# A close up of a sign Description automatically generatedAzure Monitor and Splunk Integration

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2019 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

[Azure Monitor and Splunk Integration 1](#_Toc31290273)

[Hierarchy 4](#_Toc31290274)

[Configure Azure Event Hubs sink for Virtual Machines 4](#_Toc31290275)

[Deploy the Diagnostics Extension for Windows VMs with Event Hubs integration 4](#_Toc31290276)

[Deploy the Diagnostics Extension for Linux VMs with Event Hubs integration 5](#_Toc31290277)

[Configure Azure Policy for Event Hubs integration compliance 6](#_Toc31290278)

[Policies 7](#_Toc31290279)

[Appendix A: Sample Windows Azure Diagnostic Extension configuration 8](#_Toc31290280)

[Appendix B: Sample Linux Azure Diagnostics Extension Configuration 9](#_Toc31290281)

[PublicSettings Section 9](#_Toc31290282)

[ProtectedSettings Section 9](#_Toc31290283)

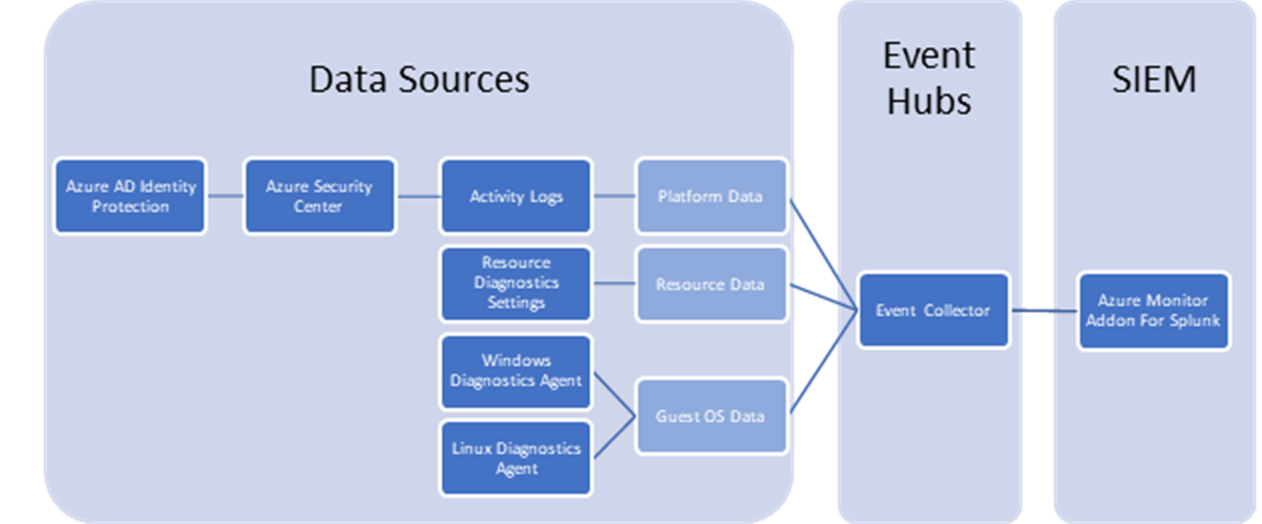
[Appendix C: Event Hubs SAS URI Generation 10](#_Toc31290284)

[Appendix D: Audit resource diagnostics settings 11](#_Toc31290285)

[Appendix E: Audit all VMs with no diagnostics extension installed 13](#_Toc31290286)

[Appendix F: Audit for expected VM diagnostic agent name 14](#_Toc31290287)

# Hierarchy



On the VMs side of things, you’ll need the following triggered in a runbook to trigger the Event Hubs event sink:

## Configure Azure Event Hubs sink for Virtual Machines

Windows and Linux virtual machines can be configured to forward event data to Event Hubs using the Windows Azure Diagnostics Extension and the Linux Diagnostic Extension. This can be achieved through deployment of a custom diagnostics agent configuration which specifies an Event Hub sink.

## Deploy the Diagnostics Extension for Windows VMs with Event Hubs integration

Deploying the Windows Azure Diagnostics Extension requires the deployment of a custom Windows Azure Diagnostics Extension configuration. This sample configuration deploys the Diagnostics Extension with the following Event Hubs forwarding configuration:

* Performance Metrics: None
* Event Logs
  + Application: Critical, Error, Warning
  + Security: Audit Fail
  + System: Critical, Error, Warning
* Crash Dumps: None
* IIS Logs: None
* .NET App logs: None
* ETW Tracing: None

See *Appendix A: Sample Windows Azure Diagnostic Extension configuration* for the sample configuration.

To deploy the sample configuration, use the following PowerShell:

Set-AzureRmVMDiagnosticsExtension -ResourceGroupName RG -VMName WinVM1 -DiagnosticsConfigurationPath 'D:\WADExt.json' -Name “LogToEventHubs”



## Deploy the Diagnostics Extension for Linux VMs with Event Hubs integration

Deploying the Linux Azure Diagnostics Extension requires the deployment of a custom Linux Azure Diagnostics Extension configuration. This sample configuration deploys the Diagnostics Extension with the following Event Hubs forwarding configuration:

* LOG\_AUTH: LOG\_DEBUG
* LOG\_AUTHPRIV: LOG\_DEBUG
* LOG\_CRON: LOG\_DEBUG
* LOG\_DAEMON: LOG\_DEBUG
* LOG\_FTP: LOG\_DEBUG
* LOG\_KERN: LOG\_DEBUG
* LOG\_LOCAL0: LOG\_DEBUG
* LOG\_LOCAL1: LOG\_DEBUG
* LOG\_LOCAL2: LOG\_DEBUG
* LOG\_LOCAL3: LOG\_DEBUG
* LOG\_LOCAL4: LOG\_DEBUG
* LOG\_LOCAL5: LOG\_DEBUG
* LOG\_LOCAL6: LOG\_DEBUG
* LOG\_LOCAL7: LOG\_DEBUG
* LOG\_LPR: LOG\_DEBUG
* LOG\_MAIL: LOG\_DEBUG
* LOG\_NEWS: LOG\_DEBUG
* LOG\_SYSLOG: LOG\_DEBUG
* LOG\_USER: LOG\_DEBUG
* LOG\_UUCP: LOG\_DEBUG

See *Appendix B: Sample Linux Azure Diagnostics Extension Configuration* for a sample configuration.

To deploy the sample configuration, use the following PowerShell:

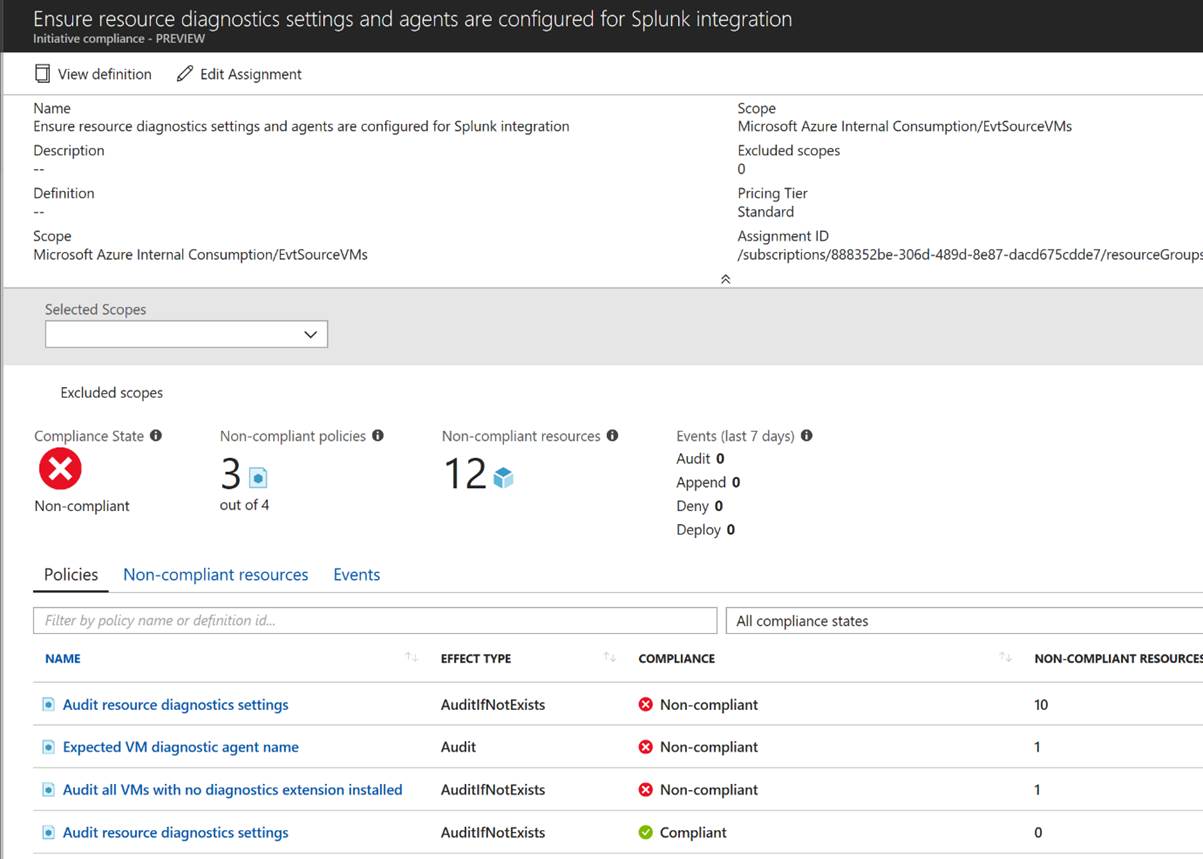
Set-AzureRmVMExtension -ResourceGroupName "RG" -VMName "LinuxVM1" -Name "LogToEventHubs" -Location "Australia Southeast" -Publisher "Microsoft.Azure.Diagnostics" -ExtensionType "LinuxDiagnostic" -TypeHandlerVersion "3.0" -SettingString (Get-Content D:\LadExtSettings.json -Raw) -ProtectedSettingString (Get-Content D:\LadExtProtectedSettings.json -Raw)



Note: The ProtectedSettings block will require a signed SAS URI. See *Appendix C: Event Hubs SAS URI Generation* for a script to assist with this.

## Configure Azure Policy for Event Hubs integration compliance

Azure Policy can be deployed to provide resource configuration compliance checking and reporting. In the context of monitoring and SIEM integration, resources and virtual machine configuration can be evaluated to ensure event logs are configured to forward to Event Hubs.

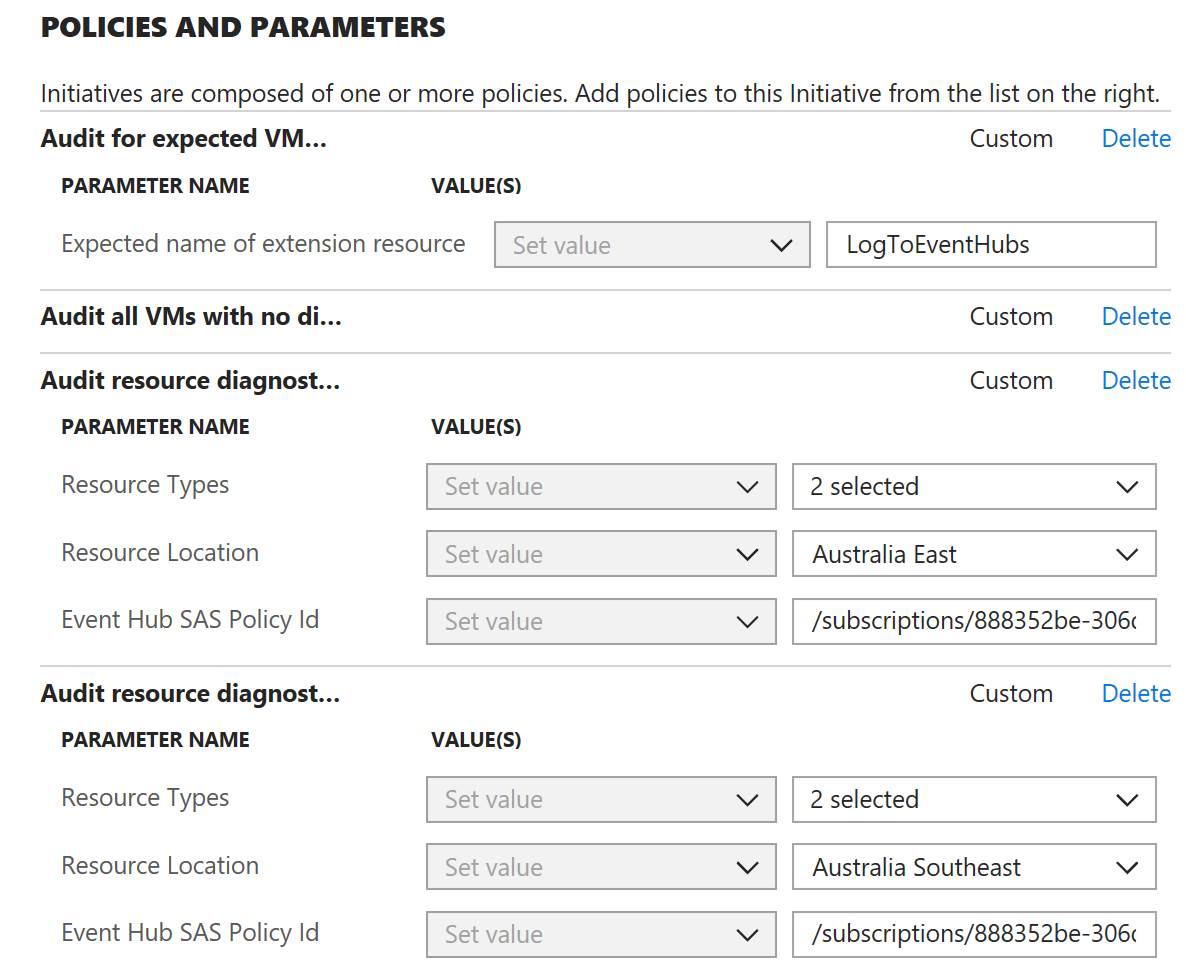


Sample Azure Policy definitions can be found in the appendices for the following compliance definitions:

* **Audit resource diagnostics settings**
  + Finds all resources found in the set of resourceTypes located in the region resourceLocation and marks as non-compliant if:
    - eventHubSasPolicyId does not point to the specified Event Hub authorisation rule, or
    - If Logs are not enabled
  + Parameters:
    - resourceTypes
    - resourceLocation
    - eventHubSasPolicyId
  + See *Appendix D: Audit resource diagnostics settings*
  + Use the list contained in this report to determine which resources to enable logging on:  [Supported services, schemas, and categories for Azure Diagnostic Logs](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fazure%2Fmonitoring-and-diagnostics%2Fmonitoring-diagnostic-logs-schema&data=02%7C01%7CJohn.Jenner%40microsoft.com%7C5e259f4f47834ad0ebe708d676def1bb%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636827097498138207&sdata=CrT5cO%2BzwjYXquQofbdoIE6MI52%2BObVlwcJbQuKYJog%3D&reserved=0)
* **Audit all VMs with no diagnostics extension installed**
  + Finds all VMs with no diagnostics extension installed and marks as non-compliant.
  + See *Appendix E: Audit all VMs with no diagnostics extension installed*
* **Audit for expected VM diagnostic agent name**
  + Finds all installed VM diagnostic extensions and marks as non-compliant if the assigned name for the extension does not match the specified in extensionName. This can be used to help detect if the extension has been deployed with the appropriate configuration that enabled the Event Hub sink.
  + Parameters:
    - extensionName
  + See *Appendix F: Audit for expected VM diagnostic agent name*

## Policies

The policies can then be combined into a single Azure Policy Initiative and assigned to the subscription:



# Appendix A: Sample Windows Azure Diagnostic Extension configuration

{

    "PublicConfig": {

            "StorageAccount": " yourstorageaccountname",

            "WadCfg": {

              "DiagnosticMonitorConfiguration": {

                "overallQuotaInMB": 5120,

                "sinks": "HotPath",

                "Metrics": {

                  "MetricAggregation": [

                    {

                      "scheduledTransferPeriod": "PT1H"

                    },

                    {

                      "scheduledTransferPeriod": "PT1M"

                    }

                  ]

                },

                "DiagnosticInfrastructureLogs": {

                  "scheduledTransferLogLevelFilter": "Error",

                  "scheduledTransferPeriod": "PT1M"

                },

                "PerformanceCounters": {

                  "scheduledTransferPeriod": "PT1M",

                  "PerformanceCounterConfiguration": []

                },

                "WindowsEventLog": {

                  "scheduledTransferPeriod": "PT1M",

                  "DataSource": [

                    {

                      "name": "Application!\*[System[(Level=1 or Level=2 or Level=3)]]"

                    },

                    {

                      "name": "System!\*[System[(Level=1 or Level=2 or Level=3)]]"

                    },

                    {

                      "name": "Security!\*[System[(band(Keywords,4503599627370496))]]"

                    }

                  ]

                },

                "Directories": {

                  "scheduledTransferPeriod": "PT1M"

                }

              },

              "SinksConfig": {

                  "Sink": [

                      {

                          "name": "HotPath",

                          "EventHub": {

                              "Url": "<https://EventCollector.servicebus.windows.net/vmdiag>",

                              "SharedAccessKeyName": "SendOnlyKey"

                          }

                      }

                  ]

              }

            }

    },

    "PrivateConfig": {

        "storageAccountName": "yourstorageaccountname",

        "storageAccountKey": "YourStorageKey==",

        "storageAccountEndPoint": "<https://core.windows.net/>",

        "EventHub": {

            "Url": "<https://EventCollector.servicebus.windows.net/vmdiag>",

            "SharedAccessKeyName": "SendOnlyKey",

            "SharedAccessKey": YourEventHubsSasPolicyKey="

        }

    }

}

# Appendix B: Sample Linux Azure Diagnostics Extension Configuration

## PublicSettings Section

{

        "StorageAccount": "yourstorageaccountname",

        "ladCfg": {

            "diagnosticMonitorConfiguration": {

            "eventVolume": "Medium",

            "metrics": {

                "metricAggregation": [

                {

                    "scheduledTransferPeriod": "PT1M"

                },

                {

                    "scheduledTransferPeriod": "PT1H"

                }

                ]

            },

            "performanceCounters": {

                "performanceCounterConfiguration": []

            },

            "syslogEvents": {

                "sinks":"HotPath",

                "syslogEventConfiguration": {

                "LOG\_AUTH": "LOG\_DEBUG",

                "LOG\_AUTHPRIV": "LOG\_DEBUG",

                "LOG\_CRON": "LOG\_DEBUG",

                "LOG\_DAEMON": "LOG\_DEBUG",

                "LOG\_FTP": "LOG\_DEBUG",

                "LOG\_KERN": "LOG\_DEBUG",

                "LOG\_LOCAL0": "LOG\_DEBUG",

                "LOG\_LOCAL1": "LOG\_DEBUG",

                "LOG\_LOCAL2": "LOG\_DEBUG",

                "LOG\_LOCAL3": "LOG\_DEBUG",

                "LOG\_LOCAL4": "LOG\_DEBUG",

                "LOG\_LOCAL5": "LOG\_DEBUG",

                "LOG\_LOCAL6": "LOG\_DEBUG",

                "LOG\_LOCAL7": "LOG\_DEBUG",

                "LOG\_LPR": "LOG\_DEBUG",

                "LOG\_MAIL": "LOG\_DEBUG",

                "LOG\_NEWS": "LOG\_DEBUG",

                "LOG\_SYSLOG": "LOG\_DEBUG",

                "LOG\_USER": "LOG\_DEBUG",

                "LOG\_UUCP": "LOG\_DEBUG"

                }

            }

            },

            "sampleRateInSeconds": 15

        }

}

## ProtectedSettings Section

{

    "storageAccountName": "yourstorageaccountname",

    "storageAccountSasToken": "sv=2017-07-29&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-12-31T07:11:08Z&st=2018-04-12T00:11:08Z&spr=https&sig=YourSignature

",

    "storageAccountEndPoint": "<https://core.windows.net/>",

    "sinksConfig": {

        "sink" : [

            {

                "name": "HotPath",

                "type": "EventHub",

                "sasURL": "<https://eventcollector.servicebus.windows.net/vmdiag?sr=eventcollector.servicebus.windows.net%2fvmdiag&sig=YourSignature&skn=SendOnlyKey>"

            }

        ]

    }

}

# Appendix C: Event Hubs SAS URI Generation

$uri = "<https://eventcollector.servicebus.windows.net/vmdiag>"

$sasKeyName = "RootManageSharedAccessKey"

$sasKeyValue = "YourSASKey"

$targetUri = $uri.ToLower()

$ExpiresDate = [System.DateTime]::new(2020,12,31)

$Expires = ($ExpiresDate - ([System.DateTime]::new(1970,1,1))).TotalSeconds

$toSign = $targetUri + "`n" + $Expires

$hmacsha = New-Object System.Security.Cryptography.HMACSHA256

$hmacsha.key = [Text.Encoding]::ASCII.GetBytes($sasKeyValue)

$signature = [Convert]::ToBase64String($hmacsha.ComputeHash([Text.Encoding]::ASCII.GetBytes($toSign)))

$signature = [System.Web.HttpUtility]::UrlEncode($Signature)

$token = "sr=" + [System.Web.HttpUtility]::UrlEncode($targetUri) + "&sig=" + $signature + "&se=" + $Expires + "&skn=" + $sasKeyName

$token

# Appendix D: Audit resource diagnostics settings

{

    "parameters": {

        "resourceTypes": {

            "type": "Array",

            "metadata": {

            "displayName": "Resource Types",

            "description": null,

            "strongType": "resourceTypes"

            }

        },

        "resourceLocation": {

            "type": "String",

            "metadata": {

            "displayName": "Resource Location",

            "description": null,

            "strongType": "location"

            }

        },

        "eventHubSasPolicyId": {

            "type": "String",

            "metadata": {

            "displayName": "Event Hub SAS Policy Id",

            "description": null

            }

        }

    },

    "policyRule": {

      "if": {

        "allOf": [

          {

            "field": "type",

            "in": "[parameters('resourceTypes')]"

          },

          {

            "field": "location",

            "equals": "[parameters('resourceLocation')]"

          }

        ]

      },

      "then": {

        "effect": "auditIfNotExists",

        "details": {

          "type": "Microsoft.Insights/diagnosticSettings",

          "existenceCondition": {

            "allOf": [

              {

                "field": "Microsoft.Insights/diagnosticSettings/logs.enabled",

                "equals": "true"

              },

              {

                "field": "Microsoft.Insights/diagnosticSettings/eventHubAuthorizationRuleId",

               "equals": "[parameters('eventHubSasPolicyId')]"

              }

            ]

          }

        }

      }

    }

  }

# Appendix E: Audit all VMs with no diagnostics extension installed

{

    "policyRule": {

      "if": {

        "allOf": [

          {

            "field": "type",

            "equals": "Microsoft.Compute/virtualMachines"

          }

        ]

      },

      "then": {

        "effect": "auditIfNotExists",

        "details": {

          "type": "Microsoft.Compute/virtualMachines/extensions",

          "existenceCondition": {

            "allOf": [

              {

                "field": "Microsoft.Compute/virtualMachines/extensions/publisher",

                "equals": "Microsoft.Azure.Diagnostics"

              }

            ]

          }

        }

      }

    }

  }

# Appendix F: Audit for expected VM diagnostic agent name

{

    "parameters": {

        "extensionName": {

          "type": "String",

          "metadata": {

            "displayName": "Expected name of extension resource",

            "description": null

          }

        }

      },

      "policyRule": {

      "if": {

        "allOf": [

          {

            "field": "type",

            "equals": "Microsoft.Compute/virtualMachines/extensions"

          },

          {

            "field": "Microsoft.Compute/virtualMachines/extensions/publisher",

            "equals": "Microsoft.Azure.Diagnostics"

          },

          {

            "field": "Microsoft.Compute/virtualMachines/extensions/type",

            "in": [

              "IaaSDiagnostics",

              "LinuxDiagnostic"

            ]

          },

          {

            "not": {

              "field": "name",

              "equals": "[parameters('extensionName')]"

            }

          }

        ]

      },

      "then": {

        "effect": "audit"

      }

    }

  }