# Create Elastic Kibana Alerts

The following guidance walks through creating an alert for use with the ADPM environment. You will create a total of two, (2) alerts. One alert will be triggered when the CPU utilization of BIG-IP(s) in the scale group exceed a level necessitating a scaling out operation. The other alert will be triggered when CPU utilization is low enough to warrant a scaling in operation.

When an alert is triggered a webhook call will be made 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.

The following guidance assumes you have access to a running ELK stack. Alert creation will be performed via the Kibana UI.

1. **Create Index Pattern** – Once telemetry data is configured and streaming to the ELK stack, (via a LogStash pipeline) you will need to create an index pattern to facilitate queries and alerts. From the side menu, select ‘***Kibana***’ 🡪 ***‘Index Patterns’.*** If the provided LogStash config file, (located in ADPM repository ‘scripts’ folder). All streaming entries will be indexed using – *‘f5-%{+YYYY.MM.dd.hh.mm}*’. Refer to sample ‘*logstash.conf’* file below. Enter ‘***f5-\****’ for the index pattern.

input {

http {

port => 8080

}

}

filter {

json {

source => "message"

}

mutate {

add\_field => { "myMaxCpu" =>" %{MaxCpu}"}

add\_field => { "myCurCons" =>" %{server\_concurrent\_conns}"}

}

mutate {

convert => { "myMaxCpu" => "integer" }

convert => { "myCurCons" => "integer" }

}

}

output {

elasticsearch {

hosts => ["https://6c5b5ef73a914a079257148492e06a99.us-central1.gcp.cloud.es.io:9243"]

user => "elastic"

password => "vn2d7xYjyXBJ8wDnhHGRIBad"

codec => json

index => "f5-%{+YYYY.MM.dd.hh.mm}"

}

}

Graphical user interface, text, application, email

Description automatically generated

1. **Add logging Columns** - From the side menu, select ‘***Observability*’** 🡪***‘Logs’****.* On the Logs main page, select the ‘***Settings***’ tab at the top of the page.
2. In the ‘*Indices*’ section add the log index value previously specified, ( *f5-\** ). You can leave the existing indices or remove.

Graphical user interface, application

Description automatically generated

1. In the ‘*Log Columns’ section search for and add the additional fields noted below.*
   * Hostname
   * Entity
   * application
   * pool\_name
   * myCurCons
   * myMaxCpu

These are the minimum fields required for monitoring and alerting the ADPM system. Two of these fields, (myCurCons &\* myMaxCpu) are derived from standard TS fields and created during the LogStash data ingestion.

1. Select ‘***Apply***’to save field updates.Graphical user interface

   Description automatically generated
2. From the upper-right of the screen select ‘***Alerts***’ 🡪 ‘***Create Alert’.***
3. Reference the below screenshot to create an alert that will trigger when a BIG-IP’s CPU utilization rises above 75% for fifteen, (15) minutes.

Graphical user interface, text, application

Description automatically generated

1. Reference the below screenshot to specify a webhook call that is triggered via the alert. The webhook payload, (example below) includes the source, scaling action, and hostname of the triggered device. The Terraform output includes the alertForwarder service endpoint. Specify this endpoint for the webhook destination. The alertForwarder service will receive the payload, normalize and proxy a call to the GitHub action workflows.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Repeat steps 6-8 referencing the screenshot below to create an additional alert for scaling in of BIG-IP devices.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated