Role-Based Access Control (RBAC) in Azure

Author: Zayan Ahmed | **Estimated Reading time:** 6 mins

What is RBAC?

RBAC (Role-Based Access Control) is a way to manage who can do what in Azure. It helps you control access to resources so that only the right people can make changes. Instead of giving everyone full control, you assign roles based on what they need to do.

How RBAC Works in Azure

Azure uses **roles** to define what actions a person can take. Each role has a set of permissions. Some common roles are:

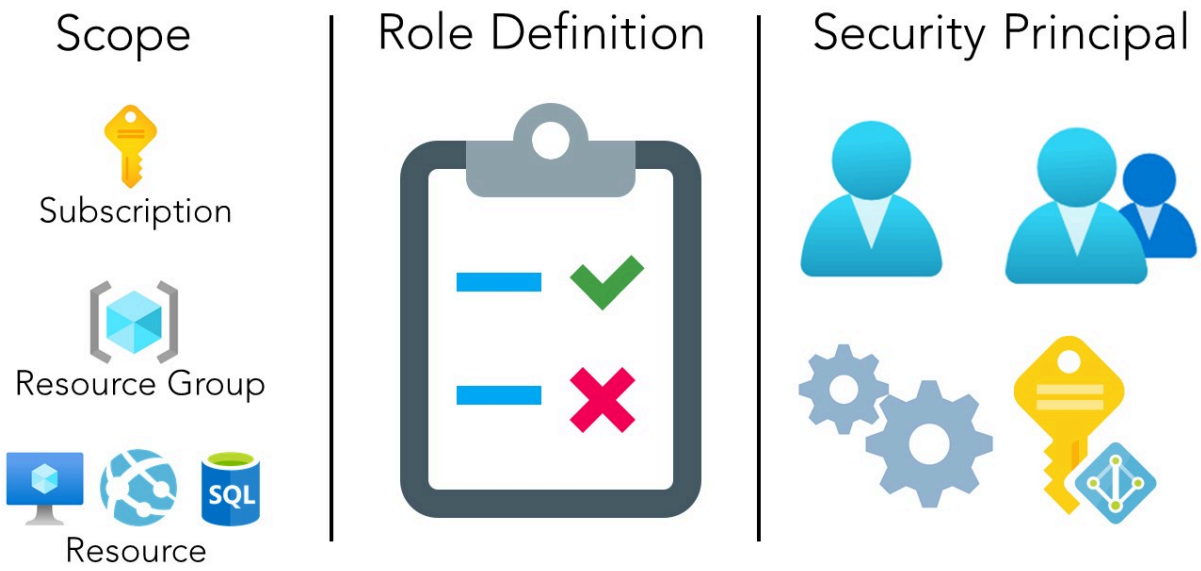
- **Owner**: Full control, can do anything.
- **Contributor**: Can create and change things but can't give others access.
- **Reader**: Can see everything but can't change anything.
- **User Access Administrator**: Can manage who has access but can’t change the resources.

RBAC Hierarchy in Azure

RBAC in Azure follows a structure, from broad to specific:

1. **Management Group** – The highest level, used to organize multiple subscriptions.
2. **Subscription** – Contains all the resources under one billing account.
3. **Resource Group** – A collection of related resources (like virtual machines, databases, and storage).
4. **Resource** – A single service, such as a virtual machine or database.

Permissions are inherited. If you give someone access at the **subscription level**, they get access to everything inside it. If you assign a role at the **resource level**, they can only control that one resource.



Scenarios & Solutions

1. Allowing a Developer to Manage Virtual Machines but Not Delete Them

- **Scenario**: You have a developer who needs to start, stop, and modify virtual machines but should not delete them.
- **Solution**: Assign the **Virtual Machine Contributor** role at the **resource group** level. This lets the developer manage VMs but prevents deletion.

2. Giving Read-Only Access to a Compliance Auditor

- **Scenario**: An auditor needs to review logs and settings but shouldn't be able to change anything.
- **Solution**: Assign the **Reader** role at the **subscription** level so they can view everything but not make changes.

3. Granting Database Access to a Data Analyst Without Giving Server Control

- **Scenario**: A data analyst needs access to a database to run reports but should not change the database server settings.

- **Solution**: Assign the **SQL Database Reader** role at the **database** level, not at the resource group level.

4. Allowing a DevOps Engineer to Manage Permissions Without Resource Control

- **Scenario**: A DevOps engineer needs to manage user permissions but should not create or delete resources.
- **Solution**: Assign the **User Access Administrator** role at the **resource group** level.

Common Mistakes & How to Avoid Them

- **Giving too much access**: Assign only the permissions needed. If someone only needs to read data, don't give them contributor access.
- **Assigning at too high a level**: If you give access at the subscription level, it applies to all resources. Use resource groups or specific resources for better control.
- **Not reviewing access regularly**: People change roles. Regularly check and remove unneeded permissions.

Summary

RBAC in Azure helps control who can do what by assigning **roles** at different **levels**. Always assign the least privilege necessary and use role hierarchy to avoid unnecessary permissions. By carefully using RBAC, you can keep your Azure environment secure and well-managed.

Want more ? 🤔
Follow me on **LinkedIn** 😃