Azure Analysis Services QPU AutoScale

This PowerShell script deploys a customizable QPU AutoScale solution for Azure Analysis Services.

## Requirements

The script requires the subscription containing the target AAS instance also contain an Azure Automation Account, where AutoScale will be deployed. The Automation Account must import the latest updated modules for Az.Accounts, Az.AnalysisServices, Az.Automation, Az.Monitor, and AzureRM.Insights.

The script must be executed by an AAD authenticated user who is a server level administrator of the AAS instance and administrator/owner of the Automation Account.

NOTE: If Active Directory Connect is not setup in AAD to translate Windows accounts to AAD automatically, it is necessary to run Login-AzAccount in the PowerShell command window before running this script.

The script must be run while the instance is started. Otherwise it will fail with an appropriate error.

Manual scaling should not be performed while AutoScale is active. However, manually scaling will also not hurt the configuration. The configuration will simply continue to function as configured, and the next time a threshold is encountered within the configured limits, AutoScale will bring the instance back into its configured limits on the next action.

## Behavior

The AutoScale behavior when reaching the defined upper bound is to first scale up until maximum tier is reached, and thereafter to scale out to the maximum replica count. Likewise, when the lower bound is reached, scaled-out instances are scaled-in first until minimum replica count is reached, and then the instance is scaled down until minimum tier is reached.

The alerting behavior is triggered when any node used for querying within the current configuration passes the configured QPU thresholds, either to increase resources or decrease. If the processing node is separated from the query nodes, it is ignored for AutoScale purposes.

## Deployment

To initially deploy AutoScale, just run DeployAASAutoscale.ps1 from a PowerShell command prompt while authenticated as an AAD user as described above in the requirements. Provide values for all the parameters as prompted.

The required parameters for all operations (deploy/remove) are:

-SubscriptionID

-ResourceGroupName

-AutomationAccount

-ASServerName

To remove AutoScale, include the -Remove parameter with the above required parameters. But it is not necessary to remove an existing configuration to update it. Simply run the command with the new desired values to reconfigure.

When deploying or reconfiguring AutoScale, the following parameters are also required:

-AlertWindowInMins

-MinTier

-MaxTier

-MinReplicas

-MaxReplicas

The script checks the region of the specified AAS instance to be sure the specified tier and replica count max/min values are all within the region’s supported limits. If they are not, an error is reported that outputs the available options within the region’s limits.

The following optional parameters for deployment can be specified, but have default values:

-SeparateProcessingNodeFromQueryReplicas = $true

For scaling up or down and adding replicas, -ScaleUpDownOutAtPctDistanceFromTierMax controls the tier QPU threshold percentage. It makes sense to set this threshold quite close to the top of the relevant tier’s range (10 by default is 90% of the max). For up and out, we want to get the most out of our current tier before we pay for a higher tier or another instance. Also, when scaling down, we want to scale down to the cheaper tier as soon as possible once we know we are under the limits for that tier.

But for scaling IN from a scaled-out scenario with multiple query replicas, then we will want to set a more relaxed threshold, so we have -ScaleInAtPctDistanceFromTierMax (25 by default is only 75% of the max). That’s because a single instance falling below the max for the tier will not typically be enough reason to remove an entire node from the replica count. Instead, we will wait for a more significant lull, and not scale in if that is the next action, until we are a greater distance from the tier maximum.

Finally, there is an optional -Force parameter, which will prevent the script from failing if the current tier/replica count is outside of the configured limit. Normally this will cause an error to be reported, but if -Force is specified, the deployment will continue. The limits will still be applied as specified, and the next scale event, up or down, will move the instance to the next appropriate selection of tier/replica count within its configuration. If there is an existing alert being processed, or if something failed while a prior alert was being processed, this can also cause failure of the script too, but the -Force parameter will ignore any errors caused by these issues and continue to deploy when specified.

## Infrastructure

AutoScale deploys the following objects into Azure:

* A runbook called AASAutoScale-*<instancename>*, deployed into the specified Automation Account.
* Two web hooks for the runbook, which are used to invoke the script for up and down events. They are named AutoScaleUpWebhook*XXXXXXXXX* and AutoScaleDownWebhook*XXXXXXXXX*.
* Two action rules called AutoScaleUpAlert and AutoScaleDownAlert, for the AS server specified.
* Two action groups called AutoScaleUpActionGroup and AutoScaleDownActionGroup, which are used to invoked the webhooks to the runbook, whenever the alerts are fired.

Calling the script with -Remove deletes all these objects from Azure so there is no cleanup required.

## Monitoring/Debugging

To monitor AutoScale, you can check a number of places:

* The AAS instance’s own diagnostics and health history
* The Alerts AutoScale creates for the AAS instance, where there is a history of when they were called
* The history for the AASAutoScale-*<instancename>* runbook in the Automation Account

The history for the runbook is particularly important. You will see a history of times AutoScale was invoked, and for each, on the output from the runbook you can find the result, including the prior and new configuration indicating the action AutoScale took, and the next action it will take when the threshold max or min values are reached. QPU values here are expressed in hard values given current actual tier settings, rather than their configured AutoScale percentage values. If there is any failure, you will find exception details, etc. here as well.