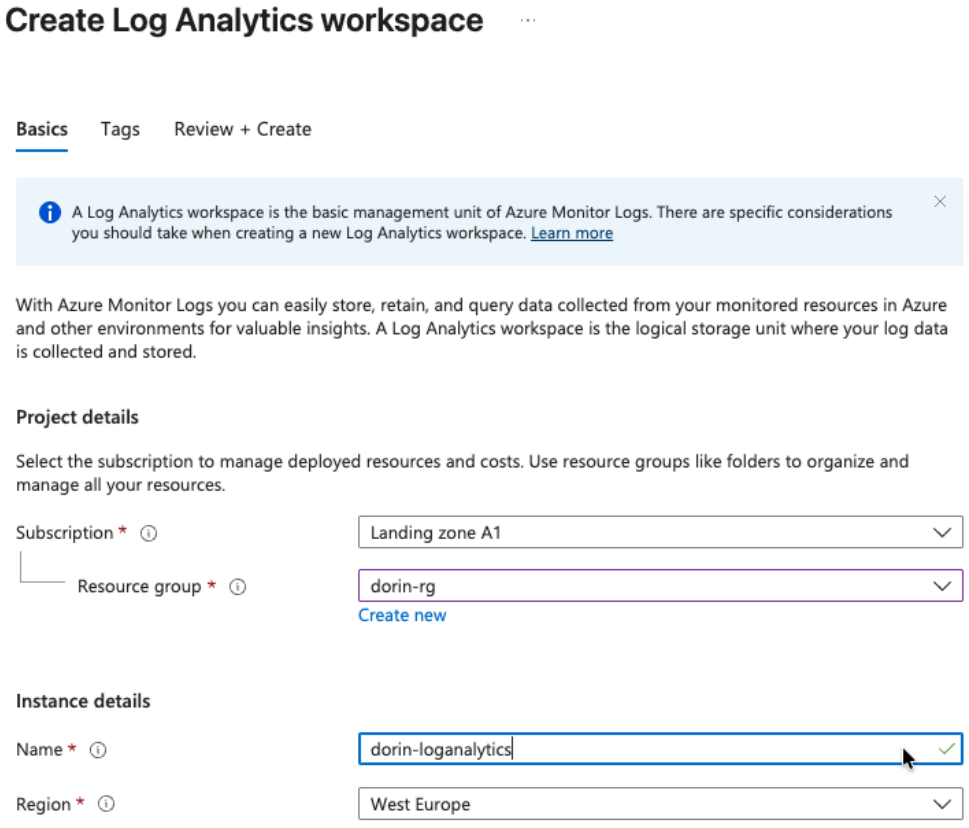
# Configure and manage Azure Monitor

Azure Log Analytics is a powerful service that allows you to query, visualize, and analyze logs from various Azure resources.

1. **Sign in to Azure Portal:**
    a. Navigate to Azure Portal https://portal.azure.com/
    b. Search for Log Analytics Workspace in the search bar
2. **Open 'Create a resource':**
    a. Click on the + Create a resource button located in the top-left corner of the dashboard
3. **Search for 'Log Analytics Workspace':**
    a. In the search box, type "Log Analytics Workspace" and select it from the results.
4. **Click 'Create':**
    a. On the Log Analytics Workspace page, click the Create button to start the workspace creation process.
5. **Configure Basic Settings:**
    a. **Subscription**: Current Subscription
    b. **Resource Group**: [**yourname**]-rg
    c. **Name**: Provide a unique name for your Log Analytics Workspace.
    d. **Region**: WestEurope



6. **Review + create:**
    a. Once you've filled out the necessary details, click the Review + create button. This will validate your configurations.
    b. After validation passes, you'll be presented with a summary of your configurations. Review them to ensure they're correct.
7. **Create Workspace:**
    a. Click on the Create button to begin the deployment process. The deployment might take a few minutes. Once completed, the workspace is ready for use.
8. **Access the Workspace:**
    a. Once deployment is successful, you can navigate to your resource group and select the Log Analytics workspace to view its dashboard and start ingesting logs from other Azure services.

**Integrate Services:**

- With the workspace ready, you can now integrate various Azure services to send their logs and metrics to the Log Analytics workspace. Integrations could include services like Virtual Machines, App Services, Azure Network Watcher, and more.
- Remember to manage and monitor the costs associated with Log Analytics, especially if you're ingesting a significant amount of data or retaining it for long periods.

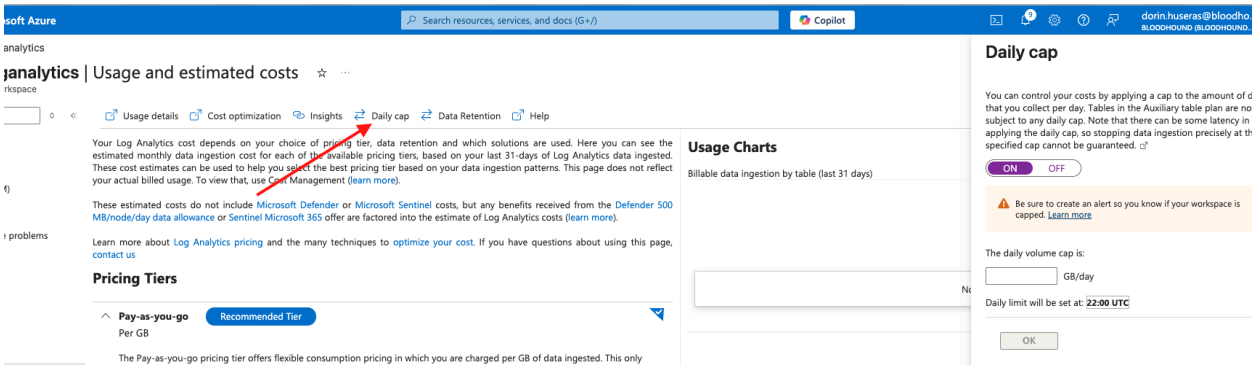
# Setting Daily Cap and Data Retention Policies in Azure Log Analytics

Setting a daily cap and adjusting data retention policies in Azure Log Analytics can help manage costs and comply with data retention standards that your organization or project may have. Here's an exercise to guide you through the process:

## Task 1: Set a Daily Cap for Data Ingestion

1. **Navigate to Log Analytics Workspace:**
    a. Log in to the Azure Portal.
    b. From the left sidebar, select "All resources" and find your Log Analytics workspace from the list. Click on its name to open it.
2. **Access Usage and Estimated Costs**:
    a. In the Log Analytics workspace's pane, under the Usage and estimated costs section, click on the Data volume management option.
3. **Set Daily Cap**:
    a. Find the Daily volume cap section.
    b. Adjust the slider or input a value to set the desired daily cap in GB.

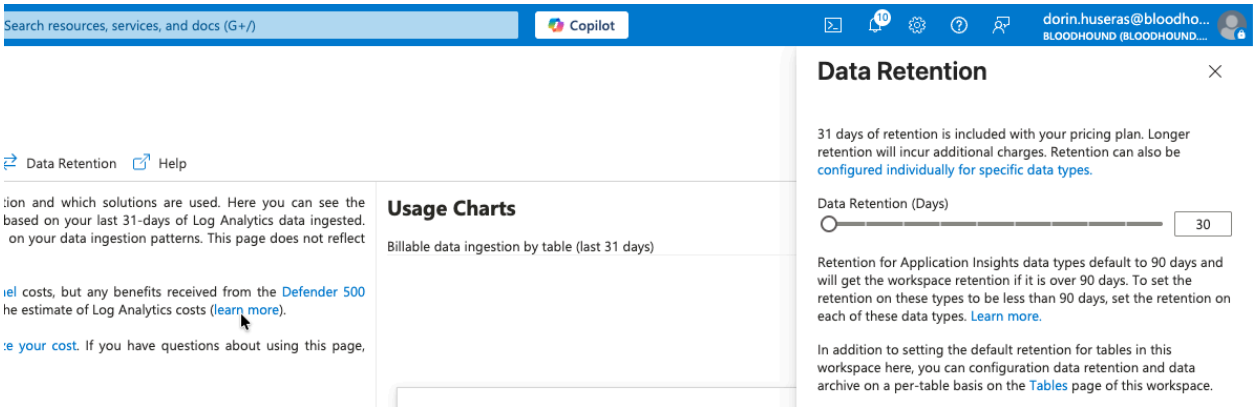
c. You can also configure an optional email alert to notify you when the data ingestion is nearing the daily cap.




4. **Once set, click Save to apply the daily cap**.

# Task 2: Adjust Data Retention Policy

1. **Navigate to Data Retention Settings**:
   a. While still in your Log Analytics workspace, find the Usage and estimated costs section.
   b. Click on the Data Retention option.




2. **Modify Retention Period**:
   a. A slider or dropdown will be available to adjust the data retention period. The retention can be set between 30 and 730 days based on the type of data and your needs.
   b. Move the slider or select the desired retention period.
3. **Click Save to apply the changes**.
4. **Review Settings**:
   a. After making changes, revisit both Data Volume Management and Data Retention sections to confirm that your configurations have been saved and applied correctly.

**Monitor Notifications**:
If you've set up email notifications for nearing the daily cap, you can test this by either nearing the cap or temporarily reducing the cap to a very low limit and generating some logs (make sure to revert back to avoid unnecessary data ingestion halts).

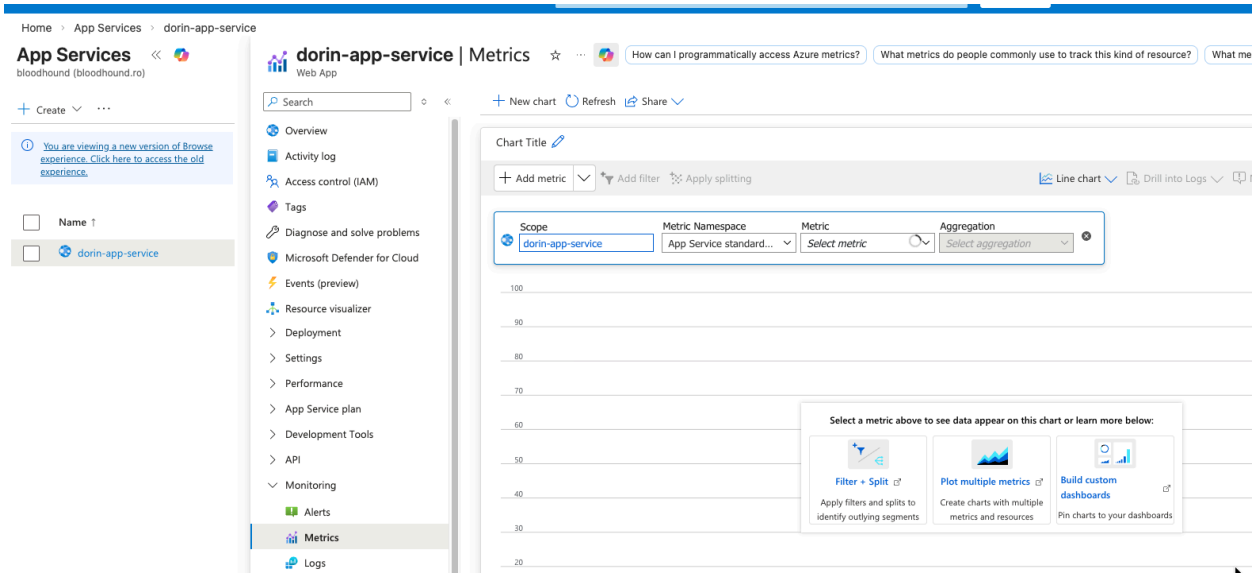
# Using Azure Metrics for Monitoring and Analysis

Azure Metrics provides real-time performance data about the Azure resources you're using. This exercise will help you familiarize yourself with Azure Metrics.

## Task 1: Accessing Azure Metrics

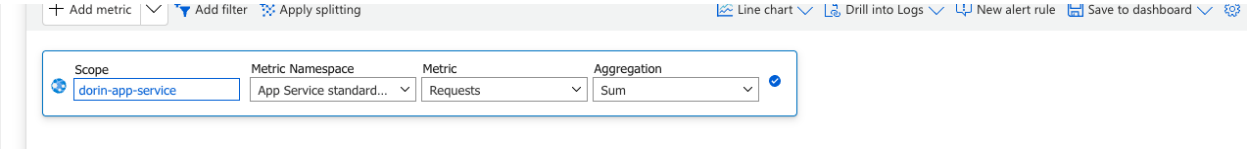
Navigate to Your Resource:

1. Log in to the Azure Portal.
    a. Find and click on your Web App resource.
2. Access Metrics:
    a. In the left-hand menu under the Monitoring section, click on Metrics.



# Task 2: Exploring Basic Metrics

1. Select Metric Namespace:
    a. Some resources have multiple metric namespaces. Use the dropdown to select one.
2. Choose a Metric:
    a. From the available metrics in the dropdown, select a metric of interest (e.g., "Requests").



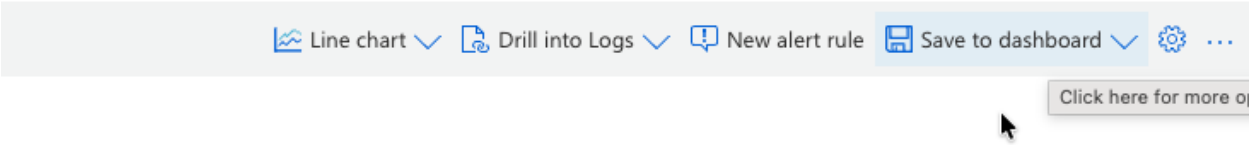

3. Adjust the Time Range and Granularity:
    a. You can adjust the time range (e.g., Last 30 minutes, Past week) to change the period of data displayed.
    b. Choose granularity (e.g., 1 minute, 1 hour) based on how detailed you want the data points to be.

# Task 3: Filtering and Splitting Metrics

1. Add a Filter:
    a. Click on the Add filter option. This lets you filter out data based on specific conditions.
    b. For instance, if you're viewing VM metrics, you might want to filter by a specific VM size or type.
2. Split Metrics:
    a. Use the Split by dropdown to divide the metric data based on a particular dimension, such as "Disk Type" or "Operating System".

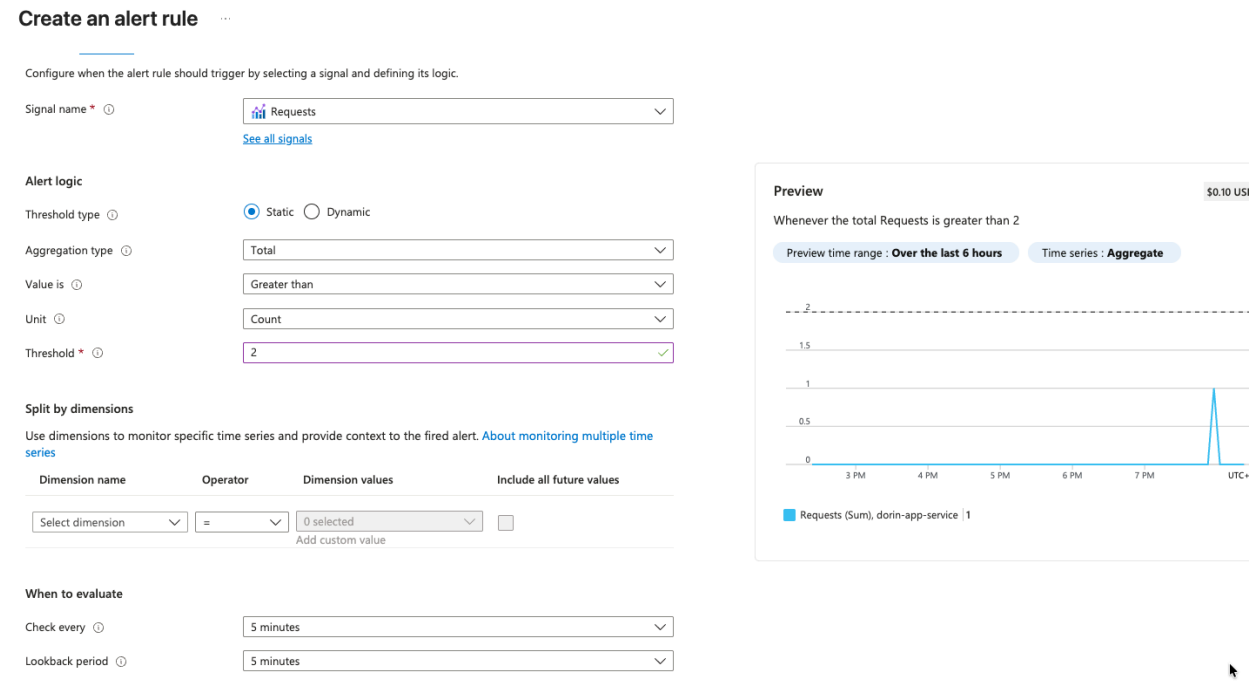
# Task 4: Pinning Metrics to Dashboard

1. Pin to Dashboard:
    a. Once you've configured the view to your satisfaction, click on the Pin to dashboard button at the top of the Metrics pane.
    b. Select the desired dashboard (or create a new one) and click Pin.
2. View Pinned Metric:
    a. Navigate to your Azure Dashboard and observe the metric chart you just pinned.




# Task 5: Creating an Alert Based on a Metric

1. Set New Alert Rule:
    a. Back in the Metrics pane, click on the New alert rule button.
2. Configure Alert Details:
    a. Set the Condition based on the metric's value.



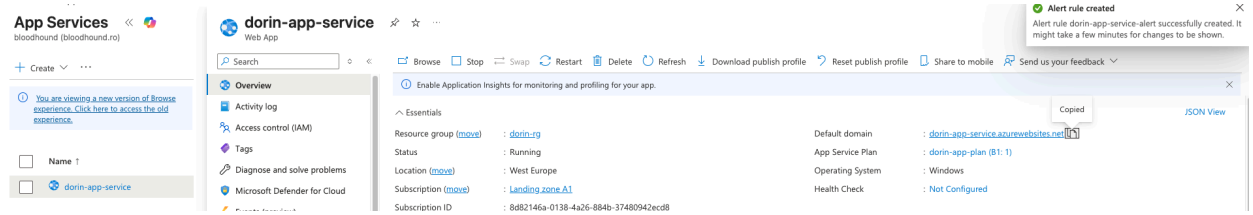

    b. Define Action Groups, which determine what actions will be taken when the alert is triggered (e.g., sending an email or SMS).

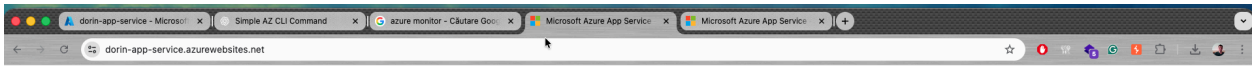

    c. Provide necessary details, including an alert name and description.
3. Create the Alert:
    a. Once you've filled out the required fields, click Create alert rule.

**Verification:**

**Trigger the Alert:**

- Go to your App service, get the URL to access it.
- Access the App Service URL multiple times to trigger the alert.





**Review the Dashboard:**

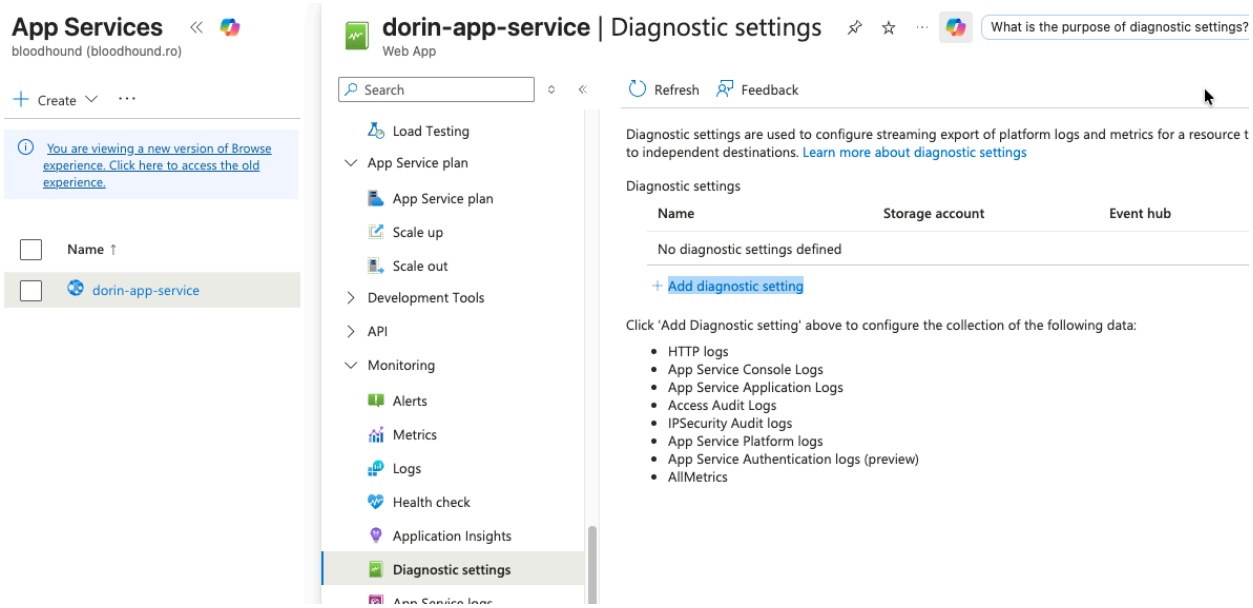
- Navigate to your Azure Dashboard to ensure the metric is visible and updating as expected.

# Using Azure Diagnostics for Monitoring and Analysis

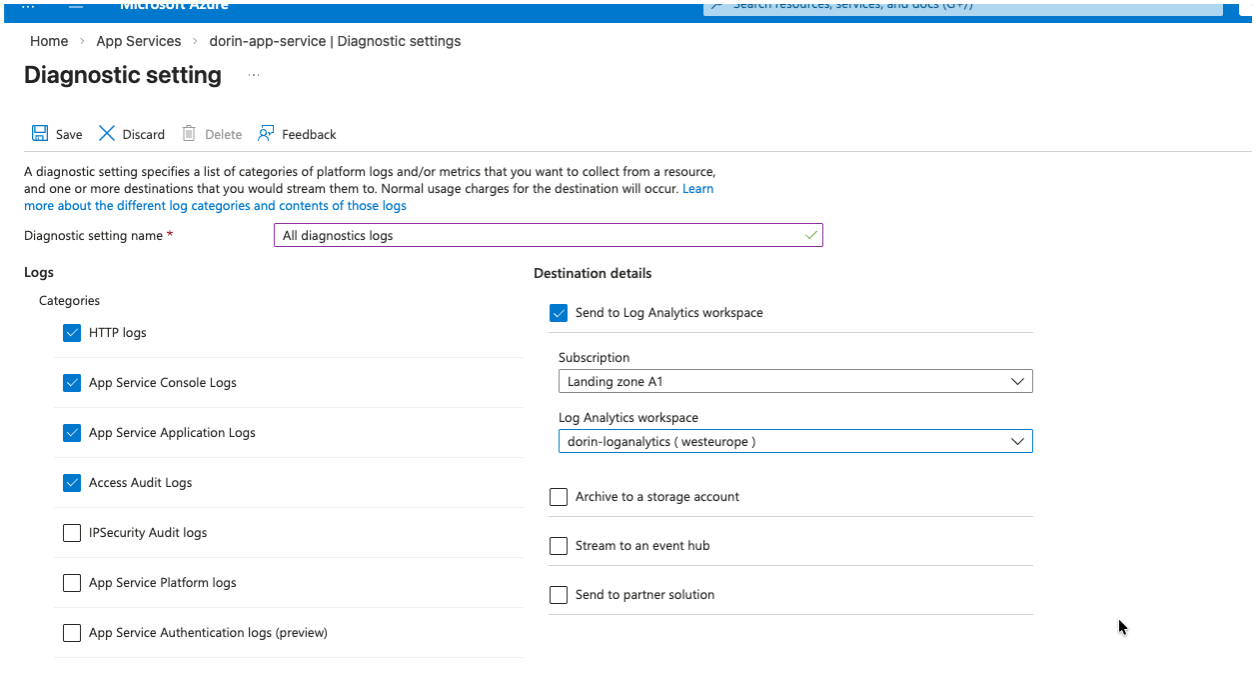
Azure Diagnostics provides detailed logging and telemetry for Azure resources, which can be invaluable for troubleshooting, monitoring, and performance tuning. This exercise will guide you through the process of enabling and using Azure Diagnostics for a specific resource, taking a Virtual Machine (VM) as an example.

## Task 1: Enabling Diagnostics

1. Navigate to Your App Service:
    a. Log in to the Azure Portal.
    b. Locate and select your **App Service** from your resource group.
2. Access Diagnostic Settings:
    a. In the App Service's left-hand menu under the **Monitoring** section, click on **Diagnostic settings**.




3. Enable Application Logging:
    a. Toggle the switch for the Application Logging you are interested in




4. Select the destination for the logs
    a. Toggle Send to **Log Analytics workspace**
    b. Select your existing log analytics workspace
5. Click the Save button to apply your changes.

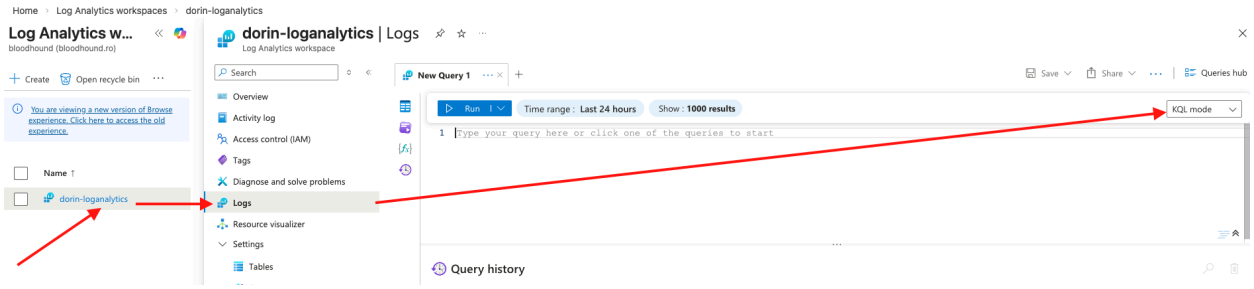
# Task 2: Generate Some Traffic

1. Access Your Web App:
    a. Open your web app in a browser to generate some user traffic.
    b. Create some scenarios you want to diagnose, like causing an error or accessing specific endpoints.
    c. https://[yourname]-service.azurewebsites.net/uraaaa
    d. https://[yourname]-service.azurewebsites.net/notfound

# Task 3 Run Kusto Queries for Web Application Logs:

Running Kusto Query Language (**KQL**) queries against web application logs in a Log Analytics workspace is an effective way to gather insights about your application's behavior. Let's go through the steps on how to query for specific HTTPS accesses on a certain path.

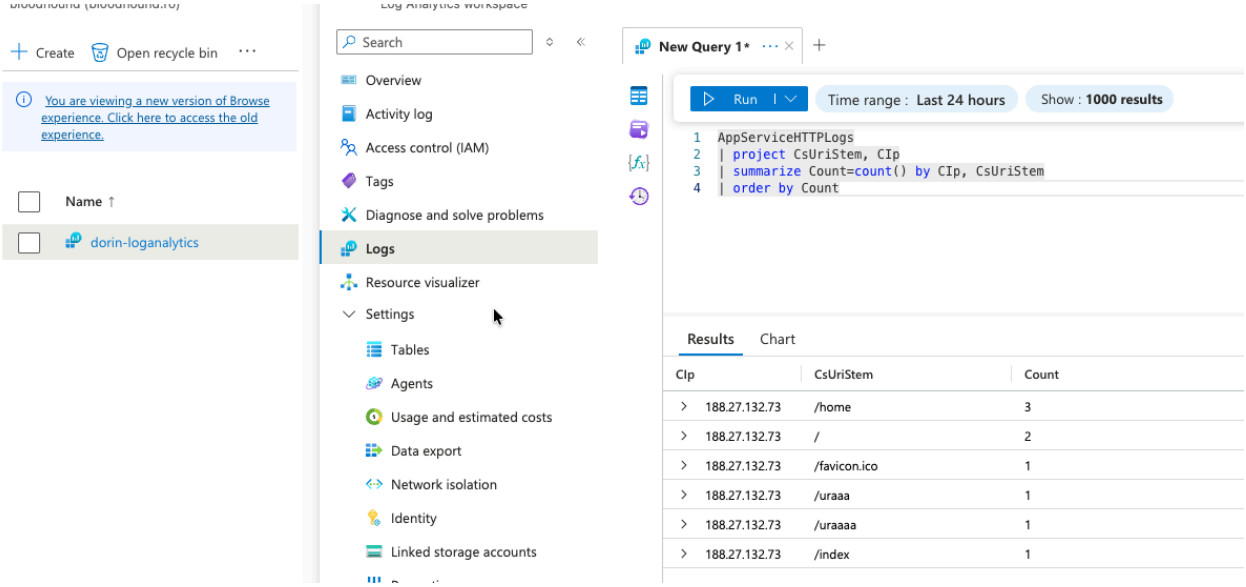
1. Navigate to the Log Analytics Workspace:
    a. Log in to the Azure Portal.
    b. Navigate to your **Log Analytics workspace** where your web application logs are being sent.
2. Access the Logs Section:
    a. Within your Log Analytics workspace, click on the **Logs** option in the left-hand menu.




3. Start Writing the Kusto Query:
    a. If you've integrated Azure Diagnostics with your Log Analytics workspace, the primary table you'll be querying is 'AppServiceAppLogs'. If not, you need to determine the appropriate table based on your log ingestion setup

```
AppServiceHTTPLogs
| project CsUriStem, CIp
| where CsUriStem has "/home"
| summarize Count=count() by CIp
| order by Count
```

- The **where** clause filters logs to only include records where the URL contains your specified path.
- The **project** clause determines which columns you want to see in the results. In this case, we're looking at when the request occurred, the URL, the result code (e.g., 200, 404, 500), the duration of the request, and the user ID.
- The **order by** clause sorts the results by timestamp in descending order, so you see the most recent logs first.




4. Run the Query:
    a. After writing your query, click the Run button to execute it.
5. Analyze the Results:
    a. Review the output to analyze accesses to the specified path. This can help you identify patterns, potential issues, or gather insights about user behavior.
6. Further Refinement:
    a. If you need to narrow down your results, you can add additional conditions to the where clause, like filtering by specific result codes or dates.
7. Search for 404 requests

# Task4: Create an alert from Kusto Query

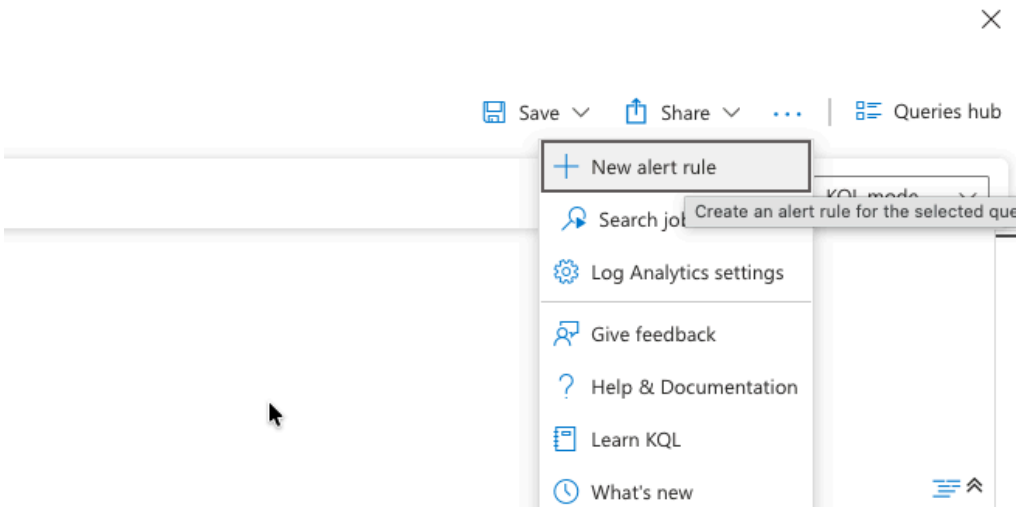
Creating an alert rule in Azure Monitor using a Kusto Query Language (KQL) query from a Log Analytics workspace allows you to be notified or take actions based on specific conditions in your data. Here are the steps to create such an alert rule:

1. Open Azure Portal:
    a. Navigate to the Azure Portal.
    b. Navigate to the Log Analytics workspace:

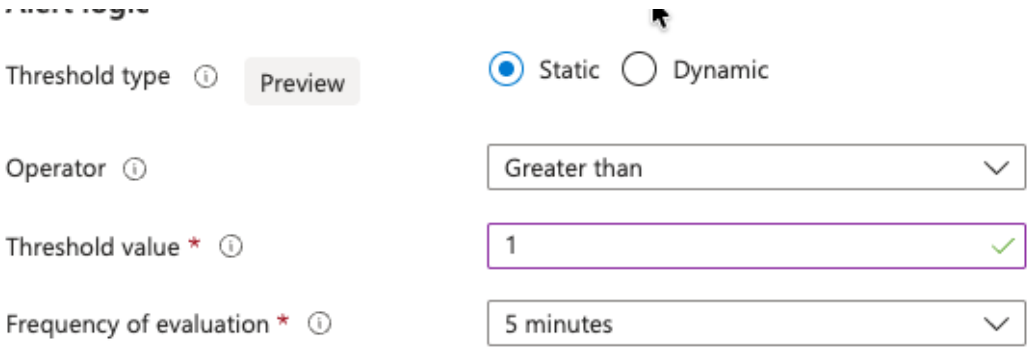
2. In the left navigation pane, select "All services".
    a. In the "All services" box, type "Log Analytics Workspaces".
    b. From the results, select "Log Analytics Workspaces".
    c. Click on the desired workspace where you have your logs.
3. Write your Kusto Query:
    a. From the Log Analytics workspace's overview page, click on the 'Logs' option.
    b. In the opened Kusto Query editor, write and test your KQL query.

```
AppServiceHTTPLogs
| project CsUriStem, CIp
| where CsUriStem has "/home"
| summarize Count = count() by CIp
| where Count > 5
| order by Count desc
```

4. Create an Alert Rule from the Query:
    a. Once you've validated your query, click on the "New alert rule" option (usually found above the query results).
    b. This will open the "Create rule" UI with your query populated.

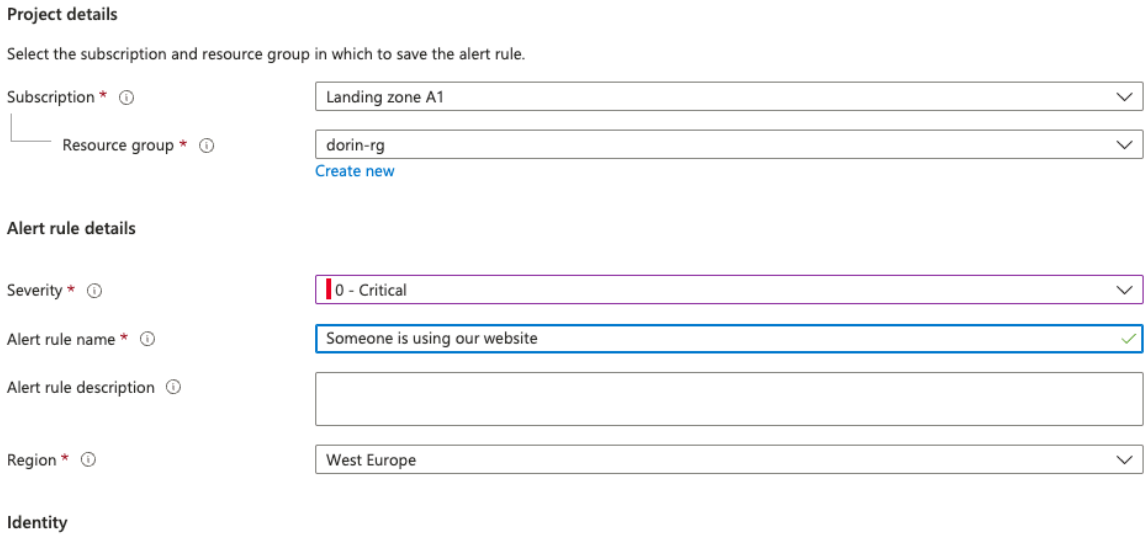



5. Configure the Alert **Condition**:
    a. Under "Condition", you'll see your query. Click on it to configure the condition further.
    b. Set the "Threshold value", evaluation based on frequency, and period.




6. Define the Alert **Actions**:
    a. Under the "Actions" section, click on "Select action group".
    b. Choose an existing action group or create a new one. An action group defines what actions to take (like sending an email or triggering a webhook) when the alert is fired.
    c. Use your previously created AG


7. Define Alert Rule **Details**:
    a. Provide an appropriate name for the alert rule.
    b. Choose a severity level from Sev0 (most critical) to Sev4 (least critical).




    c. Choose or create a new resource group.
    d. Define the alert rule description if needed.
8. Review and Create:
    a. Once everything is set, review your configurations.
    b. Click on the "Create" button to create the alert rule.
9. Testing the Alert (optional but recommended):

    a. Trigger the condition that you defined in your KQL query to ensure that the alert works as expected and that the desired actions (like sending an email) are executed.
    b. Access your App Service again via the provided URL
    c. Check your email
10. Monitoring the Alerts:
    a. Navigate back to the Log Analytics workspace.
    b. Under the "Alerts" section, you can view active and resolved alerts.