Lab 2 - Instrumenting a Function App with Application Insights

Objective

In this lab, we will create a new function app, next step you will perform is to check the box to add App Insights, create one of more function in the app and your done.

Prerequisites

- Azure Subscription
- Application Insights
- Function App

Steps

To build the solution in this lab, you have to follow the steps described in this section. From a high-level view the steps are:

- Create a resource group with a name like **rg-azurethursday-<initials>-<lab_no>**.
- Create an instance of Application Insights with **AppInsights-<initials>-labs**
- Create a Function App with a name like **FunctionApp-<initials>**
- Add a function to the Function App
- Configure the function App with Application Insights using the instrumentation key from Application Insights instance.
- Explore Application Insights.

Lab duration: **30 minutes**.

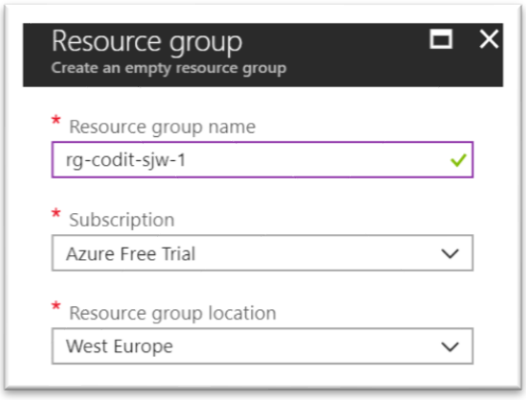
Created by Steef-Jan Wiggers, Microsoft Azure MVP, Codit Domain Lead Azure

Contact steef-jan.wiggers@codit.eu or twitter @steefJan

Step 1 - Create a resource group (optional)

The very first step in this lab is creating a resource group in your Azure subscription. A resource group is a logical container that groups all your resources. After the lab is finished, and you do not want to keep the resources, you can simply delete the resource group and the Azure Resource Manage will remove all the resources for you.

1. In the Azure Portal navigate to Resource Groups in the left menu pane.
2. Click the **+ Add**.
3. Provide a name for the resource group (**rg-azurethursday-<initials>-<lab_no>)**, specify a Subscription, and a location.

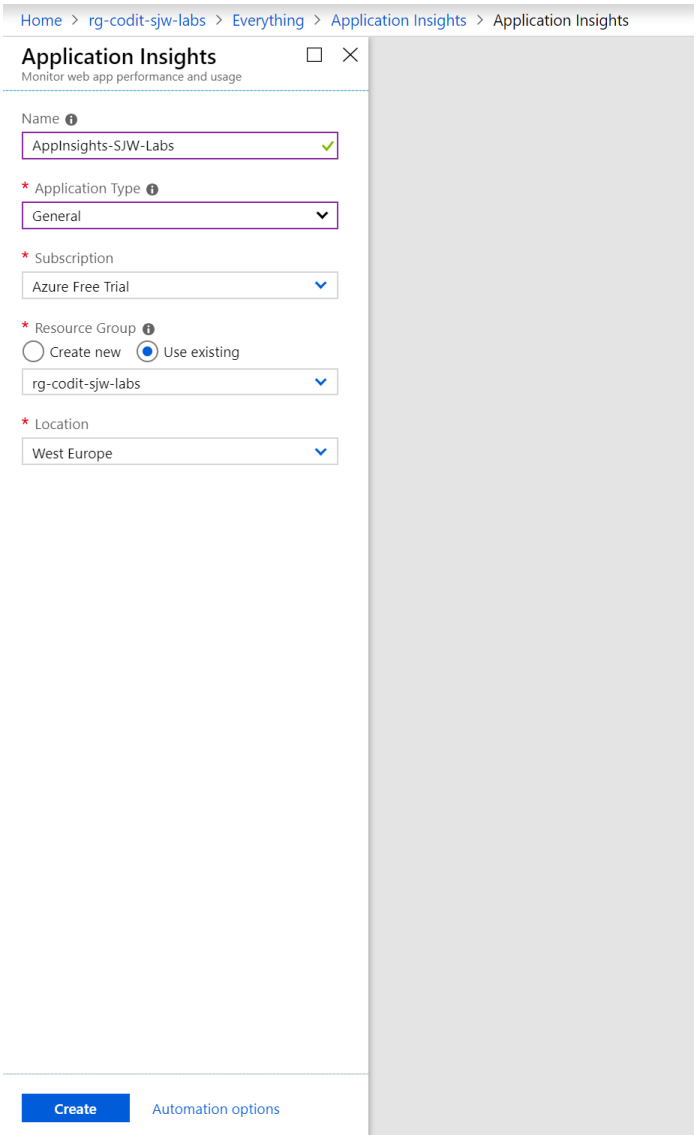


4. Finally, click **Create** and a resource group will be created for you.
5. In the top right corner, a pop-up will appear, which you can click to go to your resource group.

Step 2 – Provision an Application Insights Instance

Within the resource group, you can quickly add various types of Azure Resources. For this lab, we will need an instance of Application Insights.

1. Open the Azure Portal in your browser. If asked to log in, do so using your Microsoft account.
2. Go to your resource group. Click **+ Create a resource**, followed by **Application Insights** in search window**.** Select Application Insights and click **Create**.
3. Fill in the details like below:

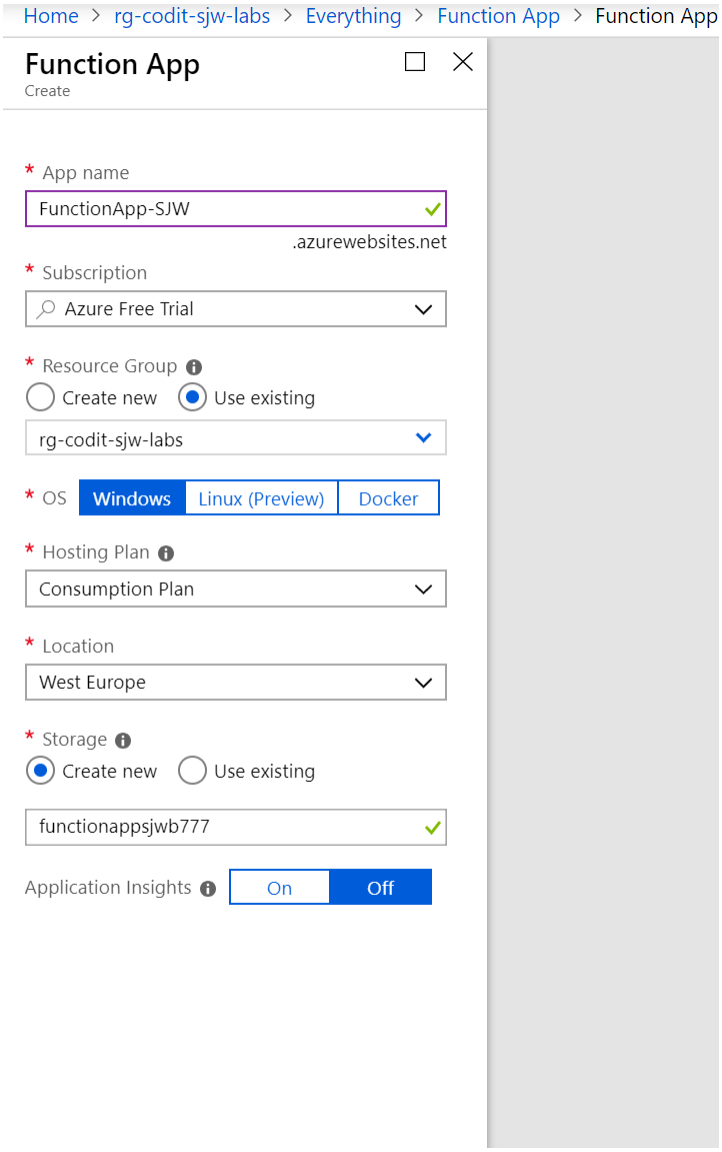


4. Click Create after specifying the details.
5. Once the instance is deployed, select it and copy the instrumentation key displayed in the overview pane.

Step 3 – Provision a Function App

The third step is to create an Azure Function App. In this exercise, you will create an Azure Function App using the Azure Portal.

1. Open the Azure Portal in your browser. If asked to log in, do so using your Microsoft account.
2. Go to your resource group. Click **+ Create a resource**, followed by **Function** App in search window. Select **Function App**.
3. Enter an app name that is unique within Azure. Under **Resource Group**, select **Exisiting** to create a resource group for the Function App. Choose the **Location** nearest you, and accept the default values for all other parameters, except ApplicationInsight – set it to **off**. Then click **Create** to create a new Function App.



The app name becomes part of a DNS name and therefore must be unique within Azure. Make sure a green check mark appears to the name indicating it is unique. You probably **won't** be able to use "functionslab" as the app name.

4. Wait until **Function App** is provisioned. Subsequently, click the created **Function App**. Click **Application Settings**.
5. Add App Setting – APPINSIGHTS_INSTRUMENTATIONKEY = {Instrumentation Key} (copied in step 2).

Application settings

APP SETTING NAME	VALUE	SLOT SETTING	DELETE
AzureWebJobsDashboard	DefaultEndpointsProtocol=https;AccountName=functionappsjwb777;AccountKey=J+vltceYZq4fhZKx1ilrtQjWBQ...	☐	✖
AzureWebJobsStorage	DefaultEndpointsProtocol=https;AccountName=functionappsjwb777;AccountKey=J+vltceYZq4fhZKx1ilrtQjWBQ...	☐	✖
FUNCTIONS_EXTENSION_VERSION	~1	☐	✖
WEBSITE_CONTENTAZUREFILECONNECTIONSTRING	DefaultEndpointsProtocol=https;AccountName=functionappsjwb777;AccountKey=J+vltceYZq4fhZKx1ilrtQjWBQ...	☐	✖
WEBSITE_CONTENTSHARE	functionapp-sjwb777	☐	✖
WEBSITE_NODE_DEFAULT_VERSION	6.5.0	☐	✖
APPINSIGHTS_INSTRUMENTATIONKEY	3e37641a-3139-4417-bf60-1da386a7d367	☐	✖

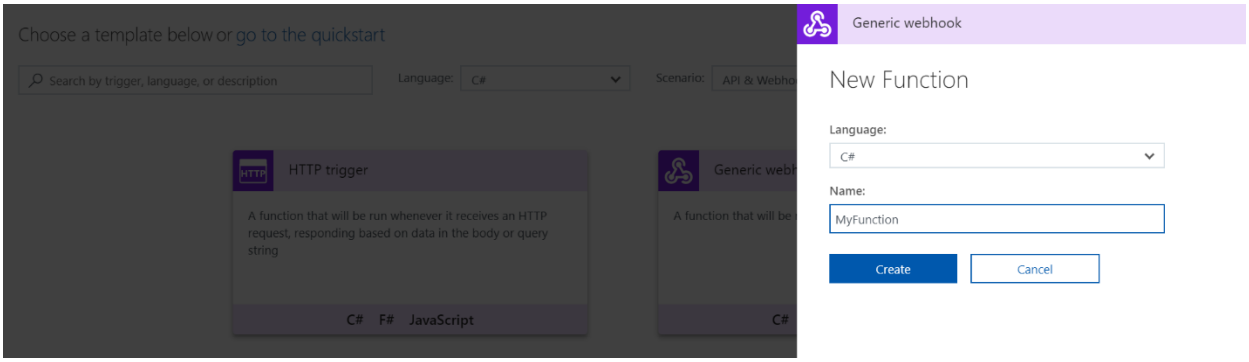
6. **Save** the Application Settings.

Step 4 – Add an Azure Function

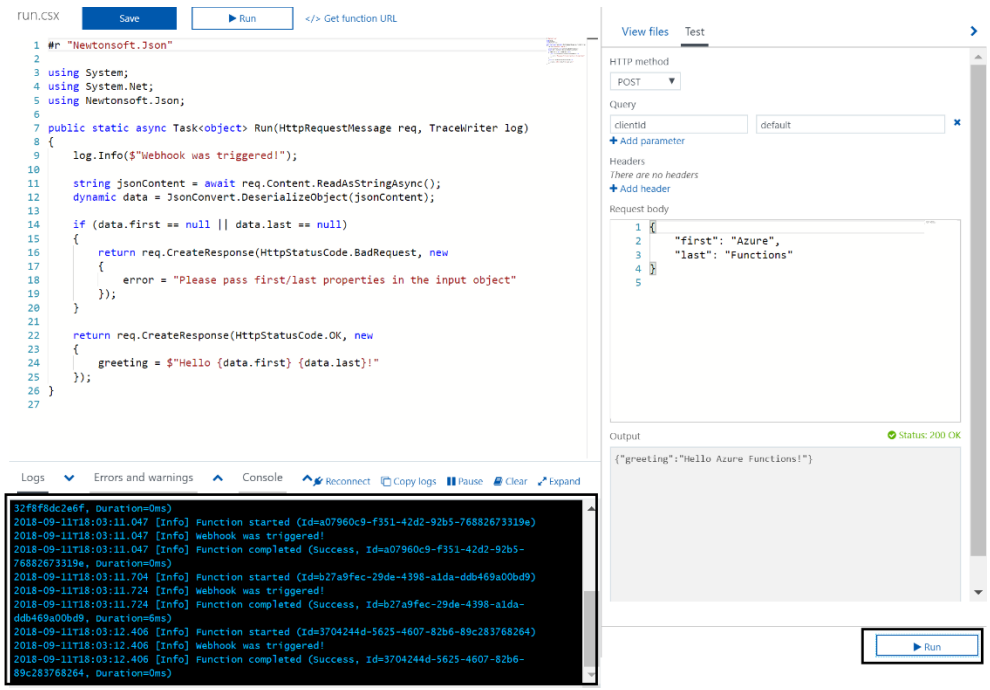
Once you have created an Azure Function App, you can add Azure Functions to it.

In this step, you will add a function to the Function App you created in step 3.

1. Click Azure Function App that you created in step 3.
2. Click the **+ sign** to the right of **Functions**. Set the language to CSharp, and then click **Custom** function.
3. Now you have to select a template for a function. Choose C# as language and API & WebHooks as scenario.
4. Next, choose the **Generic Webhook** template language C#.
5. Provide a name for the function and click Create.



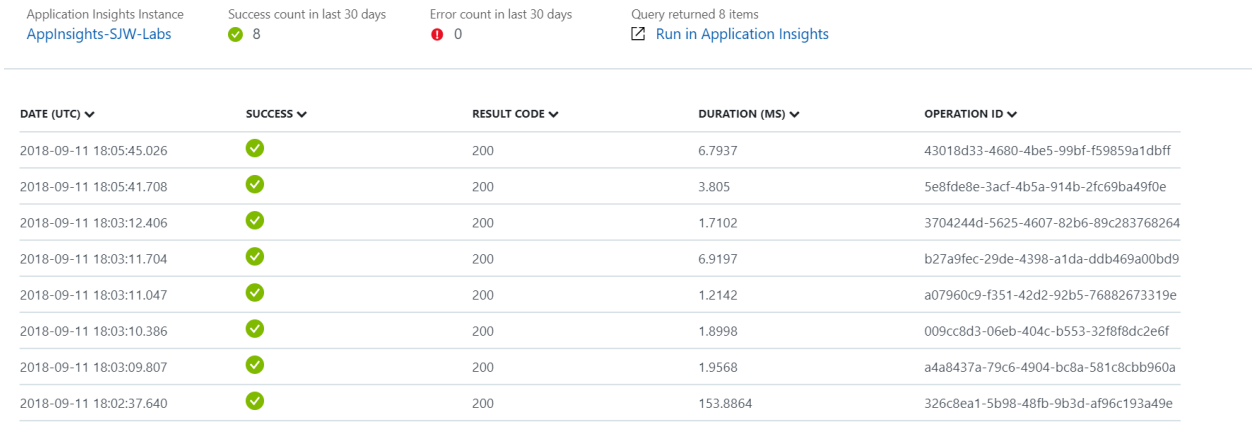
6. Leave the provided code as is and expand the test tab.
7. Hit run a few times.



Step 5 – Explore Application Insights

You will now explore the function runs with application insights.

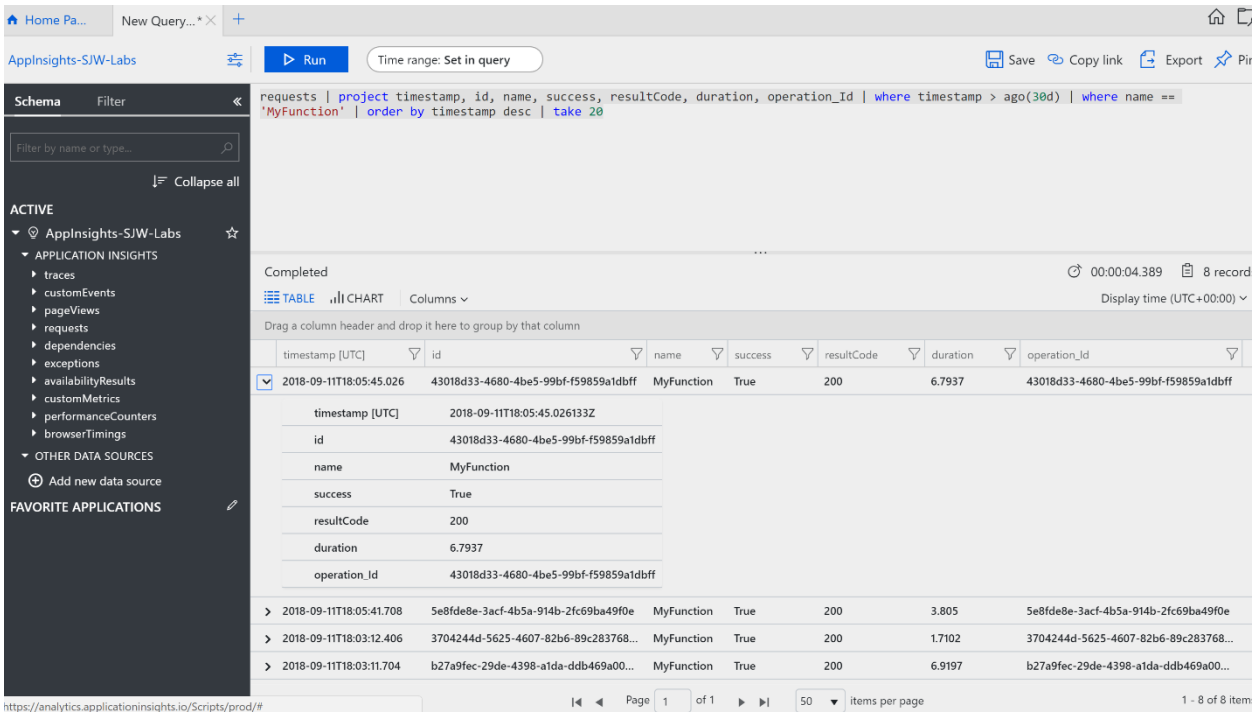
1. In the Function (**Step 4**) click Monitor.
2. You might not see results directly from the Function runs in **Step 4**. This may take up to 5 minutes.



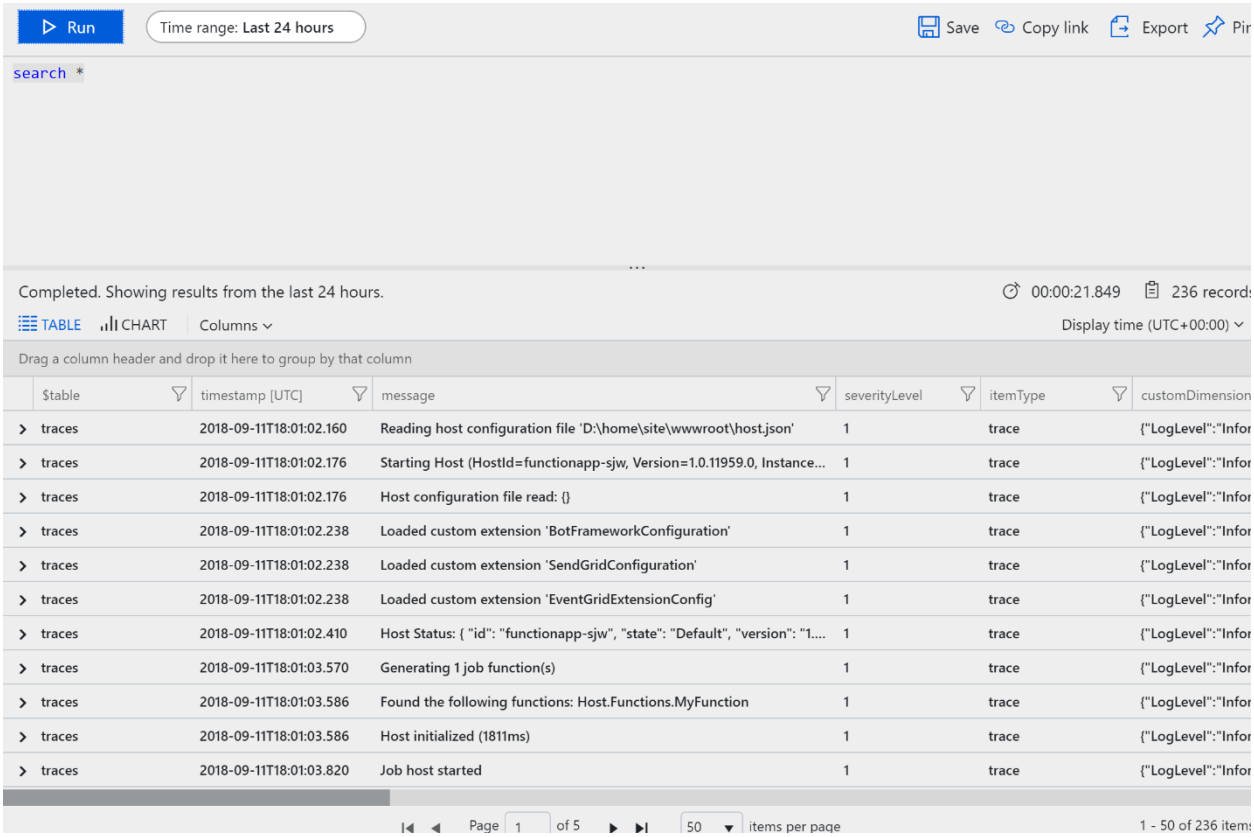
Application Insights Instance: AppInsights-SJW-Labs | Success count in last 30 days: 8 | Error count in last 30 days: 0 | Query returned 8 items: Run in Application Insights

DATE (UTC)	SUCCESS	RESULT CODE	DURATION (MS)	OPERATION ID
2018-09-11 18:05:45.026	✓	200	6.7937	43018d33-4680-4be5-99bf-f59859a1dbff
2018-09-11 18:05:41.708	✓	200	3.805	5e8fde8e-3acf-4b5a-914b-2fc69ba49f0e
2018-09-11 18:03:12.406	✓	200	1.7102	3704244d-5625-4607-82b6-89c283768264
2018-09-11 18:03:11.704	✓	200	6.9197	b27a9fec-29de-4398-a1da-ddb469a00bd9
2018-09-11 18:03:11.047	✓	200	1.2142	a07960c9-f351-42d2-92b5-76882673319e
2018-09-11 18:03:10.386	✓	200	1.8998	009cc8d3-06eb-404c-b553-32f8f8dc2e6f
2018-09-11 18:03:09.807	✓	200	1.9568	a4a8437a-79c6-4904-bc8a-581c8cbb960a
2018-09-11 18:02:37.640	✓	200	153.8864	326c8ea1-5b98-48fb-9b3d-af96c193a49e

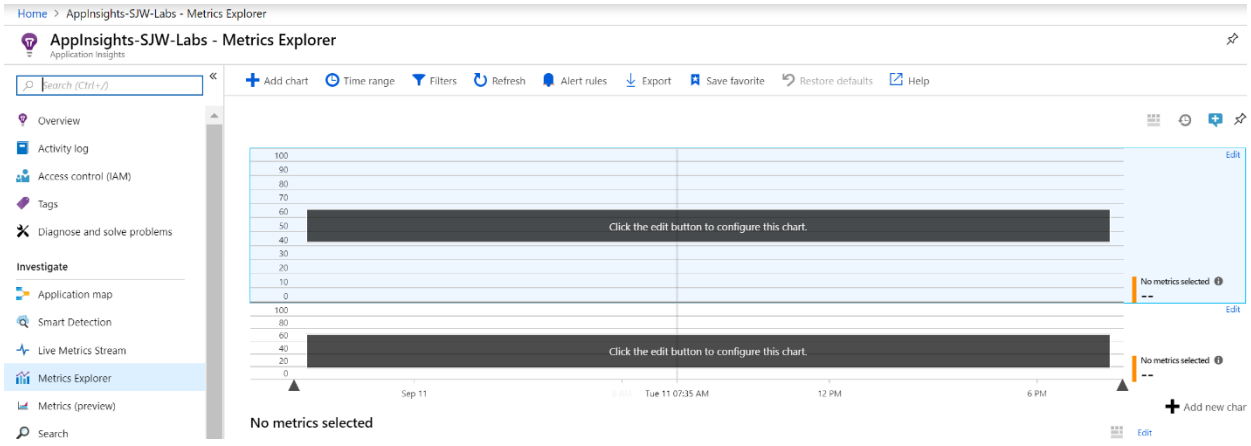
3. Click **Run** in **Application Insights**
4. An analytics window will appear.



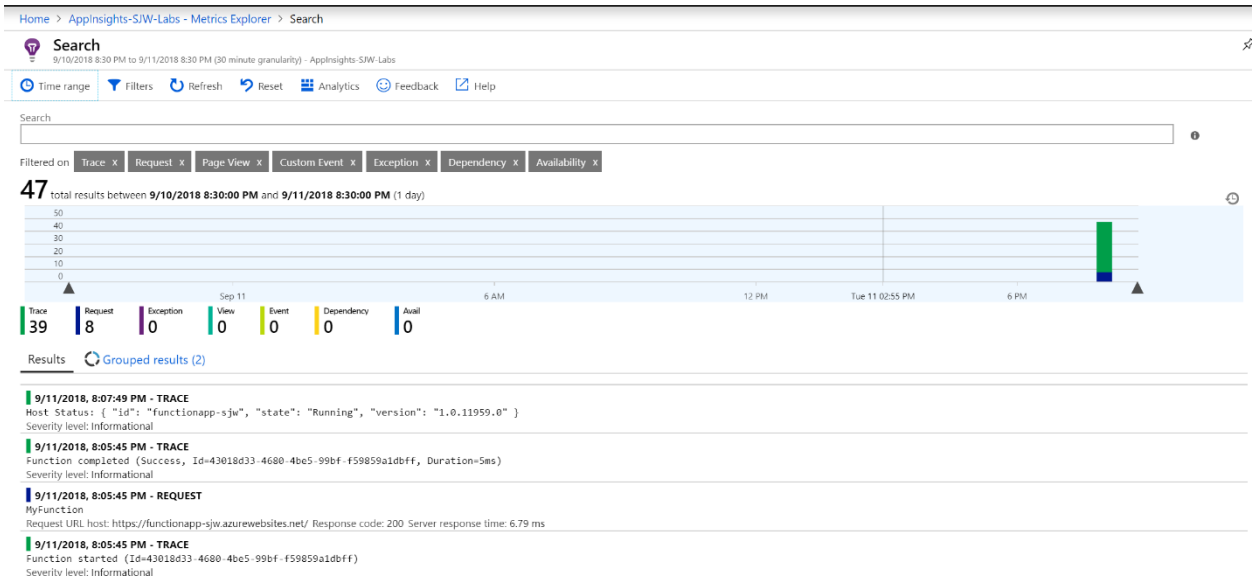
5. Create a new query like: **Search *** and click **Run**.
6. This query will take awhile. The response will contain all the traces of the Function runs.



7. Create a new query. By deleting the previous statement and enter : **requests** and hit run.
8. The response will the number of requests send to the function. The number will depend on how many times you have hit run in step 4.
9. Close the Analytics window.
10. Next, choose the Application Insights instance you created in step 2.
11. Choose the **metrics explorer**.
12. Click the bar that's indicates that metrics need to be configured.



13. You will now see the traces again.



14. Start building you custom chart.

15. Explore some of the other tabs like **Performance**, **Servers**, and **Application Map**.