**🔷 1. Virtual Network (VNet)**

A **VNet** in Azure is like a virtual version of a physical network in your on-premises setup. It gives you control over:

* IP address ranges
* Subnets
* Route tables
* Network security settings

Think of it as a **private, isolated network** in the Azure cloud where you can deploy Azure resources (VMs, containers, databases, etc.) and control how they connect with each other or the internet.

🔹 **Analogy**: A VNet is like a whole building. Inside, you can create rooms (subnets) and install doors (NSGs) to control who enters and exits.

**🔷 2. Subnet**

A **Subnet** is a range of IP addresses within a VNet. You divide a VNet into multiple subnets to better organize and secure your resources.

🔹 **Use cases**:

* Place different services in different subnets (e.g., frontend in one, database in another).
* Apply different security and routing rules.

🔹 **Analogy**: If a VNet is a building, subnets are rooms in that building.

**🔷 3. Network Security Group (NSG)**

An **NSG** acts like a **firewall**. It lets you control **inbound** and **outbound** traffic to/from network interfaces, VMs, or subnets.

🔹 You define **rules**:

* Source/destination IP
* Protocol (TCP/UDP)
* Port range
* Allow/Deny action

You can attach an NSG to:

* A **subnet** (affects all resources in that subnet)
* A **network interface** (affects a single VM)

**🔹 Public vs Private NSGs (or subnets)**

There’s no official term like "public NSG" or "private NSG", but this is how we typically refer to them:

| **Term** | **Description** |
| --- | --- |
| **Public Subnet/NSG** | Allows **inbound traffic from the internet**, e.g., for a web server. NSG rule: allow port 80/443 from Internet. |
| **Private Subnet/NSG** | Denies direct internet access. Only internal or specific traffic is allowed. NSG rule: deny all inbound from Internet. |

💡 **Note**: What makes a subnet "public" or "private" isn’t just the NSG—it's also if the subnet’s resources have **public IPs** or if **NAT gateway** is configured for outbound access.

**🧠 Summary Table:**

| **Concept** | **What It Is** | **Key Use** |
| --- | --- | --- |
| **VNet** | Private Azure network | Controls IPs, subnets, routes |
| **Subnet** | Division of a VNet | Logical separation of resources |
| **NSG** | Firewall rules | Controls traffic in/out of subnet or VM |
| **Public Subnet** | Subnet with internet-facing resources | Used for frontends, web apps |
| **Private Subnet** | Subnet with no direct internet exposure | Used for databases, internal services |
|  |  |  |

**🔹 1. Azure Group**

**✅ What is it?**

An **Azure AD group** is a collection of users, devices, or other groups that can be managed as a single unit.

**✅ Types of groups:**

* **Security groups**: Used to manage user and device access to resources.
* **Microsoft 365 groups (formerly Office 365 groups)**: Used for collaboration between users, with shared resources like Outlook, SharePoint, Teams, etc.

**✅ Why use groups?**

* Simplify **access management**: Assign permissions to a group instead of individuals.
* Apply **policies or licenses** to multiple users at once.
* Automate **user management** using dynamic group membership rules.

**🔹 2. Licenses**

**✅ What are licenses?**

Licenses grant users access to Azure and Microsoft cloud services like Microsoft 365, Intune, Power Platform, or Enterprise Mobility + Security (EMS).

**✅ License types:**

* **User-based**: Assigned to individual users or groups (e.g., Microsoft 365 E5).
* **Group-based licensing**: You can assign licenses to a group, and all users in the group inherit the license.

**✅ Key features of group-based licensing:**

* Automates license assignments when users are added/removed from the group.
* Reduces manual errors.
* Supports **inheritance**: Nested groups are not supported directly, but dynamic groups help manage based on user attributes.

**🔹 3. Azure Role Assignment**

**✅ What is it?**

Azure Role Assignment refers to the process of giving a **user, group, or service principal** specific **permissions** to Azure resources via **RBAC (Role-Based Access Control).**

**✅ Key terms:**

* **Principal**: The user, group, or application getting access.
* **Role Definition**: A collection of permissions (like "Reader", "Contributor", "Owner", or custom roles).
* **Scope**: The level at which access is granted: **subscription**, **resource group**, or **resource**.

**✅ How it works:**

To assign a role, you define:

* **Who** (user/group/service principal)
* **What** role (e.g., Reader)
* **Where** (scope of access: e.g., a particular resource group)

**✅ Example:**

You want all members of a group called "DevOps Team" to have Contributor access to a resource group:

bash

CopyEdit

az role assignment create \

--assignee "<objectId\_of\_group>" \

--role "Contributor" \

--scope "/subscriptions/<sub\_id>/resourceGroups/<resource\_group\_name>"

**🧩 How They Work Together**

Imagine you’re managing a team of developers:

1. You create a **security group** in Azure AD called DevTeam.
2. You assign a **Microsoft 365 E3 license** to the group, so all devs get access to email, Teams, etc.
3. You use **Azure Role Assignment** to give the group Contributor access to the development resource group in Azure.

**🔹 1. Subscription**

**✅ What is an Azure Subscription?**

An **Azure Subscription** is like a **container** for:

* Resources (VMs, databases, storage, etc.)
* Billing (it defines how you're charged)
* Access control (IAM roles and policies)

**✅ Key Points:**

* Every subscription is linked to **an Azure Active Directory (Azure AD)** tenant.
* **Multiple subscriptions** can exist under one Azure AD tenant.
* Subscriptions help **separate resources** (e.g., Dev vs Prod) or organize them by project, department, or customer.

**✅ Types of subscriptions:**

* **Free trial**
* **Pay-As-You-Go**
* **Enterprise Agreement**
* **Microsoft Customer Agreement**

**✅ Example:**

If you work in a company, you might have:

* A subscription for **Development**
* A subscription for **Testing**
* A subscription for **Production**

Each one can have its own billing and policies.

**🔹 2. User**

**✅ What is a User in Azure?**

A **user** is an identity in **Azure Active Directory (Azure AD)**—this could be:

* An employee (john@yourcompany.com)
* An external consultant (guest user)
* A system-managed account (like a service principal)

**✅ User characteristics:**

* Users can log in to Azure Portal, PowerShell, or CLI.
* Users are granted access to Azure resources through **Role Assignments (RBAC)**.
* Users can be added to **groups**, assigned **licenses**, or have **conditional access policies** applied.

**🔹 3. Resource Group**

**✅ What is a Resource Group?**

A **Resource Group** is a **logical container** for Azure resources.

**✅ Purpose:**

* Organize resources like VMs, databases, and storage that share a common **lifecycle**, **permissions**, and **policies**.
* Makes it easy to **deploy**, **manage**, **monitor**, and **delete** resources together.

**✅ Example:**

If you're deploying a web app:

* You might create a resource group called WebApp-RG
* Inside it, you put:
  + App Service
  + SQL Database
  + Storage Account
  + Virtual Network

You can then:

* Apply a **role** to a team for that resource group
* Apply **tags** or **cost tracking**
* Delete everything at once by deleting the resource group

**🧩 How They All Connect**

Let’s say you’re working on a project:

* Your company has an **Azure Subscription** named DevOps-Sub.
* You're a **user** in Azure AD (alice@contoso.com) and part of a DevTeam group.
* Your project resources are in a **resource group** called MyApp-RG under that subscription.
* You’re given Contributor role access to MyApp-RG, so you can deploy and manage all resources in it.

**🔹 1. Account**

**✅ What is an Azure Account?**

An **Azure Account** is typically your **email address** (Microsoft account or Work/School account) used to sign in to the Azure portal.

**✅ The account is used to:**

* **Sign in** to the Azure portal or CLI.
* **Manage subscriptions** (you can own or be a part of multiple).
* Handle **billing & payments**.
* Manage access to **Microsoft services** (Azure, Microsoft 365, Power Platform, etc.)

There are two types of Azure accounts:

| **Type** | **Example** | **Description** |
| --- | --- | --- |
| **Microsoft account** | yourname@hotmail.com | Used for personal Azure subscriptions. |
| **Work or School account (Entra ID)** | john@contoso.com | Managed by organizations through Entra ID (Azure AD). |

**🔹 2. Subscription = Associated with Payment**

**✅ What is an Azure Subscription?**

A **Subscription** is linked to your Azure account and defines:

* **Billing** (how you're charged)
* **Resource usage limits**
* **Access control scope**

You can have **multiple subscriptions** for different purposes, like:

* Development
* Production
* Testing
* Clients/Departments

💳 **Billing is per subscription.** You receive **one invoice** per subscription.

For example, if you deploy 3 VMs in a subscription, their costs will be aggregated and billed together under that subscription.

**🔹 3. Tenant = Space in Azure Cloud**

**✅ What is a Tenant?**

A **Tenant** is a **dedicated instance of Microsoft Entra ID (formerly Azure AD)**.

**✅ Think of it like:**

* Your own **secure container** in Microsoft’s cloud.
* A separate **identity and policy boundary**.

**✅ A tenant includes:**

* All **users, groups, and service principals**
* All **authentication** and **security settings**
* One or more **subscriptions** can be linked to a single tenant.

A tenant is **created automatically** when you sign up for Azure with a Work or School account.

**🔹 4. Entra ID (Azure AD) = Directory of Users**

**✅ What is Entra ID?**

**Microsoft Entra ID** is the **cloud-based identity service** that manages:

* Users
* Groups
* Devices
* Authentication
* Role assignments

It's the **backbone** for all user identities in Microsoft cloud services.

**✅ Entra ID functions like a cloud database of identities:**

* It holds **user accounts** and their properties.
* Controls **who can log in** and **what they can do**.
* Supports **RBAC**, **MFA**, **SSO**, and **conditional access**.

Every Azure subscription is **associated with one Entra ID tenant**, which handles identity and access control.

Summary Diagram (Text Format)

[Azure Account]

|

---------------------

| |

[Subscription-1] [Subscription-2]

| |

[Resource Groups] [Resource Groups]

|

[Azure Resources]

↑ Subscriptions belong to ↑

[Tenant]

|

[Microsoft Entra ID (AAD)]

|

[Users, Groups, Policies]

**🔸 Real-Life Analogy**

Imagine:

* **Tenant** = Your company (like the whole office building)
* **Entra ID** = The employee database and badge system
* **Account** = You, with your badge
* **Subscription** = A department’s cost center
* **Resource Group** = A specific project team’s workspace

**🔹 What is a CDN?**

A **Content Delivery Network (CDN)** is a globally distributed network of **servers (edge nodes)** that deliver content (like images, videos, CSS, JavaScript, etc.) to users **from a location closest to them**.

Microsoft offers **Azure CDN**, and there are other providers like Cloudflare, Akamai, and AWS CloudFront.

**🔹 What Happens When an Image Is Accessed via CDN?**

Let’s say you have an image hosted on your server or storage account (e.g., in Azure Blob Storage), and you configure it to be served through a CDN.

**✅ First request:**

1. A user requests an image (e.g., https://cdn.example.com/image.jpg)
2. The CDN checks the **closest edge location** (based on the user’s geographic location).
3. If the image **is not cached** at that edge location:
   * The CDN **fetches it from the origin** server (e.g., your Azure Storage).
   * Stores (caches) a **copy locally** at the edge server (this is called **populating the cache**).
4. The image is returned to the user.

**✅ Next requests:**

* Any user near that edge location will now get the image **directly from the CDN cache**, which is much faster.
* The image stays cached for a certain **time-to-live (TTL)** or until it's updated or purged.

**🔹 Local Availability Zone = Edge Server (CDN POP)**

When you mentioned "**local availability zone**", you’re referring to **CDN edge servers** that are located in many **regions or cities** globally—close to the user. These are often called **Points of Presence (PoPs)**.

The edge servers:

* Reduce **latency** (faster loading)
* Save **bandwidth** and **server load**
* Enable **failover and resilience**

So yes—**the image is continuously downloaded once** to each CDN edge server as needed, and then it’s served locally from there for future users in that region.

**🔹 How This Works in Azure CDN**

1. **You configure your origin** (e.g., Azure Blob Storage, Web App).
2. **Azure CDN** creates a **CDN endpoint**.
3. When users access content:
   * Azure CDN routes them to the **nearest edge location**.
   * If not cached, it pulls from your origin and caches it.
   * Then serves from local edge cache for next requests.

**🔄 Diagram (Text)**

sql

CopyEdit

User (India) ----→ CDN Edge Server (India) ----→ Origin Server (e.g., Azure Storage)

↑ ↓

Cached copy stored locally Image fetched once (on first request)

**🔍 Example Use Case**

Let’s say your website serves a lot of product images:

* Without CDN: All users hit your server, slow for international users.
* With CDN: Users in Tokyo, New York, London get fast image loads from **local edge servers**.

**Summary**

| **Term** | **Meaning** |
| --- | --- |
| CDN | A network of distributed servers that cache content close to users |
| Edge Server | Local server (availability zone) that stores cached content |
| Origin | Your main server (e.g., Azure Blob Storage) |
| TTL | Time content stays cached in CDN |
| Cache | Stored copy of content on the edge |

**🔹 1. Cost Alerts in Azure**

**✅ What is a Cost Alert?**

A **cost alert** notifies you when your **Azure spending reaches or exceeds a set threshold**. It helps you stay within budget and avoid unexpected bills.

Azure provides **three types** of cost-related alerts:

**🔸 1. Budget Alerts (Most common)**

* You define a **budget** for a **subscription**, **resource group**, or **service**.
* Azure tracks spending against the budget.
* You get **alerts** when usage hits a **percentage threshold** (e.g., 80%, 90%, 100%).

**Example:**

You create a budget of $500/month. You set alerts for:

* 80% = $400 → Get notified.
* 100% = $500 → Get notified again.

**Note**: Budget alerts don’t stop spending—they just notify you. You’d need automation or policy to enforce that.

**🔸 2. Credit Alerts**

* For accounts using **Azure credits** (like Visual Studio subscriptions).
* You get notified when credits are low or used up.

**🔸 3. Department/Enrollment Alerts (for Enterprise Agreements)**

* Used in **Enterprise Azure Portal** (EA).
* Set alerts at **department level** or **enrollment level**.

**🔹 2. Alert Rules in Azure**

**✅ What are Alert Rules?**

**Alert Rules** are part of **Azure Monitor**, and they help you **automatically detect and respond** to specific conditions or issues.

These are broader than cost alerts—they can monitor **metrics, logs, activity logs, or costs**.

**🔸 How an Alert Rule Works:**

An alert rule has **4 key parts**:

| **Component** | **Description** |
| --- | --- |
| **Scope** | What resource(s) the rule applies to (e.g., VM, Storage, Subscription) |
| **Condition** | What you want to monitor (e.g., cost > $100, CPU > 80%) |
| **Action Group** | What happens when triggered (e.g., send email, run Logic App, call webhook) |
| **Severity** | From 0 (Critical) to 4 (Informational) |

**🔸 Example Use Cases:**

| **Scenario** | **Alert Rule** |
| --- | --- |
| Spend goes above $1000/month | Cost-based budget alert |
| CPU on a VM > 90% for 5 mins | Metric alert |
| Unauthorized login attempt | Log alert |
| App Service returns 500 errors | Log-based alert |

**🔄 How They Work Together**

You can use **cost alerts** for budgeting and pair them with **action groups** in **alert rules** to automate a response.

For example:

* Create a **budget** of $1000 for a subscription.
* Set an alert at 90%.
* Create an **alert rule** to **notify your team on Teams + send an email + trigger a script to scale down resources**.

**🧠 Key Terms to Know**

| **Term** | **Meaning** |
| --- | --- |
| **Budget** | Limit you define for costs |
| **Cost Alert** | Notification when you approach or exceed the budget |
| **Alert Rule** | Azure Monitor rule that checks for a condition and triggers action |
| **Action Group** | Defines how you get notified (email, SMS, webhook, Logic App) |
| **Log Alert** | Based on Azure logs (Activity log, diagnostics) |
| **Metric Alert** | Based on performance metrics (CPU, Memory, Cost, etc.) |

**📊 Where to Create These?**

* **Cost Alerts** → Go to **Cost Management + Billing** > Budgets > Add Budget.
* **Alert Rules** → Go to **Azure Monitor** > Alerts > Create Alert Rule.