

5. Schedule a cloud function which runs on every friday that gathers information about the CPU utilization of all the VM instances along with the user information and sends the report to the devops team.

1. Create a cloud function which uses the Stackdriver monitoring library for python to gather information about CPU utilization of all VM instances.
2. Create a cloud scheduler job to schedule it to run on every friday.

Name	State	Description	Frequency	Target
av-a2-q5	Enabled	runs on every friday at 00:00	0 0 ** fri (Asia/Calcutta)	URL: https://us-central1-pe-training.cloudfunctions.net/av-a2-q5-1

3. Add the following code to your cloud function:

```
from google.cloud import monitoring_v3
import time
from datetime import datetime

def hello_world(request):
    """Responds to any HTTP request.
    Args:
        request (flask.Request): HTTP request object.
    Returns:
        The response text or any set of values that can be turned into a
        Response object using
        `make_response <http://flask.pocoo.org/docs/1.0/api/#flask.Flask.make_response>`.
    """
    client = monitoring_v3.MetricServiceClient()
    project_name = client.project_path('pe-training')
    interval = monitoring_v3.types.TimeInterval()
    now = time.time()
    interval.end_time.seconds = int(now)
    interval.end_time.nanos = int((now - interval.end_time.seconds) * 10**9)
    interval.start_time.seconds = int(now - 1200)
    interval.start_time.nanos = interval.end_time.nanos
    results = client.list_time_series(
        project_name,
        'metric.type = "compute.googleapis.com/instance/cpu/utilization"',
        interval,
        monitoring_v3.enums.ListTimeSeriesRequest.TimeSeriesView.FULL)
    for result in results:
        print('User information(labels):')
        print(result.metadata.user_labels)
        print()
        print('Instance ID: ' + str(result.resource.labels['instance_id']))
        for point in result.points:
            print(str(datetime.fromtimestamp(int(str(point.interval.end_time).split(':')[1]))) + ':' +
                  str(point.value.double_value))
```