

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

TABLE OF CONTENTS:

<b>I. Lab 7.1a: Terraform AWS Guestbook.....</b>	<b>3</b>
1. Terraform.....	3
2. Setup.....	3
3. Initial Configuration.....	3
4. Launching Configuration.....	3
5. Adding Network Access.....	4
6. Adding ssh access.....	4
7. Adding the guestbook application.....	4
8. View the guestbook.....	4
9. Clean up.....	4
<b>II. Lab 7.1g: Terraform GCP Guestbook.....</b>	<b>4</b>
1. Terraform.....	4
2. Setup.....	4
3. Initial Configuration.....	4
4. Launching Configuration.....	5
5. Adding an external IP Address.....	6
6. Adding ssh access.....	6
7. Adding the guestbook application.....	7
8. View the Guestbook.....	8
9. Clean up.....	8
<b>III. Lab 7.2g: Kubernetes Guestbook.....</b>	<b>9</b>
1. Kubernetes.....	9
2. Setup.....	9
3. Assigning Privileges.....	9
4. Create Kubernetes Cluster.....	9
5. Prepare a container image.....	11
6. kubernetes.yaml.....	11
7. Deploy the configuration.....	12
8. view the guestbook.....	13
9. delete workload and service.....	15
10. CI/CD build automation.....	16
11. Configure build automation.....	16
12. Deploy and view application.....	17
13. Clean up.....	18
<b>IV. Lab 7.3g: APIs (Slack, Knowledge Graph).....</b>	<b>18</b>
1. Slack and Knowledge Graph Integration.....	18
2. Code.....	18
3. Code.....	18

4. Knowledge Graph Setup.....	20
5. Create a Slack Workspace.....	20
6. Configure and Deploy.....	20
7. Create Slack Command.....	20
8. Test the command.....	21
<b>V. Lab 7.4g: ML APIs.....</b>	<b>21</b>

# I. Lab 7.1a: Terraform AWS Guestbook

## 1. Terraform



## 2. Setup

- In AWS cloud shell, install yum-utils, add the hashicorp repo, and install terraform. Make a directory for your terraform deployment and cd in.

## 3. Initial Configuration

- Examine and copy over the main.tf file contents from the lab.

## 4. Launching Configuration

- init, plan, and apply the terraform configuration. Show the output of the apply command.

```
[cloudshell-user@ip-10-2-5-15 tf]$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.guestbook will be created
+ resource "aws_instance" "guestbook" {
+   ami                    = "ami-04b107e90218672e5"
+   arn                    = (known after apply)
+   associate_public_ip_address = (known after apply)
+   availability_zone       = (known after apply)
+   cpu_core_count          = (known after apply)
+   cpu_threads_per_core    = (known after apply)
+   disable_api_stop        = (known after apply)
+   disable_api_termination = (known after apply)
+   ebs_optimized           = (known after apply)
+   get_password_data       = false
+   host_id                 = (known after apply)
+   host_resource_group_arn = (known after apply)
+   iam_instance_profile    = (known after apply)
+   id                      = (known after apply)
+   instance_initiated_shutdown_behavior = (known after apply)
+   instance_lifecycle      = (known after apply)
+   instance_state          = (known after apply)
+   instance_type           = "t2.micro"
+   ipv6_address_count      = (known after apply)
+   ipv6_addresses          = (known after apply)
+   key_name                = (known after apply)
+   monitoring              = (known after apply)
+   outpost_arn             = (known after apply)
+   password_data           = (known after apply)
+   placement_group         = (known after apply)
+   placement_partition_number = (known after apply)
+   primary_network_interface_id = (known after apply)
+   private_dns             = (known after apply)
+   private_ip              = (known after apply)
+   public_dns              = (known after apply)
+   public_ip               = (known after apply)
+   secondary_private_ips   = (known after apply)
+   security_groups          = (known after apply)
+   source_dest_check       = true
+   spot_instance_request_id = (known after apply)
+   subnet_id               = (known after apply)
+   tags_all                = (known after apply)
+   tenancy                 = (known after apply)
+   user_data               = (known after apply)
+   user_data_base64        = (known after apply)
+   user_data_replace_on_change = false
+   vpc_security_group_ids  = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ ec2instance = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.guestbook: Creating...
aws_instance.guestbook: Still creating... [10s elapsed]
aws_instance.guestbook: Still creating... [20s elapsed]
aws_instance.guestbook: Still creating... [30s elapsed]
aws_instance.guestbook: Creation complete after 32s [id=i-00f0a8837458956aa]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
ec2instance = "34.207.181.40"
[cloudshell-user@ip-10-2-5-15 tf]$ amminer
```

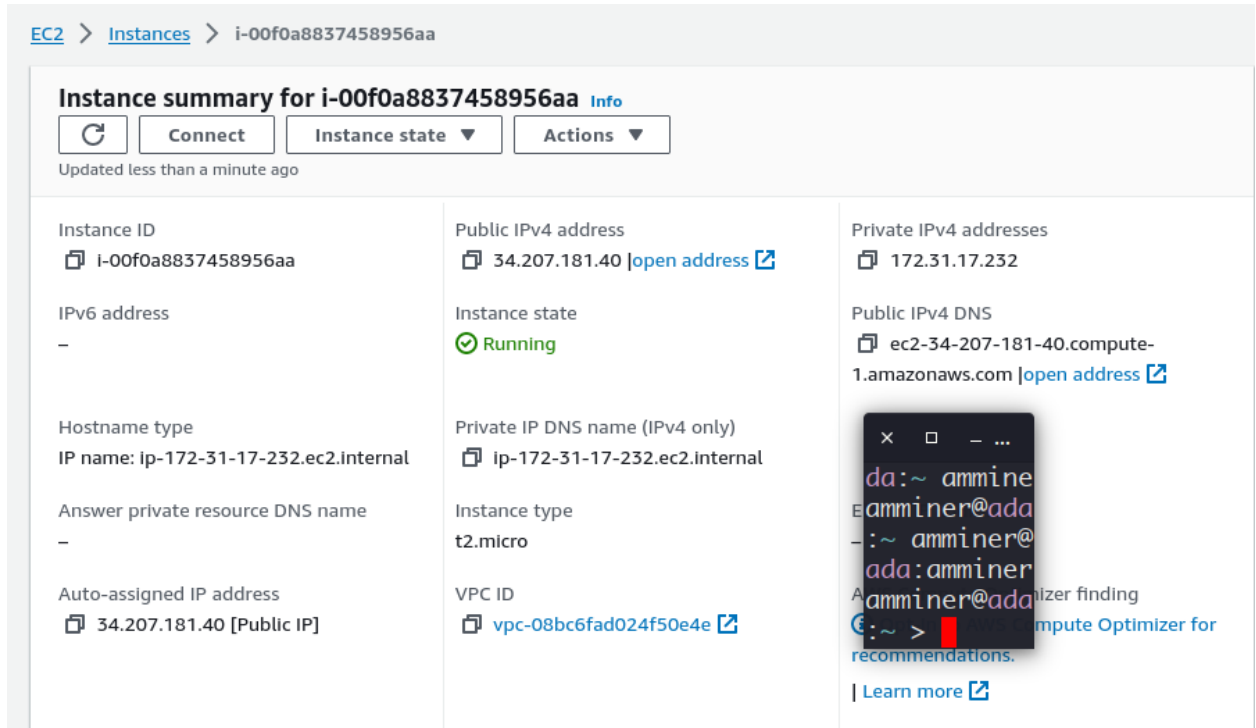
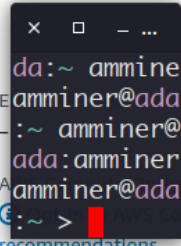
- Visit EC2 within the web console and refresh it to see that the IP address has been bound to the VM

EC2 > Instances > i-00f0a8837458956aa

### Instance summary for i-00f0a8837458956aa [Info](#)

[Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Updated less than a minute ago

<b>Instance ID</b> i-00f0a8837458956aa	<b>Public IPv4 address</b> 34.207.181.40 <a href="#">open address</a>	<b>Private IPv4 addresses</b> 172.31.17.232
<b>IPv6 address</b> -	<b>Instance state</b>  <b>Running</b>	<b>Public IPv4 DNS</b> ec2-34-207-181-40.compute-1.amazonaws.com <a href="#">open address</a>
<b>Hostname type</b> IP name: ip-172-31-17-232.ec2.internal	<b>Private IP DNS name (IPv4 only)</b> ip-172-31-17-232.ec2.internal	 <pre>da:~ ammine E amminer@ada -:~ amminer@ ada:amminer A amminer@ada €:~ &gt; </pre>
<b>Answer private resource DNS name</b> -	<b>Instance type</b> t2.micro	
<b>Auto-assigned IP address</b> 34.207.181.40 [Public IP]	<b>VPC ID</b> vpc-08bc6fad024f50e4e	

[Learn more](#)

## 5. Adding Network Access

- Copy the new blocks into the tf file, plan, and apply.



## 6. Adding ssh access

- Generate a keypair, add it to the config, redeploy, log in.

```
[cloudshell-user@ip-10-2-5-15 tf]$ ssh ubuntu@18.212.236.44
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1049-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Nov 14 06:25:43 UTC 2023

System load:  0.08               Processes:            97
Usage of /:   21.0% of 7.57GB    Users logged in:     0
Memory usage: 21%               IPv4 address for eth0: 172.31.24.188
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-24-188:~$ amminer
```

## 7. Adding the guestbook application

- copy the code from the lab in, plan, and apply.

```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee it takes exactly these actions if you run "terraform apply" now.
[cloudshell-user@ip-10-2-5-15 tf]$ terraform apply
aws_key_pair.kp: Refreshing state... [id=guestbook-key]
aws_security_group.sg-guestbook: Refreshing state... [id=sg-069c232f6d28bed24]
aws_instance.guestbook: Refreshing state... [id=i-0def848d55256335f]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place
Terraform will perform the following actions:

# aws_instance.guestbook will be updated in-place
~ resource "aws_instance" "guestbook" {
  id           = "i-0def848d55256335f"
  tags        = {}
  user_data    = "5e87ec158e53b8a9d26be3345f57b79757bd125b"
  ~ # (5 unchanged attributes hidden)
}
# (8 unchanged blocks hidden)

Plan: 0 to add, 1 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes
aws_instance.guestbook: Modifying... [id=i-0def848d55256335f]
aws_instance.guestbook: Still modifying... [id=i-0def848d55256335f, 10s elapsed]
aws_instance.guestbook: Still modifying... [id=i-0def848d55256335f, 20s elapsed]
aws_instance.guestbook: Still modifying... [id=i-0def848d55256335f, 30s elapsed]
aws_instance.guestbook: Still modifying... [id=i-0def848d55256335f, 40s elapsed]
aws_instance.guestbook: Still modifying... [id=i-0def848d55256335f, 50s elapsed]
aws_instance.guestbook: Still modifying... [id=i-0def848d55256335f, 1m0s elapsed]
aws_instance.guestbook: Modifications complete after 1m11s [id=i-0def848d55256335f]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

Outputs:
ec2instance = "18.212.236.44"
[cloudshell-user@ip-10-2-5-15 tf]$ amminer
```

- **ssh** in and perform a process listing until the **gunicorn** process appears. ✓

## 8. View the guestbook

- Sign it.



### Guestbook

Sign [here](#)

#### Entries

Amelia <amminer@pdx.edu>  
signed on 2023-11-14  
Hello Terraform on AWS!

---

## 9. Clean up



## II. Lab 7.1g: Terraform GCP Guestbook


### 1. Terraform



### 2. Setup



### 3. Initial Configuration

- Copy the main.tf code. Substitute in the fmi and your project name. 

## 4. Launching Configuration

- Init, plan and apply. Show the VM's IP addresses in the CE web console.

Google Cloud cloud-Miner-amminer  [Search](#) 6

VM instances [CREATE INSTANCE](#) [IMPORT VM](#) [REFRESH](#) [LEARN](#)

[INSTANCES](#) [OBSERVABILITY](#) [INSTANCE SCHEDULES](#)

VM instances

Filter

<input type="checkbox"/>	Status	Name	Zone	Recommendations	Internal IP	External IP	Network	Connect	
<input type="checkbox"/>		<a href="#">course-vm</a>	us-west1-b		10.138.0.2 <a href="#">(nic0)</a>		<a href="#">default</a>	SSH	
<input type="checkbox"/>		<a href="#">course-vm-usw1a</a>	us-west1-a		10.138.0.8 <a href="#">(nic0)</a>		<a href="#">default</a>	SSH	
<input type="checkbox"/>		<a href="#">tf-lab-vm</a>	us-west1-b		10.138.0.16 <a href="#">(nic0)</a>		<a href="#">default</a>	SSH	

## 5. Adding an external IP Address

- Copy the code in from the lab to add an external IP to the machine. Plan. Apply. Show the output.

```
Changes to Outputs:
+ ip = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

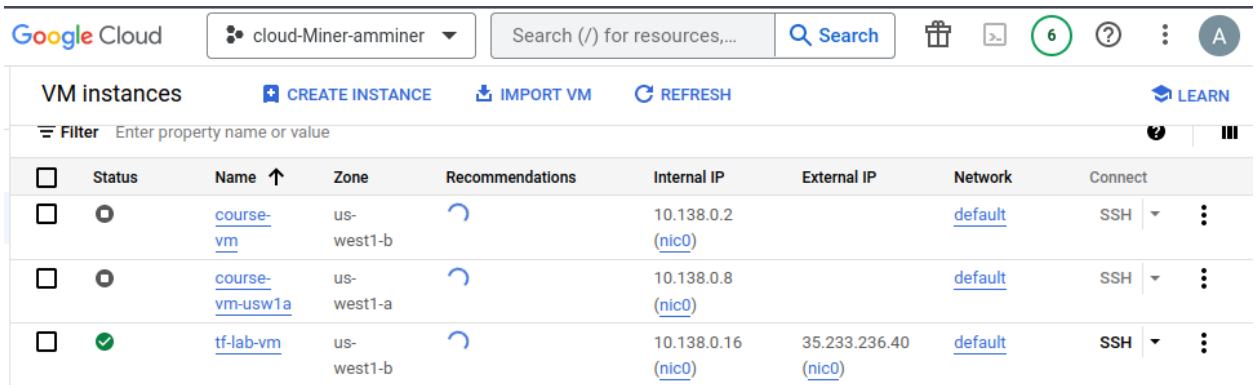
google_compute_address.static: Creating...
google_compute_address.static: Still creating... [10s elapsed]
google_compute_address.static: Creation complete after 11s [id=projects/cloud-miner-amminer/regions/us-west1/addresses/ipv4-address]
google_compute_instance.default: Modifying... [id=projects/cloud-miner-amminer/zones/us-west1-b/instances/tf-lab-vm]
google_compute_instance.default: Still modifying... [id=projects/cloud-miner-amminer/zones/us-west1-b/instances/tf-lab-vm, 10s elapsed]
google_compute_instance.default: Modifications complete after 11s [id=projects/cloud-miner-amminer/zones/us-west1-b/instances/tf-lab-vm]

Apply complete! Resources: 1 added, 1 changed, 0 destroyed.

Outputs:

ip = "35.233.236.40"
amminer@cloudshell:~/tf (cloud-miner-amminer)$
```

- Visit the Compute Engine web console and refresh it to see that the IP address has been bound to the VM. Take a screenshot that includes the VM's IP addresses.



The screenshot shows the Google Cloud VM instances page for the project 'cloud-Miner-amminer'. It displays a table of VM instances with columns for Status, Name, Zone, Recommendations, Internal IP, External IP, Network, and Connect. The 'tf-lab-vm' instance is highlighted, showing it has an external IP address of 35.233.236.40.

Status	Name	Zone	Recommendations	Internal IP	External IP	Network	Connect
Running	course-vm	us-west1-b	Recommendations	10.138.0.2 (nic0)		default	SSH
Running	course-vm-usw1a	us-west1-a	Recommendations	10.138.0.8 (nic0)		default	SSH
Running	tf-lab-vm	us-west1-b	Recommendations	10.138.0.16 (nic0)	35.233.236.40 (nic0)	default	SSH



## 6. Adding ssh access

- Create ssh keys for the cloud console and add the key file to the terraform config. ssh in from cloud shell and show the successful log in.

### Outputs:

```
ip = "35.199.159.174"
amminer@cloudshell:~/tf (cloud-miner-amminer)$ ssh 35.199.159.174
The authenticity of host '35.199.159.174 (35.199.159.174)' can't be established.
ECDSA key fingerprint is SHA256:NUP0C0E7oZBeukZAffp+udVRyIzVMBqmt0IGKAllIWs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.199.159.174' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1046-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Nov 14 07:56:24 UTC 2023

System load:  0.7               Processes:            107
Usage of /:   19.5% of 9.51GB   Users logged in:     0
Memory usage: 5%               IPv4 address for ens4: 10.138.0.25
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

amminer@tf-lab-vm:~$
```

## 7. Adding the guestbook application

- Copy the code in from the lab. Plan. What resources are being added changed or destroyed?

**Only the VM resource is being changed; many of its properties are changing, adding, or being deleted. See the screenshot...**

```
amminer@cloudshell:~/tf (cloud-miner-amminer)$ terraform plan
google_compute_address.static: Refreshing state... [id:projects/cloud-miner-amminer/regions/us-west1/addresses/ipv4-address]
google_compute_instance.default: Refreshing state... [id:projects/cloud-miner-amminer/zones/us-west1-a/instances/tf-lab-vm]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
/+ destroy and then create replacement

Terraform will perform the following actions:

# google_compute_instance.default must be replaced
/+ resource "google_compute_instance" "default" {
  ~ cpu_platform      = "Intel Broadwell" -> (known after apply)
  ~ current_status    = "RUNNING" -> (known after apply)
  ~ effective_labels   = {} -> (known after apply)
  ~ enable_display     = false -> null
  ~ guest_accelerator = [] -> (known after apply)
  ~ id                = "projects/cloud-miner-amminer/zones/us-west1-a/instances/tf-lab-vm" -> (known after apply)
  ~ instance_id       = "672168867105227011" -> (known after apply)
  ~ label_fingerprint = "42mmsp88r5m=" -> (known after apply)
  ~ labels             = {}
  ~ metadata_fingerprint = "6lkoZo3UJ5Q=" -> (known after apply)
  ~ metadata_startup_script = <<EOT # forces replacement
    #!/bin/bash
    apt update
    apt install -yq build-essential python3-pip gunicorn git
    pip install flask
    git clone https://github.com/wu4f/cs430-src /root/cs430-src
    cd /root/cs430-src/83.nginx_gunicorn_certbot
    gunicorn --bind :80 --workers 1 --threads 8 app:app
  EOT
  ~ min_cpu_platform = (known after apply)
  ~ name              = "tf-lab-vm"
  ~ resource_policies = [] -> null
  ~ self_link         = "https://www.googleapis.com/compute/v1/projects/cloud-miner-amminer/zones/us-west1-a/instances/tf-lab-vm" -> (known after apply)
  ~ tags              = [
    ~ "http-server",
  ]
  ~ tags_fingerprint = "42mmsp88r5m=" -> (known after apply)
  ~ terraform_labels = {} -> (known after apply)
  # (6 unchanged attributes hidden)

  ~ boot_disk {
    ~ device_name       = "persistent-disk-0" -> (known after apply)
    ~ disk_encryption_key_sha256 = (known after apply)
    ~ kms_key_self_link = (known after apply)
    ~ source             = "https://www.googleapis.com/compute/v1/projects/cloud-miner-amminer/zones/us-west1-a/disks/tf-lab-vm" -> (known after apply)
    # (2 unchanged attributes hidden)
  }

  ~ initialize_params {
    ~ image              = "https://www.googleapis.com/compute/v1/projects/ubuntu-os-cloud/global/images/ubuntu-2004-focal-v20231101" -> "ubuntu-os-cloud/ubuntu-2004-focal-v20231101"
    ~ labels             = {} -> (known after apply)
    ~ resource_manager_tags = {} -> null
    ~ size               = 10 -> (known after apply)
    ~ type               = "pd-standard" -> (known after apply)
  }
}

~ network_interface {
  ~ internal_ipv6_prefix_length = 0 -> (known after apply)
  ~ ipv6_access_type            = (known after apply)
  ~ ipv6_address                = (known after apply)
  ~ name                       = "nic0" -> (known after apply)
  ~ network                    = "https://www.googleapis.com/compute/v1/projects/cloud-miner-amminer/global/networks/default" -> "default"
  ~ network_ip                 = "10.130.0.25" -> (known after apply)
  ~ queue_count                = 0 -> null
  ~ stack_type                 = "IPv4_ONLY" -> (known after apply)
  ~ subnetwork                 = "https://www.googleapis.com/compute/v1/projects/cloud-miner-amminer/regions/us-west1/subnetworks/default" -> (known after apply)
  ~ subnetwork_project         = "cloud-miner-amminer" -> (known after apply)
}

~ access_config {
  ~ network_tier = "PREMIUM" -> (known after apply)
  # (1 unchanged attribute hidden)
}

}

~ scheduling {
  ~ automatic_restart = true -> null
  ~ min_node_cpus     = 0 -> null
  ~ on_host_maintenance = "MIGRATE" -> null
  ~ preemptible        = false -> null
  ~ provisioning_model = "STANDARD" -> null
}

~ shielded_instance_config {
  ~ enable_integrity_monitoring = true -> null
  ~ enable_secure_boot         = false -> null
  ~ enable_vtpm                = true -> null
}
}

Plan: 1 to add, 0 to change, 1 to destroy.
```

- What part of the config forces the replacement?  
**the startup script. See above.**

## 8. View the Guestbook

- Sign it



### Guestbook

Sign [here](#)

#### Entries

Amelia <amminer@pdx.edu>  
signed on 2023-11-14  
Hello Terraform on GCP!

---

## 9. Clean up



### III. Lab 7.2g: Kubernetes Guestbook

#### 1. Kubernetes



#### 2. Setup



#### 3. Assigning Privileges



#### 4. Create Kubernetes Cluster

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

**gke-guestbook-default-pool-357f161d**

Google Cloud cloud-Miner-amminer Search (/) for resources, docs, products, and more

Instance temp... [CREATE INSTANCE TEMPLATE](#) [REFRESH](#) [CREATE VM](#) [CREATE](#)

Instance templates are saved VM configurations used to create identical VMs, either individually or as part of managed instance groups. [Learn more](#)

**Filter** Filter instance templates

<input type="checkbox"/>	Name ↑	Machine type	Image	Disk type	Location ?	Actions
<input type="checkbox"/>	<a href="#">gke-guestbook-default-pool-357f161d</a>	e2-medium	gke-1273-gke100-cos-105-17412-101-24-c-pre	Balanced persistent disk	global	⋮

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

**gke-guestbook-default-pool-357f161d-grp**

Google Cloud cloud-Miner-amminer Search (/) for resources, docs, products, and more

Instance groups CREATE INSTANCE GROUP REFRESH DELETE LEARN

Instance groups are collections of VM instances that use load balancing and automated services, like autoscaling and autohealing. [Learn more](#)

Filter Enter property name or value

Status	Name ↑	Instances	Template	Group type	Creation time	Recommendation	Autoscaling	Zone	In Use By
<input checked="" type="checkbox"/>	<a href="#">gke-guestbook-default-pool-357f161d-grp</a>	2	<a href="#">gke-guestbook-default-pool-357f161d</a>	Managed	Nov 12, 2023, 3:32:34 PM UTC-08:00		No configuration	us-west1-b	<a href="#">guestbook</a>

- What are the names of the two nodes?  
**gke-guestbook-default-pool-357f161d-2c1n and gke-guestbook-default-pool-357f161d-p7gz**

Google Cloud cloud-Miner-amminer Search (/) for resources, docs, products, and more

VM instances CREATE INSTANCE IMPORT VM REFRESH

INSTANCES OBSERVABILITY INSTANCE SCHEDULES

VM instances

Filter Enter property name or value

Status	Name ↑	Zone	Recommendations	Internal IP	External IP	Network	Connect
<input type="checkbox"/>	<a href="#">course-vm</a>	us-west1-b		10.138.0.2 (nic0)		<a href="#">default</a>	SSH
<input type="checkbox"/>	<a href="#">course-vm-usw1a</a>	us-west1-a		10.138.0.8 (nic0)		<a href="#">default</a>	SSH
<input checked="" type="checkbox"/>	<a href="#">gke-guestbook-default-pool-357f161d-2c1n</a>	us-west1-b		10.138.0.19 (nic0)	35.199.159.174 (nic0)	<a href="#">default</a>	SSH
<input checked="" type="checkbox"/>	<a href="#">gke-guestbook-default-pool-357f161d-p7gz</a>	us-west1-b		10.138.0.18 (nic0)	34.145.75.201 (nic0)	<a href="#">default</a>	SSH
<input checked="" type="checkbox"/>	<a href="#">tf-lab-vm</a>	us-west1-b		10.138.0.16 (nic0)	35.233.236.40 (nic0)	<a href="#">default</a>	SSH

Related actions

## 5. Prepare a container image

- Visit the Artifact Registry UI and navigate to its container images section. Show the container image created

Google Cloud cloud-Miner-amminer Search (/) for re... Search

Images DELETE

Container Registry is deprecated. After May 15, 2024, Artifact Registry will host images for the gcr.io domain in projects without previous Container Registry usage. [Learn more](#)

**gcp\_gb**

gcr.io > cloud-miner-amminer > gcp\_gb

Filter Enter property name or value

Name	Tags	Virtual Size	Created	Uploaded
<a href="#">26e0cf8d3541</a>	latest	1.1 GB	15 minutes ago	11 minutes ago

## 6. kubernetes.yaml

- Copy over the kubernetes configuration yaml, editing in your project ID. 

## 7. Deploy the configuration

- Obtain credentials from the cluster:

```
gcloud container clusters get-credentials guestbook --zone  
us-west1-b
```

Then deploy the configuration on the cluster.

```
kubectl create -f kubernetes.yaml
```

Note that the configuration file is portable and can run on any other cloud provider, locally, etc. Get the status of the pods running in the cluster.

```
kubectl get pods
```

Take a screenshot of the output of the following command when all 3 replicas reach a "Running" state.

**When I did this in the order the lab instructs, it appeared to run successfully but failed to bring up my pods. When I did it manually they failed at the container image pull stage. I inverted the order of the declarations so that the Service is specified at the top of the file and the ReplicationController at the bottom. This worked.**

```
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl create -f kubernetes.yaml  
replicationcontroller/guestbook-replicas created  
amminer@cloudshell:~ (cloud-miner-amminer)$ kubernetes get pods  
bash: kubernetes: command not found  
amminer@cloudshell:~ (cloud-miner-amminer)$ ^Ckubectl create -f kubernetes.yaml  
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl get pods  
NAME                READY   STATUS             RESTARTS   AGE  
guestbook-replicas-khdjf 0/1     ContainerCreating  0          35s  
guestbook-replicas-pz2ps 0/1     ContainerCreating  0          35s  
guestbook-replicas-vrws9 0/1     ContainerCreating  0          35s  
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl get pods  
NAME                READY   STATUS    RESTARTS   AGE  
guestbook-replicas-khdjf 1/1     Running   0          2m20s  
guestbook-replicas-pz2ps 1/1     Running   0          2m20s  
guestbook-replicas-vrws9 1/1     Running   0          2m20s  
amminer@cloudshell:~ (cloud-miner-amminer)$
```

- Then, find the service that is exported from the deployment.

```
kubectl get services
```

Take a screenshot of listing services with LoadBalancer indicating an external IP address that is ready for access.

```
amminer@cloudshell:~ (cloud-miner-amminer)$ kubectl get services  
NAME          TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE  
guestbook-lb  LoadBalancer  10.20.12.144  35.197.40.21  80:31133/TCP     17m  
kubernetes    ClusterIP      10.20.0.1    <none>        443/TCP          106m  
amminer@cloudshell:~ (cloud-miner-amminer)$
```

## 8. view the guestbook

- Bring the guestbook up at its external IP and sign it "Hello Kubernetes!".



## Guestbook

[Sign here](#)

### Entries

**Amelia <amminer@pdx.edu>**  
**signed on 2023-11-13 01:26:36.732318+00:00**  
**Hello Kubernetes!**

---

**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 2023-11-03 00:43:25.874230+00:00**

- View the resources kubernetes has deployed across your cloud project:
  - Take a screenshot of the managed guestbook pods and the service being exposed:

The screenshot shows the Google Cloud console interface for the 'cloud-Miner-amminer' project. The top navigation bar includes the Google Cloud logo, the project name, and various icons for search, alerts, and user profile. The main content area is titled 'Replication Co...' and features a 'Managed pods' section with a table of three running pods. Below this is an 'Exposing services' section with a table showing a 'Load balancer' service named 'guestbook-lb' with an external endpoint of '35.197.40.21:80'.

Name	Status	Restarts	Created on ↑
<a href="#">guestbook-replicas-vrws9</a>	Running	0	Nov 12, 2023, 5:18:38 PM
<a href="#">guestbook-replicas-pz2ps</a>	Running	0	Nov 12, 2023, 5:18:38 PM
<a href="#">guestbook-replicas-khdjf</a>	Running	0	Nov 12, 2023, 5:18:38 PM

Name ↑	Type	Endpoints
<a href="#">guestbook-lb</a>	Load balancer	<a href="#">35.197.40.21:80</a>

- Take a screenshot of the load balancer and its details:

The screenshot shows the 'Service details' page for the 'guestbook' service in the 'cloud-Miner-amminer' project. The page displays various metadata fields such as Cluster, Namespace, Labels, Logs, Type, and External endpoints. Below this is a 'Load Balancer' section with details about the Cluster IP, Load balancer IP, and the Load balancer ID. At the bottom is a 'Serving pods' table listing the three running pods with their status, endpoints, restarts, and creation times.

Cluster	<a href="#">guestbook</a>
Namespace	default
Labels	app: guestbook tier: frontend
Logs	<a href="#">Service logs</a>
Type	LoadBalancer
External endpoints	<a href="#">35.197.40.21:80</a>

Cluster IP	10.20.12.144
Load balancer IP	35.197.40.21
Load balancer	<a href="#">acdecb0e1dfb84e0c9137e3a7eaaec44</a>

Name	Status	Endpoints	Restarts	Created on ↑
<a href="#">guestbook-replicas-vrws9</a>	Running	10.16.0.9	0	Nov 12, 2023, 5:18:38 PM
<a href="#">guestbook-replicas-pz2ps</a>	Running	10.16.0.10	0	Nov 12, 2023, 5:18:38 PM
<a href="#">guestbook-replicas-khdjf</a>	Running	10.16.1.15	0	Nov 12, 2023, 5:18:38 PM



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

</

## 9. delete workload and service

```
kubectl delete -f kubernetes.yaml
gcloud container images delete
gcr.io/${GOOGLE_CLOUD_PROJECT}/gcp_gb
```



## 10. CI/CD build automation

- Run:

```
GOOGLE_CLOUD_PROJECT_NUMBER=$(gcloud projects describe  
$GOOGLE_CLOUD_PROJECT --format="value(projectNumber)")
```

```
echo $GOOGLE_CLOUD_PROJECT_NUMBER
```

```
gcloud projects add-iam-policy-binding ${GOOGLE_CLOUD_PROJECT}  
--member  
serviceAccount:${GOOGLE_CLOUD_PROJECT_NUMBER}@cloudbuild.gservice  
account.com --role=roles/container.developer
```



## 11. Configure build automation

- In cloud shell create cloudbuild.yaml from the code in the lab.

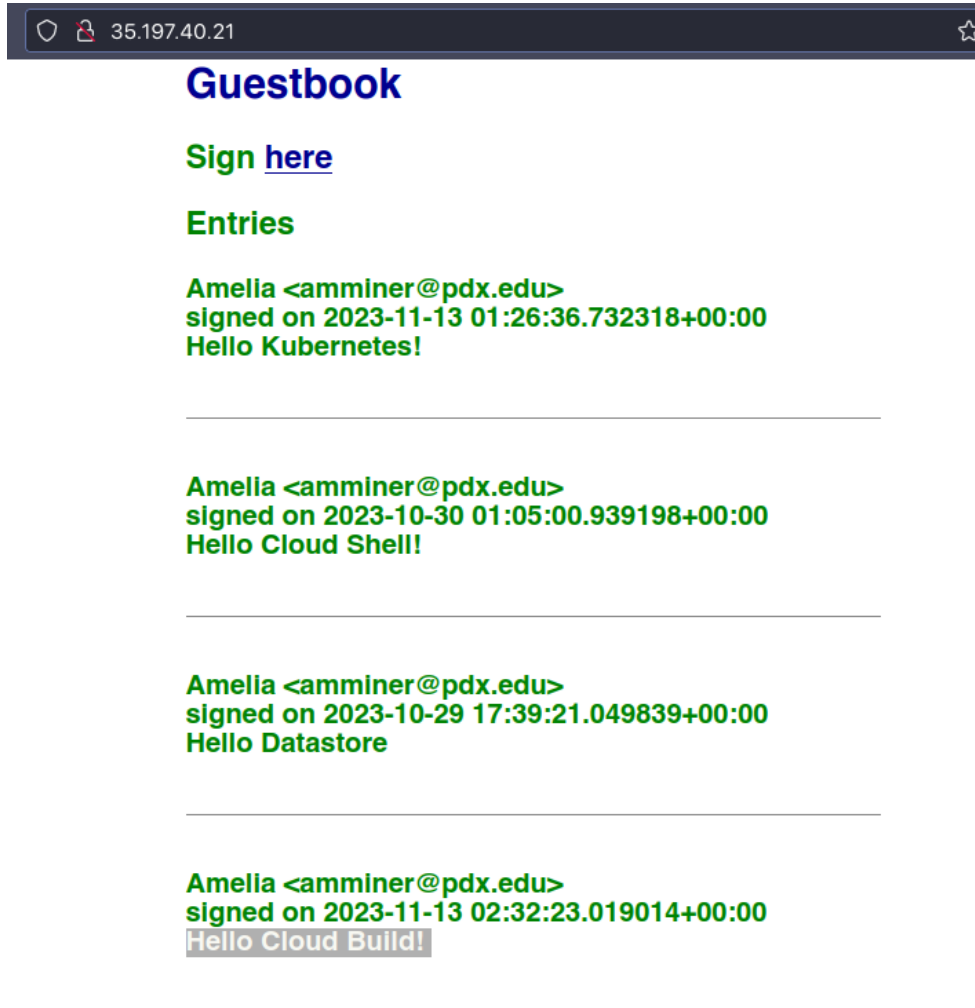


## 12. Deploy and view application

- Build and deploy the app:

```
gcloud builds submit --config=cloudbuild.yaml
```

Visit the load balancer and sign the guest book “Hello Cloud Build!”.



## 13. Clean up

- ```
kubectl delete -f kubernetes.yaml
```
- ```
gcloud container images delete  
gcr.io/${GOOGLE_CLOUD_PROJECT}/gcp_gb
```



## IV. Lab 7.3g: APIs (Slack, Knowledge Graph)

### 1. Slack and Knowledge Graph Integration

We're setting up a slack command /kg to query google's knowledge graph API on demand via a cloud function. The slack client sends a payload + token to the cloud function's trigger endpoint. The function verifies the token and sends a request to the knowledge graph API + an API key.

- `git clone`  
`https://github.com/GoogleCloudPlatform/python-docs-samples.git`
- `cd python-docs-samples/functions/slack/`



### 2. Code

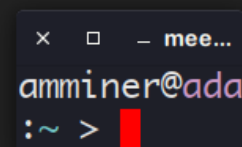
- Does Google provide a Python package specifically for accessing the Knowledge Graph API?

**No. Google provides the `googleapiclient.discovery` module as a generic interface to all of their backend APIs. It uses parameters to its `build` function to determine which API to instantiate.**

### 3. Code

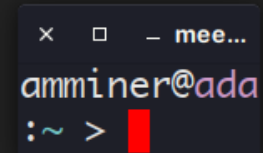
- Show the source line that constructs the query we wish to send to the Knowledge Graph API.

```
81
82     return message
83
84
85 # [END functions_slack_format]
86
87
88 # [START functions_slack_request]
89 def make_search_request(query):
90     req = kgsearch.entities().search(query=query, limit=1)
91     res = req.execute()
92     return format_slack_message(query, res)
93
94
95 # [END functions_slack_request]
96
97
98 # [START functions_slack_search]
99 @functions_framework.http
100 def kg_search(request):
```



- 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?  
**the `execute` method of the knowledge graph search query object, whose name is `req` in the program, sends the query to the API.**

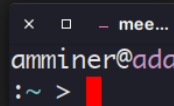
```
82     return message
83
84
85 # [END functions_slack_format]
86
87
88 # [START functions_slack_request]
89 def make_search_request(query):
90     req = kgsearch.entities().search(query=query, limit=1)
91     res = req.execute()
92     return format_slack_message(query, res)
93
94
95 # [END functions_slack_request]
96
97
98 # [START functions_slack_search]
99 @functions.framework.http
```



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

**A dictionary.**

```
42 # [START functions_slack_format]
43 def format_slack_message(query, response):
44     entity = None
45     if (
46         response
47         and response.get("itemListElement") is not None
48         and len(response["itemListElement"]) > 0
49     ):
50         entity = response["itemListElement"][0]["result"]
51
52     message = {
53         "response_type": "in_channel",
54         "text": f"Query: {query}",
55         "attachments": [],
56     }
57
58     attachment = {}
59     if entity:
60         name = entity.get("name", "")
61         description = entity.get("description", "")
62         detailed_desc = entity.get("detailedDescription", {})
63         url = detailed_desc.get("url")
64         article = detailed_desc.get("articleBody")
65         image_url = entity.get("image", {}).get("contentUrl")
66
67         attachment["color"] = "#3367d6"
68         if name and description:
69             attachment["title"] = "{}: {}".format(entity["name"], entity["description"])
70         elif name:
71             attachment["title"] = name
72         if url:
73             attachment["title_link"] = url
74         if article:
75             attachment["text"] = article
76         if image_url:
77             attachment["image_url"] = image_url
78     else:
79         attachment["text"] = "No results match your query."
80     message["attachments"].append(attachment)
81
82     return message
83
84
85 # [END functions_slack_format]
86
87
88 # [START functions_slack_request]
89 def make_search_request(query):
90     req = kgsearch.entities().search(query=query, limit=1)
91     res = req.execute()
92     return format_slack_message(query, res)
93
94
95 # [END functions_slack_request]
96
```



- What are the three main attributes of the formatted message passed back to Slack?  
**response\_type, text, and attachments... see above.**

## 4. Knowledge Graph Setup

- Enable the KG API and issue an API key.

```
gcloud services enable kgsearch.googleapis.com
gcloud alpha services api-keys create \
  --display-name="KG API Key" \
  --api-target=service=kgsearch.googleapis.com
```



## 5. Create a Slack Workspace

- Also create a slack app and associate it with your workspace. Obtain the slack app's signing secret: basic info -> app credentials -> show. Keep this page up.



## 6. Configure and Deploy

- in Cloud Shell, deploy the code substituting your kg key and slack secret. Note the URL of the function endpoint.



**[https://us-central1-cloud-miner-amminer.cloudfunctions.net/kg\\_search](https://us-central1-cloud-miner-amminer.cloudfunctions.net/kg_search)**

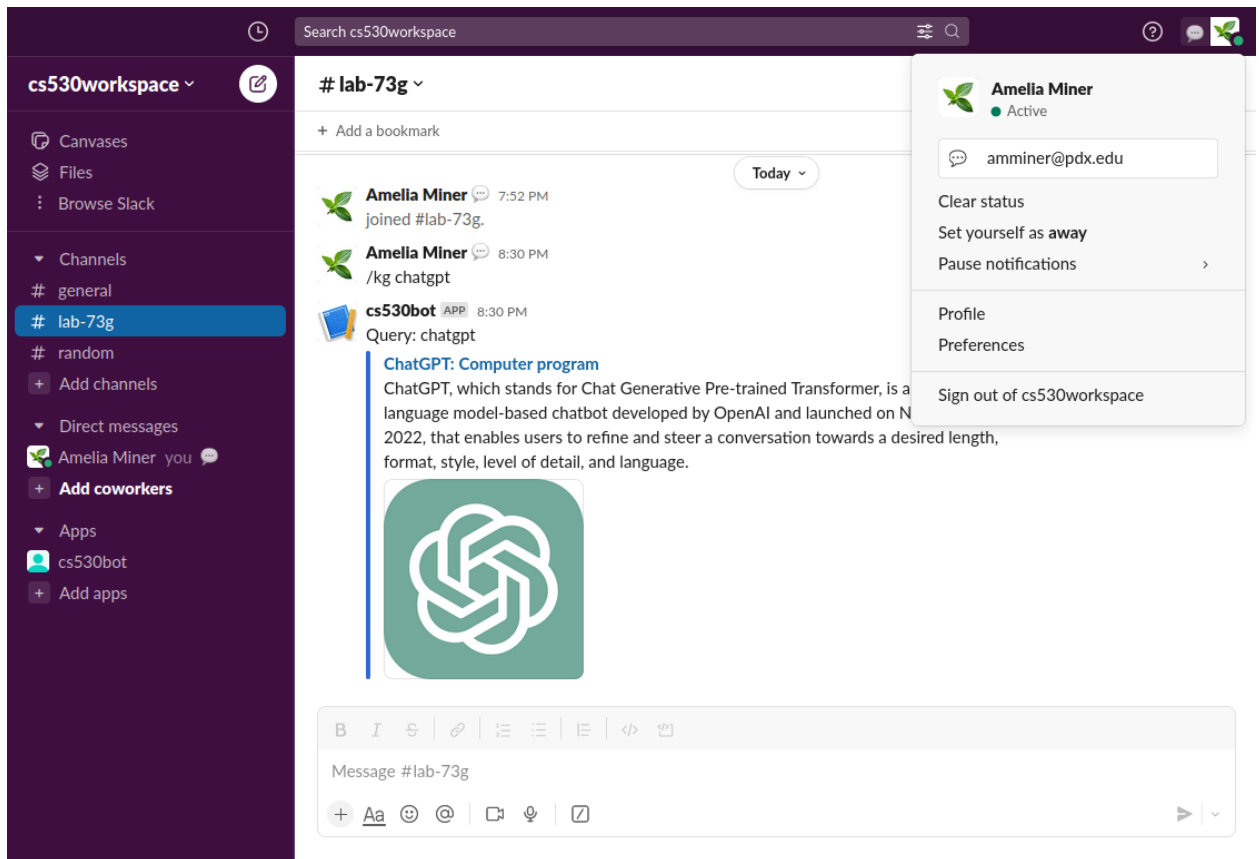
## 7. Create Slack Command

- Create the /kg command and connect it to the above endpoint. Install the app on your workspace.



## 8. Test the command

- Run the command `/kg chatgpt` in your workspace.



- Visit Cloud Shell and examine the logs for the function. Delete the function.



## V. Lab 7.4g: ML APIs

### 1. APIS #1 (Vision, Speech, Translate, Natural Language APIs)

- Enable the APIs in the cloud shell. Create a python venv. ✓

### 2. IAM Service Account Setup

- Create a service account, bind a role that allows ML services to the account, generate a key for the account, and save the key to an environment variable. ✓

### 3. Vision

- Install the cloud vision python package and cd into python-docs-samples/vision/snippets/detect. Run the following and show the output:

```
python detect.py labels-uri
```

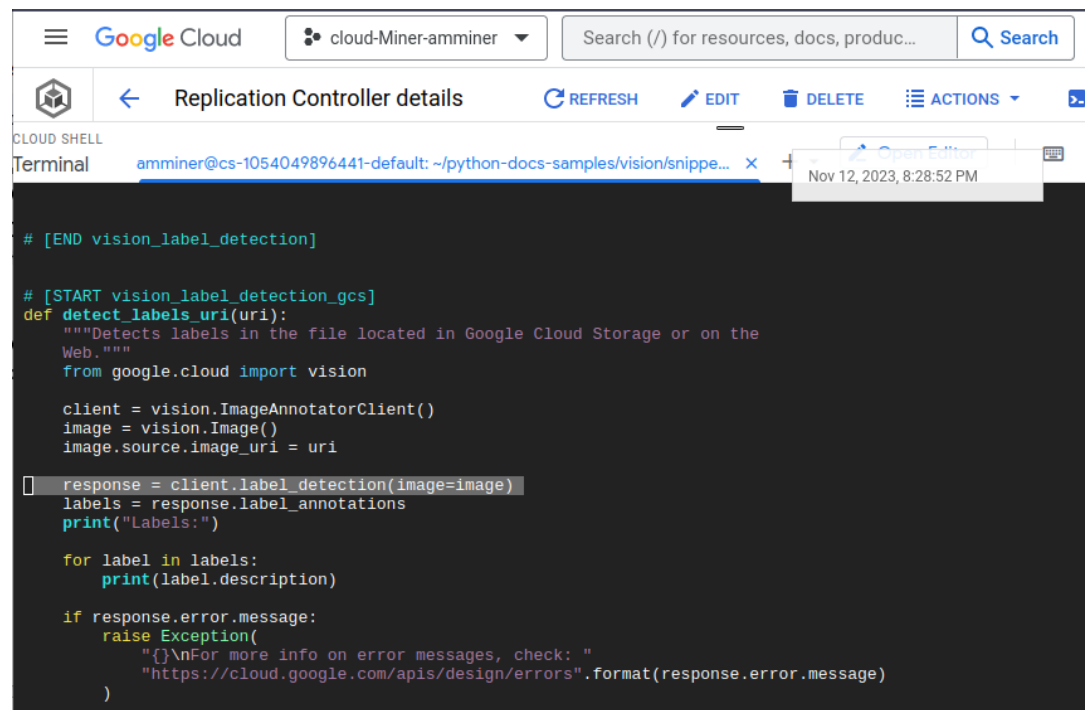
```
gs://cloud-samples-data/ml-api-codelab/birds.jpg
```

```
amminer@cs-1054049896441-default:~/python-docs-samples/vision/snippets/detect$ python detect.py labels-uri gs://cloud-samples-data/ml-api-cod
elab/birds.jpg
Labels:
Bird
Ratite
Cloud
Sky
Beak
Plant
Green
Neck
Ostrich
Casuariiformes
amminer@cs-1054049896441-default:~/python-docs-samples/vision/snippets/detect$
```

**My PS1 changed when I sourced the rc file, but rest assured I am in the venv.**

- Answer the following questions regarding the function call that handles the detection in the code:
  - What is the name of the function?

**detect\_labels\_uri**



```
# [END vision_label_detection]

# [START vision_label_detection_gcs]
def detect_labels_uri(uri):
    """Detects labels in the file located in Google Cloud Storage or on the
    Web."""
    from google.cloud import vision

    client = vision.ImageAnnotatorClient()
    image = vision.Image()
    image.source.image_uri = uri

    response = client.label_detection(image=image)
    labels = response.label_annotations
    print("Labels:")

    for label in labels:
        print(label.description)

    if response.error.message:
        raise Exception(
            "{}\nFor more info on error messages, check: "
            "https://cloud.google.com/apis/design/errors".format(response.error.message)
        )
```

- What type of Vision client is instantiated in it?  
**ImageAnnotatorClient... see above.**
- What method is invoked in the Vision client to perform the detection?  
**label\_detection(image)... see above**
- What is the name of the attribute in the response object that contains the results we seek?  
**label\_annotations... see above**



- using Google Images, download an image of a university logo to Cloud Shell via wget. Invoke detect.py to call the Vision API to determine whose logo it is. Show the output. What method is invoked *in the Vision client* to perform the detection?

**The function of the vision client that performs the detection is logo\_detection.**

**Interestingly, the detection was incorrect. The logo is not current - I think it was modernized in 2020 - but it is not uncommon as far as I'm aware.**

```
amminer@cs-1054049896441-default:~/python-docs-samples/vision/snippets/detect$ SLUG='https://keepingscore.blogs.time.com/wp-content/uploads/sites/6/2009/01/top10_mascots_banana_alt.jpg'
amminer@cs-1054049896441-default:~/python-docs-samples/vision/snippets/detect$ wget $SLUG -O slug.jpg
--2023-11-14 01:03:06-- https://keepingscore.blogs.time.com/wp-content/uploads/sites/6/2009/01/top10_mascots_banana_alt.jpg
Resolving keepingscore.blogs.time.com (keepingscore.blogs.time.com)... 192.0.66.85, 2a04:fa87:ffff::c000:4255
Connecting to keepingscore.blogs.time.com (keepingscore.blogs.time.com)|192.0.66.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33660 (33K) [image/jpeg]
Saving to: 'slug.jpg'

slug.jpg                               100%[=====>] 32.87K  --.-KB/s   in 0.03s

2023-11-14 01:03:06 (1.11 MB/s) - 'slug.jpg' saved [33660/33660]

amminer@cs-1054049896441-default:~/python-docs-samples/vision/snippets/detect$ fg
vim detect.py

[1]+  Stopped                  vim detect.py
amminer@cs-1054049896441-default:~/python-docs-samples/vision/snippets/detect$ python detect.py logos slug.jpg
Logos:
Tyler Junior College
amminer@cs-1054049896441-default:~/python-docs-samples/vision/snippets/detect$
```

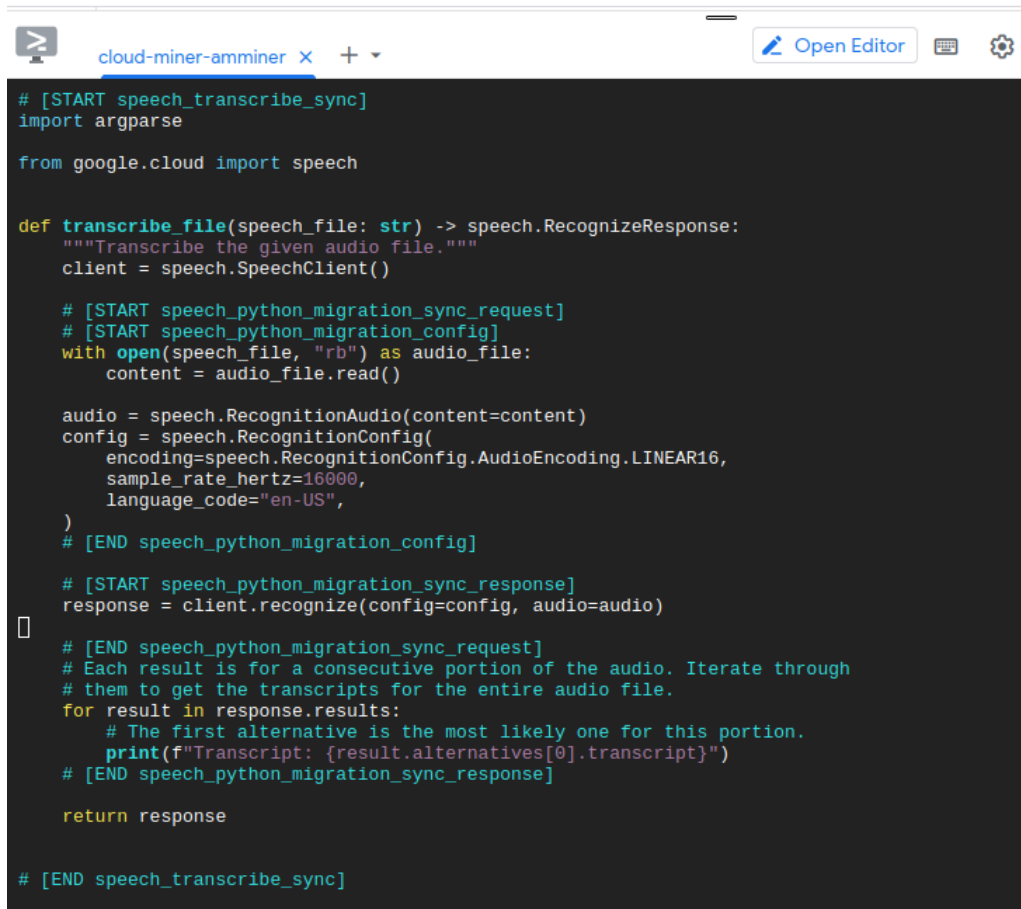
## 4. Speech

- Install the speech package. cd into python-docs-samples/speech/snippets and run python transcribe.py resources/audio.raw. Show the output for your lab notebook.

```
(env) amminer@cloudshell:~/python-docs-samples/speech/snippets (cloud-miner-amminer)$ python transcribe.py resources/audio.raw
Transcript: how old is the Brooklyn Bridge
(env) amminer@cloudshell:~/python-docs-samples/speech/snippets (cloud-miner-amminer)$
```

- What is the name of the function that handles the translation?

**transcribe\_file**

A screenshot of a code editor window titled 'cloud-miner-amminer'. The editor shows a Python function named 'transcribe\_file' which takes a 'speech\_file' as an argument and returns a 'speech.RecognizeResponse' object. The function includes comments for migration sync points and uses the Google Cloud Speech API to transcribe the audio file. The output of the transcription is printed to the console.

```
# [START speech_transcribe_sync]
import argparse

from google.cloud import speech

def transcribe_file(speech_file: str) -> speech.RecognizeResponse:
    """Transcribe the given audio file."""
    client = speech.SpeechClient()

    # [START speech_python_migration_sync_request]
    # [START speech_python_migration_config]
    with open(speech_file, "rb") as audio_file:
        content = audio_file.read()

    audio = speech.RecognitionAudio(content=content)
    config = speech.RecognitionConfig(
        encoding=speech.RecognitionConfig.AudioEncoding.LINEAR16,
        sample_rate_hertz=16000,
        language_code="en-US",
    )
    # [END speech_python_migration_config]

    # [START speech_python_migration_sync_response]
    response = client.recognize(config=config, audio=audio)
    # [END speech_python_migration_sync_response]

    # Each result is for a consecutive portion of the audio. Iterate through
    # them to get the transcripts for the entire audio file.
    for result in response.results:
        # The first alternative is the most likely one for this portion.
        print(f"Transcript: {result.alternatives[0].transcript}")
    # [END speech_python_migration_sync_response]

    return response

# [END speech_transcribe_sync]
```

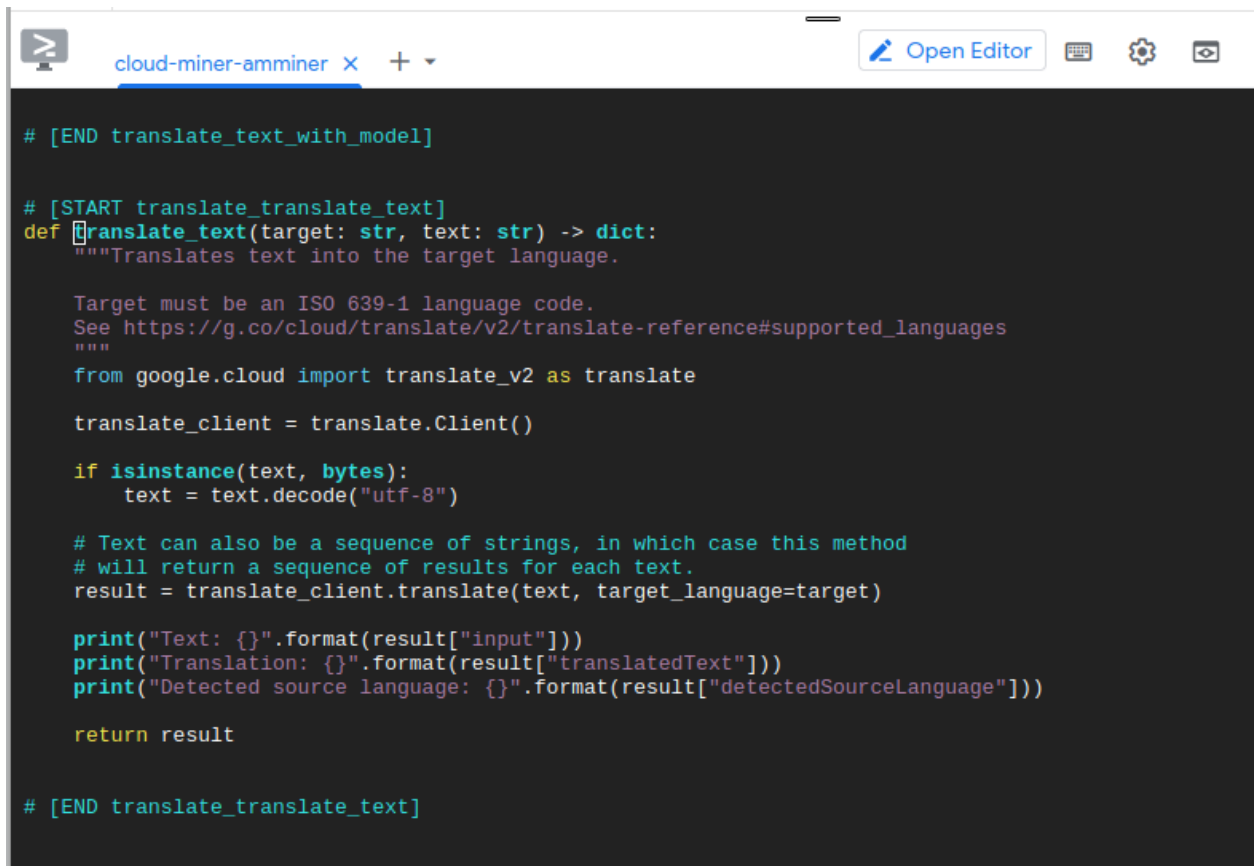
- What method is invoked in the Speech client to perform the detection?  
**recognize, see above**
- What is the name of the attribute in the response object that contains the results we seek?  
**results, see above**

## 5. Translate

- ... do the same as the above with the translate API. Run this: `python snippets.py translate-text en '你有沒有帶外套'`

```
(env) amminer@cloudshell:~/python-docs-samples/translate/samples/snippets (cloud-miner-amminer)$ python snippets.py translate-text en '你有沒有帶外套'  
Text: 你有沒有帶外套  
Translation: did you bring a coat  
Detected source language: zh-TW  
(env) amminer@cloudshell:~/python-docs-samples/translate/samples/snippets (cloud-miner-amminer)$
```

- What is the name of the function that handles this translation?  
**translate\_text.** It just occurred to me that you might want me to show the part of the script that directs us to this function under `if __name__ == "__main__":`. Please have mercy on me - trust me, I understand how I got to the function I answered with. My proficiency with python is one of the most important links in the chain of skills that puts a roof over my head and food on my table.



```
# [END translate_text_with_model]  
  
# [START translate_translate_text]  
def translate_text(target: str, text: str) -> dict:  
    """Translates text into the target language.  
  
    Target must be an ISO 639-1 language code.  
    See https://g.co/cloud/translate/v2/translate-reference#supported_languages  
    """  
    from google.cloud import translate_v2 as translate  
    translate_client = translate.Client()  
  
    if isinstance(text, bytes):  
        text = text.decode("utf-8")  
  
    # Text can also be a sequence of strings, in which case this method  
    # will return a sequence of results for each text.  
    result = translate_client.translate(text, target_language=target)  
  
    print("Text: {}".format(result["input"]))  
    print("Translation: {}".format(result["translatedText"]))  
    print("Detected source language: {}".format(result["detectedSourceLanguage"]))  
  
    return result  
  
# [END translate_translate_text]
```

- What method is invoked in the Translate client to perform the detection?  
**translate**
- What is the name of the attribute in the response object that contains the results we seek?  
There is no such attribute of the response object, strictly speaking. The response object - the return value of `translate_client.translate` - is a dictionary with key-value pairs. The keys whose values contain the results we seek are `input`, `translatedText`, and `detectedSourceLanguage`. These values are not accessible via

**an attribute of the object, only through square-bracket notation (which is just syntactic sugar for a function call if I recall correctly) or through calls to a couple of dictionary class functions.**

## 6. Natural Language

- Install the language package. Create language.py from the lab material. Run the following:

```
python language.py 'homework is awful!'
python language.py 'homework is ok'
python language.py 'homework is awesome?'
python language.py 'homework is awesome!'
python language.py 'The protestors in Oregon put on gas masks and
wore yellow t-shirts'
```

```
(env) amminer@cloudshell:~ (cloud-miner-amminer)$ python language.py 'homework is awful!'
python language.py 'homework is ok'
python language.py 'homework is awesome?'
python language.py 'homework is awesome!'
python language.py 'The protestors in Oregon put on gas masks and wore yellow t-shirts'
"homework is awful!" has sentiment=-0.800000011920929

Entities are:
name: homework
"homework is ok" has sentiment=0.30000001192092896

Entities are:
name: homework
"homework is awesome?" has sentiment=0.4000000059604645

Entities are:
name: homework
"homework is awesome!" has sentiment=0.8999999761581421

Entities are:
name: homework
"The protestors in Oregon put on gas masks and wore yellow t-shirts" has sentiment=-0.6000000238418579

Entities are:
name: protestors
name: gas masks
name: Oregon
name: t-shirts
(env) amminer@cloudshell:~ (cloud-miner-amminer)$
```

It might be good to mention the meaning of that floating point value here in the lab material...

## 7. Integration

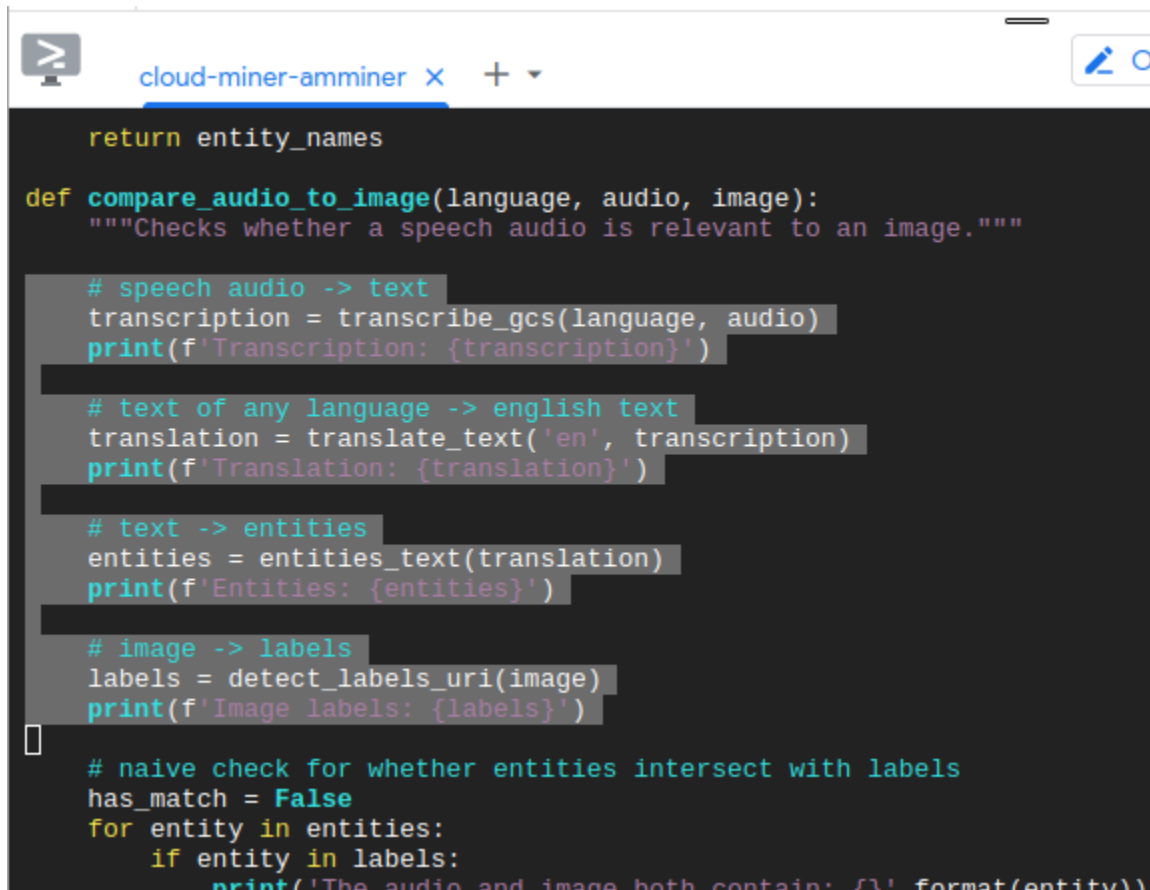


## 8. Code

- Copy the code from the lab. Examine it. What is the name of the function that performs the transcription?

**transcribe\_gcs**

- What is the name of the function that performs the translation?  
**translate\_text**
- What is the name of the function that performs the entity analysis on the translation?  
**entities\_text**
- What is the name of the function that performs the entity analysis on the image?  
**detect\_labels\_uri**



```
return entity_names

def compare_audio_to_image(language, audio, image):
    """Checks whether a speech audio is relevant to an image."""

    # speech audio -> text
    transcription = transcribe_gcs(language, audio)
    print(f'Transcription: {transcription}')

    # text of any language -> english text
    translation = translate_text('en', transcription)
    print(f'Translation: {translation}')

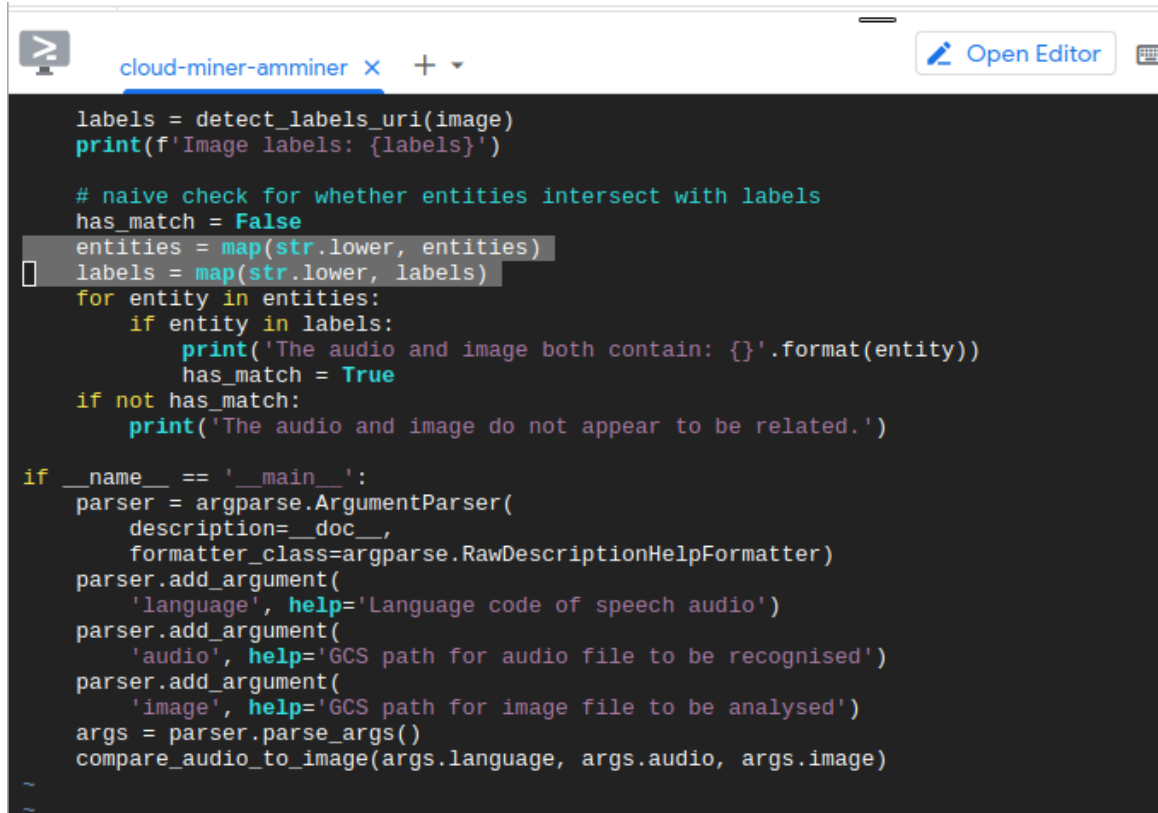
    # text -> entities
    entities = entities_text(translation)
    print(f'Entities: {entities}')

    # image -> labels
    labels = detect_labels_uri(image)
    print(f'Image labels: {labels}')

    # naive check for whether entities intersect with labels
    has_match = False
    for entity in entities:
        if entity in labels:
            print('The audio and image both contain: {}'.format(entity))
```

## 9. Test Integration

- run `python solution.py de-DE`  
`gs://cloud-samples-data/ml-api-codelab/de-ball.wav`  
`gs://cloud-samples-data/ml-api-codelab/football.jpg`. If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?  
**Heads up that we're missing a pip install step here, or at least I had to perform one for the six package in my gcloud instance.**  
**The program does deem them unrelated. It's a case sensitivity issue. The program should compare the results in a case-insensitive way:**



```
cloud-miner-amminer x + v Open Editor

labels = detect_labels_uri(image)
print(f'Image labels: {labels}')

# naive check for whether entities intersect with labels
has_match = False
entities = map(str.lower, entities)
labels = map(str.lower, labels)
for entity in entities:
    if entity in labels:
        print('The audio and image both contain: {}'.format(entity))
        has_match = True
if not has_match:
    print('The audio and image do not appear to be related.')

if __name__ == '__main__':
    parser = argparse.ArgumentParser(
        description=__doc__,
        formatter_class=argparse.RawDescriptionHelpFormatter)
    parser.add_argument(
        'language', help='Language code of speech audio')
    parser.add_argument(
        'audio', help='GCS path for audio file to be recognised')
    parser.add_argument(
        'image', help='GCS path for image file to be analysed')
    args = parser.parse_args()
    compare_audio_to_image(args.language, args.audio, args.image)
```

### Before & after this change:

```
(env) amminer@cloudshell:~ (cloud-miner-amminer)$ python solution.py de-DE gs://cloud-samples-data/ml-api-codelab/de-ball.wav
gs://cloud-samples-data/ml-api-codelab/football.jpg
Transcription: willst du mit uns Fußball spielen
Translation: Do you want to play football with us?
Entities: ['football']
Image labels: ['Sports equipment', 'Soccer', 'Football', 'Plant', 'Ball', 'Player', 'Playing sports', 'Soccer ball', 'Ball game', 'Team sport']
The audio and image do not appear to be related.
(env) amminer@cloudshell:~ (cloud-miner-amminer)$ fg
vim solution.py

[1]+  Stopped                  vim solution.py
(env) amminer@cloudshell:~ (cloud-miner-amminer)$ python solution.py de-DE gs://cloud-samples-data/ml-api-codelab/de-ball.wav
gs://cloud-samples-data/ml-api-codelab/football.jpg
Transcription: willst du mit uns Fußball spielen
Translation: do you want to play football with us?
Entities: ['football']
Image labels: ['Sports equipment', 'Soccer', 'Football', 'Plant', 'Ball', 'Player', 'Playing sports', 'Soccer ball', 'Ball game', 'Team sport']
The audio and image both contain: football
```

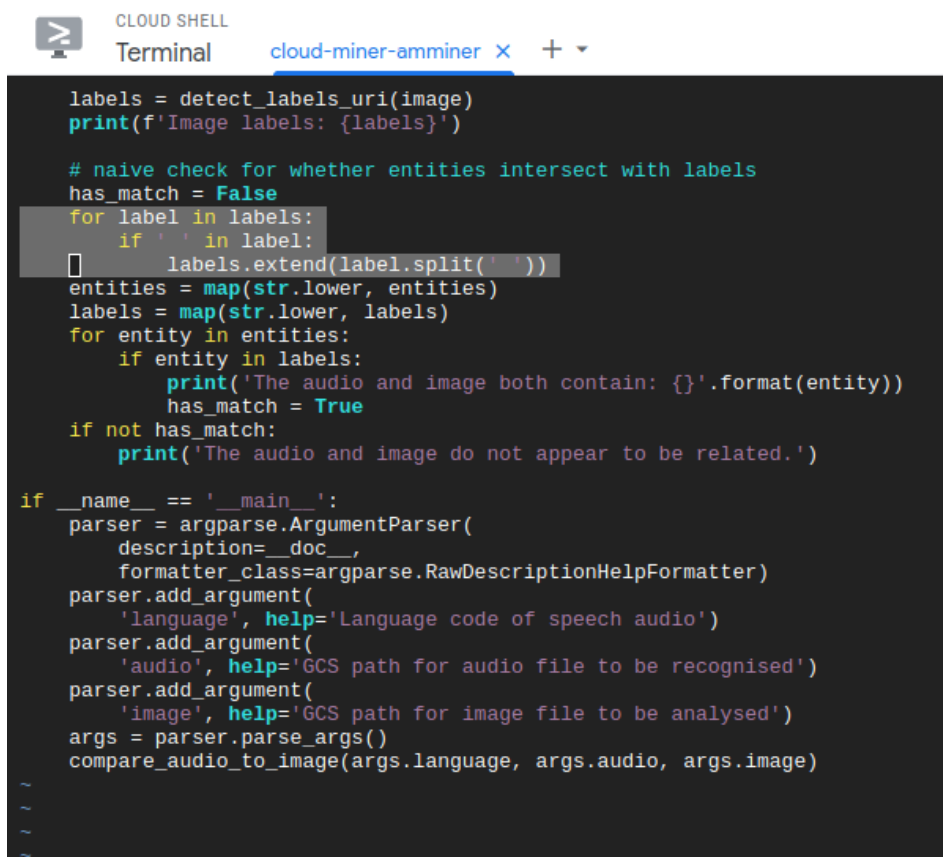
- Run `python solution.py tr-TR`

`gs://cloud-samples-data/ml-api-codelab/tr-bike.wav`

`gs://cloud-samples-data/ml-api-codelab/bicycle.jpg`. If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?

The program doesn't recognize that, in this situation, the words "bike" and "bicycle" likely refer to the same object or concept. Meaning is potentially blurred at the translation step but I don't know Turkish so I can't say. We have options:

1. Generate multiple translations and modify the code to process all of them - this is relatively complicated so I'm avoiding it.
2. Take each token the APIs return and gather a list of synonyms for it, then modify our comparison loop to iterate through these synonyms - this is relatively complicated so I'm going to try to avoid it.
3. Generate a lot more tokens per image and hope one of them matches - this is easy, the `label_detection` function takes a `max_results` parameter that controls how many tokens we get back. However I was not able to get it to spit out "bike" based on that image even with the maximum cranked past the number it will return, which is 66 for this image.
4. Do option 3, and also modify the comparison code to search for each word within multi-word labels in addition to the whole labels themselves. This is relatively easy as well and solved the problem for me.



```
CLOUD SHELL
Terminal cloud-miner-amminer x + v

labels = detect_labels_uri(image)
print(f'Image labels: {labels}')

# naive check for whether entities intersect with labels
has_match = False
for label in labels:
    if ' ' in label:
        labels.extend(label.split(' '))
entities = map(str.lower, entities)
labels = map(str.lower, labels)
for entity in entities:
    if entity in labels:
        print('The audio and image both contain: {}'.format(entity))
        has_match = True
if not has_match:
    print('The audio and image do not appear to be related.')

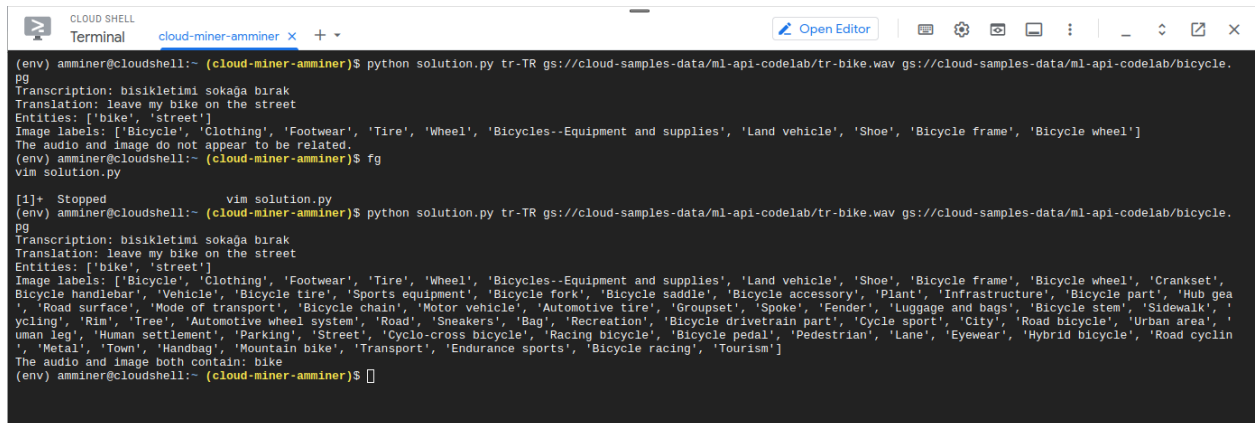
if __name__ == '__main__':
    parser = argparse.ArgumentParser(
        description=__doc__,
        formatter_class=argparse.RawDescriptionHelpFormatter)
    parser.add_argument(
        'language', help='Language code of speech audio')
    parser.add_argument(
        'audio', help='GCS path for audio file to be recognised')
    parser.add_argument(
        'image', help='GCS path for image file to be analysed')
    args = parser.parse_args()
    compare_audio_to_image(args.language, args.audio, args.image)
```

Note that we only split on spaces here,  
while a more thorough implementation might consider other separator characters.

```
def detect_labels_uri(uri):  
    """Detects labels in the file located in Google Cloud Storage or on the  
    Web."""  
  
    # create ImageAnnotatorClient object  
    client = vision.ImageAnnotatorClient()  
  
    # create Image object  
    image = vision.Image()  
  
    # specify location of image  
    image.source.image_uri = uri  
  
    # get label_detection response by passing image to client  
    response = client.label_detection(image=image, max_results=1000)  
  
    # get label_annotations portion of response  
    labels = response.label_annotations  
  
    # we only need the label descriptions  
    label_descriptions = []  
    for label in labels:  
        label_descriptions.append(label.description)  
  
    return label_descriptions
```

Amelia Miner

Before and after:



```
(env) amminer@cloudshell:~ (cloud-miner-amminer) $ python solution.py tr-TR gs://cloud-samples-data/ml-api-codelab/tr-bike.wav gs://cloud-samples-data/ml-api-codelab/bicycle.  
pg  
Transcription: bisikletimi sokağa bırak  
Translation: leave my bike on the street  
Entities: ['bike', 'street']  
Image labels: ['Bicycle', 'Clothing', 'Footwear', 'Tire', 'Wheel', 'Bicycles--Equipment and supplies', 'Land vehicle', 'Shoe', 'Bicycle frame', 'Bicycle wheel']  
The audio and image do not appear to be related.  
(env) amminer@cloudshell:~ (cloud-miner-amminer) $ fg  
vim solution.py  
[1]+ Stopped vim solution.py  
(env) amminer@cloudshell:~ (cloud-miner-amminer) $ python solution.py tr-TR gs://cloud-samples-data/ml-api-codelab/tr-bike.wav gs://cloud-samples-data/ml-api-codelab/bicycle.  
pg  
Transcription: bisikletimi sokağa bırak  
Translation: leave my bike on the street  
Entities: ['bike', 'street']  
Image labels: ['Bicycle', 'Clothing', 'Footwear', 'Tire', 'Wheel', 'Bicycles--Equipment and supplies', 'Land vehicle', 'Shoe', 'Bicycle frame', 'Bicycle wheel', 'Crankset',  
Bicycle handlebar', 'Vehicle', 'Bicycle tire', 'Sports equipment', 'Bicycle fork', 'Bicycle saddle', 'Bicycle accessory', 'Plant', 'Infrastructure', 'Bicycle part', 'Hub gear',  
, 'Road surface', 'Mode of transport', 'Bicycle chain', 'Motor vehicle', 'Automotive tire', 'Groupset', 'Spoke', 'Fender', 'Luggage and bags', 'Bicycle stem', 'Sidewalk', 'Cycling',  
'Rim', 'Tree', 'Automotive wheel system', 'Road', 'Sneakers', 'Bag', 'Recreation', 'Bicycle drivetrain part', 'Cycle sport', 'City', 'Road bicycle', 'Urban area', 'Human leg', 'Human settlement', 'Parking', 'Street', 'Cyclo-cross bicycle', 'Racing bicycle', 'Bicycle pedal', 'Pedestrian', 'Lane', 'Eyewear', 'Hybrid bicycle', 'Road cycling', 'Metal', 'Town', 'Handbag', 'Mountain bike', 'Transport', 'Endurance sports', 'Bicycle racing', 'Tourism']  
The audio and image both contain: bike  
(env) amminer@cloudshell:~ (cloud-miner-amminer) $
```



- Run the following:

```
python solution.py tr-TR
```

```
gs://cloud-samples-data/ml-api-codelab/tr-ostrich.wav
```

```
gs://cloud-samples-data/ml-api-codelab/birds.jpg
```

If the program deems them unrelated, then based on the results from the APIs, what must be changed in the program to address this?

**The most straightfoward solution I can think of is to use a library like inflect to include both plural and singular forms for each entity the google API returns. I would like to demonstrate this if I had time, as I'm confident that it would work, but I just don't have time, and the lab strictly only asks what must be changed, not to demonstrate the solution.**

## 10. APIS #2 (Video Intelligence API)

- gcloud services enable videointelligence.googleapis.com



## 11. Video Setup

- install the google-cloud-videointelligence package. Create a storage bucket for a video and save its name to an environment variable. set the bucket and its contents to be publicly readable. Download the video from the lab and put it in the bucket.



## 12. Video Intelligence Labeling Script

- Copy the script into cloud shell.



## 13. Video Intelligence

- Run `python labels.py`

```
gs://{CLOUD_STORAGE_BUCKET}/SportsBloopers2016.mp4.
```

What are the 3 labels with the highest confidence that the Video Intelligence API associates with the video and what are the confidences for each?

**sports, basketball, player**

- Open `labels.py`. What is the name of the client class in the package that is used?  
**VideoIntelligenceServiceClient**
- What method is used in that class to perform the annotation?  
**annotate\_video**

## 14. APIS #3 (Web Site Integration)

- Check out the specified revision and change into the specified directory.



## 15. IAM Service Account Setup

- Set up a new service account. Bind the storage.admin, datastore.user, and serviceusage.serviceUsageConsumer roles to it. Issue a key for the service account, download it, and set an environment variable to its path (GOOGLE\_APPLICATION\_CREDENTIALS).



## 16. Application

```
(env) amminer@cloudshell:~/python-docs-samples/codelabs/face_and_vision (cloud-miner-amminer)$ python main.py
Traceback (most recent call last):
  File "/home/amminer/python-docs-samples/codelabs/face_and_vision/main.py", line 19, in <module>
    from flask import Flask, redirect, render_template, request
  File "/home/amminer/env/lib/python3.9/site-packages/flask/__init__.py", line 7, in <module>
    from .app import Flask as Flask
  File "/home/amminer/env/lib/python3.9/site-packages/flask/app.py", line 28, in <module>
    from . import cli
  File "/home/amminer/env/lib/python3.9/site-packages/flask/cli.py", line 18, in <module>
    from .helpers import get_debug_flag
  File "/home/amminer/env/lib/python3.9/site-packages/flask/helpers.py", line 16, in <module>
    from werkzeug.urls import url_quote
  File "/home/amminer/env/lib/python3.9/site-packages/werkzeug/urls.py", line 16, in <module>
    from werkzeug.urls import url_quote
ImportError: cannot import name 'url_quote' from 'werkzeug.urls' (/home/amminer/env/lib/python3.9/site-packages/werkzeug/urls.py)
```

Looks like there is a dependency issue :( Indeed:

<https://stackoverflow.com/questions/77213053/importerror-cannot-import-name-url-quote-from-werkzeug-urls>

Specifying Werkzeug==2.2.2 works.

https://8080-cs-1054049896441-default.cs-us-west1-wolo.cloudshell.dev

## Google Cloud Platform - Face Detection Sample

This Python Flask application demonstrates App Engine Flexible, Google Cloud Storage, Datastore, and the Cloud Vision API.

Upload File:  No file selected.



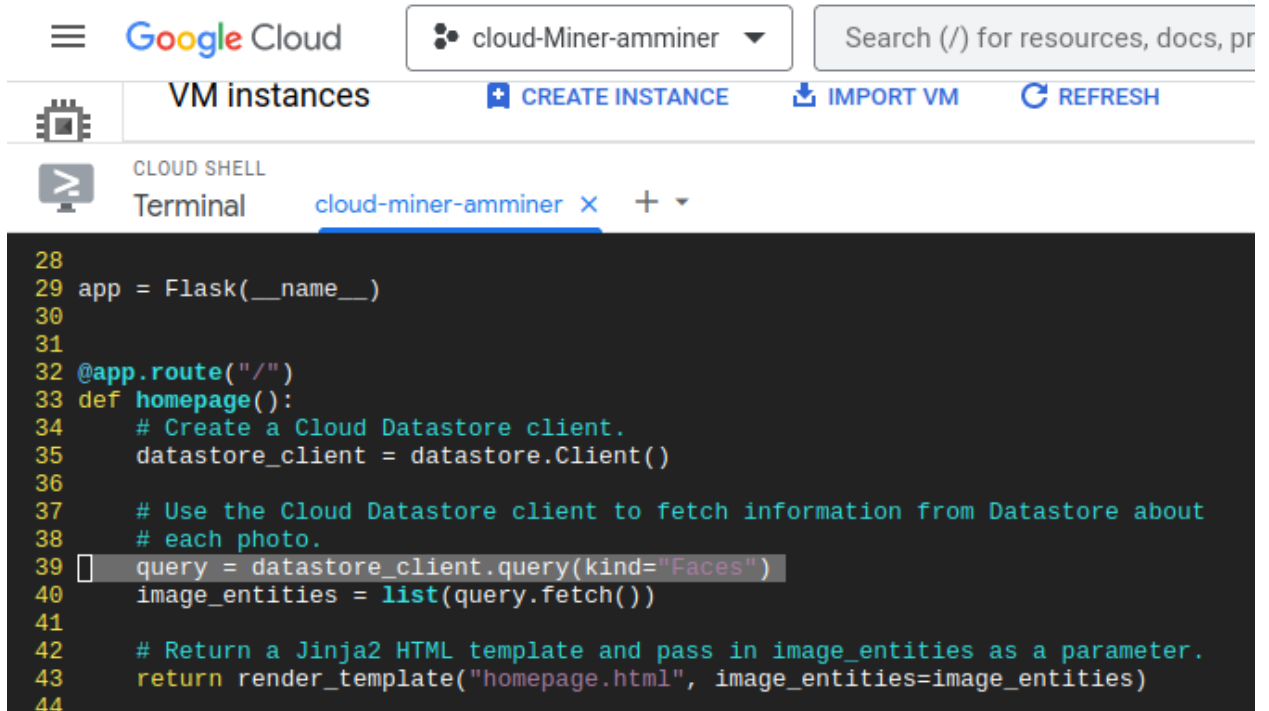
cat-thousand-yard-stare-thousand-yard-stare.gif was uploaded 2023-11-14 03:33:36.634175+00:00.

Joy Likelihood for Face: Very Unlikely

## 17. Code

- Open main.py and view the code for the default route. What line of code creates the query for previous detections?

39



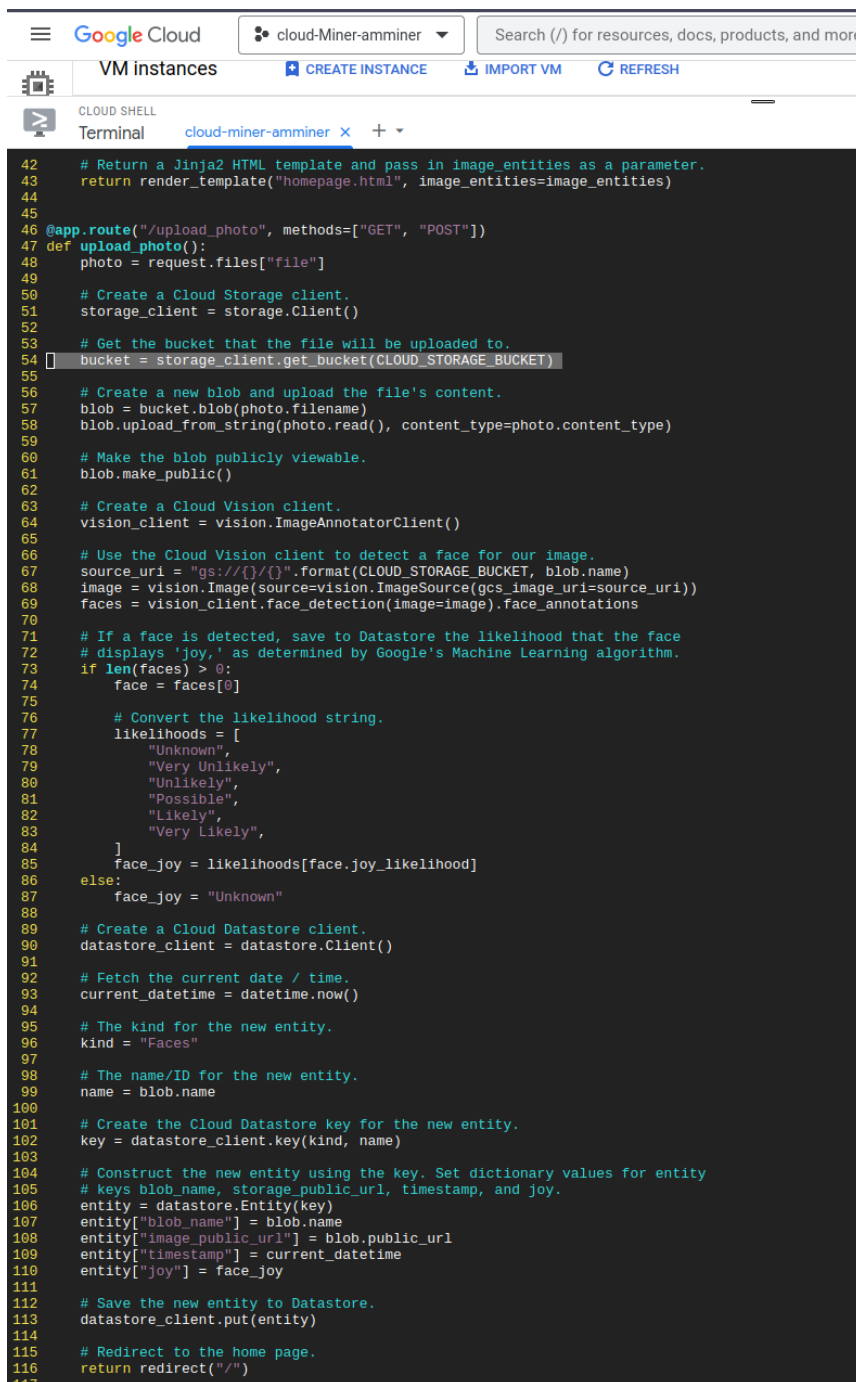
```
28
29 app = Flask(__name__)
30
31
32 @app.route("/")
33 def homepage():
34     # Create a Cloud Datastore client.
35     datastore_client = datastore.Client()
36
37     # Use the Cloud Datastore client to fetch information from Datastore about
38     # each photo.
39     query = datastore_client.query(kind="Faces")
40     image_entities = list(query.fetch())
41
42     # Return a Jinja2 HTML template and pass in image_entities as a parameter.
43     return render_template("homepage.html", image_entities=image_entities)
44
```

- What line of code sends the query to Cloud Datastore?

40 (see above)

- View the upload\_photo route. Show the line that retrieves the name of the storage bucket to use.

54



```
42 # Return a Jinja2 HTML template and pass in image_entities as a parameter.
43 return render_template("homepage.html", image_entities=image_entities)
44
45
46 @app.route("/upload_photo", methods=["GET", "POST"])
47 def upload_photo():
48     photo = request.files["file"]
49
50     # Create a Cloud Storage client.
51     storage_client = storage.Client()
52
53     # Get the bucket that the file will be uploaded to.
54     bucket = storage_client.get_bucket(CLOUD_STORAGE_BUCKET)
55
56     # Create a new blob and upload the file's content.
57     blob = bucket.blob(photo.filename)
58     blob.upload_from_string(photo.read(), content_type=photo.content_type)
59
60     # Make the blob publicly viewable.
61     blob.make_public()
62
63     # Create a Cloud Vision client.
64     vision_client = vision.ImageAnnotatorClient()
65
66     # Use the Cloud Vision client to detect a face for our image.
67     source_uri = "gs://{}/{}/{}".format(CLOUD_STORAGE_BUCKET, blob.name)
68     image = vision.Image(source=vision.ImageSource(gcs_image_uri=source_uri))
69     faces = vision_client.face_detection(image=image).face_annotations
70
71     # If a face is detected, save to Datastore the likelihood that the face
72     # displays 'joy,' as determined by Google's Machine Learning algorithm.
73     if len(faces) > 0:
74         face = faces[0]
75
76         # Convert the likelihood string.
77         likelihoods = [
78             "Unknown",
79             "Very Unlikely",
80             "Unlikely",
81             "Possible",
82             "Likely",
83             "Very Likely",
84         ]
85         face_joy = likelihoods[face.joy_likelihood]
86     else:
87         face_joy = "Unknown"
88
89     # Create a Cloud Datastore client.
90     datastore_client = datastore.Client()
91
92     # Fetch the current date / time.
93     current_datetime = datetime.now()
94
95     # The kind for the new entity.
96     kind = "Faces"
97
98     # The name/ID for the new entity.
99     name = blob.name
100
101     # Create the Cloud Datastore key for the new entity.
102     key = datastore_client.key(kind, name)
103
104     # Construct the new entity using the key. Set dictionary values for entity
105     # keys blob_name, storage_public_url, timestamp, and joy.
106     entity = datastore.Entity(key)
107     entity["blob_name"] = blob.name
108     entity["image_public_url"] = blob.public_url
109     entity["timestamp"] = current_datetime
110     entity["joy"] = face_joy
111
112     # Save the new entity to Datastore.
113     datastore_client.put(entity)
114
115     # Redirect to the home page.
116     return redirect("/")
117
```

- What form field is used to specify the uploaded photo?  
**file (line 48 above).**
- Show the line that copies the photo's contents to the storage bucket.  
**Line 58 above.**

- What method in Vision's annotation client is used to perform the analysis?  
**face\_detection (line 69 above).**
- What fields are stored in Cloud Datastore for each image?  
**blob\_name, image\_public\_url, timestamp, and joy (lines 107 through 110)**
- What happens at the end of the upload\_photo route?  
**The application redirects the browser to the home page (line 116)**

## 18. Clean Up

- Delete the service accounts, keys, and bucket.

