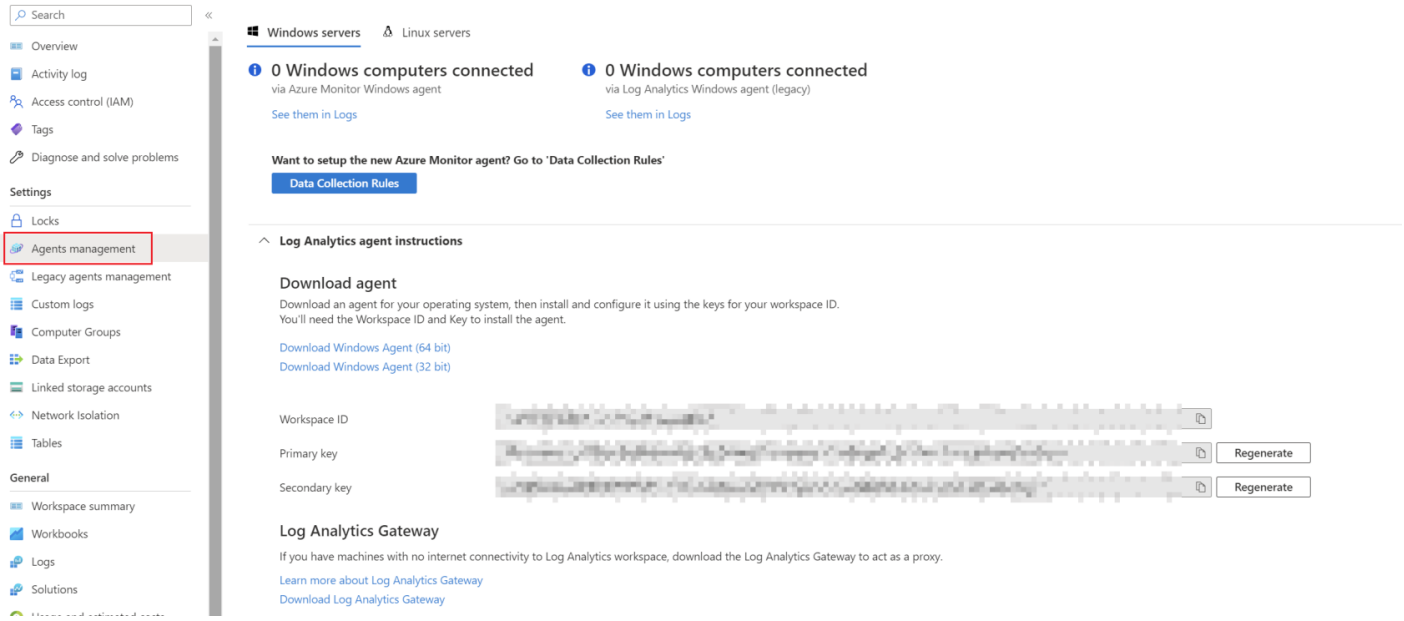
Write custom logs via PowerShell

With Log Analytics you can write custom logs. For this the API can be used to write new logs to the Log Analytics Workspace. The whole API is documented here: https://learn.microsoft.com/en-us/rest/api/loganalytics/create-request

Script template

This script template uses the first three variables. The first two can be found under "Agents management" -> "Log Analytics agent instructions".



The third variable "LogType" defines the type of log you are going to send. This can be the same as existing log entries or a completly new one. This type of log defines if a new table or an existing table is created or used to store the logs.

```
$customerId = ""
$sharedKey = ""
$LogType = ""

Function Build-Signature ($customerId, $sharedKey, $date, $contentLength, $method,
```

```
$contentType, $resource){
    $xHeaders = "x-ms-date:" + $date
    $stringToHash = $method + "`n" + $contentLength + "`n" + $contentType + "`n" + $xHeaders +
"`n" + $resource

    $bytesToHash = [Text.Encoding]::UTF8.GetBytes($stringToHash)
    $keyBytes = [Convert]::FromBase64String($sharedKey)

    $sha256 = New-Object System.Security.Cryptography.HMACSHA256
    $sha256.Key = $keyBytes
    $calculatedHash = $sha256.ComputeHash($bytesToHash)
    $encodedHash = [Convert]::ToBase64String($calculatedHash)
    $authorization = 'SharedKey {0}:{1}' -f $customerId,$encodedHash
    return $authorization
}

Function Post-LogAnalyticsData ($customerId, $sharedKey, $body, $logType){
    $method = "POST"
    $contentType = "application/json"
    $resource = "/api/logs"
    $rfc1123date = ([DateTime]::UtcNow).ToString("r")
    $contentLength = $body.Length
    $signature = Build-Signature -customerId $customerId -sharedKey $sharedKey -date
$rfc1123date -contentLength $contentLength -method $method -contentType $contentType -resource
$resource

    $uri = "https://" + $customerId + ".ods.opinsights.azure.com" + $resource + "?api-
version=2016-04-01"

    $headers = @{
        "Authorization" = $signature;
        "Log-Type" = $logType;
        "x-ms-date" = $rfc1123date;
    }

    $response = Invoke-WebRequest -Uri $uri -Method $method -ContentType $contentType -Headers
$headers -Body $body -UseBasicParsing
    return $response.StatusCode
}
```

```
$Properties = [Ordered] @{
    "ComputerName"     = $env:computername
    "User"             = $env:Username
}

$CustomLogs = New-Object -TypeName "PSObject" -Property $Properties | ConvertTo-JSON -Depth 10



#Submit the data to the API endpoint
$params = @{
    CustomerId = $customerId
    SharedKey  = $sharedKey
    Body       = ([System.Text.Encoding]::UTF8.GetBytes($CustomLogs))
    LogType    = $LogType
}
$LogResponse = Post-LogAnalyticsData @params
$LogResponse
```

The variable "LogResponse" is in a successful execution filled with the value 200.

Use Case

A possible use case is for example is to write an Azure Function to provide a custom Rest API to write logs to a function. This has multiple advantages over writing directly to the Log Analytics Workspace. Everything from less code to credential leaks will be provided with a simple REST Call to the Azure Function. More on this in the chapter "Azure Function".

Revision #5
Created 2022-12-05 09:57:45 UTC
Updated 2025-07-31 07:49:06 UTC by Luca Noah Caprez