Microsoft Azure: AZ-104 Certification

Cheat Sheet

Quick Bytes for you before the exam!

The information provided in the Cheat sheet is for educational purposes only; created in our efforts to help aspirants prepare for the Microsoft Azure Exam AZ-104 certification. Though references have been taken from Azure documentation, it's not intended as a substitute for the official docs. The document can be reused, reproduced, and printed in any form; ensure that appropriate sources are credited and required permissions are received.

Are you Ready for Microsoft Azure Administrator Associate "AZ-104" Certification?

Self-assess yourself with

Whizlabs FREE TEST

750+ Hands-on-Labs

Hands-on Labs - AWS, GCP, Azure (Whizlabs)

Cloud Sandbox environments

Cloud Sandbox - AWS, Azure, GCP & Power BI

AZ-104 Index	SNO
Azure Basics	
• Azure Portal	3
• Azure CLI	6
• Azure Powershell	7
• Azure Resource Manager	8
• Azure Pricing	9
• Azure Well-Architected Framework	10
Manage Azure identities and governance	
• Microsoft Entra	13
• Microsoft Entra ID	14
• Who uses Microsoft Entra ID?	15
• Managing Microsoft Entra Users and Groups	18
• Microsoft Entra ID Access Management and Control	23
• Managing Azure subscriptions and governance	28
Implement and manage storage	
• Azure Storage Introduction	38
• Azure Storage - Services, Types, and Benefits	38
• Configure access to storage	38
• Manage data in Azure storage accounts	47
• Configure Azure Files and Azure Blob Storage	52
• Azure Storage Firewalls and Virtual Networks	56
Deploy and manage Azure Compute Resources	
• Automate deployment of resources by using templates	58
• Create and configure Azure Virtual Machines	67
• Create and Configure Containers	78
• Create and configure an Azure App Service	82
• Application Service Environments	87
Monitor and Maintain Azure Resources	
• Introduction to Azure Monitor	90
• Monitor resources by using Azure Monitor	90
• Implement backup and recovery	95

--Back to Index--

Implement and Manage Virtual Networking	
• Introduction to Azure Virtual Network(Azure VNet)	100
• Benefits and Components of Virtual Netwroks	100
• Configure virtual networks	101
• Configure secure access to virtual networks	107
• Configure load balancing	114
• Monitor resources by using Azure Monitor	120
Extra Learning (Not part of the Syllabus)	
• Azure Storage Services	123
• Azure Key Vault Service	128
• Azure Container Registry	132
• Azure Service Health	135
• Azure Firewall	137
• Azure Traffic Manager	141
• Azure Express Route	143
• Azure VPN Gateway	145
• Azure Content Delivery Network	150
• Azure DDoS Protection	152
• Azure AKS	157

AZ-104 Exam Format and Information → 165

Azure Basics

Administration Tools

Azure provides 3 administration tools to choose from

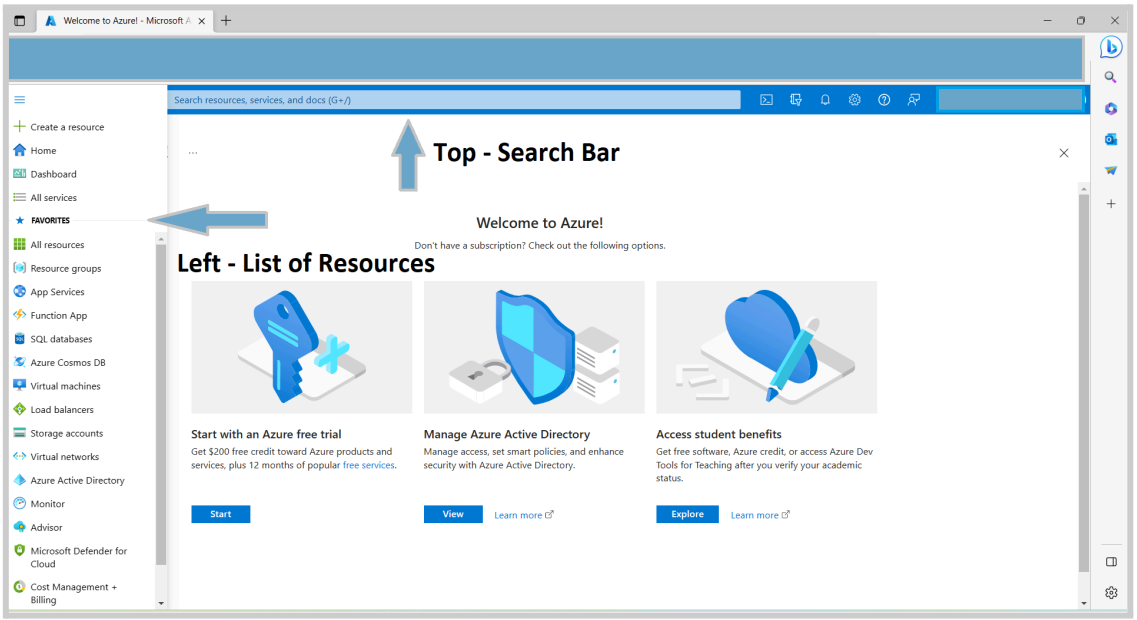
1. The Azure Portal
2. The Azure CLI
3. Azure PowerShell

Azure Portal

- We can use the **Azure GUI portal website (portal.azure.com)** to create, configure, and alter our Azure subscription resources.
- We can locate the resource needed and execute any changes. We have wizards and tooltips to guide you through various administrative tasks.
- Please note that we cannot use the portal to perform repetitive tasks like creating 12 VMs etc.
- We need to use other tools to avoid errors, and it will also be a time-consuming process to do on the portal.

The Azure portal can be divided into 3 sections.

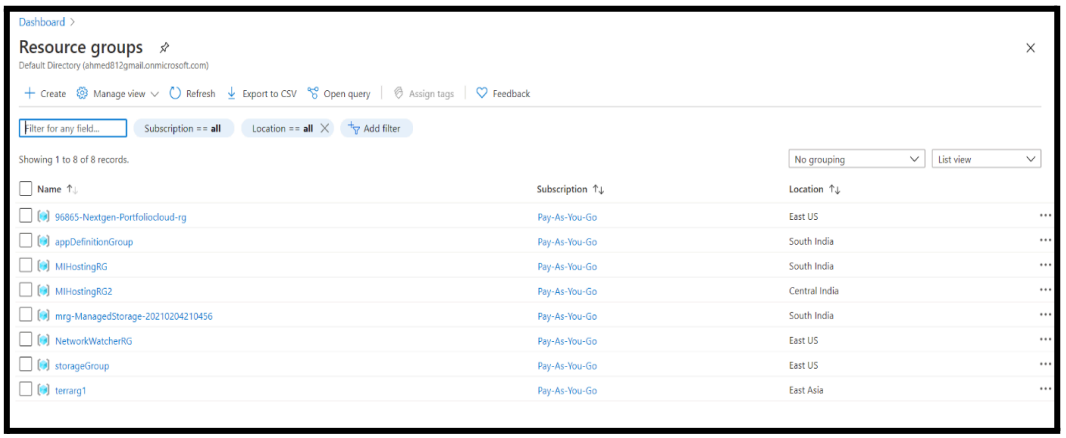
1. **Left** — A list of resources and services to create and manage your Azure environment.
2. **Center** — A dashboard that you can tailor to meet your (Public or Private dashboards) needs.
3. **Top** — A search bar to quickly find resources and services, a notification icon, access to a web-based command line, and more.



(Source: Microsoft Docs)

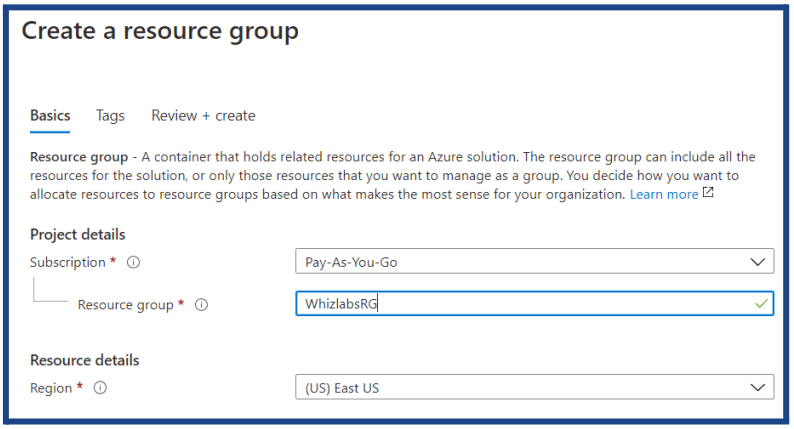
- Let's try to create a resource and see how to use the Portal. For example, let us create a resource group called **Whizlabs**.

- Click on the Burger menu on the left top and select Resource group and click on it. You will get a new Panel.



(Source: Microsoft Documentation)

- Click on the **+Create** icon. On the new Panel, add the name of the resource group and choose the desired location.



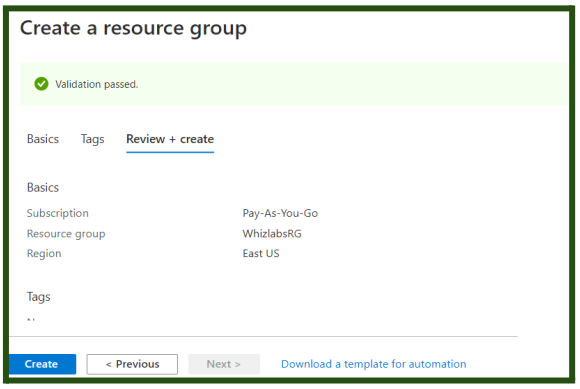
(Source: Microsoft Documentation)

- Click Next, and you will get a new panel to add Tags. Tags are helpful for accounting and segregation but are not mandatory.

(Source: Microsoft Documentation)

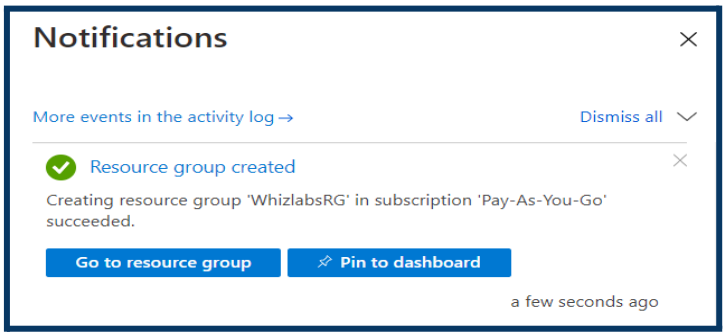
- Click NEXT, and at this point, Azure will validate all the options chosen.
- If there is any error, it will put a red dot on the tab where the error occurred, and you will need to go back to the tab and fix it before proceeding.
- If validation is passed, you would see the Validation passed with a green tick message. At this point, you can click CREATE, and the resource will be created.

- You can also click on “Download a template for automation” and download the template and save it to the library additionally for future use.



(Source: Microsoft Documentation)

You will get a notification when the resource is created. You can also click the bell icon on Top Right to view the notification.



(Source: Microsoft Documentation)

If we go back to the Resource Groups, we can see this new resource group. This is a simple example of the usage of the Portal. We can use the portal for lots of activities.

We can use the Azure Portal for

- Creating/ Modifying/ Deleting resources
- Billing and accounting
- Help and Support – Contact Microsoft
- Online Help
- Health and Service Dashboards
- Microsoft Defender for Cloud
- Access Microsoft Entra and create applications
- Azure Monitor
- Access Documentation
- Azure Marketplace for third party products and solutions
- Access Cloudshell (On top right)

Azure CLI

Azure CLI means → Azure Command Line Interface. It's a cross-platform command-line program to connect and execute administrative commands on Azure resources.

Sample command:
az VM create --resource-group WLRG --name WLVM1 --image UbuntuLTS

Azure CLI can be accessed inside a browser via Cloud Shell or with a local install on any OS like Windows/Linux or MacOS and Docker. It can also work with multiple clouds.

Let's see an example.

First, we invoke the MSI installer either in the command line or by downloading.
Here is the command line below:

```
Invoke-WebRequest -Uri https://aka.ms/installazurecliwindows -OutFile .\AzureCLI.msi; Start-Process msiexec.exe -Wait -ArgumentList '/I AzureCLI.msi /quiet'; rm .\AzureCLI.msi
```

Then we sign in with the login command **az login.**

A new browser page will open (https://aka.ms/devicelogin), and we enter the authorization code displayed on the terminal.

Some of the common commands are as follows:

S No	Azure CLI command group	Resource Type
1	az group	Resource group
2	az keyvault	Key Vault
3	az SQL server	SQL databases
4	az storage account	Storage accounts
5	az vm	Virtual machines
6	az webapp	Web applications

Let's take Storage accounts as an example and work with Azure CLI

Step 1:
Create a resource group for Storage accounts

```
az group create --name StorageRG --location westus
```

Step 2:
Create a Storage account

```
az storage account create --name WLblobSA123 --resource-group storageRG
--location westus
--sku Standard_RAGRS --kind StorageV2
```

Step 3:
Finally delete to clean up the test

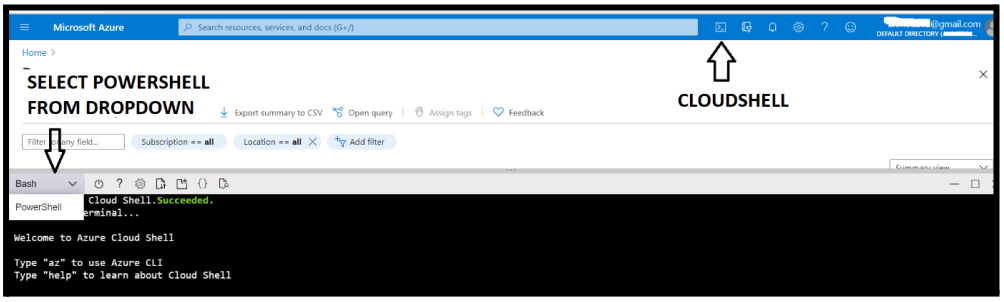
```
az storage account delete --name WLblobSA123--resource-group storageRG
```

Azure Powershell

- It's a module that allows us to connect to Azure subscriptions and manage resources.
- It uses AzureRM command modules, and it has now added Az command modules as well.
- If we used the *New-AzureRmVM* command to create a VM via the AzureRM Module, we would change to the *New-AzVM* command to create a VM via the Az Modules.

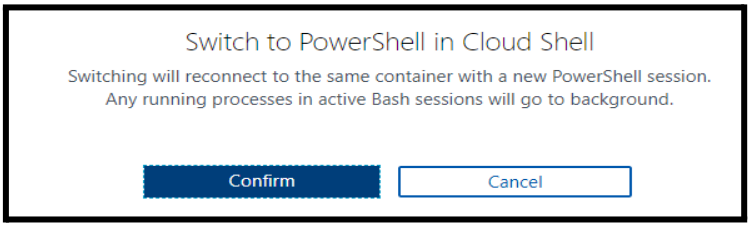
How to use Powershell in Azure Portal?

- First, click on the cloud shell icon on the top right. If you are doing this for the first time, you will be prompted to create a Storage account to host the cloud shell files.
- You can accept a default storage account and file names or choose your own.
- By default, Cloudshell launches in **BASH MODE**.
- You need to choose Powershell from the dropdown and you will be prompted for a confirmation.



(Source: Microsoft Documentation)

- Once you hit on the confirm button, you will get the Powershell command line to execute Powershell commands.



(Source: Microsoft Documentation)

How to use Powershell in your local installation?

- Windows OS comes with Powershell installed. You can select Windows Powershell and hit enter once the Powershell window is launched; type az login.
- A new browser will be launched to select an already logged-in session/ log in to new session.
- After getting a successfully logged-in message, you can close the browser and go back to your Powershell screen and continue working with Powershell commands.

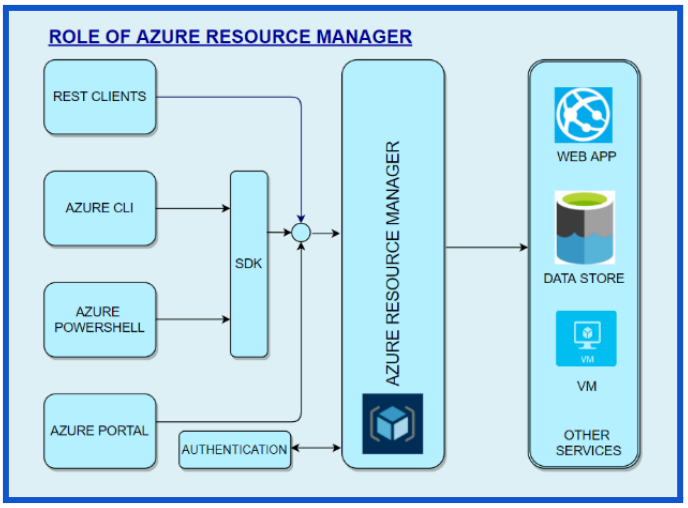
Working with Powershell:

For example: To create a VM, we launch Powershell either inside a browser or by installing locally on any OS and then run the **New-AzVM** command that creates a virtual machine in our subscription as follows:

```
New-AzVm -ResourceGroupName "WLRG" -Name "WLVM1" -Image "UbuntuLTS" ...
```

Azure Resource Manager

- Azure Resource Manager provides a management layer to ***create, update, and delete*** resources in your Azure account.
- We use management features, like access ***control, locks, and tags, to secure and organize your resources after deployment.***
- When a user sends a request from any of the tools, APIs, or SDKs, the Resource Manager receives the request and **authenticates/authorizes** it.
- Then it sends to Azure services to take action. Since it acts as a central point, it leads to consistent results.



(Source: Microsoft Documentation)

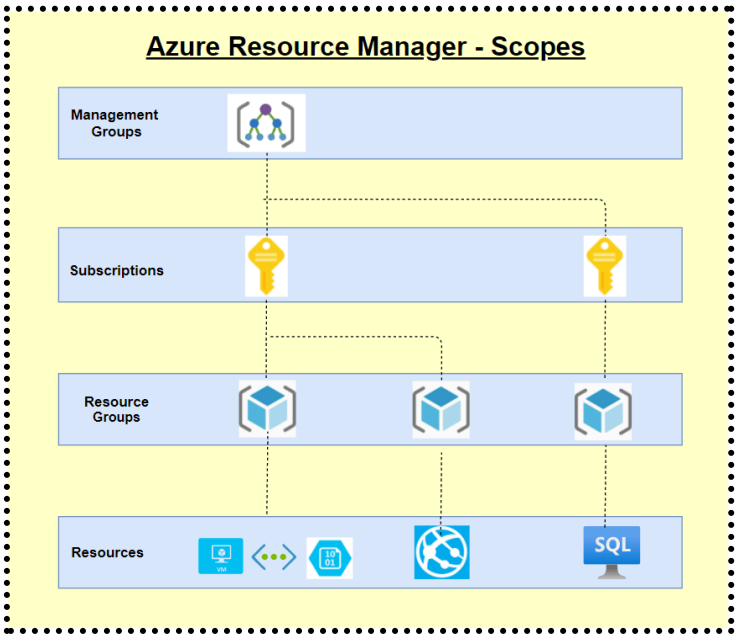
Benefits of Resource Manager:

- Declarative templates so we don't have to worry about the current state.
- Allows group deployments
- Define dependencies so the correct order of deployment is done.
- Apply tags to organize resources logically
- Allows for redeployment and confidence that the same results will be achieved
- Applies access control via RBAC natively.

When we deploy, they are done at 4 levels.

1. **Management Groups** – At this level, we can combine multiple subscriptions to apply changes at an Organizational level. We can connect Organizations with a hierarchy where there is one management group at the root level. This is called Nesting.
2. **Subscriptions** – It's a logical container used to provision resources. We will be billed at the subscription level. We can have multiple subscriptions.
3. **Resource Groups** – We can create multiple resources in a resource group. We can logically group resources at a resource group level. We can delete an entire resource group, and all resources will be deleted within the resource group. We can even move a whole resource group with all objects within it.
4. **Resource** – This is the lowest manageable item in Azure resources. Examples of Azure resources are *Virtual machines, storage accounts, web apps, databases, virtual networks,*

and tags. Resource groups, subscriptions, and management groups are also examples of resources.



(Source: Microsoft Documentation)

Azure Pricing

Azure is one of the market leaders in Cloud services and has some of SQL and Windows's best pricing. It can leverage several features to save costs, and Azure provides several tools that can help calculate costs and cost-effectively plan our infrastructure and service.

Some of the available tools are:

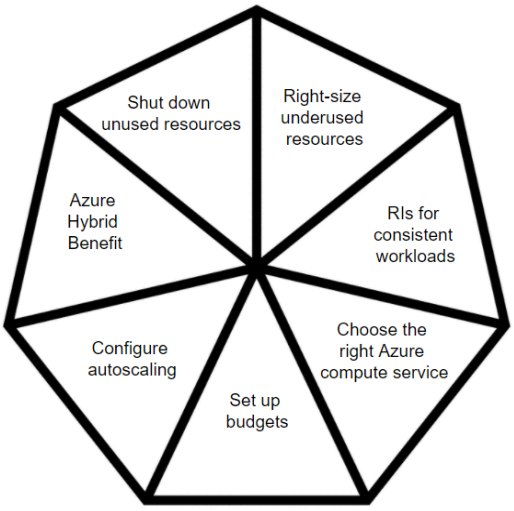
- ➔ **Azure Pricing Calculator**
- ➔ **Cost Management Center**
- ➔ **Migration planning – Estimation, Workload, and right-sizing**
- ➔ **Billing Data API & Advisor**
- ➔ **DB & Cosmos DB Capacity calculator**

Some of the features that we can leverage to save costs are as follows:

- **Azure Hybrid Benefit –** We can use our existing SQL and Windows licenses to save on costs.
- **Spot Virtual machines** - This feature allows us to take advantage of the unused CPU at a significantly lower cost at almost 90% savings.
- **Reservations** - We can commit to 1 or 3 years & choose to pay upfront/monthly to buy RIs.
- **Azure Dev/test pricing** – For development environments, we can get special discounted rates

Ways to optimize Cost

Please see the self-explanatory chart below for ways to optimize cost

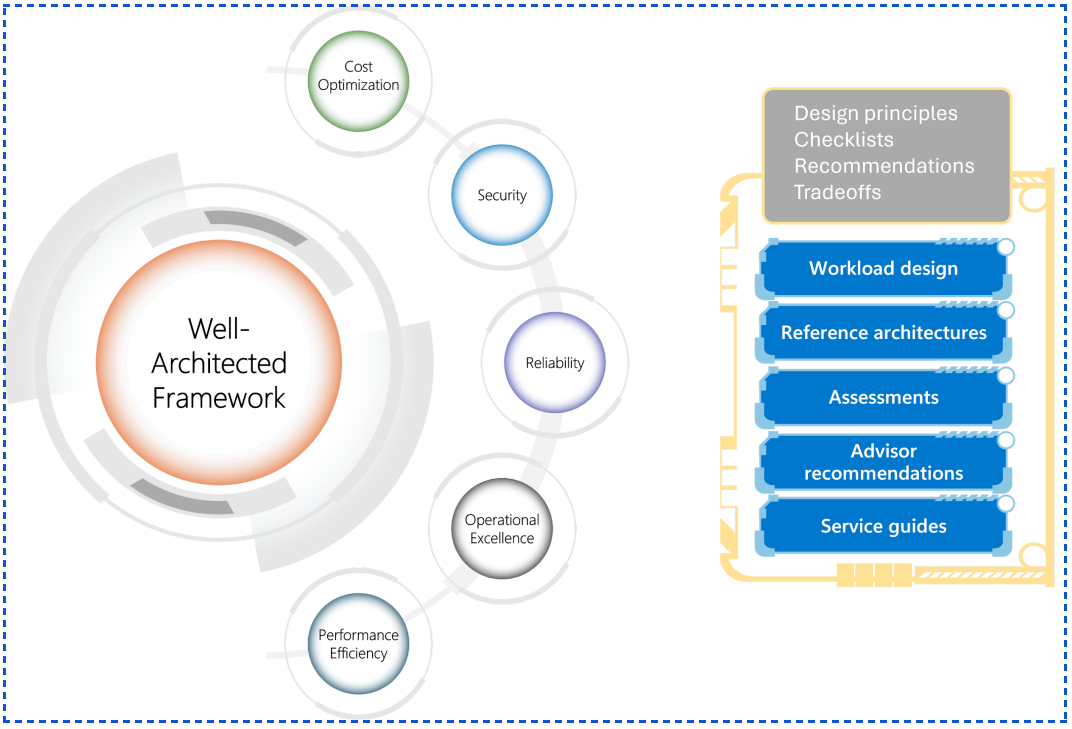


FAQs:

- Are there any other ways to save costs?
  - **EA – Enterprise Agreements** – With this, we can get good pricing offers from Azure.
  - **Price Match with AWS** – This might not be known to all, but we can ask MS to do a price match.

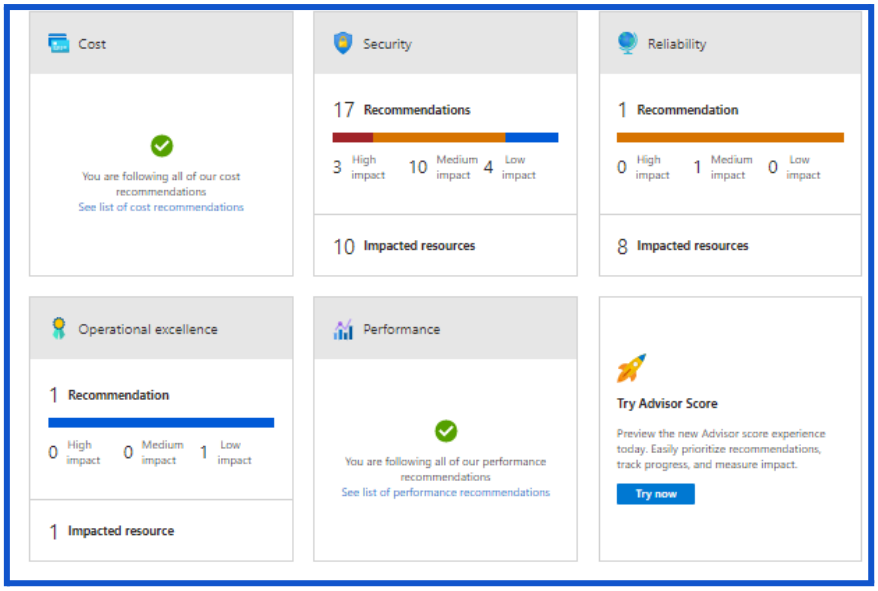
Azure Well-Architected Framework

Azure has 5 pillars called the Azure well-architected framework which provides best practices to help build and deliver great solutions.



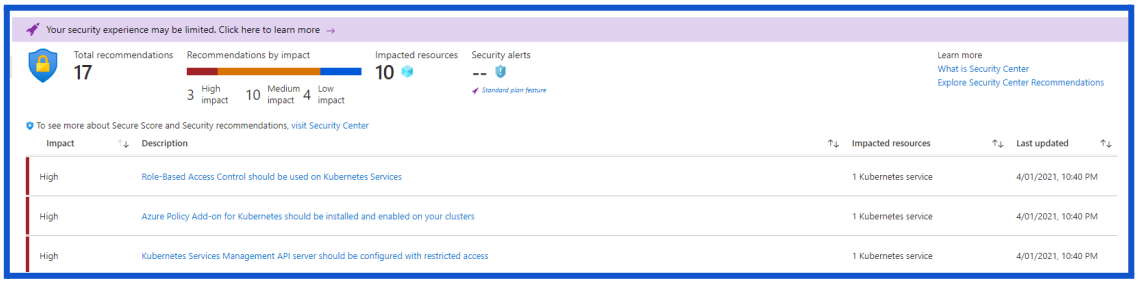
Source: [Azure Well-Architected Framework]

- On each of the 5 pillars, we will be given recommendations to optimize. Please see below.



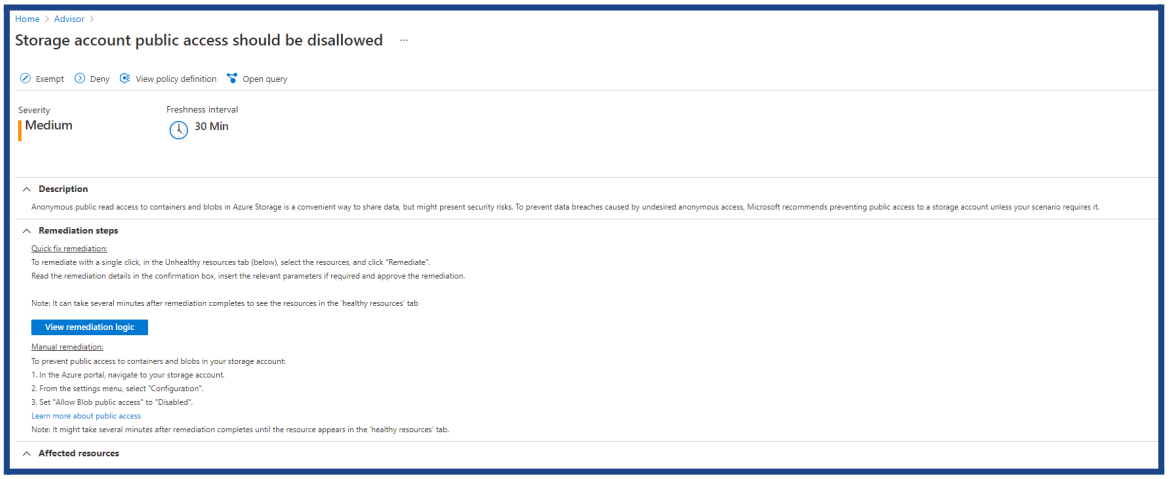
(Source: Microsoft Documentation)

- If we click on each of these recommendations, we can see what the recommendations are.



(Source: Microsoft Documentation)

- If we further click on each of the line items, we will give the list of resources that are not compliant and will provide manual and in some cases remediation action which can be deployed directly.



(Source: Microsoft Documentation)

- You can also note from the above that these recommendations are set up with the help of Azure policies.

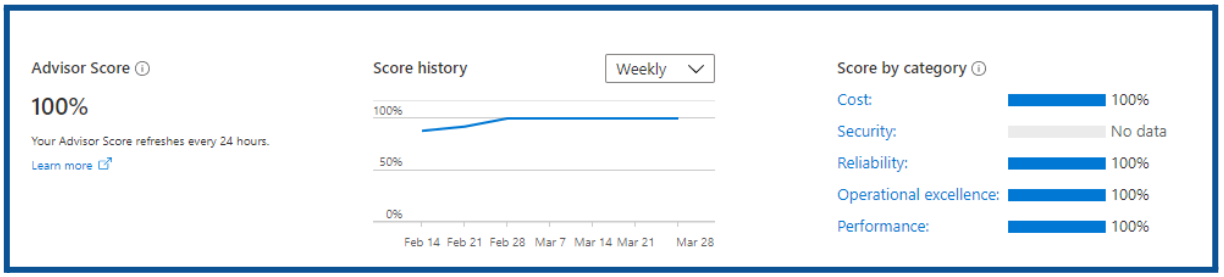
- We can see the Policy definition and we can exempt the policy itself from being flagged as non-compliant.
- We can enable the deny action also in which case the resource will be prevented from being created.
- Here we have the policy which is audit and hence the resource is created and marked as non-compliant.

Sample remediation code:

```
{
  "properties": {
    "allowBlobPublicAccess": false
  }
}
```

We can download these recommendations as a CSV or PDF file.

Microsoft Entra IDvisor also has 2 features in preview. One feature is alerts which are yet to be generally available (GA). The other feature is the Advisor score which gives us on a percentage basis if we are following best practices.



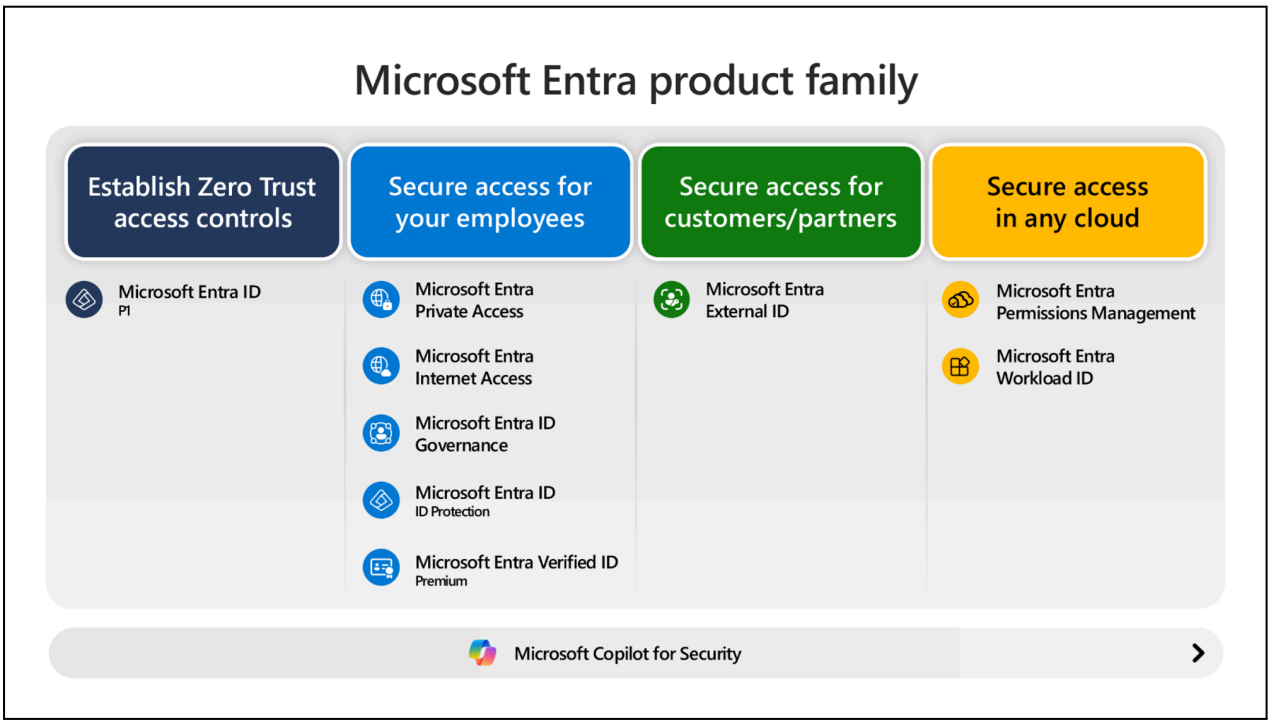
(Source: Microsoft Documentation)

Manage Azure Identities and Governance

Microsoft Entra

Microsoft Entra is a family that is mainly related to Identity and Access Management.
It enables organizations to implement a zero-trust security strategy and create monitors that verify identities, validate access conditions, check permissions, encrypt connection channels, and monitor for compromise.

Microsoft Entra product family covers four maturity stages of secure end-to-end access for any trusted identity. These steps include establishing zero trust access controls and securing access for employees, customers, partners and any cloud environment.



[Source: Microsoft Entra | Microsoft Learn]

[Source: Microsoft Entra - Secure Identities and Access | Microsoft Security]

Microsoft Entra ID - History

Microsoft Entra ID, formerly known as Azure Active Directory (Azure AD), is a cloud-based identity and access management service developed by Microsoft.

- Microsoft introduced **Active Directory** in the year 2000 which to this day is one of the best products from its stable.
- It was first launched on October 27, 2008 as Azure Active Directory. This service provides authentication and authorization for multiple Microsoft services, including Microsoft 365, Dynamics 365, and Azure, as well as third-party applications.
- Any Enterprise with Windows servers would be running the Domain Controllers in a **Domain/Tree/Forest** organization setup with multiple DCs playing different roles **(called FSMO – Flexible single master operation)** and in multiple locations for load balancing and reducing latency and increasing fault tolerance.
- Before this, Microsoft had NT4 where there was a single **PDC (Primary Domain Controller)** backed by a **BDC (Backup Domain Controller)** to provide Enterprise Identity Management.
- Windows 2000 and beyond uses the Active Directory and it uses **LDAP** (**Lightweight directory access Protocol)/Kerberos** for authentication. Here all resources like computers, printers, etc are all considered objects.
- This concept changed with Active Directory which like most cloud service providers also uses REST API in the background.
- Any service invoked on the Azure cloud is with REST APIs and this is the foundation for **Microsoft Entra ID**. Therefore, AD on the client premises and Microsoft Entra ID on the cloud will not work seamlessly.

Let's look deeper and compare the two and in that process understand better.

- **Communication**
- **Authentication**
- **Access Setup**
- **Network Organization**
- **Desktops**

Microsoft Entra ID

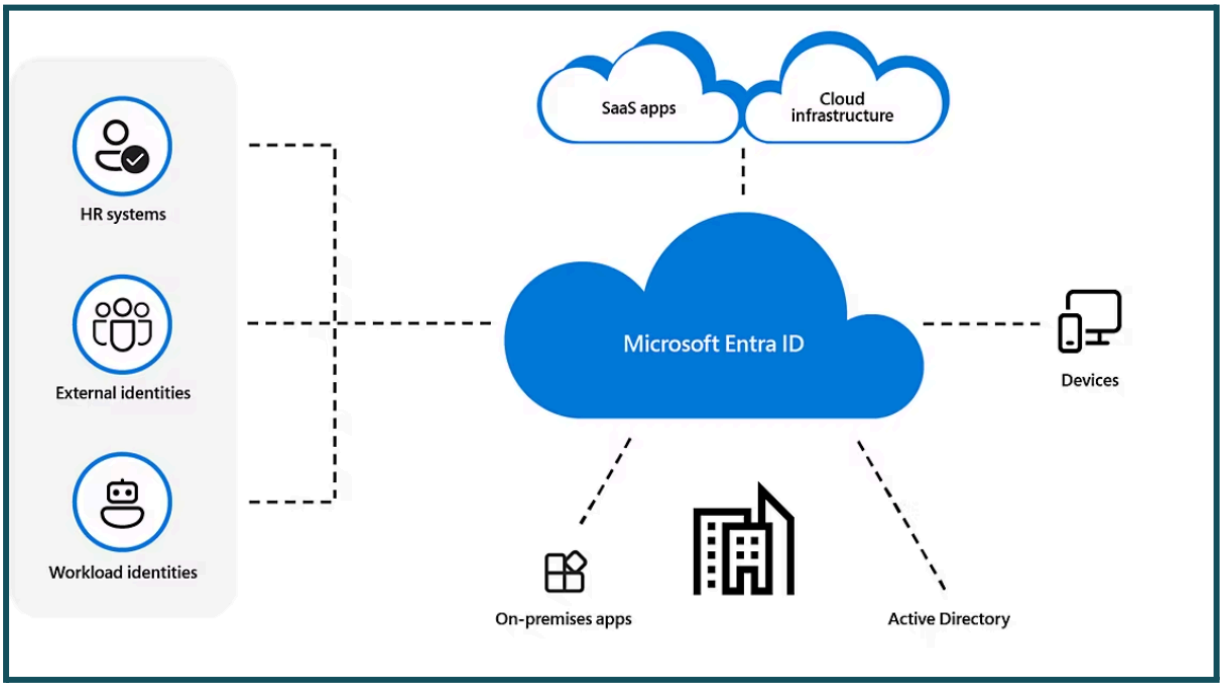
It's a **cloud-based identity and access management** service that your employees can use to access external resources. Example resources include Microsoft 365, Azure Portal, and thousands of other SaaS applications.

It's an **integrated cloud identity and access solution** and the market leader in managing directories, enabling access to applications, and protecting identities.

It helps to access internal resources such as apps on your corporate intranet and any cloud apps developed for your organization.

Who uses Microsoft Entra ID?

- **IT Admins** leverage Microsoft Entra ID to manage app access according to business needs.
- **App developers** can use Microsoft Entra ID as a standards-based authentication provider that helps them add single sign-on (SSO) to apps where the user works with existing credentials.
- **Microsoft 365, Office 365, Azure, or Dynamics CRM Online subscribers** already using a Microsoft Entra ID, every Microsoft 365, Office 365, Azure and Dynamics CRM Online tenant is automatically a Microsoft Entra tenant.



(Source: Microsoft Documentation)

- Microsoft Entra ID Free.
- Microsoft Entra ID P1
- Microsoft Entra ID P2
- Microsoft Entra External ID

Along with Microsoft Entra ID licenses, you can enhance your identity management features by opting for additional licenses for other Microsoft Entra products.

- Microsoft Entra ID Governance.
- Microsoft Entra Permissions Management.
- "Pay as you go" feature licenses.

Differences between Azure AD vs Microsoft Entra ID

1. Product Icons

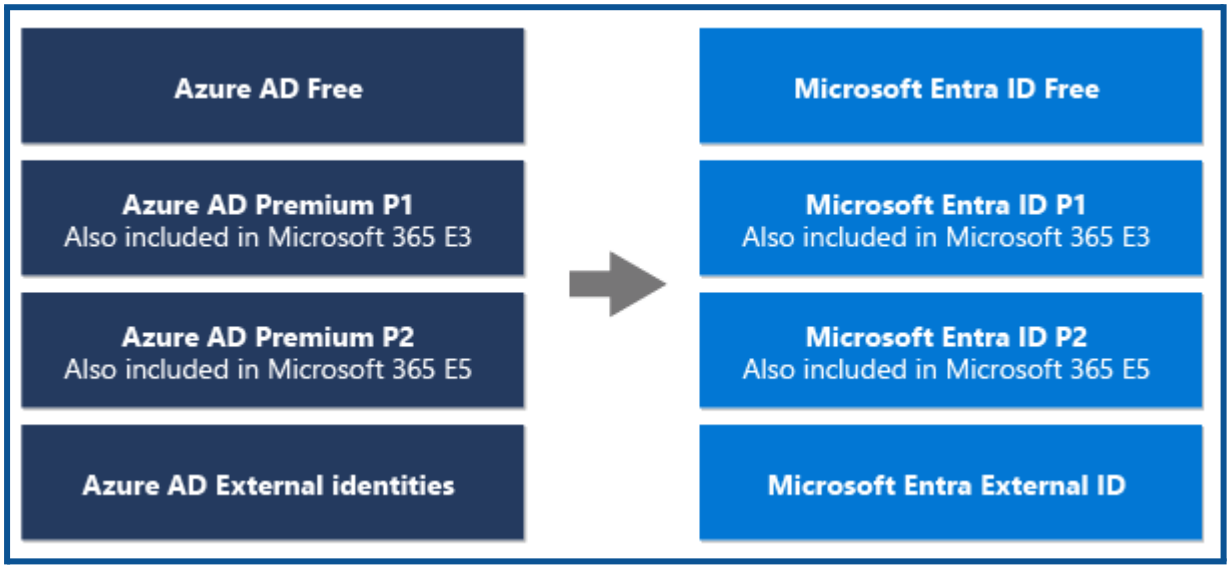
Azure AD Icon	Microsoft Entra ID Icon
	

2. Feature names

➔ "Azure AD Conditional Access" is now "Microsoft Entra Conditional Access"

➔ "Azure AD single sign-on" is now "Microsoft Entra single sign-on.

3. Service Plans



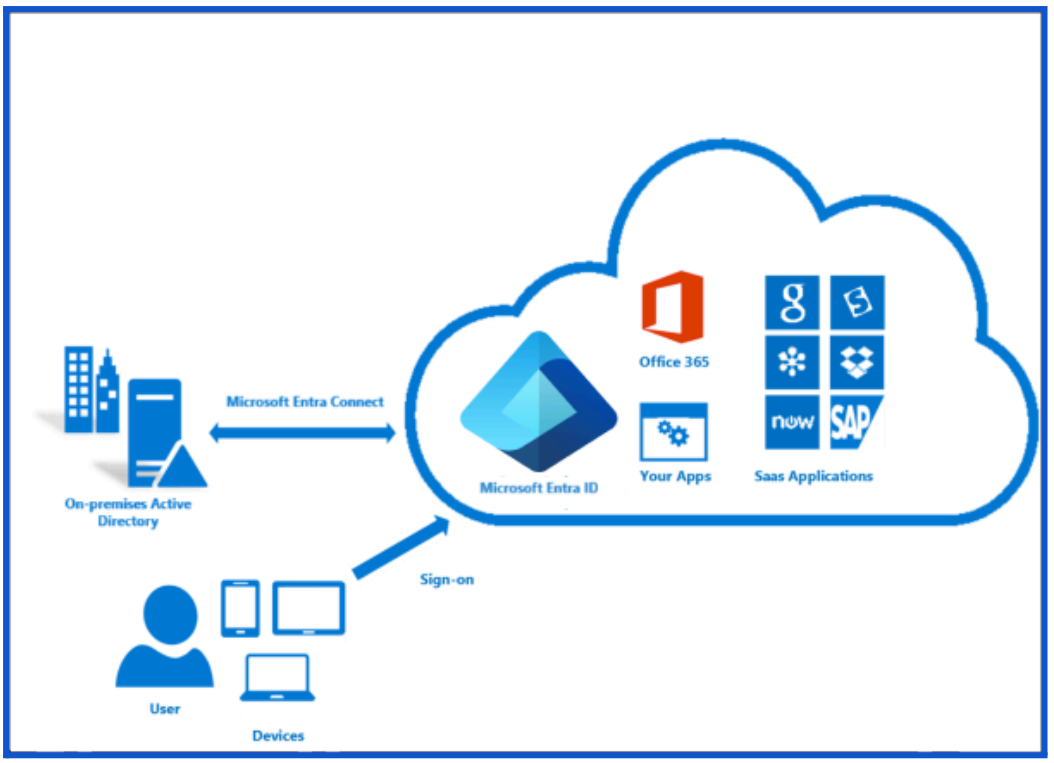
Source: [Azure Active Directory - Microsoft Entra | Microsoft Learn]

Microsoft Entra Connect

In situations where we want to enable a hybrid environment where we have both on-premises AD and Microsoft Entra ID, we should use Microsoft Entra Connect which synchronizes data between the two directories.

Microsoft Entra Connect will allow us to synchronize user accounts and passwords. There are several methods of synchronization.

- **Hash Synchronization** – Here only a hash of the password is stored on the cloud.
- **Pass-through authentication (PTA)** – Here the authentication is forwarded to the on-premises server
- **Federation** – Federation services provide authentication across several external identities in addition to providing on-prem access



(Source: Microsoft Documentation)

Entra Connect Features

Default Domain – The default domain is based on our email id. If our email id is whizlabs@gmail.com. Then our domain name will be whizlabsgmail.onmicrosoft.com. It is a combination of user and domain and then the addition of .onmicrosoft.com.

Usernames – Any user name we create will have the suffix of our domain name

Custom domain name – If we want to use our own company name, then we should create a custom domain (for example whizlabs.com) and then we can create a user smith@whizlabs.com

App registrations – We can register our applications here and grant access to application/users.

License Management – We can perform license Management here. We can track all acquired licenses and assigned licenses and make sure we don't overuse and pay heavy penalties

Enterprise Applications – we can see all the enterprise applications and assign them to our users. When a user logs in, he/she can see only the applications assigned to them.

Security – This is one of the key areas. Under security, we can see the following

- **Microsoft Entra Conditional Access –**We can add conditional access policies like restricting users from logging in from outside the office network or even outside the country
- **Microsoft Entra ID Protection –**We can assign user risk / sign-in risk and the system will dynamically assess risk and react like unusual geography of login
- **Identity Secure Score –** We are given a security score which tells us our overall security posture

- **Named locations –** If we readily identify safe locations like cities where headquarters and branch offices are located, we can create named locations and allow these under conditional access policies.
- **Authentication methods –** We can enable additional authentication methods like FIDO2 Security Key/ Microsoft Authenticator
- **Multi-Factor Authentication (MFA) –** We can configure MFA and add multi-factored authentication. Please note that this setting is outside of the Azure portal and a link will take out to the GUI. The sample screen looks like this.

Managing Microsoft Entra Users and Groups

Microsoft Entra provides robust tools for managing users and groups, which are essential for maintaining access, collaboration, and security within your organization. Here are some key points about managing users and groups in Microsoft Entra:

Users

- **Creation and Management:** You can create individual users and manage their access to resources. This includes assigning roles, permissions, and licenses.
- **Dynamic Membership:** Users can be automatically added or removed from groups based on specific attributes, making it easier to manage large numbers of users.

Groups

➔ **Types of Groups: There are two main types of groups in Microsoft Entra:**

- ◆ **Security Groups:** Used to manage user and computer access to shared resources. Members can include users, devices, service principals, and other groups.
- ◆ **Microsoft 365 Groups:** Provide collaboration features like shared mailboxes, calendars, and files. These groups can include only users**.**

➔ **Group Membership:** Groups can have assigned membership, where specific users are added manually, or dynamic membership, where users are added based on rules.

➔ **Access Management:** Groups can be used to assign access to applications, data, and resources, simplifying the management of permissions.

Benefits

Simplified Management: Using groups to manage access and permissions reduces the complexity of managing individual user permissions.

Enhanced Security: By applying the principle of least privilege, you can limit access to only those who need it, reducing the risk of security breaches.

Microsoft Entra primary methods to manage them:

1. Microsoft Entra Admin Center

- **Creating a User:**
  1. Navigate to "Users" > "All users."
  2. Click "New user."
  3. Fill in the required information, such as display name, user name, and password.
  4. Assign necessary licenses and roles.
- **Managing Groups:**
  1. Navigate to "Groups" > "All groups."
  2. Click "New group."
  3. Choose the group type (security group or Microsoft 365 group).
  4. Provide a group name and description.
  5. Add members to the group.

2. Microsoft Entra PowerShell

Creating a User:

PowerShell
Connect-AzureAD
New-AzureADUser -DisplayName "John Doe" -UserPrincipalName "johndoe@example.com" -Password "Password123!"

- Use code with caution

Creating a Group:

PowerShell
Connect-AzureAD
New-AzureADGroup -DisplayName "Marketing Team" -SecurityEnabled $true

- Use code with caution.

3. Microsoft Graph API

You can use the Microsoft Graph API to programmatically manage users and groups. This is particularly useful for automation and integration with other systems.

Key Considerations for Effective User and Group Management:

➔ **User Provisioning:**
- **Manual Provisioning:** Create users manually in the Microsoft Entra admin center.
- **Automated Provisioning:** Use tools like Microsoft Identity Manager (MIM) or Microsoft Entra Connect to automate user provisioning from on-premises Active Directory.

➔ **Group Management:**
- **Dynamic Groups:** Create groups based on specific criteria, such as attributes or claims, to automate membership.
- **Nested Groups:** Organize groups into a hierarchical structure for better management and access control.

➔ **Role-Based Access Control (RBAC):**
- Assign appropriate roles to users and groups to control access to resources and applications.
- Use built-in roles or create custom roles to tailor permissions.

➔ **Password Policies:**
- Enforce strong password policies to enhance security.
- Configure password expiration, complexity requirements, and lockout policies.

➔ **Multi-Factor Authentication (MFA):**
- Require MFA for additional security.
- Configure MFA policies to enforce them for specific users or groups.

➔ **Single Sign-On (SSO):**
- Enable SSO for seamless access to applications.
- Configure SSO for both cloud-based and on-premises applications.

You can ensure secure and efficient access to your organization's resources by effectively managing users and groups in Microsoft Entra ID.

For more information, please refer to this → How to manage groups - Microsoft Entra

Manage Licenses in Microsoft Entra ID

Microsoft Entra ID services require you to license each of your users or groups for that service. Only users with active licenses will be able to access and use the licensed Microsoft Entra ID services for which that's true. Licenses apply to Tenants but are not transferable to other Tenants.

There are several license plans available for the Microsoft Entra ID service, including Microsoft Entra ID Free, Microsoft Entra ID Premium P1 and Premium P2 plans.

You must have one of the following licenses for every user who benefits from group-based licensing:

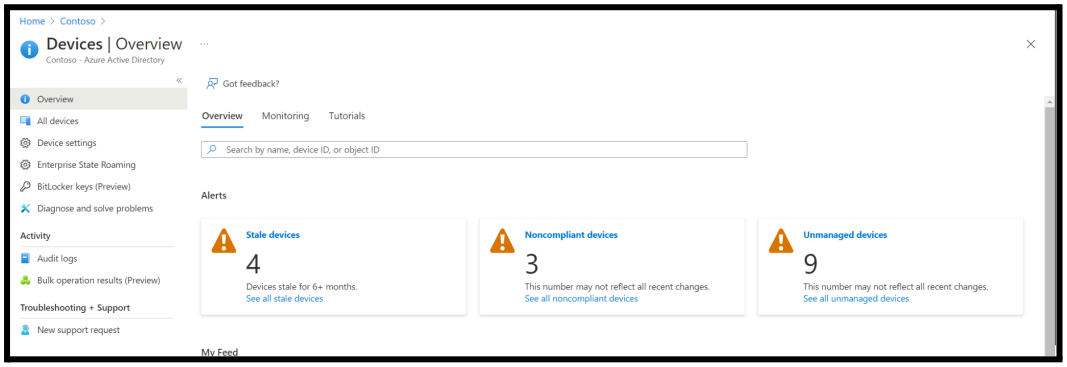
- Paid or trial subscription for Microsoft Entra ID Premium P1 and above
- Paid or trial edition of Microsoft 365 Business Premium or Office 365 Enterprise E3 or Office 365 A3 or Office 365 GCC G3 or Office 365 E3 for GCCH or Office 365 E3 for DOD and above

Managing the device settings and identity

Microsoft Entra ID provides a central place to manage device identities and monitor related event information.

You can access the devices overview by completing these steps:

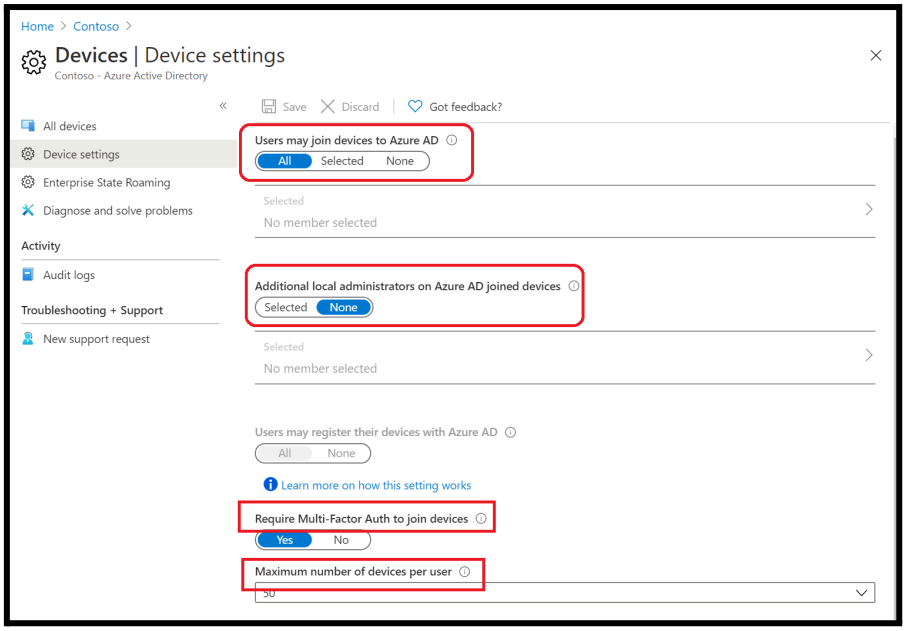
1. **Sign in to the Azure portal.**
2. Go to Microsoft Entra ID > Devices.



(Source: Microsoft Documentation)

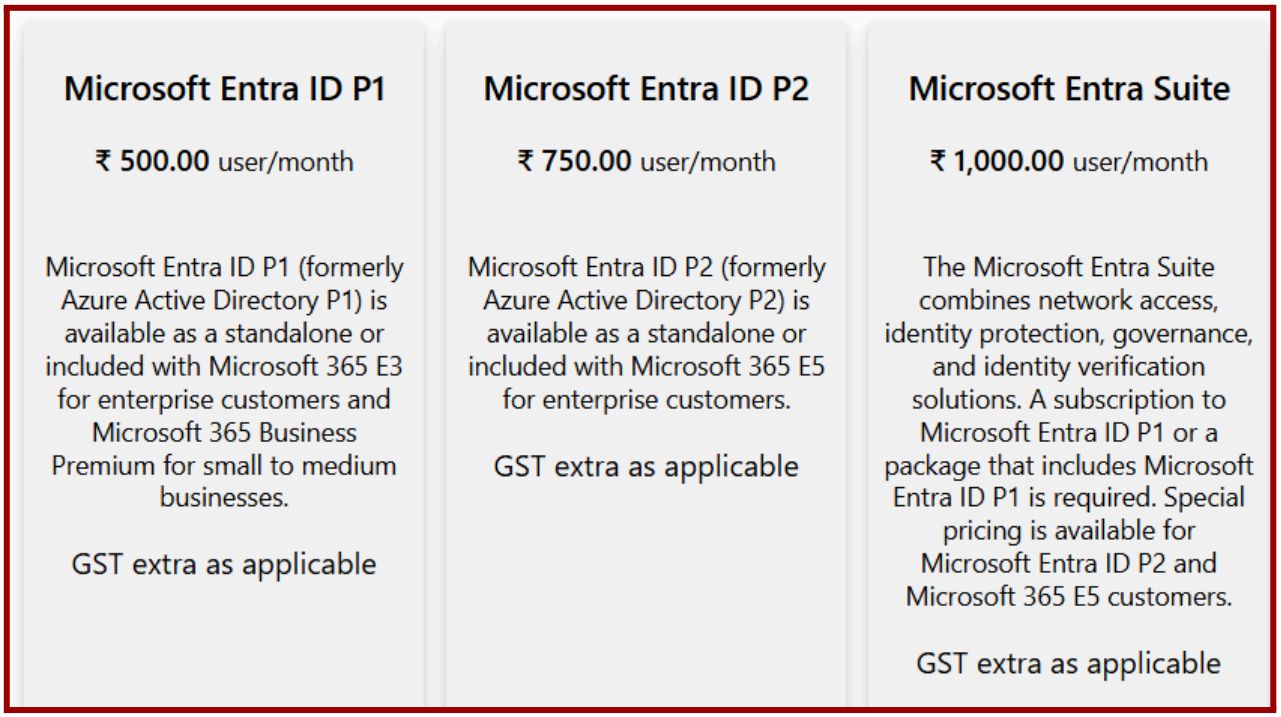
We can manage device identities by using the Azure portal. As an administrator, you can control the process of registering and joining devices by configuring the following device settings.

You must be assigned one of the following roles to view/manage device settings in the Azure portal: Global Administrator, Cloud Device Administrator, Global Reader and Directory Reader



(Source: Microsoft Documentation)

Microsoft Entra Pricing



(Source: Microsoft Documentation)

B) Microsoft Entra ID Access Management and Control

Access management in Microsoft Entra ID

There are three types of roles available for access management in Azure:

- Classic subscription administrator roles
- Azure role-based access control (RBAC) roles
- Microsoft Entra administrator roles

Azure Role-Based Access Control (Azure RBAC)

- The policy of any organization is to follow the principles of least privileges. One must not be given access beyond what is necessary to perform a role in the organization.
- The principles apply for cloud resources also. Let's take the example of a VM operator. His role dictates that he must be able to **start/stop/restart/create/delete** VMs.
- So, we use Azure **RBAC** to grant just that access. In our case, we will grant the operator the RBAC role of VM contributor.
- Azure **RBAC** is an authorization system. It uses Azure Resource Manager behind the scenes. Azure RBAC provides fine-grained control of access to Azure resources at various levels.
- The Policies can be applied with a boundary like being able to do so in a set of resource groups called scope.

Let's see some examples where we use Azure RBAC:

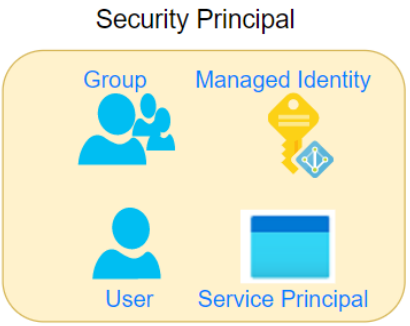
- Grant access DBA group to manage databases in 2 resource groups.
- A user can manage all resources in a resource group like VM, web apps, storage account, Vnet/Subnets.
- Grant one application to access to create resources.

How Azure RBAC works

We assign Azure roles to make RBAC work. A role assignment consists of three elements: security principal, role definition, and scope.

1. **Security principal**

   A *security principal* is an object that could represent a user or a group or a service principal or managed identity and requests access to Azure resources. We can grant access to any of these entities.



2. **Role definition**

A *role definition* is a collection of permissions and is called a *role*. A role definition will list the operations that can be performed. It could be something like read, write, and delete. We could grant access at a high level like owner, or even more specific roles like the VM operator where the access is limited to VM operations only.

Azure has *built-in roles* that you can use. For example, we have a contributor role where we can create all objects but we cannot grant. If we want to grant access to only certain resources, we will create a custom-defined role.

As you can see below, we can apply policies against the data stored within the scope's resources. For example, the secret within a key will be data, and we can dictate whether the data can be read or not.

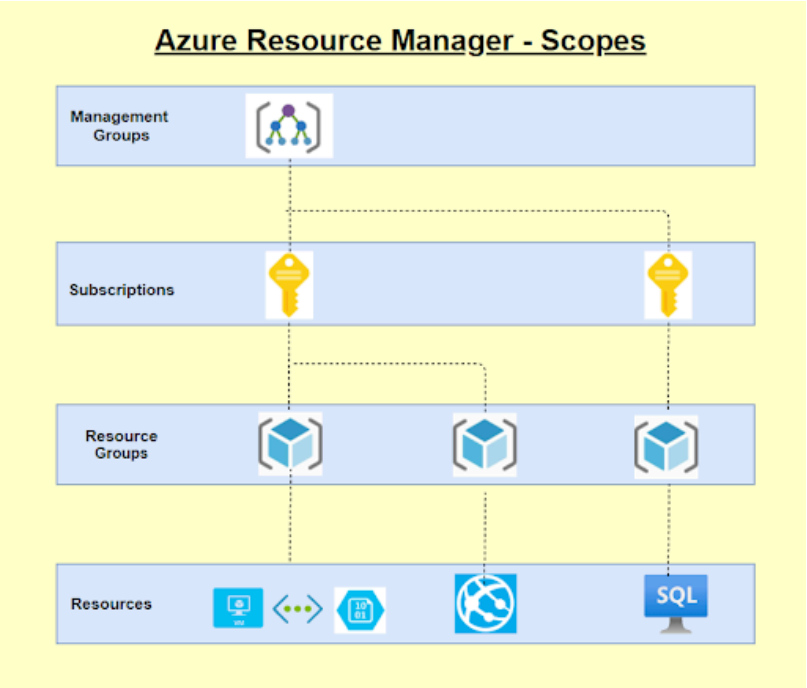
3. **Scope**

The scope is the set of resources to which we apply for the access. Let's say that we grant a VM operator role to a person, but we don't want that person to be able to stop VMs in production, then we apply the scope to non-production subscription or resource group only.

A scope can be applied at the four levels:

a. **Management group**
b. **Subscription**
c. **Resource group**
d. **Resource**

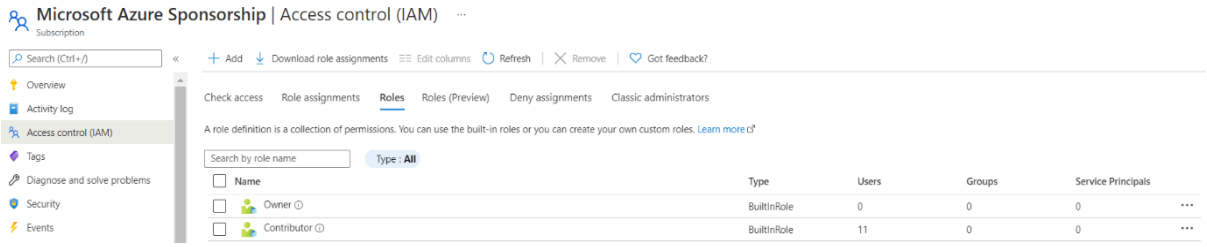
- Scopes follow a hierarchical structure, and they follow a parent-child relationship.
- Scopes applied at a higher level are inherited by the resources below them.
- For example, a policy with a scope of Management groups will be inherited by all subscriptions under it.
- Likewise, a policy scoped at the RG level will be inherited by all resources under it.



(Source: Microsoft Documentation)

4. Role assignments

We assign the role to the user or group. When we assign the role, the user gets the privileges. And we simply remove the role assignment when we want to revoke the access.
Under IAM, for every resource, we can see the roles under the roles tab.



5. Deny assignments

Earlier RBAC had only allowed, but now it can be denied assignments also. If there is a deny assignment, the user will be blocked from doing the action. Deny assignments take precedence over role assignments where a given user has both allow and deny but deny will be the end action.

6. License requirements

RBAC feature is free and included with our Azure subscription.

Access management for cloud resources is a critical function or task for any organization using the cloud. Role-Based Access Control (RBAC) is a mechanism that helps you manage who can access your Azure resources. RBAC allows you to control what specific users can do on specific resources and what resources each user can access.

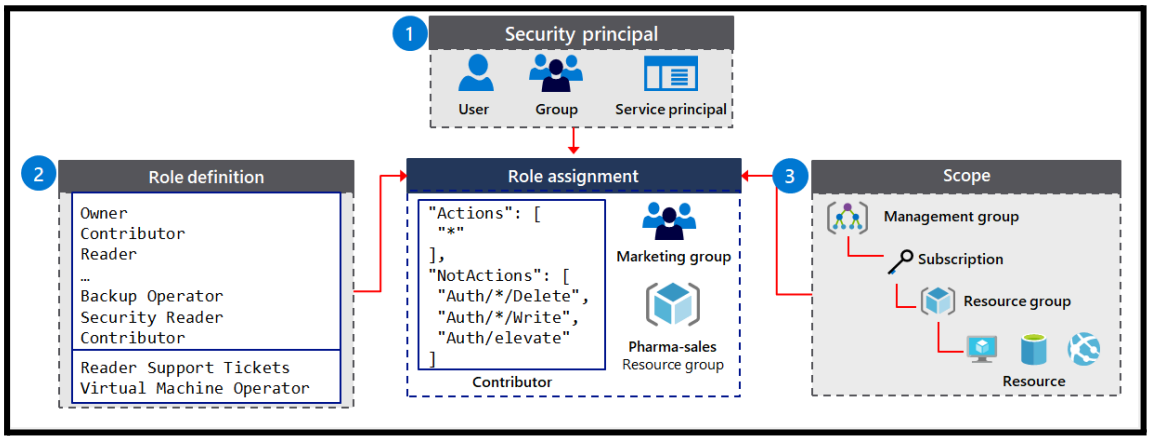
Azure RBAC is an authorization system built on top of Azure Resource Manager and provides fine-grained access management to resources in Azure.

There are a few things you need to consider when using Azure RBAC those are
- - -> Requests, Roles, Permissions and Custom (built-in) definitions

How to work with Azure RBAC - Images needs to update

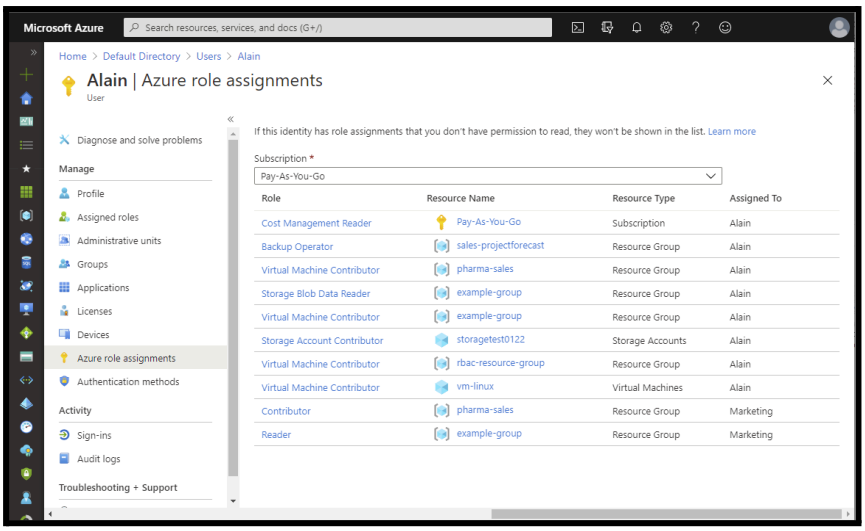
Things to consider when assigning scope levels for roles

The following diagram shows an example of how scopes can be applied for a role to grant varying levels of access for different users.



(Source: Microsoft Documentation)

Azure role assignments



(Source: Microsoft Documentation)

Manage built-in Azure roles

Managing built-in Azure roles is a key aspect of Azure Role-Based Access Control (RBAC), which helps you control access to your Azure resources. Here are some important points about managing these roles:

Built-in Roles

Azure provides several built-in roles you can assign to users, groups, service principals, and managed identities. Some of the most commonly used built-in roles include:

1. Owner - Can create and manage most Azure resources, including virtual machines, storage accounts, and networks.
2. Contributor - Can manage access to Azure resources, but cannot delete the resource group or subscription.
3. Reader - Can view Azure resources but cannot make changes.

4. User Access Administrator - Can manage user access to Azure resources, such as adding or removing users from groups

Assigning Roles

Prerequisites

➔ Step 1: Identify the needed scope
➔ Step 2: Open the Add role assignment page
➔ Step 3: Select the appropriate role
➔ Step 4: Select who needs access

➔ Step 5: Add condition [Optional]

➔ Step 6: Select the assignment type

➔ Step 7: Assign role

Custom Roles

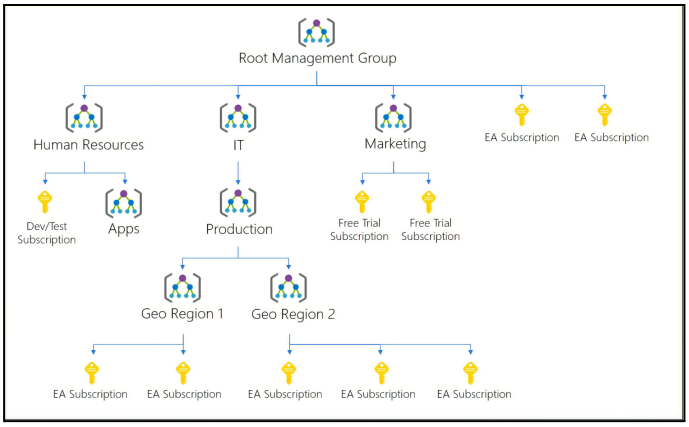
If the built-in roles do not meet your specific needs, you can create custom roles. Custom roles allow you to define specific permissions tailored to your requirements

Benefits of Using Built-in Roles

- Simplified Management: Built-in roles cover a wide range of common scenarios, making it easier to manage access without needing to create custom roles.
- Security: By assigning the appropriate roles, you can ensure that users have the minimum permissions necessary to perform their tasks, following the principle of least privilege1.

C. Managing Azure subscriptions and governance

Hierarchy of management groups and subscriptions

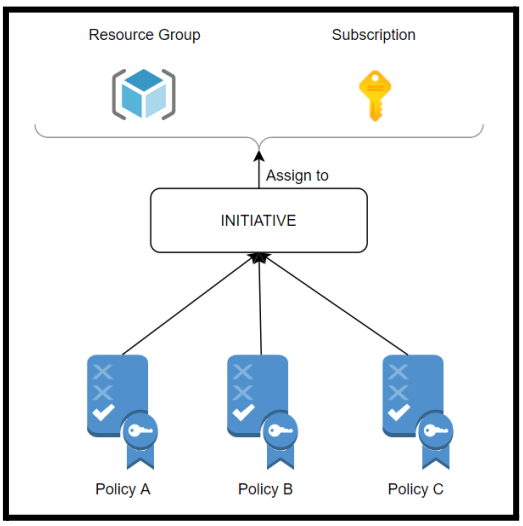


(Source: Microsoft Documentation)

Azure Policy Introduction

- Every organization has a set of standards which are set up. Some of these could be best practices for smooth functioning or cost optimization.
- Others could be mandatory compliance adhering to Government laws and/or governing bodies like **ISO** or **HIPAA**.
- Azure Policy is a free service in Azure that we could use to define, assign, and manage standards for resources.
- Let's say that **GDPR** policy mandates that data should not leave the country. T

- Then we can create a policy that could prevent or just mark as non-compliant if data were stored outside the country.
- Once such a policy is set, it would even point to such previously created resources which are non-compliant.
- With quite a few built-in policies under categories such as *Storage, Networking, Compute, Security, and Monitoring,* it is very convenient to select the policy that suits us and use them simply.



Configure and Manage Azure Policies

Azure Policy

It's a service you use to create, assign, and manage policies in Azure. It helps you manage and prevent IT problems with policy definitions that enforce rules and effects for your resources.

Azure Policy has four main benefits/advantages as follows

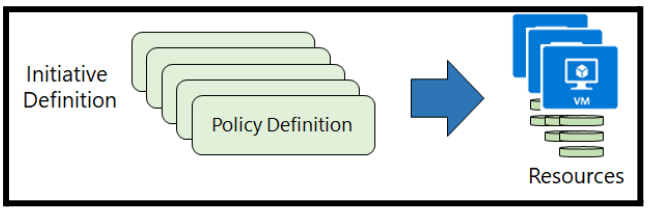
- Enforce rules and compliance
- Apply policies at scale
- Perform remediation
- Exercise governance

Common use cases:

- Implementing Governance,
- Regulatory compliance like GDPR/HIPAA/PCI DSS,
- Security, Cost and Management

Create Azure policies

There are four basic steps to creating and working with policy definitions in Azure Policy.



Step 1: Policy Definition

First, we create a policy definition then we could also use existing definitions and then we could take multiple policies and create a policy definition.

Step 2: Policy Initiative

Once the policy definition is done, we need to create the initiative definition. We can select any number of policies we need and create a group to add the policies. Then we can create initiative parameters and policy parameters and finally we can create the initiative definition.

Step 3: Scope the initiative definition

Azure Policy lets you control how your initiative definitions are applied to resources in your organization. You can limit the scope of the initiative definition to specific management groups, memberships, or resource groups.

Step 4: Determine compliance

After you assign an initiative definition, you can assess the compliance status of all your resources. Individual resources, resource groups, and subscriptions within a scope can be excluded from being affected by policy rules. Exceptions are handled individually for each assignment.

You can try this - - - -> Interactive lab simulation - Training | Microsoft Learn

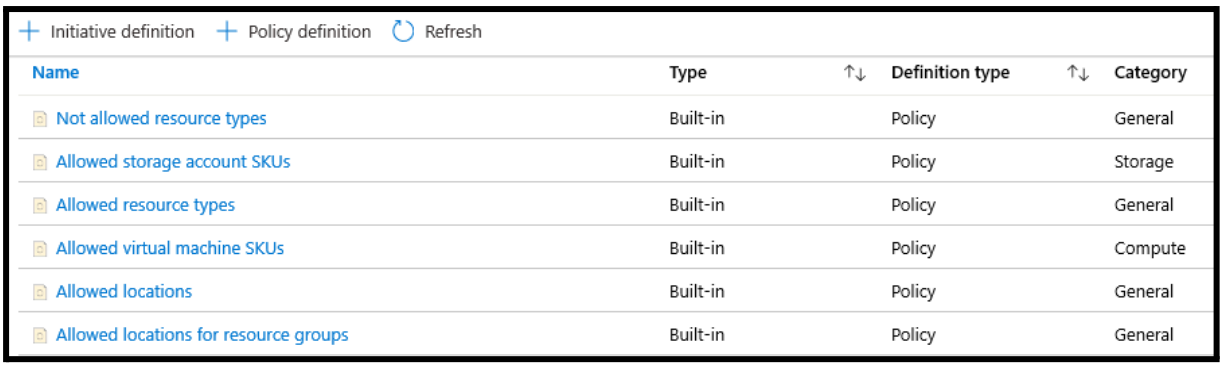
How are policies evaluated

The following are the times or events that cause a resource to be evaluated:

- During the standard compliance evaluation cycle, which occurs once every 24 hours.
- A policy or initiative is newly assigned to a scope.
- A resource is created, updated, or deleted in a scope with a policy assignment.
- A policy or initiative already assigned to a scope is updated.

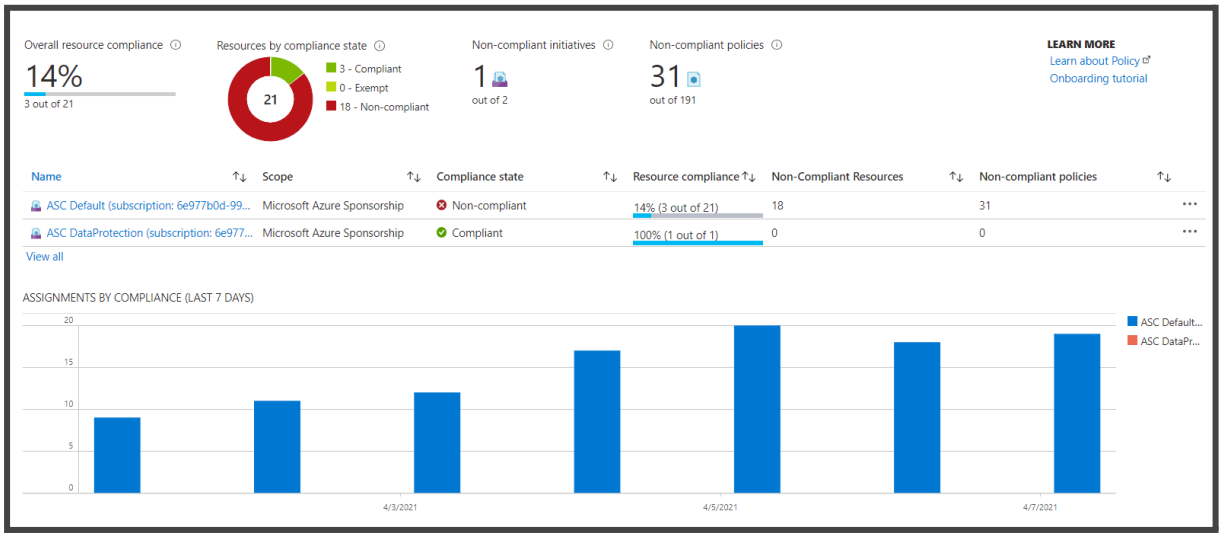
In Azure Policy built-in policy definitions:

- **Allowed Storage Account SKUs** (Deny): Determines whether a storage account is implemented in a set of SKU sizes. The effect is to reject all storage accounts that do not adhere to the defined set of SKU sizes.
- **Allowed Resource Type (Deny):** Defines the types of resources you can run. The effect is to deny all resources that are not part of this defined list.
- **Allowed Locations (Deny):** Restricts or limits the locations available for new resources. This effect is used to implement your geo-optimized requirements.
- **Allowed Virtual Machine SKUs (Deny):** Specifies the set of virtual machine SKUs you can run and deploy.
- **Add a tag to resources (Modify):** Applies the required tag and its default value if not specified by the deploy request.
- **Not allowed resource types (Deny):** Prevents list of resource types from deployed/ running.



(Source: Microsoft Documentation)

Ex: All Azure Policy data and objects are encrypted at rest.
Once set up, we can see the non-compliant policies, and we will be able to remediate.



(Source: Microsoft Documentation)

Configuring Resource locks

Being an administrator, you can lock resource groups or resource Azure subscriptions to protect them from accidental user deletions and modifications. Such locks overrides any user permissions.

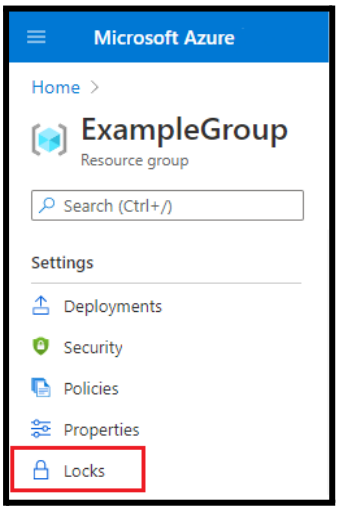
- In the portal - - - > These locks are called **“Delete”** and **“Read-only”**
- On the command line - - - > These locks are called **“CanNotDelete”** and **“ReadOnly”**
- You can set locks that prevent deletions or changes.

You can use management locks to apply a restriction across all users and roles instead of role-based access control (RBAC). For, Considerations before applying your locks,

Please refer to this → Protect your Azure resources with a lock - Azure Resource Manager

1. Go to the Show portal Menu → Select any resource, resource group, or subscription that you wish to lock then

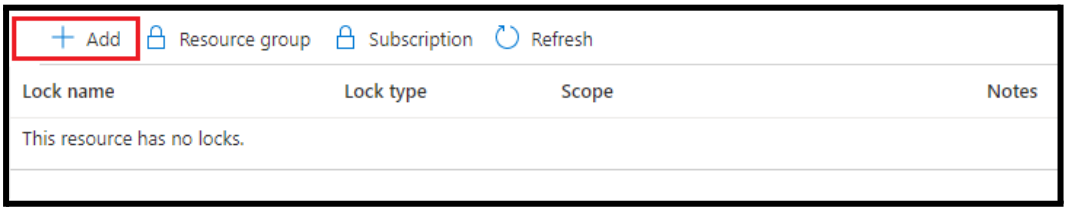
2. Under the Settings blade you can see the “Locks”



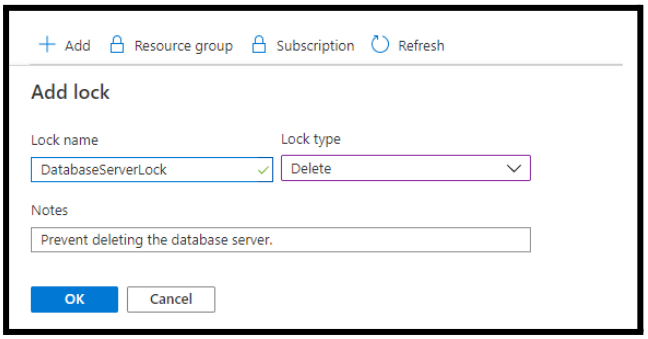
3. You can click the Locks option if you want to add lock to any resource, etc.

4. You need to click the Add option. If you want to create a lock at the parent level, select Parent. The currently selected resource acquires the lock from the parent.

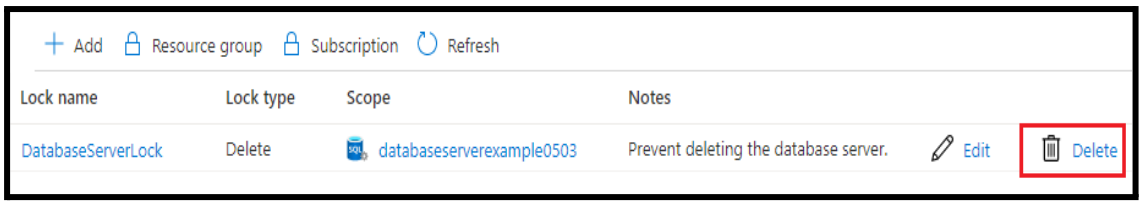
5. For example, you could lock the resource group to apply a lock to all its resources.



6. Give the lock a name and lock level. Optionally, you can add notes describing the lock.



7. To delete the lock, select the Delete option/button.



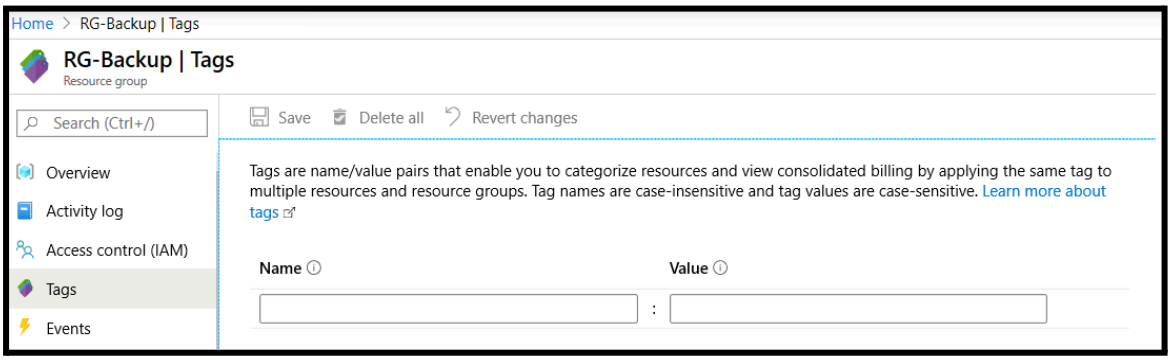
Configure lock using Template : When using an ARM template or Bicep file to implement a lock, it's a good idea to understand how the expansion scope and the lock scope work together. For more info → Pls refer to [Configure lock using an ARM template or Bicep file]

(Source: Microsoft Documentation)

Apply and manage tags on resources

Azure Tags are a part of the Azure Resource Manager that are used to track, manage, and group your resources and analyze their costs. Also, tagging helps automate resource deployments in your Azure environment. You can apply tags to your Azure resources to logically organize them by category. Tags are useful for sorting, searching, organizing and analyzing your resources.

Example that shows how to add tags for a resource group in the Azure portal:



(Source: Microsoft Documentation)

Things to know about resource Tags

- Each resource tag has a name and a value.
- You could have the tag name **Server** and the value **Production** or **Development**.
- The tag name is constant for all resources to which the tag applies.
- The tag value can be selected from a defined set of values or unique to a particular resource instance.
- Azure currently supports up to 50 tags per resource and resource group.
- Tags can be placed on a resource during creation or added to an existing resource.
- Tags applied to a resource group are not inherited by resources within resource group.

You need to consider a few things when using resource tags

- **Searching on tag data**: Search for resources in your membership by querying on tag name and value.
- **Finding relevant resources:** Retrieve related resources from other resource groups by searching on tag name or value.
- **Billing Group Data:** Group resources such as virtual machines by cost center and production environment.
- **Tags are created with PowerShell or Azure CLI:** Create multiple resource tags programmatically by using Azure PowerShell or Azure CLI.

Managing Azure resource groups

What is a Resource Group?

A resource group is a container that holds related resources for an Azure solution. A resource group stores metadata about resources. A resource group can contain all the resources for a solution or only the resources you want to manage as a group.

For managing Azure resource groups, there are three ways

- Manage resource groups by using Azure portal - Azure Resource Manager
- Manage Azure resource groups by using Azure CLI
- Manage Azure resource groups by using Azure PowerShell

We need to know how to use the Azure portal with Azure Resource Manager to manage your Azure resource groups.

Create resource groups

A. Sign in to the Azure portal → Select Resource groups → Select Add.
B. Enter the following values:
   ➔ Subscription: Select your Azure subscription.
   ➔ Resource group: Enter a new resource group name.
   ➔ Region: Select an Azure location, such as West US, South India, etc.
C. Select Review + Create
D. Select Create. It takes a few seconds to create a resource group.
E. Select Refresh from the top menu to refresh the resource group list, and then select the newly created resource group to open it.

Lock resource groups

Locking prevents other users in your organization from accidentally deleting or modifying critical resources, such as Azure subscription, resource group, or resource.

A. Open the resource group you want to lock. See Open resource groups.
B. In the left pane, select Locks.
C. To add a lock to the resource group, select Add.
D. Enter Lock name, Lock type, and Notes. The lock types include Read-only, and Delete.

2. List resource groups

A. Sign in to the Azure portal.
B. To list the resource groups, select Resource groups.
C. To customize the information displayed for the resource groups, select Edit columns.

Open resource groups

1. Sign in to the Azure portal.
2. Select Resource groups.
3. Select the resource group you want to open.

Rest other options are

Deploy resources to a resource group : After you create a Resource Manager template, you can use the Azure portal to deploy your Azure resources.

Move to another resource group or subscription: You can move the resources in the group to another resource group

Tag resource groups : You can apply tags to resource groups and resources to logically organize your assets

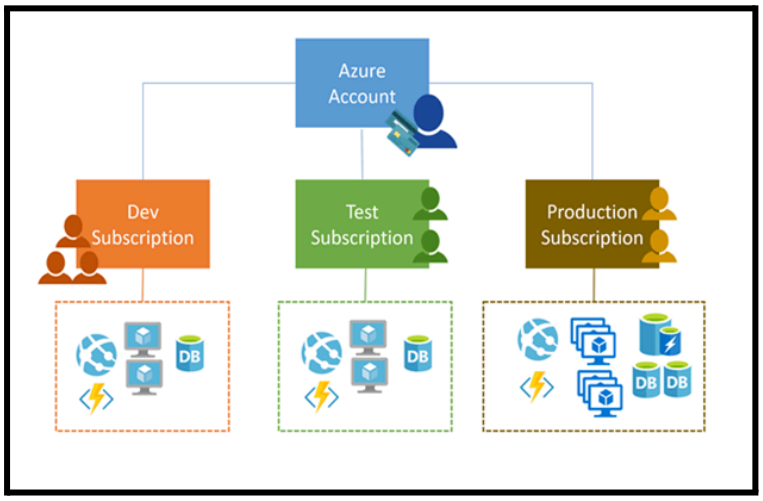
Export resource groups to templates

Manage access to resource groups by using Azure RBAC

(Source : Microsoft Documentation)

Manage and organize Azure subscriptions

An Azure subscription is a logical unit of Azure services linked to an Azure account.
An Azure account is an identity in Microsoft Entra ID or a directory trusted by Microsoft Entra ID, for example, a work or school account.



(Source : Microsoft Document)

Subscriptions help you manage access to Azure cloud service resources and help you control how resource usage is reported, billed, and paid for.

Characteristics of Azure Subscriptions are as follows:

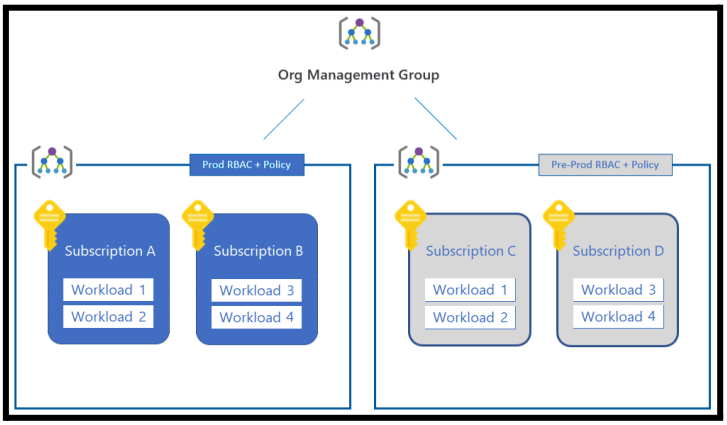
- Each Azure cloud service belongs to a subscription
- Each subscription has a different billing and payment configuration.
- You can link more than one Azure account to a single Azure subscription.
- you can link Multiple subscriptions to a single Azure account
- Billing for Azure services is on a per-subscription basis.
- Programmatic operations for the cloud service may require a subscription ID.
- If your Azure account is the only account associated with the subscription, you are responsible for billing requirements.

Azure offers free and paid subscription options to meet different needs and requirements.

Most common subscriptions: Free, Pay-As-You-Go, Enterprise Agreement & Student.

For our organization, we can choose a combination of collection options and subscription options to suit your business circumstances.

If you only have a few subscriptions, it's easy to manage them independently. But what if you have a lot of subscriptions? You can then create a management group hierarchy to help manage your memberships and resources.

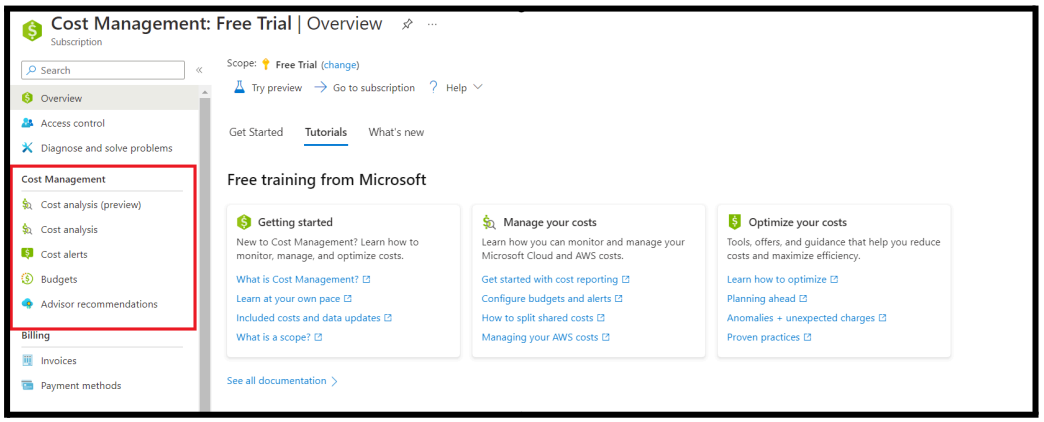


(Source: Microsoft Documentation)

Manage costs by using alerts, budgets, and recommendations

Cost control is a key factor in maximizing the value of your investment in the cloud. There are many scenarios where cost visibility, reporting, and cost-based orchestration are critical to sustaining business operations.

With Azure products and services, you only pay for what you use. When you create and use Azure resources, you are charged for the resources. Microsoft Cost Management provides support for administrative billing tasks and helps you manage billing access to expenses.



(Source: Microsoft Documentation)

Cost analysis is used to explore and analyze your organizational costs. You can view aggregated costs by an organization to understand where costs have increased and identify cost trends. Monitor accumulated costs over time to assess monthly, quarterly or annual cost trends against budget.

Budgets are commonly used as part of cost control and features are used to establish and maintain budgets. This helps prevent exceeding spending limits or limits. You can use analytics data to inform others about their spending to proactively manage spending. Budgeting features help you see how a company's spending is developing over time.

Recommendations can optimize and improve efficiency by identifying idle and unused resources. This can reveal less expensive resource options. Using the recommendations, you can change the way you use your resources to save money.

Cost Alerts are used to monitor your Azure usage and spending. Cost alerts are automatically generated when Azure resources are consumed. Alerts show all active expense management and billing alerts in one place.
There are three types: Budget alerts, Credit alerts, and Department spending quota alerts.

Configuring Azure Management Groups

Governance in Azure is one aspect of Azure Management. Management groups help you to organize or manage resources and subscriptions in Azure. Management groups give you enterprise-grade management at scale, no matter what type of subscriptions you have. However, all subscriptions in the same management group must trust the same Microsoft Entra ID tenant.

Organizations using multiple memberships need a way to efficiently manage access, policies and compliance. Azure management groups provide a level of scope and control over your subscriptions. You can use management groups as containers to manage access, policy, and compliance across your subscriptions. You can create a management group with Azure Policy by using the Portal, PowerShell, or Azure CLI.

Azure Management Groups Characteristics

- A single directory can support up to 10,000 management groups.
- The root management group cannot be moved or deleted unlike other management groups.
- All new subscriptions are placed under the top-level management group or the root group.
- The management group tree supports up to six levels of depth.
- Azure RBAC authorization is not enabled by default for management group operations.

When using management groups in Azure Policy to manage subscriptions, there are a few things you need to consider ⟶ Custom hierarchies and groups, Policy inheritance, Compliance rules, and Cost reporting.

Implement and Manage Storage

Introduction to Azure Storage

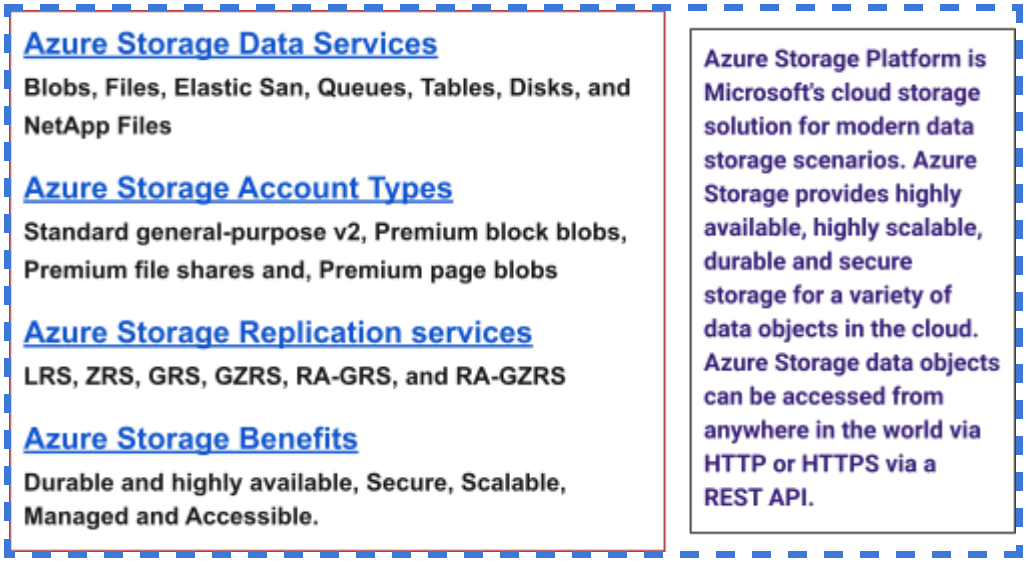
Azure offers several ways to store your data, including multiple database options such as SQL Database, Azure Cosmos DB, and Azure Table Storage.

Azure provides several ways to store and send messages, such as queues and event hubs. You can also store loose files using services like Azure Files and Azure Blobs. A storage account is a container that groups together a set of Azure Storage services. Only data services from Azure Storage are included in the storage account.

Azure Storage is a service that you can

- Used to store files, Messages, Tables, and other types of information.
- Developers use Azure Storage for working data(websites, mobile & desktop applications)
- It is also used by IaaS and PaaS cloud services.

Azure Storage - Services, Types, and Benefits



A. Configure access to storage

Create and configure storage accounts

The Azure Storage platform includes the following data services

- Azure Blobs: A massively scalable object store for text and binary data.
- Azure Files: Managed file shares for cloud or on-premises deployments.
- Azure Elastic SAN (preview): A fully integrated solution that makes it easy to deploy, scale, manage and configure a SAN on Azure.
- Azure Tables: NoSQL store for schemaless storage of structured data.
- Azure Queues: Messaging store for reliable messaging between application components.

- Azure Disks: Block-level storage volumes for Azure VMs.
- Azure NetApp Files: Managed by NetApp accounts and accessible via NFS, SMB, and dual-protocol volumes.

Create a storage account

You can create a storage account by using Azure portal, PowerShell, CLI, ARM Templates.

You need to know and fill all 7 tabs to create an Azure storage account:

1. **Basic Tab: I**t provides the essential information for your storage account.
2. **Advanced tab:** You can configure additional options and modify default settings for your new storage account.
3. **Networking tab**, You can configure network connectivity and routing preference settings for your new storage account.
4. **Data protection tab:** You can configure data protection options for blob data in your new account
5. **Encryption tab,** You can configure options for how your data is encrypted in the cloud.
6. **Tags tab:** You can specify Resource Manager tags to help manage your Azure resources.
7. **Review + Create tab:** Azure runs validation on the storage account settings you selected. If the verification is passed, you can proceed to create a storage account.

For more information, pls refer to this link →Create a storage account - Azure Storage

Network access to storage accounts

A virtual network (VNet) **Service endpoint** provides secure and direct connectivity to Azure services in an optimized path through the Azure backbone network. Endpoints allow you to secure your critical Azure service resources only to your virtual networks.

➔ You can configure storage accounts to allow access only from specific subnets.
➔ Allowed subnets can belong to a VNet in the same subscription or a different subscription, including subscriptions that belong to a different Microsoft Entra ID tenant.
➔ You can enable a Service endpoint for Azure Storage within the VNet.
➔ Service endpoint routes traffic from VNet through an optimal path to Azure Storage service.
➔ The identities of the subnet and the virtual network are also transmitted with each request.
➔ Administrators can then configure network rules for the storage account that allow requests to be received from specific subnets in a VNet.
➔ Clients granted access via these network rules must continue to meet the authorization requirements of the storage account to access the data.

Each storage account supports up to 200 virtual network rules, which may be combined with IP network rules.

For your information: You can refer to this link → Access storage - Training | Microsoft Learn

Each object you store in Azure Storage has a unique URL address. Your storage account name forms the subdomain portion of the URL address. A combination of subdomain and domain name, which is specific to each service, forms the endpoint for your storage account.

Here you can see an example: If your storage account name is ***whizstorageaccount***, the default endpoints for your storage account for Azure services are as shown in the following table:

Service	Default endpoint
Container service	//whizstorageaccount.blob.core.windows.net
Table service	//whizstorageaccount.table.core.windows.net
Queue service	//whizstorageaccount.queue.core.windows.net
File service	//whizstorageaccount.file.core.windows.net

(Source : Microsoft Documentation)

Azure Storage Security Characteristics

Administrators use various strategies to ensure that their data is secure. Common mechanisms include encryption, authentication, authorization, and user access control with credentials, file permissions, and private signatures. Azure Storage provides a suite of security capabilities based on common strategies to help you secure your data.

The below are characteristics of Azure Storage Security

★ Encryption,

★ Authentication (Microsoft Entra),

★ Data in transit,

★ Disk encryption,

★ Shared access signatures (SAS),

★ Authorization.

Shared Access Signature(SAS)

A Shared Access Signature (SAS) is a Uniform Resource Identifier (URI) that grants controlled access rights and secure delegated access to resources in a storage account.

It is a secure way to share your storage resources without compromising your account keys.

With this, you have granular control over how a client can access your data. Like a few below

- What resources the client may access?
- What permissions do they have to those resources?
- How long is the SAS valid?

Azure Storage supports three types of shared access signatures:

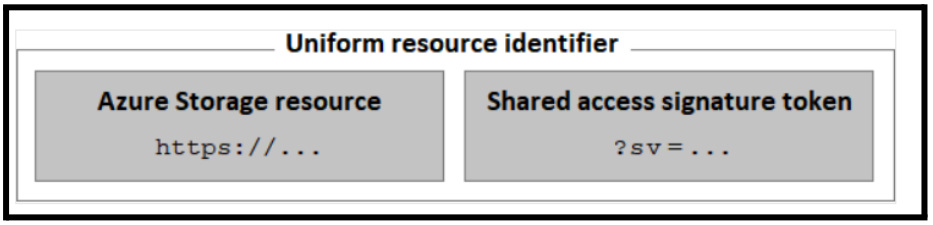
◆ User delegation SAS is secured with Microsoft Entra ID credentials and permissions specified for SAS. User delegation applies only to SAS blob storage.
For more information pls refer to —> Create a user delegation SAS (REST API).

- Service SAS is secured with the storage account key. A service SAS delegates access to a resource in only one of the Azure Storage services.
  For more information pls refer to this link —> Create a service SAS (REST API).

- Account SAS is secured with the storage account key. An account SAS delegates access to resources in one or more of the storage services.
  For more information pls refer to this link —> Create an account SAS (REST API).

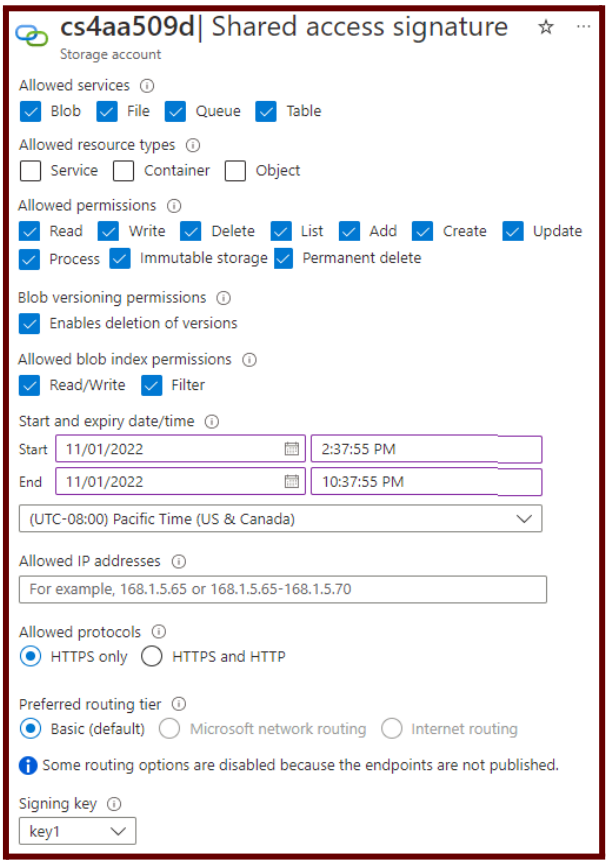
URI and SAS parameters:

When you create your Shared Access Signature (SAS), a Uniform Resource Identifier (URI) is created by using parameters and tokens. The URI contains your Azure Storage resource URI and the SAS token.



Example: Configure a shared access signature - (Source : Microsoft Documentation)

In the Azure portal, you configure several settings to create SAS. As you review these details, consider how you can implement shared access signatures in your storage security solution.



(Source: Shared access signatures to delegate access)

Stored Access Policies

A stored access policy provides an additional level of control over service-level shared access signatures (SASs) on the server side. Setting up a stored access policy can be used to group shared access signatures and provide additional restrictions for policy-bound signatures.

You can use a stored access policy to change permissions for the start time, expiration time, or signature. You can also use a stored access policy to revoke a signature after issuing it.

Stored access policies support → Blob containers, File shares, Queues, and Tables.

Example: The stored access policy you create for a blob container can be used for all the blobs in the container and for the container itself. A stored access policy is created with the following properties:

- **Identifier**: The name you use to reference the stored access policy.
- **Start time:** A DateTimeOffset value for the date and time when the policy might start to be used. This value can be null.
- **Expiry time:** A DateTimeOffset value for the date and time when the policy expires. After this time, requests to the storage will fail with a 403 error-code message.
- **Permissions:** The list of permissions as a string that can be one or all of acdlrw.

You can create stored access policy with C# code by using Azure Portal /Azure CLI commands.

With C# .NET code

C# Copy

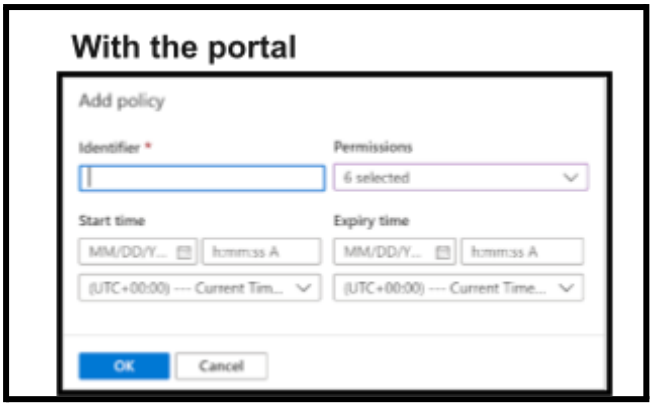
```
BlobSignedIdentifier identifier = new BlobSignedIdentifier
{
    Id = "stored access policy identifier",
    AccessPolicy = new BlobAccessPolicy
    {
        ExpiresOn = DateTimeOffset.UtcNow.AddHours(1),
        Permissions = "rw"
    }
};

blobContainer.SetAccessPolicy(permissions: new BlobSignedIdentifier[] { identifier });
```

With Azure CLI commands

Azure CLI Copy

```
az storage container policy create \
    --name <stored access policy identifier> \
    --container-name <container name> \
    --start <start time UTC datetime> \
    --expiry <expiry time UTC datetime> \
    --permissions <(a)dd, (c)reate, (d)elete, (l)ist, (r)ead, or (w)rite> \
    --account-key <storage account key> \
    --account-name <storage account name> \
```



(Source : Microsoft Documentation)

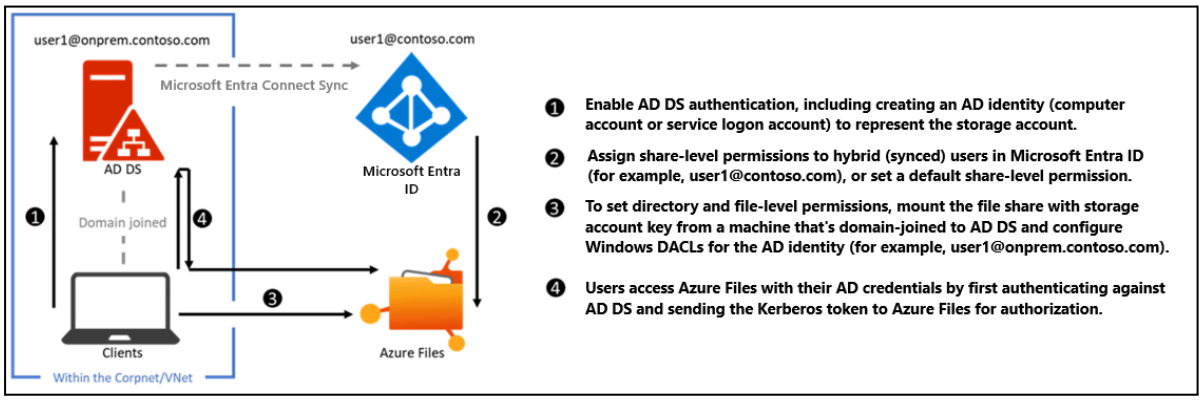
Azure Files identity-based authentication

Azure Files supports identity-based authentication, so you can control access to file shares using your existing directory services.

- **Kerberos Authentication** is a protocol designed to confirm the identity of users or hosts.
- **SMB protocol** serves as the standard method for network file sharing across the industry.
- **Microsoft Entra ID** is Microsoft's cloud-based identity management service that operates on a multi-tenant architecture. It integrates core directory services, application access management, and identity protection within a single platform.
- **Microsoft Entra Domain Services** will offer managed domain functionalities, including domain joining, group policies, LDAP, and both Kerberos and NTLM authentication, all of which are fully compatible with Active Directory Domain Services.
- **On-premises integration of Active Directory Domain Services (AD DS**) with Azure Files creates a system for storing directory data while ensuring it is accessible to users and administrators within the network.
- **Azure RBAC** facilitates precise access management for Azure, allowing you to grant users only the permissions essential for their roles.
- **Hybrid identities** in AD DS are those that sync with Microsoft Entra ID through either the on-premises Microsoft Entra Connect Sync application or the lightweight Microsoft Entra Connect cloud sync agent, which can be easily installed via the Microsoft Entra Admin Center.

Supported Authentication Methods

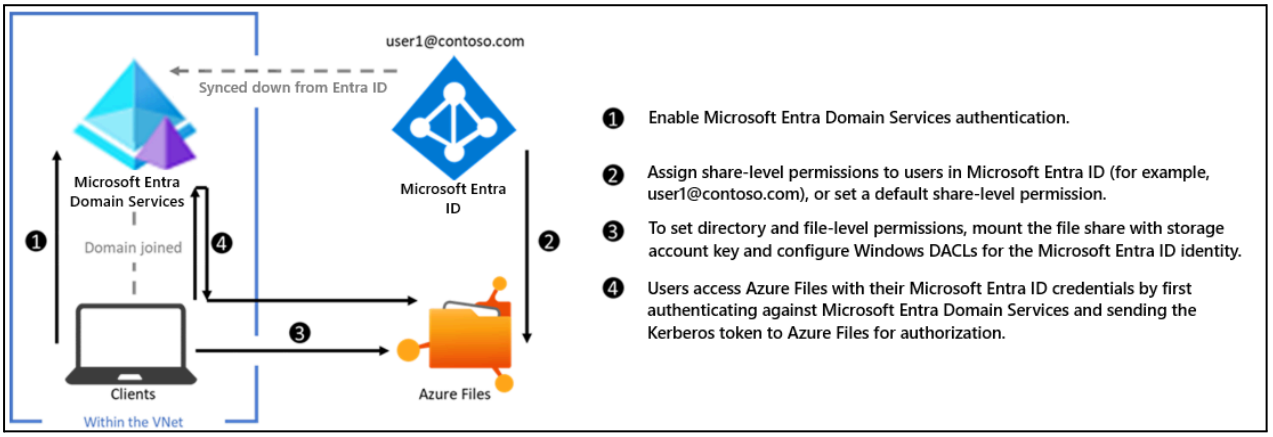
On-premises AD DS [Active Directory Domain Services]



To learn how to enable AD DS authentication, please refer to the following →

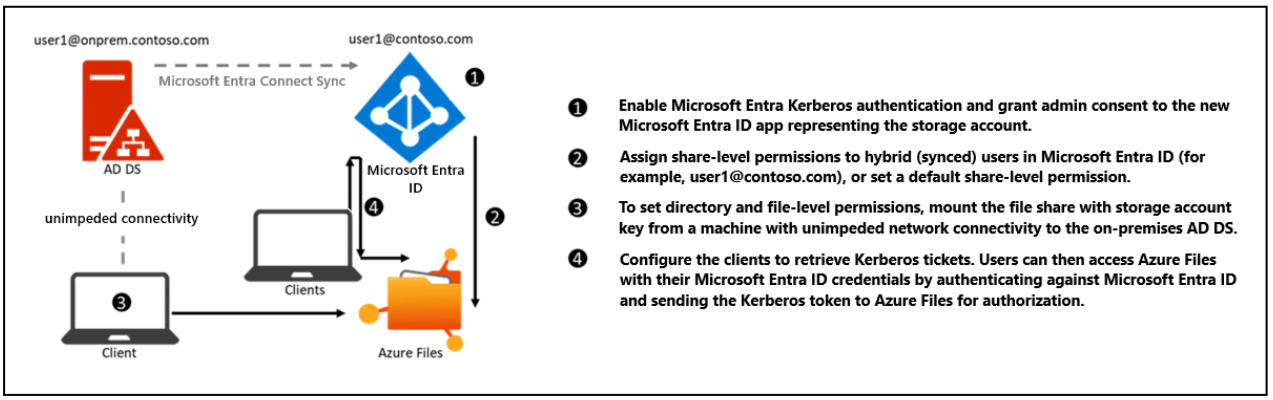
- On-premises AD DS authentication over SMB for Azure file shares.
- Enable AD DS authentication for Azure file shares.

Microsoft On-premises Domain Services:



To learn more, please refer to → Microsoft Entra Domain Services authentication on Azure Files.

Microsoft Entra Kerberos for hybrid identities:



To learn more, please refer to → Microsoft Entra Kerberos authentication.

AD Kerberos authentication for Linux clients:

Common use cases

1. **Replace on-premises file servers:** Azure file shares can replace on-premises file servers and provide high availability and scalability while maintaining seamless access for users using existing credentials.
2. **Migrate applications to Azure:** When you migrate applications to the cloud, you can maintain the same authentication model for your data.

Limitations

1. Network File System (NFS) shares do not support identity-based authentication.
2. Share-level permissions cannot be assigned to computer accounts using Azure RBAC.

Manage access keys

When you create a storage account, Azure generates two 512-bit storage account access keys for that account. These keys can be used to authenticate access to data in your storage account through shared key authentication.

Microsoft recommends that you use Azure Key Vault to manage your access keys and that you regularly rotate and regenerate your keys. For your Azure storage security solution, you can use Azure Key Vault to manage your encryption keys.

Azure Key Vault APIs can be used to generate encryption keys. You can also create your own encryption keys and store them in a key vault.

(Source: Microsoft Documentation)

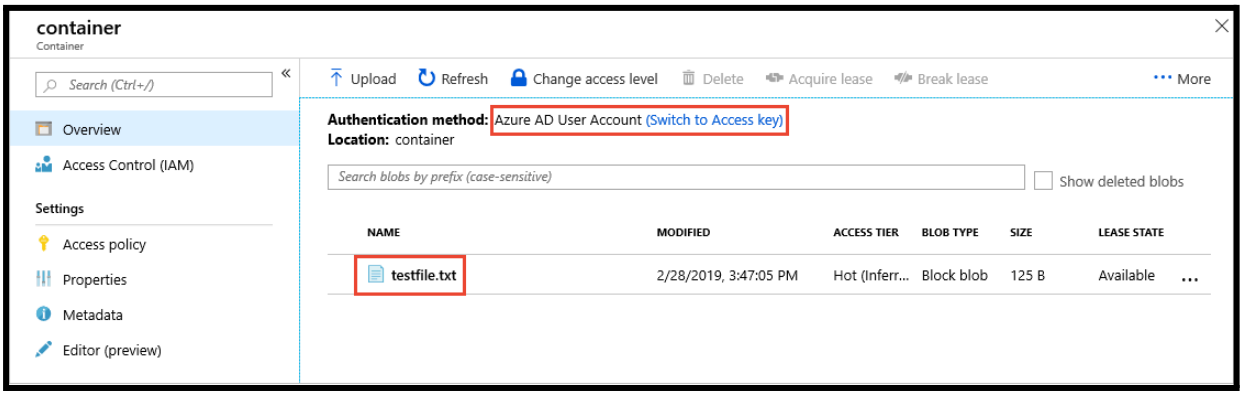
Microsoft Entra Authentication for a storage account

Azure Storage provides integration with Microsoft Entra for identity-based authorization of requests to Blob, Queue, and Table services.

Files stored in Azure storage are accessed by clients via HTTP/HTTPS. Azure checks each client request for authorization to access stored data.

Four options are available for blob storage: Public access, Entra Shared key, and SAS.

If you are authenticating using your Microsoft Entra ID account, you'll see Microsoft Entra ID User Account specified as the authentication method in the portal:



(Source: Microsoft Documentation)

Azure Storage Encryption

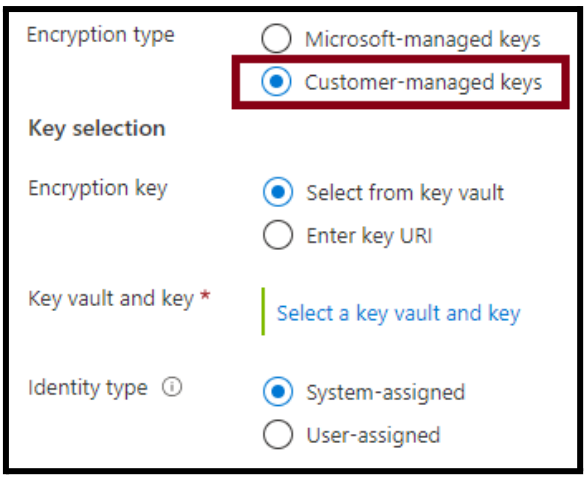
Azure Storage Encryption for data at rest protects your data by ensuring your organizational security and compliance commitments. The encryption and decryption processes are done automatically. Because your data is protected by default, you don't need to modify your code or applications.

Characteristics

- Data is automatically encrypted before persisting to Azure Managed Disks, Azure Blob Storage, Azure Queue Storage, Azure Cosmos DB (Azure Table Storage) or Azure Files.
- Data is automatically decrypted before recovery.
- Azure Storage encryption, encryption at rest, decryption, key management - transparent to users.
- All data written to Azure Storage is encrypted using 256-bit Advanced Encryption Standard (AES) encryption. AES is one of the strongest block ciphers available.
- Encryption is enabled and cannot be disabled for all new and existing storage accounts.

Configuring Azure Storage Encryption

In the Azure portal, you can configure customer-managed encryption keys. You can create your own keys or you can have keys managed by Microsoft. Consider how you can use Azure Key Vault to create your own customer-managed encryption keys.



(Source: Microsoft Documentation)

Encryption Type: Choose how the encryption key is managed: by Microsoft or by ourself

Encryption Key: Specify encryption key by entering a URI / select a key from the existing key vault.

B) Manage data in Azure storage accounts

Import and export jobs

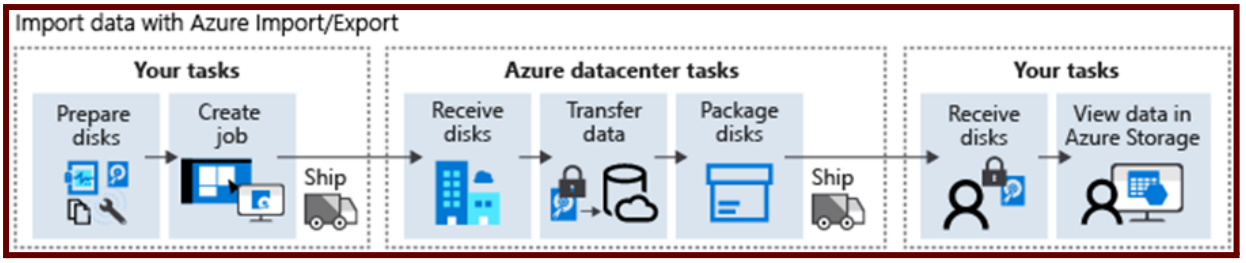
Azure Import/Export service is used for securely importing large amounts of data from on-prem or other sources to Azure blob storage and Azure Files by shipping disk drives to an Azure datacenter.

The service is also used to transfer data from Azure Blob storage → to disk drives and transport → to your on-premise sites. Data can be imported from one or more disk drives to Azure Blob storage or Azure Files.

Use cases/scenarios of Import and Export services in Azure

- Data migration to cloud: Moving large amounts of data to Azure is quick & cost effective.
- Content distribution: You can quickly send data to your customer sites
- Backup: Take backups of your on-premises data to store in Azure Storage.
- Data recovery: Retrieve large amounts of data stored in storage & deliver it to your on-premises.

The Azure Import/Export service enables data transfer into Azure Blobs and Azure Files by creating jobs. Use the Azure portal or the Azure Resource Manager REST API to create jobs. Each job is associated with a single storage account.



(Source: Microsoft Documentation)

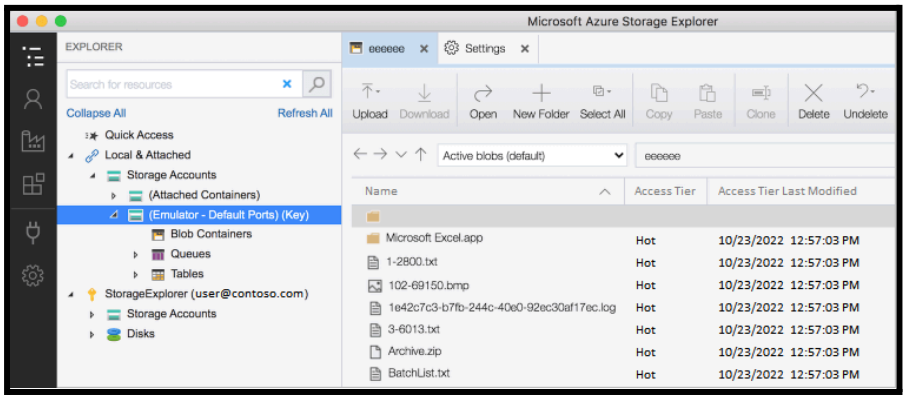
Below are steps for creating import/export jobs in Azure (Blob & File share)

Import data to Azure Blob Storage	Import data to Azure Files	Export data from Azure Blob storage
Prerequisites to import data to Azure Blob storage	**Prerequisites to import data to Azure Files**	**Prerequisites to export data from Azure Blob storage**
Step 1: Prepare the drives	Step 1: Prepare the drives	Step 1: Create an export job
Step 2: Create an import job	Step 2: Create an import job	Step 2: Ship the drives
Step 3: Configure customer managed key (Optional)	Step 3: Ship the drives to Azure datacenter	Step 3: Update the job with tracking information
Step 4: Ship the drives	Step 4: Update the job with tracking information	Step 4: Receive the disks
Step 5: Update job with tracking information	Step 5: Verify data upload to Azure	Step 5: Unlock the disks
Step 6: Verify data upload to Azure		

Managing the data by using Azure Storage Explorer and AzCopy

Azure Storage Explorer

Azure Storage Explorer is a standalone application that makes it easy to work with Azure Storage data on Windows, macOS, and Linux operating systems.You can access multiple accounts and subscriptions and manage all your storage content with Azure Storage Explorer.

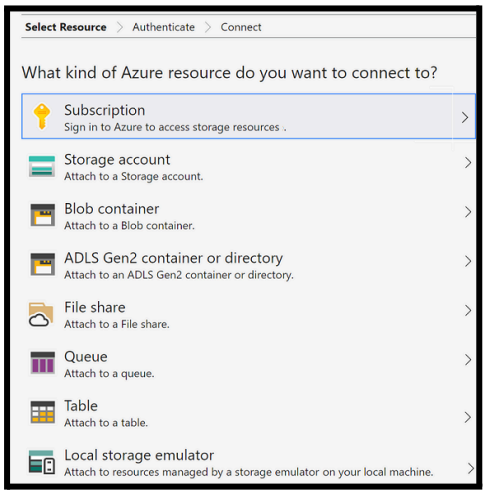


(Source: Microsoft Documentation)

Azure Storage Explorer has the following characteristics

Azure Storage Explorer requires both management (Azure Resource Manager) and data layer permissions to allow full access to your resources. You need Microsoft Entra ID permissions to

access your storage account, the containers in your account, and the data in the containers. Azure Storage Explorer lets you connect to different services : Subscriptions, Storage Account, Blob Container, Microsoft Entra ID, Fileshare, Queues, and Tables etc.



(Source: Microsoft Documentation)

Azure Storage Explorer uses the AzCopy tool for all its data transfers.

If you prefer to use a graphical UI to work with your files, you can use Azure Storage Explorer and reap the performance benefits of AzCopy.

Azure Storage Explorer uses your account key to perform operations.

After you sign in to Azure Storage Explorer, you don't need to provide your credentials again.

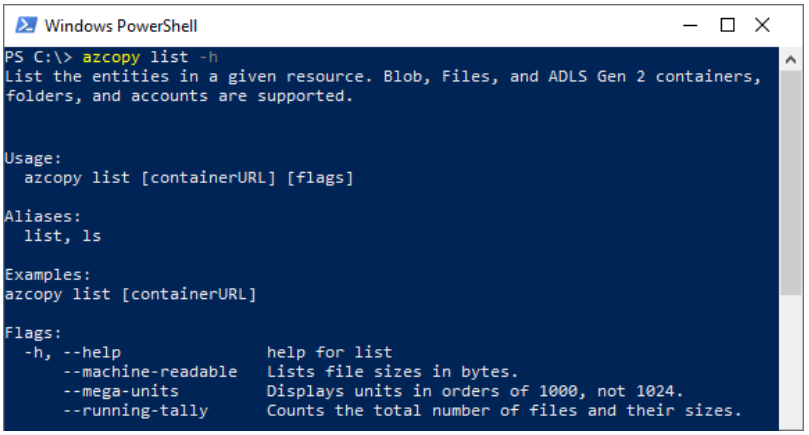
AzCopy:

This is a command-line utility that you can use to copy blobs or files to or from a storage account. This article will help you download AzCopy, connect it to your storage account, and then transfer the data and

Current version: 10.15.0

Authenticate options for AzCopy : Azure AD, and SAS tokens			Things to consider when using AzCopy
Storage type	**Currently supported method of authorization**		**Data synchronization**
Blob storage	Azure AD & SAS		Job management
Blob storage (hierarchical namespace)	Azure AD & SAS		Transfer resiliency
File storage	SAS only		Fast account to account copy

For your information: You access the AzCopy tool by using the CLI in a console or terminal window. The tool provides a simple self-publishing syntax (refer the image below)



(Source: Microsoft Documentation)

Implement Azure Storage Redundancy

Storage can be replicated for availability. Here are the options:

- **Locally redundant storage (LRS)** – 3 copies stored in a single Datacenter. Single point of failure if the data center is unavailable. Cheapest option.
- **Zone-redundant storage (ZRS)** – 3 copies in 3 zones in the primary region. Also recommended to replicate to the secondary region.
- **Geo-redundant storage (GRS)** – Here, the secondary copies are stored in another region, which protects us against a region-wide outage. Basically, it is LRS plus an additional copy in a secondary region. The primary copy process is Synchronous, while it is asynchronous for secondary.
- **Read-access geo-redundant storage (RA-GRS)** – Compared with GRS, the secondary copy will also be available only for READ access.
- **Geo-zone-redundant storage (GZRS)** – Here, it is the same as LRS except that the secondary copy will be in a zone in another region, which is the twin region of our primary region. Basically, it is ZRS plus a single copy in the secondary region. The primary copy process is Synchronous, while it is asynchronous for secondary.
- **Read-access geo-zone-redundant storage (RA-GZRS)** – Same as GZRS, except that you will be able to read data from your secondary region also. *(If it is not RA, then we need to remember that data is available but not readable until Microsoft fails over to the secondary region in case of a regional failure or if we manually failover)*

	LRS	ZRS	GRS	RA-GRS	GZRS	RA-GZRS
Node	✓	✓	✓	✓	✓	✓
Datacenter/zone	×	✓	✓	✓	✓	✓
Region	×	×	✓	✓	✓	✓
Read-access	×	×	×	✓	×	✓

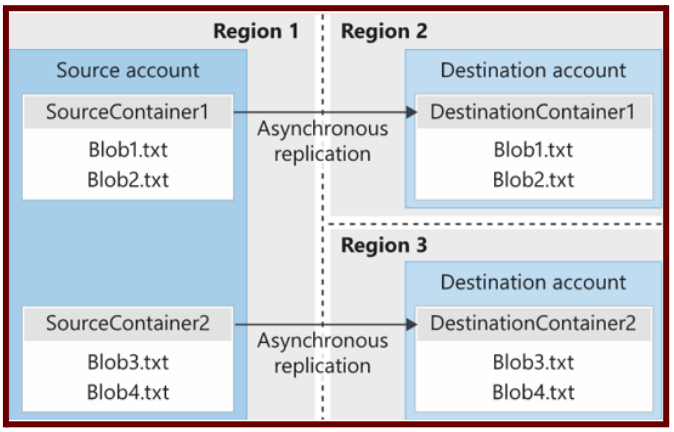
SLA	99.90%	99.90%	99.90%	99.90%	99.90%	99.90%
Durability	11 9's	12 9's	16 9's	16 9's	16 9's	16 9's

Object Replication Configuration

Object replication asynchronously copies blobs in a container according to policy rules that you configure. The following contents are copied from the source container to the destination container during the replication process.

- The blob contents
- The blob metadata and properties
- Any versions of data associated with the blob

Below is an example of asynchronous replication of blob containers between regions.



(Source: Microsoft Documentation)

Things to know about blob object replication

Object replication requires blob versioning to be enabled on both the source and destination accounts.
Object replication does not support blob snapshots
Object replication is supported when the source and destination accounts are in the hot or cool tier.
When you configure object replication, you create a replication policy that specifies the source Azure storage account and the destination storage account.
A replication policy includes one or more rules that specify a source container and a destination container.

Things to consider when configuring blob object replication

- Latency reductions,
- Efficiency for compute workloads,
- Data distribution,
- costs benefits

C) Configure Azure Files and Azure Blob Storage

Azure File Storage

File Storage:

1. It is one of the 4 storage solution offerings by Azure.
2. One of the best use cases is the offering of fully managed file shares.
   a. The file share is accessible over **Server Message Block (SMB)** protocol or **Network file system (NDS)** protocol.
   b. Can mount Azure file shares either on Cloud or on-premises.
   c. SMB file shares are accessible from Windows, Linux, and MacOS, whereas NFS file shares are accessible over Linux or MacOS clients.
3. The file share concept can be extended to caching on Windows Servers with Azure file Sync. This allows for fast access closer to the location it is being used.

Use Cases:

➔ The Company has headquarters in New York and a branch office in California. Users in California are seeing latency accessing the data which is created in New York.

- ◆ **Solution** – Use Azure File Sync, which will cache the data closer to the California location.

➔ The Company wants to migrate its application. The application has data residing on file shares mounted.

- ◆ **Solution** – Use Azure files for Lift and Shift scenarios. Create a file share and mount it as a drive, and the application can be migrated and will point to this file share mounted as a drive.

➔ One of the clients wants high availability, has had an issue with file servers being down often.

- ◆ **Solution** - Use File shares. If a server crashes, place a new Server, and it will automatically get the data from the cloud with Azure File Sync setup

FAQs

- **What ports does file share use?**
  a. SMB protocol uses 445
  b. NFS protocol uses 2049

- **How do we back up Azure file shares?**
  a. Please take snapshots.

- **What versions of SMB are there, and what to choose?**
  a. SMB 2.0 and SMB 3.0 are mostly used
  b. SMB 3.0 is the preferred version since it provides encrypted access.
  c. If a client does not support SMB 3.0, downgrade to SMB 2.0

- **Can I use Import/Export Service with Azure files?**
  a. You can import into Azure files, but you cannot export from Azure files. With Blobs, you can import and export.
- **There is a requirement to use Azure files for IO intensive workloads like hosting Databases and HPC. Is this possible?**

a. Yes, please use Premium file shares as they are stored on SSD. Please note that replication has to do with the LRS only.

- **Is the storage unlimited, or are there limitations?**
  a. Azure files work with Quotas. When you create a file share, you need to specify a quota like 100GB. You can alter if needed.

Tips

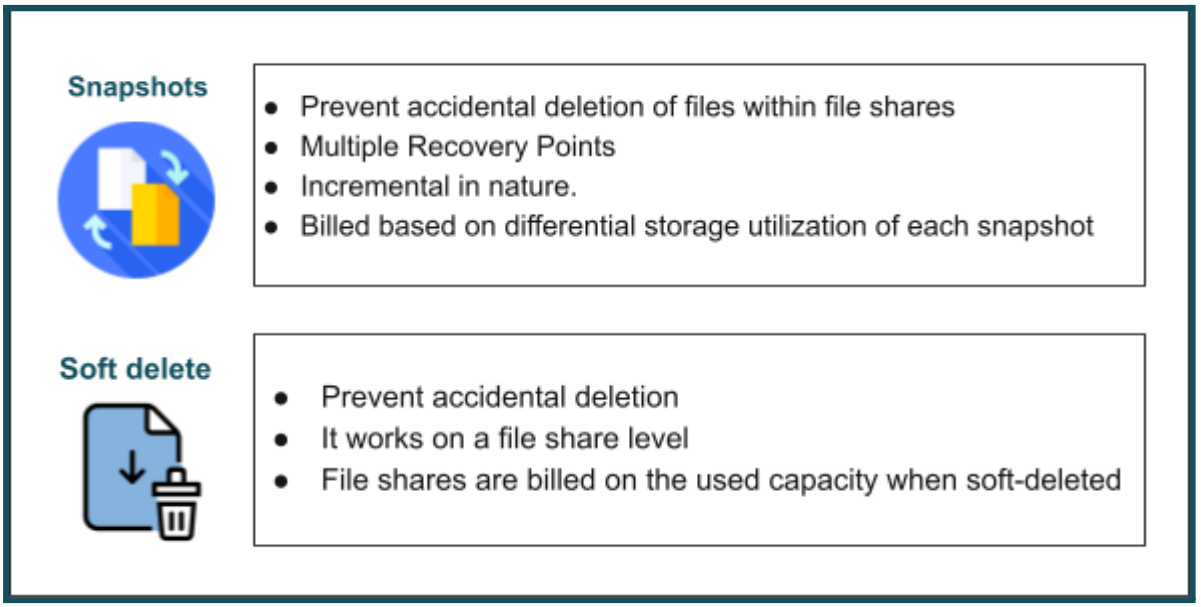
- **Can I use SAS to map a drive?**
  a. It is possible to map a drive with SAS.

- **Can we provide share level permissions? What are inbuilt roles?**
  a. *Storage File Data SMB Share Reader* – Allows READ access
  b. *Storage File Data SMB Share Contributor* – Allows read, write, delete access
  c. *Storage File Data SMB Share Elevated Contributor* - Allows read, write, delete, and modify Windows ACLs.

Configure snapshots and soft delete for Azure Files

Azure Files will have a few tools to protect your data, those are as follows

1. Soft delete,
2. Snapshots,
3. Azure Backup
4. Azure File Sync.



For more information, please refer to the following links

- Enable soft delete on Azure file shares
- Prevent accidental deletion of Azure file shares.
- Enable soft delete for Azure Files | Microsoft Learn

For more information, please refer to the following links

- Overview of share snapshots for Azure Files.
- Use Azure Files share snapshots | Microsoft Learn

Azure BLOB

Azure Storage has 5 types:

Azure Blob storage	Used to store Binary/Text data
Azure File storage	File Shares
Azure Disk Storage	Persistent data storage
Azure Queue storage	Messaging Store and Queuing
Azure Table storage	NoSQL Datastore

Blob Storage:

- Scalable and It can be used for DR purposes
- Use REST API, CLI, ARM template to create a storage account
- Blob is typically a file, can be image, file, video
- Common scenarios – backup/restore, upload large files, logging
- New version of ADLS (Azure Data Lake Storage) is built on top of Blob called ADLS Gen2
- Endpoint for Blobs is https://*.blob.core.windows.net
- For a blob, the base URI includes the name of the account (myaccount), the name of the container(mycontainer), and the name of the blob(myblob).
  Here name will as follows: https://myaccount.blob.core.windows.net/mycontainer/myblob
- You can use Storage Explorer to view/upload/copy files

Limits:

- No limits to the number of objects
- Max size of a single object in a container is about 5TB

Blob types

- **Block Blob** – Large objects that are broken and each block is uploaded in parallel.
  It is optimal for Streaming
- **Append Blobs** – We use these where we keep updating and appending to the files.
  For example, logging.
- **Page Blob** – Stores the VHD VM disks. Max size is 8TB

Access levels

- **Private (no anonymous access)** – This is the default. A valid token is needed to access data.

- **Blob (anonymous read access for blobs only)** – Globally accessible with reading access
- **Container (anonymous read access for containers and blobs)** – All blobs in the container can be read and listed. Access is at the container level, and hence it is for container level, and every blob can be read.

Configure storage tiers

Access Tiers

- There are 3 access tiers – **Hot/Cool and Archive**.
- As you move from Archive to hot, the pricing will go up, and as you move from Hot to Archive, the cost of accessing will go up.
- You need to decide based on how often you access & balance between storage, access cost.
- **Cool** – Use this for more than 30 days but less than 180 days.
- **Archive** – This is for anything accessed for more than 180 days.
  Please note that it will take several hours to access the data.
- To recall, you need to "rehydrate" the blob by changing the access tier to Hot or Cool. This can also be set at blob level only, whereas COOL/HOT is at the account level.

Configure blob lifecycle management

Lifecycle Management

- You can use lifecycle management to move your data from one access tier to another.
- For example, you can move from Hot to Cool after 30 days and then from Cool to Archive after 180 days and then delete after 1 year.

Soft Delete

- If you enable this feature, the blob will not be deleted but will be marked for deletion.
- You specify the number of days, like 90, and after 90 days, the blobs will be deleted.
- This protects against malicious or accidental deletion.
- Please note that you will pay for the 90 days of storage.

Built-in Roles for Blob Storage

Role	Access
Storage Blob Data Contributor	Read, write, and delete Azure Storage containers and blobs.
Storage Blob Data Owner	Provides full access to Azure Storage blob containers and data operations
Storage Blob Data Reader	Read and list Azure Storage containers and blobs.
Storage Blob Delegator	Get a user delegation key, which can then be used to create a

	shared access signature for a container or blob that is signed with Microsoft Entra ID credentials.

For more info —> Configure a lifecycle management policy - Azure Storage

Azure Storage Firewalls and Virtual Networks

- We can have a layered security model and specify the IP addresses from which access will be allowed.
- Also, we can specify Vnets/subnets from where access will be allowed.
- *Time-bound access – SAS Signatures*
- A Storage account key gives complete access to your data.
- If there is a need to provide access for a short/limited period, we can create a **SAS Signature** with a start and end time, and the data can be accessed during that window only.
- We can specify allowed *services/service types/permissions (Read/Write/List etc)/Start and expiry date/time/ Allowed IP address range*

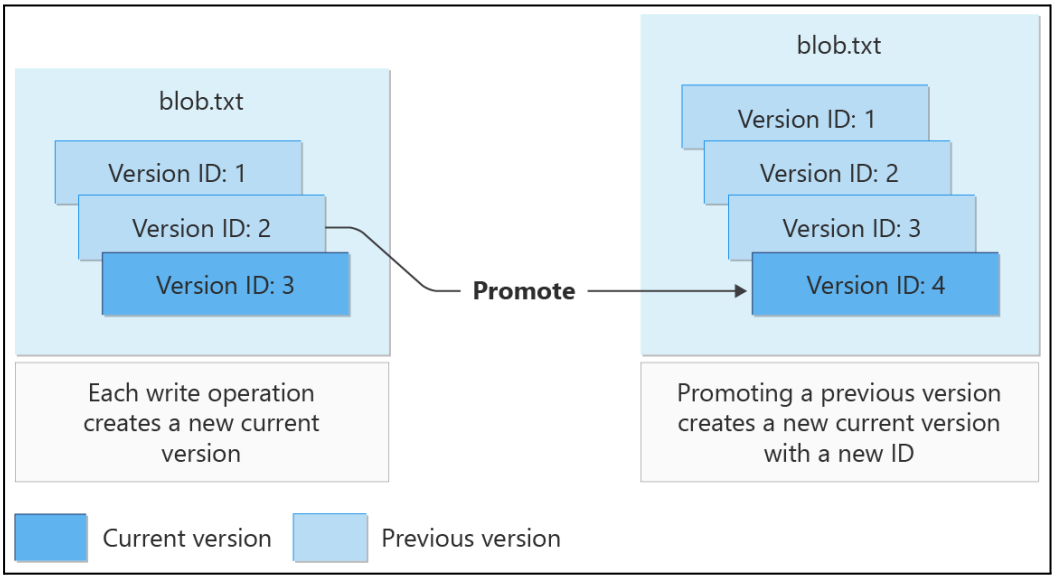
Use Case Scenarios:

- A Company wants to store more than 5TB of data. The cost must be minimized – Solution – Azure Blob Storage using Import/Export Service.
- A Company wants to use Azure Storage. The Data has various usage tiers. Tier 1 – Used regularly and needed immediately in the first 30 days, Tier 2 – Not used after 30 days, Tier 3 – Not used after 180 days, and Tier 4 – Can be deleted after 1 year. – Solution – Implement Lifecycle Management
- A Company plans to move 500MB of data to Azure Blob. What is the best Method – Solution – Download Storage Explorer (or use Storage explorer on the portal) with SAS and transfer data
- When creating a storage account, what tiers can we choose – Hot, Cool, Archive. Answer – Hot and Cool only. Archive Tier is at Blob level only.
- You want to protect your storage account against accidental deletion. What do you do? Solution – Enable Soft Delete
- With Soft delete enabled, a file is deleted. 2 snapshots are also deleted. What can be recovered? Answer – The snapshots and file can be restored

Azure Blob Versioning

Azure Blob Versioning is a feature that helps you maintain previous versions of your blobs (files) automatically. This can be crucial for data protection, allowing you to recover from accidental deletions or modifications.

The following diagram shows how versions are created on write operations, and how a previous version may be promoted to be the current version:



Source: [Blob versioning - Azure Storage | Microsoft Learn]

Key Features:

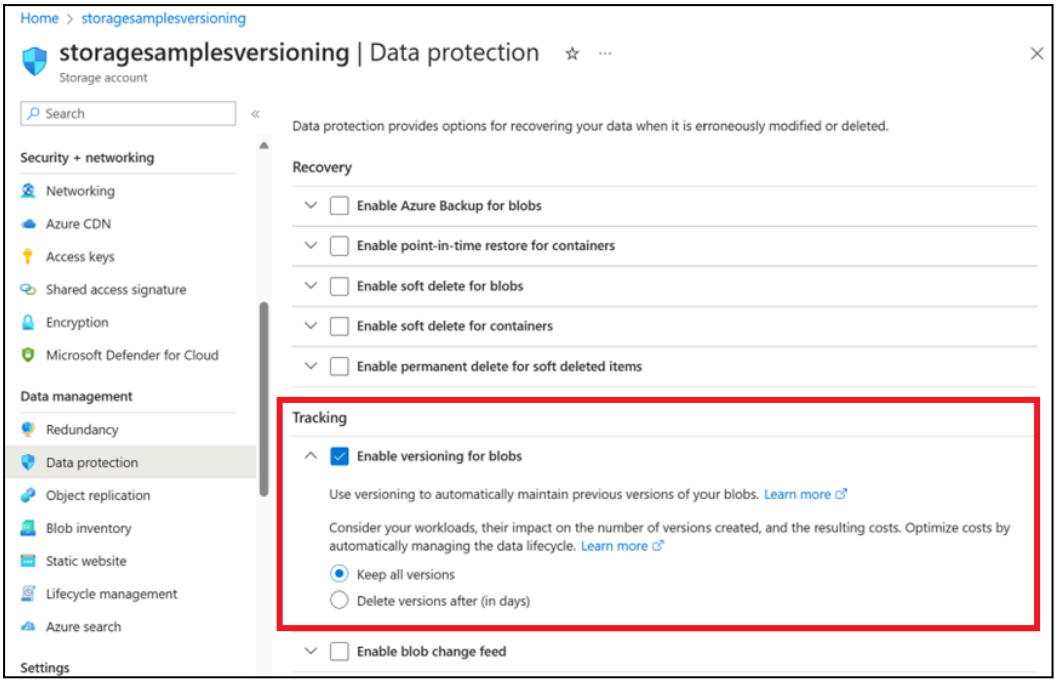
- **Automatic Versioning**: When enabled, Azure Blob Storage automatically creates a new version of a blob each time it is modified or deleted.
- **Version ID**: Each version is identified by a unique version ID, which is a timestamp of when the version was created.
- **Immutable Versions**: Blob versions are immutable, meaning once a version is created, it cannot be modified [It means you cannot modify the content or metadata of an existing blob version].
- **Data Recovery**: You can restore a previous version of a blob to recover data if it is erroneously modified or deleted.

Enable blob versioning

You can enable a blob version with the Azure portal, PowerShell, Azure CLI, or an Azure Resource Manager template.
To enable blob versioning for a storage account in the Azure portal:

1. Navigate to your storage account in the portal.

2. Under Data management, choose Data protection.
3. In the Tracking section, select Enable versioning for blobs, and then select whether to keep all versions or delete them after a certain period.



Source: Enable and manage blob versioning - Azure Storage | Microsoft Learn

Deploy and manage Azure Compute Resources

A) Automate deployment of resources by using templates

ARM Template

Azure Resource Manager templates(ARM Templates) are JavaScript Object Notation (JSON) files that define the infrastructure and configuration for your project.

Ref: ARM template documentation | Microsoft Learn

To implement infrastructure as code(IaC) for Azure solutions, you need to use ARM templates.

This template uses declarative syntax, which allows you to state what you want to execute without having to write a sequence of programming commands to create it.

In the template, specify the resources you want to implement and the properties associated with those resources.

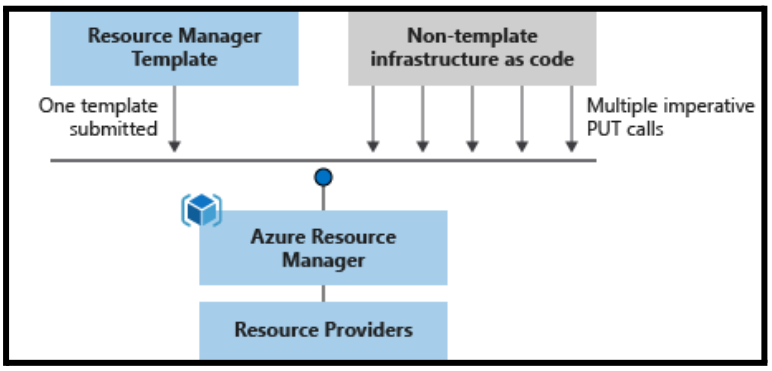
Why do we need to choose ARM templates?

The below are a few important points
Declarative syntax,Repeatable results, Modular files
Create any Azure resource, Extensibility, Testing
Extensibility, Testing, Preview changes, Built-in validation
Tracked deployments, Policy as code, Deployment Blueprints
CI/CD integration, Exportable code, Authoring tools

If you want to deploy a template, you can use any of the following options:

- ➔ Azure portal
- ➔ Azure CLI
- ➔ PowerShell
- ➔ REST API
- ➔ Button in GitHub repository
- ➔ Azure Cloud Shell

Orchestration- (Source: Microsoft Documentation)

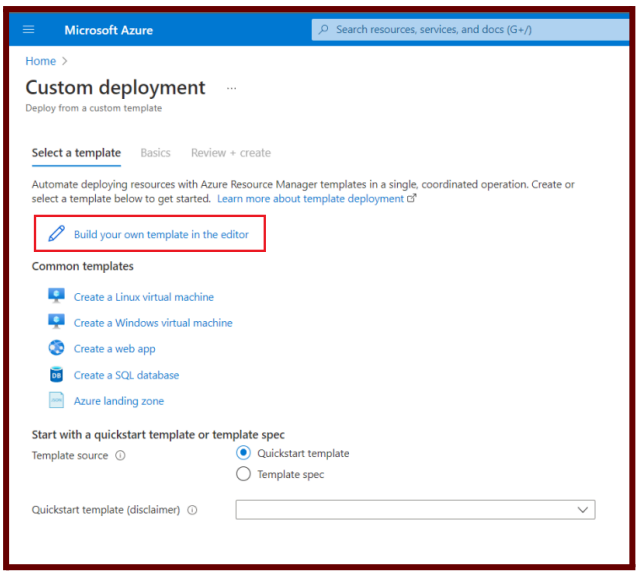


Editing and Deploying a Template

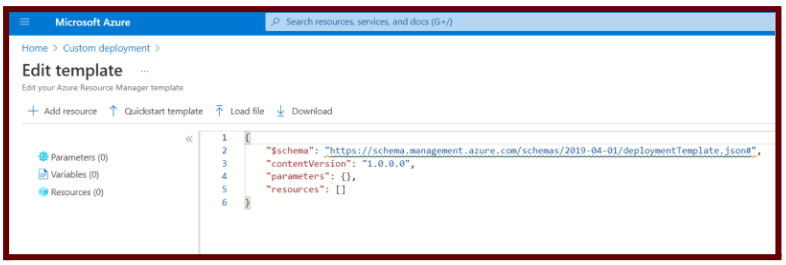
We can use the Azure portal to quickly develop and deploy the ARM templates and in general, Microsoft recommends → Visual Studio Code to develop your ARM templates and Azure CLI or Azure PowerShell to deploy the template, but here we can use the portal for quick deployments without installing those tools.

Let's assume that you have an ARM template that you want to run one time without setting up other tools. Steps are as follows

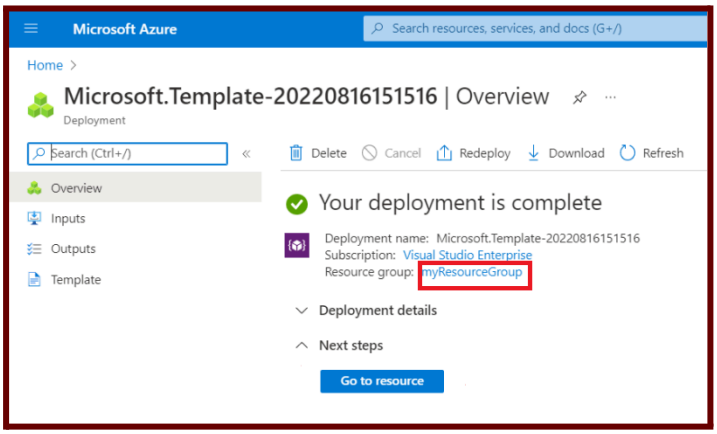
1. Again, select Deploy a custom template in the portal.
2. This time, select Build your own template in the editor.



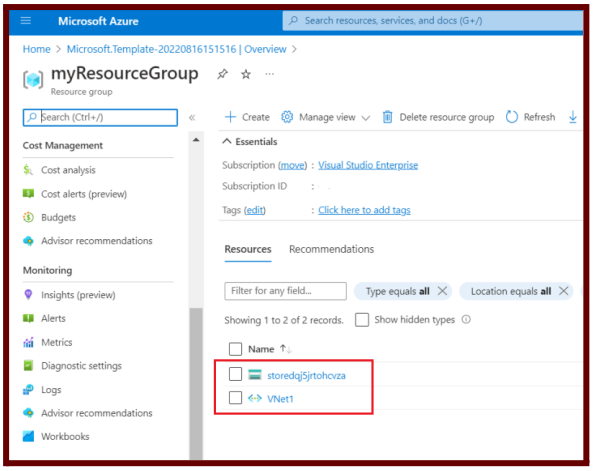
3. You see a blank template.



4. Replace the blank template with the following template.
5. It deploys a virtual network with a subnet → then Select Save Option.
6. You see the blade for providing deployment values. Again, select myResourceGroup for the resource group. You can use the other default values. When you're done providing values, select Review + create.
7. After the portal validates the template, select Create.
8. When the deployment is complete, you will see the deployment status.
   This time select a resource group name.



9. Notice that your resource group now contains a storage account and a virtual network.



Pls Refer to the below links for more information

- Deploy template - Azure portal - Azure Resource Manager | Microsoft Learn
- Tutorial - Create and deploy template - Azure Resource Manager | Microsoft Learn

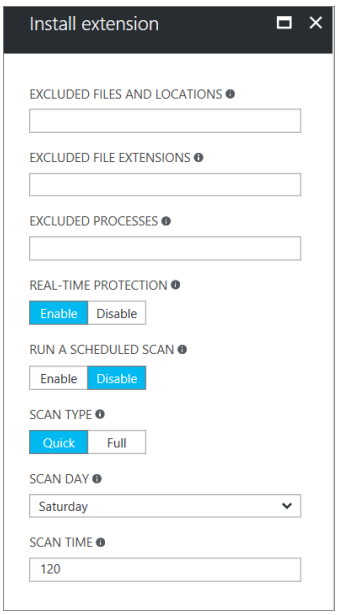
Virtual machines extensions

- Creating and managing virtual machines can be difficult and burdensome.
- Also many of the maintenance tasks are repetitive and time consuming. To solve this there are several ways to automate the tasks of creating, managing and removing virtual machines.
- Virtual machine extension is one way to meet our needs.
- Extensions are small apps that provide post-deployment configuration & automation on VMs.
- The Azure platform hosts a number of extensions that cover VM configuration, monitoring, security, and utility applications.

Example: Consider a scenario where a virtual machine needs software installation or anti-virus protection, or when a machine configuration script needs to be run.
You can use virtual machine extensions to accomplish these tasks. **Extensions are all about managing your virtual machines.**

You can apply VM extensions to an existing VM through the Azure portal. Select the VM in the portal, select Extensions, and then select Add. Select the extension you want from the list of available extensions and follow the instructions in the wizard.

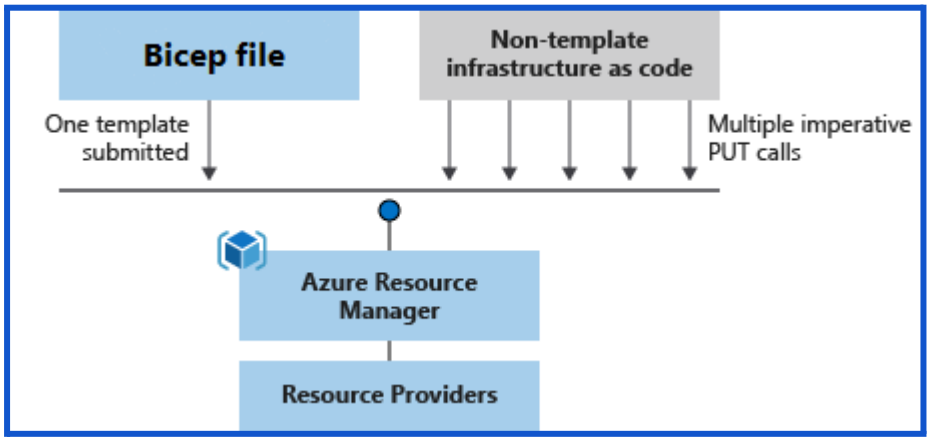
Below image shows the installation of the Microsoft Antimalware extension from the Azure portal:



Fore more information → [Virtual machine extensions and features for Windows]

Bicep File Overview

Bicep is a domain-specific language (DSL) for deploying Azure resources declaratively. It simplifies the authoring of Azure Resource Manager (ARM) templates by providing a more readable and maintainable syntax.



Source: [What is Bicep? - Azure Resource Manager | Microsoft Learn]

Key Features:

- **Declarative Syntax**: Bicep uses a declarative syntax to define Azure resources, making it easier to read and write compared to JSON-based ARM templates.
- **Modularity**: You can break down complex deployments into smaller, reusable modules, promoting better organization and reuse of code.
- **Type Safety**: Bicep provides type safety and validation at compile time, reducing errors and improving reliability.
- **Tooling Support**: Bicep integrates with Visual Studio Code, offering features like IntelliSense, syntax highlighting, and error checking.
- **Automatic Conversion**: You can decompile existing ARM templates into Bicep files, facilitating the transition from JSON to Bicep.

Basic Structure:

A Bicep file typically includes:

- **Parameters**: Define inputs to the deployment.
- **Variables**: Store values that can be reused throughout the file.
- **Resources**: Declare the Azure resources to be deployed.
- **Outputs**: Specify values to be returned after deployment.

Example:

Here's a simple example of a Bicep file that deploys an Azure Storage account:

```
param storageAccountName string
param location string = resourceGroup().location

resource storageAccount 'Microsoft.Storage/storageAccounts@2021-04-01' = {
  name: storageAccountName
  location: location
  sku: {
    name: 'Standard_LRS'
  }
  kind: 'StorageV2'
}

output storageAccountName string = storageAccount.name
```

Benefits:

- **Simple Syntax**: Easier to write and understand compared to JSON.
- **Improved Maintainability**: Modular structure and reusable components.
- **Enhanced Tooling**: Better development experience with integrated tools.
- **Support for all resource types and API versions**: Immediate support for Azure services [for both preview and GA versions]
- **Modularity:** You can use modules to break up your bicep code into manageable chunks.

Comprehensive **Support for all Azure service resource types** and API versions, alongside **modularity** for organizing your Bicep code into manageable segments.

- Modularity: You can use modules to break up your bicep code into manageable chunks.
- **Azure services integrations:** Bicep integrates with Azure services[Azure Policy, template specs, and Azure Blueprints].
- **Preview updates/Changes:** You can use the what-if operation to preview the changes before running the Bicep file.
- **No state or state files to manage:** All state is stored in Azure. Users can contribute and be confident that their updates are performed as expected.
- **No Cost and Open Source:** Since Bicep is completely free, you don't need to pay for premium capabilities. It is also supported by Microsoft support.

[Tutorial - Create and deploy template - Azure Resource Manager | Microsoft Learn]

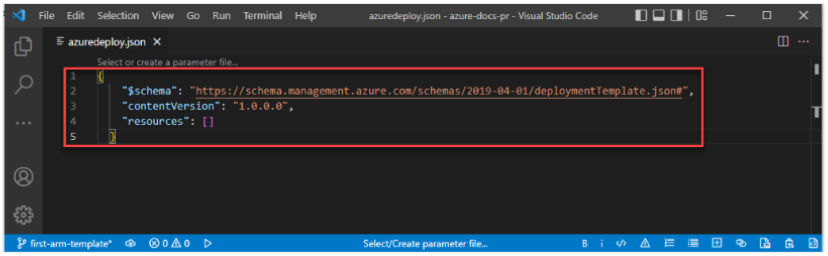
Create your first template

1. Open Visual Studio Code with the installed ARM processor Tools extension.
2. From the **File** menu, select **New File** to create a new file.
3. From the **File** menu, select **Save As**.
4. Name the file *azuredeploy* and select the *json* file extension. The complete name of the file is *azuredeploy.json*.
5. Save the file to your workstation. Select a path that's easy to remember because you need to provide that path later when deploying the template.
6. Copy and paste the following JSON into the file:

JSON Copy

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "resources": []
}
```

Here's what your Visual Studio Code environment looks like:



[Deploy resources with Azure CLI and Bicep files - Azure Resource Manager | Microsoft Learn]

Deploy local Bicep file

You can deploy a Bicep file from your local machine or one that is stored externally. This section describes deploying a local Bicep file.

If you're deploying to a resource group that doesn't exist, create the resource group. The name of the resource group can only include alphanumeric characters, periods, underscores, hyphens, and parenthesis. It can be up to 90 characters. The name can't end in a period.

Azure CLI Copy Open Cloud Shell

```
az group create --name ExampleGroup --location "Central US"
```

To deploy a local Bicep file, use the `--template-file` switch in the deployment command. The following example also shows how to set a parameter value.

Azure CLI Copy Open Cloud Shell

```
az deployment group create \
  --name ExampleDeployment \
  --resource-group ExampleGroup \
  --template-file <path-to-bicep> \
  --parameters storageAccountType=Standard_GRS
```

The deployment can take a few minutes to complete. When it finishes, you see a message that includes the result:

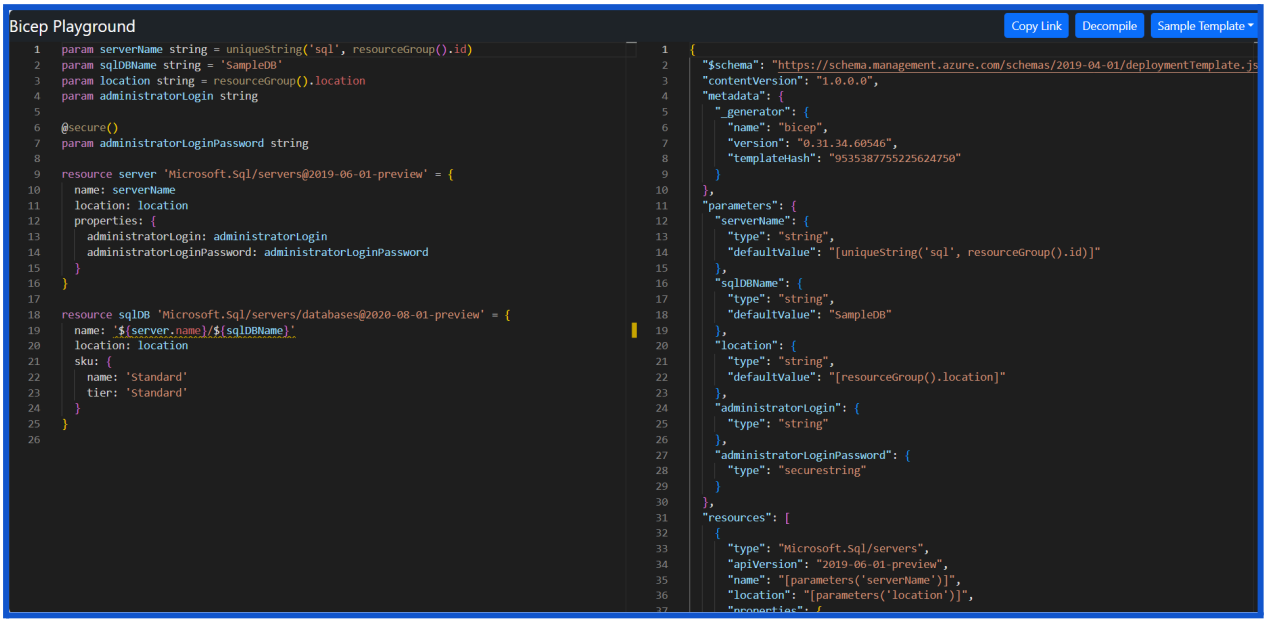
Output Copy

```
"provisioningState": "Succeeded",
```

Comparing JSON and Bicep for templates

Bicep Playground allows you to view Bicep and equivalent JSON side-by-side. You can compare implementations of the same infrastructure.

For example, you can see the file to run SQL Server and Database.
Bicep is half the size of the ARM template. [Bicep Playground 0.31.34-gec82b47d63]



To Modify an existing Bicep file, please refer to the below sources to get more information
[Reference existing resource in Bicep - Azure Resource Manager | Microsoft Learn]
[Create or update Azure custom roles using Bicep - Azure RBAC | Microsoft Learn]

Export a deployment as an ARM template or compile a deployment as a Bicep file

Decompiling the ARM template will help you start bicep development. If you have a library of ARM templates and want to use Bicep for future development, you can decompile them to Bicep.

To export a deployment as an ARM template or compile a deployment as a Bicep file, please refer to this ⟶ [Decompile ARM template JSON to Bicep]

If we want to move the Azure VM from one **subscription or region to another please refer to the below sources**

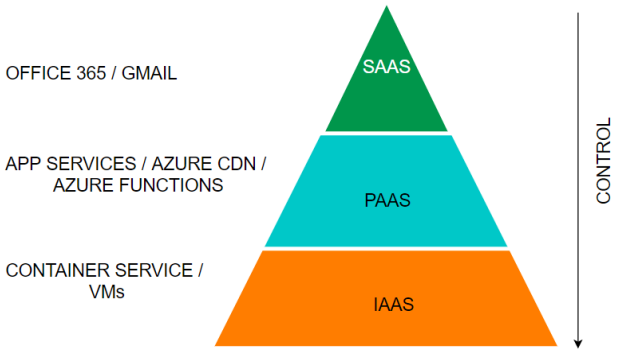
- **[Move Azure VMs across regions with Azure Resource Mover]**
- **[Special cases to move Azure VMs to new subscription or resource group]**

B) Create and configure Azure Virtual Machines

Azure Virtual Machines Introduction

There are 3 major delivery models when it comes to Cloud services. They are:

1. **SAAS – Software as a Service**
2. **PAAS – Platform as a Service**
3. **IAAS - Infrastructure as a Service**



- Azure Virtual Machines are part of the **IAAS** offering from Azure.
- As customers, we are responsible for managing the virtual machine, and just the hardware will be provided to us by the cloud provider. We can *start, stop and delete* the virtual machine.
- If we find that the capacity is insufficient or too high, we can change to a different machine type. We can install any software as we like.
- Also, please note that this is the most expensive of the three offerings.
- We can create **Windows or Linux VMs**, and there are multiple locations throughout the world where resources can run from.
- When we create a VM, we need to attach a virtual hard disk, and the location that we specify is where the hard disks are stored.

Here is the SLA table:

S NO	VM	Disk	SLA
1	2 or more VMs across 2 or more AZs		99.99% at least 1 VM
2	2 or more VMs in a same Availability Set		99.95% at least 1 VM
3	Single VM	Premium or Ultra disk for all disks	99.9%
4	Single VM	Standard SSD	99.5%
5	Single VM	Standard HDD	95%

Please see below details for VM types:

S No	Type	Sizes	Short Description	Best for
1	GP (General Purpose)	B, Dsv, Dasv, Dav, Av2, DC, Dsv	Balanced CPU to memory	Testing/ Dev, small DB, low traffic servers
2	Compute Optimized	F, Fs, Fsv2	High CPU to memory	Medium traffic servers, batch processes, app servers
3	Memory-Optimized	Esv, Ev, Eav, Mv2, M, DSv2 , Dv2	High memory to CPU ratio	RDBMS servers
4	Storage Optimized	Lsv2	High disk throughput and IO	Big data/ DB warehousing/ Large DB
5	GPU	NC, NCv2, ND, NV	Specialized VMs for heavy graphics	Model training with deep learning
6	HPC (High-performance Compute)	HB, HBv2, HC, H	Fastest and most powerful CPU	Real-time processing

Creating an Azure Virtual Machine

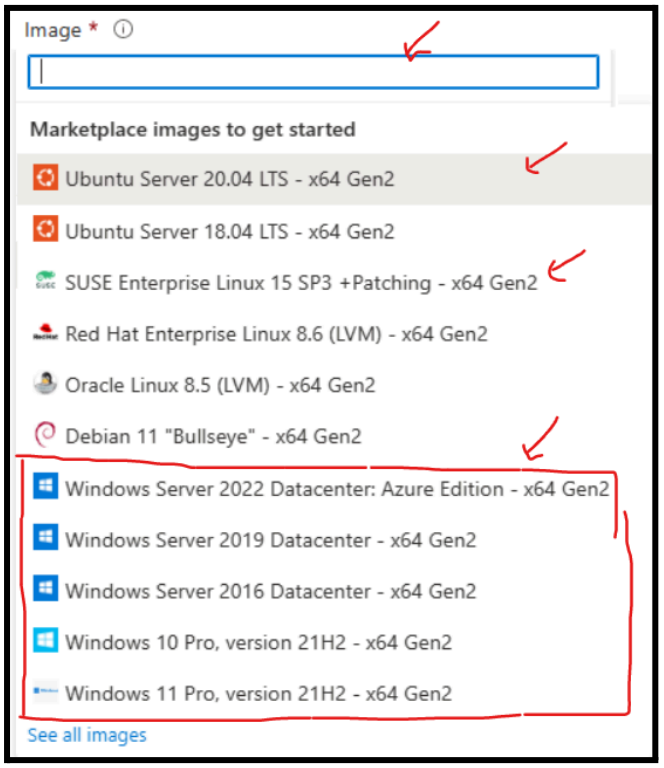
You can create and deploy the Virtual machines in different methods, Main are

→ Through Azure Portal

→ Through The Azure PowerShell/CLI

→ Through ARM Templates

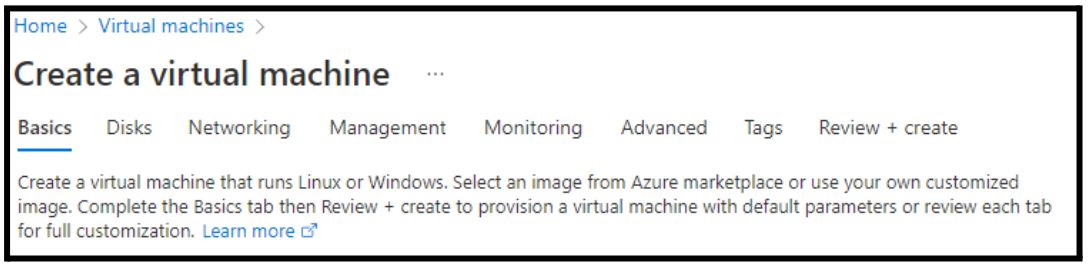
When you create virtual machines (VMs) in the Azure portal, one of your first decisions is to specify which image to use. Azure supports Windows and Linux operating systems and has server and client platforms to choose from. You can also search for other supported images(They are similar to Windows and linux OS) in the Azure Marketplace:



(Source: Microsoft Documentation)

We can create virtual machines by using Azure portal

You can use the Azure portal for reference of the process for creating the virtual machine image.



(Source: Microsoft Documentation)

The process includes configuring basic and advanced options and specifying details about disks, virtual networks, and machine management.

The below are the main key-takeaways for creating a virtual machine.

- The **Basics** tab contains the project details, administrator account, and inbound port rules.
- On the **Disks** tab, you select the OS disk type and specify your data disks.
- The **Networking** tab provides settings to create virtual networks and load balancing.
- On the **Management** tab, you can enable auto-shutdown and specify backup details.
- On the **Advanced** tab, you can configure agents, scripts, or virtual machine extensions.
- Other settings are available on the **Monitoring** and **Tags** tabs.
- If the validation is completed then we need to click the "**Review+Create**" tab for final output.

Reasons to move Azure VMs

Azure VM is already deployed in one region and a new region support is added that is closer to the end users of your application or service. In this scenario,

An Azure Virtual Machine is already deployed in the region and one more new region support is added which is closer to end users of the applications or services.

Let's assume a scenario as follows:

- You want to move your VMs to a new region to reduce latency. Use the same procedure if you want to consolidate memberships or have governance or organization rules you need to move.
- Your VM is deployed as a single-instance VM or as part of an availability set. If you want to increase the availability of SLAs, you can move your VMs into an availability zone.

Azure-managed disk types

Managing virtual machine disks in Azure involves several key tasks, including creating, attaching, resizing, and managing the performance of disks. Here are some important aspects to consider:

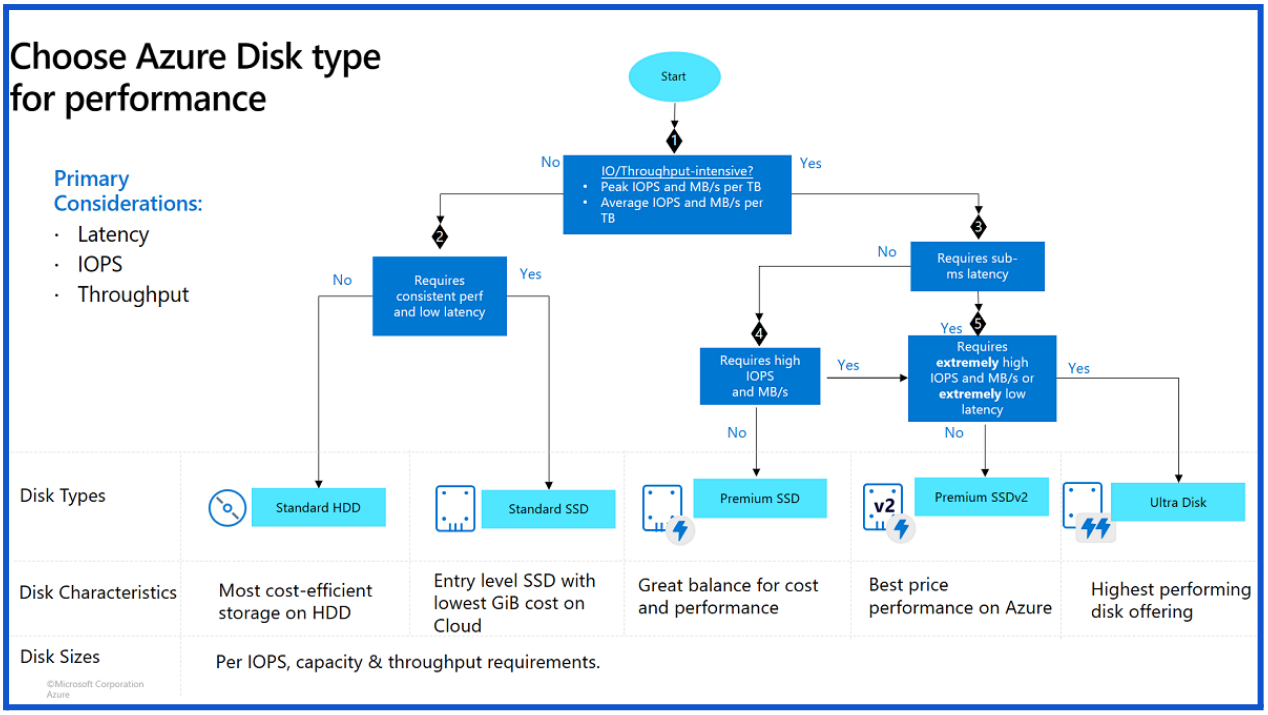
Types of Azure Managed Disks

Azure offers several types of managed disks, each suited for different workloads:

- **Ultra Disks**: High-performance, low-latency disks ideal for data-intensive applications.
- **Premium SSDs**: Suitable for production workloads requiring high IOPS and low latency.
- **Standard SSDs**: Cost-effective storage for less critical workloads.
- **Standard HDDs**: Best for backup and non-critical data.

Key Management Tasks

1. Creating and Attaching Disks:
    - You can create and attach data disks to your VMs using the Azure portal, Azure CLI, or Azure PowerShell
    - After attaching, you need to initialize and format the disk within the VM.
2. Resizing Disks:
    - Disks can be resized to accommodate growing data needs. This can be done without downtime
3. Snapshots and Backup:
    - Snapshots provide a point-in-time backup of your disks. They can be used to restore data or create new disks.
    - Azure Backup can automate the backup process for managed disks.
4. Performance Management:
    - Choose the right disk type based on your workload requirements to ensure optimal performance.
    - Monitor disk performance and adjust configurations as needed.



Source: [Select a disk type for Azure IaaS VMs - managed disks]

Tools for Management

- **Azure Portal**: Provides a graphical interface for managing disks.
- **Azure CLI and PowerShell**: Offer command-line tools for scripting and automation.
- **Azure Storage Explorer**: A tool for managing disks, including uploading and downloading VHDs.

Manage virtual machine disks

1. Azure Portal

- **Create a New Disk:**
  1. Navigate to the Azure portal.
  2. Search for and select "Disks."
  3. Click "Create disk."
  4. Specify the disk name, size, location, and type.
  5. Click "Create."
- **Attach a Disk to a VM:**
  1. Navigate to the VM's overview page.
  2. Click "Disks."
  3. Click "Create and attach a new disk."
  4. Select the desired disk from the list or create a new one.
  5. Specify the LUN (Logical Unit Number) for the disk.
  6. Click "OK."

- **Detach a Disk from a VM:**
    1. Navigate to the VM's overview page.
    2. Click "Disks."
    3. Select the disk to be detached.
    4. Click "Detach."

Azure Disk Encryption(ADE) for Windows Virtual Machines(VMs)

Azure Disk Encryption helps protect and secure your data while meeting your organizational security and compliance commitments. It uses Windows' BitLocker feature to provide volume encryption for OS and data disks of Azure virtual machines (VMs) and integrates with Azure Key Vault to help you control and manage disk encryption keys and secrets.

ADE is Zone resilient, similar to VMs. FYI→Azure Services that support Availability Zones.

If you use Microsoft Defender for Cloud, you'll get a warning if you have unencrypted VMs. Warnings are shown as high severity and it is recommended to encrypt these VMs.

Supported VMs for ADE : Generations 1 & Generation 2 VMs and also VMs with premium storage.
ADE is not available on Basic, A-series VMs, or on virtual machines with less than 2 GB of memory.

Supported operating systems are

- ➔ Windows client : Windows 8 and later,
- ➔ Windows Server: Windows Server 2008 R2 and later,
- ➔ Windows 10 Enterprise multi-session and later.

Requirements: Networking, Group Policy, and Encryption key storage requirements

Move VMs from one resource group to another

You can move a VM from one resource group to another one with Portal, CLI, and powershell.

Ex: Using Azure Portal

1. Go to the Azure portal to manage the resource group containing the VM to move. Search for and select Resource groups.
2. Choose the resource group containing the VM that you would like to move.
3. At the top of the page for the resource group, select Move and then select Move to another resource group. The Move resources page opens.
4. Select each of the resources to move. In most cases, you should move all of the related resources that are listed.
5. Select an existing resource group or enter a name to create a new resource group.
6. When you're done, select that you understand that new resource IDs will be created and that the new IDs will need to be used with the VM after the move, and then select OK.

You can practice this using with our Azure Cloud Sandbox Environment

Adding a Data disk using Azure Portal

The below are the main steps for adding a data disk to virtual machines

1. Sign in to the Azure portal.
2. Search for and select Virtual machines.
3. Select a virtual machine from the list.
4. On the Virtual machine pane, select Disks.
5. On the Disks pane, select Create and attach a new disk.
6. In the drop-downs for the new disk, make the selections you want, and name the disk.
7. Select Save to create and attach the new data disk to the VM.

You can practice this using with our Azure Cloud Sandbox Environment

Virtual Machine(VM) Availability Options

The below are the Availability Options for the Virtual Machine

- ➔ Availability zones
- ➔ Virtual Machines Scale Sets
- ➔ Availability sets
- ➔ Load balancer
- ➔ Azure Storage redundancy
- ➔ Azure Site Recovery

Here we have provided an overview of the availability options for the Azure VMs.

Azure Virtual Machine Scale Sets:

Azure Virtual Machine Scale Sets allow you to create and manage a group of load-balanced VMs. The number of VM instances automatically increases or decreases in response to demand or a defined schedule.

When you implement virtual machine scale sets and configure all your virtual machines the same way, you get true autoscaling. Virtual machine scale sets automatically increase the number of virtual machine instances as application demand increases and decrease the number of machine instances when demand decreases.

schedule. Scale sets provide the following key benefits:

- Easy to create and manage multiple VMs
- Provides high availability and application resiliency by distributing VMs across availability zones or fault domains
- Allows your application to automatically scale as resource demand changes
- Works at large-scale

For more info refer to this link: Create Virtual Machine Scale Sets - Training | Microsoft Learn

Availability Set

An availability set is a logical grouping of VMs that allows Azure to understand how your application is structured for redundancy and availability. We recommended that two or more VMs be created in an availability set to provide the most available application and meet the 99.95% Azure SLA. There is no cost for an Availability Set, you only pay for each VM instance you create.

Load Balancer

You can Combine Azure Load Balancer with an Availability Zone or Availability Set for maximum application resiliency. Azure Load Balancer distributes traffic between multiple virtual machines. For our standard tier virtual machines, Azure Load Balancer is included. Not all virtual machine tiers include Azure Load Balancer. For more information about load balancing your virtual machines, see Load balancing virtual machines for Linux or Windows.

Azure Storage Redundancy

Azure Storage always stores multiple copies of your data so that it is protected against planned and unplanned events, including temporary hardware failures, network or power outages, and massive natural disasters. Redundancy ensures that your storage account meets its availability and durability goals even in the face of failures.

Azure Site Recovery

Azure Site Recovery helps ensure business continuity by keeping business apps and workloads running during outages. Site recovery reflects workloads running on physical and virtual machines (VMs) from a primary site to a secondary location. In the event of an outage at your primary site, you fail over to the secondary location and access apps from there.

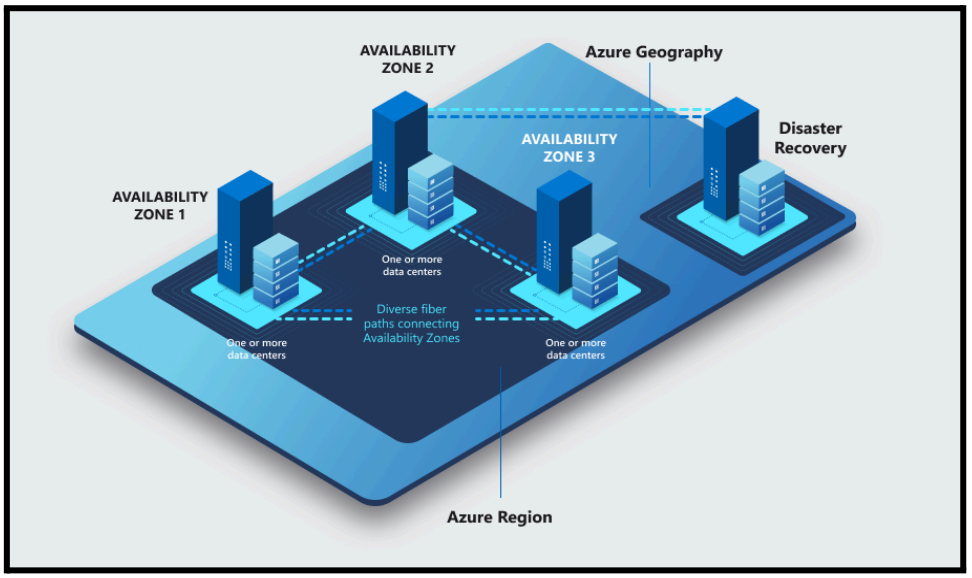
Site Recovery can manage replication for:

- Azure VMs replicating between Azure regions.
- On-premises VMs, Azure Stack VMs, and physical servers.

Availability zones

An **Availability zone** is a physically separate zone in an Azure region and it extends the level of control you have to manage the availability of applications and data in your VMs.
Azure region has three Availability Zones.

Each Availability Zone has a dedicated power source, network and cooling. By designing your solutions to use replicated VMs in zones, you can protect your apps and data from data center loss. If one zone is compromised, replicated apps and data are immediately available in the other zone.



For More info,pls refer to this → [Availability options for Azure Virtual Machines]

Deployment and configuration of Virtual Machine scale sets(VMSS)

Azure Virtual Machine Scale sets provide management capabilities for applications running across multiple VMs, automatic scaling of resources, and traffic load balancing.

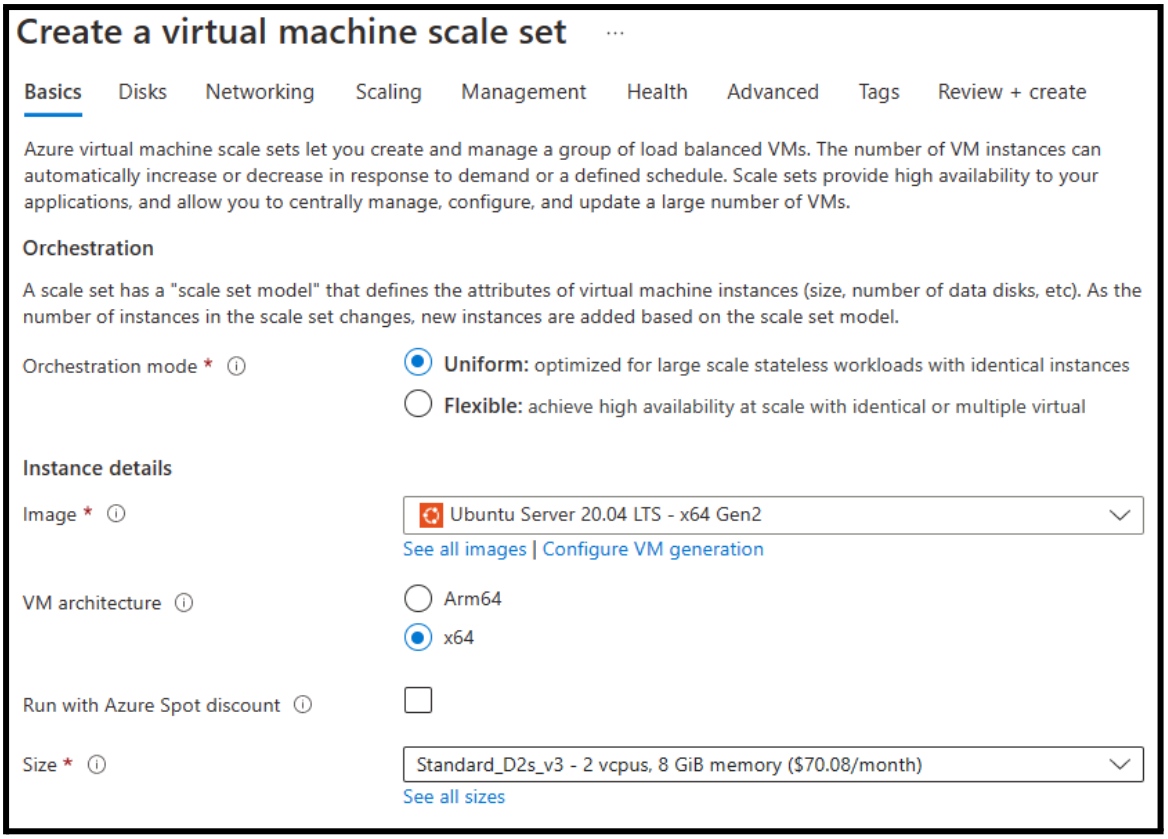
The following are the key benefits VMSS

- Easy to create and manage multiple Virtual Machines
- It Provides high availability and application resiliency
- It Allows your application to automatically scale as resource demand changes
- Works at large-scale

You can deploy Azure virtual machine scale sets in the Azure portal. You specify the number of virtual machines and their sizes, and specify preferences for using Azure Spot instances, Azure managed disks, and provisioning policies.

In the Azure portal, there are several settings to configure to create a deployment of Azure Virtual Machine Scale Sets.

Example: Refer to this → [Create virtual machines in a Flexible scale set using Azure portal]



You can practice this using with our [Azure Cloud Sandbox Environment]

FAQs

1. **How do I resize a VM?**
   You can first run the list-vm-resize-options and see available sizes. If you find the size, you can run the resize command

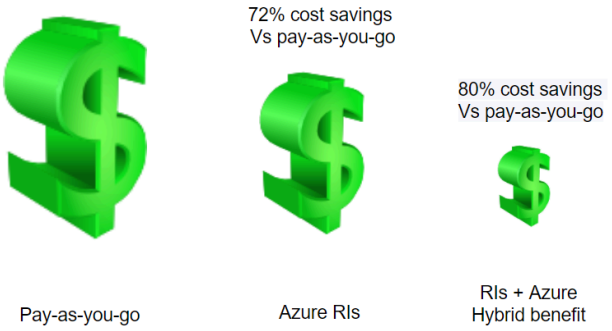
   ```
   az vm resize --resource-group WLRG --name WLVM1 --size Standard_DS3_v2
   ```

   Else you need to deallocate the VM, which will allow you to use any size. You need to deallocate, resize and start a VM.

   ```
   az vm deallocate --resource-group WLRG --name WLVM1
   az vm resize --resource-group WLRG --name WLVM1 --size Standard_DS3_v2
   az vm start --resource-group WLRG --name WLVM1
   ```
2. **What are Azure Dedicated hosts?**
   We usually shared the physical hardware with other tenants. If we want exclusively to use the physical server, then we can choose dedicated hosts.
3. **What are Azure Spot instances?**
   This feature allows us to take advantage of the unused CPU at a significantly lower cost at almost 90% savings. If there are workloads that can tolerate disruption and can be restarted, then we can choose this option. If there is another bidder who bids more than our price, we will be vacated on 30 seconds' notice. So we need to be prepared with

proper scripts to save the data or any other process from exiting gracefully.

4. **How can we save costs on VMs other than Spot instances?**
   There are two other ways we can save on costs.–

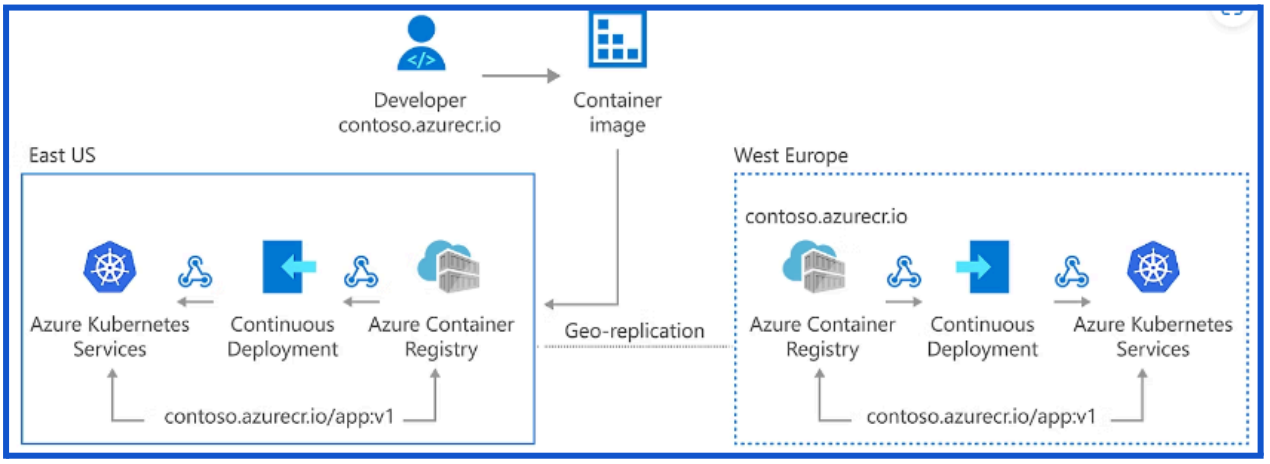


5. **Reserved Instances** – We can commit to 1-year or 3-year and choose to pay upfront or monthly to buy RIs. We have the flexibility to change size if needed.
6. **Azure Hybrid Benefit** – If you have a license already, you can use the license on Azure and get this benefit.
7. **What are Azure Images?**
   If there is a custom image that we want every VM to have when created, we can choose to create a standard VM and sysprep and then create an image. We can then use this image to create VMs.
8. **How can we make VMs highly available?**
   We had discussed in the excel above with SLAs. We can use multiple machines either in availability or in more than 1 availability zone. In addition to this, we can use Azure VMSS (Virtual machine scale sets). VMSS is automatically created from a central configuration using a standard template. More VMs will be added during peak and will be brought down when the demand goes down based on our auto-scaling options.
9. **How can we back up VMs?**
   We have 3 options:
   a. **Azure Backup** – We can create recovery vaults and configure Azure Backup to back up our VMs
   b. **ASR (Azure Site Recovery)** – Here, our VMs are replicated to another region, and our entire production region fails; we can failover to the backup areas with the click of a button
   c. **Managed Snapshots** – If we have managed disks, we can take a snapshot of our disks, a read-only copy. We leveraged this feature for quick backups in dev and test environments.
10. **How can we monitor VMs?**
    Under Monitoring tabs, we have metrics to see various parameters. We can also set alerts. We can also Log analytics by enabling the Logs option in Monitoring. We need to create a log analytics workspace.

C) Create and Configure Containers

Azure container registry [ACR]

Azure Container Registry lets you build, store, and manage container images and artifacts in a private registry for all types of container deployments. Use Azure Container Registries with your existing container development and deployment pipelines.



If you want to learn more about how to create and manage an Azure container registry, please refer to this source link ⟶ Create Registry in Portal - Azure Container Registry.

Introduction to Azure Container Instance (ACI)

- Containerization is the buzzword today. Instead of spinning Physical servers and installing all the dependencies, and installing the application, we can create a container containing all the required dependencies.
- We then package and create an image and deploy it into a container.
- **Docker** is one of the platforms where we can run these containers in the Open source world. Azure has two solutions. One of those is the ACI.
- ACI is a great solution in scenarios where we need to run isolated containers. Examples are simple applications, task automation, and build jobs.
- The drawback of ACI is that it cannot be used for full orchestration like multiple containers, auto-scaling, and coordinated application upgrades. Please consider AKS for such scenarios, which is the other offering from Azure.
- In simple terms, for Production, use **AKS (Azure Kubernetes Service)**, and for simple and isolated containers, use ACI.
- One of the other best use cases for ACI is where we have production issues, and we need to troubleshoot AKS, ACI comes to our rescue where we deploy the trouble-making container in ACI and try to debug.

Advantages of ACI

- *Fast Startup times*

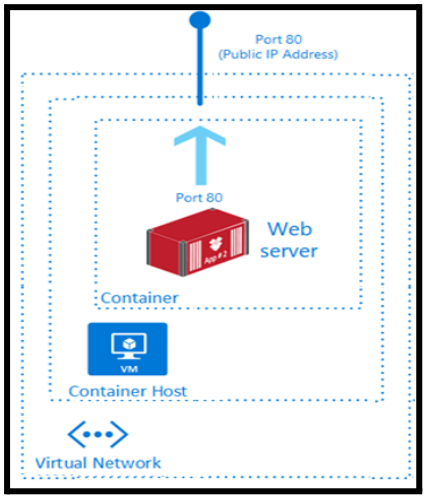
- *Container access*
- *Custom Sizes*
- *Persistent Storage* – We do this by mounting Azure file shares.
- *Virtual Network deployment* – When deployed in a Vnet, ACI can securely communicate with other resources in the Vnet.

Azure Container Instances

Azure Container Instances provide a fast and simple way to run a container on Azure, without having to manage any virtual machines and adopt a high-level service. Azure Container Instances is a great solution for any scenario that can operate in discrete containers, including common applications, task automation, and build jobs.

Containers are becoming the preferred way to package, deploy and manage cloud applications.

The following example shows a web server container built with an Azure Container instance. A container is running in a virtual machine on a virtual network.



Things to know about Azure Container Instances
Fast startup times, Public IP connectivity and DNS names
Hypervisor-level security, Custom sizes, Persistent storage
Linux and Windows containers, Coscheduled groups.
Virtual network deployment

Custom sizes for Azure Container Instances

Containers are typically optimized to run a single application, but the exact needs of those applications can be very different. Azure Container Instances provide optimal utilization by allowing precise specifications of CPU cores and memory. You pay based on what you need and get billed on the second, so you can fine-tune your spending based on actual need.

FAQs

1. **What are probes in ACI?**

- You can configure the liveness probe. We check the liveness probe to see if the container is healthy. If the container is not healthy, we need to restart. There are common scenarios when containers run for a long time.
- You can configure the readiness probe. Here we might have a scenario where the container (maybe DB for the backend) is just coming up. We run the readiness probe and send requests to the container only if the probe succeeds.

2. **How can we monitor ACI?**
   We use Azure Monitor. Here are the available metrics at this time.
   - CPU Usage measured in millicuries (One millicore is 1/1000th of a CPU core)
   - Memory Usage in bytes
   - Network bytes received per second.
   - Network bytes transmitted per second

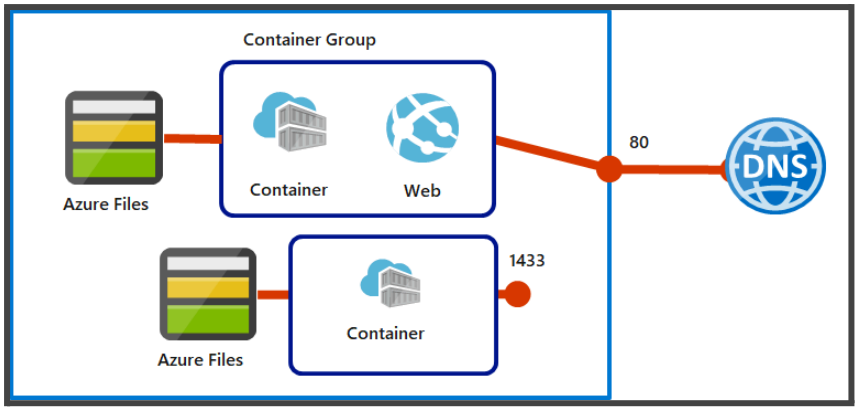
3. **What are container groups?**
   - Similar to AKS for orchestration, we can use container groups to combine and manage containers. They get scheduled on the same host machine.
   - The concept is similar to pods in Kubernetes. The use case for this is in scenarios where we want to divide a single functional task into a smaller number of container images. An example is a front-end container and a back-end container.
   - The front end might serve a web application, with the back end running a service to retrieve data.

Azure Container Groups

Container Groups are top-level resources in the Azure Container Instances.
A container group is a collection of containers scheduled on the same host machine.
Containers in a container group share a lifecycle, resources, local network, and storage volumes.
Container is a similar kind of concept to a pod in Kubernetes.

Configuration example

Consider the following example of a multi-container group consisting of two containers.



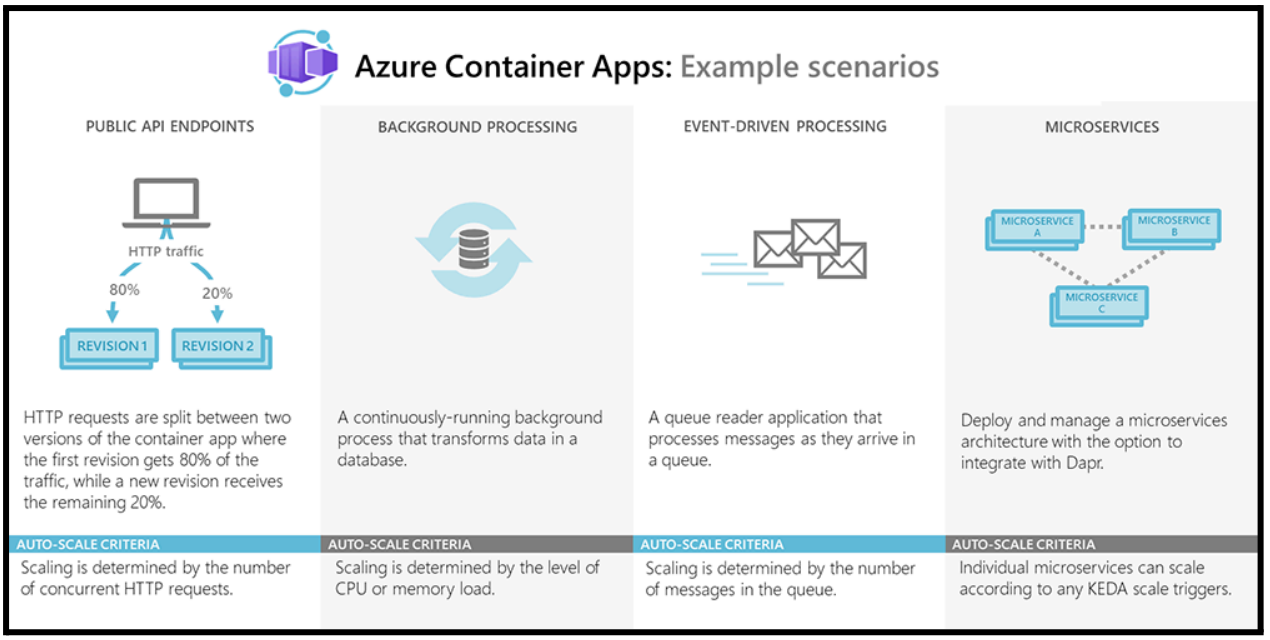
The multi-container group has the following characteristics and configuration:

- The container group is scheduled on a single host machine, and is assigned a DNS name label.
- The container group exposes a single public IP address with one exposed port.
- One container in the group listens on port 80. The other container listens on port 1433.
- The group includes two Azure Files file shares as volume mounts. Each container in the group mounts one of the file shares locally.

For more information —> Introduction to container groups - Azure Container Instances

Azure Container Apps

Azure Container Apps is used to run containerized applications without relying and worrying on orchestration or infrastructure, and Azure Container Apps allows you to run microservices and containerized applications on a serverless platform.



Azure Container Apps common use are : Deploying API endpoints, Hosting background processing applications, Handling event-driven processing and Running microservices.

Azure Container Apps Distinctive features are

- It was optimized for running general purpose containers
- Powered by Kubernetes and open-source technologies - Dapr, KEDA, and envoy.
- It supports Kubernetes style applications and microservices with features - service discovery and traffic splitting.
- It enables event-driven application architectures by supporting scale based on traffic and pulling from event sources such as queues, including scale to zero.
- It supports long running processes and can run background tasks.

D) Create and configure an Azure App Service

Introduction to Azure App Service

Azure App Service allows us to run applications on the cloud. Here are some features:

- HTTP based Service for hosting web applications, REST APIs, and mobile backends.
- Supports .NET, .NET Core, Java, Ruby, Node.js, PHP, Python
- Run and Scale on Windows/Linux

App Services run under an app service plan. An app service plan is the logical abstraction that represents one or more VMs that runs the app service. It consists of compute resources like CPU, memory and disk space. We pay for app service plans and not the app service.

Also, we can have more than one app service running inside an app service plan. The number of app services that can run inside an app service plan depends on the app service plan. Also, the amount of resources like CPU, RAM and disk space depends on the app service plan.

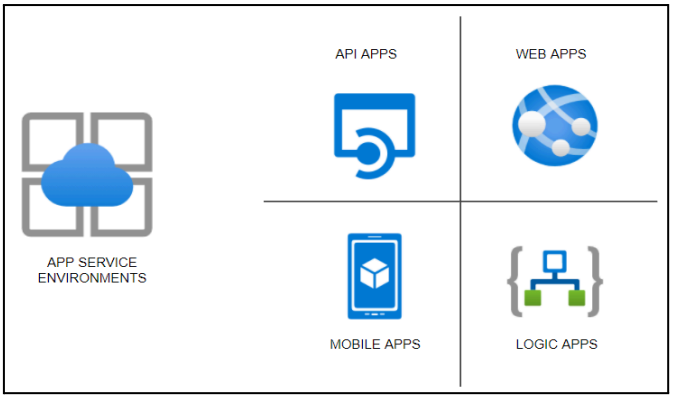
FAQs

1. **How does app service plan work?**
   App service plan is supported by Service Fabric. Service fabric replaces instances if an existing one fails. Also, it adds instances if there is a requirement.

2. **What are the types of App Services?**

   There are 4 types of services as follows:

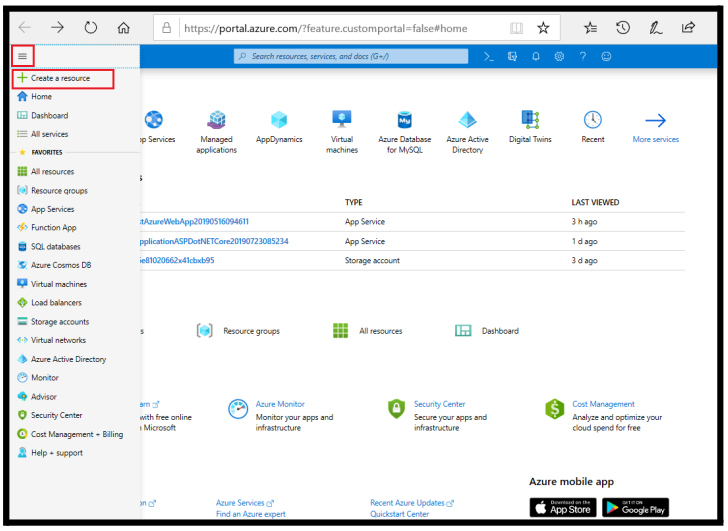
S NO	Type	Purpose
1	Web App (previously Azure Websites)	Hosting websites and web applications
2	API App	Used for hosting the RESTful APIs
3	Logic App	Used for business process automation, system integration and sharing data across clouds
4	Mobile App (previously delivered by Azure Mobile services)	Used for hosting mobile app back ends



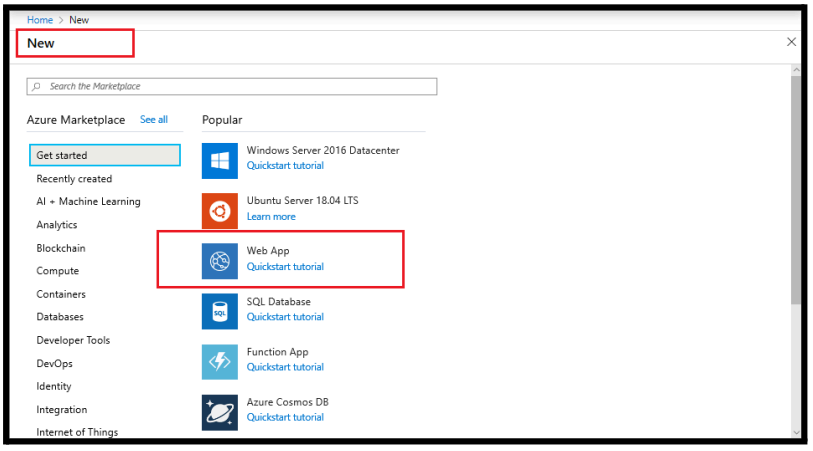
Create an App Service plan

You can create a blank app service plan or create a plan as part of app creation.

1. In the Azure portal, select Create a resource.

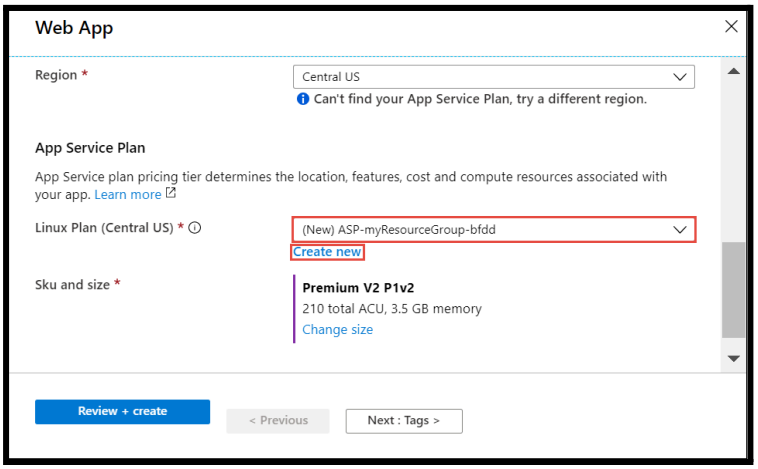


2. Select New > Web App or another kind of App service app.



3. Before you configure an App Service Plan, you must configure the Instance Details section Settings such as Publish and Operating Systems can change the pricing tiers available for your App Service Plan. Region determines where your App Service plan is created.

4. In App Service Plan section, select an **existing plan**, or create a plan by selecting **Create new**.



5. While creating a plan, you can select a new plan pricing tier.
   Select Change Size to change the Sku and Size Price Range.

Creating an Azure App Service

You can practice creating an Azure App Service using our Hands-on-Labs

Creating Azure App Service using ARM template (whizlabs.com)

Secure an Azure App Service Application

Azure App Service provides built-in authentication and authorization support. You can sign in users and access data by writing minimal or no code in your web app, API and mobile backend, and your Azure Functions apps.

Secure authentication & authorization both require a deep understanding of security including federation, encryption, JSON Web Tokens (JWT) management, grant types, and so on. An app service provides these utilities so you can spend more time and energy delivering business value to your customer. Ref: Secure your App Service app - Training | Microsoft Learn

Configuring Custom Domain Names

There are three steps to creating a custom domain name and the following steps describe how to create a domain name in the Azure portal.

1. **Reserve your domain name.**
2. **Create DNS records to map the domain to your Azure web app.**
3. **Enable the custom domain.**

Introduction to Azure Backup Service

- Azure provides an Azure backup service to perform backups.
- We need to install an extension and need to specify the frequency.
- The snapshot will be taken for the OS disk as well as the **data .disk**
- The snapshot taken here is different from the image. The disk is prepared to create an image, and no activity is allowed, and sysprep is done.
- Here, we allow the system to run in snapshotting, and we take either application-consistent snapshots or file consistent snapshots. These snapshots are moved into recovery service vaults.
- We can set up a recovery service vault to replicate to another region.
  For example, we are in the US East, and we replicate to the US West, which protects from entire East US failure.

Configure backup for an App Service

You can easily backup and restore operations in Azure App Service.

You can perform custom backups with already scheduled or on-demand adaptive backups configured in Azure App Service, and you can restore a backup by overwriting an existing application by restoring to a new application or slot.

Backup & Restore are supported in these tiers → Basic, Standard, Premium & Isolated.

Azure contains two types of backups in App Service Environment.

1. **Automatic backups** are made as long as your app is within the support price range.
2. **Custom backups** require initial configuration and can be done on schedule/demand.

For more information,please refer to → Back up an app - Azure App Service | Microsoft Learn

Plan	Compute type	Custom Domain	Scaling	Workload	Space	Backup/ Restore	No of Apps (max)
Free	Shared	No	No		Nil	No	10
Shared	Shared	Yes	Yes	Dev	1GB	No	100
Basic	Dedicated	Yes	Yes	Dev/Test	10GB	No	Unlimited
Premium	Dedicated	Yes	Yes	Prod	250GB	Yes	Unlimited
Isolated	Isolated	Yes	Yes	Prod	1TB	Yes	Unlimited

Configure networking settings

Networking features - Azure App Service | Microsoft Learn

Configure deployment settings

Let's look at some features of App services:

Deployment Slots	This concept is used for zero downtime deployments. There will be a production slot and a Staging slot. New version of the Production deployment will be done in the Staging slot. Either all at once deployment or in stages(canary) will be done.
Deployment Center	This allows for Continuous integration/ Continuous deployment (CI / CD)
Custom Domains	By default, the website will be xxxx.azurewebsites.net. We can buy a domain in your company name and use that name.
SSL Settings	You can certificates and ensure encrypted data transmission between client and Server
Scale up (App Service Plan)	You can increase the size of your VM if you need more resources
Scale out (App Service Plan)	You can also increase the number of instances. You can either do this manually with a slider or set up rules/schedule to scale automatically on schedule or CPU usage (like >70%)

App Service

- HTTP based Service for hosting web applications, REST APIs, and mobile backends.
- Supports .NET, .NET Core, Java, Ruby, Node.js, PHP, Python
- Run and Scale on Windows/Linux

Features

- **PAAS** – Patches/OS Maintenance done by Azure
- Support for Containerization and Docker
- Serverless
- **Deployments Slots** – Swap application content in Prod and avoid downtimes 
- Grouped under App Service plans with following tiers

Plan	Compute type	Custom Domain	Scaling	Workload	Space	Backup/ Restore	Others
Free	Shared	No	No		Nil	No	
Shared	Shared	Yes	Yes	Dev	1GB	No	
Basic	Dedicated	Yes	Yes	Dev/Test	10GB	No	
Premium	Dedicated	Yes	Yes	Prod	250GB	Yes	
Isolated	Isolated	Yes	Yes	Prod	1TB	Yes	Private Endpoints

Please refer to this → [Configure apps - Azure App Service | Microsoft Learn]

App Service types

1. **Webapps** – Websites/Online Apps
2. **Webapps for Containers** – Containerization
3. **API apps** – backend data

Can add – Vnet Integration / Hybrid Connections /Security, but these are not asked in the exams.

Tips

- When you move an App service from one RG to another, the App Service plan doesn't change.
- Destination RG cannot contain App Service resources like Web app or App Service plan.
- **.Net** Core application can be deployed on Windows or Linux OS
- **ASP .Net** app CANNOT be deployed on Linux OS. Only Windows OS
- Multiple Web Apps can be hosted on a single App Service plan.
- Web App and App Service plans must exist in the same region.

Application Service Environments

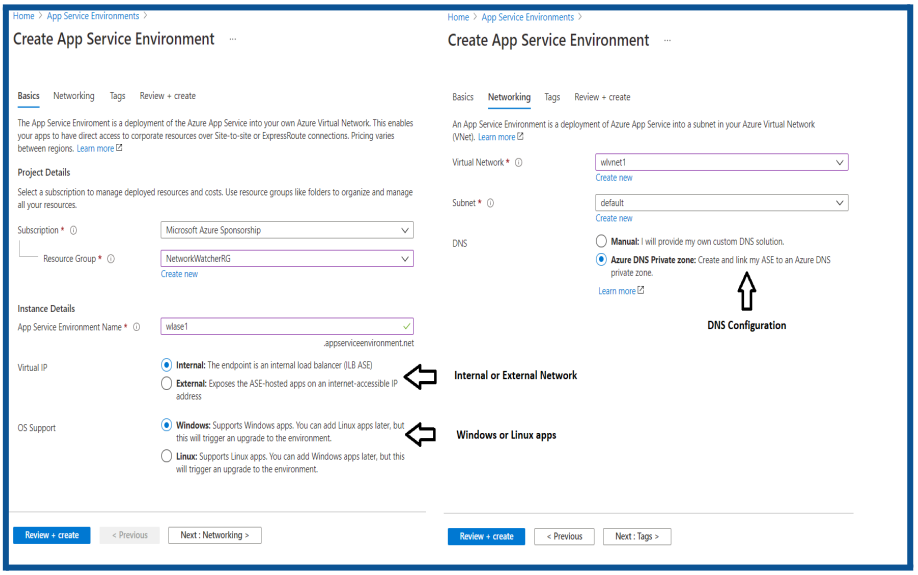
- There are 3 components for hosting *web apps/ Docker containers/ Mobile apps* and functions. There are app service plans which host the app services.
- When we host the regular app services, the apps are directly exposed to the internet, and the resources are shared.
- Some organizations prefer to host the services in the internal network, and security features like firewalls and security groups could be applied to protect the apps.
- For such scenarios, there is a feature called the **Azure App Service Environment**, which provides a fully isolated and dedicated environment for securely running App Service apps at a high scale.
- **App Service environments (ASEs)** provide very high scaling with isolation and secure network access with high memory utilization.
- We can create multiple ASEs within a single Azure region or across multiple Azure regions, making it ideal for horizontally scaling stateless application tiers when we have high **requests per second** (**RPS**) workloads.

There are three types of workloads available when choosing the workload tier. They are *Dev/test, Production, and Isolated.*

- Of these, the isolated offering provides the ASE environments which host applications within the client's VNets. As stated, we have fine-grained control over inbound and outbound application network traffic.
- While the other category of app services has a fixed suffix of azurewebsties.net, we can create our own domain name.
- Also, ASEs come with powerful computers, which is twice as powerful as the regular app service plans. They also come with **1TB** Storage as compared to **50GB** of space for the regular ones.
- We can host up to 100 instances which are sufficient to host a miniature web service hub. We can expect the service to cost us about **250-300$** per month, which is very cheap for the services being provided.

Steps to creating App Service Environment

- In the first screen, we select if the service is public-facing or internal
- Then we select whether we are hosting Windows-based or Linux-based OS.
- On the second screen, we select the Vnet where we want to host the service. *(Since services are being created in our private infrastructure, it takes much longer time to create)*
- Then we can DNS resolution. We can create our own private zone and use that name. This is not possible when choosing the other app service plans.

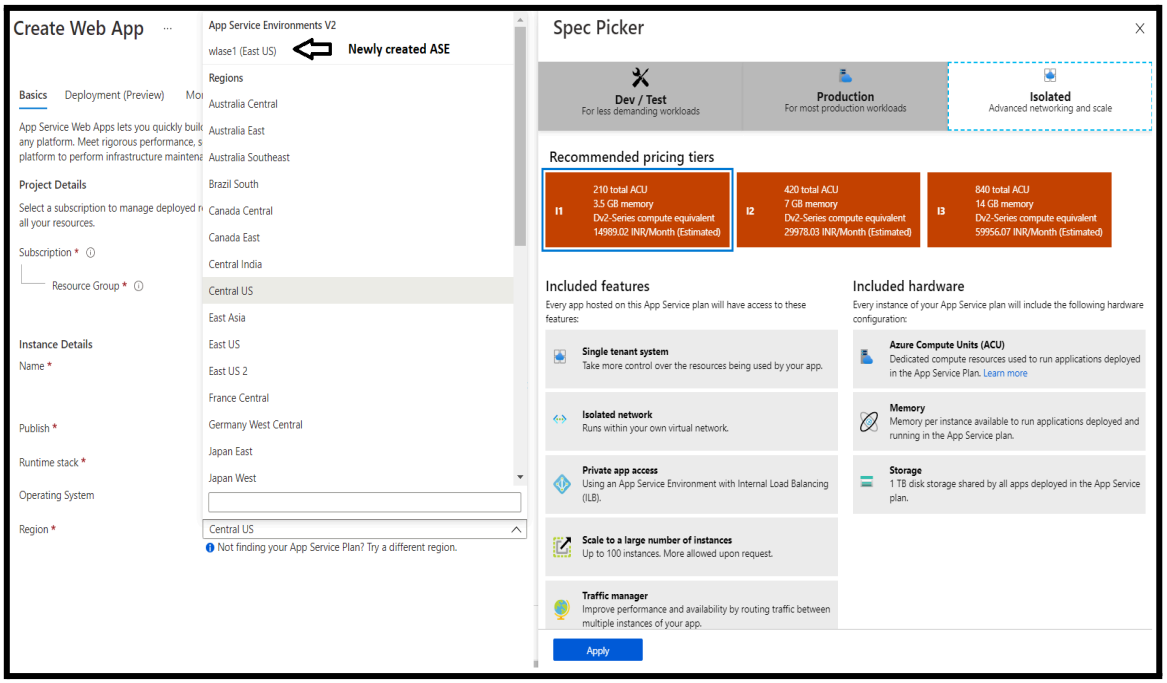


(Source: Microsot Documentation)

Steps to creating Web Apps under ASE

- Please note that the process is similar except that we drop down the region and select the ASE that we just created.

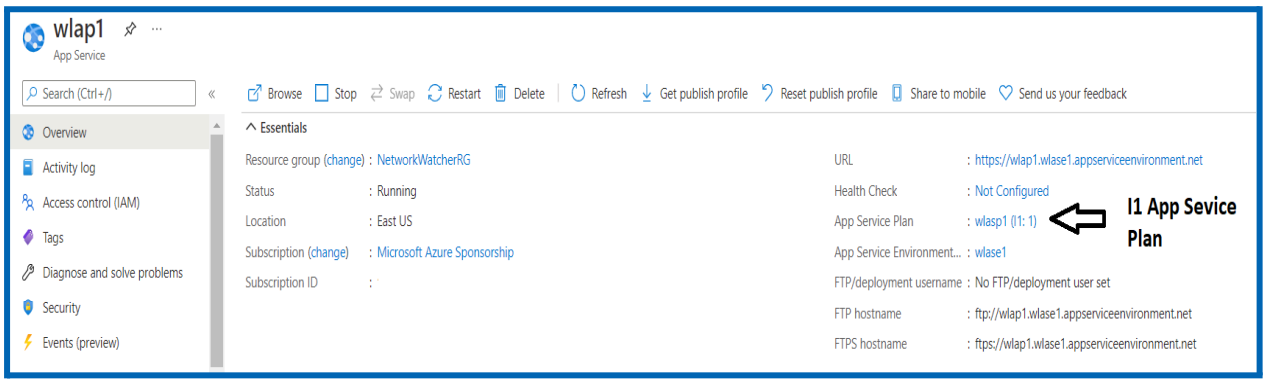
- Also, the below screen shows various features under ASE and pricing under each of the pricing tiers I1 and I2, and I3.



(Source: Microsoft Documentation)

Note: *The Private link vnetLink (wlase1.appserviceenvironment.net/vnetLink) is also created below. You can go to the Resource group and click on "Show hidden types" to see this resource.*

Note: *Please see the App Service plan as I1:1 in the screenshot below to identify the isolated service plan.*



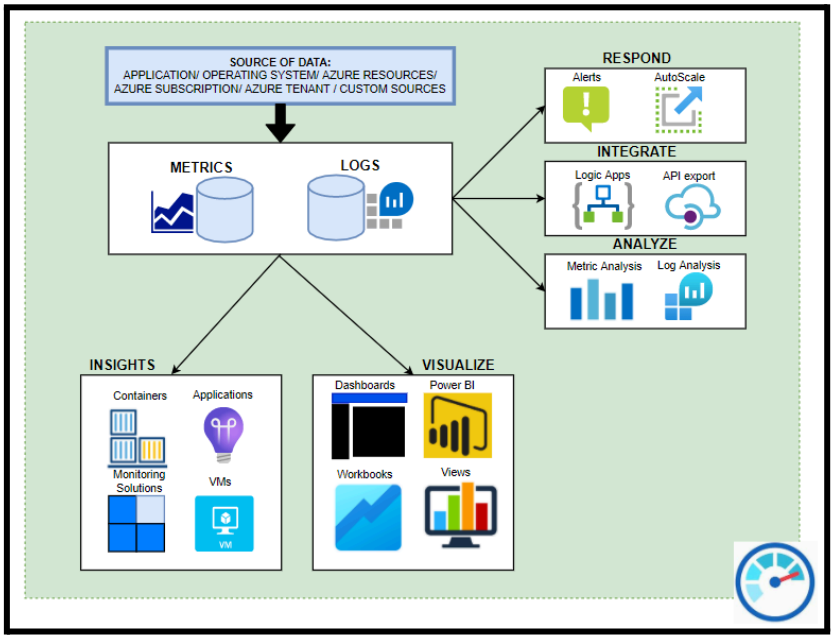
Monitor and Maintain Azure resources

Introduction to Azure Monitor

- Azure Monitor is a free service that helps increase performance and availability. We could collect telemetry data from Azure as well as on-premises.
- We could collect the metrics and logs from our resources like VMs. We could even collect more detailed logs by enabling guest diagnostics and collecting OS level information.
- We can also integrate additionally with **SIEM and ITSM** tools. We could also send data via event hubs or other services.
- Metrics are available at each resource level or they can be collectively seen at the Azure Monitor. This way Monitor acts as a central location for all our monitoring needs like Metrics, logs, alerts and activity logs.
- We also have a section on Insights where we can see more intelligent information for various resources like *Applications, VMs, Storage Accounts, Containers, Networks, SQL (Preview), CosmosDB, KeyVault, Azure Cache for Redis.*
- We could also see a map of our application and understand how the different components work together.

A) Monitor resources by using Azure Monitor

Creating, Confuguring and Manging of Azure Monitor

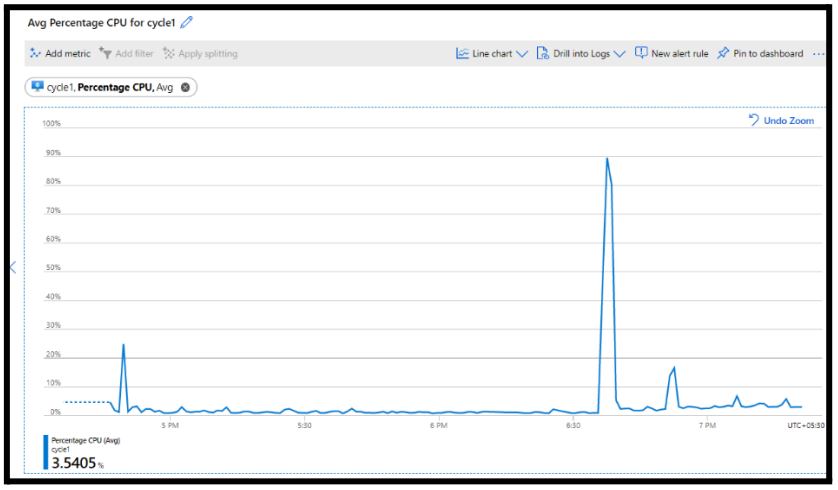


At a high level, we can do the following by using Azure Monitor

1. **Monitor & Visualize Metrics**
2. **Query & Analyze Logs**
3. **Setup Alert & Actions**

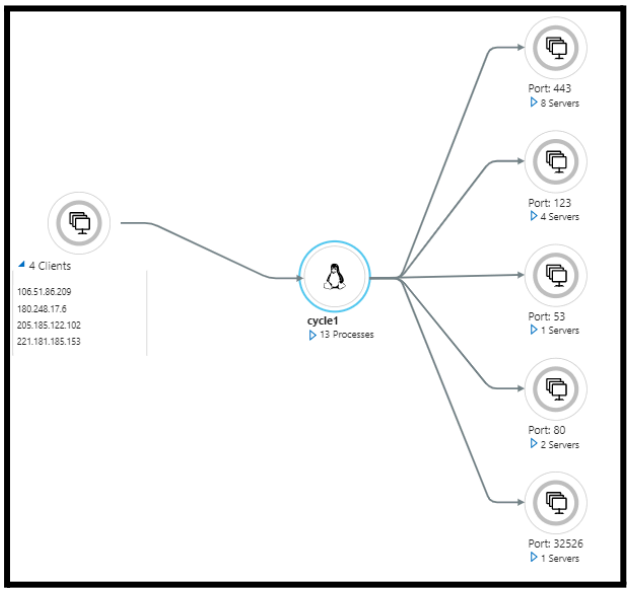
Here are some of the components which make up the Azure Monitor

1) **Inputs** –
   a. **Logs** – these are the logs generated by various resources like VMs/ Databases etc.,
   b. **Metrics** – Metrics provides numbers like *CPU percentage, Network data in/out* which helps us understand performance.
   The metrics are stored in a time series DB which helps understand real time scenarios. With metrics, we can set triggers to scale the resources up and down. Please see a metric chart below on CPU percentage usage:

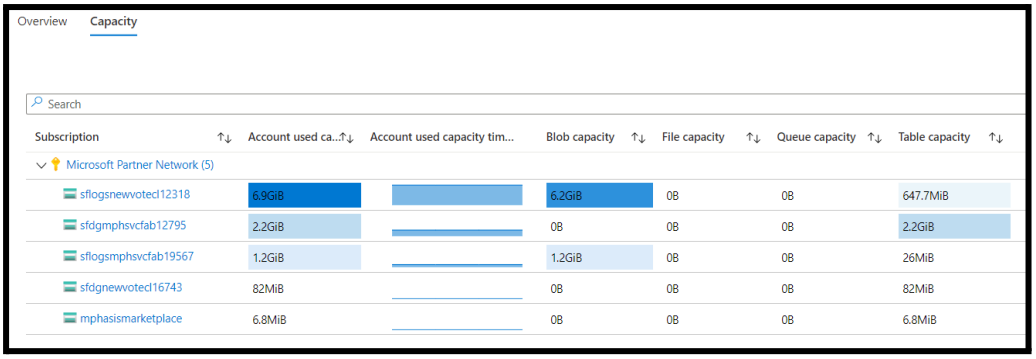


2) **Insights**
   a. With Insights, we can get a deeper view into the resources. We could see a map of the resources and get an overall view. Please see below some insights:

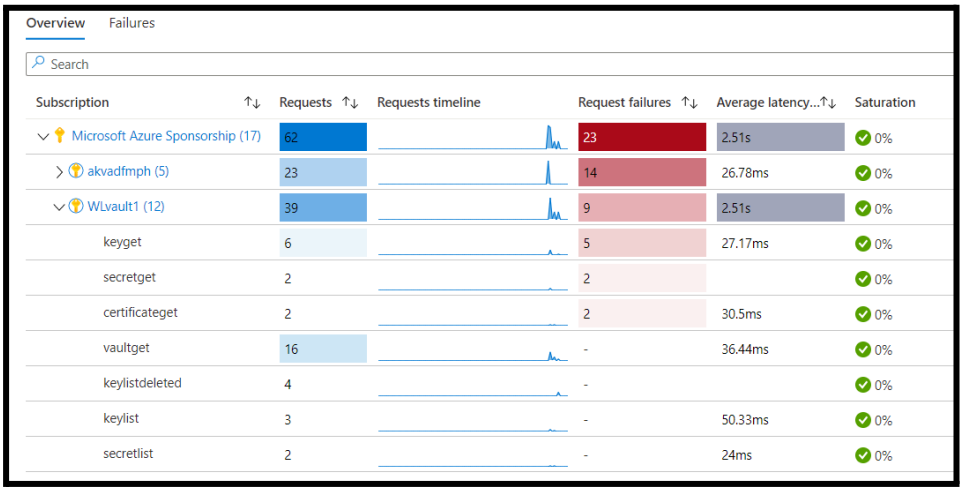
APPLICATION MAP:



STORAGE OVERVIEW



KEYVAULT INSIGHTS



3) **Analyze**
   a. **Log Analytics** – We can work with log data from multiple sources with log analytics. We can perform complex queries with *KQL (Kusto Query Language)*. We can analyse and act on that data.
   b. **Metric Analysis**

4) **Visualize**
   a. **Metrics explorer** – interactively work with metric data with metric explorer
   b. **Workbooks** – We can use a combination of text, metrics, log queries and parameters into interactive reports. There are several built-in workbooks available for use.
   c. **Dashboards** – We can add metric graphs and queries output and create dashboards.

5) **Respond**
   a. **Alerts** - When there is any issue, then we will get alerts proactively and we can automatically run *functions, runbooks, webhooks or logic apps.*
   b. **AutoScale** – With the metric as inputs, we can set up the system to scale up or down automatically.

Configure Azure Monitor Logs

Azure Monitor Logs is a feature of Azure Monitor that collects and manages log and performance data from monitored resources.

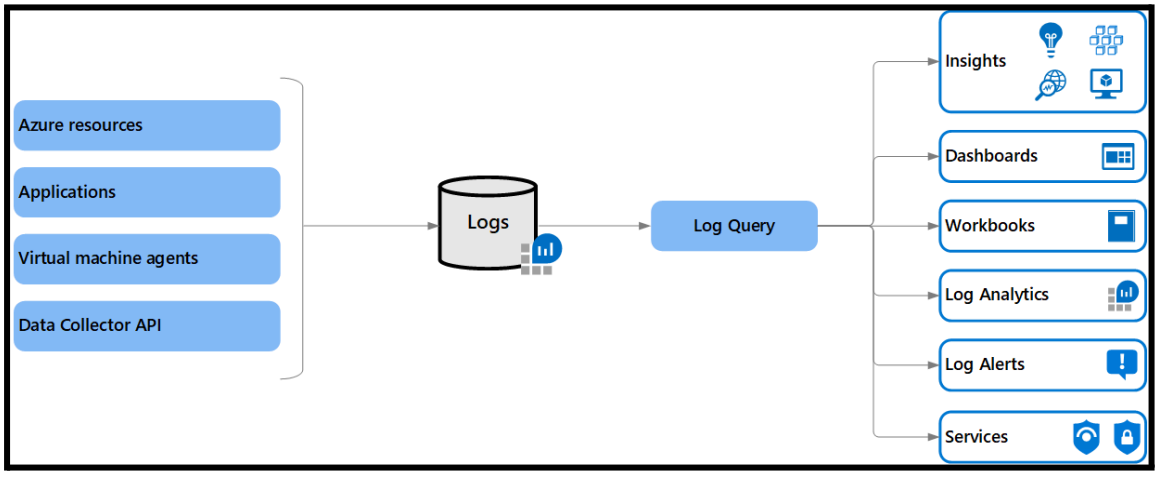
Azure Monitor Logs Benefits

Capability	Description
Analyze	Use Log Analytics in Azure Portal to write log queries and interactively analyze log data by using a powerful analysis engine.
Alert	Configure a log alert rule that sends a notification or takes an automated action when the results of a query match a specific result.
Visualize	Query results are rendered as tables or charts to an Azure dashboard. Create a workbook to combine multiple sets of data into an interactive report. Export the results of a query to Power BI to use different visualizations and share with users outside of Azure. Export query results to Grafana to use its dashboarding and combine with other data sources.
Get insights	Logs support insights that provide a customized monitoring experience for specific applications and services.
Retrieve	Access log query results from: 1. Command line via Azure CLI or Azure PowerShell cmdlets. 2. A custom app via a REST API or client library for .NET, Go, Java, JavaScript or Python.
Export	Configure automated export of log data to an Azure Storage account or Azure Event Hubs. Create a workflow to retrieve log data and copy it to an external location using Azure Logic Apps.

Data Collection: After you create a Log Analytics workspace, you must configure sources to send their data. No data is collected automatically.

Azure Monitor Logs stores the data that it collects in one or more **Log Analytics workspaces.**

- You need to create at least one workspace to use the Azure Monitor Logs service.
- **Log Analytics** is a tool in the Azure portal, with this tool you can edit and run log queries and analyze their results interactively.



Data is retrieved from a Log Analytics workspace through a **log query,** which is a read-only request to process data and return results.

Azure Monitor Logs is based on **Azure Data Explorer**. A Log Analytics workspace is roughly the equivalent of a database in Azure Data Explorer.

Configuring VM insights

VM Insights monitors the performance and health of your virtual machines and virtual machine scale sets. It monitors their running processes and dependencies on other resources.

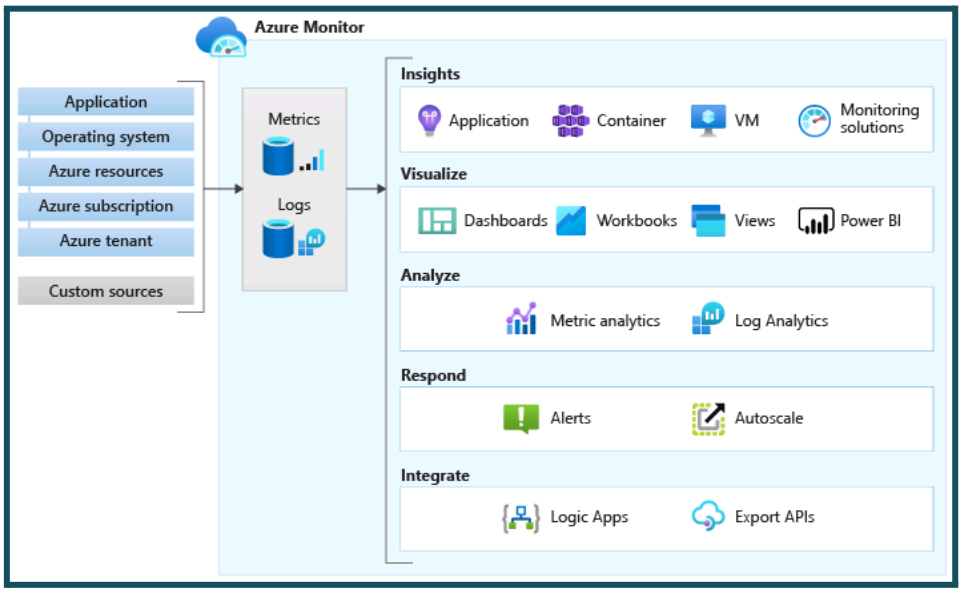
VM Insights helps deliver predictable performance and availability of critical applications by identifying performance bottlenecks and network issues. It will also help you understand if the problem is related to other dependencies or not.

Few important points about VM insights

- VM Insights stores its data in Azure Monitor Logs, which allows it to provide powerful aggregation and filtering and analyze data trends over time.
- There is no direct cost for VM Insights, but you will be charged for its activity in the Log Analytics workspace.
- VM insights doesn't support sending data to multiple Log Analytics workspaces (multi-homing).

Benefits of VM insights to monitor the health and performance of:

- Azure virtual machines.
- Azure Virtual Machine Scale Sets.
- Hybrid virtual machines connected with Azure Arc.
- On-premises virtual machines.
- Virtual machines hosted in another cloud environment.



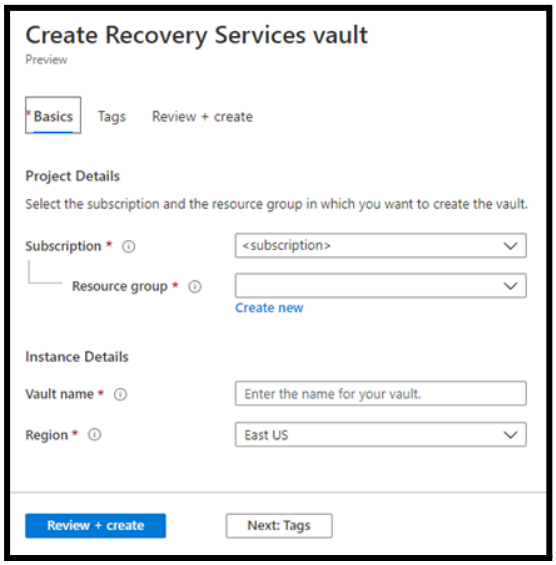
(Source: Microsoft Documentation)

B) Implement backup and recovery

Create an Azure Recovery Services vault(RSV)

A recovery services vault is a storage entity in Azure that stores data. Recovery Services Vaults make it easy to manage your backup data while reducing maintenance overhead.

1. It can be used to back up Azure Files file shares or on-premises files and folders.
2. It stores backup data for various Azure services such as IaaS virtual machines (Linux or Windows) and Azure SQL databases.
3. It supports System Center Data Protection Manager, Windows and Azure Backup Server, etc.
4. In the Azure portal, you can create a Recovery Services vault from the Backup Center dashboard.



(Source: Microsoft Documentation)

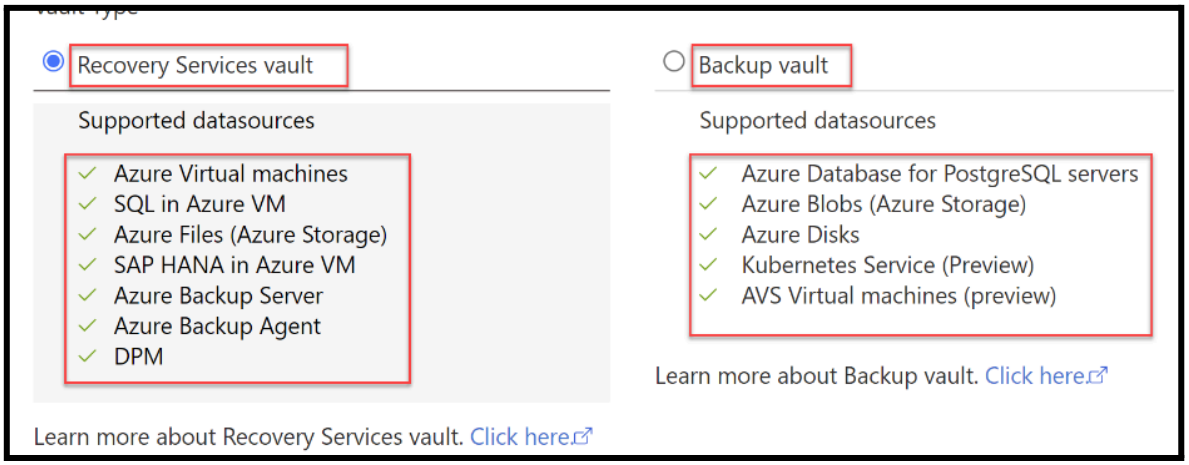
Create an Azure Backup Vault(ABV)

A backup vault is an entity that stores backups and recovery points created over time. Backup Vault also includes backup procedures associated with protected resources.

Example from Microsoft Documentation:

1. Type Backup vaults in the search box (in the top of the Console)
2. Under Services, select → Backup vaults.
3. On the Backup vaults page, select → Add.
4. On the Basics tab → Project details, make sure the correct subscription is selected and then choose Create new resource group. Type *myResourceGroup* for the name.
5. Under Instance details, type *myVault* for the Backup vault name and select your region of choice, in this case *East US* for your Region.

6. Select your Storage redundancy. It can't be changed after protecting items in the vault.
7. Microsoft recommends that if you're using Azure as a primary backup storage endpoint, continue to use the default Geo-redundant setting.
8. If you don't use Azure as a primary backup storage endpoint, Select Locally redundant
9. Select the Review + create button at the bottom of the page.



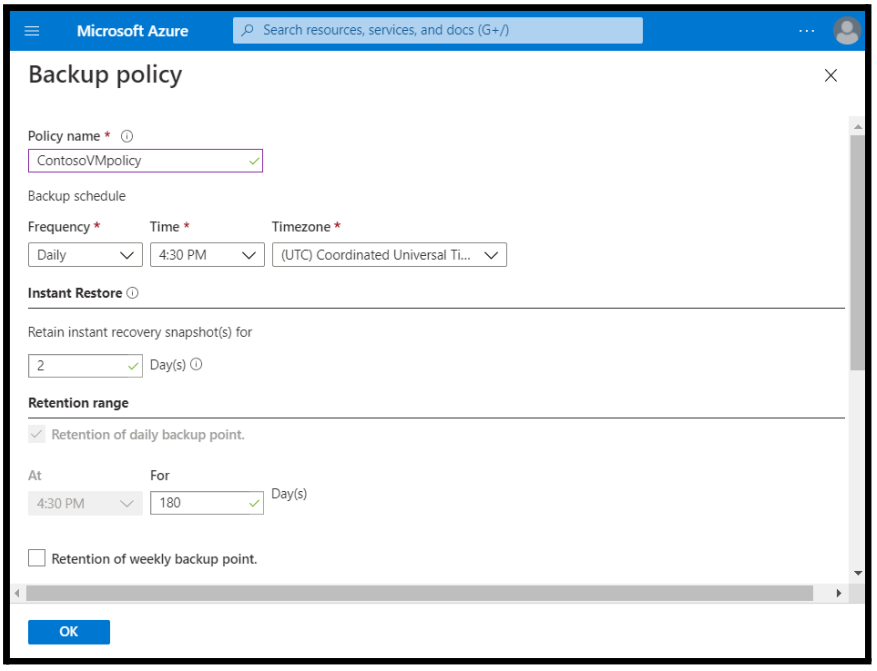
(Source: Microsoft Documentation)

Creating an Azure Backup Policy (through the Azure portal)

➔ It's better if you can create a backup policy through the Azure portal.

➔ In the below example, the Admin has selected Backup in the Operations section on a specific VM, in this case, ContosoVM1.

➔ The Admin can select the Create **a new policy** link in the Choose **backup policy** section.



(Source : Microsoft Documentation)

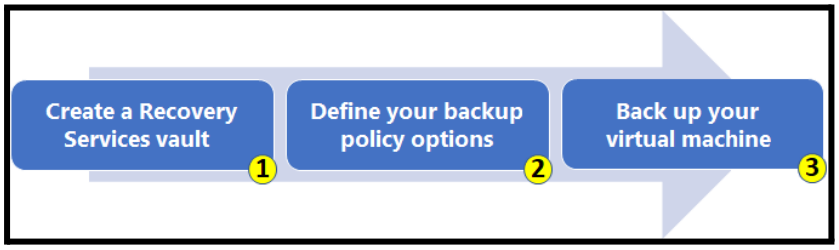
In the Backup Policy blade displayed in the screenshot, the administrator can define the following properties:

- **Policy name**
- **Backup schedule**
- **Instant Restore settings**
- **Retention range**

Azure Backup and Restore Operations

Backup your virtual machines

If you are using Azure Backup to protect your Azure virtual machines, you follow a simple three-step process: **create a vault, define your backup options, and trigger a backup job.**



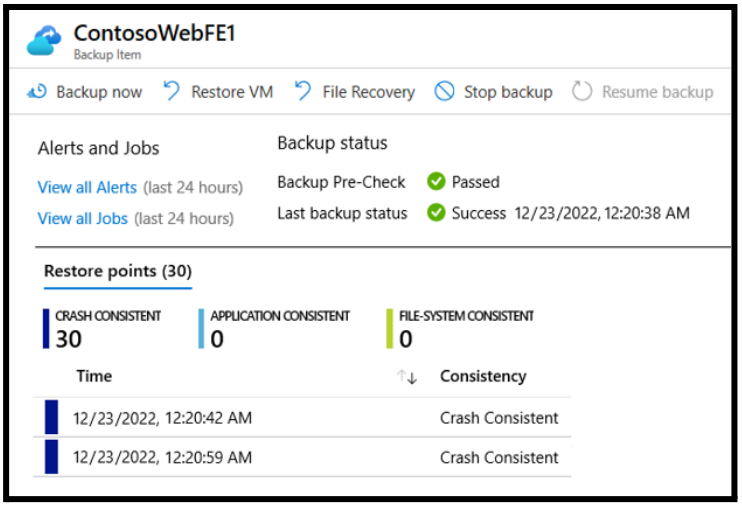
(Source : Microsoft Documentation)

There are three important steps for Backup Operations are

- Step 1. You need to create a Recovery Services Vault
- Step 2. Then you need to Define backup policy options
- Step 3. Finally you need to Backup your virtual machine

Restore your virtual machines

After you back up your virtual machine, the backup snapshots and recovery points are stored in your Recovery Services vault. You can restore your machine by accessing a snapshot or restore data to a specific point-in-time by using recovery points.



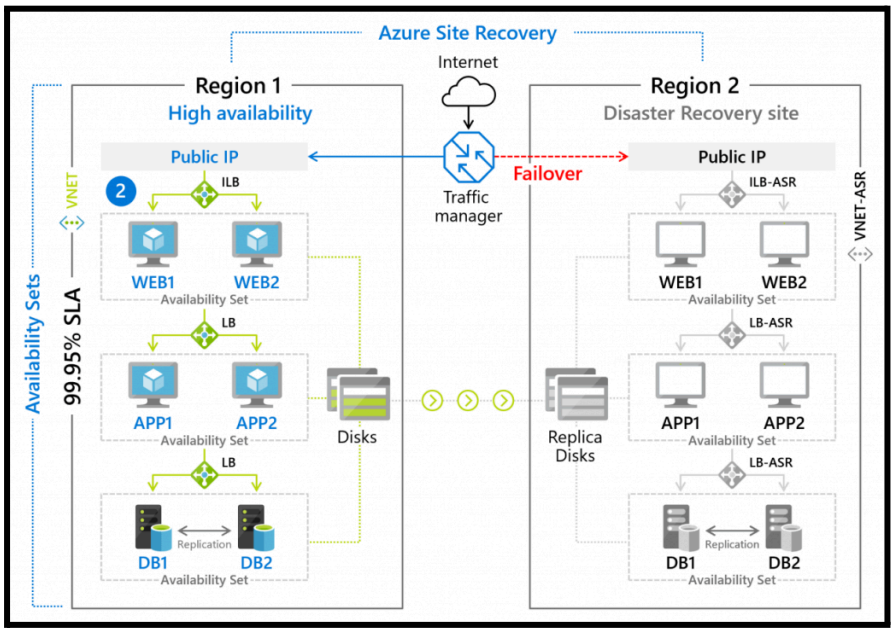
(Source : Microsoft Documentation)

Configure Azure Site Recovery for Azure resources

Azure Site Recovery(ASR) helps ensure business continuity by keeping business applications and workloads running during outages.ASR replicates workloads running on physical and virtual machines from a primary site /location to a secondary site/location.

If any outage occurs at your primary site, the Site Recovery implements a failover to your secondary location to allow continued access to your applications. After the primary location is up and running again, you can resume application access on the primary machine.

The following image shows two regions connected by Azure Traffic Manager.
Azure Site Recovery is deployed to enable failover from Region 1 to Region 2.



(Source : Microsoft Documentation)

Azure Site Recovery (ASR) Characteristics

Azure Site Recovery supports multiple configurations and complements various Azure services.

- Replicate Azure virtual machines from one Azure region to another
- Replicate on-premises VMware virtual machines, Hyper-V virtual machines, physical servers (Windows and Linux) and Azure stack virtual machines to Azure
- Replicate AWS Windows instances to Azure
- Replicate on-premises VMware virtual machines, Hyper-V virtual machines managed by System Center VMM, and physical servers to a secondary site

Azure Site Recovery(ASR) benefits:

- Consolidated management, Reduced cost and complexity,
- Replication resilience, Continuous replication,
- Snapshot recovery points, Failover and easy fall back and Integration.

Failover to a secondary region by using Azure Site Recovery

The below are the important steps for to perform failover to a secondary region by using ASR

1. Check prerequisites
2. Verify VM settings
3. Run a failover to the secondary region
4. Start replicating the VM back to the primary region.

Pls refer to below link for more information:
Tutorial to Failover to a secondary region by using Azure Site Recovery| Microsoft Learn

Configure and review Backup reports

Azure Backup provides a reporting solution that uses Azure Monitor logs and Azure workbooks. These resources help you gain great insights into your backups across your entire backup estate. The points below shows → how to configure and view Azure backup reports.

Supported scenarios for backup reports
(Source : Microsoft Documentation)

- Backup reports are supported for Azure VMs, SQL in Azure VMs, SAP HANA in Azure VMs, MARS agent, MABS, and System Center Data Protection Manager (DPM).
- For DPM workloads, Backup reports are supported for DPM Version 5.1.363.0 and above and Agent Version 2.0.9127.0 and above.
- For MABS workloads, Backup reports are supported for MABS Version 13.0.415.0 and above and Agent Version 2.0.9170.0 and above.
- Backup reports can be viewed across all backup items, vaults, subscriptions, and regions.
- If you're an Azure Lighthouse user with delegated access to your customers' subscriptions.
- Currently, data can be viewed in Backup Reports across 100 Log Analytics Workspaces.
- Data for log backup jobs currently isn't displayed in the reports.

Important steps to start using the reports are as follows

1. Create a Log Analytics workspace or use an existing one
2. Configure diagnostics settings for your vaults
3. View reports in the Azure portal

For more information, pls refer to this → Configure Azure Backup reports - Azure Backup

Implement and Manage virtual networking

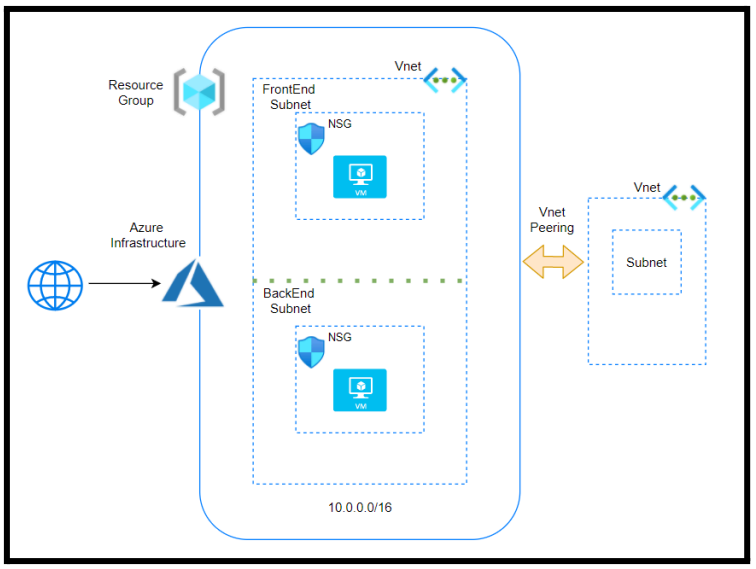
Introduction to Azure Virtual Network(Azure VNet)

A **Vnet** is the fundamental building block for a private network in Azure. VNets allow Azure resources like VMs to communicate securely with each other, to the internet, and to on-premises networks. A **Vnet** is the representation of our own network in the cloud. We can logically isolate resources within our Vnet.

Benefits and Components of Virtual Netwrok(VNet)

Benefits

- **Isolation** – As discussed, the components of a Vnet are isolated. We can connect to other Vnets or On-Premises with Vnet Peering or VPN or Express route
- Access to the public network
- Access to VMs within the Vnet
- **Name resolution** – We can resolve to other components in the Vnet and address them
- **Security** – We can secure the components at various levels in the Vnet
- Connectivity



Components

- **IP addresses**
  - **Public and private IP addresses**
    - The Vnets are configured with a range of IP addresses. The Notation is in CIDR.
    - By default, Private IP addresses are assigned to the resources with which communication takes place between the resources
    - Optionally, Public IP address can be assigned to the resources
    - Please note that we will pay for Public IPs if they are not assigned. This is to conserve Public Ips

- **Subnets**
  - A Subnet is a subcomponent of Vnet. All resources must exist in a subnet. A default subnet is created when a Vnet is created.
  - Access can be restricted at a subnet level also
  - Let's say we have 2 tiers in an application called Front end and Back end. We can create 2 subnets and configure access in such a way that internet traffic will flow to the front end subnet and from there to the back end subnet.

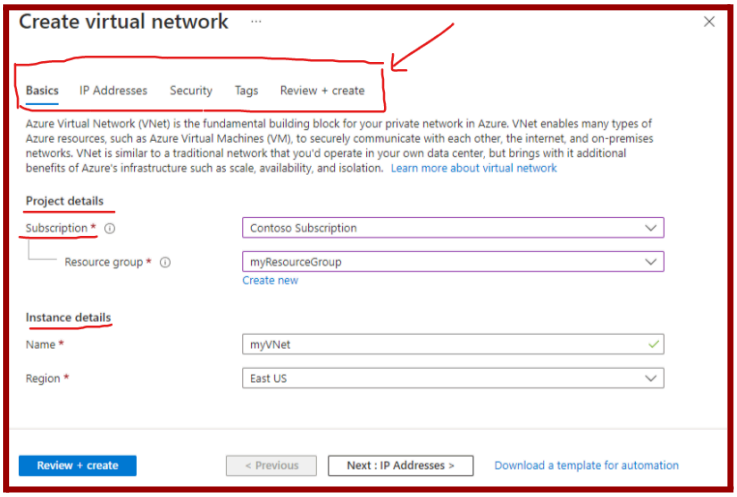
- **NIC - Network interface card**
  - A NIC is the networking component which allows traffic flow. A single NIC will contain the public and private address.

- **NSG - Network security group**
  - These are the rules that are assigned to allow traffic to flow. The NSG can be assigned at a NIC level or a subnet level. It is recommended to apply at any one level only.
  - If there is no NSG, then traffic will be allowed in and out
  - We set inbound and outbound rules
  - **Priority** – All rules are assigned a priority and the lowest number is taken first. If rule 100 says allow and 101 says deny, then the result is allow.
  - **Default Security rules** – There are 6 default rules that can neither be removed or modified.

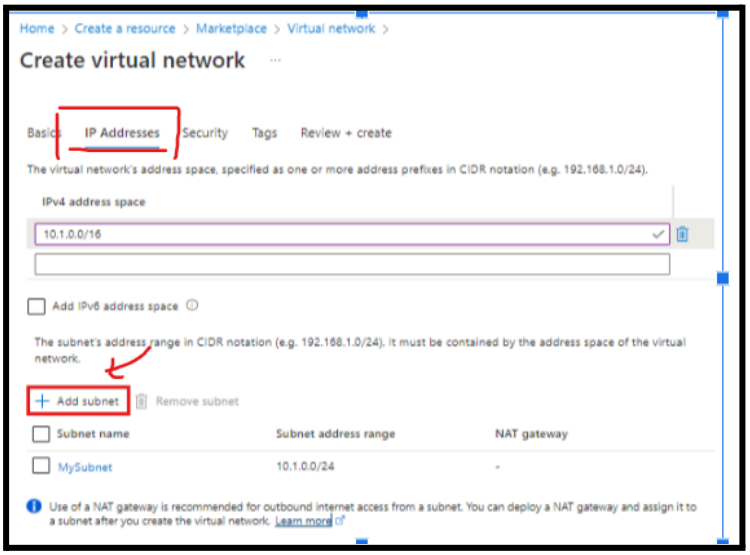
A) Configure virtual networks

Create and configure virtual networks and subnets

1. Select Create a resource in the upper left-hand corner of the portal.
2. In the search box, enter Virtual Network. Select Virtual Network in the search results.
3. In the Virtual Network page, select Create.
4. In Create virtual network, enter or select this information in the Basics tab

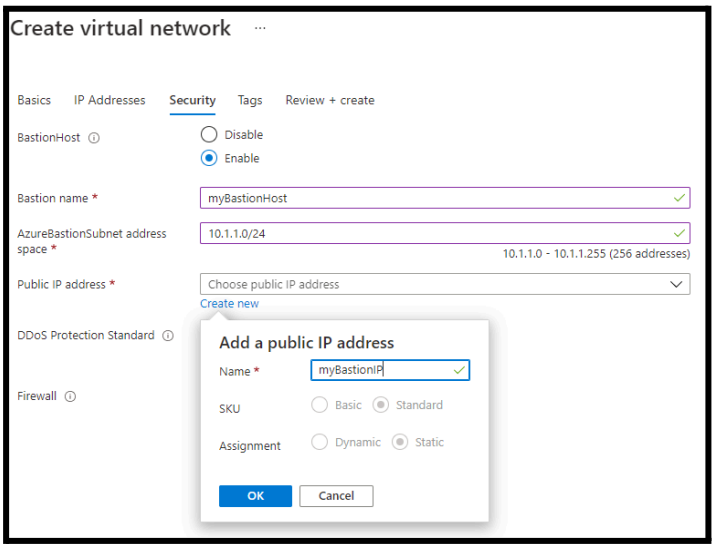


5. Select the IP Addresses tab, or select the Next: IP Addresses button at the bottom of the page and enter in the following information then select Add:



6. Select the **Security tab**, or select the **Next:** Security button at the bottom of the page.

7. Under BastionHost, select Enable. Enter this information:



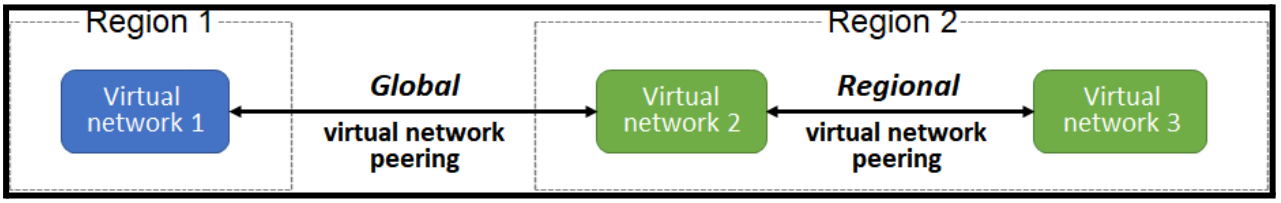
8. Select the Review + create tab or select the Review + create button.

Practice this with using our Hands-on-Labs → [Create a Virtual Network (whizlabs.com)]

Virtual Network Peering

Azure supports the following types of peering:

- **Virtual network peering**: Connecting virtual networks within the same Azure region.
- **Global virtual network peering**: Connecting virtual networks across Azure regions.

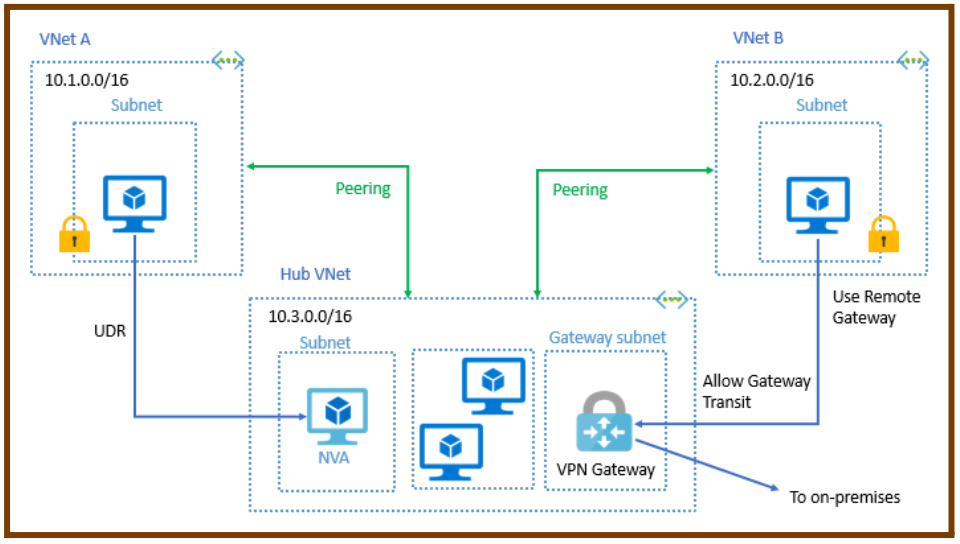


FAQs

1) **What is Vnet Peering?**
   Vnet Peering allows two Vnets either in the same region *(Default Vnet Peering) or Globally (Global Vnet Peering)*

2) **What are the pre-checks for Vnet Peering?**
   a. Peering is non-transitive. If Vnet A is peered with Vnet B and Vnet B with peered with Vnet C then it does not mean that Vnet A and Vnet C are connected
   b. The Address ranges cannot overlap between the Vnets
   c. When peered, adding or deleting address range is disabled. If we need to add address range, we need to delete the peering and add the address range and then add peering again.

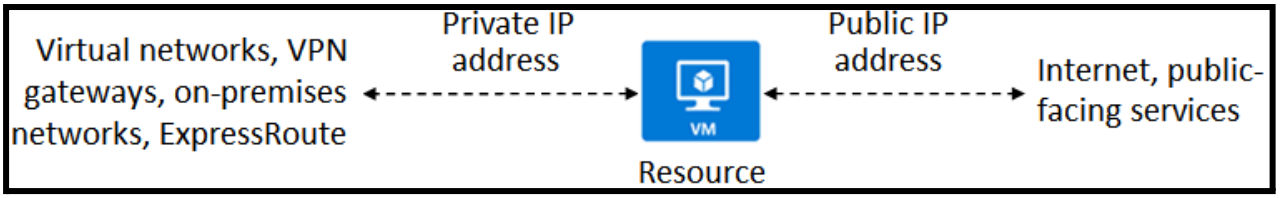
Gateways and on-premises connectivity Architecture



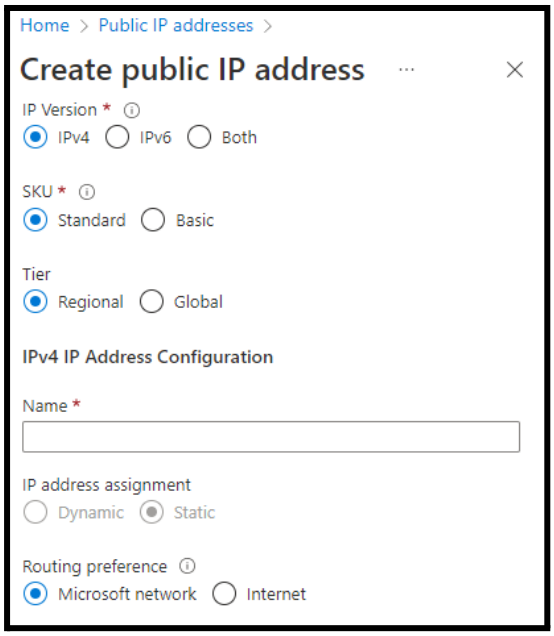
Public and Private IP Addresses

Private IP addresses Enable communication between Azure Virtual Network and your on-premises network. When you use a VPN gateway or Azure ExpressRoute circuit to extend your network to Azure you create a private IP address for your resource.

Public IP addresses Allow your resource to communicate with the Internet. You can create a public IP address to connect to Azure public facing services.



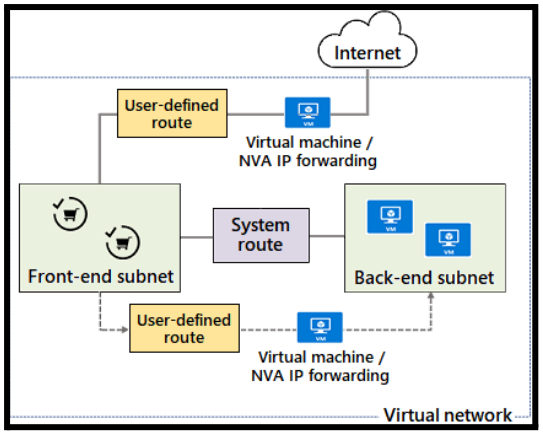
You can create a public IP address for your resource in the Azure portal.



User-Defined Routes (UDR)

You can create custom or user-defined (static) routes in Azure to override Azure's default system routes or add more routes to the subnet route table.

Azure handles all network traffic routing automatically, but in some cases, a custom configuration is preferable. In these situations, you can configure user-defined routes (UDRs) and next-hop destinations.



UDR Characteristics

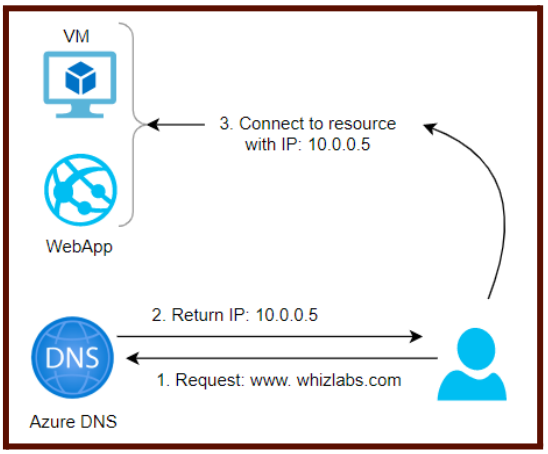
- The next hop can be one of the following targets:
  Virtual network gateway, Vnet, Internet, Network virtual appliance (NVA)
- Similar to system routes, UDRs also access route tables.
- Each route table can be associated with multiple subnets.
- Each subnet can be associated to one route table only.
- There are no charges for creating route tables in Microsoft Azure.

Configuring Azure DNS

What is DNS?

Think of the phone directory that is used at home. It is difficult to remember a string of numbers and hence the phone directory will list the phone numbers with names of persons/businesses.

- Coming back to the IT world, computers communicate with IP addresses. The DNS (Domain naming system) is a friendly name given to the computer.
- For example, a web server has an IP address of **53.102.94.86.** Instead of using the IP Address, we assign a host name as web1. In a domain, the **FQDN (Fully qualified domain name) will be web1.whizlabs.com.**



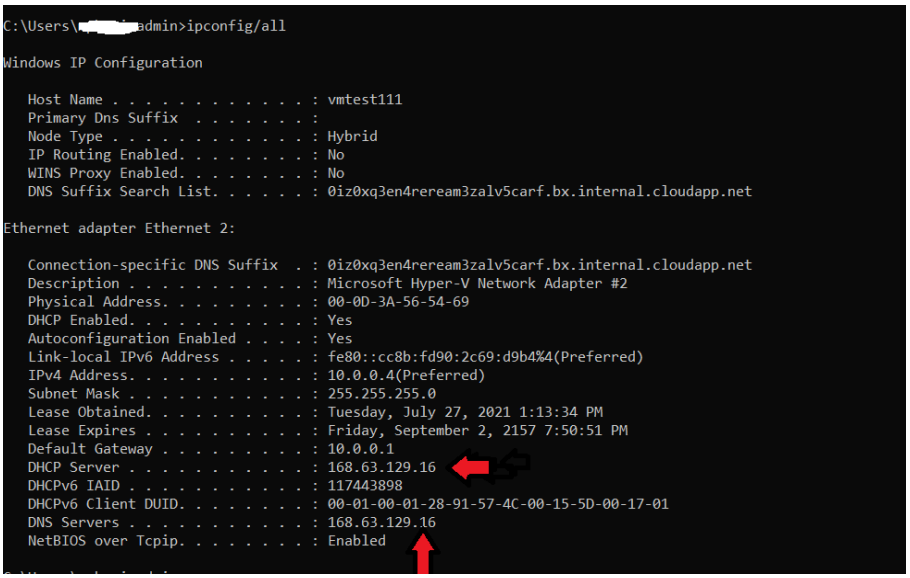
- This is facilitated by DNS Servers which are setup in a hierarchy. At the top most level, we have the ROOT and under the root, we have the top level domains (TLD) examples of which are **.ORG, .COM, .NET, .IN etc.,**

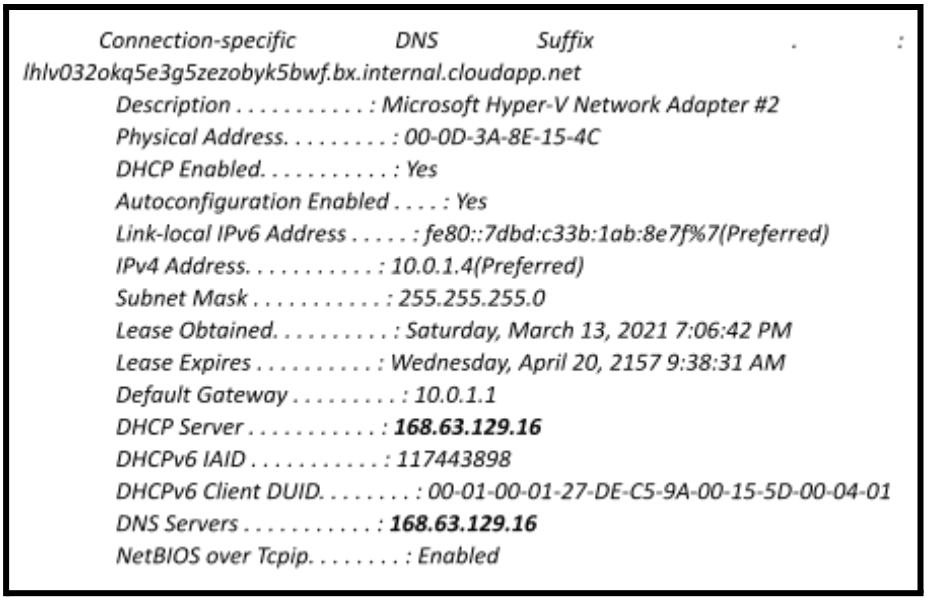
- In addition to this, we have domain registrars where we purchase a domain name.
- Examples are **Godaddy, Namecheap and Amazon too via Route53**. When a user tries to connect to a server whizlabs.com, the DNS resolves this to the IP address by going to the **ROOT** and then to the **.COM server.**DNS works with a concept of Zones. We can set up Private or Public zones. Public zones are used when we want the internet to be able to resolve our names.
- However when we want to enable internal communication, we create private zones.
- Please note that zones can also be configured with a "**Split-horizon**" view which allows a private and public DNS zone to share a name.

FAQ

1) **What is IP 168.63.129.16?**

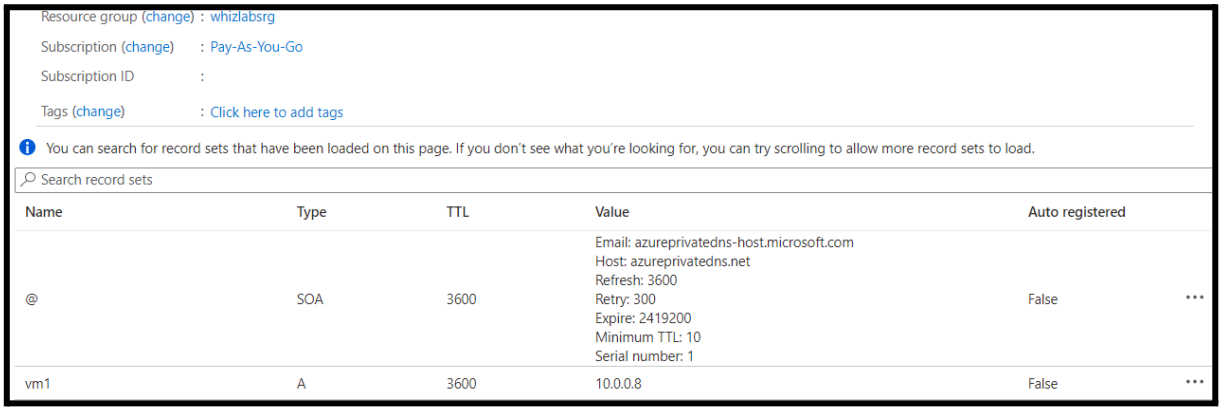
   This is actually called a Wire Server and has an IP address of 168.63.129.16. and it facilitates communication between Azure resources. It also serves as a DNS and DHCP server by default. Please ensure that this IP is not blocked.





2) **Can I buy my domain from Azure?**
   No, Azure is not a domain registrar. You need to buy from a domain registrar and you can create a zone in azure and add the records for DNS resolution.
3) **How do we configure VMs to use private zones?**
   We can configure auto registration and for Vnet that we link with the Virtual Network Link on the DNS Zone, the DNS registration will be done automatically when the VM is created.

4) **How do I use my custom website?**
   We need to create a public zone and add an alias record. Once verified with the registrar, we can start using our custom name.



B) Configure secure access to virtual networks

Network Security Groups(NSG)

Network security group is used to filter network traffic between Azure resources in an Azure virtual network. A network security group contains security rules. These can allow or deny inbound and outbound network traffic from various Azure resources.

Each rule contains a few things and you can specify them as →
Source, Destination, Port, and Protocol.

When we create any NSG, the default security rules are applied and the rule attributes you can modify to create an augmented security rule.

Network security group

- These are the rules that are assigned to allow traffic to flow. The NSG can be assigned at a NIC level or a subnet level. It is recommended to apply at any one level only.
- If there is no NSG, then traffic will be allowed in and out
- We set inbound and outbound rules
- **Priority** – All rules are assigned a priority and the lowest number is taken first. If rule 100 says allow and 101 says deny, then the result allows.

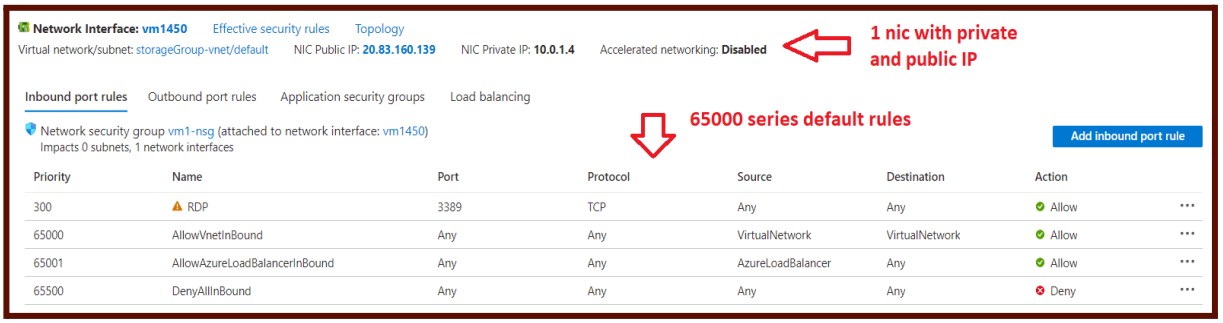
- **Default Security rules** – There are 6 default rules that can neither be removed or modified.

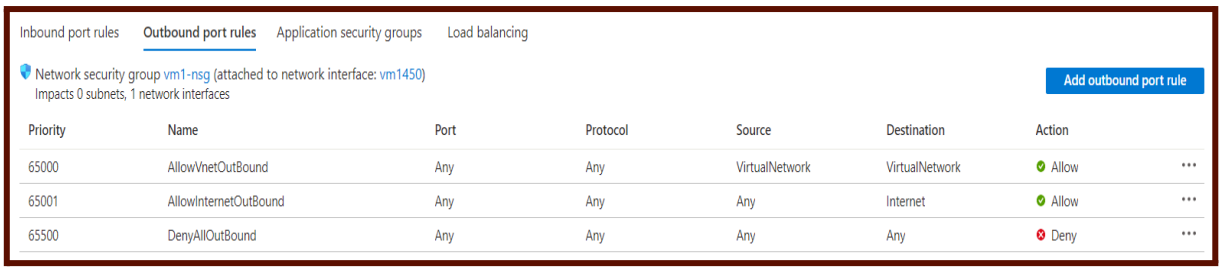
Property	Explanation
Name	A unique name within the network security group. The name can be up to 80 characters long. It must begin with a word character, and it must end with a word character or with '_'. The name may contain word characters or '.', '-', '_'.
Priority	A number between 100 and 4096. Rules are processed in priority order, with lower numbers processed before higher numbers, because lower numbers have higher priority.
Source or destination	Any, or an individual IP address, classless inter-domain routing (CIDR) block (10.0.0.0/24, for example), service tag, or application security group. If you specify an address for an Azure resource, specify the private IP address assigned to the resource.
Protocol	TCP, UDP, ICMP, ESP, AH, or Any. The ESP and AH protocols aren't currently available via the Azure portal but can be used via ARM templates.
Direction	Whether the rule applies to inbound, or outbound traffic.
Port range	You can specify an individual or range of ports. For example, you could specify 80 or 10000-10005. Specifying ranges enables you to create fewer security rules.
Action	Allow or deny

Source: Microsoft Docs → Azure network security groups overview | Microsoft Learn

Please refer to this link → Creating and Configuring -Azure Network security group

Default security rules



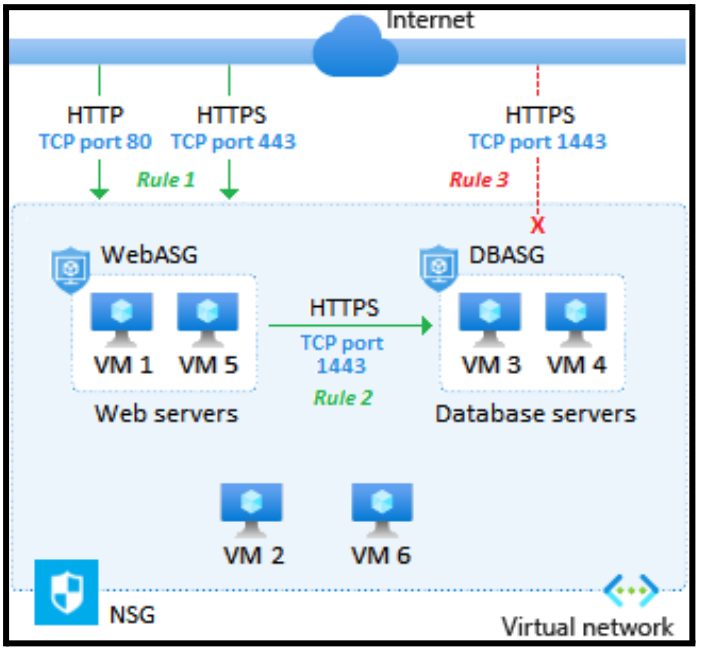


Application Security Groups(ASG)

Application Security Group (ASG) is part of the Network Security Group(NSG). You can define your network security group rules based on your application security groups.

Application security groups allow you to configure network security as a natural extension of an application's architecture, allowing you to group virtual machines and define network security policies based on those groups.

Let's take a look at how to implement application security groups by creating a configuration for an online retailer. In the below example. We need to control network traffic to virtual machines in application security groups.

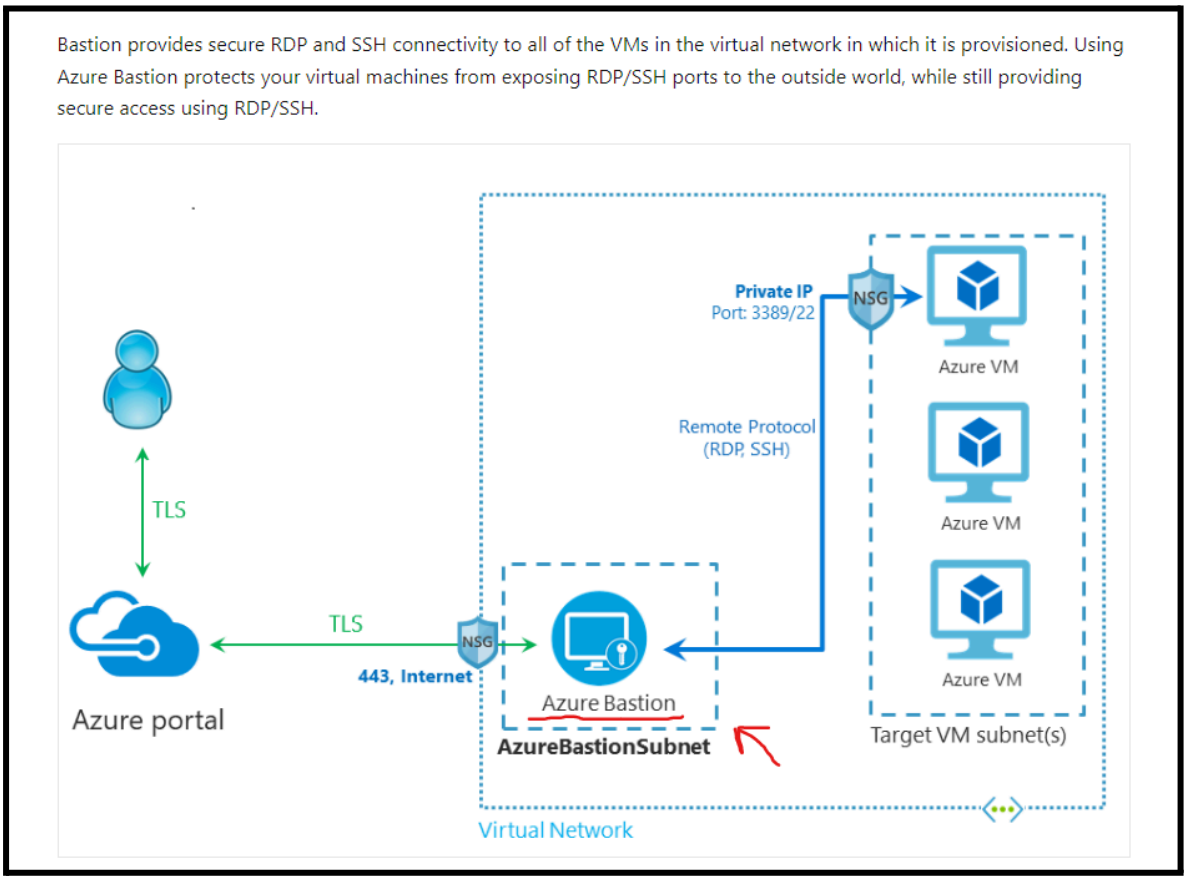


Application security groups work similarly to network security groups, but they provide an application-centric way of looking at your infrastructure. You join your virtual machines to an application security group. Then you use the application security group as the source or destination in the network security group rules.

Azure Bastion Implementation

Azure Bastion is a service you run that allows you to connect to a virtual machine using your browser and the Azure portal, or through a native SSH or RDP client already installed on your local computer.

- It is a fully managed PaaS that you run/provision inside your virtual network.
- It provides secure, seamless RDP/SSH connectivity to your VMs directly from portal over TLS.
- When you connect through this, your VM doesn't need a public IP, agent, or client software.



Prerequisites

- A virtual network. This will be the VNet to which you deploy Bastion.
- A virtual machine in the virtual network. This VM isn't a part of the Bastion configuration and doesn't become a bastion host. You connect to this VM later in this tutorial via Bastion.
- Required VM roles: Reader role on VM & Reader role on the NIC(with private IP of VM).
- Required inbound ports: For Windows VMs - RDP (3389) & For Linux VMs - SSH (22).

Pls refer to this → Tutorial: Deploy Bastion using specified settings: Azure portal

Configuring Service Endpoints for Azure PaaS

Service Endpoints in Azure enable secure and private connectivity between your virtual network and Azure PaaS services. This enhances security by preventing public internet exposure of your PaaS resources.

Here's a step-by-step guide on how to configure Service Endpoints:

1. Create or Select a Virtual Network

- **Azure Portal:**
    - Navigate to the Azure portal.
    - Search for and select "Virtual networks."
    - Create a new virtual network or select an existing one.

2. Create a Subnet

- **Azure Portal:**
    - In the selected virtual network, go to the "Subnets" section.
    - Click "Add subnet."
    - Provide a name for the subnet and an address range.

3. Enable Service Endpoint

- **Azure Portal:**
    - In the subnet's settings, go to the "Service Endpoints" section.
    - Click "Add service endpoint."
    - Select the desired PaaS service (e.g., Storage, SQL Database, Cosmos DB).

4. (Optional) Configure Network Security Groups (NSGs)

- **Azure Portal:**
    - In the subnet's settings, go to the "Network security group" section.
    - Associate an NSG to the subnet and configure inbound and outbound security rules to further restrict traffic.

Important Considerations:

- **Regional Alignment:** Ensure that the virtual network and the PaaS service are in the same region.
- **Firewall Rules:** If the PaaS service has firewall rules, configure them to allow traffic from the virtual network's IP addresses.
- **Private Link:** For more granular control over access, consider using Azure Private Link to create private endpoints to your PaaS resources.

Configuring Private Endpoints for Azure PaaS

Private endpoints in Azure enable secure and private connectivity to Azure PaaS services in a virtual network. This eliminates the need for public internet exposure, improves security, and reduces latency.

Here's a step-by-step guide on how to configure Private Endpoints:

1. Create or Select a Virtual Network

- **Azure Portal:**
  - Navigate to the Azure portal.
  - Search for and select "Virtual networks."
  - Create a new virtual network or select an existing one.

2. Create a Private Endpoint

- **Azure Portal:**
  - In the virtual network, go to the "Private endpoints" section.
  - Click "Create private endpoint."
  - Select the target resource type (e.g., Storage Account, SQL Database, Cosmos DB).
  - Select the target resource.
  - Select the subnet where the private endpoint will be created.
  - Configure any necessary DNS settings.

3. Configure DNS Settings (Optional):

- For seamless integration, configure DNS settings within your virtual network to resolve the private endpoint's IP address. You can use Azure DNS or a third-party DNS solution.

Important Considerations:

- **Regional Alignment:** Ensure that the virtual network and the PaaS service are in the same region.
- **Firewall Rules:** If the PaaS service has firewall rules, configure them to allow traffic from the private endpoint's IP address.
- **NSGs:** You can use NSGs to further restrict traffic to the private endpoint.

Additional Tips:

- **Leverage Azure Private Link:** For more granular control over access, consider using Azure Private Link to create private endpoints to specific resources within a PaaS service.
- **Monitor and Optimize:** Monitor the performance and security of your private endpoints to identify and address any issues.
- **Consider Hybrid Connectivity:** If you need to connect on-premises resources to Azure PaaS services, explore options like ExpressRoute or VPN Gateway.

By following these steps and considering the additional tips, you can establish secure and private connections to your Azure PaaS services, enhancing the overall security and performance of your applications.

Private Endpoints

Private endpoint is a network interface(NIC) that uses a private IP address from your virtual network.

This network interface(NIC) connects you privately and securely to the service powered by Azure Private Link. By enabling a private endpoint, you are bringing the service into your virtual network.

These are the properties of Private Endpoints:

Property	Description
Name	A unique name within the resource group.
Subnet	The subnet to deploy, where the private IP address is assigned. For subnet requirements, see the Limitations section later in this article.
Private-link resource	The private-link resource to connect by using a resource ID or alias, from the list of available types. A unique network identifier is generated for all traffic that's sent to this resource.
Target subresource	The subresource to connect. Each private-link resource type has various options to select based on preference.
Connection approval method	Automatic or manual. Depending on the Azure role-based access control (RBAC) permissions, your private endpoint can be approved automatically.
Request message	You can specify a message for requested connections to be approved manually. This message can be used to identify a specific request.
Connection status	A read-only property that specifies whether the private endpoint is active. Only private endpoints in an approved state can be used to send traffic.

Service Endpoints

service endpoint provides secure and direct connectivity to Azure services in an optimized path through the Azure backbone network.

These allow you to secure your critical Azure service resources only to your virtual networks. These allow private IP addresses in a VNet to reach the endpoint of an Azure service without requiring a public IP address in the VNet.

Service endpoints are available for the following Azure services and regions.

Service endpoints characteristics

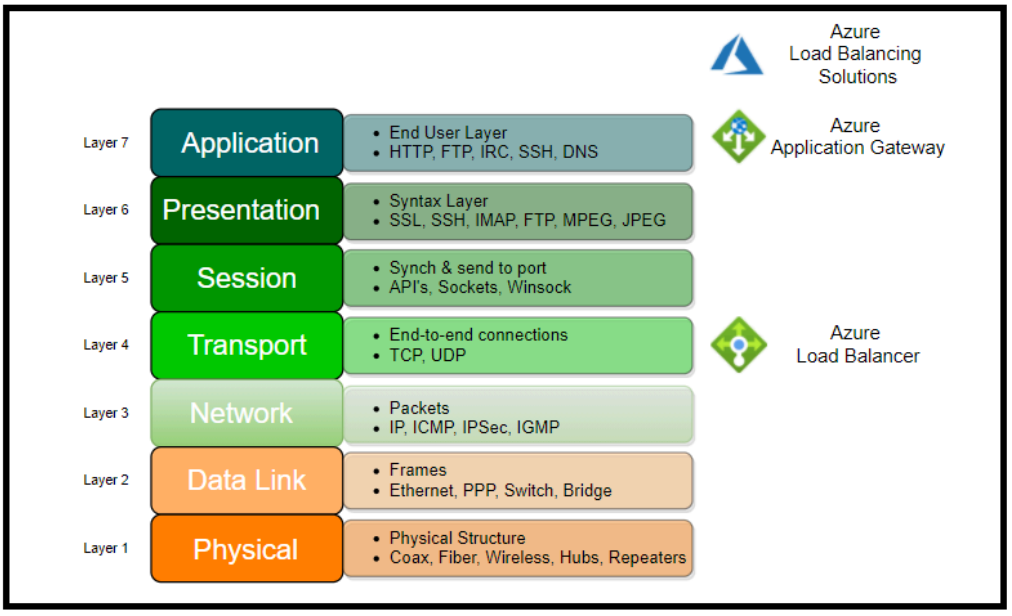
- Extends your virtual network identity to your Azure services to secure your service resources.
- You can secure your Azure service resources to your virtual network by using virtual network rules.
- VNet rules can remove public internet access to resources & allow traffic only from your VNet
- It takes service traffic directly from your VNet to service on the Microsoft Azure backbone network.
- Service endpoints are configured through the subnet (No extra overhead is required)

C) Configure Load Balancing

Introduction to Azure Load Balancer

Azure Load Balancer provides high availability and network performance for your applications. Administrators use these load balancers to efficiently distribute incoming network traffic across back-end servers and resources. It is implemented using load balancing rules & health probes.

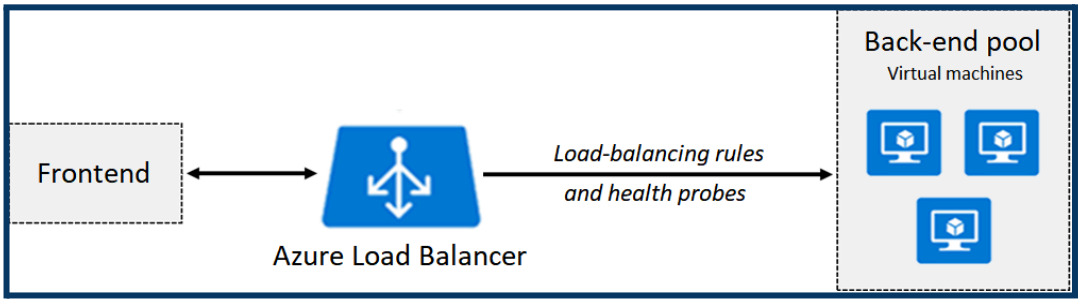
- Azure provides load balancing at **Layer 7** which is the application layer via Azure Application Gateway. This is typically http traffic.



- Azure also provides load balancing at Layer 4 which is a transport layer consisting of

TCP and UDP protocols. This is the Azure Load Balancer.

- We could use the Azure Load balancer for both public facing as well as internal application. The load balancer is set up with a backend pool which distributes traffic to a set of VMs or VM Scale sets.



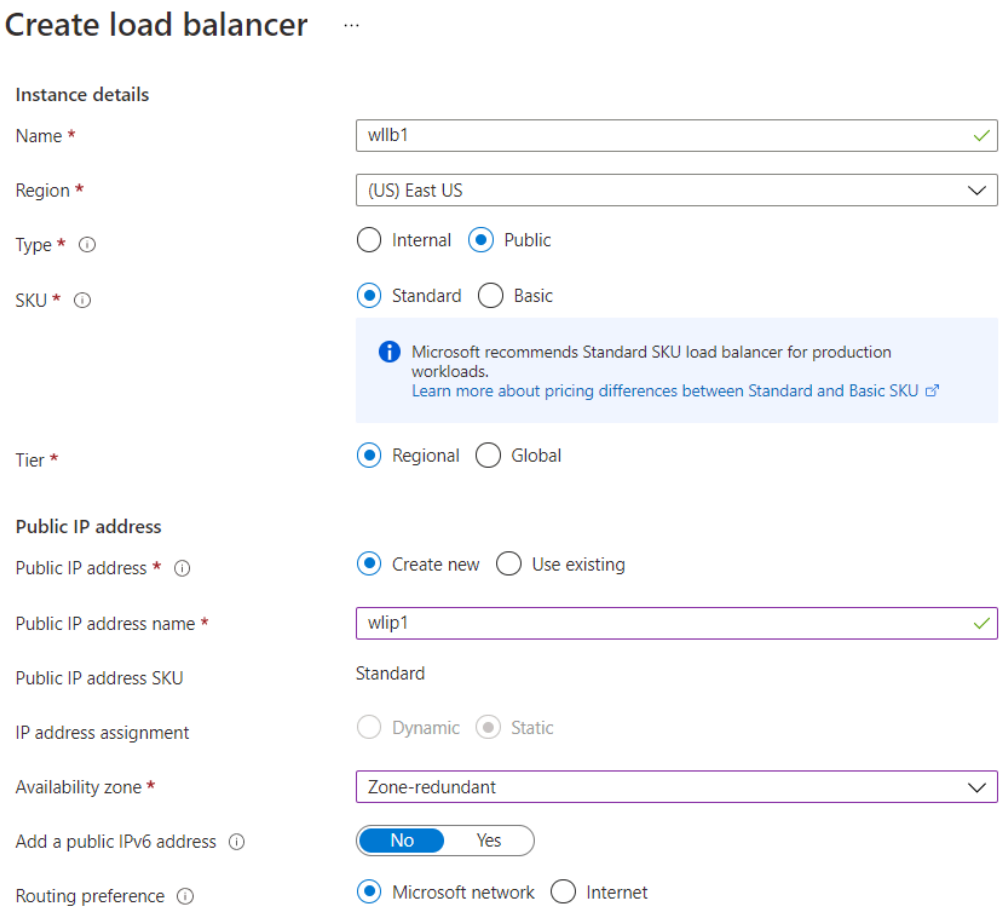
(**Source: Microsoft Documentatio)**

Here are the steps to create a load balancer:

Step 1: Create Load Balancer

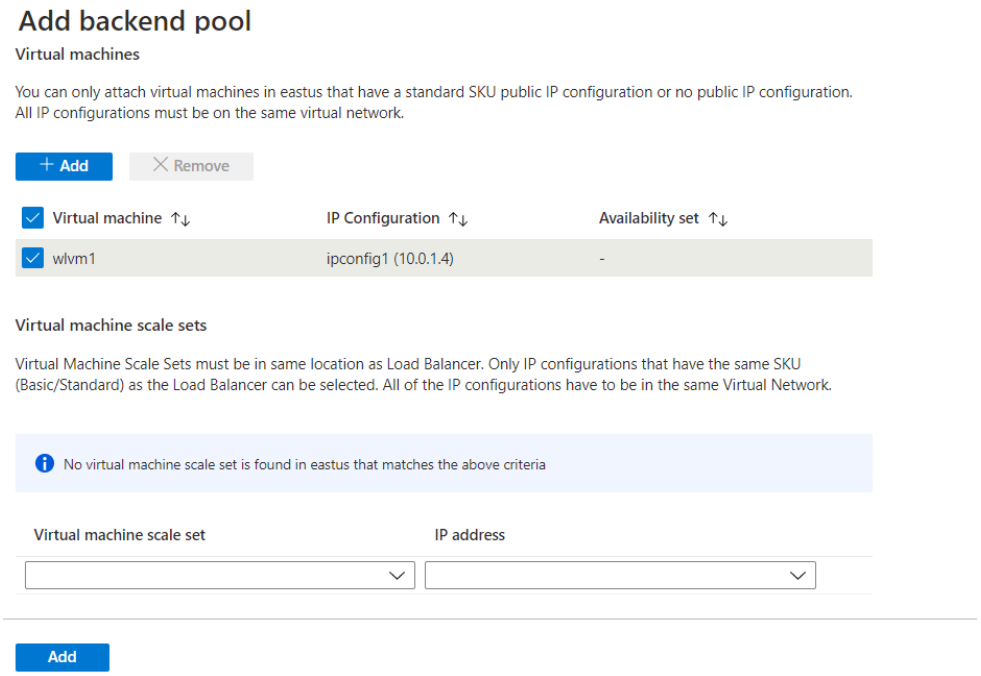
We create a load balancer with the following options:

- Name for the load balancer
- Internal or Public load balancing
- SKU type could be Standard or Basic. Since Basic does not have an SLA, Standard SKU type is recommended for Production workload which has SLA of **99.99%.** *Standard SKU* comes with many more additional / better features than Basic SKU like https.
- **Regional or Global** – This is a new feature and is available for Public Load balancers



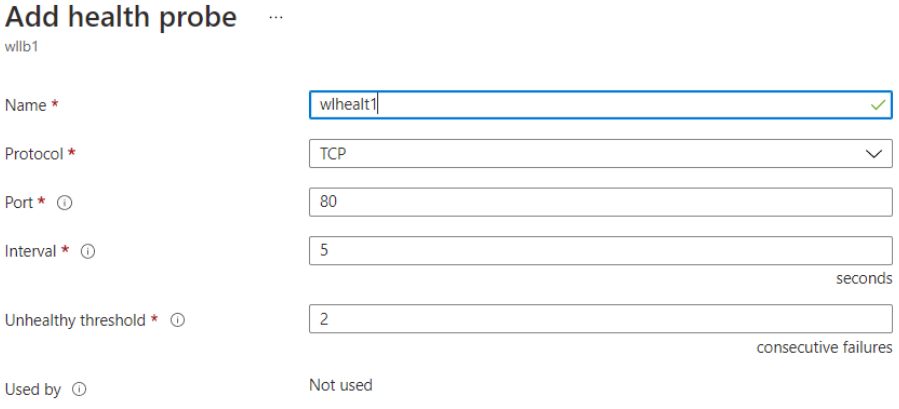
Step 2: Create Backend pool

- We create a backend pool where we attach VMs or VMSS
- VMs/ VMSS have to be in the same location.
- We could add multiple backend pools



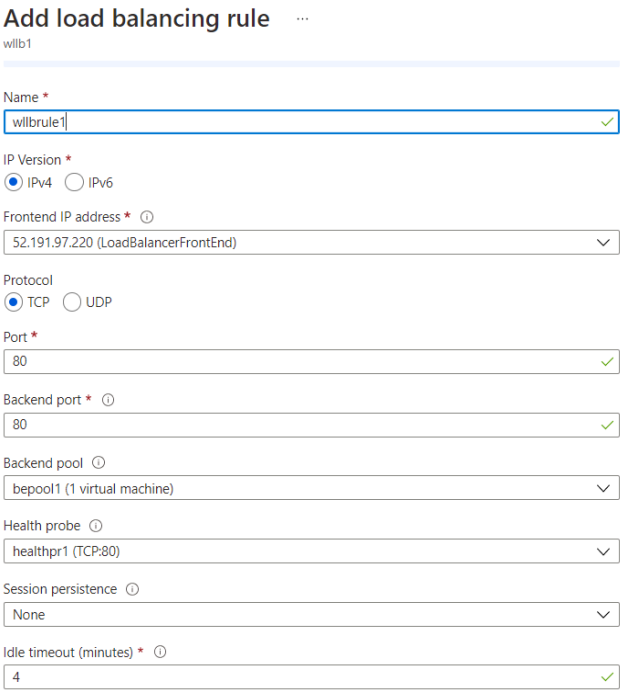
Step 3: Add Health Probe

- We need to add a health probe
- We can configure **TCP/HTTP/HTTPS** as protocol
- We add a port number
- We add an interval and unhealthy threshold which is the interval for checking where the probe passes a health check. The unhealthy threshold is the number of times a probe is allowed to fail consecutively after which the instance will be marked as unhealthy and traffic routing will be stopped.



Step 4: Add Load Balancing rule

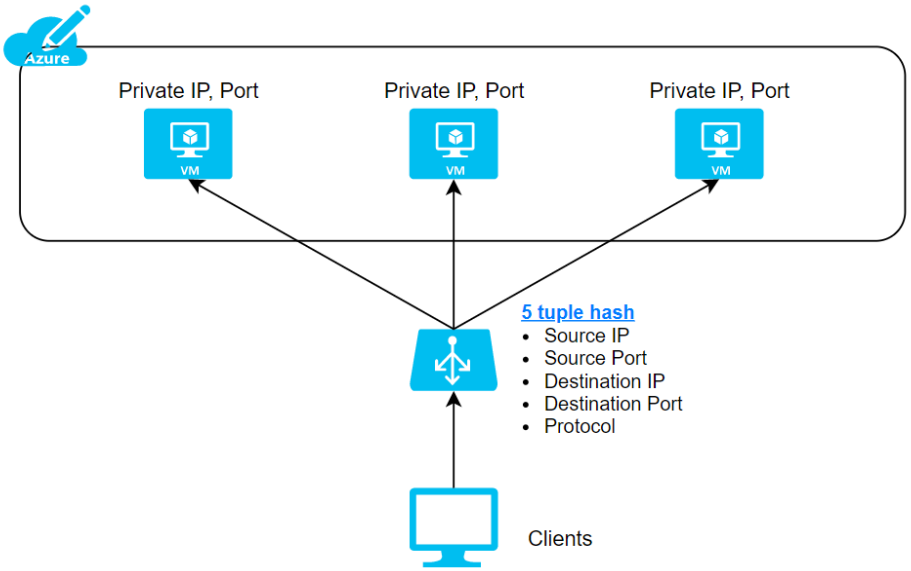
- We create a load balancing rule
- We specify frontend IP address and Protocol (TCP or UDP) and Port
- We specify the Backend port and pool
- We specify health probe
- We can also specify session persistence. If this option is enabled, the traffic will be routed to the same VM.



Azure Load Balancer can also be configured to use as follows to map traffic to the available servers:

- **2 tuple (Source IP, Destination IP)**
- **3 tuple (Source IP, Destination IP, Protocol)**
- **5 tuple (Source IP, Source Port, Destination IP, Destination Port, Protocol)**

Please see how the traffic is routed based on the 5 tuples.

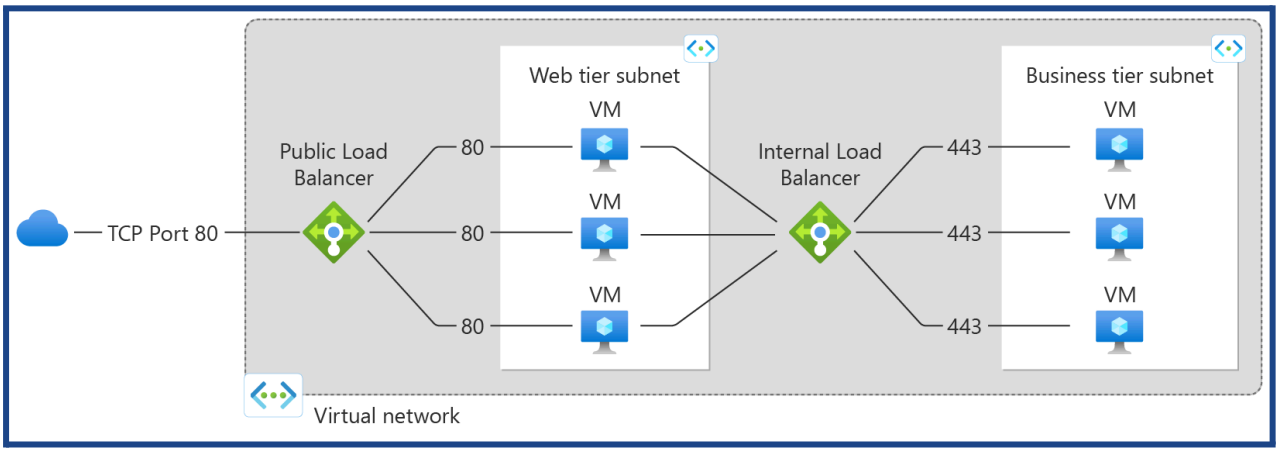


Configure an internal or public load balancer

Azure Load Balancer operates at Layer 4 of the Open Systems Interconnection (OSI) model. It is the single point of contact for clients. The load balancer distributes inbound flows that reach the backend pool instances from the front end of the load balancer.

- Public load balancer can provide outbound connections for VMs inside your virtual network.
- Internal (or private) load balancer is used where private IPs are needed at the frontend only.

Public load balancer	Internal load balancer
Frontend IP configuration: Public IP address	**Frontend IP configuration:** Private IP address
Description: A public load balancer maps the public IP and port of incoming traffic to the private IP and port of the VM. Load balancer maps traffic the other way around for the response traffic from the VM. You can distribute specific types of traffic across multiple VMs or services by applying load-balancing rules. For example, you can spread the load of web request traffic across multiple web servers.	**Description:** An internal load balancer distributes traffic to resources that are inside a virtual network. Azure restricts access to the frontend IP addresses of a virtual network that are load balanced. Front-end IP addresses and virtual networks are never directly exposed to an internet endpoint, meaning an internal load balancer cannot accept incoming traffic from the internet. Internal line-of-business applications run in Azure and are accessed from within Azure or from on-premises resources.
SKUs supported: Basic, Standard	**SKUs supported:** Basic, Standard



Troubleshoot the Azure Load balancer

The below Troubleshoot points are related to the main important issues during the Load Balancing

- ★ No outbound connectivity from Standard internal Load Balancers (ILB)
- ★ Can't change backend port for existing LB rule of a load balancer that has Virtual Machine Scale Set deployed in the backend pool.
- ★ Small traffic is still going through the load balancer after removing VMs from the backend pool of the load balancer.
- ★ Additional network captures
- ★ Load Balancer in failed state

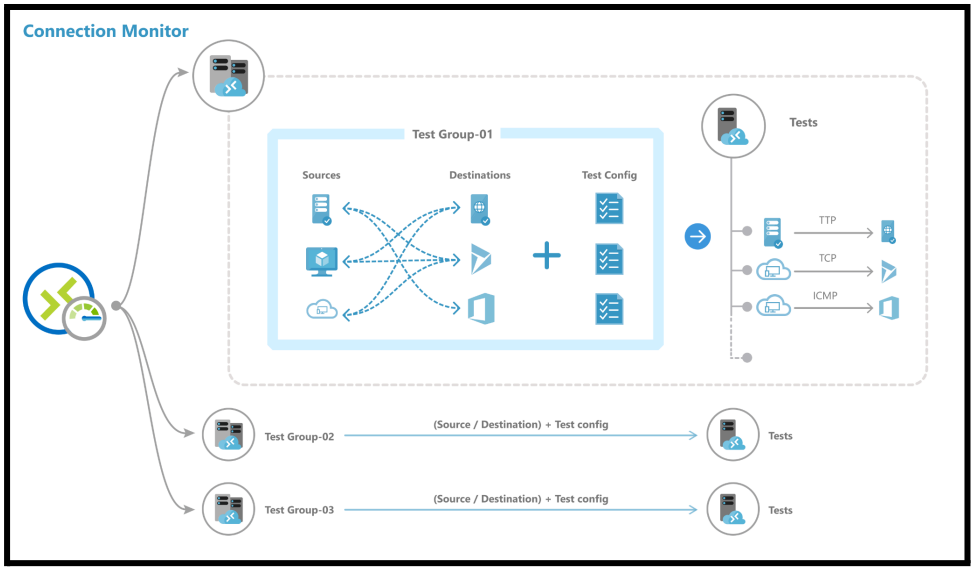
D) Monitor resources by using Azure Monitor

Connection Monitor

Azure Monitor can be used to monitor on-prem servers and for this you need to install an agent on on-premises server and remove the telemetry you want.
Additionally, you need to install it on servers located in the public cloud.

Connection Monitor provides unified, end-to-end connection monitoring. The Connection Monitor feature supports hybrid and Azure cloud deployments.

The below are few important definitions that are related to the Connection Monitor



Connection monitor resource: A region-specific Azure resource. All the following entities are properties of a connection monitor resource.
Endpoint: A source or destination that participates in connectivity checks. Examples of endpoints include: Azure VMs, Azure virtual networks Azure subnets, On-prem agents, On-prem subnets, On-prem custom networks that include multiple subnets, and URLs/IPs
Test configuration: A protocol-specific configuration for a test. Depending on the protocol you choose, you can define the port, thresholds, test frequency, and other elements.
Test Group: The group that contains source endpoints, destination endpoints, and test configurations. A connection monitor can contain more than one test group.
Test: The combination of a source endpoint, destination endpoint, and test configuration. A test is the most granular level at which monitoring data is available.

Configure and use Azure Monitor for networks

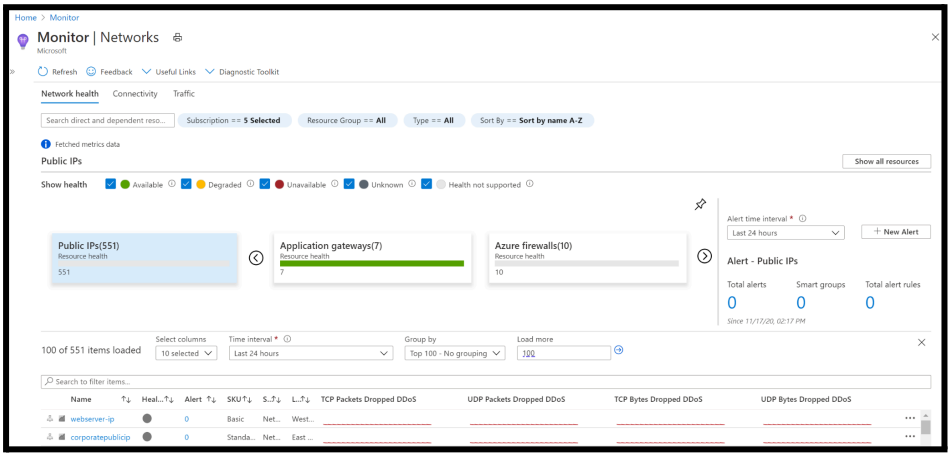
Azure Monitor Network Insights provides a comprehensive and visual representation of topologies, health and metrics for all network resources with no configuration required.

It also provides access to network monitoring capabilities such as connection monitor, flow logging for network security groups (NSGs), and traffic analytics. And it provides other network diagnostic features. You can use Network Insights for Monitoring the Virtual networks.

The below are few important key components of monitoring Vnets using Network Insights

- Topology, Network health and metrics, Connectivity, Traffic and Diagnostic Toolkit

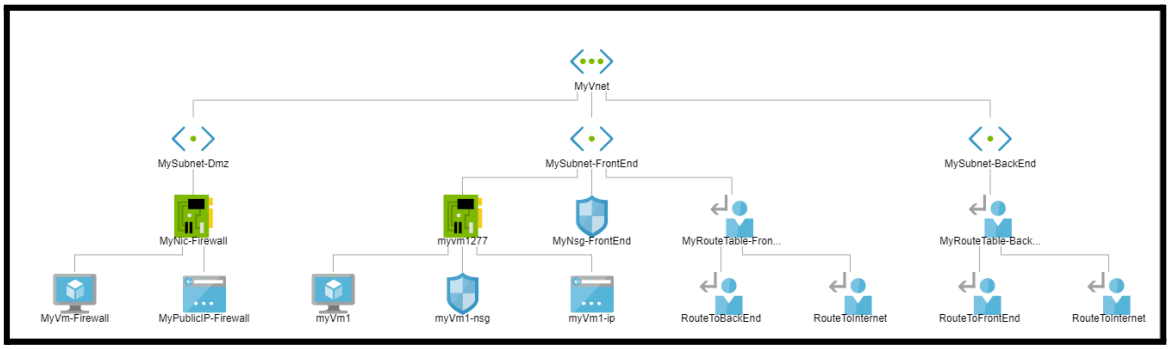
The Azure Monitor Network Insights overview page provides an easy way to visualize a list of your networking resources, along with resource health and alerts. It is divided into four key functional areas: search and filtering, resource health and metrics, alerts and resource view.



(Source: Microsoft Documentation)

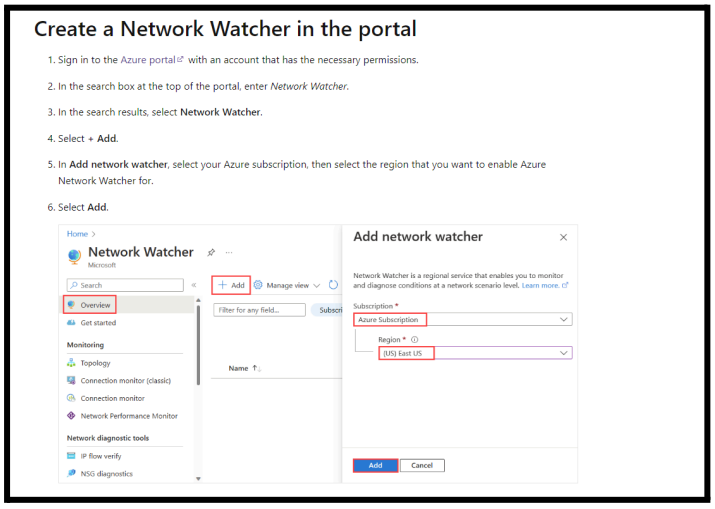
Use Azure Network Watcher

Network Watcher provides tools to monitor, diagnose, and view connectivity metrics for your Azure deployments. Azure Network Watcher provides tools to monitor, diagnose, view metrics, and enable or disable logs for resources in an Azure virtual network.



Network Watcher is designed to monitor and repair the network health of IaaS (Infrastructure-as-a-Service) products, including virtual machines (VMs), virtual networks (VNets), application gateways, load balancers, etc.

You can use this for : Monitoring, Diagnostics, Metrics, Network Monitoring Logs



Troubleshoot virtual network connectivity →

Troubleshoot connections - Azure portal - Azure Network Watcher | Microsoft Learn
Troubleshooting connectivity problems between Azure VMs

Troubleshoot external networking →

Troubleshoot Azure point-to-site connection problems - Azure VPN Gateway
Troubleshoot an Azure site-to-site VPN connection that cannot connect - Azure VPN Gateway

Extra Learning Purpose (Not part of the Syllabus)

Azure Storage Services

Azure Disk Storage

- VMs in Azure use two types of disks. One is an operating system disk, and the other is a temporary disk.
- The operating system with and without customization is stored as an image and loaded when the VM is built.
- Both the image and the operating system disk are virtual hard disks and are stored in a Storage account.
- The temporary disk will be stored as part of the hardware itself to provide faster access.
- The virtual hard disks use .vhd files and are stored as page blobs.

Unmanaged Disks

➔ This is the traditional type of disk. Here we create the storage account and specify the storage account when we use the disk.

➔ If we have too many disks, then there will be contention, and VMs will throttle, which will impact the performance.

Managed Disks

➔ This is the latest and recommended type to allocate. If we have unmanaged disks, Azure gives us the option to migrate to managed disks.

➔ We don't need to specify a storage account or manage the storage account. Azure takes care of management, including scalability. We just need to give the size and performance tier.

➔ These are the types of managed disks.

- **Standard HDD** – These are standard magnetic drives and are the cheapest. We can offer Recovery services to replicate locally or be geo-redundant
- **Standard SDD** – These are more consistent and reliable, and suitable for web servers.
- **Premium SSD** – These are backed by solid-state drives and deliver high performance, low latency, and useful workloads that are I/O intensive, like production and performance-sensitive ones.
- **Ultra disk** – This is the latest type, which has a max iops of 160K. But these can be used as data disks only and not OS disks.

Azure Backup Service

- Azure provides an Azure backup service to perform backups.
- We need to install an extension and need to specify the frequency.
- The snapshot will be taken for the OS disk as well as the **data .disk**
- The snapshot taken here is different from the image. The disk is prepared to create an image, and no activity is allowed, and sysprep is done.

- Here, we allow the system to run in snapshotting, and we take either application-consistent snapshots or file consistent snapshots. These snapshots are moved into recovery service vaults.

- We can set up a recovery service vault to replicate to another region.
  For example, we are in the US East, and we replicate to the US West, which protects from entire East US failure.

FAQs

- **A company has SAP Hana and other top tier databases like SQL and Oracle. What is the recommended disk type?**
  - Please use Ultra disks for data disks. Use Premium SSD for OS disk.
- **A company has a disk requirement of more than 32TB. What are the available options?**
  - Please use Ultra disks or use mirroring with striping.
- **A company wants more than 50,000 IOPS but does not want to use Ultra disks. What can be done?**
  - Please use mirroring with striping. If one disk has 20K iops and you do striping with 2 disks, you will get 40K iops, and with 3 disks, you will get 60K IOPS
- **Will the disk be deleted when we delete a VM?**
  - No, you need to delete disks explicitly.
- **I had allocated 100 GB, but now I want to add 100 GB more. Can I do that on my existing machine?**
  - Yes, deallocate VM and update disk.
- **Can we cache data?**
  - Yes, disk caching can be set to NONE or READ ONLY or READ/WRITE. For log disks, use READ ONLY.

- **Can Multiple VMs read the disk on a given VM?**
    - Yes, we can enable disk sharing.

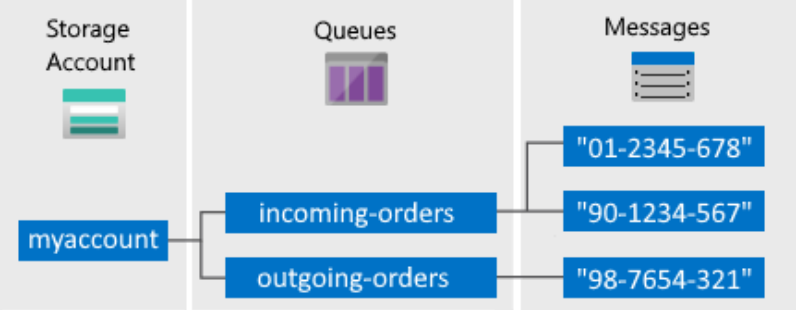
Azure Queue Storage

Queue Storage:

- This component of Azure storage is for messaging store and queuing.
- Simple and cheap
- The preferred workload of more than 80GB when compared to the Service Bus queue.
- Can Scale and message node failure will not affect Service since other nodes will process.
- Can add more worker nodes if there is a burst

The architecture of Queue storage

- We create a storage account.
- Within the storage account, we create Queues.
- For example, we create 2 queues, one incoming order and one outgoing payment.
- There will be messages which we will store under the queues.
- These messages will be read at least once and processed by the applications.



- **URL format:** Queues are addressable using the following URL format:
  http://<storage account>.queue.core.windows.net/<queue>
- The following URL addresses a queue in the diagram:
  http://myaccount.queue.core.windows.net/incoming-orders

Use Cases

- Provides a decoupling architecture. This allows for asynchronous communication.
- Let's take an example of a Purchase system integrated with a Shipping system.
- In the traditional model, both the purchase and shipping system is integrated.
- When a customer places an order, the purchase system sends the order to Shipping, and it has to get an acknowledgment.
- If there are too many orders and the shipping system does not acknowledge, it will break the system.
- In asynchronous communication, we decouple, and the purchase system does not wait for an acknowledgment.
- It will send a message, and the shipping system might check for the message queue every 5-10 minutes and process the orders. Here we use the Azure queue storage.

FAQs

- **Can we have ordering like FIFO for messages?**
  - No
- **Does Queue storage support transactions?**
  - No. Each message is independent. If 20 of 30 messages are read, and the operation fails, this is not an all-or-nothing situation to have a transaction concept to rollback 20 and process from the beginning.
- **Does Queue Storage push messages?**
  - No, it would help if you fetched the messages.
- **Can we lock messages for exclusive access?**
  - Yes, you need to acquire a lease during which period you have exclusive access.
- **What is the lease duration?**
  - 30 seconds default lease duration
  - 7 days max for lease duration
  - Can be renewed
  - The level is a message
- **Can we use batches for processing?**
  - Yes
- **Does Queue storage provide dead lettering?**
  - No
- **What are the limits?**
  - Max queue size – 500GB
  - Max message size – 64KB
  - Max number of queues – no limit

Azure Table Storage

- This component of Azure storage can be used as a **NoSQL** Datastore.
- It stores data in a key-value pair. We have a partition key and a row key. These are default columns. We can add columns as needed.
- We can query or insert data using Storage Explorer.

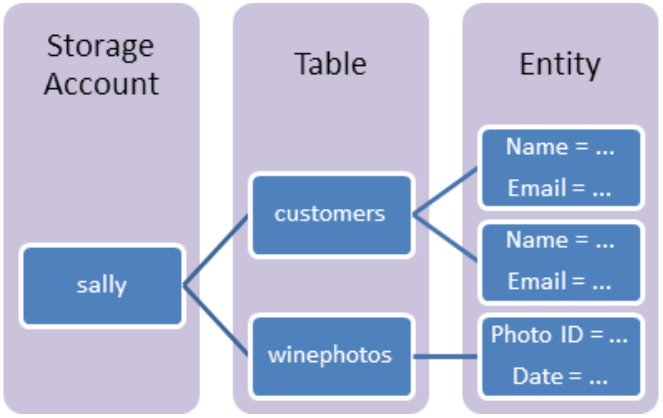


Table Storage architecture

- **URL format** for Azure Table Storage accounts: **http://<storage account>.table.core.windows.net/<table>**
- In the Storage account, we create an account (Sally)
- Under the account (Sally), we create a table (customers)
- Under the table (customers), we insert rows called Entities.
- Entities contain properties that are a key-value pair.
- Therefore Storage *Account -> Table -> Entities -> Properties*

Use Case

- Use Table storage for storing semi-structured data
- Use this for creating an app that needs a flexible data schema.

FAQs

- **How much can we store?**
  - We can store Petabytes of data.
- **Is availability a concern?**
  - With GRS, data is replicated 3 times within a region and another 3 times in an additional region. So it is highly available.
- **What is Cosmos DB table API?**
  - Cosmos has several APIs like Mongo/SQL/Gremlin, and one of the supported APIs is Table API. Both Azure Table storage and Cosmos DB table API have the same data model and support the same operations like query insert via SDK. Using the Cosmos DB table API will increase the performance like single-digit ms latency, scalability, global distribution, etc.

Azure Archive Storage

- Use Azure archive storage for rarely accessed data.
- Lowest priced storage tier
- Automatic encryption of data
- Seamless integration with Hot and cool storage tiers
- Secure data transfer with HTTPS.

- Minimum **180** days storage requirement – If we move before that, we pay early deletion fees for the number of days falling short.

Use Cases

- **Archival**
  - Healthcare and other regulations like SOX (financial records etc.) require that information be stored for multi-year periods. This provides long term compliant storage.
- **Long term Backup Retention**
  - There might be a requirement to store Database, server, desktop data for multi-years. This provides long-term storage freeing up local disk space.
- **Magnetic tape replacement**
  - If your organization has a VTL (Virtual tape library), you can move the least accessed data to archive storage.
- Other use cases are Security/Public safety data and other digital media content retention.

FAQs

- **What types of storage can be stored in Archive Storage?**
  - Only Blob storage

- **What are the retrieval options?**
  - There are two options.
    - *Standard Priority (Default)* – up to 15 hours
    - *High Priority (Max 10 GB)* – less than 1 hour

- **What are the fees associated with Archive Storage?**
  - The fess is as follows:
    - Data Retrieval – Standard – 1.3220$/GB
    - Data Retrieval – High Priority – 6.6097$/GB
    - Write Operation – 6.6097$/10000
    - List/Create container operation – 3.3049$/10000
    - Read Operation – Standard – 330.4813$/10000
    - Read Operation – High Priority – 3304$/10000

Azure Key Vault

Best practices dictate that we never hard-code sensitive information like password-strings etc., in our code. If we do so and store the code in Github, the information could be leaked and misused. Even the connection strings like urls for databases or even IP addresses or our servers must be protected.

Azure has a secret store called Azure Key Vault, which stores our secrets and passwords. One could never be able to read the secret but will be able to use it with the right set of permissions.

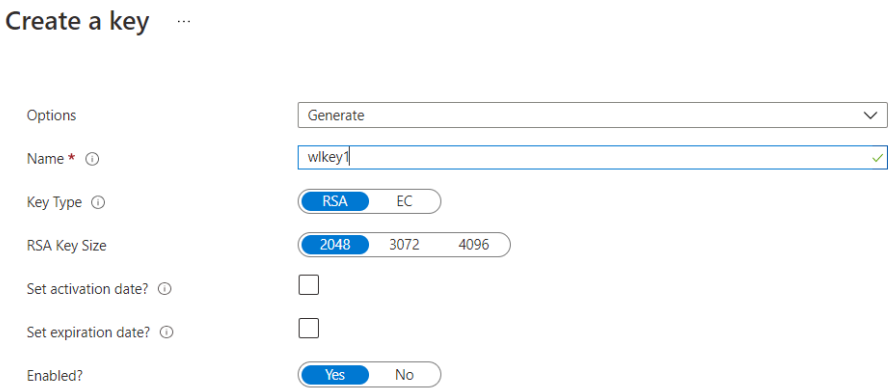
Azure Key Vault is a **PaaS platform in Azure. It** is integrated into Microsoft Entra ID. We can store secrets, Keys, and certifications and have multiple versions stored. We have audit logs as a feature. Azure Key vault is **FIPS 140-2** compliant.

Secrets

- We can store up to 25kb in size.
- We can store plain text passwords, connection strings, JSON, XML, and more.
- We can have an activation date and expiry date.
- We can create as enabled or not if we don't have immediate use etc

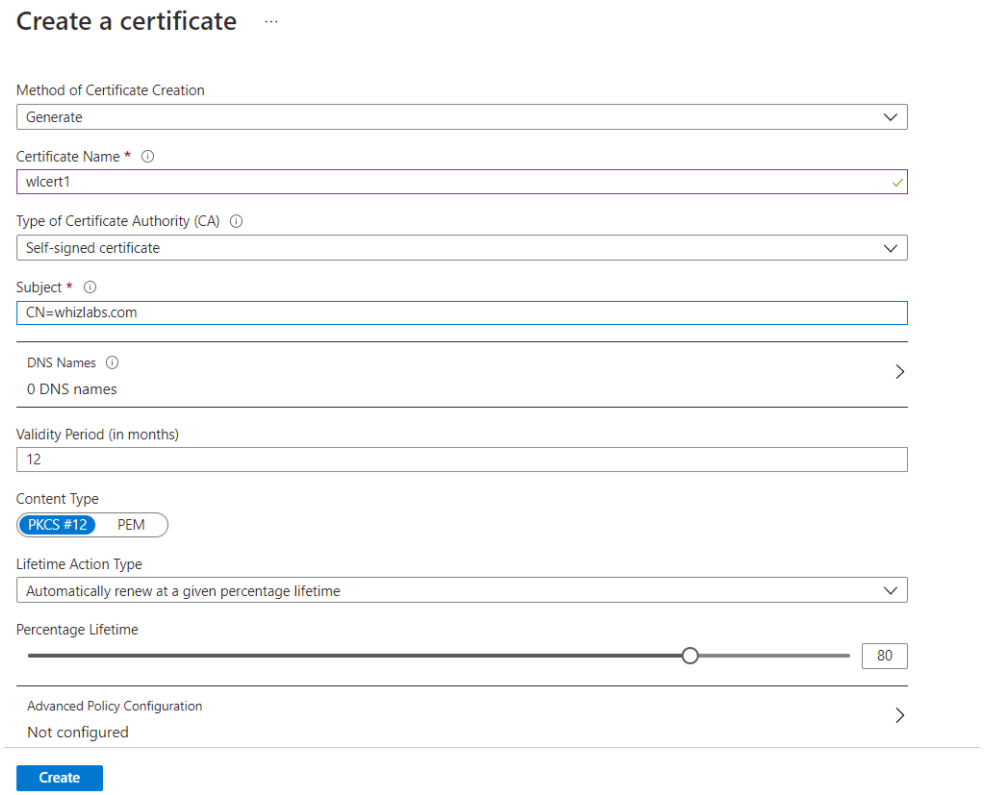
Keys

- A Key is typically asymmetric in the **PKI (Private Key Infrastructure)**. Here we have a public key and a private key. The public key is known to all, and anybody can use it to encrypt the data. But the private key is known only to the owner, and only the private key can decrypt the data.
- Azure will generate the private and public keys, but the private keys will never be disclosed.
- We could also use symmetric keys for storage and SQL data, and in this case, the symmetric key would be wrapped with an asymmetric key making it secure.
- The key type could be **RSA/EC** and **2/3/4 kb** in size.
- We can have an activation date and expiry date.
- We can create as enabled or not if we don't have immediate use etc.



Certificates

- We could either generate our keys or import keys.
- Keys could either be self-signed or use a CA (Certification Authority) like Digicert or GlobalSign, etc.
- We can have validity between 1 month to 10 years.



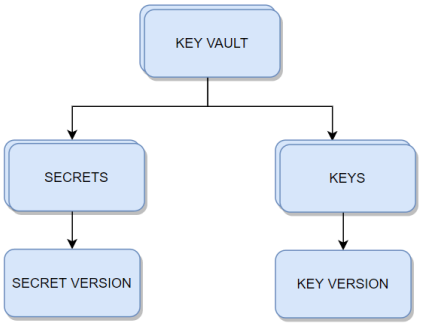
Audit

Since all activity takes place within the Azure Key vault, we can audit all types of usage. We can see who is using and type of activity.

Versioning

It is always recommended to keep changing the secrets. This will help protect in case the secrets were leaked to limit the damage. To do this, we can create a new version. Also, we need to automate the process so that we don't forget to do it.

Azure Key Vault has a **unique versioning engine.** We can rotate secrets and keys, and new versions are created. When we have used an older version of the key to encrypt, we will be able to point to the older version and decrypt the data.



Access Policy

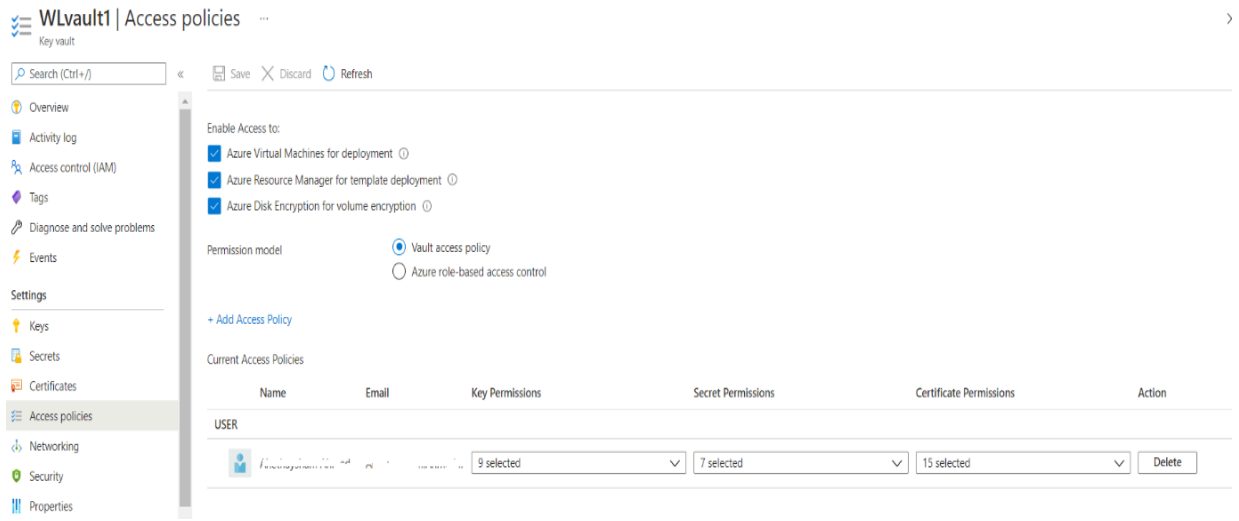
We can set access policies at the key vault level and more granularly at the **Key/Secret** and Certificate level.

We can enable access to:

- Azure Virtual Machines for deployment

- Azure Resource Manager for template deployment
- Azure Disk Encryption for volume encryption

The above will allow the usage of the key vault for the VMs/ disk and other deployments to be attached automatically.



Also, we can have access granted via the key vault policy or via RBAC.

FAQs

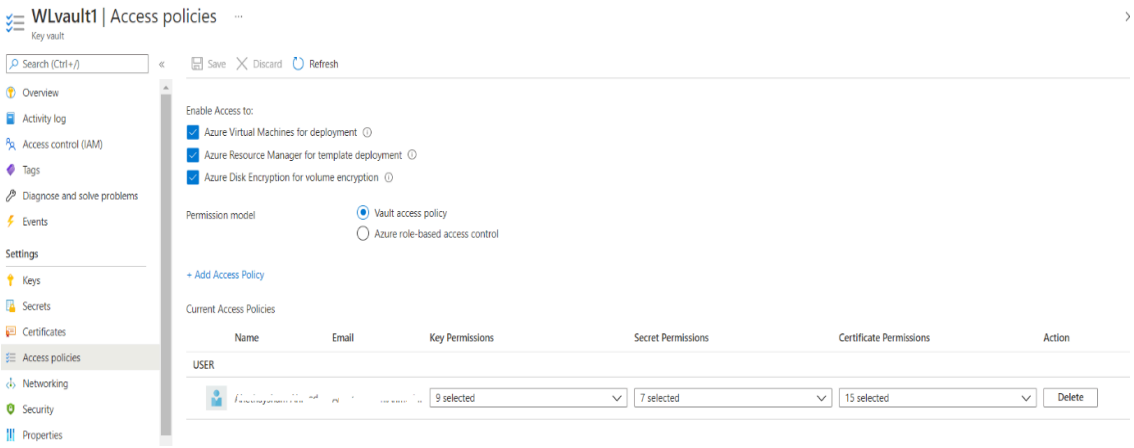
1) **We want to have a different set of access for different secrets to the same individual. How do we achieve it?**
   Create another key vault and grant access.

2) **Is Key Vault regional or global?**
   Though Key Vault is global, use key vault in the region where your data resides to reduce latency.

3) **When do we choose the access policy and when to choose RBAC?**
   There are two Access planes – one is the Management plane, and the other is the Data plane.
   a. **Management Plane**
      i. This ties to the key vault level
      ii. Operations are create/update/delete of Key vaults/ access policies / tags etc
      iii. They don't involve with what's inside the key vault, i.e., the actual content
      iv. This is controlled by RBAC only
   b. **Data Plane**
      i. This deals with secrets/Keys/Certificates
      ii. Example for Keys - encrypt, decrypt, list, delete, backup, etc
      iii. Example for Certificates - get, list, create, import, update, delete, recover
      iv. Example for Secrets - get, list, set, delete, recover, backup, restore, purge

v. This can be controlled by either RBAC or Key Vault access policy

4) **My RBAC roles for Key vault management are not working. What could be the problem?**
   a. Please see the permission model below is selected for Vault access Policy and not Azure RBAC. Please change to RBAC and retry.



5) **Please list RBAC roles for key vault Management?**
   a. Key Vault Administrator
   b. Key Vault Certificates Officer
   c. Key Vault Crypto Officer
   d. Key Vault Crypto Service Encryption User
   e. Key Vault Crypto User
   f. Key Vault Reader
   g. Key Vault Secrets Officer

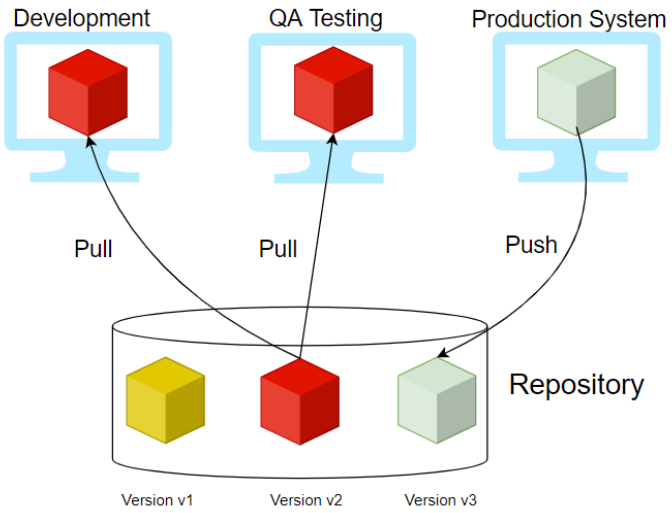
Azure Container Registry

What is a Container Registry?

A Container Registry is a central repository to store and distribute container images. A container image includes all the data needed to start a container - **for example**, the operating system, libraries, runtime environments, and the application itself.

We first build an image, and then we push the image to the repository. When needed, we pull the image into the target environment. With versioning as a feature, we have multiple versions of the container, and the different versions like the stable version would be used for Production.

Versions being tested would be in non-production regions. In the example below, v2 is a stable version, and the developer makes changes and creates v3. Once v3 is tested, it would be then pulled into Production.



Providers

Few providers provide the container registry services, and they are:

- **Docker Hub**
- **Azure ACR (Azure Container Registry)**
- **AWS ECR (Elastic Container Registry)**
- **Github Container Registry**
- **Google Container Registry**

	Amazon ECR	Docker Hub	GitHub Container Registry	Azure Container Registry (ACR)
Public Repository	No	YES	YES	No
Private Repository	Yes	YES	YES	Yes
Pricing (Public Repository)		$0	$0	$0
Pricing (Private Repository)	$	$$$	$$	
	Storage: $0.10 per GB, Data Transfer: $0.09 per GB	>= $7 per user/month	Storage: $0.25 per per GB, Outgoing Data Transfer: $0.50 per GB	Storage: $0.09 per GB
Authentication	AWS IAM	Password or Access Token	Personal Access Token (PAT)	PAT
MFA for Image Push/Pull	Yes	NO	NO	NO
SLA Availability	99.9%	N/A	N/A	99.9%
General Available	YES	YES	Beta	YES

Immutable Images	YES	NO	NO	YES
Image Scanning	YES	YES (paid plans only)	NO	YES
Regions	Choose between one of 25 regions worldwide	Not Known	Not Known	33 regions
Rate Limits	Pull: 1,000 per second, Push: 10 per second	Pull: 100/200 (Free Plan), unlimited (Paid Plan)	n/a	Pull: 1,000 per second, Push: 100 per second

ACR Service Tiers

ACR is available in 3 service tiers, also called SKUs.

1. **Basic** – Cost Optimized for developers
2. **Standard** – All features of Basic plus increased storage and image throughput. For Production
3. **Premium** – highest amount of storage and concurrent operations. It also includes geo-replication, content trust, and private link

ACR Roles

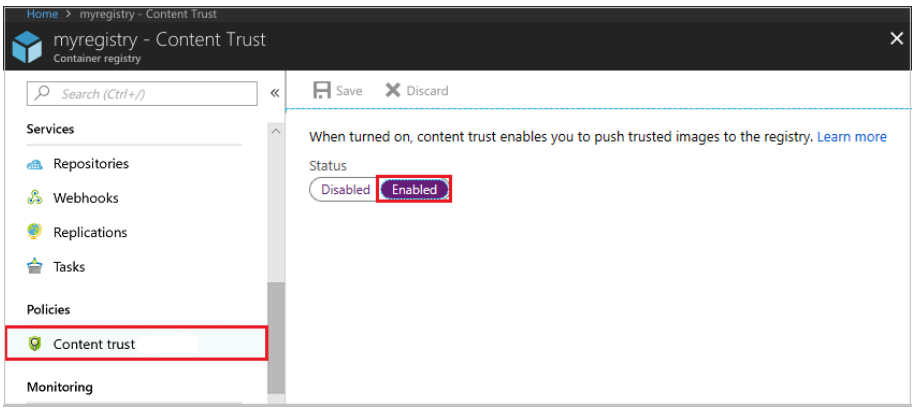
Role/Permission	Create/Delete ACR	Push	Pull	Signature Signing
Owner	X	X	X	
Contributor	X	X	X	
Reader			X	
AcrPush		X	X	
AcrPull			X	
AcrImageSigner				X

FAQs

1. **Can we change Service tiers?** –
   Yes

2. **What is geo-replication?**
   With this feature, a replica of the ACR will be created for DR purposes and local use.

3. **How can we secure the images in ACR?**
   There is a concept called CONTENT TRUST. With this, images will be signed with certificates. To enable this feature, enable registry content trust. It is available under **Policies -> Content Trust -> Enabled and then save.**



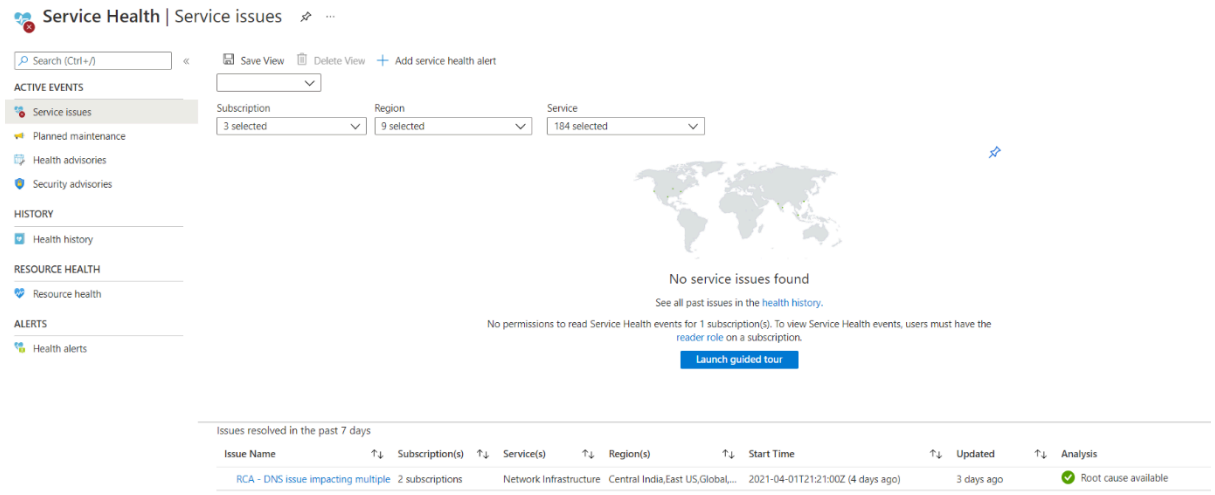
Azure Service Health

- Azure Service heath is a personalized dashboard that shows the service issues that affect you.
- It is able to dynamically do this to all the regions that we have resources in and all the resources that we have allocated for our subscriptions.
- We could even configure and add/remove regions or services or simply add all of them.
- The other features of Service health include cloud alerts that can notify us of any active issues or upcoming maintenance configured by us.
- Once we subscribe to an issue, we will get details and updates and we will get incident RCA.
- With Service Health, we get guidance and support during service incidents.

Here are some details:

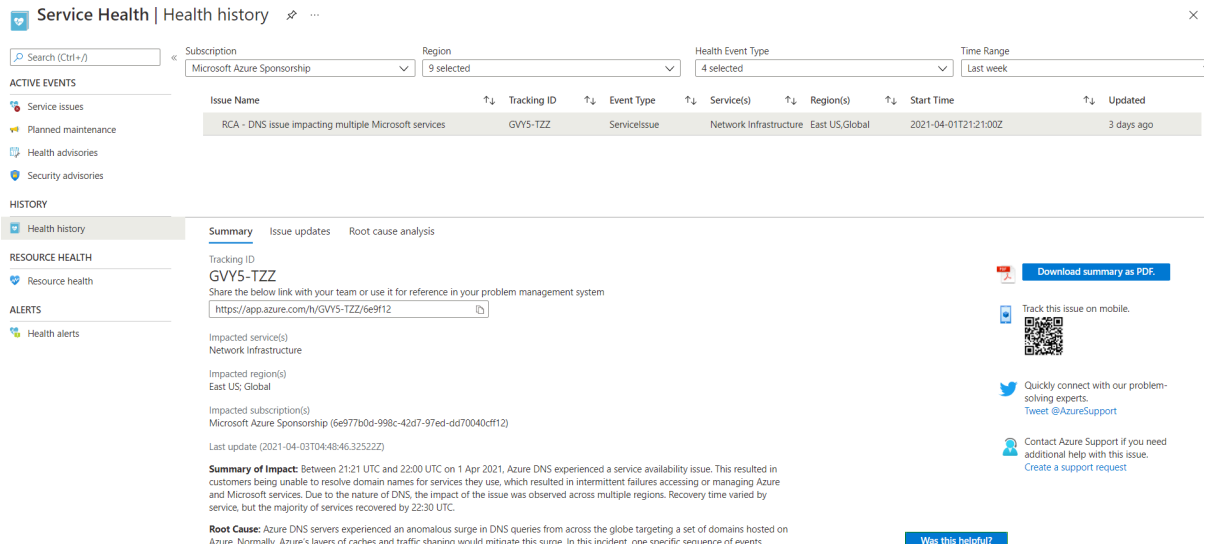
Service Issues

- This panel shows us any current issues that are on-going for the **regions/resources** where our resources exist.
- You can see that 3 subscriptions are selected with 9 regions and 184 services.
- You can also see the past incident at the bottom that has been resolved.
- We can get complete details and also download the RCA (Root Cause Analysis) for the issue.



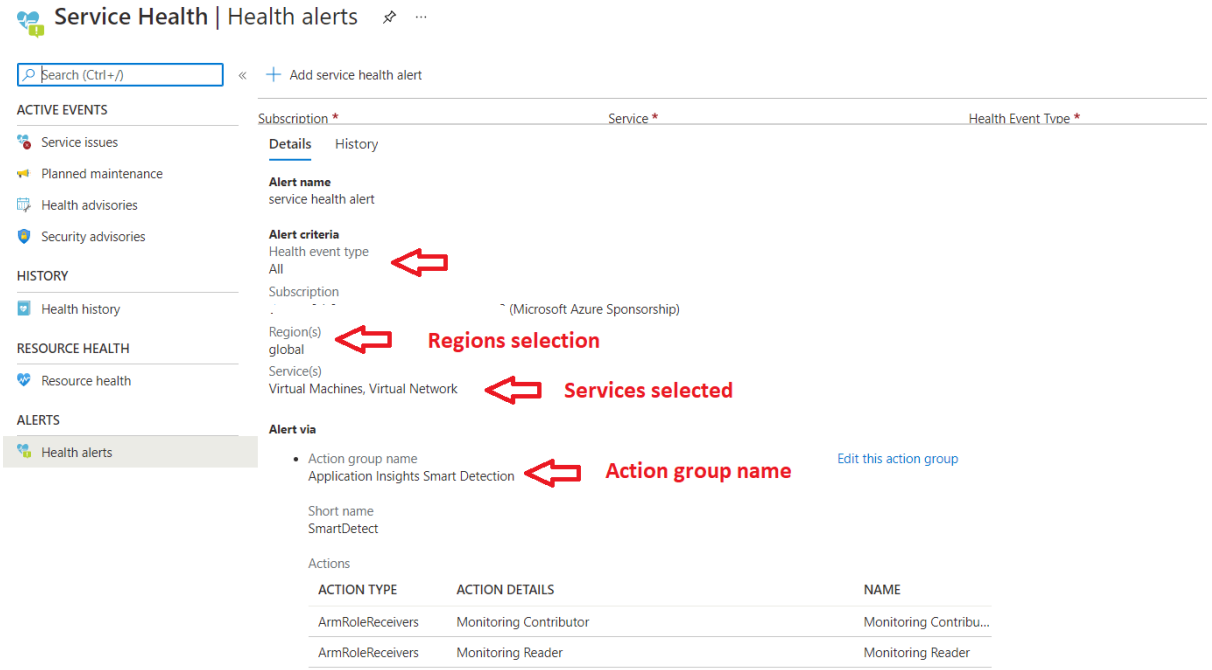
Health History

- We can see the health history and we can get details like Summary/ Issue updates/RCA.



Health Alerts

- We can set health alerts to be notified for the services we choose and for the regions which are of interest to us.
- Here we have selected to be alerted via the Action group when there are issues with VMs and VNets for all regions.
- Once set up, we will get an email when any issue occurs. We could also select the type of event. In this case, we have selected all events.



FAQs

1. **What are the permissions needed to view Service Health?**
   To view Service Health events, users must have the reader role on a subscription.

2. **How does Azure Service Health compare with Azure Status page?**
   - We use Azure status page for a global view of the health of all Azure services.
   - It serves as a quick reference for incidents with widespread impact. You can access this page at https://status.azure.com.
   - Service Health keeps us informed of the health of our environment with a personalized view of the status of our Azure services.
   - It provides us richer features including alerting and RCAs.

3. **What is the difference between Resource Health and Service Health?**
   Service Health provides information about the health of individual cloud resources, such as VMs etc. Service Health provides a personalized view of the status of our Azure services and regions

4. **If a service is down, should we contact Microsoft?**
   We need to check Service Health first to see if there is a known incident affecting us. If there are any outages reported also, we need to monitor for updates. If there is no issue listed, we need to create a support ticket.

5. **What is the cost for Azure Service Health?**
   Service Health is available at no additional cost.

6. **What are the SLAs for Azure service health?**
   Since Service Health is a free service, it does not have an SLA.

Azure Firewall

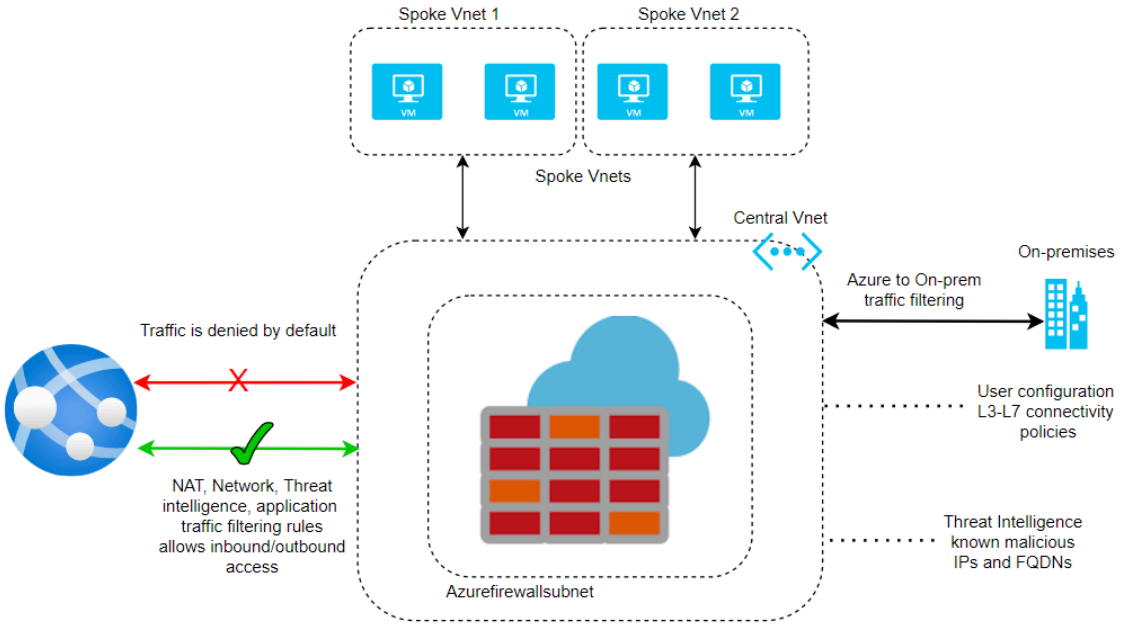
What is a Firewall?

A Firewall is a security device for the network that monitors both incoming and outgoing traffic. Based on a set of security rules, it will either allow or deny the traffic. It acts as a barrier between our network and traffic from external sources like the internet. The objective is to block malicious traffic which include hackers and viruses.

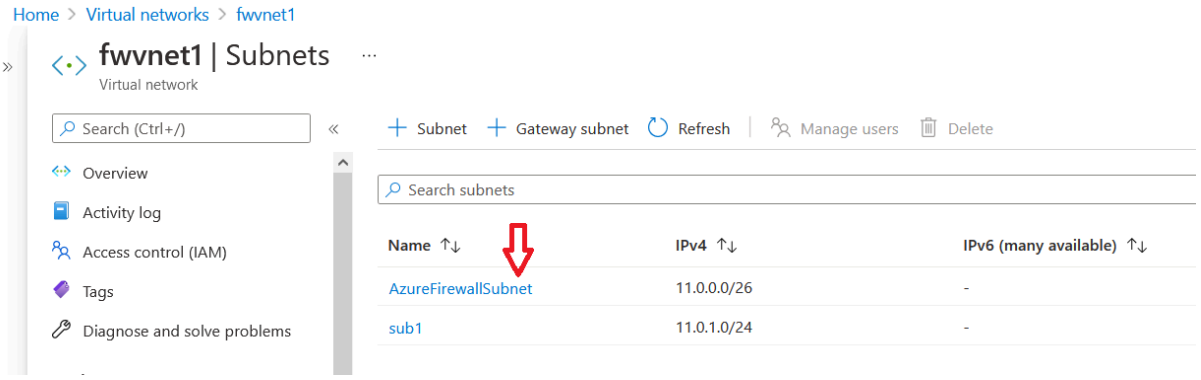
Azure Firewall

- Azure Firewall is a **network virtual appliance (NVA)** which is a managed network security device on the cloud.
- The function is to protect our network resources on the cloud. There are two types of firewalls and they are classified as either Stateful or Stateless. Let's say that you allow a certain incoming traffic *(say port 80).*
- When the same traffic returns, it is automatically allowed if it is stateless. On the other hand, Stateful traffic will need a specific rule for the outgoing traffic also, else the traffic will be blocked.
- Azure Firewall is a fully stateful firewall. So, we need to allow both incoming as well as outgoing traffic.

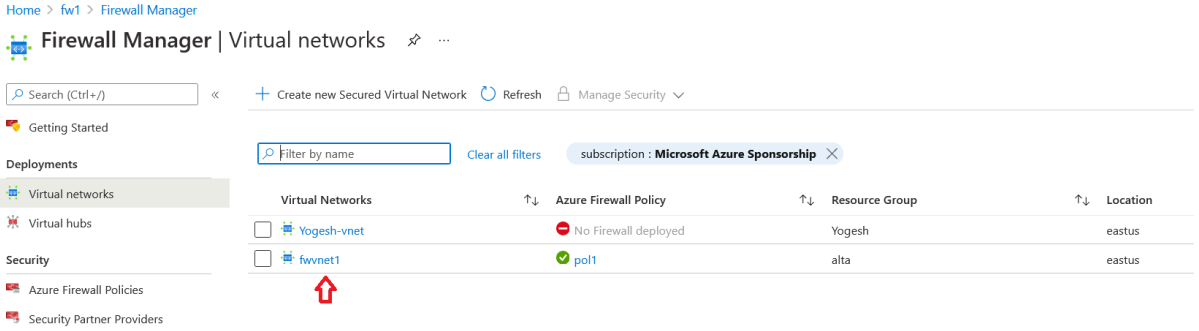
- Azure Firewall has built-in high availability and is highly scalable. We can create, enforce, and log application and network connectivity policies across subscriptions and virtual networks from a central location called **Firewall Manager.**
- We need to set up a static public IP address for the virtual network resources allowing outside firewalls to identify traffic originating from the virtual network. It is fully integrated with **Azure Monitor** for logging and analytics.
- A typical setup for the firewall is done via a hub and spoke model where the Vnet which hosts the firewall will act as a hub and the other Vnets will act as a spoke.
- The On premises and Internet is also connected to **Azure Firewall.** In this way, all traffic will enter via the firewall and the rules setup via the policies will then allow or deny the traffic.
- Please note the subnet that hosts the firewall must be named as Azurefirewallsubnet else it will not function



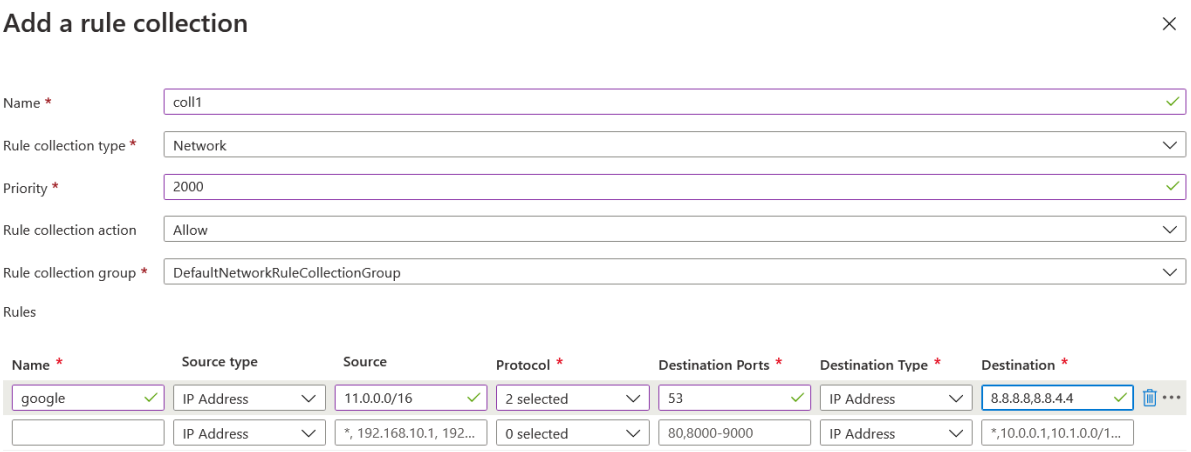
- Please see below the subnet created for the Azure firewall named as **AzureFirewallSubnet.**



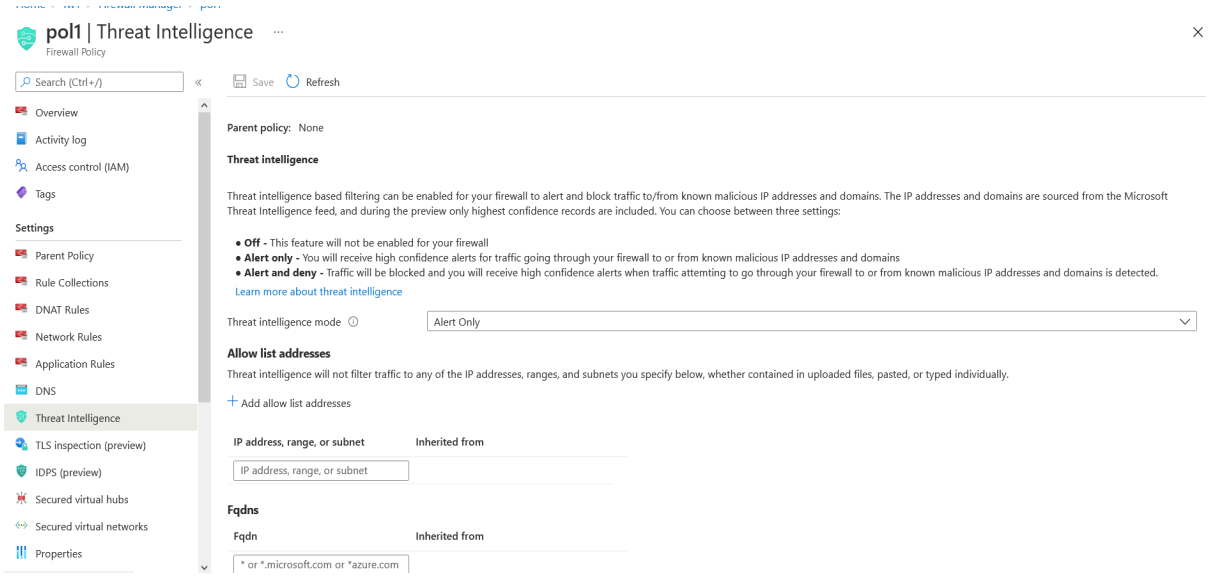
- As discussed, the rules are set up in a central location using the Firewall Manager. You can see the pol1 being assigned to **fwvnet1 Virtual Networks**. We can assign the same policy to other networks and it is easier to manage centrally.



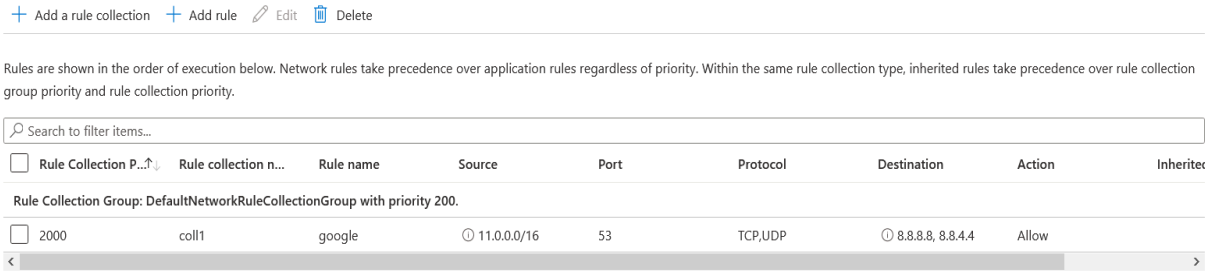
- A Policy consists of rule collections which in turn contains individual rules. Here we specify if the rule is to allow or deny.
- We assign a priority from **0 to 65535** and the lowest number takes the priority while processing the rules.
- We could place the rule collection within a group called the rule collection group. Also, the rule is available as a tab called Network rules on the main panel.
- We specify the source type as either an IP address or IP Group. We can give a range of IP addresses for Source and Destination. We can give * to indicate all.
- We can specify Protocol and Port numbers. In the example below, we have given Google a DNS server with IP of **8.8.8.8** and port of **53** which will allow DNS resolution.



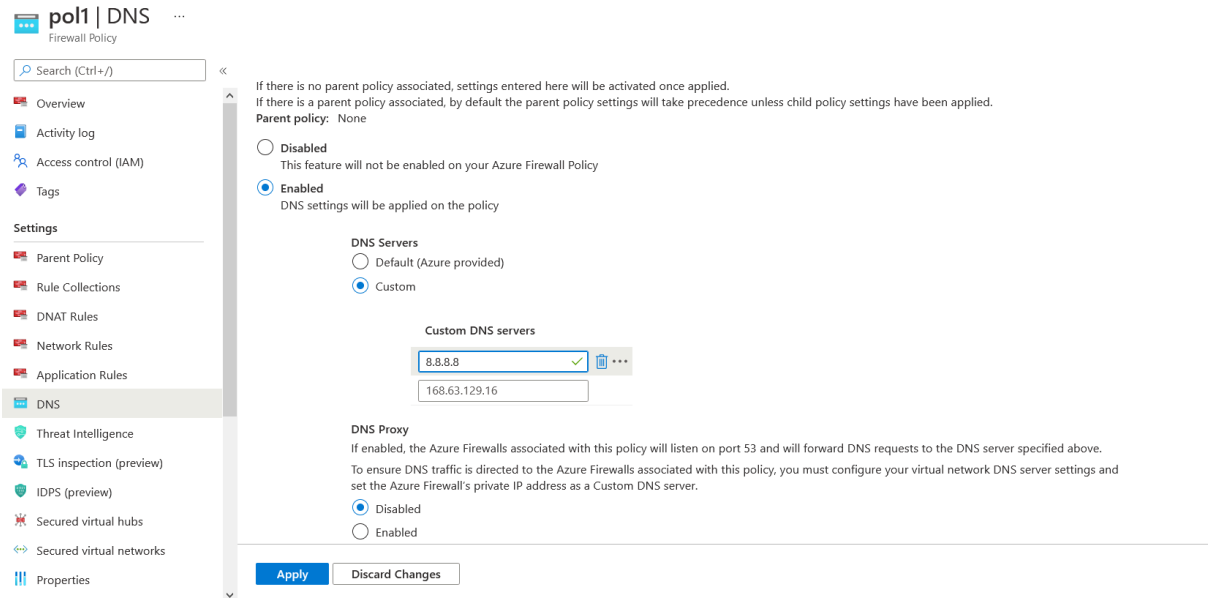
- We can optionally enable intelligence-based filtering called Threat Intelligence and the mode can be set to OFF/Alert only or Alert and deny. Microsoft threat intelligence feed provides a list of IP addresses and domains and these recorded are included as rules to allow or deny



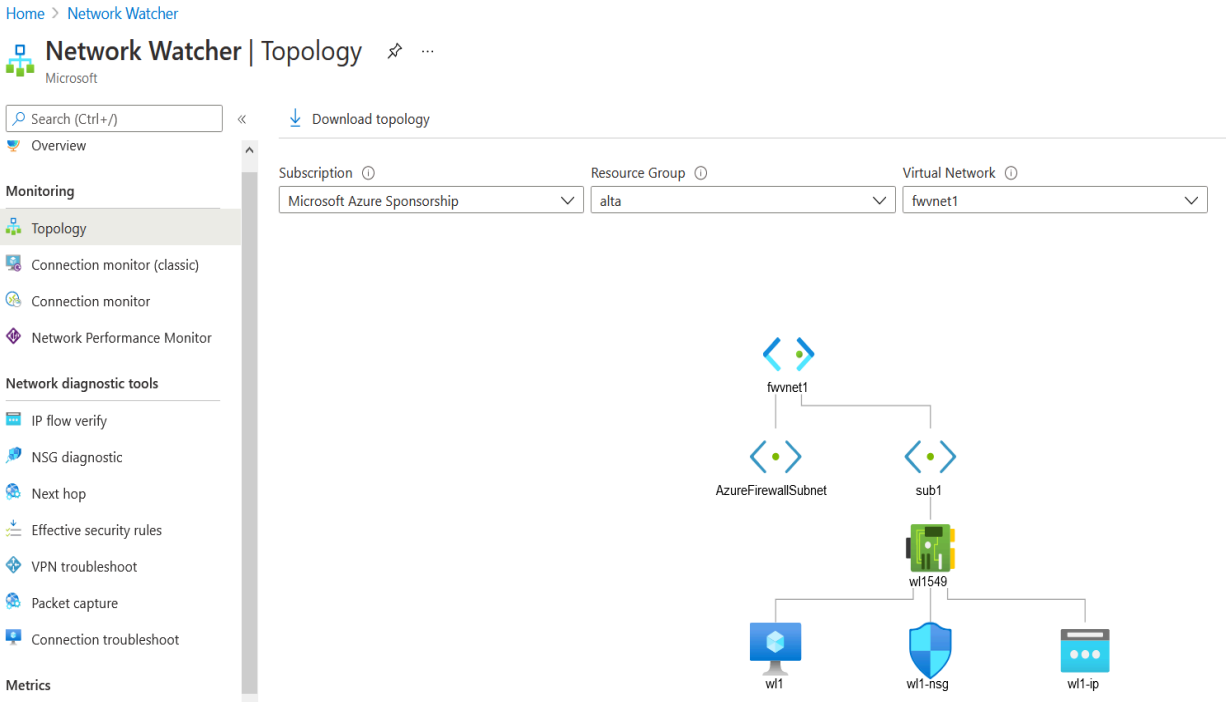
This is the Network Rule tab which lists the rules.



We can also set up DNS servers for DNS resolution on the DNS tab.



Finally, we can see the topology of the Vnet and the firewall subnet on the Network watcher blade under the Topology tab.



Azure Traffic Manager

Azure provides the following services for Delivery.

- *CDN*
- *Front Door*
- *Traffic Manager*
- *Application Gateway*
- *Load Balancer*

While Load Balancers and Application Gateways operate at **Layer 4 and 7,** Traffic Manager operates at a DNS level.

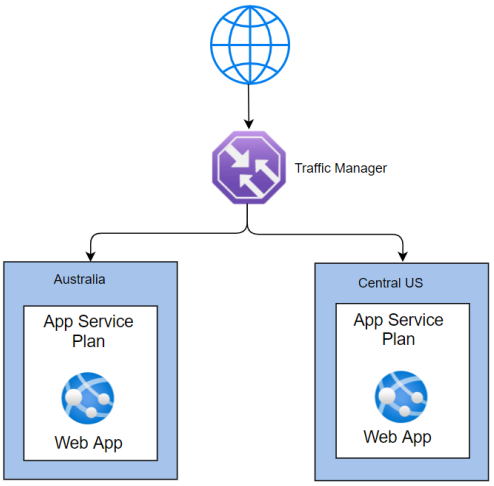
This service will distribute traffic to *public-facing azure services at a global level.* The public endpoints provided are having high availability and quick response.

We can use Traffic Manager to route traffic to regional application gateways at a global level, which could have a load balancer setup for multiple VMs at a database tier utilizing all the services.

Here are some more scenarios:
Application Gateway - to load balance between your servers in a region at the application layer.

Front Door - optimize the global routing of your web traffic and optimize top-tier end-user performance and reliability through quick global failover.

Load Balancer - Network Layer Load Balancing



How does a Traffic Manager work?

The Traffic Manager uses DNS for resolution. It uses this to find the name server. Then it locates the endpoints (which are not disabled) and routes the traffic based on the routing methods specified.

Routing Methods:

Here are the routing methods which we can configure:

Routing Method	Scenario
Priority	*When we have several endpoints, and we want to use one location preferentially, we can use this method having a primary service endpoint for all traffic.* *We can configure one or several multiple backup endpoints in case the primary is unavailable.*
Weighted	*When we want to split and route traffic to different locations, we should use this method. We have to set weights to accomplish this.* *Let's say we want to route traffic equally, we set the weight of 1 and 1 to both the endpoints* *If we give weights of 1 and 2, then the ratio will be 33:66 and one-third of traffic will go to the first endpoint, and two-thirds of traffic will go to the second endpoint.*
Performance	*Let's say that we have 3 locations like Las Vegas, Houston, and Jersey City on 3 sides of the country. We would like end-users to use the "closest" endpoint for the lowest network latency.* *For example, users in New York should connect to Jersey City, which is the closest location. Then we should select this routing method for the lowest latency by choosing the closest endpoint.*
Geographic	*Let's say that there is a requirement that data from a country (Saudi Arabia) has a mandate that data should not cross borders with sovereignty laws.* *We can use this method to direct users to specific endpoints based on where their DNS queries originate from geographically.* *So if a user from this country tried to access it, he would be routed to the servers in his country only.*

Multivalue	*If there multiple servers and we wanted to select multiple servers to select any of the available servers, we can select MultiValue.* *When a query is received for this profile, all healthy endpoints are returned. We can limit the servers returned by setting a max value.*
Subnet	*Use this method to map sets of end-user IP address ranges to a specific endpoint. When a request is received, the endpoint returned will be mapped for that request's source IP address.*

FAQs

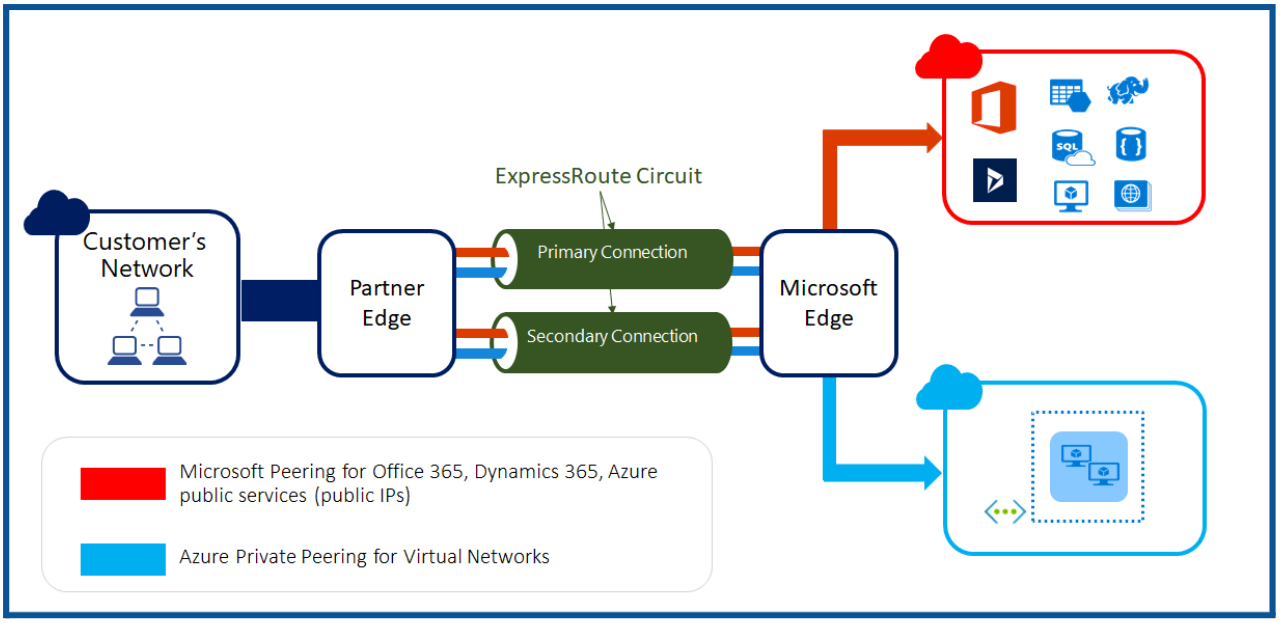
1. **What is the name of the website that will be created when we configure Traffic Manager?**
   Azure will always use azurewebsites.net as a suffix. We cannot change it

2. **So how do we use our website like whizlabs.com?**
   You need to create an alias in your DNS zone and point to the Traffic Manager.

Azure Express Route

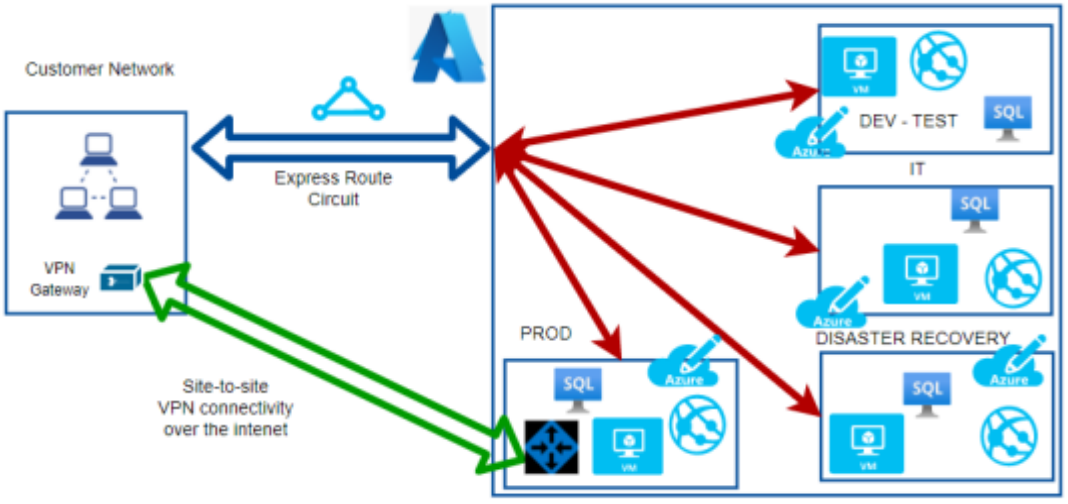
Connectivity to Azure

- There are several ways to connect to Azure. Broadly classifying them, we could either use the internet or have a direct connection.
- While connecting via the internet, we need to use a VPN to connect our infrastructure on premises with the cloud using a **VPN** gateway which encrypts our traffic by creating a tunnel.
- We could choose either a client-to-site VPN which is only one client system connected to the **cloud or site-to-site** VPN where we connect two sites.
- This setup depends on the public internet and we must secure and could have reliability issues.
- Hence it is better to use a dedicated connection between our infrastructure and the cloud
  with an **Express Route connectivity.**
- We need to locate a connectivity provider. There are several choices available based on location.
- For example, in India, we have *BSNL/AIRTEL/SIFY and in the USA, we have AT&T/ SPRINT/VERIZON and many more.*



(Source: Microsoft Documentation)

- Express Route connectivity allows us to connect to 2 Microsoft cloud services – **Microsoft Azure Services as well as Microsoft 365 services.**
- Also, we can see from the above diagram that there is an active-active redundant pair of cross connections setup for high availability. We can add further redundancy by adding up to 16 Express route connections.
- Express route has the following bandwidths to choose from based on our requirements:
- **50 Mbps, 100 Mbps, 200 Mbps, 500 Mbps, 1 Gbps, 2 Gbps, 5 Gbps, 10 Gbps**
- If we have multiple subscriptions, we can connect all of them to a single Express route connection.
- You can have upto 10 Vnet connections on a standard Express route connection and upto 100 for Premium connection. However please note that all connections will share the same bandwidth.



We could even have a site-to-site VPN for adding redundancy. If there were issues with the Express route, we can failover to the S-2-S VPN.

FAQs

1) **If I have a 100 Mbps circuit, what is ingress and egress capacity?**
   You will have an incoming capacity of 100 Mbps and outgoing capacity of 100 Mbps.
   What is the routing protocol?
   Express route uses BGP (Border gateway protocol)
2) **What happens if there is any maintenance?**
   There won't be any impact. Express route uses an active-active setup and only the circuit will be maintained at a given time.
3) **So where does the connection land on the Azure cloud?**
   We connect to one of the Vnets in a subscription. We can connect upto 10 Vnets in each of the 10 subscriptions max. We need to go for Premium if we would like to add more.
4) **How do we plan for Disaster recovery?**
   Microsoft recommends 2 Express connectivity to avoid a single point of failure. We could also set up a Site-to-site VPN instead of a second circuit.

Azure VPN Gateway

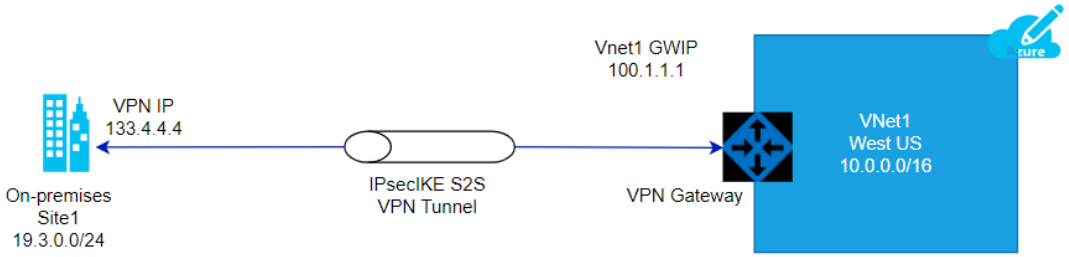
This is one of the methods that allows inter-site connectivity. Express route is the preferred connection as it has higher bandwidth and is in active-active mode. Smaller enterprises can choose VPN gateways or one could use VPN Gateway as a backup to Express route connectivity.

The VPN gateways are set up over the Public internet. Hence the traffic needs to be encrypted. **IPSEC** is used as the tunnelling protocol which creates a secure tunnel through which the data travels. Even if the traffic is intercepted, it cannot be decrypted.

VPN Gateways types

- **Site-to-Site VPN Gateways**
  - Here the On-premises will be a site and Azure VPN will be another site. We can connect multiple VNets.
- **Point to Site VPN gateways**
  - Here we connect a single client machine from on-premises to the Azure VNet.
  - We can use the same connection on multiple clients by exporting the configuration from the existing client.
- **Internal Gateway between Azure networks**
  - This is a special use case where we want to encrypt traffic between Azure Vnets.

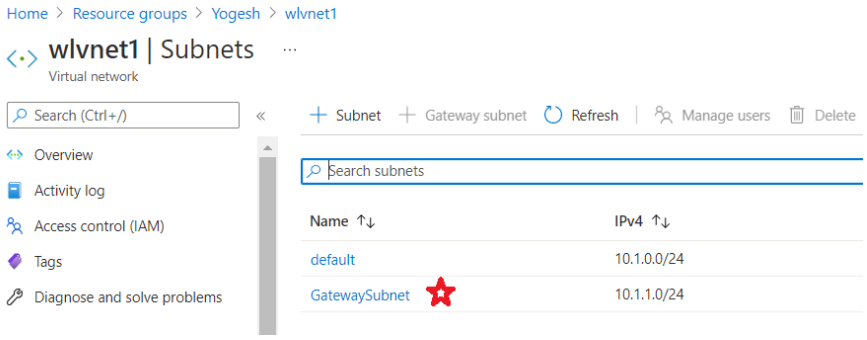
VPN Gateway Architecture



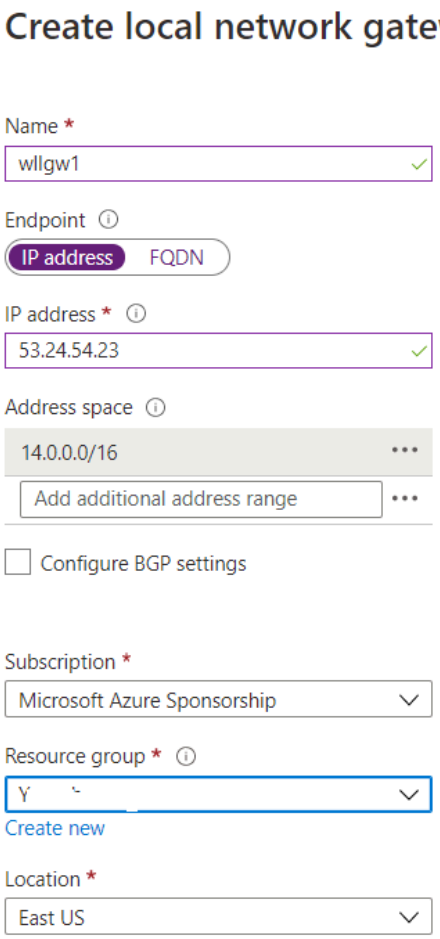
Steps to establish VPN Gateway

1. **GatewaySubnet**
    a. We need to create a subnet with the name "**gatewaysubnet**" for the setup
    b. If we are creating a Vnet, this subnet gets created automatically.

Please see the diagram below which shows the gateway subnet. This was created implicitly when the wlvnet1 was created as part of the Vnet gateway creation.

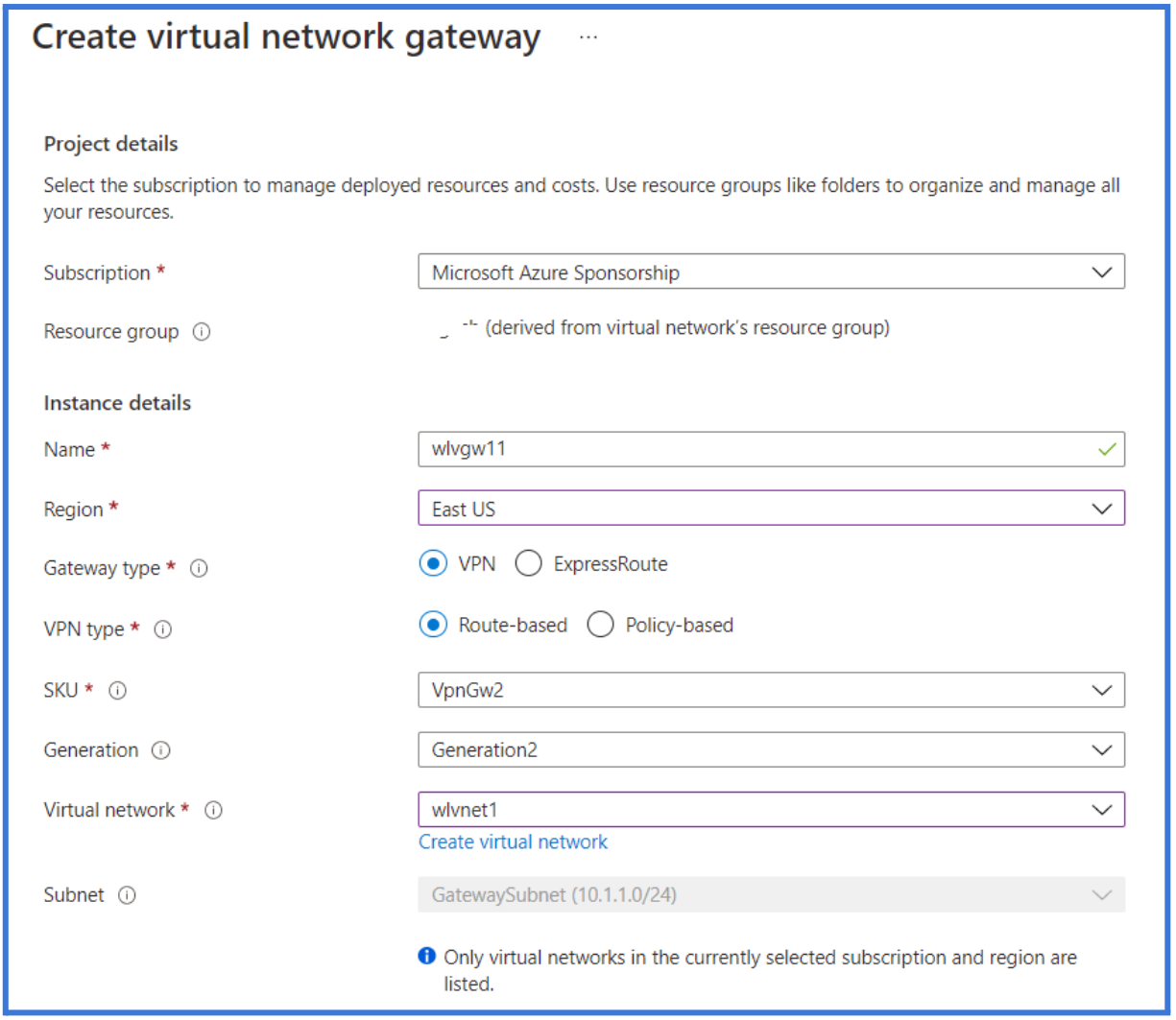


2. **Local Network Gateway**
    a. We need to obtain a Public IP address from the on-premises admin team and use that as the endpoint.
    b. See the IP address given as **53.24.54.23**. This is the ip address of the router on-premises



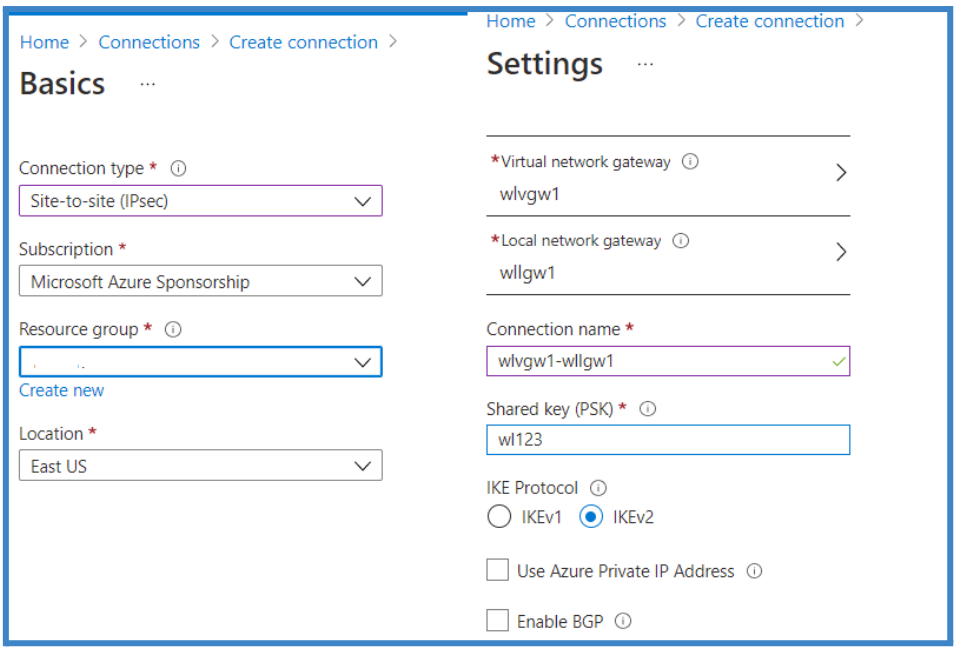
(Source: Microsoft Documentation)

3. **Virtual Network Gateway**
    a. We need to create the virtual network gateway with the following inputs
        i. **Gateway type** – in our case, we are going to use VPN
        ii. **Vpn type** – could be either Route-based or Policy-based. Please note that we cannot change the type once it is created.
        We need to delete the gateway and recreate it to make the change. Policy based is the most common type
        iii. **SKU** – There are several SKUs. Please note that Basic is considered legacy and not recommended.
        iv. **Subnet** – As mentioned, the name should be GatewaySubnet.



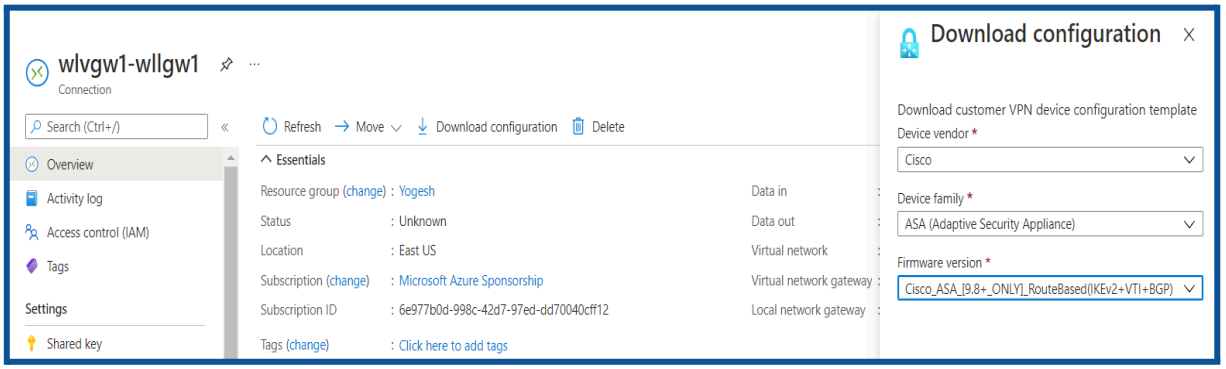
(Source: Microsoft Documentation)

4. **Connection**
    a. Once the prerequisites are fulfilled with the creation of Gateway Subnet, Local Network Gateway and the Virtual Network Gateway, we can create connection as follows
        i. **Connection type** – the options are Vnet-to-Vnet, Express Route or Site-to-Site. In our case, we choose Site-to-site which uses the IPsec tunnelling protocol by default.
        ii. **Bidirectional Connectivity** – Connections are usually unidirectional. We can select bidirectional to choose 2-way communication
        iii. **Shared Key(PSK)** - We need to create a password here and need to share this with the on-premises admin to configure from their side.



(Source: Microsoft Documentation)

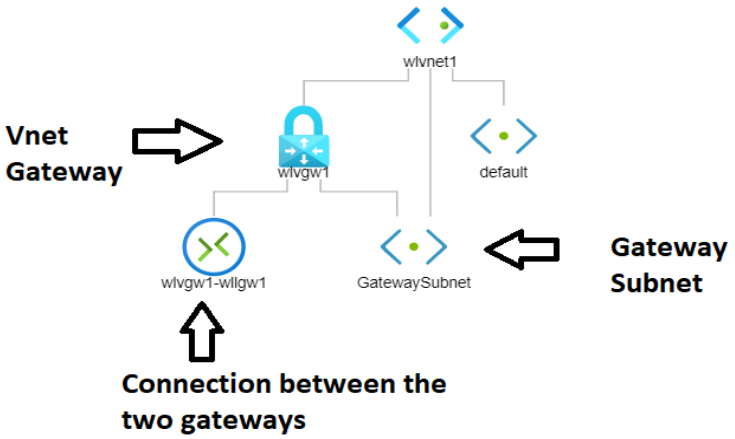
5. **On-premises setup**
   a. Once the setup is complete, we can download the configuration to be shared to the on-premises admin.
   b. We need to get the router model and select the same from the dropdown list and download the configuration and share with the admin along with the shared key.



(Source: Microsoft Documentation)

Topology

We can check for the topology from the network watcher – topology blade.



Azure Content Delivery Network(Azure CDN)

What is a CDN?

CDN stands for **Content Delivery network**. It is an architecture of distributed network of servers that can efficiently deliver web content to users.

CDNs will cache the content on edge servers in the POP (point of presence) locations keeping the content closer to the users thereby minimizing latency. This is made possible by using the existing network infrastructure of the CDN provider.

Let's say that a company Whizlabs has a headquarters in NY, USA and branches in CA, USA and Bangalore, India.

The Servers are located in NY and we have a user logging in from Bangalore, India. The data needs to traverse the network and this will cause latency.

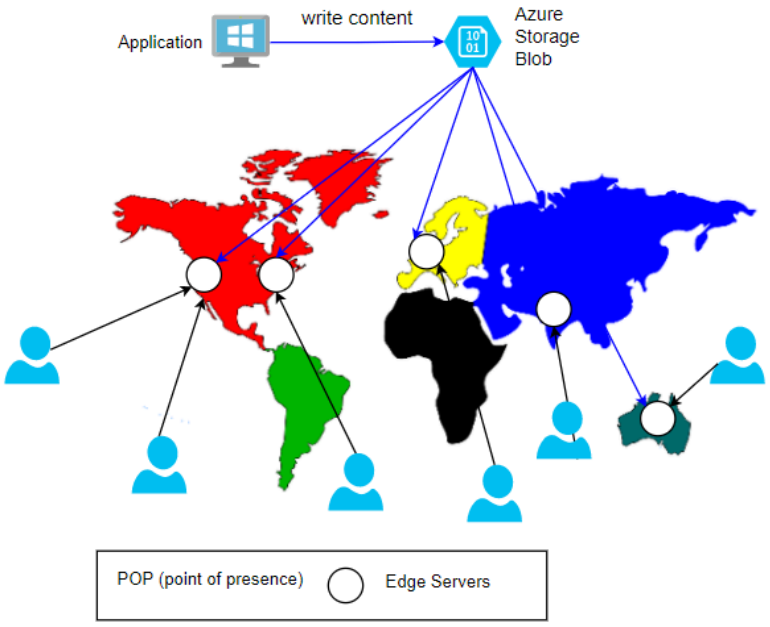
The solution here is to use CDN and use the Bangalore location to cache the data. Now, the user will be able to retrieve the data from Bangalore.

Please note that the data is not stored permanently on the edge location. This is to ensure that data does not go stale and it is current.

So we may set a cache interval of 24 hours and every day, the data will be retrieved from the Origin Server (NY, USA) and cached on the edge locations.

Also, if the data is not available (first time accessing) or if the cache has been marked as invalid, the data will be fetched from Server and sent to the user and cached on the edge location.

The next time, the request will be fulfilled by the edge server. This also reduces the load on the Origin server.



FAQs:

1) **How long is the data cached on the edge Server?**
   The TTL (time to live) by default is 7 days. This can be configured as per the application requirements. Once TTL expires, the cache will be marked as invalid.

2) **What type of azure servers can serve as Origin Servers to get source data?**
   Azure Web App, Azure Cloud Service, Azure Storage account, or any public web server.

3) **What are the CDN products available?**
   Azure has its own product. Besides that, it has tied up with Akamai and Verizon. Here are the offerings:

   a. **Azure CDN Standard from Microsoft**
   b. **Azure CDN Standard from Akamai**
   c. **Azure CDN Standard from Verizon**
   d. **Azure CDN Premium from Verizon.**

Please note that not all products might be available at all locations. You will need to check the product availability for your location.

4) **What are some of the additional features?**
   a. **Dynamic site acceleration(DSA)**
   b. **Video streaming optimization**
   c. **Customizable, rules based content delivery engine**
   d. **HTTPS support with CDN endpoint**
   e. **Compression encodings**

5) **Who are the market leaders for CDN? - PFB**

Top Competitors	Market Share	# Websites
jQuery CDN	38.60%	19,13,841.00
CloudFront	24.57%	12,18,186.00
BootstrapCDN	8.88%	4,40,178.00
Amazon S	37.79%	3,86,324.00
Vimeo CDN	5.71%	2,82,933.00
CDN JS	4.16%	2,06,402.00
OSS CDN	3.59%	1,78,093.00
CloudFlare	2.56%	1,27,104.00
Microsoft Ajax CDN	1.97%	97,471.00
Akamai	1.67%	82,949.00
MaxCDN	0.49%	24,381.00

Azure DDoS Protection

What are DoS and DDoS?

DoS stands for **denial of service** and **DDoS** stands for **distributed denial of service**.

Scenario:

*Let's say that you have a web server serving web traffic and you are a medium enterprise handling **1000** requests per second. If any malicious entity sends **100,000** requests per second, your server will be busy trying to respond to the 100K requests and unable to serve the regular customers. This is called **Flooding**.*

*Often the load will be so heavy that it will cause the server/machine to crash. This is called **denial of service** where customers are denied service by rendering the server unusable.*

*Imagine the same **100K** requests coming from multiple servers where malicious entities do a coordinated attack with multiple servers. This is called **distributed denial of service** where multiple servers hit a given target to bring it down. We have seen attacks feeding as much as **800 Gbps** which can bring the biggest servers down.*

Azure DDoS

It provides protection against DoS attacks with always-on monitoring and automatic network mitigation.

There are two levels of Service – One is **BASIC** and the other is **STANDARD**. Basic plan is free and enabled by default. After all, Azure needs to protects its resources 😊

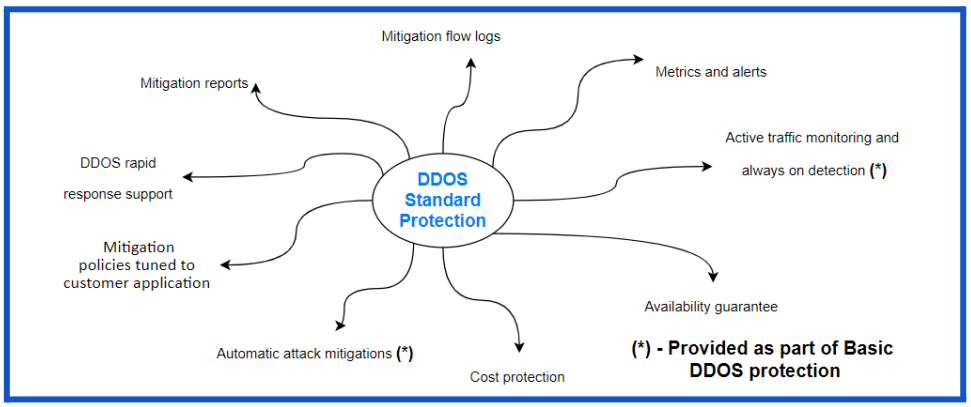
Basic plan, like the name says, provides only basic services *(always-on monitoring and automatic network mitigation)*.

Standard protection provides multiple features. This could cost you as much as **3000$** to protect about 100 resources like

- *Azure firewall, App Gateway/WAF*
- *VMs, AKS*

- *SQL, CosmosDB, Storage, App Services, etc.*
- *Vnet*
- So, let's say that a **DoS/DDoS** attack occurs. The Cloud is resilient usually due to the elasticity and if you have good autoscaling, then the cloud resources will keep scaling up like VM spinning up, App Service scaling up, etc.
- As a result, you will have a lot of traffic and you must be aware that while ingress traffic is not charged, consumers pay for egress traffic.
- So you will land with a huge compute bill and egress data charges.
- If we had the DDoS standard protection plan, we would be issued credit for the excessive charges if the plan failed to protect us.

Here's the list of services provided.



Some of the features of DDoS Standard protection are:

- **DDoS Rapid response** – We can engage the **DDRT** (DDoS Rapid Response Team) for attack investigation and analysis
- **Cost Guarantee** – As discussed, we will be issued a service credit for the application scale out and excess data transfer
- **Attack alerting/Metrics** - Alerts can be configured to be notified at the start/stop and logging will be done and metrics provided.
- **Extensive Mitigation Scale** – This works at a global scale and is highly scalable and can mitigate over 60 types of attacks.
- **Multi-layered protection** – It can protect at different layers (layer 3/4/7)
- **Adaptive tuning** –Let's say there is unusual traffic from an IP and it is determined as anomalous by the DDoS cognitive services, ddos protection will automatically deny traffic from the IP and block it.

Azure Compute Gallery

Azure Compute Gallery helps you create & build structure and organization around the Azure resources(such as images and applications) and It provides the following

- Global replication and Versioning and grouping of resources for easier management.
- Highly available resources with Zone Redundant Storage (ZRS) accounts in regions that support Availability Zones. ZRS offers better resilience against zonal failures.
- Premium storage support (Premium_LRS).
- Sharing to the community, across subscriptions, and between Active Directory tenants.
- Scaling your deployments with resource replicas in each region.

With Gallery, we can share our resources with everyone or limit sharing to different users, service principals, or AD groups in your organization. Resources can be replicated to multiple regions for rapid scaling of your deployments.

This (Azure Compute Gallery) service is not a global resource. For disaster recovery scenarios, It is best practice to have at least two galleries in different areas.

FYI: You can use the below links for more knowledge
Overview of Azure Compute Gallery - Azure Virtual Machines | Microsoft Learn
Store and share images in an Azure Compute Gallery.

Configure Azure Application Gateway

One of the main benefits of the Cloud is elasticity on-demand.

In a traditional datacenter, if there is a peak load requirement of 100 cores from 10-11 am when users login, the machines will always need to have the capacity of 100 cores.

However, in the cloud environment, we will have a single VM with 50 cores at all times and add another VM with 50 cores between 10-11 AM alone. This has reduced consumption by almost 50%.

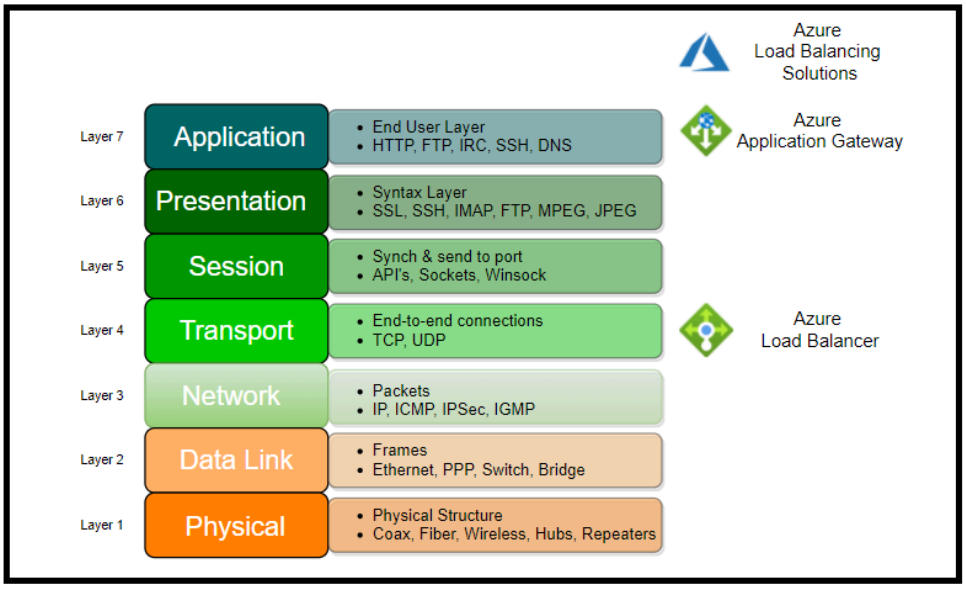
But how do we now distribute the load between the two VMs?

The solution is Load Balancing.

Load balancing can be done at 2 layers in the OSI model. One is at Layer 4 where we will use the Azure load balancer. Here a combination of source and target ip and TCP/UDP Protocol will be used to achieve routing.

The other routing type is at Layer 7, which is the Azure Application gateway. Here the application gateway uses a front-end IP address which is resolved from FQDN via DNS. It has an optional WAF (Web application firewall).

OSI LAYER and the load balancing options within Azure

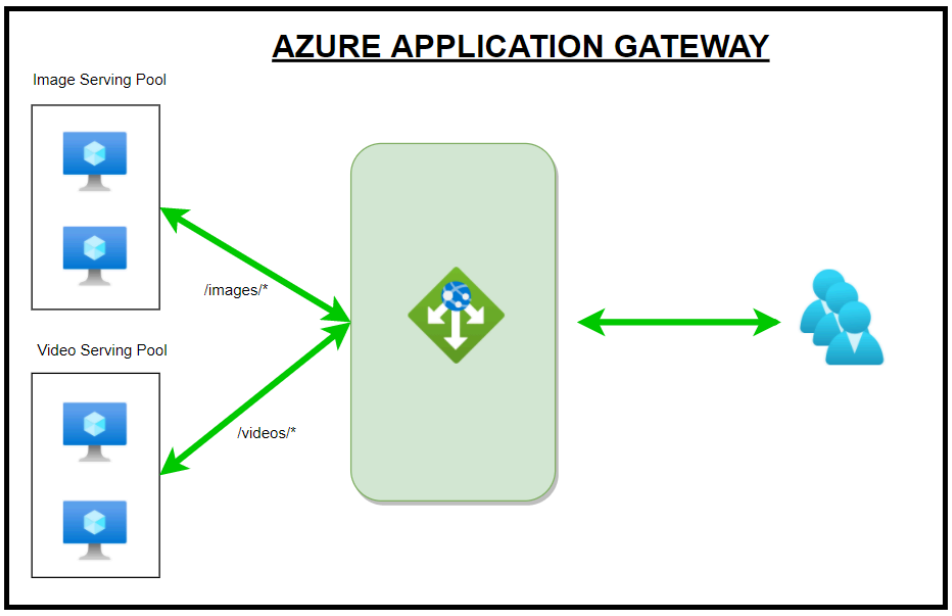


How does the Application gateway work?

Step 1: User sends a request to a website with **FQDN (fully qualified domain name)** – for example, https://whizlabs.com/videos. The query will be sent to a DNS server, and it will return the IP address.

Step 2: The application gateway will be configured with a listener, a logical entity checking for connection requests. The listener is configured with a front-end IP address, protocol, and port number for connection requests.

Step 3: The application gateway also has a backend Pool/s. The backend pools could be VMs or **VMSS (VM Scale Sets)** or external servers, or Azure App servers. Based on routing rules set up, the traffic will be routed to the appropriate backend servers.



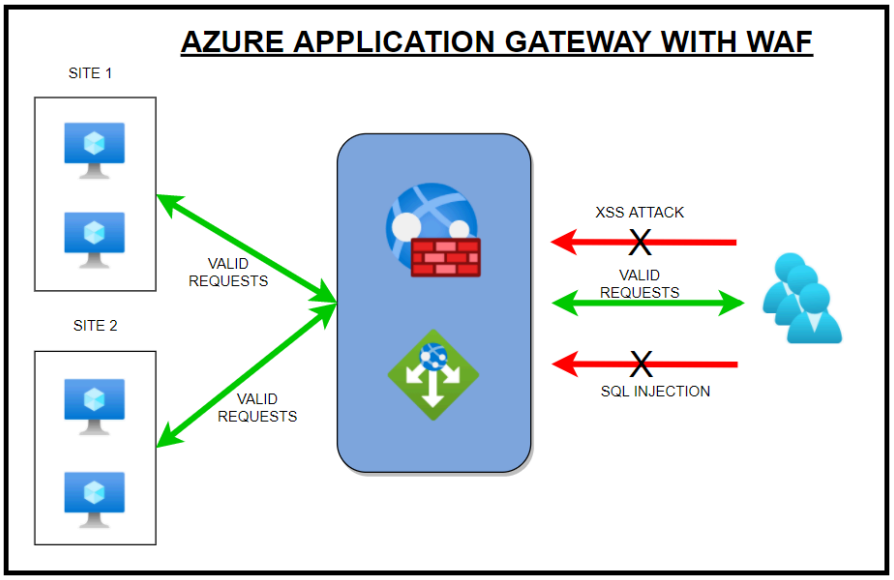
In the above example, you can see the routing rules being processed with url based routing. So when the users type the url https://whizlabs.com/videos, the gateway sees the videos in the url and sends the traffic to the Video Serving Pool.

Application Gateway with WAF

There is an optional feature WAF that can be additionally added to the application gateway. WAF is based on **Core Rule Set (CRS).**

We need to set up a WAF Policy that has rules. There are two types of rules. One is Managed rule sets which Azure preconfigures. The other is custom rules. Some of the features of WAF are

- Some of the features that WAF provides are preventing SQL injection/ **XSS/ http** protocol violations.
- It also protects against crawlers and scanners. We also can allow or block traffic coming in from certain countries/regions in preview, and it is called **Geo-filter traffic.**
- WAF can be set up in two modes which are Detection or Prevention.
- When WAF is added, the traffic will be evaluated before Step 3 above against the WAF rules.
- If violating traffic is found in Detection mode, the warning will be issued, and traffic continues to flow. In Prevention mode, the traffic will be blocked.



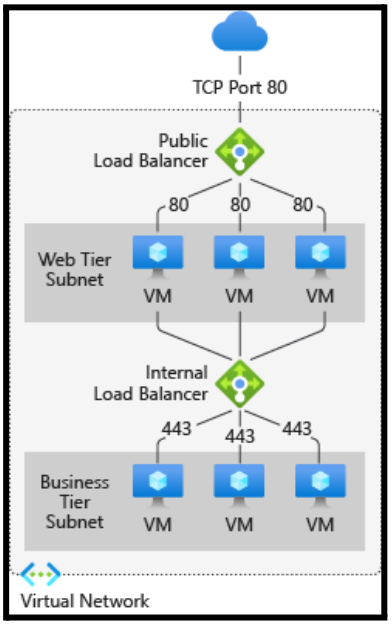
Virtual Machine Network Settings

If you are creating a virtual machine (VM), you can create a virtual network or use an existing one. You need to figure out how you want to access the virtual machines(VM) on the virtual network(VNet)

It's important to plan before creating resources and make sure you understand the limitations of networking resources.

In the below image, **VMs → are represented as web servers and application servers.**

Each set of virtual machines are assigned to separate subnets in the VNet.



You can create a virtual network before you create a virtual machine, or you can create a virtual network when you create a virtual machine(It's completely up to you)

You need to create the below resources to support communication with the virtual machine: Network interfaces, IP Addresses, and Virtual network & Subnets

Note: Additionally, you need to consider → Network Security Groups & Load balancers are optional resources.

- **Network interface card (NIC) is an interconnection between a VM and VNet.**
- **A virtual machine must have at least one NIC.**

You can assign the below types of IP addresses to a network interface in Azure:

➔ You can assign public IP addresses to: Virtual machines & Public load balancers

➔ You can assign private IP address to: Virtual machines & Internal load balancers

Introduction to Azure Kubernetes Service (AKS)

Kubernetes has become very popular, and many cloud service providers offer a Kubernetes based platform or infrastructure related *PaaS or IaaS offering.*
Google has GKE (Google Kubernetes Engine), AWS has EKS (Elastic Kubernetes Service), and Azure has AKS (Azure Kubernetes Service)

What is Containerization?

- In the traditional computing system, we had to install an Operating system and install all dependencies for an application to work. Only a single OS could be installed.
- Then came Virtualization where we could install multiple OS by introducing another layer between the hardware and the OS and this was called Virtualization. So only physical machines appeared as multiple systems.
- Then came a lightweight alternative to virtualization, which was called Containerization. This removed the drawback of having a full machine, and this had only the necessary components.
- Containers will encapsulate an application with its operating system. This would contain all the dependencies that were needed for an application to run. So we take the container and run it on any operating system, and it will run.
- Some of the containerization options are Docker, which is the most popular and sometimes equated to containers. But there are others like **LXC/LXD, ContainerD, Rocket**.

Orchestration

- Orchestration is the system that is used to manage the deployment of containers. We use Orchestrators as tools to achieve this. Some of the performed activities are automating the maintenance of those applications, replacing failed containers automatically, and managing the rollout of updates and reconfigurations of those containers during their lifecycle.
- The popular tools are *Docker Swarm by Docker, Nomad by Hashicorp, Flocker, & Kubernetes.*
- Kubernetes, also stylized as K8s, is an open-source container orchestration system. It is used for automating computer application deployment, scaling, and management. It was originally designed by Google and influenced by Google's Borg System and is now maintained by the Cloud Native Computing Foundation. It is a cluster management software for Docker containers mainly but supports others also.

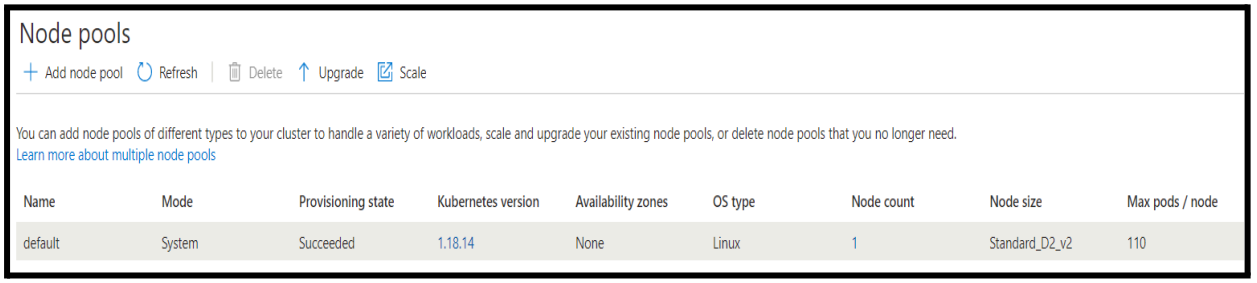
Components of AKS

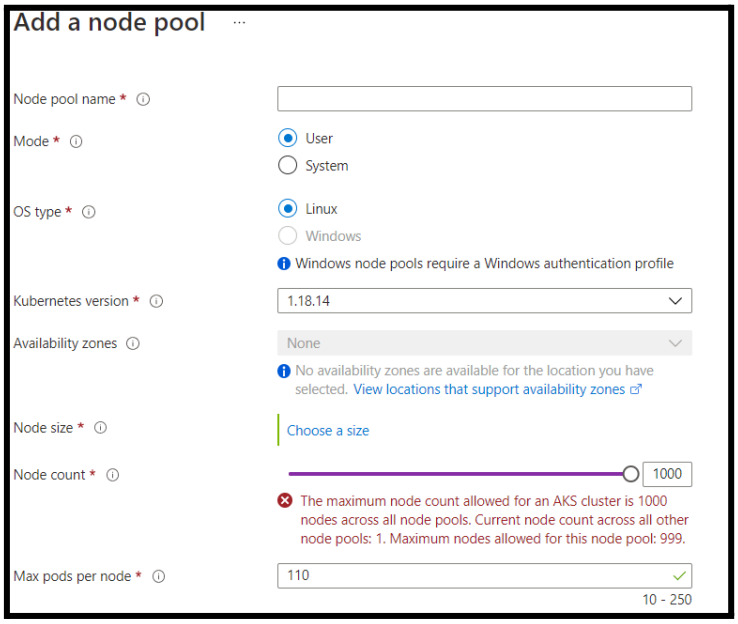
1. **The Cluster**
   - The Cluster contains 2 components
     - Control Plane – this consists of kube-apiserver, etcd, kube-scheduler and kube-controller-manager
     - Nodes that run the applications

2. **Persistent Volumes**
   - Since the nodes are added and removed on-demand and the storage associated with it is temporary, we need to create storage outside of the cluster. Hence we create persistent volumes.

3. **Node**

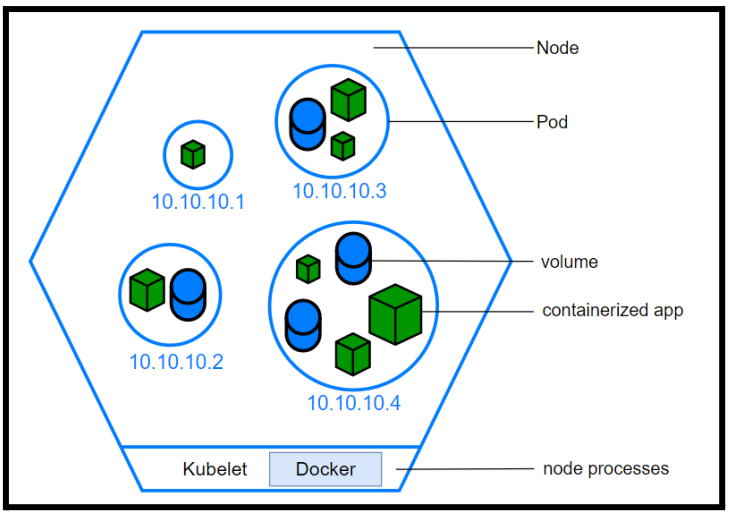
- We create Node pools in Kubernetes (as shown below). Here we choose a VM size, and that will be the unit size of the nodes within the pool.
- We can add node pools as needed. The first node pool created is the **system node** pool which hosts critical system pods like coreDNS and tunnel front.
- We then add user node pools for application support and create different pools based on the application requirements.
- Pods will be created within the nodes, and the max pod setting is configured at the node pool level.





4. Containers

- We store our code that is going to be run inside containers. There are readily available pre-built containers stored in container repositories or we can create our own containers.
- One or more programs can be run from the containers



5. Pods

- Nodes create Pods, and kubernetes use Pods to run instances. Usually, only one container is run within a pod, but multiple containers could run in a pod if there was a requirement from the application.
- We scale based on pods. When we can scale, we simply use pod replicas. A new pod will be spun up in another node, and we now have an additional pod. Same way, we can remove the pods to scale down.

6. Deployments

- We don't launch pods directly. Instead, we create deployments.
- A deployment will state how many replicas should run and the system manages that.

Sample Deployment yaml file

```
apiVersion : apps/v1
kind: Deployment
metadata:
  name: wl-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: wl-app
  template:
    metadata:
      labels:
        app: wl-app
    spec:
      containers:
        - name: wl-app
          image: wl662930
99.azurecr.io/wl-app
          ports:
          - containerPort
: 3000
```

7. Ingress

- By default, Kubernetes provides isolation between pods and the outside world. If you want to communicate with the service running in the pods, you need to open the communication. This is called Ingress.
- You can achieve this communication in several ways. The most common ways are Ingress controller or a load balancer. Please see the sample service.yaml file which creates an external load balancer. We get the IP of this service and connect.

```
apiVersion: v1
kind: Service
metadata:
    service.beta.kubern
etes.io
    name: wl-app
spec:
    type: LoadBalancer
    ports:
    - port: 3000
    selector:
        app: wl-app
```

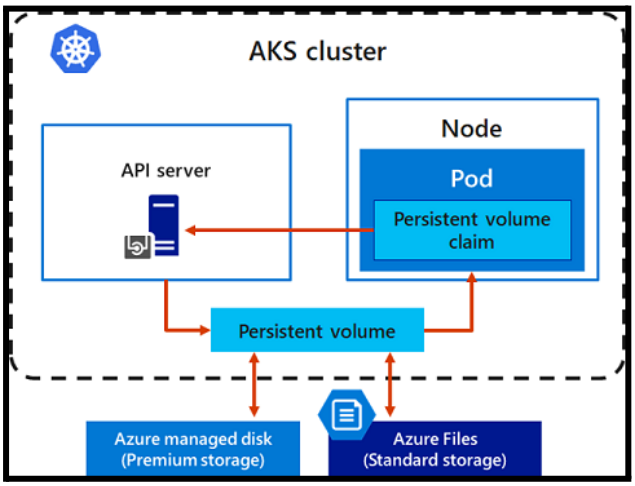
Azure Kubernetes Service Storage

Applications running on Azure Kubernetes Service (AKS) need to store and retrieve data. While some application workloads can use local, fast storage on redundant, spaced nodes, others require storage that resides in more common data volumes on the Azure platform.

Multiple pods may need to → Share the same data volumes and reattach data volumes if the pod is rescheduled on a different node.

Finally, you may need to collect & store sensitive data/ application configuration info in pods.

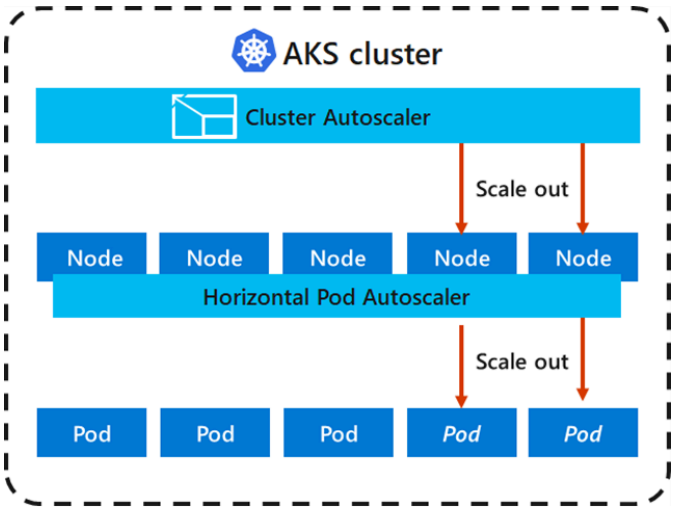
The below are the main concepts of providing storage for your applications in AKS:
Volumes, Persistent volumes, Storage classes, Persistent volume claims



Azure Kubernetes Service Scaling

The scaling process involves adjusting the compute resources allocated to your application instances to meet workload demands. When you run applications on an Azure Kubernetes Service cluster, you need to apply scaling to increase or decrease your compute resources.
If the number of your application instances changes, so will the number of underlying Kubernetes nodes. You may also need to quickly provision a large number of additional application instances.

The following illustration shows a scaling implementation for Azure Kubernetes Service.



Azure Kubernetes Service Networking

To allow access between your applications or application components, Kubernetes provides an abstraction layer for virtual networking. Kubernetes nodes connect to a virtual network, providing inbound and outbound connectivity for pods.

Things to know about Kubernetes virtual networking
Kubernetes nodes are connected to a virtual network that provides inbound and outbound connectivity for pods.
The kube-proxy component runs on each node to provide network features.
Network policies configure security and filtering of network traffic for pods.
Network traffic can be distributed by using a load balancer.
Complex routing of application traffic can be achieved with access controllers.

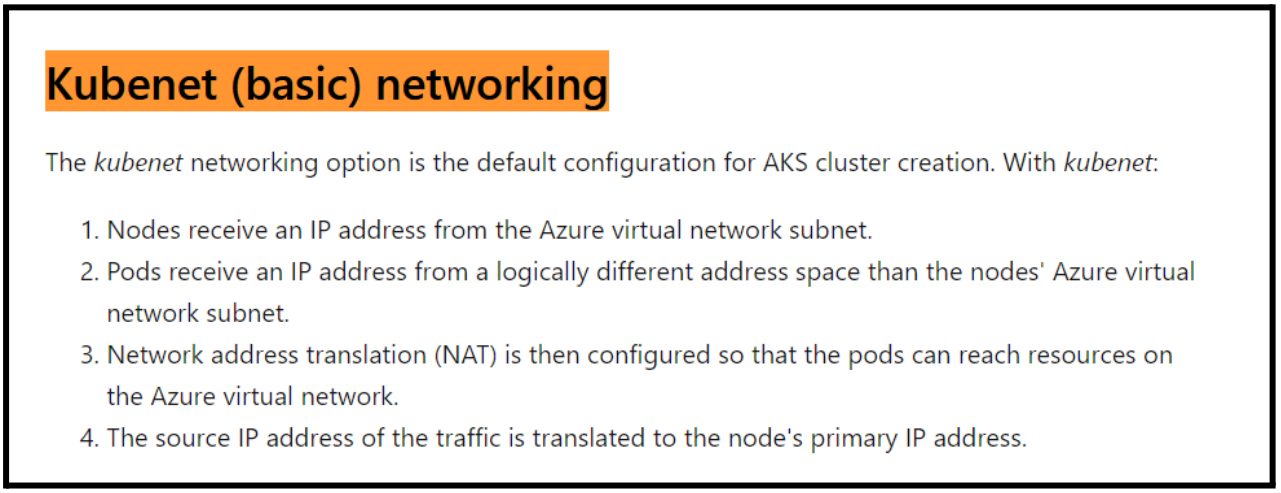
The Azure platform helps facilitate virtual networking for Azure Kubernetes service clusters.

Kubernetes Service Types

Service type	Description	Scenario
Cluster IP	Create an internal IP address for use within an Azure Kubernetes Service cluster.	Implement internal-only applications that support other workloads within the cluster
NodePort	Create a port mapping on the underlying node.	Allow direct access to the application with the node IP address and port
LoadBalancer	Create an Azure Load Balancer resource, configure an external IP address, and connect the requested pods to the load balancer back-end pool.	Allow customer traffic to reach the application by creating load-balancing rules on the desired ports
ExternalName	Create a specific DNS entry.	Support easier application acce

In AKS, You can deploy a cluster that uses one of the following network models:

- Kubenet networking: Network resources are typically created and configured as the AKS cluster is deployed.
- Azure Container Networking Interface (CNI) networking: AKS cluster is connected to existing virtual network resources and configurations.



Fore more information about AKS networking → Concepts - Networking in Azure Kubernetes Services (AKS) - Azure Kubernetes Service | Microsoft Learn

Upgrade an Azure Kubernetes Service(AKS) Cluster

Part of the AKS cluster lifecycle includes periodic upgrades to the latest Kubernetes version. It is important that you apply the latest security releases or upgrade to get the latest features. This article shows you how to check, configure, and upgrade upgrades to your AKS cluster.

AKS clusters that use multiple node pools or Windows Server nodes, see Upgrade a node pool in AKS. To upgrade a specific node pool without upgrading the Kubernetes cluster, see Upgrading a specific node pool.

An AKS cluster is divided into two parts: Azure-managed nodes and customer-managed nodes.

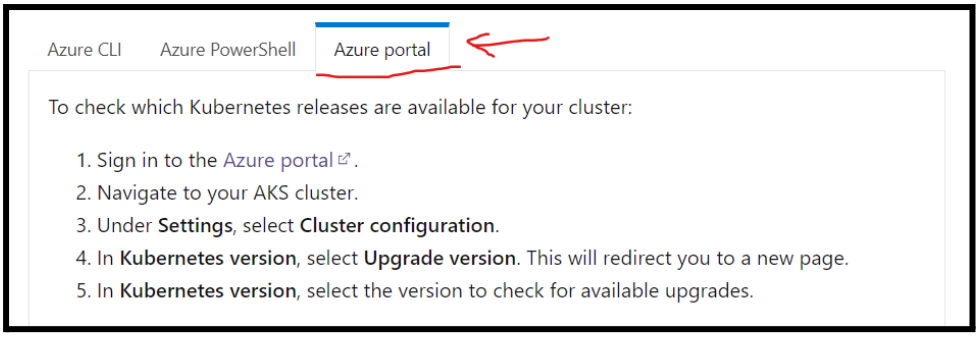
- Azure-managed nodes provide core Kubernetes services and orchestration of application workloads in your AKS cluster.
- Customer-managed nodes run your application workloads in your AKS cluster.

The below mentioned three are important during upgrading an AKS Cluster

- Check for available AKS cluster upgrades.
- Customize node surge upgrade
- Upgrade as AKS cluster

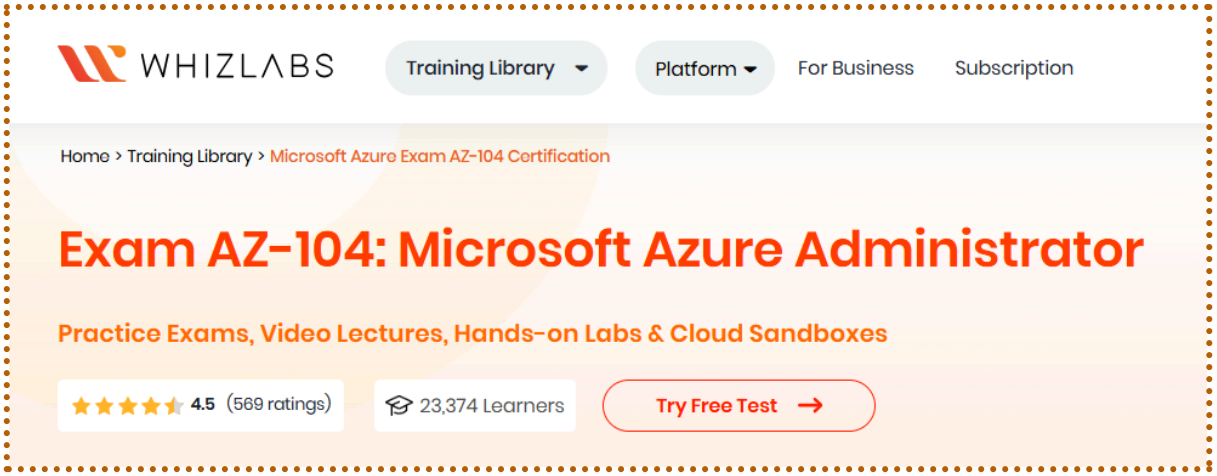
You can Upgrade your AKS cluster through →Azure Portal, Azure CLI, and Azure PowerShell

Example: Here we have provided a reference based on Azure Portal and Steps are same for both the sections: a) Check for available AKS cluster upgrades and b) Upgrade as AKS cluster

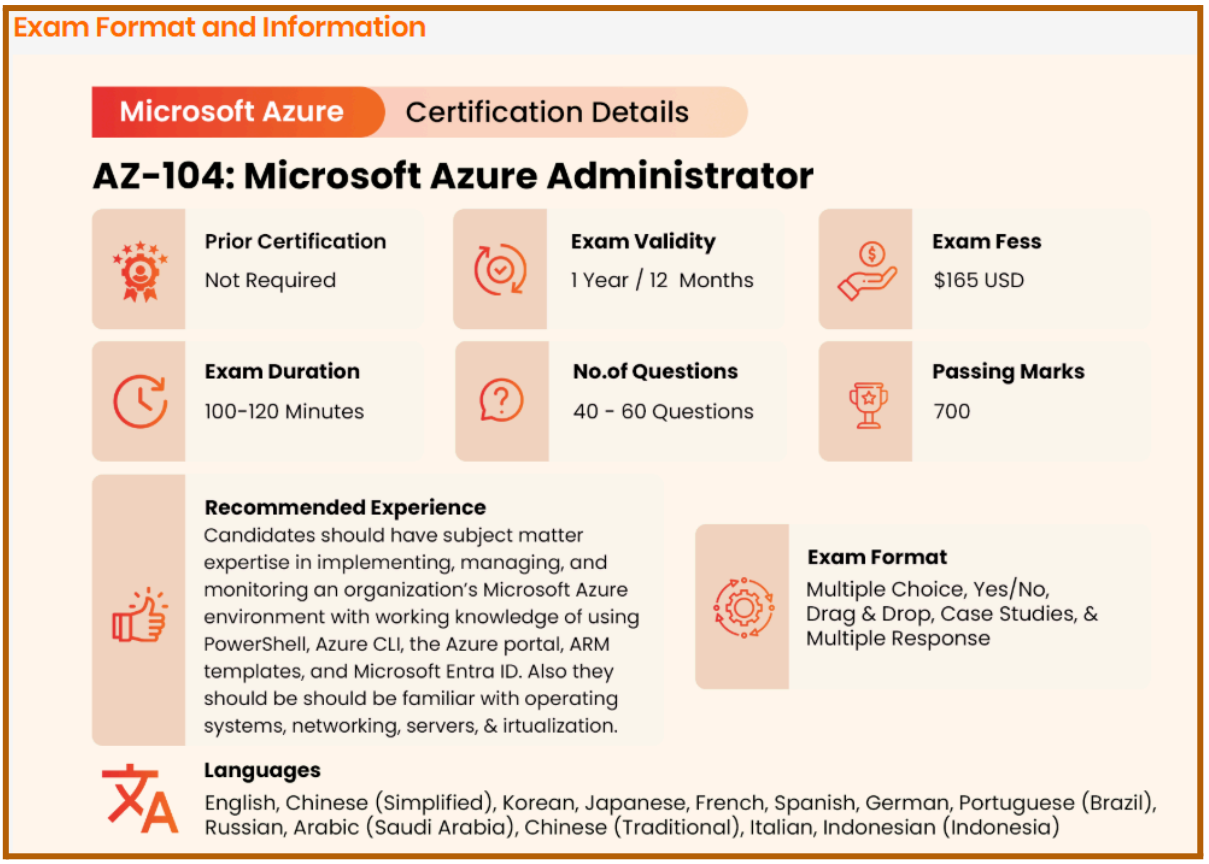


Thank you for referring the **Cheat Sheet**,

We hope the information provided will be useful to you!



Exam Format and Information



Happy Learning

--Back to Index--