**Agenda:** **Azure Monitoring and Log Analytics Workspace**

* Azure Monitor Overview
* Monitoring Metrics
* Monitoring Platform Logs
* Managing Alerts
* Azure Log Analytic Workspace
* Monitoring Azure VM

### 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

### 

### The following diagram gives a high-level view of Azure Monitor:

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

### Metrics are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios.

### Logs contain different kinds of data organized into records with different sets of properties for each type. Telemetry such as events and traces are stored as logs in addition to performance data so that it can all be combined for analysis.

### 

## **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.

Azure Monitor collects data from each of the following tiers:

* **Application monitoring data** (User Telemetry / Application Logs) - Data about the performance and functionality of the code you have written, regardless of its platform.
* **Guest OS monitoring data** (Linux Syslog and Windows Perf Counters) - 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.

### 

### A diagram of a software Description automatically generated

The following table lists the different ways that you can use metric and logs 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. |
| **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. |

### 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:

* Accessing metrics through API.
* Running a log query.
* Creating a metric or log query alert rule.
* Sending a notification from any alert rule.

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

Collecting Logs and Metrics

Diagram

Description automatically generated

**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**.
* 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.
* Metrics are **collected at regular intervals** and are useful for **alerting**.
* You can **access metrics** from the Azure Portal, Monitor APIs (REST, and .Net) and analysis solutions such as the **Log Analytics** and **Event Hubs**.

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

Graphical user interface, text, application, email

Description automatically generated

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**.

## **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.

**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**

1. Azure Portal 🡪 All Services 🡪 **Monitor 🡪 Metrics**
2. 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** or **Read/Write/Delete operations of a Storage Account**.
* The content of resource logs varies by the Azure service and resource type.
* 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. Log Analytics Workspace can be **any region**.
  + Destination service can be in any Subscription.
  + 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)

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)**

* Azure Activity Log is a subscription log that provides insight into subscription-level events that have occurred in Azure
* 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.
* 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.

**Examples of Activity Log:**

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

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

## **Query the Activity Log Options:**

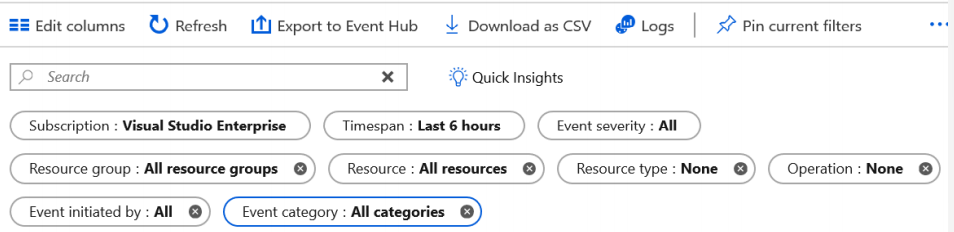
1. View the Activity Log for a particular resource from the **Activity Log** option in that resource's menu in **Azure Portal**.
2. View the **Activity Log** for all resources from the **Monitor** menu in the Azure portal.
3. You can also retrieve Activity Log records with **PowerShell, CLI, or REST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

## **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**.

**Activity Log Filters:**

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



**Event Categories**

Graphical user interface, text, application, email

Description automatically generated

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.

**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

A picture containing text, sign, screenshot

Description automatically generated

### 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
  + Logic App
  + Webhook
  + ITSM - Connect Azure and a supported IT Service Management (ITSM) product/service. This requires an ITSM Connection.
* 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**](https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-sms-alert-behavior)**/Push/Voice, Azure Function,** [**Logic App**](https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-action-groups-logic-app)**,** [**Webhook**](https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-activity-log-alerts-webhook)**,** [**IT Service Management**](https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-itsmc-overview)**, or Automation Runbook.**
* **Every alert can use one or more action groups.**
* New Action Group is required only if the set of Actions in a existing action group is different for a New Alert.
* You may configure up to 2,000 action groups in a subscription.

**Demonstration: 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**

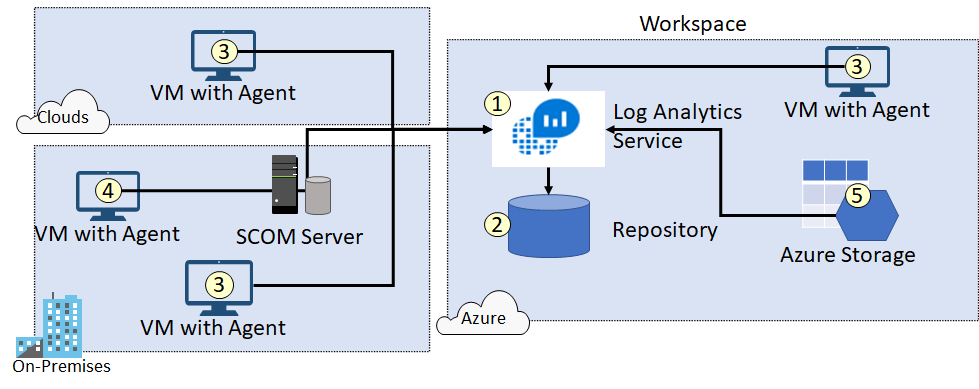
* Select signal. For example, **Used Capacity**.
* Configure signal logic. For example, **over a six-hour period whenever the Used Capacity is over 1000000 bytes**.
  + Operator = Greater Than
  + Aggregation Type = Average
  + Threshold Value = 1000000
  + Aggregation Granularity = 6 hours
  + Frequency of Evaluation = Every 5 minutes

1. **Define Action Group**. Create an action group to notify your team via **email** and **text** messages or automate actions using **webhooks** and **runbooks**.
2. **Define Alert Details:** Alert rule name, description.

**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.
* 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.

**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:

Graphical user interface, application

Description automatically generated

**Note: Log Analytics Workspace can be shared across multiple regions and multiple subscriptions.**

**Demonstration – Enable, Collect and View Data from Azure VM**

**Step1: Create a Log Analytics Workspace**

1. Search 🡪 Log Analytics workspaces 🡪 +Add
2. Provide a name = **Demo-eastus-*workspace***🡪 . . . 🡪 OK

**Monitoring VM**

* Azure Monitor starts automatically collecting metric data for your virtual machine host when you create the VM.
* To collect logs and performance data from the guest operating system of the virtual machine, though, you must install the **Azure Monitor agent**. You can install the agent and configure collection using either **VM insights** or by creating a **data collection rule** as described below.

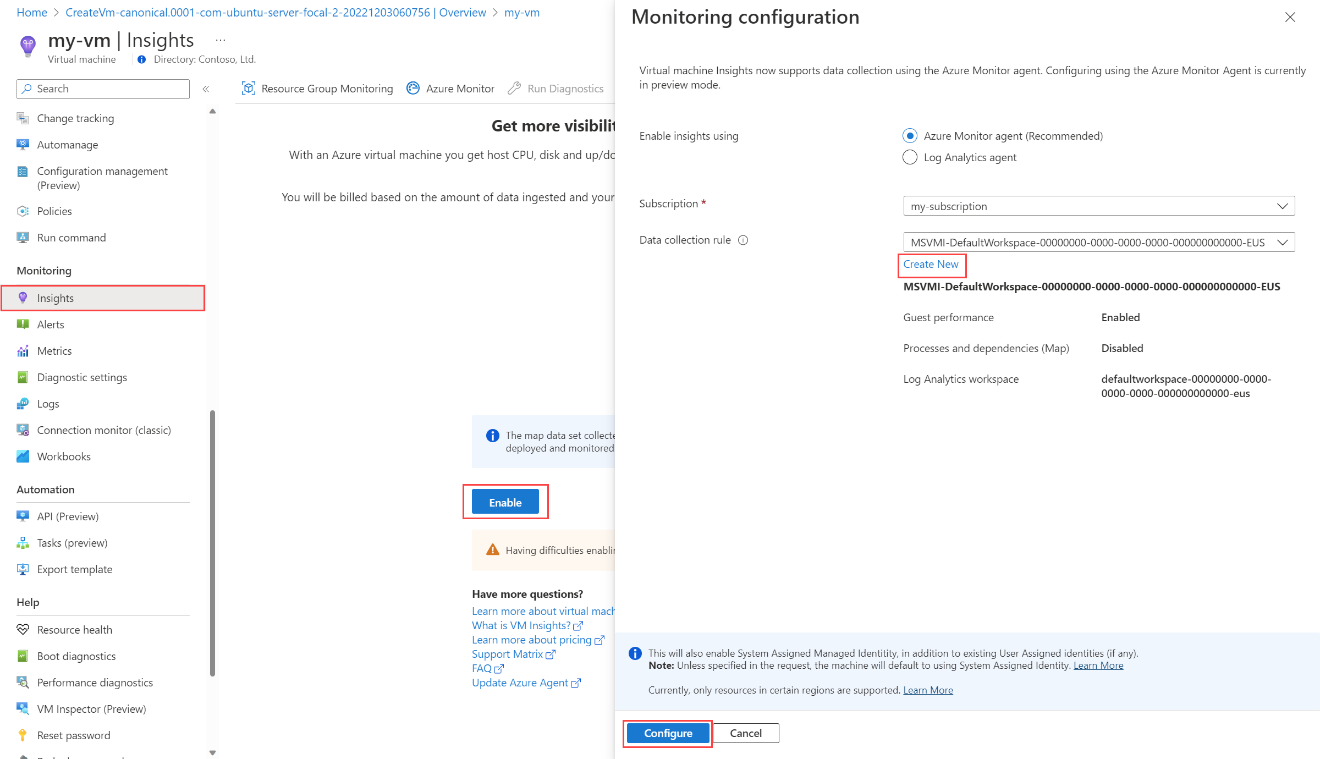
**About VM Insight**

VM insights is designed to monitor your Azure and hybrid virtual machines in a single interface. VM insights provides the following benefits beyond other features for monitoring VMs in Azure Monitor:

* Simplified onboarding of the Azure Monitor agent and the Dependency agent, so that you can monitor a virtual machine guest operating system and workloads.
* Pre-defined data collection rules that collect the most common set of performance data.
* Pre-defined trending performance charts and workbooks, so that you can analyze core performance metrics from the virtual machine's guest operating system.
* The Dependency map, which displays processes that run on each virtual machine and the interconnected components with other machines and external sources.

**Step2: Enable VM Insight:**

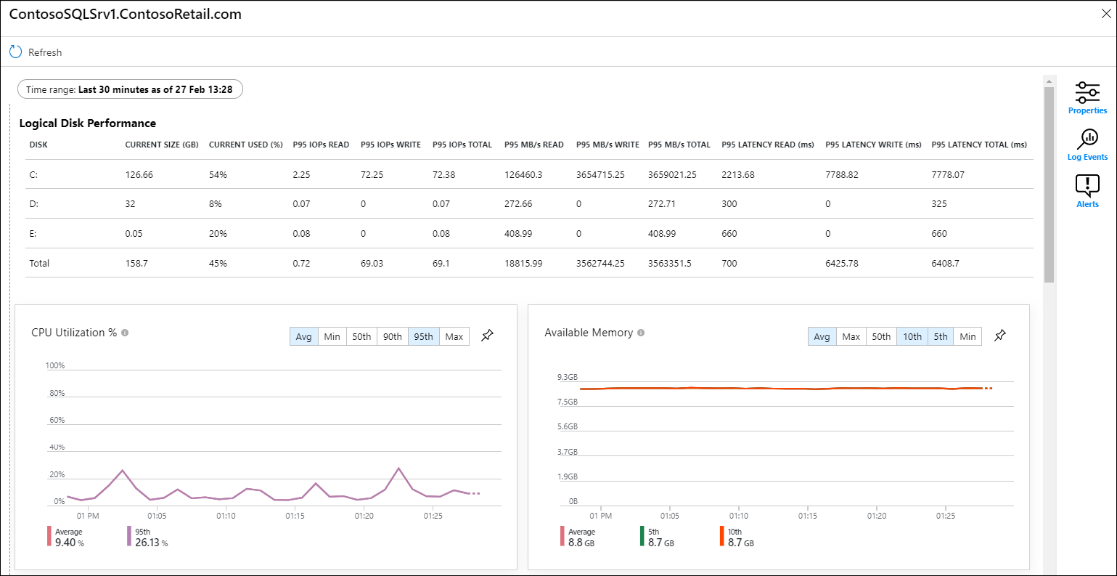
1. Demo-VM 🡪 **Insight** (Monitoring Section) 🡪 Enable 🡪 Create New
   * Data collection rule name = DemoCollectionRule
   * Log Analytics Workspaces = **Demo-eastus-*workspace***
   * Create



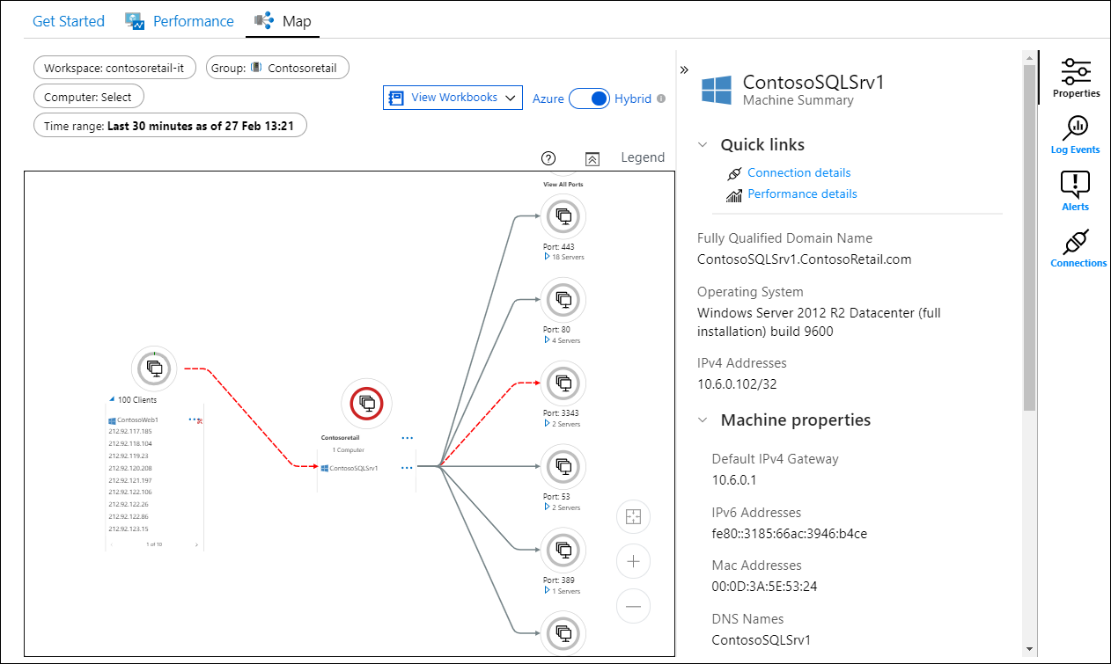
**Step3: View Performance:**

1. Refesh the browser
2. Demo-vm 🡪 Insights 🡪 Performance Tab

This shows the values of key guest metrics over time.

[](https://learn.microsoft.com/en-us/azure/virtual-machines/media/monitor-vm/vminsights-01.png#lightbox)

1. Select the **Map** tab to view processes and dependencies for the virtual machine. View the processes running on it by expanding **Processes**.

[](https://learn.microsoft.com/en-us/azure/virtual-machines/media/monitor-vm/vminsights-02.png#lightbox)

**Step4:** [**Collect guest logs and metrics from Azure virtual machine**](https://learn.microsoft.com/en-us/azure/azure-monitor/vm/tutorial-monitor-vm-guest)

VM insights collects performance data from the VM guest operating system, but it doesn't collect log data such as Windows event log or Syslog of Linux. Now that you have the machine monitored with Azure Monitor Agent, you can create another **data collection rule** to perform this collection.

1. Search **Monitor 🡪 Data Collection Rules (Settings Section) 🡪 +Create**
2. **Basic Tab:**
   * Rule Name = Demo-DataRule
   * Region = East US
   * Platform Type = Windows
   * Data Collection Endpoint = none
3. **Resources Tab**
   * + Add resources
   * Expand Resource Group 🡪 Select VM 🡪 Apply
4. **Collect and deliver Tab**
   * **+ Add data source**
     + Data source type = **Performance counters**
     + Select Basic and Check all

A screenshot of a computer

Description automatically generated

* + - Next : Destination
    - + Add destination
    - A screenshot of a computer

      Description automatically generated
    - Add data source
  + **+ Add data source**
    - Data source type = **Windows Event Logs**
    - Select Basic and Check required options
    - Add data source

A screenshot of a computer

Description automatically generated

1. **Review + Create 🡪 Create**

**Step5: View Guest Metrics**

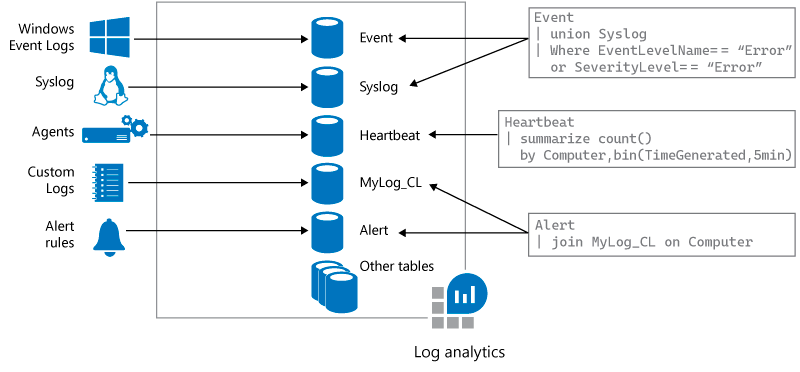
1. **Demo-vm 🡪 Metrics**
2. Select **Metric Namespace** = **Virtual Machine Guest**.

Note: If you don't see Virtual Machine Guest, you might need to wait a few minutes for the agent to deploy and data to begin collecting.

**Step6: View Logs collection by Log Analytics**

**Kusto Query Language Syntax:**

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.

****

1. **Demo-vm 🡪 Logs**

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

1. Select **Log Analytic Workspace** 🡪 **Logs**
2. Run Queries
   1. **Perf**
   2. **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:**

1. There is an arrow to the left of each record. Click this arrow to open the details for a specific record.
2. 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**

1. 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

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

StorageBlobLogs

| where TimeGenerated > ago(3d) and StatusText !contains "Success"

| summarize count() by StatusText

| top 10 by count\_ desc

1. 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