Aliyev Omar
Haider Zaheer
Michael Hernandez
Nephthro Pierre

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

Microservices & DevOps CI/CD to AWS

01 **Project Concept**
Generalize Roadmap

02 **Microservices**
Dockerize Python App, Write Kubernets YAML files prepare deployment.

03 **CI/CD Pipeline**
Microservices + AWS ECS + EC2 + Application Load Balancer

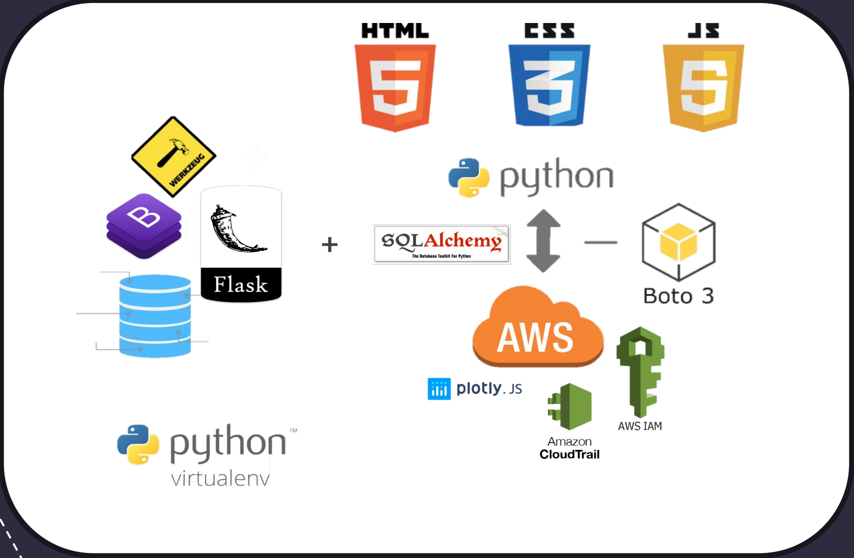
04 **GitHub Actions**
Set up, Run actions/checkout, Install dependencies.

Project Concept

Aliyev Omar

Microservices

- Python 3. with PIP
- Docker & Docker-Compose
- Kubernetes Cluster like Minikube



GitHub Actions

- Run actions and works via ECS configuration
- Set up Dependencies, creates snapshots in containers microservices.
- Post run actions extend pipeline in CI/CD

CI/CD Pipeline

Microservices

- Python 3. with PIP
- Docker & Docker-Compose
- Kubernetes Cluster like Minikube

↕

- Create/Register Elastic Container Service
- Volume up EC2 Instances to ECS
- Application Load Balancer replicating

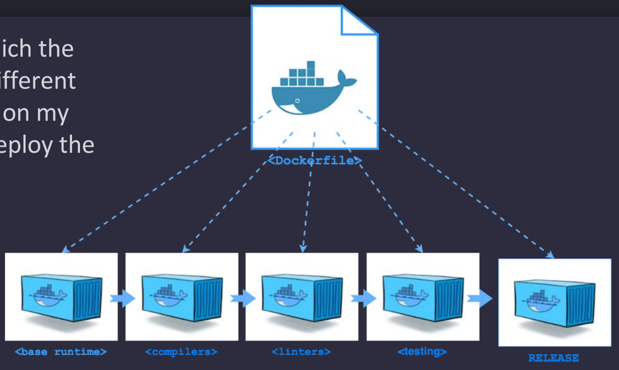
2

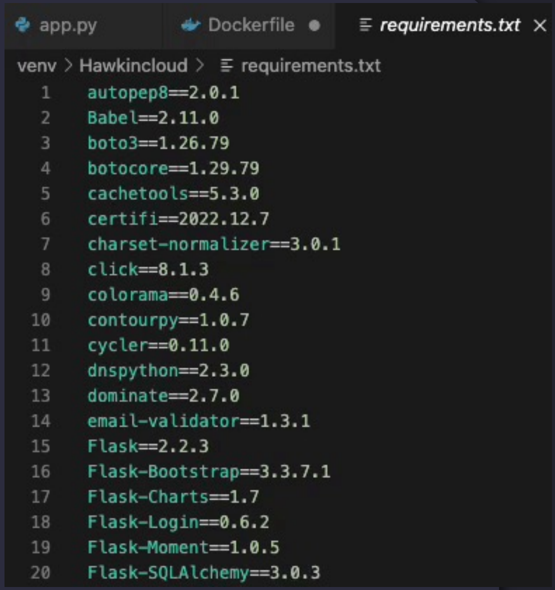
Microservices

Aliyev Omar

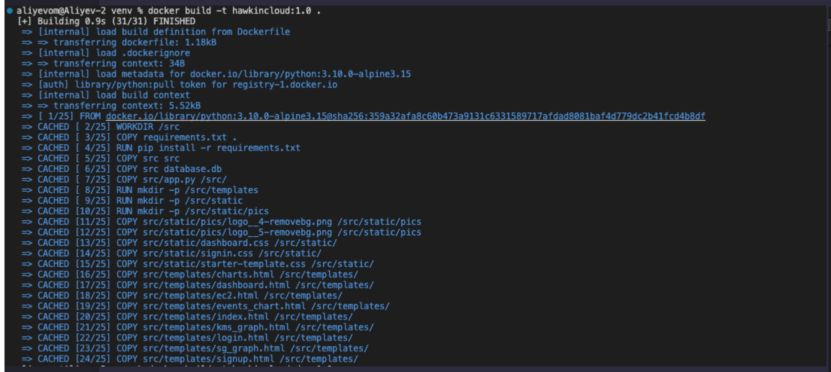
Docker ensures that the environment in which the Flask application runs is consistent across different systems. This helps to eliminate the "works on my machine" problem and makes it easier to deploy the application to different environments.

Isolation
Scalability
Reproducibility
Portability

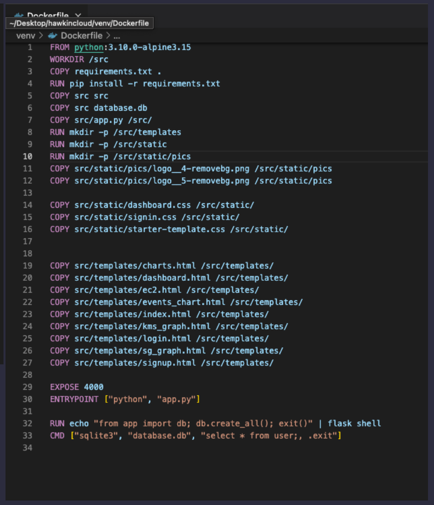




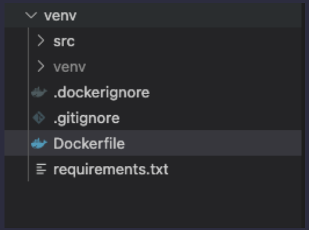
```
autopep8==2.0.1
Babel==2.11.0
boto3==1.26.79
botocore==1.29.79
cachetools==5.3.0
certifi==2022.12.7
charset-normalizer==3.0.1
click==8.1.3
colorama==0.4.6
contourpy==1.0.7
cycler==0.11.0
dnspython==2.3.0
dominate==2.7.0
email-validator==1.3.1
Flask==2.2.3
Flask-Bootstrap==3.3.7.1
Flask-Charts==1.7
Flask-Login==0.6.2
Flask-Moment==1.0.5
Flask-SQLAlchemy==3.0.3
```



```
aliyevom@Aliyev-2 venv % docker build -t hawkincloud:1.0 .
[+] Building 0.9s (31/31) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 1.18kB
 => [internal] load .dockerignore
 => => transferring context: 34B
 => [internal] load metadata for docker.io/library/python:3.10.0-alpine3.15
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load build context
 => => transferring context: 5.52kB
 => [ 1/25] FROM docker.io/library/python:3.10.0-alpine3.15@sha256:359a32afa8c60b473a9131c6331589717afdad8081baf4d779dc2b41fcd4b8df
 => CACHED [ 2/25] WORKDIR /src
 => CACHED [ 3/25] COPY requirements.txt .
 => CACHED [ 4/25] RUN pip install -r requirements.txt
 => CACHED [ 5/25] COPY src src
 => CACHED [ 6/25] COPY src database.db
 => CACHED [ 7/25] COPY src/app.py /src/
 => CACHED [ 8/25] RUN mkdir -p /src/templates
 => CACHED [ 9/25] RUN mkdir -p /src/static
 => CACHED [10/25] RUN mkdir -p /src/static/pics
 => CACHED [11/25] COPY src/static/pics/logo__4-removebg.png /src/static/pics
 => CACHED [12/25] COPY src/static/pics/logo__5-removebg.png /src/static/pics
 => CACHED [13/25] COPY src/static/dashboard.css /src/static/
 => CACHED [14/25] COPY src/static/signin.css /src/static/
 => CACHED [15/25] COPY src/static/starter-template.css /src/static/
 => CACHED [16/25] COPY src/templates/charts.html /src/templates/
 => CACHED [17/25] COPY src/templates/dashboard.html /src/templates/
 => CACHED [18/25] COPY src/templates/ec2.html /src/templates/
 => CACHED [19/25] COPY src/templates/events_chart.html /src/templates/
 => CACHED [20/25] COPY src/templates/index.html /src/templates/
 => CACHED [21/25] COPY src/templates/kms_graph.html /src/templates/
 => CACHED [22/25] COPY src/templates/login.html /src/templates/
 => CACHED [23/25] COPY src/templates/sg_graph.html /src/templates/
 => CACHED [24/25] COPY src/templates/signup.html /src/templates/
```



```
FROM python:3.10.0-alpine3.15
WORKDIR /src
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY src src
COPY src database.db
COPY src/app.py /src/
RUN mkdir -p /src/templates
RUN mkdir -p /src/static
RUN mkdir -p /src/static/pics
COPY src/static/pics/logo__4-removebg.png /src/static/pics
COPY src/static/pics/logo__5-removebg.png /src/static/pics

COPY src/static/dashboard.css /src/static/
COPY src/static/signin.css /src/static/
COPY src/static/starter-template.css /src/static/


COPY src/templates/charts.html /src/templates/
COPY src/templates/dashboard.html /src/templates/
COPY src/templates/ec2.html /src/templates/
COPY src/templates/events_chart.html /src/templates/
COPY src/templates/index.html /src/templates/
COPY src/templates/kms_graph.html /src/templates/
COPY src/templates/login.html /src/templates/
COPY src/templates/sg_graph.html /src/templates/
COPY src/templates/signup.html /src/templates/

EXPOSE 4000
ENTRYPOINT ["python", "app.py"]

RUN echo "from app import db; db.create_all(); exit()" | flask shell
CMD ["sqlite3", "database.db", "select * from user;, .exit"]

```

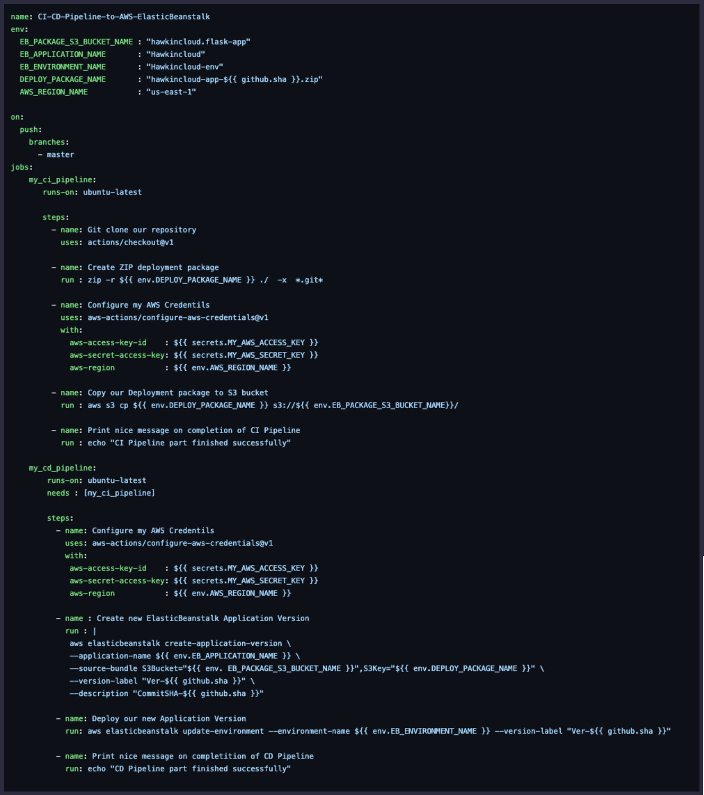


⚠️
To successfully implement microservices, configuration must be applied, and testing/checkout should be performed as part of the Continuous Integration and Continuous Delivery pipeline.

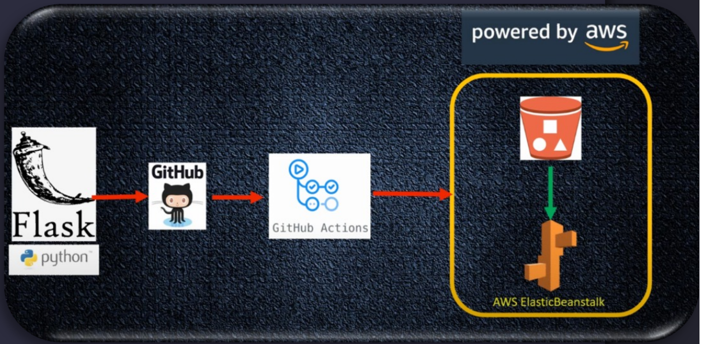
CI/CD Pipeline

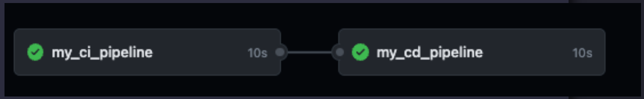
Aliyev Omar

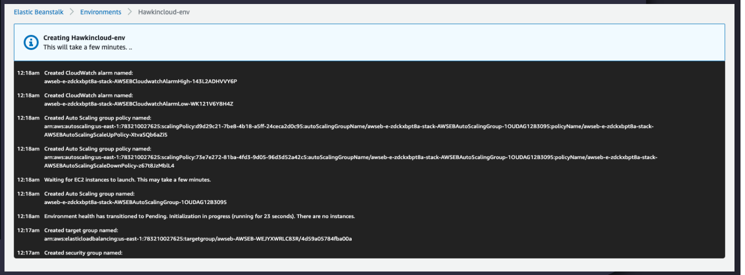
This is a YAML configuration file for a CI/CD pipeline that deploys a Flask application to an AWS Elastic Beanstalk environment. It has two jobs: my_ci_pipeline and my_cd_pipeline. The my_ci_pipeline job builds the deployment package and uploads it to an S3 bucket, while the my_cd_pipeline job deploys the new version of the application to the Elastic Beanstalk environment. Both jobs require AWS credentials stored in GitHub secrets.



```yaml
name: CI-CD-Pipeline-to-AWS-ElasticBeanstalk
env:
  EB_PACKAGE_S3_BUCKET_NAME : "hawkincloud.flask-app"
  EB_APPLICATION_NAME       : "Hawkincloud"
  EB_ENVIRONMENT_NAME       : "Hawkincloud-env"
  DEPLOY_PACKAGE_NAME       : "hawkincloud-app-${{ github.sha }}.zip"
  AWS_REGION_NAME           : "us-east-1"

on:
  push:
    branches:
      - master
jobs:
    my_ci_pipeline:
       runs-on: ubuntu-latest

       steps:
         - name: Git clone our repository
           uses: actions/checkout@v1

         - name: Create ZIP deployment package
           run : zip -r ${{ env.DEPLOY_PACKAGE_NAME }} ./  -x  *.git*

         - name: Configure my AWS Credentils
           uses: aws-actions/configure-aws-credentials@v1
           with:
             aws-access-key-id    : ${{ secrets.MY_AWS_ACCESS_KEY }}
             aws-secret-access-key: ${{ secrets.MY_AWS_SECRET_KEY }}
             aws-region           : ${{ env.AWS_REGION_NAME }}

         - name: Copy our Deployment package to S3 bucket
           run : aws s3 cp ${{ env.DEPLOY_PACKAGE_NAME }} s3://${{ env.EB_PACKAGE_S3_BUCKET_NAME}}/

         - name: Print nice message on completion of CI Pipeline
           run : echo "CI Pipeline part finished successfully"

    my_cd_pipeline:
        runs-on: ubuntu-latest
        needs : [my_ci_pipeline]

        steps:
          - name: Configure my AWS Credentils
            uses: aws-actions/configure-aws-credentials@v1
            with:
             aws-access-key-id    : ${{ secrets.MY_AWS_ACCESS_KEY }}
             aws-secret-access-key: ${{ secrets.MY_AWS_SECRET_KEY }}
             aws-region           : ${{ env.AWS_REGION_NAME }}

          - name : Create new ElasticBeanstalk Application Version
            run : |
             aws elasticbeanstalk create-application-version \
             --application-name ${{ env.EB_APPLICATION_NAME }} \
             --source-bundle S3Bucket="${{ env. EB_PACKAGE_S3_BUCKET_NAME }}",S3Key="${{ env.DEPLOY_PACKAGE_NAME }}" \
             --version-label "Ver-${{ github.sha }}" \
             --description "CommitSHA-${{ github.sha }}"

          - name: Deploy our new Application Version
            run: aws elasticbeanstalk update-environment --environment-name ${{ env.EB_ENVIRONMENT_NAME }} --version-label "Ver-${{ github.sha }}"

          - name: Print nice message on completion of CD Pipeline
            run: echo "CD Pipeline part finished successfully"
```

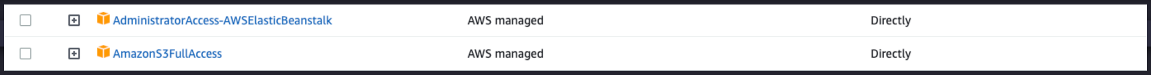






Creating Hawkincloud-env
This will take a few minutes. ..

```
12:18am  Created CloudWatch alarm named:
         awseb-e-zdckxbpt8a-stack-AWSEBCloudwatchAlarmHigh-143L2ADHVVY6P
12:18am  Created CloudWatch alarm named:
         awseb-e-zdckxbpt8a-stack-AWSEBCloudwatchAlarmLow-WK121V6Y8H4Z
12:18am  Created Auto Scaling group policy named:
         arn:aws:autoscaling:us-east-1:783210027625:scalingPolicy:d9d29c21-7be8-4b18-a5ff-24ceca2d0c95:autoScalingGroupName/awseb-e-zdckxbpt8a-stack-AWSEBAutoScalingGroup-1OUDAG12B3095:policyName/awseb-e-zdckxbpt8a-stack-
         AWSEBAutoScalingScaleUpPolicy-XtvaSQb6aZi5
12:18am  Created Auto Scaling group policy named:
         arn:aws:autoscaling:us-east-1:783210027625:scalingPolicy:73e7e272-81ba-4fd3-9d05-96d3d52a42c5:autoScalingGroupName/awseb-e-zdckxbpt8a-stack-AWSEBAutoScalingGroup-1OUDAG12B3095:policyName/awseb-e-zdckxbpt8a-stack-
         AWSEBAutoScalingScaleDownPolicy-z67t8JzMblL4
12:18am  Waiting for EC2 instances to launch. This may take a few minutes.
12:18am  Created Auto Scaling group named:
         awseb-e-zdckxbpt8a-stack-AWSEBAutoScalingGroup-1OUDAG12B3095
12:18am  Environment health has transitioned to Pending. Initialization in progress (running for 23 seconds). There are no instances.
12:17am  Created target group named:
         arn:aws:elasticloadbalancing:us-east-1:783210027625:targetgroup/awseb-AWSEB-WEJYXWRLC83R/4d59a05784fba00a
12:17am  Created security group named:
```



☐	⊞	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Directly
☐	⊞	AmazonS3FullAccess	AWS managed	Directly

4

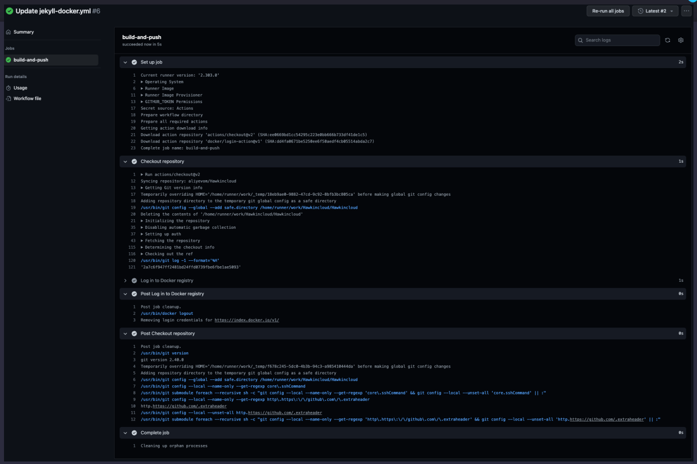
GitHub Actions

Aliyev Omar

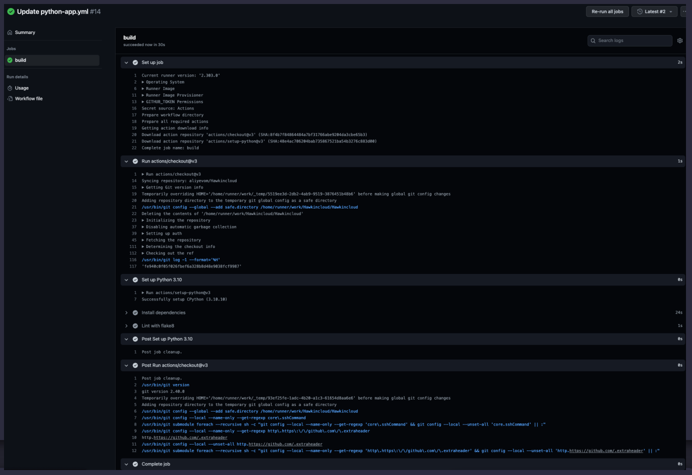
- GitHub Actions is a powerful tool for automating workflows in your software development process.
- With GitHub Actions and Docker, you can easily build and deploy your Python Flask app.
- By defining a YAML file like the one we discussed today, you can automate the entire build and deployment process, saving time and effort.

jekyll-docker.yml	Update jekyll-docker.yml
python-app.yml	Update python-app.yml

✓ Triggering the Workflow
✓ Setting Up the Environment
✓ Installing Dependencies
✓ Linting and Testing
✓ Building and Pushing the Docker Images



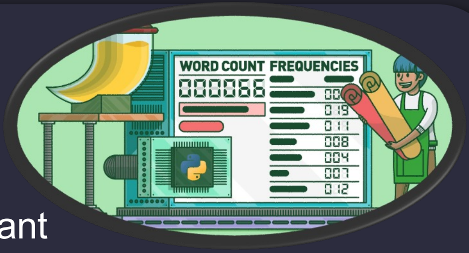
```
build-and-push
succeeded now in 5s

Set up job                                                        2s
Current runner version: '2.303.0'
▶ Operating System
▶ Runner Image
▶ Runner Image Provisioner
▶ GITHUB_TOKEN Permissions
Secret source: Actions
Prepare workflow directory
Prepare all required actions
Getting action download info
Download action repository 'actions/checkout@v2' (SHA:ee0669bd1cc54295c223e0bb666b733df41de1c5)
Download action repository 'docker/login-action@v1' (SHA:dd4fa0671be5250ee6f50aedf4cb05514abda2c7)
Complete job name: build-and-push

Checkout repository                                               1s
▶ Run actions/checkout@v2
Syncing repository: aliyevom/Hawkincloud
▶ Getting Git version info
Temporarily overriding HOME='/home/runner/work/_temp/18eb9ae0-9882-47cd-9c92-8bfb3bc805ca' before making global git config changes
Adding repository directory to the temporary git global config as a safe directory
/usr/bin/git config --global --add safe.directory /home/runner/work/Hawkincloud/Hawkincloud
Deleting the contents of '/home/runner/work/Hawkincloud/Hawkincloud'
▶ Initializing the repository
▶ Disabling automatic garbage collection
▶ Setting up auth
▶ Fetching the repository
▶ Determining the checkout info
▶ Checking out the ref
/usr/bin/git log -1 --format='%H'
'2a7c6f947ff2481bd24ffd0739fbe6fbe1ae5093'

Log in to Docker registry                                         1s

Post Log in to Docker registry                                    0s
Post job cleanup.
/usr/bin/docker logout
Removing login credentials for https://index.docker.io/v1/

Post Checkout repository                                          0s
Post job cleanup.
/usr/bin/git version
git version 2.40.0
Temporarily overriding HOME='/home/runner/work/_temp/f678c245-5dc0-4b3b-94c3-a985410444da' before making global git config changes
Adding repository directory to the temporary git global config as a safe directory
/usr/bin/git config --global --add safe.directory /home/runner/work/Hawkincloud/Hawkincloud
/usr/bin/git config --local --name-only --get-regexp core\.sshCommand
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core\.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
/usr/bin/git config --local --name-only --get-regexp http\.https\:\/\/github\.com\/\.extraheader
http.https://github.com/.extraheader
/usr/bin/git config --local --unset-all http.https://github.com/.extraheader
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http\.https\:\/\/github\.com\/\.extraheader' && git config --local --unset-all 'http.https://github.com/.extraheader' || :"

Complete job                                                      0s
Cleaning up orphan processes
```



```
build
succeeded now in 30s

Set up job                                                        2s
Current runner version: '2.303.0'
▶ Operating System
▶ Runner Image
▶ Runner Image Provisioner
▶ GITHUB_TOKEN Permissions
Secret source: Actions
Prepare workflow directory
Prepare all required actions
Getting action download info
Download action repository 'actions/checkout@v3' (SHA:8f4b7f84864484a7bf31766abe9204da3cbe65b3)
Download action repository 'actions/setup-python@v3' (SHA:48e4ac706204bab735867521ba54b3276c883d00)
Complete job name: build

Run actions/checkout@v3                                           1s
▶ Run actions/checkout@v3
Syncing repository: aliyevom/Hawkincloud
▶ Getting Git version info
Temporarily overriding HOME='/home/runner/work/_temp/5519ee3d-2db2-4ab9-9519-3876451b48b6' before making global git config changes
Adding repository directory to the temporary git global config as a safe directory
/usr/bin/git config --global --add safe.directory /home/runner/work/Hawkincloud/Hawkincloud
Deleting the contents of '/home/runner/work/Hawkincloud/Hawkincloud'
▶ Initializing the repository
▶ Disabling automatic garbage collection
▶ Setting up auth
▶ Fetching the repository
▶ Determining the checkout info
▶ Checking out the ref
/usr/bin/git log -1 --format='%H'
'fe940c0f05f026fbef6a328b8d48e9038fcf9987'

Set up Python 3.10                                                0s
▶ Run actions/setup-python@v3
Successfully setup CPython (3.10.10)

Install dependencies                                              24s

Lint with flake8                                                  1s

Post Set up Python 3.10                                           0s
Post job cleanup.

Post Run actions/checkout@v3                                      0s
Post job cleanup.
/usr/bin/git version
git version 2.40.0
Temporarily overriding HOME='/home/runner/work/_temp/93ef25fe-1adc-4b20-a1c3-61654d8aa6e6' before making global git config changes
Adding repository directory to the temporary git global config as a safe directory
/usr/bin/git config --global --add safe.directory /home/runner/work/Hawkincloud/Hawkincloud
/usr/bin/git config --local --name-only --get-regexp core\.sshCommand
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core\.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
/usr/bin/git config --local --name-only --get-regexp http\.https\:\/\/github\.com\/\.extraheader
http.https://github.com/.extraheader
/usr/bin/git config --local --unset-all http.https://github.com/.extraheader
/usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http\.https\:\/\/github\.com\/\.extraheader' && git config --local --unset-all 'http.https://github.com/.extraheader' || :"

Complete job                                                      0s
```

Michael Hernandez

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:

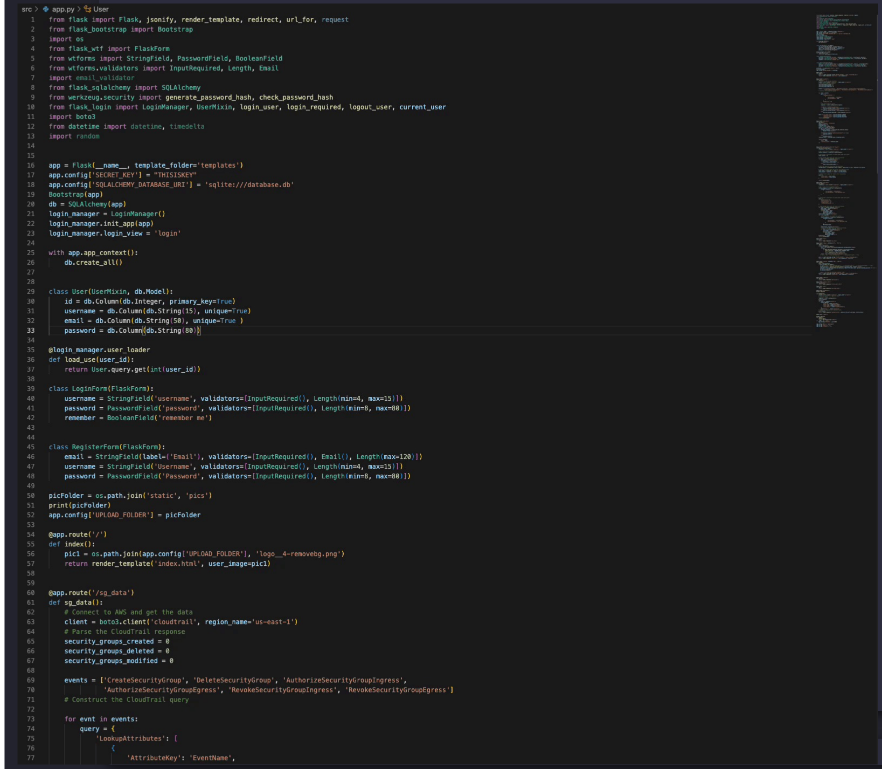
📍 Displaying meaningful metrics from the AWS Cloudtrail, operation as an essential analytics tool for personal or business use.

With just over 2 months into the initiation 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

We continue to meet more than once a week, either virtually or in person. Substantial progress on both the frontend and backend have lead HawkinCloud in the right direction.

Haider Zaheer

HawkinCloud: Proceeding

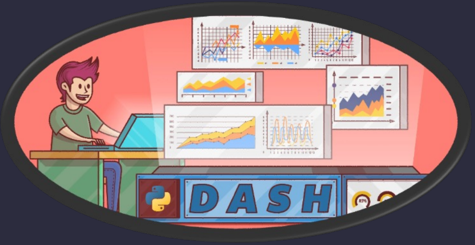


```python
from flask import Flask, jsonify, render_template, redirect, url_for, request
from flask_bootstrap import Bootstrap
import os
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, BooleanField
from wtforms.validators import InputRequired, Length, Email
import email_validator
from flask_sqlalchemy import SQLAlchemy
from werkzeug.security import generate_password_hash, check_password_hash
from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, current_user
import boto3
from datetime import datetime, timedelta
import random


app = Flask(__name__, template_folder='templates')
app.config['SECRET_KEY'] = "THISISKEY"
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
Bootstrap(app)
db = SQLAlchemy(app)
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

with app.app_context():
    db.create_all()


class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(15), unique=True)
    email = db.Column(db.String(50), unique=True )
    password = db.Column(db.String(80))

@login_manager.user_loader
def load_use(user_id):
    return User.query.get(int(user_id))

class LoginForm(FlaskForm):
    username = StringField('username', validators=[InputRequired(), Length(min=4, max=15)])
    password = PasswordField('password', validators=[InputRequired(), Length(min=8, max=80)])
    remember = BooleanField('remember me')


class RegisterForm(FlaskForm):
    email = StringField(label=('Email'), validators=[InputRequired(), Email(), Length(max=120)])
    username = StringField('Username', validators=[InputRequired(), Length(min=4, max=15)])
    password = PasswordField('Password', validators=[InputRequired(), Length(min=8, max=80)])

picFolder = os.path.join('static', 'pics')
print(picFolder)
app.config['UPLOAD_FOLDER'] = picFolder

@app.route('/')
def index():
    pic1 = os.path.join(app.config['UPLOAD_FOLDER'], 'logo__4-removebg.png')
    return render_template('index.html', user_image=pic1)


@app.route('/sg_data')
def sg_data():
    # Connect to AWS and get the data
    client = boto3.client('cloudtrail', region_name='us-east-1')
    # Parse the CloudTrail response
    security_groups_created = 0
    security_groups_deleted = 0
    security_groups_modified = 0

    events = ['CreateSecurityGroup', 'DeleteSecurityGroup', 'AuthorizeSecurityGroupIngress',
              'AuthorizeSecurityGroupEgress', 'RevokeSecurityGroupIngress', 'RevokeSecurityGroupEgress']
    # Construct the CloudTrail query

    for evnt in events:
        query = {
            'LookupAttributes': [
                {
                    'AttributeKey': 'EventName',
```

- The code is a Flask web application with routes for displaying AWS data.
- It uses Boto3 to connect to AWS services and retrieve data such as CloudTrail and EC2.
- It also includes a login system using Flask-Login and SQLAlchemy for database management.
- The application utilizes Bootstrap for styling and includes forms for user registration and login.
- The routes return JSON data to be displayed on the web page using JavaScript.

Nephthro Pierre

HawkinCloud Dashboard



Sketching out the layout of the dashboard will help you to visualize how the different elements of the dashboard will be placed. You can use pen and paper or a wireframing tool to create a rough layout.

Flask has many extensions that you can use to add functionality to your dashboard. Some popular extensions for Flask include Flask-Admin and Flask-Bootstrap.

Ensure that your dashboard is mobile-friendly and responsive. Most users will access the dashboard from a mobile device, and a responsive design will ensure that the dashboard is easily accessible from any device.

```
        88                                 88
  ,d    88                                 88
  88    88                                 88
MM88MMM 88,dPPYba,  ,adPPYYba, 8b,dPPYba,  88   ,d8 8b       d8  ,adPPYba,  88       88
  88    88P'    "8a ""     `Y8 88P'   `"8a 88 ,a8"  `8b     d8' a8"     "8a 88       88
  88    88       88 ,adPPPPP88 88       88 8888[     `8b   d8'  8b       d8 88       88
  88,   88       88 88,    ,88 88       88 88`"Yba,   `8b,d8'   "8a,   ,a8" "8a,   ,a88
  "Y888 88       88 `"8bbdP"Y8 88       88 88   `Y8a    Y88'     `"YbbdP"'   `"YbbdP'Y8
                                                        d8'
                                                       d8'
```

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