

Kubernetes Guestbook	1
Lambda API	4
APIs (Slack, Knowledge Graph)	5
Lambda, API Gateway Guestbook	7
Cloud Functions API Guestbook	9

Kubernetes Guestbook

- *What is the name of the Instance Template dynamically generated to create the two nodes (VMs)?*
 - [gke-guestbook-default-pool-489a8218](#)
- *What is the name of the Instance Group dynamically generated that the two nodes belong to?*
 - [gke-guestbook-default-pool-489a8218-grp](#)
- *What are the names of the two nodes?*
 - gke-guestbook-default-pool-489a8218-5hx3
 - gke-guestbook-default-pool-489a8218-wfhg

The screenshot shows the Google Cloud Platform Storage browser interface. The path is gcr.io > cloud-f21-robin-su-robisu > gcp_gb. A list of objects is displayed:

Name	Tags	Virtual Size	Created	Uploaded
b42f7a0c09e9	latest	1.1 GB	Just now	Just now

```
robisu@cloudshell:~/cs430-src/05_gcp_datastore (cloud-f21-robin-su-robisu)$ $kc get po
NAME          READY   STATUS    RESTARTS   AGE
guestbook-replicas-8d8sh  1/1     Running   0          90s
guestbook-replicas-fx8g2  1/1     Running   0          90s
guestbook-replicas-qx4p5  1/1     Running   0          90s
robisu@cloudshell:~/cs430-src/05_gcp_datastore (cloud-f21-robin-su-robisu)$
```

```
^Crobisu@cloudshell:~/cs430-src/05_gcp_datastore (cloud-f21-robin-su-robisu)$ $kc
get svc
NAME          TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
guestbook-lb   LoadBalancer  10.3.249.183  35.233.139.126  80:30068/TCP  52s
robisu@cloudshell:~/cs430-src/05_gcp_datastore (cloud-f21-robin-su-robisu)$
```

The screenshot shows a Cloud Shell terminal window. At the top, it displays the command output for getting a service named 'guestbook-lb' with its details: Cluster-IP 10.3.249.183, External-IP 35.233.139.126, Port(s) 80:30068/TCP, and Age 52s. Below this, a green message is displayed: "signed on 2021-11-01 19:16:14.843261+00:00" and "Hello Cloud Shell!". The terminal has a standard Linux-style command-line interface with various icons and symbols.

Robin Su <robisu@pdx.edu>
signed on 2021-11-04 23:28:38.382372+00:00
Hello Kubernetes!

Robin Su <robisu@pdx.edu>

Managed pods

Name	Status	Restarts	Created on
guestbook-replicas-8d8sh	Running	0	Nov 4, 2021, 4:26:17 PM
guestbook-replicas-fx8g2	Running	0	Nov 4, 2021, 4:26:17 PM
guestbook-replicas-qx4p5	Running	0	Nov 4, 2021, 4:26:17 PM

Exposing services ?

Name	Type	Endpoints	Labels
guestbook-lb	Load balancer	35.233.139.126:80	robisu

[Load balancer details](#)

[EDIT](#)

[DELETE](#)

a1292469a9e9f4d82becf5c71a1aead6

Frontend

Protocol	IP:Port	Network Tier
TCP	35.233.139.126:80	Premium

Backend

Name	Region	Health check
a1292469a9e9f4d82becf5c71a1aead6	us-west1	k8s-c081d165bf322cfa-node

ADVANCED CONFIGURATIONS

Instance	Zone	
gke-guestbook-default-pool-489a8218-5hx3	us-west1-b	
gke-guestbook-default-pool-489a8218-wfhg	us-west1-b	

robis

cloud-f21-Robin-Su-robisu

Search products and resources

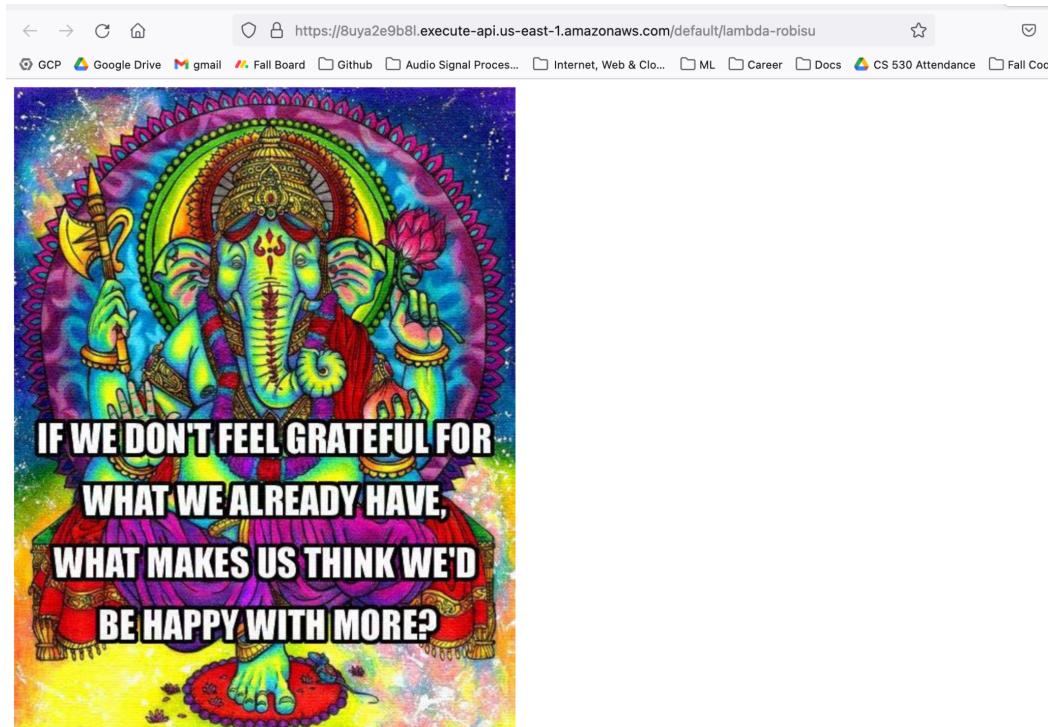
External IP addresses RESERVE STATIC ADDRESS REFRESH RELEASE STATIC ADDRESS

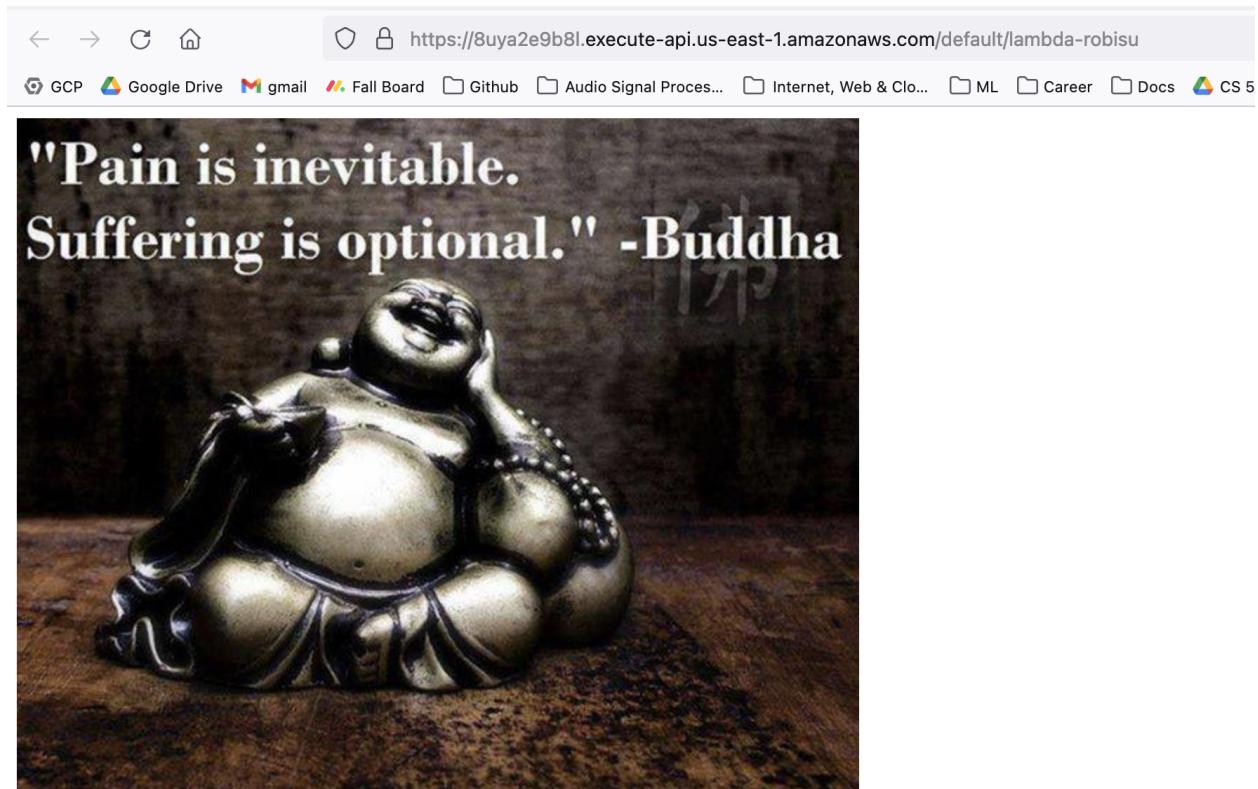
Filter Enter property name or value

Name	External Address	Region	Type	Version	In use by
—	34.82.69.14	us-west1	Ephemeral	IPv4	VM instance gke-guestbook-default-pool-489a8218-wfhg (Zone us-west1-b)
—	34.105.59.130	us-west1	Ephemeral	IPv4	VM instance gke-guestbook-default-pool-489a8218-5hx3 (Zone us-west1-b)
—	35.233.139.126	us-west1	Ephemeral	IPv4	Forwarding rule a1292469a9e9f4d82becf5c71a1aead6

In the screenshot above, the first two listings are the external IPs for the two pod replicas, and the third is associated with the load balancer.

Lambda API





```
~ curl https://1a15bgkw4.execute-api.us-east-1.amazonaws.com/default/gettime-robisu
{"currentTime": "2021-11-05 00:03:32.576440"}
```

APIs (Slack, Knowledge Graph)

- Could we have used the API Discovery package to interact with the Vision API?
 - Yes, but the Vision-specific API has more built-in functions for handling interacting with the API and with authentication. Google provides Discovery Documents that specify how external API clients can interact with their APIs when not using the specified client libraries.

- Does Google provide a Python package specifically for accessing the Knowledge Graph API?
 - There is no specific client library in Python for the Knowledge Graph API.

- Show the source line that constructs the query we wish to send to the Knowledge Graph API.
 - Line 85:

```
 85 |     req = kgsearch.entities().search(query=query, limit=1)
 86 |     res = req.execute()
```

robius
- 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?
 - Line 86 sends the query, using the method ‘execute()’:

```
 85 |     req = kgsearch.entities().search(query=query, limit=1)
 86 |     res = req.execute()
 87 |     return format_slack_message(averv, res)
```

robius
- What is the Python data type that is used to represent the formatted message?
 - The formatted message is a Python dictionary
- What are the three main attributes of the formatted message passed back to Slack?
 - ‘response_type’, ‘text’, and ‘attachments’
- What would be the difference between an adversary finding out YOUR_SLACK_SIGNING_SECRET versus finding out YOUR_KG_API_KEY?
 - If an adversary knew the Slack Signing Secret, they would be able to manipulate the Slack app/bot and change its functionality or credentials. If they had the KG_API_KEY, they would have access to querying the Knowledge Graph service itself via API, that has nothing to do with the Slack integration.

Query Response:

today ▾

 Robin Su 3:48 PM
/kg portlandia

 kg-bot APP 3:48 PM
Query: portlandia

Portland: City in Oregon
 Portland is the largest and most populous city in the U.S. state of Oregon, and the seat of Multnomah County. It is a major port in the Willamette Valley region of the Pacific Northwest, at the confluence of the Willamette and Columbia rivers in Northwestern Oregon.

robius

Gcloud Logs:

```
robisu@cloudshell:~/python-docs-samples/functions/slack (cloud-f21-robin-su-robisu)$ gcloud functions log
  read --limit 100
LEVEL: D
NAME: kg_search
EXECUTION_ID: fofeawry7qju
TIME_UTC: 2021-11-07 23:48:30.018
LOG: Function execution took 265 ms, finished with status code: 200

LEVEL: D
NAME: kg_search
EXECUTION_ID: fofeawry7qju
TIME_UTC: 2021-11-07 23:48:29.753
LOG: Function execution started
```

Lambda, API Gateway Guestbook

- *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?*
 - **The results returned by the scan call might not include the entry that was just inserted into DynamoDB, since scan uses an eventual consistency paradigm. To avoid this, the scan() call could perhaps be implemented as a callback to the put() function. The Amazon docs also recommend that there is a parameter in the scan call named ConsistentRead that can be set to true to ensure the most up-to-date version of the data.**

Entries

Robin Su <robisu@pdx.edu>
signed on 2021-11-01 09:46:16.515481
Hello DynamoDB!

Robin Su <robisu@pdx.edu>
signed on 2021-11-01 16:56:52.416960
Hello Docker DynamoDB!

Robin Su <robisu@pdx.edu>
signed on 2021-11-01 17:14:02.520862
Hello Cloud9!

Robin Su <robisu@pdx.edu>
signed on 2021-11-01 17:31:50.297167
Hello EC2!

Test POST request in API Gateway Interface:

```
[  
  {  
    "message": "Hello DynamoDB!",  
    "date": "2021-11-01 09:46:16.515481",  
    "email": "robisu@pdx.edu",  
    "name": "Robin Su"  
  },  
  {  
    "message": "Hello Docker DynamoDB!",  
    "date": "2021-11-01 16:56:52.416960",  
    "email": "robisu@pdx.edu",  
    "name": "Robin Su"  
  },  
  {  
    "message": "Hello Cloud9!",  
    "date": "2021-11-01 17:14:02.520862",  
    "email": "robisu@pdx.edu",  
    "name": "Robin Su"  
  },  
  {  
    "message": "Hello EC2!",  
    "date": "2021-11-01 17:31:50.297167",  
    "email": "robisu@pdx.edu",  
    "name": "Robin Su"  
  },  
  {  
    "message": "Hello Elastic Beanstalk!",  
    "date": "2021-11-03 21:19:15.109926",  
    "email": "robisu@pdx.edu",  
    "name": "Robin Su"  
  },  
  {  
    "message": "Hello API Gateway",  
    "date": "2021-11-08 01:07:34.285375",  
    "email": "robisu@pdx.edu",  
    "name": "Robin Su"  
  }  
]
```

The screenshot shows a guestbook form titled "Guestbook". It has fields for "Name" (Robin Su), "Email" (robisu@pdx.edu), and "Message" (Hello S3, API Gateway, Lambda!). A "Sign" button is at the bottom. The browser's address bar shows the URL: robisu-frontend.s3-website-us-east-1.amazonaws.com/index.html.

Name: Robin Su

Email: robisu@pdx.edu

Message:

Hello S3, API Gateway, Lambda!

Sign

Robin Su <robisu@pdx.edu>
signed on 2021-11-08 01:25:03.994616
Hello S3, API Gateway, Lambda!

Cloud Functions API Guestbook

The screenshot shows a JSON viewer displaying three entries from a guestbook API. Each entry contains fields: name, email, date, and message.

```

[{"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-04 00:49:53.936993+00:00", "message": "Hello Cloud Run!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-01 11:40:54.466521+00:00", "message": "Hello Datastore!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-12 21:04:06.098590+00:00", "message": "Hello Cloud Functions!"}]

```

Successful GET request:

```
>>> for k in resp_json[0].keys():
...     print(f"{k}:{resp_json[0][k]}")
...
name:Robin Su
email:robisu@pdx.edu
date:2021-11-04 00:49:53.936993+00:00
message>Hello Cloud Run!
>>> █
```

Successful POST request:

```
>>> print(f"Response Status: {resp.status_code}\n Response Headers: {resp.headers}\n Response Text: {resp.text}")
Response Status: 200
Response Headers: {'Access-Control-Allow-Origin': '*', 'Content-Type': 'application/json', 'Function-Execution-Id': 'rcb47mizvuc5', 'X-Cloud-Trace-Context': '504a6cf338bad2e193a6fb77ea6e49a2;o=1', 'Date': 'Fri, 12 Nov 2021 21:15:43 GMT', 'Server': 'Google Frontend', 'Content-Length': '1165', 'Alt-Svc': 'h3=:443'; ma=2592000,h3-29=:443'; ma=2592000,h3-Q050=:443'; ma=2592000,h3-Q046=:443'; ma=2592000,h3-Q043=:443'; ma=2592000,quic=:443'; ma=2592000; v="46,43"}'
Response Text: [{"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-04 00:49:53.936993+00:00", "message": "Hello Cloud Run!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-01 11:40:54.466521+00:00", "message": "Hello Datastore!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-12 21:04:06.098590+00:00", "message": "Hello Cloud Functions!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-01 19:16:14.843261+00:00", "message": "Hello Cloud Shell!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-04 23:28:38.382372+00:00", "message": "Hello Kubernetes!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-01 18:48:47.850947+00:00", "message": "Hello Docker Datastore!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-01 19:26:47.559153+00:00", "message": "Hello Compute Engine!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-03 21:59:18.541259+00:00", "message": "Hello App Engine!"}, {"name": "Robin Su", "email": "robisu@pdx.edu", "date": "2021-11-12 21:15:42.976460+00:00", "message": "Hello Cloud Functions from Python Requests!"}]
>>> █
```

SPA from local file:



Guestbook

Name:

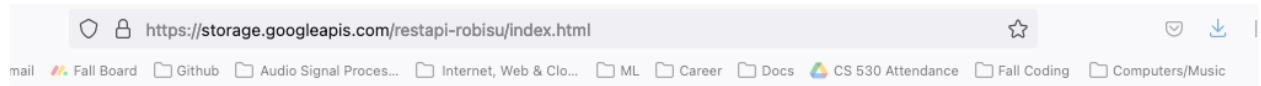
Email:

Message:

Entries

Robin Su <robisu@pdx.edu>
signed on 2021-11-12 21:31:13.377228+00:00
Hello Cloud Functions from SPA!

SPA in GCS using Cloud Functions and REST API backend:



Guestbook

Name:

Email:

Message:

```
Hello Cloud Functions from SPA in GCS!
```

Robin Su <robisu@pdx.edu>
signed on 2021-11-03 21:59:18.541259+00:00
Hello App Engine!

Robin Su <robisu@pdx.edu>
signed on 2021-11-12 21:35:03.176380+00:00
Hello Cloud Functions from SPA in GCS!
