Vectra Stream for Azure Sentinel AMA Configuration Guide

Version: May 2024

https://support.vectra.ai/s/article/KB-VS-1771

Table of Contents

Introduction *3*

Applicability *3*

Architecture Summary *3*

Impact on Existing (OMS) Deployments *3*

Configuration Method: Syslog-NG *3*

- Prepare Linux Server 4
- Install Arc Agent 4
- Modify Syslog-NG Configuration 7
- Deploy Vectra Stream App 8
- Verify Custom Tables 8
- Install Syslog via AMA Data Connector 9
- Create Data Collection Rule 1 10
- Create Data Collection Rule 2 10
- Verify Data Collection Rules 11
- Modify Stream JSON File 11
- Customize Data Collection Rule 1 11
- Modify and Customize DCR2 13
- Deploy Data Collection Rules 13
- Configure Vectra 14
- Validation 14
- Disable OMS Agent 15

Introduction

Vectra AI Stream for Microsoft Azure Sentinel v1.0 utilizes Microsoft OMS (Log Analytics) agent to collect event data from Vectra and send it to log analytics workspace. This agent is schedule for end-of-life August 31, 2024, and is replaced with the Azure Monitor Agent (AMA). This document explains how to configure Microsoft AMA to ingest Vectra network telemetry (aka metadata) into Microsoft Azure Sentinel Log Analytics.

Applicability

This document applies to environments where pre-existing deployments must migrate from OMS to AMA as well as for new deployments starting with AMA.

Architecture Summary

A data connector is deployed and configured to send Vectra metadata to log analytics. Once ingested into log analytics, Vectra metadata is stored into individual custom tables as JSON data per metadata type (there are currently 17 metadata types provided with Vectra Stream).

Impact on Existing (OMS) Deployments

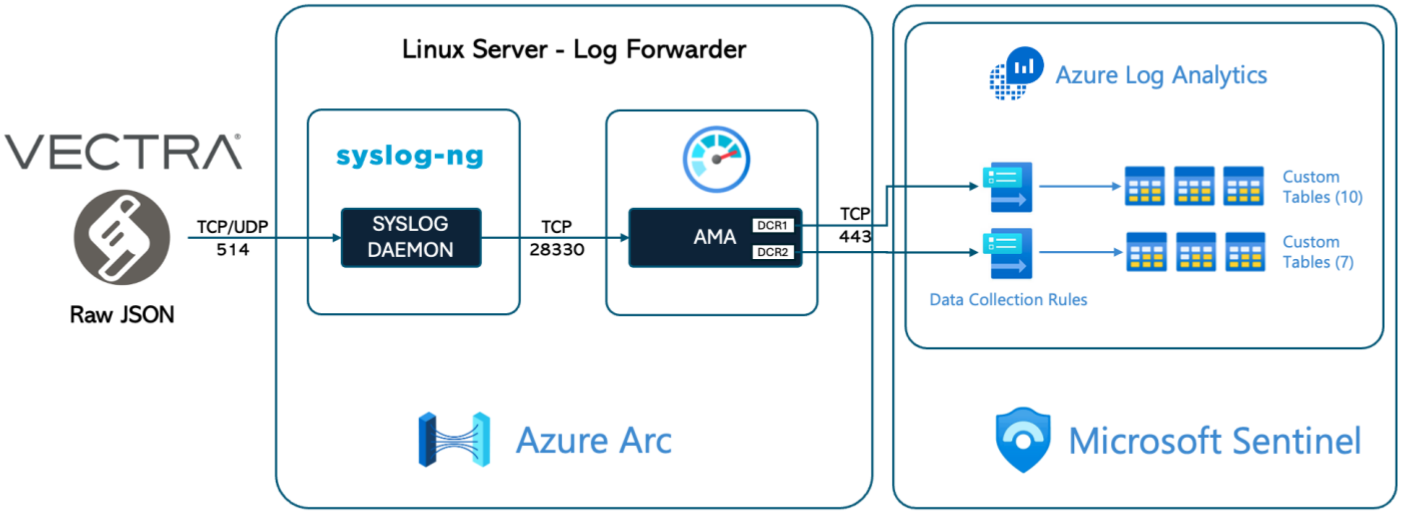
When migrating from OMS to AMA deployment model, please note that Vectra Stream metadata will be stored in multiple custom tables so data will stop being ingested into the single custom table VectraStream_CL when you stop the OMS agent.

Since all streams won't be loaded into that single custom table this means that existing Stream Workbooks and Queries will not function against the new tables. These will continue to function on the existing data until the data expires based on your retention policy.

Vectra is investigating adding a workbook and sample queries to operate on the new custom tables in the future.

Configuration Method: Syslog-NG

Using syslog-ng to filter the Vectra metadata streams (data flows) is required to enable all streams to be ingested into the same Sentinel workspace. This method uses syslog-ng to filter the incoming data streams and spread them across two outputs so that each output can use a different Data Collection Rule thus allowing all Stream metadata to be sent to the same receiver.



Prepare Linux Server

Microsoft Azure Monitor Agent (AMA) operates on an Azure Arc enabled Linux server. For this integration, Vectra sends the Stream data in raw JSON data and AMA requires a receiver. Since Vectra provides multiple metadata types and a Data Collection Rule only supports up to 10 data flows, a syslog-ng receiver running on the Linux server will be required.

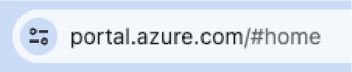
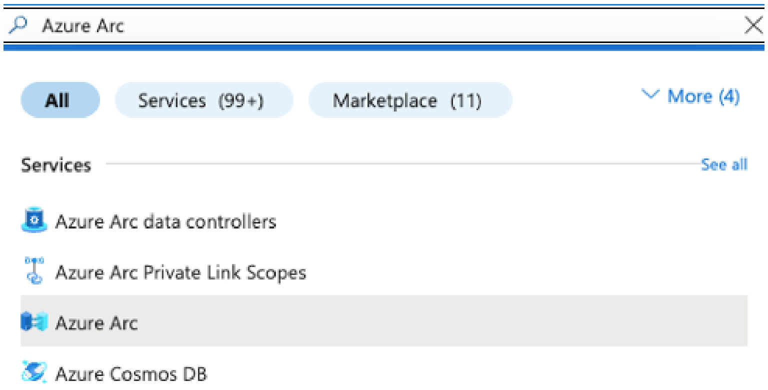
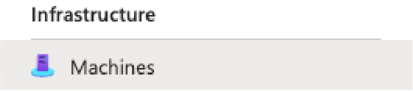
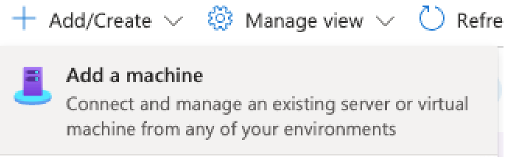
This documentation was validated on an Azure Arc enabled Linux server running Ubuntu 22.04. The default syslog configuration for Ubuntu 22.04 is rsyslog so the following steps were taken to make syslog-ng the primary syslog receiver after connecting to the server using SSH.

The commands to the right remove rsyslog
`# stop rsyslog`
`# remove the rsyslog package`
`# clean up left over configuration files`
`# verify that rsyslog has been removed`
	`sudo systemctl stop rsyslog`
	`sudo apt remove --purge rsyslog`
	`sudo apt purge --auto-remove rsyslog`
	`sudo dpkg -l | grep rsyslog`

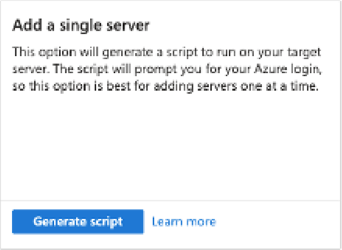
The commands to the right installs syslog-ng
`# update package lists`
`# install syslog-ng`
`# enable syslog-ng`
`# start syslog-ng`
`# verify syslog-ng is running`
	`sudo apt update`
	`sudo apt install syslog-ng`
	`sudo systemctl enable syslog-ng`
	`sudo systemctl start syslog-ng`
	`sudo systemctl status syslog-ng`

Install Arc Agent

The Linux log server that will be used to run AMA must be connected to Azure via the Arc machine agent.
If the machine is already connected to Azure using Azure Arc, this section can be skipped.

Access the Azure Portal	
Search resources for Azure Arc Select Azure Arc from the list	 Azure Arc All Services (99+) Marketplace (11) More (4) Services See all Azure Arc data controllers Azure Arc Private Link Scopes Azure Arc Azure Cosmos DB
Select Machines from left-hand window	
From Add/Create drop-down menu select Add a machine	 Add/Create Manage view Refre Add a machine Connect and manage an existing server or virtual machine from any of your environments

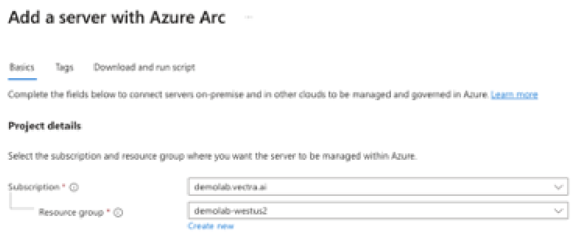
Select Generate script from Add a single server



Add a single server

This option will generate a script to run on your target server. The script will prompt you for your Azure login, so this option is best for adding servers one at a time.

Enter appropriate Project details



Add a server with Azure Arc

Complete the fields below to connect servers on-premise and in other clouds to be managed and governed in Azure. Learn more

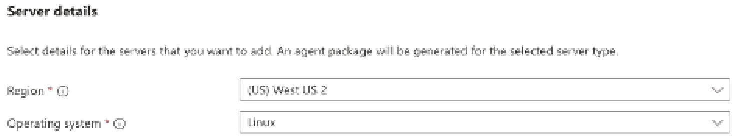
Project details

Select the subscription and resource group where you want the server to be managed within Azure.

Subscription * ⓘ demolab.vectra.ai

Resource group * ⓘ demolab-westus2

Enter Server details



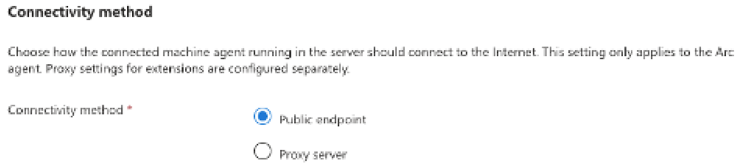
Server details

Select details for the servers that you want to add. An agent package will be generated for the selected server type.

Region * ⓘ (US) West US 2

Operating system * ⓘ Linux

Select Connectivity method



Connectivity method

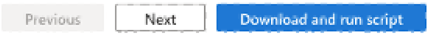
Choose how the connected machine agent running in the server should connect to the Internet. This setting only applies to the Arc agent. Proxy settings for extensions are configured separately.

Connectivity method *

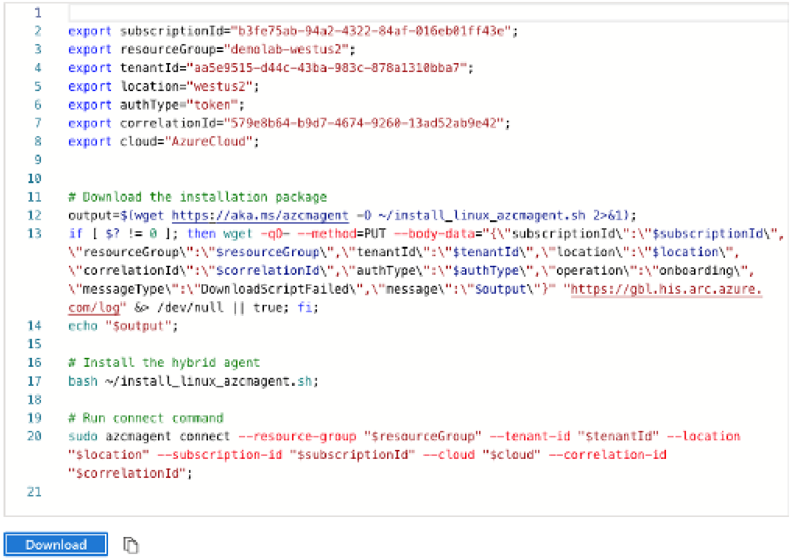
● Public endpoint

○ Proxy server

Select Download and run script

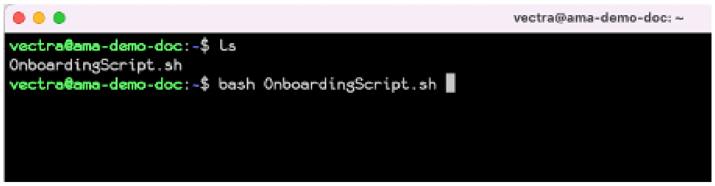


Download or copy the script code



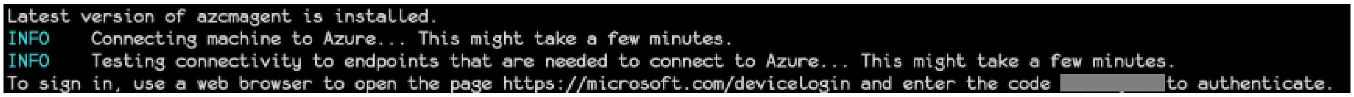
```

export subscriptionId="b3fe75ab-94a2-4322-84af-016eb01ff43e";
export resourceGroup="demolab-westus2";
export tenantId="aa5e9515-d44c-43ba-983c-878a1310bba7";
export location="westus2";
export authType="token";
export correlationId="579e8b64-b9d7-4674-9260-13ad52ab9e42";
export cloud="AzureCloud";


# Download the installation package
output=$(wget https://aka.ms/azcmagent -O ~/install_linux_azcmagent.sh 2>&1);
if [ $? != 0 ]; then wget -qO- --method=PUT --body-data="{\"subscriptionId\":\"$subscriptionId\",
\"resourceGroup\":\"$resourceGroup\",\"tenantId\":\"$tenantId\",\"location\":\"$location\",
\"correlationId\":\"$correlationId\",\"authType\":\"$authType\",\"operation\":\"onboarding\",
\"messageType\":\"DownloadScriptFailed\",\"message\":\"$output\"}" "https://gbl.his.arc.azure.
com/log" &> /dev/null || true; fi;
echo "$output";

# Install the hybrid agent
bash ~/install_linux_azcmagent.sh;

# Run connect command
sudo azcmagent connect --resource-group "$resourceGroup" --tenant-id "$tenantId" --location
"$location" --subscription-id "$subscriptionId" --cloud "$cloud" --correlation-id
"$correlationId";

```

Connect to your Linux server and upload the script and then run it



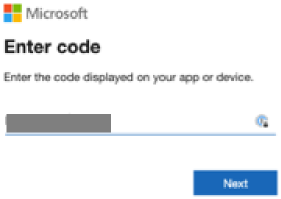
```
vectra@ama-demo-doc: ~
vectra@ama-demo-doc:~$ ls
OnboardingScript.sh
vectra@ama-demo-doc:~$ bash OnboardingScript.sh █
```

When prompted, navigate to the URL provided and enter the code to authenticate



```
Latest version of azcmagent is installed.
INFO    Connecting machine to Azure... This might take a few minutes.
INFO    Testing connectivity to endpoints that are needed to connect to Azure... This might take a few minutes.
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code [redacted] to authenticate.
```

Enter the code

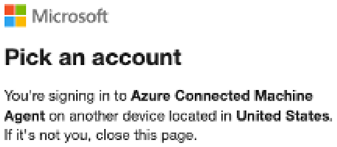


Enter code

Enter the code displayed on your app or device.

[redacted]

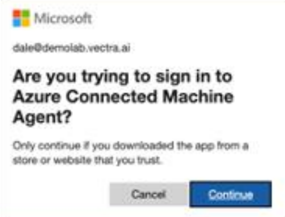
Select the account to authorize the agent with



Pick an account

You're signing in to **Azure Connected Machine Agent** on another device located in **United States**. If it's not you, close this page.

Authorize the agent

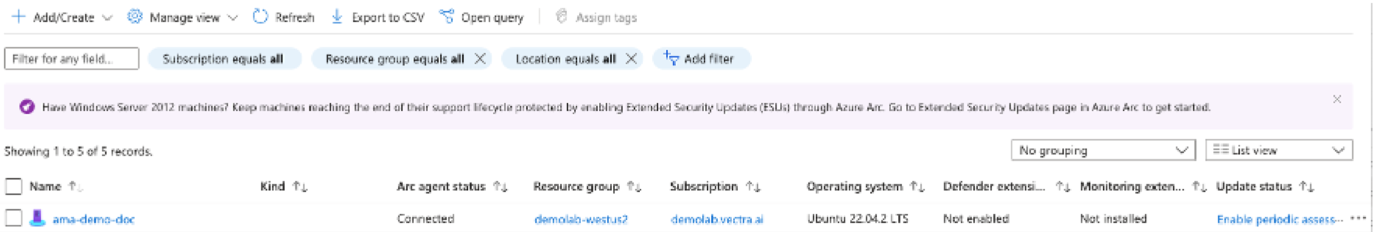


dale@demolab.vectra.ai

Are you trying to sign in to Azure Connected Machine Agent?

Only continue if you downloaded the app from a store or website that you trust.

Refresh the machines list and verify machine is present

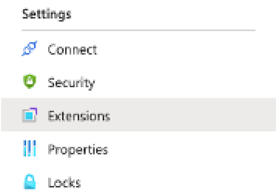


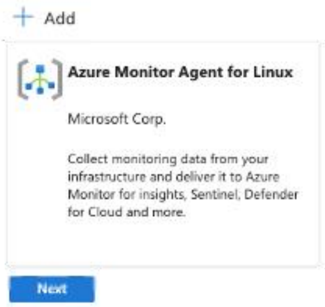
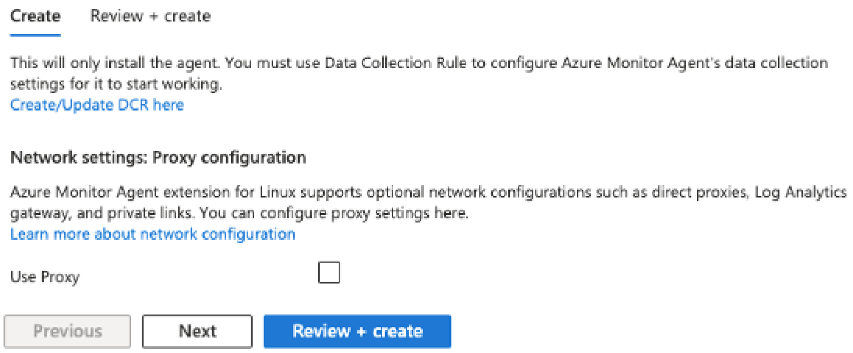
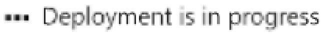
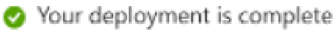
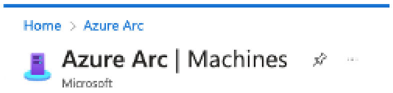
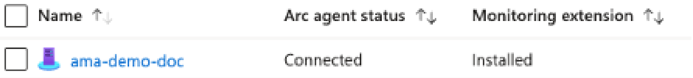
Have Windows Server 2012 machines? Keep machines reaching the end of their support lifecycle protected by enabling Extended Security Updates (ESUs) through Azure Arc. Go to Extended Security Updates page in Azure Arc to get started.

Showing 1 to 5 of 5 records.

Name	Kind	Arc agent status	Resource group	Subscription	Operating system	Defender extensi...	Monitoring exten...	Update status
ama-demo-doc		Connected	demolab-westus2	demolab.vectra.ai	Ubuntu 22.04.2 LTS	Not enabled	Not installed	Enable periodic assess...

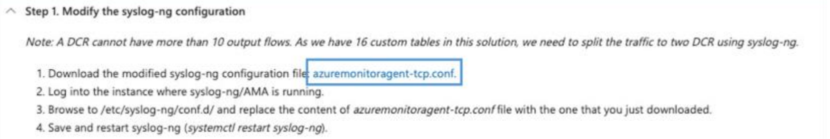
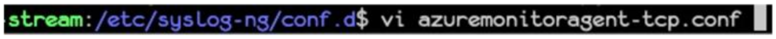
Select the machine and then select Extensions from the left-hand menu



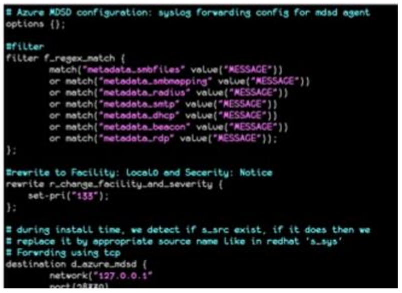
Select Add then select Azure Monitor Agent for Linux and click Next	 + Add **Azure Monitor Agent for Linux** Microsoft Corp. Collect monitoring data from your infrastructure and deliver it to Azure Monitor for insights, Sentinel, Defender for Cloud and more. Next
Select Review + create And then Create	 Create Review + create This will only install the agent. You must use Data Collection Rule to configure Azure Monitor Agent's data collection settings for it to start working. Create/Update DCR here **Network settings: Proxy configuration** Azure Monitor Agent extension for Linux supports optional network configurations such as direct proxies, Log Analytics gateway, and private links. You can configure proxy settings here. Learn more about network configuration Use Proxy □ Previous Next Review + create
Deployment will start	 ··· Deployment is in progress
Wait several minutes for deployment to complete	 ✓ Your deployment is complete
Navigate back to Azure Arc - Machines	 Home > Azure Arc Azure Arc | Machines Microsoft
Verify Arc agent status is Connected, and the Monitoring Extension is installed	 □ Name ↑↓ | Arc agent status ↑↓ | Monitoring extension ↑↓ □ ama-demo-doc | Connected | Installed

Modify Syslog-NG Configuration

Since a data collection rule can't have more than ten output flows, we need to direct the traffic to two different data collections by using syslog-ng.

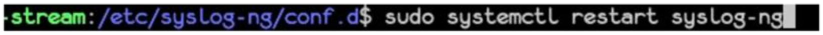
Download the syslog-ng configuration file	 **Step 1. Modify the syslog-ng configuration** *Note: A DCR cannot have more than 10 output flows. As we have 16 custom tables in this solution, we need to split the traffic to two DCR using syslog-ng.* 1. Download the modified syslog-ng configuration file: azuremonitoragent-tcp.conf. 2. Log into the instance where syslog-ng/AMA is running. 3. Browse to /etc/syslog-ng/conf.d/ and replace the content of *azuremonitoragent-tcp.conf* file with the one that you just downloaded. 4. Save and restart syslog-ng (*systemctl restart syslog-ng*).
SSH into your Azure ARC enabled Linux server running AMA and edit /etc/syslog-ng/conf.d/azuremonitoragent-tcp.conf	 `stream:/etc/syslog-ng/conf.d$ vi azuremonitoragent-tcp.conf`

Replace the contents of the conf file with the contents from the downloaded file and save the file



```
# Azure MDSD configuration: syslog forwarding config for mdsd agent
options {};

#filter
filter f_regex_match {
        match("metadata_smbfiles" value("MESSAGE"))
        or match("metadata_smbmapping" value("MESSAGE"))
        or match("metadata_radius" value("MESSAGE"))
        or match("metadata_smtp" value("MESSAGE"))
        or match("metadata_dhcp" value("MESSAGE"))
        or match("metadata_beacon" value("MESSAGE"))
        or match("metadata_rdp" value("MESSAGE"));
};

#rewrite to Facility: Local0 and Secerity: Notice
rewrite r_change_facility_and_severity {
    set-pri("133");
};

# during install time, we detect if s_src exist, if it does then we
# replace it by appropriate source name like in redhat 's_sys'
# Forwrding using tcp
destination d_azure_mdsd {
        network("127.0.0.1"
        port(28330)
```

Restart syslog-ng

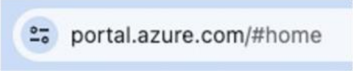


```
-stream:/etc/syslog-ng/conf.d$ sudo systemctl restart syslog-ng
```

Deploy Vectra Stream App

The Vectra Stream app is required to be installed into the Sentinel workspace where the Vectra Stream metadata should reside. This app must be installed even if there is an existing integration using OMS.

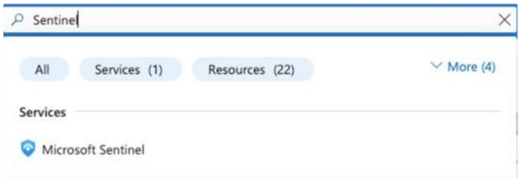
Access the Azure Portal



portal.azure.com/#home

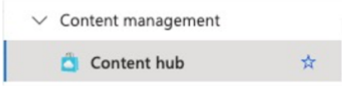
Search resources for Sentinel

Select Microsoft Sentinel from the list then enter your workspace

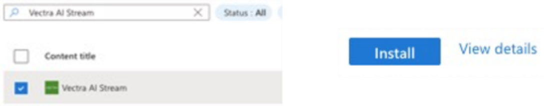


Sentinel

Under Content management, navigate to Content hub

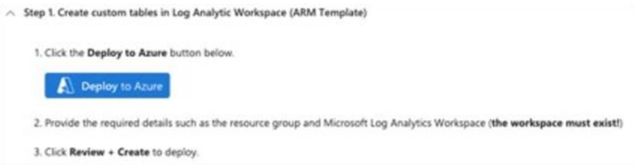


Search for Vectra AI Stream and install it



Vectra AI Stream

Select Deploy to Azure and follow the on-screen instructions



Step 1. Create custom tables in Log Analytic Workspace (ARM Template)

1. Click the **Deploy to Azure** button below.

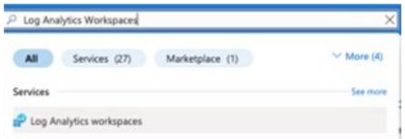
2. Provide the required details such as the resource group and Microsoft Log Analytics Workspace (**the workspace must exist!**)

3. Click **Review + Create** to deploy.

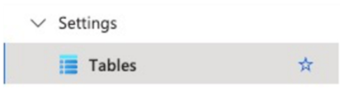
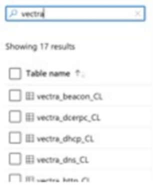
Verify Custom Tables

After the Vectra Stream for Azure Sentinel connector is deployed, the custom tables that are used to host the Stream metadata should all be created.

From the Azure Portal search bar search for and access Log Analytics workspaces

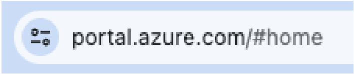
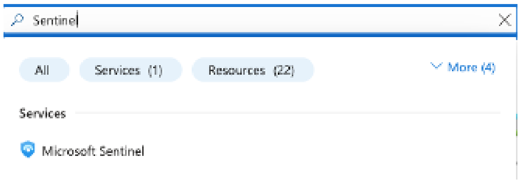
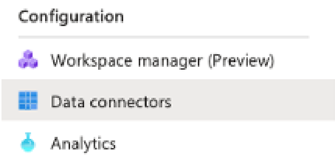
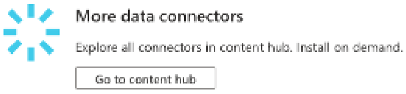
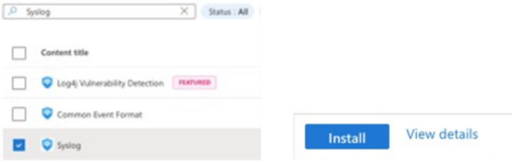
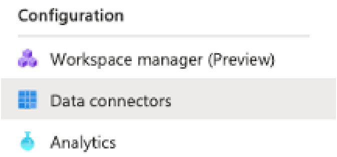
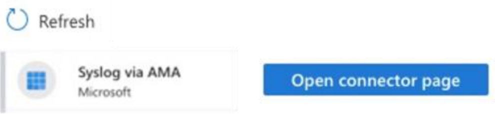


Log Analytics Workspaces

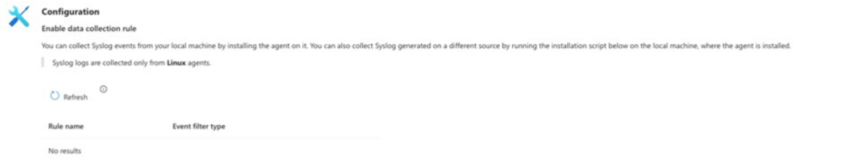
Select the log space you are installing Stream to then select Settings - Tables	 Settings Tables
Filter by string 'vectra' and verify that 17 tables are present	 vectra Showing 17 results □ Table name ↑ □ vectra_beacon_CL □ vectra_dcerpc_CL □ vectra_dhcp_CL □ vectra_dns_CL □ vectra_http_CL

Install Syslog via AMA Data Connector

The data connector is responsible for controlling how data is shipped from AMA to the Log Analytics Workspace. Vectra Stream data is provided in syslog JSON format and is stored in custom tables, so that means a syslog connector is required.

Access the Azure Portal	 portal.azure.com/#home
Search resources for Sentinel Select Microsoft Sentinel from the list then enter your workspace	 Sentinel All · Services (1) · Resources (22) · More (4) Services Microsoft Sentinel
Select Data connectors from under the Configuration menu	 Configuration Workspace manager (Preview) Data connectors Analytics
Go to content hub to install a connector on demand	 More data connectors Explore all connectors in content hub. Install on demand. Go to content hub
Search for Syslog and select the one from Microsoft with category IT Operations	 Syslog · Status : All □ Content title □ Log4j Vulnerability Detection FEATURED □ Common Event Format ☑ Syslog Install · View details
Return to the Data connectors page	 Configuration Workspace manager (Preview) Data connectors Analytics
Refresh the page and select Syslog via AMA and then Open connector page	 Refresh Syslog via AMA Microsoft Open connector page

Verify that there are no existing Data Collection Rules in place and create a new data collection



Configuration

Enable data collection rule

You can collect Syslog events from your local machine by installing the agent on it. You can also collect Syslog generated on a different source by running the installation script below on the local machine, where the agent is installed.

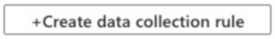
Syslog logs are collected only from **Linux** agents.

Rule name	Event filter type
No results	

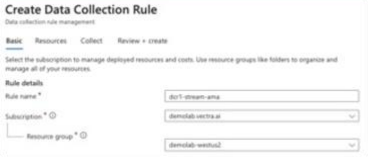
Create Data Collection Rule 1

Create the first data collection rule that will be further customized before deploying to the AMA server.

Create the 1st new data collection rule



Provide a meaningful name for the 1st DCR and use appropriate subscription and resource group and



Create Data Collection Rule

Data collection rule management

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

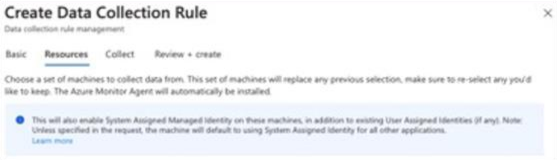
Rule details

Rule name *: dcr1-stream-ama

Subscription *: demolab.vectra.ai

Resource group *: demolab-westus2

Select the required Azure Arc enabled resource from the resources tab



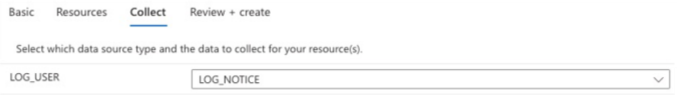
Create Data Collection Rule

Data collection rule management

Choose a set of machines to collect data from. This set of machines will replace any previous selection, make sure to re-select any you'd like to keep. The Azure Monitor Agent will automatically be installed.

This will also enable System Assigned Managed Identity on these machines, in addition to existing User Assigned Identities (if any). Note: Unless specified in the request, the machine will default to using System Assigned Identity for all other applications.

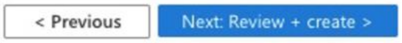
DCR1 requires just LOG_USER set to LOG_NOTICE for collection



Select which data source type and the data to collect for your resource(s).

LOG_USER: LOG_NOTICE

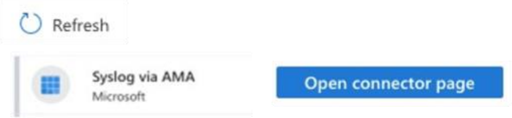
Review + Create and then Create the 1st DCR



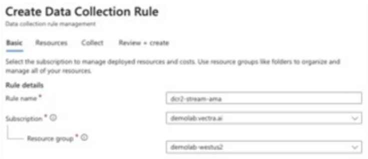
Create Data Collection Rule 2

Create the second data collection rule that will be further customized before deploying to the AMA server.

Go back into Syslog via AMA connector to create DCR2



Provide a meaningful name for the 2nd DCR and use appropriate subscription and resource group and



Create Data Collection Rule

Data collection rule management

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

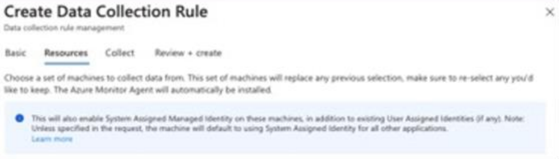
Rule details

Rule name *: dcr2-stream-ama

Subscription *: demolab.vectra.ai

Resource group *: demolab-westus2

Select the required Azure Arc enabled resource from the resources tab



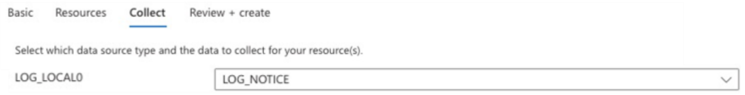
Create Data Collection Rule

Data collection rule management

Choose a set of machines to collect data from. This set of machines will replace any previous selection, make sure to re-select any you'd like to keep. The Azure Monitor Agent will automatically be installed.

This will also enable System Assigned Managed Identity on these machines, in addition to existing User Assigned Identities (if any). Note: Unless specified in the request, the machine will default to using System Assigned Identity for all other applications.

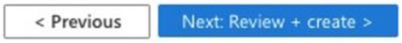
DCR2 requires just LOG_LOCAL0 set to LOG_NOTICE for collection



Select which data source type and the data to collect for your resource(s).

LOG_LOCAL0: LOG_NOTICE

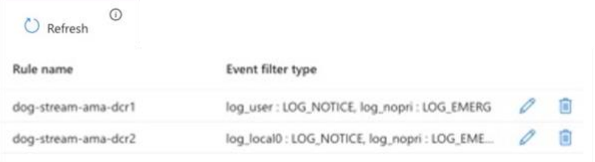
Review + Create and then Create the 2nd DCR



Verify Data Collection Rules

Next, verify that both data collection rules have been created and the event filter type looks accurate.

Verify the rules exist by going back into the data connector and hitting refresh

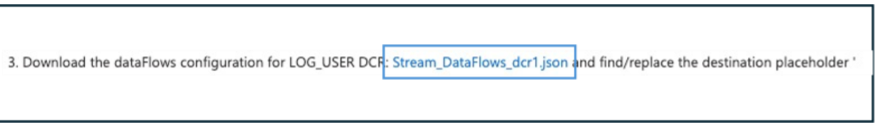


Rule name	Event filter type
dog-stream-ama-dcr1	log_user : LOG_NOTICE, log_nopri : LOG_EMERG
dog-stream-ama-dcr2	log_local0 : LOG_NOTICE, log_nopri : LOG_EME...

Modify Stream JSON File

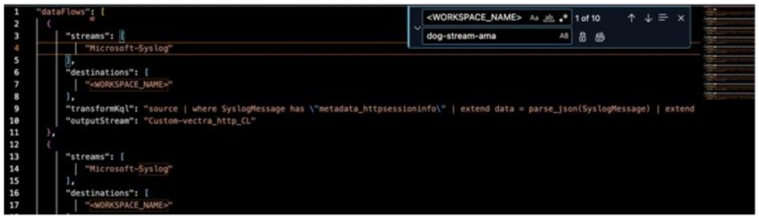
We need to prepare the json file that will be used in the data collection rule to use the desired Sentinel Workspace. There are two stream json files (one per DCR) that will need to be modified, but proceed with one at a time and revisit these instructions when it’s time to work on the second DCR.

Download ‘Stream_DataFlows_dcr1.json’ from the content hub link



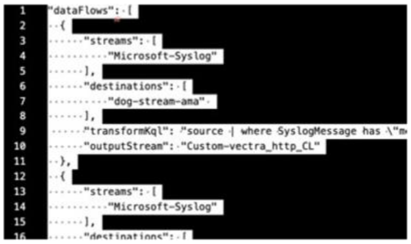
3. Download the dataFlows configuration for LOG_USER DCR: Stream_DataFlows_dcr1.json and find/replace the destination placeholder '

Open the file in an editor and use search/replace to replace all ten instances of <WORKSPACE_NAME> with the name of your Sentinel Workspace and SAVE the file



```
"dataFlows": [
  {
    "streams": [
      "Microsoft-Syslog"
    ],
    "destinations": [
      "<WORKSPACE_NAME>"
    ],
    "transformKql": "source | where SyslogMessage has \"metadata_httpsessioninfo\" | extend data = parse_json(SyslogMessage) | extend
    "outputStream": "Custom-vectra_http_CL"
  },
  {
    "streams": [
      "Microsoft-Syslog"
    ],
    "destinations": [
      "<WORKSPACE_NAME>"
```

Select the entire contents of the file and copy the entire contents into your clipboard as it will be needed during the next section

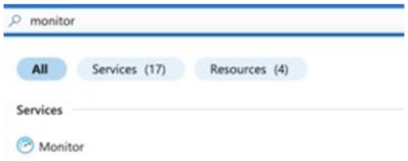


```
"dataFlows": [
  {
      "streams": [
          "Microsoft-Syslog"
      ],
      "destinations": [
          "dog-stream-ama"
      ],
      "transformKql": "source | where SyslogMessage has \"m
      "outputStream": "Custom-vectra_http_CL"
  },
  {
      "streams": [
          "Microsoft-Syslog"
      ],
      "destinations": [
```

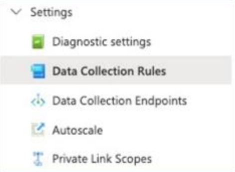
Customize Data Collection Rule 1

We need to modify our first data collection rule to send each data flow (flows 1-10) to the correct custom table based on its metadata type. This is accomplished by replacing the default dataFlows stanza of the DCR with the content just prepared.

Within your Azure Portal search for monitor and select the Monitor service

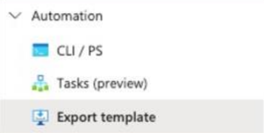


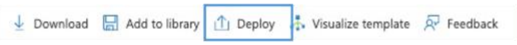
Navigate to Settings – Data Collection Rules



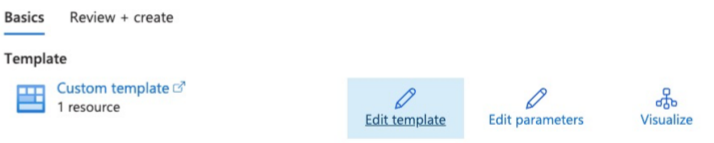
Click on the name of your first DCR so that you can work on it

Navigate to Automation – Export Template and then click Deploy





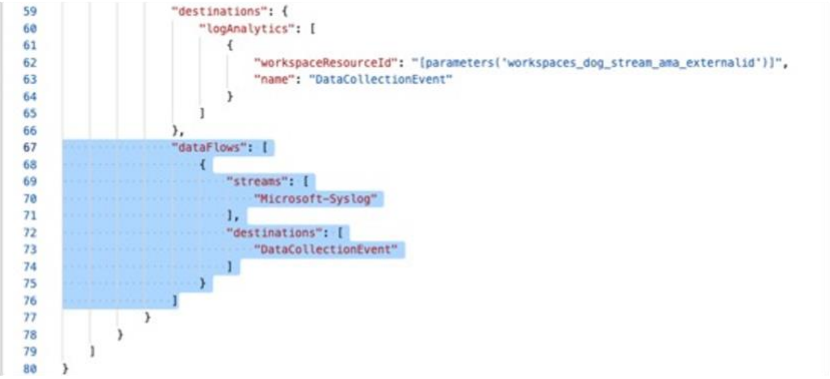
We need to edit the template to make our changes before deploying it



Locate the ”dataFlows” stanza that should start on or around row 67.

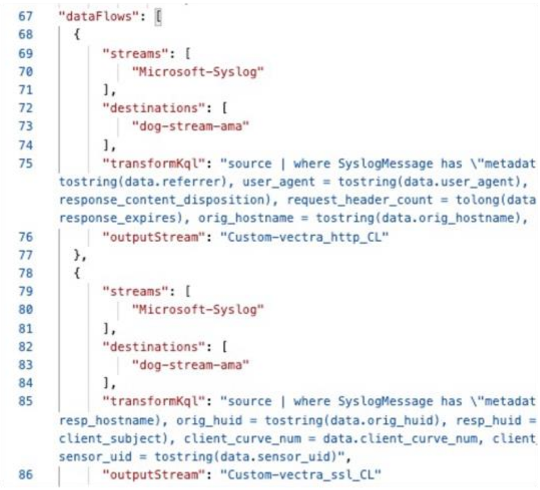
Highlight the stanza as shown (be cautious to start at “dataFlows” row and end at the closing].

Paste the data from your clipboard to overwrite this stanza.



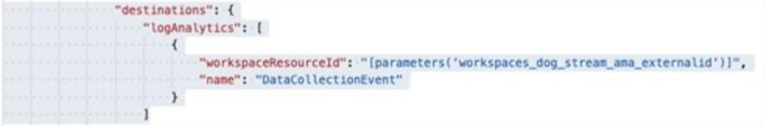
```
59                  "destinations": {
60                      "logAnalytics": [
61                          {
62                              "workspaceResourceId": "[parameters('workspaces_dog_stream_ama_externalid')]",
63                              "name": "DataCollectionEvent"
64                          }
65                      ]
66                  },
67                  "dataFlows": [
68                      {
69                          "streams": [
70                              "Microsoft-Syslog"
71                          ],
72                          "destinations": [
73                              "DataCollectionEvent"
74                          ]
75                      }
76                  ]
77              }
78          }
79      ]
80  }
```

When you paste the clipboard data in, the formatting may look strange, but it will be formatted properly once saved.



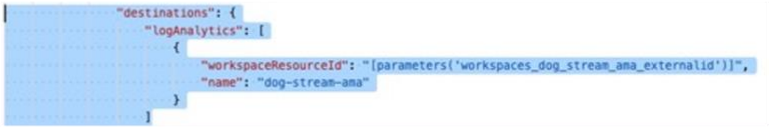
```
67  "dataFlows": [
68    {
69        "streams": [
70            "Microsoft-Syslog"
71        ],
72        "destinations": [
73            "dog-stream-ama"
74        ],
75        "transformKql": "source | where SyslogMessage has \"metadat
    tostring(data.referrer), user_agent = tostring(data.user_agent),
    response_content_disposition), request_header_count = tolong(data
    response_expires), orig_hostname = tostring(data.orig_hostname),
76        "outputStream": "Custom-vectra_http_CL"
77    },
78    {
79        "streams": [
80            "Microsoft-Syslog"
81        ],
82        "destinations": [
83            "dog-stream-ama"
84        ],
85        "transformKql": "source | where SyslogMessage has \"metadat
    resp_hostname), orig_huid = tostring(data.orig_huid), resp_huid =
    client_subject), client_curve_num = data.client_curve_num, client
    sensor_uid = tostring(data.sensor_uid)",
86        "outputStream": "Custom-vectra_ssl_CL"
```

In the same template, locate the destinations block that has the LogAnalytics name “DataCollectionEvent”



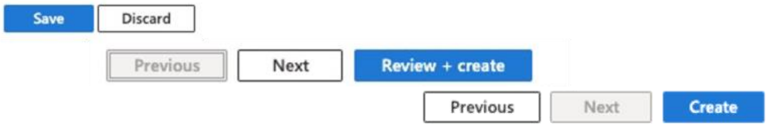
```
"destinations": {
    "logAnalytics": [
        {
            "workspaceResourceId": "[parameters('workspaces_dog_stream_ama_externalid')]",
            "name": "DataCollectionEvent"
        }
    ]
```

Replace DataCollectionEvent with the same name used previously when modifying your workspace name in ‘Stream_DataFlows_dcr1.json’



```
"destinations": {
    "logAnalytics": [
        {
            "workspaceResourceId": "[parameters('workspaces_dog_stream_ama_externalid')]",
            "name": "dog-stream-ama"
        }
    ]
```

Select Save, then Review + create, then Create to deploy the edited template



Modify and Customize DCR2

We need to modify our second data collection rule to send each data flow (flows 11-17) to the correct custom table based on its metadata type. Repeat the documented steps above (Modify Stream_DataFlows_dcr1 JSON File and Customize Data Collection Rule 2) using data collection rule 2 data. The summary of the configuration is provided here:

- Download Stream_DataFlows_dcr2.json and edit it.
- Replace all seven instances of <WORKSPACE_NAME> with your Sentinel Workspace name and copy the updated contents to your clipboard.
- Open Azure Monitor and navigate to Settings – Data Collection Rules and find your second DCR.
- Navigate to Automation – Export Template and then select Deploy and Edit template.
- Replace the DataFlows stanza starting around line 67 with the content in your clipboard.
- Replace the string “DataCollectionEvent” with your Sentinel Workspace name.
- Save, Review + create, then Create.

Deploy Data Collection Rules

To deploy the customized data collections rules, SSH into your Azure ARC enabled Linux server that is running AMA and run the following command:

```
sudo wget -O Forwarder_AMA_installer.py https://raw.githubusercontent.com/Azure/Azure-
Sentinel/master/DataConnectors/Syslog/Forwarder_AMA_installer.py&&sudo python3 Forwarder_AMA_installer.py
```

Configure Vectra

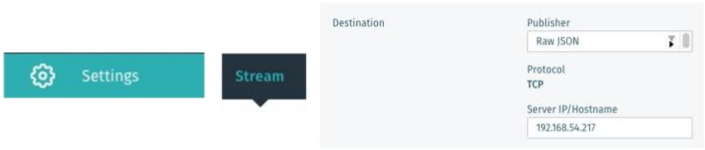
The Vectra platform must be configured to send the Stream data to the AMA Linux server. Connect to your Vectra user interface to complete this configuration.

If this is an existing OMS deployment, then the configuration must be updated to point to the AMA Linux server configured above.

Additional details for deploying Vectra Stream are available in the following knowledge base article.

https://support.vectra.ai/s/article/KB-VS-1189

Configure Vectra > Settings – Stream to send Raw JSON over TCP to your Azure Arc enabled Linux AMA server



Publisher: Raw JSON
Protocol: TCP
Server IP/Hostname: 192.168.54.217

Validation

Everything should be configured at this point and should be validated to ensure data is being ingested properly.

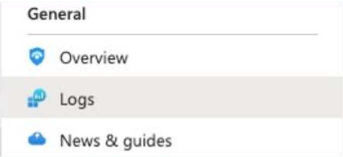
- Connect to your AMA Linux server over ssh and then tail your syslog file to make sure data is coming in.

```
tail -f /var/log/syslog
```

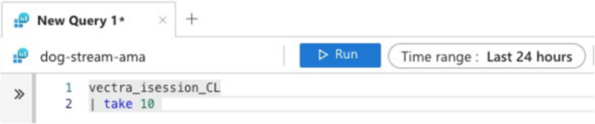
IMPORTANT: It can take 20 minutes for the initial data to make it into the system and then approximately every five minutes afterwards so please be patient.

- Navigate back to your Sentinel instance to continue validation.

Within Sentinel, navigate to the Logs page



Run this KQL and verify that there are records in that table



Time range : Last 24 hours

```
vectra_isession_CL
| take 10
```

Congratulations! Vectra data is now being ingested into Microsoft Sentinel using the Azure Monitoring Agent.

Disable OMS Agent

For existing deployments, the OMS agent should be disabled to prevent duplicate data from being ingested into the Sentinel workspace.

- Connect to your OMS Linux server using ssh and disable the agent.

  ```
  sudo /opt/microsoft/omsagent/bin/service_control disable
  ```

- Refer to Microsoft documentation for complete instructions for removing the OMS agent. The following link includes a section on removing the agent.

  https://learn.microsoft.com/en-us/troubleshoot/azure/automation/reinstall-oms-agent-linux

- While it is safe to delete the existing Vectra AI Detect data connector if you do so you will need to reinstall the Vectra workbook. Please refer to the instructions earlier in this document.