Combine Azure Log Analytics with an AKS Cluster

In this lesson, we will combine Azure Log Analytics with an AKS cluster enabling the AKS addon.

WE'LL COVER THE FOLLOWING

Enable AKS addon for logging

Just like GKE (and unlike EKS), AKS comes with an integrated logging solution. All we have to do is enable one of the AKS addons. To be more precise, we'll enable the `monitoring` addon. As the name indicates, the addon not only fulfills the need to collect logs, but it also handles metrics. However, we are interested just in logs. I believe that nothing beats `Prometheus` for metrics, especially since it integrates with `HorizontalPodAutoscaler`. Still, you should explore AKS metrics as well and reach your own conclusion. For now, we'll explore only the logging part of the addon.

```
az aks enable-addons \
  -a monitoring \
  -n devops25-cluster \
  -g devops25-group
```

The output is a rather big JSON with all the information about the newly enabled `monitoring` addon. There's nothing exciting about it.

It's important to note that we could have enabled the addon when we created the cluster by adding `-a monitoring` argument to the `az aks create` command.

If you're curious about what we got, we can list the Deployments in the `kube-system` Namespace.

```
kubectl -n kube-system get deployments
```

The **output** is as follows.

```
NAME                  DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
heapster              1       1       1          1         1m
kube-dns-v20          2       2       2          2         1h
kubernetes-dashboard 1       1       1          1         1h
metrics-server        1       1       1          1         1h
omsagent-rs           1       1       1          1         1m
tunnelfront           1       1       1          1         1h
```

The new addition is the `omsagent-rs` Deployment that will ship the logs (and metrics) to Azure Log Analytics. If you `describe` it, you'll see that it is based on `microsoft/oms` image. That makes it the first and the only time we switched from **Fluentd** to a different log shipping solution. We'll use it simply because *Azure* recommends it.

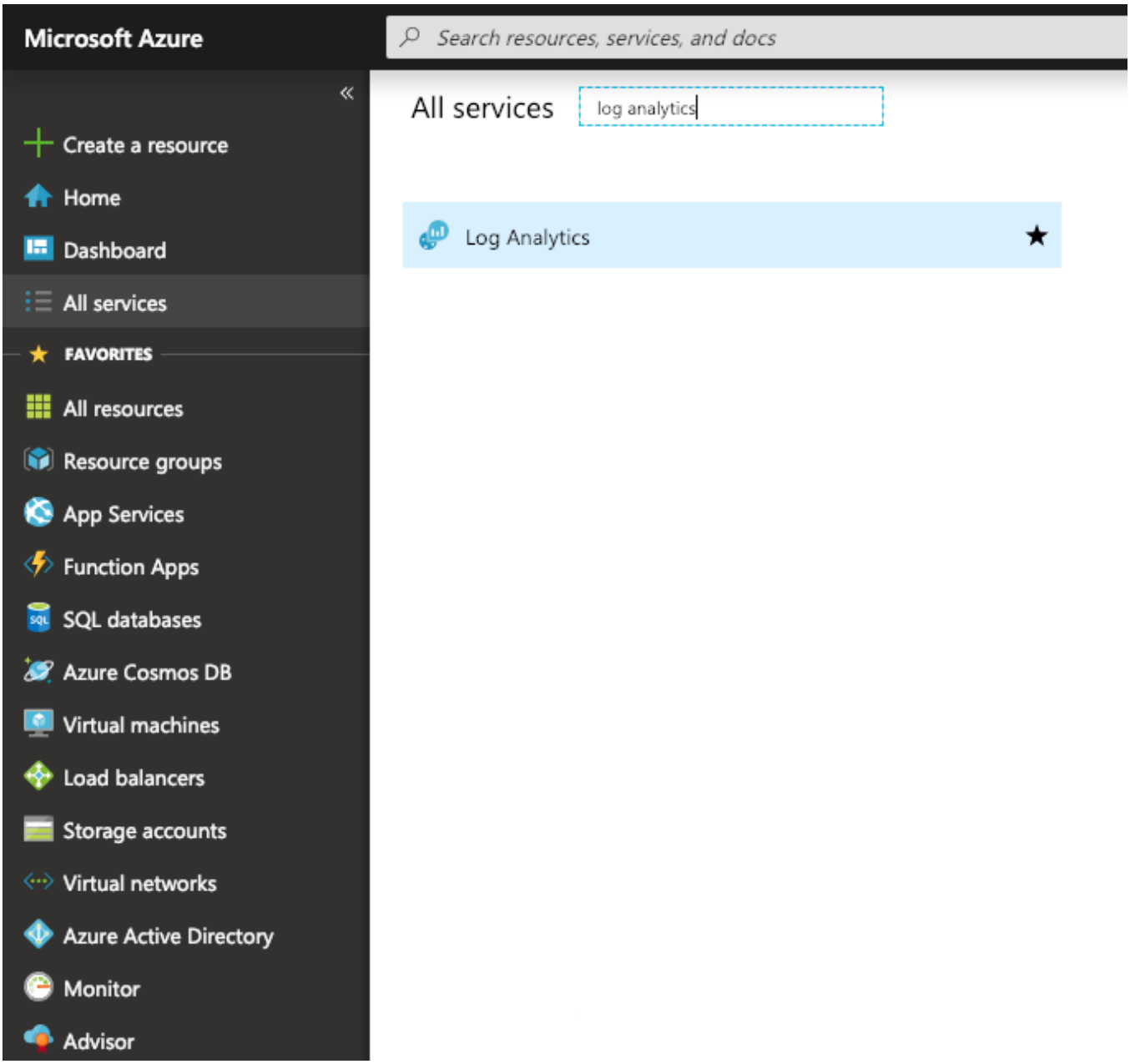
Next, we need to wait for a few minutes until the logs are propagated to **Log Analytics**. This is the perfect moment for you to take a short break. Go fetch a cup of coffee.

See Log Analytics in action

Let's open the Azure portal and see **Log Analytics** in action.

```
open "https://portal.azure.com"
```

Please click the *All services* item from the left-hand menu, type *log analytics* in the *Filter* field, and click the *Log Analytics* item.



Azure portal All services screen with log analytics filter

Q

The `omsagent-rs` Deployment will ship the logs (and metrics) to **Azure Log Analytics**.

COMPLETED 0%

1 of 1

Open the workspace

Unless you are already using **Log Analytics**, there should be only one active workspace. If that's the case, click it. Otherwise, if there are multiple workspaces, choose the one with the ID that matches the *id* entry of the `az aks enable-addons` output.

Click the menu item *Logs* in the *General* section.

Next, we'll try to limit the **output** entries only to those that contain `random-logger`. Please type the query that follows in the *Type your query here...* field.

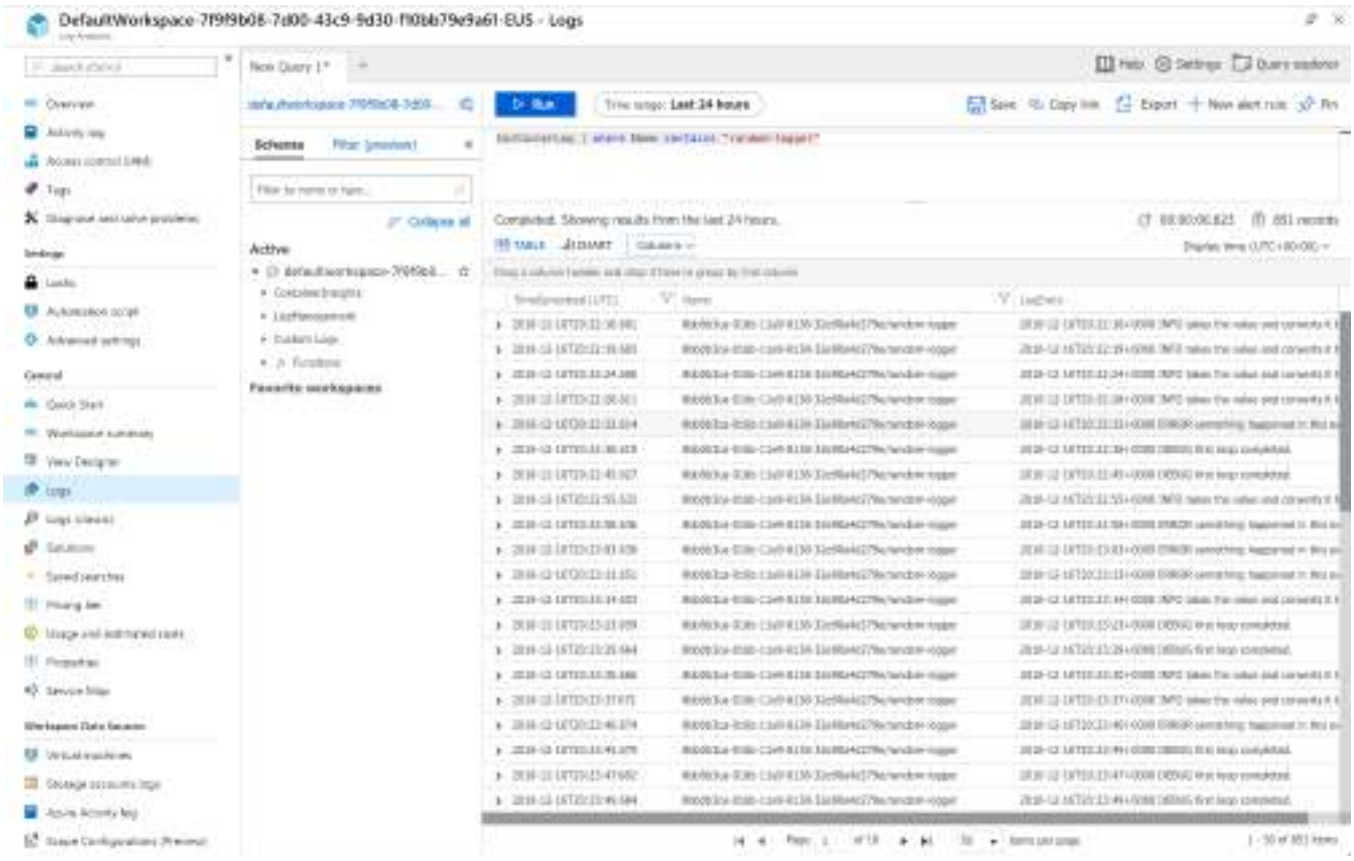
```
ContainerLog | where Name contains "random-logger"
```

Click the *Run* button, and you'll be presented with all the `random-logger` entries.

By default, all the fields are shown in the table, and many of them are either not used, or not very useful. The extra columns probably distract us from absorbing the logs, so we'll change the output.

It's easier to specify which columns we need, than which ones we don't. Please expand the *Columns* list, and click the *SELECT NONE* button. Next, select *LogEntry*, *Name*, and *TimeGenerated* fields and, once you're finished, contract the *Columns* list.

What you see in front of you are logs limited to `random-logger` and presented only through the three columns we selected.



Azure LogAnalytics screen with filtered entries

Explore Log Analytics features

I'll let you explore **Log Analytics** features on your own. Even though *Azure* portal's UI is not as intuitive as it could be, I'm sure you'll manage to get your way around it. If you choose to adopt AKS integration with Log Analytics, you should probably explore Log Analytics query language documentation that will help you write more complex queries than the one we used.

Disable the addon

Given that there is at least one more solution we should explore before we choose the one that fits your needs the best, we'll disable the addon. Later on, if you do like **Log Analytics** more than the alternatives, all you'll have to do is to enable it again.

```
az aks disable-addons \
  -a monitoring \
  -n devops25-cluster \
  -g devops25-group
```

In the next lesson, we will explore centralized logging through **Elasticsearch**, **fluentd** and **Kibana**.