😊 Azure PowerShell

Azure PowerShell is a command-line tool designed specifically for managing resources within Microsoft Azure, Microsoft's cloud computing platform. It provides a set of cmdlets (commandlets) that allow users to interact with Azure services and resources directly from the command line or through scripts.

With Azure PowerShell, users can perform various tasks such as provisioning virtual machines, managing storage accounts, configuring networking settings, deploying web applications, and more through PowerShell commands.

Azure PowerShell is built on top of the PowerShell framework, leveraging its scripting capabilities and syntax. It offers a convenient and efficient way to automate Azure management tasks and integrate Azure operations into existing workflows and scripts.

😄 Use cases of Azure PowerShell:

Azure PowerShell can be used in various scenarios and use cases for managing resources and automating tasks within the Azure cloud platform. Here are some common use cases:

1. **Resource Provisioning and Configuration:** Azure PowerShell allows you to create, manage, and configure Azure resources such as virtual machines, databases, storage accounts, networking components, and more. You can use it to automate the provisioning of resources according to your specific requirements.
2. **Deployment Automation:** You can automate the deployment of applications and services in Azure using PowerShell scripts. This includes deploying web applications, APIs, virtual machines, containers, and other types of resources required for your application stack.
3. **Infrastructure as Code (IaC):** Azure PowerShell can be used to implement Infrastructure as Code practices, where infrastructure configurations are defined and managed through code. You can use PowerShell scripts to define Azure resource configurations, making it easier to maintain, version, and replicate infrastructure setups.
4. **Configuration Management:** Azure PowerShell enables you to manage the configuration of Azure resources at scale. You can use it to enforce configuration policies, apply settings, and perform configuration drift remediation across multiple resources and environments.
5. **Monitoring and Reporting:** PowerShell scripts can be used to retrieve monitoring data and generate reports on the performance, usage, and health of Azure resources. You can automate tasks such as fetching metrics, analyzing logs, and generating custom reports tailored to your organization's needs.
6. **Security and Compliance:** Azure PowerShell provides cmdlets for managing security-related tasks such as access control, identity management, encryption,

and compliance auditing. You can use it to automate security configurations, enforce compliance policies, and respond to security incidents.

7. **Integration with DevOps Pipelines:** Azure PowerShell can be integrated into DevOps pipelines for continuous integration and continuous deployment (CI/CD). You can use it to automate the build, test, and deployment processes of Azure-based applications, enabling rapid and reliable delivery of software updates.
8. **Cost Management:** PowerShell scripts can help you manage and optimize costs in Azure by automating tasks such as resource tagging, budget monitoring, and cost analysis. You can use it to implement cost-saving strategies and optimize resource utilization to reduce expenses.

😄 To begin with the Lab:

1. First, you have to download the latest version PowerShell in your windows laptop or desktop.
2. For that you can visit the official Microsoft website mentioned below.

   https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.2

3. Now open your PowerShell and install Azure PowerShell on your local machine.
4. For that you need to visit the website mentioned below and copy the script to install Azure PowerShell.

   https://learn.microsoft.com/en-us/powershell/azure/install-azps-windows?view=azps-11.6.0&tabs=powershell&pivots=windows-psgallery

PowerShell 7 | Windows PowerShell

- Install a supported version of PowerShell version 7 or higher

- Set the PowerShell execution policy to remote signed or less restrictive
  - Check the PowerShell execution policy:

```
PowerShell                                                    Copy
Get-ExecutionPolicy -List
```

  - Set the PowerShell execution policy to remote signed:

```
PowerShell                                                    Copy
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

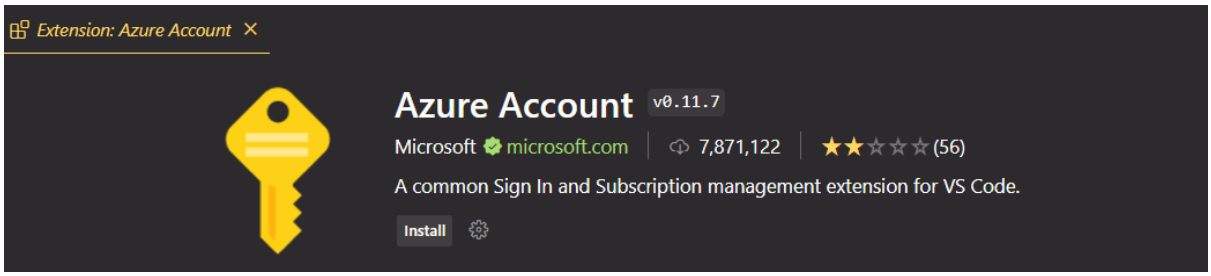
For more information about execution policies, see about_Execution_Policies.

Installation

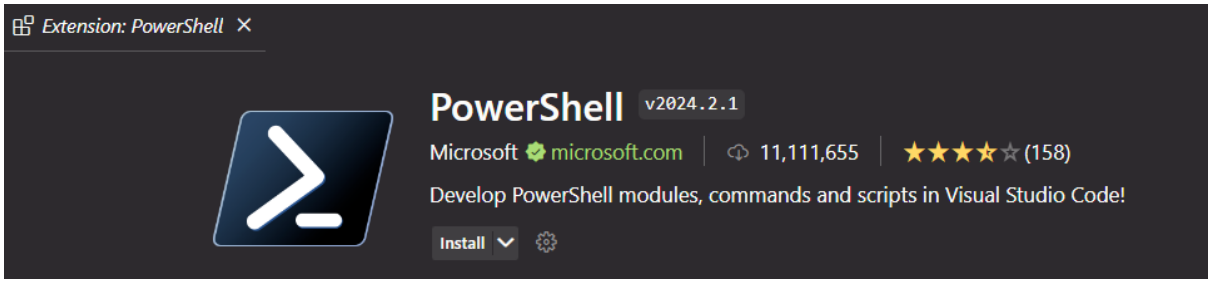
Use the Install-Module cmdlet to install the Az PowerShell module:

```
PowerShell                                                    Copy
Install-Module -Name Az -Repository PSGallery -Force
```

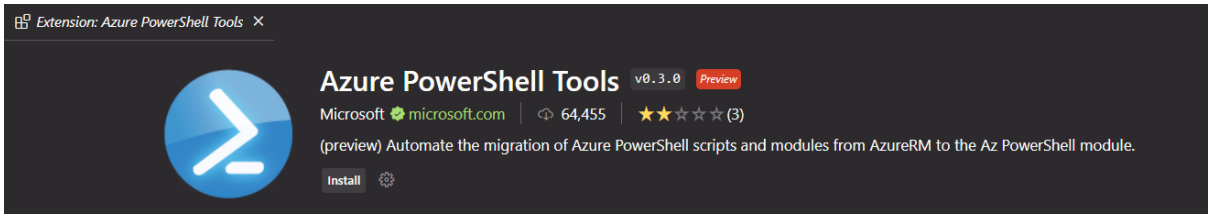
5. After that you have to download VS Code on your local machine. Then in VS code extensions, you have to install an Azure account.



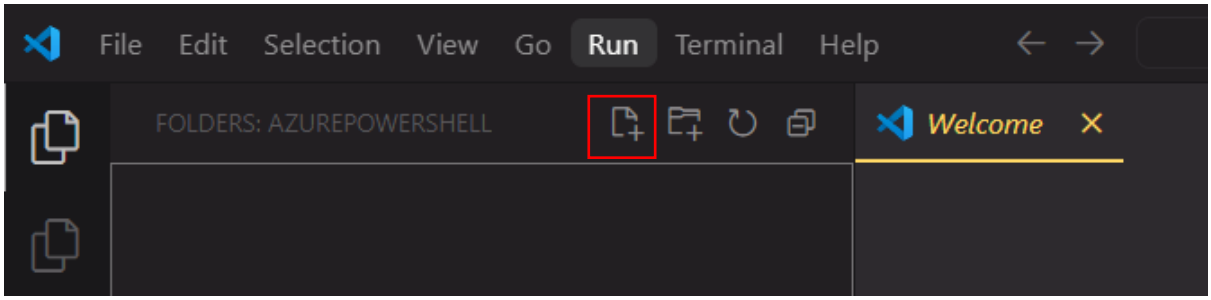
6. Then you have to install PowerShell.



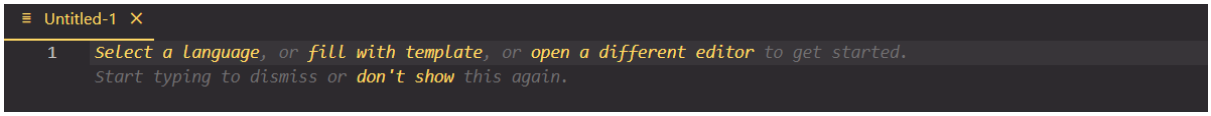
7. Now search for Azure PowerShell tools and install them.



8. Now you need to create a folder in your local machine in any of your drives then right click on that folder and open it with VS Code.
9. Then you will see your folder in VS Code and click on the highlighted icon to create a new file.



10. Now from your keyboard press ctrl + N to create a new text file. Then you have to choose a language which will be PowerShell.



11. Then save this file in your new folder whatever you might have named it.
12. Now in your VS code open your Terminal and type this command to connect your Azure account.

Connect-AzAccount