Aliyev Omar
Haider Zaheer
Michael Hernandez
Nephthro Pierre

http://hawkincloud.github.io/dev/

OVERVIEW PROJECT

1. Generalize Project
2. AWS
3. GitHub Sessions
4. Docker Configure AWS
5. Configure Platform as the service

Generalize Project

- Flask Boto3 project is a web application built using the Flask framework and the Boto3 library for AWS.
- Project includes several routes that make requests to AWS services to retrieve data about security groups, keys, and CloudTrail events.
- Retrieved data is parsed and processed before being returned to the user in JSON format.
- The project uses SQLAlchemy to store user information in an SQLite database, and Flask-Login to manage user authentication and authorization.
- Includes forms for user registration and login, which are validated using the Flask-WTF library.
- Bootstrap for front-end styling and includes static assets such as images.
- Project demonstrates how to make requests to AWS services using Boto3 and how to integrate AWS data into a Flask application.

AWS

- Flask web application connects to AWS using the Boto3 library.
- Data from various AWS services using the Boto3 library.
- Application allows the user to view data from the following AWS services: CloudTrail, KMS, and EC2.
- CloudTrail service to retrieve data about security groups and events.
- KMS service to retrieve data about encryption keys.
- EC2 service to retrieve data about EC2 instances.
- Web application uses Bootstrap to create a responsive user interface.

GitHub Session

- Name: CI/CD Pipeline to AWS ElasticBeanstalk
- Environment variables:
  - EB_PACKAGE_S3_BUCKET_NAME: the name of the S3 bucket where the deployment package will be stored
  - EB_APPLICATION_NAME: the name of the ElasticBeanstalk application
  - EB_ENVIRONMENT_NAME: the name of the ElasticBeanstalk environment
  - DEPLOY_PACKAGE_NAME: the name of the deployment package that will be created
  - AWS_REGION_NAME: the name of the AWS region to be used

my_ci_pipeline:

Runs on: Ubuntu latest

- Steps:

Git clone our repository

Create ZIP deployment package

Configure AWS credentials

Copy our Deployment package to S3 bucket

Print nice message on completion of CI Pipeline

my_cd_pipeline:

Runs on: Ubuntu latest

Needs: my_ci_pipeline

- Steps:

Configure AWS credentials

Create new ElasticBeanstalk Application Version

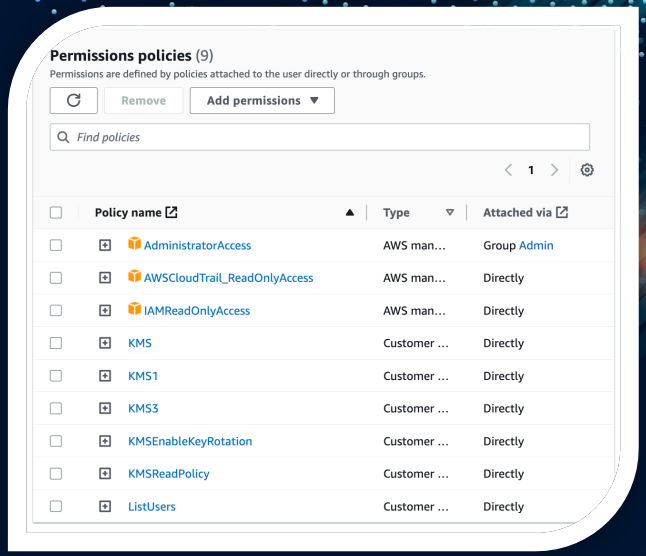
Deploy our new Application Version

Print nice message on completion of CD Pipeline

Docker configure AWS

We are currently in the process of installing AWS IAM users with specific policies to be used for certain events in our application. As shown in the following picture, there are no existing containers or images when checking 'docker ps' and 'docker images' commands. After successfully building the image and confirming its availability in your local machine, we need to attach some policies in AWS IAM as permissions.

```
aliyevom@Aliyev-2 venv % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
aliyevom@Aliyev-2 venv % docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
aliyevom@Aliyev-2 venv %
```




```
aliyevom@Aliyev-2 venv % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
aliyevom@Aliyev-2 venv % docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
test-app     latest    6701c22753f7   29 seconds ago   576MB
aliyevom@Aliyev-2 venv %
```

Configure Platform as the service

Installation Guide For Presentation 6

Docker Image Available on Dockerhub

Please install Docker on your PC and test whether the "hello-world" Docker image runs in your terminal. To do so, run the following command:

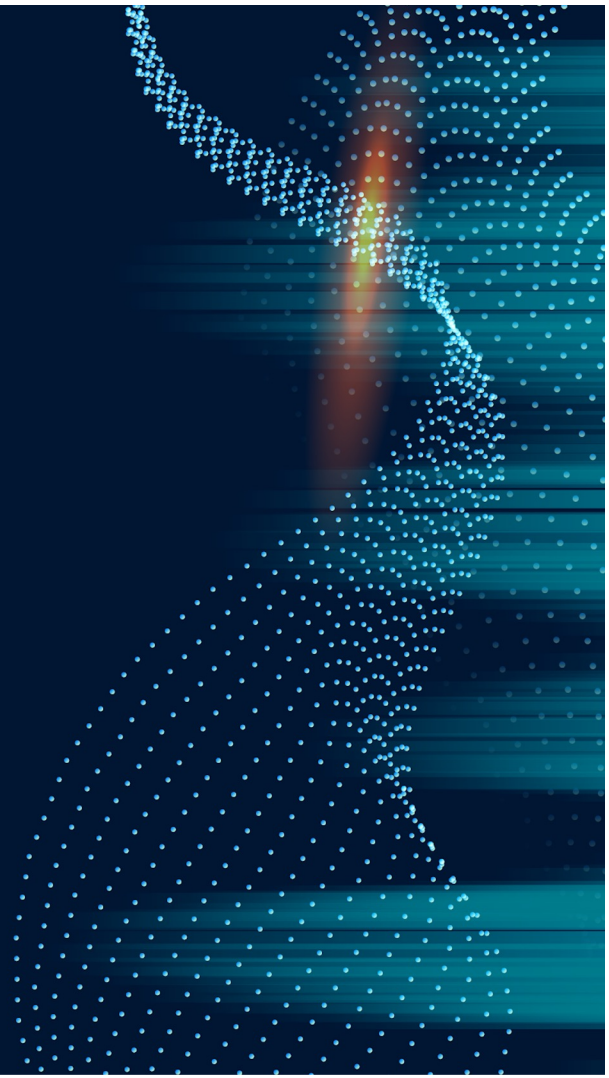
Once you have confirmed that the output is running correctly, we are ready to proceed with Project VI. For further information, we will provide paper instructions.

HawkInCloud Status

Since the last class—presentation day, our group has completed significant steps in the backend integration; that way, the frontend and backend could function properly, with the following goal in mind:

Displaying meaningful metrics from the AWS Cloudtrail, operating as an essential analytics tool for personal or business use.

Nearing the end of the AWS project, HawkinCloud, we have made drastic changes on back-end, with great efforts in maintaining a working product as fulfilled by the backend integration. Displaying significant metrics as they relate to the AWS Cloudtrail is crucial



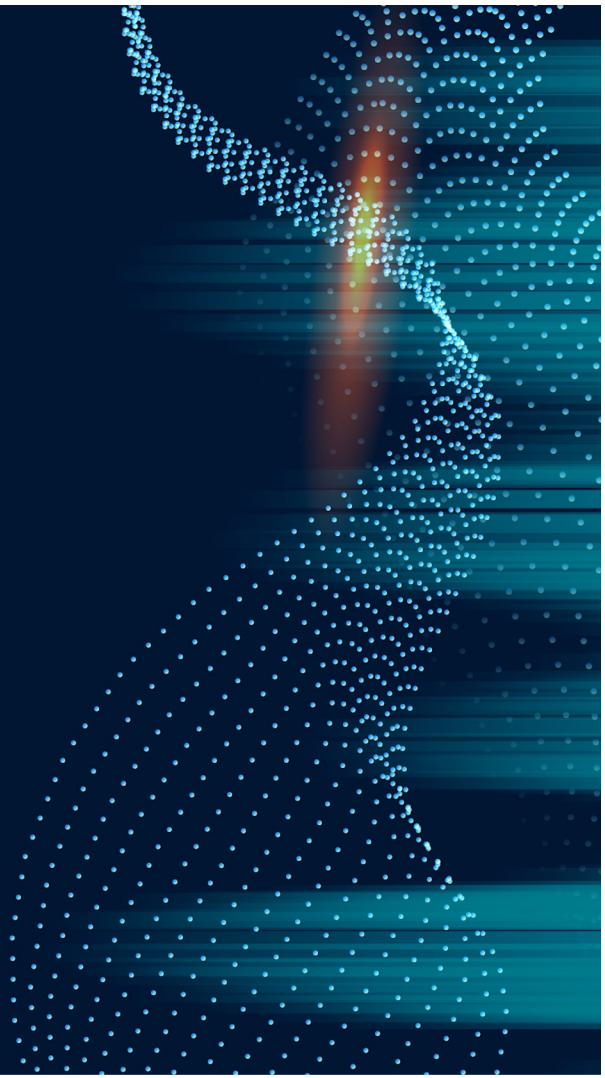
Project Roadmap/Milestones (AWS)

Basic Outline and Initialization of AWS functions

IAM Setup and SDK/CDK Configuration (CLI, Boto3, Management Console) for CloudTrail and EC2

CI/CD Pipeline, to load application using AWS Services (EC2, ECS)

UI takes/displays JSON data, and Utilizes Bootstrap for integration (CDK)

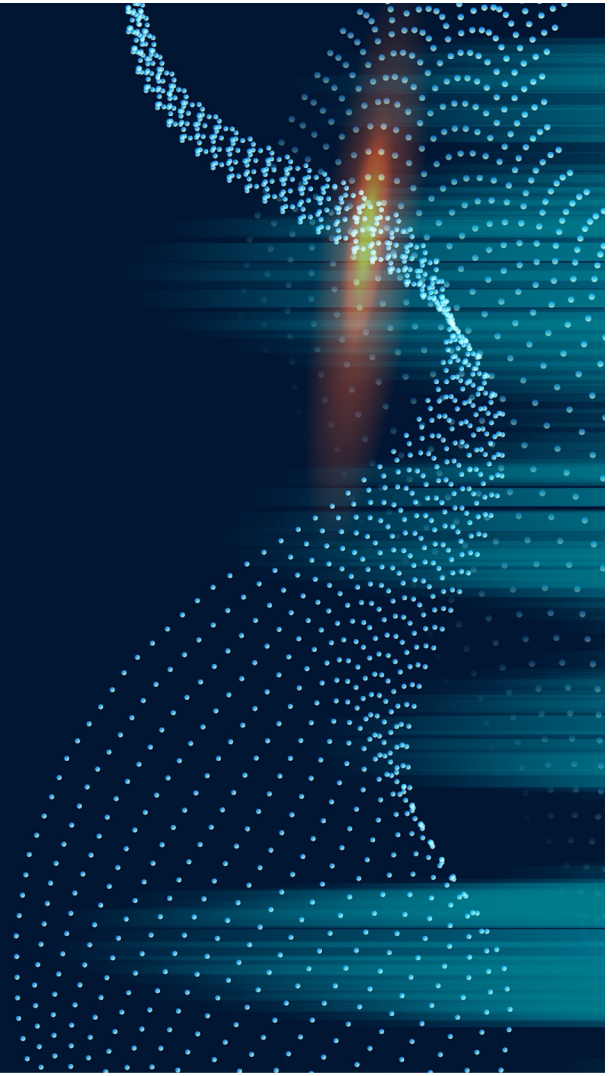


Collaboration Efforts

The HawkInCloud team has been consistently working together to make this project possible.

A few hiccups have occurred along the way, but we have all been steadfast in making our ideas a reality.

Moving Forward, Wrapping up HawkInCloud will require no less of us than we have already been putting forth.



THANK ‘YOU