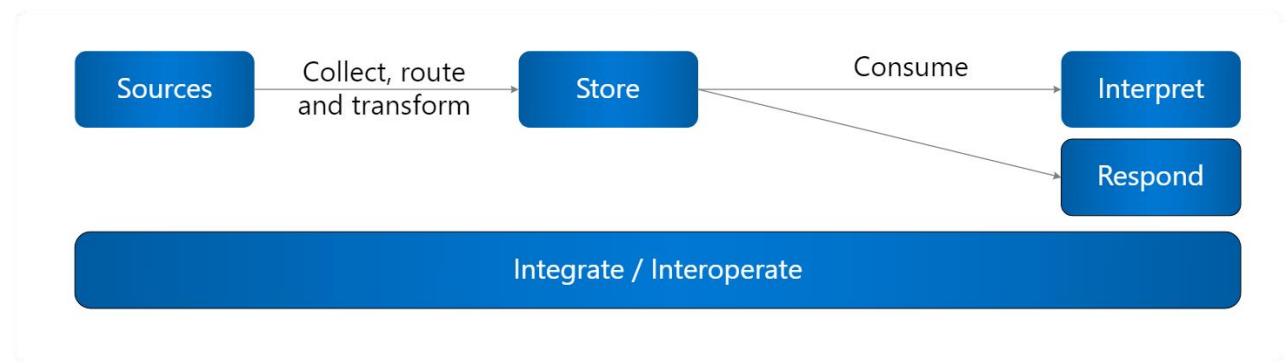


Azure Monitor overview

Azure Monitor is a comprehensive monitoring solution for collecting, analyzing, and responding to monitoring data from your cloud and on-premises environments. You can use Azure Monitor to maximize the availability and performance of your applications and services.

Azure Monitor collects and aggregates the data from every layer and component of your system across multiple Azure and non-Azure subscriptions and tenants.



High level architecture

Azure Monitor can monitor these types of resources in Azure, other clouds, or on-premises:

- Applications
- Virtual machines
- Guest operating systems
- Containers including Prometheus metrics
- Databases
- Security events in combination with [Azure Sentinel](#)
- Network events and health in combination with [Network Watcher](#)
- Custom sources that use the APIs to get data into [Azure Monitor](#)

You can also export monitoring data from Azure Monitor into other systems so you can:

- Integrate with other third-party and open-source monitoring and visualization tools
- Integrate with ticketing and other ITSM systems

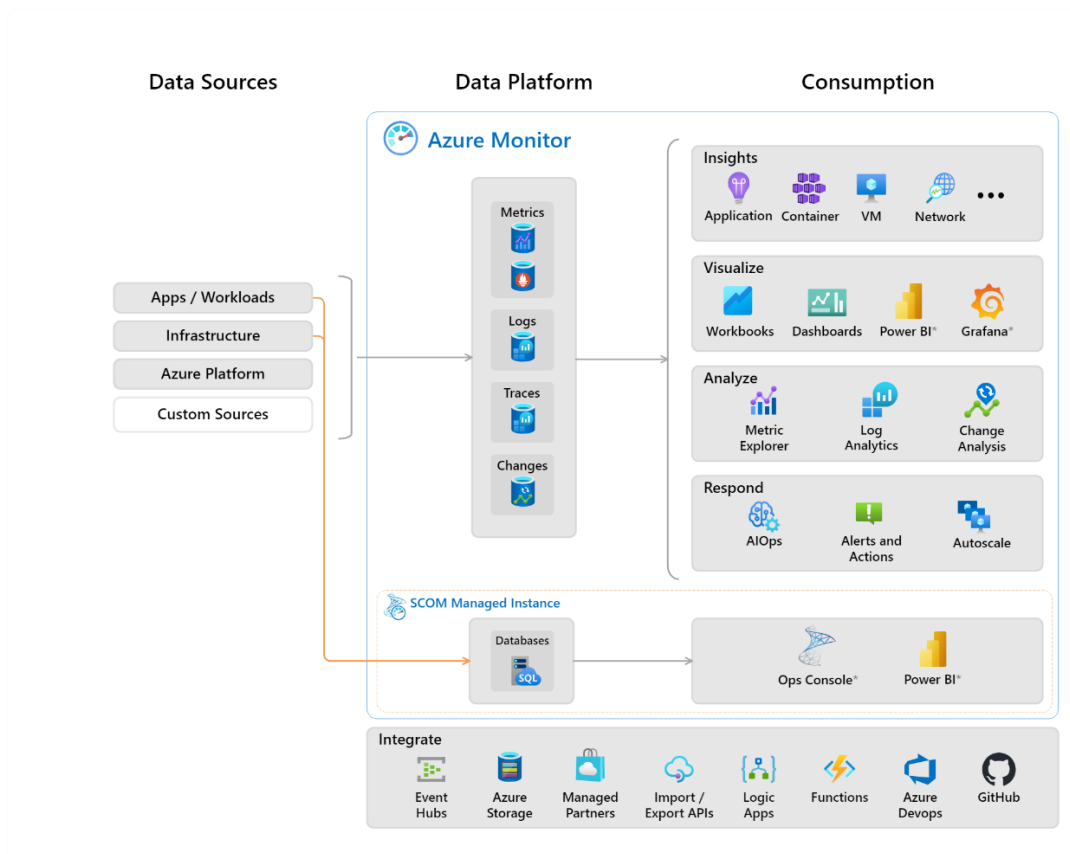
If you're a System Center Operations Manager (SCOM) user, Azure Monitor now includes Azure Monitor [SCOM Managed Instance \(SCOM MI\)](#).

Destinations

Platform logs and metrics can be sent to the destinations listed in the following table.

To ensure the security of data in transit, all destination endpoints are configured to support TLS 1.2.

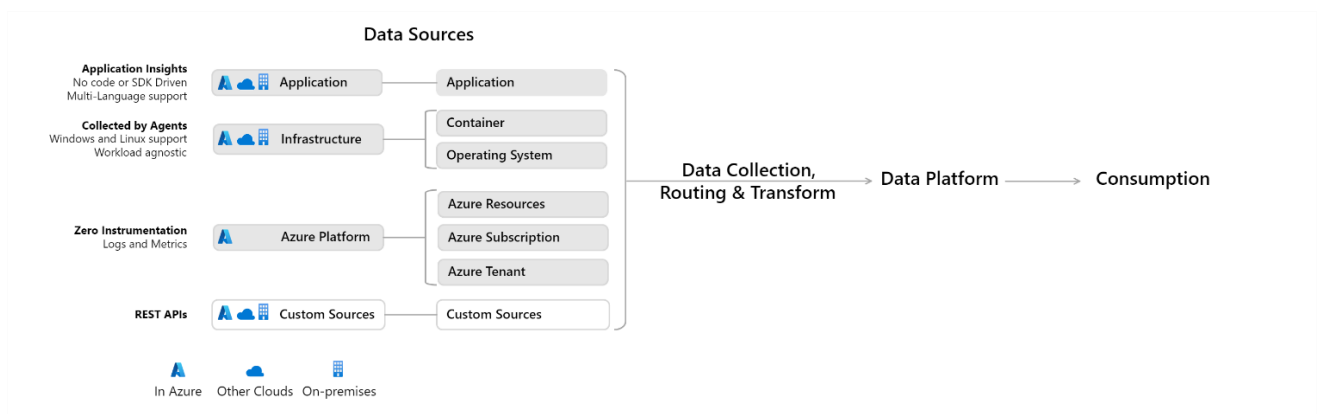
Destination	Description
Log Analytics workspace	Metrics are converted to log form. This option might not be available for all resource types. Sending them to the Azure Monitor Logs store (which is searchable via Log Analytics) helps you to integrate them into queries, alerts, and visualizations with existing log data.
Azure Storage account	Archiving logs and metrics to a Storage account is useful for audit, static analysis, or back up. Compared to using Azure Monitor Logs or a Log Analytics workspace, Storage is less expensive, and logs can be kept there indefinitely.
Azure Event Hubs	When you send logs and metrics to Event Hubs, you can stream data to external systems such as third-party SIEMs and other Log Analytics solutions.
Azure Monitor partner solutions	Specialized integrations can be made between Azure Monitor and other non-Microsoft monitoring platforms. Integration is useful when you're already using one of the partners.



Data sources

Azure Monitor can collect data from multiple sources.

The diagram below shows an expanded version of the data source types that Azure Monitor can gather monitoring data from.



You can integrate application, infrastructure, and custom data source monitoring data from outside Azure, including from on-premises, and non-Microsoft clouds.

Azure Monitor collects these types of data:

Data Type	Description and subtypes
App/Workloads	<p>App- Application performance, health, and activity data.</p> <p>Workloads - IaaS workloads such as SQL server, Oracle or SAP running on a hosted Virtual Machine.</p>
Infrastructure	<p>Container - Data about containers, such as Azure Kubernetes Service, Prometheus, and the applications running inside containers.</p> <p>Operating system - Data about the guest operating system on which your application is running.</p>
Azure Platform	<p>Azure resource - Data about the operation of an Azure resource from inside the resource, including changes. Resource Logs are one example.</p> <p>Azure subscription - The operation and management of an Azure subscription, and data about the health and operation of Azure itself. The activity log is one example.</p> <p>Azure tenant - Data about the operation of tenant-level Azure services, such as Microsoft Entra ID.</p>
Custom Sources	<p>Data that gets into the system using the</p> <ul style="list-style-type: none"> - Azure Monitor REST API - Data Collection API

SCOM MI (like on premises SCOM) collects only IaaS Workload and Operating System sources.

Data type	Description	Data collection method
Activity log	The Activity log provides insight into subscription-level events for Azure services including service health records and configuration changes .	Collected automatically. View in the Azure portal or create a diagnostic setting to send it to other destinations. Can be collected in Log Analytics workspace at no charge. See Azure Monitor activity log.
Platform metrics	Platform metrics are numerical values that are automatically collected at regular intervals for different aspects of a resource. The specific metrics vary for each type of resource.	Collected automatically and stored in Azure Monitor Metrics. View in metrics explorer or create a diagnostic setting to send it to other destinations. See Azure Monitor Metrics overview and Supported metrics with Azure Monitor for a list of metrics for different services.
Resource logs	Provide insight into operations that were performed within an Azure resource . The content of resource logs varies by the Azure service and resource type.	You must create a diagnostic setting to collect resource logs. See Azure resource logs and Supported services, schemas, and categories for Azure resource logs for details on each service.

The Activity Log includes information like when a resource is modified or a virtual machine is started. You can view the Activity Log in the Azure portal or retrieve entries with PowerShell and the Azure CLI.

Create a diagnostic setting to send the Activity Log to one or more of these locations:

- Log Analytics workspace for more complex querying and alerting
- Azure Event Hubs to forwarding logs outside of Azure.
- Azure Storage for cheaper, long-term archiving.

Send the activity log to a Log Analytics workspace to enable the Azure Monitor Logs feature, where you:

- Correlate activity log data with other monitoring data collected by Azure Monitor.
- Consolidate log entries from multiple Azure subscriptions and tenants into one location for analysis together.
- Use log queries to perform complex analysis and gain deep insights on activity log entries.
- Use log search alerts with Activity entries for more complex alerting logic.
- Store activity log entries for longer than the activity log retention period.
- Incur no data ingestion or retention charges for activity log data stored in a Log Analytics workspace.
- The default retention period in Log Analytics is 90 days

Monitor | Activity log

Microsoft

Search (Cmd+/) << Activity Edit columns Refresh Export Activity Logs Download as CSV Logs Pin current filters Reset filters

Overview

Activity log

Alerts

Metrics

Logs

Service Health

Workbooks

Insights

Applications

Virtual Machines

Storage accounts

Looking for Log Analytics? In Log Analytics you can search for performance, diagnostics, health logs, and more. Visit Log Analytics

Search

Quick Insights

Management Group : None Subscription : 2 selected Event severity : All Timespan : Last 6 hours Add Filter

First 88 items.

Operation name
Write Subnets
> Write Subnets
Write Subnets
Write Subnets

Send to Azure Event Hubs

- Send the activity log to Azure Event Hubs to send entries outside of Azure, for example, to a third-party SIEM or other log analytics solutions.
- Activity log events from event hubs are consumed in JSON format with a records element that contains the records in each payload.
- The schema depends on the category and is described in Azure activity log event schema.

```
{
  "records": [
    {
      "time": "2019-01-21T22:14:26.9792776Z",
      "resourceId":
"/subscriptions/s1/resourceGroups/MSSupportGroup/providers/microsoft.support/supporttickets/115012112305841",
    }
  ]
}
```

```
"operationName": "microsoft.support/supporttickets/write",
"category": "Write",
"resultType": "Success",
"resultSignature": "Succeeded.Created",
"durationMs": 2826,
"callerIpAddress": "111.111.111.11",
"correlationId": "c776f9f4-36e5-4e0e-809b-c9b3c3fb62a8",
"identity": {
  "authorization": {
    "scope":
"/subscriptions/s1/resourceGroups/MSSupportGroup/providers/microsoft.support/supporttickets/11
5012112305841",
    "action": "microsoft.support/supporttickets/write",
    "evidence": {
      "role": "Subscription Admin"
    }
  },
  "claims": {
    "aud": "https://management.core.windows.net/",
    "iss": "https://sts.windows.net/72f988bf-86f1-41af-91ab-2d7cd011db47/",
    "iat": "1421876371",
    "nbf": "1421876371",
    "exp": "1421880271",
    "ver": "1.0",
    "http://schemas.microsoft.com/identity/claims/tenantid": "00000000-0000-0000-0000-
000000000000",
    "http://schemas.microsoft.com/claims/authnmethodsreferences": "pwd",
    "http://schemas.microsoft.com/identity/claims/objectidentifier": "2468adf0-8211-44e3-
95xq-85137af64708",
    "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn": "admin@contoso.com",
    "puid": "20030000801A118C",
    "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier":
"9vckmEGF7zDKk1YzIY8k0t1_EAPaXoeHyPRn6f413zM",
    "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname": "John",
    "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname": "Smith",
    "name": "John Smith",
    "groups": "cacfe77c-e058-4712-83qw-f9b08849fd60,7f71d11d-4c41-4b23-99d2-
d32ce7aa621c,31522864-0578-4ea0-9gdc-e66cc564d18c",
    "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name": "
admin@contoso.com",
    "appid": "c44b4083-3bq0-49c1-b47d-974e53cbdf3c",
    "appidacr": "2",
    "http://schemas.microsoft.com/identity/claims/scope": "user_impersonation",
    "http://schemas.microsoft.com/claims/authnclassreference": "1"
  }
},
"level": "Information",
"location": "global",
```

```

    "properties": {
      "statusCode": "Created",
      "serviceRequestId": "50d5cddb-8ca0-47ad-9b80-6cde2207f97c"
    }
  ]
}

```

Send to Azure Storage

- Send the activity log to an Azure Storage account if you want to retain your log data longer than 90 days for audit, static analysis, or back up.
- If you're required to retain your events for 90 days or less, you don't need to set up archival to a storage account. Activity log events are retained in the Azure platform for 90 days.

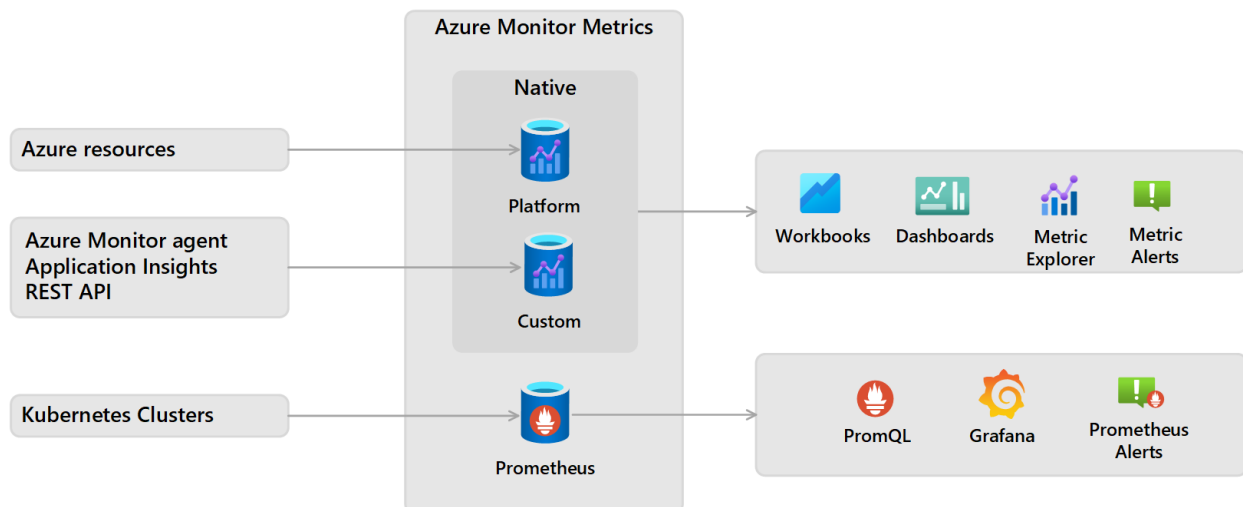
```

insights-activity-logs/resourceId=/SUBSCRIPTIONS/{subscription ID}/y={four-digit numeric
year}/m={two-digit numeric month}/d={two-digit numeric day}/h={two-digit 24-hour clock
hour}/m=00/PT1H.json

```

Azure Monitor Metrics overview

Azure Monitor Metrics is a feature of Azure Monitor that collects numeric data from monitored resources into a time-series database. Metrics are numerical values that are collected at regular intervals and describe some aspect of a system at a particular time.



The differences between each of the metrics are summarized in the following table.

Category	Native platform metrics	Native custom metrics	Prometheus metrics
Sources	Azure resources	Azure Monitor agent Application	Azure Kubernetes service (AKS) cluster Any Kubernetes cluster through remote-write (Prometheus metrics are stored for 18

		insights REST API	months , but a PromQL query can only span a maximum of 32 days.)
Configuration	None	Varies by source	Enable Azure Monitor managed service for Prometheus
Stored	Subscription	Subscription	Azure Monitor workspace
Cost	No	Yes	Yes (free during preview)
Aggregation	preaggregated	preaggregated	raw data
Analyze	Metrics Explorer	Metrics Explorer	PromQL Grafana dashboards
Alert	metrics alert rule	metrics alert rule	Prometheus alert rule
Visualize	Workbooks Azure dashboards Grafana	Workbooks Azure dashboards Grafana	Grafana
Retrieve	Azure CLI Azure PowerShell cmdlets REST API or client library .NET Go Java JavaScript Python	Azure CLI Azure PowerShell cmdlets REST API or client library .NET Go Java JavaScript Python	

Azure Monitor collects metrics from the following sources. After these metrics are collected in the Azure Monitor metric database, they can be evaluated together regardless of their source:

- **Azure resources**

Platform metrics are created by Azure resources and give you visibility into their health and performance.

Each type of resource creates a distinct set of metrics without any configuration required. Platform metrics are collected from Azure resources at one-minute frequency unless specified otherwise in the metric's definition.

- **Applications**

Application Insights creates metrics for your monitored applications to help you detect performance issues and track trends in how your application is being used. Values include Server response time and Browser exceptions.

- **Virtual machine agents**

Metrics are collected from the guest operating system of a virtual machine.

You can enable guest OS metrics for Windows virtual machines by using the Azure Monitor Agent.

Azure Monitor Agent replaces the legacy agents - Windows diagnostic extension and the InfluxData Telegraf agent for Linux virtual machines.

- **Custom metrics**

You can define metrics in addition to the standard metrics that are automatically available. You can define custom metrics in your application that's monitored by Application Insights. You can also create custom metrics for an Azure service by using the custom metrics API.

- **Kubernetes clusters**

Kubernetes clusters typically send metric data to a local Prometheus server that you must maintain.

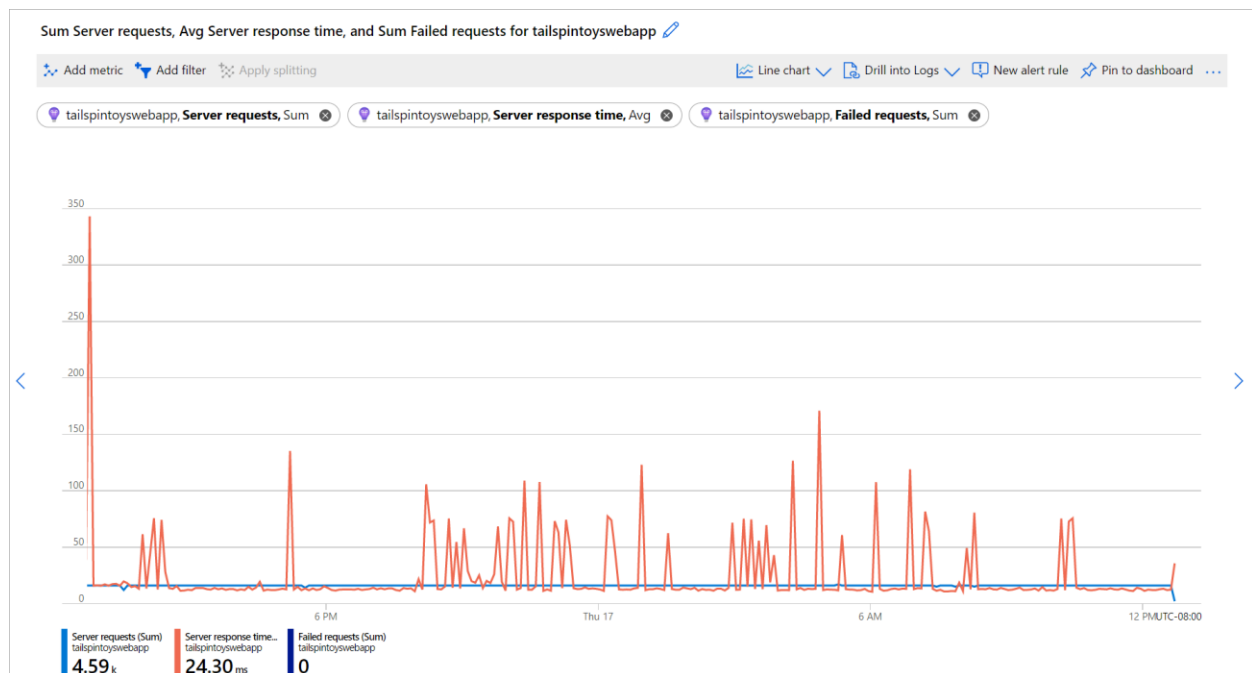
Azure Monitor managed service for Prometheus provides a managed service that collects metrics from Kubernetes clusters and store them in Azure Monitor Metrics.

REST API - Azure Monitor provides REST APIs that allow you to get data in and out of Azure Monitor Metrics.

Security - All communication between connected systems and the Azure Monitor service is encrypted using the TLS 1.2 (HTTPS) protocol

Metrics Explorer

Use Metrics Explorer to interactively analyze the data in your metric database and chart the values of multiple metrics over time. You can pin the charts to a dashboard to view them with other visualizations. You can also retrieve metrics by using the Azure monitoring REST API.



Best practices for autoscale

Autoscale concepts

- A resource can have only one autoscale setting.
- An autoscale setting can have one or more profiles, and each profile can have one or more autoscale rules.
- An autoscale setting scales instances horizontally, which is out by increasing the instances and in by decreasing the number of instances.
- An autoscale setting has a maximum, minimum, and default value of instances.
- An autoscale job always reads the associated metric to scale by, checking if it has crossed the configured threshold for scale-out or scale-in. You can view a list of metrics that autoscale can scale by at Azure Monitor autoscaling common metrics.
- All thresholds are calculated at an instance level. An example is "scale out by one instance when average CPU > 80% when instance count is 2." It means scale-out when the average CPU across all instances is greater than 80%.
- All autoscale failures are logged to the activity log. You can then configure an activity log alert so that you can be notified via email, SMS, or webhooks whenever there's an autoscale failure.
- Similarly, all successful scale actions are posted to the activity log. You can then configure an activity log alert so that you can be notified via email, SMS, or webhooks whenever there's a successful autoscale action. You can also configure email or webhook notifications to get notified for successful scale actions via the notifications tab on the autoscale setting.

Example 1.

For example, we don't recommend the following rule combination because there's no scale-in rule for memory usage:

- If CPU > 90%, scale out by 1
- If Memory > 90%, scale out by 1
- If CPU < 45%, scale in by 1

In this example, you can have a situation in which the memory usage is over 90% but the CPU usage is under 45%. This scenario can lead to flapping as long as both conditions are met.

Example 2.

Let's illustrate it with an example to ensure you understand the behavior better:

- Increase instances by 1 count when Storage queue message count >= 50
- Decrease instances by 1 count when Storage queue message count <= 10

Consider the following sequence:

1. There are two Storage queue instances.
2. Messages keep coming and when you review the Storage queue, the total count reads 50. You might assume that autoscale should start a scale-out action. However, notice that it's still $50/2 = 25$ messages per instance. So, scale-out doesn't occur. For the first scale-out action to happen, the total message count in the Storage queue should be 100.
3. Next, assume that the total message count reaches 100.
4. A third Storage queue instance is added because of a scale-out action. The next scale-out action won't happen until the total message count in the queue reaches 150 because $150/3 = 50$.
5. Now the number of messages in the queue gets smaller. With three instances, the first scale-in action happens when the total messages in all queues add up to 30 because $30/3 = 10$ messages per instance, which is the scale-in threshold.

There are cases where you might have to set multiple rules in a profile. The following autoscale rules are used by the autoscale engine when multiple rules are set:

- On *scale-out*, autoscale runs if any rule is met.
- On *scale-in*, autoscale requires all rules to be met.

To illustrate, assume that you have four autoscale rules:

- If CPU < 30%, scale in by 1
- If Memory < 50%, scale in by 1
- If CPU > 75%, scale out by 1
- If Memory > 75%, scale out by 1

Then the following action occurs:

- If CPU is 76% and Memory is 50%, we scale out.
- If CPU is 50% and Memory is 76%, we scale out.

On the other hand, if CPU is 25% and Memory is 51%, autoscale *doesn't* scale in. To scale in, CPU must be 29% and Memory 49%.

az monitor metrics alert

Commands

Expand table

Name	Description	Type	Status
az monitor metrics alert condition	Manage near-realtime metric alert rule conditions.	Core	GA

az monitor metrics alert condition create	Build a metric alert rule condition.	Core	Preview
az monitor metrics alert create	Create a metric-based alert rule.	Core	GA
az monitor metrics alert delete	Delete a metrics-based alert rule.	Core	GA
az monitor metrics alert dimension	Manage near-realtime metric alert rule dimensions.	Core	GA
az monitor metrics alert dimension create	Build a metric alert rule dimension.	Core	Preview
az monitor metrics alert list	List metric-based alert rules.	Core	GA
az monitor metrics alert show	Show a metrics-based alert rule.	Core	GA
az monitor metrics alert update	Update a metric-based alert rule.	Core	GA

Create a high CPU usage alert on a VM with email and webhook actions.

```
az monitor metrics alert create -n alert1 -g {ResourceGroup} --scopes {VirtualMachineID} --condition "avg Percentage CPU > 90" --window-size 5m --evaluation-frequency 1m --action "/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Microsoft.Insights/actionGroups/<actionGroupName>" apiKey={APIKey} type=HighCPU --description "High CPU"
```

Create an alert when a storage account shows a high number of slow transactions, using multi-dimensional filters.

```
az monitor metrics alert create -n alert1 -g {ResourceGroup} --scopes {VirtualMachineID} --condition "avg Percentage CPU > 90" --window-size 5m --evaluation-frequency 1m --action "/subscriptions/<subscriptionId>/resourceGroups/<resourceGroupName>/providers/Microsoft.Insights/actionGroups/<actionGroupName>" apiKey={APIKey} type=HighCPU --description "High CPU"
```

Required Parameters

--condition

The condition which triggers the rule. It can be created by 'az monitor metrics alert condition create' command.

Usage: --condition {avg,min,max,total,count} [NAMESPACE.]METRIC [{=,!=,>,>=,<,<=}] THRESHOLD [{<,>,>=}] dynamic SENSITIVITY VIOLATIONS of EVALUATIONS [since DATETIME]] [where DIMENSION

{includes,excludes} VALUE [or VALUE ...] [and DIMENSION {includes,excludes} VALUE [or VALUE ...] ...]]
[with skipmetricvalidation]

Sensitivity can be 'low', 'medium', 'high'.

Violations can be the number of violations to trigger an alert. It should be smaller or equal to evaluation.

Evaluations can be the number of evaluation periods for dynamic threshold.

Datetime can be the date from which to start learning the metric historical data and calculate the dynamic thresholds (in ISO8601 format).

Dimensions can be queried by adding the 'where' keyword and multiple dimensions can be queried by combining them with the 'and' keyword.

Values for METRIC, DIMENSION and appropriate THRESHOLD values can be obtained from az monitor metrics list-definitions command.

Due to server limitation, when an alert rule contains multiple criterias, the use of dimensions is limited to one value per dimension within each criterion.

Multiple conditions can be specified by using more than one --condition argument.

--name -n

Name of the alert rule.

--resource-group -g

Name of resource group. You can configure the default group using az configure --defaults group=<name>.

--scopes

Space-separated list of scopes the rule applies to. The resources specified in this parameter must be of the same type and exist in the same location.

Optional Parameters

--action -a

Add an action group and optional webhook properties to fire when the alert is triggered.

Usage: --action ACTION_GROUP_NAME_OR_ID [KEY=VAL [KEY=VAL ...]]

Multiple action groups can be specified by using more than one --action argument.

--auto-mitigate

Automatically resolve the alert.

Accepted values: false, true

--description

Free-text description of the rule.

--disabled

Create the rule in a disabled state.

Accepted values: false, true

Default value: False

--evaluation-frequency

Frequency with which to evaluate the rule in "###h###m###s" format.

Default value: 1m

--region --target-resource-region

The region of the target resource(s) in scopes. This must be provided when scopes is resource group or subscription.

--severity

Severity of the alert from 0 (critical) to 4 (verbose).

Default value: 2

--tags

Space-separated tags: key[=value] [key[=value] ...]. Use "" to clear existing tags.

--target-resource-type --type

The resource type of the target resource(s) in scopes. This must be provided when scopes is resource group or subscription.

--window-size

Time over which to aggregate metrics in "###h###m###s" format.

Default value: 5m

Track custom operations with Application Insights .NET SDK

Why do I need Application Insights?

Application Insights monitors your running web app. It tells you about failures and performance issues and helps you analyze how customers use your app.

It works for apps running on platforms like ASP.NET, Java EE, and Node.js. It's hosted in the cloud or on-premises.



Modern web applications are developed in a cycle of continuous delivery:

- Release a new feature or improvement.
- Observe how well it works for users.
- Plan the next increment of development based on that knowledge.

The most important aspect of this process is diagnostics and diagnosis. If the application fails, business is lost. The prime role of a monitoring framework is to:

- Detect failures reliably.
- Notify you immediately.
- Present you with the information needed to diagnose the problem.

1. Application Insights instruments your app and sends telemetry about it while the app is running. Either you can build the Application Insights SDK into the app or you can apply instrumentation at runtime. The former method is more flexible because you can add your own telemetry to the regular modules.
2. The telemetry is sent to the Application Insights portal, where it's stored and processed. Although Application Insights is hosted in Azure, it can monitor any web apps, not just Azure apps.
3. The *telemetry is presented to you in the form of charts and tables of events*.

There are two main types of telemetry: aggregated and raw instances.

- Instance data might include a report of a request that's been received by your web app. You can find and inspect the details of a request by using the Search tool in the Application Insights portal. The instance might include data like how long your app took to respond to the request and the requested URL and the approximate location of the client.
- Aggregated data includes counts of events per unit time so that you can compare the rate of requests with the response times. It also includes averages of metrics like request response times.

The main categories of data are:

- Requests to your app (usually HTTP requests) with data on URL, response time, and success or failure.
- Dependencies like REST and SQL calls made by your app, also with URI, response times, and success.
- Exceptions, including stack traces.
- Page view data, which comes from users' browsers.
- Metrics like performance counters and metrics you write yourself.
- Custom events that you can use to track business events.
- Log traces used for debugging.

1.1 Features

1.1.1 Availability

Multistep web tests

You can monitor a recorded sequence of URLs and interactions with a website via multistep web tests.

Recommended using TrackAvailability to submit custom availability tests instead of multistep web tests. This option is the long-term supported solution for multi-request or authentication test scenarios. With TrackAvailability() and custom availability tests, you can run tests on any compute you want and use C# to easily author new tests.

Monitor availability with URL ping tests

The name *URL ping test* is a bit of a misnomer. These tests don't use the Internet Control Message Protocol (ICMP) to check your site's availability.

Instead, they use more advanced HTTP request functionality to validate whether an endpoint is responding.

They measure the performance associated with that response. They also add the ability to set custom success criteria, coupled with more advanced features like parsing dependent requests and allowing for retries.

Monitor availability with URL ping tests

The name *URL ping test* is a bit of a misnomer. These tests don't use the Internet Control Message Protocol (ICMP) to check your site's availability.

Instead, they use more advanced HTTP request functionality to validate whether an endpoint is responding. They measure the performance associated with that response. They also add the ability to set custom success criteria, coupled with more advanced features like parsing dependent requests and allowing for retries.

Setting	Description
URL	The URL can be any webpage that you want to test, but it must be visible from the public internet. The URL can include a query string. For example, you can exercise your database a little. If the URL resolves to a redirect, you can follow it up to 10 redirects.
Parse dependent requests	The test requests images, scripts, style files, and other files that are part of the webpage under test. The recorded response time includes the time taken to get these files. The test fails if any of these resources can't be successfully downloaded within the timeout for the whole test. If the option isn't enabled, the test only requests the file at the URL that you specified. Enabling this option results in a stricter check. The test might fail for cases that aren't noticeable from manually browsing through the site.
Enable retries	When the test fails, it's retried after a short interval. A failure is reported only if three successive attempts fail. Subsequent tests are then performed at the usual test frequency. Retry is temporarily suspended until the next success. This rule is applied independently at each test location. <i>We recommend this option.</i> On average, about 80 percent of failures disappear on retry.
Test frequency	This setting determines how often the test is run from each test location. With a default frequency of five minutes and five test locations, your site is tested every minute on average.
Test locations	The values for this setting are the places from which servers send web requests to your URL. <i>We recommend a minimum of 5 test locations</i> to ensure that you can distinguish problems in your website from network issues. You can select up to 16 locations.

1.1.2 Sampling in Application Insights

Sampling is a feature in Application Insights.

It's a recommended way to reduce telemetry traffic, data costs, and storage costs, while preserving a statistically correct analysis of application data.

Sampling also helps you avoid Application Insights throttling your telemetry. The sampling filter selects items that are related, so that you can navigate between items when you're doing diagnostic investigations.

Home > Fabrikamprod - Usage and estimated costs

Fabrikamprod - Usage and estimated costs

Application Insights | Directory: Microsoft

Search (Ctrl+ /)

- Properties
- Smart Detection settings
- Usage and estimated costs**
- Continuous export

Data sampling | Data retention

The table below shows estimated monthly this Application Insights resource based on month's usage.

Application Insights

Monthly

Data sampling

Select the fraction of data received from your app that the Application Insights service will retain using ingestion sampling:

All data (100%)

There may be additional sampling also happening by the SDK before data is sent to the Application Insights servers. If your SDK has adaptive sampling enabled or you have manually added code to enable sampling, no ingestion sampling will be applied regardless of the setting above. [Learn more](#)

OK

Collect and analyze resource logs from an Azure resource

- Create a Log Analytics workspace in Azure Monitor.
- Create a diagnostic setting to collect resource logs.
- Create a simple log query to analyze logs.

Home > Log Analytics workspaces >

Create Log Analytics workspace

Basics | Tags | Review & Create

Basics

A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

Project details

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

Subscription *

Resource group * [Create new](#)

Instance details

Name *

Region *

[Review & Create](#) [Previous](#) [Next: Pricing tier >](#)

Home > Storage accounts > my-storage-account >

Diagnostic setting

[Save](#) [Discard](#) [Delete](#) [Feedback](#)

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

Diagnostic setting name *

Logs

Category groups

☐ audit ☐ allLogs

Categories

☒ StorageRead

☒ StorageWrite

☒ StorageDelete

Metrics

☒ Transaction

Destination details

☒ Send to Log Analytics workspace

Subscription

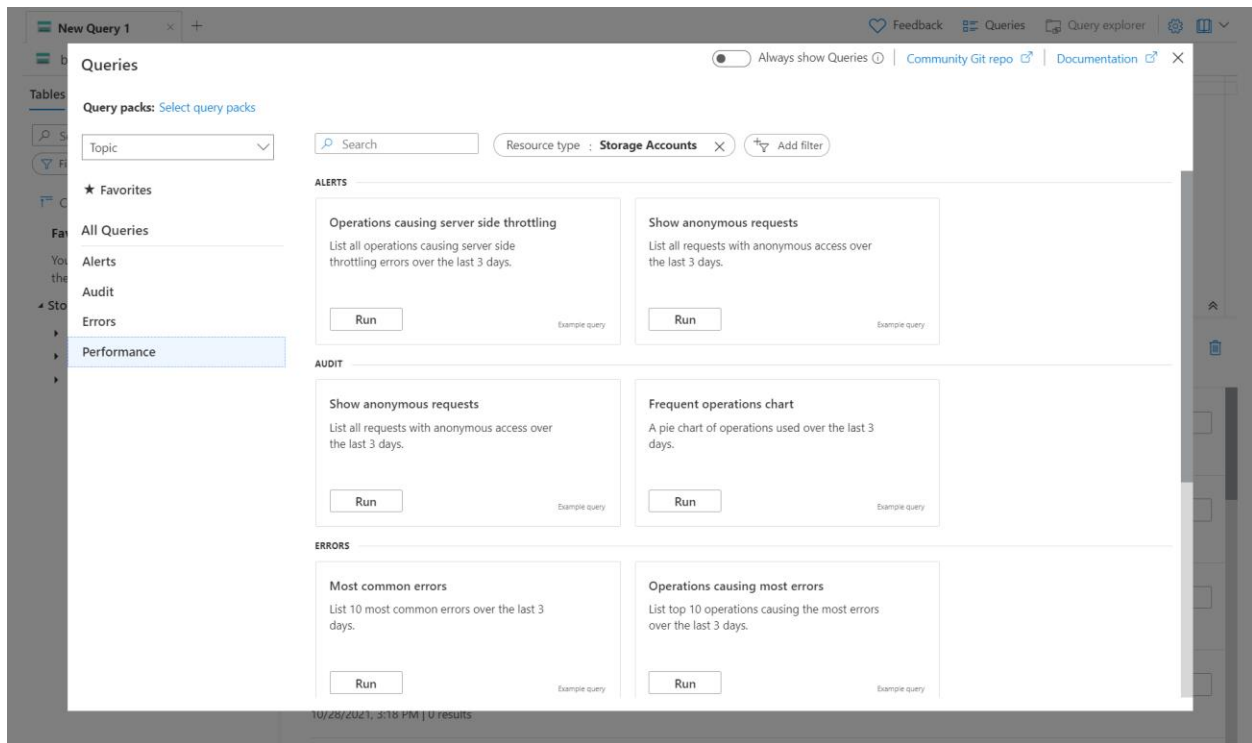
Log Analytics workspace

☐ Archive to a storage account

☐ Stream to an event hub

☐ Send to partner solution

Data is retrieved from a Log Analytics workspace by using a log query written in Kusto Query Language (KQL). A set of pre-created queries is available for many Azure services, so you don't require knowledge of KQL to get started.



1.1.2 Usage analysis with Application Insights

Application Insights is a powerful tool for monitoring the performance and usage of your applications. It provides insights into how users interact with your app, identifies areas for improvement, and helps you understand the impact of changes. With this knowledge, you can make data-driven decisions about your next development cycles.

This article covers the following areas:

- Users, Sessions & Events - Track and analyze user interaction with your application, session trends, and specific events to gain insights into user behavior and app performance.
- Funnels - Understand how users progress through a series of steps in your application and where they might be dropping off.

- User Flows - Visualize user paths to identify the most common routes and pinpointing areas where users are most engaged users or may encounter issues.
- Cohorts - Group users or events by common characteristics to analyze behavior patterns, feature usage, and the impact of changes over time.
- Impact Analysis - Analyze how application performance metrics, like load times, influence user experience and behavior, to help you to prioritize improvements.
- HEART - Utilize the HEART framework to measure and understand user Happiness, Engagement, Adoption, Retention, and Task success.

1.1.3 Usage analysis with Application Insights

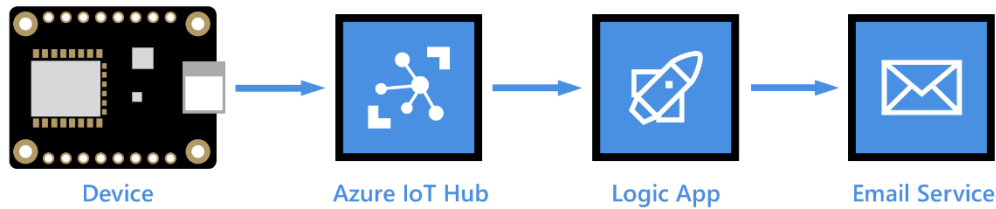
Three of the Usage panes use the same tool to slice and dice telemetry from your web app from three perspectives. By filtering and splitting the data, you can uncover insights about the relative use of different pages and features.

- Users tool: How many people used your app and its features? Users are counted by using anonymous IDs stored in browser cookies. A single person using different browsers or machines will be counted as more than one user.
- Sessions tool: How many sessions of user activity have included certain pages and features of your app? A session is reset after half an hour of user inactivity, or after 24 hours of continuous use.
- Events tool: How often are certain pages and features of your app used? A page view is counted when a browser loads a page from your app, provided you've instrumented it.

A custom event represents one occurrence of something happening in your app. It's often a user interaction like a button selection or the completion of a task. You insert code in your app to generate custom events or use the Click Analytics extension.

2.1 Monitor IoT devices and send notifications with Azure Logic Apps

Use Azure Logic Apps to monitor incoming device telemetry from IoT Hub and send notifications when alerts are triggered.







Azure Logic Apps can help you orchestrate workflows across on-premises and cloud services, multiple enterprises, and various protocols. A logic app begins with a trigger, which is then followed by one or more actions that can be sequenced using built-in controls, such as conditions and iterators.




Create a Service Bus queue. Create a route in your IoT hub that sends messages to the Service Bus queue if the messages contain anomalous temperature readings. Create a logic app that watches for messaging arriving in the queue and sends an email alert.






Create Service Bus namespace and queue


[Home](#) > [myNamespace](#) | [Overview](#) >


 **myNamespace**   


Service Bus Namespace


  


 Queue  Topic  Refresh  Delete  Give feedback


 Overview


 Activity log


 Access control (IAM)


 Tags


 Diagnose and solve problems

 Settings

 Entities

 Monitoring

 Automation

 Essentials

Resource group [\(move\)](#)

Status

Location

Subscription [\(move\)](#)

Subscription ID

Created

Updated

Pricing tier

Host name

: [myResourceGroup](#)

: Active

: West US 2

: [mySubscription](#)

: 00000000-0000-0000-0000-00000000...

: Friday, February 2, 2024

: Friday, February 2, 2024

: [Basic](#)

: mynamespace.servicebus.windows.net

[View Cost](#)

[JSON View](#)

Add a Service Bus queue to the namespace

Add a custom endpoint and routing rule to your IoT hub

Add a route ...



myHub

- 1 Endpoint** ② Route ③ Enrichment

Create an endpoint for your route—this will determine which Azure services will receive your messages. You can have a maximum of 10 endpoints for each IoT Hub. [Learn more](#)

Endpoint type ⓘ

Service bus queue



Endpoint name *

logic-apps-tutorial-endpoint

[Select existing](#)

Service bus namespaces *

mynamespace



Service bus queue *

myqueue



Authentication type *

Choose the authentication type for this routing endpoint. [Learn more](#)

- ☒ Key-based
☐ System-assigned
☐ User-assigned

Create + next

Configure Logic Apps for notifications

Create Logic App



Basics Tags Review + create

Create a logic app, which lets you group workflows as a logical unit for easier management, deployment and sharing of resources. Workflows let you connect your business-critical apps and services with Azure Logic Apps, automating your workflows without writing a single line of code.

Project Details

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

Subscription * ⓘ

mySubscription

Resource Group * ⓘ

myResourceGroup

[Create new](#)

Instance Details

Logic App name *

logic-apps-tutorial

Region *

East US

Enable log analytics *

☐ Yes ☒ No

Plan

The plan type you choose dictates how your app scales, what features are enabled, and how it is priced. [Learn more](#)

Plan type *

- ☐ **Standard:** Best for enterprise-level, serverless applications, with event-based scaling and networking isolation.
- ☒ **Consumption:** Best for entry-level. Pay only as much as your workflow runs.

📘 Looking for the classic consumption create experience? [Click here](#)

[Review + create](#)

[< Previous](#)

[Next : Tags >](#)

Create a logic app

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Development Tools
 - Logic app designer
 - Logic app code view
 - Run History
 - Versions
 - API connections
 - Quick start guides

- Settings
- Monitoring

Templates

Choose a template below to create your Logic App.

Category : All Sort by : Popularity

Blank Logic App

Azure Monitor - Metrics Alert Handler

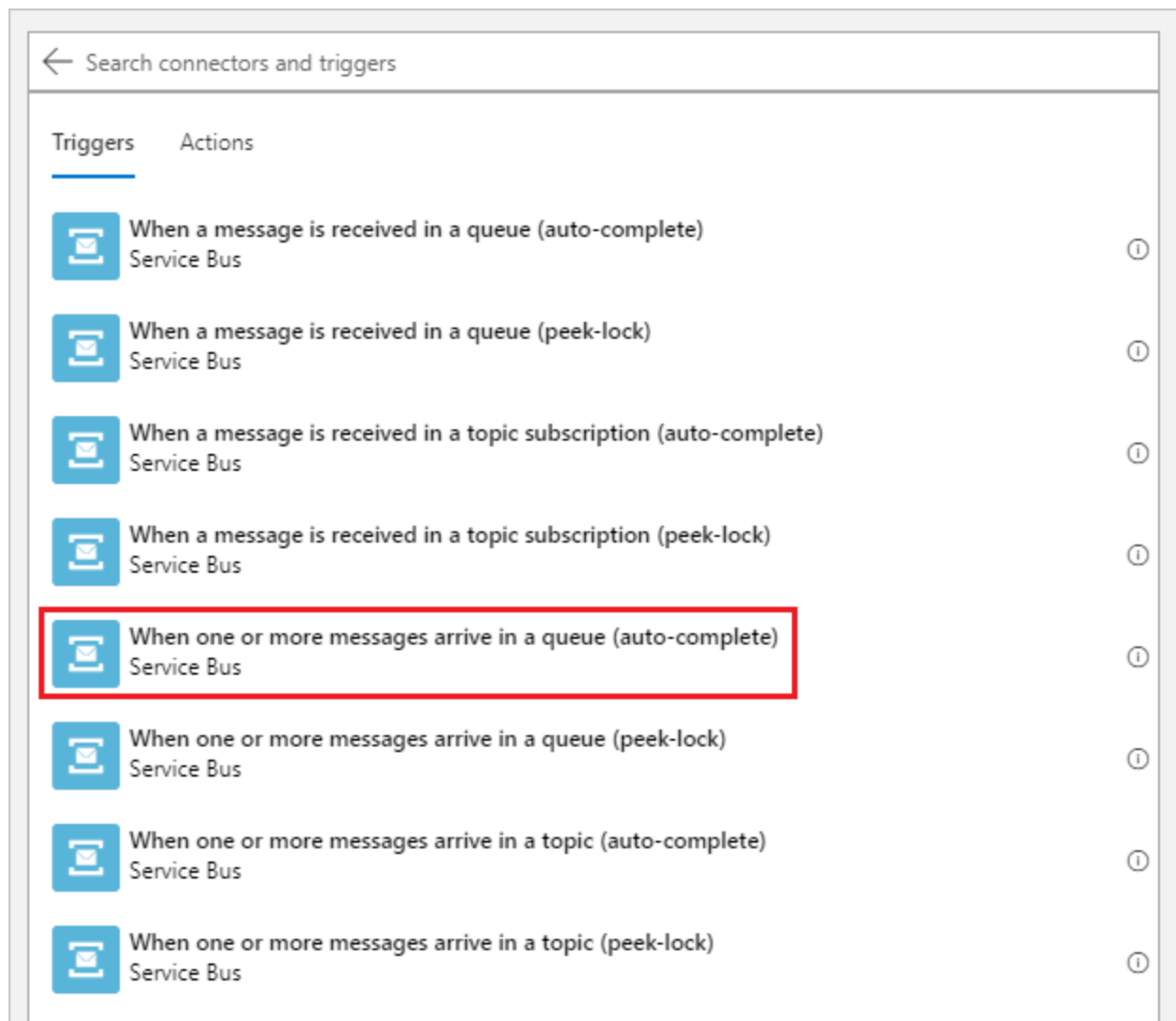
Auto tier Azure blobs based on the last modified time.

Delete old Azure blobs


HTTP Request-Response

Peek-lock receive a


Configure the logic app trigger




Configure the logic app action

 When one or more messages arrive in a queue (auto-complete) ...


* Queue name

myqueue 


Maximum message count


175 


Queue type

Main 


How often do you want to check for items?


3 


Minute 


Add new parameter 

Connected to tutorial-sb-connection. [Change connection.](#)

 When one or more messages arrive in a queue (auto-complete) ...



 Choose an operation ×

 smtp

For You


All


Built-in


Standard

Enterprise

Custom

 IA-Connect JML

 Microsoft Bookings

 SMTP

Redeploy Windows virtual machine to new Azure node

If you have been facing difficulties troubleshooting Remote Desktop (RDP) connection or application access to Windows-based Azure virtual machine (VM), redeploying the VM may help.

When you redeploy a VM, Azure will shut down the VM, move the VM to a new node within the Azure infrastructure, and then power it back on, retaining all your configuration options and associated resources.

The screenshot shows the 'Redeploy + reapply' window in the Azure portal for a virtual machine named 'Thomas'. The window has a sidebar on the left with a search bar and a list of actions: 'Export template', 'Help', 'Resource health', 'Boot diagnostics', 'Performance diagnostics', 'VM Inspector (Preview)', 'Reset password', 'Redeploy + reapply' (which is highlighted), 'Serial console', 'Connection troubleshoot', 'Learning center', and 'Support + Troubleshooting'. The main content area is titled 'Redeploy + reapply' and contains two sections. The first section, 'Can't connect to your virtual machine?', includes a 'Redeploy' button highlighted with a red rectangle. The second section, 'Virtual machine in a failed state?', includes a 'Reapply' button. Both sections provide brief explanations of the actions and links to learn more.

Thomas
Virtual machine

Search

Export template

Help

Resource health

Boot diagnostics

Performance diagnostics

VM Inspector (Preview)

Reset password

Redeploy + reapply

Serial console

Connection troubleshoot

Learning center

Support + Troubleshooting

Redeploy + reapply

Can't connect to your virtual machine?

Redeploy
Try redeploying your virtual machine, which will migrate it to a new Azure host. If you continue, the virtual machine will be restarted and you will lose any data on the temporary drive. While the redeployment is in progress, the virtual machine will be unavailable. [Learn more about Redeploy](#)

Redeploy

Virtual machine in a failed state?

Reapply
Try reapplying your virtual machine's state. This operation will rerun VM provisioning and help solve the VM failed state, in case when VM provisioning failed while executing a previous VM action. [Learn more about Reapply](#)

Reapply