

<b>07.1g: Kubernetes Guestbook</b>	<b>3</b>
4. Create Kubernetes cluster	3
What is the name of the Instance Template dynamically generated to create the two nodes (VMs)?	3
What is the name of the Instance Group dynamically generated that the two nodes belong to?	3
gke-guestbook-default-pool-0fc69732-grp	3
What are the names of the two nodes?	3
5. Prepare a container image	4
Take a screenshot of the container image created	4
7. Deploy the configuration	4
Take a screenshot of the output of the following command when all 3 replicas reach a "Running" state.	4
Take a screenshot of listing services with LoadBalancer indicating an external IP address that is ready for access.	4
8. View the Guestbook	5
Take a screenshot of the Guestbook including the URL with the entry in it.	5
Take a screenshot of the managed guestbook pods and the service being exposed.	6
<b>07.2a: APIs</b>	<b>8</b>
4. Test code	8
Take a screenshot of the resulting page including the URL bar.	8
Click "Reload" in the browser and take another screenshot showing the image has changed:	8
8. Test code	8
Use curl on your Linux VM to access the API endpoint and show the results. Take a screenshot for your lab notebook.	8
<b>07.2g: APIs (Slack, Knowledge Graph)</b>	<b>9</b>
2. Code	9
Could we have used the API Discovery package to interact with the Vision API?	9
Does Google provide a Python package specifically for accessing the Knowledge Graph API?	9
3. Code	9
Show the source line that constructs the query we wish to send to the Knowledge Graph API.	9
Show the source line that then executes the query and saves the response. What is the name of the method that sends the query to the Knowledge Graph API?	9
What is the Python data type that is used to represent the formatted message?	9
What are the three main attributes of the formatted message passed back to Slack?	9
5. Create a Slack workspace	10
What would be the difference between an adversary finding out YOUR_SLACK_SIGNING_SECRET versus finding out YOUR_KG_API_KEY?	10

8. Test the command	10
Take a screenshot of its response for your lab notebook.	10
<b>07.3a: Lambda, API Gateway Guestbook</b>	<b>11</b>
3. REST API Code	11
What might go wrong when we call scan? Think about the way DynamoDB works, and look at the scan documentation for a hint. What could be done to address this problem?	11
10. Deploy API to production and view entries	11
Take a screenshot that shows that you can view the entries in the backend database.	11
12. API endpoint for signing (2)	12
Take a screenshot showing that the submission worked.	12
16. Configure and Deploy the Frontend	12
Take a screenshot as before that shows your entry and the static website hosting URL	12
<b>07.3g: Cloud Functions API Guestbook</b>	<b>13</b>
5. Test the API via Cloud Functions (POST)	13
Take a screenshot of the output for your lab notebook.	13
6. Test the API via Python Requests (GET)	14
Take a screenshot of the loop and its output	14
7. Test the API via Python Requests (POST)	14
Take a screenshot of the output for your lab notebook	14
10. Version #1: Local file system	15
Take a screenshot of the Guestbook including the URL.	15
11. Version #2: Google Cloud Storage bucket	15
Take a screenshot of the Guestbook including the URL.	15

## 07.1g: Kubernetes Guestbook

### 4. Create Kubernetes cluster

- What is the name of the Instance Template dynamically generated to create the two nodes (VMs)?

`gke-guestbook-default-pool-0fc69732`

- What is the name of the Instance Group dynamically generated that the two nodes belong to?

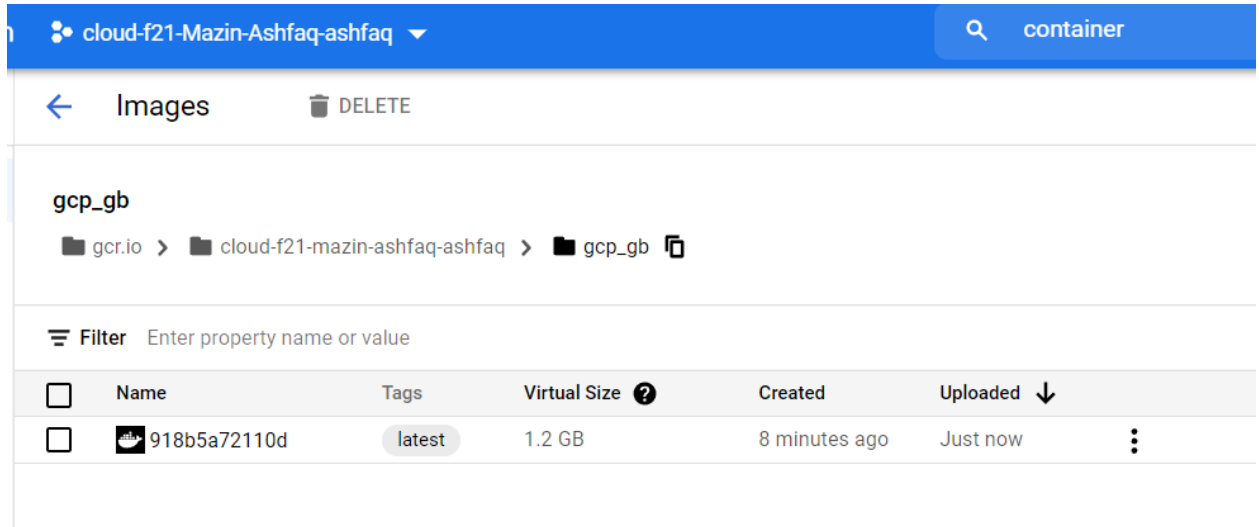
`gke-guestbook-default-pool-0fc69732-grp`

- What are the names of the two nodes?

`gke-guestbook-default-pool-0fc69732-8pqj`  
`gke-guestbook-default-pool-0fc69732-r553`

## 5. Prepare a container image

- Take a screenshot of the container image created



## 7. Deploy the configuration

- Take a screenshot of the output of the following command when all 3 replicas reach a "Running" state.
- Take a screenshot of listing services with LoadBalancer indicating an external IP address that is ready for access.

```
ashfaq@cloudshell:~/cs430-src/05_gcp_datastore (cloud-f21-mazin-ashfaq-ashfaq)$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
guestbook-replicas-45sv7           1/1     Running   0           3m9s
guestbook-replicas-f4kqw           1/1     Running   0          3m10s
guestbook-replicas-x64g2           1/1     Running   0           3m9s
ashfaq@cloudshell:~/cs430-src/05_gcp_datastore (cloud-f21-mazin-ashfaq-ashfaq)$ kubectl get services
NAME      TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
guestbook-lb  LoadBalancer  10.3.245.191  35.233.129.55  80:31789/TCP     3m21s
kubernetes   ClusterIP      10.3.240.1    <none>        443/TCP          38m
```

## 8. View the Guestbook

- Take a screenshot of the Guestbook including the URL with the entry in it.

35.233.129.55

signed on 2021-11-01 20:40:03.003588+00:00  
Hello Compute Engine!

---

Mazin <ashfaq@pdx.edu>  
signed on 2021-11-01 13:14:43.707558+00:00  
Hello Datastore!

---

Mazin <ashfaq@pdx.edu>  
signed on 2021-11-01 20:23:47.229256+00:00  
Hello Docker Datastore!

---

Mazin Ashfaq <ashfaq@pdx.edu>  
signed on 2021-11-07 04:13:41.969404+00:00  
Hello App Engine!

---

Mazin Ashfaq <ashfaq@pdx.edu>  
signed on 2021-11-01 20:29:24.099548+00:00  
Hello Cloud Shell!

---

Mazin Ashfaq <ashfaq@pdx.edu>  
signed on 2021-11-09 01:56:46.478843+00:00  
Hello Cloud Run!

---

Mazin <ashfaq@pdx.edu>  
signed on 2021-11-17 04:54:38.545161+00:00  
Hello Kubernetes!

---

- Take a screenshot of the managed guestbook pods and the service being exposed.

cloud-f21-Mazin-Ashfaq-ashfaq
kubern

Replication Controller details
REFRESH
EDIT
DELETE
AC

⚠ No data is available for the selected time frame.

UTC-8
8:30 PM
8:40 PM
8:50 PM
9:00 PM
9:10 PM
0
UTC-8

Cluster	<a href="#">guestbook</a>
Namespace	default
Created	Nov 16, 2021, 8:50:36 PM
Labels	app: guestbook tier: frontend
Annotations	Not set
Logs ?	<a href="#">Container logs</a> , <a href="#">Audit logs</a>
Pods	3 current / 3 desired
Label selector	app = guestbook tier = frontend

### Pod specification

Labels	app: guestbook tier: frontend
Termination grace period	30
Restart policy	Always
Containers	<a href="#">guestbook-app</a>

### Managed pods

Name	Status	Restarts	Created on ↑
guestbook-replicas-f4kqw	✓ Running	0	Nov 16, 2021, 8:50:36 PM
guestbook-replicas-45sv7	✓ Running	0	Nov 16, 2021, 8:50:37 PM
guestbook-replicas-x64g2	✓ Running	0	Nov 16, 2021, 8:50:37 PM

### Exposing services ?

Name ↑	Type	Endpoints
guestbook-lb	Load balancer	35.233.129.55:80 ↗

- Take a screenshot of the load balancer and its details

cloud-f21-Mazin-Ashfaq-ashfaq
load balancer

Load balancer details
EDIT
DELETE

## a1014592d66234ca2a63b0d7fe414ac9

### Frontend

Protocol ↑	IP:Port	Network Tier ?
TCP	35.233.129.55:80	Premium

### Backend

Name	Region	Health check
a1014592d66234ca2a63b0d7fe414ac9	us-west1	<a href="#">k8s-d28b67ed80e2f477-node</a>

#### ADVANCED CONFIGURATIONS

Instance ↑	Zone	35.2
gke-guestbook-default-pool-b4907d85-rrzh	us-west1-b	✓
gke-guestbook-default-pool-b4907d85-wqk5	us-west1-b	✓

- Take a screenshot of the addresses allocated and indicate the ones associated with nodes versus the one associated with the load balancer.

cloud-f21-Mazin-Ashfaq-ashfaq
vpcn

External IP addresses
RESERVE STATIC ADDRESS
REFRESH
RELEASE STATIC ADDRESS

Filter Enter property name or value

<input type="checkbox"/>	Name	External Address	Region	Type ↓	Version	In use by	Net
<input type="checkbox"/>	–	34.83.139.94	us-west1	Ephemeral	IPv4	VM instance gke-guestbook-default-pool-b4907d85-rrzh (Zone us-west1-b)	
<input type="checkbox"/>	–	34.105.121.204	us-west1	Ephemeral	IPv4	VM instance gke-guestbook-default-pool-b4907d85-wqk5 (Zone us-west1-b)	

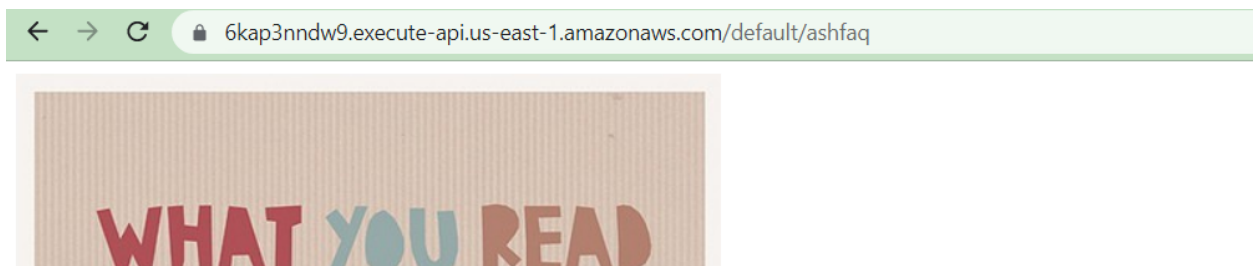
## 07.2a: APIs

### 4. Test code

- Take a screenshot of the resulting page including the URL bar.



- Click "Reload" in the browser and take another screenshot showing the image has changed:



### 8. Test code

- Use `curl` on your Linux VM to access the API endpoint and show the results. Take a screenshot for your lab notebook.

```
C:\Users\mzash>curl https://ii7iykmtlg.execute-api.us-east-1.amazonaws.com/default/gettime-ashfaq
{"currentTime": "2021-11-17 06:19:00.150406"}
C:\Users\mzash>
```



## 07.2g: APIs (Slack, Knowledge Graph)

### 2. Code

- Could we have used the API Discovery package to interact with the Vision API?

Not sure

- Does Google provide a Python package specifically for accessing the Knowledge Graph API?

Not sure

### 3. Code

- Show the source line that constructs the query we wish to send to the Knowledge Graph API.

```
req = kgsearch.entities().search(query=query, limit=1)
```

- Show the source line that then executes the query and saves the response. What is the name of the method that sends the query to the Knowledge Graph API?

```
kg_search_response = make_search_request(request.form["text"])
```

- What is the Python data type that is used to represent the formatted message?

array

- What are the three main attributes of the formatted message passed back to Slack?

Response type

Text

attachments

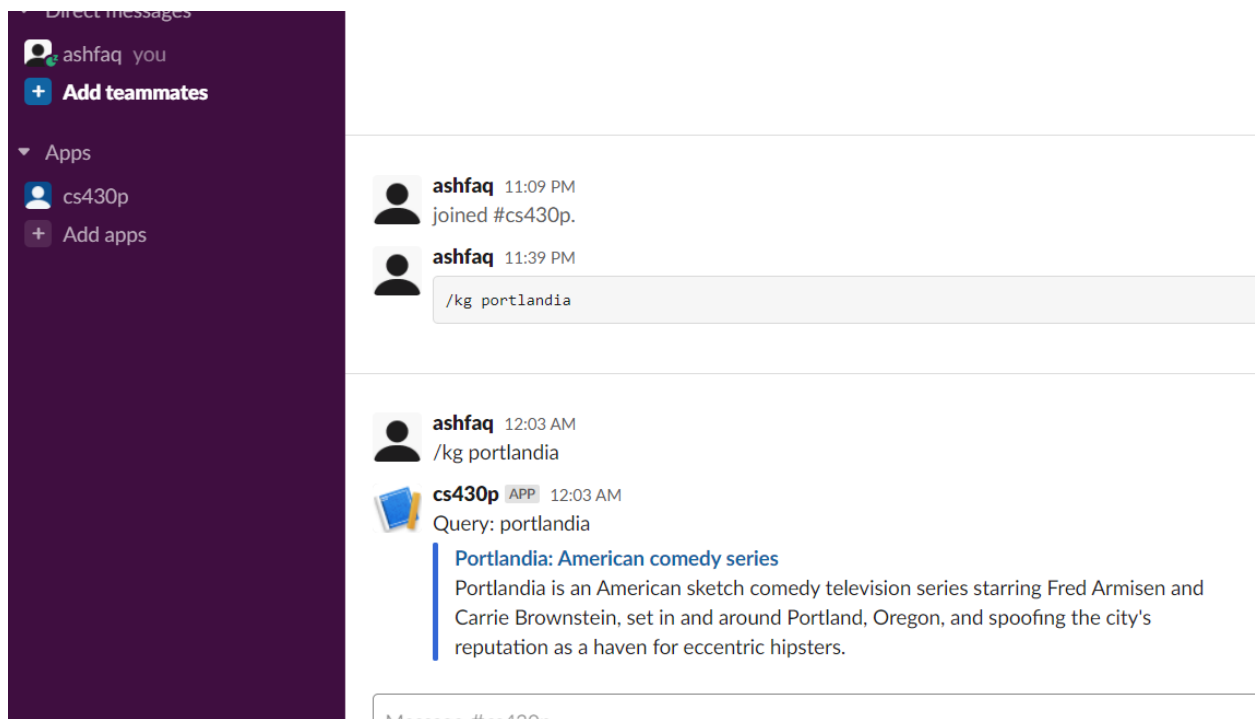
## 5. Create a Slack workspace

- What would be the difference between an adversary finding out `YOUR_SLACK_SIGNING_SECRET` versus finding out `YOUR_KG_API_KEY`?

Not sure

## 8. Test the command

- Take a screenshot of its response for your lab notebook.



## 07.3a: Lambda, API Gateway Guestbook

### 3. REST API Code

- What might go wrong when we call `scan`? Think about the way DynamoDB works, and look at the [scan documentation](#) for a hint. What could be done to address this problem?

Not sure

### 10. Deploy API to production and view entries

- Take a screenshot that shows that you can view the entries in the backend database.

Guestbook

Name:

Email:

Message:

Entries

Mazin <ashfaq@pdx.edu>  
signed on 2021-11-01 00:11:06.698997  
Hello DynamodDB!

## 12. API endpoint for signing (2)

- Take a screenshot showing that the submission worked.

```
[
  {
    "message": "Hello API Gateway",
    "date": "2021-11-18 00:44:16.458658",
    "email": "ashfaq@pdx.edu",
    "name": "Ashfaq"
  }
]
```

## 16. Configure and Deploy the Frontend

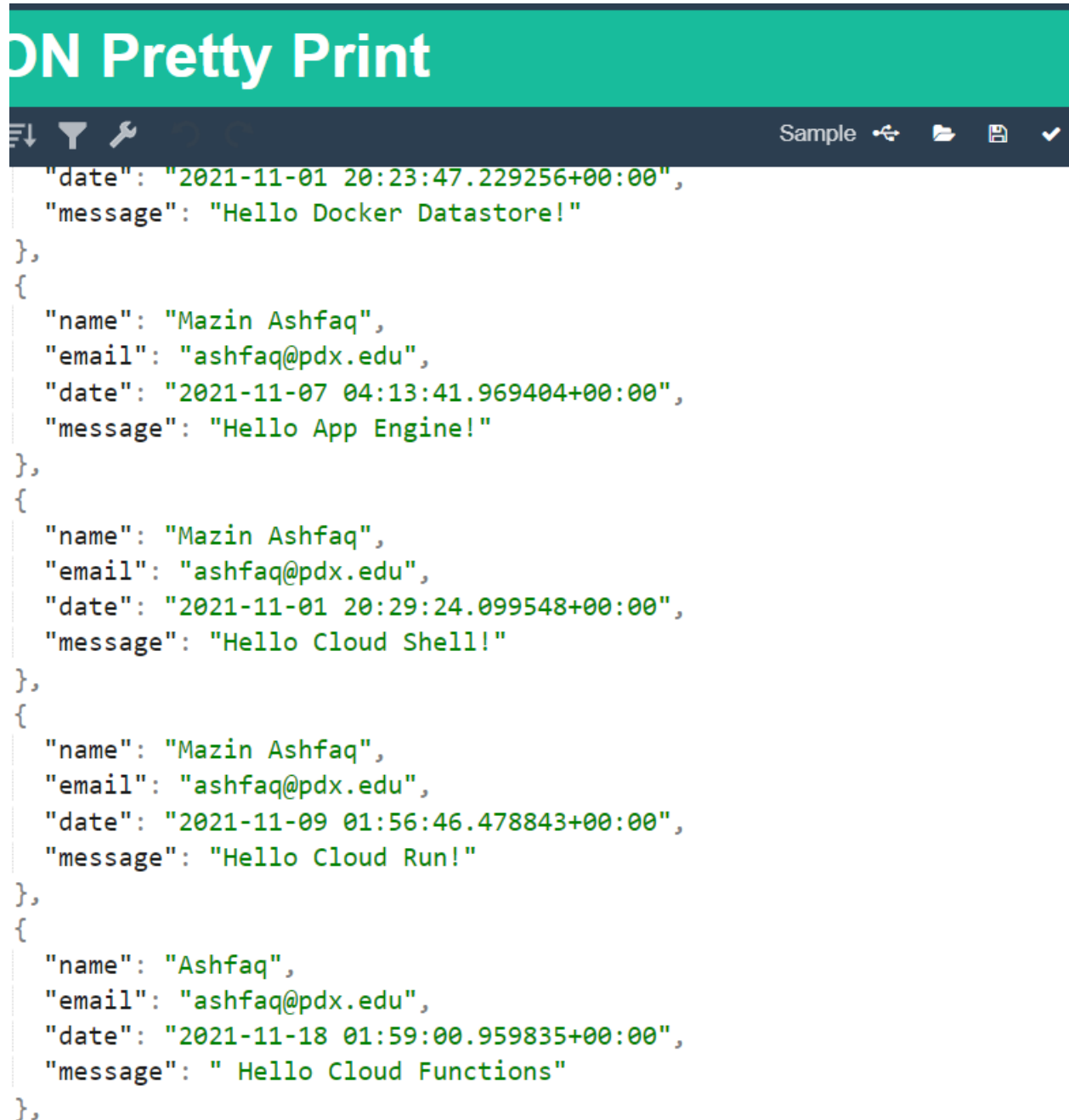
- Take a screenshot as before that shows your entry and the static website hosting URL

Could not get it to work. 403 Forbidden.

## 07.3g: Cloud Functions API Guestbook

### 5. Test the API via Cloud Functions (POST)

- Take a screenshot of the output for your lab notebook.



The screenshot shows a terminal window with a green header bar that says "ON Pretty Print". Below the header is a dark blue toolbar with icons for a dropdown menu, a filter, a wrench, and a "Sample" button. The main area of the terminal displays a JSON array of five objects, each representing a guestbook entry. The entries are for "Mazin Ashfaq" and "Ashfaq", with their respective email addresses, timestamps, and messages.

```
{
  "date": "2021-11-01 20:23:47.229256+00:00",
  "message": "Hello Docker Datastore!"
},
{
  "name": "Mazin Ashfaq",
  "email": "ashfaq@pdx.edu",
  "date": "2021-11-07 04:13:41.969404+00:00",
  "message": "Hello App Engine!"
},
{
  "name": "Mazin Ashfaq",
  "email": "ashfaq@pdx.edu",
  "date": "2021-11-01 20:29:24.099548+00:00",
  "message": "Hello Cloud Shell!"
},
{
  "name": "Mazin Ashfaq",
  "email": "ashfaq@pdx.edu",
  "date": "2021-11-09 01:56:46.478843+00:00",
  "message": "Hello Cloud Run!"
},
{
  "name": "Ashfaq",
  "email": "ashfaq@pdx.edu",
  "date": "2021-11-18 01:59:00.959835+00:00",
  "message": " Hello Cloud Functions"
},
}
```

## 6. Test the API via Python Requests (GET)

- Take a screenshot of the loop and its output

```
>>> print(json.dumps(resp.json(),indent=2)
... )
[
  {
    "name": "Mazin Ashfaq",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-01 20:40:03.003588+00:00",
    "message": "Hello Compute Engine!"
  },
  {
    "name": "Mazin",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-01 13:14:43.707558+00:00",
    "message": "Hello Datastore!"
  },
  {
    "name": "Mazin",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-01 20:23:47.229256+00:00",
    "message": "Hello Docker Datastore!"
  },
  {
    "name": "Mazin Ashfaq",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-07 04:13:41.969404+00:00",
    "message": "Hello App Engine!"
  },
  {
    "name": "Mazin Ashfaq",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-01 20:29:24.099548+00:00",
    "message": "Hello Cloud Shell!"
  },
  {
    "name": "Mazin Ashfaq",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-09 01:56:46.478843+00:00",
    "message": "Hello Cloud Run!"
  },
  {
    "name": "Ashfaq",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-18 01:59:00.959835+00:00",
    "message": " Hello Cloud Functions"
  },
  {
    "name": "Mazin",
    "email": "ashfaq@pdx.edu",
    "date": "2021-11-17 04:54:38.545161+00:00",
    "message": "Hello Kubernetes!"
  }
]
>>>
```

## 7. Test the API via Python Requests (POST)

- Take a screenshot of the output for your lab notebook

```
>>> resp = requests.post('https://us-central1-cloud-f21-mazin-ashfaq-ashfaq.cloudfunctions.net/entry', json=mydict)
>>> print(resp)
Response [200]>
>>>
```

## 10. Version #1: Local file system

- Take a screenshot of the Guestbook including the URL.

127.0.0.1:5500/06\_gcp\_restapi\_cloudfunctions/frontend-src/index.html

ashfaq <ashfaq@pdx.edu>  
signed on 2021-11-18 02:30:07.058110+00:00  
Hello Cloud Functions from SPA!

---

## 11. Version #2: Google Cloud Storage bucket

- Take a screenshot of the Guestbook including the URL.

storage.googleapis.com/restapi-ashfaq/index.html

ashfaq <ashfaq@pdx.edu>  
signed on 2021-11-18 02:34:26.563719+00:00  
Hello Cloud Functions from SPA in GCS!

---