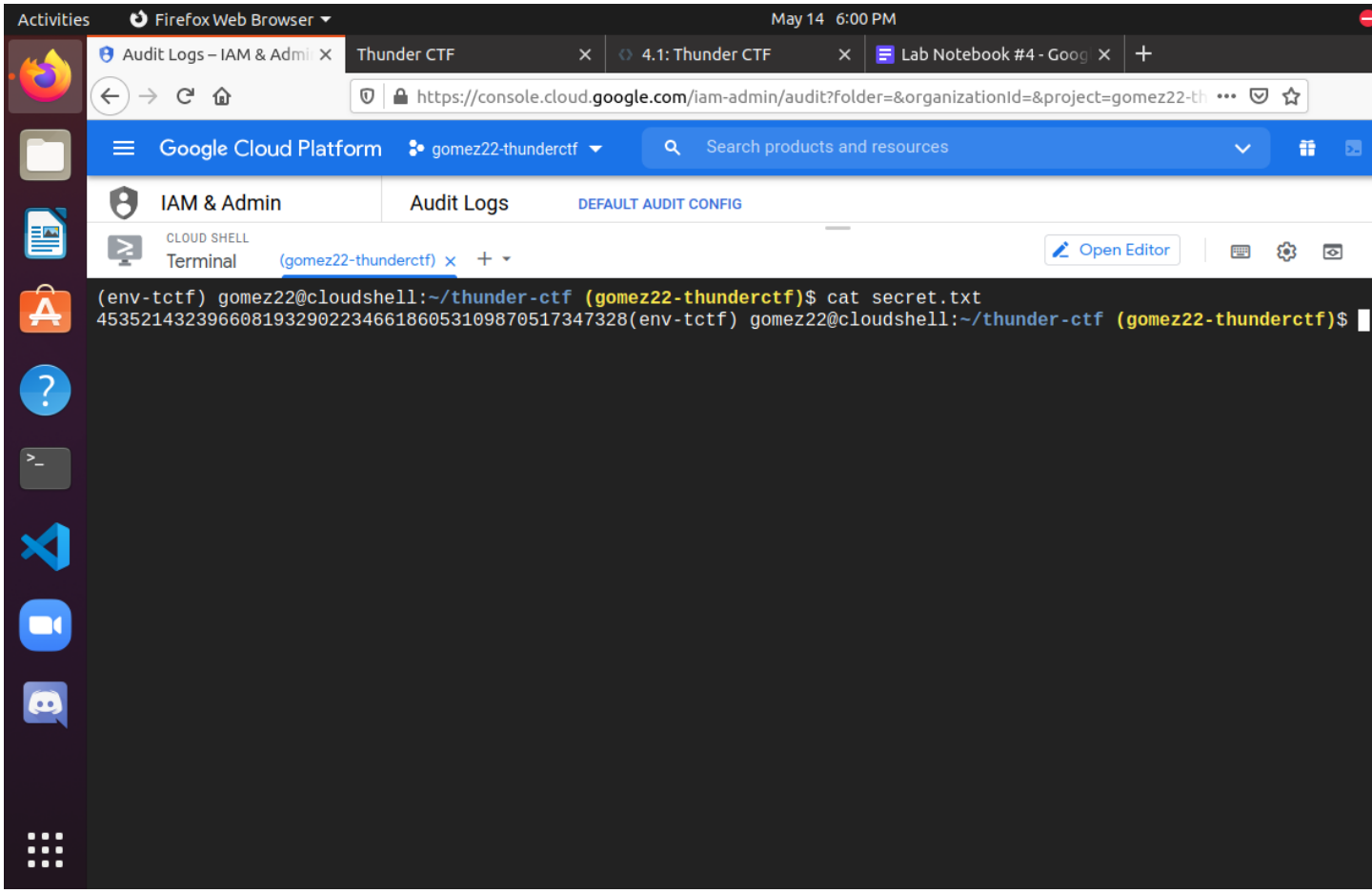
Tristan Gomez
CS 595

Notebook #4

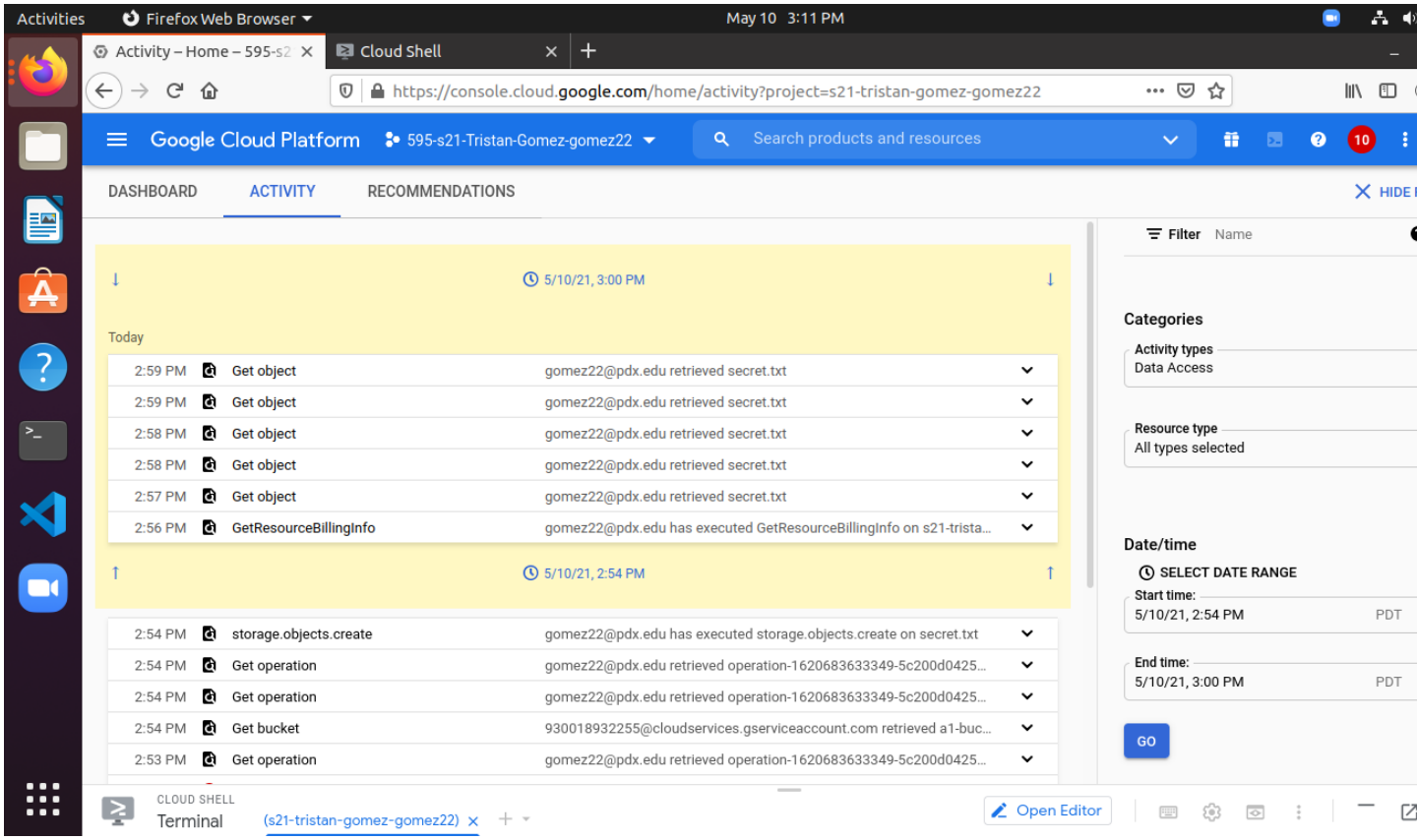
4.1 Thunder CTF

A1openbucket

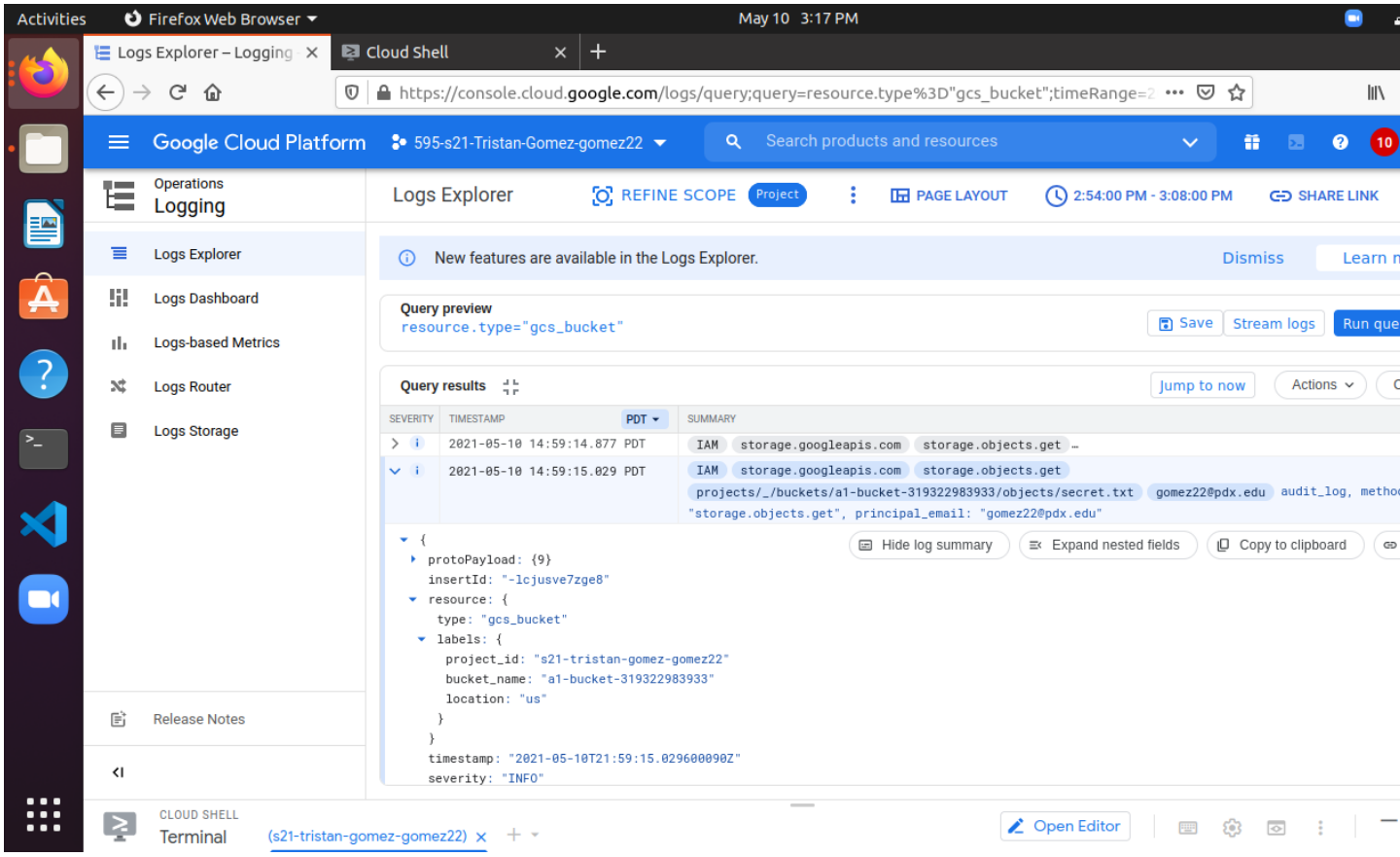
Take a screenshot of the secret obtained



Take a screenshot of these entries for your lab notebook.



Take a screenshot of this entry for your lab notebook.



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

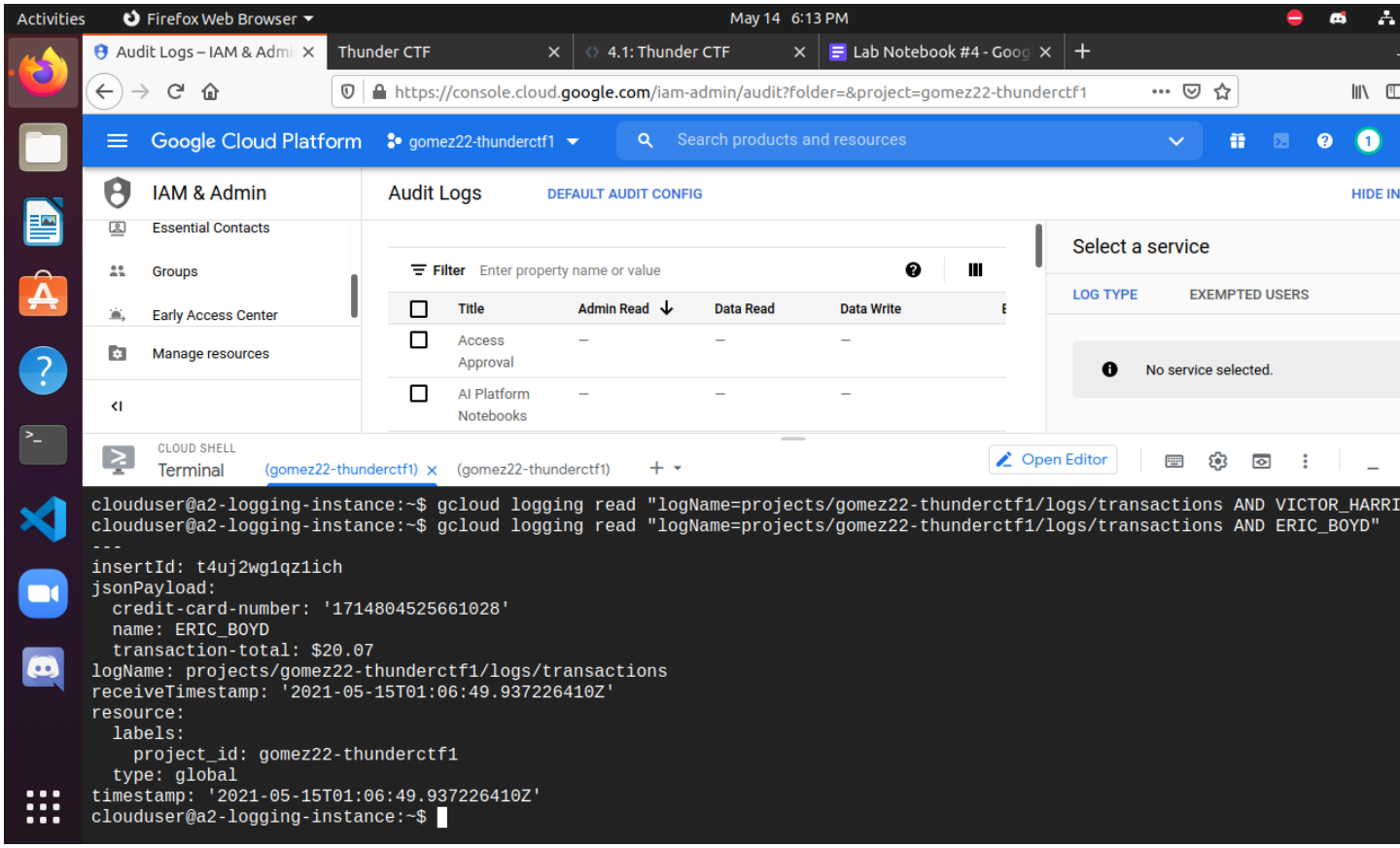
-The method name is “v2.deploymentmanager.deployments.insert” and the principalEmail address is “gomez22@pdx.edu”

What is the methodName of the deletion command?

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

A2finance

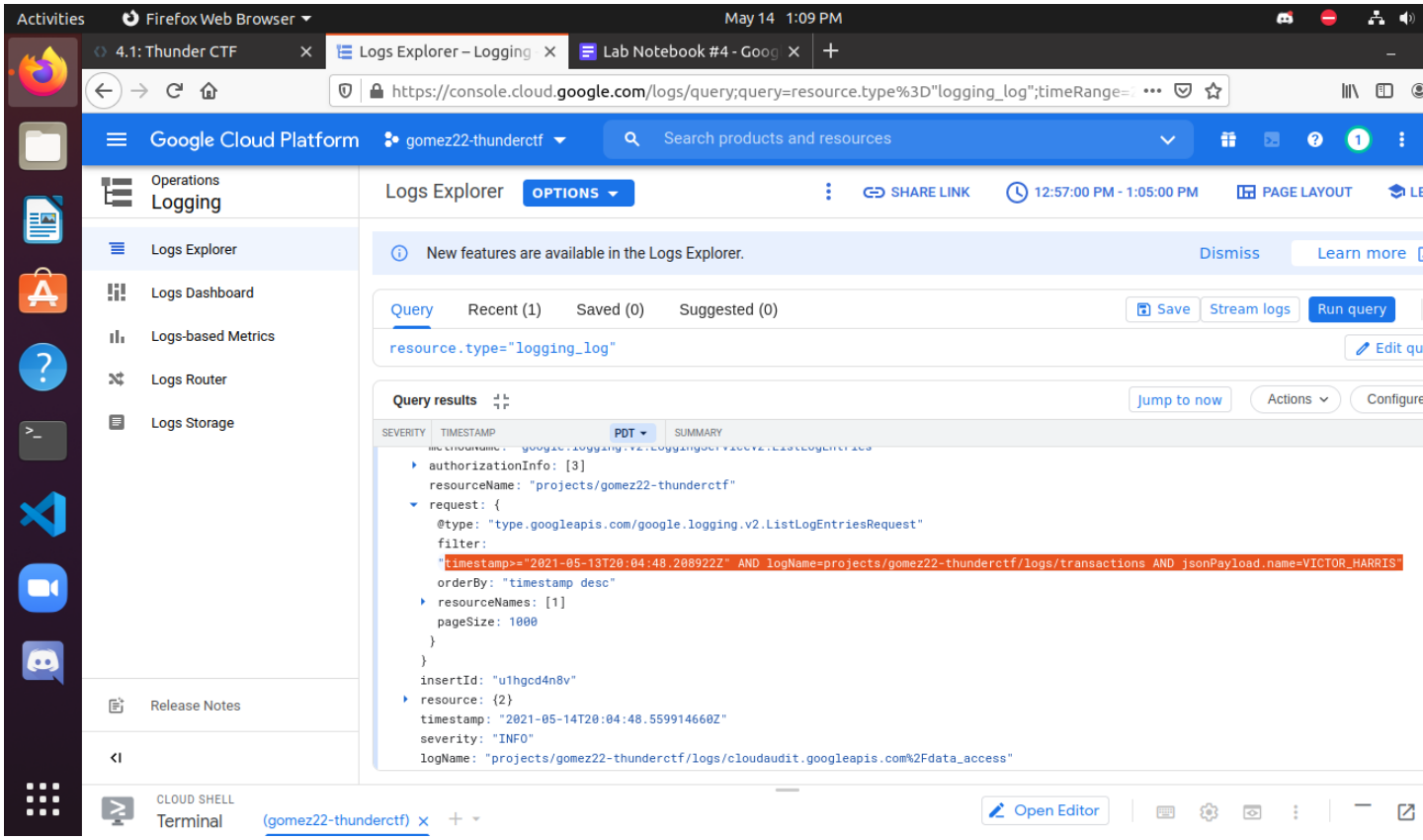
Take a screenshot of the secret obtained



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)



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.

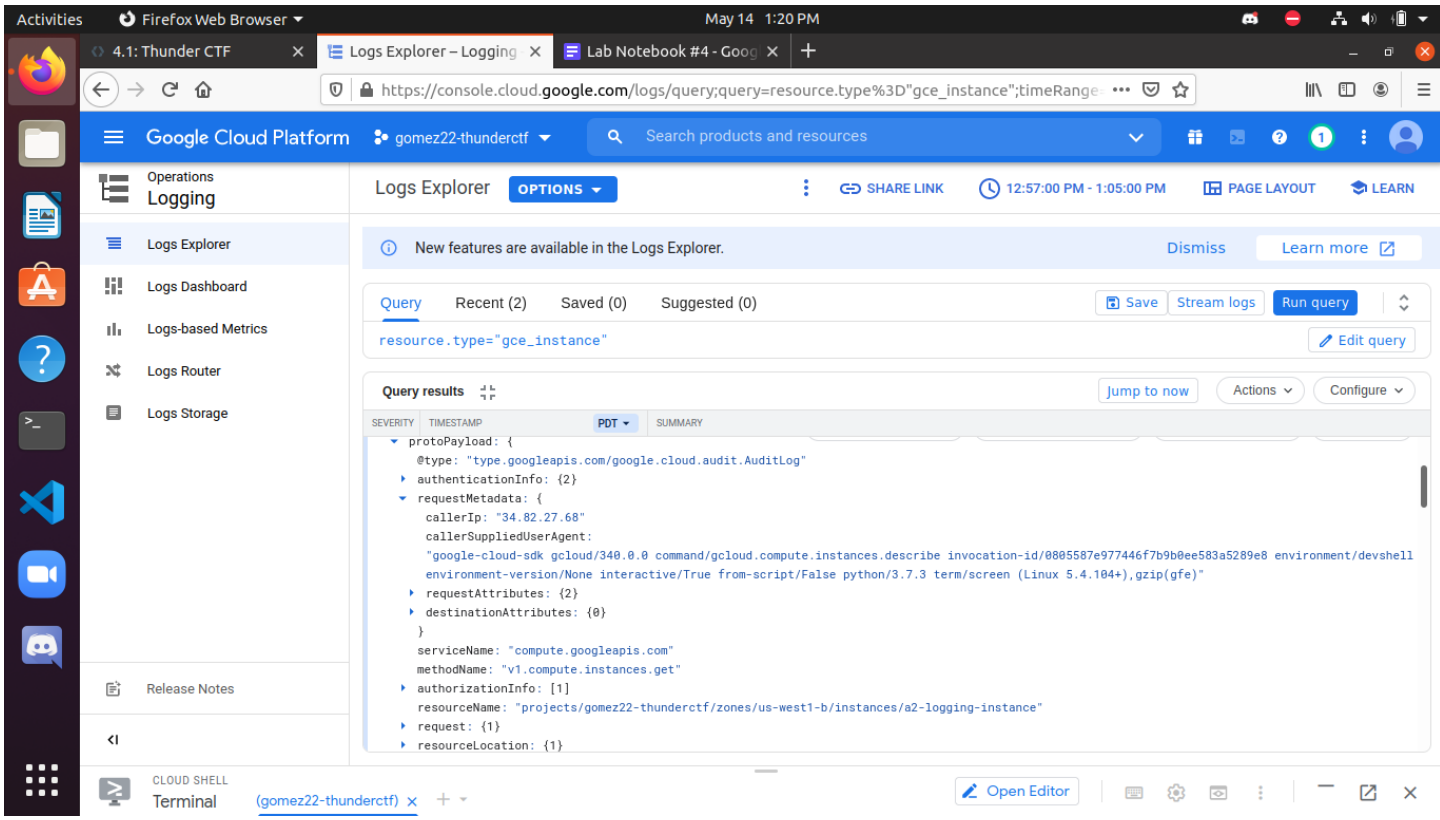
What is the service account key name used to perform this operation? (We would want to delete this key if this were an actual compromise.)

-"//iam.googleapis.com/projects/gomez22-thunderctf/serviceAccounts/a2-access@gomez22-thunderctf.iam.gserviceaccount.com/keys/26447e57632b2370d7e3849f086a8cc3017f4987"

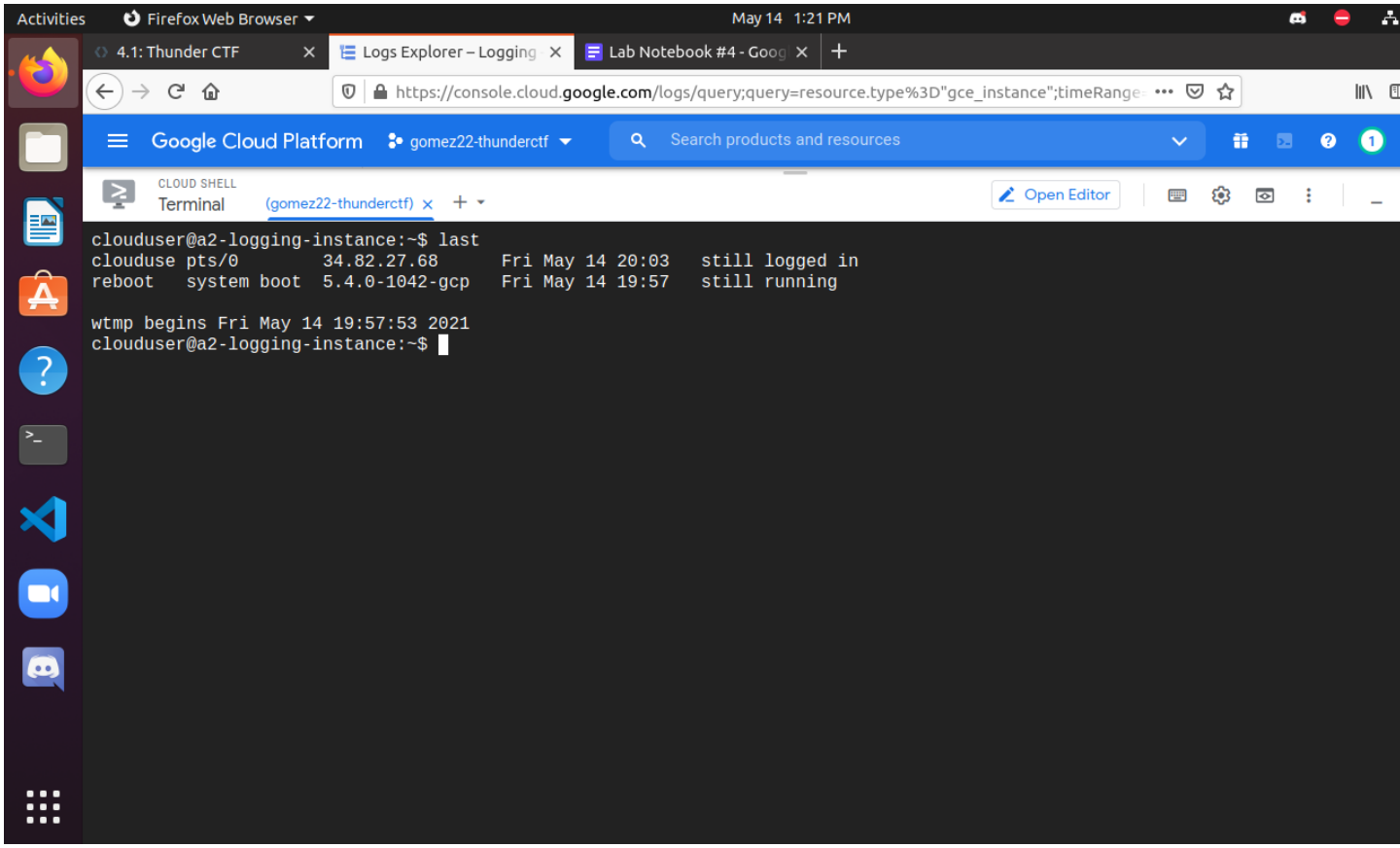
What does each log entry correspond to?

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

Show the IP address and UserAgent for this request.

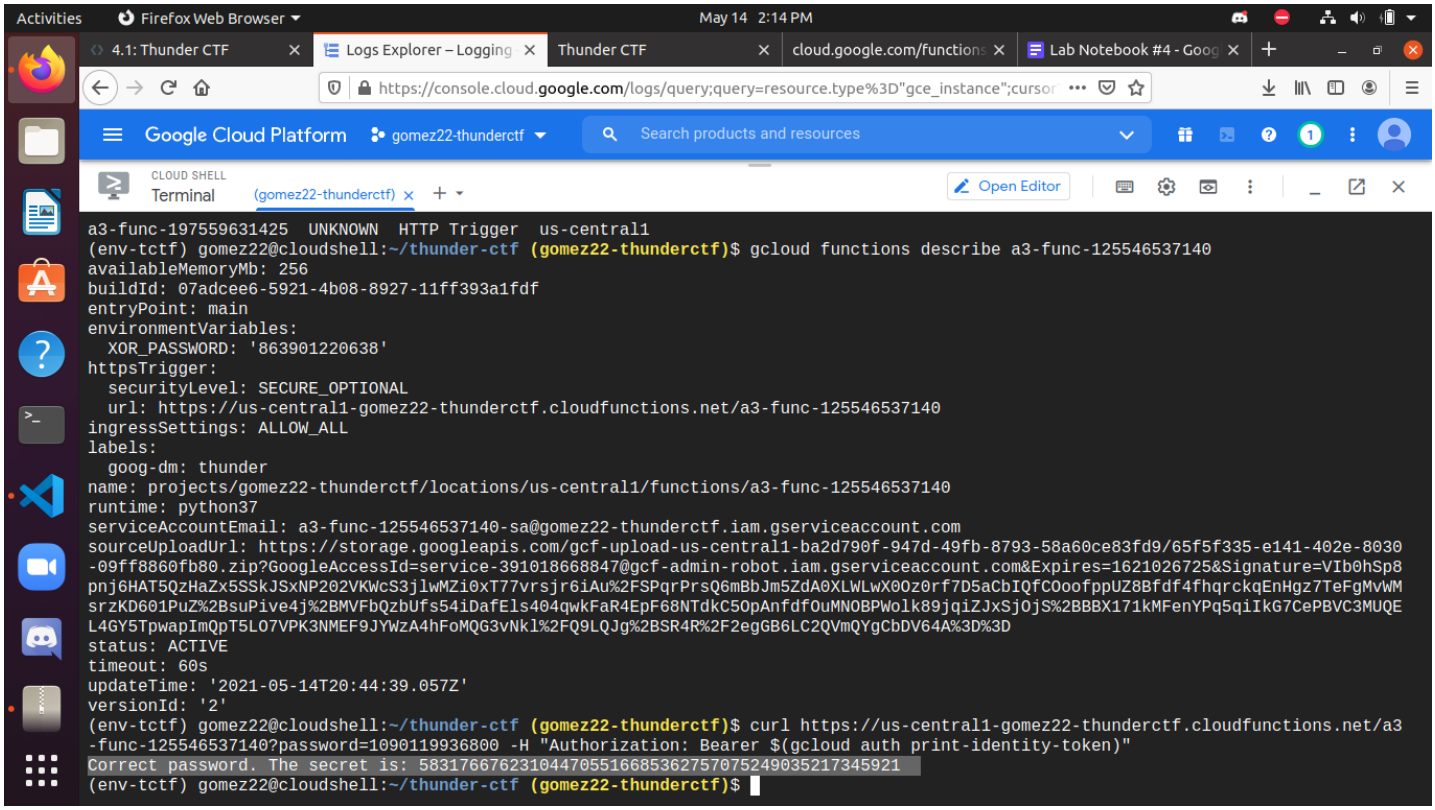


Show the output of the command when run on the VM

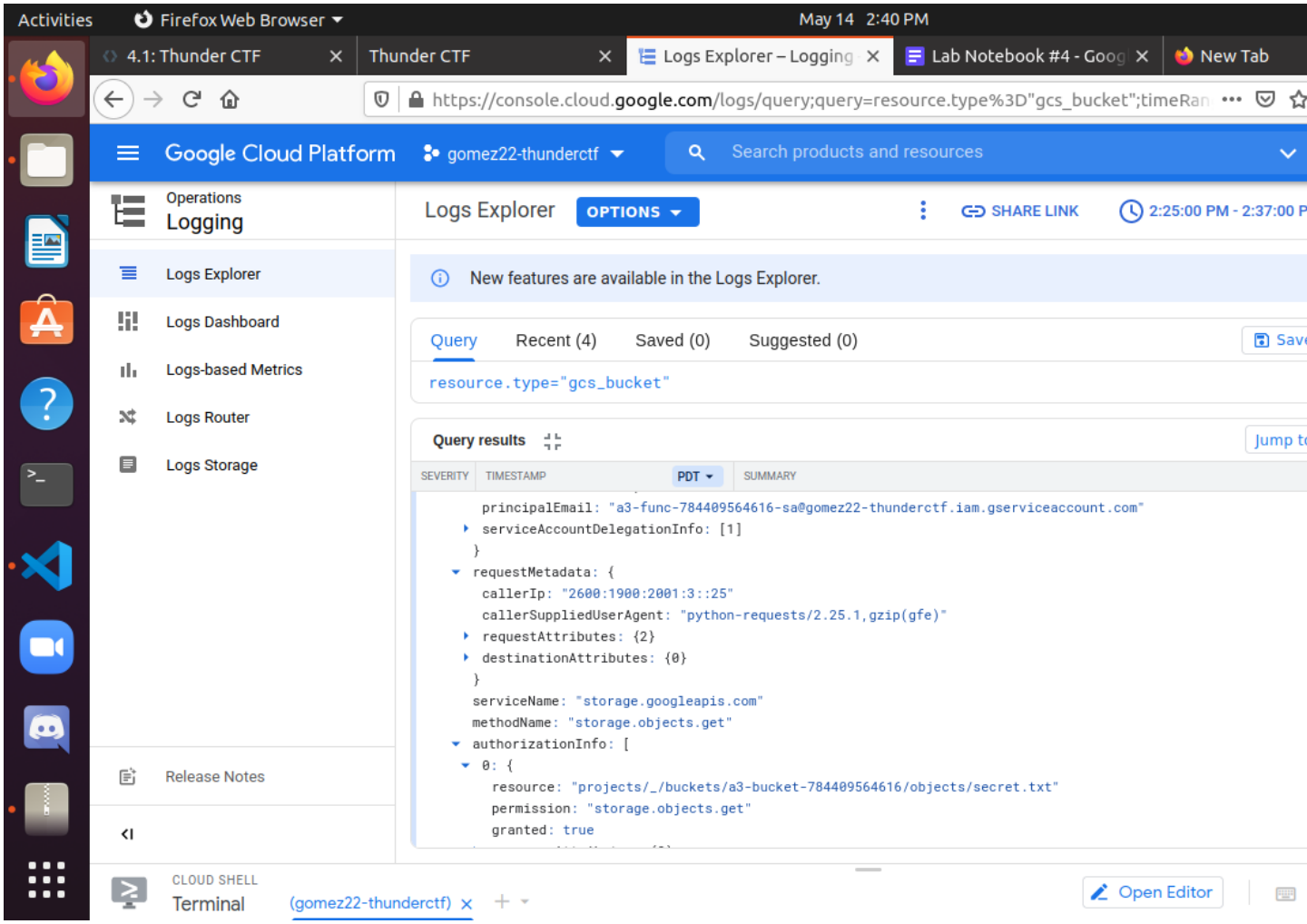


A3password

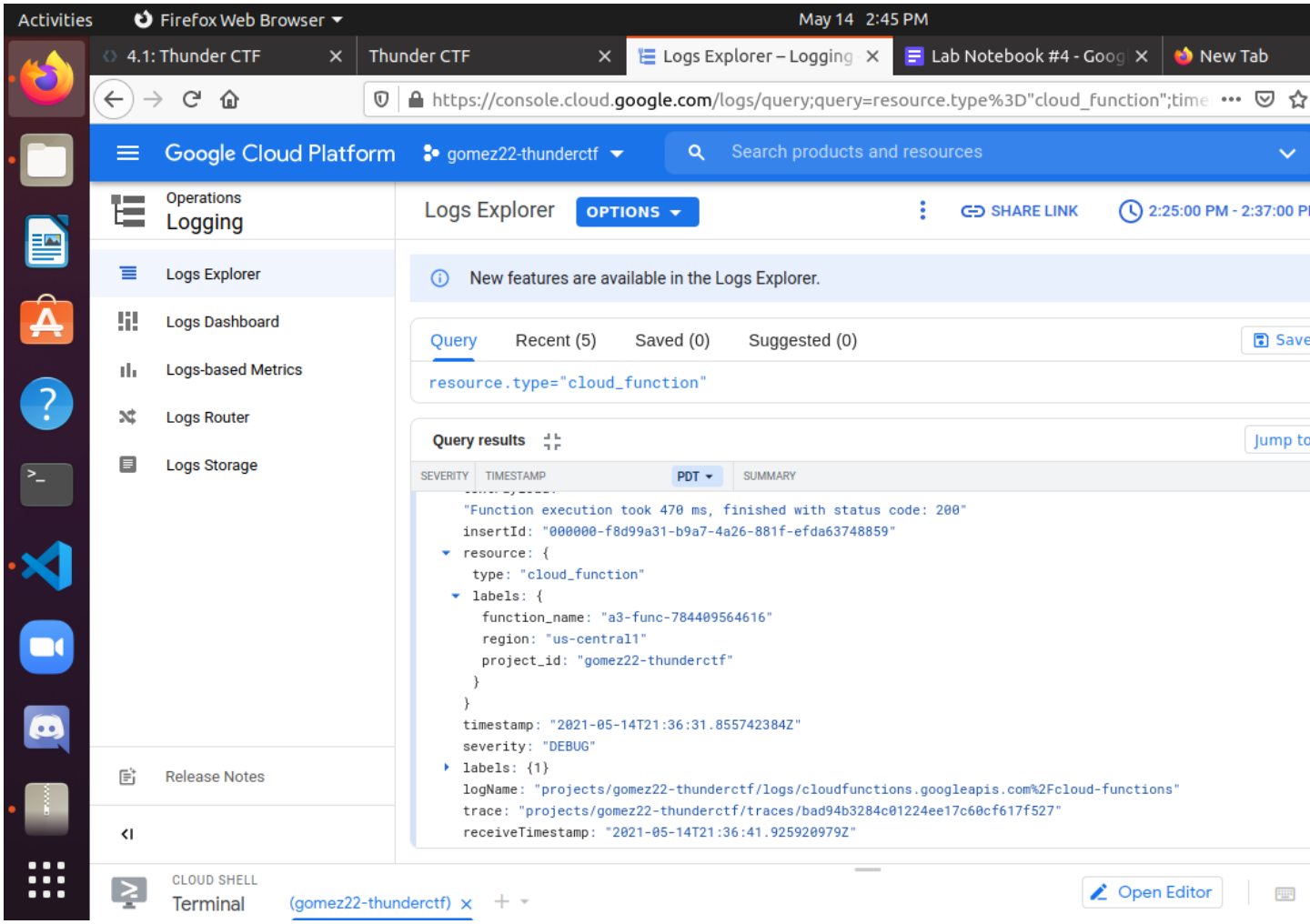
Take a screenshot of the secret obtained



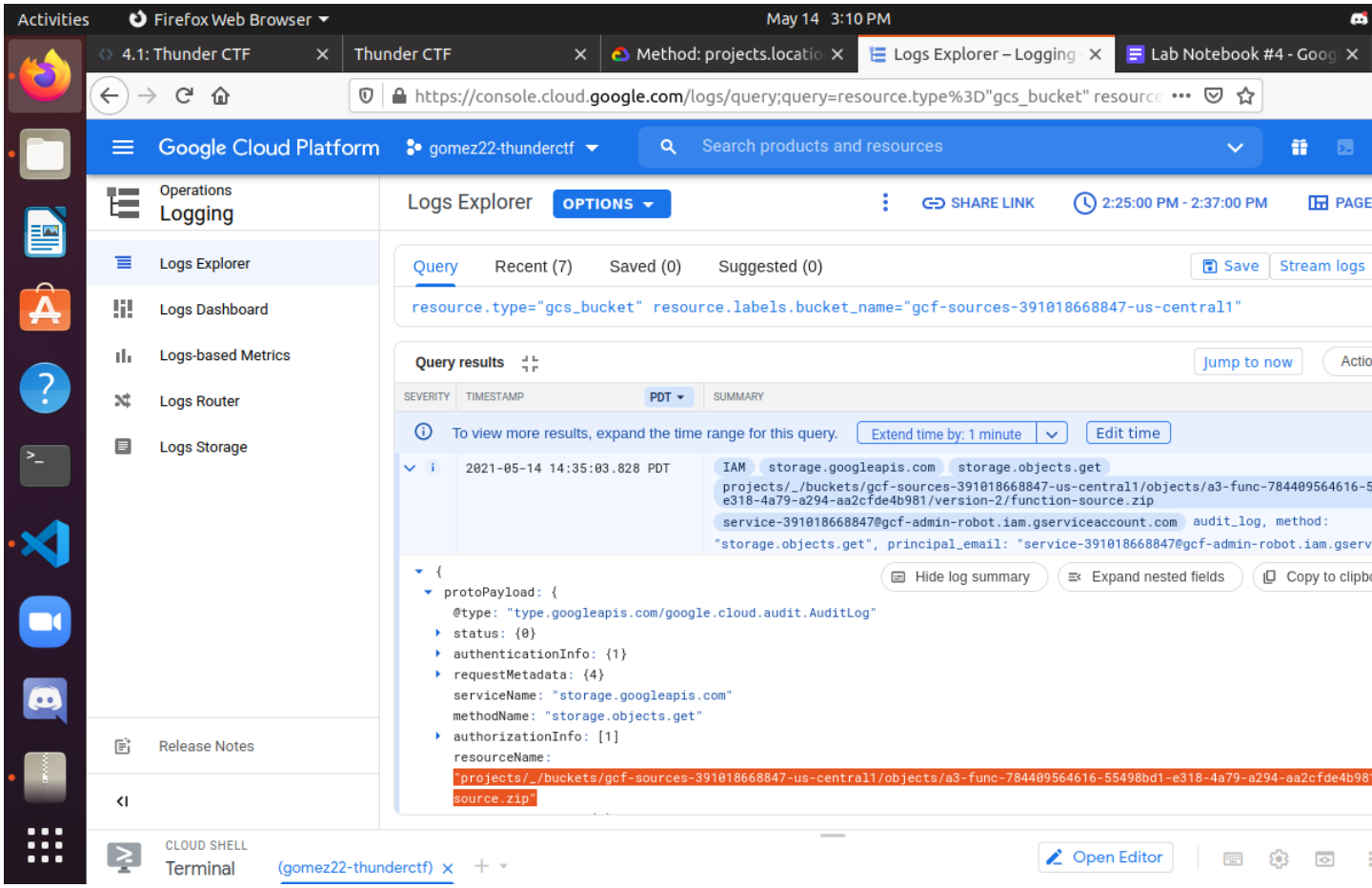
Expand the log entry to show the resource name of the file, the service account used to access it, and the User Agent that the cloud function uses to obtain the contents of it. Take a screenshot of this entry for your lab notebook.



Take a screenshot of this entry for your lab notebook.



Take a screenshot of this entry for your lab notebook.



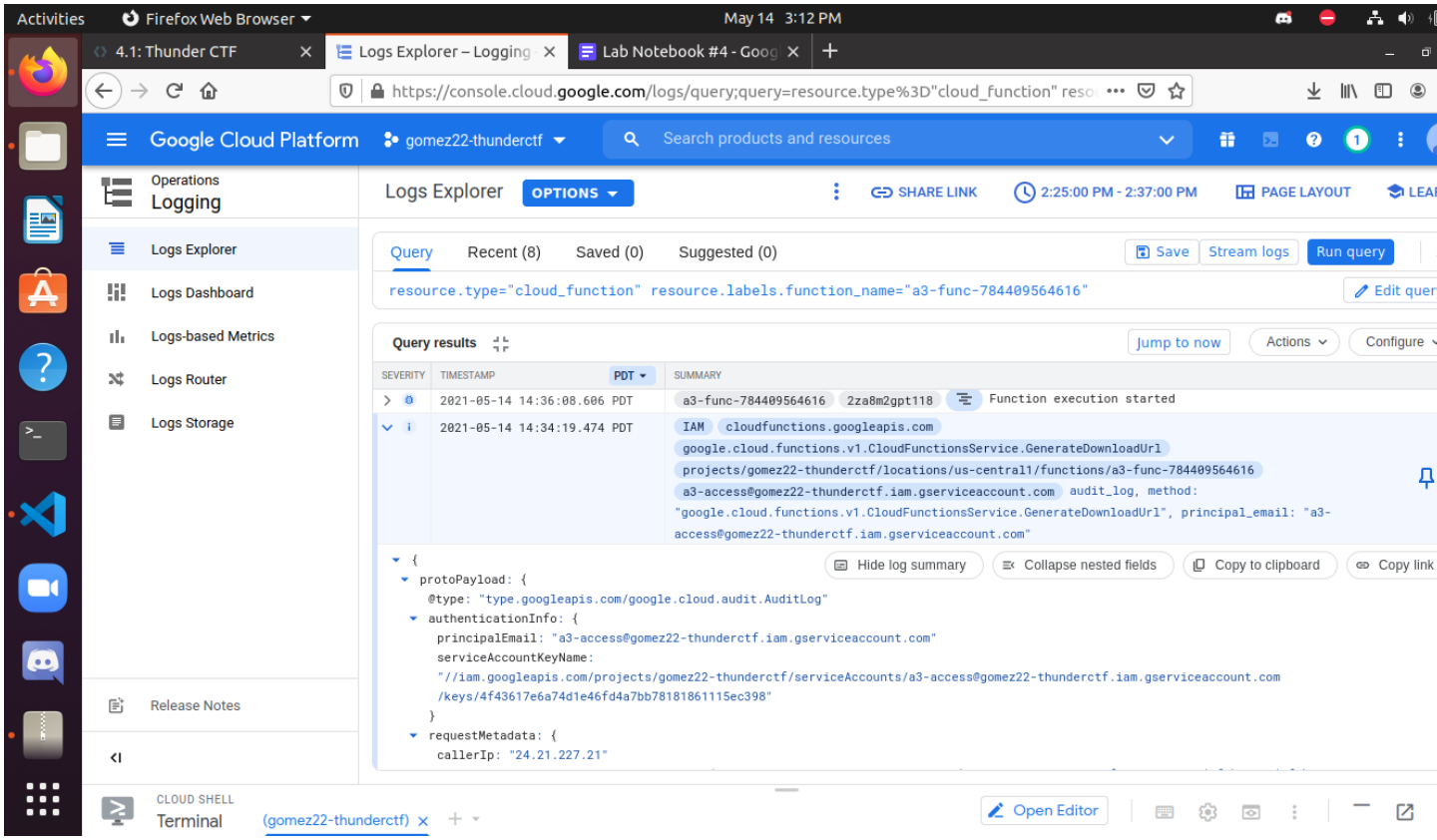
What is the service account that performs the operation, the service account key name, the authorization permission included and the methodName used in this operation?

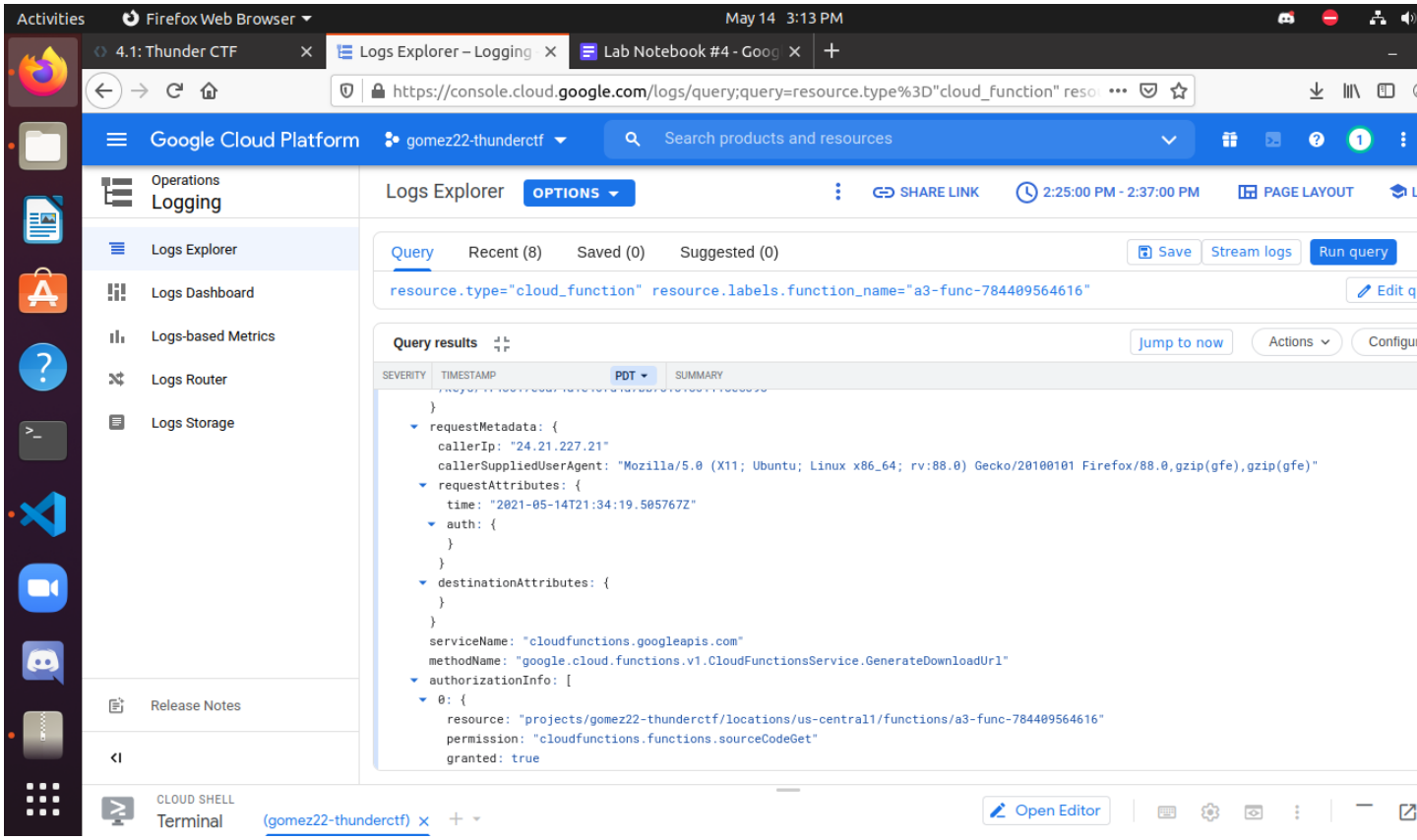
-The service account is:
“a3-access@gomez22-thunderctf.iam.gserviceaccount.com”

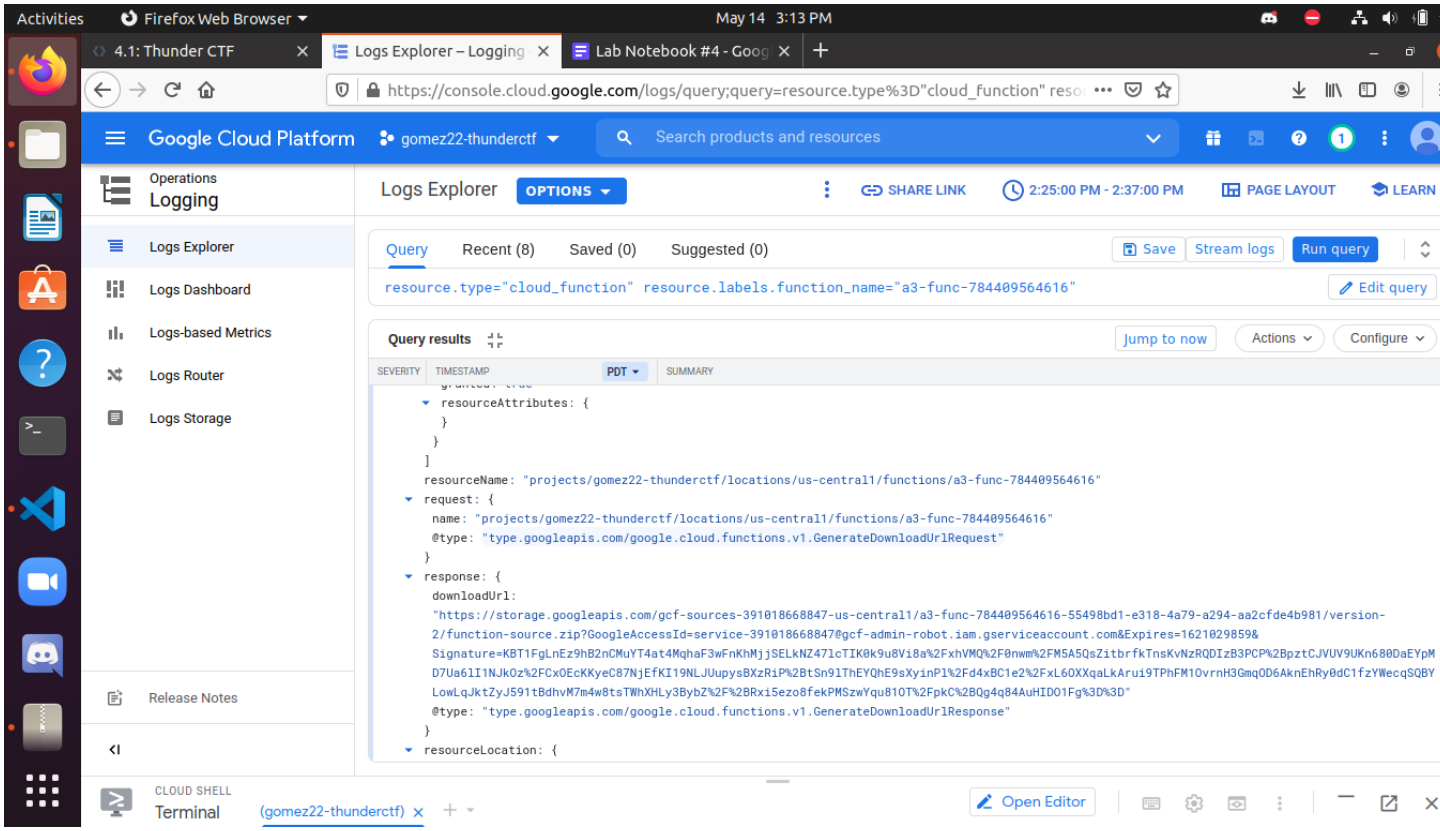
-The service account key name is:
”//iam.googleapis.com/projects/gomez22-thunderctf/serviceAccounts/a3-access@gomez22-thunderctf.iam.gserviceaccount.com/keys/4f43617e6a74d1e46fd4a7bb78181861115ec398”

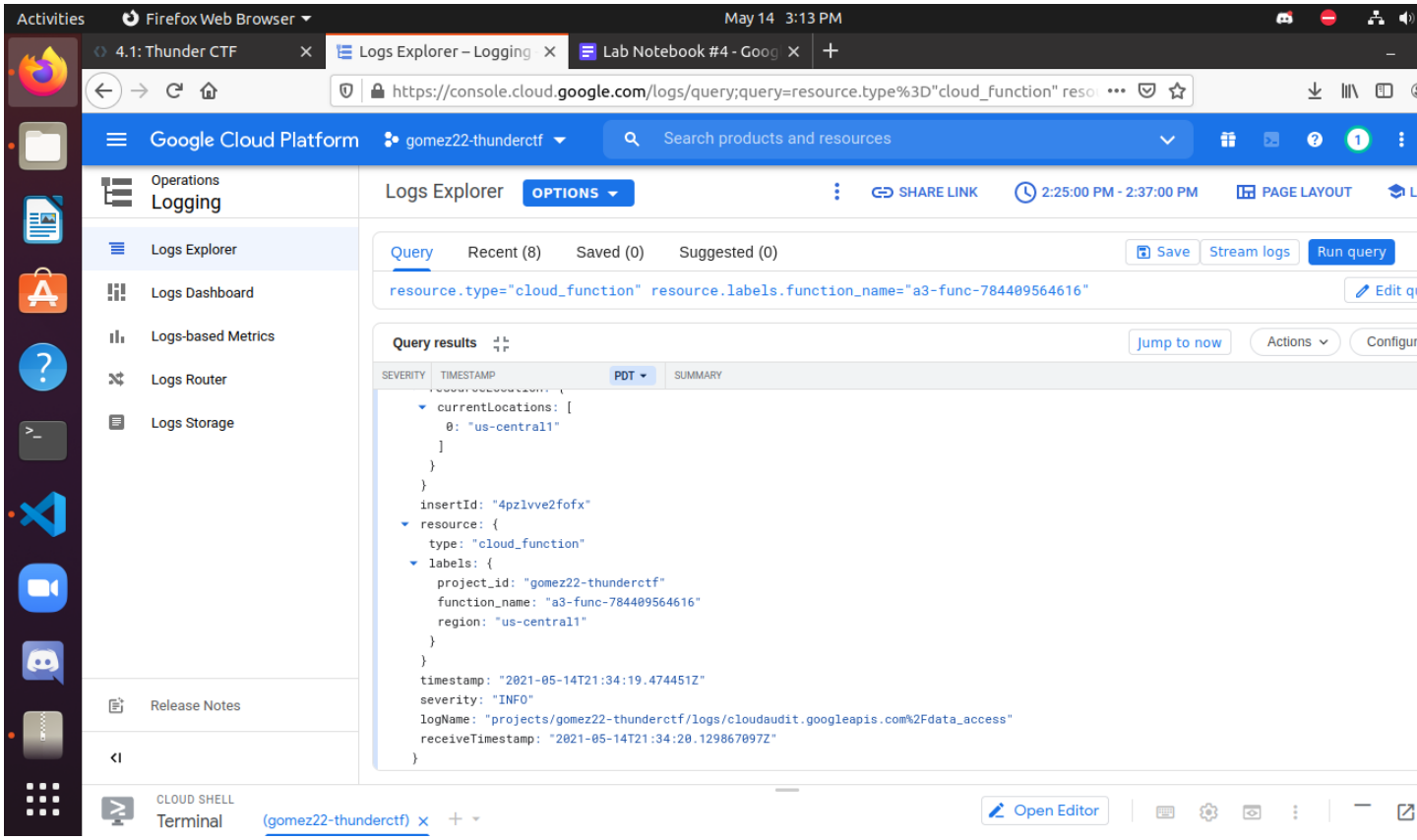
-The authorization permission is: “cloudfunctions.functions.sourceCodeGet”

-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
Logs Dashboard
Logs-based Metrics
Logs Router
Logs Storage
Release Notes
Logs Explorer
OPTIONS
SHARE LINK
2:25:00 PM - 2:37:00 PM
PAGE LAYOUT
Query
Recent (8)
Saved (0)
Suggested (0)
Save
Stream logs
Run query
resource.type="cloud_function" resource.labels.function_name="a3-func-784409564616"
Edit quer
Query results
Jump to now
Actions
Configure
SEVERITY
TIMESTAMP
PDT
SUMMARY
2021-05-14 14:36:08.606 PDT
a3-func-784409564616
2za8m2gpt118
Function execution started
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"
Hide log summary
Collapse nested fields
Copy to clipboard
Copy link
{
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/4f43617e6a74d1e46fd4a7bb78181861115ec398"
}
requestMetadata: {
callerIp: "24.21.227.21"
CLOUD SHELL
Terminal
(gomez22-thunderctf)
Open Editor

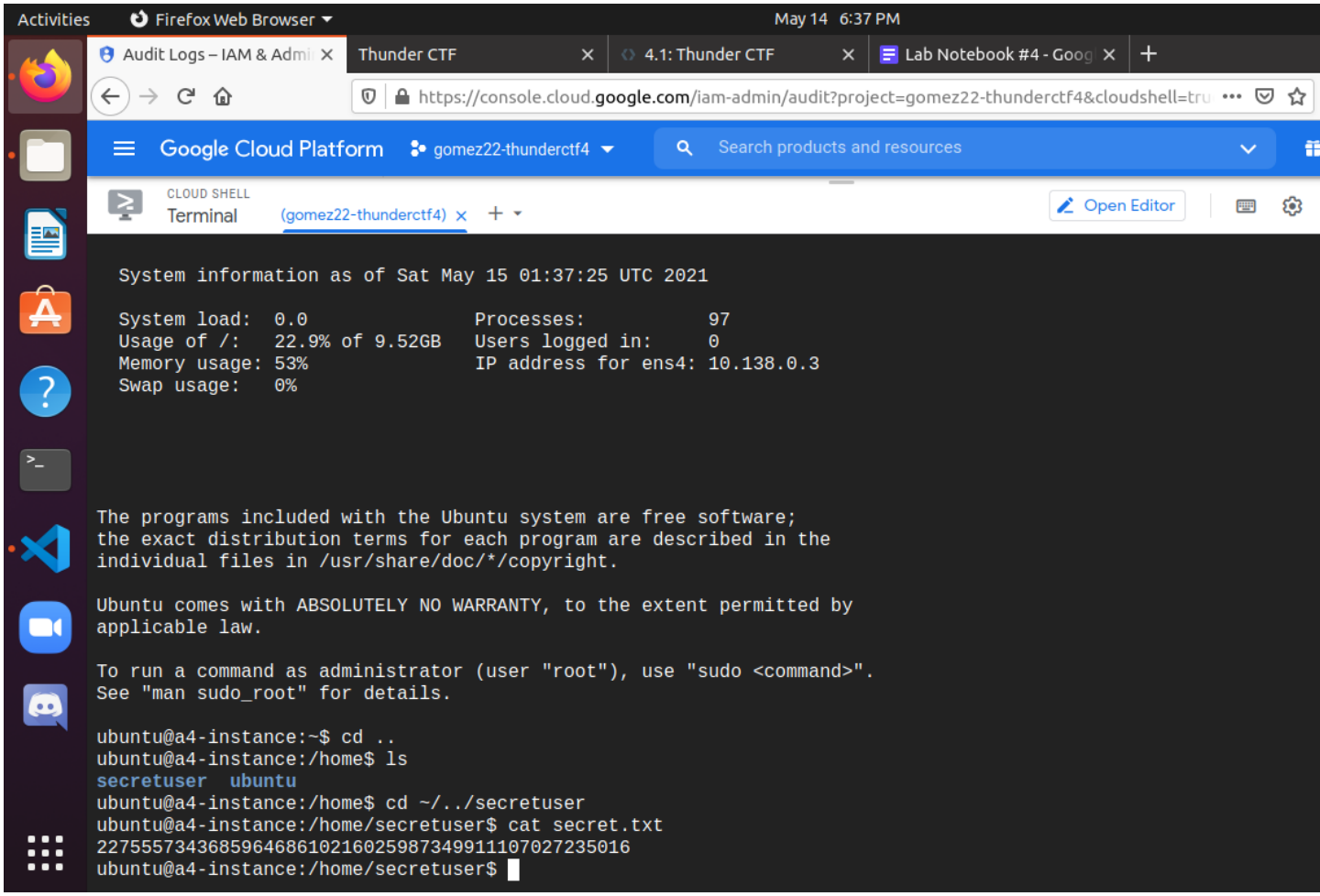

Activities
Firefox Web Browser
May 14 3:13 PM
4.1: Thunder CTF
Logs Explorer – Logging
Lab Notebook #4 - Goog
https://console.cloud.google.com/logs/query;query=resource.type%3D"cloud_function" reso
Google Cloud Platform
gomez22-thunderctf
Search products and resources
Operations
Logging
Logs Explorer
Logs Dashboard
Logs-based Metrics
Logs Router
Logs Storage
Release Notes
Logs Explorer
OPTIONS
SHARE LINK
2:25:00 PM - 2:37:00 PM
PAGE LAYOUT
Query
Recent (8)
Saved (0)
Suggested (0)
Save
Stream logs
Run query
resource.type="cloud_function" resource.labels.function_name="a3-func-784409564616"
Edit q
Query results
Jump to now
Actions
Configu
SEVERITY
TIMESTAMP
PDT
SUMMARY
}
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: "2021-05-14T21:34:19.505767Z"
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
CLOUD SHELL
Terminal
(gomez22-thunderctf)
Open Editor


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
Google Cloud Platform
gomez22-thunderctf
Search products and resources
Operations
Logging
Logs Explorer
Logs Dashboard
Logs-based Metrics
Logs Router
Logs Storage
Release Notes
Logs Explorer
OPTIONS
SHARE LINK
2:25:00 PM - 2:37:00 PM
PAGE LAYOUT
LEARN
Query
Recent (8)
Saved (0)
Suggested (0)
Save
Stream logs
Run query
resource.type="cloud_function" resource.labels.function_name="a3-func-784409564616"
Edit query
Query results
Jump to now
Actions
Configure
SEVERITY
TIMESTAMP
PDT
SUMMARY
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=1621029859&
Signature=KBT1FgLnEz9hB2nCMuYT4at4MqhaF3wFnKhMjjSELkNZ471cTIK0k9u8Vi8a%2FxhVMQ%2F0nwm%2FM5A5QsZitbrfkTnsKvNzRQDIzB3PCP%2BpztCJVUV9UKn680DaEYpMD7Ua6lI1NJkOz%2FCxOEcKKyeC87NjEfKI19NLJUupysBXzRiP%2BtSn91ThEYQhE9sXyinP1%2Fd4xBC1e2%2FxL6OXXqaLkArui9TPhFM1OvrnH3GmqOD6AknEhRy0dC1fzYWecqSQBYLowLqJktZyJ591tBdhvM7m4w8tsTWhXHLy3BybZ%2F%2BRxi5ezo8fekPMSzwYqu81OT%2FpkC%2BQg4q84AuHIDO1Fg%3D%3D"
@type: "type.googleapis.com/google.cloud.functions.v1.GenerateDownloadUrlResponse"
}
resourceLocation: {
CLOUD SHELL
Terminal
(gomez22-thunderctf)
Open Editor

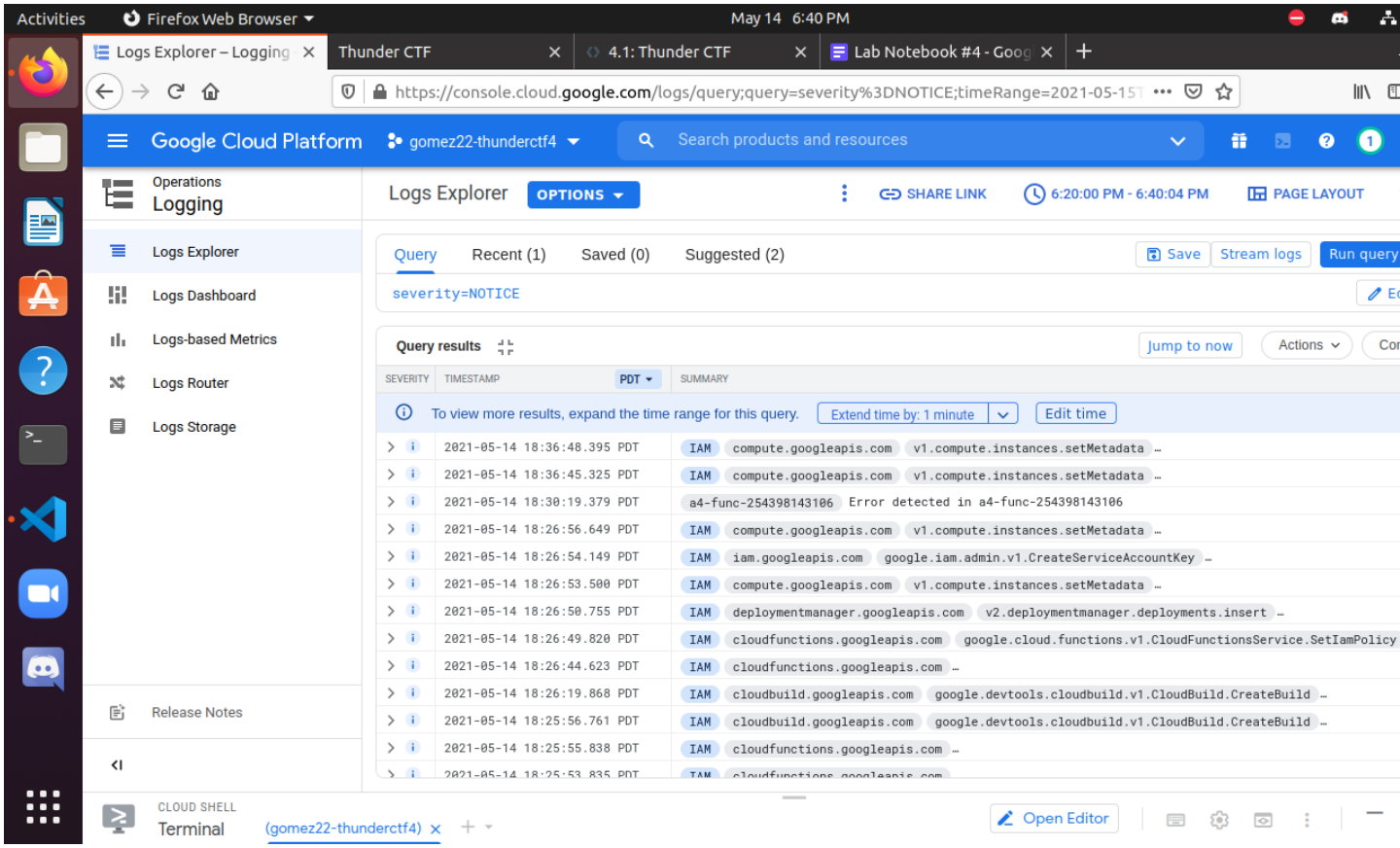


A4error

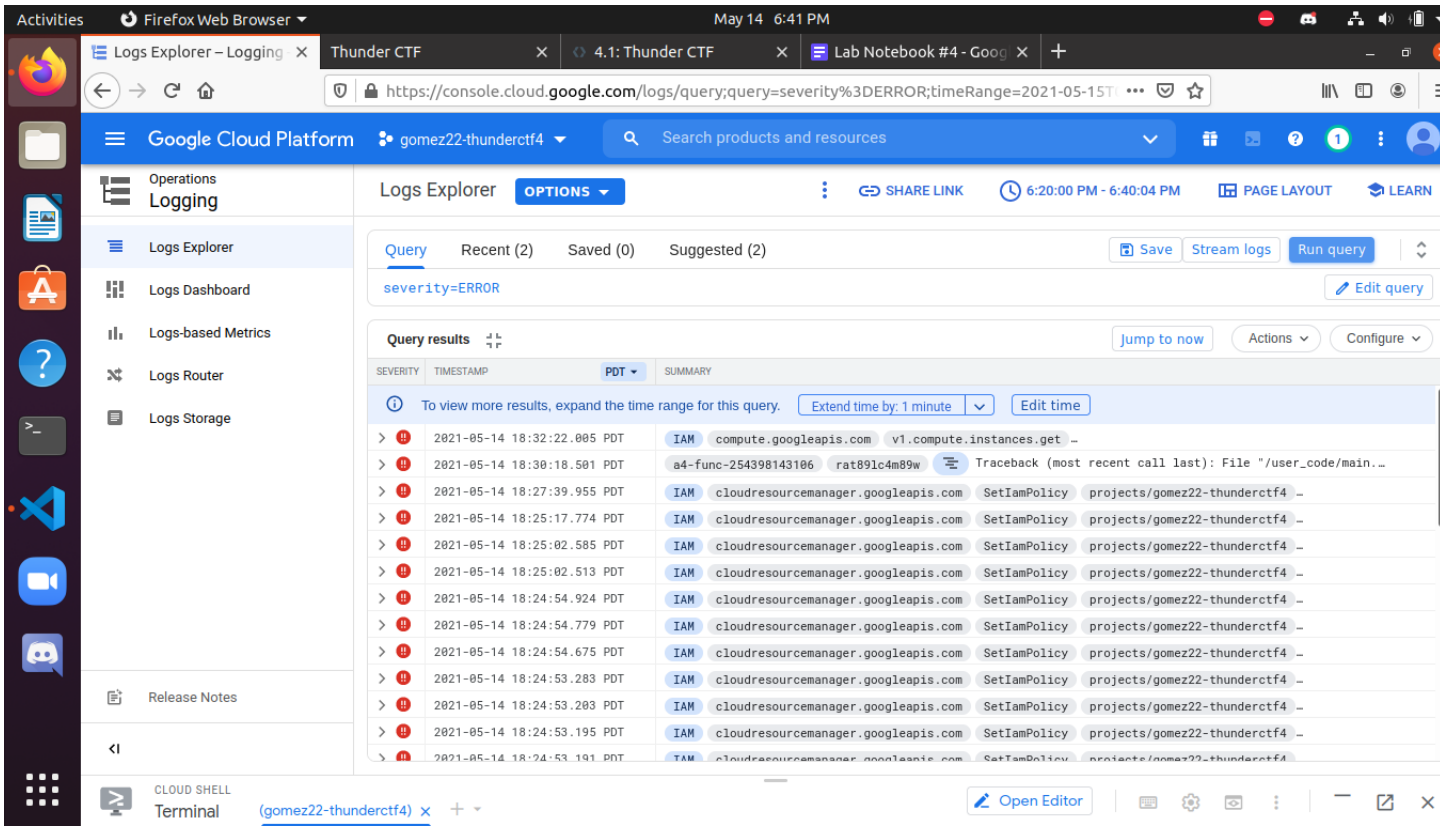
Take a screenshot of the secret obtained



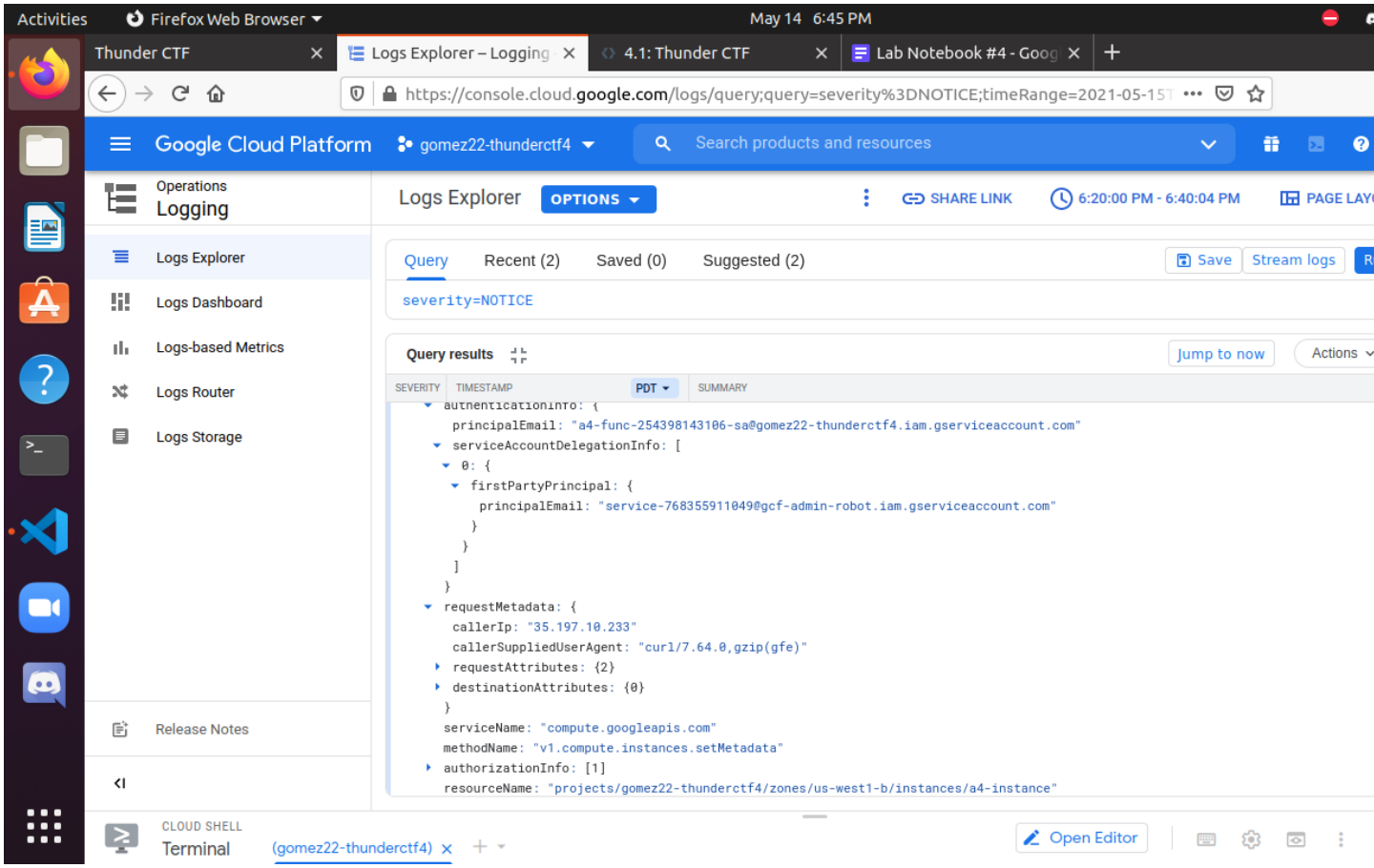
Take a screenshot showing the events at this severity


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=2021-05-15
Google Cloud Platform
gomez22-thunderctf4
Search products and resources
Operations
Logging
Logs Explorer
Logs Dashboard
Logs-based Metrics
Logs Router
Logs Storage
Release Notes
Logs Explorer
OPTIONS
SHARE LINK
6:20:00 PM - 6:40:04 PM
PAGE LAYOUT
Query
Recent (1)
Saved (0)
Suggested (2)
Save
Stream logs
Run query
severity=NOTICE
Query results
Jump to now
Actions
SEVERITY
TIMESTAMP
PDT
SUMMARY
To view more results, expand the time range for this query.
Extend time by: 1 minute
Edit time
2021-05-14 18:36:48.395 PDT IAM compute.googleapis.com v1.compute.instances.setMetadata …
2021-05-14 18:36:45.325 PDT IAM compute.googleapis.com v1.compute.instances.setMetadata …
2021-05-14 18:30:19.379 PDT a4-func-254398143106 Error detected in a4-func-254398143106
2021-05-14 18:26:56.649 PDT IAM compute.googleapis.com v1.compute.instances.setMetadata …
2021-05-14 18:26:54.149 PDT IAM iam.googleapis.com google.iam.admin.v1.CreateServiceAccountKey …
2021-05-14 18:26:53.500 PDT IAM compute.googleapis.com v1.compute.instances.setMetadata …
2021-05-14 18:26:50.755 PDT IAM deploymentmanager.googleapis.com v2.deploymentmanager.deployments.insert …
2021-05-14 18:26:49.820 PDT IAM cloudfunctions.googleapis.com google.cloud.functions.v1.CloudFunctionsService.SetIamPolicy
2021-05-14 18:26:44.623 PDT IAM cloudfunctions.googleapis.com …
2021-05-14 18:26:19.868 PDT IAM cloudbuild.googleapis.com google.devtools.cloudbuild.v1.CloudBuild.CreateBuild …
2021-05-14 18:25:56.761 PDT IAM cloudbuild.googleapis.com google.devtools.cloudbuild.v1.CloudBuild.CreateBuild …
2021-05-14 18:25:55.838 PDT IAM cloudfunctions.googleapis.com …
CLOUD SHELL
Terminal
(gomez22-thunderctf4)
Open Editor

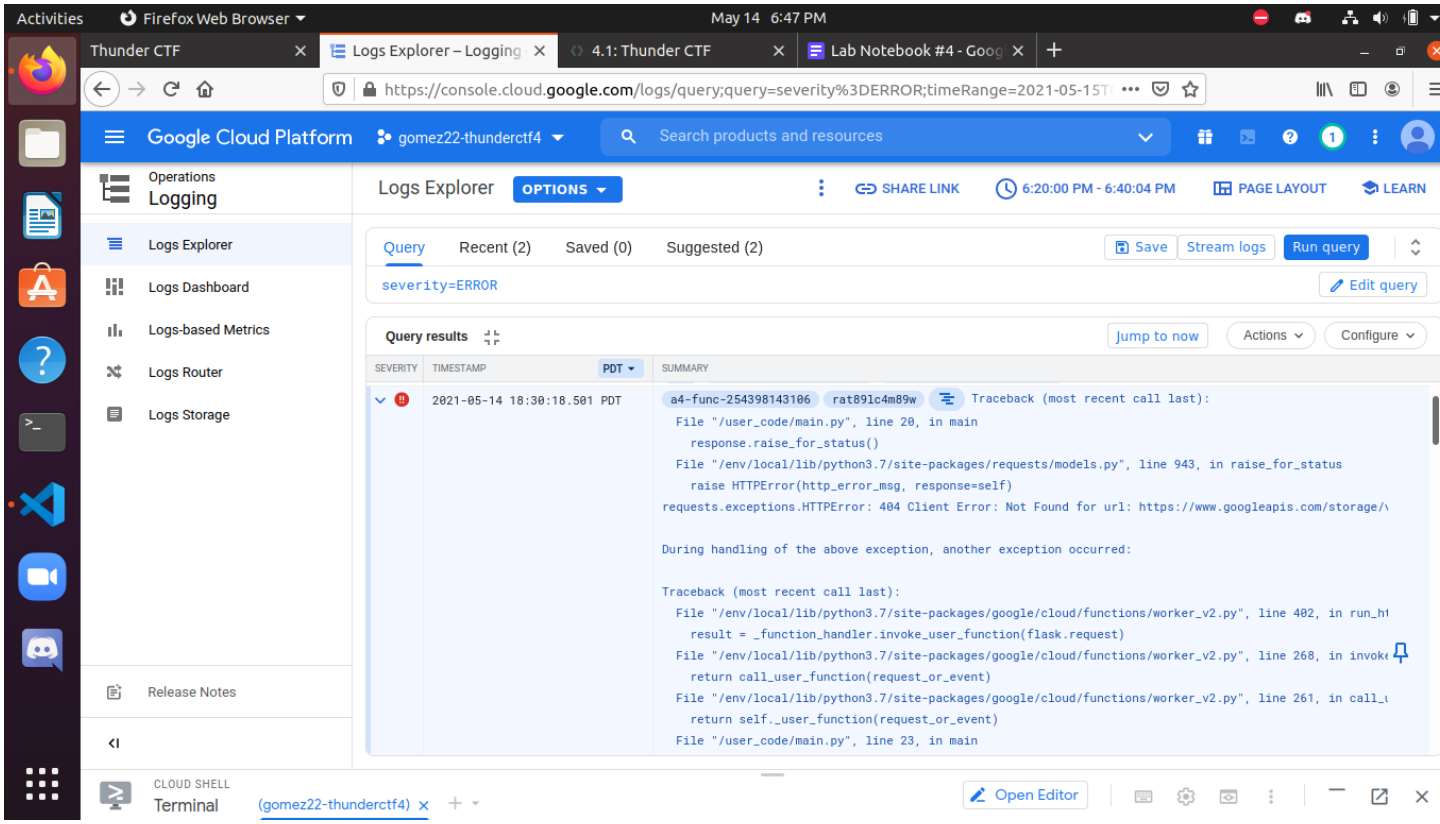
Take a screenshot showing the events at this severity

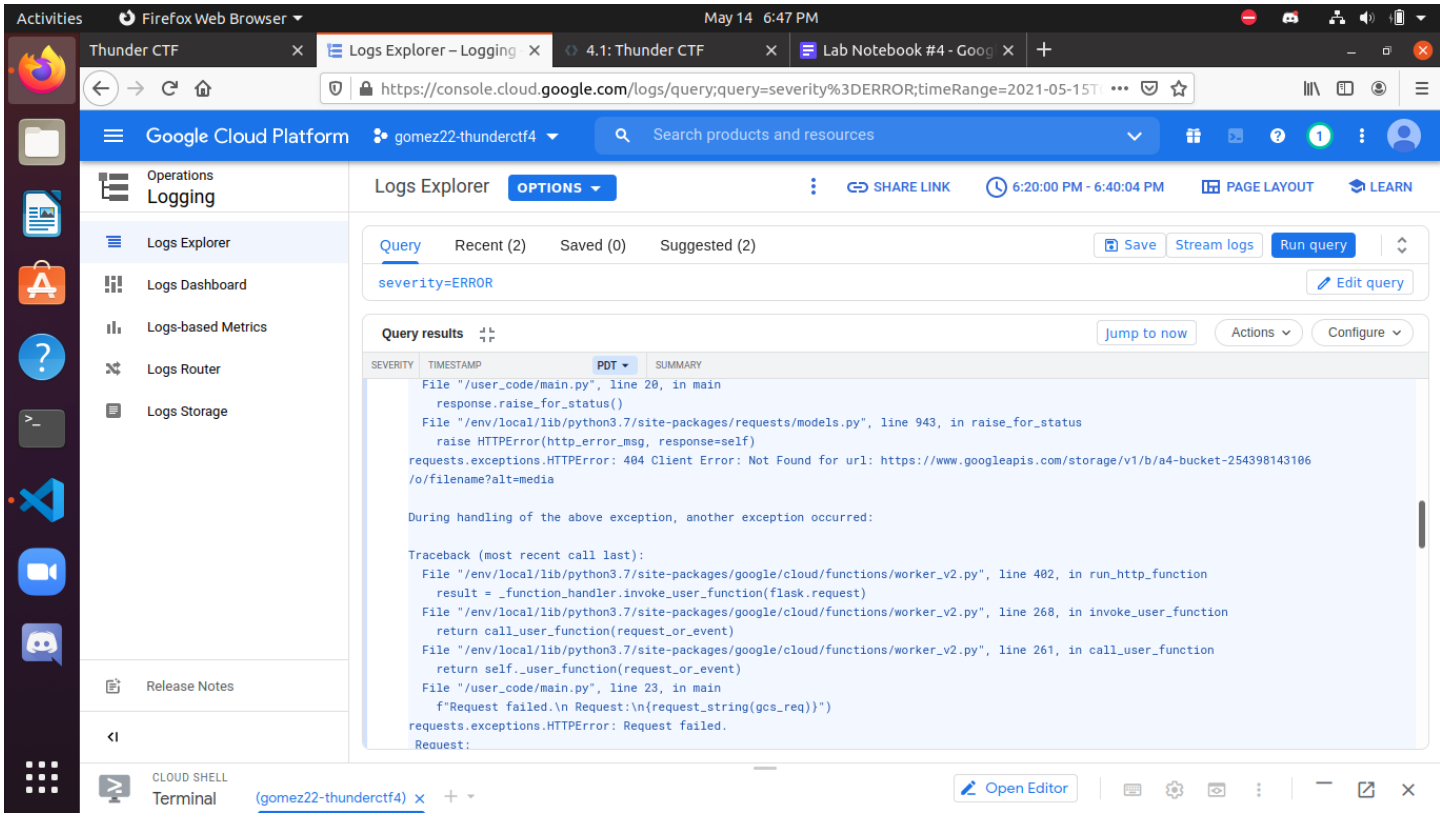


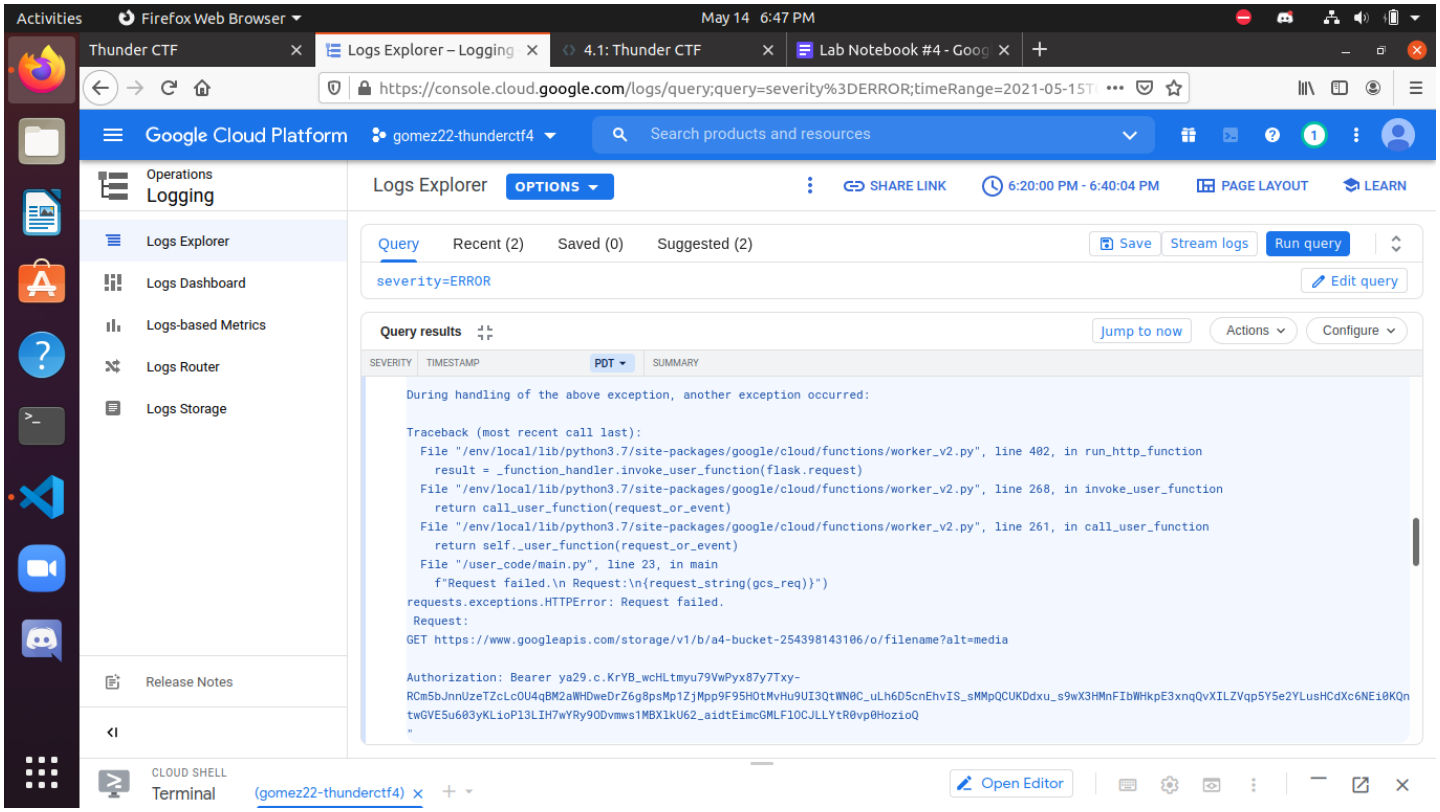
Take a screenshot showing the name of the service account that has been used to perform this operation as well as the IP address of the client and the User-Agent of the request that has performed the operation.



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.

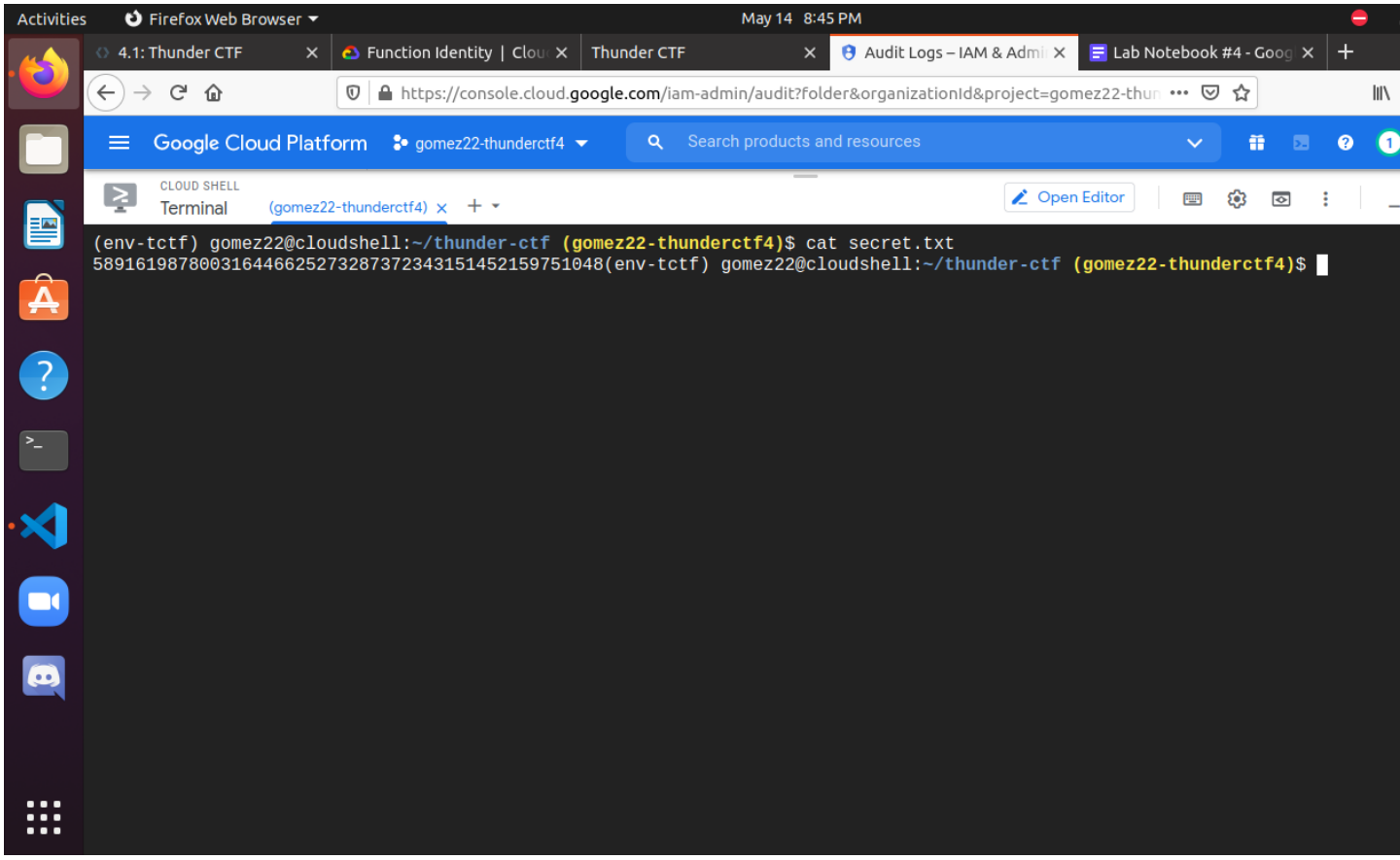

Activities
Firefox Web Browser
May 14 6:47 PM
Thunder CTF
Logs Explorer – Logging
4.1: Thunder CTF
Lab Notebook #4 - Goog
https://console.cloud.google.com/logs/query;query=severity%3DERROR;timeRange=2021-05-15T
Google Cloud Platform
gomez22-thunderctf4
Search products and resources
Operations
Logging
Logs Explorer
Logs Dashboard
Logs-based Metrics
Logs Router
Logs Storage
Release Notes
Logs Explorer
OPTIONS
SHARE LINK
6:20:00 PM - 6:40:04 PM
PAGE LAYOUT
LEARN
Query
Recent (2)
Saved (0)
Suggested (2)
Save
Stream logs
Run query
severity=ERROR
Edit query
Query results
Jump to now
Actions
Configure
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/
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_ht
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 invok
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_
return self._user_function(request_or_event)
File "/user_code/main.py", line 23, in main
CLOUD SHELL
Terminal
(gomez22-thunderctf4)
Open Editor


Activities
Firefox Web Browser
May 14 6:47 PM
Thunder CTF
Logs Explorer – Logging
4.1: Thunder CTF
Lab Notebook #4 - Goog
https://console.cloud.google.com/logs/query;query=severity%3DERROR;timeRange=2021-05-15T
Google Cloud Platform
gomez22-thunderctf4
Search products and resources
Operations
Logging
Logs Explorer
Logs Dashboard
Logs-based Metrics
Logs Router
Logs Storage
Release Notes
Logs Explorer
OPTIONS
SHARE LINK
6:20:00 PM - 6:40:04 PM
PAGE LAYOUT
LEARN
Query
Recent (2)
Saved (0)
Suggested (2)
Save
Stream logs
Run query
severity=ERROR
Edit query
Query results
Jump to now
Actions
Configure
SEVERITY
TIMESTAMP
PDT
SUMMARY
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:\n{request_string(gcs_req)}")
requests.exceptions.HTTPError: Request failed.
Request:
CLOUD SHELL
Terminal
(gomez22-thunderctf4)
Open Editor

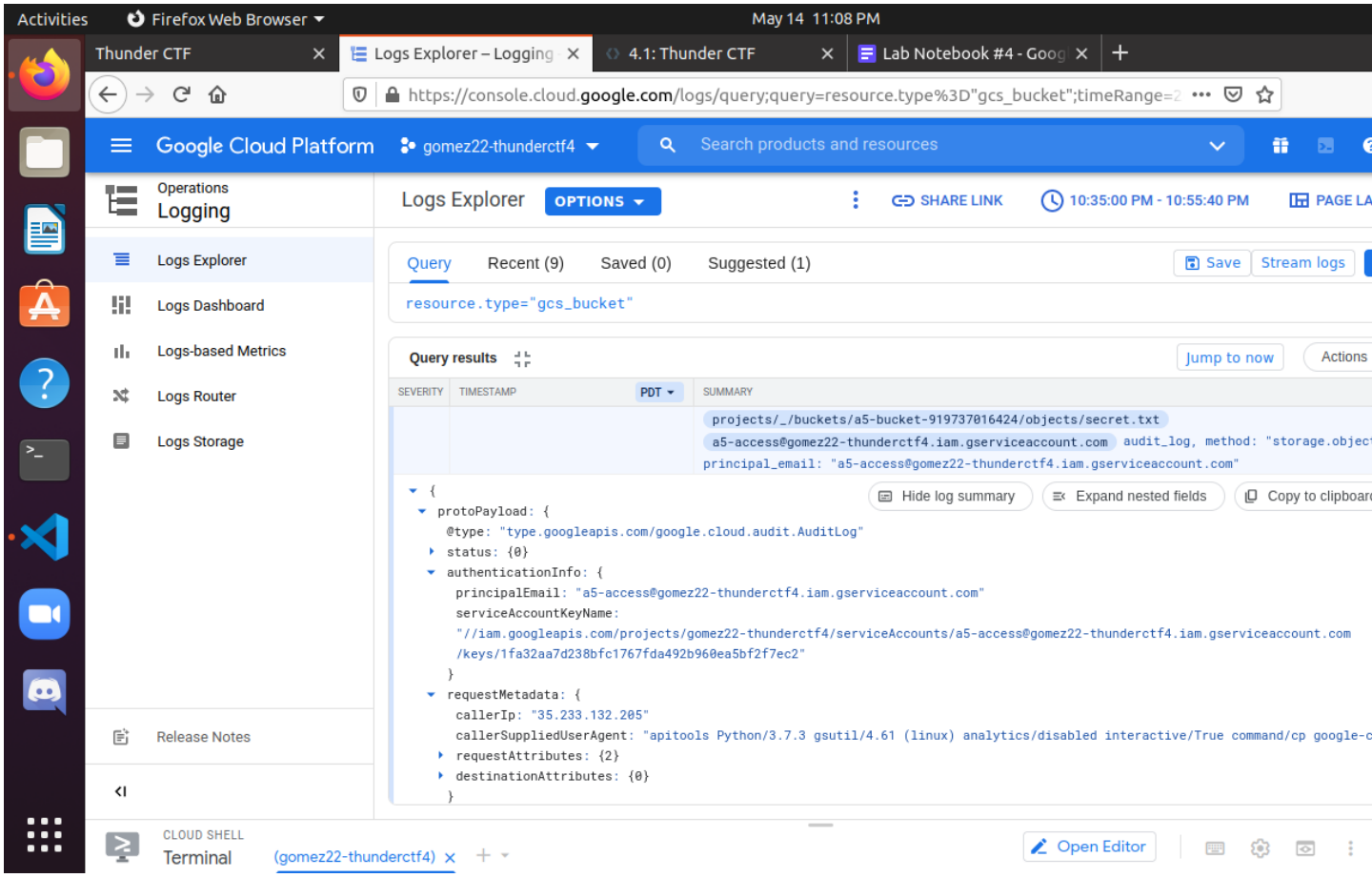


A5power

Take a screenshot of the secret obtained



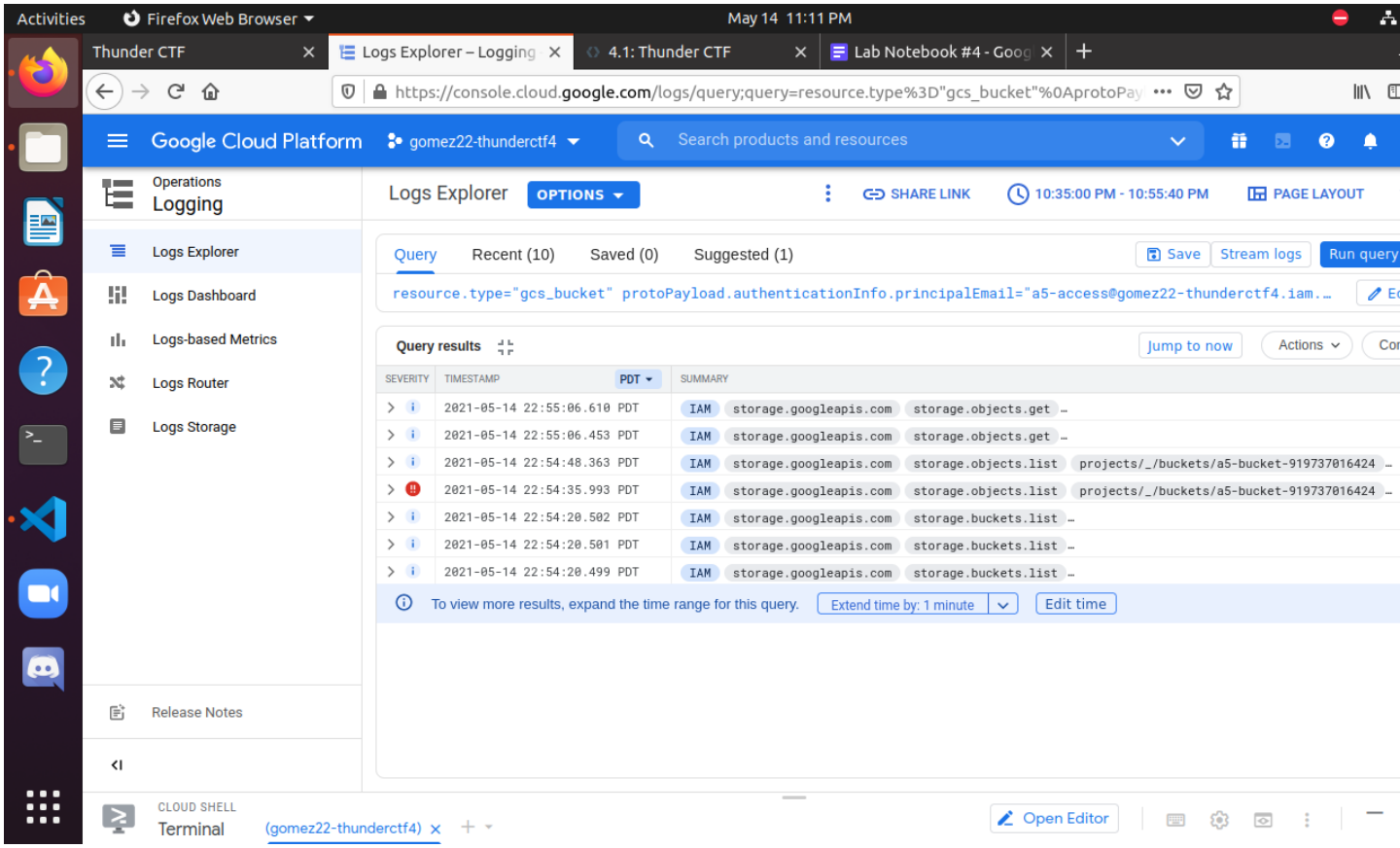
Take a screenshot of the entry that includes the service account used to access the bucket.



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.



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.

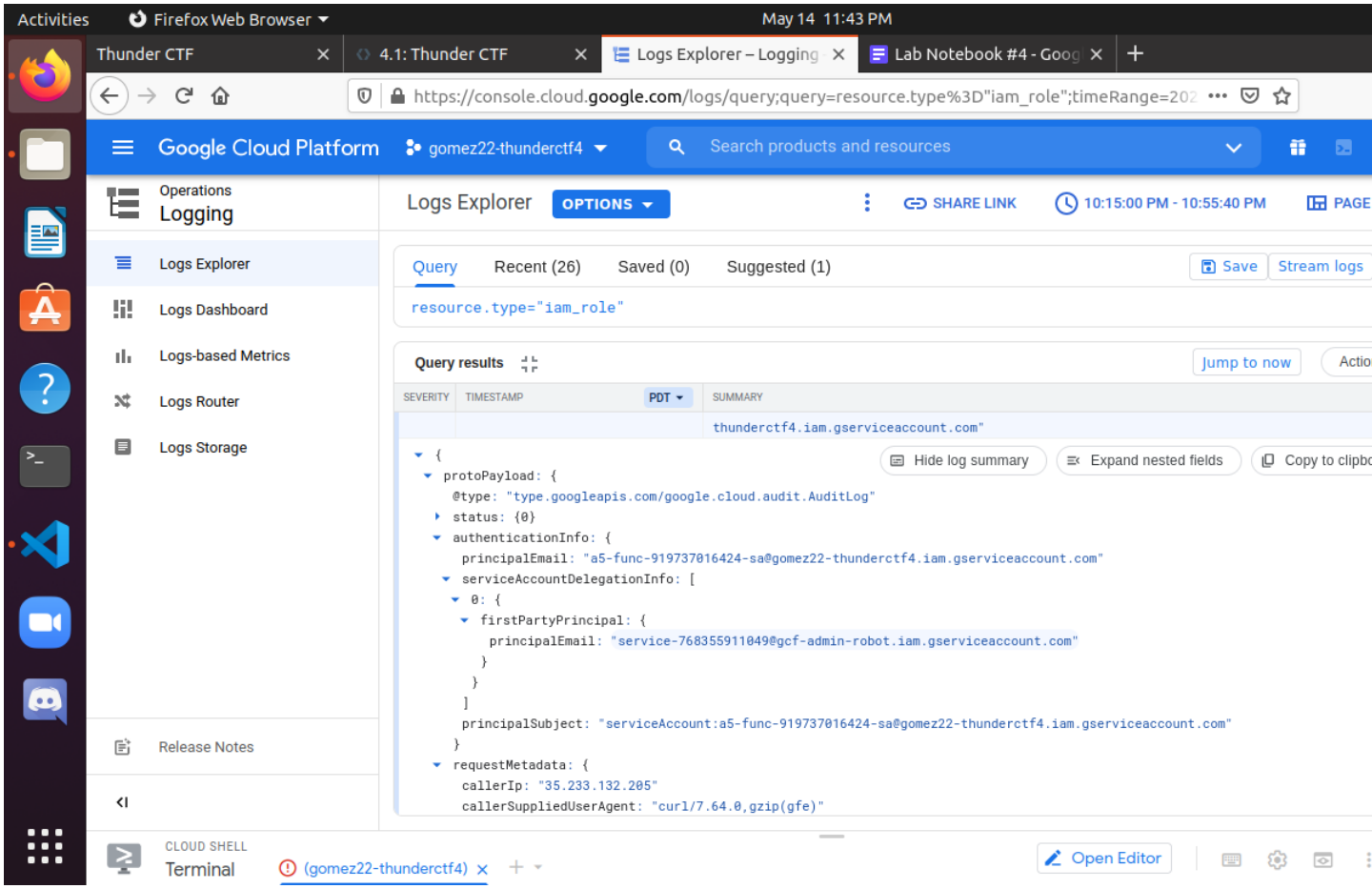
What IP address did the request originate from? What UserAgent was used?

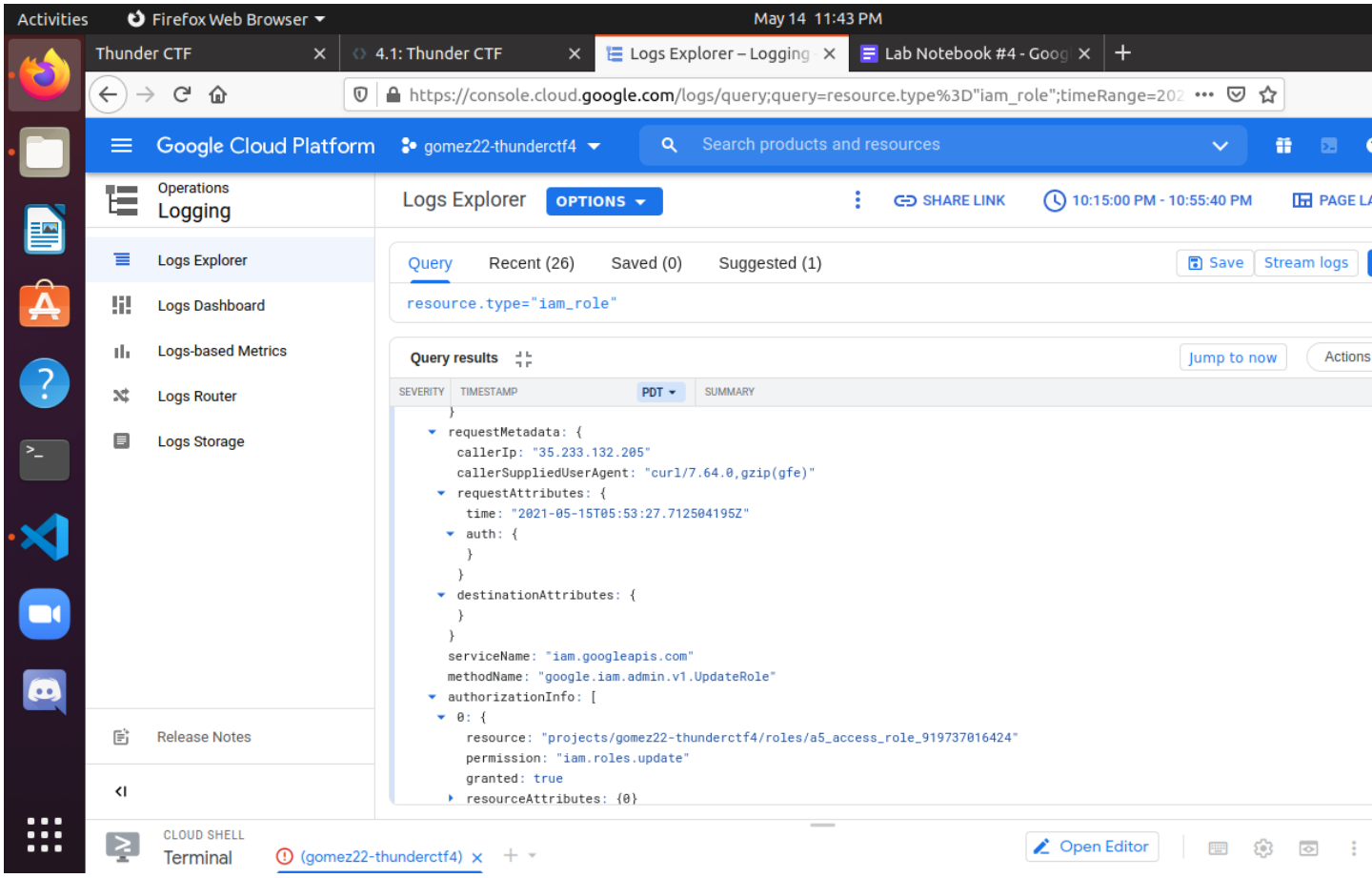
-The IP address is “35.233.132.205". The user agent is "google-cloud-sdk gcloud/340.0.0 command/gcloud.functions.deploy invocation-id/63c4e544d5bb43d09df699b54eaa001b environment/devshell environment-version/None interactive/True from-script/False python/3.7.3 term/screen (Linux 5.4.104+),gzip(gfe),gzip(gfe)".

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.

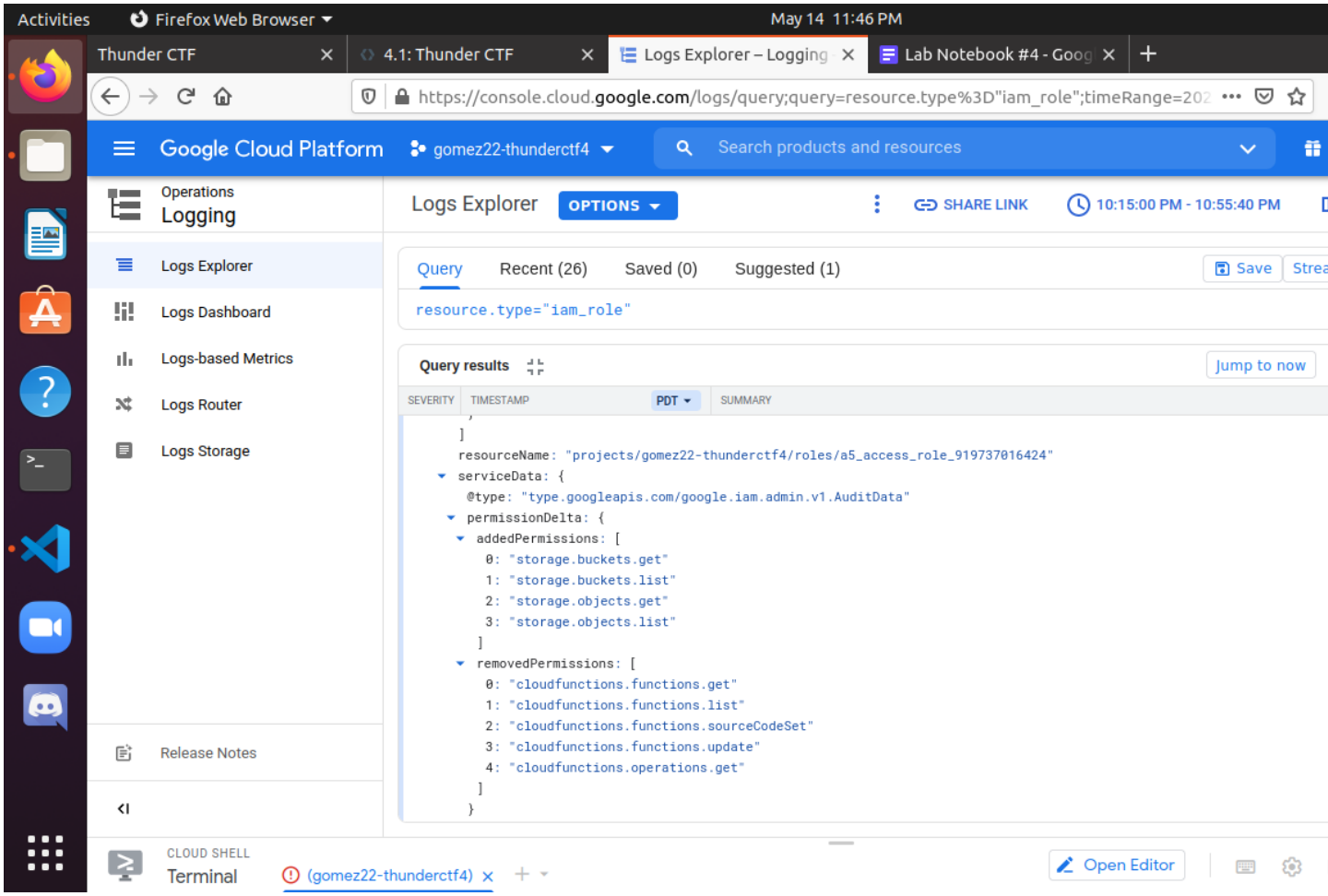




What evidence suggests that this request did not come from the Cloud Function itself?

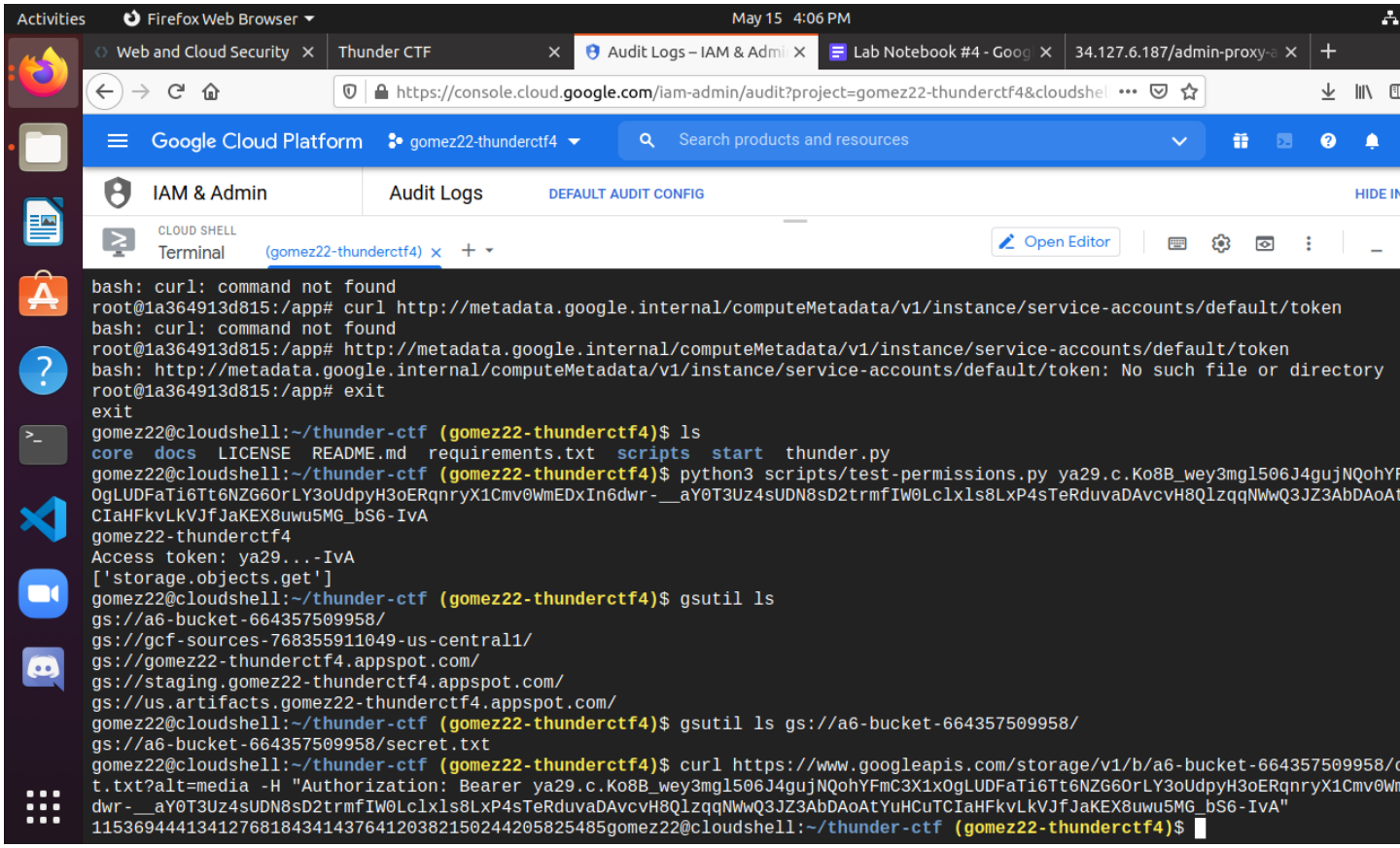
-The principalEmail has the "<function_name>@gomez22-thunderctf4.iam.gserviceaccount.com", so we can see that the function was called by a user. Also, from what I've seen in the programmatically generated logs "curl" isn't used as a user agent so this would be considered abnormal behavior and not likely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

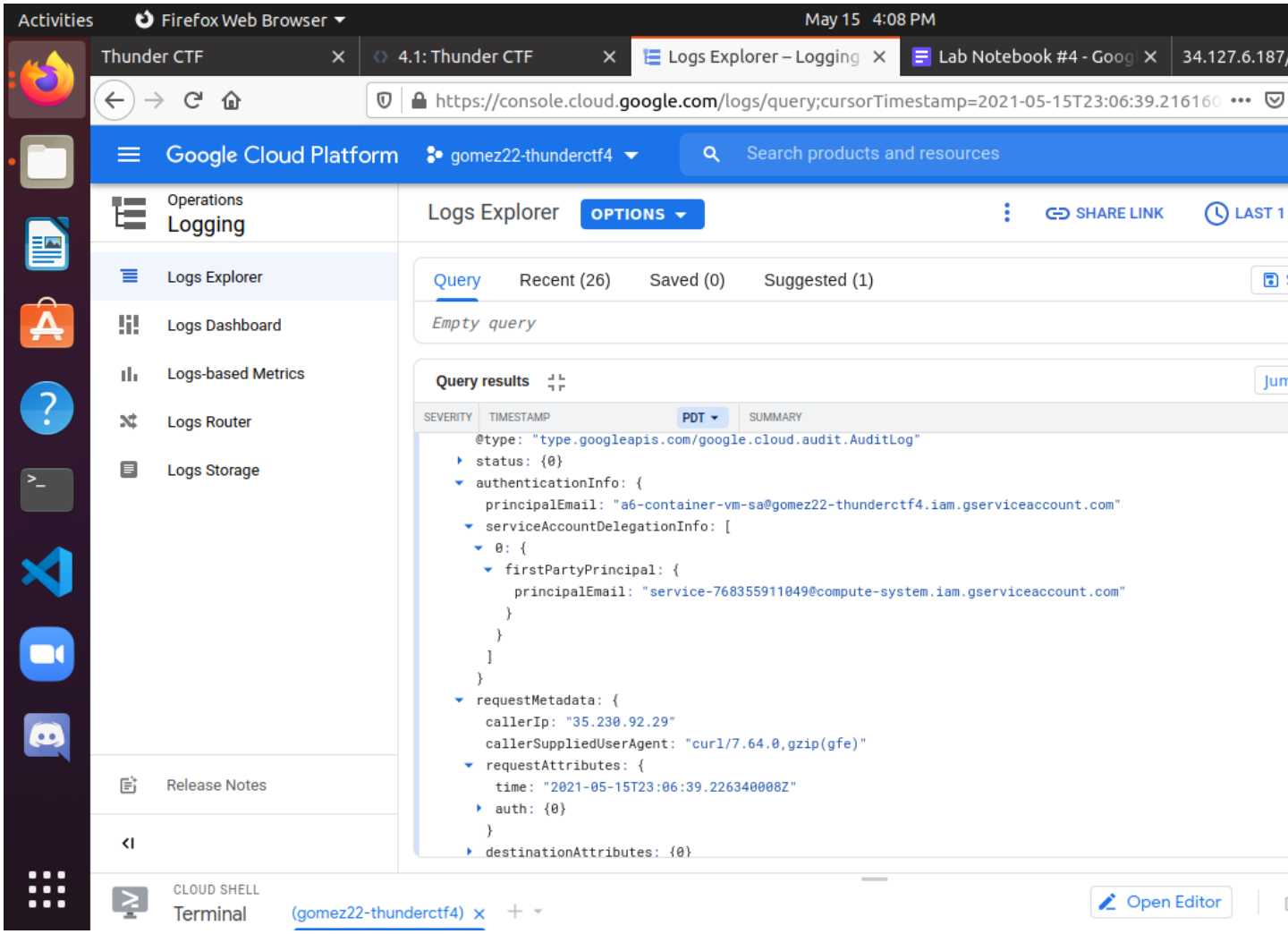


A6container

Take a screenshot of the secret obtained



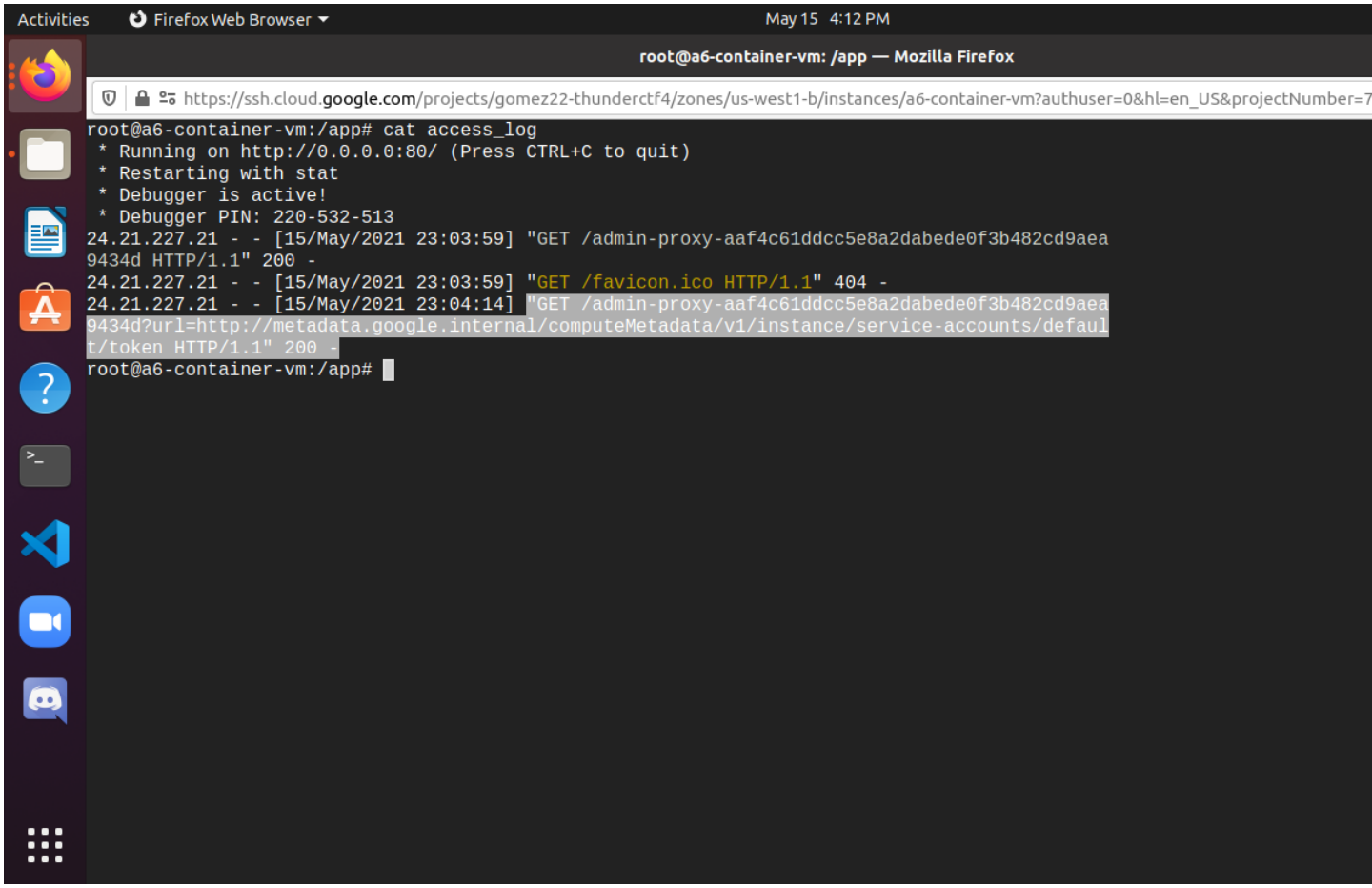
Take a screenshot of the entry that includes the service account used to access the bucket along with the requestMetadata.



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.

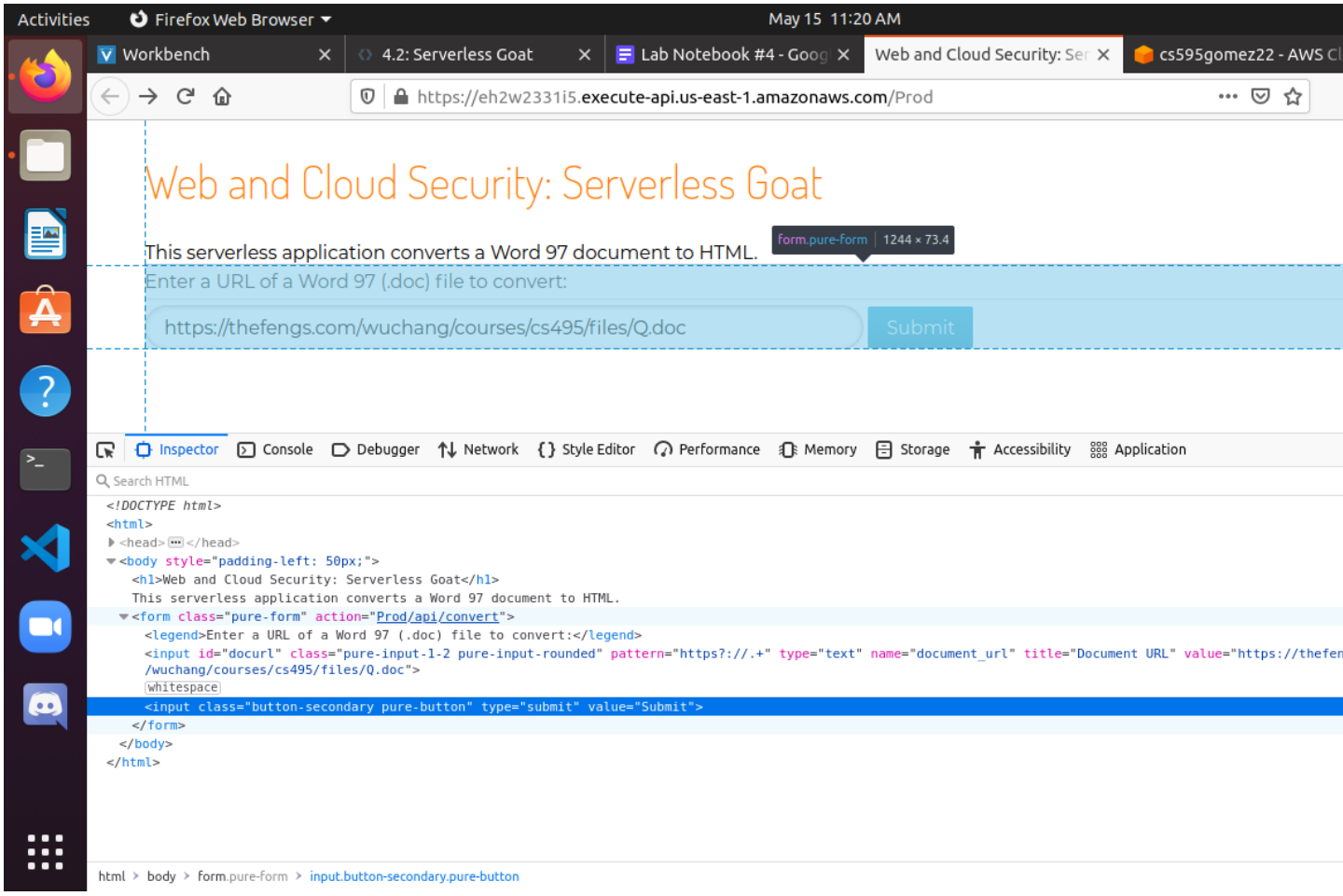


4.2 Serverless Goat

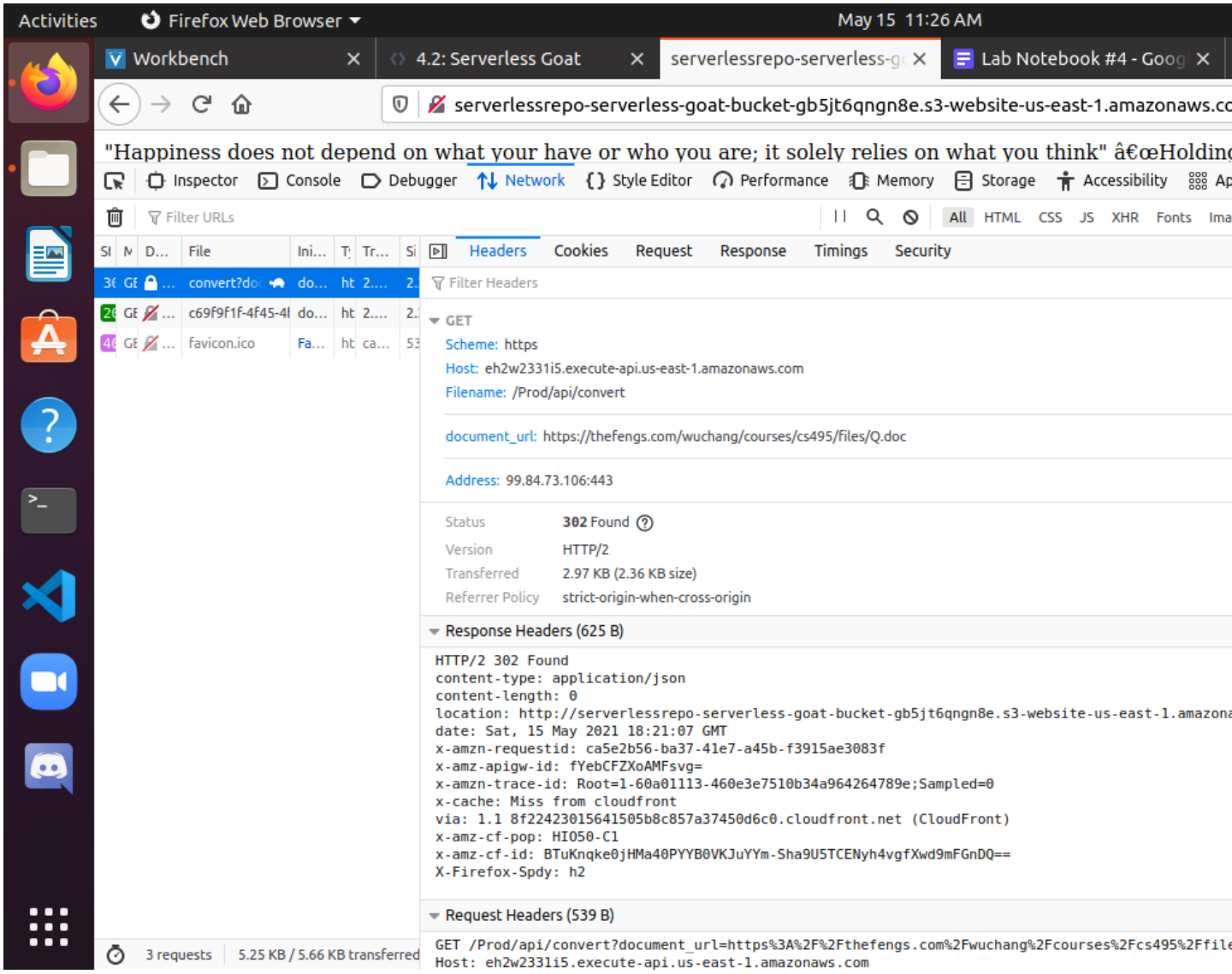
The endpoint exposes the region it is being run in. What region does it reside in?

-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



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)?

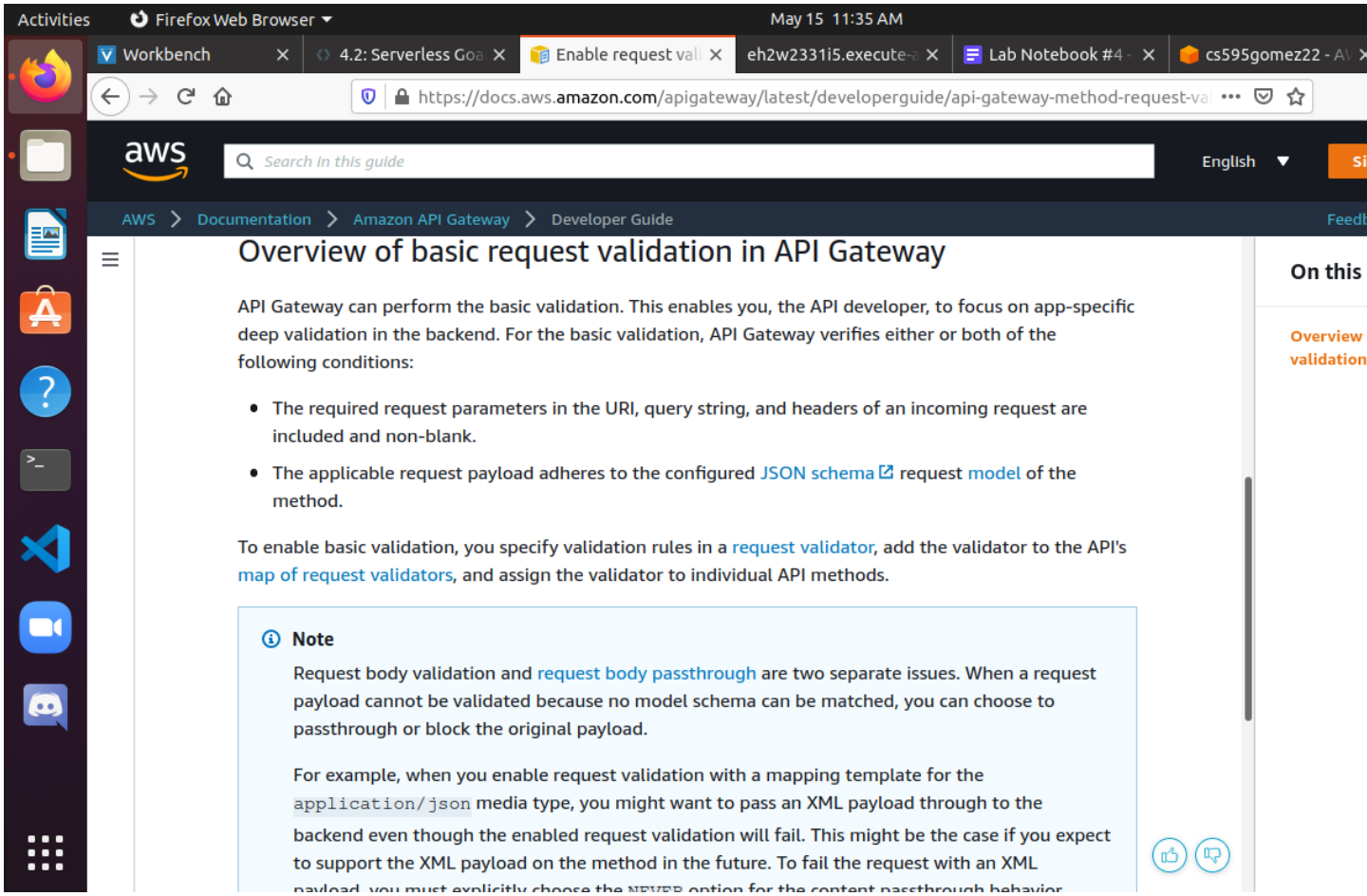
-Below if the error I received, It looks as though the error happens in line 9 of index.js

“TypeError: Cannot read property 'document_url' of null

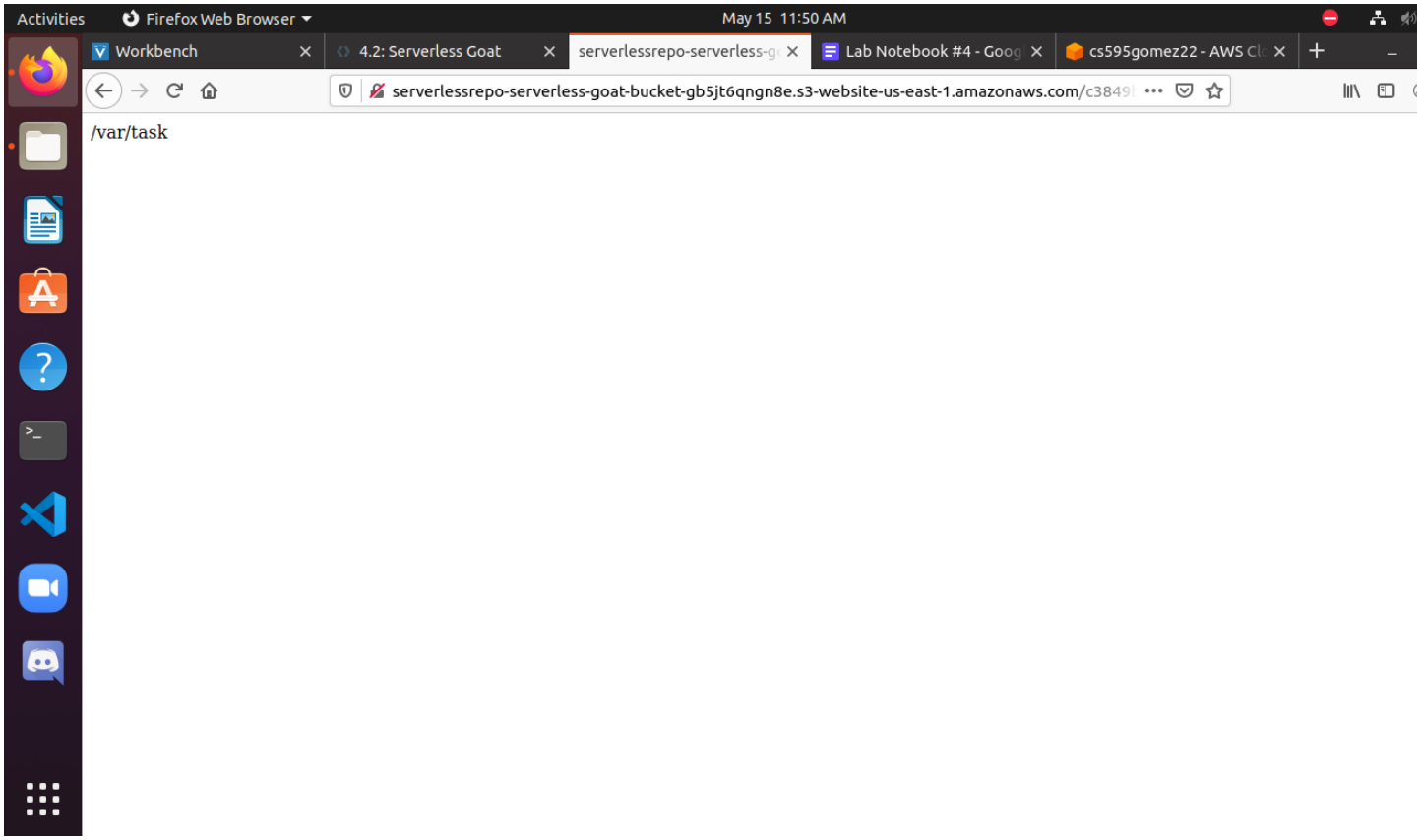
    at log (/var/task/index.js:9:49)

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

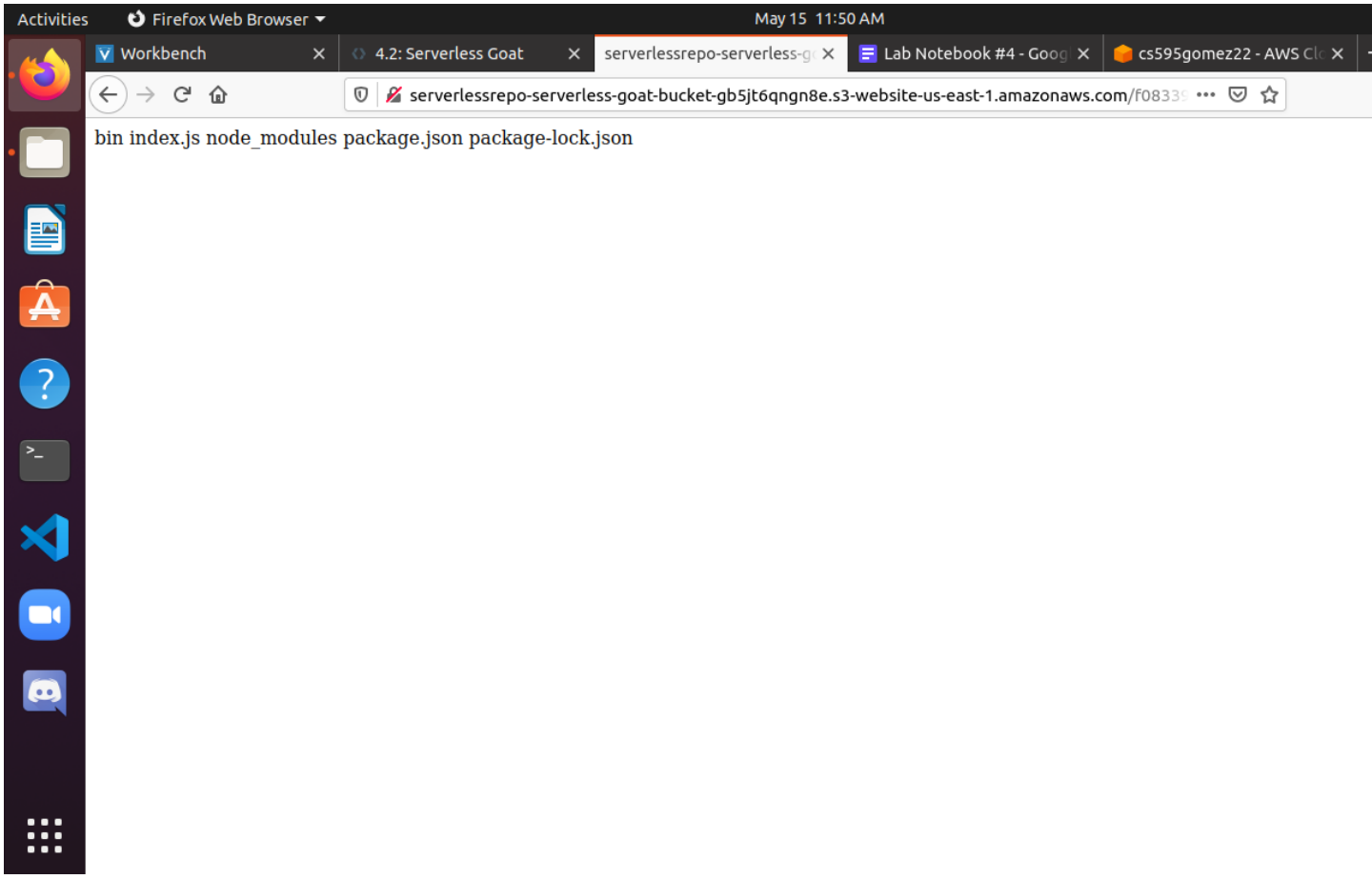
Visit the AWS API Gateway Developer Guide and examine the topics in the "Develop" section of "Working with REST APIs". Find a feature that can be enabled to help this serverless application validate its input. Take a screenshot of it.



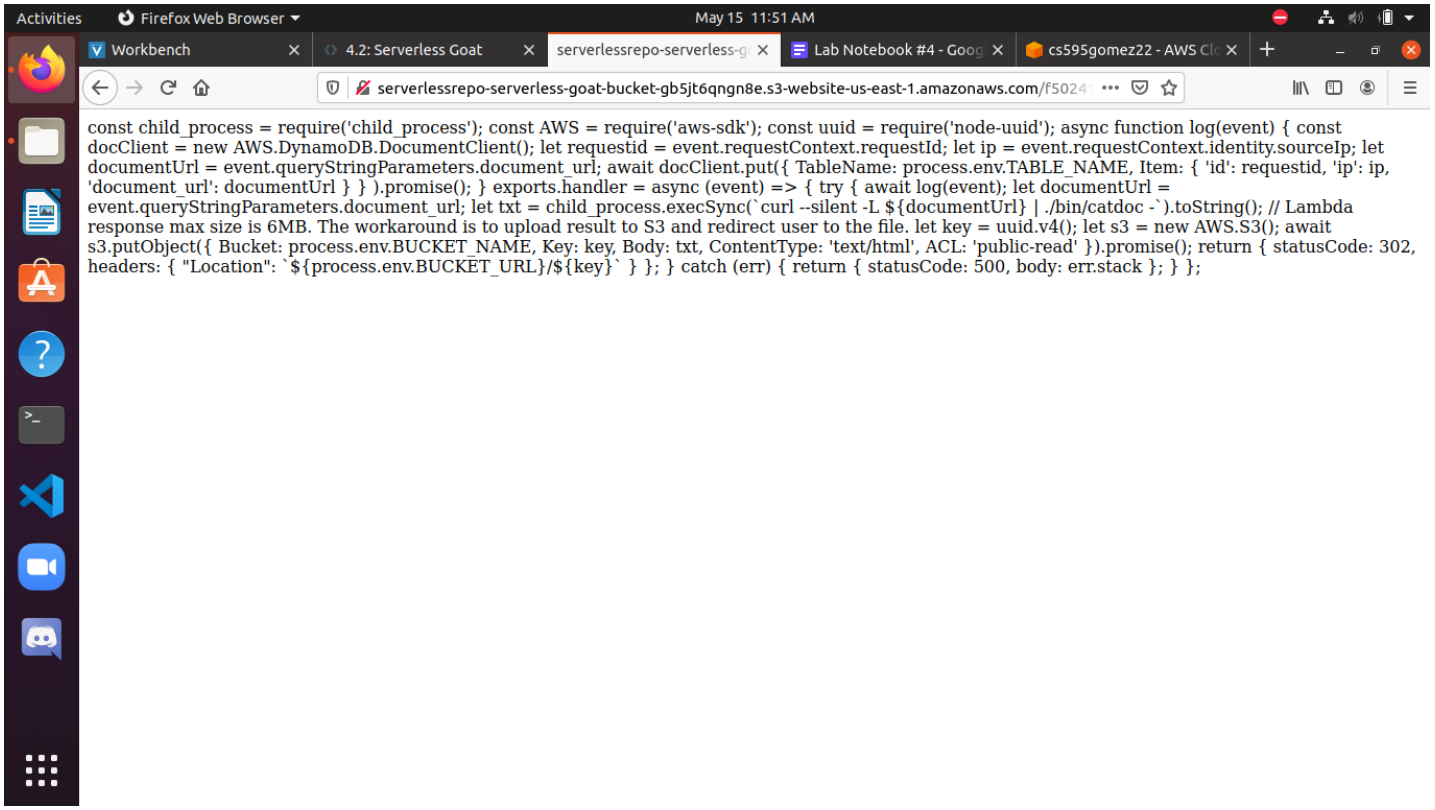
Use command injection to obtain the working directory the application is run in. Show a screenshot of the result at the end of the page returned.



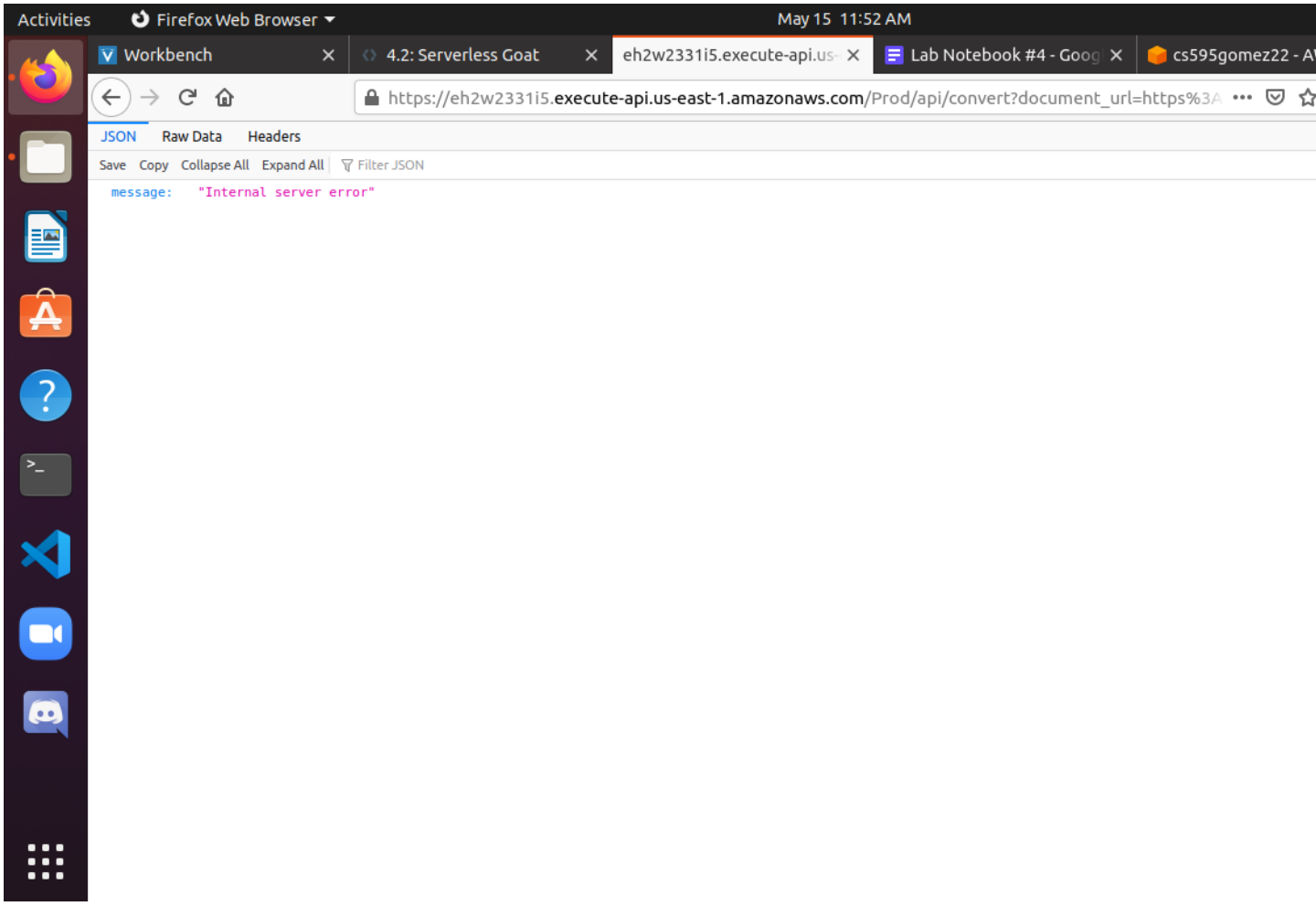
Then use command injection to obtain a listing of the directory and show the files that are there.



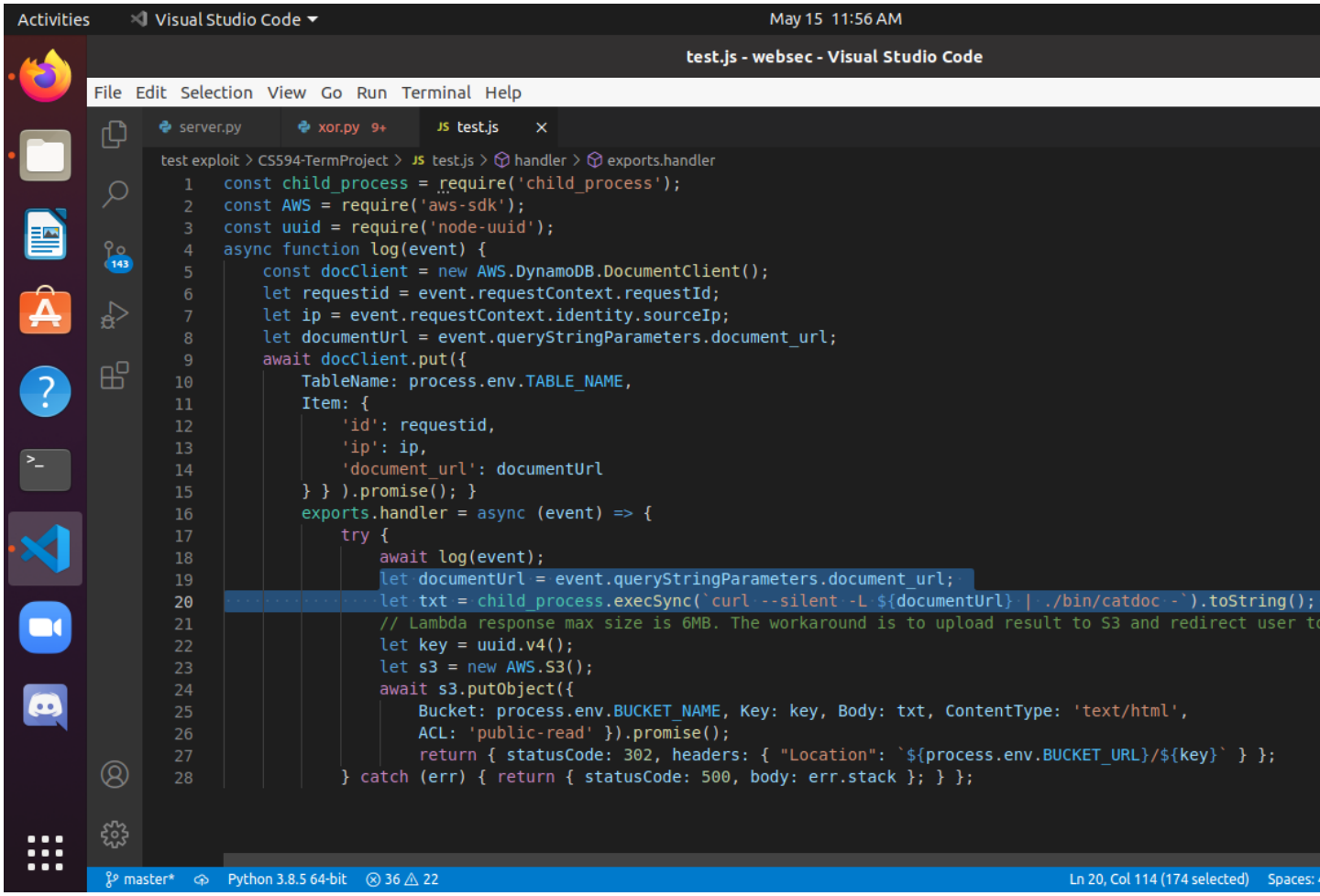
Use command injection to dump out the source file that implements this Lambda function.



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.

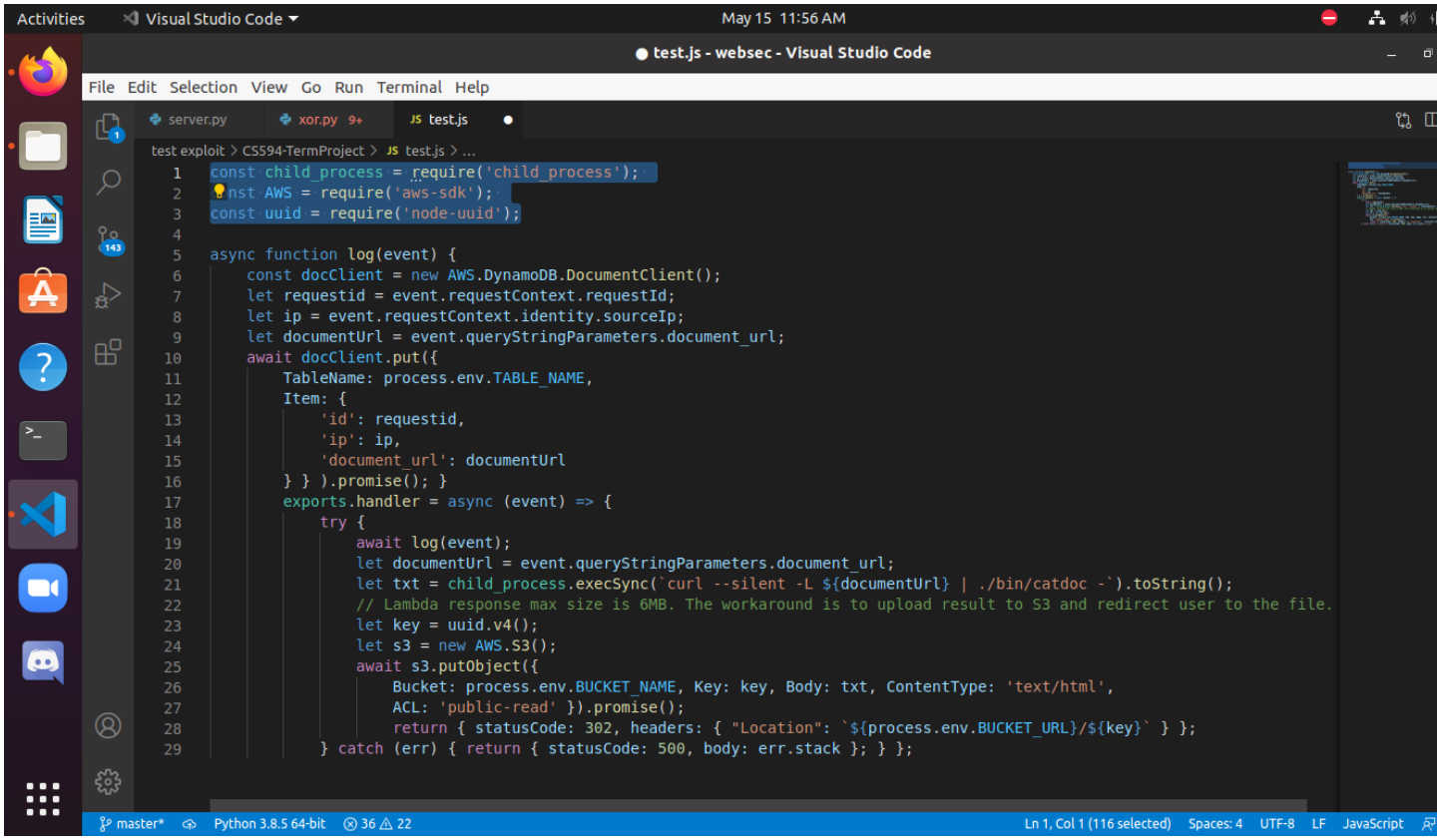


Show the packages that this file requires. How is each package used in this code?

-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.



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.

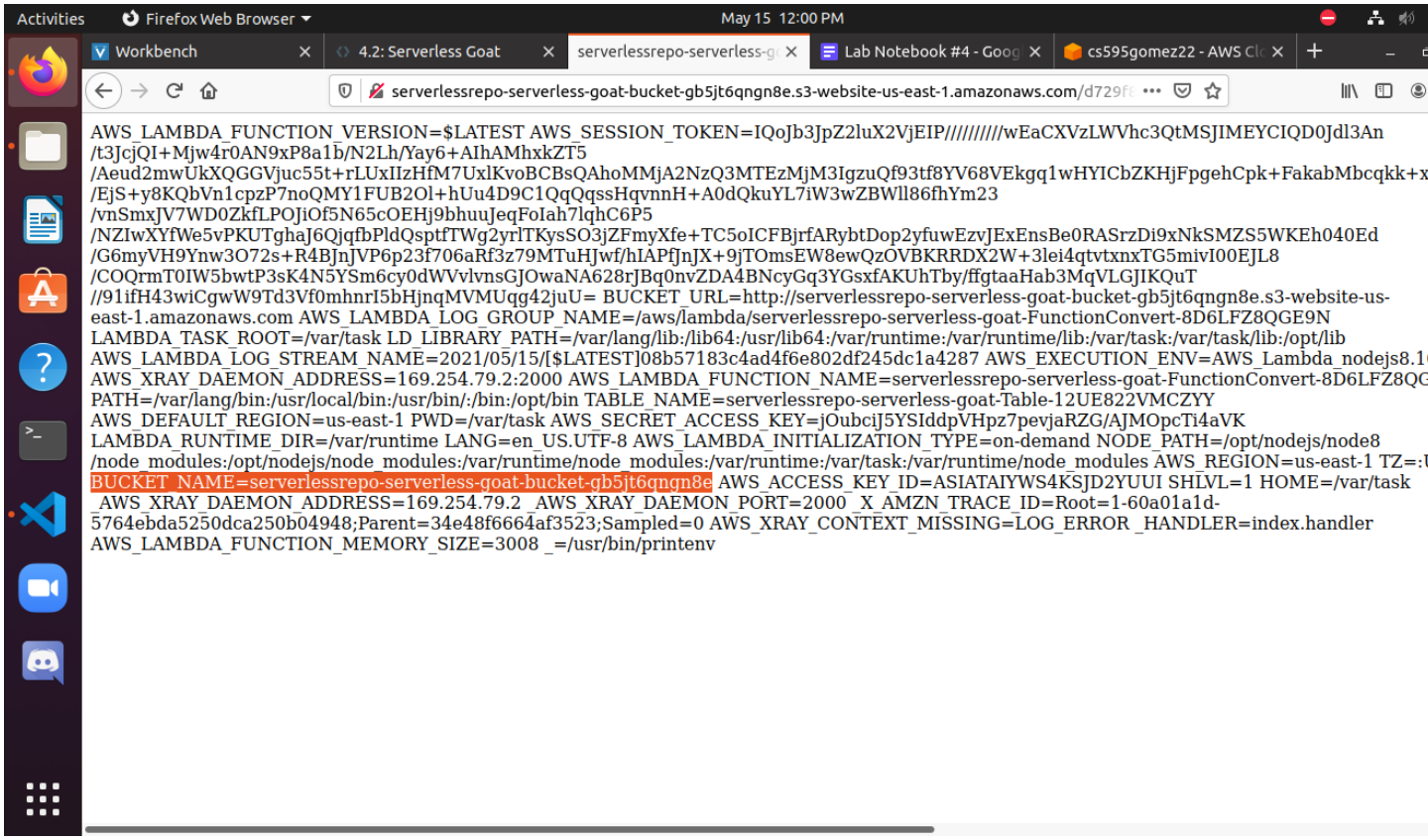
Use command injection to dump the contents of the package manifest file for the application. What version of packages does the source file depend upon? Look up this package and version to determine how old the package is? Find any known vulnerabilities in this package.

-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”.

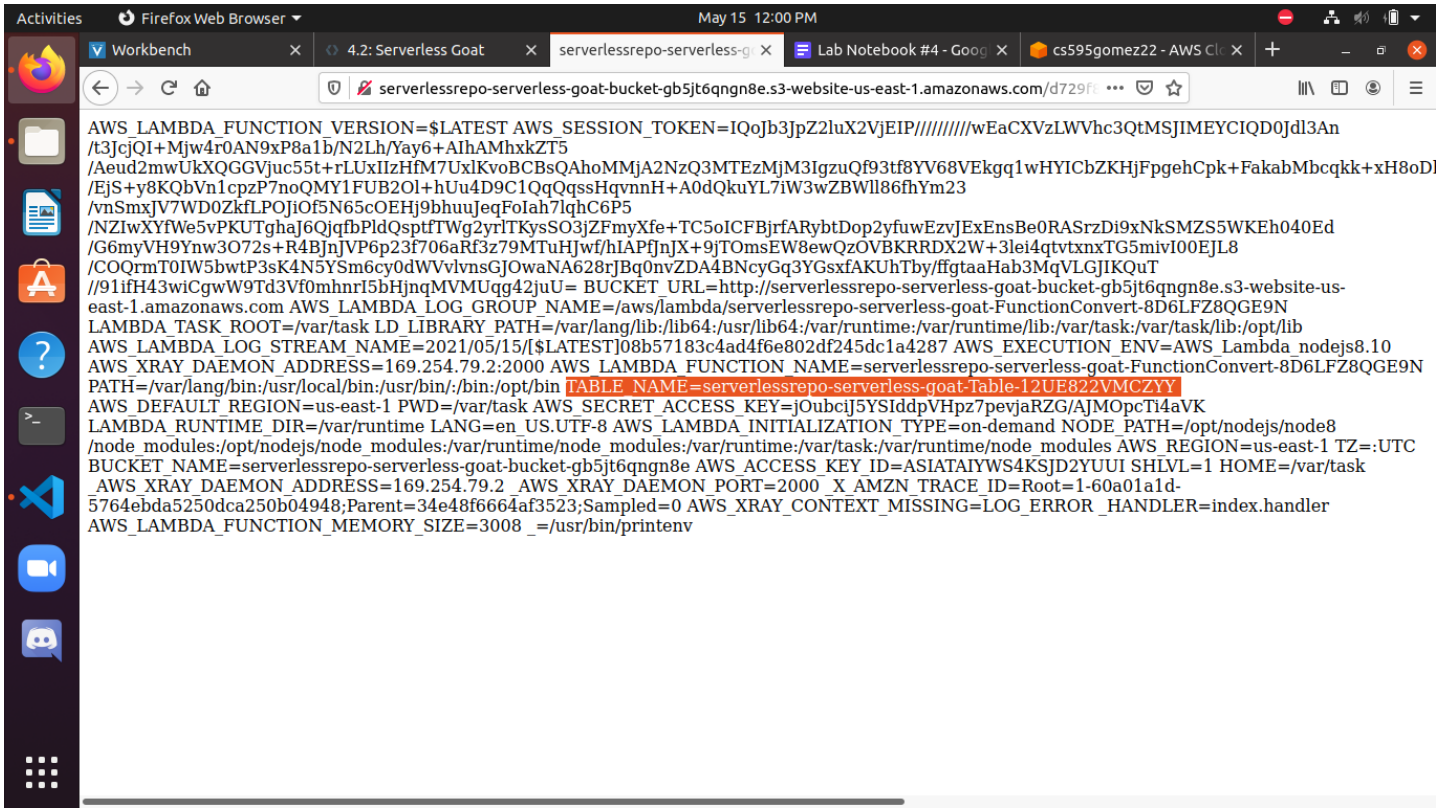
How does the application use this package in its operation? What would be the impact of a vulnerability in this package (if any)?

-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.

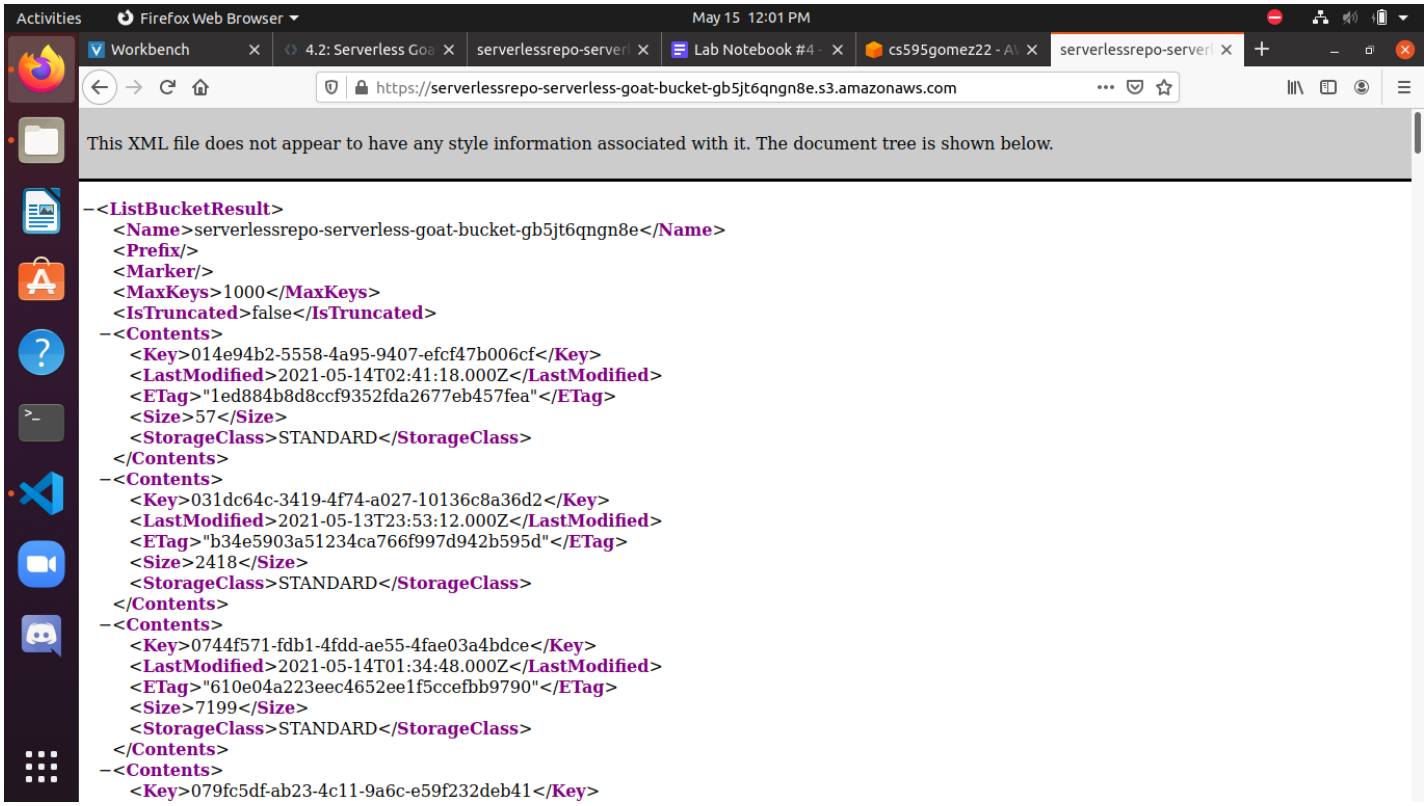
Show the variable that stores the bucket name in a screenshot



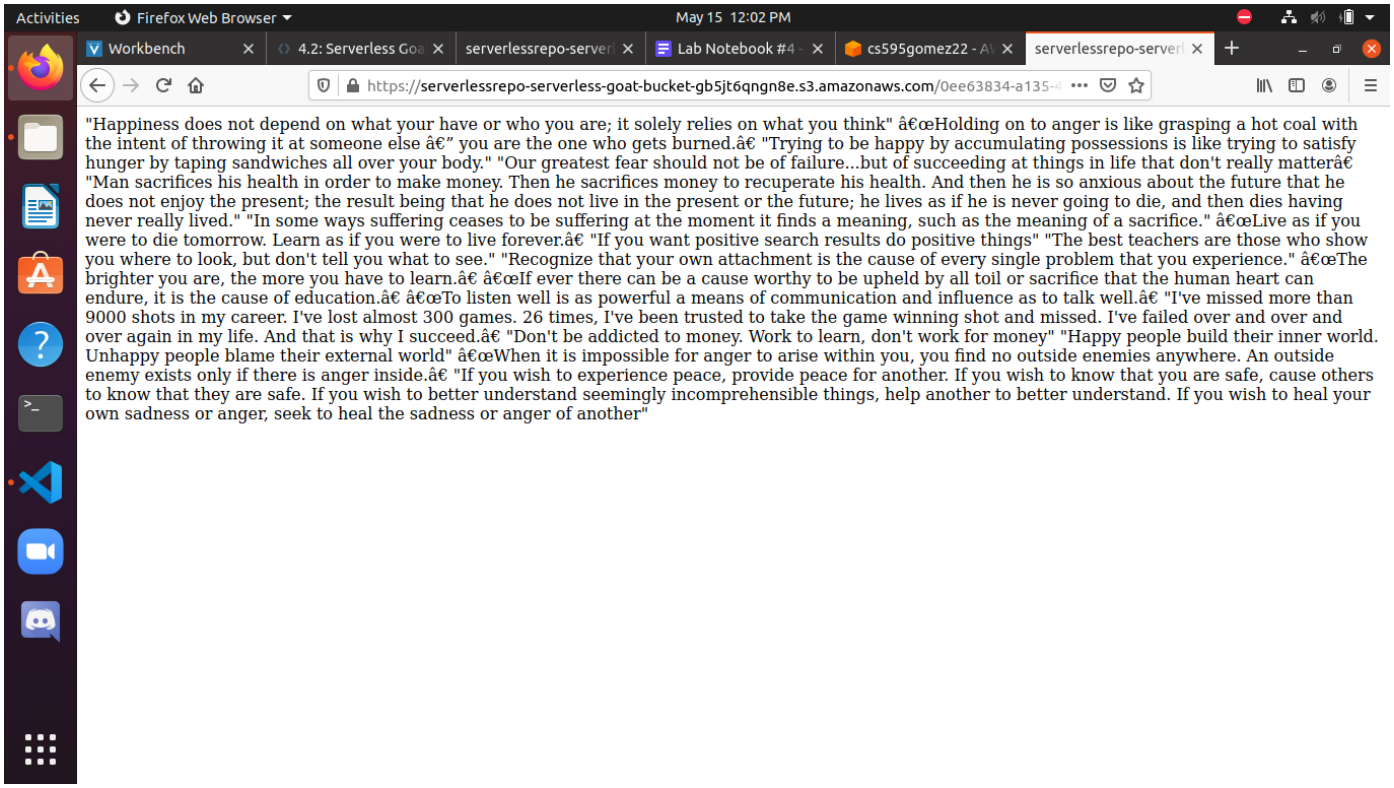
Show the table name that is used to store activity information from the application



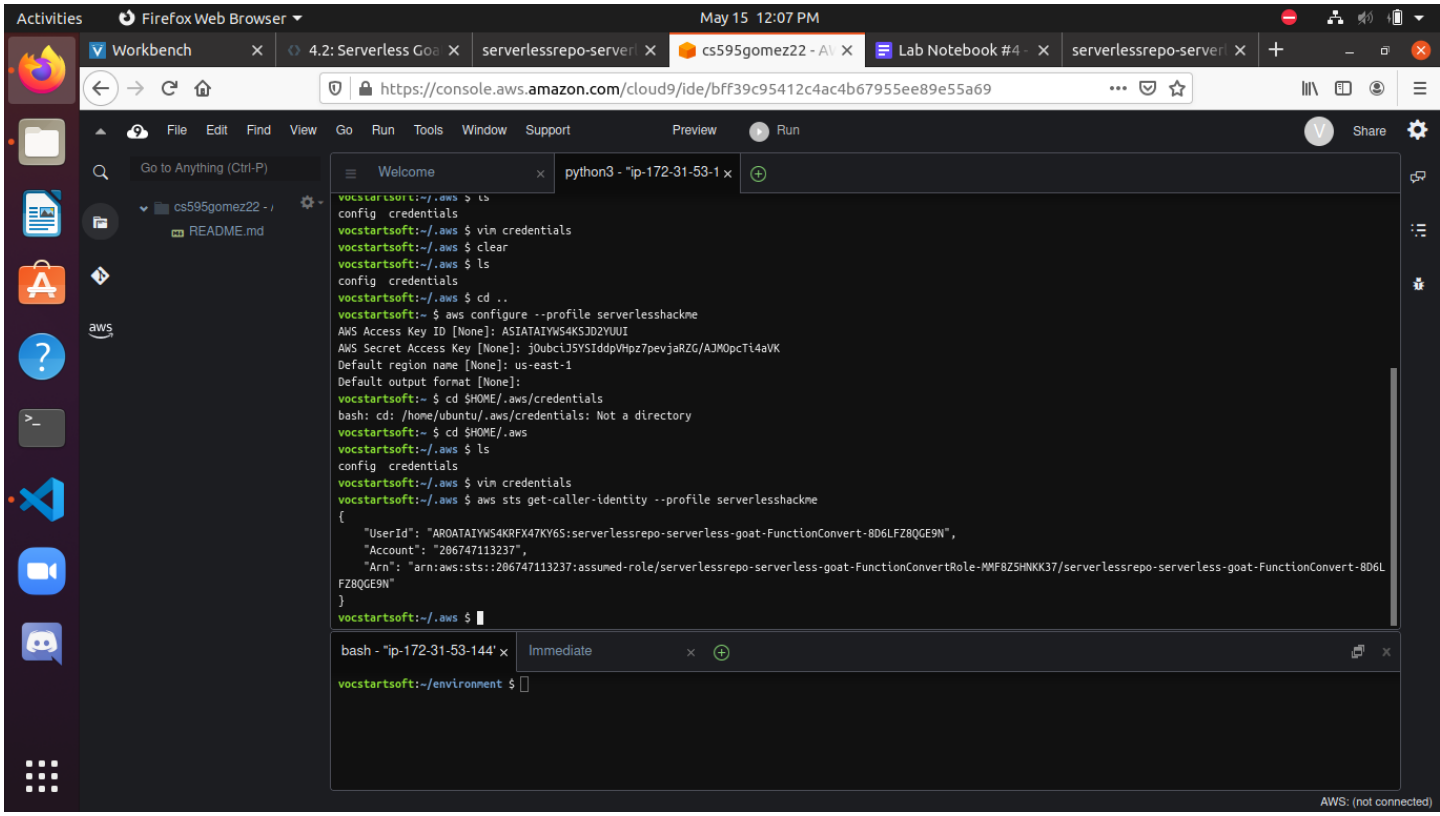
Visit the URL associated with the S3 bucket and take a screenshot of what it reveals.



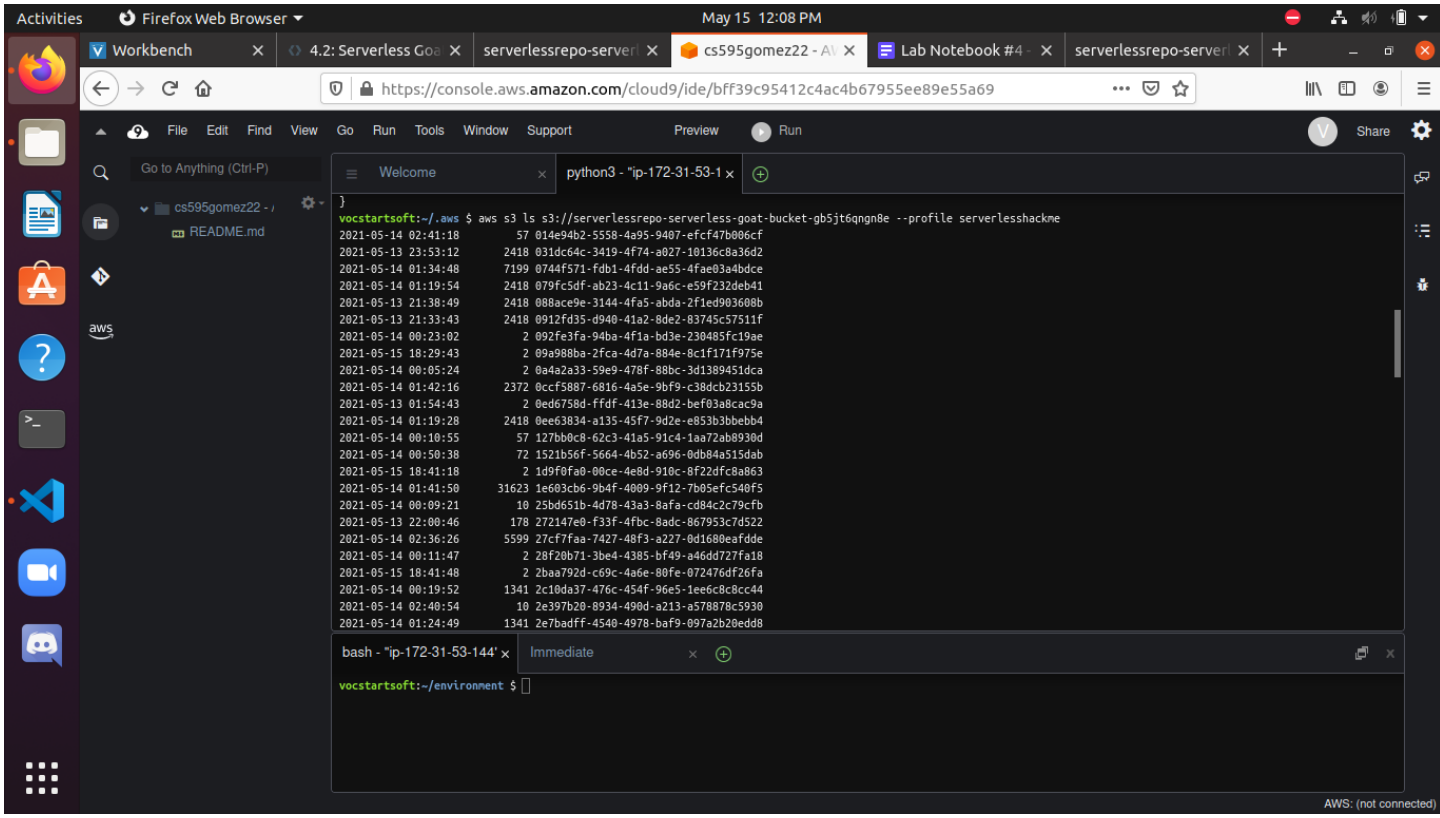
Find a document that has been converted by another user previously and use its object key to get access to the converted data



Take a screenshot of the output.



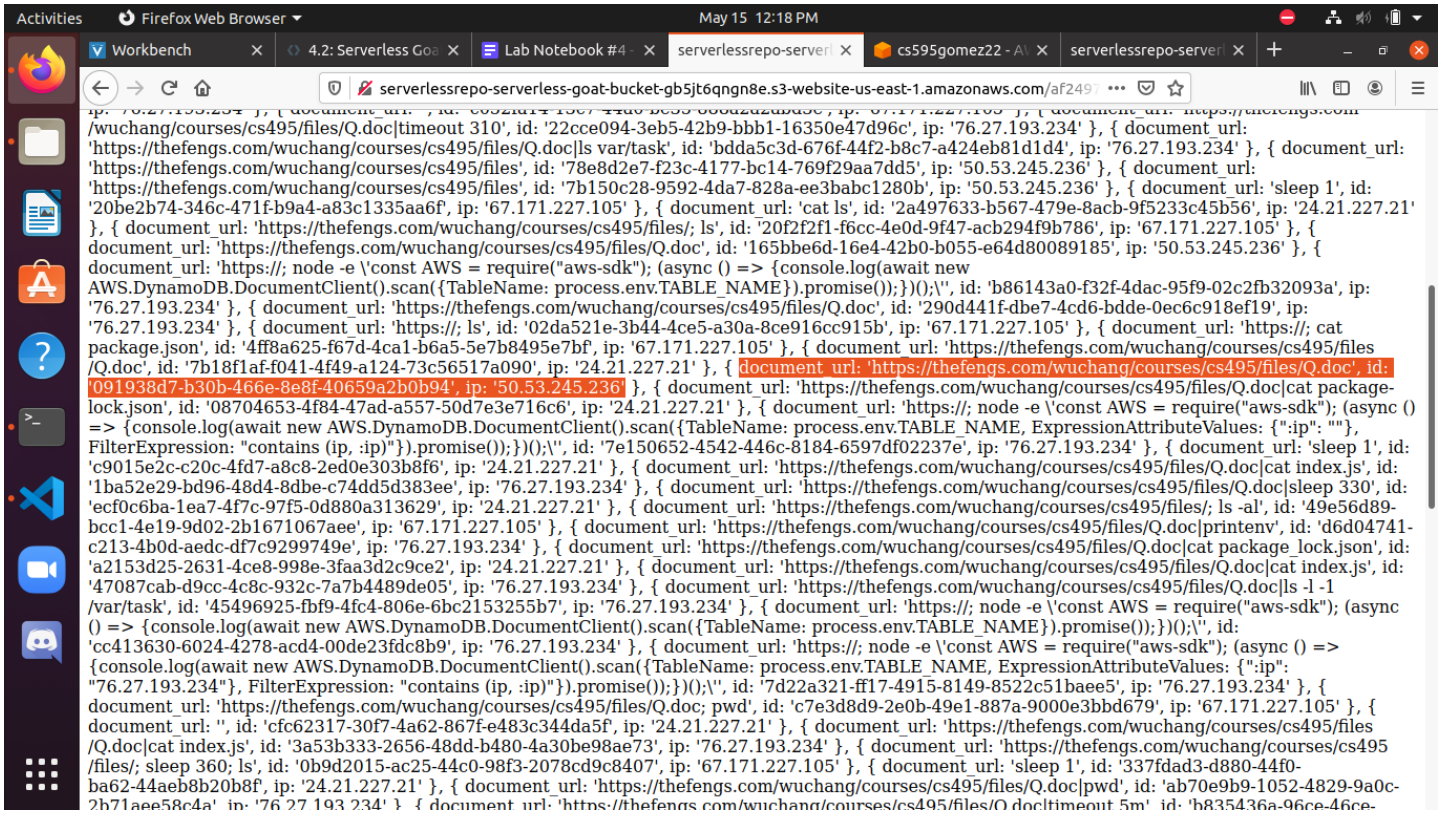
Using the profile, show a screenshot of the objects in the S3 bucket that the function is using to store its results.



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.



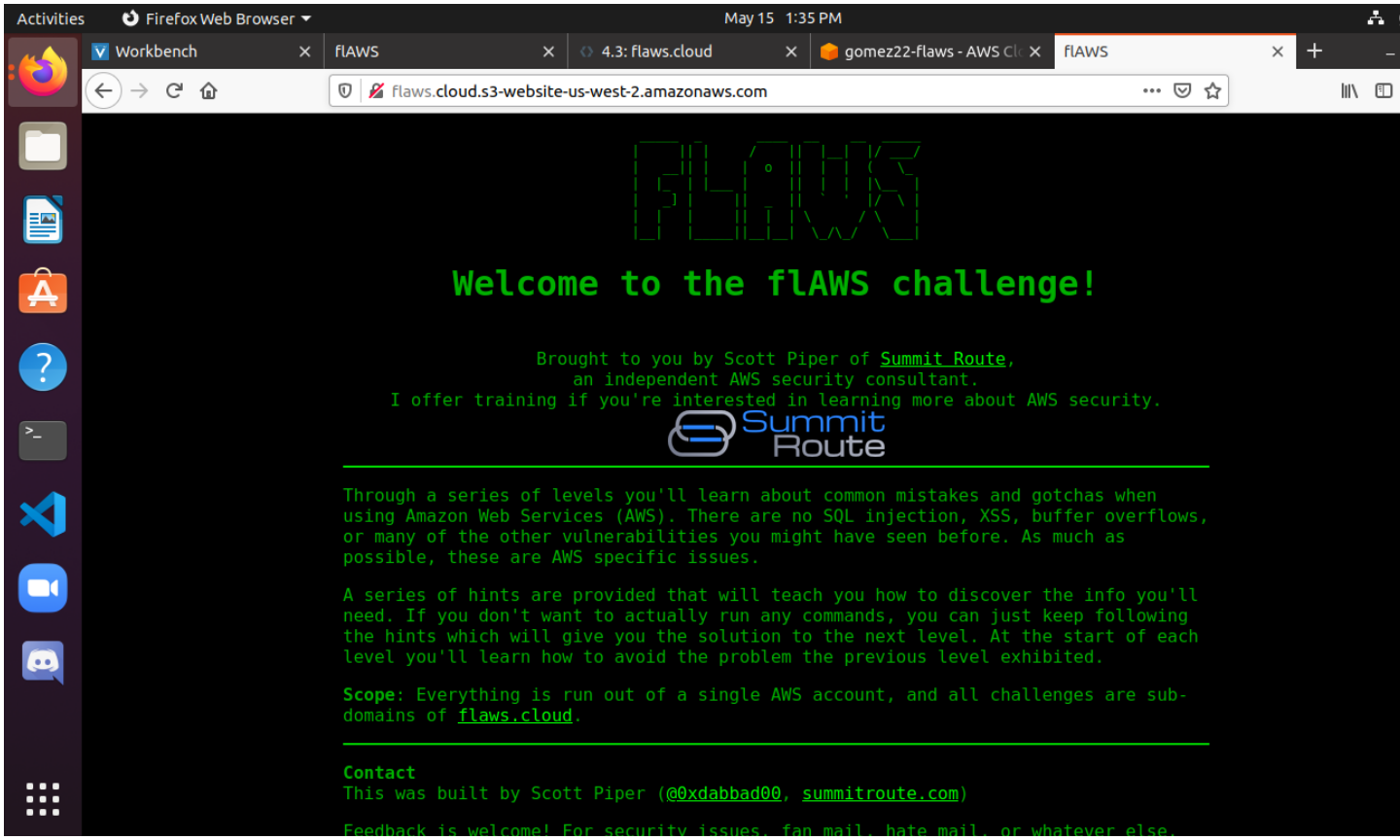
4.3 flaws.cloud

Level 1

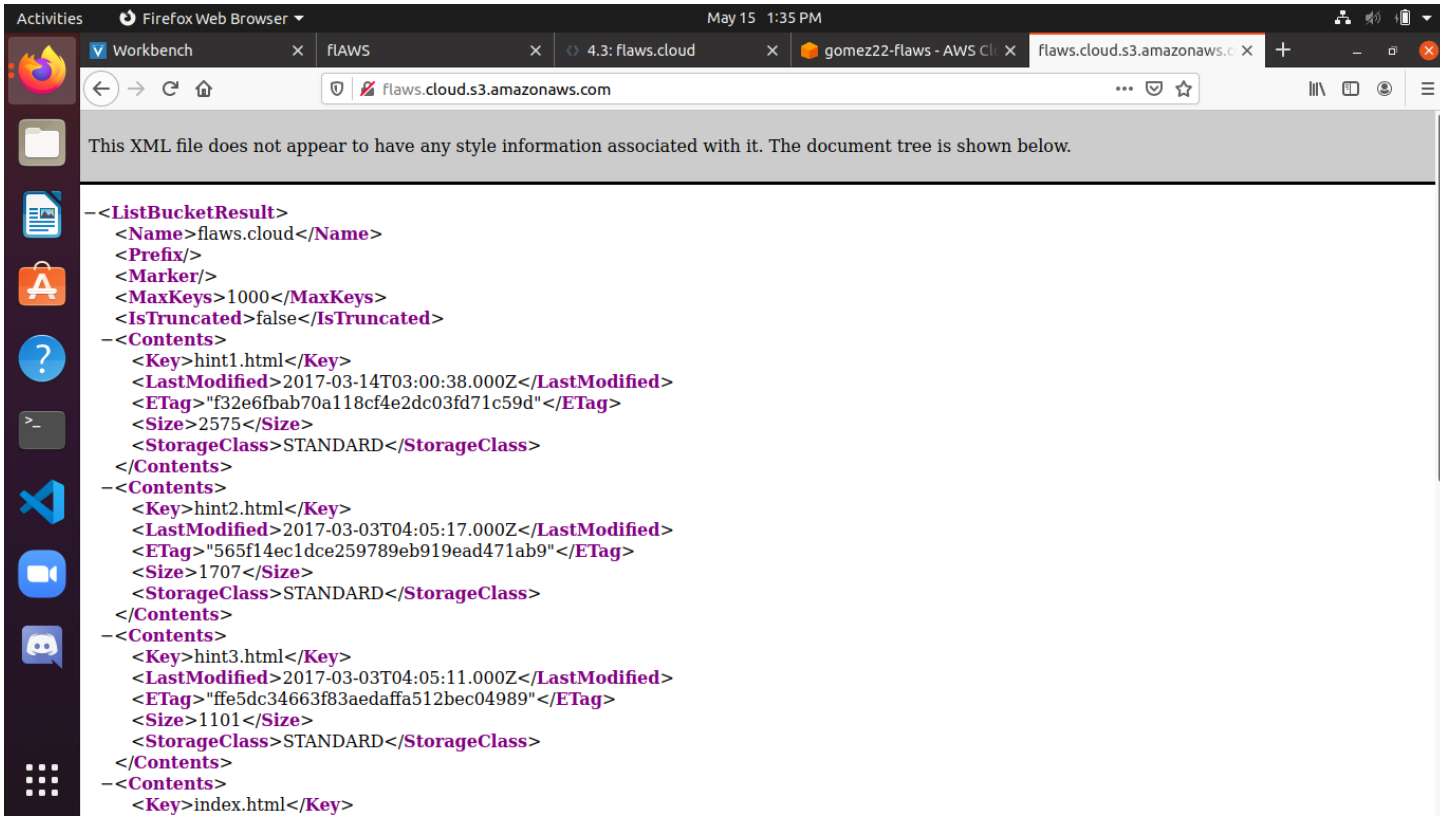
As the command output shows, it is being served out of an S3 bucket. What region is this bucket located in?

-The bucket is in “us-west-2”.

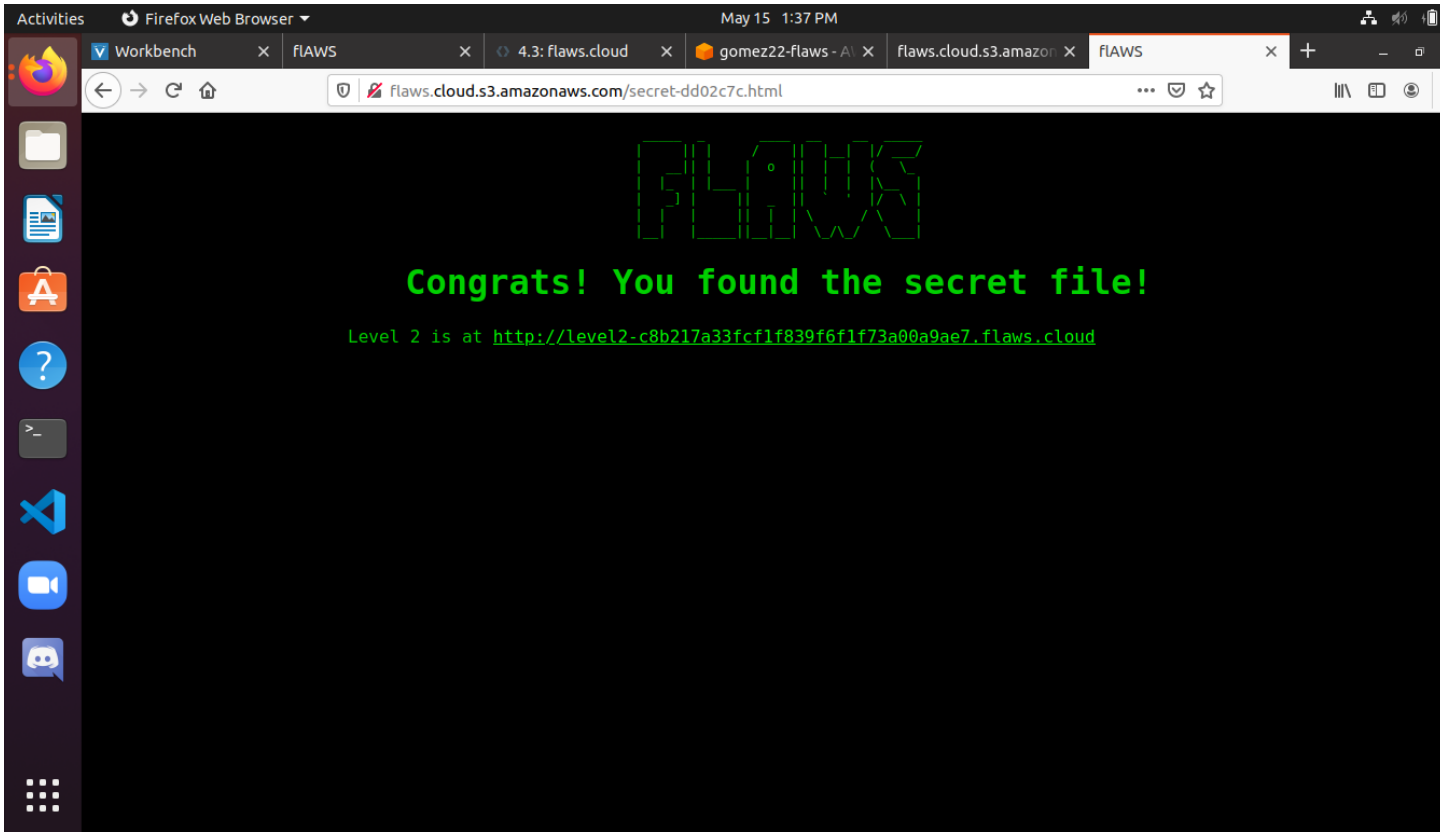
Show the site when visited via this URL



Show the results of visiting this URL.

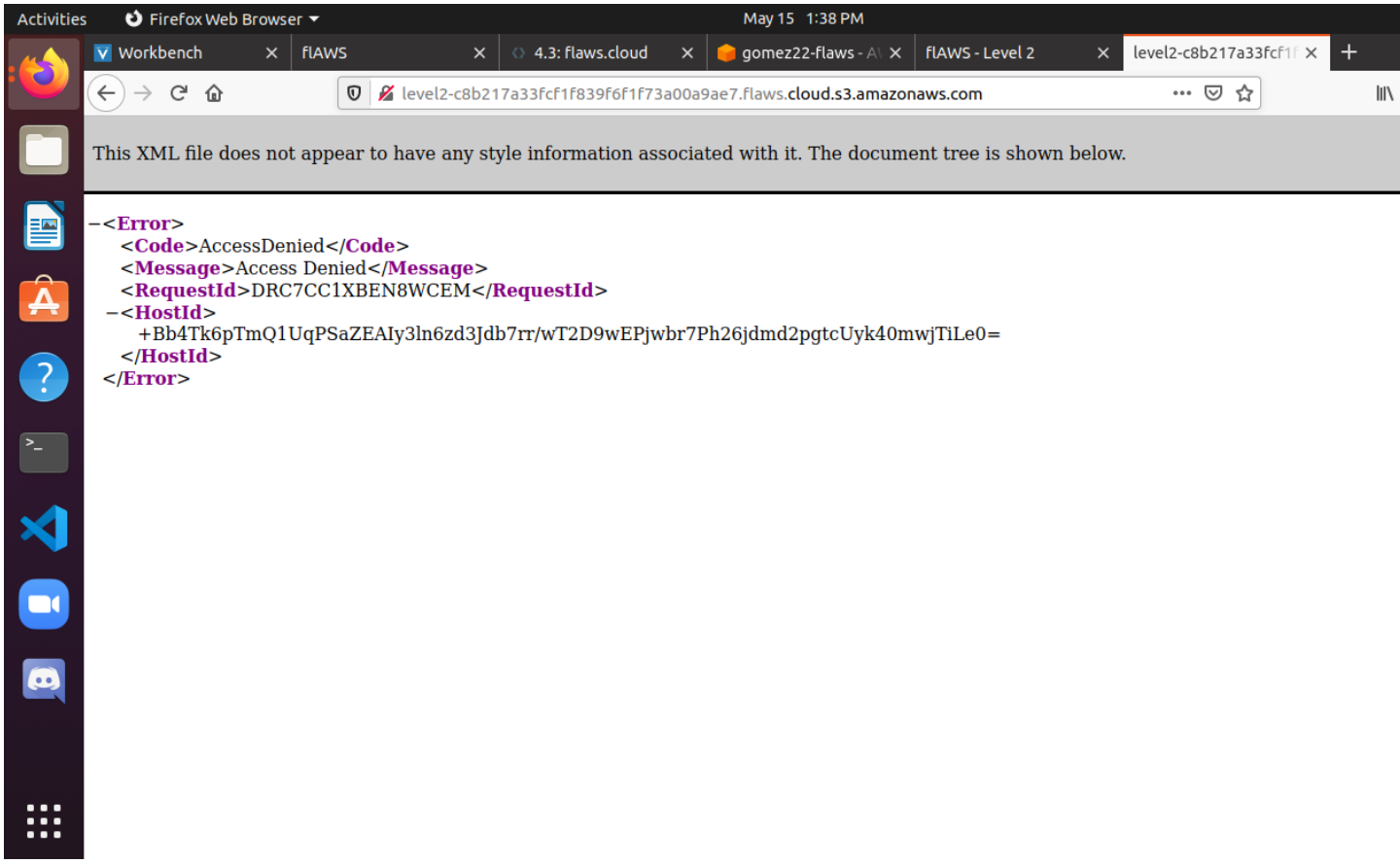


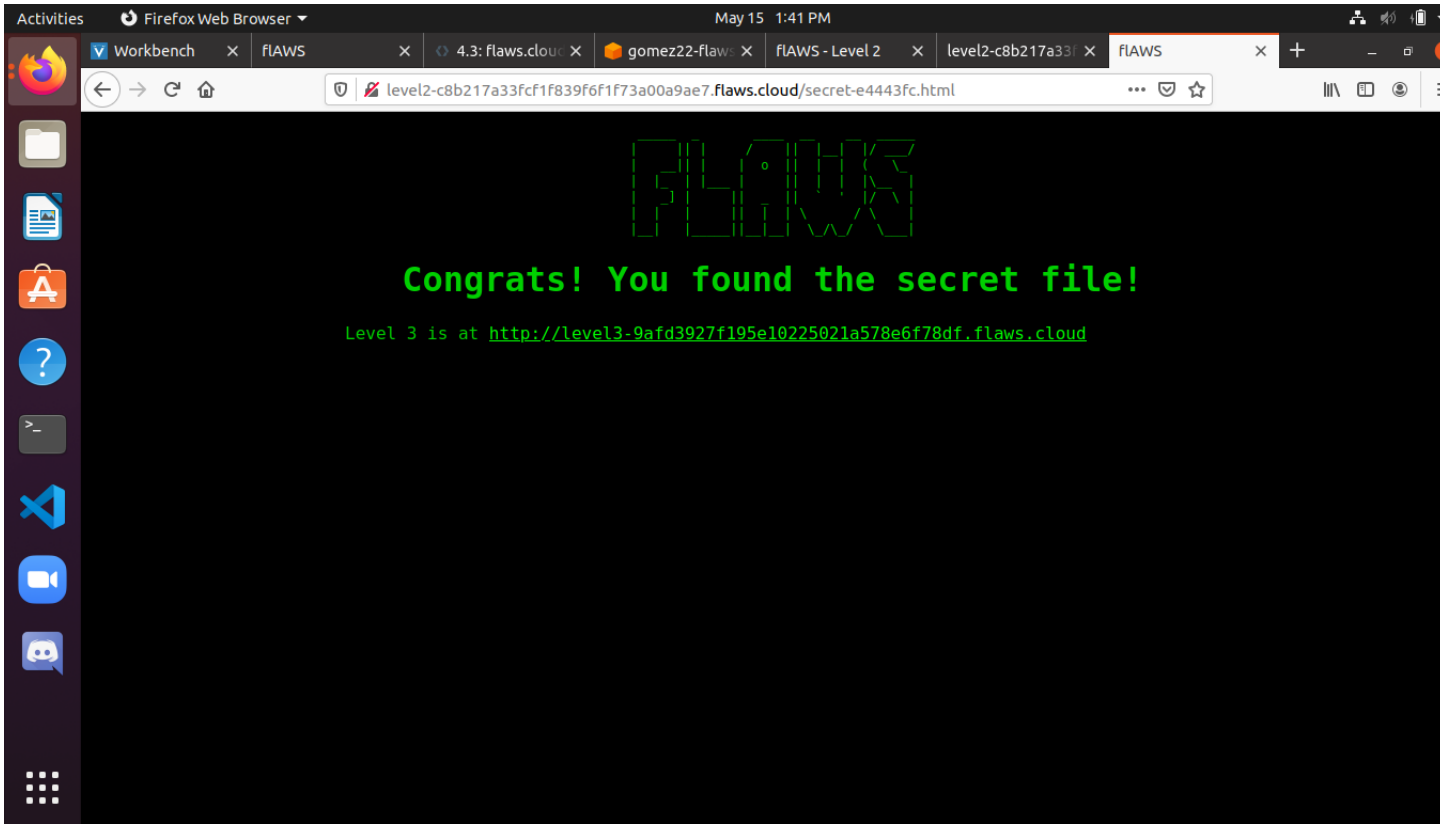
You can directly append the filename to the site's URL to access it. Show the results of visiting this URL and continue to the next level.



Level 2

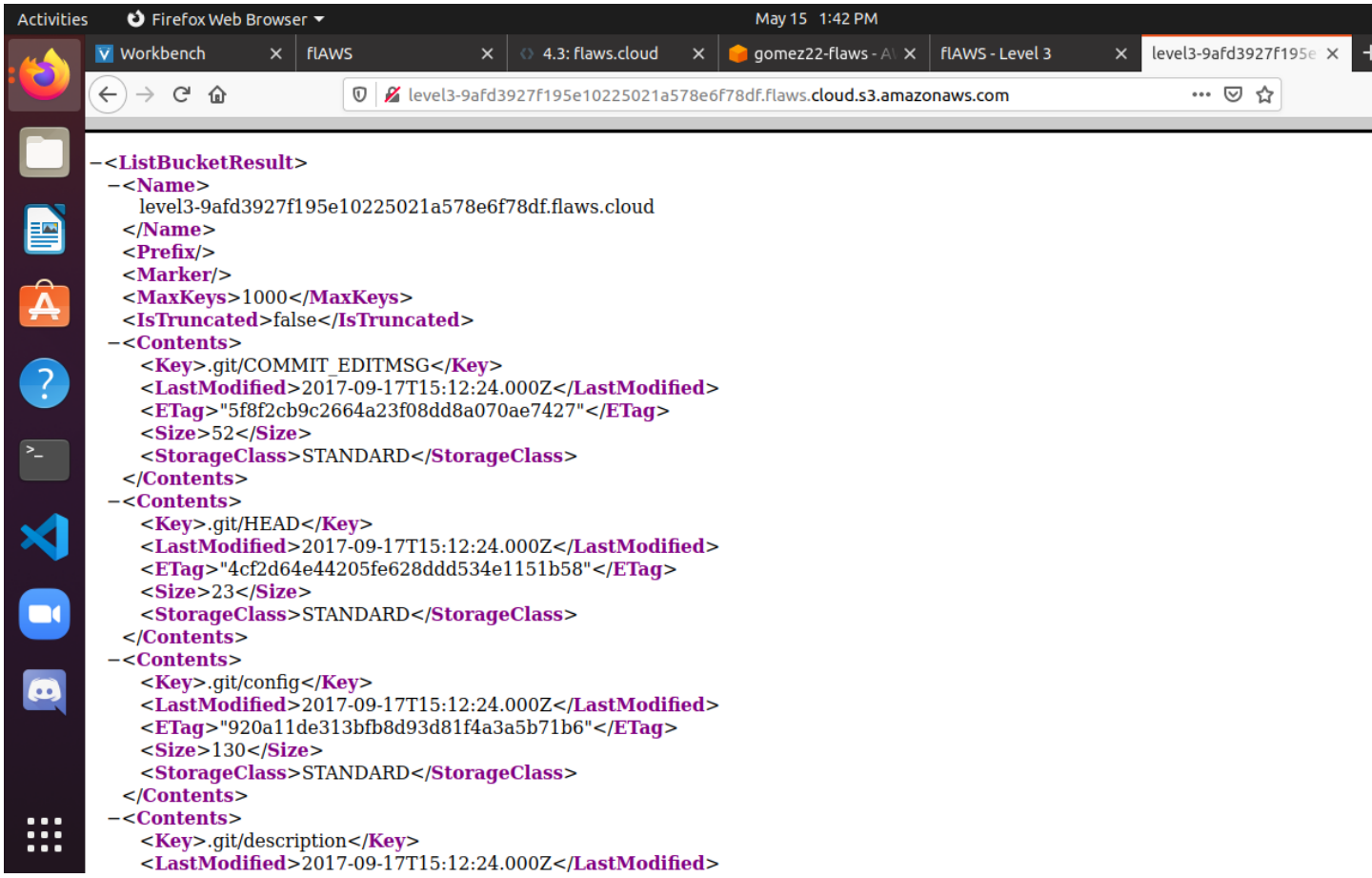
Use the prior method of accessing http://<site>.s3.amazonaws.com to attempt to list the bucket's contents. Show the result in a screenshot.


Activities
Firefox Web Browser
May 15 1:38 PM
Workbench
flAWS
4.3: flaws.cloud
gomez22-flaws - A
flAWS - Level 2
level2-c8b217a33fcf1
level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud.s3.amazonaws.com
This XML file does not appear to have any style information associated with it. The document tree is shown below.
−<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>DRC7CC1XBEN8WCEM</RequestId>
−<HostId>
+Bb4Tk6pTmQ1UqPSaZEAIy3ln6zd3Jdb7rr/wT2D9wEPjwbr7Ph26jdmd2pgtcUyk40mwjTiLe0=
</HostId>
</Error>

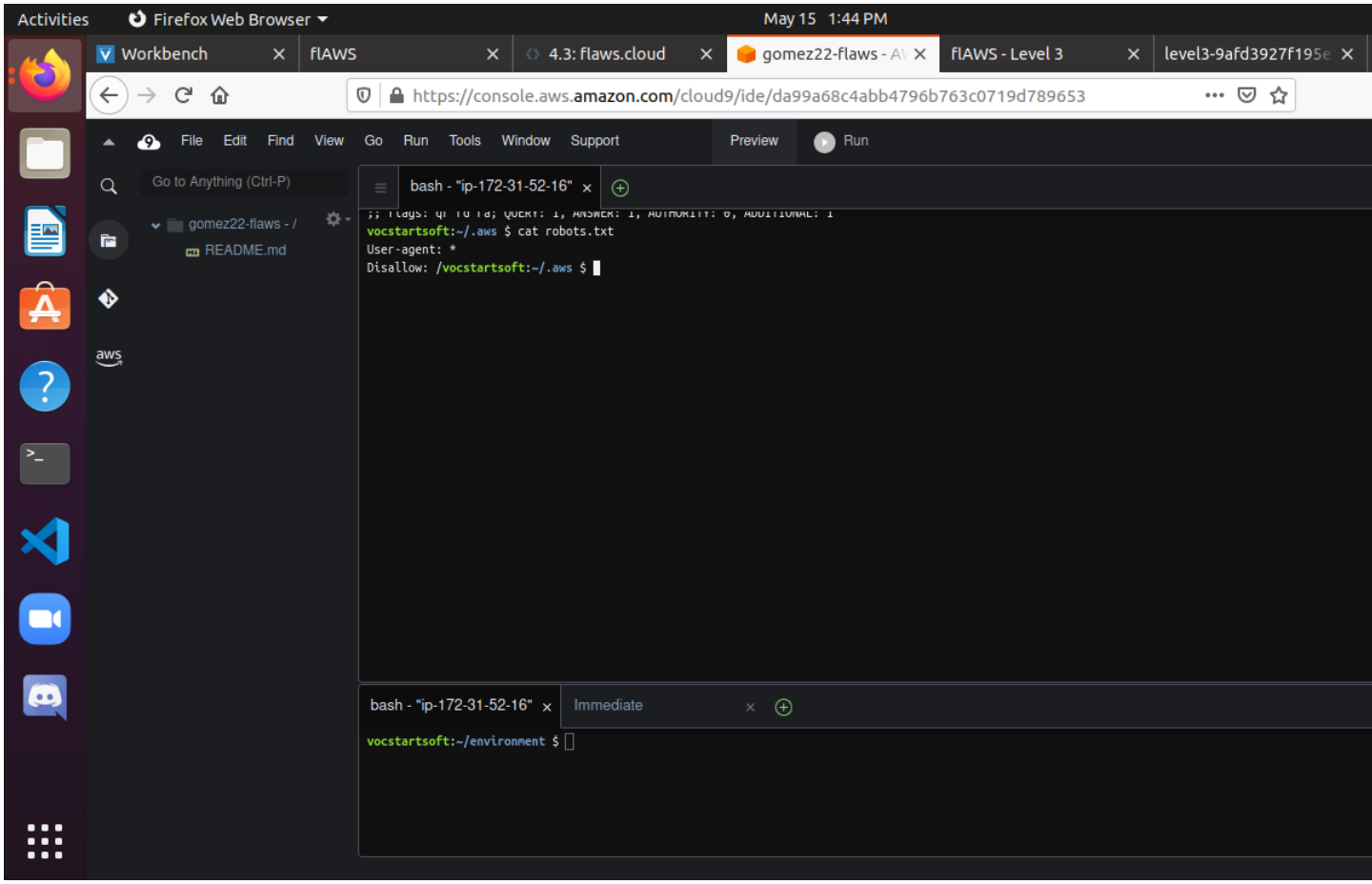


Level 3

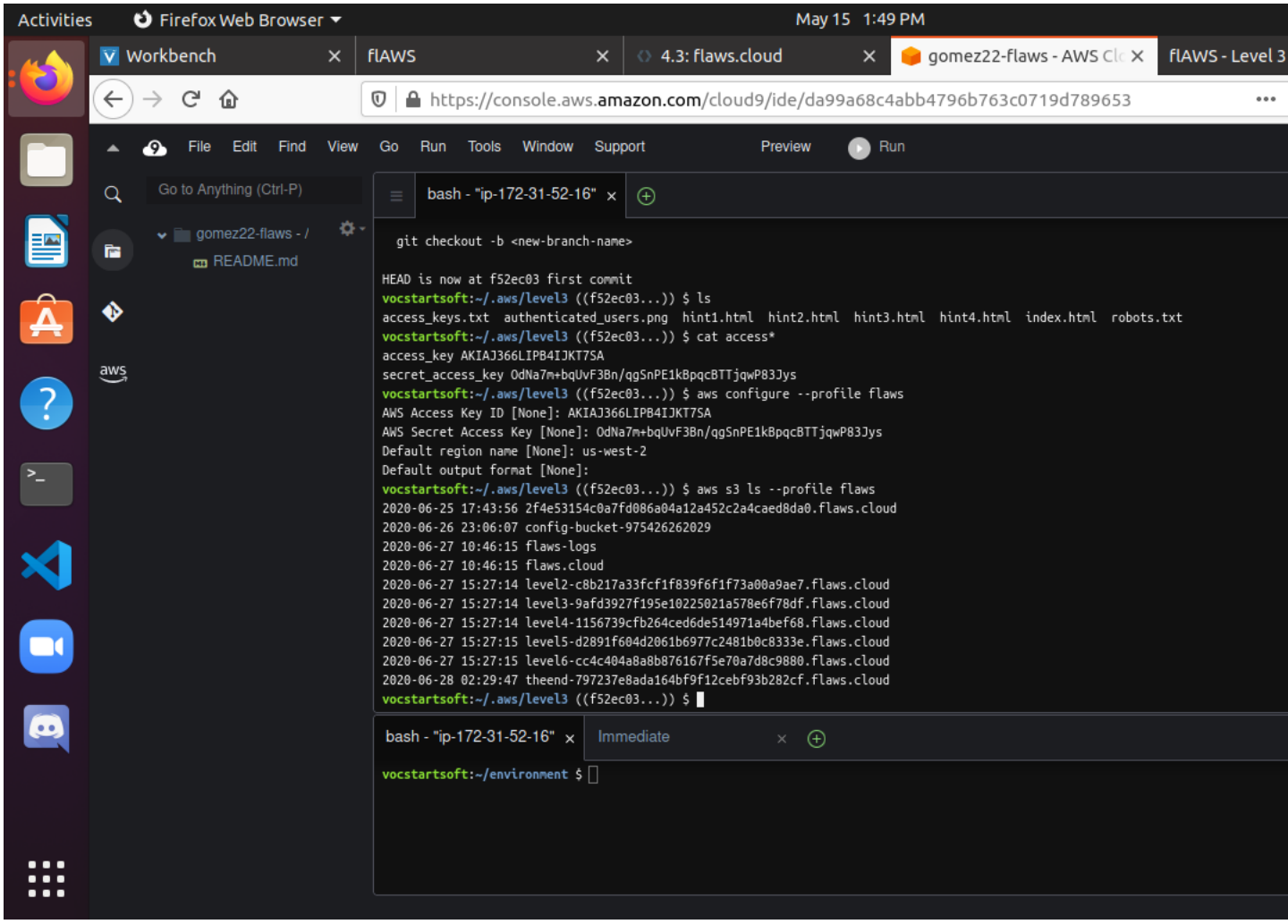
Show the results from attempting to list the bucket as an unauthenticated user via the web http://<site>.s3.amazonaws.com



Show the contents of this file

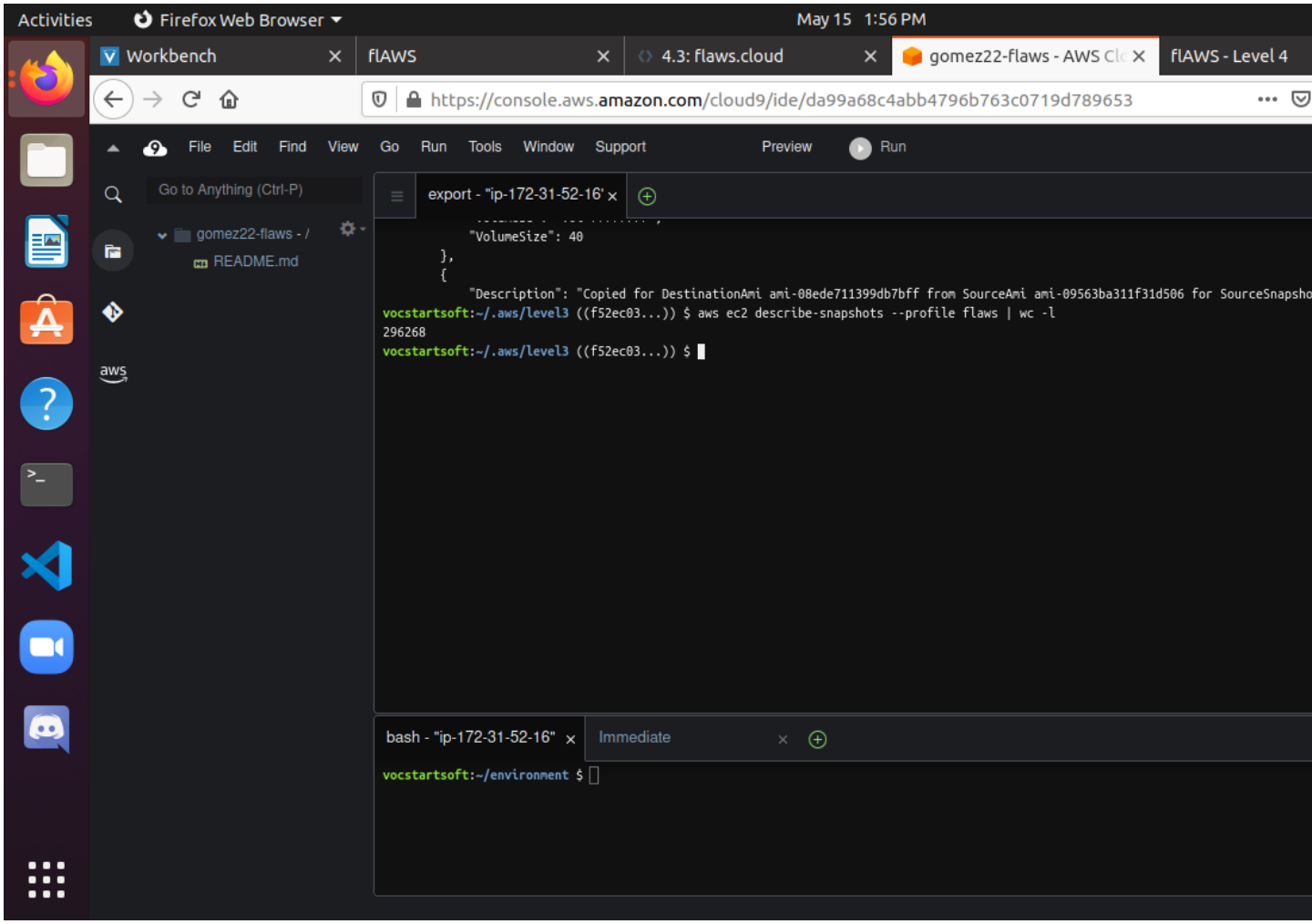


Using the new credentials, show all of the storage buckets it can list.



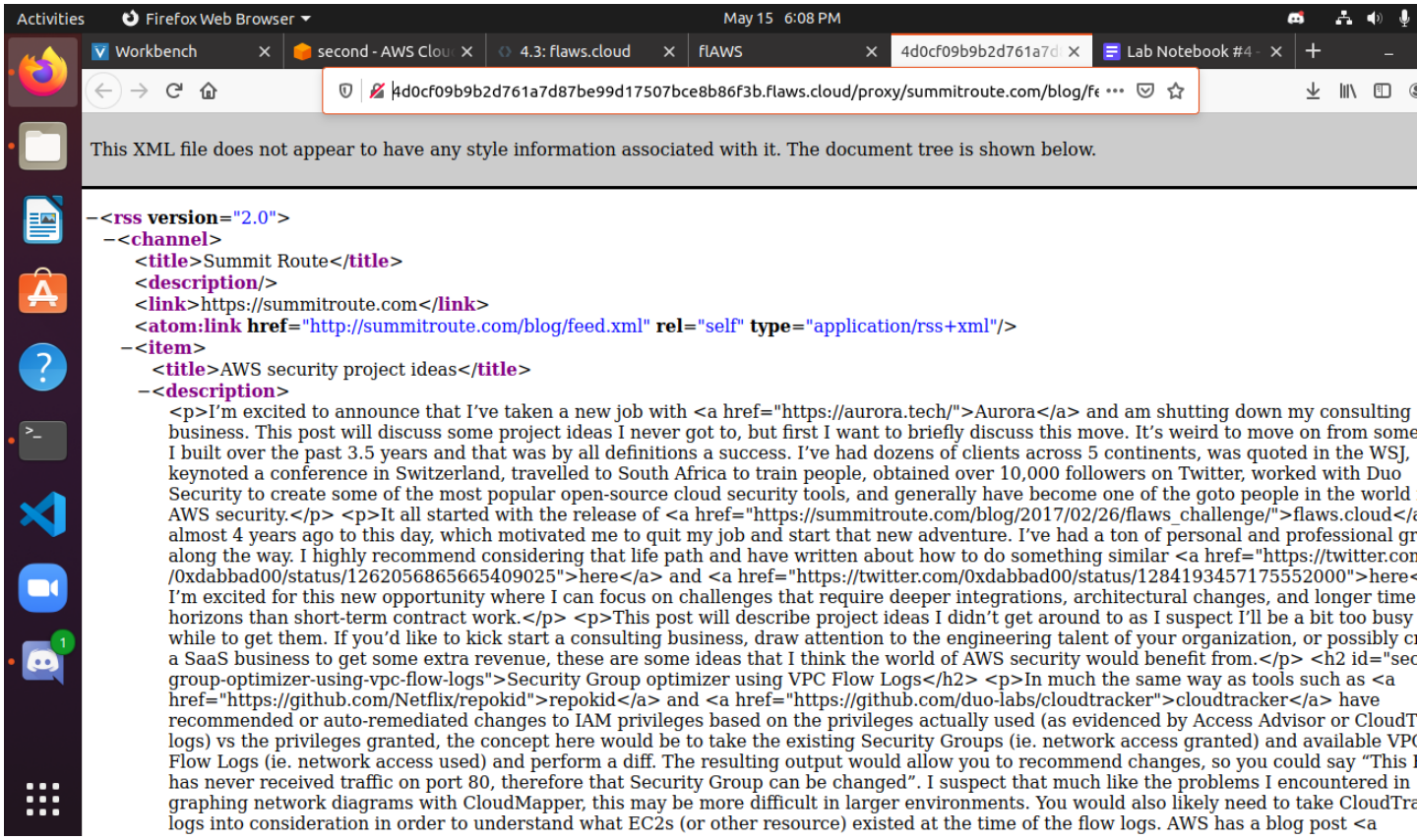
Level 4

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

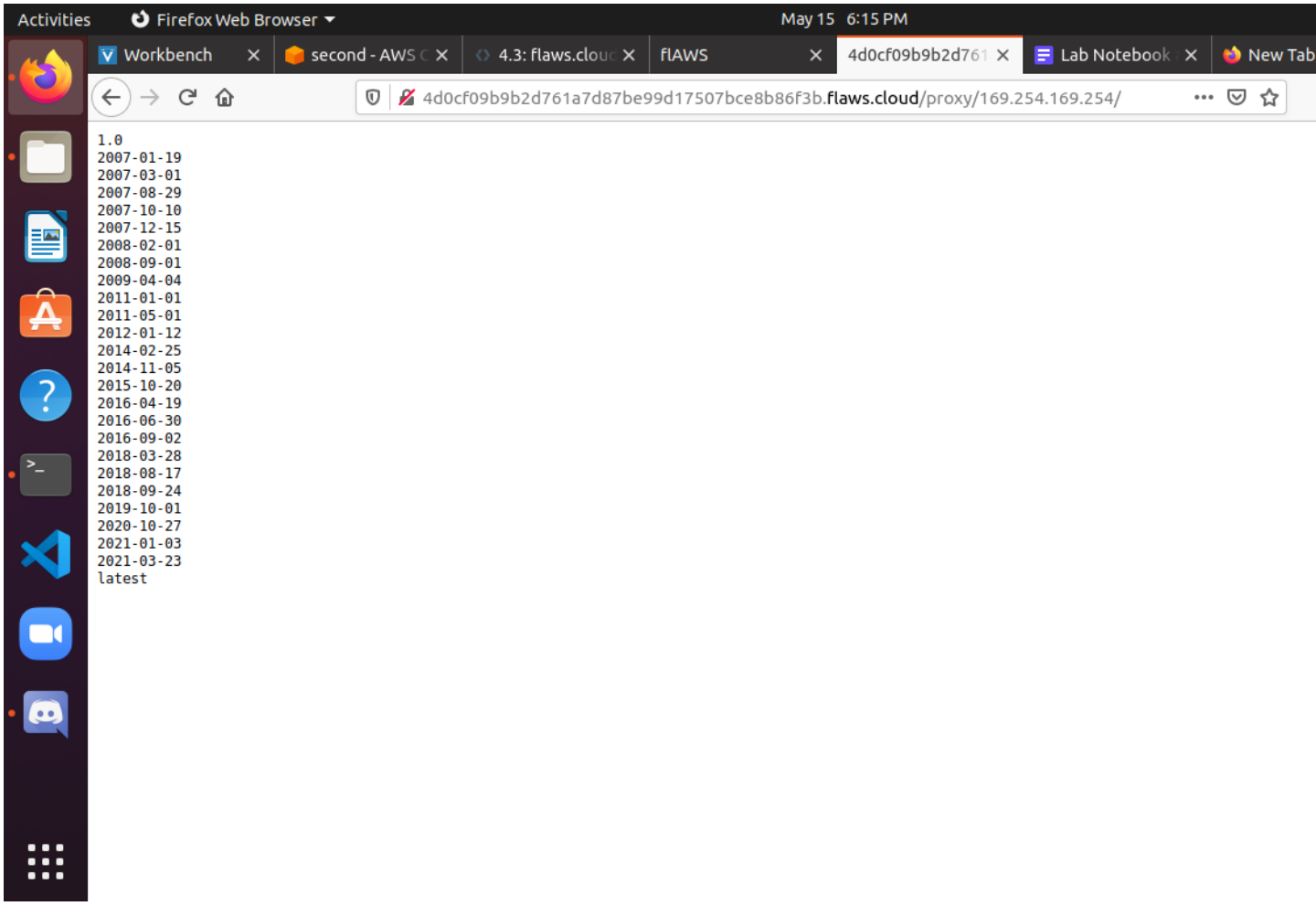


Level 5

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.



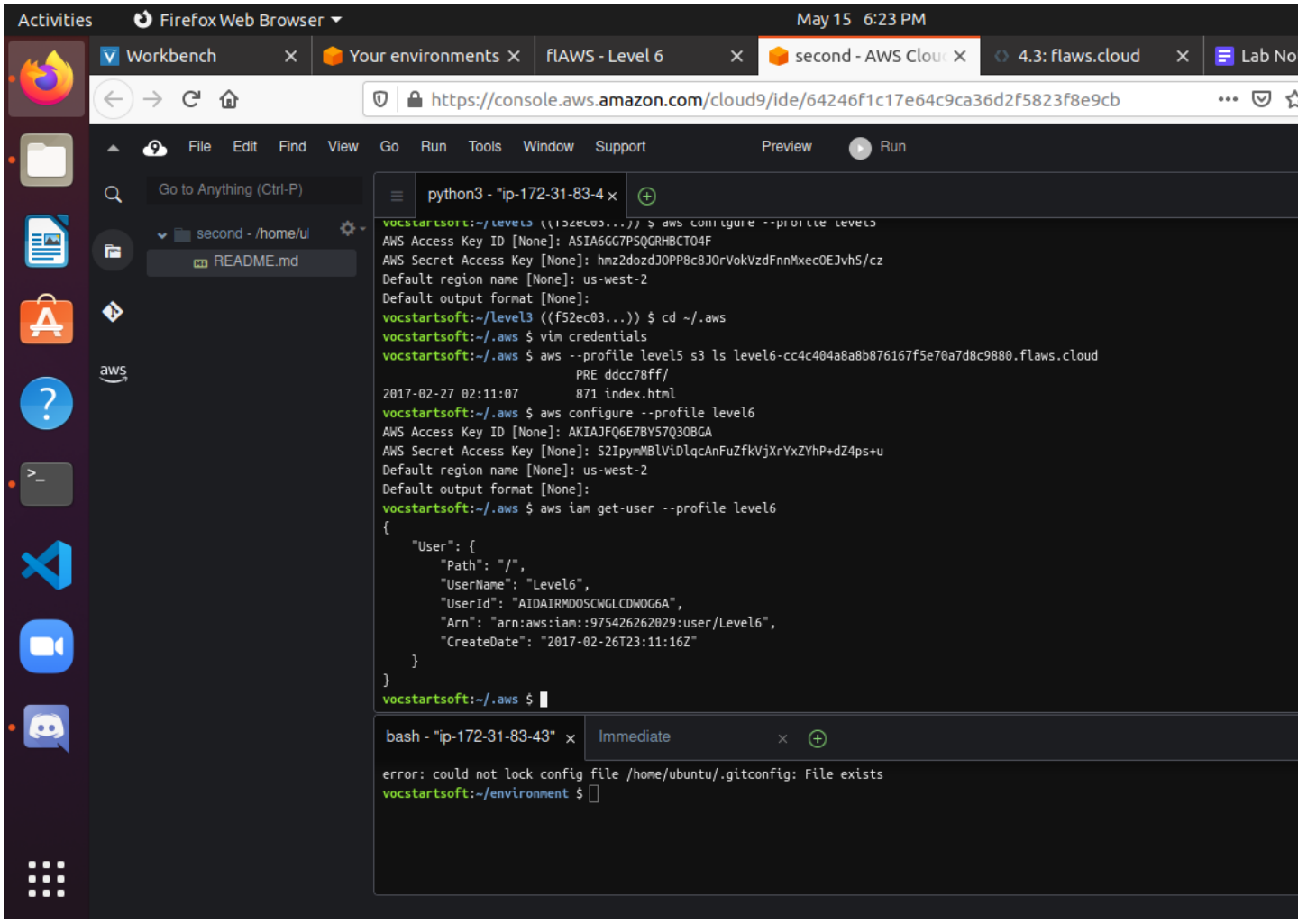
Use the proxy to access this internal metadata service and show the results.



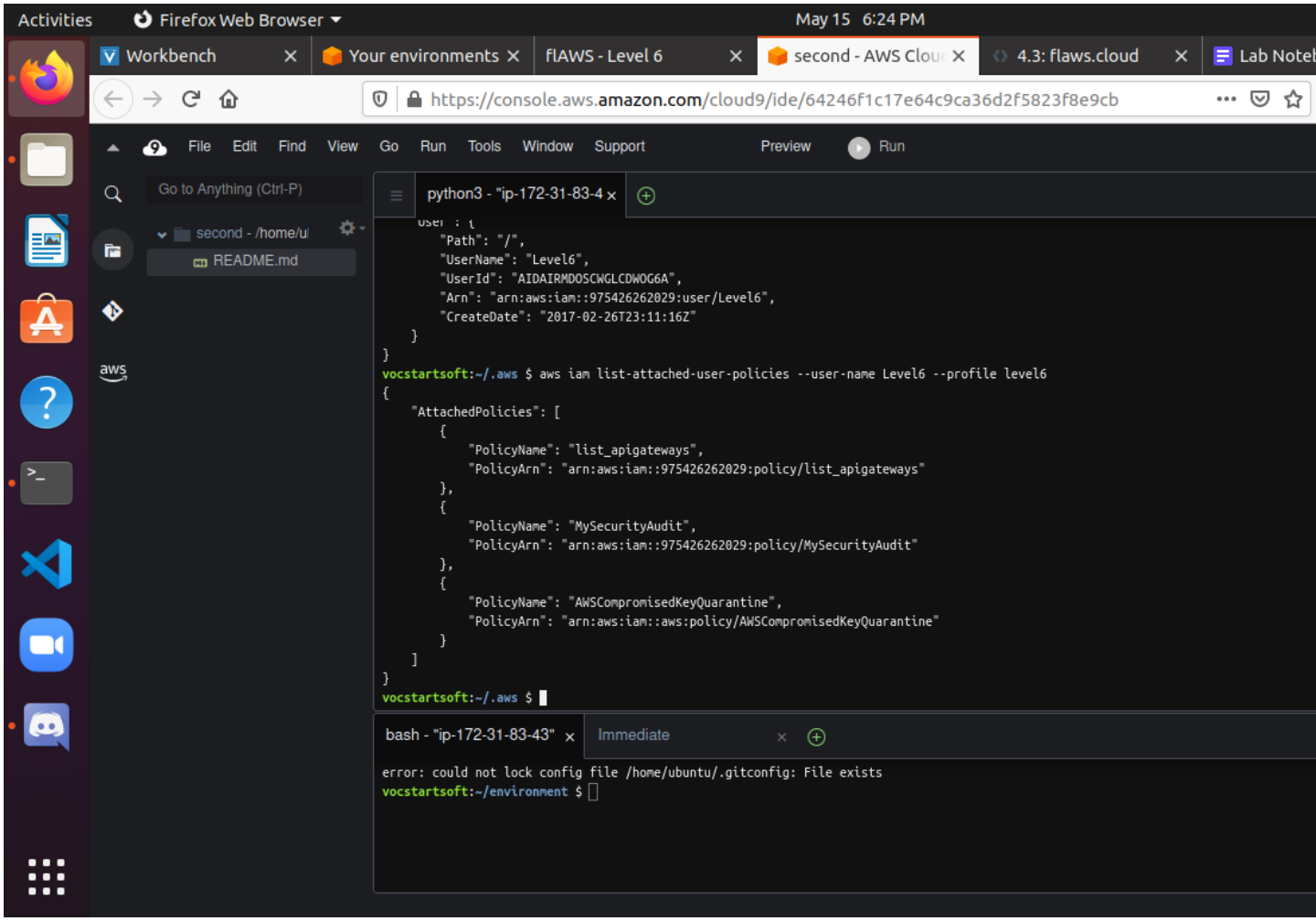
Level 6

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

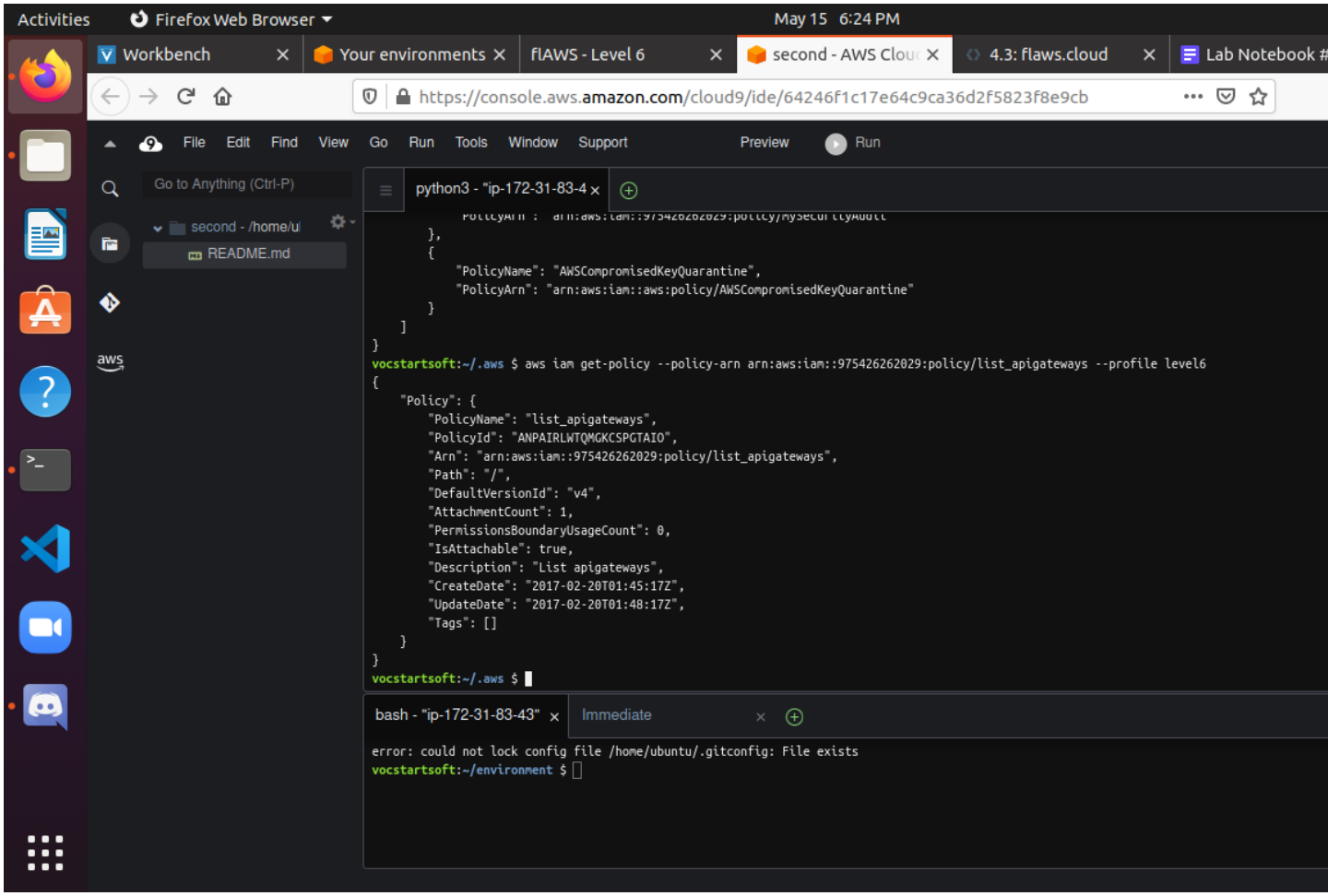
The IAM user



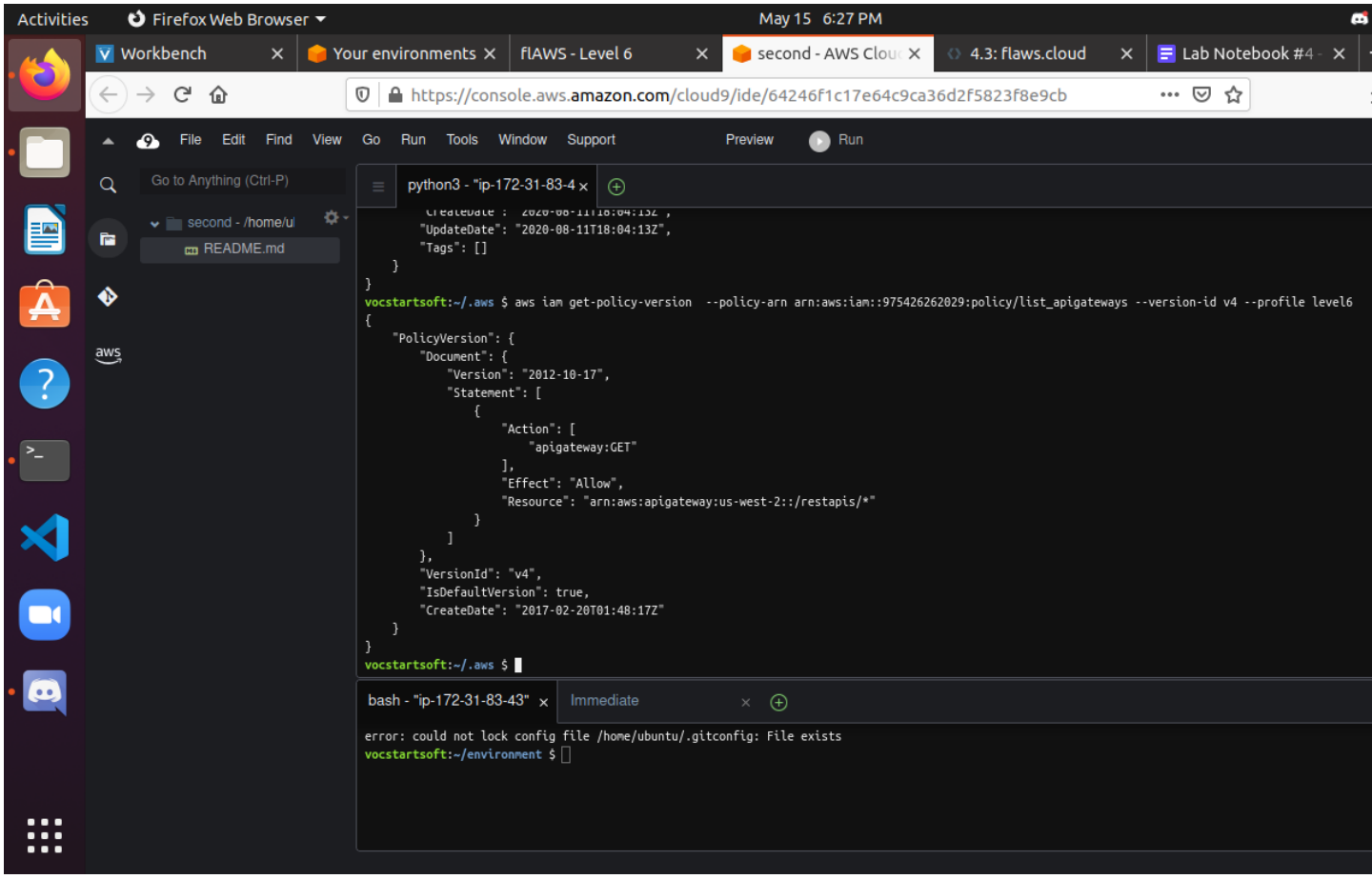
The IAM policies attached to the user including the additional one not associated with the audit



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

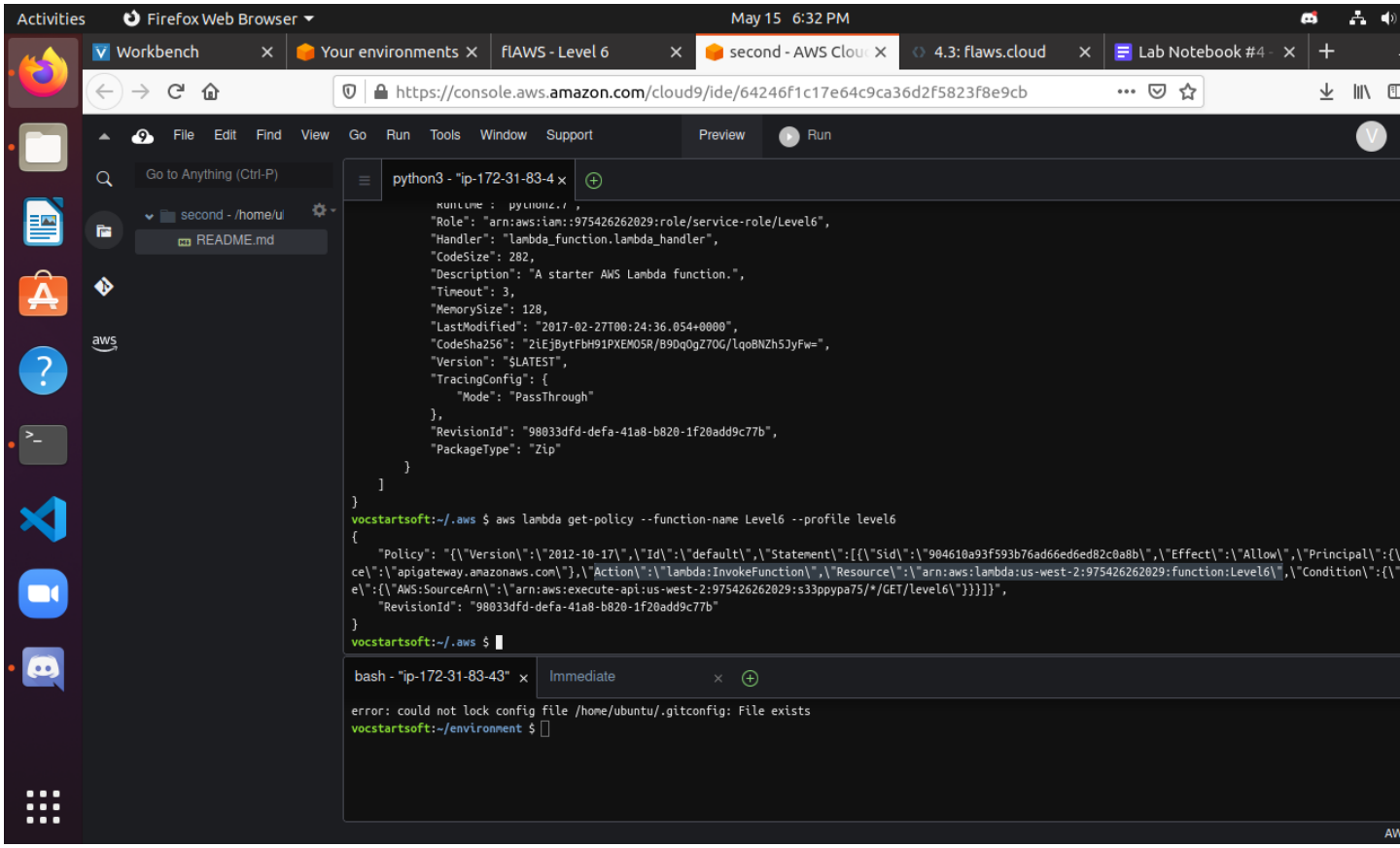


The specific permissions given by the policy for the version that is currently attached to the user.

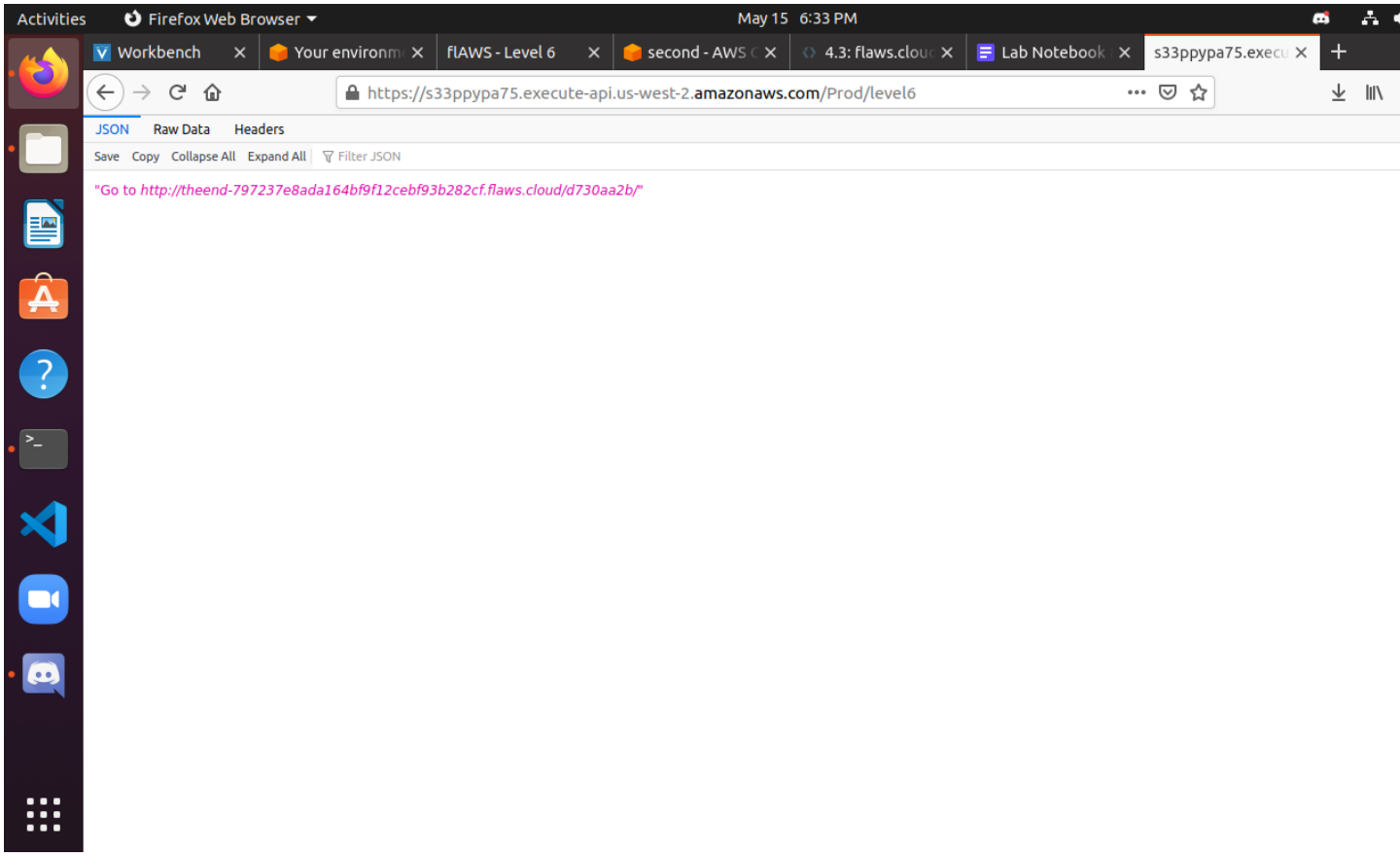


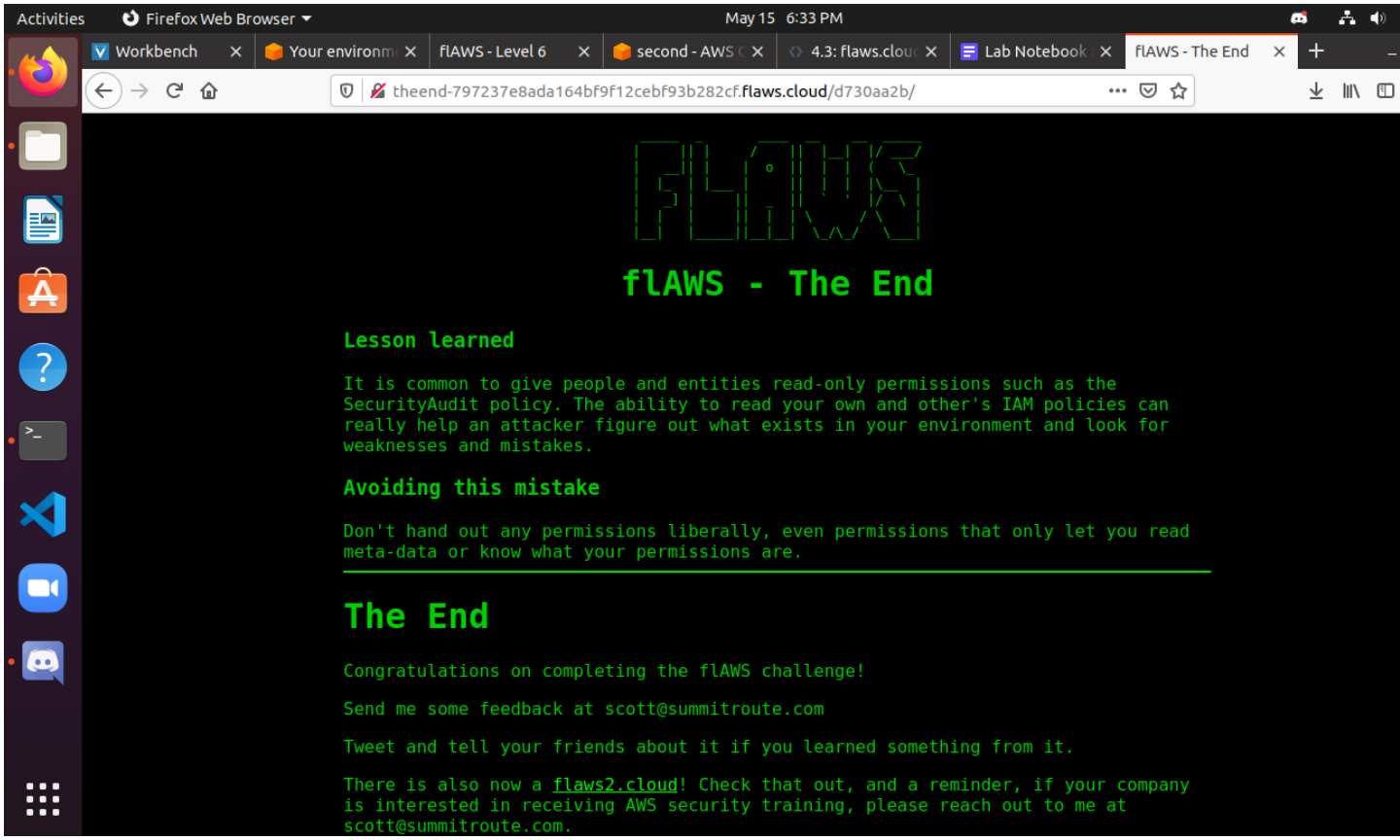
What Action on what Resource does this policy allow?
-The action is "apigateway:GET" on the “arn:aws:apigateway:us-west-2::/restapis/*” resource.

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



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

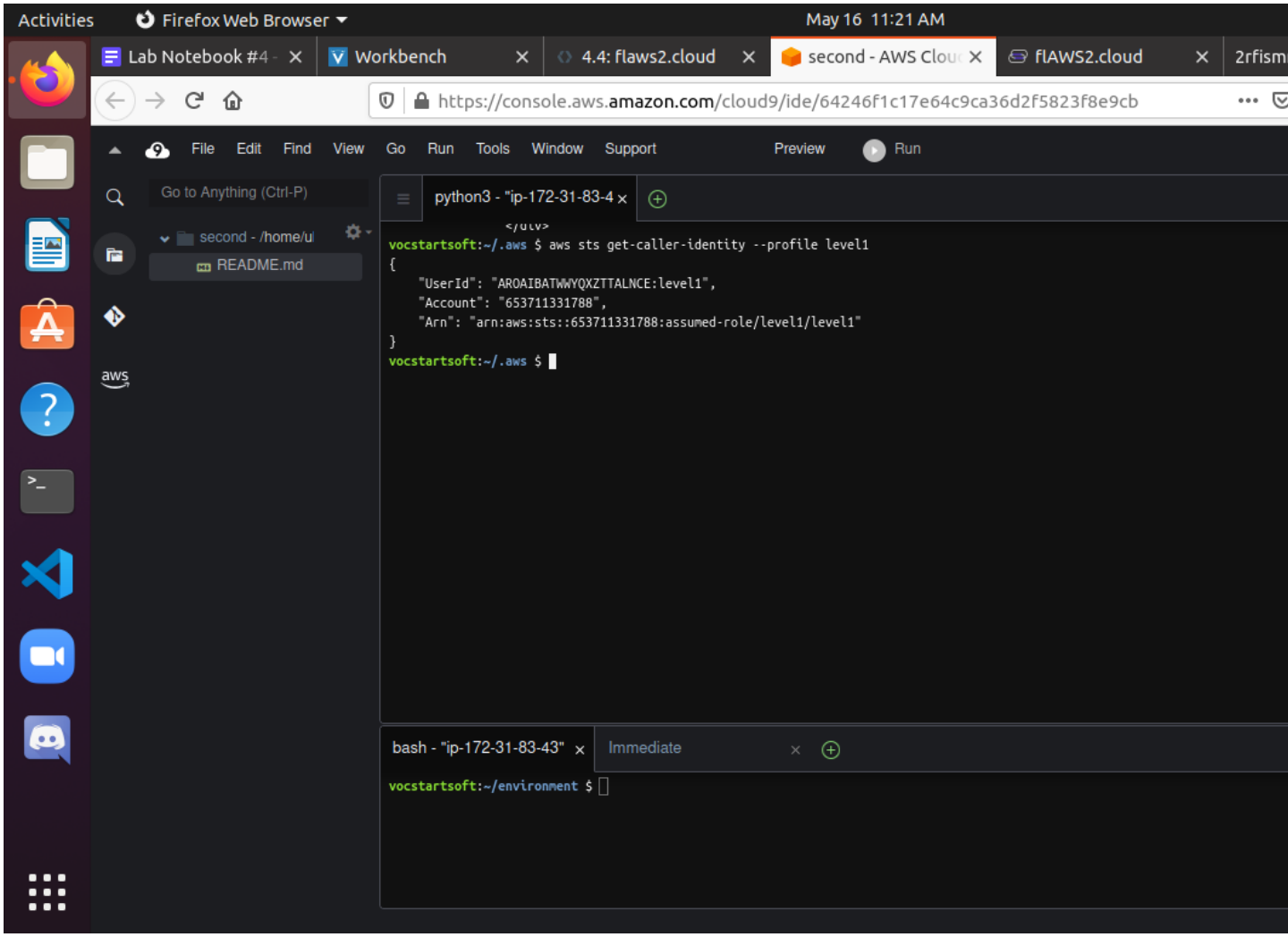

Activities
Firefox Web Browser
May 15 6:33 PM
Workbench
Your environm
flAWS - Level 6
second - AWS
4.3: flaws.cloud
Lab Notebook
s33ppypa75.execu
https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
"Go to http://theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud/d730aa2b/"



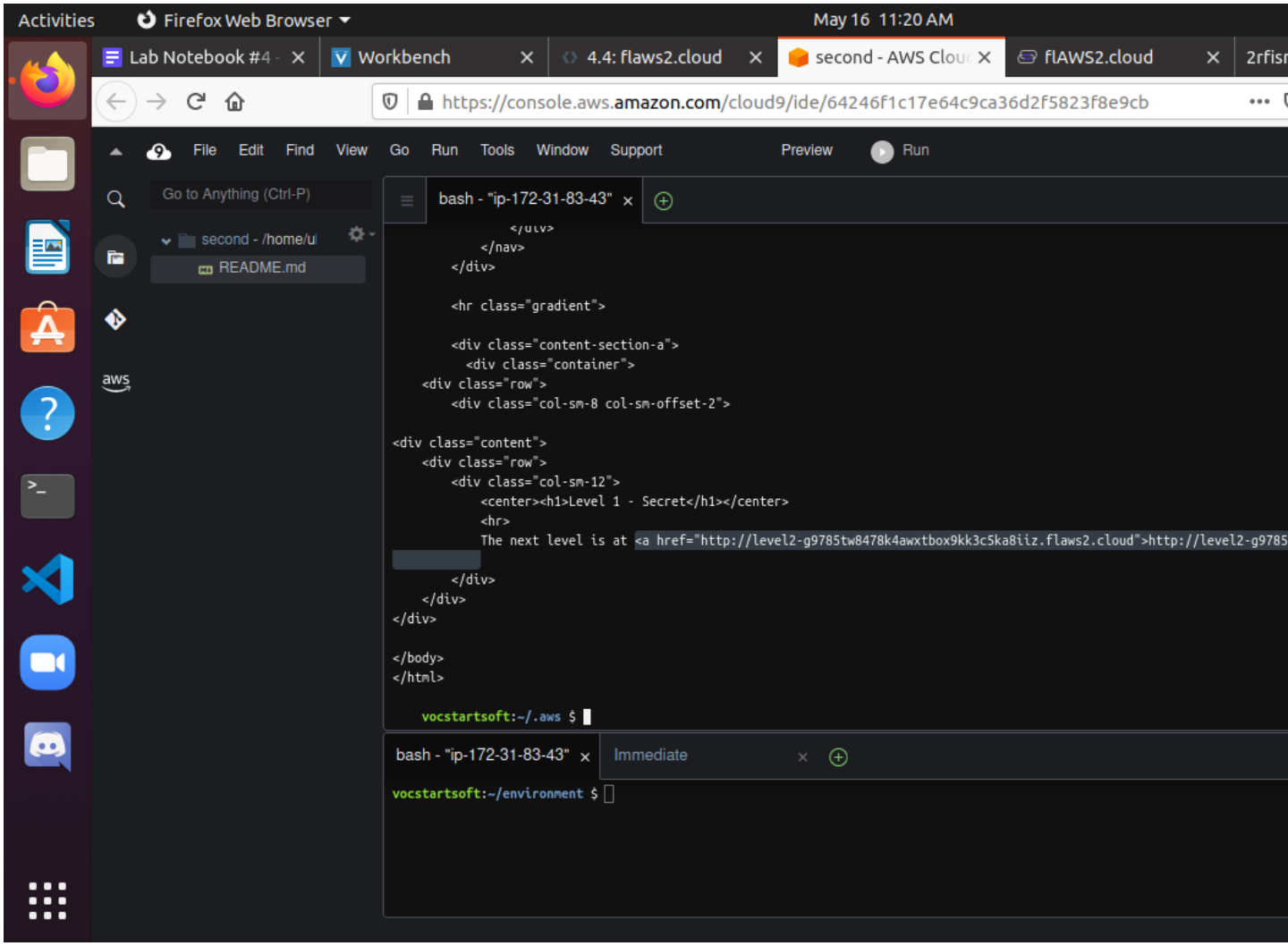
4.4 Flaws2.cloud

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



show the secret URL within the file.

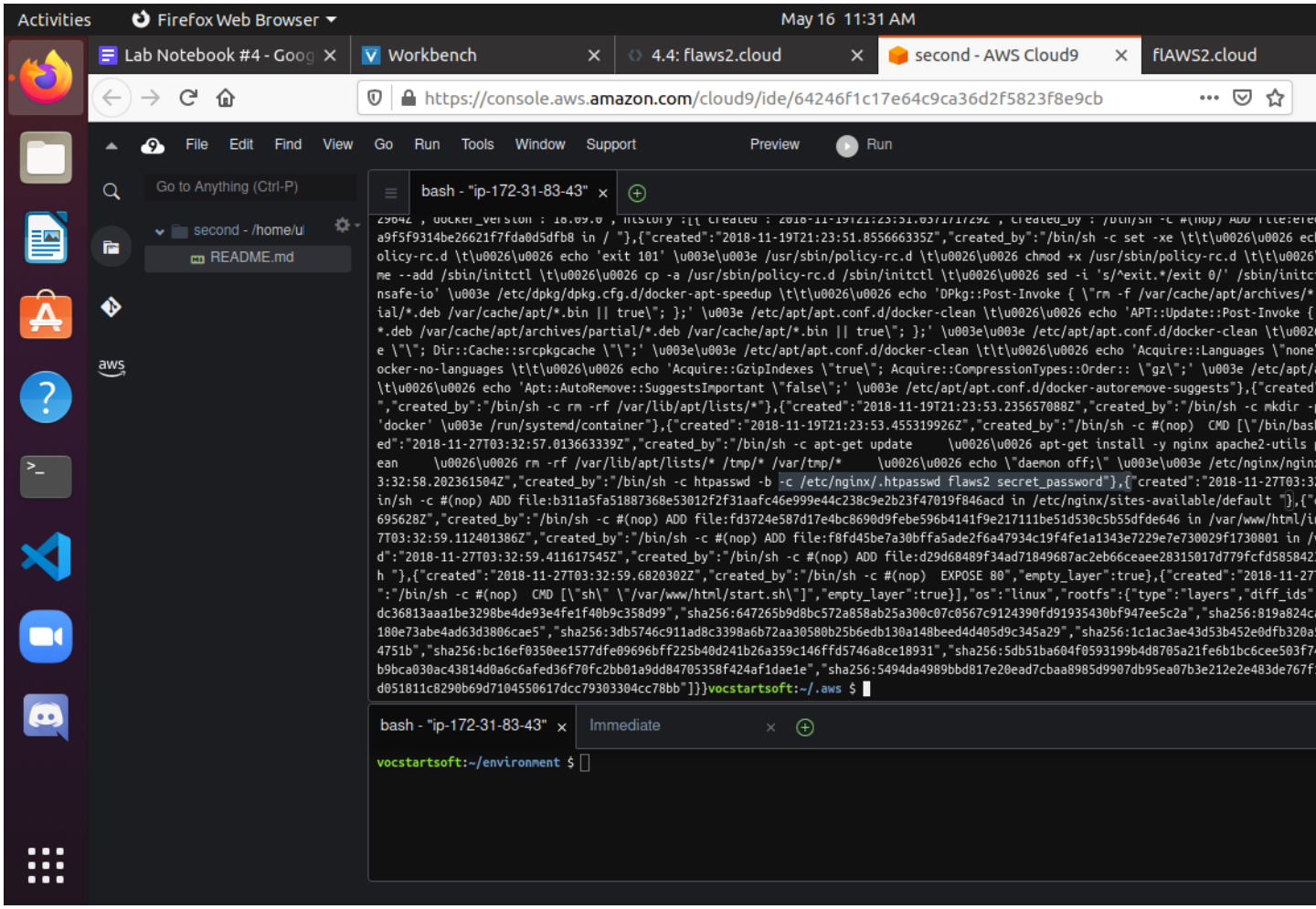

Activities
Firefox Web Browser
May 16 11:20 AM
Lab Notebook #4 -
Workbench
4.4: flaws2.cloud
second - AWS Clou
flAWS2.cloud
https://console.aws.amazon.com/cloud9/ide/64246f1c17e64c9ca36d2f5823f8e9cb
File Edit Find View Go Run Tools Window Support
Preview
Run
Go to Anything (Ctrl-P)
second - /home/u
README.md
bash - "ip-172-31-83-43"
</nav>

<hr class="gradient">

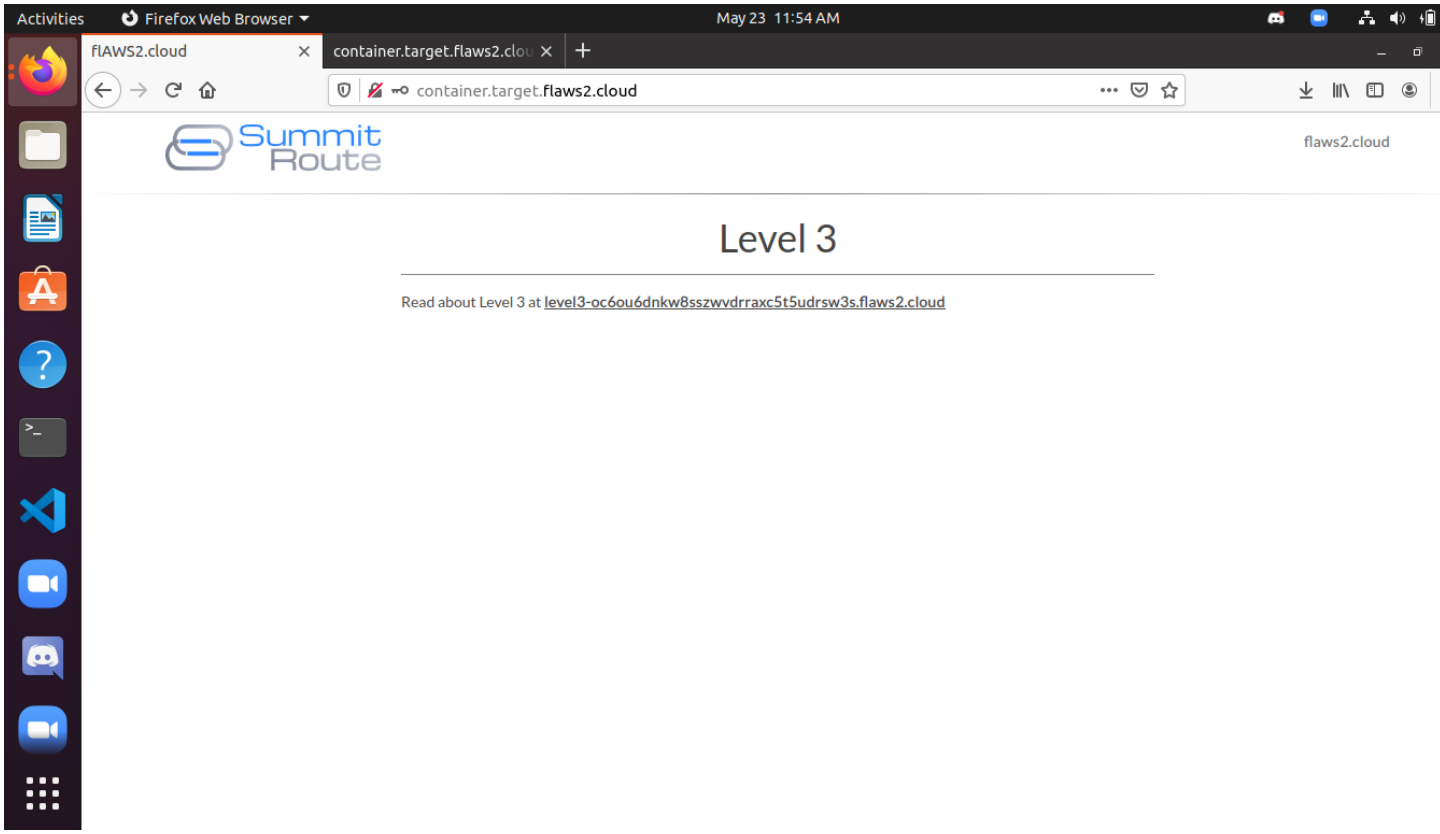
<h1>Level 1 - Secret</h1>
<hr>
The next level is at <a href="http://level2-g9785tw8478k4awxtbox9kk3c5ka8iiz.flaws2.cloud">http://level2-g9785

</body>
</html>
vocstartsoft:~/.aws $
bash - "ip-172-31-83-43"
Immediate
vocstartsoft:~/environment $

Level 2 attacker

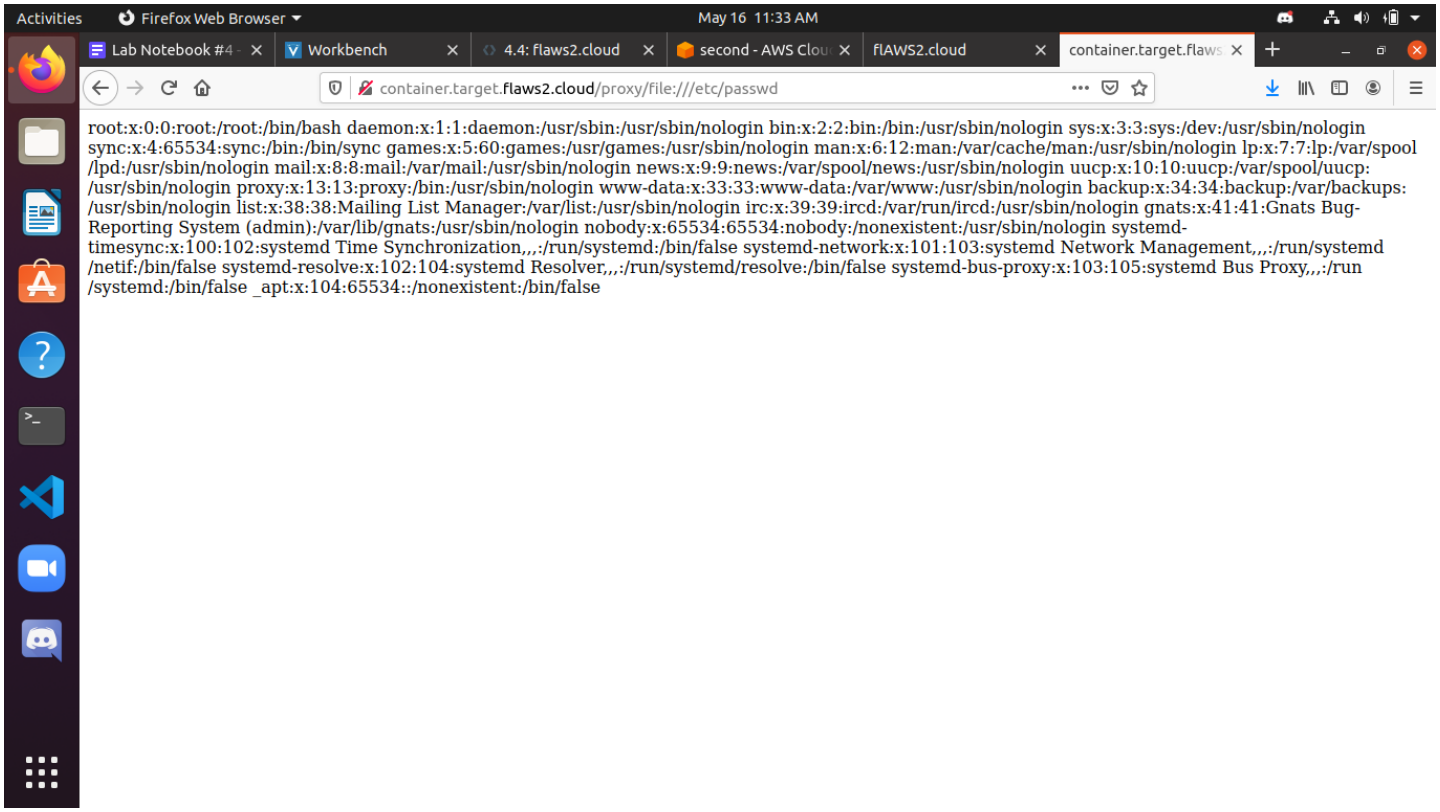


Take a screenshot of the successful login.

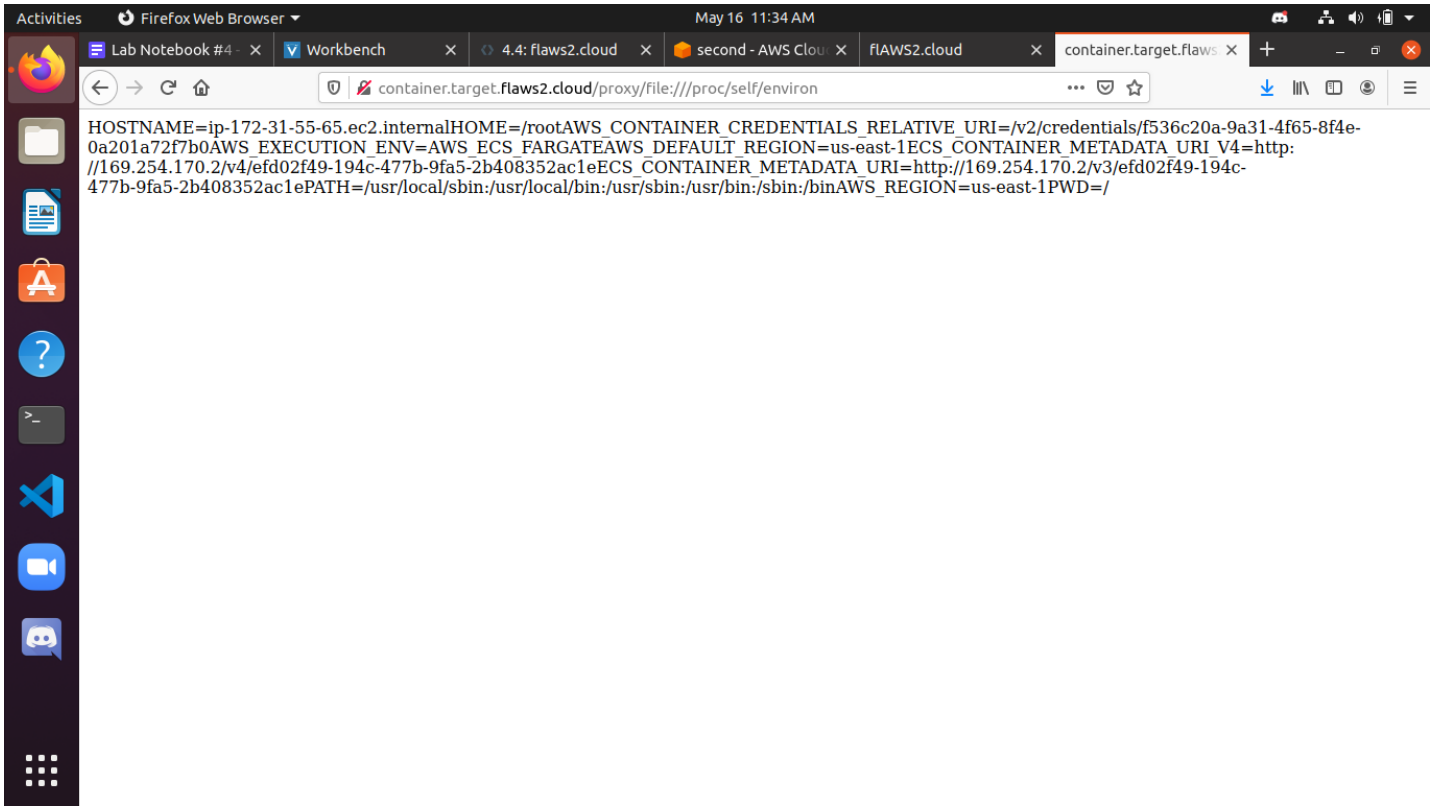


Level 3 Attacker

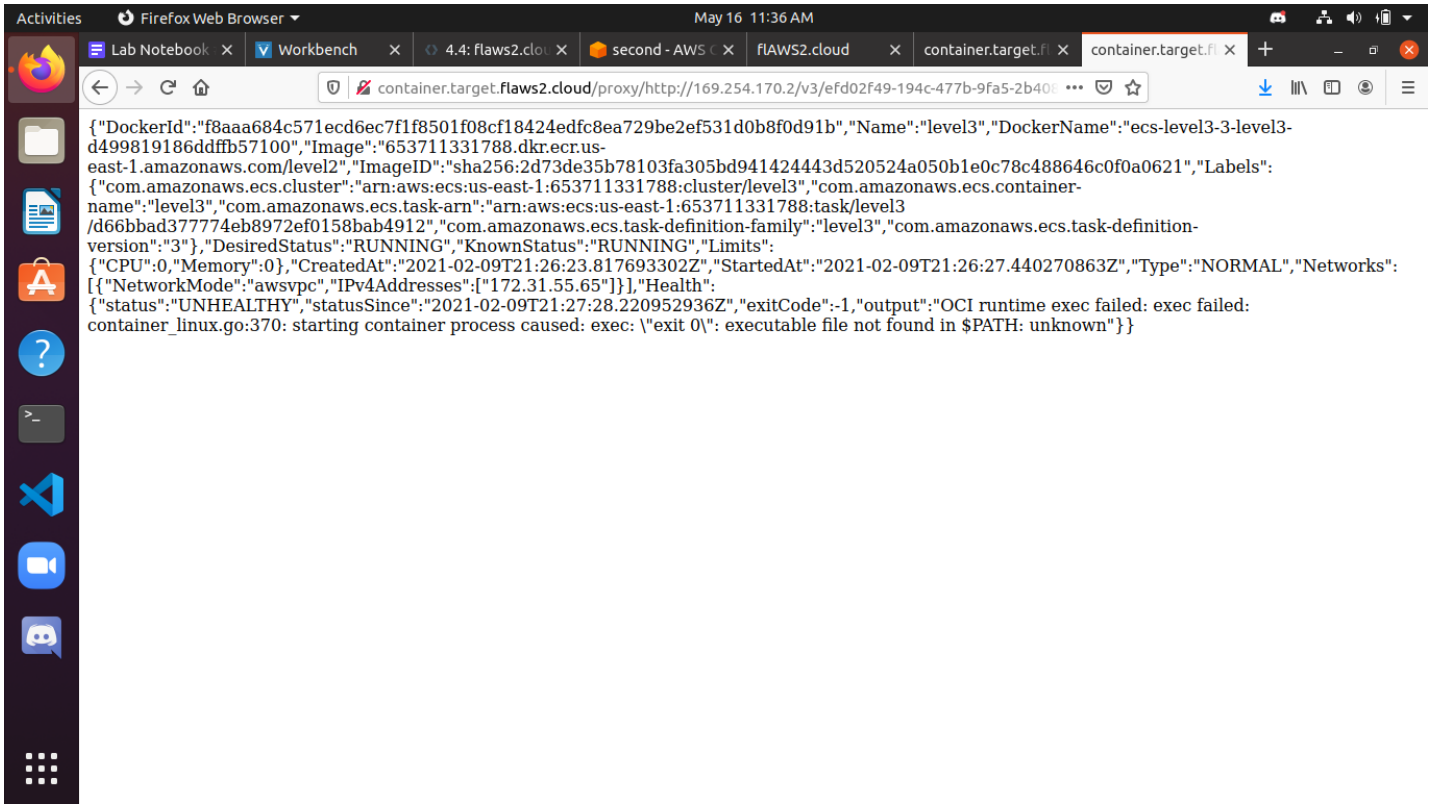
To test this, show the output when using http://container.target.flaws2.cloud/proxy/file:///etc/passwd



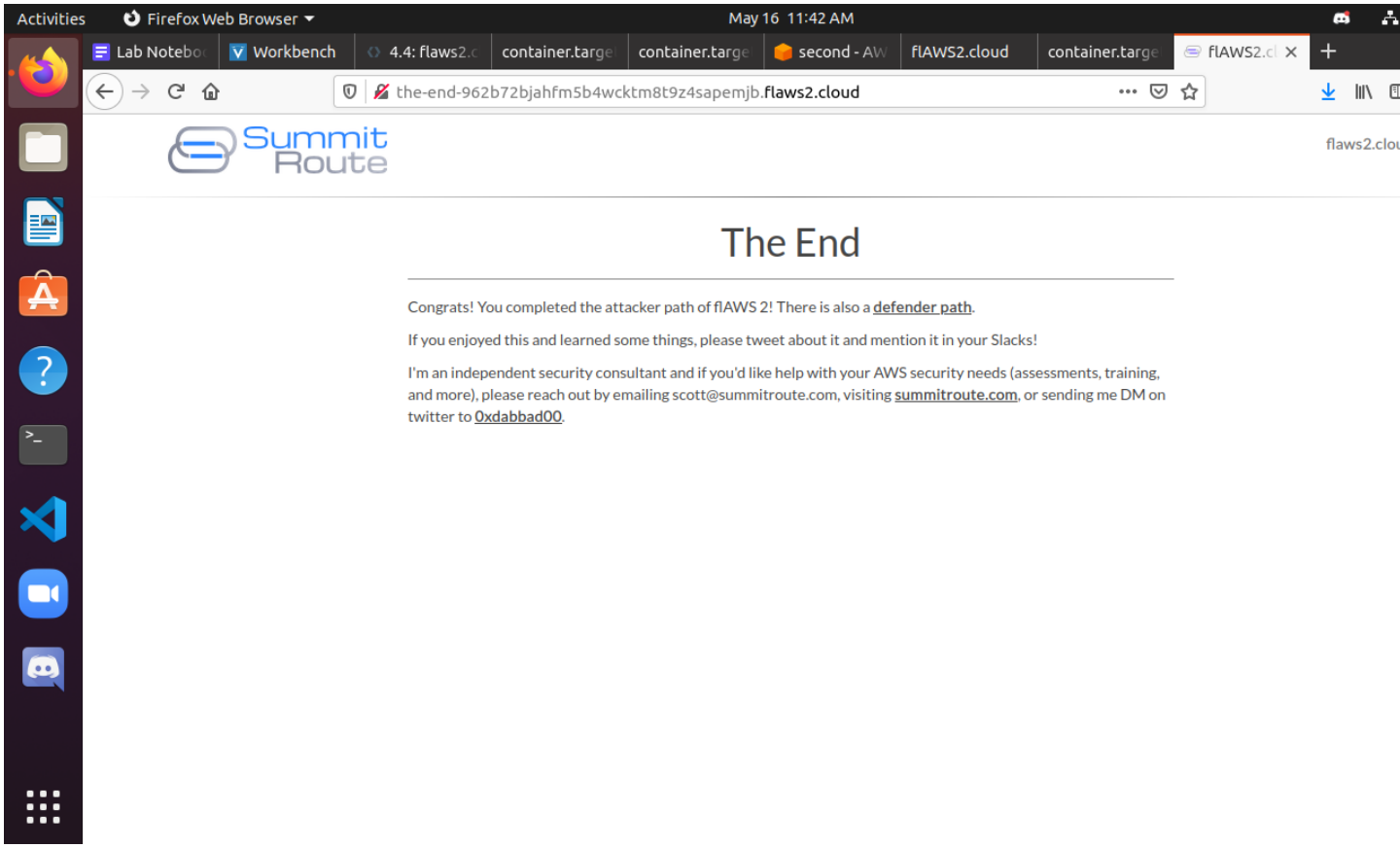
Show the environment variables for the process running the container. Make a note of the variables specific to AWS.



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

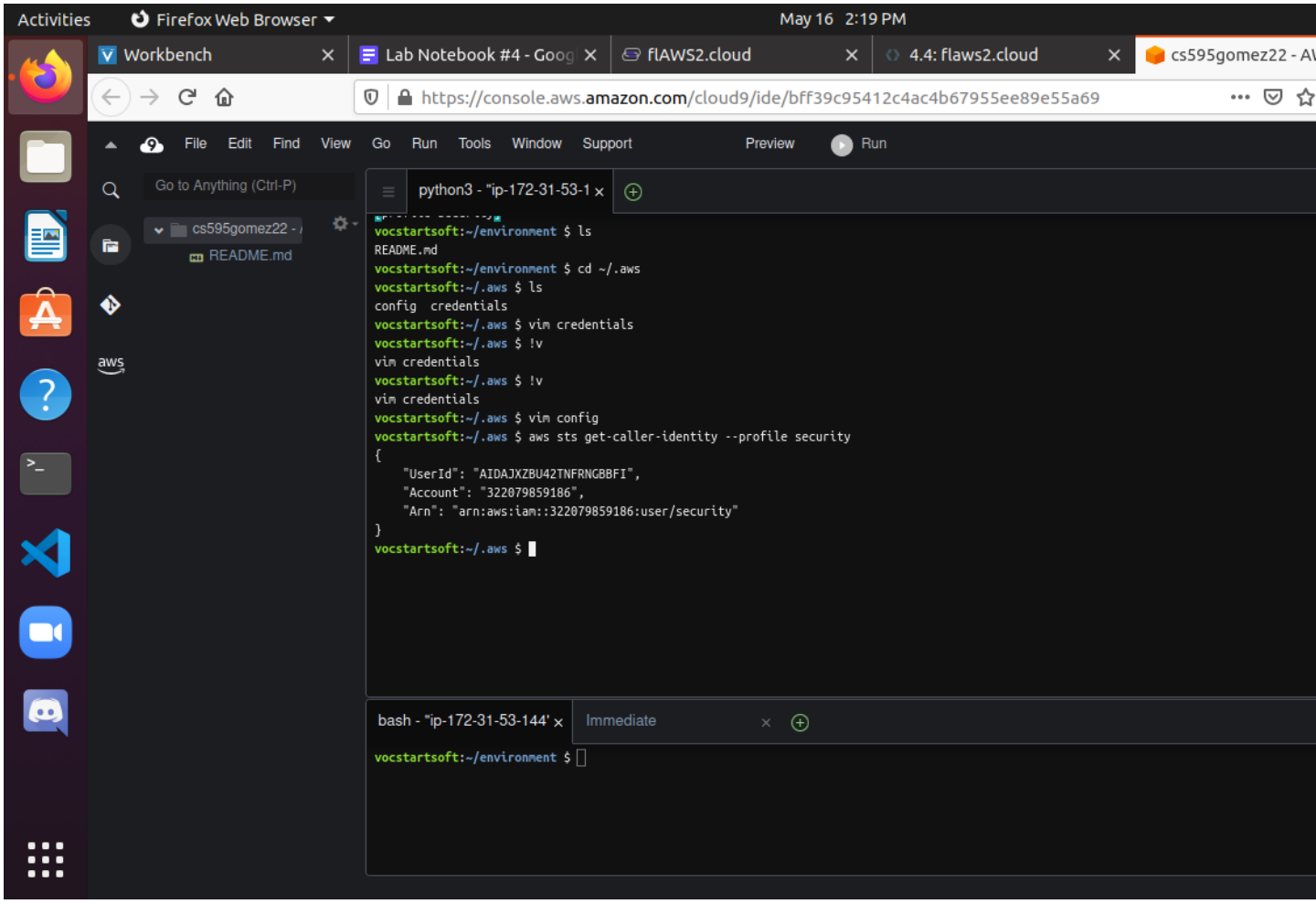


Visit the last URL and show a screenshot of the site.

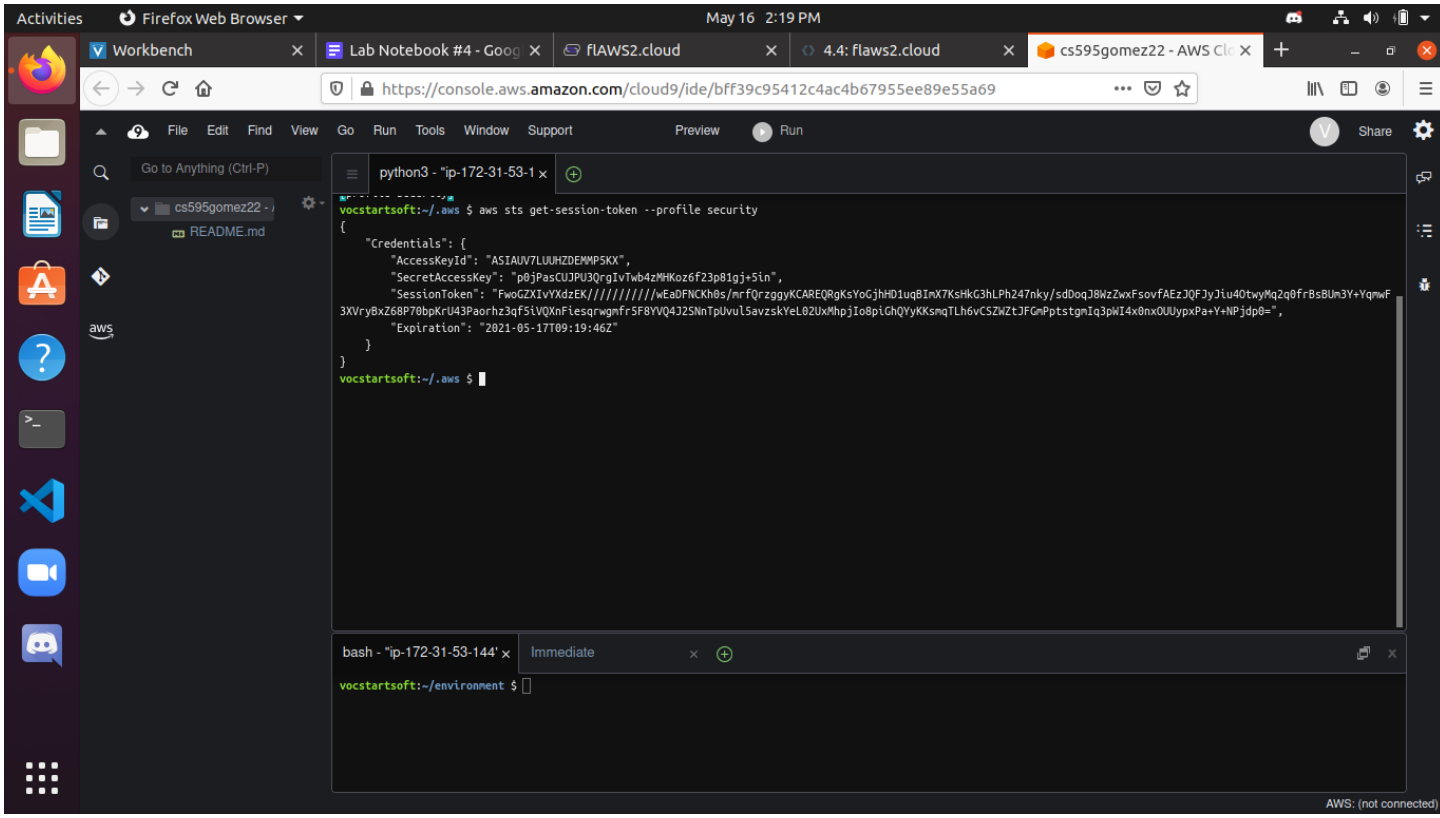


Flaws2.Cloud Defender

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

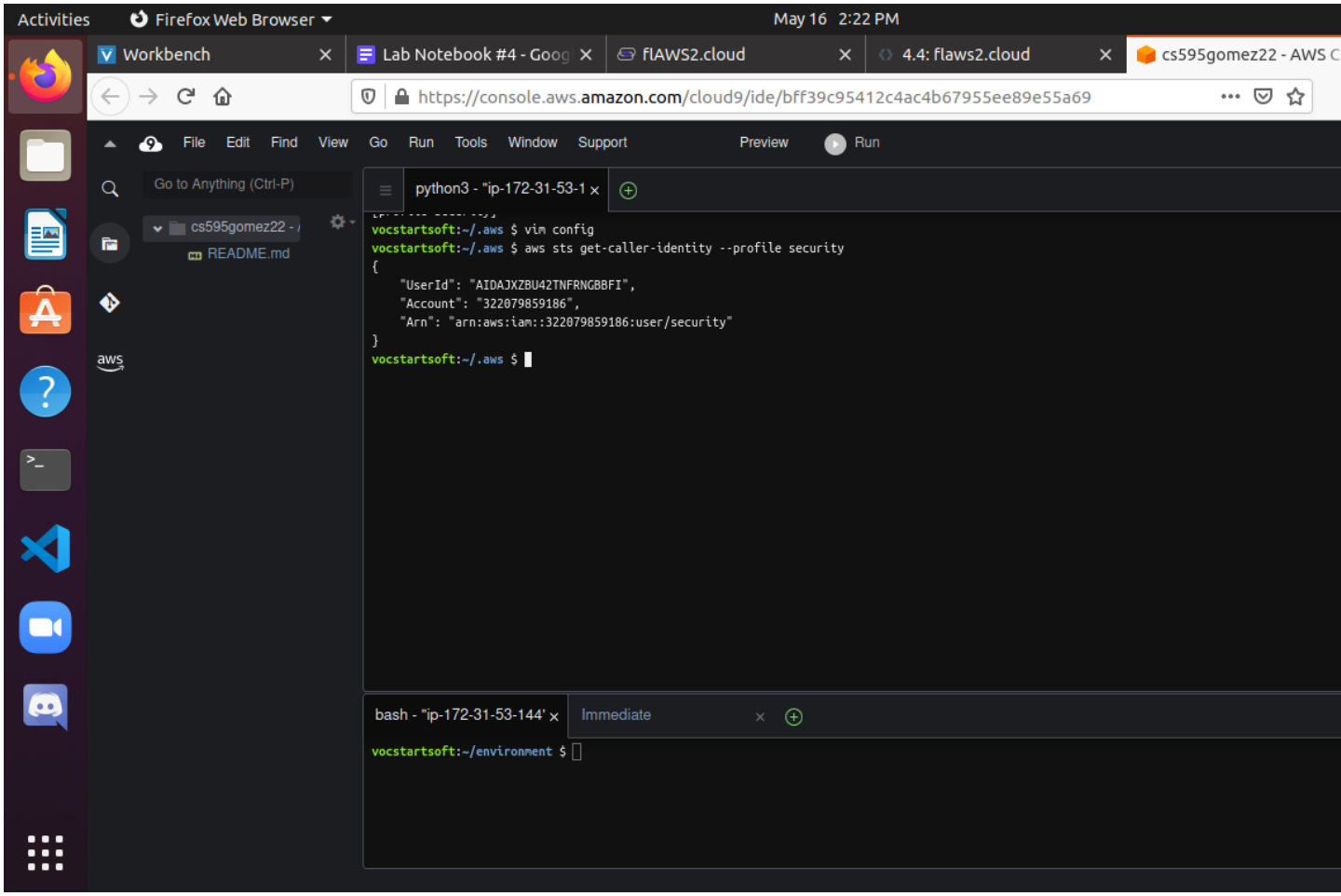


Show the token issued by using the command below
aws sts get-session-token --profile security

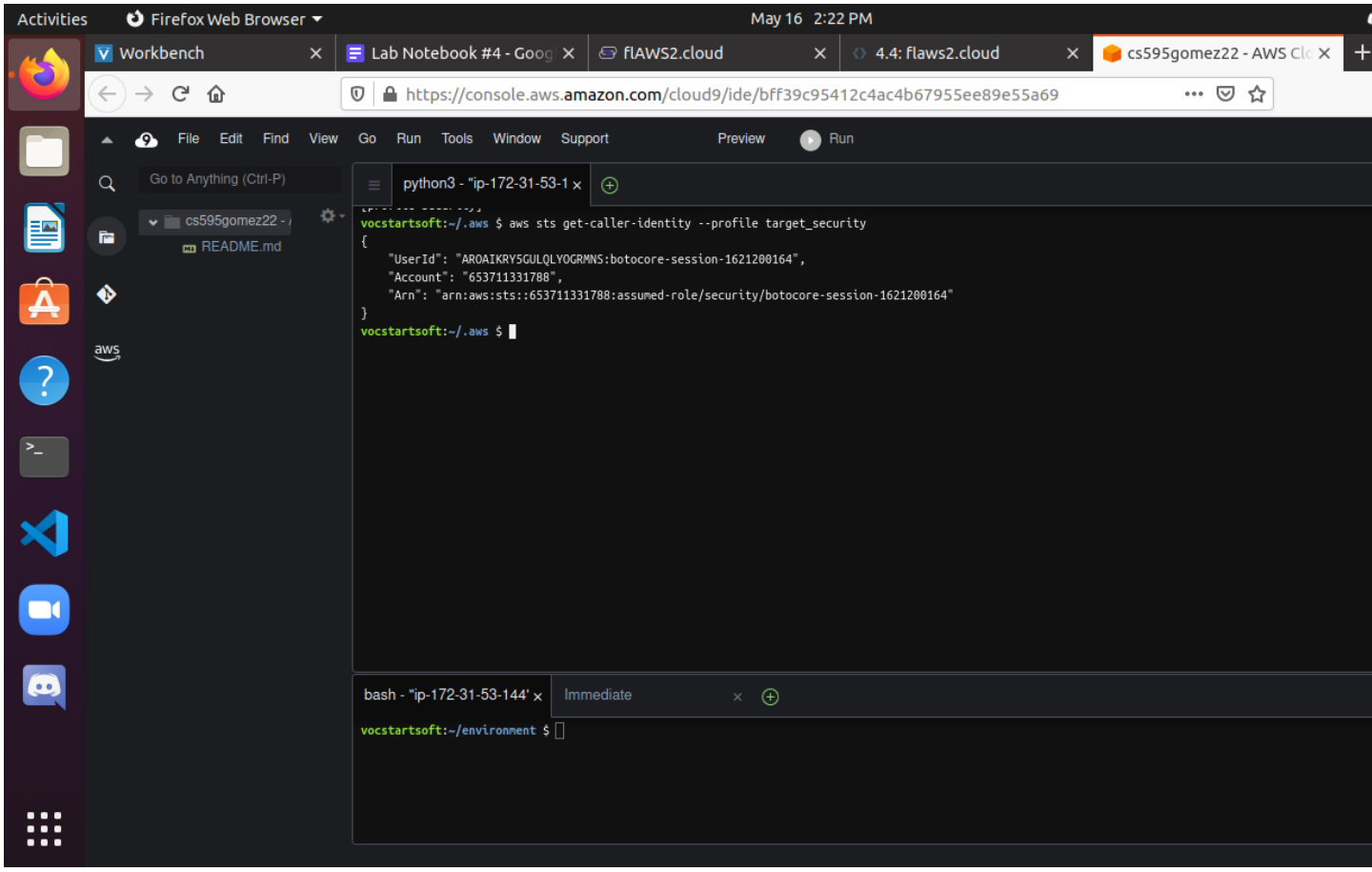


Show the output of the following commands that use STS to show what the profiles correspond to and the AWS accounts the roles assigned are associated with.

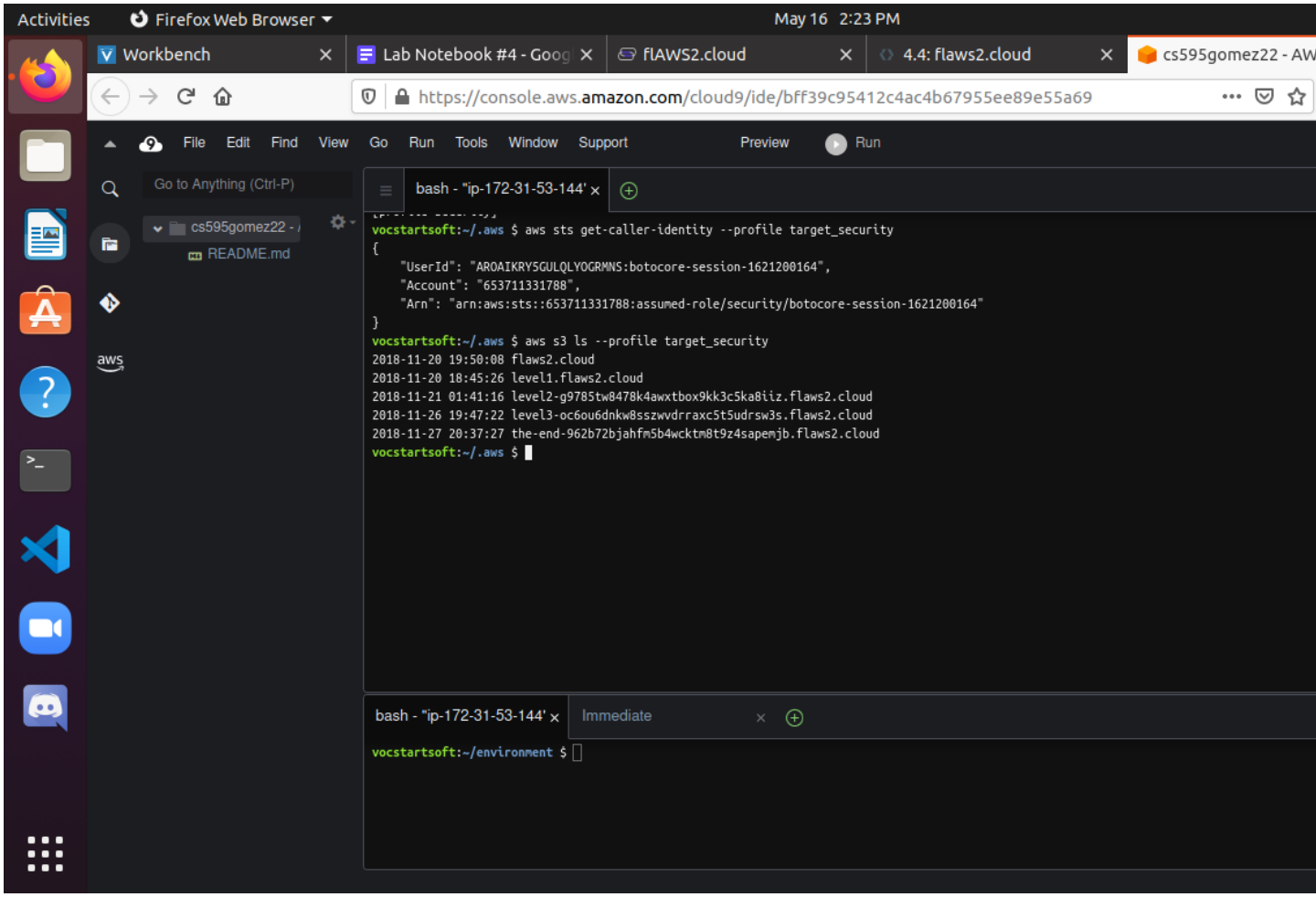
a) **aws sts get-caller-identity --profile security**



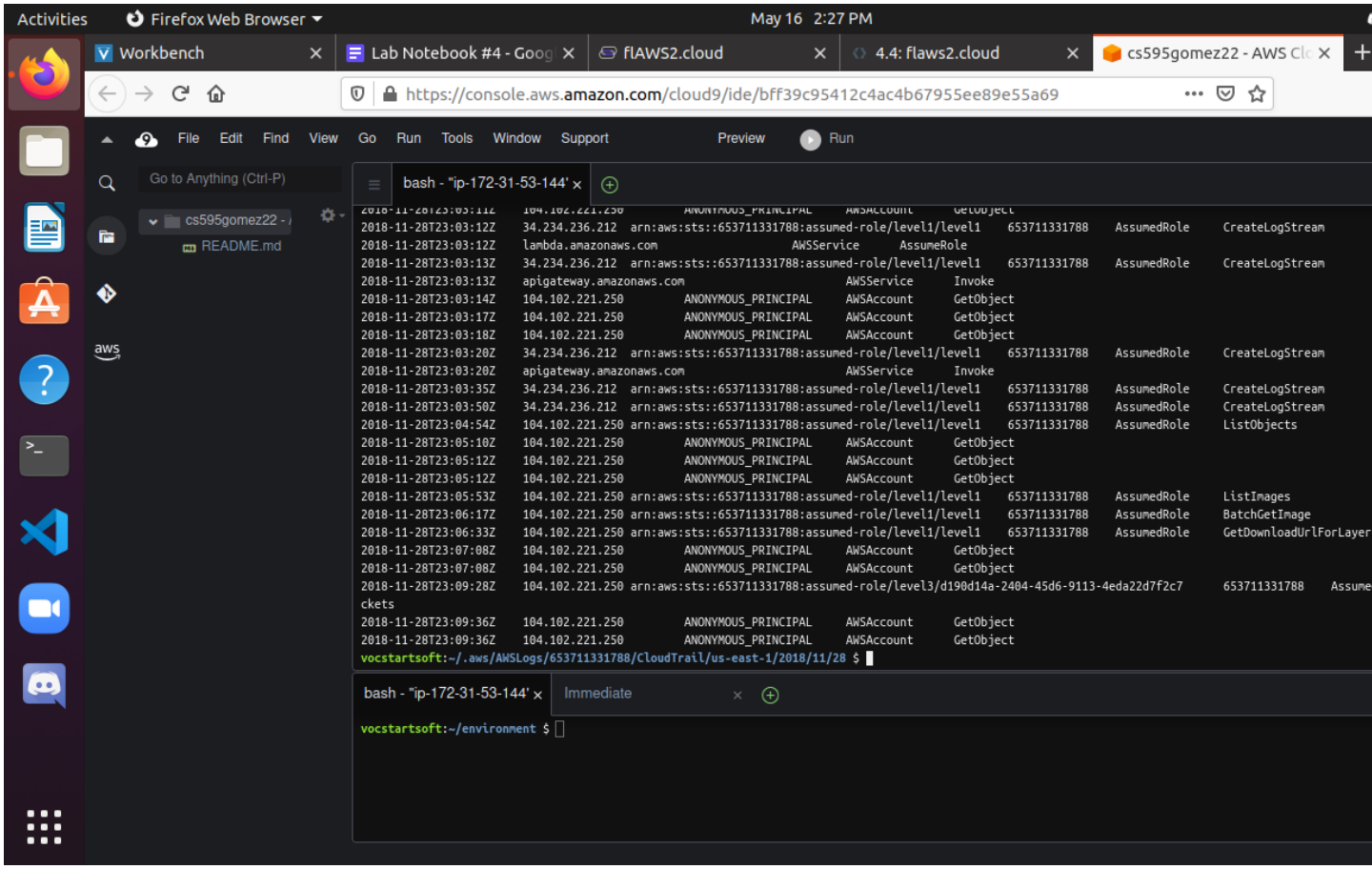
b) aws sts get-caller-identity --profile target_security



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

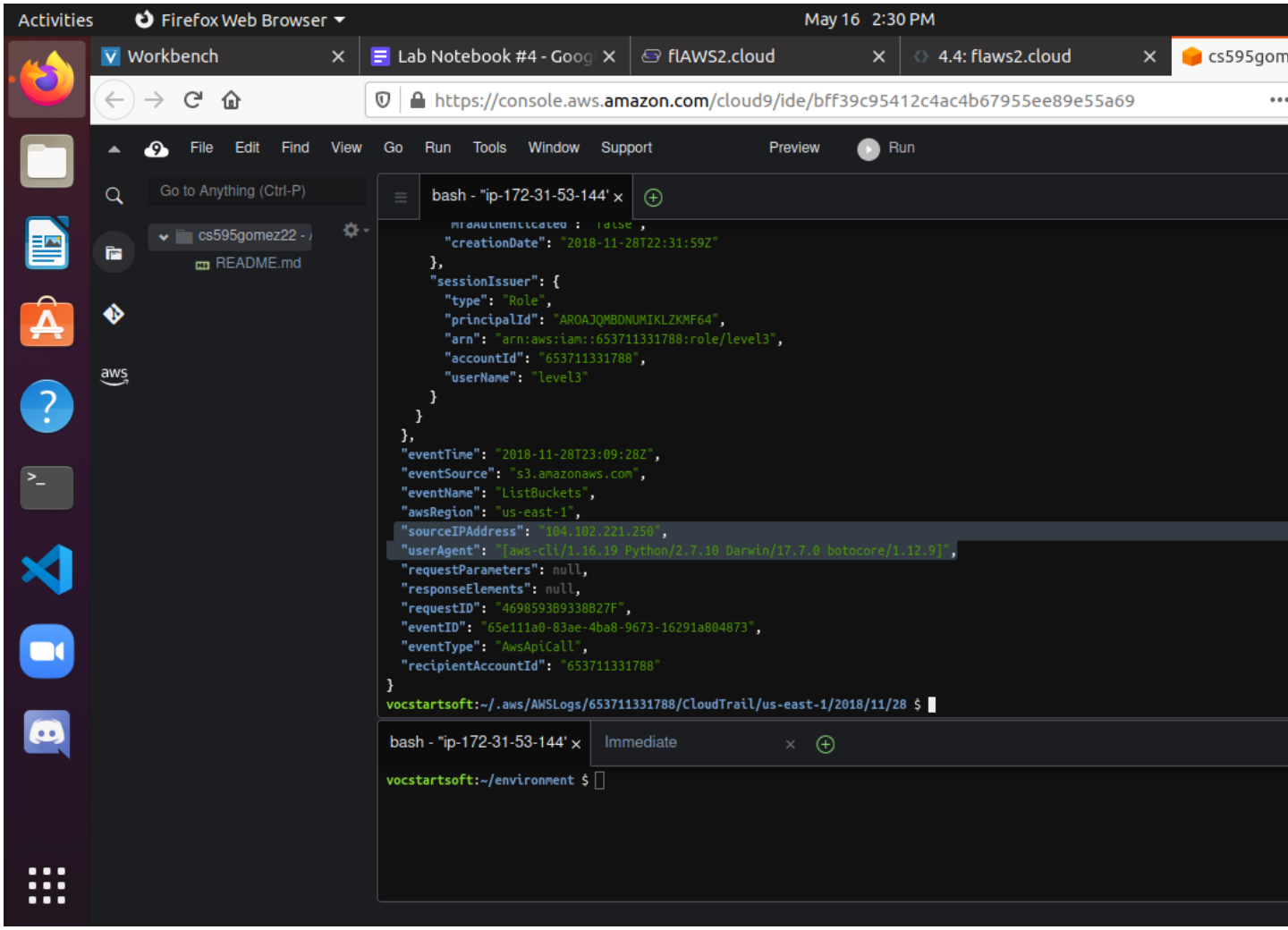


Finally, we can output identity and network address information to more fully detail the events. Show the last 10 output lines of the following: cat *.json | jq -cr '.Records[]|[.eventTime, .sourceIPAddress, .userIdentity.arn, .userIdentity.accountId, .userIdentity.type, .eventName]|@tsv' | sort



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).



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?

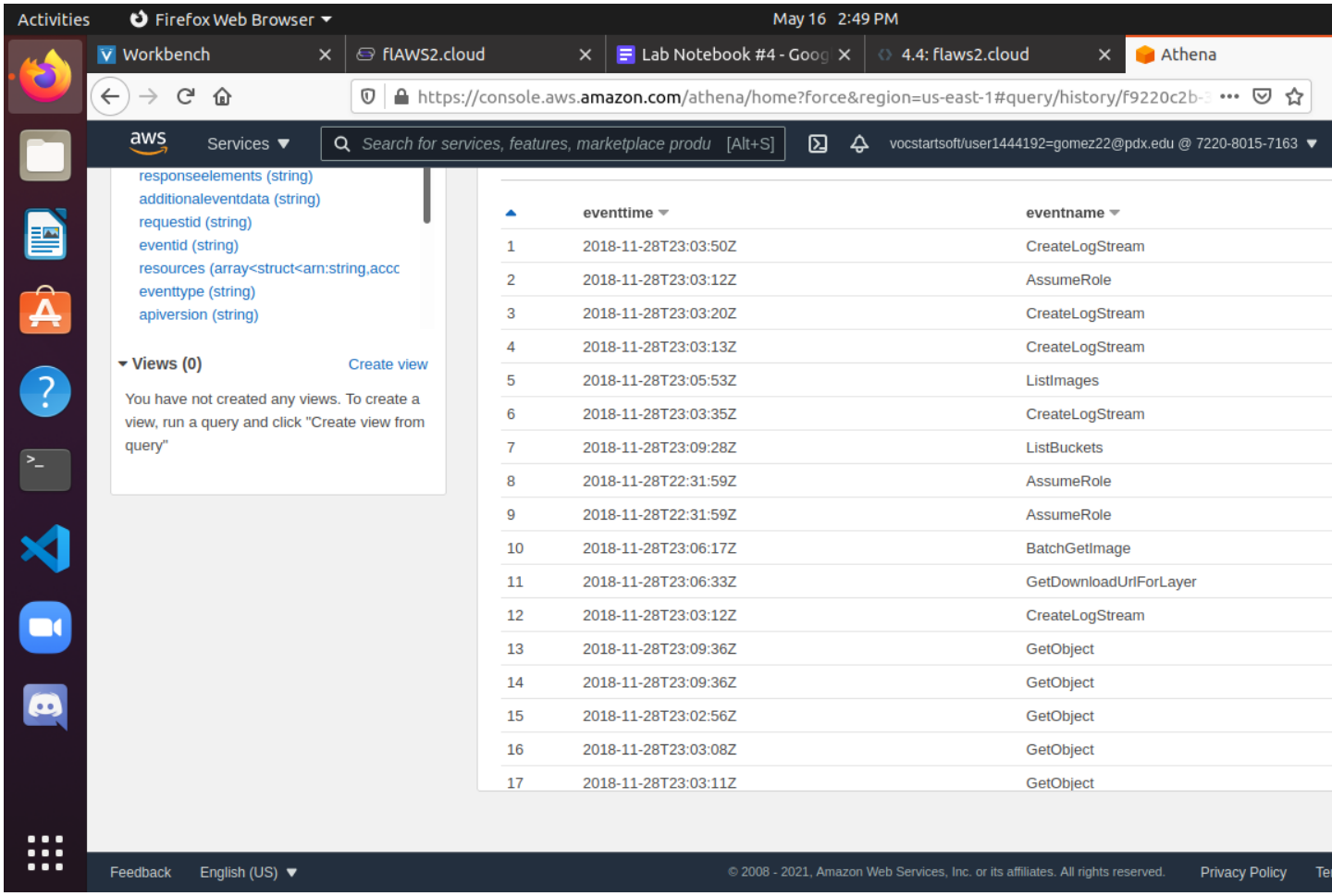
-The service is supposed to be "ecs-tasks.amazonaws.com". No it is not compatible with the userAgent in the last step.

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

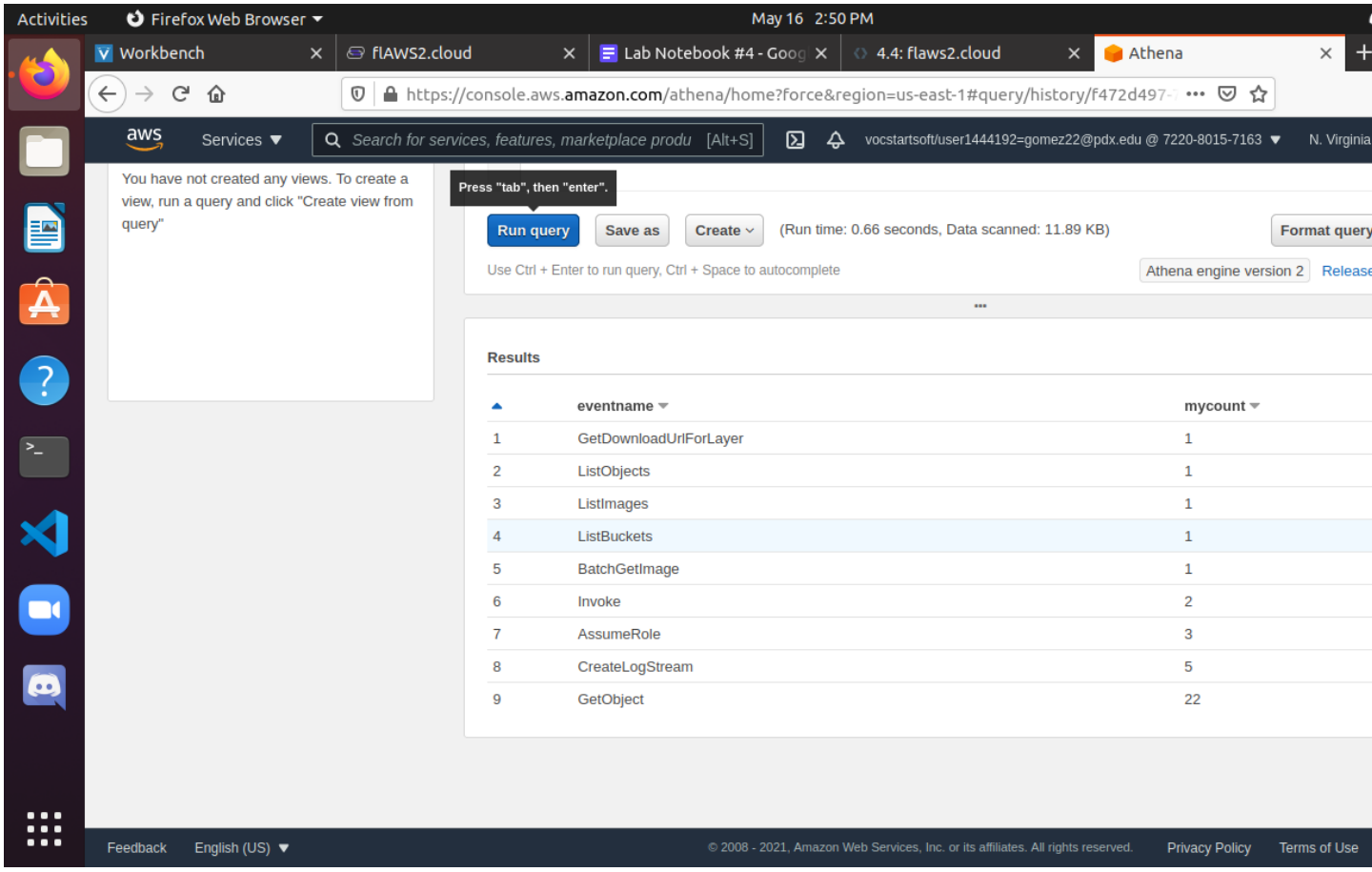
-Anyone is allowed to perform “GetDownloadUrlForLayer”, “BatchGetImage”, “BatchCheckLayerAvailability”, “ListImages”, and

“DescribeImages” 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

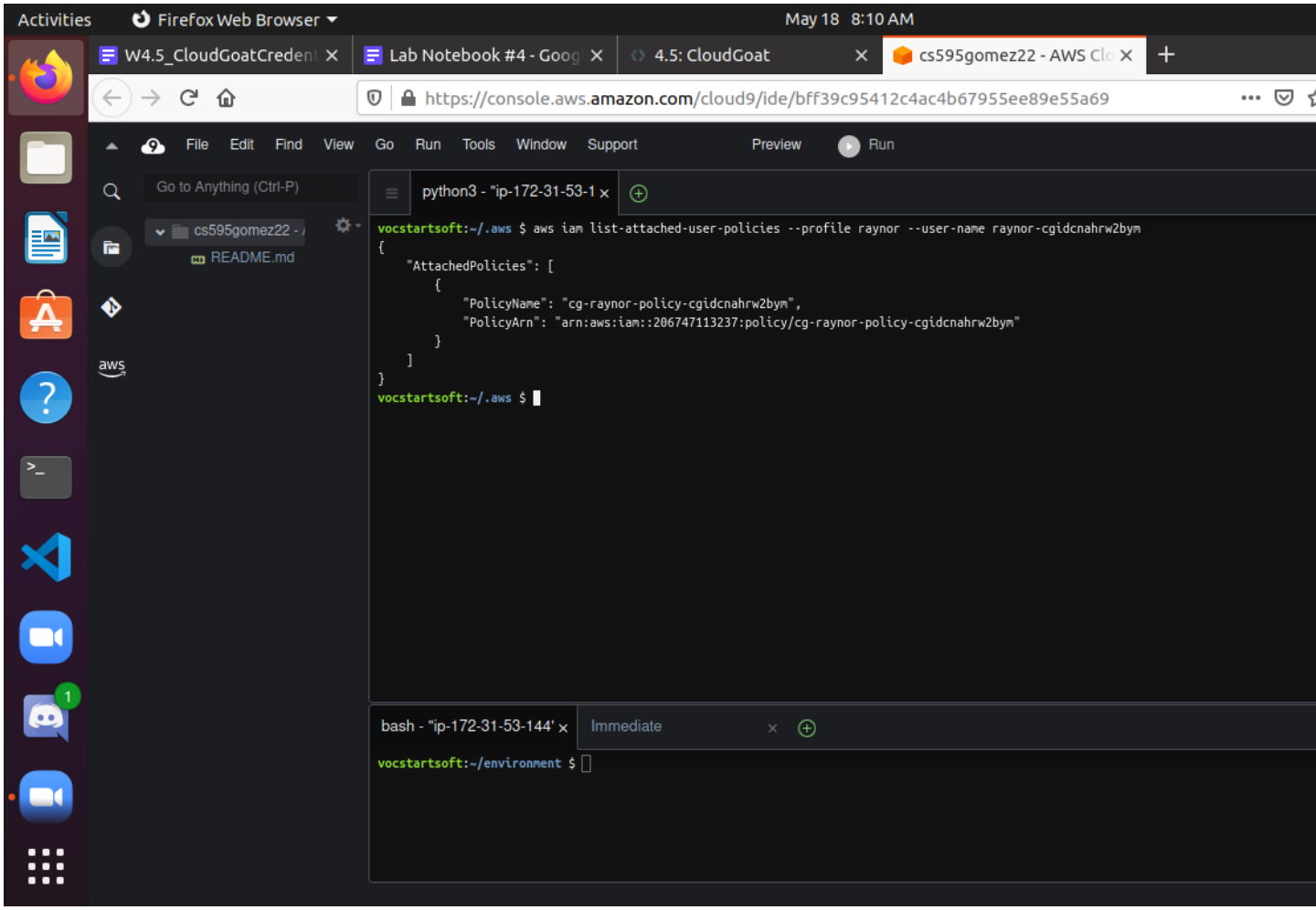


Show the output of the query.



4.5 Cloud Goat

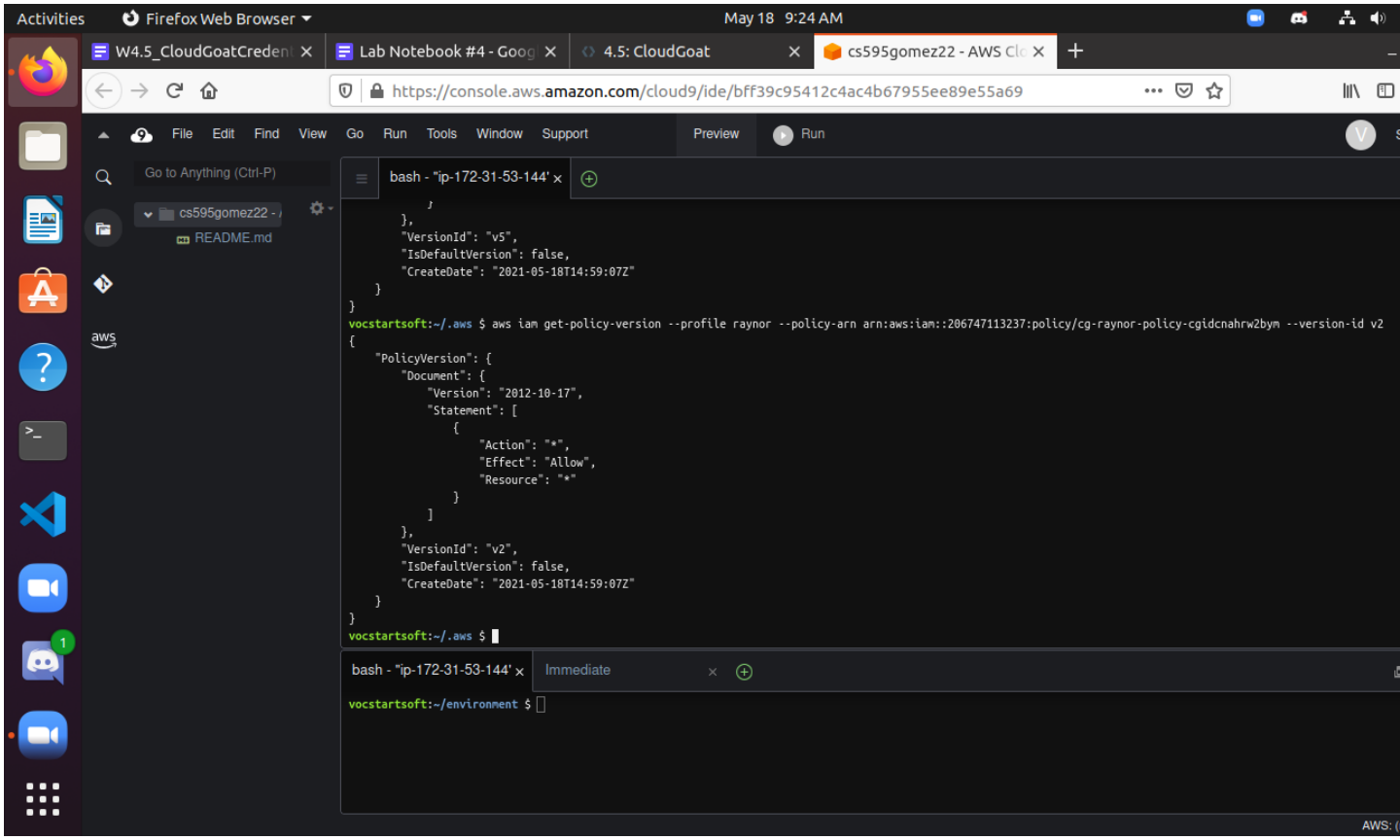
Show the policies attached to the credentials given



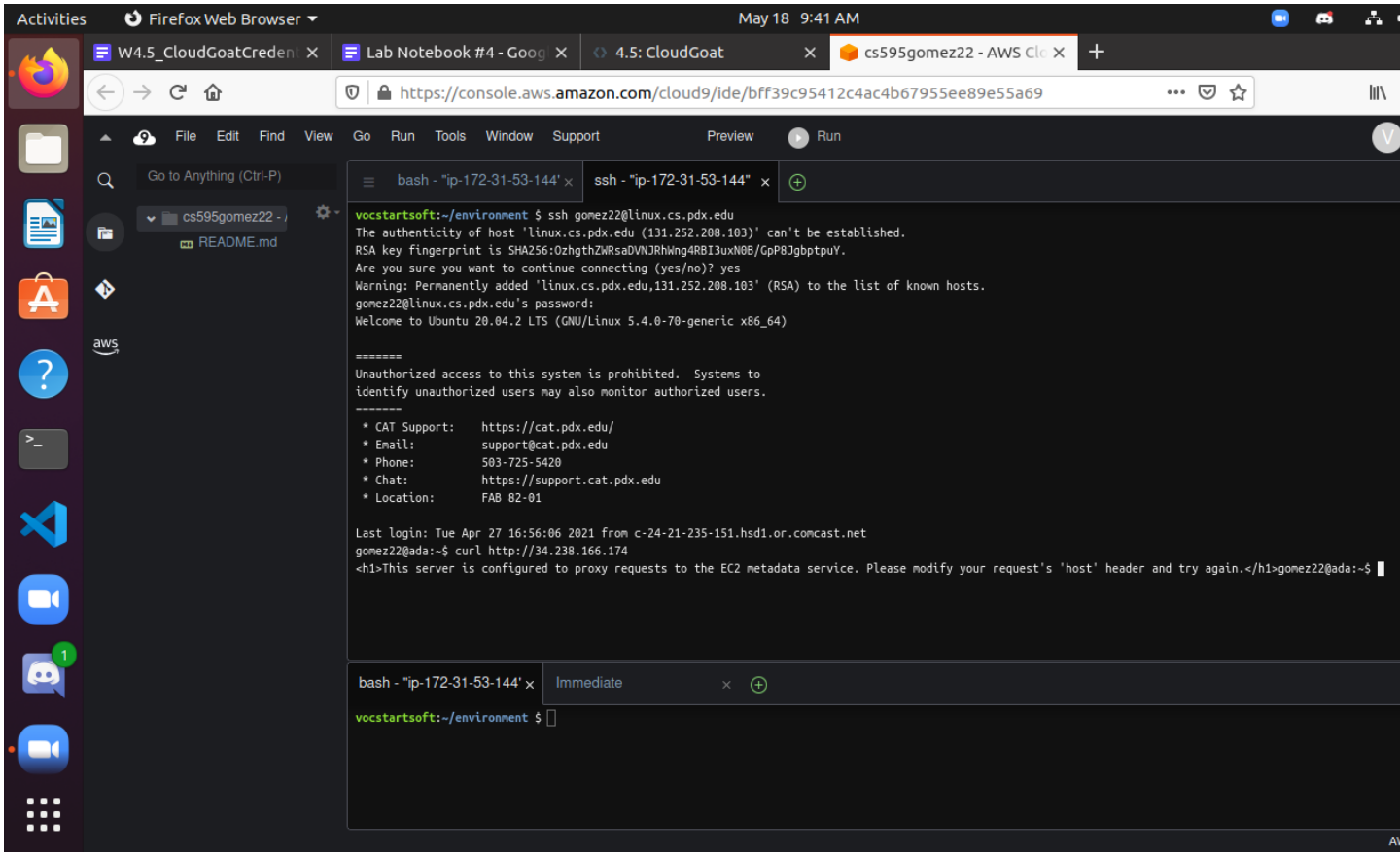
Which version of the policy is set as the default?

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

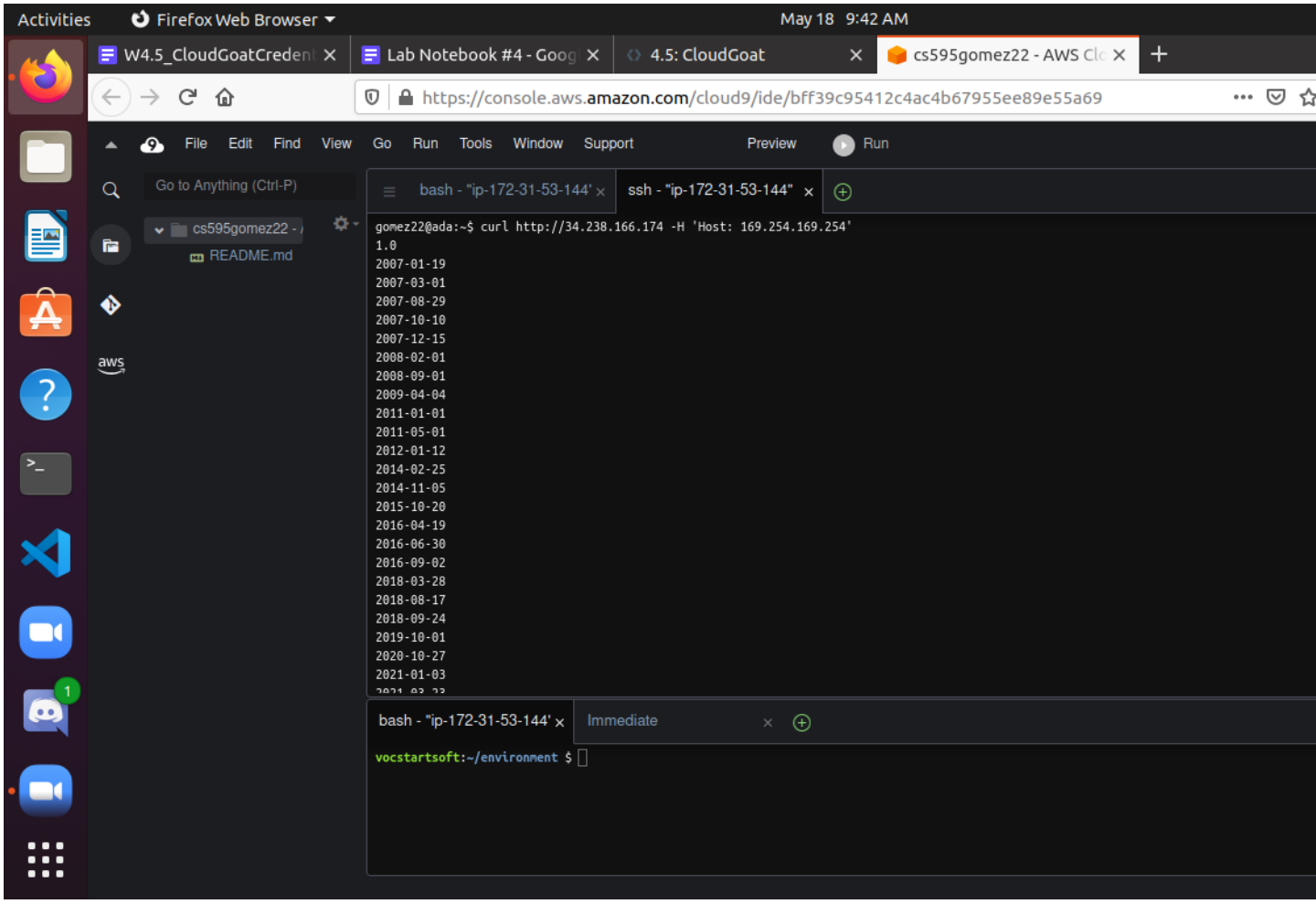
Show the output of the version in which all actions have been allowed (e.g full admin privileges)



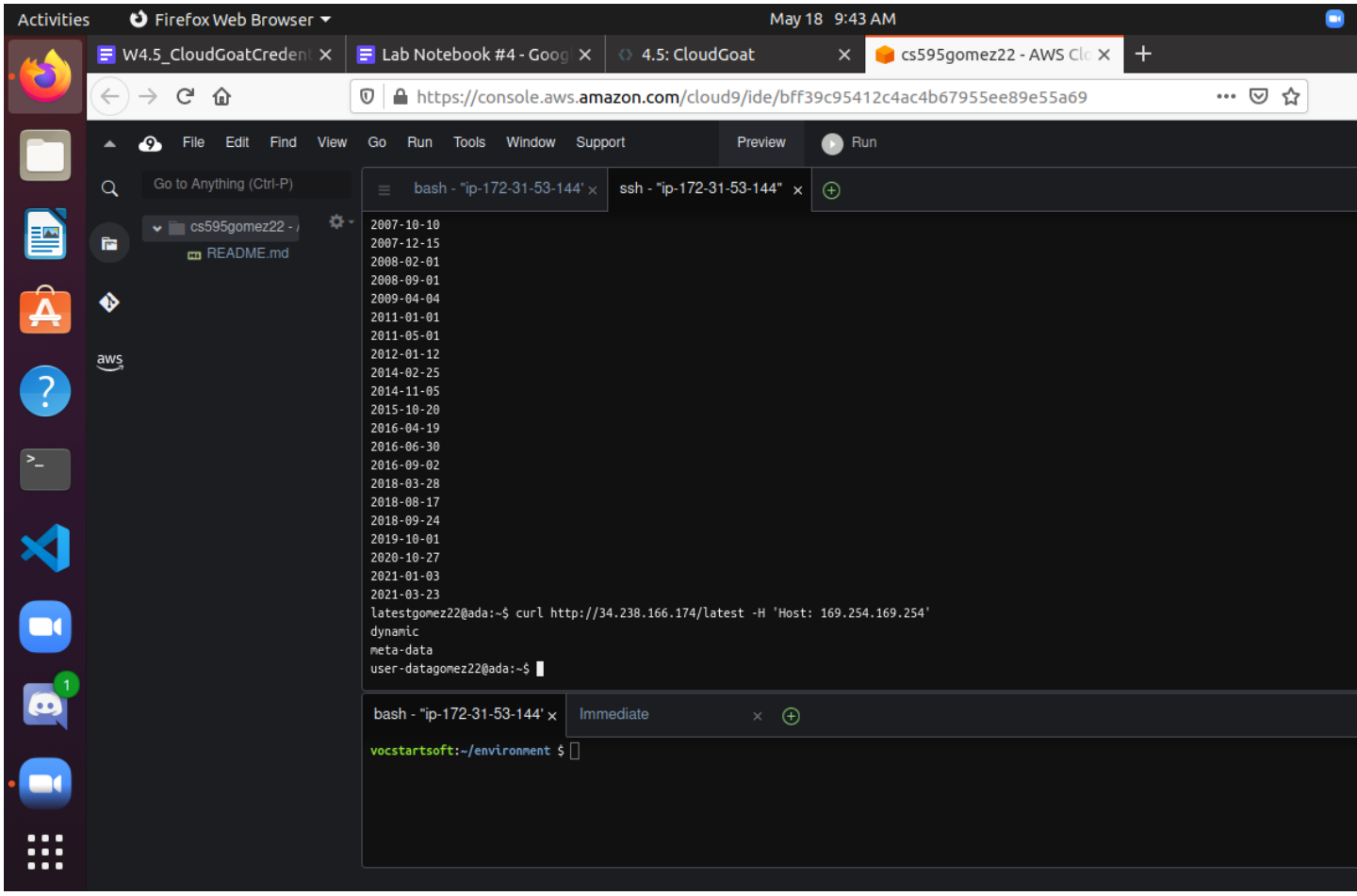
Start by visiting the site from linux.cs.pdx.edu using curl to hit the web server and show the error page returned.



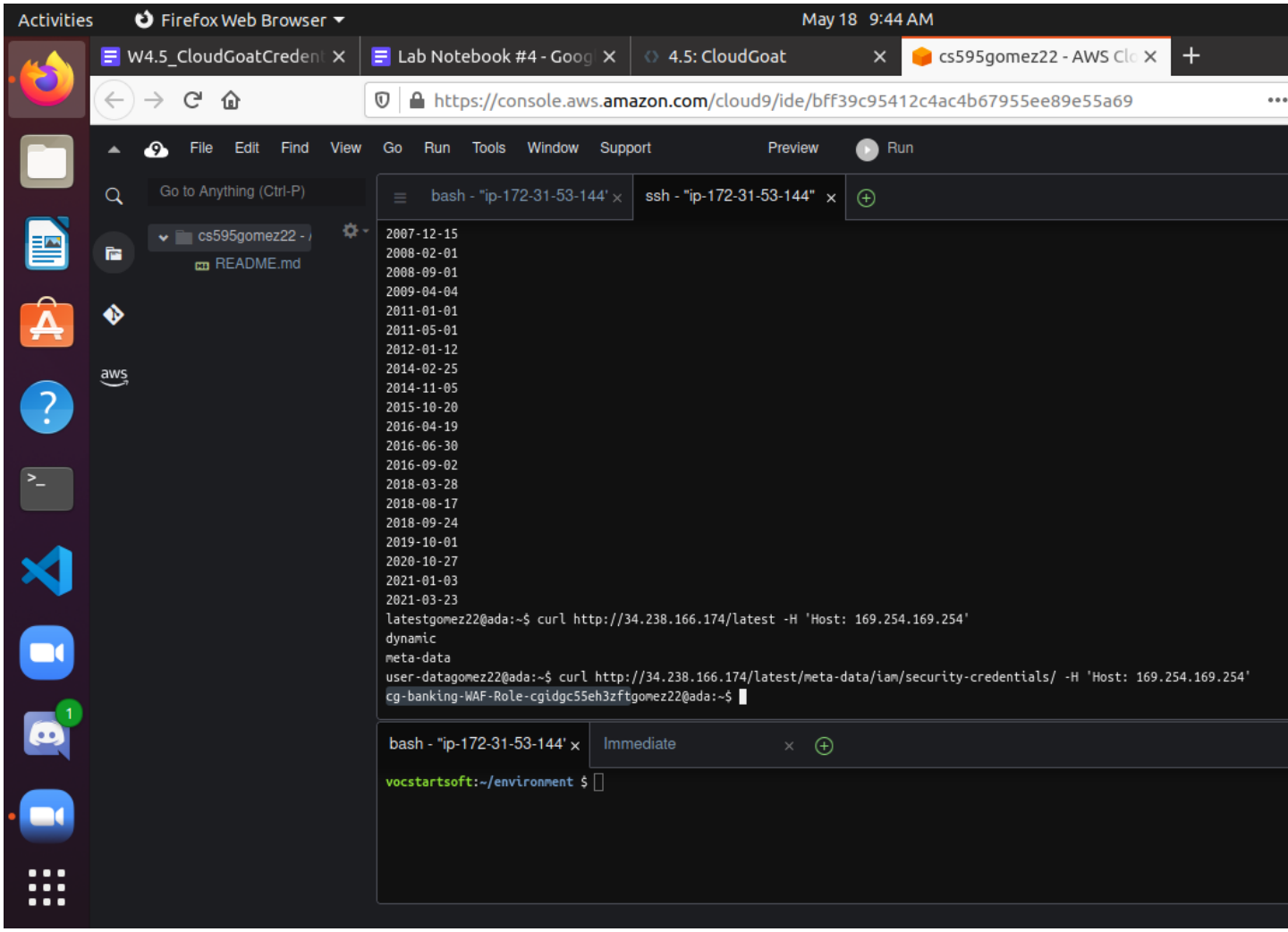
Repeat the curl command and specify the well-known IP address of the Metadata service. Show the results.



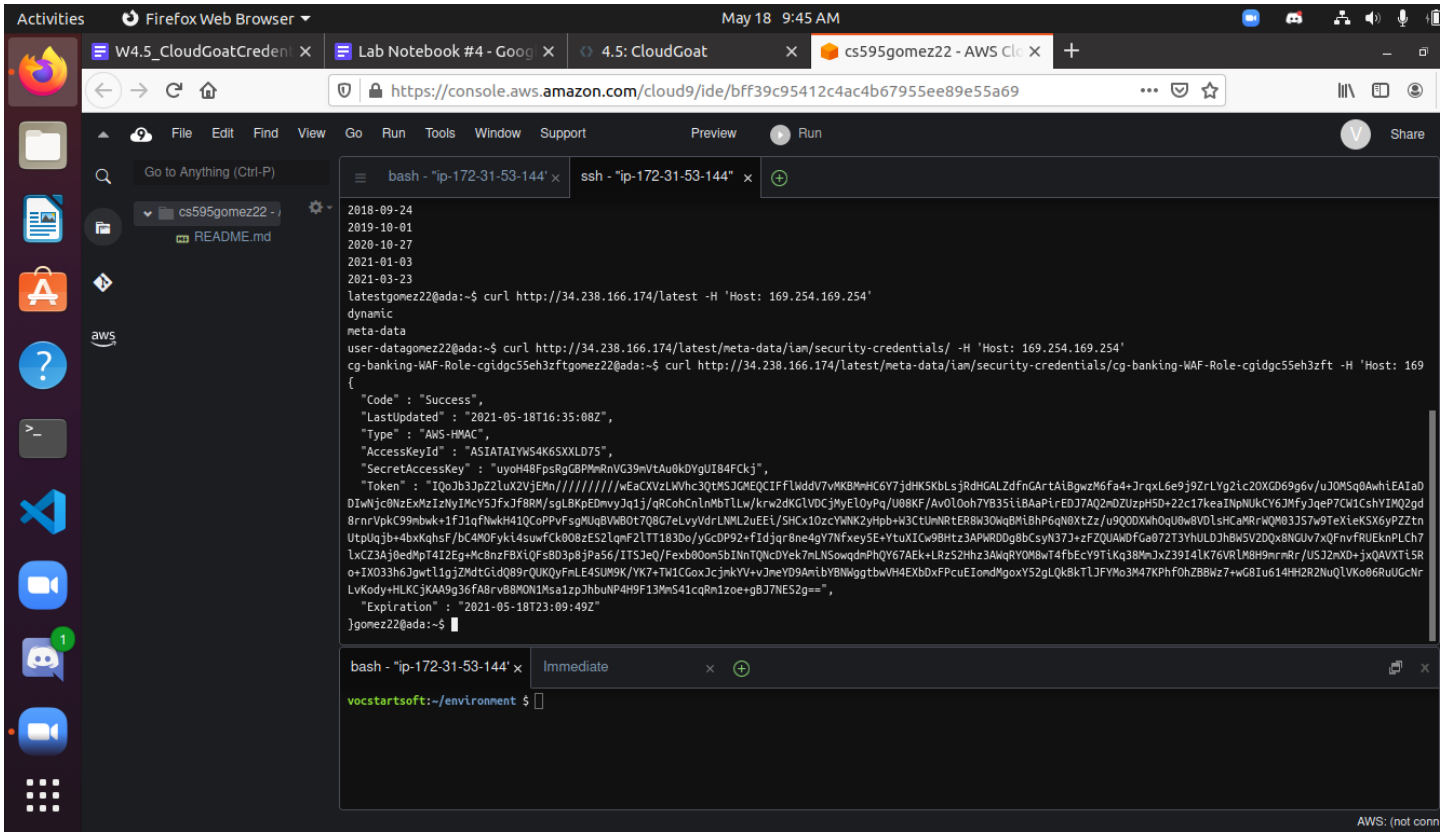
Show its contents via the following command: curl http://<ec2_instance_IP>/latest -H 'Host: 169.254.169.254'



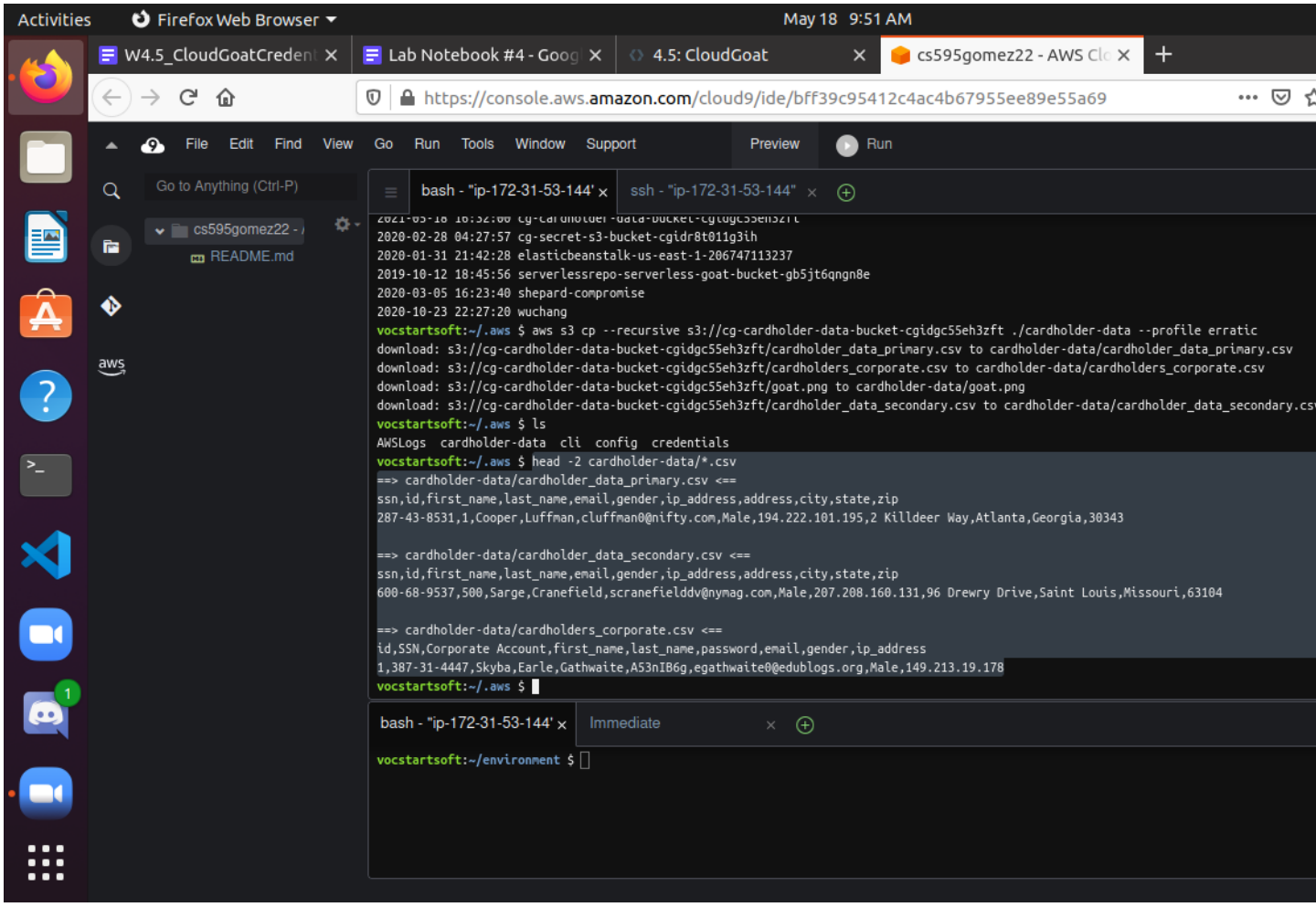
Show the name of the AWS role the following command exposes: curl http://<ec2_instance_IP>/latest/meta-data/iam/security-credentials/ -H 'Host: 169.254.16



Then, show the credentials associated with the role.

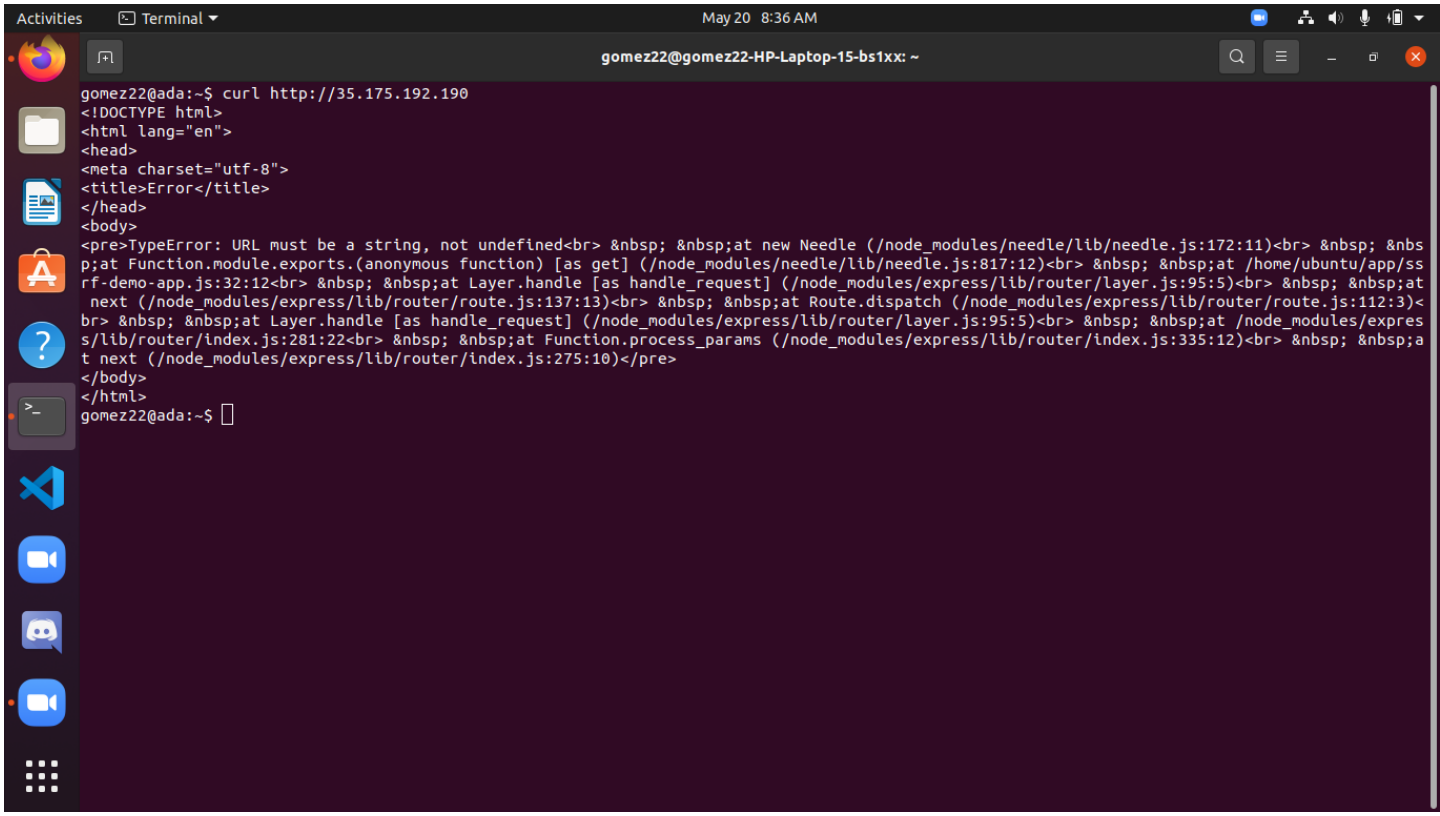


Show the first two lines of each of the CSV files you have copied over from the bucket via the command below.

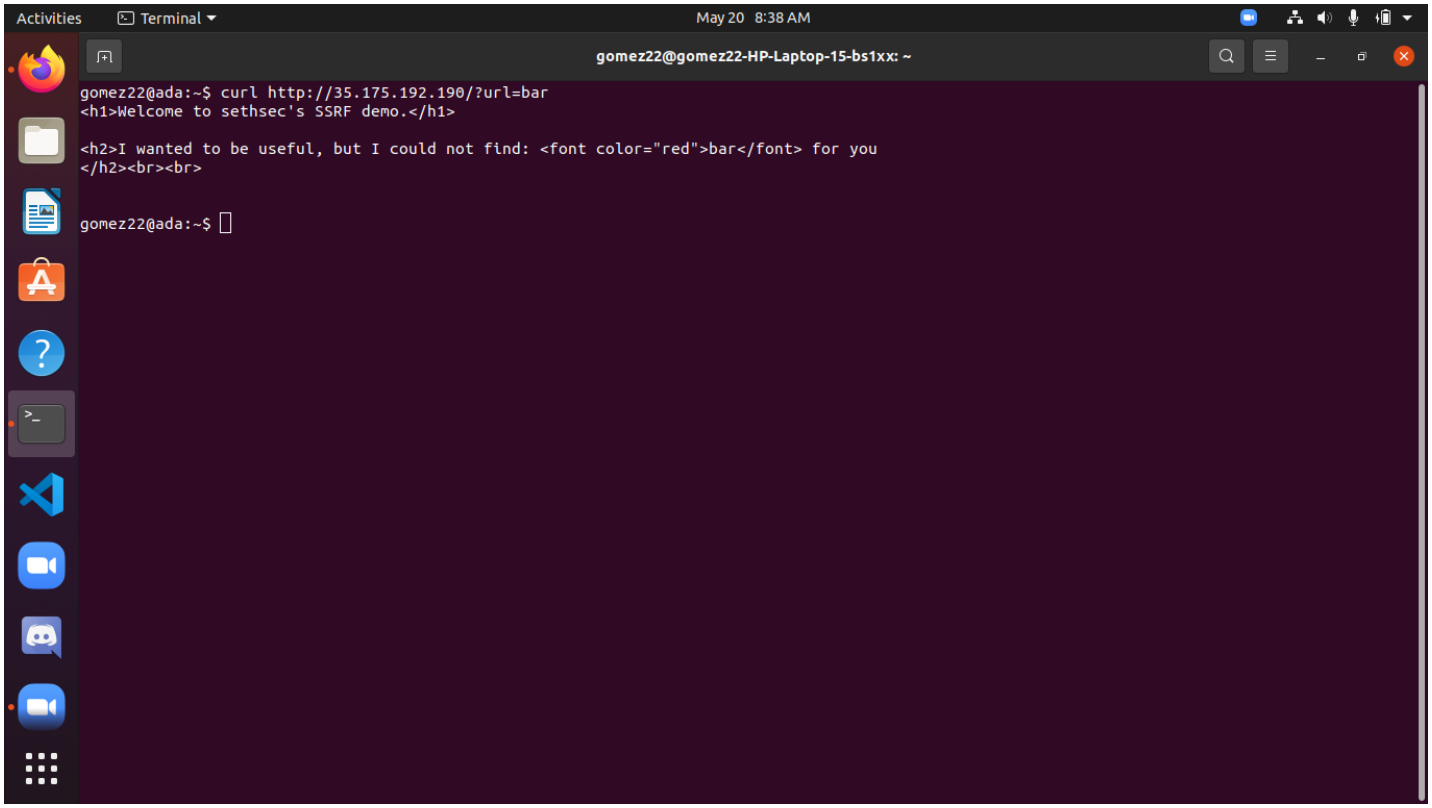


Ec2_ssrf

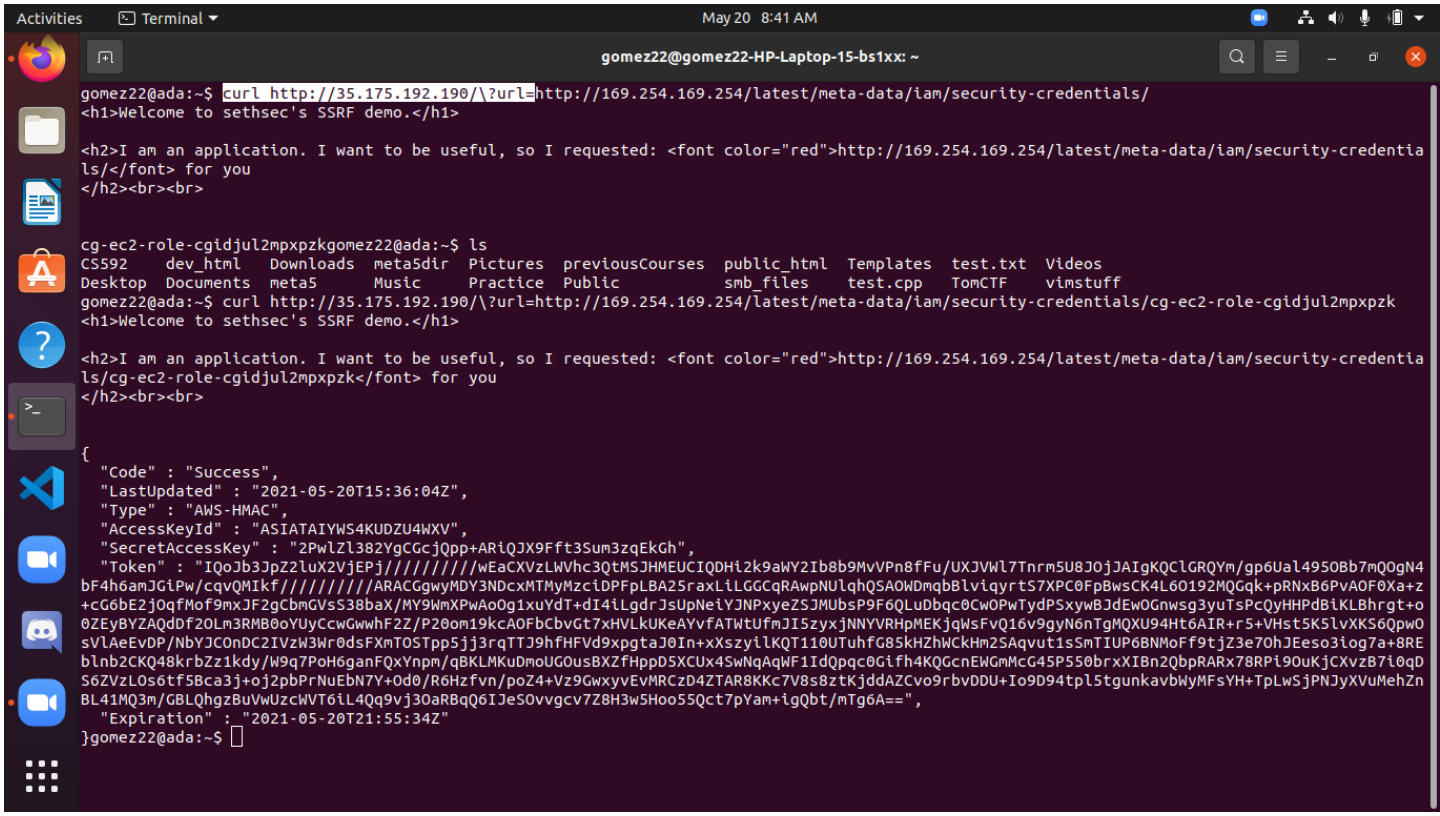
Take a screenshot of the page that is returned.



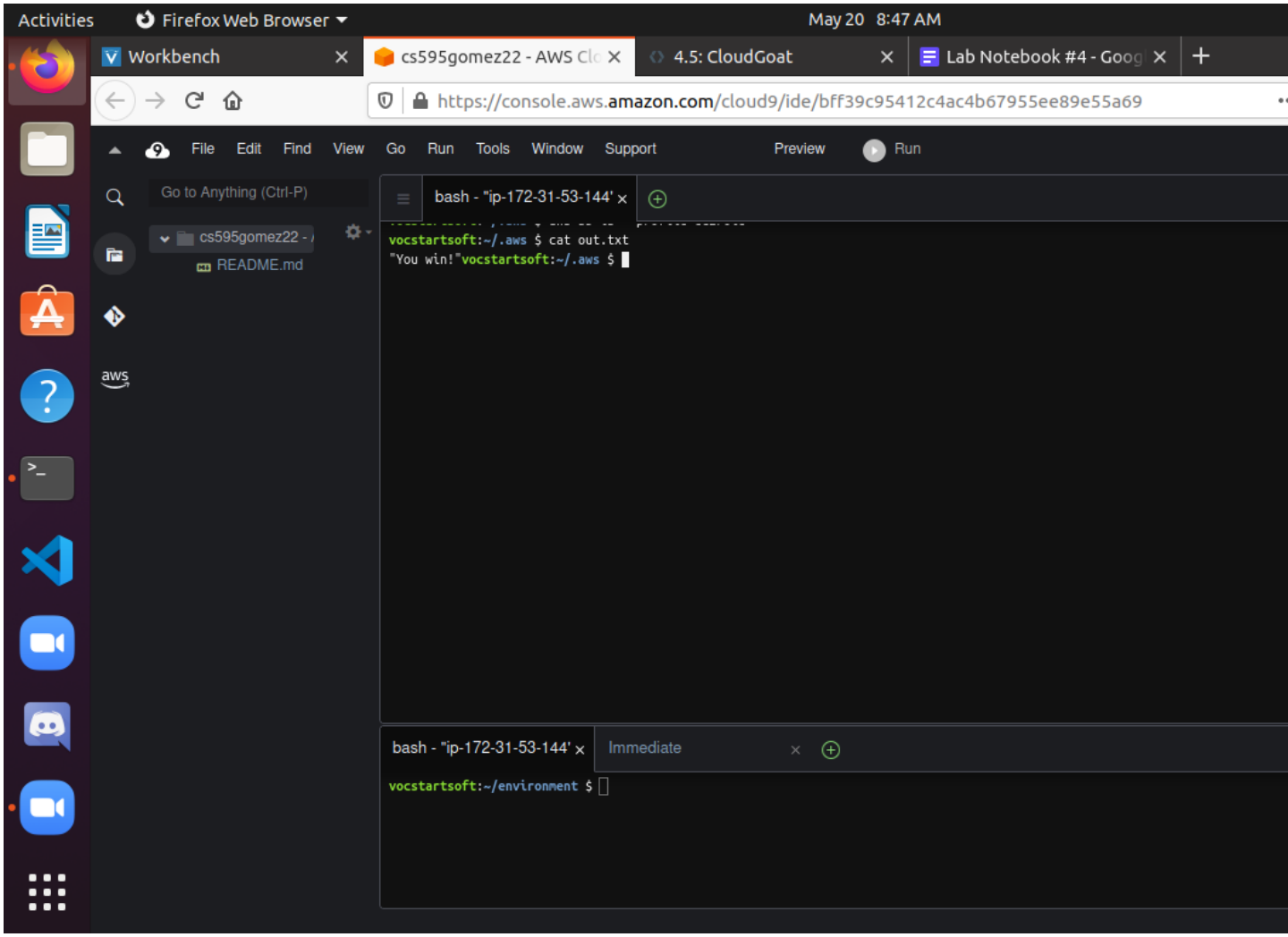
Take a screenshot of the page that is returned.



Take a screenshot showing the information associated with the role.

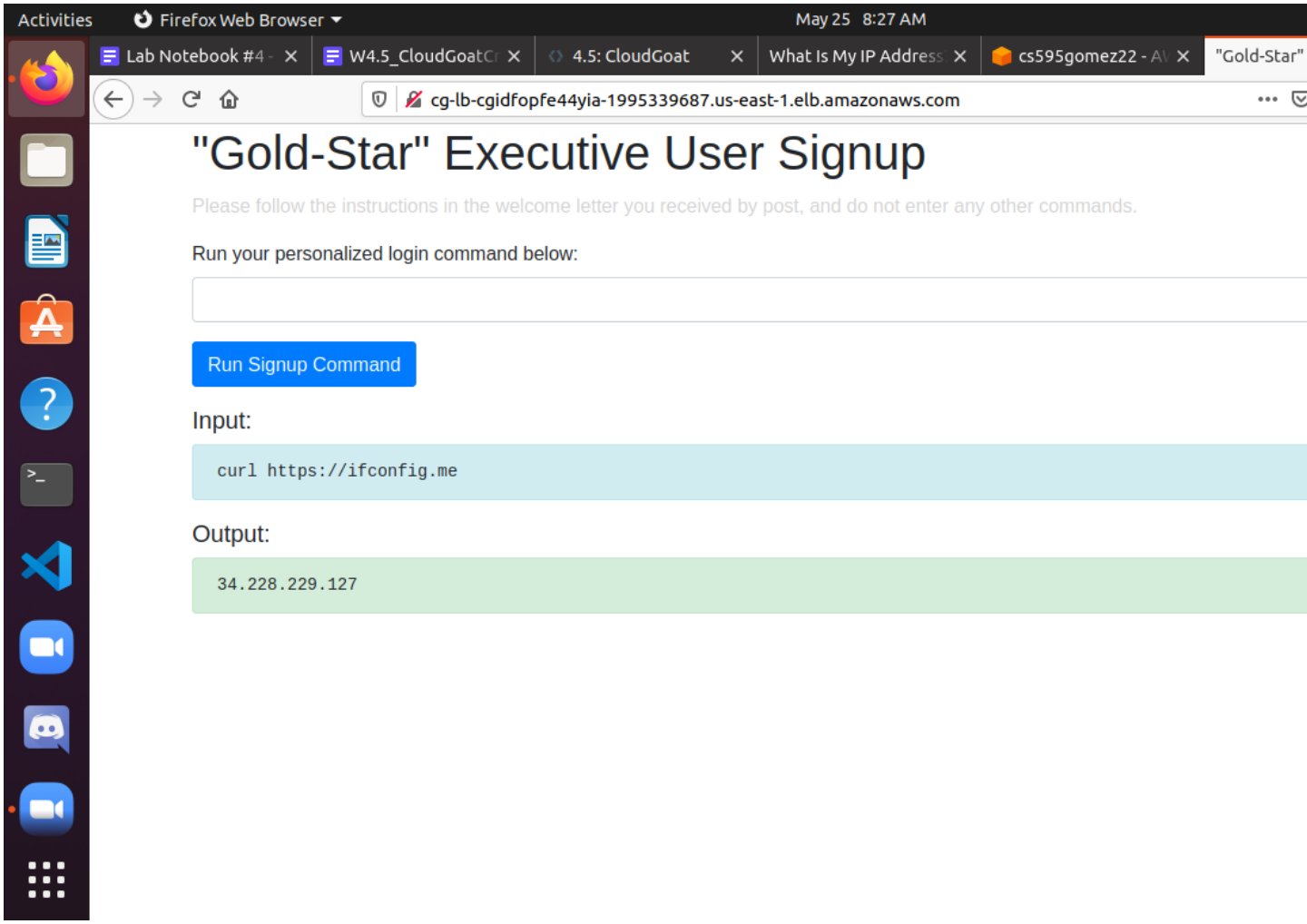


Take a screenshot of the output in out.txt

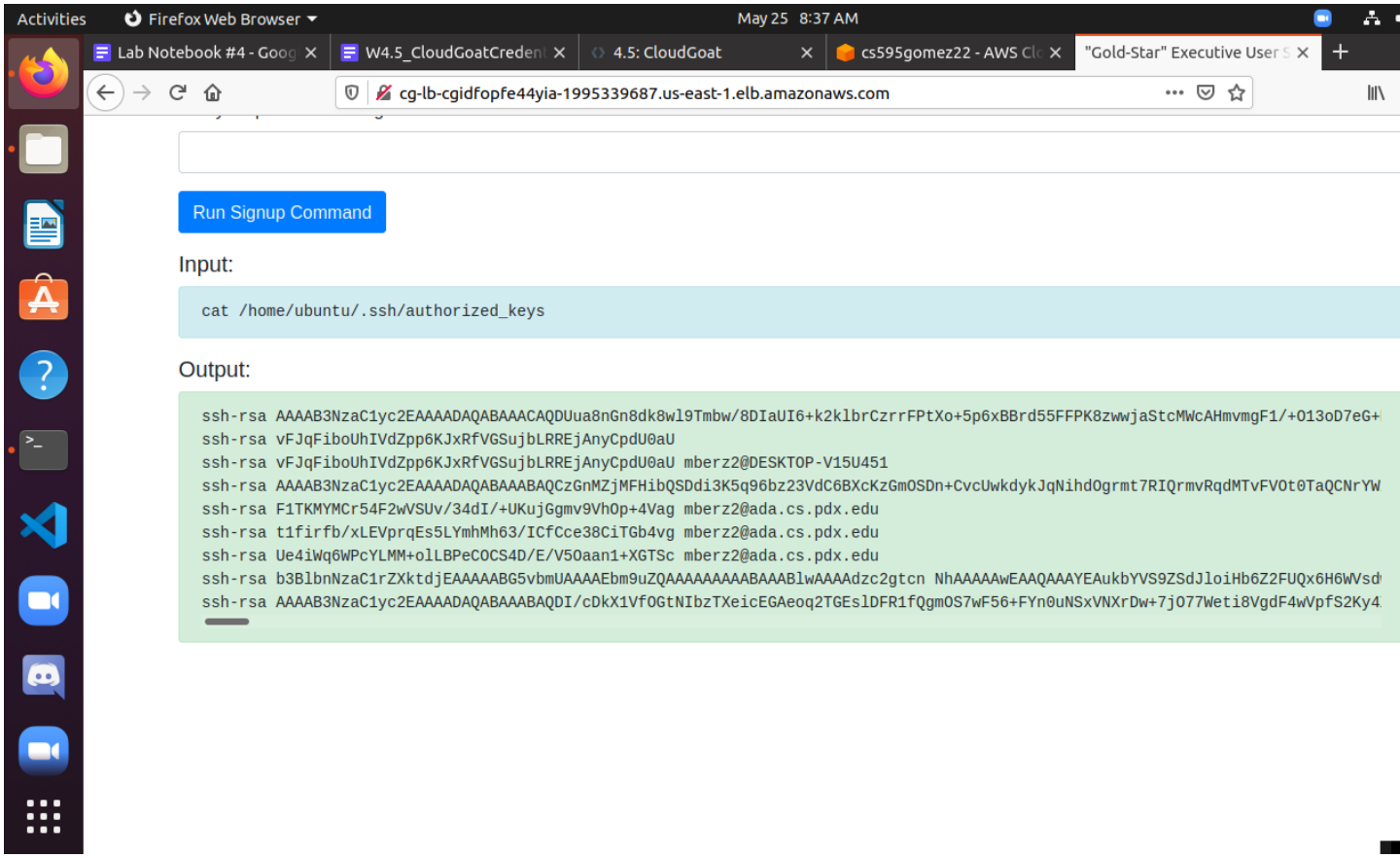


Rce_web_app

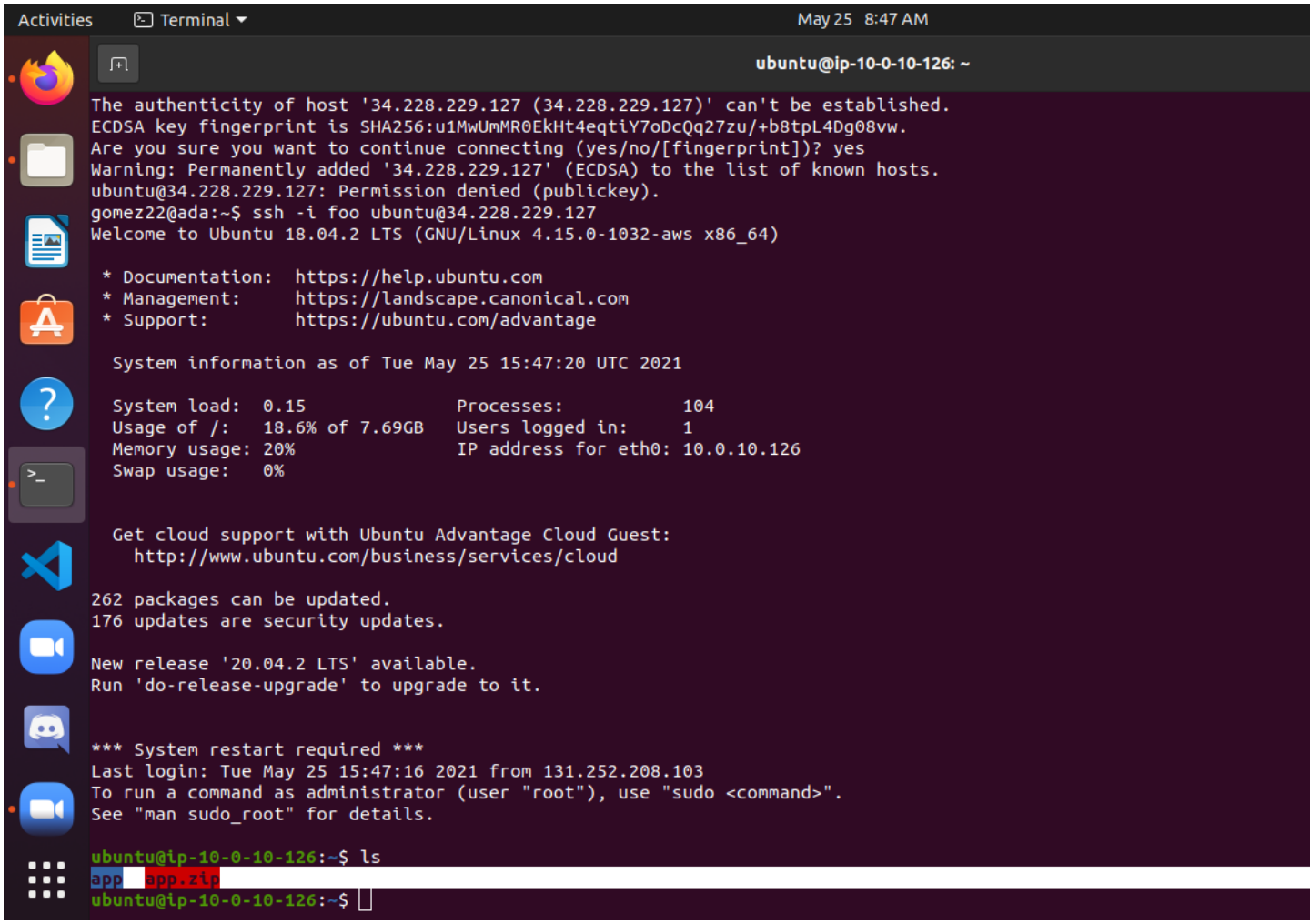
Show the IP address revealed by the command.



Show the last several lines of the file.

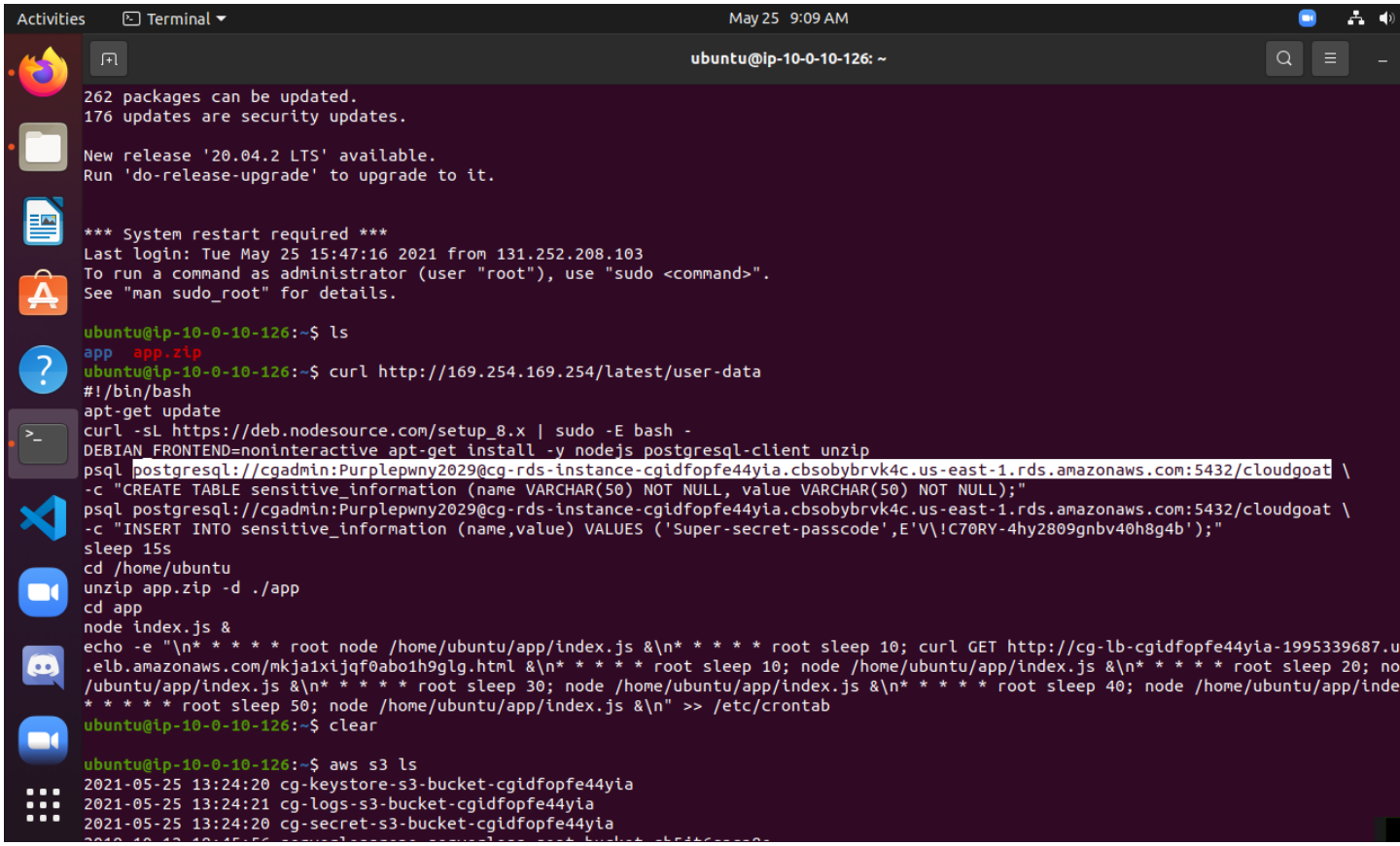


Show a directory listing of the account you've logged into.

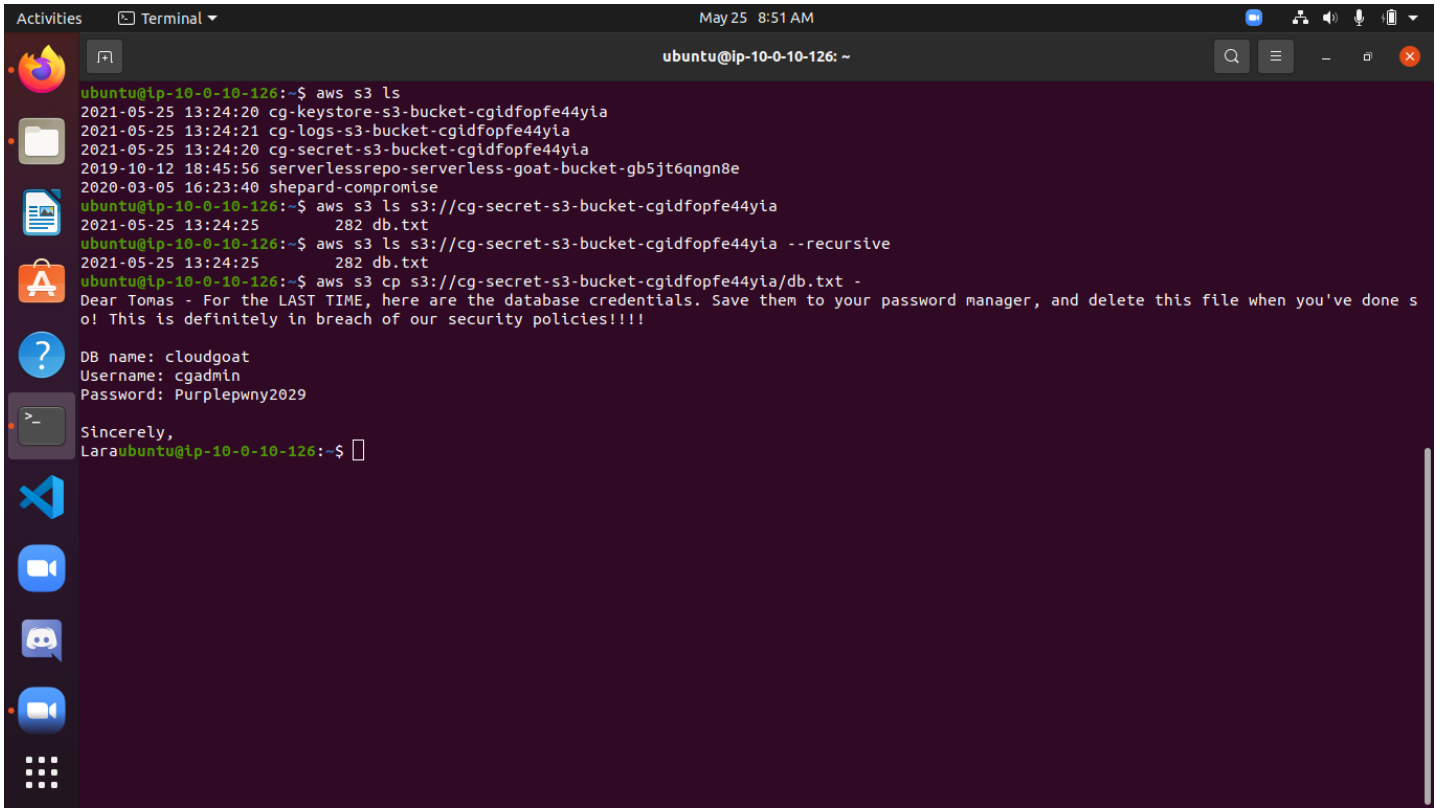


Show the output of this script and highlight how it reveals the database credentials.

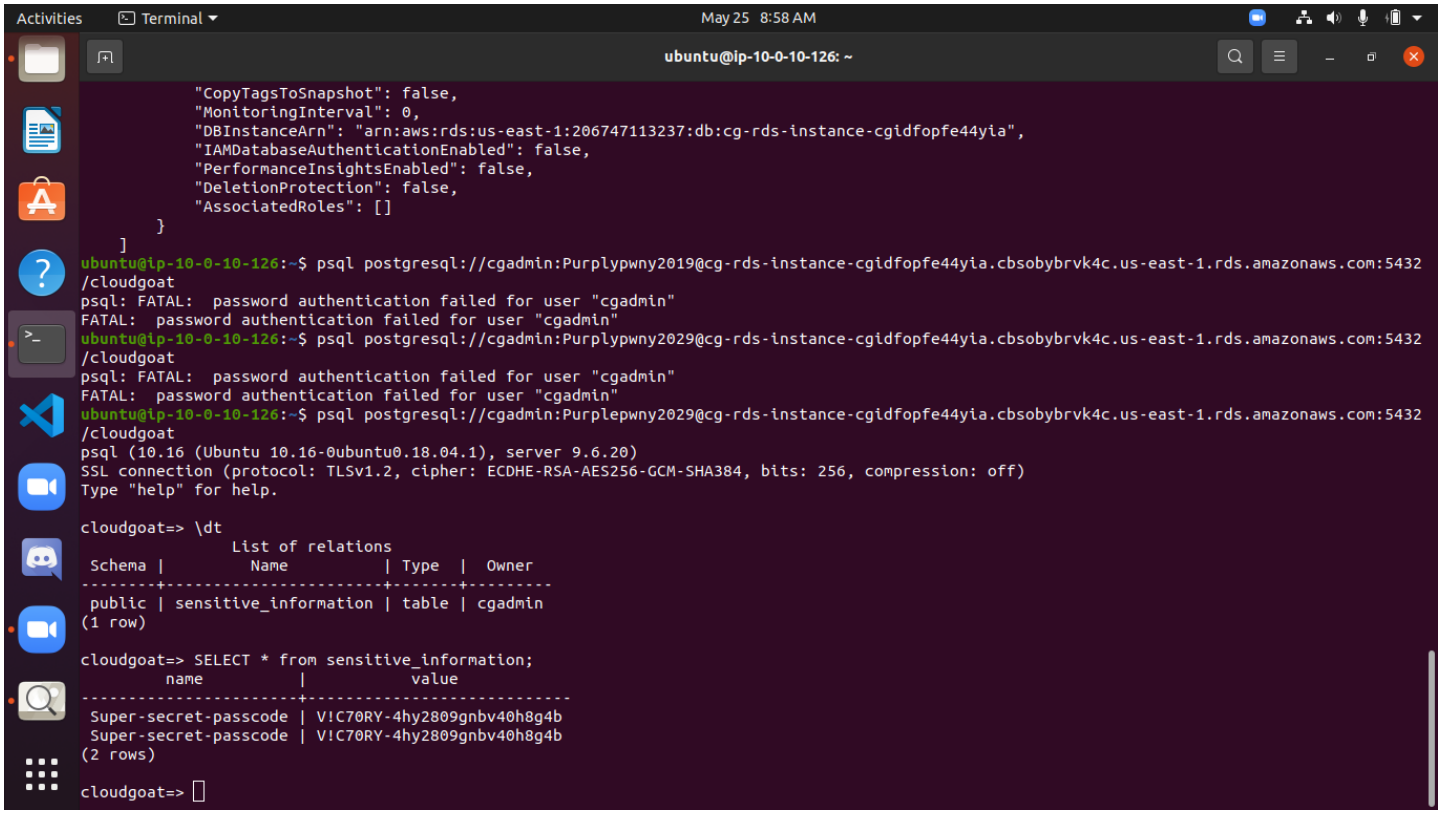
-The highlighted line in the screenshot below shows the username, password, instance name, and port number of a rds instance. This reveals everything you need to get access to whatever is in the database.



Show the contents of the file.



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:Purplypwny2019@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:Purplypwny2029@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:Purplepwny2029@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 | V!C70RY-4hy2809gnbv40h8g4b
Super-secret-passcode | V!C70RY-4hy2809gnbv40h8g4b
(2 rows)
cloudgoat=>