# Create Azure Log Analytics Alerts

The following guidance walks through creating an alert for use with the ADPM environment. You will create a total of four, (4) alerts and one, (1) action group.

When an alert is triggered, an action group specifies the response actions to take. In this case, the action will be in the form of a webhook call to the ADPM alertForwarder service. The Alert Forwarder acts as a proxy receiving the webhook, normalizing the payload and performing a webhook call to the GitHub Actions repo.

In the Azure Log Analytics workspace, select ‘***Alerts’*** from the sidebar to create a new alert and action group, (see below).

1. From the Alert screen select ‘***Manage actions’***. Select ‘***Add action group***’ from upper-left corner.

A screenshot of a computer

Description automatically generated with medium confidence

1. Provide a name and display name for the action group and select ‘***Next: Notifications>***’ to continue.

A screenshot of a computer

Description automatically generated

1. Select ‘***Next: Actions >***’ to continue.

A picture containing text, monitor, screenshot, computer

Description automatically generated

1. On the ‘**Actions**’ screen select ‘Webhook’ for action type from the drop-down list. Provide a name for the webhook. Enter the alertForwarder server address for the webhook URI. The alertForwarder server address is available from the Terraform output. Select ‘***OK****’* and ‘***Next: Tags***’ to continue.

A picture containing text, monitor, indoor, screen

Description automatically generated

1. Select ‘***Next: Review + create’*** to continue.
2. Select ‘***Create***’ to create the action group.
3. From the Alert screen select ‘***Create Alert Rule’***

A screenshot of a computer

Description automatically generated

1. Verify the scope points to your existing analytics workspace.
2. Select ‘***Select Condition***’. On the ‘Configure signal logic’ pane, select ‘***Custom Log Search***’ and enter the alert search query. The following query is used to check BIG-IP ADC CPU utilization levels. In the event the alert is triggered during the evaluation period, a scale out webhook call will be sent. Refer to the end of this document for additional queries.

*F5Telemetry\_AVR\_CL*

*| where Entity\_s == 'SystemMonitor'*

*| where todouble(MaxCpu\_s) / 100 > 75*

*| where TimeGenerated >= ago(5m)*

*| project hostname\_s*

1. For the ‘Alert Logic’ section, specify the threshold level, (number of hits) and the evaluation period. For this example, (see below) the condition will be evaluated and triggered every 15 minutes if there are at least two, (2) hits where the CPU utilization rises above 75%. Select ‘***Done***’ to save the alert condition.

A screenshot of a computer

Description automatically generated with medium confidence

1. Back on the Alert screen, select ‘**Add action groups’** and select the previously created action group to associate with the newly created alert.
2. Under ‘Customize actions’ ensure the option to include custom json payload is checked and enter the following in the payload section:

*{"source": "azureLogs", "scaleAction": "scaleOutBigip","IncludeSearchResults": true}*

The custom payload will include the source, scaling action to take, and the query search results.

1. Provide a name for the alert rule, specify the resource group, and provide a severity level. Select ‘***Create new rule’***.

A screenshot of a computer

Description automatically generated

1. Repeat steps 7-13 referencing the below values to create three additional alerts.

* Rule Name - ‘***scale-in-bigip-alert’***
* Alert Query –

*F5Telemetry\_AVR\_CL*

*| where Entity\_s == 'SystemMonitor'*

*| where todouble(MaxCpu\_s) / 100 <= 10*

*| where TimeGenerated >= ago(5m)*

*| project hostname\_s*

* Webhook Payload -

*{"source": "azureLogs", "scaleAction": "scaleInBigip","IncludeSearchResults": true}*

* Rule Name - ‘***scale-out-workload-alert’***
* Alert Query –

*F5Telemetry\_AVR\_CL*

*| where Entity\_s == 'VipStat'*

*| where TimeGenerated >= ago(5m)*

*| where todouble(server\_concurrent\_conns\_s) >= 250*

*| project application\_s, hostname\_s*

*| join kind = inner*

*(*

*F5Telemetry\_AVR\_CL*

*| where isnotempty(pool\_name\_s)*

*| where TimeGenerated >= ago(5m)*

*| where pool\_name\_s !contains "telemetry"*

*| project hostname\_s, pool\_name\_s, application\_s)*

*on application\_s*

*| project-away application\_s, application\_s1*

* Webhook Payload -

*{"source": "azureLogs", "scaleAction": "scaleOutWorkload","IncludeSearchResults": true}*

* Rule Name - ‘***scale-in-workload-alert’***
* Alert Query –

*F5Telemetry\_AVR\_CL*

*| where Entity\_s == 'VipStat'*

*| where TimeGenerated >= ago(5m)*

*| where todouble(server\_concurrent\_conns\_s) < 50*

*| project application\_s, hostname\_s*

*| join kind = inner*

*(*

*F5Telemetry\_AVR\_CL*

*| where isnotempty(pool\_name\_s)*

*| where TimeGenerated >= ago(5m)*

*| where pool\_name\_s !contains "telemetry"*

*| project hostname\_s, pool\_name\_s, application\_s)*

*on application\_s*

*| project-away application\_s, application\_s1*

* Webhook Payload -

*{"source": "azureLogs", "scaleAction": "scaleInWorkload","IncludeSearchResults": true}*