**Script Breakdown**

The script begins by executing gcloud commands to retrieve IP utilization data. This is done via the run\_command() function, which:

Executes a command using the subprocess.run() method.

Handles both standard output and JSON-formatted output.

Returns the parsed data from the command.

Once the command returns the data, the script processes the list of insights:

Filters out insights related to "IP Utilization Summary" with an "ACTIVE" state.

Collects the INSIGHT\_ID of each relevant insight for further processing.

For each relevant insight, the script runs another gcloud command to fetch more granular data using the INSIGHT\_ID. This command is designed to retrieve JSON-formatted information on IP utilization summaries.

Calculating Metrics: The script then calculates key metrics for each subnet:

Total IPs in each subnet.

Allocated IPs: Based on the subnet's allocation ratio.

Free IPs: Difference between total and allocated IPs.

Weighted Average of Free IPs: Calculated by factoring in the allocation ratio for each subnet.

Mean Allocation Ratio: Average allocation ratio for all subnets.

These metrics are calculated by the calculate\_metrics() function, which processes the subnet statistics and outputs the relevant data.

**Example of output:**

**Subnet: 10.0.0.0/24, Total IPs: 256, Allocated: 180, Free: 76**

Storing and Presenting Results:

After the calculations, the script stores the results in a dictionary (response\_dict).

The dictionary contains the resource name (subnet) as the key and the calculated metrics (mean free IPs and allocation ratio) as the value.

Key Functions

run\_command(command, is\_formatted=False):

Executes this command and parses its output.

If is\_formatted=True, the output is expected to be in JSON format, which is parsed and returned.

Otherwise, the output is treated as a tabular format and will return as a list of dictionaries (representing rows).

calculate\_metrics(subnet\_stats):

Processes subnet statistics, calculates total IPs, allocated IPs, and free IPs for each subnet.

Computes the weighted average of free IPs and the mean allocation ratio across all subnets.

Returns these metrics.

**Use Case**

IP Address Management: This script provides valuable insights into the current utilization and efficiency of IP address allocation across subnets in the organization's cloud infrastructure. This can help ensure that IP addresses are being used optimally, preventing wastage and improving resource allocation.

Cost Efficiency: By identifying subnets with a high number of free IPs, the team can adjust IP allocations to optimize costs related to IP address consumption and management within GCP.

Automation & Time Savings: The script automates the collection, filtering, and calculation processes that would otherwise be time-consuming and error-prone if done manually. This allows for more frequent assessments and quicker decision making.

Improved Monitoring: The system continuously monitors the state of IP utilization, offering a real-time view of network resources. The ability to filter insights based on activity and utilization is critical for maintaining high-performance and efficient network setups.