Exercise - Configure monitoring for your application

100 XP

10 minutes

The success of Fruit Smoothies' marketing campaign is the ongoing performance of the ratings website. The performance is depended on your cluster's performance and relies on the fact that you can monitor the different components in your application, view logs, and get alerts whenever your application goes down or some parts of it fail. You can use a combination of available tools to set up alerting capabilities for your application.

In this exercise, you will:

- ✔ Create a Log Analytics workspace
- ✔ Enable the AKS monitoring add-on
- ✔ Inspect the AKS event logs and monitor cluster health
- ✔ Configure Kubernetes RBAC to enable live log data
- ✔ View the live container logs and AKS events

Create a Log Analytics workspace

Azure Monitor for containers is a comprehensive monitoring solution for Azure Kubernetes Service. This solution gives you insight into the performance of your cluster by collecting memory and processor metrics from controllers, nodes, and containers.

You use Log Analytics in Azure Monitor to store monitoring data, events, and metrics from your AKS cluster and the applications. First, you'll pre-create the Log Analytics workspace in your assigned environment resource group.

1. Sign in to Azure Cloud Shell with an Azure account.
2. You need a unique name for the workspace. Run the command below to generate a name similar to **aksworkshop-workspace-12345**.

   bash — Copy
   ```
   WORKSPACE=aksworkshop-workspace-$RANDOM
   ```
3. Run the `az resource create` command to create the workspace in the same resource group and region as your Azure Kubernetes Service (AKS) cluster. For example, **aksworkshop** in **East US**.

   bash — Copy
   ```
   az resource create --resource-type Microsoft.OperationalInsights/workspaces \
          --name $WORKSPACE \
          --resource-group $RESOURCE_GROUP \
          --location $REGION_NAME \
          --properties '{}' -o table
   ```

Enable the AKS monitoring add-on

Once the workspace is ready, you can integrate the Azure Monitor add-on and enable container monitoring on your AKS cluster.

1. You need to provide the resource ID of your workspace to enable the add-on. Run the following command to retrieve and store the workspace ID in a Bash variable named `WORKSPACE_ID`.

   Azure CLI — Copy
   ```
   WORKSPACE_ID=$(az resource show --resource-type Microsoft.OperationalInsights/workspaces \
      --resource-group $RESOURCE_GROUP \
      --name $WORKSPACE \
      --query "id" -o tsv)
   ```
2. Next, enable the monitoring add-on by running the `az aks enable-addons` command.

   bash — Copy
   ```
   az aks enable-addons \
      --resource-group $RESOURCE_GROUP \
      --name $AKS_CLUSTER_NAME \
      --addons monitoring \
      --workspace-resource-id $WORKSPACE_ID
   ```

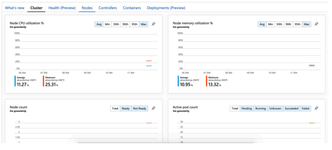
> **Note**
>
> It might take some time to establish monitoring data flow for newly created clusters. Allow at least 5 to 10 minutes for data to appear for your cluster.

Inspect the AKS event logs and monitor cluster health

We view utilization reports and charts for your cluster in the Azure portal by using Azure Monitor. Azure Monitor gives you a global perspective of all containers deployed across subscriptions and resource groups. From here, you can track containers that are monitored and those containers that aren't monitored. You can also inspect each container's statistics individually.

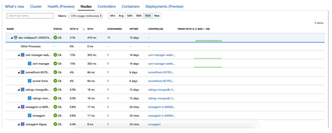
Let's look at the steps you need to take to get a detailed view of the health of nodes and pods in a cluster.

1. Switch to the Azure portal.
2. Select **Azure Monitor** from the left pane in the Azure portal.
3. Under the **Insights** section, select **Containers** to see a list of all clusters that you have access to.
4. Select the **Cluster** tab at the top of the view to check the cluster utilization. Notice how this view is again a high-level view that provides you a view on the cluster, nodes, controllers, and containers.



5. Select the **Health** tab at the top of the view to get a view on how the AKS infrastructure services of the cluster are doing.

6. Select the **Nodes** tab at the top of the view to get a detailed view of your nodes' health and pods in the cluster.



Configure Kubernetes RBAC to enable live log data

In addition to the high-level overview of your cluster's health, you can also view live log data of specific containers.

To enable and set permissions for the agent to collect the data, first, create a *Role* that has access to pod logs and events. Then you'll assign permissions to users by using *RoleBinding*.

What is role-based access control (RBAC)?

We use role-based access control (RBAC) in Kubernetes as a way of regulating access to resources based on the roles of individual users within your organization. RBAC authorization uses a set of related paths in the Kubernetes API to allow you to dynamically configure policies. The RBAC API defines four Kubernetes objects:

- Role
- ClusterRole
- RoleBinding
- ClusterRoleBinding

What is a Kubernetes Role?

The RBAC Role and ClusterRole objects allow you to set up rules that represent a set of permissions. The main difference between a Role and a ClusterRole is that a Role is used with resources in a specific namespace and ClusterRole is used with non-namespace resources in a cluster. You'll see how to define a ClusterRole later in the exercise.

What is a Kubernetes RoleBinding?

We use a role binding to grant the permissions defined in a role to a user or set of users. A role binding contains the list of users, groups, or service accounts, and a reference to the role being granted. Like the Role and ClusterRole, a RoleBinding grants permission within a specific namespace and the ClusterRoleBinding grants access to the cluster. You'll use a ClusterRoleBinding bind your ClusterRole to all the namespaces in your cluster.

In this exercise, you'll set up *Roles* and *RoleBindings* that aren't limited to a specific namespace. You can configure *Roles* and *RoleBindings* to grant permissions and bind roles to users across the entire cluster or to cluster resources outside a given namespace.

1. Create a file called `logreader-rbac.yaml` by using the integrated editor.

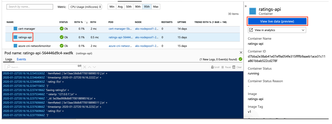
   bash — Copy
   ```
   code logreader-rbac.yaml
   ```
2. Paste the following text in the file.

   YAML — Copy
   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRole
   metadata:
       name: containerHealth-log-reader
   rules:
   - apiGroups: ["", "metrics.k8s.io", "extensions", "apps"]
     resources:
     - "pods/log"
     - "events"
     - "nodes"
     - "pods"
     - "deployments"
     - "replicasets"
     verbs: ["get", "list"]
   ---
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRoleBinding
   metadata:
       name: containerHealth-read-logs-global
   roleRef:
       kind: ClusterRole
       name: containerHealth-log-reader
       apiGroup: rbac.authorization.k8s.io
   subjects:
   - kind: User
     name: clusterUser
     apiGroup: rbac.authorization.k8s.io
   ```
3. To save the file, select `Ctrl+S`. To close the editor, select `Ctrl+Q`.
4. Apply the configuration by using the `kubectl apply` command.

   bash — Copy
   ```
   kubectl apply \
       -f logreader-rbac.yaml
   ```

View the live container logs and AKS events

1. Switch back to the AKS cluster in the Azure portal.
2. Select **Insights** under **Monitoring**.
3. Select the **Controllers** tab, and choose a container to view its live logs or event logs. For example, choose the **ratings-api** container. The new view allows you to debug the status of the container.



Summary

In this exercise, you created a Log Analytics workspace in Azure Monitor to store monitoring and logging data for your AKS cluster. You enabled the AKS monitoring add-on to enable the collection of data, and inspected the AKS cluster health. You then used Kubernetes RBAC to enable the collection of live logging data and then viewed live log data in the Azure portal.

Next, we'll take a look at scaling the Fruit Smoothies AKS cluster.

Next unit: Exercise - Scale your application to meet demand

Continue