Show off your #GoogleClout!

Stump your fellow Google Cloud Innovators with a playful challenge - give them a Google Cloud task to solve in just 15 minutes.

Our technical labs team is standing by to build out your challenge in [the Google Cloud SkillsBoost platform]() - all you have to do is describe how it'll work.

Example Challenge

To help get your creative juices flowing, here's a sample challenge celebrating the new Remote Functions for BigQuery.

Problem Title: "Working Remote"

Problem statement:

You want to use the new Remote Functions to increment arbitrary values in your BigQuery queries. You've already created a Google Cloud Function called `remote_add`, but when you try to execute it from BigQuery as follows:

```
SELECT val, my_bq_project.my_dataset.remote_add(val, 2)
  FROM UNNEST([NULL,2,3,5,8]) AS val;
```

you get an error message. Can you fix the function and integration to get a successful response? Successful response to the above query should look like this:

```
+------+-----+
|  val | f0_ |
+------+-----+
| NULL |   2 |
|    2 |   4 |
|    3 |   5 |
|    5 |   7 |
|    8 |  10 |
+------+-----+
```

Start state (initialized by lab architect): Intentionally broken Python Cloud Function (missing `import json` statement) called "remote_add" created with the following code:

```
_MAX_LOSSLESS=9007199254740992
```

```
def batch_add(request):
  try:
    return_value = []
    request_json = request.get_json()
    calls = request_json['calls']
    for call in calls:
      return_value.append(sum([int(x) if isinstance(x, str) else x for x
in call if x is not None]))
    replies = [str(x) if x > _MAX_LOSSLESS or x < -_MAX_LOSSLESS else x
for x in return_value]
    return_json = json.dumps( { "replies" :  replies} )
    return return_json
  except Exception as inst:
    return json.dumps( { "errorMessage": 'something unexpected in input'
} ), 400
```

Grading rubric

25% solution: CLOUD_RESOURCE connection created for the function

50% solution: Service account for the connection has invoke permissions on the function

75% solution: “json” import added to function code (sneaky!)

100% solution: Function successfully invoked from BigQuery

Google Cloud services used: BigQuery, Cloud Functions

Difficulty level (beginner / intermediate / advanced): Intermediate