

7.1g

Visit the Compute Engine web console and find the VM that you have deployed.

Take a screenshot that includes the VM's IP addresses

The screenshot shows the Google Cloud Compute Engine console. The 'VM instances' tab is selected. A message at the top states: 'Your project's VMs use global DNS names by default. To reduce the risk of cross-regional outages, we recommend you use zonal DNS instead. [Learn more](#)'. Below this, a table lists the VM instances:

| Status | Name | Zone | Internal IP | External IP | Network | Connect |
|--------|-----------|------------|--------------------|-------------|---------|---------|
| | course-vm | us-west1-b | 10.138.0.2 (nic0) | | default | SSH |
| | tf-lab-vm | us-west1-b | 10.138.0.16 (nic0) | | default | SSH |

Run terraform plan again. Ensure there is one resource being added (the IP address), one resource being changed (the VM), and a new output (ip). After ensuring this is the case, apply the configuration.

Take a screenshot showing the completion of the command including its output

```
jtn7@cloudshell:~/tf (cloud-nguyen-jtn7)$ terraform apply
google_compute_instance.default: Refreshing state... [id=projects/cloud-nguyen-jtn7/zones/us-west1-b/instances/tf-lab-vm]

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

Terraform will perform the following actions:

# google_compute_address.static will be created
+ resource "google_compute_address" "static" {
+   address           = (known after apply)
+   address_type      = "EXTERNAL"
+   creation_timestamp = (known after apply)
+   effective_labels   = (known after apply)
+   id                = (known after apply)
+   label_fingerprint = (known after apply)
+   name              = "ipv4-address"
+   network_tier       = (known after apply)
+   prefix_length      = (known after apply)
+   project            = "cloud-nguyen-jtn7"
+   purpose            = (known after apply)
+   region            = (known after apply)
+   self_link          = (known after apply)
+   subnetwork         = (known after apply)
+   terraform_labels   = (known after apply)
+   users              = (known after apply)
+ }

# google_compute_instance.default will be updated in-place
+ resource "google_compute_instance" "default" {
+   id     = "projects/cloud-nguyen-jtn7/zones/us-west1-b/instances/tf-lab-vm"
+   name   = "tf-lab-vm"
+   tags   = []
+ }

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

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

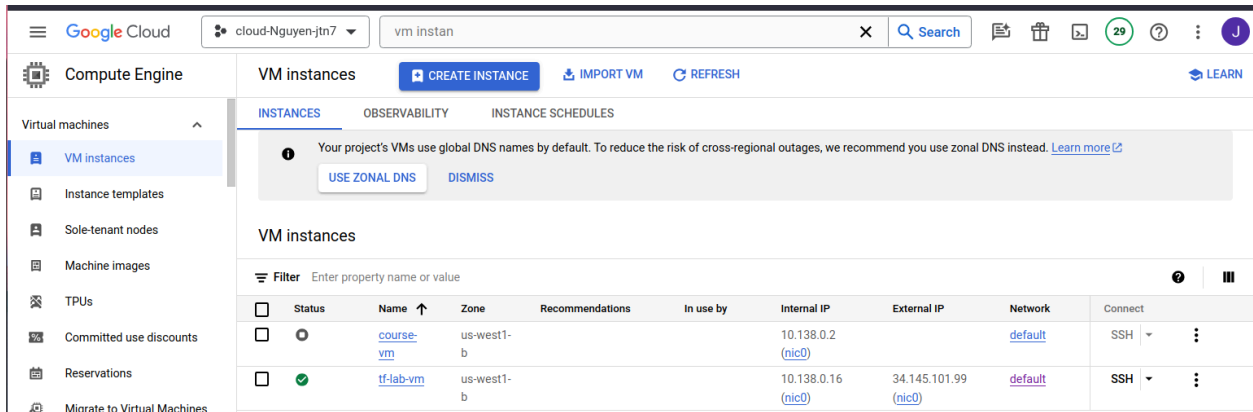
google_compute_address.static: Creating...
google_compute_address.static: Still creating... [10s elapsed]
google_compute_address.static: Creation complete after 11s [id=projects/cloud-nguyen-jtn7/zones/us-west1-b/instances/tf-lab-vm]
google_compute_instance.default: Modifying... [id=projects/cloud-nguyen-jtn7/zones/us-west1-b/instances/tf-lab-vm]
google_compute_instance.default: Still modifying... [id=projects/cloud-nguyen-jtn7/zones/us-west1-b/instances/tf-lab-vm, 10s elapsed]
google_compute_instance.default: Modifications complete after 11s [id=projects/cloud-nguyen-jtn7/zones/us-west1-b/instances/tf-lab-vm]

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

Outputs:
ip = "34.145.101.99"
```

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



| Status | Name | Zone | Recommendations | In use by | Internal IP | External IP | Network | Connect |
|-------------------------------------|---------------------------|------------|-----------------|-----------|---|---|-------------------------|---------|
| <input type="checkbox"/> | course-vm | us-west1-b | | | 10.138.0.2 (nic0) | | default | SSH |
| <input checked="" type="checkbox"/> | tf-lab-vm | us-west1-b | | | 10.138.0.16 (nic0) | 34.145.101.99 (nic0) | default | SSH |

Perform the plan again and check that the deployment is correct, then apply the change. After it has been applied, repeat the ssh command to log into the VM.

Take a screenshot of the successful ssh login from Cloud Shell.

Outputs:

```
ip = "34.145.101.99"
jtn7@cloudshell:~/tf (cloud-nguyen-jtn7)$ ssh 34.145.101.99
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1051-gcp x86_64)

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

System information as of Fri Feb 23 19:21:05 UTC 2024

System load:  0.0               Processes:    102
Usage of /:   20.6% of 9.51GB   Users logged in:  0
Memory usage: 5%               IPv4 address for ens4: 10.138.0.16
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.
```

Run terraform plan using this updated specification.

What resources are being added, changed, or destroyed?

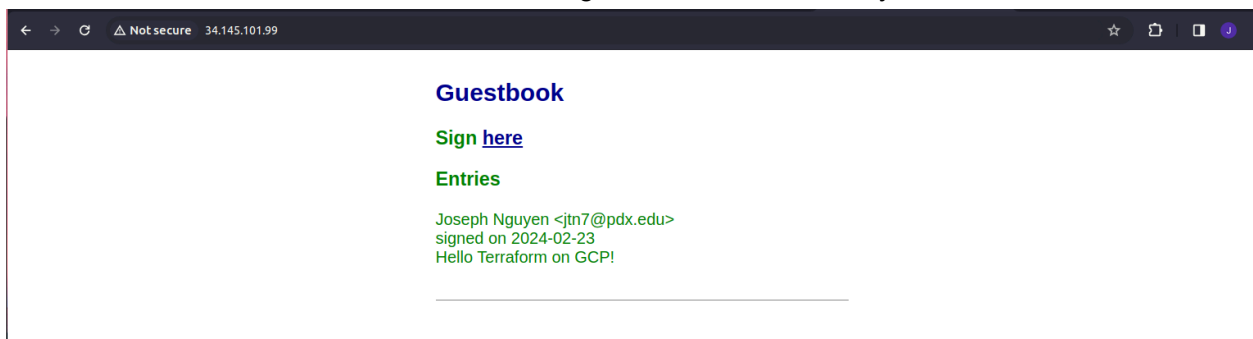
The metadata startup script is added as well as the “http-server” tag. This also changed the boot disk by adding disk_encryption_key and kms_key. Our network also have ipv6 access and address now. Scheduling and shielded_instance_config were removed. Overall the summary says 1 added and 1 removed. This is because google_cloud_instance must be replaced so we remove it and add a different one in.

What part of the configuration forces a replacement to occur?

The metadata startup script forces a replacement to occur.

Enter a message using your name, PSU e-mail address, and the message "Hello Terraform on GCP!".

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



7.2g

Go to Compute Engine and navigate around to answer the following questions in your lab notebook:

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

gke-guestbook-default-pool-919185e1

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

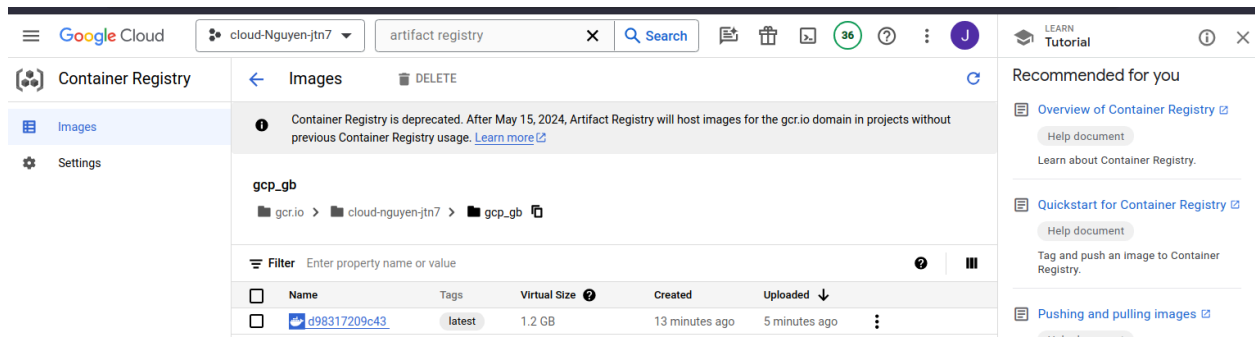
gke-guestbook-default-pool-919185e1-grp

What are the names of the two nodes?

gke-guestbook-default-pool-919185e1-b9z2

Gke-guestbook-default-pool-919185e1-vlzv

Visit the Artifact Registry UI and navigate to its container images section.
Take a screenshot of the container image created



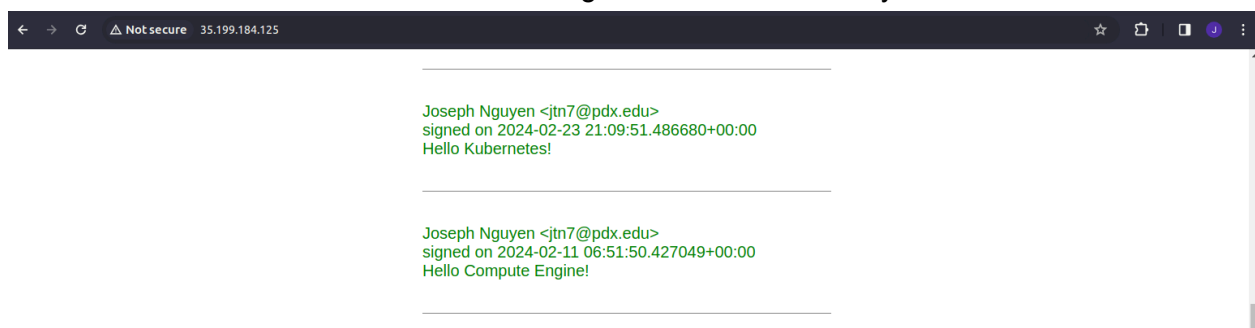
Take a screenshot of the output of the command `kubectl get pods` when all 3 replicas reach a "Running" state.

```
jtn7@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nguyen-jtn7)$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
guestbook-replicas-czhct            0/1     ContainerCreating   0          25s
guestbook-replicas-pmhc4            0/1     ContainerCreating   0          25s
guestbook-replicas-zwvr8            0/1     ContainerCreating   0          25s
jtn7@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nguyen-jtn7)$
```

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

```
jtn7@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nguyen-jtn7)$ kubectl get services
NAME            TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
guestbook-lb    LoadBalancer 10.20.3.196   35.199.184.125 80:32008/TCP     2m33s
kubernetes      ClusterIP     10.20.0.1    <none>        443/TCP          51m
jtn7@cloudshell:~/cs430-src/05_gcp_datastore (cloud-nguyen-jtn7)$
```

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



Visit Kubernetes Engine via the web console and view the "Workloads" placed on your cluster.
Click on the guestbook-replicas.
Take a screenshot of the managed guestbook pods and the service being exposed.

The screenshot shows the Google Cloud console interface for the Kubernetes Engine. The left sidebar displays the navigation menu with 'Workloads' selected. The main content area is titled 'Replication Controller details' and shows the following information:

- Pod specification:**
 - Labels: `app: guestbook`, `tier: frontend`
 - Termination grace period: 30
 - Restart policy: Always
 - Containers: [guestbook-app](#)
- Managed pods:**

| Name | Status | Restarts | Created on |
|--|---------|----------|--------------------------|
| guestbook-replicas-zwwr8 | Running | 0 | Feb 23, 2024, 1:06:08 PM |
| guestbook-replicas-pmgc4 | Running | 0 | Feb 23, 2024, 1:06:08 PM |
| guestbook-replicas-czhct | Running | 0 | Feb 23, 2024, 1:06:08 PM |
- Exposing services:**

| Name | Type | Endpoints |
|------------------------------|---------------|-----------------------------------|
| guestbook-lb | Load balancer | 35.199.184.125:80 |

Take a screenshot of the load balancer and its details

The screenshot shows the Google Cloud console interface for the Kubernetes Engine, specifically the 'Service details' page for the 'guestbook-lb' service. The left sidebar shows the 'Networking' section with 'Gateways, Services & Ingress' selected. The main content area displays the following details:

- Service overview:**
 - Cluster: [guestbook](#)
 - Namespace: default
 - Created: Feb 23, 2024, 1:06:07 PM
 - Labels: `app: guestbook`, `tier: frontend`
 - Annotations: `cloud.google.com/neg: {"ingress": true}`
 - Label selector: `app = guestbook`, `tier = frontend`
 - Pods: 3 current / 3 desired
 - Type: LoadBalancer
 - External endpoints: [35.199.184.125:80](#)
- Load Balancer details:**

| Field | Value |
|------------------|--|
| Cluster IP | 10.20.3.196 |
| Load balancer IP | 35.199.184.125 |
| Load balancer | a75dcf811e493483298f23860a21bf95 |

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

10.138.0.19 and 10.138.0.20 is associated with the 2 nodes

34.145.101.99 is associated with the load balancer

| Name | IP address | Access type | Region | Type | Version | In use by | Subnetwork |
|------|----------------|-------------|----------|-----------|---------|---|-------------------------|
| - | 10.138.0.2 | Internal | us-west1 | Ephemeral | IPv4 | VM instance course-vm (Zone us-west1-b) | default |
| - | 10.138.0.19 | Internal | us-west1 | Ephemeral | IPv4 | VM instance gke-guestbook-default-pool-919185e1-v1zv (Zone us-west1-b) | default |
| - | 10.138.0.20 | Internal | us-west1 | Ephemeral | IPv4 | VM instance gke-guestbook-default-pool-919185e1-69vz2 (Zone us-west1-b) | default |
| - | 34.145.101.99 | External | us-west1 | Ephemeral | IPv4 | VM instance gke-guestbook-default-pool-919185e1-69vz2 (Zone us-west1-b) | default |
| - | 34.168.116.210 | External | us-west1 | Ephemeral | IPv4 | VM instance gke-guestbook-default-pool-919185e1-v1zv (Zone us-west1-b) | default |
| - | 35.199.184.125 | External | us-west1 | Ephemeral | IPv4 | Forwarding rule a75dc-fb11e493483298f23860a21bf95 | |

Then, as before, examine the service that has been deployed by finding the external IP address of the load balancer and visiting it in the browser. Enter a message using your name, PSU e-mail address, and the message "Hello Cloud Build!".

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

Guestbook

[Sign here](#)

Entries

Joseph Nguyen <jtn7@pdx.edu>
signed on 2024-02-23 21:40:59.669138+00:00
Hello Cloud Build!

Joseph Nguyen <jtn7@pdx.edu>
signed on 2024-02-11 06:39:21.998981+00:00
Hello Cloud Shell!

7.3g

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

```
# [START functions_slack_request]
def make_search_request(query):
    req = kgsearch.entities().search(query=query, limit=1)
    res = req.execute()
    return format_slack_message(query, res)
```

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?

```
# [START functions_slack_search]
@functions_framework.http
def kg_search(request):
    if request.method != "POST":
        return "Only POST requests are accepted", 405

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

What is the Python data type that is used to represent the formatted message?
The query values are strings but the formatted message is stored in a dict.

What are the three main attributes of the formatted message passed back to Slack?
response_type, text, attachments

Go to your Slack workspace and join a channel. Within the text box, enter the following search.
/kg chatgpt Take a screenshot of its response for your lab notebook.



7.4

Run a detection that returns the labels generated with an image given its URI
python detect.py labels-uri gs://cloud-samples-data/ml-api-codelab/birds.jpg
Show the output for your lab notebook

```
(env) jtn7@cloudshell:~/python-docs-samples/vision/snippets/detect [cloud-nguyen-jtn7]$ python detect.py labels-uri gs://cloud-samples-data/ml-api-codelab/birds.jpg
Labels:
Bird
Ratite
Cloud
Sky
Beak
Plant
Green
Neck
Ostrich
Casuariiformes
(env) jtn7@cloudshell:~/python-docs-samples/vision/snippets/detect [cloud-nguyen-jtn7]$
```

Open up detect.py. Given the arguments of the above command, find the function that is called which handles this particular detection.

Answer the following questions:

What is the name of the function?
detect_labels

What type of Vision client is instantiated in it?
ImageAnnotatorClient

What method is invoked in the Vision client to perform the detection?
label_detection

What is the name of the attribute in the response object that contains the results we seek?
Label_annotations

The Vision API can also detect logos in images. Go back to detect.py and find how it can be used to perform a logo detection on a local file. Then, 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.

Take a screenshot of the output for the above commands

```
(env) jtn7@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-nguyen-jtn7)$ python detect.py logos/logo.png
Logos:
University of Dallas
(env) jtn7@cloudshell:~/python-docs-samples/vision/snippets/detect (cloud-nguyen-jtn7)$
```

What method is invoked in the Vision client to perform the detection?

logo_detection

Run a detection that returns the translation of a raw audio clip:

python transcribe.py resources/audio.raw

Show the output for your lab notebook

```
(env) jtn7@cloudshell:~/python-docs-samples/speech/snippets (cloud-nguyen-jtn7)$ python transcribe.py resources/audio.raw
Transcript: how old is the Brooklyn Bridge
(env) jtn7@cloudshell:~/python-docs-samples/speech/snippets (cloud-nguyen-jtn7)$
```

Open up transcribe.py. Given the arguments of the above command, find the function that is called which handles this particular translation.

What is the name of the function?

transcribe_file

What method is invoked in the Speech client to perform the detection?

recognize

What is the name of the attribute in the response object that contains the results we seek?

results

Run a detection that returns the English translation of a Chinese sentence.

python snippets.py translate-text en '你有沒有帶外套'

Show the output for your lab notebook

```
(env) jtn7@cloudshell:~/python-docs-samples/translate/samples/snippets (cloud-nguyen-jtn7)$ 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) jtn7@cloudshell:~/python-docs-samples/translate/samples/snippets (cloud-nguyen-jtn7)$
```

Open up snippets.py. Given the arguments of the above command, find the function that is called which handles this particular translation.

What is the name of the function?

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?

result["input"], result["translated_text"], and result["detectedSourceLanguage"]

Create a file called language.py that takes a text string, performs sentiment analysis on it, and then pulls out the entities in the text.

Run the following analyses:

```
(env) jtn7@cloudshell:~ (cloud-nguyen-jtn7)$ 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
```

Examine the code and answer the following questions:

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?

compare_audio_to_image

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 current program does a naive check between entities and labels. Upon closer inspection of the results, football and Football were not seen as equivalent so we can make the entities and labels case insensitive when performing the check.

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 current program does a naive check between entities and labels. Upon closer inspection of the results, bike and Bicycle were not seen as equivalent so we can make the entities and labels understand synonyms better in addition to being case insensitive.

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 current program does a naive check between entities and labels. Upon closer inspection of the results, ostriches and Ostrich were not seen as equivalent so we can make the entities and labels match singular and plural forms as the same in addition to recognizing synonyms and being case insensitive.

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, and player

Open up labels.py. Answer the following questions:

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

Joy detection website screenshot



saitama.png was uploaded 2024-02-24 04:35:29.665861+00:00.

Joy Likelihood for Face: Very Unlikely

Open main.py and view the code for the default route.

Answer the following questions:

What line of code creates the query for previous detections?

query = datastore_client.query(kind="Faces")

What line of code sends the query to Cloud Datastore?

Image_entities = list(query.fetch())

Then, view the `upload_photo` route.

Show the line that retrieves the name of the storage bucket to use.

`bucket = storage_client.get_bucket(CLOUD_STORAGE_BUCKET)`

What form field is used to specify the uploaded photo?

`blob = bucket.blob(photo.filename)`

Show the line that copies the photo's contents to the storage bucket.

`blob.upload_from_string(photo.read(), content_type=photo.content_type)`

What method in Vision's annotation client is used to perform the analysis?

`face_detection(image=image).face_annotations`

What fields are stored in Cloud Datastore for each image?

`Blob_name, image_public_url, timestamp, joy`

What happens at the end of the `upload_photo` route?

We are directed to the homepage.

7.1a

Take a screenshot showing the completion of the command (`terraform apply`) including its output

```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run 'terraform apply'
[cloudshell-user@ip-10-130-178-195 ~]$ 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-0f5515e02f9d13ed" = (known after apply)
  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" = (known after apply)
  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.

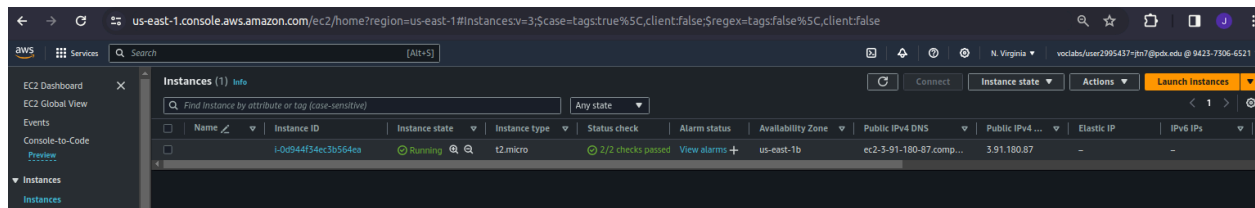
Enter a value: yes

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 35s [id=I-0d944f34ec3b554ee]

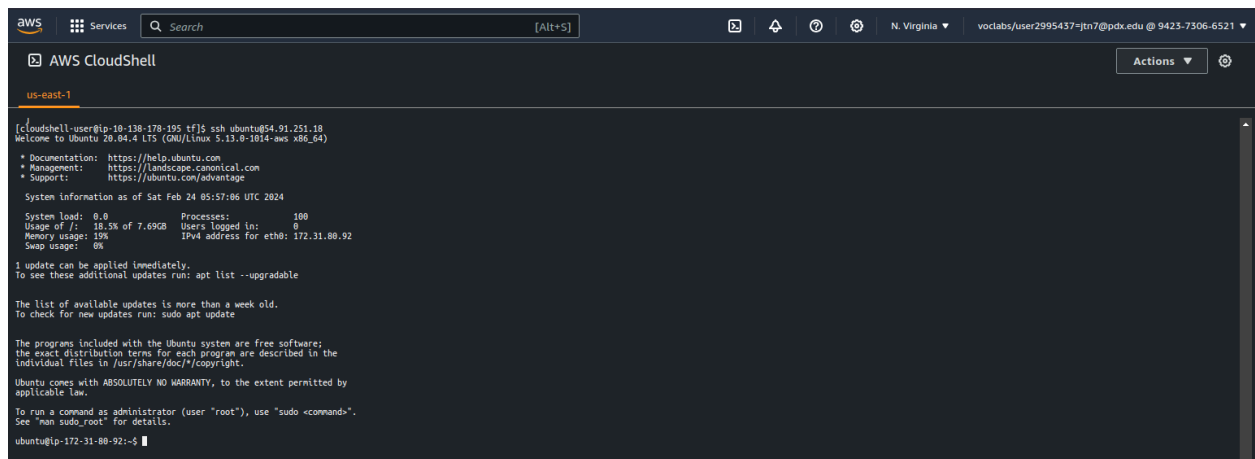
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
ec2Instance = "3.91.180.87"
[cloudshell-user@ip-10-130-178-195 ~]$
```

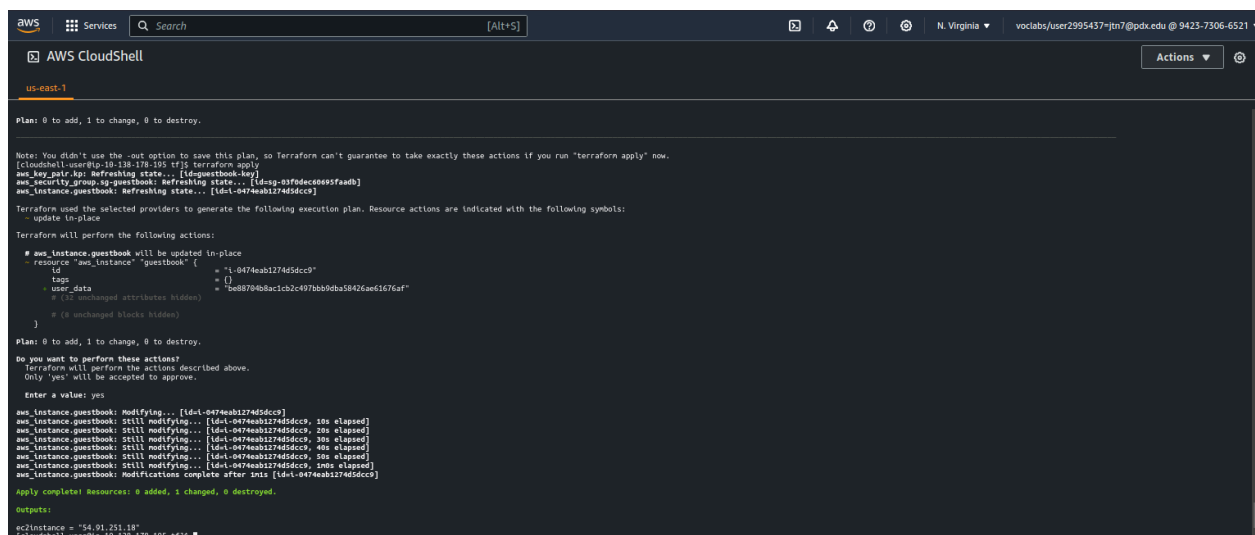
Visit EC2 within the web console and refresh it to see that the IP address has been bound to the VM. Take a screenshot that includes the VM's IP addresses



Plan and apply the change with terraform, then repeat the ssh command to log into the VM. Take a screenshot of the successful ssh login from Cloud Shell.



Plan and apply the configuration using terraform. Take a screenshot of the output of the command that includes the IP address of the instance



Visit the external IP address for the VM using a web browser. Enter a message using your name, PSU e-mail address, and the message "Hello Terraform on AWS!".

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

