Course: IT485

Project: HawkinCloud

Members: Aliyev Omar, Haider Zaheer, Michael Hernandez, Nephthro Pierre.

REPORT #4

Management Guidance:

The purpose of our previous reports, II and III, was to develop revised methods for publishing our project. We have categorized our presentations and reports into milestone structures, which are used to measure the percentage of completion of the project, as well as predict solutions to potential issues based on past tasks. We are holding weekly meetings, both in person and via Zoom, in order to learn new methods for the project environment and implement them as we move forward. In addition, we are working on the project's cover and adding new learning methods to obtain AWS certifications. We have created shares for project parts, including high-level assessments and coding infrastructure base models. This helps us to complete any part of the project, including AWS documentation, by following the steps required to migrate to the project.

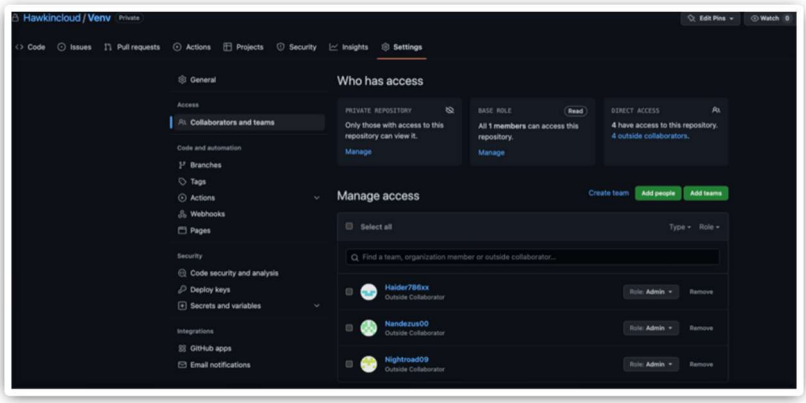
Team Member participant:

Currently, we are working on informing each group member about the base types of learning skills in AWS and Boto3 programmatic use protocols. Depending on each team member's knowledge, we can share the "pizza slice theory," which Jeff Bezos mentioned, to divide knowledge into eight pieces and share them among group members to successfully complete the project. As the AWS environment requires intermediate learning types, we are currently working on developing those skills. As Cloud Practitioners, we are looking for events, data protocols, and access methods to implement CloudTrail processes. This will require weekly resources to understand the big data center background of AWS and help meet customer demands as we work with shared responsibility models for projects. It is important to access and maintain the project structure as developers or system administrators. We assume that our project members will collaborate strongly and finish the project before the deadline. At the end of the project roadmap, we will hire a senior developer and senior system operations member to check each protocol and develop an easy way to include all statements in the final version.

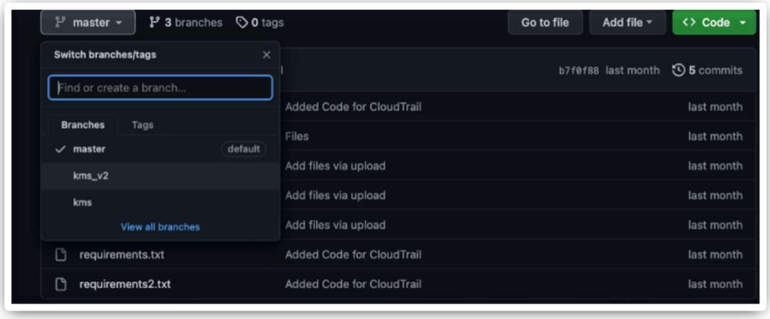
o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu

Output Deliverables:

Github Merge/Push:



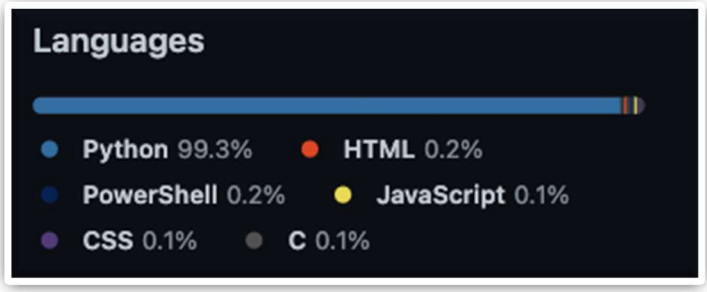
Organizing the project types in GitHub is useful for efficiently deploying each programming language into packages. This allows for project protection, shareability, and accessibility across a wide infrastructure. We are organizing our private repository to share real-time programming updates, and the creator of the repository can grant access to collaborators and teams based on their permission levels. We also use failure attempts blocking actions to enhance the security of our repository. This method is commonly used in project features, and we supervise our repository methods via the GitHub dashboard or Gitlab command line interface.



Once any group member is working on a project concept, they can merge it into different branches and push it to the repository. This method allows for easy access by the master branch developer, such as "Aliyev Omar," who can efficiently see the final output. Additionally, to provide information about the project's characteristics, we can easily demonstrate which programming language is being used by utilizing the Languages dashboard and categorizing it in the workflow features.

In the further project's final version, we will deploy the **".gitignore"** file to suppress all dependencies and packages. This way, downloading them requires only one command, and the project can run on any other local environment using a particular method.

```
aliyevom@Aliyev-2 Hawkincloud % git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   ../.DS_Store
        modified:   .DS_Store
        deleted:    .gitignore
        deleted:    instance/database.db
        modified:   requirements.txt
        modified:   templates/index.html
        modified:   ../requirements.txt
        new file:   ../requirements2.txt
        new file:   ../requirements3.txt
```

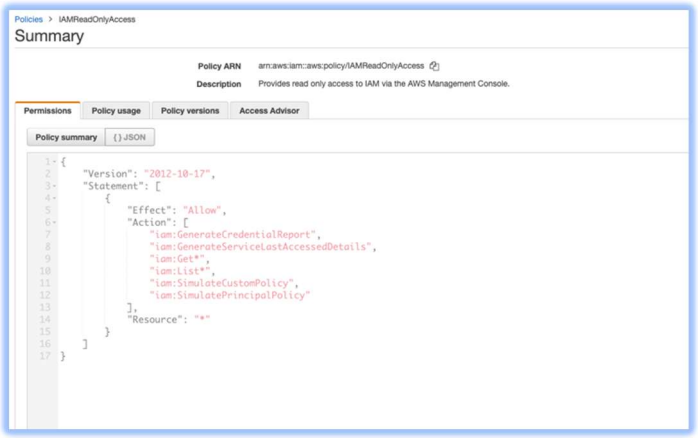


o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu

AWS IAM Events/Policy/Roles:

The simulation of policies and events roles by policy makers is a crucial aspect of supporting the AWS console's centralized management of all accounts. This activity centralizes and utilizes IAM to control each user group with key pair values assigned to the users. We have already created a group named "Admin" and assigned each user roles depending on their specific project requirements. This enables various permissions and access advisors, granting policies permission to users who accessed it the last time and are currently developing it, thus enabling us to monitor each user activity daily.

In the AWS infrastructure, we are developing a related console in the middle performance settings to attach/detach permissions. The policy maker or use of existing policies is the main stereotype in AWS. We are currently obtaining and adding up to five permissions to the policies to be attached depending on users/groups. This is done in the boto3 programmatic SDK, which relates to policies after being attached to users/groups. As trusted entities, AWS services configure each event created/built/supported to access programming routes that can be handled by output API activity in the backend.

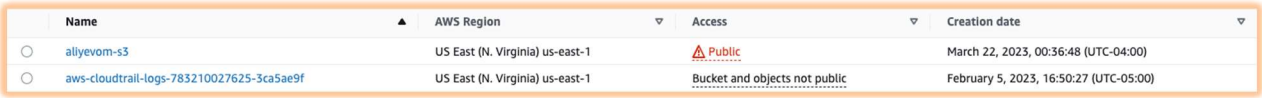


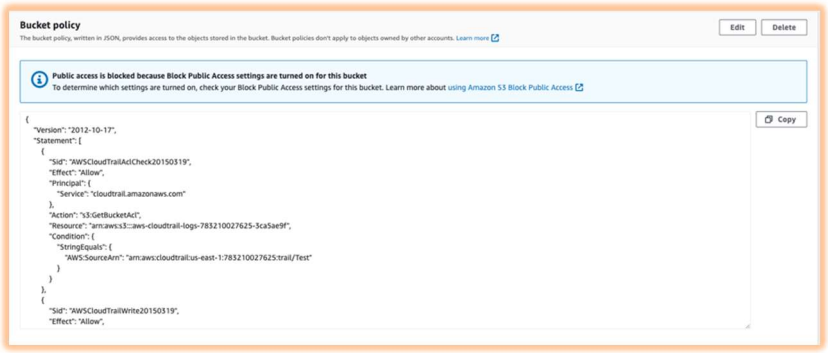
Some specific policies that we are using in the project grant access to Key Management Services, which can rotate TRUE or FALSE outputs. Additionally, there is the AWS CloudTrail read-only access policy, which covers our project permission types, allowing any user types to get, describe, and list each event's pros and cons and resource types. Furthermore, in case of a situation, the **IAMReadOnlyAccess** generates access roles that limit any overload access in AWS to the programmatic way. This is because retrieving more data in a programmatic way can lead to billing dashboard over-pay due in the AWS environment. This policy was created to optimize system control and prevent the migration of too much data into the system and back to boto3.

o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu

S3 bucket/CloudTrail:

By default, CloudTrail logs created in S3 buckets are called objects and are kept private. However, users or groups can be granted permission to access them by modifying the S3 bucket's access policies. In order to receive CloudTrail logs, we have modified an S3 bucket and changed its policy trail using the AWS command line interface. The following policies were included: bucket name, service principal name of the CloudTrail, and the name of the folders where the log files are stored, based on the day/time/attempting services prefix and our group members' AWS account ID.





With the SDK programmatic routes, we can access the output either directly from the CloudTrail dashboard or through the S3 bucket. This enables us to easily retrieve logs and analyze them to identify any issues or security concerns. We are also exploring ways to automate this process by using Lambda functions to trigger specific actions based on the logs received. This will help us streamline our security and compliance monitoring efforts and ensure that we are always up-to-date with the latest events happening in our AWS environment.

Furthermore, we are implementing best practices for S3 bucket policies, including using the principle of least privilege to ensure that users or groups are only granted the necessary permissions to perform their tasks. We are also enforcing strong password policies and multi-factor authentication for all users to prevent unauthorized access to our AWS resources. By taking these proactive measures, we can better protect our sensitive data and ensure that our AWS environment remains secure and compliant with industry standards.

o.aliyev001@umb.edu
m.zaheer001@umb.edu
michael.hernandez003@umb.edu
nephthro.pierre001@umb.edu