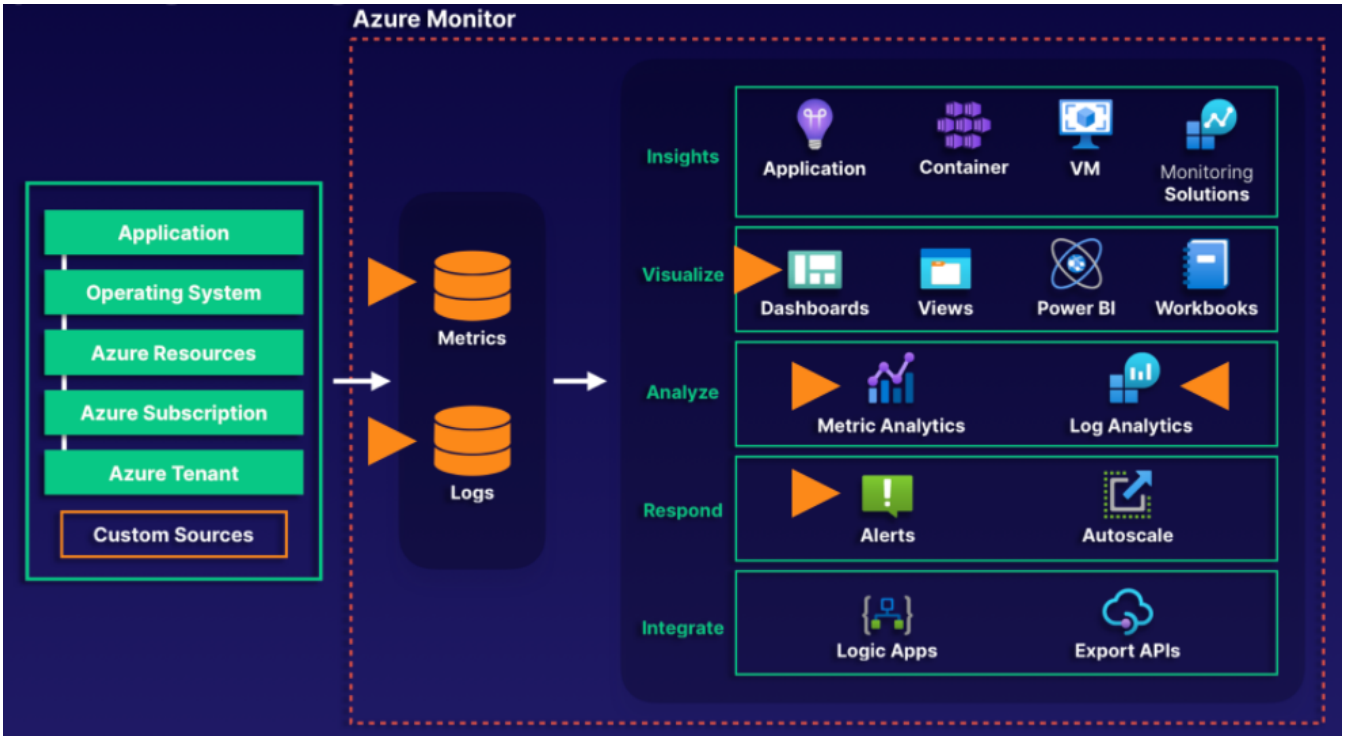
Implementing Azure Monitor

Introduction

This hands-on lab is designed to teach you the basics of Azure Monitor. You will explore how Azure Monitor works in a SQL database through the implementation of Azure Monitor and the configuration of alerts, security logs, log analytics queries, and basic metric charts. This is designed as a beginner-friendly lab with little to no prior experience required. We will not be diving deep into Log Analytics or query language but will instead focus on the implementation aspects of Azure Monitor.



Prepare the Environment

Provision a SQL Database

1. In the search bar at the top, enter "sql database".
2. Select SQL databases in the search results.
3. Click Create SQL database.
4. Set the following values:
   - Resource group: Select the lab-provided resource group
   - Database name: Enter a database name of your choice
   - Server: Click Create new.

- Server name: Enter a globally unique name (e.g., sqlserver<TODAY'S DATE_YOUR INITIALS>)
- Server admin login: admin
- Password: Enter a password of your choice (make sure you can remember it, as we'll use it later)
- Location: (US) West US
- Click OK.

- Want to use SQL elastic pool?: No
- Compute + storage:
    1. Click Configure database.
    2. Click Looking for basic, standard, premium?.
    3. Click Basic.
    4. Ensure it's set to 5 (Basic) DTUs at 2 GB.
    5. Click Apply.

5. Click Next: Networking.
6. Set the following values:
    - Connectivity method: Public endpoint
    - Allow Azure services and resources to access this server: Yes
    - Add current client IP address: Yes

7. Click Next: Additional settings.
8. Set the following values:
    - Use existing data: Sample
    - Enable Azure Defender for SQL: Not now

9. Click Next: Tags.
10. Click Review + create > Create.

Provision a Log Analytics Instance

1. In the search bar at the top, enter "log analytics".
2. Select Log Analytics workspaces in the search results.
3. Click Create log analytics workspace.
4. Set the following values:

- Resource group: Select the lab-provided resource group
- Name: Enter a globally unique name (e.g., loganalyticsmonitor-<TODAY'S DATE_YOUR INITIALS>
- Region: West US

5. Click Review + create > Create.

Implement Monitoring in SQL Database

Enable Auditing and Send Logs to a Newly Created Storage Account

1. Click the hamburger menu icon in the upper left, and select All resources.
2. Select the SQL database you just created.
3. Click Auditing in the left-hand menu.
4. Set Enable Azure SQL Auditing to On.
5. Under Audit log destination, check the box for Storage.
6. Click Configure.
7. On the Storage settings page, set the following values:

- Subscription: Leave default
- Storage account:
    - Click Configure required settings.
    - Click Create new.
    - Set the following values:
        - Name: Enter a globally unique name (e.g., storagemonitoring-<TODAY'S DATE_YOUR INITIALS>
        - Account kind: StorageV2 (general purpose v2)
        - Performance: Standard
        - Replication: Locally-redundant storage (LRS)
    - Click OK.
- Retention (Days): Leave default (0) Storage access key: Primary

8. Click OK.
9. Click Save.

Create an Alert Rule

1. Click Alerts in the left-hand menu.
2. Click New Alert Rule.
3. In the Condition section, click Select condition.
4. In the Configure signal logic pane:

- Click DTU Used in the list.
- In the Alert logic section, set the following values:
    - Threshold: Static
    - Operator: Greater than
    - Aggregation type: Average
    - Threshold value: 70
- Click Done.

5. In the Actions section, click Select action group.
6. In the Select an action group to attach to this alert rule pane:

- Click Create action group.
    - For both Action group name and Display name, enter "Email".
    - Click Next: Notifications.
    - Set Notification type to Email/SMS message/Push/Voice.
    - For Name, enter "Email".
    - In the Email/SMS message/Push/Voice pane, check Email and enter "test@test.com" in the box.
    - Click OK.
    - Click Review + create > Create.
7. In the Alert rule details section, enter an Alert rule name of "dtuused".
8. Click Create alert rule.
9. Click Manage alert rules.
10. Verify you see the new dtuused rule listed.

Create Charts

1. Navigate back to your SQL database page (you can click its name in the breadcrumb trail link at the top).
2. Click Metrics in the left-hand menu.
3. Click the pencil icon next to Chart Title to edit it.
4. In the box, enter "DTU Limit"
5. Set Metric to DTU Limit.
6. Ensure Aggregation is set to Avg.
7. Click New chart.
8. Click the pencil icon next to Chart Title to edit it.
9. In the box, enter "Data Space".
10. Set Metric to DTU space used percent.
11. Ensure Aggregation is set to Max.
12. Click Pin to dashboard.
13. Click Create new.
14. For Dashboard name, enter "Test".
15. Click Create and pin.
16. In the DTU Limit chart, click Pin to dashboard.
17. Leave it as the Test dashboard, and click Pin.

Note: It's important to pin the charts to the dashboard, as they'll disappear from the Metrics page when you navigate back there.

Create a Diagnostic Setting to Send Logs to Log Analytics

1. Navigate back to your SQL database page.
2. Click Diagnostic settings in the left-hand menu.
3. For Diagnostic setting name, enter "sqldatabase".
4. In the Category details section, check the boxes for:
   Errors
   Timeouts
   Basic
5. In the Destination details section, check the box to Send to Log Analytics workspace.
6. Click Save.

Review and Create a Log Query

1. Navigate back to your SQL database page.
2. Click Logs in the left-hand menu.
3. Note: For the Azure Log Analytics page to work properly, you must enable cookies for the site.
4. Enter the following in the query box:

```
// Avg CPU usage
// Avg CPU usage in the last hour by resource name.
//consistently high averages could indicate a customer needs to move to a larger SKU
AzureMetrics
| where ResourceProvider == "MICROSOFT.SQL" // /DATABASES
| where TimeGenerated >= ago(60min)
| where MetricName in ('cpu_percent')
| parse _ResourceId with * "/microsoft.sql/servers/" Resource  // subtract Resource name for _ResourceId
| summarize CPU_Maximum_last15mins = max(Maximum), CPU_Minimum_last15mins = min(Minimum), CPU_Average_last15mins = avg(Average) by Resource , MetricName
```

5. Click Run. There won't be any data that appears.
6. In the left-hand menu, click Query editor.
7. Enter the username and password you entered earlier when creating the SQL database server.
8. Click OK.
9. Enter the following in the query box:

```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
    FROM SalesLT.ProductCategory pc
    JOIN SalesLT.Product p
    ON pc.productcategoryid = p.productcategoryid;
```

10. Click Run. View the results below.
11. Delete the query from the box.

12. Enter the following query in the query box:

```
 INSERT INTO [SalesLT].[Product]
( [Name]
, [ProductNumber]
, [Color]
, [ProductCategoryID]
, [StandardCost]
, [ListPrice]
, [SellStartDate]
)
VALUES
('myNewProduct'
,123456789
,'NewColor'
,1
,100
,100
,GETDATE() );
```

13. Click Run.
14. Click Logs in the left-hand menu.
15. In the Queries History section, click Run.

    Note: This query may not show immediately in your lab. If you run the query and get no results, you likely need to let more time elapse in the environment for Azure to pick up on the activity.

Review Audit Logs and Find Storage Logs

1. Click Query editor in the left-hand menu.
2. Enter the username and password you entered earlier when creating the SQL database server.
3. Click OK.
4. Enter the following in the query box:

```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
    FROM SalesLT.ProductCategory pc
    JOIN SalesLT.Product p
    ON pc.productcategoryid = p.productcategoryid;
```

5. Click Run. View the results below.
6. Delete the query from the box.
7. Enter the following query in the query box:

```
INSERT INTO [SalesLT].[Product]
      ( [Name]
      , [ProductNumber]
      , [Color]
      , [ProductCategoryID]
```

```
        , [StandardCost]
        , [ListPrice]
        , [SellStartDate]
        )
    VALUES
        ('myNewProduct'
        ,123456789
        ,'NewColor'
        ,1
        ,100
        ,100
        ,GETDATE() );
```

8. Click Auditing in the left-hand menu.
9. Click View audit logs.
10. Click the hamburger menu icon in the upper left, and click All resources.
11. Click your storage account.
12. Click Containers.
13. Click the listed sqldbauditlogs.
14. Click through the folders listed.
15. Click the hamburger menu icon in the upper left, and click All resources.
16. Click your SQL database.
17. Click Logs in the left-hand menu.
18. Find the **Avg CPU usage** query, and click Run. This time, some data will appear.