Azure API Training <> Revanture

Day 2

Create First Resource in Azure

Project

Warnings

- Don’t leave Virtual Machines or Databases running — stop or delete them after practice.
- Avoid using large or paid resources — stick to the free tier or smallest options.
- Don’t delete default resources or system-generated items in resource groups.
- Never share or expose your access keys, passwords, or connection strings.
- Always check your region and monitor your usage and costs regularly.
- Use separate resource groups for each project — easy cleanup, no surprise charges.

What is an Azure Region?

What is an Azure Region?

- An Azure Region is a geographical area that contains one or more data centers, connected through a dedicated, low-latency network.

- It allows Microsoft to provide cloud services closer to customers, ensuring performance, availability, and compliance with local regulations.

Advantages?

Why Azure Uses Regions (Advantages)

Data Residency & Compliance

- Organizations can store data within their own country or region to comply with local laws (like GDPR, India's Data Protection Act, etc.).

Low Latency & Better Performance

- Resources are deployed closer to users, improving speed and responsiveness.

Service Availability

- Regions provide redundancy; if one goes down, others can handle workloads.

Scalability

- Users can scale resources easily in multiple regions worldwide.

Why Azure Uses Regions (Advantages)

Cost Optimization

- Different regions may offer different pricing, allowing cost-effective deployment.

Different Regions

Azure Global Regions

Azure provides the most extensive global footprint of any cloud provider and is rapidly opening new regions.

Azure Regions List

Questions?

Availability Zone?

What is an Availability Zone?

An Availability Zone (AZ) is a physically separate datacenter within an Azure region.

Each zone has independent power, cooling, and networking to protect applications from datacenter-level failures.

To ensure high availability (HA) and fault tolerance by isolating workloads within a region.

Each region typically has at least three Availability Zones.

These zones are connected by high-speed, private fiber networks.

What is an Availability Zone?

In the Central India region, there may be

- Zone 1 (Datacenter A)
- Zone 2 (Datacenter B)
- Zone 3 (Datacenter C)

Advantages?

Advantages of Availability Zones?

High Availability (99.99% Uptime)

- Even if one zone fails due to a power outage or disaster, the others keep services running.

Fault Isolation

- Each zone is isolated — so failures in one don’t affect others in the same region.

Disaster Recovery within a Region

- Data and applications can be replicated across zones for resilience without leaving the region.

Scalable and Redundant Deployments

- Users can distribute VMs or databases across zones for load balancing and better performance.

Advantages of Availability Zones?

Business Continuity

- Guarantees mission-critical apps remain available even in adverse conditions.

Example — Availability Zones in Action

Scenario

A company deploys its web application in East US with:

- Zone 1: Web servers
- Zone 2: Database servers
- Zone 3: Backup and failover instances

If Zone 1 experiences an outage, traffic automatically redirects to Zone 2 — ensuring uninterrupted service.


Azure region 1
Availability zone 1
Datacenter(s)
Availability zone 2
Datacenter(s)
Availability zone 3
Datacenter(s)
Azure region 2
Availability zone 1
Datacenter(s)
Availability zone 2
Datacenter(s)
Availability zone 3
Datacenter(s)
Azure region 3
Datacenter(s)
Azure region 4
Datacenter(s)

Questions?

Resource Group

What is a Resource Group?

- A Resource Group (RG) is a logical container in Azure that holds related resources for an application, workload, or environment.
- It helps manage, monitor, and control access to all associated resources as a single unit.
- To organize and manage Azure resources efficiently — such as virtual machines, databases, and storage accounts — based on lifecycle or project.

Example

- Resource Group: **RG-WebApp-Prod**
  - Virtual Machine (App Server)
  - SQL Database
  - Storage Account
  - Virtual Network

Advantages of Resource Groups?

Easy Organization

- Group related resources together for clarity and management simplicity.

Access Control (RBAC)

- Apply role-based access control to restrict permissions at the resource group level.

Cost Tracking & Budgeting

- Helps monitor and analyze cost usage per project or environment.

Unified Management

- Perform actions like start, stop, or delete on all resources in one go.

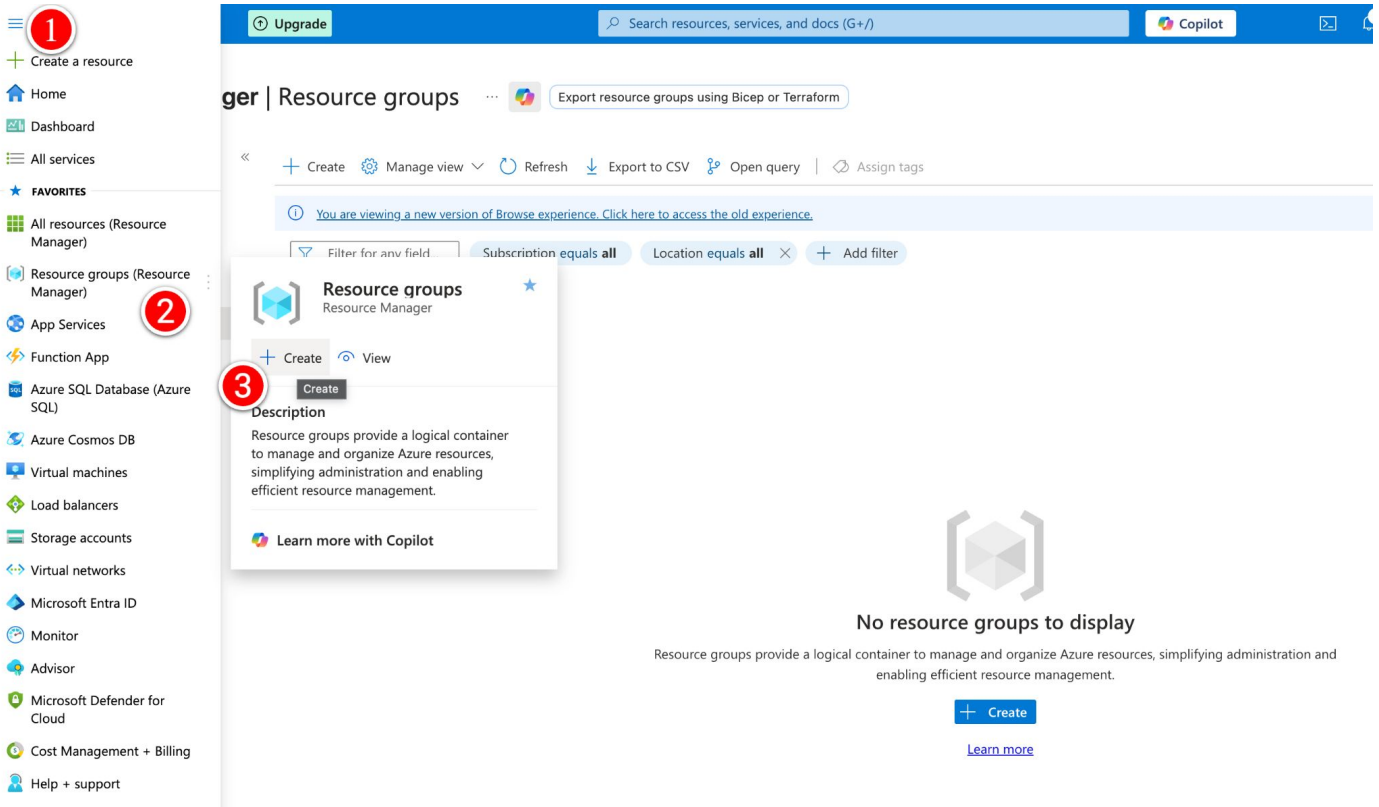
Advantages of Resource Groups?

Automation & Deployment

- Supports Azure Resource Manager (ARM) templates for consistent, repeatable deployments.

Create Resource Group

Create Resource Group



Questions?

Azure Subscription

What is an Azure Subscription?

It defines who can use Azure services, how much they can use, and how they'll be billed.

Why Azure Uses Subscriptions (Purpose)

Resource Organization

- Separate workloads or environments — e.g., Development, Testing, Production.

Access Management

- Apply Role-Based Access Control (RBAC) at the subscription level.

Billing & Cost Tracking

- Each subscription gets its own invoice and usage reports.

Scalability & Governance

- Enterprises can manage multiple subscriptions under one Azure account (Tenant).

Why Azure Uses Subscriptions (Purpose)

Separation of Responsibility

- Different teams or departments can work in isolated subscriptions safely.

Azure Subscriptions types

Subscription Type	Description	Example Use
Free Account	12 months of free services + ₹14,500 credit	Learning or testing
Pay-As-You-Go	Pay only for what you use	Small businesses, startups
Student Subscription	Free access for verified students (no credit card)	Cloud training or labs
Enterprise Agreement (EA)	Bulk subscription for large organizations	Corporate IT workloads
Microsoft Customer Agreement (MCA)	Flexible, modern billing agreement	Mid-size enterprises

Questions?

Azure Billing

Azure Billing Concepts

Pay-as-you-go Model

- You’re charged **only for the resources you consume** — like electricity or mobile data.

Meters & Units

- Each service (VM, Storage, Database) has its **own meter** that tracks usage per hour or GB.

Billing Cycle

- Azure bills monthly. You can view details in the **Cost Management + Billing** section.

Resource Tags for Cost Management

- Add labels like `Project=WebApp` or `Team=DevOps` to identify where costs come from.

Example — Billing in Action

Resource	Meter	Usage	Cost (per month)
Virtual Machine	Compute Hours	730 hrs	₹1,200
Storage	GB Stored	100 GB	₹100
SQL Database	DTUs (Compute Units)	30 days	₹500
Total			**₹1,800/month**

Questions?

Resource Management Hierarchy and Organization

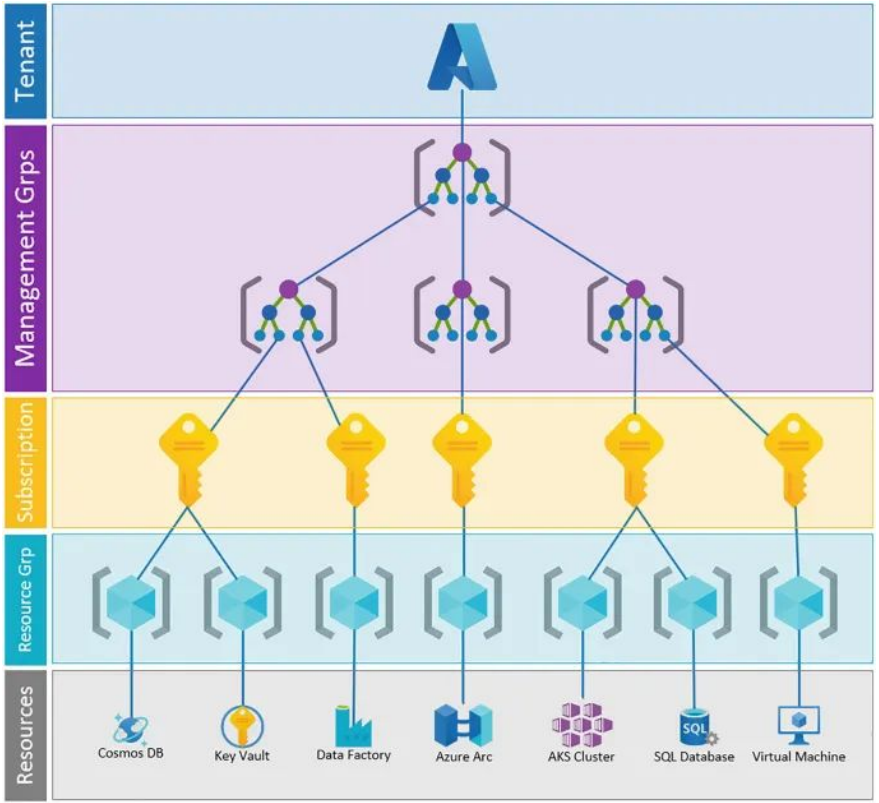
What is Azure Resource Hierarchy?

Azure resource hierarchy is the way resources and services are organized within Microsoft Azure. It provides a logical structure for managing, accessing, and controlling cloud resources.

The hierarchy consists of four key levels: Management Groups, Subscriptions, Resource Groups, and Resources.

Understanding this structure helps with efficient governance, cost tracking, and access management.

What is Azure Resource Hierarchy?



Tenant (Top Layer — Blue)

- The Tenant represents your Microsoft Entra ID (previously Azure AD) — the identity layer of Azure.
- It defines who you are in Azure — your organization or account boundary.

Management Groups (Purple Layer)

- Management Groups organize multiple subscriptions under one umbrella.
- They allow applying policies, security rules, and access controls across several subscriptions at once.

Subscriptions (Yellow Layer)

- Each subscription represents a billing and access boundary.
- You pay for what’s inside your subscription — all resources created within it are billed together.

Resource Groups (Teal Layer)

- A Resource Group (RG) is a logical container for related resources.
- It helps manage, monitor, and delete everything in that project together.

Resources (Bottom Layer — Grey Icons)

- Resources are the actual services you use in Azure.
- Each resource lives inside exactly one Resource Group and one Subscription.

Examples of resources:

- Azure Arc
- AKS Cluster
- SQL Database
- Virtual Machine

Summary

Hierarchy Level	Example	Purpose
Tenant (Entra ID)	google.onmicrosoft.com	Identity & Access Control
Management Group	Prod-MG, Dev-MG	Apply governance across subscriptions
Subscription	Dev-Sub, Prod-Sub	Billing & Access Boundary
Resource Group	RG-WebApp, RG-DB	Logical grouping of resources
Resources	VM, SQL, Storage	Actual Azure services

Questions?

Azure Portal

What is Azure Portal?

- The Azure Portal is a web-based interface provided by Microsoft to create, configure, and manage all Azure services in one place.

Features of Azure Portal?

User-Friendly Interface

- Perfect for beginners — no coding needed.

Real-Time Feedback

- See live updates as resources are created or changed.

Integrated Tools

- Works seamlessly with Azure CLI, PowerShell, and REST APIs.

Consistent Experience

- Same interface across browsers and devices.

Warnings & Best Practices for Beginners

Check Region Before Creating Resource

- Choose a nearby region (e.g., *Central India*) to avoid latency and compliance issues.

Avoid Unnecessary Resource Creation

- Some services (like VMs or Databases) charge hourly — even when idle.

Don’t Delete Resource Groups Carelessly

- Deleting a resource group removes *everything* inside it.

Always Monitor Costs

- Use **Cost Management + Billing** to track your daily spend.

Warnings & Best Practices for Beginners

Be Careful with Access Permissions

- Don’t assign “Owner” roles casually — use **least privilege principle**.

Demo

Questions?

Azure CLI

What Azure CLI?

The Azure Command-Line Interface (Azure CLI) is a cross-platform tool used to manage Azure resources through text-based commands instead of the portal.

It allows users to automate tasks, deploy services, and manage infrastructure directly from a terminal or script.

Key Features:

- Works on **Windows, macOS, and Linux**.
- Uses **commands starting with az**.
- Ideal for **DevOps, scripting, and automation**.
- Can be run in **Cloud Shell** or **locally installed**.

How to install Azure CLI?

The Azure CLI is available to install in Windows, Linux, and macOS environments. It can also be run in a Docker container and Azure Cloud Shell.

Download Azure CLI / Follow Instructions

Follow the above link for latest instructions

Basic commands of Azure CLI

- Login to Azure CLI
  - az login
- Create resource group with Azure CLI
  - az group create --name myRg --location eastus
- View active subscription
  - az account show
- List All Subscriptions
  - az account list --output table
- List All Resource Groups
  - az group list --output table / yaml
- Check Resource Usage and Costs
  - az consumption usage list --output table

Questions?

Azure PowerShell

Azure PowerShell

Azure PowerShell is a set of cmdlets (command-lets) built on the PowerShell framework that allows you to manage Azure resources directly from the command line or scripts.

Purpose:

- Automate tasks and manage resources through **scripts**.
- Ideal for **Windows administrators** and **DevOps engineers**.
- Can perform **everything the Azure Portal can — faster and repeatably**.

Why Use Azure PowerShell?

Automation and Scripting

- Automate repetitive deployment or maintenance tasks.

Integration with Windows

- Works seamlessly with Windows tools and administrative scripts.

Consistency

- Same commands can be reused across environments and projects.

Ideal for Infrastructure as Code (IaC)

- Combine with scripts or pipelines for DevOps workflows.

Install Azure PowerShell

The Az PowerShell module is a rollup module. Installing the Az PowerShell module downloads the generally available modules and makes their cmdlets available for use.

Download and Install Azure PowerShell

Basic commands of Azure PowerShell

- List Subscriptions
  - Get-AzSubscription | Format-Table
- Set Active Subscription
  - Set-AzContext -Subscription "sub_id"
- Create a Resource Group
  - New-AzResourceGroup -Name "RG-Demo_PS" -Location "EastUS"
- List Resource Groups
  - Get-AzResourceGroup | Format-Table
- View All Available Azure Locations
  - Get-AzLocation | Format-Table

Azure CLI vs Azure PowerShell

Feature / Category	Azure CLI	Azure PowerShell
Purpose	Command-line tool to manage Azure resources using simple commands	PowerShell module with cmdlets for advanced automation and scripting
Command Style	Linux-style commands (starts with `az`)	PowerShell cmdlets (verb-noun format like `Get-AzVM`)
Syntax Example	`az group create --name RG1 --location eastus`	`New-AzResourceGroup -Name RG1 -Location EastUS`
Language Type	Text-based (command-line)	Object-based (PowerShell framework)
Output Format	JSON (default), table, TSV, YAML	PowerShell objects (can be piped, formatted, or filtered)
Best Suited For	Developers, testers, Linux/macOS users	System admins, DevOps engineers, Windows users
Learning Curve	Easier for beginners	Slightly steeper (requires PowerShell basics)
Login Command	`az login`	`Connect-AzAccount`
Ideal Use Case	Quick resource management and automation	Complex automation, orchestration, and integration with Windows tools

Questions?

Azure Identity and Access Management (IAM)

What is IAM in Azure?

Identity and Access Management (IAM) in Azure is the framework that ensures the right people have the right access to the right resources.

It combines authentication (proving who you are) and authorization (what you're allowed to do).

Purpose:

- Protect resources from unauthorized access.
- Provide controlled and auditable access to Azure services.
- Manage users, groups, and service identities across the organization.

Why IAM is Important?

Security and Compliance

- Prevent unauthorized access to critical data and apps.

Access Governance

- Manage who can view, create, or delete resources.

Centralized Identity Management

- One identity for all Microsoft cloud services (Azure, Microsoft 365, etc.).

Automation and Scalability

- Use groups and roles to manage access at scale.

IAM in Azure — Key Building Blocks

Component	Purpose	Example
Microsoft Entra ID	Identity provider for Azure users, groups, and apps	User: *Avinash@google.onmicrosoft.com*
User	Represents a person who can access Azure	*user@google.onmicrosoft.com*
Group	Collection of users with similar access rights	*DevelopersGroup*
Service Principal	Identity for an application or automation script	*App registration for HR app*
Managed Identity	Auto-created identity for an Azure resource	*VM connecting securely to Azure Key Vault*
Role Assignment	Defines what actions are allowed	*Contributor role on RG-WebApp*

What is Role-Based Access Control (RBAC)?

RBAC in Azure allows administrators to assign specific permissions to users, groups, or services based on roles.

Purpose:

- Grant the **minimum access required** (“Least Privilege Principle”).
- Control actions like *Read, Write, Delete, or Manage*.

RBAC Components:

- **Security Principal** — Who? (User, Group, Service Principal)
- **Role Definition** — What can they do?
- **Scope** — Where? (Subscription, Resource Group, or Resource)

Built-in Roles in Azure

Role Name	Access Level	Example Use
Owner	Full access, can assign roles	Project manager or admin
Contributor	Can create/manage resources, but not assign roles	Developer
Reader	Can only view resources	Auditor or monitoring user
User Access Administrator	Can manage role assignments	IAM manager or team lead

Demo

Questions?