

## CHAPTER 2

# The Scenarios and the Tools

The Microsoft Azure platform supports enterprise-grade, hyperscale applications providing all the economies of scale, availability, scalability, and security. This makes it a strategic and preferred choice for any organization to host their applications and workload. Every organization intends to utilize a single system that can provide end-to-end monitoring of applications hosted both on-premises and on Azure. Second, some organizations look to continue to utilize their existing Security Information and Event Management (SIEM) systems and integrate the Azure services logs. Now, let us look at the details of the Azure monitoring and diagnostic platform and how it integrates with the various systems and services.

## Azure Monitoring Platform

Microsoft Azure provides a comprehensive monitoring platform and solution to monitor all infrastructure and platform resources. It helps by collecting, analyzing, monitoring, and reporting all logs and telemetry from various resources and helps to be more operationally effective, efficient, secured, and proactive. The Azure monitoring documentation contains a very beautiful representation of the service as an image here: <https://docs.microsoft.com/en-us/azure/azure-monitor/overview>.

## Basics

### Logs

Azure produces extensive logging for every service. These logs are categorized as the following:

- a. **Control/Management logs:** They give visibility into the Azure Resource Manager CREATE, UPDATE, and DELETE operations. These logs include Azure subscription-level and Tenant-level events and operations. For example, any create, deploy, delete resource operations, or any Azure Active Directory-level operational events.
- b. **Data Plane logs:** These give visibility into the events raised when using an Azure resource. For example, these could be Windows Event logs from an Azure virtual machine; security and application logs in a virtual machine; application-specific performance and functionality data; or any other Azure resource-specific data, for example, Network Security Group logs or Application Gateway diagnostic data, etc.

We can enable these Logs either by PowerShell, using Azure Diagnostics SDK and Visual Studio; from the Azure Portal Diagnostics/monitoring settings; or by using a JSON template incorporating the diagnostics extension. The resource-specific logs can be enabled either at the time when we deploy the resource or anytime later.

You can also ingest data into the Azure Monitor from a custom resource using Data Collector APIs. This addresses any custom scenarios wherein the resources do not have an inherent way to expose any telemetry. More documentation can be found here: <https://docs.microsoft.com/en-us/azure/azure-monitor/platform/data-collector-api>.

## Services and Resources

There are different services and resources that one can create on Azure. Hence, the collection methods and type of logs can be very different. Various Azure resources emit different logs; for example, VMs will have Event viewer logs and other performance metrics; the storage will have performance metrics and access logs, the Load Balancer and Network Security Group will emit other sets of logs. The collection method may be different in each case. We will discuss each type and its formats in our upcoming discussions.

## Storing Logs

We can either store these logs in an Azure Storage account (tables, blobs) or Event Hubs, etc. It may be Event logs forwarded to a collector system. Hence, the logs, data generated by Azure resources can be XML, CSV, Evtx, TXT, or JASON based. We need to make sure to store them effectively and so that they are easily accessible and interpretable by the different SIEM system if required.

## Exporting Logs

Azure has its own set of tools ([Log Integrator](#)) enabling these logs to be either streamed, exported into the SIEM system, or Convert them into a standard format, for example, JSON, which can be fed into the SIEM system. The preferred method is using a vendor-specific connector.

## Monitoring Logs

The Azure Monitor Views and dashboards, Log analytics views, and PowerBI are all used to monitor logs.

Azure monitoring is undergoing several changes, including consolidation and integration of various separate services into it.

For example, Azure log analytics and Azure Application Insights were part of a separate brand called Azure Operations Management Suite (OMS), and they are now integrated into Azure Monitor. Although the data analytics are still being done by Log Analytics, Azure Monitor now provides the single pane of glass for a seamless experience on insight, visualize, and analyze.

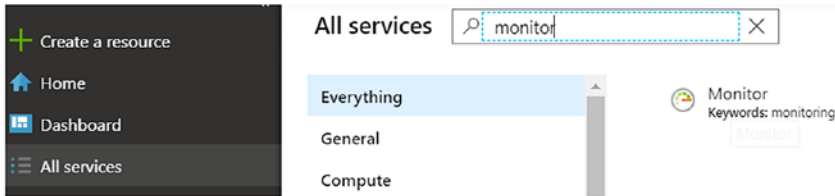
## Types of Data

Fundamentally there are two types of data that are in use in any monitoring system.

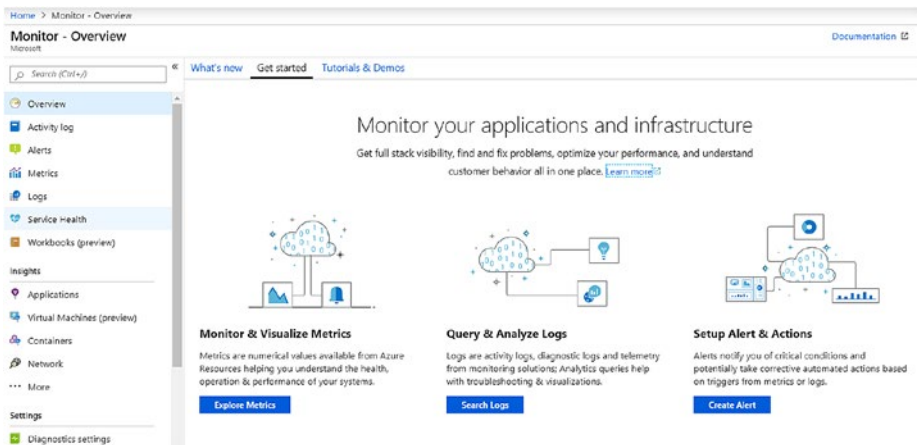
- **Logs:** Logs are a set of various data organized together to provide meaningful insights. It may contain additional properties or attributes: for example, Windows event logs that consist of various kinds of data related to an event. It contains time, user, details of the event, id that references the event and is well documented, etc. Then there are logs that might be specific to a service. For example, there are IIS logs, which contain service start, service stop, what is loaded, and what went wrong in the event of a failure of the service.
- **Metrics:** These are point-in-time information or numerical values about the system performance or other aspects of a service. For example, it could be CPU and Memory usage collected in a specific interval over a period of time. Depending on requirements, we can choose the sampling interval, and calculate and interpret values based on Max, Min, Avg, or Sum. Another interesting example would be Endpoints of an Azure Traffic Manager or Azure Application Gateway – number of endpoints available at any point in time.

## Azure Monitor: First Look

To access Azure Monitor, you can use the “All Services” menu and search for “Monitor” once logged onto the Azure portal as shown in Figure 2-1 and Figure 2-2.



**Figure 2-1.** Search for monitor in All services



**Figure 2-2.** Azure Monitor

In the Overview pane, you have the option to create and review various charts using **metrics**, review logs, and create alerts as required. Here is a simple example how a CPU metric looks like for a virtual machine named CorpDc01. You can access and navigate to the same view from the virtual machine as well (Figure 2-3).

## CHAPTER 2 THE SCENARIOS AND THE TOOLS

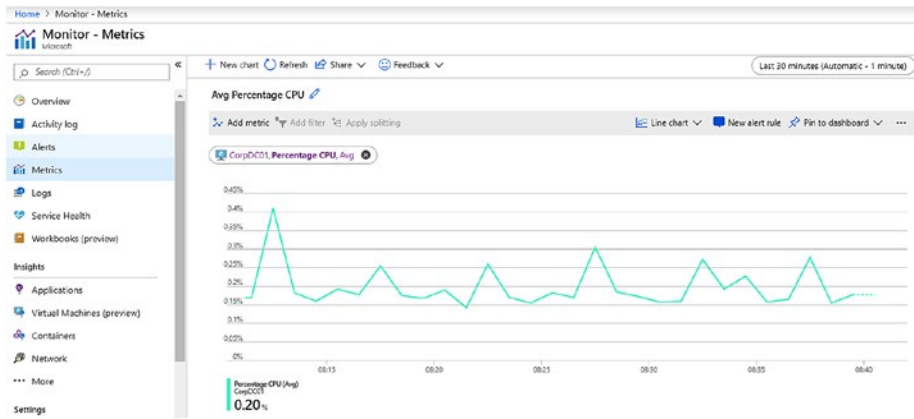


Figure 2-3. Metrics explorer

**Log data** is collected, analyzed, and queried using **Log Analytics** in the Azure portal. The query language used by Azure Monitor is **Kusto query language**. It can be used to perform advanced operations, including joins and aggregations, etc. (Figure 2-4). We will discuss this more in upcoming chapters.

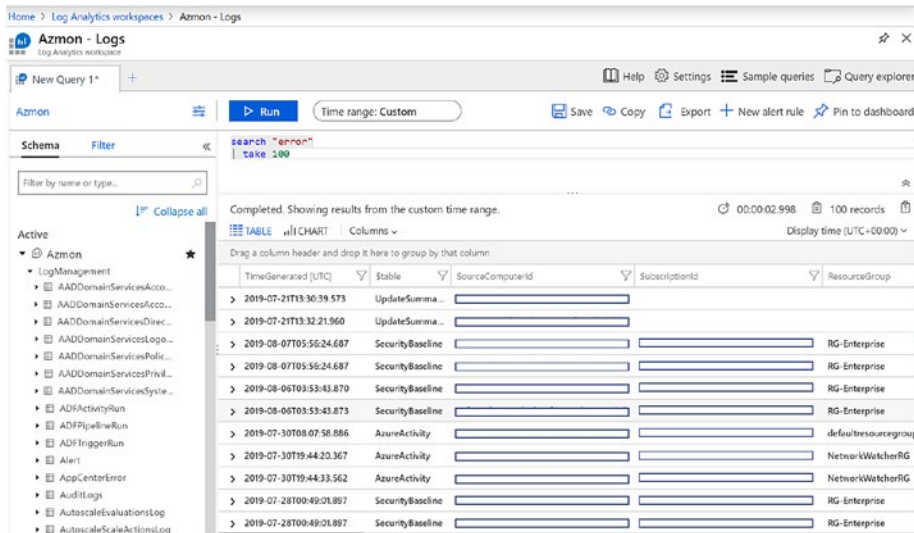
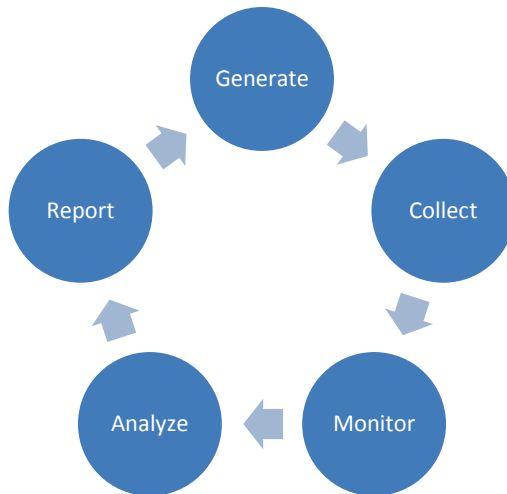


Figure 2-4. Azure Log analytics in Azure Monitor

## Monitoring Data Life Cycle

Azure monitoring of a data life cycle can be explained with the help of Figure 2-5 wherein it goes through various phases, and specific activities can be performed to get better insight.



*Figure 2-5. Monitoring Data Life Cycle*

### Generate

Azure Monitor starts collecting basic data from the moment a subscription is created or any resource is created in it. It collects all the subscription-related events relating to CREATE, DELETE, UPDATE operations. These subscription-level data are called **Activity Logs**.

**Metrics** of any specific resource is collected by the Azure platform once it is created. You can create various **dashboards** to review and analyze such data.

It is very useful for identifying resource performance and functionality issues and behaviors.

Should you require advanced, telemetric, and specific operational information, enabling **diagnostics** on the resources is the way to go. You can enable diagnostics for different Azure resources at the platform level and integrate those into Azure Monitor or other analysis tools.

For Compute resources such as virtual machines, virtual machine scale sets, a monitoring agent can be installed to ingest monitoring data into log analytics. These agents are commonly known as Log Analytics agents or **Microsoft Monitoring Agents (MMA)**.

You can use Azure **Application Insights** to collect diagnostics information from your application and analyze them for any anomalies. A small instrumentation package is installed into your application to achieve this. It will monitor the application and send all telemetric information to Azure for further analytics. AppInsight supports various apps and platforms including Azure App services, Cloud Services, .Net, Docker, Java, JavaScript, Node.js; whether hosted on-premises or on Azure. With Microsoft Monitoring Agent (MMA) and Application Insights, you cannot only gain insights from your Azure applications but from your on-premises systems and applications too. Since they can be easily installed and integrated, you will be able to have a complete monitoring view of the estate.

## Collect

Log analytics supports various sources to collect data Storage accounts, Event Hubs, Azure VM agents, and resources that can ingest data directly.

- Windows VM on-premises and on Azure: VMs with internet connectivity and with MMA agent installed, can report all configured logs to Azure log Analytics. However, in real life, there are scenarios wherein the systems do not have any internet connectivity, so a proxy service called Operations Management Suite Gateway (OMS Gateway) can be used. The OMS

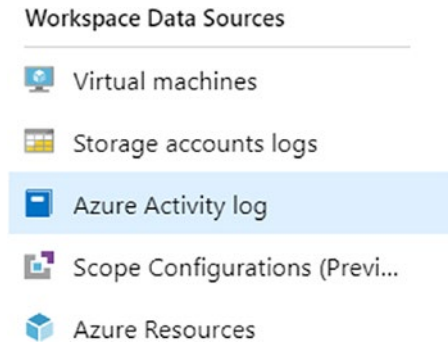


Gateway works as a collector machine onto which all logs are collected and saved, and it is connected to the internet. The Gateway system is configured to connect and ingest data to the Log Analytics portal. A similar configuration can exist on Azure virtual network as well, where the outgoing internet connectivity is blocked by default by a network security group or a network virtual appliance firewall. OMS Gateway will help collect the traffic and ingest the log into Log Analytics. On an Azure VM, however, you can install the Agent as an extension that can be easily integrated to the log analytics portal that you specify. It also simplifies the upgrade process because since they are an extension, they are auto-upgraded without having to manually update them.

- Linux systems: Azure Log Analytics also provides for a Linux agent that can help collect System logs and performance counters from Linux systems. The scenarios for Windows VMs not connected to the internet apply here as well.
- Azure resources: There are a few different ways how Azure resources can ingest or integrate data into log analytics. These include the following:
  - Connectors
  - Custom scripts to collect and post the data
  - Integrated Azure diagnostic platform that can push the data into log analytics
  - Direct diagnostics data to a storage account

- System center integration: You can also integrate your existing System center environment with Configuration manager and Operations manager to extend your existing monitoring and logging functionality.

The Log analytics portal provides for these sources to be added as a source (Figure 2-6).



**Figure 2-6.** *Workspace Data Sources*

## Monitor

Monitoring, analyzing, and reporting all are a continuous process. It all depends on one another and the continuous efforts to monitor the platform through metrics, alerts, events, and integration with ITSM systems. Fundamentals of monitoring remain the same even on Azure.

A few aspects of monitoring include the following:

- What system do you wish to monitor?
- Why do you want to monitor?
- For how long?
- Are you troubleshooting an existing issue?

- Which metrics and in which interval do you want to monitor?
- Which tool and reporting system to use?
- What action to take if such events are important or requires attention?
- Who should be notified of such an event?
- How should he be notified?
- What actions are expected from that person/team?
- Is it required to open a support incident or log another event somewhere?

The overall objective of monitoring is to ensure that you are aware of the system's functioning or some important part of the system's performance and activity, and you want to achieve some level of SLA if something goes wrong with it. In other words, you wish to achieve Quality-of-Service (QoS). We will discuss some of these scenarios in greater detail in upcoming sections.

## Analyze

Once we have the data from all various sources of your environment and collected in Log analytics, we can now query them to achieve our monitoring objectives.

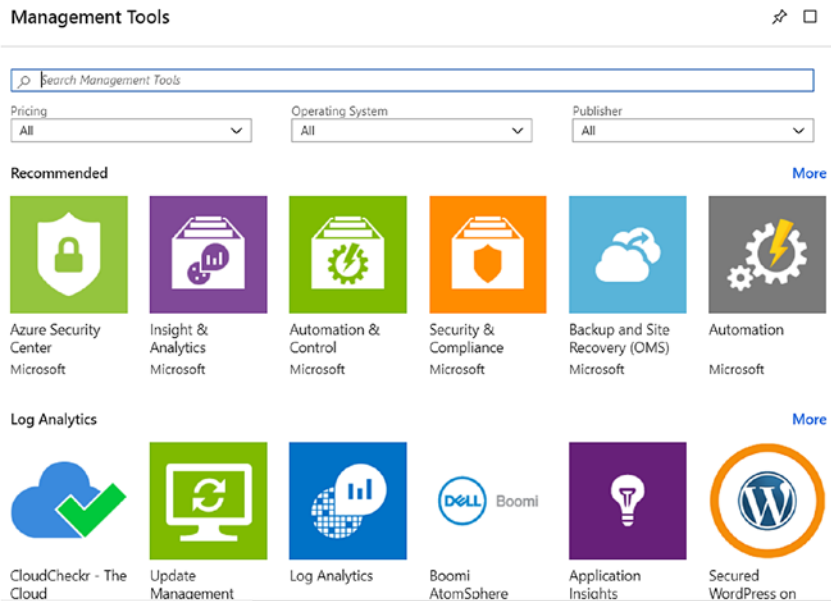
The **Log Search** feature helps to query this huge amount of data, correlate them, gather meaningful insight, and take action on them.

- Create a simple Log Search query or an advanced query to filter or transform the data.
- Keep all your important repeatable queries saved.

- Identify if an action needs to be taken and trigger an operational runbook or an automation script to trigger another action, email, SMS someone, or create a service request.

Most IT organizations already have an IT Service Management system, for example, ServiceNow, Provance, or System Center Service Manager, and it is advised to integrate it with Log analytics. By doing so, service requests can be raised as part of an automation process and all necessary activities can be centrally managed and monitored. IT Service Management Connector has the capability for bidirectional integration to create incidents, alerts, and events in its ITSM solution.

There are also **Management solutions** that can be used along with Azure Monitor to gather better insights from the resources and services in use. You can either enable them using the Azure Monitor console or from the All services menu in the Azure portal. Once integrated, it can help to provide additional insight of a particular application or service. It provides built-in queries, and it views and leverages other Azure services to analyze data. Most solutions are not charged for its integration. They may, however, attract a cost due to the collection, usage, and storage of the data. There are solutions available from partners and customers other than Microsoft (Figure 2-7).



*Figure 2-7. Management solution from Azure Marketplace*

## Report

There are various ways you can create reports and views to interpret the collected data. Here are some of the ways.

- **Log Analytics Dashboards:** This is the easiest way to create visualization for all saved searches that you may use every day based on your custom requirements.
- **Metrics Dashboards:** Similarly, you can create dashboards with all important metrics of your environment from different systems. You can include various views from various sources and create one integrated view. You can then publish them as a resource and apply Role-Based Access Control (RBAC) on them to ensure only necessary users can access them.

- **View Designer:** You can create custom views in the Log Analytics console to display various views of the data from the Log Analytics repository.
- **Export data to PowerBI:** This is another way to create beautiful visualization by exporting all results and datasets to PowerBI by integrating Log Analytics queries with PowerBI. The queries run on a specified schedule to keep the results up to date.

## Scenario 1: SLA

SLAs are often defined in terms of availability, throughput, or response time. Microsoft Azure provides for SLA for most of its services. A comprehensive detail of the various services of SLA can be found here: <https://azure.microsoft.com/en-ca/support/legal/sla/>. Since partners and customers design, develop, and use their products and solutions based on the Azure cloud platform, they would like to ensure that the platform is able to handle the defined set of availability and performance standards. Hence, SLA monitoring is an important aspect of the overall monitoring solution.

Remember that performance monitoring and SLA monitoring are closely related. The primary purpose of performance monitoring ensures optimal system functionality where the contractual obligation to that defines the “optimal” state is defined by the SLA. It also defines what happens if the standards are not met.

Cloud solutions may include decoupled components, which means that multiple different services can be used to create an entire solution; for example, a simple web application can include three or more various cloud services; a virtual machine, cloud storage (blob, queues or table), SQL database. In this case, each service may have a different SLA and availability specifications. At the same time, designing such solutions

may have included redundancy so that in the event of a single instance failure, the entire solution should not be impacted. Hence, while designing a monitoring solution, it is of the utmost important to ensure that all components of a solution is monitored to ensure the entire solution requirement is met.

SLA monitoring can be achieved by combining performance, availability, and health monitoring (we will discuss more on them in the next couple of pages). In our case of a web application, we should at least to the following:

- Monitor web endpoints on the VMs hosting the website;
- Virtual machine events and errors;
- Monitoring user requests tracing;
- Performance metrics of the virtual machine, storage, and SQL service;
- Service health for each system;
- Any other system's availability that is in use, for example, an application gateway.

You can also add diagnostics if you want to monitor what went wrong, when, and what time to identify and remediate granular levels of failure. Also, the overall system's uptime will not necessarily be the composite uptime of all services.

The SLA monitoring should result in identifying the overall aggregated performance values of the system, which may include:

- Percentage of time availability of the individual components during a period of time;
- Overall availability of the system as a percentage of uptime for any specific period;

- User response time breakup for each individual work item;
- Overall user response time during a specified period;
- Calculate success and failure rates of the user requests during a period of time.

These Azure tools can help achieve create a solution to monitor SLAs:

- Log Analytics
- AppInsight
- Performance monitoring with metrics
- Alerts
- Azure diagnostics
- Azure service status
- Azure service health monitoring
- Azure activity logs monitoring

## Scenario 2: Auditing and Compliance

Depending on the region, business, industry, or type of data an application handles, there may be specific legal or statutory regulations that require specific operations or all operations to be audited, monitored, and logged and saved over a period of time.

For example, banking or the insurance industry requires different auditing standards than an e-retail organization. Again, different parts of the whole system may require different kinds of data logging and retention.

Such audit data should be able to identify each user's action, sequence of events, time, and manner to ensure that appropriate authenticity and accountability can be determined. Since this data is very confidential and



sensitive in nature, required auditing and compliance data should be stored and retained securely and only accessed by specific, responsible, delegated individuals. The assigned auditor or analyst should be able to generate various reports to ensure all legal, statutory, and compliance standards are met.

Microsoft Azure provides a list of tools and methods to meet any of these needs. Here are a few of the Azure tools:

- Azure Activity logs help to identify the various tenant- and subscription-level operations performed. For example: if a storage account was deleted it will be captured in the activity log with proper date and time stamp, the user who initiated the activity, which storage account etc. Hence, all series of events and details in Activity log becomes an authentic source of all subscription level and tenant level operations.
- You can maintain all activity logs as audit trails for a longer-term retention on Azure log analytics or on a storage account based on the organization's needs.
- Virtual machine system, security, and audit logs.
- Azure Active directory reports.
- Azure RBAC to ensure only authorized users have access to specific resources.
- Azure standard regulatory and compliance certifications.
- Azure compliance Manager (<https://servicetrust.microsoft.com/ComplianceManager>), which is a workflow based risk assessment tool.
- Azure Trust portal, which showcases all Azure data privacy, trust, compliance, and industry certifications.

Services trust portal is the one-stop location for all security-, compliance-, and privacy-related documentation; certifications, and audit reports (Figure 2-8).



Figure 2-8. Service Trust Portal

## Scenario 3: Security

Security monitoring is one of the crucial operational activities of any enterprise today. All organization data based on data sensitivity or classification has to be securely handled, stored, and transferred at all times. The growing complexity of today's system also invites a growing need for data security, communication, and storage. For security monitoring, the key activities include the following:

- Detect any unauthenticated intrusion attempt;
- Identify unauthenticated and unauthorized data access and any attempt in this regard;
- Identify if any component or part of the system is under attack of any kind.

Most organizations today have already adopted a Security Information and Event Management (SIEM) system to include, analyze, and predict various events, alerts from different sources – systems, applications, devices, firewall, antivirus and intrusion-prevention systems. Apart from traditional security, modern organizations also explore ways to include new security systems that can perform vulnerability, port, and intrusion scanning in their IT infrastructure and deployments. This also includes cloud scale advanced analytics systems to aggregate, correlate, and identify security issues.

Azure provides a host of tools and services that can be leveraged to create a robust security system. It is not only secured at the platform level but also enables its customer to utilize the hosts of options to stay secure.

- With cloud, identity is the new security boundary with the diminishing border of network as a security perimeter. All logon events can be tracked and identified with Azure AD logging and reporting. Also, with dual factor authentication and location-based access and its reporting, this makes it possible to monitor global deployment scenarios with the cloud.
- All security, as well as diagnostic logs from virtual machines can be easily integrated into log analytics or SIEM systems to gain security insights.
- Role-based access control of each resource and implementing on-premises identity integration and monitoring really helps avoid the case of dual identity for the same user.
- Access-level telemetry of storage, network, and security center alerts can easily be leveraged to monitor any security issues.
- Azure has a host of other services that enable easy, scalable, robust monitoring of its resources and services.

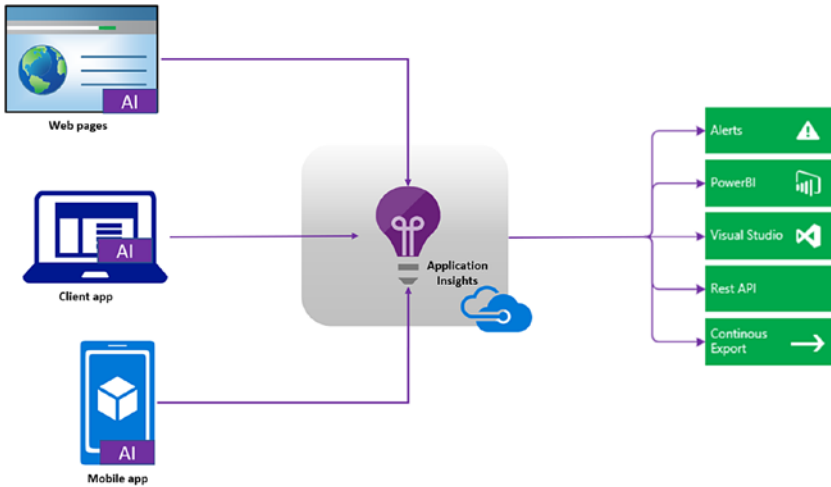
## Scenario 4: Availability

Availability monitoring is the cornerstone of the monitoring strategy of any organization. The simple truth is that if your applications and services are not available, you are out of business. Proactive monitoring and remediation of availability-related issues are high priority for any organization. The Azure Monitor service forms the nucleus of monitoring in Azure, acting as a centralized location to collate and analyze availability and performance information from sources in Azure as well as on-premises. Application Insights are the components of the Azure Monitor service that helps you track the availability of your applications irrespective of whether they are hosted in Azure PaaS services, VMs, containers, or on on-premise servers.

### Application Insights

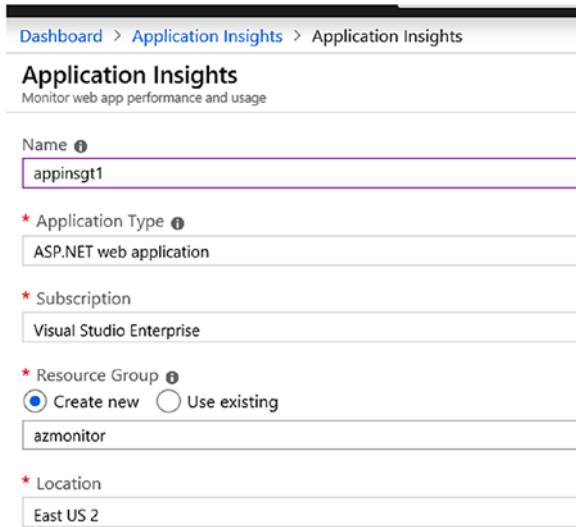
Application Insights, as the name indicates, is designed to give you deeper insights into the inner workings of your applications and flag any performance or availability issues. This is facilitated by deeper integration with the analytics platform that forms the core of Azure Monitor. It can be used to monitor applications developed in multiple platforms like .Net, Java, or Node.JS. It is also compatible with mobile apps, and can be used to monitor them by easy integration with the Visual Studio app center. Installation of a small instrumentation package in your application is all it takes to send telemetry information to the Application Insights service in Azure. Application Insights doesn't limit you to add the instrumentation package to applications hosted in Azure, which makes it a choice for enterprises with large-scale hybrid cloud or multi-cloud deployments. The response time metrics collected by Application Insights help you keep a tab on the status of application availability. The resultant telemetry data can be accessed and analyzed using PowerBi, used for configuring alerts, integrated to dashboards, etc.

A high-level workflow of how Application Insights collects and analyzes metrics from various sources is shown in Figure 2-9.



**Figure 2-9.** *Application Insights high-level architecture*

To start monitoring your application using Application Insights, deploy the service from the Azure portal. From the Azure portal, go to All services and search for Application Insights, select the service, and deploy a new instance (Figure 2-10).



Dashboard > Application Insights > Application Insights

### Application Insights

Monitor web app performance and usage

Name ⓘ  
appinsgt1

\* Application Type ⓘ  
ASP.NET web application

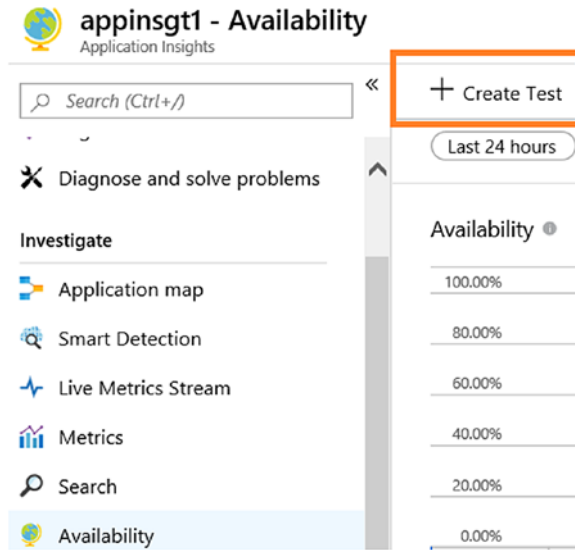
\* Subscription  
Visual Studio Enterprise

\* Resource Group ⓘ  
 Create new  Use existing  
azmonitor

\* Location  
East US 2

**Figure 2-10.** *Application Insights new instance*

In addition to including the Application Insights instrumentation package, you can also monitor the availability of the application using the Azure Monitor availability testing feature. From the Application Insights dashboard, go to Availability ► Create test (Figure 2-11).



**Figure 2-11.** Create a test

Provide the required input parameters to create the test (Figure 2-12).

^ Basic Information

\* Test name  
 ✓  
[Learn more about configuring tests against applications hosted behind a firewall](#)

Test type

\* URL ⓘ  
 ✓

Parse dependent requests ⓘ

Enable retries for availability test failures. ⓘ

Test frequency ⓘ

∨ Test locations  
 5 location(s) configured

∨ Success criteria  
 HTTP response: 200, Test Timeout: 120 seconds

∨ Alerts  
 Alert if 3/5 locations fails in 5 minutes.

**Figure 2-12.** *Inputs for creating a test*

Provide the following details to create the availability test:

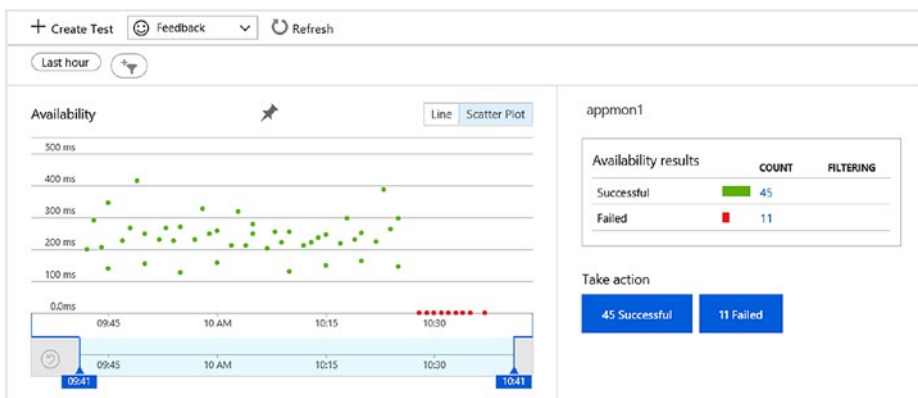
- A name to identify the test.
- The test can be created for a single URL or a sequence of multiple URLs using Multi-step web test option. For the latter, the scenario should be recorded using Visual Studio and uploaded to Application Insights. In this example, we are using a single URL. Note that the URL should be accessible over the internet.
- Select the option of Parse dependent request to check the time taken for all the elements of the web page such as scripts and images to be available. If any of those components fail to download within the timeout period, the test fails.



- Enable retries to avoid false alarms raised due to transient issues. The test is reported three times within an interval of 20 seconds before raising an availability alert.
- Select the test frequency and test locations. It is recommended to have a minimum of 5 locations to isolate website and network issues. The maximum number of locations possible is 16.
- The success criteria in this example is selected as HTTP response 200, with a timeout period of 120 seconds. You can also configure the test to check for content match to check for a specific string (case sensitive) in the response.
- Alerts are sent when the failed location count is above the set threshold

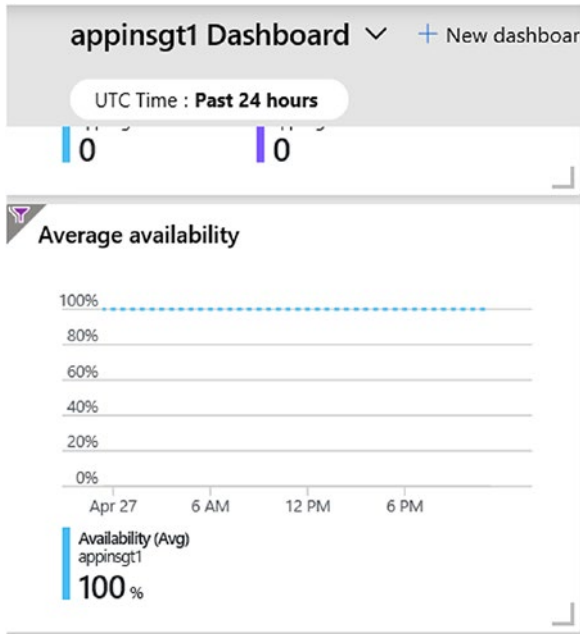
## Viewing the Insights

The availability can be reviewed as a line or scatter plot graph (Figure 2-13).



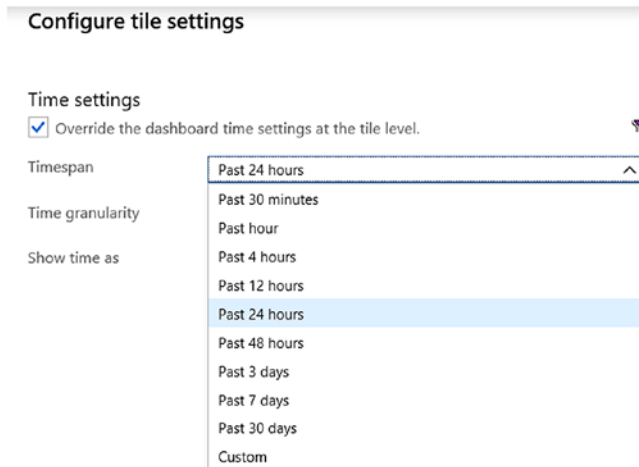
**Figure 2-13.** Availability scatter plot graph

You can also review the average availability of the application over a time span from the application dashboard. From the Application Insights overview tab, click on “Application Dashboard” (Figure 2-14).



**Figure 2-14.** *Availability average availability*

The default time span is 24 hours, but it can be customized by clicking the filter icon in the graph (Figure 2-15).



*Figure 2-15. Customize Time Span*

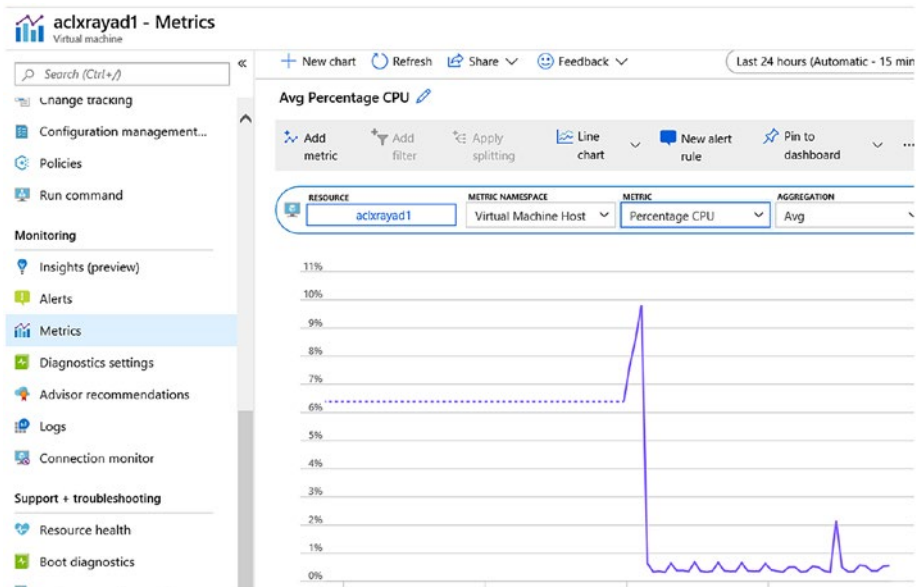
## Scenario 5: Performance

Azure Monitor can collect data from various sources like applications, VMs, or on-premises systems to give better insights to performance of respective resources. Each source system needs its own specific configuration as well as monitoring component service. For example, Application performance can be monitored using Application Insights by monitoring page views, load performance, response times, failure rates, and performance counters of VMs hosting the application. For VMs hosted in Azure, a set of performance metrics are collected by default by the platform. You can enable Azure Monitor for VMs (in preview at the time of writing this book) to get additional deeper insights on operating system health as well as performance. If containers are part of your application landscape in Azure, with Azure Kubernetes Service as hosting platform, you can use Azure Monitor for containers to monitor the performance of container workloads.

## VM Performance Monitoring

Performance of a system depends on multiple parameters like response rates to user requests, processing time, volume of concurrent users, network bandwidth, etc. To ensure that the system is performing at an optimal level, multiple performance counters can be used on a case-by-case basis. CPU processing time, memory utilization, Disk I/O, Network I/O and errors are some of these counters.

To view the default performance metrics of Azure VM, go to Monitoring ► Metrics ► Add metric and select from available options in the drop-down (Figure 2-16). This example shows the Percentage CPU Utilization metrics of the selected Azure VM. Other metrics that you could select for performance monitoring include Disk read operations/sec, Network Out Total, Inbound flows, etc.



**Figure 2-16.** Percentage CPU Utilization

For more in-depth metrics on performance, health, and dependency mapping of VMs, you should onboard the machine to Azure Monitor for VMs. It is prebuilt with performance charts based on guest VM OS performance metrics. To enable Azure Monitor for VMs, navigate to VM settings ► Monitoring ► Insights (preview). Select the subscription and log analytics workspace to store the data and click Enable (Figure 2-17).

\* Workspace Subscription ⓘ  
 Microsoft . ✓

Choose a Log Analytics Workspace ⓘ  
 iaasoms [eastus] ✓

Enable

**Figure 2-17.** Select workspace

The Performance charts will be available in a dashboard about 20–30 minutes after the configuration (Figure 2-18).

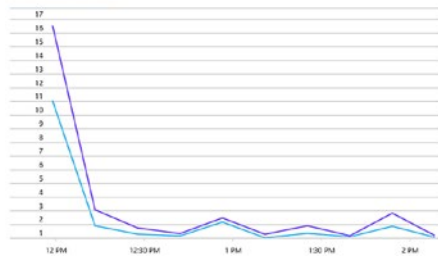
#### Performance Charts

⚡ Customize or add your own charts below in edit mode or by using the advanced editor

##### CPU Utilization %

Aggregates

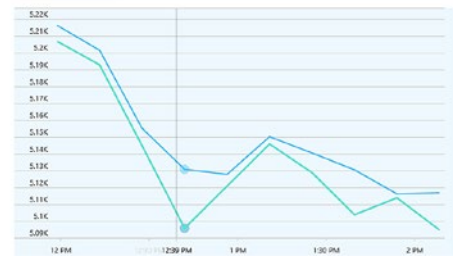
2 selected



##### Available Memory MB

Aggregates

3 selected



**Figure 2-18.** Performance Charts

## Application Performance

Along with availability monitoring, Application Insights can be used for performance monitoring of your applications. Monitoring can be integrated using Application Insights instrumentation or even during runtime in the case of Azure Web Apps. If your applications are running in an IIS server on-premise, application insight performance counters can be used to monitor system CPU, memory, disk, and network usage that could impact the application performance. Among the various parameters monitored by Application Insights, the following can be considered as major contributors toward performance monitoring:

- Request times, Response times, and failure rates of application pages;
- The response rates of any external services that the application is dependent on;
- Browser or server exceptions;
- AJAX call rates, response, and failures if any;
- Windows or Linux machines performance counters.

## Scenario 6: Usage

Tracking cloud resource usage is important for cost optimization and management, especially in large-scale deployments. Azure provides usage monitoring integrated into billing scopes, and the consolidated usage view is available from a cost analysis dashboard. Cost analysis shows the Azure resource usage and cost associated with it in selected scopes like subscription, resource groups, or even individual resources.

Cost analysis includes a feature to create a budget for a given scope and to monitor usage. This feature becomes very relevant in large-scale deployments where expenses could get out of hand if not monitored

and managed properly. Permissions to create and manage a budget depends on the role assigned to the user. Users with owner permission for a subscription can create, modify, or delete budgets. Users with a contributor and cost management contributor role can modify budgets created by others but cannot delete them. They can, however, create, modify, or delete budgets independently. Reader role and cost management reader roles give users read access to budgets. You can also configure alerts to be sent out when a configuration threshold is met.

A sample budget is shown below that would alert stakeholders when usage exceeds an acceptable monthly threshold (Figure 2-19).

The screenshot shows the Azure Budgets management interface. At the top, there are '+ Add' and 'Refresh' buttons. Below that is a search bar with the text 'Scope: Microsoft Azure Int'. A 'Search by name' input field and a 'All periods' dropdown menu are also visible. The main part of the image is a table with the following columns: NAME, SCOPE, RESET..., START D..., END DATE, BUDGET, CURREN..., and PROGRESS. A single row is displayed with the following data: NAME: AzureInternalU..., SCOPE: 0caabf5..., RESET...: Monthly, START D...: 4/1/2019, END DATE: 3/31/2021, BUDGET: ₹5,000.00, CURREN...: ₹171.37, and PROGRESS: 3.43%. A progress bar is shown next to the 3.43% value, and a three-dot menu icon is at the end of the row.

NAME	SCOPE	RESET...	START D...	END DATE	BUDGET	CURREN...	PROGRESS
AzureInternalU...	0caabf5...	Monthly	4/1/2019	3/31/2021	₹5,000.00	₹171.37	3.43%

*Figure 2-19. Sample Monthly threshold*

## Scenario 7: Health

While using Azure services, the context of health monitoring includes the health of the Azure platform, health monitoring of specific Azure services, and a level deeper into the diagnostics of individual resources.

**Azure Service status page:** Azure provides a comprehensive status of the platform in multiple geographies in the service status page that can be accessed at <https://azure.microsoft.com/en-us/status/>. The current status as well as status history is available for review from the above link. The status history shows previous outages, Root cause of the outage, Mitigation, and next steps.

**Azure Service health dashboard:** The Service health dashboard in Azure gives a personalized view of any Azure service outages as well as

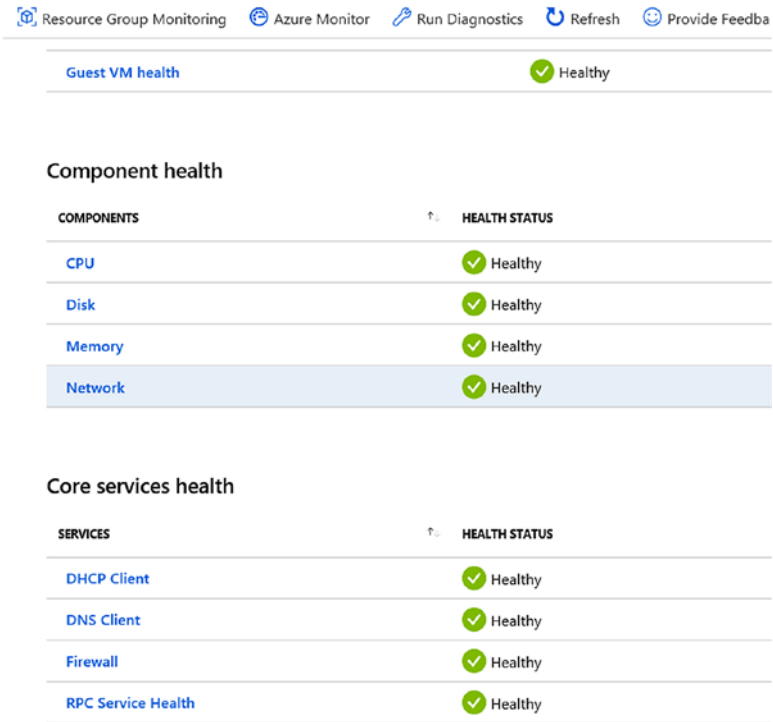
a summary of any potential impact to your resources. It can be used to create personalized dashboards as well as be configured to send alerts, should there be an outage impacting your services.

It covers the following three types of events that could affect the health of deployed services:

- Ongoing service issues impacting your deployed Azure services;
- Scheduled platform maintenance activities that could affect service availability;
- Health advisories related to deprecated Azure services and features or notifications on exceeding the usage quota;

**Azure Resource health:** The health status of each resource can be seen by selecting the respective resource ► Support + Troubleshooting ► Resource health. Any platform or non-platform-related issues that could impact the availability of the service will be notified here. Additionally, Workloads hosted in Azure VMs can have Azure Monitor for VMs enabled to give comprehensive component-level health data. This includes platform health, Guest VM health, component health, as well as the health of core services like DHCP, DNS, Firewall, etc. (Figure 2-20).





*Figure 2-20. Azure Resource Health*