to get real time information about performance we can use StackDriver monitoring.

We can create an alert policy in stackdriver .

Stackdriver workspaces are associated with projects, not organizations or compute resources such as App Engine projects or Kubernetes engine clusters.

Latency , traffic , saturation , errors .

Latency : time to complete a request , distinguish failed and successful requests .

Traffic : measure of demand on system , https request / second, transaction / second

Error : rate of request that fail , explicit , implicit

Error code : failed / wrong response /

Saturation : measure of capacity in use , resources that are most constrained , predictive eg : disk fill in 1 hour .

Cloud monitor : collect metric from gcp , aws , azure and other cloud resources .

Dashboard and visuals , alerting and anomaly reporting , predefined and custom metrics . monitoring agent .

Dashboard :

Metrics explore : allow us to look at resources metrics , history of cpu usage of particular time .

alerting :

Uptime checks : lightweight .

Cloud logging : central repo for logs

System and app logs , managed service , uses google customized fluentd agent retain log for 30 day , stream log to pub sub for 3rd party tools , analytics with big query .

Log viewer :

Metrics editor :

This is a use case for Stackdriver Monitoring which collects and displays metrics on resource utilization such as CPU, memory, and network utilization. Stackdriver Debugger is used to analyze code but does not help with performance. Developers have already reviewed logs so Stackdriver Logging will not add more information. The Linux top command shows the top resource-consuming process and might help identify processes consuming CPU resources but not at the detailed level of Stackdriver Monitoring.