Monitor Cognitive Services

Azure Cognitive Services can be a critical part of an overall application infrastructure. It's important to be able to monitor activity and get alerted to issues that may need attention.

Clone the repository for this course

If you have already cloned **AI-102-AIEngineer** code repository to the environment where you're working on this lab, open it in Visual Studio Code; otherwise, follow these steps to clone it now.

1. Start Visual Studio Code.
2. Open the palette (SHIFT+CTRL+P) and run a **Git: Clone** command to clone the `https://github.com/MicrosoftLearning/AI-102-AIEngineer` repository to a local folder (it doesn't matter which folder).
3. When the repository has been cloned, open the folder in Visual Studio Code.
4. Wait while additional files are installed to support the C# code projects in the repo.

   > **Note**: If you are prompted to add required assets to build and debug, select **Not Now**.

Provision a Cognitive Services resource

If you don't already have one in your subscription, you'll need to provision a **Cognitive Services** resource.

1. Open the Azure portal at `https://portal.azure.com`, and sign in using the Microsoft account associated with your Azure subscription.
2. Select the **+Create a resource** button, search for *cognitive services*, and create a **Cognitive Services** resource with the following settings:
   - **Subscription**: *Your Azure subscription*
   - **Resource group**: *Choose or create a resource group (if you are using a restricted subscription, you may not have permission to create a new resource group - use the one provided)*
   - **Region**: *Choose any available region*
   - **Name**: *Enter a unique name*
   - **Pricing tier**: Standard S0
3. Select any required checkboxes and create the resource.
4. Wait for deployment to complete, and then view the deployment details.
5. When the resource has been deployed, go to it and view its **Keys and Endpoint** page. Make a note of the endpoint URI - you will need it later.

Configure an alert

Let's start monitoring by defining an alert rule so you can detect activity in your cognitive services resource.

1. In the Azure portal, go to your cognitive services resource and view its **Alerts** page (in the **Monitoring** section).
2. Select **+ New alert rule**
3. In the **Create alert rule** page, under **Scope**, verify that the your cognitive services resource is listed.
4. Under **Condition**, click **Add Condition**, and view the **Select a signal** pane that appears on the right, where you can select a signal type to monitor.
5. In the **signal type** list, select **Activity Log**, and then in the filtered list, select **List Keys**.
6. Review the activity over the past 6 hours.
7. Select the **Actions** tab. Note that you can specify an *action group*. This enables you to configure automated actions when an alert is fired - for example, sending an email notification. We won't do that in this exercise; but it can be useful to do this in a production environment.
8. In the **Details** tab, set the **Alert rule name** to **Key List Alert**.
9. Select **Review + create**.

10. Review the configuration for the alert. Select **Create** and wait for the alert rule to be created.

11. In Visual Studio Code, right-click the **03-monitor** folder and open an integrated terminal. Then enter the following command to sign into your Azure subscription by using the Azure CLI.

    Code Copy

    ```
    az login
    ```

    A web browser tab will open and prompt you to sign into Azure. Do so, and then close the browser tab and return to Visual Studio Code.

    > **Tip**: If you have multiple subscriptions, you'll need to ensure that you are working in the one that contains your cognitive services resource. Use this command to determine your current subscription.
    >
    > Code Copy
    >
    > ```
    > az
    > ac
    > sh
    > ```
    >
    > If you need to change the subscription, run this command, changing *<subscriptionName>* to the correct subscription name.
    >
    > Code Copy
    >
    > ```
    > az
    > account
    >  set -
    > -
    > subscri
    >
    > <subscr
    > ```

12. Now you can use the following command to get the list of cognitive services keys, replacing *<resourceName>* with the name of your cognitive services resource, and *<resourceGroup>* with the name of the resource group in which you created it.

    Code Copy

    ```
    az cognitiveservices account keys list --name <resourceName> --resource-group
    <resourceGroup>
    ```

The command returns a list of the keys for your cognitive services resource.

1. Switch back to the browser containing the Azure portal, and refresh your **Alerts page**. You should see a **Sev 4** alert listed in the table (if not, wait up to five minutes and refresh again).
2. Select the alert to see its details.

Visualize a metric

As well as defining alerts, you can view metrics for your cognitive services resource to monitor its utilization.

1. In the Azure portal, in the page for your cognitive services resource, select **Metrics** (in the **Monitoring** section).
2. If there is no existing chart, select **+ New chart**. Then in the **Metric** list, review the possible metrics you can visualize and select **Total Calls**.
3. In the **Aggregation** list, select **Count**. This will enable you to monitor the total calls to you Cognitive Service resource; which is useful in determining how much the service is being used over a period of time.

4. To generate some requests to your cognitive service, you will use **curl** - a command line tool for HTTP requests. In Visual Studio Code, in the **03-monitor** folder, open **rest-test.cmd** and edit the **curl** command it contains (shown below), replacing *<yourEndpoint>* and *<yourKey>* with your endpoint URI and **Key1** key to use the Text Analytics API in your cognitive services resource.

    Code Copy

```
curl -X POST "<yourEndpoint>/text/analytics/v3.1/languages?" -H "Content-Type: application/json" -H "Ocp-Apim-Subscription-Key: <yourKey>" --data-ascii "{'documents':[{'id':1,'text':'hello'}]}"
```

5. Save your changes, and then in the integrated terminal for the **03-monitor** folder, run the following command:

   Code Copy

   ```
   rest-test
   ```

The command returns a JSON document containing information about the language detected in the input data (which should be English).

1. Re-run the **rest-test** command multiple times to generate some call activity (you can use the **^** key to cycle through previous commands).
2. Return to the **Metrics** page in the Azure portal and refresh the **Total Calls** count chart. It may take a few minutes for the calls you made using *curl* to be reflected in the chart - keep refreshing the chart until it updates to include them.

More information

One of the options for monitoring cognitive services is to use *diagnostic logging*. Once enabled, diagnostic logging captures rich information about your cognitive services resource usage, and can be a useful monitoring and debugging tool. It can take over an hour after setting up diagnostic logging to generate any information, which is why we haven’t explored it in this exercise; but you can learn more about it in the Cognitive Services documentation.