# AZ-104T00A – Administer Monitoring

Good day everyone.

## Configure Azure Monitor

Azure Monitor maximizes the availability and performance of your applications by delivering a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. It helps you understand how your applications are performing and proactively identifies issues affecting them and the resources on which they depend. The Azure Monitor landing page provides a jumping off point to configure other more specific monitoring services, such as Application Insights, Network Watcher, Log Analytics, Management Solutions, and so on. Azure Monitor helps you track performance, maintain security, and identify trends by ingesting metrics and telemetry from multiple areas, including applications and the operating systems of virtual machines. It also allows you to query your Azure resources (which emit performance counters), your Azure subscriptions, Azure AD tenant, and event custom sources. The data from your Azure resources is ingested into either metrics stored within the Azure platform and accessible by the monitor service or as logs into Log Analytics. So, Azure Monitor can ingest many different data sources. Sources include application code, operating system, resource, subscription, and tenant data. You can even create your own custom data source. Data sources generally fall into two categories metrics and logs. Metrics are numerical values that describe some aspect of a system at a point in time. For example, virtual machine CPU performance. Logs contain data organized into records with different sets of properties for each type. For example, the activity log shows subscription-level events. This includes such information as when a resource is modified or when a virtual machine is started.

Configure Azure Monitor Introduction

Azure Monitor provides features and capabilities in three areas:

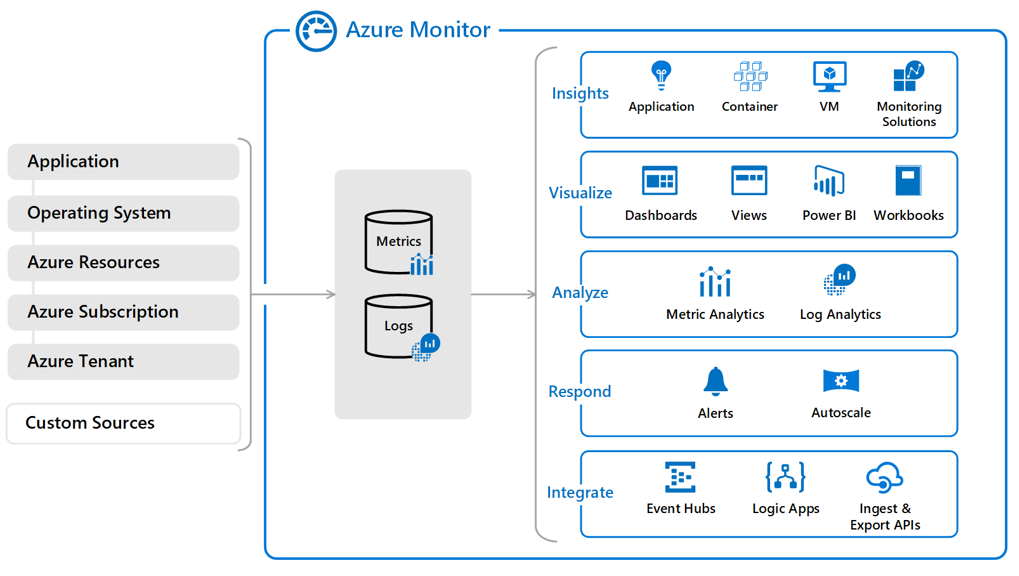
**Monitor and visualize metrics**: Azure Monitor gathers numerical metric values from your Azure resources according to your preferences. Azure Monitor offers different methods for viewing your metric data to help you understand the health, operation, and performance of your system.

**Query and analyze logs**: Azure Monitor Logs (Log Analytics) generates activity logs, diagnostic logs, and telemetry information from your monitoring solutions. The service provides analytics queries that you can use to help with troubleshooting and visualizations of your log data.

**Set up alerts and actions**: Azure Monitor lets you set up alerts for your gathered data to notify you when critical conditions arise. You can configure actions based on the alert conditions, and take automated corrective steps based on triggers from your metrics or logs.

Understand Azure Monitor Components

The following diagram provides a high-level view of how Azure and Azure Monitor work together to provide you with a robust monitoring and diagnostics solution. Azure Monitor can ingest many different data sources. Sources include application code, operating system, resource, subscription, and tenant data. You can even create your own custom data source.

The monitoring and diagnostic services offered in Azure are divided into broad **categories** such as Core, Application, Infrastructure, and Shared Capabilities.

**Data stores** in Azure Monitor hold your metrics and logs. [Azure Monitor Metrics](https://learn.microsoft.com/en-us/azure/azure-monitor/essentials/data-platform-metrics) and [Azure Monitor Logs](https://learn.microsoft.com/en-us/azure/azure-monitor/logs/data-platform-logs) are the two base types of data used by the service.

Various **monitoring sources** provide Azure Monitor with the metrics and logs data to analyze. These sources can include your Azure subscription and tenant, your Azure service instances, your Azure resources, data from your applications, and more.

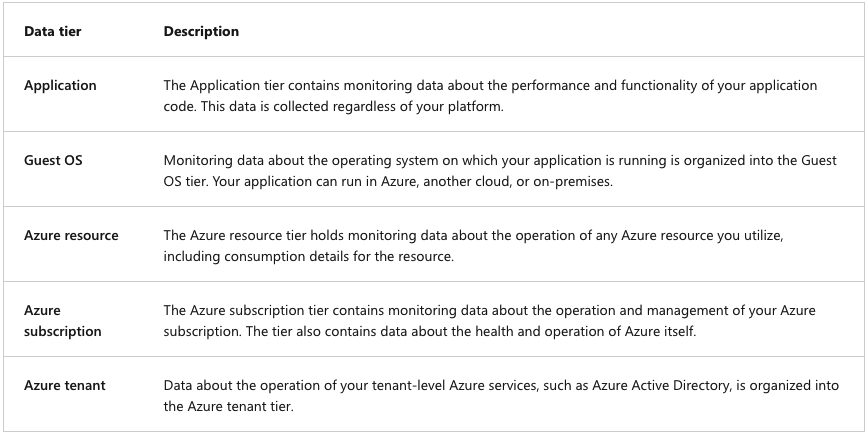
[Azure Monitor Insights](https://learn.microsoft.com/en-us/azure/azure-monitor/insights/insights-overview) performs different functions with the collected data, including analysis, alerting, and streaming to external systems. So, we're talking about applications that you're running in Azure, the host operating system in your virtual machines, Azure resources at the subscription level, and right up to the Azure Active Directory tenant. For that matter, we can capture metric and log data from Azure Active Directory. Now the bottom part of this highlight says custom sources, so you should understand that the Azure Monitor services are available not only for native Azure resources, but also your own resources. This is your application source code but also, just as importantly, related workload resources that maybe running in another cloud like Amazon Web Services, Google Cloud, and/or your local data centers. It's all possible to surface that in Azure Monitor. It's pretty cool. And as you can see on the right side of this diagram, there's several different ways that you can look at that metrics and log data. There is a whole platform called Insights, where Microsoft chooses what it thinks are the highest impact views and presents those to you as a way to give you a head start on your performance, monitoring, and troubleshooting. You can visualize the data directly with or without Power BI. Power BI is part of the Microsoft 365 platform, allows you to populate these live dashboards. But in Azure Monitor natively, we now have these interactive workbooks where you can surface data. For analysis, we've got the Log Analytics workspace that we'll talk about later. For response, you can configure alerts and action groups. Again, we'll get to that in a moment. And there's also autoscale capability. For virtual machines there's virtual machine scale sets. For Azure App Service there's these App Service plan and dynamic autoscale. And lastly, we can look at an occurrence of a metrics reading or a log entry as an event, and you can use any event in Azure to trip off workflows. So, you can stitch together Logic Apps, Function Apps, APIs that you're hosting in the Azure API management service, and create a full end‑to‑end lifecycle to respond to these diagnostics performance events.

Define Metrics and Logs

I want to stress you to understand that there's two types of diagnostics data available in Microsoft Azure for Azure native resources, metrics and logs. Technically, there's a third type that we're going to hit by looking at Application Insights and basically is code streams coming from your application itself. But in Azure Monitor proper, we're talking about metric and logs. Now metrics refers to time‑sampled numeric data. Think of an Azure virtual machine and you're wanting to keep track of central processing unit, or CPU, count. Or perhaps you're looking at a storage account and you're looking at data ingress and egress, or you're looking at an application code and you're looking at the number of page views per day. Metric data. Now we can view the metrics as a live stream. In many cases, those metrics are persisted in logs. A log is simply a text file of some format or another, could be plain text, but more likely, it's comma‑separated value, tab‑separated value. Maybe Azure is using the Table service, which is key‑value storage, to persist metric data, but other events can also be persisted in these logs.

Identify Data Types

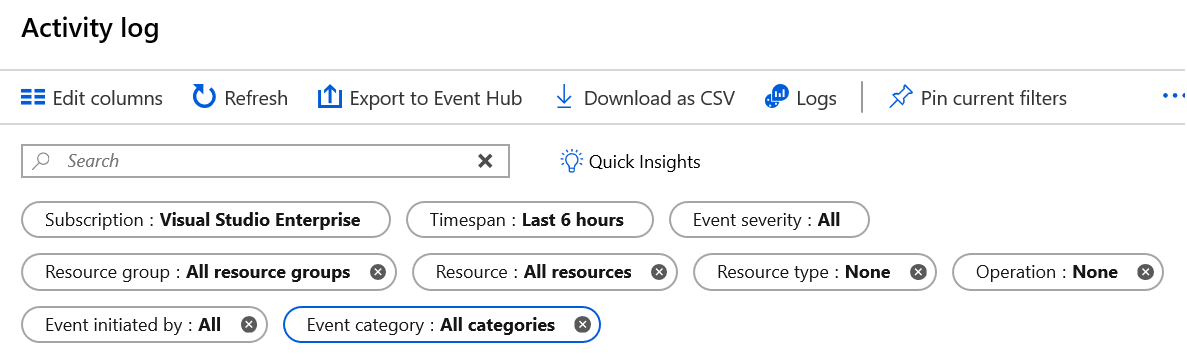
The following table summarizes the tiers of monitoring data that are collected by Azure Monitor.

Describe Activity Log Events

The Azure Monitor activity log is a subscription log that provides insight into subscription-level events that occur in Azure. Events can include a range of data from Azure Resource Manager operational data to updates on Azure service health events. You can use the information in activity logs to understand the status of resource operations and other relevant properties. Activity logs can help you can determine the "what, who, and when" for any write operation (PUT, POST, DELETE) performed on resources in your subscription. Activity logs are kept for 90 days. You can query for any range of dates in an activity log, as long as the starting date isn't more than 90 days in the past. You can retrieve events from your activity logs by using the Azure portal, the Azure CLI, PowerShell cmdlets, and the Azure Monitor REST API.

Query the Activity Log

In the Azure portal, you can filter your Azure Monitor activity logs so you can view specific information. The filters enable you to review only the activity log data that meets your criteria. You might set filters to review monitoring data about critical events for your primary subscription and production virtual machine during peak business hours.

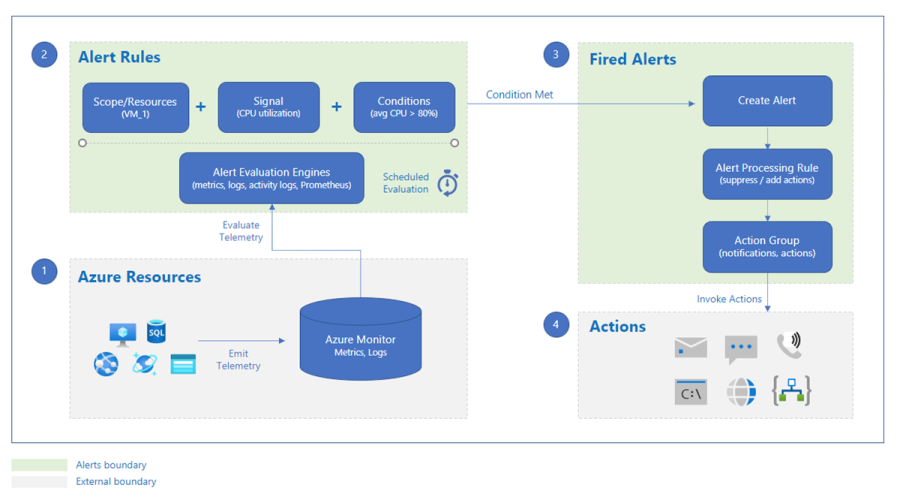


Let's review some of the filters you can set to control what data to review in your activity log:

* **Subscription**: Show the data for one or more specified Azure subscription names.
* **Timespan**: Show data for a specified time by choosing the start and end time for events, such as a six-hour period.
* **Event Severity**: Show events at the selected severity levels, including *Informational*, *Warning*, *Error*, or *Critical*.
* **Resource group**: Show data for one or more specified resource groups within your specified subscriptions.
* **Resource (name)**: Show data for the specified resources.
* **Resource type**: Show data for resources of a specified type, such as Microsoft.Compute/virtualmachines.
* **Operation name**: Show data for a selected Azure Resource Manager operation, such as Microsoft.SQL/servers/Write.
* **Event initiated by**: Show operation data for a specified user who performed the operation, referred to as the "caller."

After you define a set of filters, you can pin the filter set to the Azure Monitor dashboard. You can also download your activity log search results as a CSV file. In addition to the filters, you can enter a text string in the **Search** box. Azure Monitor tries to match your search string against data returned for all fields in all events that corresponds to your filter settings.

## Configure Azure Alerts

So, what we have in Azure Monitor is the capacity to specify a target resource or resources. We could watch an entire resource group, for example, that contains a number of resources and then define one or more criteria. These are logic tests that when they evaluate to true will fire the alert and trip off a separate resource you create called an action group. You might have one action group that performs simple notification, another one that not only takes care of notification, but also runs code in response to the alert firing. You'll find, of course, that the alert will continue to monitor the condition and retrigger the alert based upon how you've configured it. An alert consists of *alert rules* that combine the settings and conditions you want to monitor, including Resources to monitor, Signals or telemetry to gather from the resources, Conditions to match. An alert rule specifies *action groups* to fulfill responsive steps when an alert triggers, such as sending notifications.

Create Alert Rules

You can work with Azure Monitor alerts in the Azure portal. There are options for creating and managing your alert instances, alert rules and conditions, and alert response actions. There are different types of alerts to support various configuration and monitoring scenarios, such as metrics, logs, and events. You define alert rules for the different types of telemetry gathered from your monitored data sources. Trigger alerts according to conditions for your metrics and logs data, or by watching for activity log events. You can also implement alerts to help maintain the health of the underlying Azure platform and test your website availability. Here are some of the most common types of alerts:

**Metric alerts**: Evaluate metrics data from your resources at regular intervals. Gather metrics data from your platform, Azure Monitor logs converted to metrics, Azure Application Insights, and custom metrics. Metric alerts can apply multiple conditions and dynamic thresholds.

**Log alerts**: Use Log Analytics queries in the Azure portal to evaluate resource logs at a predefined frequency.

**Activity log events**: Implement alerts to trigger when a new activity log event occurs that meets your conditions. Resource Health alerts and Service Health alerts are two types of activity log alerts.

**Smart detection alerts**: Receive automatic warnings about potential performance issues and failure anomalies in your web apps by using smart detection on your Application Insights resources. Migrate smart detection on your Application Insights resources to create alert rules for the different smart detection modules.

In the Azure portal, each alert has a state indicator to identify where the alert and corresponding issue are in the monitoring process. There are three alert states: **New**: The issue is new (open) and not in review. **Acknowledged**: The issue is in review and work is in progress. **Closed**: The issue is complete. During the alert monitoring process, when the conditions of an alert rule match telemetry data for your specified resource, an alert triggers and invokes the specified action groups. The system sets the alert state to New. After the system sets an alert state to New, you can change the state to specify where the related issue is in the resolution process. When the issue for alert is in review, you can change the alert state to Acknowledged. After the issue for an alert clears, you can change the alert state to Closed. If an alert is in the Closed state, you can "reopen" the alert by changing the alert state to New or Acknowledged. The history of the alert stores all state changes.

You can configure most types of alerts as stateful or stateless, which corresponds to how many instances of the same alert can be active at the same time. **Stateless alerts** trigger each time your alert rule condition matches your data, even if the same alert already exists. You can configure log alerts and metric alerts as stateless. **Stateful alerts** trigger when your alert rule condition matches your data and the same alert doesn't exist. A stateful alert doesn't trigger any more actions until the current alert rule conditions clear. You can configure log alerts and metric alerts as stateful. Activity log alerts are always stateless.

An alert rule consists of several key attributes: the target resource, an alert signal, the rule criteria, the issue severity, and a name and description. Your **target resource** defines the scope and signals that are available for your alert operation. A target can be any Azure resource such as a virtual machine, an Azure storage account, or a Virtual Machine Scale Sets instance. A target can also be a Log Analytics workspace or an Azure Application Insights resource. For certain resources like Azure Virtual Machines, you can specify multiple resources as the target for your alert rule. The target resource for your alert emits a **signal** based on your selected resource type. The emitted signal can be Metric, Activity log, Application Insights, or Log. You define **criteria** for your alert rule that combines your signal with processing logic. The criteria apply to your target resource. An example criteria combination is \\* Percentage CPU &gt; 70%; Server Response Time &gt; 4 ms; and Result count of a log query &gt; 100. You can specify the **severity** level for your alert rule that corresponds to the issue pertaining to your alert. The severity can be in the range from 0 to 4. When an issue matches your rule conditions, the system invokes the **actions** for your alert rule. The actions are the responsive steps pertaining to the issue such as sending notifications. By default, the system sets a new alert rule to enabled. If you don't want an alert to trigger, set the alert rule to disabled. An alert can only trigger when the alert rule is in the enabled state.

Create Action Groups

An action group is a collection of notification preferences that you define as an Azure subscription owner. When Azure Monitor detects an issue in your telemetry data, your alert triggers and your alert rule actions initiate. Azure Monitor, Azure Service Health, and Azure Advisor use action groups to notify users when an alert triggers. Multiple alerts can use the same action group or different action groups depending on the user's requirements. Notifications specify how to notify users when your action group triggers. Actions specify how to invoke your defined actions when your action group triggers. You can configure notifications to operate by configuring email or similar communication settings. In the Azure portal, you can select the **Email Azure Resource Manager role** option to send email notifications to the members of your Azure subscription's role. The system sends email to Azure Active Directory (Azure AD) user members of the role only, and not to Azure AD groups or service principals. You can also select the **Email/SMS message/Push/Voice** option to specify any email, SMS, push, or voice actions. You provide each action with a unique name and details, and define the notifications or actions to perform. You might specify actions to send a voice call, an SMS message, or an email message. You can configure the action group to use an automated action via the **Action type** attribute.

**Automation runbook**: An automation runbook is the ability to define, build, orchestrate, manage, and report on workflows that support system and network operational processes. A runbook workflow can potentially interact with all types of infrastructure elements, such as applications, databases, and hardware.

**Azure Functions**: Azure Functions is a serverless compute service that lets you run event-triggered code without having to explicitly provision or manage infrastructure.

**ITSM**: The action can connect Azure and a supported IT Service Management (ITSM) product or service. This action requires an ITSM connection.

**Logic Apps**: Azure Logic Apps connects your business-critical apps and services by automating your workflows.

**Webhook**: A webhook is an HTTPS or HTTP endpoint that allows external applications to communicate with your system.

Demonstration Alerts

**Create an alert rule**

1. In Azure portal, click on **Monitor**. The Monitor blade consolidates all your monitoring settings and data in one view.

2. Click **Alerts** then click **+ New alert rule**. As most resource blades also have Alerts in their resource menu under Monitoring, you could create alerts from there as well.

**Explore alert targets**

1. Click **Select** under Target, to select a target resource that you want to alert on. Use **Subscription** and **Resource type** dropdowns to find the resource you want to monitor. You can also use the search bar to find your resources.

2. If the selected resource has metrics, you can create alerts on, Available signals on the bottom right will include metrics. You can view the full list of resource types supported for metric alerts in this article.

3. Click **Done** when you have made your selection.

**Explore alert conditions**

1. Once you have selected a target resource, click on **Add condition**.

2. You will observe a list of signals supported for the resource; select the metric you want to create an alert on.

3. Optionally, refine the metric by adjusting Period and Aggregation. If the metric has dimensions, the Dimensions table will be presented.

4. Observe a chart for the metric for the last 6 hours. Adjust the **Show history** drop-down.

5. Define the **Alert logic**. This will determine the logic which the metric alert rule will evaluate.

6. If you are using a static threshold, the metric chart can help determine what might be a reasonable threshold. If you are using a Dynamic Thresholds, the metric chart will display the calculated thresholds based on recent data.

7. Click **Done**.

8. Optionally, add another criteria if you want to monitor a complex alert rule.

**Explore alert details**

1. Fill in Alert details like **Alert Rule Name**, **Description** and **Severity**.

2. Add an action group to the alert either by selecting an existing action group or creating a new action group.

3. Click **Done** to save the metric alert rule.

## Configure Log Analytics

Log Analytics is the future of monitoring in Azure. It uses this concept of the workspace. You'll find that Log Analytics is also called Azure Monitor Logs, so whenever you see references to Log Analytics or Azure Monitor Logs, we're talking about the same thing. And the idea here is that you've got data sources. These could be an Azure local data center or other clouds sending their log data into the centralized workspace.

Determine Log Analytics Uses

Log Analytics in Azure Monitor offers query features and tools that help you answer virtually any question about your monitored configuration. Log Analytics supports the Kusto Query Language (KQL). You can create simple or complex queries with KQL, including: Search and sort by value, time, property state, and more, Join data from multiple tables, Aggregate large sets of data, Perform intricate operations with minimal code. When your Azure Monitor Logs contain sufficient collected data, and you understand how to construct the appropriate query, you can use Log Analytics to complete detailed analysis and problem solving.

Create a Workspace

When you capture logs and data in Azure Monitor, Azure stores the collected information in a Log Analytics workspace. Your Log Analytics workspace is the basic management environment for Azure Monitor Logs. To get started with Log Analytics in Azure Monitor, you need to create your workspace. Each workspace has a unique workspace ID and resource ID. After you create your workspace, you configure your data sources and solutions to store their data in your workspace. To create your Log Analytics workspace, configure the following parameters:

**Name**: Provide a name for your new Log Analytics workspace. The name for your workspace must be unique within your resource group.

**Subscription**: Specify the Azure Subscription to associate with your workspace.

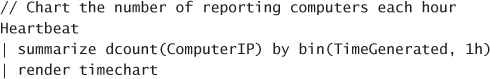
**Resource Group**: Specify the resource group to associate with your workspace. You can choose an existing resource group or create a new one. The resource group must contain at least one Azure Virtual Machines instance.

**Region**: Select the region where you deploy your virtual machines.

**Pricing**: The default pricing tier for a new workspace is pay-as-you-go. Charges incur only after you start collecting data.

Each Log Analytics workspace in Azure Monitor can have a different pricing tier. You can change the pricing tier for a workspace and also track the changes.

Query Log Analytics Data

The query language used by Log Analytics is called Kusto. Kusto queries are used to generate read-only requests to process data and return results. This means that the logs stored in Log Analytics are immutable and are only removed from a workspace based on the retention configuration. Queries are authored in plain-text, and the schema used by Log Analytics is like SQL’s with databases and tables composed of columns and rows. In each table, data is organized in columns with different data types as indicated by icons next to the column name. Column data types include text, numbers, and datetime. Authored queries in Log Analytics can take many forms, from basic queries to very advanced queries with multiple aggregates and summarizations. Queries can be used to search terms, identify trends, analyze patterns, and provide many other insights. Queries search tables and can start with either a table name or a search command that defines scope. The pipe (|) character separates commands, and you can add as many commands as required.

To run this query, browse to Azure Monitor and select Logs to open the query interface. This query will not return data if you do not have any virtual machines deployed and running. Those machines must also be associated with the Log Analytics Workspace you are querying. The query shown above is a table-based query. Queries always begin with a scope–either a table or search-based query. Kusto queries are case-sensitive. Typically, language keywords are written in lowercase. When using the names of tables and columns in queries, you must ensure you are using the correct case. Table-based queries target a single table in a Log Analytics Workspace (or database), while search-based queries target all tables by default. Table-based queries start by scoping the query, and therefore tend to be very efficient and generally faster than search queries. Search queries are less structured by nature, which makes them the better choice when searching for a specific value across columns or tables. In other words, a search can scan all columns in a given table or in all tables across an entire workspace for the defined value. The amount of data being processed by a query could be enormous, which is why these queries can take longer to complete and might return large result sets which are limited by the Log Analytics service to 10,000 results. To author queries in the Azure portal, browse to Azure Monitor and select the **Logs** blade. From this blade, you can access all the subscriptions and workspaces you have rights to read from.

Structure Log Analytics Queries

Administrators build Log Analytics queries from data stored in dedicated tables in a Log Analytics workspace. Some common dedicated tables include Event, Syslog, Heartbeat, and Alert. When you build a Kusto Query Language (KQL) query, you begin by determining which tables in the Azure Monitor Logs repository have the data you're looking for. The following illustration highlights how KQL queries use the dedicated table data for your monitored services and resources.