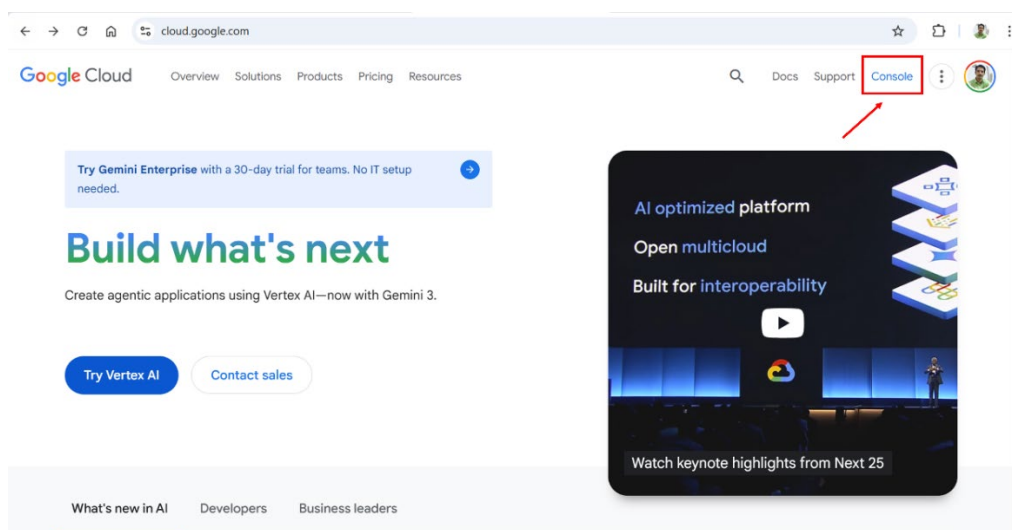


CLOUD COMPUTING LABORATORY

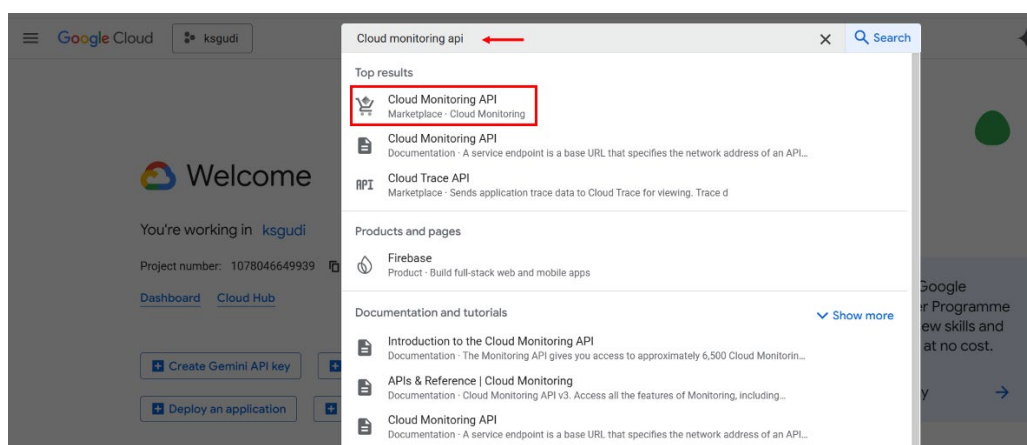
Experiment – 9:

Discover how Cloud Monitoring help in tracking and analyzing the performance and health of cloud resources.

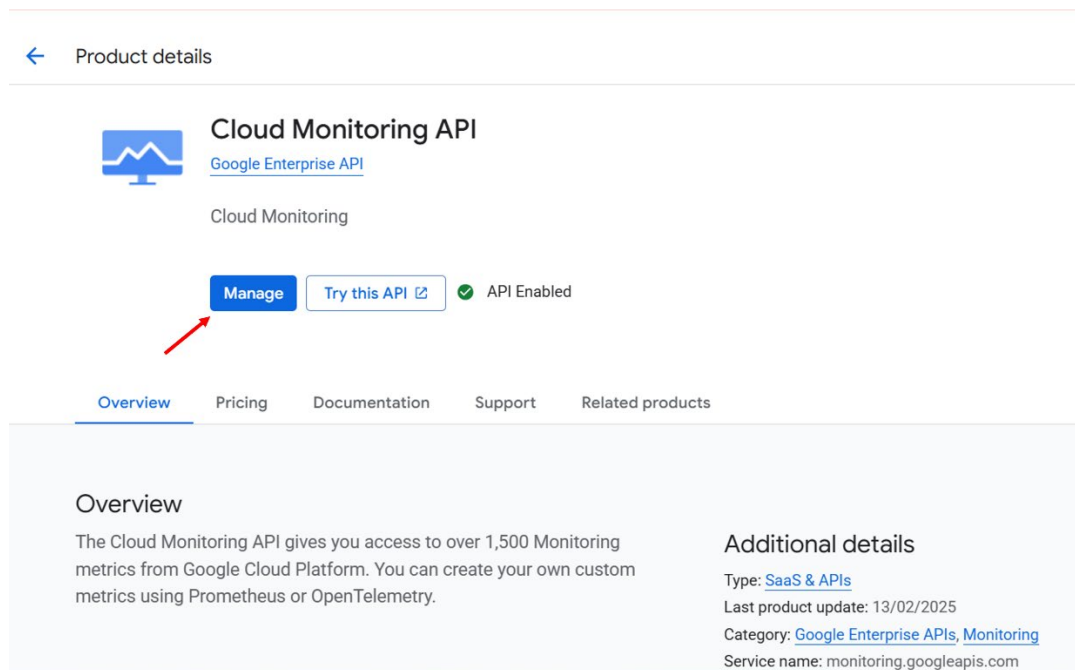
Step 1: From the Google Cloud homepage, click on **Console** to open the Google Cloud Console.



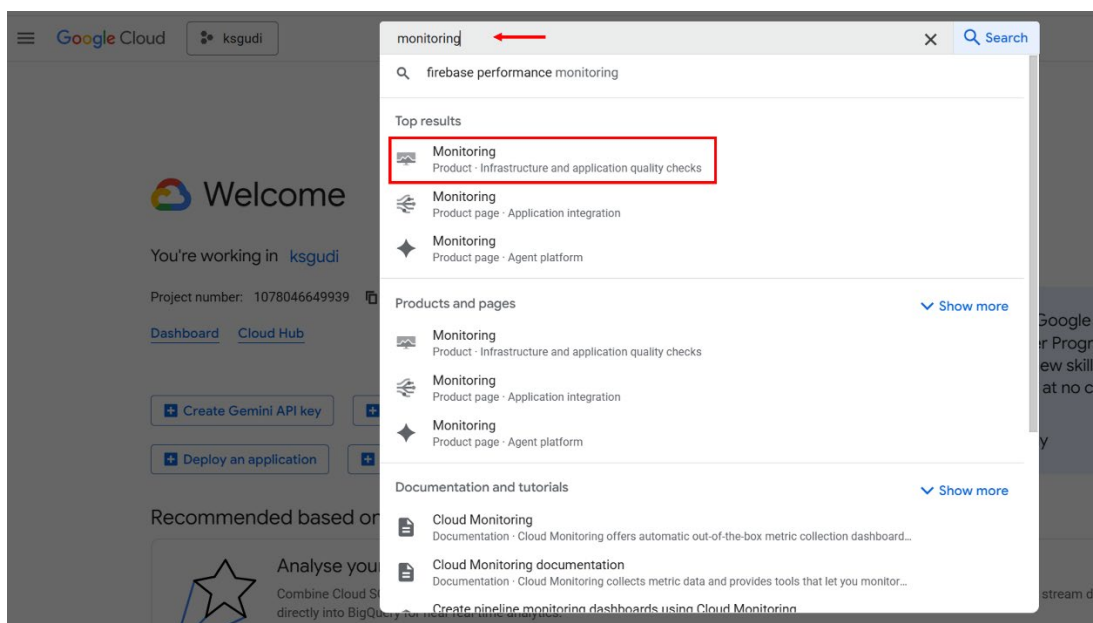
Step 2: In the Google Cloud Console search bar, type **Cloud Monitoring API** and select **Cloud Monitoring API** from the search results.



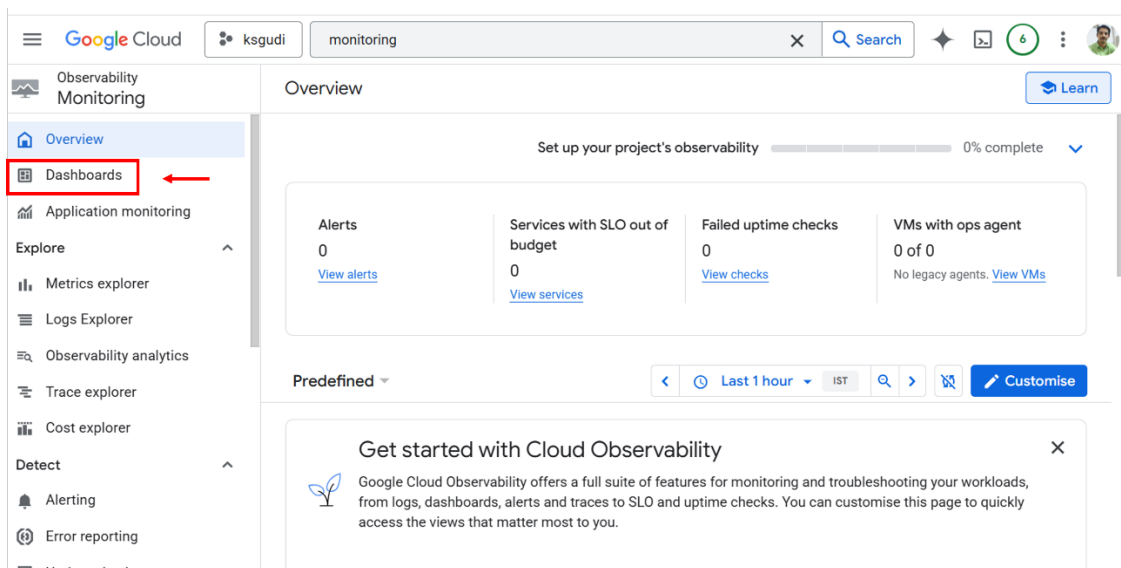
Step 3: Verify that the **Cloud Monitoring API** is enabled and click **Manage**.



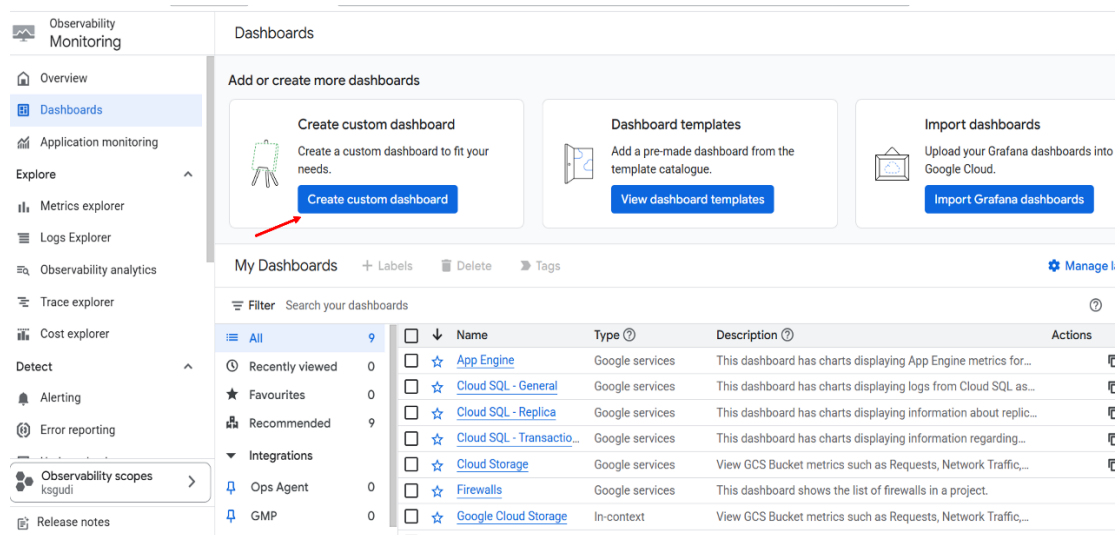
Step 4: Search for **Monitoring** in the console search bar and select **Monitoring** from the search results.



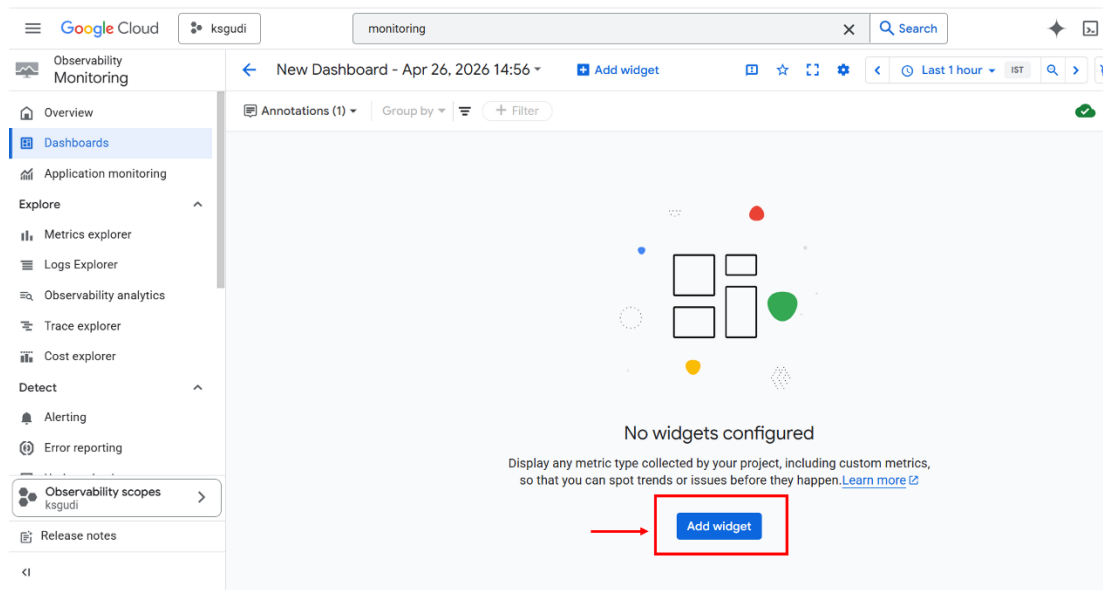
Step 5: On the Monitoring overview page, click **Dashboards** from the left-side navigation menu.



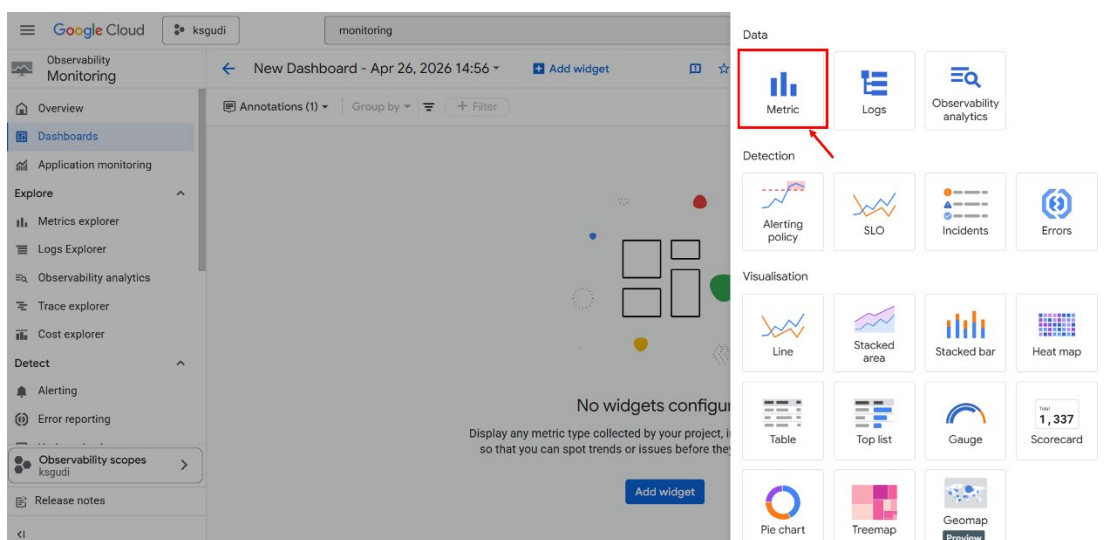
Step 6: In the Dashboards page, click **Create custom dashboard**.



Step 7: In the new dashboard page, click **Add widget** to add a monitoring chart.



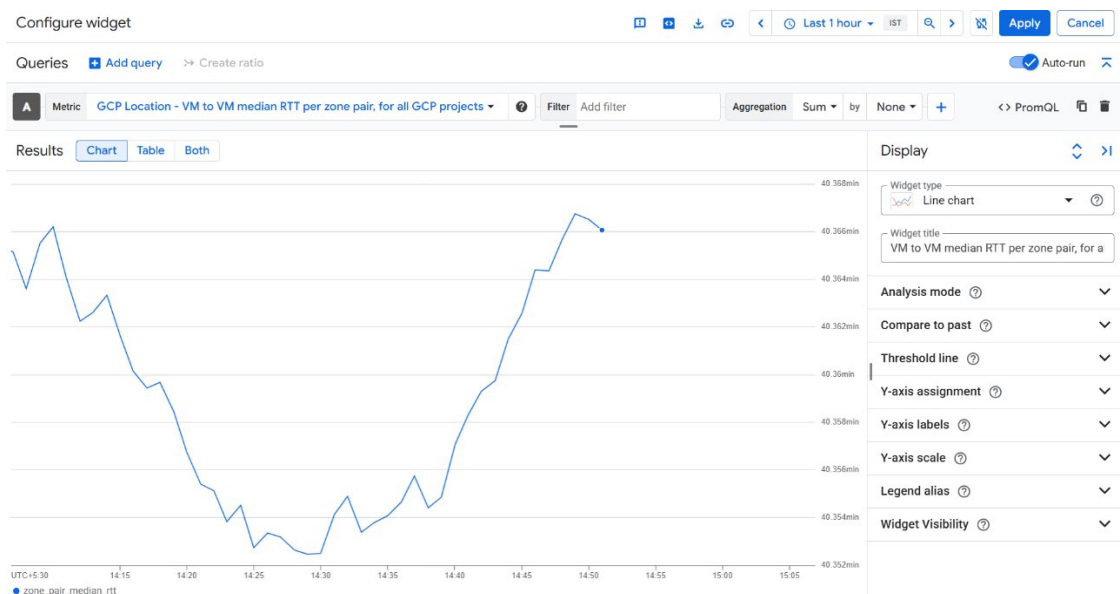
Step 8: Select **Metric** from the widget options.



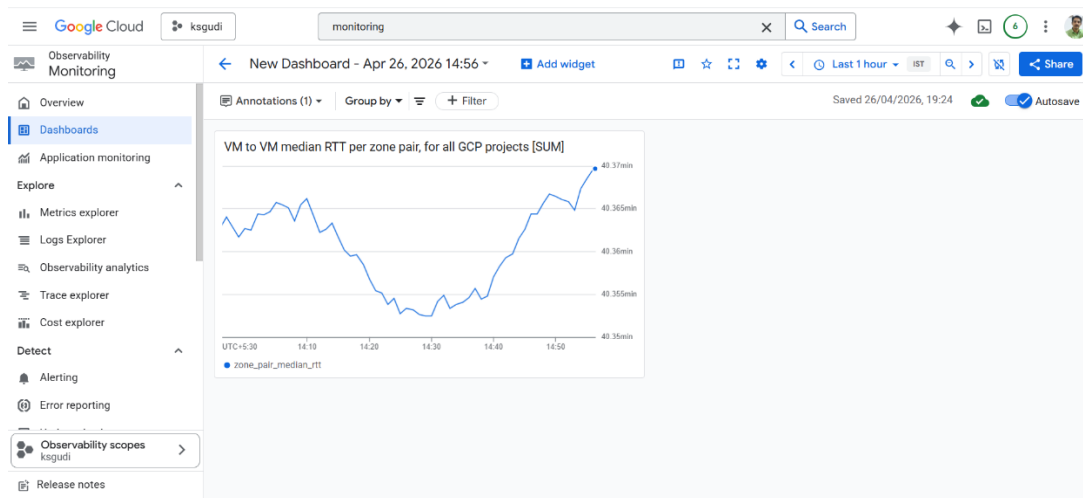
Step 9: Click **Select a metric**, choose **GCP Location**, select **All_gcp**, and then select **VM to VM median RTT per zone pair, for all GCP projects** and Click **Apply**.

The screenshot shows the 'Configure widget' interface in Google Cloud Monitoring. A 'Select a metric' dialog is open, displaying a list of metric categories on the left and a list of active metrics on the right. The 'GCP Location' category is selected (2), and the 'All_gcp' category is selected (3). The 'VM to VM median RTT per zone pair, for all GCP projects' metric is selected (4). The 'Apply' button is visible at the bottom right of the dialog.

Step 10: Verify that the selected metric graph is displayed and click **Apply** to add the chart to the dashboard.



Step II: Verify that the selected metric chart is added to the dashboard and the dashboard is saved automatically.



To make the monitoring more effective, the following activities can also be performed:

1. Create a **Cloud SQL instance**, run basic SQL queries, and monitor database metrics such as CPU utilization, connections, and storage usage.
2. Create an **alert policy** by setting a threshold value for any metric, so that Cloud Monitoring can notify when resource usage crosses the defined limit.
