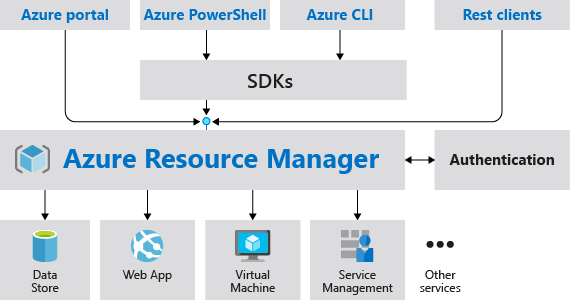
**Azure Resource Manager**

The following image shows how all the tools interact with the same Azure Resource Manager API



## **Resource providers**

Each resource provider offers a set of resources and operations for working with an Azure service. For example, if you want to store keys and secrets, you work with the **Microsoft.KeyVault** resource provider. This resource provider offers a resource type called vaults for creating the key vault.

The name of a resource type is in the format: **{resource-provider}/{resource-type}**. For example, the key vault type is **Microsoft.KeyVault/vaults**.

* To prevent resources from getting deleted, You can associate the lock with a subscription, resource group, or resource.
* Owner and User Access Administrator roles can create or delete management locks.

# **Determine resource limits**

https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits?toc=%2Fazure%2Fnetworking%2Ftoc.json

# **Configure resources with Azure Resource Manager templates**

https://docs.microsoft.com/en-us/learn/modules/configure-resources-arm-templates/

* **Templates are linkable**. You can link Resource Manager templates together to make the templates themselves modular. You can write small templates that each define a piece of a solution, and then combine them to create a complete system.
* Azure Resource Manager templates are written in JSON, which allows you to express data stored as an object (such as a virtual machine) in text. A JSON document is essentially a collection of key-value pairs. Each key is a string, whose value can be:
  + - A string
    - A number
    - A Boolean expression
    - A list of values
    - An object (which is a collection of other key-value pairs)

{

"$schema": "http://schema.management.​azure.com/schemas/2019-04-01/deploymentTemplate.json#",​

"contentVersion": "",​

"parameters": {},​

"variables": {},​

"functions": [],​

"resources": [],​

"outputs": {}​

}

* You're limited to 256 parameters in a template. You can reduce the number of parameters by using objects that contain multiple properties.
* [Azure Bicep](https://docs.microsoft.com/en-us/azure/azure-resource-manager/bicep/overview) is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources. It provides concise syntax, reliable type safety, and support for code reuse.
* Bicep is a transparent abstraction over ARM template JSON and doesn't lose any of the JSON template capabilities.
  + Exercise - Create and deploy an Azure Resource Manager template :- https://docs.microsoft.com/en-us/learn/modules/build-azure-vm-templates/

**Azure Powershell**

* Cmdlets follow a verb-noun naming convention; for example, Get-Process, Format-Table, and Start-Service. There is also a convention for verb choice: "get" to retrieve data, "set" to insert or update data, "format" to format data, "out" to direct output to a destination, and so on
* **What is a PowerShell module**?:- Cmdlets are shipped in modules.
* Az is the formal name for the Azure PowerShell module, which contains cmdlets to work with Azure features. It contains hundreds of cmdlets that let you control nearly every aspect of every Azure resource.
* Creating resource with powershell:- https://docs.microsoft.com/en-us/learn/modules/automate-azure-tasks-with-powershell/5-create-resource-interactively?pivots=linux
* exersise:- <https://docs.microsoft.com/en-us/learn/modules/automate-azure-tasks-with-powershell/6-exercise-create-resource-interactively>

## **What is a PowerShell script?**

* A PowerShell script is a text file containing commands and control constructs. The commands are invocations of cmdlets. The control constructs are programming features like loops, variables, parameters, comments, etc., supplied by PowerShell.
* PowerShell script files have a **.ps1** file extension. You can create and save these files with any text editor.
* **Excersize using powershell script. :- https://docs.microsoft.com/en-us/learn/modules/automate-azure-tasks-with-powershell/8-exercise-create-resource-using-script**