Param(

     [Parameter(Mandatory=$True,

                 ValueFromPipelineByPropertyName=$false,

                 HelpMessage='Specify the AKS cluster name.',

                 Position=1)]

                 [String]

                 $aksClusterName,

     [Parameter(Mandatory=$True,

                 ValueFromPipelineByPropertyName=$false,

                 HelpMessage='Specify the name of the resoure group containing the AKS cluster.',

                 Position=2)]

                 [String]

                 $resourceGroupName,

     [Parameter(Mandatory=$True,

                 ValueFromPipelineByPropertyName=$false,

                 HelpMessage='Specify the operation to be performed on the AKS cluster name (Start/Stop).',

                 Position=2)]

                 [ValidateSet('Start','Stop')]

                 [String]

                 $operation

     )

 try

 {

     # Ensures you do not inherit an AzContext in your runbook

     Disable-AzContextAutosave -Scope Process | Out-Null

     # Connect to Azure with system-assigned managed identity

     $AzureContext = (Connect-AzAccount -Identity).context

     #"Setting context to a specific subscription"

     $AzureContext = Set-AzContext -SubscriptionName $AzureContext.Subscription -DefaultProfile $AzureContext

     #Setting REST API Authentication token

     $accToken = Get-AzAccessToken | Select-Object -Property Token

     $AccessToken = $accToken.Token

     $headers\_Auth = @{'Authorization'="Bearer $AccessToken"}

     #Setting GET RestAPI Uri

     $getRestUri = "https://management.azure.com/subscriptions/$($AzureContext.Subscription)/resourceGroups/$resourceGroupName/providers/Microsoft.ContainerService/managedClusters/$($aksClusterName)?api-version=2021-05-01"

     #Setting POST RestAPI Uri

     $postRestUri = "https://management.azure.com/subscriptions/$($AzureContext.Subscription)/resourceGroups/$resourceGroupName/providers/Microsoft.ContainerService/managedClusters/$aksClusterName/$($operation.ToLower())?api-version=2021-05-01"

     try

     {

         #Getting the cluster state

         Write-Output "Invoking RestAPI method to get the cluster state. The request Uri is ==$getRestUri==."

         $getResponse = Invoke-WebRequest -UseBasicParsing -Method Get -Headers $headers\_Auth -Uri $getRestUri

         $getResponseJson = $getResponse.Content | ConvertFrom-Json

         $clusterState = $getResponseJson.properties.powerState.code

         Write-Output "AKS Cluster ==$aksClusterName== is currently ==$clusterState=="

         #Checking if the requested operation can be performed based on the current state

         Switch ($operation)

         {

             "Start"

             {

                 If ($clusterState -eq "Running")

                 {

                     Write-Output "The AKS Cluster ==$aksClusterName== is already ==$clusterState== and cannot be started again."

                 }

                 else

                 {

                     Write-Output "Invoking RestAPI method to perform the requested ==$operation== operation on AKS Cluster ==$aksClusterName==. The request Uri is ==$postRestUri==."

                     $postResponse = Invoke-WebRequest -UseBasicParsing -Method Post -Headers $headers\_Auth -Uri $postRestUri

                     $StatusCode = $postResponse.StatusCode

                 }

             }

             "Stop"

             {

                 If ($clusterState -eq "Stopped")

                 {

                     Write-Output "The AKS Cluster ==$aksClusterName== is already ==$clusterState== and cannot be stopped again."

                 }

                 else

                 {

                     Write-Output "Invoking RestAPI method to perform the requested ==$operation== operation on AKS Cluster ==$aksClusterName==. The request Uri is ==$postRestUri==."

                     $postResponse = Invoke-WebRequest -UseBasicParsing -Method Post -Headers $headers\_Auth -Uri $postRestUri

                     $StatusCode = $postResponse.StatusCode

                 }

             }

             Default

             {

                 Write-Output "Unexpected scenario. The requested operation ==$operation== was not matching any of the managed cases."

             }

         }

     }

     catch

     {

         $StatusCode = $\_.Exception.Response.StatusCode.value\_\_

         $exMsg = $\_.Exception.Message

         Write-Output "Response Code == $StatusCode"

         Write-Output "Exception Message == $exMsg"

     }

     if (($StatusCode -ge 200) -and ($StatusCode -lt 300))

     {

         Write-Output "The ==$operation== operation on AKS Cluster ==$aksClusterName== has been completed succesfully."

     }

     else

     {

         Write-Output "The ==$operation== operation on AKS Cluster ==$aksClusterName== was not completed succesfully."

     }

 }

 catch

 {

     if (!$AzureContext)

     {

         $ErrorMessage = "Connect to Azure with system-assigned managed identity was not found"

         throw $ErrorMessage

     }

     else

     {

         Write-Error -Message $\_.Exception

         throw $\_.Exception

     }

 }