EMAIL ALERT IF THE AJP COUNT IS HIGH

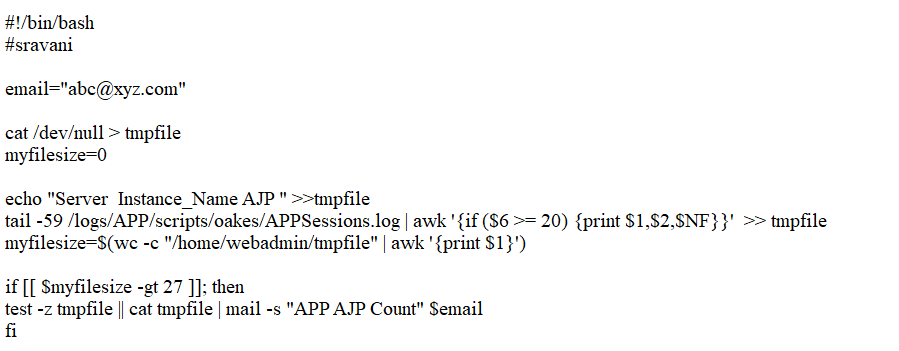
**Problem statement:**

We see servers with High AJP counts reaching to high level at certain period, then an alert must be for high CPU utilization or when it crosses the threshold limit. In such cases we perform bounce. So, as it is causing a delay to get notified and then bounce.

**Idea description:**

To avoid such issue, we can create an email alert prior to specific limit as a condition and then we can perform bounce by as soon as the alert triggers and we bounce at peak position. But if the bounce is done at the email alert only then the time can be minimized.

SCRIPT for AJP that are running on JBOSS:



Here in the above script,

* **email** must be the Email of the group to whom the Email alert must be sent.
* We are using a **tmpfile** , to append the data that we get from the APPSessions.log file.
* **APPSession.log** file is the output file that has all the fields of Server name , JVM name, CPU % and AJP counts etc which is generated by a **APPSessions.sh** file.
* **APP** is to be replaced with the application id that’s running on JBOSS.
* Now we are using tail command from the APPSessions.log file and Checking if any of the AJP counts are greater than 20 and if so, it will append the server list and AJP count to tmpfile. If not, it won’t alert any email as the counts are normal.
* We are considering a file where we append the Server name, AJP count and we are comparing if the file size is greater than the a particular value as we have empty text and fields sometimes, then if it is also satisfied then we get an email Alert with APP Server names and AJP count.
* As the APPSessions.log is generated from APPSession.sh, Let us see the AJP generation in APPSession.sh

A screenshot of a cell phone

Description automatically generated

So, from this part of the APPSession.sh we get the AJP counts to our APPSession.log

Note : APP = the application id that you are using on the JBOSS.