Shutdown Automation

# 1.0.0 Goal

The goal of this solution is to control spend in Azure by shutting down virtual machines outside of working hours. This solution must be able to run across all desired subscriptions.

# 2.0.0 Implementation

The following are characteristics of the implementation:

* Azure Automation is used to host the script for this solution.
* Account credentials are stored in Azure Automation.
  + The password can only be retrieved as a secure credential (not in clear text).
* Azure Table Storage is used to hold configuration information for the solution.
* Azure SQL Database is used to hold log information.
* VMs will be considered for shutdown between a start time and stop time in their local time zone.
* VMs are given a grace period after a startup whereby the VM will not be shut down until after the grace period.
* The solution will run hourly ensure VMs are shut down appropriately across all time zones.
* The solution will send a summary email of activity once a day.

# 3.0.0 Azure Automation

The following Azure Automation Account is used for this solution:

|  |  |
| --- | --- |
| Subscription |  |
| Resource Group |  |
| Location |  |
| Automation Account |  |
| Runbooks |  |

## 3.1.0 Modules

An Azure Automation Account has a set of PowerShell modules associated with it. These modules provide functionality via PowerShell cmdlets. There are some modules that are required for the PowerShell environment to run at, but beyond those standard modules, this solution requires the following modules:

* Azure (v1.6.0+)
* AzureRM.Compute (v1.3.3+)
* AzureRM.Insights (v1.0.11+)
* AzureRM.Network (v1.0.12+)
* AzureRM.Profile (v1.0.11+)

The version numbers listed above are all related to Azure PowerShell version 1.6.0. You can upgrade all these components to version 2.0.x or higher, but all Azure PowerShell must be from the same release (v2.0.x components all seem to have the same version number, hopefully that will continue.

## 3.2.0 Connections, Certificates, Variables, and Credentials

The following assets in the Azure Automation Account are necessary for the proper operation of this solution:

* **AzureRunAsConnection** [Connection] – This is an AzureServicePrincipal type account that is associated to an Azure Active Directory Application. You will specify the ApplicationId, TenantId, CertificateThumbprint (more on this below), and SubscriptionId (CorpCoreTools, but it will access any number of subscriptions). See Section 3.2.1 on how this Service Principal is used.
* **AzureRunAsCertificate** [Certificate] – You will need to generate a certificate using New-AzureAutomationCertificate or similar and then upload it. You can find full details here: <https://azure.microsoft.com/en-us/documentation/articles/automation-certificates/>. See Section 3.2.2 for details on the expiration of this certificate.
* **DataWarehouse** [Credential] – This is a simple username and password of the SQL account that has access to write to the ShutdownLog table of the data warehouse. See Section 6.0.0 for information about the logging.
* **SendGrid** [Credential] – This is a simple username and password to send mail to SendGrid via SMTP. See Section 7.0.0 for more details on SendGrid.

## 3.2.1 Service Principal

The Service Principal account is the identity by which all API calls to Azure will be made. It must be associated with an Azure Active Directory Application, which is described in detail here: <https://www.microsoftpressstore.com/articles/article.aspx?p=2473127>.

Every Azure Subscription that the solution will attempt to shutdown VMs in must list the Service Principal account as at least “Contributor” permission to the Subscription.

## 3.2.2 Certificate Expiration

When a certificate is generated for the Service Principal is must have an expiration date. It is recommended that is one-year.

Each year before the certificate expires, someone must:

* Generate a new certificate
* Upload the certificate
* Change the Connection to use the new certificate thumbprint

# 4.0.0 Scripts

The following PowerShell scripts are contained in the PGES private GitHub repository:

* **shutdown-arm.ps1** – This script is the same as can be deployed in Azure Automation to perform shutdowns. It can also be run locally with the modifications discussed in Section 4.2.0.
* **initialize-storage.ps1** – This script can be executed to create the appropriate tables in Azure Storage for configuration. Those tables are discussed in Sections 5.1.0 and 5.2.0.
* **initialize-warehouse.ps1** – This script can be executed to create the appropriate table in Azure SQL Database for logging. That table is discussed in Section 6.1.0.
* **email-shutdown-logs.ps1** – This script is the same as can be deployed in Azure Automation to email log entries for the past 24 hours. It can also be run locally with the modifications discussed in Section 4.4.0.

## 4.1.0 Running shutdown-arm in Azure Automation

Section 3.0.0 identifies the appropriate Azure Automation Account. To execute the runbook once, simply select it in the portal and click “Start”.

This runbook should be configured to run every hour.

There are 2 sections of the script that are related to authentication that must be properly configured:

* In Log-Operation, the “# credentials for production” section must be uncommented and “# credentials for local testing” must be commented out.
* In the “# login” section of the main body, the “$conn” and “$acnt” lines must be uncommented.

## 4.2.0 Running shutdown-arm.ps1 locally

The script can be run locally with minimal changes. This can be helpful when debugging or making significant changes. There are 2 sections of the script that are related to authentication that must be properly configured:

* In Log-Operation, the “# credentials for production” section must be commented out and “# credentials for local testing” must be uncommented.  
  + The “$dbPassword” must be supplied.
* In the “# login” section of the main body, the “$conn” and “$acnt” lines must be commented out.

You should have the latest version of PowerShell and Azure PowerShell installed locally. You must login to Azure before executing the script using “Login-AzureRmAccount”.

## 4.3.0 Running email-shutdown-logs in Azure Automation

Section 3.0.0 identifies the appropriate Azure Automation Account. To execute the runbook once, simply select it in the portal and click “Start”.

This runbook should be configured to run once a day at 8pm Eastern Time.

There are 2 sections of the script that are related to authentication that must be properly configured:

* At the beginning of the main body for the database credentials, the “# credentials for local testing” must be commented out and “# credentials for production” must be uncommented.
* At the end of the main body for the email credentials, the “# credentials for local testing” must be commented out and “# credentials for production” must be uncommented.

## 4.4.0 Running email-shutdown-logs.ps1 locally

The script can be run locally with minimal changes. This can be helpful when debugging or making significant changes. There are 2 sections of the script that are related to authentication that must be properly configured:

* At the beginning of the main body for the database credentials, the “# credentials for local testing” must be uncommented and “# credentials for production” must be commented out.
* At the end of the main body for the email credentials, the “# credentials for local testing” must be uncommented and “# credentials for production” must be commented out.

You should have the latest version of PowerShell and Azure PowerShell installed locally.

# 5.0.0 Configuration Tables

There are a couple of configuration tables that are stored in Azure Storage:

|  |  |
| --- | --- |
| Subscription |  |
| Resource Group |  |
| Location |  |
| Storage Account |  |
| Tables | automationRules  automationTimezones |

## 5.1.0 automationRules

The automationRules table contains a set of rules that allow you to manage the scope of what is considered for shutdown.

This table is created by the “initialize-storage.ps1” script. A sample row is created, but this can be deleted or modified as appropriate.

At a minimum, there should be 1 rule for each subscription that should be considered for shutdown.

More specific rules take precedent over less specific rules. Specifying a “SubscriptionId” (which is required) is worth 4 points, specifying a “Region” is worth 8 points, and specifying a “VM” is worth 32 points. For example, a rule that specifies a subscription and region would have a priority of 12 and be rated higher than a rule that specifies just a subscription which would have a priority of just 4.

It is recommended that you use [Microsoft Azure Storage Explorer](http://storageexplorer.com/) (available for Windows, macOS, and Linux) to interact with the storage tables.

|  |  |  |
| --- | --- | --- |
| Column | Required? | Description |
| PartitionKey | Yes | This should always be “automation”, there is no need to partition this small amount of data. |
| RowKey | Yes | This should be a unique name for the rule, ex. “Shutdown East US”. It can be up to 255 characters. |
| SubscriptionId | Yes | This should be the ID of the subscription that contains VMs that should be shutdown, ex. “bb8b8c18-67da-4a87-be7a-680da44f18e0”. |
| Region | No, \* | An Azure region code can optionally be used to scope this rule to only those VMs within a region, ex. “eastus”.  \* This is required if a VM is specified. |
| VM | No | The name of a VM can optionally be used to scope this rule to a specific VM, “plasne-testshutdown”. |
| ShutdownWindowFrom | No | An integer can optionally be used to describe the hour to start considering the VMs for shutdown, ex. “18”. The hour is 24-time in the local time zone. For instance, if the VM is located in “westus” and the shutdown window starts at “18”, it will start considering the VM for shutdown at 6pm in the Pacific time zone.  If this is not specified, a default value of “19” (7pm local time) is used. |
| ShutdownWindowTo | No | An integer can optionally be used to describe the hour to stop considering the VMs for shutdown, ex. “6”. The hour is 24-time in the local time zone. For instance, if the VM is located in “westus” and the shutdown window stops at “6”, it will stop considering the VM for shutdown at 6am in the Pacific time zone.  If this is not specified, a default value of “5” (5am local time) is used. |
| GracePeriod | No | An integer can optionally be used to describe the number of hours a VM is allowed to run after a startup, ex. “4”. Using this example, even if a VM is within the shutdown window, it will be allowed to run for a minimum of 4 hours since when it was last started up.  If this is not specified, a default value of “4” (4 hours) is used. |
| Shutdown | No | This should be a value of “true”, “false”, or it can be left blank.  If it is “true”, VMs that match this rule will be shutdown (provided they are within the window and the grace period is not preventing a shutdown).  If it is “false”, VMs that match this rule will not be shutdown. Generally this is used to specify a higher priority (more granular) rule that prevents VMs in a specific region or a specific VM from shutting down.  If it is “blank” (or anything other than “true” or “false”), the rule is ignored. |
| Configure | No | This is not used for shutdown automation. Instead, please see the documentation for the DSC solution. |
| Timestamp | Automatic | This is an field that is automatically populated with the timestamp that this rule last was created or modified. |

## 5.2.0 automationTimezones

The automationTimezones table is used by the solution to determine the appropriate time zone code used by Windows for resources in the specified Azure region. This table is what enables the solution to normalize the shutdown window so it is always in the local time.

This table is created ***and populated*** by the “initialize-storage.ps1” script. It is only necessary to modify this table if there is a new Azure region.

|  |  |  |
| --- | --- | --- |
| Column | Required? | Description |
| PartitionKey | Yes | This should always be “timezone”, there is no need to partition this small amount of data. |
| RowKey | Yes | This should be the code for the Azure region, ex. “eastus”. |
| Location | No | This is the code that was used by the Azure Classic APIs. It is not used in this Azure Resource Manager solution. |
| Timestamp | Automatic | This is an field that is automatically populated with the timestamp that this rule last was created or modified. |
| Timezone | Yes | This should be the time zone name used by Windows to describe the time zone that should be used for the region code. For example, the Azure region code of “eastus” is in the “Eastern Standard Time” time zone. |

# 6.0.0 Logging Table

There is a logging table contained in an Azure SQL Database:

|  |  |
| --- | --- |
| Subscription |  |
| Resource Group |  |
| Location |  |
| Azure SQL Server |  |
| Azure SQL Database |  |
| Table | ShutdownLog |

## 6.1.0 ShutdownLog

The shutdown script will log all activity to the ShutdownLog table. There should be a log entry for any VM that was within the shutdown window. The following status messages will be logged:

* successfully shutdown
* ignored, already shutdown
* ignored, in grace period

VMs that are not within the shutdown window are not logged as this would create a lot of noise in the logs.

|  |  |  |
| --- | --- | --- |
| Column | Required? | Description |
| Timestamp | Yes | This will be the timestamp in UTC when the activity being logged was performed. |
| RuleId | Yes | This will be the name of the Rule (RowKey from Section 5.1.0) that was the highest priority rule affecting this activity. |
| SubscriptionId | Yes | This is the ID of the Azure Subscription containing the VM that reported this activity. |
| Region | Yes | This is the Azure Region code containing the VM that reported this activity. |
| VM | Yes | This is the name of the VM that reported this activity. |
| Status | Yes | This is the status message explaining what happened when the VM was asked to shut down. You can see those messages above. |

Currently none of the jobs clean data out of the ShutdownLog table, but it is likely this data does not need to retain forever.

# 7.0.0 SendGrid

The “email-shutdown-logs” script must send email messages to the administrators of the server once a day. Microsoft does not provide SMTP capability in Azure, but a 3rd party solution is provided via SendGrid.

Should SendGrid ever be changed to a pricing tier that costs something, that charge will bill as an “overage” charge (meaning ABB ES will need to send money to Microsoft instead of the charges being deducted from the pre-commit).

|  |  |
| --- | --- |
| Subscription | Corp Core Tools ce74f56c-fc45-4a2a-aefa-fd2e041a7f77 |
| Resource Group | pgessendgrid |
| Location | Central US |
| SendGrid Account | pgessendgrid |
| Pricing Tier | F1 Free 25,000 emails/month |