

* Again, due to availability issues, all actions were performed in the us-west1-a zone instead of us-west1-b.

✓ indicates that a section was completed but did not request any screenshots or written answers.

I missed some exclamation points in the guestbook entries. I know I could have edited them in the dynamo db shell - please take pity on me.

TABLE OF CONTENTS:

I. Lab 5.1 - Storage, IAM.....	2
1. GCP Cloud Storage.....	2
2. Storage, IAM.....	2
3. Configuring Permissions.....	4
4. USGS Data and Setup.....	5
5. Python Plotting Code.....	6
6. GCP Cloud Storage #2 (IAM Roles).....	7
7. Create Service Account.....	7
8. Create Compute Engine VM.....	7
9. Service account roles (Compute).....	7
10. Service Account Roles (Storage).....	9
11. GCP Cloud Storage #3 (Python).....	10
12. Python Storage Code.....	10
13. View Object.....	11
14. Clean Up.....	11
15. IAM and Least Privileges.....	11
II. Lab 5.2a - DynamoDB Guestbook.....	12
1. DynamoDB.....	12
2. model_dynamodb.....	12
3. Version 1: Ubuntu VM Python.....	12
4. Obtain AWS Credentials.....	12
5. Run the Application.....	13
6. Version 2: Ubuntu VM Docker.....	13
7. Run the Application.....	14
8. Push the Container Image.....	15
9. Version 3: AWS Cloud9 IDE.....	15
10. Configure the Security Group.....	15
11. Run the Application.....	16
12. Version 4: AWS EC2.....	16
13. Connect to the Instance.....	16
14. Set Up the Instance.....	17
15. Visit the Application.....	17
16. View the Database.....	18
18. Clean Up.....	18
III. Lab 5.2g - Cloud Datastore Guestbook.....	19
1. Cloud Datastore.....	19

2. model_datastore setup.....	19
3. model_datastore.....	19
4. Datastore Setup.....	19
5. 5. Version 1: Ubuntu VM Python.....	19
6. Obtain GCP Credentials.....	19
7. Run the Application.....	19
8. Version 2: Ubuntu VM Docker.....	19
9. Run the Application.....	20
10. Push the Container Image.....	20
11. Version 3: GCP Cloud Shell.....	20
12. Run the Application.....	20
13. Version 4: GCP Compute Engine.....	20
14. Set Up the Instance.....	20
15. Visit the Application.....	20
16. View the Database.....	20
17. Clean Up.....	20

I. Lab 5.1 - Storage, IAM

1. GCP Cloud Storage

Understood 

2. Storage, IAM

Access roles:

- Begin to create a new U2004 VM in us-west1-b and scroll to the Identity and API Access section. What role is attached to the Compute Engine default service account?

Editor.

Permissions for project "cloud-Miner-amminer"

These permissions affect this project and all of its resources. [Learn more](#)

☐ Include (

VIEW BY PRINCIPALS



VIEW BY ROLES

GRANT ACCESS

REMOVE ACCESS

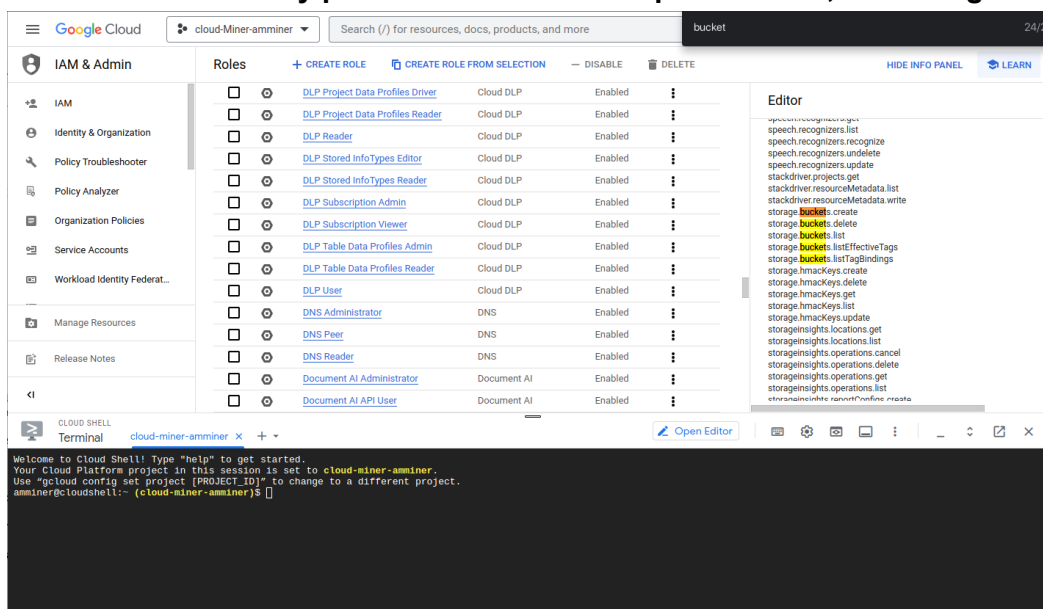
Filter

Enter property name or value

<div><input type="checkbox"/></div>	Type	Principal <div>↑</div>	Name	Role	Security Insights <div>?</div>
<div><input type="checkbox"/></div>	<div></div>	376082578160-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor	<div>7593/7595 excess permissions</div> <div></div>
<div><input type="checkbox"/></div>	<div></div>	amminer@pdx.edu	Amelia Miner	Owner	<div>8477/8694 excess permissions</div> <div></div>

- Would it be sufficient for the VM to perform its functions (creating buckets and reading/writing in them?)

Yes. The VM is heavily provisioned in terms of permissions, this role gets a lot.



The screenshot displays the Google Cloud IAM & Admin console for the project 'cloud-Miner-amminer'. The 'Roles' tab is selected, showing a list of roles. The 'Editor' role is highlighted, and its permissions are listed on the right. The permissions include various speech, stackdriver, storage, and storageinsights operations. A terminal window at the bottom shows the Cloud Shell environment with the project ID 'cloud-Miner-amminer'.

Access scopes:

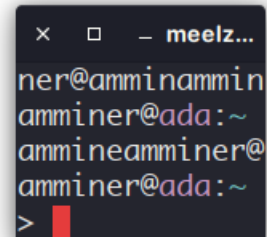
- What permissions are given by the default access scope to Cloud Storage?

Read-only.

Default scopes

When you create a new Compute Engine instance, it is automatically configured with the following access scopes:

- Read-only access to Cloud Storage:
`https://www.googleapis.com/auth/devstorage.read_only`
- Write access to write Compute Engine logs:
`https://www.googleapis.com/auth/logging.write`
- Write access to publish metric data to your Google Cloud projects:
`https://www.googleapis.com/auth/monitoring.write`
- Read-only access to Service Management features required for Google Cloud Endpoints^(Alpha):
`https://www.googleapis.com/auth/service.management.readonly`
- Read/write access to Service Control features required for Google Cloud Endpoints^(Alpha):
`https://www.googleapis.com/auth/servicecontrol`
- Write access to Cloud Trace allows an application running on a VM to write trace data to a project.
`https://www.googleapis.com/auth/trace.append`



- Would they be sufficient for the VM to perform its functions (see above)?

No - the VM can neither create nor write to buckets under the default scope. The above screenshot should be enough for both of these, right? Just in case:

A screenshot of the Google Cloud console. The left pane shows '2. GCP Cloud Storage #1 (USGS)' with instructions on creating a VM and a storage bucket. The right pane shows 'Permissions via service accounts' with a dropdown menu for 'Service account' set to 'Compute Engine default service account'. Below this, there's a section for 'Identity and API access' with a table showing 'IAM & Admin' permissions. The rightmost pane shows the 'default scopes' section, which lists the same permissions as the first screenshot.

- What settings are possible for the VM's access to the Storage API?
Read-only, write-only, read/write, and full.

cloud-Miner-amminer ▼

Search (/) for resources, docs, products, ...

one of the

nce from

n

nce from an

Stackdriver Trace

Write Only ▼

Storage

None

Read Only

Write Only

Read Write

Full

F

Add tags and firewall rules to allow specific network traffic from the Internet

3. Configuring Permissions

- Get the vm up and running with full storage perms via an access role. SSH in. ✓

4. USGS Data and Setup

- Clone the lab repo and download the latest earthquake data as a CSV:

```
git clone
```

```
https://github.com/GoogleCloudPlatform/training-data-analyst
```

```
cd training-data-analyst/CPB100/lab2b
```

```
wget
```

```
https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_wei
```

```
k.csv -O earthquakes.csv
```

```
head -2 earthquakes.csv
```

- What time did the latest earthquake happen?

Tomorrow at 01:31:00 (so in another time zone).

```
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ head -2 earthquakes.csv | awk --field-separator ',' '{ print $1 }'
```

time
2023-10-27T01:31:00.434Z

- What was the magnitude (mag)?

1.4

```
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ head -2 earthquakes.csv | awk --field-separator ',' '{ print $5 }'
```

mag
1.4

- Where was the place it happened?

47 km NNW of Paxson, Alaska

```
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ head -2 earthquakes.csv | awk --field-separator ',' '{ print $14 }'
```

place
"47 km NNW of Paxson

```
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ head -2 earthquakes.csv | awk --field-separator ',' '{ print $15 }'
```

type
Alaska"

```
amminer@usgs:~/training-data-analyst/CPB100/lab2b$
```

5. Python Plotting Code

- Run transform.py, make a new bucket with a unique name, and upload to it like so:

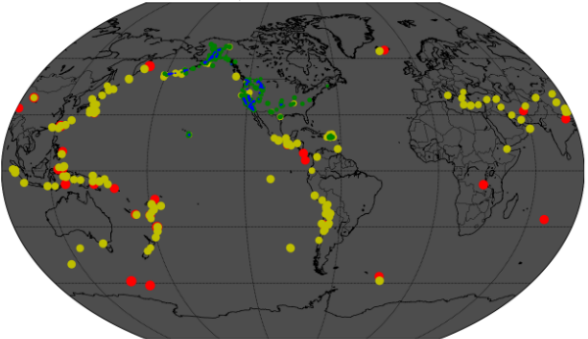
```
gsutil cp earthquakes.* gs://<UNIQUE_BUCKET_NAME>
```

Google Cloud console interface showing the Cloud Storage bucket 'superuniquebucketofearthquakes' and a world map visualization of earthquakes.

Object details for earthquakes.png:

gsutil URI	gs://superuniquebucketofearthquakes/earthquakes.png
Permissions	
Public access	Not public
Protection	
Version history	—
Retention policy	None
Hold status	None
Encryption type	Google-managed

Earthquakes 2023-10-20 to 2023-10-27



Terminal Output:

```
Processing triggers for systemd (245.4-4ubuntu3.22) ...
Processing triggers for man-db (2.9.1-1) ...
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ python3 transform.py
{'timestamp': '2023-10-27T01:31:00.434Z', 'lat': 63.4603, 'lon': -146.0601, 'magnitude': 1.4}
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ ls
commands.sh  earthquakes.csv  earthquakes.htm  earthquakes.png  ingest.sh  install_missing.sh  scheduled  transform.py
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ gsutil mb gs://superuniquebucketofearthquakes
Creating gs://superuniquebucketofearthquakes/...
amminer@usgs:~/training-data-analyst/CPB100/lab2b$ gsutil cp earthquakes.* gs://superuniquebucketofearthquakes
Copying file://earthquakes.csv [Content-Type=text/csv]...
Copying file://earthquakes.htm [Content-Type=text/html]...
Copying file://earthquakes.png [Content-Type=image/png]...
- [3 files][703.7 KiB/703.7 KiB]
Operation completed over 3 objects/703.7 KiB.
amminer@usgs:~/training-data-analyst/CPB100/lab2b$
```

6. GCP Cloud Storage #2 (IAM Roles)

- best practices for implementing least-privileges on Google Cloud is to set the access scope to allow the entire platform, but to create service accounts with the minimal roles and permissions attached to them. ✓

7. Create Service Account

- ```
gcloud iam service-accounts create gcs-lab
gcloud projects add-iam-policy-binding ${GOOGLE_CLOUD_PROJECT} \
 --member serviceAccount:gcs-lab@${GOOGLE_CLOUD_PROJECT}.iam.gserviceaccount.com \
 --role roles/storage.objectViewer
```

 ✓

## 8. Create Compute Engine VM

- Make a VM for the lab:  

```
gcloud compute instances create gcs-lab-vm \
 --machine-type e2-medium --zone us-west1-b \
 --image-project ubuntu-os-cloud --image-family ubuntu-2004-lts \
 --scopes cloud-platform \
 --service-account
gcs-lab@${GOOGLE_CLOUD_PROJECT}.iam.gserviceaccount.com
```

 ✓

## 9. Service account roles (Compute)

- ssh in and attempt to list gcloud compute instances. What is the exact error message that is returned?

**ERROR: (gcloud.compute.instances.list) Some requests did not succeed:**

**- Required 'compute.instances.list' permission for 'projects/cloud-miner-amminer'**

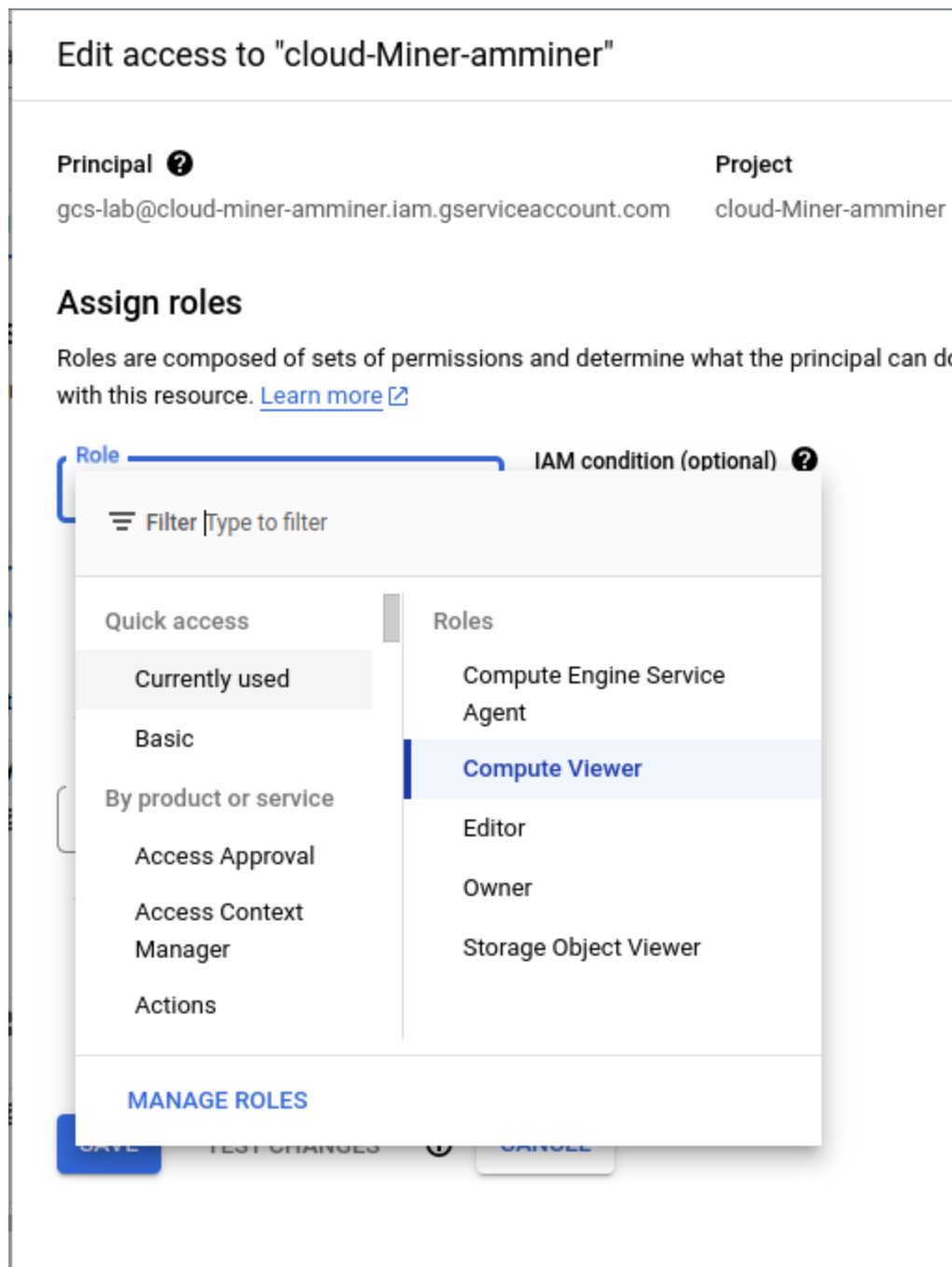
```
amminer@gcs-lab-vm:~$ gcloud compute instances list
ERROR: (gcloud.compute.instances.list) Some requests did not succeed:
- Required 'compute.instances.list' permission for 'projects/cloud-miner-amminer'

amminer@gcs-lab-vm:~$ ^C
amminer@gcs-lab-vm:~$
```

- What role needs to be added to the service account's permissions for the VM to have access to list the project's Compute Engine instances?

**Compute Viewer**





- Add the role and save the changes. Go back to the VM and repeat the command until it succeeds.

```
amminer@gcs-lab-vm:~$ gcloud compute instances list
NAME ZONE MACHINE_TYPE PREEMPTIBLE INTERNAL_IP EXTERNAL_IP STATUS
course-vm-usw1a us-west1-a e2-medium 10.138.0.8
gcs-lab-vm us-west1-a e2-medium 10.138.0.13 34.82.94.116 RUNNING
usgs us-west1-a e2-medium 10.138.0.11 TERMINATED
course-vm us-west1-b e2-medium 10.138.0.2 TERMINATED
amminer@gcs-lab-vm:~$
```

## 10. Service Account Roles (Storage)

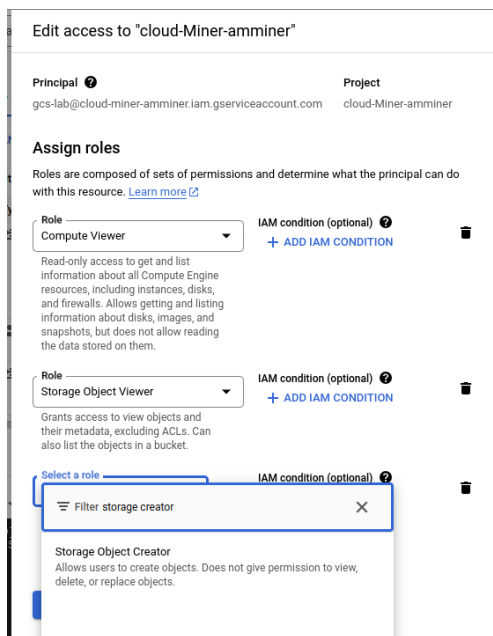
- use the `gsutil` command to copy the earthquake image file in the previous lab from the storage bucket onto the VM. Rename the file to a different name and then attempt to copy it back into the bucket. What error message is returned?

**AccessDeniedException: 403 gcs-lab@cloud-miner-amminer.iam.gserviceaccount.com does not have storage.objects.create access to the Google Cloud Storage object. Permission 'storage.objects.create' denied on resource (or it may not exist).**

```
amminer@gcs-lab-vm:~$ gsutil cp moonquakes.png gs://$bkt
Copying file://moonquakes.png [Content-Type=image/png]...
AccessDeniedException: 403 gcs-lab@cloud-miner-amminer.iam.gserviceaccount.com does not have storage.objects.create access to the Google Cloud Storage object. Permission 'storage.objects.create' denied on resource (or it may not exist).
```

- What role needs to be added to the service account's permissions for the VM to have access to add an object to a storage bucket?

**In order for a VM to add (but not replace) to a bucket it can be given the Storage Object Creator role. To replace as well as add I think we would need to apply Storage Admin instead.**



- Add the role and save the changes. Go back to the VM and repeat the `gsutil` command until it succeeds.

```
amminer@gcs-lab-vm:~$ gsutil cp moonquakes.png gs://$bkt
Copying file://moonquakes.png [Content-Type=image/png]...
/ [1 files][316.3 KiB/316.3 KiB]
Operation completed over 1 objects/316.3 KiB.
amminer@gcs-lab-vm:~$
```

## 11. GCP Cloud Storage #3 (Python)


- Bring up a Cloud Shell session and download an image of your choice by filling in a number (00 to 19) and storing it in image.jpg:

```
wget -O image.jpg http://chi-ni.com/motd/<NUM>.jpg
```

- Set up a Python environment and install the Google Cloud SDK's storage package (google-cloud-storage).
- Launch a Python 3 interpreter.

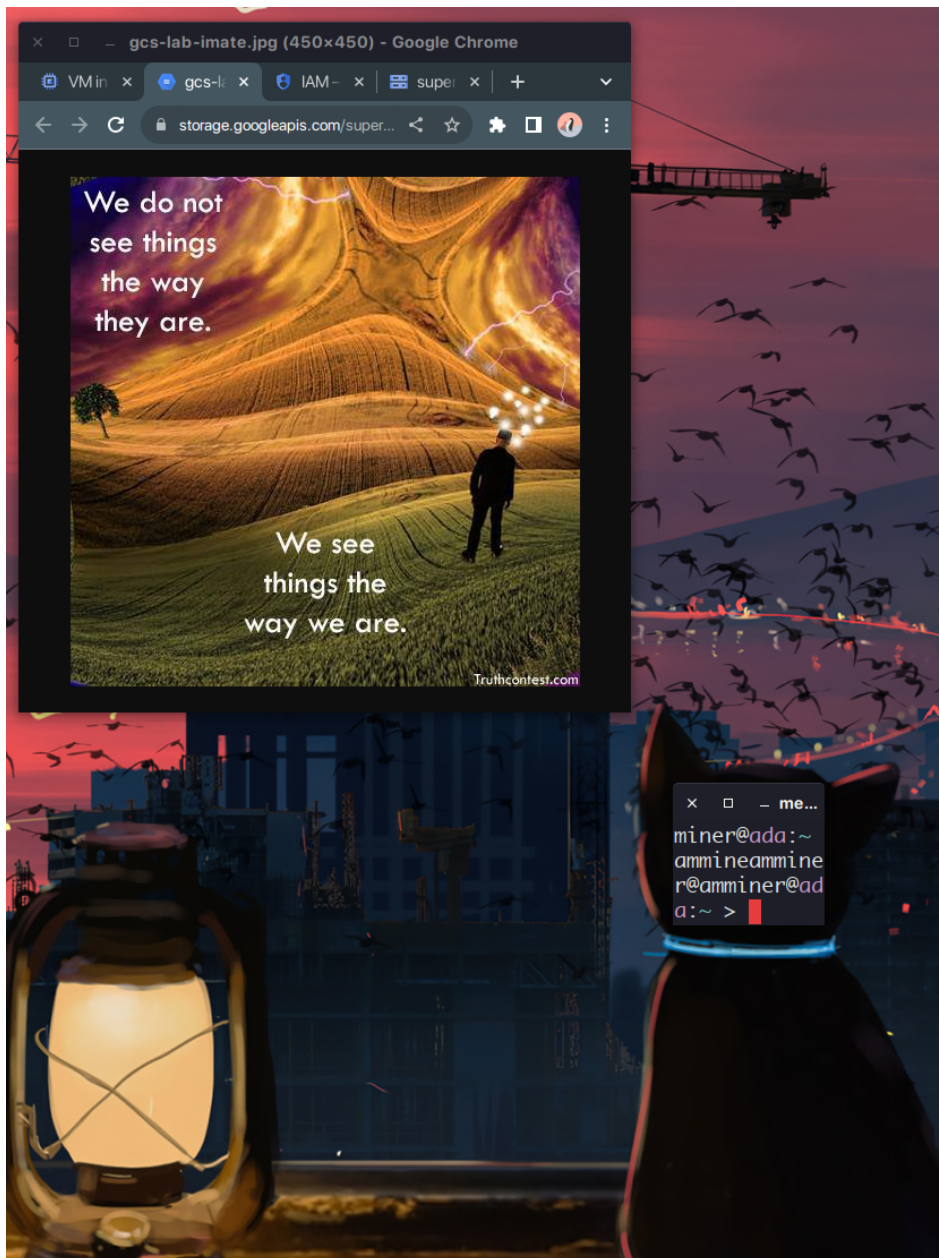


## 12. Python Storage Code

- In the python interpreter, get a google storage client, then ask the client for the bucket, then create a blob in the bucket and upload the raw binary of the image file. Make the blob public and get its URL. 

### 13. View Object

- View the image at its URL and delete it.



### 14. Clean Up




### 15. IAM and Least Privileges


- Due to deployment issues this section is optional - I may come back to it later.

## II. Lab 5.2a - DynamoDB Guestbook


### 1. DynamoDB

- In your local Ubuntu VM checkout the course repository and change into the code directory.  
  
**I decided to stop using the kali VM I had set up in gcloud and use an actual local Kubuntu vm for this lab.**


### 2. model\_dynamodb

- Examine the new model source code.  


### 3. Version 1: Ubuntu VM Python

- `boto3` requires a valid set of AWS credentials in order to authenticate properly to the backend DynamoDB instance of your AWS account. Unfortunately, our classroom environment will not allow us to create credentials of our own that have least privileges. We will instead use the full credentials of the account we are given.  


### 4. Obtain AWS Credentials

- Set up your keys as environment variables & `export AWS_DEFAULT_REGION=us-east-1`.  


## 5. Run the Application

- Sign it as per the lab instructions

The screenshot shows a terminal window titled "05\_aws\_dynamodb : python — Konsole" with a menu bar (File, Edit, View, Bookmarks, Plugins, Settings, Help). The terminal output includes environment variables like SHLVL=1, XDG\_VTNR=2, XDG\_SESSION\_ID=3, and various paths. It shows the user 'meelz' logging in and then a series of status messages: "[1]+ Stopped (env) meelz", "vim keyvars.", "[1]+ Stopped (env) meelz", "vim keyvars.", "[1]+ Stopped (env) meelz", "KsSFvqqu2Qb6", "5W3M0bSv0UBI", "(env) meelz", "AWS\_DEFAULT", "AWS\_SECRET\_A", "AWS\_ACCESS\_K", "AWS\_SESSIONL", "JZXnrQWnlug2", "mcrDxFbGRuZ6", "(env) meelz", "\* Serving f", "\* Debug mod", "WARNING: TH", "\* Running c", "\* Running c", "\* Running c", "Press CTRL+C", "\* Restarti", "\* Debugger", "\* Debugger", "127.0.0.1 -", "127.0.0.1 -", "127.0.0.1 - [29/Oct/2023 09:44:05] \"GET /favicon.ico HTTP/1.1\" 404 -", "127.0.0.1 - [29/Oct/2023 09:45:22] \"GET /sign/ HTTP/1.1\" 200 -", "127.0.0.1 - [29/Oct/2023 09:45:22] \"GET /static/style.css HTTP/1.1\" 304 -", "127.0.0.1 - [29/Oct/2023 09:45:41] \"POST /sign/ HTTP/1.1\" 302 -", "127.0.0.1 - [29/Oct/2023 09:45:41] \"GET / HTTP/1.1\" 200 -", "127.0.0.1 - [29/Oct/2023 09:45:41] \"GET /static/style.css HTTP/1.1\" 304 -", and an empty prompt "[]".

Overlaid on the terminal is a web browser window titled "My Visitors" with the address bar showing "localhost:5000". The browser displays a "Guestbook" page with a "Sign [here](#)" link, an "Entries" section, and a single entry: "Amelia <amminer@pdx.edu> signed on 2023-10-29 09:45:40.914621 Hello DynamoDB".

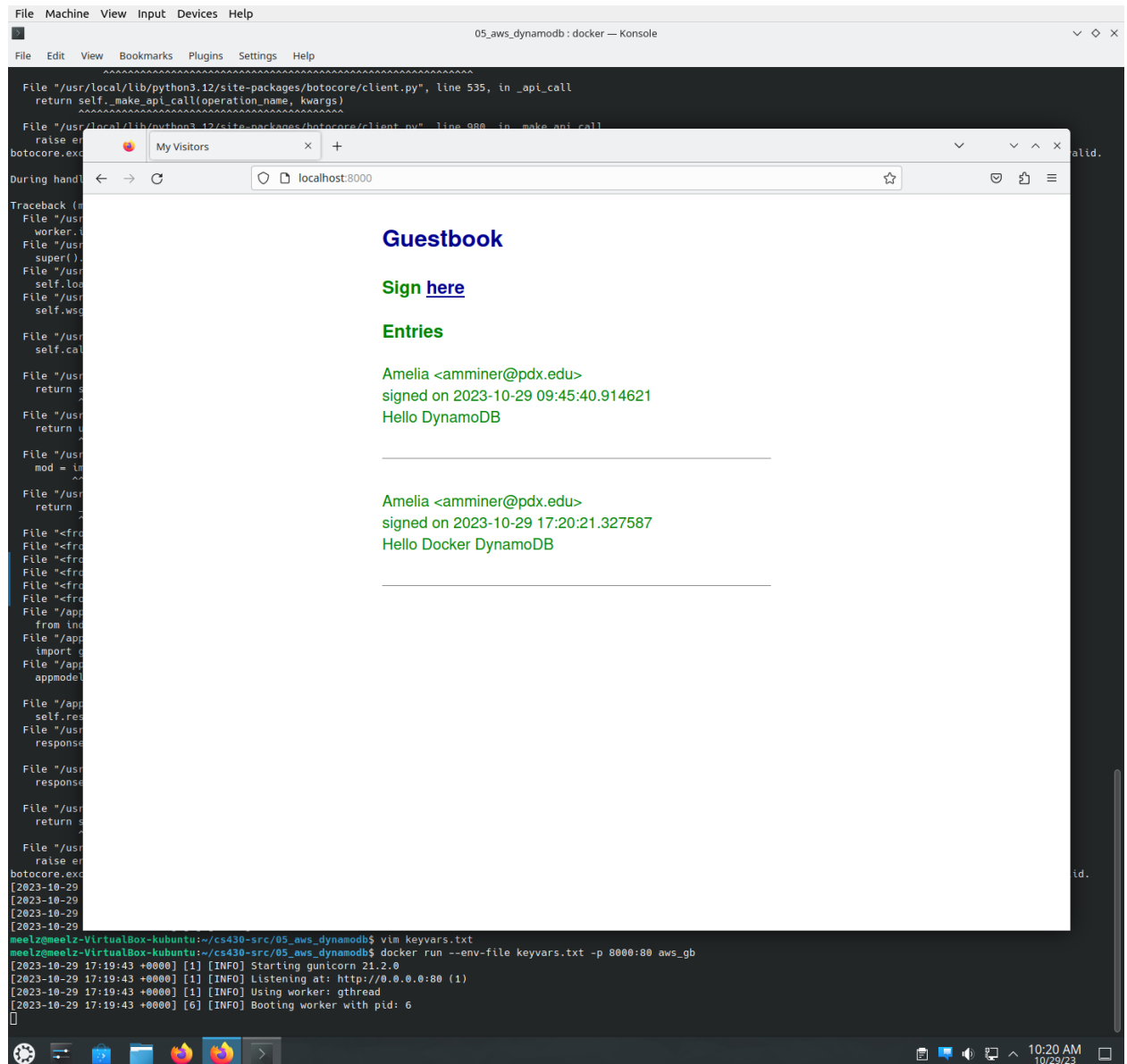
## 6. Version 2: Ubuntu VM Docker

- Examine the Dockerfile and .dockerignore



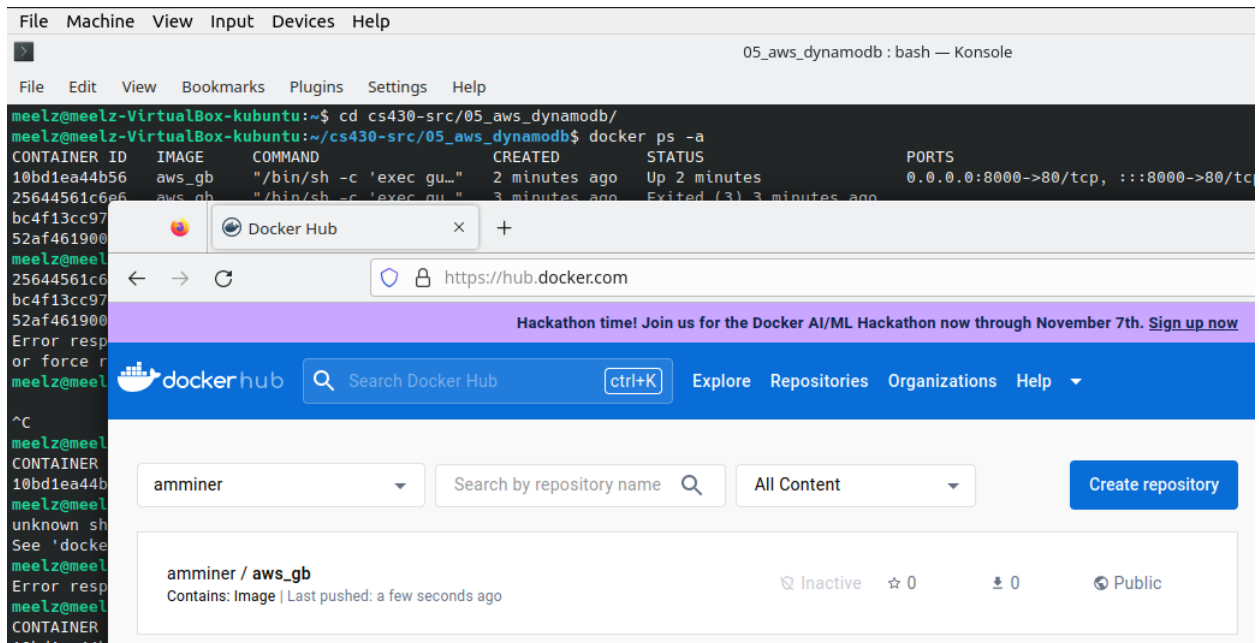
## 7. Run the Application

- Build the container. Run it, passing in the env vars etc. like so:  
`docker run --env AWS_ACCESS_KEY_ID=ASIA...32F --env AWS_SECRET_ACCESS_KEY=p22Z...C0e --env AWS_DEFAULT_REGION=us-east-1 --env AWS_SESSION_TOKEN=FwoG...A== --env PORT=80 -p 8000:80 aws_gb`
- Bring up a browser on your Ubuntu VM and visit the application. Sign it again.



## 8. Push the Container Image

- Take a screenshot of the container image on DockerHub.



## 9. Version 3: AWS Cloud9 IDE

- Create a new cloud9 instance and take note of its external IP.

44.192.132.183

## 10. Configure the Security Group

- Add an inbound rule to allow any IPv4 TCP traffic on port 5000.





## 11. Run the Application

- Set up the environment and run the application. Visit it. Sign it with “Hello Cloud9”.

⚠ Not secure | 44.192.132.183:5000

### Guestbook

[Sign here](#)

#### Entries

Amelia <amminer@pdx.edu>  
signed on 2023-10-29 09:45:40.914621  
Hello DynamoDB

---

Amelia <amminer@pdx.edu>  
signed on 2023-10-29 17:20:21.327587  
Hello Docker DynamoDB

---

Amelia Miner <amminer@pdx.edu>  
signed on 2023-10-29 23:08:38.137692  
Hello Cloud9

---

## 12. Version 4: AWS EC2

- Create an EC2 VM with the name guestbook, Ubuntu 64 bit, a key pair named awslab, and give it the IAM profile LabInstanceProfile. Launch it and note its IP.

34.230.87.40

## 13. Connect to the Instance

- The private key downloaded during setup as `awslab.pem`. Change its permissions to read-only by the owner and use it to ssh into the VM (`ssh -i awslab.pem ubuntu@<IP_address_of_EC2>`).



## 14. Set Up the Instance

- Run the guestbook image from dockerhub (`sudo docker run --env AWS_DEFAULT_REGION=us-east-1 -p 80:80 <dockerhub_id>/aws_gb`)



## 15. Visit the Application

- Visit the application via the EC2 instance's public IP and leave an entry that says "Hello EC2!".

⚠ Not secure | 34.230.87.40

### Guestbook

[Sign here](#)

#### Entries

Amelia <amminer@pdx.edu>  
signed on 2023-10-29 09:45:40.914621  
Hello DynamoDB

---

Amelia <amminer@pdx.edu>  
signed on 2023-10-29 17:20:21.327587  
Hello Docker DynamoDB

---

Amelia Miner <amminer@pdx.edu>  
signed on 2023-10-29 23:08:38.137692  
Hello Cloud9

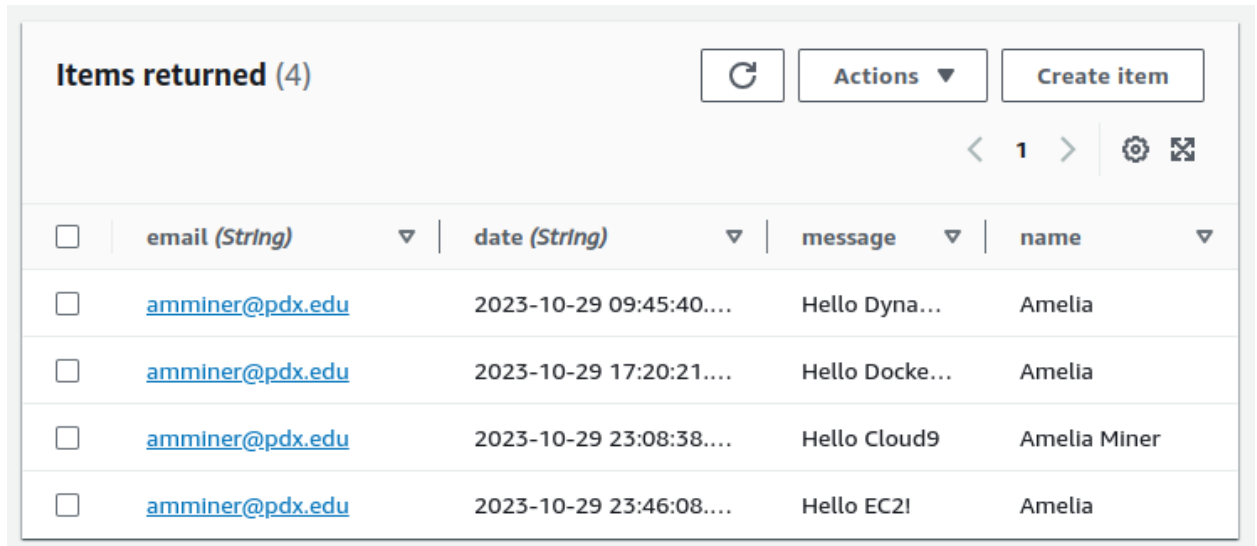
---

Amelia <amminer@pdx.edu>  
signed on 2023-10-29 23:46:08.954406  
Hello EC2!

---

## 16. View the Database

17. Take a screenshot that shows all of the guestbook entries that you added to the DynamoDB table including their timestamps in the web UI.



| Items returned (4)                                                |                                                      |                         |                |              |
|-------------------------------------------------------------------|------------------------------------------------------|-------------------------|----------------|--------------|
| <div><div>⌂</div><div>Actions ▼</div><div>Create item</div></div> |                                                      |                         |                |              |
| <div>&lt; 1 &gt; ⚙️ 🔍</div>                                       |                                                      |                         |                |              |
| <input type="checkbox"/>                                          | email (String) ▼                                     | date (String) ▼         | message ▼      | name ▼       |
| <input type="checkbox"/>                                          | <a href="mailto:amminer@pdx.edu">amminer@pdx.edu</a> | 2023-10-29 09:45:40.... | Hello Dyna...  | Amelia       |
| <input type="checkbox"/>                                          | <a href="mailto:amminer@pdx.edu">amminer@pdx.edu</a> | 2023-10-29 17:20:21.... | Hello Docke... | Amelia       |
| <input type="checkbox"/>                                          | <a href="mailto:amminer@pdx.edu">amminer@pdx.edu</a> | 2023-10-29 23:08:38.... | Hello Cloud9   | Amelia Miner |
| <input type="checkbox"/>                                          | <a href="mailto:amminer@pdx.edu">amminer@pdx.edu</a> | 2023-10-29 23:46:08.... | Hello EC2!     | Amelia       |

## 18. Clean Up

- Delete the Cloud9 environment, which will automatically delete the EC2 VM via the CloudFormation service.



## III. Lab 5.2g - Cloud Datastore Guestbook

### 1. Cloud Datastore

- in your local Ubuntu VM checkout the course repository and change into the code directory.



### 2. model\_datastore setup

- Edit model\_datastore.py to change YOUR\_PROJECT\_ID to point to your project. Note that the project id should be all in lowercase letters. It can be obtained within Cloud Shell



### 3. model\_datastore

- Review the model\_datastore.py code.



### 4. Datastore Setup

- Create a new DB in datastore in a us-west region.



### 5. 5. Version 1: Ubuntu VM Python

- Review the requirements.txt file. Set up the application environment.



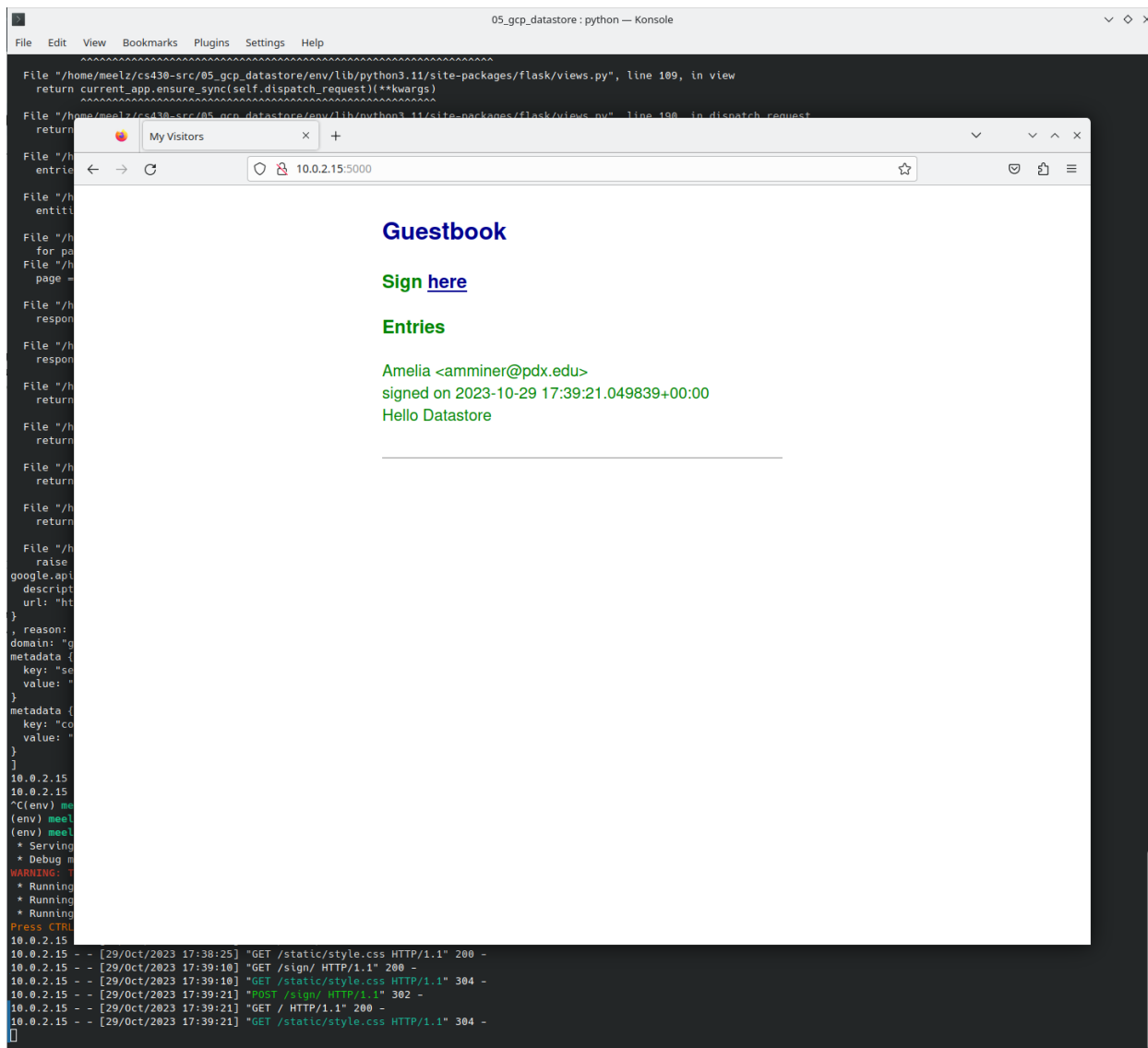
### 6. Obtain GCP Credentials

- Create the service account and give it a role that authorizes it to make changes to the datastore backend. We will pass the credentials of this account in when we run the app. Copy and paste the contents of the JSON file into a new local file on your Ubuntu VM named /tmp/guestbook-key.json and set the environment variable GOOGLE\_APPLICATION\_CREDENTIALS to this path.



## 7. Run the Application

- Run the app. Sign the guestbook with the string "Hello Datastore".



## 8. Version 2: Ubuntu VM Docker

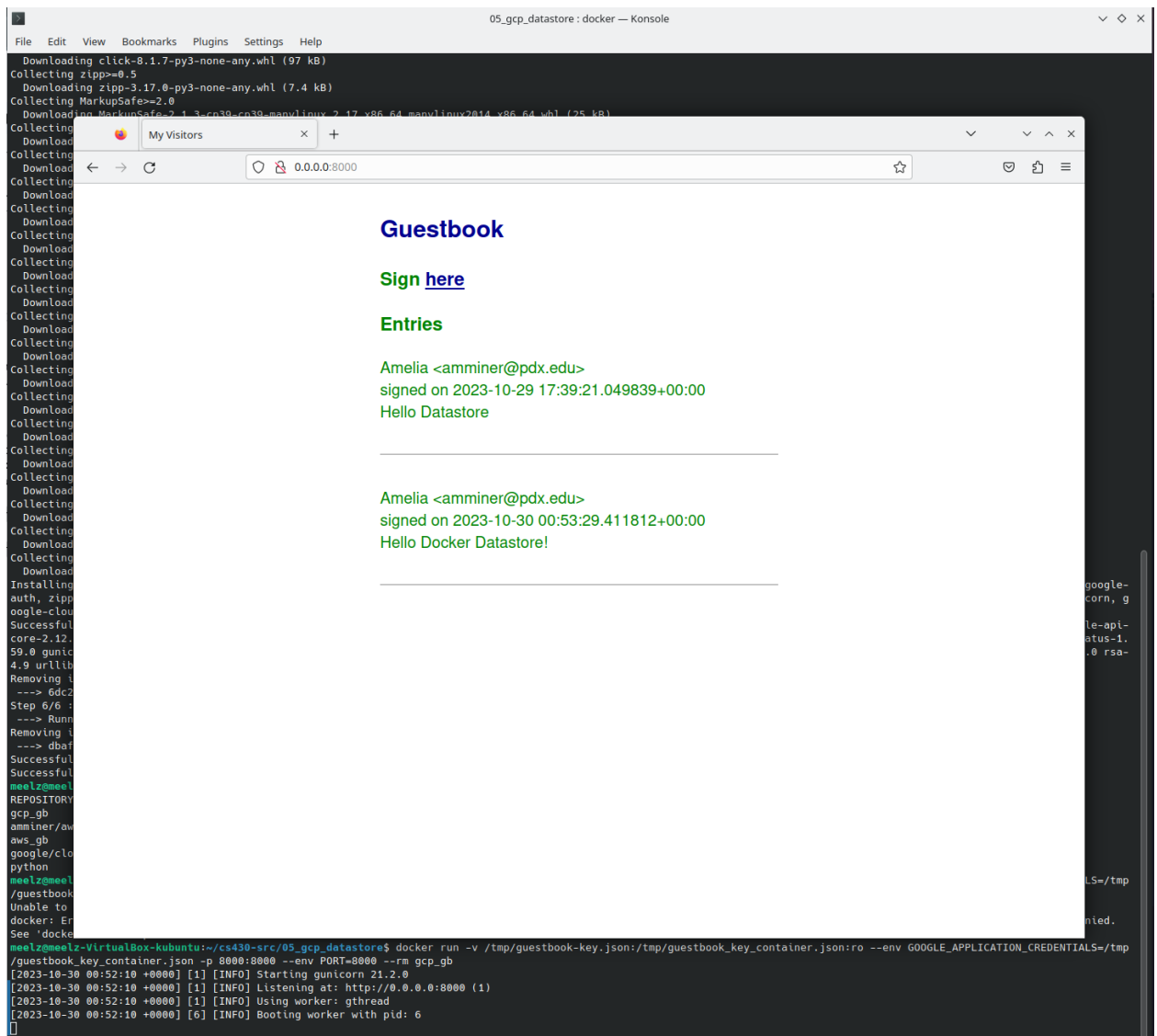
- Review the Dockerfile.



## 9. Run the Application

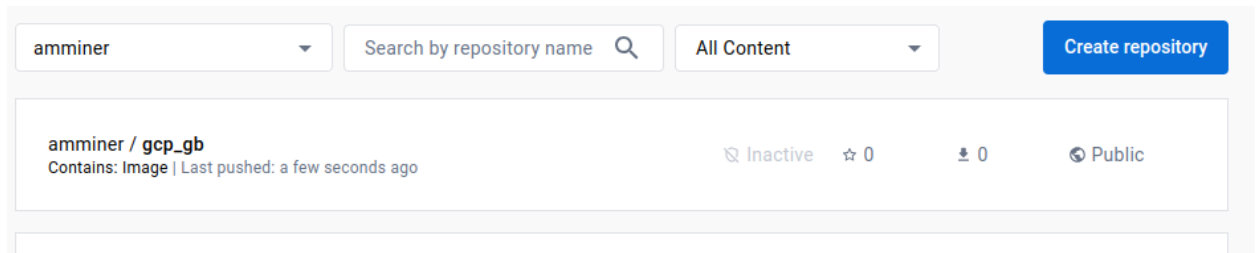
- Build the image tagged gcb-gb. For the app to run there needs to be an environment variable set for the key file. We don't want to copy the key file into the docker image for security reasons - it would be visible on Docker Hub.

- ```
docker run -v
/tmp/guestbook-key.json:/tmp/guestbook_key_container.json:ro
--env
GOOGLE_APPLICATION_CREDENTIALS=/tmp/guestbook_key_container.json
-p 8000:8000 --env PORT=8000 --rm gcp_gb
```



10. Push the Container Image

- Re-tag the container with your Docker ID and push it. Show the container on Docker Hub.



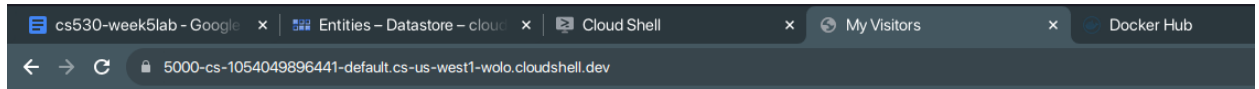
11. Version 3: GCP Cloud Shell

- In cloud shell, clone into the class repo and cd into the gcp datastore directory. Launch a full window version of the cloud shell. Use the editor to change model_datastore.py to point the Datastore client to your own project.



12. Run the Application

- Set the environment up using `virtualenv -p python3 env` instead of the usual `venv` command. Run the app. Using the cloud shell editor's web preview function, visit the site at port 5000. Sign the guestbook with "Hello Cloud Shell!".



Guestbook

[Sign here](#)

Entries

Amelia <amminer@pdx.edu>
signed on 2023-10-30 01:05:00.939198+00:00
Hello Cloud Shell!

Amelia <amminer@pdx.edu>
signed on 2023-10-29 17:39:21.049839+00:00
Hello Datastore

Amelia <amminer@pdx.edu>
signed on 2023-10-30 00:53:29.411812+00:00
Hello Docker Datastore!

13. Version 4: GCP Compute Engine

- Create a compute engine instance with limited permissions for the app's needs. We can use the guestbook service account from earlier and attach the Cloud Datastore User role. Note that we could do the same with the default service account for CE instances.




14. Set Up the Instance

- ssh into the instance. Install docker and run the gcp_gb image from Docker Hub.



15. Visit the Application

- Visit the site using the instance's IP address via `http://<IP_address_of_ComputeEngine_instance>` and sign the guestbook with the string "Hello Compute Engine!".

 Not secure | 35.247.74.112

Guestbook

[Sign here](#)

Entries

Amelia <amminer@pdx.edu>
signed on 2023-10-30 01:05:00.939198+00:00
Hello Cloud Shell!

Amelia <amminer@pdx.edu>
signed on 2023-10-29 17:39:21.049839+00:00
Hello Datastore

Amelia <amminer@pdx.edu>
signed on 2023-10-30 01:35:11.596005+00:00
Hello Compute Engine!

Amelia <amminer@pdx.edu>
signed on 2023-10-30 00:53:29.411812+00:00
Hello Docker Datastore!

16. View the Database

- Take a screenshot of all of the entries that have been added including their timestamps in the Cloud Datastore UI.

Query results					
<input type="checkbox"/>	Name/ID ↑	date	email	message	name
<input type="checkbox"/>	id=5632499082330112	October 29, 2023 at 6:05:00.939 PM UTC-7	amminer@pdx.edu	Hello Cloud Shell!	Amelia
<input type="checkbox"/>	id=5634161670881280	October 29, 2023 at 10:39:21.049 AM UTC-7	amminer@pdx.edu	Hello Datastore	Amelia
<input type="checkbox"/>	id=5642368648740864	October 29, 2023 at 6:35:11.596 PM UTC-7	amminer@pdx.edu	Hello Compute Engine!	Amelia
<input type="checkbox"/>	id=5644004762845184	October 29, 2023 at 5:53:29.411 PM UTC-7	amminer@pdx.edu	Hello Docker Datastore!	Amelia

17. Clean Up

