Repository Details:
https://github.com/SJSU-Mahmood/lab-iam-juniemariam

Hash Commit: fdd8780

Task 1: Assign a Role to an EC2 Instance to Access an S3 Bucket (2 pts)

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

bucket_name = "my-bucket-640168430849"
ec2_private_ip = "172.31.11.16"
```

```
ubuntu@ip-172-31-11-16:~$ aws s3 ls s3://my-bucket-640168430849
2024-10-02 02:48:19         44 Junie-Mariam-Varghese.txt
ubuntu@ip-172-31-11-16:~$ aws s3 cp s3://my-bucket-640168430849/Junie-Mariam-Varghese.txt .
download: s3://my-bucket-640168430849/Junie-Mariam-Varghese.txt to ./Junie-Mariam-Varghese.txt
ubuntu@ip-172-31-11-16:~$ cat Junie-Mariam-Varghese.txt
My name is Junie, my partner's name is Rinkuubuntu@ip-172-31-11-16:~$
```

Task 2: Allow Your partner's IAM Role to Access Your S3 Bucket (3 pts)

- Listing the contents of your partner(s) buckets from your EC2 instance

```
ubuntu@ip-172-31-11-16:~$ aws s3 ls s3://my-bucket-571600871347
2024-10-02 03:49:23         44 Rinku_Tekchandani.txt
```

- Copying the files from your partner(s) buckets to your EC2 instance

```
ubuntu@ip-172-31-11-16:~$ aws s3 cp s3://my-bucket-571600871347/Rinku_Tekchandani.txt .
download: s3://my-bucket-571600871347/Rinku_Tekchandani.txt to ./Rinku_Tekchandani.txt
ubuntu@ip-172-31-11-16:~$ ls -l
total 12
-rw-rw-r-- 1 ubuntu ubuntu   44 Oct  2 02:48 Junie-Mariam-Varghese.txt
-rw-rw-r-- 1 ubuntu ubuntu   44 Oct  2 03:49 Rinku_Tekchandani.txt
drwx------ 3 ubuntu ubuntu 4096 Oct  2 03:33 snap
```

- Reading the files from your instance using cat Partner's_Filename.txt

```
ubuntu@ip-172-31-11-16:~$ cat Rinku_Tekchandani.txt
My name is Rinku, my partner's name is Junieubuntu@ip-172-31-11-16:~$
```

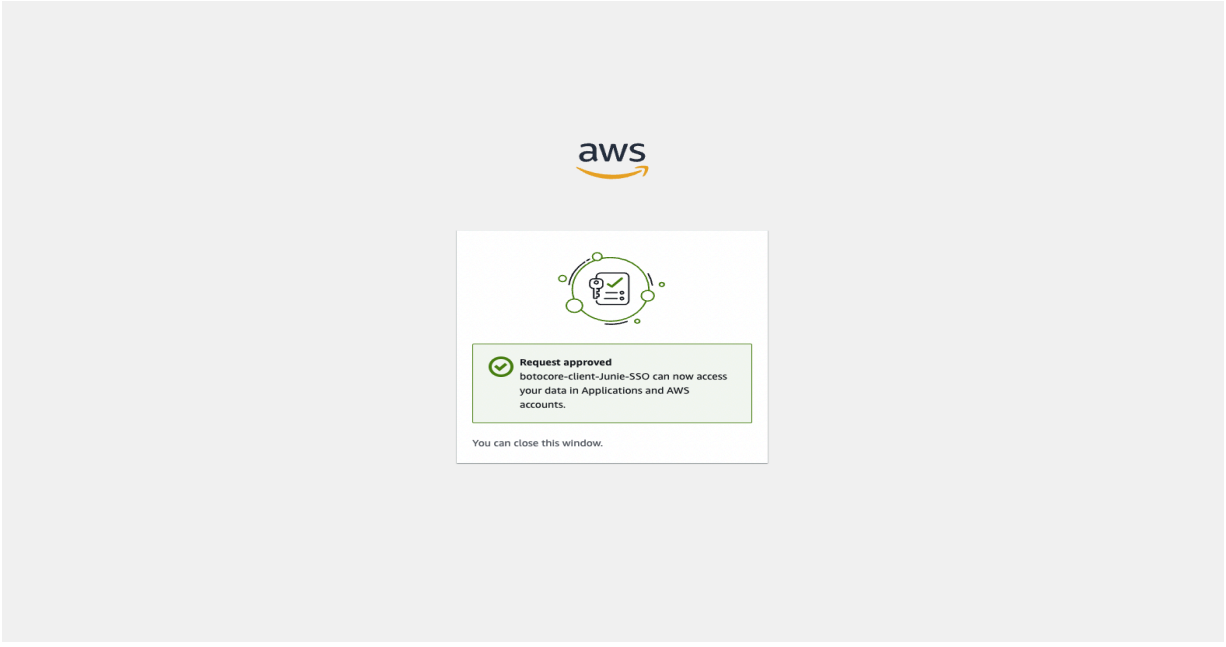
Task 3: Revisiting SSO (2 pts)

```
rohan@junie lab-iam-juniemariam-main % aws configure sso
SSO session name (Recommended): Junie-SSO
SSO start URL [None]: https://d-9167189022.awsapps.com/start
SSO region [None]: us-west-1
SSO registration scopes [sso:account:access]:
Attempting to automatically open the SSO authorization page in your default browser.
If the browser does not open or you wish to use a different device to authorize this request, open the following URL:

https://device.sso.us-west-1.amazonaws.com/

Then enter the code:

TMPR-HJGH
```



```
rohan@junie lab-iam-juniemariam-main % aws sts get-caller-identity
{
    "UserId": "AIDAZKDIDBUA7ATL244R5",
    "Account": "640168430849",
    "Arn": "arn:aws:iam::640168430849:user/JunieMV"
}
```

Knowledge Check (3 pts)

1. Did you run into any errors while working with ChatGPT? What were they and how did they impact your performance in this lab?

While working with ChatGPT, I encountered a few issues, primarily because some of the resources it suggested were deprecated. This happened because its training data included outdated information. Additionally, there were instances where I failed to provide specific details, which led to responses that did not fully align with my requirements.

For my first task, ChatGPT was extremely helpful. It provided a quick reference to commands I needed, which speed up my progress. However, during the second task, things got a bit confusing. When suggesting the IAM role ARN for my partner's access to the S3 bucket, ChatGPT did not specify that we needed to include my partner's bucket in the IAM policy too. This led to some confusion and took extra time to resolve.

The effectiveness of using ChatGPT comes down to how quickly you can validate the information it provides. The key to effectively using ChatGPT is validating the information promptly, which can significantly enhance both its utility and your overall performance when working with LLMs.

2. What was your biggest takeaway from the two articles you read?

The biggest takeaway from this article is the emphasis on using secure and appropriate authentication methods for different AWS use cases, rather than relying on static AWS access keys and secret keys, which pose a significant security risk. The author highlights how AWS's default authentication mechanism can lead to poor security practices and

advocates for using temporary credentials, IAM roles, AWS SSO, and other alternatives tailored to specific scenarios, such as EC2 instances, CI/CD pipelines, and external compute environments. The core message is that adopting these secure authentication strategies significantly reduces the risk of credential leakage and enhances security across AWS environments.

Amazon Cognito is a practical solution for managing user identities in web and mobile apps. It consists of two key components: user pools and identity pools. User pools are designed to handle user sign-ups and logins, allowing users to either create accounts directly or sign in using their existing accounts from providers like Google or Facebook. This feature also generates tokens that facilitate access to your APIs.On the other hand, identity pools are focused on providing users with temporary AWS credentials. This means that whether users are logged in or just browsing as guests, they can still access AWS resources. Together, user pools and identity pools work in harmony, making it easier for developers to manage user identities and control what users can do within their applications.

3. What is the difference between AWS Cognito User Pools and Identity Pools?

<u>AWS Cognito User Pools</u>

- Function: Focused on user authentication and management.
- Key Features: User sign-up/sign-in, profile management, security features (MFA, password recovery), and OAuth 2.0 support.
- Use Case: Ideal for creating and managing user directories and handling user login.

<u>AWS Cognito Identity Pools</u>

- Function: Provides temporary AWS credentials for accessing AWS resources.
- Key Features: Supports federated identities from various providers, grants access to AWS services and allows for unauthenticated access.
- Use Case: Suitable for enabling access to AWS resources for both authenticated and unauthenticated users.