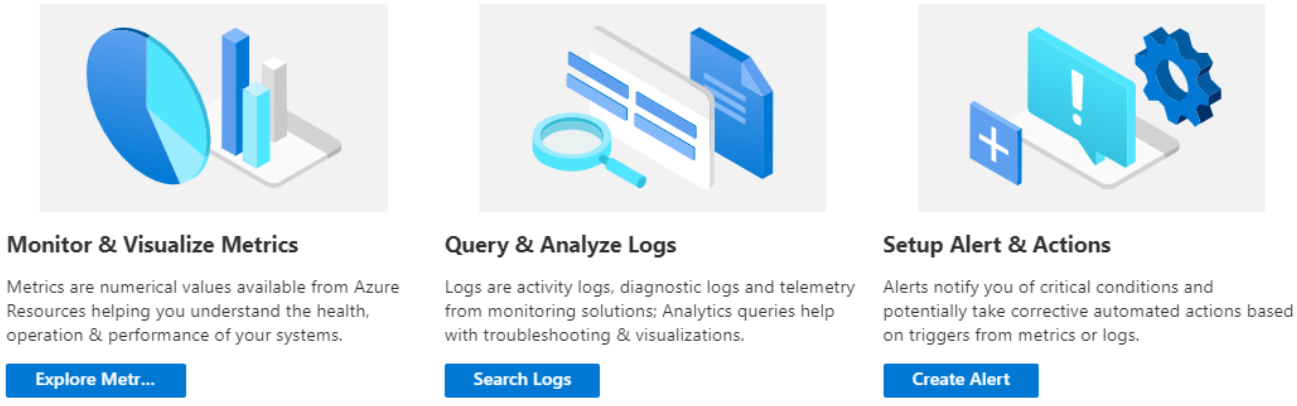
Agenda: Azure Monitoring and Log Analytics Workspace

- Azure Monitor Overview
- Monitoring Metrics
- Monitoring Platform Logs
    - Resource Logs
    - Azure Activity Logs
- Alerts
- Azure Log Analytic Workspace
- Collect data from an Azure VM
- Configure Azure SQL Database Auditing
- Azure Advisor
- Application Insight

Azure Monitor Overview

- Monitoring is the act of **collecting and analyzing data** to determine the **performance**, **health**, and **availability** of your business application and the resources that it depends on.
- An effective monitoring helps you increase your uptime by proactively notifying you of critical issues so that you can resolve them before they become problems.
- Azure Monitor enables core monitoring for Azure services by allowing the collection of **metrics, activity logs, and diagnostic logs**. For example, the activity log tells you when new resources are created or modified.

Key Capabilities of Azure Monitor



What Data does Azure Monitor collect?

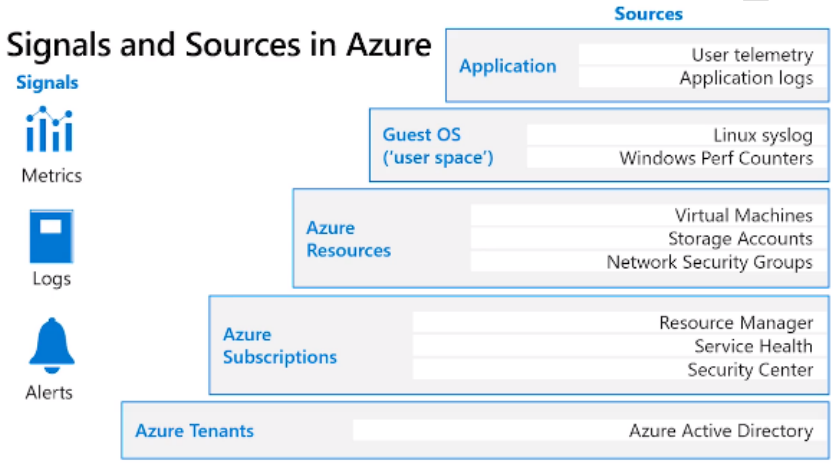
Azure Monitor can collect data from a **variety of sources**. You can think of monitoring data for your applications in tiers ranging from your application, any operating system and services it relies on, down to the platform itself.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

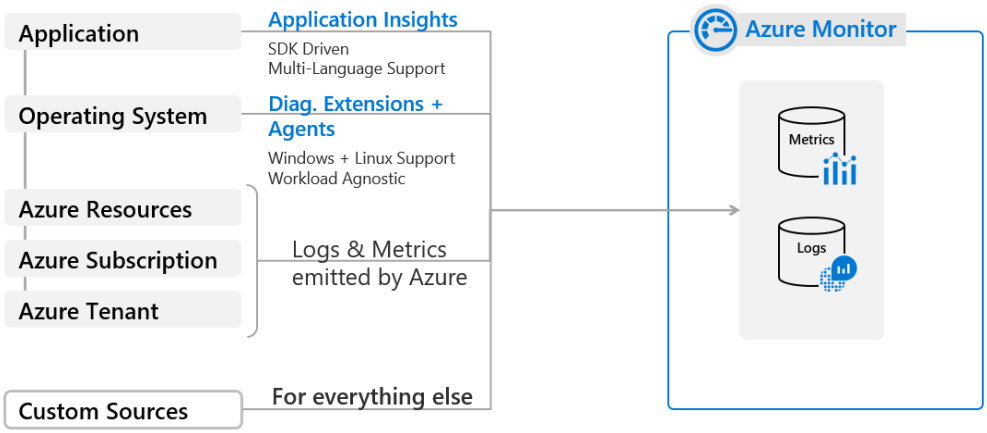
Azure Monitor collects data from each of the following tiers:

- **Application monitoring data**: Data about the performance and functionality of the code you have written, regardless of its platform.
- **Guest OS monitoring data**: Data about the operating system on which your application is running. This could be running in Azure, another cloud, or on-premises. Agents will be installed in the guest OS.
- **Azure resource monitoring data**: Data about the operation of an Azure resource including Metrics and Resource Logs.
- **Azure subscription monitoring data**: Data about the operation and management of an Azure subscription, as well as data about the health and operation of Azure itself.
- **Azure tenant monitoring data**: Data about the operation of tenant-level Azure services, such as Azure Active Directory.



All data collected by Azure Monitor fits into one of two fundamental types:

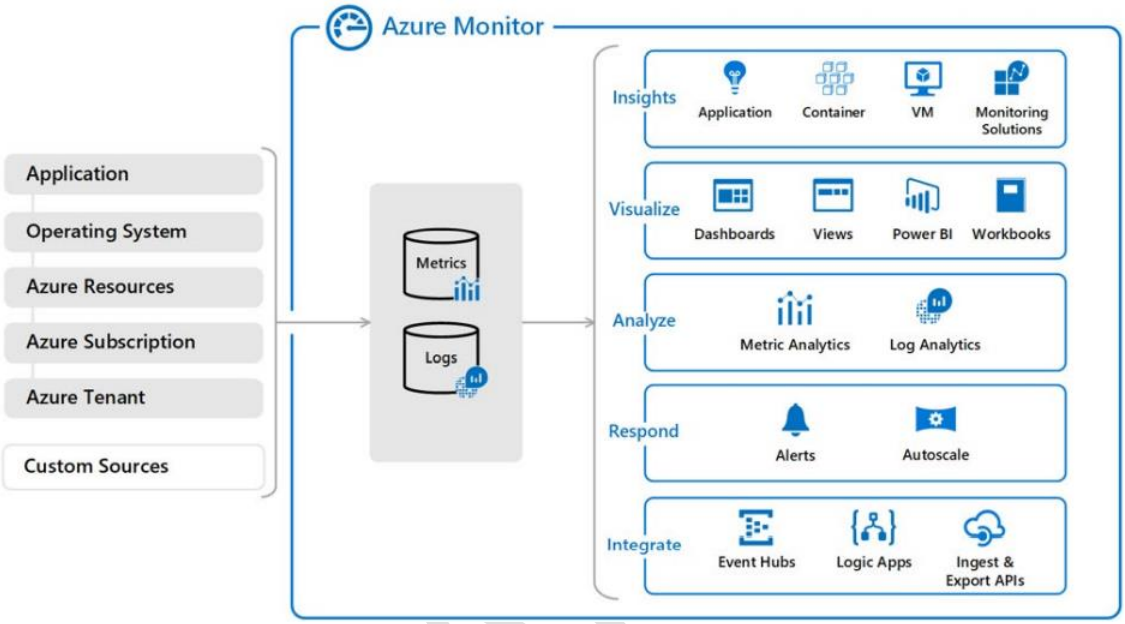
1. Metrics
2. Logs



Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

The following diagram gives a high-level view of Azure Monitor. At the center of the diagram are the data stores for metrics and logs, which are the two fundamental types of data use by Azure Monitor. On the left are the sources of monitoring data that populate these data stores. On the right are the different functions that Azure Monitor performs with this collected data such as analysis, alerting, and streaming to external systems.



Costs associated with monitoring

There **is no cost** for analyzing monitoring data that is collected by default. This includes the following:

- Collecting **platform metrics** and analyzing them with metrics explorer.
- Collecting **Activity log** and analyzing it in the Azure portal.
- Creating an **Activity log alert rule**.

There are no **Azure Monitor** costs for collecting and exporting logs and metrics, but there may be related costs associated with the destination:

- Costs associated with data ingestion and retention when collecting logs and metrics in Log Analytics workspace.
- Costs associated with data storage when collecting logs and metrics to an Azure storage account.
- Costs associated with event hub streaming when forwarding logs and metrics to Azure Event Hubs.

There may be Azure Monitor costs associated with the following:

- Running a log query.
- Creating a metric or log query alert rule.
- Sending a notification from any alert rule.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

- Accessing metrics through API.

https://azure.microsoft.com/en-in/pricing/details/monitor/

Monitoring – Metrics

- **Metrics** provide **performance statistics** for different resources.
- They are lightweight and capable of supporting **near real-time** scenarios.
- Metrics are **numerical values** that describe some aspect of a system at **a particular time**.
- Metrics are **collected at regular intervals** and are useful for **alerting**.
- For many Azure resources, the data collected by Azure Monitor is displayed on the **Overview page** in the Azure portal.
- You can **access metrics** from the Azure Portal, Monitor APIs (REST, and .Net) and analysis solutions such as the **Log Analytics** and **Event Hubs**.
- Metrics are enabled by default, and you can **access the past 90 days** of data. If you need to retain data for a longer period, you can **archive** metrics data to an Azure Storage account.
- **Metrics is a Shared Service** where you can specify the **resource, sub-service, metric, and aggregation criteria**. Additionally, you specify more than one metric, filter by a metric, and Export to Excel.

The following table lists the different ways that you can use metric data in Azure Monitor.

Analyze	**Use metrics explorer** to analyze collected metrics on a chart and compare metrics from different resources.
Visualize	Pin a chart from metrics explorer to an **Azure dashboard**.
Alert	Sends a **notification** or takes **automated action** when the metric value crosses a threshold.
Automate	To **increase or decrease** resources based on a metric value crossing a threshold.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

Export	**Route Metrics to Logs** to analyze data in Azure Monitor Metrics together with data in Azure Monitor Logs and to store metric values for longer than 93 days. **Stream Metrics** to an Event Hub to route them to external systems.
Retrieve	Access metric values from a command line using **PowerShell cmdlets or CLI** Access metric values from custom application using **REST API**.
Archive	Archive the **performance or health history** of your resource for compliance, auditing, or offline reporting purposes.

Retention of Metrics

For most resources in Azure, metrics are stored for **93 days**. There are some exceptions:

Guest OS metrics: These are performance counters collected by agent and routed to an Azure storage account

- **Classic guest OS metrics**: Retention for these metrics is 14 days.
- **Guest OS metrics sent to Azure Monitor Metrics**: Retention for these metrics is 93 days.
- **Guest OS metrics collected by Log Analytics agent**. Retention for these metrics is 31 days, and can be extended up to 2 years.

Application Insights log-based metrics.

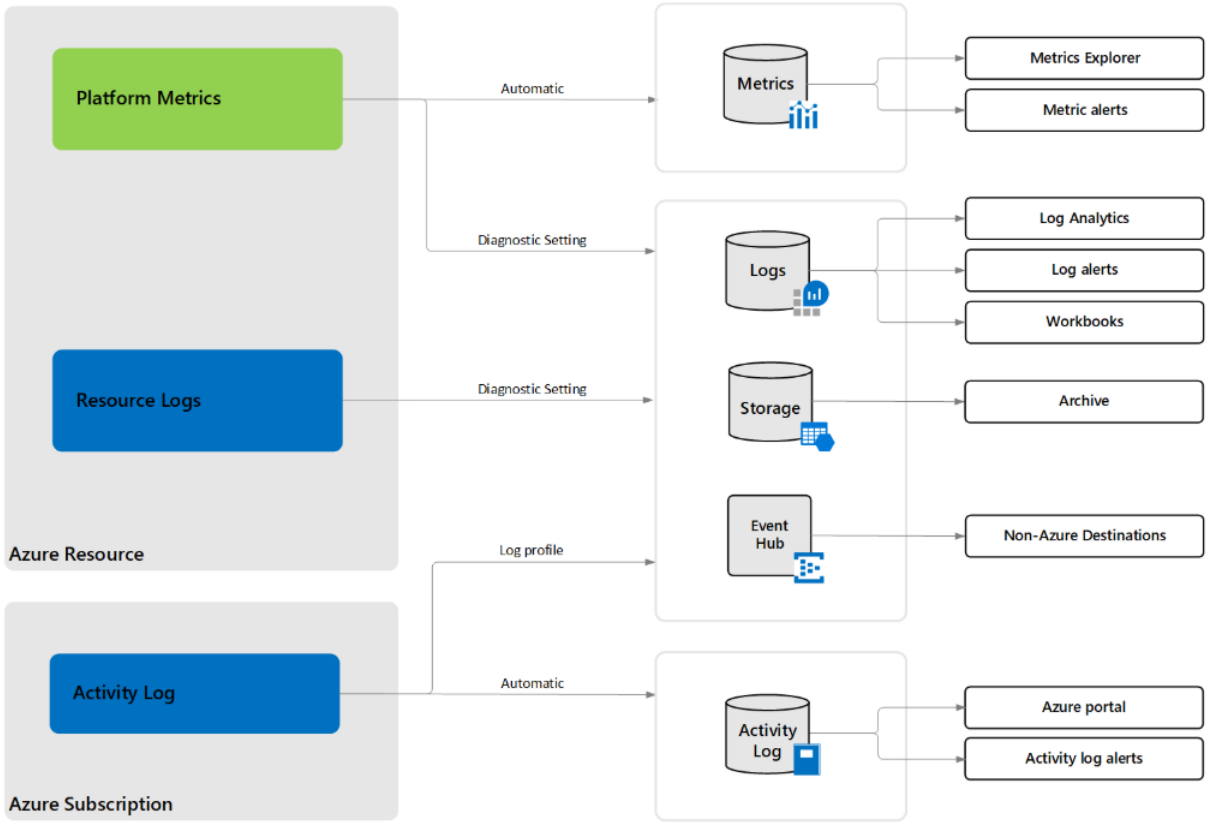
- Behind the scene, log-based metrics translate into log queries. For Application Insights resources, logs are stored **for 90 days**.

Note: Most metrics in Azure are stored for 93 days. However, you can only query for no more than 30 days worth of data on any single chart. This limitation doesn't apply to log-based metrics. In case , if you see a blank chart or your chart only displays part of metric data, verify that the difference between start- and end- dates in the time picker doesn't exceed the 30-day interval. Once you have selected a 30 day interval, you can pan the chart to view the full retention window.

Resources in Azure generate **logs** and **metrics** shown in the following diagram.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)



Platform Metrics

- **Numerical** values that are automatically collected at regular intervals and describe some aspect of a resource at a particular time.
- Platform metrics are **collected automatically** into Azure Monitor Metrics with no configuration required. Create a **diagnostic setting** to send entries to Azure Monitor Logs or to forward them outside of Azure.

Data collected by Azure Monitor Metrics is stored in a **time-series database** which is optimized for analyzing time-stamped data. Each set of metric values is a time series with the following properties:

- The time the value was collected
- The resource the value is associated with
- A namespace that acts like a category for the metric
- A metric name
- The value itself

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**.

Demo: Monitor Platform Logs

Option1: Locate the resource → **Overview** → Note the graphs which are provided by Azure Monitor

Option2: Locate the resource → Monitoring Section → **Metrics** → Opens Metric Explorer → Click Add metric

Option3: Using Monitor Service

a) Azure Portal → All Services → **Monitor → Metrics**

b) Add metric → Resource = <Storage Account> . . .

Monitoring Resource Logs

- Azure resource logs are platform logs that provide **insight into operations** that were performed within an Azure resource (the data plane)
- Resource logs are automatically generated by Azure resources but not collected without a diagnostic setting. Create a diagnostic setting to send entries to Azure Monitor Logs or to forward them outside of Azure.
- **Example**: Getting a secret from a Key Vault or making a request to a database.
- The content of resource logs varies by the Azure service and resource type.
- Resource logs are automatically generated by supported Azure resources, but they aren't available to be viewed unless you send them to a destination.
- You must create a **diagnostic setting** for each Azure resource to send its resource logs to a **Log Analytics workspace** to use with Azure Monitor Logs, **Azure Event Hubs** to forward outside of Azure, or to **Azure Storage** for archiving.
    - Event Hub and Azure Storage must be in **same region** as the resource.
    - For Azure Storage destination we can specify the **retention period** for each category of log (Retention can be between 1 to 365 days. 0 is infinite)
    - Not all resources have diagnostic settings
        - Logic App
        - CosmosDb
        - SQL Database
        - Etc…

The tables used by resource logs depend on what type of collection the resource is using:

- **Azure diagnostics** - All data written is to the ***AzureDiagnostics*** table. It's in NoSQL Format
- **Resource-specific** - Data is written to individual table for each category of the resource.

Note: All Azure services will eventually migrate to the Resource-Specific mode.

Example: Storage Data is stored in these tables.

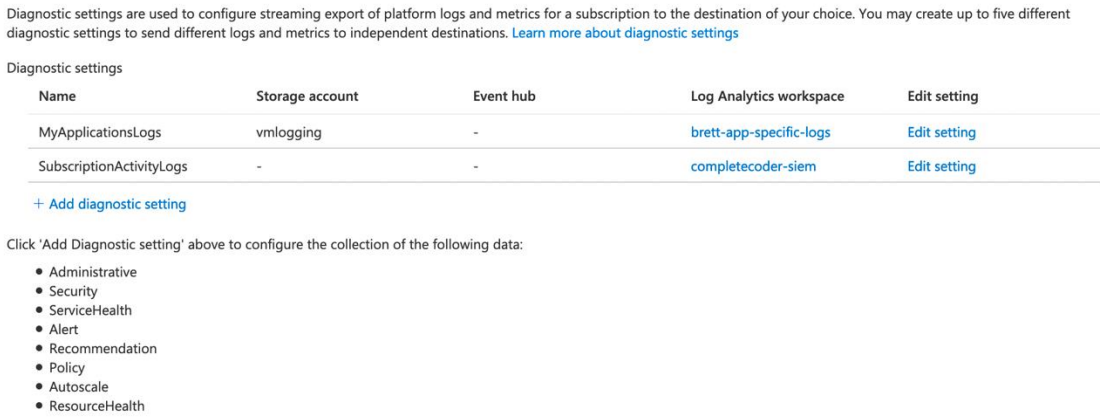
Table	Description
StorageBlobLogs	Logs that describe activity in blob storage.
StorageFileLogs	Logs that describe activity in file shares.
StorageQueueLogs	Logs that describe activity in queues.
StorageTableLogs	Logs that describe activity in tables.

Azure Activity Logs (at subscription level)

- The **activity log** tells you when new resources are created or modified.
- The Activity log is collected automatically with no configuration required and can be view in the Azure portal. Create a diagnostic setting to copy them to Azure Monitor Logs or to forward them outside of Azure.
- The Azure Activity Log is a **subscription log** that provides insight into **subscription-level events** that have occurred in Azure.
- Using the Activity Log, you can determine the '**what, who, and when'** for any **write operation** taken on the resources in your subscription. This includes such information as when a resource is modified or when a virtual machine is started.
- For additional functionality, you should create a **diagnostic setting** to send the Activity log to Log Analytic workspace, to Azure Event Hubs to forward outside of Azure, or to Azure Storage for archiving.

Through activity logs, you can determine:

- **What** operations were taken on the resources in your subscription.
- **Who** started the operation.
- **When** the operation occurred.
- The **status** of the operation.
- The values of other properties that might help you research the operation.


Diagnostic settings are used to configure streaming export of platform logs and metrics for a subscription to the destination of your choice. You may create up to five different diagnostic settings to send different logs and metrics to independent destinations. Learn more about diagnostic settings

Diagnostic settings

Name	Storage account	Event hub	Log Analytics workspace	Edit setting
MyApplicationsLogs	vmlogging	-	brett-app-specific-logs	Edit setting
SubscriptionActivityLogs	-	-	completecoder-siem	Edit setting

+ Add diagnostic setting

Click 'Add Diagnostic setting' above to configure the collection of the following data:

- Administrative
- Security
- ServiceHealth
- Alert
- Recommendation
- Policy
- Autoscale
- ResourceHealth

Configuring logs from the Activity Log blade allows you to collect the following specific log types:

- **Administrative**: Creating, updating, deleting, and action events. For example, creating a storage account or starting a virtual machine.
- **Security**: Security center alerts, such as suspicious double extension files executed.
- **Service Health**: Region-wide health events
- **Alert**: Alerts that you can define, for example, if you create an alert to monitor for high CPU usage.
- **Recommendation**: Azure Advisor recommendations,
- **Policy**: Events triggered by policies.
- **Autoscale**: VM Scale sets or Apps scaling out or in.
- **Resource** Health: Resource-specific health issues.

Activity Log retention

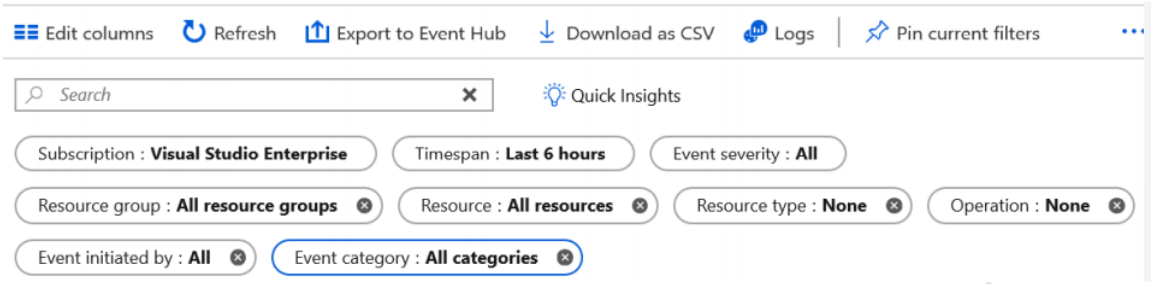
Once created, Activity Log entries are not modified or deleted by the system. Also, you can't change them in the interface or programmatically. Activity Log events are stored for **90 days**. To store this data for longer periods, collect it in Azure Monitor or export it to **Storage Account, Log Analytics or Event Hubs**.

Query the Activity Log Options:

1. View the Activity Log for a particular resource from the **Activity Log** option in that resource's menu in **Azure Portal**.
2. View the **Activity Log** for all resources from the **Monitor** menu in the Azure portal.
3. You can also retrieve Activity Log records with **PowerShell, CLI, or REST API**

Query the Activity Log in Portal

In the Azure portal, you can filter your Activity Log by these fields:



Examples:

1. Create or update Virtual Machine
2. Change the Size of VM
3. Create or Update SQL Database
4. Change the configuration of SQL Database
5. Wait for sometime…

Note: For some events example configuration/size changes, you can view the **Change history**, which shows what changes happened during that event time.

About Alerts

Alerts can be authorized in a consistent manner regardless of the monitoring service or signal type. All alerts fired and related details are available in single page.

Authoring an alert is a **three-step task** where the user first picks a target for the alert, followed by selecting the right signal and then specifying the logic to be applied on the signal as part of the alert rule.

Managing Alerts

You can alert on metrics and logs. These include but are not limited to:

- Metric values
- Log search queries
- Activity Log events
- Health of the underlying Azure platform
- Tests for web site availability

Create Alert Rule:

Select Any Service → Monitoring Section → Alerts → **New alert rule**

OR

When you are monitoring metrics, you can choose the menu "**New alert rule**"

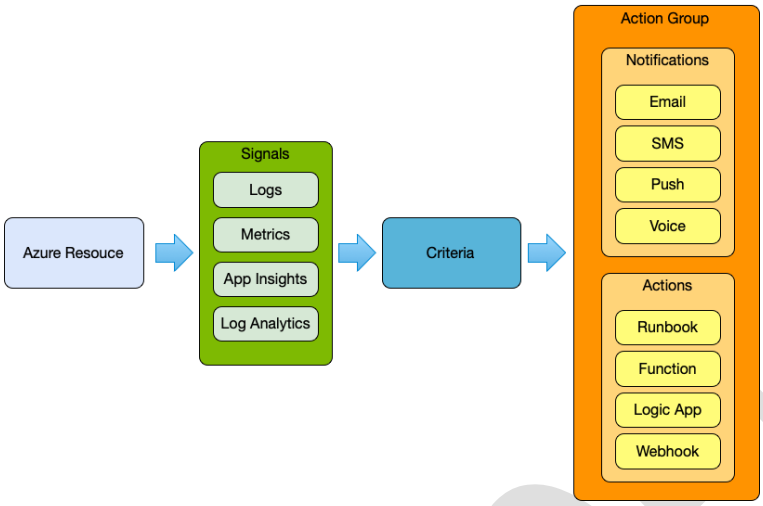
1. **Select Resource:** For example, Storage account.
2. **Add Condition**
   - Select signal. For example, **Used Capacity**.
   - Configure signal logic. For example, **over a six-hour period whenever the Used Capacity is over 1000000 bytes**.
     - Operator = Greater Than
     - Aggregation Type = Average
     - Threshold Value = 1000000
     - Aggregation Granularity = 6 hours
     - Frequency of Evaluation = Every 5 minutes
3. **Define Action Group**. Create an action group to notify your team via **email** and **text** messages or automate actions using **webhooks** and **runbooks**.
4. **Define Alert Details:** Alert rule name, description.

Action Groups

- Action groups enable you to configure a **list of actions** to take when the alert is triggered.
  - Automatic runbook
  - Azure Function
  - Email Azure Resource Manager role
  - Email/SMS/Push/Voice
  - ITSM
  - Logic App
  - Webhook
- Action groups ensure that the same actions are taken each time an alert is triggered.
- There are several action types you can select when defining the group: Select **Email/SMS/Push/Voice, Azure Function, Logic App, Webhook, IT Service Management, or Automation Runbook.**
- Various alerts may use the same action group or different action groups depending on the user's requirements.
- You may configure up to 2,000 action groups in a subscription.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

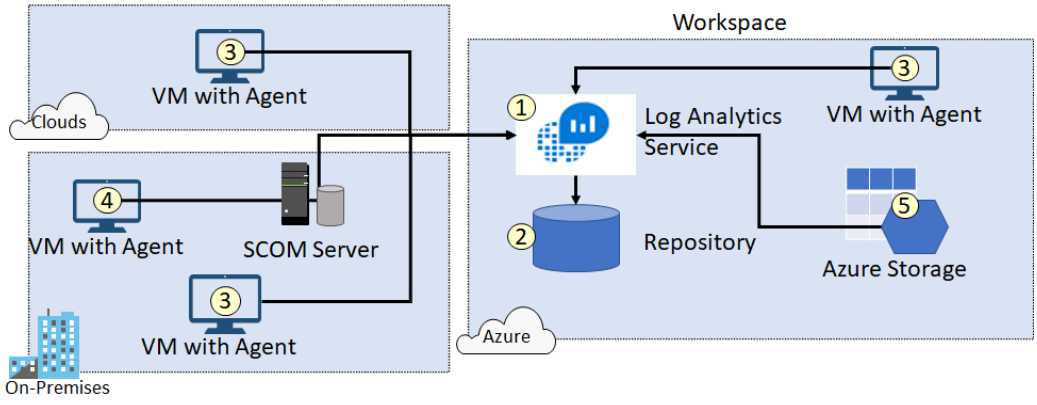
Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)



Azure Log Analytics Workspace

- **Log Analytics** helps you collect, correlate, search, and act on log and performance data generated by **operating systems and applications**.
- Log Analytics gives you a single interface for consuming and correlating the data, covering both Linux and Windows Server.
- It gives you real-time operational insights using integrated search and custom dashboards to readily analyze **millions of records** across all your workloads and servers regardless of their physical location.

Connected Sources: Connected sources are resources that generate data collected by Log Analytics



Ensure you can locate each of the following.

- The Log Analytics service (1) collects data and stores it in the repository (2). The repository is hosted in Azure.
- Computer agents (3) generate data to the Log Analytics service. These agents can run on Windows or Linux computers, virtual or physical computers, on-premises or cloud computers, and Azure or other cloud providers.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

- A System Center Operations Manager (SCOM) management group can be connected to Log Analytics. SCOM agents (4) communicate with management servers which forward events and performance data to Log Analytics.
- An Azure storage account (5) can also collect Azure Diagnostics data from virtual machine in Azure.

Data Sources: Data sources are the different kinds of data collected from each connected source.

1. Windows Event Logs
2. Windows Performance Counters
3. Linux Performance Counters
4. IIS Logs
5. Custom Fields
6. Custom Logs
7. Syslog for Linux.

Each data source has additional configuration options. For example, the Windows Event Log can be configured to forward Error, Warning, or Informational messages.

The Log Analytics agent and its data are used by a number of Azure monitoring tools, including.

- Logs Analytics
- VM Insights
- Azure Automation
- Azure Security Center
- Azure Sentinel

Demonstration – Log Analytics - Enable, Collect and View Data from Azure VM

Create a Log Analytics Workspace

1. All Services → Log Analytics workspaces → +Add
2. Provide a name = *DefaultLAWorkspace* → . . . → OK

Connect VM to Log Analytics

1. **Under Log Analytic Workspace → Data Sources Section → Virtual Machine → Connect**

   The agent is automatically installed and configured for your Log Analytics workspace. This process takes a few minutes, during which time the **Status** is **Connecting**.

   After you install and connect the agent, the **Log Analytics connection status** will be updated with **This workspace**.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

OR

Go to VM → Insights → Enable → Select the Log Analytic workspace

Go to VM → Insights → Azure Monitor → You'll see your VM with any other VMs in your subscription that are onboarded.

Alternatively, agents can be installed via a PowerShell or Azure CLI script or as part of an ARM template when deploying VMs, which is ideal when you want to automate your deployments. The following is an example ARM template snippet we can use:

```
{
  "type": "extensions",
  "name": "OMSExtension",
  "apiVersion": "[variables('apiVersion')]",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[concat('Microsoft.Compute/virtualMachines/', variables('vmName'))]"
  ],
  "properties": {
    "publisher": "Microsoft.EnterpriseCloud.Monitoring",
    "type": "MicrosoftMonitoringAgent",
    "typeHandlerVersion": "1.0",
    "autoUpgradeMinorVersion": true,
    "settings": {
      "workspaceId": "xxxxxxxxx"
    },
    "protectedSettings": {
      "workspaceKey": "xxxxxxxxxx"
    }
  }
}
```

Because the configuration can be written as JSON using an ARM template, you can also create an Azure Policy with a deployIfNotExists setting to automatically configure agents whenever a VM is created. Policies can also be used on other Azure resources to similarly configure the outputs of activity logs and diagnostics settings whenever a resource is defined.

Configure workspace to collect event and performance data:

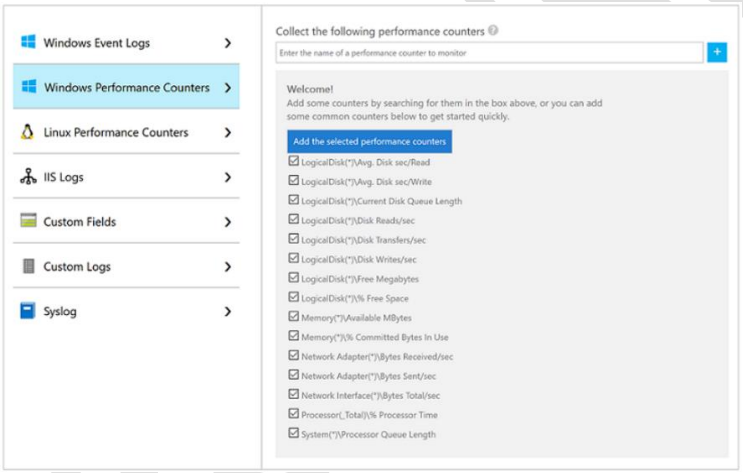
When you create a new Log Analytics workspace, it needs to be configured to collect logs. This configuration only needs to be performed once since configuration is sent to any virtual machines that connect to it.

Log Analytics can collect **events** from the Windows event logs or Linux Syslog and **performance counters** that you specify for longer term analysis and reporting, and take action when a particular condition is detected.

1. Select Log Analytic Workspace → Under Settings Section → **Advanced Settings** → **Data**
   a. Select **Windows Events Logs** → Collect events from the following event logs = Type "**System**" → Click + Sign → Check **Error and Warnings** → Save
   b. Select **Windows Performance Data** to enable collection of performance counters on a Windows computer.

      When you first configure Windows Performance counters for a new Log Analytics workspace, you are given the option to quickly create several common counters. They are listed with a checkbox next to each.

      Select **Windows Performance Data** → Click **Add the selected performance counters** →**Save**



View data collected

1. VM → Insights → **Performance** → This shows a select group of **performance counters** collected from the guest operating system of your VM.
2. VM → Insights → **Map** to open the maps feature which shows the processes running on the virtual machine and their dependencies.
   a) Select **Properties** to open the property pane if it isn't already open. Expand the processes for your virtual machine. Select one of the processes to view its details and to highlight its dependencies.
   b) Select your virtual machine again and then select **Log Events**. You see a list of tables that are stored in the Log Analytics workspace for the virtual machine. This list will be different depending whether

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

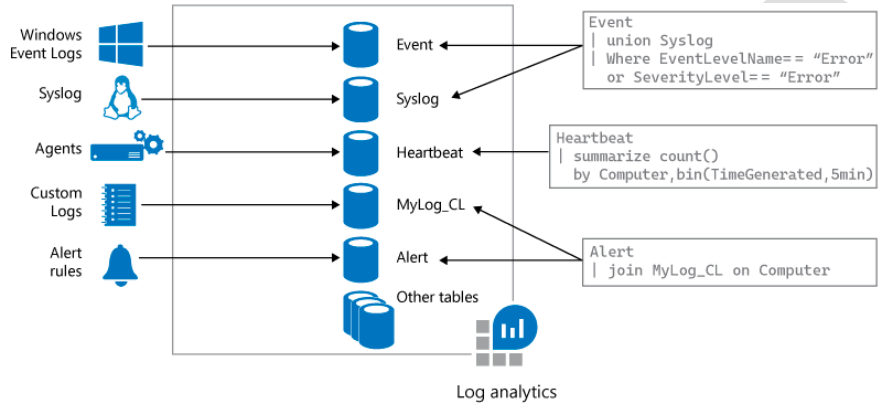
you're using a Windows or Linux virtual machine. Click the **Event** table. This includes all events from the Windows event log. Log Analytics opens with a simple query to retrieve event log entries.

View data collected in Log Analytics

Query Language Syntax:

Logs can be more easily filtered using a query language called **Kusto**, which provides greater flexibility and power when trying to find the data you need.

Each data source and solution stores its data in dedicated tables in the Log Analytics workspace. Documentation for each data source and solution includes the name of the data type that it creates and a description of each of its properties.



Now that you have enabled data collection, lets run a simple log search example to see some data from the target VMs.

2. Select **Log Analytic Workspace → Logs**
3. Run Queries
   a. **Perf**
   b. **Event** | search "error" (This query searches the *Event* table for records that contain the term *error* in any property.)

Note: The pipe (|) character separates commands, so the output of the first command is the input of the next command.

Add a filter to the query:

a) There is an arrow to the left of each record. Click this arrow to open the details for a specific record.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com
Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

b) Hover above a column name for the "+" and "-" icons to display. To add a filter that will return only records with the same value, click the "+" sign. Click "-" to exclude records with this value and then click **Run** to run the query again.

```
Event
| search "error"
| where EventLevel == 1
```

Use the Time range control

To use the **Time range** control, select it in the top bar, and then select a value from the dropdown list

If the query explicitly sets a filter for **TimeGenerated**, the time picker control shows **Set in query**, and is disabled to prevent a conflict.

```
Event
| search "error"
| where EventLevel == 1
| where TimeGenerated > ago(3d)
```

View and modify charts

- By default, results appear in a table. Select **Chart** above the table to see the results in a graphic view.
- The results appear in a stacked bar chart. Select other options like **Stacked Column** or **Pie** to show other views of the results.

You can Save, load and export queries:

- To load a saved query, select **Query explorer** at upper right.
- To export a query, select **Export** on the top bar, and then select **Export to CSV**

Azure Storage Log Analytics queries in Azure Monitor

a) To list all requests with anonymous access over the last three days.

```
StorageBlobLogs
| where TimeGenerated > ago(3d) and AuthenticationType == "Anonymous"
| project TimeGenerated, OperationName, AuthenticationType, Uri
```

b) To list the 10 most common errors over the last three days.

```
StorageBlobLogs
```


Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

```
| where TimeGenerated > ago(3d) and StatusText !contains "Success"
| summarize count() by StatusText
| top 10 by count_ desc
```

c) To list the top 10 operations that caused the most errors over the last three days.

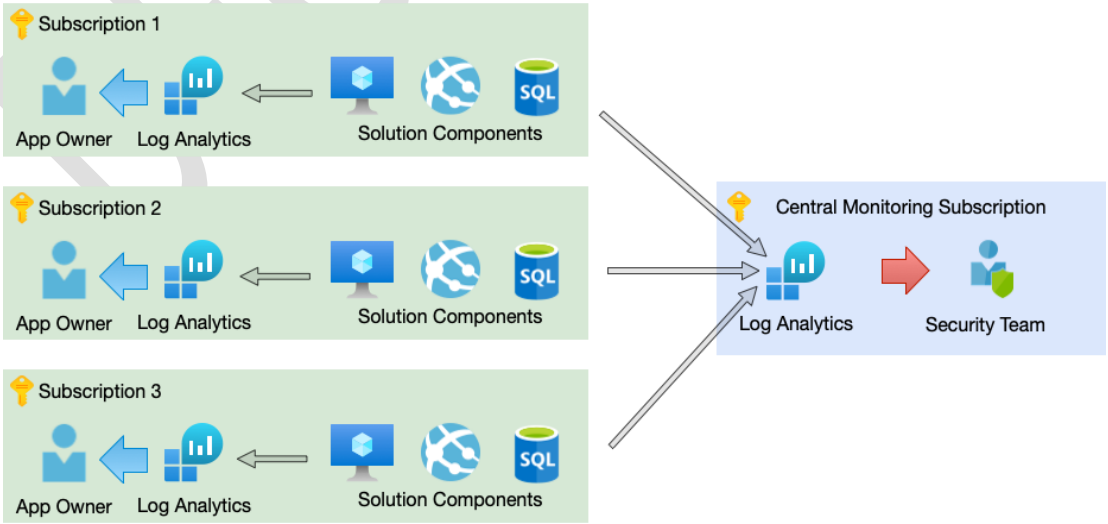
```
StorageBlobLogs
| where TimeGenerated > ago(3d) and StatusText !contains "Success"
| summarize count() by OperationName
| top 10 by count_ desc
```

Single vs Multiple Log Analytics Workspace:

A large multi-national organization may have multiple subscriptions either across regions or departments. We must then consider how systems are managed and monitored. For example, if there is a single team responsible for everything across the enterprise you would be better with a single workspace, with all resources, regardless of which subscription they are in, send the logs to that workspace. This can have implications on network traffic if those services are in different regions as you will incur additional ingress and egress costs.

Alternatively, you would have workspaces per region, or even per subscription, this helps provide more granular control, however your data is now spread across multiple workspaces.

A hybrid approach could include sending some logs to a central workspace, and other logs to individual workspaces. This design pattern is useful whereby different teams needs access to different logs and have different responsibilities. For example, a central workspace maybe used by a company-wide monitoring team, but individual service owners need visibility of application specific information and metrics, as we can see in the following diagram:

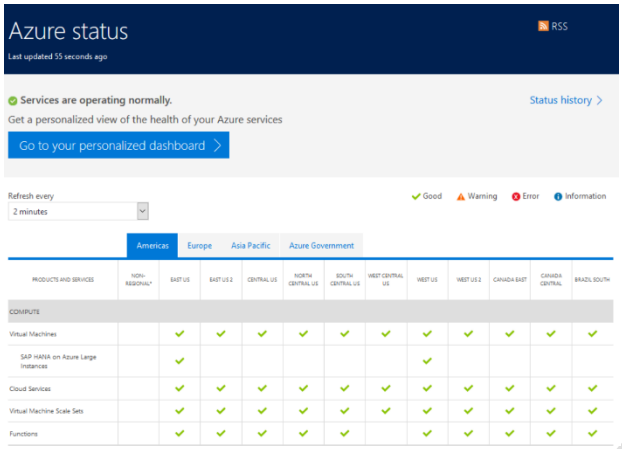


Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

Azure Service Health

- **Azure status provides a global view of the health of Azure services and regions**



Service health events:

- Service issues
- Planned maintenance
- Health advisories
- Security advisories

Service Health allows you to:

- See current issues which impact your services
- Get links and downloadable explanations
- Get support from Microsoft
- Pin a personalized health map to your dashboard
- Configure service health alerts

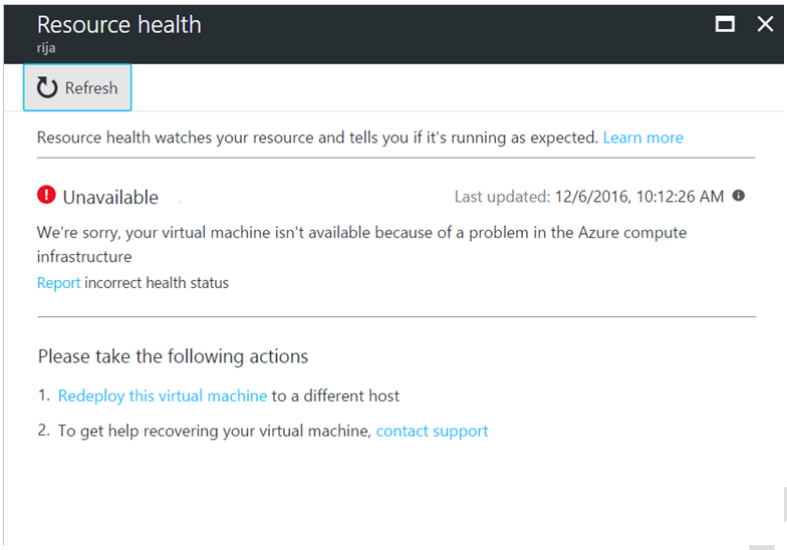
Resource Health Overview

Resource definition and health assessment

- Health status
  - Available
  - Unavailable (includes platform events and non-platform events)
  - Unknown
  - Degraded
- Reporting an incorrect status
- History information

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)



Configure Azure SQL Database Auditing

Azure SQL Database Auditing tracks database events and writes them to an audit log in your Azure Storage account.

Auditing can help you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.

SQL Database Auditing allows you to:

- **Retain** an audit trail of selected events. You can define categories of database actions to be audited.
- **Report** on database activity. You can use preconfigured reports and a dashboard to get started quickly with activity and event reporting.
- **Analyze** reports. You can find suspicious events, unusual activity, and trends.

1. **Create a Storage Account**
2. Navigate to **DemoDb (database)** → Security blade → **Auditing**
3. **Clear Inherit settings from server** check box and apply the following settings:
   a. Auditing: **ON**
   b. Auditing log destination: Check **Storage and Log Analytics**
   c. **Save**

Note: If Blob Auditing is enabled on the server, it will always apply to the database, regardless of the database settings.

To view audit logs

4. Perform CRUD Operations
5. Select Database → **Auditing** → **View Audit logs.**
6. Then, you have two ways to view the logs:

a. Clicking on **Log Analytics**

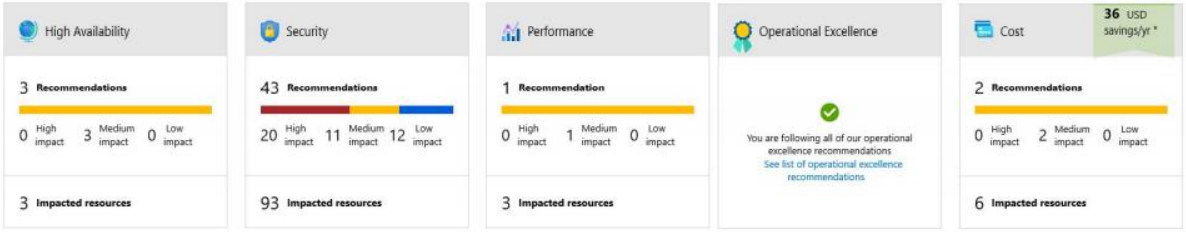
b. Clicking **View dashboard** at the top of the **Audit records**

More: https://docs.microsoft.com/en-us/azure/azure-sql/database/auditing-overview

Log Analytics Querying Demonstration page with Dummy Data: https://portal.loganalytics.io/demo#/discover/query/main This page provides a live demonstration workspace where you can run and test queries.

Azure Advisor

Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, high availability, and security of your Azure resources. The Advisor cost recommendations page helps you optimize and reduce your overall Azure spend by identifying idle and underutilized resources.



✔ Advisor provides recommendations for virtual machines, availability sets, application gateways, App Services, SQL servers, and Redis Cache.

Azure Application Insight

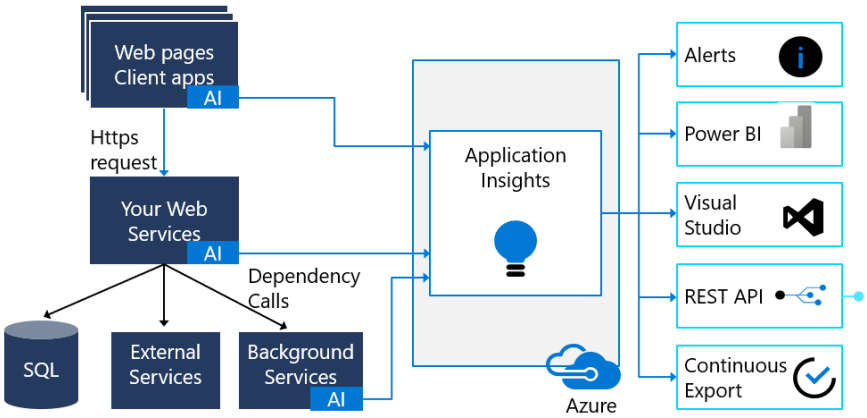
- Application Insights, a feature of Azure Monitor, is an extensible **Application Performance Management (APM) service** for developers and DevOps professionals. Use it to monitor your live applications.
- It will automatically detect performance anomalies, and includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app.
- It's designed to help you continuously improve performance and usability.
- It works for apps on a wide variety of platforms including .NET, Node.js, Java, and Python hosted on-premises, hybrid, or any public cloud.
- It integrates with your DevOps process, and has connection points to a variety of development tools.
- It can monitor and analyze telemetry from mobile apps by integrating with Visual Studio App Center.

What does Application Insights monitor?

- **Request rates, response times, and failure rates** - Find out which pages are most popular, at what times of day, and where your users are. See which pages perform best. If your response times and failure rates go high when there are more requests, then perhaps you have a resourcing problem.
- **Dependency rates, response times, and failure rates** - Find out whether external services are slowing you down.
- **Exceptions** - Analyze the aggregated statistics, or pick specific instances and drill into the stack trace and related requests. Both server and browser exceptions are reported.
- **Page views and load performance** - reported by your users' browsers.
- **AJAX calls** from web pages - rates, response times, and failure rates.
- **User and session counts**.
- **Performance counters** from your Windows or Linux server machines, such as CPU, memory, and network usage.
- **Host diagnostics** ingested from Docker or Azure.
- **Diagnostic trace logs** from your app - so that you can correlate trace events with requests.
- **Custom events and metrics** that you write yourself in the client or server code, to track business events such as items sold or games won.

How does Application Insight work?

- You install a small **instrumentation package** in your application, and set up an Application Insights resource in the Microsoft Azure portal. The instrumentation monitors your app and sends telemetry data to the portal. (The application can run anywhere—it doesn't have to be hosted in Azure.)
- You can instrument not only the web service application but also any background components and the JavaScript in the webpages themselves.
- In addition, you can pull in telemetry from the host environments such as performance counters, Azure diagnostics, or Docker logs.
- You can also set up web tests that periodically send synthetic requests to your web service.
- All these telemetry streams are integrated in the Azure portal, where you can apply powerful analytic and search tools to the raw data.



There are two ways to configure your app to send data to Application Insights:

- **Runtime instrumentation:** Runtime instrumentation captures telemetry without requiring you to change the web app's source code. You can quickly enable this turnkey solution from the Azure portal when you first create your web app or anytime afterwards. Use this method when you want to set up Application Insights without involving developers or when code management policies prevent you from changing the app's source code. Note that some advanced data displays aren't available when you use only runtime instrumentation.
- **Build-time instrumentation.** With this method, developers add a server-side SDK to the web app's code. For example, in an ASP.NET Core app, a developer could reference a NuGet package to access the SDK. When you instrument your app with the Application Insights SDK, you can enable full functionality and the richest set of visualizations in Application Insights. This type of instrumentation also enables you to add custom events and telemetry to your code to monitor unusual or unique behavior.

Note:

- Runtime instrumentation and automatic client-side instrumentation is supported only on Windows web apps. These features rely on capabilities of IIS, the web server technology that powers Windows apps on App Service. The use of Application Insights in Linux apps is fully supported, but you need to modify application code to reference the Application Insights SDK.

Adding JavaScript SDK

- To automatically inject the JavaScript SDK and necessary configuration into pages served by your web app, add a new application setting named **APPINSIGHTS_JAVASCRIPT_ENABLED** and set the value to true.
- OR add the below script to every page just before </head> (Ideally do in Master Layout page)

```
<script type="text/javascript">
var appInsights=window.appInsights||function(a){
```

```
function b(a){c[a]=function(){var b=arguments;c.queue.push(function(){c[a].apply(c,b)})}}var
c={config:a},d=document,e=window;setTimeout(function(){var
b=d.createElement("script");b.src=a.url||"https://az416426.vo.msecnd.net/scripts/a/ai.0.js",d.getElementsByTag
Name("script")[0].parentNode.appendChild(b)});try{c.cookie=d.cookie}catch(a){}c.queue=[];for(var
f=["Event","Exception","Metric","PageView","Trace","Dependency"];f.length;)b("track"+f.pop());if(b("setAuthentic
atedUserContext"),b("clearAuthenticatedUserContext"),b("startTrackEvent"),b("stopTrackEvent"),b("startTrackPag
e"),b("stopTrackPage"),b("flush"),!a.disableExceptionTracking){f="onerror",b("_"+f);var
g=e[f];e[f]=function(a,b,d,e,h){var i=g&&g(a,b,d,e,h);return!0!==i&&c["_"+f](a,b,d,e,h),i}}return c
}({ instrumentationKey:"<your instrumentation key>" });
window.appInsights=appInsights,appInsights.queue&&0===appInsights.queue.length&&appInsights.trackPageVie
w();
</script>
```

To Enable Server Side telemetry in the app:

1. Add the NuGet Package:
    a. **Microsoft.ApplicationInsights.AspNetCore**
2. ConfigureSevice:
    services.AddApplicationInsightsTelemetry();
3. appsettings.json
    "**ApplicationInsights": { "InstrumentationKey": "putinstrumentationkeyhere" }**

Note: Set the browser window for the app side-by-side with the portal showing the Live Metrics Stream. Notice the incoming requests on the **Live Metrics Stream** as you navigate around the web app.

There are several parameters that you can set, although in most cases, you shouldn't need to. For example, you can disable or limit the number of AJAX calls reported per page view (to reduce traffic). Or you can set debug mode to have telemetry move rapidly through the pipeline without being batched.

To set these parameters, add them after the **instrumentationKey** as properties of the same JSON object.

```
// Send telemetry immediately without batching.
// Remember to remove this when no longer required, as it can affect browser performance.
enableDebug: boolean,
// Don't log browser exceptions.
disableExceptionTracking: boolean,
// Don't log ajax calls.
disableAjaxTracking: boolean,
```

```
// Limit number of Ajax calls logged, to reduce traffic.
maxAjaxCallsPerView: 10, // default is 500
// Time page load up to execution of first trackPageView().
overridePageViewDuration: boolean,
// Set dynamically for an authenticated user.
accountId: string,
```

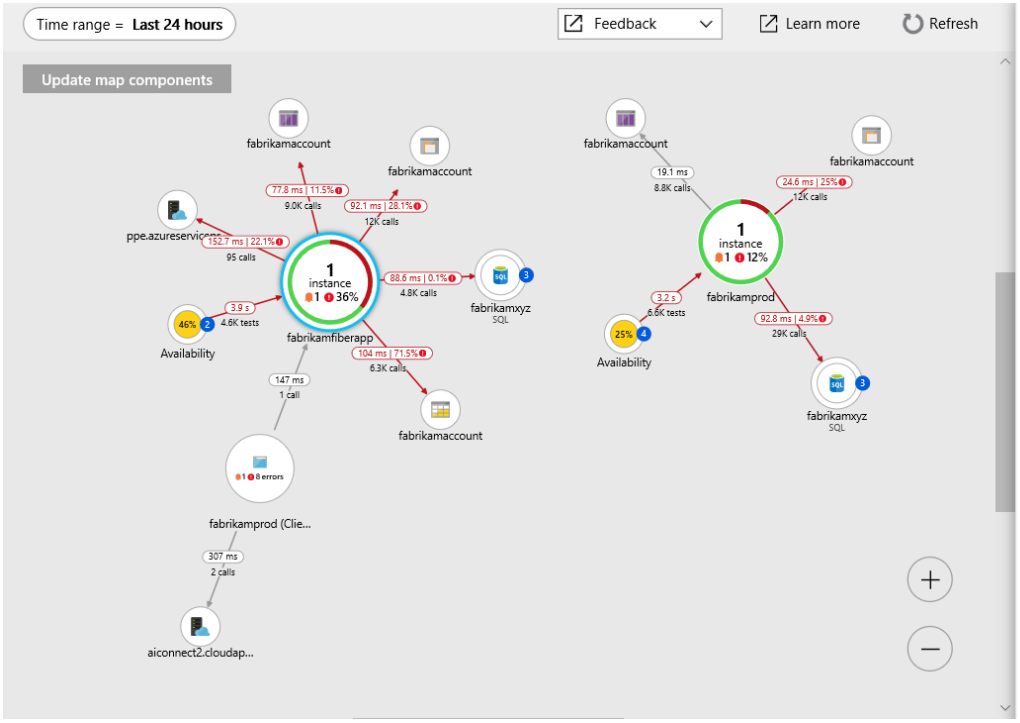
URL ping test:

It uses more advanced HTTP request functionality to validate whether an endpoint is responding. It also measures the performance associated with that response, and adds the ability to set custom success criteria coupled with more advanced features like parsing dependent requests, and allowing for retries.

AppInsight → Availability

Application Maps:

Application Map helps you spot performance bottlenecks or failure hotspots across all components of your distributed application. Each node on the map represents an application component or its dependencies; and has health KPI and alerts status. You can select through from any component to more detailed diagnostics, such as Application Insights events. If your app uses Azure services, you can also select through to Azure diagnostics, such as SQL Database Advisor recommendations.



Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)