

Azure Portal Overview

What is Azure Portal?

A web-based interface to manage Azure resources.

Allows you to create, manage, and monitor Azure services in one place.



Azure Portal Basics.

- Access Azure Portal:
 - Visit [Azure Portal](#).
 - Log in with your Azure account credentials.
- Search for Services:
 - Use the Search Bar at the top.
 - Type Virtual Machines or Resource Groups to navigate quickly.
- Customize Dashboard:
 - Click Dashboard > Edit Dashboard.
 - Drag widgets (like CPU or Network Metrics) to monitor your resources.
 - Save the changes.

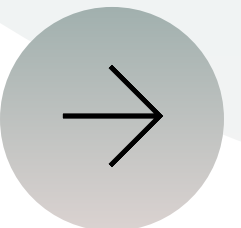


Azure Regions

What are Azure Regions?

A region is a geographical area where Azure resources are deployed.

Each region contains multiple datacenters



Objective: Deploy a Virtual Machine in a specific region.

- Navigate to Virtual Machines > Create.
- Configure Basic Settings:
 - Subscription: Choose your Azure subscription.
 - Resource Group: Create or use an existing group (e.g., LabRG).
 - Region: Select a region, such as East US.
 - VM Name:
 - Enter RegionLabVM.
- Configure VM Settings.
 - Size: Choose a size, such as Standard B2s.
 - Authentication: Select Password, and
 - set Username and Password.
- Review and Create:
 - Click Review + Create, then Create.
 - Monitor the deployment in the Notifications section.



Availability Zones

What are Availability Zones?

Physically separate datacenters within a region.

Designed to offer high availability and protect against datacenter failures.



Objective: Deploy VMs in two different Availability Zones.

- Navigate to Virtual Machines > Create.
- Configure VM1:
 - Set Region to a zone-supported region (e.g., East US).
 - Under Availability Options, select Availability Zone.
 - Choose Zone 1 for VM1.
- Configure VM2:
 - Repeat the process for another VM.
 - Select Zone 2 for VM2.
- Deploy:
 - Deploy both VMs and ensure they are in different zones.
- Optional: Use a Load Balancer:
 - Go to Load Balancers > Create.
 - Assign both VMs as backend pool members to distribute traffic.



Availability Sets

What are Availability Sets?

Logical grouping of VMs to ensure they are distributed across fault domains and update domains.

Protects against hardware failures and planned maintenance.



Objective: Create an Availability Set for Fault and Update Domain separation.

Fault and Update Domain separation.

- Navigate to Virtual Machines > Create.
- Under Availability Options, select Availability Set.
- Click Create New Availability Set:
 - Set Fault Domains: 2.
 - Set Update Domains: 2.
 - Name the Availability Set as LabAvailabilitySet.
- Deploy VMs:
 - Create two VMs and assign them to the LabAvailabilitySet.
 - After deployment, check the VM properties to confirm fault/update domain allocation.



How to Verify the Setup

Regions:

- Go to the VM Overview page and check the Region property.

Availability Zones:

- Verify the Zone assignment under VM Settings.

Availability Sets:

- Open the Availability Set in the Azure Portal.
- Check the Fault Domain and Update Domain placement of each VM.



Azure Resource Groups Overview

What is a Resource Group?

- A logical container for Azure resources (VMs, storage accounts, databases, etc.).
- Simplifies resource management (deployment, monitoring, billing).

Example:

- Resource Group Name: MyAppResources.
 - Resources:
 - VM1 (Backend Server).
 - Storage Account (File storage).
 - SQL Database (App database).

Key Features:

- Resources in a group share the same lifecycle.
- Apply tags for better organization.



Creating a Resource Group

Steps to Create a Resource Group:

1. Go to Azure Portal > Resource Groups > Create.
2. Enter details:
 - Subscription: Select your subscription.
 - Resource Group Name: WebAppRG.
 - Region: Choose a region like East US.
3. Click Review + Create, then Create.



Azure Resource Manager (ARM) Templates Overview

What is an ARM Template?

- A JSON file defining your Azure infrastructure.
- Allows Infrastructure-as-Code (IaC) for consistent and repeatable deployments.

Key Benefits:

- Declarative syntax: Specify what you want to deploy.
- Version control for templates using tools like Git.



Example ARM Template (Snippet):

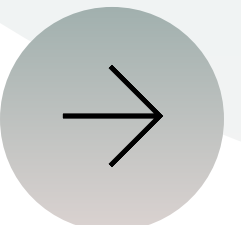
```
{
  "$schema":
    "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "resources": [
    {
      "type":
        "Microsoft.Compute/virtualMachines",
      "apiVersion": "2021-07-01",
      "name": "MyVM",
      "location": "East US",
      "properties": {
        "hardwareProfile": {
          "vmSize": "Standard_B2s"
        }
      }
    }
  ]
}
```



Deploying Resources with ARM Templates

Steps to Deploy an ARM Template:

1. Go to Azure Portal > Deploy a custom template.
2. Select Build your own template in the editor.
3. Paste the JSON template code.
4. Click Save, then provide deployment parameters like resource group and location.
5. Click Review + Create, then Create.



Step 1: Define the Objective

Objective: Create a Web App Infrastructure by deploying:

1. Resource Group: WebAppRG

- Region: East US

2. Resources:

- A Virtual Machine named WebServerVM.
- A Storage Account named WebAppStorage.



Step 2: Prepare the ARM Template

The ARM Template defines the VM and Storage Account in a single deployment.

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "type": "string",
      "defaultValue": "WebServerVM"
    },
    "storageAccountName": {
      "type": "string",
      "defaultValue": "webappstorage"
    },
    "location": {
      "type": "string",
      "defaultValue": "East US"
    }
  },
  "resources": [
    {
      "type": "Microsoft.Compute/virtualMachines",
      "apiVersion": "2021-07-01",
      "name": "[parameters('vmName')]",
      "location": "[parameters('location')]",
      "properties": {
        "hardwareProfile": {
          "vmSize": "Standard_B2s"
        },
        "storageProfile": {
          "osDisk": {
            "createOption": "FromImage"
          },
          "imageReference": {
            "publisher": "MicrosoftWindowsServer",
            "offer": "WindowsServer",
            "sku": "2019-Datacenter",
            "version": "latest"
          }
        },
        "osProfile": {
          "computerName": "[parameters('vmName')]",
          "adminUsername": "azureuser",
          "adminPassword": "Password123!"
        },
        "networkProfile": {
          "networkInterfaces": [
            {
              "id": "[resourceId('Microsoft.Network/networkInterfaces', 'WebServerNIC')]"
            }
          ]
        }
      }
    },
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2021-06-01",
      "name": "[parameters('storageAccountName')]",
      "location": "[parameters('location')]",
      "sku": {
        "name": "Standard_LRS"
      },
      "kind": "StorageV2"
    }
  ]
}
```



Step 3: Steps to Deploy

1. Create the Resource Group

Use Azure Portal, Azure CLI, or PowerShell to create the resource group.

Azure CLI Command:

```
az group create  
--name WebAppRG  
--location "East US"
```

2. Deploy Using Azure Portal

- Navigate to Azure Portal > Deploy a Custom Template.
- Select Build your own template in the editor.
- Paste the ARM Template code and save it.
- Enter parameters:
 - vmName: WebServerVM
 - storageAccountName: webappstorage
 - location: East US
- Click Review + Create > Create.



3. Deploy Using Azure CLI

Save the ARM Template to a file
e.g.,webapp_template.json.

CLI Commands: Create the Resource Group:

```
az group create  
--name WebAppRG  
--location "East US"
```

Deploy the Template:

```
az deployment group create  
--resource-group WebAppRG  
--template-file  
webapp_template.json
```



4. Deploy Using PowerShell

Save the ARM Template as webapp_template.json.

PowerShell Commands: Create the Resource Group:

```
New-AzResourceGroup  
-Name WebAppRG  
-Location "East US"
```

Deploy the Template:

```
New-AzResourceGroupDeployment  
-ResourceGroupName WebAppRG  
-TemplateFile "webapp_template.json"
```



step 4: Verify Deployment

Azure Portal:

Navigate to Resource Groups > WebAppRG.

Verify that WebServerVM and WebAppStorage are deployed.

Azure CLI Verification:

```
az resource list  
--resource-group WebAppRG
```

PowerShell Verification:

```
Get-AzResource  
-ResourceGroupName WebAppRG
```

