

Unleash the Power of AI: Streamlining Airflow DAG Development with AI-Driven Automation

Sriharsh Adari, Solutions Architect , AWS

Jeetendra Vaidya, Solutions Architect , AWS

Joe Morotti, Solutions Architect, AWS



Agenda

- The Productivity Challenge
- Opportunities with Generative AI
- Data Engineer and Analysts – A Day in life.
- Generative AI on AWS
- Demo's
- Q&A

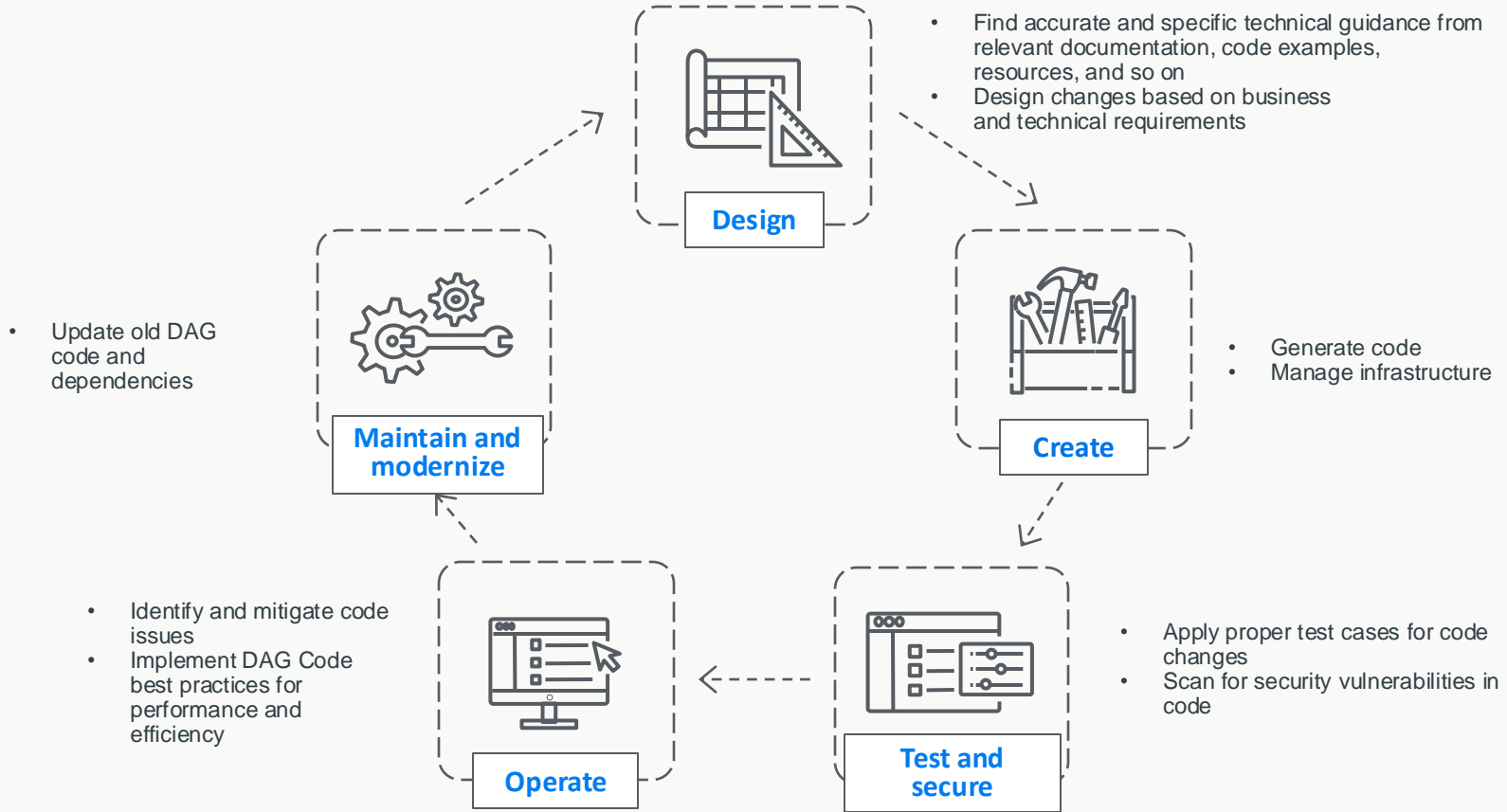


The Productivity Challenge

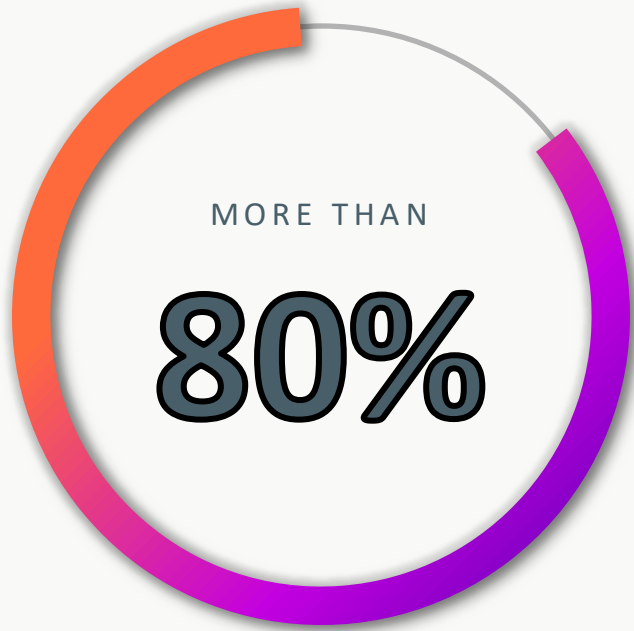
- Manual processes hinder productivity.
- Businesses seek ways to maximize output and efficiency.
- Resource constraints.
- Inadequate training.
- Unclear goals and unrealistic goals.



Where are Data Engineers & Analysts spending time with Airflow?



Opportunities with generative AI



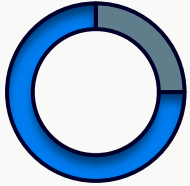
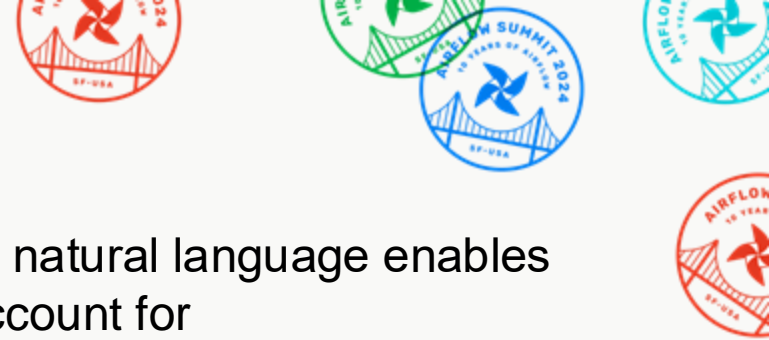
ACCORDING TO GARTNER, INC.®

of enterprises will have used generative AI APIs or deployed generative AI-enabled apps by 2026¹

¹ Gartner, "More than 80% of Enterprises," October 11, 2023.



Opportunities with generative AI



Generative AI's ability to understand natural language enables automation for work activities that account for

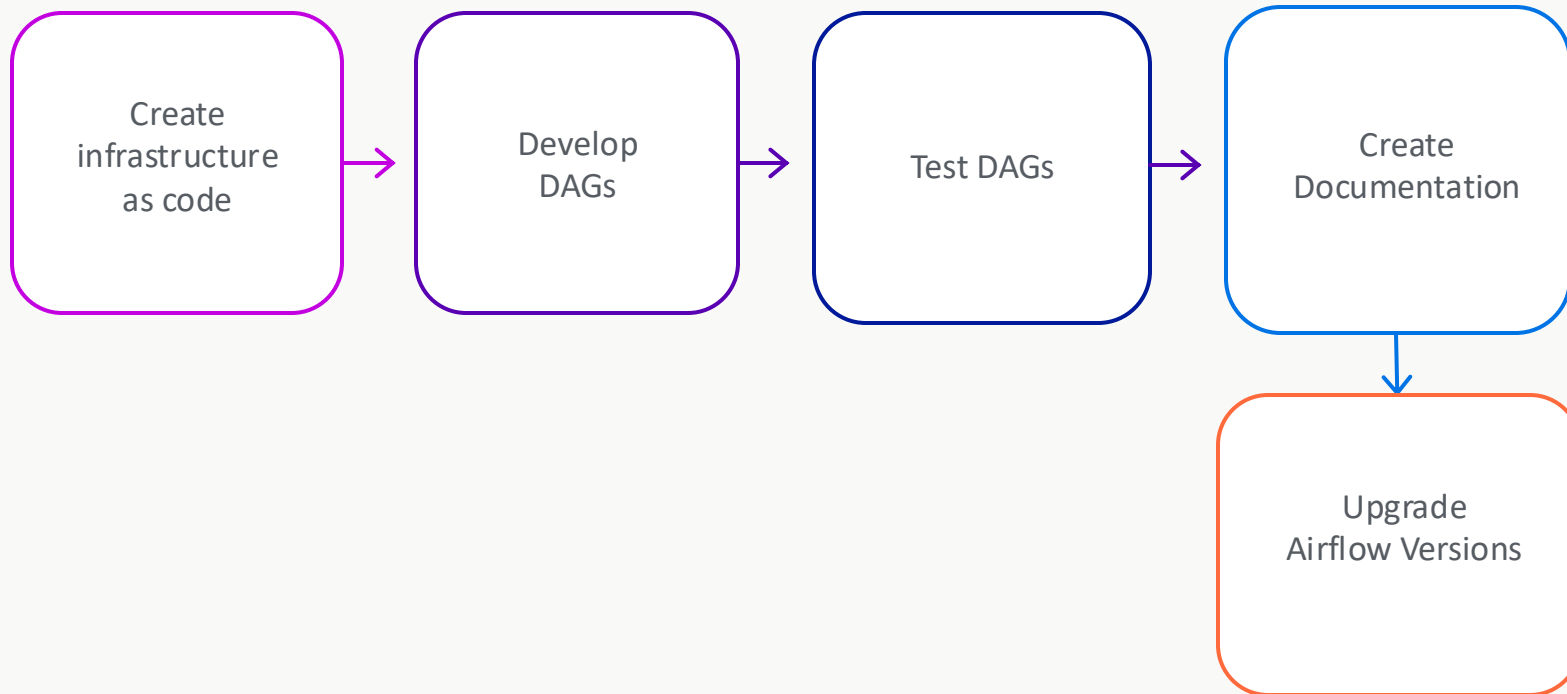
25% of total work time



Imagine a generative AI powered assistant that saves you

2 hours every day

How AI drives Productivity gains?



Generative AI Stack

APPLICATIONS THAT USE LLMs AND OTHER FMs



Amazon Q
Business



Amazon Q
Developer



Amazon Q in
QuickSight



Amazon Q in
Connect

TOOLS TO BUILD WITH LLMs AND OTHER FMs



Amazon Bedrock

Guardrails | Agents | Customization capabilities

INFRASTRUCTURE FOR FM TRAINING AND INFERENCE



GPUs



AWS
Trainium



AWS
Inferentia



Amazon
SageMaker



Amazon EC2
UltraClusters



Elastic Fabric
Adapter (EFA)



Amazon EC2 Capacity
Blocks



AWS
Nitro



AWS
Neuron

Amazon Q Developer



Reimagines the experience across the entire software development lifecycle (SDLC)

Helps developers and IT professionals build and manage secure, scalable, and highly available applications

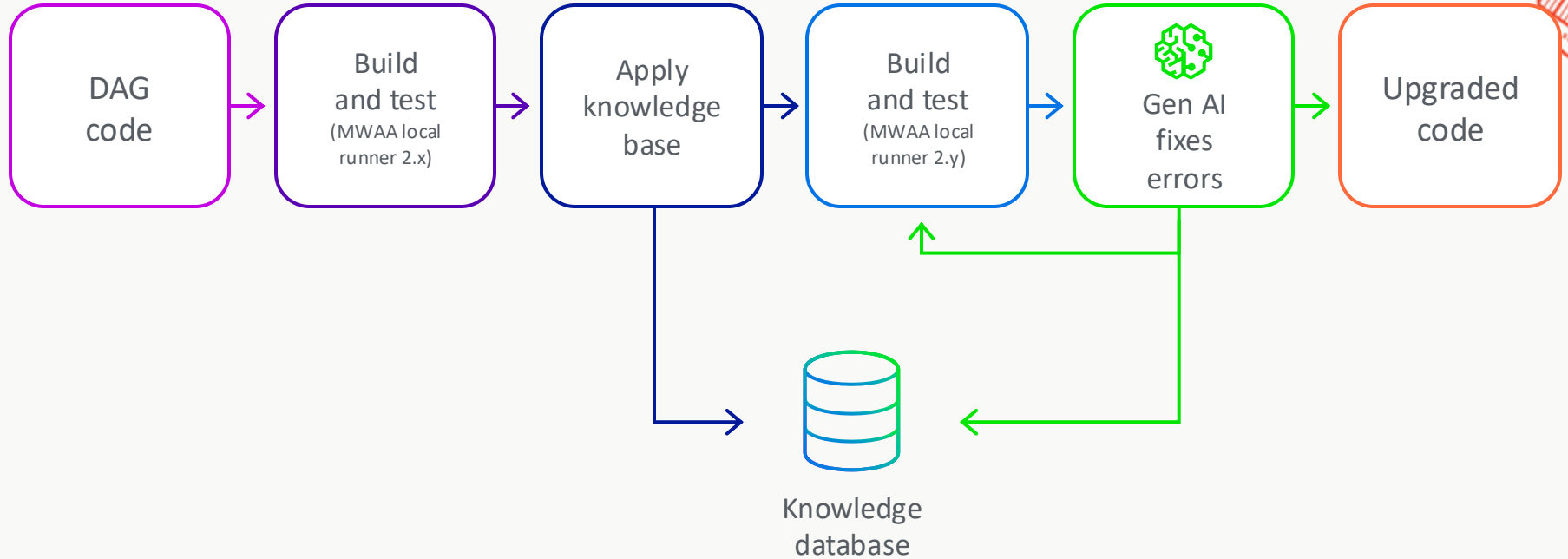
Helps you write, debug, test, optimize, and upgrade your code faster

Converses with you to explore new AWS capabilities, learn unfamiliar technologies, and architect solutions

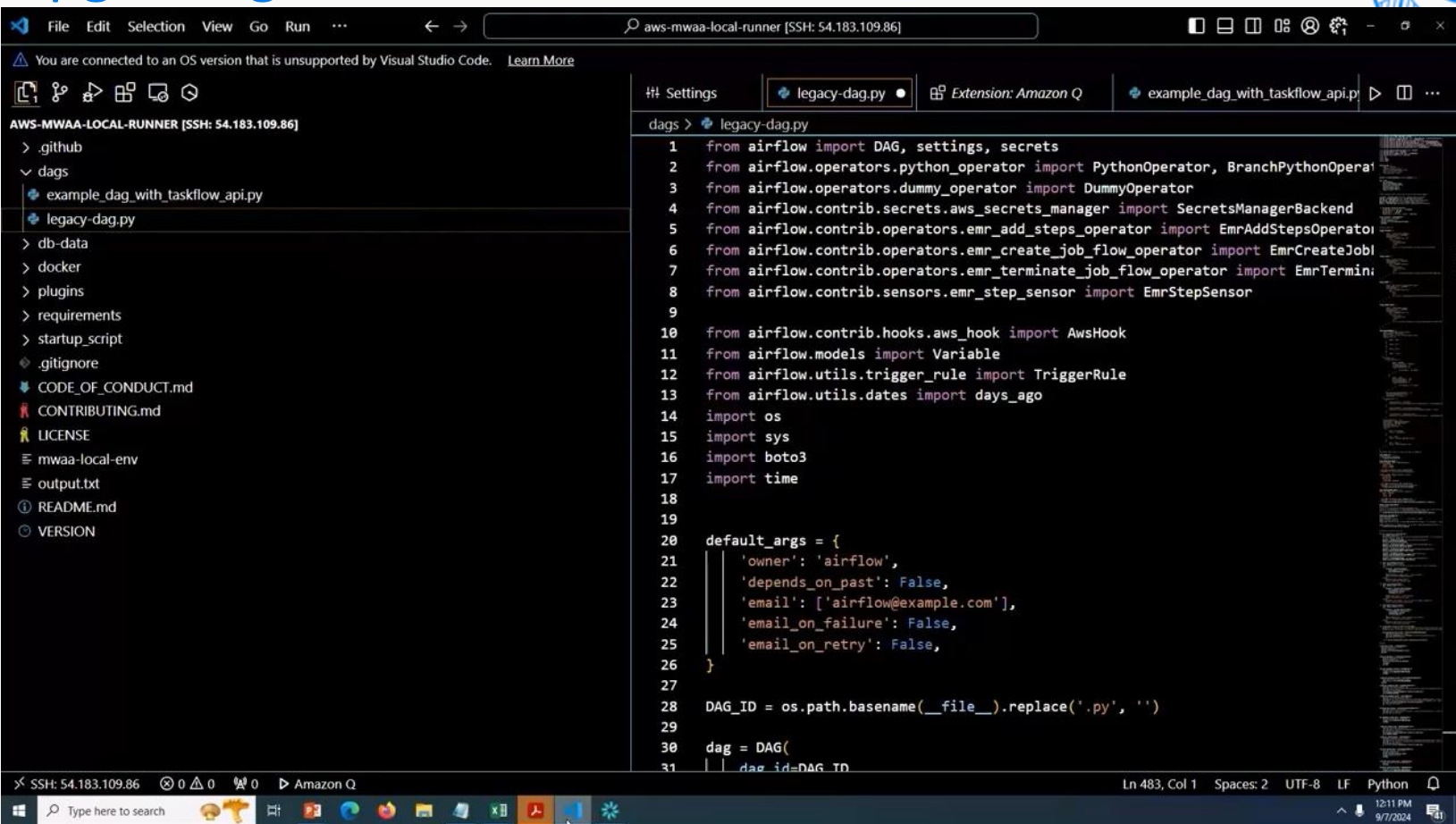
Amazon Q is built with security and privacy in mind from the start, making it easier for organizations to use generative AI safely.



Challenge: Airflow version upgrades



Upgrading Airflow Versions



The screenshot displays a Visual Studio Code editor window with the following components:

- File Explorer (Left):** Shows the project structure for 'AWS-MWAA-LOCAL-RUNNER [SSH: 54.183.109.86]'. The 'dags' directory is expanded, showing files like 'example_dag_with_taskflow_api.py' and 'legacy-dag.py'.
- Settings Pane (Top):** Shows the active file 'legacy-dag.py' and the installed extension 'Amazon Q'.
- Code Editor (Center):** Displays the content of 'legacy-dag.py'. The code defines an Airflow DAG with the following imports and logic:

```
1 from airflow import DAG, settings, secrets
2 from airflow.operators.python_operator import PythonOperator, BranchPythonOperator
3 from airflow.operators.dummy_operator import DummyOperator
4 from airflow.contrib.secrets.aws_secrets_manager import SecretsManagerBackend
5 from airflow.contrib.operators.emr_add_steps_operator import EmrAddStepsOperator
6 from airflow.contrib.operators.emr_create_job_flow_operator import EmrCreateJobFlowOperator
7 from airflow.contrib.operators.emr_terminate_job_flow_operator import EmrTerminateJobFlowOperator
8 from airflow.contrib.sensors.emr_step_sensor import EmrStepSensor
9
10 from airflow.contrib.hooks.aws_hook import AwsHook
11 from airflow.models import Variable
12 from airflow.utils.trigger_rule import TriggerRule
13 from airflow.utils.dates import days_ago
14 import os
15 import sys
16 import boto3
17 import time
18
19
20 default_args = {
21     'owner': 'airflow',
22     'depends_on_past': False,
23     'email': ['airflow@example.com'],
24     'email_on_failure': False,
25     'email_on_retry': False,
26 }
27
28 DAG_ID = os.path.basename(__file__).replace('.py', '')
29
30 dag = DAG(
31     dag_id=DAG_ID,
```
- Status Bar (Bottom):** Shows the current file path 'Ln 483, Col 1', encoding 'UTF-8', and language 'Python'.

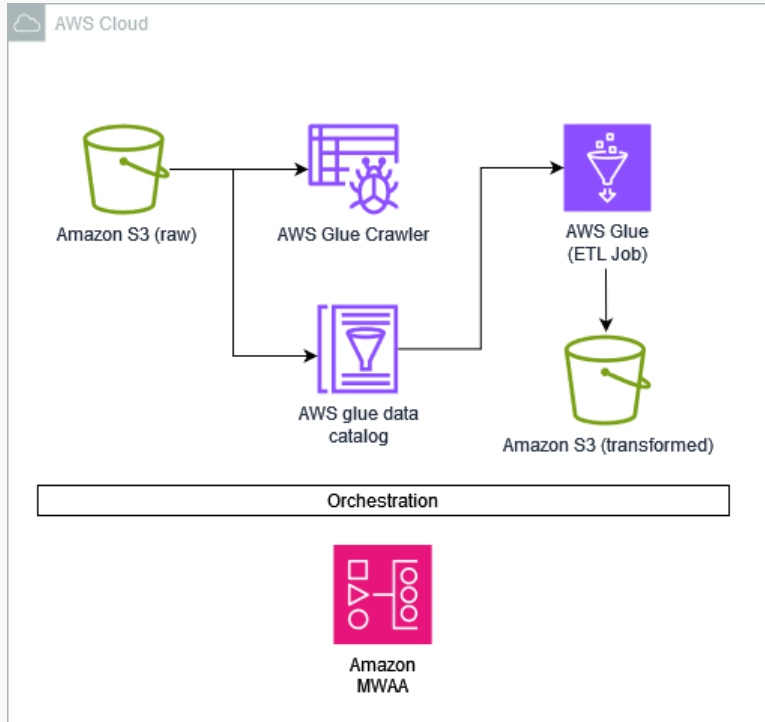
AWS Services Introductions

- Amazon MWAA (Managed Workflow for Apache Airflow)
 - A Managed service for Apache Airflow, making it easy for data engineers and data scientists to invoke data processing workflows on AWS.
 - Easy to set up and Maintain with High availability.

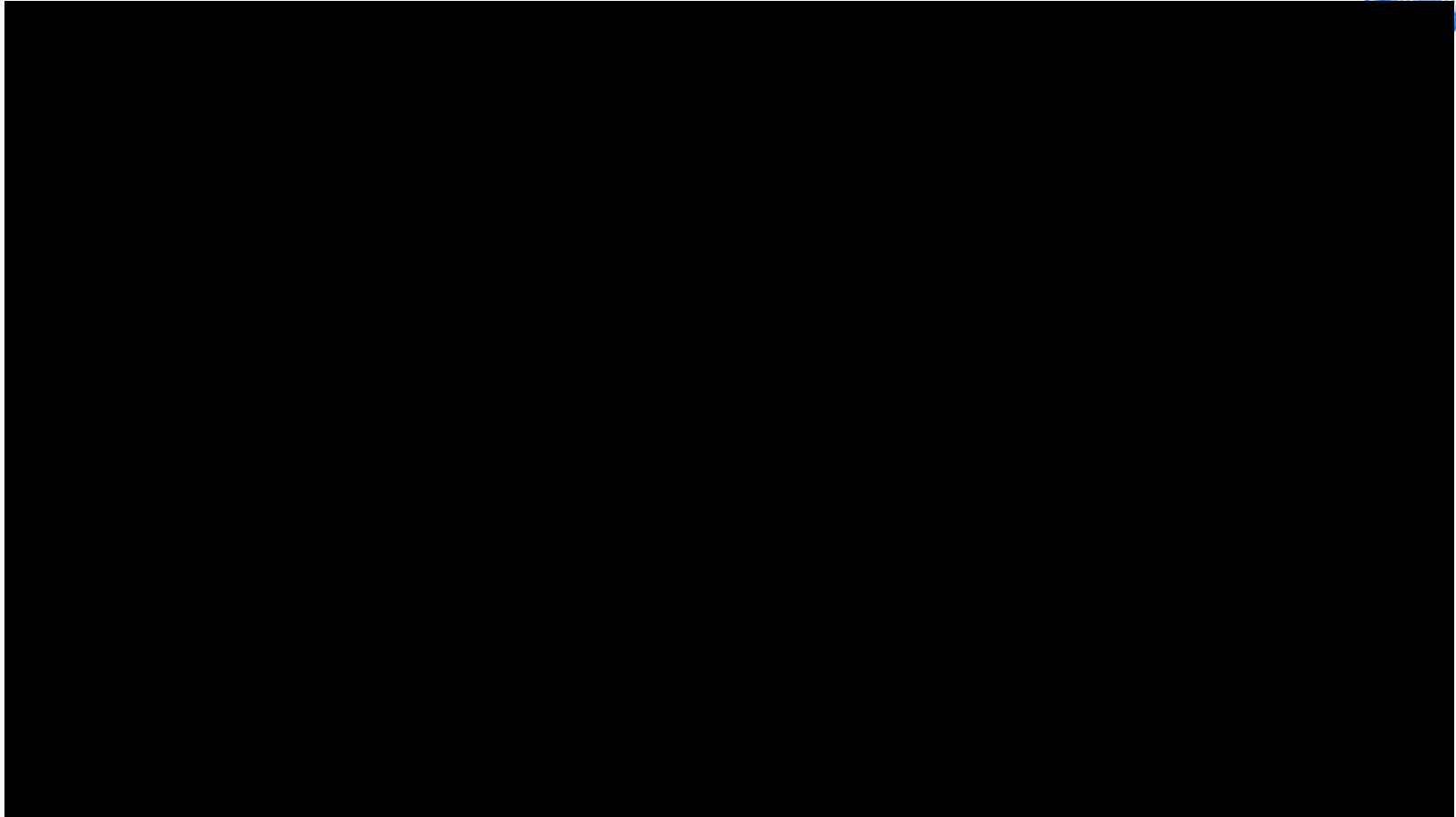
- AWS Glue
 - AWS Glue is a serverless data integration service that makes data preparation simpler, faster, and cheaper.
 - Cost effective, serverless, and scalable

Demo – Workflow Creation

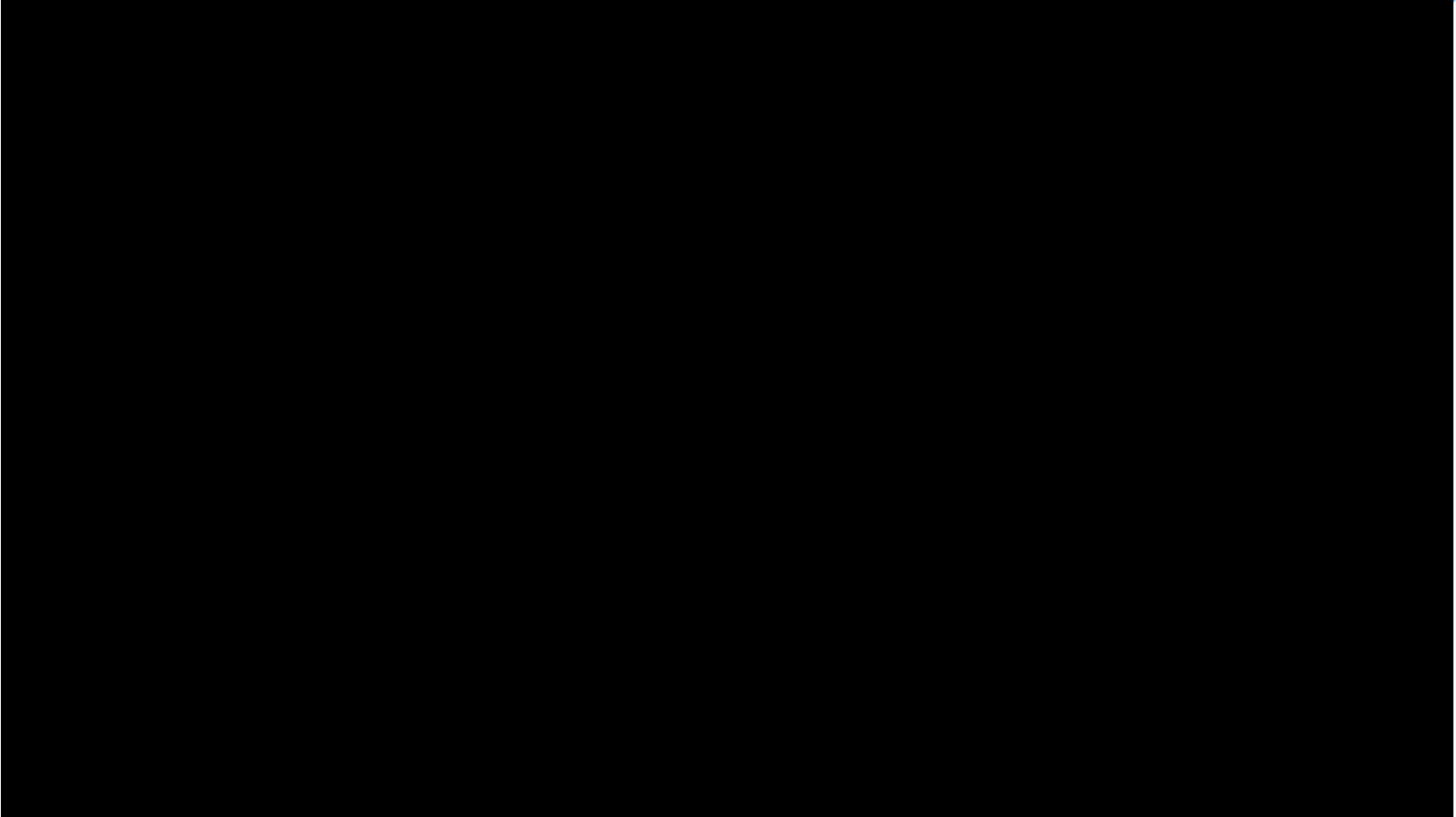
- Architecture



Creating IaC to deploy Apache Airflow



Creating DAGs and boiler plate code



Demystifying DAGs: Building Clear Documentation

```
EXPLORER: MWAA
└─ DAGS
  └─ process_covid_data_us.py

process_covid_data_us.py 7
1  """ Copyright Amazon.com, Inc. or its affiliates.
2  All Rights Reserved.SPDX-License-Identifier: MIT-0"""
3
4  from airflow import DAG
5  from airflow.decorators import task
6  from airflow.models import Variable
7  import pendulum
8
9  ## Top Level Imports ##
10 import io
11 import boto3
12 import pandas as pd
13 import numpy as np
14
15 ## Top Level Variables ##
16 S3_BUCKET = Variable.get("S3_BUCKET_NAME", "")
17 INPUT_KEY = Variable.get("INPUT_KEY", "")
18 OUTPUT_KEY = Variable.get("OUTPUT_KEY", "")
19
20 COUNTRY = 'US'
21
22 with DAG(
23     dag_id="top_level_code_us",
24     schedule_interval=None,
25     start_date=pendulum.datetime(2023, 1, 1, tz="UTC"),
26     catchup=False,
27     tags=["Module1"],
28 ) as dag:
29
30     @task()
31     def process_covid_data():
32
33         s3 = boto3.client('s3')
34         response = s3.get_object(Bucket=S3_BUCKET, Key=INPUT_KEY)
35
36         status = response.get("ResponseMetadata", {}).get("HTTPStatusCode")
37
38         if status == 200:
39             print(f"Successful S3 get_object response. Status - {status}")
40             df = pd.read_csv(response.get("Body"))
41             df.drop(['Last_Update', 'Lat', 'Long_', 'Recovered', 'Active', 'Combined_Key', 'Incident_Rate', 'Case_Fatality_Ratio'], axis=1, inplace=True)
42             df.rename(columns={'Province_State': 'State', 'Country_Region': 'Country'}, inplace=True)
43             df = df.fillna("NA")
44             df2 = df[df['Country'] == COUNTRY]
45             df3 = df2.groupby('State')[['Confirmed', 'Deaths']].sum().reset_index()
46             df3['IsXorMore'] = np.where((df3['Deaths'] > 10000), 1, 0)
47             df3.sort_values(by='State')
48
49             with io.StringIO() as csv_buffer:
50                 df3.to_csv(csv_buffer, index=False)
51
52                 response = s3.put_object(
```


Embracing AI for a Productive Future for Airflow

- Key Benefits of AI for Productivity.
- Measure Impact.
- Security.
- Review and test generated code.
- Customized Code recommendations based on Org guidelines.



Questions?



Scan me!

