AZURE LOG FORWARDING INTEGRATION

## Prerequisites

Before proceeding with the integration, ensure the following components are already set up in your Azure environment:

* **API Access**: Service principal or user credentials with sufficient permissions to read and modify relevant Azure resources.
* **Microsoft.Web**: The **Microsoft.Web** resource provider must be registered (used for Azure App Services and related components).
* **Microsoft.Insights**: The **Microsoft.Insights** resource provider must be registered (used for monitoring and diagnostics, such as Application Insights or Log Analytics).
* **Event Hub Namespace:** An existing Event Hub namespace to host the Event Hub.
* **Event Hub:** An Event Hub within the namespace to receive data forwarded from Azure.
* **Log Analytics Workspace:** Required to collect and analyze log data.
* **Application Insights:** Used for application monitoring and telemetry.
* **Storage Account:** Required to support Azure Function App operations (e.g., runtime storage, diagnostics).
* **Subnet Flow Log:** To capture network traffic data.
* **DNS Security Policy:** Required for advanced DNS logging

## Required Variables

To successfully configure the integration, gather the following details from both your **Azure environment** and your **Celerium account**:

**Azure Environment:**

* Subscription Name or ID
* Resource Group Name
* Location
* Function App Name
* Event Hub Name
* Event Hub Connection String
* Protocol

**Celerium Account:**

* Syslog Server
* Syslog Port

For additional configuration options and optional variables, refer to the [readme file](https://github.com/Celerium-Inc/AzureNetworkFirewallSupport/blob/main/Event%20Hub%20Syslog%20Forwarder/README.md) provided in the GitHub repository.

## Data Retrieval

Before deploying the integration, you need to collect specific information from both your **Azure environment** and the **Celerium portal**. Follow the steps below:

* **Subscription Name or ID**

To locate your Azure subscription details:

1. In the **Azure Portal**, navigate to **Subscriptions**.
2. Ensure that no filters (e.g., directories or tags) are applied so all subscriptions are visible.
   * + A screenshot of a chat

       AI-generated content may be incorrect.
   1. Identify and **note** the **Subscription Name** or **Subscription ID** you intend to use for the integration.

* **Resource Group Name**
  1. In the **Azure Portal**, go to **Resource groups**.
  2. Locate and **note** the name of the **Resource Group** where your resources (e.g., Function App, Firewall) are or will be deployed.
* **Location**
  1. In the **Azure Portal**, open the **Resource Group** you selected earlier.
  2. The **Location** (Azure region) will be listed at the top of the overview pane (e.g., eastus, westeurope).
  3. **Note** the location for use in your deployment.
* **Function App Name**
  1. Choose a name for your **Function App** that meets the following Azure naming requirements:
     + Must be **1 to 60 characters** long
     + Can include **lowercase letters**, **numbers**, and **hyphens**
     + Must **start and end** with a letter or number
     + Cannot contain **consecutive hyphens**
     + Must be **globally unique** across all of Azure
  2. **Suggested name**: celerium-log-fwd
  3. **Note** the Function App name you’ve selected for use during deployment.
* **Event Hub Name**
  1. In the **Azure Portal**, search for **Event Hubs**.
  2. Click the **Event Hub Namespace** you are using.
  3. Navigate to **Entities → Event Hubs**.
  4. Note the **name** of your **Event Hub**.
* **Event Hub Connection String**
  1. From the **Event Hub** identified above:
     + Click the **Event Hub name**.
     + Navigate to **Settings → Shared access policies**.
     + Locate the policy named **PreviewDataPolicy**.
       - **If this policy does not exist**, create one:
         * Click **+ Add.**
         * Policy name : **ListenOnly.**
         * Select **Listen** permission.
         * Click **Create.**
  2. Click on the **ListenOnly** policy (or equivalent).
  3. Copy the **Primary Connection String** and save it for later use.
     + A screenshot of a computer

       AI-generated content may be incorrect.
* **Protocol**
  1. Select the protocol you will use for forwarding:
     + **UDP** or **SSL**.
  2. Note your chosen **protocol**.
* **Celerium Syslog Server**
  1. Log in to the **Celerium portal**.
  2. Locate your **assigned syslog server**.
  3. Note the **syslog server address**.
* **Celerium Syslog Port**
  1. In the **Celerium portal**, locate your **assigned syslog port**.
  2. Note the **port number**.

## Preparing for Deployment

There are several ways to deploy this integration, such as:

* **Azure Cloud Shell** (recommended – no local installation required)
* **Local machine** with the Azure PowerShell module installed
* **Visual Studio Code** with Azure extensions

This guide uses **Azure Cloud Shell**, which is accessible directly from the Azure Portal.

**Accessing Azure Cloud Shell**

You can access **Azure Cloud Shell** directly from the Azure Portal by clicking the **Cloud Shell** icon in the top navigation bar. Cloud Shell supports both **PowerShell** and **Bash** environments.

**Browser Compatibility Recommendation**  
For the most stable experience with Azure Cloud Shell:

* Use the **Firefox** browser, which has shown the **least number of connectivity and rendering issues** during deployment and scripting workflows.
* While **Chrome** and **Edge** are supported, Firefox performs better with shell responsiveness and file uploads.

**Troubleshooting Tips**  
If you encounter issues like:

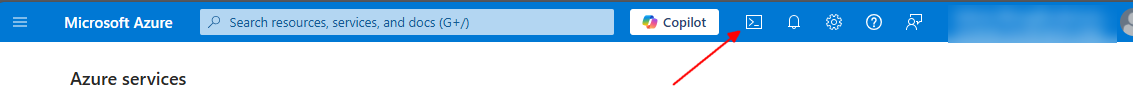
* Cloud Shell becoming **unresponsive**
* The **prompt not scrolling** or **overwriting previous lines**

**Try the following:**

1. Click **Restart** in the Cloud Shell window.
2. If the issue persists, **close and reopen** Cloud Shell from the portal.

**Getting Started with Azure Cloud Shell**

1. Log in to your **Azure Portal** at <https://portal.azure.com>.
2. Click the **Cloud Shell icon** (command prompt symbol) in the top-right of the portal, as shown below:



1. A **PowerShell prompt (PS>)** will appear at the bottom of your browser window.

A screenshot of a computer

AI-generated content may be incorrect.

* If Bash is shown, click **"Switch to PowerShell"** in the upper-left corner.

A screen shot of a computer

AI-generated content may be incorrect.

**Prepare Your Environment**

1. Have the **environment variables** you gathered earlier on hand — you’ll need them during deployment.
2. Download the integration files from **GitHub** to your local machine.

* **Important**: Maintain the **original folder structure** as shown in the GitHub repository.

**Set the Correct Azure Subscription**

1. Verify that you're in the correct subscription before proceeding:

* To **check your current subscription**:
  1. **Get-AzContext**

A blue screen with green text

AI-generated content may be incorrect.

1. To **switch subscriptions**, use either the name or the ID:

* By name
  + **Set-AzContext -Subscription "your subscription name"**
* By ID
  + **Set-AzContext -SubscriptionId "your-subscription-guid-here"**

A screenshot of a computer program

AI-generated content may be incorrect.

1. Confirm the subscription was updated:

* Press the **up arrow twice** to recall **Get-AzContext**, then **press Enter**.
* Confirm the correct subscription is now active.

**Upload Integration Files**

1. Upload the necessary files from GitHub:

* In the Cloud Shell window, click **Manage Files > Upload**.

A screenshot of a computer

AI-generated content may be incorrect.

* Upload **only the files from the src directory** of the **Event Hub Syslog Forwarder.**
* These files will be placed in the /home/admin/ directory by default.

1. Create the required folder structure:

**mkdir ./forward**

**mkdir ./forward/src**

1. Move the uploaded src files into the new folder:

**move-item -path "/home/admin/\*.\*" -destination "/home/admin/forward/src"**

1. Upload the **cleanup.ps1** and **deploy.ps1** scripts from the root of the **Event Hub Syslog Forwarder** directory.
2. Move those scripts to the forward directory:

**move-item -path "/home/admin/\*.\*" -destination "/home/admin/forward"**

## Deploying Forwarder

Follow the steps below to deploy the Log Forwarding integration using your prepared environment variables and the deployment script.

**Prepare the Deployment Command**

Using the variables you gathered earlier, build your deployment command in a text editor (e.g., Notepad). Use the format below as a reference.

Example:

**./forward/deploy.ps1 `**

**-ResourceGroupName "your-rg" `**

**-Location "your-loc" `**

**-FunctionAppName "your-func-name" `**

**-SyslogServer "syslog.example.com" `**

**-SyslogPort 6514 `**

**-EventHubName "your-eventhub" `**

**-EventHubConnection "your-connection-string" `**

**-Protocol "SSL"**

For more information and other optional variables (i.e. Gov cloud environment), see the [readme file](https://github.com/Celerium-Inc/AzureNetworkFirewallSupport/blob/main/Event%20Hub%20Syslog%20Forwarder/README.md) on GitHub.

**Run the Deployment Script**

1. **Copy** the command you created and **paste it into the Azure Cloud Shell PowerShell prompt**.

A blue screen with white text

AI-generated content may be incorrect.

1. When prompted, enter **Y** to proceed.

**Review Script Output**

1. After initial execution, you will see progress messages.

A blue screen with yellow and green text

AI-generated content may be incorrect.

1. You may be prompted again — enter **Y** to continue.
2. Once the Azure resources are deployed, you'll see additional output.

A blue background with white and red dots

AI-generated content may be incorrect.

## Post-Deployment

After deploying the log forwarder, follow these steps to verify that the integration is functioning correctly and forwarding data as expected.

### Monitor the Function App

1. In the **Azure Portal**, search for **Function App** in the top search bar.
2. Click on the **Function App** you deployed (based on the name you provided).
3. Click on the function named **EventHubTrigger**.

A screenshot of a computer

AI-generated content may be incorrect.

1. Click on the **Logs** tab to open the logging console.
   * The logs should indicate that the function successfully **connected** to the Event Hub.

A screenshot of a computer

AI-generated content may be incorrect.

**Wait for Log Ingestion**

* The function may take **up to 15 minutes** to begin receiving and displaying log data.
* Once the function executes, the **log screen will populate with output** from processed messages.

**Verify Data in the Celerium Portal**

* Open the **Celerium portal**.
* Navigate to your assigned sensor.
* A **green sensor status** means data is being received successfully.

## Removing the Function App

If you need to remove the log forwarding integration, follow the steps below. This process deletes the deployed Function App and associated resources created during deployment.

**Note:** Before proceeding, ensure you have followed the steps in the **"Preparing for Deployment"** section to access Azure Cloud Shell and gather the required variables.

**Prepare the Cleanup Command**

1. Open a text editor (e.g., Notepad).
2. Using the same environment variable values you collected earlier, build your cleanup command in the format below:

Example:

./forward/cleanup.ps1 `

-ResourceGroupName "your-rg" `

-FunctionAppName "your-func-name"

For more information and other optional variables (i.e. Gov cloud environment), see the [readme file](https://github.com/Celerium-Inc/AzureNetworkFirewallSupport/blob/main/Event%20Hub%20Syslog%20Forwarder/README.md) on GitHub.

**Run the Cleanup Script**

1. Copy the command you created and paste it into the Azure Cloud Shell PowerShell prompt.

A screen shot of a computer program

AI-generated content may be incorrect.

1. When prompted, enter **Y** to confirm and proceed.

**Verify Cleanup Completion**

* Once the script finishes, you should see confirmation output in the shell indicating that the Function App and related resources have been successfully removed.

A blue screen with white text

AI-generated content may be incorrect.

* This process fully removes the log forwarding integration from your Azure environment.