



Secure Secrets with Secrets Manager

R

Russell Geisler

[Code](#) [Blame](#) 39 lines (2B loc) · 1.17 KB

[Raw](#) [Copy](#) [Download](#) [Edit](#) [View](#)

```
1 import boto3
2 import json
3 from botocore.exceptions import ClientError
4
5 def get_secret():
6
7     secret_name = "aws-access-key"
8     region_name = "us-east-2" # Replace with your AWS region if different
9
10    # Create a Secrets Manager client
11    session = boto3.Session()
12    client = session.client(
13        service_name='secretsmanager',
14        region_name=region_name
15    )
16
```

Introducing Today's Project!

In this project I will demonstrate how to store code securely using AWS Secrets Manager

Tools and concepts

I learned about GitHub and AWS Secrets Manager

Project reflection

This project took about 3 hours

I did this project to learn more about AWS Secrets Manager

Hardcoding credentials

Credentials in code can leave you open to attackers easily being able to access and change or delete your resources

In the config file I put my own credentials

```
# config.py - TEMPORARY for demonstration only
# WARNING: This is NOT safe for production! We'll
fix it with Secrets Manager.

AWS_ACCESS_KEY_ID = "AKIAW3MEFRAFTQM5FHKE"
AWS_SECRET_ACCESS_KEY =
"F0b8s5m+p0ZsttvBCirr1B0utuvCpqXMW2Y1qAxY""YOUR_ACT
UAL_SECRET_ACCESS_KEY"
AWS_REGION = "us-east-2"
```

Pushing Insecure Code to GitHub

I fork the original repository so i can add my own edits to it. Cloning would have made an offline copy

I used git remote add origin <my URL> to add repository, git commit adds the changes, git push -u origin main pushes the changes to the repository

Github blocked the request because it detected some sensitive credentials

```
remote: error: GH013: Repository rule violations found for refs/heads/main
remote:
remote: - GITHUB PUSH PROTECTION
remote:   _____
remote:   Resolve the following violations before pushing again
remote:
remote:   - Push cannot contain secrets
remote:
remote:
remote:   (?) Learn how to resolve a blocked push
remote:       https://docs.github.com/code-security/secret-scanning/working-
with-secret-scanning-and-push-protection/working-with-push-protection-from-
the-command-line#resolving-a-blocked-push
remote:
```

Secrets Manager

Secrets Manager is used for keeping credentials

Secret rotation is the process of automatically changing your secrets on a regular schedule. This is a security best practice because it reduces the risk of compromised credentials.

Secret Manager shows sample code which is very helpful! It shows you exactly how to retrieve your secret from Secrets Manager in your application code.

Sample code
Use these code samples to retrieve the secret in your application.

[Java](#) | [JavaScript](#) | [C#](#) | [Python3](#) | [Ruby](#) | [Go](#) | [Rust](#)

```
1 // Use this code snippet in your app.
2 // If you need more information about configurations or implementing the sample
3 // code, visit the AWS docs:
4 // https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/home.html
5
6 // Make sure to import the following packages in your code
7 // import software.amazon.awssdk.regions.Region;
8 // import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
9 // import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
10 // import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
11
12 public static void getSecret() {
13
14     String secretName = "aws-access-key";
15     Region region = Region.of("us-east-2");
```

Java Line 1, Column 1 | Errors: 0 | Warnings: 0

[Download AWS SDK for Java](#)

Updating the web app code

I used the sample code and made a new config file using notepad

These lines are responsible for actually retrieving the credentials from the secret and assigning them to the AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, and AWS_REGION variables that our app.py code expects

```
def get_secret():

    secret_name = "aws-access-key"
    region_name = "us-east-2"

    # Create a Secrets Manager client
    session = boto3.session.Session()
    client = session.client(
        service_name='secretsmanager',
        region_name=region_name
    )

    try:
        get_secret_value_response =
            client.get_secret_value(
                SecretId=secret_name
            )
    except ClientError as e:
        # For a list of exceptions thrown, see
        #
        https://docs.aws.amazon.com/secretsmanager/latest
        /apireference/API_GetSecretValue.html
        raise e

    secret =
    get_secret_value_response['SecretString']

    # Your code goes here.
    |
```

Rebasing the repository

Git rebasing is used to rewrite commit history

Git detects conflicting changes in a file. I resolved the issue by deleting conflicts

I was able to pull up the config.py file in my repository and verify the credentials were not able to be seen

```
pick fbcc895 Updated config.py
# Rebase fbcc895 onto 91f1de9 (6 commands)
#
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

