



Azure Automation

1. **Azure Automation for Azure VMs with PowerShell**
2. **Schedule and Run PowerShell Scripts for Azure VMs using Azure Automation**

Table of Contents

1. Introduction
2. Overview of Azure Automation
3. Setting Up Azure Automation
4. Creating and Managing Runbooks
 1. Types of Runbooks
 2. Creating a PowerShell Runbook
 3. Publishing and Scheduling Runbooks
 - Creating a Schedule
 - Linking the Schedule to a Runbook
5. Monitoring and Managing Automation Jobs
6. Best Practices for Azure Automation
7. Conclusion
8. References

1. Introduction

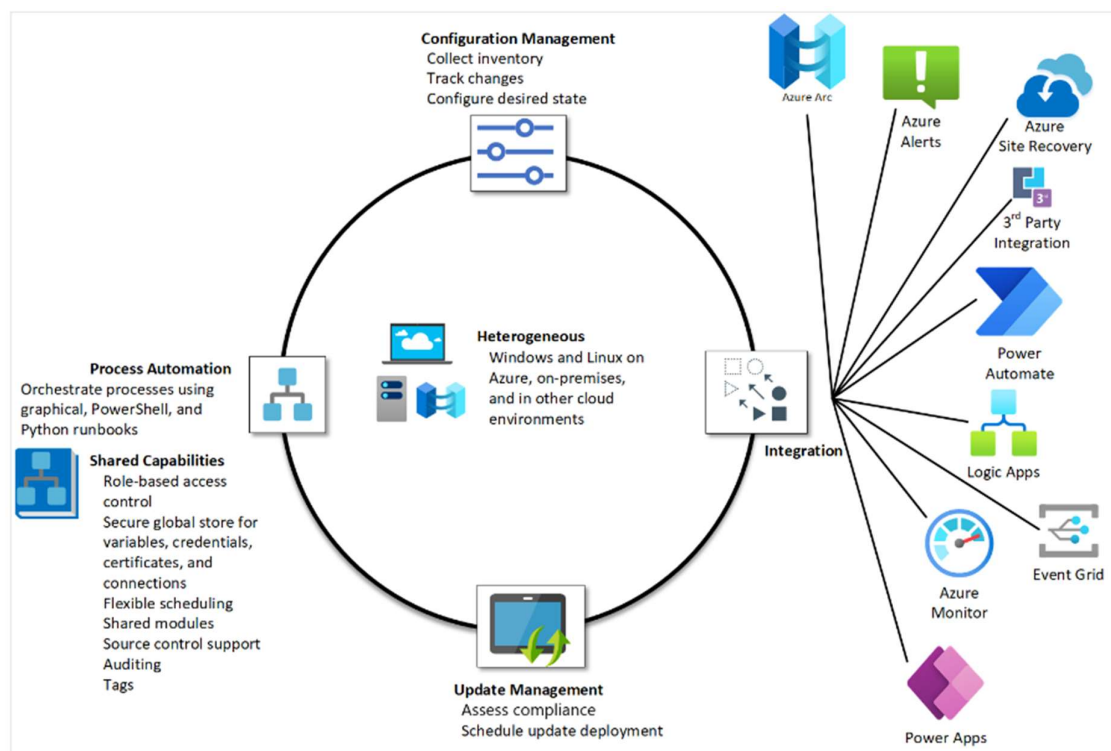
Azure Automation is a powerful service that allows you to automate repetitive tasks, manage cloud infrastructure efficiently, and reduce operational overhead. This document provides a comprehensive guide on using Azure Automation for managing Azure Virtual Machines (VMs) with PowerShell. It covers the setup process, creation and management of runbooks, common automation tasks, and best practices.

2. Overview of Azure Automation

Azure Automation is a cloud-based automation and configuration service that provides a way to automate the management and deployment of resources in your Azure environment. It supports various scripting languages, including PowerShell, Python, and graphical runbooks.

Key Features:

- **Runbooks:** Automate repetitive tasks.
- **Configurations:** Manage and maintain the desired state of your resources.
- **Schedules:** Trigger automation tasks at specific times.
- **Job Monitoring:** Track the status and results of automation jobs.



There are several Azure services that can deliver the above requirements, where each service includes a set of capabilities and serves a role as a programmable platform to build cloud solutions. For example, Azure Bicep and Resource Manager provide a language to develop repeatable and consistent deployment templates for Azure resources. Azure Automation can

process that template to deploy an Azure resource and then process a set of post-deployment configuration tasks.

Automation gives you complete control during deployment, operations, and decommissioning of enterprise workloads and resources.

2.1 Process Automation

Azure Automation's Process Automation allows you to streamline repetitive, time-consuming, and error-prone management tasks, helping you focus on work that adds significant business value. By minimizing errors and enhancing efficiency, it also helps reduce operational costs. Detailed information about the process automation operating environment can be found in the Runbook execution in Azure Automation documentation.

Process automation supports integration with Azure services and third-party systems needed for deploying, configuring, and managing end-to-end processes. You can create graphical, PowerShell, and Python runbooks to run directly on Windows or Linux machines, or against resources in on-premises or other cloud environments using a Hybrid Runbook Worker.

Webhooks enable continuous delivery and operations by triggering automation from Azure Logic Apps, Azure Function, ITSM products or services, DevOps tools, and monitoring systems.

2.2 Configuration Management

Azure Automation's Configuration Management is supported by two main capabilities:

1. Change Tracking and Inventory

Change Tracking and Inventory enable you to monitor changes in your Linux and Windows virtual machines and server infrastructure. The service tracks changes across services, daemons, software, registry, and files, helping you diagnose unwanted changes and raise alerts. It also provides inventory capabilities to query in-guest resources, giving visibility into installed applications and configuration items. The feature is now supported with the Azure Monitoring Agent.

2. Azure Automation State Configuration

Azure Automation State Configuration is a cloud-based service for PowerShell Desired State Configuration (DSC), allowing you to manage DSC resources and apply configurations to virtual or physical machines from a DSC pull server in the Azure cloud.

2.3 Update Management

Azure Automation's Update Management feature provides visibility into update compliance across Windows and Linux systems in hybrid environments. It allows you to schedule deployments and orchestrate the installation of updates within a defined maintenance window. You can also exclude specific updates from being installed on particular machines.

2.4 Shared Capabilities

Azure Automation offers several shared capabilities, including:

- **Schedules:** Trigger automation operations at predefined times.
- **Modules:** Manage Azure and other systems with imported modules for Microsoft, third-party, community, and custom-defined cmdlets and DSC resources.
- **Modules Gallery:** Integrate with the PowerShell Gallery to view and import runbooks into the Automation account, facilitating quick start with PowerShell gallery and Microsoft Script Center.
- **Python 2 and 3 Packages:** Support for Python 2 and 3 runbooks in your Automation account.
- **Credentials:** Securely store sensitive information used by runbooks and configurations at runtime.
- **Connections:** Store name-value pairs for system connections, defined by module authors for use at runtime.
- **Certificates:** Define information for authentication and securing deployed resources accessed by runbooks or DSC configurations at runtime.
- **Variables:** Store content for use across runbooks and configurations, allowing value changes without modifying the runbooks or configurations referencing them.

2.5 Role-Based Access Control

Azure Automation supports Azure role-based access control (Azure RBAC) to manage access to the Automation account and its resources. For more information on configuring Azure RBAC for your Automation account, runbooks, and jobs, refer to the Role-based access control for Azure Automation documentation.

2.6 Source Control Integration

Azure Automation supports source control integration, promoting configuration as code by allowing runbooks or configurations to be checked into a source control system.

2.7 Heterogeneous Support (Windows and Linux)

Azure Automation works across Windows and Linux physical servers and virtual machines, whether on your corporate network, in Azure, or with other cloud providers. It offers a consistent method to automate and configure deployed workloads and operating systems. The Hybrid Runbook Worker feature enables running runbooks directly on non-Azure physical servers or virtual machines, managing local resources in the environment.

With Arc-enabled servers, Azure Automation provides a consistent deployment and management experience for non-Azure machines, integrating with the Automation service using the VM extension framework to deploy the Hybrid Runbook Worker role, and simplifying onboarding to Update Management and Change Tracking and Inventory.

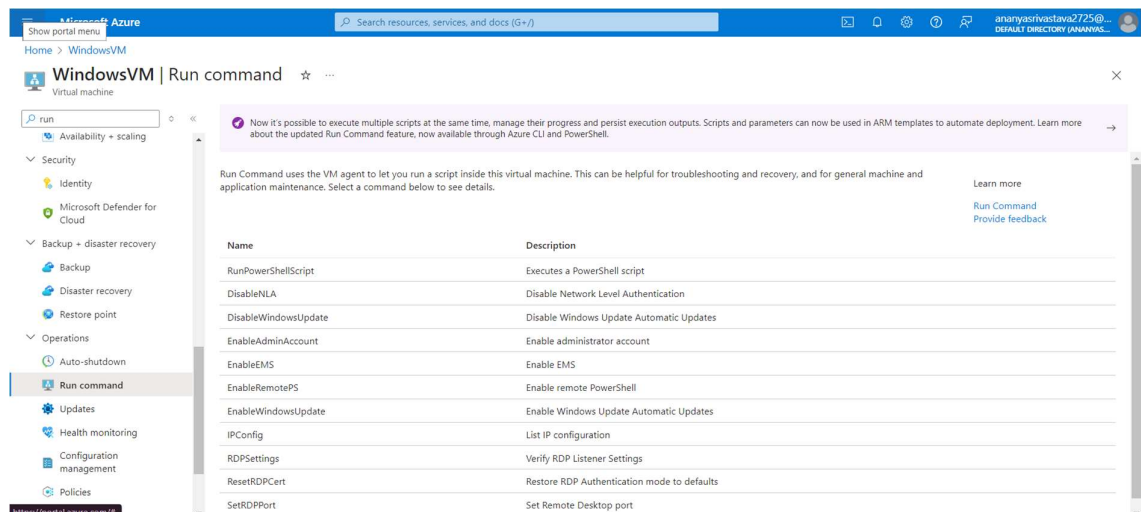
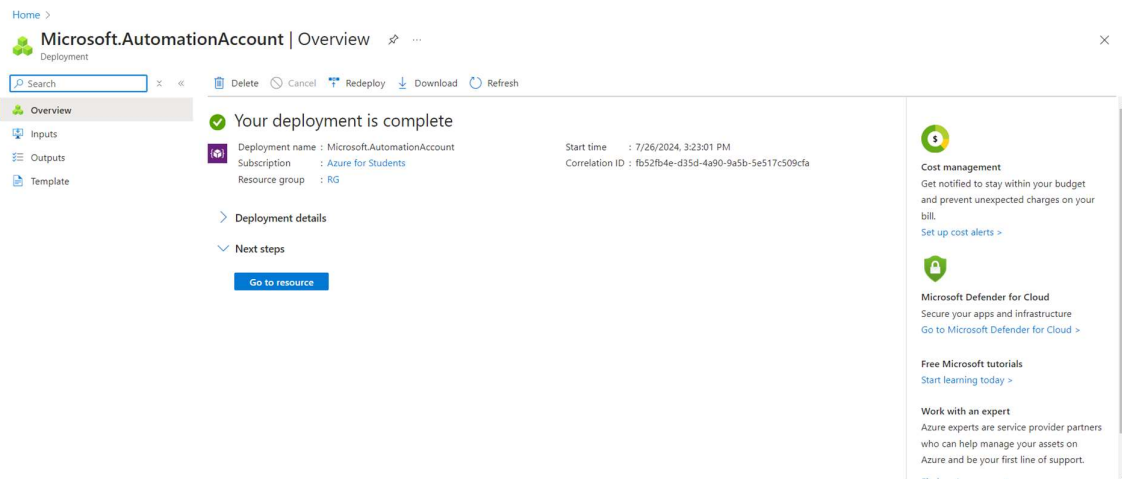


3. Setting Up Azure Automation

To begin using Azure Automation, follow these steps:

1. Create an Automation Account:

- Navigate to the Azure portal.
- Click on "Create a resource" and search for "Automation."
- Select "Automation" and click "Create."
- Fill in the required details and create the Automation account.



Run Command Script

IPConfig

 Script execution complete

Output

```
Windows IP Configuration

Host Name . . . . . : WindowsVM
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : hppo02tbtqcqe3oivh5iatyaqha.px.internal.cloudapp.net

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : hppo02tbtqcqe3oivh5iatyaqha.px.internal.cloudapp.net
Description . . . . . : Microsoft Hyper-V Network Adapter
Physical Address. . . . . : 00-0D-3A-CC-1B-7A
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::27f:2c09:2182:9cfb%6(Preferred)
IPv4 Address. . . . . : 10.0.0.4(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Thursday, July 25, 2024 11:06:23 PM
Lease Expires . . . . . : Monday, September 1, 2160 4:01:19 PM
Default Gateway . . . . . : 10.0.0.1
DHCP Server . . . . . : 168.63.129.16
DHCPv6 IAID . . . . . : 100666682
DHCPv6 Client DUID. . . . . : 00-01-00-01-2E-34-92-C7-00-0D-3A-CC-1B-7A
```

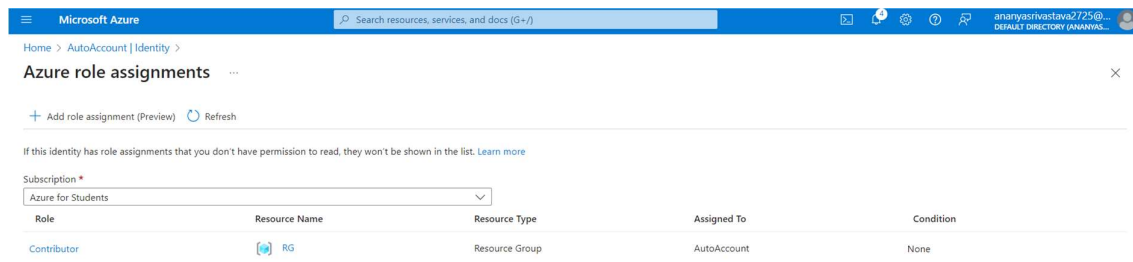
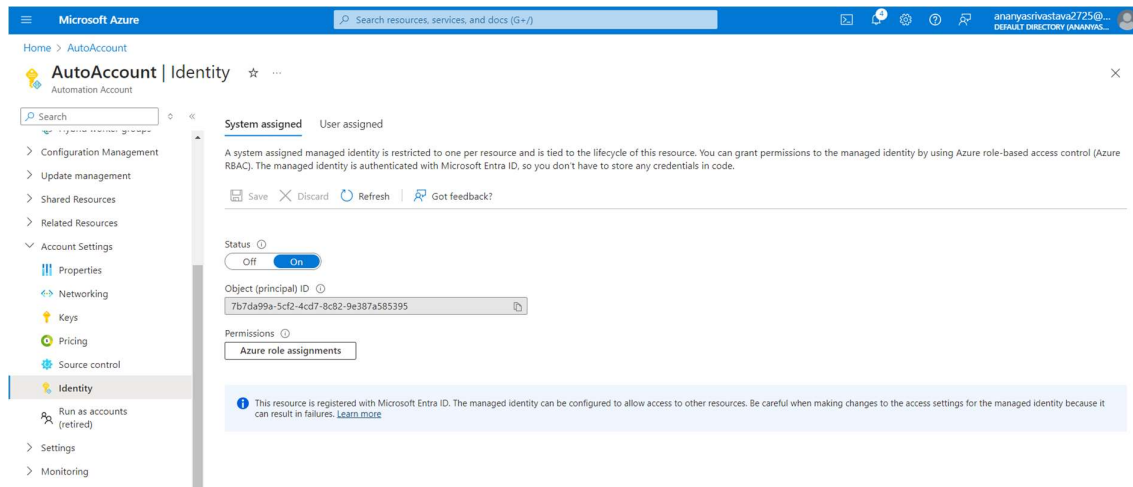
PS /home/ananya> Get-AzVM

ResourceGroupName	Name	Location	VmSize	OsType	NIC	ProvisioningState	Zone
RG	LinuxVM	australiaeast	Standard_D2s_v3	Linux	linuxvm297_z1	Succeeded	1
RG	WindowsVM	australiaeast	Standard_D2s_v3	Windows	windowsvm785_z1	Succeeded	1

2. Configure Permissions:

- Ensure that the Automation account has the necessary permissions to manage your Azure resources. This can be done by assigning the appropriate roles to the Automation account's managed identity

3. Set up a system assigned managed identity, role assignment and grant permissions



4. Creating and Managing Runbooks

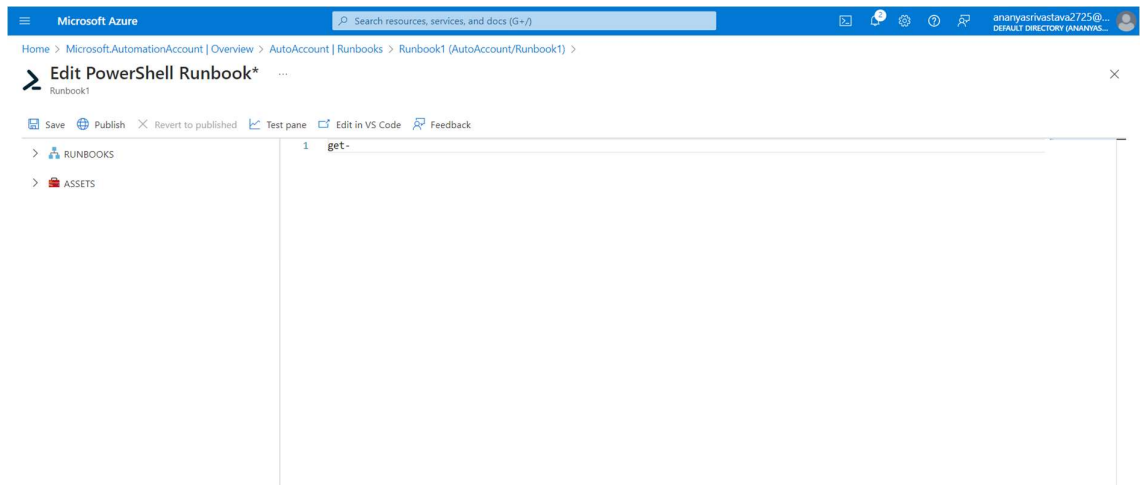
Runbooks are the core components of Azure Automation. They contain the logic and commands to perform specific tasks.

4.1 Types of Runbooks

- **PowerShell Runbooks:** Use PowerShell scripts to automate tasks.
- **Graphical Runbooks:** Use a drag-and-drop interface to create workflows.
- **Python Runbooks:** Use Python scripts for automation.

4.2 Creating a PowerShell Runbook

1. Navigate to the Automation account.
2. Under "Process Automation," select "Runbooks."
3. Click "Add a runbook" and choose "Create a new runbook."
4. Enter a name, select "PowerShell" as the runbook type, and create the runbook.
5. Write your PowerShell script in the editor and save the runbook.



4.3 Run Azure Automation Runbook PowerShell Script against Azure VMs

Resource group : [RG](#)

Account : AutoAccount

Location : East US

Subscription : [Azure for Students](#)

Subscription ID : 84e9a260-5ff7-4a17-b61f-33c673272cd4

Status : Published

Runbook type : Graphical PowerShell

Runtime version : 5.1

Last modified : 7/26/2024, 3:23 PM

Status	Created	Last updated
✓ Completed	7/26/2024, 3:43:38 PM	7/26/2024, 3:44:09 PM

Input **Output** Errors Warnings All Logs Exception

```
Connecting to azure via Connect-AzAccount -Identity
```

Input Output Errors Warnings All Logs Exception

```
Connecting to azure via Connect-AzAccount -Identity

Account      SubscriptionName      TenantId      Environment
-----
MSI@50342    ms-cxa-thmaure-demo-test 72f988bf-86f1-41af-91ab-2d7cd011db47 AzureC...
Successfully connected with Automation account's Managed Identity
The following VMs are running and are running Windows:
ttazurewinvm01
Run Script Against Machines

Value      : {Microsoft.Azure.Management.Compute.Models.InstanceViewStatus,
Microsoft.Azure.Management.Compute.Models.InstanceViewStatus}
Name      :
```

4.4 Publishing and Scheduling Runbooks

- **Publish:** After creating and testing the runbook, click "Publish" to make it available for use.
- **Schedule:** To automate the execution of the runbook, create a schedule and link it to the runbook.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information. The main content area is titled 'Schedules' and shows a list of schedules with columns for Name, Next run, and Time zone. A 'New Schedule' form is open on the right side, allowing users to create a new schedule. The form includes fields for Name, Description, Start date and time, Time zone, Recurrence (Once or Recurring), Recur every (1 or more), and On these days (Monday, Tuesday, Wednesday).

Microsoft Azure

Home > AutoAccount | Runbooks > AzureAutomationTutorialWithIdentityGraphical (AutoAccount/AzureAutomationTutorialWithIdentityGraphical) > Schedule Runbook >

Schedules

+ Add a schedule

Name	Next run	Time zone
No schedules found.		

New Schedule

Name * Friday

Description

Starts * 07/26/2024 4:14 PM

Time zone

Recurrence

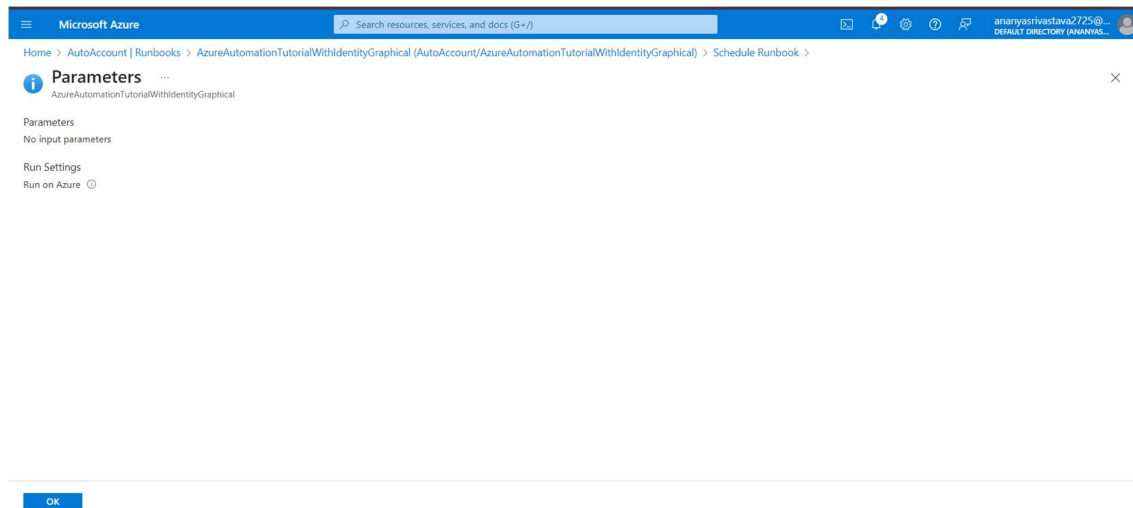
Once Recurring

Recur every * 1 Week

On these days

Monday Tuesday Wednesday

Create



5. Monitoring and Managing Automation Jobs

Azure Automation provides tools for monitoring and managing jobs:

- **Job Status:** Check the status of runbook jobs.
- **Job Output:** View the output and logs of completed jobs.
- **Alerts:** Configure alerts to notify you of job failures or other issues.

6. Best Practices for Azure Automation

- **Modular Runbooks:** Create reusable, modular runbooks to simplify maintenance.
- **Error Handling:** Implement error handling and logging in runbooks.
- **Testing:** Thoroughly test runbooks in a development environment before deploying them to production.
- **Security:** Secure credentials and sensitive data using Azure Key Vault.

7. Conclusion

Azure Automation with PowerShell provides a robust and flexible way to manage Azure VMs, enabling you to automate routine tasks, improve efficiency, and reduce the risk of human error. By following best practices and leveraging the powerful features of Azure Automation, you can streamline your cloud operations and enhance your overall cloud management strategy.

8. References

- Azure Automation Documentation - <https://www.thomasmaurer.ch/2022/08/schedule-and-run-powershell-scripts-for-azure-vms-using-azure-automation/>
- PowerShell Documentation - <https://learn.microsoft.com/en-us/powershell/azure/new-azureps-module-az?view=azps-12.1.0>
- Azure VM Documentation - <https://learn.microsoft.com/en-us/azure/virtual-machines/extensions/custom-script-windows>
- Youtube - <https://www.youtube.com/watch?v=rgOyxd-RhVM>