ZSCALER AND MICROSOFT SENTINEL DEPLOYMENT GUIDE

DECEMBER 2025, VERSION 1.21

BUSINESS DEVELOPMENT GUIDE

Contents

Terms and Acronyms 9

Trademark Notice 10

About This Document 11

- Zscaler Overview 11
- Audience 11
- Software Versions 11
- Request for Comments 11

Zscaler and Microsoft Introduction 12

- ZIA Overview 12
- Zscaler Resources 12
- Microsoft Sentinel Overview 13
- Microsoft Sentinel Resources 13
- Document Prerequisites 14

Syslog, CEF, and LEEF 15

- Syslog 15
- CEF 16
- LEEF 16

Zscaler Logging Architecture 17

- NSS 17
- About Cloud-To-Cloud Log Streaming 18
- ZIA Log Feeds 18

Configuring Cloud NSS-Based Log Ingestion for ZIA 19

- In the Azure Portal, Register an Entra ID Application, and Create a Client Secret 19
- In the Azure Portal, Create a Log Analytics Workspace, and Add Microsoft Sentinel to the Workspace 22

©2026 Zscaler, Inc. All rights reserved.

Deploy a Data Collection Rule and Data Collection Endpoint Using a Bicep Template 25
In the Azure Portal, Assign Permissions to the Data Collection Rule 28
Configuring ZIA for Cloud NSS-Based Log Export 32
In the ZIA Admin Portal, Add a Cloud NSS Feed 32
Cloud NSS Feeds for Web Logs 33
Cloud NSS Feeds for Firewall Logs 35
Cloud NSS Feeds for DNS Logs 36
Cloud NSS Feeds for Admin Audit Logs 37
Cloud NSS Feeds for Tunnel Logs 38
Cloud NSS Feeds for Endpoint DLP Logs 40
Cloud NSS Feeds for Email DLP Logs 42
Cloud NSS Feeds for SaaS Security Activity Logs 44
Cloud NSS Feeds for SaaS Security Logs 45
Cloud NSS Feeds for Cloud & Branch Connector Session Logs 59
Cloud NSS Feeds for Cloud & Branch Connector Event Logs 61
Cloud NSS Feeds for Cloud & Branch Connector DNS Logs 63
Using PowerShell, Verify Log Data Arrives at the Endpoint 65
In Microsoft Sentinel, View Log Details 68
Add the ZIA Content Pack to Install ZIA Workbooks, Analytics Rules, and Parsers 69
Examples of the Workbooks 72
Configuring NSS VM-Based Log Ingestion for ZIA 75
Logging In to Azure Portal 75
Deploying the Data Connector Host VM 76
Bind this VM to a Resource Group 78
Allowing Inbound Ports 79
Add Inbound Security Rule for Syslog 81
Add Inbound Security Rule for SSH 82

©2026 Zscaler, Inc. All rights reserved.

Configuring ZIA for NSS VM-Based Log Export 83
Logging In to ZIA 83
Configuring NSS 83
Verify NSS Server State 85
Add NSS Feed 85
Configure NSS Feed 86
Edit NSS Feed (Admin Audit) 87
Edit NSS Feed (SaaS Security Activity) 87
Edit NSS Feed (SaaS Public Cloud Storage) 87
SaaS Security Collaboration 88
Edit NSS Feed (SaaS Security CRM) 88
Edit NSS Feed (SaaS Security Email) 89
Edit NSS Feed (SaaS Security File) 89
Edit NSS Feed (SaaS Security ITSM) 90
Edit NSS Feed (SaaS Security Repository) 91
Edit NSS Feed (DNS) 91
Edit NSS Feed (Endpoint DLP) 92
Edit NSS Feed (Email DLP) 92
Edit NSS Feed (Firewall) 93
Edit NSS Feed (Tunnel) 93
Edit NSS Feed (Web) 94
Edit NSS Feed (Cloud Connector—Session) 95
Edit NSS Feed (Cloud Connector—Event) 95
Edit NSS Feed (Cloud Connector—DNS) 96
Edit NSS Feed (Cloud Connector—Metrics) 96
Activate Changes 97
Create a Log Analytics Workspace, and Add Microsoft Sentinel to the Workspace 98
Install the Common Event Format Content Pack 100

©2026 Zscaler, Inc. All rights reserved.

Configure the VM Connector to Receive Logs 108
View Log Details 109
Add the ZIA Content Pack to Install ZIA Workbooks, Analytics Rules, and Parsers 110
Examples of the Workbooks 114
Configuring Log Ingestion for ZPA 116
Logging In to Azure Portal 116
Deploying the Data Connector Host VM 117
Create Virtual Machine 117
Bind this VM to a Resource Group 118
Allowing Inbound Ports 119
Add Inbound Security Rule for Syslog 121
Add Inbound Security Rule for SSH 122
Create and Configure Sentinel Instance 123
Create Log Analytics Workspace 123
Name, Add, and Link a Resource Group to a Workspace 123
Add Microsoft Sentinel Workspace 124
Configure Data Collection 124
Configure Azure Monitor Agent on the Linux Virtual Machine 129
Create a New ZPA_CL Custom Log Table in Your Log Analytics Workspace 130
Modify the DCR Using a PowerShell Script to Send Logs to the ZPA_CL Table 132
Configure ZPA User Activity Logs 134
Configure Log Receiver 134
Configure the ZPAUserActivity Log Analytics Workspace Function 137
Configure the ZPAUserActivity Sentinel Workbook 140
Configure ZPA User Status Logs 144
Configure Log Receiver 144
Configure the ZPAUserStatus Log Analytics Workspace Function 147
Configure the ZPAUserStatus Sentinel Workbook 150

©2026 Zscaler, Inc. All rights reserved.

Configure ZPA Private Service Edge Status Logs 154
Configure Log Receiver 154
Configure the ZPAPSEStatus Log Analytics Workspace Function 157
Configure the ZPAPSEStatus Sentinel Workbook 160
Configure ZPA Private Service Edge Metrics Logs 164
Configure Log Receiver 164
Configure the ZPAPSEMetrics Log Analytics workspace function 167
Configure the ZPAPSEMetrics Sentinel Workbook 170
Configure ZPA Browser Access Logs 174
Configure Log Receiver 174
Configure the ZPABrowserAccess Log Analytics Workspace Function 177
Configure the ZPABrowserAccess Sentinel Workbook 179
Configure ZPA Audit Logs 183
Configure Log Receiver 183
Configure the ZPAAudit Log Analytics Workspace Function 186
Configure the ZPAAudit Sentinel workbook 188
Configure ZPA App Protection Logs 192
Configure Log Receiver 192
Configure the App Protection Log Analytics Workspace Function 195
Configure the ZPAAppProtection Sentinel Workbook 197
Configure ZPA App Connector Status Logs 201
Configure Log Receiver 201
Configure the ZPAAppConnectorStatus Log Analytics workspace function 204
Configure the ZPAAppConnectorStatus Sentinel Workbook 207
Configure ZPA App Connector Metrics Logs 211
Configure Log Receiver 211
Configure the ZPAAppConnectorMetrics Log Analytics Workspace Function 214
Configure the ZPAAppConnectorMetrics Sentinel Workbook 216

©2026 Zscaler, Inc. All rights reserved.

Microsoft ASIM Support 220
Key Benefits 220
Resources 220
Using Zscaler ASIM Parser Prerequisites 221
Zscaler ASIM Parsers 221
Using Zscaler ASIM Parsers 222
Microsoft Sentinel Playbooks Overview 224
Microsoft Sentinel Playbooks Resources 224
Document Prerequisites 224
ZIA API Architecture Overview 225
OAuth 2.0 Authentication 225
OAuth 2.0 Prerequisites 226
Benefits of OAuth 2.0 Authentication 226
Configuring a ZIA API Role 227
Logging In to ZIA 227
Create an API Role 227
Configuring Microsoft Entra ID for OAuth 2.0 Authentication 229
Register the ZIA API Client Application 229
Configure Client Credentials 231
Configure the ZIA API Web Service Application 233
Configure the ZIA API Client Application to Access the ZIA API Web Service Application 238
Collect Information for Zscaler OAuth 2.0 Server Configuration 240
Configuring the OAuth2.0 Authorization Server 242
Authenticating an OAuth 2.0 Session in Postman 243
Installing and Configuring Postman for Windows, macOS, or Linux 243
OneAPI Authentication 248
How OneAPI Authentication Works 248
Configuring OneAPI Authentication 249

©2026 Zscaler, Inc. All rights reserved.

Create an API Role 249
Deployment of Zscaler Playbooks for Microsoft Sentinel 252
Giving Permission to Microsoft Sentinel to Run Playbooks 252
Deploying the Zscaler-Oauth2-BlockIP Playbook 256
Deploying the Zscaler-Oauth2-BlacklistURL Playbook 257
Deploying the Zscaler-Oauth2-BlockURL Playbook 258
Deploying the Zscaler-Oauth2-LookupIP Playbook 259
Deploying the Zscaler-Oauth2-LookupSandboxReport Playbook 260
Deploying the Zscaler-Oauth2-LookupURL Playbook 260
Deploying the Zscaler-Oauth2-UnblacklistURL Playbook 262
Deploying the Zscaler-Oauth2-UnblockIP Playbook 264
Deploying the Zscaler-Oauth2-UnblockURL Playbook 265
Deploying the Zscaler-Oauth2-WhitelistURL Playbook 266
Appendix A: Requesting Zscaler Support 267

©2026 Zscaler, Inc. All rights reserved.

Terms and Acronyms

This table defines acronyms used in this deployment guide. When applicable, a Request for Change (RFC) is included in the Definition column for your reference.

Acronym	Definition
AMA	Azure Monitor Agent
ASIM	Advanced Security Information Model
CEF	Common Event Format
GRE	Generic Routing Encapsulation (RFC2890)
JWT	JSON Web Token
KQL	Kusto Query Language
LEEF	Log Event Extended Format
LSS	Log Streaming Service (For ZPA)
OMS	Operations Management System (Microsoft)
NSG	Network Service Group (Microsoft)
NSS	Nanolog Streaming Service (for ZIA)
SOC	Security Operations Center
SIEM	Security Information and Event Management
SSH	Secure Socket Shell
SSL	Secure Socket Layer (RFC6101)
TCP	Transmission Control Protocol
TLS	Transport Layer Security (RFC5246)
UDP	User Datagram Protocol
VNet	Virtual Network (Microsoft)
ZIA	Zscaler Internet Access (Zscaler)
ZPA	Zscaler Private Access (Zscaler)

©2026 Zscaler, Inc. All rights reserved.

Trademark Notice

© 2026 Zscaler, Inc. All rights reserved. Zscaler™ and other trademarks listed at zscaler.com/legal/trademarks are either (i) registered trademarks or service marks or (ii) trademarks or service marks of Zscaler, Inc. in the United States and/or other countries. Any other trademarks are the properties of their respective owners.

About This Document

The following sections describe the organizations and requirements of this deployment guide.

Zscaler Overview

Zscaler (NASDAQ: ZS) enables the world's leading organizations to securely transform their networks and applications for a mobile and cloud-first world. ZIA and ZPA, create fast, secure connections between users and applications, regardless of device, location, or network. Zscaler delivers its services 100% in the cloud and offers the simplicity, enhanced security, and improved user experience that traditional appliances or hybrid solutions can't match. Used in more than 185 countries, Zscaler operates a massive, global cloud security platform that protects thousands of enterprises and government agencies from cyberattacks and data loss. To learn more, see Zscaler's website.

Microsoft Overview

Microsoft (Nasdaq: MSFT), Microsoft develops and licenses consumer and enterprise software. It is known for its Windows operating systems and Office productivity suite. The company is organized into three equally sized broad segments: productivity and business processes (legacy Microsoft Office, cloud-based Microsoft 365, Exchange, SharePoint, Skype, LinkedIn, Dynamics), intelligence cloud (infrastructure- and platform-as-a-service offerings Azure, Windows Server OS, SQL Server), and more personal computing (Windows Client, Xbox, Bing search, display advertising, and Surface laptops, tablets, and desktops). To learn more, refer to Microsoft's website.

Audience

This guide is for network administrators, endpoint and IT administrators, and security analysts responsible for deploying, monitoring, and managing enterprise security systems. For additional product and company resources, see the following sections:

- Zscaler Resources
- Microsoft Sentinel Resources
- Appendix A: Requesting Zscaler Support

Software Versions

This document was written using the latest Zscaler software and Microsoft Sentinel versions.

Request for Comments

- **For prospects and customers**: Zscaler values reader opinions and experiences. Contact partner-doc-support@zscaler.com to offer feedback or corrections for this guide.
- **For Zscaler employees**: Contact z-bd-sa@zscaler.com to reach the team that validated and authored the integrations in this document.

©2026 Zscaler, Inc. All rights reserved.

Zscaler and Microsoft Introduction

Overviews of the Zscaler and Microsoft applications are described in this section.

If you are using this guide to implement a solution at a government agency, some of the content might be different for your deployment. Efforts are made throughout the guide to note where government agencies might need different parameters or input. If you have questions, contact your Zscaler Account team.

ZIA Overview

ZIA is a secure internet and web gateway delivered as a service from the cloud. Think of ZIA as a secure internet on-ramp—just make Zscaler your next hop to the internet via one of the following methods:

- Setting up a tunnel (GRE or IPSec) to the closest Zscaler data center (for offices).
- Forwarding traffic via our lightweight Zscaler Client Connector or PAC file (for mobile employees).

No matter where users connect—a coffee shop in Milan, a hotel in Hong Kong, or a VDI instance in South Korea—they get identical protection. ZIA sits between your users and the internet and inspects every transaction inline across multiple security techniques (even within SSL).

You get full protection from web and internet threats. The Zscaler cloud platform supports Cloud Firewall, IPS, Sandboxing, DLP, and Isolation, allowing you to start with the services you need now and activate others as your needs grow.

ZPA Overview

ZPA is a cloud service that provides secure remote access to internal applications running on a cloud or a data center using a zero trust framework. With ZPA, applications are never exposed to the internet, making them completely invisible to unauthorized users. The service enables the applications to connect to users via inside-out connectivity rather than extending the network to them.

ZPA provides a simple, secure, and effective way to access internal applications. Access is based on policies created by the IT administrator within the ZPA Admin Portal and hosted within the Zscaler cloud. On each user device, software called Zscaler Client Connector is installed. Zscaler Client Connector ensures the user's device posture and extends a secure microtunnel out to the Zscaler cloud when a user attempts to access an internal application.

Zscaler Resources

The following table contains links to Zscaler resources based on general topic areas.

Name	Definition
ZIA Help Portal	Help articles for ZIA.
ZPA Help Portal	Help articles for ZPA.
Zscaler Tools	Troubleshooting, security and analytics, and browser extensions that help Zscaler determine your security needs.
Zscaler Training and Certification	Training designed to help you maximize Zscaler products.
Submit a Zscaler Support Ticket	Zscaler Support portal for submitting requests and issues.

©2026 Zscaler, Inc. All rights reserved.

The following table contains links to Zscaler resources for government agencies.

Name	Definition
ZIA Help Portal	Help articles for ZIA.
ZPA Help Portal	Help articles for ZPA.
Zscaler Tools	Troubleshooting, security and analytics, and browser extensions that help Zscaler determine your security needs.
Zscaler Training and Certification	Training designed to help you maximize Zscaler products.
Submit a Zscaler Support Ticket	Zscaler Support portal for submitting requests and issues.

Microsoft Sentinel Overview

Microsoft Sentinel (formerly Azure Sentinel) is a scalable, cloud-native, security information and event management (SIEM) and security orchestration, automation, and response (SOAR) solution. Microsoft Sentinel delivers intelligent security analytics and threat intelligence across the enterprise, providing a single solution for attack detection, threat visibility, proactive hunting, and threat response. Microsoft Sentinel is your birds-eye view across the enterprise alleviating the stress of increasingly sophisticated attacks, increasing volumes of alerts, and long resolution time frames.

- Collect data at cloud scale across all users, devices, applications, and infrastructure, both on-premises and in multiple clouds.
- Detect previously undetected threats and minimize false positives using Microsoft's analytics and unparalleled threat intelligence.
- Investigate threats with artificial intelligence, and hunt for suspicious activities at scale, tapping into years of cybersecurity work at Microsoft.
- Respond to incidents rapidly with built-in orchestration and automation of common tasks.

For more information on Azure, refer to the Microsoft Azure website or follow them on Twitter @microsoft.

Microsoft Sentinel Resources

The following table contains links to Azure support resources.

Name	Definition
Microsoft Sentinel Documentation	Article with use cases to get started using Microsoft Sentinel.
Azure support	Support portal for Azure problems and help.
Kusto Query Language Reference	Kusto Query Language (KQL) overview.

©2026 Zscaler, Inc. All rights reserved.

Document Prerequisites

To use this document, the following prerequisites are required:

- ZIA:
  - An active instance of the latest version of ZIA.
  - A working deployment of NSS. If not, see Understanding Nanolog Streaming Service (government agencies, see Understanding Nanolog Streaming Service) for more information.
  - Administrator login credentials to ZIA.
- Microsoft Sentinel:
  - Administrator login credentials to Microsoft Azure.
  - Active subscription with Microsoft Sentinel.

©2026 Zscaler, Inc. All rights reserved.

Syslog, CEF, and LEEF

This section explains Syslog. If you are already familiar with Syslog, skip to the section Zscaler Logging Architecture.

Syslog

Syslog has been used for many decades. Over time, new standards were created to define new message formats and support new use cases. Briefly, a Syslog message has the following structure (in order): a header, structured data (SD), and a message. In this section, the Syslog header and Syslog message (the body of the message) are explained.

RFC 3164 is considered the original BSD structure from 2001. The following is an example log message:

```
<34>Oct 11 22:14:15 mymachine su: 'su root' failed for lonvick on /dev/pts/8
```

Figure 1. Syslog message in RFC 3164 format

- Syslog Header:
  - `<34>` is a priority number. It represents the sum of the facility number multiplied by eight and severity. In this case, facility=4 and severity=2.
  - `Oct 11 22:14:15` is the timestamp. It doesn't include the year, time zone, and subsecond information.
  - `mymachine` is a host name where the message was written.
  - `su` is a tag. Typically, this is the process name – sometimes having a PID (e.g., `su[1234]`).
- Syslog Message:
  - The remainder of the message (MSG) is everything after the `su` tag.

The new (2009) Syslog format (RFC 5424) is three parts: *Syslog Header*, *Structured Data*, and the actual log *message*.

- Syslog Header. Consists of priority, version, timestamp, hostname, etc.
- Structured Data. This is in key=value format. It provides a mechanism to express information in a defined, parsable and interpretable data format (e.g., SD-ID, SD-PARAM).
- Actual log message. This follows the previous two fields (the Message field is free-form).

```
<34>1 2003-10-11T22:14:15.003Z mymachine.example.com su - - - 'su root' failed for lon
vick on /dev/pts/8
```

Figure 2. Syslog message in RFC 5424 format

©2026 Zscaler, Inc. All rights reserved.

- The dashes are places for the PID, message ID, and other structured data.

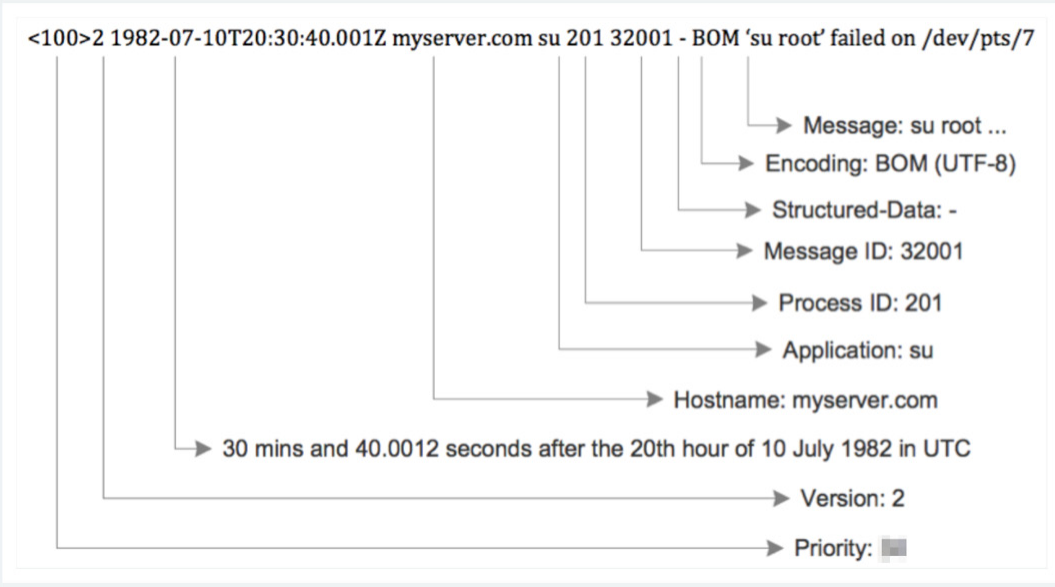


Figure 3. Another Syslog message in RFC 5424 format

CEF

Common Event Format (CEF) is an open log management standard that improves the interoperability of security-related information from different security and network devices and applications.

Base CEF format:

```
CEF:Version|Device Vendor|Device Product|Device Version|Signature
ID|Name|Severity|Extension
```

LEEF

Log Event Extended Format (LEEF) is a customized event format created by IBM QRadar. It is designed to describe (network) security events and uses encoding and transport like those used by CEF. However, the two formats differ in the number and types of fields.

Base LEEF Format:

```
LEEF:2.0|Vendor|Product|Version|EventID|(Delimiter Character, optional if the
Delimiter Character is tab)|Extension
```

Although Syslog usually uses UDP and destination port 514, NSS only supports TCP. By using TCP, NSS can detect if the SIEM becomes unavailable by the loss of the TCP connection. In the event of a failure, NSS queues log messages until the SIEM returns (subject to storage).

©2026 Zscaler, Inc. All rights reserved.

Zscaler Logging Architecture

Zscaler has two core products: ZIA and ZPA. ZIA's and ZPA's ability to send log messages outside of Zscaler's cloud requires a product known as Nanolog Streaming Service (NSS).

When customers use ZIA or ZPA, every customer-initiated transaction that traverses Zscaler generates a corresponding log message. These logs messages are retained by Zscaler for 6 months (or longer through a paid-for service). Customers can view and search these logs using the dashboard of the ZIA Admin Portal or ZPA Admin Portal.

NSS

Log messages are stored within Nanolog. When an organization deploys NSS for various log feeds, each NSS opens a secure tunnel to the Nanolog in the Zscaler cloud. The Nanolog then streams copies of the logs to each NSS in a highly compressed format to reduce bandwidth footprint. The original logs are retained on the Nanolog.

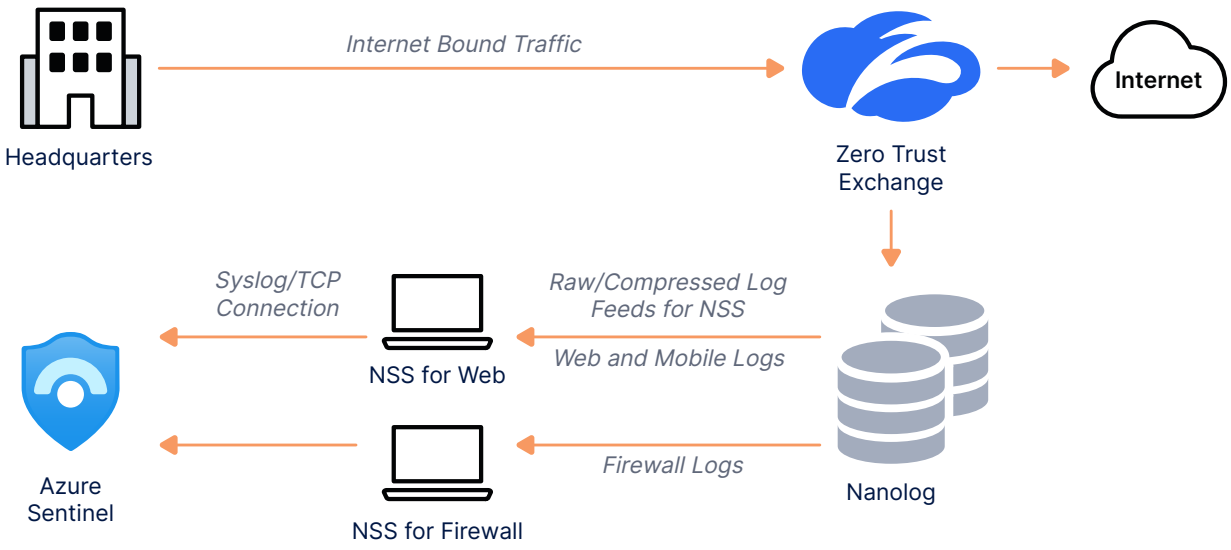


Figure 4. Nanolog Streaming Service (NSS) overview

When an NSS receives the logs from the Nanolog, it unscrambles them, applies the configured filters to exclude unwanted logs, converts the filtered logs to the configured output format so they are parsed by your SIEM, and then streams the logs to your SIEM over a raw TCP connection.

Zscaler NSS is required for customers who want to send these logs to a SIEM (on-premises or in-the-cloud). Think of NSS as an intermediate log gateway. NSS uses a virtual machine (VM) to stream traffic logs in real time from ZIA.

©2026 Zscaler, Inc. All rights reserved.

About Cloud-To-Cloud Log Streaming

Your organization can optionally subscribe to cloud-to-cloud log streaming (Cloud NSS), which allows direct cloud-to-cloud log streaming for all types of ZIA logs into a compatible cloud-based SIEM solution. Rather than deploying, managing, and monitoring on-premises NSS VMs, you can simply configure an HTTPS API feed to push logs from the Zscaler cloud service into an HTTPS API-based log collector on the SIEM.



Figure 5. Cloud-to-cloud log streaming

ZIA Log Feeds

An NSS feed specifies the data from the logs that the NSS sends to the SIEM. You can filter the data to send only the data you need to the SIEM. You can add one or more fields for the logs and one field for alerts. You can add up to 8 NSS feeds for each NSS. Each feed can have a different list of fields, a different format, and different filters. The following are the supported feeds.

Zscaler Platform & Product	Feed Types
ZIA NSS	Web Logs
ZIA NSS	Firewall Logs
ZIA NSS	DNS Logs
ZIA NSS	Tunnel Logs
ZIA NSS	SaaS Security Logs
ZIA NSS	SaaS Security Activity Logs
ZIA NSS	Admin Audit Logs
ZIA NSS	Endpoint DLP Logs

©2026 Zscaler, Inc. All rights reserved.

Configuring Cloud NSS-Based Log Ingestion for ZIA

Perform the following steps to configure Cloud NSS and Microsoft Sentinel.

In the Azure Portal, Register an Entra ID Application, and Create a Client Secret

1. Log in to the Azure portal.
2. Click the **Microsoft Entra ID** tile.

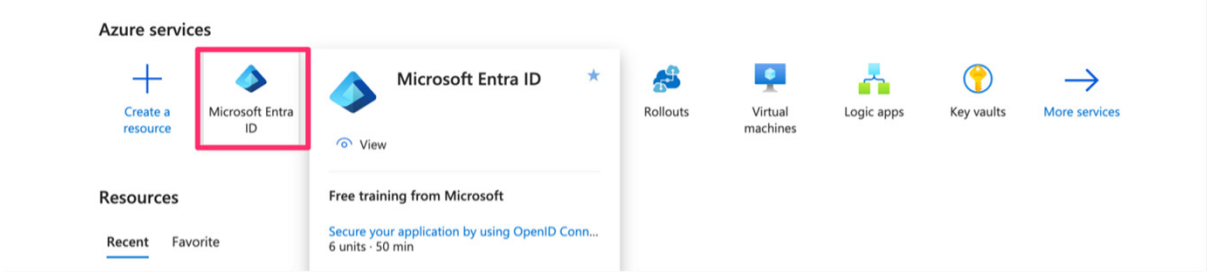


Figure 6. Microsoft Entra ID tile

3. Click **Manage** > **App registrations**.

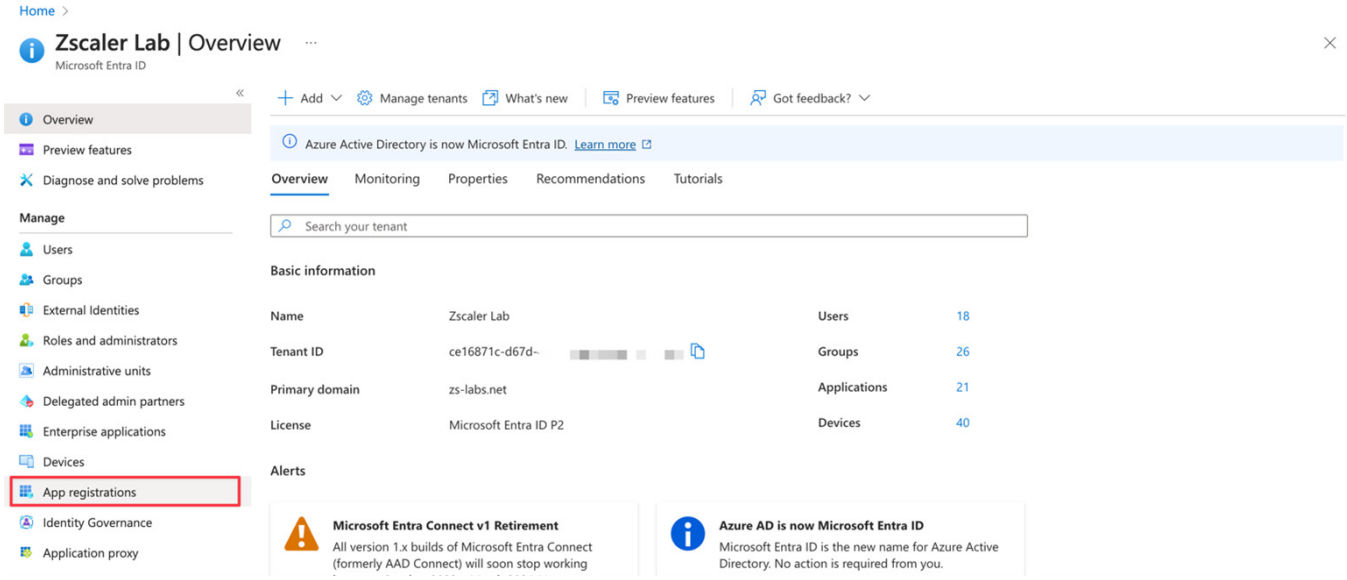


Figure 7. Microsoft Sentinel create

4. Click **New registration**.

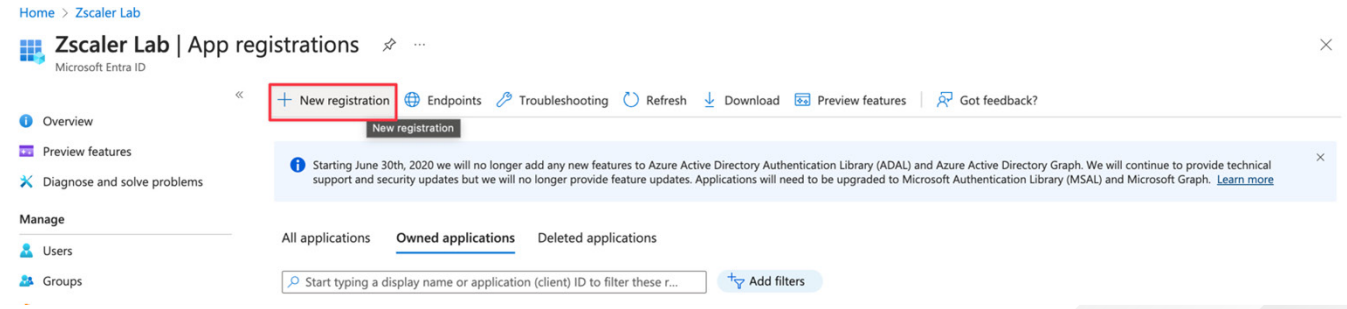


Figure 8. Create a new registration

©2026 Zscaler, Inc. All rights reserved.

5. In the **Name** field, enter an application name and click **Register**.

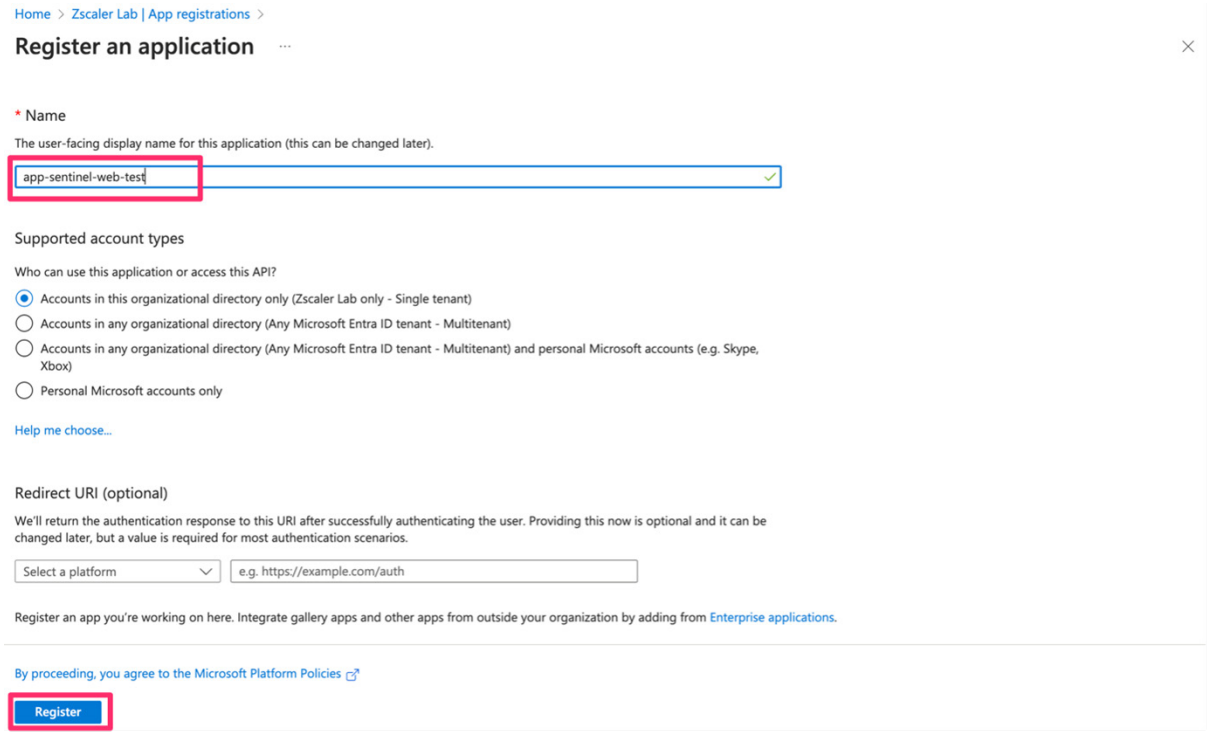


Figure 9. Registering the application

6. In the newly created application, click **Certificates & secrets**.

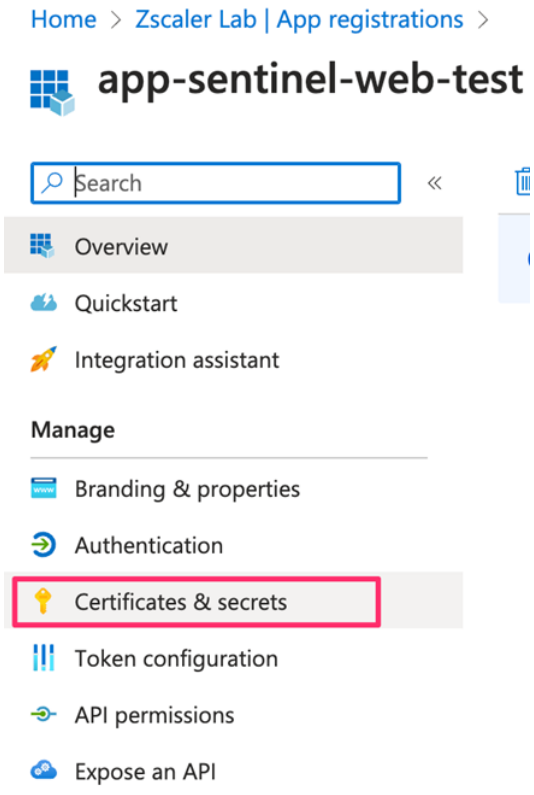


Figure 10. Certificates & secrets

©2026 Zscaler, Inc. All rights reserved.

7. Click **New client secret**.

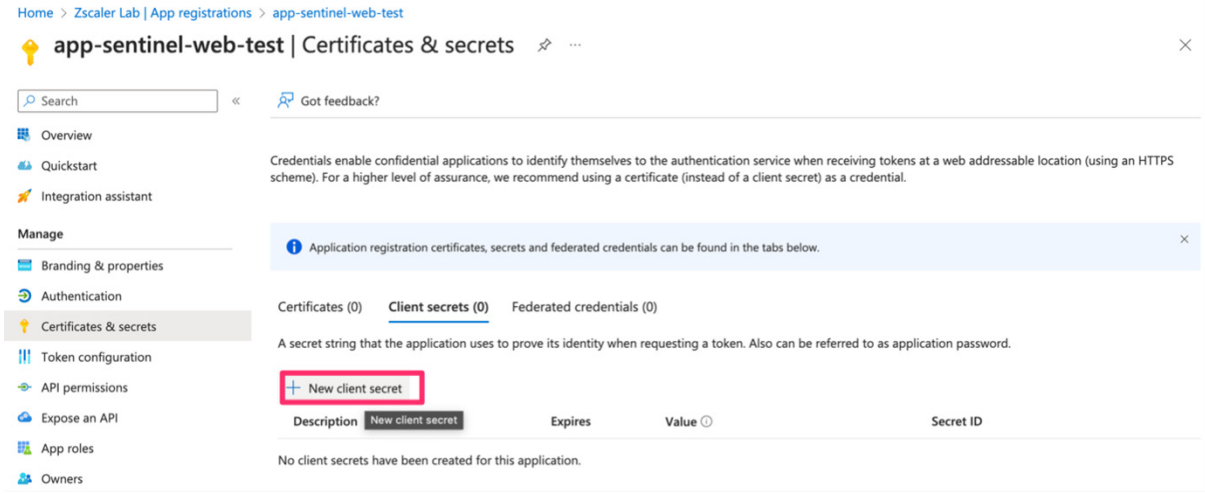


Figure 11. New client secret

8. Click **Add a secret** and enter a **Description.**
9. Click **Add**.

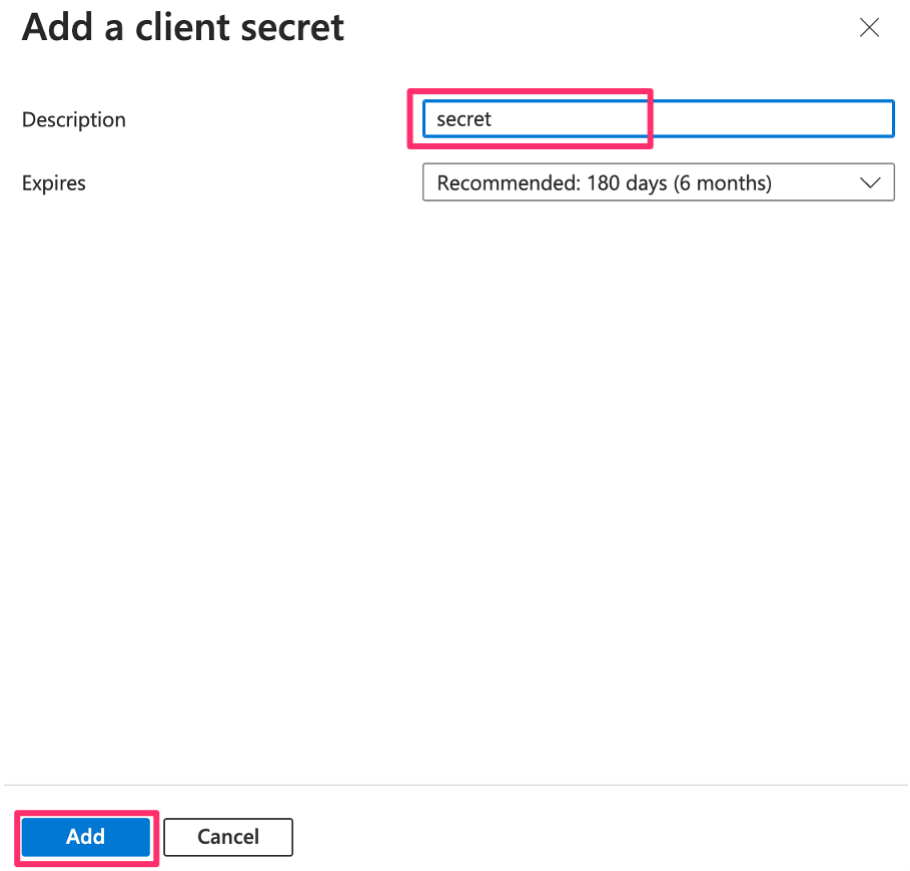


Figure 12. Add a client secret

©2026 Zscaler, Inc. All rights reserved.

10. Copy the secret **Value**, which you need to add a Cloud NSS Feed in ZIA.

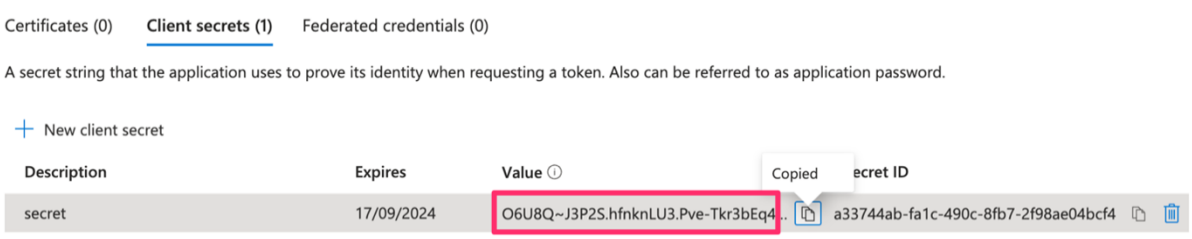


Figure 13. Secret value

In the Azure Portal, Create a Log Analytics Workspace, and Add Microsoft Sentinel to the Workspace

To create a log analytics workspace:

1. Log in to the Azure portal.
2. Click the **Microsoft Sentinel** tile.

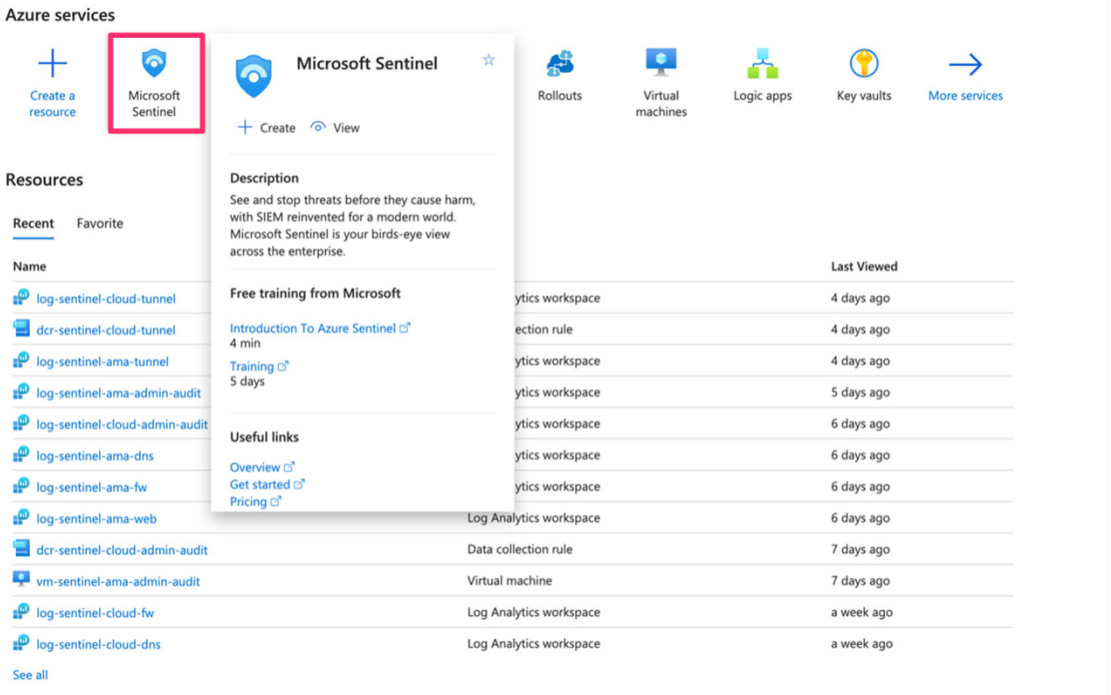


Figure 14. Click the Microsoft Sentinel tile

3. Click **Create**.

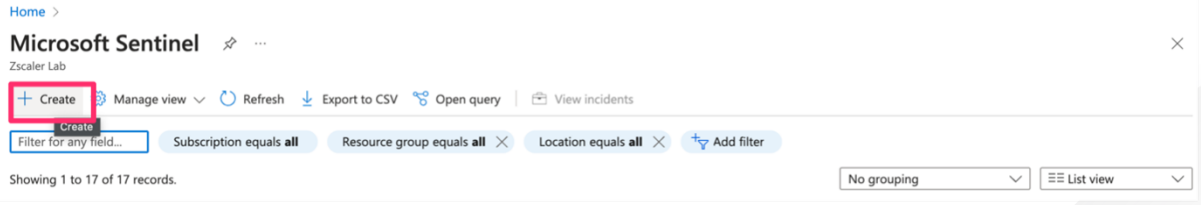


Figure 15. Click Create

©2026 Zscaler, Inc. All rights reserved.

4. Click **Create a new workspace**.

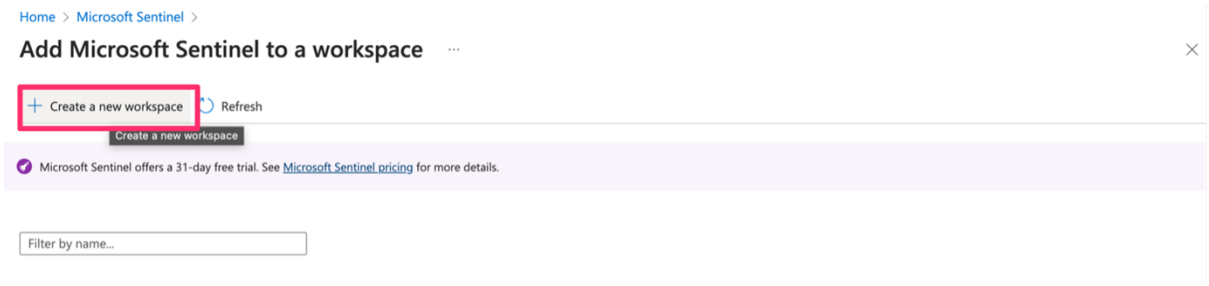


Figure 16. Click Create a new workspace

5. Create a new resource group by clicking **Create new**.
6. Enter a **Name**.
7. Click **OK**.

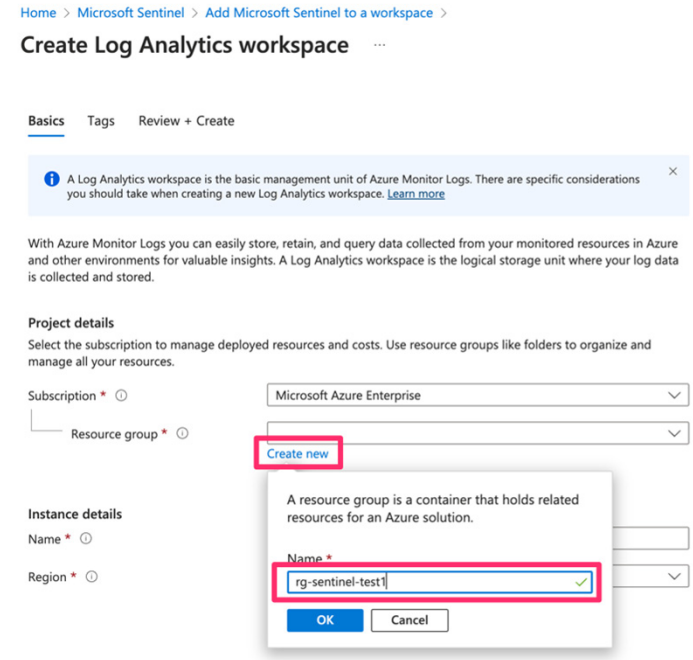


Figure 17. Create a new resource group

©2026 Zscaler, Inc. All rights reserved.

8. Enter a workspace name and select a **Region**.
9. Click **Review + Create**.

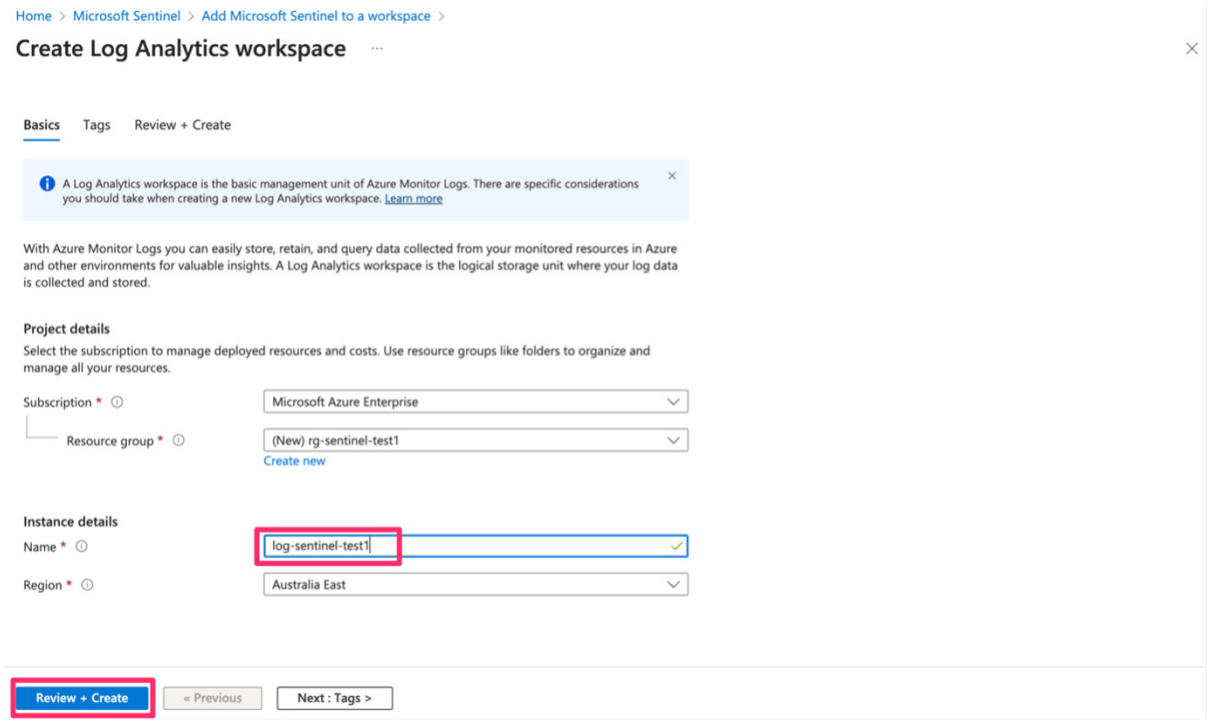


Figure 18. Create a workspace

10. Wait a moment for the workspace to be created, then click **Refresh**, and then **Add**.

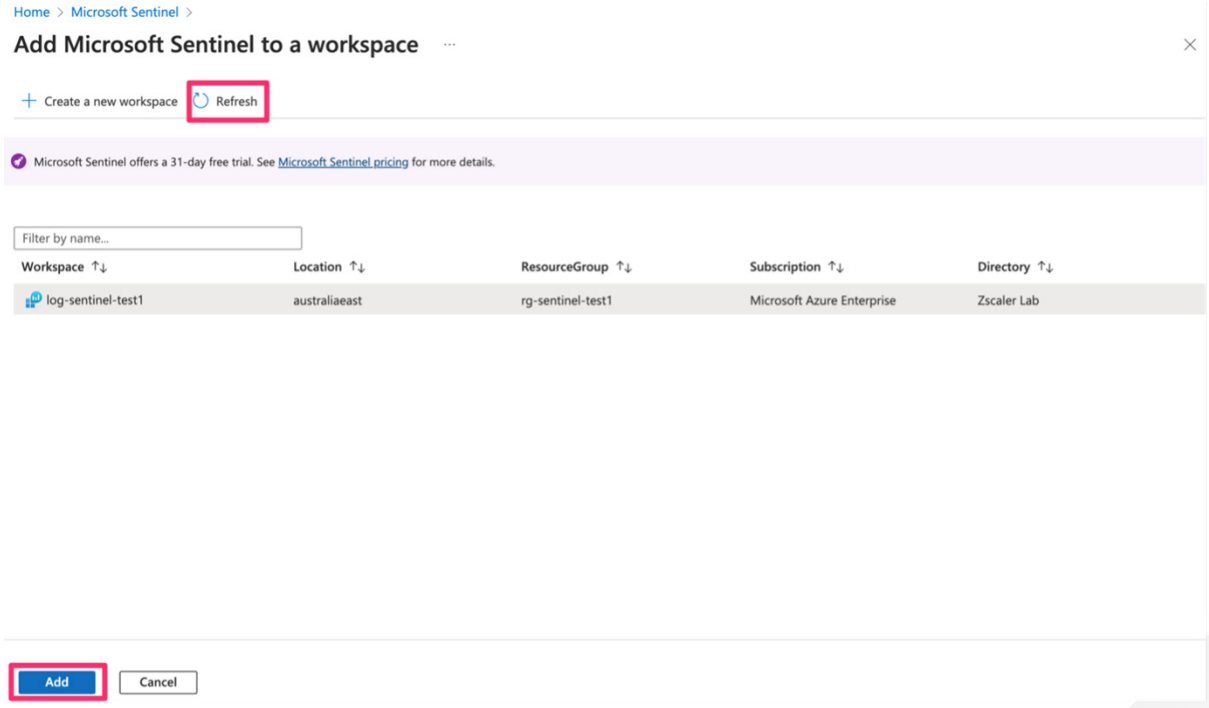


Figure 19. Add Microsoft Sentinel to a workspace

You are redirected to the Microsoft Sentinel | News & guides page and a success message appears.

©2026 Zscaler, Inc. All rights reserved.

Deploy a Data Collection Rule and Data Collection Endpoint Using a Bicep Template

To deploy a bicep template through a device connected to ZIA, configure an SSL Bypass exemption for aka.ms to successfully run the template and deploy the stack.

To deploy a bicep template:

1. Log in to the Azure portal.
2. Click the **Log Analytics workspaces** tile.

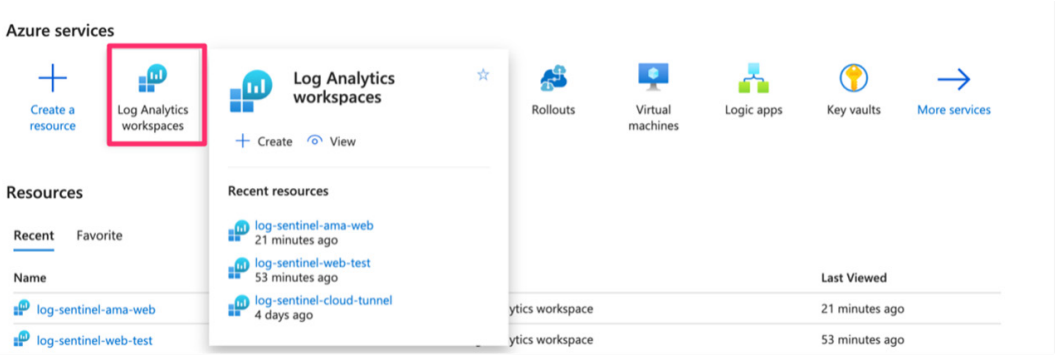


Figure 20. Log Analytics workspaces tile

3. Select the workspace you've just created.

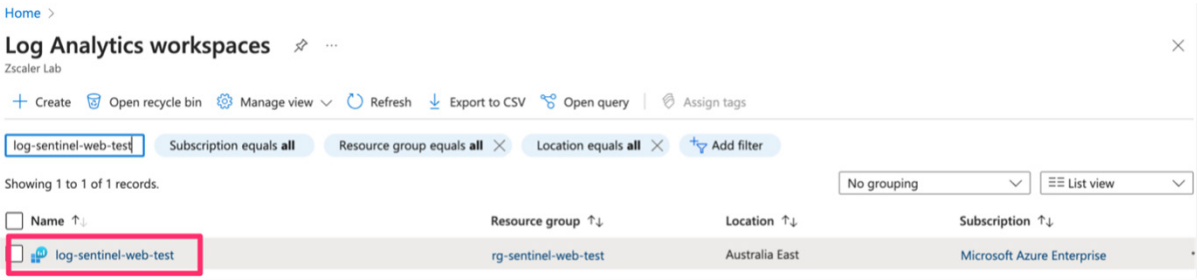


Figure 21. Select workspace

©2026 Zscaler, Inc. All rights reserved.

4. Click **Overview**, and note the values in the following fields:
   a. **Resource group**
   b. **Workspace Name**
   c. **Workspace ID**
   d. **Location**
   e. **Subscription ID**

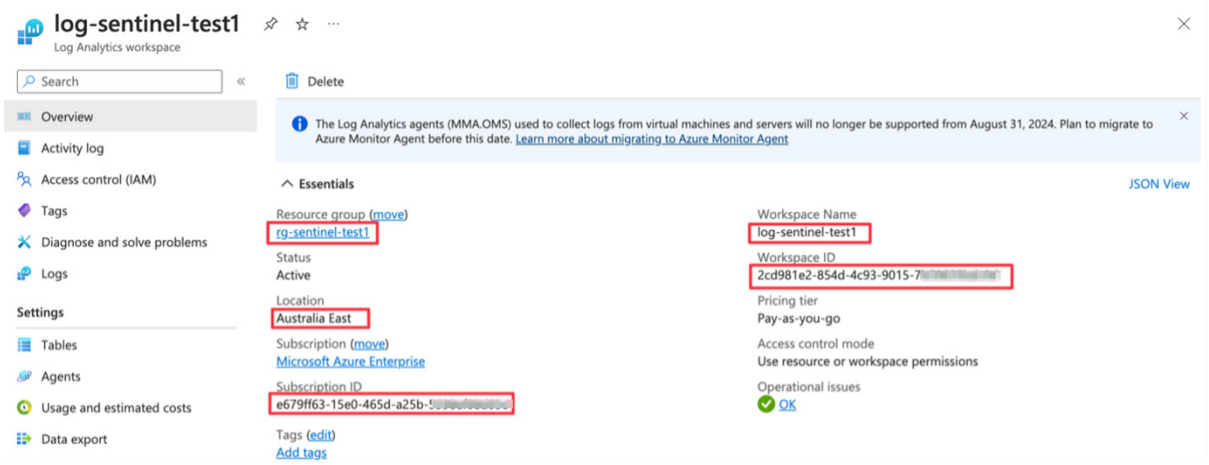


Figure 22. Overview

5. Go to the [Zscaler GitHub repo] location that holds the Bicep templates.

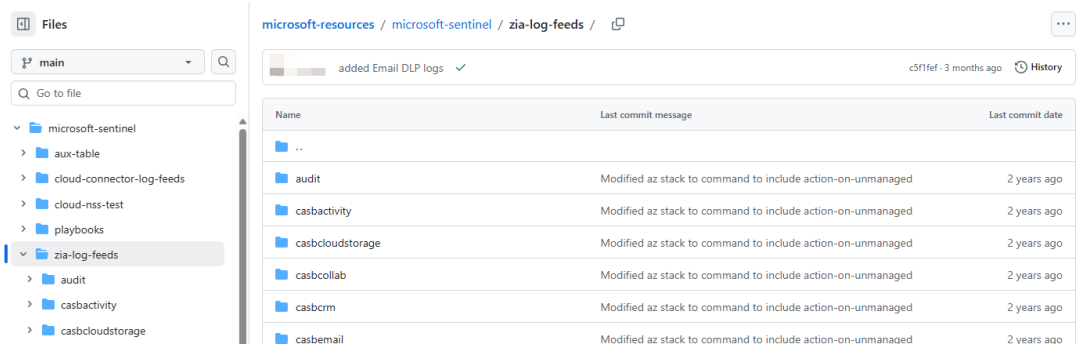


Figure 23. GitHub repo location

©2026 Zscaler, Inc. All rights reserved.

6. Download the .bicep and .bicepparam file for the Cloud NSS Feed you want to deploy.
7. Fill out the .bicepparam file with the information collected earlier.

```
using 'cloud-nss-web.bicep'

// Define the 'param' values below

// Found under Log Analytics workspace -> your workspace -> Overview -> Resource group
param resourceGroup = 'rg-sentinel-test1'

// The name of your Log Analytics workspace
param workspaceName = 'log-sentinel-test1'

// Found under Log Analytics workspace -> your workspace -> Overview -> JSON View -> "location": (i.e. useast)
param location = 'australiaeast'

// Found under Log Analytics workspace -> your workspace -> Overview -> Subscription ID
param subscriptionId = 'e679ff63-15e0-465d-a25b-[redacted]'

// Found under Log Analytics workspace -> your workspace -> Overview -> Workspace ID
param workspaceId = '2cd981e2-854d-4c93-9015-[redacted]'

// The name you want to provide the Data Collection Endpoint that the template will create
param dceName = 'dce-sentinel-test1'

// The name you want to provide the Data Collection Rule that the template will create
param dcrName = 'dcr-sentinel-test1'
```

Figure 24. .biceparam file

8. Run the bicep template with the following command (this example uses `cloud-nss-web.bicepparam` as an example):

   `az stack group create --name cloud-nss-web --resource-group <resource group containing your log analytics workspace> --parameters cloud-nss-web.bicepparam --deny-settings-mode 'none' --action-on-unmanage deleteResources`

9. After the stack has deployed, run the following command to get the API URL, which is needed for the Cloud NSS configuration within the ZIA Admin Portal:

   `az stack group show -g <resouce group containing your log analytics workspace> -n cloud-nss-web --query outputs.api_url`

This outputs a JSON object. Record the Value, as this is needed for the Cloud NSS Feed configuration. The following is an example of the output:

```
{

 "type": "String",

 "value": "https://dce-sentinel-test1-1tej.australiaeast-1.ingest.monitor.azure.com/dataCollectionRules/dcr-b6907e2f6c144367acbaf4f58f919f59/streams/Custom-nss_web_CL?api-version=2021-11-01-preview"

}
```

©2026 Zscaler, Inc. All rights reserved.

In the Azure Portal, Assign Permissions to the Data Collection Rule

To assign permissions:

1. Log in to the Azure portal.
2. Click the **Monitor** tile, then **Data Collection Rules**, and select the **DCR** you created (e.g., `dcr-sentinel-test1`).

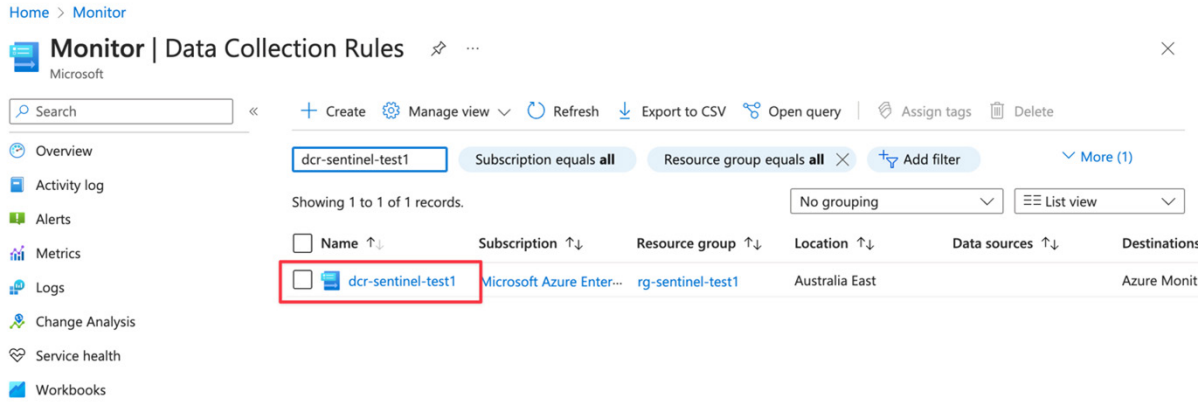


Figure 25. Data Collection Rules

3. In the left-side navigation, select **Access control (IAM)**.

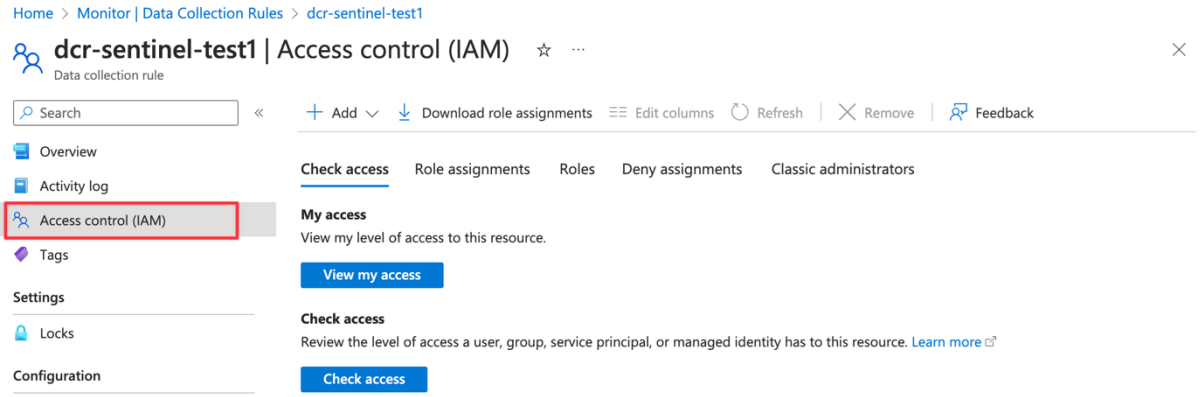


Figure 26. Access control (IAM)

©2026 Zscaler, Inc. All rights reserved.

4. Select **Add** > **Add role assignment**.

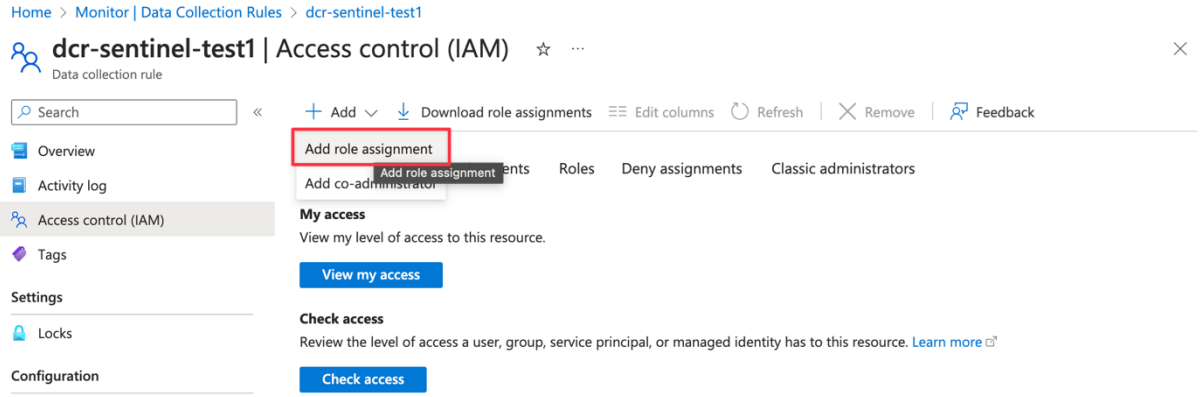


Figure 27. Add role assignment

5. Select **Monitoring Metrics Publisher**.

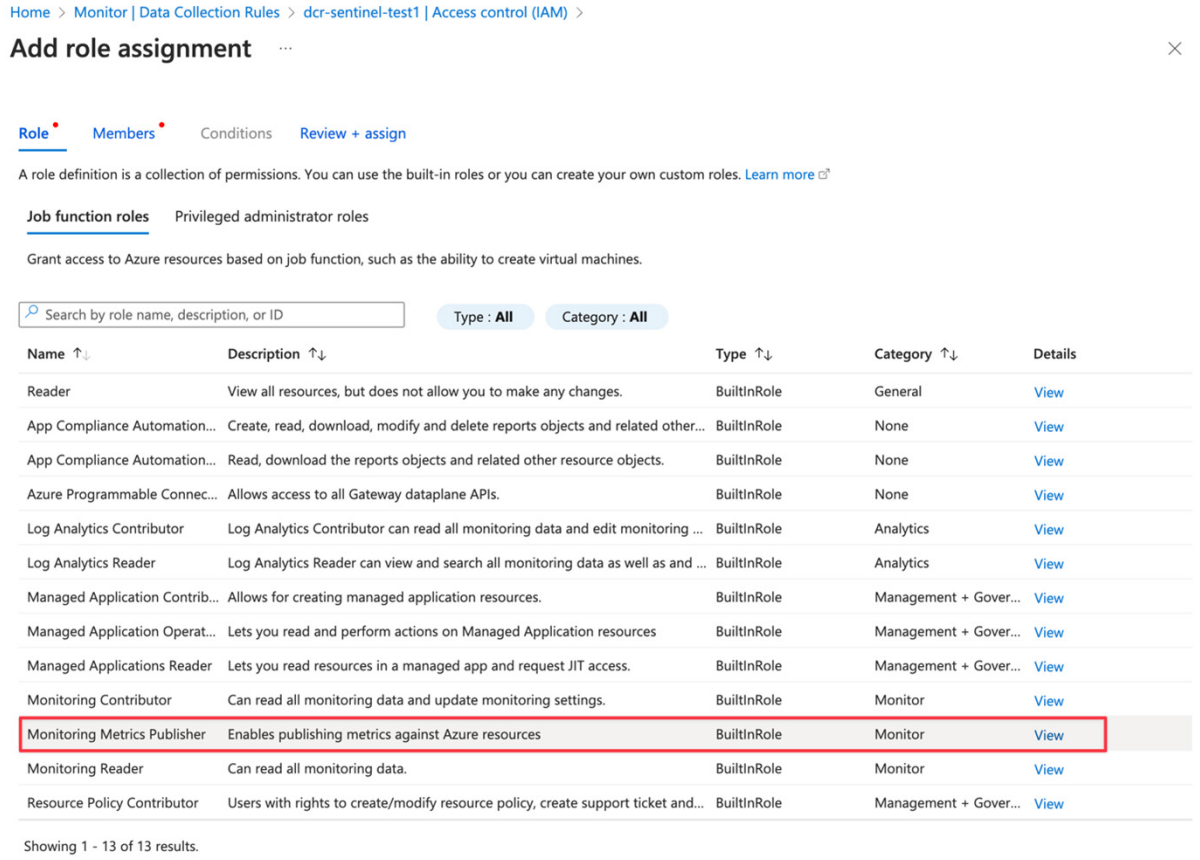


Figure 28. Monitoring Metrics Publisher

©2026 Zscaler, Inc. All rights reserved.

6. Select **Next** > **Select members**.

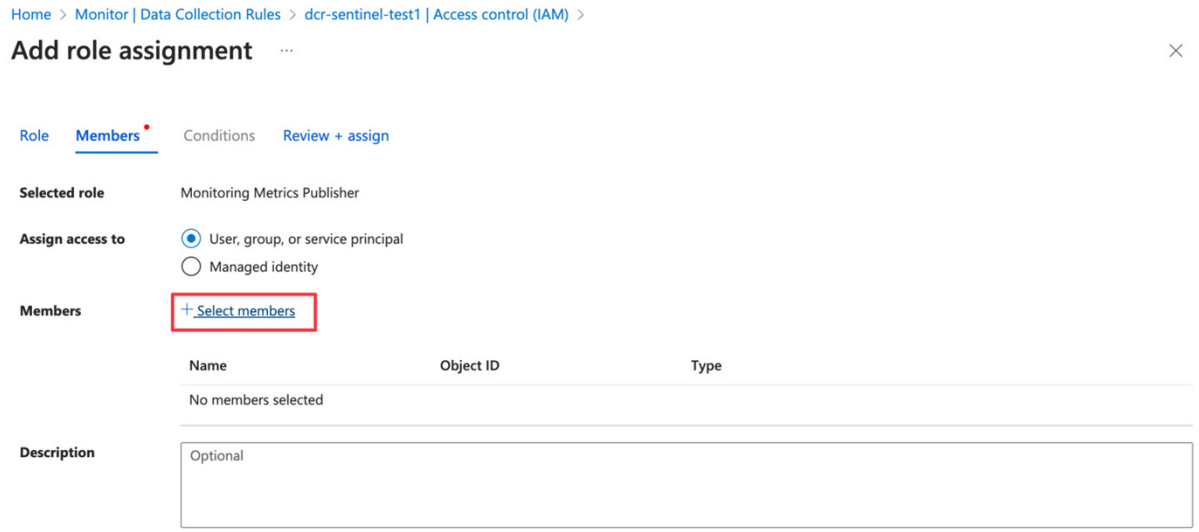


Figure 29. Add role assignment

7. Enter the name of the app registration in to the **Select** field, and click it to add it.
8. Click **Select**.

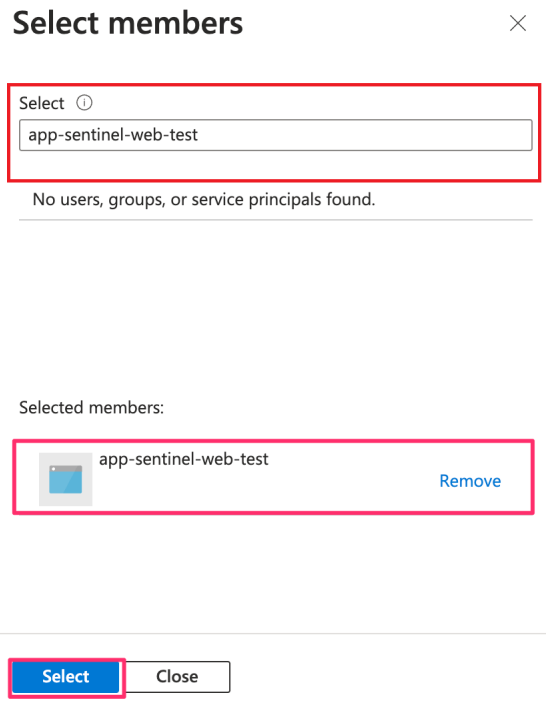


Figure 30. Select members

©2026 Zscaler, Inc. All rights reserved.

9. Click **Review + assign**.

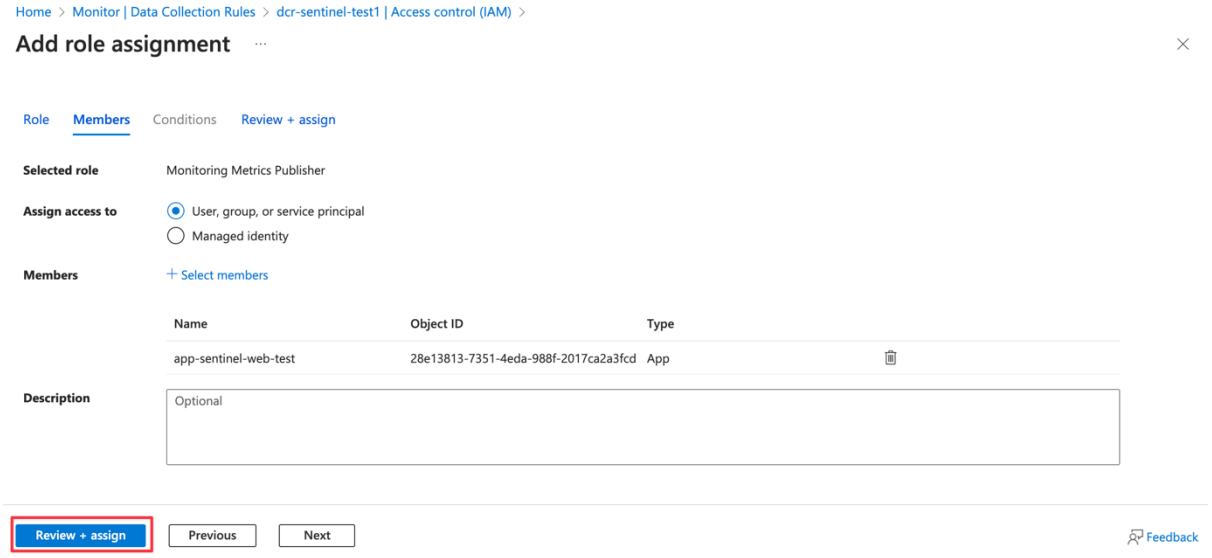


Figure 31. Review + assign

You have completed the setup needed on the Microsoft Azure side for Cloud NSS.

©2026 Zscaler, Inc. All rights reserved.

Configuring ZIA for Cloud NSS-Based Log Export

The following steps demonstrate how to configure ZIA for cloud NSS-based log export.

In the ZIA Admin Portal, Add a Cloud NSS Feed

See Adding Cloud NSS Feeds (government agencies, see Adding Cloud NSS Feeds), and select the type of feed (e.g., Web Logs) that you want to add. The following fields require specific inputs:

- **SIEM Type**: Select **Azure Sentinel**.
- **OAuth 2.0 Authentication**: Enabled by default. The toggle is not editable.
- **Client ID**: Enter the Application (client) ID generated in the Azure portal.
- **Client Secret**: Enter the application client secret value generated in the Azure portal.
- **Scope**: Enter the following URL: `https://monitor.azure.com//.default`
- **Grant Type**: Enter the following string: `client_credentials`
- **Authorization URL**: Enter the authorization URL with the Directory (tenant) ID generated in the Azure Portal (e.g., `https://login.microsoftonline.com/<$tenantid>/oauth2/v2.0/token`). Replace `$tenantid` in the URL with the Directory (tenant) ID from your deployment.
- **API URL**: Enter the API URL noted earlier using the `az stack group show` command.

  ```
  $dceEndpoint/dataCollectionRules/<$dcrImmutableId>/streams/<Custom-table_name_CL>?api-version=2021-11-01-preview
  ```

  Replace the values shown in red with those from your deployment:

  ```
  https://<test-1-dce-XXw5.eastus-1.ingest.monitor.azure.com/dataCollectionRules/<dcr-XXabc28ce0514dXXX55766bdc7328XX>/streams/<Custom-table_1_web_CL>?api-version=2021-11-01-preview).
  ```

  You can also get this value using the following command:

  ```
  az stack group show -g <resouce group containing your log analytics workspace> -n <stack name> --query outputs.api_url
  ```

- **HTTP Headers**:
  - **Key 1**: Enter `Content-Type`
  - **Value 1**: Enter `application/json`
- **Feed Output Type**: Select **JSON**.
- **JSON Array Notation**: Enable this setting.
- **Feed Escape Character**: Enter `"\,`

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Web Logs

For Cloud NSS Feeds for Web Logs (government agencies, see Cloud NSS Feeds for Web Logs), copy and paste the prepopulated **Feed Output Format** from the file cloud-nss-web.fof in the Zscaler GitHub repo or with the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype
":"zscalernss","vendor":"Zscaler","product":"NSSWeblog","version":"5.7","act":"%s{a
ction}","Severity":"3","name":"%s{reason}","deviceEventClassId":"%s{action}","app":
"%s{proto}","cat":"%s{urlcat}","dhost":"%s{ehost}","dst":"%s{sip}","src":"%s{cip}",
"in_bytes":"%d{respsize}","in":"%d{respsize}","outcome":"%s{respcode}","out":"%d{re
qsize}","request":"%s{eurl}","rt":"%s{mon}","sourceTranslatedAddress":"%s{cintip}",
"requestClientApplication":"%s{ua}","requestMethod":"%s{reqmethod}","suser":"%s{log
in}","spriv":"%s{location}","externalId":"%d{recordid}","fileType":"%s{filetype}","re
ason":"%s{reason}","destinationServiceName":"%s{appname}","cn1":"%d{riskscore}","cn
1Label":"riskscore","cs1":"%s{dept}","cs1Label":"dept","cs2":"%s{urlsupercat}","cs2
Label":"urlsupercat","cs3":"%s{appclass}","cs3Label":"appclass","cs4":"%s{malwarecat
}","cs4Label":"malwarecat","cs5":"%s{threatname}","cs5Label":"threatname","cs6":"%s{
dlpeng}","cs6Label":"dlpeng","ZscalerNSSWeblogURLClass":"%s{urlclass}","ZscalerNSSWe
blogDLPDictionaries":"%s{dlpdict}","requestContext":"%s{ereferer}","contenttype":"%s
{contenttype}","unscannabletype":"%s{unscannabletype}","deviceowner":"%s{deviceowner
}","devicehostname":"%s{devicehostname}","keyprotectiontype":"%s{keyprotectiontype}"
,"cloudname":"%s{cloudname}","company":"%s{company}","throttlereqsize":"%d{throttler
eqsize}","throttlerespsize":"%d{throttlerespsize}","bwthrottle":"%s{bwthrottle}","bw
classname":"%s{bwclassname}","bwrulename":"%s{bwrulename}","module":"%s{module}","a
pp_risk_score":"%s{app_risk_score}","datacenter":"%s{datacenter}","datacentercity":"%
s{datacentercity}","datacentercountry":"%s{datacentercountry}","dlpdicthitcount":"%s{
dlpdicthitcount}","dlpidentifier":"%d{dlpidentifier}","dlpmd5":"%s{dlpmd5}","dlprulena
me":"%s{dlprulename}","fileclass":"%s{fileclass}","fname":"%s{filename}","filesubtype":"
%s{filesubtype}","upload_fileclass":"%s{upload_fileclass}","upload_filetype":"%s{upload_
filetype}","upload_filename":"%s{upload_filename}","upload_filesubtype":"%s{upload_
filesubtype}","upload_doctypename":"%s{upload_doctypename}","rdr_rulename":"%s{rdr_
rulename}","fwd_type":"%s{fwd_type}","fwd_gw_name":"%s{fwd_gw_name}","fwd_gw_
ip":"%s{fwd_gw_ip}","zpa_app_seg_name":"%s{zpa_app_seg_name}","reqdatasize":"%d{reqda
tasize}","reqhdrsize":"%d{reqhdrsize}","respdatasize":"%d{respdatasize}","resphdrsize
":"%d{resphdrsize}","totalsize":"%d{totalsize}","df_hosthead":"%s{df_hosthead}","df_
hostname":"%s{df_hostname}","erefererhost":"%s{erefererhost}","refererpath":"%s{
erefererpath}","eurlpath":"%s{eurlpath}","reqversion":"%s{reqversion}","respvers
ion":"%s{respversion}","ua_token":"%s{ua_token}","uaclass":"%s{uaclass}","mobappna
me":"%s{mobappname}","mobappcat":"%s{mobappcat}","mobdevtype":"%s{mobdevtype}","c
lt_sport":"%d{clt_sport}","cpubip":"%s{cpubip}","alpnprotocol":"%s{alpnprotocol}","tr
afficredirectmethod":"%s{trafficredirectmethod}","euserlocationname":"%s{euserlocationna
me}","erulelabel":"%s{erulelabel}","ruletype":"%s{ruletype}","eurlfilterrulelabel":"%
s{eurlfilterrulelabel}","ourlfilterrulelabel":"%s{ourlfilterrulelabel}","eapprulelabel
":"%s{eapprulelabel}","fileHash":"%s{bamd5}","sha256":"%s{sha256}","ssldecrypted":"%s
{ssldecrypted}","externalspr":"%s{externalspr}","clientsslcipher":"%s{clientsslciphe
r}","clienttlsversion":"%s{clienttlsversion}","clientsslsessreuse":"%s{clientsslsess
reuse}","cltsslfailreason":"%s{cltsslfailreason}","cltsslfailcount":"%d{cltsslfailco
unt}","srvsslcipher":"%s{srvsslcipher}","srvtlsversion":"%s{srvtlsversion}","srvocsp
result":"%s{srvocspresult}","srvcertchainvalpass":"%s{srvcertchainvalpass}","srvwild
cardcert":"%s{srvwildcardcert}","serversslsessreuse":"%s{serversslsessreuse}","srvce
rtvalidationtype":"%s{srvcertvalidationtype}","srvcertvalidityperiod":"%s{srvcertval
```

©2026 Zscaler, Inc. All rights reserved.

```
idityperiod}","is_ssluntrustedca":"%s{is_ssluntrustedca}","is_sslselfsigned":"%s{is_
sslselfsigned}","is_sslexpiredca":"%s{is_sslexpiredca}","threatseverity":"%s{threats
everity}","malwareclass":"%s{malwareclass}","urlcatmethod":"%s{urlcatmethod}","bypa
ssed_traffic":"%d{bypassed_traffic}","bypassed_etime":"%s{bypassed_etime}","deviceappve
rsion":"%s{deviceappversion}","devicemodel":"%s{devicemodel}","devicename":"%s{devi
cename}","deviceostype":"%s{deviceostype}","devicetype":"%s{devicetype}","external_
devid":"%s{external_devid}","flow_type":"%s{flow_type}","ztunnelversion":"%s{ztunnelver
sion}","productversion":"%s{productversion}","nsssvcip":"%s{nsssvcip}"\}"
```

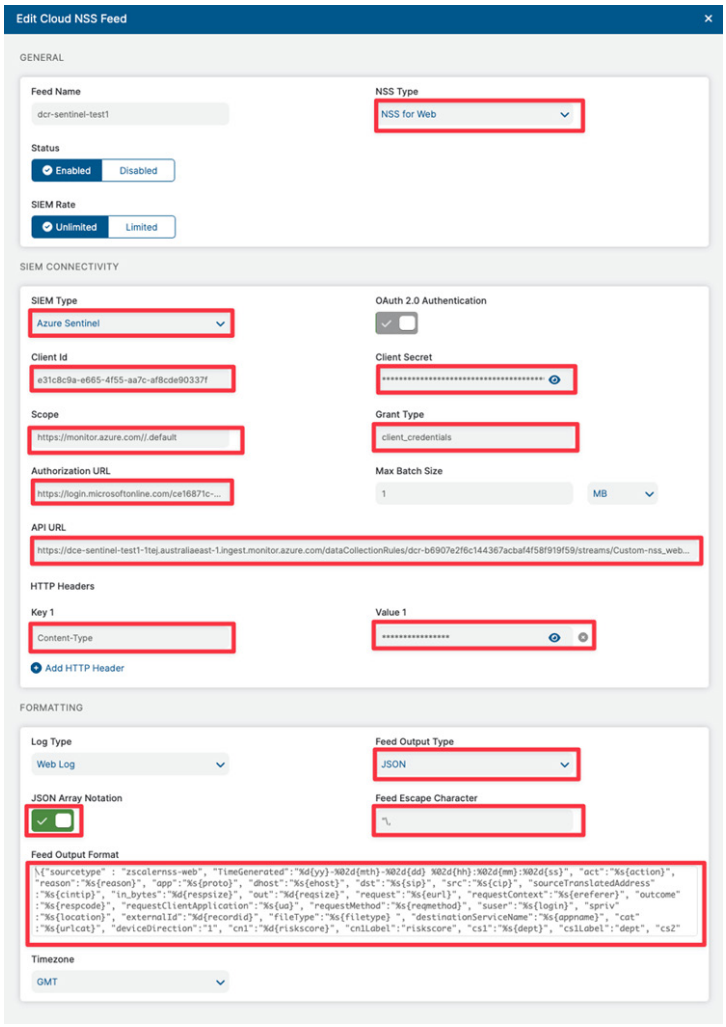



Figure 32. Add Cloud NSS logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Firewall Logs

For Cloud NSS Feeds for Firewall Logs (government agencies, see Cloud NSS Feeds for Firewall Logs), copy and paste the prepopulated **Feed Output Format** from the file cloud-nss-fw.fof in the Zscaler GitHub repo, or use the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype"
:"zscaler-nss-fw","vendor":"Zscaler","product":"NSSFWlog","version":"5.7","Severity":
"3","name":"%s{rulelabel}","deviceEventClassId":"%s{action}","act":"%s{action}","suse
r":"%s{login}","src":"%s{csip}","spt":"%d{csport}","dst":"%s{cdip}","dpt":"%d{cdport}
","deviceTranslatedAddress":"%s{ssip}","deviceTranslatedPort":"%d{ssport}","destinati
onTranslatedAddress":"%s{sdip}","destinationTranslatedPort":"%d{sdport}","sourceTrans
latedAddress":"%s{tsip}","sourceTranslatedPort":"%d{tsport}","proto":"%s{ipproto}","t
unnelType":"%s{ttype}","dnat":"%s{dnat}","stateful":"%s{stateful}","spriv":"%s{locati
on}","reason":"%s{rulelabel}","in_bytes":"%ld{inbytes}","in":"%ld{inbytes}","out":"%l
d{outbytes}","deviceDirection":"1","cs1":"%s{dept}","cs1Label":"dept","cs2":"%s{nwsvc
}","cs2Label":"nwService","cs3":"%s{nwapp}","cs3Label":"nwApp","cs4":"%s{aggregate}",
"cs4Label":"aggregated","cs5":"%s{threatcat}","cs5Label":"threatcat","cs6":"%s{threat
name}","cs6label":"threatname","cn1":"%d{durationms}","cn1Label":"durationms","cn2":"
%d{numsessions}","cn2Label":"numsessions","flexString1Label":"ipCat","flexString1":"%s{
ipcat}","destCountry":"%s{destcountry}","avgduration":"%d{avgduration}"\}
```

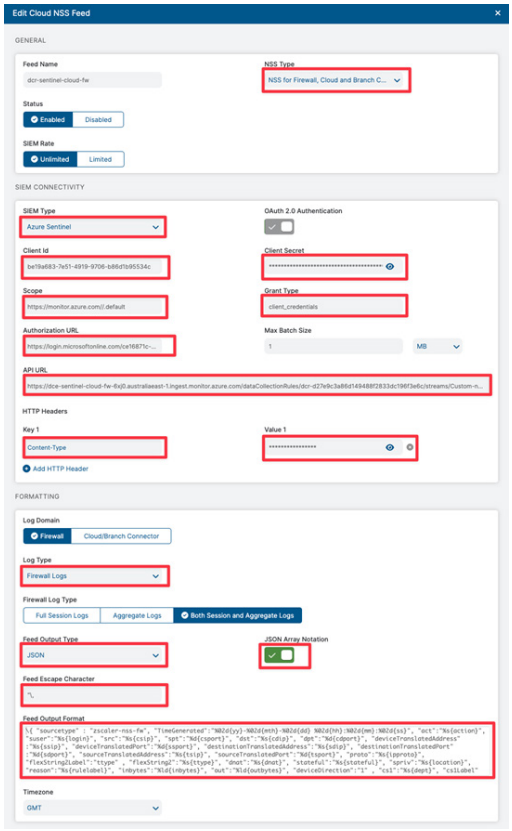


Figure 33. Add Cloud NSS Feed for Firewall

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for DNS Logs

For Cloud NSS Feeds for DNS Logs (government agencies, see Cloud NSS Feeds for DNS Logs), copy and paste the prepopulated **Feed Output Format** from the file cloud-nss-dns.fof in Zscaler GitHub repo, or use the following:

```
\{"sourcetype":"zscaler-nss-fw","TimeGenerated":"%02d{yy}-%02d{mth}-%02d{dd}
%02d{hh}:%02d{mm}:%02d{ss}", "suser":"%s{login}","act":"%s{action}", "rulelabel":
"%s{rulelabel}" ,"cat":"%s{domcat}", "cs1":"%s{dept}","cs1Label":"department", "cs2":
"%s{reqaction}","cs2Label":"reqaction","cs3":"%s{resaction}","cs3Label": "resaction",
"cs4":"%s{reqtype}","cs4Label":"dns_reqtype","cs5":"%s{req}","cs5Label": "dns_req","c
s6":"%s{res}","cs6Label":"dns_resp","cn1":"%d{durationms}","cn1Label": "durationms","
flexString1":"%s{reqrulelabel}","flexString1Label":"reqrulelabel","flexString2":"%s{resr
ulelabel}","flexString2Label":"resrulelabel", "cat":"%s{domcat}","src":"%s{cip}","dst"
:"%s{sip}", "dpt":"%d{sport}","spriv":"%s{location}","suid":"%s{deviceowner}", "dvcho
st":"%s{devicehostname}","DeviceVendor":"Zscaler" , "DeviceProduct":"NSSDNSlog"\}
```

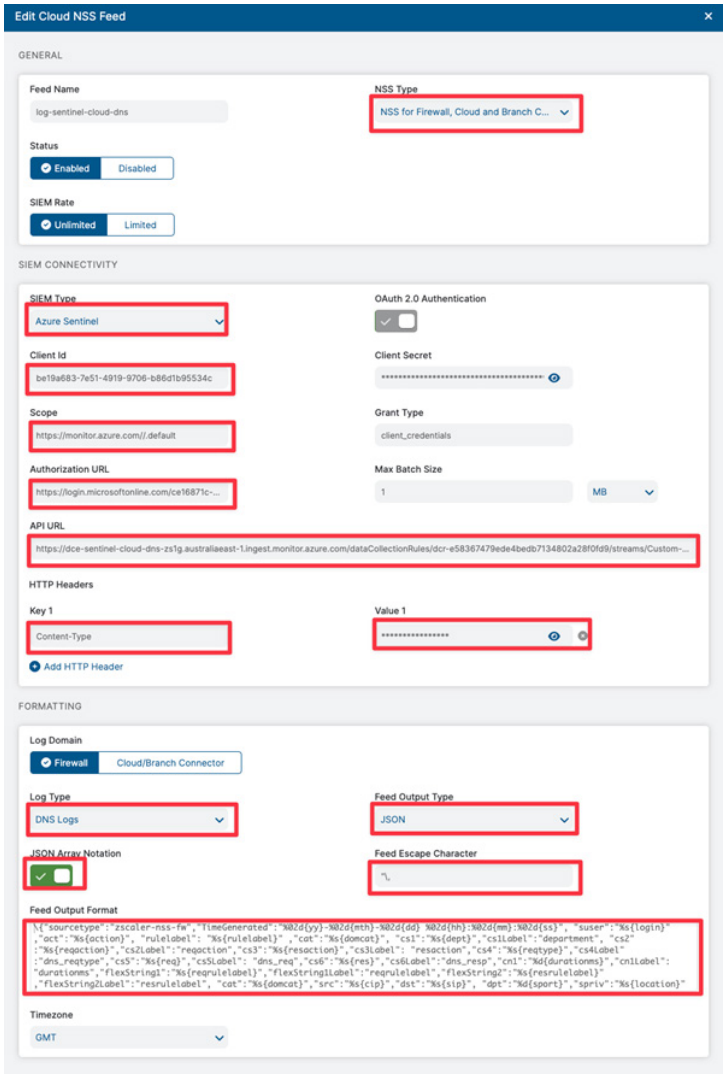


Figure 34. Add Cloud NSS Feed for DNS logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Admin Audit Logs

For Cloud NSS Feeds for Admin Audit Logs (government agencies, see Cloud NSS Feeds for Admin Audit Logs), copy and paste the prepopulated **Feed Output Format** from the file cloud-nss-audit.fof in the Zscaler GitHub repo, or use the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype"
:"zscalernss-audit","vendor":"Zscaler","product":"NSSAuditlog","version":"5.7","Sever
ity":"3","name":"%s{category}","deviceEventClassId":"%s{action}","act":"%s{action}","
cs1Label":"category","cs1":"%s{category}","cs2Label":"subcategory","cs2":"%s{subcateg
ory}","cs3Label":"resource","cs3":"%s{resource}","cat":"%s{interface}","suser":"%s{ad
minid}","src":"%s{clientip}","outcome":"%s{result}","reason":"%s{errorcode}","sourceS
erviceName":"%s{auditlogtype}","cs4Label":"preaction","cs4":"%s{epreaction}","cs5Labe
l":"postaction","cs5":"%s{epostaction}","deviceExternalId":"%d{recordid}"\}
```

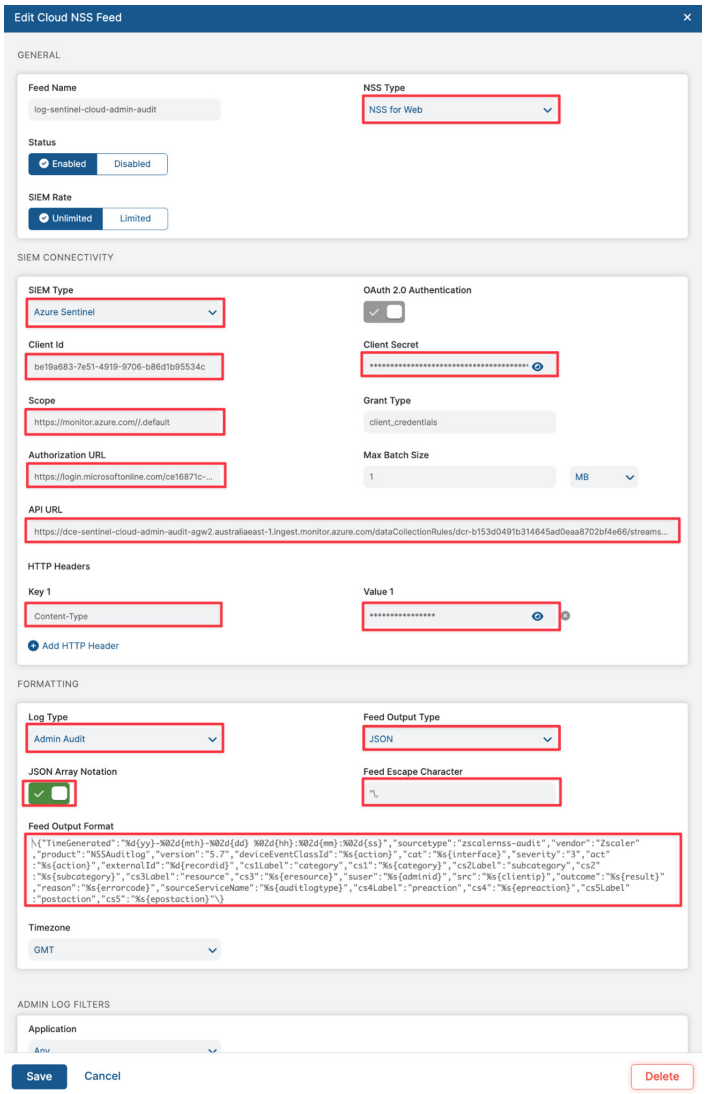


Figure 35. Add Cloud NSS Feed for Audit logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Tunnel Logs

For Cloud NSS Feeds for Tunnel Logs, (government agencies, see Cloud NSS Feeds for Tunnel Logs), copy and paste the prepopulated **Feed Output Format** from the file cloud-nss-tunnel.fof in Zscaler GitHub repo, or use the following:

Tunnel logs - IKE Phase 1

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel CEF:0|Zsca
ler|NSSTunnellog|5.7|%s{tunnelactionname}|%s{destvip}|3| dpt=%d{dstport}
cn1Label=ikeversion cn1=%d{ikeversion} cfp1Label=lifetime cfp1=%d{lifetime}
spt=%d{srcport} spi_in=%lu{spi_in} spi_out=%lu{spi_out} cs1Label=algo cs1=%s{algo}
cs2Label=authentication cs2=%s{authentication} cs3Label=authtype cs3=%s{authtype}
dst=%s{destvip} cs4Label=locationname cs4=%s{locationname} src=%s{sourceip} deviceEve
ntClassId=%s{tunnelactionname} vendorname=%s{vendorname} suser=%s{vpncredentialname}
deviceExternalId=%d{recordid} olocationname=%s{olocationname} ovpncredentialname=%s{o
vpncredentialname} dtz=%s{tz}\n
```

Tunnel logs - IKE Phase 2

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel
CEF:0|Zscaler|NSSTunnellog|5.7|%s{tunnelactionname}|%s{tunnelprotocol}|3|
destipstart=%s{destipstart} dst=%s{destvip} cs4Label=locationname
cs4=%s{locationname} proto=%s{protocol} src=%s{sourceip} srcipend=%s{srcipend}
srcipstart=%s{srcipstart} deviceEventClassId=%s{tunnelactionname} proto=%s{protocol}
suser=%s{vpncredentialname} destportstart=%d{destportstart} cn1Label=ikeversion
cn1=%d{ikeversion} lifebytes=%d{lifebytes} cfp1Label=lifetime cfp1=%d{lifetime}
spi=%d{spi} srcportstart=%d{srcportstart} cs1Label=algo cs1=%s{algo}
cs2Label=authentication cs2=%s{authentication} cs3Label=authtype cs3=%s{authtype}
destipend=%s{destipend} \n
```

Tunnel Events

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel CEF:0|Zscaler|NSS
Tunnellog|5.7|%s{tunnelactionname}|%s{event}|3| spt=%d{srcport} dst=%s{destvip}
name=%s{event} reason=%s{eventreason} cs4Label=locationname cs4=%s{locationname}
src=%s{sourceip} deviceEventClassId=%s{tunnelactionname} cs5Label=tunneltype
cs5=%s{tunneltype} suser=%s{vpncredentialname} deviceExternalId=%d{recordid}
olocationname=%s{olocationname} ovpncredentialname=%s{ovpncredentialname} dtz=%s{tz}
\n
```

Tunnel Samples

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel CEF:0|Zscal
er|NSSTunnellog|5.7|%s{tunnelactionname}|%s{tunneltype}|3| spt=%d{srcport}
cn2Label=rxpackets cn2=%d{rxpackets} cn3Label=txpackets cn3=%d{txpackets}
dpdrec=%d{dpdrec} in=%lu{rxbytes} out=%lu{txbytes} dst=%s{destvip}
cs4Label=locationname cs4=%s{locationname} src=%s{sourceip} deviceEventClassId=%s{t
unnelactionname} cs5Label=tunneltype cs5=%s{tunneltype} suser=%s{vpncredentialname}
deviceExternalId=%d{recordid} olocationname=%s{olocationname} ovpncredentialname=%s{o
vpncredentialname} dtz=%s{tz} \n
```

©2026 Zscaler, Inc. All rights reserved.

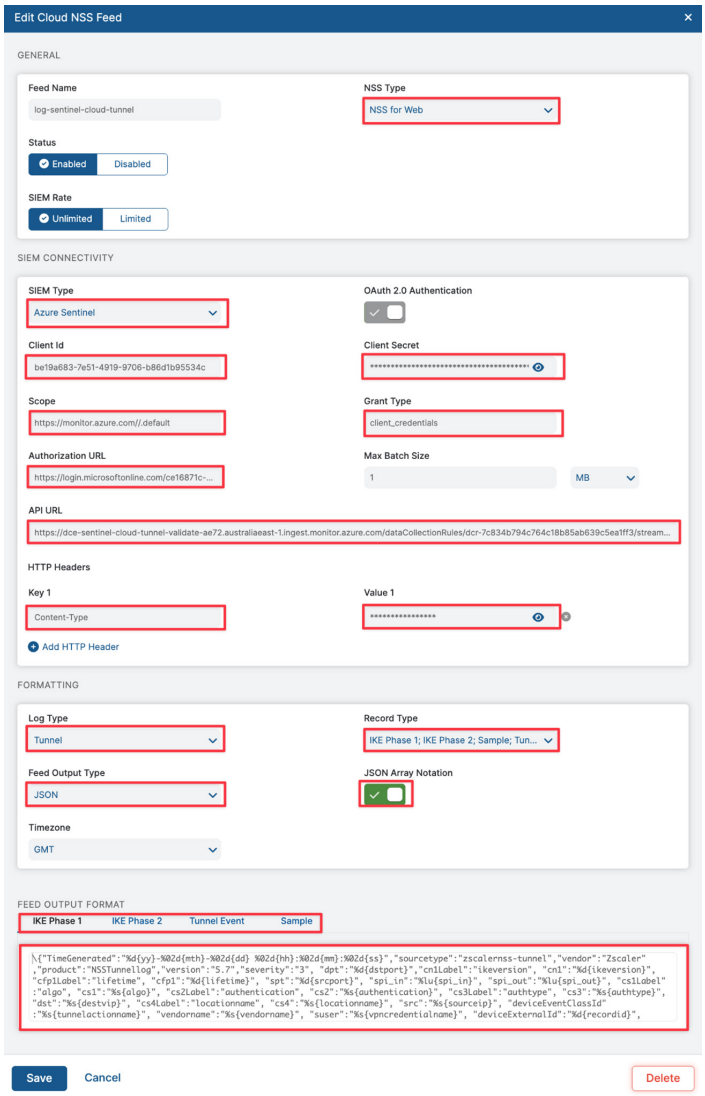


Figure 36. Add Cloud NSS Feed for Tunnel logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Endpoint DLP Logs

For Cloud NSS Feeds for Endpoint DLP Logs (government agencies, see Cloud NSS Feeds for Endpoint DLP Logs), copy and paste the prepopulated **Feed Output Format** from the file cloud-nss-endpointdlp.fof in the Zscaler GitHub repo, or use the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype"
:"zscalernss-edlp","vendor":"Zscaler","product":"NSSEndpointdlplog","version":"5.7","
Severity":"3","name":"%s{activitytype}","rt":"%s{time}","deviceCustomDate1Label":"rti
me","deviceCustomDate1":"%s{rtime}","deviceCustomDate2Label":"eventtime","deviceCusto
mDate2":"%s{eventtime}","dtz":"%s{tz}","deviceCustomNumber1Label":"scantime","cn1":"%
llu{scantime}","deviceCustomNumber2Label":"numdlpengids","cn2":"%u{numdlpengids}","de
viceCustomNumber3Label":"numdlpdictids","cn3":"%u{numdlpdictids}","externalId":"%llu{
recordid}","scanned_bytes":"%llu{scanned_bytes}","dlpidentifier":"%llu{dlpidentifier}",
"suser":"%s{user}","ouser":"%s{ouser}","department":"%s{department}","odepartment":"%
s{odepartment}","devicename":"%s{devicename}","odevicename":"%s{odevicename}","device
type":"%s{devicetype}","deviceostype":"%s{deviceostype}","deviceplatform":"%s{devicep
latform}","deviceosversion":"%s{deviceosversion}","devicemodel":"%s{devicemodel}","de
viceappversion":"%s{deviceappversion}","deviceowner":"%s{deviceowner}","odeviceowner"
:"%s{odeviceowner}","dvchost":"%s{devicehostname}","odevicehostname":"%s{odevicehostn
ame}","datacenter":"%s{datacenter}","datacentercity":"%s{datacentercity}","datacenter
country":"%s{datacentercountry}","dsttype":"%s{dsttype}","filedoctype":"%s{filedoctype}
","filedstpath":"%s{filedstpath}","fileHash":"%s{filemd5}","filesha":"%s{filesha}","filePath
":"%s{filesrcpath}","filetypecategory":"%s{filetypecategory}","fileType":"%s{filetypename}
","itemdstname":"%s{itemdstname}","itemname":"%s{itemname}","itemsrcname":"%s{itemsrc
name}","itemtype":"%s{itemtype}","ofiledstpath":"%s{ofiledstpath}","ofilesrcpath":"%s{ofi
lesrcpath}","oitemdstname":"%s{oitemdstname}","oitemname":"%s{oitemname}","oitemsrcna
me":"%s{oitemsrcname}","sourceServiceName":"%s{srctype}","act":"%s{actiontaken}","cs1
Label":"activitytype","cs1":"%s{activitytype}","cs2Label":"addinfo","cs2":"%s{addinfo
}","cs3Label":"channel","cs3":"%s{channel}","cs4Label":"confirmaction","cs4":"%s{confir
maction}","cs5Label":"confirmjust","cs5":"%s{confirmjust}","cs6Label":"dlpcounts","cs6"
:"%s{dlpcounts}","dlpdictnames":"%s{dlpdictnames}","dlpengnames":"%s{dlpengnames}","e
xpectedaction":"%s{expectedaction}","deviceEventClassId":"%s{logtype}","odlpdictnames
":"%s{odlpdictnames}","odlpengnames":"%s{odlpengnames}","ootherrulelabels":"%s{oother
rulelabels}","otherrulelabels":"%s{otherrulelabels}","otriggeredrulelabel":"%s{otrigg
eredrulelabel}","agentSeverity":"%s{severity}","triggeredrulelabel":"%s{triggeredrule
label}","zdpmode":"%s{zdpmode}"\}
```

©2026 Zscaler, Inc. All rights reserved.

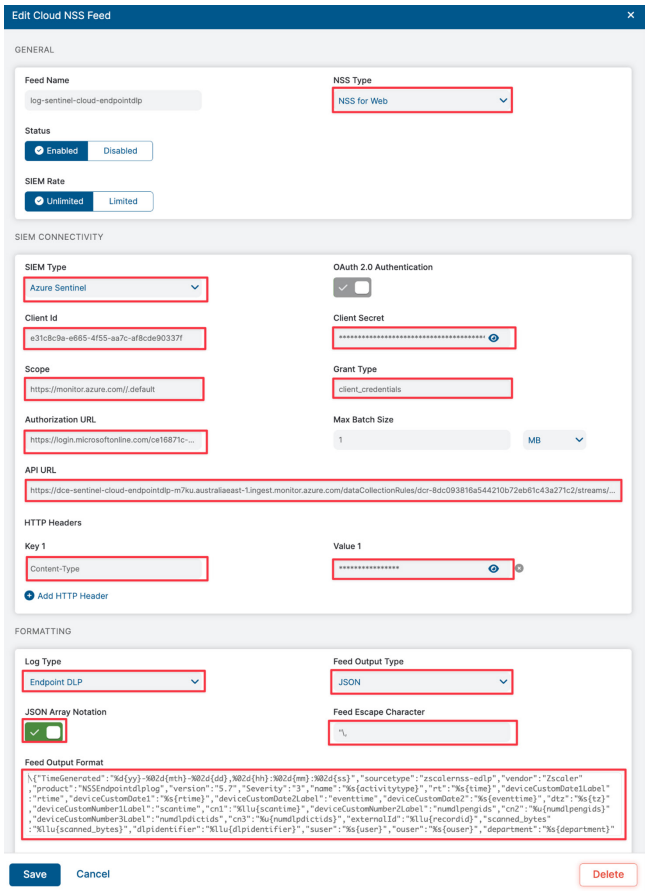


Figure 37. Add Cloud NSS Feed for Endpoint DLP logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Email DLP Logs

For Cloud NSS Feeds for Email DLP Logs (government agencies, see Cloud NSS Feeds for Email DLP Logs) copy and paste the prepopulated Feed Output Format from the file cloud-nss-emaildlp.fof in the Zscaler GitHub repo, or use the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype"
:"zscalernss-emaildlp","vendor":"Zscaler","product":"NSSEmaildlplog","version":"5.7",
"Severity":"3","act":"%s{actions}","deviceEventClassId":"%s{logtype}","agentSeverity"
:"%s{severity}","app":"%s{appname}","cn1":"%llu{scan_time}","cn1Label":"scan_time","
cn2":"%llu{dlpidentifier}","cn2Label":"dlpidentifier","cs1":"%s{dlpdictnames}","cs1Lab
el":"dlpdictnames","cs2":"%s{dlpdictcnts}","cs2Label":"dlpdictcnts","cs3":"%s{dlpeng
names}","cs3Label":"dlpengnames","cs4":"%s{rulelabels}","cs4Label":"rulelabels","cs5
":"%s{mail_sent_time}","cs5Label":"mail_sent_time","cs6":"%s{subject}","cs6Label":"s
ubject","suser":"%s{owner}","username":"%s{username}","extusername":"%s{extusername}
","sender":"%s{sender}","flexString1":"%s{trigg_rcpt_doms}","flexString1Label":"trigg_
rcpt_doms","flexString2":"%s{other_rcpt_doms}","flexString2Label":"other_rcpt_doms","da
tacenter":"%s{datacenter}","datacentercity":"%s{datacentercity}","datacentercountry":
"%s{datacentercountry}","company":"%s{company}","departmentname":"%s{departmentname}"
,"tenant":"%s{tenant}","msgid":"%s{msgid}","ac_md5s":"%s{ac_md5s}","ac_sizes":"%s{ac_
sizes}","ac_filetypes":"%s{ac_filetypes}","ac_doctypes":"%s{ac_doctypes}","ac_
names":"%s{ac_names}","trigg_rcpts":"%s{trigg_rcpts}","other_rcpts":"%s{other_rcpts}"
,"deviceExternalId":"%llu{recordid}","Name":"%s{actions}"\}
```

©2026 Zscaler, Inc. All rights reserved.

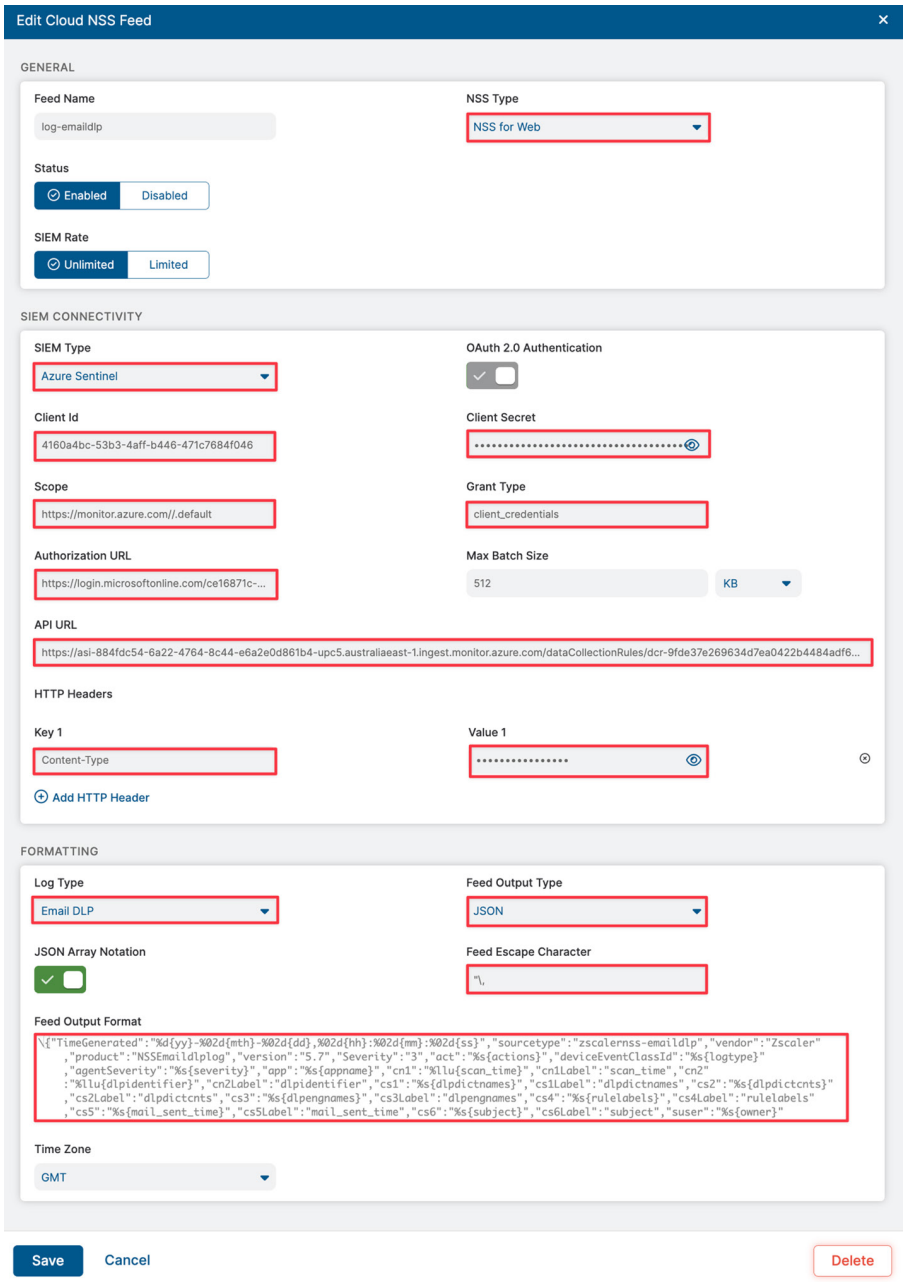


Figure 38. *Add Cloud NSS Feed for Email DLP logs*

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for SaaS Security Activity Logs

For Cloud NSS Feeds for SaaS Security Activity Logs (government agencies, see Cloud NSS Feeds for SaaS Security Activity Logs), copy and paste the prepopulated **Feed Output Format** from the file cloud-nss-casbactivity.fof in the Zscaler GitHub repo, or use the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype":"zscalernss-casb","vendor":"Zscaler","product":"NSSCasbactivitylog","version":"5.7","Severity":"3","name":"%s{act_type_name}","suser":"%s{username}","cs1Label":"tenant","cs1":"%s{tenant}","cs2Label":"objtypename1","cs2":"%s{objtypename1}","cs3Label":"objtypename2","cs3":"%s{objtypename2}","deviceEventClassId":"%s{appname}","cs4Label":"objnames1","cs4":"%s{objnames1}","cs5Label":"objnames2","cs5":"%s{objtypename2}","eventCount":"%d{act_cnt}","act":"%s{act_type_name}","cs6Label":"extownername","cs6":"%s{extownername}","src":"%s{src_ip}","flexString1Label":"isadmin","flexString1":"%s{is_admin_act}"\}
```

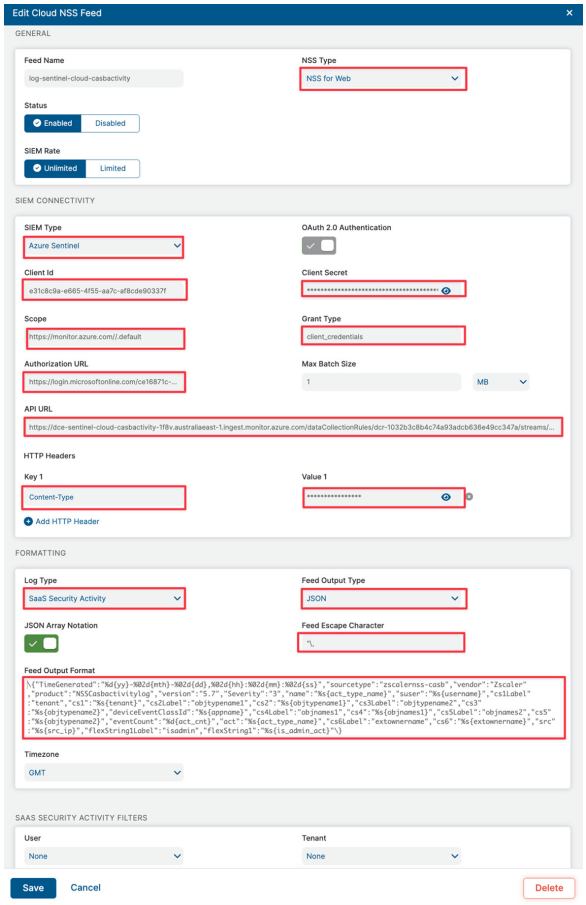


Figure 39. Add Cloud NSS Feed for SaaS Security Activity logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for SaaS Security Logs

For Cloud NSS Feeds for SaaS Security Logs (government agencies, see Cloud NSS Feeds for SaaS Security Logs), copy and paste the prepopulated **Feed Output Format** from the Zscaler GitHub repo for one of the following files depending on which feed you want to ingest:

- Public Cloud Storage: cloud-nss-casbcloudstorage.fof
- SaaS Security Collaboration: cloud-nss-casbcollab.fof
- SaaS Security CRM: cloud-nss-casbcrm.fof
- SaaS Security Email: cloud-nss-casbemail.fof
- SaaS Security File: cloud-nss-casbfilesharing.fof
- SaaS Security ITSM: cloud-nss-casbitsm.fof
- SaaS Security Repository: cloud-nss-casbrepo.fof

SaaS Public Cloud Storage

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype"
:"zscalernss-casb","vendor":"Zscaler","product":"NSSCasbcloudstoragelog","version":"5
.7","Severity":"3","name":"%s{rulelabel}","deviceEventClassId":"%s{applicationname}",
"act":"%s{policy}","bucketid":"%d{bucketid}","cs1":"%d{dlpidentifier}","cs1Label":"dlp
identifier","start":"%d{epochtime}","numcollab":"%d{numcollab}","deviceExternalId":"%d
{recordid}","bucketname":"%s{bucketname}","bucketowner":"%s{bucketowner}","collabname
s":"%s{collabnames}","company":"%s{company}","datacenter":"%s{datacenter}","datacente
rcity":"%s{datacentercity}","datacentercountry":"%s{datacentercountry}","department":
"%s{department}","cs2":"%s{dlpdictcount}","cs2Label":"dlpdictcount","cs3":"%s{dlpdict
names}","cs3Label":"dlpdictnames","cs4":"%s{dlpenginenames}","cs4Label":"dlpenginenam
es","extownername":"%s{extownername}","fileId":"%s{fileid}","request":"%s{fullurl}","dh
ost":"%s{hostname}","flexString1Label":"malware","flexString1":"%s{malware}","flexStrin
g2Label":"malwareclass","flexString2":"%s{malwareclass}","obucketname":"%s{obucketnam
e}","obucketowner":"%s{obucketowner}","ocollabnames":"%s{ocollabnames}","odlpdictnam
es":"%s{odlpdictnames}","odlpenginenames":"%s{odlpenginenames}","oextownername":"%s{o
extownername}","ofileid":"%s{ofileid}","ofullurl":"%s{ofullurl}","ohostname":"%s{ohostn
ame}","oowner":"%s{oowner}","orulelabel":"%s{orulelabel}","otenant":"%s{otenant}","su
ser":"%s{owner}","cs5":"%s{rulelabel}","cs5Label":"rulelabel","cs6":"%s{ruletype}","c
s6Label":"ruletype","agentSeverity":"%s{severity}","tenant":"%s{tenant}","threatname"
:"%s{threatname}","deviceCustomDate1Label":"time","deviceCustomDate1":"%s{time}","dtz
":"%s{tz}","upload_doctypename":"%s{upload_doctypename}"\}
```

©2026 Zscaler, Inc. All rights reserved.

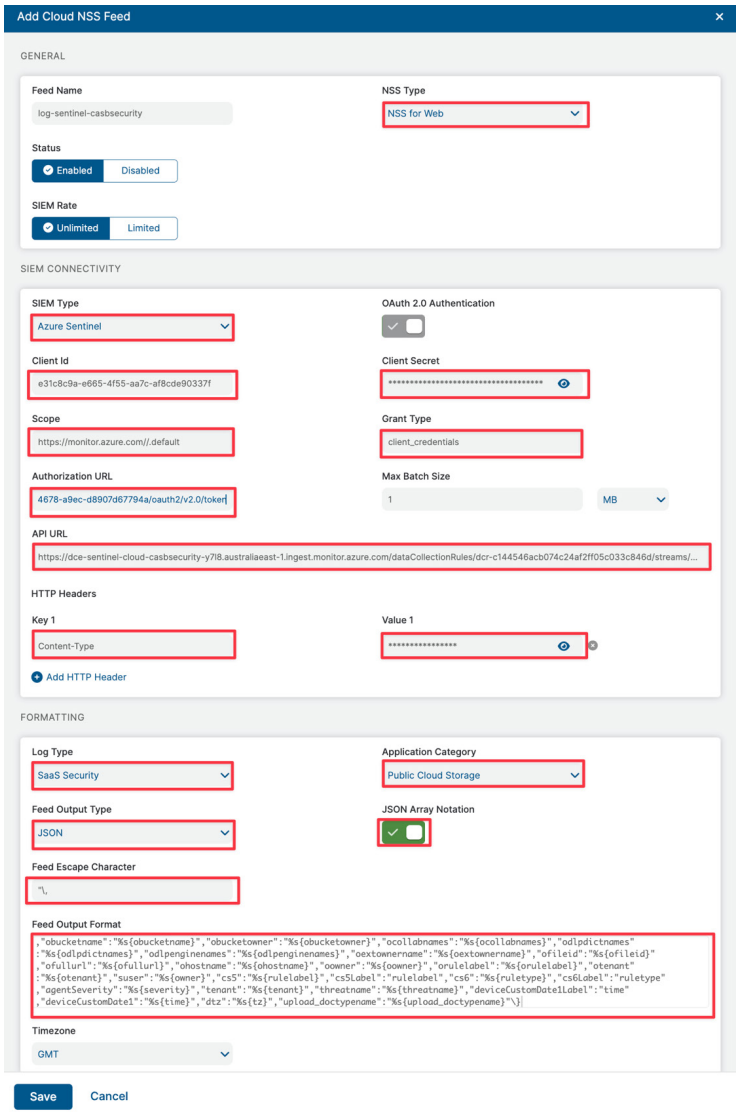


Figure 40. Add Cloud NSS Feed for SaaS Security Public Cloud Storage logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

SaaS Security Collaboration

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetyp
e":"zscalernss-casb","vendor":"Zscaler","product":"NSSCasbcollablog","version":"5.7
","Severity":"3","name":"%s{rulelabel}","deviceEventClassId":"%s{applicationname}",
"act":"%s{policy}","cs1":"%d{dlpidentifier}","cs1Label":"dlpidentifier","start":"%d{e
pochtime}","deviceExternalId":"%d{recordid}","outcome":"%s{any_incident}","channel_
name":"%s{channel_name}","company":"%s{company}","datacenter":"%s{datacenter}","dat
acentercity":"%s{datacentercity}","datacentercountry":"%s{datacentercountry}","dep
artment":"%s{department}","cs2":"%s{dlpdictcount}","cs2Label":"dlpdictcount","cs3"
:"%s{dlpdictnames}","cs3Label":"dlpdictnames","cs4":"%s{dlpenginenames}","cs4Label
":"dlpenginenames","external_recptnames":"%s{external_recptnames}","extownername":
"%s{extownername}","internal_recptnames":"%s{internal_recptnames}","flexString1Labe
l":"malware","flexString1":"%s{malware}","flexString2Label":"malwareclass","flexStrin
g2":"%s{malwareclass}","msgid":"%s{msgid}","ochannel_name":"%s{ochannel_name}","od
lpdictnames":"%s{odlpdictnames}","odlpenginenames":"%s{odlpenginenames}","oextern
al_recptnames":"%s{oexternal_recptnames}","oextownername":"%s{oextownername}","ointer
nal_recptnames":"%s{ointernal_recptnames}",
"omsgid":"%s{omsgid}","oowner":"%s{oowner}","orulelabel":"%s{orulelabel}"
,"osender":"%s{osender}","osharedchannel_hostname":"%s{osharedchannel_ho
stname}","otenant":"%s{otenant}","suser":"%s{owner}","cs5":"%s{rulelab
el}","cs5Label":"rulelabel","cs6":"%s{ruletype}","cs6Label":"ruletype"
,"sender":"%s{sender}","agentSeverity":"%s{severity}","sharedchannel_
hostname":"%s{sharedchannel_hostname}","tenant":"%s{tenant}","threatname":"%s{threatn
ame}","deviceCustomDate1Label":"time","deviceCustomDate1":"%s{time}","dtz":"%s{tz}","
upload_doctypename":"%s{upload_doctypename}"\}
```

©2026 Zscaler, Inc. All rights reserved.

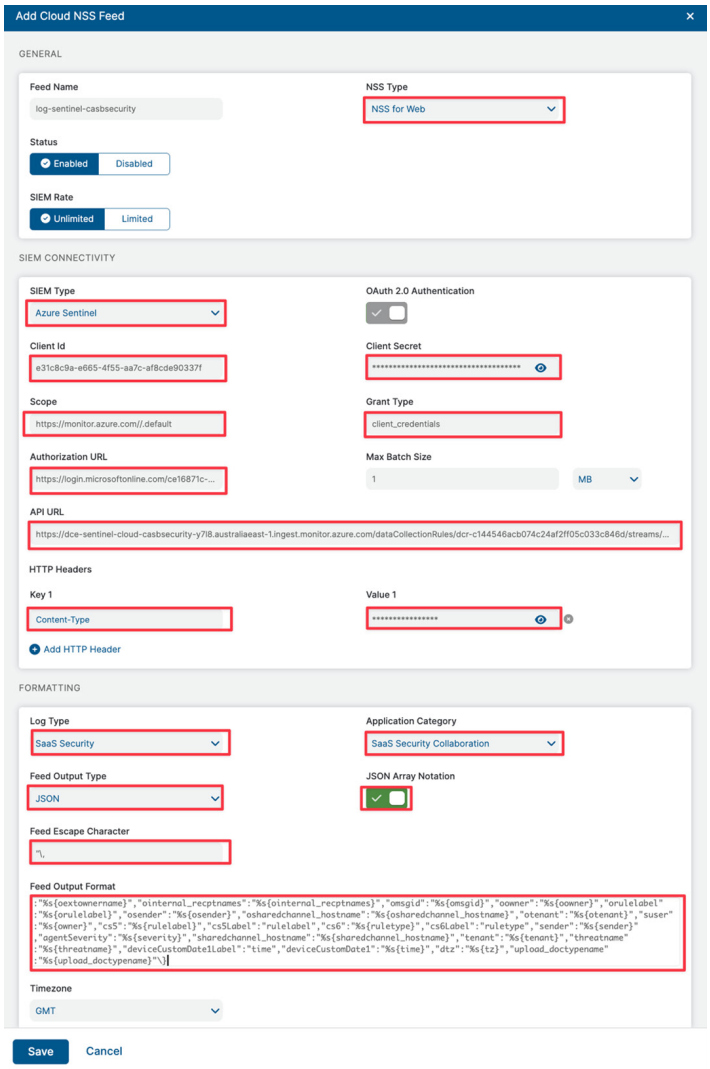


Figure 41. Add Cloud NSS Feed for SaaS Security Collaboration logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

SaaS Security CRM

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcety
pe":"zscalernss-casb","vendor":"Zscaler","product":"NSSCasbcrmlog","version":"5.7"
,"Severity":"3","name":"%s{rulelabel}","deviceEventClassId":"%s{applicationname}",
"act":"%s{policy}","cs1":"%d{dlpidentifier}","cs1Label":"dlpidentifier","start":"%d{
epochtime}","fsize":"%d{filesize}","cn1":"%d{num_external_collab}","cn1Label":"num_
external_collab","cn2":"%d{num_internal_collab}","cn2Label":"num_internal_collab"
,"deviceExternalId":"%d{recordid}","collabscope":"%s{collabscope}","company":"%s{
company}","component":"%s{component}","datacenter":"%s{datacenter}","datacenterci
ty":"%s{datacentercity}","datacentercountry":"%s{datacentercountry}","department"
:"%s{department}","cs2":"%s{dlpdictcount}","cs2Label":"dlpdictcount","cs3":"%s{dl
pdictnames}","cs3Label":"dlpdictnames","cs4":"%s{dlpenginenames}","cs4Label":"dlp
enginenames","external_collabnames":"%s{external_collabnames}","extownername":"%s
{extownername}","fileId":"%s{file_msg_id}","fileModificationTime":"%s{file_msg_mod_tim
e}","fileHash":"%s{filemd5}","fname":"%s{filename}","filePath":"%s{filepath}","fileType
":"%s{filetypecategory}","request":"%s{fullurl}","dhost":"%s{hostname}","internal_
collabnames":"%s{internal_collabnames}","flexString1Label":"malware","flexString1":"%s{
malware}","flexString2Label":"malwareclass","flexString2":"%s{malwareclass}","objectna
me":"%s{objectname}","objecttype":"%s{objecttype}","odlpdictnames":"%s{odlpdictname
s}","odlpenginenames":"%s{odlpenginenames}","oexternal_collabnames":"%s{oexternal_
collabnames}","oextownername":"%s{oextownername}","ofile_msg_id":"%s{ofile_
msg_id}","ofullurl":"%s{ofullurl}","ohostname":"%s{ohostname}","ointernal_
collabnames":"%s{ointernal_collabnames}","oowner":"%s{oowner}","orulelabel":"%s{orul
elabel}","otenant":"%s{otenant}","suser":"%s{owner}","cs5":"%s{rulelabel}","cs5Label
":"rulelabel","cs6":"%s{ruletype}","cs6Label":"ruletype","agentSeverity":"%s{severit
y}","tenant":"%s{tenant}","threatname":"%s{threatname}","deviceCustomDate1Label":"t
ime","deviceCustomDate1":"%s{time}","dtz":"%s{tz}","upload_doctypename":"%s{upload_
doctypename}"\}
```

©2026 Zscaler, Inc. All rights reserved.

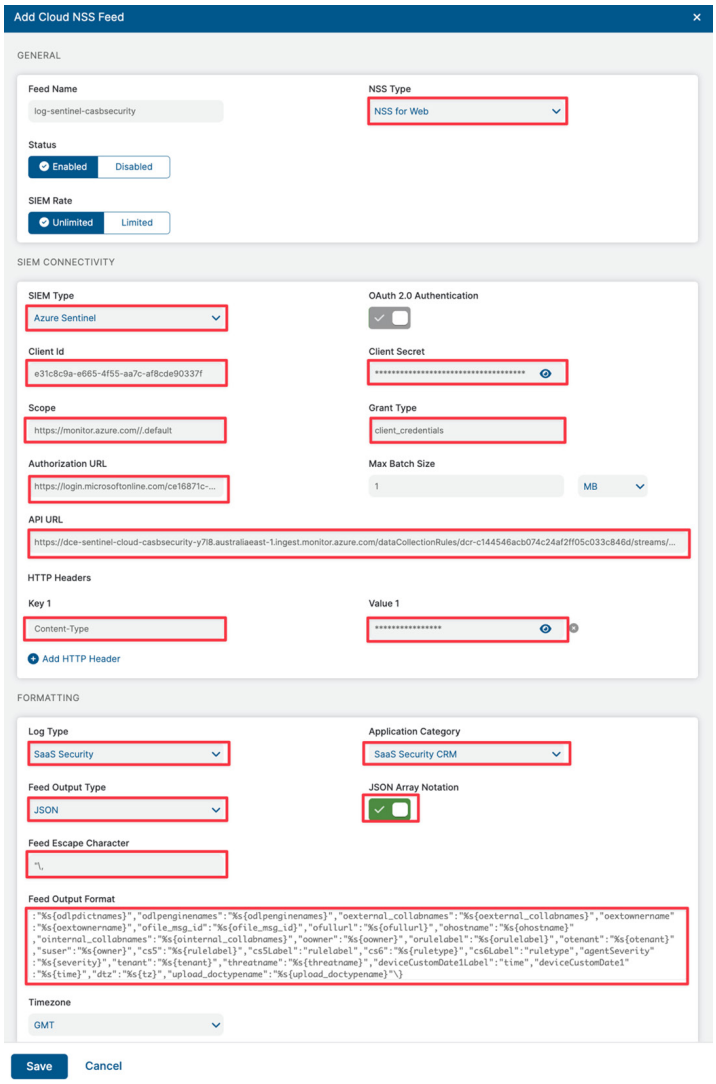


Figure 42. Add Cloud NSS Feed for SaaS Security CRM logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

SaaS Security Email

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype"
:"zscalernss-casb","vendor":"Zscaler","product":"NSSCasbemaillog","version":"5.7","Se
verity":"3","name":"%s{rulelabel}","deviceEventClassId":"%s{applicationname}","act":"
%s{policy}","oexternalownername":"%s{oexternalownername}","companyid":"%d{companyid}"
,"dlpidentifier":"%d{dlpidentifier}","epochtime":"%d{epochtime}","filedownloadtimems":"%
d{filedownloadtimems}","filescantimems":"%d{filescantimems}","msgsize":"%d{msgsize}","n
um_ext_recpts":"%d{num_ext_recpts}","num_int_recpts":"%d{num_int_recpts}","deviceExte
rnalId":"%d{recordid}","repochtime":"%d{repochtime}","outcome":"%s{any_incident}","fn
ame":"%s{attchcomponentfilenames}","fsize":"%s{attchcomponentfilenames}","fileType":"%s{
attchcomponentfiletypes}","fileHash":"%s{attchcomponentmd5s}","company":"%s{company}","
datacenter":"%s{datacenter}","datacentercity":"%s{datacentercity}","datacentercountry
":"%s{datacentercountry}","department":"%s{department}","dlpdictcount":"%s{dlpdictcou
nt}","dlpdictnames":"%s{dlpdictnames}","dlpenginenames":"%s{dlpenginenames}","extern
alownername":"%s{externalownername}","extrecptnames":"%s{extrecptnames}","intrecptna
mes":"%s{intrecptnames}","is_inbound":"%s{is_inbound}","malware":"%s{malware}","malw
areclass":"%s{malwareclass}","messageid":"%s{messageid}","oattchcomponentfilenames":"
%s{oattchcomponentfilenames}","odlpdictnames":"%s{odlpdictnames}","odlpenginenames":"
%s{odlpenginenames}","oextrecptnames":"%s{oextrecptnames}","ointrecptnames":"%s{oint
recptnames}","omessageid":"%s{omessageid}","oowner":"%s{oowner}","orulelabel":"%s{oru
lelabel}","otenant":"%s{otenant}","owner":"%s{owner}","rtime":"%s{rtime}","rulelabel
":"%s{rulelabel}","ruletype":"%s{ruletype}","severity":"%s{severity}","tenant":"%s{t
enant}","threatname":"%s{threatname}","tz":"%s{tz}","upload_doctypename":"%s{upload_
doctypename}"\}
```

©2026 Zscaler, Inc. All rights reserved.

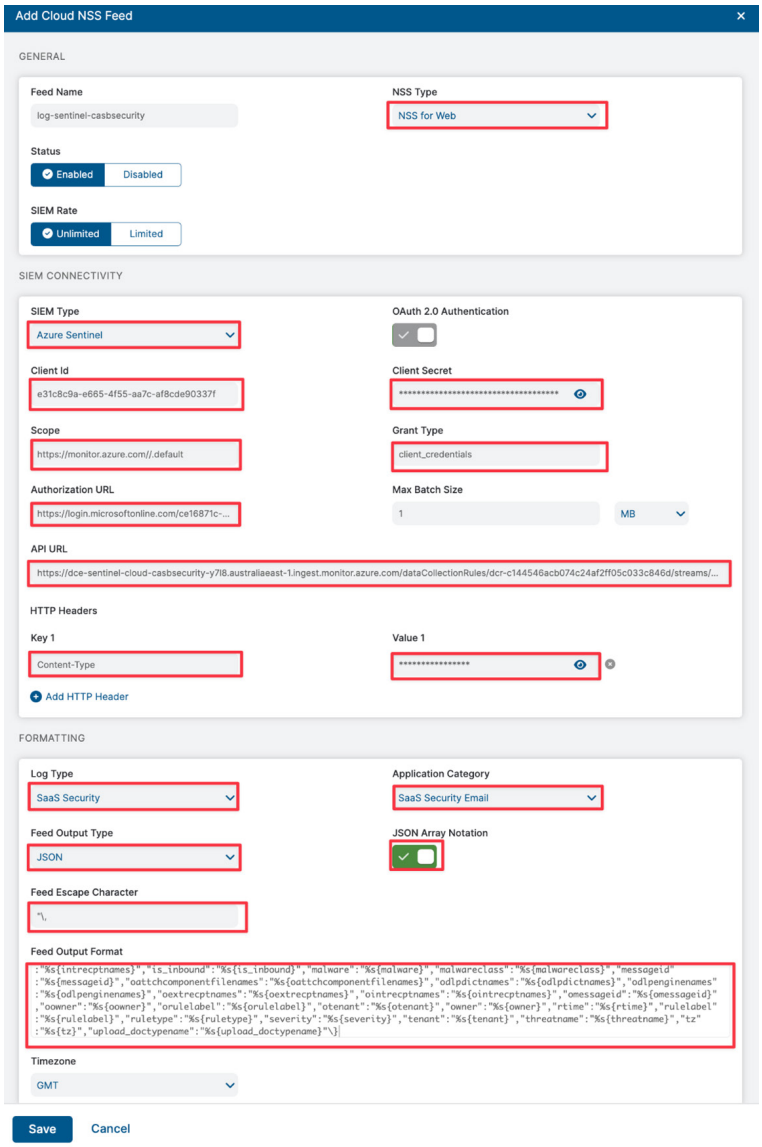


Figure 43. Add Cloud NSS Feed for SaaS Security Email logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

SaaS Security File

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype"
:"zscalernss-casb","vendor":"Zscaler","product":"NSSCasbfilesharinglog","version":"5.7
","Severity":"3","name":"%s{rulelabel}","deviceEventClassId":"%s{applicationname}","a
ct":"%s{policy}","cs1":"%d{dlpidentifier}","cs1Label":"dlpidentifier","fileModificationTi
me":"%d{epochlastmodtime}","start":"%d{epochtime}","filedownloadtimems":"%d{filedownloa
dtimems}","filescantimems":"%d{filescantimems}","fsize":"%d{filesize}","deviceExternalId
":"%d{recordid}","collabscope":"%s{collabscope}","company":"%s{company}","datacenter"
:"%s{datacenter}","datacentercity":"%s{datacentercity}","datacentercountry":"%s{datac
entercountry}","department":"%s{department}","cs2":"%s{dlpdictcount}","cs2Label":"dlp
dictcount","cs3":"%s{dlpdictnames}","cs3Label":"dlpdictnames","cs4":"%s{dlpenginename
s}","cs4Label":"dlpenginenames","extcollabnames":"%s{extcollabnames}","extownername":
"%s{extownername}","fileId":"%s{fileid}","fileHash":"%s{filemd5}","fname":"%s{filename}","
filePath":"%s{filesource}","fileType":"%s{filetypename}","request":"%s{fullurl}","dhost"
:"%s{hostname}","intcollabnames":"%s{intcollabnames}","flexString1Label":"malware","fle
xString1":"%s{malware}","flexString2Label":"malwareclass","flexString2":"%s{malwareclas
s}","odlpdictnames":"%s{odlpdictnames}","odlpenginenames":"%s{odlpenginenames}","oext
collabnames":"%s{oextcollabnames}","oextownername":"%s{oextownername}","ofileid":"%s{o
fileid}","efullurl":"%s{efullurl}","ehostname":"%s{ehostname}","ointcollabnames":"%s{o
intcollabnames}","orulelabel":"%s{orulelabel}","otenant":"%s{otenant}","ouser":"%s{ou
ser}","cs5":"%s{rulelabel}","cs5Label":"rulelabel","cs6":"%s{ruletype}","cs6Label":"r
uletype","agentSeverity":"%s{severity}","tenant":"%s{tenant}","threatname":"%s{threat
name}","deviceCustomDate1Label":"time","deviceCustomDate1":"%s{time}","dtz":"%s{tz}",
"upload_doctypename":"%s{upload_doctypename}","suser":"%s{user}"\}
```

©2026 Zscaler, Inc. All rights reserved.

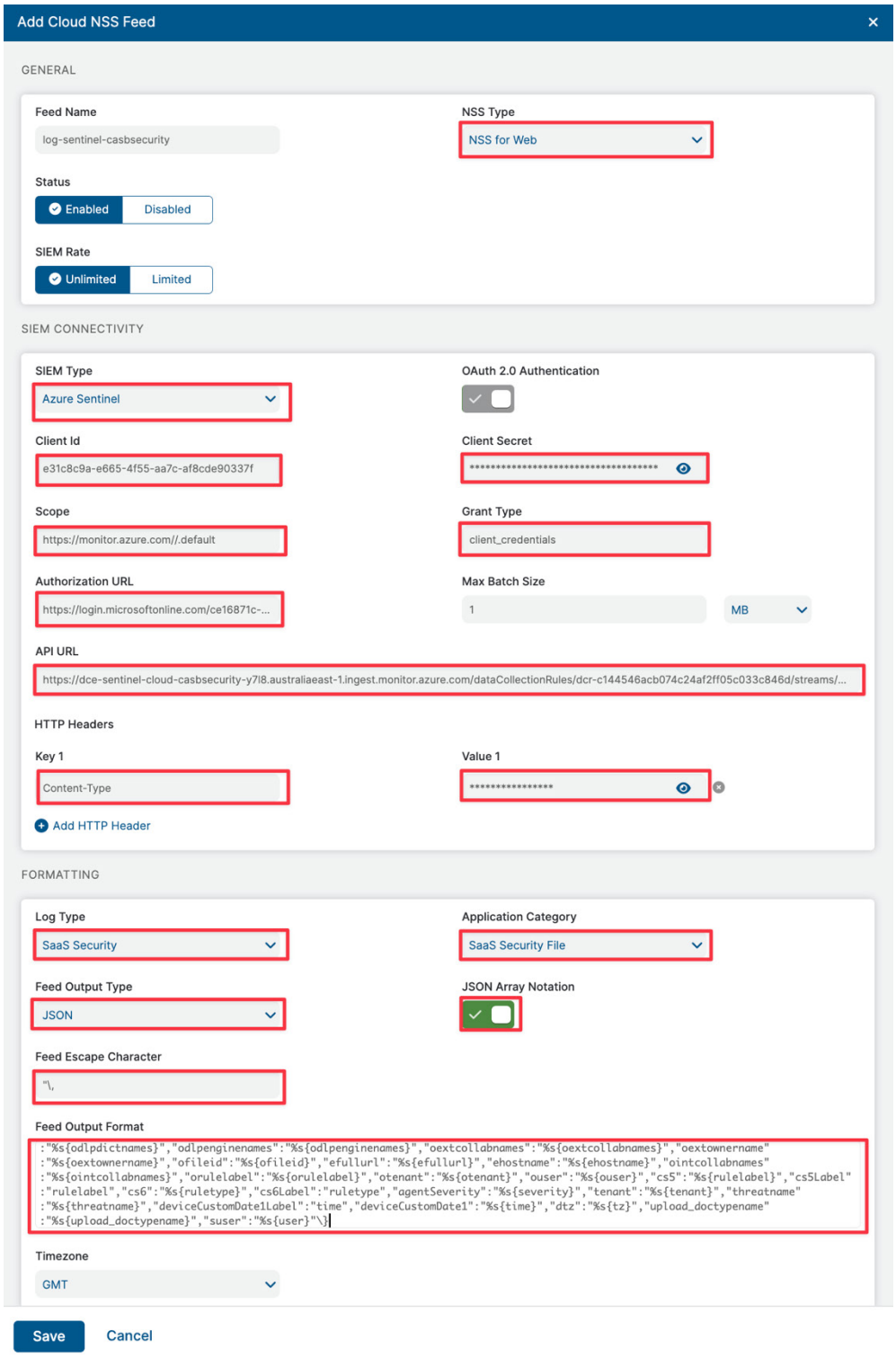


Figure 44. Add Cloud NSS Feed for SaaS Security File logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

SaaS Security Repository

```
\{“TimeGenerated”:”%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}”,”sourcetype”
:”zscalernss-casb”,”vendor”:”Zscaler”,”product”:”NSSCasbrepolog”,”version”:”5.7”,”Sev
erity”:”3”,”name”:”%s{rulelabel}”,”deviceEventClassId”:”%s{applicationname}”,”act”:”
%s{policy}”,”cs1”:”%d{dlpidentifier}”,”cs1Label”:”dlpidentifier”,”start”:”%d{epochtime
}”,”fsize”:”%d{filesize}”,”cn1”:”%d{num_external_collab}”,”cn1Label”:”num_external_co
llab”,”deviceExternalId”:”%d{recordid}”,”company”:”%s{company}”,”datacenter”:”%s{da
tacenter}”,”datacentercity”:”%s{datacentercity}”,”datacentercountry”:”%s{datacente
rcountry}”,”department”:”%s{department}”,”cs2”:”%s{dlpdictcount}”,”cs2Label”:”dlpd
ictcount”,”cs3”:”%s{dlpdictnames}”,”cs3Label”:”dlpdictnames”,”cs4”:”%s{dlpenginena
mes}”,”cs4Label”:”dlpenginenames”,”external_collabnames”:”%s{external_collabnames}
”,”extownername”:”%s{extownername}”,”fileId”:”%s{fileid}”,”fileHash”:”%s{filemd5}”,”fn
ame”:”%s{filename}”,”filePath”:”%s{filepath}”,”fileType”:”%s{filetypecategory}”,”intern
al_collabnames”:”%s{internal_collabnames}”,”flexString1Label”:”malware”,”flexString1”
:”%s{malware}”,”flexString2Label”:”malwareclass”,”flexString2”:”%s{malwareclass}”,”o
dlpdictnames”:”%s{odlpdictnames}”,”odlpenginenames”:”%s{odlpenginenames}”,”oextern
al_collabnames”:”%s{oexternal_collabnames}”,”oextownername”:”%s{oextownername}”,”ofile
id”:”%s{ofileid}”,”ointernal_collabnames”:”%s{ointernal_collabnames}”,”oowner”:”%s{oow
ner}”,”orulelabel”:”%s{orulelabel}”,”otenant”:”%s{otenant}”,”suser”:”%s{owner}”,”proj
ectname”:”%s{projectname}”,”reponame”:”%s{reponame}”,”cs5”:”%s{rulelabel}”,”cs5Label”
:”rulelabel”,”cs6”:”%s{ruletype}”,”cs6Label”:”ruletype”,”agentSeverity”:”%s{severity}
”,”tenant”:
“%s{tenant}”,”threatname”:”%s{threatname}”,”deviceCustomDate1Label”:”time”,
“deviceCustomDate1”:”%s{time}”,”dtz”:”%s{tz}”,”upload_doctypename”:”%s{upload_
doctypename}”\}
```

©2026 Zscaler, Inc. All rights reserved.

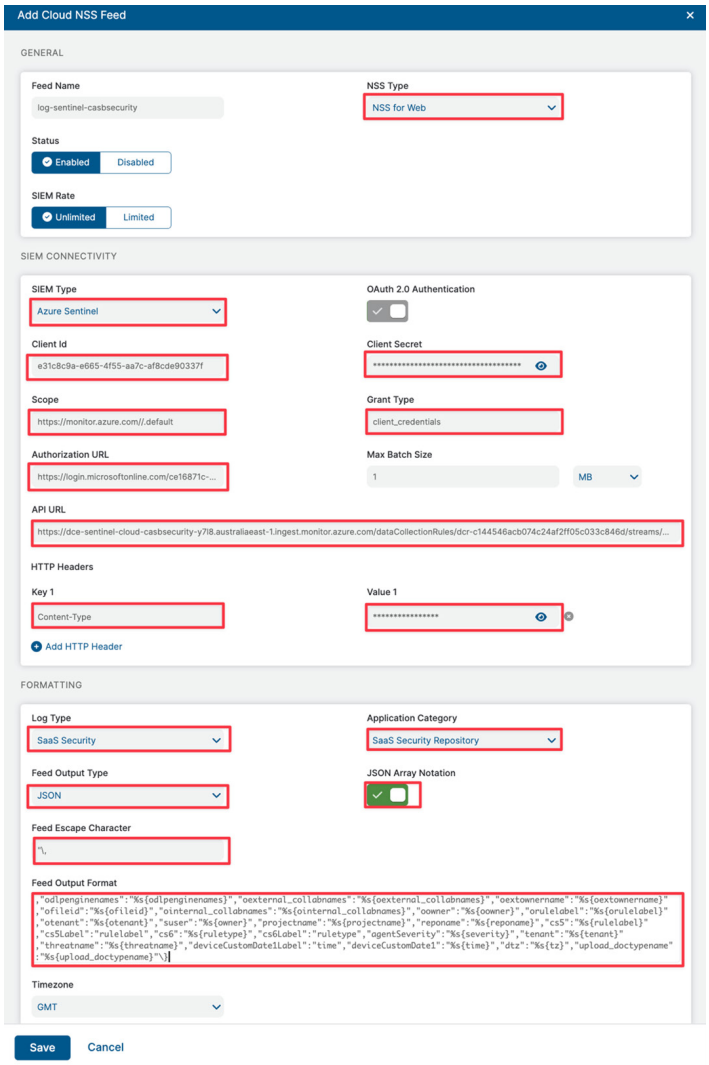


Figure 45. Add Cloud NSS Feed for SaaS Security Repository logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

SaaS Security ITSM

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetyp
e":"zscalernss-casb","vendor":"Zscaler","product":"NSSCasbitsmlog","version":"5.7"
,"Severity":"3","name":"%s{rulelabel}","deviceEventClassId":"%s{applicationname}",
"act":"%s{policy}","cs1":"%d{dlpidentifier}","cs1Label":"dlpidentifier","start":"%d{
epochtime}","fsize":"%d{filesize}","cn1":"%d{num_external_collab}","cn1Label":"num_
external_collab","cn2":"%d{num_internal_collab}","cn2Label":"num_internal_collab
","deviceExternalId":"%d{recordid}","company":"%s{company}","component":"%s{comp
onent}","datacenter":"%s{datacenter}","datacentercity":"%s{datacentercity}","dat
acentercountry":"%s{datacentercountry}","department":"%s{department}","cs2":"%s
{dlpdictcount}","cs2Label":"dlpdictcount","cs3":"%s{dlpdictnames}","cs3Label":"
dlpdictnames","cs4":"%s{dlpenginenames}","cs4Label":"dlpenginenames","external_
collabnames":"%s{external_collabnames}","extownername":"%s{extownername}","fileId":
"%s{file_msg_id}","fileModificationTime":"%s{file_msg_mod_time}","fileHash":"%s{filemd5}
","fname":"%s{filename}","filePath":"%s{filepath}","fileType":"%s{filetypecategory}","r
equest":"%s{fullurl}","dhost":"%s{hostname}","internal_collabnames":"%s{internal_
collabnames}","flexString1Label":"malware","flexString1":"%s{malware}","flexStri
ng2Label":"malwareclass","flexString2":"%s{malwareclass}","objectname":"%s{obj
ectname}","objecttype":"%s{objecttype}","odlpdictnames":"%s{odlpdictnames}","
odlpenginenames":"%s{odlpenginenames}","oexternal_collabnames":"%s{oexternal_
collabnames}","oextownername":"%s{oextownername}","ofile_msg_id":"%s{ofile_
msg_id}","ofullurl":"%s{ofullurl}","ohostname":"%s{ohostname}","ointernal_
collabnames":"%s{ointernal_collabnames}","oowner":"%s{oowner}","orulelabel":"%s{orul
elabel}","otenant":"%s{otenant}","suser":"%s{owner}","cs5":"%s{rulelabel}","cs5Label
":"rulelabel","cs6":"%s{ruletype}","cs6Label":"ruletype","agentSeverity":"%s{severit
y}","tenant":"%s{tenant}","threatname":"%s{threatname}","deviceCustomDate1Label":"t
ime","deviceCustomDate1":"%s{time}","dtz":"%s{tz}","upload_doctypename":"%s{upload_
doctypename}"\}
```

©2026 Zscaler, Inc. All rights reserved.

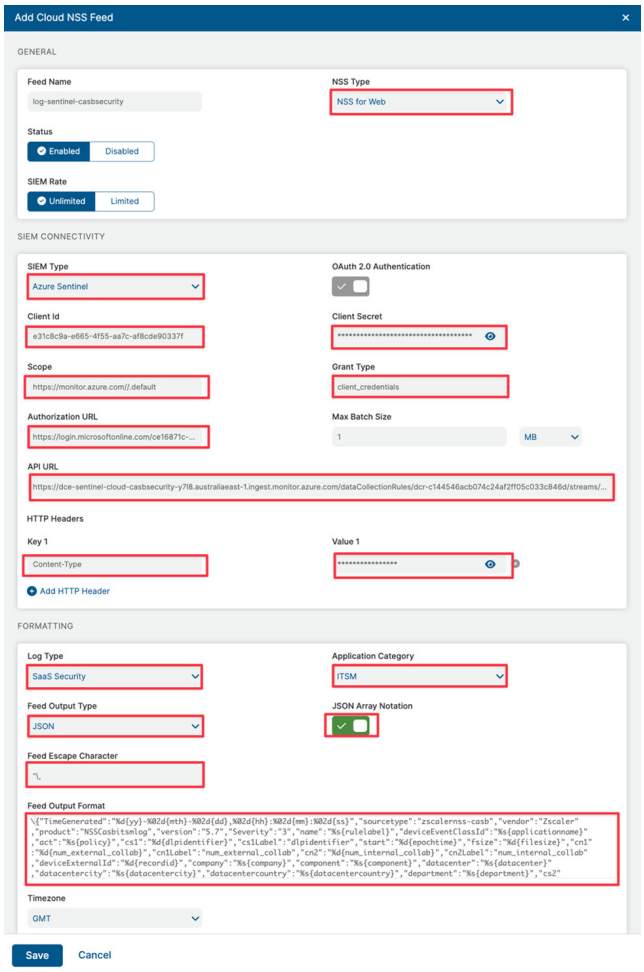


Figure 46. Add Cloud NSS Feed for SaaS Security ITSM logs

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Cloud & Branch Connector Session Logs

For Cloud NSS Feeds for Cloud & Branch Connector Session Logs (government agencies, see Cloud NSS Feeds for Cloud & Branch Connector Session Logs), copy and paste the prepopulated Feed Output Format from the file cloud-nss-cloudbranch-session.fof in the Zscaler GitHub repo or with the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","s
ourcetype":"zscalernss-session","vendor":"Zscaler","product":"NSSSessionlog-
","version":"5.7","act":"%s{nw_service}","end":"%s{eventtime}","rt":"%s{time}","d
tz":"%s{tz}","dhost":"%s{clt_dst_name}","src":"%s{clt_src_ip}","spt":"%d{clt_src_
port}","dst":"%s{clt_dst_ip}","dpt":"%d{clt_dst_port}","deviceCustomString2":"%s{fwdt
ypename}","deviceCustomString2Label":"fwdtypename","deviceCustomString3":"%s{zapptun
name}","deviceCustomString3Label":"zapptunname","deviceCustomNumber1":"%d{srv_dst_po
rt}","deviceCustomNumber1Label":"srv_dst_port","deviceCustomNumber2":"%d{gw_dst_por
t}","deviceCustomNumber2Label":"gw_dst_port","deviceCustomNumber3":"%d{zia_src_port
}","deviceCustomNumber3Label":"zia_src_port","deviceCustomString4":"%s{srv_dst_ip}"
,"deviceCustomString4Label":"srv_dst_ip","deviceCustomString5":"%s{gw_name}","devic
eCustomString5Label":"gw_name","deviceCustomString6":"%s{ogw_name}","deviceCustomS
tring6Label":"ogw_name","gw_dst_ip":"%s{gw_dst_ip}","deviceCustomString1":"%s{zia_
src_ip}","deviceCustomString1Label":"zia_src_ip","srv_src_ip":"%s{srv_src_ip}","srv_
src_port":"%d{srv_src_port}","srv_ip_country":"%s{srv_ip_country}","session_
duration":"%ld{session_duration}","cnt":"%d{num_sess}","num_aggr_sess":"%d{num_aggr_
sess}","aggrname":"%s{aggrname}","rdrrulename":"%s{rdrrulename}","ordrrulename":"
%s{ordrrulename}","self_rulename":"%s{self_rulename}","oself_rulename":"%s{oself_
rulename}","in_bytes":"%ld{clt_tx_bytes}","out_bytes":"%ld{clt_rx_bytes}","nw_
service":"%s{nw_service}","proto":"%s{clt_nw_protocol}","srv_nw_protocol":"%s{srv_
nw_protocol}","zia_gw_protocol":"%s{zia_gw_protocol}","app":"%s{zpa_appseg}","dv
chost":"%s{name}","vm_name":"%s{vm_name}","group_name":"%s{group_name}","region_
name":"%s{region_name}","location":"%s{location}","platform_name":"%s{platform_
name}","platformgeo_name":"%s{platformgeo_name}","account_id":"%s{account_id}","Sever
ity":"3","deviceEventClassId":"%s{nw_service}","name":"%s{traffic_type}"\}
```

©2026 Zscaler, Inc. All rights reserved.

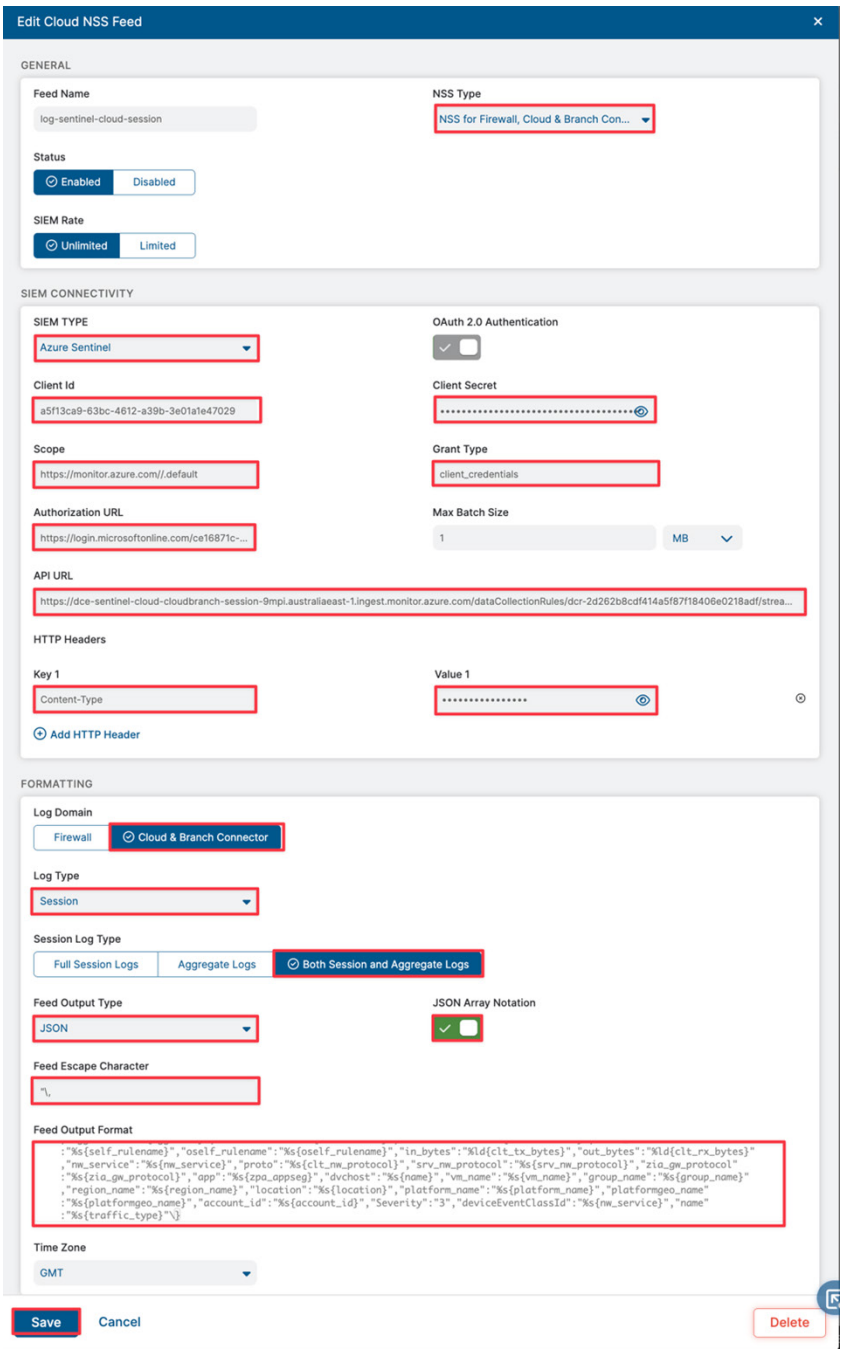


Figure 47. Add Cloud NSS Feed for Cloud & Branch Connector Session Logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Cloud & Branch Connector Event Logs

For Cloud NSS Feeds for Cloud & Branch Connector Event Logs (government agencies, see Cloud NSS Feeds for Cloud & Branch Connector Event Logs), copy and paste the prepopulated Feed Output Format from the file cloud-nss-cloudbranch-event.fof in the Zscaler GitHub repo or with the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","sourcetype
":"zscalernss-event","vendor":"Zscaler","product":"NSSEventlog","version":"5.7","act
":"%s{category}","cat":"%s{severity}","deviceEventClassId":"%s{event_name}","deviceC
ustomDate2":"%d{inteventtime}","deviceCustomDate2Label":"inteventtime","deviceCustom
Date1":"%d{abstime}","deviceCustomDate1Label":"abstime","flexDate1":"%s{time}","flexDa
te1Label":"time","dtz":"%s{tz}","rt":"%s{rtime}","company":"%s{company}","location":
"%s{location}","dvchost":"%s{name}","vm_name":"%s{vm_name}","group_name":"%s{group_
name}","platform_name":"%s{platform_name}","platformgeo_name":"%s{platformgeo_
name}","region_name":"%s{region_name}","connector_type":"%s{connector_type}","device
CustomString2":"%s{attr1_val}","deviceCustomString2Label":"%s{attr1_name}","deviceCu
stomString3":"%s{attr2_val}","deviceCustomString3Label":"%s{attr2_name}","deviceCust
omString4":"%s{attr3_val}","deviceCustomString4Label":"%s{attr3_name}","deviceCustom
String5":"%s{attr4_val}","deviceCustomString5Label":"%s{attr4_name}","deviceCustomSt
ring6":"%s{error_code}","account_id":"%s{account_id}","name":"%s{category}","Severi
ty":"3"\}
```

©2026 Zscaler, Inc. All rights reserved.

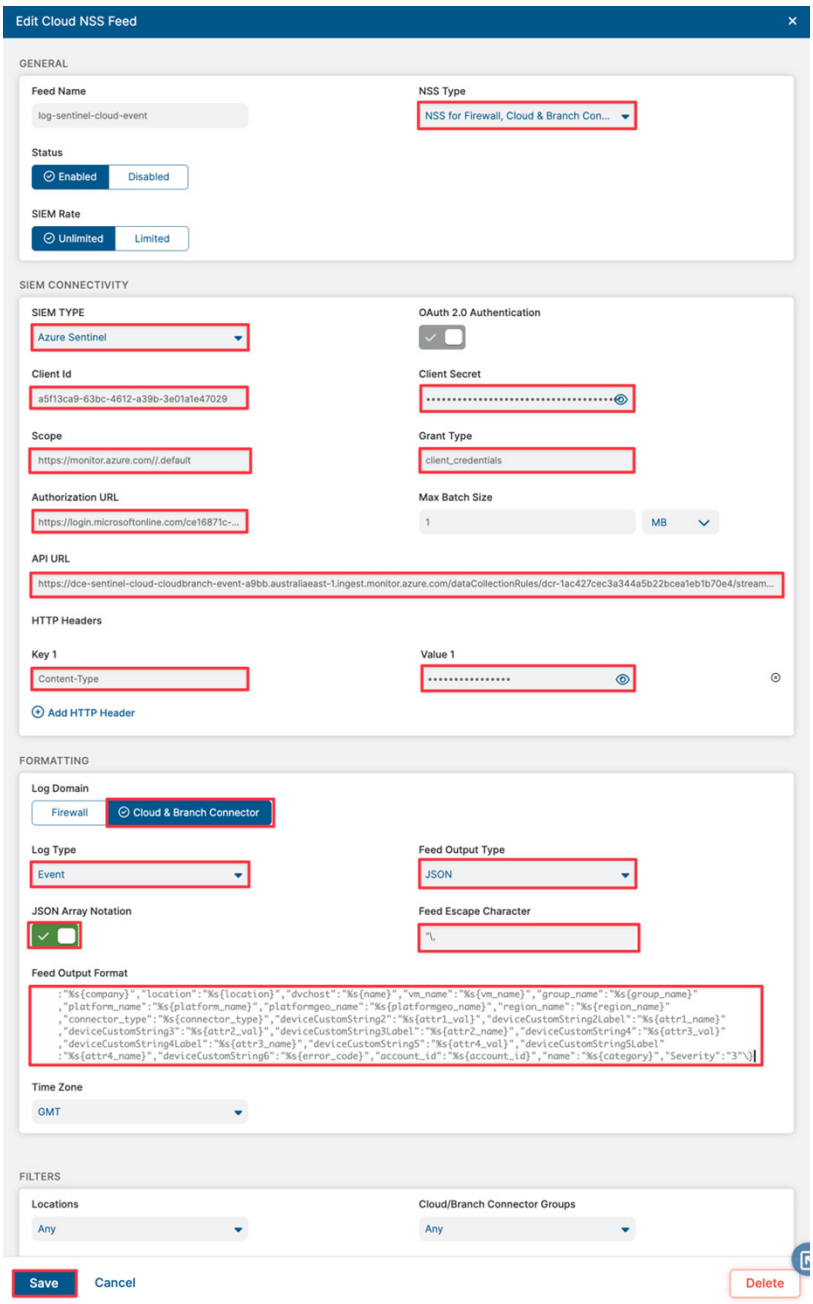


Figure 48. Add Cloud NSS Feed for Cloud & Branch Connector Event Logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Cloud NSS Feeds for Cloud & Branch Connector DNS Logs

For Cloud NSS Feeds for Cloud & Branch Connector DNS Logs (government agencies, see Cloud NSS Feeds for Cloud & Branch Connector DNS Logs) copy and paste the prepopulated Feed Output Format from the file cloud-nss-cloudbranch-dns.fof in the Zscaler GitHub repo or with the following:

```
\{"TimeGenerated":"%d{yy}-%02d{mth}-%02d{dd},%02d{hh}:%02d{mm}:%02d{ss}","
sourcetype":"zscalernss-dns","vendor":"Zscaler","product":"NSSDnslog","ver
sion":"5.7","act":"%s{req_action}","end":"%s{eventtime}","dtz":"%s{tz}","in_
bytes":"%d{inbytes}","out_bytes":"%d{outbytes}","src":"%s{clt_ip}","cn1Label":"req_
duration","cn1":"%d{req_duration}","cs1Label":"res_rulename","cs1":"%s{res_rul
ename}","recordid":"%d{recordid}","cs2Label":"location","cs2":"%s{location}","
destinationDnsDomain":"%s{dom_str}","odom_str":"%s{odom_str}","cs3Label":"res_
type","cs3":"%s{res_type}","cs4Label":"resp_val_str","cs4":"%s{resp_val_
str}","oresp_val_str":"%s{oresp_val_str}","cs5Label":"dns_err","cs5":"%s{dns_
err}","dpt":"%d{srv_port}","cnt":"%d{num_resp}","proto":"%s{protocol}","FlexStr
ing1Label":"dnsgw_flags","FlexString1":"%s{dnsgw_flags}","FlexString2Label":"dns
gw_slot","FlexString2":"%s{dnsgw_slot}","dvchost":"%s{name}","vm_name":"%s{vm_
name}","group_name":"%s{group_name}","region_name":"%s{region_name}","platform_
name":"%s{platform_name}","platformgeo_name":"%s{platformgeo_name}","account_
id":"%s{account_id}","dst":"%s{resolver}","res_action":"%s{res_action}","deviceEventC
lassId":"%s{req_rulename}","Severity":"3","name":"%s{req_type}"\}
```

©2026 Zscaler, Inc. All rights reserved.

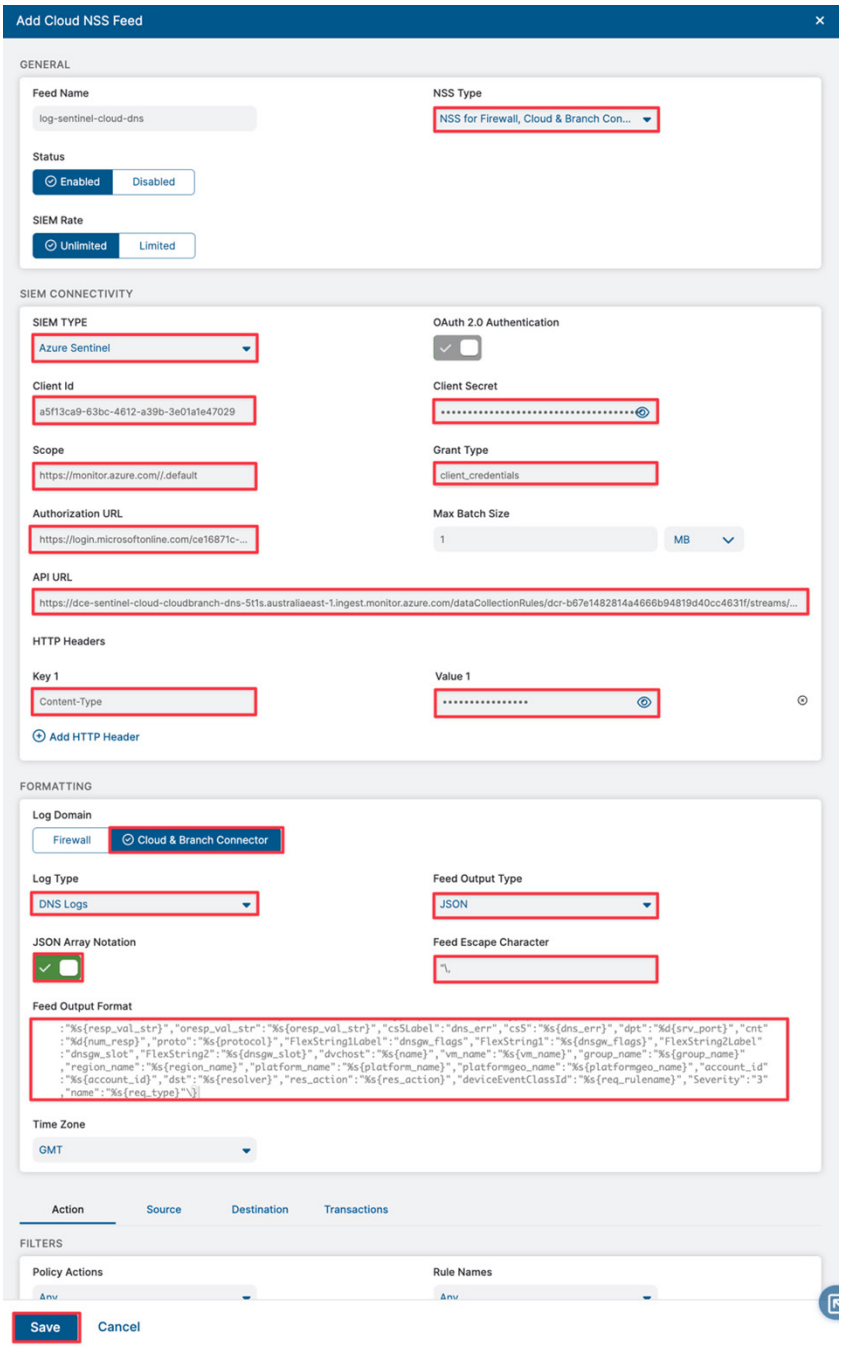


Figure 49. Add Cloud NSS Feed for Cloud & Branch Connector DNS Logs

Fields that do not map to one of the standard CommonSecurityLog fields are added to the AdditionalExtensions column as semicolon-separated key-value pairs.

©2026 Zscaler, Inc. All rights reserved.

Using PowerShell, Verify Log Data Arrives at the Endpoint

(Optional) You can manually push log data to the endpoint to verify your deployment and confirm that the Sentinel credentials are correct. Refer to the PowerShell script in the Microsoft documentation. The following is an example using web logs. Replace the values shown in red with those from the resources you created in the Azure portal.

```
##################

### Step 0: set parameters required for the rest of the script

##################

#information needed to authenticate to AAD and obtain a bearer token

$tenantId = "XX-XX-XX-8330-d05f70XX"; #Tenant ID the data collection endpoint resides
in

$appId = "XXX-XX-4da9-bb7d-ba39bXXX"; #Application ID created and granted permissions

$appSecret = "XXXc6XJmEJ5vJBnl2B8v3Vu-a5H8XXX"; #Secret created for the application

#information needed to send data to the DCR endpoint

$dcrImmutableId = "dcr-XX20dfb3018XX"; #the immutableId property of the DCR object

$dceEndpoint = "https://testnss3-dce-XX-1.ingest.monitor.azure.com"; #the endpoint
property of the Data Collection Endpoint object

##################

### Step 1: obtain a bearer token used later to authenticate against the DCE

##################

$scope= [System.Web.HttpUtility]::UrlEncode("https://monitor.azure.com//.default")

$body = "client_id=$appId&scope=$scope&client_secret=$appSecret&grant_type=client_
credentials";

$headers = @{"Content-Type"="application/x-www-form-urlencoded"};

$uri = "https://login.microsoftonline.com/$tenantId/oauth2/v2.0/token"

$bearerToken = (Invoke-RestMethod -Uri $uri -Method "Post" -Body $body -Headers
$headers).access_token

##################

### Step 2: Load up some sample data.

##################

$currentTime = Get-Date ([datetime]::UtcNow) -Format O

$staticData = @"

[{ "sourcetype" : "zscalernss-web", "TimeGenerated":"2023-03-17 15:01:21",
"act":"Blocked", "reason":"Blocked", "app":"HTTPS", "dhost":"www.etsy.com",
"dst":"104.94.233.143", "src":"40.83.138.250", "sourceTranslatedAddress":"10.2.3.4",
"in":"50", "out":"10", "request":"www.1etsy.com/dac/common/web-toolkit/
```

©2026 Zscaler, Inc. All rights reserved.

scoped/scoped_responsive_base.20220526203537%2csite-chrome/deprecated/global-nav.20220526203537%2ccommon/web-toolkit/a11y_colors/overrides.20220526203537.css", "requestContext":"www.1etsy.com/c/clothing-and-shoes?ref=catnav-10923", "outcome":"200", "requestClientApplication":"Mozilla/5.0 (Windows NT 6.2; Win64; x64; rv:16.0.1) Gecko/20121011 Firefox/21.0.1", "requestMethod":"GET", "suser":"testuser2@bd-dev.com", "spriv":"Road Warrior", "externalId":"8106135709380313090", "fileType":"GZIP ", "destinationServiceName":"Etsy", "cat":"Professional Services", "deviceDirection":"1", "cn1":"10", "cn1Label":"riskscore", "cs1":"General Group", "cs1Label":"dept", "cs2":"Phishing", "cs2Label":"urlcat", "cs3":"None", "cs3Label":"malwareclass", "cs4":"None", "cs4Label":"malwarecat", "cs5":"Bad_Threat", "cs5Label":"threatname", "cs6":"None", "cs6Label":"md5hash", "rulelabel":"None", "ruletype":"None", "urlclass":"Advanced Security Risk", "DeviceVendor":"Zscaler" , "DeviceProduct":"NSSWeblog" ,"devicemodel":"Virtual Machine" , "flexString1":"Virtual Machine", "flexString1Label":"devicemodel", "flexString2":"Advanced Security Risk", "flexString2Label":"urlclass"},

{ "sourcetype" : "zscalernss-web", "TimeGenerated":"2023-03-17 01:45:22", "act":"Allowed", "reason":"Allowed", "app":"HTTP_PROXY", "dhost":"c.bing.com", "dst":"204.79.197.200", "src":"40.90.198.229", "sourceTranslatedAddress":"40.90.198.229", "in":"6500", "out":"110", "request":"c.bing.com:443", "requestContext":"None", "outcome":"200", "requestClientApplication":"Windows Microsoft Windows 10 Pro ZTunnel/1.0", "requestMethod":"CONNECT", "suser":"testuser2@bd-dev.com", "spriv":"Road Warrior", "externalId":"7093275726860451849", "fileType":"None", "destinationServiceName":"SharePoint", "cat":"Web Search", "deviceDirection":"1", "cn1":"0", "cn1Label":"riskscore", "cs1":"Service Admin", "cs1Label":"dept", "cs2":"Web Search", "cs2Label":"urlcat", "cs3":"None", "cs3Label":"malwareclass", "cs4":"None", "cs4Label":"malwarecat", "cs5":"None", "cs5Label":"threatname", "cs6":"None", "cs6Label":"md5hash", "rulelabel":"None", "ruletype":"None", "urlclass":"Business Use", "DeviceVendor":"Zscaler" , "DeviceProduct":"NSSWeblog" , "devicemodel":"Lenovo" , "flexString1":"Lenovo", "flexString1Label":"devicemodel", "flexString2":"Advanced Security Risk", "flexString2Label":"urlclass" },

{ "sourcetype" : "zscalernss-web", "TimeGenerated":"2023-03-17 02:38:23", "act":"Blocked", "reason":"Access denied due to bad server certificate", "app":"HTTP_PROXY", "dhost":"hm.baidu.com", "dst":"103.235.46.191", "src":"52.233.90.167", "sourceTranslatedAddress":"52.233.90.167", "in":"65", "out":"55", "request":"ps.eyeota.net/pixel?pid=gdomg51&t=gif&cat=Economy&us_privacy=&random=1654532044229.2", "requestContext":"None", "outcome":"200", "requestClientApplication":"Windows Microsoft Windows 10 Pro ZTunnel/1.0", "requestMethod":"CONNECT", "suser":"testuser2@bd-dev.com", "spriv":"Road Warrior", "externalId":"9346135709564534789", "fileType":"None ", "destinationServiceName":"General Browsing", "cat":"Web Search", "deviceDirection":"1", "cn1":"0", "cn1Label":"riskscore", "cs1":"General Group", "cs1Label":"dept", "cs2":"Adware/Spyware Sites", "cs2Label":"urlcat", "cs3":"None", "cs3Label":"malwareclass", "cs4":"None", "cs4Label":"malwarecat", "cs5":"None", "cs5Label":"threatname", "cs6":"None", "cs6Label":"md5hash", "rulelabel":"Inspect_All", "ruletype":"SSLPol", "urlclass":"Business Use", "DeviceVendor":"Zscaler" , "DeviceProduct":"NSSWeblog" ,"devicemodel":"macbookpro", "flexString1":"macbookpro", "flexString1Label":"devicemodel", "flexString2":"Advanced Security Risk", "flexString2Label":"urlclass" }]

"@;

©2026 Zscaler, Inc. All rights reserved.

```
##################

### Step 3: send the data to Log Analytics via the DCE.

##################

$body = $staticData;

$headers = @{"Authorization"="Bearer $bearerToken";"Content-Type"="application/
json"};

$uri = "$dceEndpoint/dataCollectionRules/$dcrImmutableId/streams/Custom-XXnss3_table_
CL?api-version=2021-11-01-preview"

$uploadResponse = Invoke-RestMethod -Uri $uri -Method "Post" -Body $body -Headers
$headers

#Invoke-WebRequest -UseBasicParsing -Uri $uri -Method "Post" -Body $body -Headers
$headers
```

©2026 Zscaler, Inc. All rights reserved.

In Microsoft Sentinel, View Log Details

After running your script in PowerShell, you can view log details in Microsoft Sentinel:

1. In Microsoft Sentinel, select your Log Analytics workspace (e.g., `log-sentinel-test1`).
2. In the left-side navigation, select **Logs**. Wait until the **Queries History** populates with logs or enter `CommonSecurityLog` in the query window.
3. Click **Run**.

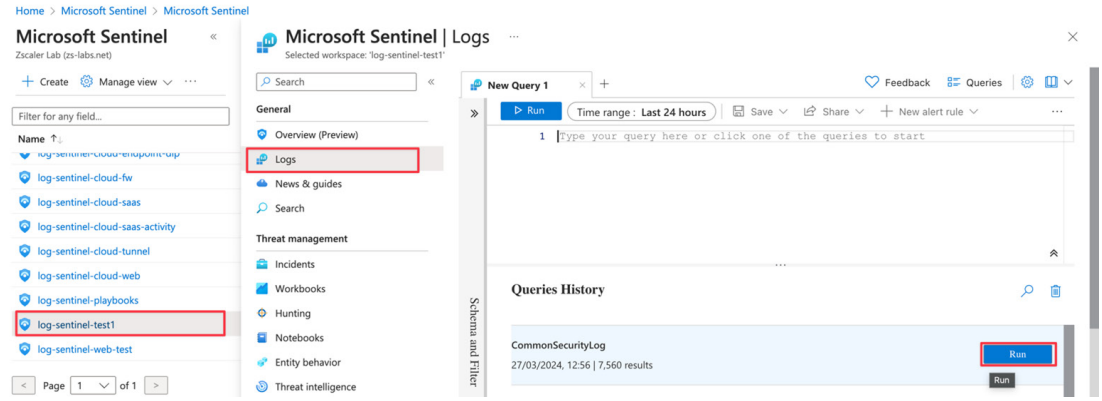


Figure 50. Queries history

Query **Results** display after a few minutes.

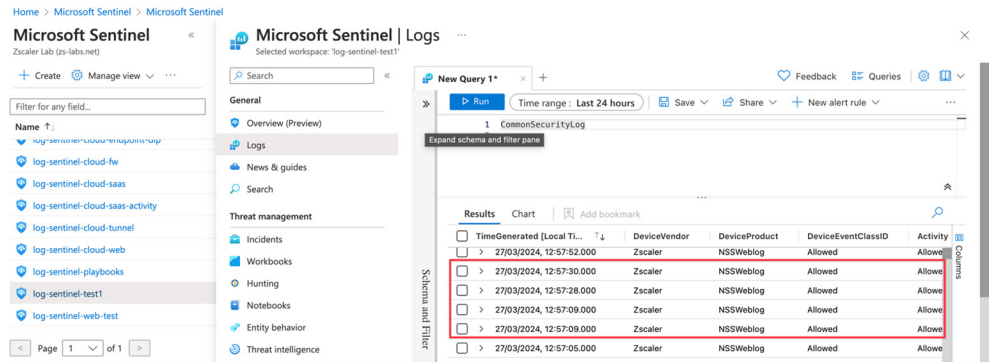


Figure 51. Log query results

©2026 Zscaler, Inc. All rights reserved.

Add the ZIA Content Pack to Install ZIA Workbooks, Analytics Rules, and Parsers

To add the ZIA content pack:

1. In Microsoft Sentinel, select your Log Analytics workspace (e.g., `log-sentinel-test1`).
2. In the left-side navigation, click **Content hub**, and search for `ZIA` or `Zscaler`.

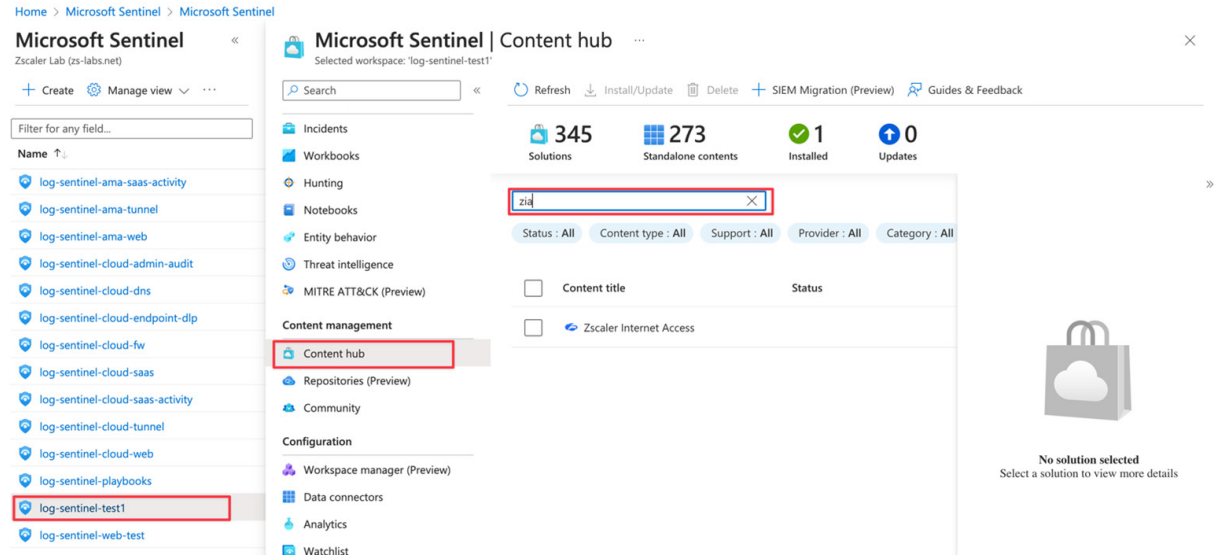


Figure 52. Content hub

3. Select the **Zscaler Internet Access** content pack, and click **Install**.

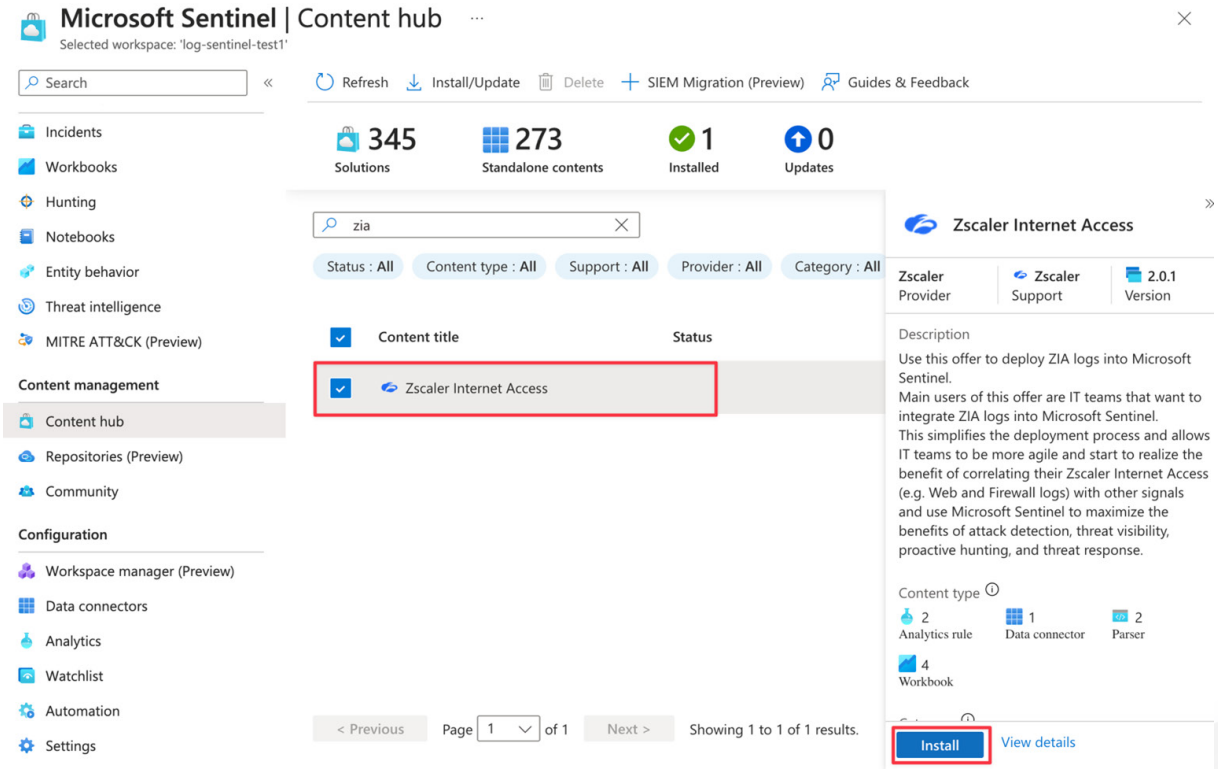


Figure 53. Install

©2026 Zscaler, Inc. All rights reserved.

When installed, the content pack shows a **Status** of **Installed**.

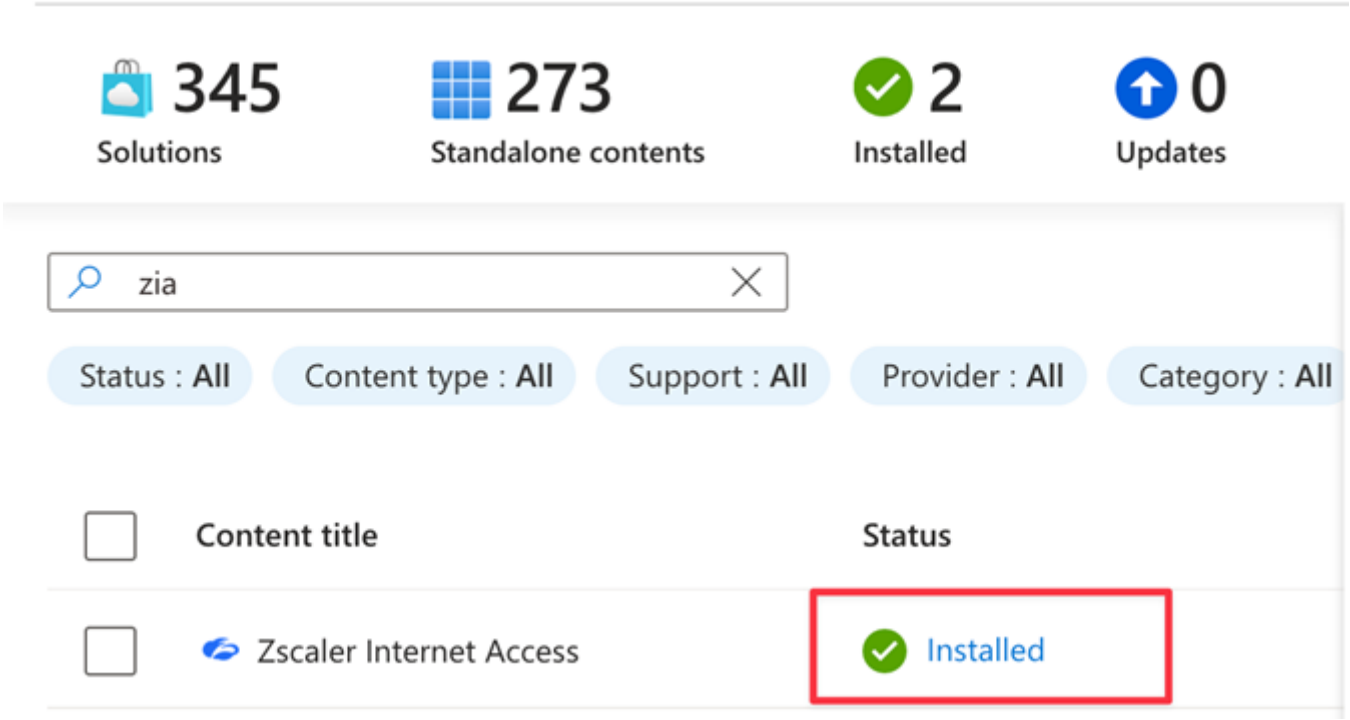


Figure 54. Status installed

4. When installed, click **Workbooks** and then **Templates**.

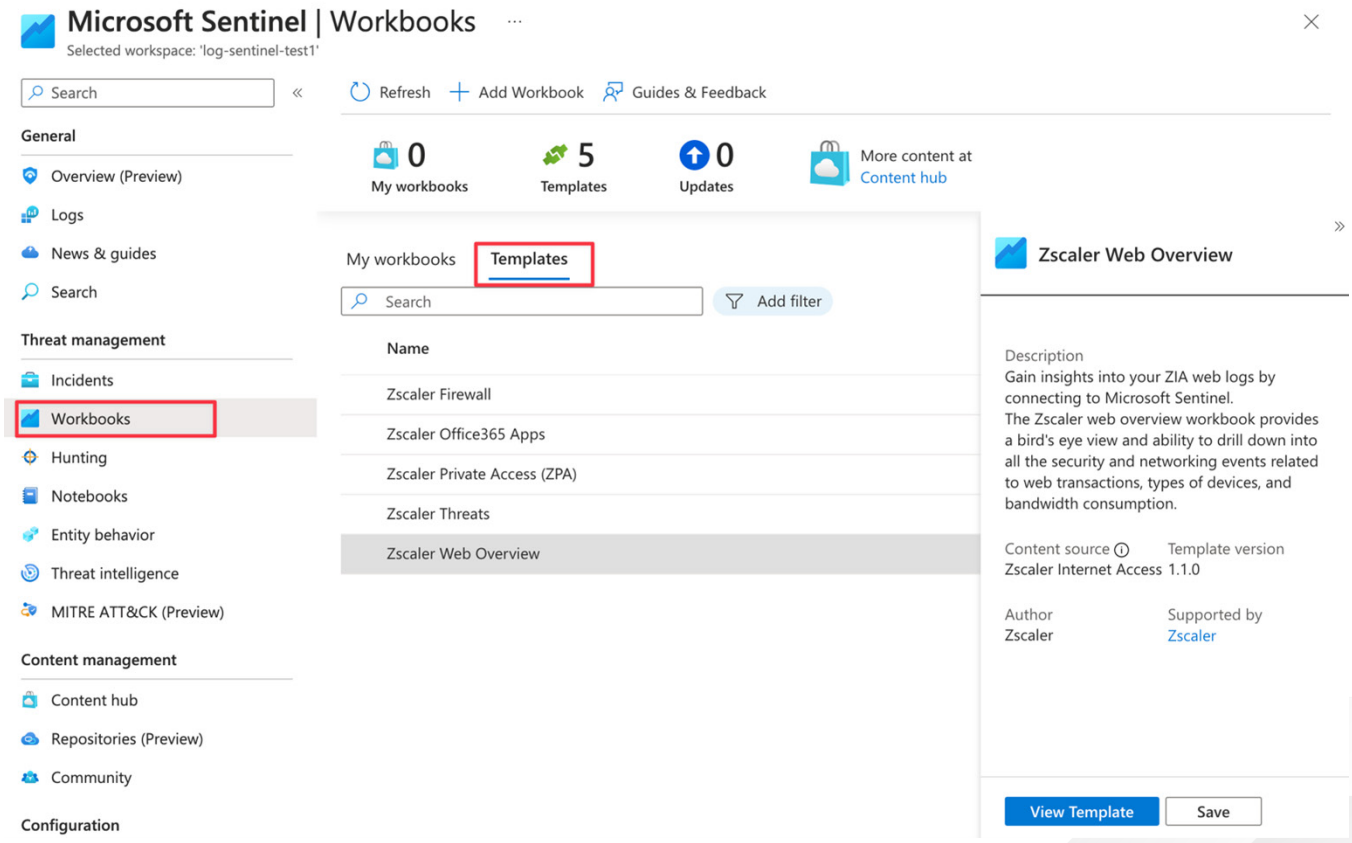


Figure 55. Workbooks

©2026 Zscaler, Inc. All rights reserved.

5. For each workbook you want to install, click the name of the workbook, then **Save**.

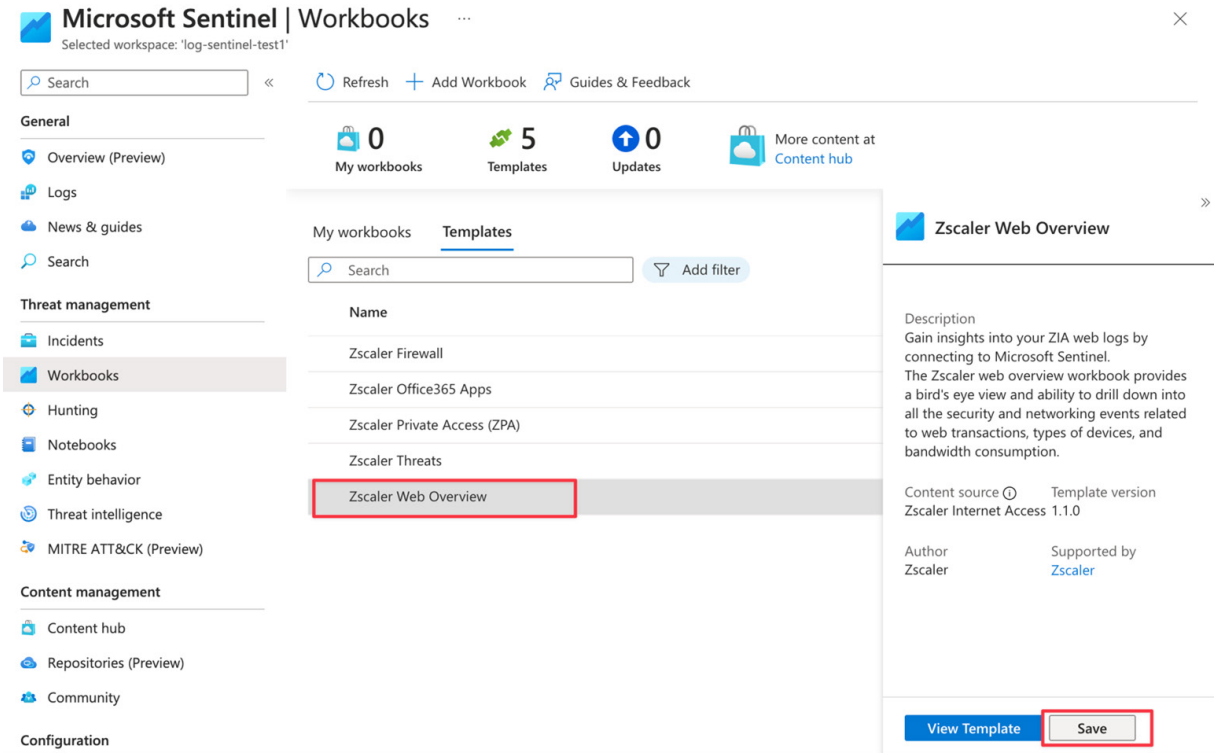


Figure 56. Save workbook

6. Select the region in which to save your workbook and click **Yes**. The saved workbooks appear under **My workbooks**.

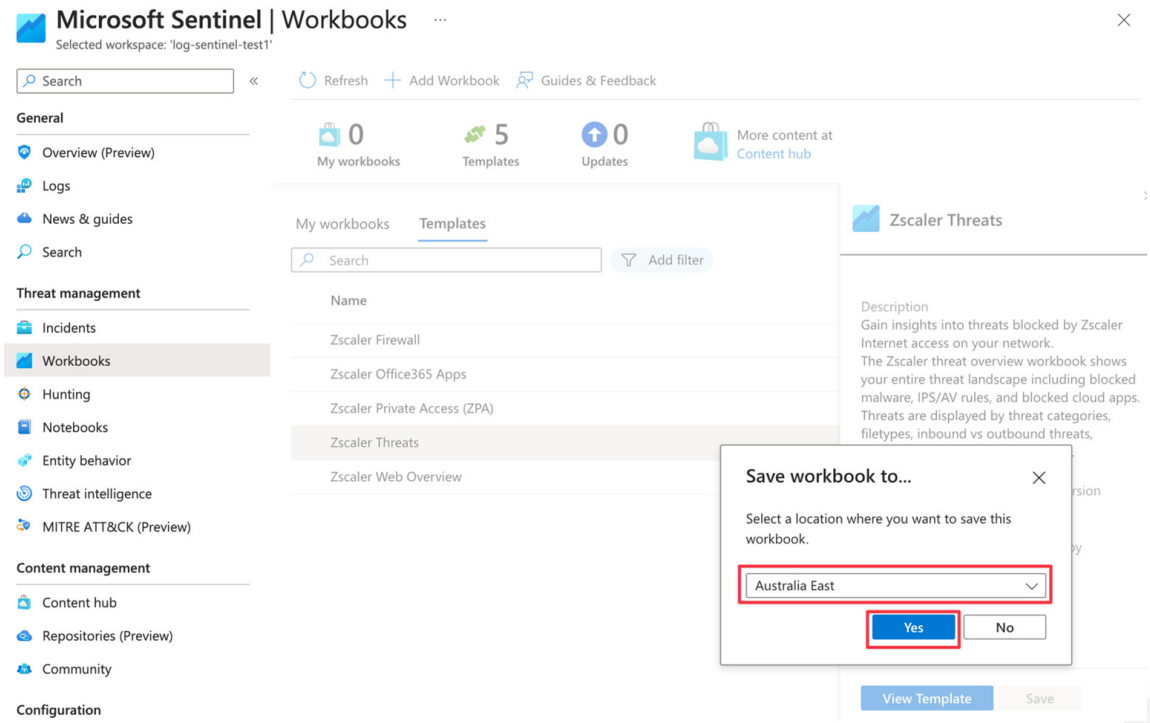


Figure 57. Workbook region

©2026 Zscaler, Inc. All rights reserved.

7. Click **View saved workbook** to view the workbook.

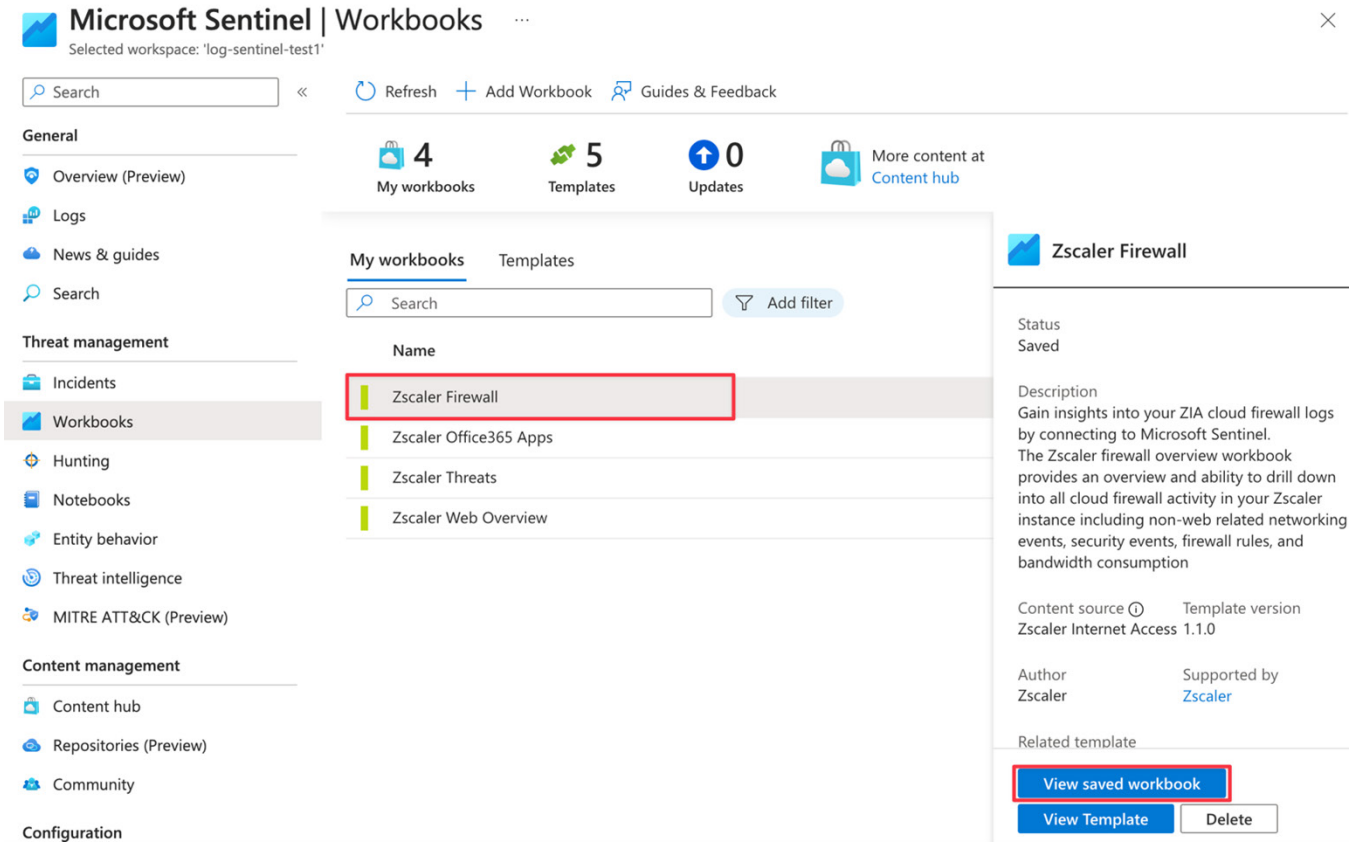


Figure 58. View saved workbooks

Examples of the Workbooks

The following show examples of the workbooks.

The following image is an example Zscaler Web Overview workbook.

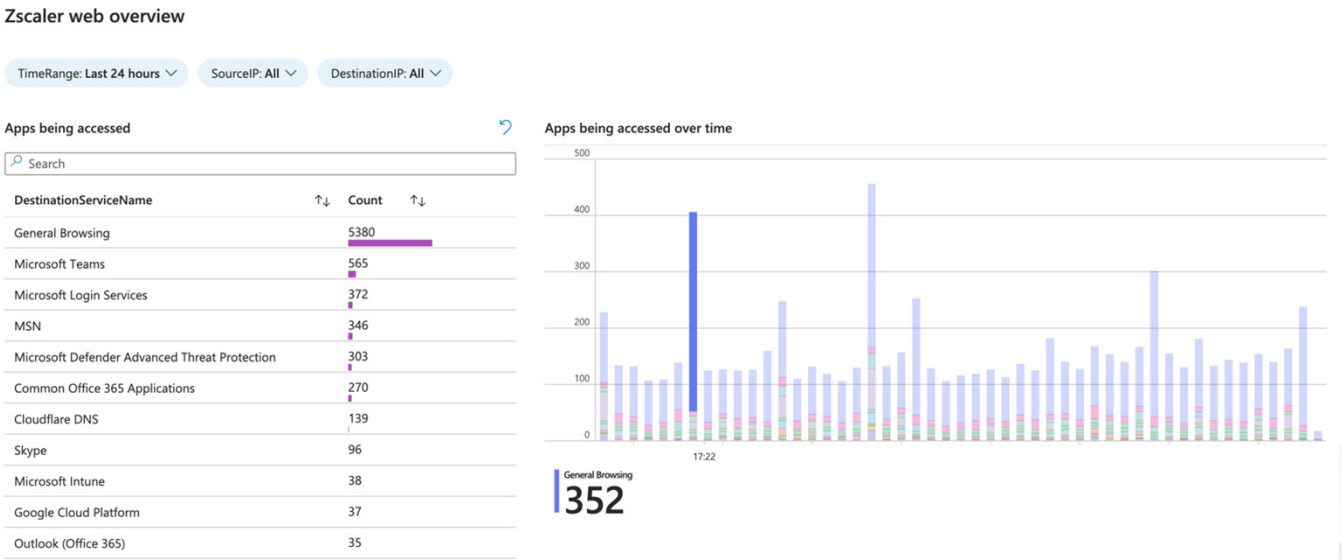


Figure 59. Zscaler Web Overview

©2026 Zscaler, Inc. All rights reserved.

The following image is an example Zscaler Threats Overview workbook.

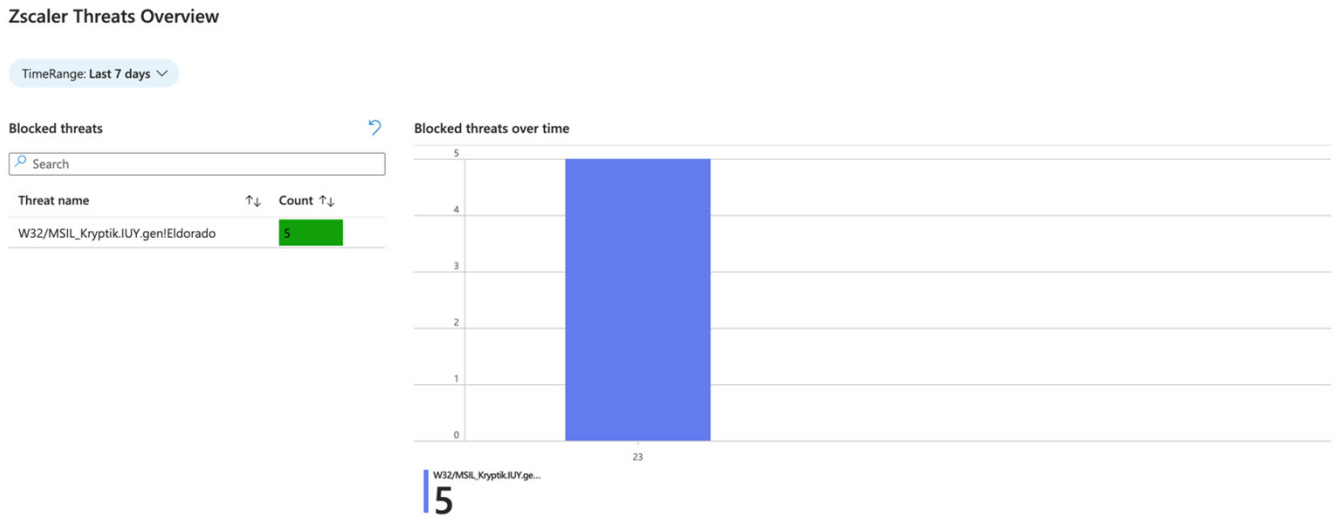


Figure 60. Zscaler Threats Overview

The following image is an example Zscaler Microsoft 365 Overview workbook.

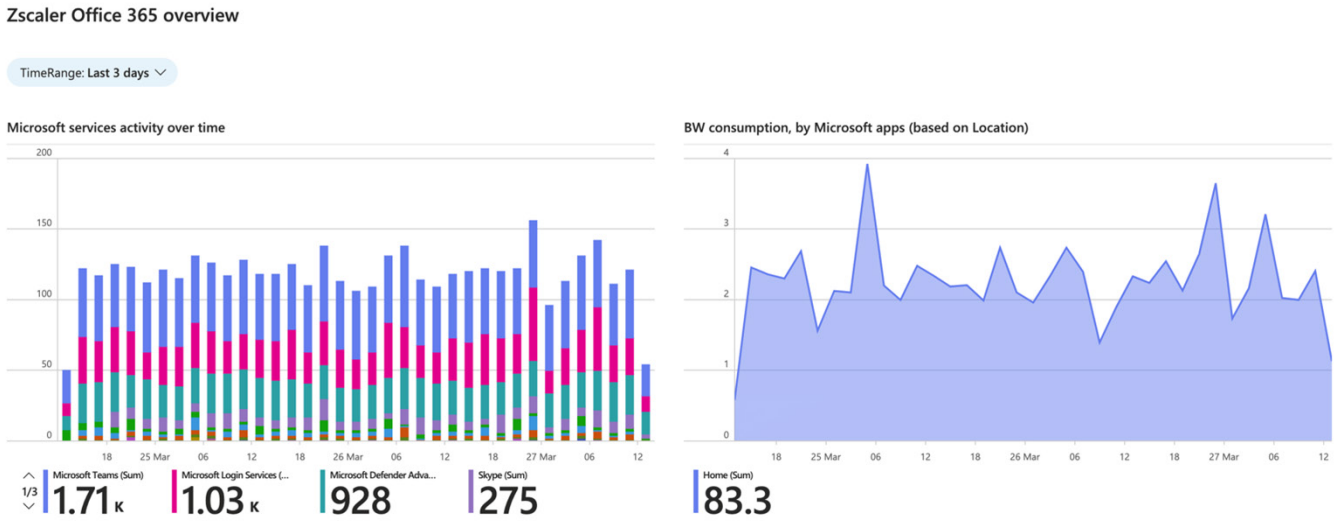


Figure 61. Zscaler Microsoft 365 overview

©2026 Zscaler, Inc. All rights reserved.

The following image is an example Zscaler Firewall Overview workbook.

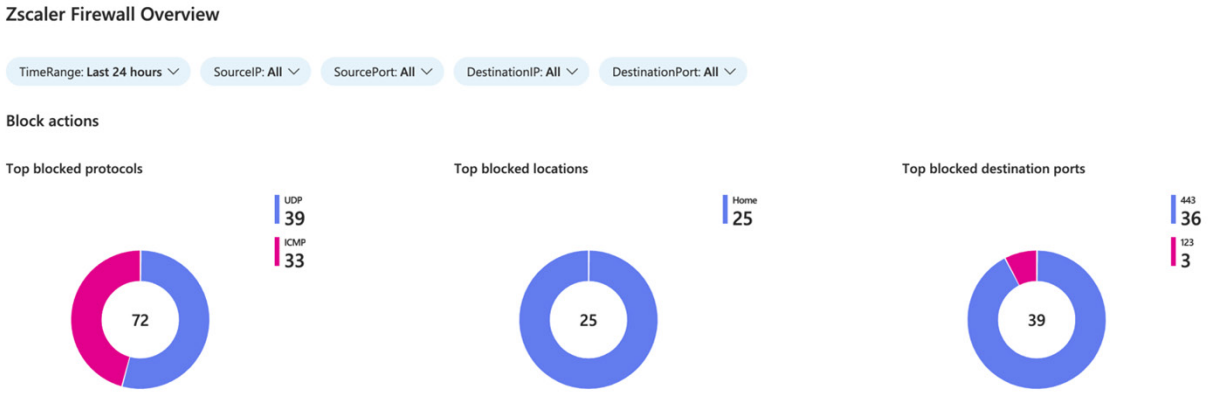


Figure 62. Zscaler Firewall Overview

©2026 Zscaler, Inc. All rights reserved.

Configuring NSS VM-Based Log Ingestion for ZIA

The following steps assume that you have Admin access to the Azure portal.

Logging In to Azure Portal

Normally you would go to the Azure portal and sign in using your account.

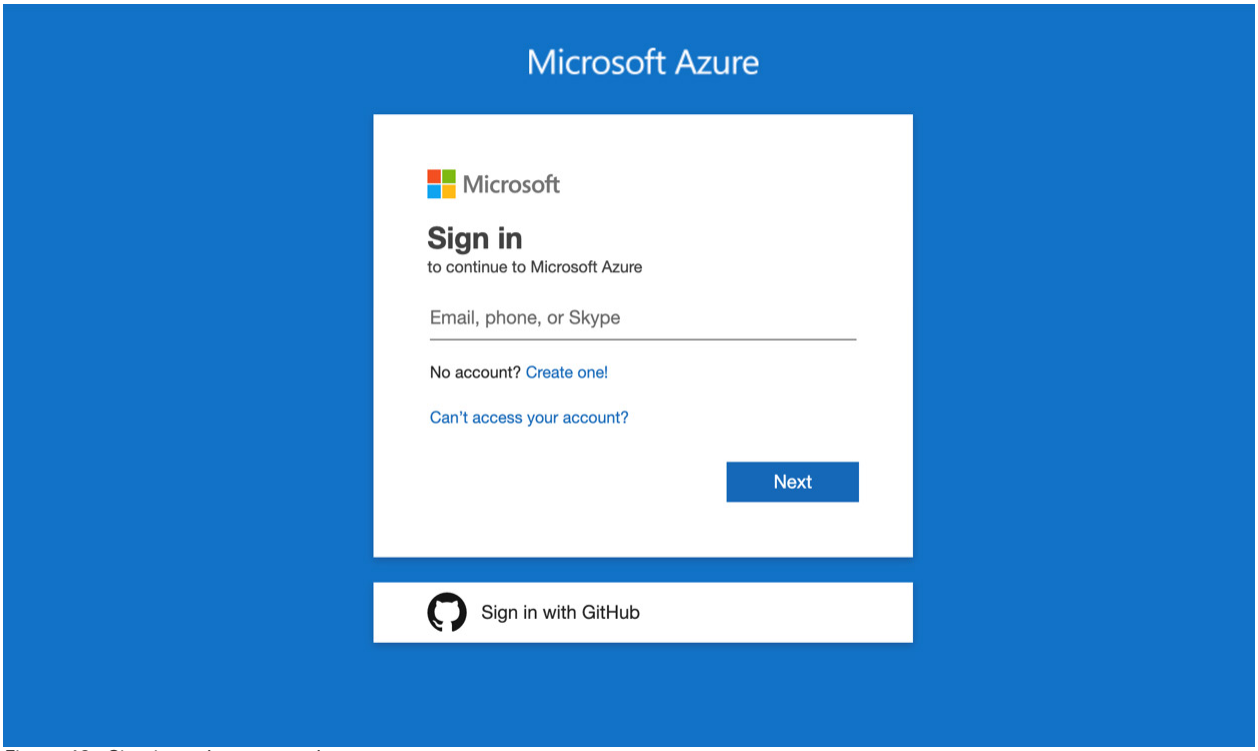


Figure 63. Sign in to Azure portal

©2026 Zscaler, Inc. All rights reserved.

Deploying the Data Connector Host VM

The first step in Azure is to deploy a Linux VM. This Linux VM is the Zscaler data connector and it runs Microsoft's Azure Monitor Agent. The Azure Monitor Agent is the software component that sends log messages to Microsoft Sentinel. There are more software components on this VM that enable this data pipeline. These components are automatically configured by Azure.

Ensure the machine meets Microsoft requirements are per the Microsoft guidance.

1. Log in to the Azure portal. After going to the **Home** window, click **Virtual machines**.

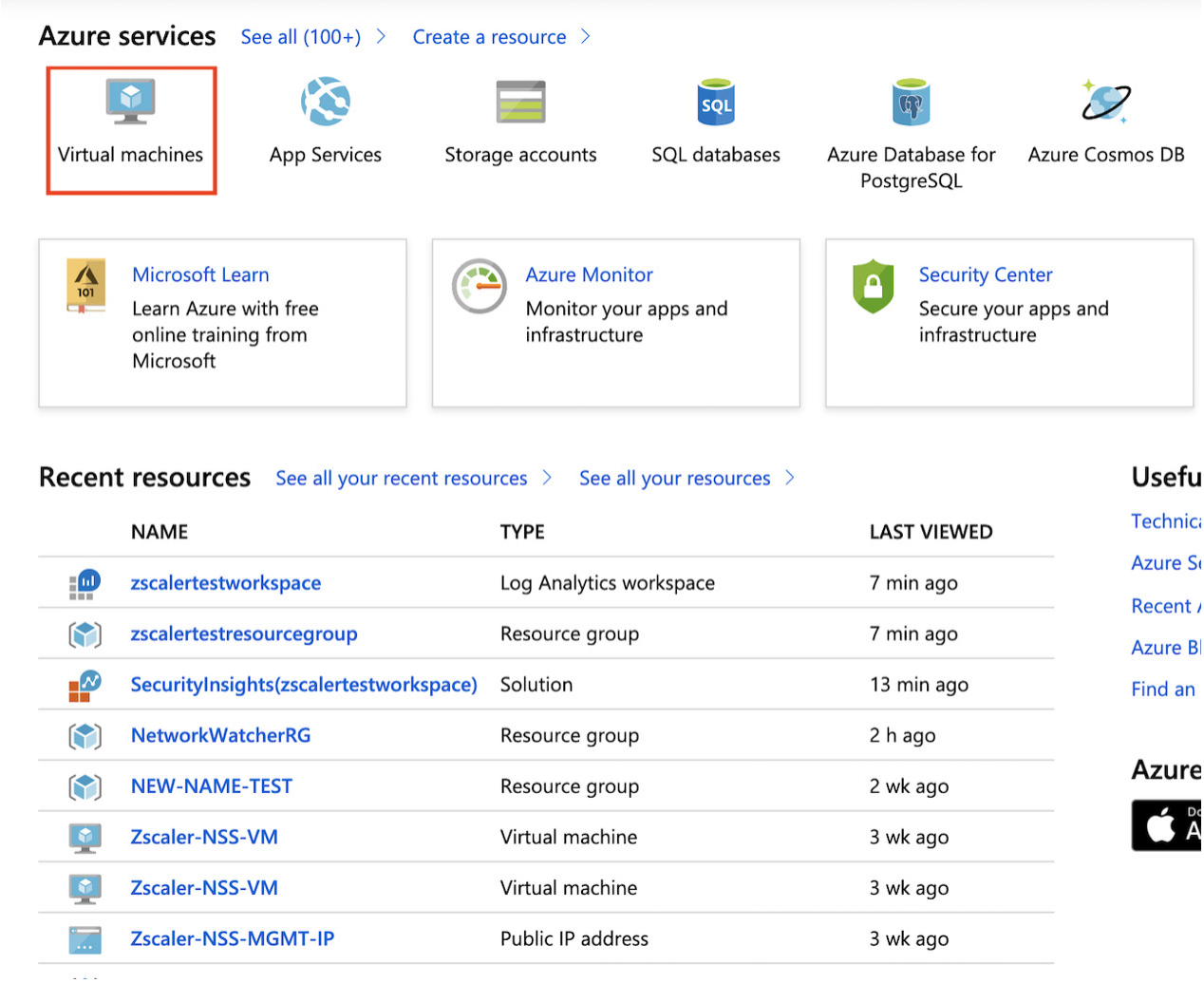


Figure 64. VM Syslog server that runs Azure's data connector

©2026 Zscaler, Inc. All rights reserved.

2. Click **Create virtual machine**. Your **Virtual machines** window might have additional information displayed if you have existing virtual machines.

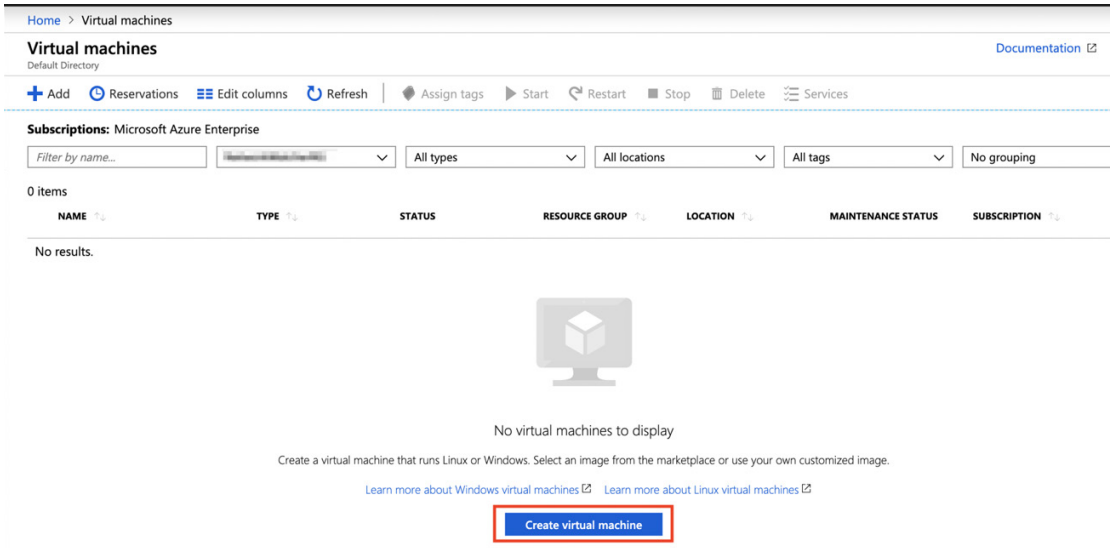


Figure 65. Create virtual machine in Microsoft Azure

©2026 Zscaler, Inc. All rights reserved.

Bind this VM to a Resource Group

A Resource Group (RG) is a way to group a collection of assets in logical containers for automatic provisioning, monitoring, access control, and more effective management of their costs. One benefit of using RGs is to group related application resources together, as they share a unified lifecycle from creation, usage, and deprovisioning.

You can select an existing resource group or you can create a new one. To create a new resource group:

1. Click **Create new**, and enter a **Virtual machine name**. This example uses `Ubuntu Server 18.04 LTS`.

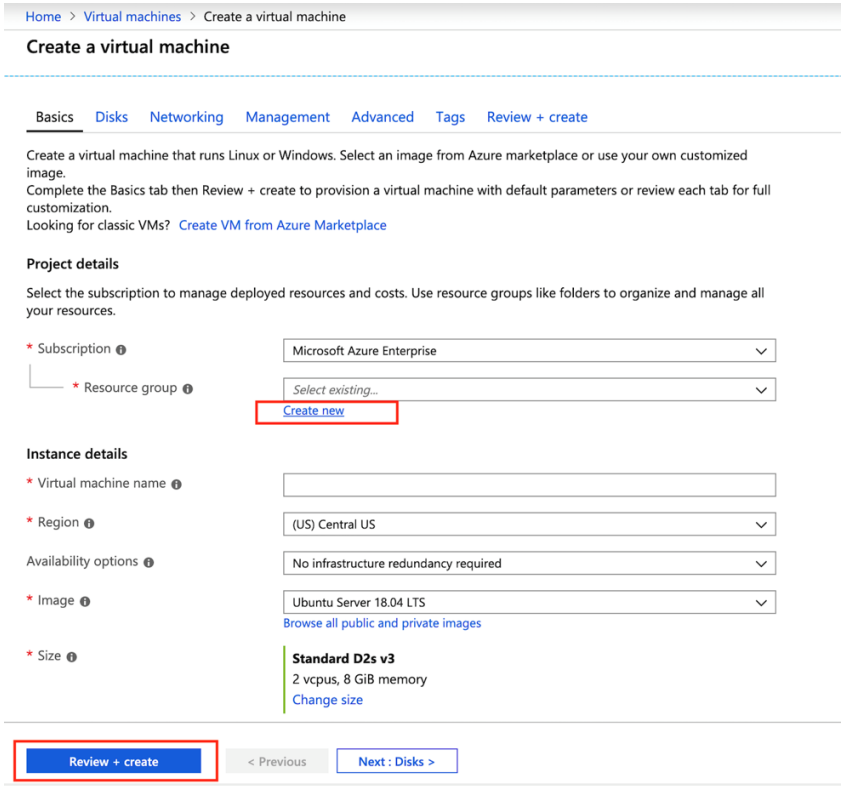


Figure 66. Initiate VM deployment

2. Fill in the pertinent details, then click **Review + create**.
3. Follow subsequent prompts to finish VM creation.
4. Proceed to the next section after the deployment is complete.

©2026 Zscaler, Inc. All rights reserved.

Allowing Inbound Ports

Configure the Network Security Group (NSG) to permit inbound SSH access for management, and to allow inbound TCP/514 from NSS and the OMS. By default, these ports are not permitted in Azure and you must manually allow inbound connections to TCP/22 and TCP/514. Lock down this access to only permitted specific source IPs.

On the Azure home page:

1. In the Search bar at the top, enter `Network Security Groups`.
2. Select **Network security groups**. A new security group is automatically created by Azure with your resource group tied to it.

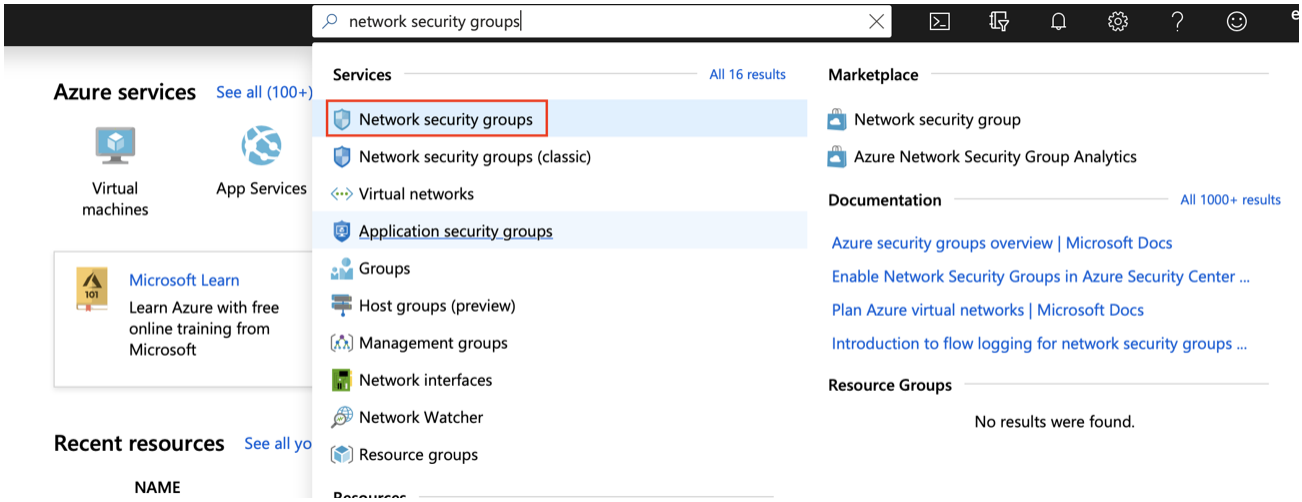


Figure 67. Network security groups

3. Open the auto-created network security group.

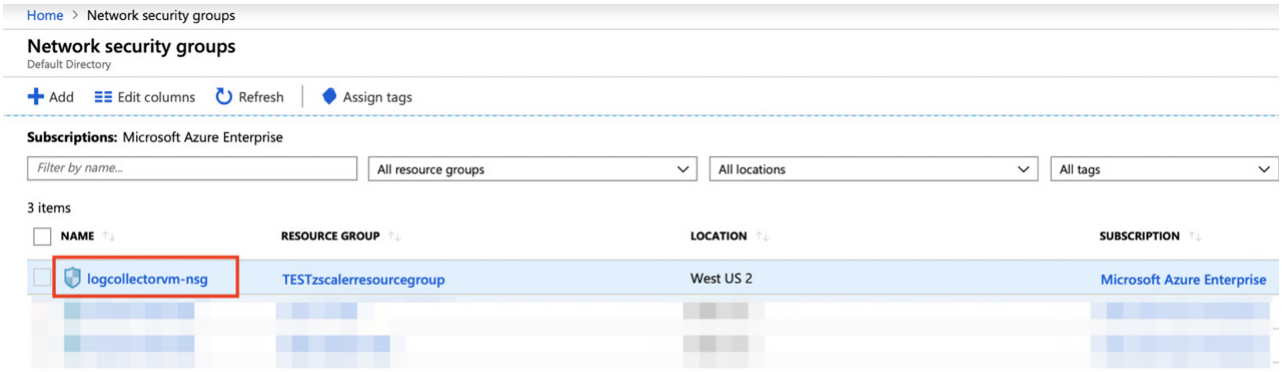


Figure 68. Network security groups, continued

©2026 Zscaler, Inc. All rights reserved.

4. Select the **Inbound security rules** option and configure the following rules to allow inbound connections.
   - Protocol and Port TCP/514 from your Zscaler NSS IP
   - Protocol and Port TCP/22 from your trusted network or management station

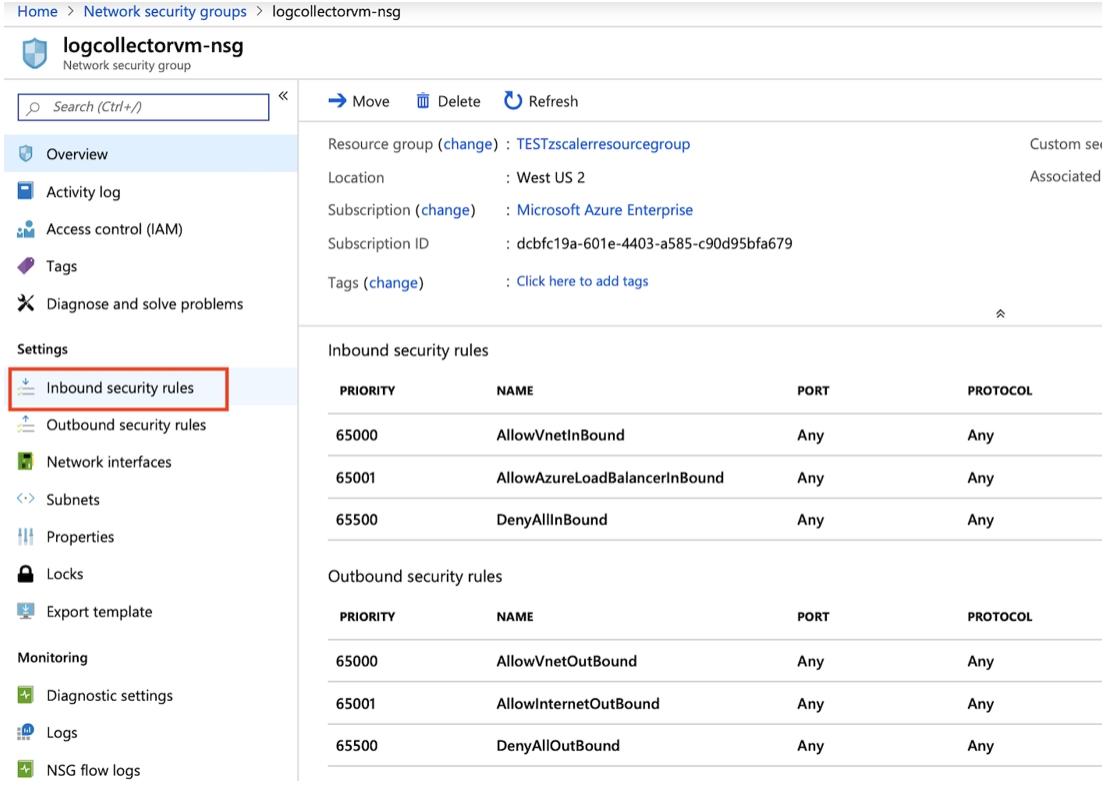


Figure 69. Inbound security rules

Review the following sections to see how to configure each rule separately.

©2026 Zscaler, Inc. All rights reserved.

Add Inbound Security Rule for Syslog

Configure these fields to match your environment. Although the destination port is 514, you can configure the destination on the ZIA side to other ports. If you set this port to something other than 514 in ZIA, the port number must match here. After you are done, select **Add** and proceed to the next section.

For testing purposes, any source IP can connect to this data connector VM on port 514. Post testing, restrict this access to the NSS source IP only.

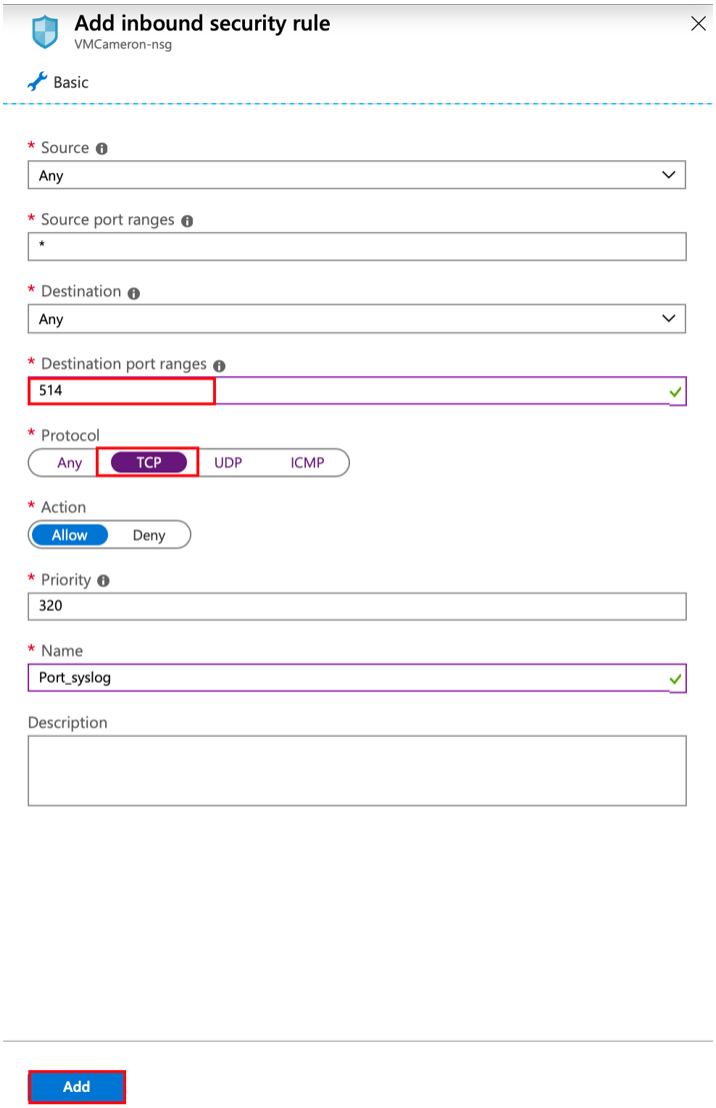


Figure 70. Add inbound security rule for NSS

©2026 Zscaler, Inc. All rights reserved.

Add Inbound Security Rule for SSH

Configure these fields to match your environment. Ensure that the **Source** matches your trusted network or management station from which you want to allow the SSH access.

For testing purposes, any source IP can connect to this data connector VM on port 22. Post testing, restrict this access to the trusted management source IP only.

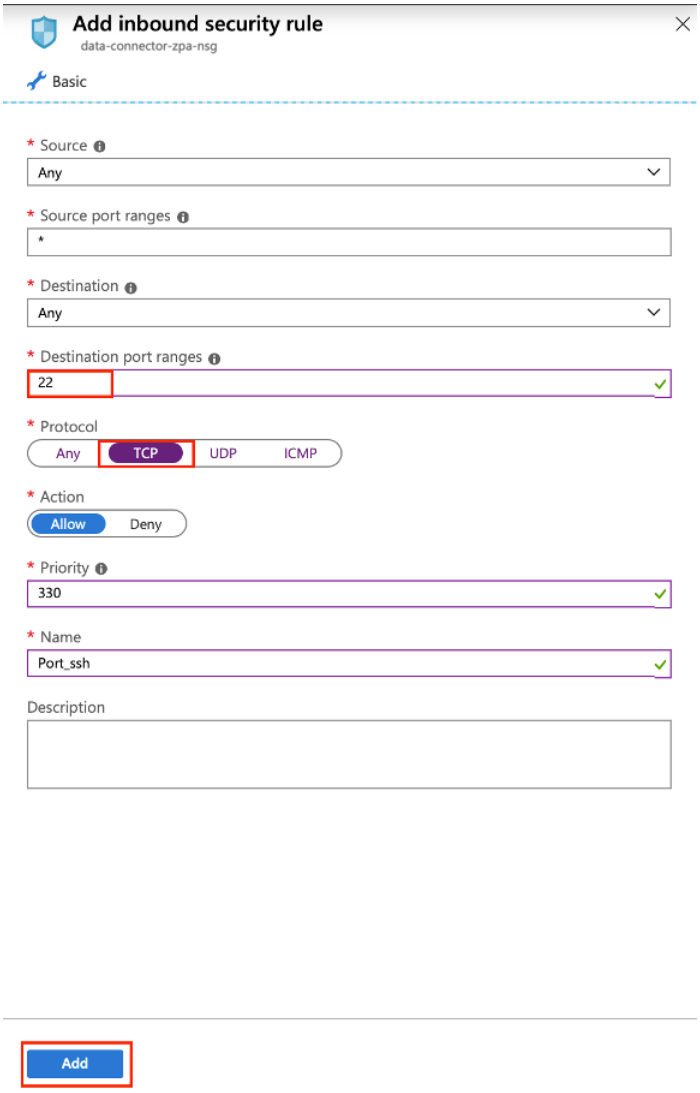


Figure 71. Add inbound security rule for SSH

©2026 Zscaler, Inc. All rights reserved.

Configuring ZIA for NSS VM-Based Log Export

The following sections describe how to configure ZIA to work with NSS.

Logging In to ZIA

First, set up the Zscaler side of this service. Log in to your ZIA Admin Portal using your administrator account. If you are unable to log in using your administrator account, contact Zscaler Support (government agencies, contact Zscaler Support).

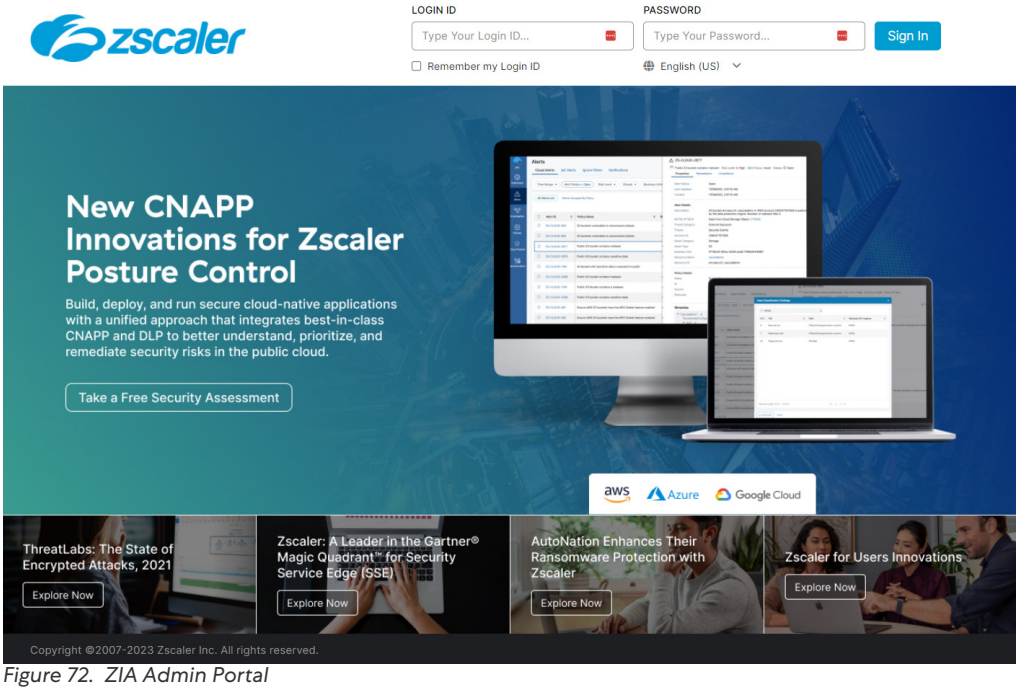


Figure 72. ZIA Admin Portal

Configuring NSS

Log messages sent between Zscaler NSS and Azure data connector are not encrypted.

Zscaler strongly recommends that you spin up your NSS in the same VNet as the Azure data connector VM so that plain text log messages traffic doesn't leave your VNet.

If you are deploying NSS in a different network, use an external mechanism (e.g., IPSec tunnel) to encrypt plain text communication between NSS and the Azure data connector VM.

After logging in to ZIA, add an **NSS server** and **NSS feed**.

1. Go to **Administration** > **Cloud Configuration** > **Nanolog Streaming Service**.
2. Follow the instructions in the NSS Deployment Guide help (government agencies, see NSS Deployment Guide help) based on your deployment type to set up web and firewall NSS.

©2026 Zscaler, Inc. All rights reserved.

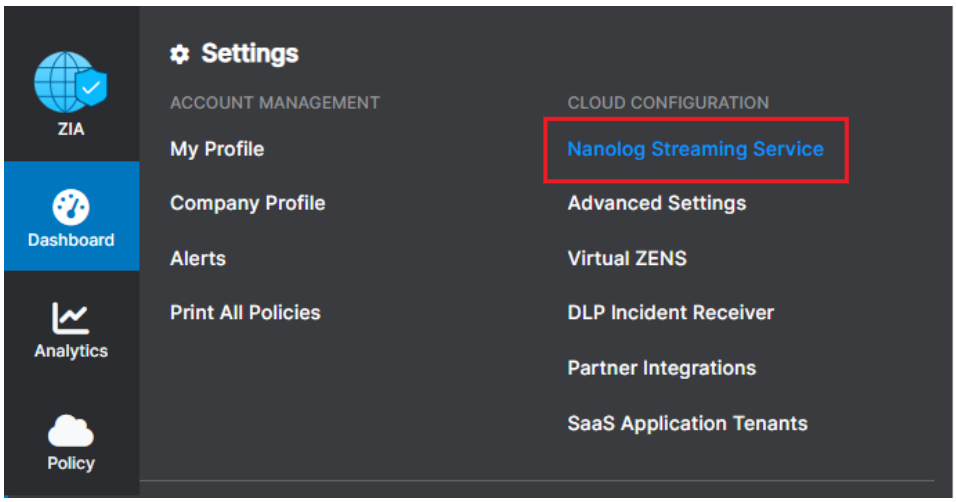


Figure 73. Go to NSS

If you are deploying a new NSS in Azure:

Zscaler NSS Azure Resource Manager (ARM) Template was developed to automate setting up an NSS in Azure. This avoids the need for manually running PowerShell scripts.

Deployment can take up to one hour to finish.

Access the Zscaler NSS ARM Template from the Zscaler GitHub repository.

After deploying the ARM template deployment, check the IP address using the `ifconfig -a` command. Configure another IP in the same network range. NSS requires two interfaces in same subnet.

Connect to the NSS VM using SSH and execute the following commands as root user. Change the IPs as needed to fit your environment.

```
[root@NSS /usr/home/zsroot]# nss configure
Detected an Azure VM!!
nameserver:168.63.129.16 (Options <c:change, d:delete, n:no change>) [n]
Do you wish to add a new nameserver? <n:no y:yes> [n]:
smnet_dev (Service interface IP address with netmask) [192.168.100.4/24]: 192.168.100.5/24
smnet_dflt_gw (Service interface default gateway IP address) [192.168.100.1]: 192.168.100.1
Successfully Applied Changes
[root@NSS /usr/home/zsroot]#
[root@NSS /usr/home/zsroot]# nss update-now
Connecting to server...
Connecting to update server 104.129.193.117.
Installed build version: 209302
Latest available build version: 209302
Build file is up-to-date

Checking if installation required...
Service is up-to-date.
[root@NSS /usr/home/zsroot]# nss restart
Detected an Azure VM!!
NSS service stopped
NSS service running with pid 3000
[root@NSS /usr/home/zsroot]# nss enable-autostart
Detected an Azure VM!!
Auto-start of NSS enabled
[root@NSS /usr/home/zsroot]#
```

©2026 Zscaler, Inc. All rights reserved.

Verify NSS Server State

Before proceeding to further steps, ensure that NSS **State** is **Healthy**. If NSS is **Unhealthy**, see the troubleshooting steps listed in Zscaler Resources. If everything is as expected, proceed to the next section.

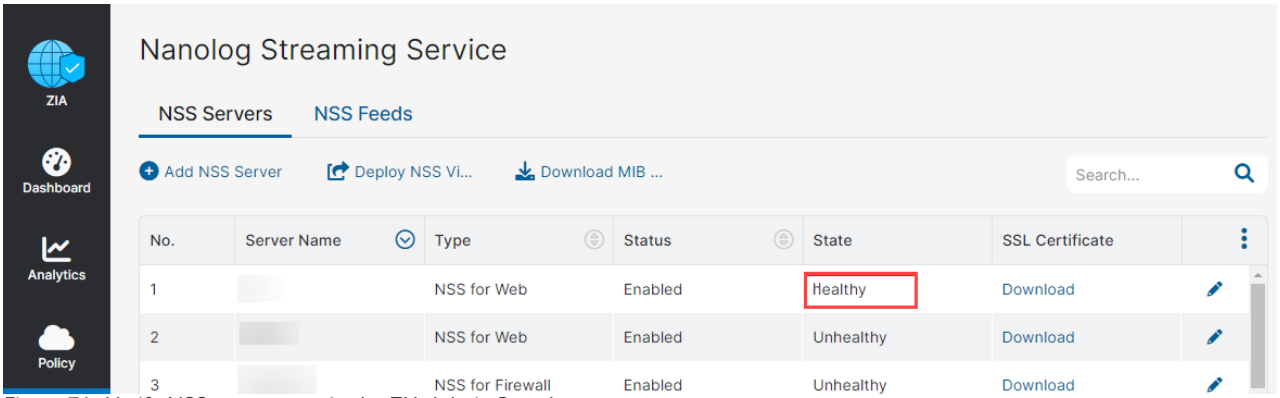


Figure 74. Verify NSS server state in the ZIA Admin Portal

Add NSS Feed

An NSS feed specifies what data from the logs that the NSS sends to the SIEM. Each feed can have a different list of fields, a different format, and different filters. You can add one or more fields for the logs and one field for alerts.

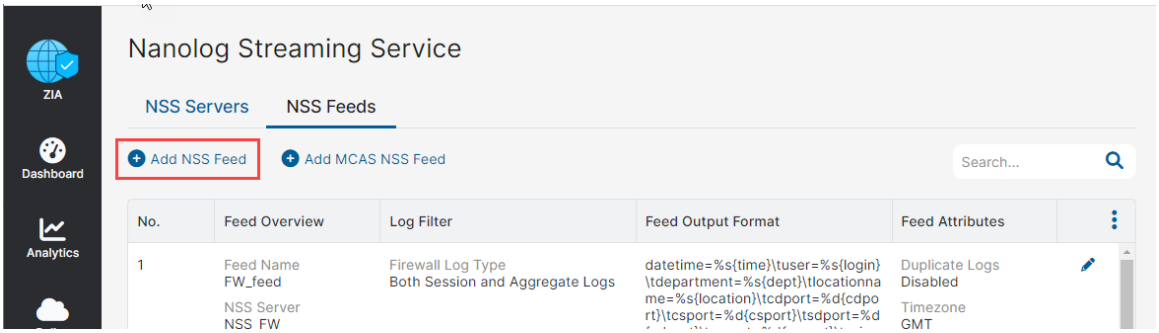


Figure 75. Add NSS feed in the ZIA Admin Portal

©2026 Zscaler, Inc. All rights reserved.

Configure NSS Feed

To configure the NSS feed, ensure you complete the following required fields:

1. **NSS Type**: Select **NSS for Web** or **NSS for Firewall**, **Cloud & Branch Connector**.
2. **NSS Server**: Select your configured NSS Server.
3. **SIEM IP Address**: Enter the public IPv4 address of your Microsoft Sentinel data connector.
4. **SIEM TCP Port**: Enter `514`.
5. **(For Cloud & Branch Connector) Log Domain**: Select **Firewall** or **Cloud & Branch Connector**.
6. **Log Type**: Select your desired log type.
7. **Feed Output Type**: Set to **Custom** from the drop-down menu.
8. **Feed Escape Character**: Enter `"\,=`.
9. **Feed Output Format**: Enter one of the CEF strings from the following pages.

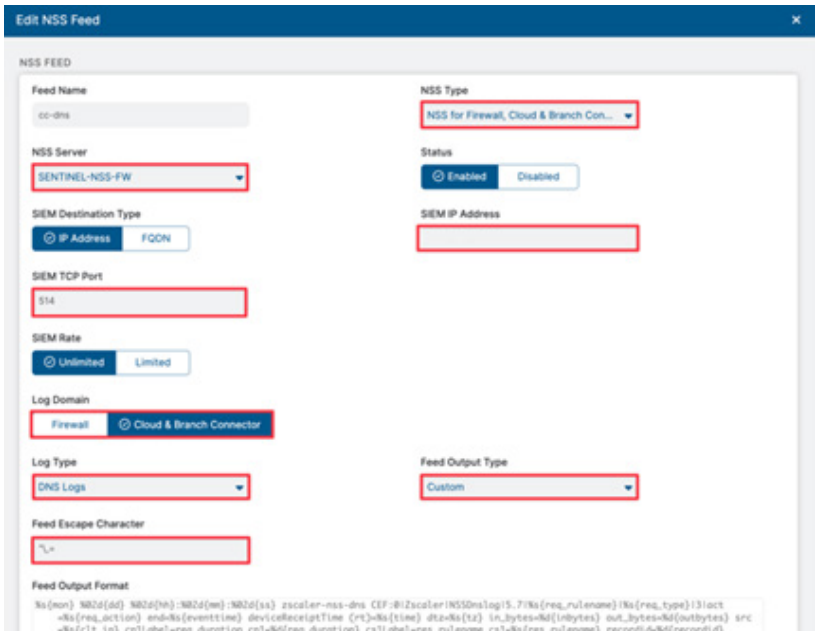


Figure 76. Configure NSS feed

When configuring the log feed to Sentinel, enter `"\,=` in the Feed Escape Character field. All CEF feeds for the following section are available in the Zscaler GitHub repo with an extension of .cef, (i.e., nss-audit.cef for the Admin Audit feed).

If copying and pasting from the GitHub repo, note there is a newline character at the end of the string which must be included, or alternatively, you can manually append it.

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (Admin Audit)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-audit CEF:0|Zscale
r|NSSAuditlog|5.7|%s{action}|%s{category}|3| Severity=3 name=%s{category}
deviceEventClassId=%s{action} act=%s{action} cs1Label=category
cs1=%s{category} cs2Label=subcategory cs2=%s{subcategory} cs3Label=resource
cs3=%s{resource} cat=%s{interface} suser=%s{adminid} src=%s{clientip}
outcome=%s{result} reason=%s{errorcode} sourceServiceName=%s{auditlogtype}
cs4Label=preaction cs4=%s{epreaction} cs5Label=postaction cs5=%s{epostaction}
deviceExternalId=%d{recordid}\n
```

Edit NSS Feed (SaaS Security Activity)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb CEF:0|Zsca
ler|NSSCasbactivitylog|5.7|%s{appname}|%s{act_type_name}|3| Severity=3
name=%s{act_type_name} suser=%s{username} cs1Label=tenant cs1=%s{tenant}
cs2Label=objtypename1 cs2=%s{objtypename1} cs3Label=objtypename2 cs3=%s{objtypename2}
deviceEventClassId=%s{appname} cs4Label=objnames1 cs4=%s{objnames1}
cs5Label=objnames2 cs5=%s{objtypename2} eventCount=%d{act_cnt} act=%s{act_type_name}
cs6Label=extownername cs6=%s{extownername} src=%s{src_ip} flexString1Label=isadmin
flexString1=%s{is_admin_act}\n
```

Edit NSS Feed (SaaS Public Cloud Storage)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb CEF:0|Zscaler|
NSSCasbcloudstoragelog|5.7|%s{applicationname}|%s{rulelabel}|3| Severity=3
name=%s{rulelabel} deviceEventClassId=%s{applicationname} act=%s{policy}
bucketid=%d{bucketid} cs1=%d{dlpidentifier} cs1Label=dlpidentifier
start=%d{epochtime} numcollab=%d{numcollab} deviceExternalId=%d{recordid}
bucketname=%s{bucketname} bucketowner=%s{bucketowner} collabnames=%s{collabnames}
company=%s{company} datacenter=%s{datacenter} datacentercity=%s{datacente
rcity} datacentercountry=%s{datacentercountry} department=%s{department}
cs2=%s{dlpdictcount} cs2Label=dlpdictcount cs3=%s{dlpdictnames}
cs3Label=dlpdictnames cs4=%s{dlpenginenames} cs4Label=dlpenginenames
extownername=%s{extownername} fileId=%s{fileid} request=%s{fullurl} dhost=%s{hostname}
flexString1Label=malware flexString1=%s{malware} flexString2Label=malwareclass
flexString2=%s{malwareclass} obucketname=%s{obucketname} obucketowner=%s{obucketowner}
ocollabnames=%s{ocollabnames} odlpdictnames=%s{odlpdictnames} odlpenginenames=%s{odlp
enginenames} oextownername=%s{oextownername} ofileid=%s{ofileid} ofullurl=%s{ofullurl}
ohostname=%s{ohostname} oowner=%s{oowner} orulelabel=%s{orulelabel}
otenant=%s{otenant} suser=%s{owner} cs5=%s{rulelabel} cs5Label=rulelabel
cs6=%s{ruletype} cs6Label=ruletype agentSeverity=%s{severity} tenant=%s{tenant}
threatname=%s{threatname} deviceCustomDate1Label=time deviceCustomDate1=%s{time}
dtz=%s{tz} upload_doctypename=%s{upload_doctypename}\n
```

©2026 Zscaler, Inc. All rights reserved.

SaaS Security Collaboration

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb
CEF:0|Zscaler|NSSCasbcollablog|5.7|%s{applicationname}|%s{rulelabel}|3|
Severity=3 name=%s{rulelabel} deviceEventClassId=%s{applicationname}
act=%s{policy} cs1=%d{dlpidentifier} cs1Label=dlpidentifier start=%d{epochtime}
deviceExternalId=%d{recordid} outcome=%s{any_incident} channel_name=%s{channel_
name} company=%s{company} datacenter=%s{datacenter} datacentercity=%s{datace
ntercity} datacentercountry=%s{datacentercountry} department=%s{department}
cs2=%s{dlpdictcount} cs2Label=dlpdictcount cs3=%s{dlpdictnames} cs3Label=dlpdictnames
cs4=%s{dlpenginenames} cs4Label=dlpenginenames external_recptnames=%s{external_
recptnames} extownername=%s{extownername} internal_recptnames=%s{internal_recptnames}
flexString1Label=malware flexString1=%s{malware} flexString2Label=malwareclass
flexString2=%s{malwareclass} msgid=%s{msgid} ochannel_name=%s{ochannel_
name} odlpdictnames=%s{odlpdictnames} odlpenginenames=%s{odlpenginenames}
oexternal_recptnames=%s{oexternal_recptnames} oextownername=%s{oextownername}
ointernal_recptnames=%s{ointernal_recptnames} omsgid=%s{omsgid}
oowner=%s{oowner} orulelabel=%s{orulelabel} osender=%s{osender} osharedchannel_
hostname=%s{osharedchannel_hostname} otenant=%s{otenant} suser=%s{owner}
cs5=%s{rulelabel} cs5Label=rulelabel cs6=%s{ruletype} cs6Label=ruletype
sender=%s{sender} agentSeverity=%s{severity} sharedchannel_hostname=%s{sharedchannel_
hostname} tenant=%s{tenant} threatname=%s{threatname} deviceCustomDate1Label=time
deviceCustomDate1=%s{time} dtz=%s{tz} upload_doctypename=%s{upload_doctypename}\n
```

Edit NSS Feed (SaaS Security CRM)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb CEF:0|Zscaler|NSSCa
sbcrmlog|5.7|%s{applicationname}|%s{rulelabel}|3| Severity=3 name=%s{rulelabel}
deviceEventClassId=%s{applicationname} act=%s{policy} cs1=%d{dlpidentifier}
cs1Label=dlpidentifier start=%d{epochtime} fsize=%d{filesize} cn1=%d{num_external_
collab} cn1Label=num_external_collab cn2=%d{num_internal_collab} cn2Label=num_
internal_collab deviceExternalId=%d{recordid} collabscope=%s{collabscope}
company=%s{company} component=%s{component} datacenter=%s{datacenter} datacentercity=
%s{datacentercity} datacentercountry=%s{datacentercountry} department=%s{department}
cs2=%s{dlpdictcount} cs2Label=dlpdictcount cs3=%s{dlpdictnames}
cs3Label=dlpdictnames cs4=%s{dlpenginenames} cs4Label=dlpenginenames
external_collabnames=%s{external_collabnames} extownername=%s{extownername}
fileId=%s{file_msg_id} fileModificationTime=%s{file_msg_mod_time} fileHash=%s{filemd5}
fname=%s{filename} filePath=%s{filepath} fileType=%s{filetypecategory}
request=%s{fullurl} dhost=%s{hostname} internal_collabnames=%s{internal_collabnames}
flexString1Label=malware flexString1=%s{malware} flexString2Label=malwareclass
flexString2=%s{malwareclass} objectname=%s{objectname} objecttype=%s{objecttype}
odlpdictnames=%s{odlpdictnames} odlpenginenames=%s{odlpenginenames} oexternal_
collabnames=%s{oexternal_collabnames} oextownername=%s{oextownername} ofile_msg_
id=%s{ofile_msg_id} ofullurl=%s{ofullurl} ohostname=%s{ohostname} ointernal_
collabnames=%s{ointernal_collabnames} oowner=%s{oowner} orulelabel=%s{orulelabel}
otenant=%s{otenant} suser=%s{owner} cs5=%s{rulelabel} cs5Label=rulelabel
cs6=%s{ruletype} cs6Label=ruletype agentSeverity=%s{severity} tenant=%s{tenant}
threatname=%s{threatname} deviceCustomDate1Label=time deviceCustomDate1=%s{time}
dtz=%s{tz} upload_doctypename=%s{upload_doctypename}\n
```

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (SaaS Security Email)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb
CEF:0|Zscaler|NSSCasbemaillog|5.7|%s{applicationname}|%s{rulelabel}|3| Severity=3
name=%s{rulelabel} deviceEventClassId=%s{applicationname} act=%s{policy} oexternalow
nername=%s{oexternalownername} companyid=%d{companyid} dlpidentifier=%d{dlpidentifier}
epochtime=%d{epochtime} filedownloadtimems=%d{filedownloadtimems} filescantimems=%
d{filescantimems} msgsize=%d{msgsize} num_ext_recpts=%d{num_ext_recpts} num_int_
recpts=%d{num_int_recpts} deviceExternalId=%d{recordid} repochtime=%d{repochtime}
outcome=%s{any_incident} fname=%s{attchcomponentfilenames} fsize=%s{attchcompon
entfilenames} fileType=%s{attchcomponentfiletypes} fileHash=%s{attchcomponentmd5s}
company=%s{company} datacenter=%s{datacenter} datacentercity=%s{datacente
rcity} datacentercountry=%s{datacentercountry} department=%s{department}
dlpdictcount=%s{dlpdictcount} dlpdictnames=%s{dlpdictnames} dlpenginenames=%s{dlpen
ginenames} externalownername=%s{externalownername} extrecptnames=%s{extrecptnames}
intrecptnames=%s{intrecptnames} is_inbound=%s{is_inbound} malware=%s{malware}
malwareclass=%s{malwareclass} messageid=%s{messageid} oattchcomponentfile
names=%s{oattchcomponentfilenames} odlpdictnames=%s{odlpdictnames} odlpe
nginenames=%s{odlpenginenames} oextrecptnames=%s{oextrecptnames} ointre
cptnames=%s{ointrecptnames} omessageid=%s{omessageid} oowner=%s{oowner}
orulelabel=%s{orulelabel} otenant=%s{otenant} owner=%s{owner} rtime=%s{rtime}
rulelabel=%s{rulelabel} ruletype=%s{ruletype} severity=%s{severity} tenant=%s{tenant}
threatname=%s{threatname} tz=%s{tz} upload_doctypename=%s{upload_doctypename}\n
```

Edit NSS Feed (SaaS Security File)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb CEF:0|Zscaler|NSSCasbfil
esharinglog|5.7|%s{applicationname}|%s{rulelabel}|3| Severity=3 name=%s{rulelabel}
deviceEventClassId=%s{applicationname} act=%s{policy} cs1=%d{dlpidentifier}
cs1Label=dlpidentifier fileModificationTime=%d{epochlastmodtime} start=%d{epochtime}
filedownloadtimems=%d{filedownloadtimems} filescantimems=%d{filescantimems
} fsize=%d{filesize} deviceExternalId=%d{recordid} collabscope=%s{collabscope}
company=%s{company} datacenter=%s{datacenter} datacentercity=%s{datacente
rcity} datacentercountry=%s{datacentercountry} department=%s{department}
cs2=%s{dlpdictcount} cs2Label=dlpdictcount cs3=%s{dlpdictnames} cs3Label=dlpdictnames
cs4=%s{dlpenginenames} cs4Label=dlpenginenames extcollabnames=%s{extcolla
bnames} extownername=%s{extownername} fileId=%s{fileid} fileHash=%s{filemd5}
fname=%s{filename} filePath=%s{filesource} fileType=%s{filetypename} request=%s{fullurl}
dhost=%s{hostname} intcollabnames=%s{intcollabnames} flexString1Label=malware
flexString1=%s{malware} flexString2Label=malwareclass flexString2=%s{malwareclass}
odlpdictnames=%s{odlpdictnames} odlpenginenames=%s{odlpenginenames} oextcolla
bnames=%s{oextcollabnames} oextownername=%s{oextownername} ofileid=%s{ofileid}
efullurl=%s{efullurl} ehostname=%s{ehostname} ointcollabnames=%s{ointcollabnames}
orulelabel=%s{orulelabel} otenant=%s{otenant} ouser=%s{ouser} cs5=%s{rulelabel}
cs5Label=rulelabel cs6=%s{ruletype} cs6Label=ruletype agentSeverity=%s{severity}
tenant=%s{tenant} threatname=%s{threatname} deviceCustomDate1Label=time
deviceCustomDate1=%s{time} dtz=%s{tz} upload_doctypename=%s{upload_doctypename}
suser=%s{user}\n
```

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (SaaS Security ITSM)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb
CEF:0|Zscaler|NSSCasbitsmlog|5.7|%s{applicationname}|%s{rulelabel}|3| Severity=3
name=%s{rulelabel} deviceEventClassId=%s{applicationname} act=%s{policy}
cs1=%d{dlpidentifier} cs1Label=dlpidentifier start=%d{epochtime} fsize=%d{filesize}
cn1=%d{num_external_collab} cn1Label=num_external_collab cn2=%d{num_internal_collab}
cn2Label=num_internal_collab deviceExternalId=%d{recordid} company=%s{company}
component=%s{component} datacenter=%s{datacenter} datacentercity=%s{datacen
tercity} datacentercountry=%s{datacentercountry} department=%s{department}
cs2=%s{dlpdictcount} cs2Label=dlpdictcount cs3=%s{dlpdictnames}
cs3Label=dlpdictnames cs4=%s{dlpenginenames} cs4Label=dlpenginenames
external_collabnames=%s{external_collabnames} extownername=%s{extownername}
fileId=%s{file_msg_id} fileModificationTime=%s{file_msg_mod_time} fileHash=%s{filemd5}
fname=%s{filename} filePath=%s{filepath} fileType=%s{filetypecategory}
request=%s{fullurl} dhost=%s{hostname} internal_collabnames=%s{internal_collabnames}
flexString1Label=malware flexString1=%s{malware} flexString2Label=malwareclass
flexString2=%s{malwareclass} objectname=%s{objectname} objecttype=%s{objecttype}
odlpdictnames=%s{odlpdictnames} odlpenginenames=%s{odlpenginenames} oexternal_
collabnames=%s{oexternal_collabnames} oextownername=%s{oextownername} ofile_msg_
id=%s{ofile_msg_id} ofullurl=%s{ofullurl} ohostname=%s{ohostname} ointernal_
collabnames=%s{ointernal_collabnames} oowner=%s{oowner} orulelabel=%s{orulelabel}
otenant=%s{otenant} suser=%s{owner} cs5=%s{rulelabel} cs5Label=rulelabel
cs6=%s{ruletype} cs6Label=ruletype agentSeverity=%s{severity} tenant=%s{tenant}
threatname=%s{threatname} deviceCustomDate1Label=time deviceCustomDate1=%s{time}
dtz=%s{tz} upload_doctypename=%s{upload_doctypename}\n
```

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (SaaS Security Repository)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-casb
CEF:0|Zscaler|NSSCasbrepolog|5.7|%s{applicationname}|%s{rulelabel}|3|
Severity=3 name=%s{rulelabel} deviceEventClassId=%s{applicationname}
act=%s{policy} cs1=%d{dlpidentifier} cs1Label=dlpidentifier start=%d{epochtime}
fsize=%d{filesize} cn1=%d{num_external_collab} cn1Label=num_external_collab
deviceExternalId=%d{recordid} company=%s{company} datacenter=%s{datacenter}
datacentercity=%s{datacentercity} datacentercountry=%s{datacentercount
ry} department=%s{department} cs2=%s{dlpdictcount} cs2Label=dlpdictcount
cs3=%s{dlpdictnames} cs3Label=dlpdictnames cs4=%s{dlpenginenames}
cs4Label=dlpenginenames external_collabnames=%s{external_collabnames}
extownername=%s{extownername} fileId=%s{fileid} fileHash=%s{filemd5}
fname=%s{filename} filePath=%s{filepath} fileType=%s{filetypecategory}
internal_collabnames=%s{internal_collabnames} flexString1Label=malware
flexString1=%s{malware} flexString2Label=malwareclass flexString2=%s{malwareclass}
odlpdictnames=%s{odlpdictnames} odlpenginenames=%s{odlpenginenames} oexternal_
collabnames=%s{oexternal_collabnames} oextownername=%s{oextownername}
ofileid=%s{ofileid} ointernal_collabnames=%s{ointernal_collabnames}
oowner=%s{oowner} orulelabel=%s{orulelabel} otenant=%s{otenant} suser=%s{owner}
projectname=%s{projectname} reponame=%s{reponame} cs5=%s{rulelabel}
cs5Label=rulelabel cs6=%s{ruletype} cs6Label=ruletype agentSeverity=%s{severity}
tenant=%s{tenant} threatname=%s{threatname} deviceCustomDate1Label=time
deviceCustomDate1=%s{time} dtz=%s{tz} upload_doctypename=%s{upload_doctypename}\n
```

Edit NSS Feed (DNS)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-fw CEF:0|Zscaler|NSSDNSl
og|5.7|%s{action}|%s{rulelabel}|3| act=%s{action} Severity=3 name=%s{rulelabel}
deviceEventClassId=%s{action} suser=%s{login} cs1=%s{dept} cs1Label=department
cs2=%s{reqaction} cs2Label=reqaction cs3=%s{resaction} cs3Label=resaction
cs4=%s{reqtype} cs4Label=dns_reqtype cs5=%s{req} cs5Label=dns_req cs6=%s{res}
cs6Label=dns_resp cn1=%d{durationms} cn1Label=durationms flexString1=%s{reqrulelabel}
flexString1Label=reqrulelabel flexString2=%s{resrulelabel} flexString2Label=resrulelabel
cat=%s{domcat} src=%s{cip} dst=%s{sip} dpt=%d{sport} spriv=%s{location}
suid=%s{deviceowner} dvchost=%s{devicehostname}\n
```

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (Endpoint DLP)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-edlp CEF:0|Zscaler|NSSEnd
pointdlplog|5.7|%s{logtype}|%s{activitytype}|3| Severity=3 name=%s{activitytype}
rt=%s{time} deviceCustomDate1Label=rtime deviceCustomDate1=%s{rtime} devic
eCustomDate2Label=eventtime deviceCustomDate2=%s{eventtime} dtz=%s{tz} de
viceCustomNumber1Label=scantime cn1=%llu{scantime} deviceCustomNumber
2Label=numdlpengids cn2=%u{numdlpengids} deviceCustomNumber3Label=nu
mdlpdictids cn3=%u{numdlpdictids} externalId=%llu{recordid} scanned_
bytes=%llu{scanned_bytes} dlpidentifier=%llu{dlpidentifier} suser=%s{user}
ouser=%s{ouser} department=%s{department} odepartment=%s{odepartment}
devicename=%s{devicename} odevicename=%s{odevicename} devicetype=%s{devicetype}
deviceostype=%s{deviceostype} deviceplatform=%s{deviceplatform} deviceosversion=%s{
deviceosversion} devicemodel=%s{devicemodel} deviceappversion=%s{deviceappversion}
deviceowner=%s{deviceowner} odeviceowner=%s{odeviceowner} dvchost=%s{devicehostname}
odevicehostname=%s{odevicehostname} datacenter=%s{datacenter} datacentercity=%
s{datacentercity} datacentercountry=%s{datacentercountry} dsttype=%s{dsttype}
filedoctype=%s{filedoctype} filedstpath=%s{filedstpath} fileHash=%s{filemd5}
filesha=%s{filesha} filePath=%s{filesrcpath} filetypecategory=%s{filetypecatego
ry} fileType=%s{filetypename} itemdstname=%s{itemdstname} itemname=%s{itemname}
itemsrcname=%s{itemsrcname} itemtype=%s{itemtype} ofiledstpath=%s{ofiledstpath}
ofilesrcpath=%s{ofilesrcpath} oitemdstname=%s{oitemdstname} oitemname=%s{oitemname}
oitemsrcname=%s{oitemsrcname} sourceServiceName=%s{srctype} act=%s{actiontaken}
cs1Label=activitytype cs1=%s{activitytype} cs2Label=addinfo cs2=%s{addinfo}
cs3Label=channel cs3=%s{channel} cs4Label=confirmaction cs4=%s{confirmaction}
cs5Label=confirmjust cs5=%s{confirmjust} cs6Label=dlpcounts cs6=%s{dlpcounts}
dlpdictnames=%s{dlpdictnames} dlpengnames=%s{dlpengnames} expectedaction=%s{e
xpectedaction} deviceEventClassId=%s{logtype} odlpdictnames=%s{odlpdictnames}
odlpengnames=%s{odlpengnames} ootherrulelabels=%s{ootherrulelabels} othe
rrulelabels=%s{otherrulelabels} otriggeredrulelabel=%s{otriggeredrulelab
el} agentSeverity=%s{severity} triggeredrulelabel=%s{triggeredrulelabel}
zdpmode=%s{zdpmode}\n
```

Edit NSS Feed (Email DLP)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscalernss-emaildlp CEF:0|Zscal
er|NSSEmaildlplog|5.7|%s{logtype}|%s{actions}|3| Severity=3 act=%s{actions}
deviceEventClassId=%s{logtype} agentSeverity=%s{severity} app=%s{appname}
cn1=%llu{scan_time} cn1Label=scan_time cn2=%llu{dlpidentifier} cn2Label=dlpidentifier
cs1=%s{dlpdictnames} cs1Label=dlpdictnames cs2=%s{dlpdictcnts} cs2Label=dlpdictcnts
cs3=%s{dlpengnames} cs3Label=dlpengnames cs4=%s{rulelabels} cs4Label=rulelabels
cs5=%s{mail_sent_time} cs5Label=mail_sent_time cs6=%s{subject} cs6Label=subject
suser=%s{owner} username=%s{username} extusername=%s{extusername} sender=%s{sender}
flexString1=%s{trigg_rcpt_doms} flexString1Label=trigg_rcpt_doms flexString2=%s{other_
rcpt_doms} flexString2Label=other_rcpt_doms datacenter=%s{datacenter} datacentercity
=%s{datacentercity} datacentercountry=%s{datacentercountry} company=%s{company} de
partmentname=%s{departmentname} tenant=%s{tenant} msgid=%s{msgid} ac_md5s=%s{ac_
md5s} ac_sizes=%s{ac_sizes} ac_filetypes=%s{ac_filetypes} ac_doctypes=%s{ac_doctypes}
ac_names=%s{ac_names} trigg_rcpts=%s{trigg_rcpts} other_rcpts=%s{other_rcpts}
deviceExternalId=%llu{recordid} Name=%s{actions}\n
```

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (Firewall)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-fw CEF:0|Zscaler
|NSSFWlog|5.7|%s{action}|%s{rulelabel}|3| Severity=3 name=%s{rulelabel}
deviceEventClassId=%s{action} act=%s{action} suser=%s{login} src=%s{csip}
spt=%d{csport} dst=%s{cdip} dpt=%d{cdport} deviceTranslatedAddress=%s{s
sip} deviceTranslatedPort=%d{ssport} destinationTranslatedAddress=%s{sd
ip} destinationTranslatedPort=%d{sdport} sourceTranslatedAddress=%s{ts
ip} sourceTranslatedPort=%d{tsport} proto=%s{ipproto} tunnelType=%s{ttype}
dnat=%s{dnat} stateful=%s{stateful} spriv=%s{location} reason=%s{rulelabel} in_
bytes=%ld{inbytes} in=%ld{inbytes} out=%ld{outbytes} deviceDirection=1 cs1=%s{dept}
cs1Label=dept cs2=%s{nwsvc} cs2Label=nwService cs3=%s{nwapp} cs3Label=nwApp
cs4=%s{aggregate} cs4Label=aggregated cs5=%s{threatcat} cs5Label=threatcat
cs6=%s{threatname} cs6label=threatname cn1=%d{durationms} cn1Label=durationms
cn2=%d{numsessions} cn2Label=numsessions flexString1Label=ipCat flexString1=%s{ipcat}
destCountry=%s{destcountry} avgduration=%d{avgduration}\n
```

Edit NSS Feed (Tunnel)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

Tunnel logs - IKE Phase 1

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel CEF:0|Zsca
ler|NSSTunnellog|5.7|%s{tunnelactionname}|%s{destvip}|3| dpt=%d{dstport}
cn1Label=ikeversion cn1=%d{ikeversion} cfp1Label=lifetime cfp1=%d{lifetime}
spt=%d{srcport} spi_in=%lu{spi_in} spi_out=%lu{spi_out} cs1Label=algo cs1=%s{algo}
cs2Label=authentication cs2=%s{authentication} cs3Label=authtype cs3=%s{authtype}
dst=%s{destvip} cs4Label=locationname cs4=%s{locationname} src=%s{sourceip} deviceEve
ntClassId=%s{tunnelactionname} vendorname=%s{vendorname} suser=%s{vpncredentialname}
deviceExternalId=%d{recordid} olocationname=%s{olocationname} ovpncredentialname=%s{o
vpncredentialname} dtz=%s{tz}\n
```

Tunnel logs - IKE Phase 2

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel
CEF:0|Zscaler|NSSTunnellog|5.7|%s{tunnelactionname}|%s{tunnelprotocol}|3|
destipstart=%s{destipstart} dst=%s{destvip} cs4Label=locationname
cs4=%s{locationname} proto=%s{protocol} src=%s{sourceip} srcipend=%s{srcipend}
srcipstart=%s{srcipstart} deviceEventClassId=%s{tunnelactionname} proto=%s{protocol}
suser=%s{vpncredentialname} destportstart=%d{destportstart} cn1Label=ikeversion
cn1=%d{ikeversion} lifebytes=%d{lifebytes} cfp1Label=lifetime cfp1=%d{lifetime}
spi=%d{spi} srcportstart=%d{srcportstart} cs1Label=algo cs1=%s{algo}
cs2Label=authentication cs2=%s{authentication} cs3Label=authtype cs3=%s{authtype}
destipend=%s{destipend}\n
```

©2026 Zscaler, Inc. All rights reserved.

Tunnel Events

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel CEF:0|Zscaler|NSS
Tunnellog|5.7|%s{tunnelactionname}|%s{event}|3| spt=%d{srcport} dst=%s{destvip}
name=%s{event} reason=%s{eventreason} cs4Label=locationname cs4=%s{locationname}
src=%s{sourceip} deviceEventClassId=%s{tunnelactionname} cs5Label=tunneltype
cs5=%s{tunneltype} suser=%s{vpncredentialname} deviceExternalId=%d{recordid}
olocationname=%s{olocationname} ovpncredentialname=%s{ovpncredentialname}
dtz=%s{tz}\n
```

Tunnel Samples

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-tunnel CEF:0|Zscal
er|NSSTunnellog|5.7|%s{tunnelactionname}|%s{tunneltype}|3| spt=%d{srcport}
cn2Label=rxpackets cn2=%d{rxpackets} cn3Label=txpackets cn3=%d{txpackets}
dpdrec=%d{dpdrec} in=%lu{rxbytes} out=%lu{txbytes} dst=%s{destvip}
cs4Label=locationname cs4=%s{locationname} src=%s{sourceip} deviceEventClassId=%s{t
unnelactionname} cs5Label=tunneltype cs5=%s{tunneltype} suser=%s{vpncredentialname}
deviceExternalId=%d{recordid} olocationname=%s{olocationname} ovpncredentialname=%s{o
vpncredentialname} dtz=%s{tz}\n
```

Edit NSS Feed (Web)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss CEF:0|Zscal
er|NSSWeblog|5.7|%s{action}|%s{reason}|3|act=%s{action} Severity=3
name=%s{reason} deviceEventClassId=%s{action} app=%s{proto} cat=%s{urlcat}
dhost=%s{ehost} dst=%s{sip} src=%s{cip} in_bytes=%d{respsize} in=%d{respsize}
outcome=%s{respcode} out=%d{reqsize} request=%s{eurl} rt=%s{mon} %02d{dd}
%d{yy} %02d{hh}:%02d{mm}:%02d{ss} sourceTranslatedAddress=%s{cintip}
requestClientApplication=%s{ua} requestMethod=%s{reqmethod} suser=%s{login}
spriv=%s{location} externalId=%d{recordid} fileType=%s{filetype} reason=%s{reason} de
stinationServiceName=%s{appname} cn1=%d{riskscore} cn1Label=riskscore cs1=%s{dept}
cs1Label=dept cs2=%s{urlsupercat} cs2Label=urlsupercat cs3=%s{appclass}
cs3Label=appclass cs4=%s{malwarecat} cs4Label=malwarecat cs5=%s{threatname}
cs5Label=threatname cs6=%s{dlpeng} cs6Label=dlpeng ZscalerNSSWeblogURLClass=%s{u
rlclass} ZscalerNSSWeblogDLPDictionaries=%s{dlpdict} requestContext=%s{ereferer}
contenttype=%s{contenttype} unscannabletype=%s{unscannabletype}
deviceowner=%s{deviceowner} devicehostname=%s{devicehostname} keyprotectiontype=%s{ke
yprotectiontype}\n
```

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (Cloud Connector—Session)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-session CEF:0|Zscaler|NSSS
essionlog|5.7|%s{nw_service}|%s{traffic_type}|3|act=%s{nw_service} end=%s{eventtime}
rt=%s{time} dtz=%s{tz} dhost=%s{clt_dst_name} src=%s{clt_src_ip} spt=%d{clt_src_port}
dst=%s{clt_dst_ip} dpt=%d{clt_dst_port} deviceCustomString2=%s{fwdtypename} deviceCu
stomString2Label=fwdtypename deviceCustomString3=%s{zapptunname} deviceCustomString3
Label=zapptunname deviceCustomNumber1=%d{srv_dst_port} deviceCustomNumber1Label=srv_
dst_port deviceCustomNumber2=%d{gw_dst_port} deviceCustomNumber2Label=gw_dst_
port deviceCustomNumber3=%d{zia_src_port} deviceCustomNumber3Label=zia_src_
port deviceCustomString4=%s{srv_dst_ip} deviceCustomString4Label=srv_dst_ip
deviceCustomString5=%s{gw_name}deviceCustomString5Label=gw_namedeviceCustomString6=%s{ogw_
name} deviceCustomString6Label=ogw_name gw_dst_ip=%s{gw_dst_ip}
deviceCustomString1=%s{zia_src_ip} deviceCustomString1Label=zia_src_ip srv_src_
ip=%s{srv_src_ip} srv_src_port=%d{srv_src_port} srv_ip_country=%s{srv_ip_country}
session_duration=%ld{session_duration} cnt=%d{num_sess} num_aggr_sess=%d{num_aggr_
sess} aggrname=%s{aggrname} rdrrulename=%s{rdrrulename} ordrrulename=%s{ordrrulename}
self_rulename=%s{self_rulename} oself_rulename=%s{oself_rulename} in=%ld{clt_
tx_bytes} out=%ld{clt_rx_bytes} nw_service=%s{nw_service} proto=%s{clt_nw_
protocol} srv_nw_protocol=%s{srv_nw_protocol} zia_gw_protocol=%s{zia_gw_protocol}
app=%s{zpa_appseg} dvchost=%s{name} vm_name=%s{vm_name} group_name=%s{group_name}
region_name=%s{region_name} location=%s{location} platform_name=%s{platform_
name} platformgeo_name=%s{platformgeo_name} account_id=%s{account_id} Severity=3
deviceEventClassId=%s{nw_service} name=%s{traffic_type}\n
```

Edit NSS Feed (Cloud Connector—Event)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-event CEF:0|Zscaler|NS
SEventlog|5.7|%s{event_name}|%s{category}|3| act=%s{category} cat=%s{severity}
deviceEventClassId=%s{event_name} deviceCustomDate2=%d{inteventtime} deviceCustomDa
te2Label=inteventtime deviceCustomDate1=%d{abstime} deviceCustomDate1Label=abstime
flexDate1=%s{time} flexDate1Label=time dtz=%s{tz} rt=%s{rtime} company=%s{company}
location=%s{location} dvchost=%s{name} vm_name=%s{vm_name} group_name=%s{group_
name} platform_name=%s{platform_name} platformgeo_name=%s{platformgeo_name} region_
name=%s{region_name} connector_type=%s{connector_type} deviceCustomString2=%s{attr1_
val} deviceCustomString2Label=%s{attr1_name} deviceCustomString3=%s{attr2_val} dev
iceCustomString3Label=%s{attr2_name} deviceCustomString4=%s{attr3_val} deviceCusto
mString4Label=%s{attr3_name} deviceCustomString5=%s{attr4_val} deviceCustomString5
Label=%s{attr4_name} deviceCustomString6=%s{error_code} account_id=%s{account_id}
name=%{category} Severity=3\n
```

©2026 Zscaler, Inc. All rights reserved.

Edit NSS Feed (Cloud Connector—DNS)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-dns CEF:0|Zscaler|NSSD
nslog|5.7|%s{req_rulename}|%s{req_type}|3|act=%s{req_action} end=%s{eventtime}
deviceReceiptTime (rt)=%s{time} dtz=%s{tz} in_bytes=%d{inbytes} out_
bytes=%d{outbytes} src=%s{clt_ip} cn1Label=req_duration cn1=%d{req_duration}
cs1Label=res_rulename cs1=%s{res_rulename} recordid=%d{recordid} cs2Label=location
cs2=%s{location} destinationDnsDomain=%s{dom_str} odom_str=%s{odom_str} cs3Label=res_
type cs3=%s{res_type} cs4Label=resp_val_str cs4=%s{resp_val_str} oresp_val_
str=%s{oresp_val_str} cs5Label=dns_err cs5=%s{dns_err} dpt=%d{srv_port} cnt=%d{num_
resp} proto=%s{protocol} FlexString1Label=dnsgw_flags FlexString1=%s{dnsgw_flags}
FlexString2Label=dnsgw_slot FlexString2=%s{dnsgw_slot} dvchost=%s{name} vm_
name=%s{vm_name} group_name=%s{group_name} region_name=%s{region_name} platform_
name=%s{platform_name} platformgeo_name=%s{platformgeo_name} account_id=%s{account_
id} dst=%s{resolver} res_action=%s{res_action} deviceEventClassId=%s{req_rulename}
Severity=3 Name=%s{req_type}\n
```

Edit NSS Feed (Cloud Connector—Metrics)

Override the prepopulated CEF feed by replacing it with the following block and then click **Save**.

Device

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-metrics CEF:0|Zscaler|NSSMet
ricslog|5.7|resource|%s{vm_name}|3| DeviceEventClassId=resource cat=%s{resourcename}
flexDate1=%s{time} flexDate1Label=time deviceCustomDate1=%s{eventtime} device
CustomDate1Label=eventtime deviceCustomDate2=%d{inteventtime} deviceCustomD
ate2Label=inteventtime rt=%s{rtime} dtz=%s{tz} rss=%02d{rss} rmm=%02d{rmm}
rhh=%02d{rhh} rdd=%02d{rdd} rmth=%02d{rmth} ryyyy=%04d{ryyyy} rdd=%02d{rdd}
rmth=%02d{rmth} deviceExternalId=%s{recordid} location=%s{location} group_
name=%s{group_name} vm_name=%s{vm_name} platform_name=%s{platform_name} platformgeo_
name=%s{platformgeo_name} region_name=%s{region_name} connector_type=%s{connector_
type} cs1Label=rdiskUtil cs1=%s{rdiskutil} cs2Label=rdiskUsed cs2=%s{rdiskused}
cs3Label=rdiskFree cs3=%s{rdiskfree} Severity=3 name=resource\n
```

Instance

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-metrics CEF:0|Zscaler|NSSMe
tricslog|5.7|instance|%s{vm_name}|3| DeviceEventClassId=instance flexDate1=%s{time}
flexDate1Label=time deviceCustomDate1=%s{eventtime} deviceCustomDate1Label=eventtime
deviceCustomDate2=%d{inteventtime} deviceCustomDate2Label=inteventtime rt=%s{rtime}
dtz=%s{tz} rss=%02d{rss} rmm=%02d{rmm} rhh=%02d{rhh} rdd=%02d{rdd} rmth=%02d{rmth}
ryyyy=%04d{ryyyy} rday=%s{rday} rmon=%s{rmon} deviceExternalId=%s{recordid}
location=%s{location} group_name=%s{group_name} vm_name=%s{vm_name} platform_
name=%s{platform_name} platformgeo_name=%s{platformgeo_name} region_name=%s{region_
name} connector_type=%s{connector_type} name=%s{instance_name} cs1Label=icpu
cs1=%s{icpu} cs2Label=imem cs2=%s{imem} cs3Label=ithroughput_tx cs3=%s{ithroughput_
tx} cs4Label=ithroughput_rx cs4=%s{ithroughput_rx} cs5Label=ziathroughput_tx
cs5=%s{ziathroughput_tx} cs6Label=ziathroughput_rx cs6=%s{ziathroughput_rx}
flexString1Label=nodata flexString1=%s{nodata} Severity=3\n
```

©2026 Zscaler, Inc. All rights reserved.

VM

```
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-metrics CEF:0|Zscaler
|NSSMetricslog|5.7|vm|%s{vm_name}|3| DeviceEventClassId=vm flexDate1=%s{time}
flexDate1Label=time deviceCustomDate1=%s{eventtime} deviceCustomDate1Label=
eventtime deviceCustomDate2=%d{inteventtime} deviceCustomDate2Label=intev
enttime rt=%s{rtime} dtz=%s{tz} rss=%02d{rss} rmm=%02d{rmm} rhh=%02d{rhh}
rdd=%02d{rdd} rmth=%02d{rmth} ryyyy=%04d{ryyyy} rday=%s{rday} rmon=%s{rmon}
deviceExternalId=%s{recordid} location=%s{location} group_name=%s{group_name} vm_
name=%s{vm_name} platform_name=%s{platform_name} platformgeo_name=%s{platformgeo_
name} region_name=%s{region_name} connector_type=%s{connector_type}
cs1Label=syscpu cs1=%s{syscpu} cs2Label=rdiskUtil cs2=%s{cpu} cs3Label=rdiskUtil
cs3=%s{sendbytes} cs4Label=rdiskUtil cs4=%s{rcvdbytes} cs5Label=rdiskUtil
cs5=%s{totalmem} cs6Label=rdiskUtil cs6=%s{memutil} totthroughput=%s{totthroughput}
mgmtcpu=%s{mgmtcpu} mgmtmem=%s{mgmtmem} flexString1Label=nodata flexString1=%s{nodata}
Severity=3\n
```

Activate Changes

The last step on the Zscaler side is to activate the configuration. All configurations up to this point are candidate configurations. The activated changes become active in production.

Go to **Activation** from the left-side navigation, and then click **Activate** to commit your changes.

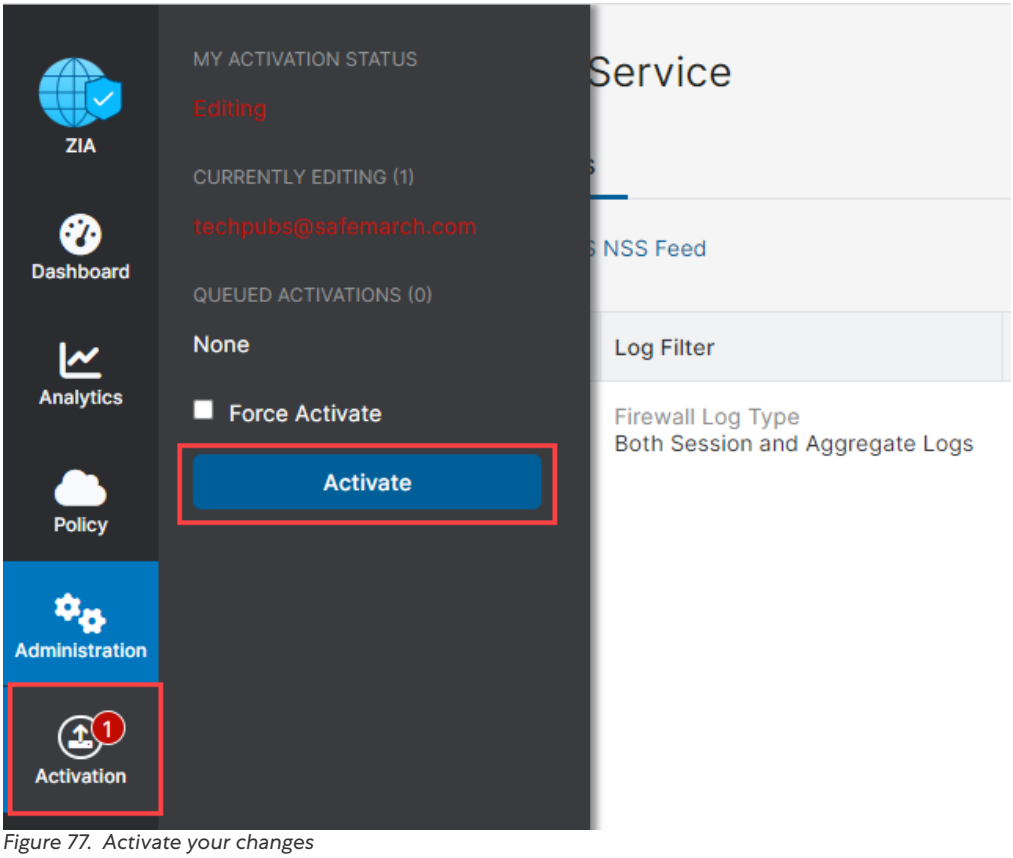


Figure 77. Activate your changes

©2026 Zscaler, Inc. All rights reserved.

Create a Log Analytics Workspace, and Add Microsoft Sentinel to the Workspace

To create a log analytics workspace:

1. Log in to the Azure portal.
2. Click the **Microsoft Sentinel** tile.

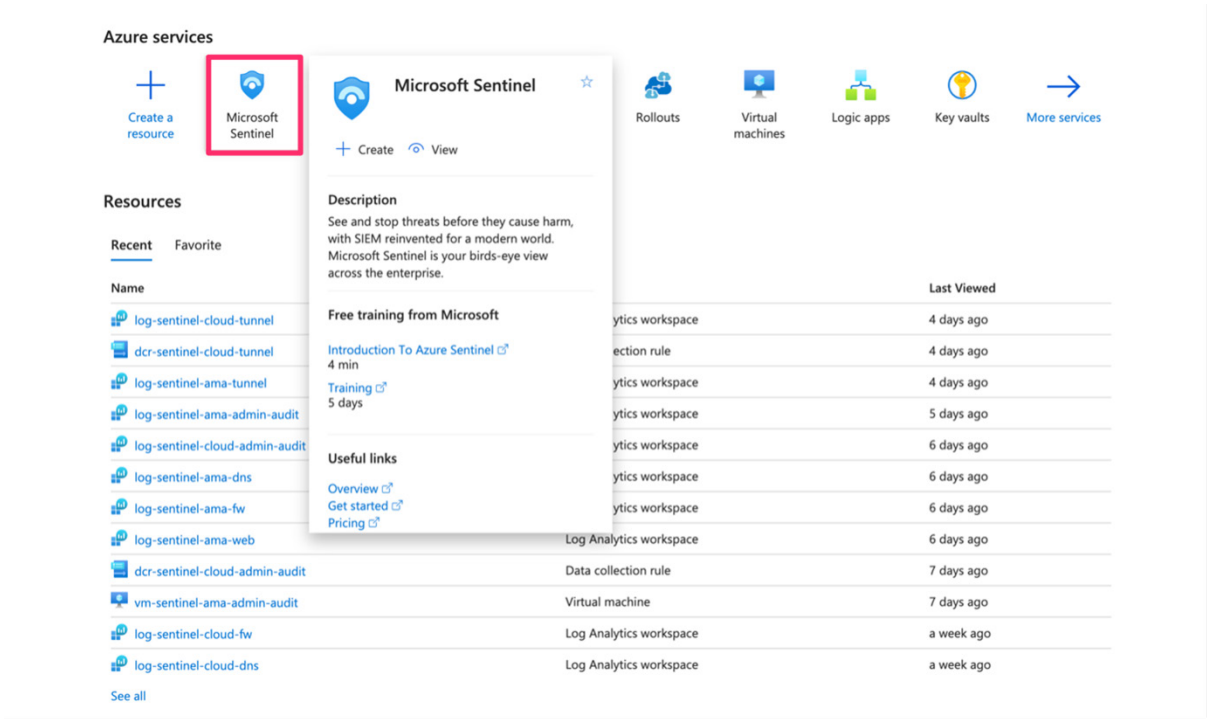


Figure 78. Click the Microsoft Sentinel tile

3. Click **Create**.

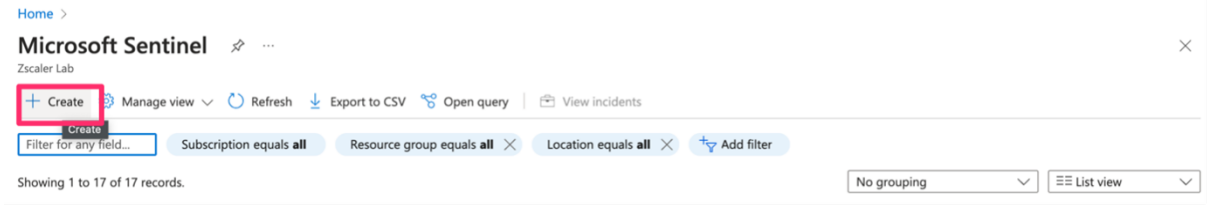


Figure 79. Click Create

4. Click **Create a new workspace**.

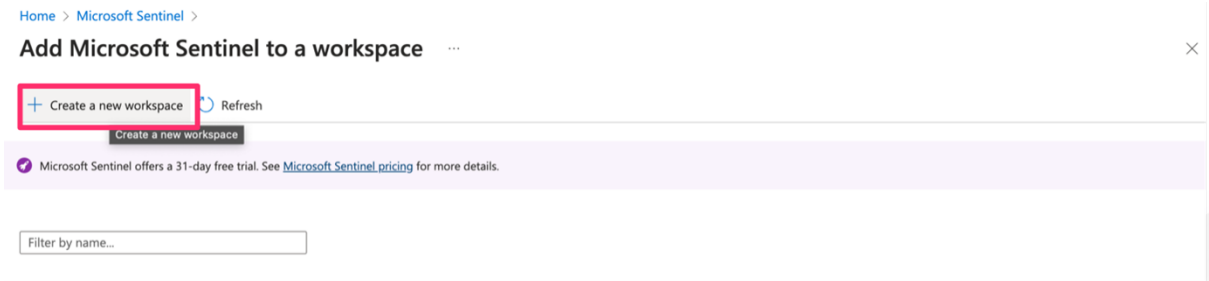


Figure 80. Add Microsoft Sentinel to a workspace

©2026 Zscaler, Inc. All rights reserved.

5. Create a new resource group. Click **Create new**.
6. Enter a **Name**.
7. Click **OK**.

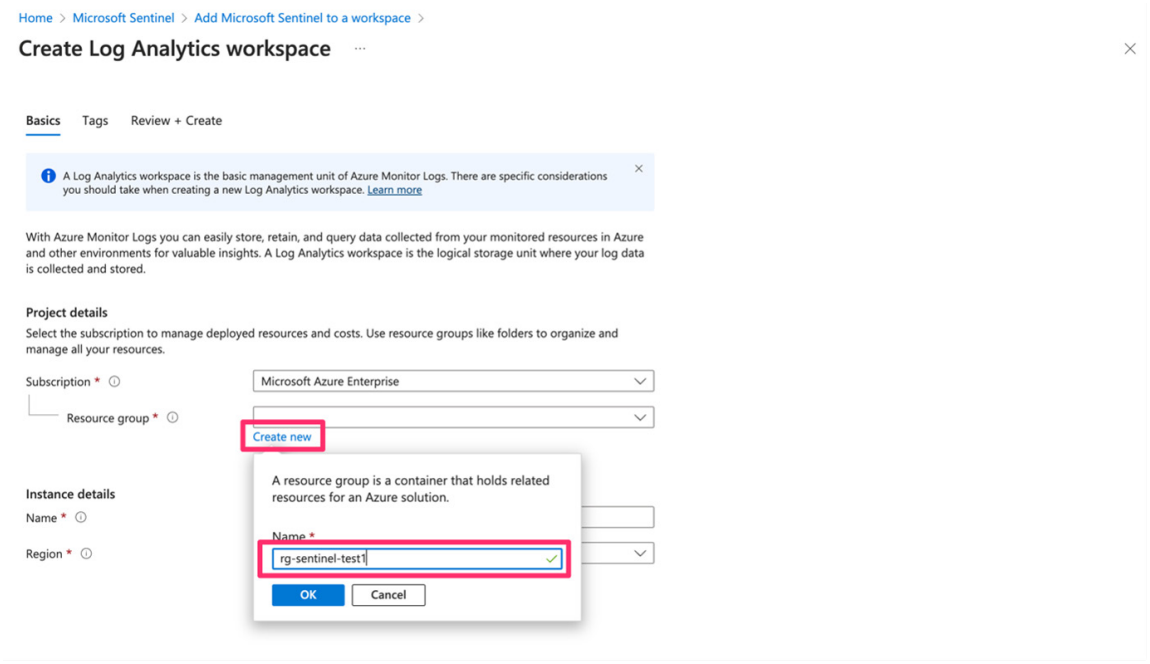


Figure 81. Configure workspace

8. Enter a workspace name, then select a **Region**.
9. Click **Review + Create**.

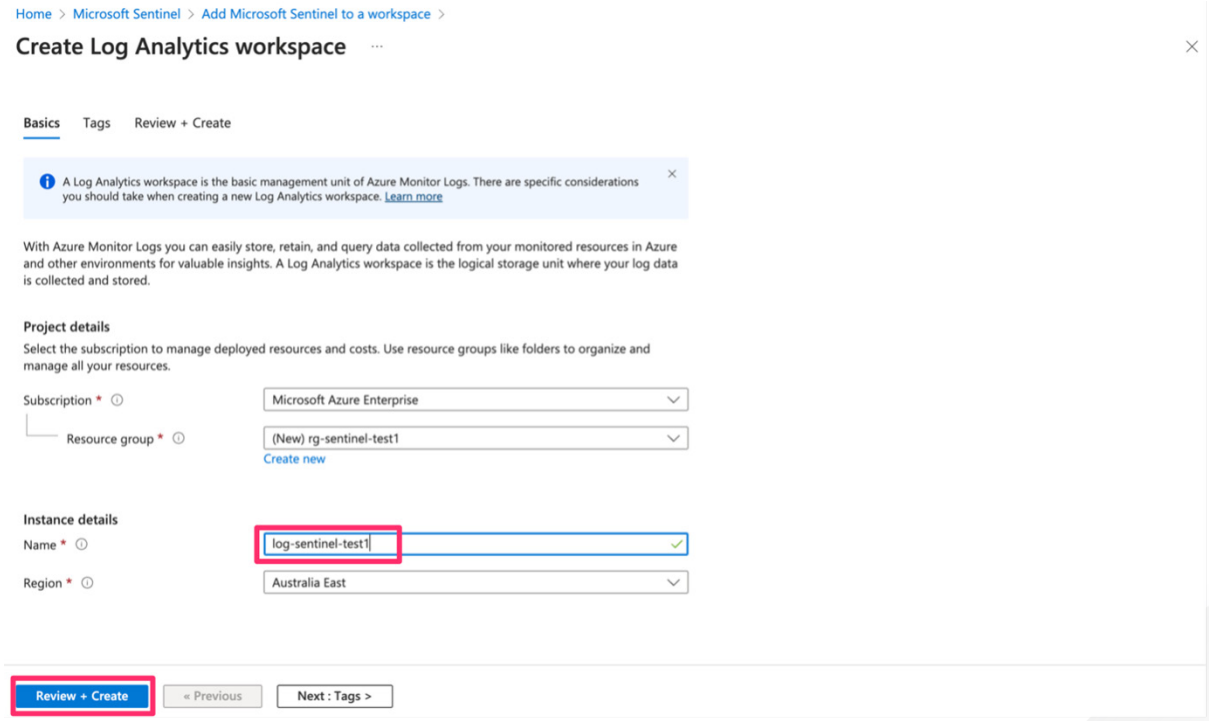


Figure 82. Review + Create

©2026 Zscaler, Inc. All rights reserved.

10. Wait a moment for the workspace to be created, then click **Refresh**, and **Add**.

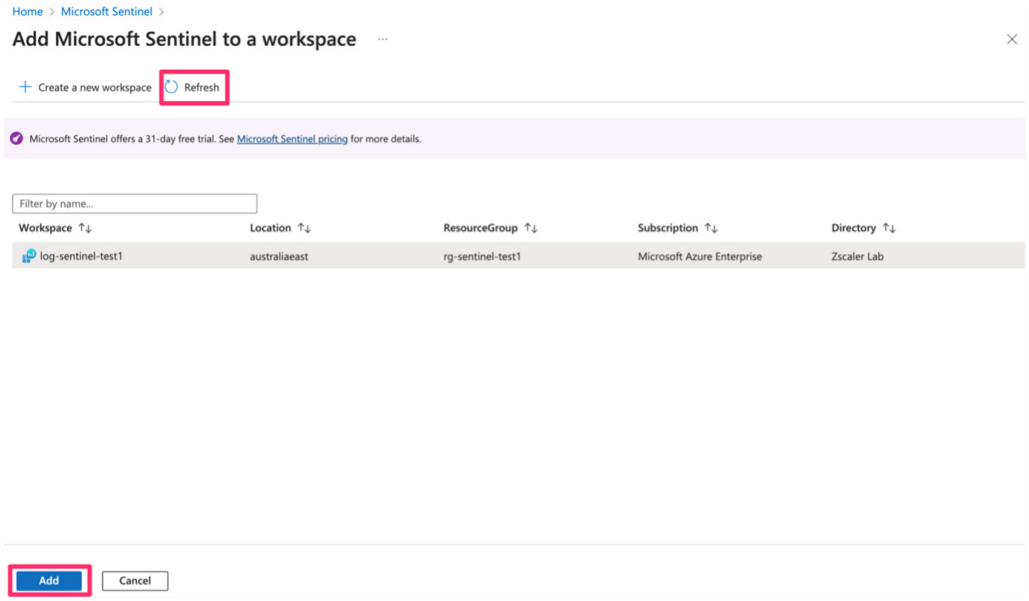


Figure 83. Refresh

You are redirected to the Microsoft Sentinel | News & guides page and a success message appears.

Install the Common Event Format Content Pack

To install the CEF content pack:

1. Go to Microsoft Sentinel.
2. Select your workspace, click **Content hub**, then search for `CEF`.

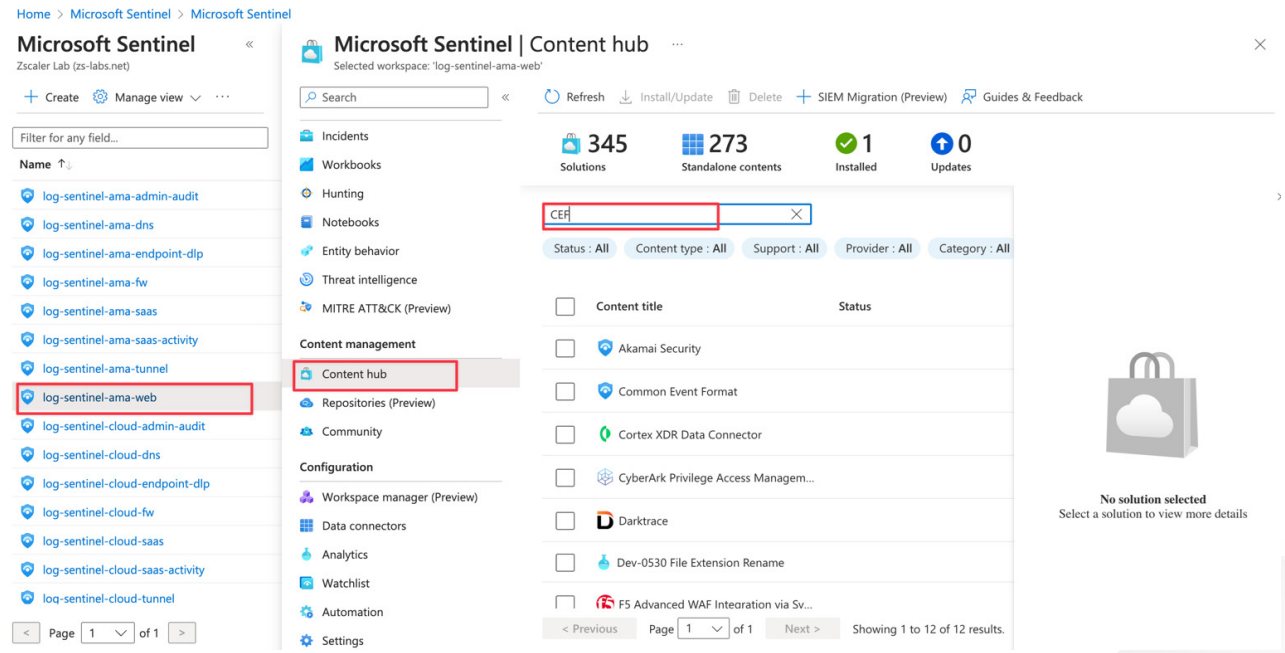


Figure 84. Content hub

©2026 Zscaler, Inc. All rights reserved.

3. Click **Common Event Format**, then click **Install**.

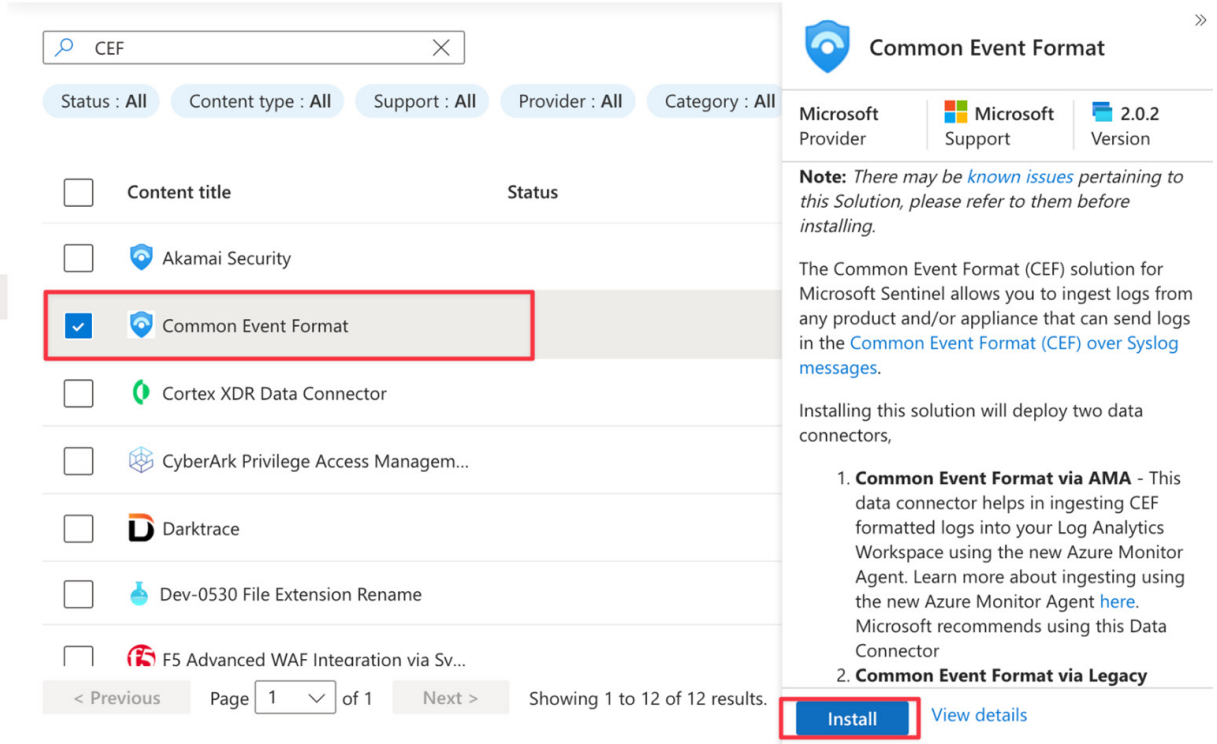


Figure 85. Common Event Format

4. When installed, select the **Common Event Format** content pack and click **Manage**.

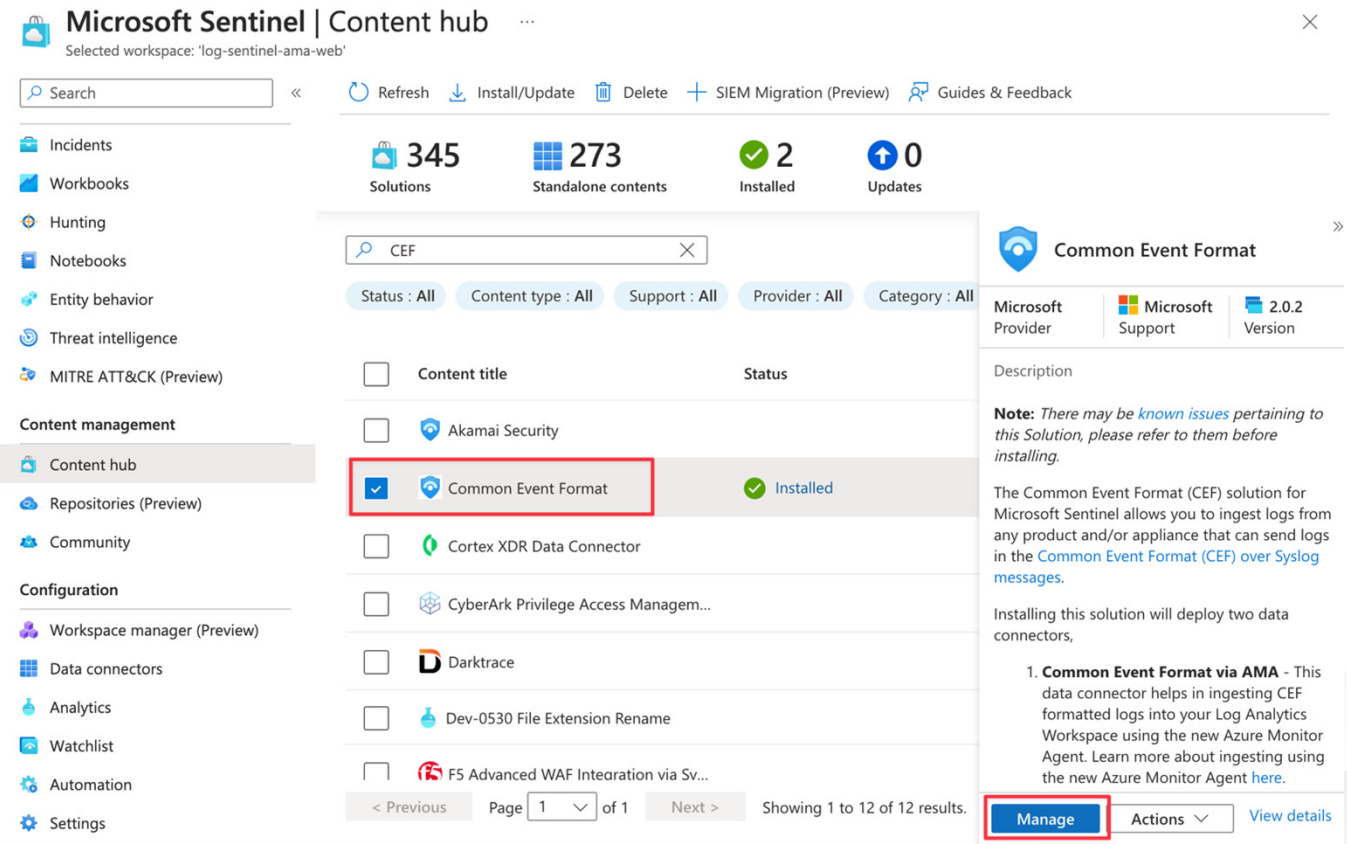


Figure 86. Manage Common Event Format

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Common Event Format (CEF) via AMA** content name.
6. Click **Open connector page**.

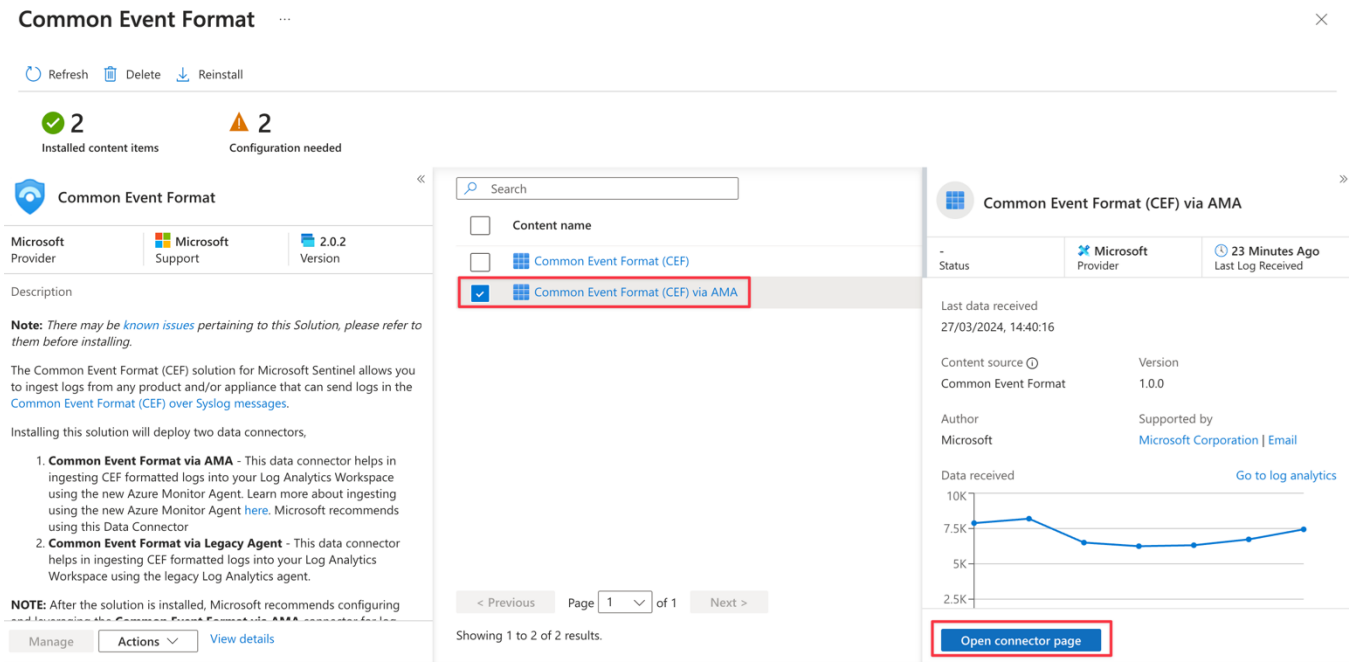


Figure 87. Common Event Format (CEF) via AMA

7. Click **Create data collection rule**.

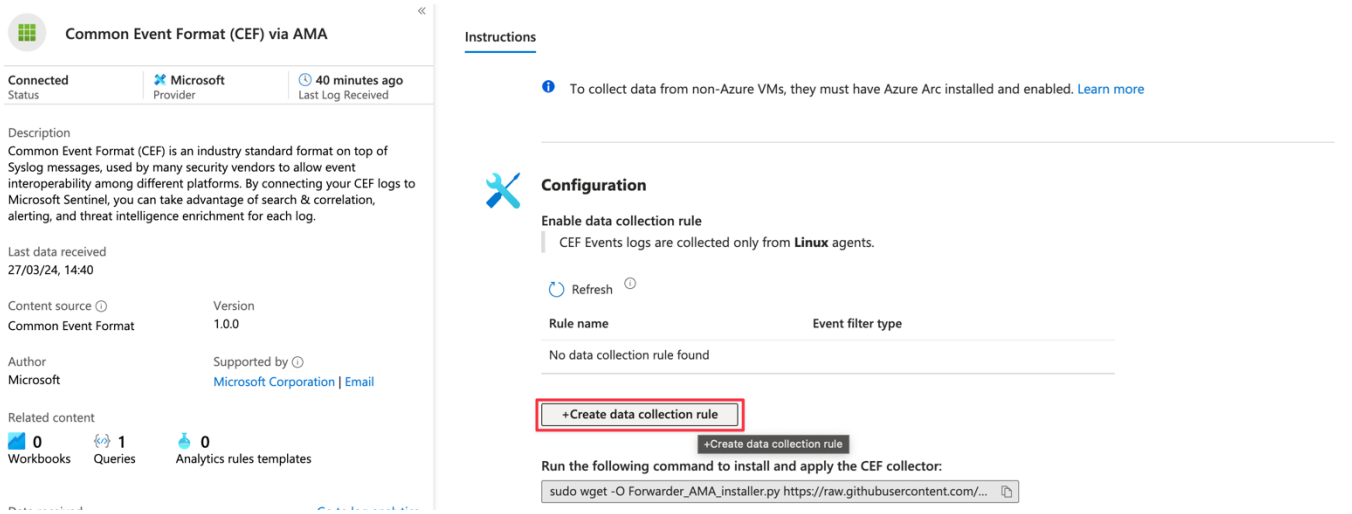


Figure 88. Data collection rule

©2026 Zscaler, Inc. All rights reserved.

8. Enter a **Rule Name** and click **Next: Resources**.

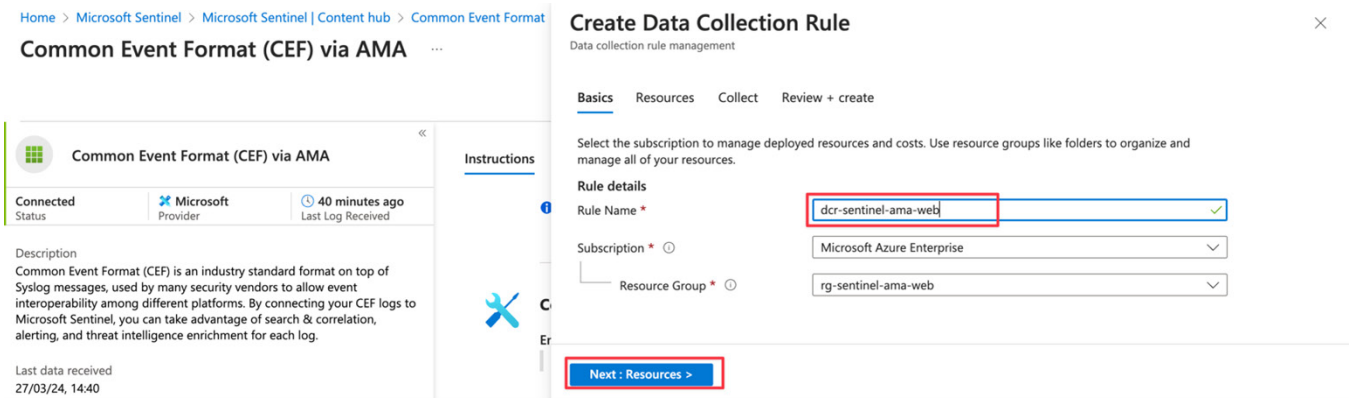


Figure 89. Name data collection rule

9. Click **Add resource(s)** and then click **Next: Collect**.

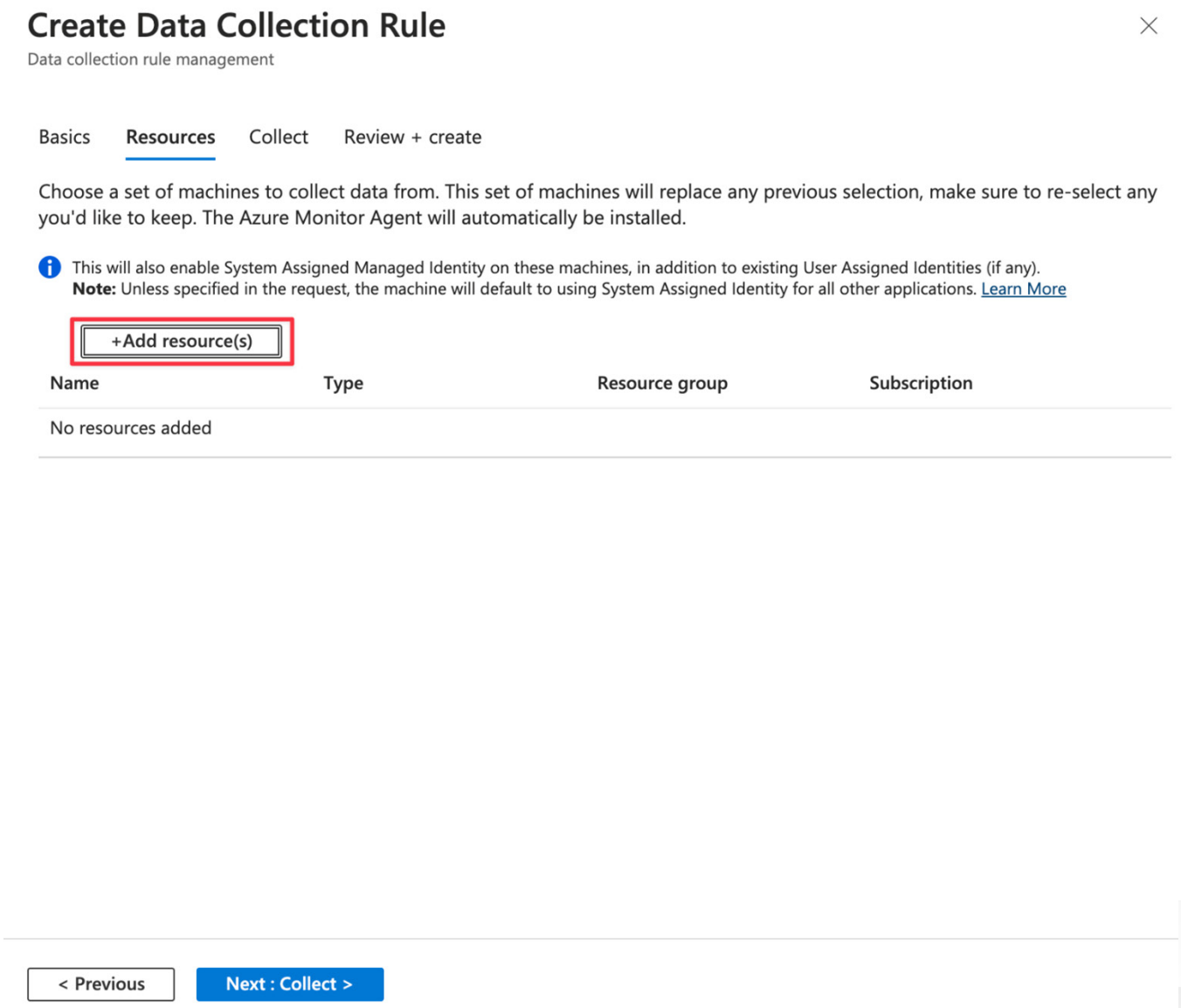


Figure 90. Add resource(s)

©2026 Zscaler, Inc. All rights reserved.

10. Select the VM you created earlier and click **Apply**.

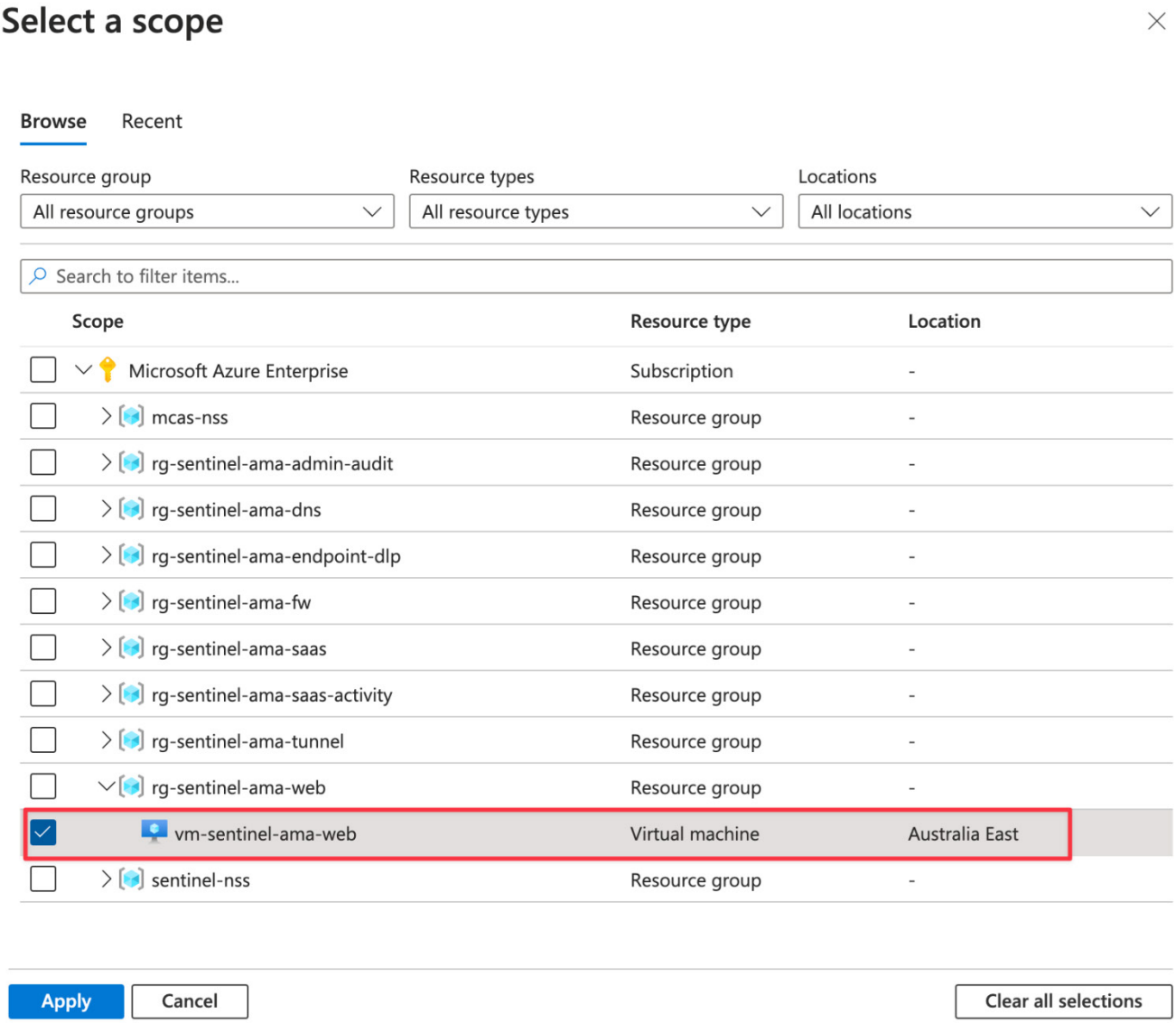


Figure 91. Select a scope

©2026 Zscaler, Inc. All rights reserved.

11. Click **Next: Collect**.

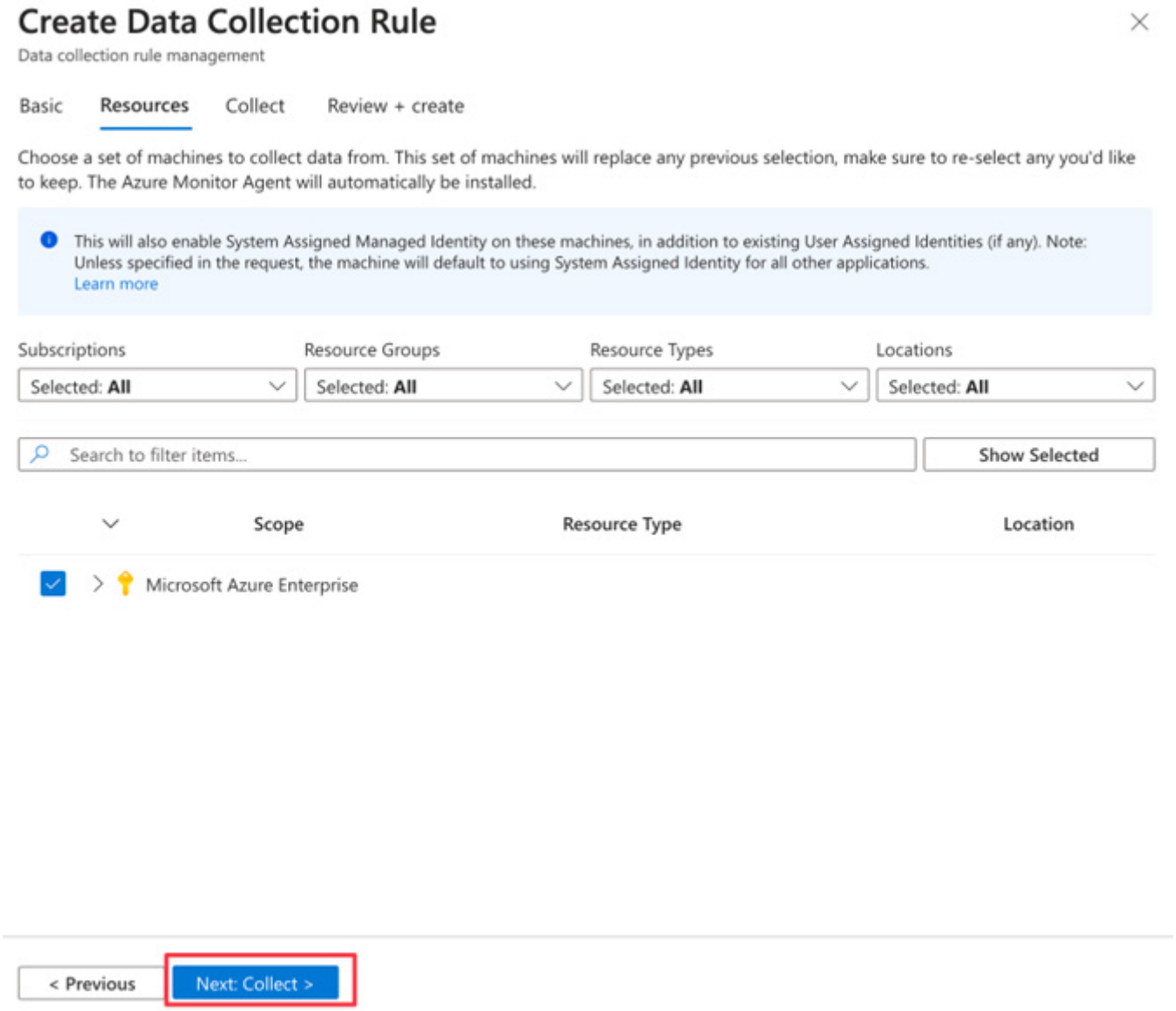


Figure 92. Collect

12. Clear the **Collect messages without PRI header (facility and severity)** checkbox.

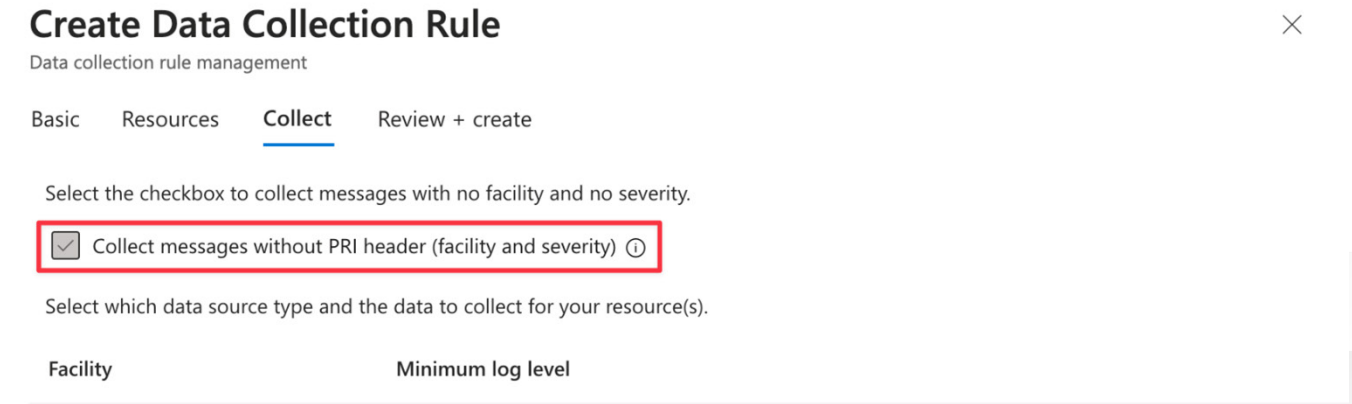


Figure 93. Collect messages with PRI

©2026 Zscaler, Inc. All rights reserved.

13. For the **Facility**, ensure **LOG_USER** has a minimum log level of **LOG_DEBUG**, and then click **Next: Review + create**.

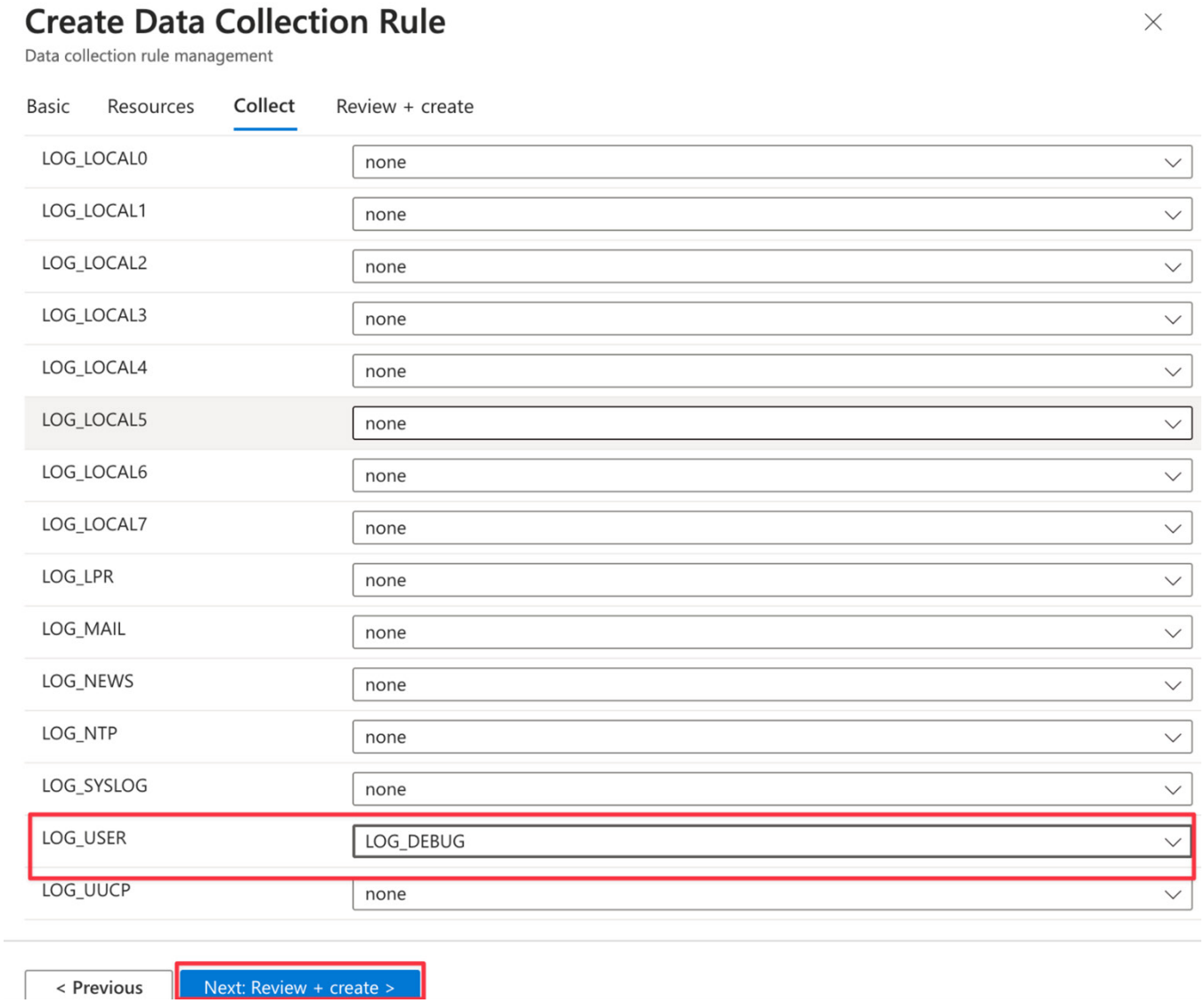


Figure 94. Create Data Collection Rule

©2026 Zscaler, Inc. All rights reserved.

14. Click **Create**. After it is created, you can see the **Data Collection Rule** listed under **Configuration**.

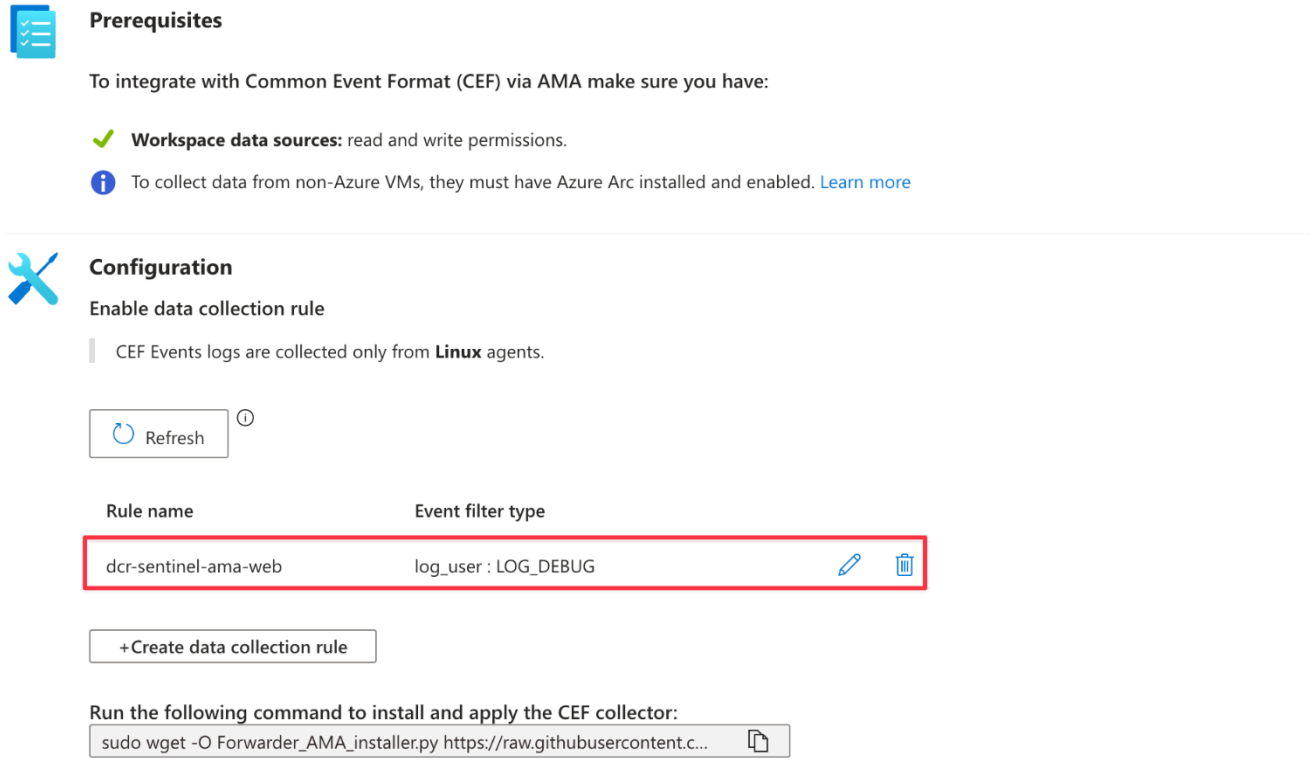


Figure 95. Data collection rule listed

©2026 Zscaler, Inc. All rights reserved.

Configure the VM Connector to Receive Logs

This guide uses Ubuntu as a base image, and rsyslog as the syslog daemon.

Log in to your VM, and run the following commands:

1. Install the latest stable build rsyslog using the following commands:

   ```
   sudo add-apt-repository ppa:adiscon/v8-stable

   sudo apt-get update

   sudo apt-get install rsyslog
   ```

2. Install python-is-python3 to run the Microsoft provided script for log configuration and ingestion:

   ```
   sudo apt install python-is-python3
   ```

3. Run the Microsoft-provided command in the Common Event Format via AMA connector page to enable the CEF collector.

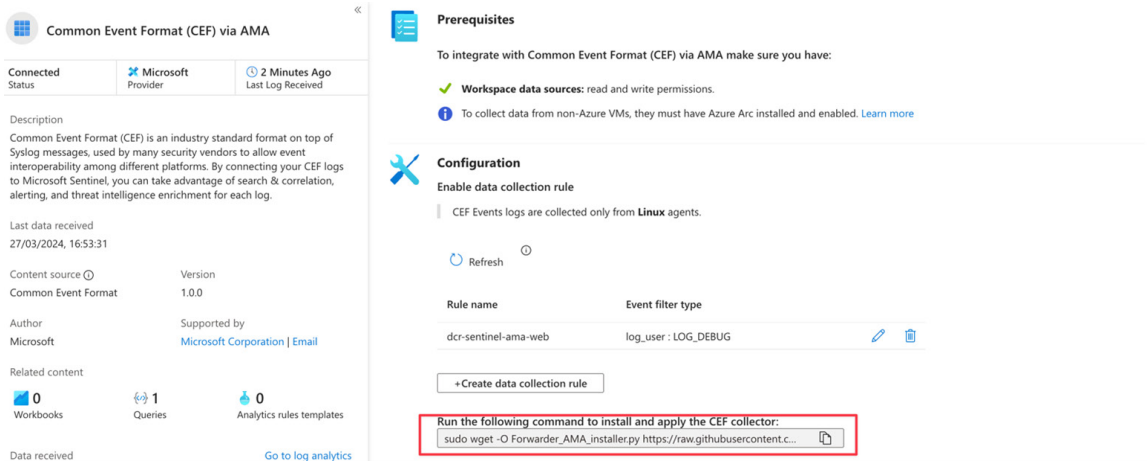


Figure 96. CEF collector

In the next 10 minutes, logs are populated in your CommonSecurityLog table.

©2026 Zscaler, Inc. All rights reserved.

View Log Details

To view the ingested logs:

1. In Microsoft Sentinel, select your Log Analytics workspace (e.g., `log-sentinel-test1`).
2. In the left-side navigation, select **Logs** and either enter `CommonSecurityLog` into the query window or wait until the Queries History populates with logs.
3. Click **Run**.

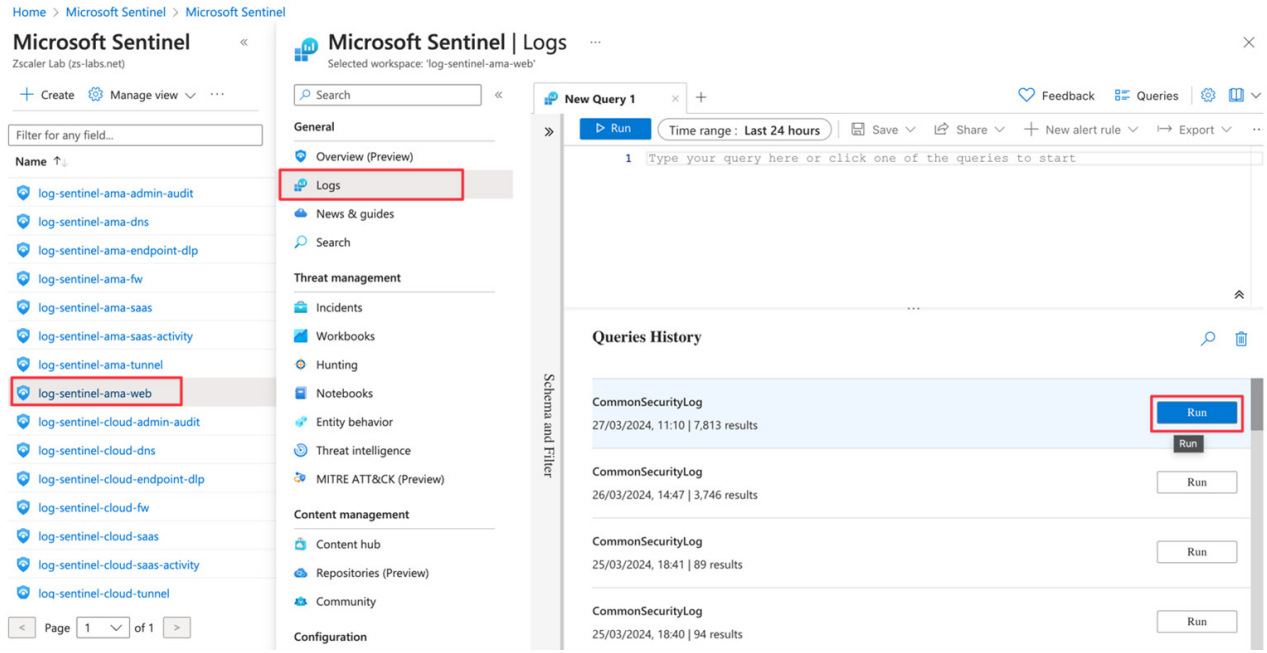


Figure 97. Queries history

Query Results are displayed after a few minutes.

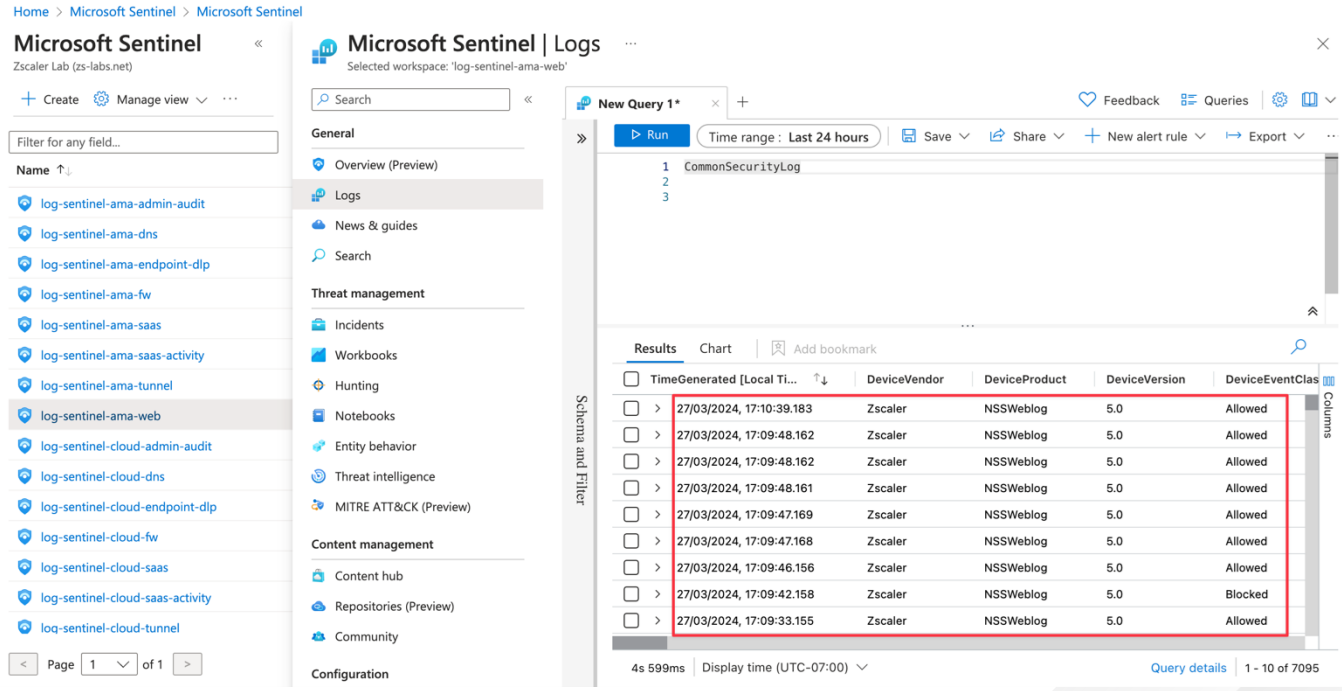


Figure 98. Log query results

©2026 Zscaler, Inc. All rights reserved.

Add the ZIA Content Pack to Install ZIA Workbooks, Analytics Rules, and Parsers

Complete the following instructions:

1. In Microsoft Sentinel, select your Log Analytics workspace.
2. In the left-side navigation, click **Content hub**, and search for `ZIA` or `Zscaler`.

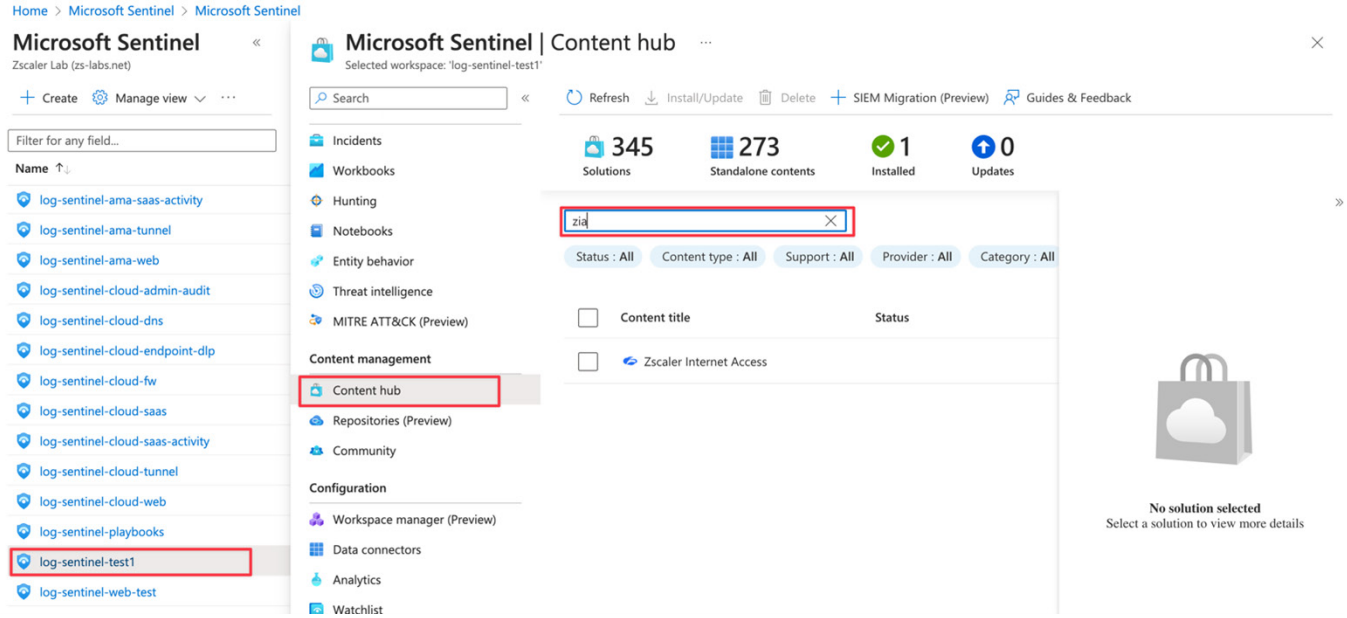


Figure 99. Content hub

©2026 Zscaler, Inc. All rights reserved.

3. Click the **Zscaler Internet Access** content pack and click **Install**.

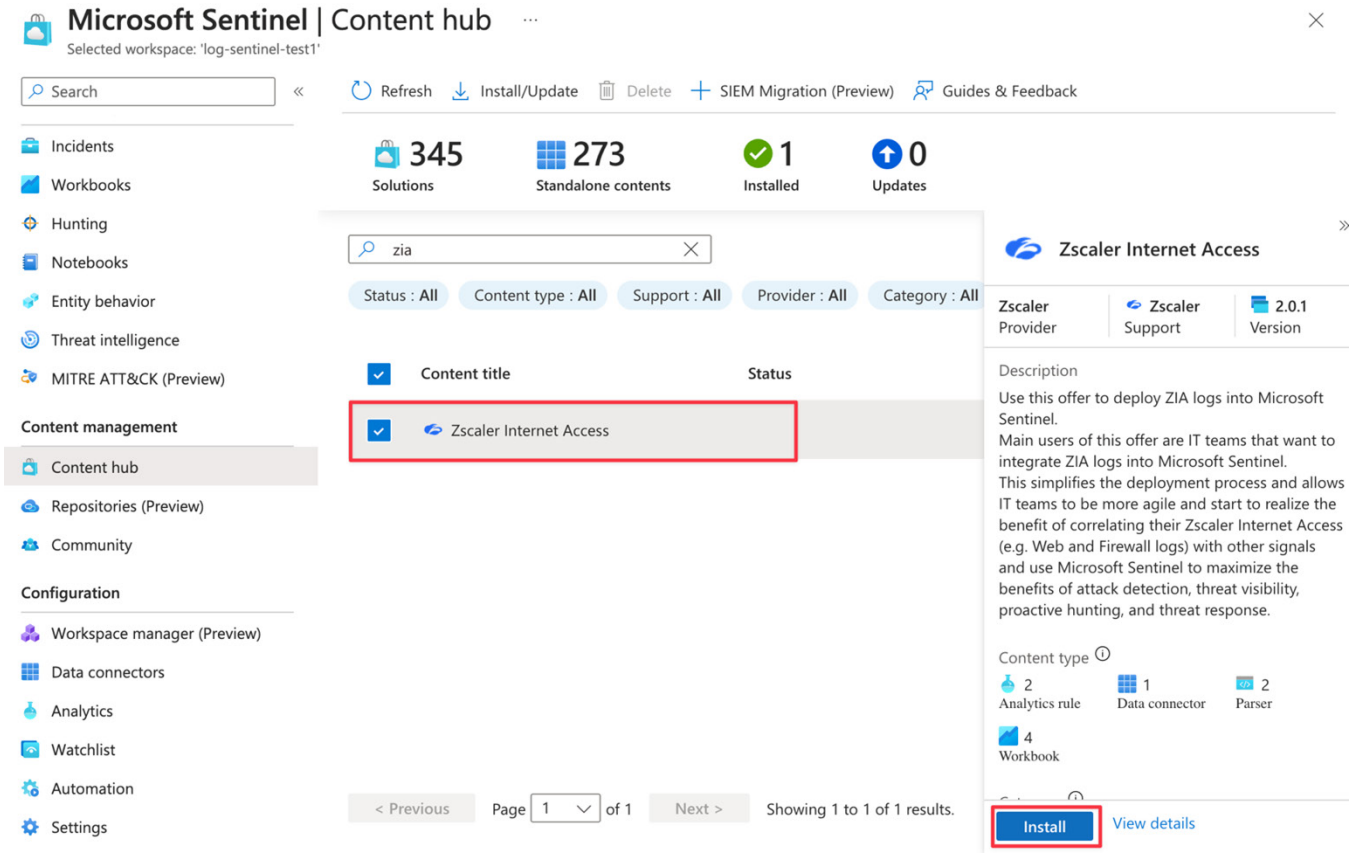


Figure 100. Zscaler Internet Access

When installed, the content pack shows a **Status** of **Installed**.

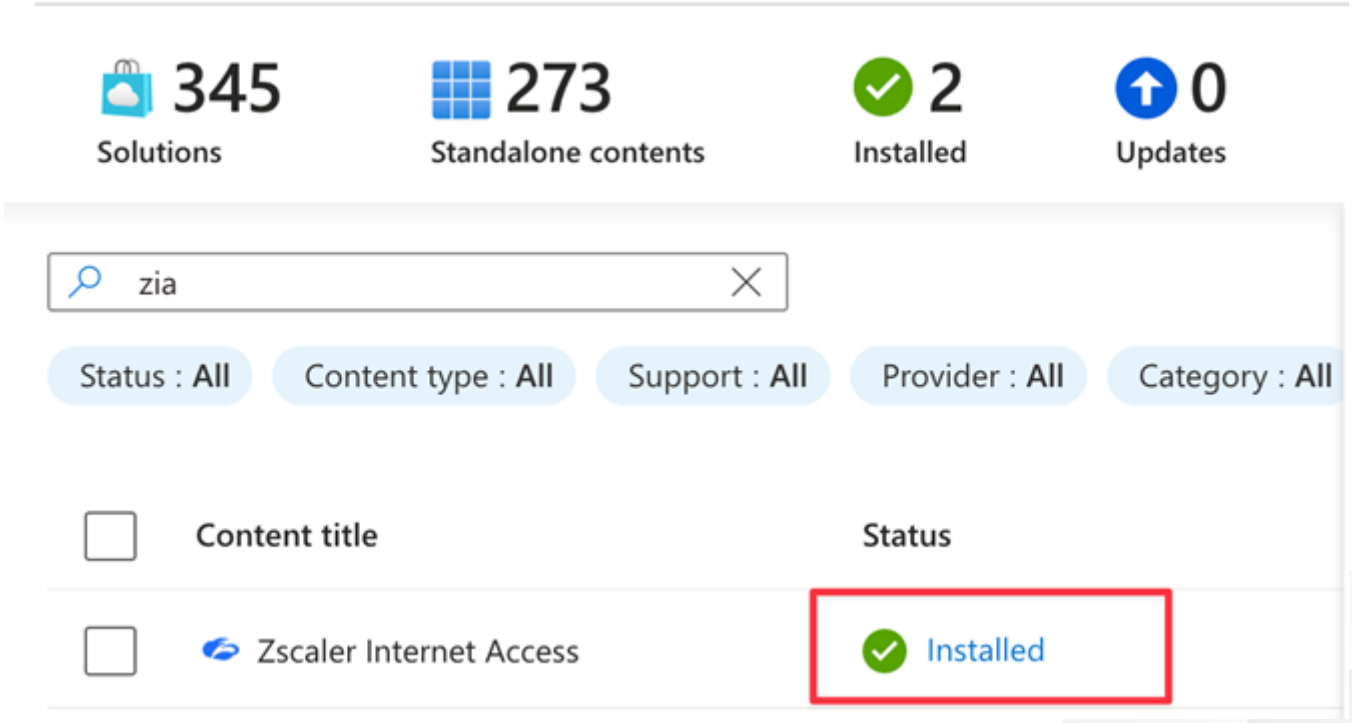


Figure 101. Content pack status

©2026 Zscaler, Inc. All rights reserved.

4. Click **Workbooks** and then **Templates**.

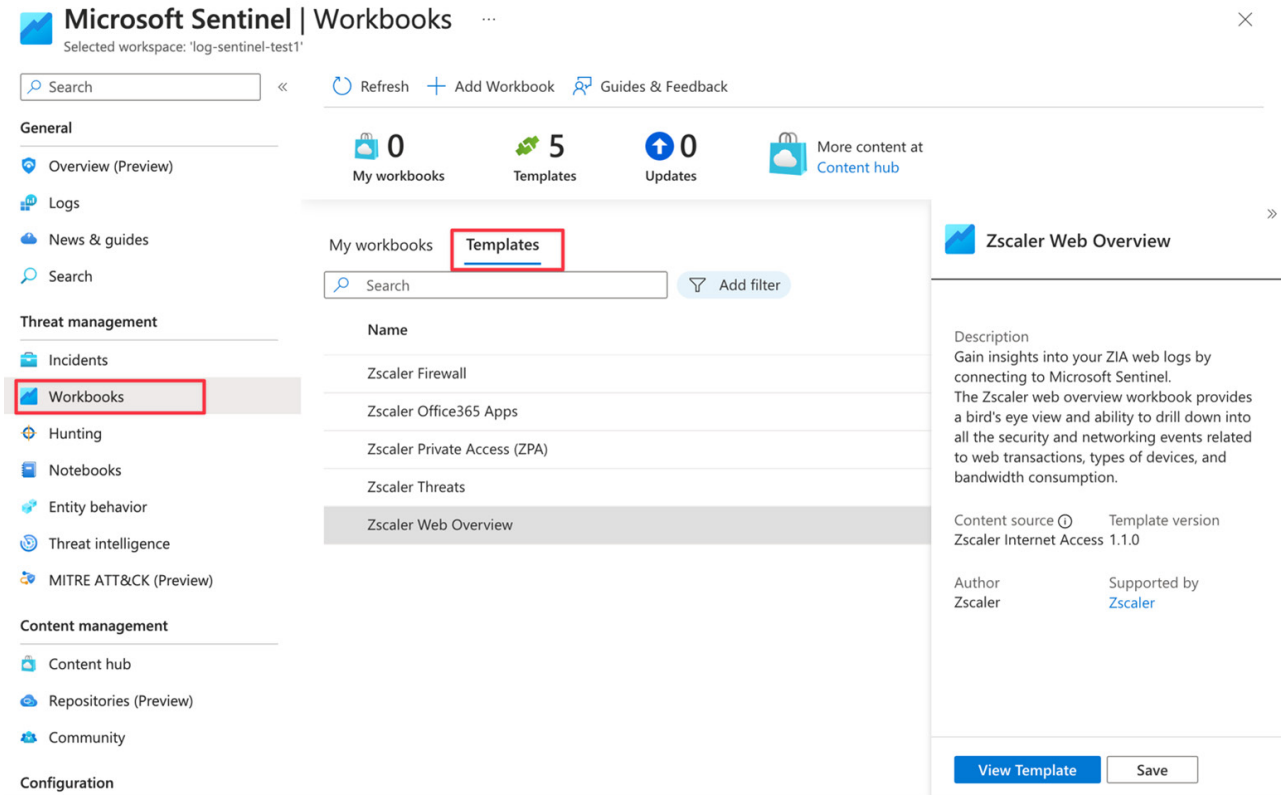


Figure 102. Workbooks and templates

5. For each workbook you want to install, click the name of the workbook, then **Save**.

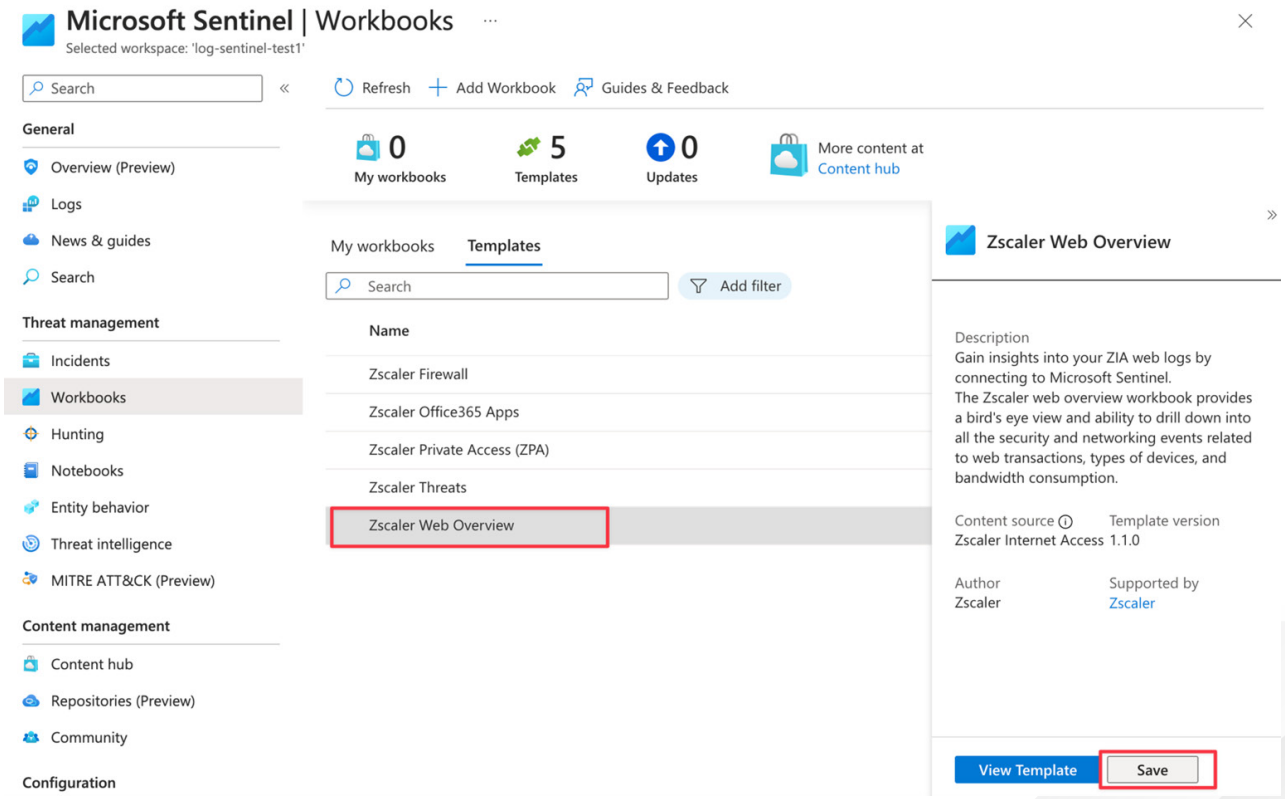


Figure 103. Save workbooks

©2026 Zscaler, Inc. All rights reserved.

6. Select the region in which to save your workbook and click **Yes**. The saved workbooks appear under **My workbooks**.

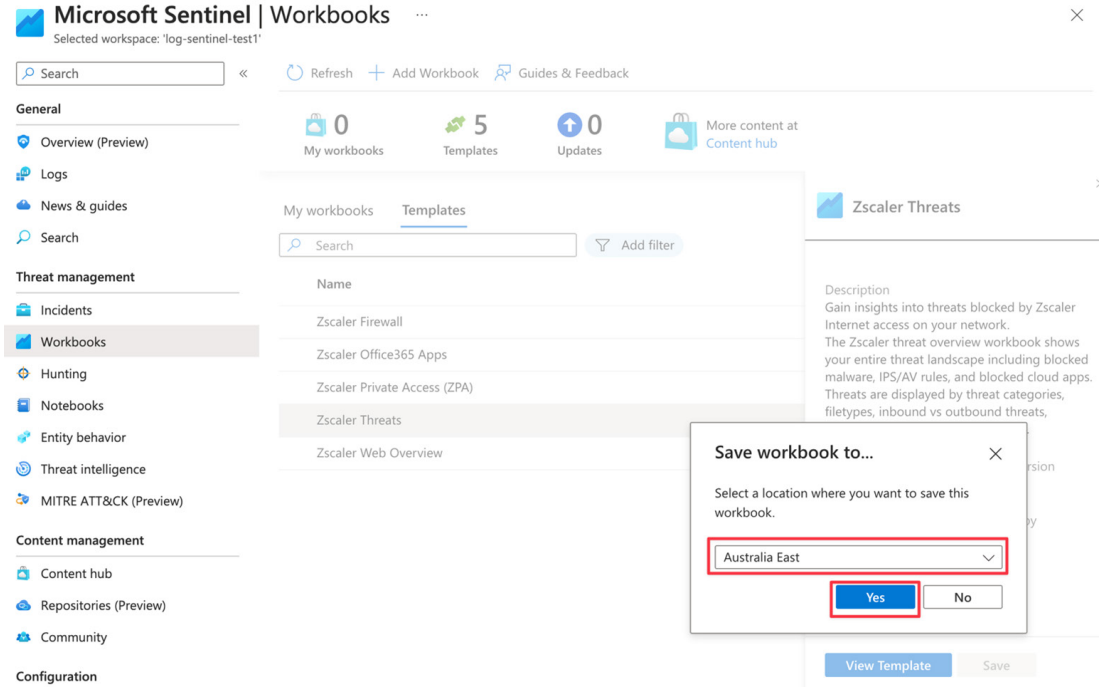


Figure 104. Workbook region

7. Click **View saved workbook** to view the workbook.

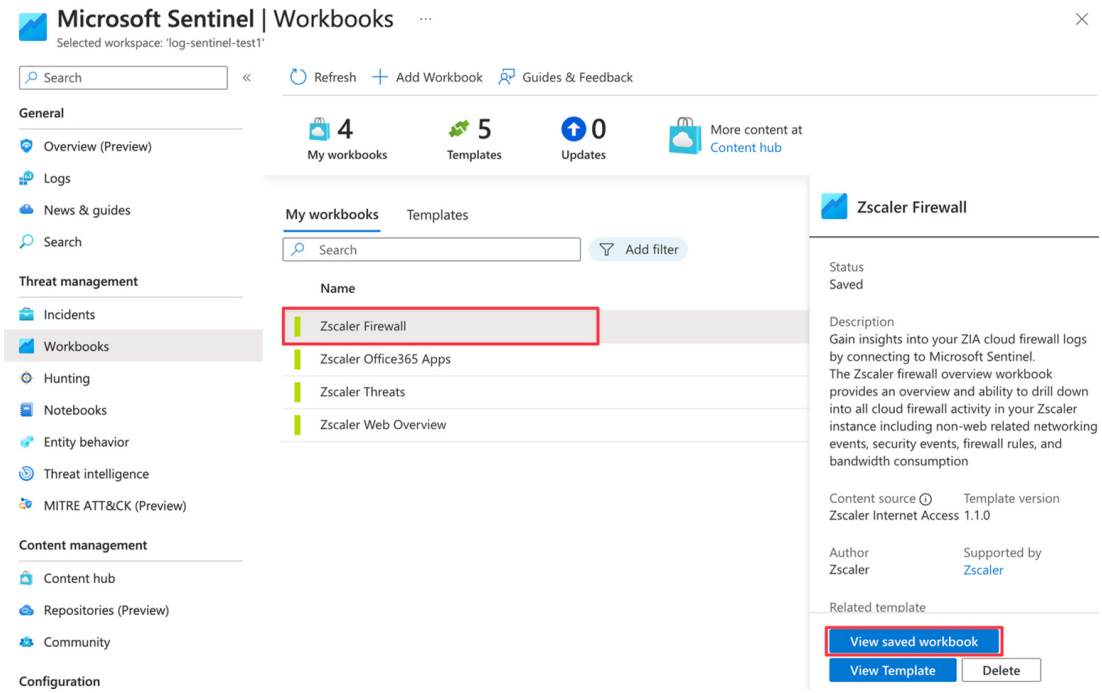


Figure 105. Saved workbooks

©2026 Zscaler, Inc. All rights reserved.

Examples of the Workbooks

The following are examples of the workbooks.

The following image is an example of the Zscaler Web Overview workbook.

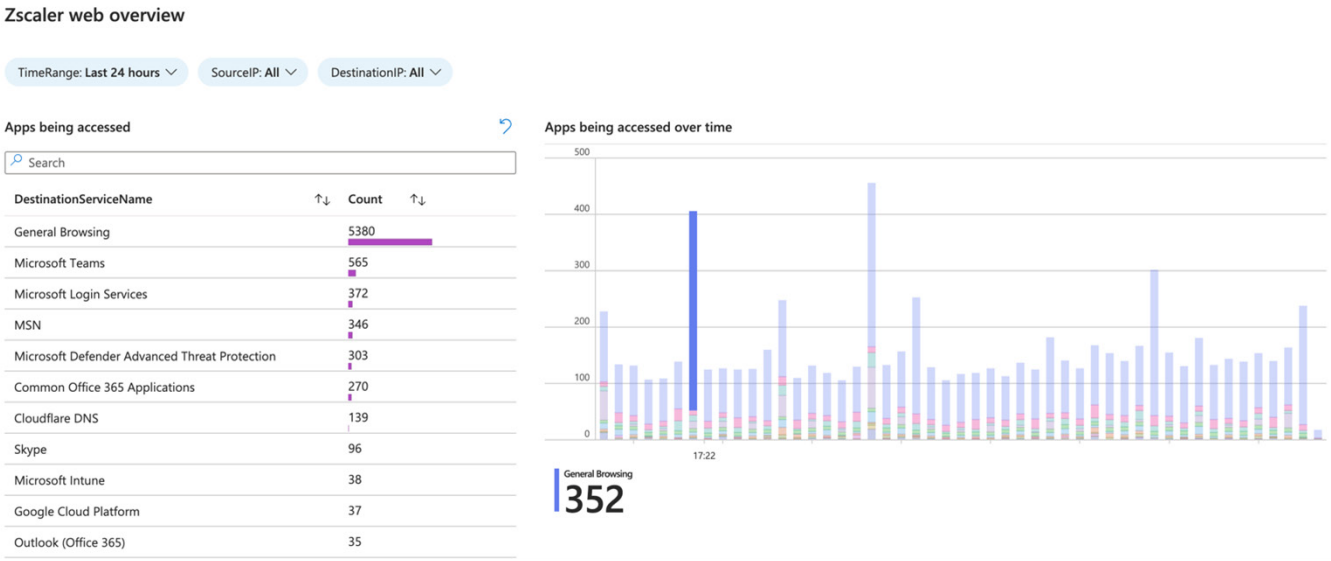


Figure 106. Zscaler Web Overview

The following image is an example of the Zscaler Threats Overview workbook.

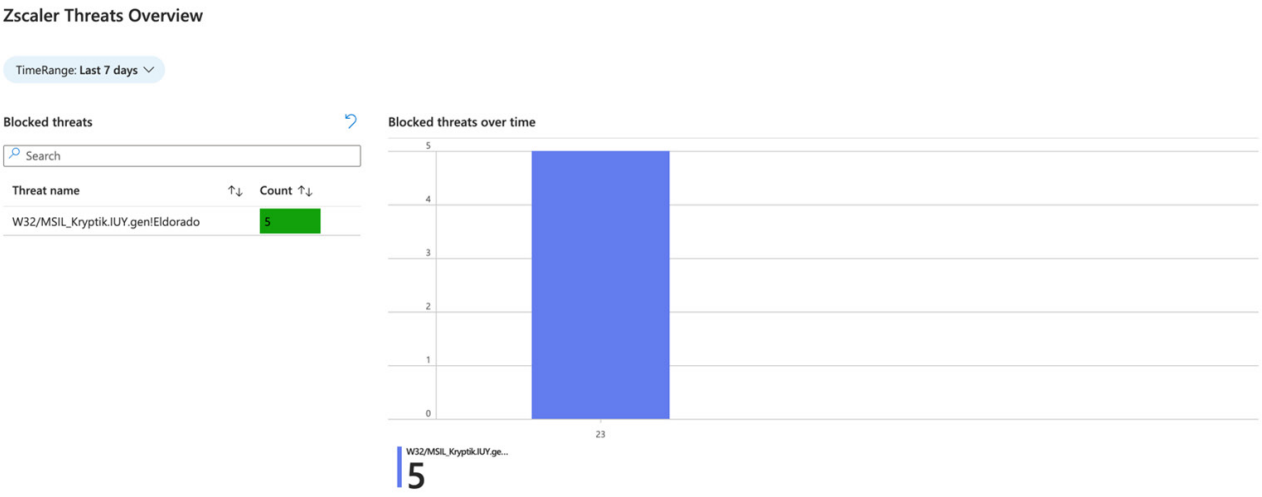


Figure 107. Zscaler Threats Overview

©2026 Zscaler, Inc. All rights reserved.

The following image is an example of the Zscaler Microsoft 365 Overview workbook.

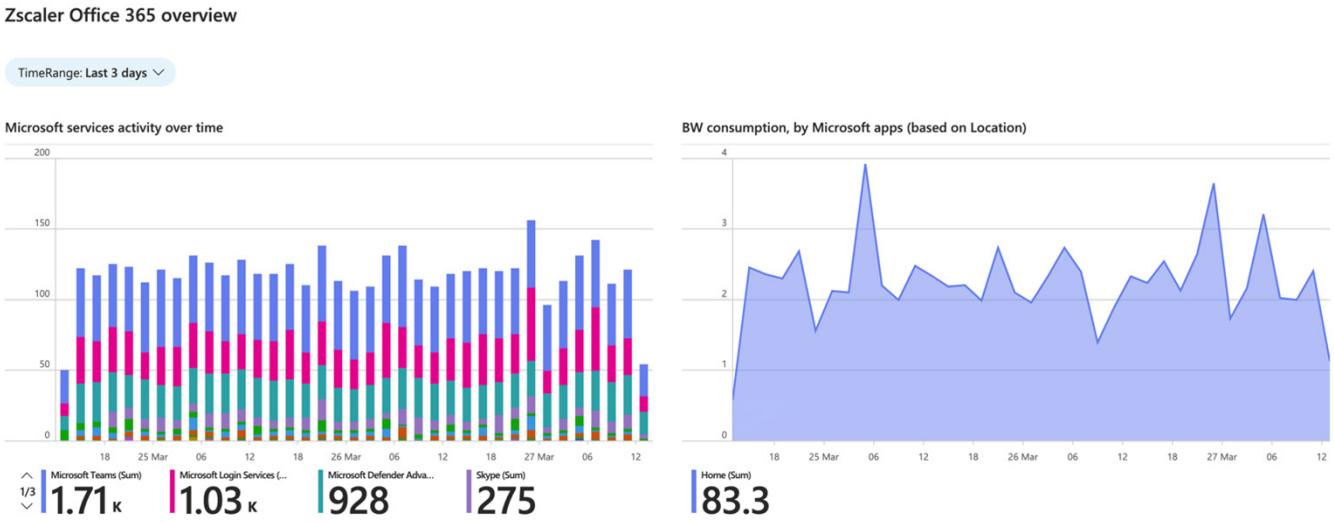


Figure 108. Zscaler Office 365 overview

The following image is an example of the Zscaler Firewall Overview workbook.

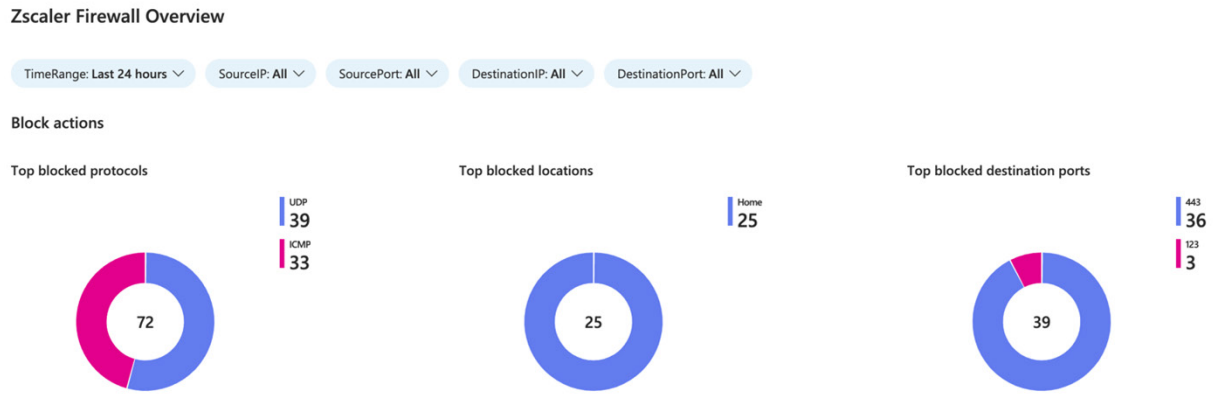


Figure 109. Zscaler Firewall Overview

©2026 Zscaler, Inc. All rights reserved.

Configuring Log Ingestion for ZPA

The following steps assume that you have Admin access to the Azure portal.

Logging In to Azure Portal

Log in to the Azure portal.

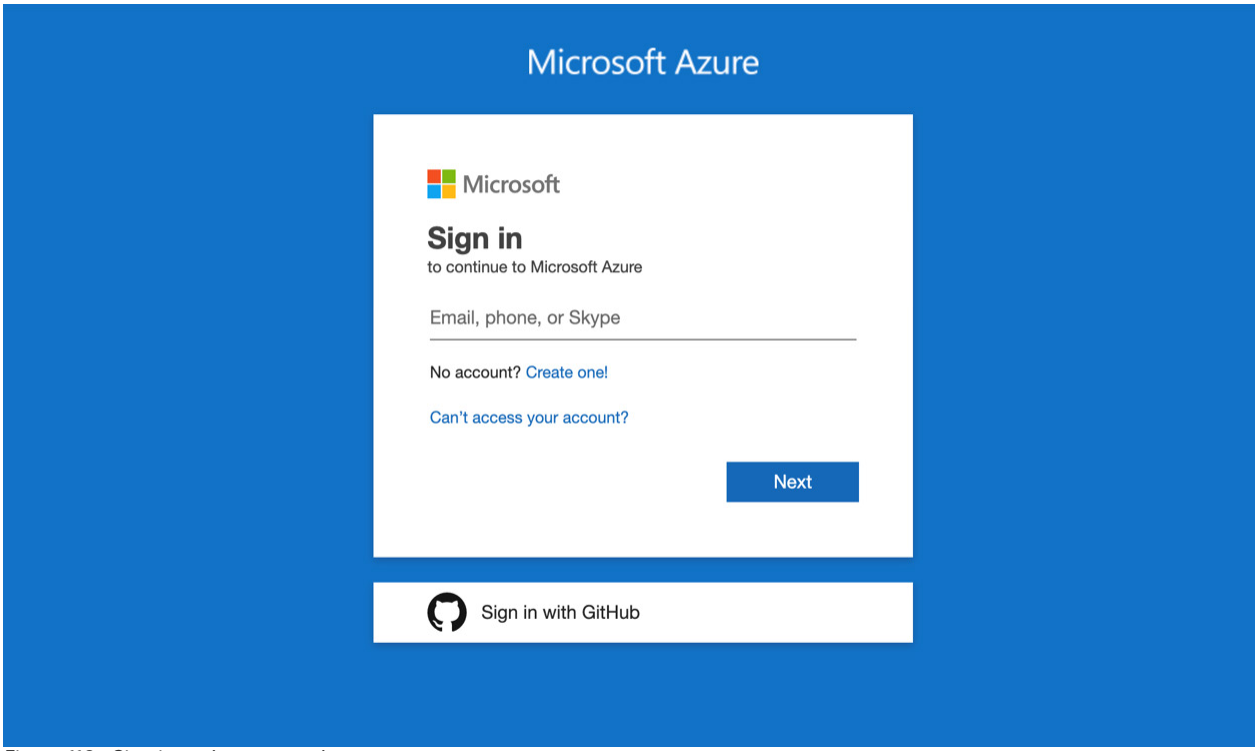


Figure 110. Sign in to Azure portal

©2026 Zscaler, Inc. All rights reserved.

Deploying the Data Connector Host VM

The first Microsoft Sentinel step is to deploy a Linux VM. This Linux VM is the Sentinel ZPA data connector. In the ZPA Admin Portal, you configure ZPA to stream logs to the IP address of the Linux VM. The VM then forwards received log messages to the Microsoft Sentinel backend.

After navigating to the **Home** window, click **Virtual machines**.

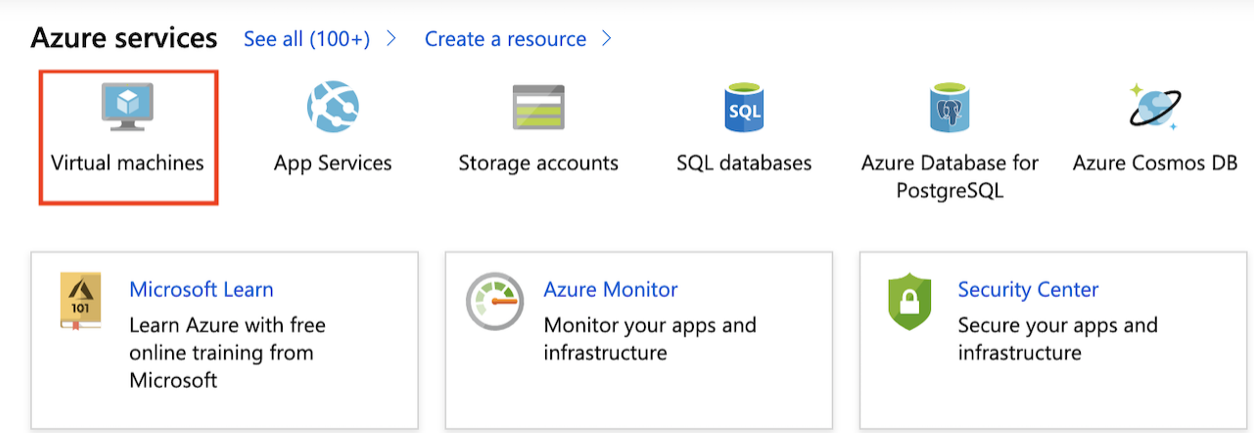


Figure 111. This VM is a Syslog server and runs Azure's data connector

Create Virtual Machine

Your **Virtual machines** window might have additional information displayed if you have existing virtual machines. Click **Create virtual machine**.

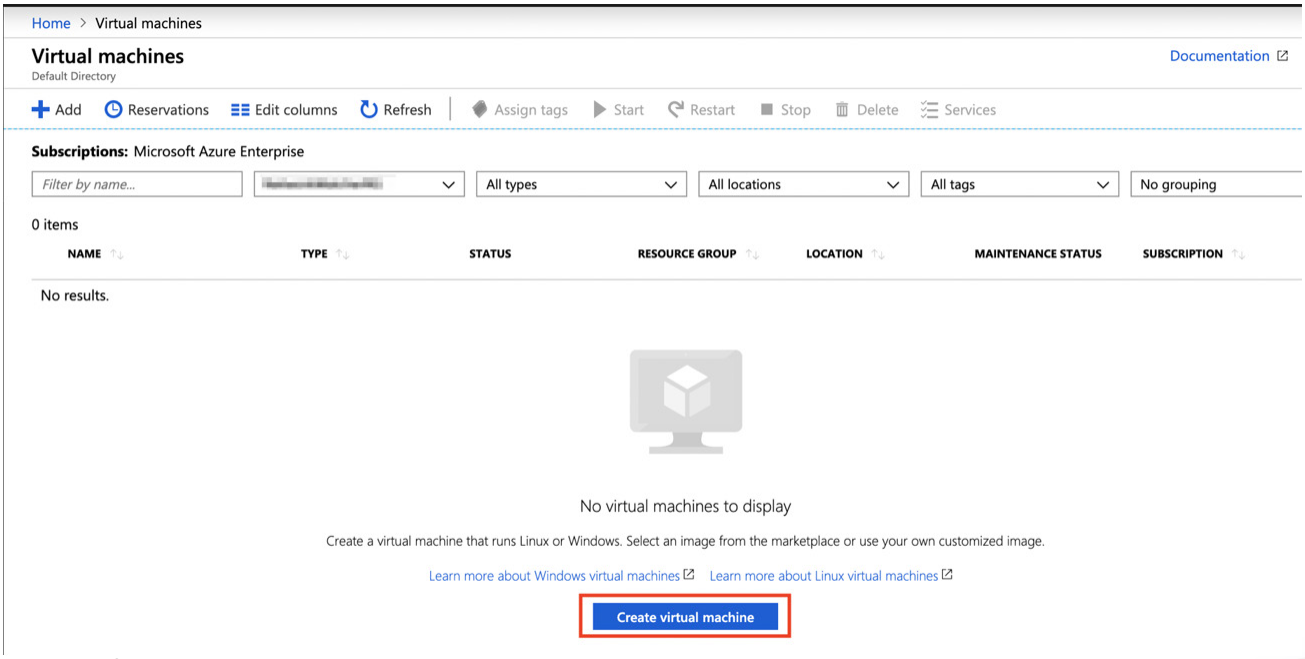


Figure 112. Create virtual machine

©2026 Zscaler, Inc. All rights reserved.

Bind this VM to a Resource Group

A Resource Group (RG) is a way to group a collection of assets in logical containers for automatic provisioning, monitoring, access control, and more effective management of their costs. One benefit of using RGs is to group related application resources together, as they share a unified lifecycle from creation, usage, and deprovisioning.

You can bind to an existing resource group, or you can create a new one.

1. To create a new resource group, click **Create new**, and enter a Virtual machine name. The following example uses `Ubuntu Server 18.04 LTS`.Fill in the pertinent details, then click **Review + create**.
2. Follow subsequent prompts to finish VM creation.

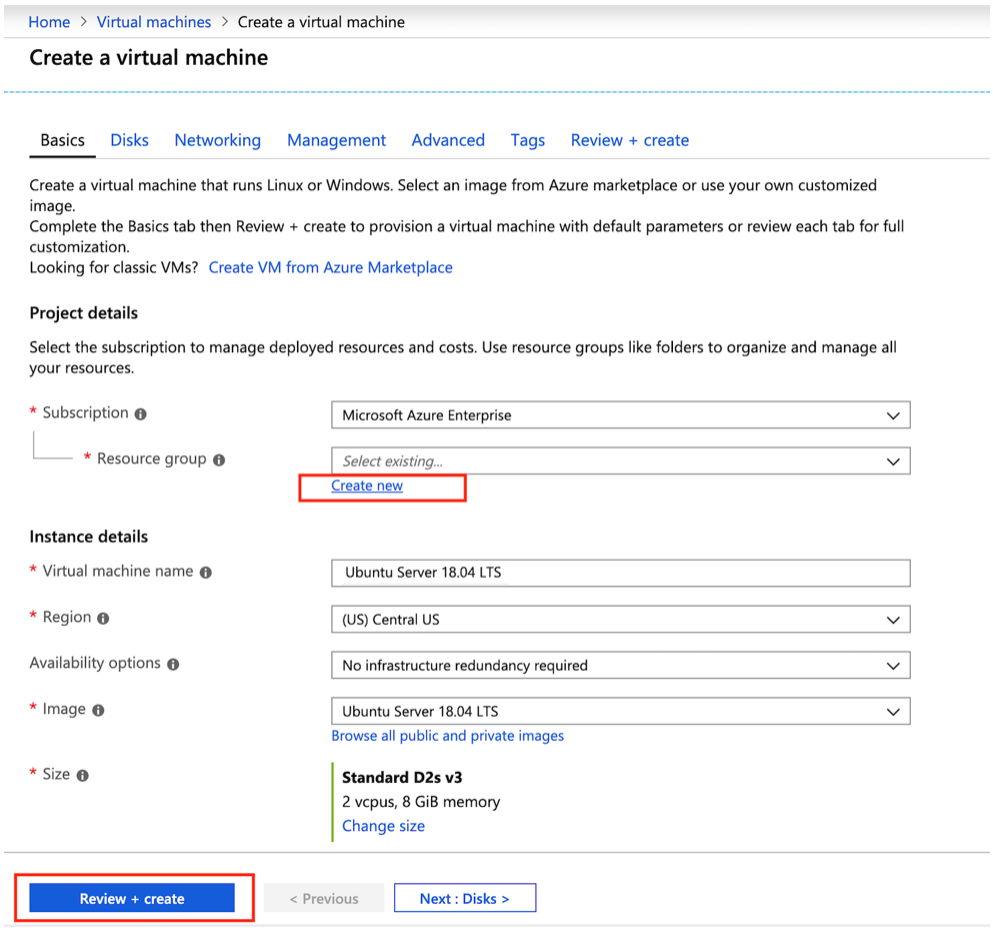


Figure 113. Initiate VM deployment

3. Proceed to the next section after the deployment is complete.

©2026 Zscaler, Inc. All rights reserved.

Allowing Inbound Ports

To allow inbound ports:

1. Configure the Network Security Group (NSG) to permit inbound SSH access for management, and to allow inbound TCP/514 from the LSS App Connector. By default, these ports are not open in Azure and you must manually allow the inbound connections. Lock down this access to only permitted specific source IPs.
2. In the Search bar on the Azure home page, enter `network security groups`, then select **Network security groups**. A new security group is automatically created by Azure with your resource group tied to it.

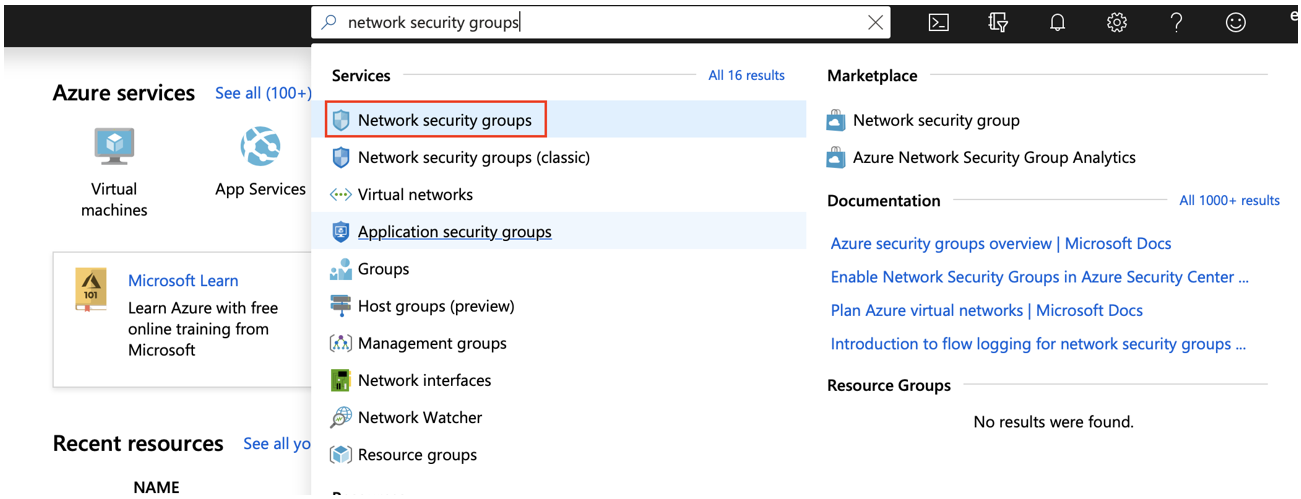


Figure 114. Network Security Groups

3. Open the auto-created network security group.

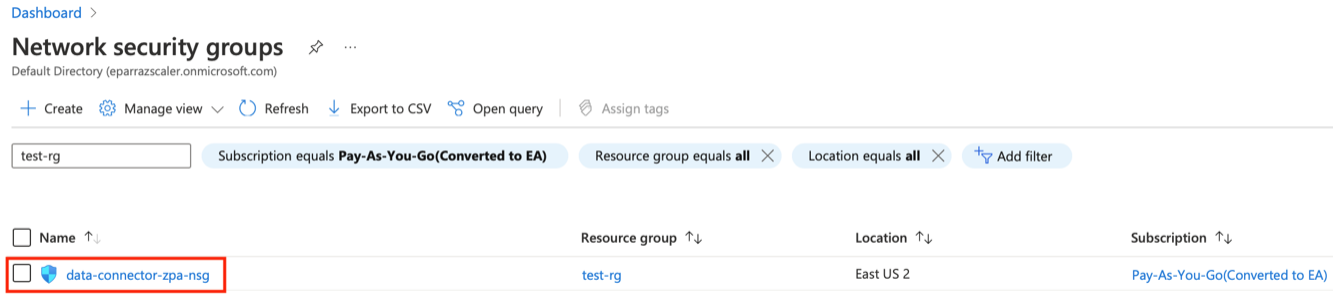


Figure 115. Network Security Groups, continued

©2026 Zscaler, Inc. All rights reserved.

4. Select the **Inbound Security Rules** option and configure the following rules to allow inbound connections.
   - Protocol and port TCP/514 from your Zscaler LSS App Connector IP.
   - Protocol and port TCP/22 from your trusted network or management station.
5. Review the following sections to see how to configure each rule separately.

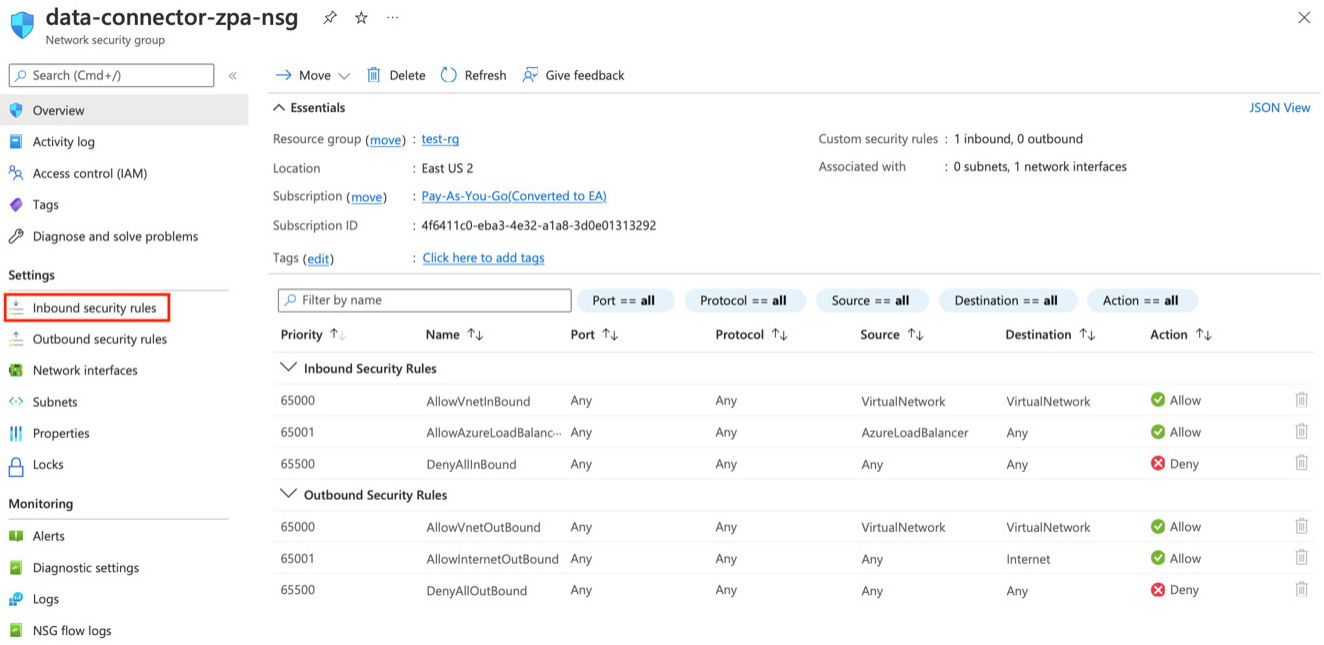


Figure 116. Inbound security rules

©2026 Zscaler, Inc. All rights reserved.

Add Inbound Security Rule for Syslog

Configure these fields to match your environment. Although the destination port is TCP/514, you can configure the destination port on the ZPA side to be other ports. If you set this port to something other than 514 in ZPA, then the port number must match here. After you are done, click **Add** and proceed to the next section.

For testing purposes, any source IP can connect to this ZPA data connector VM on port 514. Post testing, restrict this access to the LSS App Connector source IP only.

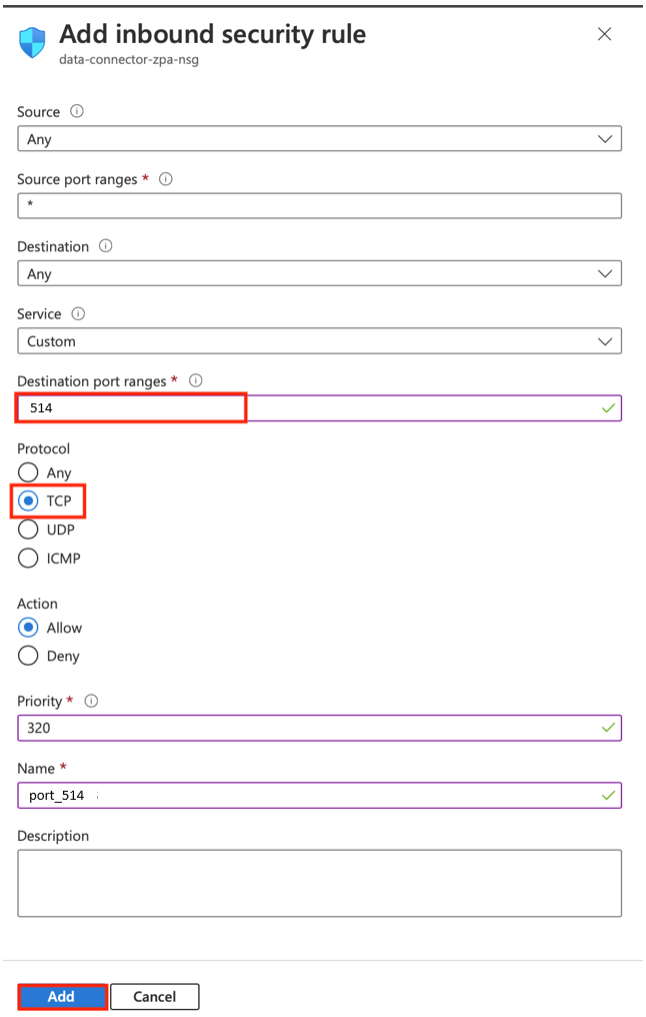


Figure 117. Add inbound security rule for LSS

©2026 Zscaler, Inc. All rights reserved.

Add Inbound Security Rule for SSH

Configure these fields to match your environment. Ensure the **Source** matches your trusted network or management station from which you want to allow the SSH access.

For testing purposes, any source IP can connect to this data connector VM on port 22. Post testing, restrict this access to the trusted management source IP only.

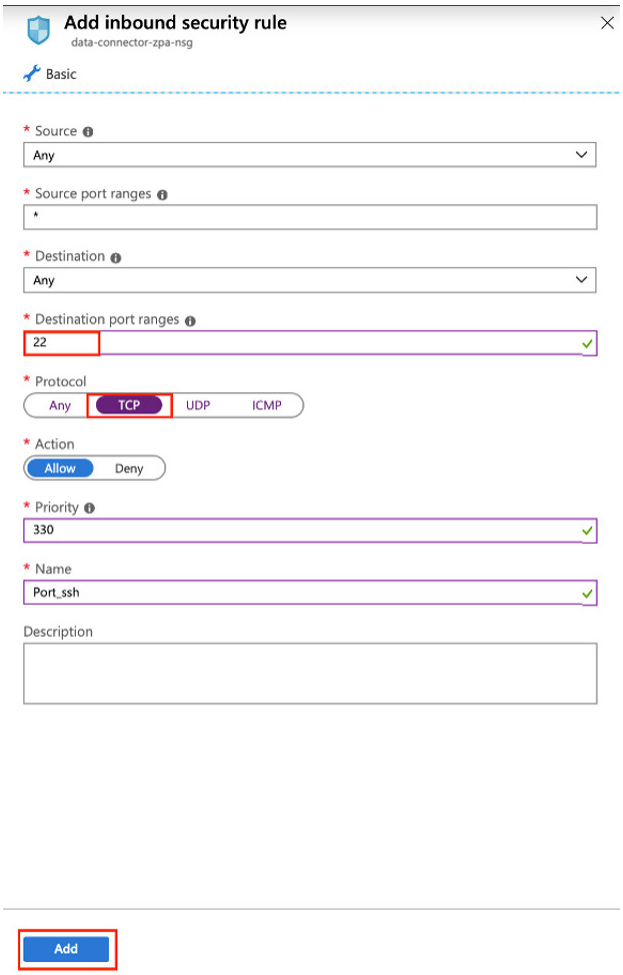


Figure 118. Add inbound security rule for SSH

©2026 Zscaler, Inc. All rights reserved.

Create and Configure Sentinel Instance

In Azure, search for `Sentinel` and select **Microsoft Sentinel**.

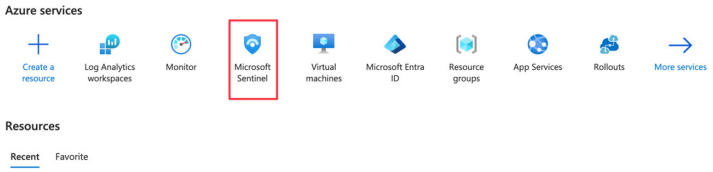


Figure 119. Return to Microsoft Sentinel

Create Log Analytics Workspace

First, create a workspace for Microsoft Sentinel. A log analytics workspace is a unique environment for Azure Monitor log data. Each workspace has its own data repository. Configuration, data sources, and solutions are configured to store their data in a particular workspace. Click **Create**.

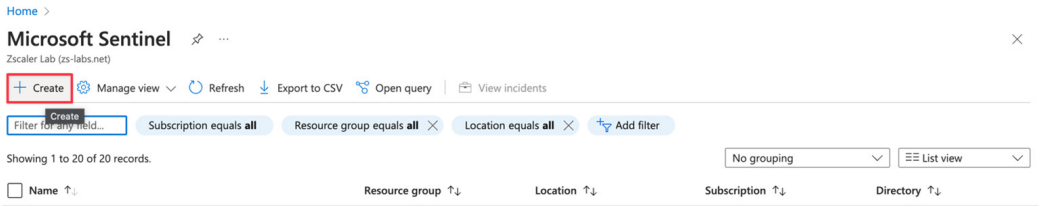


Figure 120. Create a new workspace

Name, Add, and Link a Resource Group to a Workspace

This document presumes the following steps are performed in a new, non-production environment. In this guide, the resource group created in Bind this VM to a Resource Group is reused. If you are configuring this in an existing production environment, your steps might be slightly different. Proceed with one of the following two steps:

- Link an existing resource group to the new workspace, or
- Create and then link an entirely new resource group to this new workspace.

To name, add, and link a resource group:

1. Select a **Resource Group**, **Name**, and **Region** for your workspace.
2. Click **Review + Create**.

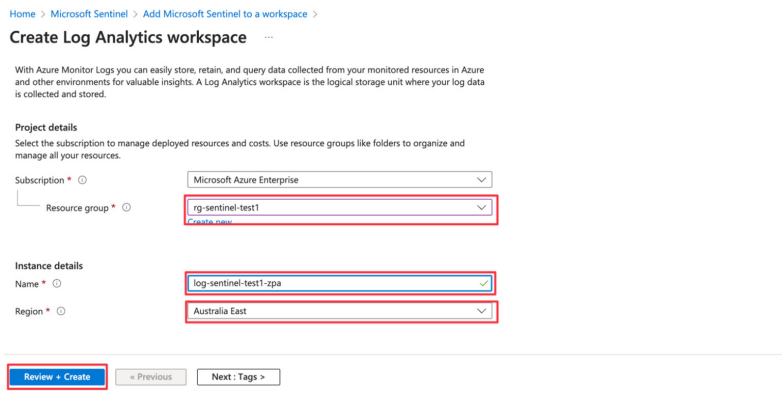


Figure 121. Name, add, and link a resource group for your workspace

3. Click **OK**.

©2026 Zscaler, Inc. All rights reserved.

Add Microsoft Sentinel Workspace

To add a Microsoft Sentinel Workspace:

1. Select the newly created workspace.
2. Click **Add**. Wait until the deployment finishes before proceeding to the next section. This takes a few minutes.
3. Click the workspace you've just created, then click **Add** to add Microsoft Sentinel to the workspace.

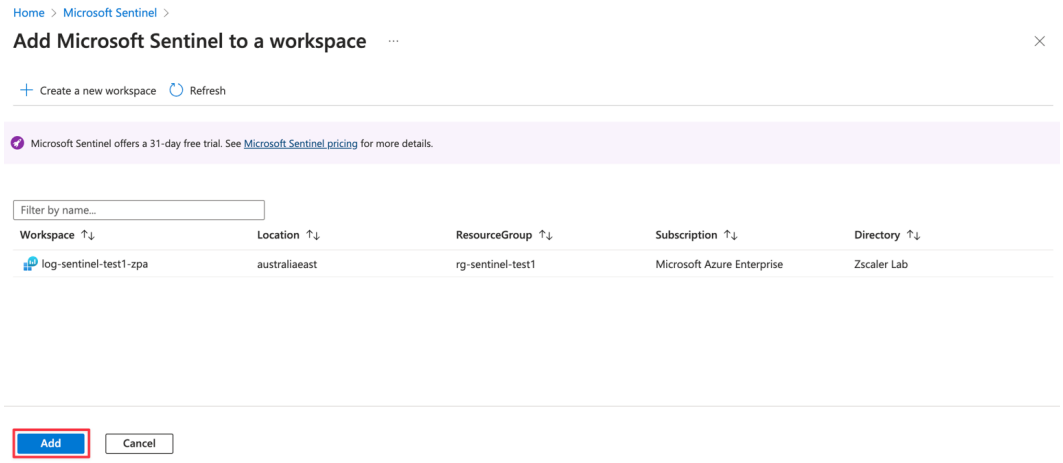


Figure 122. Add Microsoft Sentinel workspace

Configure Data Collection

To configure data collection:

1. Configure data collection by going to **Content hub**, and searching for `azure monitor agent`.
2. Select **Syslog**.
3. Click **Install**.

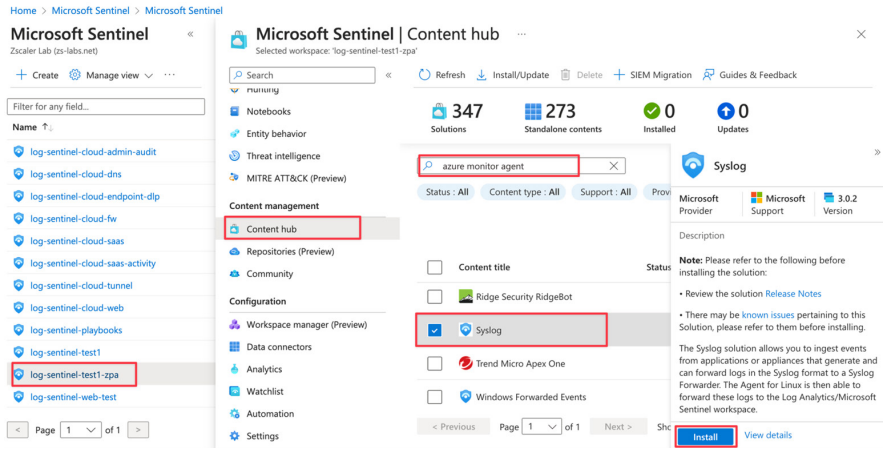


Figure 123. Configure Zscaler data collection

©2026 Zscaler, Inc. All rights reserved.

4. After the Syslog is installed, click **Manage**.

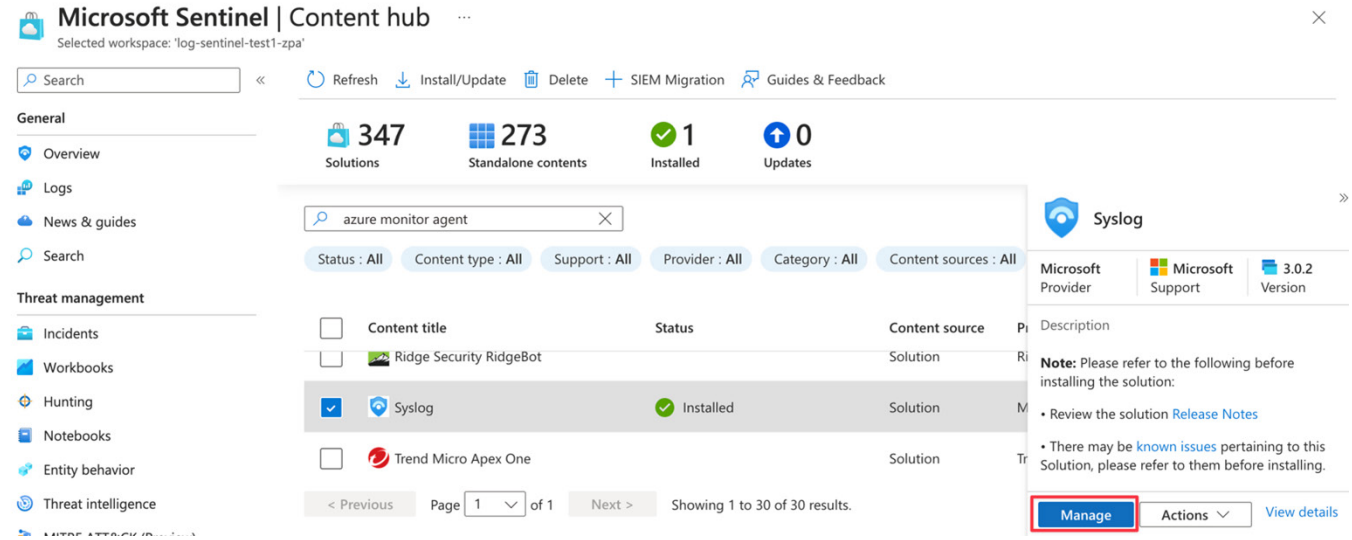


Figure 124. Click Manage

5. Select **Syslog via AMA**, then click **Open connector page**.

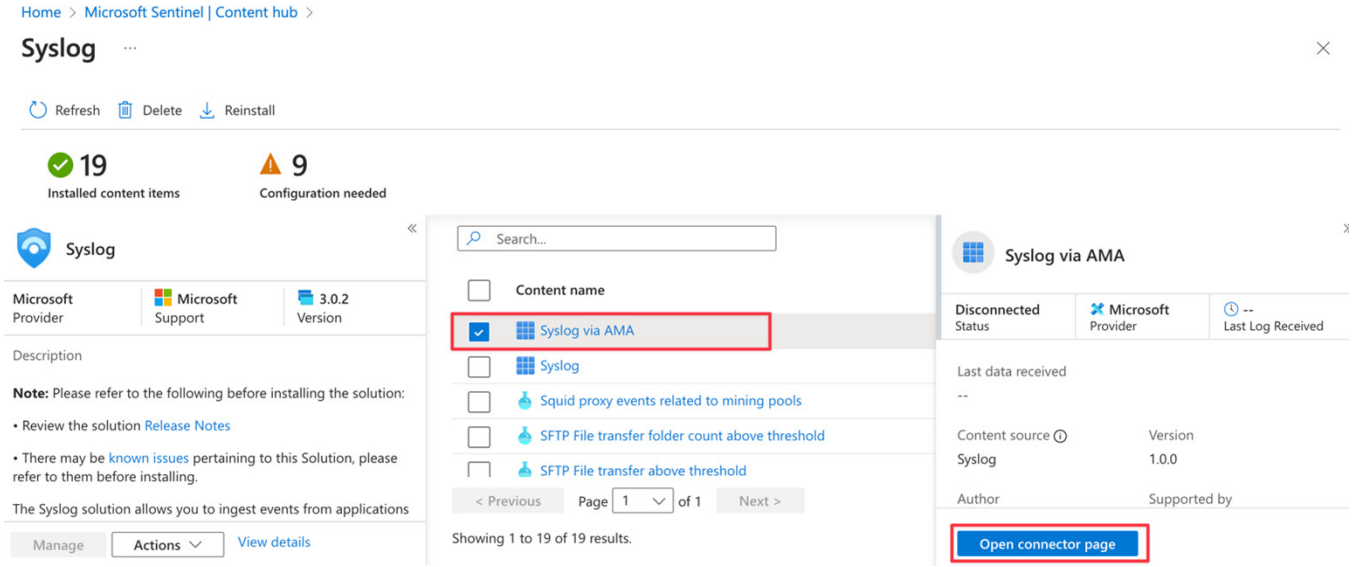


Figure 125. Steps to configure ZPA data connector

©2026 Zscaler, Inc. All rights reserved.

6. Click **Create data collection rule**.

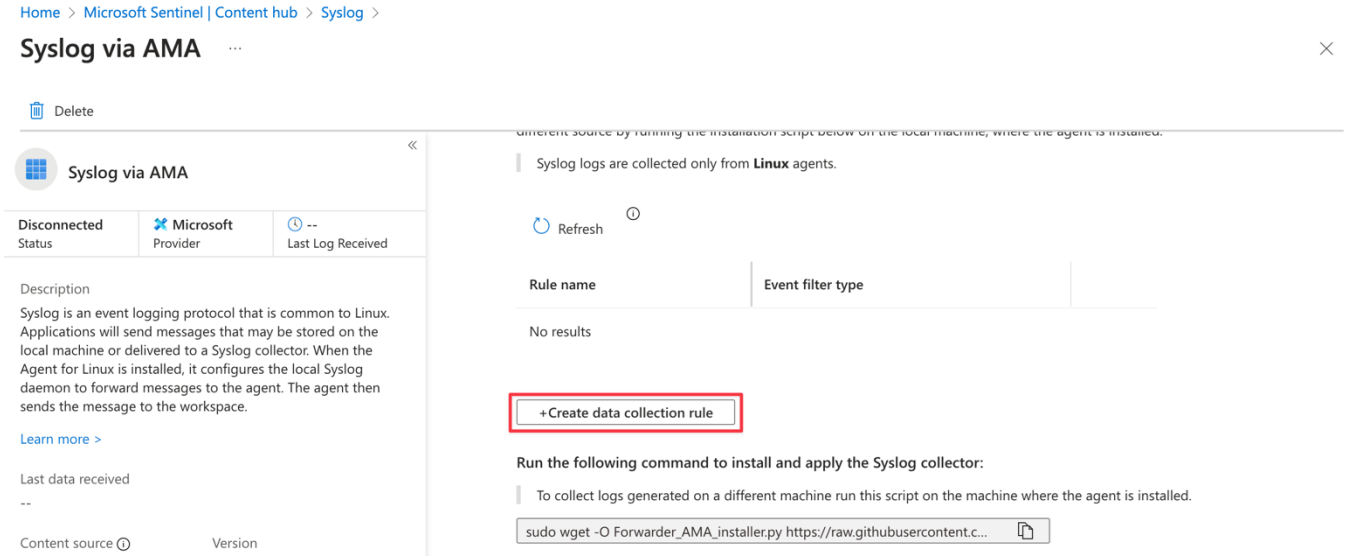


Figure 126. Create data collection rule

7. Enter a Data Collection **Rule name** and click **Next: Resources**.

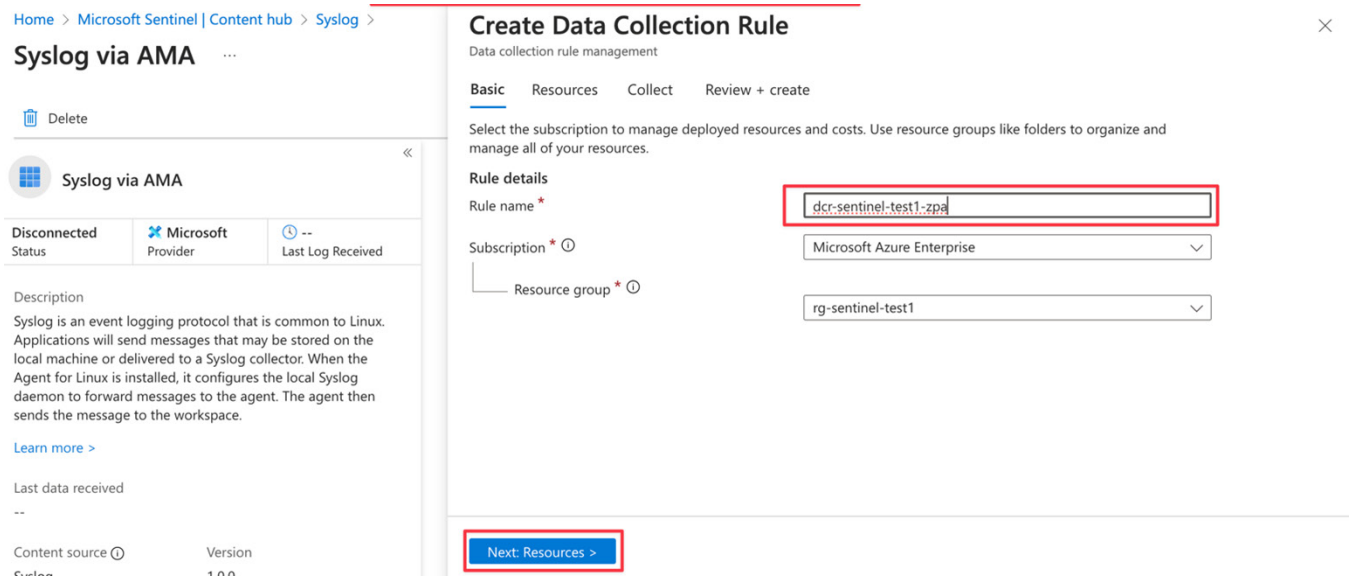


Figure 127. Rule name

©2026 Zscaler, Inc. All rights reserved.

8. Select the virtual machine you created earlier and click **Next: Collect**.

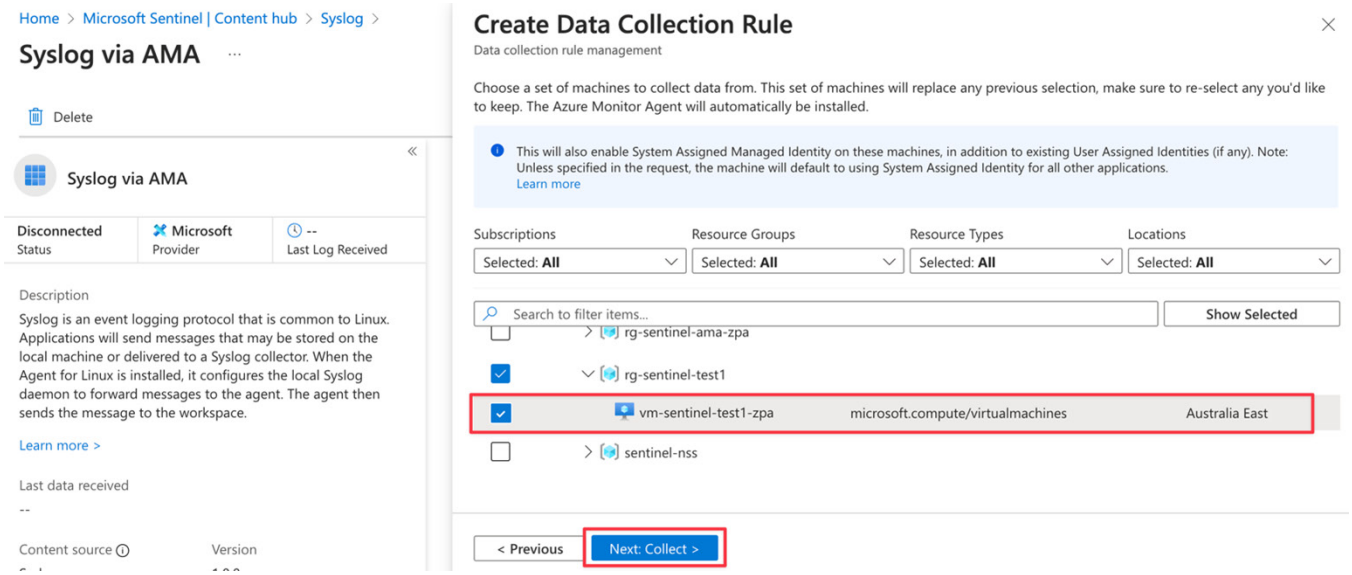


Figure 128. Select VM

©2026 Zscaler, Inc. All rights reserved.

9. Deselect **Collect messages without PRI header (facility and severity)**, then ensure that the **LOG_USER** facility is set to log level **LOG_DEBUG**.
10. Click **Next: Review + create**.

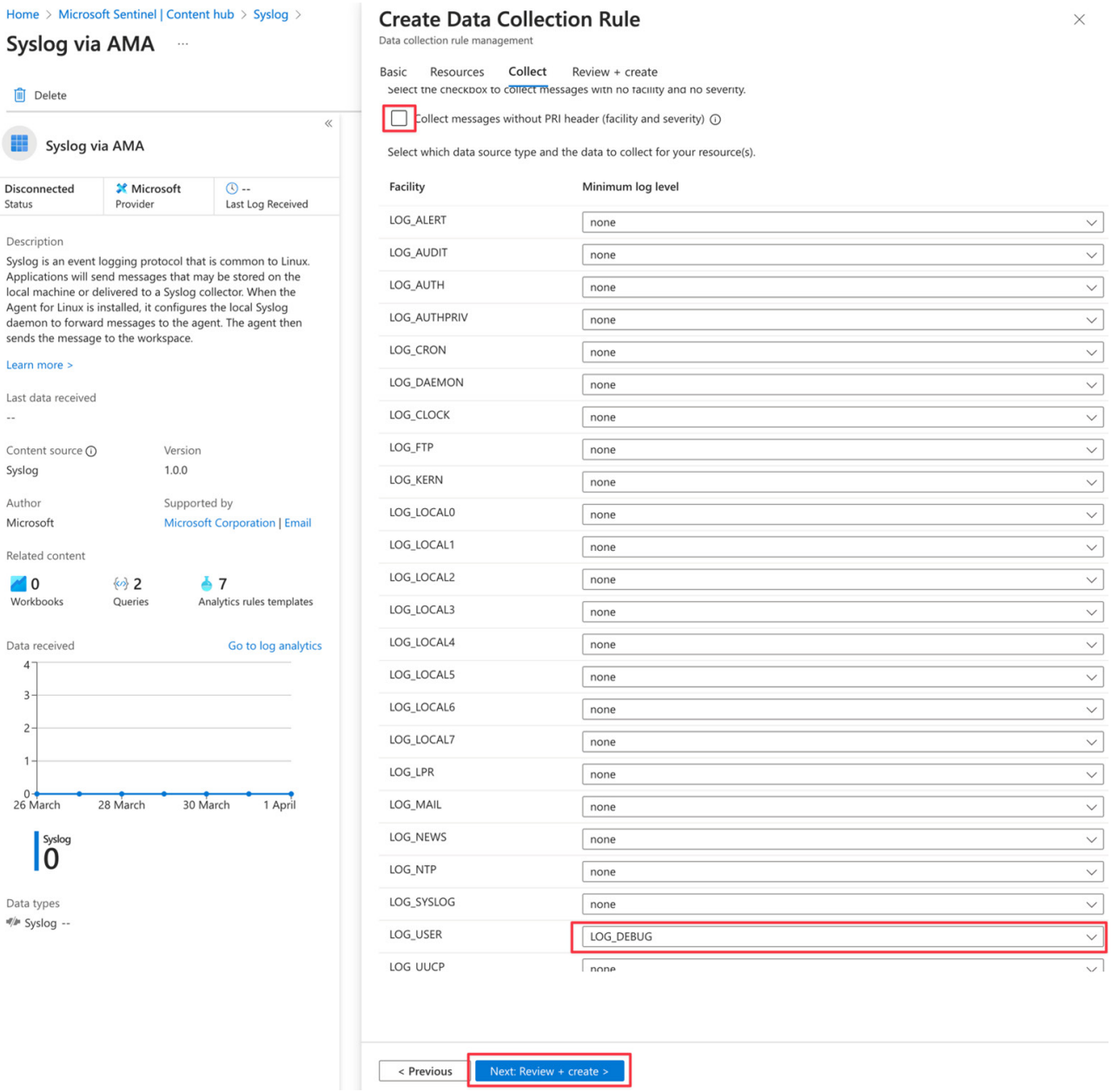


Figure 129. Collect messages without PRI header (facility and severity)

11. Click **Create**.

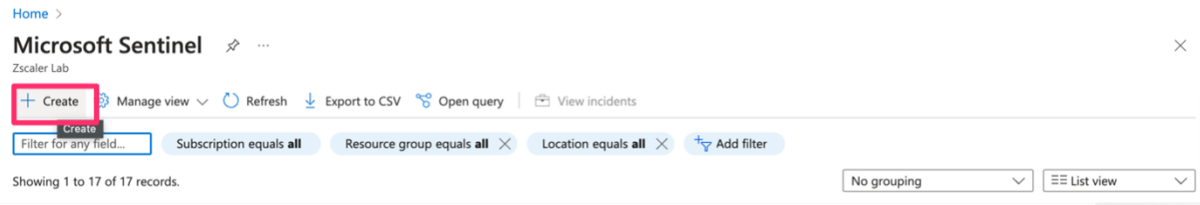


Figure 130. Create rule

©2026 Zscaler, Inc. All rights reserved.

Configure Azure Monitor Agent on the Linux Virtual Machine

The Microsoft AMA requires a syslog daemon to be installed so the VM can forward logs to the Log Analytics workspace. This guide installs the following

- rsyslog
- python-is-python3

If you want to install alternatives, review the Microsoft Sentinel Resources.

1. Connect to the VM via SSH and run the following commands:

```
sudo add-apt-repository -y ppa:adiscon/v8-stable &&

sudo apt-get update

sudo apt-get install -y rsyslog

sudo apt install python-is-python3
```

2. After the commands are finished running, copy the link found on the **Syslog via AMA** page in Microsoft Sentinel.

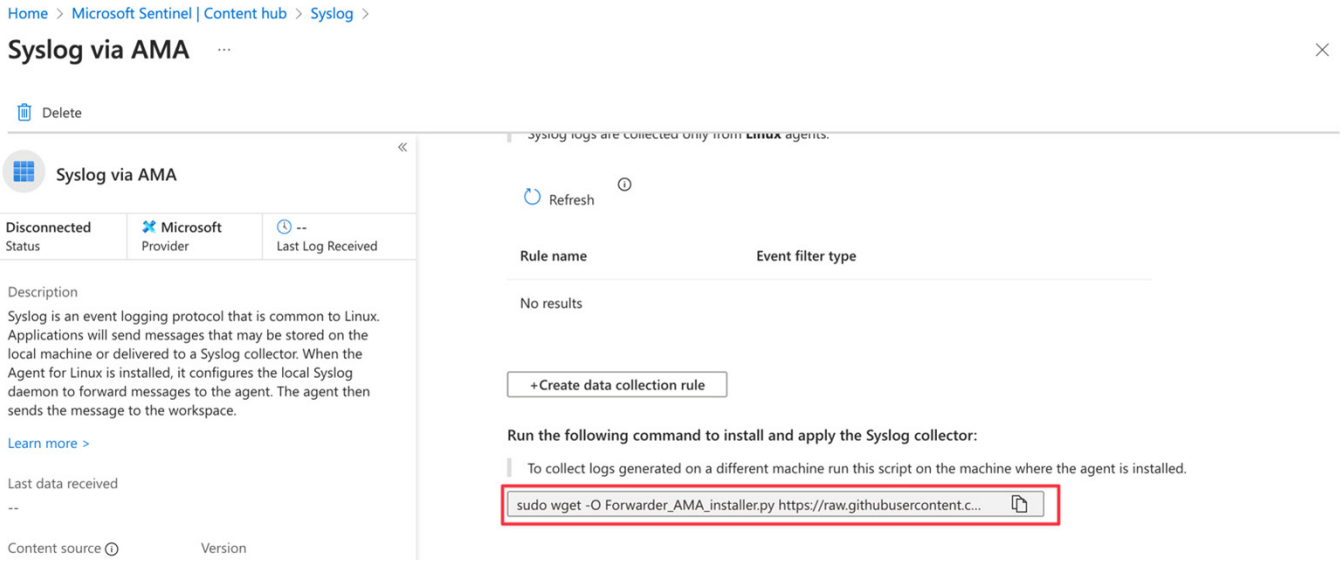


Figure 131. Syslog via AMA page

©2026 Zscaler, Inc. All rights reserved.

Run the command in your VM. When complete, the output looks like the following:

```
azureuser@vm-sentinel-test1-zpa:~$ sudo wget -O Forwarder_AMA_installer.py https://raw.githubusercontent.com/Azure/Azure-Sentinel/master/DataConnectors/Syslog/Forwarder_AMA_installer.py&&sudo python Forwarder_AMA_installer.py
--2024-04-02 07:14:07--  https://raw.githubusercontent.com/Azure/Azure-Sentinel/master/DataConnectors/Syslog/Forwarder_AMA_installer.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13545 (13K) [text/plain]
Saving to: 'Forwarder_AMA_installer.py'

Forwarder_AMA_installer.py     100%[===========================================>]  13.23K  --.-KB/s    in 0s

2024-04-02 07:14:07 (59.4 MB/s) - 'Forwarder_AMA_installer.py' saved [13545/13545]

Located rsyslog daemon running on the machine
Rsyslog.conf configuration was changed to fit required protocol - /etc/rsyslog.conf
Restarting rsyslog daemon.
sudo service rsyslog restart
Rsyslog daemon restarted successfully
Please note that the installation script opens port 514 to listen to incoming messages in both UDP and TCP protocols. To change this setting, refer to the Rsyslog configuration file located at '/etc/rsyslog.conf'.

Warning: please make sure your Syslog daemon configuration does not store unnecessary logs. This may cause a full disk on your machine, which will disrupt the function of the oms agent installed. For more information:
https://www.rsyslog.com/doc/master/configuration/actions.html
Installation completed successfully
azureuser@vm-sentinel-test1-zpa:~$
```

Figure 132. Output

Create a New ZPA_CL Custom Log Table in Your Log Analytics Workspace

To create a new ZPA custom log:

1. Click the **Cloud Shell** icon in your Azure portal.

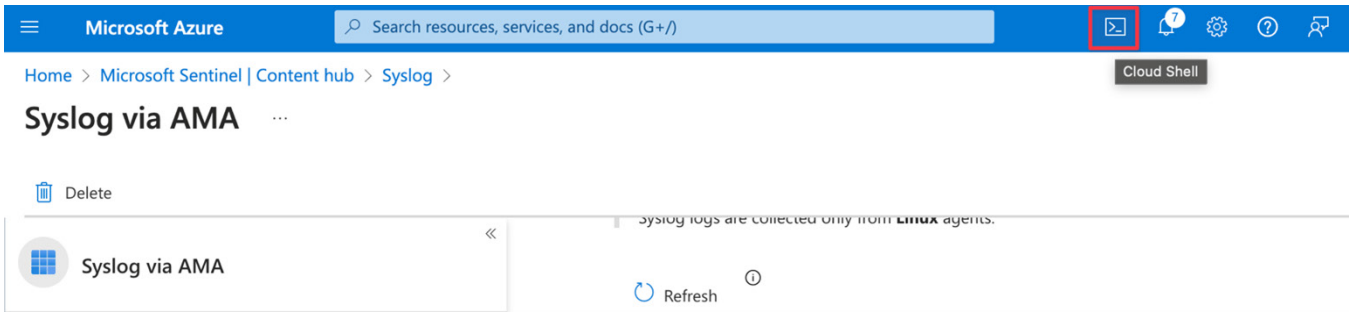


Figure 133. Cloud Shell icon

2. Run the following command:

```
az monitor log-analytics workspace table create --resource-group <log analytics resource group> --workspace-name <log analytics workspace name> -n ZPA_CL
--columns TimeGenerated=DateTime Message=String SourceSystem=String RawData=String ManagementGroupName=String Computer=String
```

©2026 Zscaler, Inc. All rights reserved.

After running the command, the following output is shown:

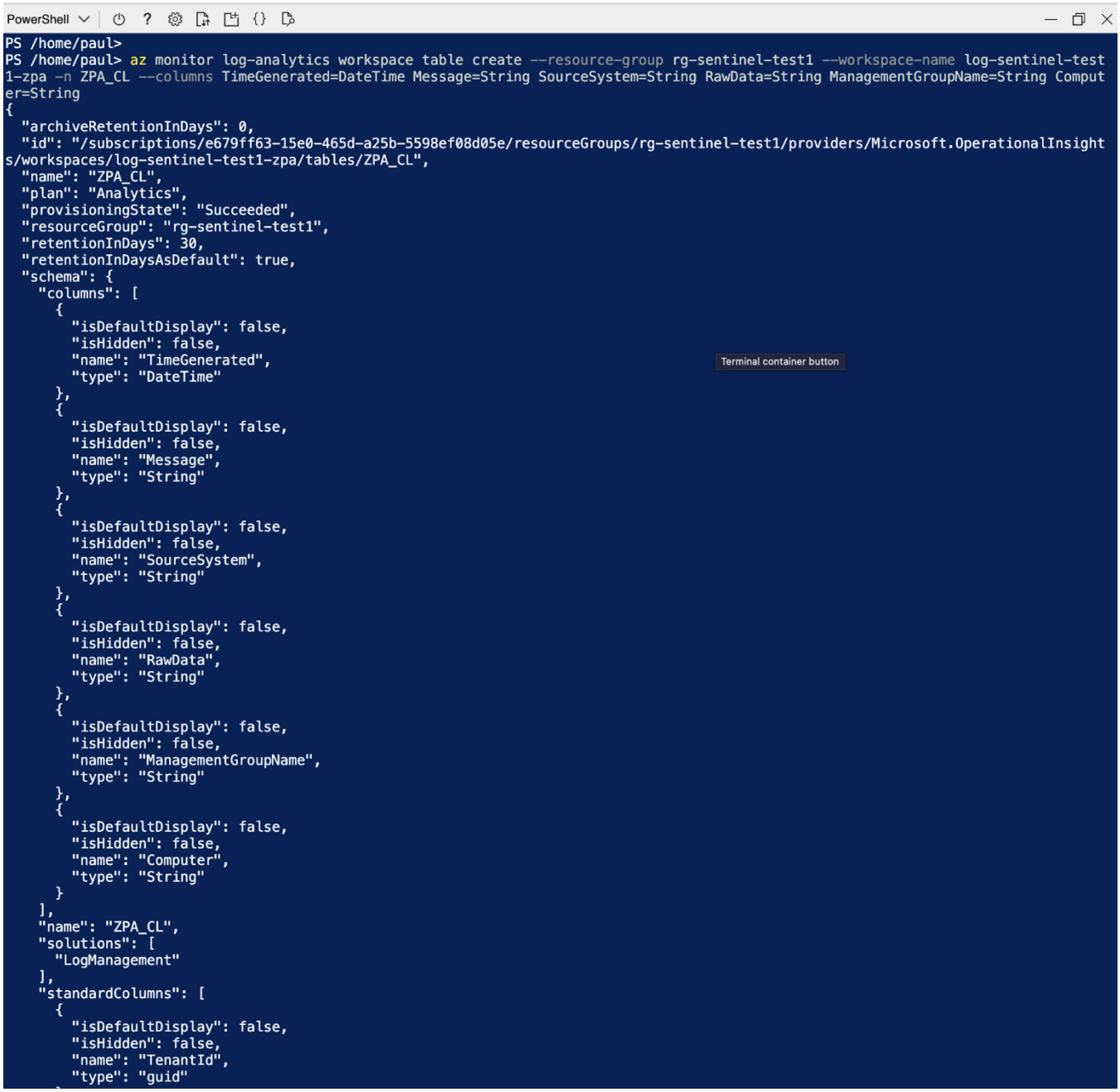


Figure 134. Command output

©2026 Zscaler, Inc. All rights reserved.

Modify the DCR Using a PowerShell Script to Send Logs to the ZPA_CL Table

To modify the DCR using PowerShell:

1. Download a copy of the PowerShell script from the Zscaler GitHub repo.
2. Modify the script with the **Data Collection Rule subscription ID**, the **Data Collection Rule Name**, and the **Data Collection Rule Resource Group** name. You can find these under the **Azure Portal** > **Monitor** > **Data Collection Rules**.
3. Click **Overview**.

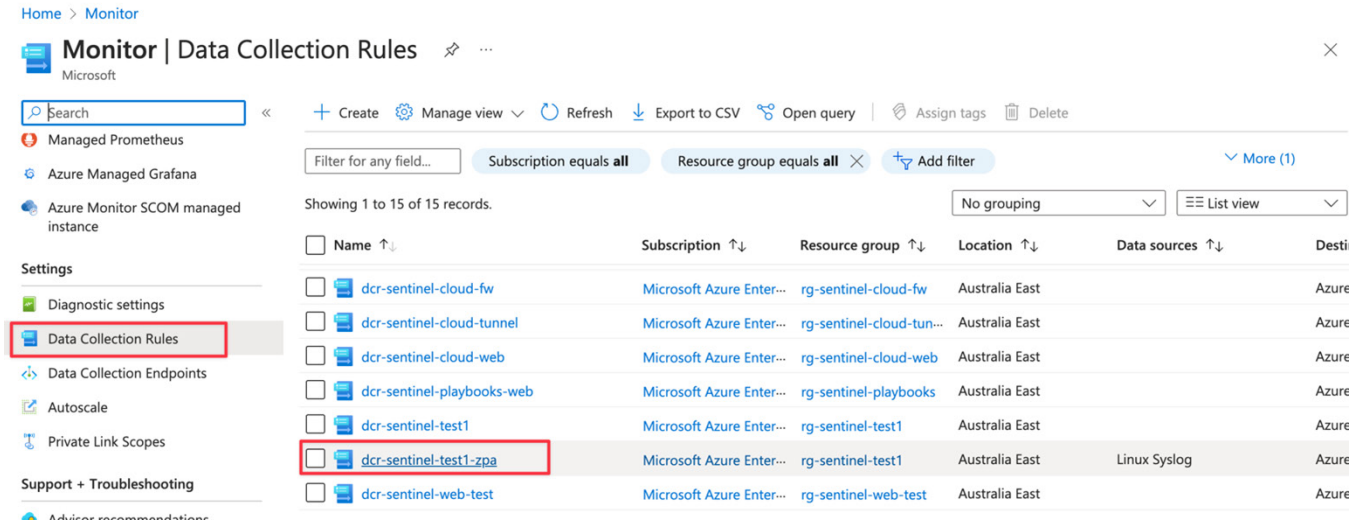


Figure 135. Script modification data

4. Select the **Data Collection Rule**.

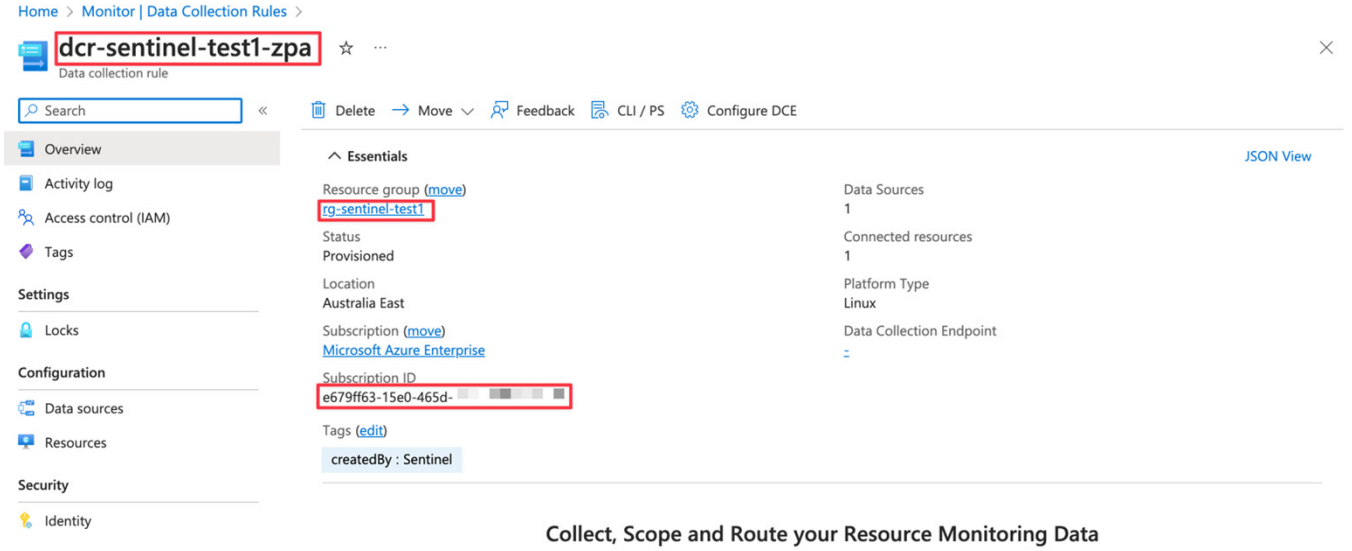


Figure 136. Overview

©2026 Zscaler, Inc. All rights reserved.

5. After you have modified the variables for your tenant, copy and paste the contents of the PowerShell script in your Azure Cloud Shell window.

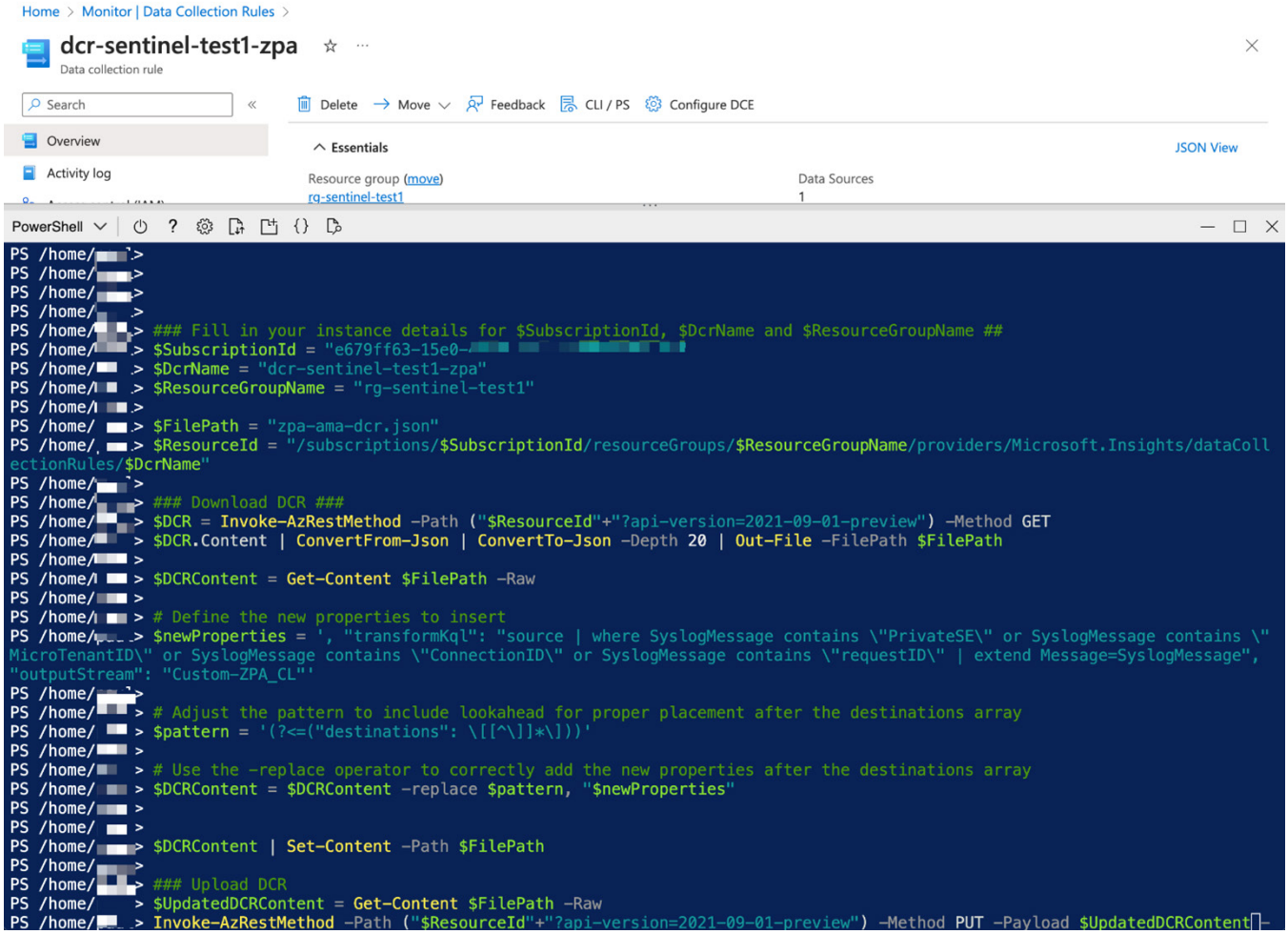


Figure 137. Azure Cloud Shell

6. To check that the PowerShell script has correctly updated the Data Collection Rule, go to **DCR** > **Overview** > **JSON View**.

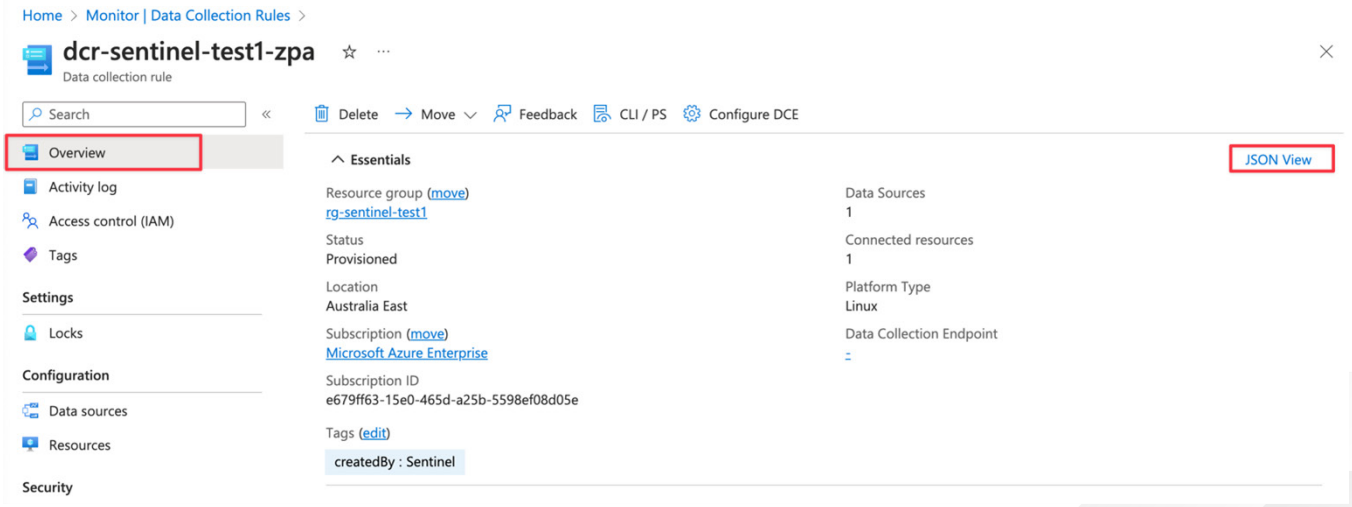


Figure 138. JSON View

©2026 Zscaler, Inc. All rights reserved.

7. Select **2021-09-01-preview** from the **API Versions** drop-down menu, and a transformKql statement is displayed under the dataFlows section.



Figure 139. dataFlows section

Configure ZPA User Activity Logs

The following sections describe how to configure ZPA user activity logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

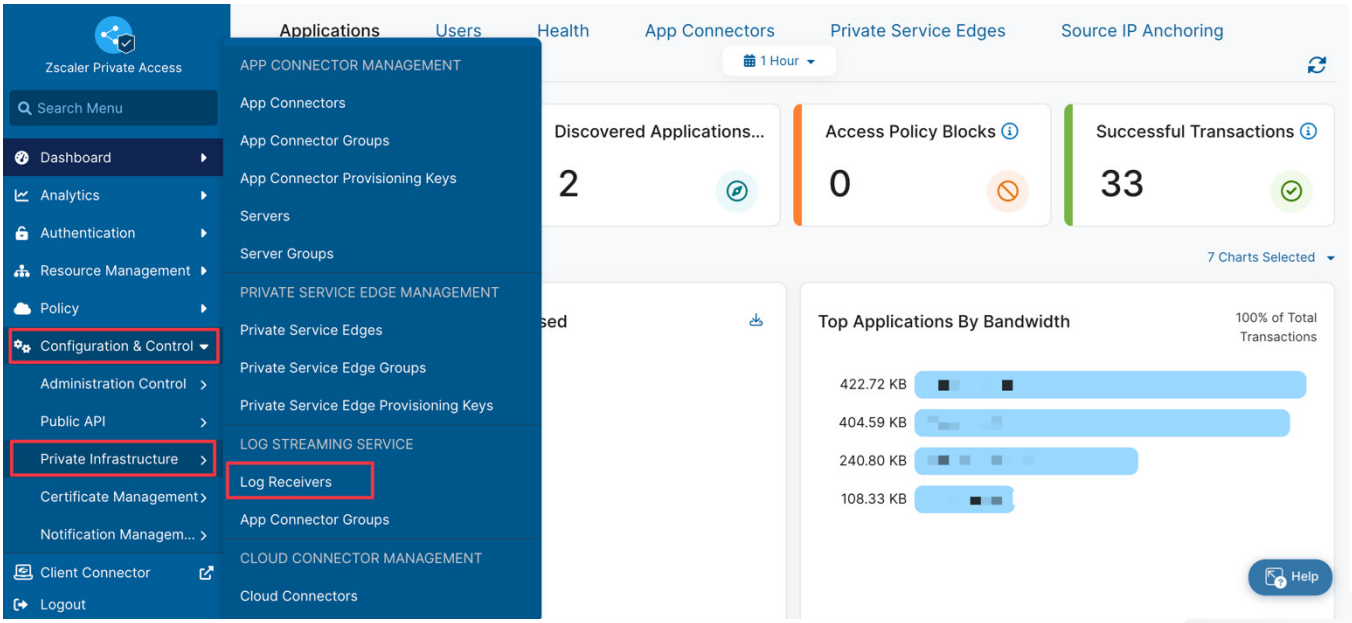


Figure 140. Log Receivers

©2026 Zscaler, Inc. All rights reserved.

2. Click **Add** (+) to add a new Log Receiver.

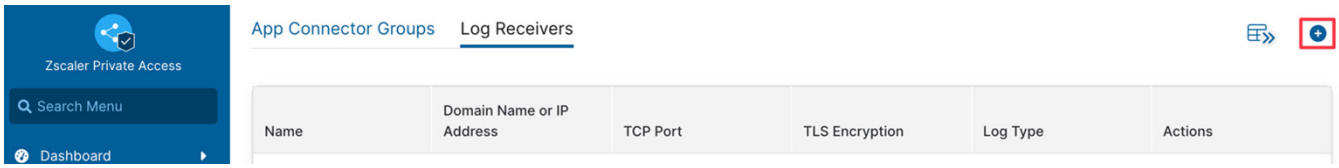


Figure 141. Add new Log Receiver

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

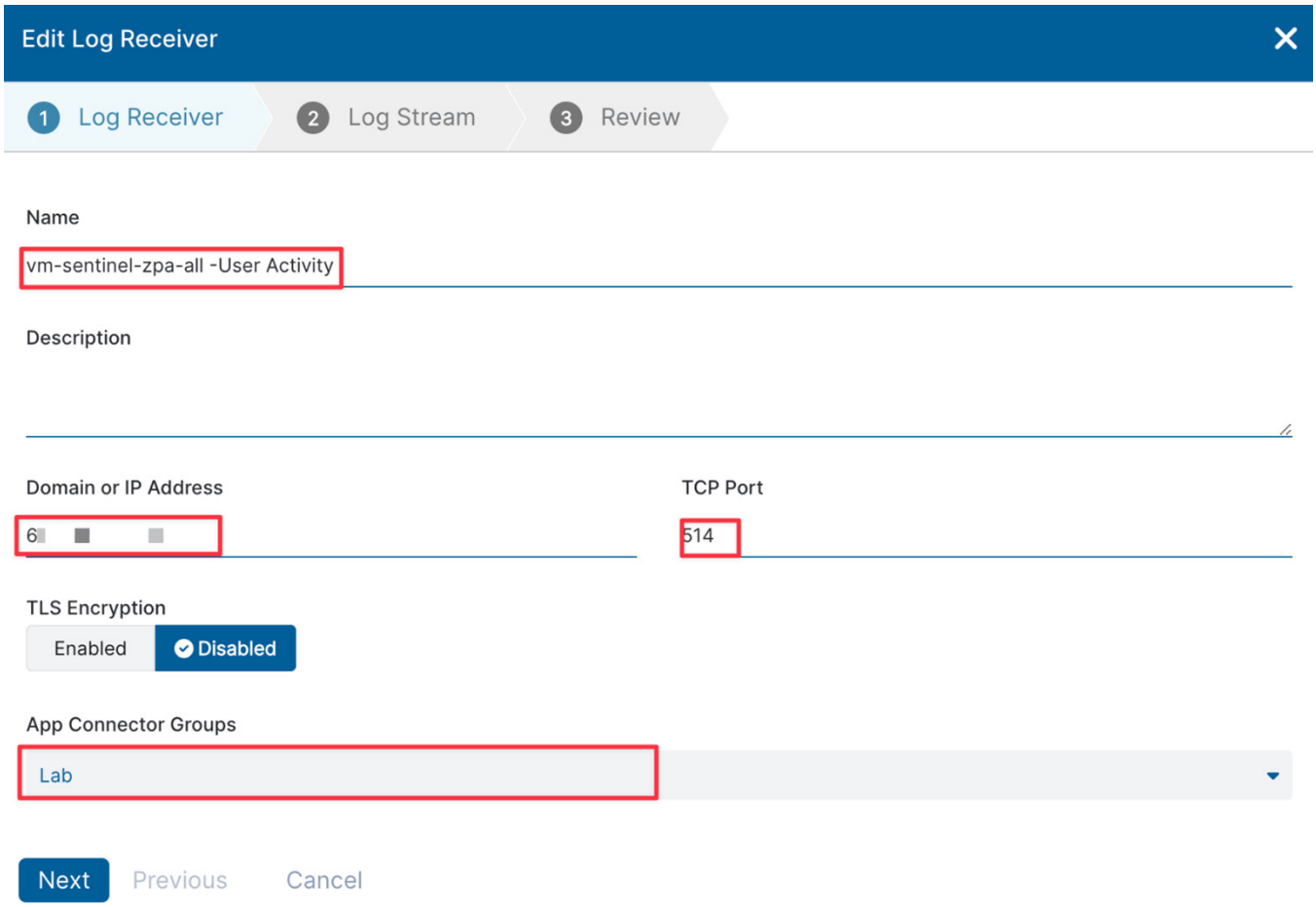


Figure 142. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

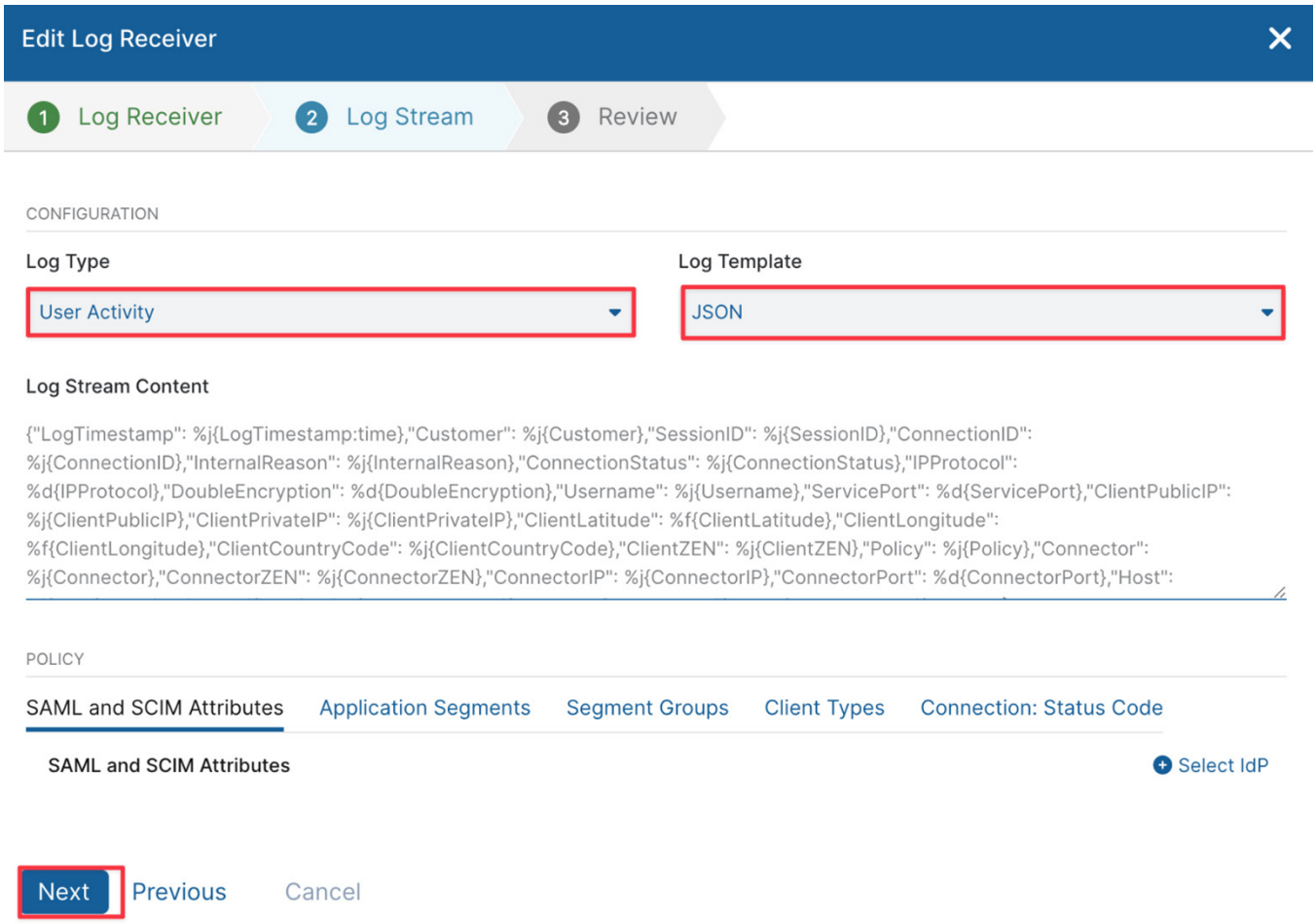


Figure 143. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

7. Click **Save**.

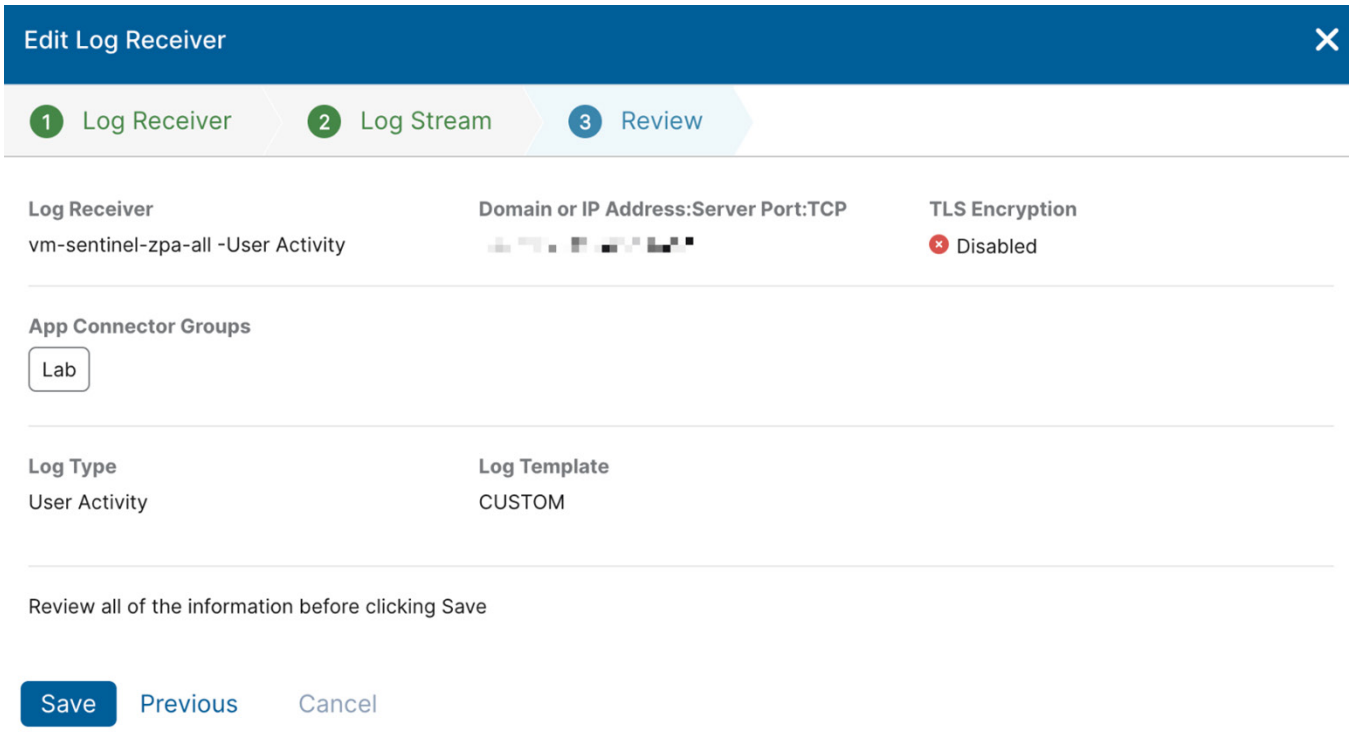


Figure 144. Save log receiver

Configure the ZPAUserActivity Log Analytics Workspace Function

The ZPAUserActivity function is used to parse the ZPA User Activity Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

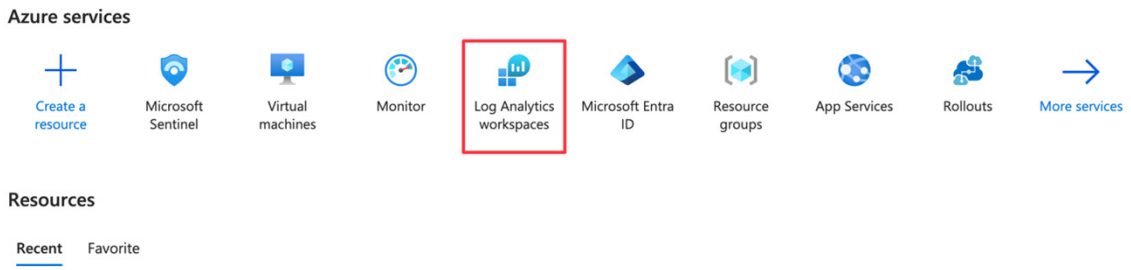


Figure 145. Log Analytics workspaces

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA User Activity logs**.

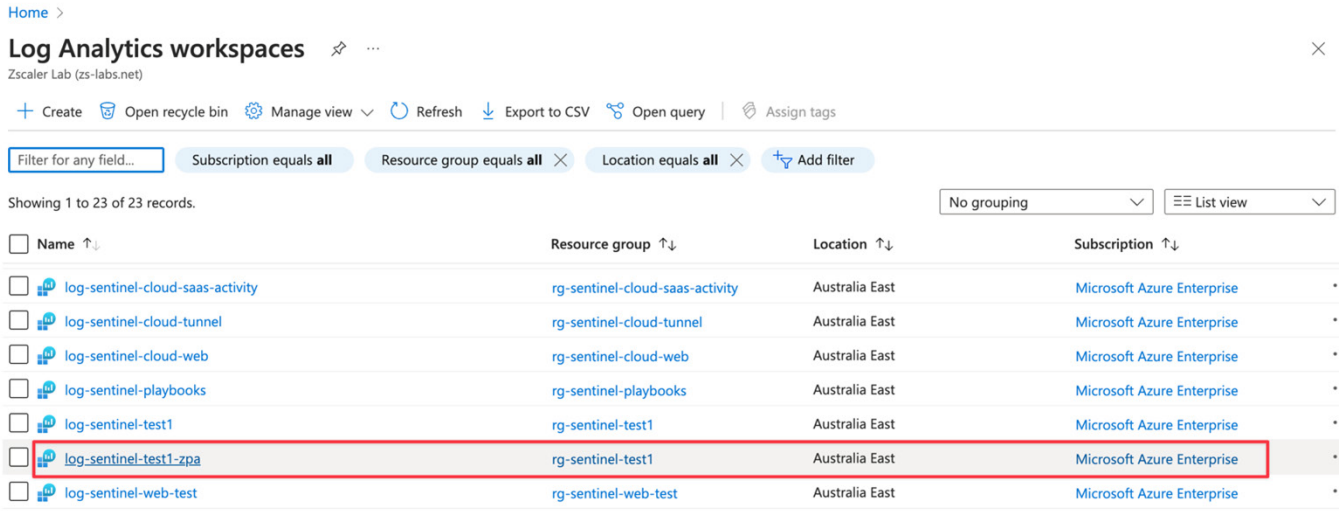


Figure 146. ZPA workspace with logs

4. Click **Logs**.

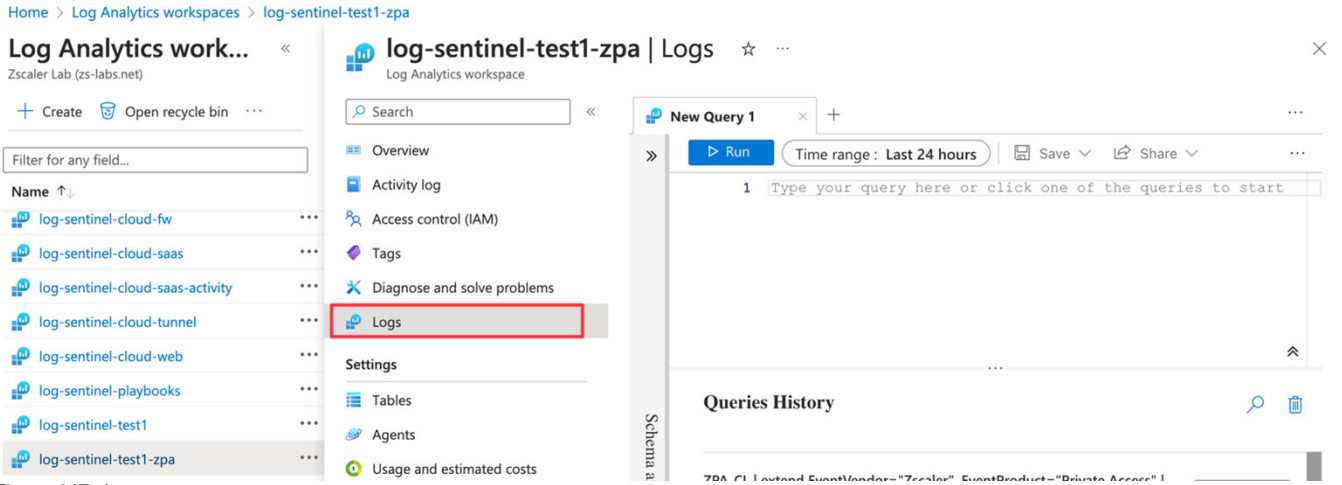


Figure 147. Logs

©2026 Zscaler, Inc. All rights reserved.

5. Paste the content from the `ZPAUserActivity.kql` file into the query window, and select **Save** > **Save as function**.

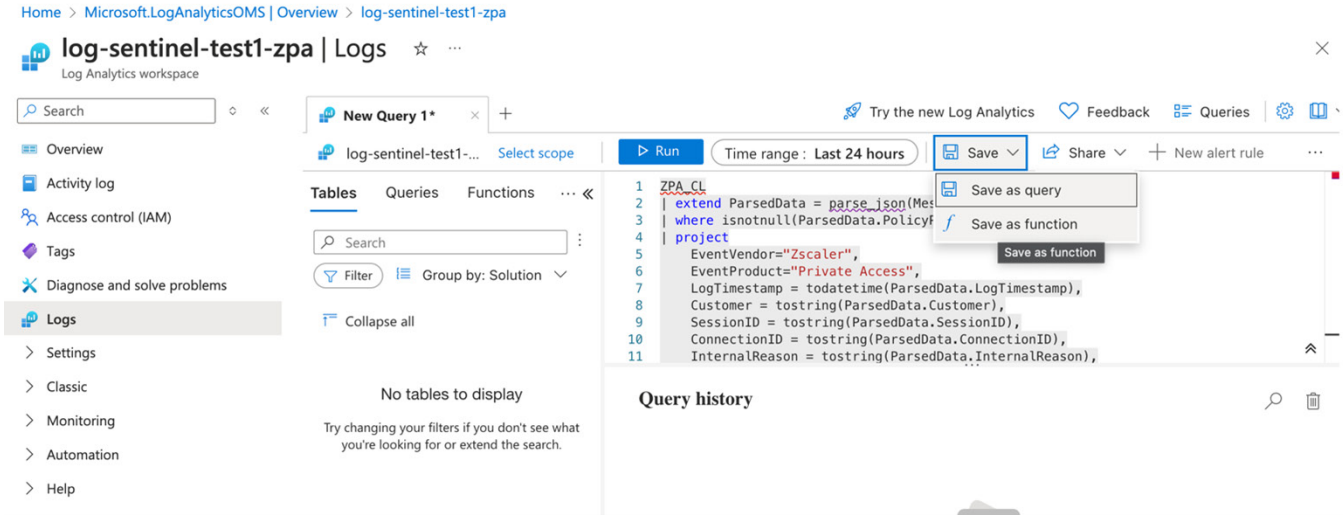


Figure 148. ZPAUserActivity.kql content

6. Enter the **Function name** `ZPAUserActivity`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

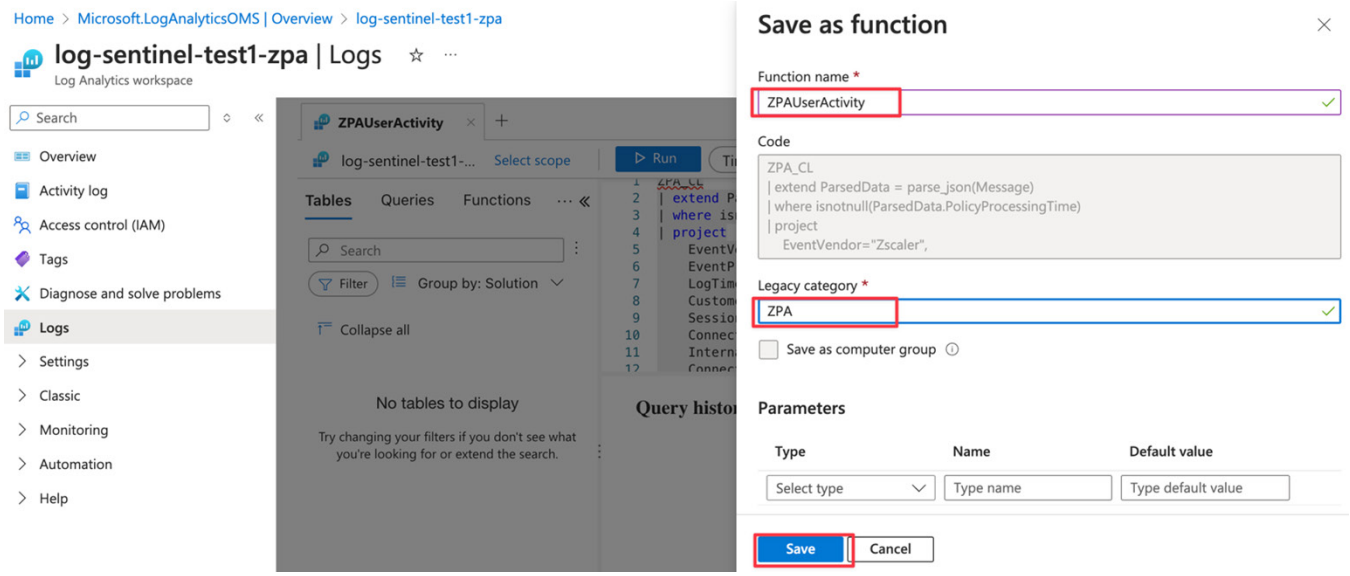


Figure 149. Save ZPAUserActivity

©2026 Zscaler, Inc. All rights reserved.

8. Run queries using the **ZPAUserActivity** function.

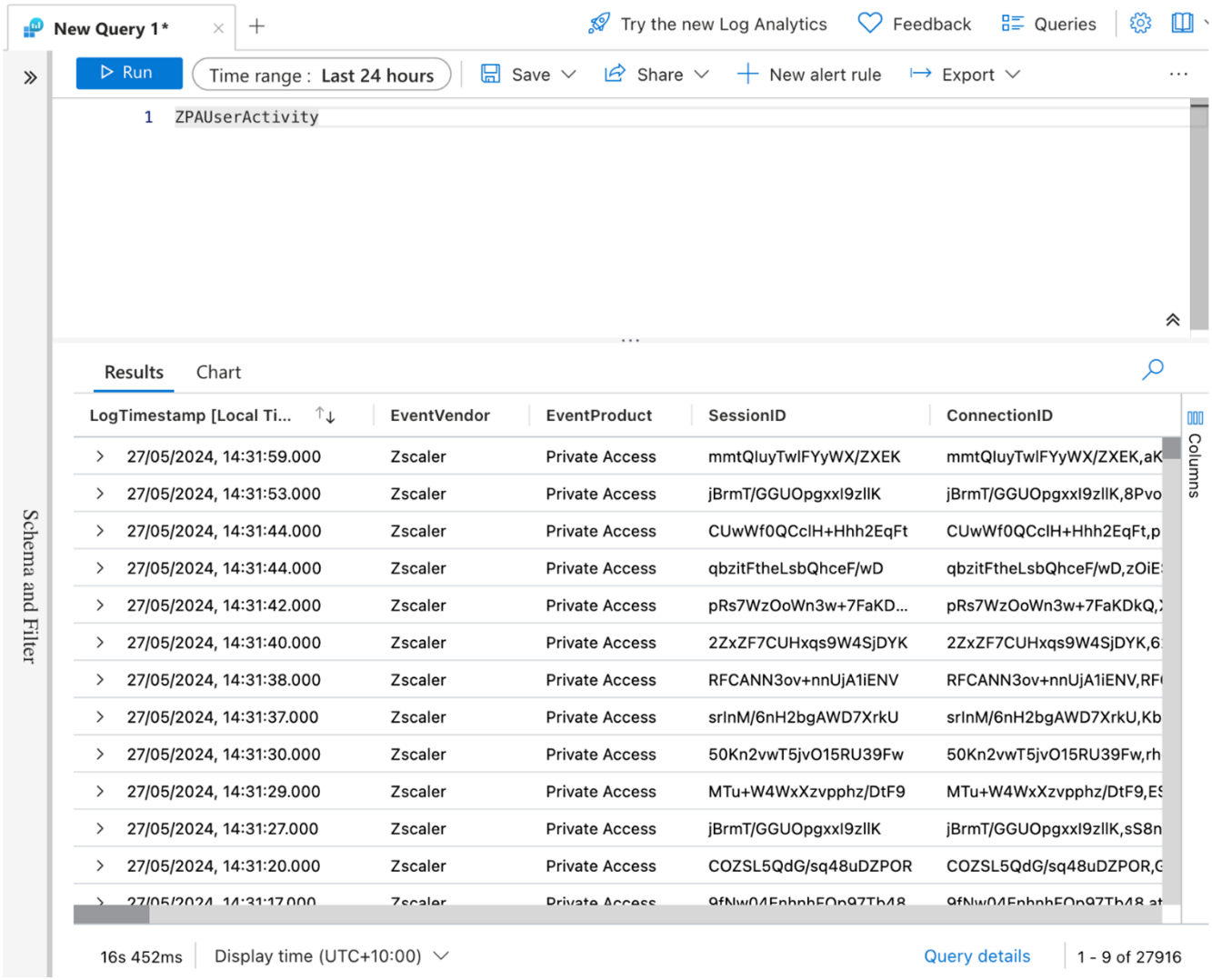


Figure 150. Run queries

Configure the ZPAUserActivity Sentinel Workbook

The ZPAUserActivity workbook is used to graphically display information related to the ZPA User Activity Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you installed your ZPAUserActivity parser.

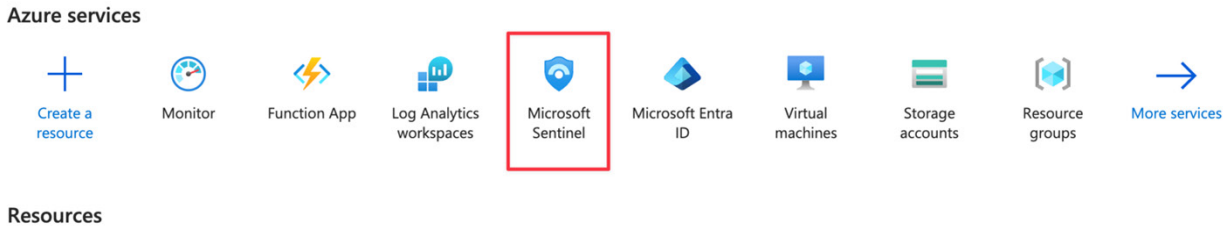


Figure 151. Microsoft Sentinel

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA User Activity logs**.

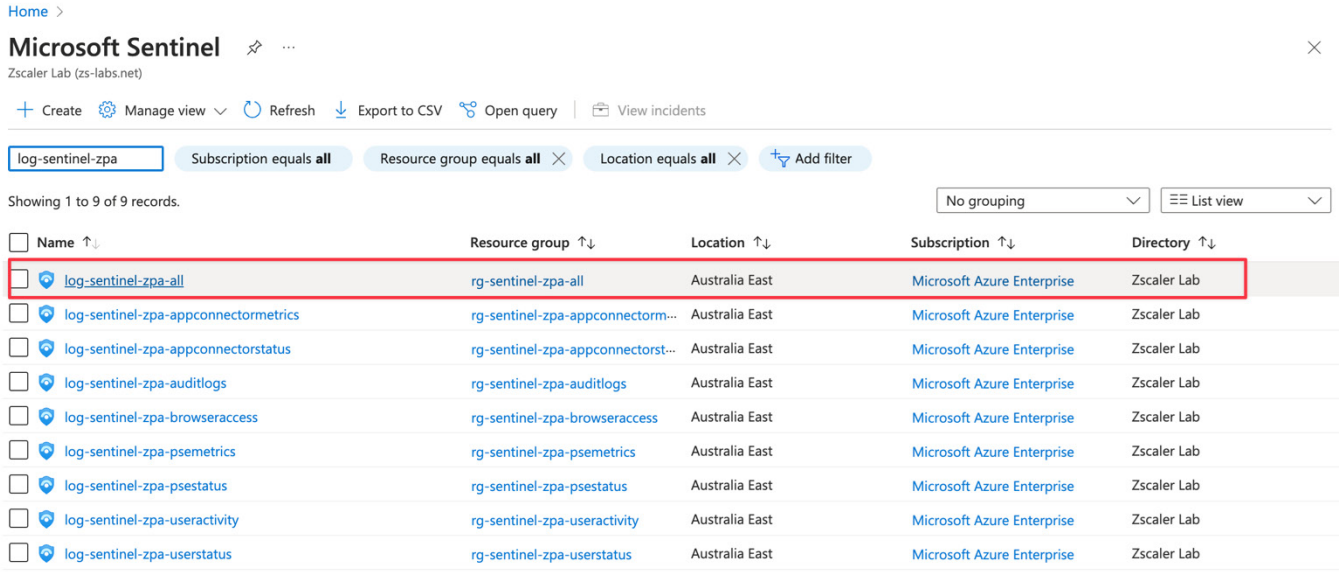


Figure 152. ZPA workspace with logs

4. Click **Workbooks** under **Threat management**.

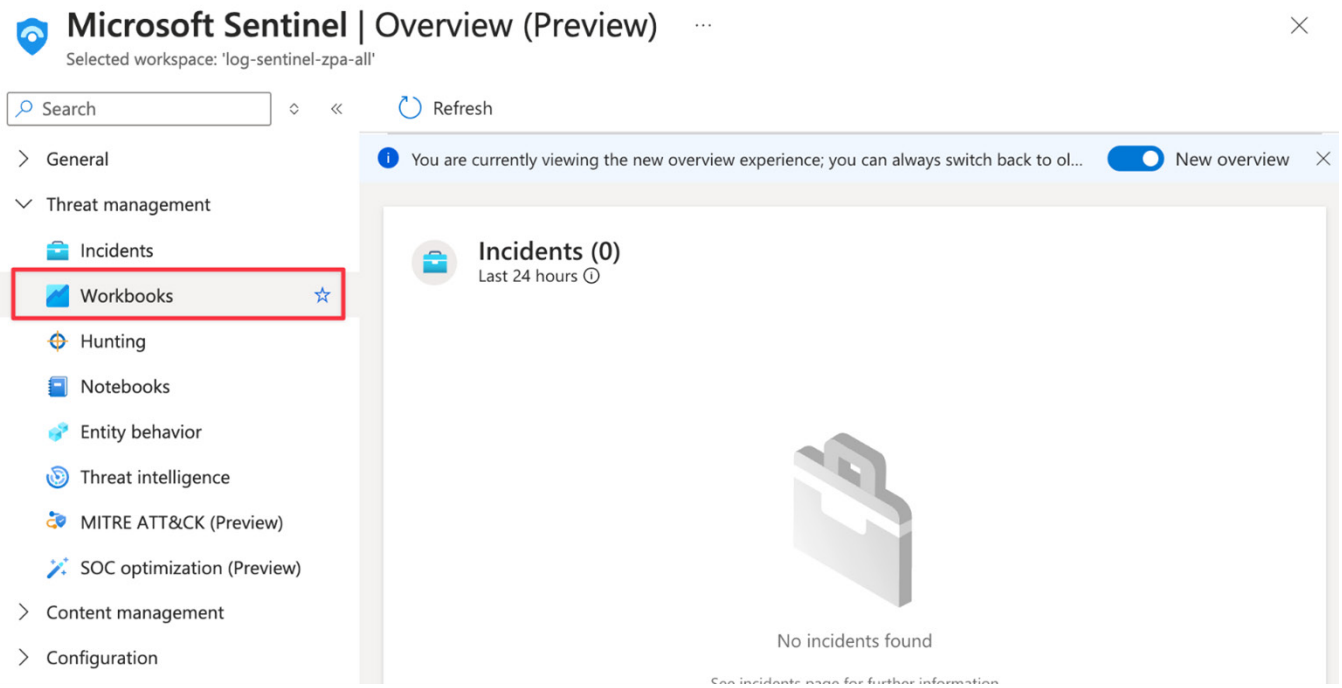


Figure 153. Workbooks

©2026 Zscaler, Inc. All rights reserved.

5. Click **Add Workbook**.

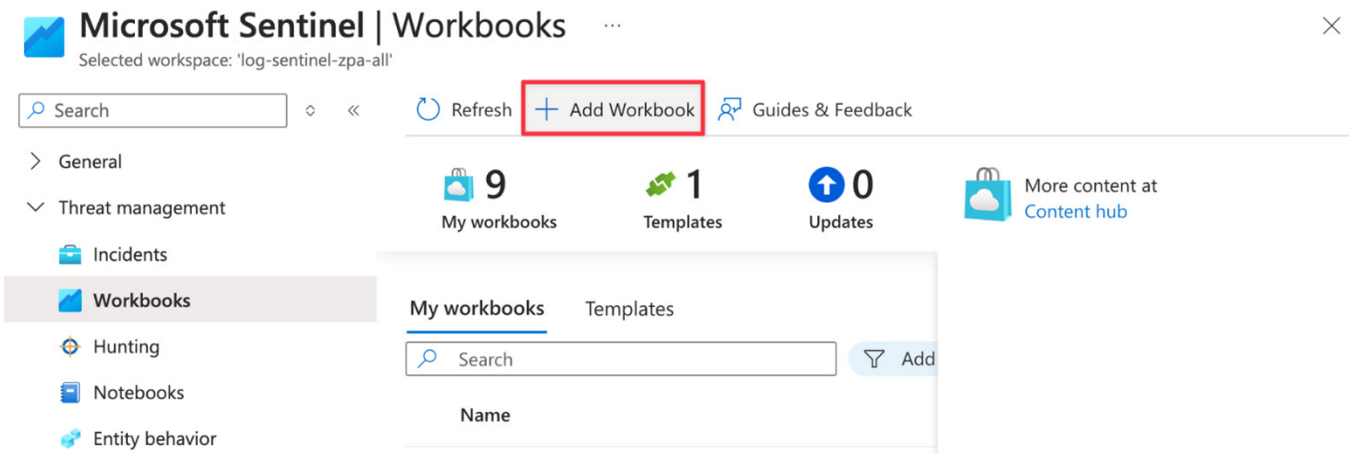


Figure 154. Add Workbook

6. Click **Edit**.



Figure 155. Edit

7. Click the **Advanced Editor** icon.

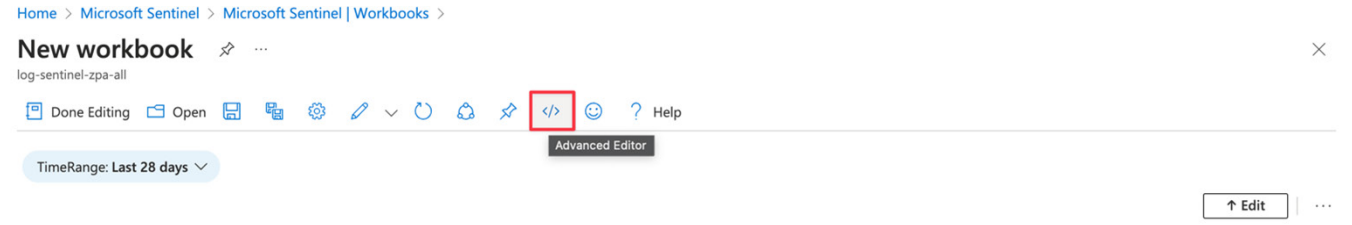


Figure 156. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook, and click **Apply**. The workbook loads the **ZPAUserActivity** function to display various graphs.

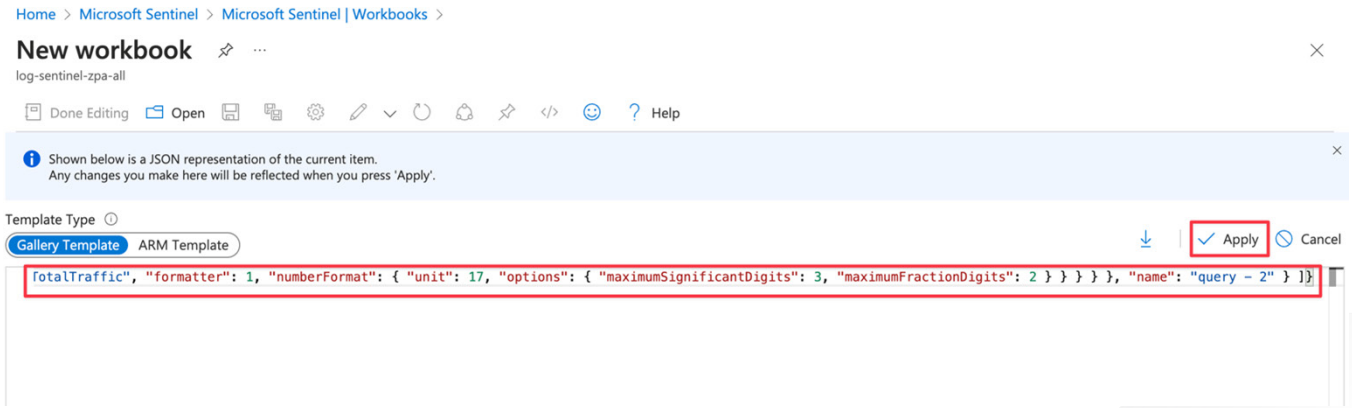


Figure 157. New code text

©2026 Zscaler, Inc. All rights reserved.

9. Click **Done Editing**.

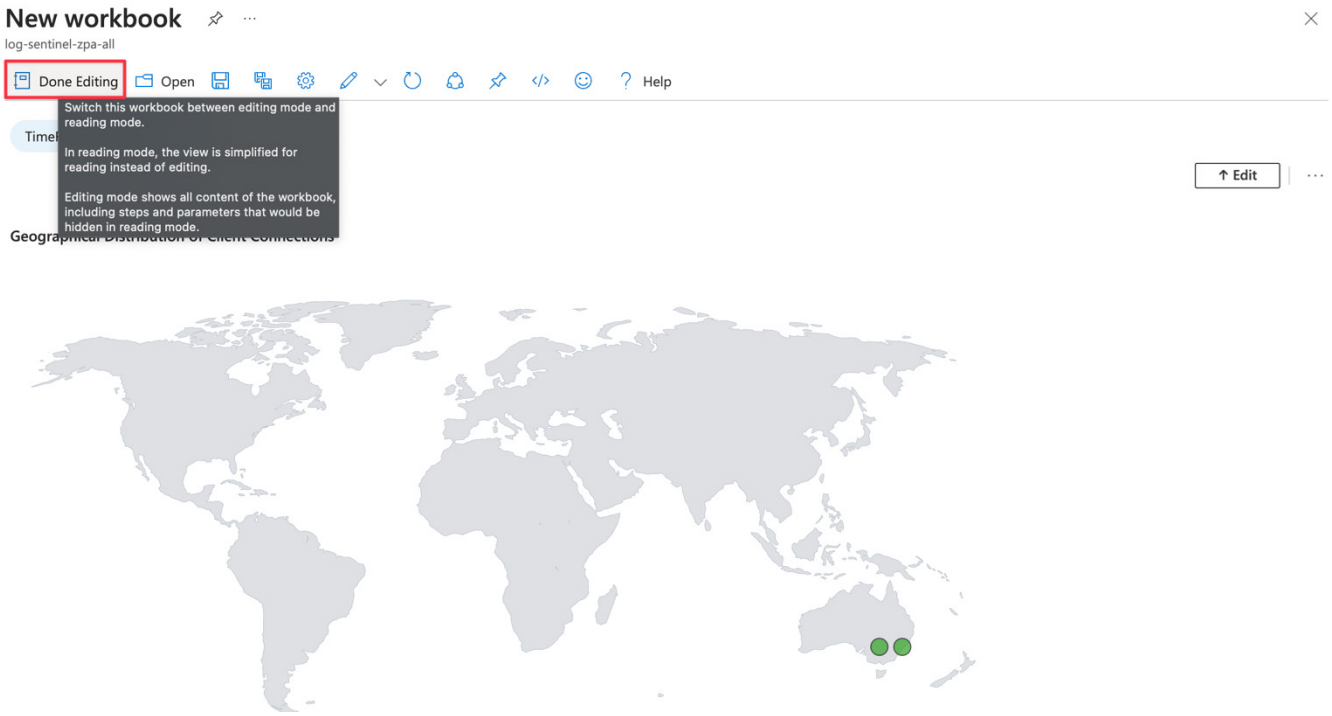


Figure 158. ZPAUserActivity Function

10. Click **Save**.



Figure 159. New workbook

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

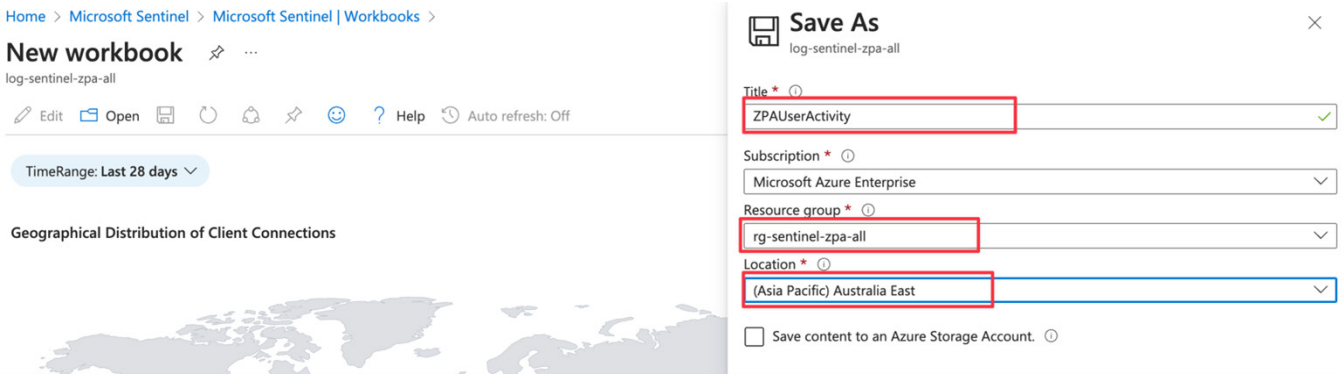


Figure 160. Save the workbook

©2026 Zscaler, Inc. All rights reserved.

Configure ZPA User Status Logs

The following section describes configuring ZPA User Status logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

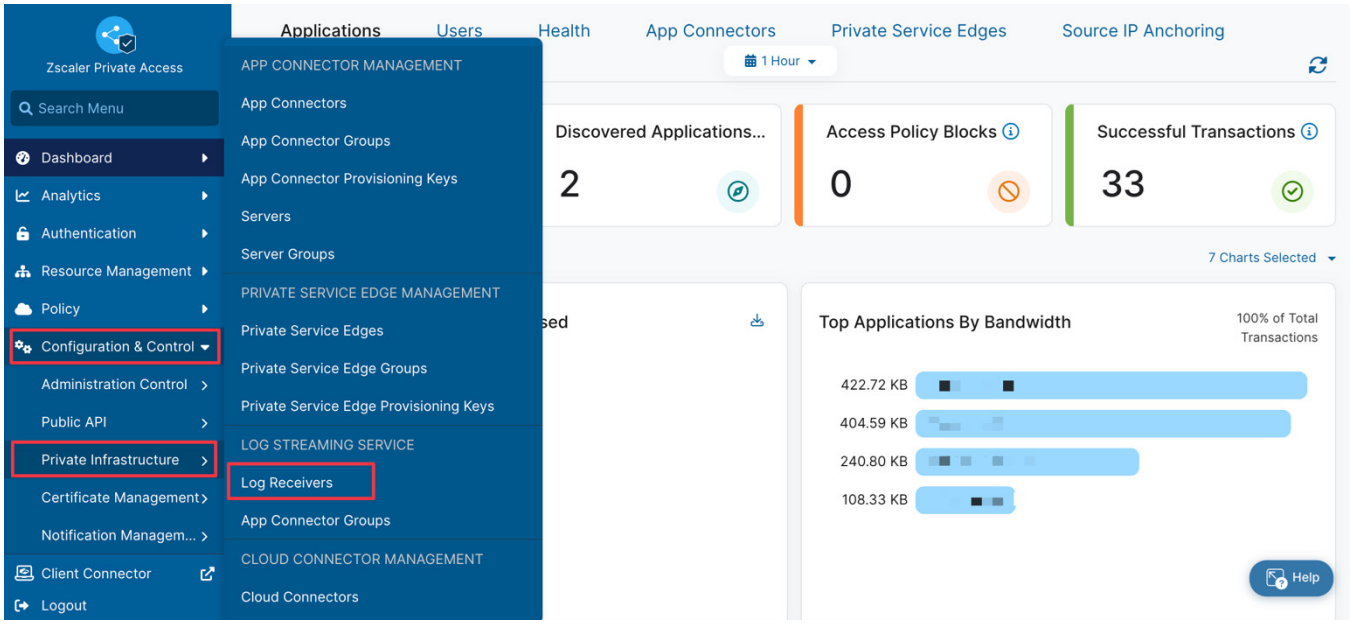


Figure 161. Log Receivers

2. Click **Add** (+) to add a new Log Receiver.

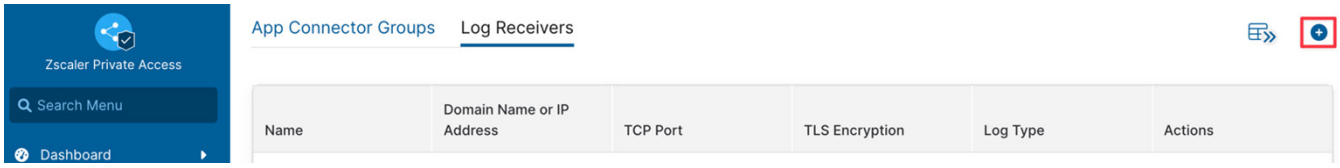


Figure 162. Add new Log Receiver

©2026 Zscaler, Inc. All rights reserved.

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Group**s to use.
4. Click **Next**.

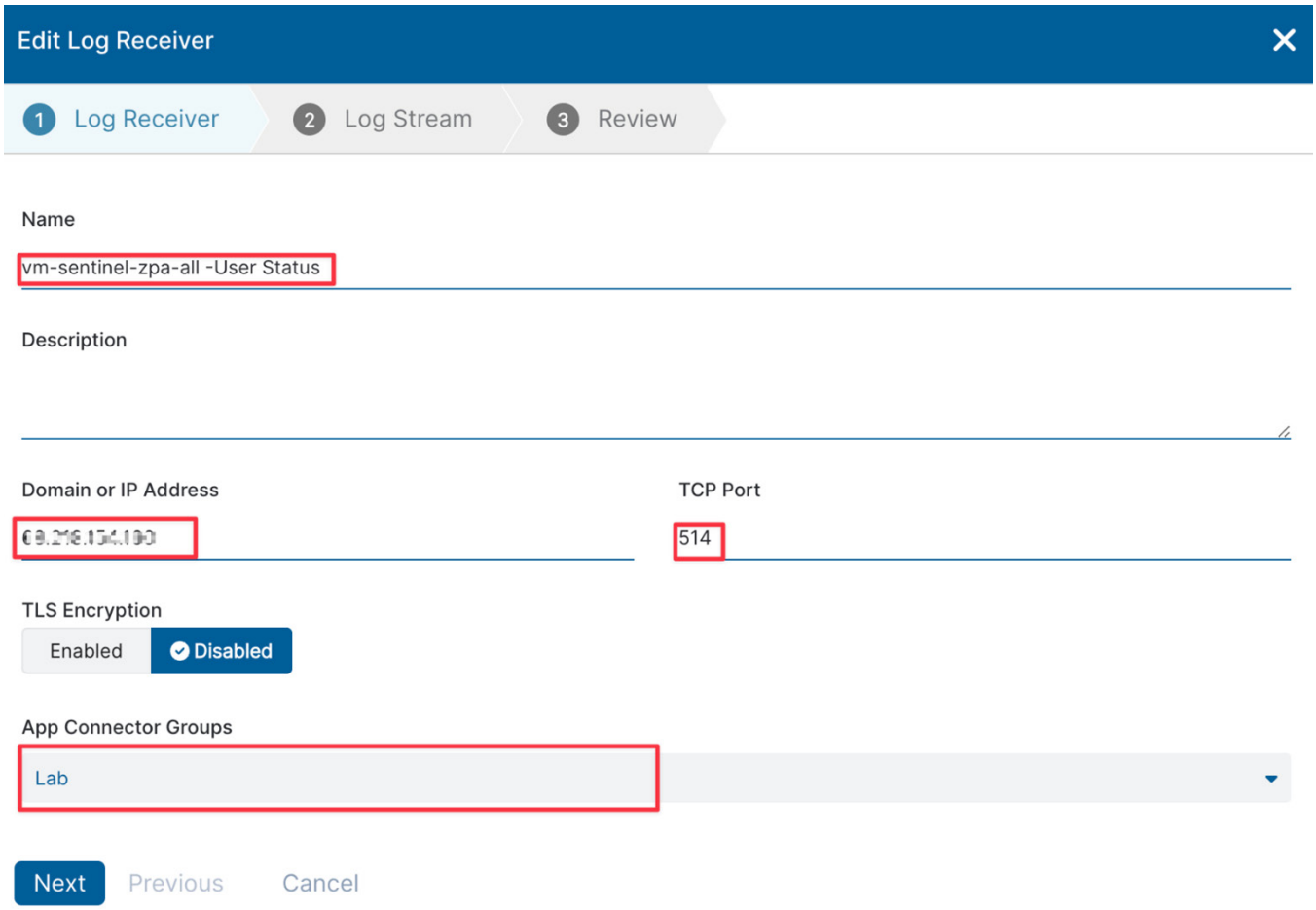


Figure 163. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

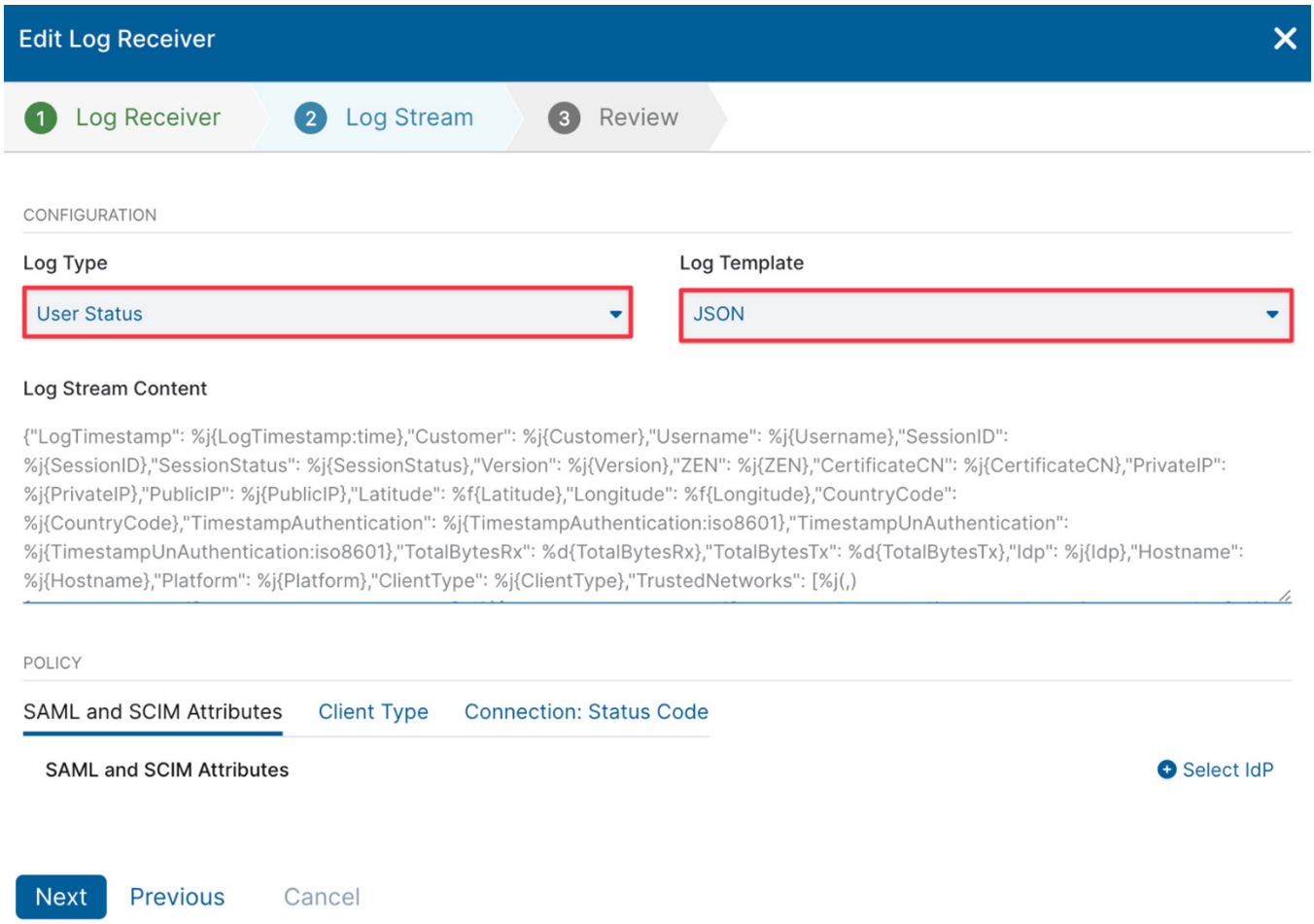


Figure 164. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

7. Click **Save**.

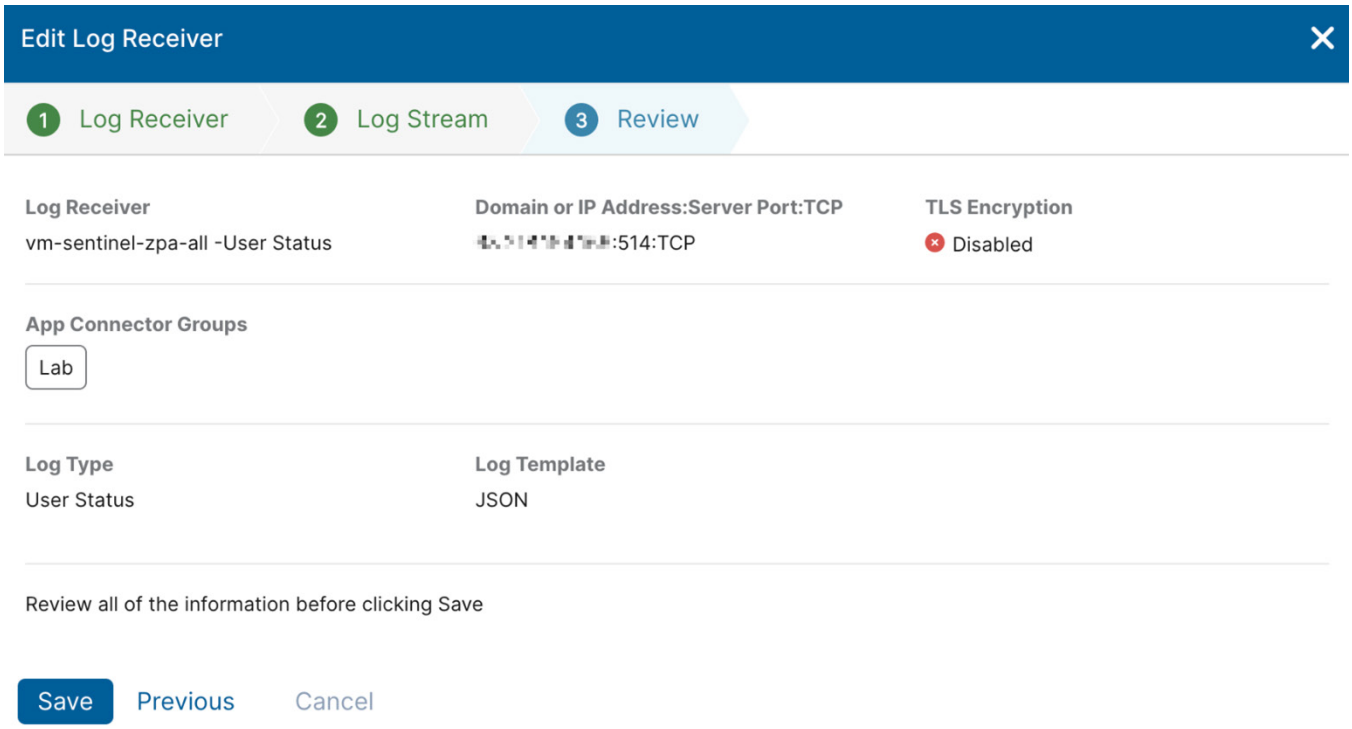


Figure 165. Save log receiver

Configure the ZPAUserStatus Log Analytics Workspace Function

The ZPAUserStatus function is used to parse the ZPA User Status Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

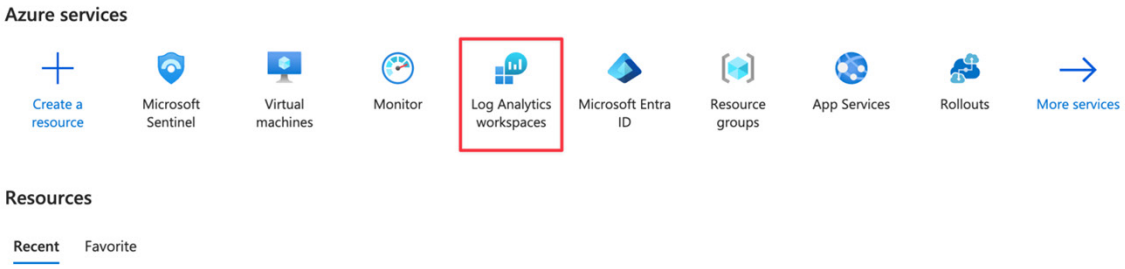


Figure 166. Log Analytics workspaces

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA User Status logs**.

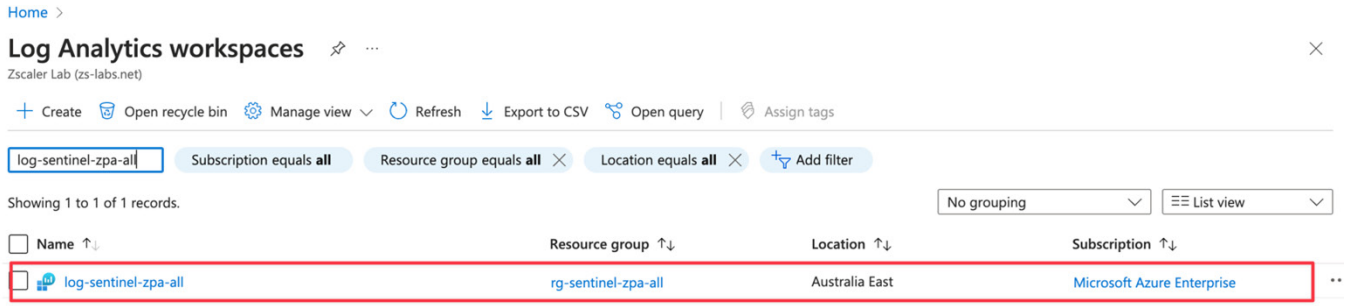


Figure 167. ZPA workspace with logs

4. Click **Logs**.

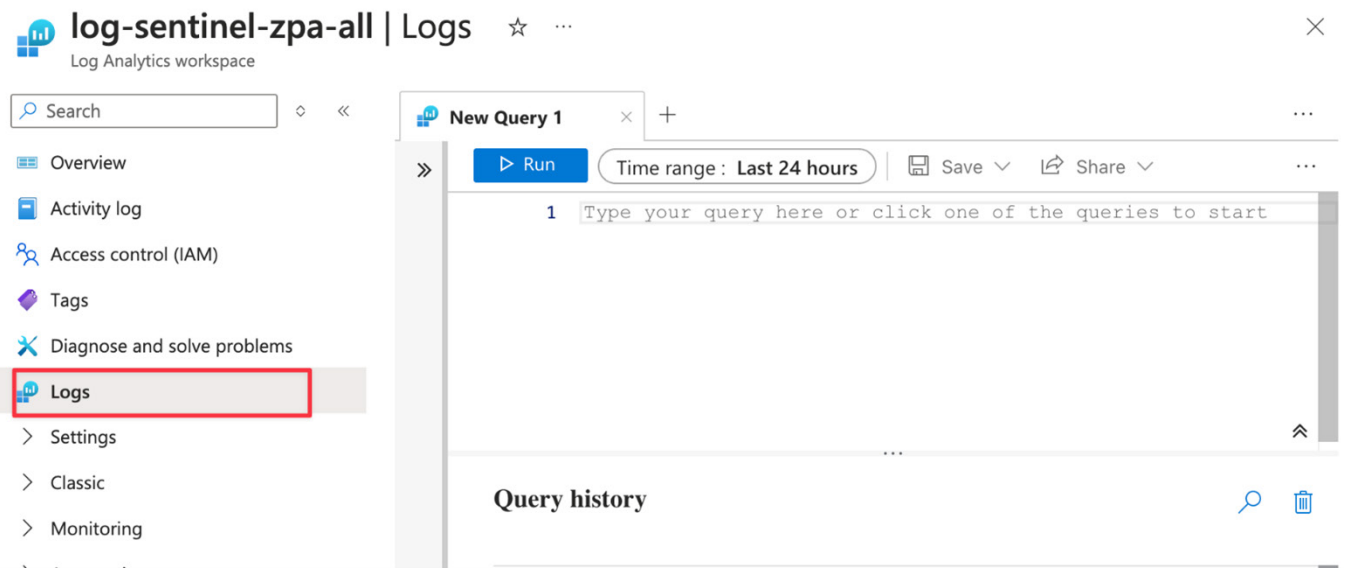


Figure 168. Logs

5. Paste the content from the `ZPAUserStatus.kql` file into the query window, and select **Save** > **Save as function**.

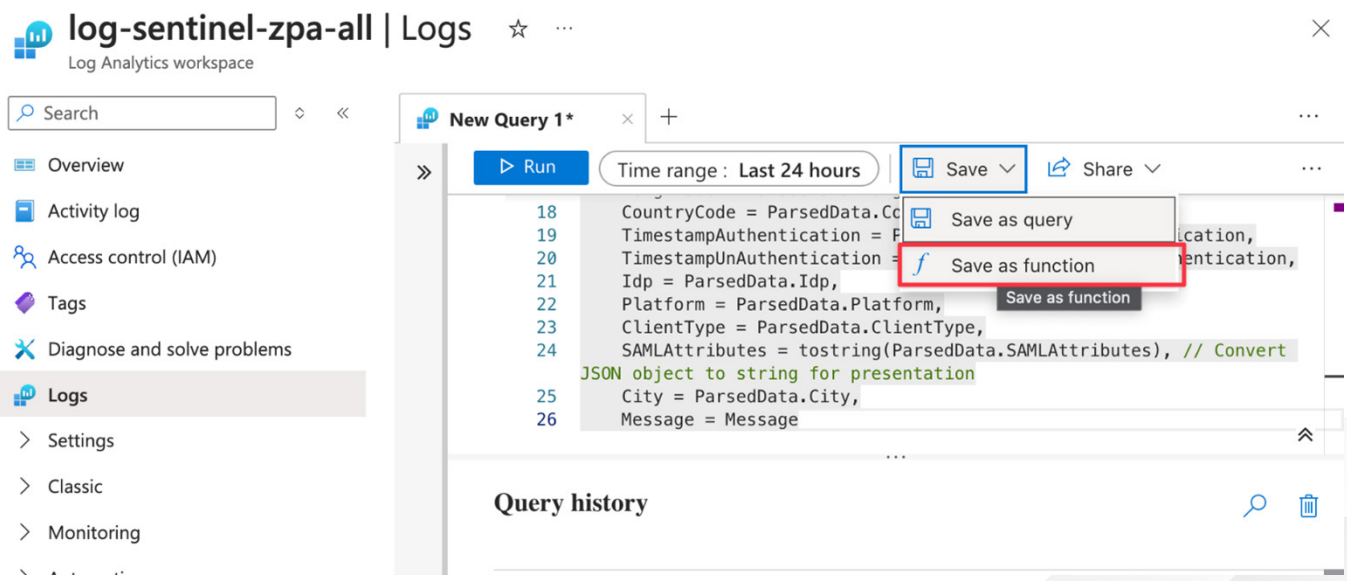


Figure 169. ZPAUserStatus.kql content

©2026 Zscaler, Inc. All rights reserved.

6. Enter the **Function name** `ZPAUserStatus`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

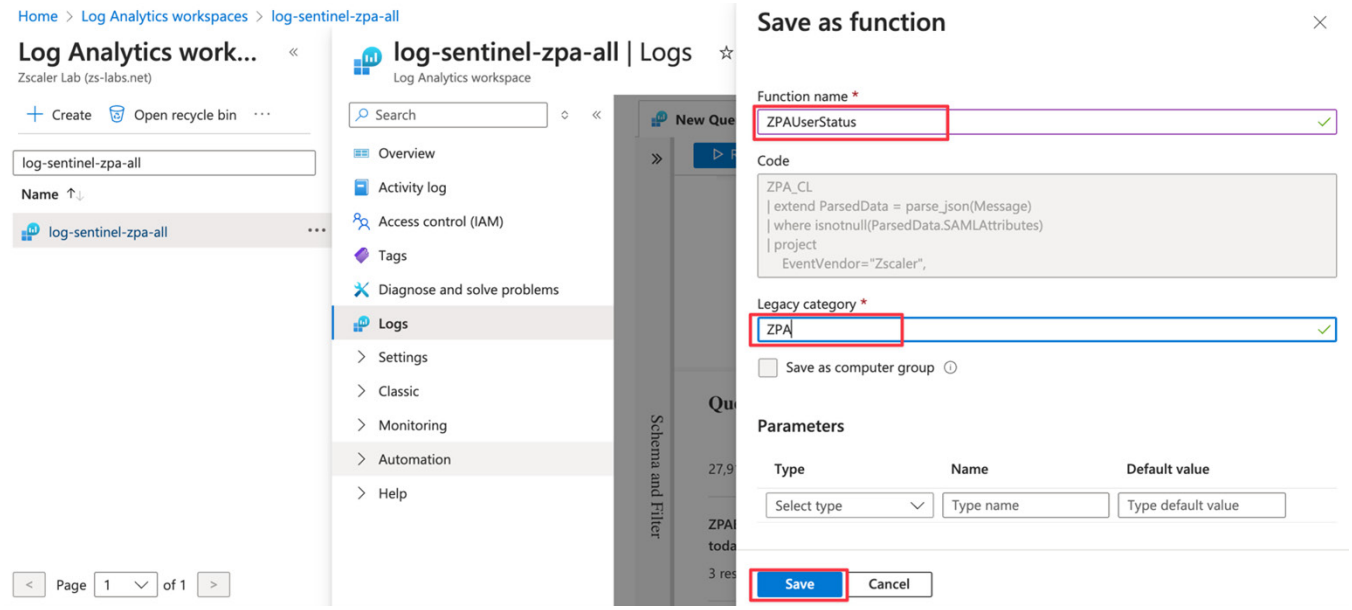


Figure 170. Save ZPAUserStatus

8. Run queries using the **ZPAUserStatus** function.

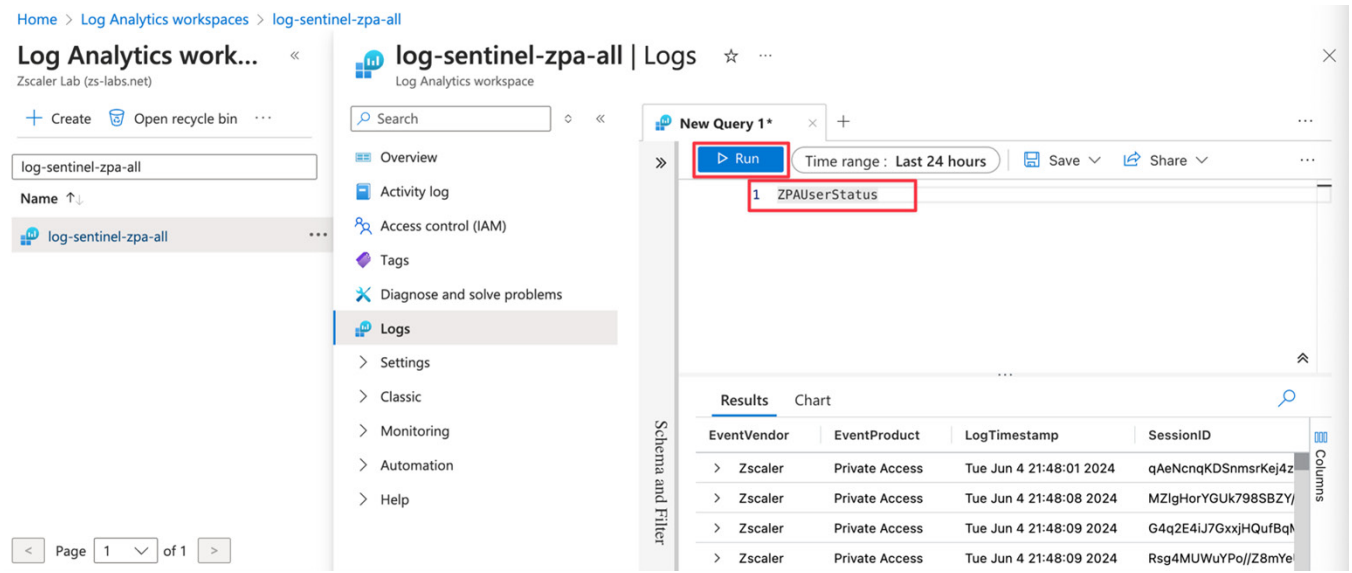


Figure 171. Run queries

©2026 Zscaler, Inc. All rights reserved.

Configure the ZPAUserStatus Sentinel Workbook

The ZPAUserStatus workbook is used to graphically display information related to the ZPA User Status Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your ZPAUserStatus parser.

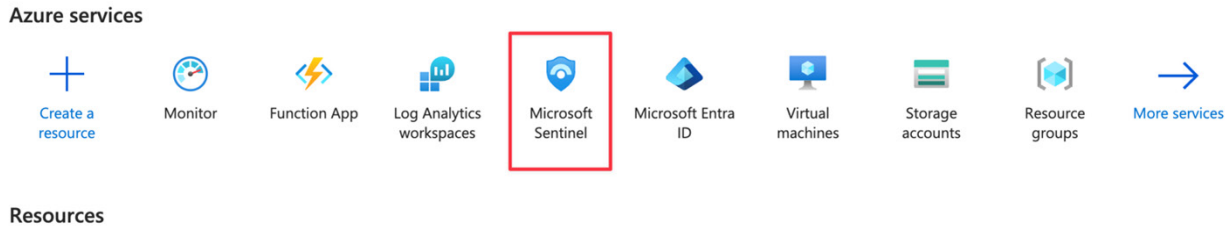


Figure 172. Microsoft Sentinel

3. Select your workspace with the **ZPA User Status logs**.

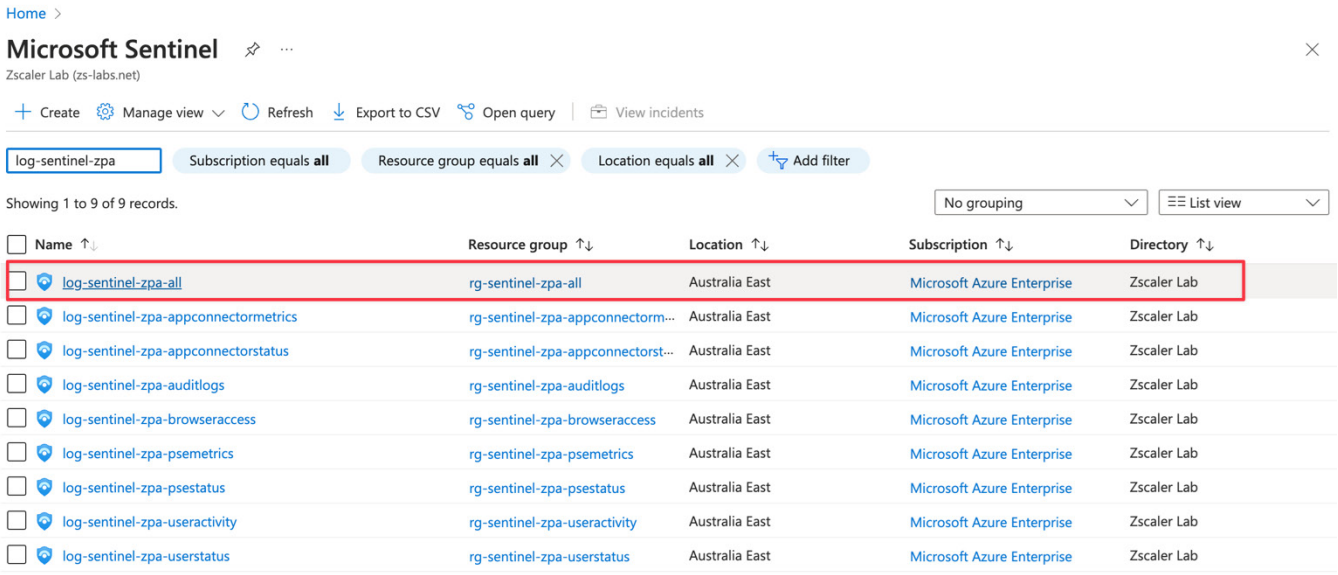


Figure 173. ZPA workspace with logs

©2026 Zscaler, Inc. All rights reserved.

4. Click **Workbooks** under **Threat management**.

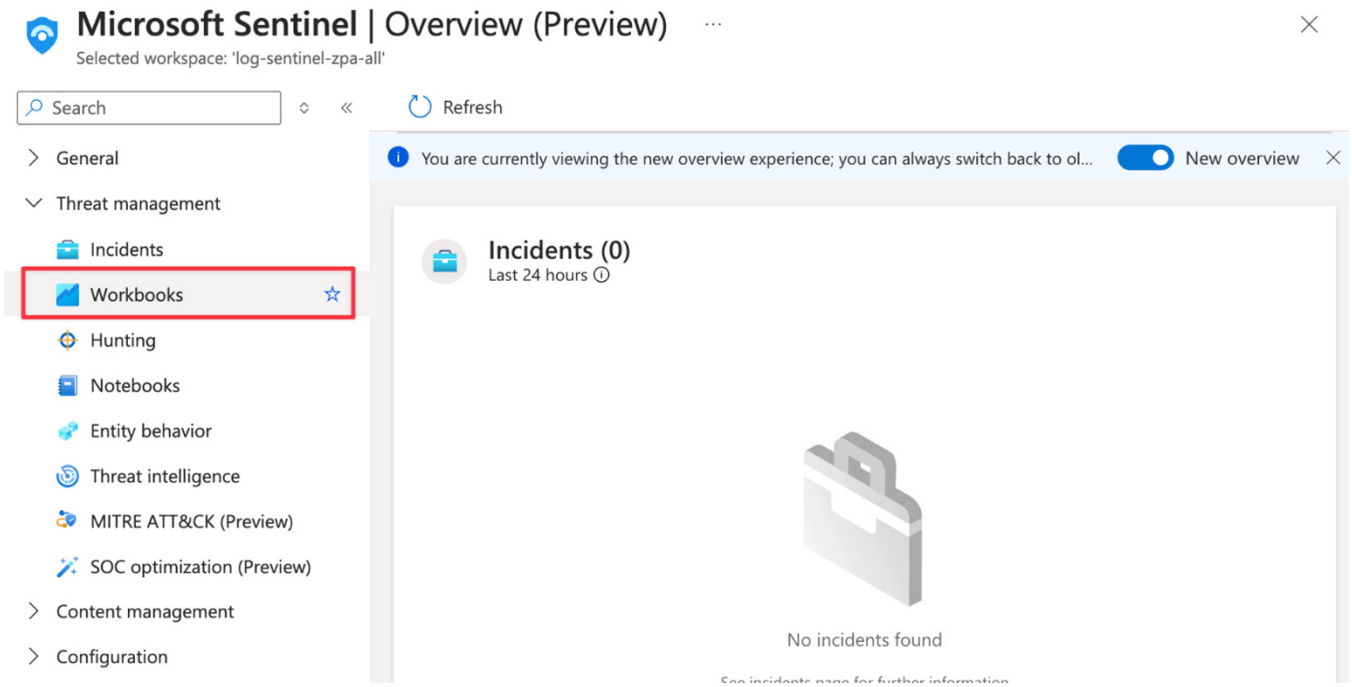


Figure 174. Workbooks

5. Click **Add Workbook**.

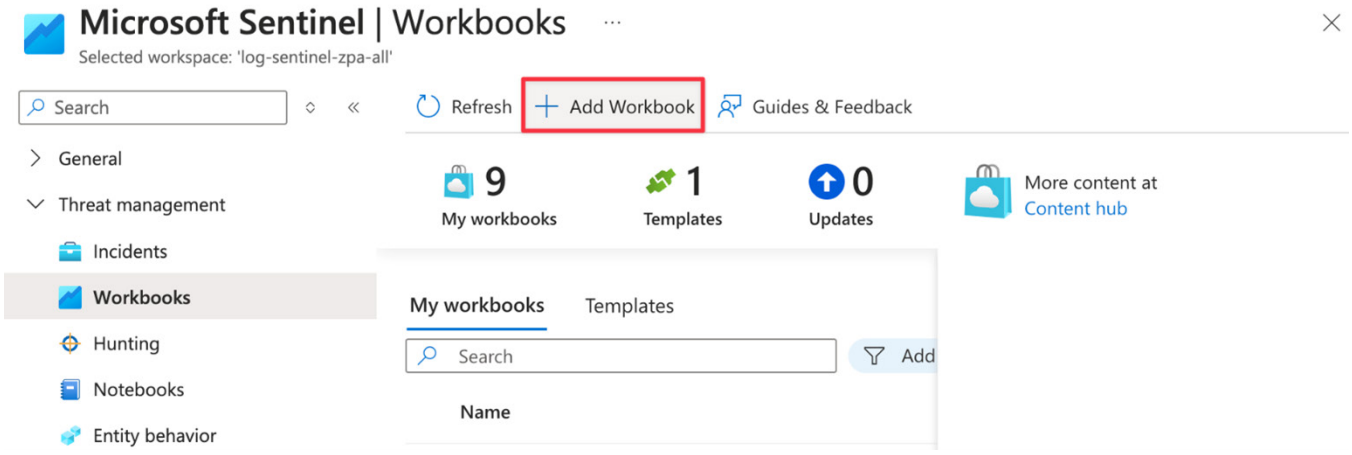


Figure 175. Add Workbook

6. Click **Edit**.



Figure 176. Edit

©2026 Zscaler, Inc. All rights reserved.

7. Click the **Advanced Editor** icon.

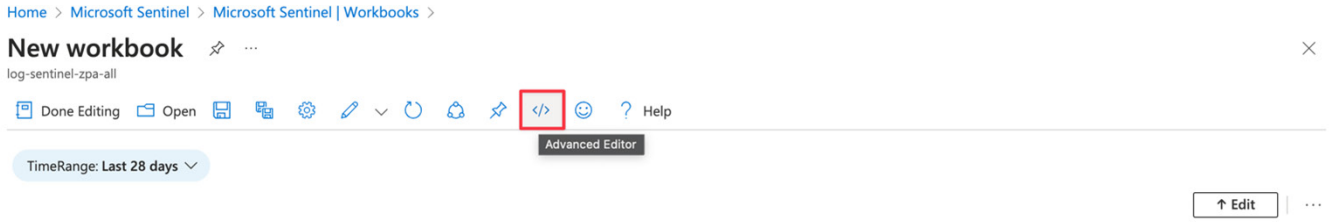


Figure 177. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPAUserStatus** function to display various graphs.

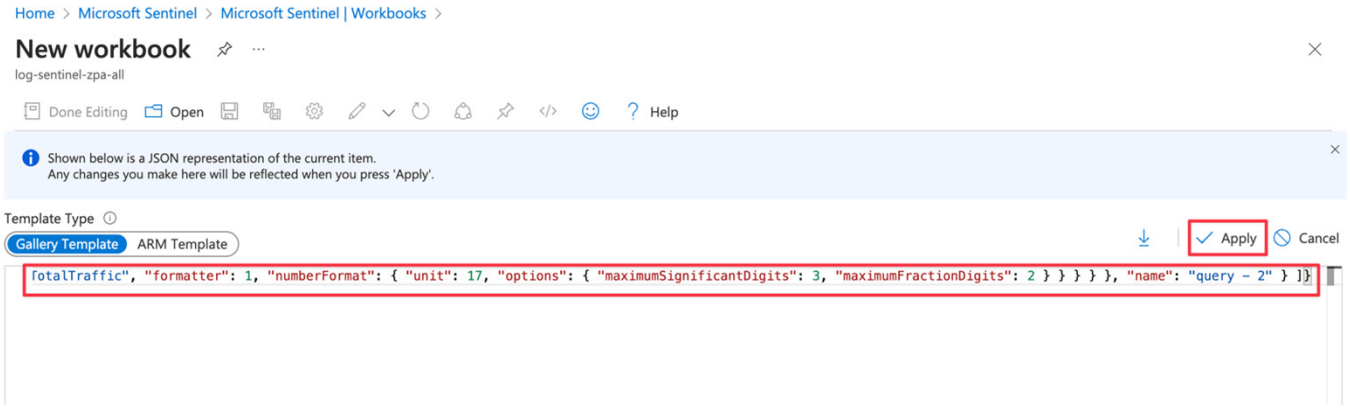


Figure 178. Replace code

©2026 Zscaler, Inc. All rights reserved.

9. Click **Done Editing**.

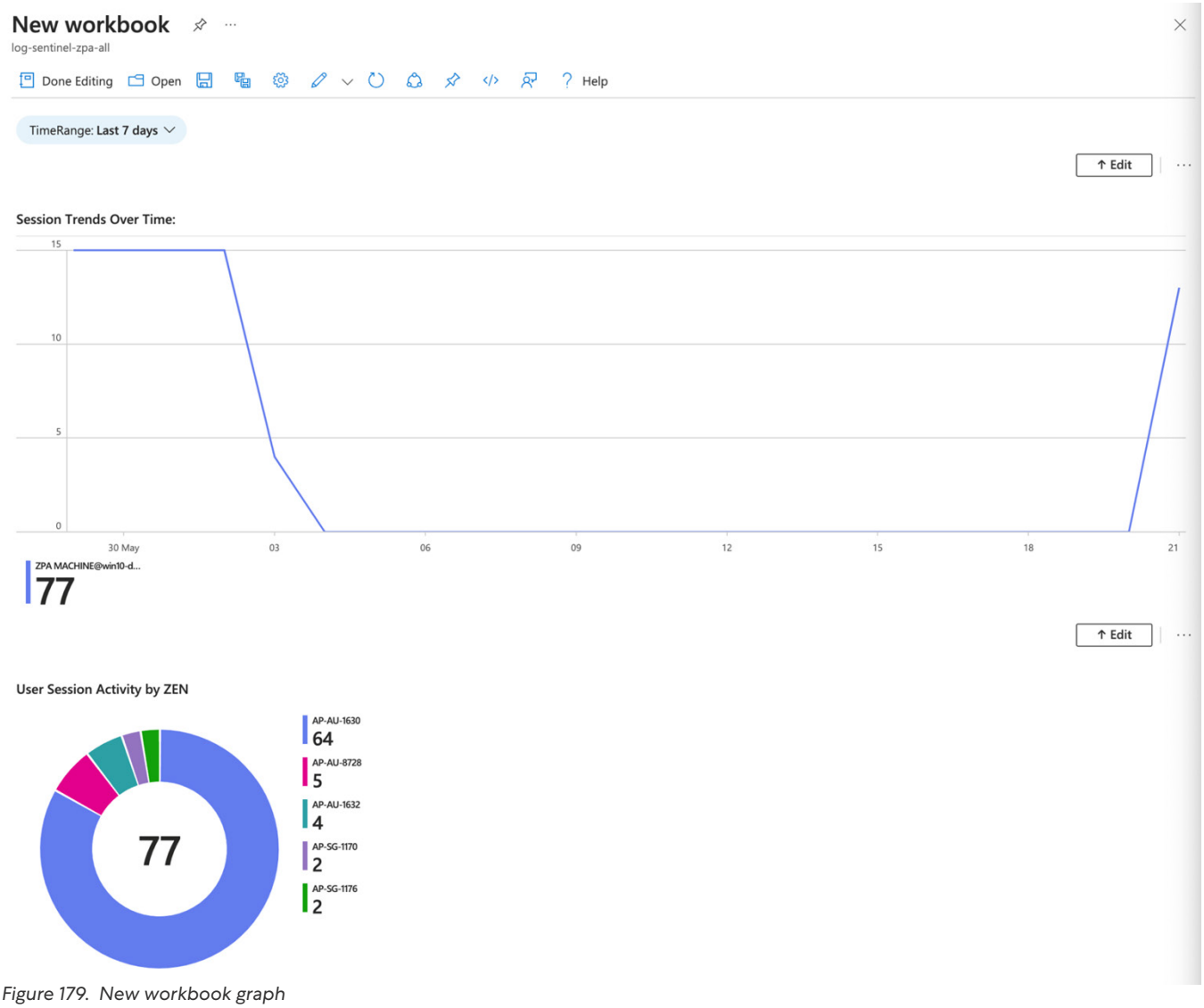


Figure 179. New workbook graph

10. Click **Save**.



Figure 180. Save the workbook

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

©2026 Zscaler, Inc. All rights reserved.

Configure ZPA Private Service Edge Status Logs

The following sections describe how to configure ZPA Private Service Edge Status logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

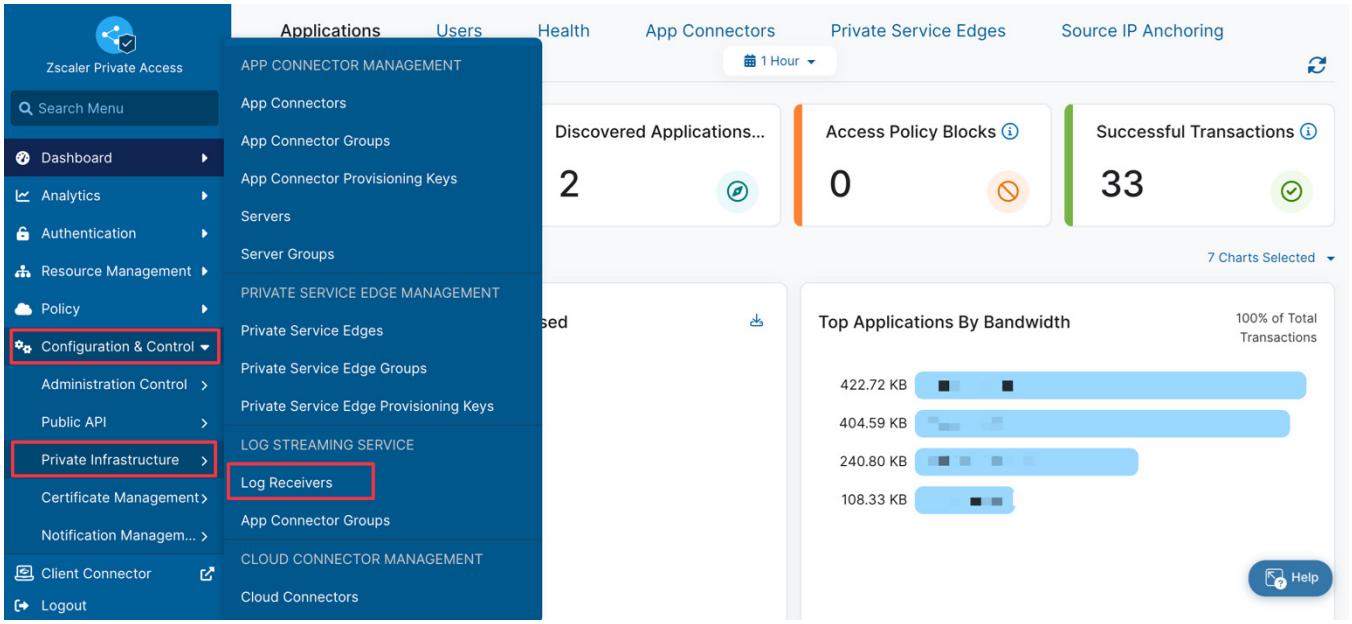


Figure 181. Log Receivers

2. Click **Add** (+) to add a new Log Receiver.

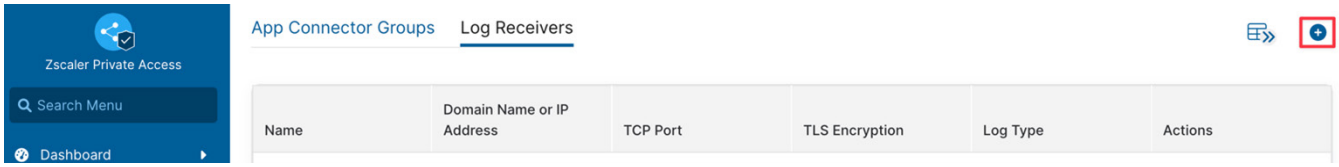


Figure 182. Add new Log Receiver

©2026 Zscaler, Inc. All rights reserved.

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

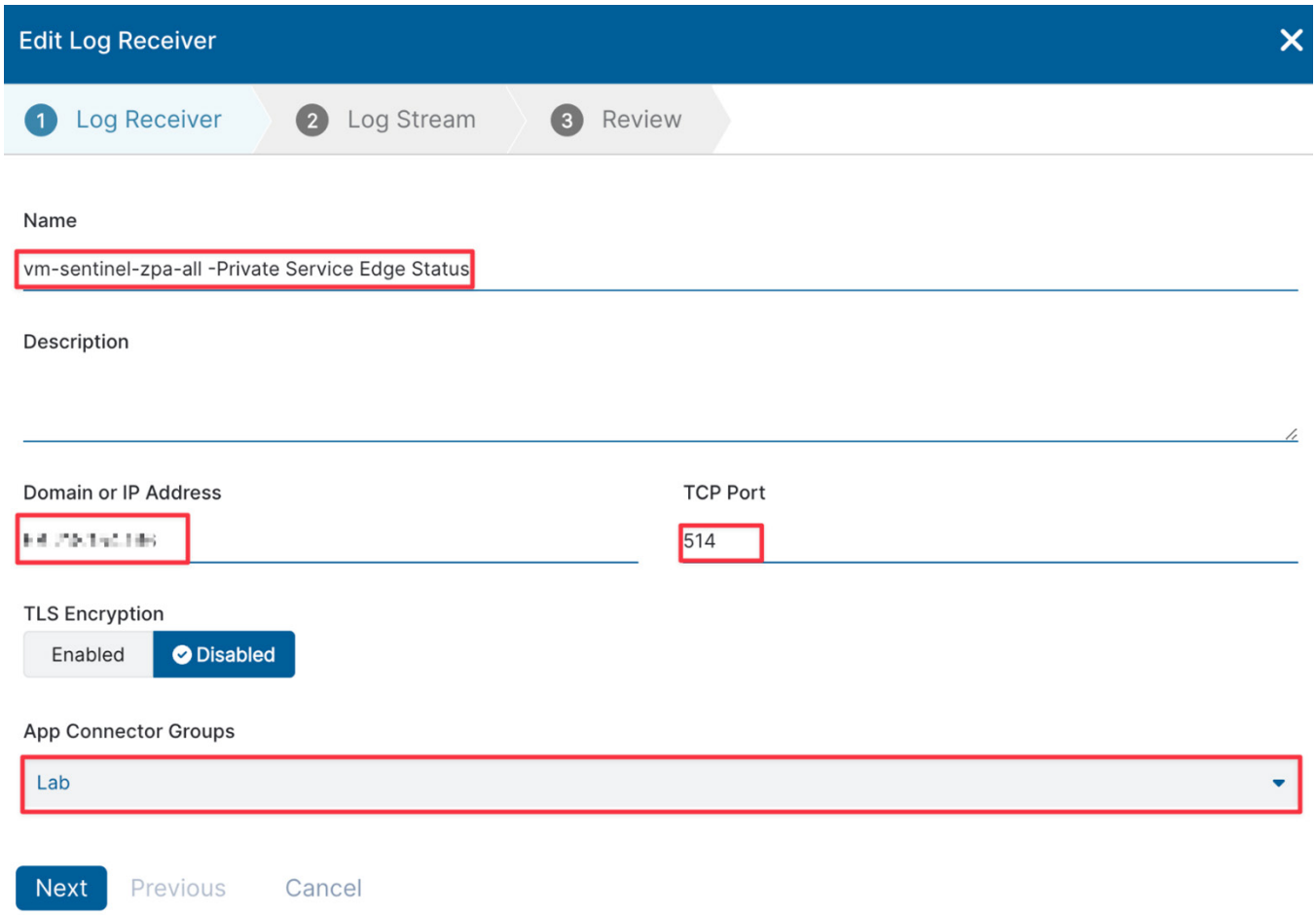


Figure 183. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

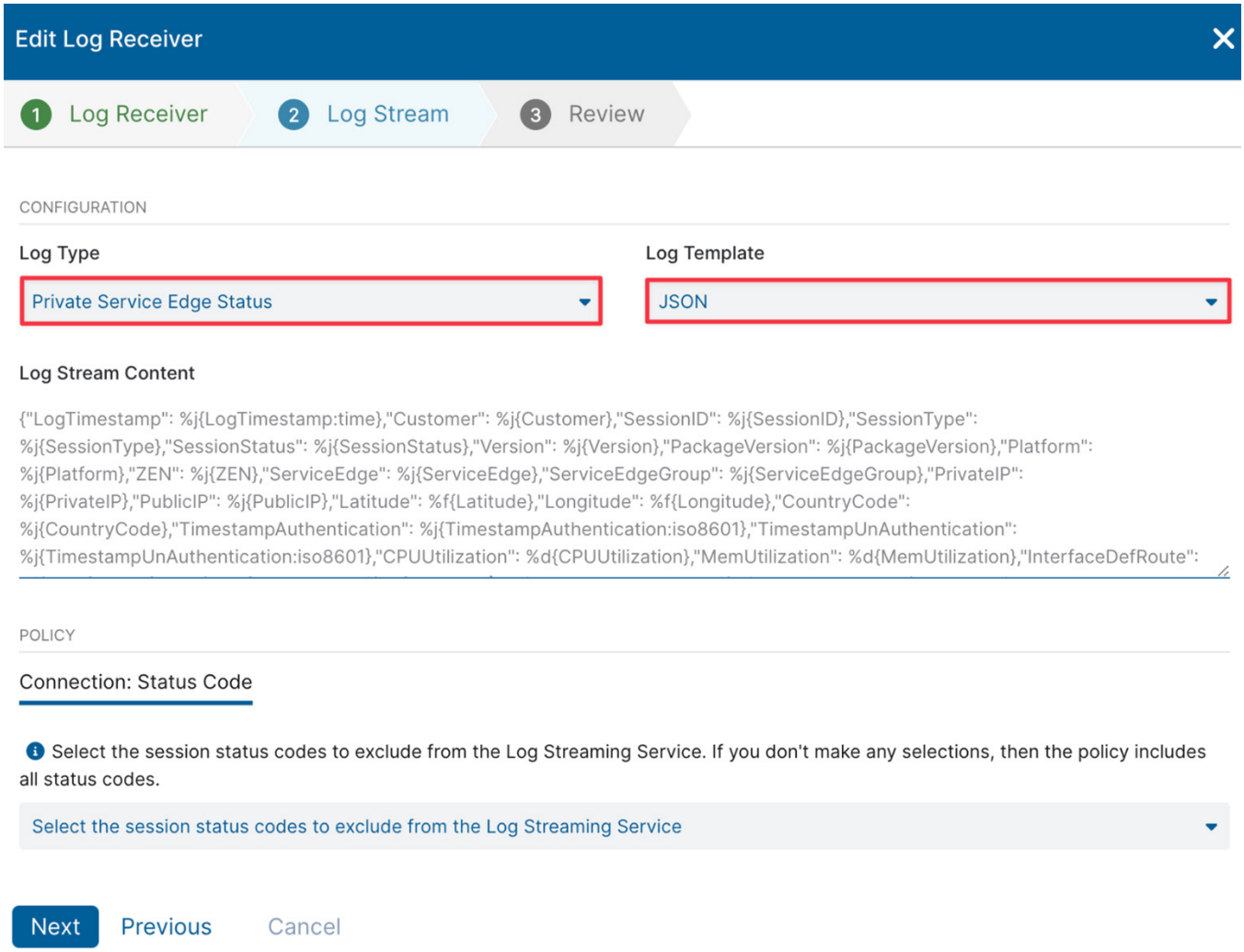


Figure 184. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

7. Click **Save**.

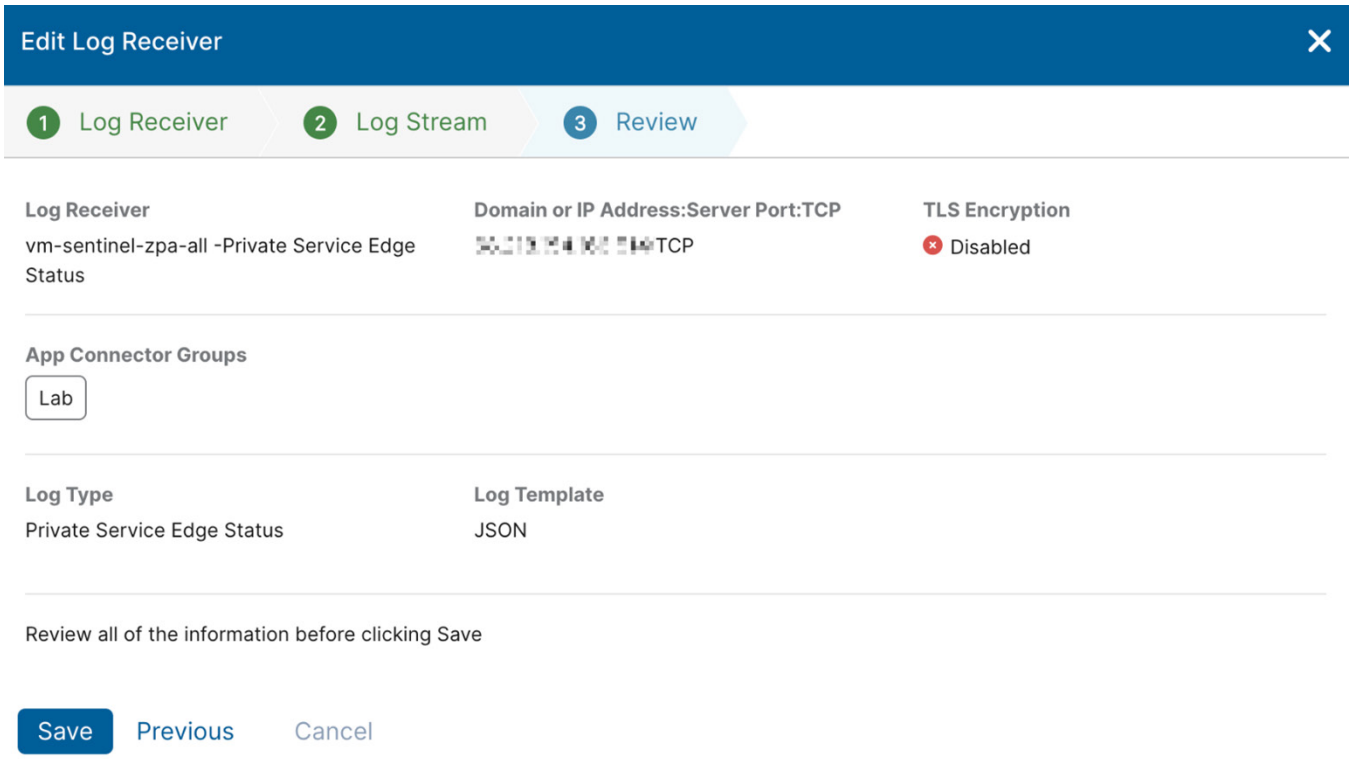


Figure 185. Save log receiver

Configure the ZPAPSEStatus Log Analytics Workspace Function

The ZPAPSEStatus function is used to parse the ZPA Private Service Edge Status Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

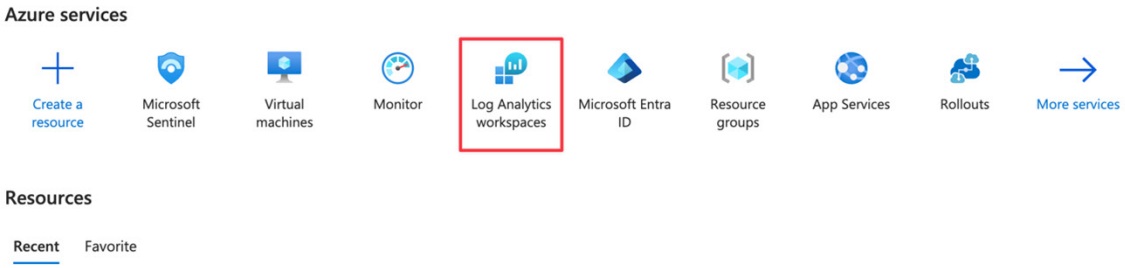


Figure 186. Log Analytics workspaces

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA Private Service Edge Status logs**.

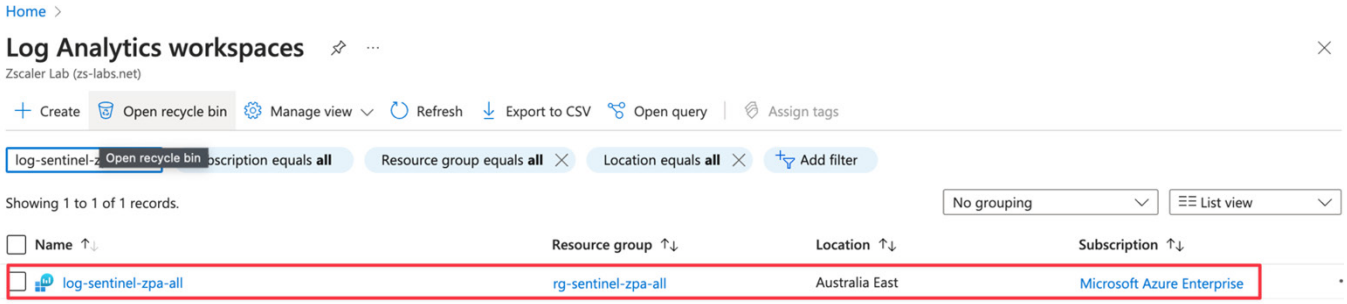


Figure 187. ZPA workspace with logs

4. Click **Logs**.

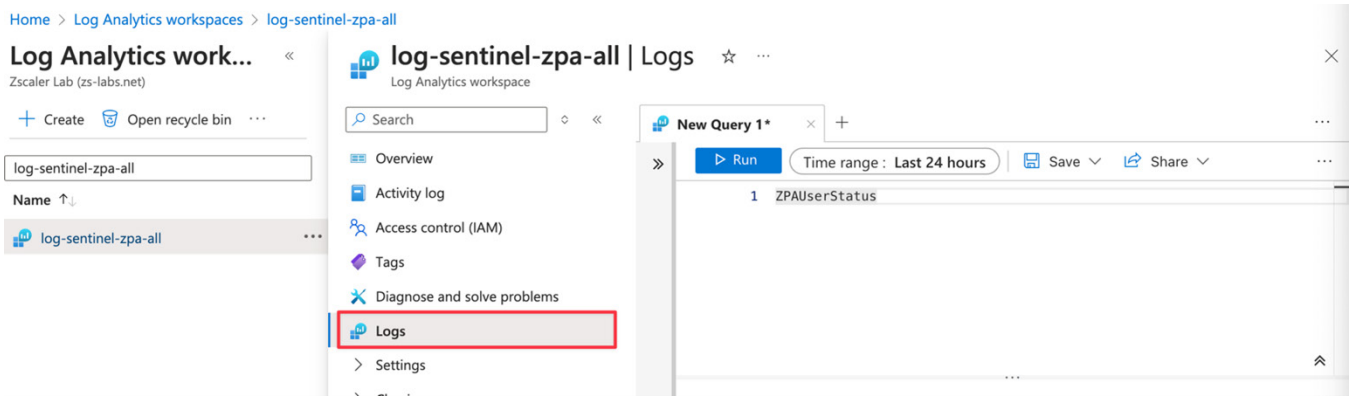


Figure 188. Logs

5. Paste the content from the `ZPAPSEStatus.kql` file into the query window, and select **Save** > **Save as function**.

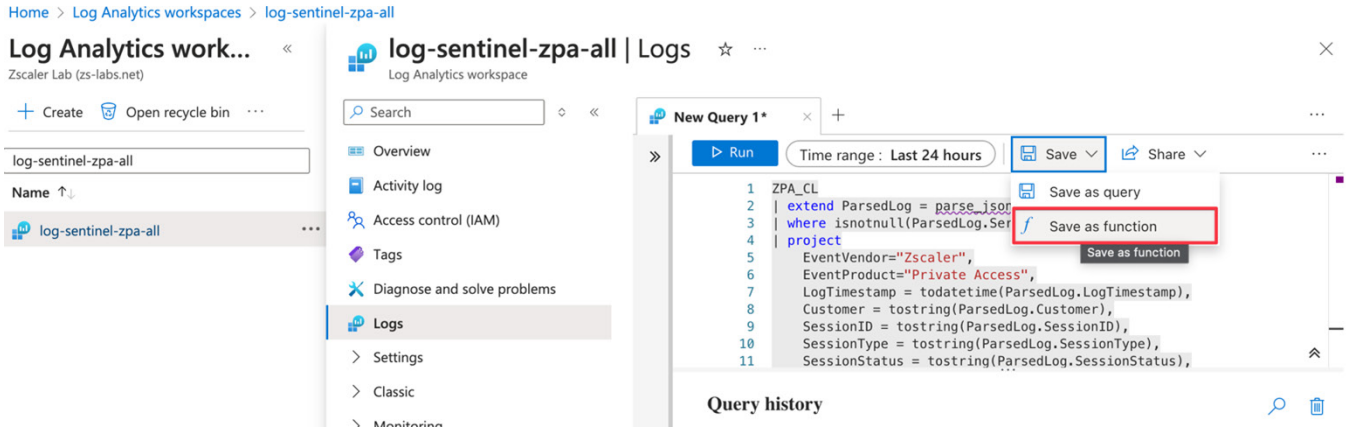


Figure 189. ZPAPSEStatus.kql content

©2026 Zscaler, Inc. All rights reserved.

6. Enter the **Function name** `ZPAPSEStatus`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

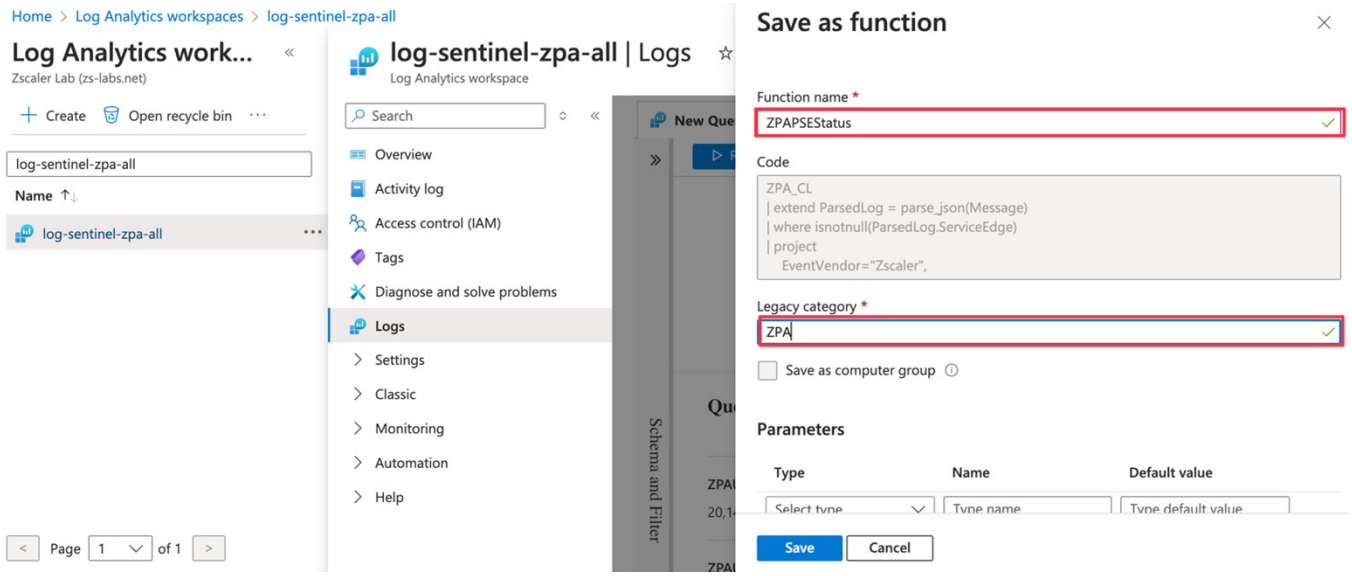


Figure 190. Save ZPAPSEStatus

8. Run queries using the **ZPAPSEStatus** function.

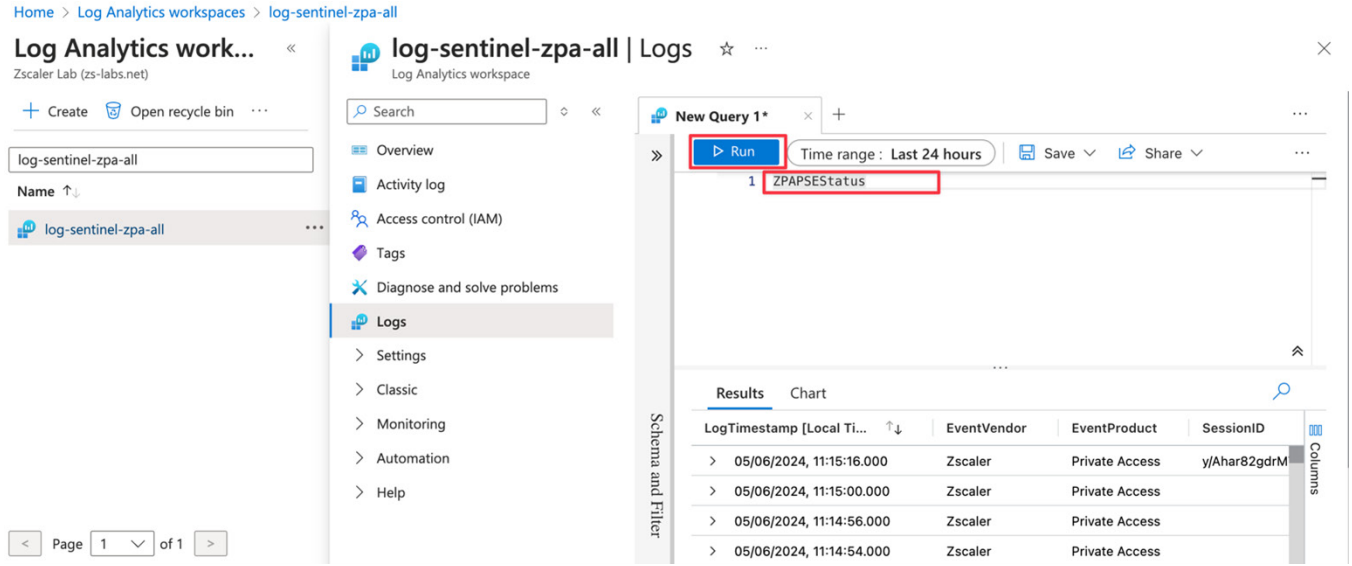


Figure 191. Run queries

©2026 Zscaler, Inc. All rights reserved.

Configure the ZPAPSEStatus Sentinel Workbook

The ZPAPSEStatus workbook is used to graphically display information related to the ZPA User Status Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your ZPAPSEStatus parser.

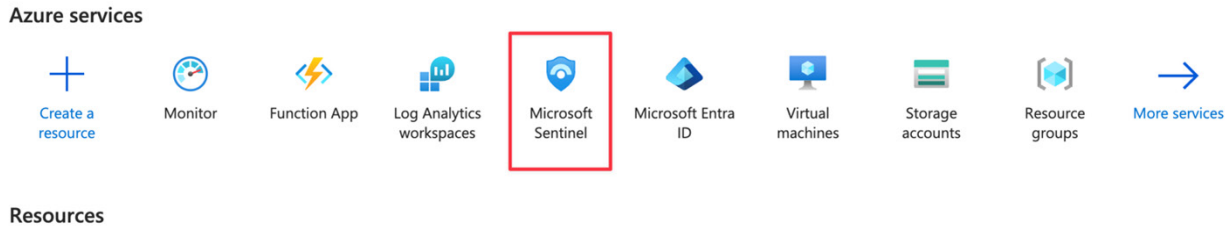


Figure 192. Microsoft Sentinel

3. Select your workspace with the **ZPA Private Service Edge Status logs**.

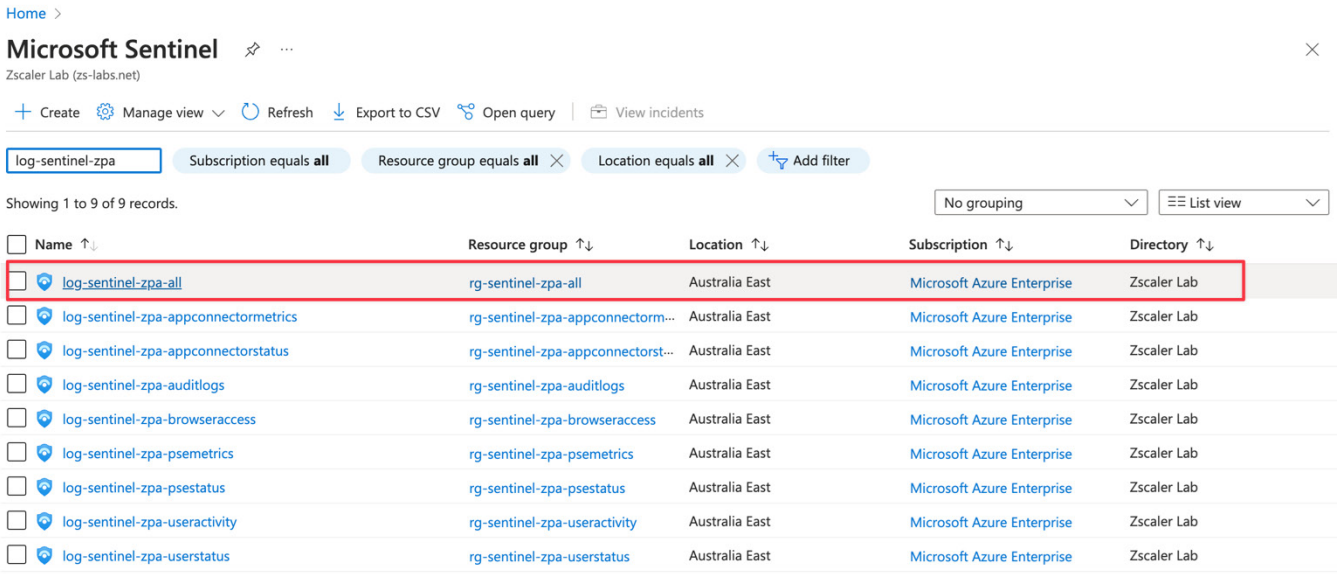


Figure 193. ZPA workspace with logs

©2026 Zscaler, Inc. All rights reserved.

4. Click **Workbooks** under **Threat management**.

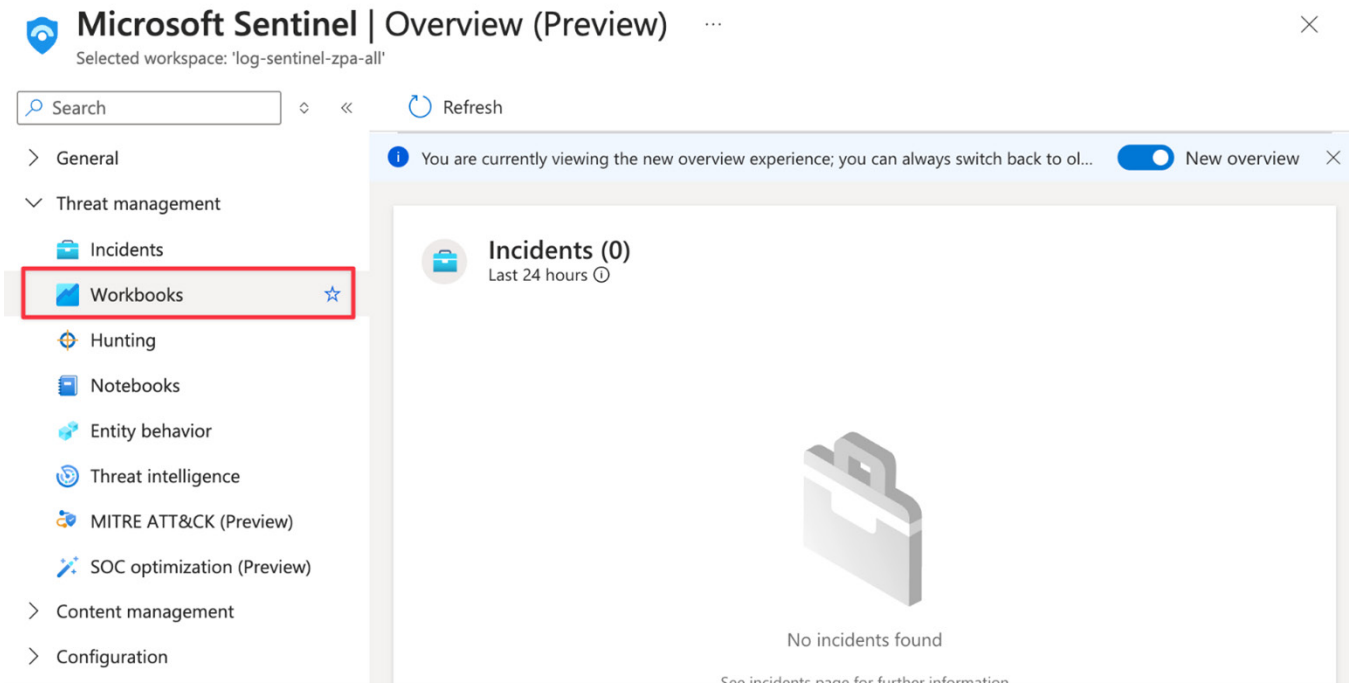


Figure 194. Workbooks

5. Click **Add Workbook**.

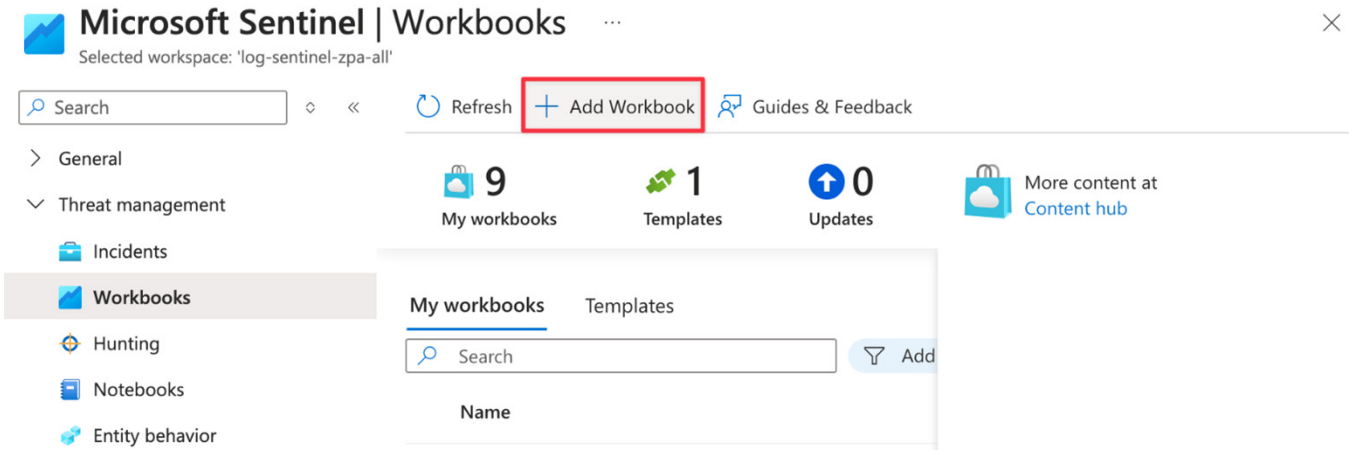


Figure 195. Add Workbook

6. Click **Edit**.



Figure 196. Edit

©2026 Zscaler, Inc. All rights reserved.

7. Click the **Advanced Editor** icon.

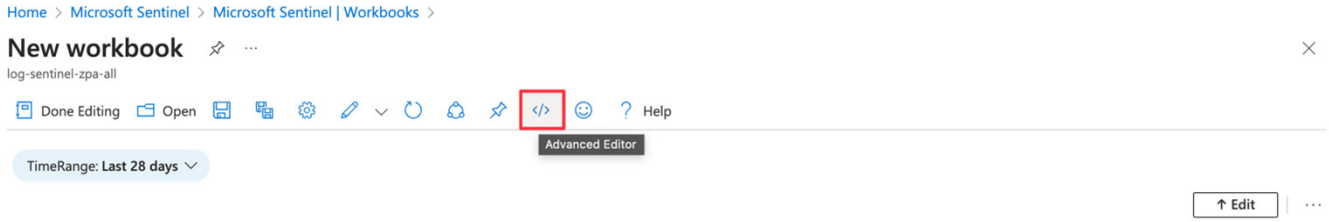


Figure 197. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPAPSEStatus** function to display various graphs.

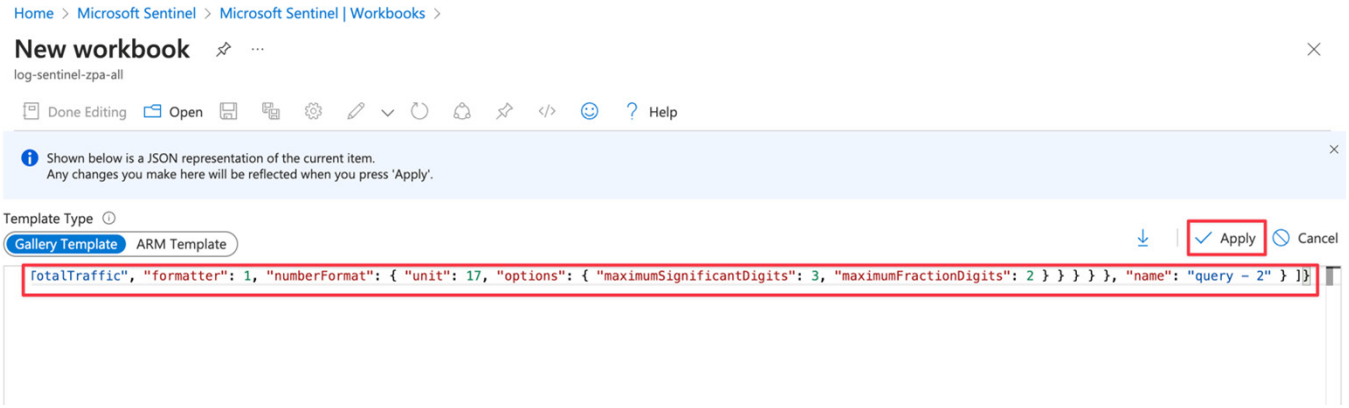


Figure 198. New code

9. Click **Done Editing**.



Figure 199. Graph

10. Click **Save**.

©2026 Zscaler, Inc. All rights reserved.



Figure 200. New workbook

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

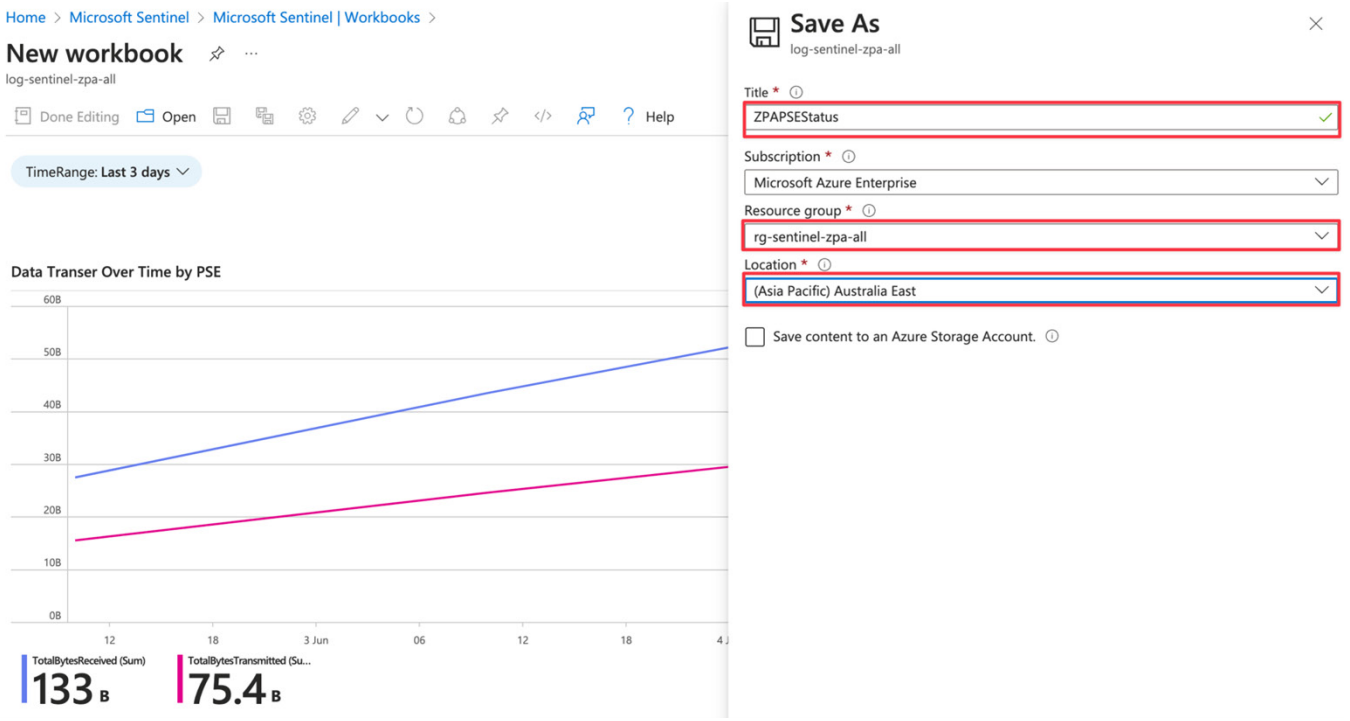


Figure 201. Save the workbook

©2026 Zscaler, Inc. All rights reserved.

Configure ZPA Private Service Edge Metrics Logs

The following sections describe configuring ZPA Private Service Edge Metric logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

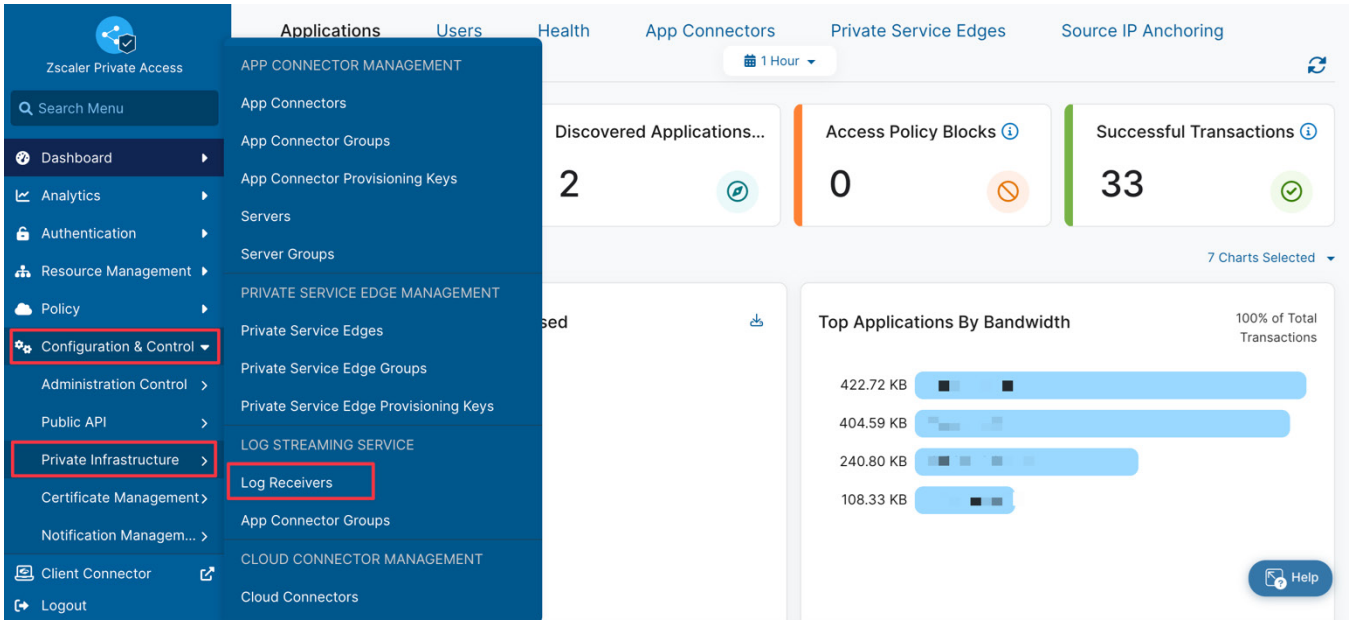


Figure 202. Log Receivers

2. Click **Add** (+) to add a new Log Receiver.

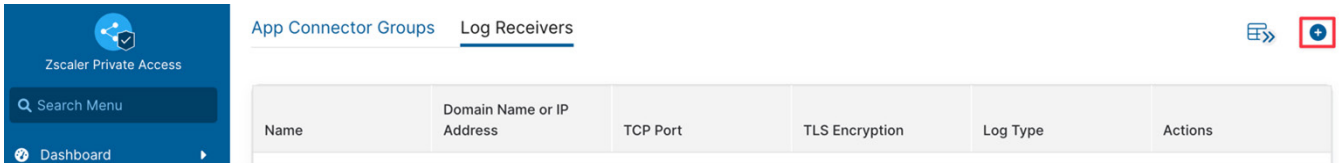


Figure 203. Add new Log Receiver

©2026 Zscaler, Inc. All rights reserved.

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

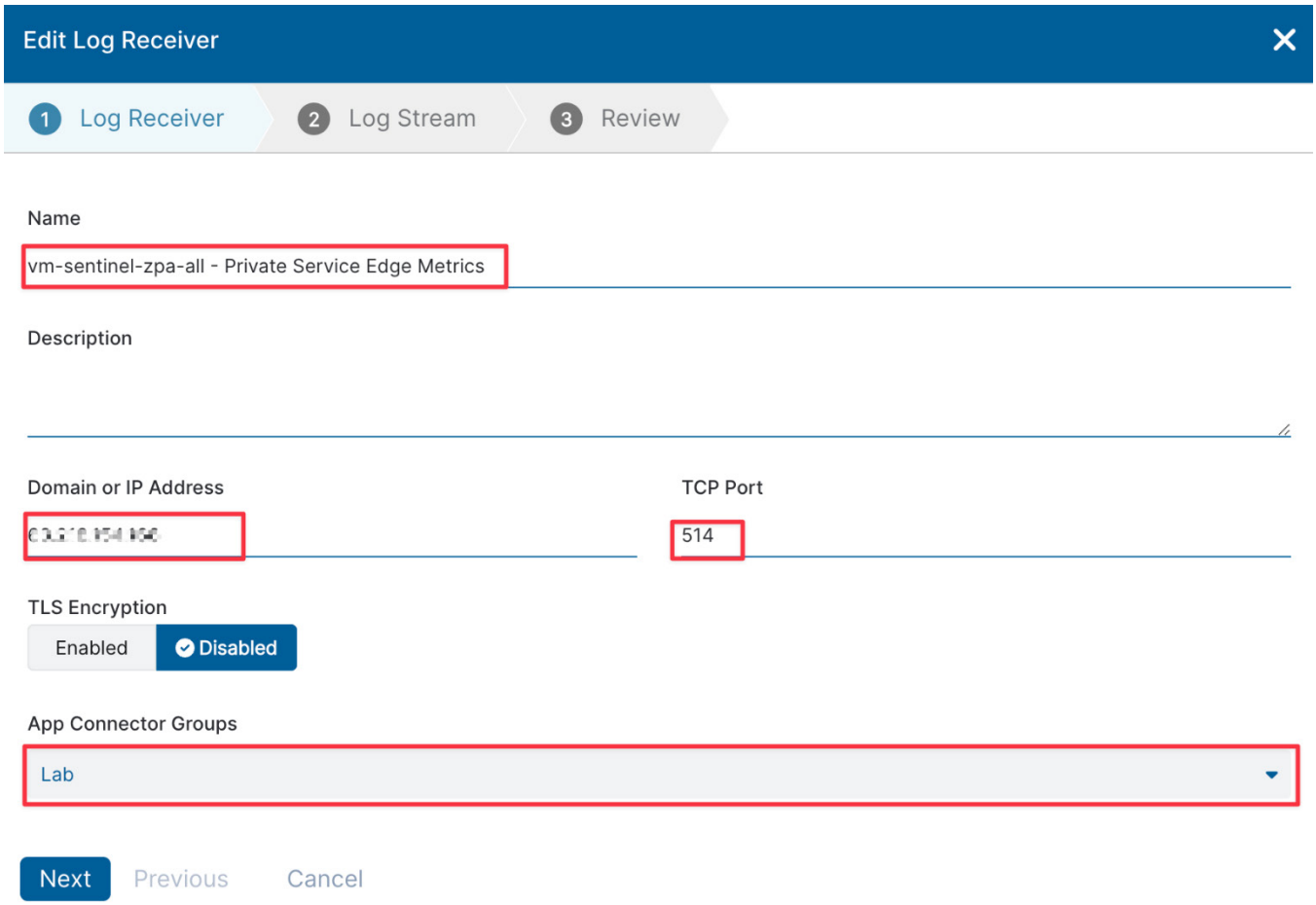


Figure 204. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

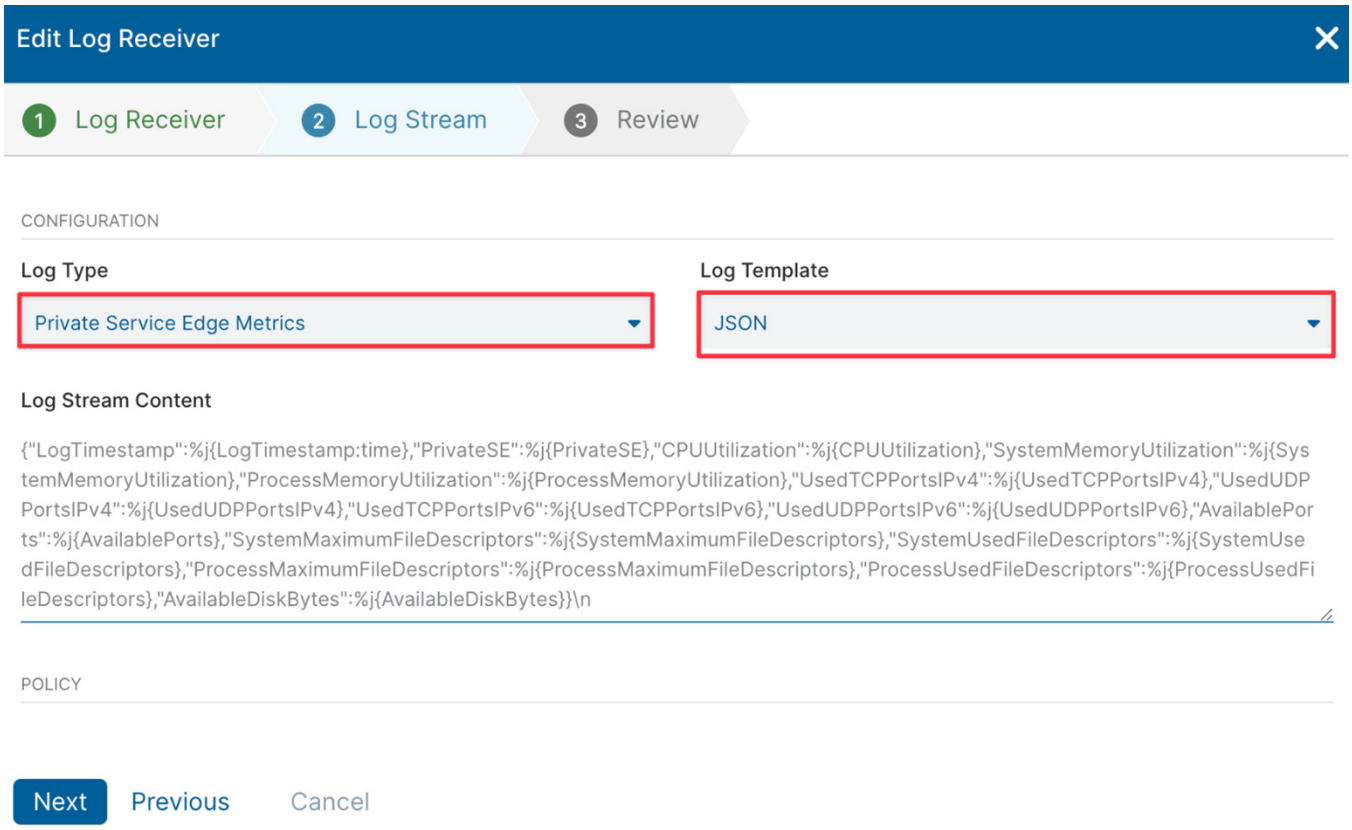


Figure 205. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

7. Click **Save**.

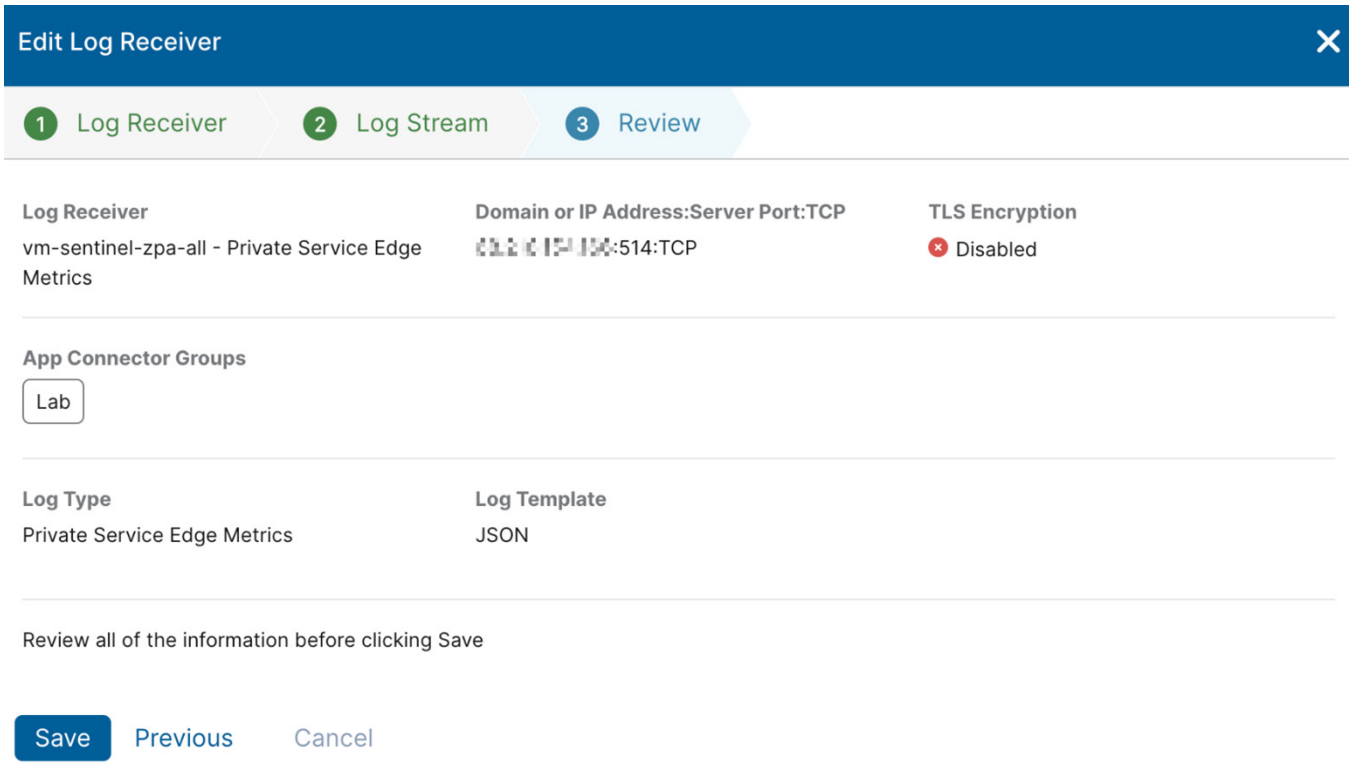


Figure 206. Save log receiver

Configure the ZPAPSEMetrics Log Analytics workspace function

The ZPAPSEMetrics function is used to parse the ZPA Private Service Edge Metrics Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

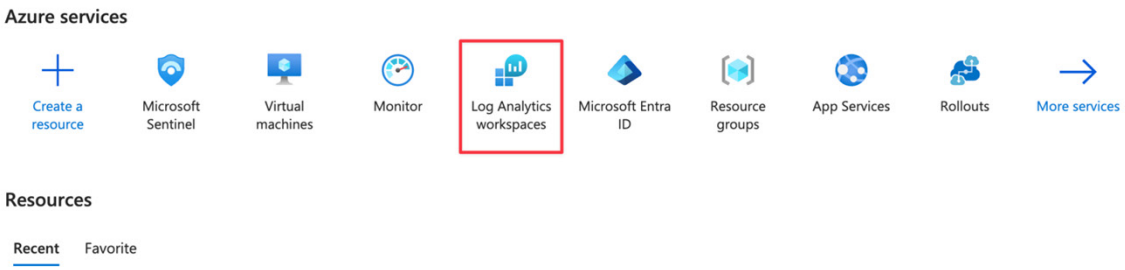


Figure 207. Log Analytics workspaces

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA Private Service Edge Metrics logs**.

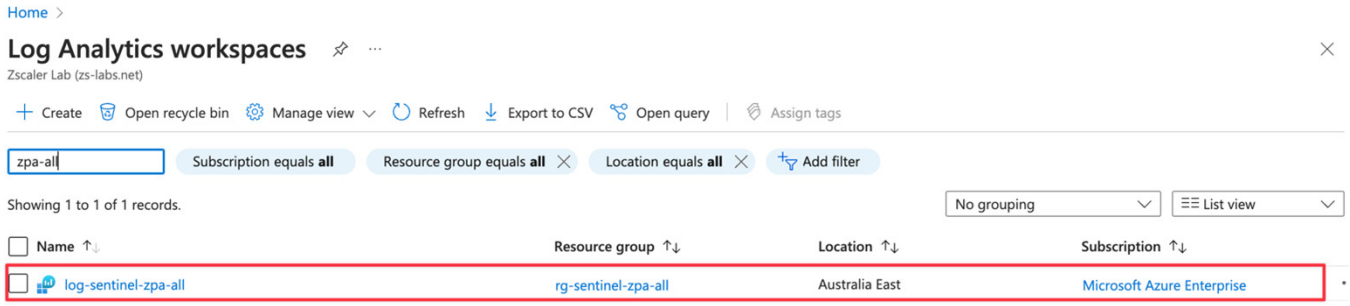


Figure 208. ZPA workspace with logs

4. Click **Logs**.

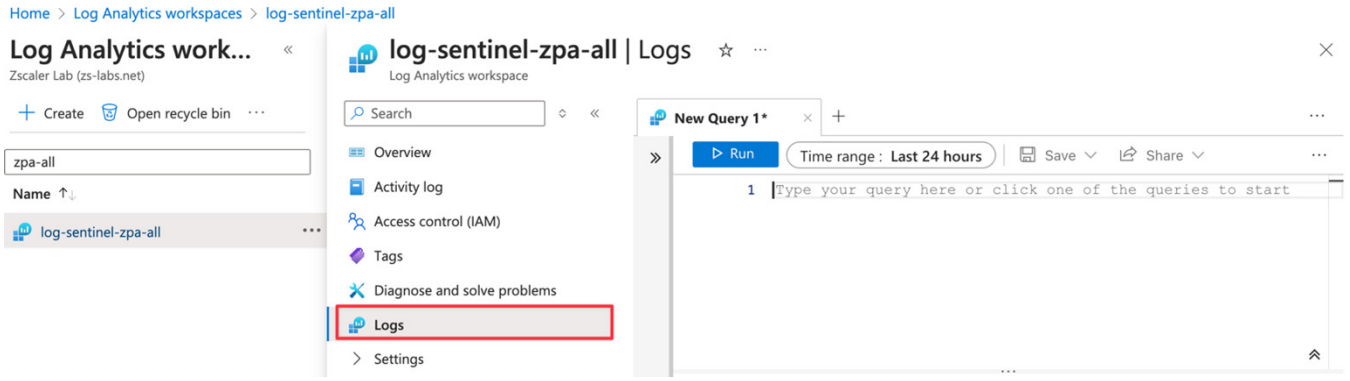


Figure 209. Logs

5. Paste the content from the `ZPAPSEMetrics.kql` file into the query window, and select **Save** > **Save as function**.

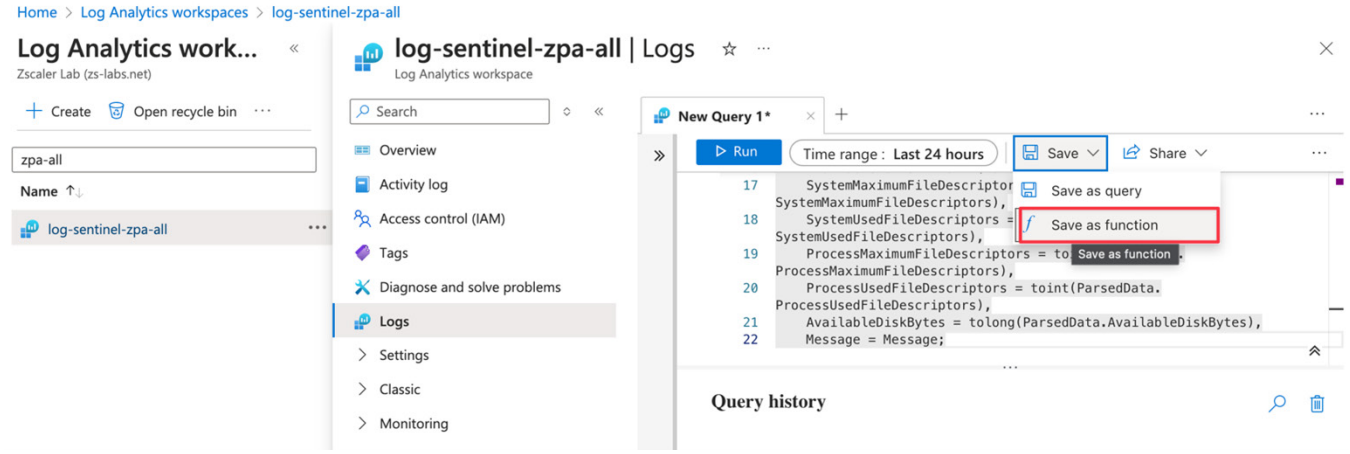


Figure 210. ZPAPSEMetrics.kql content

©2026 Zscaler, Inc. All rights reserved.

6. Enter the **Function name** `ZPAPSEMetrics`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

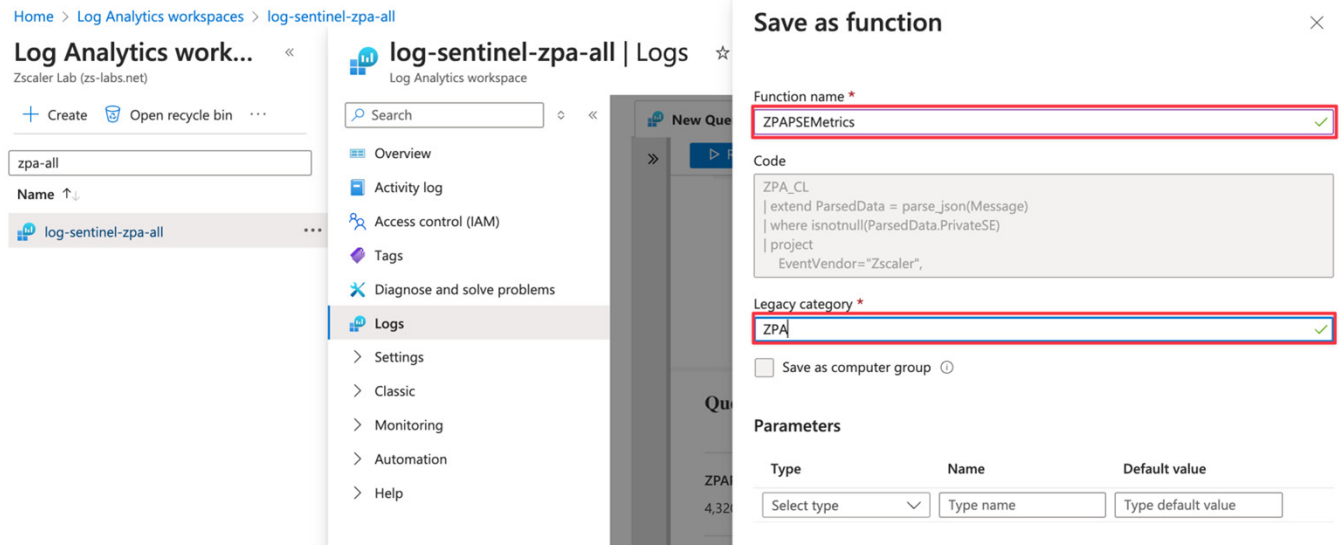


Figure 211. Save ZPAPSEMetrics

8. Run queries using the **ZPAPSEMetrics** function.

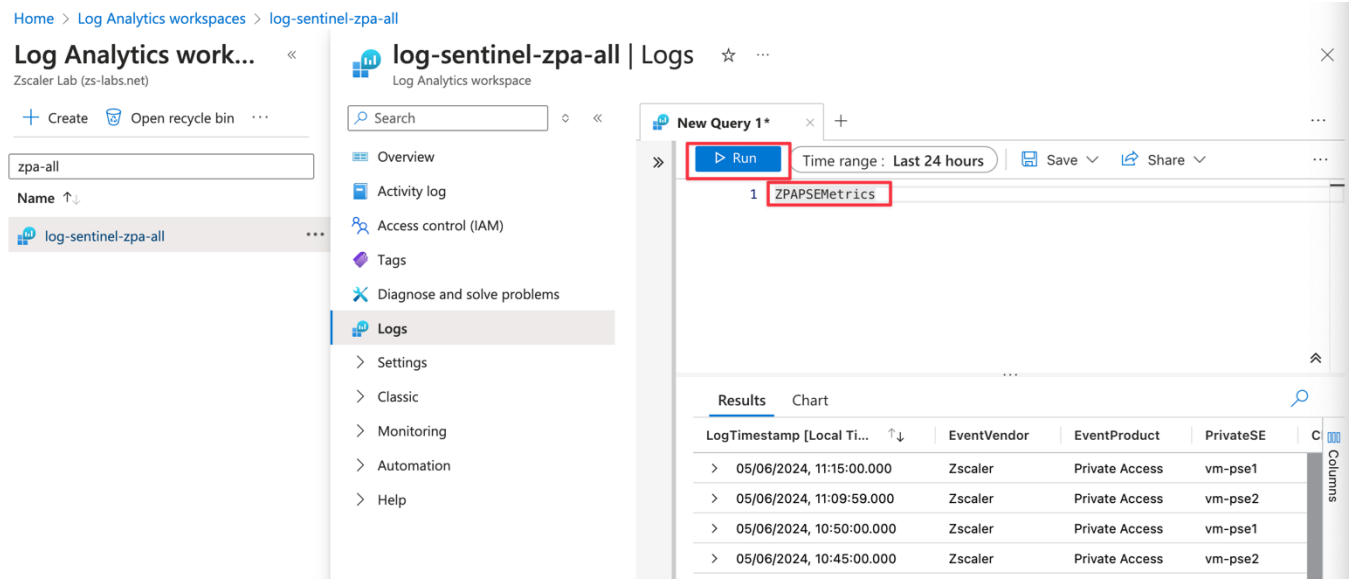


Figure 212. Run queries

©2026 Zscaler, Inc. All rights reserved.

Configure the ZPAPSEMetrics Sentinel Workbook

The ZPAPSEMetrics workbook is used to graphically display information related to the ZPA Private Service Edge Metrics Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your ZPAPSEMetrics parser.

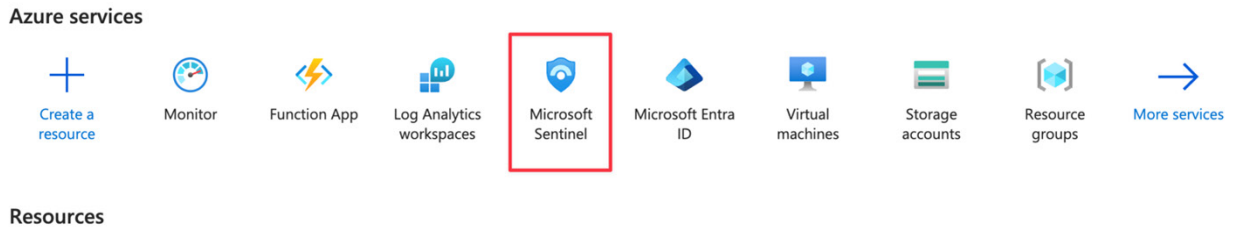


Figure 213. Microsoft Sentinel

3. Select your workspace with the **ZPA User Status logs**.

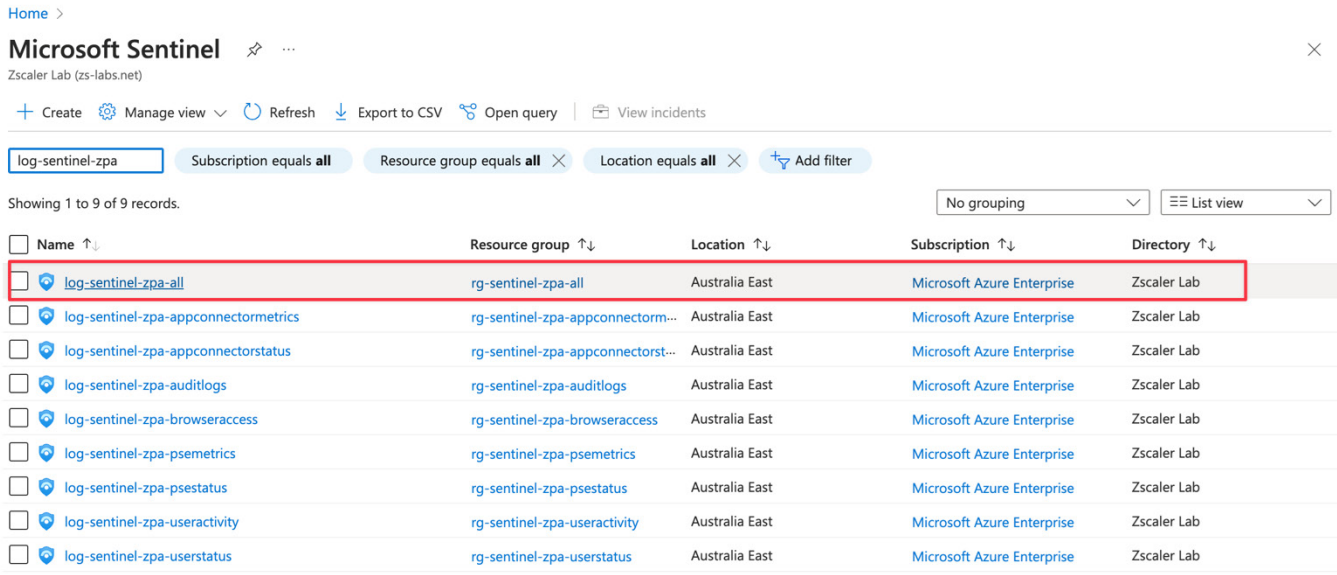


Figure 214. ZPA workspace with logs

©2026 Zscaler, Inc. All rights reserved.

4. Click **Workbooks** under **Threat management**.

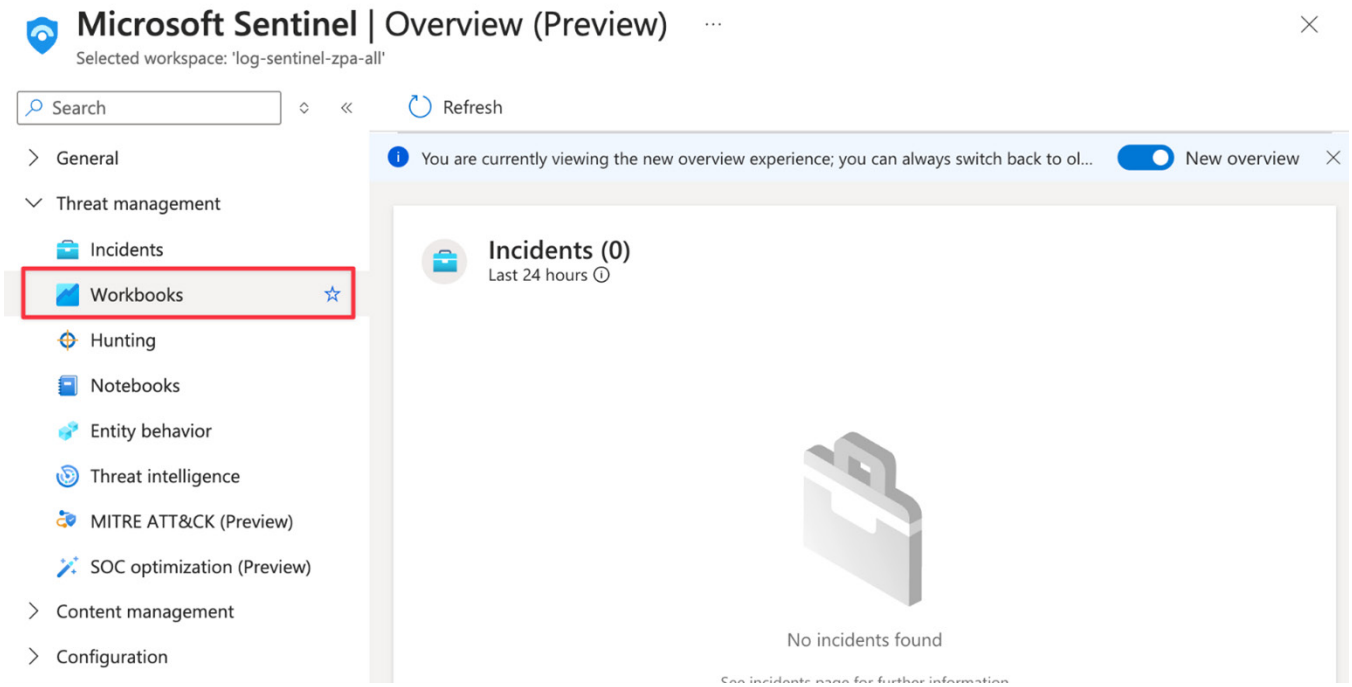


Figure 215. Workbooks

5. Click **Add Workbook**.

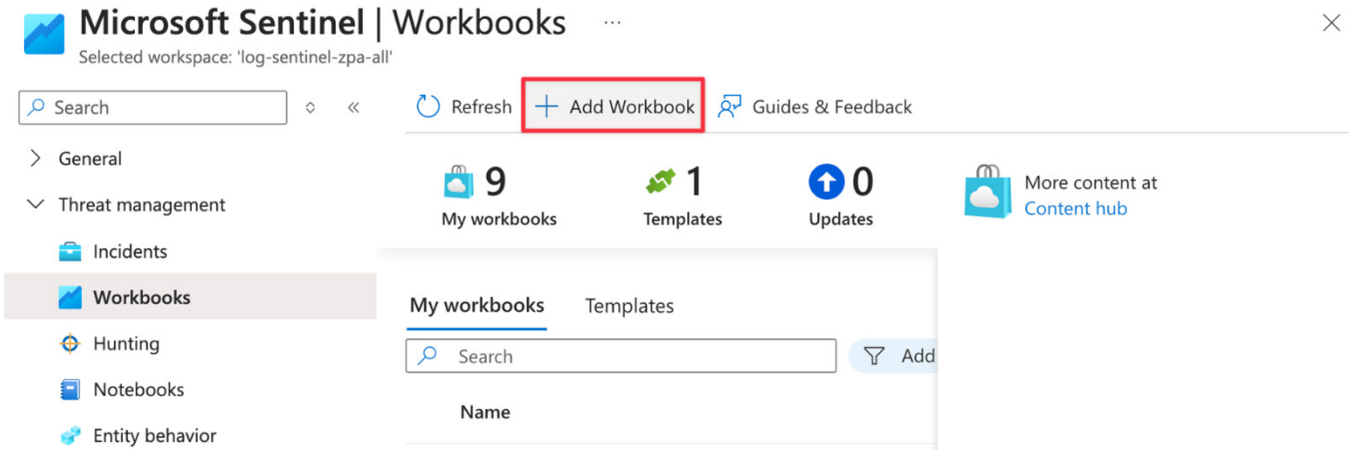


Figure 216. Add Workbook

6. Click **Edit**.



Figure 217. Edit

©2026 Zscaler, Inc. All rights reserved.

7. Click the **Advanced Editor** icon.

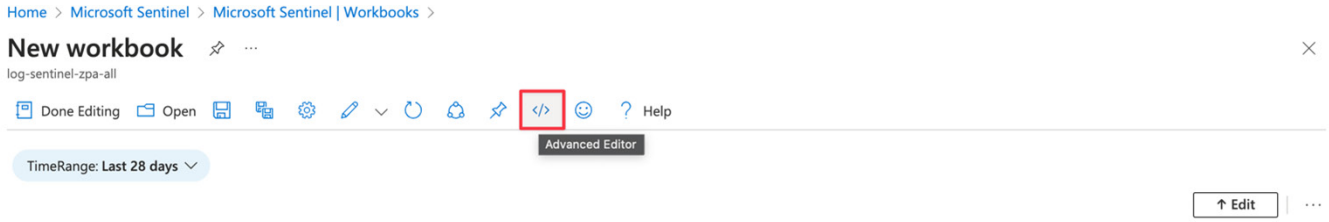


Figure 218. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPAPSEMetrics** function to display various graphs.

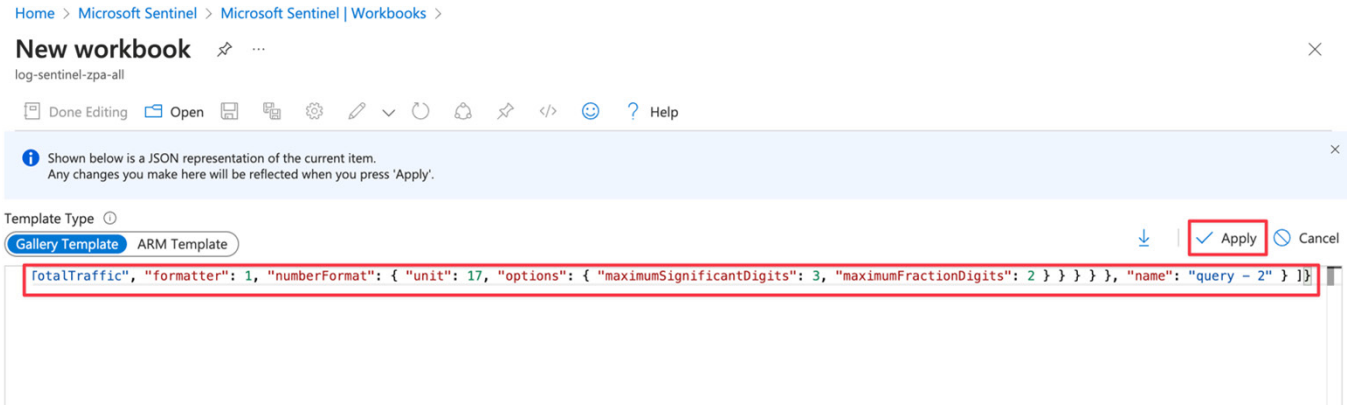


Figure 219. New code text

©2026 Zscaler, Inc. All rights reserved.

9. Click **Done Editing**.

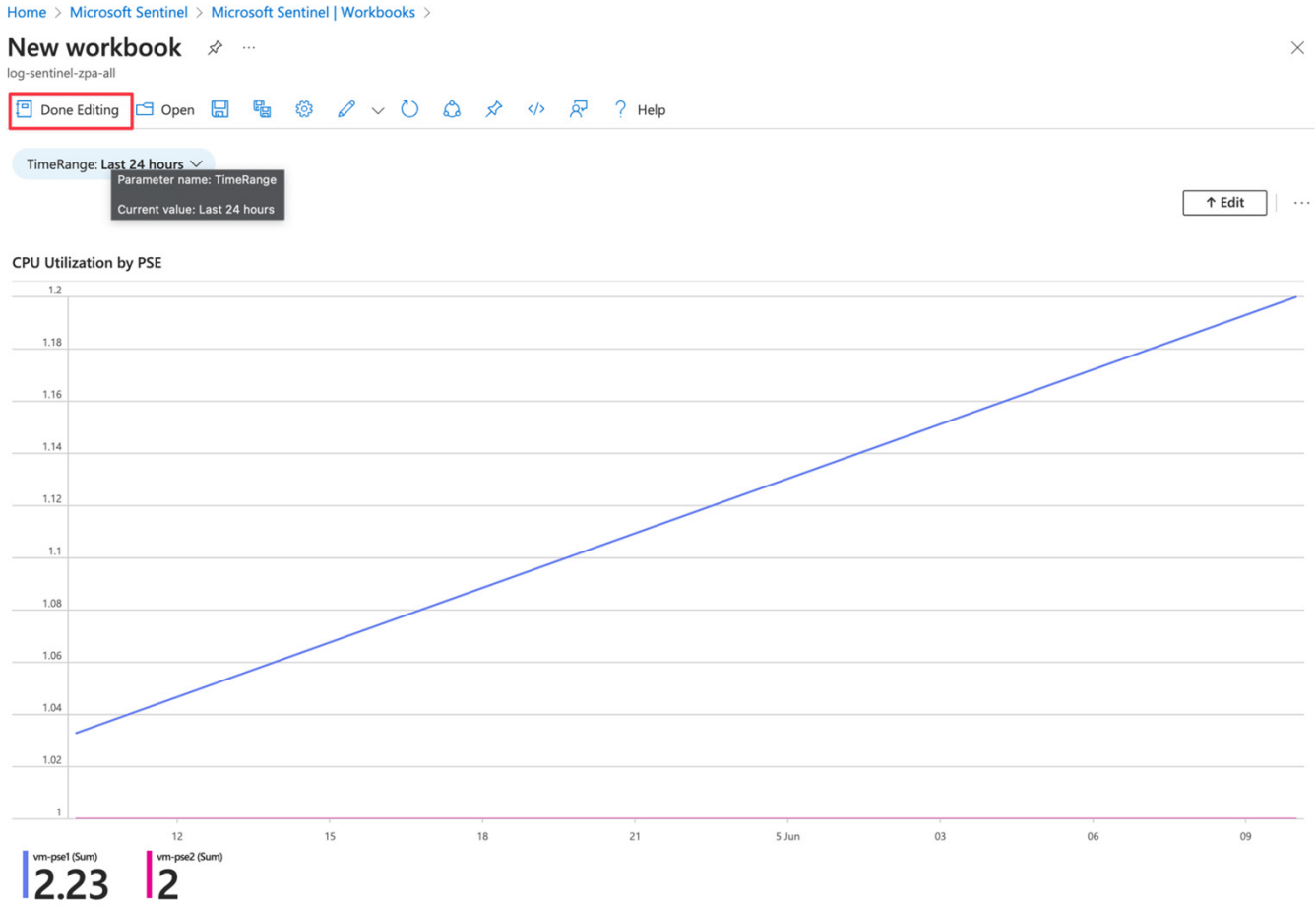


Figure 220. New workbook

10. Click **Save**.



Figure 221. Save new workbook

©2026 Zscaler, Inc. All rights reserved.

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

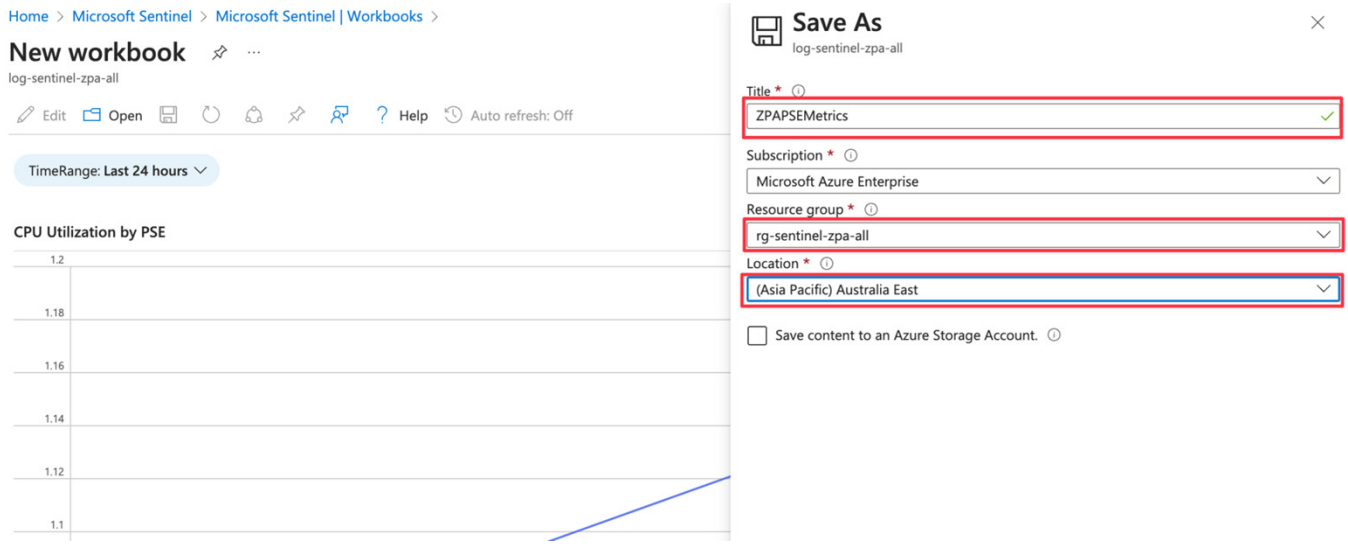


Figure 222. Save the workbook

Configure ZPA Browser Access Logs

The following sections describe how to configure ZPA Browser Access logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

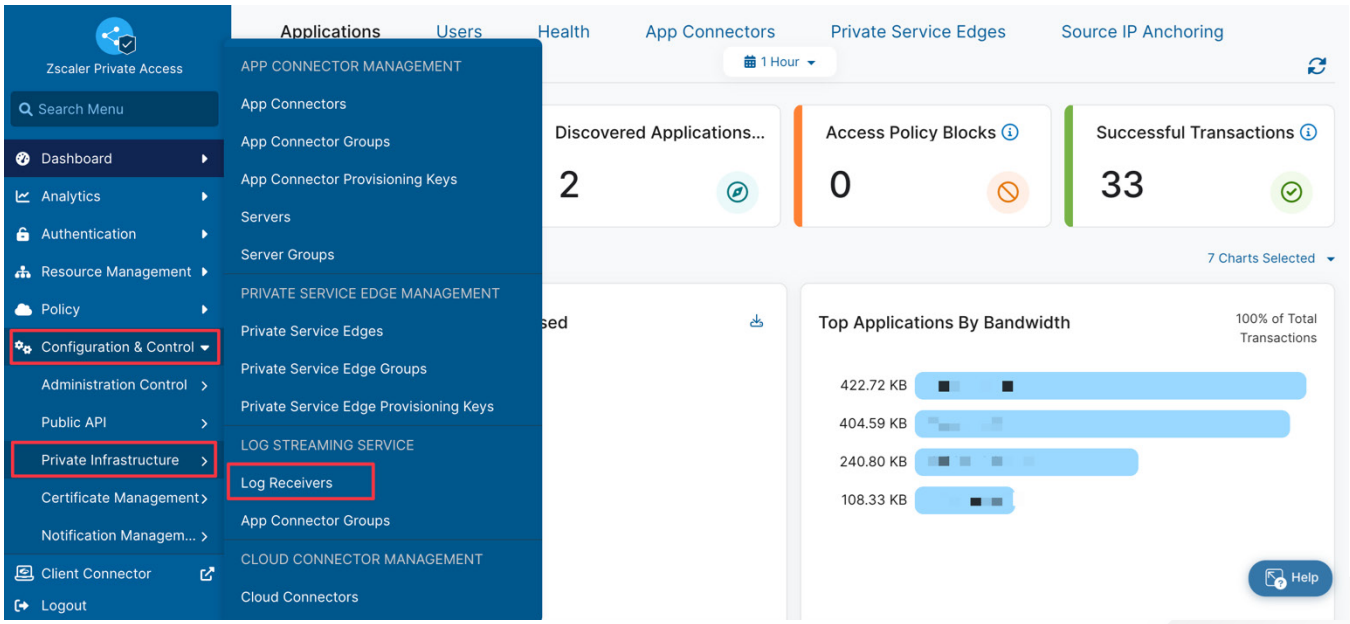


Figure 223. Log Receivers

©2026 Zscaler, Inc. All rights reserved.

2. Click **Add** (+) to add a new Log Receiver.

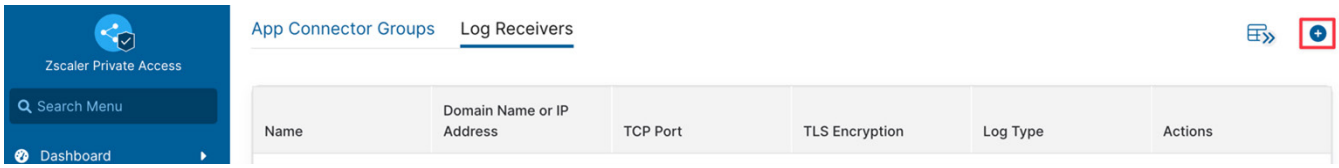


Figure 224. Add new Log Receiver

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

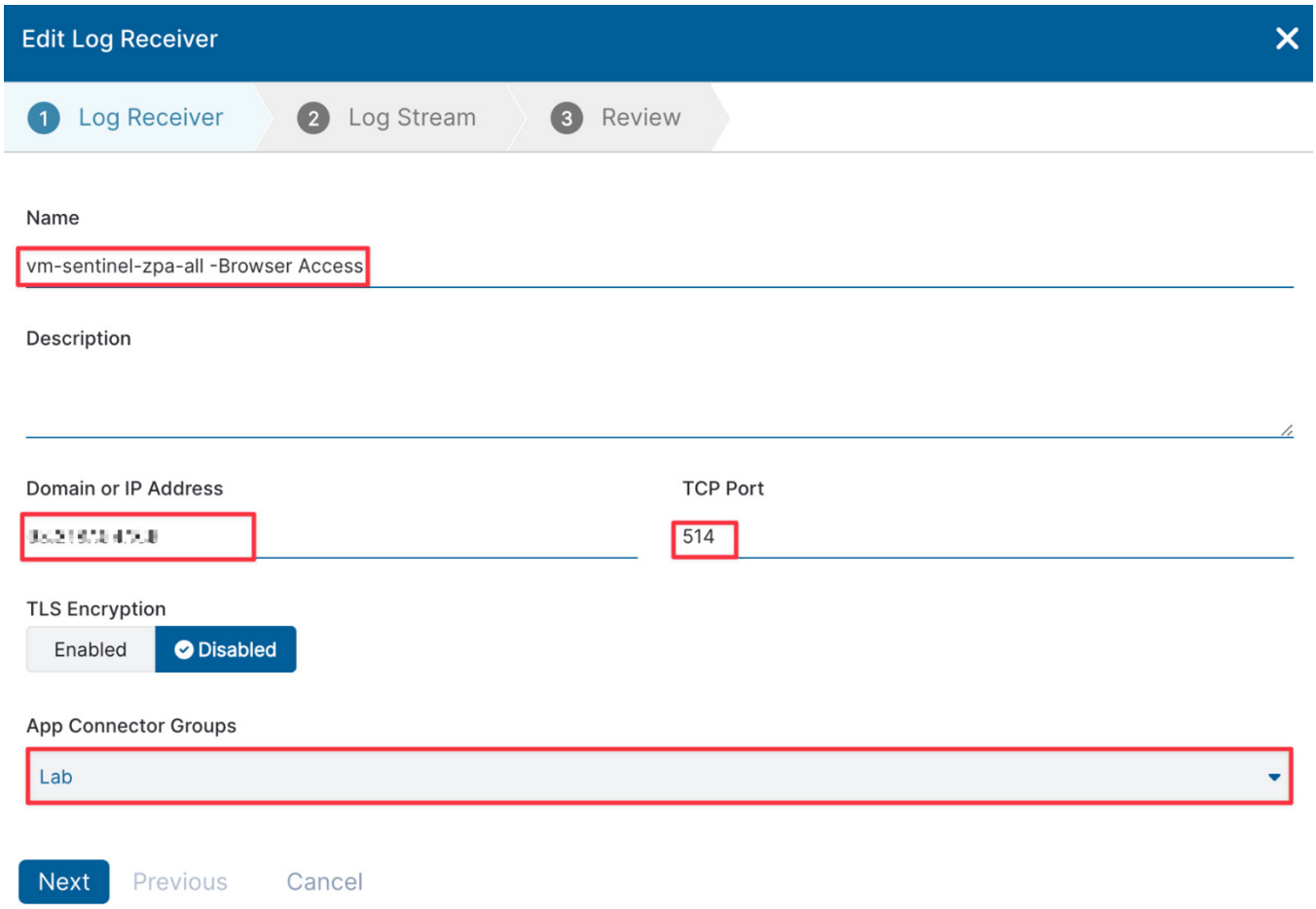


Figure 225. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

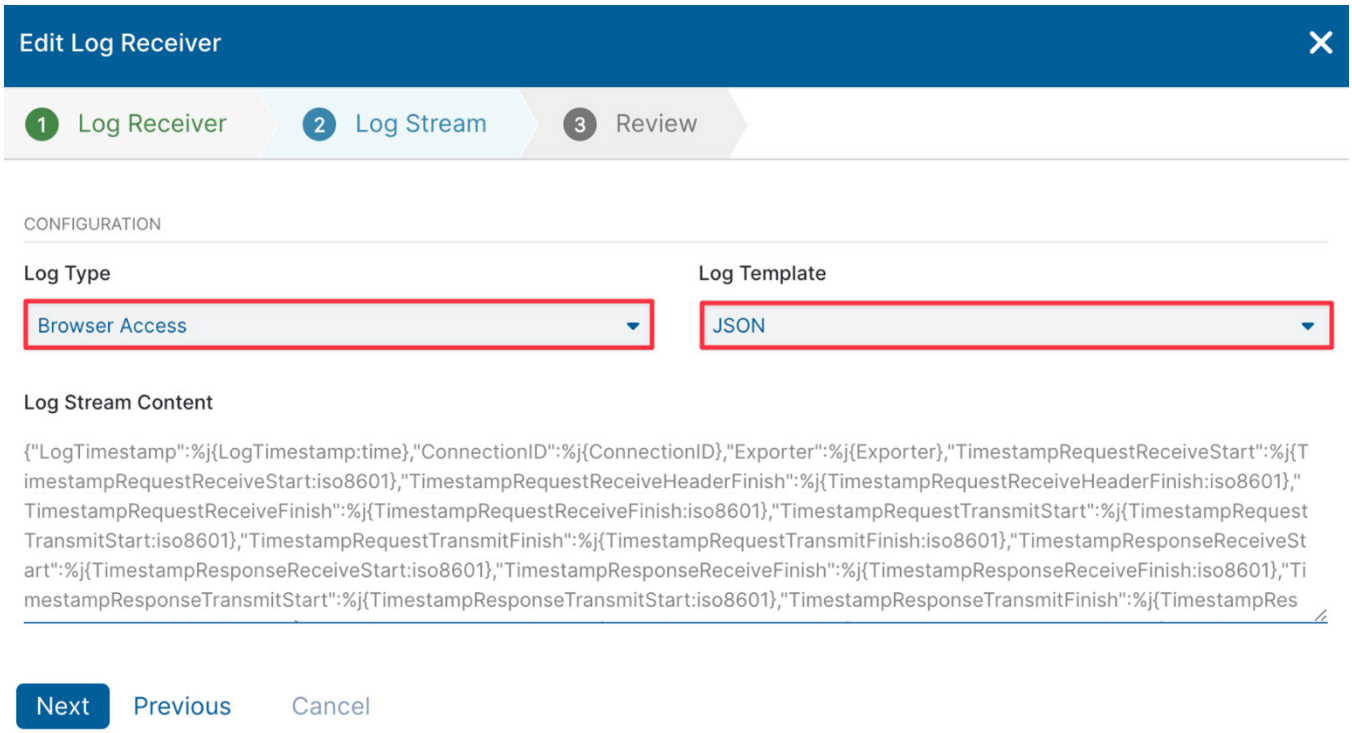


Figure 226. Configure log receiver

7. Click **Save**.

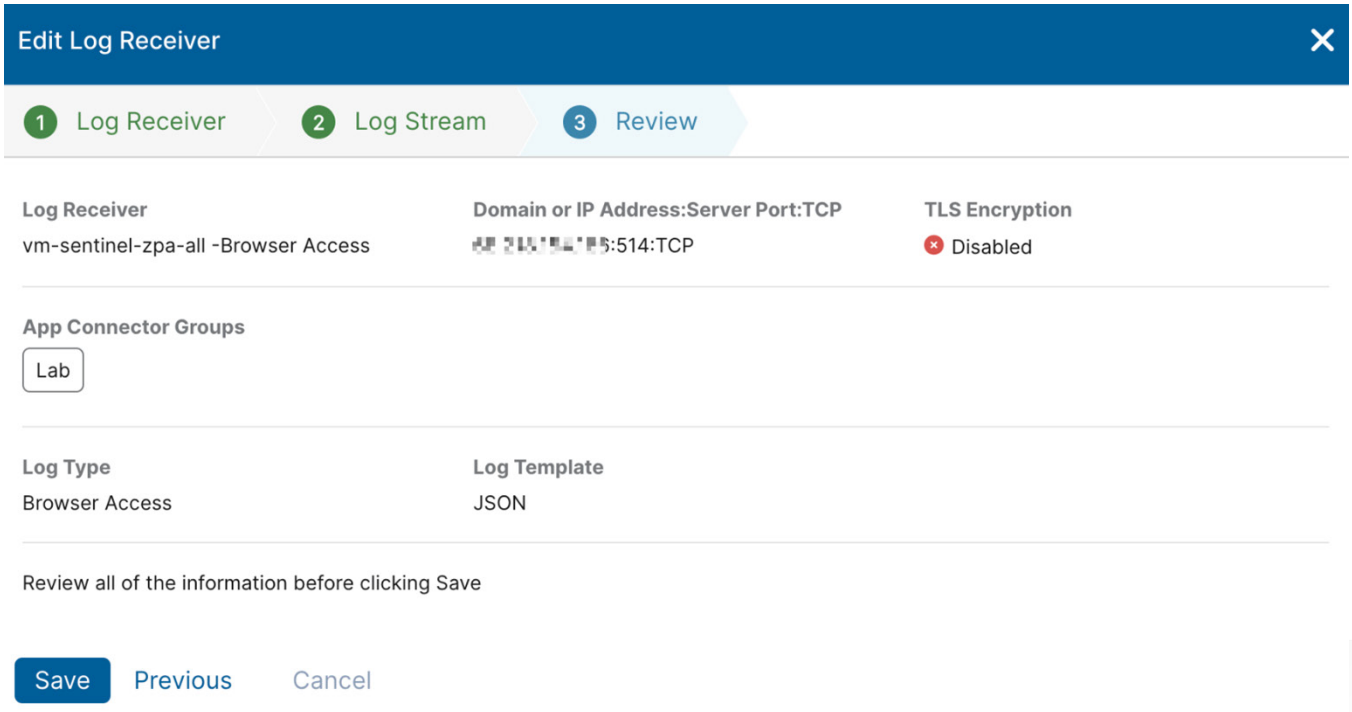


Figure 227. Save log receiver

©2026 Zscaler, Inc. All rights reserved.

Configure the ZPABrowserAccess Log Analytics Workspace Function

The ZPABrowserAccess function is used to parse the ZPA Browser Access Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

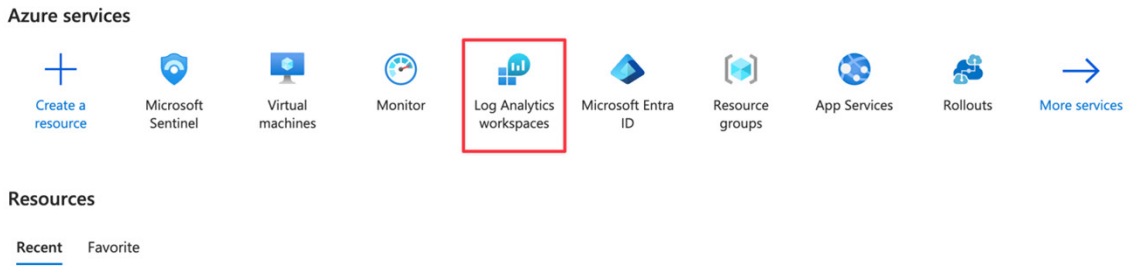


Figure 228. Log Analytics workspaces

3. Select your workspace with the **ZPA Browser Access logs**.

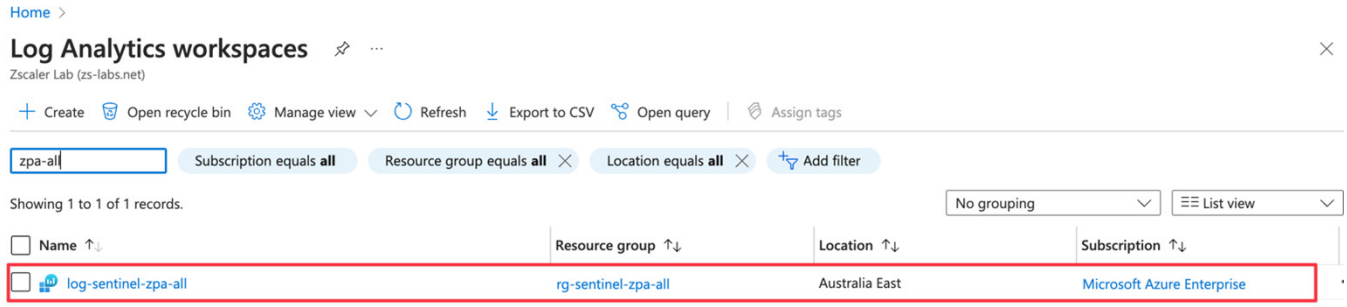


Figure 229. ZPA workspace with logs

4. Click **Logs**.

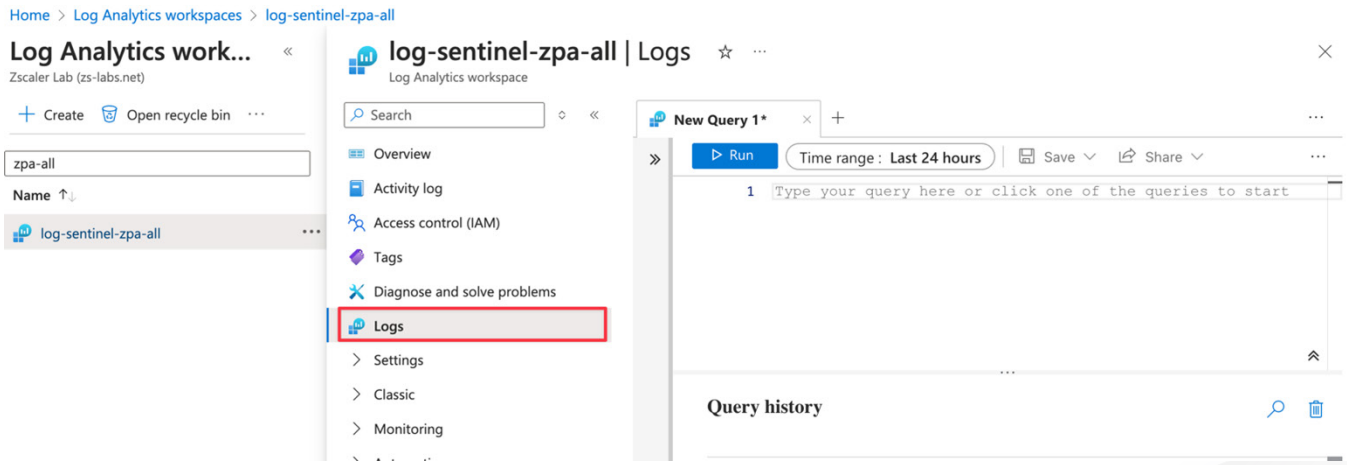


Figure 230. Logs

©2026 Zscaler, Inc. All rights reserved.

5. Paste the content from the `ZPABrowserAccess.kql` file into the query window, and select **Save** > **Save as function**.

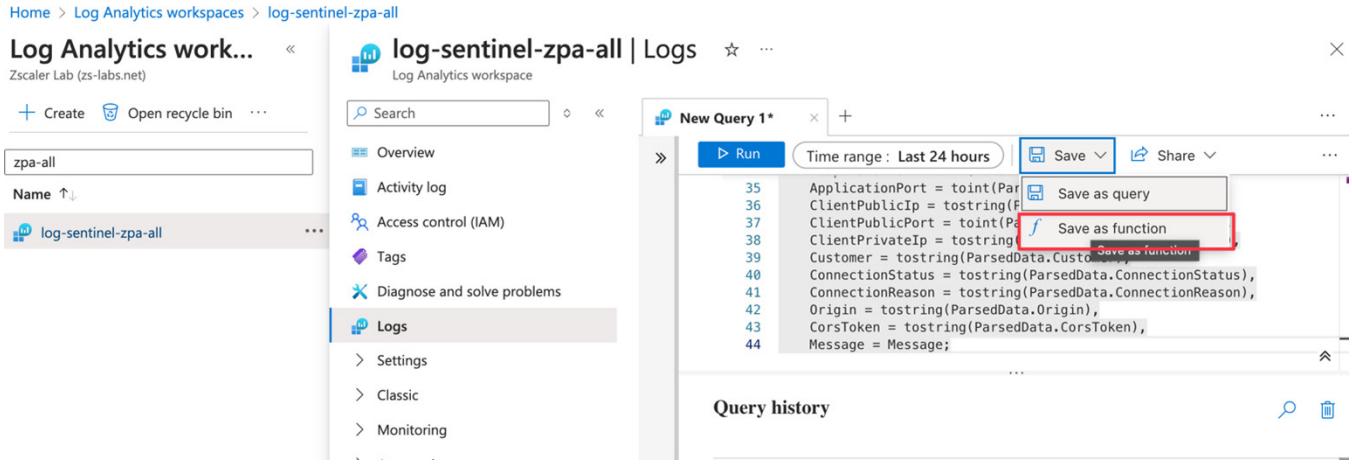


Figure 231. ZPABrowserAccess.kql content

6. Enter the **Function name** `ZPABrowserAccess`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

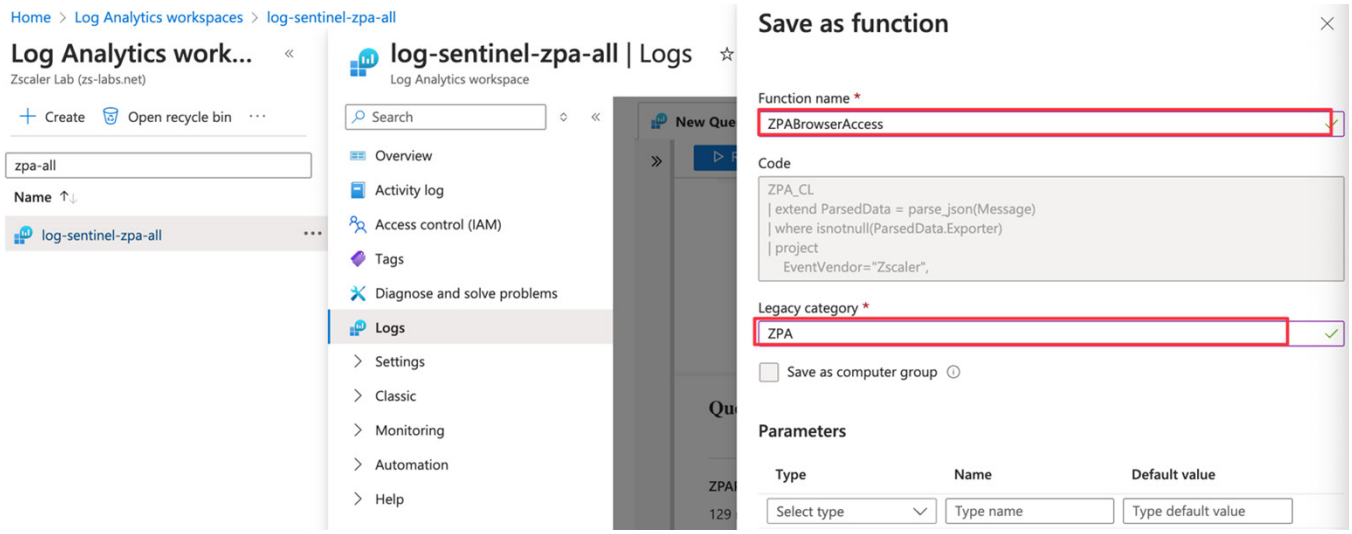


Figure 232. Save ZPABrowserAccess

©2026 Zscaler, Inc. All rights reserved.

8. Run queries using the **ZPABrowserAccess** function.

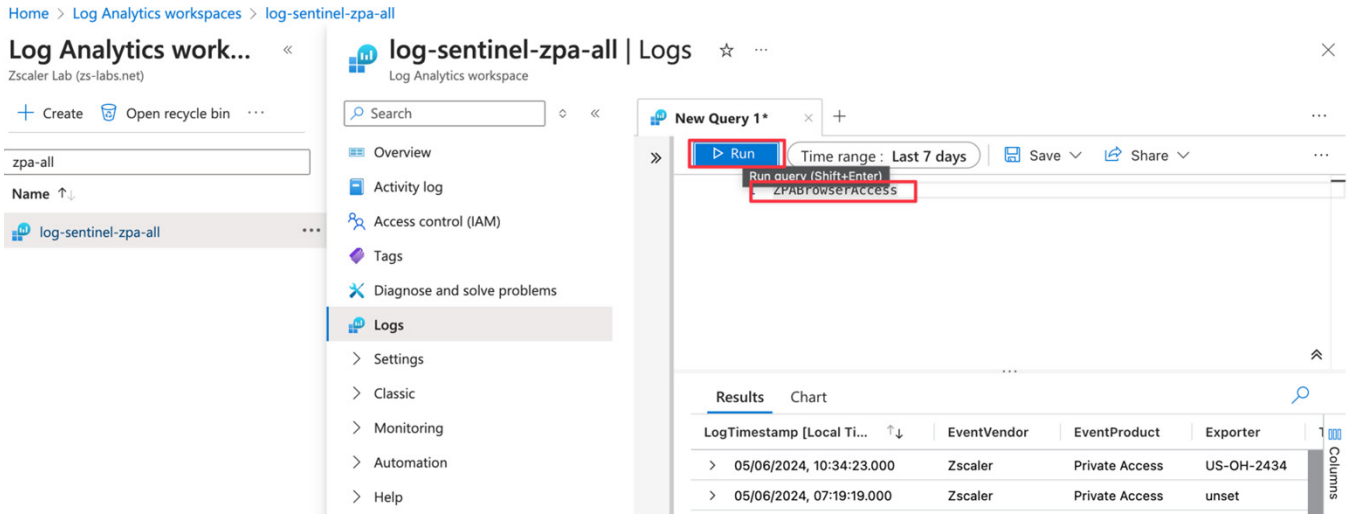


Figure 233. Run queries

Configure the ZPABrowserAccess Sentinel Workbook

The ZPABrowserAccess workbook is used to graphically display information related to the ZPA Browser Access Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your ZPABrowserAccess parser.

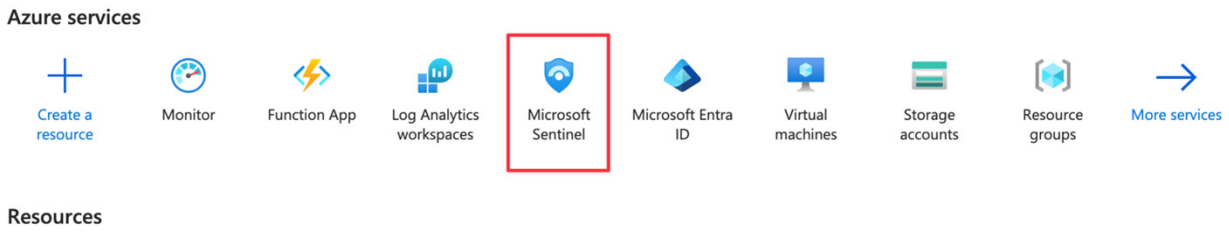


Figure 234. Microsoft Sentinel

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA Browser Access logs**.

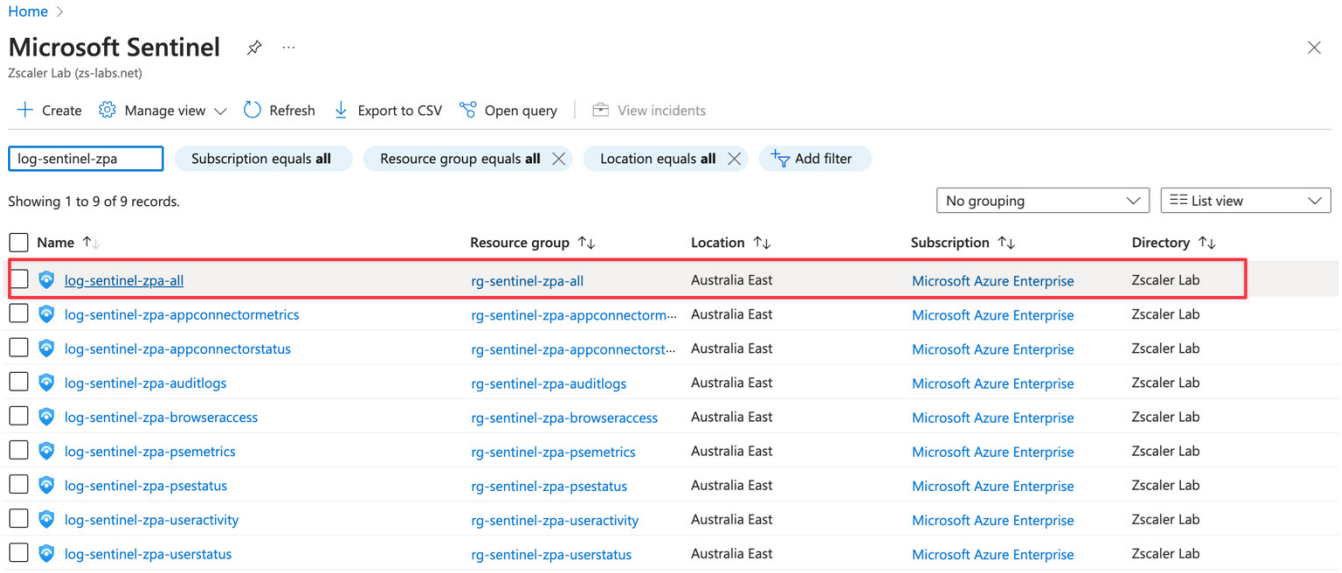


Figure 235. ZPA workspace with logs

4. Click **Workbooks** under **Threat management**.

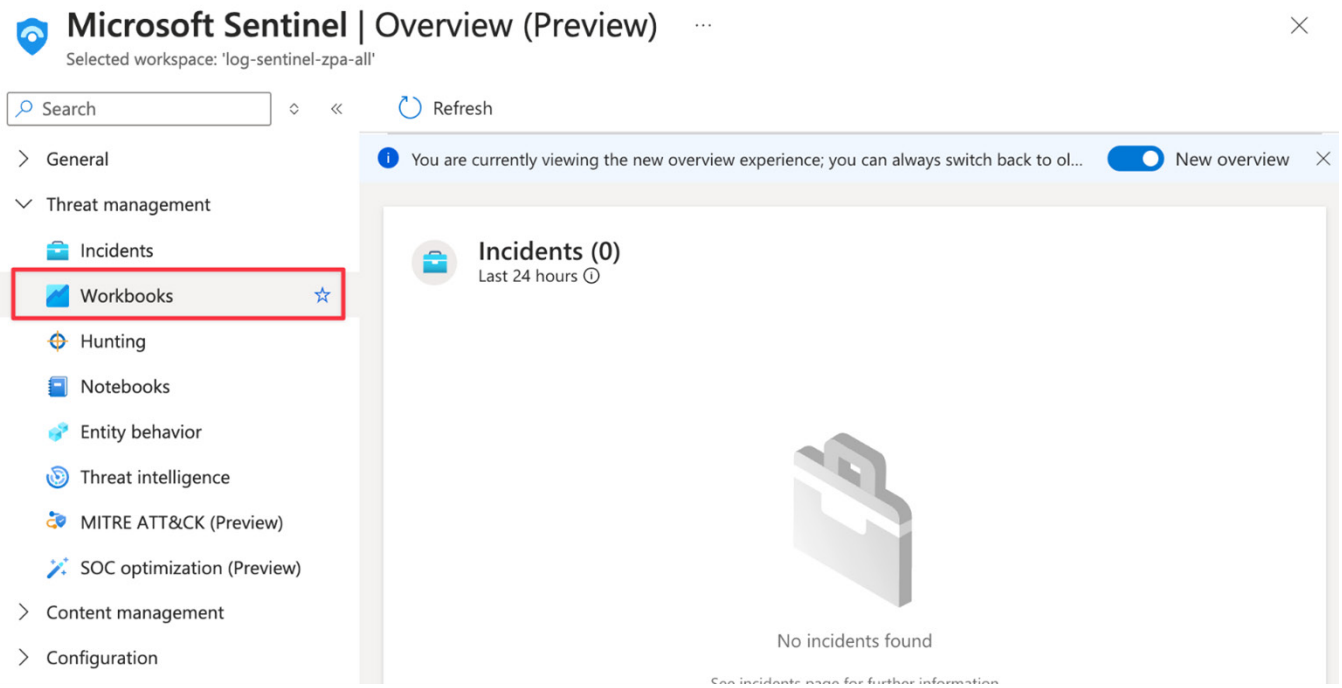


Figure 236. Workbooks

©2026 Zscaler, Inc. All rights reserved.

5. Click **Add Workbook**.

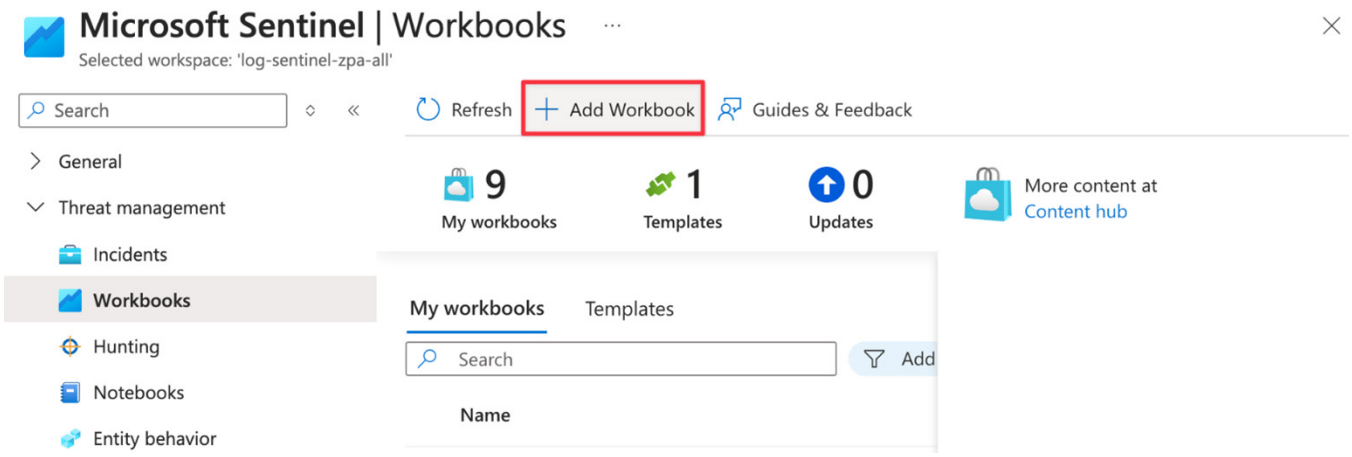


Figure 237. Add Workbook

6. Click **Edit**.



Figure 238. Edit

7. Click the **Advanced Editor** icon.

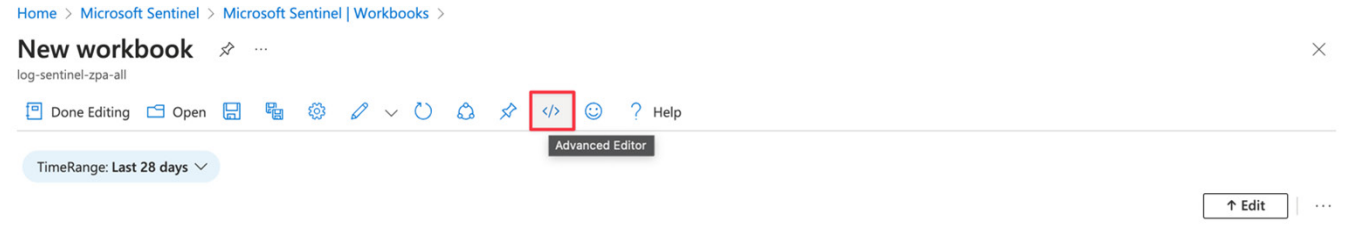


Figure 239. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPABrowserAccess** function to display various graphs.

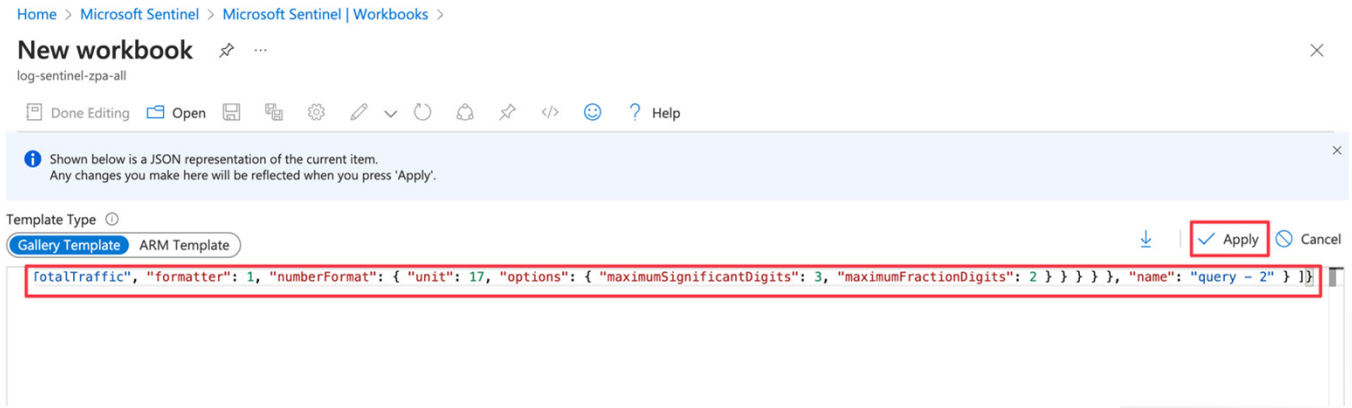


Figure 240. New code text

©2026 Zscaler, Inc. All rights reserved.

9. Click **Done Editing**.

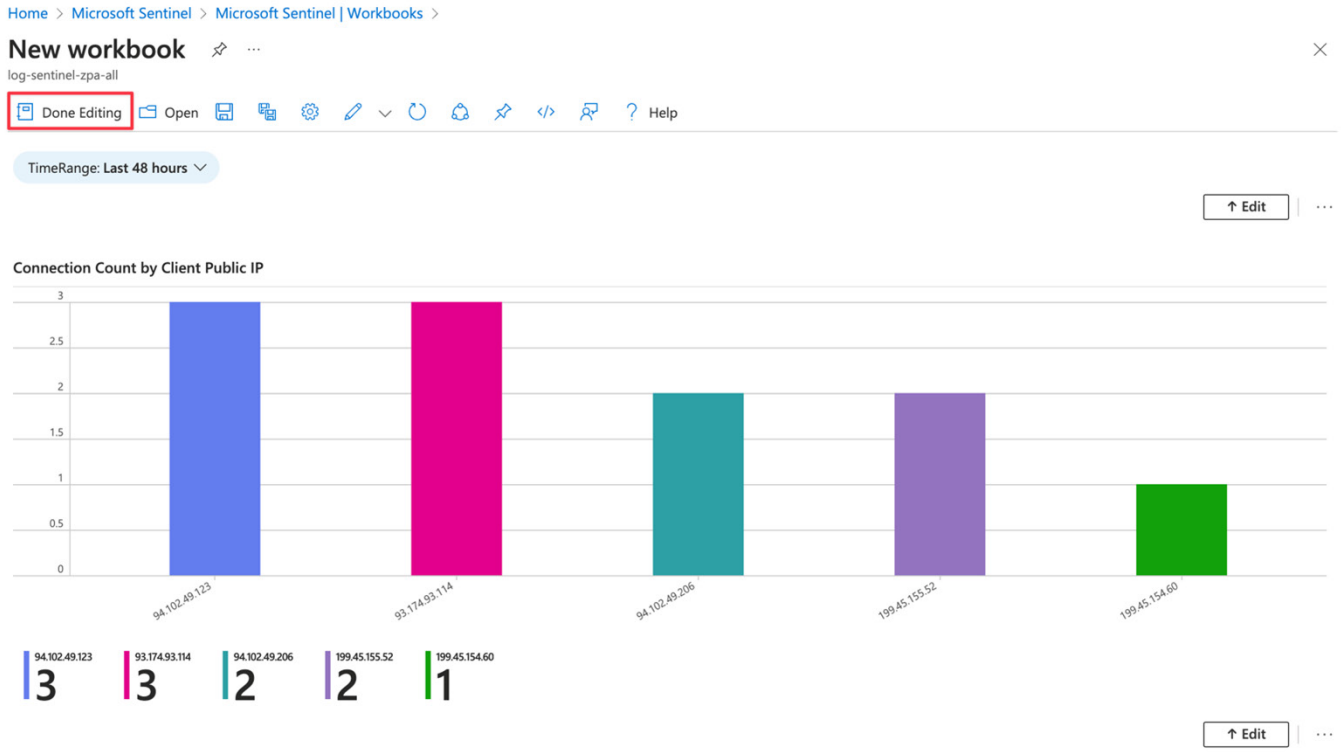


Figure 241. New workbook

10. Click **Save**.



Figure 242. Save workbook

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

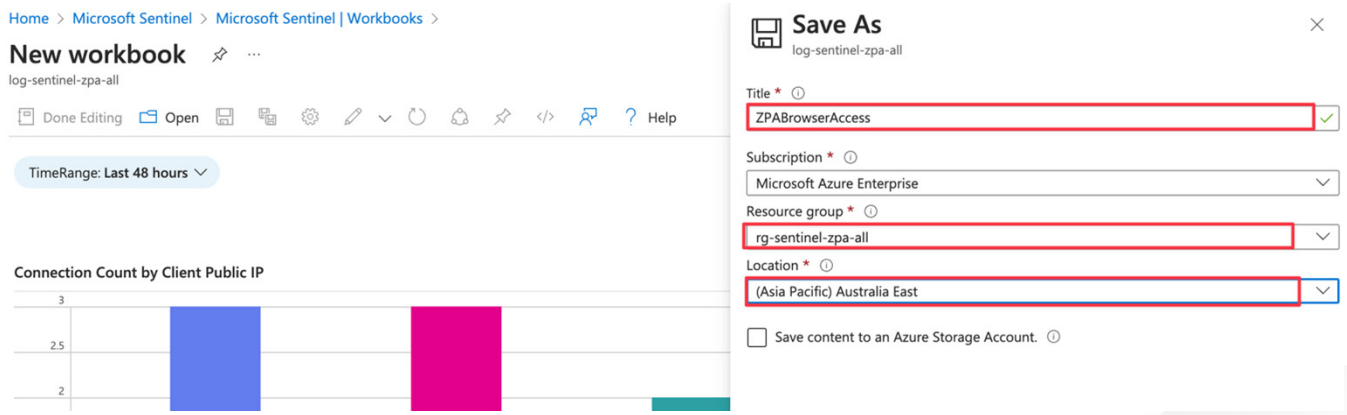


Figure 243. Save the workbook

©2026 Zscaler, Inc. All rights reserved.

Configure ZPA Audit Logs

The following sections describe how to configure ZPA Audit logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

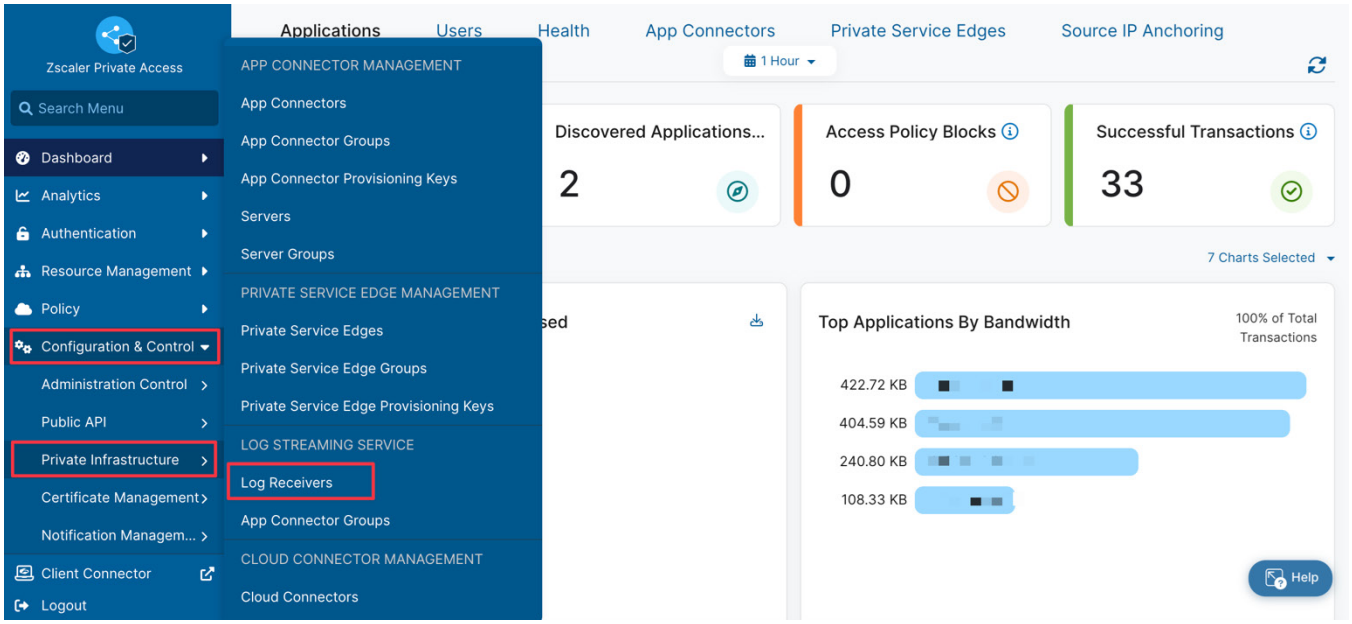


Figure 244. Log Receivers

2. Click **Add** (+) to add a new Log Receiver.

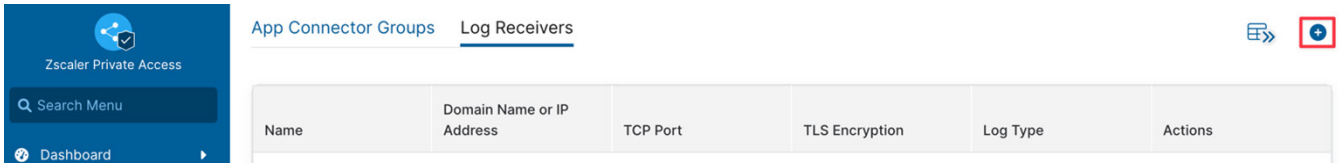


Figure 245. Add new Log Receiver

©2026 Zscaler, Inc. All rights reserved.

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

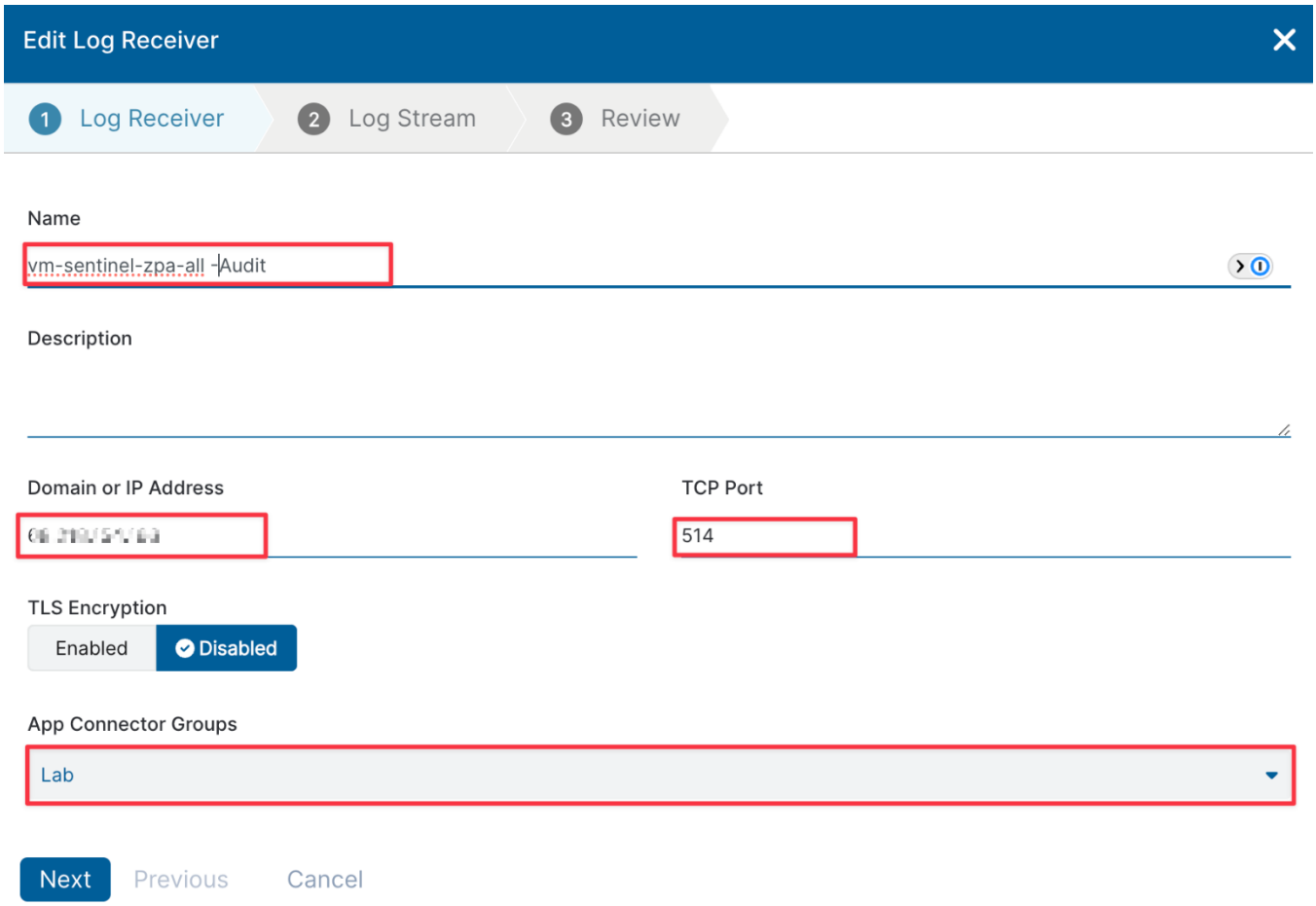


Figure 246. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

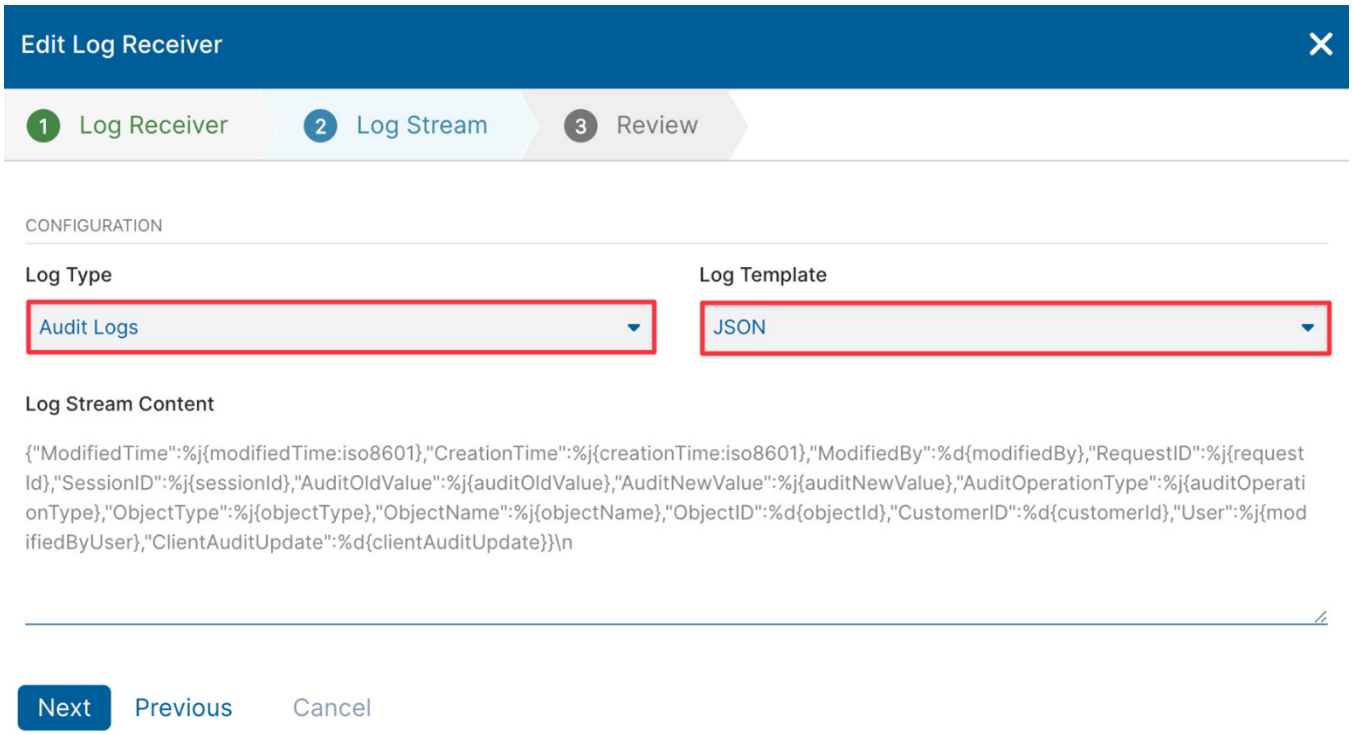


Figure 247. Configure log receiver

7. Click **Save**.

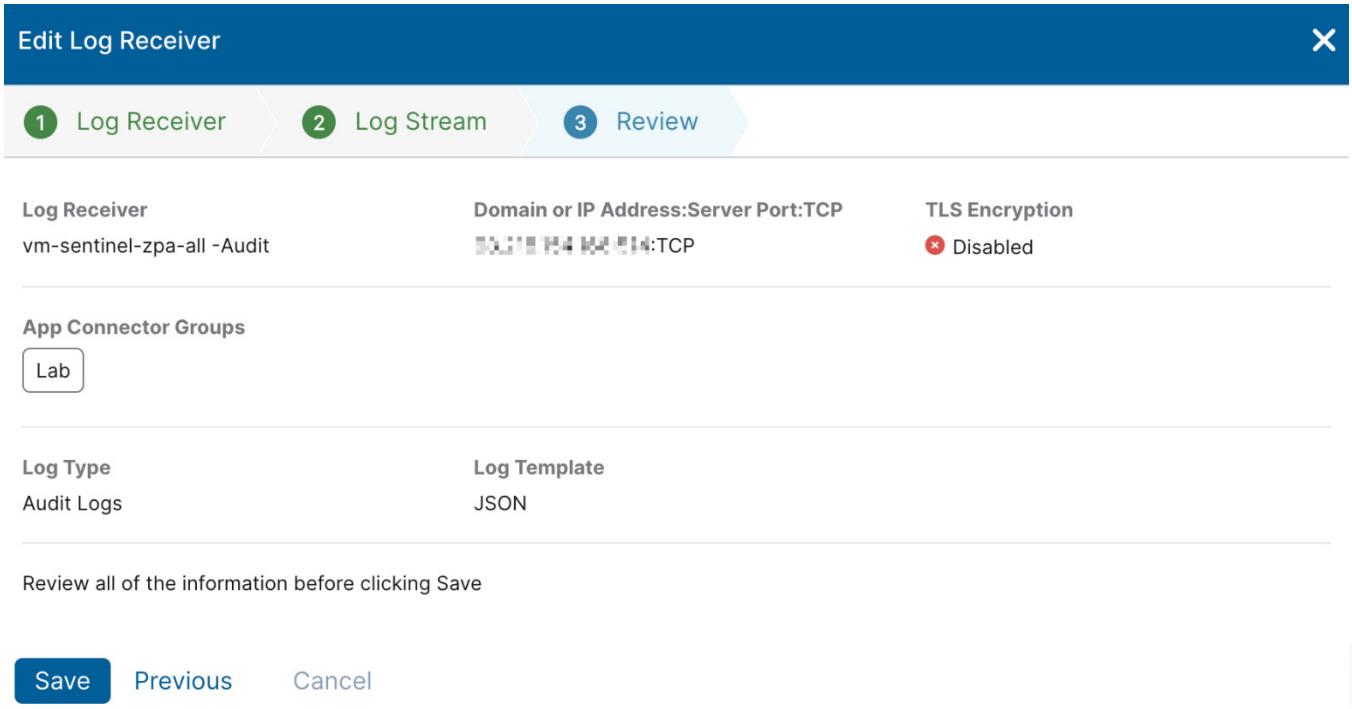


Figure 248. Save log receiver

©2026 Zscaler, Inc. All rights reserved.

Configure the ZPAAudit Log Analytics Workspace Function

The ZPAAudit function is used to parse the ZPA Audit Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

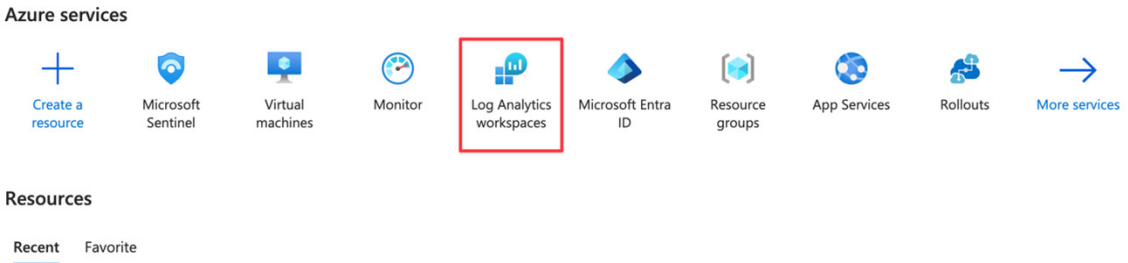


Figure 249. Log Analytics workspaces

3. Select your workspace with the **ZPA Audit logs**.

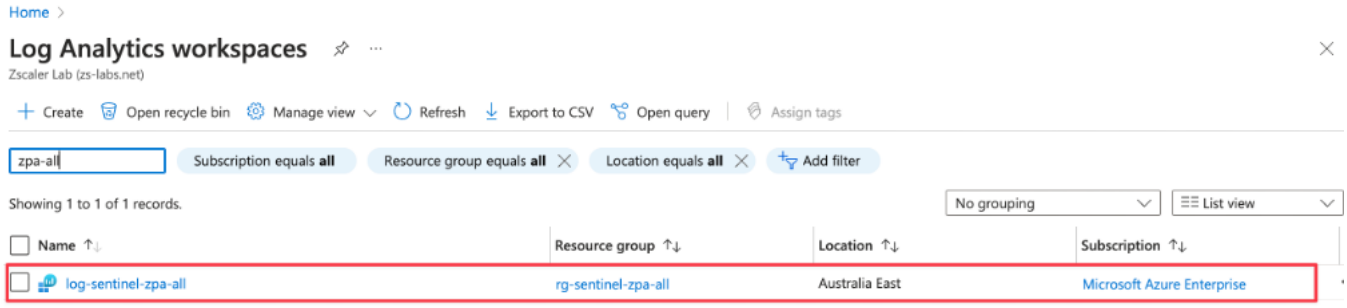


Figure 250. ZPA workspace with logs

4. Click **Logs**.

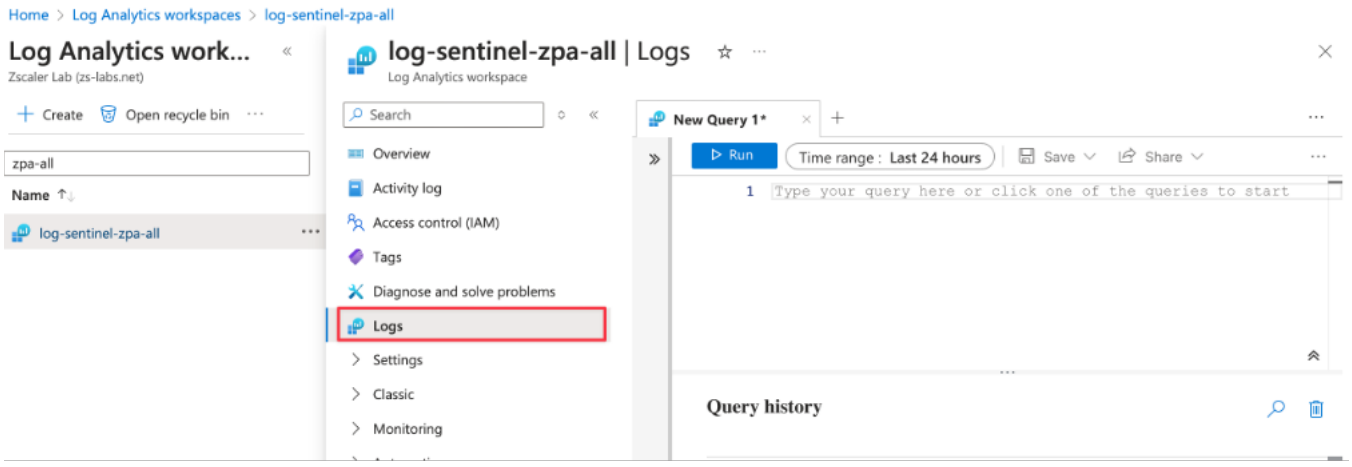


Figure 251. Logs

©2026 Zscaler, Inc. All rights reserved.

5. Paste the content from the `ZPAAudit.kql` file into the query window, and select **Save** > **Save as function**.

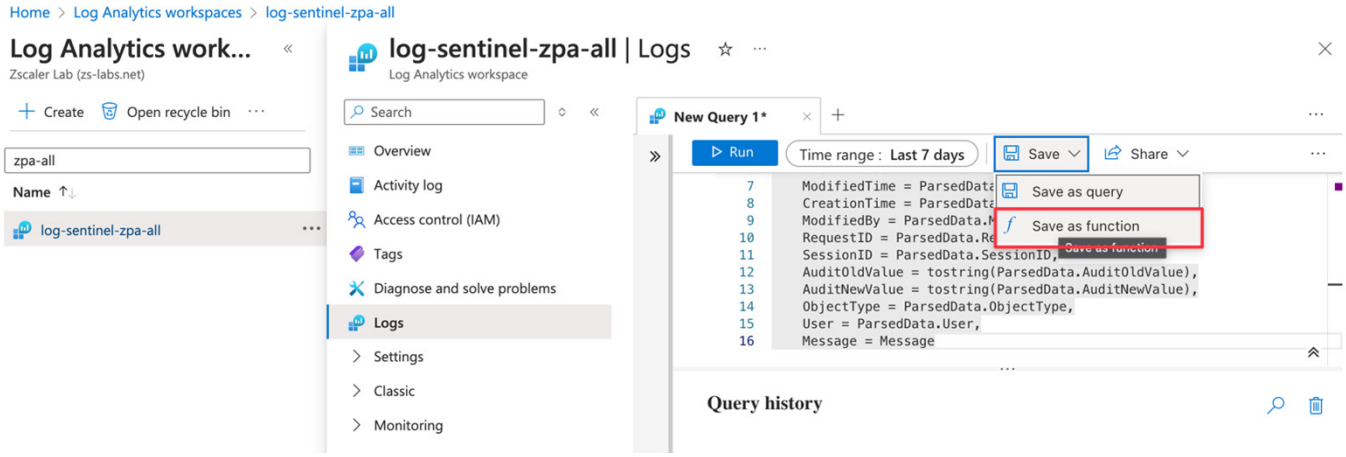


Figure 252. ZPAAudit.kql content

6. Enter the **Function name** `ZPAAudit`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

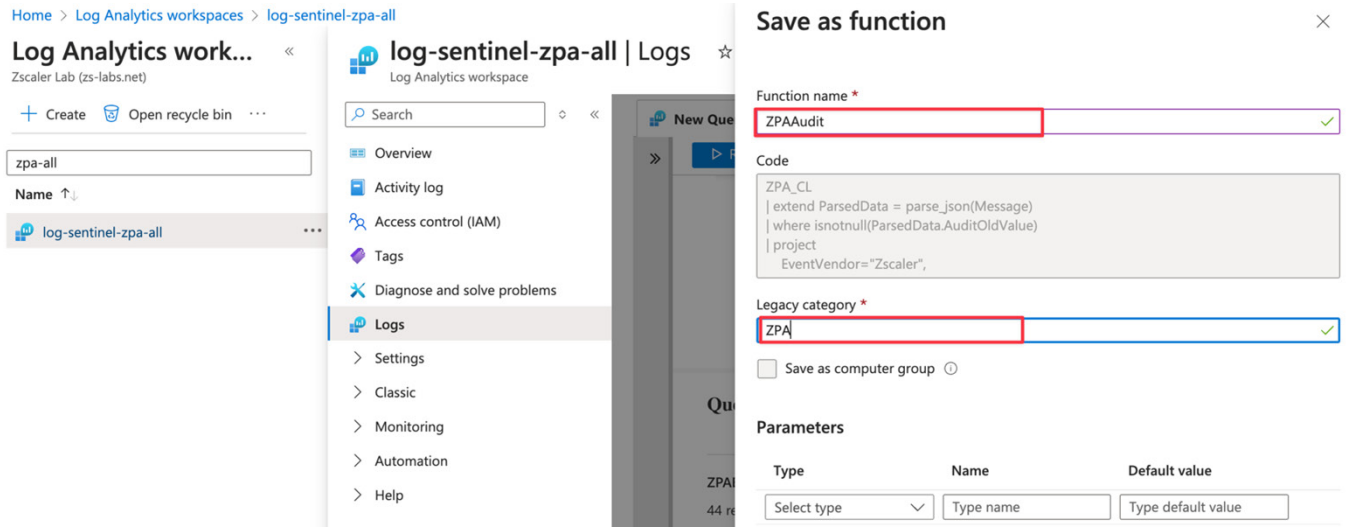


Figure 253. Save ZPAAudit

©2026 Zscaler, Inc. All rights reserved.

8. Run queries using the **ZPAAudit** function.

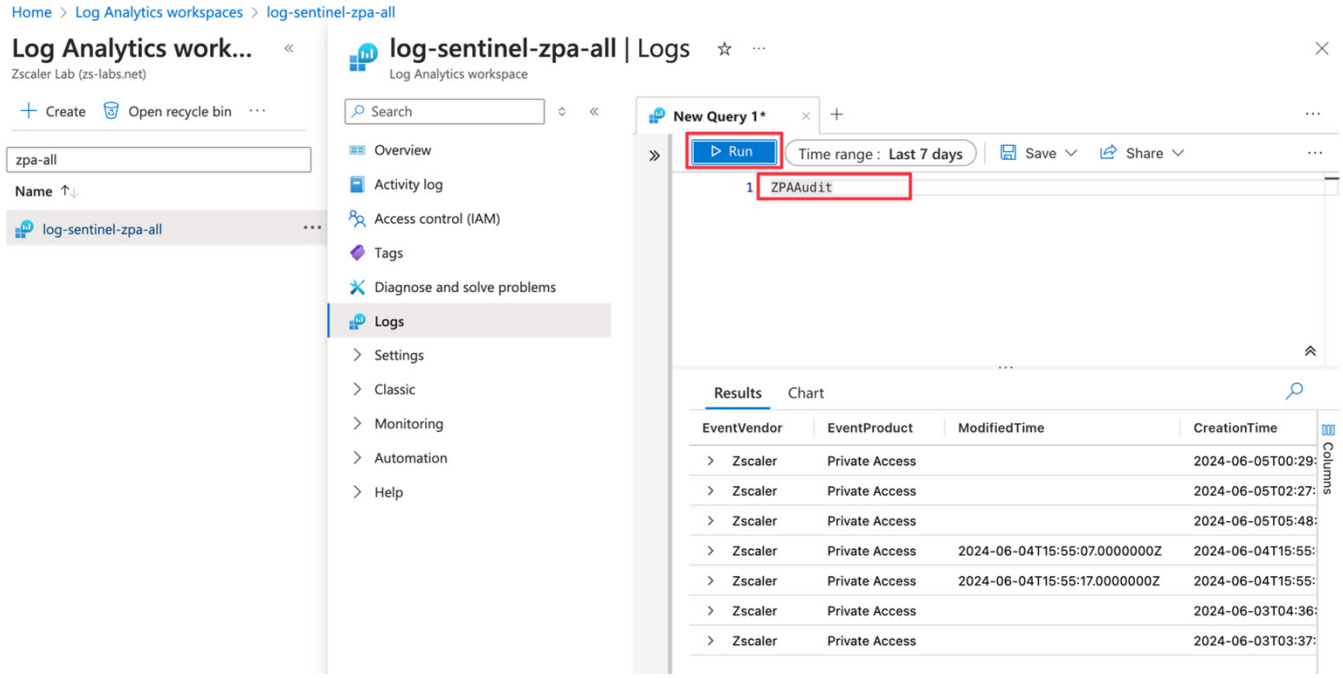


Figure 254. Run queries

Configure the ZPAAudit Sentinel workbook

The ZPAAudit workbook is used to graphically display information related to the ZPA Audit Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your ZPAAudit parser.

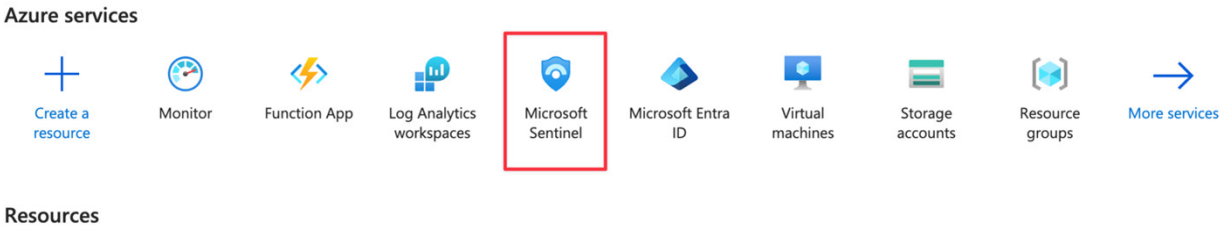


Figure 255. Microsoft Sentinel

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA Audit logs**.

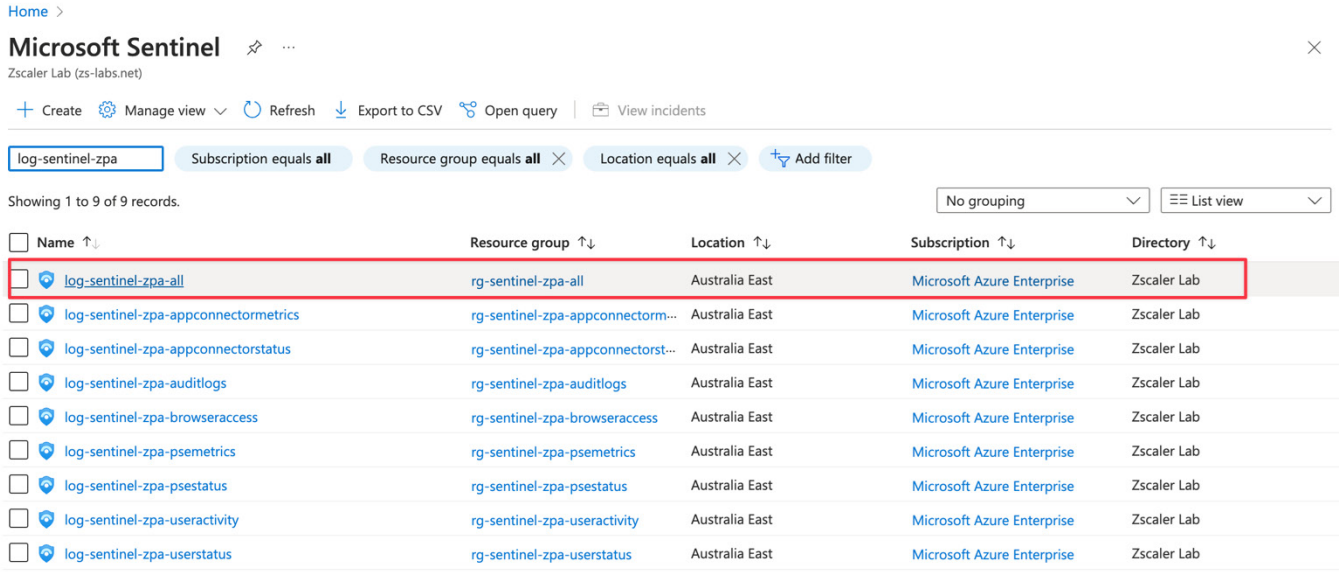


Figure 256. ZPA workspace with logs

4. Click **Workbooks** under **Threat management**.

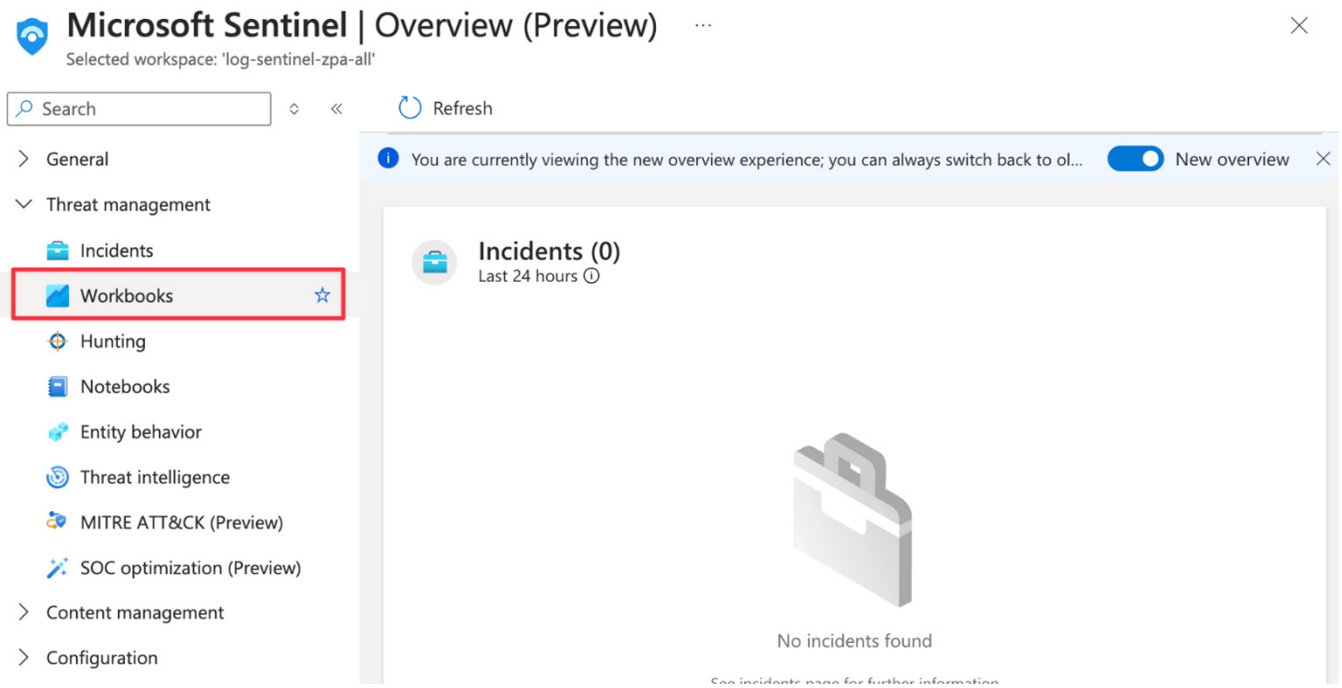


Figure 257. Workbooks

©2026 Zscaler, Inc. All rights reserved.

5. Click **Add Workbook**.

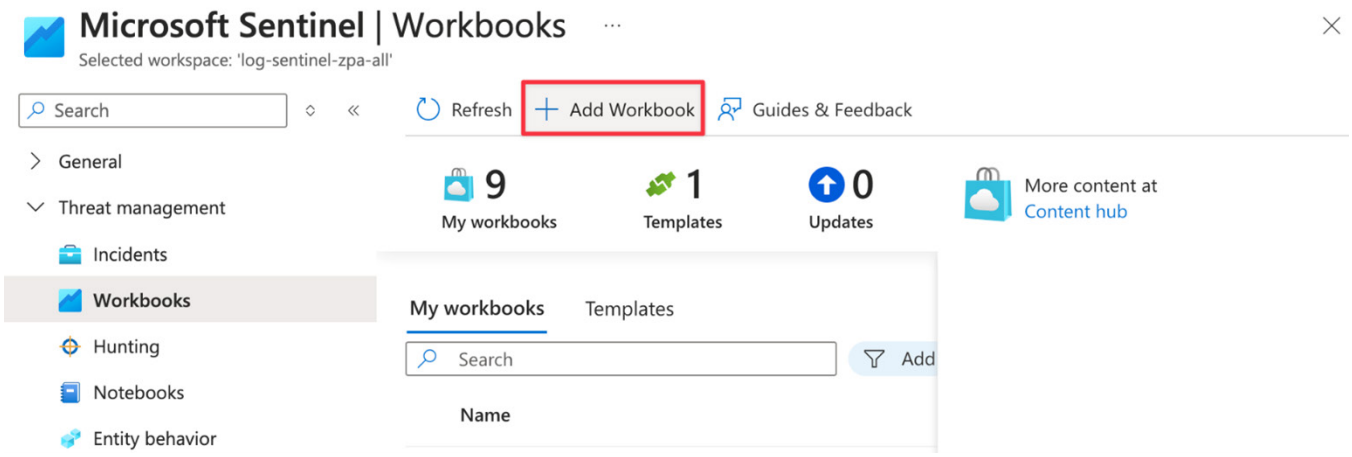


Figure 258. Add Workbook

6. Click **Edit**.



Figure 259. Edit

7. Click the **Advanced Editor** icon.

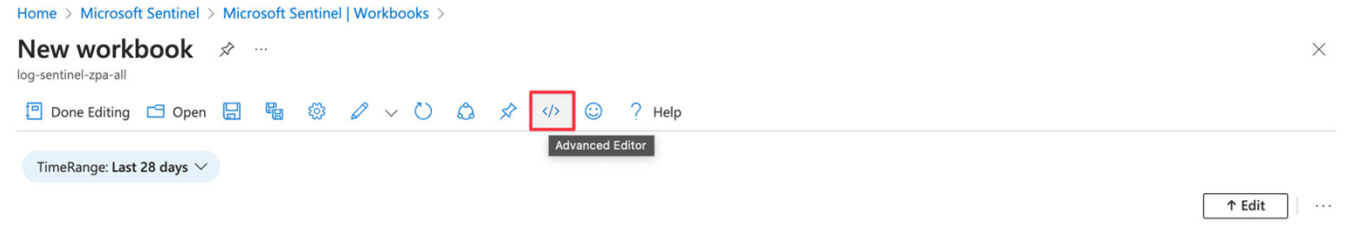


Figure 260. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPAAudit** function to display various graphs.

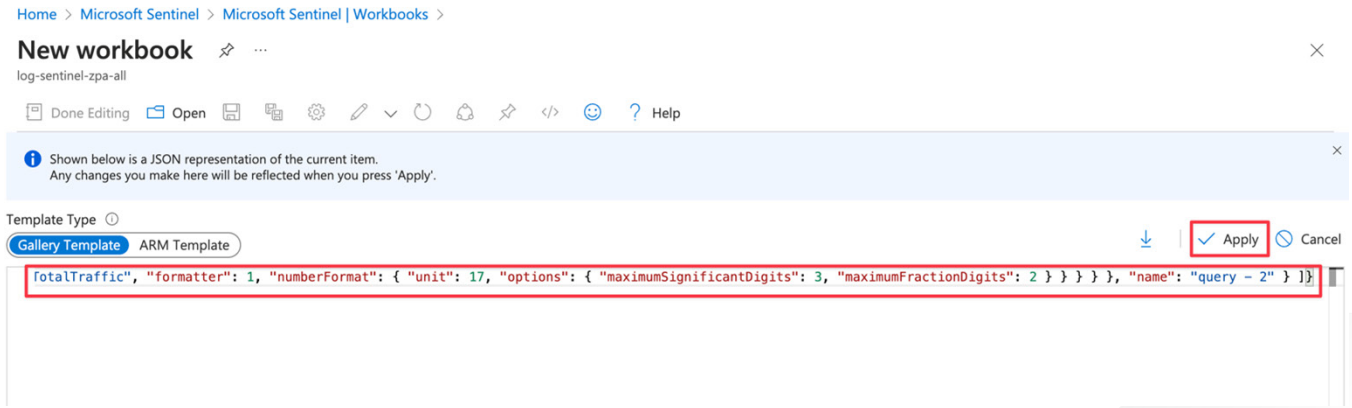


Figure 261. New code text

©2026 Zscaler, Inc. All rights reserved.

9. Click **Done Editing**.

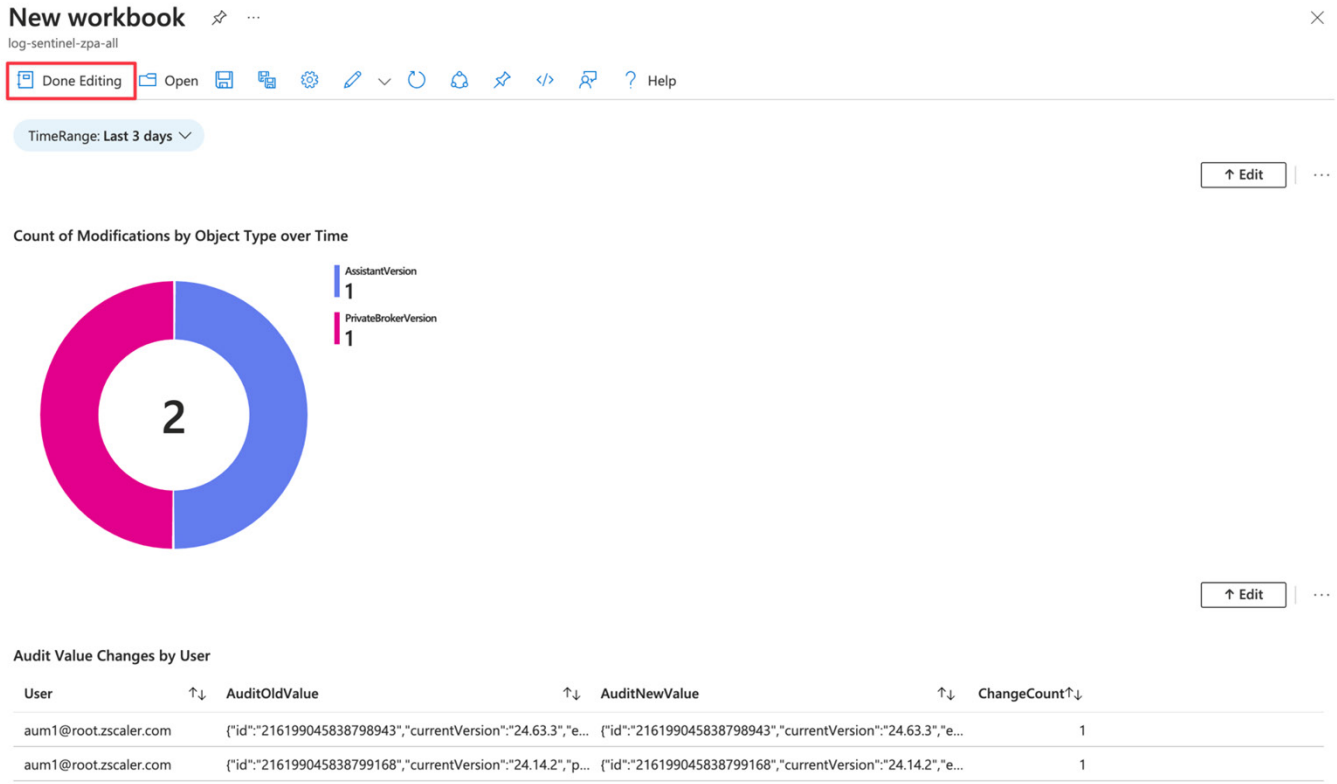


Figure 262. New workbook

10. Click **Save**.



Figure 263. Save workbook

©2026 Zscaler, Inc. All rights reserved.

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

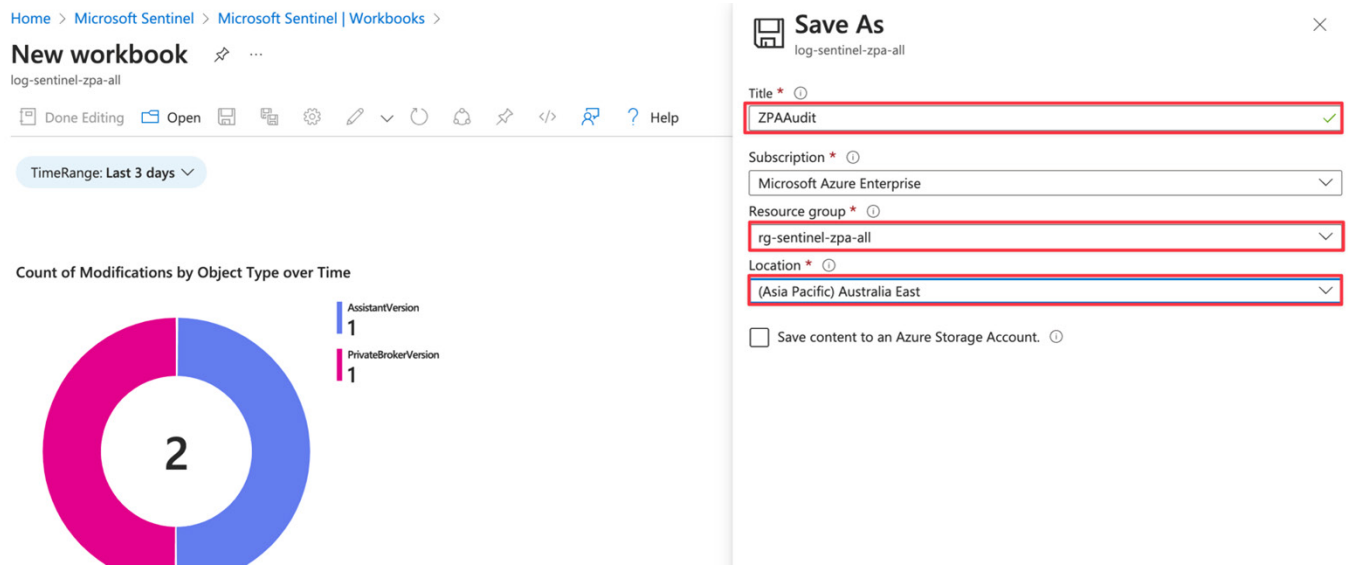


Figure 264. Save the workbook

Configure ZPA App Protection Logs

The following sections describe configuring ZPA App Protection logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

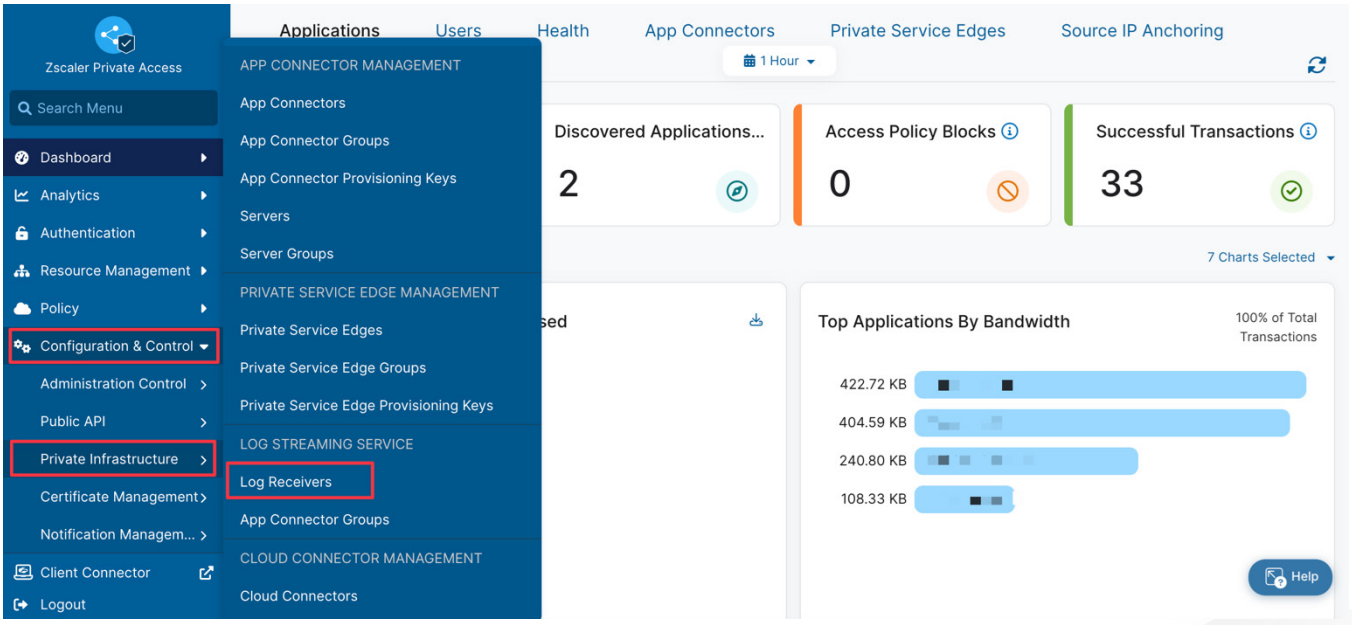


Figure 265. Log Receivers

©2026 Zscaler, Inc. All rights reserved.

2. Click **Add** (+) to add a new **Log Receiver**.

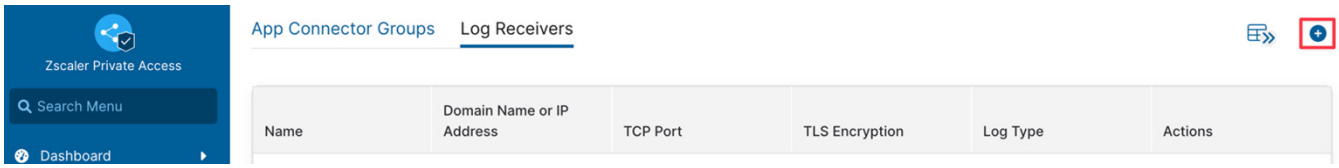


Figure 266. Add new Log Receiver

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

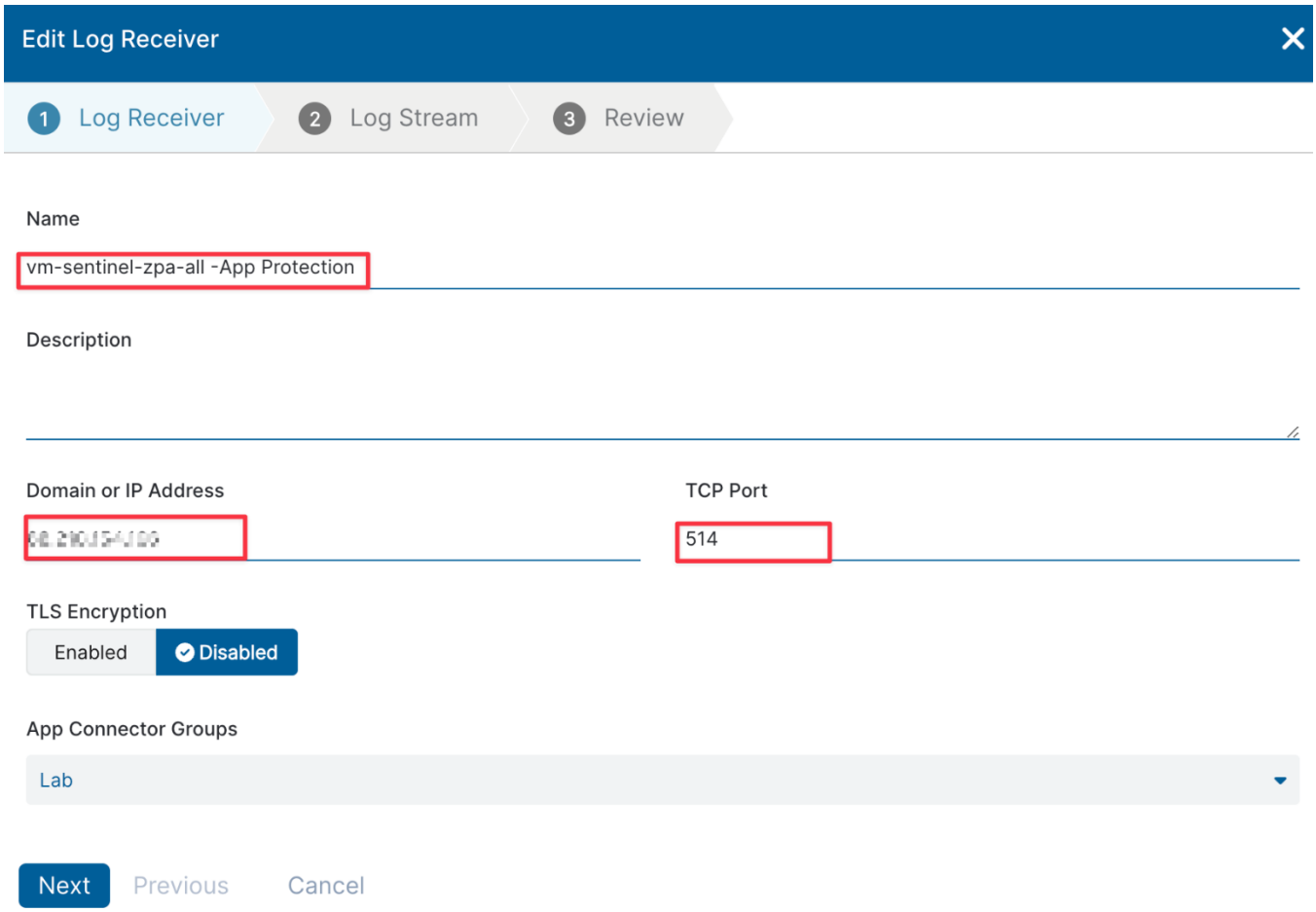


Figure 267. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

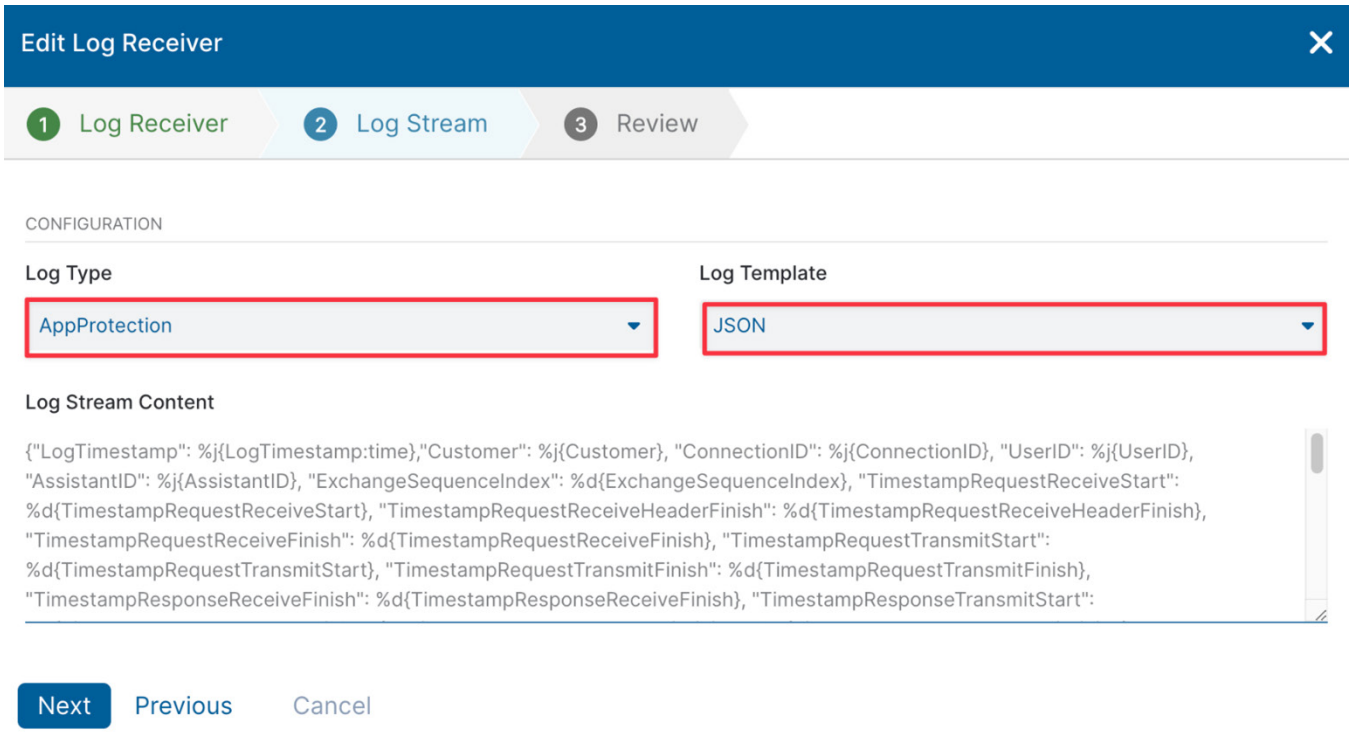


Figure 268. Configure log receiver

7. Click **Save**.

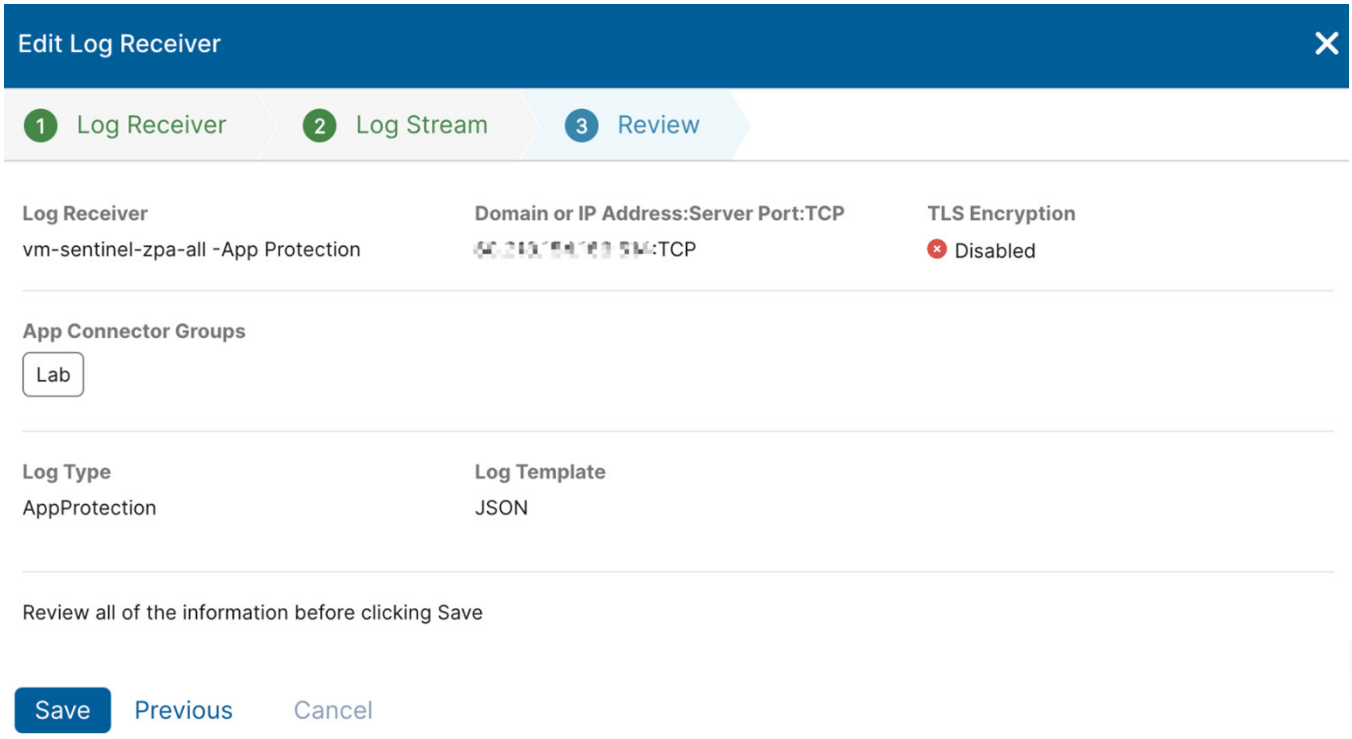


Figure 269. Save log receiver

©2026 Zscaler, Inc. All rights reserved.

Configure the App Protection Log Analytics Workspace Function

The ZPAAppProtection function is used to parse the ZPA App Protection Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics** workspaces.

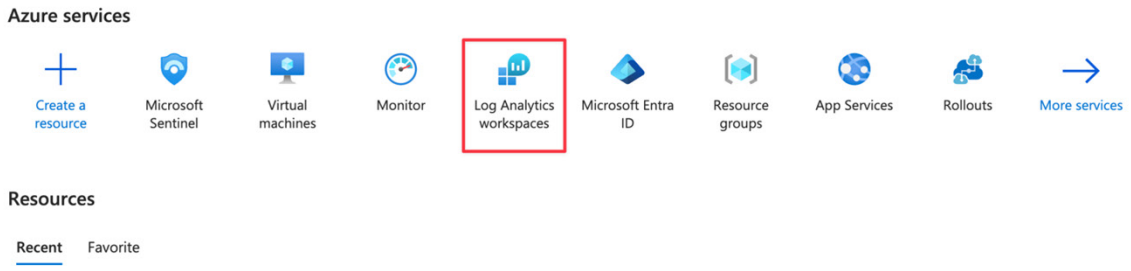


Figure 270. Log Analytics workspaces

3. Select your workspace with the **ZPA User Activity logs**.

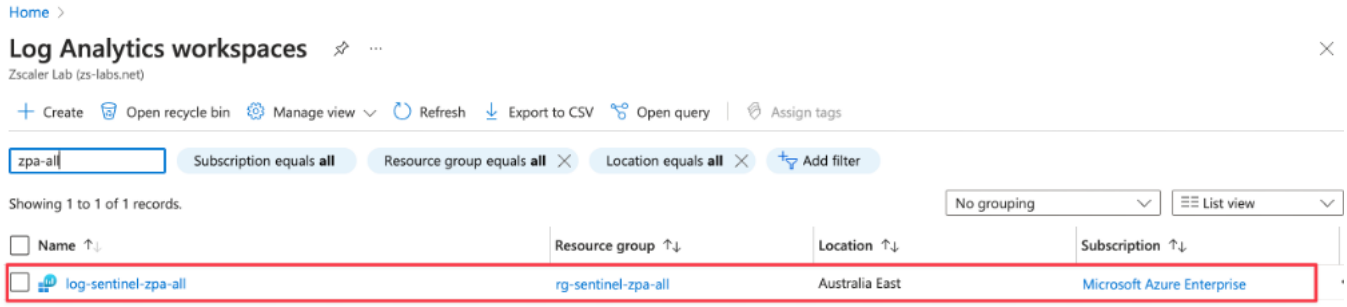


Figure 271. ZPA workspace with logs

4. Click **Logs**.

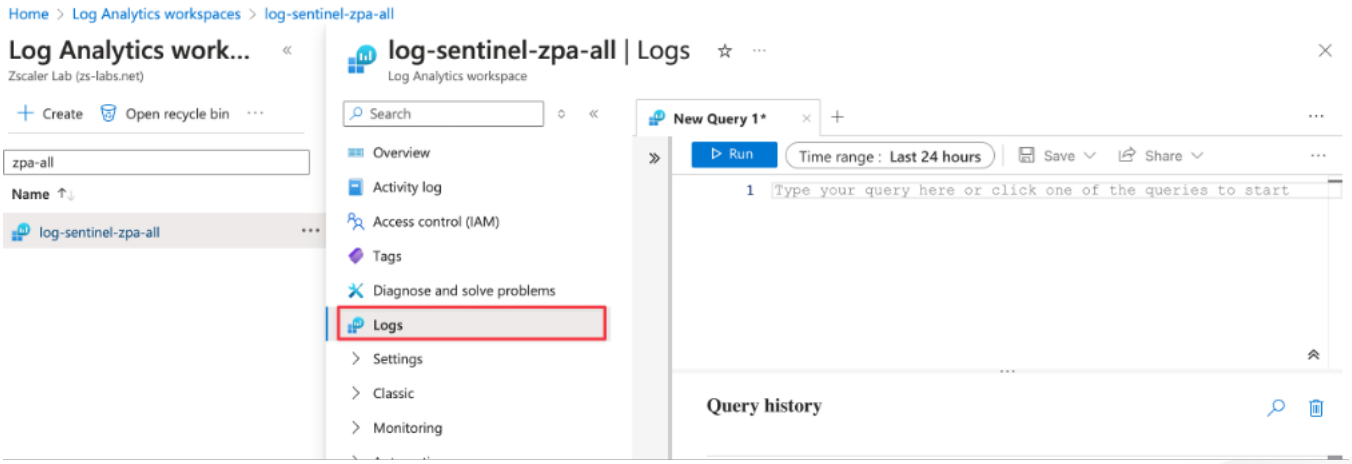


Figure 272. Logs

©2026 Zscaler, Inc. All rights reserved.

5. Paste the content from the `ZPAAppProtection.kql` file into the query window, and select **Save** > **Save as function**.

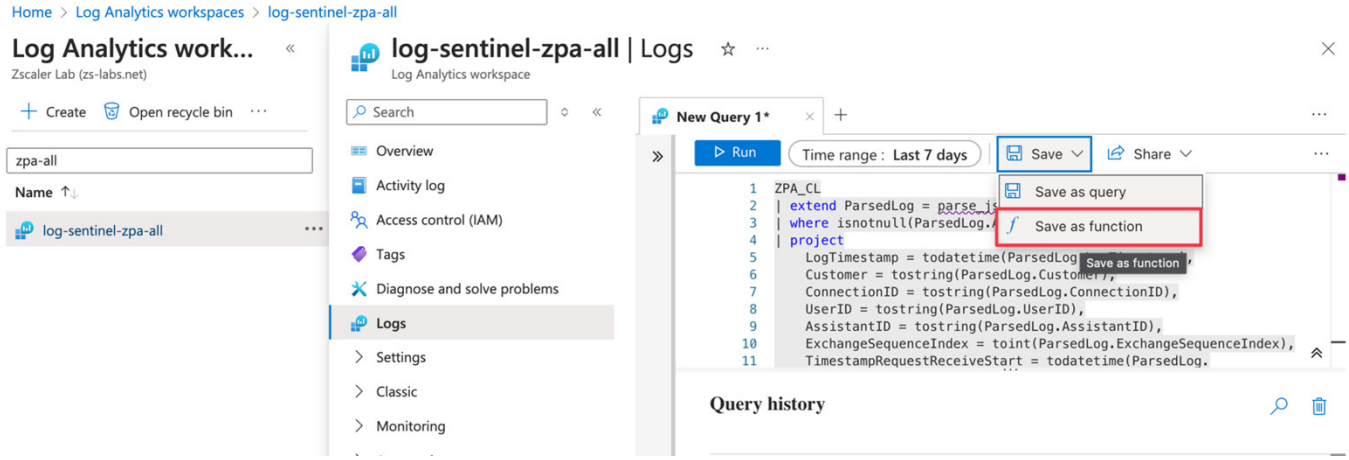


Figure 273. ZPAAppProtection.kql content

6. Enter the **Function name** `ZPAAppProtection`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

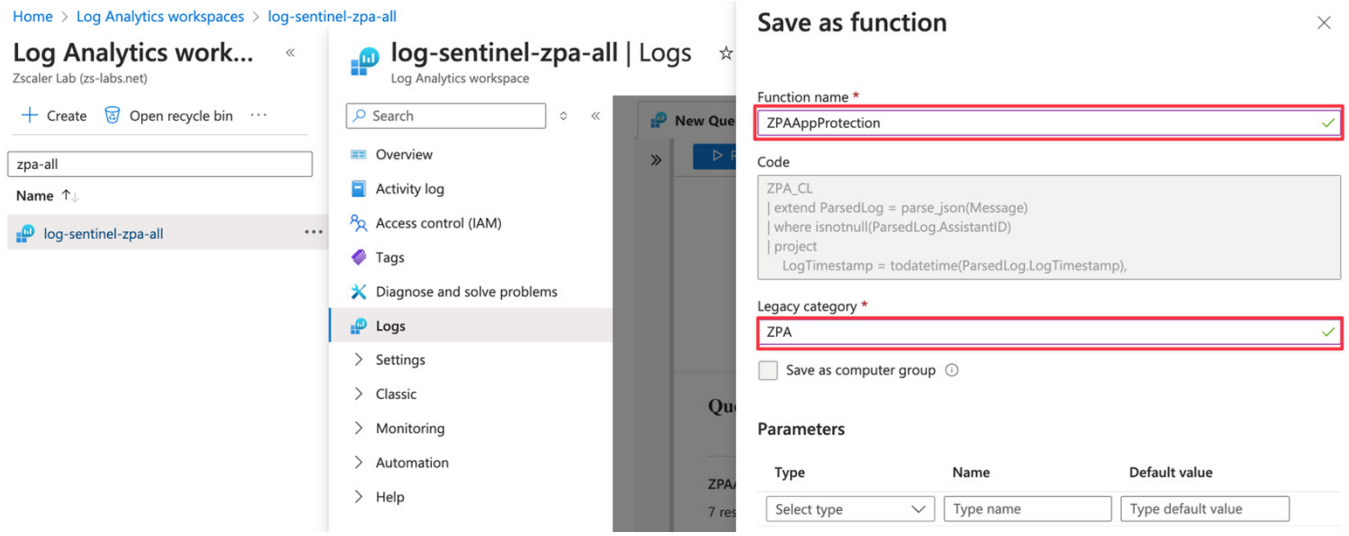


Figure 274. Save ZPAAppProtection

©2026 Zscaler, Inc. All rights reserved.

8. Run queries using the **ZPAAppProtection** function.

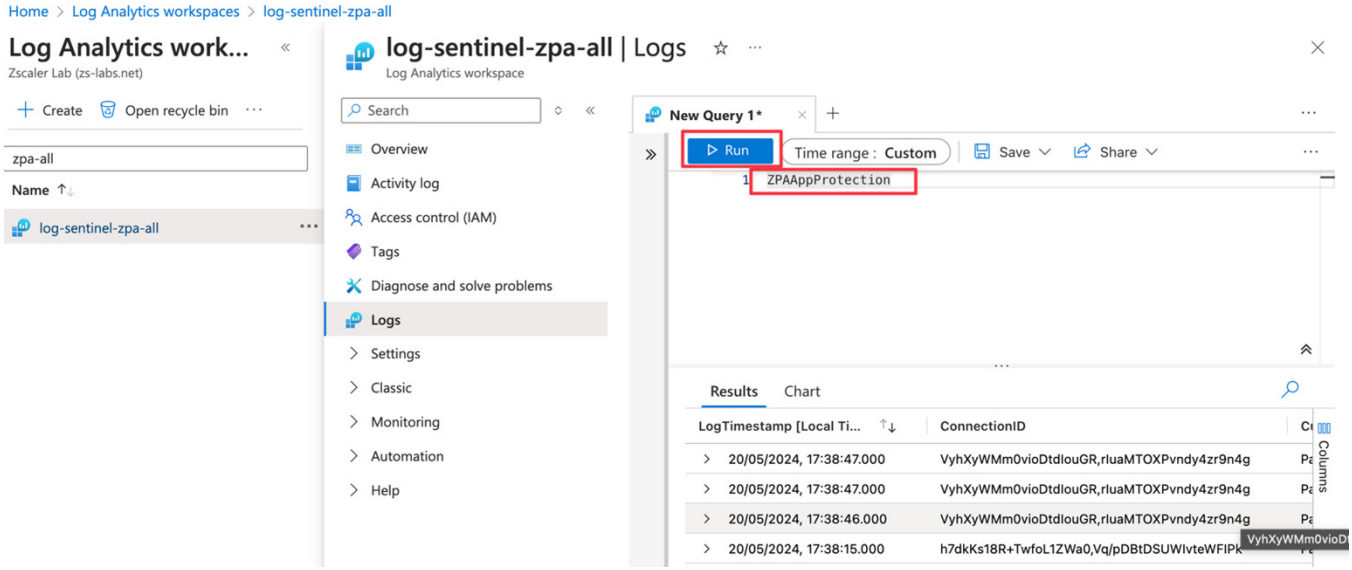


Figure 275. Run queries

Configure the ZPAAppProtection Sentinel Workbook

The ZPAAppProtection workbook is used to graphically display information related to the ZPA App Protection Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your ZPAAppProtection parser.

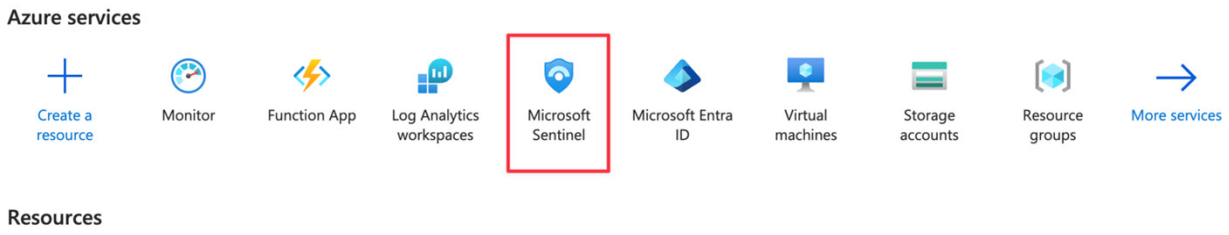


Figure 276. Microsoft Sentinel

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA App Protection logs**.

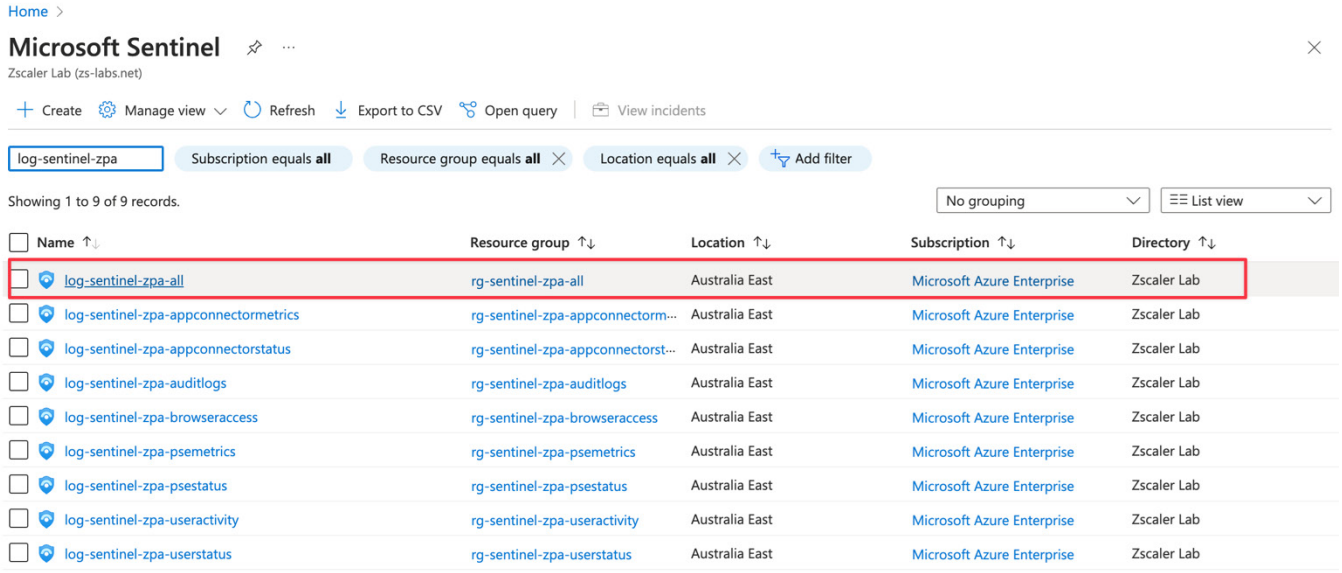


Figure 277. ZPA workspace with logs

4. Click **Workbooks** under **Threat management**.

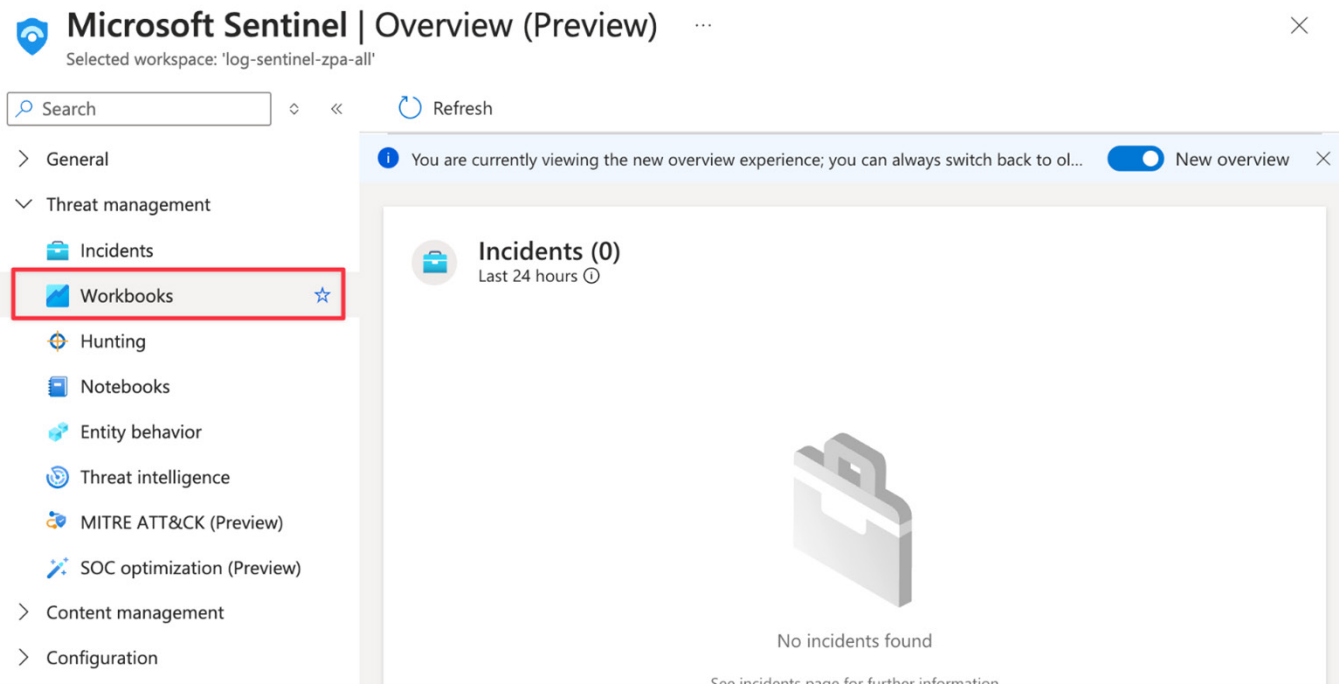


Figure 278. Workbooks

©2026 Zscaler, Inc. All rights reserved.

5. Click **Add Workbook**.

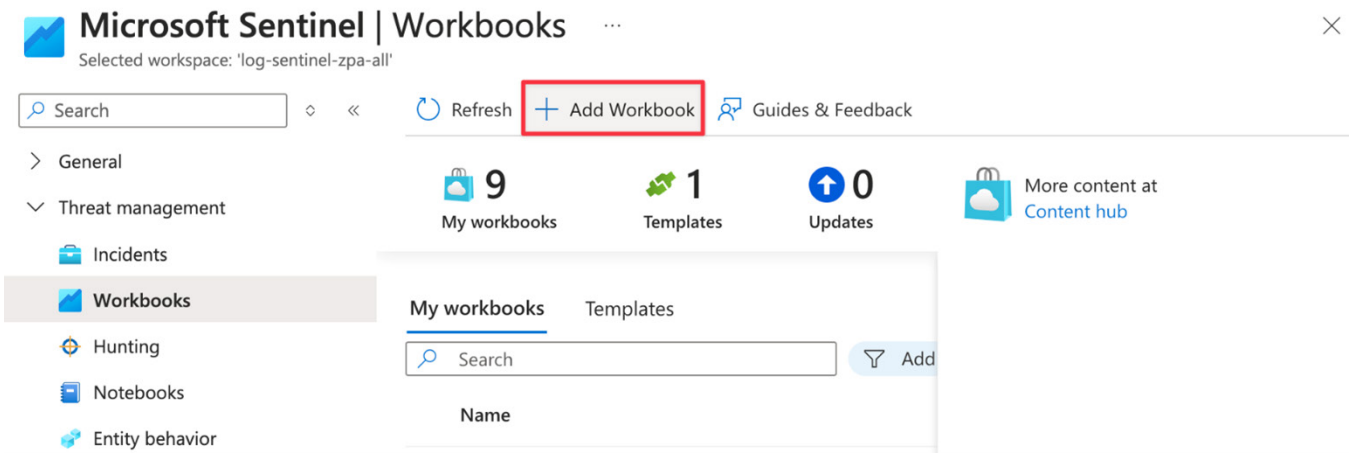


Figure 279. Add Workbook

6. Click **Edit**.



Figure 280. Edit

7. Click the **Advanced Editor** icon.

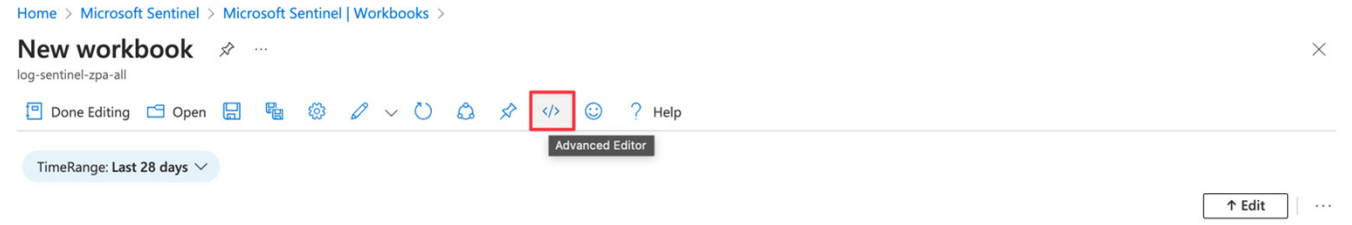


Figure 281. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPAAppProtection** function to display various graphs.

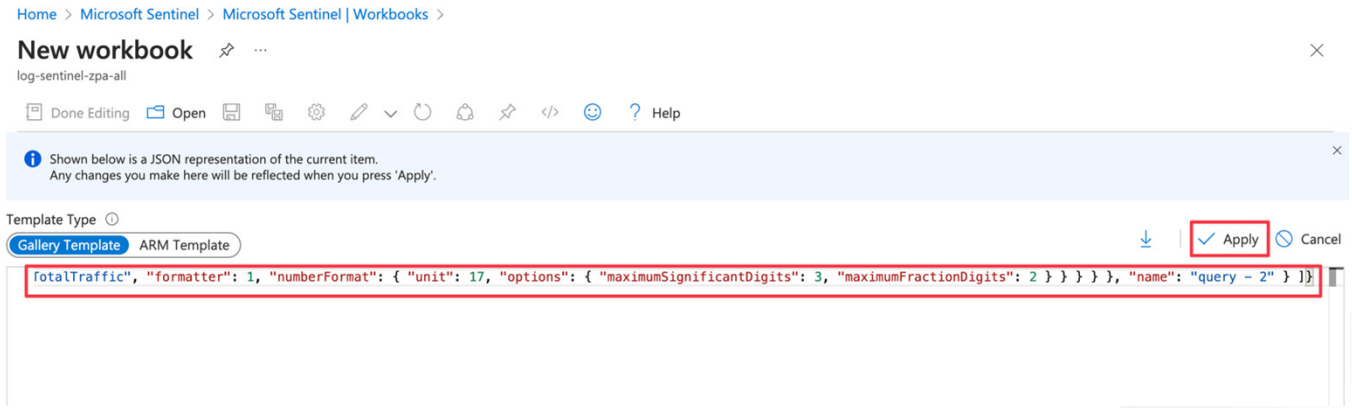


Figure 282. New code text

©2026 Zscaler, Inc. All rights reserved.

9. Click **Done Editing**.

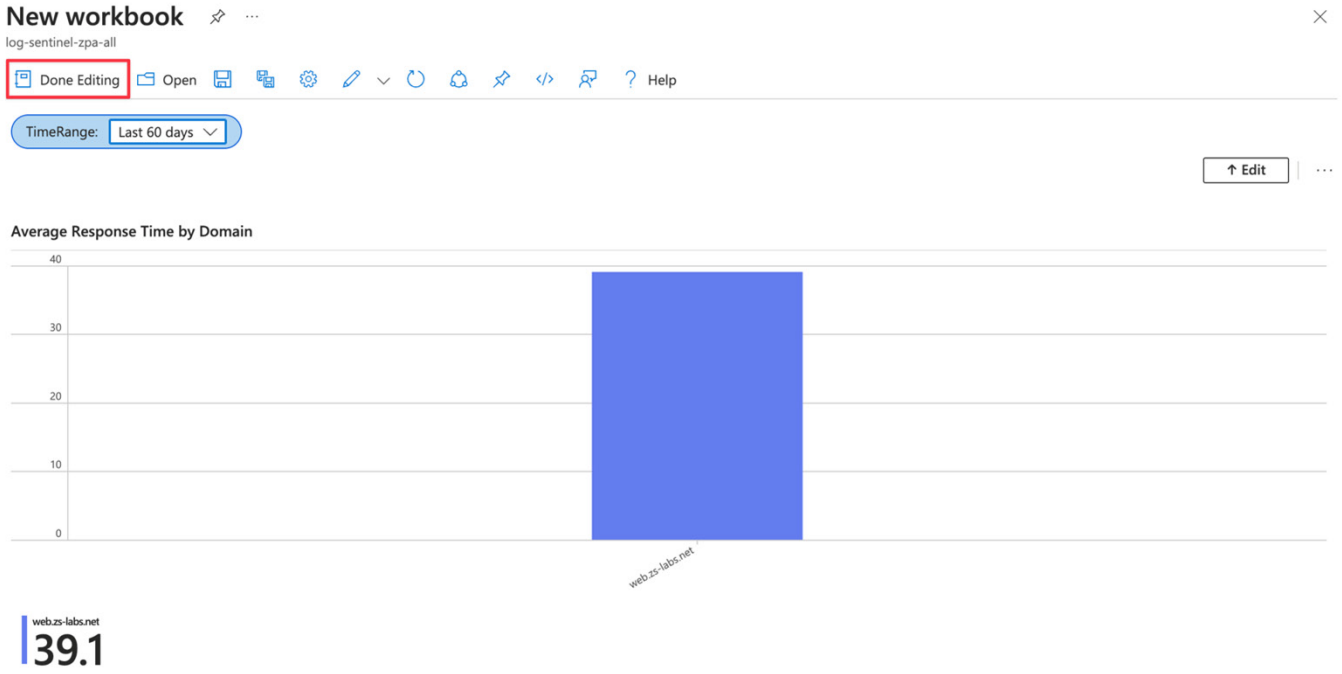


Figure 283. New workbook

10. Click **Save**.



Figure 284. Save workbook

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

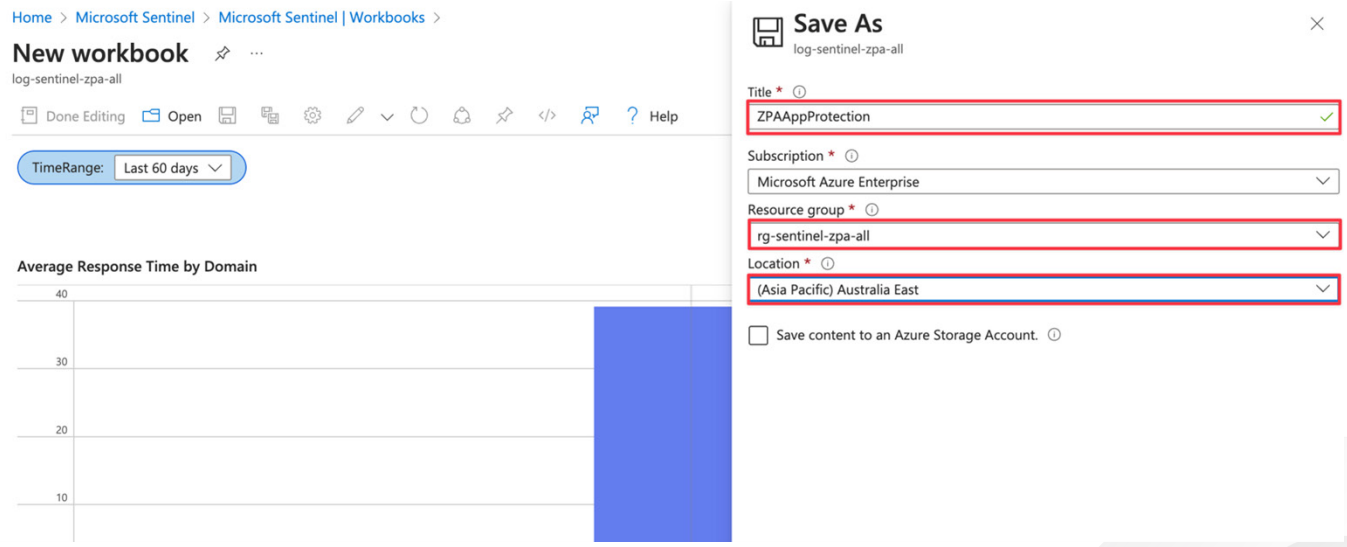


Figure 285. Save the workbook

©2026 Zscaler, Inc. All rights reserved.

Configure ZPA App Connector Status Logs

The following sections describe how to configure ZPA App Connector Status logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

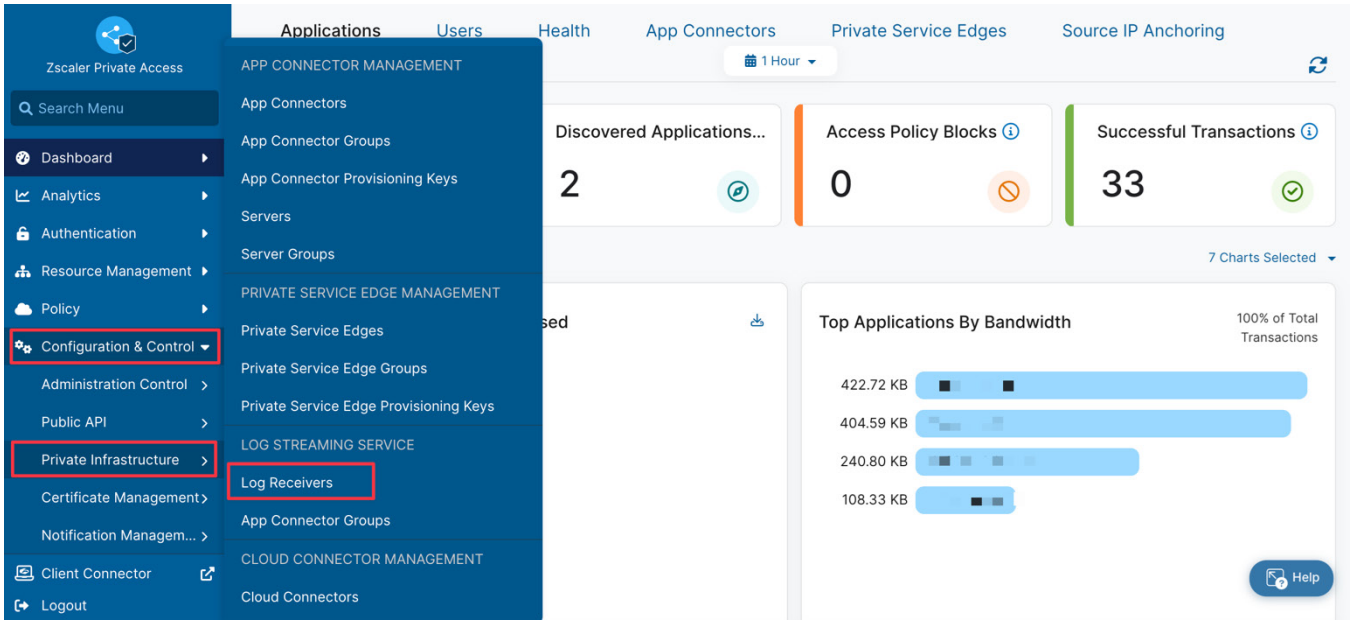


Figure 286. Log Receivers

2. Click **Add** (+) to add a new Log Receiver.

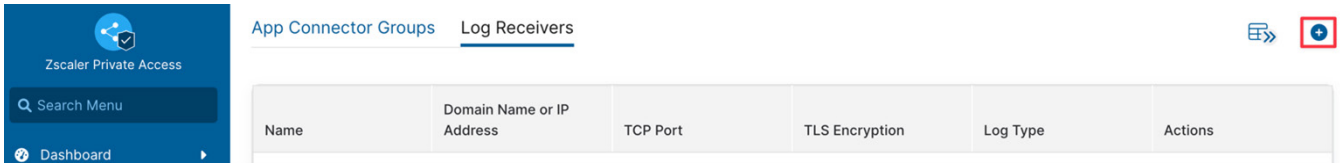


Figure 287. Add new Log Receiver

©2026 Zscaler, Inc. All rights reserved.

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

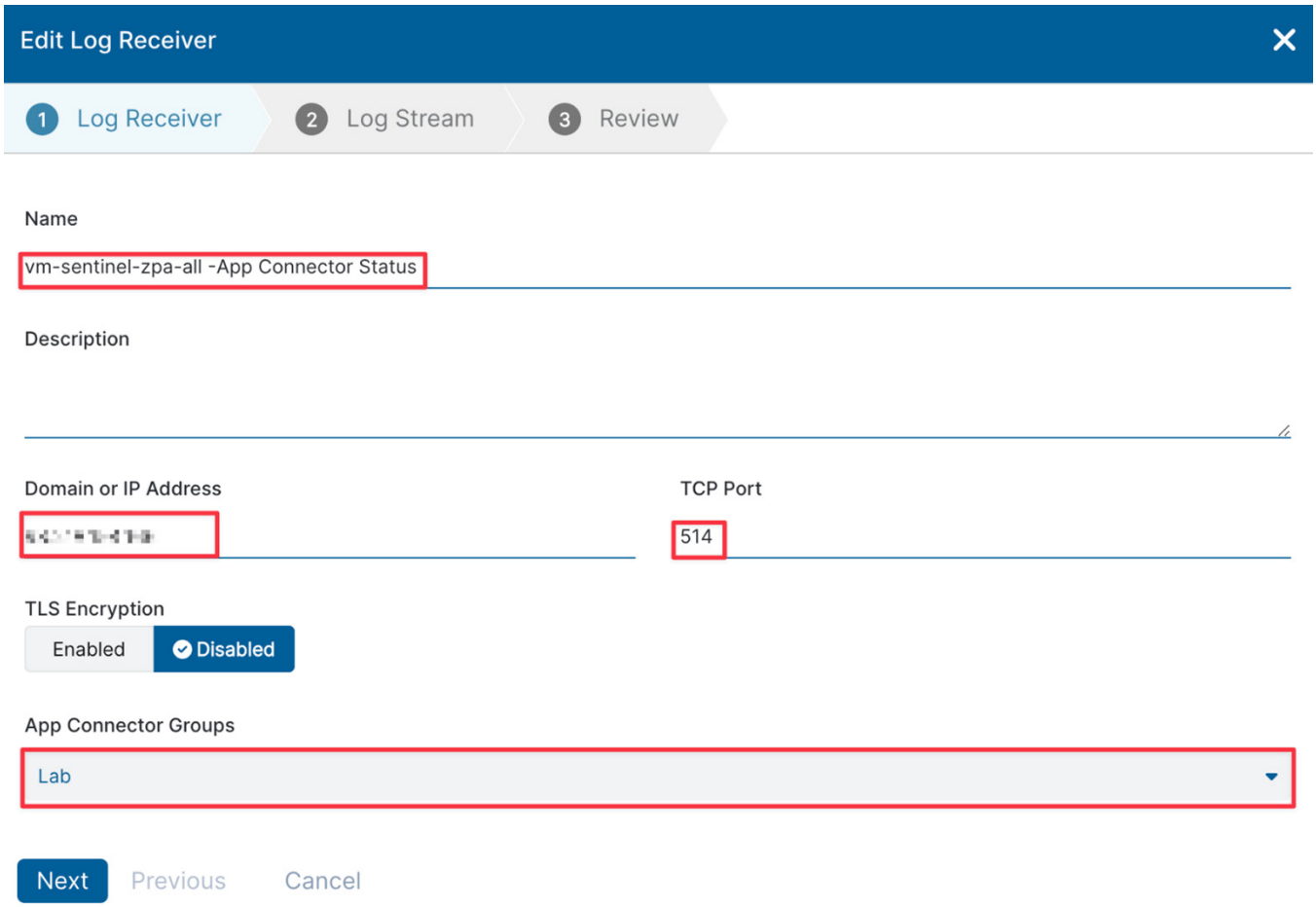


Figure 288. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

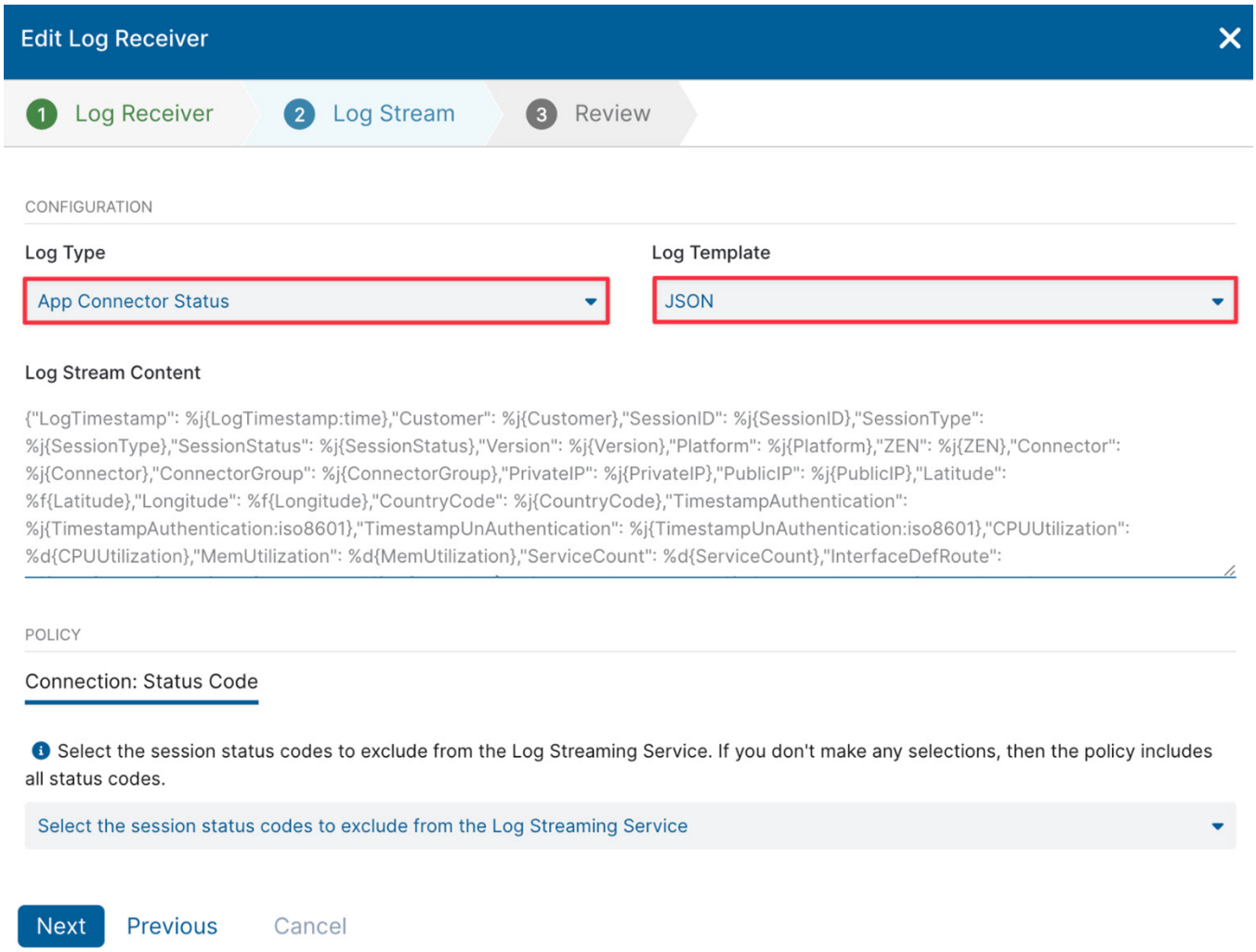


Figure 289. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

7. Click **Save**.

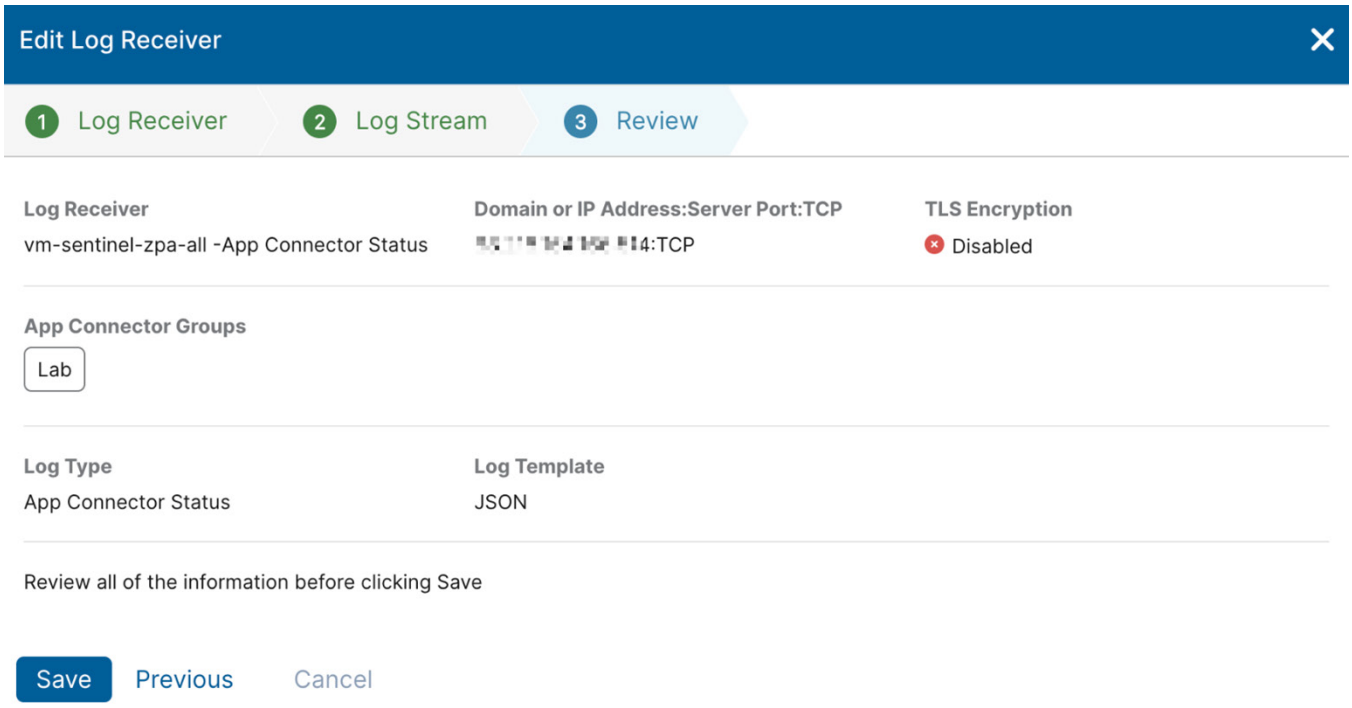


Figure 290. Save log receiver

Configure the ZPAAppConnectorStatus Log Analytics workspace function

The ZPAAppConnectorStatus function is used to parse the ZPA App Connector Status Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

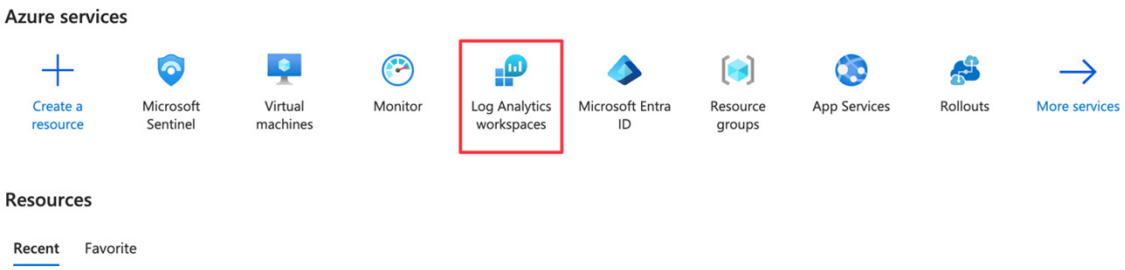


Figure 291. Log Analytics workspaces

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA User Activity logs**.

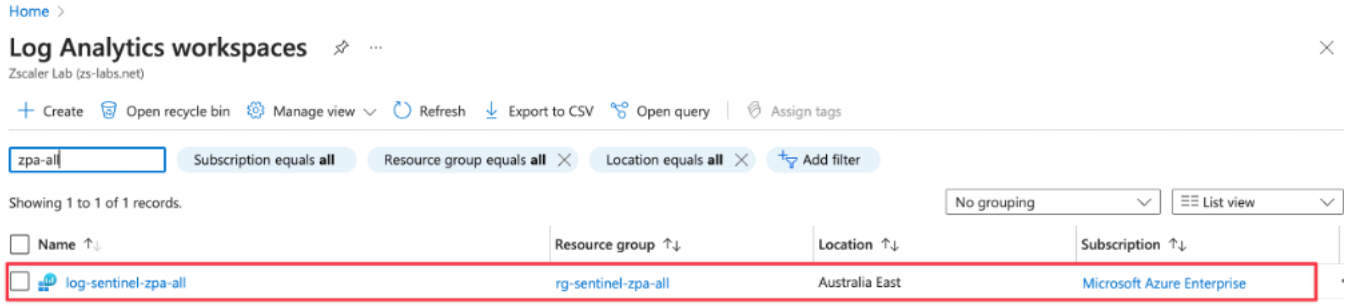


Figure 292. ZPA workspace with logs

4. Click **Logs**.

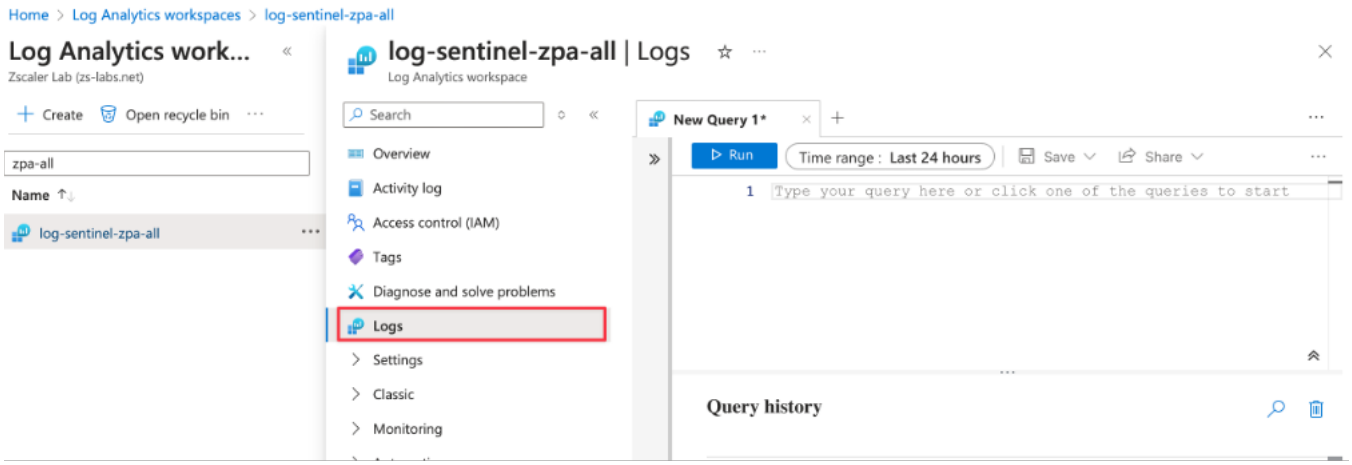


Figure 293. Logs

5. Paste the content from the `ZPAAppConnectorStatus.kql` file into the query window, and select **Save** > **Save as function**.

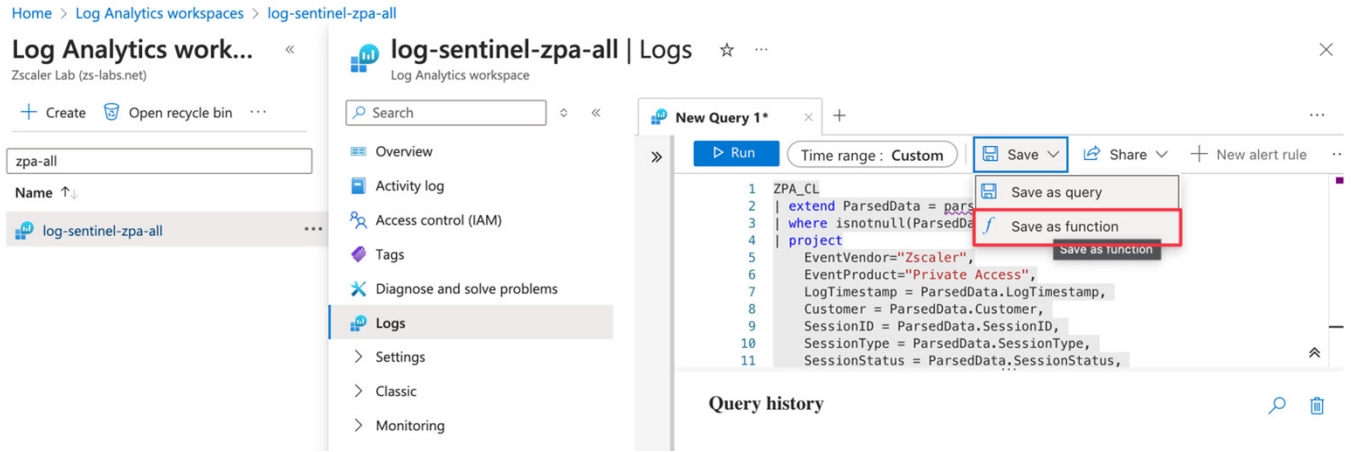


Figure 294. ZPAAppConnectorStatus.kql content

©2026 Zscaler, Inc. All rights reserved.

6. Enter the **Function name** `ZPAAppConnectorStatus`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

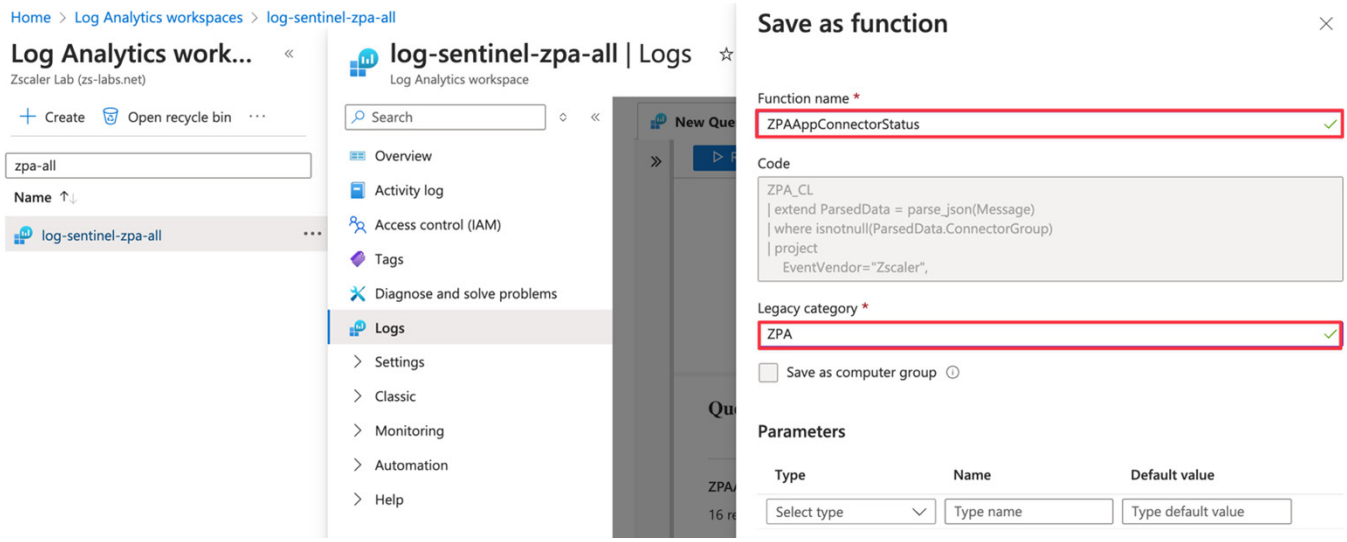


Figure 295. Save ZPAAppConnectorStatus

8. Run queries using the **ZPAAppConnectorStatus** function.

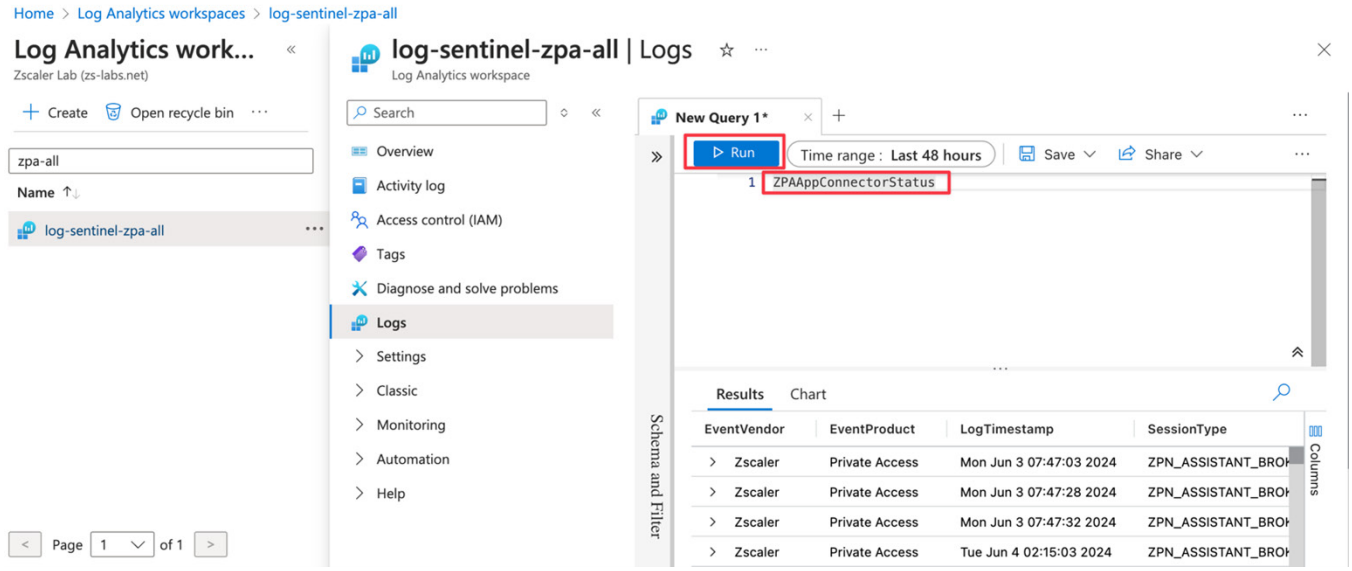


Figure 296. Run queries

©2026 Zscaler, Inc. All rights reserved.

Configure the ZPAAppConnectorStatus Sentinel Workbook

The ZPAAppConnectorStatus workbook is used to graphically display information related to the ZPA App Connector Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your **ZPAAppConnectorStatus** parser.

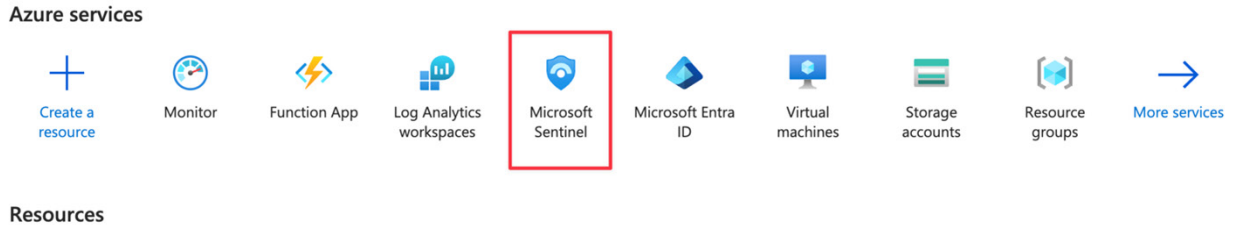


Figure 297. Microsoft Sentinel

3. Select your workspace with the **ZPA App Connector Status logs**.

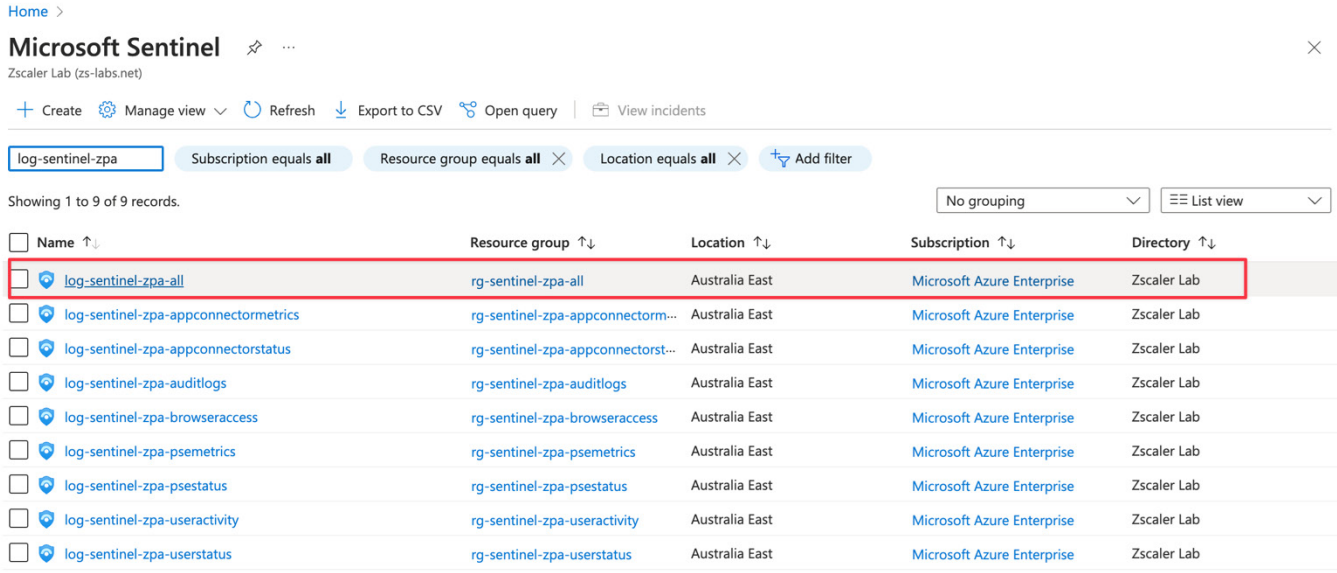


Figure 298. ZPA workspace with logs

©2026 Zscaler, Inc. All rights reserved.

4. Click **Workbooks** under **Threat management**.

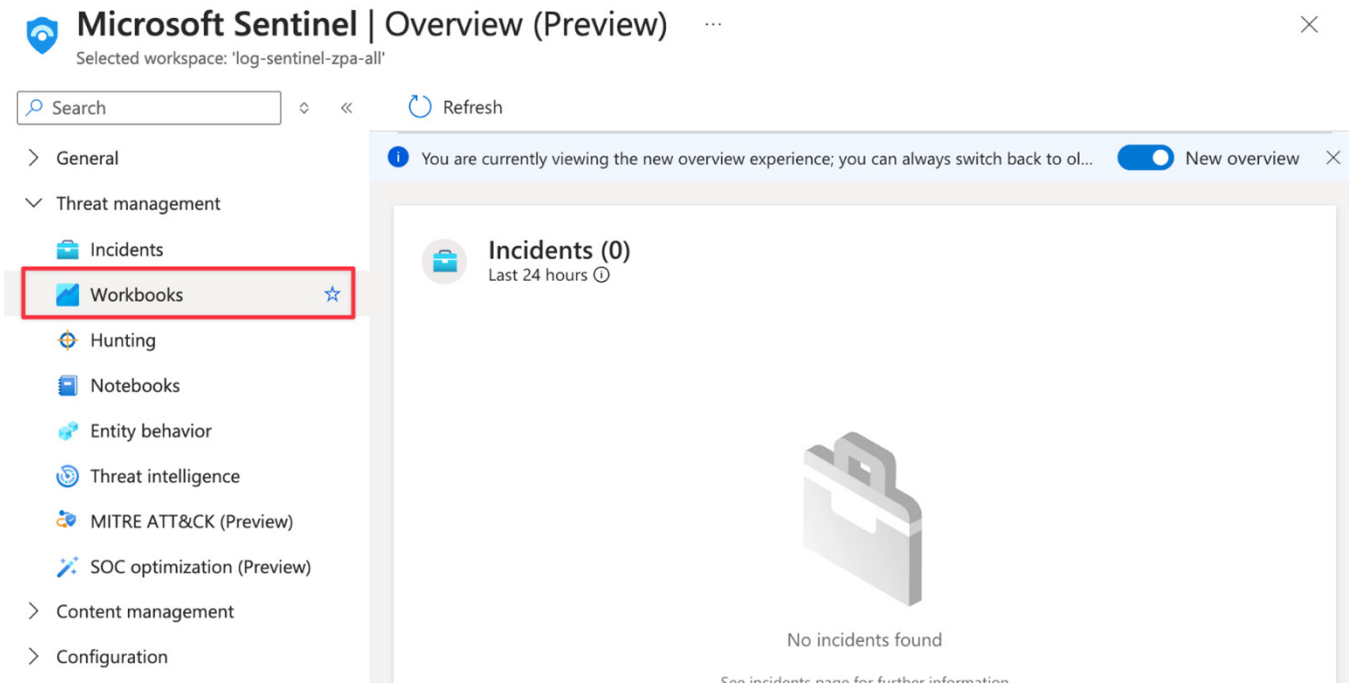


Figure 299. Workbooks

5. Click **Add Workbook**.

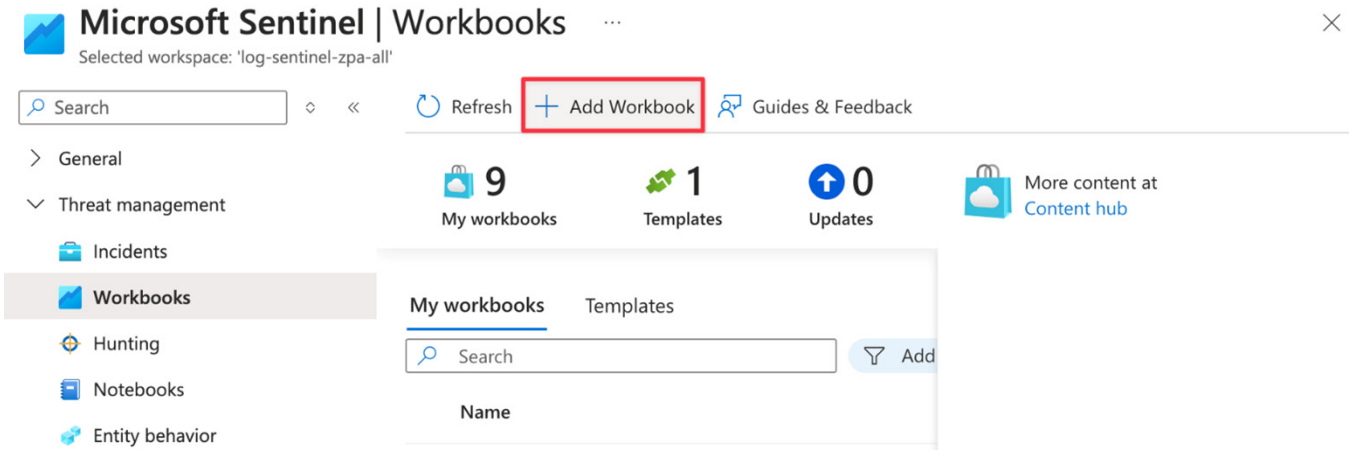


Figure 300. Add Workbook

6. Click **Edit**.



Figure 301. Edit

©2026 Zscaler, Inc. All rights reserved.

7. Click the **Advanced Editor** icon.

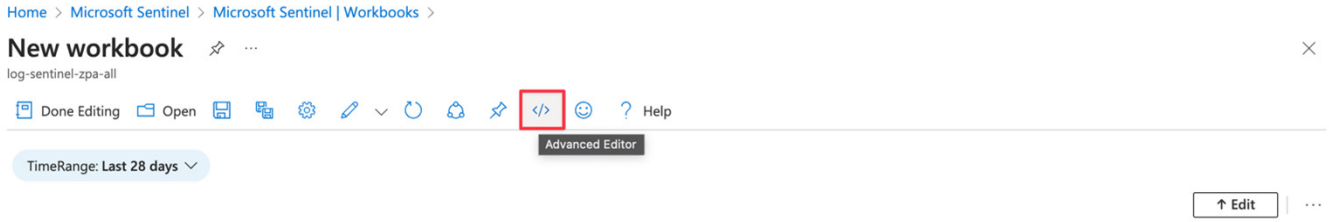


Figure 302. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPAAppConnectorStatus** function to display various graphs.

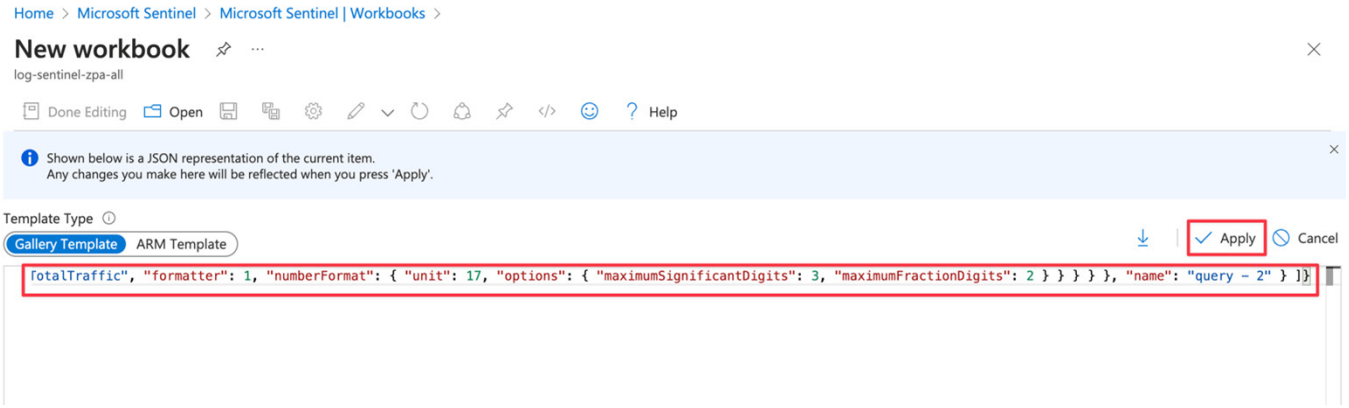


Figure 303. New code text

9. Click **Done Editing**.

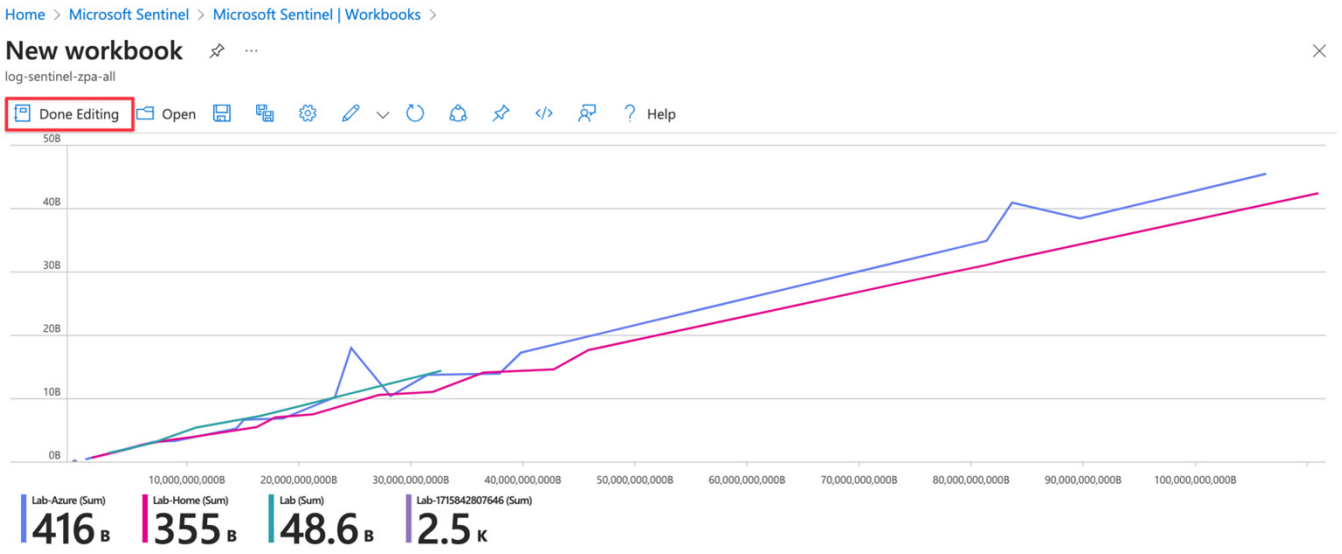


Figure 304. New workbook

©2026 Zscaler, Inc. All rights reserved.

10. Click **Save**.



Figure 305. Save workbook

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

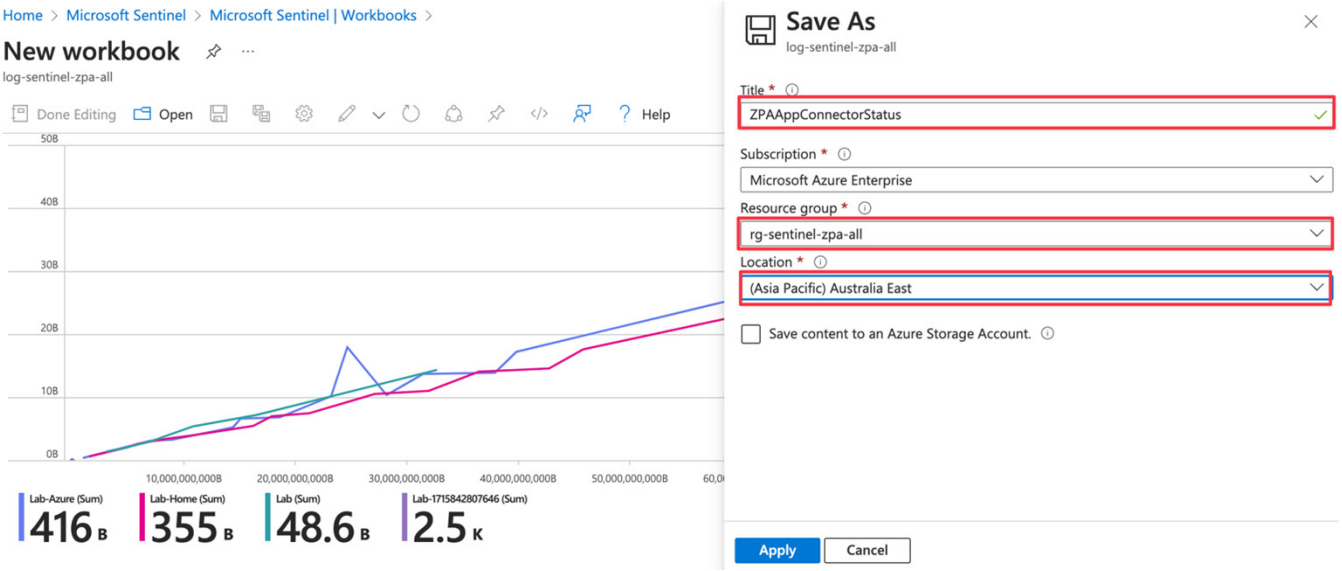


Figure 306. Save the workbook

©2026 Zscaler, Inc. All rights reserved.

Configure ZPA App Connector Metrics Logs

The following sections describe how to configure ZPA App Connector Metrics logs.

Configure Log Receiver

To send logs to the data connector:

1. Log in to the ZPA Admin Portal, then go to **Configuration & Control** > **Private Infrastructure** > **Log Receivers**.

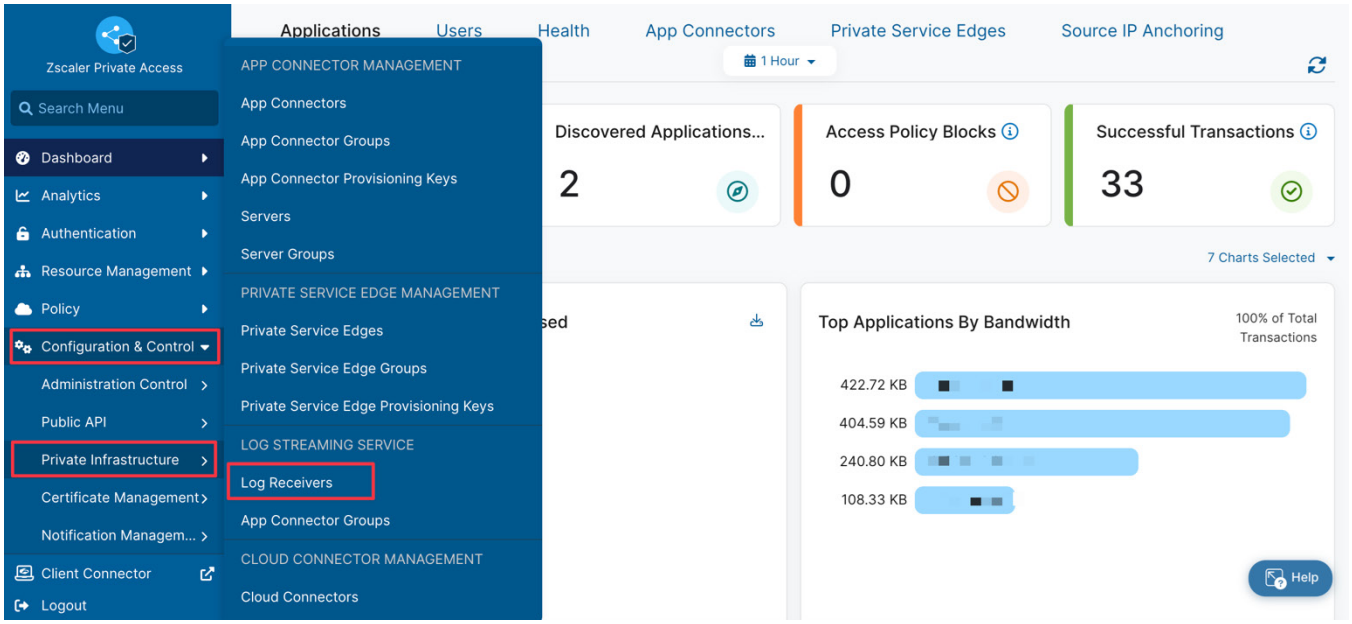


Figure 307. Log Receivers

2. Click **Add** (+) to add a new Log Receiver.

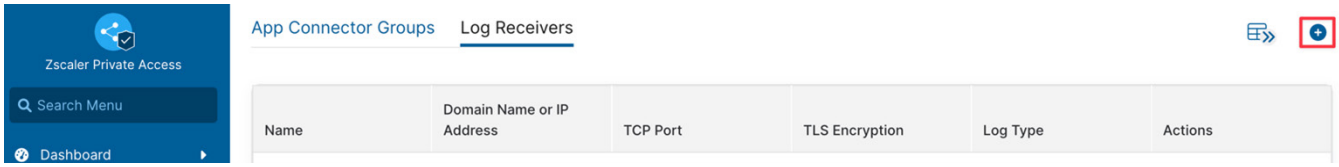


Figure 308. Add new Log Receiver

©2026 Zscaler, Inc. All rights reserved.

3. Enter a Log Receiver **Name**, **Domain or IP Address**, and **TCP Port**, then select which **App Connector Groups** to use.
4. Click **Next**.

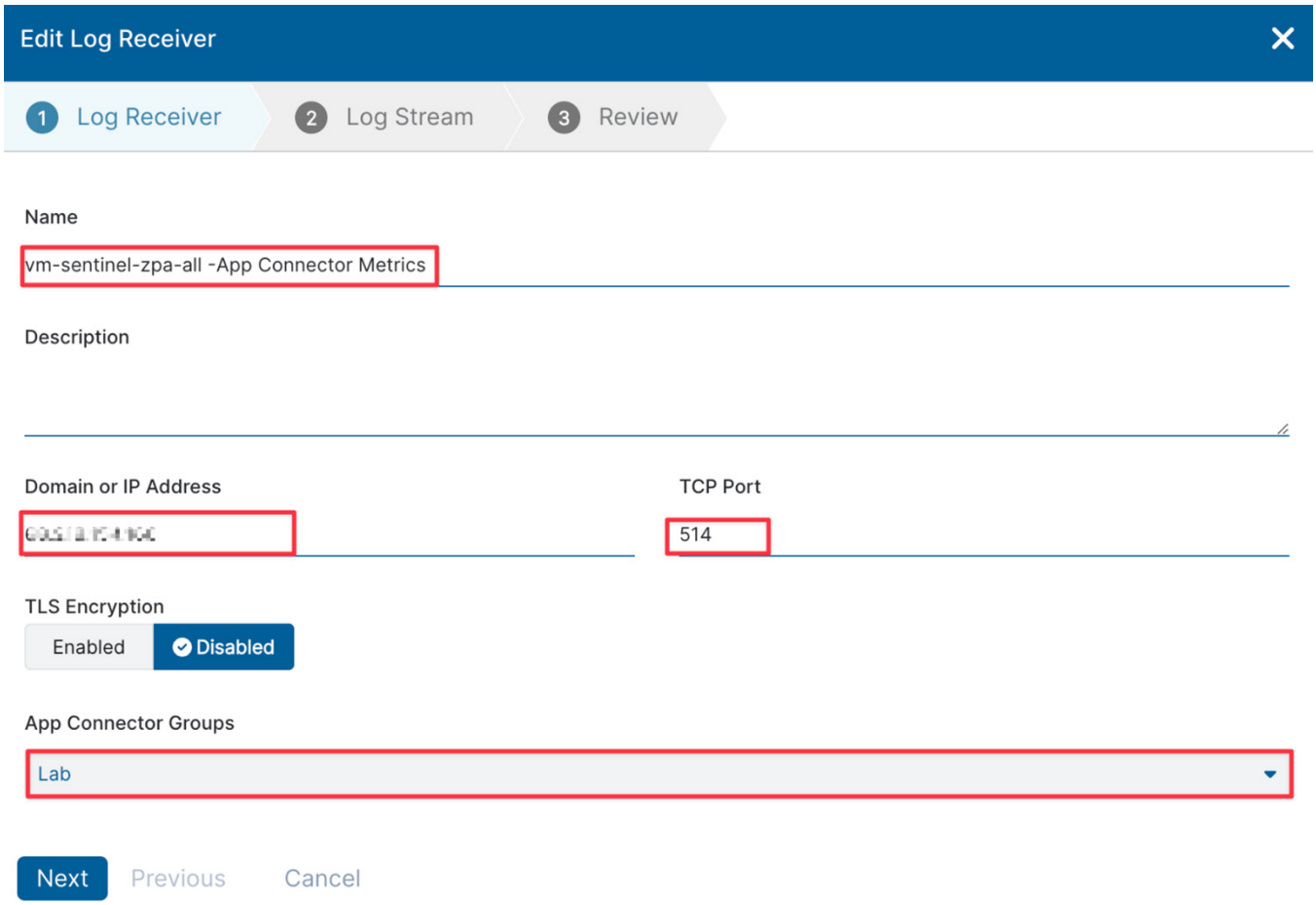


Figure 309. Configure log receiver

©2026 Zscaler, Inc. All rights reserved.

5. Select the **Log Type**, and ensure the **Log Template** is set to **JSON**.
6. Click **Next**.

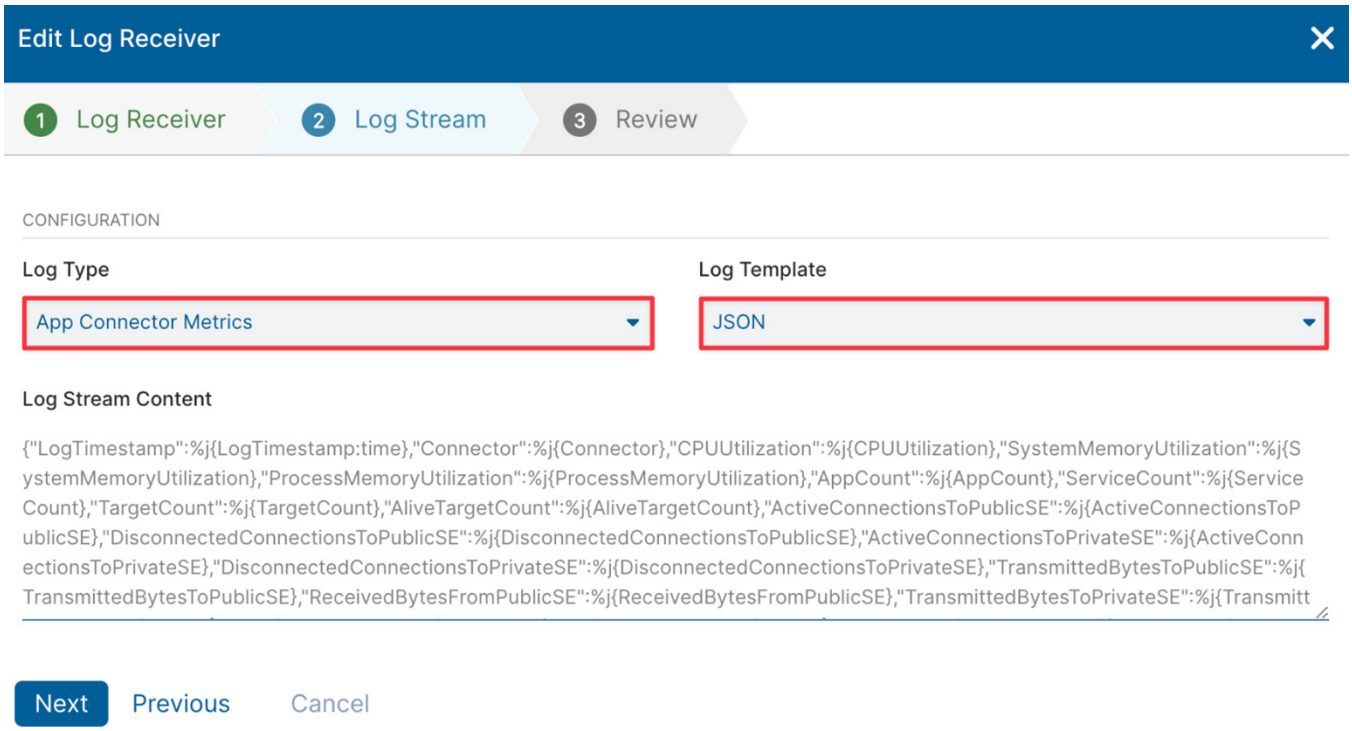


Figure 310. Configure log receiver

7. Click **Save**.

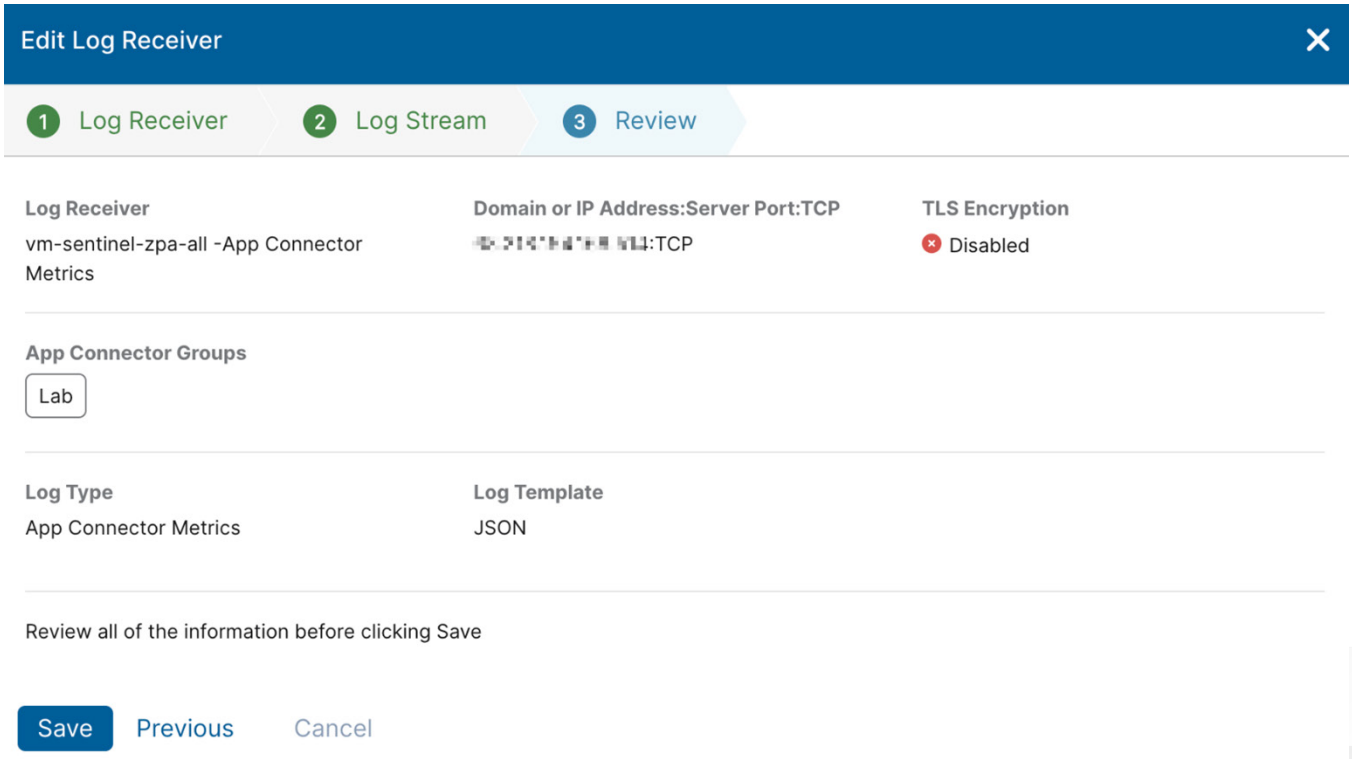


Figure 311. Save log receiver

©2026 Zscaler, Inc. All rights reserved.

Configure the ZPAAppConnectorMetrics Log Analytics Workspace Function

The ZPAAppConnectorMetrics function is used to parse the ZPA App Connector Metrics Log Feed to assist with query generation and investigation.

To use the function:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Log Analytics workspaces**.

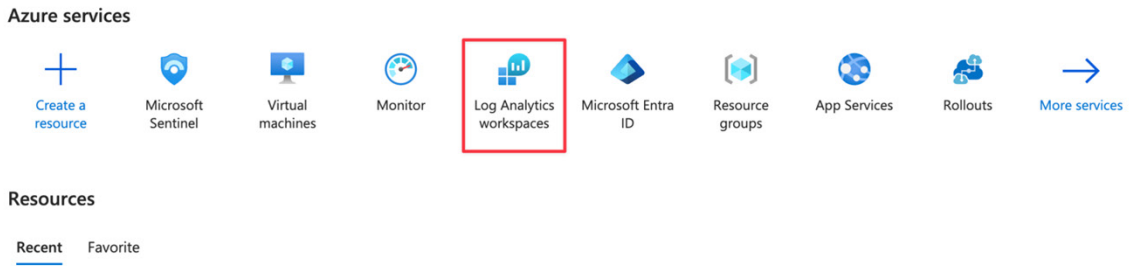


Figure 312. Log Analytics workspaces

3. Select your workspace with the **ZPA User Activity logs**.

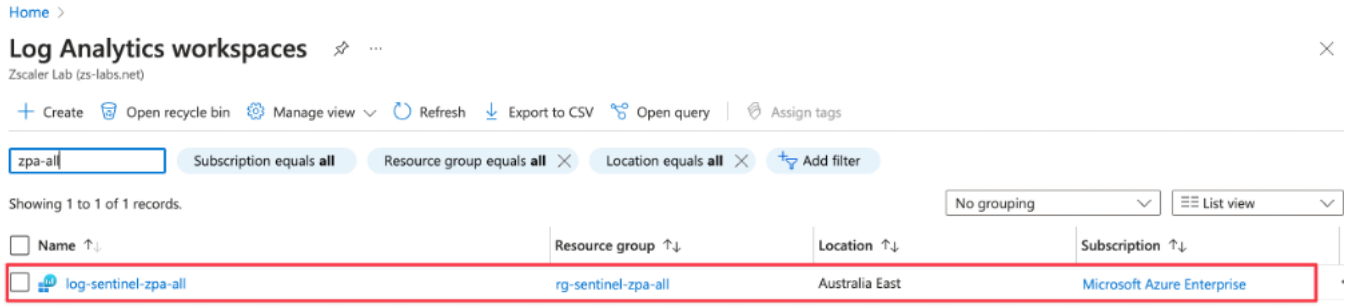


Figure 313. ZPA workspace with logs

4. Click **Logs**.

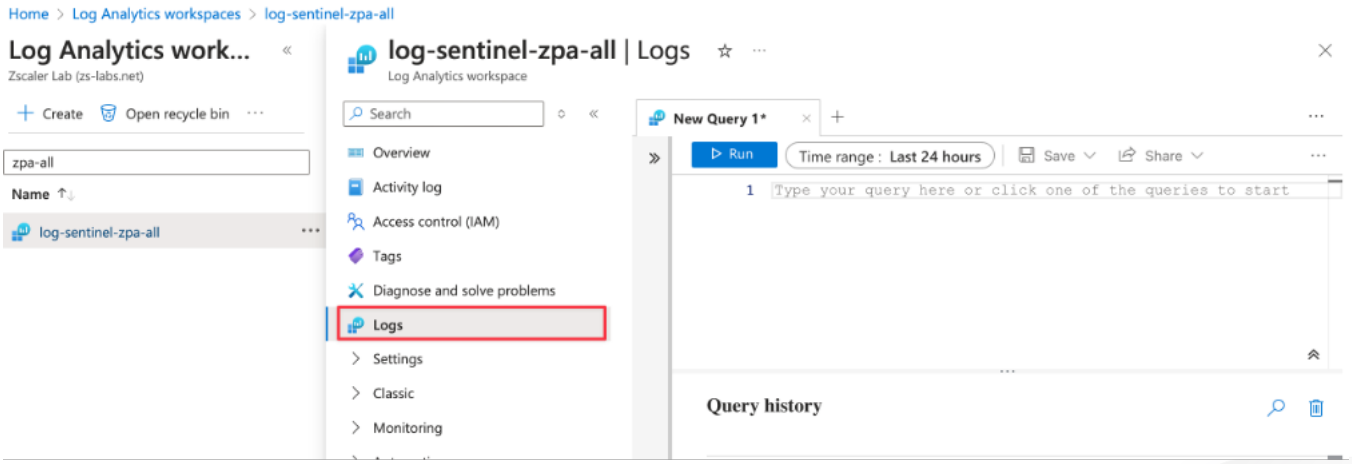


Figure 314. Logs

©2026 Zscaler, Inc. All rights reserved.

5. Paste the content from the `ZPAAppConnectorMetrics.kql` file into the query window and select **Save** > **Save as function**.

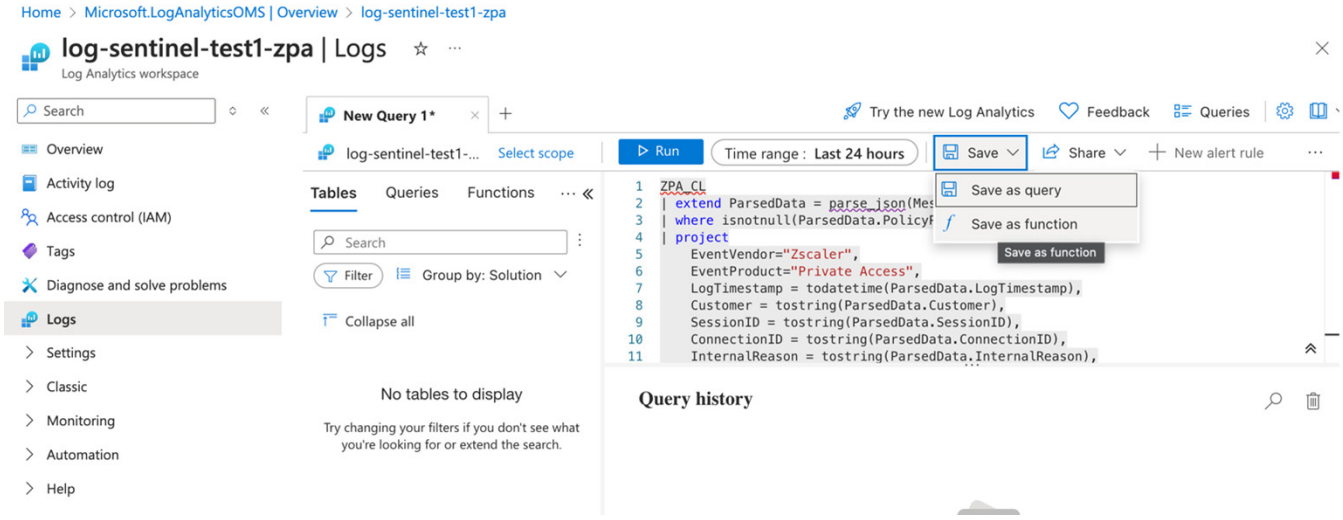


Figure 315. ZPAAppConnectorMetrics.kql content

6. Enter the **Function name** `ZPAAppConnectorMetrics`, and enter a **Legacy category** of `ZPA`.
7. Click **Save**.

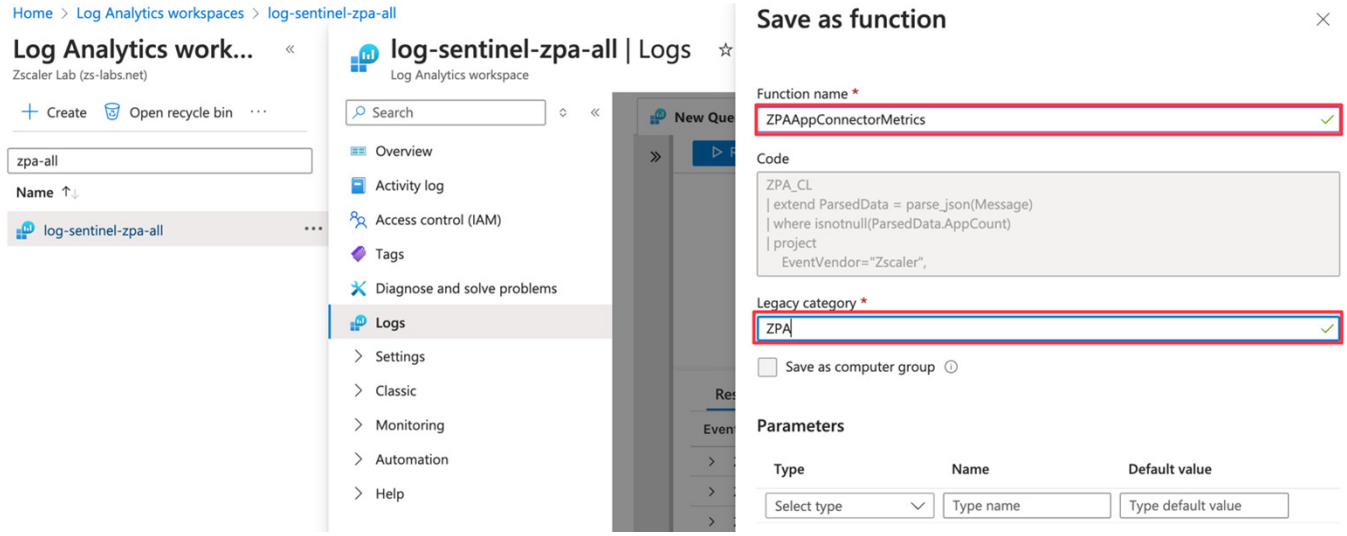


Figure 316. Save ZPAAppConnectorMetrics

©2026 Zscaler, Inc. All rights reserved.

8. Run queries using the **ZPAAppConnectorMetrics** function.

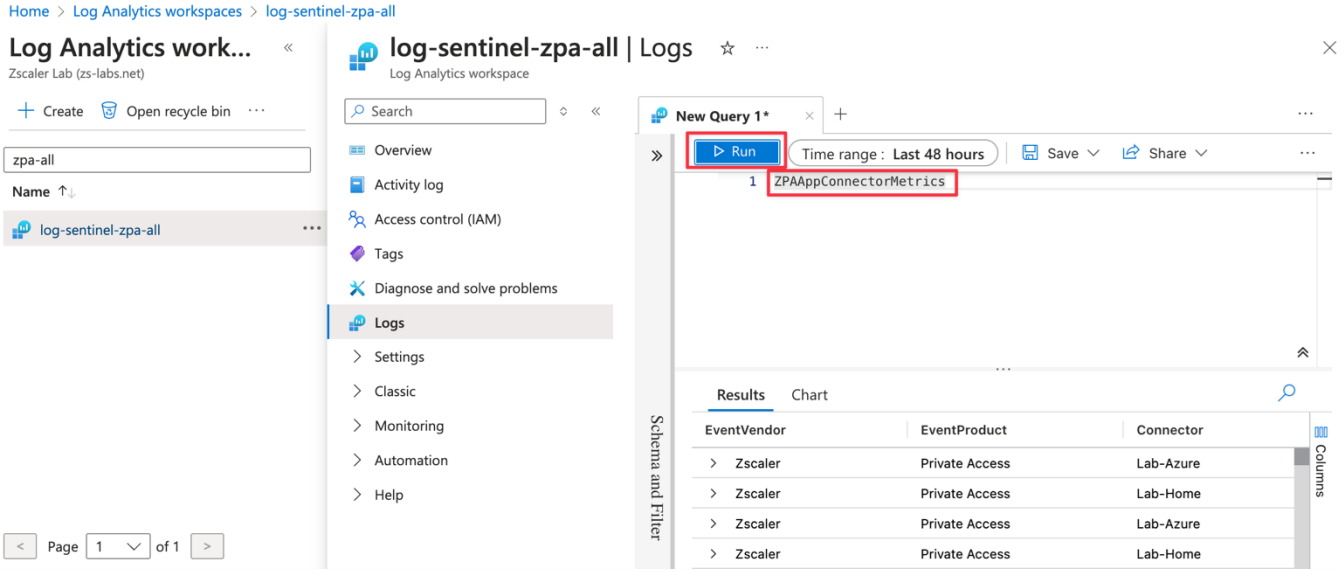


Figure 317. Run queries

Configure the ZPAAppConnectorMetrics Sentinel Workbook

The ZPAAppConnectorMetrics workbook is used to graphically display information related to the ZPA App Connector Metrics Log Feed.

To use the workbook:

1. Download the function code from the Zscaler GitHub repo.
2. Go to your **Microsoft Sentinel Workspace** where you have installed your ZPAAppConnectorMetrics parser.

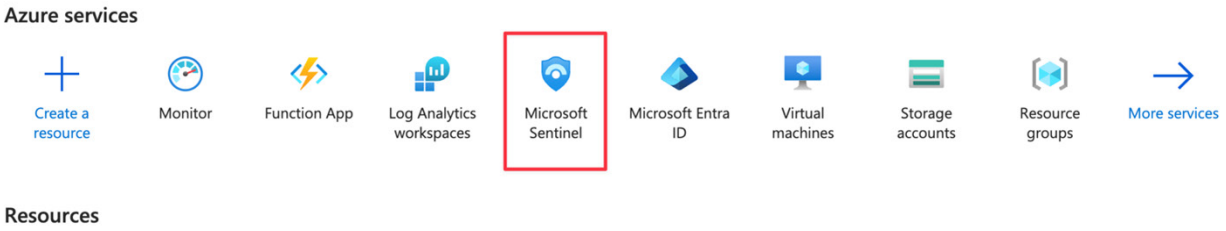


Figure 318. Microsoft Sentinel

©2026 Zscaler, Inc. All rights reserved.

3. Select your workspace with the **ZPA App Connector Metrics logs**.

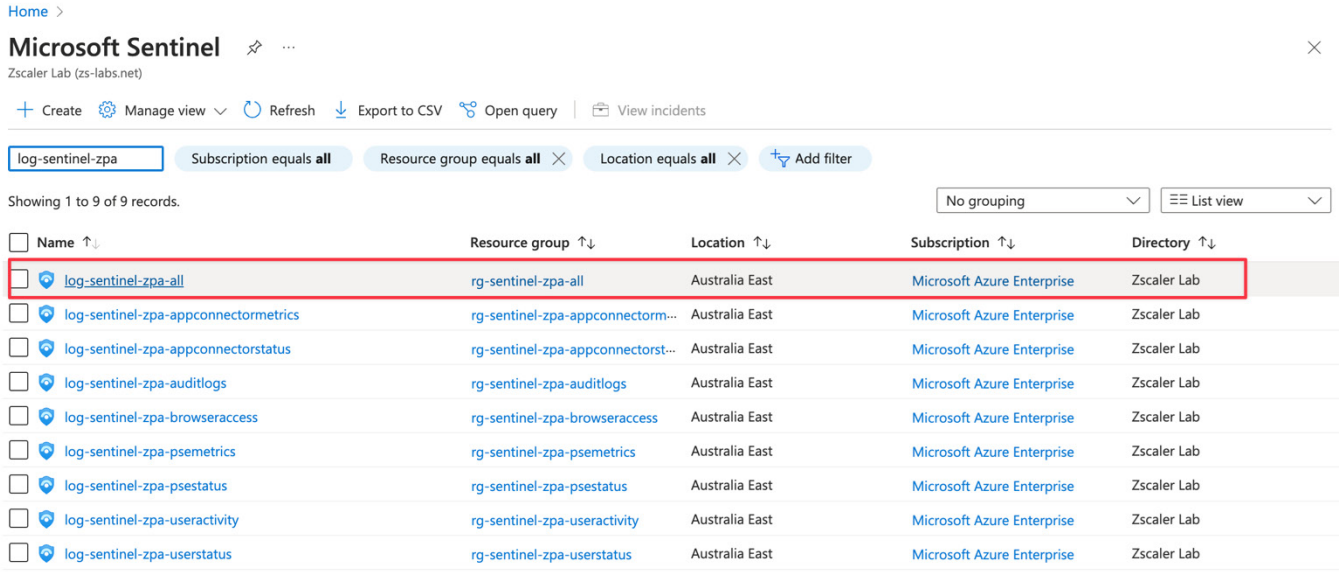


Figure 319. ZPA workspace with logs

4. Click **Workbooks** under **Threat management**.

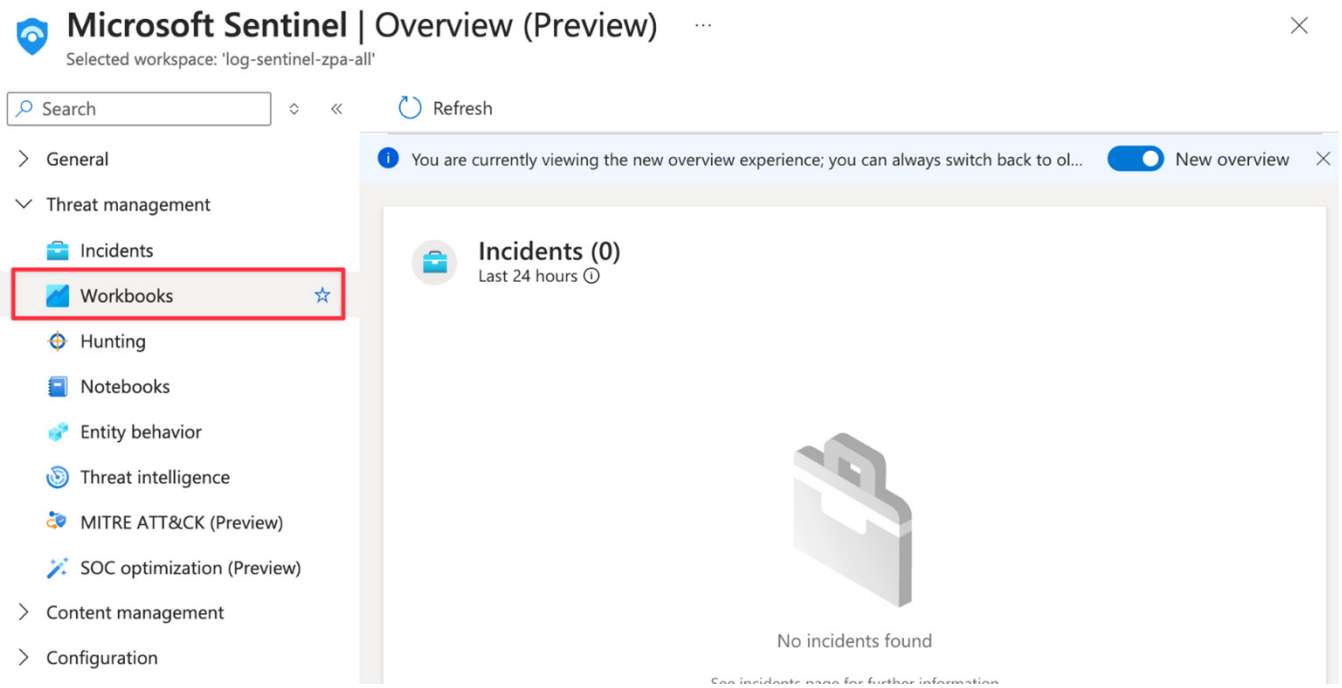


Figure 320. Workbooks

©2026 Zscaler, Inc. All rights reserved.

5. Click **Add Workbook**.

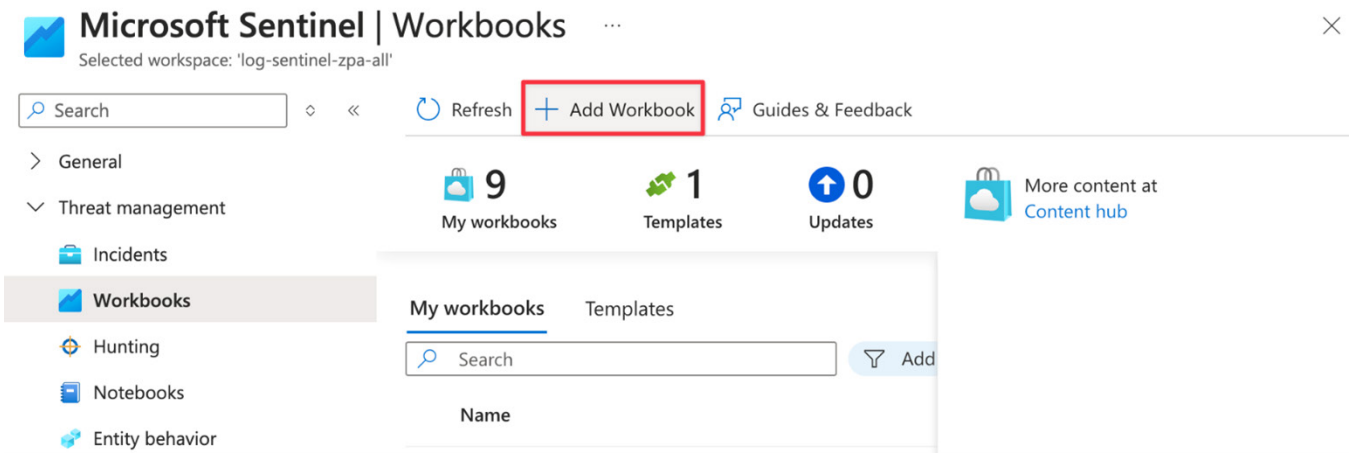


Figure 321. Add Workbook

6. Click **Edit**.



Figure 322. Edit

7. Click the **Advanced Editor** icon.

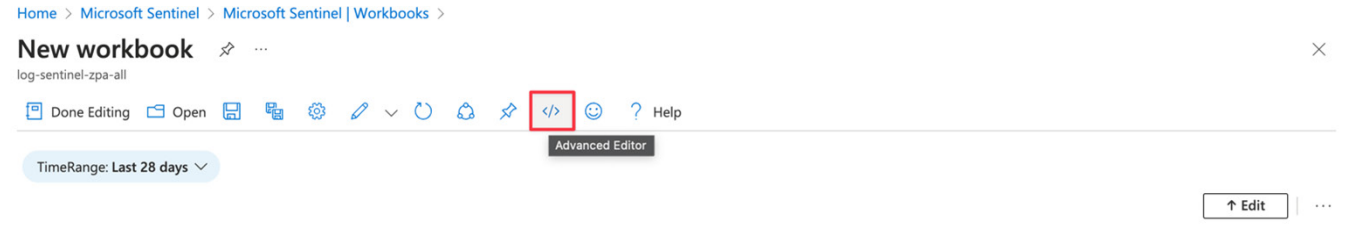


Figure 323. Advanced Editor

8. Replace the existing code in the text field with the code from the GitHub page for the workbook and click **Apply**. The workbook loads the **ZPAAppConnectorMetrics** function to display various graphs.

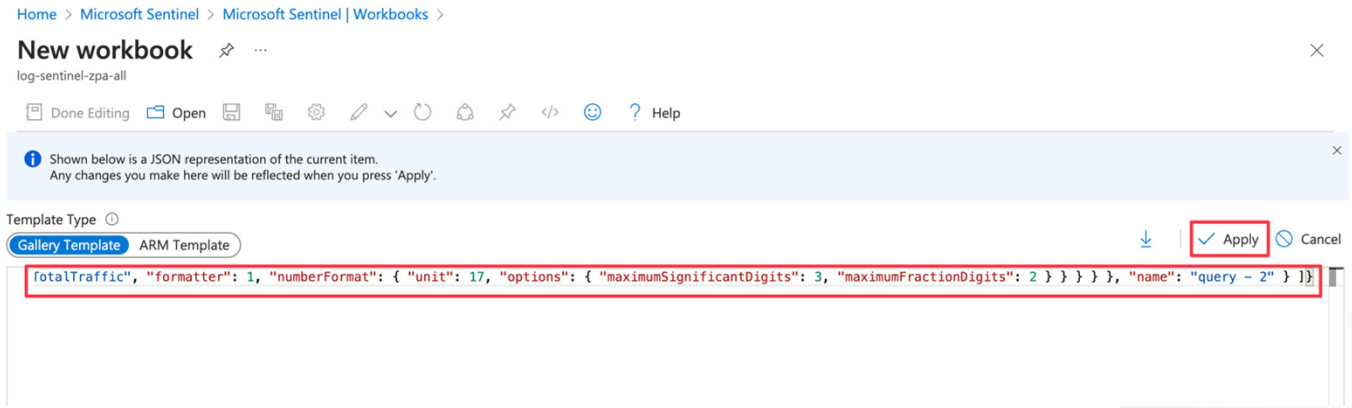


Figure 324. New code text

©2026 Zscaler, Inc. All rights reserved.

9. Click **Done Editing**.

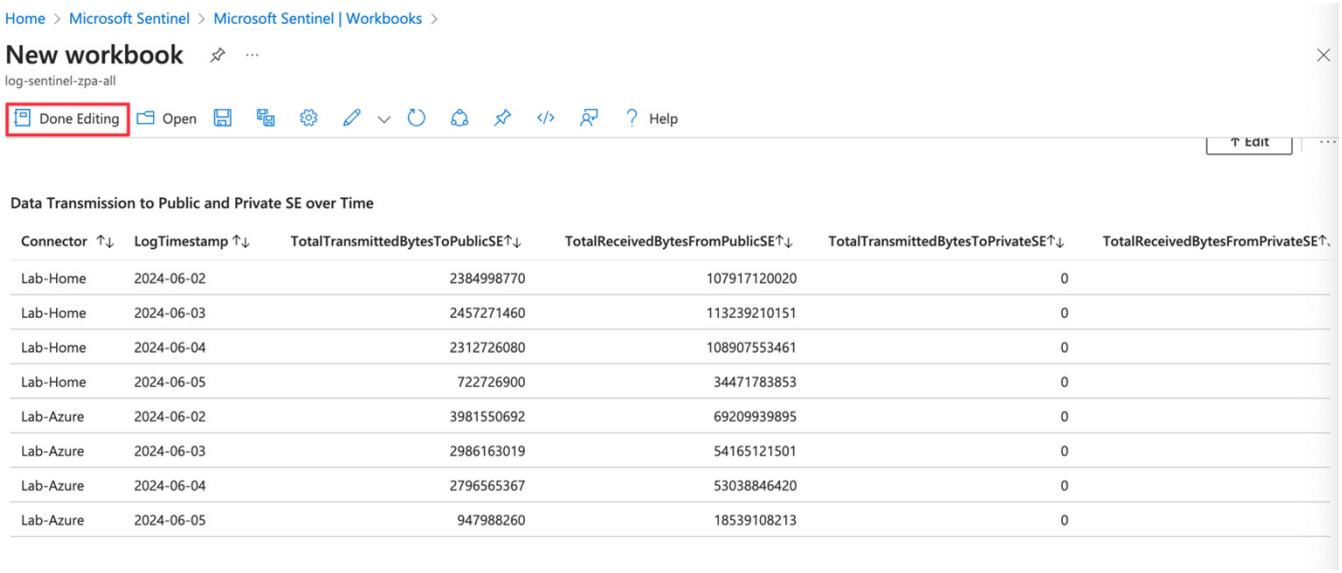


Figure 325. New workbook

10. Click **Save**.



Figure 326. Save workbook

11. Enter the workbook **Title**, select a **Resource group** and **Location**, then click **Apply**.

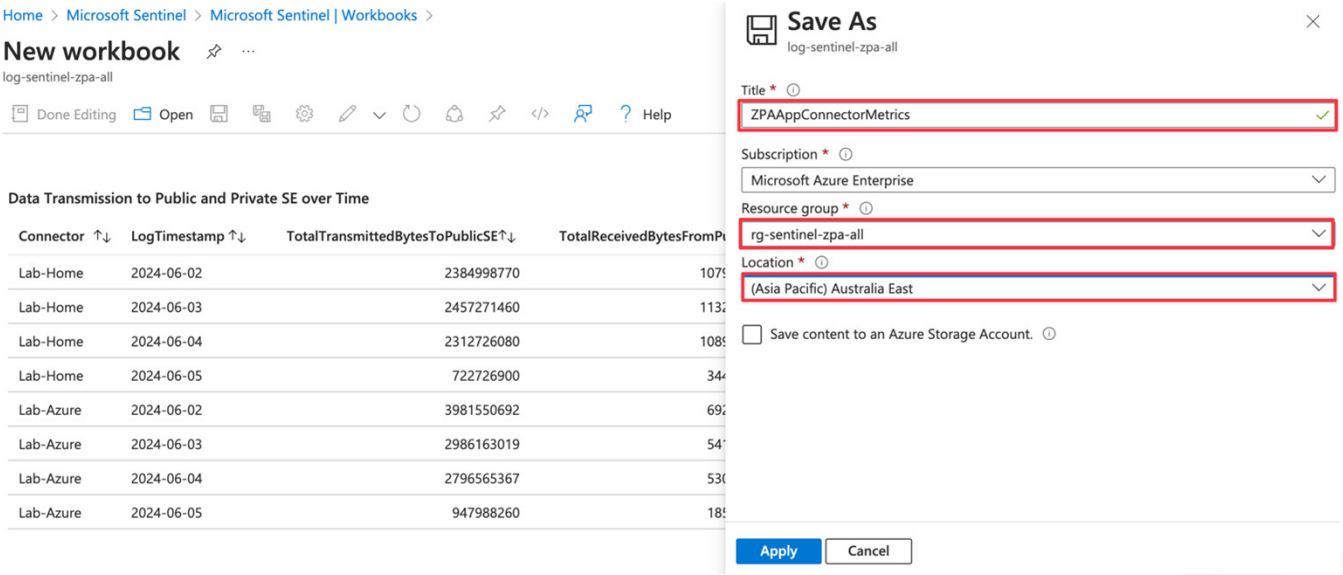


Figure 327. Save the workbook

©2026 Zscaler, Inc. All rights reserved.

Microsoft ASIM Support

Microsoft Sentinel uses a wide variety of data types and sources, requiring deep understanding of each for effective use in analytics rules, workbooks, and hunting queries. Often, even similar types of data, like firewall devices, require a distinct set of rules and queries. This diversity can make correlating data during investigations and hunting a complex task.

To address these challenges, Microsoft has introduced the Advanced Security Information Model (ASIM). Positioned between the diverse data sources and the user, ASIM acts as an intermediary layer that simplifies data handling.

ASIM leverages parsers that are built as KQL user-defined functions that transform data in existing tables, such as CommonSecurityLog, custom logs tables, or Syslog, into a normalized schema.

Users can use ASIM parsers instead of table names in their queries to view data in a normalized format, and to include all data relevant to the schema in their query.

Key Benefits

- ASIM functions use existing Zscaler NSS Web, Firewall, and DNS log feeds, which are sent to the CommonSecurityLog table in CEF format. This removes the need for any further integration and allows administrators to realize ASIM benefits immediately.
- Enhanced Threat Detection and Response: ASIM's standardized data model significantly improves the capability of Microsoft Sentinel to detect and respond to threats. It allows for more accurate and rapid identification of malicious activities across different data sources, leading to quicker response times and reduced potential damage.
- Streamlined Data Management: The normalization of data simplifies the management and analysis of security logs. Security teams can focus more on strategic threat analysis rather than dealing with the complexities of diverse log formats, thus improving operational efficiency.
- Support for Advanced Analytics and AI: With data normalized into a consistent format, ASIM enhances the effectiveness of advanced analytics and AI-driven threat detection methodologies in Microsoft Sentinel. It enables more sophisticated analytics and machine learning models that can learn from a standardized dataset, leading to improved threat detection capabilities.

Resources

The following contains links on Microsoft ASIM support resources.

- For articles with use cases on how to get started using Microsoft Sentinel, refer to the Microsoft Sentinel Documentation.
- For information on ASIM, refer to Normalization and the Advanced Security Information Model (ASIM).

©2026 Zscaler, Inc. All rights reserved.

Using Zscaler ASIM Parser Prerequisites

Using Zscaler ASIM parsers requires the following prerequisites:

ZIA:

- An active instance of ZIA 6.2 or later.
- Administrator login credentials to ZIA.
- Nanolog streaming from ZIA into Microsoft Sentinel for DNS, Web, and Firewall Logs.

Microsoft Sentinel:

- An active instance of Microsoft Sentinel.

Zscaler ASIM Parsers

Zscaler and Microsoft collaborated to develop three source-specific parsers for Microsoft Sentinel's ASIM, which seamlessly integrate into Microsoft's unifying parser framework for enhanced query utilization. The parsers built into every Log Analytics workspace are:

1. _Im_Dns_ZscalerZIAVxx Parser: This parser focuses on DNS logs and is sourced from the Zscaler DNS log feed (NSSDNSlog) in the CommonSecurityLog table. It standardizes log entries related to DNS queries and responses, enabling Microsoft Sentinel to efficiently track and analyze domain lookup activities and detect potential DNS-based threats or anomalies.
2. _Im_NetworkSessionZscalerZIAVxx Parser: This is dedicated to network session logs and is sourced from the Zscaler Firewall log feed (NSSFWlog) in the CommonSecurityLog table. It can, for example, normalize logs related to network connections and sessions, helping Microsoft Sentinel to monitor and analyze network traffic patterns, identify unauthorized access attempts, or detect unusual data flows.
3. _Im_WebSessionZscalerZIAVxx Parser: Tailored for web session logs, this parser standardizes entries detailing user web activities and is sourced from the Zscaler Web log feed (NSSWeblog) in the CommonSecurityLog table. It aids in the analysis of web browsing patterns, the identification of potentially malicious URLs accessed, or the tracking of user behavior for security monitoring.



The xx in the parser name refers to the current version of the parser in development. For example, the current version of the network session parser (_Im_NetworkSessionZscalerZIAVxx) is _ASim_NetworkSession_ZscalerZIAV04.

©2026 Zscaler, Inc. All rights reserved.

Using Zscaler ASIM Parsers

ASIM source-specific parsers and unifying parsers serve different roles in data normalization within Microsoft Sentinel.

- Source-Specific Parsers: These are designed to handle data from specific sources, i.e., from ZIA in this case. They focus on transforming the unique log formats and data structures of a particular source into a standardized format that can be understood within the Sentinel ecosystem.
- Unifying Parsers: These are broader and more general. They take the output from various source-specific parsers and integrate it into a cohesive, unified schema. This allows for a more consistent and streamlined approach to analyzing data from multiple sources within Microsoft Sentinel.

To effectively use Zscaler ASIM parsers in Microsoft Sentinel for threat hunting or query purposes, an administrator leverages Microsoft Sentinel's unifying parsers. These unifying parsers are designed to handle and consolidate the Zscaler source-specific parsers together with any other available source-specific parsers into a standardized data output format.

- _Im_Dns - for the DNS schema.
- _Im_NetworkSession for the Network Session schema.
- _Im_WebSession for the Web Session schema.

The following image shows a query using _Im_Dns.

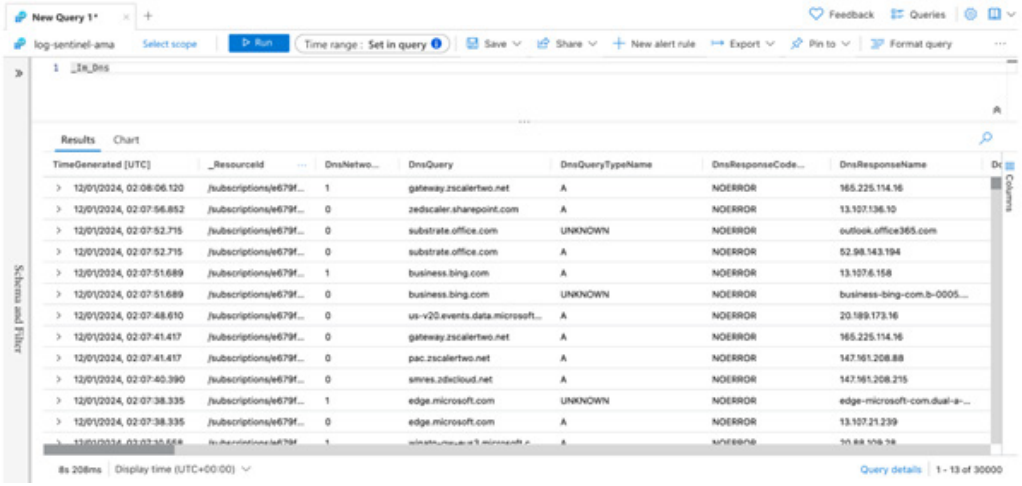


Figure 328. Running a query using _Im_Dns

The following image shows a query using _Im_NetworkSession.

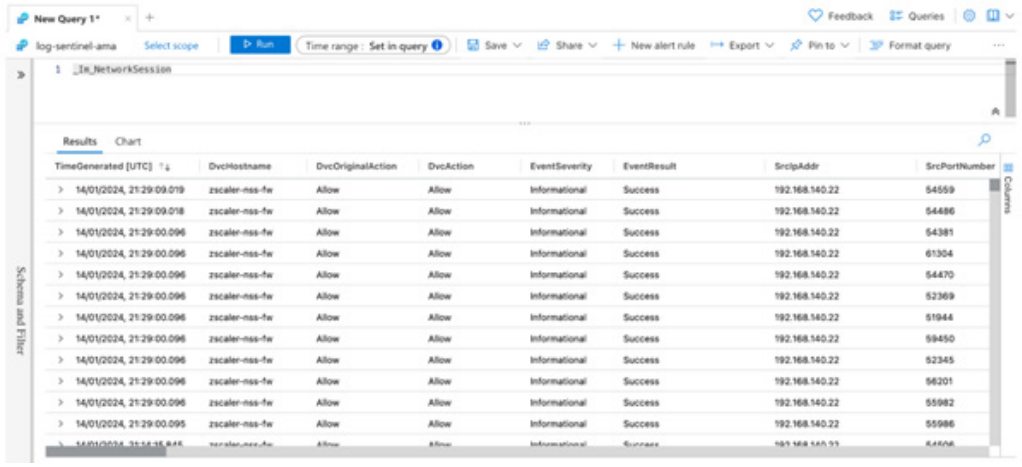


Figure 329. Running a query using _Im_NetworkSession

©2026 Zscaler, Inc. All rights reserved.

The following image shows a query using _Im_WebSession.

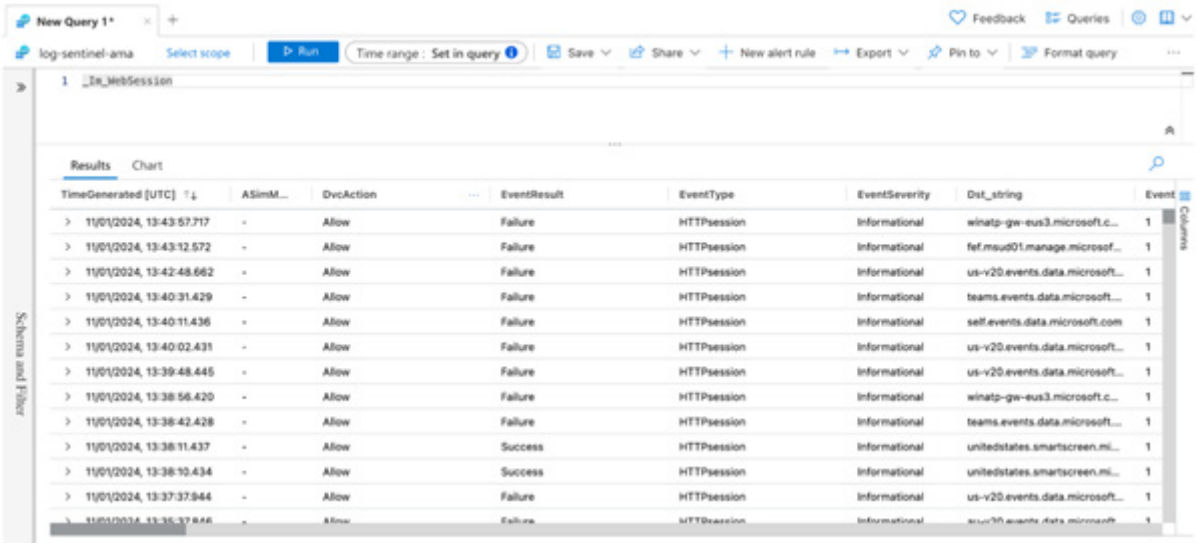


Figure 330. Running a query using _Im_WebSession

©2026 Zscaler, Inc. All rights reserved.

Microsoft Sentinel Playbooks Overview

SOC analysts are regularly inundated with security alerts and incidents. This results in situations where many alerts are ignored and many incidents aren't investigated, leaving the organization vulnerable to attacks that go unnoticed.

Many of these alerts and incidents conform to recurring patterns that you can address by specific and defined sets of remediation actions. Analysts are also tasked with basic remediation and investigation of the incidents they do manage to address. You can automate many of these activities. Automating tasks allows SOCs more time for productivity, efficiency, and investigative activity.

This playbook is a collection of remediation actions that you run from Microsoft Sentinel to help automate and orchestrate your threat response. It can be run in two ways:

- Manually on-demand, on a particular entity or alert.
- Automatically in response to specific alerts or incidents, when triggered by an automation rule.

For example, if an account and machine are compromised, a playbook can isolate the machine from the network and block the account by the time the SOC team is notified of the incident.

Microsoft Sentinel Playbooks Resources

The following table contains links to Microsoft Sentinel Playbook support resources.

- For articles with use cases to get started using Microsoft Sentinel, refer to the Microsoft Sentinel documentation.
- For Microsoft Sentinel Playbooks, refer to Security, Orchestration, Automation, and Response (SOAR).

Document Prerequisites

To use this document, the following prerequisites are required:

ZIA:

- An active instance of ZIA 6.2 or later.
- Administrator login credentials to ZIA.
- Nanolog streaming from ZIA into Microsoft Sentinel.
- OAuth 2.0 Authentication have been met.

Microsoft Sentinel:

- An Azure Active Directory license and tenant, or an individual account with a valid payment method, are required to access Azure and deploy resources.
- After you have a subscription, you'll need the relevant permissions to begin using your subscription.
- A Log Analytics workspace is required to house all of the data that Microsoft Sentinel investigates and uses for its detections, analytics, and other features.
- Zscaler recommends that when you set up your Microsoft Sentinel workspace, create a resource group that's dedicated to Microsoft Sentinel and the resources that Microsoft Sentinel uses, including the Log Analytics workspace, any playbooks, workbooks, and so on.

©2026 Zscaler, Inc. All rights reserved.

ZIA API Architecture Overview

Zscaler has two core products: ZIA and ZPA. This section only addresses ZIA.

Zscaler provides secured access to Cloud Service API and Sandbox Submission API using different authentication schemes:

API	Supported Authentication Methods
Cloud Service API	OAuth 2.0 (recommended) Combination of Basic Authentication and API Key
Sandbox Submission API	Combination of Basic Authentication and API Token

The instructions in this section and the Microsoft Sentinel Playbooks have been developed to leverage the OAuth 2.0 model for authentication to the Cloud Service API.

OAuth 2.0 Authentication

OAuth 2.0 allows third-party applications to obtain controlled access to protected resources using access tokens. Zscaler uses the Client Credentials grant type, in which the clients exchange their credentials for an access token.

In this model, client applications make API calls to the cloud service API using an access token obtained from the authorization server in exchange for their credentials. Therefore, the clients access the cloud service API resources on their behalf without requiring any user interaction.

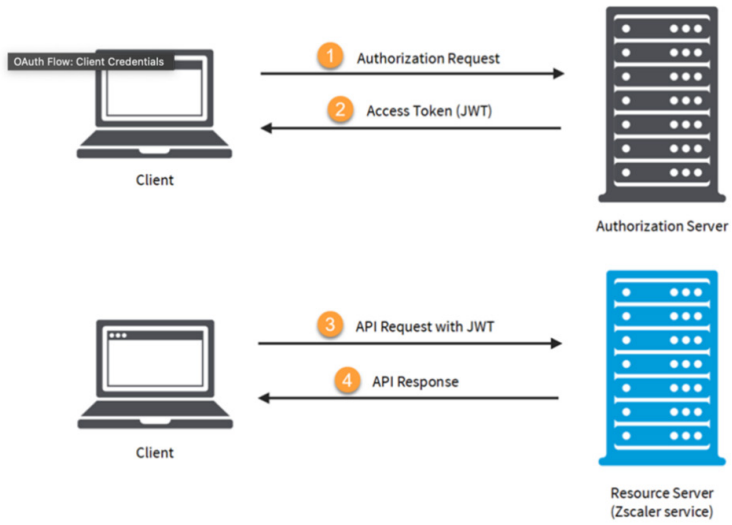


Figure 331. Obtaining an access token

The following steps describe the process:

1. A client requests an access token from the authorization server. A client application registered with the authorization server sends an authorization request with its credentials (i.e., client ID and client secret) to the authorization server. In addition to the client credentials, the authorization request must specify the required scope and the grant type.
2. The authorization server authenticates the client and provides an access token. The authorization server validates the client's credentials and provides the client with a signed JSON Web Token (JWT) access token upon successful authorization. The response from the authorization server contains the access token, token type (bearer token), and the token expiry time.
3. The client sends the access token to the resource server. The client sends an API request to the resource server (i.e., cloud service API) with the signed JWT access token in the request authorization header.

©2026 Zscaler, Inc. All rights reserved.

4. The resource server grants access to protected resources. The following series of events take place before the resource server (i.e., cloud service API) can accept the API request:
    a. The Zscaler service extracts the JWT access token from the API request header and decodes the token to fetch information such as the key ID, algorithm, scope, client ID, audience, expiry, and other configured values.
    b. The Zscaler service cryptographically verifies the signature of the JWT token using the authorization server's public key.
    c. If the JWT signature verification is successful, the Zscaler service validates the JWT's scope claim, which is in <Zscaler Cloud Name>::<Org ID>::<API Role> format. The <API Role> value in the scope is used to authorize the API request. This value must match with one of the API Roles configured in the ZIA Admin Portal. If no match is found, the API request is rejected.
    d. Finally, the Zscaler service grants the client application access to the requested API resources.

OAuth 2.0 Prerequisites

Your organization must meet the following prerequisites before you can access the API:

- You must have an API subscription. If you do not have a subscription, submit a Zscaler Support ticket.
- You must have the API Roles (government agencies, see API Roles) configured in the ZIA Admin Portal.
- You must have your client applications registered on your authorization server (i.e., PingFederate, Okta, or Entra ID) with the required scope and configured appropriately. This document walks you through the required setup using Entra ID (formerly known as Azure AD).
- You must have your OAuth 2.0 authorization server (government agencies, see OAuth 2.0 authorization server) added to the ZIA Admin Portal.

Benefits of OAuth 2.0 Authentication

- Better Security: OAuth 2.0 secures your APIs with dynamic credentials, which are time-bound and generated on demand for a client.
- Limits Exposure of Credentials: Unlike the authentication model that uses API keys and ZIA admin credentials and might involve user management outside the organization's identity provider, OAuth 2.0 does not require ZIA admin credentials for authentication.
- Granular Access Control: The Client Credentials OAuth flow employs API Roles (government agencies, see API Roles) to define permissions required to access specific categories of cloud service API. Unlike admin roles, API roles are not assigned to ZIA admin users. Instead, API roles are associated with the client applications that are accessing the API. OAuth 2.0 provides added security to API access by isolating API permissions from admin users with access to the ZIA Admin Portal.
- Reduced Maintenance: OAuth 2.0 does not require obfuscation of credentials, unlike API keys which must be obfuscated (government agencies, see obfuscated) on the client with additional programming for enhanced security.

©2026 Zscaler, Inc. All rights reserved.

Configuring a ZIA API Role

The following sections demonstrate how to configure a ZIA API role.

Logging In to ZIA

First, set up the Zscaler side of this service. Log in to the ZIA Admin Portal. If you are unable to log in using your Administrator Account, contact Zscaler Support (government agencies, contact Zscaler Support).

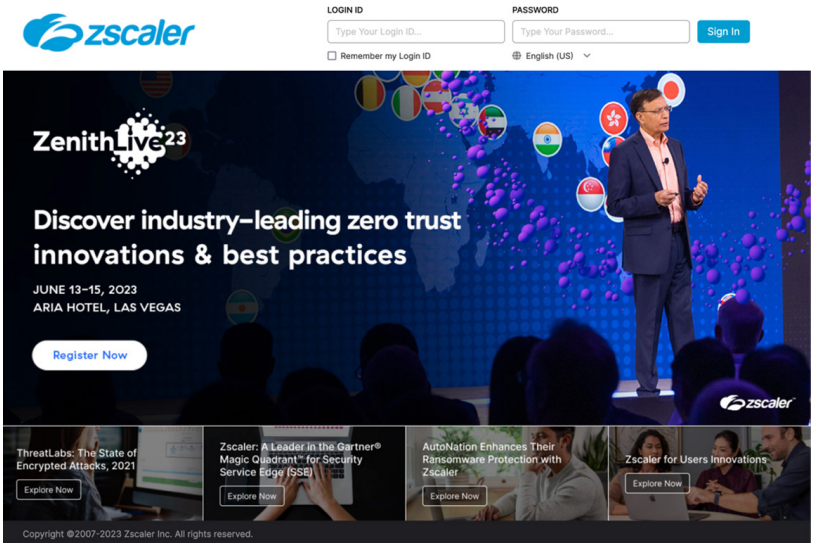


Figure 332. ZIA Admin Portal

Create an API Role

To create an API role:

1. Go to **Administration** > **Role Management**.

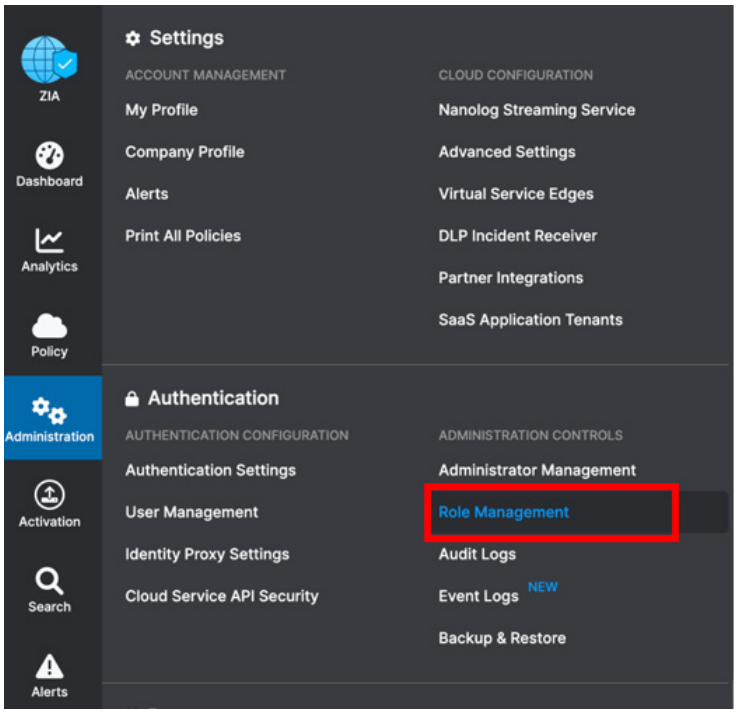


Figure 333. Role Management

©2026 Zscaler, Inc. All rights reserved.

2. Click **Add API Role**.

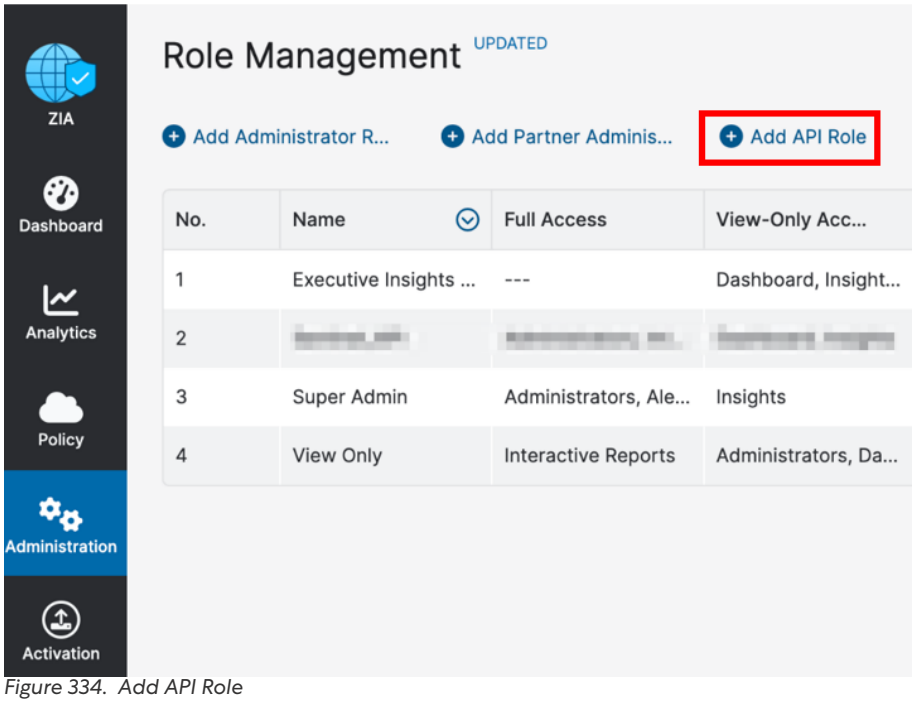


Figure 334. Add API Role

3. Enter an API role named `Sentinel-Playbooks` and enable the configuration shown in the following figure.

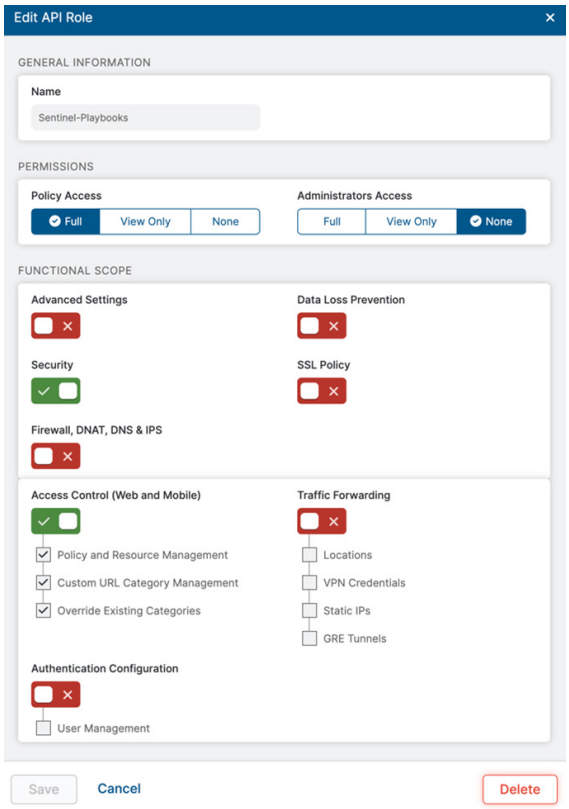


Figure 335. Configure API Role

4. Click **Save**.
5. **Activate** the changes.

©2026 Zscaler, Inc. All rights reserved.

Configuring Microsoft Entra ID for OAuth 2.0 Authentication

This guide uses Microsoft Entra ID as the IdP for OAuth 2.0. For other IdP's, including Okta and PingFederate, see Adding Identity Providers (government agencies, see Adding Identity Providers).

For each API role configured in the ZIA Admin Portal, you must complete the steps in the following sections to set up OAuth 2.0 authorization using Entra ID.

You must create two app registrations in the Microsoft Entra ID:

1. The ZIA API Client Application.
2. The ZIA API Web Service Application.

Register the ZIA API Client Application

To register the ZIA API Client Application:

1. Sign in to the Microsoft Azure portal.
2. Go to **Microsoft Entra ID** under **Azure services**.
3. Click **App registrations** on the left-side navigation.
4. Click **New registration**.

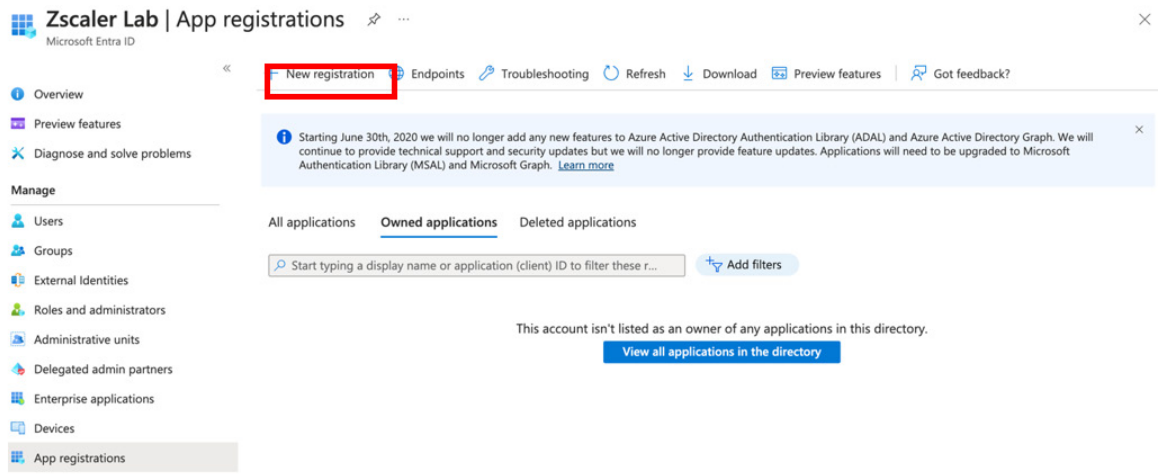


Figure 336. App registrations

©2026 Zscaler, Inc. All rights reserved.

5. In the **Register an application** window:
   a. Enter a **Name** for the application that is representative of the ZIA API role used (e.g., `ZIA API Microsoft Sentinel Playbooks`).
   b. Set the **Supported account types** option to **Accounts in this organizational directory only (Zscaler Lab only - Single tenant)**.

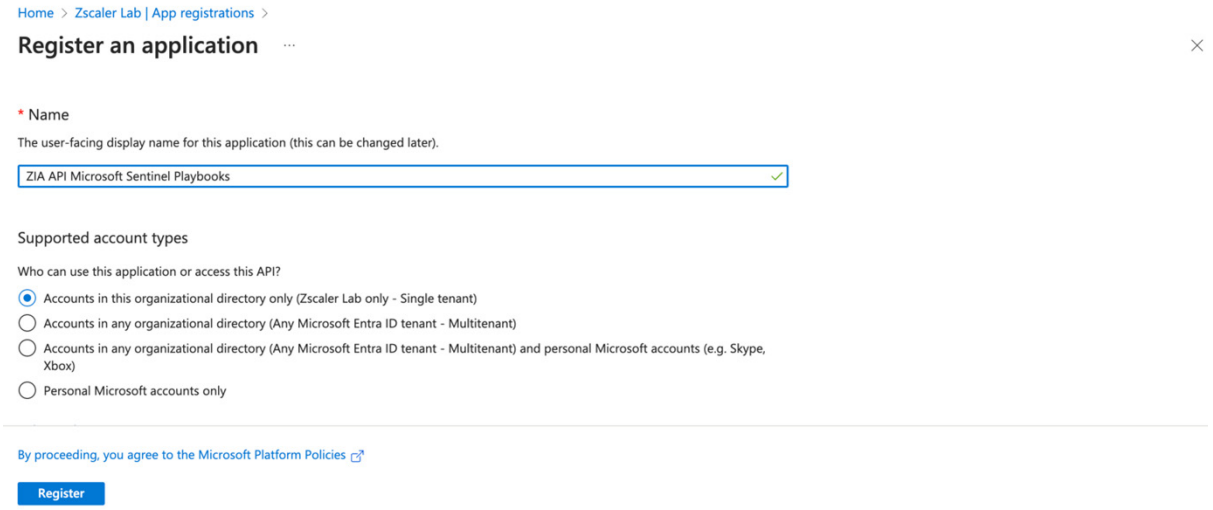


Figure 337. Register an application

6. Click **Register**. The application is registered, and the application's **Overview** page is displayed.
7. Copy the **Application (client) ID** value from the **Overview** page and save it for later use.

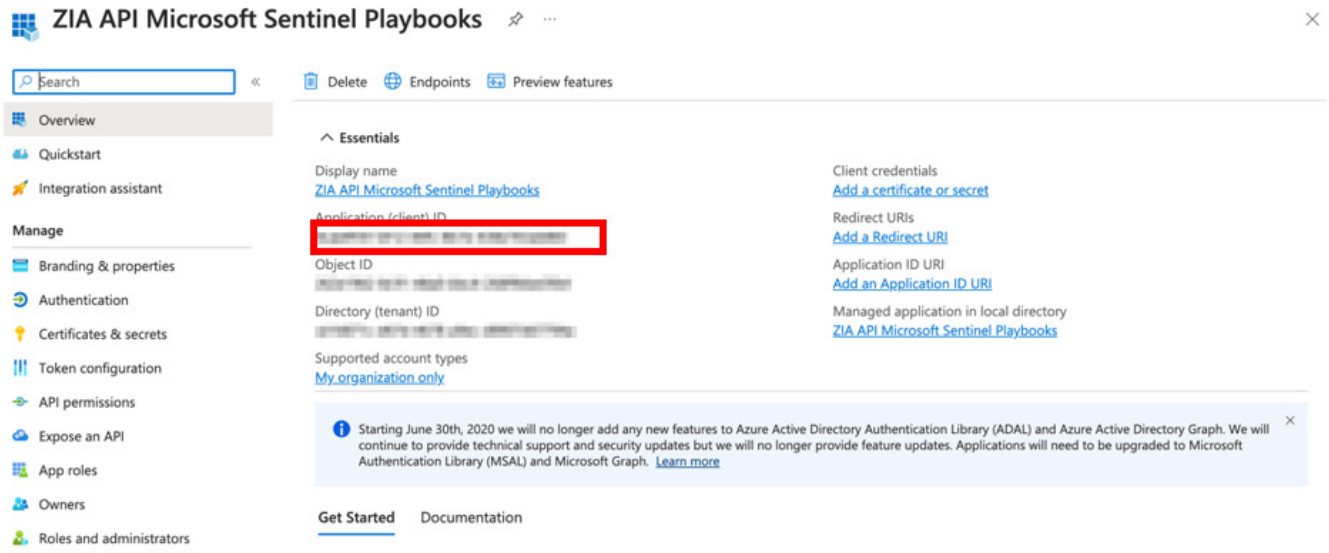


Figure 338. Application Overview page

©2026 Zscaler, Inc. All rights reserved.

8. Go to **Owners**.
9. Click **Add owners** and add yourself as an owner.

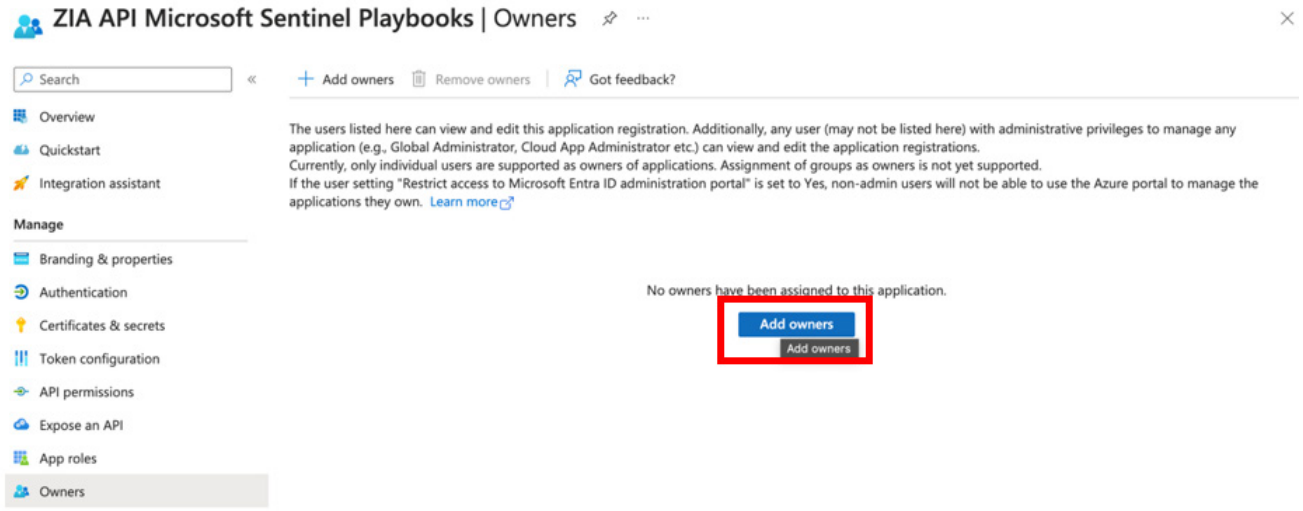


Figure 339. Owners

Configure Client Credentials

To configure the credentials for your ZIA API Client Application:

1. Click **Certificates & secrets** in the left-side navigation.
2. Click **New client secret** on the **Client secrets** tab

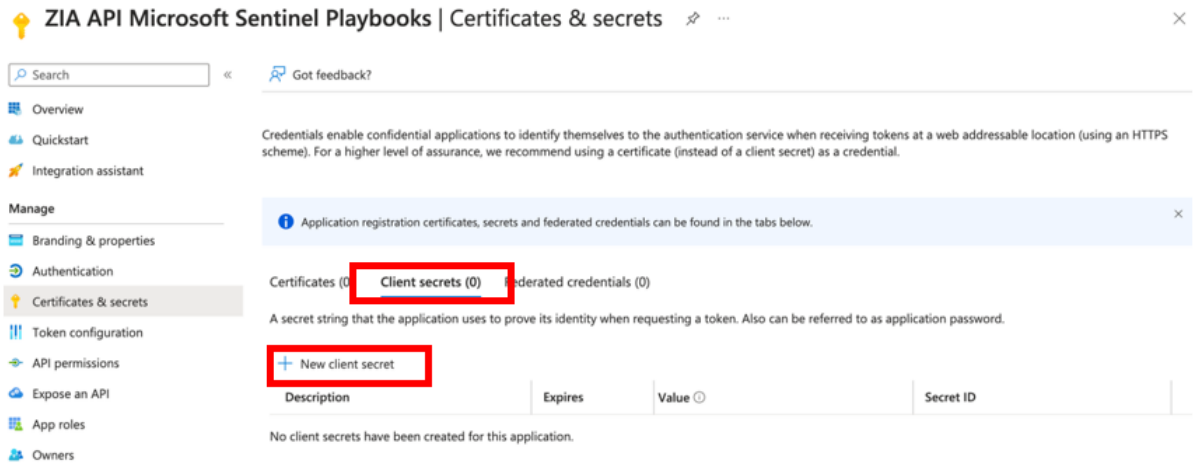


Figure 340. Certificates & secrets

©2026 Zscaler, Inc. All rights reserved.

3. In the **Add a client secret** pane:
   a. **Description**: Enter information about the client's secret.
   b. **Expires**: Select the appropriate expiration time from the drop-down menu.

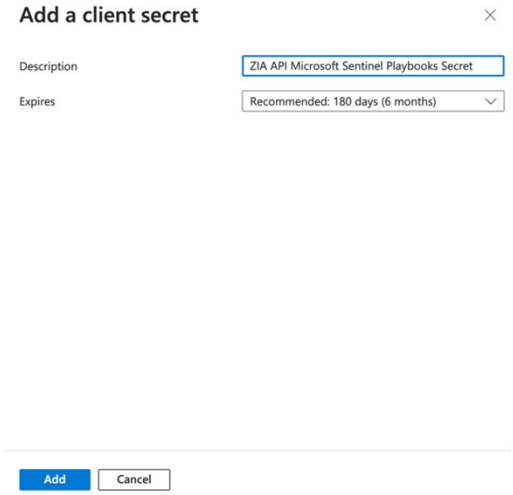


Figure 341. Add client secret

   c. Click **Add**. The client secret value is generated and displayed.

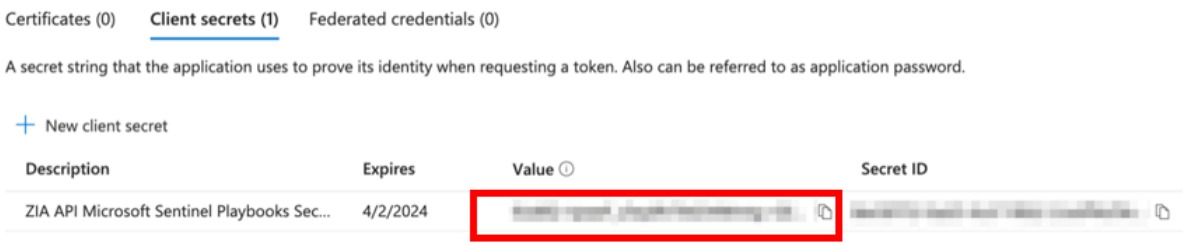


Figure 342. Client secret value

   d. Copy the secret value and save it for later use.

©2026 Zscaler, Inc. All rights reserved.

Configure the ZIA API Web Service Application

This section describes how to register the ZIA API Web Service Application in Microsoft Entra ID and configure the app to expose a Web API that is accessible by the ZIA API client application.

1. To complete the application registration process:
   a. Go to **Microsoft Entra ID** under **Azure services**.
   b. Click **App registrations** on the left-side navigation.

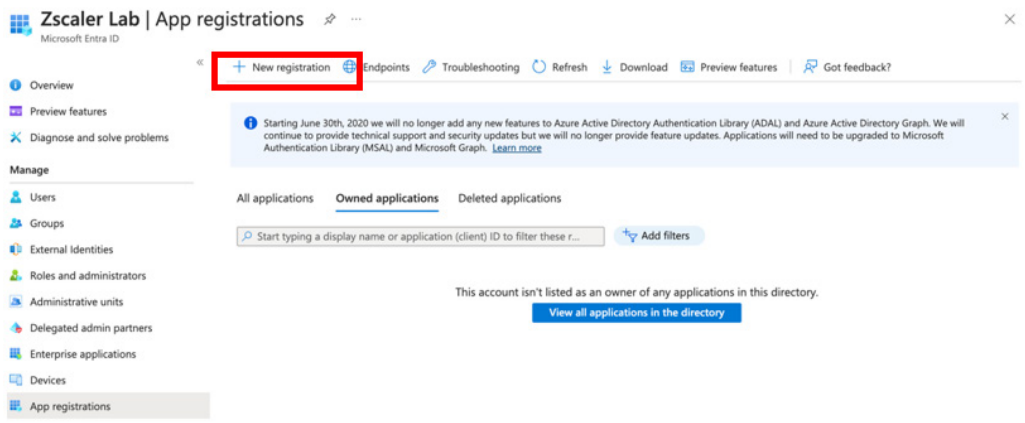


Figure 343. App registrations

2. Click **New registration**. The **Register an application** window opens.
3. In the **Register an application** window, complete the following:
   a. **Name**: Enter a name for the ZIA API Web Service Application (e.g., `ZIA API Web Service`).
   b. **Supported account types**: Ensure that this option is set to the **Accounts in this organizational directory only (Zscaler Lab only - Single tenant)** value.
4. Click **Register**. The application is registered and displayed.

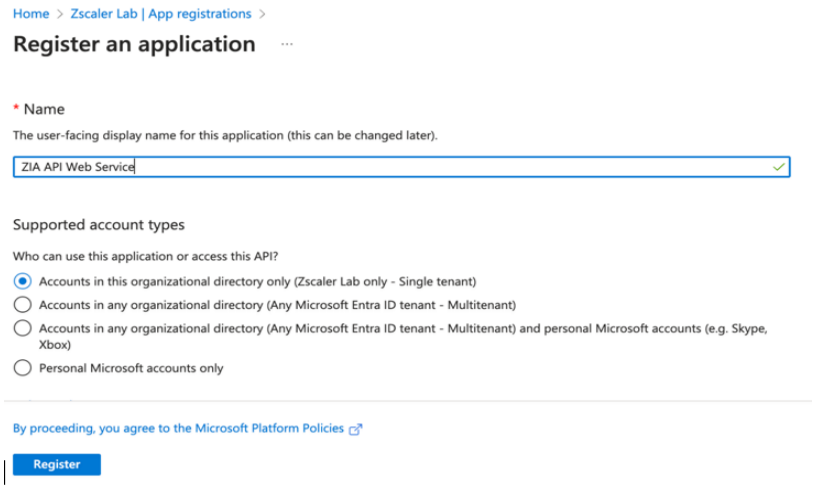
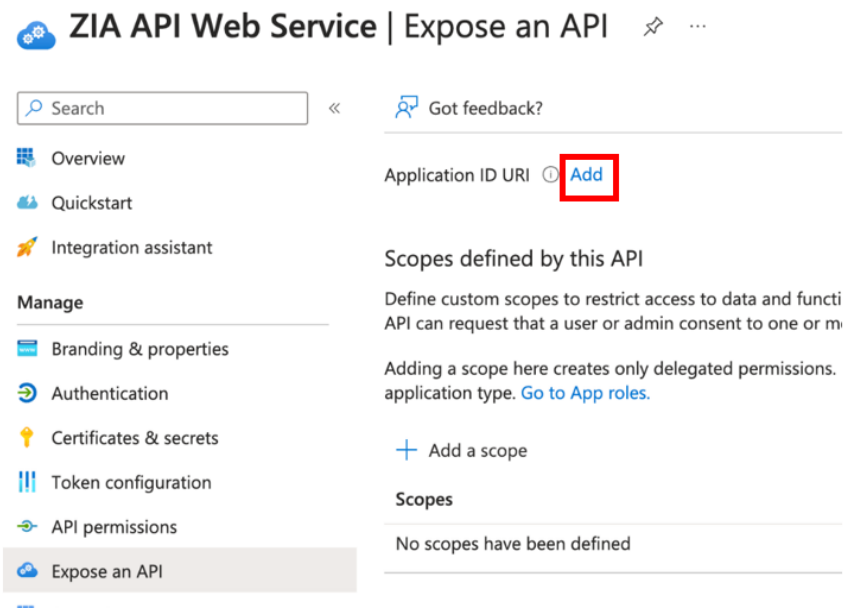


Figure 344. Register an application

©2026 Zscaler, Inc. All rights reserved.

5. To configure the ZIA API Web Service Application to expose a Web API:
   a. Click **Expose an API** on the left-side navigation of the app in your ZIA API web service.
   b. Click **Add** next to the **Application ID URI**, and then click **Save** with the default value that is provided. Copy the



Application ID URI and save it for later use.*Expose an API*

   c. Click **Add a scope** under **Scopes defined by this API**. The **Add a Scope** section appears.

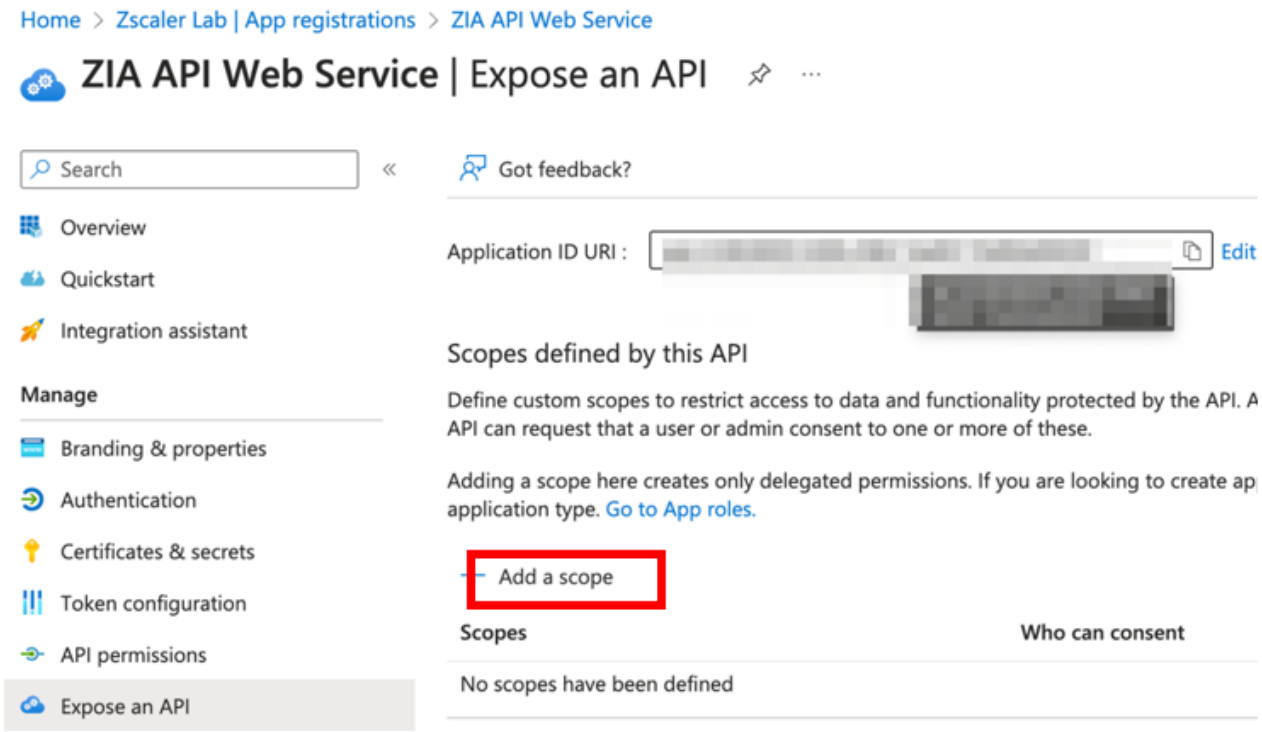


Figure 345. Add a scope

©2026 Zscaler, Inc. All rights reserved.

6. In the **Add a Scope** section:
   a. **Scope name**: Enter a relevant scope name (e.g., `Sentinel-Playbooks`).
   b. **Who can consent?**: Choose which users can consent to this scope based on your business requirements and organization policies.
   c. **Admin consent display name**: Enter a consent name that is displayed on the consent window when admins consent to this scope.
   d. **Admin consent description**: Enter a detailed description that must be displayed when tenant admins expand a scope on the consent window.
   e. **User consent display name**: Enter a consent name that must be displayed on the consent window when users consent to this scope.
   f. **User consent description**: Enter a detailed description that must be displayed when users expand a scope on the consent window.
   g. **State**: Ensure that the scope is **Enabled**.
   h. Click **Add scope**.

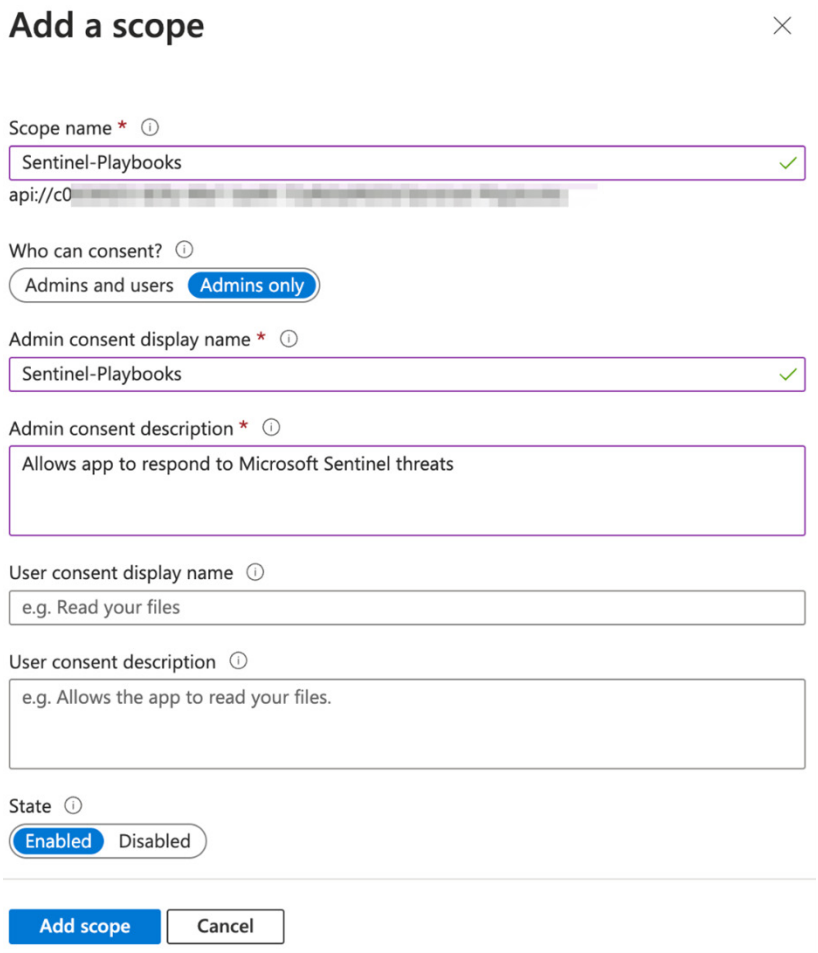


Figure 346. Configure scope

©2026 Zscaler, Inc. All rights reserved.

7. To authorize the client application, click **Add a client application** under **Authorized client applications**.

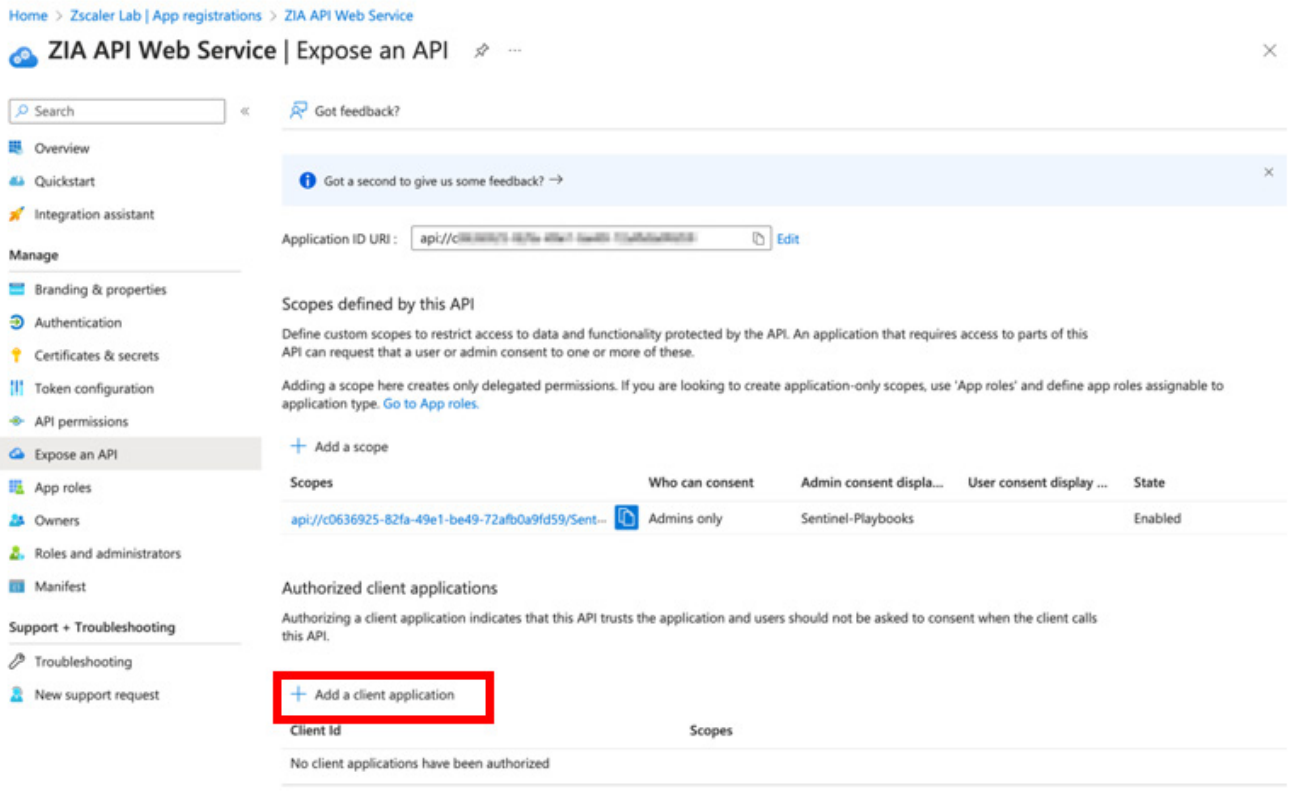


Figure 347. Add a client application

8. In the **Client ID** field, enter the **Application (client) ID** from the ZIA API client application and select the required scope under **Authorized scopes**.
9. Click **Add application**.

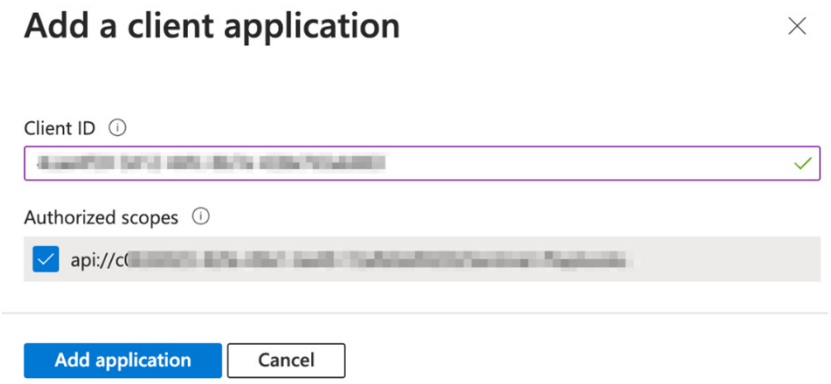


Figure 348. Authorized scopes

©2026 Zscaler, Inc. All rights reserved.

10. Click **App roles** on the left-side navigation and then click **Create app role**.

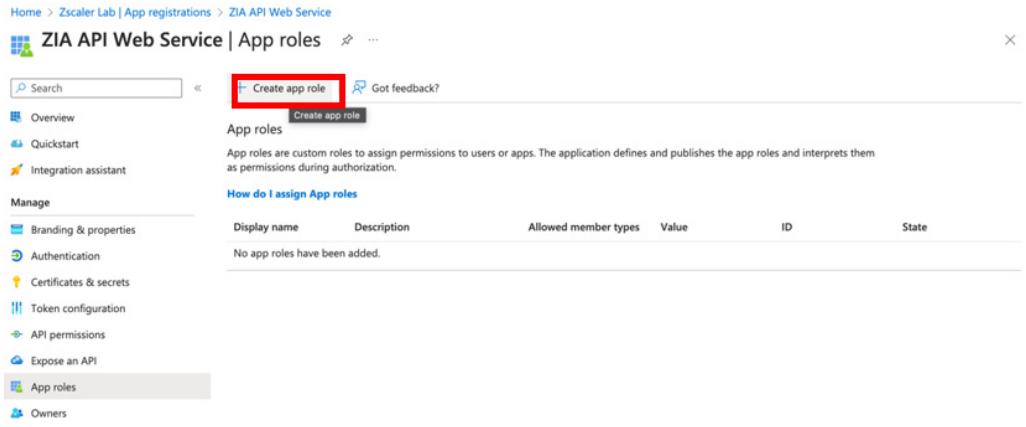


Figure 349. Create app role

11. In the **Create app role** pane, after the app role is created, go to the **Manifest** page from the left-side navigation and ensure that the app role you configured appears in the **appRoles** field within the JSON fields:
    a. **Display name**: Enter a display name for the app role.
    b. **Allowed member types**: Select **Applications**.
    c. **Value**: Enter a value in the format supported by the Zscaler service, `<Zscaler Cloud Name>::<Org ID>::<API Role>`, where:
        - <Zscaler Cloud Name> represents the name of the cloud on which your organization is provisioned.
        - <Org ID> is an identifier assigned to your organization by Zscaler. You can obtain this value from the Company Profile page within the ZIA Admin Portal.
        - <API Role> represents the API Role configured on the Role Management page within the ZIA Admin Portal. For example, `zscaler.net::272378::sampleRole`.
    d. **Description**: Enter a description of the app role.
    e. **Do you want to enable this app role?**: Select this checkbox.
    f. Click **Apply**.

©2026 Zscaler, Inc. All rights reserved.

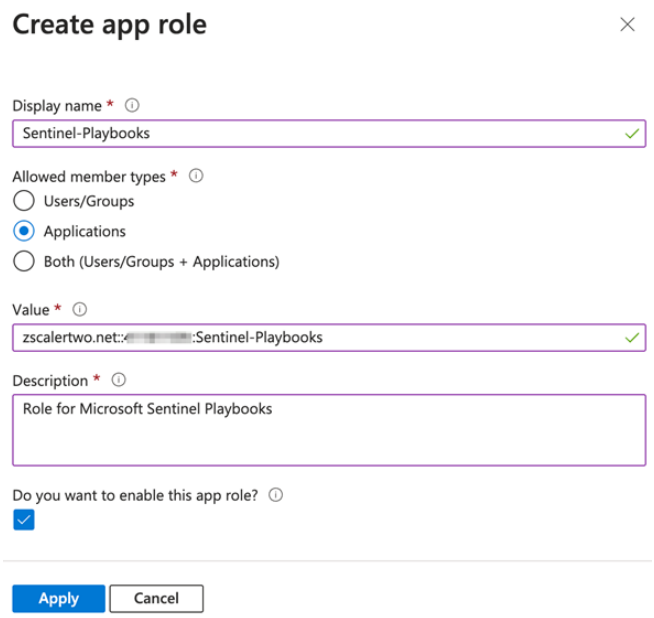


Figure 350. Configure new app role

12. Click **Owners**, select **Add owners**, and add yourself as an owner.

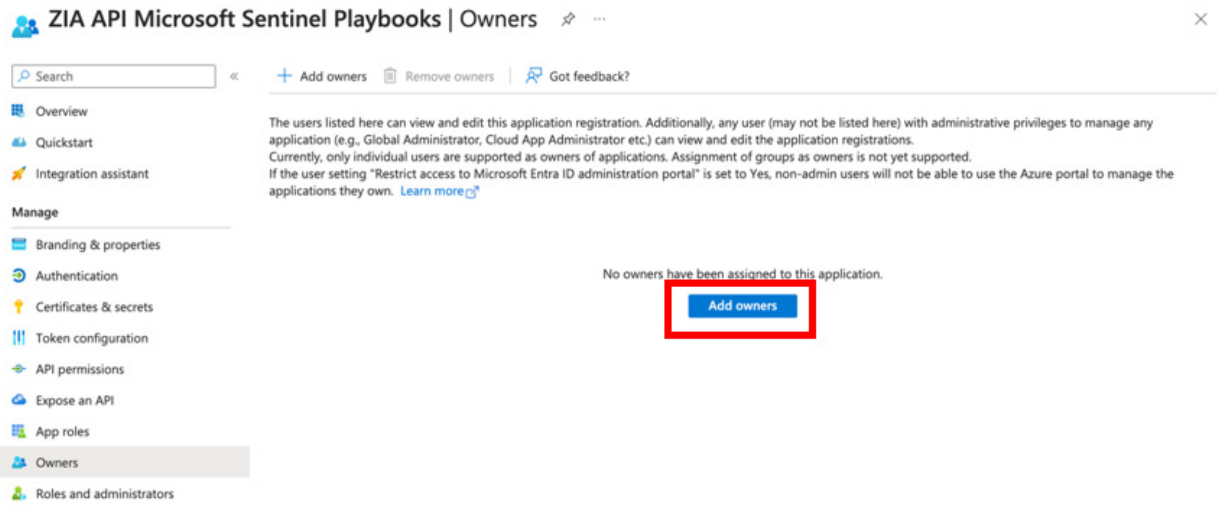


Figure 351. Add owners

Configure the ZIA API Client Application to Access the ZIA API Web Service Application

To configure the ZIA API Client Application permissions to use the ZIA API Web Service Application:

1. Go to **Microsoft Entra ID** under **Azure services**.
2. Click **App Registrations** on the left-side navigation and then click the ZIA API client application. The **Overview** page of the app is displayed.
3. Click **API permissions** on the left-side navigation and then click **Add a permission** under **Configured permissions**. The **Request API permissions** pane opens.

©2026 Zscaler, Inc. All rights reserved.

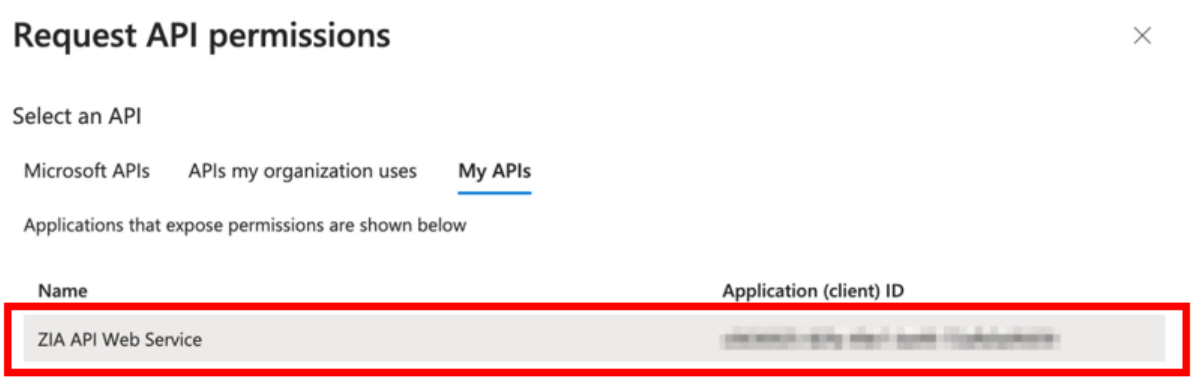


Figure 352. Request API permissions

4. In the **Request API permissions** pane:
   a. Go to the **My APIs** tab and then click the **Web Service API App (Application B)**.
   b. Choose the **Application permissions** as the type of permission required by the client application.
   c. Under **Select permissions**, select the **API Role** that was created in Configure the ZIA API Web Service Application).

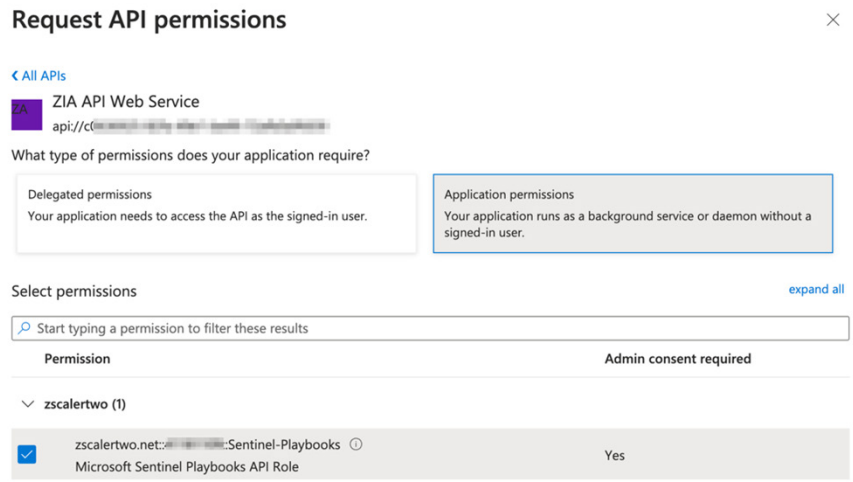


Figure 353. Configure API permissions

©2026 Zscaler, Inc. All rights reserved.

d. Click **Application permissions**.

e. If you are testing the configurations, enable the **Grant admin consent for Zscaler Lab** option under the **Configured permissions** section and click **Yes** when prompted.

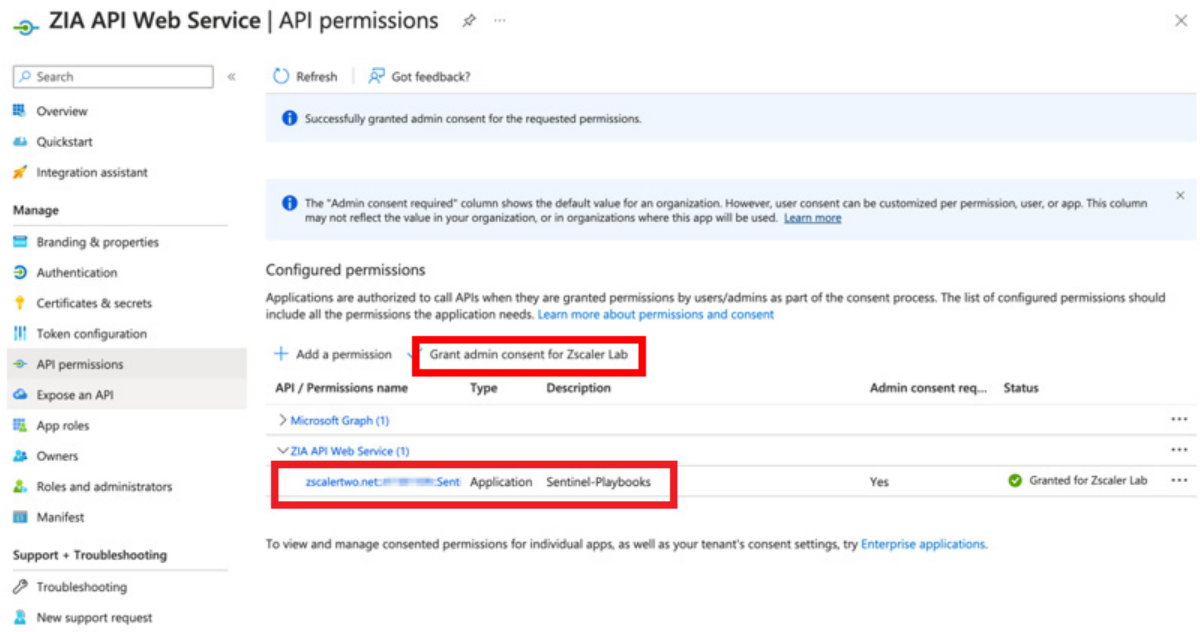


Figure 354. Configured permissions

Collect Information for Zscaler OAuth 2.0 Server Configuration

To collect information from Microsoft Entra ID used to configure the OAuth 2.0 authorization server in the ZIA Admin Portal:

1. Go to **Microsoft Entra ID** under **Azure services**.
2. Click **App Registrations** on the left-side navigation and then click the ZIA API Web Service configured in Configure the ZIA API Web Service Application.
3. Click the **Endpoints** tab on the **Overview** page. The **Endpoints** pane opens.

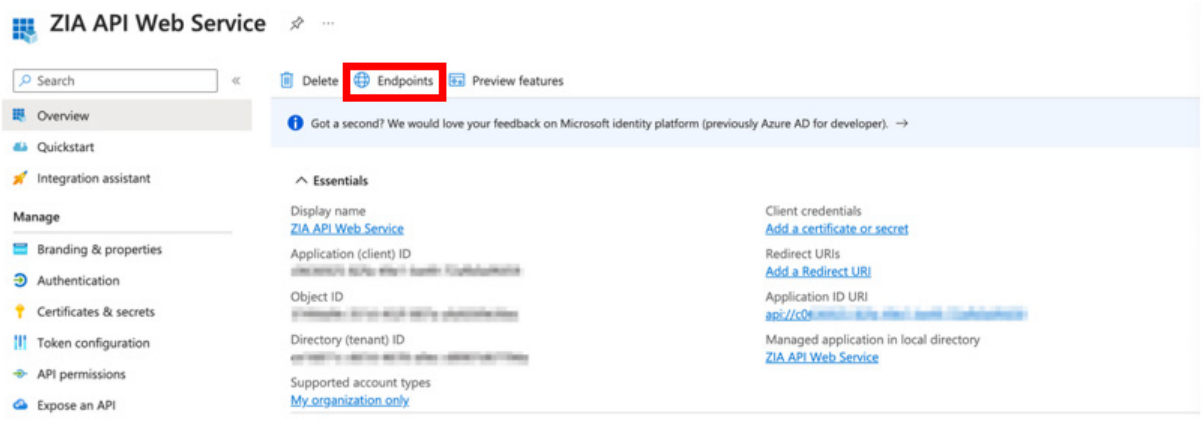


Figure 355. Endpoints

©2026 Zscaler, Inc. All rights reserved.

4. In the **Endpoints** pane:
   a. Copy the **OAuth 2.0 token endpoint (v2) URL**. The typical format is `https://login.microsoftonline.com/<Application ID>/oauth2/v2.0/token`
   b. Copy the URL displayed in the **OpenID Connect metadata document** field and open the link in a new browser window. Locate the jwks_uri parameter in the metadata displayed and copy its value. The typical format is `https://login.microsoftonline.com/<Application ID>/discovery/v2.0/keys`



Figure 356. Endpoints URL

These values are required for configuring the OAuth 2.0 authorization server in the ZIA Admin Portal, as explained in the next section.

©2026 Zscaler, Inc. All rights reserved.

Configuring the OAuth2.0 Authorization Server

After completing the necessary configurations in Microsoft Entra ID, you can validate the authorization server configuration and retrieve an access token.

1. Log in to the ZIA Admin Portal.
2. Go to **Administration** > **Cloud Service API Security**.
3. Click the **OAuth 2.0 Authorization Servers** tab and then click **Add Authorization Server**. The **Add Authorization Server** window opens.
4. In the **Add Authorization Server** window, fill out the following information, which is necessary to validate your authorization server configuration:
   a. **Enable**: Ensure this toggle is on.
   b. **Name**: Enter a name for your authorization server configuration (e.g., `Entra ID`). The name can only contain alphanumeric characters without spaces and cannot exceed 64 characters.
   c. **OAuth 2.0 JWKS Location**: Enter the JSON Web Key Set (JWKS) endpoint value obtained through the jwks_uri parameter in Collect Information for Zscaler OAuth 2.0 Server Configuration. The JWKS endpoint returns the public key set of the authorization server that is used by Zscaler to cryptographically verify the authenticity of the JWT in API requests.

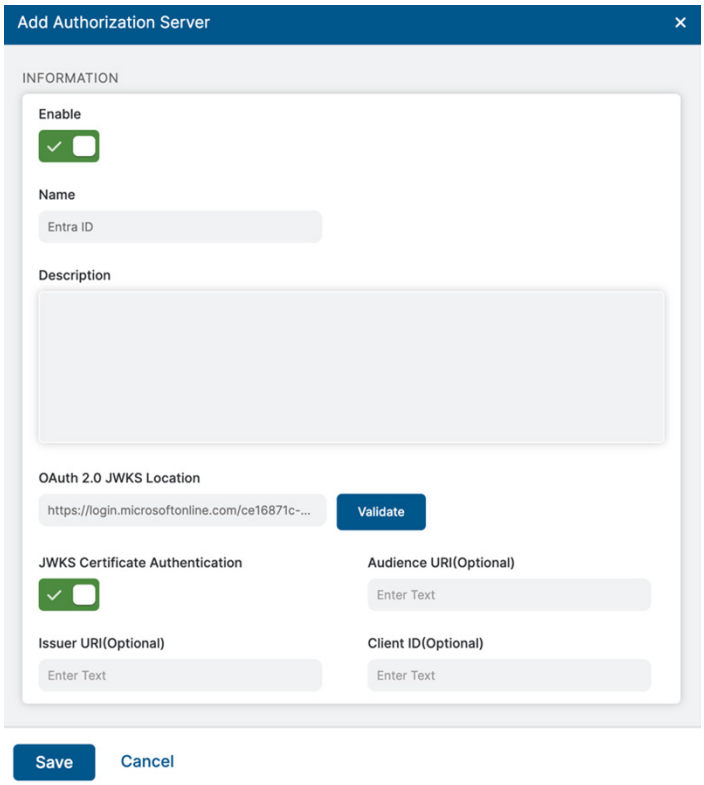


Figure 357. Add Authorization Server

©2026 Zscaler, Inc. All rights reserved.

Authenticating an OAuth 2.0 Session in Postman

Zscaler supports the Windows, macOS, and Linux versions of the Postman REST API app. To learn more about the app and its features, refer to the Postman documentation.

Installing and Configuring Postman for Windows, macOS, or Linux

To install and configure Postman:

1. Go to the Postman website and download the app for your OS (i.e., Windows, macOS, or Linux).
2. Install the app.
3. After installation, open the app and log in using your account.
4. Download the latest version of the cloud service API collection file from the Reference Guide (government agencies, see Reference Guide). Click Try in Postman to download the collection.
5. From the main window in Postman, click **Import**.

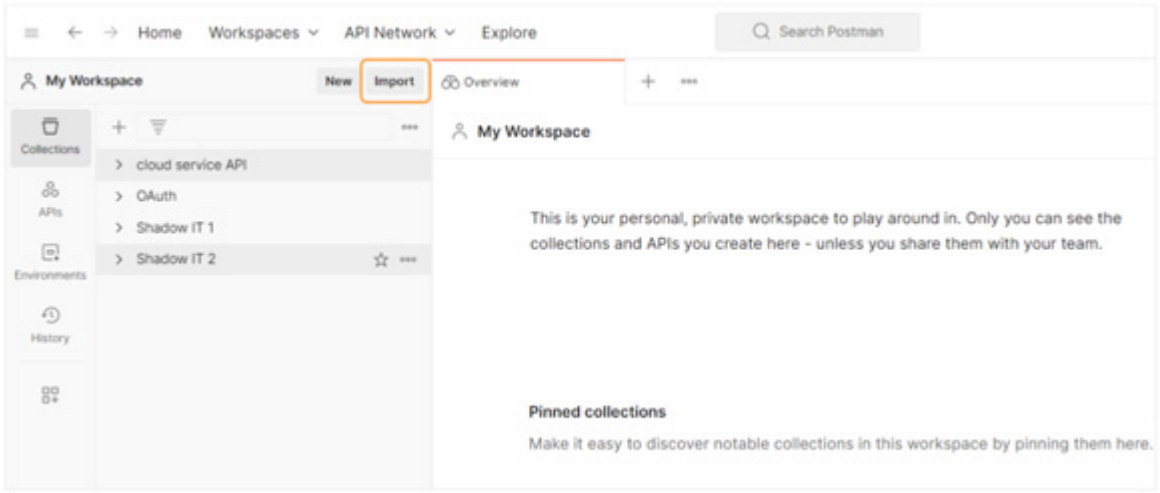


Figure 358. Install Postman

6. In the **Import** window that appears, select your **Postman collection** file, or drag the file to the selection area.

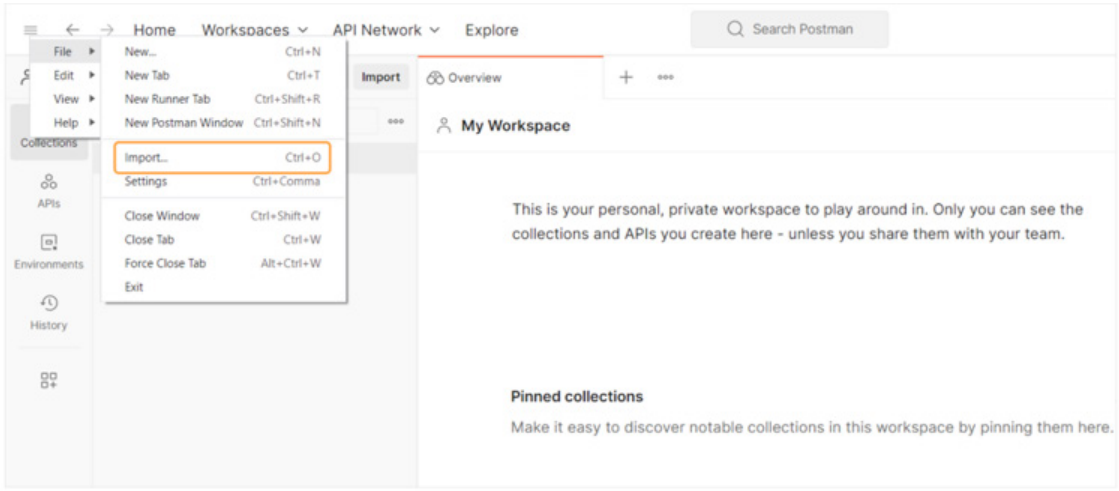


Figure 359. Postman file

After the file is imported, a new folder with the name used within the Postman collection file (e.g., cloud service API) is displayed within **Collections**.

©2026 Zscaler, Inc. All rights reserved.

7. Ensure that **No Environment** is selected in the environment drop-down menu on the top right, and then click the **Environment** quick look icon.
8. Click **Add**.

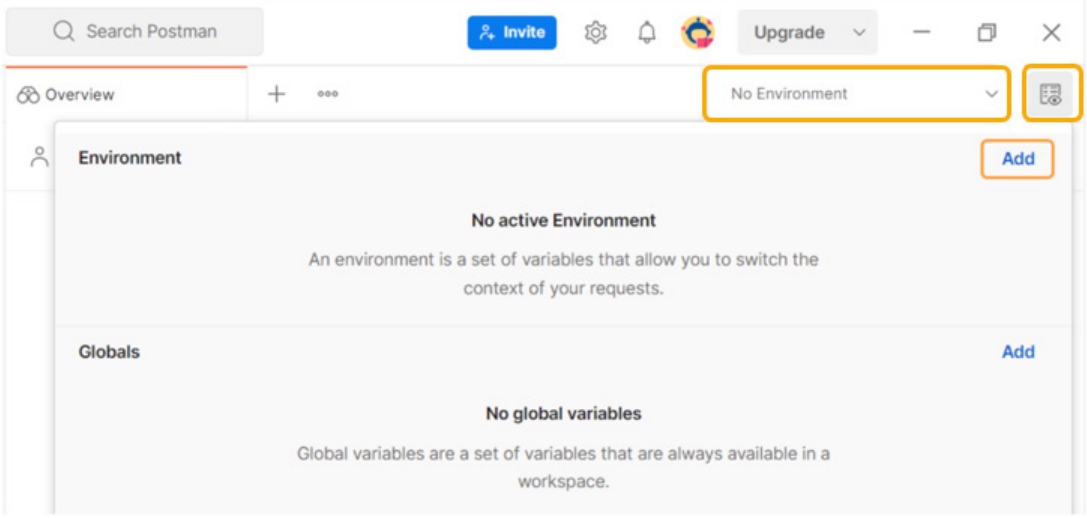


Figure 360. Add environment

9. On the **New Environment** tab that appears, complete the following steps:
   a. Enter a descriptive name for the environment (e.g., `ZIA – zscalertwo.net`).
   b. Create a new variable the base URL.
      - Under **Variable**, enter `url`.
      - For **Type**, leave as default.
      - For **Initial Value**, enter the URL found in your ZIA Admin Portal, under **Administration** > **Cloud Service API Key**.

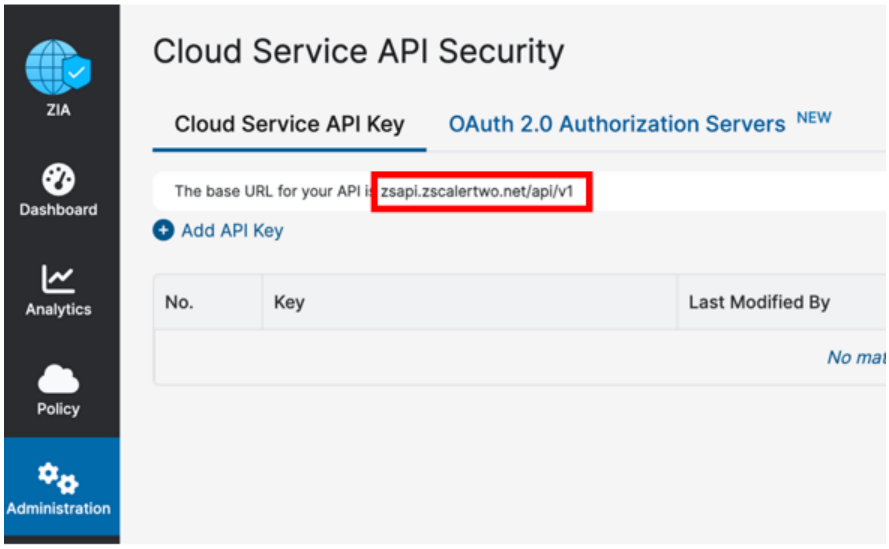


Figure 361. Cloud Service API Security

©2026 Zscaler, Inc. All rights reserved.

c. Create a new variable for the **OAuth 2.0 Token Endpoint**:
   - Under **Variable**, enter `tokenUrl`.
   - For **Type**, leave as default.
   - For **Initial Value**, enter the OAuth 2.0 token endpoint (v2) URL that you copied from the Collect Information for Zscaler OAuth 2.0 Server Configuration section.

d. Create a new variable for the ZIA API Client Application client ID:
   - Under **Variable**, enter `clientId`.
   - For **Type**, leave as default.
   - For **Initial Value**, enter the **Application (client) ID** for your ZIA API Client Application that you copied in the Register the ZIA API Client Application section.

e. Create a new variable for the ZIA API Client Application client secret:
   - Under **Variable**, enter `clientSecret`.
   - For **Type**, leave as default.
   - For **Initial Value**, enter the client secret value for your ZIA API Client Application that you copied in the Configure Client Credentials section.

f. Create a new variable for the ZIA API Client Application scope:
   - Under **Variable**, enter `scope`.
   - For **Type**, leave as default.
   - For **Initial Value**, enter the **Application ID URI** from your ZIA API Web Service Application, appended with `./default`. For example, `api://c0636925-82fa-49e1-be49-72afb0a9fd59/.default`

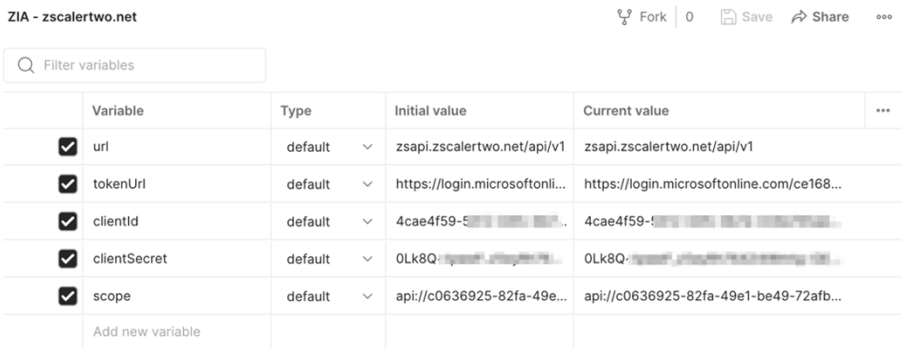


Figure 362. Create variables

g. Click **Save** and close the tab.

h. Select the environment you configured (e.g., `ZIA - zscalertwo.net`) from the environment drop-down menu on the top right. To learn more, refer to the Postman documentation.

©2026 Zscaler, Inc. All rights reserved.

10. Go to **Collections** > **Cloud service API** > **URL Categories** > **GET URL Categories – Get all**. Under **Authorization**, select **OAuth 2.0** from the drop-down menu.

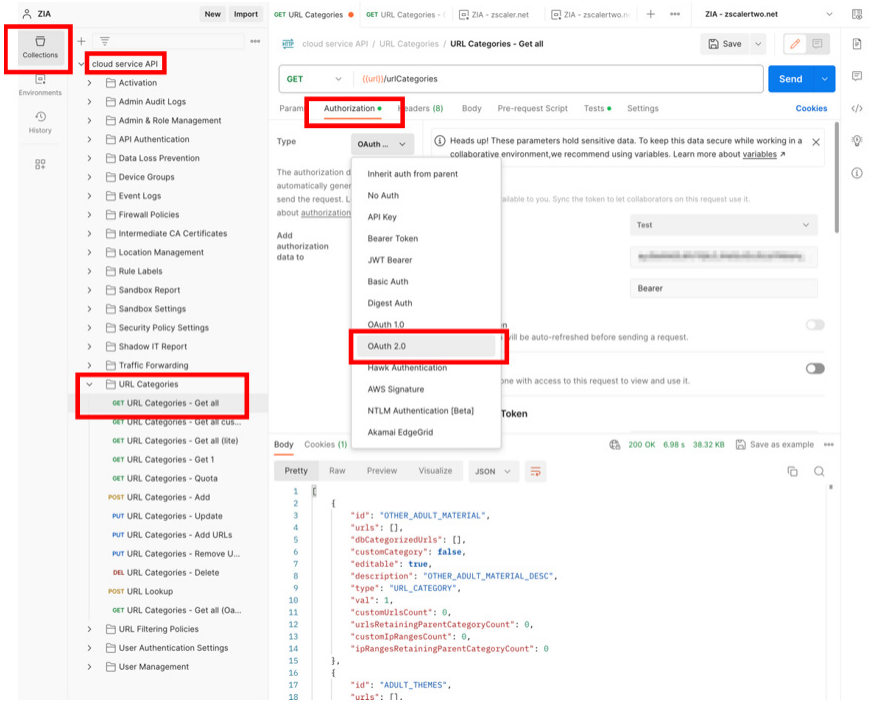


Figure 363. Get URL Categories – Get all

©2026 Zscaler, Inc. All rights reserved.

11. Ensure the environment you have created is selected in the top right. Under **Configure New Token**, complete the following:
    a. For **Access Token URL**, enter `{{tokenURL}}`
    b. For **Client ID**, enter `{{clientId}}`
    c. For **Client Secret**, enter `{{clientSecret}}`
    d. For **Scope**, enter `{{scope}}`

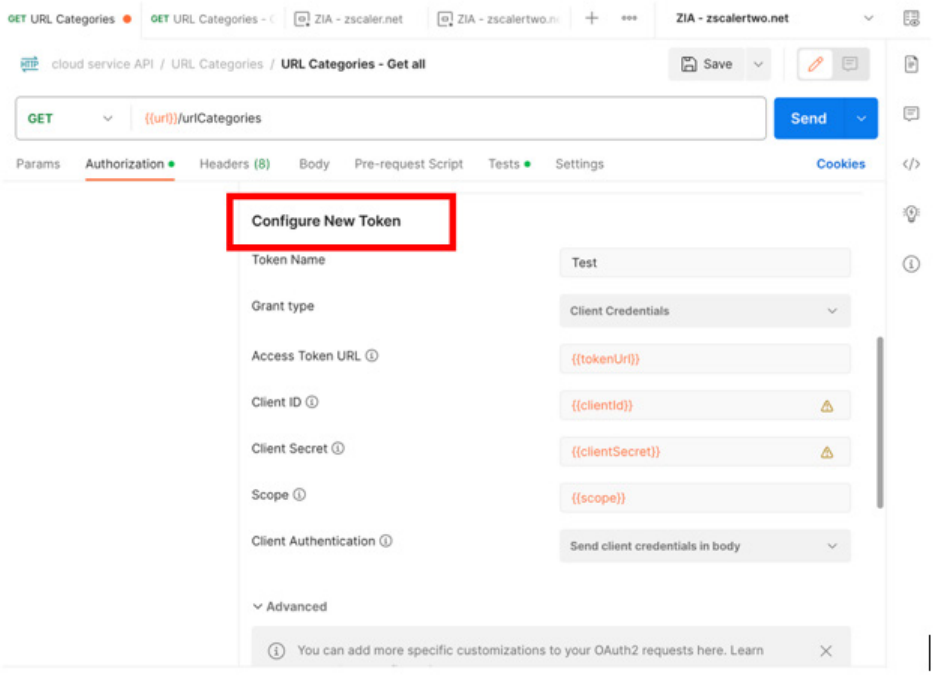


Figure 364. Configure new token

12. Click **Get New Access Token**, and then select **Use Token**. This is your access token, which must be sent in the API calls made to the ZIA API service. You can present the access token in the request Authorization header using the bearer authentication scheme along with the token expiration time.

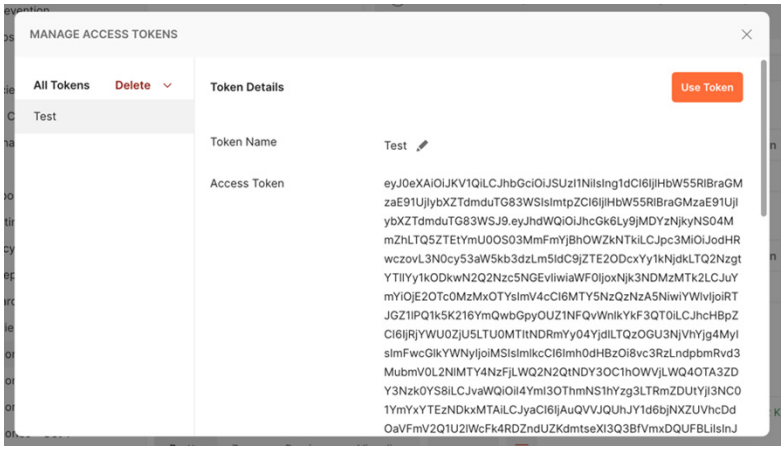


Figure 365. Get new access token

13. Click **Send** and receive a response back with a list of all URL categories configured in your tenant.

©2026 Zscaler, Inc. All rights reserved.

OneAPI Authentication

OneAPI authentication is the process of verifying the identity of an API client attempting to access the Zscaler service API resources. ZIdentity is the centralized platform that acts as an authorization server to validate the API clients and provide access to the API resources. This process ensures that only authorized API clients can interact with the API resources, protecting the applications and sensitive data from unauthorized access and potential security breaches.

How OneAPI Authentication Works

ZIdentity provides support for JSON Web Key Set (JWKS) URL, Certificate or Public Key, and Client Secret authentication methods for authenticating to OneAPI. You can choose any of these authentication methods.

JWKS and Certificate

The authentication workflow for JWKS and Certificate includes the following stages:

1. The API client that is configured in the ZIdentity Admin Portal generates a JSON Web Token (JWT) on its local system, signing it with a private key.
2. The public key corresponding to the private key used to sign the JWT is hosted at the JWKS URL on a web server that is accessible and trusted by ZIdentity.
3. The API client uses the JWT and makes a request to ZIdentity. ZIdentity retrieves the public key from the JWKS URL to verify the JWT's signature. If the signature is valid, the API client is then authenticated and granted access to the API resource.

Client Secret

The authentication workflow for Client Secret includes the following stages:

1. The API client and ZIdentity (API server) must agree on a shared secret key prior to accessing the API resources. This secret is stored securely on both the API client and ZIdentity. For OneAPI, the secret key is created when defining an API client.
2. When the API client makes a request, the secret is included in the request header, usually in an encrypted or hashed form for security purposes.
3. ZIdentity validates the secret against its saved copy. If the secret matches, the API client is authenticated and granted access to the API resource.

©2026 Zscaler, Inc. All rights reserved.

Configuring OneAPI Authentication

The following sections describe configuring OneAPI Authentication.

Create an API Role

To create an API role:

1. Go to **Administration** > **Role Management**.

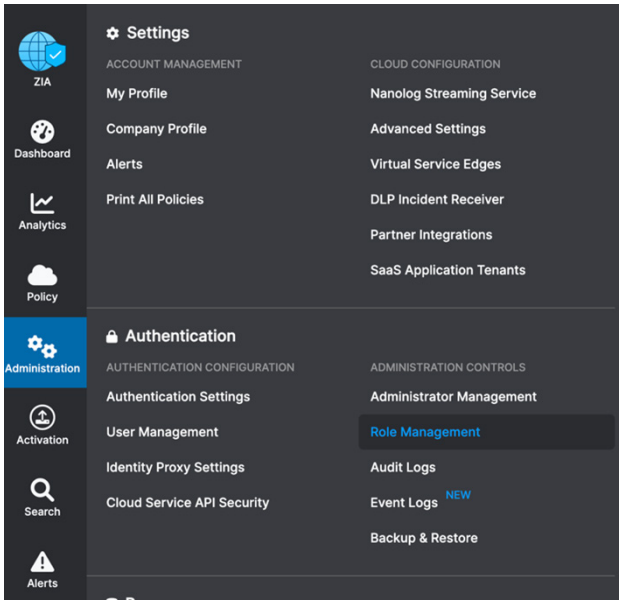


Figure 366. Role Management

2. Click **Add API Role**.

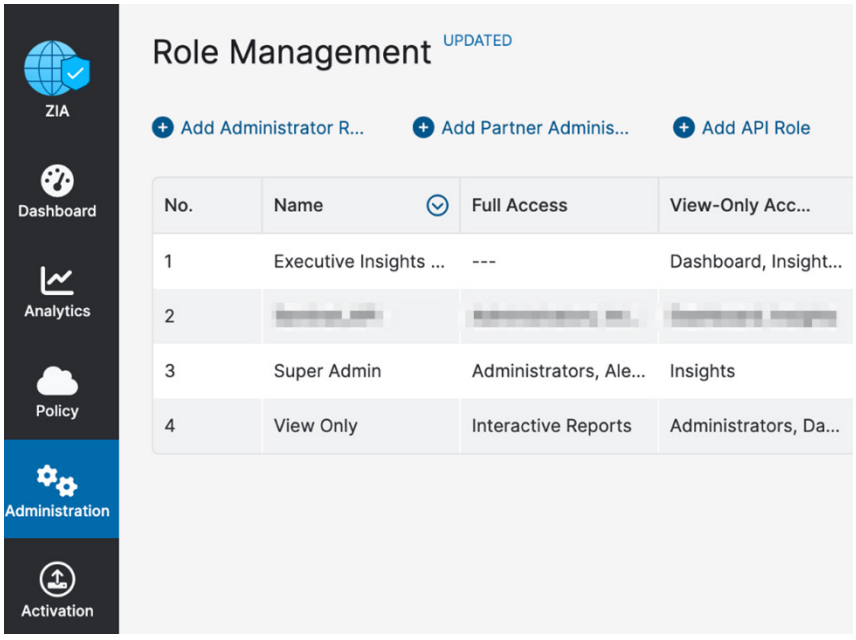


Figure 367. Add API Role

©2026 Zscaler, Inc. All rights reserved.

3. Add an API role named `Sentinel-Playbooks` and enable the configuration shown in the following figure.

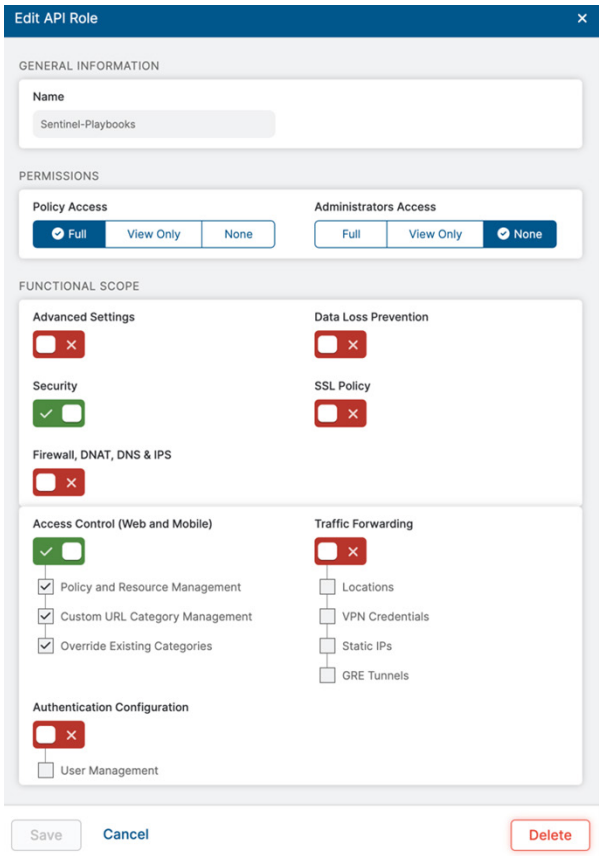


Figure 368. Add API Role

4. Click **Save**.

©2026 Zscaler, Inc. All rights reserved.

To add an API client:

1. Log in to ZIdentity, then go to **Integration** > **API Clients**.
2. Click **Add API Client**.
3. In the **Add API Client** drawer, the **Client** tab is selected by default. Enter the following details:
   a. **Name**: Enter a name to identify the API client (e.g.,, `Sentinel-Playbooks`).
   b. **Description**: Enter a brief description that indicates the purpose of the API client.
   c. **Status**: Enable the status. This allows the API client to authenticate and use the API resources.
   d. **Access Token Validity**: The validity of the access token that is used to access the API resource. Enter the time in minutes. The minimum and maximum validity periods are 1 minute and 24 hours, respectively.
   e. **Client Authentication**: Select **Secret**, then click **+ Add**. A key is auto-generated and displayed along with the validity period. This key is generated once, and it cannot be updated. The minimum duration is 30 days and the maximum duration is 365 days. Ensure to copy the secret key and save it in your local folder as it is not going to be displayed again.

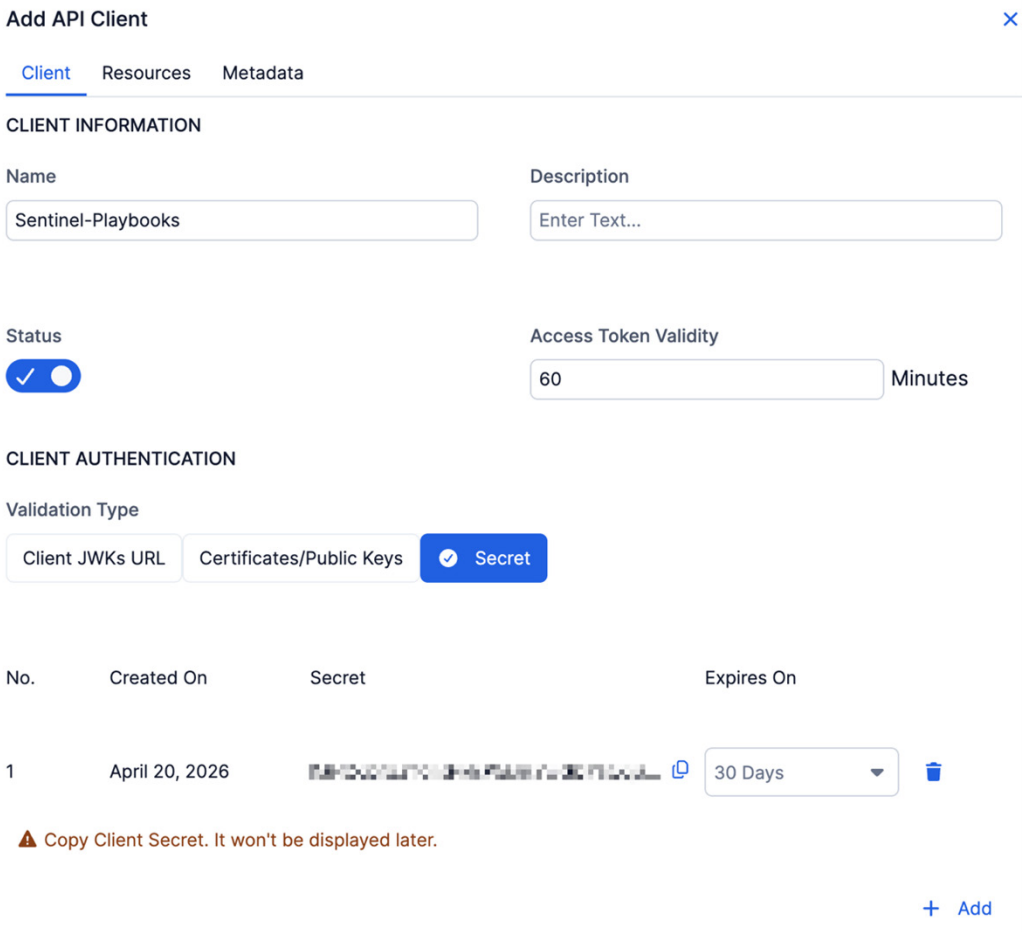


Figure 369. Configure API Client

4. Click **Resources**. Under ZIA, select the role you created earlier.

©2026 Zscaler, Inc. All rights reserved.

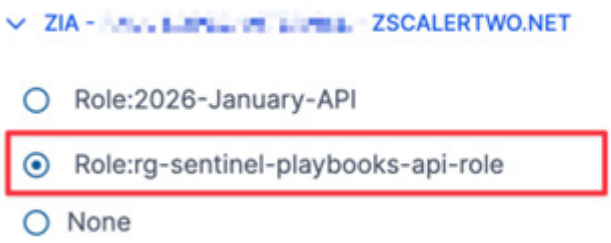


Figure 370. Resources

5. Click **Save**.

Deployment of Zscaler Playbooks for Microsoft Sentinel

Zscaler created several tested and ready-to-use Microsoft Sentinel Playbook templates that you can customize to meet a variety of needs.

These include:

Name	Definition
Zscaler-Oauth2-Authentication	This provides the authentication module which allows a Playbook to authentication to the ZIA API service using OAuth 2.0.
Zscaler-Oauth2-DenylistURL	Add URLs to the ATP Blocked Malicious URLs list.
Zscaler-Oauth2-BlockIP	Block an IP by adding to a URL category.
Zscaler-Oauth2-BlockURL	Block a URL by adding to a URL category.
Zscaler-Oauth2-LookupIP	Lookup the categories related to an IP.
Zscaler-Oauth2-LookupSandboxReport	Lookup a Sandbox Report using a MD5 Hash.
Zscaler-Oauth2-LookupURL	Lookup the categories related to a URL.
Zscaler-Oauth2-UndenylistURL	Remove URLs from the ATP Blocked Malicious URLs list.
Zscaler-Oauth2-UnblockIP	Remove an IP from a URL category.
Zscaler-Oauth2-UnblockURL	Remove a URL from a URL category.
Zscaler-Oauth2-AllowlistURL	Add a URL to the ATP allowlist.

For more information about playbook automation responses, refer to the Microsoft Sentinel Playbook documentation.

The remainder of this guide demonstrates the process of deploying the Playbook templates into Microsoft Sentinel.

Giving Permission to Microsoft Sentinel to Run Playbooks

Microsoft Sentinel needs permission to run playbooks. This is done on the resource group that contains the permissions. To configure permissions, complete the following:

1. Sign in to the Microsoft Azure portal.
2. Click the **Microsoft Sentinel** tile.
3. Go to **Microsoft Sentinel** and select your Log Analytics workspace.

©2026 Zscaler, Inc. All rights reserved.

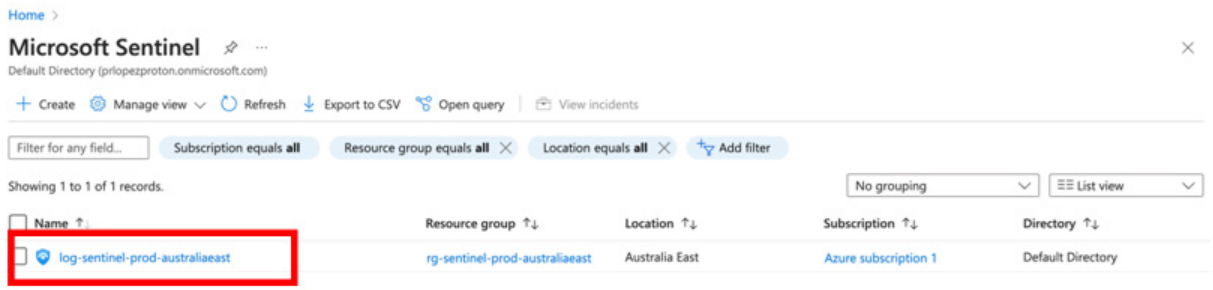


Figure 371. Log Analytics workspace

4. Go to **Settings** > **API permissions** > **Configured permissions** > **Grant admin consent for Zscaler Lab** and select the **zscalertwo.net API**.

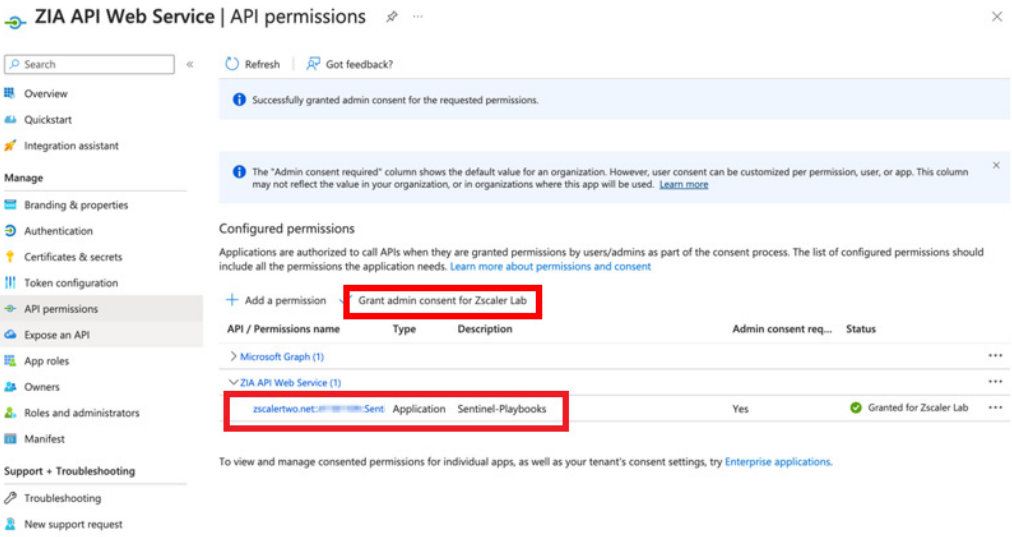


Figure 372. Configure permissions

5. Select the resource group that contains your playbooks and select **Apply**.

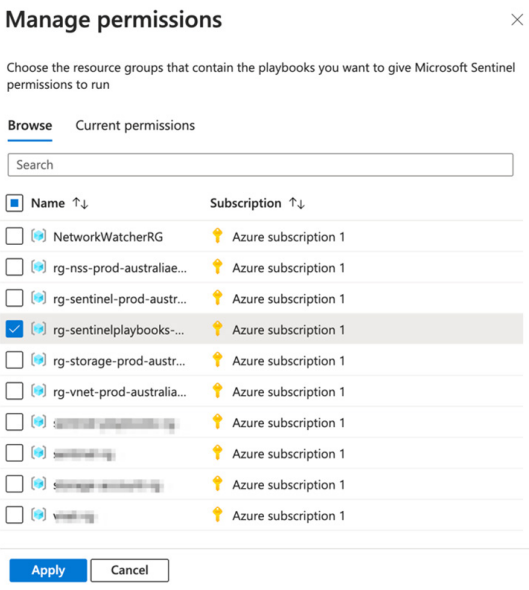


Figure 373. Manage permissions

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-Authenticaton Playbook

To install and configure this playbook to use the Azure Key Vault, complete the following:

1. Download the Zscaler-Oauth2-Authentication playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-Authentication** folder, review the `README.md` file, and click **Deploy to Azure**.

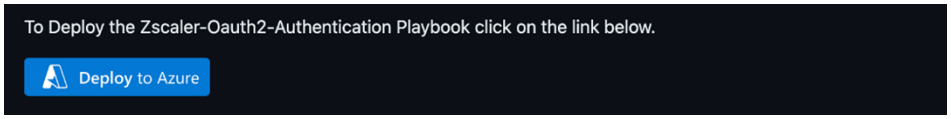


Figure 374. Deploy to Azure

3. Select the subscription, resource group, and region to which the template is deployed.
4. If using OneAPI, complete the following:
   a. **Resource Group**: Select the resource group to deploy to.
   b. **OneAPI Vanity Domain**: Select the vanity domain of your OneAPI deployment (e.g., `z-41181109`).
   c. **OneAPI Client Id**: Enter the client ID of your OneAPI Client.
   d. **OneAPI Client Secret**: Select the client secret of your OneAPI Client.
   e. **Legacy Base URL**: Leave as is as this won't be used for OneAPI.

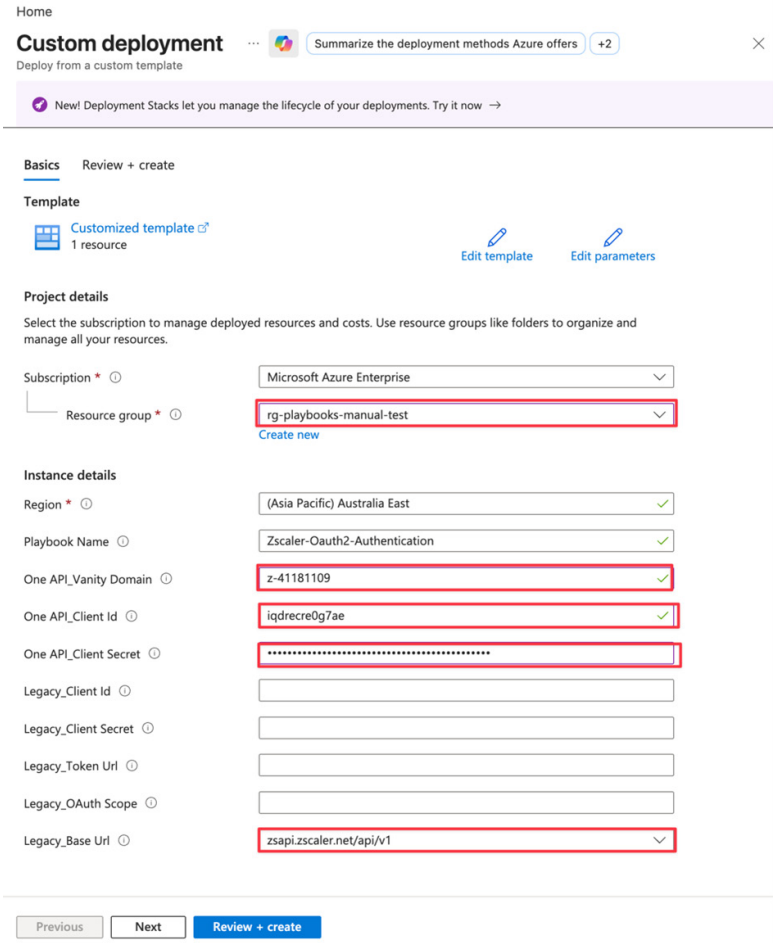


Figure 375. Custom Deployment

©2026 Zscaler, Inc. All rights reserved.

5. Click **Review + create**, then **Create**.
6. If using Legacy OAuth 2.0, complete the following:
   a. **Resource Group**: Select the resource group to deploy to.
   b. **Legacy Client ID**: Select your client_id from ZIA API Client Application.
   c. **Legacy Client Secret**: Enter your secret from the ZIA API Playbooks Applications.
   d. **Legacy Token URL**: Select your OAuth 2.0 token endpoint (v2) found under Endpoints in your ZIA API Playbooks Application.
   e. **Legacy OAuth Scope**: Select the Application ID URI from your ZIA API Web Service Application, appended with "./default".
   f. **Legacy Base URL**: Select your Zscaler cloud.

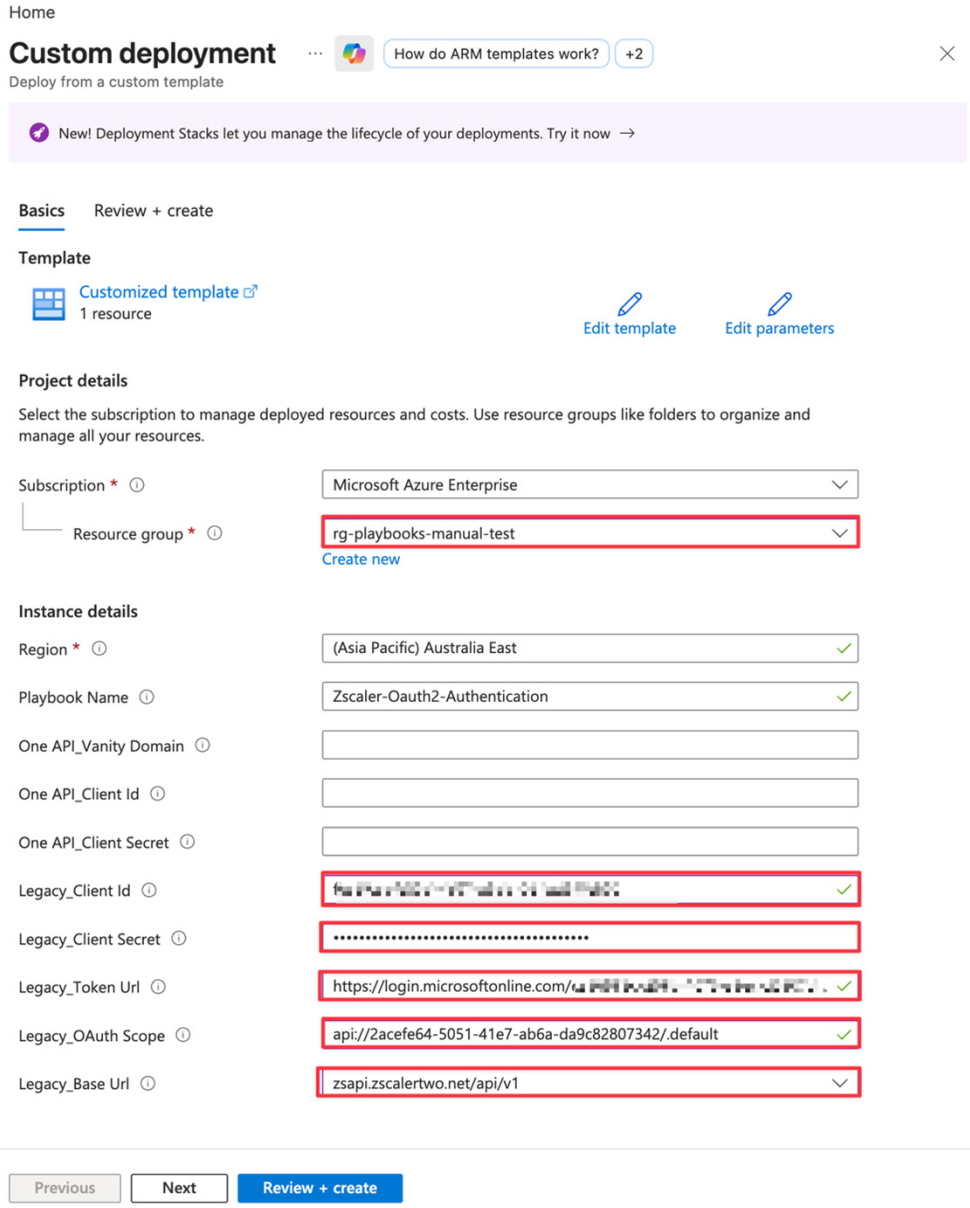


Figure 376. Custom Deployment

7. Click **Review + Create**, then **Create**.

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-BlockIP Playbook

To install and configure this playbook:

1. Download the Zscaler-Oauth2-BlockIP playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the Zscaler-Oauth2-BlockIP folder, review the `README.md` file, and click **Deploy to Azure**.
3. Select your subscription, resource group, and region you want to deploy the template to.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then click **Create**.

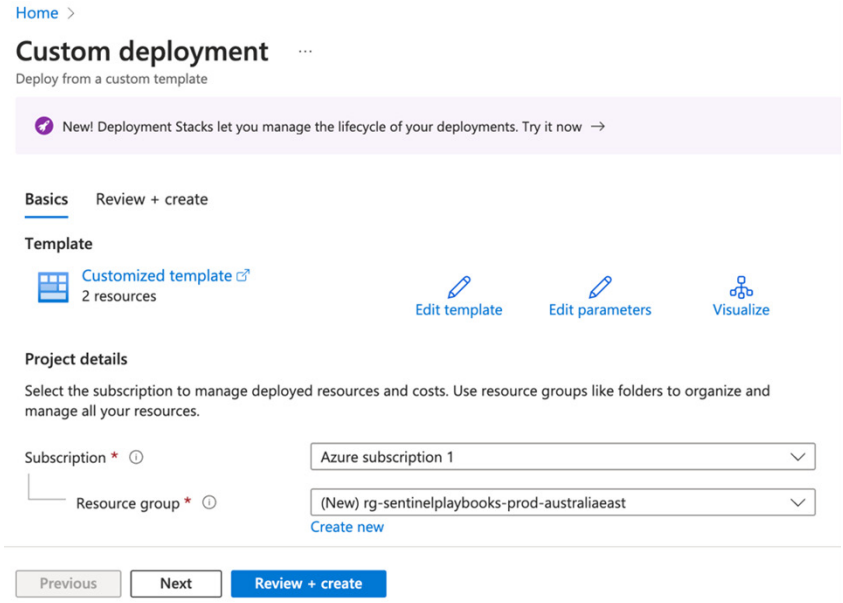


Figure 377. Custom deployment

6. Go to **Azure services** > **Logic apps** and your imported logic app appears on your list of logic apps.

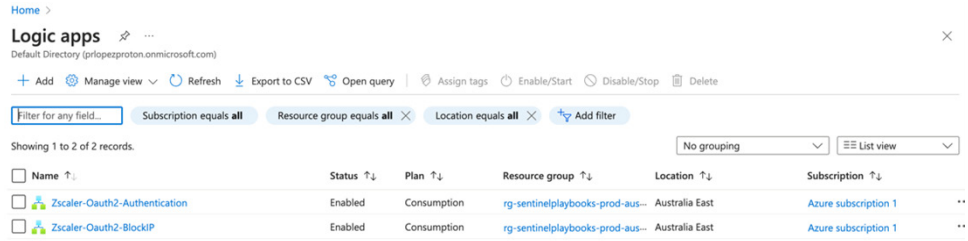


Figure 378. Logic apps

The playbook has been configured to add an IP to the **Other Miscellaneous** category. To change this, modify the Microsoft Logic app designer **Define Base URL** widget.

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-BlacklistURL Playbook

To install and configure this playbook, complete the following:

1. Download the Zscaler-Oauth2-BlacklistURL playbook, from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the Zscaler-Oauth2-BlacklistURL folder, review the `README.md` file, and click **Deploy to Azure**.

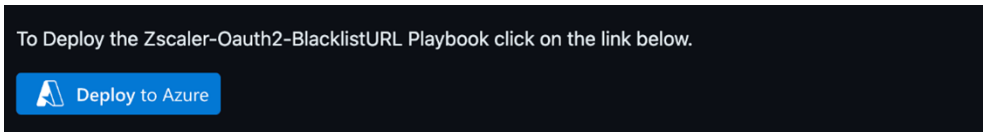


Figure 379. Deploy to Azure

3. Select your subscription, resource group, and region to which you want to deploy the template.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then click **Create**.

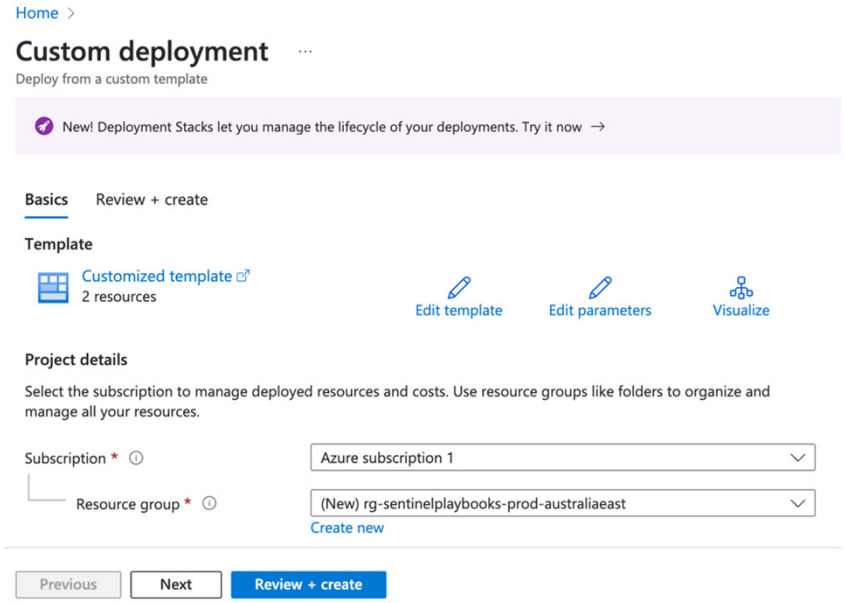


Figure 380. Custom Deployment

6. Go to **Azure services** > **Logic apps** and your imported logic app appears on your list of logic apps.

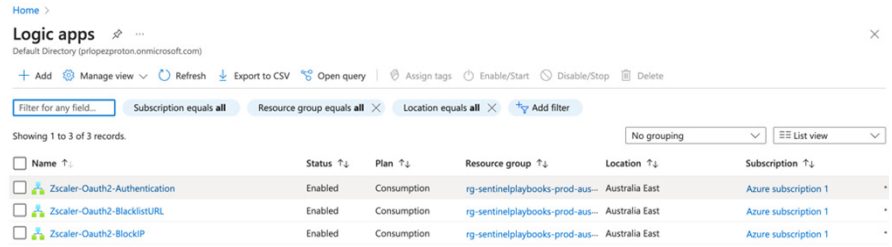


Figure 381. Logic apps

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-BlockURL Playbook

To install and configure this playbook, complete the following:

1. Download the Zscaler-Oauth2-BlockURL playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-BlockURL** folder, review the `README.md` file, and click **Deploy to Azure**.

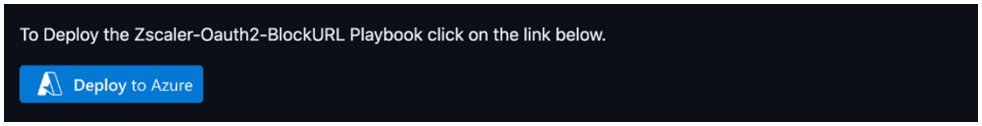


Figure 382. Deploy to Azure

3. Select the subscription, resource group, and region to which you want to deploy the template.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then **Create**.

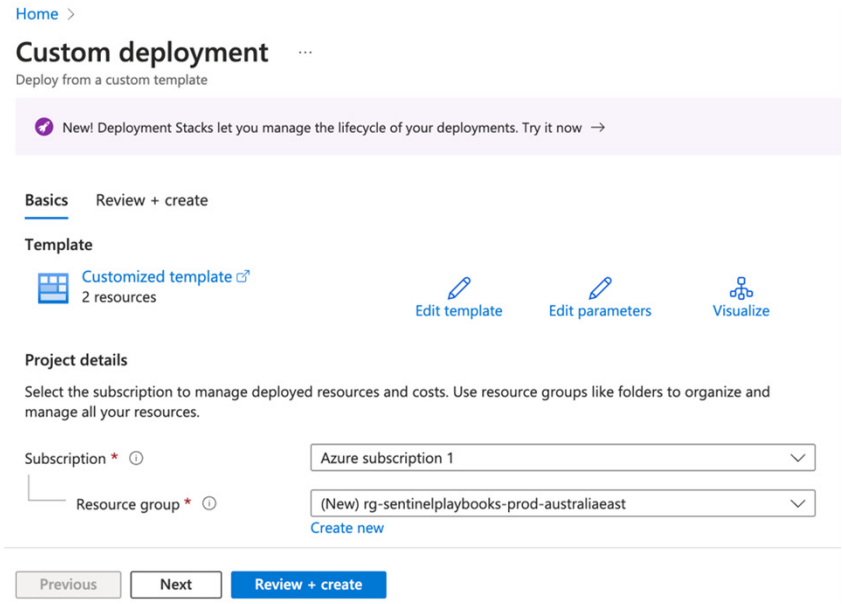


Figure 383. Custom deployment

6. Go to **Azure services** > **Logic apps** and your imported logic app appears on the list of logic apps.

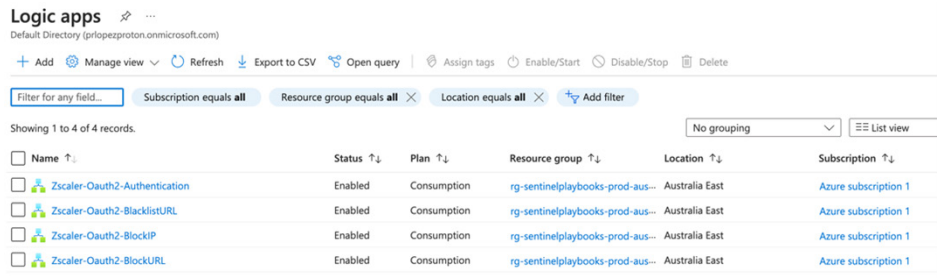


Figure 384. Logic apps

To change the URL category for this playbook, you can modify the Logic app designer **Define URL Category** widget.

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-LookupIP Playbook

To install and configure this playbook:

1. Download the Zscaler-Oauth2-LookupIP playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-LookupIP** folder, review the `README.md` file, and click **Deploy to Azure**.

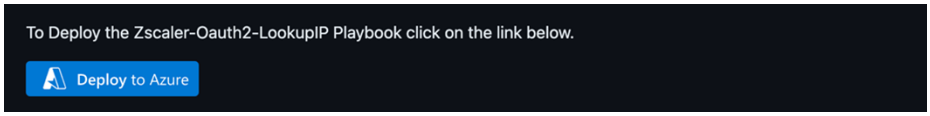


Figure 385. Deploy to Azure

3. Select the subscription, resource group, and region to which you want to deploy the template.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then **Create**.

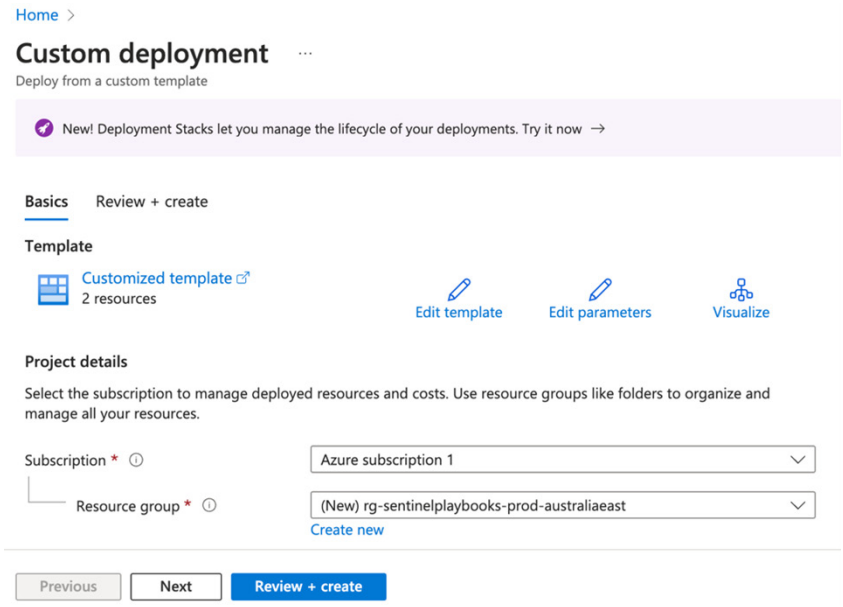


Figure 386. Custom deployment

6. Go to **Azure services** > **Logic apps** and your imported logic app appears on your list of logic apps.

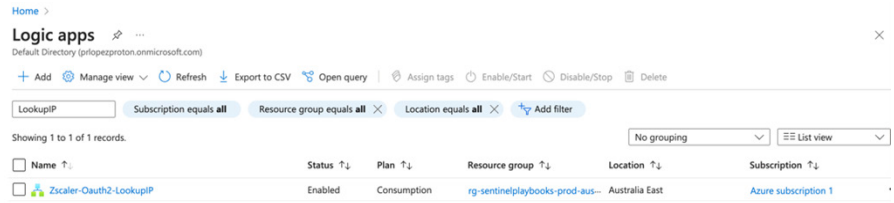


Figure 387. Logic apps

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-LookupSandboxReport Playbook

To install and configure this playbook, complete the following:

1. Download the Zscaler-Oauth2-LookupSandboxReport playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the Zscaler-Oauth2-LookupSandboxReport folder, review the `README.md` file, and click **Deploy to Azure**.

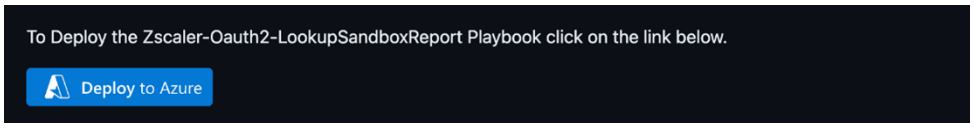


Figure 388. Deploy to Azure

3. Select your subscription, resource group, and region to which you want to deploy the template.
4. Click **Review + create**, then **Create**.

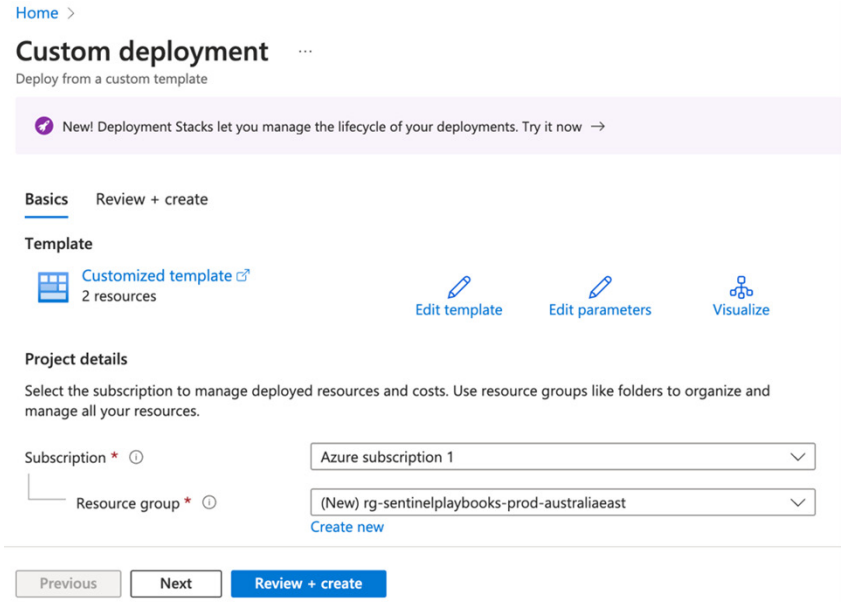


Figure 389. Custom deployment

Deploying the Zscaler-Oauth2-LookupURL Playbook

To install and configure this playbook, complete the following:

1. Download the Zscaler-Oauth2-LookupURL playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-LookupURL** folder, review the `README.md` file, and click the **Deploy to Azure**.

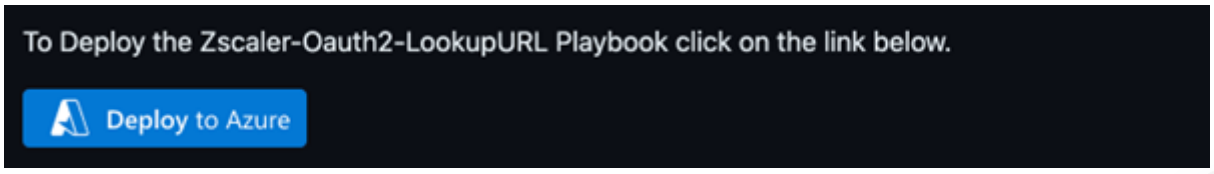


Figure 390. Deploy to Azure

©2026 Zscaler, Inc. All rights reserved.

3. Select your subscription, resource group, and region to which you want to deploy the template.
4. Click **Review + create**, then **Create**.

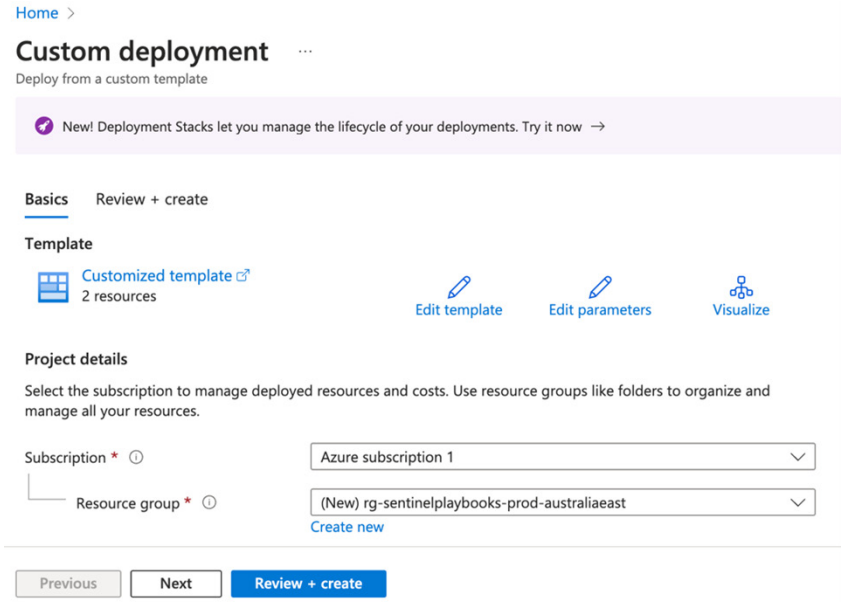


Figure 391. Custom deployment

5. Go to **Azure services** > **Logic apps** and your imported logic app appears on your list of logic apps.

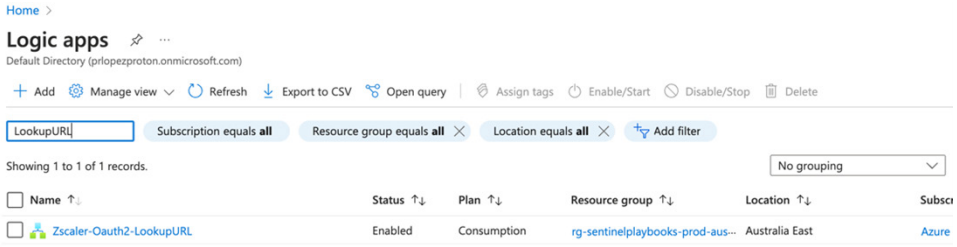


Figure 392. Logic apps

6. To authorize the connection, select the **Zscaler-Oauth2-LookupURL** logic app, then go to **Logic app designer** and select the **Entities – Get URLs** widget.
7. Click **Change connection**.

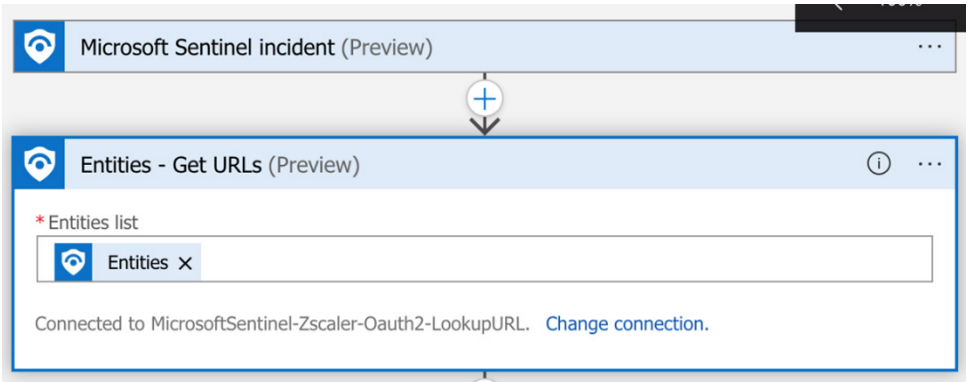


Figure 393. Change connection

©2026 Zscaler, Inc. All rights reserved.

8. Select **Add new** > **Connect with managed identity**.

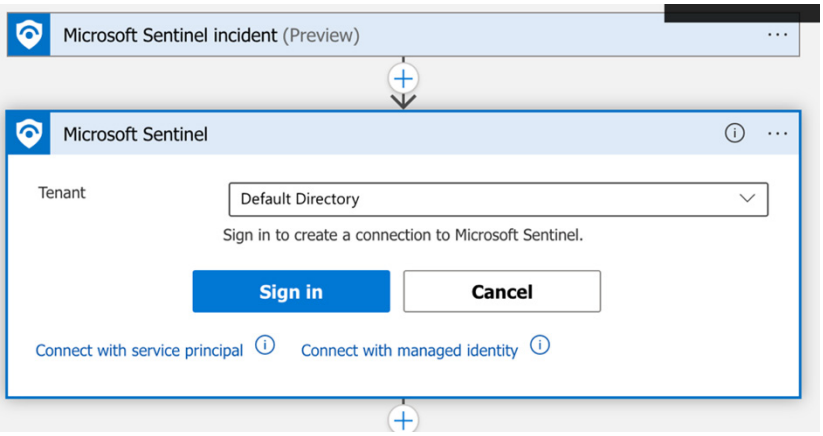


Figure 394. Connect with manage identity

9. For **Connection name**, input a connection name, such as `Zscaler-Oauth2-LookupURL`.
10. Click **Create**, then click **Save**.

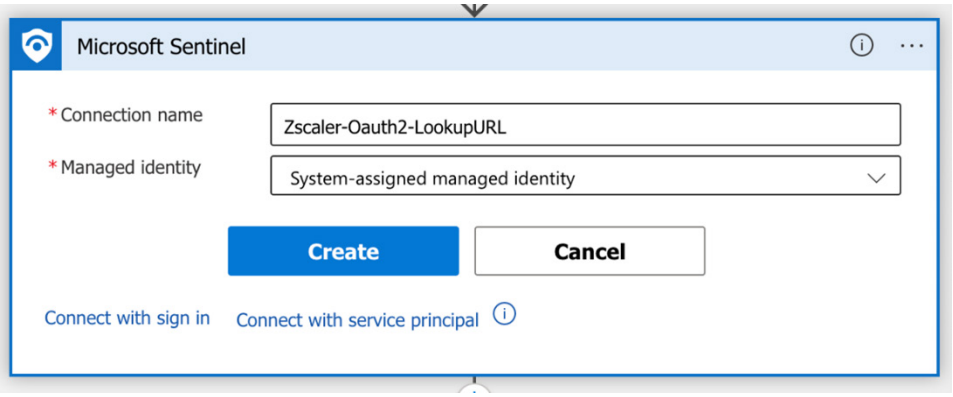


Figure 395. Connection name

To change the ZIA base URL for this playbook, you can modify the Microsoft Logic app designer **Initialize variable - Define Base URL** widget.

Deploying the Zscaler-Oauth2-UnblacklistURL Playbook

To install and configure this playbook, complete the following:

1. Download the **Zscaler-Oauth2-UnblacklistURL** playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-UnblacklistURL** folder, review the `README.md` file, and click **Deploy to Azure**.

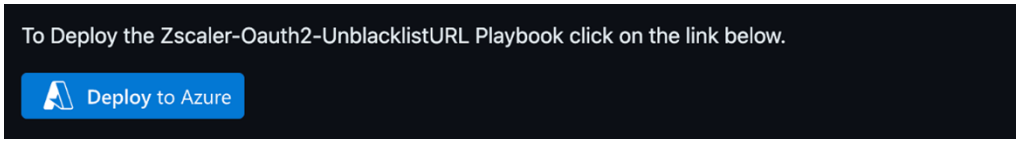


Figure 396. Deploy to Azure

©2026 Zscaler, Inc. All rights reserved.

3. Select your subscription, resource group, and region to which you want to deploy the template.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then **Create**.

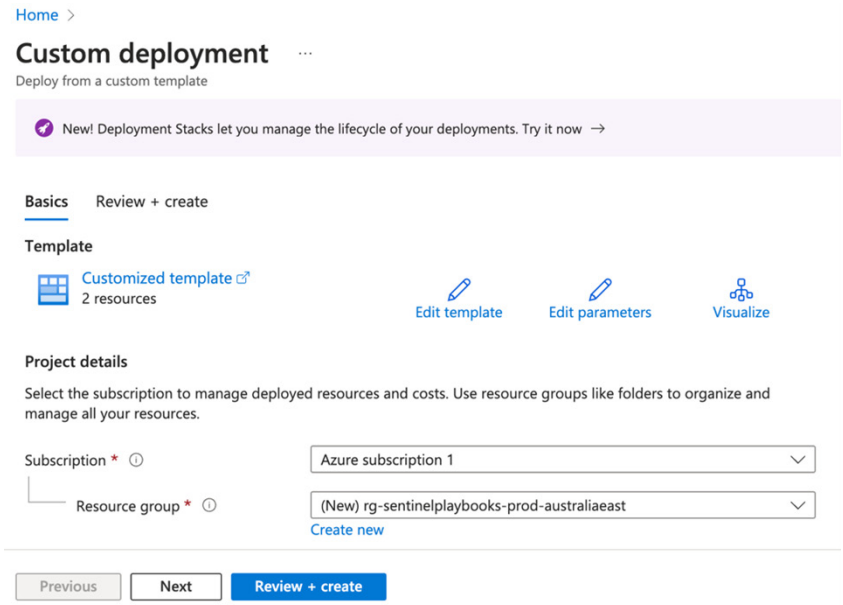


Figure 397. Custom deployment

6. Go to **Azure services** > **Logic apps** and your imported logic app appears on your list of logic apps.

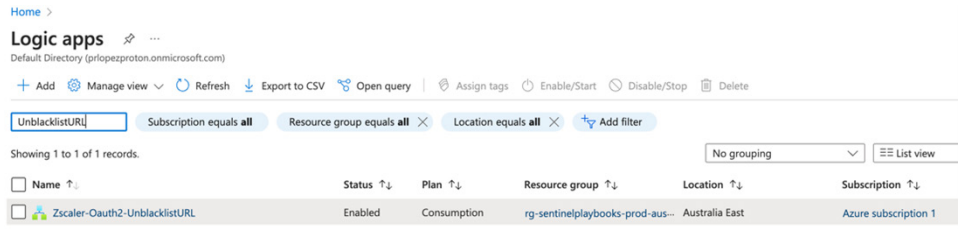


Figure 398. Logic apps

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-UnblockIP Playbook

To install and configure this playbook, complete the following:

1. Download the Zscaler-Oauth2-UnblockIP playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-UnblockIP** folder, review the `README.md` file, and click **Deploy to Azure**.

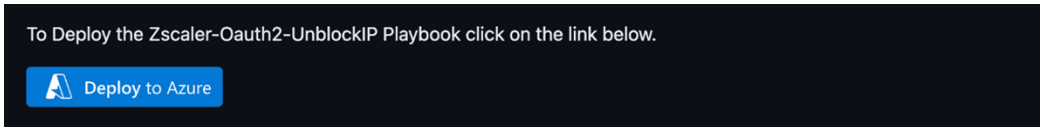


Figure 399. Deploy to Azure

3. Select your subscription, resource group, and region to which you want to deploy the template.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then **Create**.

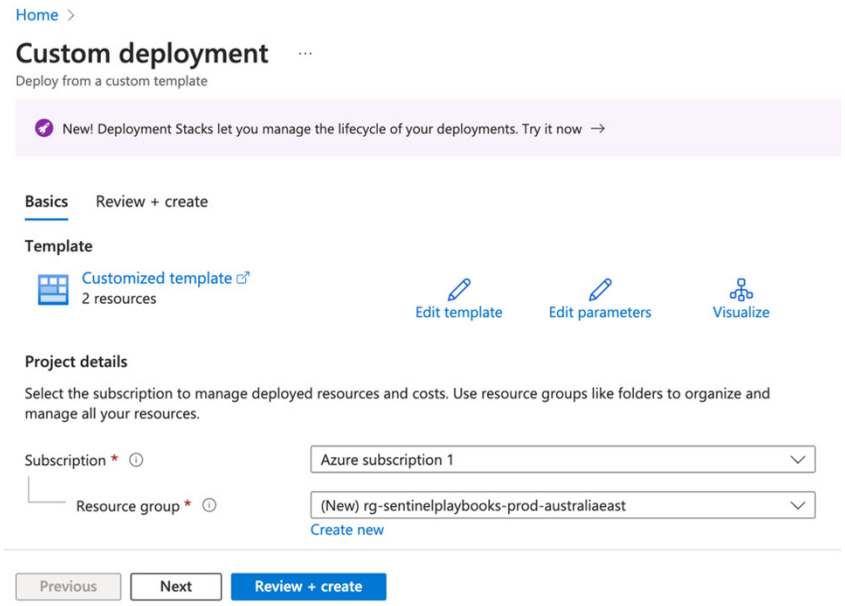


Figure 400. Custom deployment

6. Go to **Azure services** > **Logic apps** and your imported logic app appears on your list of logic apps.

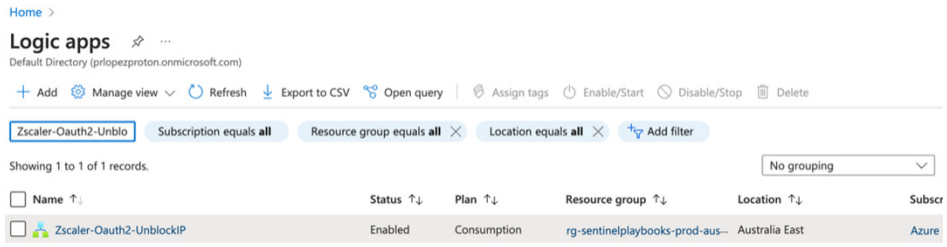


Figure 401. Logic apps

The playbook has been configured to remove an IP in the **Other Miscellaneous** category. To change it, modify the Microsoft Logic app designer **Define URL Category** widget.

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-UnblockURL Playbook

To install and configure this playbook:

1. Download the Zscaler-Oauth2-UnblockURL playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-UnblockURL** folder, review the `README.md` file, and click **Deploy to Azure**.

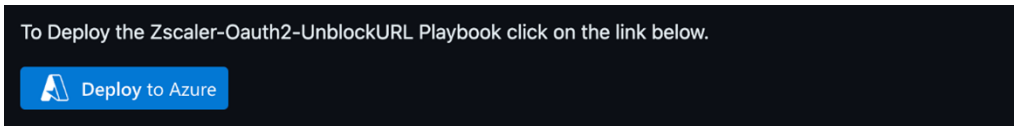


Figure 402. Deploy to Azure

3. Select your subscription, resource group, and region to which you want to deploy the template.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then **Create**.

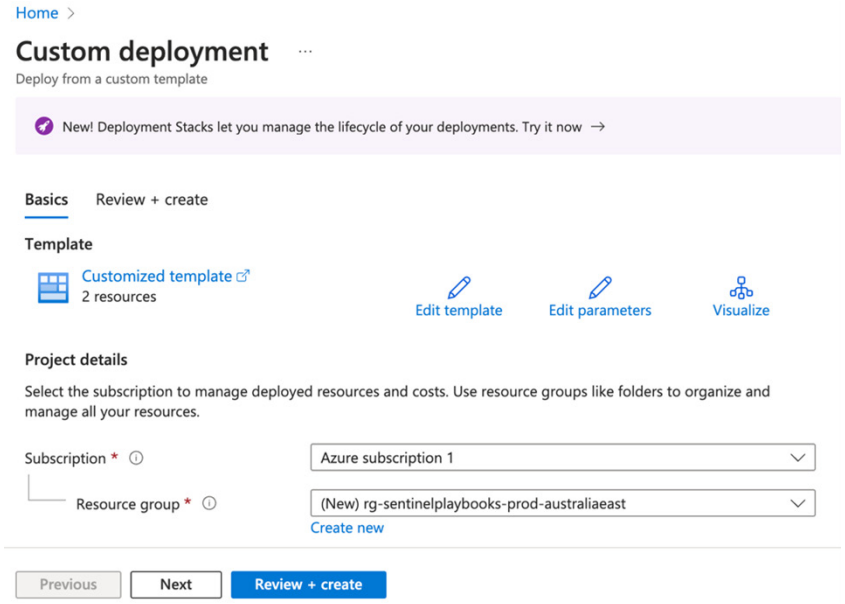


Figure 403. Custom deployment

6. Go to **Azure services** > **Logic apps** and the imported logic app appears on your list of logic apps.

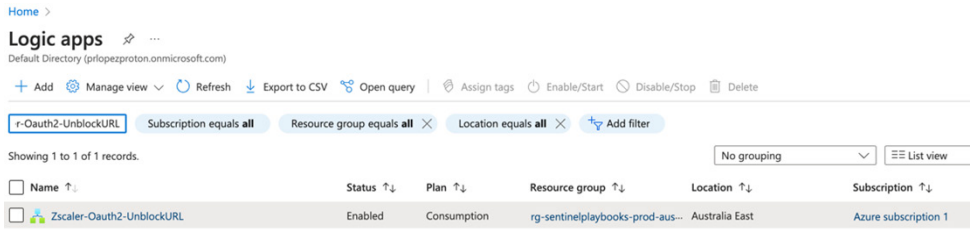


Figure 404. Logic apps

To change the URL category for this playbook, you can modify the Microsoft Logic app designer **Define URL category** widget.

©2026 Zscaler, Inc. All rights reserved.

Deploying the Zscaler-Oauth2-WhitelistURL Playbook

To install and configure this playbook, complete the following:

1. Download the Zscaler-Oauth2-WhitelistURL playbook from the Zscaler GitHub repo for Microsoft Sentinel Playbooks.
2. Go to the **Zscaler-Oauth2-WhitelistURL** folder, review the `README.md` file, and click **Deploy to Azure**.

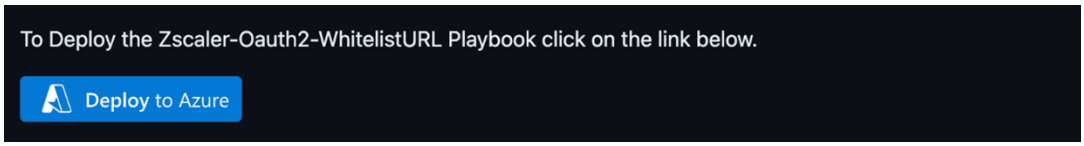


Figure 405. Deploy to Azure

3. Select your subscription, resource group, and region to which you want to deploy the template.
4. Ensure the existing Zscaler OAuth 2.0 Authentication Playbook name is correct.
5. Click **Review + create**, then **Create**.

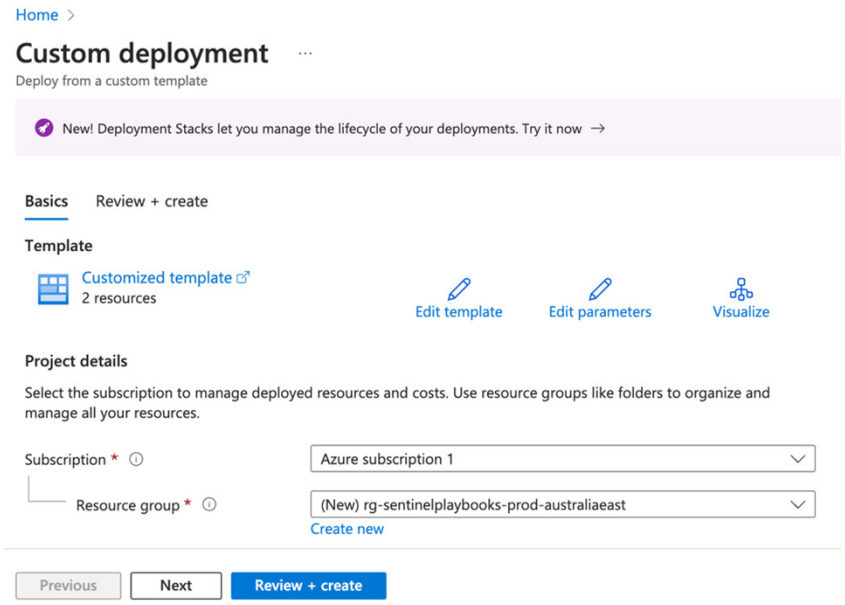


Figure 406. Custom deployment

6. Go to **Azure services** > **Logic apps** and your imported logic app appears on your list of logic apps.

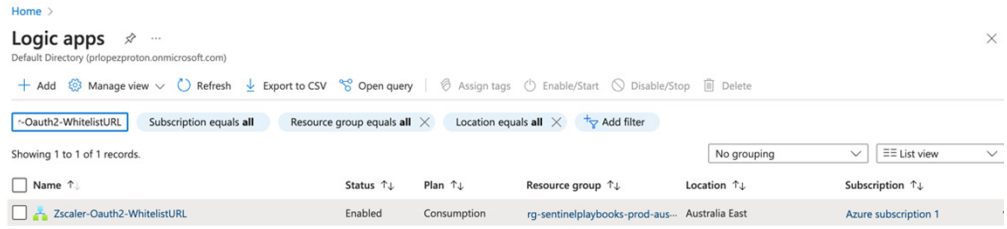


Figure 407. Logic apps

©2026 Zscaler, Inc. All rights reserved.

Appendix A: Requesting Zscaler Support

You might sometimes need Zscaler Support for provisioning certain services, or to help troubleshoot configuration and service issues. Zscaler Support is available 24/7/365.

To contact Zscaler Support:

1. Go to **Administration** > **Settings** > and then click **Company Profile**.

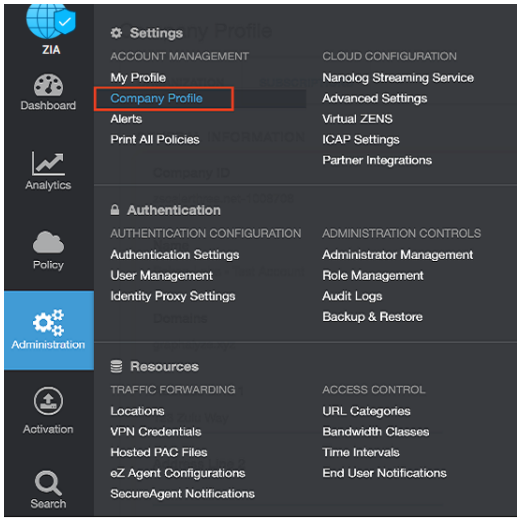


Figure 408. Collecting details to open support case with Zscaler TAC

2. Copy the Company ID if you are using ZIA Admin Portal. Copy the tenant ID if you are using ZPA Admin Portal.

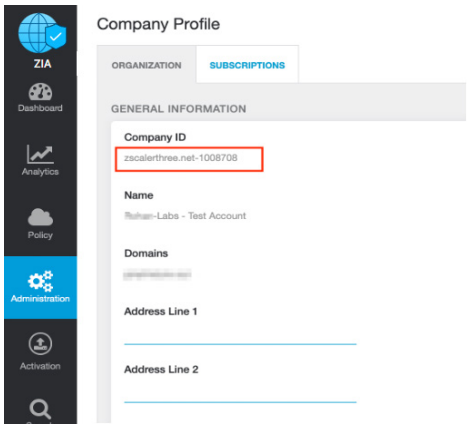


Figure 409. ZIA Company ID

©2026 Zscaler, Inc. All rights reserved.

3. Now that you have your ID, you can open a support ticket. For ZIA, go to **Dashboard** > **Support** > **Submit a Ticket**. For ZPA, go to **Dashboard** > **Tools** > **Submit a Ticket**.

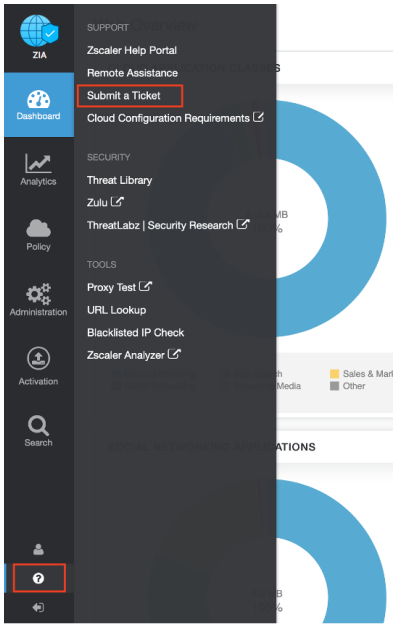


Figure 410. Submit a ticket

©2026 Zscaler, Inc. All rights reserved.