use.
    - Enterprise: A single Enterprise agreement is established for large subscription purchases, including savings for new licensesand Software Assurance.
    - Student: This membership includes $100 for 12 months and may be activated without a credit card.

Trust Relation between AAD/Tenant and Subscription

* An Azure subscription has a trust relationship with Azure Active Directory (Azure AD).
* A subscription trusts Azure AD to authenticate users, services, and devices.
* Multiple subscriptions can trust the same Azure AD directory. Each subscription can only trust a single directory.



Azure Tags

* Azure tags are the name-value pairs that help to organize the Azure resources in the Azure portal.
* Azure Tags are simply labels that you can attach to your Azure resources.
* You can use tags to easily group and classify resources and assets in Azure.For example, explore of the costs generated by resources having the same tag applied.
* Tagging is a primary way to understand the data in any cost or billing reporting.
* Resources don’t inherit any Azure tags applied at the Resource Group level.
* It’s a fundamental part of any well-manage environment. It’s also the first step in establishing proper governance of any environment.
* Azure Policy can be used to enforce tagging rules and conventions.For example, you can require that certain tags be added to new resources as they are provisioned.

Azure POlicy

* Azure Policy can help you control or restrict or audit your resources.
* Enforce rules on Azure resources configurations to make sure they remain compliant with corporate standards.
* You can apply individual policy or group of policy (initiatives).
* Two imp tasksPrevent noncompliant resources from being created
* Highlights existing resources that aren't compliant with the policies.
* Examples:Allows only a certain SKU size for the virtual machines (VMs) to be provisioned.
* Mandatory tags to be created while provisioning resources
* MFA should be enabled on accounts with write permissions on your subscription
* Assign policy within a specific scope (management group, a single subscription, or a resource group.)
* Policy assignments are inherited by all child resources within that scopeYou can exclude specific child resources you need to be exempt from the policy assignment
* You can review the noncompliant policy results and take any action that's needed.

Management groups

* Management groups let you organize multiple subscriptions as a single management entity to facilitate easier management.
* You can create managements groups in a hierarchical structure with the top level of the hierarchy at the tenant level and containing all subscriptions in that tenant.
* Any conditions applied to a management group apply to all subscriptions contained in that management group object.
* Each management group and subscription can support only one parent.
* Each management group can have many children.
* The root management group can't be moved or deleted, unlike other management groups.

Cost Management

* This is a built-in service that gives you a breakdown of the usage and cost of your Azure resources.
* This allows you to see what is costing you money and how it compares against your budget.
* You use Cost Management + Billing features to:Conduct billing administrative tasks such as paying your bill
* Manage billing access to costs
* Download cost and usage data that was used to generate your monthly invoice
* Proactively apply data analysis to your costs
* Set spending thresholds
* Identify opportunities for workload changes that can optimize your spending

**Azure Storage :**

* 1. Diff types of data and requirements
  + Relational, non-relational/No-SQL, datasheets, images, videos, backups
  + Storage, access, security, availability, latency, processing, backup
  1. Diff types of Data Service
* Azure Blobs: Text and binary data
* Azure Files: Managed file shares (SMB Protocol)
* Azure Queues: Messaging
* Azure Tables: NoSQL store

Features

* Durable and highly available –
* redundancy across datacenters or regions
* Secure –all data encrypted by default
* Scalable –massively scalable
* Managed -Azure handles hardware maintenance, updates, and critical issues for you.
* Accessible -accessible from anywhere in the world over HTTP or HTTPS.Clients libraries are available in all languages
* Support scripting in PowerShell or Azure CLI
* You can create storage account by searching storage account> create new storage

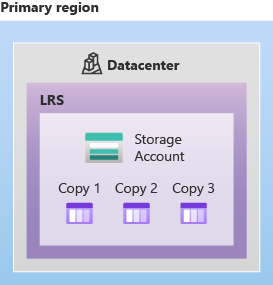
Azure Storage Data Redundancy :

* Protect your data from hardware failures, network or power outages, and massive natural disasters.
* Even in the event of a failure, redundancy ensures your storage account's availability and durability.
* Tradeoffs between lower costs and higher availability
* Redundancy in the primary region
  + Locally redundant storage (LRS) –Three synchronous copies in same data center
  + Zone-redundant storage (ZRS) –Three synchronous copies in three availability zones (AZs)
* Redundancy in a secondary region
  + Geo-redundant storage (GRS) –LRS + Asynchronous copy to secondary region ()
  + Geo-zone-redundant storage (GZRS)
* With GRS or GZRS, the data in the secondary region isn't available for read or write access unless there is a failover to the secondary region.
* For read access to the secondary region, configure your storage account to use Read-access geo-redundant storage (RA-GRS)
* Read-access geo-zone-redundant storage (RA-GZRS).

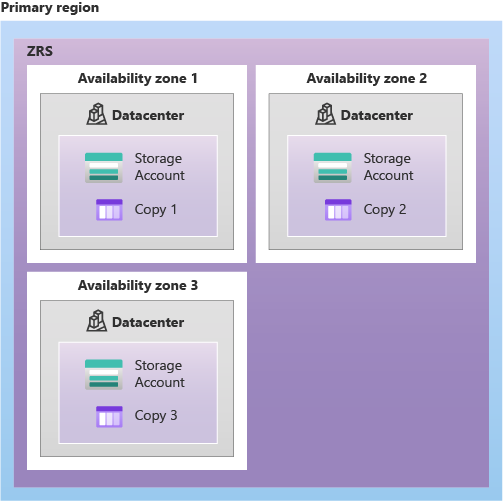
Azure Storage Redundancy

* Locally redundant storage (LRS) –Three synchronous copies in same data center
* Zone-redundant storage (ZRS) –Three synchronous copies in three availability zones (AZs)
* Geo-redundant storage (GRS) -LRS + Asynchronous copy to secondary region (three more copies using LRS) –Read only access Read-access geo-redundant storage (RA-GRS) –Read Access on GRS
* Geo-zone-redundant storage (GZRS) –ZRS + Asynchronous copy to secondary region (three more copies using LRS) –Read only access Read-access geo-zone-redundant storage (RA-GZRS) –Read Access on GZRS

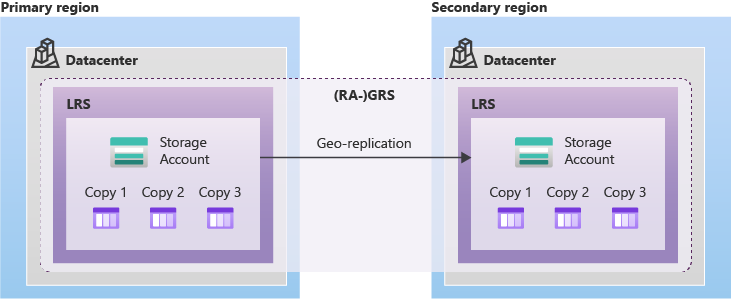
Locally Redundant Storage (LRS) has the lowest level of consistency in Azure storage redundancies. The LRS option stores three copies of the data within a single data center. So even if there is one failure within the data center, your data will still be available. But, if the entire region experiences an outage, your data may not be available anymore. Cost-wise, LRS is the cheapest option. Only use LRS if you can tolerate some degree of data loss.



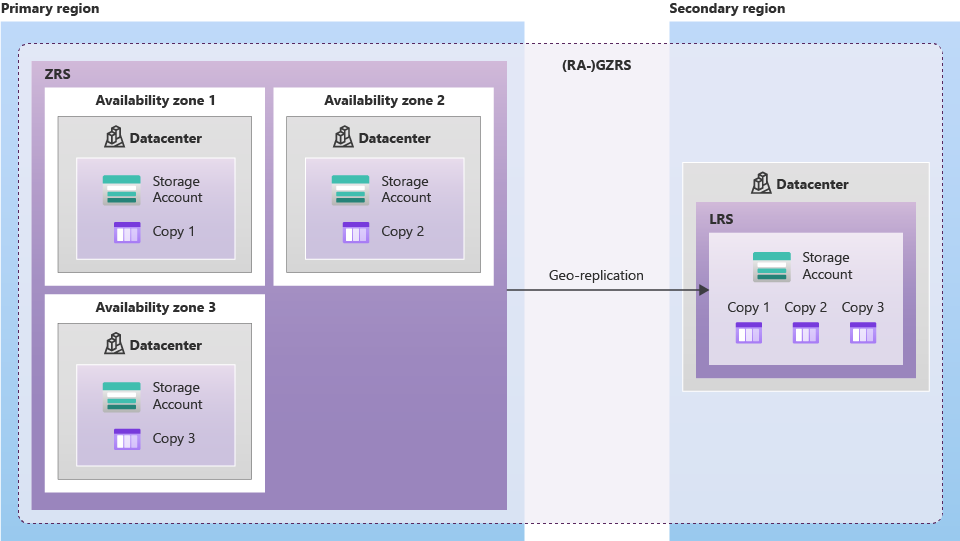
Zone-Redundant Storage (ZRS) is your best redundancy option when it comes to single-region redundancy. ZRS replicates your data synchronously across three Azure availability zones in your primary region. Remember that each availability zone is a data center with independent cooling, power, and networking. This means that even if there is a failure in one availability zone, your data can still be reached from another availability zone. ZRS provides a higher level of protection than LRS, but it is also more expensive.



Geo-Redundant Storage (GRS) is like the LRS option, but it adds a secondary region for durability. GRS copies your data synchronously three times within a single data center in your primary region. It then copies your data asynchronously to a single data center in a secondary region using LRS, which provides protection against regional and data center failures. This is a much more expensive version of LRS since you have multiple copies of your data in your primary region and another copy in a region, but it also provides a high level of data protection and availability to your data.



Geo-Zone-Redundant Storage (GZRS) protects your data by copying it to three availability zones in the primary region using ZRS. It then replicates data asynchronously to a single data center in the secondary region using LRS. It is similar to GRS and ZRS but also provides a higher level of data protection and availability by storing data in multiple availability zones. GZRS is the most expensive redundancy option. So you should only use this if you have applications that require maximum durability, consistency, availability, and resilience for disaster recovery.



For storage accounts in GRS or GZRS, in the event that you are experiencing an outage, data in the secondary region is not accessible right away to your applications or users unless you initiate a failover. This failover updates the DNS of your storage account. The DNS change will convert your secondary region to your new primary region. Once the failover is successful, your applications or users will have read-and-write access to the new primary region.

Read-Access Geo-Redundant Storage (RA-GRS) is almost the same as GRS, but it provides read-only access to data in the secondary region during an outage in the primary region. This means that even if the primary region is unavailable, the data is still accessible for read-only operations. This is helpful for applications that require read-only operations, even with failure.

Choosing the right storage consistency option for your application depends on your specific needs and budget. LRS is the cheapest option and is suitable for applications that can accommodate some degree of data loss. ZRS provides a higher level of protection than LRS, but it is also more expensive. GRS is the least expensive form of consistency in multi-region redundancy of Azure storage, while GZRS is the most expensive, providing the highest data protection and availability level.

Blob Storage

* 1. Blob -Binary Large Object
  + Any type or format
  + Text, Images, audio, video, excel, backup files

1. A diagram of a diagram

   Description automatically generated
   1. Use cases:
   * Storing files for shared access
   * Video and audio streaming
   * Storing data for analysis (Data Lake Gen2)
   * Writing to the log file
   * Storing data for disaster recovery, backup, and archiving
   1. Flat structure
   2. Provides a unique namespace in Azure for your data.
   * http://mystorageaccount.blob.core.windows.net

go to storage account > in side the storage account > left side container > create container > insode container

A screenshot of a computer

Description automatically generated

Three types of Blob Storage

* 1. Block Blobs:
  + For large objects that doesn't use random read and write operations, files that are read from beginning to end
  + Such as media files or image files for websites.
  1. Page Blobs:
  + Optimized for random read and write operations.
  + Provide durable disks for Azure Virtual Machines (Azure VMs)
  1. Append Blobs:
  + Optimized for append operations. e. g. Logs
  + When you modify an append blob, blocks are added to the end of the blob only
  + Updating or deleting of existing blocks is not supported
  + For example, you might write all of your trace logging to the same append blob for an application running on multiple VMs

A diagram of a diagram

Description automatically generated

**Storage Access Tiers**

* Data stored in the cloud can be different based on how it's generated, processed, and accessed over its lifetime.
* Pricing
  + The volume of data stored/month
  + Types of operations performed
  + Number of operations performed
  + Data transfer cost, if any
  + The selected data redundancy option
* Organize your data based on attributes like frequency of access and planned retention period.

**Performance Tier:**

* The performance tier refers to the level of performance that is associated with Azure Blob Storage. As of my knowledge cutoff in January 2022, Azure Blob Storage offers two performance tiers:

**Standard Performance Tier:**

* Suitable for general-purpose storage needs.
* Offers a balance between storage cost, availability, and durability.
* Provides higher transaction costs compared to the Premium tier.

**Premium Performance Tier:**

* Designed for scenarios that require high-performance and low-latency storage.
* Ideal for workloads that demand high transaction rates and consistent low latency.
* Generally associated with higher storage costs compared to the Standard tier.
* Organizations choose the performance tier based on the specific requirements of their applications and workloads. The Premium tier is typically used for scenarios where the highest levels of performance are critical.

**Blob Access Tier:**

The blob access tier refers to how frequently the data in a blob is accessed and the associated storage costs. Azure Blob Storage supports the following access tiers:

* 1. Hot
  + Frequently accessed data
  + Example -images for your website
  + Low latency
  + Higher access cost
  1. Cool
  + Infrequent accessed data
  + Example -invoices for your customers
  + High latency
  + Lower cost
  + Stored for at least 30 days
  1. Archive
  + Rarely accessed data Example -long-term backups
  + Highest access times and access cost
  + Latency in hours
  + Stored for at least 180 days
  + Use Case: Business policy mandated Data Archiving, long term retention like healthcare data

|  |  |
| --- | --- |
| Standard | Premium |
| * + Backed by magnetic drives   + Support –All storage account   + Optimized for high capacity and high throughput   + Provides the lowest cost per GB.   + Best for applications that require bulk storage or where data is accessed infrequently. Example: Backup and DR datasets, media, pictures, videos. | * + Backed by solid state drives (SSD)   + Good for virtual machines and workloads that need low latency and high I/O performance   + Examples: transactional databases, big data analysis, IOT, AI or ML   + Block Blob: Best for high transaction rates or low storage latency   + File Shares: Best for enterprise or high performance applications that need to scale   + Page blobs: Best for random read and write operations |

Note: You cannot change performance tier after account creation

**Azure Table Storage**

* + NoSQL key-value Storage
  + Items are referred to as rows, and fields are known as columns
  + All rows in a table must have a key
  + No concept of relationships, stored procedures, secondary indexes, or foreign keys
  + Data will usually be denormalized
  + To help ensure fast access, Azure Table Storage splits a table into partitions
  + Support very large volume of Data
  + Consider Cosmos DB for new development
  1. Advantages
  + It's simpler to scale
  + A table can hold semi-structured data
  + No complex relationships
  + Data insertion and retrieval is fast
  1. Good to use for:
  + Storing TBs of structured data capable of serving web scale applications
  + Storing datasets that don't require complex joins, foreign keys, or stored procedures, and that can be denormalized for fast access.
  + Capturing event logging and performance monitoring data.

**Azure Queue Storage**

* + Store large numbers of messages.
  + Access messages via authenticated calls using HTTP or HTTPS.
  + May contain millions of messages, up to the total capacity limit of a storage account.
  + Queues are commonly used to create a backlog of work to process asynchronously.

**Azure File Storage**

* Enables you to create files shares in the cloud, and access these file shares from anywhere with an internet connection
* Mounted concurrently by cloud or on-premises deployments.
* Accessible from Windows, Linux, and macOS clients.
* Accessible Server Message Block (SMB) protocol or Network File System (NFS) protocol
* Azure Files ensures the data is encrypted at rest, and the SMB protocol ensures the data is encrypted in transit.
* Use Cases
  + Replace or supplement on-premises file servers
  + Share application settings
  + Dev/Test/Debug
* Key Benefits
  + Shared access: Replace on-premises file shares with Azure file shares without application compatibility issues
  + Fully managed: Azure will manage hardware or an OS
  + Resiliency: you don’t have to deal with local power and network issues.

Azure Disk Storage

* VM uses disks as a place to store an operating system, applications, and data in Azure.
* One virtual machine can have one OS disk and multiple Data disk but one data disk can only be link with one VM.
* Both the OS disk and the data disk are virtual hard disks (VHDs) stored in an Azure storage account.
* The VHDs used in Azure is .
* vhd files stored as page blobs in a standard or premium storage account in Azure.
* Unmanaged disks: We can create a storage account and specify it when we create the disk.
  + Not recommended, previous unmanaged disks should migrate to managed disk
* Managed disk
  + Azure creates and manages storage accounts in the background.
  + We don't have to worry about scalability issues.
  + Azure creates and manages the disk for us based on the size and performance tier we specify.
* Managed Disk types:
  + Standard HDD: Backup, non
  + critical, infrequent access
  + Standard SSD: lightly used production applications or dev/test environments
  + Premium SSD disks: Super fast and high performance, very low latency, recommended for production and performance sensitive wo
  + rkl oads
  + Ultra disks (SSD): for most demanding IOintensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), an

Azure Storage security :

As of my last knowledge update in January 2022, there isn't a specific service or tool called "Azure File Explorer" provided by Microsoft Azure. However, there are several tools and methods you can use to explore and manage files in Azure Storage, particularly in Azure File Storage.

Azure Storage Explorer:

Azure Storage Explorer is a standalone application that you can use to manage Azure Storage resources, including Blob Storage, Table Storage, Queue Storage, and Azure Files. It allows you to view and interact with your storage resources in a graphical user interface (GUI).

Key features of Azure Storage Explorer:

- Connect to Multiple Accounts: You can connect to multiple Azure subscriptions and storage accounts.

- Browse and Manage Files: Navigate through containers, shares, and file directories in Azure Storage.

- Upload and Download: Transfer files between your local machine and Azure Storage.

- Manage Blob Containers, Queues, and Tables: Perform various operations on blobs, queues, and tables.

Azure Portal: You can also explore Azure Files directly through the Azure portal. Navigate to your storage account and look for the "Azure Files" option. This allows you to browse and manage your Azure file shares within the portal.

PowerShell and Azure CLI: Command-line tools such as PowerShell and Azure CLI also provide options for exploring and managing files in Azure Storage. You can use commands to list files, upload/download files, and perform other operations.

Visual Studio Code Extensions: There are extensions available for Visual Studio Code that provide functionality for exploring and managing Azure Storage resources. These extensions often include features like browsing containers, uploading and downloading files, and managing Azure Storage entities.

Third-Party Tools: Various third-party tools also offer file explorer-like functionalities for Azure Storage. These tools may provide additional features and customization options beyond what the official tools offer.

Azure storage Authorization options:

Data Storage Authorization

Anonymous

* Public access, no authorization is required
* Use Case: Website, online documentation
* Only for Blob

Shared Key authorization

* Data can be accessed using access keys

Shared Access Signatures (SAS)

* Provide limited delegated access
* Constraints: time interval, permissions

Azure Active Directory

* Role-based access control -Fine-grained access can be provided to users, groups, or applications

**Shared Access Keys Authorization**

* Two default access keys are generated with every storage account
* Whoever has these keys, can access entire storage account
* Make sure they're safe by not storing or hard coding them in your application code.Store these on key vault
* If keys compromised, you can regenerate keys
* Recommendation: periodically rotate the keys
* Consider Azure AD instead



* In shared access signature, you will have option to choose the access key number
* You will be able to connect to file explorer or access file storage by using that key only.
* If you change the SAS key number in SAS, you will not be able to access.
* You can get the keys and regenerate keys in access keys option.

Stored access policy:

A stored access policy is a set of constraints that can be used by multiple shared access signatures.

* Contains permission and time interval
* Only work with service level shared access signature
* Go to storage accounts > go to specific object like container for blon storage > lefdt side you can see the option access policy
* You can see this access policy is SAS key

Azure AD Authorization

* Azure storage can use Azure AD to authorize requests
* Storage Account level or service level (currently Blob and Queue only)
* Use Role-based access control (RBAC)
  + User, group or service principle
* Microsoft recommended approach.
  + Benefit: No longer need to store credentials within application config files
  + More secure, easy to implement and manage
* Similar to IIS application pool identity approach

Azure Storage Netwroking :

* You can modify the options in networking to allow or bloc, who can access or from which ip address you can access the storage account.

A screenshot of a computer

Description automatically generated

### Overview:

- Essentials: Basic information about the storage account, including performance, replication, and access tier.

### Activity log:

- Audit Trail: A log of activities and events related to the storage account.

### Tags:

- Metadata: Label resources with descriptive tags for better organization and management.

### Diagnose and solve problems:

- Troubleshooting: Tools and resources to identify and resolve issues with the storage account.

### Access control (IAM):

- Role Assignments: Manage access to resources by assigning roles to users, groups, or applications.

### Data migration:

- Transfer Services: Options and tools for migrating data to and from Azure Storage.

### Events:

- Event Grid: Event-driven programming with serverless architecture.

### Storage browser:

- Azure Storage Explorer: A tool to explore and manage Azure Storage resources.

### Storage mover:

- Azure Data Box: Physical devices for offline data transfer to Azure.

### Data storage:

- Container: Organize blobs (objects) within the storage account.

- File shares: Manage file shares in Azure Files.

- Queues: Message-based communication between components.

- Tables: NoSQL data storage.

### Security + networking:

- Networking: Configuring network rules and firewalls.

- Front Door and CDN: Content Delivery Network configurations for performance.

- Access Keys: Securely manage access keys for authentication.

- Shared Access Signature: Grant limited access to resources.

- Encryption: Data encryption at rest and in transit.

- Microsoft Defender for Cloud: Security and threat protection.

### Data Management:

- Redundancy: Configure data redundancy for high availability.

- Data Protection: Recover data from accidental modification or deletion.

- Object Replication: Replicate data across Azure regions.

- Blob Inventory: Inventory and manage blobs.

- Static Website: Host static websites from Azure Storage.

- Lifecycle Management: Automate data lifecycle policies. You can write policies to move your storage from hot to cool or vice versa.

- Azure AI Search: Integration with Azure Cognitive Search.

### Settings:

- Configuration: Configure storage account settings.

- Data Lake Gen2 Upgrade: Upgrade to Azure Data Lake Storage Gen2.

- Resource Sharing CORS: Configure Cross-Origin Resource Sharing.

- Advisor Recommendations: Recommendations for optimizing performance.

- Endpoints: Configure public and private endpoints.

- Locks: Prevent accidental deletion or modification.

### Monitoring:

- Insights: Gain insights into storage account usage.

- Alerts: Set up alerts for specific metrics.

- Metrics: Monitor performance metrics.

- Workbooks: Create customized data visualizations.

- Diagnostic Settings: Configure data collection for diagnostics.

- Logs: Access logs for auditing and analysis.

### Automation:

- Tasks: Automate common tasks.

- Export Template: Export ARM templates for deployment.

Blob lifecycle management

* Transition blobs to a cooler storage tier (hot to cool, hot to archive, or cool to archive) to optimize for performance and cost.
* General Purpose V2 and blob stored
* Generally take 24 hours to deploy

Blob object replication

* Copy data from source to destination storage account
* A replication policy includes one or more rules that specify a source container and a destination container and indicate which block blobs in the source container will be replicated.
* What exactly copy?
  + All versions Blob
  + Blob’s metadata
  + Blob’s properties
  + NOT –snapshots are not copied
* Both the source and destination accounts should have:
  + Versioning enable
  + Either from Hot and Cool tier
* Scenarios
  + Minimize read request latency
  + Process data at single location and replicate results across multiple regions so that users in regions only access the results instead of entire dataset.
* Consider Cost while replicating

1. Source Storage

* Location –East US
* Enable Versioning and change feed
* Create Replication Rules at source

1. Destination Storage

* Location -India
* Enable Versioning
* Asynchronous replication

A yellow box with black text

Description automatically generated

Azure File Sync

* Replication occurs between Windows servers in your data centers and Azure.
* Provide local caching for your users. You can have as many caches as you want.
* By default, all files are tied to Azure Files, but with Cloud Tiering enabled, only frequently accessed files are cached locally on the server.
* You can access your data locally using SMB, NFS, or FTPS on Windows Server.

Advantages

* Lift and shift
* Backup and Disaster Recovery
* File Archiving

A diagram of a cloud computing process

Description automatically generated

**Import and export service**

Move small amount of data –Internet

* AzCopy
* Azure Storage Explorer

Move large amount of data (TBs) between on-premises and Azure storage securely.

Scenarios

* Migrating data to the cloud
* Backup
* Data recovery

Issues

* Network is slow
* Getting more network bandwidth is cost-prohibitive

Solution

* Ship disk drive physically
* Disk drive –you can use your own or ones provided by Microsoft
  + Own -Solid-state drives (SSDs) or Hard disk drives (HDDs)
  + Microsoft –Azure Data Box
* Import large amounts of data to Azure Blob storage and Azure Files by shipping disk drives to an Azure datacenter.
* Export large amounts of data from Azure Blob storage to disk drives and ship to your on-premises sites.

Import jobs

A diagram of data transfer

Description automatically generated

Note: Import large amounts of data to Azure Blob storage and Azure Files

Prepare disks

* Use WAImportExporttool to copy the data to disk/drive
* Generate journal files
  + Journal file stores basic information such as drive serial number, encryption key, and storage account details.
* Use BitLockerto encrypt the drive

Create import job

* Refer Azure Storage Account
* Specify the shipping and receiving address

Ship the disks to the destination that you specified when creating the import job

Update the job by providing the shipment tracking number.

**Export jobs**

1. Export large amounts of data from Azure Blob storage
2. Identify the data and number of disks using WAImportExporttool

Create Export job

* + Specify Azure Storage account -> Blobs
  + Return address

Ship the disk

* + Ship the disk to the Azure region hosting the storage account
  + Update the job by providing the shipment tracking number

Azure datacenter tasks

* + Receive and copy the data
  + Encrypt the data using Bitlocker
  + Ship them back

Your tasks

* + Bitlockerkeys will be available on Azure portal.
  + Unlock and verify drive with WAImportExportunlock command
  + Copy data to your on-premises storage

**Import/Export Tool (WAImportExport)**

* Command line tool, downloadable from Microsoft
* Need 64-bit windows only, no Linux or MacOS support
  1. **What it does?**
* Determine number of drives needed for export job
* Data copy
* Encryption or decryption of drive with BitLocker
* Creation of journal files needed for import jobs
  1. Two versions
* Version 1 is for import/export for Azure Blob storage
* Version 2 is for import of Azure Files
  1. The Azure Import/Export Tool prepares and repairs drives for the Microsoft Azure Import/Export service.
* Before Import job –copy data to hard drives
* After import job -repair any blobs that were corrupted, were missing, or conflicted with other blobs.
* Before an export job –Identify the number of drives needed for export jobs.
* After an export job -you can use this tool to repair any corrupted or missing files on the drives.

**Azure Data Box**

* Microsoft provides you a piece of hardware in three different sizes developed specifically for import and export tasks.
* You can order the Data Box device via the Azure portal to import or export data from Azure.
* Ideally suited to transfer data sizes larger than 40 TBs
* Scenarios: Onetime migration, Initial bulk transfer, Disaster recovery, Migrate back to on-premises or to another cloud service provider

**AzCopy**

* Command-line utility
* Available to download and install on Windows, Linux, and Mac
* Use it to copy data to/from Microsoft Azure Blob and File storage
  + you can copy data between a file system and a storage account, or between storage accounts.
* AzCopyis preinstalled in Azure Cloud Shell, so you can use it there if you can't run it locally.
* Simple commands
  + List of available commands: azcopy–help
  + Basic syntax for AzCopycommands: azcopy copy [source] [destination] [flags]
  + azcopycopy"C:\local\path" "https://account.blob.core.windows.net/mycontainer1/?sv=2018-03-28&ss=bjqt&srt=sco&sp=rwddgcup&se=2019-05-01T05:01:17Z&st=2019-04-30T21:01:17Z&spr=https&sig=MGCXiyEzbtttkr3ewJIh2AR8KrghSy1DGM9ovN734bQF4%3D" --recursive=true

A screenshot of a computer

Description automatically generated

**Virtual Machine :**

* You can create VM by using virtual machine service.
* There are different configurations available while creating the VM.
* We have to mention bellow setting configuration for creating the VM.
* Choose size as per the requirement. There are variety of sizes available.
* You can choose the images in the image setting.
* Resize VM : Once the vm is created you can increase or decrease the size by changing the configuration in availability + scale > size > select the VM size > click on resize

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Move VM 🡺

* Move VM : you can see the move option in overview > Resource group or overview>subscription. That means you can move your VM to another resource group or move your vm from from current subscription to different subscription.
* You have to move the VM along with dependent resources. Or else it will fail.
* Move dependent resources to another resource group or subscription : go to resource group > select all > click on move
* It will create new resource ids after moving and update the scripts.
* It is not possible to move VM scale sets with standard load balancer and standard public ip.
* Virtual machine that use a key vault for disk encryption cant be moved. You can disable the encryption and move.
* Prior to moving , restore points on vms configured with azure backup must be deleted.

A screenshot of a computer

Description automatically generated

Disk Type :

* You can add additional disk to your vm. Choose option : settings > Disks > create and attach a new disk
* VM uses disks as a place to store an operating system, applications, and data in Azure. These are like physical disk in an on-premises server but, virtualized.
* Types of Disks
  + - Standard harddisk drives (HDD) -Backup, non-critical, infrequent access
    - Standard solid-state drives (SSD) -Web servers, lightly used enterprise applications and dev/test
    - Premium solid-state drives (SSD) -Production and performance sensitive workloads
    - Ultra disks -IO-intensive and other transaction-heavy workloads.
* one virtual machine can have one OS disk and multiple Data disk
* One data disk can only be link with one VM
* Both the OS disk and the data disk are virtual hard disks (VHDs) stored in an Azure storage account.
* Benefits
* Highly available (three replicas)99.999% availability
* Scalable- 50,000 VM disks
* Granular access control-Use RBAC to assign specific permissions for a managed disk to one or more users.
* Integration with availability sets and Availability Zones, Backups and so on
* Security
  + Private links-allow you to generate SAS token
* Encryption
  + Server -side encryption data encryption at rest, default
  + Azure Disk encryption-encrypt the OS and Data disks
* Disk Roles
  + OS Disk-Pre installed OS, contain boot volume
  + Data Disk-Store application data
* Temporary Disk
* Short-term storage for applications and processes
* May be lost during a maintenance or redeploy.
* Successful reboot will persist data.

ReDeploy Virtual machine : left pane options > HELP > redeploy +reapply > redeploy

* Redeploy a azure vm may help trouble shoot remote desktop (RDP) or application access issue.
  + Can not connect via RDPO
  + If you are not able to access applications on VMS
* Steps: power off VM, move to a new node and then power back on
* Retain all configuration options and resources. All data saved on the temporary disks will be lost.
* The virtual network interfaces dynamic IP address will be updated.
* Use Azure PowerShell or Azure portal
* Redeploy and Reapply , both are different.

A screenshot of a computer

Description automatically generated

Latency Vs IOPS Vs Throughput :

* Latency is a time taken to respond to an I/O request.
* IOPS is a unit of measurement for the number of read and write operations performed per second.
  + Number of read write operations mostly useful for OLTP transactions used in Azure for DBs like sql server
  + Latency is proportional to IOPS

A screenshot of a computer

Description automatically generated

* You can change an ultra-disk’s performance parameters without having to restart your virtual machines.
* Ultra disks can't be used as OS disks, they can only be created as empty data disks. Ultra disks also can't be used with some features and functionality, including disk snapshots, disk export, changing disk type, VM images, availability sets, Azure Dedicated Hosts, or Azure disk encryption.

Disk snapshot:

* A snapshot is a full, read-only copy of a virtual hard disk (VHD).
* You can take a snapshot of both operating system (OS) or data disk VHDs.
* Use Case:
  + Point-in-time backup
  + Restore or rebuild a VM
  + Troubleshoot virtual machine (VM) issues

Go to VM > left pane > Disks > go inside the disk > top menu click on create snapshot

Go to snap shot > create a disk from the snapshot

Go to VM > attach the new disk

A screenshot of a computer

Description automatically generated

Azure Shared Disks:

* Allow you to attach a managed disk to multiple virtual machines (VMs)simultaneously.
* Only ultra disks, premium SSDs, and standard SSDs can enable shared disks
* Shared managed disks do not come with a fully managed file system accessible through SMB/NFS.
* Use a cluster manager like Windows Server Failover Cluster (WSFC) or Pacemaker to handle cluster node communication and write locking.
* Go to Disk > configuration > Enable Shared Disk

**Managed vs Unmanaged Disk**

Managed Disks

* Managed by Azure
* High availability, secure
* Azure create storage account behind the scene

Unmanaged Disks

* Not Managed by Azure
* You create storage account
* Full control over data
* You have to take care of encryption, data recovery plans etc.
* You cannot create both managed and unmanaged disks on a VM

Data Encryption in VM :

* Server Side Encryption encrypts your disks at the storage account level, at rest.
  + Default, always enabled, can’t turn off.
* Azure Disk Encryption-encrypts your disks at the VM OS level.
  + Use BitLocker for Windows VMs and DM-Crypt for Linux VMs.
  + Data encrypt during transit
  + Temporary disk and OS data disk caches are also encrypted.
  + Requires an Azure key vault to control and manage disk encryption keys and secrets.
  + Key vault and VMs must reside in the same location and subscription
  + ADE can only be paired with server side encryption with PMK.
* Encryption at host - Also encrypt your temporary disk and cache at host.
  + Does not use your VM's CPU and doesn't impact your VM's performance.
  + The temp disk will always be encrypted by a PMK
  + End to End data encryption at rest.
  + Encryption at host does not use your vms CPU and does not impact our vms performance.

A screenshot of a computer

Description automatically generated

Virtual Machine Networking :

* Virtual network enables Azure resources to securely communicate with each other.
* Subnet is nothing but it divide virtual network to multiple components.
* Virtual Network (VNet) enables Azure resources to securely communicate with each other
* Subnets:
  + Divide a network into two or more networks
  + A subnet is a range of IP addresses in the virtual network.
  + For organization and security.
  + No security boundary b/w subnets by default
* Network Interface Card (NIC)
  + Interconnection between a virtual machine and a virtual network.
  + IMP : Each NIC attached to a VM must exist in the same location and subscription as the VM.
* Network security group (NSG)
  + Allow or deny network traffic to subnets, NICs, or both.
  + NSGs can be associated with either subnets or individual NICs connected to a subnet.
  + When an NSG is associated with a subnet, the ACL rules apply to all the
  + VMs in that subnet. Traffic to an individual NIC can be restricted by
  + associating an NSG directly to a NIC.
* IP addresses
  + Enable Azure resources to communicate to Internet and public facing Azure services.
  + Can be dynamic (Default) or static.
  + Reserve a static IP address if you need a fixed IP address that won't change for example, if you need to add the IP address to a safe list.
* IMP
  + : Deallocating your virtual machine releases your dynamic public IP
* A load balancer balance or distribute incoming Internet traffic to multiple VMs
* Public Load Balancers are used to load balance internet traffic to your VMs.
* Internal load balancers are used to load balance traffic inside a virtual network.

A diagram of a computer network

Description automatically generated

Azure Bastion

* Provide secure management connectivity to virtual machines in a virtual network.
* Enables connections without exposing a public IP on the VM.
* Fully managed platfor,m PaaS service from Azure.
* No public IP address required on the Azure VM.
* Through bastion you can protct your VM from exposing RDP/SSH ports to outside world, while you can still proving secure access using RDP/SSH
* Steps:
  + Disassociate the ip address of an existing VM or you can dis associate while creating the VM
  + Vm > connect > deploy bastion (it will do all the steps automatically including adding subnets)

Virtual Machine restart VS Stop-start

|  |  |
| --- | --- |
| Restart | Stop-start |
| * IP address will remain unchanged. * Temporary storage data will remain unchanged. | * Ip address will be changed * Will loose temporary data as it deallocates once it is stopped |

Creating Linux VM :

* Use can create linux VM using password or SSH key.
* If you choose password, you can connect using public ip, user id, password in putty
* If you choose to ssh key,
  + SSH is an encrypted connection protocol that allows secure sign-ins over unsecured connections.
  + SSH is the default connection protocol for Linux VMs hosted in Azure.
  + SSH Keys -Secure and preferred method of connecting to a VM using SSH is by using a public-private key pairPublic key is placed on your Linux VM, or any other service that you wish to use with public-key cryptography.
  + Private key remains on your local system. Protect this private key. Do not share it.
  + As soon as you connect with an SSH client to your Linux VM, it verifies that you have the private key on your PC. The client gets access to the VM if it has the private key
  + Download the pem file> open key gen > upload the pem file > save private key
  + Open using putty

Boot Diagnostics:

* You can check screenshots and logs of vm logs.
* It will help you to analyse if there is an issue in restarting.

Cost Reduction methods:

* Reservation for longer time
* You ca calculate the saving using pricing calculator
* You can choose hybrid model , it will reduce the licensing part
* Use SPOT VM :
  + Spot pricing provides access to azure compute resources at deep discounts when unused azure capacity is available
  + If azure needs the capacity back, spot vms can be evicted with a 30 second notice
  + You can set the maximum price that you agree to pay.
  + Your vms are automatically evicted when the current spot price is higher than the maximum price you agree to pay of if Azure no longer has compute capacity available
  + Best for interruptible workloads ( batch processing , dev/test environments, non critical tasks, etc..)
  + You can check Advisor section > cost section. It will show you some advise to save some cost.
  + Reside underutilised vm
  + Deallocate vm during off hours
  + Delete unused resources
  + Migrate from iaas to paas
  + Choose low cost location and reservation
* A screenshot of a computer

  Description automatically generated

Availability Sets:

* Availability Sets make use of two key concepts Fault Domains, and Update Domains.
* Update domains define the group of virtual machines that are going to be patched/maintained/rebooted at same time.
* Fault domains define the group of virtual machines that share a common power source and network switch.
* It saves from rackwide failure, or a rackwide maintenance window that can take down all VMs hosted on this single point of failure.
* Availability sets are free to use! You only pay for the virtual machines being created.
* It does not protect your application from operating system or application specific failures, it does limit the impact of potential physical hardware

A row of blue boxes with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

Virtual machine scale sets:

* Create and manage a group of load balanced VMs.
* Allows your application to automatically scale as resource demand changes
  + The number of VM instances can automatically increase or decrease in response to demand or a defined schedule.
* All VM instances are created from the same base OS image and configuration.
  + VM size, disk configuration, and application installs should match across all VMs.
* Provides high availability and application resiliency.
  + Can use availability zones or availability sets
* There is no cost for the scale set itself, you only pay for each VM instance that you create.
* A diagram of a person with a load balancer

  Description automatically generated

A screenshot of a computer program

Description automatically generated

How to create a scale set :

* Search scale set > create a new scale set
* Mention the resource group name and choose manual or auto scaling
* Click on edit option and change the configuration of scaling > mention scale in , scale out details
* Once you create the scale set, go to scale sets> selected new created scale set > go to instances
* You can see there would be 1 instance running, but if we have more load on the cpu, we will get one mora vm created automatically.
* Search scale set > create > mention all the details > scaling mode auto scaling > configure > scaling condition > edit option > mention scale in out rules
* It will create scale set . inside scale set > instances, you ca see the VM is created. Login to vm and increase the cpu usage by mre than 80 %. You can see 1a new vm will be created automatically. stress --cpu 1000

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Azure Resource Manager :

* Every resource gets created or modified through ARM
* When you use azure portal, it sends the request to ARM. ARM then deploys the resources
* Atomate resource deployments (create, update, and delete) using templates.
* ARM template is a JSON file that defines what you want to deploy to Azure.
* Integrates with Azure portal, PowerShell, CLI, and REST API to perform deployment and management tasks.
* Easy way to deploy multiple resource instances or reliably redeploy resources.
* ARM template can be used to deploy the resources consistently and repeatedly.
* Define the dependencies between resources so they're deployed in the correct order.

A diagram of a software resource manager

Description automatically generated

* Deployment and management service for Azure
* All Azure resource activities are routed via ARM.
* Describe the resources in a declarative JSON format
* ARM template is verified before any code is executed to ensure that the resources will be created and connected correctly
* Automatic Rollback in case of failure
* The template then orchestrates the creation of those resources in parallel. You can provision multiple vms parallelly
* Templates can even execute PowerShell and Bash scripts before or after the resource has been set up
* Creates all dependencies in the correct order
* Save previous scripts for version control
* ARM templates define your application's infrastructure requirements for a repeatable deployment that is done in a consistent manner
* Why not PowerShell or CLI?No validation step in these tools
* If a script encounters an error, the dependency resources can't be rolled back easily
* Deployments happen serially
* You have to figure out dependencies

ARM Template structure :

A screenshot of a computer

Description automatically generated

* Schema: defines the location of the JSON file that describes the version of the template language.▪Content version: Version of the template. Use this value to document significant changes in your template.
* apiprofile: an API version that serves as a collection of API versions for resource types.
* Parameters: Values that are provided when deployment is executed to customize resource deployment.
* Variables: Values used in templates as JSON fragments to simplify template language expressions.
* Functions: User-defined functions that are available within the template.
* Resources: Resource types that are deployed or updated in a resource group or subscription.
* Outputs: Values that are returned after deployment.

ARM Template editor :

* Go to azure portal
* Search deploy > custom deployment template
* Edit your ARM template
* You can do in visual studio code > download install ARM tools plugin

Example: create storage account using ARM

* Write the ARM template file for creating storage account
* Go to custom deployment template >Edit your ARM > upload your template file >choose subscription > review create

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"resources": [

{

"type": "Microsoft.Compute/virtualMachines",

"apiVersion": "2022-11-01",

"name": "[parameters('vmName')]",

"location": "[parameters('location')]",

"properties": {

"hardwareProfile": {

"vmSize": "[parameters('vmSize')]"

},

"osProfile": {

"computerName": "[parameters('vmName')]",

"adminUsername": "[parameters('adminUsername')]",

"adminPassword": "[parameters('adminPassword')]"

},

"storageProfile": {

"imageReference": {

"publisher": "Canonical",

"offer": "UbuntuServer",

"sku": "18.04-LTS",

"version": "latest"

},

"osDisk": {

"createOption": "FromImage"

}

},

"networkProfile": {

"networkInterfaces": [

{

"id": "[resourceId('Microsoft.Network/networkInterfaces', parameters('nicName'))]"

}

]

}

}

},

{

"type": "Microsoft.Network/networkInterfaces",

"apiVersion": "2022-11-01",

"name": "[parameters('nicName')]",

"location": "[parameters('location')]",

"dependsOn": [

"[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicIPName'))]"

],

"properties": {

"ipConfigurations": [

{

"name": "ipconfig1",

"properties": {

"privateIPAllocationMethod": "Dynamic",

"publicIPAddress": {

"id": "[resourceId('Microsoft.Network/publicIPAddresses', parameters('publicIPName'))]"

}

}

}

]

}

},

{

"type": "Microsoft.Network/publicIPAddresses",

"apiVersion": "2022-11-01",

"name": "[parameters('publicIPName')]",

"location": "[parameters('location')]",

"properties": {

"publicIPAllocationMethod": "Dynamic"

}

}

],

"parameters": {

"vmName": {

"type": "string",

"metadata": {

"description": "Name of the virtual machine"

}

},

"location": {

"type": "string",

"metadata": {

"description": "Location for all resources."

}

},

"adminUsername": {

"type": "string",

"metadata": {

"description": "Admin username for the virtual machine."

}

},

"adminPassword": {

"type": "secureString",

"metadata": {

"description": "Admin password for the virtual machine."

}

},

"vmSize": {

"type": "string",

"metadata": {

"description": "Size of the virtual machine."

}

},

"nicName": {

"type": "string",

"metadata": {

"description": "Name of the network interface."

}

},

"publicIPName": {

"type": "string",

"metadata": {

"description": "Name of the public IP address."

}

}

}

}

**Create ARM for multiple instances :**

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"variables": {

"numberOfStorageAccounts": 3

},

"resources": [

{

"type": "Microsoft.Storage/storageAccounts",

"apiVersion": "2022-06-01",

"name": "[concat('storageaccount', copyIndex())]",

"location": "[parameters('location')]",

"sku": {

"name": "Standard\_LRS",

"tier": "Standard"

},

"kind": "StorageV2",

"copy": {

"name": "storageCopy",

"count": "[variables('numberOfStorageAccounts')]"

}

}

],

"parameters": {

"location": {

"type": "string",

"metadata": {

"description": "Location for all resources."

}

}

}

}

Virtual Machine Extensions:

* Automate the repetitive work of post-deployment configuration and tasks on Azure VMs.
* Software installation, anti-virus protection, or a configuration script inside
* Managed with Azure CLI, PowerShell, Azure Resource Manager templates, and the Azure portal.
* Bundled with a new VM deployment or run against any existing system.

A screenshot of a computer

Description automatically generated

To install extension >

* go to resource : left side pane > settings > extensions + application
* Or while creating the VM, in advanced tab > select extension to install
* If required extension is not available, you can write a custom extension script.
* The Custom Script Extension downloads and runs scripts on Azure virtual machines (VMs).
* This extension is useful for post-deployment configuration, software installation, or any other configuration or management task. Example: Install IIS on Windows server, or Apache on a Linux server with a web app
* You can download scripts from Azure Storage or GitHub, or provide them to the Azure portal at extension runtime.
* You can add extensions in the new or existing instance as well
* Can integrates with Azure Resource Manager templates.
* Script only runs once, schedule a task to run again
* Removing the extension does not undo what the script did
* You can also run it by using the Azure CLI, PowerShell, the Azure portal, or the Azure Virtual Machines REST API.
* Considerations
  + Time out. Custom Script extensions have 90 minutes to run.
  + Dependencies. If your extension requires networking or storage access, make sure that content is available.
  + Failure events. Be sure to account for any errors that might occur when running your script. For example, running out of disk space, or security and access restrictions. What will the script do if there is an error?
  + Sensitive data. Your extension may need sensitive information such as credentials, storage account names, and storage account access keys. How will you protect/encrypt this information?
  + Reboot: It's best to avoid including reboots in the script because the extension will stop working after the reboot. As a result, if you have any other commands that require the extension to run after the reboot, they will not run. If you need rebook, look for other solutions like Desired state configuration, chef or puppet
* Custom scripts can be as per the OS. If its windows, it should be in PowerShell. If its in Linux, it should be shell script. Basically, you will write scripts to install or execute the script in the server once it is created.

Azure compute gallery :

* Image: This is a copy of the full VM (include OS and data disks)
* After you build and customize a virtual machine, you can save the new image as a set of VHDs.
* You can put this new image in to Azure compute gallery.
* Two types of Images
  + Specialized VM images: copy of a live virtual machine
  + Copy of the configured operating system, software, user accounts, databases, connection information, and other data for your system.
  + Use as a backup of your system at a particular point in time, you can restore your virtual machine from this image a
  + New VM created out of image will have same host name, user accounts, and other settings
* Generalized VM Images: No information retained.
  + Original VM is unusable after you perform the process
  + Tools for preparing a virtual machine for generalization –For Windows, use the Microsoft System Preparation (Sysprep) tool.
  + For Linux, use the Windows Azure Linux Agent (waagent) tool.

Azure Bicep

Azure Bicep: Streamlining Azure Resource Deployment with Declarative Syntax

Introduction:

* Azure Bicep is a specialized Domain-Specific Language (DSL) designed to simplify the deployment of Azure resources.
* It employs a declarative syntax, focusing on expressing the desired state of resources rather than the deployment process.

Key Features:

1. Declarative Approach:
   * Bicep embraces a declarative paradigm where developers define the intended configuration of resources.
   * Unlike imperative approaches, Bicep doesn't dictate the deployment steps; it handles the "what" while leaving the "how" to Azure.
2. Enhanced Readability:
   * Bicep's syntax is concise, clear, and more human-friendly compared to traditional Azure Resource Manager (ARM) templates.
   * This readability facilitates collaboration among developers, operators, and stakeholders.
3. Type Safety:
   * Bicep enforces strong typing, ensuring that data types match expectations.
   * This feature prevents common errors related to data type mismatches, improving code reliability.
4. Modularity:
   * Bicep promotes modularization through the use of reusable modules.
   * Developers can create smaller, self-contained modules and compose them to build complex resource configurations.
5. Compilation Process:
   * Bicep files are compiled into ARM JSON templates, which are then used for resource deployment.
   * This compilation maintains compatibility with existing deployment practices.

Workflow:

1. Install Bicep CLI:
   * Begin by installing the Bicep CLI, which is essential for compiling Bicep files into ARM templates.
2. Author Bicep Files:
   * Create .bicep files using the streamlined Bicep syntax to define Azure resources and configurations.
3. Compile Bicep Files:
   * Use the Bicep CLI to compile .bicep files into ARM JSON templates, ready for deployment.
4. Deploy ARM Templates:
   * Deploy the generated ARM templates using familiar tools like Azure CLI or Azure PowerShell.

Advantages:

* Simplicity: Bicep simplifies resource deployment through its intuitive syntax.
* Efficiency: Declarative nature eliminates the need for defining deployment steps, saving time.
* Modularity: Creating reusable modules enhances code organization and maintainability.
* Type Safety: Strong typing minimizes errors and enhances code robustness.
* Readability: Clear syntax improves code comprehension and teamwork.
* Compatibility: Bicep-generated ARM templates fit seamlessly into existing Azure deployment workflows.

Example

A simple example of a Bicep file that defines an Azure Storage Account:

1. bicepCopy code
2. param storageAccountName string
3. param location string = resourceGroup().location
5. resource storageAccount 'Microsoft.Storage/storageAccounts@2021-04-01' = {
6. name: storageAccountName
7. location: location
8. sku: {
9. name: 'Standard\_LRS'
10. }
11. kind: 'StorageV2'
12. properties: {
13. accessTier: 'Hot'
14. enableHttpsTrafficOnly: true
15. }
16. }

In this example:

* The param keyword defines parameters that can be passed when deploying the template. storageAccountName and location are parameters that can be provided during deployment.
* The resource keyword is used to define an Azure resource. The resource type is Microsoft.Storage/storageAccounts and the API version is specified as 2021-04-01.
* Inside the resource block, properties such as name, location, sku, kind, and properties are defined. These properties configure the storage account.
* The resourceGroup().location function is used to get the location of the resource group where the template will be deployed.

This example showcases the basic structure of a Bicep file and how it defines an Azure resource. You can extend and modify this template to create more complex deployments with multiple resources and configurations. Remember that the Bicep file can be compiled into an ARM JSON template, which is then used to deploy resources in Azure.

Conclusion:

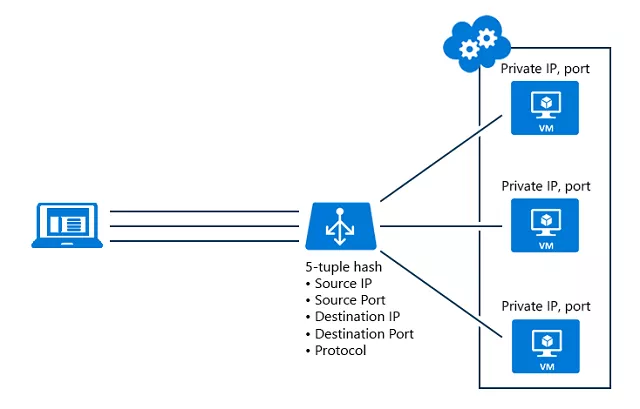
* Azure Bicep streamlines Azure resource deployment through its user-friendly, declarative syntax.
* By focusing on defining desired configurations, Bicep simplifies the management of Azure resources while maintaining compatibility with established deployment practices.

**Load Balancer :**

* Load balancer evenly distribute the incoming traffic to backend resources.
* LB provides high availability and scalable performance.

Load Balancer components:

* Frontend configuration –IP address of your Azure Load Balancer. It's the point of contact for clients.
* Backend pool –Group of virtual machines that is serving incoming traffic.
  + Adding or removing VMs from the backend pool reconfigures the load balancer without additional operations.
* Load-balancing rules determine how traffic is distributed to the backend pool.
* Health probes ensure the resources in the backend are healthy.
  + Health probe dynamically adds or removes VMs from the load balancer rotation based on their response to health checks.
  + When a probe fails to respond, the load balancer stops sending new connections to the unhealthy instances.



A blue and white symbol with white text

Description automatically generated

Types of load balancer :

* Public
* Internal

A diagram of a computer network

Description automatically generated

* Two types of load balancers: Public and Internal
* A public load balancer maps the public IP address and port number of incoming traffic to the private IP address and port number of the VM.
* An internal load balancer directs traffic to resources that are inside a virtual network or that use a VPN to access Azure infrastructure.

Load balancer SKUs (stock keeping unit):

* when you create LB, you can choose basic or standard.

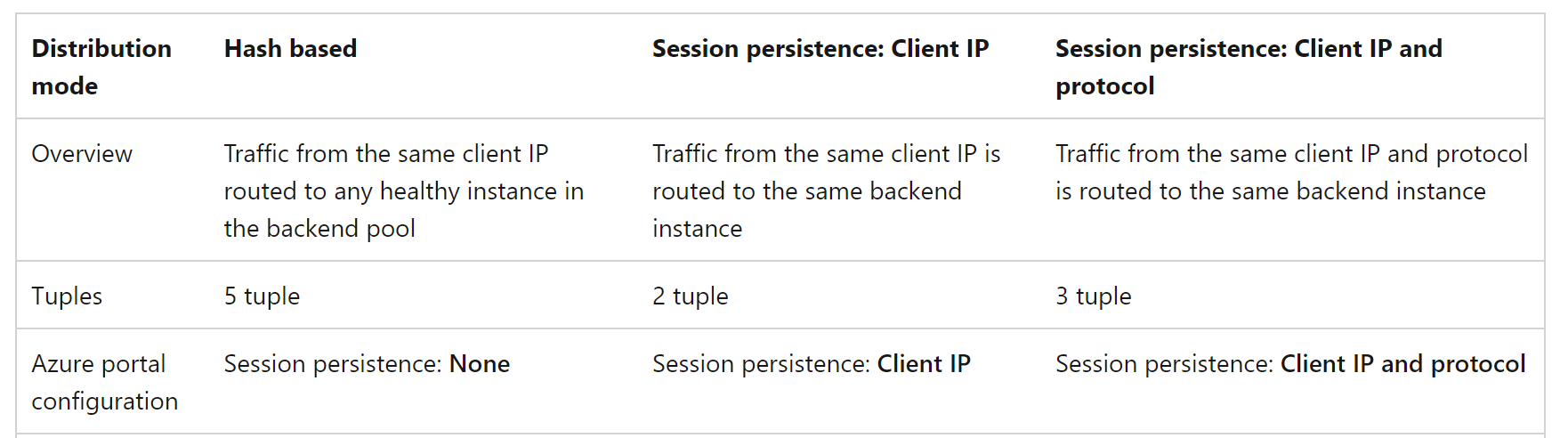
|  |  |  |
| --- | --- | --- |
|  | Basic | Standard |
| Cost | Free | Cost per hour |
| Backend pool Size | Supports up to 300 instances | Support up to 1000 instances |
| Backend pool endpoints | Only VMs in single availability set or scale set | Any VM in single Vnet. |
| VM Public IP type | Basic SKU Public IP or No Public IP | Standard SKU Public IP or No Public IP |
| NSG Rules | Open by default. Network security group optional | Closed to inbound flows unless allowed by a network security group. |
| Health Probe | TCP, HTTP | TCP, HTTP, HTTPS |
| Availability Zones | No Support | Support |
| SLA | Not available | 99.99% |
| Use | Dev/Test | Production |

Load Balancer distribution modes:

A diagram of a private network

Description automatically generatedA diagram of a private network

Description automatically generated

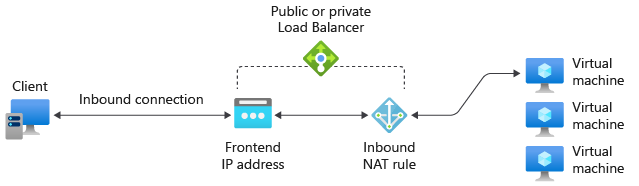


Load Balancer-NAT Rules:

* An inbound NAT rule forwards incoming traffic sent to frontend IP address and port combination.
* The traffic is sent to a specific virtual machine or instance in the backend pool.
* You can create multiple frontend ip of load balancer and while selecting nat rule, you can chose any of the load balancer frontend ip.

A diagram of a load balancer

Description automatically generated



Azure Load Balancer provides inbound and outbound traffic load balancing for Azure virtual machines. Network Address Translation (NAT) rules within Azure Load Balancer enable you to map a range of ports on the public IP address to a set of ports on an internal IP address and port.

Inbound NAT Rules:

* NAT rules allow you to map a public port on the Load Balancer to a specific port on an internal VM.

- Port Forwarding: You can use inbound NAT rules to forward traffic from a specific port on the public IP address to a specific port on a virtual machine in your backend pool.

- Multi-Port Load Balancing: If you have multiple virtual machines in a backend pool, you can distribute traffic across them based on the port number.

\*\*Example:\*\*

- Suppose you have a web server on port 80 of a virtual machine in the backend pool. You can create an inbound NAT rule to forward external traffic on port 8080 to port 80 of that specific VM.

Outbound NAT Rules:

Outbound NAT rules allow virtual machines in a backend pool to initiate outbound connections through the public IP address of the Load Balancer.

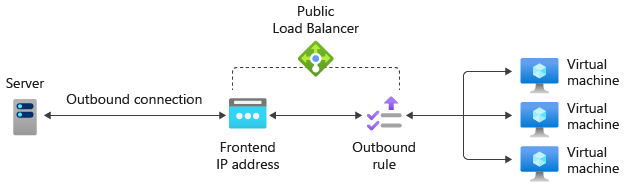
\*\*Use Cases:\*\*

- \*\*Outbound Internet Access:\*\* When VMs in the backend pool need to access the internet, outbound NAT rules can be used to map their outbound traffic to the public IP address.

- \*\*Source IP Address Preservation:\*\* Outbound NAT rules also preserve the source IP address of the VM, helping external services see the actual source of the request.

\*\*Example:\*\*

- If your VMs in the backend pool need to access external services on the internet, you can use outbound NAT rules to map their internal IP addresses to the public IP address of the Load Balancer.



### Port Ranges:

- Both inbound and outbound NAT rules support port ranges. This flexibility allows you to define multiple rules that map different ports or ranges of ports to different backend VMs.

### Configuration:

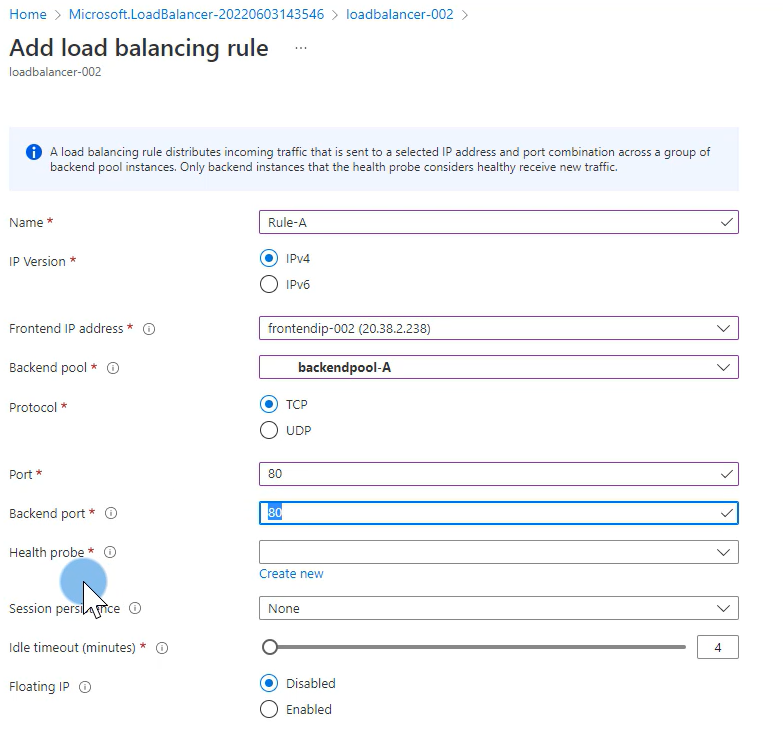
- You can configure NAT rules using the Azure portal, Azure CLI, or PowerShell. In azure portal, > load balancer>left pane> NAT

- When you create an Azure Load Balancer, it is essential to configure NAT rules according to your specific application requirements for handling inbound and outbound traffic effectively.

By leveraging NAT rules in Azure Load Balancer, you can optimize and control how traffic is directed to and from your virtual machines in Azure.

Load balancer using scale set :

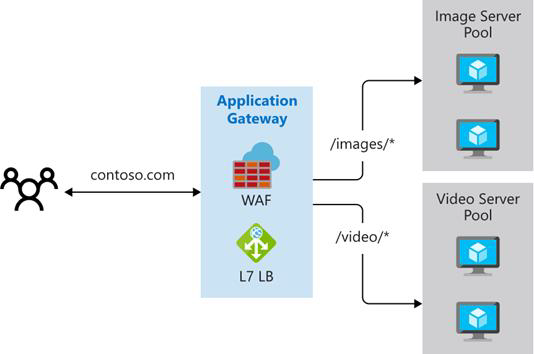
* Create vm scale set
* Create load balancer > go to backend pool
* Backend pool A > virtual network > mention the virtual network for vm scale set rg virtual network
* Backend pool A > Associate backend pool a to > virtual machine scale set
* Go to load balancer > health probe > protocol : tcp, port :80 , interval5 secos, un healthy threshold 2 consecutive failure
* Go to load balancing rule > apply the rule



Application Gateway

* Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications.
* OSI layer-7 (Application layer) load balancer.
  + HTTP and HTTPS traffic only
* Traditional Load Balancer –> Layer 4 –TCP and UDP ( source/destination IP/port and protocol)
* Routing rules based on HTTP request parameters:
  + URL path (web address)
  + Host headers (request data)
  + Can be configured public facing, internal or combination of both.
  + You can only assign Static IP address to application gateway.
  + Needs a separate empty subnet to install application gateway components

URL Path-based routing:



Multiple site routing:

A diagram of a computer application gateway

Description automatically generated

Web application firewall:

* Checks each request for many common threats
* Based on the Open Web Application Security Project (OWASP)
  + SQL-injection, Cross-site scripting, Command injection, HTTP request smuggling, HTTP response splitting, Remote file inclusion, Bots, crawlers, and scanners, and HTTP protocol violations and anomalies.

A diagram of a computer application

Description automatically generated

Application Gateway Components :

A diagram of a computer network

Description automatically generated

Frontend FQDN : in application gateway you will have only one public ip address/private ip associated with.

HTTP/HTTPS listen: multiple listener can be attached to application gate way. It listen the traffic from frontend ip.

* Basic listen
* Multi site listener

Rule : rule n how the traffic will be managed

* Basic rule
* Path based rule