24-04-2025

**Regional pairs:**

Regional pairs are pre-defined sets of two Azure regions within the same geography (e.g., within the same country or continent), created by Microsoft to ensure:

* High availability
* Data residency
* Disaster recovery (DR)

When you deploy services across a regional pair, Azure ensures at least one of the regions is available, even during large-scale outages.

**Edge Locations:**

An **Edge Location** is a small data center **closer to the end-users** that is part of a **Content Delivery Network (CDN)**. Edge locations are **distributed all around the world**—in **major cities and populated areas**. These locations are chosen to **reduce latency** by being physically closer to users

(605)

**Azure Resource Group:**

An Azure Resource Group is a container that holds related resources for an Azure solution. These resources can include virtual machines, databases, storage accounts, web apps, networks, and more.

Think of it like a folder where you group resources that share a **common lifecycle**—you deploy, manage, and monitor them together.

**Example**

Let’s say you're deploying a web app:

You can create a **resource group** called WebApp-RG and include:

* Azure App Service (for the web app)
* Azure SQL Database
* Azure Storage Account
* Application Insights (monitoring)

All these can be deployed together, monitored together, and even deleted together when no longer needed.

Azure resources:

1.Logical Containers

2.No nesting

3.Global uniqueness

Benefits:

1.Cost management: easy to tracks costs associated with in a specific application or project by grouping its resources as a single unit.

2.Logical organization: Group resources based on application , enivronement(dev,prod,test)

3. Deployment and Deletion: Deploy and delete entire application stacks or environments as a single unit.

Manage Resource Groups :

1.Azure CLI

2.Azure Portal

3.ARM Template

4.Azure Powershell

5.Azure SDK’s

**Azure resource manager:**

1.It is a deployment and management service for azure .It acts as the control plane for all the resources.

2.Consistent State Management: ARM ensures that your resources are deployed in a consistent state. If a deployment of multiple resources fails, ARM can automatically roll back the changes, preventing partially configured environments.

3.Integration with Azure AD: ARM integrates with Azure Active Directory (Azure AD) for authentication and authorization, enabling you to manage access control and enforce policies consistently across your Azure resources.

**Azure Subscriptions:**

An **Azure Subscription** is like a **container for your Azure resources**. It defines a **billing boundary** and a **management boundary** for using Azure services.

You can have many subscriptions to one account and subscription is associated to the payments.

| **Feature** | **Description** |
| --- | --- |
| **Billing Boundary** | All costs from services used under a subscription go to a single invoice. |
| **Resource Limit** | Sets quotas for resources (e.g., number of VMs, storage capacity). |
| **Access Control** | You can assign RBAC (Role-Based Access Control) at the subscription level. |
| **Isolation** | Resources in one subscription are isolated from others unless explicitly linked. |
|  |  |

**Azure Entra id:**

Microsoft Entra ID is the new name for Azure Active Directory (AzureAD).

It's Microsoft’s cloud-based identity and access management (IAM) service. It that helps organizations manage:

* **User authentication** (sign-in)
* **Access control** to apps and resources
* **Security policies** (like MFA, conditional access)
* **Device identity**
* **App registration & SSO**

🧠 Think of Entra ID as the *"brain"* behind login and identity in Azure and Microsoft Cloud.

**Azure tenant:**

**🏢 What is an Azure Tenant?**

Definition:

An Azure Tenant is a dedicated, isolated instance of Microsoft Entra ID assigned to an organization when they create a Microsoft cloud account (Azure, Microsoft 365, etc).

It’s like your company's private identity space in the Microsoft cloud.

It's like your organization's private version of Microsoft's identity system.

Each tenant:

* Is unique to your organization
* Has its own users, groups, apps, policies
* Is tied to a domain name, like yourcompany.onmicrosoft.com
* Can have one or more subscriptions

**LAB : VNET PEERING**

**Step 1 :** Created a two vnets. (vnet1,vnet2)

**Step 2 :** Created one subnet in each of the vnet : Subnet1 , Subnet 2

**Step 3** : Created VM’s for subnets : VM1,VM2

**Step4 :** We will go to the both NSG’s and allow ICMP protocol for the purpose of communication .

**Step 5:** We will create peering for the vnet1-vnet2 and vnet2-vnet 1

Step 6 : To check whether peering has been done or not :

In VM1 :

ping @private ip of Vm2

In VM2:ping@private ip of Vm1

**Created a user from the main account**

**Step 1:**  Go to entra id and click on add then click on create new user

**Step 2:**  fill required details.

**Step 3 :** add group

**Step 4:** then review and create