



IST 651 Scripting for Ent Data Sys

CLOUD RESOURCE MONITORING AND ALERTING USING SCRIPTING

Lokasani Sailaja
Supraja Ramachandran Nair

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES --  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```



Introduction:

- The cloud resource monitoring and the altering project aims to develop a system that can help organizations monitor their cloud resources and automate resource management tasks. This project will use Python as the primary scripting language to create a system that can monitor the cloud resources and automatically alter them according to the needs of the organization.

Overview:

- Cloud computing is becoming increasingly popular among organizations and businesses.
- It provides the flexibility to scale up or down the resources according to the needs of the organization.
- As the number of cloud resources grows, monitoring and managing these resources become a challenging task.
- The cloud resource monitoring and altering project aim to provide a solution to monitor the cloud resources and automate the resource management tasks using scripting (Python).



Goals & Objectives:

- To create a system that can monitor the cloud resources.
- To develop Python scripts to automate resource management tasks.
- To integrate the monitoring and resource management script into a single system.
- The goal of this script is to monitor the cloud resources and perform actions based on the usage levels.
- To demonstrate how the Azure SDK and psutil library can be used to obtain resource usage information and perform actions based on that information.
- To develop a system that can generate alerts in case of any issues with the cloud resources.
- To test and validate the system's performance and functionality.



Positive Outcomes:

There are several advantages and benefits of implementing cloud resource monitoring and altering using scripting:

- **Cost optimization:** By monitoring resource usage and scaling resources up or down as needed, organizations can save money by only using the resources they need at any given time.
- **Improved performance and availability:** Scaling resources as needed can help ensure that applications and services are always available and perform optimally, even during periods of high demand.
- **Automation:** Automating the process of monitoring and altering resources eliminates the need for manual intervention, which saves time and reduces the risk of human error.

Positive Outcomes:

There are several advantages and benefits of implementing cloud resource monitoring and altering using scripting:

- **Flexibility**: Cloud resource monitoring and altering can be customized to meet the specific needs of an organization, allowing for greater flexibility and agility.
- **Better decision-making**: By having real-time visibility into resource usage and performance, organizations can make data-driven decisions about resource allocation and management, which can lead to better outcomes and more efficient resource usage.

Tools & Techniques:

1. **Azure SDK:** This script uses the Azure SDK for Python to interact with Azure resources, such as virtual machines, networks, and resources. It includes modules for authentication, management clients, and resource classes.
2. **psutil:** This script uses the psutil module to get system information, such as CPU utilization, memory usage, and disk usage.
3. **smtplib:** This script uses the smtplib module to send emails when certain conditions are met, such as high memory usage.
4. **DefaultAzureCredential and AzureCliCredential:** These are authentication classes provided by the Azure SDK for Python to authenticate and authorize access to Azure resources.
5. **Conditional statements:** This script uses if-else statements to determine whether certain conditions are met before taking certain actions, such as scaling up a virtual machine or sending an email.
6. **Azure PowerShell:** This script may require Azure PowerShell to be installed and configured on the system, as it is used to initially authenticate and authorize access to Azure resources using AzureCliCredential.

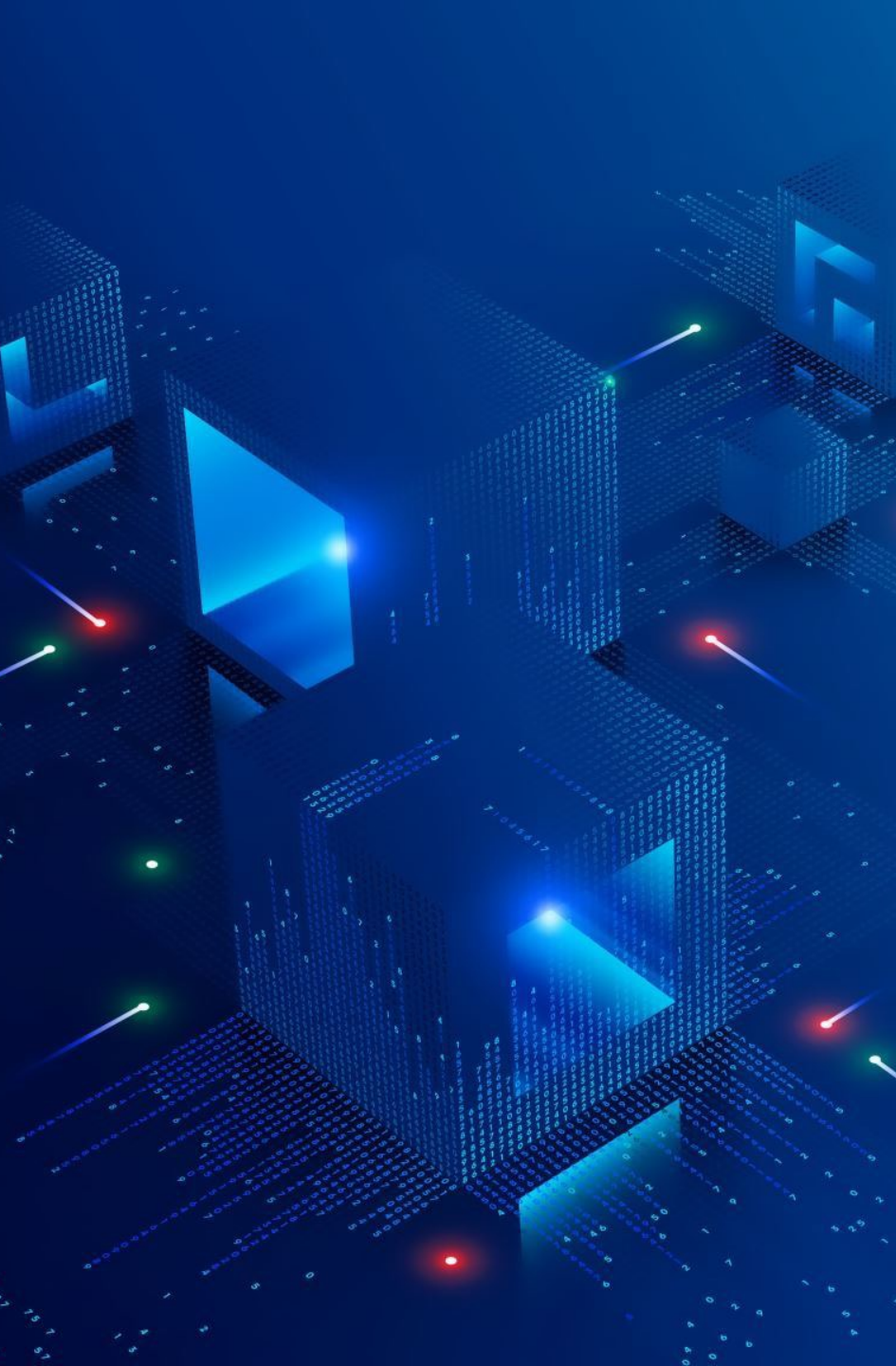


Implementation

This project script checks the following metrics:

- **CPU utilization** - The script monitors the CPU utilization of the virtual machine. If the CPU utilization is greater than 5%, the script scales up the virtual machine to a larger size.
- **Memory usage** - The script monitors the memory usage of the virtual machine. If the memory usage exceeds 15%, the script sends an email to notify the user.
- **Disk utilization** - The script monitors the disk utilization of the virtual machine. If the disk utilization is greater than 10%, the script shuts down the virtual machine.

By monitoring these metrics, the script can help ensure that the virtual machine is running efficiently and prevent any potential issues that may arise from exceeding the available resources.



Implementation:

1. Import the necessary modules and libraries
2. Set up Azure credentials and subscription ID
3. Initialize Azure clients
4. Get virtual machine details
5. Get virtual machine network details
6. Get virtual machine resource details
7. Calculate CPU utilization
8. Rescale the virtual machine



Implementation:

9. Calculate memory utilization
10. Send an email if memory utilization exceeds 15%
11. Calculate disk utilization
12. Power off the virtual machine if disk utilization exceeds 10%
13. Print the virtual machine details

A top-down view of a collaborative workspace. Several people are visible, some with their heads down, others with hands on the table. The table is covered with hand-drawn diagrams and sketches. A large yellow circle in the center contains a lightbulb with three lightning bolts above it. To the left, a red circle contains an open book. To the right, a blue circle contains an eye. There are also gears, puzzle pieces, a magnifying glass, and various colored circles and rectangles. Dashed lines connect different parts of the diagrams, suggesting a flow or process. The overall atmosphere is one of creative collaboration and problem-solving.

Implementation:

DEMO: https://video.syr.edu/media/t/1_44p23lfm



Issues Encountered:

- Installing certain libraries was a challenge since some of the latest versions did not support what we wanted to do.
- Setting up Email notifications after fetching CPU utilization.
- Fetching CPU Utilization.



Conclusion:

- The implementation of the cloud resource monitoring and scaling script using Python and Azure SDK can greatly help in automating the scaling process for cloud resources based on their performance metrics. This can greatly improve resource utilization and reduce the need for manual intervention. The use of Python and Azure SDK in this project makes the implementation easy and scalable.

Thank You!

