Azure Quota Checker Script

Table of Contents

- Prerequisites
- Usage Instructions
- What the Script Does
- Troubleshooting
- Notes

Prerequisites

Before running this script, ensure that the following conditions are met:

Azure CLI is available. Azure Cloud Shell already has Azure CLI installed.

Logged into Azure. Azure Cloud Shell is automatically logged in.

Usage Instructions

This script is intended to be uploaded and executed in Azure Cloud Shell. Please follow these steps:

1. Obtain the Script

- Upload the script file from your local machine to Azure Cloud Shell.
    - Click the "Upload/Download" button at the top of the Azure Cloud Shell window and select "Upload".
    - Choose the file check_azure_quota.ps1 to upload.

2. Run the Script

- In Azure Cloud Shell, execute the following command:

```
./check_quota.ps1
```

3. Enter the Azure Region

- When prompted, enter the Azure region you want to check (e.g., japaneast).
- You can check the region names using the following command:

```
az account list-locations -o table
```

Refer to the Name column.

```
DisplayName               Name                 RegionalDisplayName
------------------------  -------------------  ----------------------------------
--
```

```
East US                 eastus              (US) East US
Japan East              japaneast           (Asia Pacific) Japan East
West Europe             westeurope          (Europe) West Europe
```

What the Script Does

This PowerShell script checks the available quotas for specific Azure VM instance types in a specified region.

Main Features:

1. Error Handling
   - If any command fails, the script displays an error message and exits.
2. Azure Region Input
   - Prompts the user to input the Azure region they wish to check.
3. Login Verification
   - Checks if Azure CLI is logged in. Azure Cloud Shell is already logged in, but if not, it prompts the user to log in.
4. Quota Filtering
   - Retrieves and displays VM usage filtered by specific quota names (Standard, HB, HC, etc.).

PowerShell Script

```powershell
# Define a function for error handling
function Stop-Error {
    Write-Host "An error occurred. Exiting."
    exit 1
}

# Set the script to stop on any error
$ErrorActionPreference = "Stop"

# Prompt for Azure region input
$REGION = Read-Host "Please enter the Azure region"

# Verify that Azure CLI is logged in
try {
    az account show > $null
} catch {
    Write-Host "Please log in to Azure using 'Connect-AzAccount'."
    exit 1
}

# Define filters for quota names
$requiredFilter = "Standard"
$additionalFilters = @("HB", "HC", "Esv", "Fsv", "Dsv3", "Ebds")

# Retrieve and filter VM usage data
try {
    # Get usage information from Azure CLI
    $usagesJson = az vm list-usage --location $REGION
```

```
    if (-not $usagesJson) {
        Write-Host "No data retrieved from Azure CLI."
        exit 1
    }

    # Convert JSON to PowerShell objects
    $usages = $usagesJson | ConvertFrom-Json
    if (-not $usages) {
        Write-Host "Failed to convert usage data."
        exit 1
    }

    # Apply the "Standard" filter
    $filteredUsages = $usages | Where-Object {
        $_.name.value -like "*$requiredFilter*"
    }

    if (-not $filteredUsages) {
        Write-Host "No data found matching the 'Standard' filter."
        exit 1
    }

    # Apply additional filters
    $furtherFilteredUsages = $filteredUsages | Where-Object {
        $match = $false
        foreach ($filter in $additionalFilters) {
            if ($_.name.value -like "*$filter*") {
                $match = $true
                break
            }
        }
        $match
    }

    if (-not $furtherFilteredUsages) {
        Write-Host "No data found after applying additional filters."
        exit 1
    }

    # Display the filtered usage data
    Write-Host "Filtered usage data:" -ForegroundColor Green
    $furtherFilteredUsages | Select-Object @{Name="Quota Name";Expression=
{$_.name.value}}, CurrentValue, Limit | Format-Table
} catch {
    Stop-Error
}
```

Troubleshooting

- **If an error message appears**

    - Ensure that the region name you entered is correct. You can check the region names using az account list-locations -o table.
    - Verify that you have appropriate permissions to access Azure CLI.

- **If quota information cannot be retrieved**

    - Check whether the specified region has the relevant quotas available.
    - Adjust the filter conditions in the script to obtain the necessary quota information.

Notes

- **Using Azure Cloud Shell**

    - Azure Cloud Shell is a browser-based shell environment with PowerShell and Azure CLI pre-installed.
    - Use the "Upload/Download" feature in Cloud Shell to upload files.

- **Customizing the Script**

    - You can modify $requiredFilter and $additionalFilters within the script to change the types of quotas being checked.

- **Permissions and Security**

    - Running the script requires appropriate access rights to your Azure subscription.