Resource: A manageable item that's available through Azure. Virtual machines (VMs), storage accounts, web apps, databases, and virtual networks are examples of resources.

Resource group: A container that holds related resources for an Azure solution. The resource group includes resources that you want to manage as a group. You decide which resources belong in a resource group based on what makes the most sense for your organization.

**Azure resource groups**

Resource groups are a fundamental element of the Azure platform. A resource group is a logical container for resources deployed on Azure.

These resources are anything you create in an Azure subscription like VMs, Azure Application Gateway instances, and Azure Cosmos DB instances.

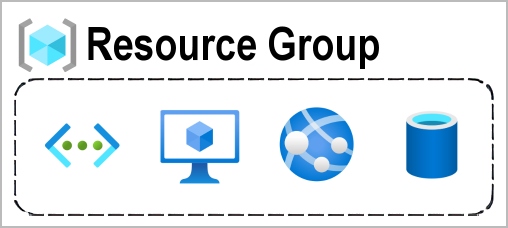
All resources must be in a resource group, and a resource can only be a member of a single resource group.

Many resources can be moved between resource groups with some services having specific limitations or requirements to move.

Resource groups can't be nested. Before any resource can be provisioned, you need a resource group for it to be placed in.

### Logical grouping

Resource groups exist to help manage and organize your Azure resources. By placing resources of similar usage, type, or location in a resource group, you can provide order and organization to resources you create in Azure



### Life cycle

If you delete a resource group, all resources contained within it are also deleted. Resource groups make it easy to remove a set of resources all at once.

### Authorization

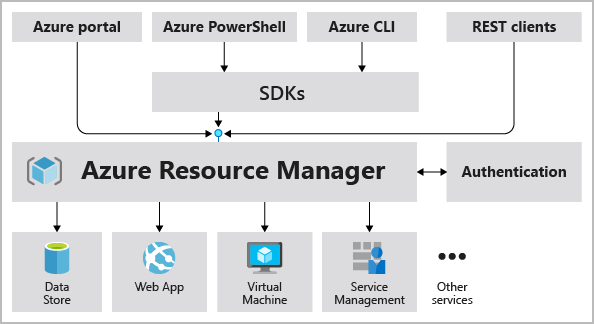
Resource groups are also a scope for applying role-based access control (RBAC) permissions. By applying RBAC permissions to a resource group, you can ease administration and limit access to allow only what's needed.

## Azure Resource Manager

Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account. You use management features like access control, locks, and tags to secure and organize your resources after deployment.

When a user sends a request from any of the Azure tools, APIs, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request. Resource Manager sends the request to the Azure service, which takes the requested action. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools.

The following image shows the role Resource Manager plays in handling Azure requests.



All capabilities that are available in the Azure portal are also available through PowerShell, the Azure CLI, REST APIs, and client SDKs. Functionality initially released through APIs will be represented in the portal within 180 days of initial release.

### The benefits of using Resource Manager

With Resource Manager, you can:

* Manage your infrastructure through declarative templates rather than scripts. A Resource Manager template is a JSON file that defines what you want to deploy to Azure.
* Deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually.
* Redeploy your solution throughout the development life cycle and have confidence your resources are deployed in a consistent state.
* Define the dependencies between resources so they're deployed in the correct order.
* Apply access control to all services because RBAC is natively integrated into the management platform.
* Apply tags to resources to logically organize all the resources in your subscription.
* Clarify your organization's billing by viewing costs for a group of resources that share the same tag.