

Ace the Exam Series[®]

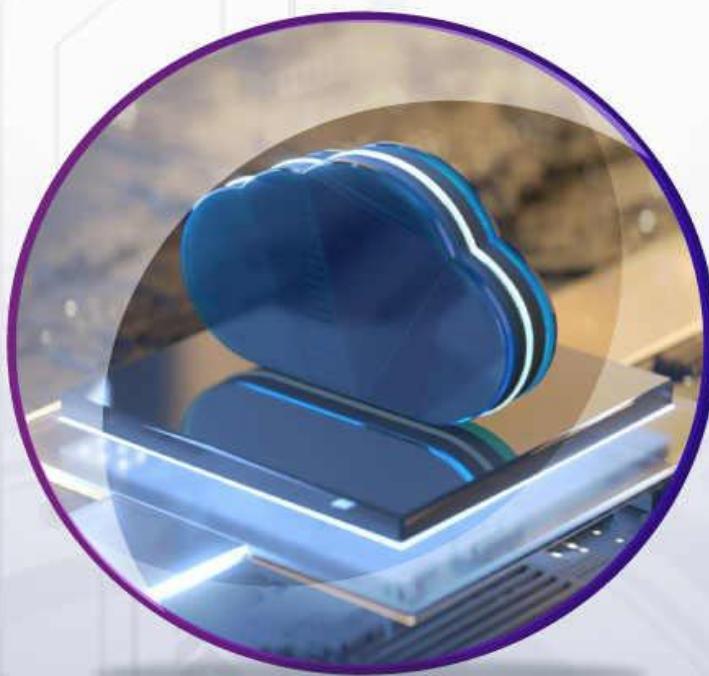
FIRST EDITION

2022

HANDS-ON LABS

MLS-C01: AWS CERTIFIED MACHINE LEARNING SPECIALTY

BASED ON
REAL WORLD CASE STUDIES



IPSpecialist
Sharpen Your Skills for the Digital Future

Hands-On Labs

MLS-CO1: AWS Certified Machine Learning Specialty

Based on Real-World Case Studies

Lab Guide

Document Control

Proposal Name : AWS Certified Machine Learning Specialty – Hands-on Labs
Document Edition : First Edition
Document Release Date : 16th August 2022
Reference : MLS-C01

Copyright © 2022 IPSpecialist LTD.

Registered in England and Wales

Company Registration No: 10883539

Registration Office at: Office 32, 19-21 Crawford Street, London W1H 1PJ, United Kingdom

www.ipspecialist.net

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without the written permission from IPSpecialist LTD, except for the inclusion of brief quotations in a review.

Feedback:

If you have any comments regarding the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at info@ipspecialist.net

Please make sure to include the book's title and ISBN in your message.

About IPSpecialist

[IPSPECIALIST](#) LTD. IS COMMITTED TO EXCELLENCE AND DEDICATED TO YOUR SUCCESS.

Our philosophy is to treat our customers like family. We want you to succeed, and we are willing to do everything possible to help you make it happen. We have the proof to back up our claims. We strive to accelerate billions of careers with great courses, accessibility, and affordability. We believe that continuous learning and knowledge evolution are the most important things to keep re-skilling and up-skilling the world.

Planning and creating a specific goal are where IPSpecialist helps. We can create a career track that suits your visions as well as develop the competencies you need to become a professional Network Engineer. We can also assist you with the execution and evaluation of your proficiency level, based on the career track you choose, as they are customized to fit your specific goals.

We help you STAND OUT from the crowd through our detailed IP training content packages.

Course Features:

- ❖ Self-Paced Learning
 - Learn at your own pace and in your own time
- ❖ Covers Complete Exam Blueprint
 - Prep-up for the exam with confidence
- ❖ Case Study Based Learning
 - Relate the content with real-life scenarios
- ❖ Subscriptions that Suit You
 - Get more and pay less with IPS subscriptions
- ❖ Career Advisory Services
 - Let the industry experts plan your career journey
- ❖ Virtual Labs to test your skills
 - With IPS vRacks, you can evaluate your exam preparations
- ❖ Practice Questions
 - Practice questions to measure your preparation standards
- ❖ On Request Digital Certification
 - On request digital certification from IPSpecialist LTD.

About the Authors:

This book has been compiled with the help of multiple professional engineers. These engineers specialize in different fields. For example, Networking, Security, Cloud, Big Data, IoT, etc. Each engineer develops content in its specialized field that is compiled to form a comprehensive certification guide.

About the Technical Reviewers:

Nouman Ahmed Khan

AWS-Architect, CCDE, CCIEX5 (R&S, SP, Security, DC, Wireless), CISSP, CISA, CISM is a Solution Architect working with a major telecommunication provider in Qatar. He works with enterprises, mega-projects, and service providers, helping them select the best-fit technology solutions. He also works closely as a consultant to understand customer business processes and helps select an appropriate technology strategy to support business goals. He has more than 14 years of experience working in Pakistan/Middle-East & UK. He holds a Bachelor of Engineering Degree from NED University, Pakistan, and an M.Sc. in Computer Networks from the UK.

Abubakar Saeed

Abubakar Saeed has more than twenty-five years of experience, Managing, Consulting, Designing, and implementing large-scale technology projects, extensive experience heading ISP operations, solutions integration, heading Product Development, Presales, and Solution Design. Emphasizing adhering to Project timelines and delivering as per customer expectations, he always leads the project in the right direction with his innovative ideas and excellent management.

Dr. Fahad Abdali

Dr. Fahad Abdali is a seasoned leader with extensive experience managing and growing software development teams in high-growth start-ups. He is a business entrepreneur with more than 18 years of experience in management and marketing. He holds a Bachelor's Degree from NED University of Engineering and Technology and a Doctor of Philosophy (Ph.D.) from the University of Karachi.

Mehwish Jawed

Mehwish Jawed is working as a Senior Research Analyst. She holds a Master's and Bachelors of Engineering degree in Telecommunication Engineering from NED University of Engineering and Technology. She also worked under the supervision of HEC Approved supervisor. She has more than three published papers, including both conference and journal papers. She has a great knowledge of TWDM Passive Optical Network (PON). She also worked as a Project Engineer, Robotic Trainer in a private institute and has research skills in the field of communication networks. She has both technical knowledge and industry-sounding information, which she utilizes effectively when needed. She also has expertise in cloud platforms like AWS, GCP, Oracle, and Microsoft Azure.

Syeda Fariha Ashrafi

Syeda Fariha Ashrafi is working as a technical content developer. She has completed bachelor's degree in telecommunication engineering from NED University of Engineering and Technology. She has also completed the CCNA (Routing and Switching) course. During her bachelor's program, she has worked on the project "Smart metering using PLC (Power Line Communication).

Free Resources

For Free Resources: Please visit our website and register to access your desired Resources Or contact us at: helpdesk@ipspecialist.net

Career Report: This report is a step-by-step guide for a novice who wants to develop his/her career in the field of computer networks. It answers the following queries:

- What are the current scenarios and future prospects?
- Is this industry moving towards saturation, or are new opportunities knocking at the door?
- What will the monetary benefits be?
- Why get certified?
- How to plan, and when will I complete the certifications if I start today?
- Is there any career track that I can follow to accomplish the specialization level?

Furthermore, this guide provides a comprehensive career path towards being a specialist in networking and highlights the tracks needed to obtain certification.

IPS Personalized Technical Support for Customers: Good customer service means helping customers efficiently, in a friendly manner. It is essential to be able to handle issues for customers and do your best to ensure they are satisfied. Providing good service is one of the most important things

that can set our business apart from the others of its kind.

Excellent customer service will result in attracting more customers and attain maximum customer retention.

IPS offers personalized TECH support to its customers to provide better value for money. If you have any queries related to technology and labs, you can simply ask our technical team for assistance via Live Chat or Email.

Our Products

Study Guides

IPSpecialist Study Guides are the ideal guides to developing the hands-on skills necessary to pass the exam. Our workbooks cover the official exam blueprint and explain the technology with real-life case study-based labs. The content covered in each workbook consists of individually focused technology topics presented in an easy-to-follow, goal-oriented, step-by-step approach. Every scenario features detailed breakdowns and thorough verifications to help you completely understand the task and associated technology.

We extensively used mind maps in our workbooks to visually explain the technology. Our workbooks have become a widely used tool to learn and remember information effectively.

vRacks

Our highly scalable and innovative virtualized lab platforms let you practice the IPSpecialist Study Guide at your own time and your own place as per your convenience.

Exam-Cram Notes

Our Exam Cram notes are concise bundling of condensed notes of the complete exam blueprint. It is an ideal and handy document to help you remember the most important technology concepts related to the certification exam.

Practice Questions

IP Specialists' Practice Questions are dedicatedly designed from a certification exam perspective. The collection of these questions from our Study Guides is prepared to keep the exam blueprint in mind, covering important and necessary topics. It's an ideal document to practice and revise your certification.

Hans on Labs

IPSpecialist Hands-on Labs are the fastest and easiest way to learn the real-world use cases. These

labs are carefully designed to prepare you for the certification exams and your next job role. Whether you are starting to learn technology or solving a real-world scenario, our labs will help you learn the core concepts in no time.

IPSpecialist self-paced labs were designed by subject matter experts and provide an opportunity to use products in a variety of pre-designed scenarios and common use cases, giving you hands-on practice in a simulated environment to help you gain confidence. You have the flexibility to choose from topics and products about which you want to learn more.

Content at a glance

[Course Introduction \(MLS-C01\)](#)

[Lab 01: Use Amazon Rekognition](#)

[Lab 02: Creating a Lambda Function](#)

[Lab 03: Build A Serverless Webpage with API Gateway and AWS Lambda](#)

[Lab 04: Streaming Data Collection](#)

[Lab 05: Performing Real-Time Data Analysis with Kinesis](#)

[Lab 06: Data Analysis & Visualization](#)

[Lab 07: Modeling](#)

[Lab 08: Introducing Jupyter Notebooks \(AWS SageMaker\)](#)

[Lab 09: Algorithms](#)

[Lab 10: Understanding CloudFormation Template Anatomy](#)

[Lab 11: Creating a TensorFlow Image Classifier in AWS SageMaker](#)

[Lab 12: Creating an MXNet Image Classifier in AWS SageMaker](#)

[Lab 13: Creating a scikit-learn Random Forest Classifier in AWS SageMaker](#)

[Lab 14: Evaluation and Optimization](#)

[Lab 15: Categorizing Uploaded Data Using AWS Step Functions](#)

[Lab 16: Coordinating AI Services with Step Functions](#)

[About Our Products](#)

Table of Contents

Course Introduction (MLS-C01)

Certification Approved Skills

Recommended AWS Knowledge

Amazon Machine Learning

Machine Learning Concepts

Lab 01: Use Amazon Rekognition

Service Introduction

Case Study Enterprise Healthcare - Cleveland Medical Center

Background

Business Challenge

Proposed Solution

Lab Diagram

Implementation Steps

Solution

Lab 02: Creating a Lambda Function

Service Introduction

Case Study Enterprise Healthcare - Cleveland Medical Center

Business Challenge

Proposed Solution

Lab Diagram

Implementation Steps

Solution

Lab 03: Build A Serverless Webpage with API Gateway and AWS Lambda

Service Introduction

Case Study Enterprise Educational Services – Abhay Engineering College (AEC)

Background

Business Challenge

Proposed Solution

Lab Diagram

Implementation Steps

Solution

Lab 04: Streaming Data Collection

Service Introduction

Kinesis Data Streams

[Kinesis Data Analytics](#)

[Kinesis Data Firehose](#)

[AWS Lambda](#)

[Amazon Simple Storage Service S3](#)

[AWS CloudFormation](#)

[AWS Identity and Access Management \(IAM\)](#)

[Case Study Enterprise Biopharmaceutical – Neumora Therapeutics](#)

[Background](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

[Lab 05: Performing Real-Time Data Analysis with Kinesis](#)

[Service Introduction](#)

[Kinesis Data Firehose](#)

[Kinesis Data Analytics](#)

[Amazon Simple Storage Service S3](#)

[Amazon Elastic Compute Cloud \(EC2\)](#)

[Case Study Enterprise Biopharmaceutical – Neumora Therapeutics](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

[Lab 06: Data Analysis & Visualization](#)

[Service Introduction](#)

[AWS QuickSight](#)

[AWS SageMaker](#)

[AWS Simple Storage Service S3](#)

[Case Study Enterprise Public Service – IPS](#)

[Background](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

[Lab 07: Modeling](#)

[Service Introduction](#)

[AWS QuickSight](#)

[AWS SageMaker](#)

[AWS Simple Storage Service S3](#)

[Case Study Enterprise EMRs – Savana](#)

[Background](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

[Lab 08: Introducing Jupyter Notebooks \(AWS SageMaker\)](#)

[Service Introduction](#)

[Case Study Enterprise EMRs – Savana](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

[Lab 09: Algorithms](#)

[Service Introduction](#)

[XGBoost](#)

[Linear Learner Algorithm](#)

[Case Study Enterprise Financial Service– Kasasa](#)

[Background](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

Lab 10: Understanding CloudFormation Template Anatomy

[Service Introduction](#)

[Case Study Security Products Manufacturers – Foxis Security Solutions](#)

[Background](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

Lab 11: Creating a TensorFlow Image Classifier in AWS SageMaker

[Service Introduction](#)

[Case Study Enterprise Financial Service– Kasasa](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

Lab 12: Creating an MXNet Image Classifier in AWS SageMaker

[Service Introduction](#)

[Case Study Enterprise Financial Service– Kasasa](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

Lab 13: Creating a scikit-learn Random Forest Classifier in AWS SageMaker

[Service Introduction](#)

[Case Study Enterprise Financial Service– Kasasa](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

Lab 14: Evaluation and Optimization

Service Introduction

[AWS SageMaker](#)

[AWS Simple Storage Service \(S3\)](#)

[AWS CloudWatch](#)

Case Study Enterprise Retail – LexisNexis Legal & Professional

[Background](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

Lab 15: Categorizing Uploaded Data Using AWS Step Functions

Service Introduction

[Amazon Identity Access Management \(IAM\)](#)

[Amazon Step Function](#)

[Amazon Simple Storage Service \(S3\)](#)

[Amazon Lambda Function](#)

[Amazon Transcribe](#)

Case Study Enterprise Internet Television Network – Netflix

[Background](#)

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

Lab 16: Coordinating AI Services with Step Functions

Service Introduction

[AWS Lambda](#)

[AWS Simple Storage S3](#)

[AWS Translate](#)

[AWS Transcribe](#)

[AWS Comprehend](#)

[AWS Polly](#)

[Amazon Step Function](#)

Case Study Enterprise Internet Television Network – Netflix

[Business Challenge](#)

[Proposed Solution](#)

[Lab Diagram](#)

[Implementation Steps](#)

[Solution](#)

[About Our Products](#)

AWS Cloud Certifications

AWS Certifications are industry-recognized credentials that validate your technical cloud skills and expertise while assisting in your career growth. These are one of the most valuable IT certifications right now since AWS has established an overwhelming lead in the public cloud market. Even with the presence of several tough competitors such as Microsoft Azure, Google Cloud Engine, and Rackspace, AWS is by far the dominant public cloud platform today, with an astounding collection of proprietary services that continues to grow.

The two key reasons as to why AWS certifications are prevailing in the current cloud-oriented job market:

- There's a dire need for skilled cloud engineers, developers, and architects – and the current shortage of experts is expected to continue into the foreseeable future.
- AWS certifications stand out for their thoroughness, rigor, consistency, and appropriateness for critical cloud engineering positions.

Value of AWS Certifications

AWS places equal emphasis on sound conceptual knowledge of its entire platform, as well as on hands-on experience with the AWS infrastructure and its many unique and complex components and services.

For Individuals

- Demonstrates your expertise to design, deploy, and operate highly available, cost-effective, and secure applications on AWS.
- Gain recognition and visibility for your proven skills and proficiency with AWS.
- Earn tangible benefits such as access to the AWS Certified LinkedIn Community, invite to AWS Certification Appreciation Receptions and Lounges, AWS Certification Practice Exam Voucher, Digital Badge for certification validation, AWS Certified Logo usage, access to AWS Certified Store.
- Foster credibility with your employer and peers.

For Employers

- Identify skilled professionals to lead IT initiatives with AWS technologies.
- Reduce risks and costs to implement your workloads and projects on the AWS platform.
- Increase customer satisfaction.

Types of Certification

Role-Based Certifications:

- **Foundational** - Validates overall understanding of the AWS Cloud. Prerequisite to achieving Specialty certification or an optional start towards Associate certification.
- **Associate** - Technical role-based certifications. No prerequisite.
- **Professional** - Highest level technical role-based certification. Relevant Associate certification required.

Specialty Certifications:

- Validate advanced skills in specific technical areas.
- Require one active role-based certification.

About AWS – Certified Machine Learning – Specialty Exam

Exam Questions	Case study, short answer, repeated answer, MCQs
Number of Questions	100-120
Time to Complete	180 minutes
Exam Fee	300 USD

Overview of AWS Machine Learning – Specialty Certification

Individuals who work in artificial intelligence/machine learning (AI/ML) development or data science should take the AWS Certified Machine Learning – Specialty (MLS-C01) exam. The exam verifies a candidate's competence to use the AWS Cloud to design, construct, deploy, optimize, train, tune, and manage machine learning solutions for specific business challenges.

A candidate's ability to accomplish the following tasks is also validated by the exam:

- Select and justify the appropriate ML approach for a given business problem
- Identify appropriate AWS services to implement ML solutions
- Design and implement scalable, cost-optimized, reliable, and secure ML solutions

The target candidate is expected to have 2 or more years of hands-on experience developing, architecting, and running ML or deep learning workloads in the AWS Cloud.

Recommended AWS knowledge

The target candidate should have the following knowledge:

- The ability to express the intuition behind basic ML algorithms
- Experience performing basic hyperparameter optimization
- Experience with ML and deep learning frameworks
- The ability to follow model-training best practices
- The ability to follow deployment best practices
- The ability to follow operational best practices

What is considered out of scope for the target candidate?

The following is a non-exhaustive list of related job tasks that the target candidate is not expected to be able to perform. These items are considered out of scope for the exam:

- Extensive or complex algorithm development
- Extensive hyperparameter optimization
- Complex mathematical proofs and computations
- Advanced networking and network design
- Advanced database, security, and DevOps concepts
- DevOps-related tasks for Amazon EMR

AWS Knowledge

- Minimum one year of hands-on experience with the AWS platform
- Professional experience managing/operating production systems on AWS

- A firm grasp of the seven AWS tenets – architecting for the cloud
- Hands-on experience with the AWS CLI and SDKs/API tools
- Understanding of network technologies as they relate to AWS
- Good grasp of fundamental Security concepts with hands-on inexperience in implementing Security controls and compliance requirements

General IT Knowledge

- 1-2 years' experience as a system's administrator in a systems operations role
- Experience in understanding virtualization technology
- Monitoring and auditing system's experience
- Knowledge of networking concepts (DNS, TCP/IP, and Firewalls)
- Ability to collaborate with developers

Intended Audience

Eligible candidates for this exam must have:

- One or more years of hands-on experience in operating AWS-based applications
- Experience in provisioning, operating, and maintaining systems running on AWS
- Ability to identify and gather requirements to define a solution to be built and operated on AWS
- Capabilities to provide AWS operations and deployment guidance and best practices throughout the life cycle of a project

Recommended AWS Knowledge

- Create data repositories for machine learning.
- Identify and implement a data ingestion solution.
- Identify and implement a data transformation solution.
- Sanitize and prepare data for modeling.
- Perform feature engineering.
- Analyze and visualize data for machine learning.
- Frame business problems as machine learning problems.
- Select the appropriate model(s) for a given machine learning problem.
- Train machine learning models.
- Perform hyperparameter optimization.
- Evaluate machine learning models.
- Build machine learning solutions for performance, availability, scalability, resiliency, and fault tolerance.
- Recommend and implement the appropriate machine learning services and features for a given problem.
- Apply basic AWS security practices to machine learning solutions.
- Deploy and operationalize machine learning solutions.

The table below lists the main content domains and their weightings on the exam.

	Domain	Percentage
Domain 1	Data Engineering	20%
Domain 2	Exploratory Data Analysis	24%
Domain 3	Modeling	36%

Domain 4	Machine Learning Implementation and Operations	20%
Total		100%

Course Introduction (MLS-C01)

Machine Learning (ML) is a form of Artificial Intelligence (AI) that allows software programs to improve their prediction accuracy without being expressly designed to do so. Machine learning algorithms use historical data to anticipate new output values.

Machine learning is critical because it helps companies see trends in consumer behavior and business operations while developing new products. Many of today's most successful firms, such as Facebook, Google, and Uber, use machine learning. ML has become a major competitive differentiator for many organizations.

In this course, you will get extensive knowledge of machine learning on AWS. Suppose you are an individual who works in development or data science. In that case, you should pursue the AWS Certified Machine Learning - Specialty Certification as it verifies the candidate's ability to create, deploy, and manage machine learning solutions for specific business issues.

Certification Approved Skills

1. Choose and justify the best machine learning technique for a particular business challenge.
2. Implement ML solutions and identify relevant AWS services.
3. Create scalable, cost-effective, dependable, and secure machine learning systems.

In this course, you will learn about data engineering, data analysis, model design, algorithm selection, optimization, and implementation of machine learning on AWS. This course covers SageMaker, Kinesis Glue, and all the other services that come together to create one of the most powerful ML platforms in the industry.

Recommended AWS Knowledge

The ideal applicant should have one to two years of hands-on experience building, architecting, or operating machine learning/deep learning workloads on the AWS Cloud, as well as the following skills:

- The ability to communicate the intuition underlying fundamental machine learning techniques
- Working knowledge of fundamental hyperparameter optimization
- Working knowledge of machine learning and deep learning frameworks is advantageous
- The capacity to follow best practices in model training
- The ability to adhere to optimal deployment and operating procedures

The AWS Certified Machine Learning Specialty test includes:

- Machine learning algorithms built into Amazon SageMaker (XGBoost, BlazingText, Object Detection, etc.)
- Imputation, outliers, binning, and normalizing are examples of feature engineering techniques
- Comprehend, Translate, Polly, Transcribe, Lex, Rekognition, and other high-level ML services
- S3, Glue, Kinesis, and DynamoDB are all useful for data engineering
- Data exploration using scikit learn, Athena, Apache Spark, and EMR
- Deep neural network hyperparameter tweaking and deep learning
- SageMaker automates model tuning and operations
- Regularization at the L1 and L2 levels
- Machine learning pipelines using security best practices

Amazon Machine Learning

AWS now offers Amazon SageMaker, a sophisticated cloud-based solution that allows developers of all skill levels to leverage machine learning technologies. SageMaker is a fully managed machine learning service that assists you in developing sophisticated machine learning models. Data scientists and developers may use SageMaker to create and train machine learning models, then deploy them into a production-ready hosted environment.

Machine Learning Concepts

Machine learning can assist you in making better business decisions by utilizing previous data. ML algorithms look for patterns in data and use that information to build mathematical models. The models can then be used to create predictions based on future data. For instance, a machine learning model may be used to forecast how likely a buyer is to purchase a specific product based on their previous behavior.

Solving Business Problems with Amazon Machine Learning

Supervised machine learning is a method of learning from data that has been labeled with the correct answer. You can use Amazon Machine Learning to apply machine learning to issues when you already have examples of real-world solutions. To use Amazon ML to predict if an email is spam, for example, you will need to gather email examples that have been appropriately tagged as spam or not. Then, using machine learning, you can generalize from these email examples to forecast whether a new email is likely to be spam or not.

For these specific machine learning problems, you can utilize supervised ML approaches binary classification (predicting one of two possible outcomes), multi-class classification (predicting one of more than two possible outcomes), and regression (predicting a numeric value).

Lab 01: Use Amazon Rekognition

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	Rekognition, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

Amazon Rekognition is a way of converting images into tags or texts. It is a way of allowing your software to see the image. You can upload an image, and Rekognition will tell you what it thinks the image is with a certain degree of confidence. It can be used with lots of apps.

You can recognize objects, scenes, people, activities, and language in photographs and videos and detect inappropriate content using Amazon Rekognition. You may utilize Amazon Rekognition's highly accurate facial analysis and facial search capabilities to recognize, analyze, and compare faces for a range of user verification, people counting, and public safety use cases.

Case Study Enterprise Healthcare - Cleveland Medical Center

Background

Cleveland Medical Centers are stretched across the three states of California, Oregon, and Hawaii. The centers provide service to millions of lives through hospital admissions, emergency, clinic, outpatient, and home care visits. Cleveland uses advanced technology that helps restore patient health and empowers physicians to deliver better care to achieve better outcomes.

Cleveland is currently in the middle of a digital transformation to bring patient care closer to home. They aim to treat patients at the nearest hospital or resource center in their area or dispatch healthcare workers to assist them in their homes. To do this, they plan to use video conferencing, IoT, remote monitoring, and mobile solutions that will enable people to get the healthcare they need, thus limiting the number of patients who have to travel to one of their hospitals. This way is not only cost-effective, but it also gives patients a better experience.

An external consultant has done an extensive study on the Cleveland infrastructure. The consultant has

advised the board to build a “Cloud First” strategy. This will reduce the need for organizations to invest in and maintain their on-premises IT resources, scalability to meet workload and user demands, and fewer wasted resources because the company will only pay for what they use.

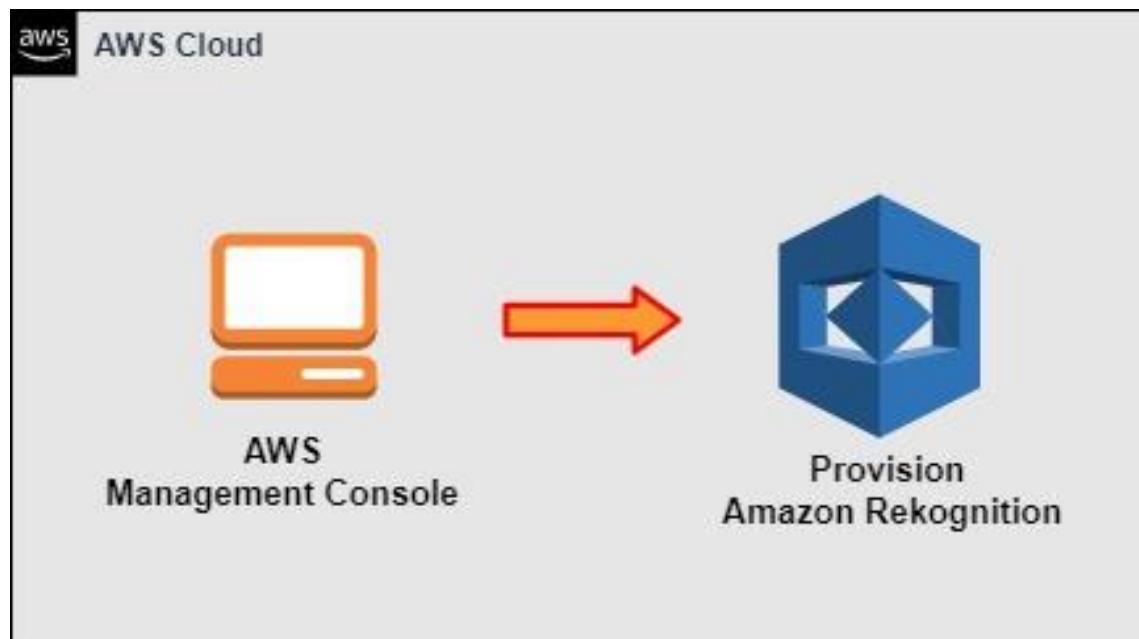
Business Challenge

Cleveland's business development team has deployed an online appointment portal where users conveniently schedule their appointments. Hundreds of patients come to the medical centers daily through online appointments, and management has faced trouble identifying the patients by appointment number only. Cleveland's business development team wants a solution that identifies the patient by face and shows the basic details. The management team hired a Cloud Specialist to propose a solution for the current requirement.

Proposed Solution

You are the Cloud Specialist for Cleveland medical center. You proposed a solution to this requirement by using Amazon Rekognition. Cleveland's management team can easily identify the patient by using the facial analysis feature of Amazon Rekognition. To use this feature, the picture of the patients should be uploaded at the time of the appointment. When patients come for medical consultancy, management identifies the patient through the Amazon Rekognition service.

Lab Diagram



Implementation Steps

1. Navigate to the Amazon Rekognition.
2. Use Facial Analysis.

Solution

Step 1: Navigate to the Amazon Rekognition

1. Log in to AWS Management Console.



Services ▾



More ▾

AWS Management Console

AWS services

▼ Recently visited services



IAM



EC2



Billing

► All services

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Click on Services.

3. Click on Amazon Rekognition under Machine Learning.

Compute	Machine Learning
EC2	Amazon SageMaker
Lightsail	Amazon Augmented AI
Lambda	Amazon CodeGuru
Batch	Amazon DevOps Guru
Elastic Beanstalk	Amazon Comprehend
Serverless Application Repository	Amazon Forecast
AWS Outposts	Amazon Fraud Detector
EC2 Image Builder	Amazon Kendra
AWS App Runner	Amazon Lex
Containers	Amazon Personalize
Elastic Container Registry	Amazon Polly
Elastic Container Service	Amazon Rekognition
Elastic Kubernetes Service	Amazon Textract
	Amazon Transcribe

4. On the Amazon Rekognition page, click on the Facial analysis

feature.

Amazon Rekognition



▼ Custom Labels

Use Custom Labels

▼ Demos

Label detection

Image moderation

Facial analysis



Celebrity recognition

Face comparison

Text in image

Step 2: Use Facial Analysis

1. When the **Facial analysis** page appears, scroll down and click on **Upload** to upload the patient's picture.

Facial analysis

Get a complete analysis of facial attributes, including confidence scores.

[Learn more](#)

[Leave us feedback](#)

Choose a sample image



Use your own image

Image must be .jpeg or .png format and no larger than 5MB. Your image isn't stored.

Upload

or drag and drop

[Use image URL](#)

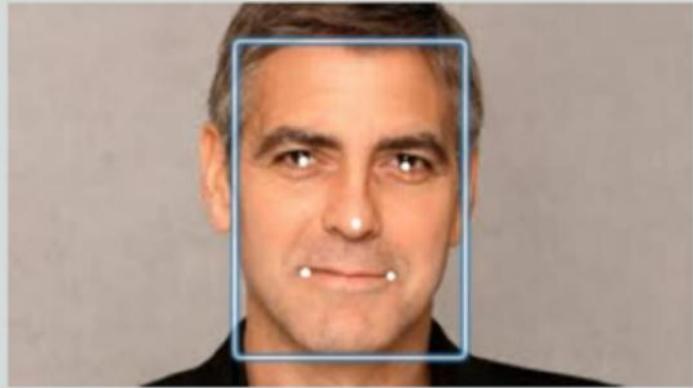
Go

Facial analysis

Get a complete analysis of facial attributes, including confidence scores.

[Learn more](#) 

[Leave us feedback](#)



2. The facial analysis will show the **Result** after analyzing the **picture**.
3. The **Result** will be shown below.

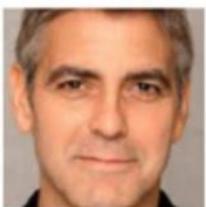
Facial analysis

Get a complete analysis of facial attributes, including confidence scores.

[Learn more](#)

[Leave us feedback](#)

▼ Results



looks like a face 99.9 %

appears to be male 98.5 %

age range 34 - 50 years old

not smiling 91.6 %

appears to be calm 90.6 %

not wearing glasses 98.9 %

not wearing sunglasses 99.5 %

eyes are open 99.1 %

mouth is closed 98.5 %

does not have a mustache 98.7 %

does not have a beard 84.5 %

Lab 02: Creating a Lambda Function

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time	

Required	1 Hr.
Tags	Lambda, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

AWS Lambda is a serverless compute service that lets you run code without needing to deploy or manage servers, design workload-aware cluster scaling logic, maintain event integrations, or manage runtimes. You can run code for nearly any form of application or backend service with Lambda, and you do not have to worry about administration. Therefore, you do not need to worry about which AWS resources to launch or how well you manage them. Instead, you must upload the code to Lambda, and it will execute.

The code in AWS Lambda is performed in reaction to events in AWS services, such as adding/deleting files in an S3 bucket, making an HTTP call to the Amazon API gateway, and so on. On the other hand, Amazon Lambda can only be utilized to do background activities.

Case Study Enterprise Healthcare - Cleveland Medical Center

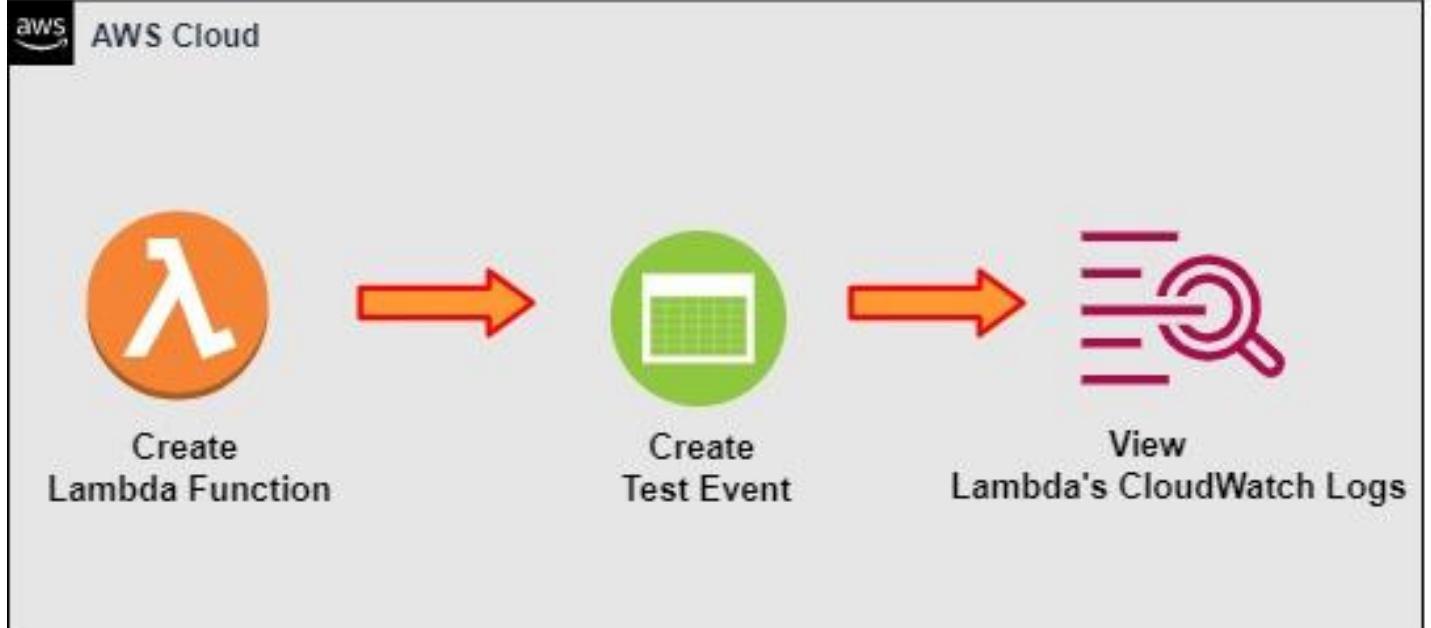
Business Challenge

Cleveland's business development team plans to shift its resources from on-premises to the cloud. One of the employees suggested using AWS cloud service providers to pursue this requirement. After migration, Cleveland wants automation for managing the underlying compute service. The management team hired a Cloud Specialist to propose a solution for the current requirement.

Proposed Solution

You are the Cloud Specialist for Cleveland medical center. You proposed a solution to this requirement by using AWS Lambda, and Cleveland can automatically manage the underlying compute resources for you with high scalability, performance, and security.

Lab Diagram



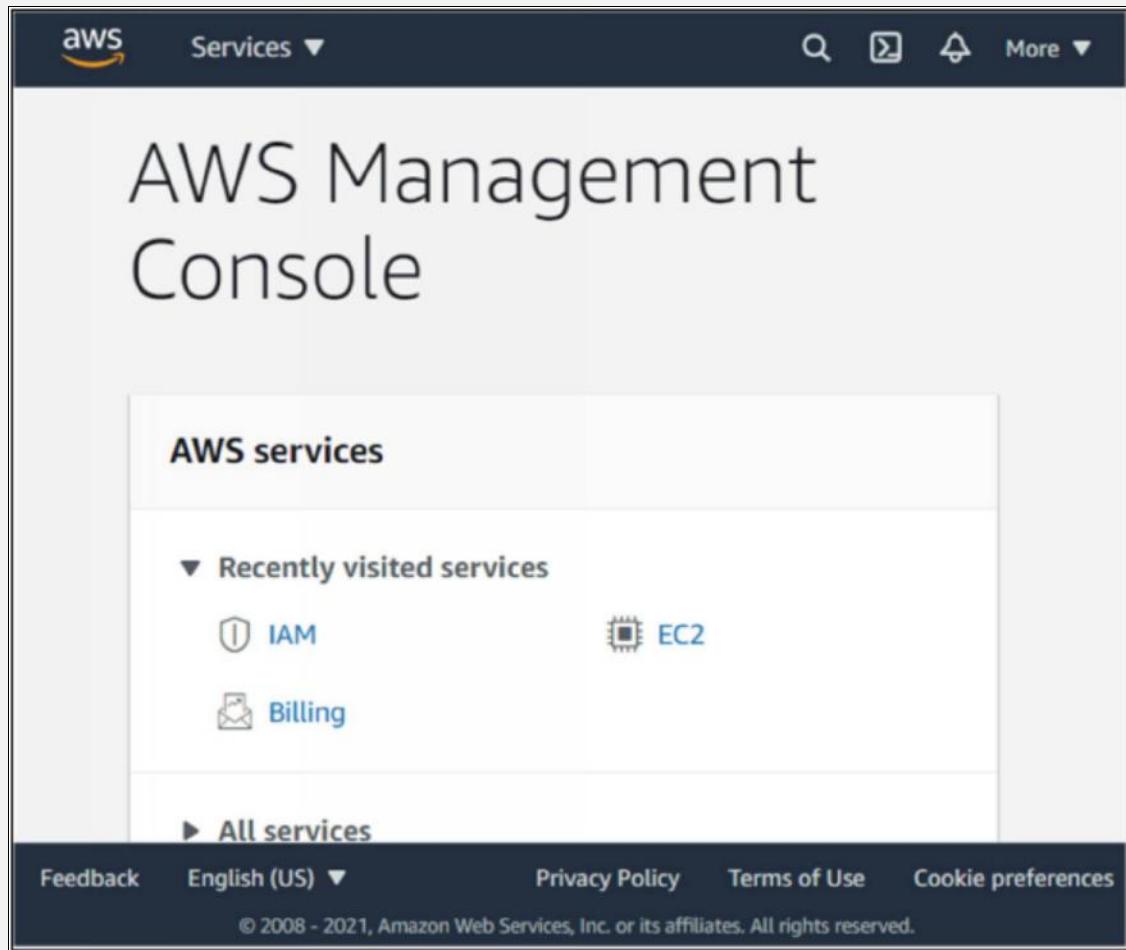
Implementation Steps

1. Author Node.js Lambda Function in AWS Console.
2. Test Function Using a Test Event.
3. View Lambda's CloudWatch Logs.

Solution

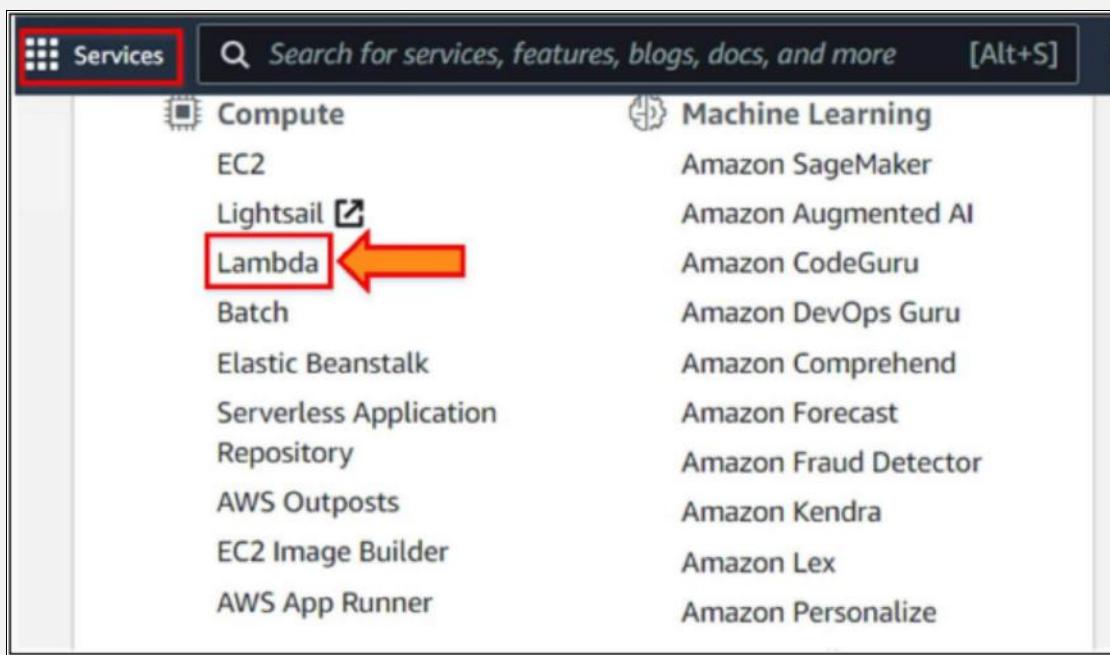
Step 1: Author Node.js Lambda Function in AWS Console

1. Log in to AWS Management Console.

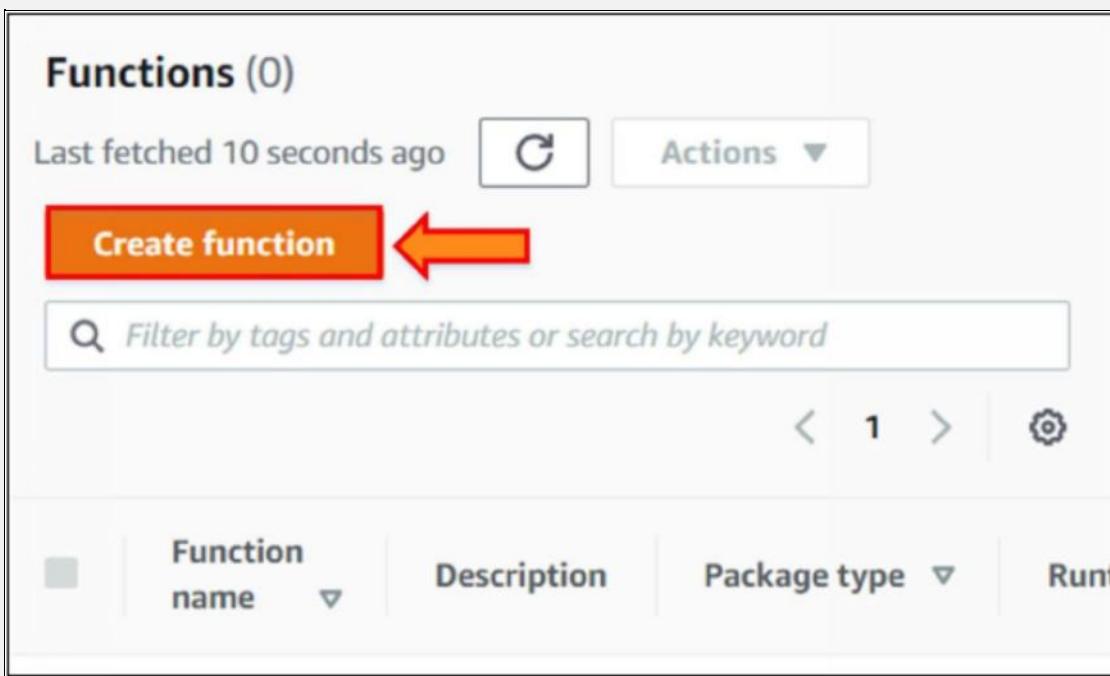


2. Click on Services.

3. Click on Lambda under Compute.



4. Click on Create function.



5. Select Author from scratch.

Create function Info

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.

Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

6. Set the following **values**:

Function name: **AWSURLChecker**

Runtime: **Node.js**

7. Expand **Change default execution role**.
8. For the **Execution role**, select **Create a new role with basic Lambda permissions**.
9. Click **Create function**.

Basic information

Function name

Enter a name that describes the purpose of your function.

AWSURLChecker

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Node.js 14.x

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Create a new role with basic Lambda permissions

Use an existing role

Create a new role from AWS policy templates

Ca

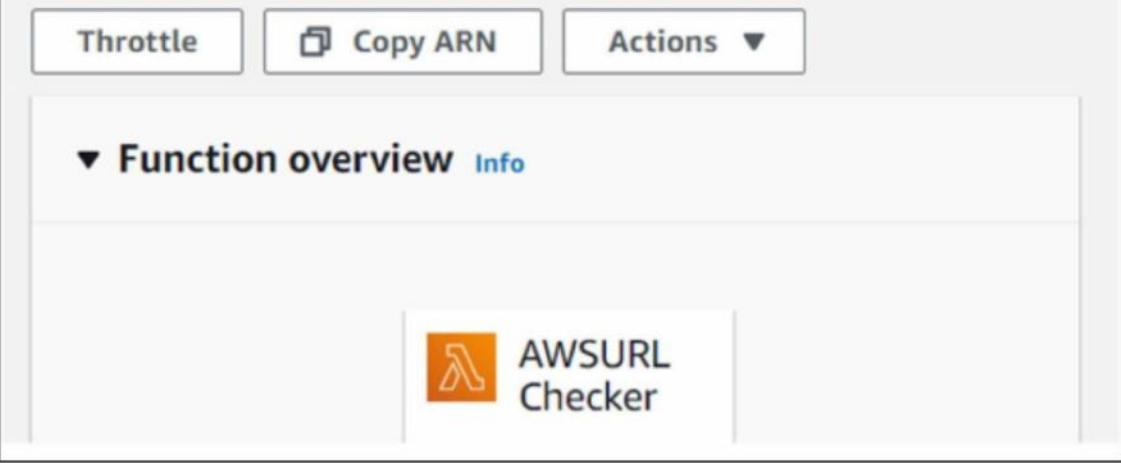
Create function

10.

The function is now successfully created.

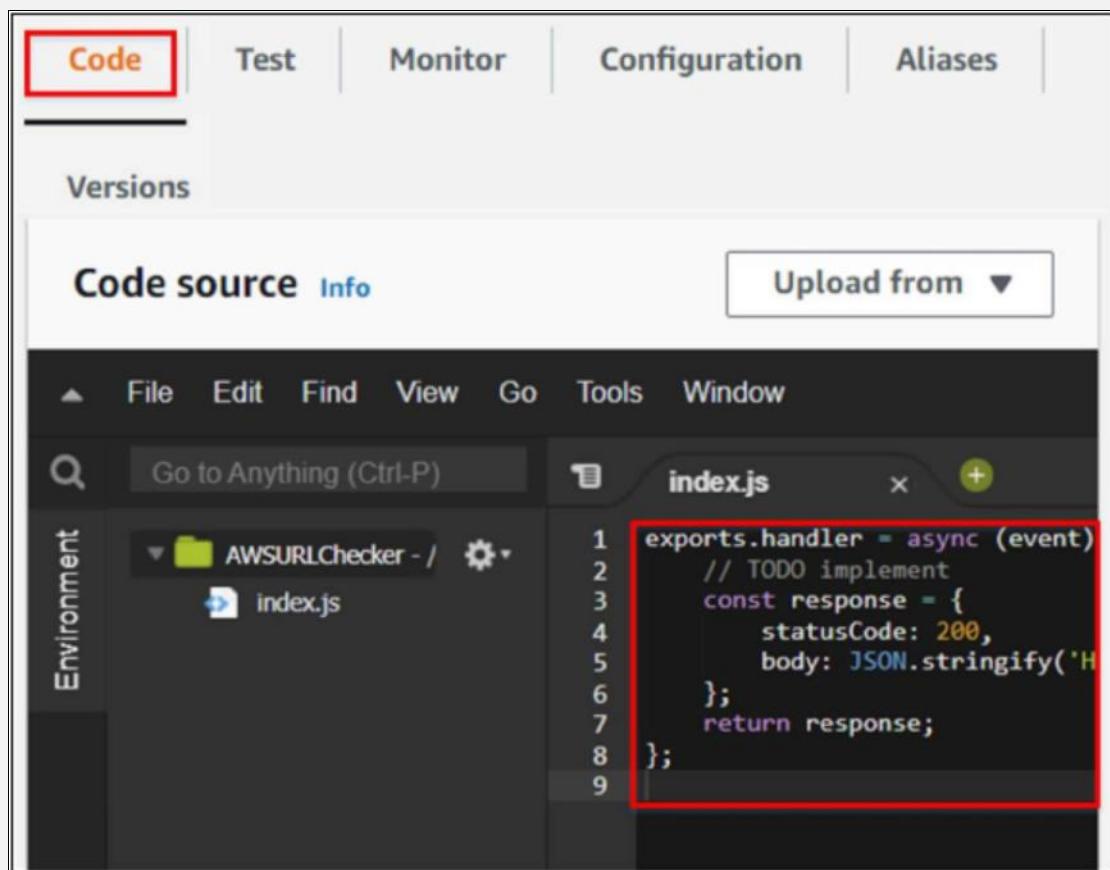
- ✓ Successfully created the function **AWSURLChecker**. You can now change its code and configuration. To invoke your function with a test event, choose "Test". X

AWSURLChecker



11.

Now, click on **Code**. Click on **index.js** and **delete** the code.



12.

Then, go to the **GitHub link** provided and copy the code.

<https://github.com/linuxacademy/content-aws-ccp-labs/blob/master/creating-a-lambda-function-using-the-aws-console/index.js>

master

...

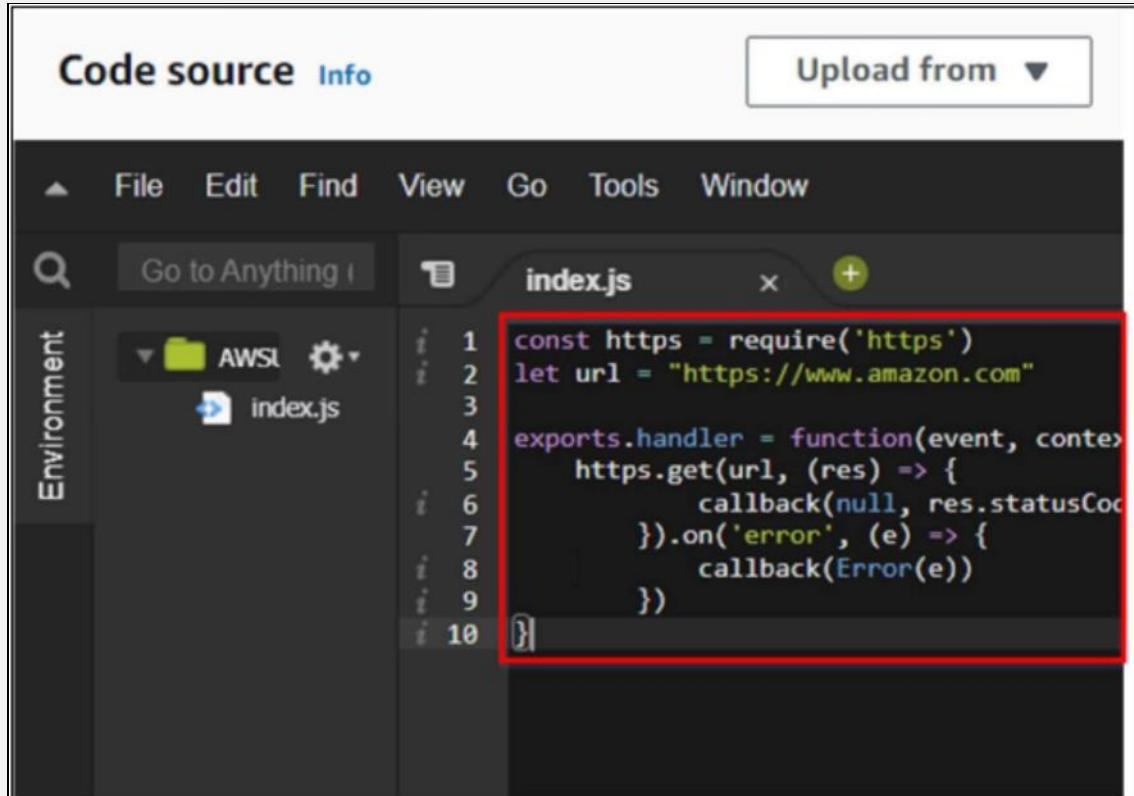
[content-aws-ccp-labs / creating-a-lambda-function-using-the-aws-console / index.js](#) / < Jump to keshawilliams lambda function lab [History](#)

1 contributor

10 lines (9 sloc) | 259 Bytes [...](#)

```
1 const https = require('https')
2 let url = "https://www.amazon.com"
3
4 exports.handler = function(event, context, callback) {
5     https.get(url, (res) => {
6         callback(null, res.statusCode)
7     }).on('error', (e) => {
8         callback(Error(e))
9     })
10 }
```

13.

Paste it into **index.js** and deploy.

The screenshot shows the AWS Lambda code source editor. The title bar says "Code source" and "Info". There is a "Upload from" button. The menu bar includes File, Edit, Find, View, Go, Tools, and Window. A search bar says "Go to Anything". The sidebar shows an "Environment" section with an "AWS" folder containing an "index.js" file. The main editor area shows the following code:

```
const https = require('https')
let url = "https://www.amazon.com"

exports.handler = function(event, context, callback) {
    https.get(url, (res) => {
        callback(null, res.statusCode)
    }).on('error', (e) => {
        callback(Error(e))
    })
}
```

The code is highlighted with a red box around the entire function definition.

Step 2: Test Function Using a Test Event

1. Click **Test** in the upper right.
2. For the **Event template**, enter **AWSTestEvent**.
3. Leave the other **default** values.
4. Click **Save changes**.

The screenshot shows the AWS Lambda function configuration interface. The top navigation bar has tabs for Code, Test (which is highlighted with a red border), Monitor, Configuration, and Aliases. Below the tabs is a section titled 'Versions'. Under 'Test event', there are two options: 'New event' (selected) and 'Saved event'. A 'Template' dropdown is set to 'hello-world'. In the 'Name' field, 'AWSTestEvent' is entered, with a red arrow pointing to it. Below the name is a code editor window displaying a JSON event template:

```
1  []
2  {
3      "key1": "value1",
4      "key2": "value2",
5      "key3": "value3"
6  }
```

5. When the **Test event** is saved, click on **Test**.

Test event

Delete

Format

Save cha

Test

Invoke your function with a test event. Choose a template that matches the service that triggers your function, or enter your event document in JSON.

- New event
 - Saved event
- Saved event

AWSTestEvent



6. Review the **output response**.

7. It should be **200**, which means the website is **up and running**.



Execution result: succeeded ([logs](#))



▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{  
  "statusCode": 200,  
  "body": "\"Hello from Lambda!\""  
}
```

Summary

Code SHA-256

uTJFxT0sQYd8f6CtxoZoBcLT6Hd0A48LniMm4gpxgDw=

Request ID

d4a895d2-6933-4a76-88ac-3a3145dbd267

Step 3: View Lambda's CloudWatch Logs

1. Click the **Monitoring** tab.
2. Click **View logs in CloudWatch**.

[Code](#)[Test](#)[Monitor](#)[Configuration](#)[Aliases](#)

Versions

[Metrics](#)[Logs](#)[Traces](#)[View logs in CloudWatch](#) ↗[View X-Ray traces in ServiceLens](#) ↗[View Lambda Insights](#) ↗[View profiles in CodeGuru](#) ↗

3. Click on the most recent log stream in the **Log streams** section.

/aws/lambda/AWSURLChecker

Actions ▾

[View in Logs Insights](#)

[Search log group](#)

► Log group details

[Log streams](#)

Metric filters

Subscription filters

Contributor Insights

Tags

Log streams (1)



Delete

[Create log stream](#)

[Search all](#)



Filter log streams or try prefix search



1



Log stre... ▾

Last event time ▾



2021/12/0...

2021-12-02 19:11:07 (UTC+...

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your logs.

View as text



Actions ▾

Create Metric Filter

Filter events

Clear

▶ **Timestamp**

Message

No older events at this moment.

▶ 2021-12-02T19:11:07.469+05:00

START RequestId: d4a895d2-6933-4a76-88ac-3a3145dbd267 Version: \$LATEST

▶ 2021-12-02T19:11:07.474+05:00

END RequestId: d4a895d2-6933-4a76-88ac-3a3145dbd267 Duration: 2.13 ms

▶ 2021-12-02T19:11:07.474+05:00

REPORT RequestId: d4a895d2-6933-4a76-88ac-3a3145dbd267 Duration: 2.13 ms

4. Notice the **Billed duration**.

Message
No older events at this moment. Retry
START RequestId: d4a895d2-6933-4a76-88ac-3a3145dbd267 Version: \$LATEST
END RequestId: d4a895d2-6933-4a76-88ac-3a3145dbd267 Duration: 2.13 ms
REPORT RequestId: d4a895d2-6933-4a76-88ac-3a3145dbd267 Duration: 2.13 ms 88ac-3a3145dbd267 Duration: 2.13 ms Billed Duration: 3 ms Memory B Init Duration: 159.50 ms

Lab 03: Build A Serverless Webpage with API Gateway and AWS Lambda

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	Lambda, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

AWS Lambda is the ultimate abstraction layer. Lambda is a Function as a Service (FaaS) product, which means you have a function provided with the code, and AWS executes that code. In this, unlike traditional compute, you are only billed for the number of milliseconds in which your code runs. Lambda runs the code in an isolated environment, and it is stateless, but you are responsible for persistence as the environment where Lambda runs has default access to no information except incoming payload. Lambda does not maintain any result of data after the termination of a function, so it is your responsibility to put the result in a secure place.

In serverless architecture, the user sends a request to API Gateway, which then responds to AWS Lambda. AWS Lambda can communicate with databases like Dynamo DB, Amazon Aurora, or serverless Aurora. It will then send a response back to the user.

Case Study Enterprise Educational Services – Abhay Engineering College (AEC)

Background

AEC in New Zealand has more than 30,000 students and 10,000 university employees who work over seven campuses in Auckland and one campus in the Northland region of New Zealand. To keep its position as New Zealand's top university, AEC embraced digital transformation across its operations to effectively meet the current and future needs of its students and staff.

The students at the college are actively researching in the areas of machine learning, image recognition, and artificial intelligence. AEC is looking to opt for a robust strategy for the composition of these services. The demand varies based on the number of research fellows involved in a particular area at one time.

AEC is currently managing two datacenters in Auckland. The datacenters are hosting around 30 applications for students and staff. The college started its e-learning program a couple of years back, which enables remote students from all over the world to get an AEC degree online. It has put a significant load on the already stressed legacy infrastructure.

The management has restricted any further expansion in the on-premises datacenters. The CTO is working on a cloud strategy to migrate the applications to Cloud in a phased approach.

You are working as a consultant for AEC. You will support AEC in solving a number of business and technical challenges as part of this case study.

Business Challenge

AEC is planning to launch a new student portal on AWS Cloud where a student submits their assignments, taking lectures and notes from there. The portal should be available 24/7 for all students. The management wants a solution that does not require anything to manage. They have approached you for this task.

Proposed Solution

You are the Cloud Specialist for AEC. You proposed a solution to their requirement by using AWS

Lambda. AEC can automatically manage the underlying compute resources for you with high scalability, performance, and security. This solution will configure a Serverless Webpage with API Gateway and AWS Lambda.

Lab Diagram



Implementation Steps

1. Create a Lambda Function
2. Edit Code Source
3. Add Trigger
4. Deploy API

Solution

Step 1: Create a Lambda Function

1. Log in to the **AWS Management Console**.



Services ▾



More ▾

AWS Management Console

AWS services

▼ Recently visited services



IAM



EC2



Billing

► All services

[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Go to Services.



Services ▾



More ▾

AWS Management Console

AWS services

▼ Recently visited services



IAM



EC2



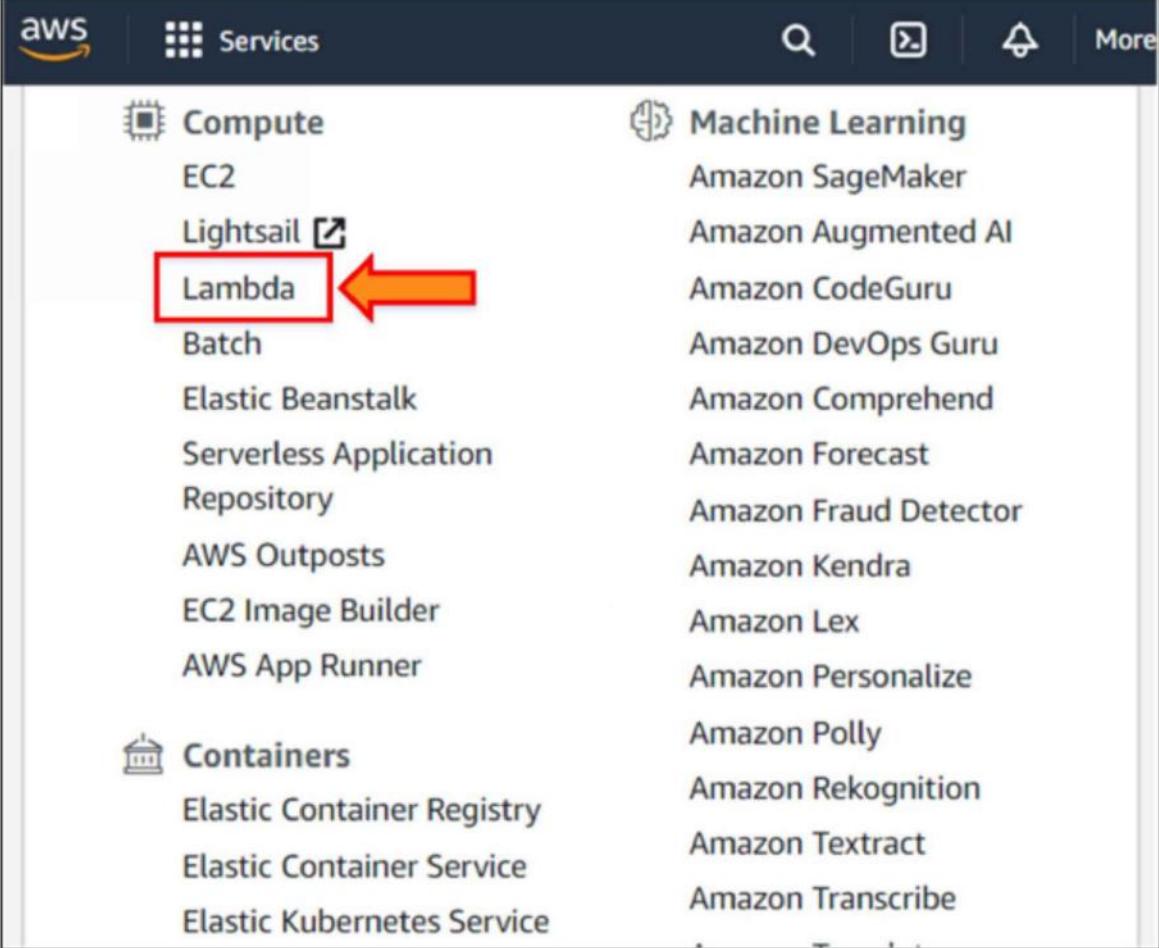
Billing

► All services

[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Click on Lambda, which is present under Compute.



4. The screen of **AWS Lambda** will appear; click on **Create function**.

The screenshot shows the AWS Lambda Functions page. At the top, it displays 'Functions (0)' and 'Last fetched 1 minute ago'. Below this are buttons for 'Create function' (highlighted with a red box and orange arrow) and 'Actions'. A search bar with the placeholder 'Filter by tags and attributes or search by keyword' is also present. At the bottom, there are filters for 'Function name', 'Description', 'Package type', and 'Runtime', and a message stating 'There is no data to display.'

5. You can make **three** different **Lambda functions**. You can use the **author from scratch**, you can **use blueprint**, or you can use the **serverless application repository**.
6. In this solution, you will need to **Author from scratch** for creating the

Lambda function.

Create function Info

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.



Use a blueprint



Build a Lambda application from sample code and configuration presets for common use cases.

7. Write the **function name** as **AWSLambdaFunction**.

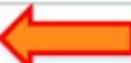
8. Select **Python 3.6** from the given list in **Runtime**.

Basic information

Function name

Enter a name that describes the purpose of your function.

AWSLambdaFunction



Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.6



9. Select the execution role as **Create a new role from AWS policy templates**.

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates



10. Give the name **LambdaRole** to the role.
11. Choose the policy template **Simple microservice permissions** from the given list.
12. After that, click on **Create function**.

Role name

Enter a name for your new role.

LambdaRole



Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates - optional Info

Choose one or more policy templates.

Simple microservice permissions X

DynamoDB



Create function



13. The **notification of a successful creation of Lambda function** will appear.

✓ Successfully created the function **AWSLambdaFunction**. You can now change its code and configuration. To invoke your function with a test event, choose "Test". X

AWSLambdaFunction

Throttle

Copy ARN

Actions ▾

▼ Function overview Info



AWSLambdaFunction

Step 2: Edit Code Source

1. Go to the **Code source** section, and the default **coding** for the **Lambda function** is shown.

Code source Info

Upload from ▾

File Edit Find View Go Tools Window



Go to Anything (Ctrl-P)



lambda_function



▼ AWSLambdaFunction Edit Find View Go Tools Window

lambda_function.py

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('AEC')
8     }
9 |
```

2. Go to your **resources section** and type the following **script**:

```
def lambda_handler(event, context)
    print("In lambda handler")
```

```
resp = {
    "statusCode": 200,
    "headers" : {
```

```
        "Access-Control-Allow-Origin": "*",
    },
    "body": "IPS Study Content"
}

return resp
```

Code source [Info](#)

Upload from ▾

File Edit Find View Go Tools Window

Go to Anything (Ctrl-P)

Environment

AWSLambdaFunction λ lambda_function.py

```
lambda_function x +  
1 def lambda_handler(event, context)  
2     print("In lambda handler")  
3  
4     resp = {  
5         "statusCode": 200,  
6         "headers" : {  
7             "Content-Type": "application/json"  
8         },  
9         "body": "AEC Study Content"  
10    }  
11  
12    return resp  
13
```

3. Go to **File**, and click on **Save As**.
4. Click on **Save**.

Code source [Info](#)

Upload from ▾

File Edit Find View Go Tools Window

New File Alt-N

New From Template

Open Recent

Save Ctrl-S

Save As... Ctrl-Shift-S

Save All

Line Endings

Close File Alt-W

Close All Files Alt-Shift-W

```
lambda_function x +  
1 def lambda_handler(event, context)  
2     print("In lambda handler")  
3  
4     resp = {  
5         "statusCode": 200,  
6         "headers" : {  
7             "Content-Type": "application/json"  
8         },  
9         "body": "AEC Study Content"  
10    }  
11  
12    return resp  
13
```

Step 3: Add Trigger

1. After that, go to **add a trigger**.
2. Click on **+ Add trigger**.

AWSLambdaFunction

Throttle

Copy ARN

Actions ▾

▼ Function overview [Info](#)

 AWSLam
bdaFunc
tion



Layers (0)

+ Add trigger

+ Add destination



3. Select **API Gateway** from the given list.
4. Select **API type** as **REST API**.
5. Select **IAM** as a security mechanism for your API endpoint.
6. Leave the remaining options as default and click on **Add**.

Add trigger

Trigger configuration

API Gateway application-services 

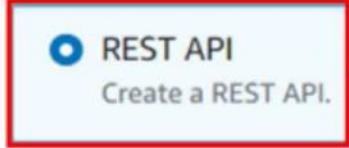
Add an API to your Lambda function to create an HTTP endpoint that invokes your function. API Gateway supports two types of RESTful APIs: HTTP APIs and REST APIs. [Learn more](#)

API
Create a new API or attach an existing one.

Create an API  

API type

HTTP API
Create an HTTP API.

REST API
Create a REST API.  

Security
Configure the security mechanism for your API endpoint.

IAM  

Add 

7. The **API Gateway** as a **trigger** is shown. Click on the given link.

Triggers (1)

[Enable](#)[Disable](#)[Fix errors](#)[Delete](#)[Add trigger](#) Find triggers

< 1 >



Trigger

**API Gateway: AWSLambdaFunction-API**

arn:aws:execute-api:us-east-1:734404852544:yyssx5x3d1/*/*AWSLambdaFunction

API endpoint: <https://yyssx5x3d1.execute-api.us-east-1.amazonaws.com/default/AWSLambdaFunction>

8. You will get an **error message**.

A screenshot of a web browser window. The address bar shows the URL: https://yyssx5x3d1.execute-api.us-east-1.a... . The page content displays a JSON object: {"message": "Missing Authentication Token"}. The browser interface includes standard navigation buttons (back, forward, refresh), a search bar, and a star icon.

9. Now, go back and click on **AWSLambdaFunction-API**.

10. The API Gateway is shown in the form of a **User Interface**.

Note: The API Gateway is shown in the form of a User Interface (UI). The client is present on one side, and the Lambda function is present on another. The client sends a request of type “ANY.” The request proxies to the lambda function, and then the lambda function responds to the client.

11. Go to **Actions** and click on **Delete Method**.

The screenshot shows the AWS Lambda Function API configuration page. At the top, the path is: Amazon API Gateway > APIs > AWSLambdaFunction-API (yyssx5x3d1) > Resources. On the left, there's a tree view with a node expanded to show 'ANY' under '/AWSLambdaFunction'. In the center, there's a 'Actions' dropdown menu with several options: 'Edit Method Documentation', 'Delete Method' (which is highlighted with a red box and has a red arrow pointing to it), 'Create Method', 'Create Resource', 'Enable CORS', 'Edit Resource Documentation', 'Delete Resource', 'Deploy API', 'Import API', 'Edit API Documentation', and 'Delete API'. To the right, there are two sections: 'Method Request' (Auth: AWS IAM, ARN: arn:aws:execute-api:us-east-1:734404852544:yyssx5x3d1/ANY) and 'Method Response' (HTTP Status: Proxy, Models:). A large orange arrow points from the 'Actions' dropdown towards the 'Delete Method' option.

12. A dialog box will appear. Click on **Delete**.



13. Go to Actions and click on **Create Method** to create a new method.

The screenshot shows the AWS Lambda API Gateway interface. In the top navigation bar, it says "Amazon API Gateway", "APIs > AWSLambdaFunction-API (yyssx5x3d1) > Resources". Below this, there's a breadcrumb trail: "Resources / /AWSLambdaFunction Methods". On the left, there's a sidebar with "RE SOURCE ACTIONS" and "API ACTIONS" sections. The "Create Method" option under "RE SOURCE ACTIONS" is highlighted with a red box and an orange arrow pointing to it from the left.

14. Select the function type **GET** from the given list.

The screenshot shows the AWS Lambda API Gateway interface. In the top navigation bar, it says "Amazon API Gateway", "APIs > AWSLambdaFunction-API (yyssx5x3d1) > Resources". Below this, there's a breadcrumb trail: "Resources / /AWSLambdaFunction Methods". Under " /AWSLambdaFunction", there's a dropdown menu showing "GET" selected. A red box highlights the "GET" button, and an orange arrow points to it from the left.

15. Select the **Integration type** as **Lambda Function**.
16. Enable **Use Lambda Proxy Integration**.
17. Select the **N.Virginia us-east-1** region.
18. Give the name of **Lambda Function** as **AWSLambdaFunction**.
19. Enable **Use Default Timeout**.
20. Click on **Save**.

/AWSLambdaFunction - GET - Setup

Choose the integration point for your new method.

Integration type Lambda Function 

HTTP 
 Mock 
 AWS Service 
 VPC Link 

Use Lambda Proxy integration 

Lambda Region 

Lambda Function

Use Default Timeout 



21. Click on **OK**.

Add Permission to Lambda Function

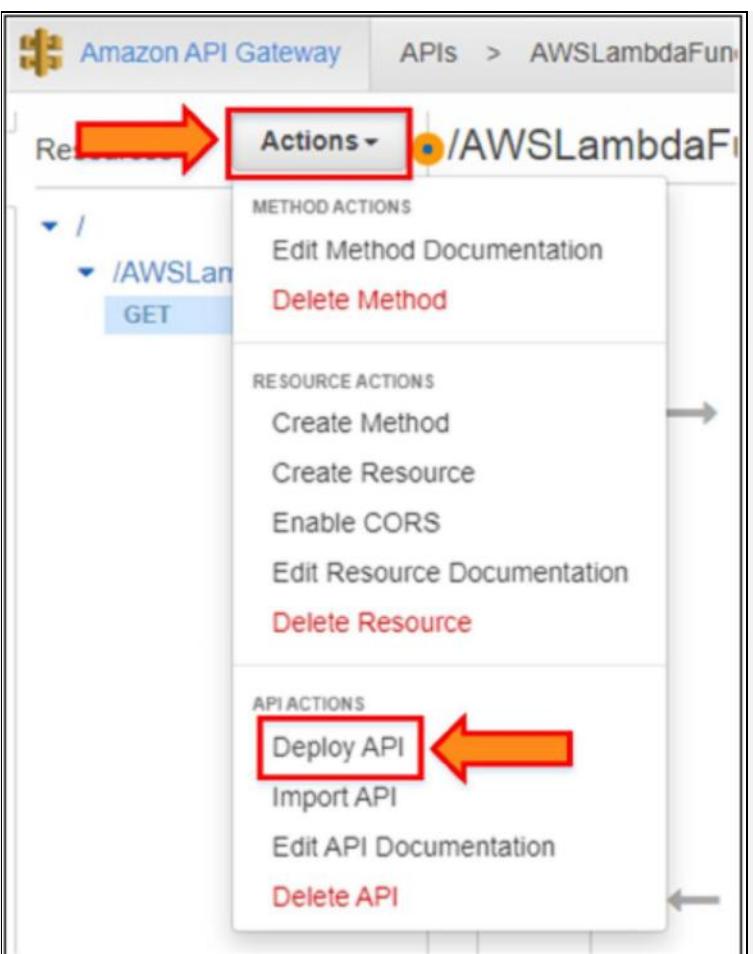
You are about to give API Gateway permission to invoke your Lambda function:

arn:aws:lambda:us-east-1:734404852544:function:AWSLambdaFunction



Step 4: Deploy API

1. Now to **Deploy API**, go to **Actions**.
2. Click on **Deploy API**.



3. Select the **Deployment stage**.
4. Enter **Stage name**.
5. Enter **Stage description**.
6. Write the **Deployment description** as **AEC Website Deployment**.
7. Click on **Deploy**.

The screenshot shows the "Deploy API" dialog box. It has a heading "Deploy API" and a close button. The instructions say: "Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta." Below are four input fields with red arrows pointing to them:

Deployment stage	[New Stage]
Stage name*	First
Stage description	Initial Stage
Deployment description	AEC Website Deployment

At the bottom right is a large blue "Deploy" button with a red arrow pointing to it.

8. After **deployment of API**, click on **GET**.
9. The deployed API is shown in the figure. Click on this to **Invoke the**

URL.

The screenshot shows the AWS API Gateway console. In the top navigation bar, it says "Amazon API Gateway" and "APIs > AWSLambdaFunction-API (yyssx5x3d1) > Stages > F". On the left sidebar, under "Stages", there are two entries: "default" and "First". The "First" entry is expanded, showing a "GET" method mapped to the path "/AWSLambdaFunction". A red box highlights the "GET" button, and a red arrow points from it to the URL field. The URL field contains "https://yyssx5x3d1.execute-api.us-east-1.". Below the URL field, there is a note: "Use this page to override the First stage settings for the GET to /A". Under "Settings", there are two radio buttons: "Inherit from stage" (selected) and "Override for this method".

10. The new tab opens with **AEC Study Content**.

The screenshot shows a browser window with the URL "https://yyssx5x3d1.execute-api.us-east-1.a...". The main content area displays the text "AEC Study Content!".

Lab 04: Streaming Data Collection

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	Lambda, S3, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

Kinesis Data Streams

You may use Amazon Kinesis Data Streams to create custom applications that process or analyze streaming data for specific purposes. To an Amazon Kinesis data stream, you may constantly add data from hundreds of thousands of sources, such as clickstreams, application logs, and social media. Within seconds, your Amazon Kinesis Applications will be able to read and analyze data from the stream.

Kinesis Data Analytics

Amazon Kinesis Data Analytics is the most straightforward method for converting and analyzing real-time streaming data using Apache Flink. Apache Flink is an open-source data stream processing framework and engine. Apache Flink applications' creation, maintenance, and integration with other AWS services are made easier using Amazon Kinesis Data Analytics.

Amazon Kinesis Data Analytics handles everything needed to constantly operate streaming applications and grows automatically to meet the amount and throughput of your incoming data. There are no servers to operate with Amazon Kinesis Data Analytics, no minimum charge or setup cost, and you pay only for the resources your streaming applications utilize.

Kinesis Data Firehose

Kinesis Data Firehose is a solution for streaming ETL (Extract, Transform, and Load). It is the most convenient method for loading streaming data into data storage and analytics tools. It can collect, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, enabling near-real-time analytics with your existing business intelligence tools and dashboards. It is a fully managed service that scales automatically to meet your data flow and does not require any ongoing management. It may also batch, compress, and encrypt data before loading it, reducing storage requirements at the destination and enhancing security.

AWS Lambda

AWS Lambda allows you to run code without creating or managing servers. There is no charge when your code is not executing; you only pay for the compute time you use. You can run code for nearly any application or backend service with Lambda, and you do not have to worry about administration. Upload your code, and Lambda will handle everything necessary to run and grow it with high availability. You may configure your code to be automatically triggered by other AWS services, or you can access it directly from any computer or smartphone app.

Amazon Simple Storage Service S3

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 is a web service that allows you to store and retrieve an infinite quantity of data from any place and at any time. Using this service, you may quickly create projects that integrate cloud-native storage because Amazon S3 is easily customizable. You only pay for what you use; you can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. Build a simple FTP system or a complex web application

like the Amazon.com retail website; read the same piece of data a million times or only for emergency disaster recovery; store whatever type and amount of data you desire.

AWS CloudFormation

AWS CloudFormation is a tool that makes it simple for developers and organizations to construct a collection of linked AWS and third-party resources and then provision and manage them logically and reasonably.

Developers may use a simple, declarative approach to deploy and change compute, database, and many other resources, abstracting away the complexities of individual resource APIs. AWS CloudFormation is meant to make resource lifecycle management repeatable, predictable, and safe, with features such as automatic rollbacks, automated state management, and resource management across accounts and regions. Multiple ways to generate resources have recently been added, including leveraging AWS CDK for writing in higher-level languages, importing existing resources, and detecting configuration drift. A new registry simplifies the creation of custom types that inherit several of CloudFormation's fundamental features.

AWS Identity and Access Management (IAM)

Individuals and groups can be granted secure access to your AWS resources by using IAM. It allows you to create and manage IAM users and provide them access to your resources. Additionally, you have the option of granting access to users outside of AWS (federated users).

- **Managed Policy:** This contains the permission required to stop an EC2 instance
- **Inline Policy:** This allows this role to be passed to another service
- **Trust Policy:** Allows System Manager and EC2 to assume the role. It enables EC2 to register with the Systems Manager and Systems Manager to stop the EC2 instance

Case Study Enterprise Biopharmaceutical – Neumora Therapeutics

Background

Neumora Therapeutics is a clinical-stage biopharmaceutical company leveraging advanced data science and artificial intelligence to develop targeted treatments for neurobehavioral disorders.

Business Challenge

[Neumora Therapeutics](#) has thousands of users interacting with the organization application. The organization gives [Neumora Therapeutics](#) a task to capture real-time user data for a marketing campaign. For users aged 21 and up, it must collect information such as name, age, gender, and location.

Proposed Solution

The solution is simple! You use AWS services to automate your work. You use Kinesis Data Streams to generate or collect data. Then, you use Kinesis Data Analytics to analyze data by running SQL queries. For storing the real-time analyzing data, you use Kinesis Data Firehose to store data at any storage destination. The other helping service you can use is AWS Lambda to transform data into proper JSON format. An S3 bucket is used to store that transformed data.

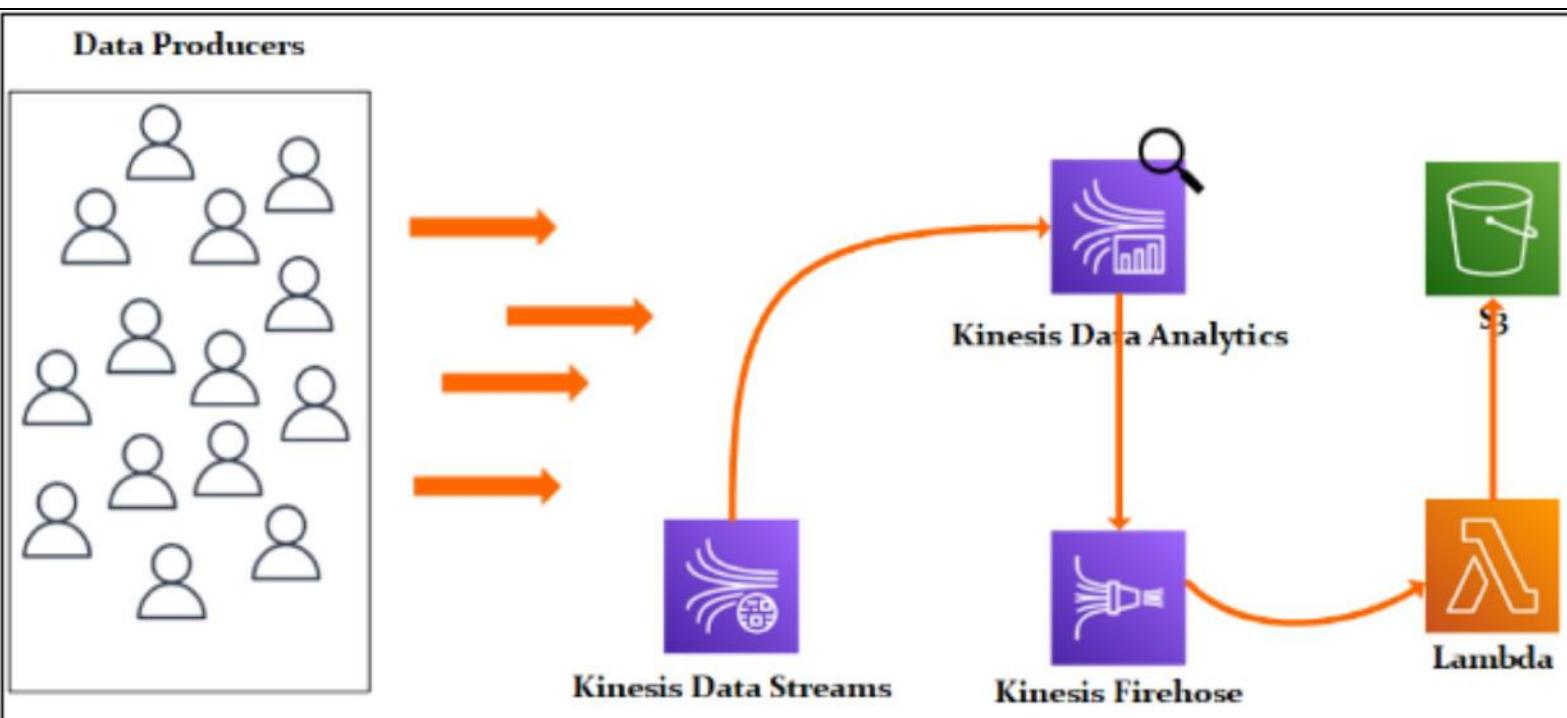
Before starting the lab, a Python script is used to collect the data. The Python code is provided in the following GitHub link (Chapter-3)

[https://github.com/12920/IPSpecialist01/blob/main/Course_AWS_Certified_Machine_Learning-master%20\(1\).zip](https://github.com/12920/IPSpecialist01/blob/main/Course_AWS_Certified_Machine_Learning-master%20(1).zip)

This Python script is integrated into the CloudFormation stack YAML file. In this lab, only a template is used to create a stack and collect the data.

```
1 import requests
2 import boto3
3 import uuid
4 import time
5 import random
6 import json
7
8 client = boto3.client('kinesis', region_name='<INSERT_YOUR_REGION>')
9 partition_key = str(uuid.uuid4())
10
11 # Added 08/2020 since randomuser.me is starting to throttle API calls
12 # The following code loads 500 random users into memory
13 number_of_results = 500
14 r = requests.get('https://randomuser.me/api/?exc=login&results=' + str(number_of_results))
15 data = r.json()["results"]
16
17 while True:
18     # The following chooses a random user from the 500 random users pulled from the API in a single API call.
19     random_user_index = int(random.uniform(0, (number_of_results - 1)))
20     random_user = data[random_user_index]
21     random_user = json.dumps(data[random_user_index])
22     client.put_record(
23         StreamName='<INSERT_YOUR_STREAM_NAME>',
24         Data=random_user,
25         PartitionKey=partition_key)
26     time.sleep(random.uniform(0, 1))
```

Lab Diagram



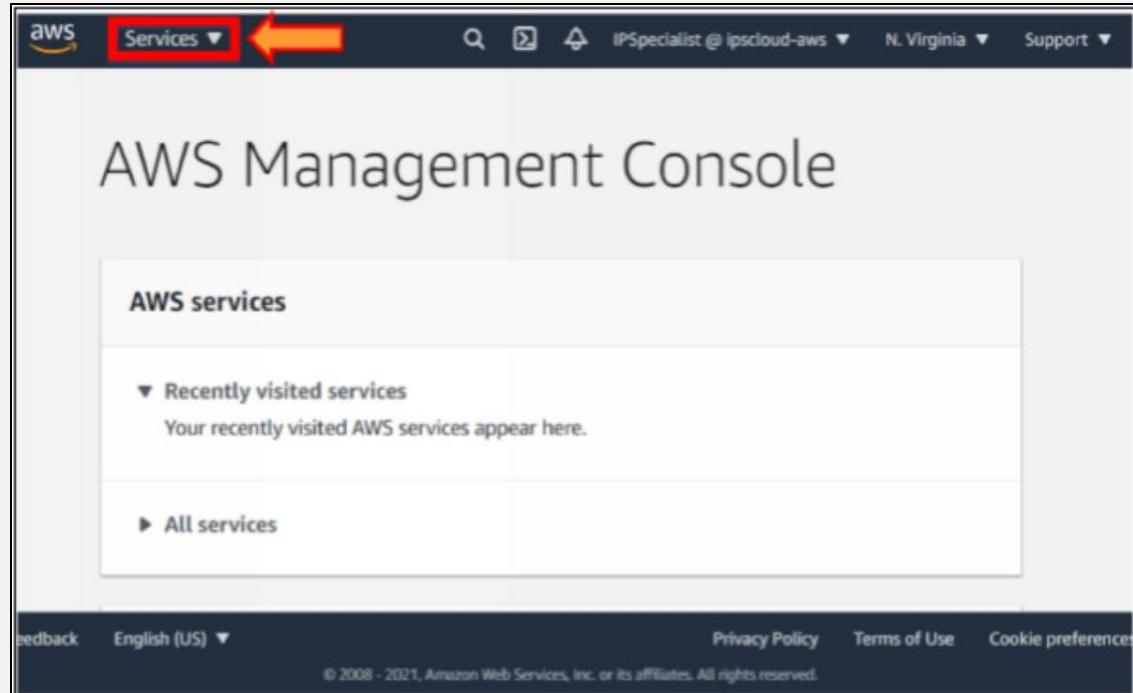
Implementation Steps

1. Create Kinesis Data Stream.
2. Create CloudFormation Stack.
3. Create Kinesis Data Analytics.
4. Create Kinesis Data Firehose.
5. View Collected Data.

Solution

Step 1: Create Kinesis Data Stream

1. Log in to the **AWS Console**.
2. Click on **Services**.



3. Select **Kinesis** from **Analytics**.

AWS Services ▾ Q Search for services, features, marketplace product: [Alt+S] IPSpecialist @ ipscloud-aws N. Virginia Support

★ Favorites

Add favorites by clicking on the star next to the service name.

Recently visited

Console Home

All services

DynamoDB	AWS AppConfig	Managed Apache Airflow
ElastiCache	Trusted Advisor	
Neptune	Control Tower	
Amazon QLDB	AWS License Manager	
Amazon DocumentDB	AWS Well-Architected Tool	
Amazon Keyspaces	Personal Health Dashboard	
Amazon Timestream	AWS Chatbot	
Amazon MemoryDB for Redis	AWS Compute Optimizer	
	Resource Groups & Tag Editor	
		Kinesis ←
		QuickSight ↗
		Data Pipeline

Get started with simple wizards and automated workflows.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

4. Select **Kinesis Data Streams**.
5. Click on the **Create Data Stream** button.

AWS Services ▾ Q Search for services, features, marketplace product: [Alt+S] IPSpecialist @ ipscloud-aws N. Virginia Support

Analytics

Amazon Kinesis services

Collect, process, and analyze data streams in real time.

Get started

- Kinesis Data Streams**
Collect streaming data with a data stream.
- Kinesis Data Firehose
Process and deliver streaming data with data delivery stream.
- Kinesis Data Analytics
Analyze streaming data with data analytics application.

Create data stream ←

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

6. Give the name **IPS-my-data-stream**.

AWS Services ▾ Q Search for services, features, marketplace product: [Alt+S] IPSpecialist @ ipscloud-aws N. Virginia Support

Amazon Kinesis > Data streams > Create data stream

Create a data stream Info

Data stream configuration

Data stream name ←

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.

Data stream capacity Info Request limit increase ↗

Data records are stored in Kinesis Data Stream. A shard is a uniquely identified sequence of data records in a

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

7. To calculate the number of shards, click on the **Shard Estimator**.

The screenshot shows the 'Data stream capacity' section of the AWS Kinesis console. At the top right is a 'Request limit increase' button. Below it is a 'Shard estimator' button, which is highlighted with a red box and an orange arrow pointing to it. The 'Shard estimator' button has a tooltip 'Shard estimator'. The page also includes sections for 'Number of open shards' and 'Total data stream capacity', along with various input fields and account information at the bottom.

8. Give the following values in **Shard Calculator**.

- Average record size: **1 KB**
- Max records are written: **5 per second**
- Number of consumer applications: **1**
- The result is Estimated shards: **1**

The screenshot shows the 'Shard calculator' page. It features input fields for 'Average record size (in KiB)', 'Maximum records written per second', and 'Total number of consumers', each with a value of 1. The page also includes sections for 'Estimated number of open shards' (showing 1) and 'Number of open shards' (with a 'Cost calculator' button). The bottom of the page contains standard AWS navigation links like 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

- Set number of shards: **1**

AWS Services ▾ Search for services, features, marketplace product: [Alt+S] IP Specialist @ ipscloud-aws N. Virginia

Apply this value Cost calculator ↗

Number of open shards
Each shard ingests up to 1 MiB/second and 1000 records/second and emits up to 2 MiB/second.

1 

Minimum: 1, Maximum: 500, Account limit: 500.

Total data stream capacity
Total data stream capacity is calculated based on the number of shards entered above.

Write
1 MiB/second, 1000 Data records/second

Read
2 MiB/second

Cancel Create data stream

Feedback English (US) ▾ Privacy Policy Terms of Use © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Click on the Create Data Stream button.

AWS Services ▾ Search for services, features, marketplace product: [Alt+S] IP Specialist @ ipscloud-aws N. Virginia

Apply this value Cost calculator ↗

Number of open shards
Each shard ingests up to 1 MiB/second and 1000 records/second and emits up to 2 MiB/second.

1 

Minimum: 1, Maximum: 500, Account limit: 500.

Total data stream capacity
Total data stream capacity is calculated based on the number of shards entered above.

Write
1 MiB/second, 1000 Data records/second

Read
2 MiB/second

Cancel Create data stream 

Feedback English (US) ▾ Privacy Policy Terms of Use © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hence, the Kinesis data stream has been created.

AWS Services ▾ Search for services, features, marketplace product: [Alt+S] IP Specialist @ ipscloud-aws N. Virginia Support

Amazon Kinesis  IPS-my-data-stream was successfully created. 

Dashboard

Data streams  IPS-my-data-stream  Delete

Stream details

Status	ARN	Data retention period	Creation time
Active	arn:aws:kinesis:us-east-1:54473827:stream/IPS-my-data-stream	1 day	September 23, 2021, 04:40 PDT

Applications Monitoring Configuration Enhanced fail-over (0)

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie prefs © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2: Create CloudFormation Stack

1. Click on Services.

The screenshot shows the AWS Management Console with the 'Services' button highlighted by a red box and a yellow arrow pointing to it. The main content area is titled 'AWS Management Console' and contains a sidebar titled 'AWS services'. The sidebar includes sections for 'Recently visited services' (with a note: 'Your recently visited AWS services appear here.') and 'All services'. At the bottom of the sidebar, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the very bottom states: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

2. Select CloudFormation from Management & Governance.

The screenshot shows the 'All services' page of the AWS Management Console. On the left, there's a sidebar with 'Favorites' (empty), 'Recently visited' (Console, Home), and a 'Feedback' link. The main area is titled 'All services' and lists various AWS services under four categories: 'Management & Governance', 'Storage', 'CloudFormation' (which is highlighted with a red box and a yellow arrow), and 'Application Integration'. Each category has a list of services underneath. At the bottom, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

3. Click on the Create Stack button.

AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your environment in a safe, repeatable way.

Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

Create stack

How it works

edbck English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4. Select Template is ready.

CloudFormation > Stacks > Create stack

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready Use a sample template Create template in Designer

edbck English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Download the template file provided in the following GitHub link:

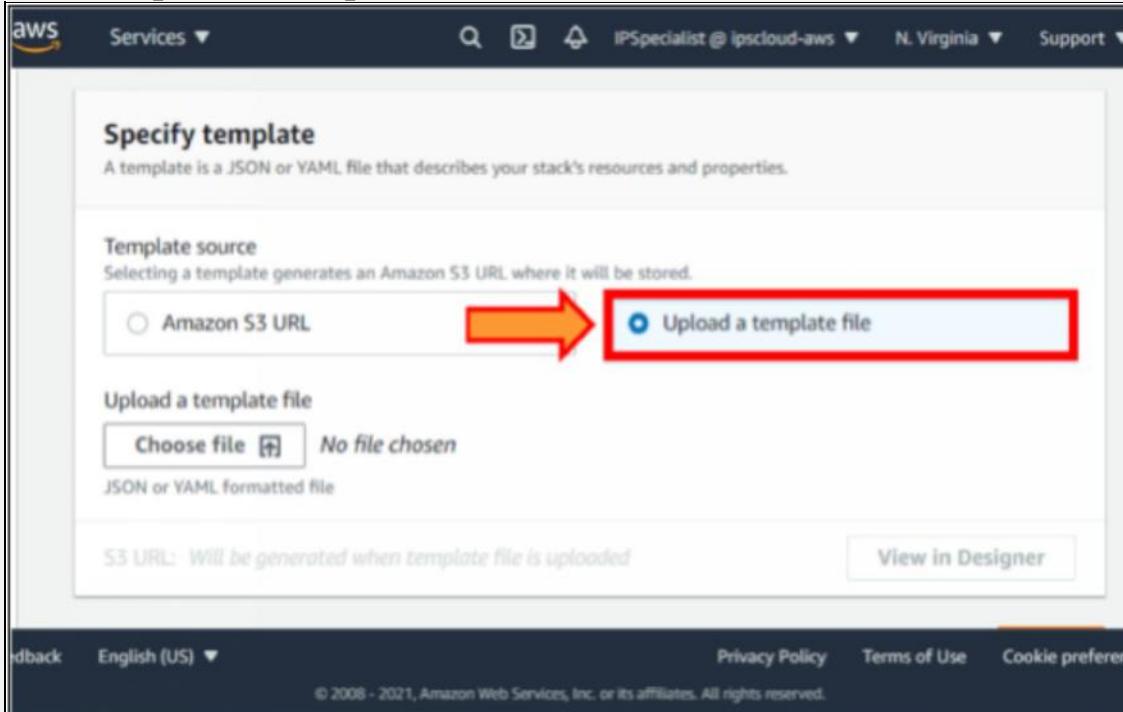
https://raw.githubusercontent.com/ACloudGuru/Resources/Course_AWS_Certified_Machine_Learning/master/Chapter3/setup-data-producer.yml.

```

Parameters:
  KinesisDataStream:
    Description: The name of your Kinesis Data Stream.
    Type: String
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0080e4c5bc078760e
    us-east-2:
      AMI: ami-0cd3dfa4e37921605
    us-west-1:
      AMI: ami-0ec6517f6edbf8044
    us-west-2:
      AMI: ami-01e24be29428c15b2
Resources:
  LnDataProducerInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t2.micro
      ImageId:
        Fn::FindInMap:
          - RegionMap
          - !Ref AWS::Region
          - AMI
      IamInstanceProfile: !Ref LnInstanceProfiler
    UserData:
      Fn::Base64:
        !Join [ "", [
          "#!/bin/bash -xe\n",
          "sudo /opt/aws/bin/cfn-init -v ", #use cfn-init to install packages in cloudformation init
          !Sub "--stack ${AWS::StackName} ",
          "--resource LnDataProducerInstance ",
          "--configsets InstallAndConfigure ",
          !Sub "--region ${AWS::Region}".
        ]]

```

6. Select Upload a template file.



The screenshot shows the AWS CloudFormation 'Specify template' interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and user information ('IPSpecialist @ ipscloud-aws', 'N. Virginia', 'Support'). Below the navigation, the title 'Specify template' is displayed, followed by the sub-instruction: 'A template is a JSON or YAML file that describes your stack's resources and properties.' Under the 'Template source' section, there are two options: 'Amazon S3 URL' (radio button) and 'Upload a template file' (radio button, which is selected and highlighted with a red box and an orange arrow). Below these options, there's a 'Choose file' button with the text 'No file chosen'. Further down, there's a note about 'JSON or YAML formatted file' and a 'S3 URL: Will be generated when template file is uploaded' field. At the bottom of the page, there are links for 'View in Designer', 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

7. Click on the Choose file button.

Screenshot of the AWS CloudFormation 'Specify template' page. The 'Upload a template file' section is highlighted with a red box and a yellow arrow pointing to the 'Choose file' button. The URL field below it shows 'chosen'.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file

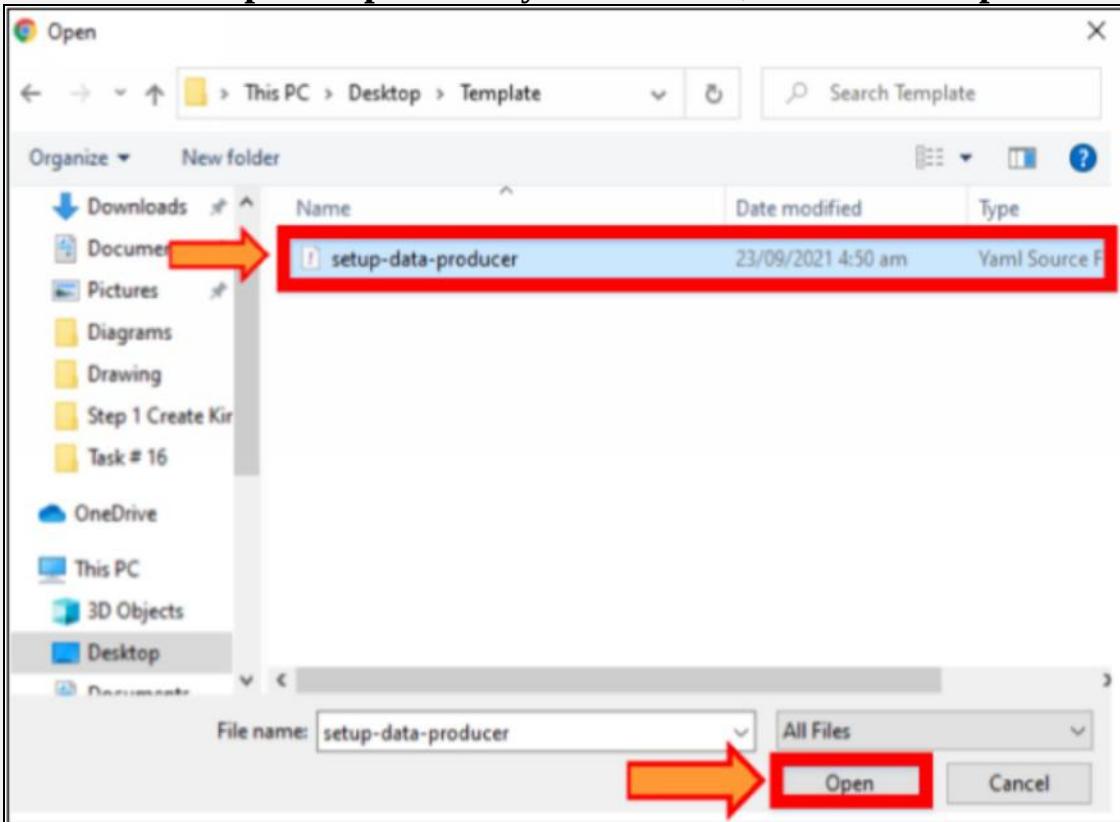
Upload a template file
Choose file

JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded

View in Designer

8. Select the **setup-data-producer.yml** file. Then, click on the **Open** button.



9. Click on the **Next** button.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file

Upload a template file

Choose file setup-data-producer.yml

JSON or YAML formatted file

S3 URL: <https://s3-external-1.amazonaws.com/cf-templates-1taywmeqps8cu-us-east-1/20212666s0-setup-data-producer.yml> View in Designer

Next

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10. Name the stack as **IPS-data-producer-stack**.

CloudFormation > Stacks > Create stack

Specify stack details

Stack name

Stack name IPS-data-producer-stack

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11. Then, give the Kinesis data stream name **IPS-my-data-stream**.

Sales

Services ▾

Stack name

IPS-data-producer-stack

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

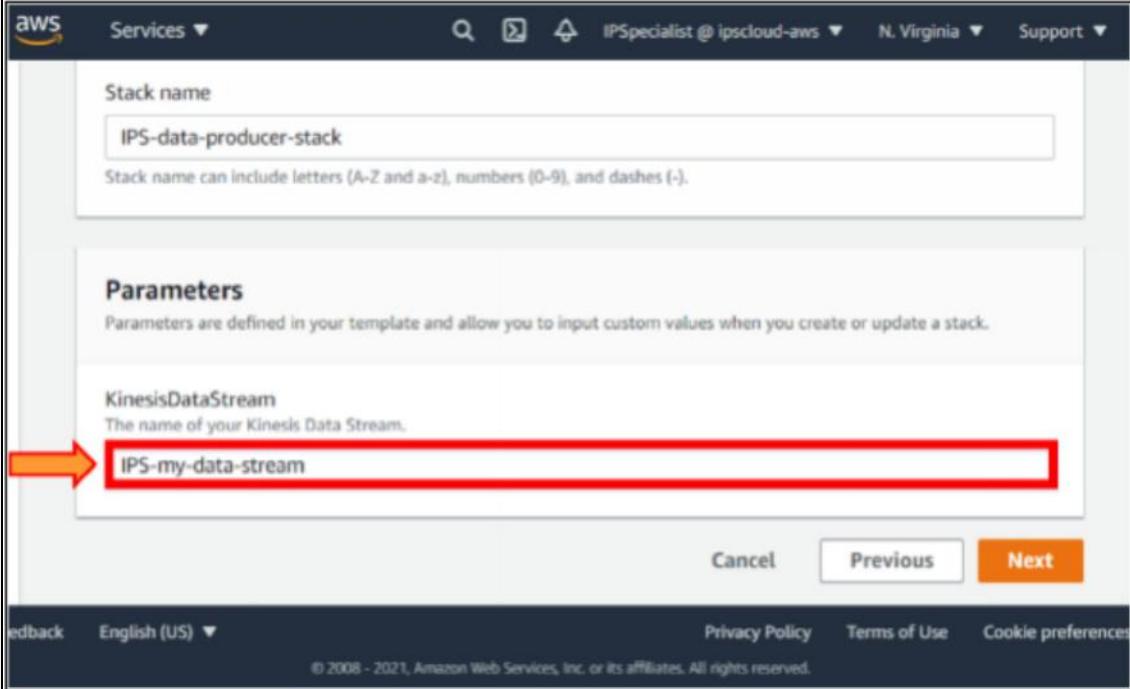
KinesisDataStream
The name of your Kinesis Data Stream.

IPS-my-data-stream

Cancel Previous Next

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



12. Click on the **Next** button.

Sales

Services ▾

Stack name

IPS-data-producer-stack

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

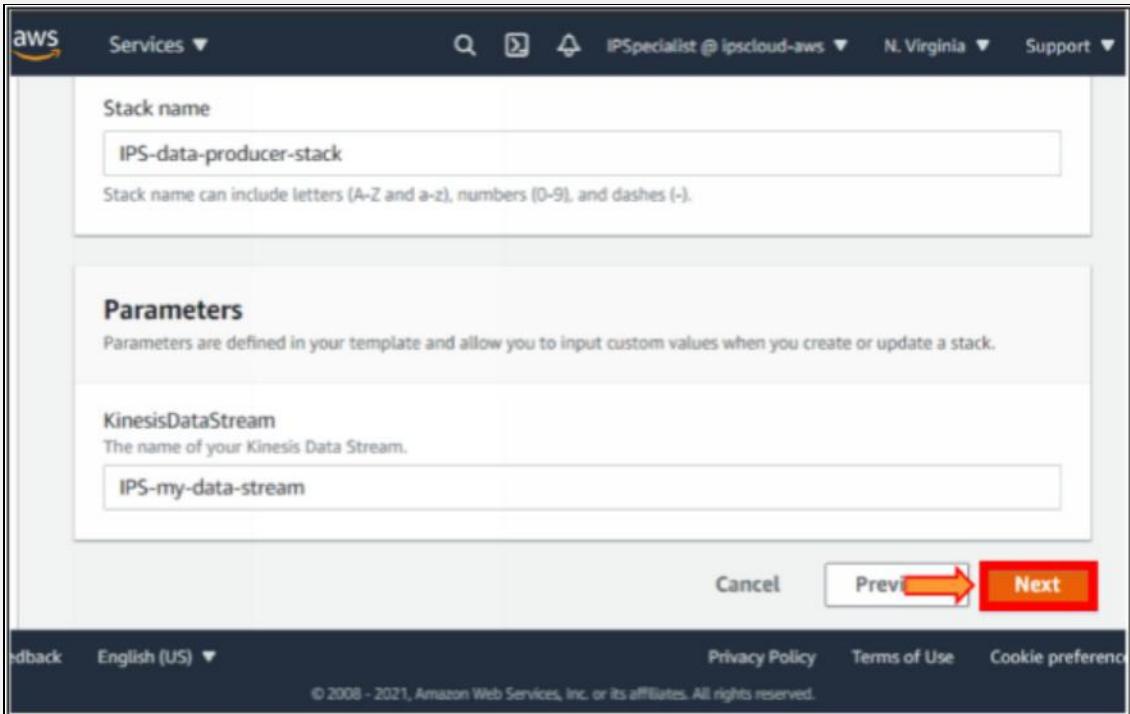
KinesisDataStream
The name of your Kinesis Data Stream.

IPS-my-data-stream

Cancel Previous Next

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



13. Scroll down. Click on the **Next** button.

▶ Rollback configuration

Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)

▶ Notification options

▶ Stack creation options

Cancel Previous **Next**

edbck English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

14. Scroll down. Click on the check box.

▶ Quick-create link

Capabilities

ⓘ The following resource(s) require capabilities: [AWS::IAM::InstanceProfile, AWS::IAM::Role]

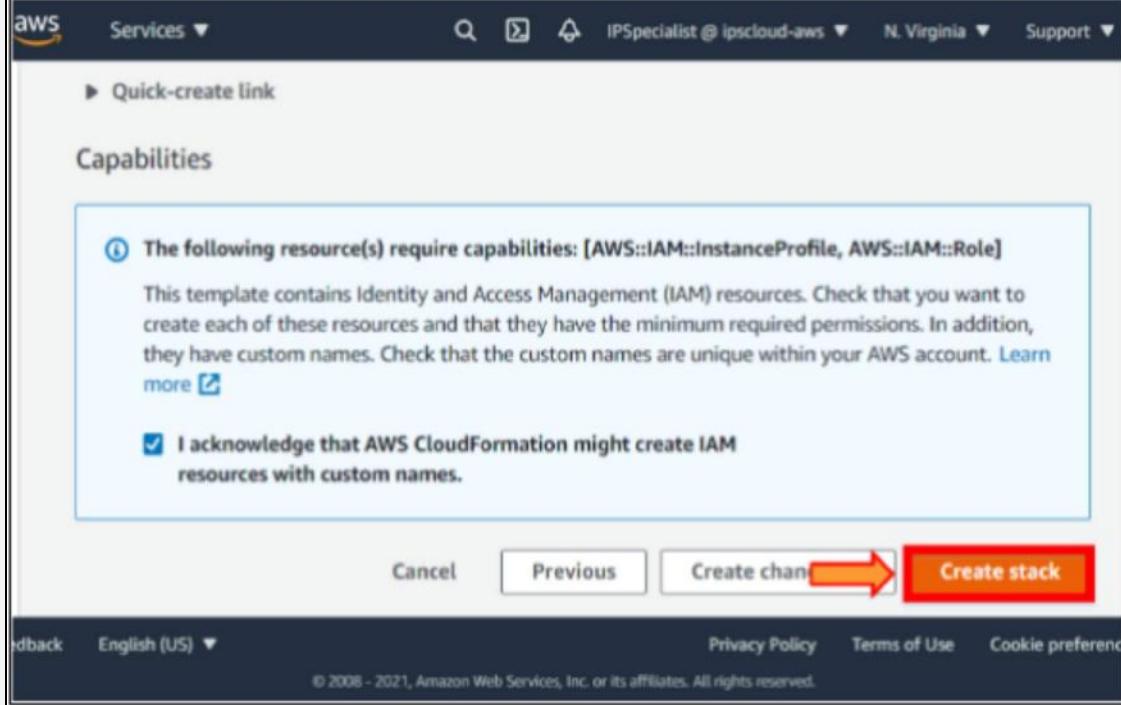
This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

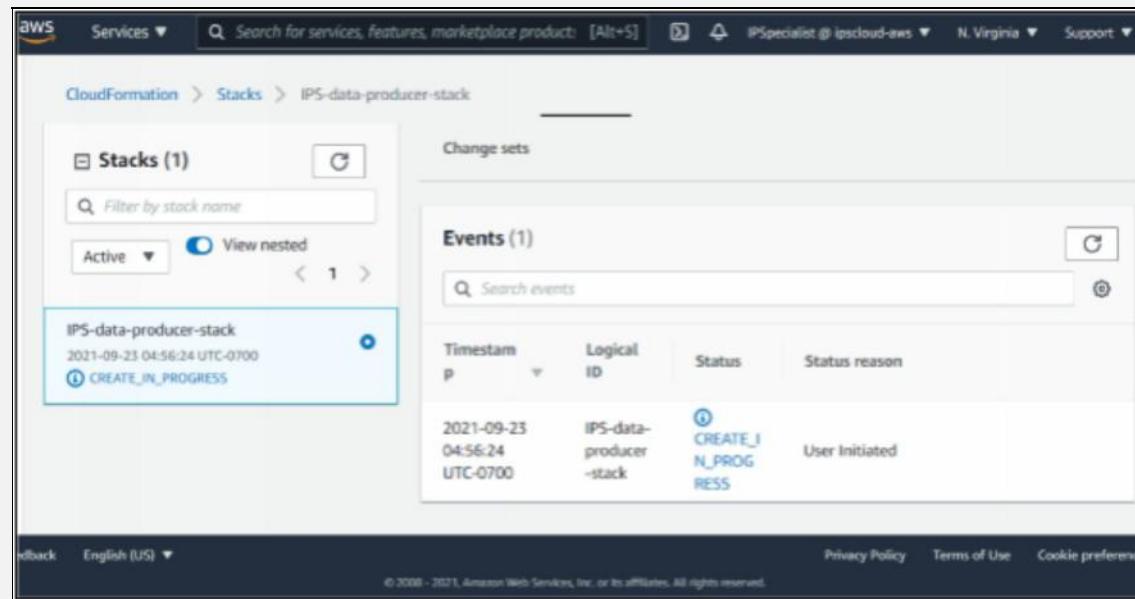
Cancel Previous Create change set **Create stack**

edbck English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

15. Then, click on the **Create Stack** button.



16. It will take a few minutes to create a stack. Once completed, CloudFormation will create all the services and automatically start putting that random user data into the Kinesis stream.



Hence, CloudFormation Stack work is completed.

The screenshot shows the AWS CloudFormation console. On the left, under 'Stacks (1)', there is a list with one item: 'IPS-data-producer-stack' (Status: CREATE_COMPLETE). On the right, under 'Events (14)', there are two entries:

Timestamp	Logical ID	Status
2021-09-23 04:59:44 UTC-0700	IPS-data-producer-stack	CREATE_COMPLETE
2021-09-23	LnDataPr	CREATE_IN_PROGRESS

At the bottom, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

17. Go back to the **Kinesis Data Stream** dashboard, and click on the **IPS-my-data-stream**.

The screenshot shows the Amazon Kinesis Data Streams console. Under 'Data streams (1)', the stream 'IPS-my-data-stream' is listed. A red arrow points to the first column of this row, which contains a checkbox and the stream name 'IPS-my-data-stream'. The other columns show 'Active' status, 1 shard, and a 1 day retention period.

18. Click on the **Monitoring** tab.

Sales

Services ▾

Search

IPSspecialist @ ipscloud-aws

N. Virginia

Support

Applications Monitoring Configuration Enhanced fan-out (0)

Stream metrics Info

1h 3h 12h 1d 3d 1w Custom (30m) C ▾

Get records - sum (Bytes)

1 No data available.

0.5

0.5

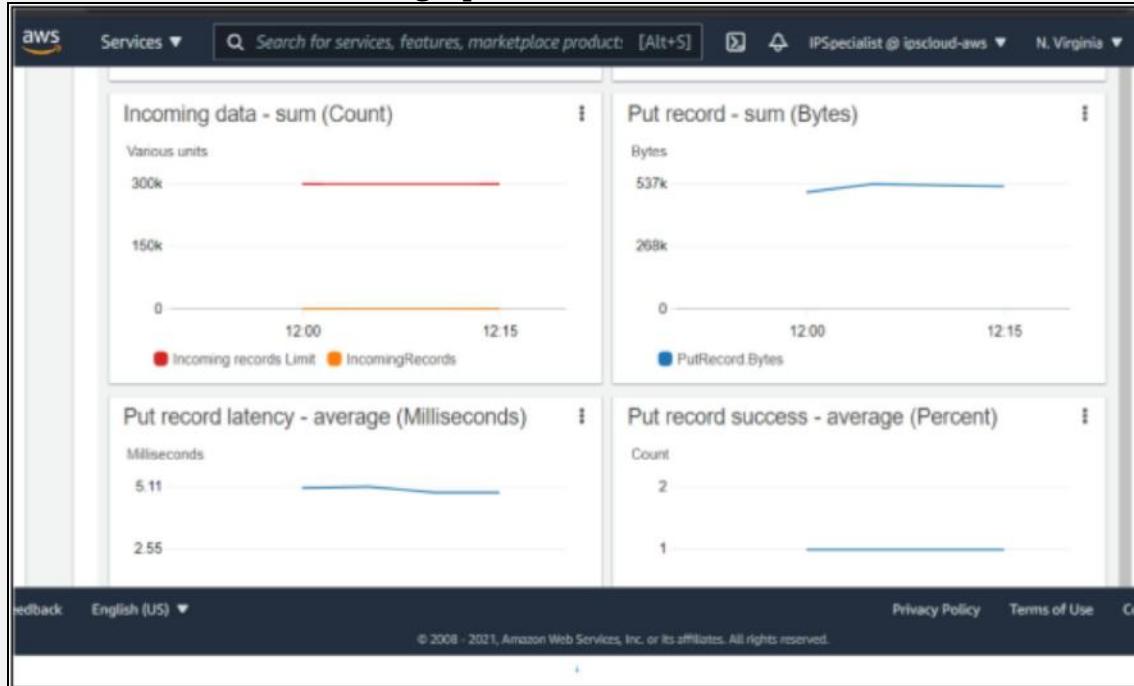
Get records iterator age

1 No data available.

0.5

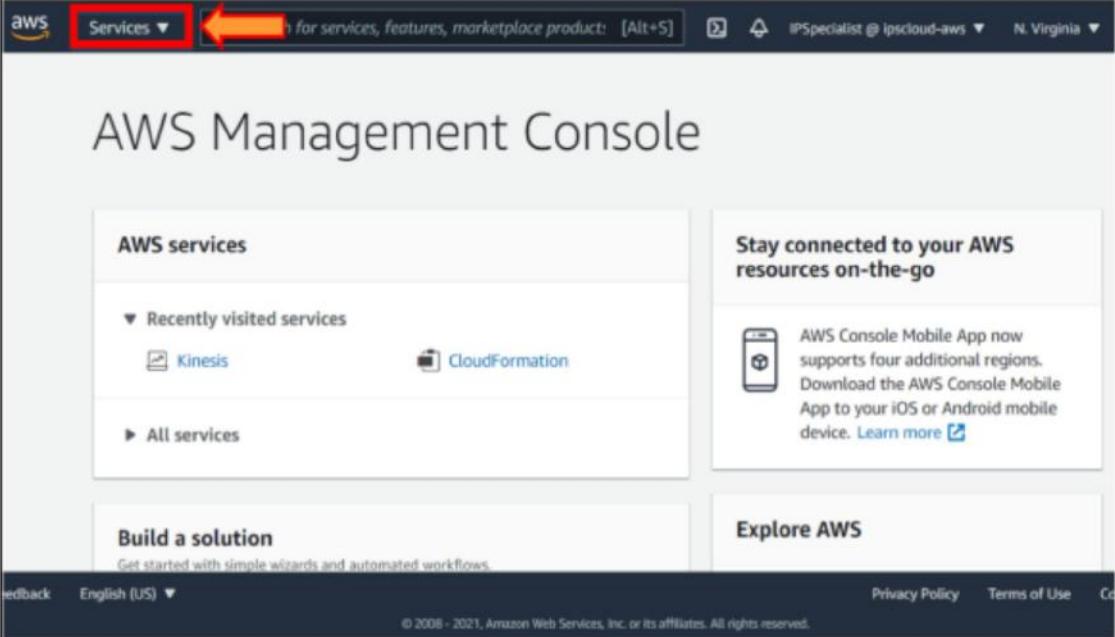
Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

19. You will see different graphs.



Step 3: Create Kinesis Data Analytics

1. Click on Services.



AWS Management Console

AWS services

- Recently visited services
 - Kinesis
 - CloudFormation
- All services

Stay connected to your AWS resources on-the-go

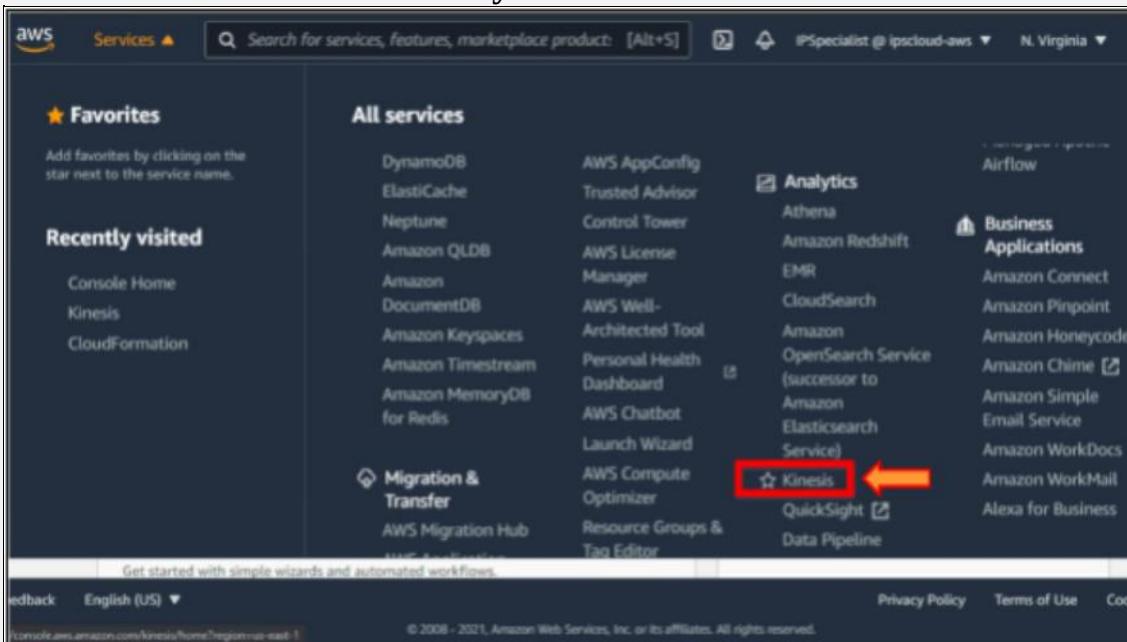
AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. [Learn more](#)

Build a solution

Explore AWS

Feedback English (US) Privacy Policy Terms of Use © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Select the Kinesis from the Analytics.



Services ▾ Search for services, features, marketplace product: [Alt+S] iPSpecialist @ ipscloud-aws N. Virginia

Favorites

Add favorites by clicking on the star next to the service name.

Recently visited

- Console Home
- Kinesis
- CloudFormation

All services

DynamoDB	AWS AppConfig	Airflow
ElastiCache	Trusted Advisor	
Neptune	Control Tower	
Amazon QLDB	AWS License Manager	Athena
Amazon DocumentDB	AWS Well-Architected Tool	Amazon Redshift
Amazon Keyspaces	Architected Tool	EMR
Amazon Timestream	Personal Health Dashboard	CloudSearch
Amazon MemoryDB for Redis	AWS Chatbot	Amazon OpenSearch Service (successor to Elasticsearch Service)
	Launch Wizard	Amazon
	AWS Compute Optimizer	OpenSearch Service
	Resource Groups & Tag Editor	Amazon Chime
		Amazon Honeycode
		Amazon Simple Email Service
		Amazon WorkDocs
		Amazon WorkMail
		Alexa for Business

Migration & Transfer

AWS Migration Hub

Kinesis

Get started with simple wizards and automated workflows.

Feedback English (US) Privacy Policy Terms of Use © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Select Kinesis Data Analytics.

4. Click on the Create Application button.



Services ▾

Q Search for services, features, marketplace product: [Alt+S]



IPSpecialist @ ipscloud-aws ▾

N. Virginia ▾

Analytics

Amazon Kinesis services

Collect, process, and analyze data streams in real time.

Get started

- Kinesis Data Streams
Collect streaming data with a data stream.
- Kinesis Data Firehose
Process and deliver streaming data with data delivery stream.
- Kinesis Data Analytics
Analyze streaming data with data analytics application.

[Create application](#)

5. Select Legacy SQL.

Choose a method to set up the application

Runtime

After you create the application, you can't change the type or version of the runtime environment.

Apache Flink - Streaming application
Apache Flink is an open-source framework and distributed processing engine for stateful computations over unbounded and bounded data streams. Use this option to build streaming application using Apache Flink in Java, Scala, and Python. You can also build Java-based streaming applications using Apache Beam. Apache Beam is an open source, unified model and set of language-specific SDKs for defining and executing data processing workflows.

Legacy SQL
Process data in real-time using Kinesis Data Analytics legacy SQL engine, which provides an easy way to quickly query large volumes of streaming data. We do not recommend this option for new applications, instead use [Studio applications](#).

⚠️ For new applications, we recommend that you use [Kinesis Data Analytics Studio](#) instead of Kinesis Data Analytics for SQL (legacy) for running SQL queries. Kinesis Data

Privacy Policy Terms of Use Cookie preferences

6. Give an application name **IPS-data-transformation-application**.

Sales

Services ▾

Search

IPSspecialist @ ipscloud-aws ▾

N. Virginia ▾

Support

Application configuration

Application name

IPS-data-transformation-application

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Description - optional

Enter description

Access to application resources

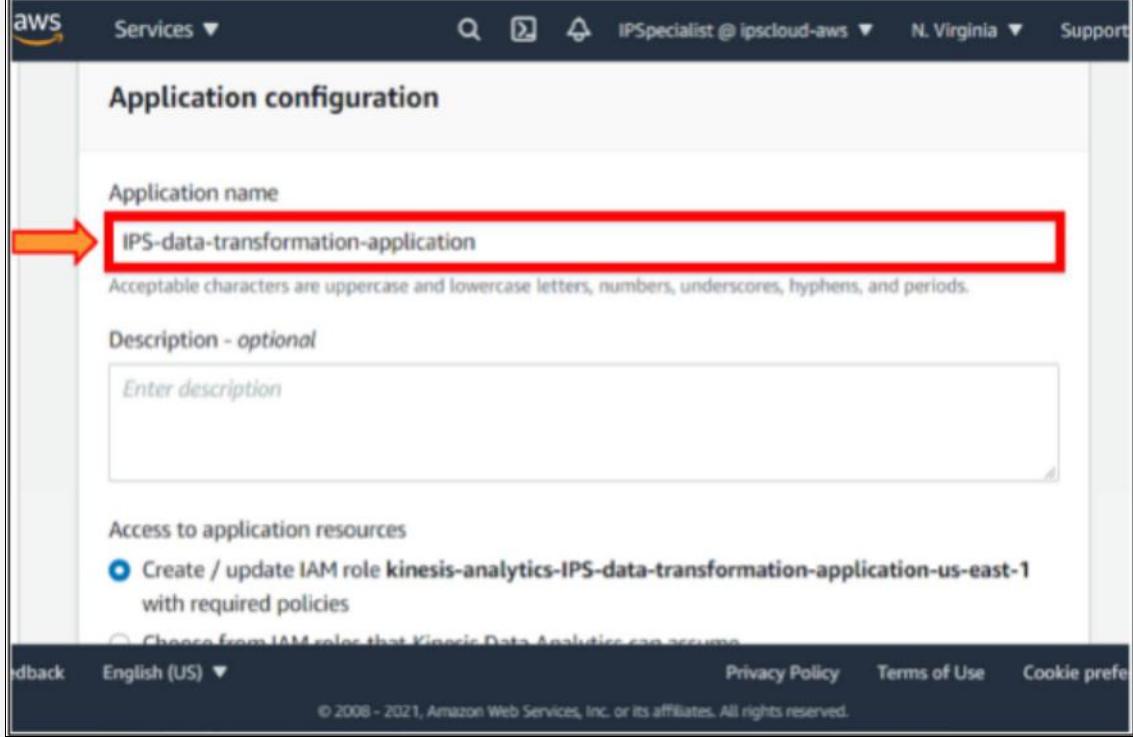
Create / update IAM role kinesis-analytics-IPS-data-transformation-application-us-east-1 with required policies

Choose from IAM roles that Kinesis Data Analytics can assume

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



7. In the **Description** box, paste the following **Help transform user data.**

Sales

Services ▾

Search

IPSspecialist @ ipscloud-aws ▾

N. Virginia ▾

Support

Application configuration

Application name

IPS-data-transformation-application

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Description - optional

Help transform user data.

Access to application resources

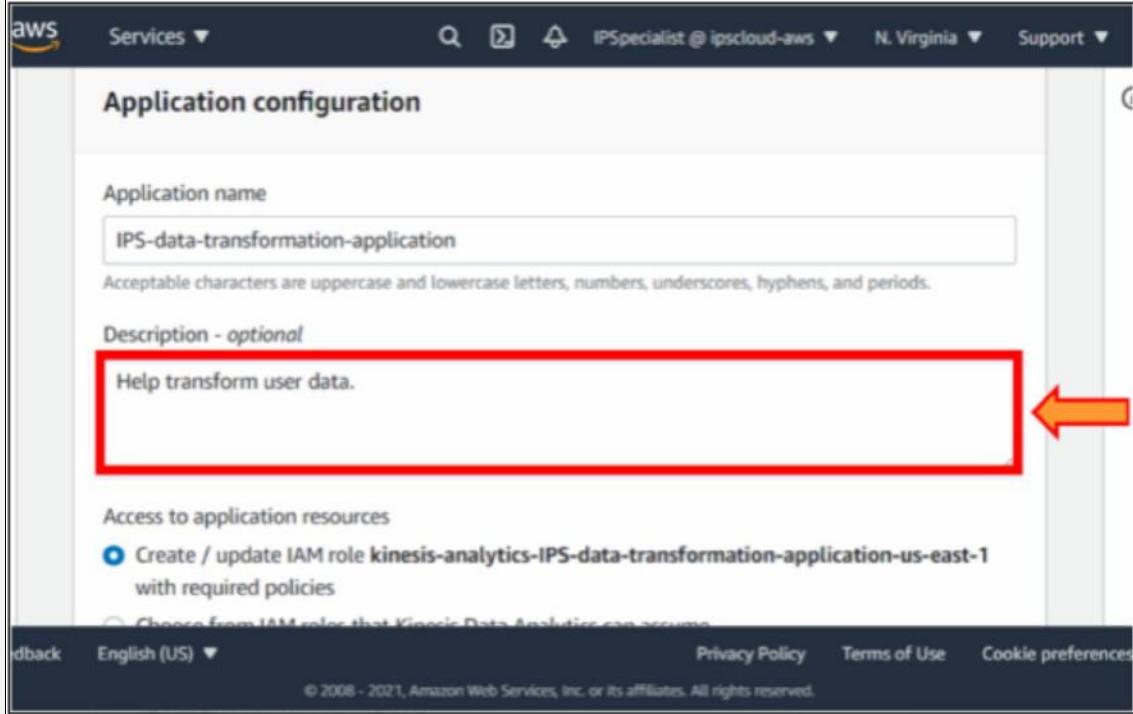
Create / update IAM role kinesis-analytics-IPS-data-transformation-application-us-east-1 with required policies

Choose from IAM roles that Kinesis Data Analytics can assume

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



8. Click on the **Create application** button.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. [Learn more](#)

No tags associated with this application.

Add tag

You can add up to 50 tags.

▶ **Create application**

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Click on Steps to configure your application.

Amazon Kinesis

Dashboard

Data streams

Delivery streams

Analytics applications

Resources

What's new

AWS Streaming Data Solution for Amazon Kinesis

AWS Glue Schema Registry

▶ **Steps to configure your application** Info

Application details

Status	ARN
Ready	arn:aws:kinesisanalytics:us-east-1:644738277497:application/IPS-data-transformation-application
Runtime	Application version ID
SQL	1
Last updated	Description
September 23, 2021, 05:34 PDT	Help transform user data.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10. Click on the Configure source stream button.

The screenshot shows the 'Amazon Kinesis' service dashboard. On the left, a sidebar lists 'Analytics applications' under the 'Resources' section. The main content area is titled 'Steps to configure your application' with 'Step 1: Configure source stream'. It describes using an existing Kinesis data stream or Kinesis Data Firehose delivery stream as input. A red arrow points to the 'Configure source stream' button.

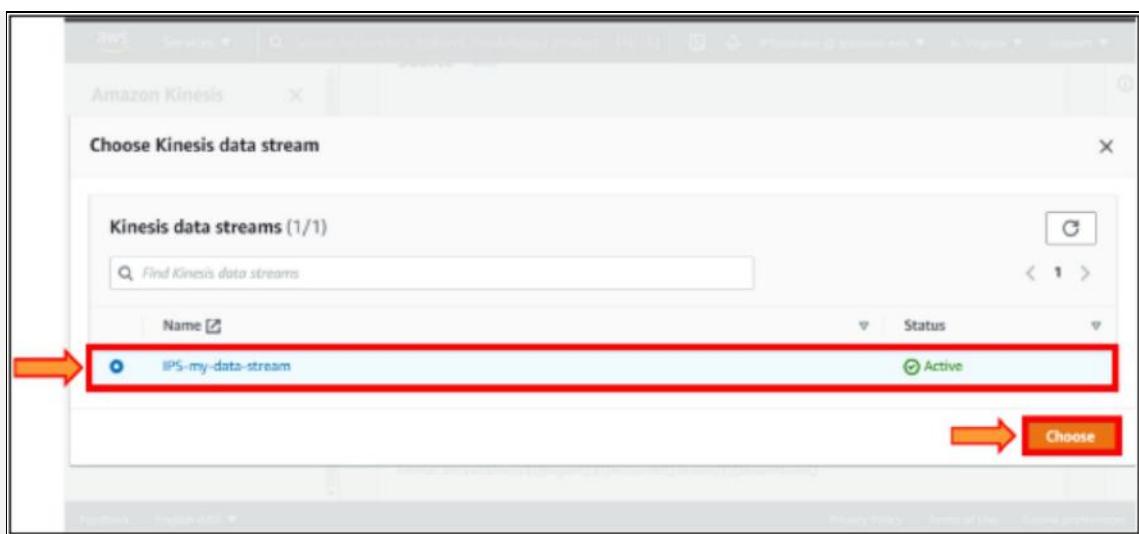
11. Select Kinesis Data Stream.

The screenshot shows the 'Source' configuration page. Under 'Source', the 'Kinesis data stream' option is selected, indicated by a blue radio button. A red arrow points to this selection. Below it, the 'Kinesis Data Firehose delivery stream' option is available. Further down, there's a 'Use demo stream and SQL code' section with a 'Use demo stream' button. At the bottom, there's a 'Kinesis data stream' section with a 'Choose a data stream or enter a data stream ARN' input field, a 'Browse' button (which is redboxed), and a 'Create' button.

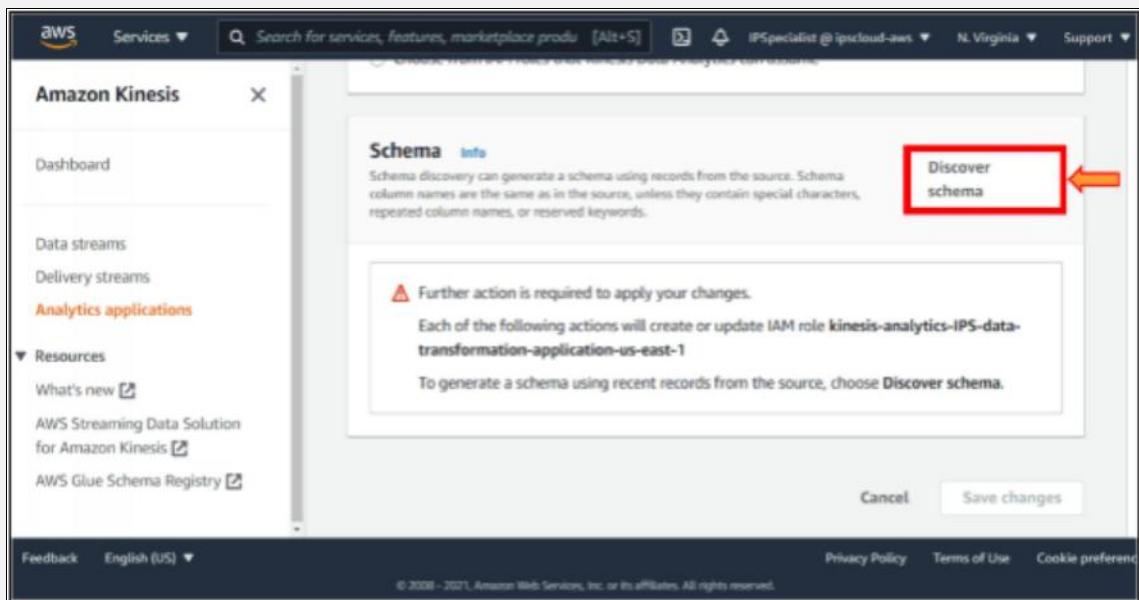
12. Click on the Browse button.

This screenshot is identical to the previous one, showing the 'Source' configuration page. The 'Kinesis data stream' option is selected. A red arrow points to the 'Browse' button in the 'Kinesis data stream' section, which is highlighted with a red box.

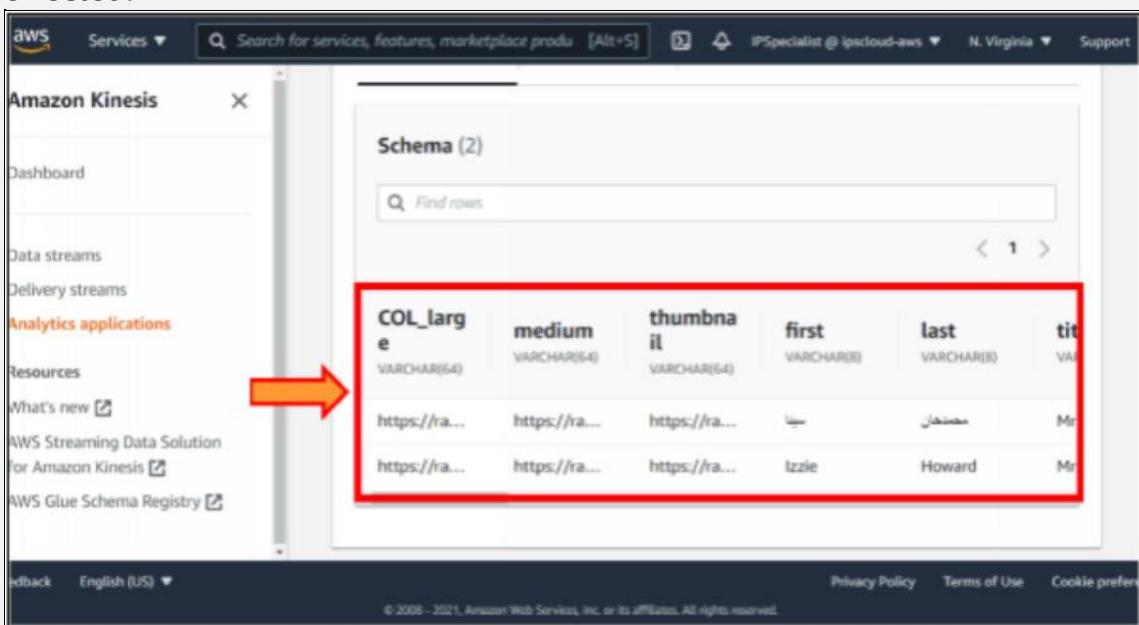
13. Select IPS-my-data-stream. Then, click on the Choose button.



14. Scroll down. Click on the **Discover Schema** button.



15. You will see the Raw and Formatted streaming data which has been collected.



16. Click on the **Save changes** button.

AWS Services Search for services, features, marketplace products [Alt+S] IPSpecialist @ ipscloud-aws N. Virginia Support

Amazon Kinesis

Dashboard Data streams Delivery streams **Analytics applications** Resources What's new AWS Streaming Data Solution for Amazon Kinesis AWS Glue Schema Registry

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

COL_larg medium thumbna first last tit

https://ra... https://ra... https://ra... https://ra... https://ra... https://ra...

https://ra... https://ra... https://ra... tzzie Howard Mr...

Save changes

17. Click on the **Real-time analytics** tab. Then, click on the **Configure** button.

AWS Services Search for services, features, marketplace products [Alt+S] IPSpecialist @ ipscloud-aws N. Virginia Support

Amazon Kinesis

Dashboard Data streams Delivery streams **Analytics applications** Resources What's new AWS Streaming Data Solution for Amazon Kinesis AWS Glue Schema Registry

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

Creation time
September 23, 2021, 05:34 PDT

Source **Real-time analytics** Tags

SQL code Info View raw SQL Configure

```
1 /**
2 * Welcome to the SQL editor
3 *
4 *
5 * The SQL code you write here will continuously transform your streaming
6 * data
7 *
8 * Get started by clicking "Add SQL from templates" or pull up the
9 * "Transform" tab to see what's happening to your data.
```

18. Now copy and paste the SQL queries in the code editor from the file provided in the following GitHub link: https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter3/create-subset-transformation-query.sql
19. Click on the **Save and run application** button.

```

1 CREATE OR REPLACE STREAM "DESTINATION_USER_DATA" (
2     first VARCHAR(16),
3     last VARCHAR(16),
4     age INTEGER,
5     gender VARCHAR(16),
6     latitude FLOAT,
7     longitude FLOAT
8 );
9 CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO
10      "DESTINATION_USER_DATA"
11      SELECT STREAM "first", "last", "age", "gender", "latitude", "longitude"
12      FROM "SOURCE_SQL_STREAM_001"
13      WHERE "age" >= 21;

```

20. It will take a few minutes. Once it is completed, you will see the subset of data.

ROWTIME	FIRST	LAST	AGE	GENDER	LATITUDE
2021-09-15T12:00:00.000Z	Oscar	Mortense	52	male	58.0
2021-09-15T12:00:00.000Z	Anastaci	Vieira	42	female	13.0
2021-09-15T12:00:00.000Z	Amelia	Lee	30	female	85.0
2021-09-15T12:00:00.000Z	Kjell	Mohammad	72	male	-28.0
2021-09-15T12:00:00.000Z	Chloe	Banks	60	female	-6.0

Step 4: Create Kinesis Data Firehose

1. Click on the **Destination** tab.
2. Click on **Add destination**.

The screenshot shows the AWS CloudWatch Metrics console. At the top, there are tabs for 'Source' and 'Real-time analytics'. Below these, a red arrow points to the 'Destinations' tab, which is highlighted with a red border. To the right of the 'Destinations' tab is another red arrow pointing to the 'Add destinations' button, also highlighted with a red border. The main area displays a table with columns: Destination ..., Destination ..., In-applicati..., and ID. A message at the bottom of the table says 'No destination connected'. At the bottom of the page, there are links for 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

3. Select Kinesis Data Firehose delivery stream.

The screenshot shows the 'Destination' configuration screen. The title is 'application'. It says 'Point to the destinations where you want the results loaded.' Below this, there is a section titled 'Destination' with three options:

- Kinesis data stream
- Kinesis Data Firehose delivery stream
- AWS Lambda function

The second option, 'Kinesis Data Firehose delivery stream', is selected and highlighted with a red box. A red arrow points to this red box from the left. Below the options, there is a note: 'Use a delivery stream to deliver real-time streaming data to destinations such as Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, or Splunk.' At the bottom of the page, there are links for 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

4. Click on the Create button.

aws Services ▾

KINESIS DATA FIREHOSE delivery stream
Use a delivery stream to deliver real-time streaming data to destinations such as Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, or Splunk.

AWS Lambda function
Use an AWS Lambda function to process streaming data.

Delivery stream

Choose a delivery stream or enter a delivery stream ARN

Format: arn:aws:firehose:\$\{Region\}:\\$\{AccountId\}:deliverystream/\$\{DeliveryStreamName\}

Access permissions for writing output stream
Create or choose IAM role with the required permissions. [Learn more](#)

Create / update IAM role kinesis-analytics-IPS-data-transformation-application-us-east-1 with required policies

Choose from IAM roles that Kinesis Data Analytics can assume

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

5. Select the source **Direct PUT**. Then set the destination as **Amazon S3**.

aws Services ▾

▶ Amazon Kinesis Data Firehose: How it works

Choose source and destination
Specify the source and the destination for your delivery stream. You cannot change the source and destination of your delivery stream once it has been created.

Source [Info](#)

Destination [Info](#)

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Give the delivery stream name **IPS-my-delivery-stream**.

Salesback English (US) ▾

Services ▾

IPSspecialist @ ipscloud-aws ▾ N. Virginia ▾ Support

Destination [Info](#)

Amazon S3

Delivery stream name

Delivery stream name

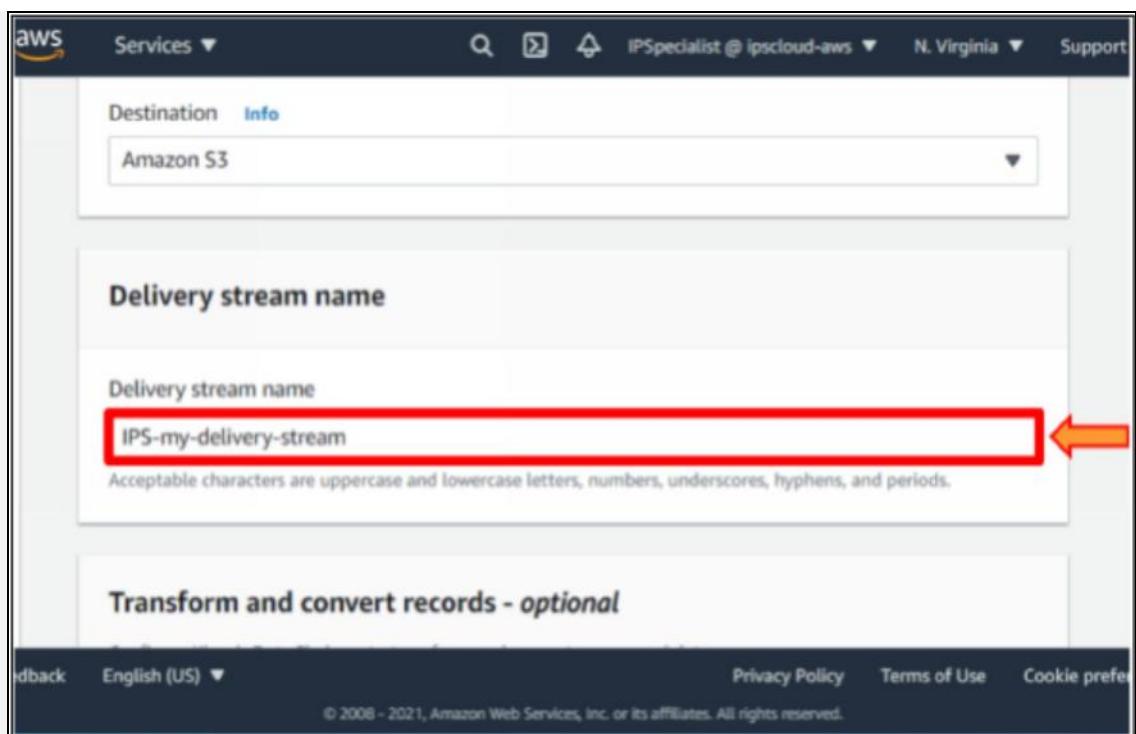
IPS-my-delivery-stream

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Transform and convert records - *optional*

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



7. Scroll down. Select Enabled.

Salesback English (US) ▾

Services ▾

IPSspecialist @ ipscloud-aws ▾ N. Virginia ▾ Support

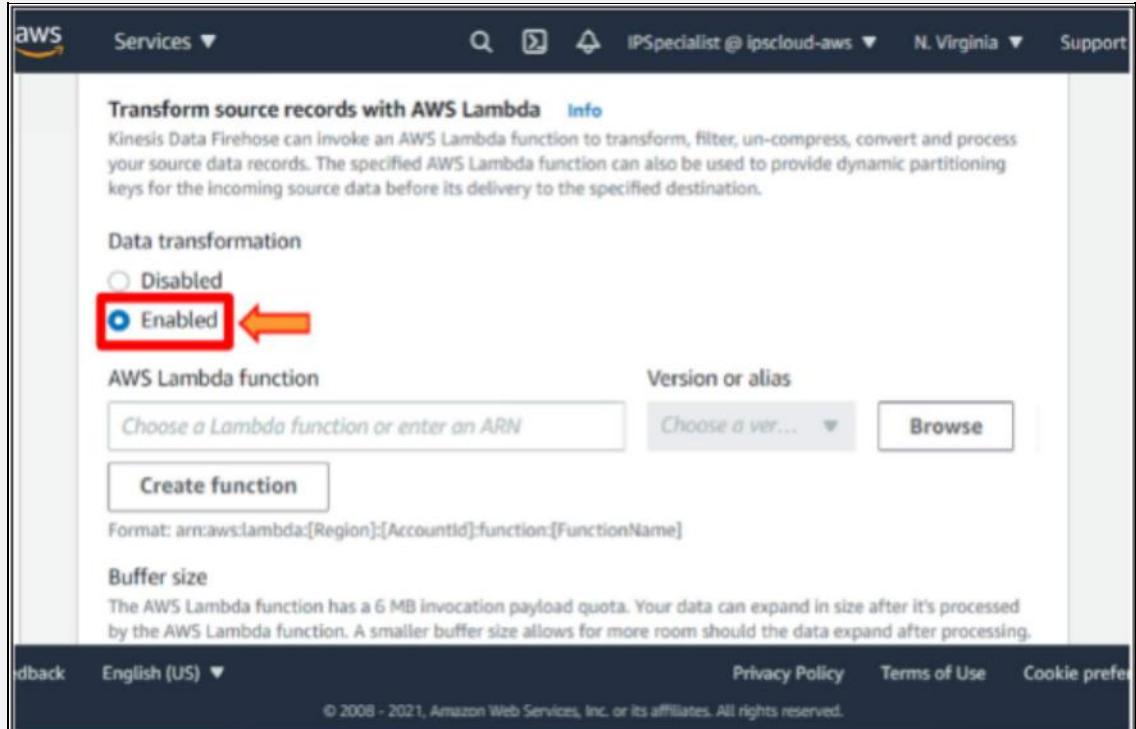
Transform source records with AWS Lambda [Info](#)

Kinesis Data Firehose can invoke an AWS Lambda function to transform, filter, un-compress, convert and process your source data records. The specified AWS Lambda function can also be used to provide dynamic partitioning keys for the incoming source data before its delivery to the specified destination.

Data transformation

Disabled

Enabled



AWS Lambda function

Choose a Lambda function or enter an ARN

Version or alias

Choose a ver... ▾

Browse

Create function

Format: arn:aws:lambda:[Region]:[Accountid]:function:[FunctionName]

Buffer size

The AWS Lambda function has a 6 MB invocation payload quota. Your data can expand in size after it's processed by the AWS Lambda function. A smaller buffer size allows for more room should the data expand after processing.

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. Click on the **Create function** button to create the Lambda function.

AWS Services ▾

Transform source records with AWS Lambda [Info](#)

Kinesis Data Firehose can invoke an AWS Lambda function to transform, filter, un-compress, convert and process your source data records. The specified AWS Lambda function can also be used to provide dynamic partitioning keys for the incoming source data before its delivery to the specified destination.

Data transformation

Disabled

Enabled

AWS Lambda function

Version or alias

Choose a ver... [Browse](#)

[Create function](#) 

Format: arn:aws:lambda:[Region]:[AccountId]:function:[FunctionName]

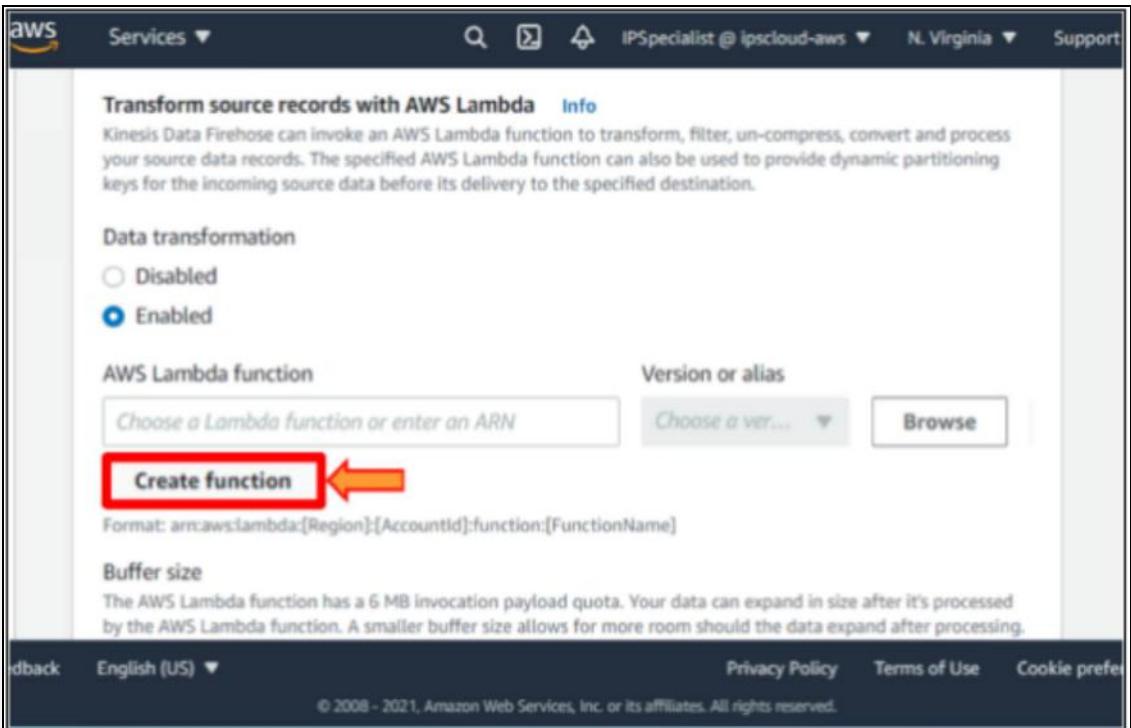
Buffer size

The AWS Lambda function has a 6 MB invocation payload quota. Your data can expand in size after it's processed by the AWS Lambda function. A smaller buffer size allows for more room should the data expand after processing.

dback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



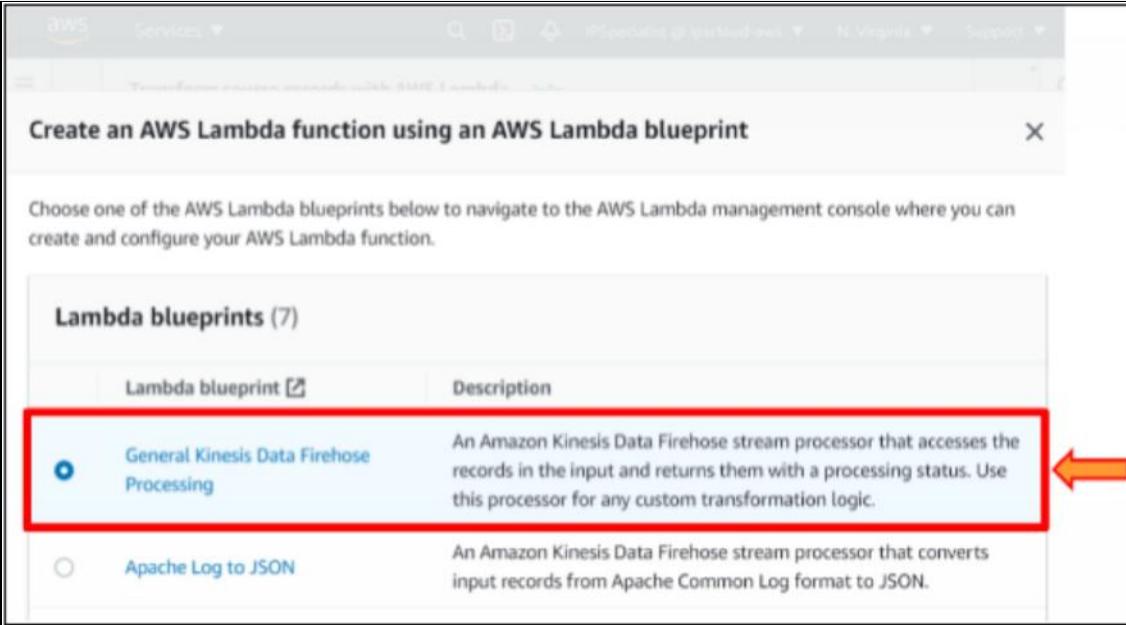
9. Select General Kinesis Data Firehose Processing.

Create an AWS Lambda function using an AWS Lambda blueprint

Choose one of the AWS Lambda blueprints below to navigate to the AWS Lambda management console where you can create and configure your AWS Lambda function.

Lambda blueprints (7)

Lambda blueprint	Description
<input checked="" type="radio"/> General Kinesis Data Firehose Processing	An Amazon Kinesis Data Firehose stream processor that accesses the records in the input and returns them with a processing status. Use this processor for any custom transformation logic. 
<input type="radio"/> Apache Log to JSON	An Amazon Kinesis Data Firehose stream processor that converts input records from Apache Common Log format to JSON.



10. Scroll down. Click on the Use blueprint button.

The screenshot shows a list of AWS Lambda blueprints. The items listed are:

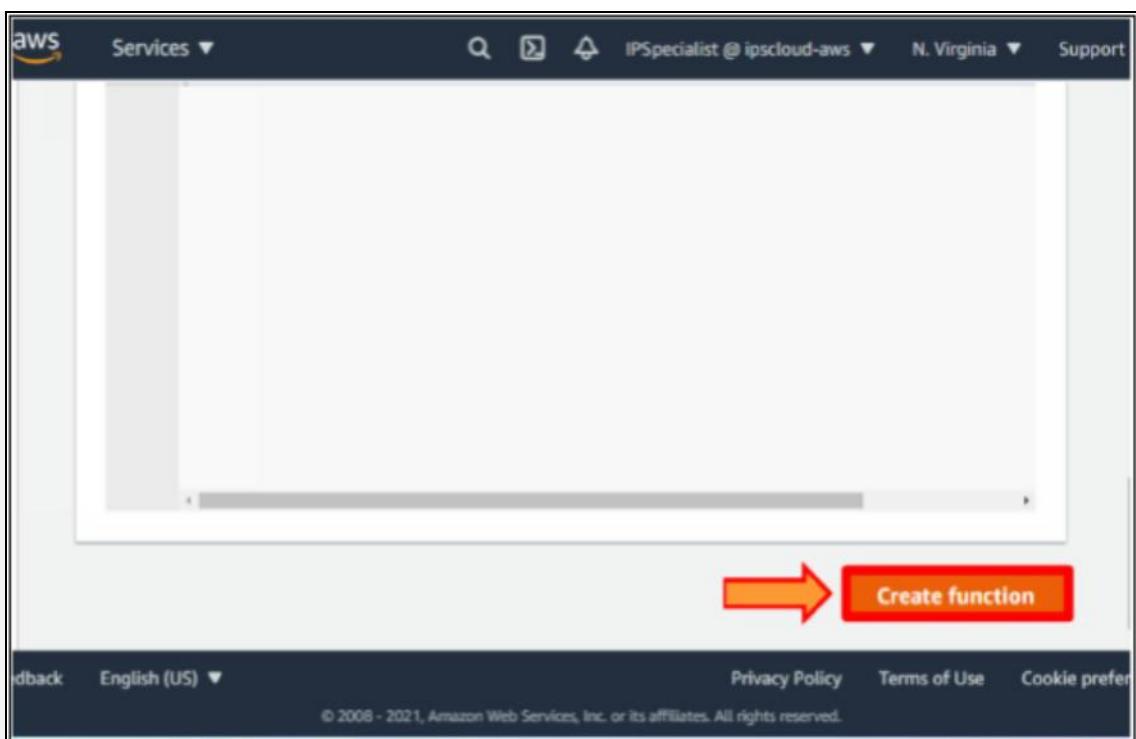
- Apache Log to CSV
- Syslog to JSON
- Syslog to CSV
- Kinesis Data Firehose Process Record Streams as source
- Kinesis Data Firehose Cloudwatch Logs Processor

At the bottom right of the list, there is a red rectangular button labeled "Use blueprint" with a small icon next to it.

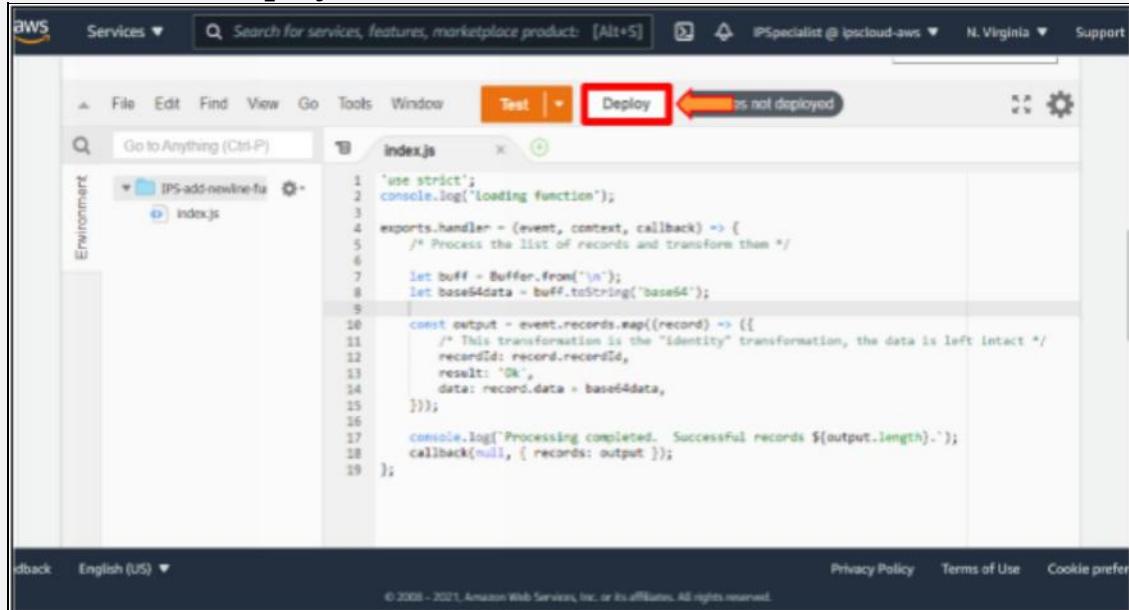
11. Give a name **IPS-add-newline-function**.
12. Select the **Create a new role with basic Lambda permissions**.

The screenshot shows the "Create function" configuration page in the AWS Lambda console. The "Basic information" section is visible. The "Function name" field contains the value "IPS-add-newline-function". The "Execution role" section includes a note about choosing a role or creating a custom one via the IAM console. Three options are shown: "Create a new role with basic Lambda permissions" (selected), "Use an existing role", and "Create a new role from AWS policy templates". A note at the bottom states that role creation might take a few minutes. The bottom navigation bar includes links for Privacy Policy, Terms of Use, and Cookie preferences.

13. Scroll down. Click on the **Create function** button.



14. Now copy and paste the **index.js** code provided in the following GitHub link:
https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter3/index.js
15. Scroll down and paste the **index.js** code into the code editor.
16. Click on the **Deploy** button.



17. Click on the **Test**. Then, click on **Configure test event**.

The screenshot shows the AWS Lambda function editor. The code in `index.js` is:

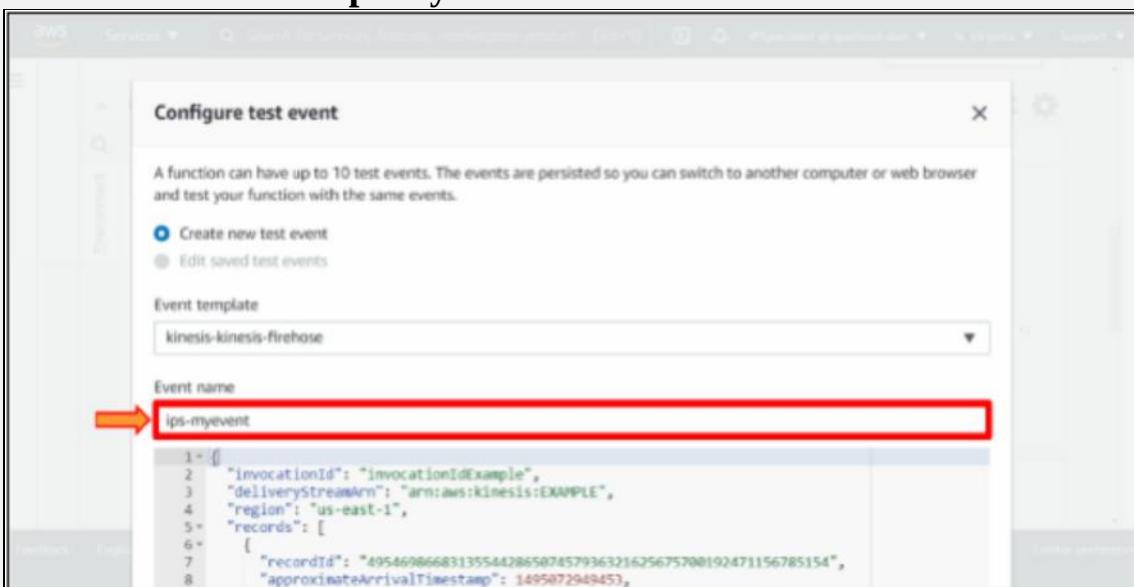
```
'use strict';
console.log('Loading Function');

exports.handler = (event, context, callback) => {
    /* Process the list of records and transform them */
    let buff = Buffer.from('\n');
    let base64data = buff.toString('base64');

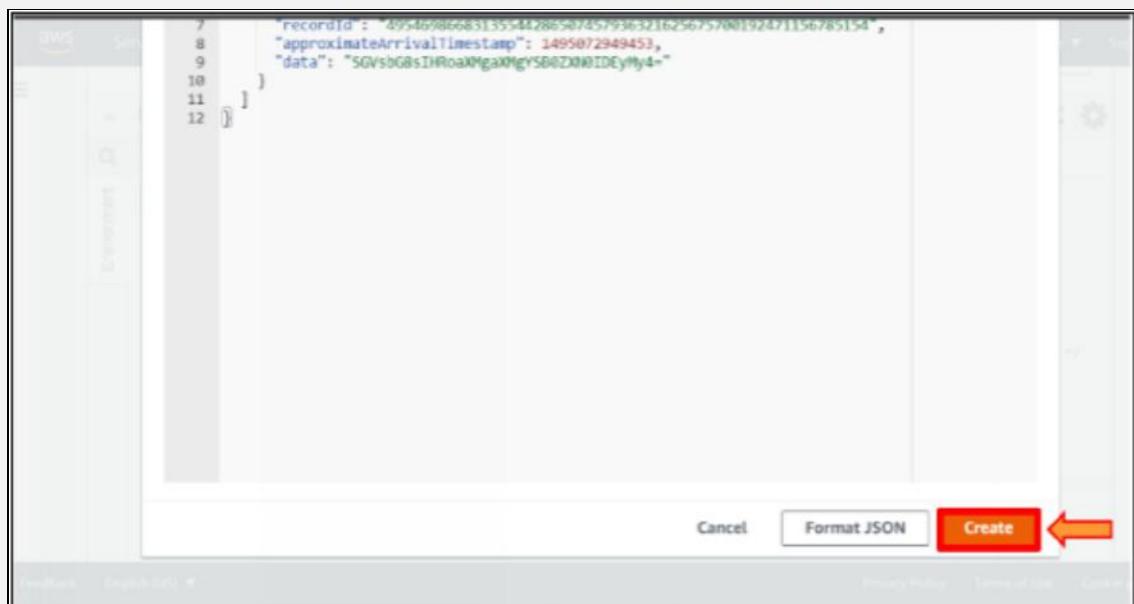
    const output = event.records.map((record) => {
        /* This transformation is the "identity" transformation, the data is left intact */
        recordId: record.recordId,
        result: 'Ok',
        data: record.data + base64data,
    });
    console.log('Processing completed. Successful records ${output.length}.');
    callback(null, { records: output });
};
```

The **Test** tab is selected, indicated by a red arrow pointing to the tab name. A green button labeled **Changes deployed** is visible. A blue button labeled **Configure test event... (Edit)** is also highlighted with a red arrow.

18. Give an event name **ips-myevent**.



19. Scroll down. Click on the **Create** button.



20. Click on the **Test** tab. Then, click on the **Test** button.

The screenshot shows the AWS Lambda function configuration interface. At the top, there are tabs for 'Code', 'Test' (which is highlighted with a red box and arrow), 'Monitor', 'Configuration', 'Aliases', and 'Versions'. Below the tabs, there's a section titled 'Test event' with a 'Test' button (also highlighted with a red box and arrow). A code editor window displays a JSON test event template for an Amazon Kinesis stream processor. The template includes fields like invocationId, deliveryStreamArn, region, and records. At the bottom of the page, there are links for 'Feedback', 'English (US)', and 'Cookie preferences'.

Hence, the execution result is a success.

This screenshot shows the same AWS Lambda configuration page after a successful test execution. A red box highlights the 'Execution result: succeeded (logs)' message, which includes a 'Details' link. The rest of the interface is identical to the previous screenshot, including the tabs at the top and the code editor below.

21. Click on the Configuration tab. Then, click on the Edit button.

The screenshot shows the AWS Lambda configuration page with the 'Configuration' tab selected (highlighted with a red box and arrow). On the left, a sidebar lists 'General configuration', 'Triggers', 'Permissions', 'Destinations', and 'Environment'. The 'General configuration' section on the right contains fields for 'Description' (an Amazon Kinesis Firehose stream processor that accesses the records in the input and returns them with a processing status), 'Memory (MB)' (set to 128), and 'Timeout' (set to 0 min 3 sec). An 'Edit' button (highlighted with a red box and arrow) is located in the top right of this section. The bottom of the page includes standard AWS navigation links for 'Feedback', 'English (US)', and 'Cookie preferences'.

22. Scroll down and change the **time out** to 1 min. Then, click on the **Save** button.

The screenshot shows the AWS Lambda function configuration page. At the top, there's a 'Timeout' section with input fields for minutes (1) and seconds (0). A red box highlights these fields, and a large orange arrow points to the 'Save' button at the bottom right. Below the timeout section, there's an 'Execution role' section with two options: 'Use an existing role' (selected) and 'Create a new role from AWS policy templates'. Under 'Existing role', a dropdown menu shows 'service-role/IPS-add-newline-function-role-k2mjp0xi' and a 'View the IPS-add-newline-function-role-k2mjp0xi role on the IAM console' link. At the bottom of the page, there's a footer with links for 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with copyright information: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

23. Now, go back to the **Create a delivery stream** page.
24. Click on the **Browse** button to select the Lambda function.

The screenshot shows the 'Create a delivery stream' page. In the 'AWS Lambda function' section, there's a text input field labeled 'Choose a Lambda function or enter an ARN' and a dropdown menu labeled 'Choose a ver...'. To the right of the dropdown is a red box highlighting the 'Browse' button. Below these fields, there's a 'Create function' button and a note about the ARN format: 'Format: arn:aws:lambda:[Region]:[AccountId]:function:[FunctionName]'. Under 'Buffer size', there's a text input field with '3' MB selected, and a note: 'The AWS Lambda function has a 6 MB invocation payload quota. Your data can expand in size after it's processed by the AWS Lambda function. A smaller buffer size allows for more room should the data expand after processing.' At the bottom of the page, there's a footer with links for 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with copyright information: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

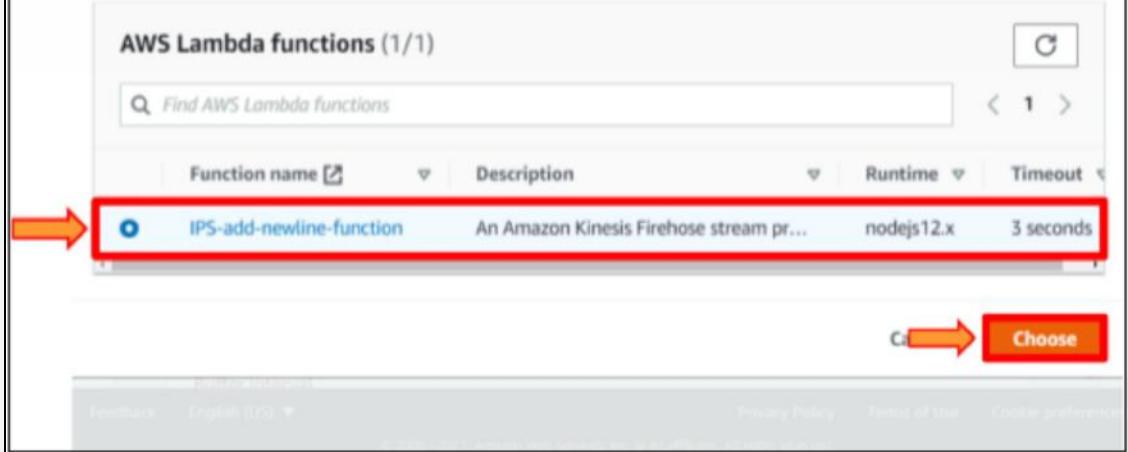
25. Select the **IPS-add-newline-function**. Then, click on the **Choose** button.

The current timeout of the specified AWS Lambda function is 3 seconds. To reduce the risk of the AWS Lambda function timing out before data transformation is complete, increase the timeout to 1 minute or longer in the Advanced settings section of your AWS Lambda configuration. [Go to AWS Lambda configuration](#).

AWS Lambda functions (1/1)

Function name	Description	Runtime	Timeout
IPS-add-newline-function	An Amazon Kinesis Firehose stream pr...	nodejs12.x	3 seconds

Create **Choose**



26. Set the Buffer size to **1 MB**.
27. Set the Buffer interval to **60 sec**.

Format: arn:aws:lambda:[Region]:[AccountId]:function:[FunctionName]

Buffer size
The AWS Lambda function has a 6 MB invocation payload quota. Your data can expand in size after it's processed by the AWS Lambda function. A smaller buffer size allows for more room should the data expand after processing.

1 MB

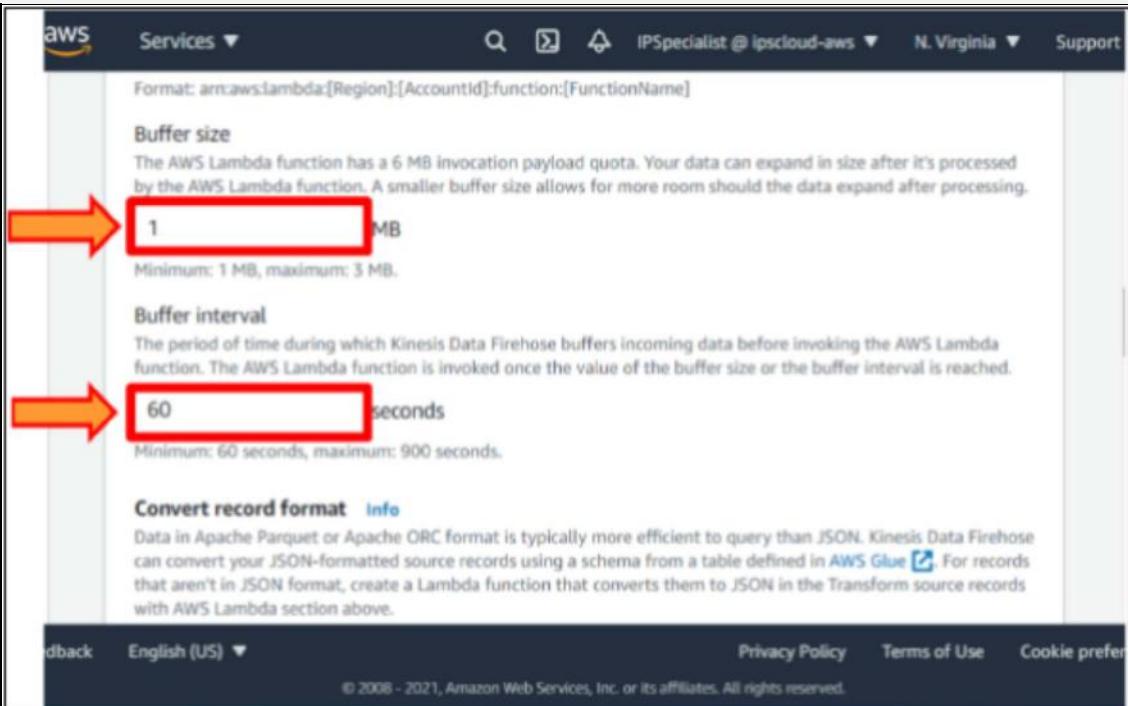
Minimum: 1 MB, maximum: 3 MB.

Buffer interval
The period of time during which Kinesis Data Firehose buffers incoming data before invoking the AWS Lambda function. The AWS Lambda function is invoked once the value of the buffer size or the buffer interval is reached.

60 seconds

Minimum: 60 seconds, maximum: 900 seconds.

Convert record format [Info](#)
Data in Apache Parquet or Apache ORC format is typically more efficient to query than JSON. Kinesis Data Firehose can convert your JSON-formatted source records using a schema from a table defined in [AWS Glue](#). For records that aren't in JSON format, create a Lambda function that converts them to JSON in the Transform source records with AWS Lambda section above.



28. Click on the **Create** button to create an S3 bucket.

Sales Support Services ▾

Search icon

IPSpecialist @ ipscloud-aws ▾ N. Virginia ▾ Support

Destination settings Info

Specify the destination settings for your delivery stream.

S3 bucket

Choose a bucket or enter a bucket URI

Browse

Create 

Format: S3://bucket

Dynamic partitioning Info

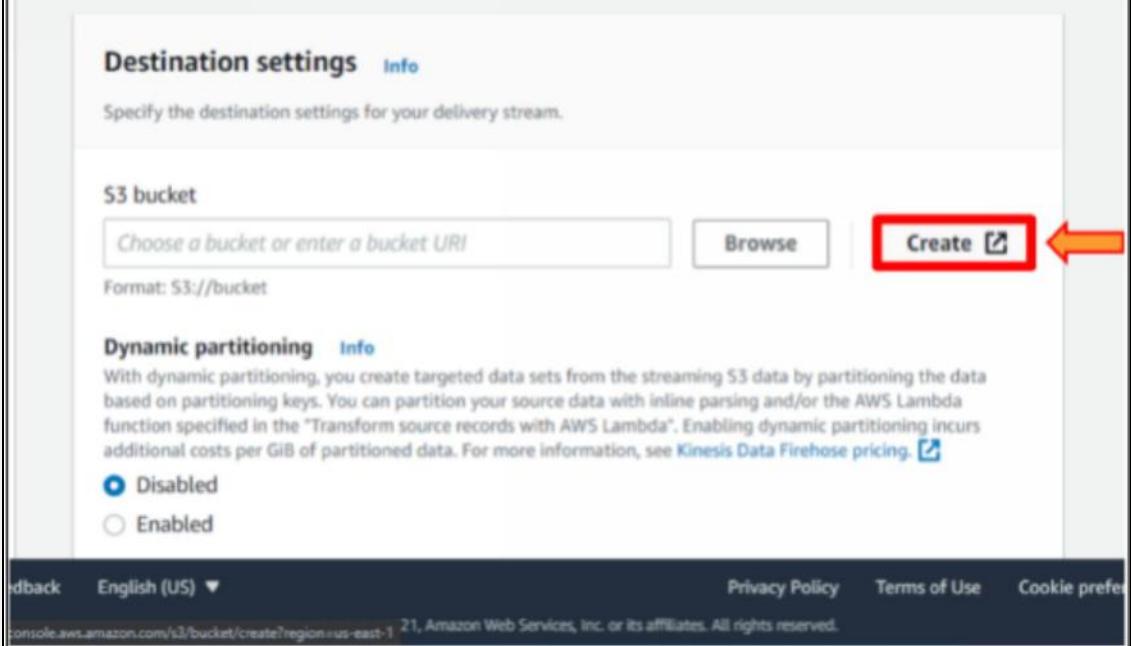
With dynamic partitioning, you create targeted data sets from the streaming S3 data by partitioning the data based on partitioning keys. You can partition your source data with inline parsing and/or the AWS Lambda function specified in the "Transform source records with AWS Lambda". Enabling dynamic partitioning incurs additional costs per GiB of partitioned data. For more information, see Kinesis Data Firehose pricing. 

Disabled

Enabled

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

console.aws.amazon.com/s3/bucket/create?region=us-east-1 21, Amazon Web Services, Inc. or its affiliates. All rights reserved.



29. Give the S3 bucket name **ips-my-user-data-output-bucket**.

Sales Support Services ▾

Search icon

IPSpecialist @ ipscloud-aws ▾ Global ▾ Support

General configuration

Bucket name 

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#) 

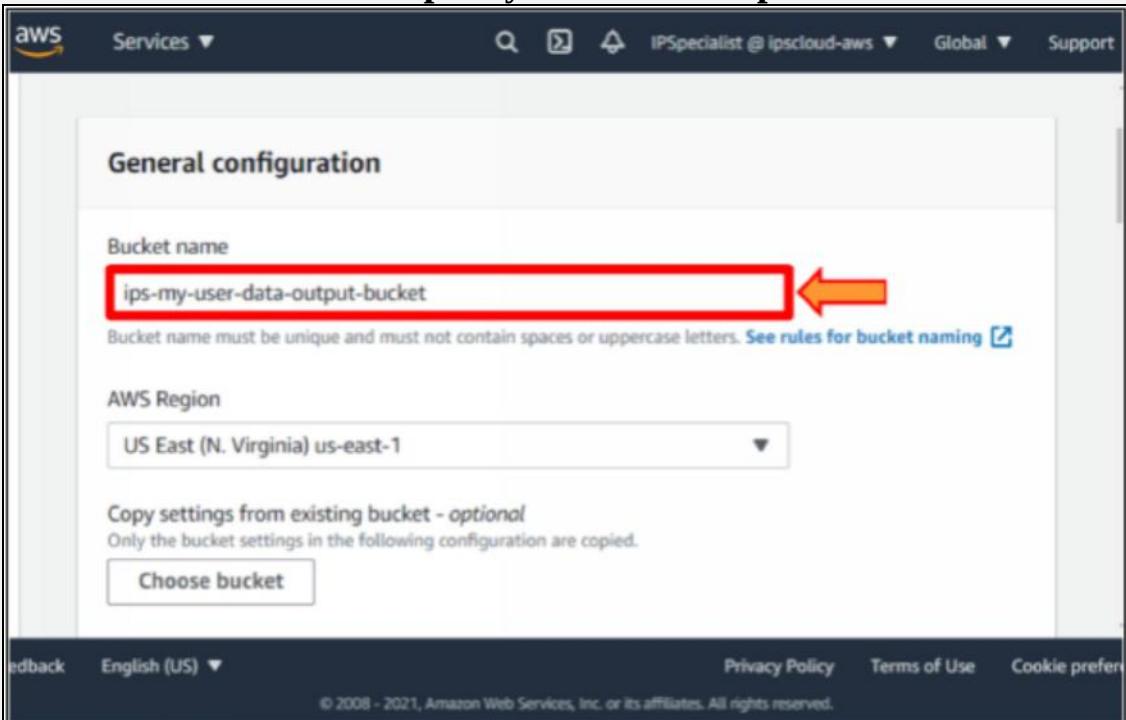
AWS Region

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

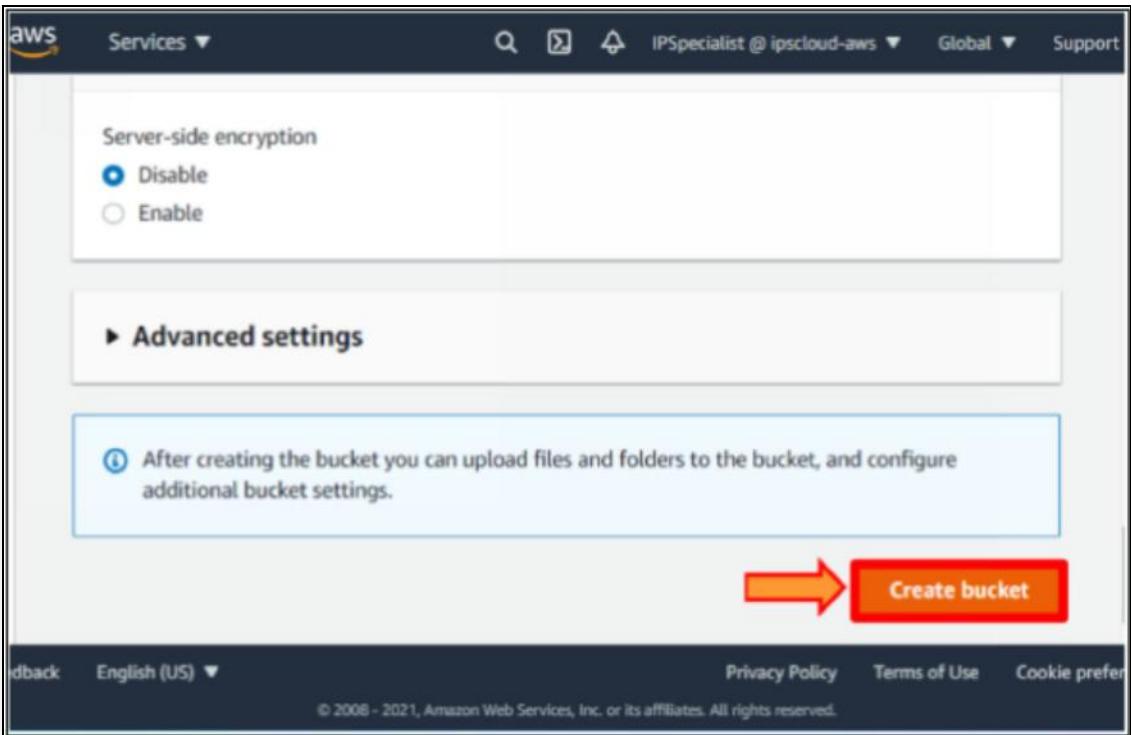
Choose bucket

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

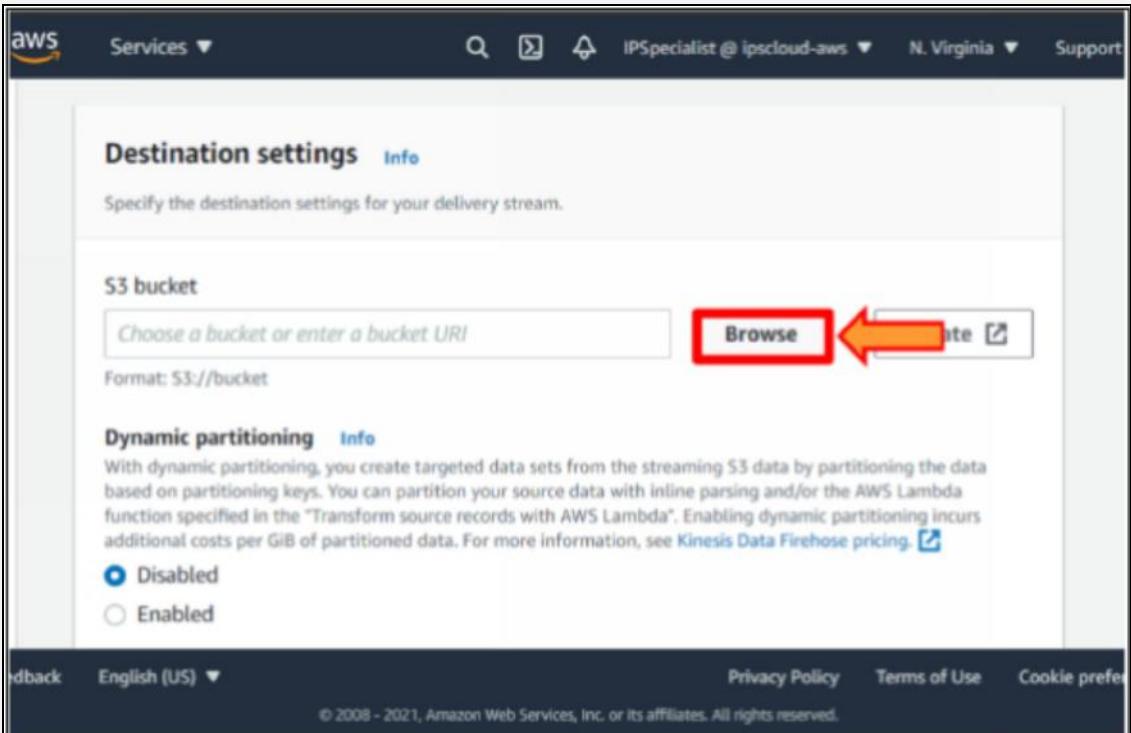
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



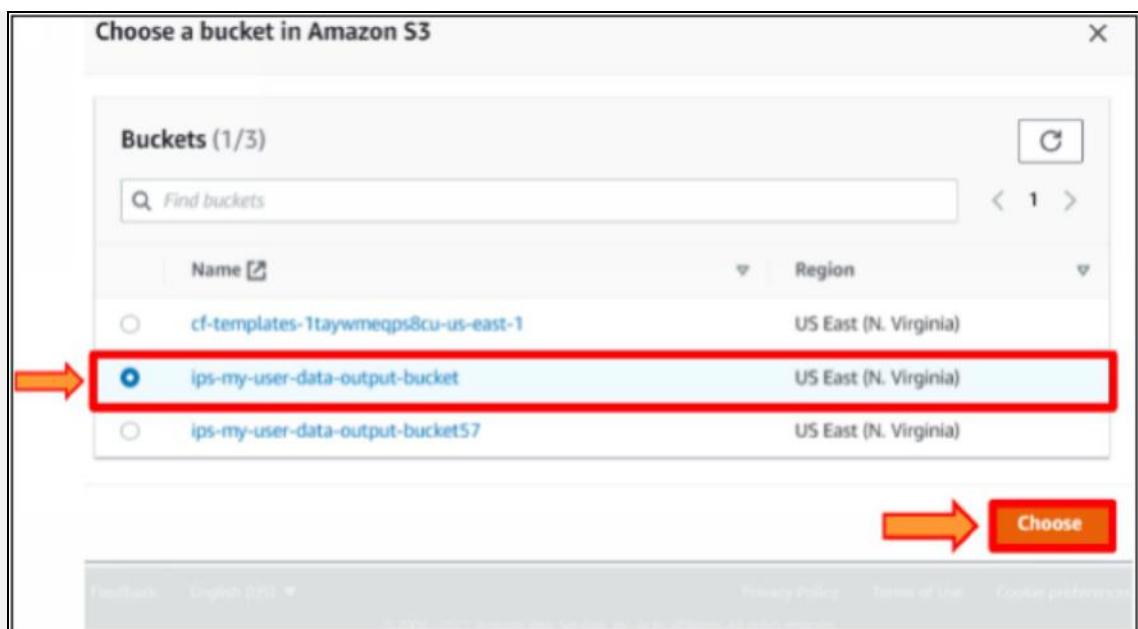
30. Click on **Create bucket**.



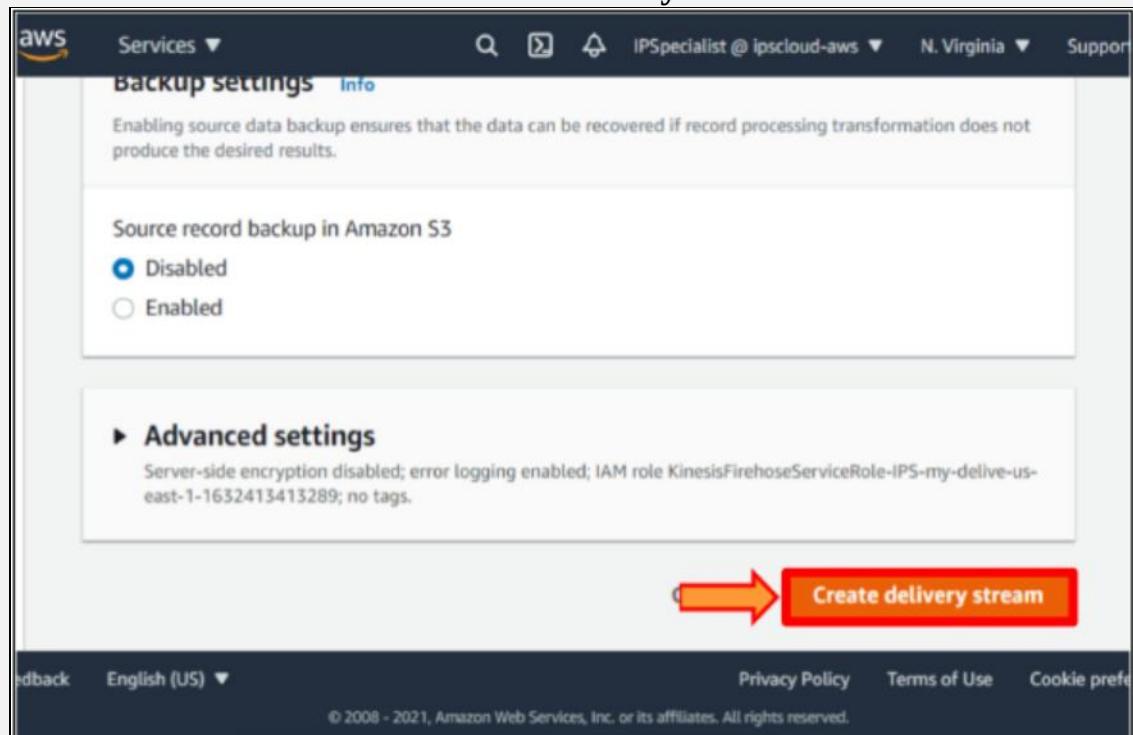
31. Click on the **Browse** button.



32. Select the **ips-my-user-data-output-bucket**. Then, click on the **Choose** button.



33. Scroll down. Click on the **Create delivery stream** button.



Hence, the delivery stream has been created.

34. Now go to the **Configure destination for IPS-data-transformation-application**.

35. Click on the **Browse** button to select the delivery stream.

Sales

Services ▾

Redshift, Amazon Elasticsearch Service, or Splunk.

AWS Lambda function
Use an AWS Lambda function to process streaming data.

Delivery stream

Choose a delivery stream or enter a delivery stream ARN **Browse** 

Format: arn:aws:firehose:\$\{Region\}:\\$\{AccountId\}:deliverystream/\$\{DeliveryStreamName\}

Access permissions for writing output stream
Create or choose IAM role with the required permissions. [Learn more](#)

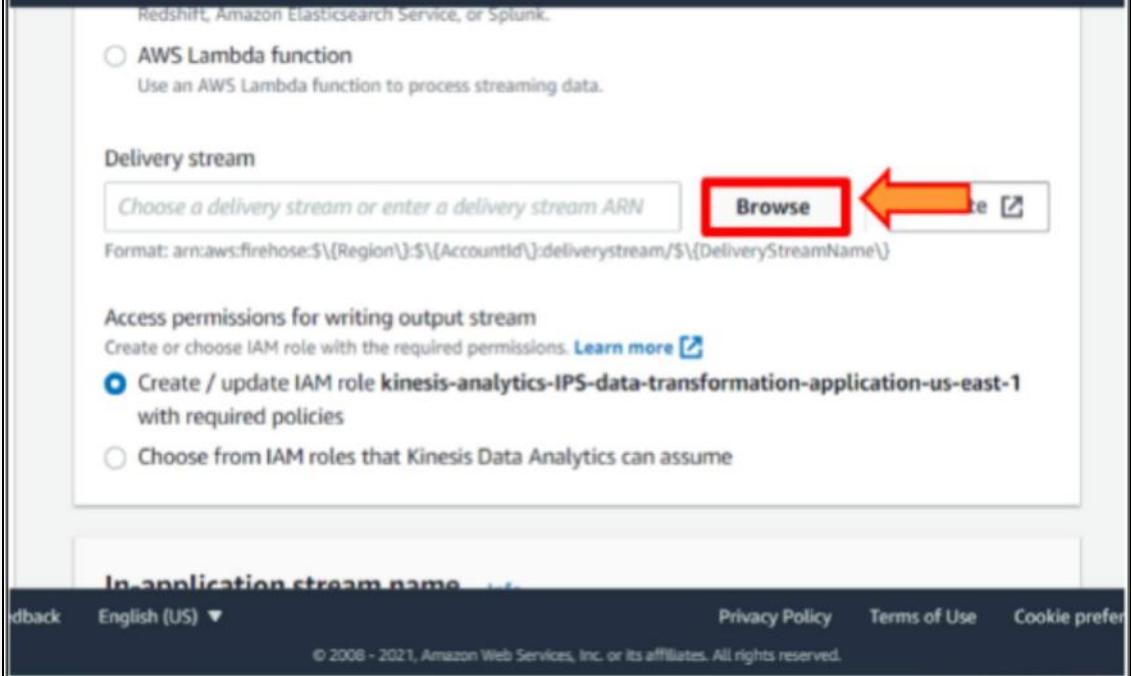
Create / update IAM role **kinesis-analytics-IPS-data-transformation-application-us-east-1** with required policies

Choose from IAM roles that Kinesis Data Analytics can assume

In-application stream name 

Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



36. Select the **ips-my-delivery-stream**. Then, click on the **Choose** button.

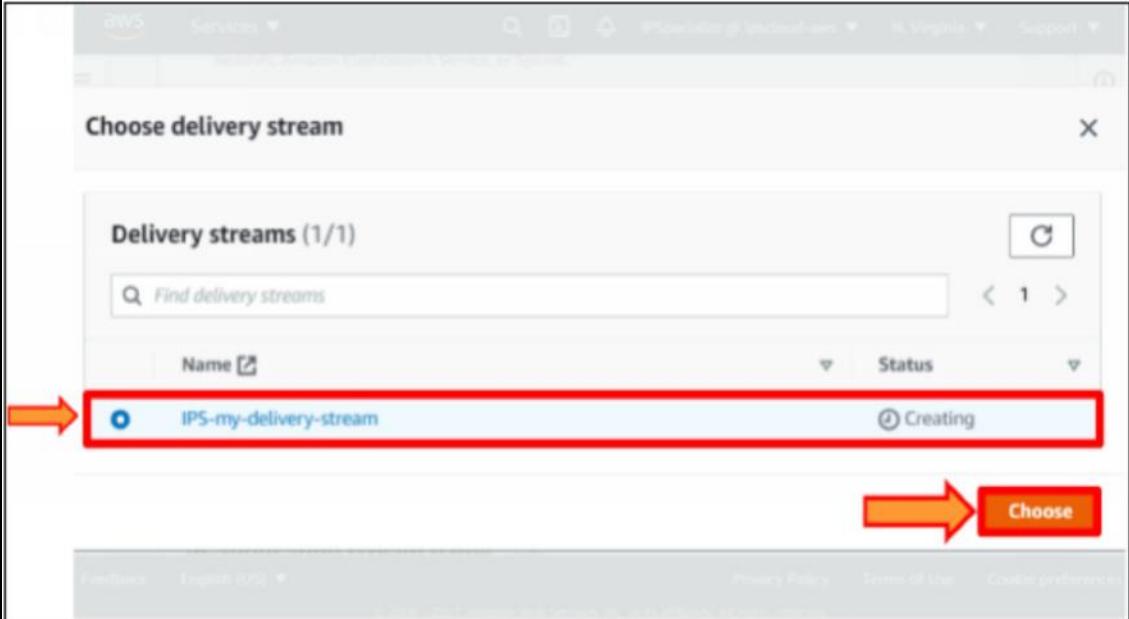
Choose delivery stream

Delivery streams (1/1)

Name	Status
<input checked="" type="radio"/> IPS-my-delivery-stream	 Creating

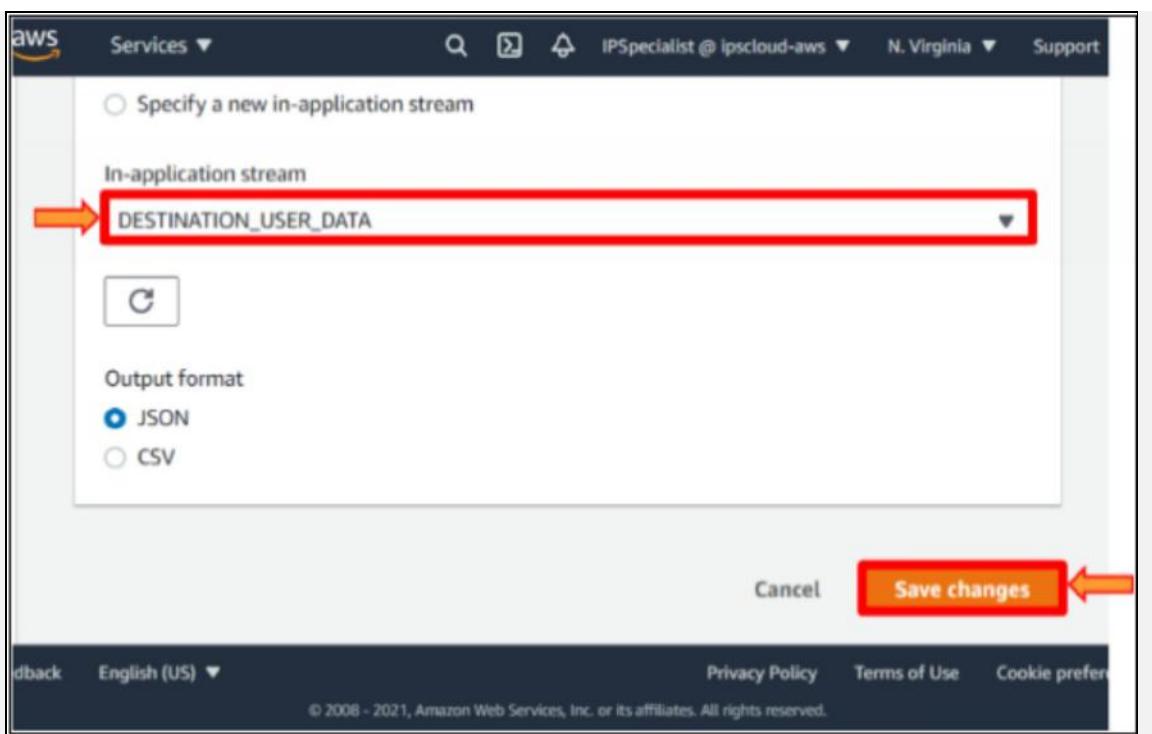
 

Privacy Policy Terms of Use Cookie preferences

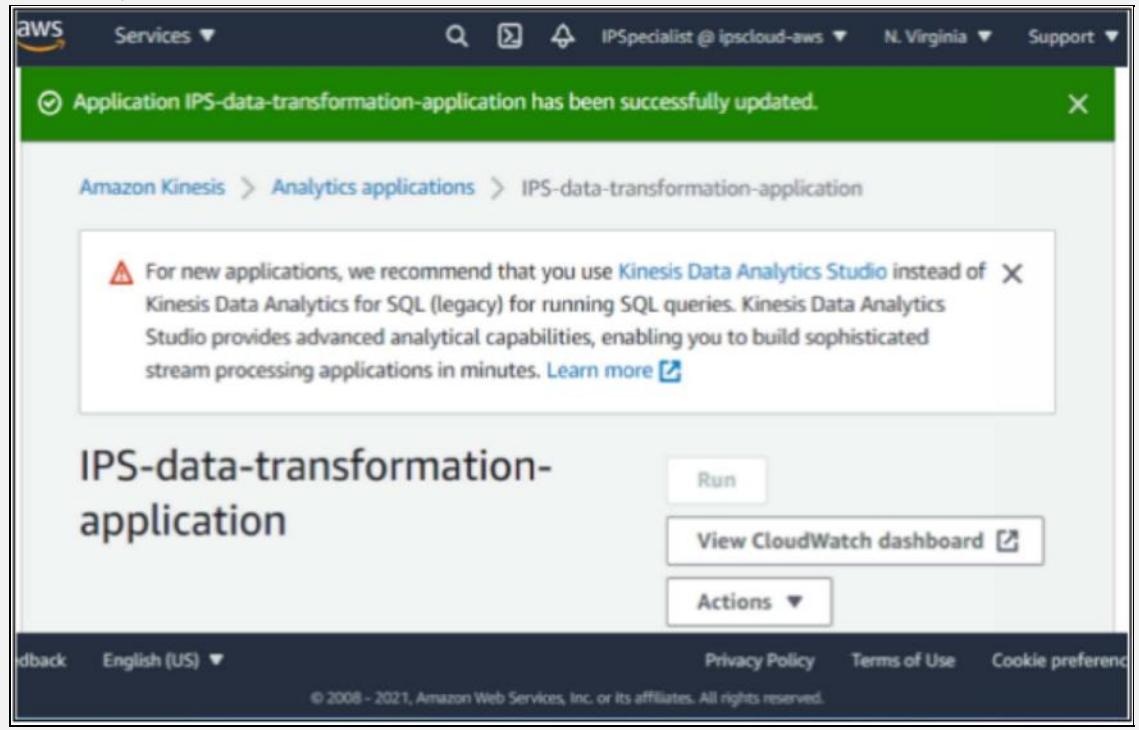


37. Scroll down. Select the **DESTINATION_USER_DATA**.

38. Then, click on the **Save changes** button.



39. Hence, the Kinesis Data Firehose is created.



Step 5: View Collected Data

1. Click on Services.

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with the AWS logo, a 'Services' dropdown menu (highlighted with a red arrow), search, notifications, and account information ('IPSpecialist @ ipscloud-aws', 'N. Virginia', 'Support'). Below the navigation bar is the main title 'AWS Management Console'. On the left, there is a sidebar titled 'AWS services' with sections for 'Recently visited services' (S3, Kinesis, Lambda, CloudFormation) and 'All services'. At the bottom of the sidebar are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the very bottom states '© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

2. Select the S3 from Storage.

The screenshot shows the 'All services' page in the AWS Management Console. On the left, there is a sidebar with 'Favorites' (empty), 'Recently visited' (Console, Home, S3, Lambda, Kinesis, CloudFormation), and a 'Storage' section with 'S3' highlighted by a red box and a red arrow pointing to it. The main area is titled 'All services' and lists various AWS services in a grid format. The 'Storage' column includes EFS, FSx, S3 Glacier, Storage Gateway, and AWS Backup. The 'Amazon' column includes Transcribe, Translate, AWS Auto Scaling, CloudWatch Metrics, CloudTrail, Config, OpsWorks, and Service Quotas. The 'Application Integration' column includes Step Functions, AWS AppFlow, DeepComposers, DeepLens, AWS DeepRacer, AWS Panorama, and Simple Notification Service.

3. Click on IPS-my-user-data-output-bucket.

Sales Services ▾ Search for services, features, marketplace products, and docs [Alt+S] IPSpecialist @ ipscloud-aws Global ▾

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Lock Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feedback English (US) ▾ Privacy Policy Terms of Use

Buckets (2) Info

Buckets are containers for data stored in S3. [Learn more](#)

Copy ARN Empty Delete Create bucket

Find buckets by name

Name	AWS Region	Access	Creation date
cf-templates-1taywmeqps8cu-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	September 23, 2021, 04:52:18 (UTC-07:00)
ips-my-delivery-stream	US East (N. Virginia) us-east-1	Objects can be public	September 23, 2021, 06:17:35 (UTC-07:00)

4. Click on the **Folder**. Finally, you can see a file uploaded onto the S3 bucket.

Sales Services ▾ Search for services, features, marketplace products, and docs [Alt+S] IPSpecialist @ ipscloud-aws Global ▾

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Lock Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

WS Marketplace for S3

Feedback English (US) ▾ Privacy Policy Terms of Use

Amazon S3 > ips-my-delivery-stream > 2021/ > 09/ > 23/ > 13/ Copy S3 URI

13/

Objects Properties

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions

Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
IPS-my-delivery-stream-1-2021-09-23-15-19-17-6b2498a9-a884-406e-bb24-4963707125cd	-	September 23, 2021, 06:26:46 (UTC-07:00)	631.7 KB	Standard

5. Click on the **Refresh** button; you will see more data is uploaded.

Sales Services ▾ Search for services, features, marketplace products, and docs [Alt+S] IPSpecialist @ ipscloud-aws Global ▾

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Lock Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

WS Marketplace for S3

Feedback English (US) ▾ Privacy Policy Terms of Use

Objects (8)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI Copy URL Download Open Delete Actions

Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
IPS-my-delivery-stream-1-2021-09-23-16-48-55-f60bb808-9b40-4b75-9e78-66bedc536a81	-	September 23, 2021, 09:53:58 (UTC-07:00)	43.8 KB	Standard
IPS-my-delivery-stream-1-2021-09-23-16-52-59-20571293-1151-4e30-9ef5-e81b9af1b4d3	-	September 23, 2021, 09:58:01 (UTC-07:00)	45.6 KB	Standard
IPS-my-delivery-stream-1-2021-09-23-16-57-03-e560a05f-0b8d-4355-9163-80843158860b	-	September 23, 2021, 10:02:05 (UTC-07:00)	43.9 KB	Standard

Hence, you have successfully streamed data into Kinesis. You have also queried and transformed it with Kinesis Data Analytics. Then, you have to set up a destination using a Kinesis Data Firehose to output the data onto the S3 bucket.

Lab 05: Performing Real-Time Data Analysis with Kinesis

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	Kinesis, S3, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

Kinesis Data Firehose

Kinesis Data Firehose is a solution for streaming ETL. It is the most convenient method for loading streaming data into data storage and analytics tools. It can gather, convert, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, allowing you to do near-real-time analytics with your existing business intelligence tools and dashboards. It is a fully managed service that scales automatically to meet your data flow and does not require any ongoing management. It may also batch, compress, and encrypt data before loading it, reducing storage requirements at the destination and enhancing security.

Kinesis Data Analytics

Amazon Kinesis Data Analytics is the most straightforward method for converting and analyzing real-time streaming data using Apache Flink. Apache Flink is an open-source data stream processing framework and engine. Amazon Kinesis Data Analytics makes it easier to create, administer, and integrate Apache Flink applications with other AWS services.

Amazon Kinesis Data Analytics covers everything required to run streaming applications continuously and scales automatically to meet the volume and throughput of your incoming data. There are no servers to operate with Amazon Kinesis Data Analytics, no minimum charge or setup cost, and you pay for the resources your streaming applications utilize.

Amazon Simple Storage Service S3

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 provides a simple web service interface for storing and retrieving any quantity of data from any place at any time. Using this service, you may quickly create projects that integrate cloud-native storage because Amazon S3 is easily customizable. You only pay for what you use; you can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. Build a simple FTP system or a complex web application like the Amazon.com retail website; read the same piece of data a million times or only for emergency disaster recovery; store whatever type and amount of data you desire.

Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a cloud computing service that gives the durability of computing power. It is intended to make web-scale computing more accessible to IT engineers.

Amazon EC2 offers “compute” in the cloud, much as Amazon Simple Storage Service (Amazon S3) enables “storage” in the cloud. The easy web service interface of Amazon EC2 allows you to obtain and configure capacity quickly. It lets you control your computer resources entirely and enable you to run on Amazon’s tried-and-true computing infrastructure. The time to buy and boot new Amazon EC2 server instances is reduced to minutes, allowing you to scale up quickly and down capacity as your computing needs change. Amazon EC2 revolutionizes computing economics by enabling you to pay only for the resources you utilize.

Case Study Enterprise Biopharmaceutical – Neumora Therapeutics

Business Challenge

[Neumora Therapeutics](#) has thousands of users interacting with the organization application. The organization gives [Neumora Therapeutics](#) a task of analyzing the data in real-time. The organization also wants you to load real-time data into a cloud storage destination as a backup copy.

Proposed Solution

The solution is you use AWS services to automate your task and analyze the stored data in real-time within minutes. For storing the data in real-time, you use Amazon Simple Storage S3. You use Amazon Kinesis Data Analytics to run SQL queries to analyze the stored data in real-time. To store real-time analyzed data at any storage destination, you use Amazon Kinesis Firehose.

Before starting the lab, launch an EC2 instance and install the related python packages used in the Python script provided in the following GitHub link:

[https://github.com/12920/IPSpecialist01/blob/main/Course_AWS_Certified_Machine_Learning-master%20\(1\).zip](https://github.com/12920/IPSpecialist01/blob/main/Course_AWS_Certified_Machine_Learning-master%20(1).zip)

The following Python script is used to generate the dummy data in this lab (Lab 03-02).

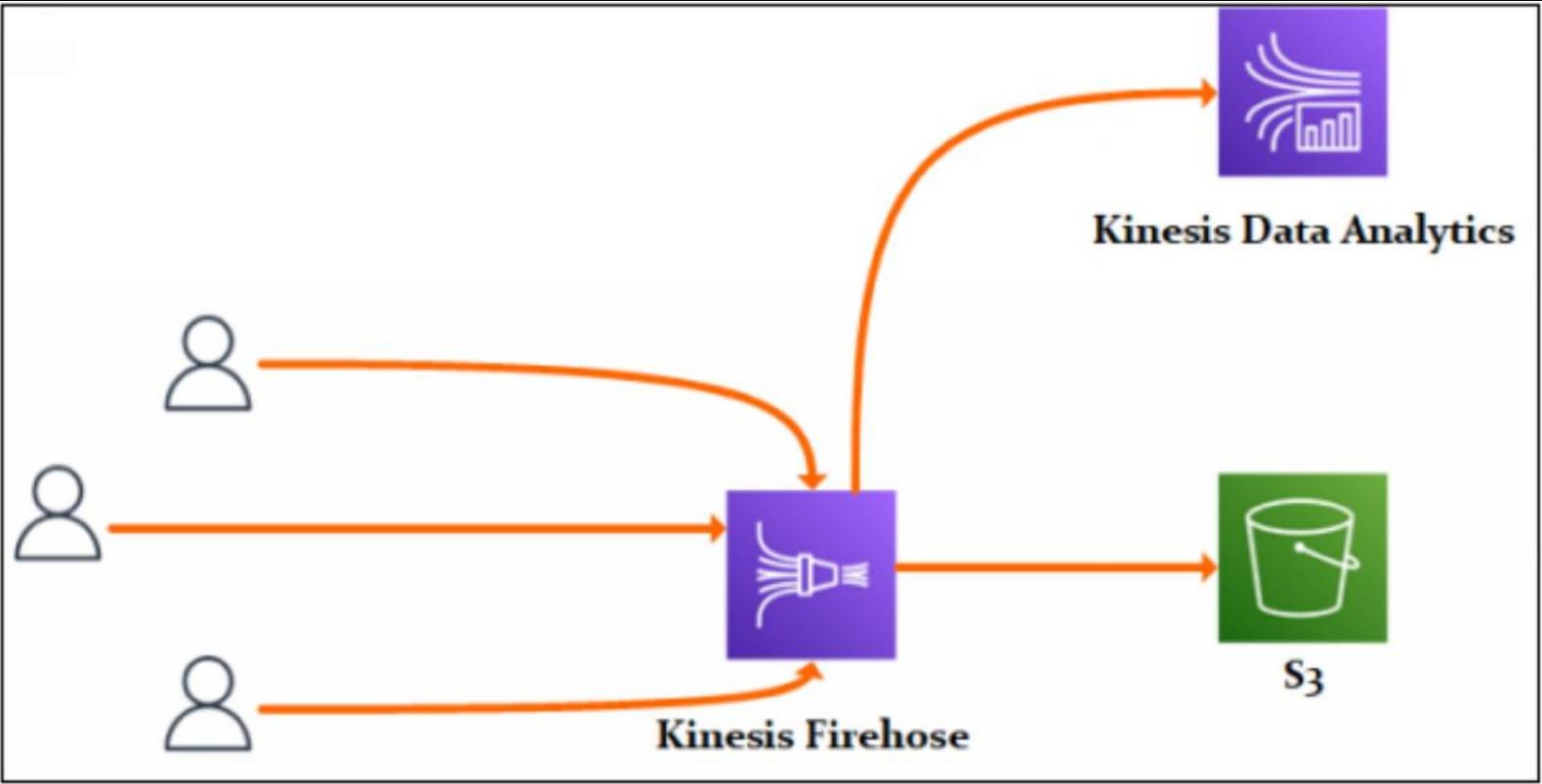
```
#!/usr/bin/env python
import boto3
from faker import Faker
import random
import time
import json

DeliveryStreamName = 'captains-kfh'
client = boto3.client('firehose')
fake = Faker()

captains = [
    "Jean-Luc Picard",
    "James T. Kirk",
    "Han Solo",
    "Kathryn Janeway",
    "Malcolm Reynolds",
    "William Adama",
    "Turanga Leela",
    "Jacob Keyes",
    "Wilhuff Tarkin",
    "Christopher Pike",
    "David Bowman",
    "The Doctor",
    "John Robinson",
    "Khan Noonien Singh"
];
record = {}
while True:

    record['user'] = fake.name();
    if random.randint(1,100) < 5:
        record['favoritecaptain'] = "Neil Armstrong":
```

Lab Diagram



Implementation Steps

1. Create Kinesis Data Firehose.
2. Generate Data to the Stream.
3. Create Kinesis Data Analytics.

Solution

Step 1: Create Kinesis Data Firehose

1. Log in to the **AWS Console**.
2. Click on **Services**.

AWS Management Console

AWS services

Recently visited services

- Kinesis
- EC2
- S3

All services

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Select **Kinesis** from **Analytics**.

Sales Services ▾ Q Search for services, features, marketplace prod [Alt+S] cloud_user @ 4830-5410-9269 N. Virginia Support

Favorites

Add favorites by clicking on the star next to the service name.

Recently visited

- Console Home
- Kinesis
- S3
- EC2

All services

Category	Service
DynamoDB	AWS AppConfig
ElastiCache	Trusted Advisor
Neptune	Control Tower
Amazon QLDB	AWS License Manager
Amazon	AWS Well-Architected Tool
DocumentDB	Personal Health Dashboard
Amazon Keyspaces	AWS Chatbot
Amazon Timestream	Launch Wizard
Amazon MemoryDB for Redis	AWS Compute Optimizer
	Resource Groups & Tag Editor
Migration & Transfer	Kinesis
AWS Migration Hub	QuickSight
SageMaker	Data Pipeline

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

4. Select **Kinesis Data Firehose**. Then, click on the **Create delivery stream** button.

Sales Services ▾ Q Search for services, features, marketplace prod [Alt+S] cloud_user @ 4830-5410-9269 N. Virginia Support

Get started

Kinesis Data Streams
Collect streaming data with a data stream.

Kinesis Data Firehose
Process and deliver streaming data with data delivery stream. 

Kinesis Data Analytics
Analyze streaming data with data analytics application.

Create delivery stream 

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Select **Direct PUT in Source**.

Sales Support

Services ▾

Cloud User @ 4830-5410-9269 ▾ N. Virginia ▾ Support

Choose source and destination

Specify the source and the destination for your delivery stream. You cannot change the source and destination of your delivery stream once it has been created.

Source [Info](#)

Direct PUT

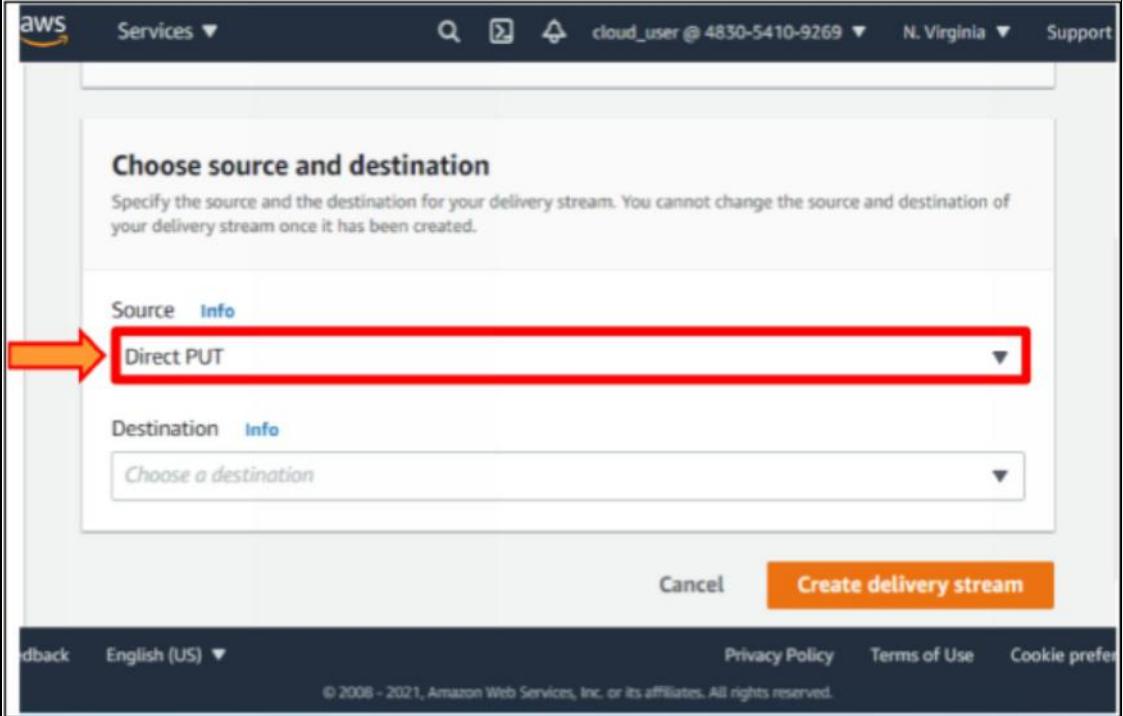
Destination [Info](#)

Choose a destination

Cancel **Create delivery stream**

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



6. Select Amazon S3 in Destination.

Sales Support

Services ▾

Cloud User @ 4830-5410-9269 ▾ N. Virginia ▾ Support

▶ Amazon Kinesis Data Firehose: How it works

Choose source and destination

Specify the source and the destination for your delivery stream. You cannot change the source and destination of your delivery stream once it has been created.

Source [Info](#)

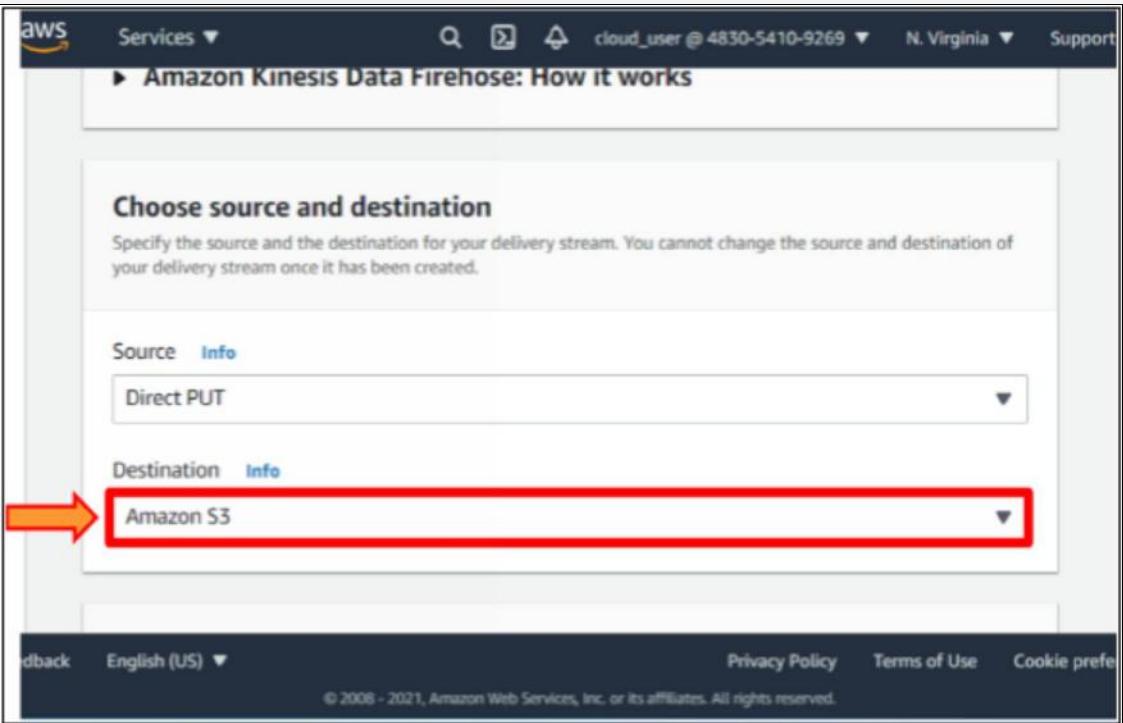
Direct PUT

Destination [Info](#)

Amazon S3

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



7. Name the delivery stream **captains-kfh**.

The screenshot shows the 'Delivery stream name' section of the AWS Kinesis Data Firehose configuration page. A red arrow points to the input field where 'captains-kfh' has been typed. Below the input field, a note states: 'Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.' The rest of the page includes sections for 'Transform and convert records - optional' and a footer with navigation links.

8. Scroll down. Click on the **Create** button to create an S3 bucket.

The screenshot shows the 'Destination settings' section of the AWS S3 console. A red arrow points to the 'Create' button, which is highlighted with a red box. The 'S3 bucket' section contains a text input field with 'Choose a bucket or enter a bucket URI' placeholder text. The 'Format: S3://bucket' note is visible below the input field. The 'Dynamic partitioning' section includes a note about targeted data sets and two radio button options: 'Disabled' (selected) and 'Enabled'. The rest of the page includes a footer with navigation links.

9. Give a bucket name **ips-s3-bucket**.

AWS Services ▾

Cloud User @ 4830-5410-9269 ▾ Global ▾ Support

Create bucket Info

Buckets are containers for data stored in S3. [Learn more](#) 

General configuration

Bucket name  

ips-s3-bucket

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#) 

AWS Region

US East (N. Virginia) us-east-1 

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

10. Scroll down. Click on the **Create bucket** button.

Server-side encryption

Disable

Enable

▶ Advanced settings

 After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.



11. Go back to the **Create Delivery Stream** page.

12. Click on the **Browse** button.

S3 bucket

Choose a bucket or enter a bucket URI

Browse

Format: S3://bucket

Dynamic partitioning

With dynamic partitioning, you create targeted data sets from the streaming S3 data by partitioning the data based on partitioning keys. You can partition your source data with inline parsing and/or the AWS Lambda function specified in the "Transform source records with AWS Lambda". Enabling dynamic partitioning incurs additional costs per GiB of partitioned data. For more information, see [Kinesis Data Firehose pricing](#).

Disabled

Enabled

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

13. Select **ips-s3-bucket**. Then, click on the **Choose** button.

Choose a bucket in Amazon S3

Buckets (1/2)

Find buckets

Name Region

cfst-3254-7248ef1e09b4577ea6f22d-cloudtrailbucket-1p8kkkhifayml US East (N. Virginia)

ips-s3-bucket US East (N. Virginia)

Choose

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

14. Scroll down. Click on the **Create delivery stream** button.

AWS Services ▾

cloud_user @ 4830-5410-9269 ▾ N. Virginia ▾ Support

Enter a prefix

▶ Buffer hints, compression and encryption
The fields below are pre-populated with the recommended default values for S3. Pricing may vary depending on storage and request costs.

▶ Advanced settings
Server-side encryption disabled; error logging enabled; IAM role KinesisFirehoseServiceRole-captains-kfh-us-east-1-1632494377850; no tags.

 Create delivery stream

dback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2: Generate Data to the Stream

1. Access the Ec2 instance via a putty.

```
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[cloud_user@ip-10-0-1-7 ~]$
```

2. Type the `vi send_captains_to_cloud.py` command to view the Python script in the shell.

```
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
```

3. Type `:q`, then press **Enter** to quit from the vi editor.

```
#!/usr/bin/env python
import boto3
from faker import Faker
import random
import time
import json

DeliveryStreamName = 'captains-kfh'
client = boto3.client('firehose')
fake = Faker()

captains = [
    "Jean-Luc Picard",
    "James T. Kirk",
    "Han Solo",
    "Kathryn Janeway",
    "Malcolm Reynolds",
    "William Adama",
    "Turanga Leela",
    "Jacob Keyes",
    "Wilhuff Tarkin",
    "Christopher Pike",
    "David Bowman",
    "The Doctor",
    "John Robinson",
    "Khan Noonien Singh"
];

record = {}
```

- Type the `python3 send_captains_to_cloud.py` command to run the python script.

```
[cloud_user@ip-10-0-1-7 ~]$ python3 send_captains_to_cloud.py
```

```

Record: {'user': 'Sheryl Brown', 'favoritecaptain': 'Neil Armstrong', 'rating': 8888, 'timestamp': 1632495155.879136}
Record: {'user': 'William Edwards', 'favoritecaptain': 'William Adama', 'rating': 57, 'timestamp': 1632495155.9417439}
}
Record: {'user': 'Jessica Hill', 'favoritecaptain': 'Kathryn Janeway', 'rating': 486, 'timestamp': 1632495155.9632537}
}
Record: {'user': 'David Marshall', 'favoritecaptain': 'Han Solo', 'rating': 818, 'timestamp': 1632495155.9840274}
Record: {'user': 'Cynthia Carlson', 'favoritecaptain': 'William Adama', 'rating': 559, 'timestamp': 1632495156.0061295}
}
Record: {'user': 'Carolyn McCullough', 'favoritecaptain': 'James T. Kirk', 'rating': 667, 'timestamp': 1632495156.028292}
}
Record: {'user': 'Brianna Miller', 'favoritecaptain': 'Kathryn Janeway', 'rating': 635, 'timestamp': 1632495156.042191}
}
Record: {'user': 'Stacey Bradshaw', 'favoritecaptain': 'Turanga Leela', 'rating': 763, 'timestamp': 1632495156.065037}
}
Record: {'user': 'Andre Smith', 'favoritecaptain': 'Kathryn Janeway', 'rating': 139, 'timestamp': 1632495156.0791755}
Record: {'user': 'Mary Welch', 'favoritecaptain': 'Neil Armstrong', 'rating': 7988, 'timestamp': 1632495156.0930712}
Record: {'user': 'Ashlee Gould', 'favoritecaptain': 'Kathryn Janeway', 'rating': 694, 'timestamp': 1632495156.106503}
Record: {'user': 'William Kaufman', 'favoritecaptain': 'Kathryn Janeway', 'rating': 396, 'timestamp': 1632495156.1271433}
}
Record: {'user': 'George Haynes', 'favoritecaptain': 'Jean-Luc Picard', 'rating': 845, 'timestamp': 1632495156.1407242}
}
Record: {'user': 'Shelly Cox', 'favoritecaptain': 'Han Solo', 'rating': 430, 'timestamp': 1632495156.1612885}
Record: {'user': 'Chad Lyons', 'favoritecaptain': 'William Adama', 'rating': 107, 'timestamp': 1632495156.1749692}
Record: {'user': 'Pam Archer', 'favoritecaptain': 'Han Solo', 'rating': 756, 'timestamp': 1632495156.1893845}
Record: {'user': 'Michael Harrison', 'favoritecaptain': 'James T. Kirk', 'rating': 28, 'timestamp': 1632495156.2131677}
}
Record: {'user': 'Nicholas Hopkins', 'favoritecaptain': 'The Doctor', 'rating': 983, 'timestamp': 1632495156.2267468}
Record: {'user': 'Rebecca Morgan', 'favoritecaptain': 'William Adama', 'rating': 991, 'timestamp': 1632495156.2482626}
}

```

5. Go back to the Kinesis Data Firehose dashboard.

6. Click on **captain-kfh**.

The screenshot shows the Amazon Kinesis Data Firehose console. On the left, there's a sidebar with 'Amazon Kinesis' at the top, followed by 'Dashboard', 'Data streams', 'Delivery streams' (which is highlighted in red), 'Analytics applications', and 'Resources'. Under 'Resources', there's a link 'What's new'. At the bottom of the sidebar are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. The main content area is titled 'Delivery streams (1)'. It features a search bar with 'Find delivery streams' and a table with one row. The table columns are 'Name', 'Status', 'Creation time', 'Source', and 'Data transformation'. The single row shows 'captains-kfh' as the name, 'Active' as the status, 'September 12, 2023' as the creation time, 'Direct PUT' as the source, and 'Not enabled' as the data transformation. A red arrow points to the 'captains-kfh' entry in the table.

7. Click on the **Monitoring** tab.

Amazon Kinesis

Source: Direct PUT

ARN: arn:aws:firehose:us-east-1:483054109269:deliverystream/captains-kfh

Data transformation: Disabled

Creation time: September 24, 2021, 07:43 PDT

Dynamic partitioning: Disabled

Monitoring | Configuration | Destination error logs

Delivery stream metrics Info

1h 3h 12h 1d 3d 1w Custom

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. Click on 15 min.

Amazon Kinesis

Monitoring | Configuration | Destination error logs

Delivery stream metrics Info

Absolute Relative

Minutes: 1 3 5 15 **45**

Hours: 1 2 3 6 8 12

Days: 1 2 3 4 5 6

Weeks: 1 2 4 6

Months: 5 6 12 15

9. The graphs may take a few minutes to populate with data points.

Amazon Kinesis

Direct PUT

arn:aws:firehose:us-east-1:483054109269:deliverystream/captains-kfh

Disabled

Monitoring | Configuration | Destination error logs

Delivery stream metrics Info

1h 3h 12h 1d 3d 1w Custom (15m) Add to dashboard

Incoming bytes

Bytes: 5.24M 2.62M

Incoming put requests

Count: 2.00K 1.00K

Incoming records

Count: 500K 250K

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10. Go to the Amazon S3 dashboard.

11. Select ips-s3-bucket.

The screenshot shows the AWS S3 Buckets page. At the top, there are buttons for 'Create bucket', 'Copy ARN', 'Empty', and 'Delete'. Below is a search bar and a table with columns: Name, AWS Region, Access, and Creation date. A red box highlights the row for 'ips-s3-bucket', which is located in the 'US East (N. Virginia) us-east-1' region. An orange arrow points to the first column of this row.

Name	AWS Region	Access	Creation date
ips-s3-bucket	US East (N. Virginia) us-east-1	Bucket and objects not public	September 24, 2021, 07:41:35 (UTC-07:00)

12. Click on the Year, Month, Day, Hour folders to view the objects.

The screenshot shows the AWS S3 Objects page. At the top, there are buttons for 'Create folder' and 'Upload'. Below is a search bar and a table with columns: Name, Type, Last modified, Size, and Storage class. A red box highlights the row for a file named 'captains-kfh-1-2021-09-24-14-52-34-a3b411d4-55ef-41c8-b49c-90455098e540'. An orange arrow points to the first column of this row.

Name	Type	Last modified	Size	Storage class
captains-kfh-1-2021-09-24-14-52-34-a3b411d4-55ef-41c8-b49c-90455098e540	-	September 24, 2021, 07:57:37 (UTC-07:00)	2.3 MB	Standard

13. Go back to the terminal. Press **Ctrl+C** to stop sending data to the stream.

```
request, operation_model, context)
File "/usr/local/lib/python3.7/site-packages/botocore/endpoint.py", line 167, in _get_response
    request, operation_model)
File "/usr/local/lib/python3.7/site-packages/botocore/endpoint.py", line 200, in _do_get_response
    http_response = self._send(request)
File "/usr/local/lib/python3.7/site-packages/botocore/endpoint.py", line 269, in _send
    return self.http_session.send(request)
File "/usr/local/lib/python3.7/site-packages/botocore/httpsession.py", line 353, in send
    chunked=chunked(request.headers),
File "/usr/local/lib/python3.7/site-packages/urllib3/connectionpool.py", line 706, in urlopen
    chunked=chunked,
File "/usr/local/lib/python3.7/site-packages/urllib3/connectionpool.py", line 445, in _make_request
    six.raise_from(e, None)
File "<string>", line 3, in raise_from
File "/usr/local/lib/python3.7/site-packages/urllib3/connectionpool.py", line 440, in _make_request
    httplib_response = conn.getresponse()
File "/usr/lib64/python3.7/http/client.py", line 1369, in getresponse
    response.begin()
File "/usr/lib64/python3.7/http/client.py", line 310, in begin
    version, status, reason = self._read_status()
File "/usr/lib64/python3.7/http/client.py", line 271, in _read_status
    line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-1")
File "/usr/lib64/python3.7/socket.py", line 589, in readinto
    return self._sock.recv_into(b)
File "/usr/lib64/python3.7/ssl.py", line 1071, in recv_into
    return self.read(nbytes, buffer)
File "/usr/lib64/python3.7/ssl.py", line 929, in read
    return self._sslobj.read(len, buffer)
KeyboardInterrupt
[cloud_user@ip-10-0-1-7 ~]$
```

14. Type the **aws s3 ls** command to find the bucket name.

```
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$ aws s3 ls ←  
2021-09-24 14:32:33 cfst-3254-7248ef1e09b4577ea6f22d-cloudtrailbucket-1p8kkkhifayml  
2021-09-24 14:41:35 ips-s3-bucket  
[cloud_user@ip-10-0-1-7 ~]$
```

15. Type **aws s3 sync s3://ips-s3-bucket .** to view the bucket contents.

```
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$ aws s3 ls  
2021-09-24 14:32:33 cfst-3254-7248ef1e09b4577ea6f22d-cloudtrailbucket-1p8kkkhifayml  
2021-09-24 14:41:35 ips-s3-bucket  
[cloud_user@ip-10-0-1-7 ~]$ aws s3 sync s3://ips-s3-bucket . ←
```

16. Type the **ll** command to view the folder structure.

```
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$ ll  
total 4  
drwxrwxr-x 3 cloud_user cloud_user 16 Sep 24 15:01 2021  
-rw-r--r-- 1 cloud_user cloud_user 1052 Sep 24 14:39 send_captains_to_cloud.py  
[cloud_user@ip-10-0-1-7 ~]$
```

17. Type **cd <YEAR>/<MONTH>/<DAY>/<HOUR>/** to change the directory.

```
[cloud_user@ip-10-0-1-7 14]$  
[cloud_user@ip-10-0-1-7 14]$ cd 2021/09/24/14
```

18. Type the **ll** command to view the folder structure.

```
[cloud_user@ip-10-0-1-7 14]$  
[cloud_user@ip-10-0-1-7 14]$ ll
```

19. Copy one of the file names.

20. Type the **cut -c -200 <INSERT_FILE_NAME>** command to view the first 200 characters of the file.

```
[cloud_user@ip-10-0-1-7 14]$  
[cloud_user@ip-10-0-1-7 14]$ ll  
total 3072  
-rw-rw-r-- 1 cloud_user cloud_user 2373184 Sep 24 14:57 captains-kfh-1-2021-09-24-14-52-34-a3b411d4-55ef-41c8-b49c-90455098e540  
455098e540  
-rw-rw-r-- 1 cloud_user cloud_user 769093 Sep 24 15:02 captains-kfh-1-2021-09-24-14-57-35-be2b8ac5-cb7d-4130-8b0e-ab74527b9364  
[cloud_user@ip-10-0-1-7 14]$ cut -c -200 captains-kfh-1-2021-09-24-14-52-34-a3b411d4-55ef-41c8-b49c-90455098e540  
{"user": "Brenda Barker", "favoritecaptain": "Malcolm Reynolds", "rating": 283, "timestamp": 1632495154.0062149}, {"user": "Caitlin Daniel", "favoritecaptain": "Neil Armstrong", "rating": 8732, "timestamp": 1632495154.0062149}  
[cloud_user@ip-10-0-1-7 14]$
```

21. Type the **cd ..** command to return to the home directory.

```
[cloud_user@ip-10-0-1-7 14]$  
[cloud_user@ip-10-0-1-7 14]$ cd ..
```



22. Type the `python3 send_captains_to_cloud.py` command to start the data-generating stream again.

```
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$  
[cloud_user@ip-10-0-1-7 ~]$ python3 send_captains_to_cloud.py
```



Step 3: Create Kinesis Data Analytics

1. Click on Services.

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with the AWS logo, a "Services" dropdown menu (which is currently selected and highlighted with a red box and an orange arrow), a search bar, and other account information like "cloud_user @ 4830-5410-9269" and "N. Virginia". Below the navigation bar, the main content area is titled "AWS Management Console". On the left, there is a sidebar titled "AWS services" with sections for "Recently visited services" (Kinesis, EC2, S3) and "All services". At the bottom of the page, there are links for "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preferences".

2. Select Kinesis from Analytics.

The screenshot shows the AWS Services dashboard. On the left, there's a sidebar with 'Favorites' and 'Recently visited' sections. The 'Recently visited' section includes links for 'Console', 'Home', 'Kinesis', 'S3', and 'EC2'. Below this is a 'All services' button. The main area is titled 'All services' and lists various AWS services in a grid. The 'Kinesis' service is highlighted with a red box and a red arrow pointing to it. Other services listed include Transfer, AWS, Migration, Hub, AWS Application Migration Service, Grafana, Amazon Application Discovery Service, Amazon Database Migration Service, AWS QuickSight, Prometheus, AWS Proton, Incident Manager, Amazon OpenSearch Service, Groups & Tag Editor, Elasticsearch Service, Amazon Resource Optimizer, Resource (successor to), Amazon Elasticsearch Service, Alexa for Business, WorkMail, WorkSpaces, AWS Data Exchange, AWS Glue, End User Computing, and AppStream 2.0. At the bottom of the page are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

3. Select Kinesis Data Analytics. Then, click on the Create application button.

The screenshot shows the 'Get started' screen for Kinesis Data Analytics. It features three options: 'Kinesis Data Streams' (with a description of collecting streaming data with a data stream), 'Kinesis Data Firehose' (with a description of processing and delivering streaming data with a data delivery stream), and 'Kinesis Data Analytics' (with a description of analyzing streaming data with a data analytics application). The 'Kinesis Data Analytics' option is selected and highlighted with a red box and a red arrow. Below these options is a large orange 'Create application' button, which is also highlighted with a red box and a red arrow.

4. Select Legacy SQL.

aws Services ▾

cloud_user @ 4830-5410-9269 ▾ N. Virginia ▾ Support

Runtime
After you create the application, you can't change the type or version of the runtime environment.

Apache Flink - Streaming application
Apache Flink is an open-source framework and distributed processing engine for stateful computations over unbounded and bounded data streams. Use this option to build streaming application using Apache Flink in Java, Scala, and Python. You can also build Java-based streaming applications using Apache Beam. Apache Beam is an open source, unified model and set of language-specific SDKs for defining and executing data processing workflows.

Legacy SQL
Process data in real-time using Kinesis Data Analytics legacy SQL engine, which provides an easy way to quickly query large volumes of streaming data. We do not recommend this option for new applications, instead use [Studio applications](#).

⚠️ For new applications, we recommend that you use [Kinesis Data Analytics Studio](#) instead of Kinesis Data Analytics for SQL (legacy) for running SQL queries. Kinesis Data Analytics Studio provides advanced analytical capabilities, enabling you to build

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Give an application name **IPS-DataAnalytics**.
6. Type **IPSpecialist data analytics lab** in the **Description** box.

aws Services ▾

cloud_user @ 4830-5410-9269 ▾ N. Virginia ▾ Support

Application name
IPS-DataAnalytics

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Description - optional
IPSpecialist data analytics lab

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. [Learn more](#)

No tags associated with this application.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

7. Click on the **Create application** button.

The screenshot shows the AWS Kinesis Data Analytics console. At the top, there's a navigation bar with the AWS logo, 'Services ▾', a search icon, a bell icon, the user 'IPSpecialist @ ipscloud-aws ▾', the region 'N. Virginia ▾', and 'Support ▾'. Below the navigation bar is a table titled 'Setting' with columns 'Setting' and 'Value'. The table contains the following rows:

Setting	Value
Snapshots	On
Logging with Amazon CloudWatch Logs	INFO
Monitoring metrics level with Amazon CloudWatch	Task
Parallelism	12
Parallelism per KPU	1
Automatic scaling	On

At the bottom right of the table area is a large orange arrow pointing right, followed by a red rectangular button with the text 'Create application'.

Below the table, there's a footer with links: 'Feedback', 'English (US) ▾', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the bottom states '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

Hence, the Kinesis Data Analytics application has been created.

8. Click on **Steps to configure your application**.

The screenshot shows the AWS Kinesis Data Analytics console. At the top, there's a navigation bar with the AWS logo, 'Services ▾', a search icon, a bell icon, the user 'cloud_user @ 4830-5410-9269 ▾', the region 'N. Virginia ▾', and 'Support ▾'. Below the navigation bar, the application name 'IPS-DataAnalytics' is displayed. To the right of the application name are three buttons: 'Run', 'View CloudWatch dashboard', and 'Actions ▾'. A large orange arrow points from the previous screenshot to this section.

Below the application name, there's a section titled 'Application details' with a red box around the link '▶ Steps to configure your application'. This link is blue and underlined. To the right of the link is a small 'Info' icon. The rest of the page below this section is mostly blank.

At the bottom, there's a footer with links: 'Feedback', 'English (US) ▾', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the bottom states '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

9. Click on the **Configure source stream** button.

Sales

Services ▾

cloud_user @ 4830-5410-9269 ▾ N. Virginia ▾ Support

Step 1
Configure source stream

Choose an existing Kinesis data stream or Kinesis Data Firehose delivery stream as input. Kinesis Data Analytics ingests the data, automatically recognizes standard data formats, and suggests a schema.

Configure source stream

Step 2
Run real-time analytics with your SQL code

Use the Kinesis Data Analytics SQL editor and built-in templates to write queries to process streaming data.

Configure SQL

Step 3
Configure destinations

Point to the destinations where you want the data processed.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



10. Select Kinesis Data Firehose delivery stream.

11. Click on the Browse button.

Sales

Services ▾

cloud_user @ 4830-5410-9269 ▾ N. Virginia ▾ Support

Source Info

Source

Kinesis data stream
Use a data stream for rapid and continuous data intake and aggregation.

Kinesis Data Firehose delivery stream
Use a delivery stream to deliver real-time streaming data to destinations such as Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, or Splunk.

Delivery stream

Choose a delivery stream or enter a delivery stream ARN

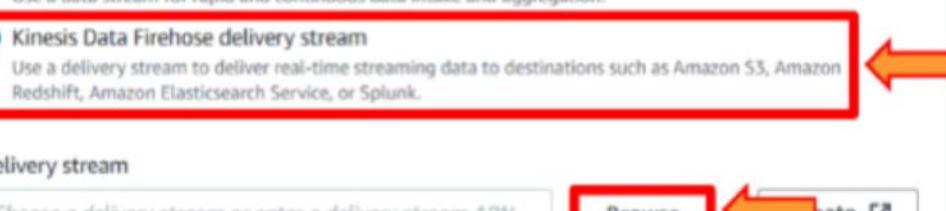
Browse

Format: arn:aws:firehose:[Region]:[AccountId]:deliverystream/[DeliveryStreamName]

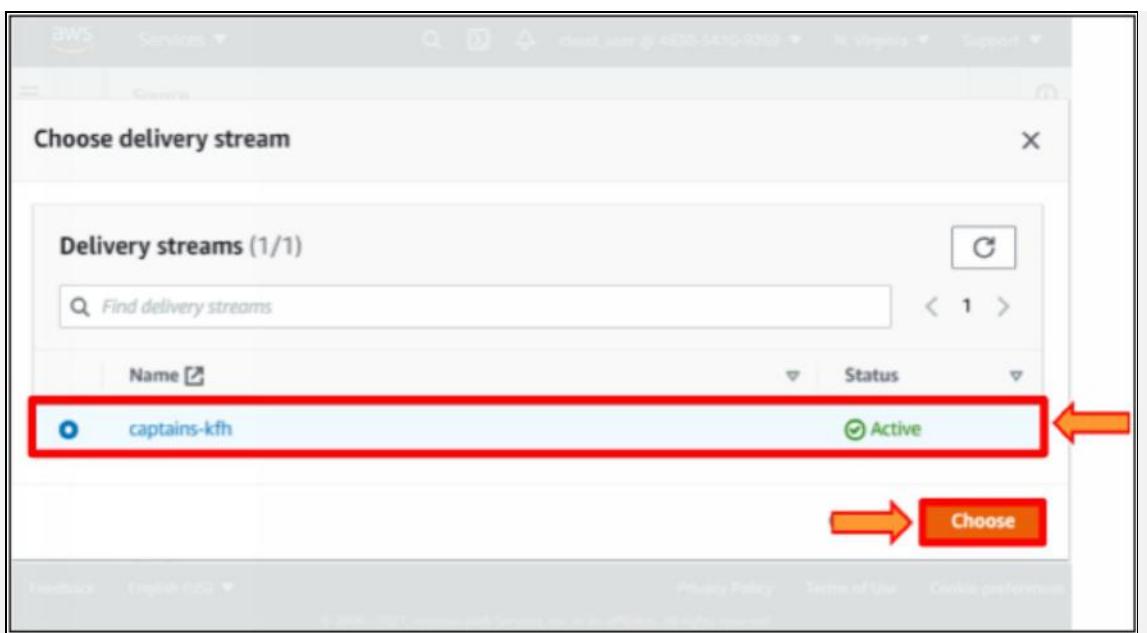
Record preprocessing with AWS Lambda

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

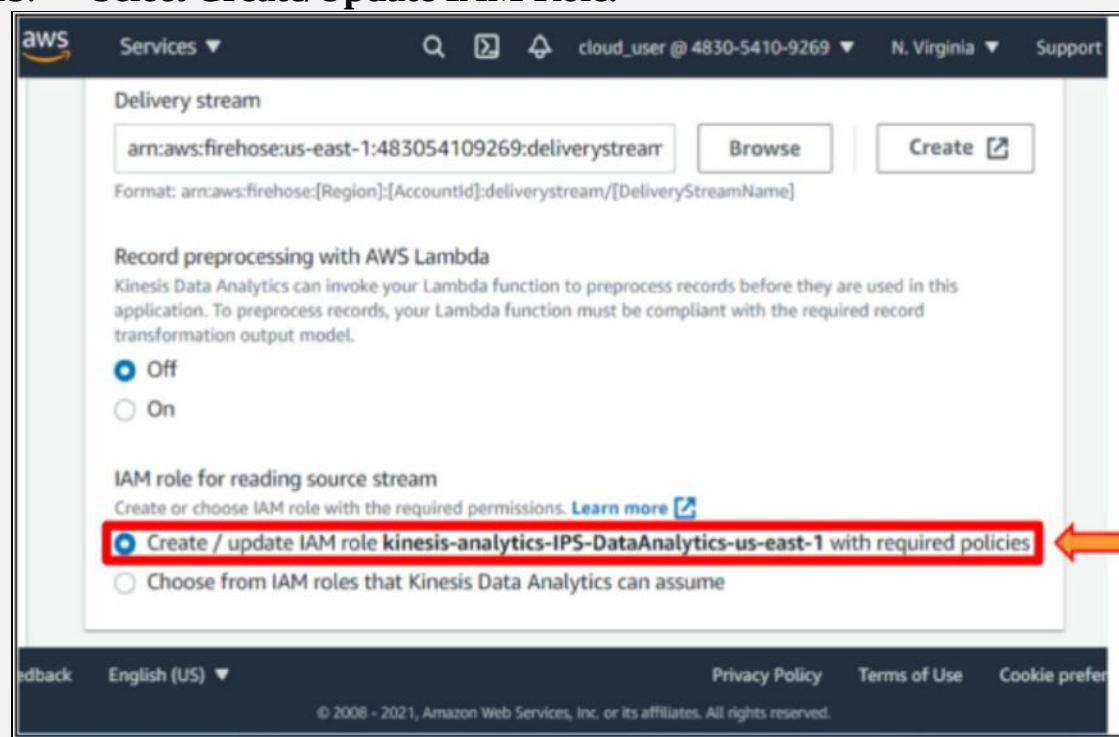
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



12. Select captains-kfh. Then, click on the Choose button.



13. Select Create/Update IAM Role.



14. Scroll down. Click on the Discover Schema button,

The screenshot shows the AWS Schema discovery interface. At the top right, there is a red box and an arrow pointing to the 'Discover schema' button. Below it, a message box contains the text: 'Further action is required to apply your changes. To generate a schema using recent records from the source, choose Discover schema.' At the bottom right of the main area are 'Cancel' and 'Save changes' buttons.

Schema Info

Schema discovery can generate a schema using records from the source. Schema column names are the same as in the source, unless they contain special characters, repeated column names, or reserved keywords.

⚠ Further action is required to apply your changes.

To generate a schema using recent records from the source, choose **Discover schema**.

Cancel Save changes

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

15. It will take a few minutes to populate the table.

The screenshot shows the AWS Lambda output interface. On the left, there is a red arrow pointing to the 'Formatted data' tab. In the main area, there is a red box around the table schema and data. The schema includes columns: COL_user (VARCHAR(32)), favoritecaptain (VARCHAR(16)), rating (INTEGER), and COL_timestamp (DOUBLE). The data table lists four rows:

COL_user	favoritecaptain	rating	COL_timestamp
Christina English	Malcolm Reynolds	74	1.6324975377215276E9
Michelle Barton	The Doctor	203	1.632497537734063E9
Erin Alvarado	The Doctor	574	1.6324975377049947E9
Jeffrey Monroe	Han Solo	197	1.6324975377484715E9

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

16. Scroll down. Click on the **Save changes** button.

A screenshot of the AWS Lambda function configuration page. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and user information ('cloud_user @ 4830-5410-9269', 'N. Virginia', 'Support'). Below the navigation is a table with six rows of data:

Erin Alvarado	The Doctor	574	1.63249753770...
Jeffrey Monroe	Han Solo	197	1.63249753774...
Ryan Velazquez	Neil Armstrong	8799	1.63249753776...
Isaac Cruz	The Doctor	402	1.63249753768...
Mrs. Amy Wall	The Doctor	284	1.63249753767...
Kristin Patrick	Jacob Keyes	639	1.63249753766...

At the bottom right of the table area is a red arrow pointing to a red-bordered 'Save changes' button.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

17. Scroll down. Click on the **Real-time analytics** tab.

18. Click on the **Configure** button.

A screenshot of the Real-time analytics configuration page. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and user information ('cloud_user @ 4830-5410-9269', 'N. Virginia', 'Support'). Below the navigation, there are three tabs: 'Source' (disabled), 'Real-time analytics' (selected and highlighted with a red box and arrow), and 'Tags'. A callout box states: 'This page is only for viewing your code. Any changes that you make to the code in this page can't be saved. To edit the code, choose Configure.' Below the tabs is a SQL editor with the following code:

```

1 /**
2 * Welcome to the SQL editor
3 *
4 */

```

Below the SQL editor are two buttons: 'View raw SQL' and 'Configure' (highlighted with a red box and arrow). At the bottom, there's a feedback link and a footer with 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

19. Copy and paste the below SQL code.

```

CREATE OR REPLACE STREAM "CAPTAIN_SCORES" ("favoritecaptain"
varchar(32),      "average_rating"      DOUBLE,      "total_rating"
INTEGER);

CREATE OR REPLACE PUMP "STREAM_PUMP" as
INSERT INTO "CAPTAIN_SCORES"
SELECT      STREAM      "favoritecaptain",      avg("rating"),
sum("rating")
FROM SOURCE_SQL_STREAM_001
GROUP      BY      "favoritecaptain",      STEP
("SOURCE_SQL_STREAM_001".ROWTIME BY INTERVAL '1' MINUTE)
ORDER BY STEP("SOURCE_SQL_STREAM_001".ROWTIME BY INTERVAL '1'
MINUTE), avg("rating") desc;

```

20. Click on the Save and run application button.

The screenshot shows the AWS Lambda SQL code editor interface. At the top, there are tabs for 'SQL code' and 'Info'. Below the tabs are three buttons: 'View raw SQL', 'Add SQL from templates', 'Save application', and 'Save and run application'. The 'Save and run application' button is highlighted with a red arrow. The main area contains a block of SQL code:

```
1 CREATE OR REPLACE STREAM "CAPTAIN_SCORES" ("favoritecaptain" varchar(32),  
2 "average_rating" DOUBLE, "total_rating" INTEGER);  
3  
4 CREATE OR REPLACE PUMP "STREAM_PUMP" as  
5 INSERT INTO "CAPTAIN_SCORES"  
6 SELECT STREAM "favoritecaptain", avg("rating"), sum("rating")  
7 FROM SOURCE_SQL_STREAM_001  
8 GROUP BY "favoritecaptain", STEP ("SOURCE_SQL_STREAM_001".ROWTIME BY  
9 INTERVAL '1' MINUTE)  
8 ORDER BY STEP("SOURCE_SQL_STREAM_001".ROWTIME BY INTERVAL '1' MINUTE),  
avg("rating") desc;
```

At the bottom of the editor, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the very bottom reads: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

21. It will take a few minutes to populate the table.

The screenshot shows the AWS Lambda CAPTAIN_SCORES - sample results table. The table has four columns: ROWTIME, favoritecaptain, average_rating, and total_rating. The data is as follows:

ROWTIME	favoritecaptain	average_rating	total_rating
2021-09-...	Neil Arms...	7972.0	1610440
2021-09-...	Han Solo	533.0	146143
2021-09-...	John Robi...	522.0	157665
2021-09-...	Jacob Keyes	520.0	164630

At the bottom of the table, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the very bottom reads: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

22. Copy and paste the below SQL code to find the anomalous ratings.

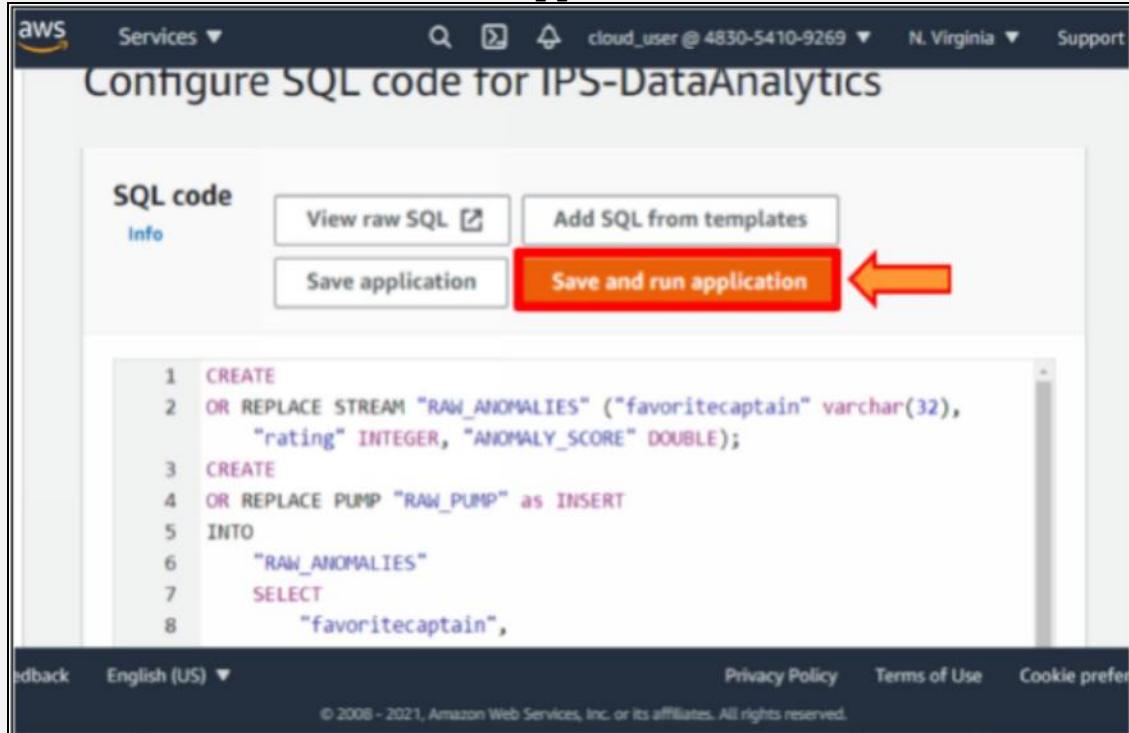
```
CREATE  
OR REPLACE STREAM "RAW_ANOMALIES" ("favoritecaptain"  
varchar(32), "rating" INTEGER, "ANOMALY_SCORE" DOUBLE);  
CREATE  
OR REPLACE PUMP "RAW_PUMP" as INSERT  
INTO  
"RAW_ANOMALIES"  
SELECT  
"favoritecaptain,"  
"rating",
```

```

"ANOMALY_SCORE"
FROM
  TABLE(  RANDOM_CUT_FOREST(CURSOR(SELECT
    STREAM "favoritecaptain",
    "rating"
  FROM
    "SOURCE_SQL_STREAM_001")) );
CREATE
OR REPLACE STREAM "ORDERED_ANOMALIES"
("favoritecaptain" varchar(32),
"rating" INTEGER,
"ANOMALY_SCORE" DOUBLE);
CREATE
OR REPLACE PUMP "ORDERED_PUMP" as INSERT
INTO
  "ORDERED_ANOMALIES"
SELECT
  STREAM *
FROM
  RAW_ANOMALIES
ORDER BY
  FLOOR("RAW_ANOMALIES".ROWTIME TO SECOND),
  "ANOMALY_SCORE" desc;

```

23. Click on the **Save and run application** button.



24. It will take a few minutes to populate the table.

ROWTIME (T1...)

favoritecaptain

rating

ANOMALY_SCORE

2021-09-24 15:3...	Neil Armstrong	8125	1.7595832655430987
2021-09-24 15:3...	Neil Armstrong	7532	1.6886275421454413
2021-09-24 15:3...	David Bowman	998	1.2246133284816703
2021-09-24 15:3...	The Doctor	994	1.1056937041537935

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lab 06: Data Analysis & Visualization

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	QuickSight, S3, SageMaker, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

AWS QuickSight

Amazon QuickSight is a cloud-based business analytics solution that enables all employees to create visualizations, do ad-hoc analysis, and quickly get business insights from their data, anytime and on any device. Upload CSV and Excel files; connect to SaaS apps like Salesforce; connect to on-premises databases like SQL Server, MySQL, and PostgreSQL; and locate your AWS data sources in real-time, including Amazon Redshift, Amazon RDS, Amazon Aurora, Amazon Athena, and Amazon S3. QuickSight allows businesses to extend their business analytics capabilities to thousands of users while

still providing quick and responsive query processing. QuickSight enables businesses to expand their business analytics capabilities to thousands of users. It uses a powerful in-memory engine to offer quick and responsive query performance (SPICE).

AWS SageMaker

Amazon SageMaker is a fully managed service that allows any developer or data scientist to quickly create, train, and deploy Machine Learning (ML) models. SageMaker makes it easy to create high-quality models by removing the heavy lifting from each phase of the machine learning process.

AWS Simple Storage Service S3

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 is a web service that allows you to store and retrieve an infinite quantity of data from any place and at any time. You may quickly create projects that integrate cloud-native storage using this service because Amazon S3 is easily customizable, and you only pay for what you use. You can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. Build a simple FTP system or a complex web application like the Amazon.com retail website; read the same piece of data a million times or only for emergency disaster recovery; store whatever type and amount of data you desire.

Case Study Enterprise Public Service – IPS

Background

The Social Security Institute, known as IPS is decentralized public service with its own legal team and resources that was established in accordance with Article 53 of Law 20.255, which implemented a Social Security System overhaul in 2008. Using the Ministry of Labor and Social Welfare as a conduit, as well as the Undersecretary of Social Welfare, it operates under the authority of the President of the Republic. This service, which is a public utility overseen by Chile's Senior Public Management System, is responsible for managing the solidarity pension system and pension plans formerly managed by the former INP. Additionally, it runs the ChileAtiende network, which informs citizens on policies and government benefits.

Business Challenge

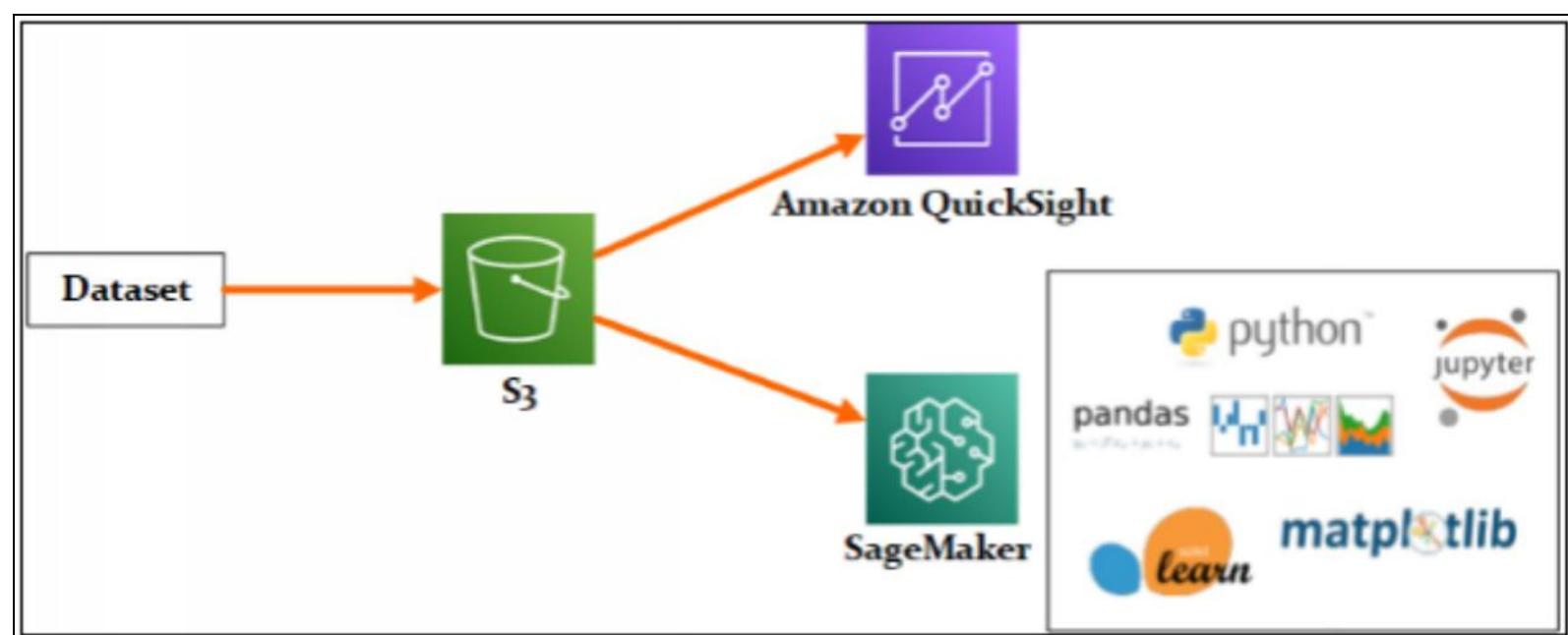
You work as a Data Analytics in IPS. IPS gives you the task of analyzing and visualizing the sales data of the car. To see the sales figures using the visualization of data. Which month generates the most sales? Which dealership sold the most cars? How can you automate visualizing and analyzing the data with drag and drop?

Proposed Solution

The solution is to use AWS services to visualize and analyze the data. You use AWS QuickSight service to drag and drop the visualizing and analyzing of data. Using Python libraries, you can also use AWS SageMaker Jupyter notebook to visualize the data. In this lab, you use AWS S3 to upload your data. Then, you will use AWS QuickSight to visualize data using a pie chart and vertical bar

chart. For the box plot, you will use AWS SageMaker Jupyter Notebook.

Lab Diagram



Implementation Steps

1. Create an S3 Bucket.
2. Visualize Data on AWS QuickSight.
3. Visualize Data on AWS SageMaker.

Solution

Step 1: Create an S3 Bucket

1. Log in to the **AWS Console**.
2. Click on **Services**.

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with the AWS logo, a 'Services' dropdown menu (which is currently selected), a search bar, and other account information like 'IPSpecialist @ ipscloud-aws' and 'N. Virginia'. Below the navigation bar, the main title is 'AWS Management Console'. On the left, there is a sidebar titled 'AWS services' with sections for 'Recently visited services' (listing 'S3') and 'All services'. At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with a copyright notice: '© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

3. Select **S3** from the **Storage**.

Sales & Marketing

IP Specialist @ ipscloud-aws N. Virginia Support

★ Favorites

Add favorites by clicking on the star next to the service name.

Recently visited

Console Home

All services

- OpenShift
- Service on AWS
- & Governance
- AWS Organization
- S3**
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup
- Amazon Transcribe
- Amazon Translate
- CloudWatch Metrics
- AWS Auto Scaling
- CloudFormation
- CloudTrail
- Config
- OpsWorks
- Service
- Amazon Step Functions
- Amazon Functions
- Amazon AppFlow
- Amazon EventBridge
- Amazon MQ
- Simple Notification Service
- Amazon Simple Service
- Amazon Simple Queue Service

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

4. Click on the Create bucket button.

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Buckets (0)

Buckets are containers for data stored in S3. [Learn more](#)

Create bucket

Find buckets by name

No buckets

You don't have any buckets.

Create bucket

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Give a bucket name **ips-s3-bucket**.

Amazon S3 > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more [\[?\]](#)

General configuration

Bucket name ←

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#) [\[?\]](#)

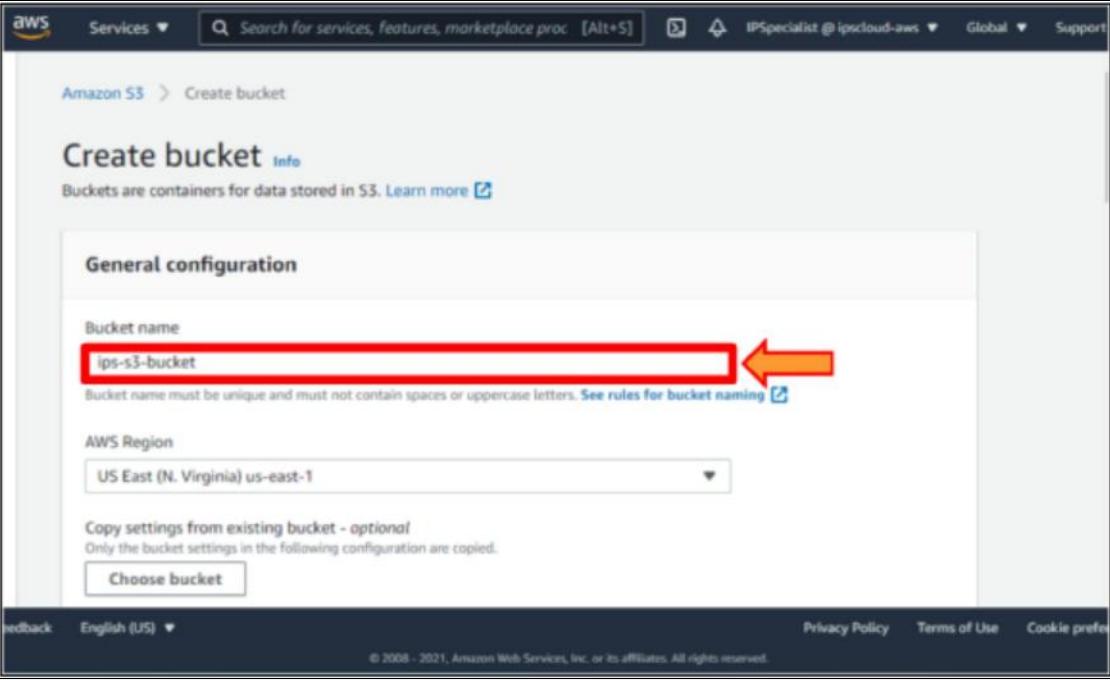
AWS Region

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Feedback English (US) [▼](#) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



6. Scroll down. Click on the **Create bucket** button.

Services [▼](#) Search for services, features, marketplace proc [Alt+S] IPSpecialist @ ipscloud-aws Global Support

Add tag

Default encryption
Automatically encrypt new objects stored in this bucket. [Learn more](#) [\[?\]](#)

Server-side encryption

Disable Enable

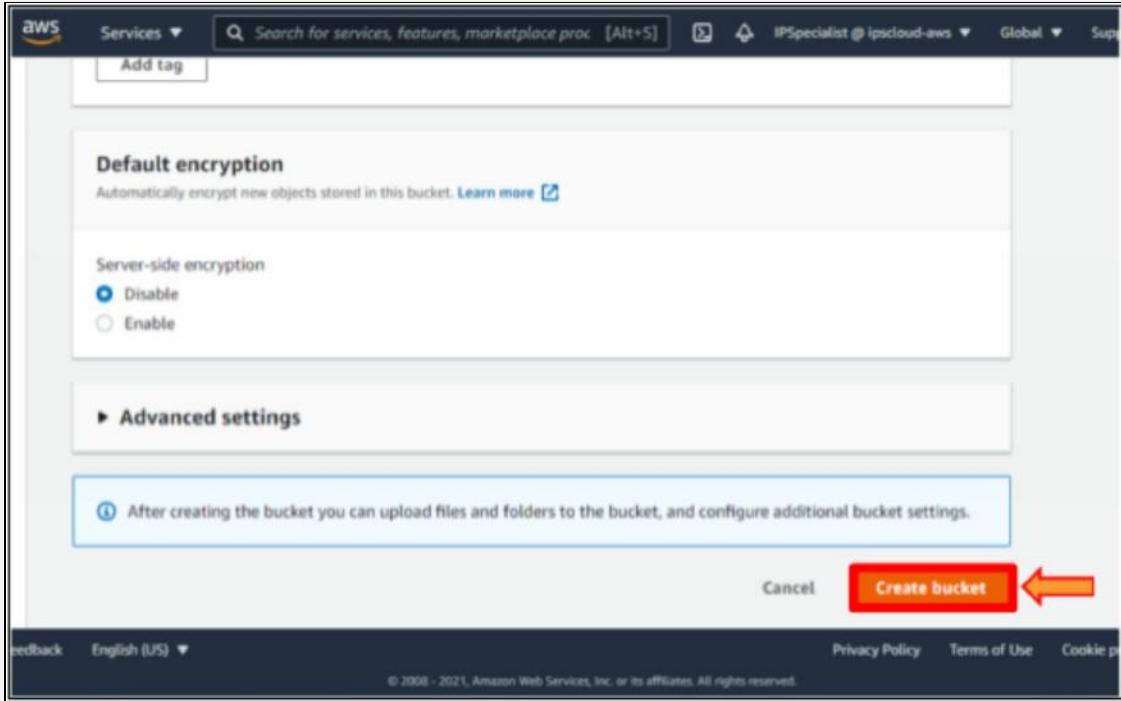
Advanced settings

After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) Create bucket ←

Feedback English (US) [▼](#) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



7. Download the car dataset provided in the following GitHub link:

https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter5/car_data.csv

Q Search this file...

1	car	year	engine_hp	avg_mpg	price	salesman	dealership	sold_date	sold_month
2	Corvette	2011	335.0	22.5	46135	2	Big Bobs	2012-05-06	5
3	Corvette	2011	300.0	23.5	40650	2	Uptown Cars	2011-05-16	5
4	Corvette	2011	300.0	24.0	36350	2	Uptown Cars	2013-07-31	7
5	Corvette	2011	230.0	23.0	29450	2	Uptown Cars	2014-07-05	7
6	Corvette	2011	230.0	23.0	34500	2	Uptown Cars	2013-05-20	5
7	Corvette	2012	230.0	23.0	31200	3	Car Town	2013-04-10	4
8	Corvette	2012	300.0	21.5	44100	3	Uptown Cars	2015-12-13	12
9	Corvette	2012	300.0	24.0	39300	2	Big Bobs	2013-05-19	5
10	Corvette	2012	230.0	23.0	36900	3	Uptown Cars	2015-04-27	4
11	Corvette	2013	230.0	22.5	37200	2	Car Town	2013-12-15	12
12	Corvette	2013	300.0	24.0	39600	3	Big Bobs	2015-05-30	5
13	Corvette	2013	230.0	23.5	31500	2	Uptown Cars	2014-03-21	3
14	Corvette	2013	300.0	23.5	44400	3	Uptown Cars	2016-03-18	3
15	Corvette	2013	230.0	23.5	37200	3	Uptown Cars	2013-02-19	2
16	Corvette	2013	230.0	23.5	31500	3	Car Town	2013-10-27	10
17	Corvette	2013	320.0	21.5	48250	2	Car Town	2016-01-27	1
18	Corvette	2013	320.0	24.0	43550	3	Big Bobs	2014-03-25	3

8. Click on the ips-s3-bucket.

The screenshot shows the AWS S3 Buckets page. At the top, there's an 'Account Snapshot' section with metrics: Total storage (38.4 KB), Object count (7), and Avg. object size (5.5 KB). A note says you can enable advanced metrics in the 'default-account-dashboard' configuration. Below this is a 'Buckets (1) Info' section. It shows one bucket named 'ips-s3-bucket' located in 'US East (N. Virginia)' (Region: us-east-1). The bucket has 'Access' set to 'Bucket and objects not public' and was created on 'October 2, 2021, 04:10:07 (UTC-07:00)'. There are buttons for 'Create bucket', 'Copy ARN', 'Empty', and 'Delete'. A red box highlights the 'ips-s3-bucket' row, and a red arrow points to the 'ips-s3-bucket' name.

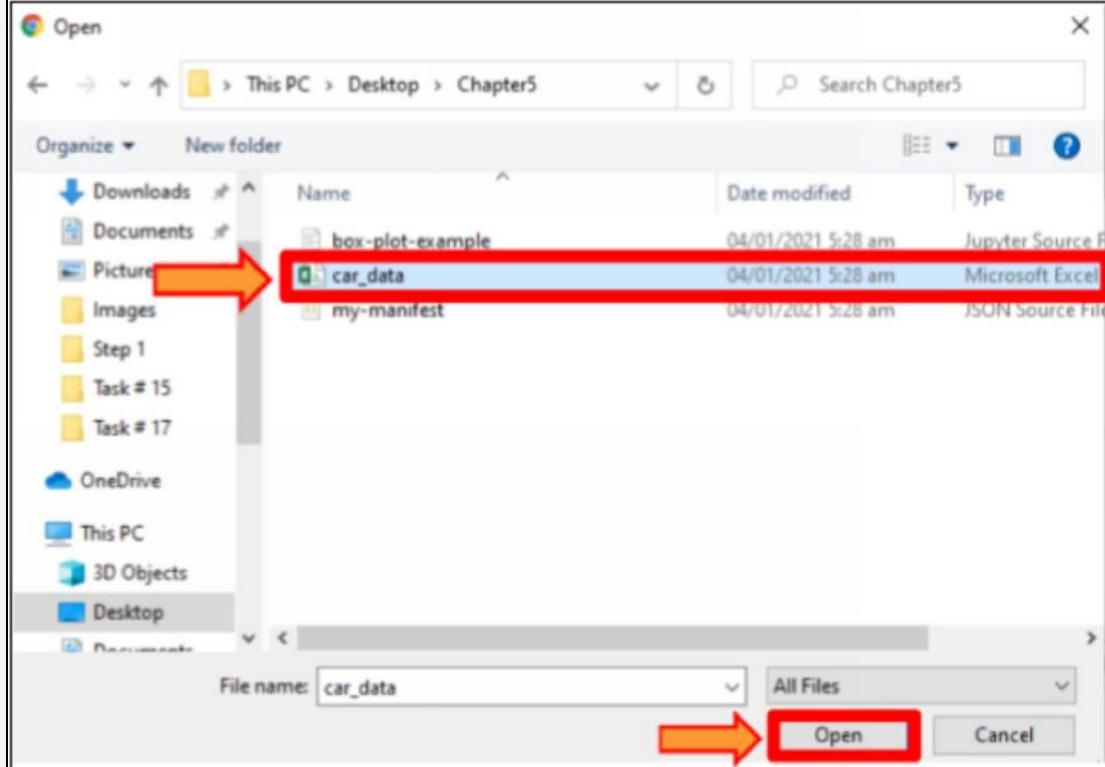
9. Click on the Upload button.

Screenshot of the AWS S3 console showing the 'Objects (0)' page. The top navigation bar includes 'Services', a search bar, and user information. Below the bar, there are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', and 'Actions'. A red arrow points to the 'Upload' button. A search bar below the buttons contains the placeholder 'Find objects by prefix'. The main content area shows a table header for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. Below the header, it says 'No objects' and 'You don't have any objects in this bucket.' A large 'Upload' button is centered at the bottom of the page. At the very bottom, there are links for 'Feedback', 'English (US)', and 'Cookie preferences'.

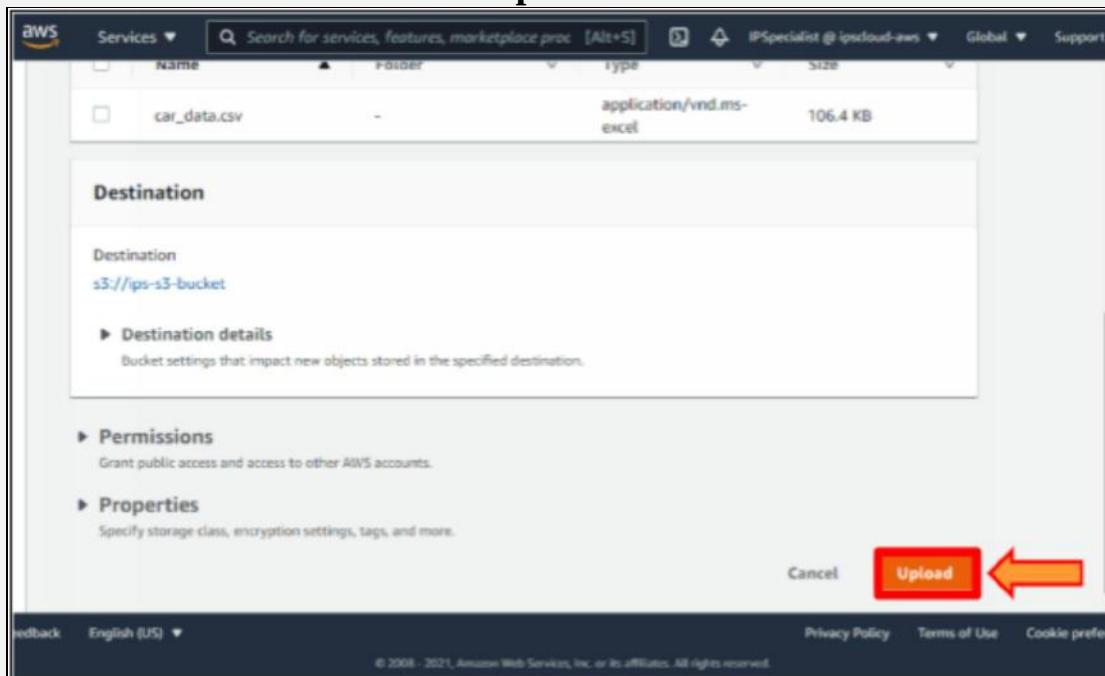
10. Click on the Add files button.

Screenshot of the AWS S3 'Upload' page. The top navigation bar shows the path 'Amazon S3 > ips-s3-bucket > Upload'. The main title is 'Upload' with a 'Info' link. Below the title, instructions say 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more'. A large dashed box in the center says 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.' Below this is a table titled 'Files and folders (0)'. The table has columns for 'Name', 'Folder', 'Type', and 'Size'. A red arrow points to the 'Add files' button in the top right of the table area. A search bar 'Find by name' is also present. At the bottom, there are links for 'Feedback', 'English (US)', and 'Cookie preferences'.

11. Select the car_data.csv. Then click on the Open button.



12. Scroll down. Click on the **Upload** button.



13. So now, the file has been uploaded.

Services ▾ IP Specialist @ ipscloud-aws ▾ Global ▾ Support ▾

Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://ips-s3-bucket	1 file, 106.4 KB (100.00%)	0 files, 0 B (0%)

[Files and folders](#) [Configuration](#)

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2: Visualize Data on AWS QuickSight

1. Click on Services.

[Services ▾](#)  IP Specialist @ ipscloud-aws ▾ N. Virginia ▾ Support ▾

AWS Management Console

AWS services

▼ Recently visited services
Your recently visited AWS services appear here.

▶ All services

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Select the QuickSight from the Analytics.

aws Services ▾ Q Search for services, features, marketplace product: [Alt+S] 🔍 IP Specialist @ ipscloud-aws N. Virginia Support

★ Favorites

Add favorites by clicking on the star next to the service name.

Recently visited

Console Home

All services

DynamoDB	AWS AppConfig	Analytics	Airflow
ElastiCache	Trusted Advisor	Athena	Business Applications
Neptune	Control Tower	Amazon Redshift	Amazon Connect
Amazon QLDB	AWS License Manager	EMR	Amazon Pinpoint
Amazon DocumentDB	AWS Well-Architected Tool	CloudSearch	Amazon Honeycode
Amazon Keyspaces	Personal Health Dashboard	Amazon OpenSearch Service (successor to AWS Chatbot)	Amazon Chime [↗]
Amazon Timestream	Launch Wizard	Amazon Elasticsearch Service	Amazon Simple Email Service
Amazon MemoryDB for Redis	AWS Compute Optimizer	Kinesis	Amazon WorkDocs
Migration & Transfer	Resource Groups & Tag Editor	QuickSight [↗]	Amazon WorkMail
AWS Migration Hub		Data Pipeline	Alexa for Business
AWS Application			

Get started with simple wizards and automated workflows.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences quicksight.amazonaws.com © 2008–2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Click on the Sign up for QuickSight button.

Quicksight

Your AWS Account is not signed up for QuickSight. Would you like to sign up now?

AWS Account

Sign up for QuickSight [↗]

To access QuickSight with a different account, log in again.

aws

4. Select Standard.

Quicksight

Create your QuickSight account

Edition

Enterprise [selected] Standard [↗]

	FREE	FREE
Team trial for 30 days (4 authors)*	\$18	\$28
Author per month (yearly)**	\$24	\$34
Author per month (monthly)**	\$0.30 / session (max \$5)***	\$0.30 / session (max \$10)***
Readers (pay-per-Session)	\$0.38 per GB	\$0.38 per GB
Additional SPICE per month	N/A	\$250 / mo / region
QuickSight Q regional fee	N/A	Starting from \$199
Personalized Q authoring workshop	N/A	INCLUDED
Natural language query with QuickSight Q	N/A	
Single Sign On with SAML or OpenID Connect	✓	✓

5. Select Use IAM federated identities & QuickSight-managed users.

6. Select **US East (N. Virginia)** in the QuickSight capacity region.

The screenshot shows the 'Create your QuickSight account' wizard. In the 'Authentication method' section, the radio button for 'Use IAM federated identities & QuickSight-managed users' is selected, and its description is visible. In the 'QuickSight region' section, a dropdown menu shows 'US East (N. Virginia)' selected. Both the 'Authentication method' and 'QuickSight region' sections are highlighted with red boxes and have orange arrows pointing towards them from the left.

7. Name your QuickSight account.
8. Set your Notification email address.

The screenshot shows the 'Name your account' step of the wizard. It includes fields for 'QuickSight account name' (containing 'ipspecialist-quicksight') and 'Notification email address' (containing 'ipsexample@email.com'). Both of these fields are highlighted with red boxes and have orange arrows pointing towards them from the left.

9. Click on the **Amazon S3** Check box.

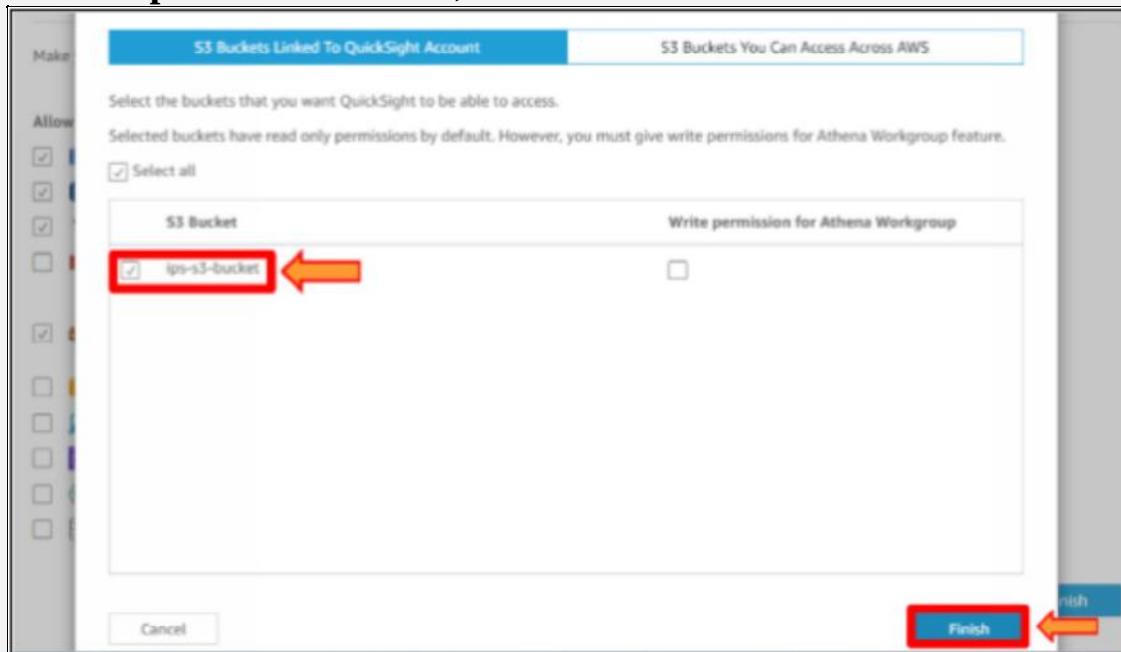
Make your existing AWS data and users available in QuickSight. Learn more

Allow access and autodiscovery for these resources

- Amazon Redshift
- Amazon RDS
- IAM
- Amazon S3** Select S3 buckets
- Amazon Athena
Make sure you've chosen the right Amazon S3 buckets for QuickSight access
- Amazon S3 Storage Analytics
- AWS IoT Analytics
- Amazon OpenSearch Service
- Amazon SageMaker
- Amazon Timestream

Finish

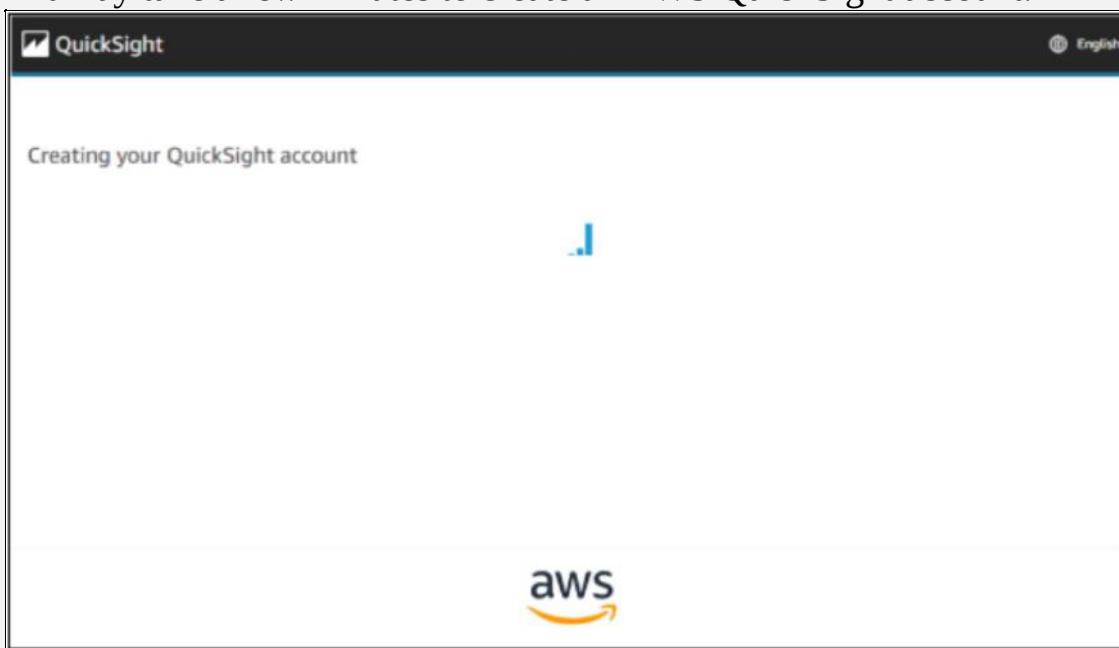
10. Select **ips-s3-bucket**. Then, click on the **Finish** button.



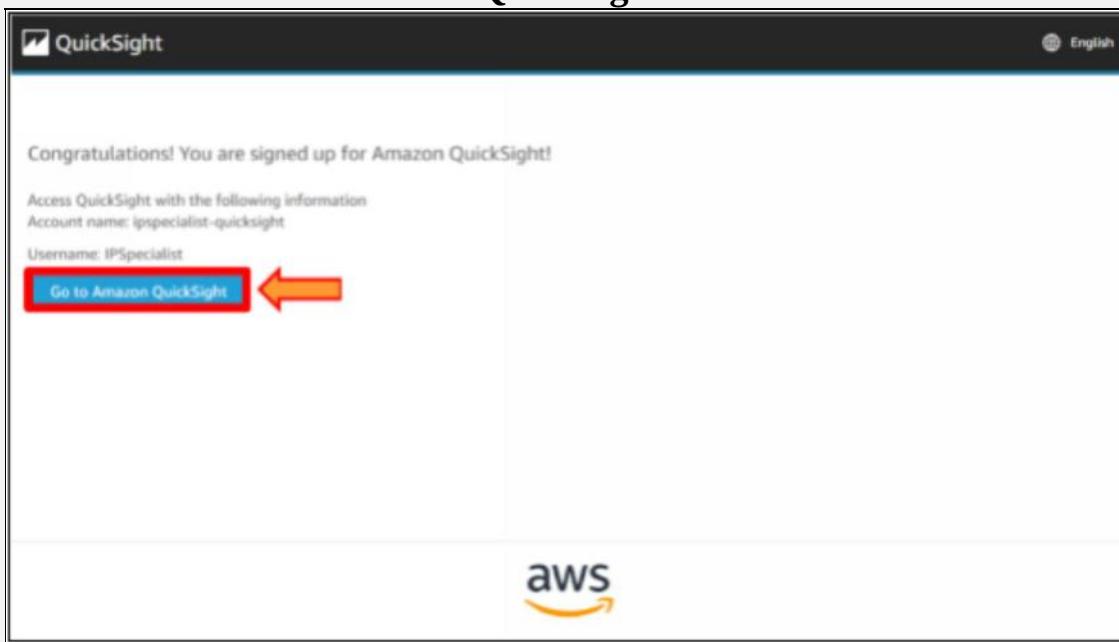
11. Click on the **Finish** button.



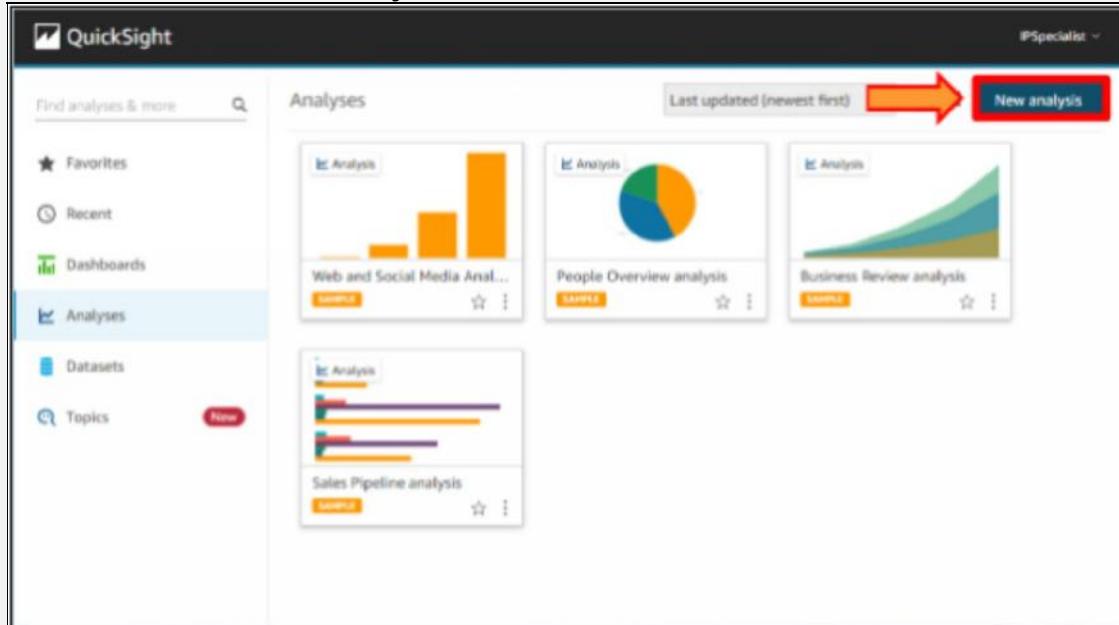
12. It may take a few minutes to create an AWS QuickSight account.



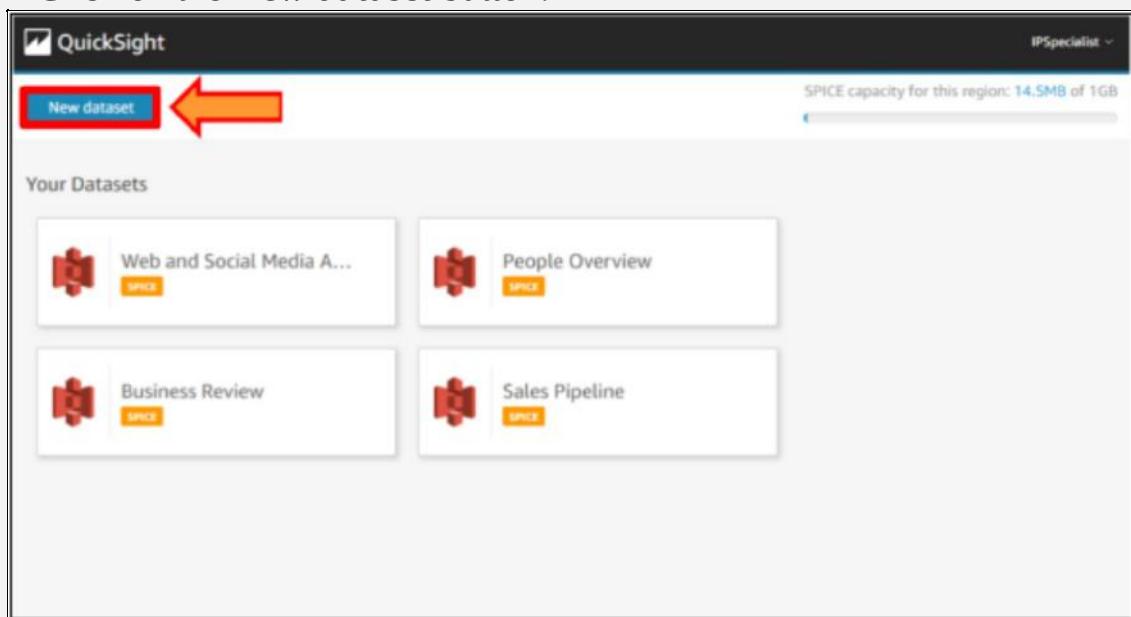
13. Click on the **Go to Amazon QuickSight** button.



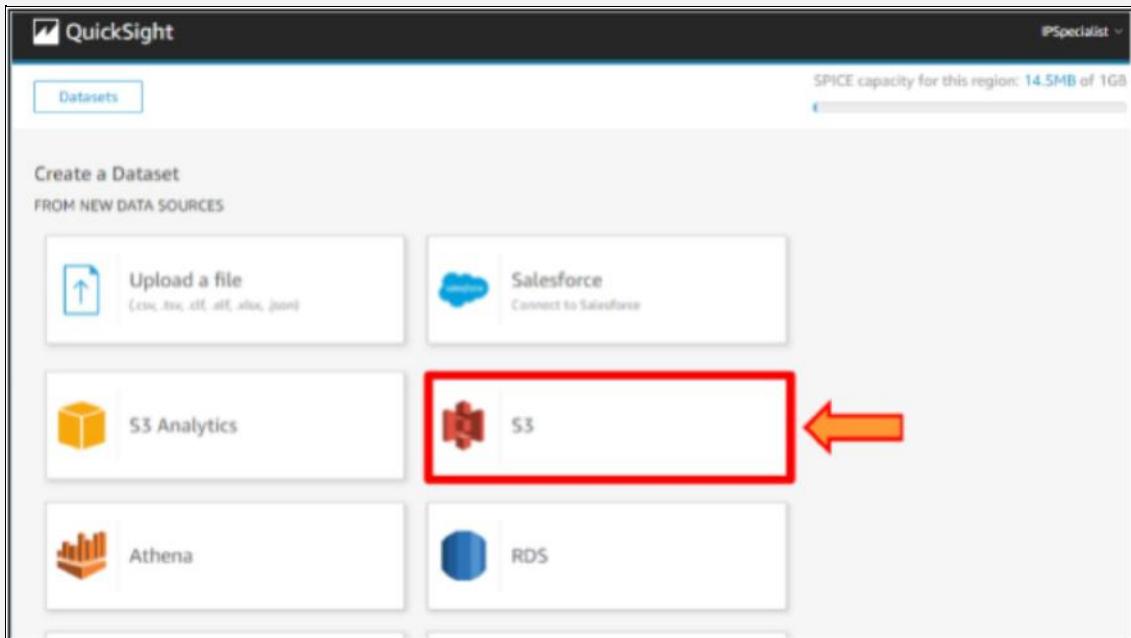
14. Click on the **New Analysis** button.



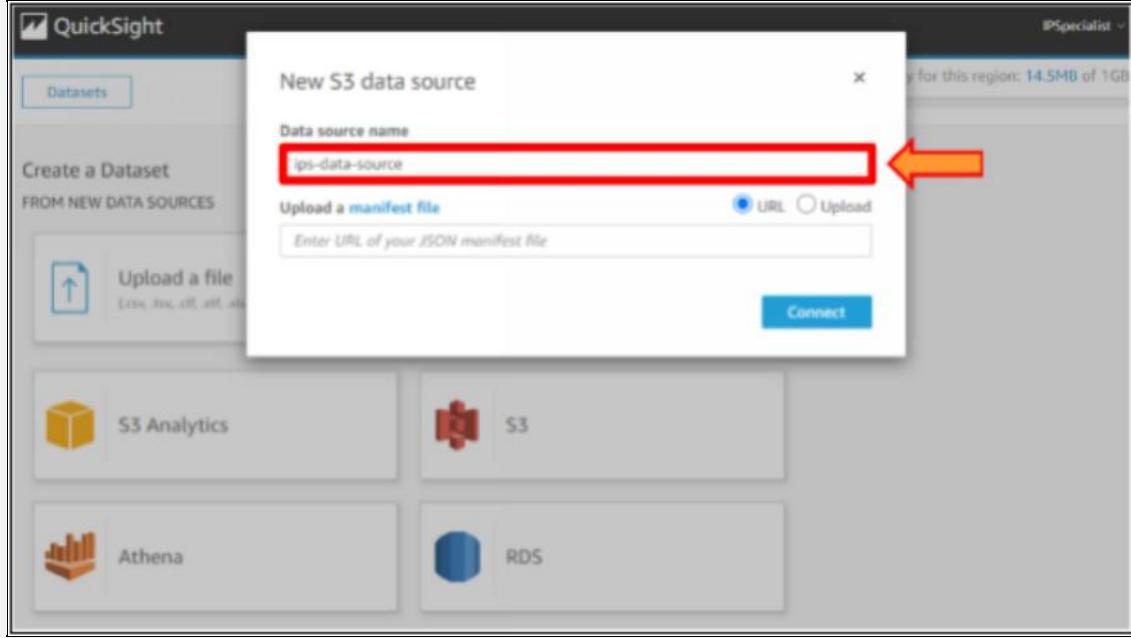
15. Click on the New dataset button.



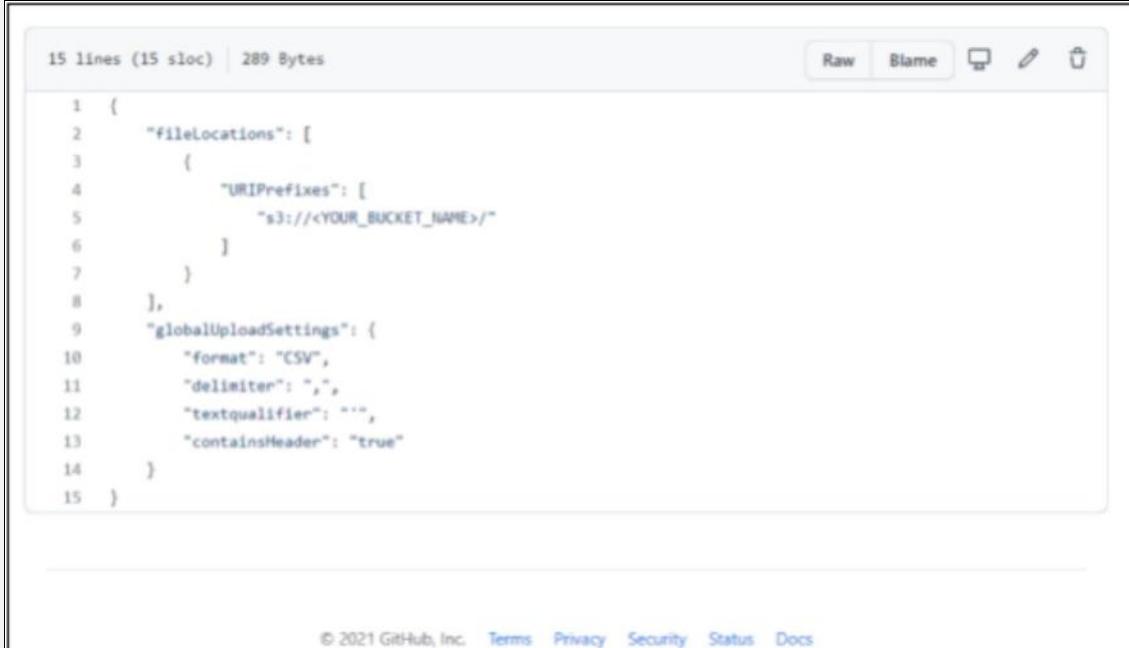
16. Click on the S3.



17. Give the Data Source name **ips-data-source**.



18. Download the my-manifest.json file from the following GitHub link:
https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter5/my-manifest.json

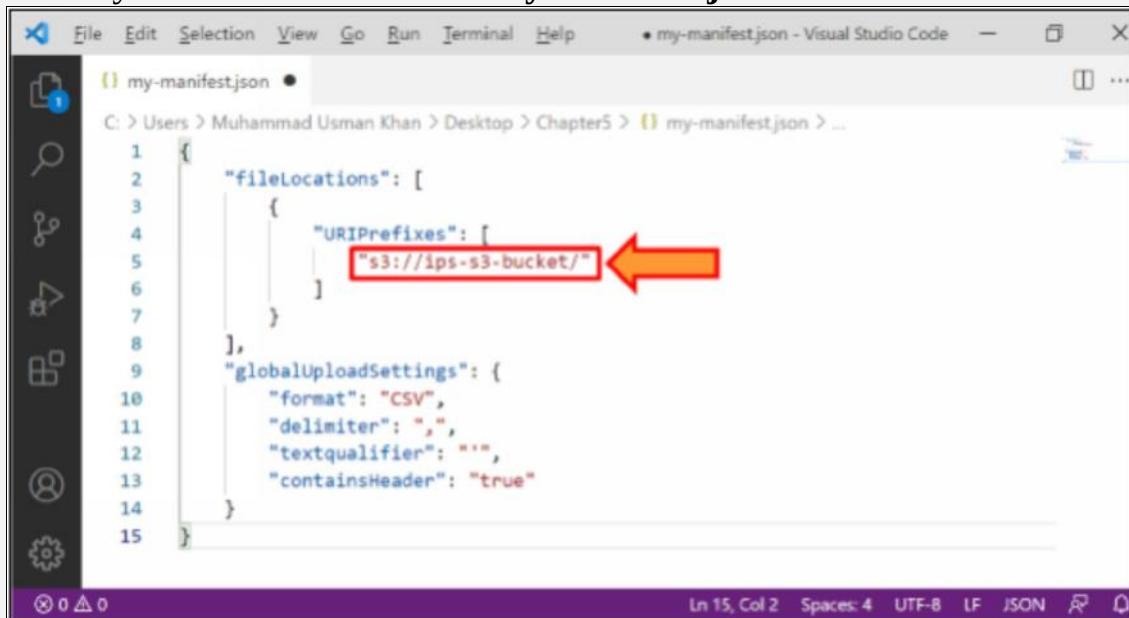


The screenshot shows a GitHub code viewer with the following details:

- File content: my-manifest.json
- Line count: 15 lines (15 sloc)
- Byte count: 289 Bytes
- Actions: Raw, Blame, Copy, Edit, Delete
- Code content:

```
1  {
2      "fileLocations": [
3          {
4              "URIPrefixes": [
5                  "s3://<YOUR_BUCKET_NAME>/"
6              ]
7          }
8      ],
9      "globalUploadSettings": {
10         "format": "CSV",
11         "delimiter": ",",
12         "textqualifier": """",
13         "containsHeader": "true"
14     }
15 }
```
- Footer: © 2021 GitHub, Inc. Terms Privacy Security Status Docs

19. Add your bucket name in the **my-manifest.json** file.

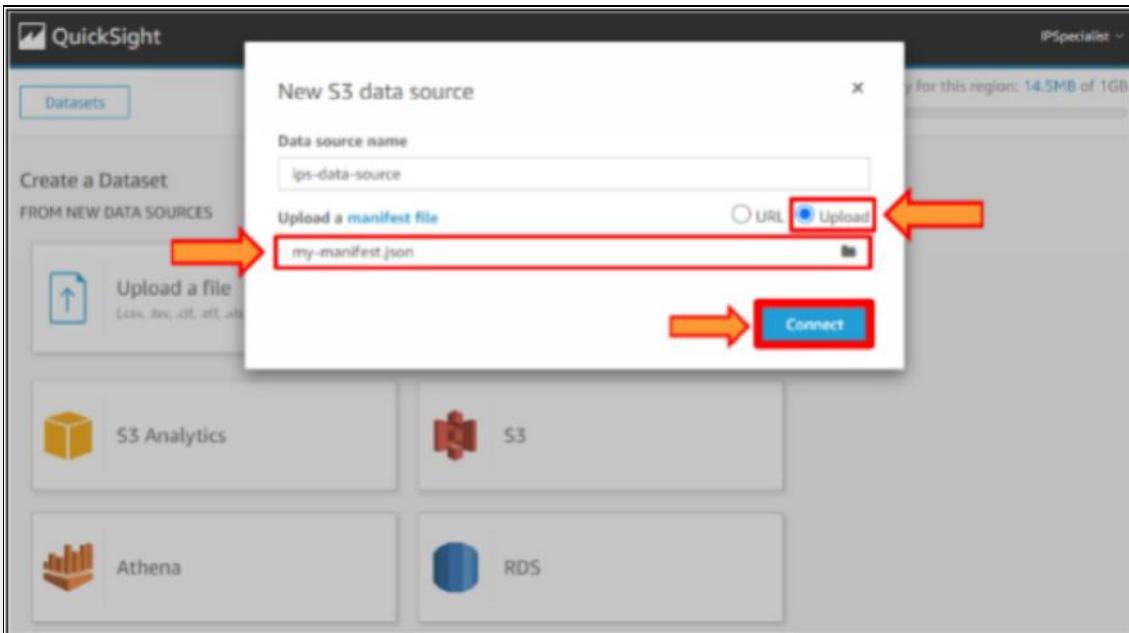


The screenshot shows the Visual Studio Code interface with the following details:

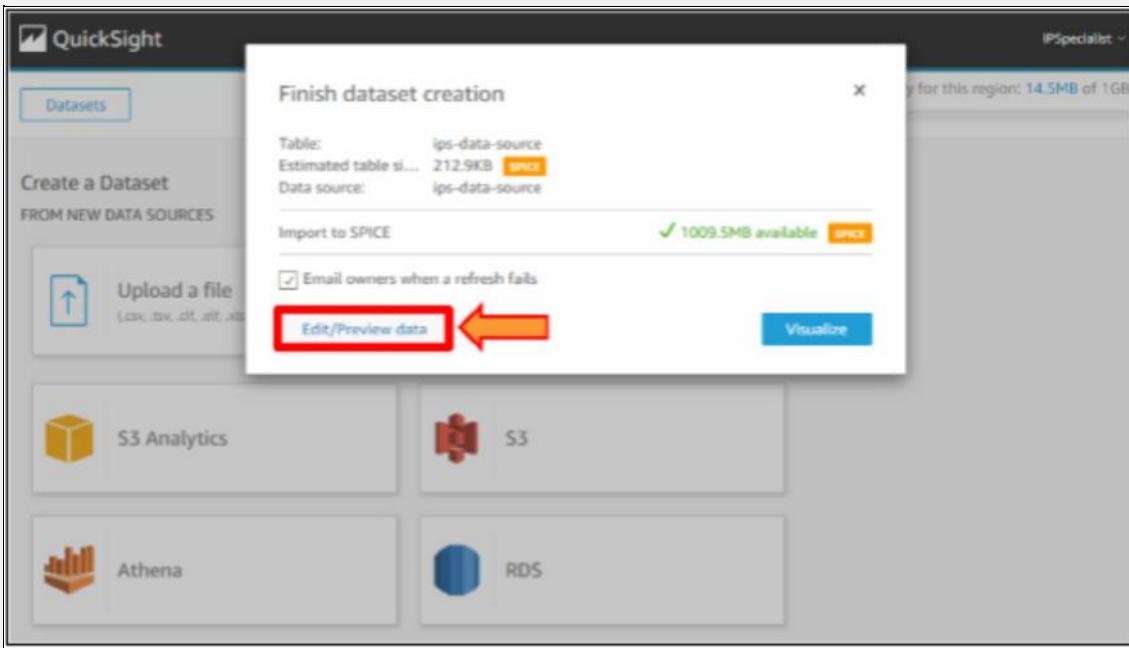
- File: my-manifest.json
- Path: C:\Users\Muhammad Usman Khan\Desktop\Chapter5\my-manifest.json
- Code content:

```
1  {
2      "fileLocations": [
3          {
4              "URIPrefixes": [
5                  "s3://ips-s3-bucket/" ←
6              ]
7          }
8      ],
9      "globalUploadSettings": {
10         "format": "CSV",
11         "delimiter": ",",
12         "textqualifier": """",
13         "containsHeader": "true"
14     }
15 }
```
- Bottom status bar: In 15, Col 2 Spaces: 4 UTF-8 LF JSON ⚡ ⚡

20. Select **Upload**. Then, upload the **my-manifest.json** file.
21. Click on the **Connect** button.



22. Click on the **Edit and Preview** button.



23. The data will be uploaded.

Dataset Name: ips-data-source

Fields: All fields included

Add calculated field

Search fields

Focus: All fields

Select: All | None

- car
- year
- engine_hp
- avg_mpg

Excluded fields: No fields excluded

Filters: No filters applied

Query mode: SPICE

1009.5MB of remaining

Manage Save & visualize Save Cancel

Auto-preview

Add data

ips-data-source

car	year	engine_hp	avg_mpg	price	salesman	deal
String	Integer	Integer	Decimal	Integer	Integer	S
Corvette	2011	335	22.5	46135	2	B
Corvette	2011	300	23.5	40650	2	U
Corvette	2011	300	24.0	36350	2	U

24. Click on the Save & Visualize button.

The screenshot shows the QuickSight interface with the 'ips-data-source' dataset selected. On the right, there's a preview of the data with three rows of car information. At the top right, there are four buttons: 'Save & visualize' (highlighted with a red arrow), 'Save', and 'Cancel'. Below these buttons is an 'Auto-preview' toggle switch and a zoom slider set to 100%.

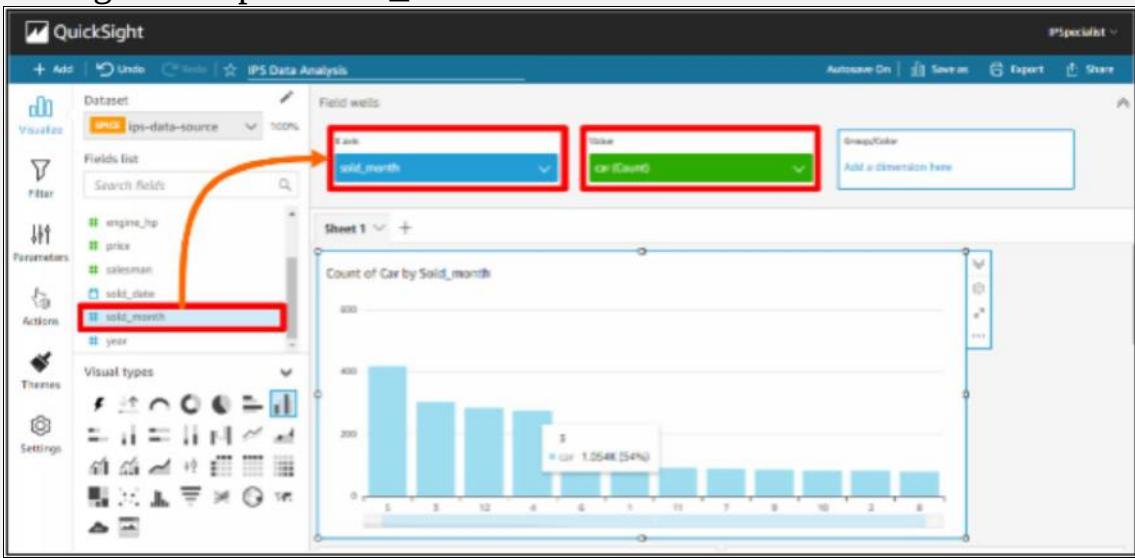
25. Give the name IPS Data Analysis.

The screenshot shows the QuickSight interface with the 'ips-data-source' dataset imported. The title bar at the top says 'IPS Data Analysis'. A message box on the right indicates 'Import complete: 100% success, 1961 rows were imported to SPICE, 0 rows were skipped'. The left sidebar shows various analysis tools like 'Visualize', 'Filter', 'Parameters', 'Actions', 'Themes', and 'Settings'. The main area is titled 'AutoGraph' and says 'Choose 1 or more fields and let QuickSight choose the most appropriate chart'.

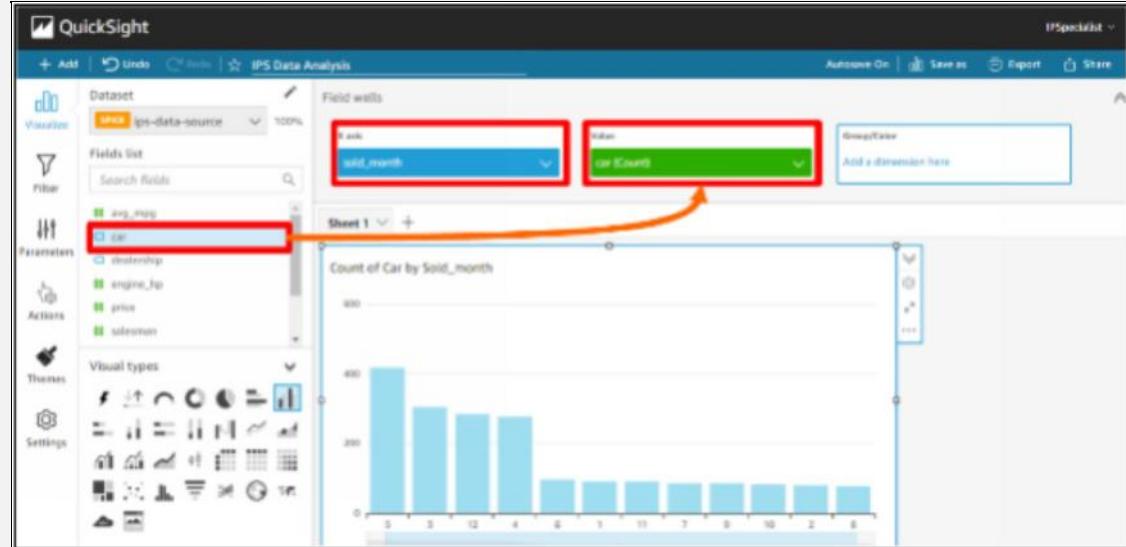
26. Click on the Vertical bar chart.

The screenshot shows the QuickSight interface with the 'ips-data-source' dataset imported. The title bar says 'IPS Data Analysis'. A message box on the right indicates 'Import complete: 100% success, 1961 rows were imported to SPICE, 0 rows were skipped'. In the 'Visual types' section of the left sidebar, there are several chart icons. One specific icon, which is a vertical bar chart, is highlighted with a red arrow. A callout box next to it says 'Vertical bar chart: requires 1 dimension in Xaxis'.

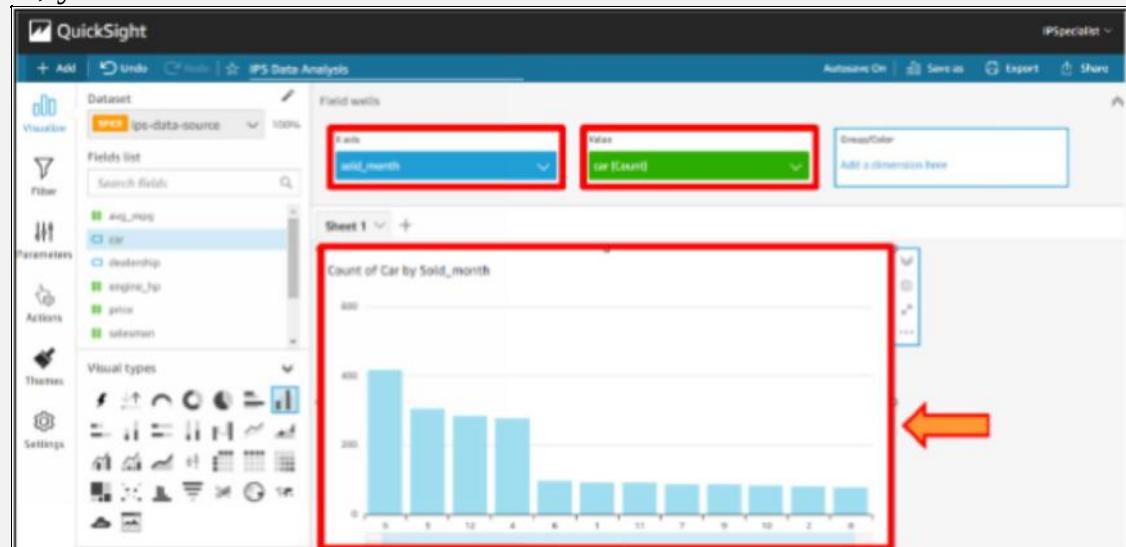
27. Drag and drop the **sold_month** feature onto the X-Axis box.



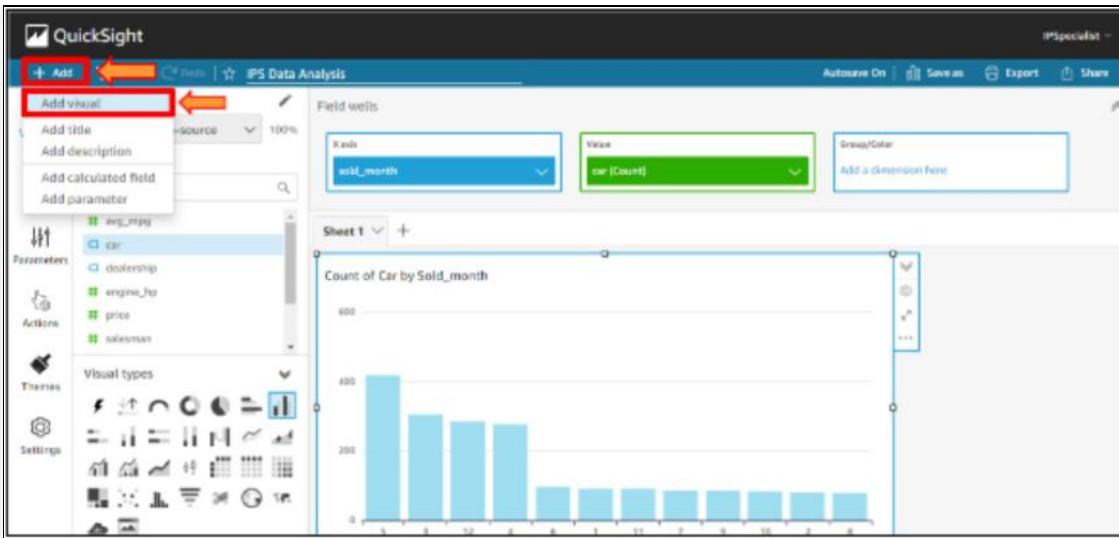
28. Drag and drop the **Car** feature onto the **Value** box.



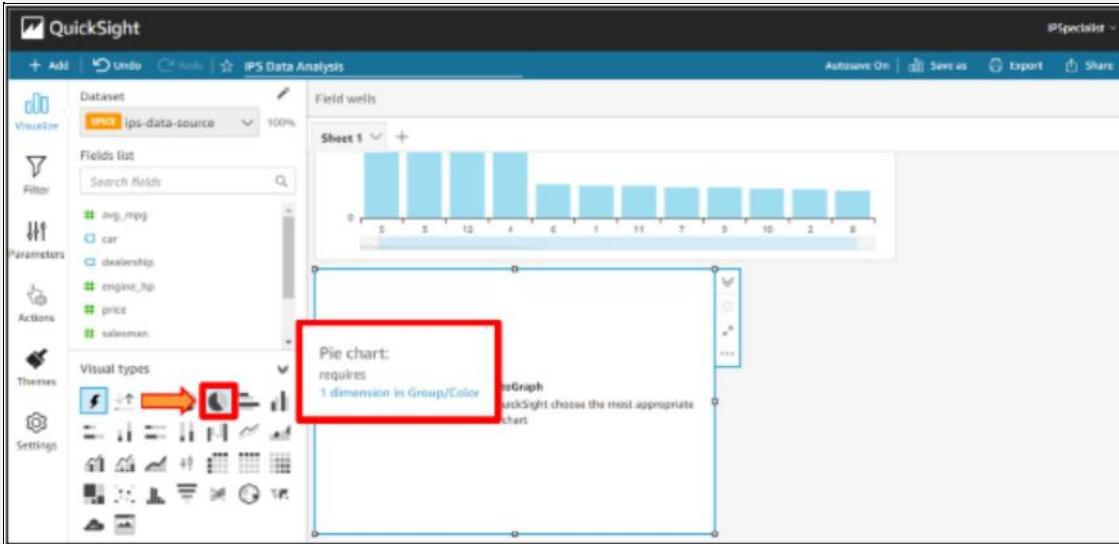
Hence, you will see the visualization of data.



29. Click on the **Add** button. Then, click on the **Add Visual**.

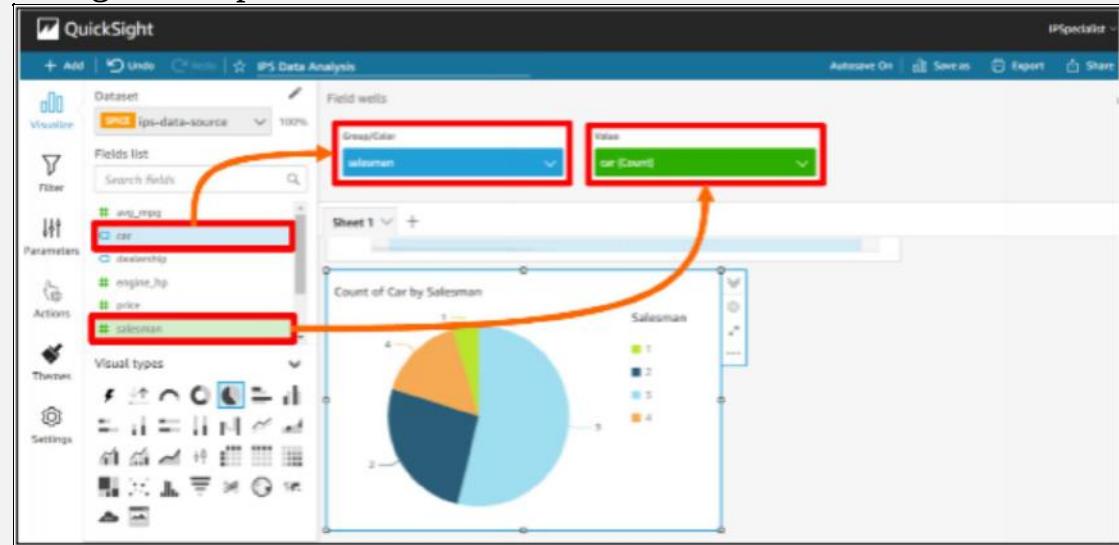


30. Click on the **Pie Chart**.

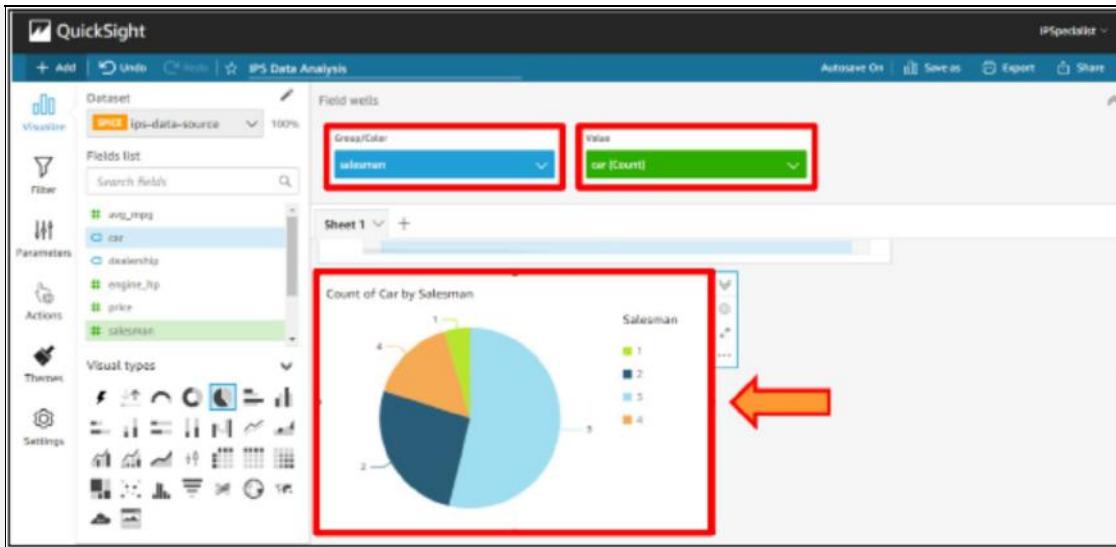


31. Drag and drop the **Salesman** feature onto the **Group/Colour** box.

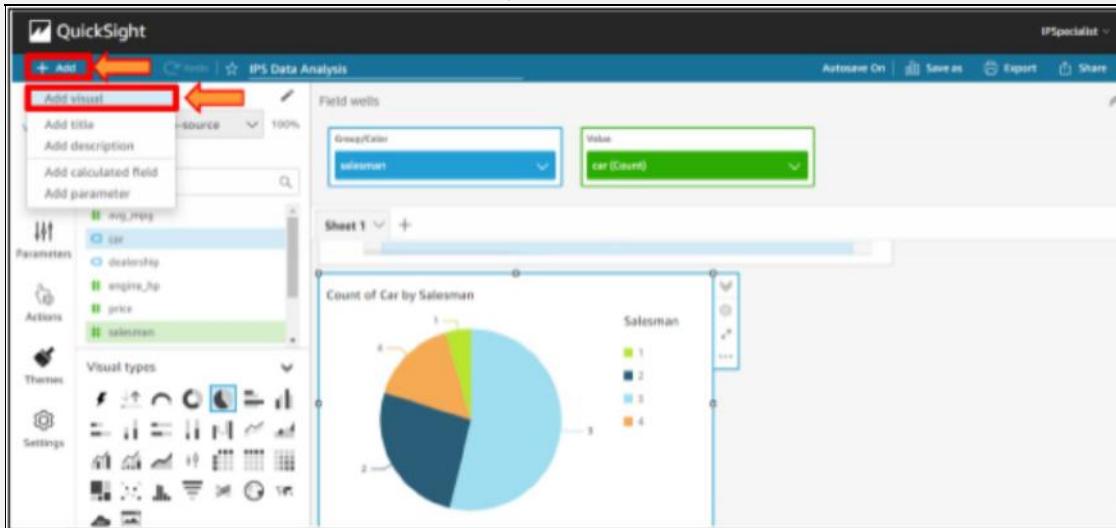
32. Drag and drop the **Car** feature onto the **Value** box.



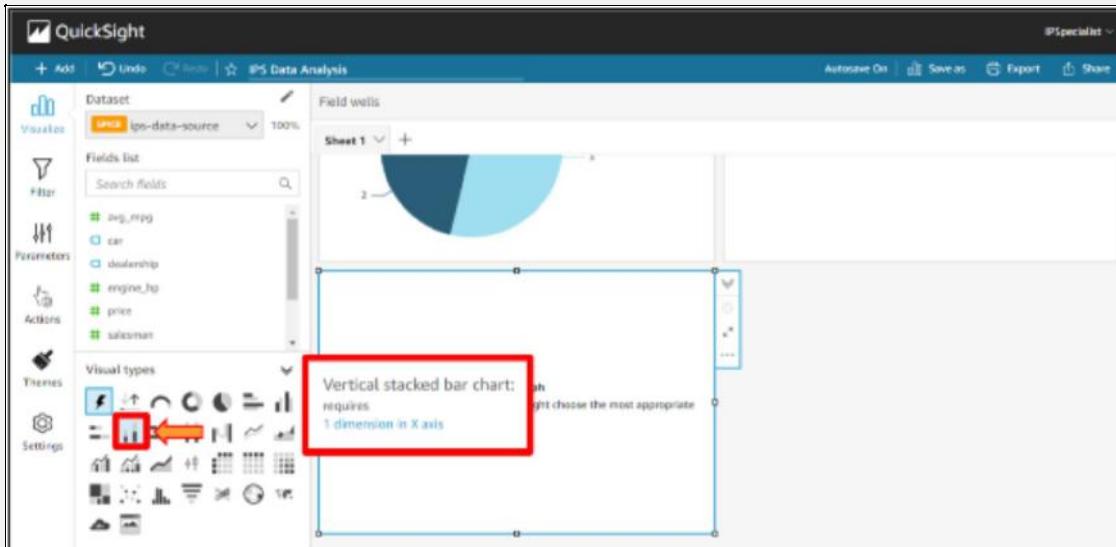
Hence, you will see the visualization of the Pie Chart.



33. Click on the **Add** button. Then, click on the **Add Visual**.



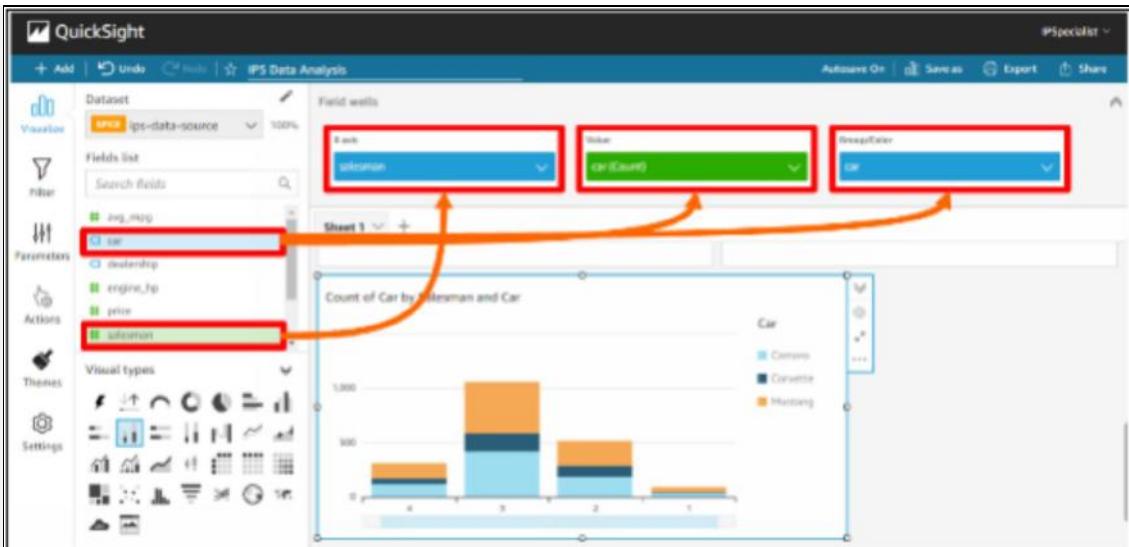
34. Click on the **Vertical bar chart**.



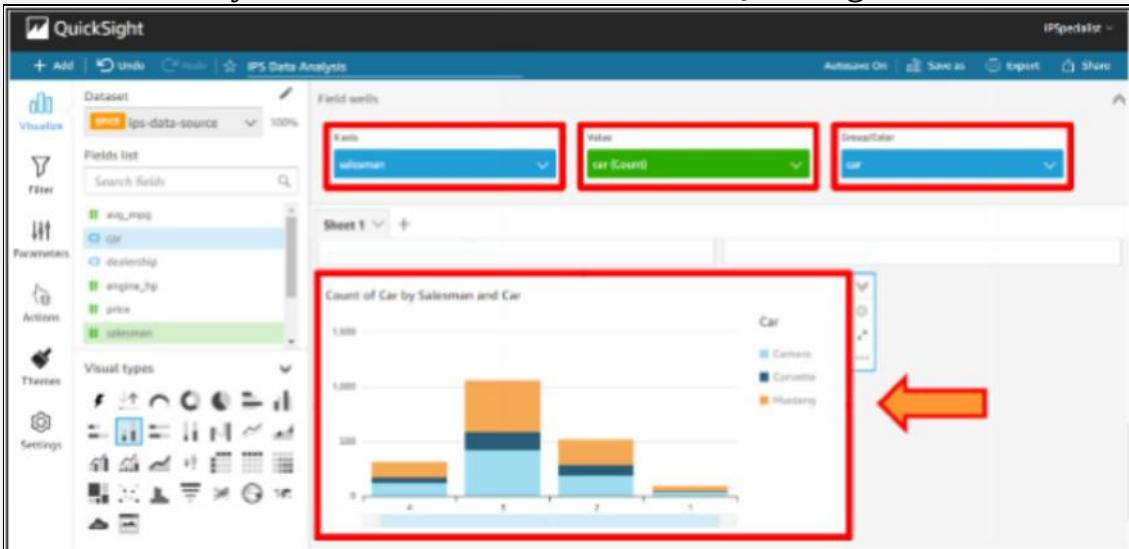
35. Drag and drop the **Salesman** feature onto the **X-Axis** box.

36. Drag and drop the **Car** feature onto the **Value** box.

37. Drag and drop the **Car** feature onto the **Group/Colour** box.



Hence, this is how you can visualize data on AWS QuickSight.



38. Unsubscribe your AWS QuickSight account so that you do not get charged.

Step 3: Visualize Data on AWS SageMaker

1. Click on Services.

The screenshot shows the AWS Management Console homepage. At the top left, there is a red box around the "Services" dropdown menu. The main content area is titled "AWS Management Console" and contains a sidebar titled "AWS services". Under "Recently visited services", it says "Your recently visited AWS services appear here." Below that is a link to "All services". At the bottom of the page, there is a footer bar with links for "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preferences".

2. Select the SageMaker from the Machine Learning.

The screenshot shows the AWS Management Console with the "Machine Learning" section selected. On the left, there is a sidebar with "Favorites" and "Recently visited" sections. The main area is titled "All services" and lists various AWS services. The "Amazon SageMaker" service is highlighted with a red box and a red arrow pointing to it. Other services listed include Compute (EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts), Customer Enablement (AWS IQ Support, Managed Services, Activate for Startups), Machine Learning (Amazon Augmented AI, Amazon CodeGuru, Amazon DevOps Guru, Amazon Comprehend), AWS Cost Management (AWS Cost Explorer, AWS Budgets, AWS Marketplace, AWS Subscriptions, AWS Application Cost Profiler), and Robotics (AWS RoboMaker). At the bottom, there is a footer bar with links for "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preferences".

3. Click on the Notebook Instance on the left-hand side menu.

The screenshot shows the Amazon SageMaker Studio interface. On the left, there is a sidebar with options like "Dashboard", "Search", "Images", "Ground Truth", "Notebook" (which is expanded to show "Notebook instances"), "Lifecycle configurations", "Git repositories", "Processing", "Training", and "Training". The "Notebook instances" option is highlighted with a red box and a red arrow pointing to it. The main content area is titled "MACHINE LEARNING" and features the heading "Amazon SageMaker" with the subtext "Build, train, and deploy machine learning models at scale". Below that, it says "The quickest and easiest way to get ML models from idea to production." At the bottom, there is a footer bar with links for "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preferences".

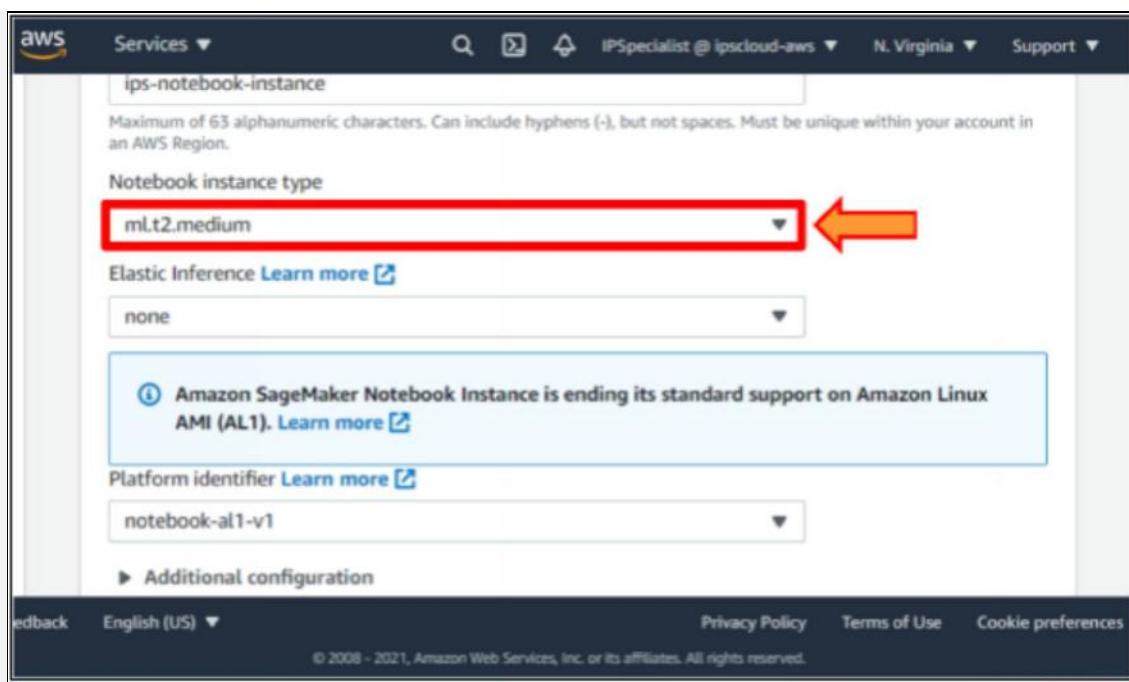
4. Click on the **Create notebook instance** button.

The screenshot shows the 'Notebook instances' page in the AWS Management Console. At the top right, there is a prominent orange 'Create notebook Instance' button with a red arrow pointing to it. Below the button is a search bar labeled 'Search notebook instances'. Underneath the search bar is a table header with columns: Name, Instance, Creation time, Status, and Actions. A message at the bottom of the table says 'There are currently no resources.'

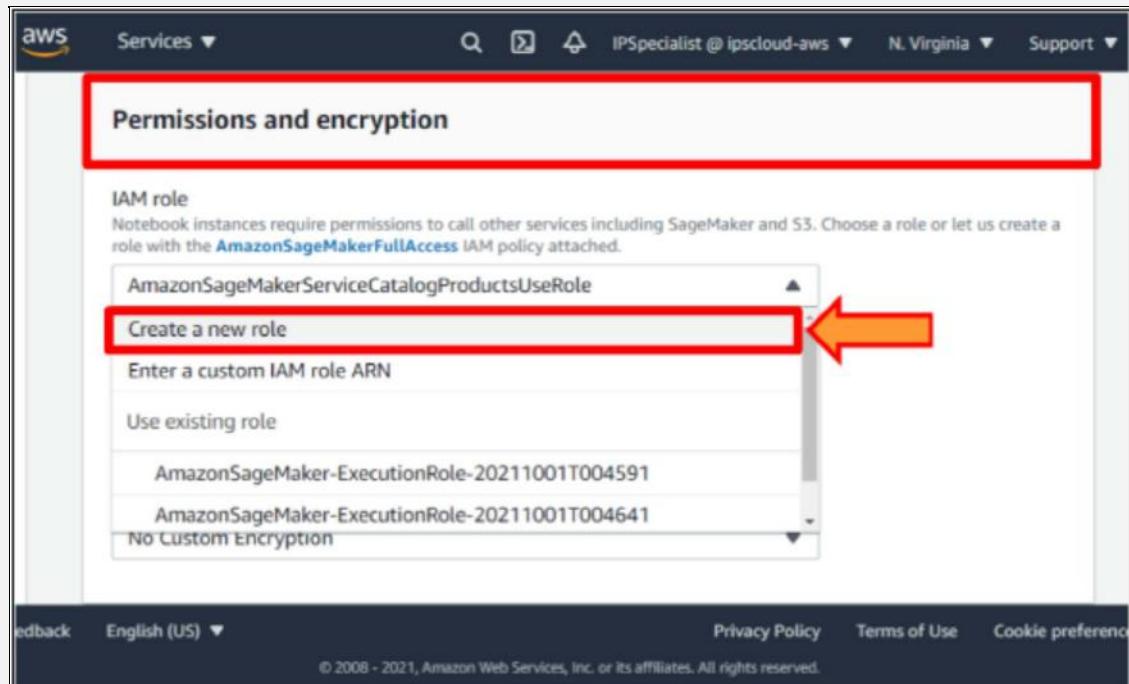
5. Give the name **ips-notebook-instance**.

The screenshot shows the 'Create notebook instance' settings page. The 'Notebook instance name' field contains the value 'ips-notebook-instance', which is highlighted with a red box and a red arrow. Below the field, a note states: 'Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.' At the bottom of the page, there is a 'Notebook instance type' section and a footer with standard AWS links.

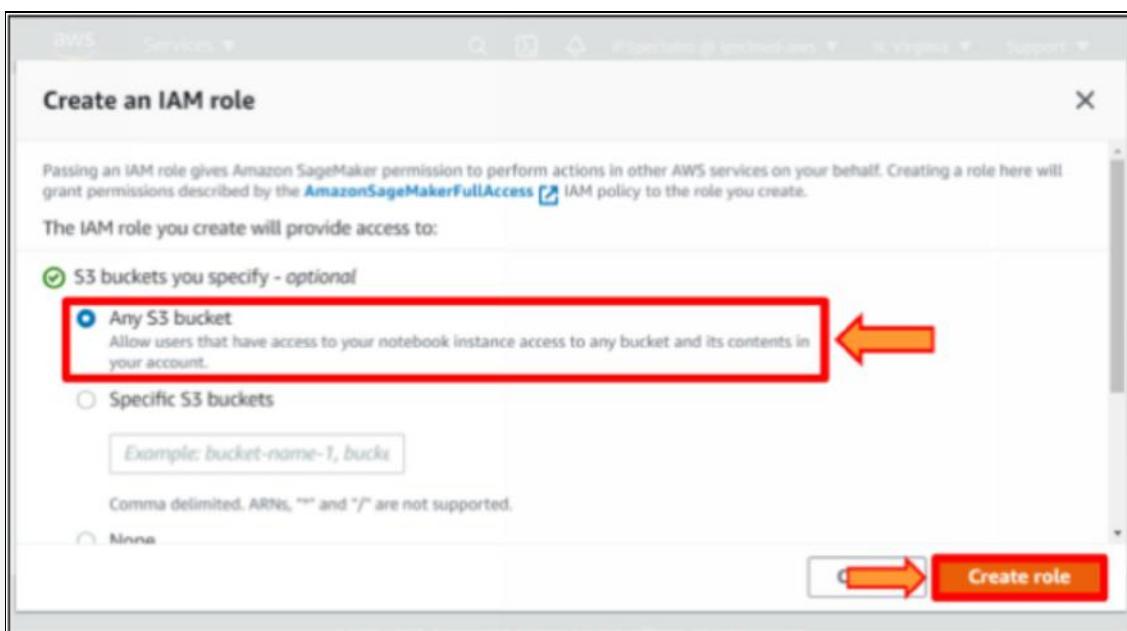
6. Select the **ml.t2.medium** Notebook instance type.



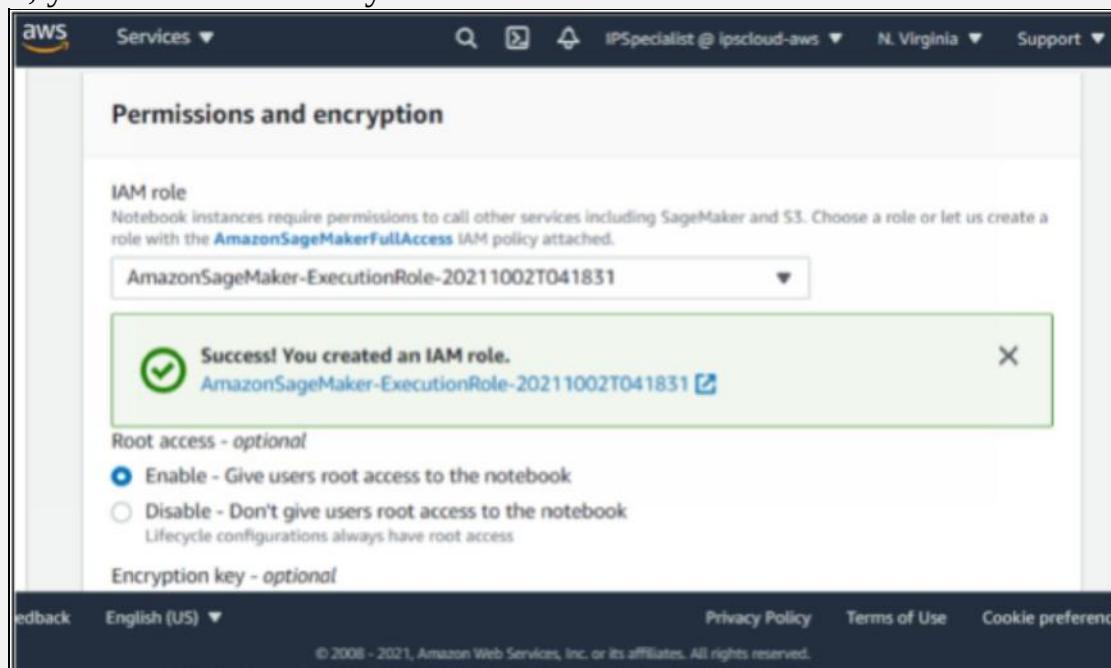
7. Scroll down under permission and encryption.
8. Click on the **Create a new role**.



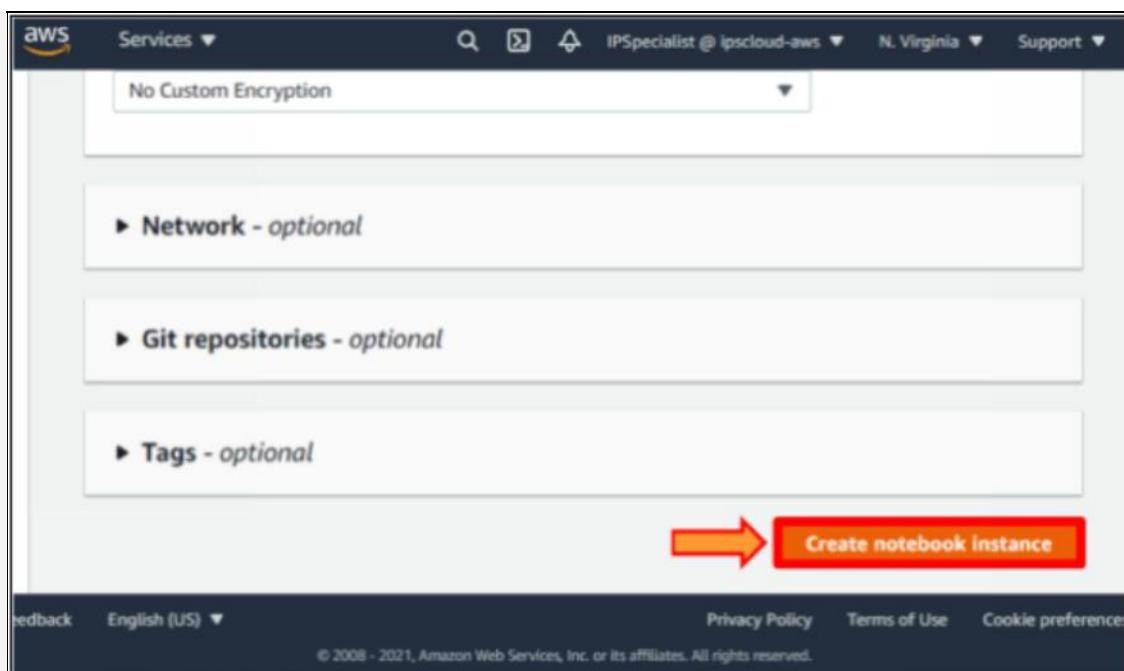
9. Select Any S3 Bucket.
10. Click on the **Create role** button.



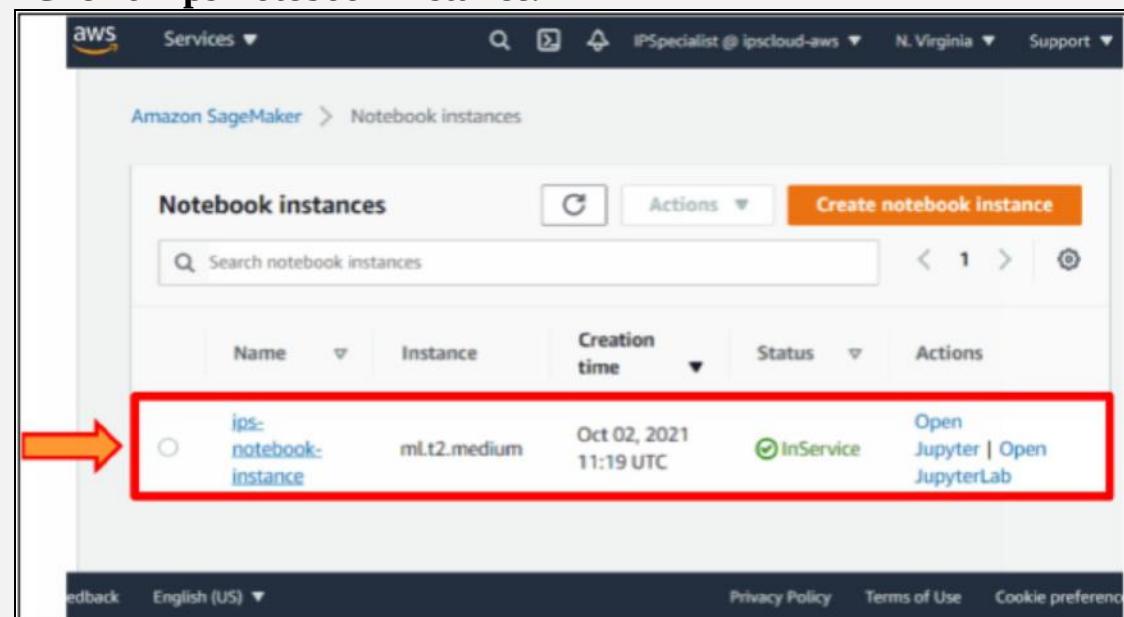
Hence, you have successfully created an IAM role.



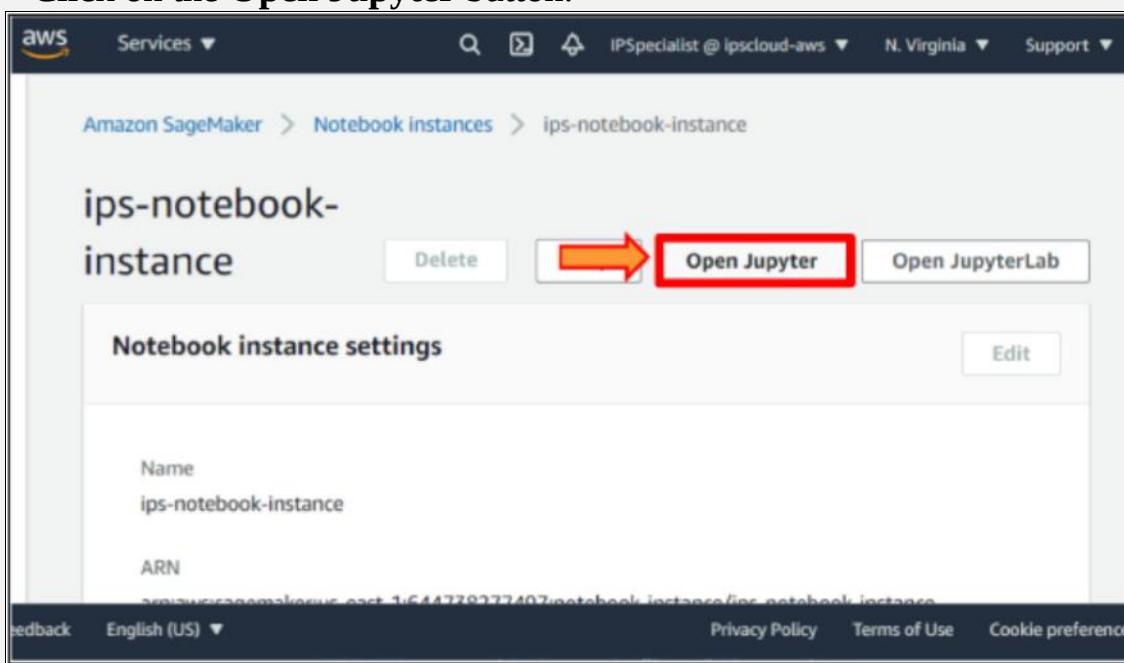
11. Scroll down. Click on the **Create notebook instance** button. It will take a few minutes.



12. Click on **ips-notebook-instance**.

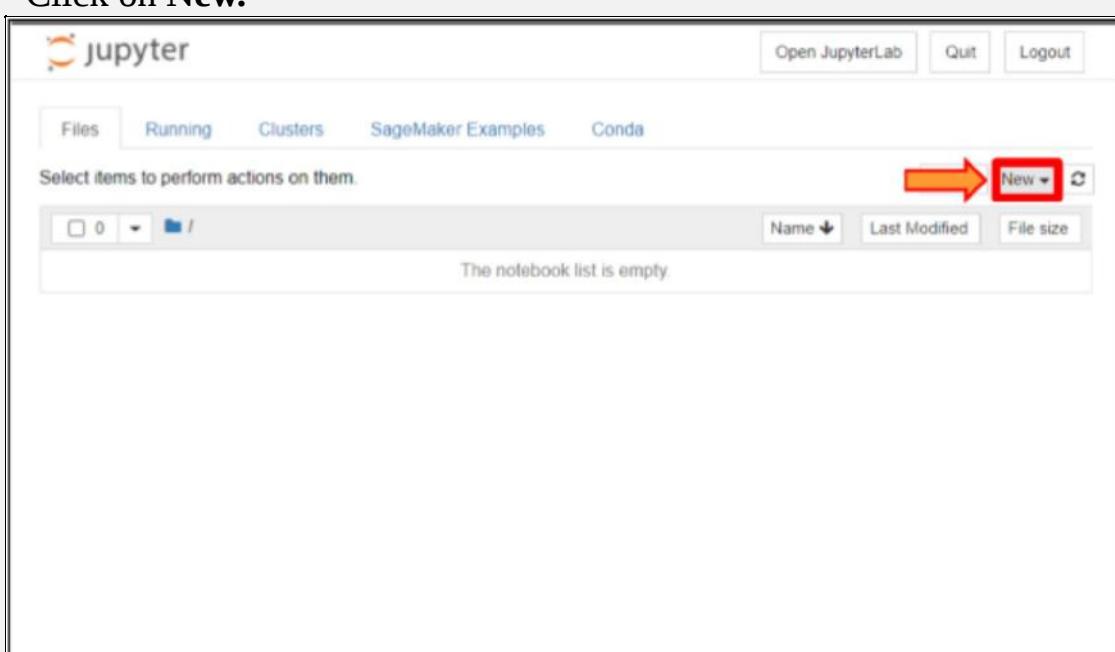


13. Click on the **Open Jupyter** button.



14.

Click on New.

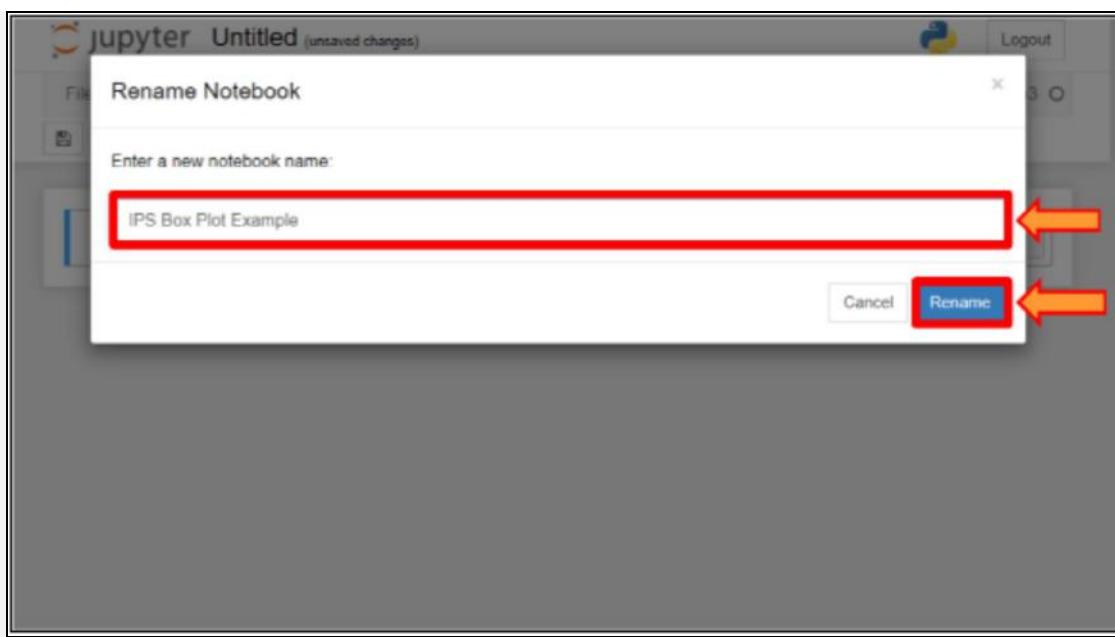


15.

Click on **Conda_python3**.

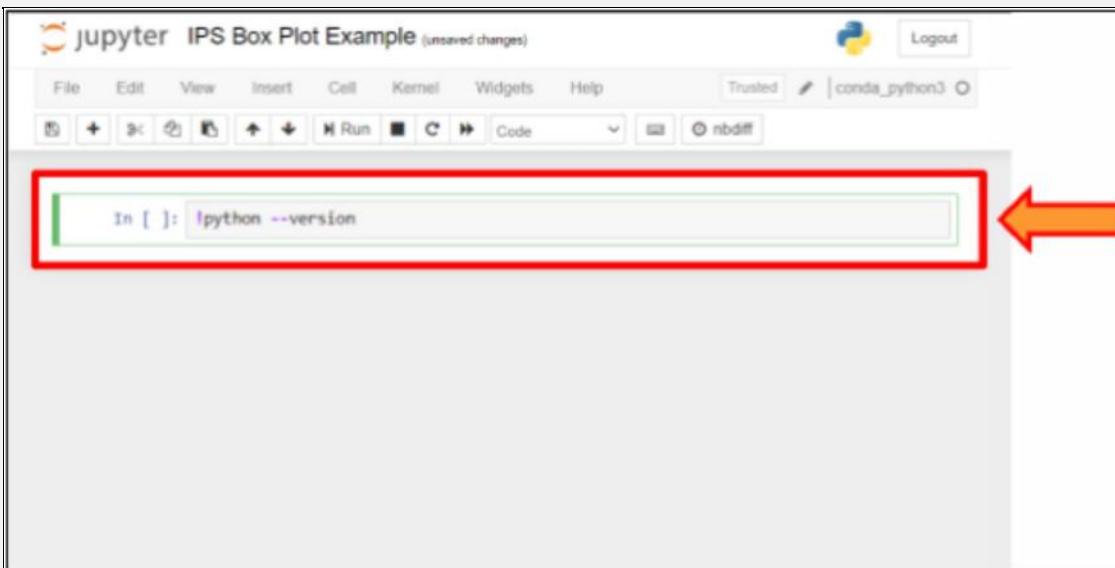


16. Give the Jupyter notebook name **IPS Box Plot Example**. Then, click on the **Rename** button.

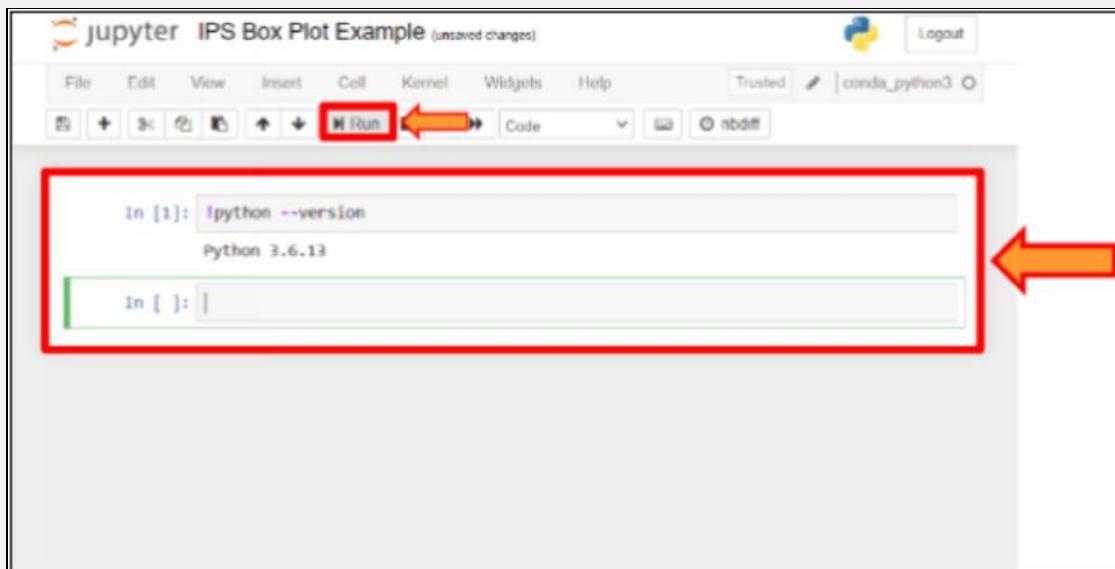


17. Copy and paste the below code on the cell to check the Python version.

```
! Python --version
```

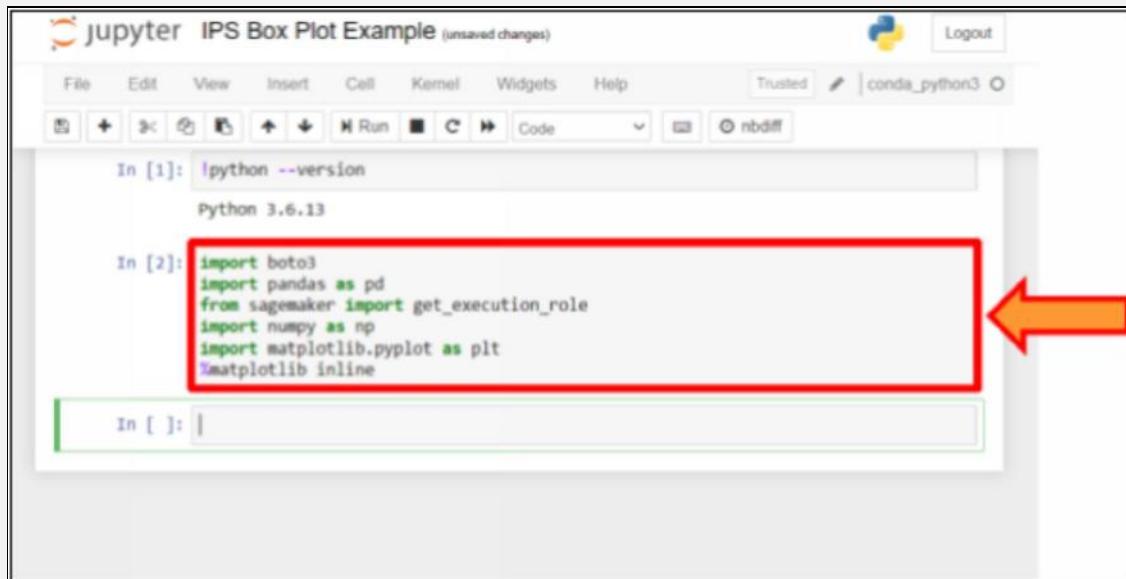


18. Click on the **Run** button to execute the cell or press **Shift + Enter** to run the cell.



19. Copy and paste the below code to **Import the python libraries**. Then, press **Shift + Enter** to run the cell.

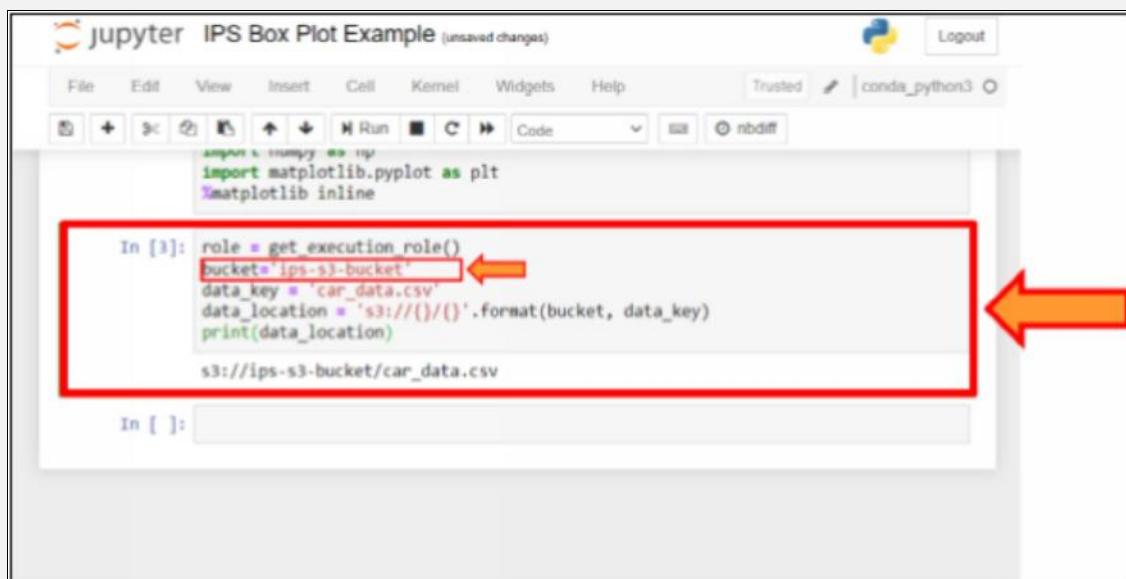
```
import boto3
import pandas as pd
from sagemaker import get_execution_role
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```



```
jupyter IPS Box Plot Example (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted conda_python3 nbdiff
In [1]: !python --version
Python 3.6.13
In [2]: import boto3
import pandas as pd
from sagemaker import get_execution_role
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

20. Copy and paste the below code to **Get the car data from S3**. Also, add the **ips-s3-bucket name**. Then, press **Shift + Enter** to run the cell.

```
role = get_execution_role()
bucket='<YOUR_BUCKET_NAME_HERE>'
data_key = 'car_data.csv'
data_location = 's3://{}{}'.format(bucket, data_key)
print(data_location)
```



```
jupyter IPS Box Plot Example (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted conda_python3 nbdiff
In [1]: !python --version
Python 3.6.13
In [2]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
In [3]: role = get_execution_role()
bucket='ips-s3-bucket'
data_key = 'car_data.csv'
data_location = 's3://{}{}'.format(bucket, data_key)
print(data_location)
s3://ips-s3-bucket/car_data.csv
```

21. Copy and paste the code below to **read a CSV file into a data frame** and **see the first five rows of the data frame**. Then, press **Shift + Enter** to run the cell.

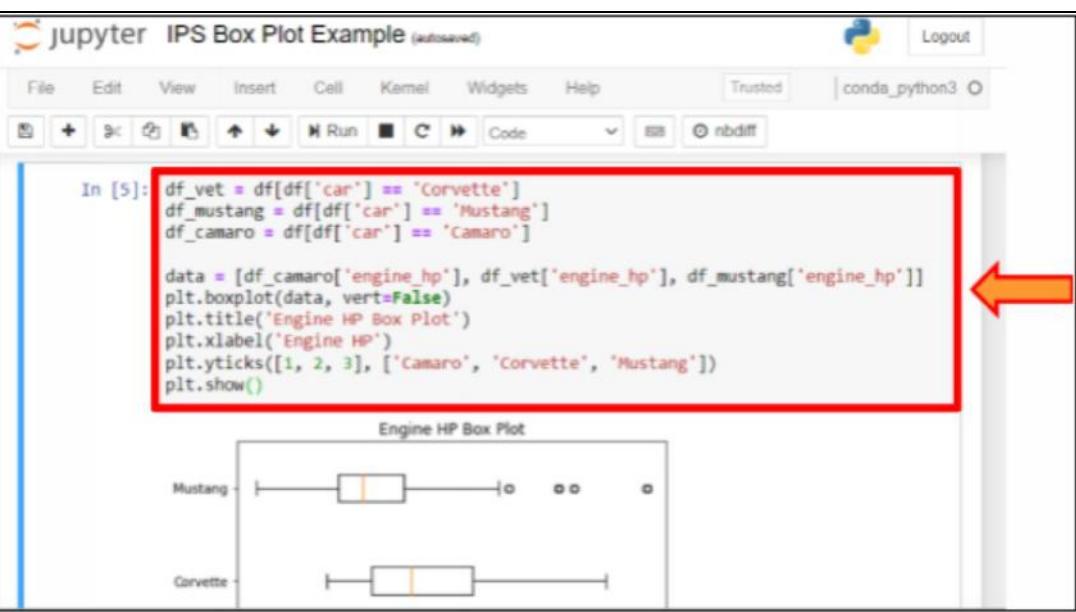
```
df = pd.read_csv(data_location)  
df.head()
```

The screenshot shows a Jupyter Notebook interface. The top bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', 'Help', 'Trusted', and 'Logout'. Below the menu is a toolbar with various icons. The main area has two sections: 'In [4]' and 'Out[4]'. The 'Out[4]' section contains a table with 5 rows of data. A large red box highlights this table, and an orange arrow points from the right side of the image towards the red box. The table data is as follows:

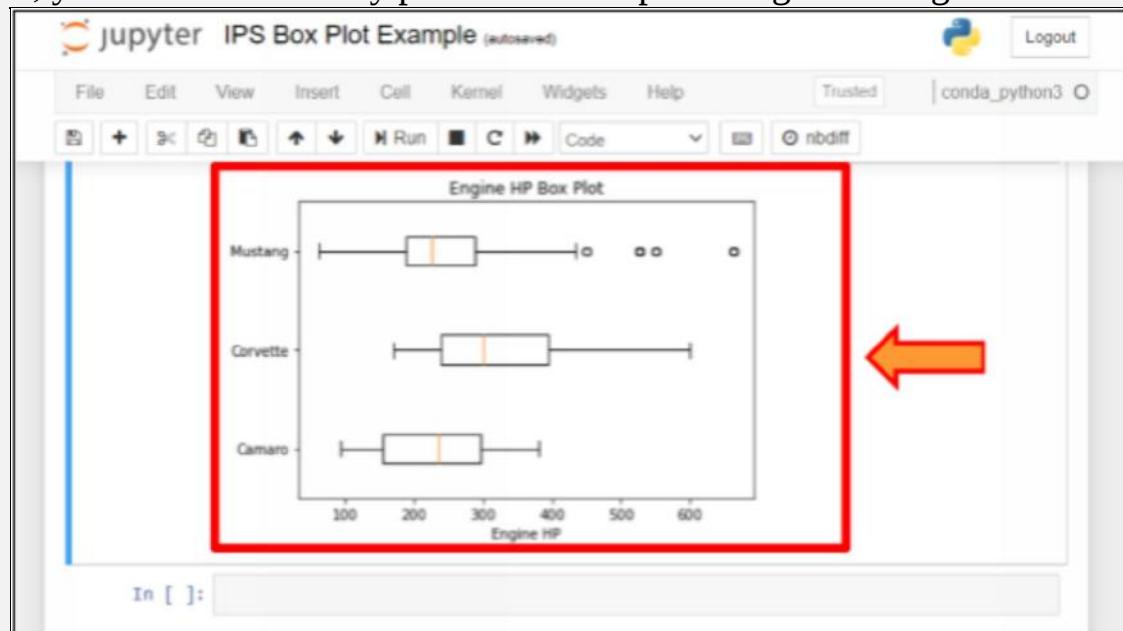
	car	year	engine_hp	avg_mpg	price	salesman	dealership	sold_date	sold_month
0	Corvette	2011	335.0	22.5	46135	2	Big Bobs	2012-05-06	5
1	Corvette	2011	300.0	23.5	40650	2	Uptown Cars	2011-06-16	5
2	Corvette	2011	300.0	24.0	36350	2	Uptown Cars	2013-07-31	7
3	Corvette	2011	230.0	23.0	29450	2	Uptown Cars	2014-07-06	7
4	Corvette	2011	230.0	23.0	34500	2	Uptown Cars	2013-05-20	5

22. Copy and paste the below code to the **plot box plot**. Then, press **Shift + Enter** to run the cell.

```
df_vet = df[df['car'] == 'Corvette']  
df.mustang = df[df['car'] == 'Mustang']  
df.camaro = df[df['car'] == 'Camaro']  
  
data = [df.camaro['engine_hp'], df_vet['engine_hp'],  
df.mustang['engine_hp']]  
plt.boxplot(data, vert=False)  
plt.title('Engine HP Box Plot')  
plt.xlabel('Engine HP')  
plt.yticks([1, 2, 3], ['Camaro', 'Corvette', 'Mustang'])  
plt.show()
```



Hence, you have successfully plotted the box plot using AWS SageMaker.



23. The Jupyter Notebook is provided in the following Github link:

https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter5/box-plot-example.ipynb

```
In [5]: python --version
Python 3.6.5 :: Anaconda custom (64-bit)

Importing the important libraries

In [6]: import boto3
import pandas as pd
from sagemaker import get_execution_role
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

Getting the car data from S3

In [7]: role = get_execution_role()
bucket='<YOUR_BUCKET_NAME_HERE>'
data_key = 'car_data.csv'
data_location = 's3://{}{}'.format(bucket, data_key)
print(data_location)

s3://car-data-analysis-acg/car_data.csv
```

So this is how you can visualize the data using AWS SageMaker.

24. After completing the lab, delete all the AWS services; Therefore, you do not get charged.

Lab 07: Modeling

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	QuickSight, S3, SageMaker, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

AWS QuickSight

Amazon QuickSight is a cloud-based business analytics solution that enables all employees to create visualizations, do ad-hoc analysis, and quickly get business insights from their data, anytime and on any device. Upload CSV and Excel files; connect to SaaS apps like Salesforce; connect to on-premises

databases like SQL Server, MySQL, and PostgreSQL; and locate your AWS data sources in real-time, including Amazon Redshift, Amazon RDS, Amazon Aurora, Amazon Athena, and Amazon S3. QuickSight allows businesses to extend their business analytics capabilities to thousands of users while still providing quick and responsive query processing. QuickSight enables businesses to expand their business analytics capabilities to thousands of users. It uses a powerful in-memory engine to offer quick and responsive query performance (SPICE).

AWS SageMaker

Amazon SageMaker is a fully managed service that allows any developer or data scientist to quickly create, train, and deploy Machine Learning (ML) models. SageMaker makes it easy to create high-quality models by removing the heavy lifting from each phase of the machine learning process.

AWS Simple Storage Service S3

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 is a web service that allows you to store and retrieve an infinite quantity of data from any place and at any time. You may quickly create projects that integrate cloud-native storage using this service because Amazon S3 is easily customizable, and you only pay for what you use. You can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. Build a simple FTP system or a complex web application like the Amazon.com retail website; read the same piece of data a million times or only for emergency disaster recovery; store whatever type and amount of data you desire.

Case Study Enterprise EMRs – [Savana](#)

Background

Savana offers processing and analytic services for Electronic Medical Records (EMRs). With more than 180 hospitals spread over 15 nations, Savana, a Madrid-based company, runs one of the world's largest Artificial Intelligence (AI), multicentric, and multilingual research networks.

Savana, a company, established in Madrid, assists healthcare providers in maximizing the value of their EMRs for research. To produce pertinent results for healthcare and life science providers looking into disease prediction and therapy, it blends research-grade methods with Natural Language Processing (NLP) and predictive analytics. It can process the massive amounts of data needed to run machine learning algorithms using Amazon Web Services (AWS). Along with meeting regional regulatory requirements, it can scale its infrastructure to support its rapid growth across international markets, including the US, Latin America, and Western Europe. Compared to the prior technology, it has shortened study processing times by 25% and IT expenditures by up to 90%.

Business Challenge

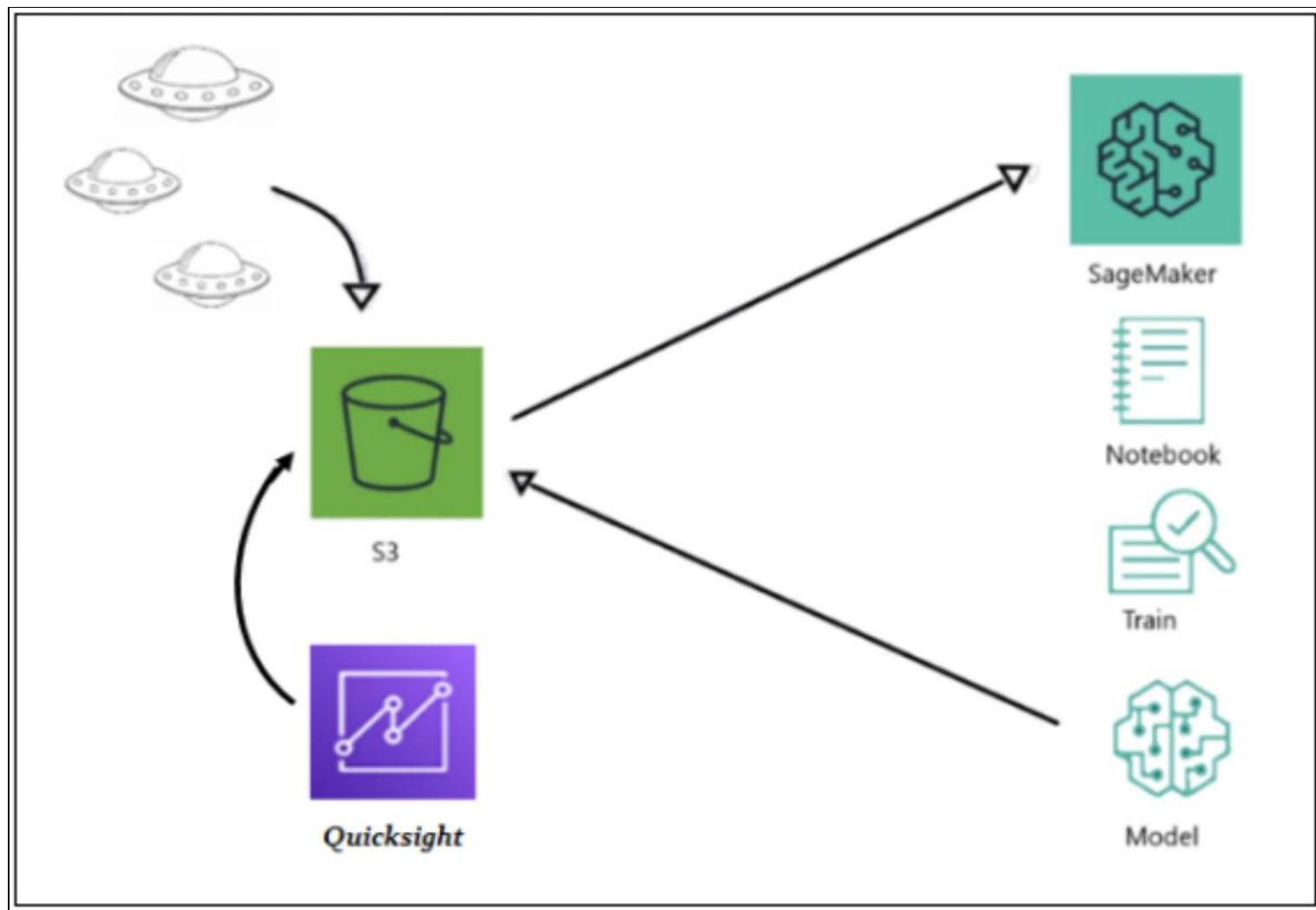
Through an automated data pipeline fed by hospitals looking into disease prediction and therapy, Savana processes billions of EMRs and data points. To properly complete this, it is required to have a reliable architecture, handle the volume of data, and have the computing power required to run sophisticated algorithms while maintaining the greatest levels of security. The company's infrastructure

must scale as it expands to meet regulatory requirements for data management, storage, and transmission worldwide.

Proposed Solution

The solution is for you to use AWS services to automate all the tasks. You create an AWS SageMaker hyperparameter tuning job with different ranges of values for the hyperparameter to find the best configuration, which minimizes the validation: objective_loss metric. The reason that you use this metric is that it is used in multi-classification problems. This metric measures the performance of the classification model, and what it does is repeatably calculate the difference between the values that the model predicts and the actual values of a label. Hence, every time it passes over data and makes predictions, it recalculates the objective loss and tries to minimize this value overall. Hence, that is the task here. AWS recommends that you minimize this value when using it as our objective metric.

Lab Diagram



Implementation Steps

1. Create an S3 bucket.
2. Create Notebook instances.
3. Create a QuickSight account.

Solution

Step 1: Create an S3 Bucket

1. Log in to AWS Management Console.

The screenshot shows the AWS Management Console homepage. On the left, there's a sidebar titled "AWS services" with sections for "Recently visited services" (Amazon AppStream) and "All services". Below that is a "Build a solution" section with links to "Launch a virtual machine", "Build a web app", and "Build using virtual servers". On the right, there are two boxes: "Stay connected to your AWS resources on the go" (with a link to "AWS Device Hub") and "Explore AWS" (with links to "Amazon Location Service" and "Free AWS Training"). At the bottom of the sidebar, there are links for "AWS Lambda", "Amazon CloudWatch Metrics", and "Amazon CloudWatch Metrics Insights".

2. Search for S3 in the search bar.

The screenshot shows the search results for 's3'. The search bar at the top contains 's3'. The results are categorized into "Services (7)", "Features (10)", and "Documentation (317,648)". The "Documentation" category has a red arrow pointing to it. The "S3" service card is highlighted with a red box and labeled "Scalable Storage in the Cloud".

3. Click on Create bucket.

The screenshot shows the Amazon S3 "Buckets" list. It displays 8 buckets. There are buttons for "Copy ARN" and "Empty". A large red arrow points to the "Create bucket" button, which is highlighted with a red box.

4. Define bucket name.

Create bucket

Buckets are containers for data stored in S3. Learn more 

General configuration

Bucket name

ml-labs-acg



Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming 

Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

5. Click on **Create bucket**.

No tags associated with this bucket.

[Add tag](#)

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#) 

Server-side encryption

Disable

Enable

► Advanced settings

 After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.



Create bucket 

6. Click on **Create folder**.

ml-labs-acg

Objects Properties Permissions Metrics Management Access Points

Objects (2)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more [\[?\]](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	encrypted_data/	Folder	-	-	-
<input type="checkbox"/>	ARCHIVE/	Folder	-	-	-

7. Define folder name.

8. Click on **Create folder**.

configuration to upload an empty folder and specify the appropriate settings.

Folder

Folder name /

Folder names can't contain "/". See rules for naming [\[?\]](#)

Server-side encryption

The following settings apply only to the new folder object and not to the objects contained within it.

Server-side encryption

Disable

Enable

Create folder

9. Click on the name of the object.

Amazon S3 > ml-labs-acg

ml-labs-acg

Objects Properties Permissions Metrics Management Access Points

Objects (3)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more [\[?\]](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	ufo_dataset/	Folder	-	-	-
<input type="checkbox"/>	encrypted_data/	Folder	-	-	-

10. Click on **Upload**.

Objects | Folder properties

Objects (0)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Delete Actions Create folder Upload

Find objects by prefix:

Name Type Last modified Size Storage class

No objects

You don't have any objects in this folder.

Upload



11. Click on **Upload**.

 We recommend that you enable Bucket Versioning to help protect against unintentionally overwriting or deleting objects. [Learn more](#)

Enable Bucket Versioning

Additional upload options

Step 2: Create Notebook Instances

12. Search **Amazon SageMaker** in the search bar.

The screenshot shows the AWS Services search interface. The search bar at the top contains "SageMaker". Below the search bar, the results are displayed under the heading "Services". There are two services listed: "Amazon SageMaker" and "AWS Glue DataBrew". The "Amazon SageMaker" card is highlighted with a red box and has a large red arrow pointing towards it from the right side. The "AWS Glue DataBrew" card is partially visible below it.

13. Click on **Notebook instances** under dashboards.

14. Click on **Create Notebook instances**.

The screenshot shows the "Notebook instances" section of the Amazon SageMaker console. On the left, there is a sidebar with navigation links: "Dashboard", "Search", "SageMaker Domain Studio", "Images", "Ground Truth", and "Notebook". The "Notebook" link is highlighted with a red box and has a red arrow pointing towards it from the bottom left. In the main content area, there is a table titled "Notebook instances" with columns: "Name", "Instance", "Creation time", "Status", and "Actions". A search bar above the table also has a red box around it and a red arrow pointing towards it from the bottom right. At the bottom right of the table, there is a prominent orange button labeled "Create notebook instance", which also has a red box around it and a red arrow pointing towards it from the bottom right.

15. Define Notebook instance name.

Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. [Learn more](#)

Notebook instance settings

Notebook instance name

ml-labs-notebook-instance



Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type

ml.t2.medium

Elastic Inference [Learn more](#)

none

▶ Additional configuration

16. Click on Create Notebook Instance.

Permissions and encryption

IAM role

Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the [AmazonSageMakerFullAccess](#) IAM policy attached.

Choose an IAM role

Root access - optional

- Enable - Give users root access to the notebook
- Disable - Don't give users root access to the notebook

Lifecycle configurations always have root access

Encryption key - optional

Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.

No Custom Encryption

▶ Network - optional

▶ Git repositories - optional

▶ Tags - optional



Create notebook instance

17. Click on Open Jupyter.

Success! Your notebook instance is being created.
Open the notebook instance when status is InService and open a template notebook to get started.

[View details](#)

X

Amazon SageMaker > Notebook Instances

Notebook instances

Actions

Create notebook instance

< 1 >

@

Name	Instance	Creation time	Status	Actions
mi-labs-notebook-instance	ml.t2.medium	May 16, 2019 15:38 UTC	InService	Open Jupyter Open JupyterLab



[Open Jupyter](#) | [Open JupyterLab](#)

18. Click on Upload.

The screenshot shows the Jupyter interface with a red arrow pointing to the 'Upload' button in the top right corner of the file list area.

19. Select an item to perform actions.

The screenshot shows the Jupyter interface with a red arrow pointing to the file 'ufo-modeling-lab.ipynb' in the list, which is currently selected.

20. Import library.

The screenshot shows a Jupyter notebook cell with the following Python code:

```
In [1]: import pandas as pd
import numpy as np
from datetime import datetime

import boto3
from sagemaker import get_execution_role
import sagemaker.amazon.common as smac
```

A large red box highlights the entire code block, and a red arrow points to the left side of the code block.

21. Import library.

jupyter ufo-modeling-lab Last Checkpoint: 20 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted conda_python3

```
In [2]: role = get_execution_role()
bucket = 'ml-labs-acv'
prefix = 'ufo_dataset'
data_key = 'ufo_fullset.csv'
data_location = 's3://{}//{}//{}'.format(bucket, prefix, data_key)

df = pd.read_csv(data_location, low_memory=False)
```

```
In [3]: df.head()
```

	reportedTimestamp	eventDate	eventTime	shape	duration	witnesses	weather	firstName	lastName	latitude	longitude	sighting	physicalEvidence
0	1980-11-29T10:01:48.297Z	1980-11-28	03:17	oval	71	1	snow	Munel	Barrett	28.039167	-81.950000	Y	N
1	2006-03-05T18:36:08.186Z	2006-03-05	04:56	light	75	1	partly cloudy	Roy	Heaney	33.660278	-117.998333	Y	Y
2	2002-07-26T23:33:55.223Z	2002-07-26	13:43	----	25	1	rain	Evelyn	Champlin	41.325278	-72.193611	Y	Y
3	1986-08-27T00:50:08.017Z	1986-08-27	16:12	sphere	47	1	mostly cloudy	Hoder	Ward	38.254167	-85.759444	Y	N
4	2004-09-25T08:47:29.886Z	2004-09-25	17:21	disk	58	1	rain	Abigayle	Grady	22.308085	69.600603	Y	N

22. Import library.

```
2010-04-29 08:47:39.886Z 2/3
```

```
In [4]: df.shape
```

```
Out[4]: (18000, 15)
```

Step 2: Cleaning, transforming, and preparing the data

Create another DataFrame with just the latitude and longitude attributes

```
In [ ]:
```

23. Import library.

jupyter ufo-modeling-lab Last Checkpoint: 20 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted conda_python3

Step 2: Cleaning, transforming, and preparing the data

Create another DataFrame with just the latitude and longitude attributes

```
In [5]: df_geo = df[['latitude', 'longitude']]
```

```
In [6]: df_geo.head()
```

	latitude	longitude
0	28.039167	-81.950000
1	33.660278	-117.998333
2	41.325278	-72.193611
3	38.254167	-85.759444
4	22.308085	69.600603

24. Import library.

```
In [7]: df_geo.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18000 entries, 0 to 17999
Data columns (total 2 columns):
latitude    18000 non-null float64
longitude   18000 non-null float64
dtypes: float64(2)
memory usage: 281.3 KB
```

```
In [ ]:
```

Next, let's go ahead and transform the pandas DataFrame (our dataset) into a numpy.ndarray. When we do this each row is converted to a Record object. According to the documentation, this is what the K-Means algorithm expects as training data. This is what we will use as training data for our model.

25. Import library.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18000 entries, 0 to 17999
Data columns (total 2 columns):
latitude    18000 non-null float64
longitude   18000 non-null float64
dtypes: float64(2)
memory usage: 281.3 KB
```

```
In [8]: missing_values = df_geo.isnull().values.any()
print('Are there any missing values? {}'.format(missing_values))
if(missing_values):
    df_geo[df_geo.isnull().any(axis=1)]
```

```
Are there any missing values? False
```

26. Import library.

See the documentation for `input_training`

```
In [9]: data_train = df_geo.values.astype('float32')
data_train
```

```
Out[9]: array([[ 28.039167, -81.95      ],
 [ 33.66028 , -117.99834 ],
 [ 41.32528 , -72.19361 ],
 ...,
 [ 37.49472 , -120.84556 ],
 [ 40.771946, -73.93056 ],
 [ 64.837776, -147.71638 ]], dtype=float32)
```

27. Import library.

```
In [10]: from sagemaker import KMeans

num_clusters = 10
output_location = 's3://'+ bucket + '/model-artifacts'

kmeans = KMeans(role=role,
                 train_instance_count=1,
                 train_instance_type='ml.c4.xlarge',
                 output_path=output_location,
                 k=num_clusters)

In [11]: job_name = 'kmeans-geo-job-{}'.format(datetime.now().strftime("%Y%m%d%H%M%S"))
print('Here is the job name {}'.format(job_name))

Here is the job name kmeans-geo-job-20190517133912
```

28. Import library.

```
In [11]: job_name = 'kmeans-geo-job-{}'.format(datetime.now().strftime("%Y%m%d%H%M%S"))
print('Here is the job name {}'.format(job_name))

Here is the job name kmeans-geo-job-20190517133512
```

```
In [14]: #time
kmeans.fit(kmeans.record_set(data_train), job_name=job_name)

2019-05-17 13:37:19 Starting - Starting the training job...
2019-05-17 13:37:21 Starting - Launching requested ML instances.....
2019-05-17 13:38:26 Starting - Preparing the instances for training.....
2019-05-17 13:39:44 Downloading - Downloading input data..

2019-05-17 13:40:12 Training - Training image download completed. Training in progress.
2019-05-17 13:40:12 Uploading - Uploading generated training model
2019-05-17 13:40:12 Completed - Training job completed
Docker entrypoint called with argument(s): train
[05/17/2019 13:40:01 INFO 14061972299200] Reading default configuration from /opt/amazon/lib/python2.7/site-packages/ml/algorithms/resources/default-input.jsonc ({'enable_profiler': false, '--tuning_objective_metric': '', 'num_dp_us': 'auto', 'local_lloyd_num_trials': 'auto', 'log_level': 'info', 'vectors': 'auto', 'local_lloyd_init_method': 'kmeans++', 'force_dense': true, 'epochs': 1, 'init_method': 'random', 'local_lloyd_tol': 0.001, 'local_lloyd_max_iter': 300, 'disable_wait_to_read': false, 'extra_center_factor': 'auto', 'eval_metrics': {'md': 'md'}, 'num_kv_servers': 1, 'mini_batch_size': 5000, 'half_life_time_size': 0, 'num_slices': 1})
[05/17/2019 13:40:02 INFO 14061972299200] Reading provided configuration from /opt/ml/input/config/hyperparameters.json: {'feature_dim': 2, 'k': 10, 'force_dense': True}
[05/17/2019 13:40:02 INFO 14061972299200] Final configuration: {'tuning_objective_metric': '', 'extra_center_fac
```

29. Search for S3 in the search bar.

aws Services ▾ Q s3

Search results for 's3'

Services (7)

Features (10)

Documentation (317,648) 

Marketplace (527)

S3
Scalable Storage In the Cloud

30. Select the bucket.

Amazon S3

Buckets

Batch operations

Block public access (account settings)

Feature spotlight 3

Search for buckets

+ Create bucket Edit public access settings Empty

aws-glue-scripts-442771530490-us-east-1

aws-glue-temporary-442771530490-us-east-1

car-data-analysis-acg

cf-templates-1cp6p2blczk1o-us-east-1

mi-labs-acg 

31. Select model-artifacts.

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with 'Services' and 'Resource Groups'. Below it, the path 'Amazon S3 > ml-labs-acg' is shown. There are three tabs: 'Overview', 'Properties' (which is selected), and 'Permissions'. A search bar says 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. The main area lists objects in the 'ml-labs-acg' bucket. Two objects are visible: 'model-artifacts' and 'ufo_dataset'. The 'model-artifacts' object is highlighted with a red box and has a large yellow arrow pointing to it. The 'ufo_dataset' object is partially visible below it.

32. Click on Kmeans-geo-jobs.

This screenshot shows the contents of the 'model-artifacts' folder within the 'ml-labs-acg' bucket. The interface is similar to the previous one, with tabs for 'Overview' and 'Actions'. A search bar allows searching for prefixes. Below are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. The list of objects in this folder shows one item: 'kmeans-geo-job-20190517133512'. This item is highlighted with a red box and has a large yellow arrow pointing to it.

33. Click on output.

Overview

Type a prefix and press Enter to search. Press ESC to clear.

Upload

Create folder

Download

Actions

Name ▾

output



34. Import library.

After we extract the results into a DataFrame of latitudes and longitudes, we can create a CSV with that data, load it onto S3 and then visualize it using QuickSight.

```
In [17]: import os  
model_key = 'model-artifacts/' + job_name + '/output/model.tar.gz'  
  
boto3.resource('s3').Bucket(bucket).download_file(model_key, 'model.tar.gz')  
os.system('tar -xvf model.tar.gz')  
os.system('unzip model algo-1')
```

Out[17]: 2304



35. Import library.

Out[17]: 2304

In [18]: pip install mxnet



```
(from mxnet) (2.20.1)  
Collecting graphviz<0.9.0,>=0.8.1 (from mxnet)  
  Downloading https://files.pythonhosted.org/packages/53/39/4ab213673844e0c004bed8a0721a3f6bb23eb8854ee75c234428  
892/graphviz-0.8.4-py3-none-any.whl  
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from requests<>2.21.0->mxnet) (1.23)  
Requirement already satisfied: idna<2.8,>=2.5 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from requests<>2.20.0->mxnet) (2.4)  
Requirement already satisfied: certifi<=2017.4.17 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from requests<>2.20.0->mxnet) (2019.3.9)  
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from requests<>2.20.0->mxnet) (3.0.4)  
Installing collected packages: numpy, graphviz, mxnet  
  Found existing installation: numpy 1.15.4  
    Uninstalling numpy-1.15.4:  
      Successfully uninstalled numpy-1.15.4  
Successfully installed graphviz-0.8.4 mxnet-1.4.1 numpy-1.14.6  
You are using pip version 10.0.1, however version 19.1.1 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.
```

36. Import library.

```
Successfully installed gAPHViz-0.8.4 mxnet-1.4.1 numpy-1.14.6  
You are using pip version 10.0.1, however version 19.1.1 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command
```

```
In [19]: import mxnet as mx  
Kmeans_model_params = mx.ndarray.load('model_algo-1')
```



37. Import library.

File Edit View Insert Cell Kernel Widgets Help

```
Kmeans_model_params = np.load('model_algo-1')
```

In [20]: cluster_centroids_kmeans = pd.DataFrame(Kmeans_model_params[0].tolist())
cluster_centroids_kmeans.columns=df_geo.columns
cluster_centroids_kmeans

Out[20]:

	latitude	longitude
0	41.286369	-74.856453
1	-3.558636	115.825752
2	35.375927	-117.235794
3	48.477852	3.664200
4	36.134438	-97.897385
5	26.707329	-81.378113
6	46.405426	-120.561981
7	25.992533	-146.748108
8	38.832069	-85.299072
9	61.760826	-148.924332



Let's go ahead and upload this dataset onto S3 and view within QuickSight

38. Import library.

```
8 38.832069 -85.299072
9 61.760826 -148.924332
```

Let's go ahead and upload this dataset onto S3 and view within QuickSight

In [23]:

```
from io import StringIO
csv_buffer = StringIO()
cluster_centroids_kmeans.to_csv(csv_buffer, index=False)
s3_resource = boto3.resource('s3')
s3_resource.Object(bucket, 'results/ten_locations_kmeans.csv').put(Body=csv_buffer.getvalue())
```



Out[23]:

```
{'ResponseMetadata': {'RequestId': '5B2C5F338C4D94A2',
'HostId': 'EBaTdqN46uapRaIWFzr0UNENSLV4vuhXsUML5389b4QC4MP0heG2FEcRTKJqYeSum2J8ikhBdrY',
'HTTPStatusCode': 200,
'HTTPHeaders': {'x-amz-id-2': 'EBaTdqN46uapRaIWFzr0UNENSLV4vuhXsUML5389b4QC4MP0heG2FEcRTKJqYeSum2J8ikhBdrY',
'x-amz-request-id': '5B2C5F338C4D94A2',
'date': 'Fri, 17 May 2019 13:53:38 GMT',
'etag': '5le129ef7a05a163e90bd3fd0433c70',
'content-length': '0',
'server': 'AmazonS3'},
'RetryAttempts': 0,
'ETag': '5le129ef7a05a163e90bd3fd0433c70'}
```

39. Click on results.

Amazon S3 > ml-labs-acg

ml-labs-acg

Objects Properties Permissions Metrics Management Access Points

Objects (4)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Delete Actions Create folder Upload

Find objects by prefix

< 1 >

	Name	Type	Last modified	Size	Storage class
	ARCHIVE/	Folder	-	-	-
	model-artifacts/	Folder	-	-	-
	results/	Folder	-	-	-
	udr_database	Folder	-	-	-



40. Select object.
41. Click on **Actions**.
42. Click on **Download**.

Amazon S3 > mi-labs-acg > results/

Copy
Move
Initiate restore
Query with S3 Select
Download actions
Download
Download as
Edit actions
Actions ▾
Upload

Objects (1)

Objects are the fundamental...
Find objects by prefix

Name	Type	Last modified	Size	Storage class
ten_locations_kmeans.csv	csv	February 8, 2021, 11:15:11 (UTC-05:00)	216.0 B	Standard

Step 3: Create a QuickSight Account

43. Search for **QuickSight** in the search bar.

aws Services ▾ Quick Connect workflow X

Search results for 'Quick'

Services (6) Services

Features (6)

Documentation (17,093)

Marketplace (512)

QuickSight
Fast, easy to use business analytics

AWS Glue DataBrew
Visual data preparation tool to clean and normalize data for analyt

44. Define the QuickSight account name.

QuickSight

Edition

QuickSight account name

QuicksightAccountACG

YOU WILL NEED THIS FOR YOU AND OTHERS TO SIGN IN.

45. Select Amazon S3 Bucket.
46. Click Select bucket.

Select Amazon S3 buckets

S3 buckets linked to QuickSight account	S3 buckets you can access across AWS
Select AWS S3 buckets to give QuickSight access permission to the bucket.	
<input type="checkbox"/> Select all	
<input type="checkbox"/> aws-athena-query-results-442771530490-us-east-1	
<input type="checkbox"/> aws-glue-scripts-442771530490-us-east-1	
<input type="checkbox"/> aws-glue-temporary-442771530490-us-east-1	
<input type="checkbox"/> car-data-analysis-acg	
<input type="checkbox"/> cf-templates-1cp6p2blczk1o-us-east-1	
<input checked="" type="checkbox"/> mi-labs-acg	
<input type="checkbox"/> my-user-data-output-bucket	
<input type="checkbox"/> output-user-data-csv-transformed	
<input type="checkbox"/> sagemaker-us-east-1-442771530490	
<input type="button" value="Close"/>	 <input type="button" value="Select buckets"/>

47. Click on **Finish**.

Choose S3 buckets
<input checked="" type="checkbox"/> Amazon S3 (1 buckets selected) Enables QuickSight to auto-discover your Amazon S3 buckets
<input type="checkbox"/> Amazon S3 Storage Analytics Enables QuickSight to visualize your S3 Storage Analytics data
<input type="checkbox"/> AWS IoT Analytics Enables QuickSight to visualize your IoT Analytics data
 <input type="button" value="Finish"/>

48. Click on **Go to Amazon QuickSight**.

Congratulations! You are signed up for Amazon QuickSight!

Access QuickSight with the following information

Account name: quicksightaccountacg

49. Click on “Manage data.”

QuickSight

Search for analyses, data sets and dashboards

New analysis

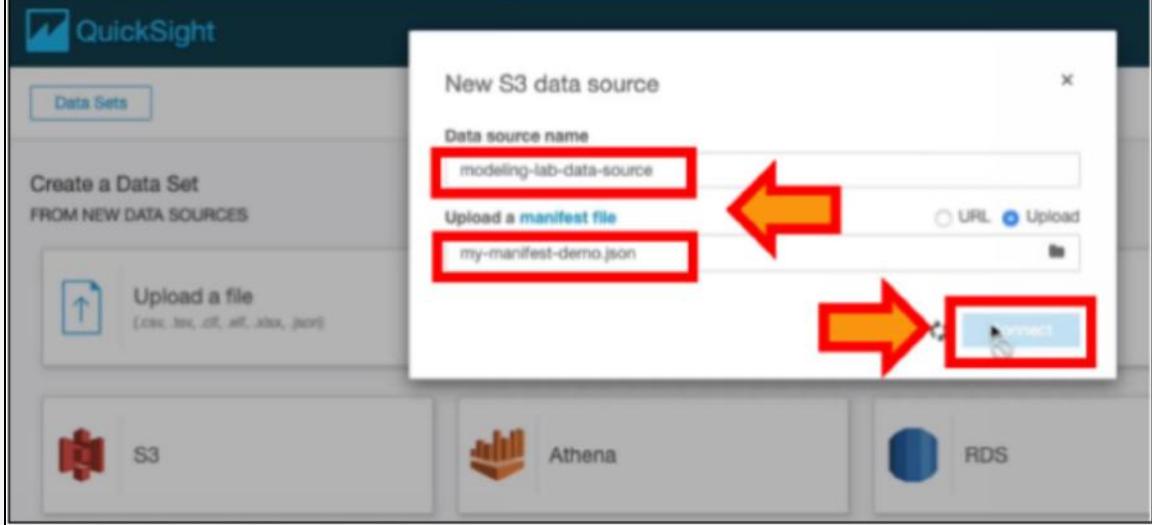
Manage data

50. Click on **S3**.

51. Define the data source name.

52. Upload a manifest file.

53. Click on **Connect**.

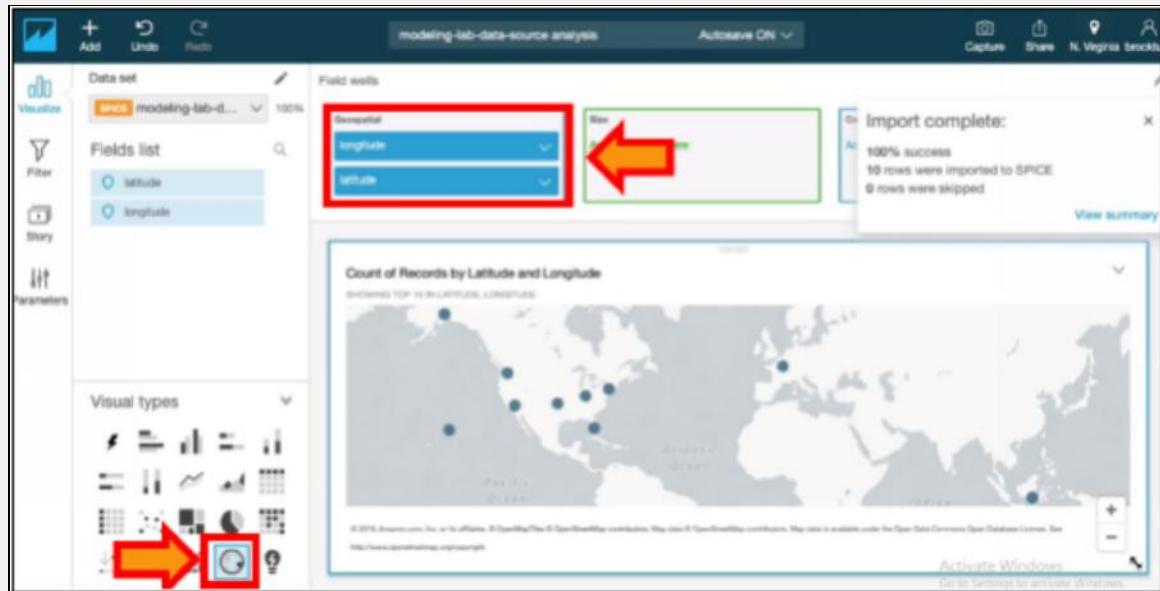


54. Click on **Visualize**.

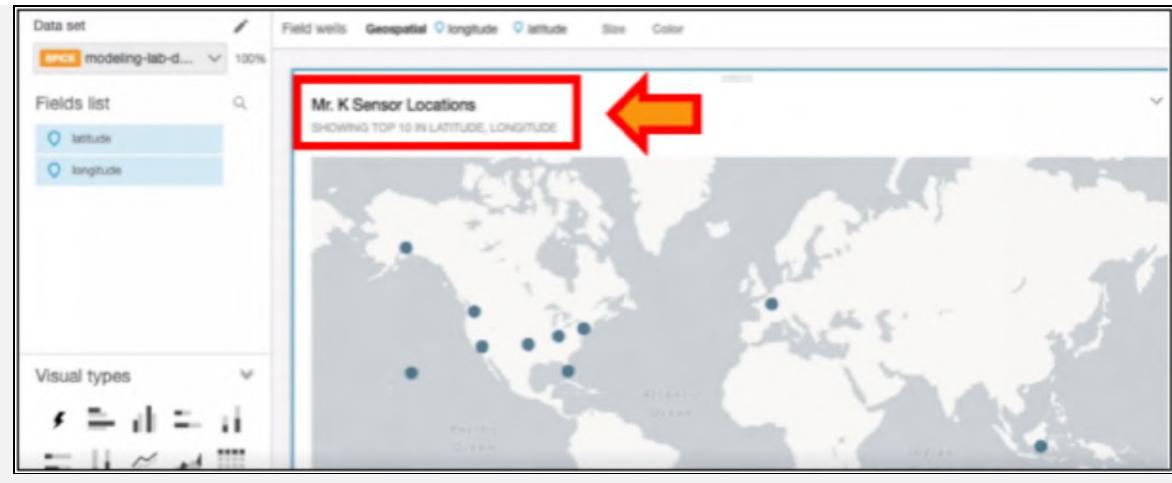


55. Select the Visual type.

56. Drag the Field list into Geospatial.



57. Rename Geospatial if you want.



Lab 08: Introducing Jupyter Notebooks (AWS SageMaker)

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	SageMaker, AWS

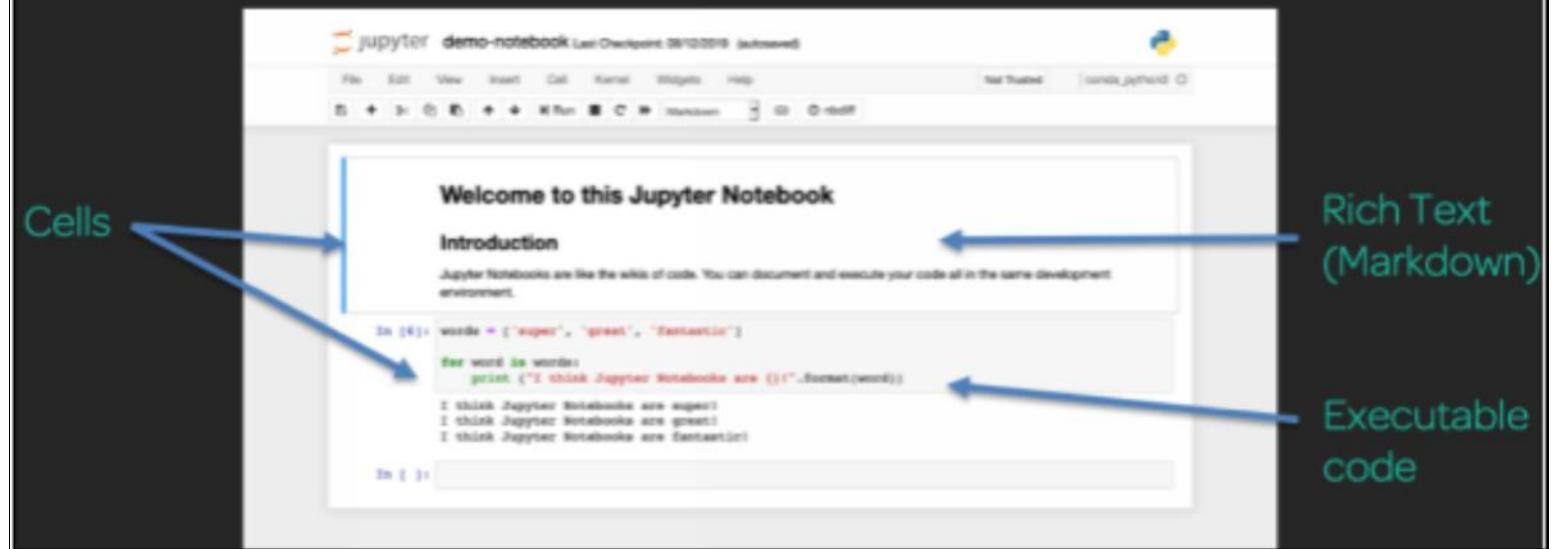
Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

You may create and share documents with live code, mathematics, visuals, and narrative text using the open-source web application Jupyter Notebook.

You can build and operate Jupyter notebooks from your DLAMI instance using a Jupyter notebook server. You can use Jupyter notebooks to run Machine Learning (ML) experiments for inference and training while utilizing the AWS infrastructure and DLAMI package libraries.



Case Study Enterprise EMRs – [Savana](#)

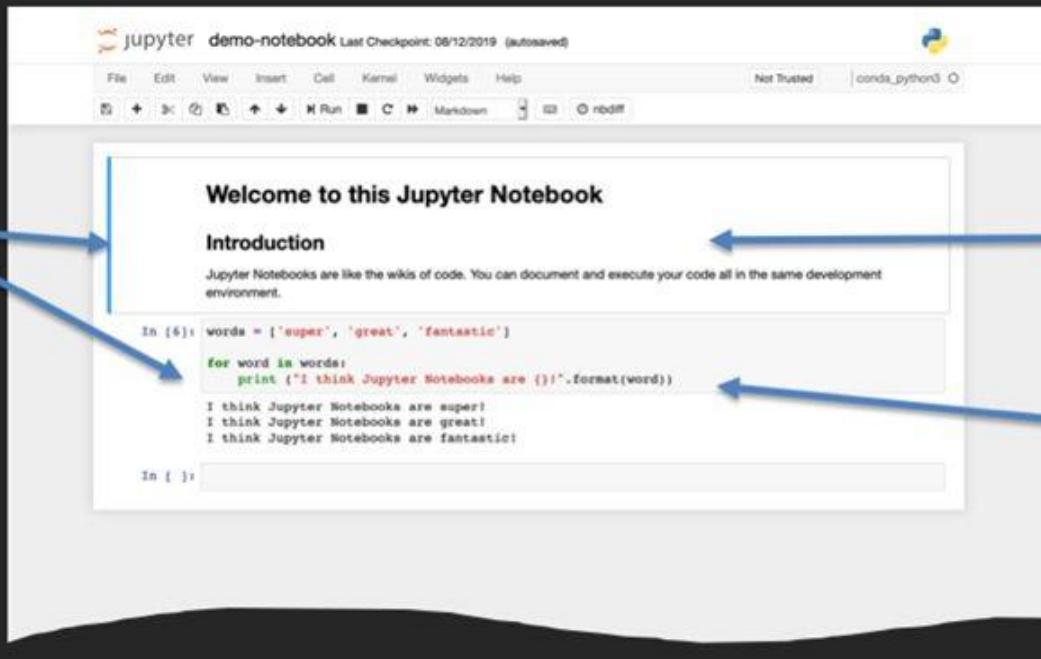
Business Challenge

You are working in an organization. You are provided some sample heights and weights of individual penguins. You have to use the data (stored in a csv file) to train a data model and create inferences so that you can guess what a penguin weighs if you know its height. Hence, you need to build a machine learning model that tells you approximately what a penguin should weigh based on height.

Proposed Solution

The solution is for you to use AWS services to automate all the tasks. You create an AWS SageMaker hyperparameter tuning job with different ranges of values for the hyperparameter to find the best configuration, which minimizes the validation: objective_loss metric. The reason that you use this metric is that it is used in multi-classification problems. This metric measures the performance of the classification model, and what it does is repeatedly calculate the difference between the values that the model predicts and the actual values of a label. Hence, every time it passes over data and makes predictions, it recalculates the objective loss and tries to minimize this value overall. Hence, that is the task here. AWS recommends that you minimize this value when using it as our objective metric.

Lab Diagram



Implementation Steps

1. Navigate to Jupyter Notebook.
2. Browse Jupyter Notebooks.
3. Create New Folders and Files
4. Build Your Own Notebook
5. Use Markdown to Add Richly Formatted Text to a Notebook

Solution

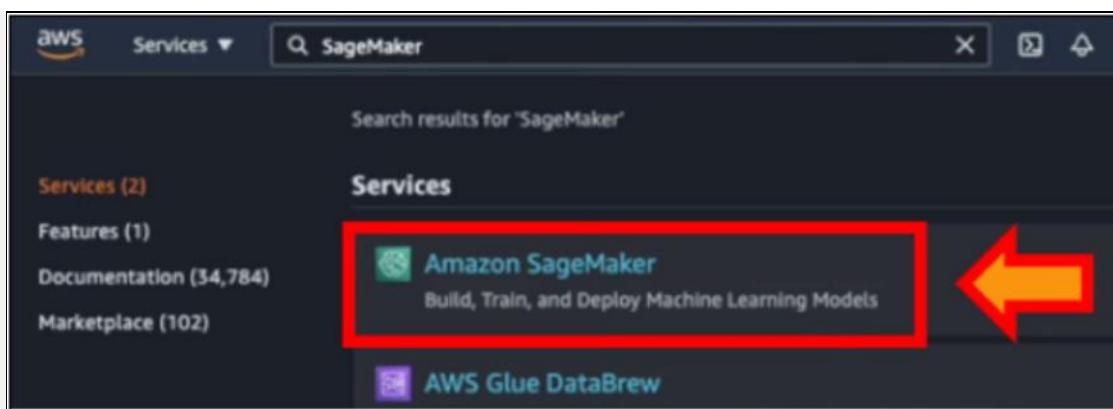
Step 1: Navigate to Jupyter Notebook

1. Login to AWS Management Console.

The screenshot shows the AWS Management Console homepage. At the top, there's a search bar and a navigation bar with links for "Services", "AWS Support", "Documentation", and "Help". The main content area has several sections:

- AWS services**: A list of recently visited services, with a link to "All services".
- Build a solution**: Options to "Launch a virtual machine", "Build a web app", or "Build using virtual servers".
- Stay connected to your AWS resources on-the-go**: Information about the AWS Console Mobile App, available for iOS and Android.
- Explore AWS**: A section featuring the "Amazon Location Service" and "Free AWS Training".

2. Search for SageMaker in the search bar.



3. Click on **Notebook Instances** under the dashboard.

Step 2: Browse Jupyter Notebooks

4. Click on **Open Jupyter**.

This screenshot shows the 'Notebook Instances' section of the Amazon SageMaker console. It lists a single instance named 'Notebook-T2v7ue802v79h'. The 'Actions' column contains a green circular icon followed by a red box containing the text 'Open Jupyter | Open JupyterLab'. A large orange arrow points from the left towards this button. On the far left, there's a sidebar with 'Dashboard', 'Search', 'SageMaker Domain', 'Studio', 'Images', and a 'Notebook' section which is expanded, showing 'Notebook instances' and 'Lifecycle configurations'. The 'Lifecycle configurations' link is also highlighted with a red box and has an orange arrow pointing towards it.

Step 3: Create New Folders and Files

5. Click on **New**.
6. Click on **Folder**.

This screenshot shows the Jupyter Notebook interface. On the left, there's a file browser window showing files like 'lost+found', 'Introduction_to_Jupyter_Notebooks.ipynb', and 'animal-animal-photography-covid-86405.jpg'. On the right, a context menu is open over a list of available kernels. The 'New' option is highlighted with a red box and has an orange arrow pointing towards it. The menu also includes 'Text File' and 'Folder' options, both of which are also highlighted with red boxes and have orange arrows pointing towards them.

7. Select **Folder**.
8. Click on **Rename**.

Files Running Clusters SageMaker Examples Conda

Upload New

Name Last Modified File size

	18 minutes ago	
	seconds ago	
	17 minutes ago	12.8 kB
	17 minutes ago	36.4 kB
	17 minutes ago	8.7 kB
	17 minutes ago	134 B
	17 minutes ago	33.3 kB

9. Define Name.
10. Click on **Rename**.

Rename directory

Enter a new directory name:

my-new-folder

Rename

11. Click on the **new folder**.

0 /

lost+found

my-new-folder

Introduction_to_Jupyter_Notebooks.ipynb

12. Select the item to perform the action.

Select items to perform actions on them

Upload New

Name Last Modified File size

	22 minutes ago	
	4 minutes ago	
	21 minutes ago	12.8 kB
	21 minutes ago	36.4 kB
	21 minutes ago	8.7 kB
	21 minutes ago	134 B
	21 minutes ago	33.3 kB

13. Go through to the **Introduction to Jupyter Notebooks**.

Introduction

In this Linux Academy hands-on lab we take an introductory look at Jupyter Notebooks from within AWS SageMaker.

Jupyter Notebooks are designed to provide an easy way to interact with data and documentation at the same time. Data scientists often use these books while designing and sharing solutions.

Think of Jupyter Notebooks as a Wiki for code and data!

Scenario

Most of this Linux Academy hands-on lab is designed to familiarize you with the basic components of Jupyter Notebooks. We cover some basic libraries, commands, and their uses. We also have a data scenario to play with:

14. Click on **Insert**.
15. Click on **Insert Cell Below**.

1) Markdown

This is a `Markdown` cell.

Let's upload and display an image. When an image is uploaded we can add the following code to insert it into the document:

```
(pinehead.jpg)
```

1. Add a cell below this one.
2. Set it as a `Markdown` cell.
3. Copy the code snippet above into the cell.

16. Select the cell. Click on **Run**.

1) Markdown

This is a `_Markdown_` cell.

Let's upload and display an image. When an image is uploaded we can add the following code to insert it into the document:

```
(pinehead.jpg)
```

1. Add a cell below this one.
2. Set it as a `Markdown` cell.
3. Copy the code snippet above into the cell.

In []: Here is an image of Pinehead:

2) Command-Line Operations

What happens when we run this cell?

17. Go through the output result.

![Pinehead](pinehead.jpg)

1. Add a cell below this one.
2. Set it as a Markdown cell.
3. Copy the code snippet above into the cell.

Here is an image of Pinehead:



18. Select the cell. Click on Run.

jupyter Introduction_to_Jupyter_Notebooks (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted conda_python3 O

Insert Insert Cell Above Insert Cell Below

Howdy! Insert Cell Below In more about what is happening in the code and then take the time to experiment with the code; make changes, break it, fix it, and learn!

1) Markdown

This is a `_Markdown_` cell.

Let's upload and display an image. When an image is uploaded we can add the following code to insert it into the document:

```
... [[Pinehead]](pinehead.jpg) ...
```

1. Add a cell below this one.
2. Set it as a Markdown cell.
3. Copy the code snippet above into the cell.

What happens when we run this cell?

In [4]:

```
whoami  
which python
```

ec2-user
/home/ec2-user/anaconda3/envs/python3/bin/python

In []:

19. Select the cell. Click on Run.

jupyter Introduction_to_Jupyter_Notebooks (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run

1) Markdown

This is a *Markdown* cell.

Let's upload and display an image. When an image is uploaded we can add the following code to insert it into the document:

```
I [Pinehead](pinehead.jpg)
```

1. Add a cell below this one.
2. Set it as a Markdown cell.
3. Copy the code snippet above into the cell.

2) Command-Line Operations

What happens when we run this cell?

```
In [4]: whoami  
whoami  
ec2-user  
/home/ec2-user/anaconda3/envs/python3/bin/python
```

In []: !python --version



20. Go through the output result.

```
In [2]: !python --version
```

Python 3.6.5 :: Anaconda, Inc.

21. Select the cell. Click on Run.

jupyter Introduction_to_Jupyter_Notebooks (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run

```
In [2]: !python --version
```

Python 3.6.5 :: Anaconda, Inc.

3) Python Operations

Notebooks are run within a kernel. These are like virtual environments that contain the notebook in the top right. We can even change it.

This notebook is running with a Python kernel. Python is a common language for Jupyter notebooks and key libraries for the processing of data.

```
In [3]: words = ['awesome', 'amazing', 'great']  
for w in words:  
    print('This Linux Academy lab is %s!' % w)
```



22. Go through the output result.

3) Python Operations

Notebooks are run within a kernel. These are like virtual environments that contain the running code and data. You can see the kernel being used for this notebook in the top right. We can even change it.

This notebook is running with a Python kernel. Python is a common language for Jupyter Notebooks but you can use others. Other kernels include different versions of Python and key libraries for the processing of data.

```
In [3]: words = ['awesome', 'amazing', 'great']
for w in words:
    print('This Linux Academy lab is %s!' % w)

This Linux Academy lab is awesome!
This Linux Academy lab is amazing!
This Linux Academy lab is great!
```



23. Select the cell. Click on **Run**.

jupyter Introduction_to_Jupyter_Notebooks (autosaved)

File Edit View Insert Cell Kernel Widgets Help

SyntaxError: invalid syntax

```
In [ ]: words = ['awesome', 'amazing', 'great']
for w in words:
    print('This Linux Academy lab is %s!' % w)
```

Run

4) Python Lists

```
In [ ]: myList = [0, 1, 2, 3, 4, 5]
```



24. Go through the output result.

4) Python Lists

```
In [6]: myList = [0, 1, 2, 3, 4, 5]
myList
```

```
Out[6]: [0, 1, 2, 3, 4, 5]
```

25. Similarly, select each cell and click on **Run** and see the output.

4) Python Lists

```
In [6]: myList = [0, 1, 2, 3, 4, 5]  
myList
```

```
Out[6]: [0, 1, 2, 3, 4, 5]
```

```
In [7]: myList.append('blue')  
myList
```

```
Out[7]: [0, 1, 2, 3, 4, 5, 'blue']
```

```
In [8]: myList[3]
```

```
Out[8]: 3
```

```
In [9]: myList[3:]
```

```
Out[9]: [3, 4, 5, 'blue']
```

```
In [10]: len(myList)
```

```
Out[10]: 7
```

26. Select the cell. Click on Run.

jupyter Introduction_to_Jupyter_Notebooks (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | conda_python

Out[8]: 7

5) NumPy

NumPy is the fundamental package for scientific computing with Python. Some of the elements it contains are:

- a powerful N-dimensional array object.
- sophisticated (broadcasting) functions.
- tools for integrating C/C++ and Fortran code.
- useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data types can be defined. This enables NumPy to seamlessly and speedily integrate with a wide variety of databases.

(Source: <https://www.numpy.org/>)

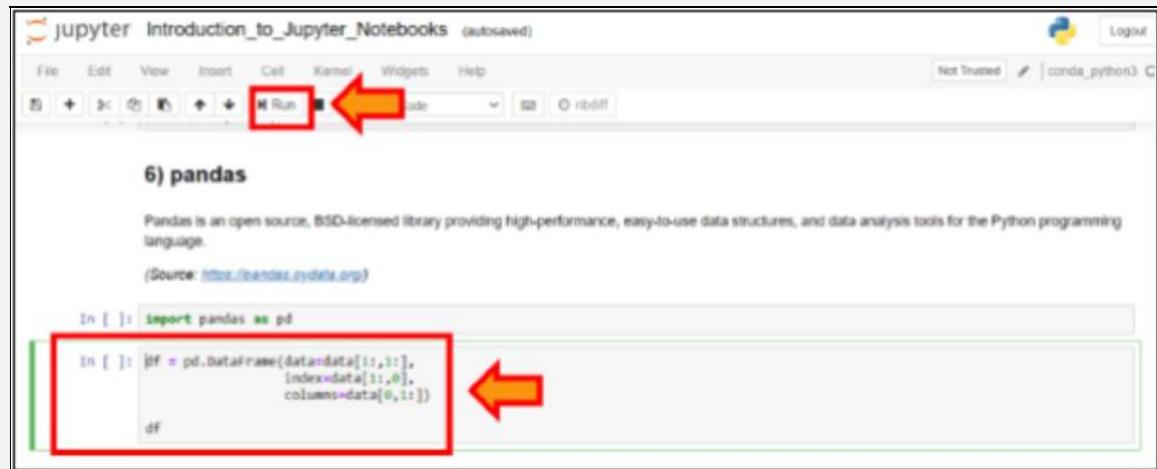
```
In [ ]: import numpy as np
```

```
In [ ]: np.pi
```

27. Go through the output result.

```
In [10]: np.pi  
Out[10]: 3.141592653589793
```

28. Select the cell. Click on Run.



The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A red arrow points to the 'Run' button in the toolbar. Below the toolbar, there's a section titled '6) pandas'. It contains a brief introduction to Pandas and its source link. In the code editor, two cells are shown. The first cell contains the command 'import pandas as pd'. The second cell, which is selected and highlighted with a red box and a red arrow, contains the command 'df = pd.DataFrame(data=data[1:,1:], index=data[1:,0], columns=data[0,1:])'. The variable 'df' is also defined in this cell.

29. Go through the output result.

```
In [15]: df = pd.DataFrame(data=data[1:,1:],  
                           index=data[1:,0],  
                           columns=data[0,1:])
```

```
df
```

Out[15]:

	Col1	Col2
Row1	1	2
Row2	3	4
Row3	5	6

30. Go through penguin-data.csv

Select items to perform actions on them.

		Name	Last Modified	File size
□	0	lost+found	an hour ago	
□	my-new-folder		an hour ago	
□	Introduction_to_Jupyter_Notebooks.ipynb	Running	2 minutes ago	16.3 kB
□	animal-animal-photographycold-86405.jpg	an hour ago	36.4 kB	
□	is-loss.png	an hour ago	8.7 kB	
□	penguin-data.csv	an hour ago	134 B	
□	pinehead.jpg	an hour ago	33.3 kB	

jupyter penguin-data.csv ✓ an hour ago

File Edit View Language

```
1 Height,Weight
2 11,15
3 11,15
4 11,20
5 12,17
6 13,18
7 14,17
8 15,21
9 15,14
10 16,20
11 16,24
12 17,24
13 17,20
14 18,22
15 17,20
16 17,27
17 18,19
18 20,24
19 20,25
20 19,24
21 20,26
22
```

31. Select the cell. Click on **Run**.

jupyter Introduction_to_Jupyter_Notebooks (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted conda_python3

Run

```
index=data[1:,0],  
columns=data[0,1:])  
df  
NameError: name 'pd' is not defined
```

Sample Data

Let's load some sample data from a CSV.

```
In [ ]: penguin_data = pd.read_csv("penguin-data.csv")  
penguin_data.shape
```

```
In [ ]: penguin_data.head()
```

32. Go through the output result.

```
In [16]: penguin_data = pd.read_csv("penguin-data.csv")  
  
penguin_data.shape
```

Out[16]: (20, 2)

33. Select the cell. Click on Run.

jupyter Introduction_to_Jupyter_Notebooks (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run

Out[15]:

	Col1	Col2
Row1	1	2
Row2	3	4
Row3	5	6

Sample Data

Let's load some sample data from a CSV.

```
In [16]: penguin_data = pd.read_csv("penguin-data.csv")  
  
penguin_data.shape
```

```
Out[16]: (20, 2)
```

```
In [17]: penguin_data.head()
```

34. Go through the output result.

```
In [17]: penguin_data.head()
```

Out[17]:

	Height	Weight
--	--------	--------

0	11	15
1	11	15
2	11	20
3	12	17
4	13	18

35. Select the cell. Click on Run.

The screenshot shows a Jupyter Notebook window titled "jupyter Introduction_to_Jupyter_Notebooks (unsaved changes)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A red box highlights the "Run" button in the toolbar, which has a yellow arrow pointing to it. Below the toolbar, a section titled "Sample Data" contains the text "Let's load some sample data from a CSV." Two code cells are visible:

```
In [16]: penguin_data = pd.read_csv("penguin-data.csv")
penguin_data.shape
```

```
Out[16]: (20, 2)
```

```
In [17]: penguin_data.head(2)
```

A red box highlights the "In [17]" cell, with a yellow arrow pointing to it.

36. Go through the output result.

In [18]: penguin_data.head(2)

Out[18]:

	Height	Weight
0	11	15
1	11	15

37. Select the cell. Click on Run.

jupyter introduction_to_Jupyter_Notebooks (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | conda_python3

NameError: name 'pd' is not defined

In []: penguin_data.head()

7) Matplotlib

Matplotlib is a Python, 2D-plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits.

(Source: <https://matplotlib.org/>)

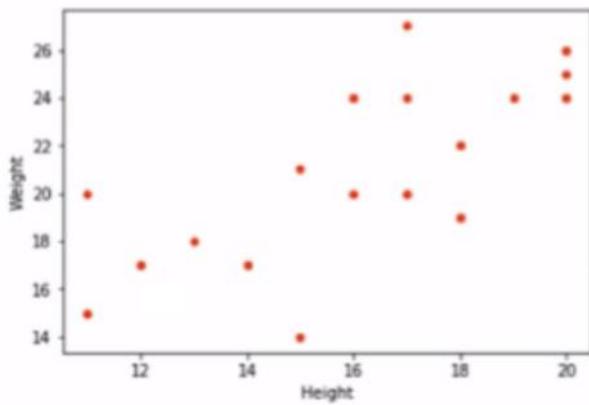
In []: `import matplotlib.pyplot as plt`
`%matplotlib inline`
`penguin_data.plot(kind='scatter',x='Height',y='Weight',color='red')`

38. Go through the output result.

In [20]: `import matplotlib.pyplot as plt`
`%matplotlib inline`

`penguin_data.plot(kind='scatter',x='Height',y='Weight',color='red')`

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb802fcc908>



39. Select the cell. Click on Run.



Train a Linear Regression Model

Train a very simple model:

```
In [ ]: from sklearn.linear_model import LinearRegression  
from sklearn.model_selection import train_test_split  
  
# Set up the Linear Regression model  
model = LinearRegression()  
  
# Train the model with our data  
model.fit(penguin_data[['Height']], penguin_data['Weight'])
```

Show the very simple model mapped to the training data:



40. Go through the output result.

Train a Linear Regression Model

Train a very simple model:

```
In [21]: from sklearn.linear_model import LinearRegression  
from sklearn.model_selection import train_test_split  
  
# Set up the Linear Regression model  
model = LinearRegression()  
  
# Train the model with our data  
model.fit(penguin_data[['Height']], penguin_data['Weight'])
```

Out[21]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

41. Select the cell. Click on Run.

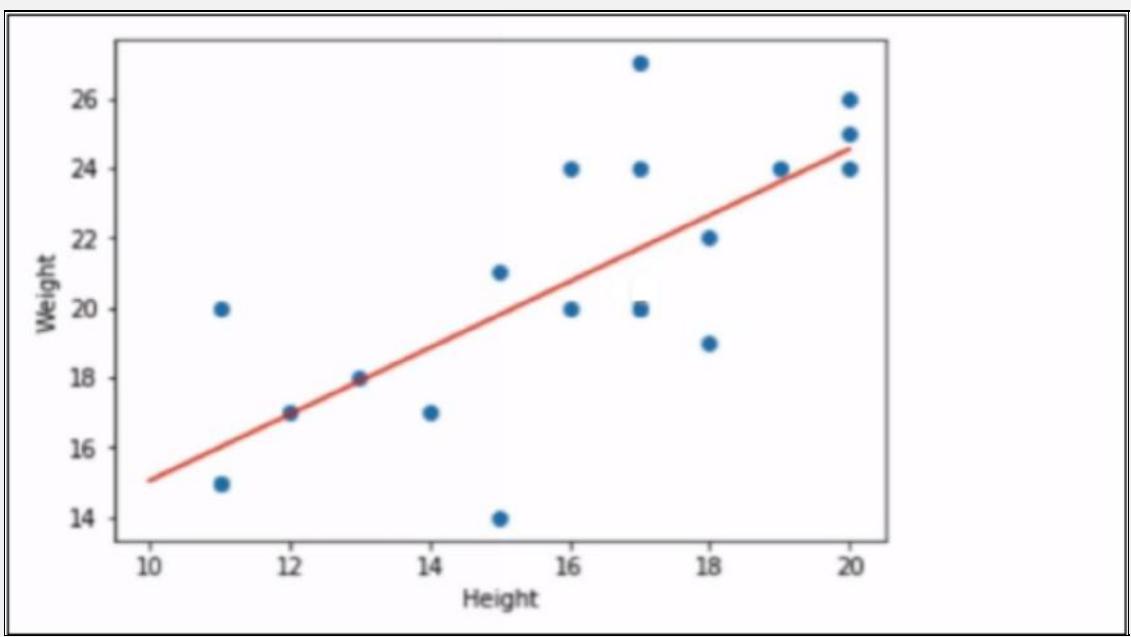


Show the very simple model mapped to the training data:

```
In [ ]: # Plot our original training data  
axes = plt.axes()  
axes.scatter(x=penguin_data['Height'], y=penguin_data['Weight'])  
  
# Determine the best fit line  
slope = model.coef_[0]  
intercept = model.intercept_  
  
# Plot our model line  
x = np.linspace(10,20)  
y = slope*x+intercept  
axes.plot(x, y, 'r')  
  
# Add some labels to the graph  
axes.set_xlabel('Height')  
axes.set_ylabel('Weight')  
  
plt.show()
```

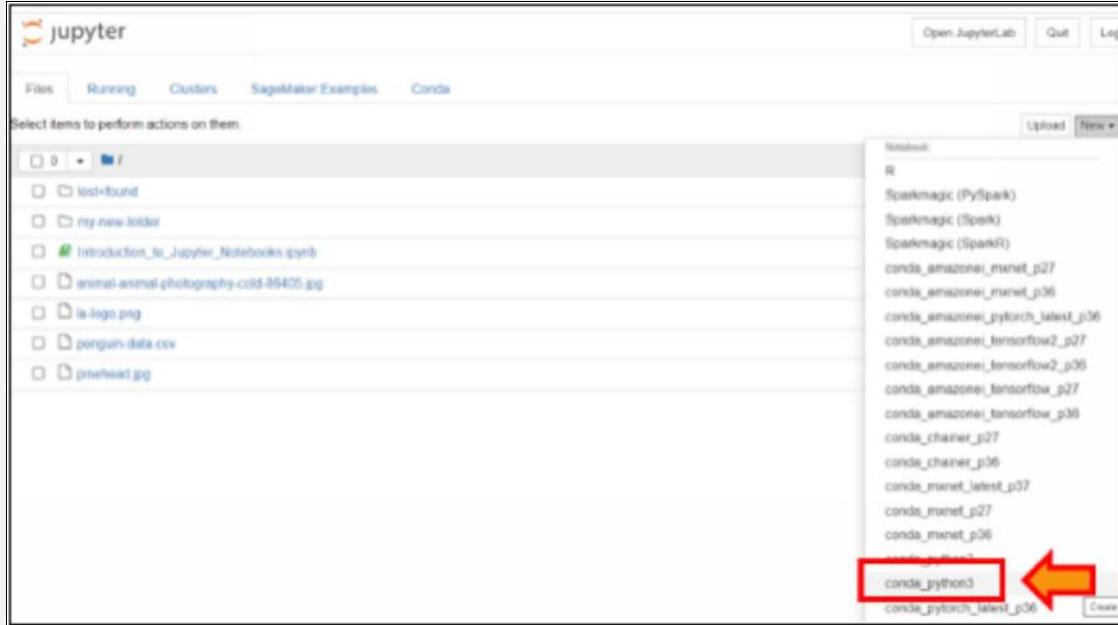


42. Go through the output result.

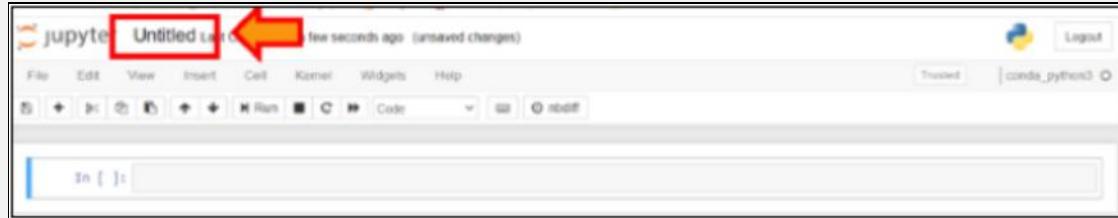


Step 4: Build Your Own Notebook

43. Click on **conda_python3**.



44. Click on **Untitled**.



45. Define notebook name.

46. Click on **Rename**.

Rename Notebook

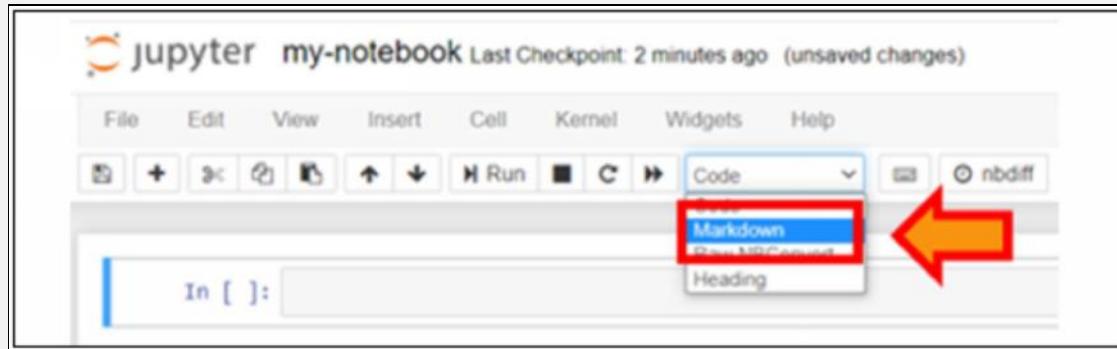
Enter a new notebook name:

my-notebook

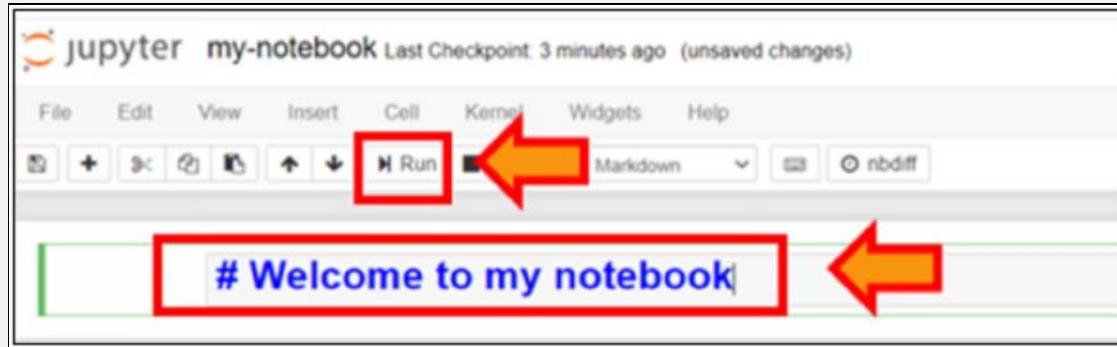
Rename

Step 5: Use Markdown to Add Richly Formatted Text to a Notebook

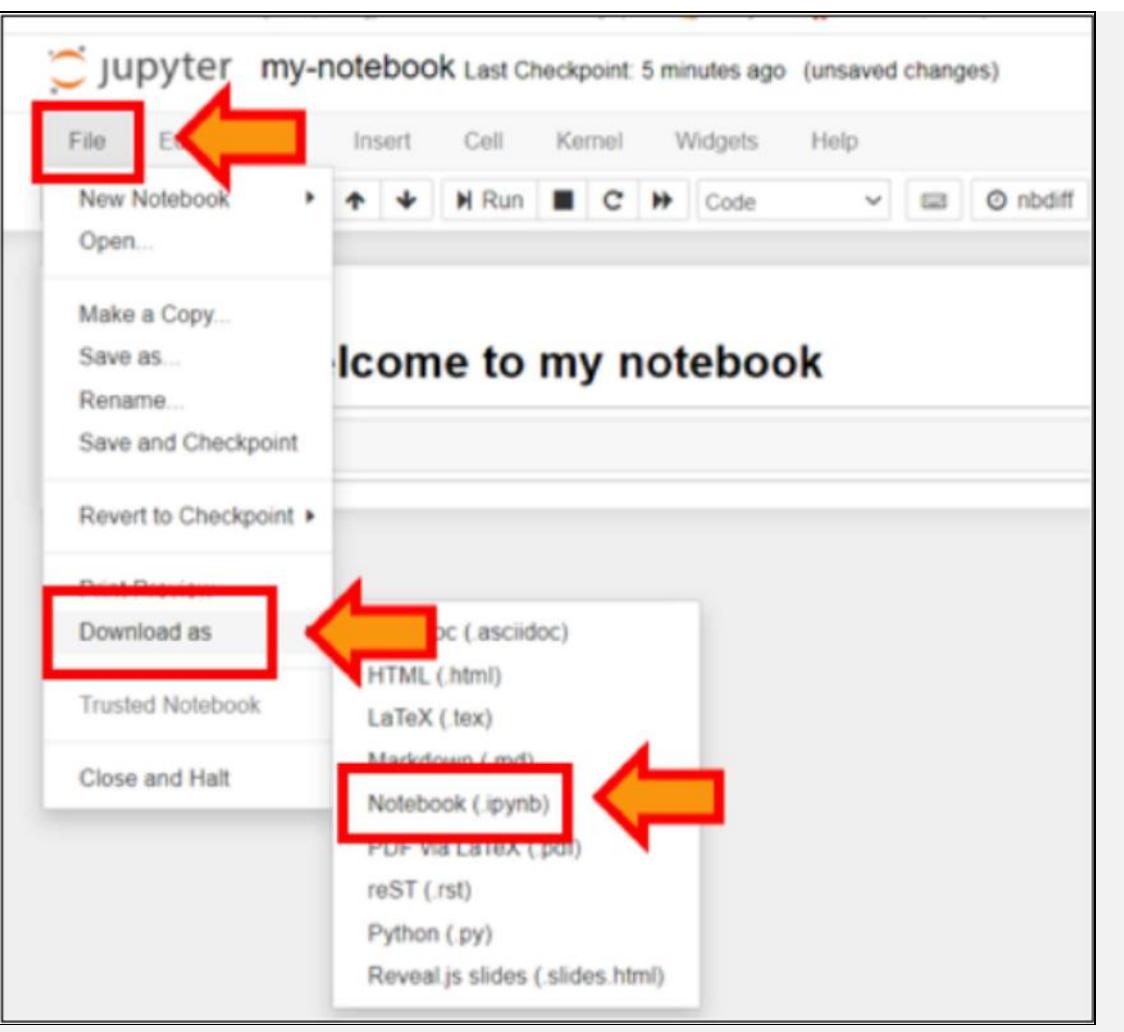
47. Click on **Markdown**.



48. Add **#Welcome to my notebook** in the cell.



49. Click on **File**.
50. Click on **Download as**.
51. Click on **Notebook (.ipynb)**.



Lab 09: Algorithms

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	S3, SageMaker, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

XGBoost

The XGBoost (eXtreme Gradient Boosting) version of the gradient boosted trees method is a popular and efficient open-source implementation. Gradient boosting is a supervised learning approach for effectively predicting a target variable by combining an ensemble of estimates from a series of simpler and weaker models. Because of its strong handling of a range of data kinds, relationships, and distributions, as well as the diversity of hyperparameters that can be fine-tuned, the XGBoost method does well in machine learning contests. XGBoost may be used to solve issues involving regression, classification (binary and multi-class), and ranking.

Linear Learner Algorithm

Linear models have supervised learning algorithms for regression, binary classification, or multi-class classification problems. You give the model labels (x,y) with x being a high dimensional vector and y numeric. The algorithm learns a linear function, or, for classification problems, a linear threshold function and maps a vector x to an approximation of label y . Both classification and regression problems may be solved with Amazon SageMaker's Linear Learner algorithm.

Case Study Enterprise Financial Service— Kasasa

Background

Community banks and credit unions use Kasasa's decades-old, cutting-edge financial technology and marketing services to help people feel proud of their money. The business's main office is in Austin, Texas.

Business Challenge

You work as an AWS architect in Kasasa. Kasasa wants to build a model that can predict the legitimacy of a UFO sighting based on information supplied by the submitter.

Proposed Solution

We can use the ground truth dataset that has been collected by the organization previously. The organization has previously determined whether a sighting could be explained (as a hoax or other natural explanation), unexplained or probable.

We will build a model to determine whether newly reported UFO sightings are legitimate or not (explained, unexplained or probable). The organization requires the model to be at least 90% accurate. Hence, we will choose an algorithm, prepare and transform our data and determine how accurate our model is.

Our final result is that we need to have a model artifact stored in S3. We also want to present different model validation metrics, like accuracy, recall, precision, and the F1 score. And we would also present the model error rate to show that our model is getting smarter and smarter, and the error rate decreases as we train our model.

We will build a model using the XGBoost algorithm as a multi-classification problem with `researchOutcome`, which is an attribute in our dataset as the target attribute. Our goal when using this algorithm is to minimize the training and the validation error.

The reason to choose XGBoost is that it is simple to implement and only needs two hyperparameters with 35 optional hyperparameters. We can use XGBoost for different problems, like ranking problems or regression problems, but here we will use it as a classification problem, specifically, a multi-classification problem.

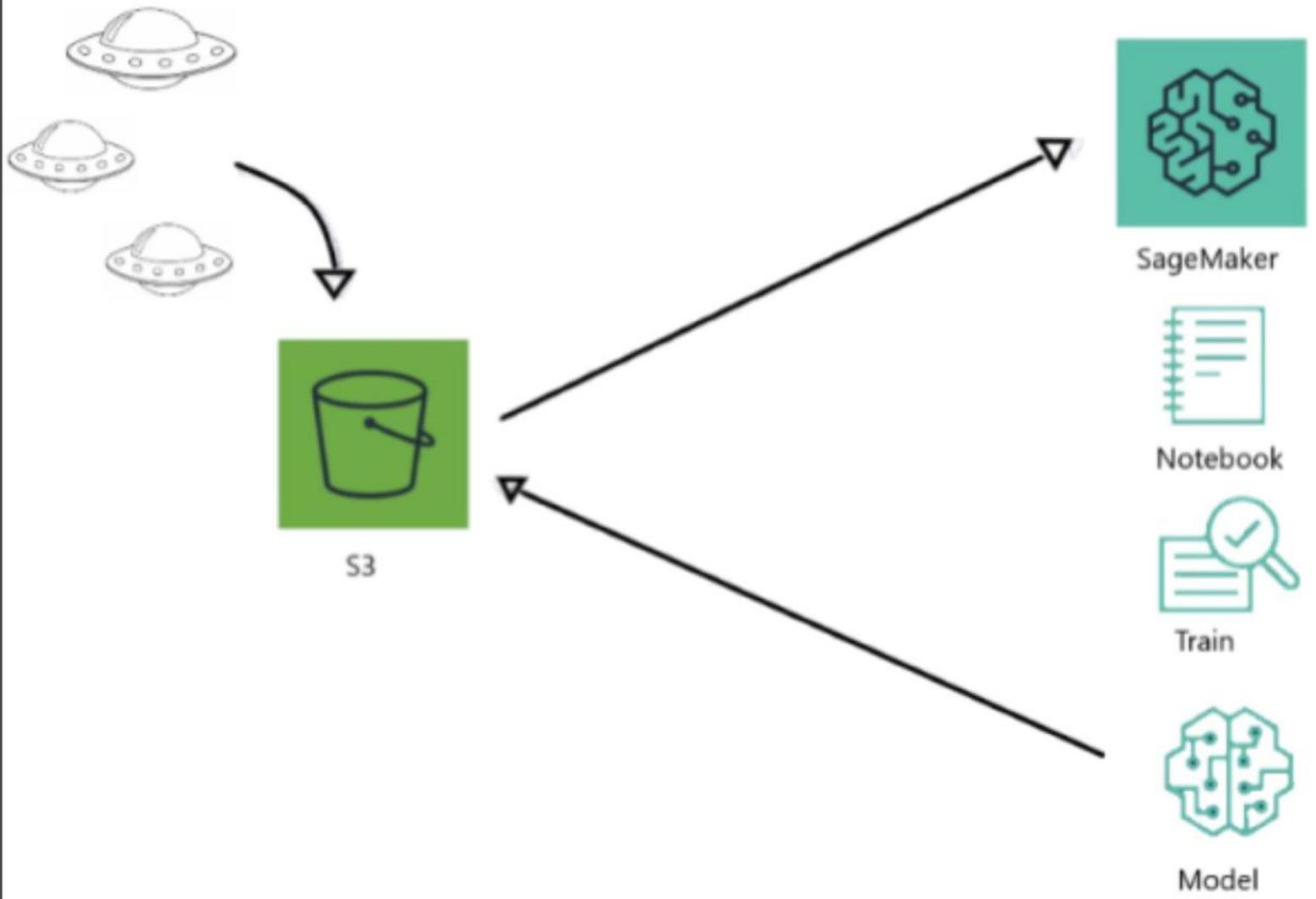
Since we are not exactly sure which attributes are most important, we need to use a classification algorithm over a clustering algorithm because if we choose a clustering algorithm, we need to know which attributes within our dataset will be the most determining factor.

We will employ a different algorithm to categorize the veracity of a UFO sighting. We will build a model using the Linear Learner algorithm as another multi-classification problem with the same attribute as our target attribute, researchOutcome. The goal is to maximize the training accuracy along with other metrics, like precision, F1 score, and recall. Linear Learner is a very flexible algorithm, and we can train different models and determine which attributes from our data are the most determining factors. Another feature of Linear Learner is it also has built-in hyperparameter tuning.

The same dataset that the organization previously gathered will be used. We are going to use this dataset and create a notebook instance. Within this notebook, we will analyze, visualize, prepare, transform, and get our data ready for our machine learning algorithm. In our case, XGBoost and Linear Learner both expect all of our values to be in numeric format, so we will need to encode those values or transform those values into numeric attributes.

Once we do the data transformation, we will divide our dataset into a training set, a validation set, and a testing set. Once we have our training and validation dataset ready, we will then run it through our algorithm, which will create a model and automatically upload that model onto S3. The logs from each of these algorithms from both the XGBoost and the Linear Learner will show us our model's error rate and accuracy.

Lab Diagram



Implementation Steps

1. Upload Data into S3.
2. Import all the necessary libraries.
3. Load the data from Amazon S3.
4. Clean, transform, analyze, and prepare the dataset.
5. Create and Train our model (XGBoost).
6. Create and train our model (Linear Learner).

Solution

Note: A lot of trial and error happens during the machine learning process. This is just part of the machine learning process. You will continuously create jobs that fail, and you will figure out the reasons why. You will realize that you do not have enough data, so you will have to get more. You will realize your data may not be in the right format. You might have accidentally left out a hyperparameter or used the wrong value for a hyperparameter. You will realize that your model is extremely inaccurate, so you must change something. You will realize that one algorithm works better than the other. But you need to remember that the machine learning process is a trial-and-error.

Step 1: Upload Data into S3

1. Log in to the AWS Console.
2. Go to Services and click on S3.

The screenshot shows the AWS search interface. At the top, there is a search bar with the text "S3" and a magnifying glass icon. To the right of the search bar are icons for refresh, alert, and more, followed by a "More ▾" button. Below the search bar, there are four tabs: "Services (7)" (which is highlighted in red), "Features (10)", "Documentation (419,077)", and "Knowledge Art". The main content area displays several service cards. The first card, for "S3 Scalable Storage in the Cloud", has a red border around it. The second card, for "S3 Glacier Archive Storage in the Cloud", the third for "Athena Query Data in S3 using SQL", and the fourth for "AWS Snow Family Large Scale Data Transport" are also visible. At the bottom of the page, there is a footer with links for "Feedback", "English (US) ▾", "Privacy Policy", "Terms of Use", and "Cookie preferences", along with a copyright notice: "© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved."

3. Click on the bucket **ml-labs-ips**.

[+ Create bucket](#)[Edit public access settings](#)[Empty](#)[Delete](#)

<input type="checkbox"/>	aws-glue-temporary-442771530490-us-east-1	Objects can be public
<input type="checkbox"/>	car-data-analysis-acg	Bucket and objects not public
<input type="checkbox"/>	cf-templates-1cp6p2biczk1o-us-east-1	Objects can be public
<input type="checkbox"/>	ml-labs-ips	Bucket and objects not public
<input type="checkbox"/>	my-user-data-output-bucket	Objects can be public
<input type="checkbox"/>	output-user-data-csv-transformed	Bucket and objects not public
<input type="checkbox"/>	sagemaker-us-east-1-442771530490	Objects can be public

4. Click on the subfolder **ufo_dataset**.

Overview

Properties

Permissions

Type a prefix and press Enter to search. Press ESC to clear.

Upload

+ Create folder

Download

Actions ▾

<input type="checkbox"/>	Name ▾	Last modified ▾
<input type="checkbox"/>	model-artifacts	--
<input type="checkbox"/>	results	--
<input type="checkbox"/>	ufo_dataset	--

5. Here, you can see the UFO dataset named **uf0_fullset.csv**.

Overview



Type a prefix and press Enter to search. Press ESC to clear.

Upload

+ Create folder

Download

Actions ▾

Name ▾

ufo_fullset.csv



If you do not have this dataset, you can get it from the GitHub link below:

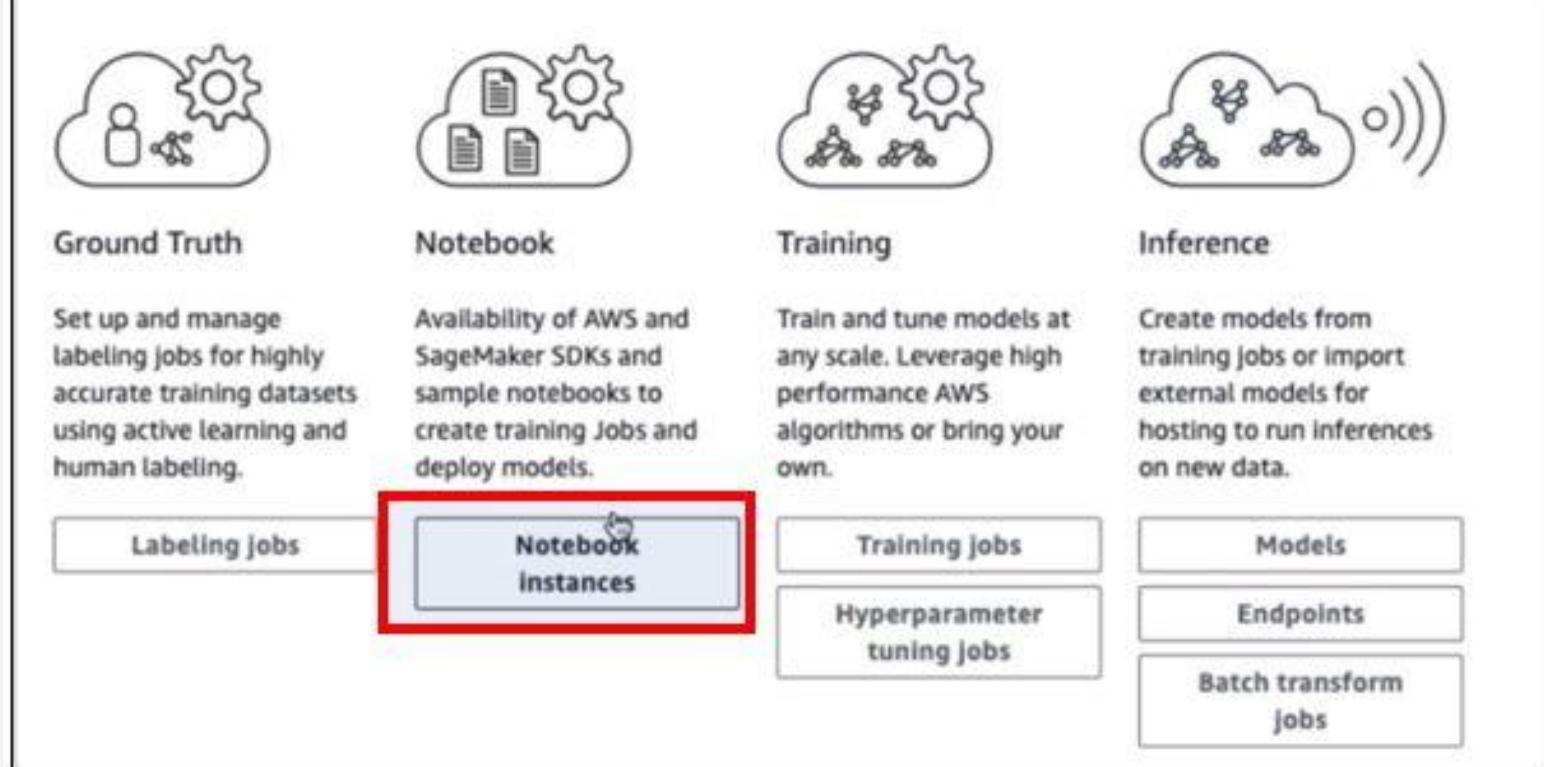
https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter7/ufo_fullset.csv

6. Navigate to **SageMaker**.

The screenshot shows the AWS SageMaker Studio search results. The search bar at the top contains the text "SageMaker Studio". Below the search bar, there are four tabs: "Services (2)", "Features (3)", "Documentation (45,399)", and "Knowledge Articles (0)". The "Services (2)" tab is selected. Under this tab, two service cards are listed. The first card, "Amazon SageMaker", is highlighted with a red box. It features a blue icon with a neural network symbol, the text "Amazon SageMaker", and the subtitle "Build, Train, and Deploy Machine Learning Models". The second card, "AWS Glue DataBrew", has a purple icon with a beaker symbol, the text "AWS Glue DataBrew", and the subtitle "Visual data preparation tool to clean and normalize data for analytics and machine learn...". At the bottom of the page, there is a footer with links for "Feedback", "English (US)", "Privacy Policy", "Terms of Use", "Cookie preferences", and copyright information: "© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved."

7. Click on **Notebook Instances**.

Overview



8. Click on **ml-labs-notebook-instance**.

Screenshot of the Amazon SageMaker Notebook Instances page:

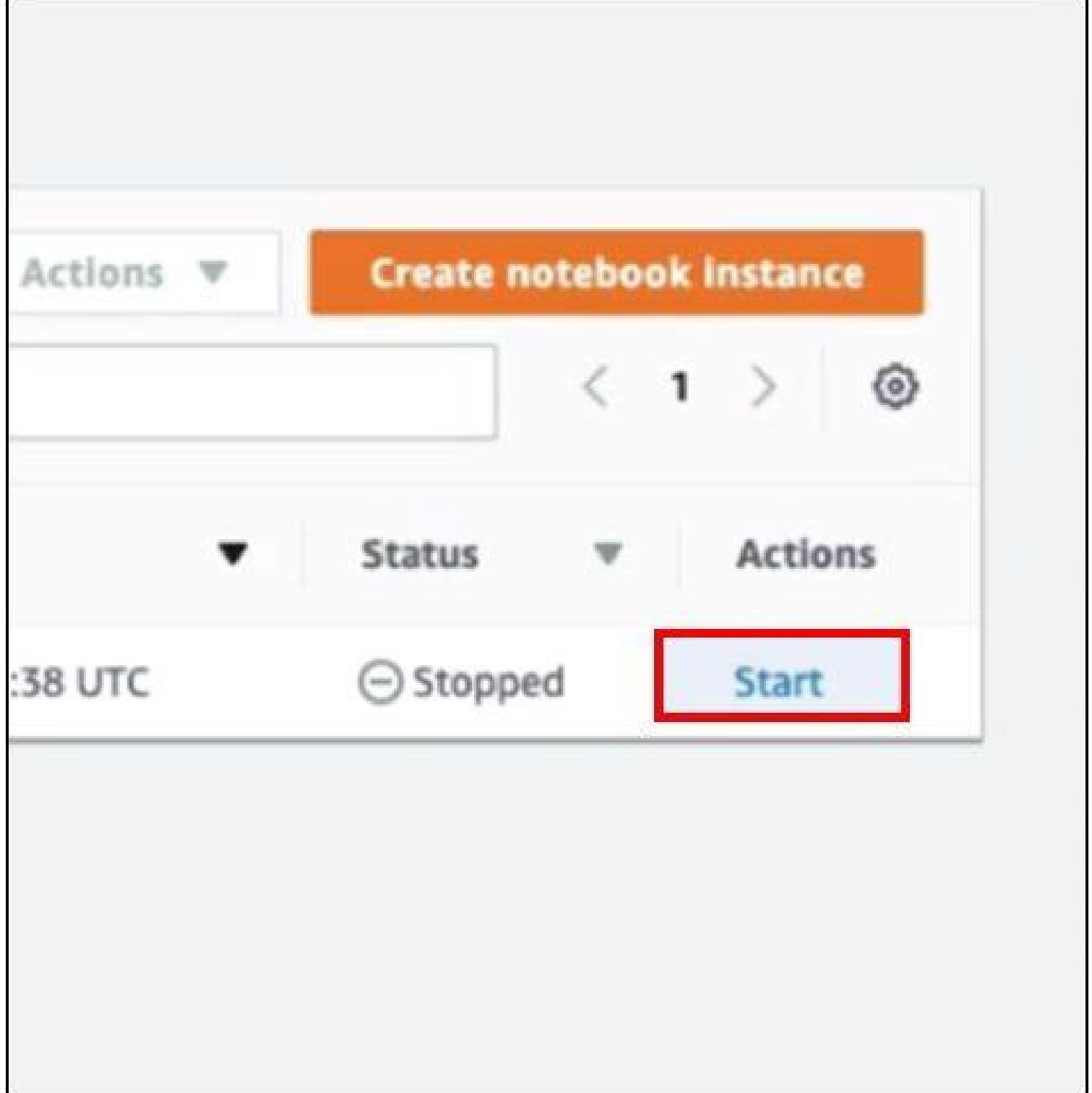
Amazon SageMaker > Notebook instances

Notebook instances

Actions ▾

Name	Instance	Creation time
ml-labs-notebook-instance	mLt2.medium	May 16, 2022 15:38 UTC

9. Click on **Start**.



10. Click on **Open Jupyter**.

Notebook instances

Actions ▾

Create notebook instance

Q Search notebook instances

< 1 /

Name

Instance

Creation time

Status

Actions

ml-labs-notebook-instance

ml.t2.medium

May 16, 2022 15:38 UTC

InService

Open Jupyter | Open JupyterLab



Files

Running

Clusters

SageMaker Examples

Conda

Select items to perform actions on them.

 0 / ufo-modeling-lab.ipynb model.tar.gz model_algo-1 state_b5c6eca3-3736-409d-9776-c3c271435d8811. Click on **Upload**.

 Upload	New	C
Name	Last Modified	File size
	6 days ago	36.4 kB
	3 hours ago	1.4 kB
	3 hours ago	152 B
	6 days ago	1.33 kB

12. Select **ufo-algorithms-lab.ipynb**.

Name

ufo-algorithms-lab.ipynb
ufo_fullset.csv



You can also download it from Github from the link given below:

https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter7/ufo-algorithms-lab.ipynb

13. Navigate into the **Jupyter notebook**.



Files

Running

Clusters

SageMaker Examples

Conda

Select items to perform actions on them.

0 /

- [ufo-algorithms-lab.ipynb](#)
- [ufo-modeling-lab.ipynb](#)
- [model.tar.gz](#)
- [model_algo-1](#)
- [state_b5c6eca3-3736-409d-9776-c3c271435d88](#)

Step 2: Import all the necessary libraries

1. Use the code given below to import all the necessary libraries.

```
In [ ]: import pandas as pd
import numpy as np
from datetime import datetime
import io
import sagemaker.amazon.common as smac

import boto3
from sagemaker import get_execution_role
import sagemaker

import matplotlib.pyplot as plt
import seaborn as sns
```

We are going to use NumPy and Pandas along with some other libraries. We are also going to use the Python SageMaker libraries that AWS offers. We are also going to use Matplotlib and Seaborn. These Python libraries allow us to build visualizations and graphs to help us analyze our data.

Step 3: Load the data from Amazon S3

We will load our data from S3 into the memory of our notebook instance.

1. Insert your bucket name in the code.

```
In [ ]: role = get_execution_role()
bucket='<INSERT_BUCKET_NAME_HERE>' ←
sub_folder = 'ufo_dataset'
data_key = 'ufo_fullset.csv'
data_location = 's3://{}//{}//{}'.format(bucket, sub_folder, data_key)

df = pd.read_csv(data_location, low_memory=False)
df.head()
```

2. Name your bucket, such as **ml-labs-ips**.

```
In [ ]: role = get_execution_role()
bucket='ml-labs-acg' ←
sub_folder = 'ufo_dataset'
data_key = 'ufo_fullset.csv'
data_location = 's3://{}//{}//{}'.format(bucket, sub_folder, data_key)

df = pd.read_csv(data_location, low_memory=False)
df.head()
```

3. Once we run this, it will download the dataset from S3 and load it into the memory of our notebook instance. We will store that into a data frame and look at what that data frame or our dataset looks like.

Out[2]:

	reportedTimestamp	eventDate	eventTime	shape	duration	witnesses	weather	firstName	lastName	latitude	longitude	sighting	physicalEvidence
0	1977-04-04T04:02:23.340Z	1977-03-31	23:46	circle	4	1	rain	Ila	Bashirian	47.329444	-122.578889	Y	N
1	1982-11-22T02:06:32.019Z	1982-11-15	22:04	disk	4	1	partly cloudy	Eriberto	Rundfsson	52.664913	-1.034894	Y	Y
2	1992-12-07T19:06:52.482Z	1992-12-07	19:01	circle	49	1	clear	Miller	Watsica	38.951667	-92.333889	Y	N
3	2011-02-24T21:06:34.898Z	2011-02-21	20:56	disk	13	1	partly cloudy	Clifton	Bechtelar	41.496944	-71.367778	Y	N
4	1991-03-09T16:18:45.501Z	1991-03-09	11:42	circle	17	1	mostly cloudy	Jayda	Ebert	47.606389	-122.330833	Y	N

4. We are trying to solve the problem of determining what the researchOutcome is.

Note: All of this is explained, but there are explained, unexplained, and probable options available.

Out[2]:

eventDate	eventTime	shape	duration	witnesses	weather	firstName	lastName	latitude	longitude	sighting	physicalEvidence	contact	researchOutcome
1977-03-31	23:46	circle	4	1	rain	Ila	Bashirian	47.329444	-122.578889	Y	N	N	explained
1982-11-15	22:04	disk	4	1	partly cloudy	Eriberto	Rundfsson	52.664913	-1.034894	Y	Y	N	explained
1992-12-07	19:01	circle	49	1	clear	Miller	Watsica	38.951667	-92.333889	Y	N	N	explained
2011-02-21	20:56	disk	13	1	partly cloudy	Clifton	Bechtelar	41.496944	-71.367778	Y	N	N	explained
1991-03-09	11:42	circle	17	1	mostly cloudy	Jayda	Ebert	47.606389	-122.330833	Y	N	N	explained

Step 4: Clean, transform, analyze, and prepare the dataset

1. We need to clean, transform, analyze, and start preparing our dataset. We need to check to see any missing values within our dataset. If there are missing values, then it will display those, and we can look at what we might need to do to fix our missing values.

```
In [ ]: # Let's check to see if there are any missing values
missing_values = df.isnull().values.any()
if(missing_values):
    display(df[df.isnull().any(axis=1)])
```

```
In [ ]: df['shape'].value_counts()
```

```
In [ ]: # Replace the missing values with the most common shape
df['shape'] = df['shape'].fillna(df['shape'].value_counts().index[0])
```

2. When we run the code, we get some missing values. If we look, the shape for a couple of these observations is null or not a number.

```
In [3]: # Let's check to see if there are any missing values  
missing_values = df.isnull().values.any()  
if(missing_values):  
    display(df[df.isnull().any(axis=1)])
```

	reportedTimestamp	eventDate	eventTime	shape	duration	witnesses	weather	firstName	lastName	latitude	longitude	sighting	ph
1024	2011-03-23T18:32:20.473Z	2011-03-22	21:12	NaN	3	1	rain	Deon	Fell	37.681944	-121.786944	Y	
2048	1998-04-23T18:47:16.029Z	1998-04-23	10:07	NaN	40	2	partly cloudy	Vincenzo	Rohan	38.254167	-85.759444	Y	

3. We will replace those missing values with the most common shape or the most common value. We could do this by getting the count for each shape; hence, we can see that a circle is the most common shape.

```
In [4]: df['shape'].value_counts()
```

```
Out[4]: circle      6047 ←  
        disk       5920  
        light      1699  
        square     1662  
        triangle   1062  
        sphere     1020  
        box        200  
        oval       199  
        pyramid    189  
Name: shape, dtype: int64
```

4. We will replace the missing values with the most common shape; in this case, it will replace those two missing values with the circle shape.

```
In [5]: # Replace the missing values with the most common shape ←  
df['shape'] = df['shape'].fillna(df['shape'].value_counts().index[0])
```

5. Now, we will start preparing our dataset by transforming some of the values into the correct data types.

We are going to:

- Convert the reportedTimestamp and eventDate to datetime data types
- Convert the shape and weather to a category data type

- Map the physicalEvidence and contact from 'Y,' 'N' to 0, 1
- Convert the researchOutcome to category data.type (target attribute)

Once we successfully run these, we should be able to see the data types in the new format.

```
In [ ]: df['reportedTimestamp'] = pd.to_datetime(df['reportedTimestamp'])
df['eventDate'] = pd.to_datetime(df['eventDate'])

df['shape'] = df['shape'].astype('category')
df['weather'] = df['weather'].astype('category')

df['physicalEvidence'] = df['physicalEvidence'].replace({'Y': 1, 'N': 0})
df['contact'] = df['contact'].replace({'Y': 1, 'N': 0})

df['researchOutcome'] = df['researchOutcome'].astype('category')

In [ ]: df.dtypes
```

6. If we type **df.dtypes**, we should be able to see the different data types that our attributes are in now.

```
In [7]: df.dtypes
```

Out[7]:	reportedTimestamp	datetime64[ns]
	eventType	datetime64[ns]
	eventTime	object
	shape	category
	duration	int64
	witnesses	int64
	weather	category
	firstName	object
	lastName	object
	latitude	float64
	longitude	float64
	sighting	object
	physicalEvidence	int64
	contact	int64
	researchOutcome	category
	dtype: object	

7. We can visualize some of our data to find any important information about that data. We might be

able to find things like outliers, or we might be able to find the distribution of our data. We may be able to find any imbalances in our data as well.

8. We will configure Matplotlib and the Seaborn library, so we can start drawing out some visualizations.

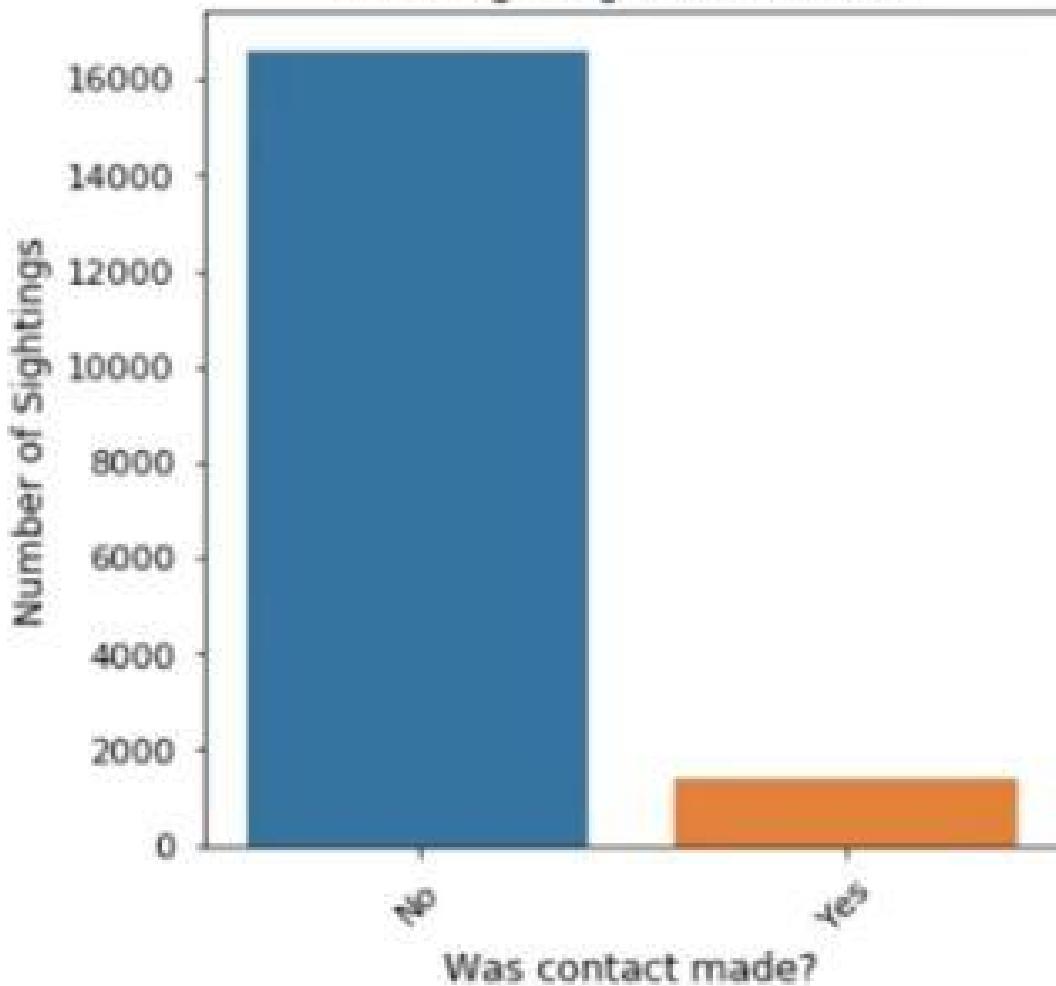
The first visualization is the UFO Sightings and Contact. Was contact made?

```
In [8]: %matplotlib inline  
sns.set_context("paper", font_scale=1.4)  
  
In [ ]: m_cts = (df['contact'].value_counts())  
m_ctsx = m_cts.index  
m_ctsy = m_cts.get_values()  
f, ax = plt.subplots(figsize=(5,5))  
  
sns.barplot(x=m_ctsx, y=m_ctsy)  
ax.set_title('UFO Sightings and Contact')  
ax.set_xlabel('Was contact made?')  
ax.set_ylabel('Number of Sightings')  
ax.set_xticklabels(['No', 'Yes'])  
plt.xticks(rotation=45)  
plt.show()
```



9. We can see that with most of our sightings, no contact was made.

UFO Sightings and Contact

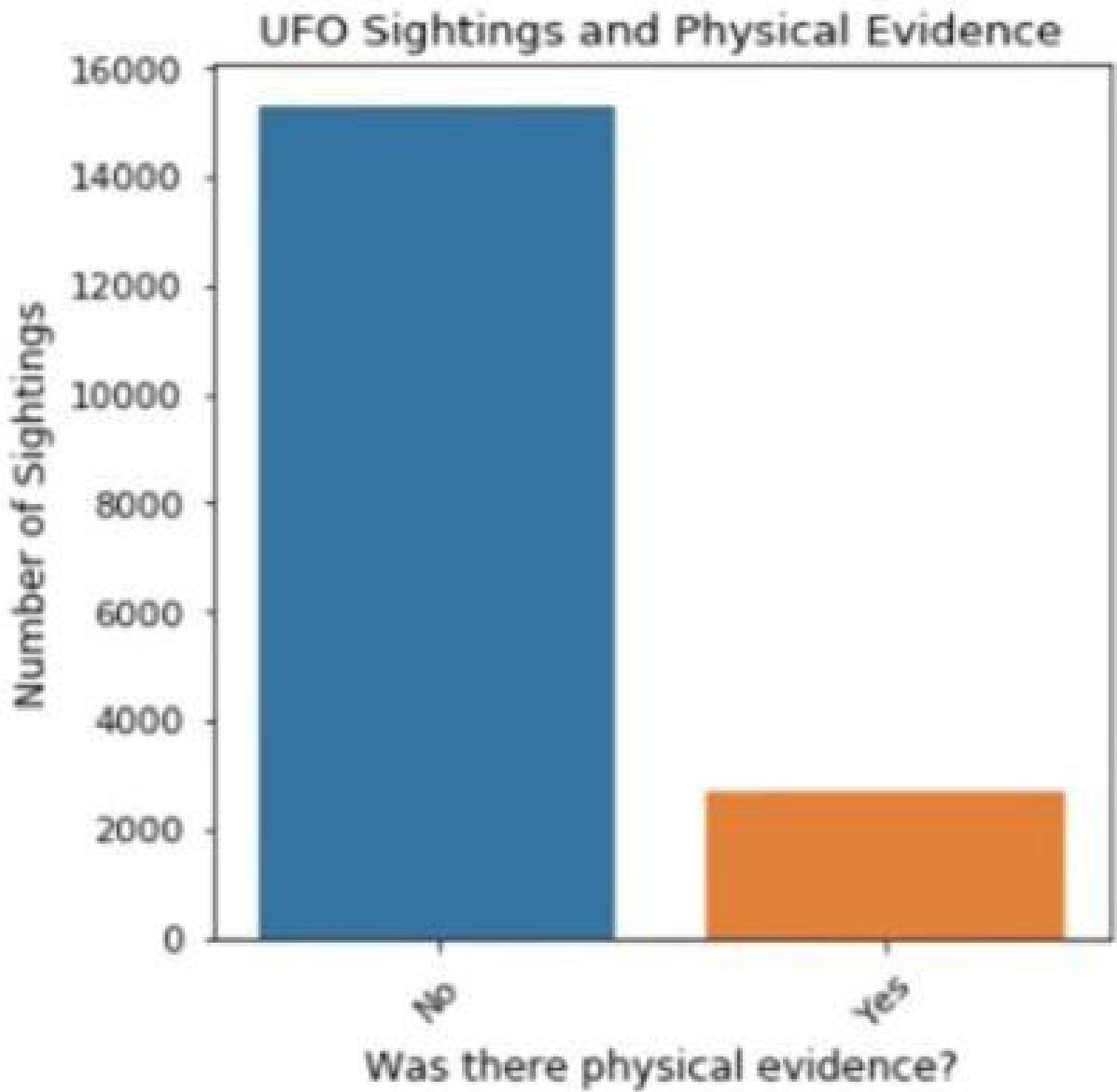


10. Now, to see how many sightings had some physical evidence associated with them.

```
In [ ]: m_cts = (df['physicalEvidence'].value_counts())
m_ctsx = m_cts.index
m_ctsy = m_cts.get_values()
f, ax = plt.subplots(figsize=(5,5))

sns.barplot(x=m_ctsx, y=m_ctsy)
ax.set_title('UFO Sightings and Physical Evidence')
ax.set_xlabel('Was there physical evidence?') ←
ax.set_ylabel('Number of Sightings')
ax.set_xticklabels(['No', 'Yes'])
plt.xticks(rotation=45)
plt.show()
```

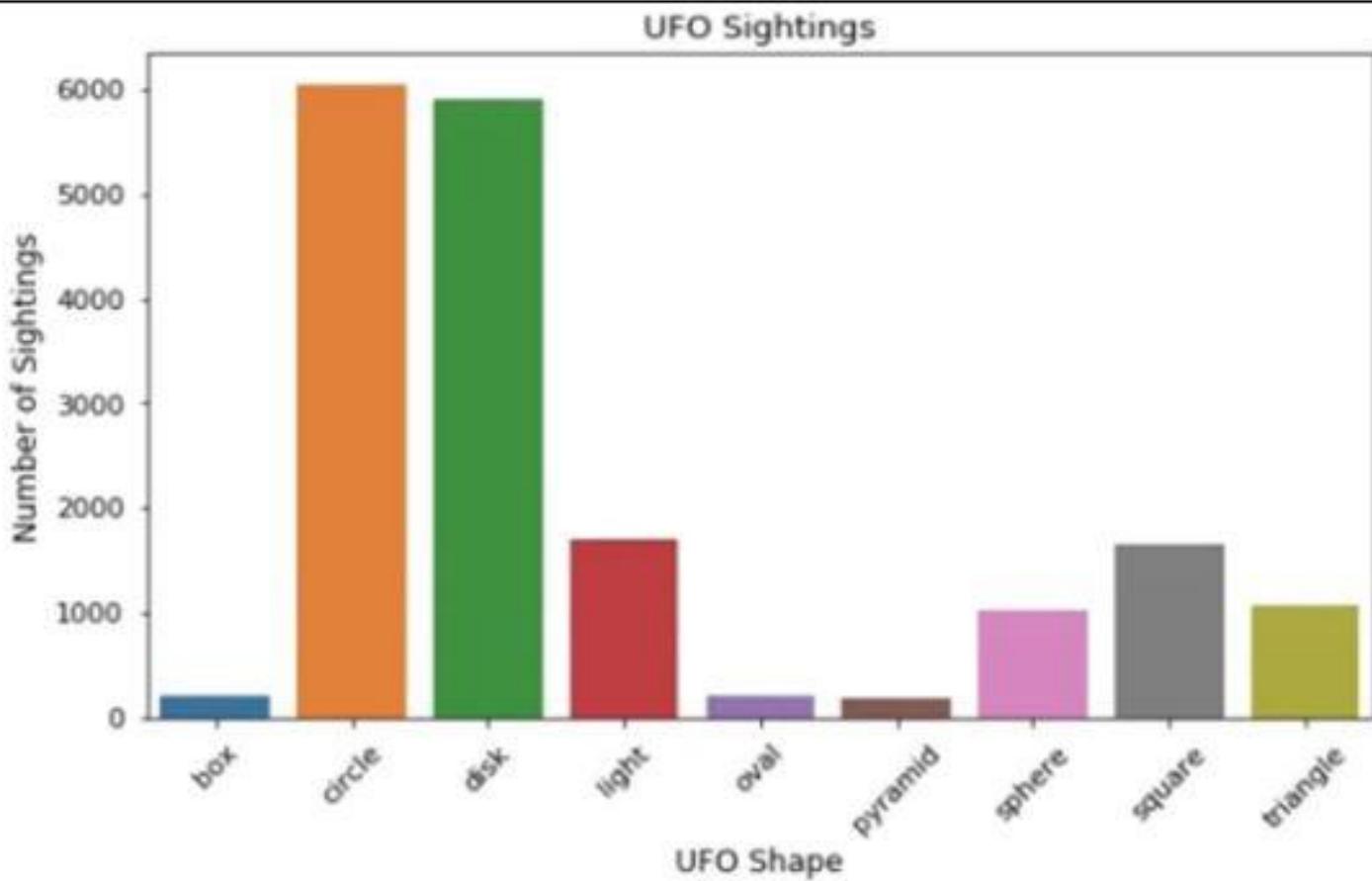
11. Once we create this visual, you can see that the majority, again, there was not much physical evidence left behind.



12. Next, look at the distribution of the different shapes for the UFO sightings.

```
In [ ]: m_cts = (df['shape'].value_counts())  
m_ctsx = m_cts.index  
m_ctsy = m_cts.get_values()  
f, ax = plt.subplots(figsize=(9,5))  
  
sns.barplot(x=m_ctsx, y=m_ctsy)  
ax.set_title('UFO Sightings')  
ax.set_xlabel('UFO Shape')  
ax.set_ylabel('Number of Sightings')  
plt.xticks(rotation=45)  
plt.show()
```

13. We can see that most of these sightings were either circles or disks, with a few more that were light and square. Then, they slowly decrease in value with the other shapes.

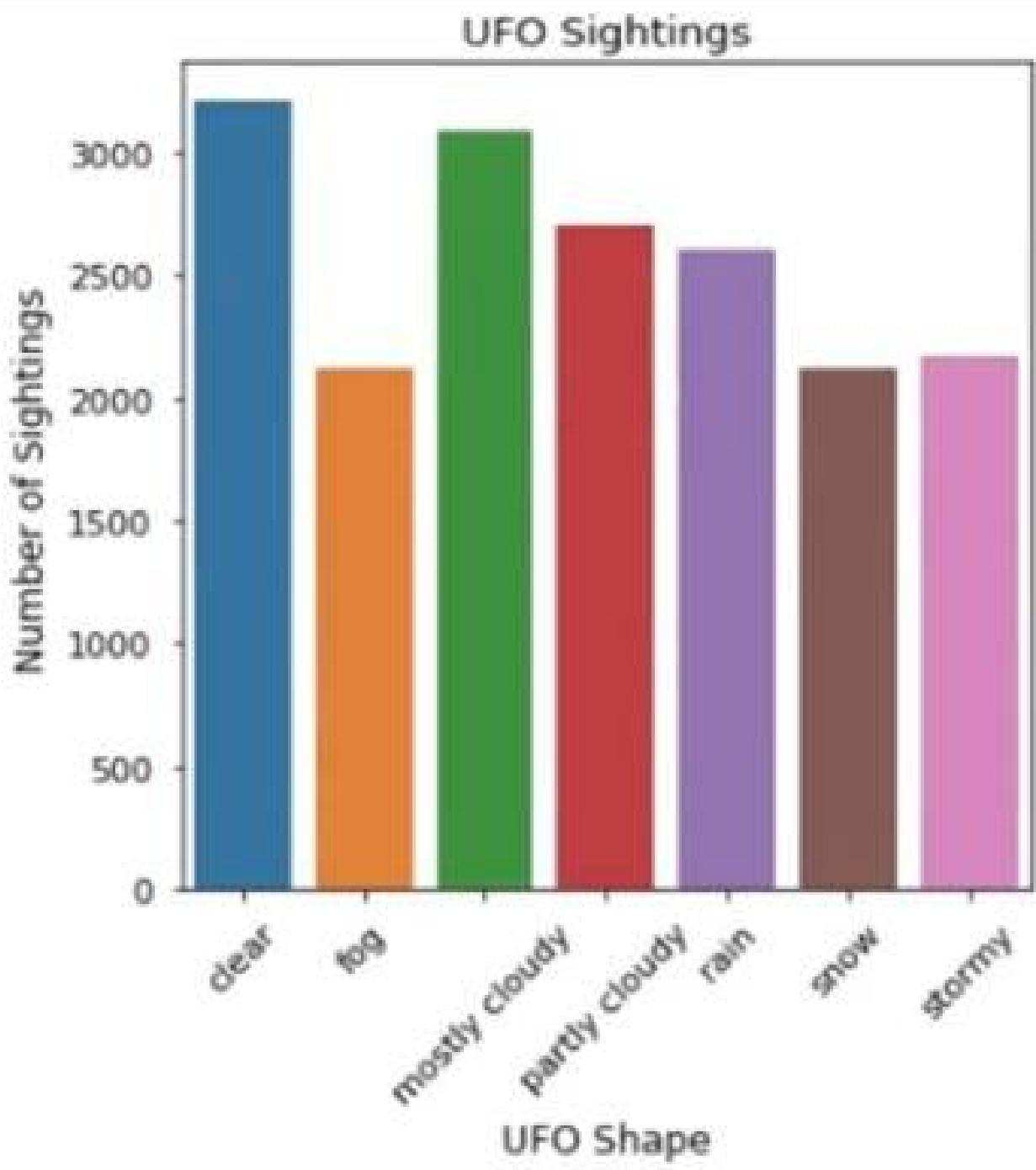


14. Now, to look at the distribution of weather, we can run the code below.

```
In [ ]: m_cts = (df['weather'].value_counts())
m_ctsx = m_cts.index
m_ctsy = m_cts.get_values()
f, ax = plt.subplots(figsize=(5,5))

sns.barplot(x=m_ctsx, y=m_ctsy)
ax.set_title('UFO Sightings')
ax.set_xlabel('UFO Shape')
ax.set_ylabel('Number of Sightings')
plt.xticks(rotation=45)
plt.show()
```

15. We can see here that most of the sightings were either clear or cloudy, and the other weather attributes follow. This distribution looks fairly even, so this may not correlate much with the outcome.

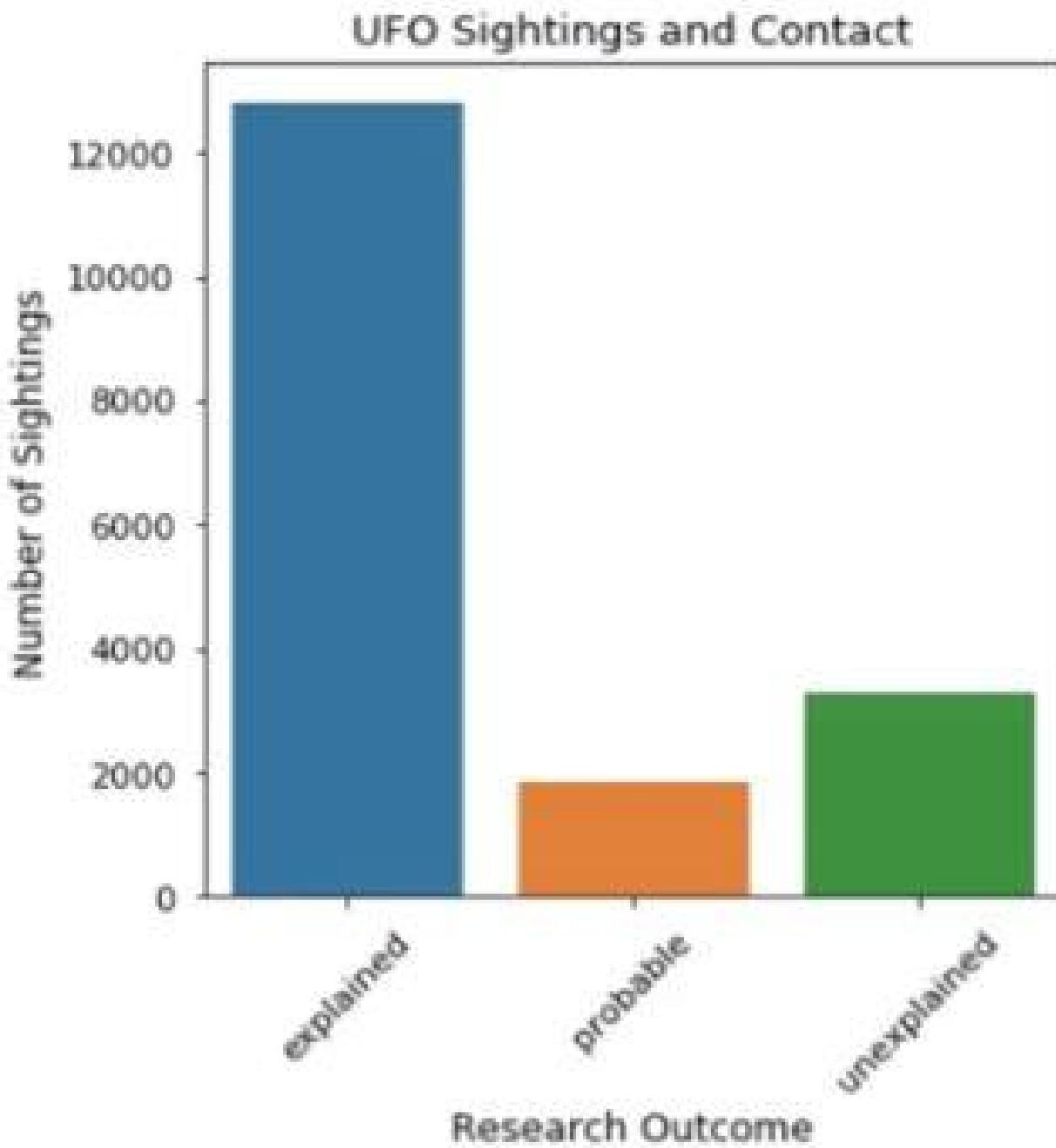


16. Now, we can run the code below to see what the researchOutcome distribution looks like.

```
In [ ]: m_cts = (df['researchOutcome'].value_counts())
m_ctsx = m_cts.index
m_ctsy = m_cts.get_values()
f, ax = plt.subplots(figsize=(5,5))

sns.barplot(x=m_ctsx, y=m_ctsy)
ax.set_title('UFO Sightings and Contact')
ax.set_xlabel('Research Outcome')
ax.set_ylabel('Number of Sightings')
# ax.set_xticklabels(['Unexplained', 'Explained', 'Probable'])
plt.xticks(rotation=45)
plt.show()
```

17. We can see the distribution in the researchOutcomes looks like the majority of our sightings were explained.



18. Looking at the distribution of sightings per year, we can run the code below.

```
In [ ]: ufo_yr = df['eventDate'].dt.year # series with the year exclusively
```

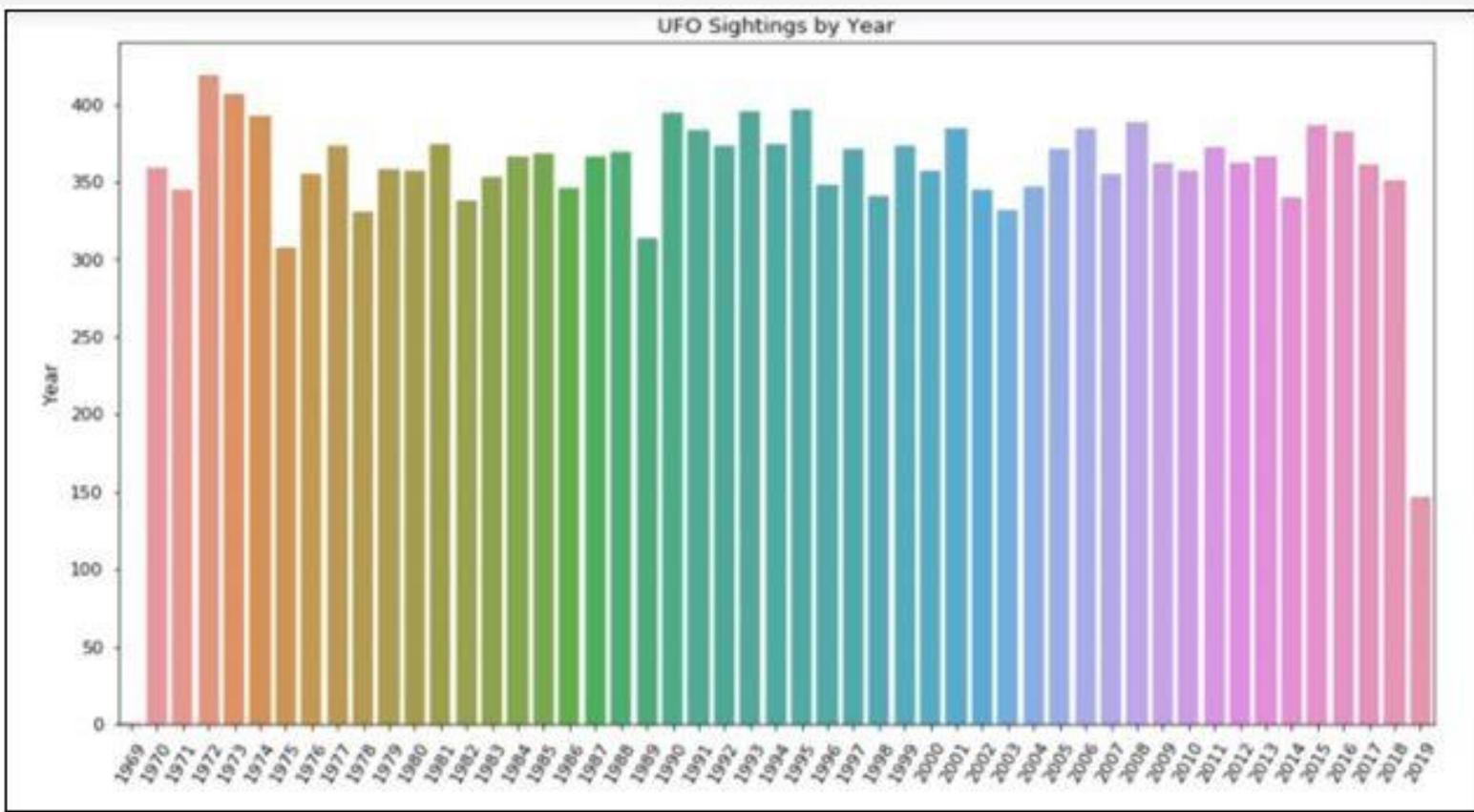
← Red arrow pointing to the line of code: ufo_yr = df['eventDate'].dt.year # series with the year exclusively

```
## Set axes ##
years_data = ufo_yr.value_counts()
years_index = years_data.index # x ticks
years_values = years_data.get_values()

## Create Bar Plot ##
plt.figure(figsize=(15,8))
plt.xticks(rotation = 60)
plt.title('UFO Sightings by Year')
plt.ylabel('Number of Sightings')
plt.xlabel('Year')

years_plot = sns.barplot(x=years_index[:60],y=years_values[:60])
```

19. You can see that the distribution has the sightings per year is fairly even. This may not be much of a deciding factor in what the researchOutcome is.



20. We are going to look at the correlation of these values.

In [15]: `df.corr()`

Out[15]:

	duration	witnesses	latitude	longitude	physicalEvidence	contact
duration	1.000000	0.020679	0.000243	-0.010529	0.016430	0.015188
witnesses	0.020679	1.000000	0.010229	0.003449	0.009186	-0.000651
latitude	0.000243	0.010229	1.000000	-0.394536	0.006465	0.004284
longitude	-0.010529	0.003449	-0.394536	1.000000	-0.004519	-0.004828
physicalEvidence	0.016430	0.009186	0.006465	-0.004519	1.000000	0.693276
contact	0.015188	-0.000651	0.004284	-0.004828	0.693276	1.000000

21. You can see some strong correlations between physical evidence and contact. So usually, there is physical evidence at least 69% of the time whenever there is contact. We need to include both of these within our training data. We do not see shape and weather because these are categorical variables, so we cannot find any correlation between them. We will include those within our training data as well.

22. We are going to drop the columns that are not important. If you look at the dataset, there are several columns that are not important to include within our training dataset. For instance, we can drop the sighting because it is always Yes. We can also drop the first and last names because they are unimportant when determining the researchOutcome. We can also drop the reportedTimeStamp. And,

we could create some buckets for the eventDate and the eventTime, for instance, the season or what quarter it was in. But since the distribution of the dates is even, we can drop those.

```
In [ ]: df.drop(columns=['firstName', 'lastName', 'sighting', 'reportedTimestamp', 'eventDate', 'eventTime'], inplace=True)  
In [ ]: df.head()
```

To drop these values, we can simply call df.drop and pass in the name of the columns we want to get rid of.

23. If we take a look at our dataset by calling df.head, we can see that these are the only attributes left within our dataset.

```
In [16]: df.drop(columns=['firstName', 'lastName', 'sighting', 'reportedTimestamp', 'eventDate'])  
In [17]: df.head()  
Out[17]:
```

	shape	duration	witnesses	weather	latitude	longitude	physicalEvidence	contact	researchOutcome
0	circle	4	1	rain	47.329444	-122.578889	0	0	explained
1	disk	4	1	partly cloudy	52.664913	-1.034894	1	0	explained
2	circle	49	1	clear	38.951667	-92.333889	0	0	explained
3	disk	13	1	partly cloudy	41.496944	-71.367778	0	0	explained
4	circle	17	1	mostly cloudy	47.806389	-122.330833	0	0	explained

24. The XGBoost and the Linear Learner are both expecting numeric inputs. Hence, shape, weather, and researchOutcome need to be converted into numeric values.

Therefore, we will hot encode the weather and the shape attribute and map the researchOutcome, or the target attribute, into numeric values.

25. We will assign unexplained to zero, explained to one, and probable to two. Our machine learning algorithm is going to take all of the numeric input attributes from our dataset and try to find some correlation or some mapping that can be done to determine the researchOutcome as zero, one, or two.

```
In [18]: # Let's one-hot the weather and shape attribute  
df = pd.get_dummies(df, columns=['weather', 'shape'])  
  
# Let's replace the researchOutcome values with 0, 1, 2 for Unexplained, Explained, and Probable  
df['researchOutcome'] = df['researchOutcome'].replace({'unexplained': 0, 'explained': 1, 'probable': 2})  
  
In [ ]: display(df.head())  
In [ ]: display(df.shape)
```

26. We can call df.head and df.shape to see what our dataset looks like now and how many attributes and columns we have.

```
In [19]: display(df.head())
display(df.shape)
```



	duration	witnesses	latitude	longitude	physicalEvidence	contact	researchOutcome	weather_clear	weather_fog	weather_mostly	...	weather_stormy
0	4	1	47.329444	-122.578889	0	0	1	0	0	0	...	0
1	4	1	52.664913	-1.034894	1	0	1	0	0	0	...	0
2	49	1	38.951667	-92.333889	0	0	1	1	0	0	...	0
3	13	1	41.496944	-71.367776	0	0	1	0	0	0	...	0
4	17	1	47.606389	-122.330833	0	0	1	0	0	1	...	0

5 rows × 23 columns

(18000, 23)

We can see here that we have one hot encoded the weather and the shape attribute, and every other attribute is in numeric format.

27. Before we load our data into our machine learning algorithm, we will randomize and split the data into a couple of different datasets. We will randomize the data and split it into training, validation, and a testing dataset. We will split the data into 80% training, 10% validation, and 10% testing.

```
In [ ]: # Let's go ahead and randomize our data.
df = df.sample(frac=1).reset_index(drop=True)

# Next, Let's split the data into a training, validation, and testing.
rand_split = np.random.rand(len(df))
train_list = rand_split < 0.8 # 80% for training
val_list = (rand_split >= 0.8) & (rand_split < 0.9) # 10% for validation
test_list = rand_split >= 0.9 # 10% for testing

# This dataset will be used to train the model.
data_train = df[train_list]

# This dataset will be used to validate the model.
data_val = df[val_list]

# This dataset will be used to test the model.
data_test = df[test_list]
```



You can see here that the first thing we do is randomize the data and then split it into three different datasets; data_train, data_val, and data_test.

28. Now, we need to rearrange our attributes, so the very first attribute is our target attribute or researchOutcome. The reason for doing this is that this is what XGBoost requires for the input data. It requires the first attribute in the dataset by the target attribute.

29. After rearranging those columns, we then want to store those as CSV files in our notebook instance. And then, we can load those files to S3 into their specific folders.

We will create a new folder called algorithms_lab and a sub_folder called xgboost_train and xgboost_validation. We will create the appropriate files for the training.csv and the validation.csv.

```
In [ ]: # Simply moves the researchOutcome attribute to the first position before creating CSV files
pd.concat([data_train['researchOutcome'], data_train.drop(['researchOutcome'], axis=1)], axis=1).to_csv('train.csv', index=False)
pd.concat([data_val['researchOutcome'], data_val.drop(['researchOutcome'], axis=1)], axis=1).to_csv('validation.csv', index=False)

# Next we can take the files we just stored onto our Notebook instance and upload them to S3.
boto3.Session().resource('s3').Bucket(bucket).Object('algorithms_lab/xgboost_train/train.csv').upload_file('train.csv')
boto3.Session().resource('s3').Bucket(bucket).Object('algorithms_lab/xgboost_validation/validation.csv').upload_file('validation.csv')
```

We are using CSV and not LibSVM because a data frame can quite easily be transformed into a CSV file by calling CSV.

Step 5: Creating and Training our model (XGBoost)

1. The first thing is to get the container hosted in the Elastic Container Repository within AWS. We will specify the region name and the algorithm for which we would like to get the container.

```
In [ ]: from sagemaker.amazon.amazon_estimator import get_image_uri
container = get_image_uri(boto3.Session().region_name, 'xgboost')
```

Next, because we're training with the CSV file format, we'll create inputs that our training function can use as a pointer to the files in S3, which also specify that the content type is CSV.

```
In [ ]: s3_input_train = sagemaker.s3_input(s3_data='s3://{}{}'.format(bucket), content_type='csv')
s3_input_validation = sagemaker.s3_input(s3_data='s3://{}{}'.format(bucket), content_type='csv')
```

We will create a pointer that points to our S3 files for our training and our validation data.

2. First, we will create a specific job name with the job we are about to run.

```
In [ ]: # Create a training job name
job_name = 'ufo-xgboost-job-{}'.format(datetime.now().strftime("%Y%m%d%H%M%S"))
print('Here is the job name {}'.format(job_name))

# Here is where the model artifact will be stored
output_location = 's3://{}{}'.format(bucket, 'algorithms_lab/xgboost_output')
```

We can run the cell and see the job name.

```
In [24]: # Create a training job name
job_name = 'ufo-xgboost-job-{}'.format(datetime.now().strftime("%Y%m%d%H%M%S"))
print('Here is the job name {}'.format(job_name))

# Here is where the model artifact will be stored
output_location = 's3://{}{}'.format(bucket, 'algorithms_lab/xgboost_output')

Here is the job name ufo-xgboost-job-20190523170413
```

3. Next, we can specify the training parameters we want to use with the XGBoost algorithm. We will pass in the algorithm container, IAM role, training instance type, and the count. In our case, we will

use an ml.m4.xlarge as our training instance type and only use one of them. We will also pass in the S3 location for our output model artifact. We will also pass in the XGBoost hyperparameters. These are the only hyperparameters we will include within our XGBoost algorithm.

We can see that our objective is going to be multisoftmax. We could use softprob, but since we have a multi-classification problem, we need to set this to multisoftmax or multisoftprob. Both of these will give us our desired output.

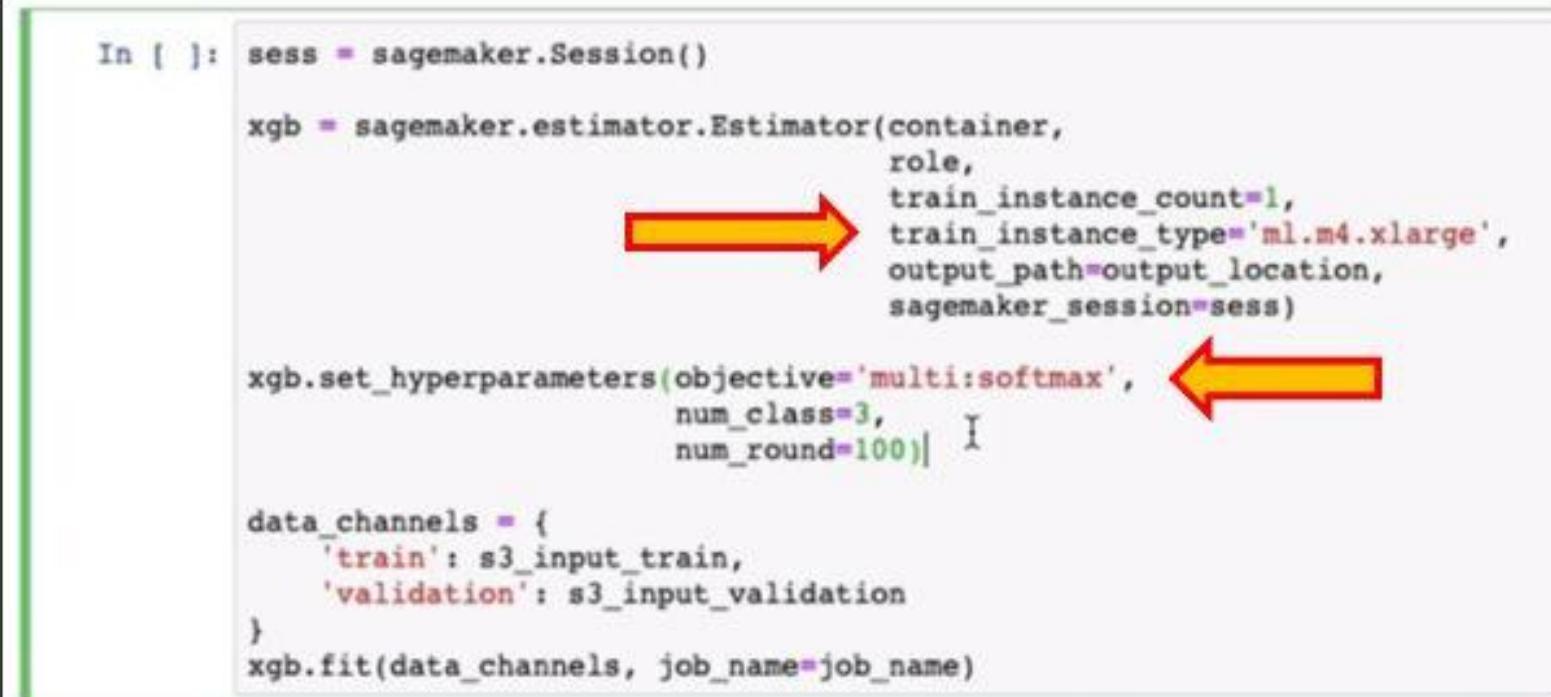
We will also include the number of classes, which is three, for explained, unexplained, and probable. We will also pass in the num_rounds, the number of rounds we want our algorithm to pass over our data.

```
In [ ]: sess = sagemaker.Session()

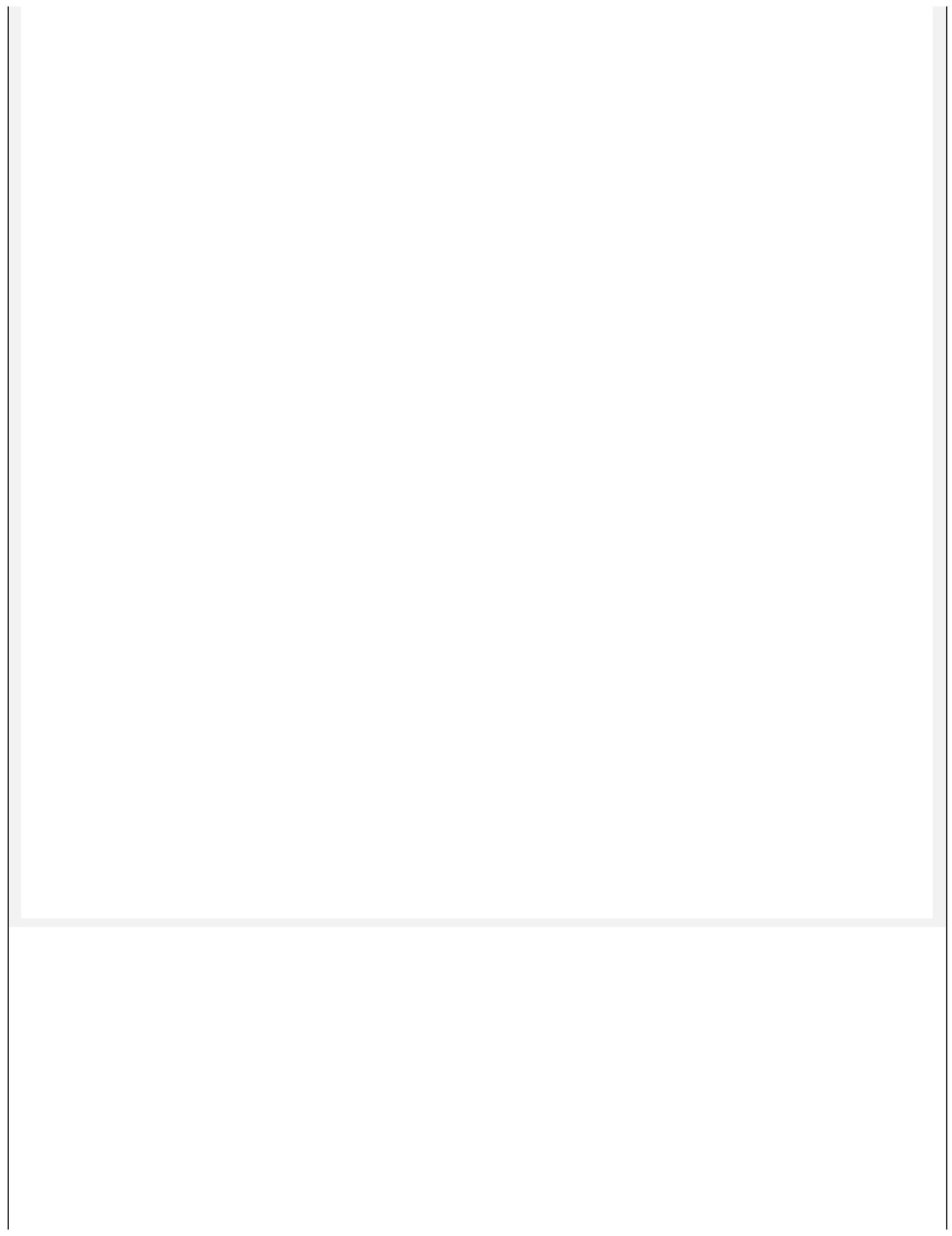
xgb = sagemaker.estimator.Estimator(container,
                                      role,
                                      train_instance_count=1,
                                      train_instance_type='ml.m4.xlarge',
                                      output_path=output_location,
                                      sagemaker_session=sess)

xgb.set_hyperparameters(objective='multi:softmax',
                        num_class=3,    I
                        num_round=100) | I

data_channels = {
    'train': s3_input_train,
    'validation': s3_input_validation
}
xgb.fit(data_channels, job_name=job_name)
```



4. To start the job and our training process, we can use the .fit method and pass in the training and validation data.
5. You can see that the training job has started.



```
2019-05-23 17:11:21 Starting - Starting the training job...
2019-05-23 17:11:23 Starting - Launching requested ML instances.....
2019-05-23 17:12:33 Starting - Preparing the instances for training.....
2019-05-23 17:14:17 Downloading - Downloading input data
2019-05-23 17:14:17 Training - Downloading the training image.
Arguments: train
[2019-05-23:17:14:20:INFO] Running standalone xgboost training.
[2019-05-23:17:14:20:INFO] File size need to be processed in the node: 1.08mb. Available memory size in the node: 840
7.47mb
[2019-05-23:17:14:20:INFO] Determined delimiter of CSV input is ','
[17:14:20] S3DistributionType set as FullyReplicated
[17:14:20] 14368x22 matrix with 318096 entries loaded from /opt/ml/input/data/train?format=csv&label_column=0&deli
mter=","
[2019-05-23:17:14:20:INFO] Determined delimiter of CSV input is ','
[17:14:20] S3DistributionType set as FullyReplicated
[17:14:20] 1816x22 matrix with 39952 entries loaded from /opt/ml/input/data/validation?format=csv&label_column=0&deli
mter=","
[17:14:20] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 94 extra nodes, 0 pruned nodes, max_depth=6
[17:14:20] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 114 extra nodes, 0 pruned nodes, max_depth=6
```

6. As we scroll down, we can see the model's output as it was training. Here, we have 0.03, which is about a 3% error; for validation, we can see that it has about a 6% error. That is around 94 to 97% accuracy.

```
[42]#011train-merror:0.03306#011validation-merror:0.064978
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[43]#011train-merror:0.032642#011validation-merror:0.066079
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[44]#011train-merror:0.032085#011validation-merror:0.065529
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[45]#011train-merror:0.0317#011validation-merror:0.065529
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[46]#011train-merror:0.030624#011validation-merror:0.064427
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
[17:14:22] src/tree/updater_prune.cc:74: tree pruning end, 1 roots,
```

7. We will continue to scroll down and see what the final output was. You can see that the final output was about .8% training error, which is about 99% accuracy. And about a 7% error, which is around 93% accuracy.

```
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 80 extra nodes,  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 76 extra nodes,  
[96]#01ltrain-merror:0.009883#01lvalidation-merror:0.071035  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 78 extra nodes,  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 64 extra nodes,  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 64 extra nodes,  
[97]#01ltrain-merror:0.009535#01lvalidation-merror:0.071586  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 76 extra nodes,  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 74 extra nodes,  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 98 extra nodes,  
[98]#01ltrain-merror:0.009187#01lvalidation-merror:0.071586  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 70 extra nodes,  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 78 extra nodes,  
[17:14:23] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 56 extra nodes,  
[99]#01ltrain-merror:0.008561#01lvalidation-merror:0.071586
```

2019-05-23 17:15:02 Uploading - Uploading generated training model

2019-05-23 17:15:02 Completed - Training job completed

Billable seconds: 61

This is exactly what we need. We have some data to show to our organization.

8. Now, we can print out the location of our trained model. If we look at this location S3, we can see our model artifact, and we can use this model artifact for inference to deploy it into a production environment later.

```
In [26]: print('Here is the location of the trained XGBoost model: {}//{}//output/model.tar.gz'.format(output_location, job_name))  
Here is the location of the trained XGBoost model: s3://ml-labs-acq/algorithms_lab/xgboost_output/ufo-xgboost-job-20190523170413/output/model.tar.gz
```



Step 6: Create and train our model (Linear Learner)

Now, we can start creating and training a Linear Learner model, where we can start to compare the accuracy between the two.

1. Since our data is already prepared and ready to go, we will randomize and split the data again for the Linear Learner algorithm.
2. We will split the data into three different datasets, a training, validation, and testing dataset. It will be 80% training, 10% validation, and 10% testing.

```
In [ ]: np.random.seed(0)
rand_split = np.random.rand(len(df))
train_list = rand_split < 0.8
val_list = (rand_split >= 0.8) & (rand_split < 0.9)
test_list = rand_split >= 0.9

# This dataset will be used to train the model.
data_train = df[train_list]

# This dataset will be used to validate the model.
data_val = df[val_list]

# This dataset will be used to test the model.
data_test = df[test_list]
```

3. It will create a data frame for each dataset type, and it will rearrange the researchOutcome to where it is the very first column within the datasets for each of them.

We will also break the datasets into NumPy ndarrays. This is getting our data ready for the recordIO-protobuf. These are the necessary steps to get it ready for that format.

```
# This rearranges the columns
cols = list(data_train)
cols.insert(0, cols.pop(cols.index('researchOutcome')))
data_train = data_train[cols]

cols = list(data_val)
cols.insert(0, cols.pop(cols.index('researchOutcome')))
data_val = data_val[cols]

cols = list(data_test)
cols.insert(0, cols.pop(cols.index('researchOutcome')))
data_test = data_test[cols]

# Breaks the datasets into attribute numpy.ndarray and the same for target attribute.
train_X = data_train.drop(columns='researchOutcome').as_matrix()
train_y = data_train['researchOutcome'].as_matrix()

val_X = data_val.drop(columns='researchOutcome').as_matrix()
val_y = data_val['researchOutcome'].as_matrix()

test_X = data_test.drop(columns='researchOutcome').as_matrix()
test_y = data_test['researchOutcome'].as_matrix()
```

4. Now, we will create a recordIO file for training data and upload it to S3.

```
In [ ]: train_file = 'ufo_sightings_train_recordIO_protobuf.data'
f = io.BytesIO()
smac.write_numpy_to_dense_tensor(f, train_X.astype('float32'), train_y.astype('float32'))
f.seek(0)

boto3.Session().resource('s3').Bucket(bucket).Object('algorithms_lab/linearlearner_train/{}'.format(train_file)).upload_file(training_recordIO_protobuf_location = 's3://{}//algorithms_lab/linearlearner_train/{}'.format(bucket, train_file))
print('The Pipe mode recordIO protobuf training data: {}'.format(training_recordIO_protobuf_location))
```



It is building a recordIO-protobuf for that dataset, which in our case is the training dataset, by taking the x-values, which are everything but the researchOutcome, and the y-values, which are only the researchOutcome.

5. When we run the code, we can see this is the location of where our recordIO-protobuf is stored.

```
In [28]: train_file = 'ufo_sightings_train_recordIO_protobuf.data'
f = io.BytesIO()
smac.write_numpy_to_dense_tensor(f, train_X.astype('float32'), train_y.astype('float32'))
f.seek(0)

boto3.Session().resource('s3').Bucket(bucket).Object('algorithms_lab/linearlearner_train/{}'.format(train_file)).upload_file(training_recordIO_protobuf_location = 's3://{}//algorithms_lab/linearlearner_train/{}'.format(bucket, train_file))
print('The Pipe mode recordIO protobuf training data: {}'.format(training_recordIO_protobuf_location))

The Pipe mode recordIO protobuf training data: s3://ml-labs-acg/algorithms_lab/linearlearner_train/ufo_sightings_train_recordIO_protobuf.data
```

6. We are going to perform the same steps for validation data.

When we run the code, we can see the S3 location of our validation data in recordIO-protobuf format.

```
In [29]: validation_file = 'ufo_sightings_validation_recordIO_protobuf.data'
f = io.BytesIO()
smac.write_numpy_to_dense_tensor(f, val_X.astype('float32'), val_y.astype('float32'))
f.seek(0)

boto3.Session().resource('s3').Bucket(bucket).Object('algorithms_lab/linearlearner_validation/{}'.format(validation_file)).upload_file(validate_recordIO_protobuf_location = 's3://{}//algorithms_lab/linearlearner_validation/{}'.format(bucket, validation_file))
print('The Pipe mode recordIO protobuf validation data: {}'.format(validate_recordIO_protobuf_location))

The Pipe mode recordIO protobuf validation data: s3://ml-labs-acg/algorithms_lab/linearlearner_validation/ufo_sightings_validation_recordIO_protobuf.data
```

7. We will go through the same setup we did for the XGBoost algorithm, but we will use the Linear Learner algorithm.

We are going to get everything we need from the ECR repository. This will create a container containing the Linear Learner docker image that we can use to train our model with.

```
In [ ]: from sagemaker.amazon.amazon_estimator import get_image_uri
import sagemaker

container = get_image_uri(boto3.Session().region_name, 'linear-learner', "1")
```

8. We will create a custom job name and specify the output location for our model artifact.

```
In [ ]: # Create a training job name  
job_name = 'ufo-linear-learner-job-{}'.format(datetime.now().strftime("%Y%m%d%H%M%S"))  
print('Here is the job name {}'.format(job_name))  
  
# Here is where the model-artifact will be stored  
output_location = 's3://{}{}/algorithms_lab/linearlearner_output'.format(bucket)
```

9. This will be the name of the training job we are about to create.

```
In [31]: # Create a training job name  
job_name = 'ufo-linear-learner-job-{}'.format(datetime.now().strftime("%Y%m%d%H%M%S"))  
print('Here is the job name {}'.format(job_name))  
  
# Here is where the model-artifact will be stored  
output_location = 's3://{}{}/algorithms_lab/linearlearner_output'.format(bucket)  
  
Here is the job name ufo-linear-learner-job-20190523180102
```



10. We will call the SageMaker library to create an estimator, which is a way to create a machine learning model by inputting the container and some other parameters about the algorithm we want to use. We can include the container, the IAM role, the training instance count, and the training instance type, and AWS recommends using a CPU instance. Hence, we will use an ml.c4.xlarge to train our model. We can specify the output location where the model artifact will live. We will pass in the SageMaker session and include the input_mode type of Pipe.

We can take advantage of using recordIO-protobuf as the input type and stream the data directly from S3. This will optimize the streaming process of getting the data from S3 onto our notebook instance.

```
In [ ]: print('The feature_dim hyperparameter needs to be set to {}'.format(data_train.shape[1] - 1))  
  
In [ ]: sess = sagemaker.Session()  
  
# Setup the LinearLearner algorithm from the ECR container  
linear = sagemaker.estimator.Estimator(container,  
                                         role,  
                                         train_instance_count=1,  
                                         train_instance_type='ml.c4.xlarge',  
                                         output_path=output_location,  
                                         sagemaker_session=sess,  
                                         input_mode='Pipe')  
  
# Setup the hyperparameters
```

11. We can specify all the hyperparameters associated with the Linear Learner algorithm. We can scroll down and take a look at which ones are required. The ones that are required are the featured dimensions. In our case, it will be the number of attributes minus the researchOutcome, so we have 22 total features or 22 total attributes.

12. We will specify the predictor type, which is a multiclass_classifier because this is a multi-classification problem. We will also specify the number of classes, similar to the XGBoost, we have explained unexplained and probable. These are the only hyperparameters that are required for the Linear Learner algorithm.

```

# Setup the hyperparameters
linear.set_hyperparameters(feature_dim=22, # number of attributes (minus the researchOutcome attribute)
                           predictor_type='multiclass_classifier', # type of classification problem
                           num_classes=3) # number of classes in our researchOutcome (explained, unexplained, probable)

# Launch a training job. This method calls the CreateTrainingJob API call
data_channels = {
    'train': training_recordIO_protobuf_location,
    'validation': validate_recordIO_protobuf_location
}
linear.fit(data_channels, job_name=job_name)

```

13. We will leave the others as default and see what our results look like. We will call the .fit method, pass in our training data, which in this case is recordIO-protobuf, and pass in the job name that we created earlier.

```

# Launch a training job. This method calls the CreateTrainingJob API call
data_channels = {
    'train': training_recordIO_protobuf_location,
    'validation': validate_recordIO_protobuf_location
}
linear.fit(data_channels, job_name=job_name)

```

2019-05-23 18:05:07 Starting - Starting the training job. 

14. The first thing you can do is the default configurations for the hyperparameters. So if we look at this JSON file or this JSON data, we should be able to see each hyperparameter that the Linear Learner algorithm uses to create the model.

```

2019-05-30 13:17:16 Downloading - Downloading input data...
2019-05-30 13:17:27 Training - Downloading the training image...
2019-05-30 13:18:17 Training - Training image download completed. Training in progress..
Docker entrypoint called with argument(s): train
[05/30/2019 13:18:19 INFO 139734425163584] Reading default configuration from /opt/amazon/lib/python2.7/site-packages
s/algorithms/default-input.json: {"loss_insensitivity": "0.01", "epochs": "15", "init_bias": "0.0", "lr_scheduler_factor": "auto", "use_bias": "true", "num_point_for_scaler": "10000", "log_level": "info", "quantile": "0.5", "bias_lr_mult": "auto", "lr_scheduler_step": "auto", "init_method": "uniform", "init_sigma": "0.01", "lr_scheduler_minimum_lr": "auto", "target_recall": "0.9", "num_models": "auto", "early_stopping_patience": "3", "momentum": "auto", "unbias_label": "auto", "wd": "auto", "optimizer": "auto", "tuning_objective_metric": "0", "early_stopping_tolerance": "0.001", "learning_rate": "auto", "kvstore": "auto", "normalize_data": "true", "binary_classifier_model_selection_criteria": "accuracy", "use_lr_scheduler": "true", "target_precision": "0.8", "unbias_data": "auto", "init_scale": "0.07", "bias_wd_mult": "auto", "f_beta": "1.0", "mini_batch_size": "1000", "huber_delta": "1.0", "num_classes": "1", "beta_1": "auto", "loss": "auto", "beta_2": "auto", "enable_profile": "false", "normalize_label": "auto", "num_gpus": "auto", "balance_multiclass_weights": "false", "positive_example_weight_mult": "1.0", "ll": "auto", "margin": "1.0"}
[05/30/2019 13:18:19 INFO 139734425163584] Reading provided configuration from /opt/ml/input/config/hyperparameters.json: {"feature_dim": "22", "predictor_type": "multiclass_classifier", "num_classes": "3"}

```

15. In our case, we can see the hyperparameters that we input. We can see num_classes.

```
m_lr': u'auto', u'target_recall': u'0.8', u'num_models': u'auto', u'early_stopping_patience': u'to', u'unbias_label': u'auto', u'wd': u'auto', u'optimizer': u'auto', u'_tuning_objective_metric': u'ing_tolerance': u'0.001', u'learning_rate': u'auto', u'_kvstore': u'auto', u'normalize_data': u'ifier_model_selection_criteria': u'accuracy', u'use_lr_scheduler': u'true', u'target_precision': a': u'auto', u'init_scale': u'0.07', u'bias_wd_mult': u'auto', u'f_beta': u'1.0', u'mini_batch_size': r_delta': u'1.0', u'num_classes': u'1', u'beta_1': u'auto', u'loss': u'auto', u'beta_2': u'auto', r': u>false', u'normalize_label': u'auto', u'_num_gpus': u'auto', u'balance_multiclass_weights': _example_weight_mult': u'1.0', u'll': u'auto', u'margin': u'1.0'}  
[05/30/2019 13:18:19 INFO 139734425163584] Reading provided configuration from /opt/ml/input/config: {u'feature_dim': u'22', u'predictor_type': u'multiclass_classifier' u'num_classes': u'3'}  
[05/30/2019 13:18:19 INFO 139734425163584] Final configuration: {u'loss_in sensitivity': u'0.01', u'feature_dim': u'22', u'init_bias': u'0.0', u'lr_scheduler_factor': u'auto', u'num_calibration': 0', u'accuracy_top_k': u'3', u'_num_kv_servers': u'auto', u'use_bias': u'true', u'num_point_for': u'_log_level': u'info', u'quantile': u'0.5', u'bias_lr_mult': u'auto', u'lr_scheduler_step': u'd': u'uniform', u'init_sigma': u'0.01', u'lr_scheduler_minimum_lr': u'auto', u'target_recall': u'auto', u'early_stopping_patience': u'3', u'momentum': u'auto', u'unbias_label': u'auto', u'wd': r': u'auto', u'_tuning_objective_metric': u'', u'early_stopping_tolerance': u'0.001', u'learning': vstore': u'auto', u'normalize_data': u'true', u'binary_classifier_model_selection_criteria': u'cheduler': u'true', u'target_precision': u'0.8', u'unbias_data': u'auto', u'init_scale': u'0.07', u'uto'. u'f_beta': u'1.0'. u'mini_batch_size': u'1000'. u'huber_delta': u'1.0'. u'num_classes': u'
```

16. We can also see the featured dimensions.

```
m_lr': u'auto', u'target_recall': u'0.8', u'num_models': u'auto', u'unbias_label': u'auto', u'wd': u'auto', u'optimizer': u'_ing_tolerance': u'0.001', u'learning_rate': u'auto', u'_kvstore': ifier_model_selection_criteria': u'accuracy', u'use_lr_scheduler': a': u'auto', u'init_scale': u'0.07', u'bias_wd_mult': u'auto', r_delta': u'1.0', u'num_classes': u'1', u'beta_1': u'auto', u'lr': u>false', u'normalize_label': u'auto', u'_num_gpus': u'auto', _example_weight_mult': u'1.0', u'll': u'auto', u'margin': u'1.0'}  
[05/30/2019 13:18:19 INFO 139734425163584] Reading provided config: {u'feature_dim': u'22', u'predictor_type': u'multiclass_classifier'}  
[05/30/2019 13:18:19 INFO 139734425163584] Final configuration: {u'feature_dim': u'22', u'init_bias': u'0.0', u'lr_scheduler_factor': u'auto', u'accuracy_top_k': u'3', u'_num_kv_servers': u'auto', u'use_bias': u'true', u'_log_level': u'info', u'quantile': u'0.5', u'bias_lr_mult': u'd': u'uniform', u'init_sigma': u'0.01', u'lr_scheduler_minimum_lr': u'auto', u'early_stopping_patience': u'3', u'momentum': u'auto', r': u'auto', u'_tuning_objective_metric': u'', u'early_stopping_tolerance': u'0.001', vstore': u'auto', u'normalize_data': u'true', u'binary_classifier_model_selection_criteria': u'cheduler': u'true', u'target_precision': u'0.8', u'unbias_data': u'uto'. u'f_beta': u'1.0'. u'mini_batch_size': u'1000'. u'huber_delta': u'1.0'. u'num_classes': u'3'}
```

17. If we scroll down, we can see the output from our Linear Learner algorithm. The metrics that mean a lot to us are the validation_score metric and the quality_metric.

18. We can start to see the multiclass_accuracy is around 0.94%, which is around 94% accuracy. And

we can also see recall, precision, and the F1 score.

```
[2019-05-30 13:18:25.929] [tensorio] [info] data_pipeline_stats={"name": "/opt/ml/input/data/validation", "epoch": 1, "duration": 53, "num_examples": 2}
[05/30/2019 13:18:25 INFO 139734425163584] #validation score (algo-1): {'multiclass_cross_entropy_objective': 0.19592909105605347}
[05/30/2019 13:18:25 INFO 139734425163584] #validation score (algo-1): {'validation score': algo-1, 'metric': 'multiclass_accuracy', value: 0.94472645234066553}
[05/30/2019 13:18:25 INFO 139734425163584] #validation score (algo-1): {'validation score': algo-1, 'metric': 'multiclass_top_k_accuracy_3', value: 1.0}
[05/30/2019 13:18:25 INFO 139734425163584] #validation score (algo-1): {'dcg': 0.97768016972248661}
[05/30/2019 13:18:25 INFO 139734425163584] #validation score (algo-1): {'macro_recall': 0.9270454}
[05/30/2019 13:18:25 INFO 139734425163584] #validation score (algo-1): {'macro_precision': 0.89689428}
[05/30/2019 13:18:25 INFO 139734425163584] #validation score (algo-1): {'macro_f_1_500': 0.91093159}
[05/30/2019 13:18:25 INFO 139734425163584] #quality_metric: host=algo-1, validation multiclass_cross_entropy_objective <loss>=0.195929091056
[05/30/2019 13:18:25 INFO 139734425163584] #quality_metric: host=algo-1, validation multiclass_accuracy <score>=0.94472645234066553
[05/30/2019 13:18:25 INFO 139734425163584] #quality_metric: host=algo-1, validation multiclass_top_k_accuracy_3 <score>=1.0
[05/30/2019 13:18:25 INFO 139734425163584] #quality_metric: host=algo-1, validation dcg <score>=0.97768016972248661
[05/30/2019 13:18:25 INFO 139734425163584] #quality_metric: host=algo-1, validation macro_recall <score>=0.9270454
11
```

19. If we scroll down all the way to the bottom, we can see that the training job is completed.

```
[2019-05-30 13:18:25.993] [tensorio] [info] data_pipeline_stats={"name": "/opt/ml/input/data/validation", "epoch": 1, "duration": 54, "num_examples": 2}
[2019-05-30 13:18:25.993] [tensorio] [info] data_pipeline_stats={"name": "/opt/ml/input/data/validation", "duration": 6663, "num_epochs": 13, "num_examples": 25}
#metrics ('Metrics': {"totaltime": {"count": 1, "max": 6755.892992019653, "sum": 6755.892992019653, "min": 6755.892992019653}, "finalize.time": {"count": 1, "max": 113.7399673461914, "sum": 113.7399673461914, "min": 113.7399673461914}, "initialize.time": {"count": 1, "max": 439.47577478501465, "sum": 439.47577478501465, "min": 439.47577478501465}, "check_early_stopping.time": {"count": 11, "max": 1.132965097890625, "sum": 6.217479705810547, "min": 0.19215179443359375}, "setoptime": {"count": 1, "max": 15.5029296875, "sum": 15.5029296875, "min": 15.5029296875}, "update.time": {"count": 10, "max": 700.309786605835, "sum": 6071.307215499878, "min": 522.4170684814453}, "epochs": {"count": 1, "max": 15, "sum": 15.0, "min": 15}}, "EndTime": 1559222305.99424, "Dimensions": {"Host": "algo-1", "Operation": "training", "Algorithm": "Linear Learner"}, "StartTime": 1559222299.314418}
```

2019-05-30 13:18:34 Uploading - Uploading generated training model
2019-05-30 13:18:34 Completed - Training job completed
Billable seconds: 78

20. If we run the cell, we can see where the Linear Learner model artifact is stored.

```
In [35]: print('Here is the location of the trained Linear Learner model: {}://{}{}'.format(output_location, job_name))
Here is the location of the trained Linear Learner model: s3://ml-labs-acg/algorithms_lab/linearlearner_output/ufo-linear-learner-job-20190530131517/output/model.tar.gz
```

21. It is important to know how long the training job took, and you can see the billable seconds were 78 seconds. So, it charged us 78 seconds when it spun up the compute-optimized instance and trained our model.

```
2019-05-30 13:18:34 Uploading - Uploading generated training model
2019-05-30 13:18:34 Completed - Training job completed
Billable seconds: 78
```

Hence, we have created an XGBoost model and a Linear Learner model. Our XGBoost model has around 93% accuracy, and our Linear Learner has around 94% accuracy. Therefore, we have two models that are above 90% accurate. We can use these metrics to show the organization that we can use them in production or make inference calls.

22. We can now clean up our environment. Navigate back to SageMaker and select your notebook instance.

The screenshot shows the 'Notebook instances' section of the Amazon SageMaker console. A single instance is listed:

Name	Instance	Creation time	Status	Actions
ml-labs-notebook-instance	ml.t2.medium	May 16, 2019 15:38 UTC	InService	Open Jupyter Open JupyterLab

23. Use the Actions dropdown and select Stop.

The screenshot shows the same 'Notebook instances' section. The 'Actions' dropdown menu is open over the selected instance row. The 'Stop' option is highlighted with a red arrow.

Actions ▾

- Open Jupyter
- Open JupyterLab
- Stop**
- Start
- Update settings
- Add/Edit tags
- Delete

Lab 10: Understanding CloudFormation Template Anatomy

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time	

Required	1 Hr.
Tags	CF, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

AWS CloudFormation is a tool that makes it simple for developers and organizations to construct a collection of linked AWS and third-party resources and then provision and manage them logically and reasonably.

Developers may use a simple, declarative approach to deploy and change compute, database, and many other resources, abstracting away the complexities of individual resource APIs. AWS CloudFormation is meant to make resource lifecycle management repeatable, predictable, and safe, with features such as automatic rollbacks, automated state management, and resource management across accounts and regions. Multiple ways to generate resources have recently been added, including leveraging AWS CDK for writing in higher-level languages, importing existing resources, and detecting configuration drift. A new registry simplifies the creation of custom types that inherit several of CloudFormation's fundamental features.

Case Study Security Products Manufacturers – Foxis Security Solutions

Background

Foxis Security Systems International manufactures security equipment, such as surveillance systems, alarms, bulletproof glass and vehicles, safes, and associated technology. Foxis is a 2,500-employee business with global offices. Its IT team of 15 is in charge of the day-to-day operation of its datacenters and the improvement of the business through innovative use of technology.

Foxis Security Solutions has two datacenters, one in New York and the other in Los Angeles, housed in a co-location facility. The environment is largely comprised of Web servers, with around 60% running on a VMware cluster, 30% on a newer Hyper-V cluster, and the rest on older, physical servers. Some Linux servers and two major, in-house-developed Line-Of-Business (LOB) applications. The IT department is currently facing a difficult scenario, with outdated technology in both datacenters and a declining budget.

Mike, the CIO, joins a meeting with the whole IT employees to discuss how to improve Foxis' IT environment over the next 12 to 24 months. Selecting a public/hybrid cloud provider is a long-term plan, and CIO believes Amazon Web Services provides the best solution for a gradual transition from Foxis' present systems to a hybrid/mostly public cloud-based infrastructure.

Business Challenge

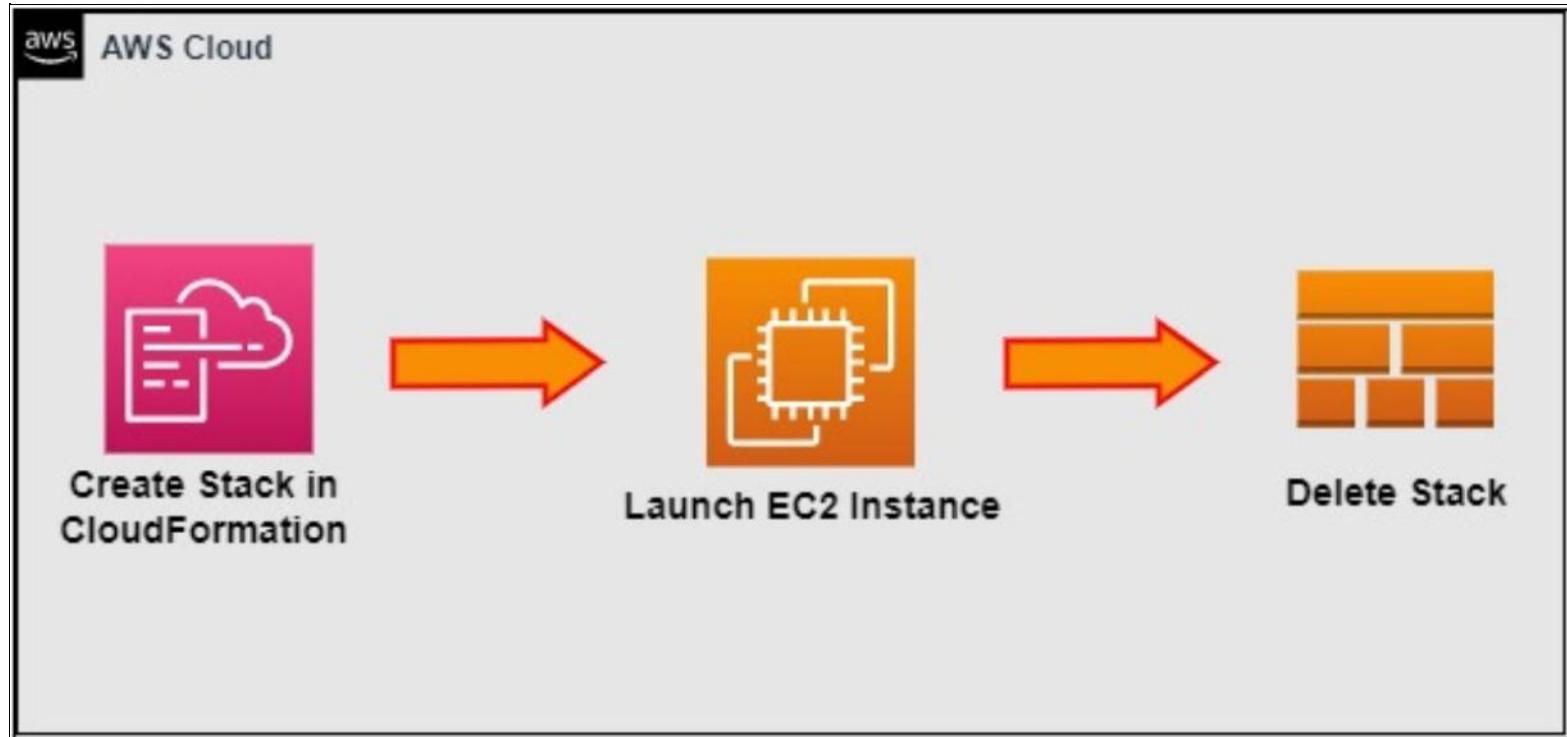
You are a DevOps engineer in Foxis Security Solutions. Your task is to continue innovating with technology and platform improvements to create an excellent customer experience. For this, you need a

tool that will automate all of your manual work, such as launching and configuring EC2 instances or creating S3 storage buckets. How can you automate this work?

Proposed Solution

As the DevOps engineer, you have been asked to use the AWS CloudFormation service to automate all the work. In CloudFormation, you run a template of JSON or YAML to deploy the entire frontend and backend stack into the AWS environment.

Lab Diagram



Implementation Steps

1. Create Stack in CloudFormation.
2. Observe the Stack Events.
3. Delete a CloudFormation stack.

Solution

Step 1: Create Stack in CloudFormation

1. Log in to the **AWS Management Console**.



Services ▾



More ▾

AWS Management Console

AWS services

▼ Recently visited services



IAM



EC2



Billing

► All services

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Click on Services.



Services ▾



More ▾

AWS Management Console

AWS services

▼ Recently visited services



IAM



EC2



Billing

► All services

Feedback

English (US) ▾

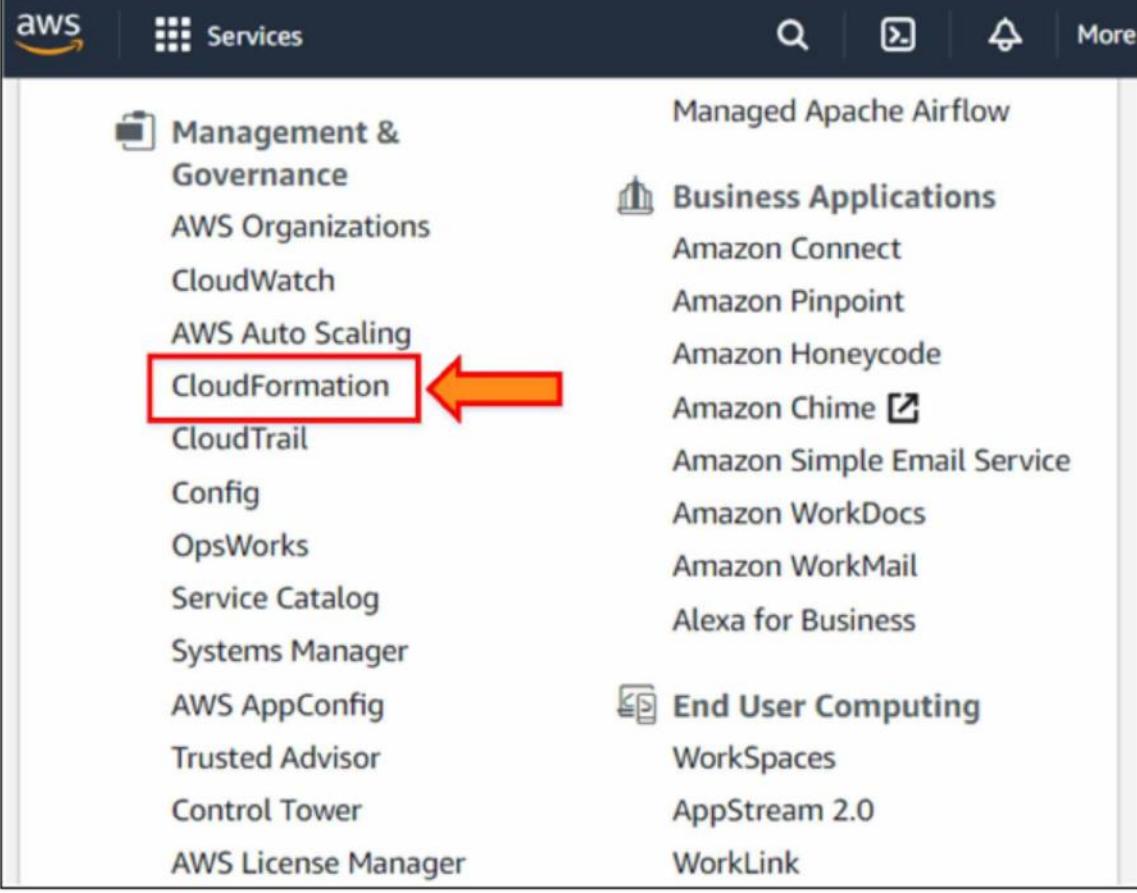
Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Scroll down to Management & Governance. Select Cloud Formation.



4. If you have never used **CloudFormation** before, you will be getting a screen like the following.
5. From the options given on the main **dashboard**, click on **Create stack**.



6. On the **Create stack** page, select **Create template in Designer**.

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready

Use a sample template

Create template in Designer



7. Then, click **Next**.

Create template in Designer

Use the AWS CloudFormation Designer to graphically design your stack on a simple, drag-and-drop interface. The Designer automatically updates and validates the template JSON or YAML.

[Create template in designer](#)

S3 URL: *Will be generated when sample template is created in Designer*

[View in Designer](#)

[Next](#)

8. Click the **Template** tab at the bottom.

The screenshot shows the AWS CloudFormation template editor interface. At the top, there are standard file operations like 'File', 'Edit', 'CloudWatch Metrics', 'Check', and 'Close'. Below that is a header bar with the title 'Resource types' and a sub-header 'File: 'new.template''. A note says 'Building your template, drag resources here...'. On the right side of the header, there are icons for download, help, and refresh.

The main area is titled 'new.template' with a pencil icon. To the right, there's a 'Choose template...' button. The code editor shows the following YAML:

```
1 1
2 "AWSTemplateFormatVersion": "2010-09-09"
3 }
```

Below the code editor, there are two tabs: 'Components' and 'Template'. The 'Template' tab is highlighted with a red border and has a large orange arrow pointing to it from the left.

9. Copy everything in the [Template_Anatomy2.yaml](#) file (found on GitHub).

https://github.com/natonic/CloudFormation-Deep-Dive

natonic / CloudFormation-Deep-Dive Public

Code Issues 2 Pull requests 2 Actions ...

master ...

CloudFormation-Deep-Dive / Labs / TemplateAnatomy / Template_Anatomy2.yaml

natonic Update Template_Anatomy2.yaml History

1 contributor

399 lines (398 sloc) | 8.3 KB

```
1 AWSTemplateFormatVersion: 2010-09-09
2 Description: >-
3   This template creates an EC2 instance based on the r
4   AMI ID. It also will create a Security Group.
5 Parameters:
6   MySubnet:
7     Description: My subnet from my VPC
8     Type: String
9     Default: subnet-YYYYYYYY
10  MySG:
11    Description: My Security Group from my VPC
12    Type: String
13    Default: SG-YYYYYYYY
14  KeyName:
```

Open with Desktop

View raw

Copy raw contents

View blame

10. Then, paste it into the **Template** window.

The screenshot shows the AWS CloudFormation 'new.template' editor. The template content is as follows:

```
1 AWSTemplateFormatVersion: 2010-09-09
2 Description: >-
3   This template creates an EC2 instance based on the region and selection of
4   AMI ID. It also will create a Security Group.
5 Parameters:
6   MySubnet:
7     Description: My subnet from my VPC
8     Type: String
9     Default: subnet-YYYYYYYY
10  MySG:
11    Description: My Security Group from my VPC
12    Type: String
```

The 'Parameters' section (lines 5-12) is highlighted with a red box.

Below the code editor, there are two tabs: 'Components' and 'Template'. The 'Template' tab is selected.

11. In a new browser tab, navigate to **EC2 > Key Pairs**.
12. Click **Create Key Pair**.

The screenshot shows the AWS Key Pairs list page. The table has columns for Name and Type. There are four existing key pairs listed:

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	AWSKeyPair	rsa
<input type="checkbox"/>	IMGZain	rsa
<input type="checkbox"/>	IPS-holprivkeypair	rsa

At the top right of the table, there is a 'Create key pair' button with a red arrow pointing to it.

13. Give it a **key pair name of tempanatomy**, and click **Create**.

Create key pair Info

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

tempanatomy

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Tags (Optional)

No tags associated with the resource.

Add tag

You can add 50 more tags.

Create key pair



14. Click **Security Groups** in the left-hand side given menu.

15. Copy the **Security Group ID** and paste it into a text file; we will need it later.

Security Groups (1/16) Info



Actions ▾

Export security groups to CSV

-

sg-07dc1c80eba777652

default

sg-07dc1c80eba777652 - default

Details

Inbound rules

Outbound rules

Tags

Security group name

default

Security group ID

sg-07dc1c80eba777652



16.
17.

Navigate to **VPC > Subnets**.

Select one of the listed **Subnets**, and copy its **Subnet ID**. Paste it into a text file since we will also need it later.

Subnets (1/6) [Info](#) [C](#) [Actions ▾](#) [Create subnet](#)

Filter subnets

< 1 > | [⚙️](#)

-	Name	Subnet ID	Status
<input checked="" type="checkbox"/>	-	subnet-091994c95a33c1faa	✓

subnet-091994c95a33c1faa

[Details](#) [Flow logs](#) [Route table](#) [Network ACL](#)

[CIDR reservations](#) [Sharing](#) [Tags](#)

Details

Subnet ID
 subnet-091994c95a33c1faa 

18. Back in the **CloudFormation template window**, click the **checkbox** at the top to **validate the template**.



Resource types

File: 'new.template'

new.template 

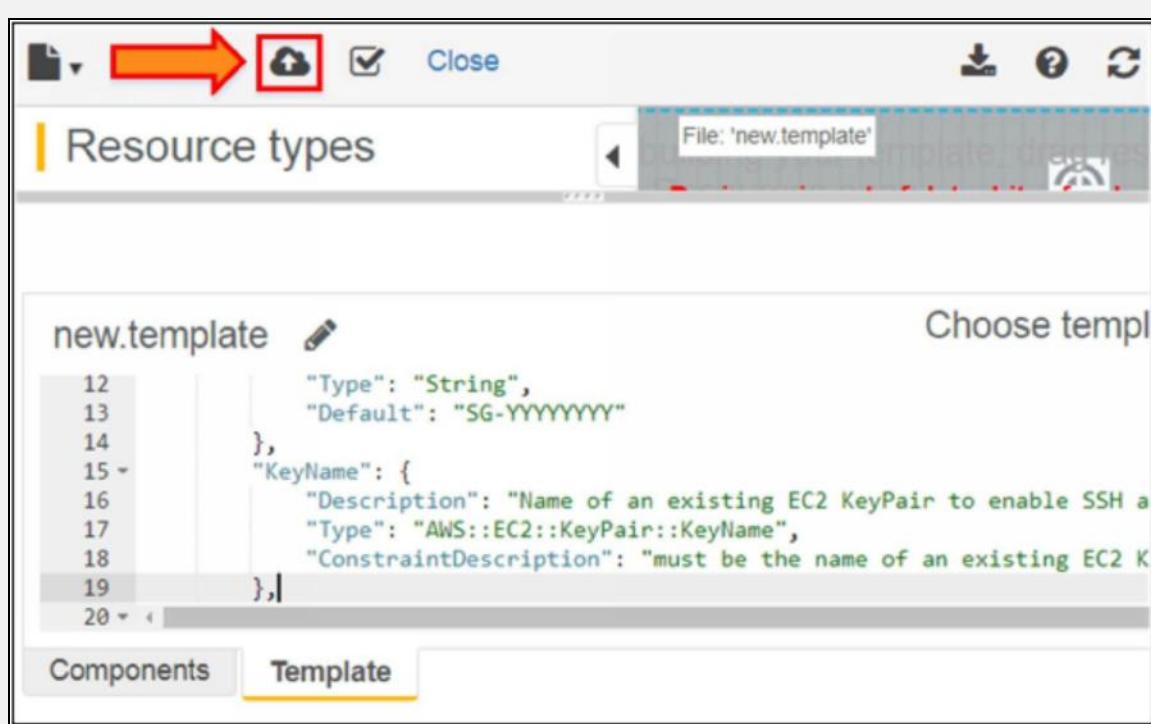
Choose template

```
12     "Type": "String",
13     "Default": "SG-YYYYYYYY"
14   },
15   "KeyName": {
16     "Description": "Name of an existing EC2 KeyPair to enable SSH access",
17     "Type": "AWS::EC2::KeyPair::KeyName",
18     "ConstraintDescription": "must be the name of an existing EC2 KeyPair"
19   }
20
```

Components [Template](#)

19.

Then, click the **cloud icon** with the up arrow to **create the stack**.



20.

It will be redirected to the **Create stack**

page where the **Template is ready** already selected then click **Next**.

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready

Use a sample template

S3 URL: <https://s3-external-1.amazonaws.com/cf-templates-1j7a97b4a35qv-us-east-1/2022034aOY-new.templates2ol225sq2>

View
in
Desi
gner

Next

21.

On the **Specify stack details** page, set a

Stack name as **tempanatomy**.

Specify stack details

Stack name

Stack name

tempanatomy



Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

22. In the **Parameters** option, set the following values:
InstanceType: **t2.micro**
KeyName: **tempanatomy**
MySG: Paste in the security group ID you copied earlier
MySubnet: Paste in the subnet ID you copied earlier
SSHLlocation: **0.0.0.0/0**
23. Click **Next**.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

InstanceType

WebServer EC2 instance type

t2.micro



KeyName

Name of an existing EC2 KeyPair to enable SSH access to the instance

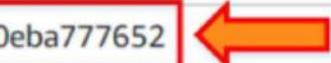
tempanatomy



MySG

My Security Group from my VPC

sg-07dc1c80eba777652



MySubnet

My subnet from my VPC

subnet-091994c95a33c1faa



SSHLocation

The IP address range that can be used to SSH to the EC2 instances

0.0.0.0/0



Cancel

Pre

Next

24. On the stack options page, set the **Key** as **name** and **Value** as **tempanatomy**.
25. Click **Next**.

Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

The screenshot shows a 'Tags' section with two entries: 'name' and 'tempanatomy'. Each entry has a red double-headed arrow icon next to it. Below the entries is a button labeled 'Add tag'.

Cancel Next

26. Review the provided details and click **Create stack**.

Review tempanatomy

Step 1: Specify template

Edit

Template

Template URL

<https://s3-external-1.amazonaws.com/cf-templates-1j7a97b4a35qv-us-east-1/2022034aOY-new.templates2ol225sq2>

Stack description

This template creates an EC2 instance based on the region and selection of an AMI ID. It also will create a Security Group.

[Estimate cost](#) 

Step 2: Specify stack details

Edit

Parameters (5)

 *Search parameters*



Key	▲	Value	▼
InstanceType		t2.micro	
KeyName		tempanatomy	

► Quick-create link

[Cancel](#)

[Previous](#)

[Create chan](#)

[Create stack](#)

27. It will take a few minutes for it to be created fully.

The screenshot shows the AWS CloudFormation console. On the left, there's a sidebar with a search bar for 'Stacks' and a dropdown for 'Active'. A large blue box highlights the stack 'tempanatomy' in the list, which has a status of 'CREATE_IN_PROGRESS'. At the top right, there are buttons for 'Delete', 'Update', 'Stack actions', and 'Create stack'. Below the stack list is a table with columns 'Timestamp', 'Logical ID', and 'Status'. One row in the table corresponds to the stack creation.

Timestamp	Logical ID	Status
2022-02-03 17:33:06 UTC +0500	tempanatomy	CREATE_IN_PROGRESS

Step 2: Observe the Stack Events

- When the **tempanatomy** stack is created, click **refresh**, and you will see an EC2 is also **created**.

The screenshot shows the 'Events' page for the 'tempanatomy' stack. It lists five events. Two events are highlighted with a red box and orange arrows pointing to them. Both events are for the creation of an EC2 instance, with the first one being completed successfully and the second one still in progress. The third event is a note about resource creation initiation.

Timestamp	Logical ID	Status	Status reason
2022-02-03 17:34:09 UTC+0500	tempanatomy	CREATE_COMPLETE	-
2022-02-03 17:34:07 UTC+0500	EC2Instance	CREATE_COMPLETE	-
2022-02-03 17:33:14 UTC+0500	EC2Instance	CREATE_IN_PROGRESS	Resource creation Initiated

- Go to the EC2; you will see EC2 instance is running there.

Launch Instance ▾

Connect

Actions ▾



Filter by tags and attributes or search by key? < < 1 to 1 of 1 > >

Instance ID ▾ Instance Type ▾ Availability Zone ▾ Instance State

i-01b83fe9bec298e81 t2.micro us-east-1d ● running

- EC2 is running because of the **template** we are using in the stack **template**. If we delete the **Stack**, the **EC2 instance** is also **deleted**.

Step 3: Delete a CloudFormation Stack

- Once it is created, click **Delete** at the top.

tempanatomy

Delete



Stack actions ▾

Create stack ▾

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

- In the confirmation dialog, click **Delete stack**.

Delete tempanatomy?



Deleting this stack will delete all stack resources. Resources will be deleted according to their DeletionPolicy. [Learn more](#)

Delete stack

- Monitor the deletion process by watching the **Events** tab.

Stacks (1)			Events (7)		
<input type="text"/> Filter by stack name			<input type="text"/> Search events		
Active ▾			Timestamp ▾	Logical ID	Status
<input checked="" type="checkbox"/> View nested	< 1 >		2022-02-03 17:53:33 UTC+0500	EC2Instance	DELETE_IN_PROGRESS
tempanatomy		2022-02-03 17:53:30 UTC+0500	tempanatomy	DELETE_IN_PROGRESS	
2022-02-03 17:33:06 UTC+0500	00				
DELETE_IN_PROGRESS					

4. When Stack is deleted, you will see an EC2 instance also **deleted**.

Stacks (0)			Events (9)		
<input type="text"/> Filter by stack name			<input type="text"/> Search events		
Active ▾			Timestamp ▾	Logical ID	Status
No stacks			2022-02-03 17:54:04 UTC+0500	tempanatomy	DELETE_COMPLETE
No stacks to display					
View getting started guide					

Launch Instance		Connect	Actions ▾					
<input type="text"/> Filter by tags and attributes or search by keyword				<	<	1 to 1 of 1	>	>
Instance ID	Instance Type	Availability Zone	Instance State					
i-01b83fe9bec298e81	t2.micro	us-east-1d	terminated					

Lab 11: Creating a TensorFlow Image Classifier in AWS SageMaker

Certifications	Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	SageMaker, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

TensorFlow is a popular open-source Machine Learning (ML) framework for creating neural networks and machine learning models. To speed up the training, such models are generally trained on many GPU instances, resulting in longer training times and model sizes of several gigabytes. These models are then used in production to make conclusions once trained. Workloads might be synchronous, asynchronous, or batch-based. Those endpoints must be highly scalable and robust to process zero to millions of requests.

Case Study Enterprise Financial Service— Kasasa

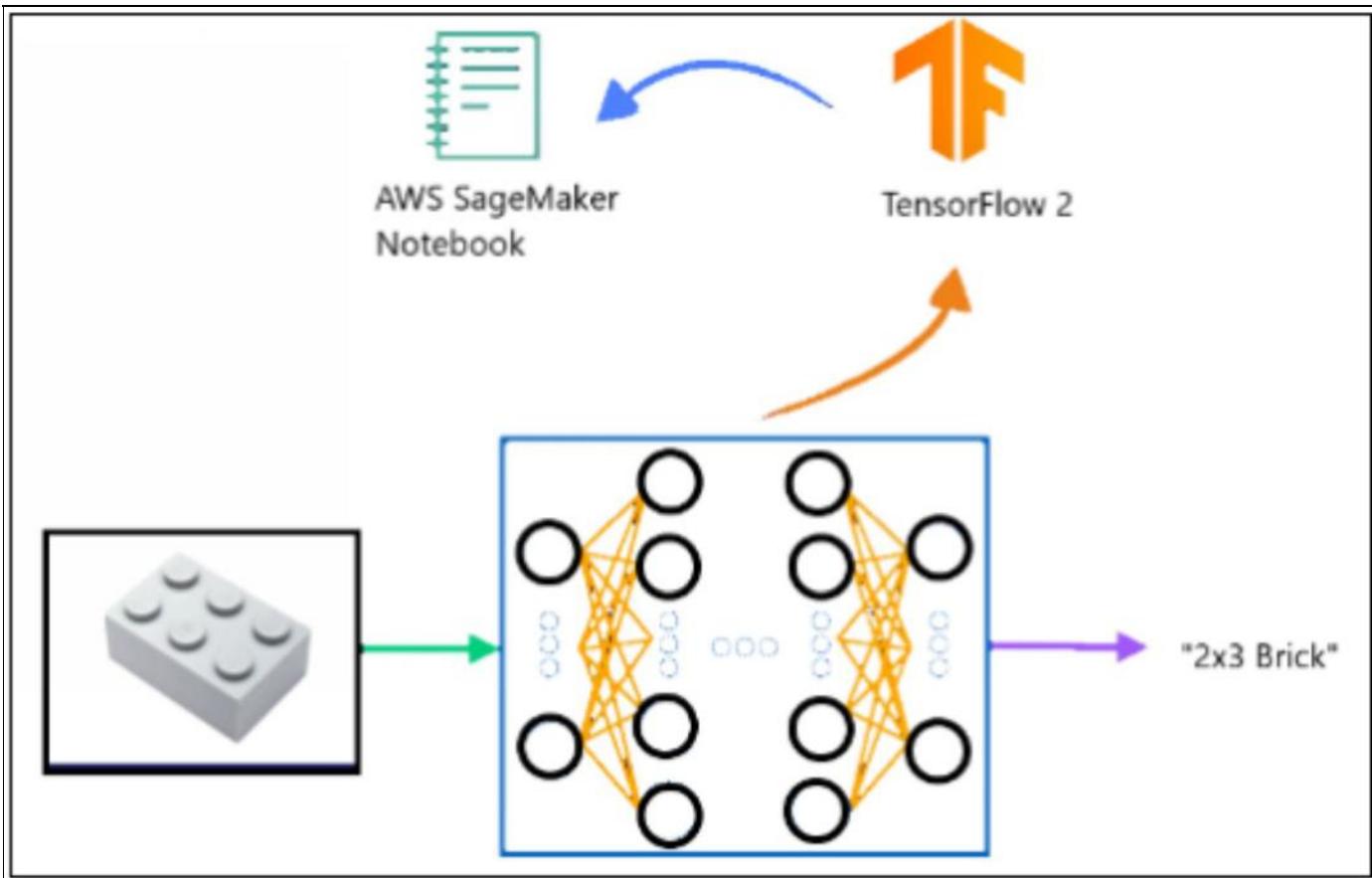
Business Challenge

You work as an AWS architect in Kasasa. They gave the task of building a machine that could reliably sort lego bricks. You have got a lot of LEGO bricks to sort through. It will take even longer to do it by hand, and you will get a lot of eye strain from gazing at the small parts.

Proposed Solution

The solution is you use AWS services to automate your task, and you will need to teach a machine to recognize the bricks and inform you (or, better yet, a robot) which bin each piece belongs in. You already have a lot of LEGO block images, which you will utilize to teach the model you are using.

Lab Diagram



Implementation Steps

1. Navigate to the Jupyter Notebook.
2. Load and Prepare the Data.
3. Train the TensorFlow Model.
4. Test and Analyze the Model.
5. Make a Batch Prediction Using the Testing Data.

Solution

Step 1: Navigate to the Jupyter Notebook

Note: Before starting the lab, you must have a Jupyter notebook. You can create it by following the steps given in the link:

<https://docs.aws.amazon.com/dlami/latest/devguide/setup-jupyter.html>

1. Log in to the **AWS Console**.
2. Go to **Services** and click on **SageMaker**.



Services ▾



More ▾

SageMaker Studio



Cancel

Services (2)

Features (3)

Documentation (46,756)

Knowledge Article

Amazon SageMaker

Build, Train, and Deploy Machine Learning Models



AWS Glue DataBrew

Visual data preparation tool to clean and normalize data for analytics and machine learn...

[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Click on In-Service.



Recent activity

Recent activity within the

Last 7 days

Ground Truth

Notebook

Labeling jobs

Notebook instances

No recent activity

1 In Service



1 Created

Training

Inference

Training jobs

Models

No recent activity.

No recent activity.

[Feedback](#)

English (US) ▼

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4. Click on the **Notebook instance**.



Not
ebo
ok
inst
anc
e



Actions ▾

Create notebook instance



Search notebook instances



1



Name

Instance

Creation
time

Stat

Notebook-
3yAsOzpWnFJw

medium

Oct 16,
2021 09:48
UTC

In

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Click on Open Jupyter.



Not
ebo
ok
inst
ance
s



Actions ▾

Create notebook instance



Search notebook instances



1



Instance	Creation time	Status	Actions
ml.t3.medium	Oct 16, 2021 09:48 UTC	Idle	Open Jupyter Open JupyterLab

[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Click on **CreateATensorFlowImageClassifier.ipynb**.

Files

Running

Clusters

SageMaker Examples

Conda

Select items to perform actions on them.

Upload

New ▾

 0

/

Name ↓

Last Modified

File size

 lost+found

an hour ago

 CreateATensorFlowImageClassifier.ipynb

an hour ago 20.5 kB

 acg_logo.png

an hour ago 13.7 kB

 lego-simple-test-images.npy

an hour ago 2.76 MB

 lego-simple-test-labels.npy

an hour ago 1.33 kB

 lego-simple-train-images.npy

an hour ago 8.31 MB

 lego-simple-train-labels.npy

an hour ago 3.74 kB

 lego.jpg

an hour ago 81.6 kB

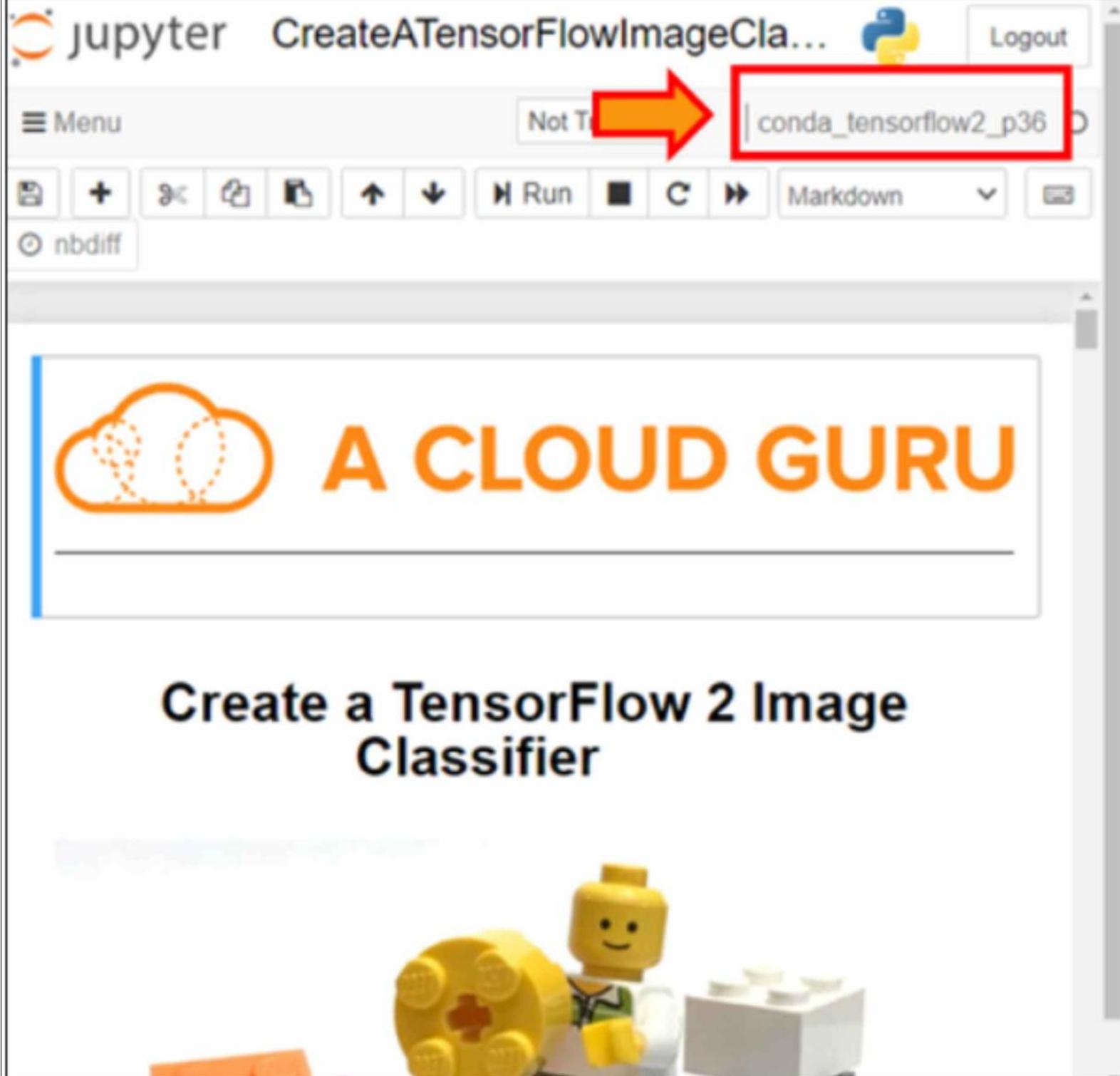
 sample-after.png

an hour ago 2.71 kB

 sample-before.png

an hour ago 35.6 kB

7. Make sure our kernel will support **TensorFlow 2**.



8. If the kernel is not set to TensorFlow 2, go to **Menu** and click on **Kernel**.

☰ Menu



Not Trusted

| conda_tensorflow2_p36 O

File

Edit

View

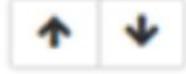
Insert

Cell

Kernel

Widgets

Help



Markdown



9. Click on **Change Kernel**.

Cell

Kernel

Interrupt

Restart

Restart & Clear Output

Restart & Run All

Reconnect

Shutdown

Change kernel

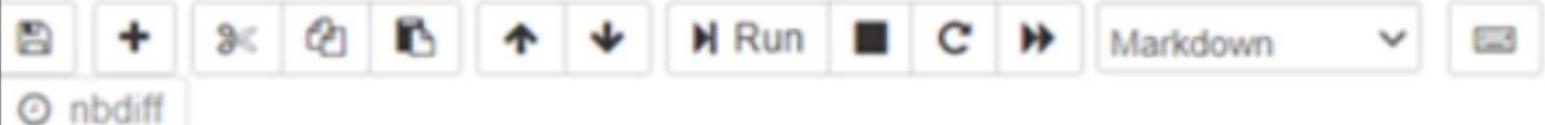


Conda Packages

Visit anaconda.org

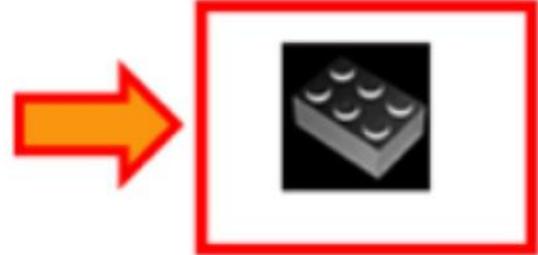
Widgets

Help



10. We have a collection of photographs of different LEGO bricks taken from various perspectives. There are 600 pictures in all, all tagged with the brick type.

Each image has been enhanced. Increasing the contrast, sharpening, eliminating the color, inverting the color, and shrinking the image were all part of the process. We put them all into data arrays to make importing the photos into the notebook easier. The folders lego-simple-train-images.npy and lego-simple-test-images.npy include these pictures.



Sample before processing Sample after processing

11. Our first step is to import all the necessary libraries. We need to ensure we are using the right version of tensorflow; hence, we will print the version as well.

```
In [*]: import tensorflow as tf  
from tensorflow import keras  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
print(tf.__version__)
```

```
In [1]: import tensorflow as tf  
from tensorflow import keras  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
print(tf.__version__)
```

2.1.3

Step 2: Load and Prepare the Data

1. We have a dataset built from a collection of LEGO block pictures. A total of four data arrays have been stored in files as NumPy arrays. The data is first loaded into arrays in the notebook.

- `lego-simple-train-images.npy` - Training images, around 80% of the data collected.
- `lego-simple-train-labels.npy` - A list of integer labels identifying the classes of the training images.
- `lego-simple-test-images.npy` - Testing images, around 20% of the data collected.
- `lego-simple-test-labels.npy` - A list of integer labels identifying the classes of the testing images.

```
In [ ]: train_images = np.load('lego-simple-train-images.npy')
train_labels = np.load('lego-simple-train-labels.npy')
test_images = np.load('lego-simple-test-images.npy')
test_labels = np.load('lego-simple-test-labels.npy')
```

2. If we look back in our jupyter notebook server, we can see they are stored already.

	Name	Last Modified	File size
□ 0	/		
□	lost+found	an hour ago	
□	CreateATensorFlowImageClassifier.ipynb	Running a minute ago	20.6 kB
□	acg_logo.png	an hour ago	13.7 kB
□	lego-simple-test-images.npy	an hour ago	2.76 MB
□	lego-simple-test-labels.npy	an hour ago	1.33 kB
□	lego-simple-train-images.npy	an hour ago	8.31 MB
□	lego-simple-train-labels.npy	an hour ago	3.74 kB
□	lego.jpg	an hour ago	81.6 kB
□	sample-after.png	an hour ago	2.71 kB
□	sample-before.png	an hour ago	35.6 kB

Integer values 1 through 10 are represented by the labels we loaded. Integer values are useful for the model to understand and forecast, but they are difficult to comprehend while working with data.

3. Next, we will give the labels we are trying to anticipate human-readable names.

```
In [3]: # For humans:  
class_names = ['2x3 Brick', '2x2 Brick', '1x3 Brick', '2x1 Brick',  
               '2x2 Macaroni', '2x2 Curved End', 'Cog 16 Tooth',  
  
# Or the real LEGO codes:  
# class_names = ['3002', '3003', '3622', '3004', '3005', '3063',  
    ]
```

4. Take a look at one of the photos loaded with data. This tells us that our image is 48 by 48 pixels. We have normalized the values in this array from 0-1 by dividing by 255; since it is a greyscale image normally, it would be between zero and 255 as brightness for an eight-bit image, but with a floating-point value, it is easier for the machine learning model to understand.

```
In [4]: print(train_images[0].shape)
train_images[0]
(48, 48)

Out[4]: array([[0.01568627, 0.01568627, 0.01568627, ..., 0.01568627, 0.01568627,
   0.01568627],
   [0.01568627, 0.01568627, 0.01568627, ..., 0.01568627, 0.01568627,
   0.01568627],
   [0.01568627],
   [0.01568627, 0.01568627, 0.01568627, ..., 0.01568627, 0.01568627,
   0.01568627],
   [0.01568627],
   ...,
   [0.03921569, 0.03921569, 0.03921569, ..., 0.01960784, 0.01960784,
   0.01960784],
   [0.03921569, 0.03921569, 0.03921569, ..., 0.01960784, 0.01960784,
   0.01960784],
   [0.01960784],
   [0.03529412, 0.03529412, 0.03529412, ..., 0.01960784, 0.01960784,
   0.01960784],
   [0.01960784]])
```

5. We will also look at the testing data to see whether it is similar.

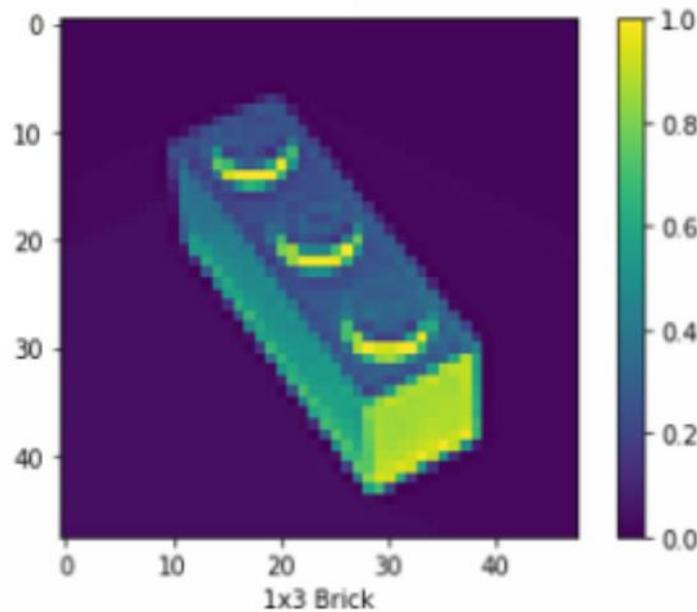
```
In [5]: print(test_images[0].shape)
test_images[0]
```

(48, 48)

```
Out[5]: array([[0.00784314, 0.00784314, 0.00784314, ... , 0.00784314, 0.00
784314,
               0.00784314],
              [0.00784314, 0.00784314, 0.00784314, ... , 0.00784314, 0.00
784314,
               0.00784314],
              [0.00784314, 0.00784314, 0.00784314, ... , 0.00784314, 0.00
784314,
               0.00784314],
              ... ,
              [0.01568627, 0.01568627, 0.01568627, ... , 0.01568627, 0.01
568627,
               0.01568627],
              [0.01960784, 0.01960784, 0.01960784, ... , 0.01568627, 0.01
568627,
               0.01568627],
              [0.01960784, 0.01960784, 0.01960784, ... , 0.01568627, 0.01
568627,
               0.01568627]])
```

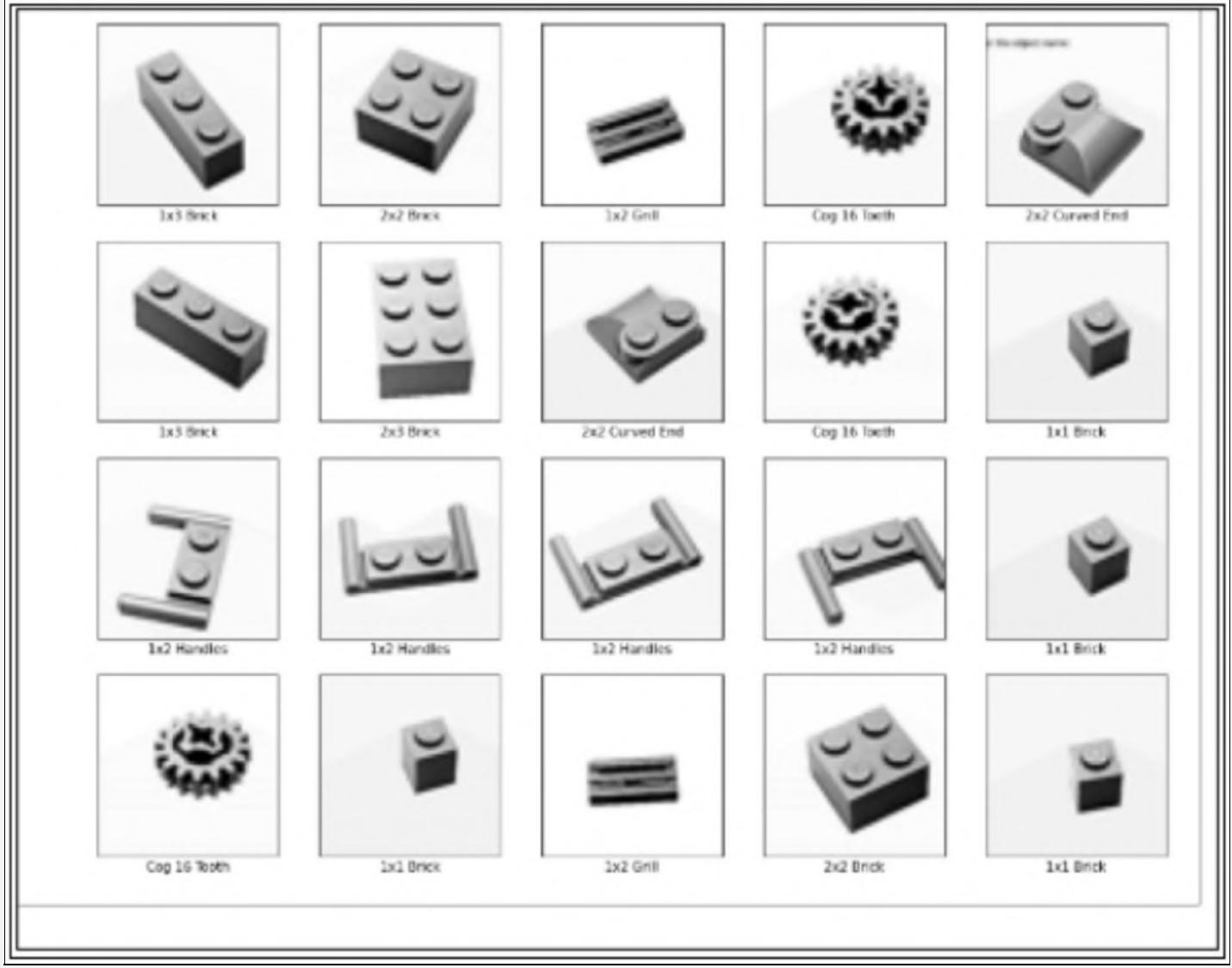
6. We will then plot these values to obtain a clearer view.

```
In [6]: plt.figure()
plt.imshow(train_images[0])
plt.colorbar()
plt.xlabel(class_names[train_labels[0]])
plt.show()
```



7. The top and bottom few rows are dark, as can be seen. Those floating-point values are close to zero, as we observed previously. We will take a look at some additional data in a more appealing manner. This will change the colors of the output. As a result, the background appears light, but the bricks are dark, making them easier to notice.

```
In [ ]: plt.figure(figsize=(15,15))
for i in range(20):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.imshow(train_images[i], cmap=plt.cm.binary)
    plt.xlabel(class_names[train_labels[i]])
plt.show()
```



Step 3: Train the TensorFlow Model

1. Now that the data has been imported, we must construct a model. We are constructing an artificial neural network. For this, we will utilize Keras and create a sequential model, which means each layer will pass to the next.

We are going to utilize the first layer as a flattened layer. Our input shape is truly 48 by 48, as seen above. Still, the model works better if it is a one-dimensional array rather than a two-dimensional array, which will be 2,304 values, which is actually 48 by 48. We will link the flattened layer's output to our next layer, which will be our concealed layer. Because this is a dense layer, every one of the input values will be linked to each of the 128 neurons triggered by the ReLU function. The network's final layer will be a dense layer of ten nodes, one for each of the classes we are trying to predict, which ten distinct sorts of Lego bricks are. We utilize the softmax activation function for this layer, which generates a set of probabilities for each of the ten nodes. The neuron will determine the class that the model predicts with the highest probability.

```
In [8]: model = keras.Sequential([
    keras.layers.Flatten(input_shape=(48, 48)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

2. Now, we will compile the model.

```
In [9]: model.compile(optimizer='adam',
                      loss='sparse_categorical_crossentropy',
                      metrics=['accuracy'])
```

3. Finally, it is time to put the model through its tests. Keep track of the epochs, and the accuracy as the process unfolds.

```
In [11]: history = model.fit(train_images, train_labels, epochs=4)

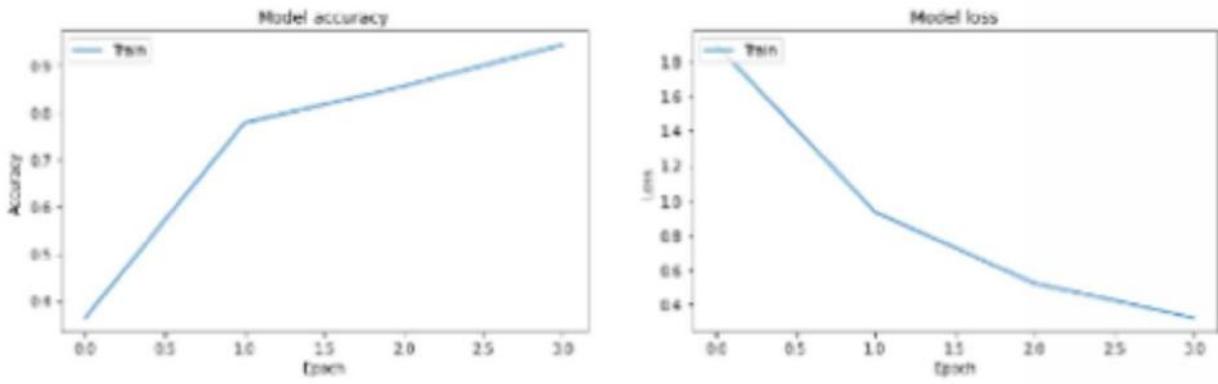
[2021-10-16 11:09:13.815 ip-172-16-35-238:27397 INFO utils.py:27]
RULE_JOB_STOP_SIGNAL_FILENAME: None
[2021-10-16 11:09:13.922 ip-172-16-35-238:27397 INFO profiler_config_parser.py:111] Unable to find config at /opt/ml/input/config/profilerconfig.json. Profiler is disabled.
Train on 451 samples
Epoch 1/4
451/451 [=====] - 2s 5ms/sample - loss: 1.8926 - accuracy: 0.3636
Epoch 2/4
451/451 [=====] - 0s 97us/sample - loss: 0.9308 - accuracy: 0.7783
Epoch 3/4
451/451 [=====] - 0s 100us/sample - loss: 0.5223 - accuracy: 0.8559
Epoch 4/4
451/451 [=====] - 0s 93us/sample - loss: 0.3239 - accuracy: 0.9446
```

4. As you can see from the output above, the model's accuracy score improves with each epoch. When we trained the model, we saved the training history in a variable from plotting the training progress. Over the epochs, we may plot accuracy and loss.

```
In [12]: plt.figure(figsize=(15, 4))

plt.subplot(1, 2, 1)
# Plot training accuracy values
plt.plot(history.history['accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')

plt.subplot(1, 2, 2)
# Plot training loss values
plt.plot(history.history['loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



Step 4: Test and Analyze the Model

1. Now, we will compare the testing results to the accuracy. The model is probably overfitted if the accuracy attained during training is significantly higher than what we compute for the testing data. This implies it is good for classifying training data but not for classifying testing data.

```
In [13]: test_loss, test_acc = model.evaluate(test_images, test_labels)
print('Test accuracy:', test_acc)

150/150 [=====] - 0s 498us/sample - los
s: 0.4387 - accuracy: 0.86667
Test accuracy: 0.8666667
```

2. We will now apply our model to our testing data to generate some predictions.
3. We will use one of the images from the test dataset.

```
In [ ]: test_image_number = 8  
  
img = test_images[test_image_number]  
plt.figure()  
plt.xticks([])  
plt.yticks([])  
plt.imshow(img, cmap=plt.cm.binary)  
plt.show()
```



4. Before we can examine what the model thinks, we will quickly modify this image. We build a collection of one image since the forecast function wants a collection of photos.

```
In [15]: img = (np.expand_dims(img,0))  
img.shape  
  
Out[15]: (1, 48, 48)
```

5. The shape is now a collection of one 48x48 data item. The picture is then passed to the predict method. The end result is a list of probability that the image belongs to a certain class.

```
In [16]: predictions_single = model.predict(img)
predictions_single
```

```
Out[16]: array([[0.00144515, 0.01388346, 0.00972838, 0.5691016 , 0.0557792
                 3,
                 0.23145372, 0.09599604, 0.01162628, 0.00249234, 0.0084938
                 5]],
                dtype=float32)
```

6. The argmax function in NumPy can be used to discover the index of the biggest value in a list. We may utilize a second NumPy function to get the real maximum value.

```
In [17]: prediction_result = np.argmax(predictions_single[0])
probability = np.max(predictions_single[0])
print(f"Predicted class {prediction_result}, which is {class_names[prediction_result]}")
print(f"The model thinks there is a {probability*100:.2f}% probability of this brick being correct.")
```

Predicted class 3, which is 2x1 Brick.
The model thinks there is a 56.91% probability of this brick being correct.

7. Next, we will search for the test image's class name (i.e., the real class name, not the predicted one). We are hoping for the same!

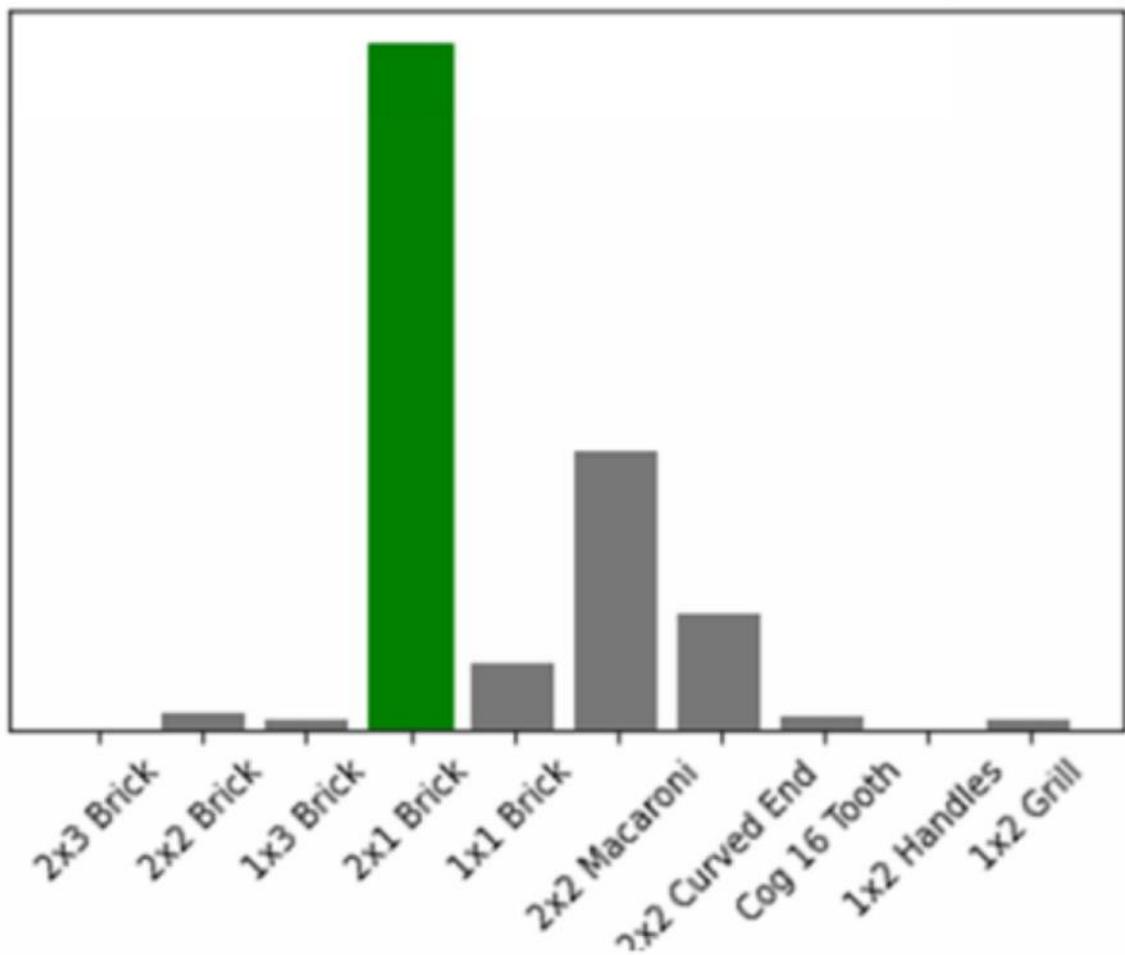
```
In [18]: class_names[test_labels[test_image_number]]
Out[18]: '2x1 Brick'
```

8. These functions aid in the presentation of prediction data.

```
In [19]: # Function to display the image:  
def plot_image(predictions_array, true_label, img):  
    plt.xticks([])  
    plt.yticks([])  
    plt.imshow(img, cmap=plt.cm.binary)  
    predicted_label = np.argmax(predictions_array)  
    predicted_probability = 100 * np.max(predictions_array)  
    color = 'green' if predicted_label == true_label else 'red'  
    # Print a label with 'predicted class', 'probability %', 'actual class'  
    plt.xlabel(f'{class_names[predicted_label]} {predicted_probability} %',  
              color=color)  
  
# Function to display the prediction results in a graph:  
def plot_value_array(predictions_array, true_label):  
    plt.xticks(range(10))  
    plt.yticks([])  
    plot = plt.bar(range(10), predictions_array, color="#777777")  
    predicted_label = np.argmax(predictions_array)  
    plot[predicted_label].set_color('red')  
    plot[true_label].set_color('green')
```

9. We will then use the helper function to create a bar chart. This tells us how accurate the model was in classifying this image. If the projected label differs from the actual label, it will be highlighted in red. Green will be the real label.

```
In [ ]: plot_value_array(predictions_single[0], test_labels[test_image_number])  
plt.xticks(range(10), class_names, rotation=45)  
plt.show()
```



Step 5: Make a Batch Prediction Using the Testing Data

1. We will now obtain prediction values for our test photos.

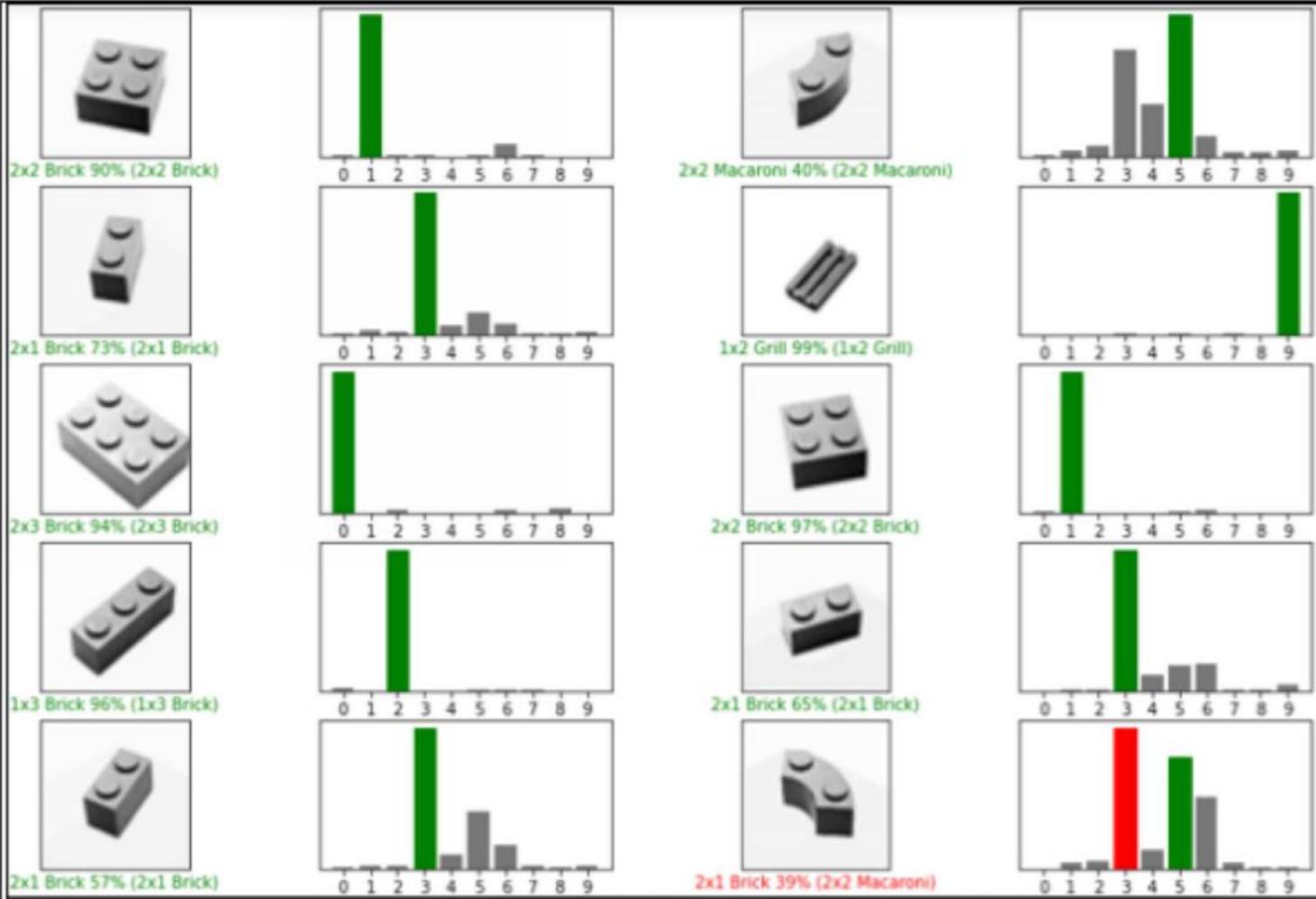
```
In [21]: predictions = model.predict(test_images)
```

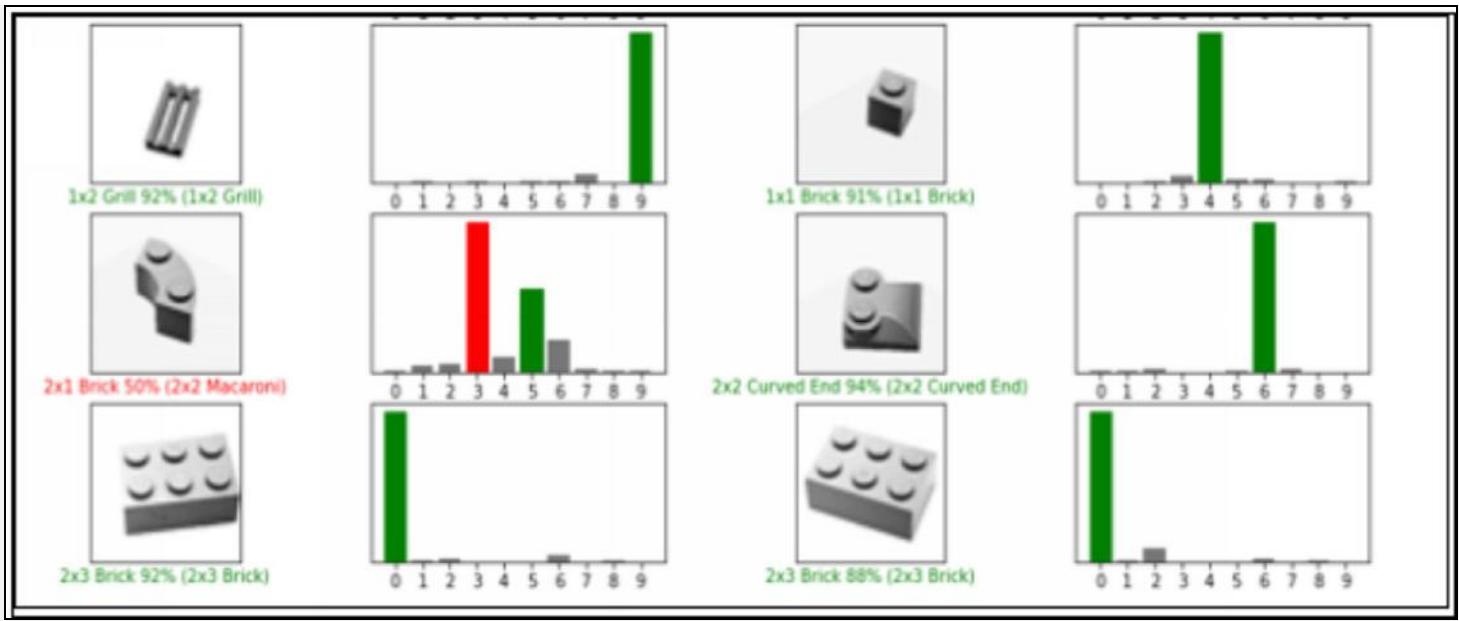
```
In [22]: predictions[0]
```

```
Out[22]: array([5.2462332e-03, 9.0270740e-01, 1.1476018e-03, 2.3488151e-03,
 7.4006454e-07, 4.8946785e-03, 8.1795849e-02, 1.6155281e-03,
 4.2367505e-05, 2.0071988e-04], dtype=float32)
```

2. Our helper functions will now be used to summarize the first 16 photos in our test data.

```
In [ ]: num_rows = 8
num_cols = 2
num_images = num_rows*num_cols
plt.figure(figsize=(15, 16))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(predictions[i], test_labels[i], test_images[i])
    plt.subplot(num_rows, 2*num_cols, 2*i+2)
    plot_value_array(predictions[i], test_labels[i])
plt.show()
```





Given that our model is about 90% accurate, one or two of them may have been classified incorrectly. That is still not bad for only training for four epochs on such little data!

Lab 12: Creating an MXNet Image Classifier in AWS SageMaker

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	SageMaker, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

Apache MXNet is an open-source machine learning framework that focuses on deep neural network learning. Apache MXNet is a fast and scalable machine learning training and inference framework with a simple and clear API.

The Gluon interface included in MXNet allows developers of various skill levels to get started with deep learning on the cloud, in edge devices, and mobile apps. You can construct linear regression, convolutional networks, and recurrent LSTMs for object identification, speech recognition, recommendation, and customization with just a few lines of Gluon code.

Case Study Enterprise Financial Service—Kasasa

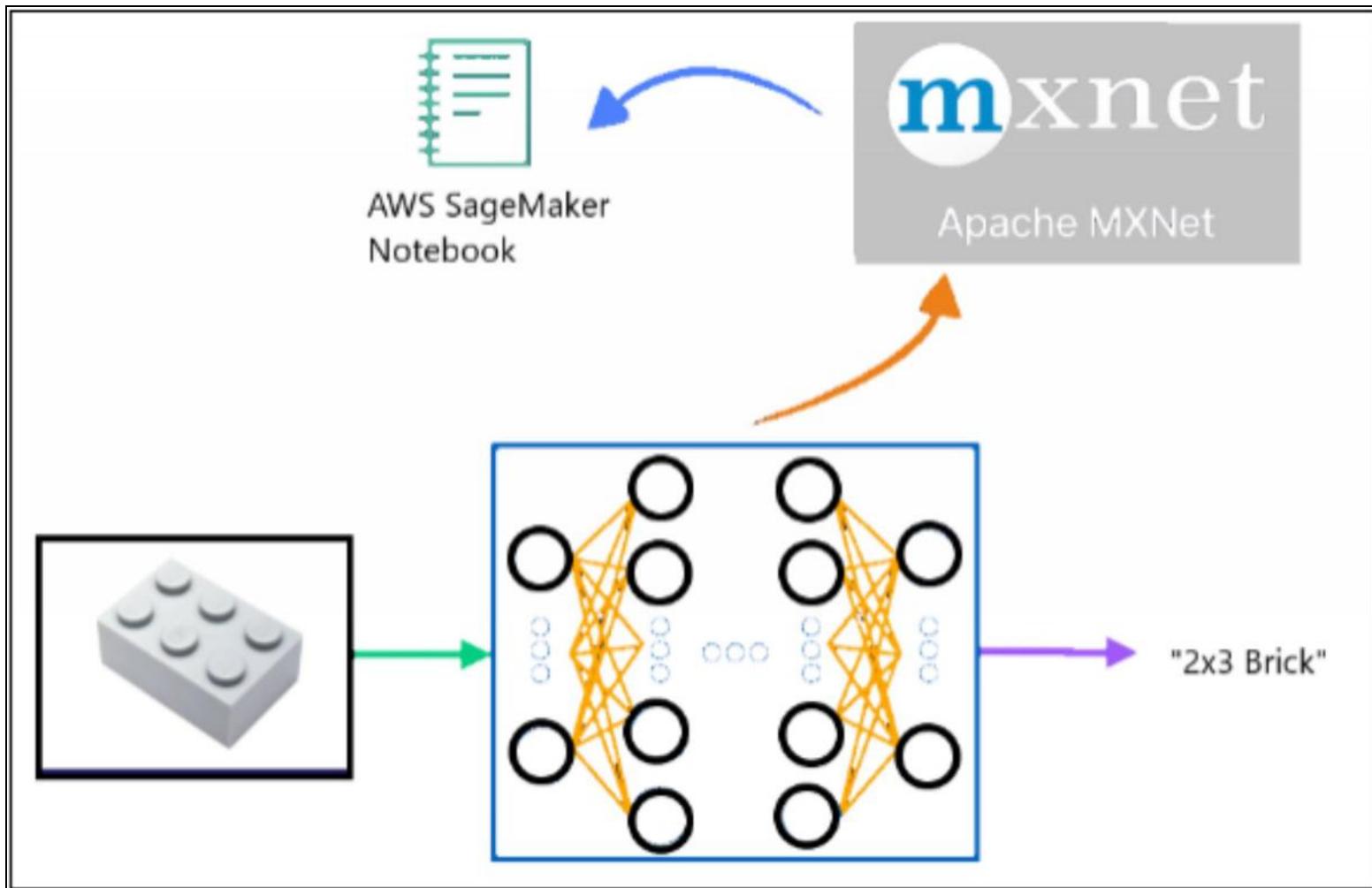
Business Challenge

You work as an AWS architect in Kasasa. They gave you the task of building a machine that can reliably sort lego bricks. You have a lot of LEGO bricks to sort. Doing it by hand will take forever and cause you so much eye strain from looking at the tiny pieces. Hence, you need to train a machine to recognize the bricks and tell you (or ideally, a robot) which bin the piece should be placed in. You already have a lot of pictures of LEGO bricks, which you will use to train the model.

Proposed Solution

The solution is you use AWS services to automate your task, and you will need to teach a machine to recognize the bricks and inform you (or, better yet, a robot) which bin each piece belongs in. You already have a lot of LEGO block images, which you will utilize to teach the model you are using.

Lab Diagram



Implementation Steps

1. Navigate to Jupyter Notebook.
2. Load and Prepare the Data.
3. Train the MXNet Model.
4. Evaluate the Model.
5. Make a Batch Prediction.

Solution

Step 1: Navigate to Jupyter Notebook

Note: Before starting the lab, you must have a Jupyter notebook. You can create it by following the steps given in the link:

<https://docs.aws.amazon.com/dlami/latest/devguide/setup-jupyter.html>

1. Log in to the **AWS Console**.
2. Go to **Services** and click on **SageMaker**.

The screenshot shows the AWS Services search interface. A search bar at the top contains the text "SageMaker Studio". Below the search bar, there are four tabs: "Services (2)", "Features (3)", "Documentation (46,756)", and "Knowledge Articles". The "Services (2)" tab is selected. In the main content area, there are two service cards. The first card, "Amazon SageMaker", is highlighted with a red box and has a large orange arrow pointing to its title. The second card is for "AWS Glue DataBrew". At the bottom of the page, there are links for "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preferences", along with a copyright notice: "© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved."

3. Click on **In-Service**.



Recent activity

Recent activity within the

Last 7 days



Ground Truth

Notebook

Labeling jobs

Notebook instances

No recent activity.

1 In Service

1 Created

Training

Inference

Training jobs

Models

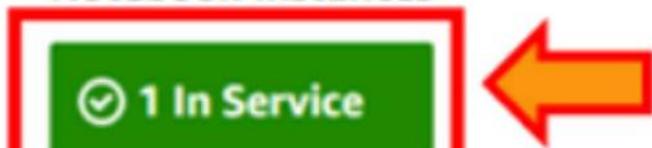
No recent activity.

No recent activity.

[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4. Click on the Notebook instance.





Services ▾



More ▾

ok
inst
ance
s

Actions ▾

Create notebook instance



Search notebook instances

Status : InService X



1



Name	Instance	Creation time	Status
Notebook-BBxveuLrS78O	medium	Oct 16, 2021 09:19 UTC	InS

[Feedback](#)

English (US) ▾

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Click on Open Jupyter.

ok
inst
ance
s

Actions ▾

Create notebook instance



Search notebook instances

Status : InService



1



Instance	Creation time	Status	Actions
ml.t3.medium	Oct 16, 2021 09:19 UTC		<div style="border: 2px solid red; padding: 5px;">Open Jupyter Open JupyterLab</div>

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Click on **CreateAnMXNetImageClassifier.ipynb**.

Files

Running

Clusters

SageMaker Examples

Conda

Select items to perform actions on them.

Upload

New ▾

 0

▼



/

Name ↓

Last Modified

File size

 lost+found

2 hours ago

 CreateAnMXNetImageClassifier.ipynb

2 hours ago 20.5 kB

 acg_logo.png

2 hours ago 13.7 kB

 lego-simple-mx-test

2 hours ago 346 kB

 lego-simple-mx-train

2 hours ago 1.04 MB

 lego.jpg

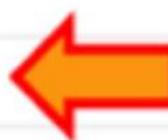
2 hours ago 81.6 kB

 sample-after.png

2 hours ago 2.71 kB

 sample-before.png

2 hours ago 35.6 kB



7. Make sure the kernel supports **MXNet**.

☰ Menu

Not

conda_mxnet_p36 C

+ ↺ ↻ ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋

nbdiff

Run

C

»

Run

Markdown

▼

Cell

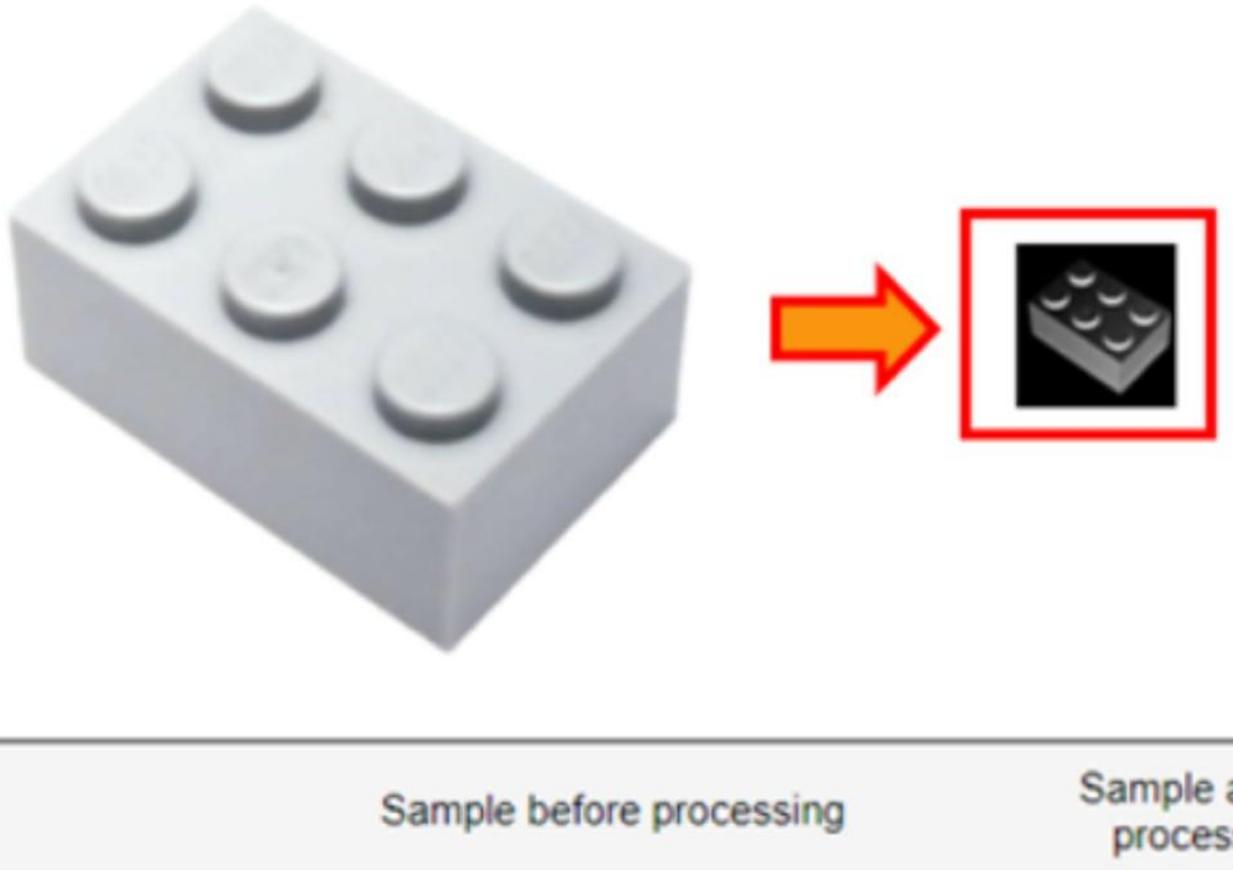


MXNet Imgage Classification



8. We have a collection of photographs of different LEGO bricks taken from various perspectives. There are 600 pictures in all, all tagged with the brick type.

Each image has been enhanced. Increasing the contrast, sharpening, eliminating the color, inverting the color, and shrinking the image were all part of the process. We put them all into data arrays to make importing the photos into the notebook easier. The folders `lego-simple-train-images.npy` and `lego-simple-test-images.npy` include these pictures.



10. Our first step is to import all the necessary libraries. We will use a fixed seed that is generated at random to ensure that our models are reproducible.

```
In [1]: import mxnet as mx
from mxnet.gluon.data.vision import transforms

from mxnet import nd, gluon, autograd
from mxnet.gluon import nn

import pickle

import numpy as np
import matplotlib.pyplot as plt

# Select a fixed random seed for reproducibility
mx.random.seed(42)
```

11. MXNet's processor type will be determined. We let MXNet figure out that this Jupyter Notebook server does not have access to GPUs.

```
In [2]: ctx = mx.gpu(0) if mx.context.num_gpus() > 0 else mx.cpu(0)
```

Step 2: Load and Prepare the Data

1. We have a dataset built from a collection of LEGO block pictures. We have two sets of data saved to files as NDArrays in total.

- lego-simple-mx-train - Combines training pictures and labels, accounting for around 80% of the data gathered
- lego-simple-mx-test - Images and labels for testing or validation, accounting for around 20% of the data gathered

The data is first loaded into runtime arrays. Because these object files were created with Pickle, we will also use Pickle to load them.

```
In [3]: # Open the files:  
train_fh = open('lego-simple-mx-train', 'rb')  
test_fh = open('lego-simple-mx-test', 'rb')  
  
# Use pickle to load files into runtime objects:  
train_data = pickle.load(train_fh)  
test_data = pickle.load(test_fh)
```

2. We loaded integer values for the label data (1,2,3). We want to give the data classes we are working with human names.

```
In [5]: # For humans:  
class_names = ['2x3 Brick', '2x2 Brick', '1x3 Brick', '2x1  
               '2x2 Macaroni', '2x2 Curved End', 'Cog 16 T'  
  
# Or the real LEGO codes:  
# class_names = ['3002', '3003', '3622', '3004', '3005', '
```

3. The data has been put into NDArrays. We now convert the data to MXNet tensors.

Tensors are similar to arrays, but they have more capabilities.

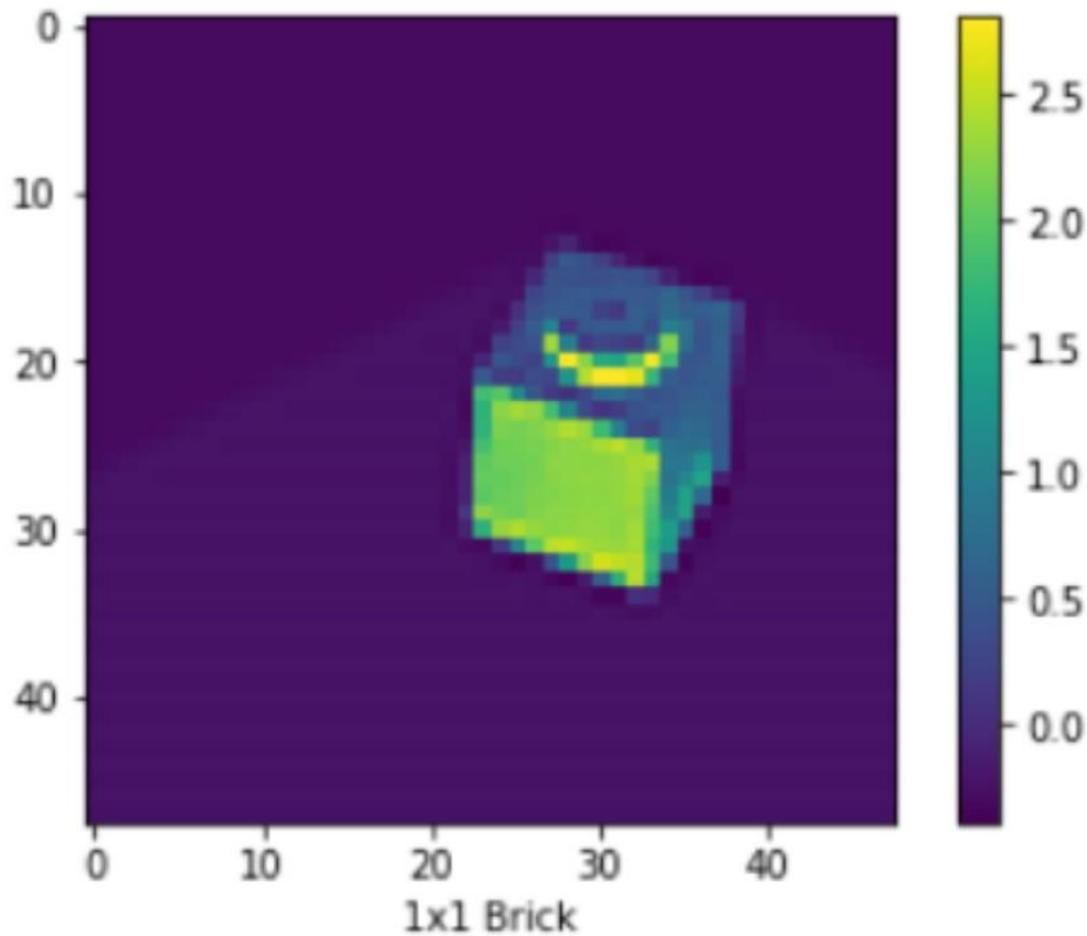
```
In [6]: transformer = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize(0.13, 0.31)])

train_data = train_data.transform_first(transformer)
test_data = test_data.transform_first(transformer)
```

4. We will look at one of the pictures with the data loaded on it.

```
In [ ]: train_image_no = 0

images_data, label_data = train_data[train_image_no]
plt.figure()
plt.imshow(images_data.reshape((48,48)).asnumpy())
plt.colorbar()
plt.xlabel(class_names[label_data])
plt.show()
```



5. We will look closely at some more data and improve the formatting.

```
In [ ]: plt.figure(figsize=(15,15))
for i in range(20):
    images_data, label_data = train_data[i]
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.imshow(images_data.reshape((48,48)).asnumpy(), cmap='gray')
    plt.xlabel(class_names[label_data])
plt.show()
```



Step 3: Train the MXNet Model

1. Now that the data has been imported, we can begin training. We must first construct a model. The model object net is what we call MXNet. We are constructing an artificial neural network. It is made up of four layers:

- The input layer for our picture data, with enough nodes
- A 128-node hidden layer with ReLU activation

- A 64-node hidden layer with ReLU activation
- There are ten nodes in the output layer, one for each of the classes we wish to identify

Each layer is densely connected, which means that every neuron in one layer is linked to every neuron in the next.

```
In [9]: net = nn.HybridSequential(prefix='MLP_')
with net.name_scope():
    net.add(
        nn.Flatten(),
        nn.Dense(128, activation='relu'),
        nn.Dense(64, activation='relu'),
        nn.Dense(10, activation=None)
    )
```

2. We may utilize data loaders provided by MXNet Gluon to make data loading easier when training our model. We will get them set up.

```
In [10]: batch_size = 34
train_loader = mx.gluon.data.DataLoader(train_data, shuffle=True)
```

3. We will now initialize the model.

```
In [11]: net.initialize(mx.init.Xavier(), ctx=ctx)
```

4. A trainer object is provided by Gluon to keep track of the training state. It is something we make here and use in the training process.

```
In [12]: trainer = gluon.Trainer(
    params=net.collect_params(),
    optimizer='sgd',
    optimizer_params={'learning_rate': 0.04},
```

5. First, we choose a measure and a loss function to utilize when training. We just utilize the softmax loss function provided by Gluon.

```
In [13]: metric = mx.metric.Accuracy()
loss_function = gluon.loss.SoftmaxCrossEntropyLoss()
```

6. We are now in the process of training. We could put the following code in a fit function, but this inline code is sufficient.

```
In [ ]: num_epochs = 10
history = []

for epoch in range(num_epochs):
    for inputs, labels in train_loader:
        # Possibly copy inputs and Labels to the GPU
        inputs = inputs.as_in_context(ctx)
        labels = labels.as_in_context(ctx)

        # Forward pass
        with autograd.record():
            outputs = net(inputs)
            loss = loss_function(outputs, labels)

        # Backpropagation
        loss.backward()
        metric.update(labels, outputs)

        # Update
        trainer.step(batch_size=inputs.shape[0])

        # Print the evaluation metric and reset it for the next epoch
        name, acc = metric.get()
        history.insert(epoch, acc)
        print('.', end='')
        metric.reset()

    print('[Done]')
```

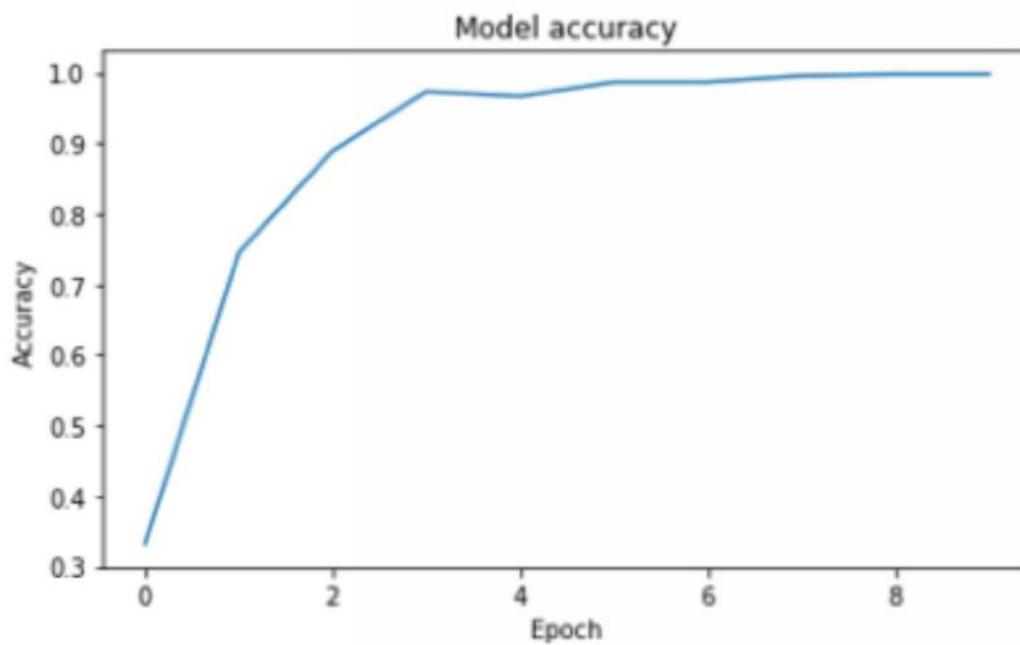
```
# Print the evaluation metric and reset it for the next epoch
name, acc = metric.get()
history.insert(epoch, acc)
print('.', end='')
metric.reset()

print('[Done]')
.....[Done]
```

1. We recorded accuracy statistics for each epoch within the training loop. We will plot this data to obtain a feel of how the training went.

```
In [15]: plt.figure(figsize=(7, 4))
plt.plot(history)
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
```

```
Out[15]: Text(0.5, 0, 'Epoch')
```



2. We utilize the Gluon data loader once more with the test data.

```
In [16]: test_loader = mx.gluon.data.DataLoader(test_data, shuffle=False, t
```

3. We will evaluate the accuracy.

```
In [17]: metric = mx.metric.Accuracy()
for inputs, labels in test_loader:
    # Possibly copy inputs and labels to the GPU
    inputs = inputs.as_in_context(ctx)
    labels = labels.as_in_context(ctx)
    metric.update(labels, net(inputs))
metric_name, metric_value = metric.get()
print(f'Validation: {metric_name} = {metric_value}')

Validation: accuracy = 0.9733333333333334
```

4. We write a few functions to show the results to make our tests seem good.

```
In [18]: # Function to display the image:
def plot_image(predictions_array, true_label, img):
    plt.xticks([])
    plt.yticks([])
    #Image data is currently a flat array. Convert to matrix for a
    plt.imshow(img.reshape((48,48)).asnumpy(), cmap=plt.cm.binary)
    predicted_label = np.argmax(predictions_array)
    probability = 100 * np.max(predictions_array)
    color = 'green' if predicted_label == true_label else 'red'
    # Print a Label with 'predicted class' 'actual class'
    plt.xlabel(f'{class_names[predicted_label]} ({true_label}) - {color=}', color=color)

# Function to display the prediction results in a graph:
def plot_value_array(predictions_array, true_label):
    plt.xticks(range(10))
    plt.yticks([])
    plot = plt.bar(range(10), predictions_array, color="#777777")
    predicted_label = np.argmax(predictions_array)
    plot[predicted_label].set_color('red')
    plot[true_label].set_color('green')
```

5. We will test our model using one of the photos from our test set.

```
In [19]: prediction_image_number = 25
```

6. We are going to make a prediction now.

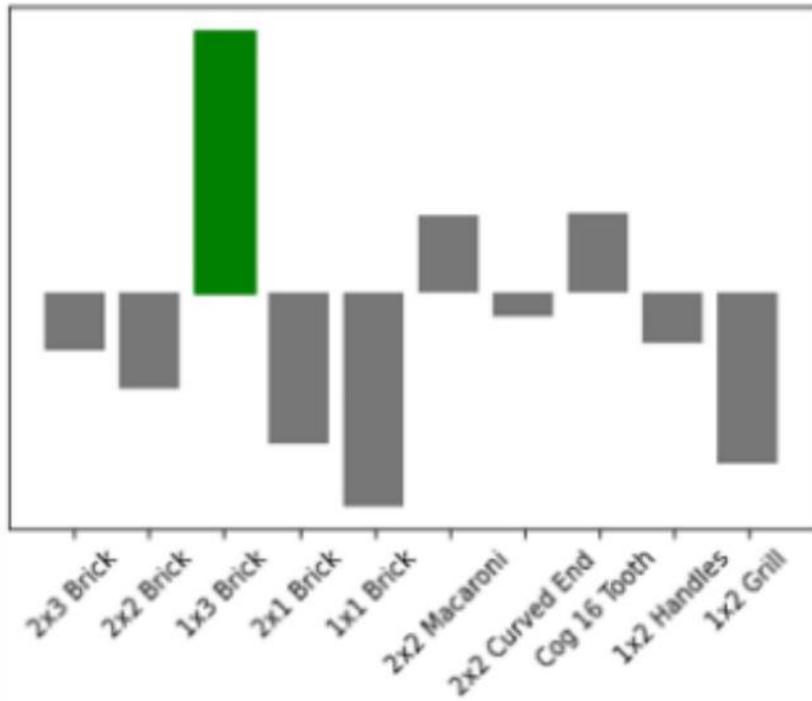
```
In [20]: prediction_image, prediction_label = test_data[prediction_image_nu  
predictions_single = net(prediction_image)  
predictions_single
```

Out[20]:

```
[[ -1.5633558 -2.5973158  6.8950944 -4.0451927 -5.649682   1.98286  
66  
-0.6797639  2.0568316 -1.3924415 -4.5649447]]  
<NDArray 1x10 @cpu(0)>
```

7. Using the helper function, we will create a bar chart. This tells us how accurate the model was in classifying this image. If the projected label differs from the actual label, it will be highlighted in red. Green will be the real label.

```
In [21]: plot_value_array(predictions_single[0].asnumpy(), prediction_label  
plt.xticks(range(10), class_names, rotation=45)  
plt.show()
```



8. To see which block we should have found, we run the following code.

```
In [22]: plot_image(predictions_single[0].asnumpy(), prediction_label, prec
```



1x3 Brick (2 - 1x3 Brick)

Step 5: Make a Batch Prediction

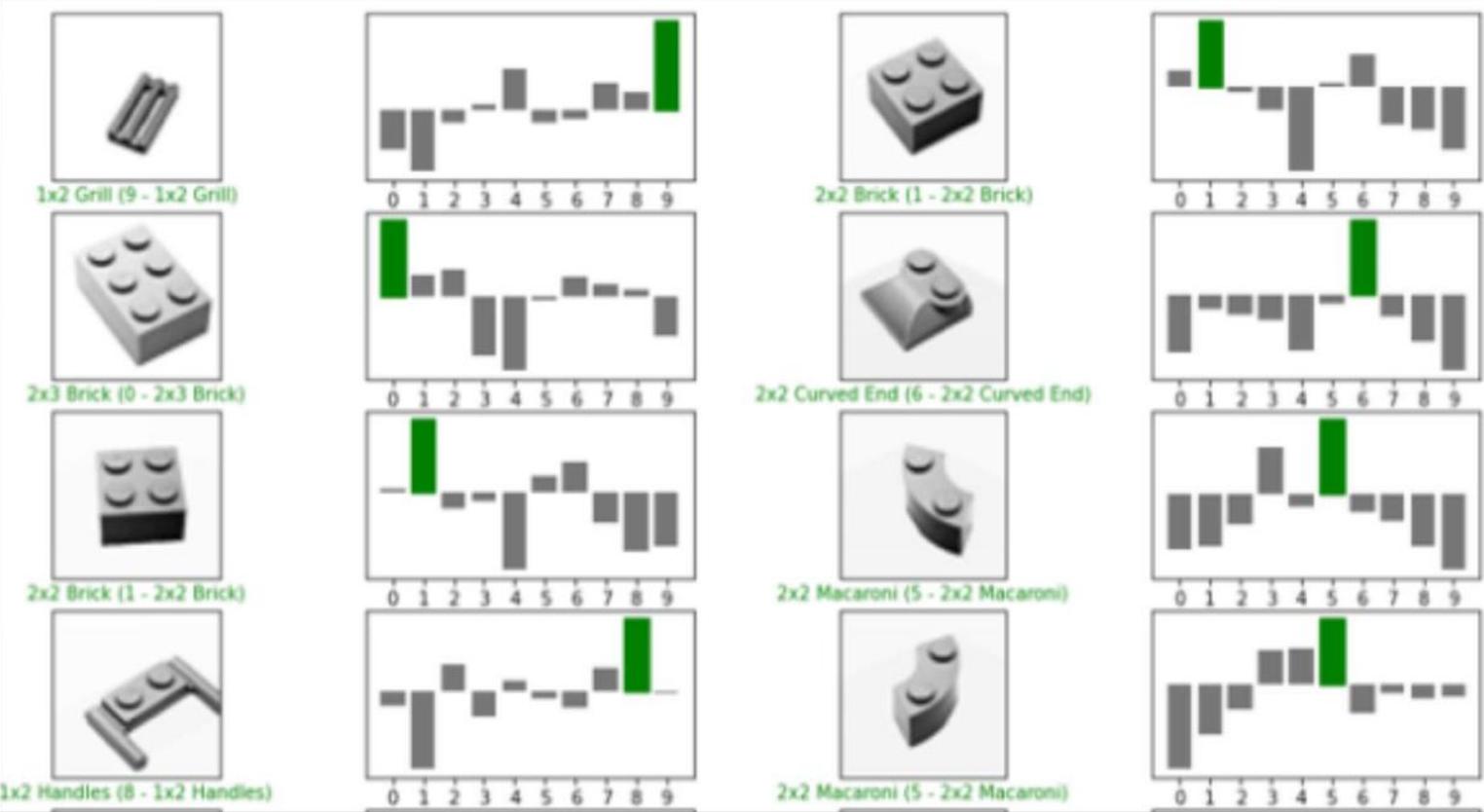
1. We will now obtain prediction values for our test photos. We will not utilize a data loader; instead, we will loop through the test picture data in its raw form.

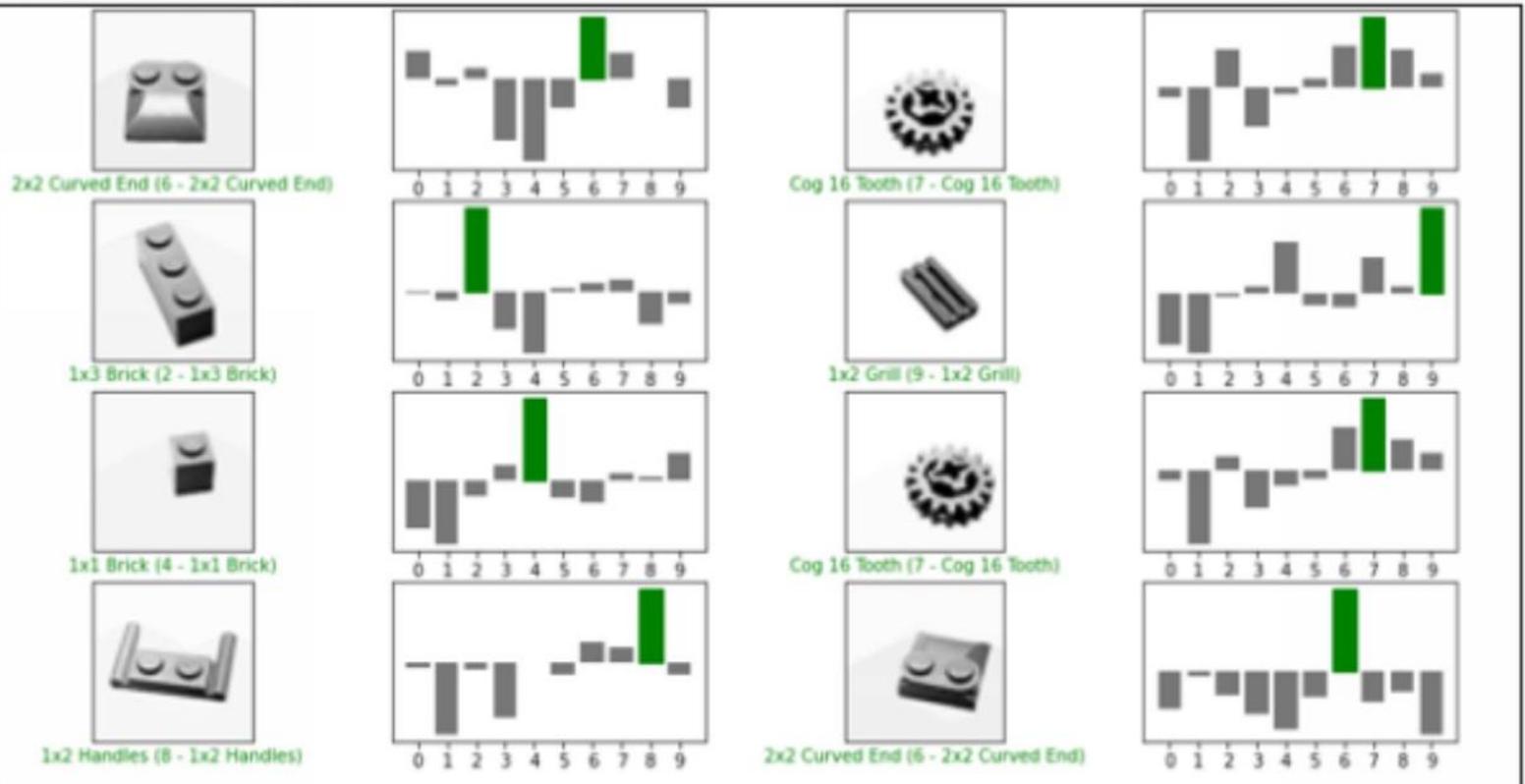
```
In [23]: predictions = []
test_labels = []

for i in test_data:
    pred_image, pred_label = i
    p = net(pred_image)
    predictions.append(p)
    test_labels.append(pred_label)
```

2. To summarize the first 16 photos in our test set, we will use our helper methods.

```
In [ ]: num_rows = 8
        num_cols = 2
        num_images = num_rows*num_cols
        plt.figure(figsize=(15, 16))
        for i in range(num_images):
            plt.subplot(num_rows, 2*num_cols, 2*i+1)
            plot_image(predictions[i].asnumpy(), test_data[i][1], test_dat
            plt.subplot(num_rows, 2*num_cols, 2*i+2)
            plot_value_array(predictions[i][0].asnumpy(), test_data[i][1])
        plt.show()
```





Lab 13: Creating a scikit-learn Random Forest Classifier in AWS SageMaker

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	SageMaker, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

Scikit-learn (Sklearn) is the most usable and robust machine learning package in Python. It uses a Python consistent interface to give a set of fast tools for machine learning and statistical modelings, such as classification, regression, clustering, and dimensionality reduction.

A random forest is a meta estimator that utilizes averaging to enhance prediction accuracy and control over-fitting by fitting several decision tree classifiers on various sub-samples of the dataset. If

bootstrap=True (default), the sub-sample size is regulated by the max samples argument; otherwise, the whole dataset is utilized to create each tree.

Case Study Enterprise Financial Service— Kasasa

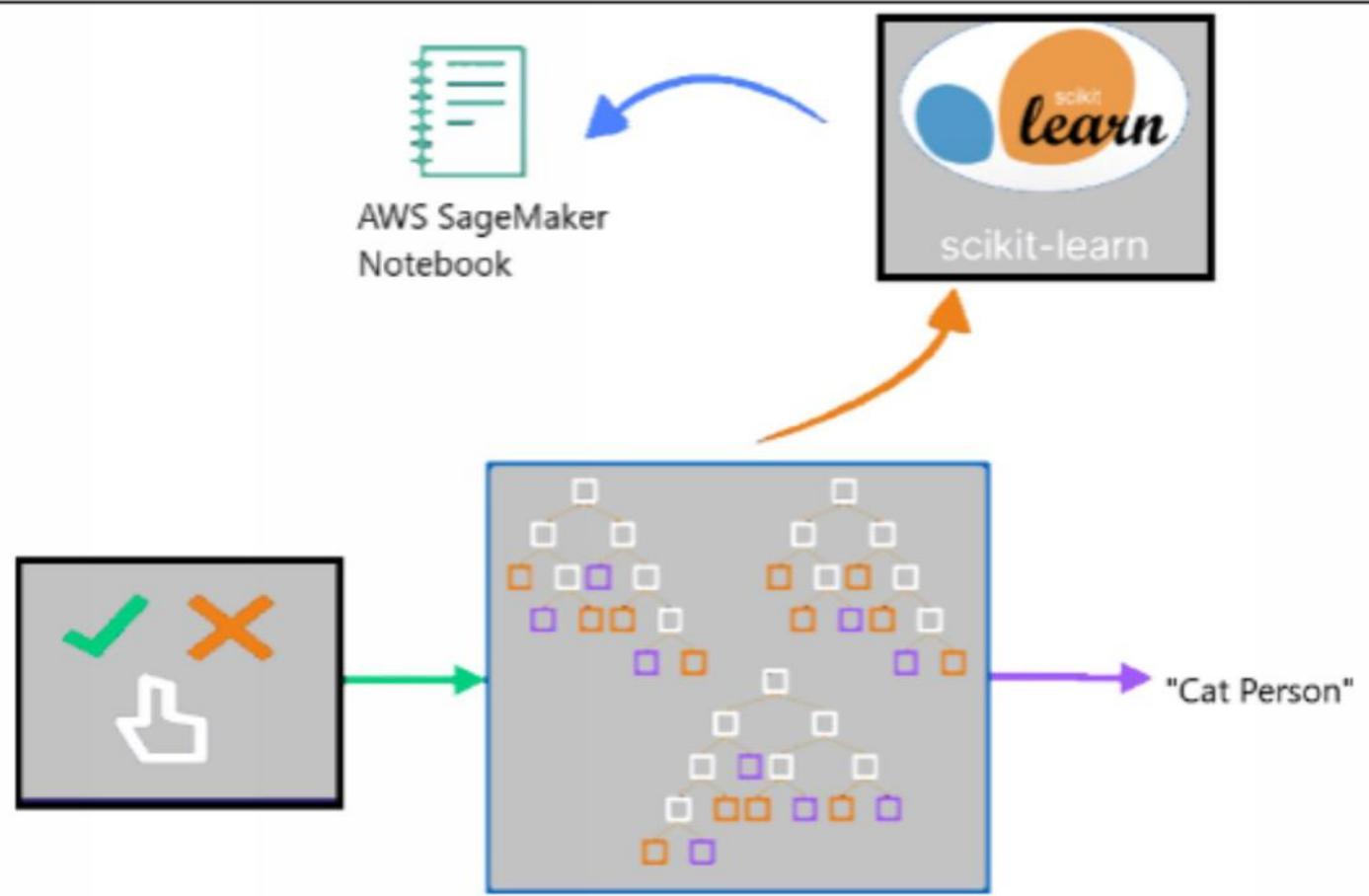
Business Challenge

Kasasa owns and operates a pet store in its neighborhood. Kasasa hired an agency to survey 199 of its current customers to assist you in figuring out what type of items to promote to new consumers. Kasasa clients were asked what their favorite color is, if they enjoy walking or running, and how many kilometers they walk each day. Last but not least, they were asked if they preferred cats or dogs. Subsequently, you will construct a Random Forest Classifier with scikit-learn to determine whether your customers like cats or dogs.

Proposed Solution

The solution is you use AWS services to automate your task and analyze the stored data in real-time within minutes. For storing the data in real-time, you use Amazon Simple Storage S3. You will also construct a Random Forest Classifier with scikit-learn to determine whether your customers like cats or dogs.

Lab Diagram



Implementation Steps

1. Navigate to Jupyter Notebook.
2. Load and Prepare the Data.
3. Create the scikit-learn Model.

4. Evaluate the Model.
5. Predict for Yourself.

Solution

Step 1: Navigate to the Jupyter Notebook

Note: Before starting the lab, you must have a Jupyter notebook. You can create it by following the steps given in the link:

<https://docs.aws.amazon.com/dlami/latest/devguide/setup-jupyter.html>

1. Log in to the **AWS Console**.
2. Go to **Services** and click on **SageMaker**.



Services ▾



More ▾

SageMaker Studio



Cancel

Services (2)

Features (3)

Documentation (46,756)

Knowledge Artic...

Amazon SageMaker

Build, Train, and Deploy Machine Learning Models



AWS Glue DataBrew

Visual data preparation tool to clean and normalize data for analytics and machine learn...

[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

3. Click on In-Service.



Recent activity

Recent activity within the

Last 7 days



Ground Truth

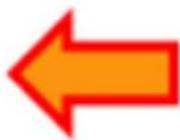
Notebook

Labeling jobs

Notebook instances

No recent activity

1 In Service



1 Created

Training

Inference

Training jobs

Models

No recent activity.

No recent activity.

Hyperparameter tuning jobs

Endpoints

[Feedback](#)

English (US) ▾

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.

4. Click on the **Notebook instance**.

ok
inst
ance
s

Actions ▾

Create notebook instance



Search notebook instances

Status : InService

< 1 >

Name	Instance	Creation time	Status
Notebook-qkmWyCuRVLA2	mlt-jupyter	Oct 16, 2021 12:22 UTC	I

[Feedback](#)

English (US) ▾

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Click on Open Jupyter.



ok
inst
ance
s



Actions ▾

Create notebook instance



Search notebook instances

Status : InService



1



Instance

Creation
time ▾

Status ▾

Actions

ml.t3.medium

Oct 16,
2021 12:22
UTC

Actions

Open
Jupyter | Open
JupyterLab

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Click on CreateAScikit-LearnRandomForestClassifier.ipynb

Files

Running

Clusters

SageMaker Examples

Conda

Select items to perform actions on them.

Upload

New ▾

 0 /

Name ↴

Last Modified

File size

 lost+found

an hour ago

 CreateAScikit-LearnRandomForestClassifier.ipynb

an hour ago

19 kB

 acg_logo.png

an hour ago

13.7 kB

 data.csv

an hour ago

2.84 kB

 penny.png

an hour ago

175 kB

 sparky.png

an hour ago

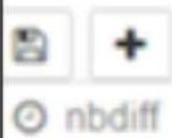
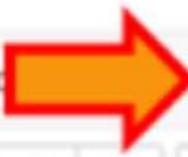
186 kB

7. Make sure you are using the **conda_python3** kernel.

☰ Menu

No

conda_python3 O



A CLOUD GURU

Create a Random Forest Classifier Using scikit-learn



8. We will import all the necessary libraries. We use a magic command to have matplotlib dump the information directly into the notebook. We will use a random seed that ensures that our models are reproducible.

```
In [1]: import numpy as np
import pandas as pd
from pandas.api.types import CategoricalDtype
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sn
#Use a magic command to display the graphs better
%matplotlib inline

from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import export_graphviz
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import train_test_split

# Set the random seed to have reproducible results
np.random.seed(42)

# These two includes help us render a graph
from subprocess import call
from IPython.display import Image
```

Step 2: Load and Prepare the Data

1. We have a dataset of people who like dogs or cats. We begin by loading the data. Pandas have a very simple way of importing CSV data, which is how our survey results are saved.

```
In [2]: df = pd.read_csv("data.csv")
```

2. We will take a look at the top 10 results.

```
In [3]: df.head(10)
```

Out[3]:

	walking	running	color	distance_walked	dogs_cats
0	1	1	red	1	1
1	1	1	blue	2	1
2	1	1	green	3	1
3	1	1	blue	4	1
4	1	1	blue	5	1
5	1	1	red	6	1
6	1	1	green	7	1
7	1	1	green	8	1
8	1	1	blue	9	1
9	1	1	red	1	1

3. We will check to see what data types we have.

```
In [4]: df.dtypes
```

```
Out[4]: walking           int64
running            int64
color              object
distance_walked   int64
dogs_cats          int64
dtype: object
```

4. We will start by changing the labels of some of the columns and identifying dogs_cats as an out label for our first data preparation.

```
In [5]: df.columns = ['walk', 'run', 'color', 'distance', 'label']
```

5. The data is not adequately shuffled. We come across a lot of information that is similar. We do not wish to examine the remainder of our data separately to better understand. Rather, we want to describe the data as its whole. We will use Pandas to provide extra statistical information about the data.

```
In [6]: df.describe()
```

Out[6]:

	walk	run	distance	label
count	199.000000	199.000000	199.000000	199.000000
mean	0.497487	0.497487	4.979899	0.321608
std	0.501255	0.501255	2.597512	0.468272
min	0.000000	0.000000	1.000000	0.000000
25%	0.000000	0.000000	3.000000	0.000000
50%	0.000000	0.000000	5.000000	0.000000
75%	1.000000	1.000000	7.000000	1.000000
max	1.000000	1.000000	9.000000	1.000000

6. We will structure the data in columns for our next data preparation phase so the model can comprehend it better.

Even though the **walk**, **run**, and **label** columns are integers, they indicate a binary value of 1 or 0. This relates to True or False for liking that activity for walk and run. This indicates whether you favor cats or dogs as a label. We will use the boolean data type instead of numbers for these fields.

Color_type is not just any text. It belongs to one of three groups (red, green, or blue). We will convert it to a definite data type instead of considering it as text.

```
In [7]: df['walk'] = df['walk'].astype('bool')
df['run'] = df['run'].astype('bool')
color_type = CategoricalDtype(categories=['red', 'green',
df['color'] = df['color'].astype(color_type)
df['label'] = df['label'].astype('bool')
```

7. We will check to see what data types we have.

```
In [8]: df.dtypes
```

```
Out[8]: walk          bool  
run           bool  
color         category  
distance      int64  
label          bool  
dtype: object
```

8. The algorithm will not understand color as a list of strings. When we try to match the data with the current data, we may expect to obtain the following error:

`ValueError: could not convert string to float: 'green.'`

As a result, we utilize the 'get dummies' command to conduct one-hot encoding straight into the dataframe.

```
In [9]: df = pd.get_dummies(df, prefix=['color'])
```

9. We will take a quick look at our data now that the data formatting is complete.

```
In [10]: df.head(10)
```

Out[10]:

	walk	run	distance	label	color_red	color_green	color_blue
0	True	True	1	True	1	0	0
1	True	True	2	True	0	0	1
2	True	True	3	True	0	1	0
3	True	True	4	True	0	0	1
4	True	True	5	True	0	0	1
5	True	True	6	True	1	0	0
6	True	True	7	True	0	1	0
7	True	True	8	True	0	1	0
8	True	True	9	True	0	0	1
9	True	True	1	True	1	0	0

10. We use train_test_split from scikit-learn to split out 20% of the data for testing, leaving us with 80% for training. We also set random_state to ensure that the data split is random yet reproducible.

```
In [11]: X_train, X_test, y_train, y_test = train_test_split(df.dro
```

Step 3: Create the scikit-learn Model

1. We are going to create a RandomForestClassifier model.

```
In [12]: model = RandomForestClassifier(max_depth=5)
```

2. Now, we will fit the model on the training data.

```
In [13]: model.fit(X_train, y_train)
```

```
Out[13]: RandomForestClassifier(max_depth=5)
```

Step 4: Evaluate the Model

1. First, we will grab the estimator from the trained model.

```
In [14]: estimator = model.estimators_[0]
feature_names = [i for i in X_train.columns]
```

2. Then, we will use export_graphviz to get a graphical representation of one of the trees in our forest.

```
In [15]: export_graphviz(estimator, out_file='tree.dot',
                      feature_names = feature_names,
                      rounded = True,
                      filled = True)

# Use CLI Graphviz to create a png file from our graph data:
call(['dot', '-Tpng', 'tree.dot', '-o', 'tree.png', '-Gdpi=600'])

Image(filename = 'tree.png')
```

```
Out[15]:
```



You may double-click to zoom the image.

Out[16]:

```
run <= 0.5
gini = 0.44
samples = 107
value = [107, 52]
```

3. Now, we will run our testing data through the model.

```
In [16]: y_predict = model.predict(X_test)
```

4. To see how it performed, we will use confusion_matrix to create a confusion matrix for the test data predictions.

```
In [17]: conf_matrix = confusion_matrix(y_test, y_predict)
conf_matrix
```

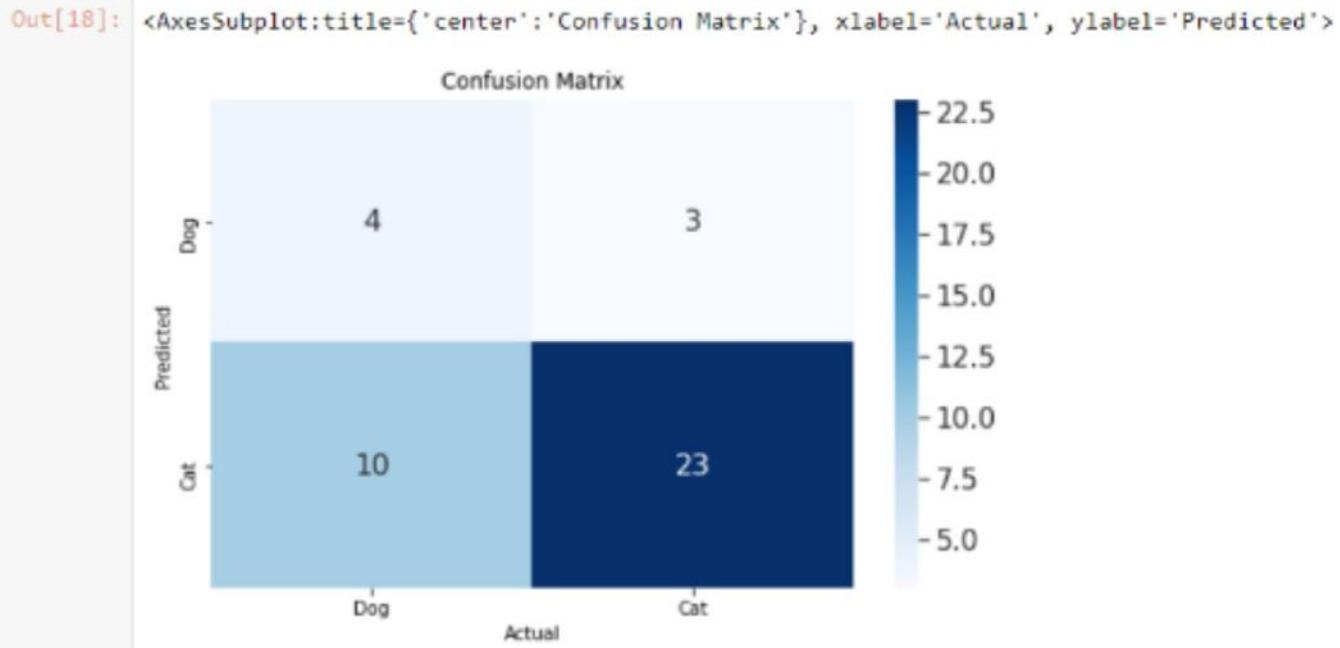
```
Out[17]: array([[23,  3],
                 [10,  4]])
```

5. Since it is not easy to read, we will add some style to it using the seaborn library.

```
In [18]: tn, fp, fn, tp = conf_matrix.ravel()
cleaned_conf_matrix = np.array([[tp, fp], [fn, tn]])
y_true = ["Dog", "Cat"]
df_cm = pd.DataFrame(cleaned_conf_matrix, columns=y_true, index=y_true)
df_cm.index.name = 'Predicted'
df_cm.columns.name = 'Actual'

df_cm.dtypes

plt.figure(figsize = (8,5))
plt.title('Confusion Matrix')
sn.set(font_scale=1.4)#for label size
sn.heatmap(df_cm, cmap="Blues", annot=True, annot_kws={"size": 16})# font size
```



6. From the confusion matrix, we will calculate sensitivity and specificity.

```
In [19]: sensitivity = tp / (tp + fn)
print('Sensitivity : ', sensitivity)

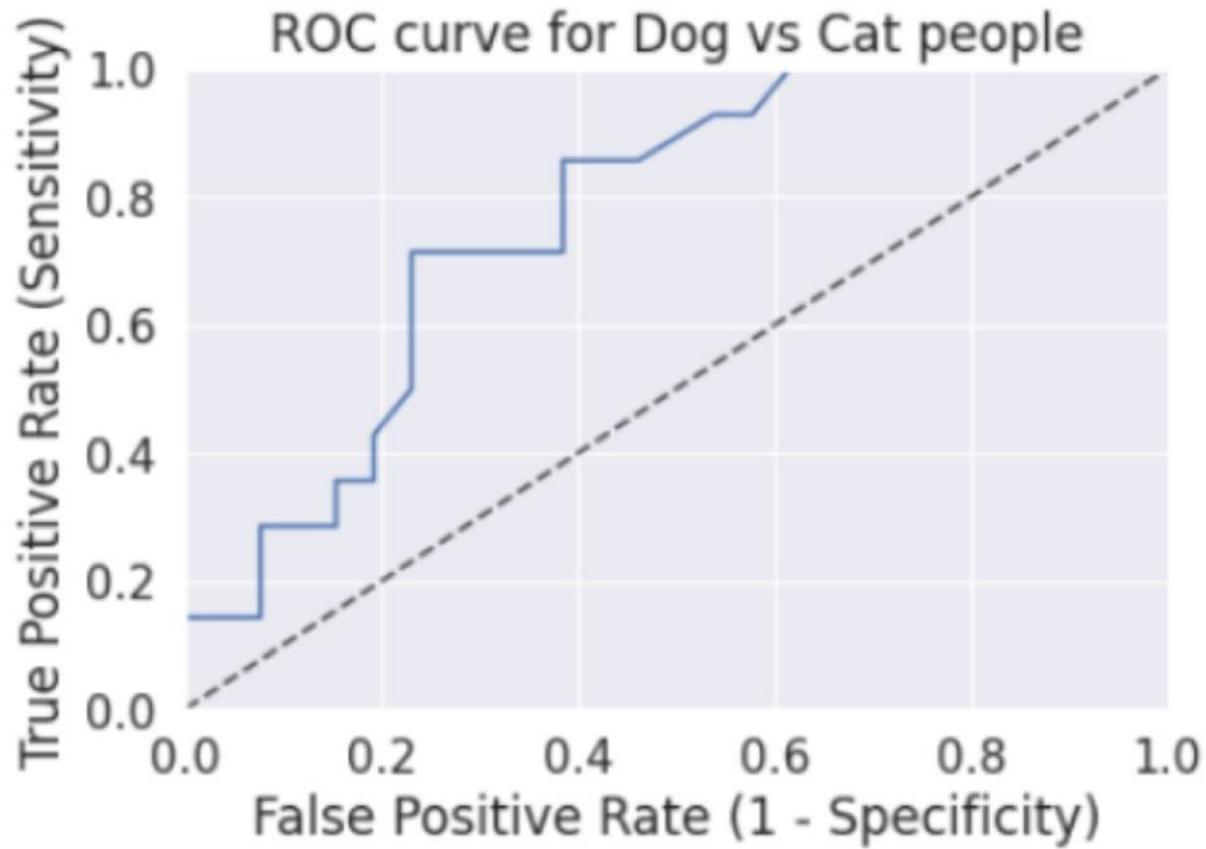
specificity = tn / (tn + fp)
print('Specificity : ', specificity)

Sensitivity :  0.2857142857142857
Specificity :  0.8846153846153846
```

7. Now, we will look at the ROC graph.

```
In [20]: y_pred_proba = model.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)

fig, ax = plt.subplots()
ax.plot(fpr, tpr)
ax.plot([0, 1], [0, 1], transform=ax.transAxes, ls="--", c=".3")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.rcParams['font.size'] = 12
plt.title('ROC curve for Dog vs Cat people')
plt.xlabel('False Positive Rate (1 - Specificity)')
plt.ylabel('True Positive Rate (Sensitivity)')
plt.grid(True)
```



8. Now, we will calculate the AUC (area under the curve).

```
In [21]: auc(fpr, tpr)
```

```
Out[21]: 0.7664835164835165
```

Step 5: Predict for Yourself

1. You can change the values for the survey questions below and see what the model predicts for you!

```
In [ ]: # Do you like walking? 1=yes, 0=no
like_walking = 1

# Do you like running? 1=yes, 0=no
like_running = 0

# How far do you walk in a day?
distance = 3

# What is your favorite color? Answer ONLY ONE with a 1
red = 0
green = 0
blue = 1

# Get the prediction:
prediction = model.predict([[like_walking, like_running, distance, red, green, blue]])

# The following is all display code:
if prediction[0]:
    animal = 'dog'
    pic = 'sparky.png'
else:
    animal = 'cat'
    pic = 'penny.png'

img=mpimg.imread(pic)
imgplot = plt.imshow(img)
plt.grid(False)
plt.xticks([])
plt.yticks([])
plt.title("We predict that you're a %s person!" % animal)
plt.show()
```

We predict that you're a cat person!



Lab 14: Evaluation and Optimization

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	CloudWatch, S3, SageMaker, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

AWS SageMaker

Amazon SageMaker is a fully managed service that allows any developer or data scientist to quickly create, train, and deploy Machine Learning (ML) models. SageMaker makes it easy to create high-quality models by removing the heavy lifting from each phase of the machine learning process.

AWS Simple Storage Service (S3)

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 is a web service that allows you to store and retrieve an infinite quantity of data from any place and at any time. You may quickly create projects that integrate cloud-native storage using this service. Because Amazon S3 is easily customizable and you only pay for what you use, you can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. You can build a simple FTP system or a complex web application like the Amazon.com retail website, read the same piece of data a million times or only for emergency disaster recovery, and store whatever type and amount of data you desire.

AWS CloudWatch

Amazon CloudWatch is a tracking service for Amazon Web Services (AWS) cloud services and software. Amazon CloudWatch may be used to collect and monitor data, monitor log files, and trigger alarms. Amazon CloudWatch can monitor AWS resources, such as Amazon EC2 instances, Amazon DynamoDB tables, Amazon RDS DB instances, and custom metrics and log files created by your applications and services. You can watch your system's resource use, application performance, and operational health using Amazon CloudWatch. These insights might help you react and keep your app working smoothly.

Case Study Enterprise Retail – LexisNexis Legal & Professional

Background

A leading global source of legal, regulatory, and business information and analytics, LexisNexis Legal & Professional supports customers' efforts to strengthen the rule of law globally, boost productivity, and improve decision-making and outcomes.

Business Challenge

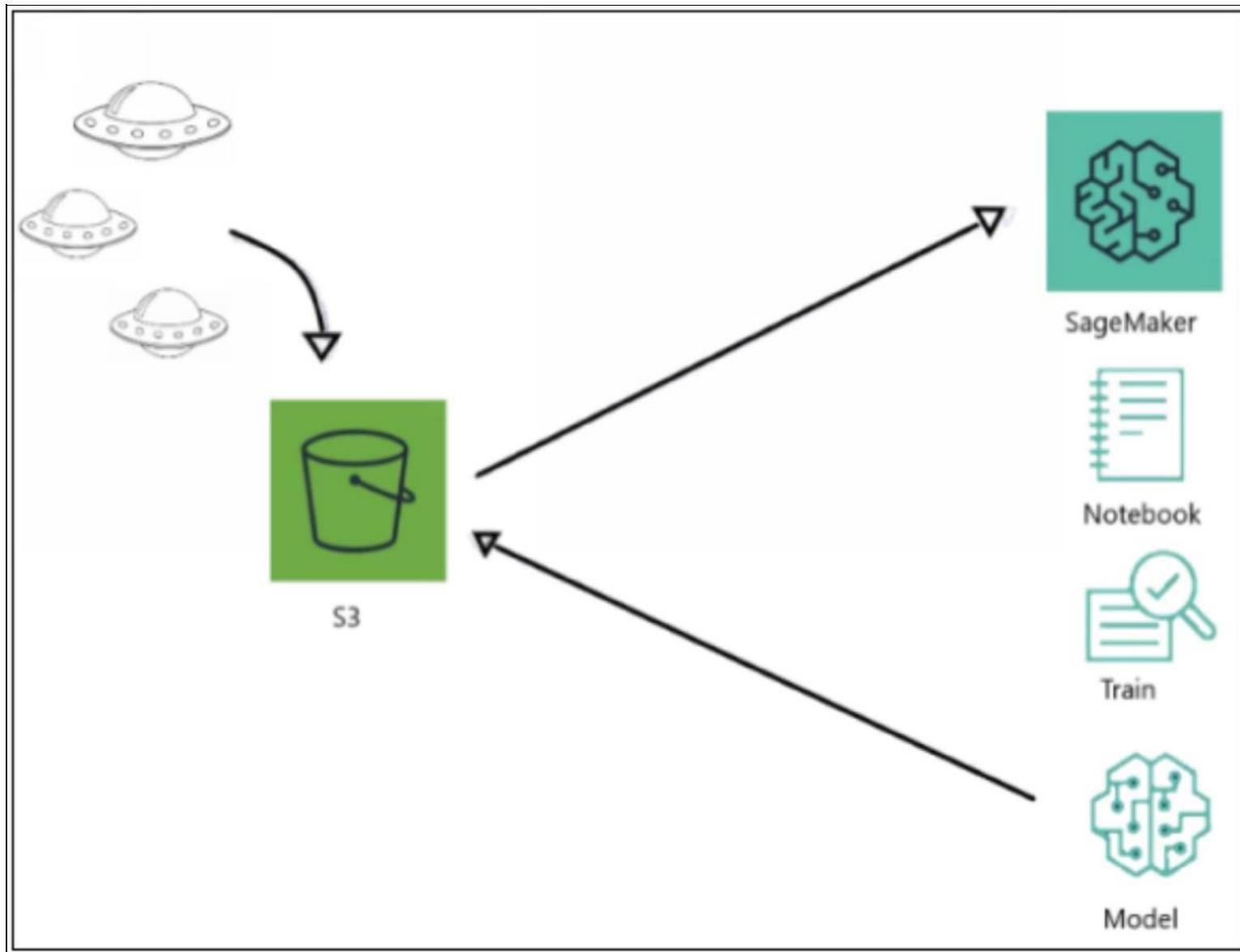
You work as a Machine Learning developer in LexisNexis Legal & Professional. Your organization gives you a task to train the customer classification model. They also give you the task tune model to find the most optimized model for the problem. Determine if the model is less accurate or about the same. What objective metric would you want to monitor to ensure this? How do you plan on measuring success? Which hyperparameters need to be tuned? What combination do hyperparameters need to be used? How much faster was training time improved?

Proposed Solution

The solution is for you to use AWS services to automate all the tasks. You create an AWS SageMaker hyperparameter tuning job with different ranges of values for the hyperparameter to find the best

configuration, which minimizes the validation: objective_loss metric. The reason that you use this metric is that it is used in multi-classification problems. This metric measures the performance of the classification model, and what it does is repeatably calculate the difference between the values that the model predicts and the actual values of a label. Hence, every time it passes over data and makes predictions, it recalculates the objective loss and tries to minimize this value overall. Hence, that is the task here. AWS recommends that you minimize this value when using it as our objective metric.

Lab Diagram



Implementation Steps

1. Create an S3 Bucket.
2. Create SageMaker Hyperparameter Training Job.
3. Analyze CloudWatch Training Logs.
4. Create SageMaker Notebook.

Solution

Step 1: Create S3 Bucket

1. Log in to the **AWS Console**.
2. Click on the **Services**.

The screenshot shows the AWS Management Console with the 'Services' dropdown menu open. A red box highlights the 'Services' button, and a red arrow points to it from the top left. The 'AWS services' section is visible, showing 'Recently visited services' and a link to 'All services'. The bottom navigation bar includes links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

3. Select S3 from the Storage.

The screenshot shows the 'All services' page in the AWS Management Console. On the left, there's a sidebar with 'Favorites' and 'Recently visited' sections. The main area lists various AWS services under categories like 'Storage', 'Compute', etc. A red box highlights the 'S3' service under the 'Storage' category, and a red arrow points to it from the bottom right.

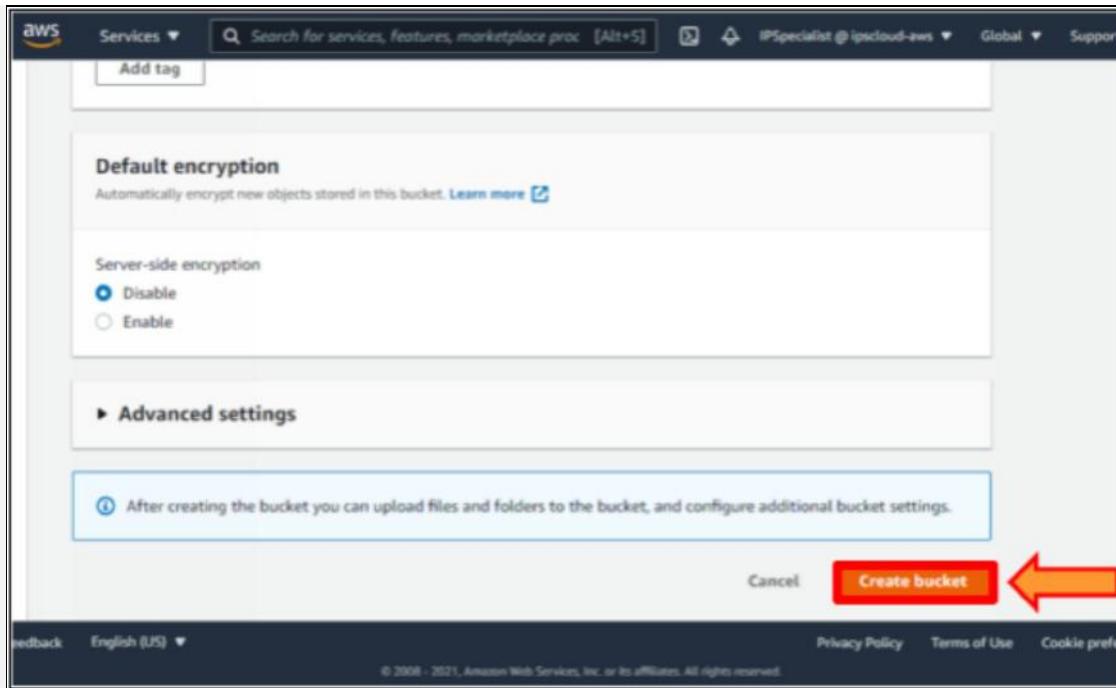
4. Click on the **Create bucket** button.

The screenshot shows the AWS S3 service page. On the left, there's a sidebar with links like 'Buckets', 'Access Points', and 'Storage Lens'. The main area has a section titled 'Account snapshot' with metrics: Total storage (106.5 KB), Object count (1), and Avg. object size (106.5 KB). A note says you can enable advanced metrics in the 'default-account-dashboard' configuration. Below this is a list of buckets with one item: 'Buckets (1)'. It includes a 'Create bucket' button highlighted with a red arrow.

5. Give a bucket name ips-s3-bucket.

The screenshot shows the 'Create bucket' wizard. In the 'General configuration' step, the 'Bucket name' field contains 'ips-s3-bucket', which is also highlighted with a red arrow. Other fields include 'AWS Region' set to 'US East (N. Virginia) us-east-1' and a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button.

6. Scroll down. Click on the Create bucket button.



7. Download the `ufo_sightings_train_recordIO_protobuf.data` and `ufo_sightings_validation_recordIO_protobuf.data` files from the following Github link:

https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/tree/master/Chapter

The screenshot shows a GitHub repository page for 'Course_AWS_Certified_Machine_Learning / Chapter8 /'. It displays a list of commits, with the last three being:

- brocktube Major updates to fix issues with deprecated SageMaker library methods... - ab99654 on Nov 10, 2020
- ufo-evaluation-optimization-lab.ipynb Major updates to fix issues with deprecated SageMaker library methods... 11 months ago
- ufo_sightings_train_recordIO_protobuf.data Added training and validation files 2 years ago
- ufo_sightings_validation_recordIO_protobuf.data Added training and validation files 2 years ago

8. Click on **ips-s3-bucket**.

The screenshot shows the AWS Storage Lens dashboard. At the top, there's a summary section with metrics: Total storage (106.5 KB), Object count (1), and Avg. object size (106.5 KB). A note says you can enable advanced metrics in the "default-account-dashboard" configuration. Below this is a section titled "Buckets (1) Info". It shows one bucket named "ips-s3-bucket" located in "US East (N. Virginia) us-east-1". The bucket has "Access" set to "Bucket and objects not public" and was created on "October 2, 2021, 04:10:07 (UTC-07:00)". Below the table are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". A search bar and navigation controls are also present.

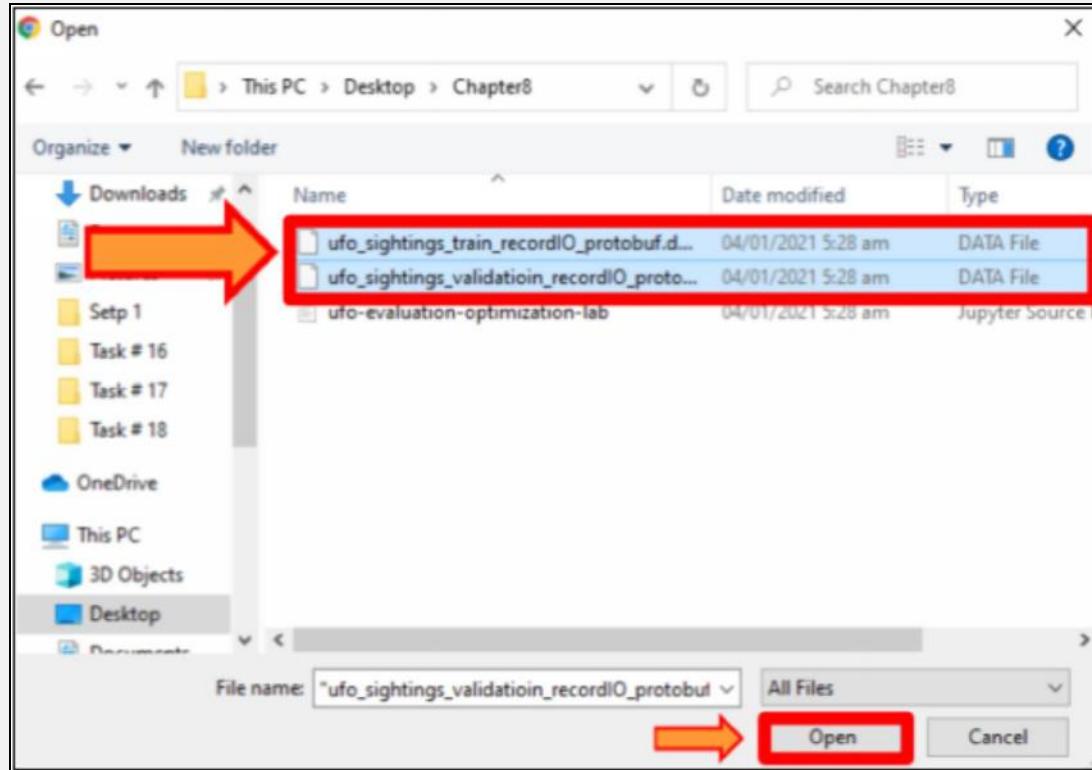
9. Click on the **Upload** button.

The screenshot shows the "Objects (0)" page for a bucket. At the top, there are buttons for "Create folder" and "Upload", with "Upload" being highlighted by a red arrow. Below the buttons is a search bar and a table header with columns: Name, Type, Last modified, Size, and Storage class. The main area displays "No objects" and "You don't have any objects in this bucket." A large "Upload" button is centered at the bottom of the page.

10. Click on the **Add files** button.

The screenshot shows the AWS S3 console interface. At the top, there's a search bar and navigation links. Below it, a large text area says "Drag and drop files and folders you want to upload here, or choose Add files, or Add folders." A red arrow points to the "Add files" button, which is highlighted with a red border. Next to it is the "Add folder" button. Below this, there's a table header for "Files and folders (0)" with columns for Name, Folder, Type, and Size. A search bar labeled "Find by name" is at the top of the table. The main message in the center says "No files or folders" and "You have not chosen any files or folders to upload." At the bottom, there's a "Destination" section and a footer with links for Privacy Policy, Terms of Use, and Cookie preferences.

11. Select the **ufo_sightings_train_recordIO_protobuf.data** and **ufo_sightings_validation_recordIO_protobuf.data** files.
12. Click on the **Open** button.



13. Scroll down. Click on the **Upload** button.

Screenshot of the AWS S3 console showing the upload progress of two files: "ufo_sightings_validation_recordIO.protobuf.data" and "228.6 KB".

Destination:
s3://ips-s3-bucket

Permissions:
Grant public access and access to other AWS accounts.

Properties:
Specify storage class, encryption settings, tags, and more.

Upload button highlighted with a red arrow.

Hence, the files have been uploaded.

Screenshot of the AWS S3 console showing the upload status summary.

Upload succeeded:
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://ips-s3-bucket	2 files, 2.0 MB (100.00%)	0 files, 0 B (0%)

Files and folders | Configuration

Feedback: English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2: Create SageMaker Hyperparameter Training Job

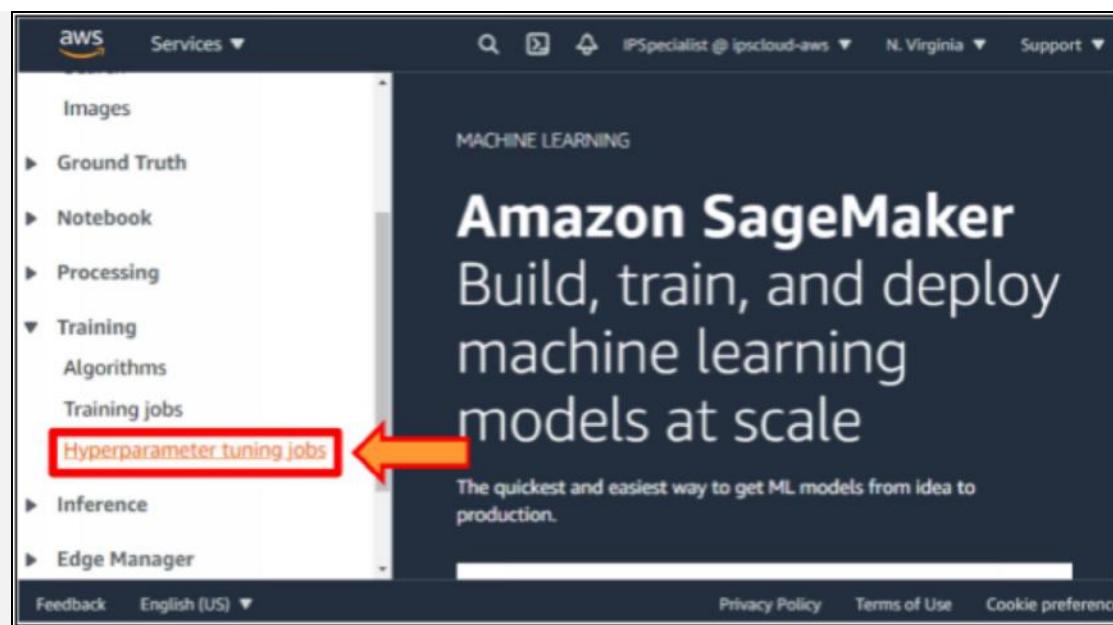
1. Click on Services.

The screenshot shows the AWS Management Console homepage. At the top left, there is a red box around the "Services" dropdown menu. An orange arrow points from the top center towards this red box. The main title "AWS Management Console" is centered at the top. Below it is a sidebar titled "AWS services" with sections for "Recently visited services" and "All services". At the bottom of the page, there is a footer bar with links for "Privacy Policy", "Terms of Use", and "Cookie preferences".

2. Select SageMaker from Machine Learning.

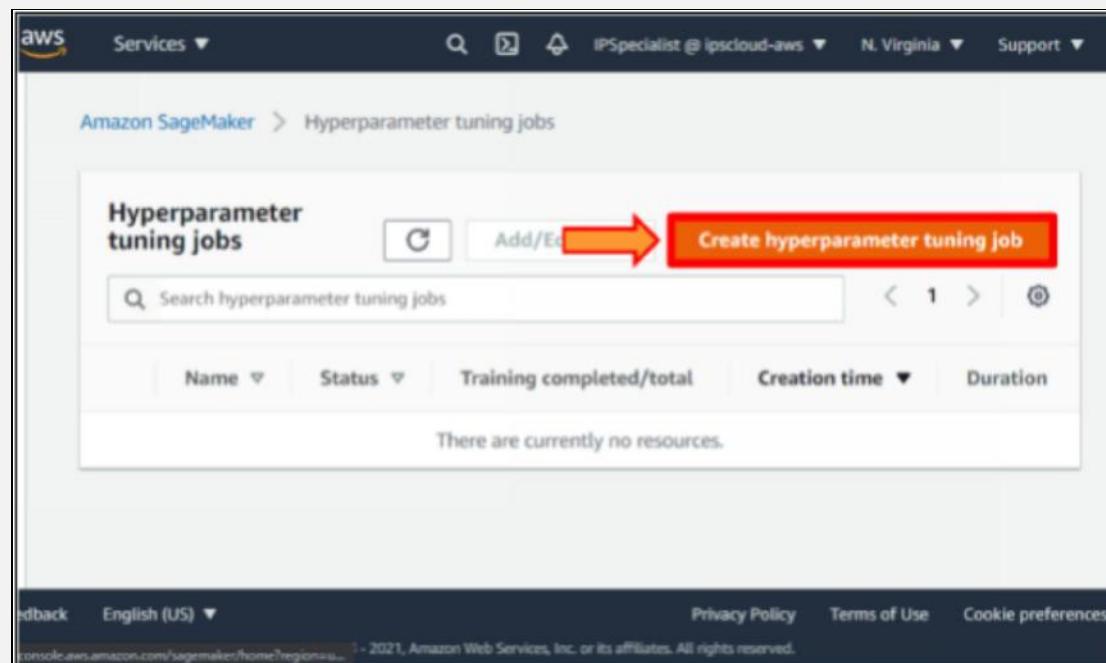
The screenshot shows the AWS Management Console with the "Machine Learning" section selected. A red box highlights the "Amazon SageMaker" link, and an orange arrow points to it from the right side. The "Favorites" and "Recently visited" sections are also visible on the left.

3. Click on the Hyperparameter training jobs from the left-hand side.



The screenshot shows the Amazon SageMaker home page. On the left, there is a sidebar with the following navigation options: Images, Ground Truth, Notebook, Processing, Training (with Algorithms and Training jobs), Hyperparameter tuning jobs (which is highlighted with a red box and has an orange arrow pointing to it), Inference, and Edge Manager. The main content area features the heading "MACHINE LEARNING" and the title "Amazon SageMaker: Build, train, and deploy machine learning models at scale". Below the title, a sub-section titled "Hyperparameter tuning jobs" is described with the text: "The quickest and easiest way to get ML models from idea to production." At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

4. Click on the **Create hyperparameter tuning job** button.



The screenshot shows the "Hyperparameter tuning jobs" page within the Amazon SageMaker service. The top navigation bar includes the AWS logo, Services dropdown, search icon, user info (IPSpecialist @ ipscloud-aws), region (N. Virginia), and support link. The main content area displays a table header for "Hyperparameter tuning jobs" with columns: Name, Status, Training completed/total, Creation time, and Duration. A search bar labeled "Search hyperparameter tuning jobs" is present. Below the table, a message states "There are currently no resources." At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences, along with a copyright notice: "console.aws.amazon.com/sagemaker/home?region=us-east-1 © 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved."

5. Give the job name **IPS-linear-learner-tuning-job**.

Sales

Services ▾

parameters

Define data input and output

Configure resources

Step 3

Configure tuning job resources

Step 4

Review and create

Job settings

Hyperparameter tuning job name

Amazon SageMaker adds this name to the name of training jobs launched by this tuning job. For example, for the name IPS-linear-learner-tuning-job, the training job name is IPS-linear-learner-tuning-job-XXXX-XXXX-XXXX-XXXX.

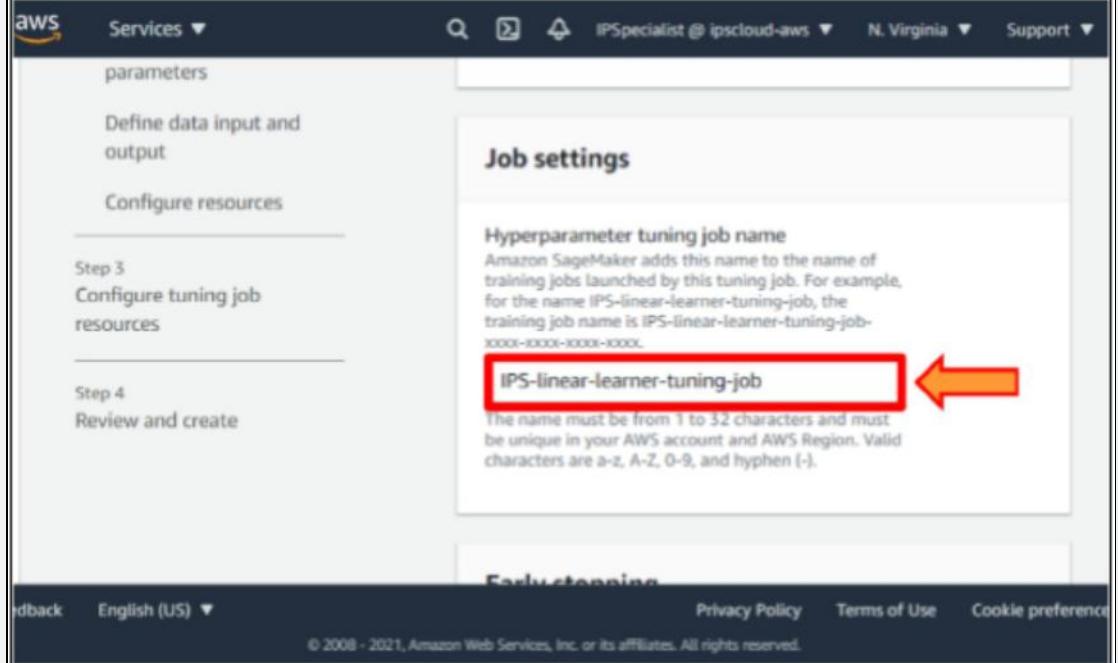
IPS-linear-learner-tuning-job

The name must be from 1 to 32 characters and must be unique in your AWS account and AWS Region. Valid characters are a-z, A-Z, 0-9, and hyphen (-).

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



6. Select **off** for early stopping.

Sales

Services ▾

Early stopping

Early stopping stops training jobs when they are unlikely to improve the current best objective metric of the hyperparameter tuning job. [Learn more](#)

Training job early stopping type

Off

Tuning Strategy

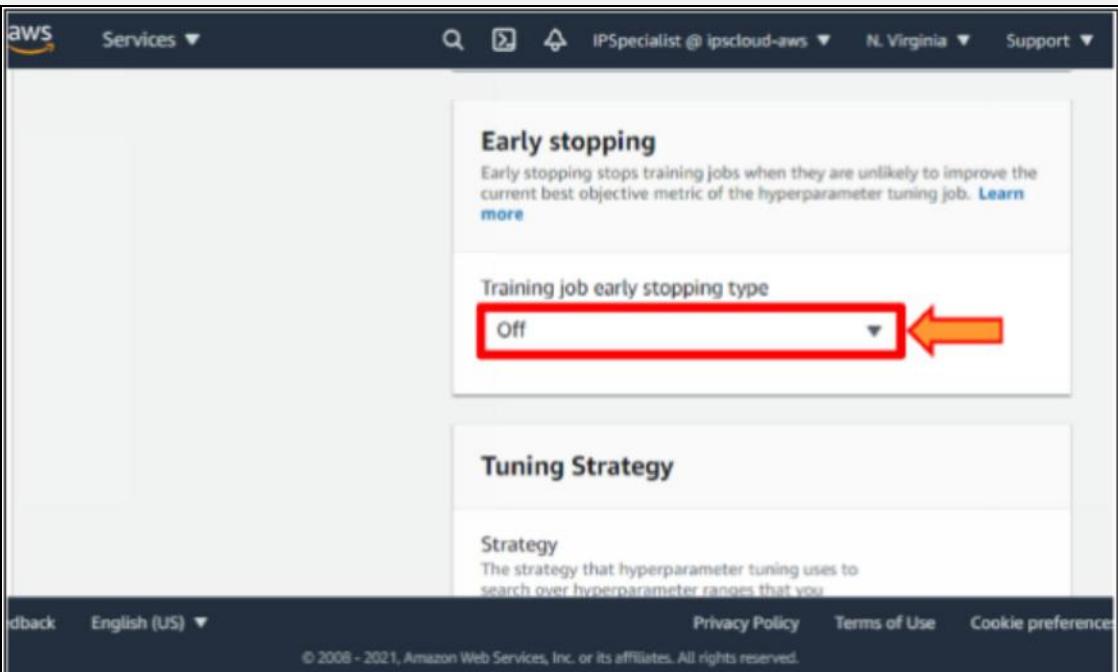
Strategy

The strategy that hyperparameter tuning uses to search over hyperparameter ranges that you

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



7. Select the **Bayesian** strategy.

The screenshot shows the 'Tuning Strategy' configuration page. At the top, there's a search bar and navigation links for 'Services', 'Support', and account information ('IPSpecialist @ ipscloud-aws', 'N. Virginia'). Below the header, a section titled 'Tuning Strategy' contains a 'Strategy' dropdown menu. The dropdown is currently set to 'Bayesian' and is highlighted with a red rectangular border and a red arrow pointing towards it from the right side of the image. Below the dropdown is a link 'Tags - optional'. At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the very bottom states '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

8. Click on the **Next** button.

This screenshot is identical to the one above, showing the 'Tuning Strategy' configuration page. It features the same layout, including the 'Strategy' dropdown set to 'Bayesian' with a red border and arrow, and the 'Tags - optional' section. At the bottom, the 'Cancel' and 'Next' buttons are visible. The 'Next' button is highlighted with a large red rectangular border and a red arrow pointing towards it from the left side of the image. The footer links and copyright notice are also present.

9. Click on the **Add training job definition** button.

Sales

Services ▾

IPSspecialist @ ipscloud-aws ▾ N. Virginia ▾ Support ▾

Create hyperparameter tuning job

Step 1
Define job settings

Step 2
Create training job definition

Configure algorithm and parameters
Define data input and output
Configure resources

Training Job Definition(s)
Choose one or more algorithms you want to use for hyperparameter tuning. [Learn more](#)

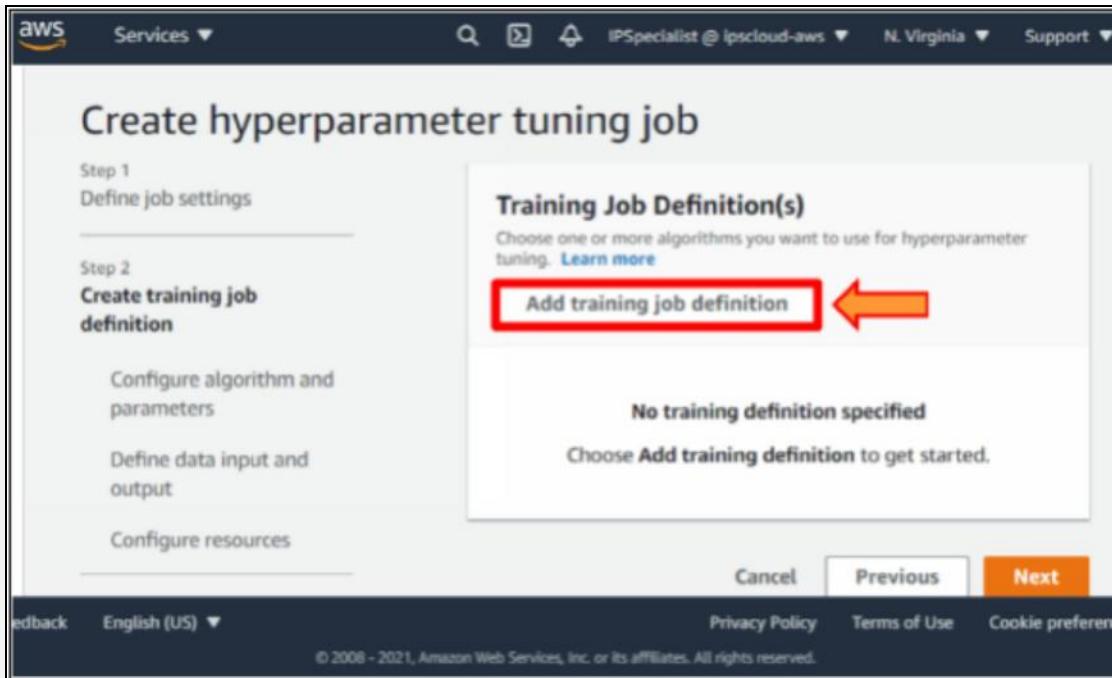
Add training job definition

No training definition specified
Choose Add training definition to get started.

Cancel Previous **Next**

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



10. Give the training job definition name **IPS-linear-learner-tuning-job**.

Sales

Services ▾

IPSspecialist @ ipscloud-aws ▾ N. Virginia ▾ Support ▾

Create hyperparameter tuning job

Step 1
Define job settings

Step 2
Create training job definition

Configure algorithm and parameters

Define data input and output
Configure resources

Step 3
Configure tuning job

RESOURCES

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

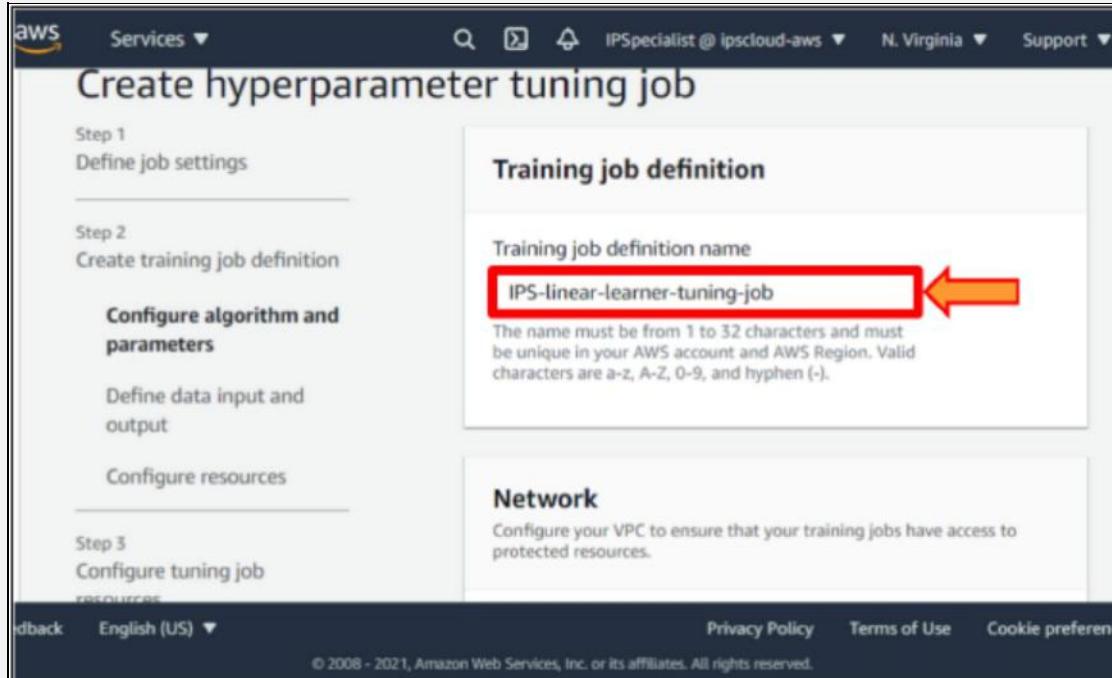
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training job definition

Training job definition name **IPS-linear-learner-tuning-job**

The name must be from 1 to 32 characters and must be unique in your AWS account and AWS Region. Valid characters are a-z, A-Z, 0-9, and hyphen (-).

Network
Configure your VPC to ensure that your training jobs have access to protected resources.



11. Select the **AmazonSageMaker-ExecutionRole** IAM role.

Screenshot of the AWS SageMaker console showing the IAM role selection step. The IAM role dropdown menu is open, displaying several options. The option 'AmazonSageMaker-ExecutionRole-20211001T004641' is highlighted with a red box and an orange arrow points to it.

12. Scroll down. Select the **Linear Learner** algorithm.

Screenshot of the AWS SageMaker console showing the algorithm selection step. The 'Choose an algorithm' dropdown menu is open, displaying the 'Linear Learner' option, which is highlighted with a red box and an orange arrow points to it.

13. Select the **Pipe** input mode.

AWS Services ▾

IPSspecialist @ ipscloud-aws ▾ N. Virginia ▾ Support ▾

Choose an algorithm

Linear Learner

Container

The registry path where the training image is stored in Amazon ECR. [Learn more](#)

382416733822.dkr.ecr.us-east-1.amazonaws.com/linear-learner:1

Input mode

You can provide your training data as a file or pipe.

Pipe

Enable SageMaker metrics time series

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

14. Select the **validation:objective_loss** objective. Then Select the **minimize** type.

validation:roc_auc_score

#quality_metric: host=\\$+, validation roc_auc_score <score>=(\\$+)

Objective metric

To find the best training job, set an objective metric and tuning type. See the hyperparameter tuning job detail page for a summary of the best training job.

Objective metric

validation:objective_loss

Type

minimize

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

15. Type 22 in the **feature_dim**.

Sales Support

Services ▾

Hyperparameter configuration

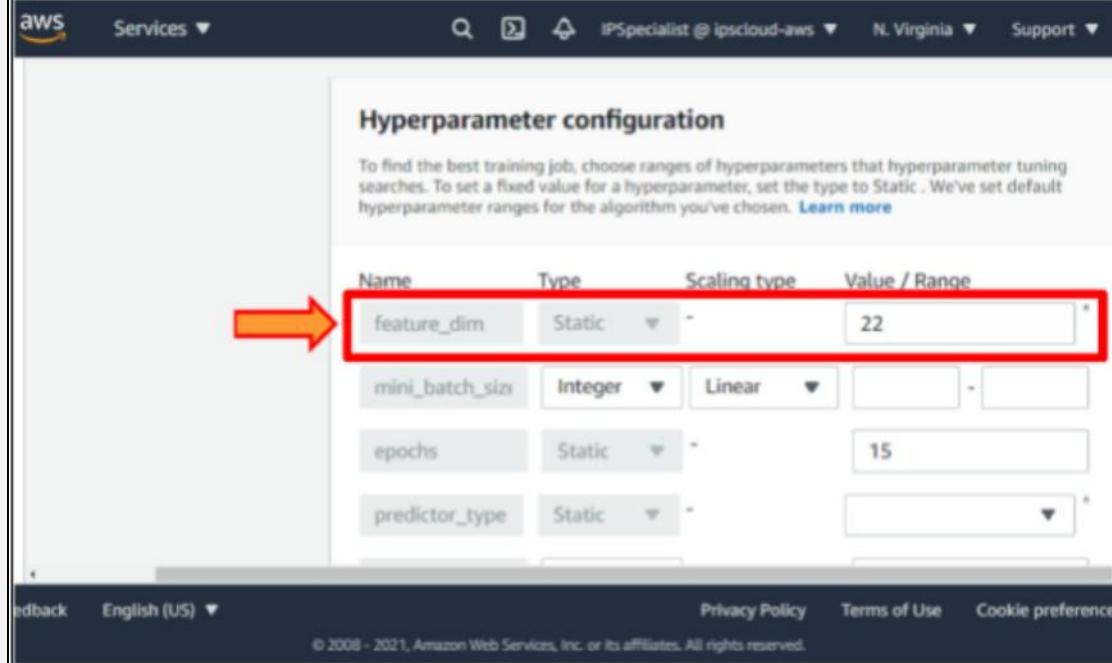
To find the best training job, choose ranges of hyperparameters that hyperparameter tuning searches. To set a fixed value for a hyperparameter, set the type to Static . We've set default hyperparameter ranges for the algorithm you've chosen. [Learn more](#)

Name	Type	Scaling type	Value / Range
feature_dim	Static		22
mini_batch_size	Integer	Linear	
epochs	Static		15
predictor_type	Static		

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



16. Type 500 – 5000 in the **mini_batch_size**

Sales Support

Services ▾

Hyperparameter configuration

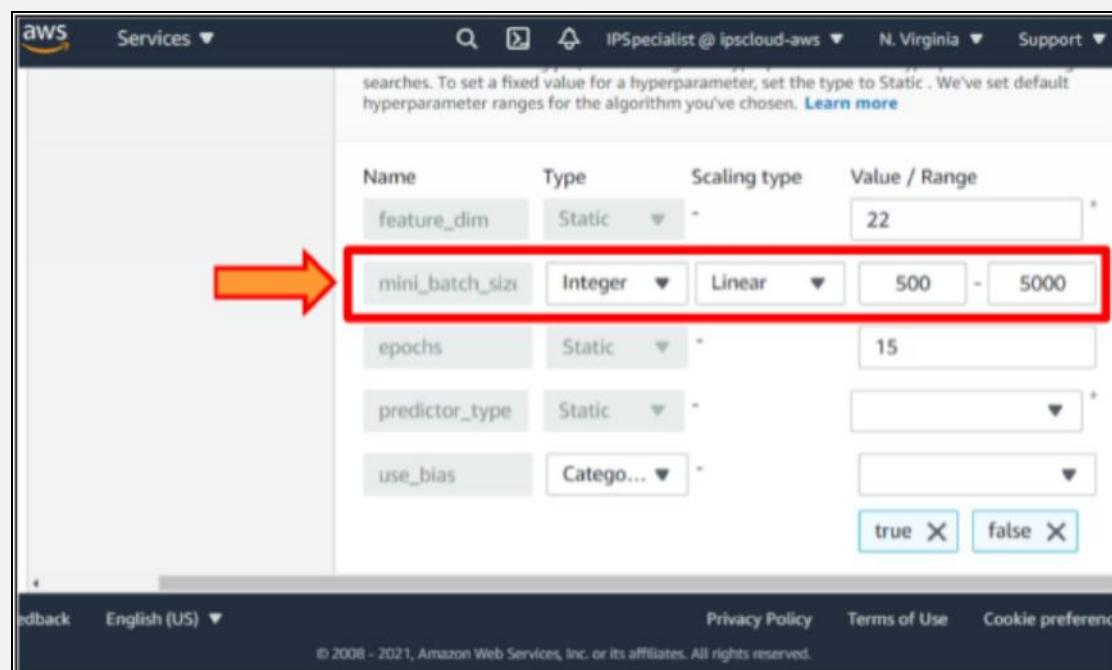
To find the best training job, choose ranges of hyperparameters that hyperparameter tuning searches. To set a fixed value for a hyperparameter, set the type to Static . We've set default hyperparameter ranges for the algorithm you've chosen. [Learn more](#)

Name	Type	Scaling type	Value / Range
feature_dim	Static		22
mini_batch_size	Integer	Linear	500 - 5000
epochs	Static		15
predictor_type	Static		
use_bias	Catego...		true X false X

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



17. Select the **multiclass_classifier** in the **predictor_type**.

aws Services Search IPSpecialist @ ipscloud-aws N. Virginia Support

searches. To set a fixed value for a hyperparameter, set the type to Static . We've set default hyperparameter ranges for the algorithm you've chosen. [Learn more](#)

Name	Type	Scaling type	Value / Range
feature_dim	Static	-	22
mini_batch_size	Integer	Linear	500 - 5000
epochs	Static	-	15
predictor_type	Static	-	multiclass_classifier
use_bias	Catego...	-	

Feedback English (US) Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



18. Scroll down. Type **.0001 – 1.0** in the **wd**.

aws Services Search IPSpecialist @ ipscloud-aws N. Virginia Support

loss	Static	-	auto
margin	Static	-	
quantile	Static	-	
loss_insensitivity	Static	-	
huber_delta	Static	-	
wd	Contin...	Logarithmic	.0001 - 1.0
l1	Contin...	Logarithmic	
momentum	Static	-	

Feedback English (US) Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



19. Type the **.0001 – 1.0** in the **L1**.

AWS Lambda function configuration page. The 'Hyperparameters' section shows the following parameters:

loss	Static	auto			
margin	Static				
quantile	Static				
loss_insensitivity	Static				
huber_delta	Static				
wd	Continuous	Logarithmic	.0001	-	1.0
l1	Continuous	Logarithmic	.0001	-	1.0
momentum	Static				

Feedback English (US) Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

20. Type the **.0001 – 1.0** in the **learning_rate**.

AWS Lambda function configuration page. The 'Hyperparameters' section shows the following parameters:

huber_delta	Static				
wd	Continuous	Logarithmic	.0001	-	1.0
l1	Continuous	Logarithmic	.0001	-	1.0
momentum	Static				
learning_rate	Continuous	Logarithmic	.0001	-	1.0
beta_1	Static				
beta_2	Static				
lambda_per_node	Constant				

Feedback English (US) Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

21. Scroll down. Type the 3 in the **num_classes**.

Screenshot of the AWS Lambda function configuration page for a Privacy-preserving Machine Learning job.

The configuration section shows various parameters:

- positive_examplar_label: Contin... Logarithmic
- early_stopping_criteria: Static 3
- early_stopping_epsilon: Static 0.001
- num_classes: Static 3 (highlighted with a red box and arrow)
- accuracy_top_k: Static
- f_beta: Static
- balance_multiclass_weighting: Static

Feedback: English (US) | Privacy Policy | Terms of Use | Cookie preferences | © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

22. Click on the **Next** button.

Screenshot of the AWS Lambda function configuration page for a Privacy-preserving Machine Learning job, showing the configuration step.

The configuration section shows the same parameters as the previous screenshot, with the "num_classes" field set to "Static 3".

At the bottom right, there are "Cancel" and "Next" buttons. The "Next" button is highlighted with a red box and arrow.

Feedback: English (US) | Privacy Policy | Terms of Use | Cookie preferences | © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

23. Select the **Pipe** input mode for the train channel.

Screenshot of the AWS SageMaker console showing the 'Create training job definition' step. The 'Channels' section is highlighted. A red arrow points to the 'Channel name' field containing 'train'. Another red arrow points to the 'Input mode - optional' dropdown menu showing 'Pipe'.

24. Go to the S3 dashboard.
25. Click on **ips-s3-bucket**.

Screenshot of the AWS S3 dashboard showing the 'Account snapshot' and 'Buckets' sections. The 'Buckets' section lists one bucket named 'ips-s3-bucket'. A red arrow points to this row, highlighting the creation date 'October 2, 2021, 04:10:07 (UTC-07:00)'.

26. Click on the **ufo_sightings_train_recordIO_protobuf.data** file.

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions ▾](#)

[Create folder](#) [Upload](#)

Find objects by prefix

< 1 > ⌂

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	ufo_sightings_train_recordIO_protobuf.data	data	October 7, 2021, 03:59:10 (UTC-07:00)	1.8 MB	Standard
<input type="checkbox"/>	ufo_sightings_validation_recordIO_protobuf.data	data	October 7, 2021, 03:59:11 (UTC-07:00)	228.6 KB	Standard

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

27. Copy the key name.

aws Services ▾ Search for services, features, marketplace proc [Alt+S] IPSpecialist @ ipscloud-aws Global ▾ Support

US East (N. Virginia) us-east-1

Last modified: October 7, 2021, 03:59:10 (UTC-07:00)

Size: 1.8 MB

Type: data

Key: [ufo_sightings_train_recordIO_protobuf.data](#)

Amazon Resource Name (ARN): [arn:aws:s3:::ips-s3-bucket/ufo_sightings_train_recordIO_protobuf.data](#)

Entity tag (Etag): [9426a555823e39123ec2666ece8d7c3d](#)

Object URL: https://ips-s3-bucket.s3.amazonaws.com/ufo_sightings_train_recordIO_protobuf.data

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

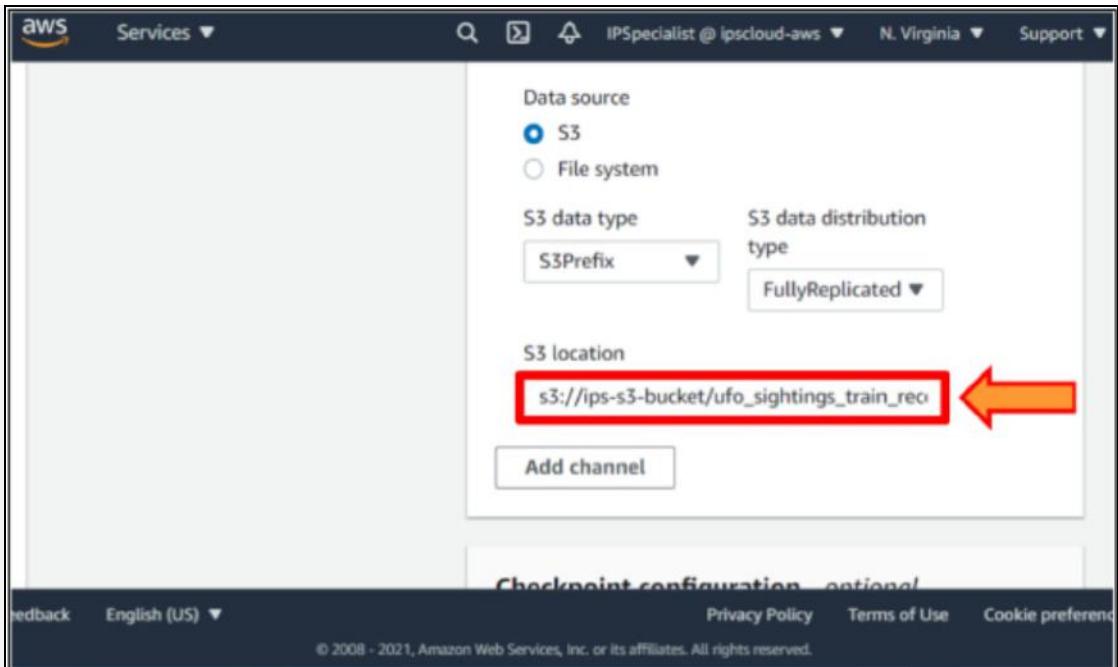
Bucket properties	Management configurations
Bucket Name: ips-s3-bucket	Replication status: Not configured

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

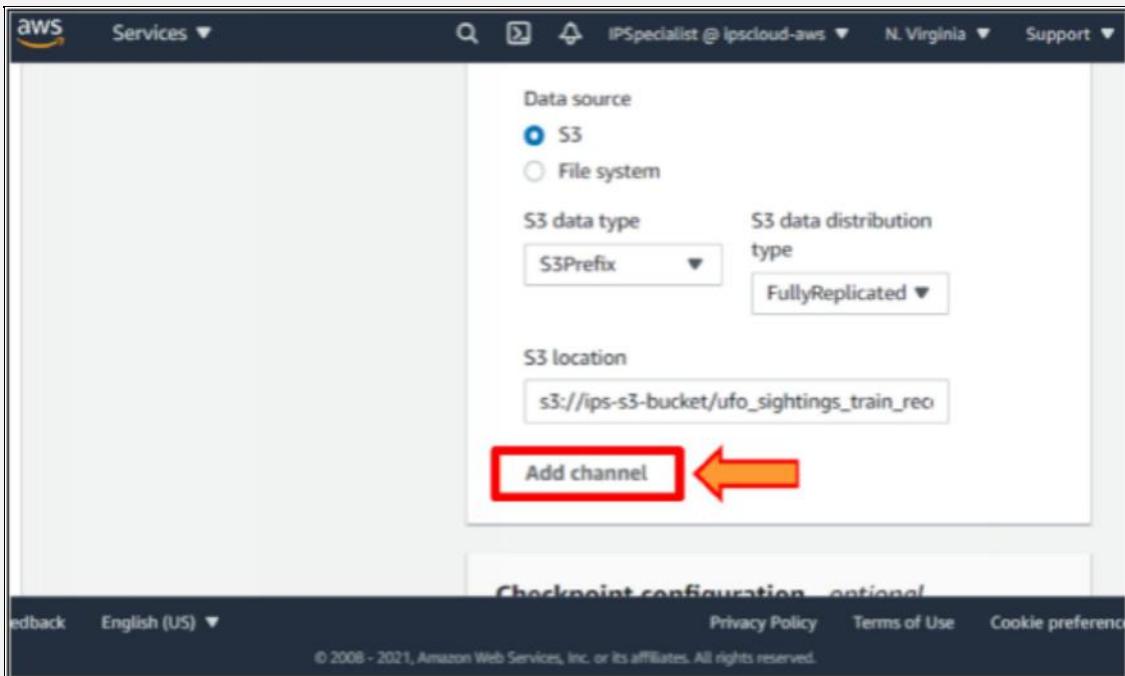
© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

28. Go back to the SageMaker dashboard.

29. Give the S3 location path of the training file in the following manner **s3://ips-s3—bucket/<Train_Key_Name>**.



30. Click on the **Add channel** button.



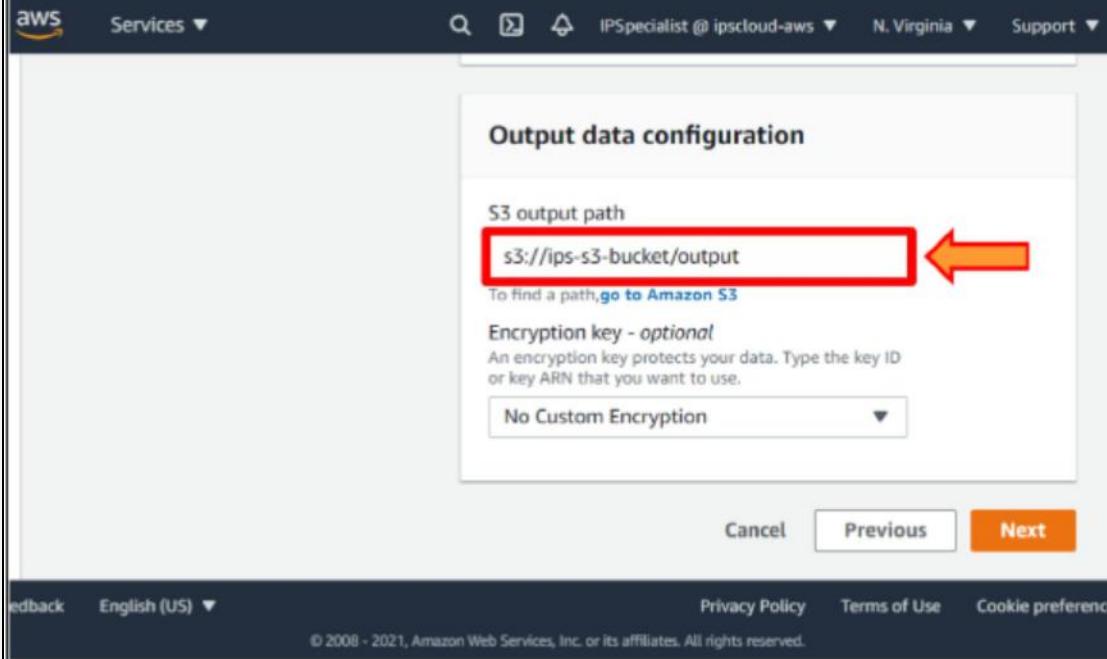
31. Give a channel name: **Validation**.
32. Select the **Pipe** input mode for the validation channel.

The screenshot shows the AWS Step Functions console. In the 'validation' section, there is a 'Channel name' input field containing 'validation' and an 'Input mode' dropdown set to 'Pipe'. Both of these fields are highlighted with red boxes and orange arrows pointing to them from the left and right respectively. The rest of the validation configuration screen is visible, including 'Content type - optional', 'Compression type', and 'Record wrapper'.

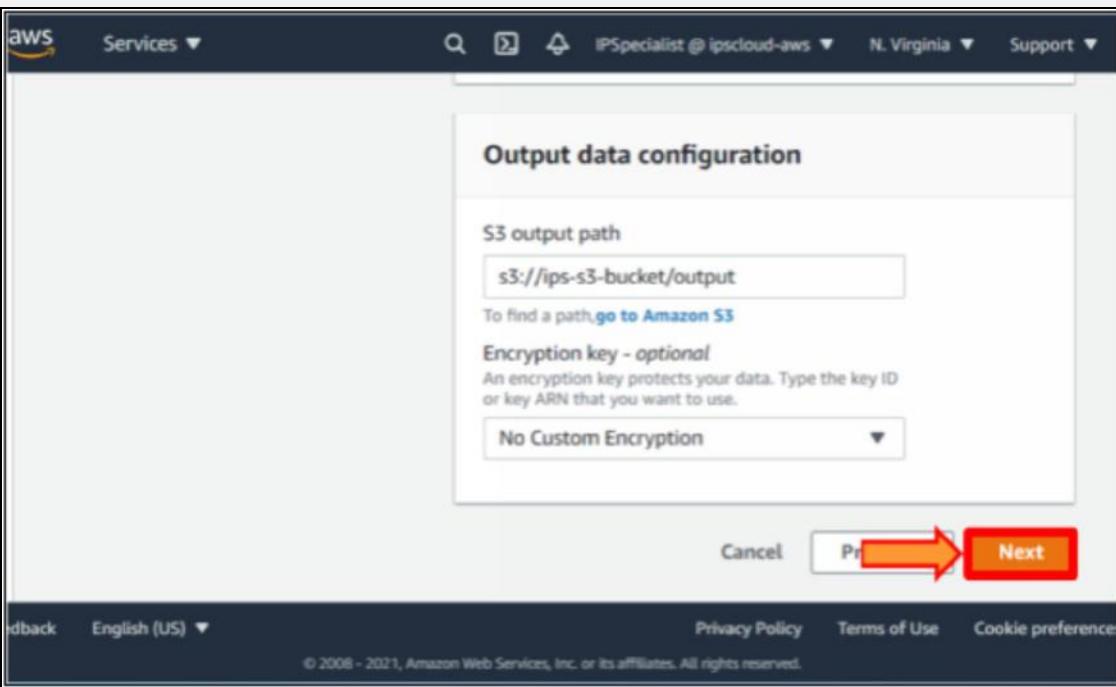
33. Give the S3 location path of the validation file in the following manner **s3://ips-s3-bucket/<Validation_Key_Name>**.

The screenshot shows the AWS Step Functions console. In the 'validation' section, the 'S3 location' input field contains 's3://ips-s3-bucket/ufo_sightings_validation' and is highlighted with a red box and an orange arrow pointing to it from the right. The rest of the validation configuration screen is visible, including 'Data source' (set to S3), 'S3 data type' (set to S3Prefix), 'S3 data distribution type' (set to FullyReplicated), and an 'Add channel' button.

34. Give the **S3 location path** of output data. **s3://ips-s3-bucket-output**.



35. Click on the **Next** button.



36. Select the **ml.m4.xlarge** instance type.

Sales Support

Services ▾

IPSpecialist @ ipscloud-aws N. Virginia Support

Resource configurations

The following resources will be applied to each training job.

Instance type: ml.m4.xlarge

Instance count: 1

Additional volume size per instance (GB): 1

Encryption key - optional: No Custom Encryption

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

37. Set 5 minutes in Maximum duration per training job.

At the end of this duration you will receive the complete or partial results of your managed spot training job.

48 hours

Stopping condition

Maximum duration per training job: 5 minutes

Maximum of 28 days

Cancel Previous

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

38. Click on the Next button.

Choose one or more algorithms you want to use for hyper-parameter tuning. [Learn more](#)

Start training job definition

Configure algorithm and parameters

Define data input and output

Configure resources

Configure tuning job sources

View and create

Name	Algorithm	Objective metric	Instance type	Instance count	Action
-	-	-	ml.m4.xlarge	1	Edit Clone Remove
IPS-linear-learner-tuning-job	Linear Learner	validation:objective_loss	ml.m4.xlarge	1	Edit Clone Remove

Cancel Next

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

39. Type **5** in the **Maximum Parallel Training Jobs** box.
40. Type **50** in the **Maximum Training Jobs** box.

The screenshot shows the 'Resource limits' section of the AWS Step Functions 'Create training job definition' wizard. It includes two input fields: 'Maximum parallel training jobs' set to 5 and 'Maximum training jobs' set to 50. Both fields have orange arrows pointing to them, indicating they are the focus of step 41. The page also shows navigation steps: Step 1 (Define job settings), Step 2 (Create training job definition), Step 3 (Configure tuning job resources), and Step 4 (Review and create). The status bar at the bottom indicates the page is on Step 3 of 4.

41. Click on the **Next** button.

The screenshot shows the same 'Resource limits' section as before, but now the 'Next' button at the bottom right is highlighted with an orange arrow, indicating it is the next action to take. The rest of the interface remains the same, showing the four-step navigation and the AWS branding.

42. Click on the **Create hyperparameter tuning job** button.

IPS-linear-learner-tuning-job

Linear Learner validation:objective_loss ml.m4.xlarge 1 Edit

Step 3: Tuning job resource configuration

Resource limits

Maximum number of parallel training jobs: 5 Maximum total number of training jobs: 50

Create hyperparameter tuning job

A successfully created job will take time up to 30 minutes to complete.

Amazon SageMaker X

Tuning job IPS-linear-learner-tuning-job was successfully created

Amazon SageMaker Studio

Dashboard

Search

Images

Ground Truth

Notebook

Processing

Training

Inference

Code Management

Add/Edit tags

Create hyperparameter tuning job

Search hyperparameter tuning jobs

Name Status Training completed/total Creation time Duration

IPS-linear-learner-tuning-job InProgress - Oct 07, 2021 11:31 UTC a few seconds

43. Click on the IPS-linear-learner-tuning-job

Amazon SageMaker > Hyperparameter tuning jobs

Hyperparameter tuning jobs

Add/Edit tags Create hyperparameter tuning job

Search hyperparameter tuning jobs

Name Status Training completed/total Creation time Duration

IPS-linear-learner-tuning-job InProgress 4 / 7 Oct 07, 2021 11:31 UTC 6 minutes

44. Check the Training job status counter.

The screenshot shows the AWS SageMaker Training Jobs page. At the top, there are tabs: Best training job, Training jobs (which is selected), Training job definitions, Tuning Job configuration, and Tags. Below the tabs is a red box labeled "Training job status counter" containing four status categories: Completed (green), In Progress (blue), Stopped (grey), and Failed (red). An orange arrow points to the "Completed" category. The main area below the status counter is titled "Training jobs" and includes a search bar and navigation controls.

45. Hence, the training is completed.
46. Click on **IPS-linear-learner-tuning-job**.

The screenshot shows the AWS SageMaker Hyperparameter tuning jobs page. At the top, there are tabs: Services (selected), Amazon SageMaker, and Hyperparameter tuning jobs. Below the tabs is a red box labeled "Hyperparameter tuning jobs" containing a table with columns: Name, Status, Training completed/total, Creation time, and Duration. A row for "IPS-linear-learner-tuning-job" is highlighted with a red box and an orange arrow pointing to it. The status is "Completed" with a green checkmark. The creation time is "Oct 07, 2021 11:31 UTC" and the duration is "an hour". The table also includes a "Create hyperparameter tuning job" button and a search bar.

47. Click on the **Best training job** tab.

The screenshot shows the AWS SageMaker Best training job summary page. At the top, there are tabs: Best training job (selected), Training job definitions, Tuning Job configuration, and Tags. A large orange arrow points to the "Best training job" tab. Below the tabs is a section titled "Best training job summary" which states "This training job is the best training job for only this hyperparameter tuning job." There is a "Create model" button. The bottom of the page includes standard AWS footer links: Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

48. Scroll down you will see the best training job hyperparameters.

The screenshot shows a table titled "Best training job hyperparameters". The columns are "Name", "Type", and "Value". The rows contain the following data:

Name	Type	Value
l1	-	0.038943769997401095
learning_rate	-	0.042844922253720365
mini_batch_size	-	500
use_bias	-	true
wd	-	0.0004309501638880474

Below the table, there is a footer with links: "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preferences". A copyright notice at the bottom states: "© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved."

49. Copy the name **IPS-linear-learner-tuning-job**.

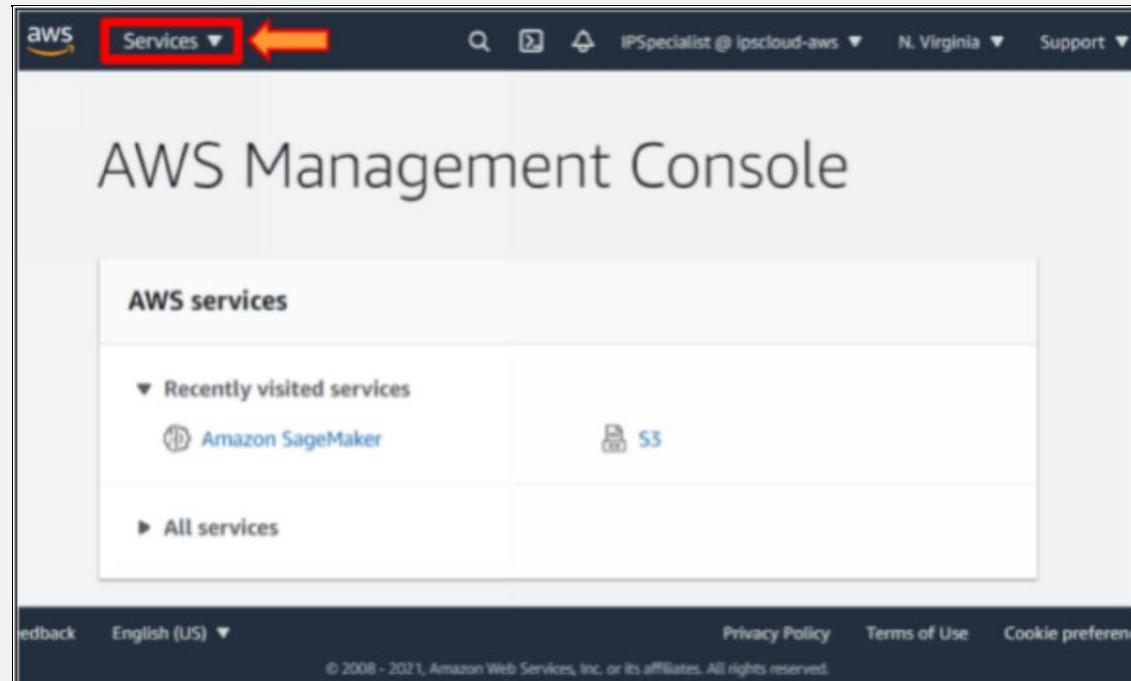
The screenshot shows a table titled "Best training job summary". The columns are "Name", "Status", "Objective metric", and "Value". The rows contain the following data:

Name	Status	Objective metric	Value
IPS-linear-learner-tuning-job-030-7d1fc387	Completed	validation:objective_loss	0.19607816636562347

Below the table, there is a section titled "Best training job hyperparameters". At the bottom, there is a footer with links: "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preferences".

Step 3: Analyze CloudWatch Training Logs

1. Click on the Services.



2. Select the CloudWatch from the Management & Governance.

The screenshot shows the AWS Services dashboard. On the left, there's a sidebar with 'Favorites' and 'Recently visited' sections. The main area is titled 'All services' and lists various AWS services. A red arrow points to the 'CloudWatch' service, which is highlighted with a red box. Other services listed include Elastic Kubernetes Service, Amazon Braket, Amazon Rekognition, Amazon Sumerian, Red Hat OpenShift Service on AWS, Management & Governance, Amazon Textract, Application Integration, AWS Organization, Amazon Transcribe, Step Functions, AWS S3, AWS Auto Scaling, DeepCompos, Amazon EventBridge, AWS FSx, CloudFormation, AWS DeepLens, and Amazon MQ.

3. Click on the Logs groups from the left-hand side.

The screenshot shows the CloudWatch Overview page. On the left, there's a sidebar with 'Logs' expanded, showing 'Log groups' and 'Logs Insights'. A red arrow points to the 'Log groups' link. The main area has two informational boxes: one about updating composite alarms and another about the new design for automatic dashboards. Below these is a 'CloudWatch: Overview' section with time range options (1h, 3h, 12h, 1d, 3d, 1w, custom). At the bottom, there are links for 'Feedback', 'English (US)', and 'Cookie preferences'.

4. Click on the /aws/sagemaker/TrainingJobs.

The screenshot shows the 'Log groups (9)' page. It features a search bar, filter options, and a table of log groups. One specific log group, '/aws/sagemaker/TrainingJobs', is highlighted with a red box and a red arrow pointing to it. The table columns include 'Log group', 'Retention', and 'Metric filters'.

5. Paste the name ips-linear-learner-tuning-job.

6. Click on ips-linear-learner-tuning-job.

The screenshot shows the AWS CloudWatch Log Stream interface. At the top, there are tabs for 'Log streams', 'Metric filters', 'Subscription filters', 'Contributor Insights', and 'Tags'. Below the tabs, a search bar contains the query 'IPS-linear-learner-tuning-job-030-7d1fc387'. A red arrow points to this search bar. The results section shows 'Log streams (50)' and a table with columns for 'Log stream' and 'Last event time'. One row in the table is highlighted with a red box and a red arrow pointing to it. This row corresponds to the log entry shown below the table. The log entry details the validation score for the 'multiclass_accuracy' metric.

Log stream	Last event time
IPS-linear-learner-tuning-job-030-7...	2021-10-07 05:04:33 (UTC-07:00)

7. You will see the accuracy and other metrics.

The screenshot shows the AWS CloudWatch Log Stream interface displaying log entries. A red box highlights a specific log entry, and a red arrow points to the 'accuracy' metric within that entry. The log entry shows the validation score for the 'multiclass_accuracy' metric, which is 0.9452904681331077.

Time	Message
2021-10-07T05:04:31.339-07:00	[10/07/2021 12:04:30 INFO 140363874973504] #progress...
2021-10-07T05:04:31.339-07:00	#metrics {"StartTime": 1633608268.807434, "EndTime": ...
2021-10-07T05:04:31.339-07:00	[10/07/2021 12:04:30 INFO 140363874973504] #through...
2021-10-07T05:04:31.339-07:00	[10/07/2021 12:04:30 WARNING 140363874973504] wait_...
2021-10-07T05:04:31.339-07:00	[10/07/2021 12:04:30 WARNING 140363874973504] wait_...
2021-10-07T05:04:32.339-07:00	[2021-10-07 12:04:31.894] [tensorio] [info] epoch_s...
2021-10-07T05:04:32.339-07:00	[10/07/2021 12:04:31 INFO 140363874973504] #early_s...
2021-10-07T05:04:33.339-07:00	[2021-10-07 12:04:32.971] [tensorio] [info] epoch_s...
2021-10-07T05:04:33.339-07:00	[10/07/2021 12:04:32 INFO 140363874973504] #validat...
2021-10-07T05:04:33.340-07:00	[10/07/2021 12:04:32 INFO 140363874973504] #validat...
	[10/07/2021 12:04:32 INFO 140363874973504] #validation_score (algo-1) : ('multiclass_accuracy', 0.9452904681331077)
2021-10-07T05:04:33.340-07:00	[10/07/2021 12:04:32 INFO 140363874973504] #validat...
2021-10-07T05:04:33.340-07:00	[10/07/2021 12:04:32 INFO 140363874973504] #validat...
2021-10-07T05:04:33.340-07:00	[10/07/2021 12:04:32 INFO 140363874973504] #validat...

Step 4: Create SageMaker Notebook

1. Click on Services.

AWS Management Console

AWS services

Recently visited services

Your recently visited AWS services appear here.

All services

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Select SageMaker from Machine Learning.

Services ▾

Favorites

Add favorites by clicking on the star next to the service name.

Recently visited

Console Home

All services

Compute	Customer Enablement	Machine Learning	AWS Cost Management
EC2	AWS IQ	Amazon SageMaker	AWS Cost explorer
Lightsail	Support	Amazon Augmented AI	AWS Budgets
Lambda	Managed Services	Amazon CodeGuru	AWS Marketplace
Batch	Activate for Startups	Amazon DevOps Guru	Subscriptions
Elastic Beanstalk	Robotics	Amazon Comprehend	AWS Application Cost Profiler
Serverless Application Repository	AWS RoboMaker	Front-end	
AWS Outposts			

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

3. Click on Notebook Instance on the left-hand side menu.

The screenshot shows the Amazon SageMaker Studio interface. On the left, there is a navigation sidebar with the following menu items:

- Dashboard
- Search
- Images
- ▶ Ground Truth
- ▼ Notebook
 - Notebook instances
- Lifecycle configurations
- Git repositories
- ▶ Processing
- ▶ Training

The "Notebook instances" item is highlighted with a red box and has an orange arrow pointing to it from the left.

The main content area is titled "MACHINE LEARNING" and features a large heading "Amazon SageMaker" followed by the subtext "Build, train, and deploy machine learning models at scale". Below this, a subtext reads "The quickest and easiest way to get ML models from idea to production." At the bottom of the main area, there are links for "Privacy Policy", "Terms of Use", and "Cookie preferences".

4. Click on the **Create notebook instance** button.

The screenshot shows the "Notebook instances" page within the Amazon SageMaker service. The top navigation bar includes the AWS logo, "Services", and user information. The main content area displays the "Notebook instances" heading and a search bar. To the right of the search bar is a "Create notebook instance" button, which is highlighted with a red box and an orange arrow pointing to it from the left.

The table below the heading shows columns for "Name", "Instance", "Creation time", "Status", and "Actions". A message at the bottom of the table states "There are currently no resources."

At the bottom of the page, there are links for "Feedback", "English (US)", "Privacy Policy", "Terms of Use", and "Cookie preference".

5. Give the name **ips-notebook-instance**.



Services ▾



IPSpecialist @ ipscloud-aws ▾

N. Virginia ▾

Support ▾

[Amazon SageMaker](#) > [Notebook instances](#) > Create notebook instance

Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. [Learn more](#)

Notebook instance settings

Notebook instance name

ips-notebook-instance

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type

Feedback English (US) ▾

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Select the **ml.c4.medium** Notebook instance type.

The screenshot shows the 'Create notebook instance' page. The 'Notebook instance type' dropdown is open, showing 'ml.c4.xlarge' as the selected option, which is highlighted with a red box and an orange arrow pointing to it. Below the dropdown, there is a note: 'Amazon SageMaker Notebook Instance is ending its standard support on Amazon Linux AMI (AL1). Learn more'. Further down, there is a 'Platform identifier' field containing 'notebook-al1-v1' and a 'Additional configuration' section.

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

7. Scroll down under permission and encryption.

8. Click on **Create a new role**.

Permissions and encryption

IAM role
Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the [AmazonSageMakerFullAccess](#) IAM policy attached.

AmazonSageMakerServiceCatalogProductsUserRole

Create a new role

Enter a custom IAM role ARN

Use existing role

AmazonSageMaker-ExecutionRole-20211001T004591

AmazonSageMaker-ExecutionRole-20211001T004641

No custom encryption

Feedback English (US) Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Select Any S3 Bucket.

10. Click on the **Create role** button.

Create an IAM role

Passing an IAM role gives Amazon SageMaker permission to perform actions in other AWS services on your behalf. Creating a role here will grant permissions described by the [AmazonSageMakerFullAccess](#) IAM policy to the role you create.

The IAM role you create will provide access to:

S3 buckets you specify - optional

Any S3 bucket
Allow users that have access to your notebook instance access to any bucket and its contents in your account.

Specific S3 buckets

Example: bucket-name-1, bucke

Comma delimited. ARNs, ":" and "/" are not supported.

Minna

Create role

Hence, you have successfully created an IAM role.

Permissions and encryption

IAM role
Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the [AmazonSageMakerFullAccess](#) IAM policy attached.

AmazonSageMaker-ExecutionRole-20211002T041831

Success! You created an IAM role.
AmazonSageMaker-ExecutionRole-20211002T041831

Root access - optional

Enable - Give users root access to the notebook

Disable - Don't give users root access to the notebook
Lifecycle configurations always have root access

Encryption key - optional

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11. Scroll down. Click on the **Create notebook instance** button. It will take a few minutes.

No Custom Encryption

▶ Network - optional

▶ Git repositories - optional

▶ Tags - optional

Create notebook instance

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

12. Click on **ips-notebook-instance**.

Amazon SageMaker > Notebook instances

Notebook instances

Name	Instance	Creation time	Status	Actions
ips-notebook-instance	ml.c4.xlarge	Oct 07, 2021 13:57 UTC	InService	Open Jupyter Open JupyterLab

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preference

13. Click on the **Open Jupyter** button.

Amazon SageMaker > Notebook instances > ips-notebook-instance

ips-notebook-instance

Delete Open Jupyter Open JupyterLab

Notebook instance settings Edit

Name: ips-notebook-instance

ARN: arn:aws:sagemaker:us-east-1:644779277407:notebook-instance/ips-notebook-instance

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preference

14. Download the jupyter notebook from the following Github link:
https://github.com/ACloudGuru-Resources/Course_AWS_Certified_Machine_Learning/blob/master/Chapter8/ufo-evaluation-optimization-lab.ipynb

UFO Sightings Evaluation and Optimization Lab

The goal of this notebook is to find out if our optimized model hyperparameters out performs the training of our baseline Linear Learner model. We can also compare things like accuracy and see if they differ.

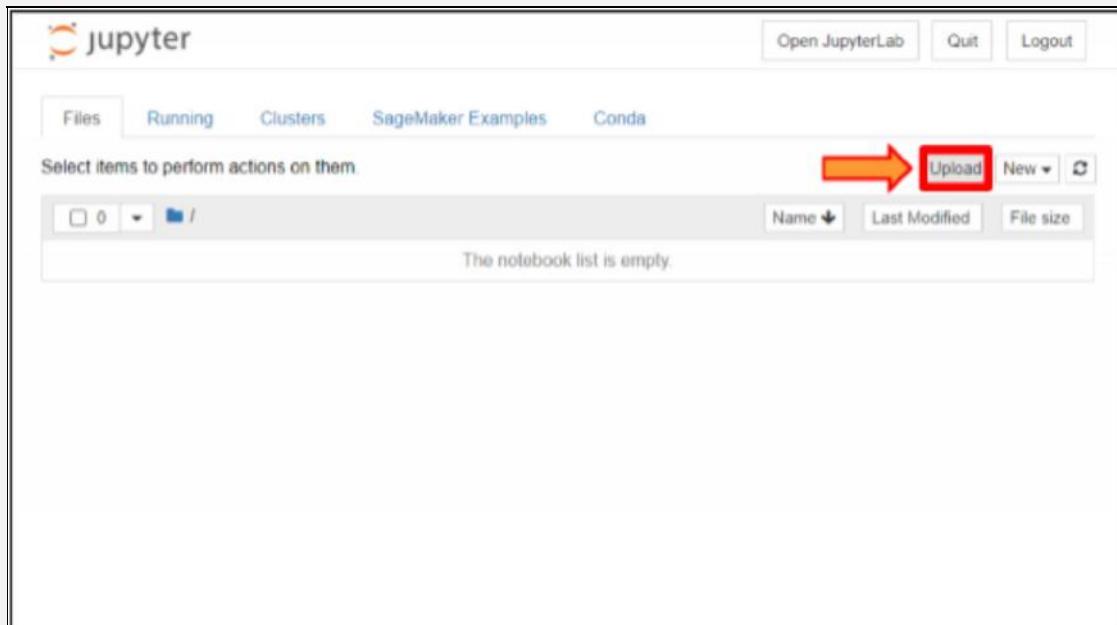
What we plan on accomplishing is the following:

1. Create and train our "optimized" model (Linear Learner)
2. Compare the results!

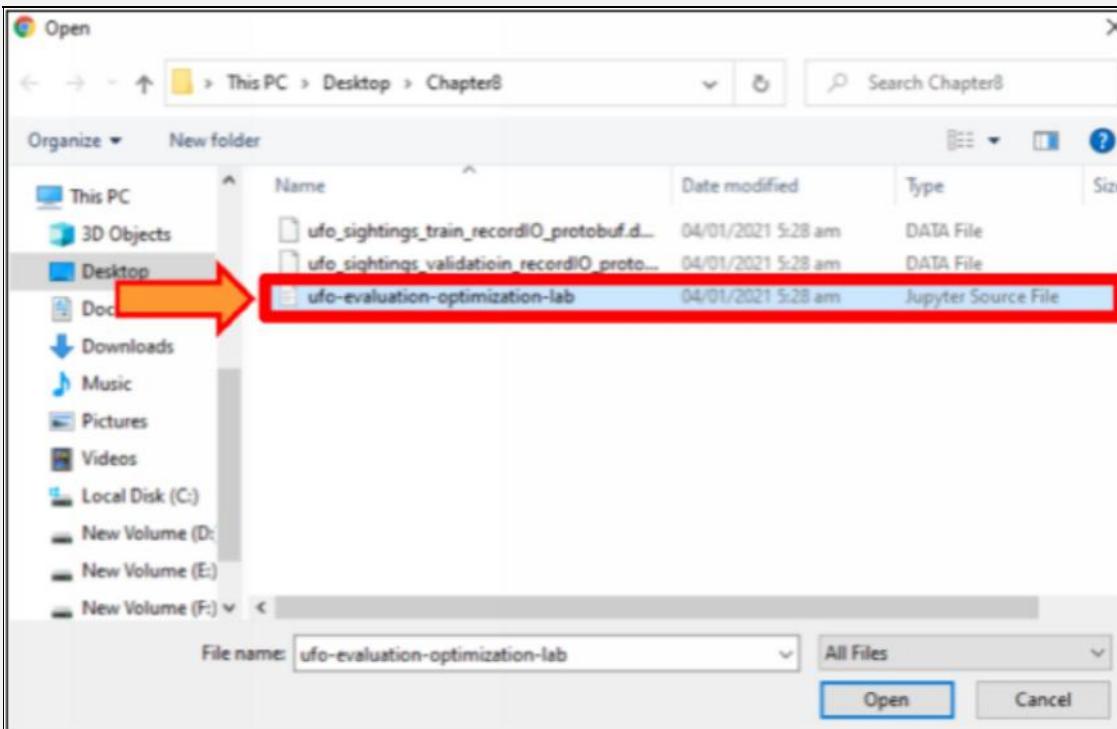
First let's go ahead and import all the needed libraries.

```
In [ ]: import pandas as pd  
import numpy as np  
from datetime import datetime  
  
import boto3  
from sagemaker import get_execution_role  
import sagemaker  
  
In [ ]: role = get_execution_role()  
bucket='<INSERT_BUCKET_NAME_HERE>'
```

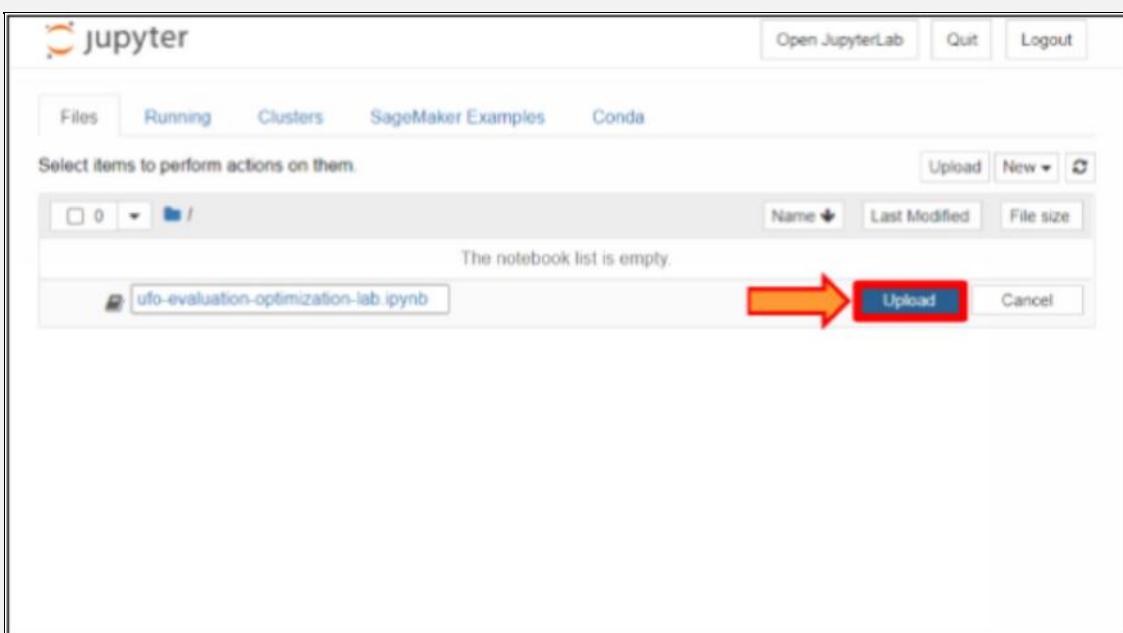
15. Click on the Upload button.



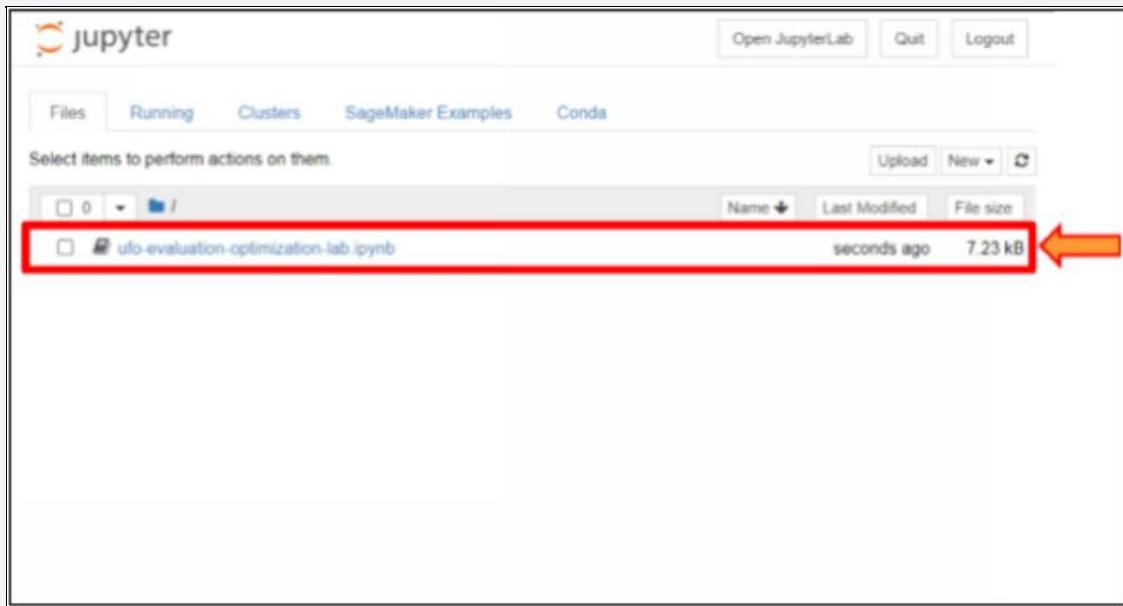
16. Select the **ufo-evaluation-optimization-lab.ipynb** jupyter notebook file.



17. Click on the **Upload** button.



18. Click on the **ufo-evaluation-optimization-lab.ipynb** jupyter notebook.



19. Run the cell to import python libraries. Click on the **Run** button or press **Shift + Enter** to execute the cell.

A screenshot of a Jupyter Notebook interface. The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and conda_python3. Below the menu is a toolbar with icons for file operations like New, Open, Save, and Run. A red arrow points to the 'Run' button. The main area shows two code cells. The first cell, labeled 'In [1]', contains Python code to import pandas, numpy, datetime, boto3, sagemaker, and get_execution_role. The second cell, labeled 'In []', contains code to set 'role = get_execution_role()' and 'bucket = '<INSERT_BUCKET_NAME_HERE>'. A red box highlights the first cell, and another red arrow points to the right edge of the notebook.

```
In [1]: import pandas as pd  
import numpy as np  
from datetime import datetime  
  
import boto3  
from sagemaker import get_execution_role  
import sagemaker  
  
In [ ]: role = get_execution_role()  
bucket='<INSERT_BUCKET_NAME_HERE>'
```

20. Insert the bucket name **ips-s3-bucket**. Then press **Shift + Enter** to execute the cell.

A screenshot of a Jupyter Notebook interface. The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and conda_python3. Below the menu is a toolbar with icons for file operations like New, Open, Save, and Run. A red arrow points to the 'Run' button. The main area shows two code cells. The first cell, labeled 'In [1]', contains Python code to import boto3, sagemaker, and get_execution_role. The second cell, labeled 'In []', contains code to set 'role = get_execution_role()' and 'bucket = 'ips-s3-bucket''. A red box highlights the second cell, and another red arrow points to the right edge of the notebook. Below the notebook, a section titled '1. Create and train our "optimized" model (Linear Learner)' is visible.

```
In [1]: import boto3  
from sagemaker import get_execution_role  
import sagemaker  
  
In [ ]: role = get_execution_role()  
bucket='ips-s3-bucket'
```

1. Create and train our "optimized" model (Linear Learner)

Let's evaluate the Linear Learner algorithm with the new optimized hyperparameters. Let's go ahead and get the data that we already stored into S3 as recordIO protobuf data.

21. Run the cell to get the train data stored in S3. Press **Shift + Enter** to execute the cell.

A screenshot of a Jupyter Notebook interface. The top menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and conda_python3. Below the menu is a toolbar with icons for file operations like New, Open, Save, and Run. A red arrow points to the 'Run' button. The main area shows two code cells. The first cell, labeled 'In [1]', contains a section titled '1. Create and train our "optimized" model (Linear Learner)'. It describes evaluating the Linear Learner algorithm with optimized hyperparameters and getting data from S3 as recordIO protobuf data. The second cell, labeled 'In [2]', contains Python code to define 'train_file' as 'ufo_sightings_train_recordIO_protobuf.data', set 'training_recordIO_protobuf_location' to 's3://{}//{}'.format(bucket, train_file), and print the training data location. A red box highlights the second cell, and another red arrow points to the right edge of the notebook.

```
1. Create and train our "optimized" model (Linear Learner)  
  
Let's evaluate the Linear Learner algorithm with the new optimized hyperparameters. Let's go ahead and get the data that we already stored into S3 as recordIO protobuf data.  
  
Let's get the recordIO file for the training data that is in S3  
  
In [2]: train_file = 'ufo_sightings_train_recordIO_protobuf.data'  
training_recordIO_protobuf_location = 's3://{}//{}'.format(bucket, train_file)  
print('The Pipe mode recordIO protobuf training data: {}'.format(training_recordIO_protobuf_location))  
  
The Pipe mode recordIO protobuf training data: s3://ips-s3-bucket/ufo_sightings_train_recordIO_protobuf.data
```

22. Run the cell to get the validation data stored in S3.



The screenshot shows a Jupyter Notebook interface with the title "jupyter ufo-evaluation-optimization-lab Last Checkpoint: 6 minutes ago (unsaved changes)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and a dropdown for "conda_python3". Below the toolbar is a toolbar with icons for file operations like Open, Save, and Run, and a dropdown for nbdl. The current cell is titled "ufodata". The code cell contains the following Python code:

```
In [4]: validation_file = 'ufo_sightings_validationin_recordIO.protobuf.data'  
validate_recordIO_protobuf_location = 's3://{}//{}',format(bucket, validation_file)  
print('The Pipe mode recordIO protobuf validation data: {}'.format(validate_recordIO_protobuf_location))
```

The output of the code is:

```
The Pipe mode recordIO protobuf validation data: s3://ips-s1-bucket/ufo_sightings_validationin_recordIO.protobuf.data
```

An orange arrow points to the code cell.

23. Run the cell to call the Linear Learner algorithm.

Alright we are good to go for the Linear Learner algorithm. Let's get everything we need from the ECR repository to call the Linear Learner algorithm.

```
In [5]: from sagemaker import ImageUri  
container = ImageUri.retrieve('linear-learner', boto3.Session().region_name, '1')
```

24. Run the cell to create a job and use the optimized hyperparameters.

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter ufo-evaluation-optimization-lab Last Checkpoint: 8 minutes ago (unsaved changes) Logout
- Toolbar:** File Edit View Insert Cell Kernel Widgets Help Trusted nbviewer nbdiff
- Code Cell [5]:** from sagemaker import image_uris
container = image_uris.retrieve('linear-learner', boto3.Session().region_name, '1')
- Text Cell:** Let's create a job and use the optimized hyperparameters.
- Code Cell [6] (highlighted with a red box and arrow):**

```
# Create a training job name
job_name = 'ufo-linear-learner-job-optimized-{}'.format(datetime.now().strftime("%Y%m%d%H%M%S"))
print('Here is the job name {}'.format(job_name))

# Here is where the model-artifact will be stored
output_location = 's3://{}{}/output'.format(bucket)

Here is the job name ufo-linear-learner-job-optimized-20211007141913
```

25. Insert the hyperparameters in the red box below.

```
In [ ]: %%time
sess = sagemaker.Session()

# Setup the LinearLearner algorithm from the ECR container
linear = sagemaker.estimator.Estimator(container,
                                         role,
                                         instance_count=1,
                                         instance_type='ml.c4.xlarge',
                                         output_path=output_location,
                                         sagemaker_session=sess,
                                         input_mode='Pipe')

# Setup the hyperparameters
linear.set_hyperparameters( feature_dim=22,
                            predictor_type='multiclass_classifier',
                            num_classes=3,
                            #%% enter optimized hyperparameters here
                            #%% enter optimized hyperparameters here)
)

# Launch a training job. This method calls the CreateTrainingJob API call.
data_channels = {
    'train': train_data.s3_data(),
    'validation': validation_data.s3_data()
}
```

26. Insert the below hyperparameters.

early_stopping_patience=3

`early_stopping_tolerance=0.001`

epochs=15

```
l1=0.0647741539306635
```

```
learning_rate=0.09329042024421902
```

```
loss='auto'
```

```
mini_batch_size=744
```

```
normalize_data='true'
```

```
normalize_label='auto'
```

```
num_models='auto'
```

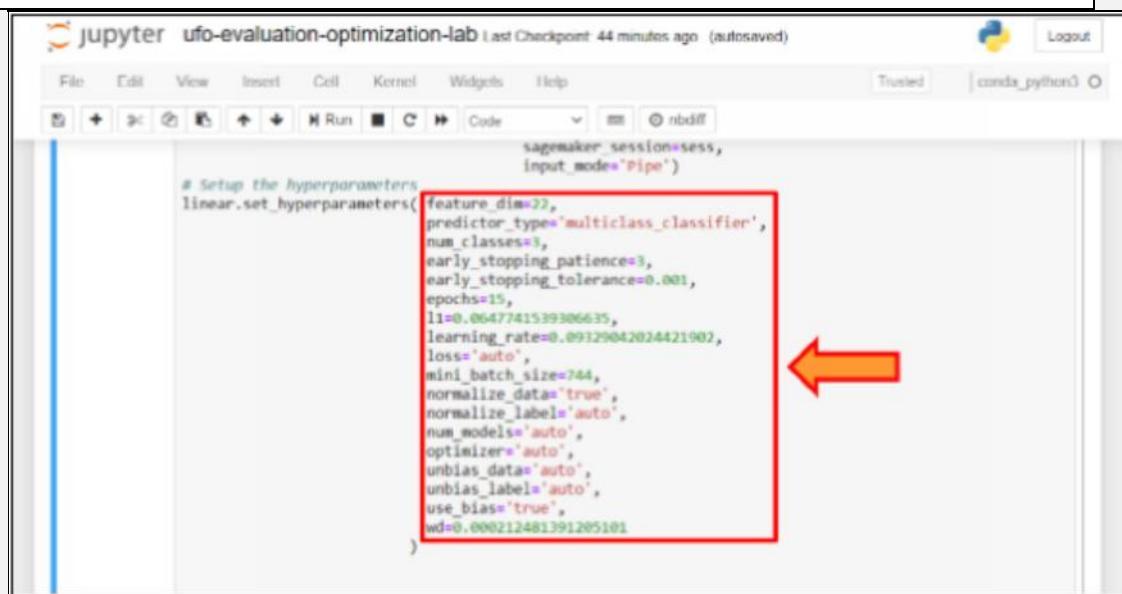
```
optimizer='auto'
```

```
unbias_data='auto'
```

```
unbias_label='auto'
```

```
use_bias='true'
```

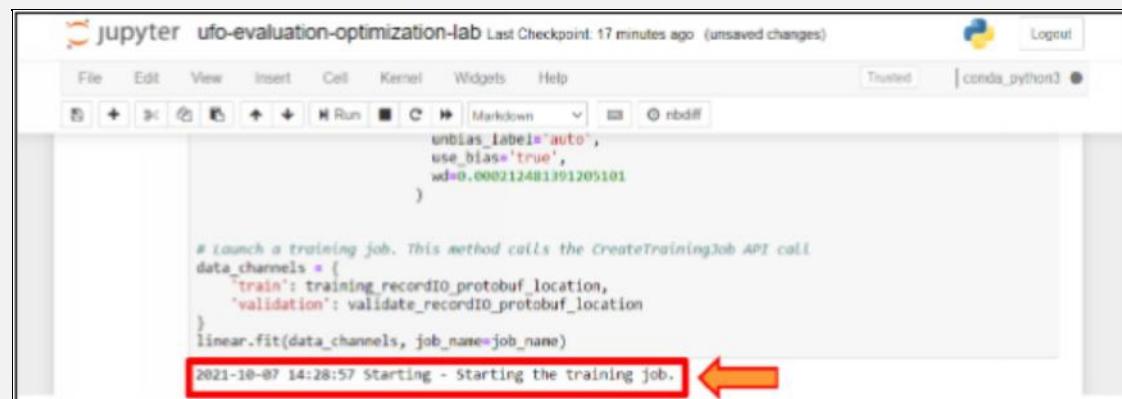
```
wd=0.000212481391205101
```



```
# Setup the hyperparameters
linear.set_hyperparameters(
    feature_dim=22,
    predictor_type='multiclass_classifier',
    num_classes=3,
    early_stopping_patience=3,
    early_stopping_tolerance=0.001,
    epochs=15,
    l1=0.0647741539306635,
    learning_rate=0.09329042024421902,
    loss='auto',
    mini_batch_size=744,
    normalize_data='true',
    normalize_label='auto',
    num_models='auto',
    optimizer='auto',
    unbiased_data='auto',
    unbiased_label='auto',
    use_bias='true',
    wd=0.000212481391205101
)
```

27. Run the cell to start the training job.

28. The training has started. It will take a few minutes to complete.



```
unbias_label='auto',
use_bias='true',
wd=0.000212481391205101
)

# Launch a training job. This method calls the CreateTrainingJob API call.
data_channels = [
    'train': training_recordIO_protobuf_location,
    'validation': validate_recordIO_protobuf_location
]
linear.fit(data_channels, job_name=job_name)
2021-10-07 14:28:57 Starting - Starting the training job.
```

29. Now, the training is completed, and you will see the training accuracy logs.

```

linear.fit(data_channels, job_name=job_name)

v store is not running distributed
[2021-10-07 14:47:18.317] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/validation", "epoch": 47, "duration": 1071, "num_examples": 3, "num_bytes": 234036}
[10/07/2021 14:47:18 INFO 140502297552704] #early_stopping_criteria_metric: host=algo-1, epoch=14, criteria=multiclass_cross_entropy_objective, value=0.19614590915393776
[2021-10-07 14:47:19.373] [tensorio] [info] epoch_stats={"data_pipeline": "/opt/ml/input/data/validation", "epoch": 49, "duration": 1043, "num_examples": 3, "num_bytes": 234036}
[10/07/2021 14:47:19 INFO 140502297552704] #validation_score (algo-1) : ('multiclass_cross_entropy_o
[10/07/2021 14:47:19 INFO 140502297552704] #validation_score (algo-1) : ('multiclass_accuracy', 0.9405440319255499)
[10/07/2021 14:47:19 INFO 140502297552704] #validation_score (algo-1) : ('multiclass_top_k_accuracy_3', 1.0)
[10/07/2021 14:47:19 INFO 140502297552704] #validation_score (algo-1) : ('dcg', 0.9779487863746122)
[10/07/2021 14:47:19 INFO 140502297552704] #validation_score (algo-1) : ('macro_recall', 0.9259293)
[10/07/2021 14:47:19 INFO 140502297552704] #validation_score (algo-1) : ('macro_precision', 0.900500
7)
[10/07/2021 14:47:19 INFO 140502297552704] #validation_score (algo-1) : ('macro_f_1.000', 0.9124998
5)
[10/07/2021 14:47:19 INFO 140502297552704] #quality_metric: host=algo-1, validation_multiclass_cross

```

Now we can compare the amount of time billed and the accuracy compared to our baseline model.

30. After completing the lab, delete all the AWS services used in this lab so you would not get charged.

Lab 15: Categorizing Uploaded Data Using AWS Step Functions

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	Lambda, S3, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

Amazon Identity Access Management (IAM)

Individuals and groups can be granted secure access to your AWS resources by using IAM. It allows you to create and manage IAM users and provide them access to your resources. Additionally, you have the option of granting access to users outside of AWS (federated users).

- **Managed Policy:** This contains the permission required to stop an EC2 instance
- **Inline Policy:** This allows this role to be passed to another service
- **Trust Policy:** This allows the System Manager and EC2 to assume the role. It enables EC2 to register with the Systems Manager and Systems Manager to stop the EC2 instance

Amazon Step Function

AWS Step Functions is a fully managed solution that makes it simple to coordinate the components of distributed applications and microservices using visual workflows. You may easily grow and alter applications by creating them from separate components that execute a specific function. Step Operations is a dependable method for coordinating components and stepping through your application's functions. Step Functions provides a graphical interface for organizing and visualizing your application's components as a series of stages. It makes creating and running multi-step apps a breeze. Step Functions automates the triggering and tracking of each step and retries when problems occur, ensuring that your application runs smoothly. Step Functions records the condition of each step so that you can immediately detect and troubleshoot problems if something goes wrong. You can alter and add stages without writing code, allowing you to quickly grow and innovate your application.

Amazon Simple Storage Service (S3)

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 provides a straightforward web service interface for storing and retrieving any amount of data at any time and from any location. You may quickly create projects that integrate cloud-native storage using this service. Because Amazon S3 is easily customizable and you only pay for what you use, you can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. Build a simple FTP system or a complex web application like the amazon.com retail website; read the same piece of data a million times or only for emergency disaster recovery; store whatever type and amount of data you desire.

Amazon Lambda Function

AWS Lambda allows you to run code without creating or managing servers. There is no charge when your code is not executing; you only pay for the compute time you use. You can run code for nearly any application or backend service with Lambda, and you do not have to worry about administration. Upload your code, and Lambda will handle everything necessary to run and grow it with high availability. You may configure your code to be automatically triggered by other AWS services, or you can access it directly from any computer or smartphone app.

Amazon Transcribe

Conduct a (text-based) content analysis on audio/video content using Automatic Speech Recognition (ASR) technology. Amazon Transcribe is an AWS service that makes converting speech to text simple for customers. Customers may utilize Amazon Transcribe for various commercial applications, such as voice-based customer care call transcription and the production of subtitles for audio/video material.

Case Study Enterprise Internet Television Network – Netflix

Background

Netflix is the world's leading internet television network, with over 200 million members in more than 190 countries enjoying 125 million hours of TV shows and movies daily. Netflix uses AWS for nearly all its computing and storage needs, including databases, analytics, recommendation engines, video

transcoding, and more—hundreds of functions that in total use more than 100,000 server instances on AWS.

Business Challenge

You are a Machine Learning engineer at Netflix. Your firm has just begun recording meetings to increase openness across the board. Your organization creates a lot of audio daily because of the number of meetings they have. It is wonderful to have these recordings, but being able to search through them as text would be far more beneficial since it would save a lot of time listening back over the material. This appears to be a machine learning job! One problem is that discussions regarding secret projects are taped because every meeting is taped. For better access control, those should be relocated to the side.

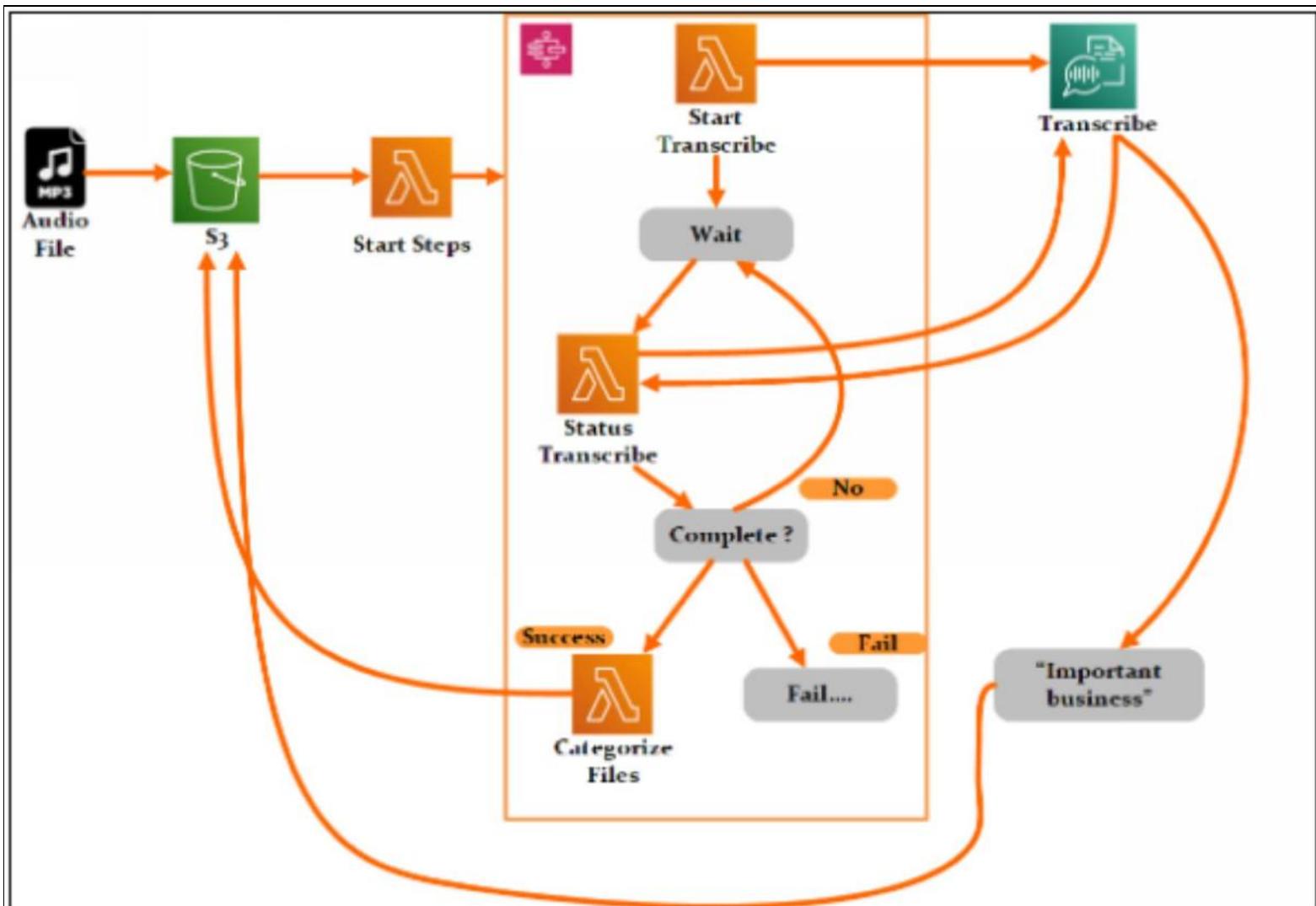
Your company asks you to extract the key features from the customer calls. Your goal is to create a pipeline to process all this audio and convert it to text. The transcript must be checked to determine whether any secret project names appear. Audio containing secret projects should be placed in the "important" folder rather than the normal "processed" folder.

Proposed Solution

The solution is to use AWS services to automate this task. You use AWS Transcribe, which converts real-time speech into text. Then you use Lambda functions as a helping service. After that, you use the Step Function to create a model. To store sensitive data, you use S3.

Before starting the lab, create an S3 bucket and lambda functions mentioned in previous labs.

Lab Diagram



Implementation Steps

1. Create an IAM Role.
2. Create Step Function.
3. Update Lambda to Work with Step Functions.
4. Update S3 to Work with Step Functions.
5. Create Step Function Flow.
6. Create the Lambda Business Logic.
7. Categorize Audio Data.

Solution

Step 1: Create an IAM Role

1. Log in to the **AWS Console**.
2. Click on **Services**.

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with the AWS logo, a 'Services' dropdown menu (which is highlighted with a red box and has a yellow arrow pointing to it), a search icon, a refresh icon, the user name 'cloud_user @ 5128-3147-4478', a region selector 'N. Virginia', and a 'Support' link. Below the navigation bar, the title 'AWS Management Console' is displayed. Underneath the title, there is a section titled 'AWS services' with a heading 'Recently visited services' and a note: 'Your recently visited AWS services appear here.' There is also a link to 'All services'. At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with a copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

3. Select **IAM** from **Security, Identity & Compliance**.

The screenshot shows the 'All services' list in the AWS Management Console. On the left, there is a sidebar with sections for 'Favorites' (including 'Resources') and 'Recently visited' (including 'Console Home'). The main area lists various AWS services under categories. One category, 'Security, Identity, & Compliance', is expanded, showing sub-services like 'IAM' (which is highlighted with a red box and has a yellow arrow pointing to it), 'Resource Access Manager', 'Access Manager', 'Cognito', and 'Secrets Manager'. Other categories listed include 'Networking & Content Delivery', 'AWS Snow Family', 'DataSync', 'AWS Elemental MediaConnect', 'AWS Elemental MediaConvert', 'AWS Elemental MediaLive', 'AWS Elemental MediaPackaging', 'AWS Elemental MediaStore', 'AWS Elemental MediaTailor', 'AWS Elemental Elemental', 'AWS Elemental Appliances & Software', 'AWS Elemental Interactive Video', 'Amazon Direct Connect', 'Amazon CloudFront', 'Amazon Route 53', 'Amazon API Gateway', 'Amazon VPC', and 'Amazon IoT SiteWise'. At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with a copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

4. In the left-hand menu, click on Roles.

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left sidebar, under the 'Access management' section, the 'Roles' option is highlighted with a red box and an orange arrow pointing to it. The main content area displays 'Security recommendations' with two items: 'Add MFA for root user' and 'Add MFA for yourself'. A message at the top right says 'Introducing the new IAM dashboard experience'.

5. Click on the Create role button.

The screenshot shows the 'Roles' list page in the AWS IAM service. The 'Roles' option in the left sidebar is highlighted with a red box and an orange arrow pointing to it. The main content area shows a table with 20 roles listed, a search bar, and a 'Create role' button which is also highlighted with a red box and an orange arrow pointing to it. A message at the top right says 'Introducing the new Roles list experience'.

6. Click on Step Functions.

The screenshot shows the AWS Step Functions service selection screen. The 'Step Functions' button is highlighted with a red box and an orange arrow pointing to it. Other services listed include Connect, DMS, Data Lifecycle Manager, Data Pipeline, DataBrew, DataSync, DeepLens, Directory Service, DynamoDB, ElasticLoadBalancing, EventBridge, Forecast, GameLift, Global Accelerator, Glue, Greengrass, GuardDuty, Health Organizational View, License Manager, MQ, MSK Connect, Machine Learning, Macie, Managed Blockchain, MediaConvert, Migration Hub, Network Firewall, SWF, SageMaker, Security Hub, Service Catalog, Storage Gateway, Systems Manager, Textract, and Transfer.

7. Click on the **Next: Permissions** button.

The screenshot shows the 'Next: Permissions' step of the AWS Step Functions setup wizard. The 'Required' section is selected. A red box highlights the 'Next: Permissions' button, which is also pointed to by an orange arrow. Other sections include 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

8. Click on the **Next: Tags** button.

The screenshot shows the 'Next: Tags' step of the AWS Step Functions setup wizard. It displays a table of policies. A red box highlights the 'Next: Tags' button, which is also pointed to by an orange arrow. Other sections include 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

9. Create a tag by entering the **app** under **Key** and **meeting-transcriber** under **Value**.

10. Click on the **Next: Review** button.

aws Services ▾ Search for services, features, marketplace [Alt+S] cloud_user @ 1885-6494-7610 ▾ Global ▾ Support ▾

Create role

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Re
app	meeting-transcriber	
Add new key		

Cancel **Next: Review**

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11. In Role name, enter the name **step-functions-lambda-role**.
12. Click on the **Create role** button.

aws Services ▾ Search for services, features, marketplace [Alt+S] cloud_user @ 1885-6494-7610 ▾ Global ▾ Support ▾

Create role

Review

Provide the required information below and review this role before you create it.

Role name*	step-functions-lambda-role
Use alphanumeric and '!', '@', '-' characters. Maximum 64 characters.	
Role description	Allows Step Functions to access AWS resources on your behalf.
Maximum 1000 characters. Use alphanumeric and '!', '@', '-' characters.	

Required

Cancel **Create role**

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hence, the IAM role has been created.

aws Services ▾ Search for services, features, marketplace [Alt+S] cloud_user @ 1885-6494-7610 ▾ Global ▾ Support ▾

✔ The role **step-functions-lambda-role** has been created.

Identity and Access Management (IAM) ✖

- [Dashboard](#)
- [Access management](#)
 - [User groups](#)
 - [Users](#)
 - Roles** (29) Info
 - [Policies](#)
 - [Identity providers](#)
 - [Account settings](#)
- [Access reports](#)

IAM > Roles

Roles (29) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

1 2 > ⌂

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2: Create Step Function

1. Click on Services.

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with the AWS logo, a 'Services' dropdown menu (which is highlighted with a red box and has a yellow arrow pointing to it), a search icon, a refresh icon, and user information (cloud_user @ 5128-3147-4478, N. Virginia, Support). Below the navigation bar, the title 'AWS Management Console' is displayed. On the left, there is a sidebar titled 'AWS services' with sections for 'Recently visited services' (listing 'Your recently visited AWS services appear here.') and 'All services'. At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with a copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

2. Select the Step Function from Application Integration.

The screenshot shows the 'All services' list in the AWS Management Console. On the left, there is a sidebar with 'Favorites' (Registry, Station, Amazon Personalize, AR & VR), 'Recently visited' (Console Home, IAM), and 'Storage' (Amazon Sumerian, Amazon Rekognition, Amazon Textract, Application Integration, Step Functions). The main area lists various AWS services under categories like 'Quantum Technologies', 'Management & Governance', and 'AWS Storage'. An orange arrow points to the 'Step Functions' service, which is highlighted with a red box. At the bottom, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

3. Click on the Get started button.

Sales & Fulfillment by Amazon

Services ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

can create and run a series of checkpointed and event-driven workflows that maintain the application state.

Get started

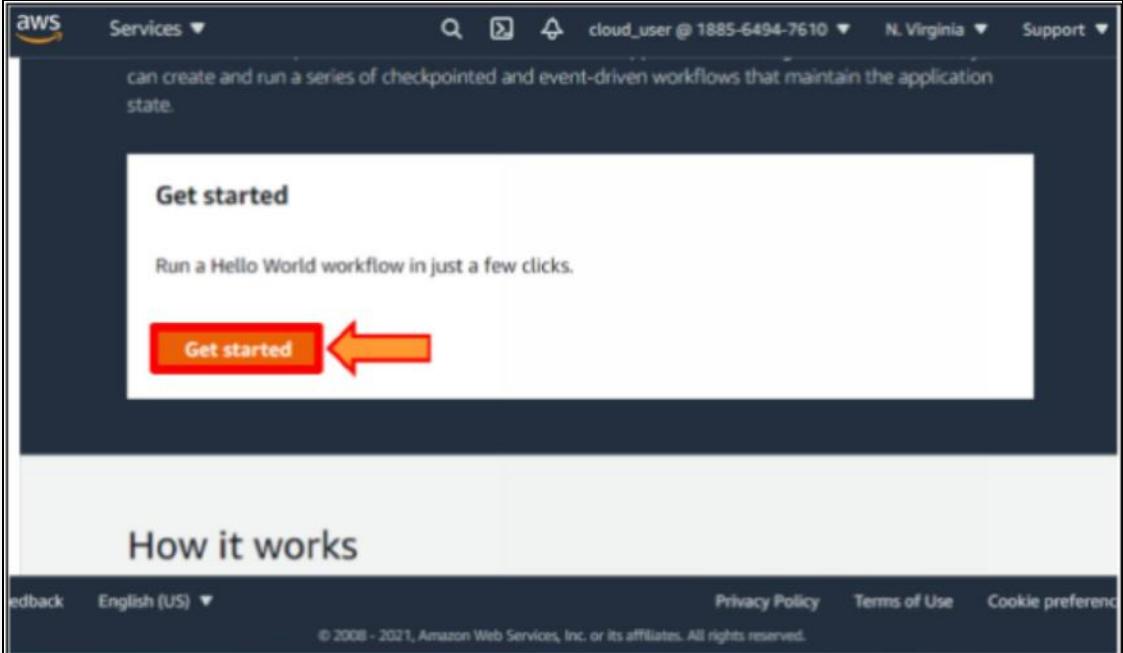
Run a Hello World workflow in just a few clicks.

[Get started](#)

How it works

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



4. Click on **State machines** from the left-hand side menu.

Sales & Fulfillment by Amazon

Services ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

Step Functions

[State machines](#) 

Activities

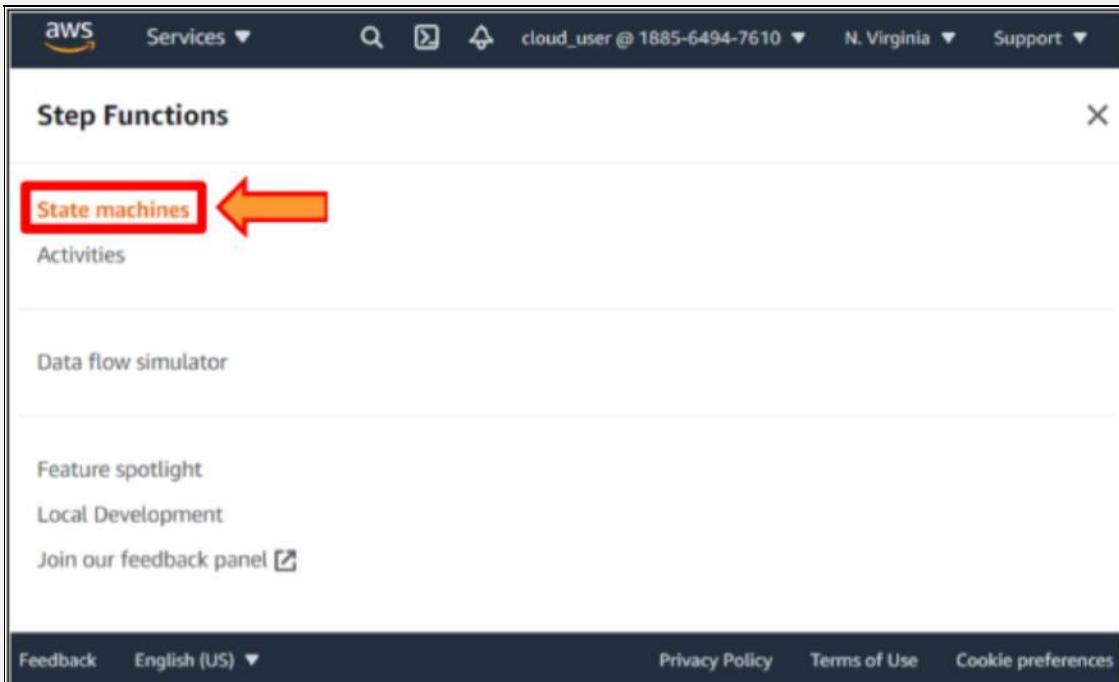
Data flow simulator

Feature spotlight

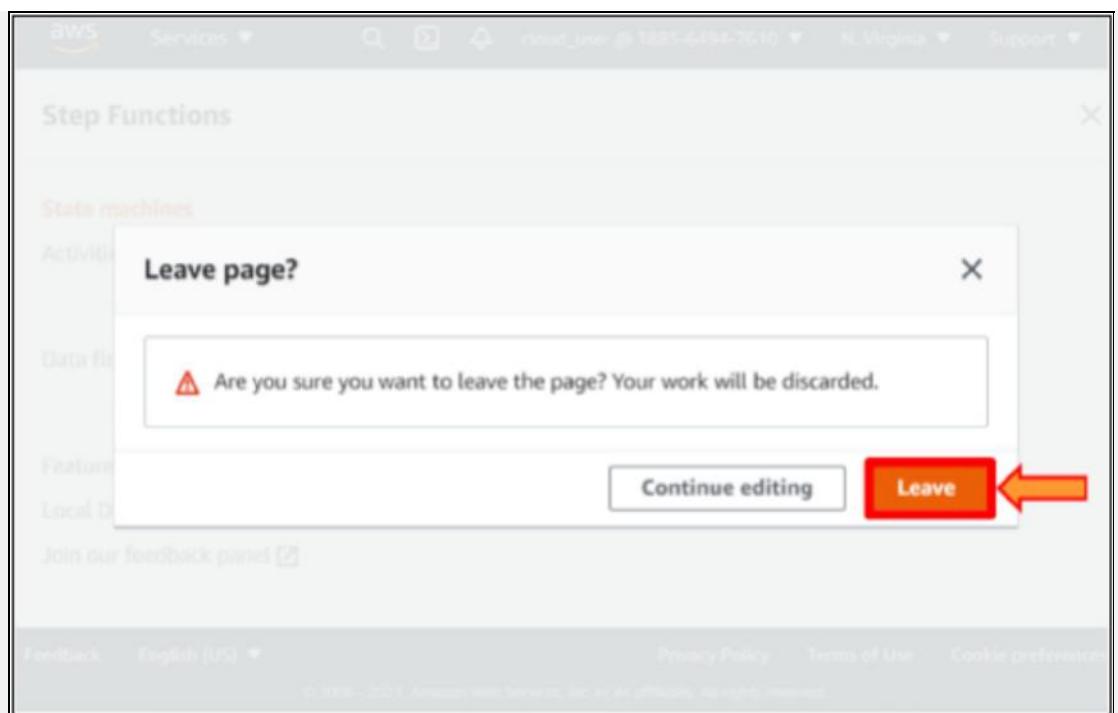
Local Development

Join our feedback panel 

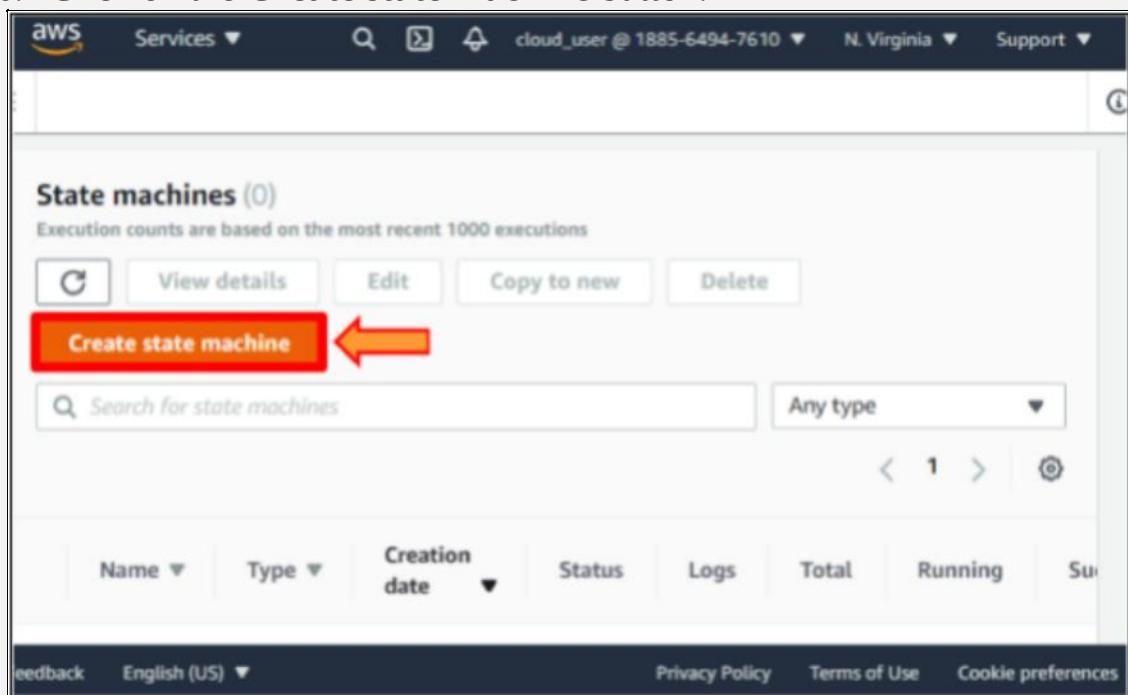
Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences



5. Click on the **Leave** button.



6. Click on the **Create state machine** button.



7. Select **Write your workflow in code.**

Sales Services Search User N. Virginia Support

Write your workflow in code

Author your workflow using Amazon States Language. You can generate code snippets to easily build out your workflow steps.

Run a sample project

Deploy and run a fully functioning sample project in minutes using CloudFormation.

Type

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. Select Standard under Type.

Sales Services Search User N. Virginia Support

Type

Standard
Durable, checkpointed workflows for machine learning, order fulfillment, IT/DevOps automation, ETL jobs, and other long-duration workloads.

Express
Designed for workflows for streaming data microservices orchestration, IoT data ingestion, mobile backends, and other short duration, high-event-rate workloads.

Help me decide

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Click on the Next button.

The screenshot shows the AWS Lambda function editor interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and account information ('cloud_user @ 1885-6494-7610', 'N. Virginia', 'Support'). Below the navigation is a code editor window containing the following JSON-like code:

```
11     "Type": "Pass",
12     "Result": "World",
13     "End": true
14   }
15 }
16 }
```

At the bottom right of the editor is a red arrow pointing to a large orange 'Next' button. Below the editor, there's a footer with links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. The footer also includes the copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

10. Under Name, enter the name **Categorize-Audio-Data-Pipeline**.

The screenshot shows the 'Specify details' step of the AWS Lambda state machine creation wizard. The title 'Specify details' is at the top, followed by 'Step 2 of 2'. There is a 'Name' input field with the placeholder 'State machine name'. The text 'Categorize-Audio-Data-Pipeline' is entered into this field and is highlighted with a red border. A red arrow points from the left towards this input field. Below the input field is a note: 'Must be 1-80 characters. Can use alphanumeric characters, dashes, or underscores.' At the bottom of the screen is a footer with links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with the copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

11. Click on **Choose an existing role**.



Permissions

Execution role

The IAM role that defines which resources your state machine has permission to access during execution. To create a custom role, go to the [IAM console](#).

Create new role

Let Step Functions create a new role for you based on your state machine's definition and configuration details.

Choose an existing role

Enter a role ARN

Existing roles

step-functions-lambda-role

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

12. Under Tags - optional, create a tag by entering **app** under Key and **meeting-transcriber** under Value.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

app

Value - optional

meeting-transcriber

Remove tag

Add tag

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

13. Click on the **Create state machine** button.

Sales Support

Services ▾

cloud_user @ 1885-6494-7610 N. Virginia ▾ Support

▶ Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

An execution role will be created with full permissions.
A new execution role named StepFunctions-MyStateMachine-role-49db3b38 will be created.
All required permissions for the actions specified in your state machine will be auto-generated.

▶ Review auto-generated permissions

Cancel  Create state machine

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

14. Under ARN in Details, copy the ARN into your clipboard.

Sales Support

Services ▾

cloud_user @ 1885-6494-7610 N. Virginia ▾ Support

Categorize-Audio-Data-Pipeline

Edit Start execution Delete Actions ▾

Details

ARN  

IAM role ARN
[arn:aws:iam::188564947610:role/step-functions-lambda-role](#)

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3: Update Lambda to Work with Step Functions

1. Click on Services.

The screenshot shows the AWS Management Console homepage. At the top left, there is a 'Services' dropdown menu with a red box and a yellow arrow pointing to it. The main title 'AWS Management Console' is centered above a sidebar titled 'AWS services'. The sidebar contains sections for 'Recently visited services' (with a note: 'Your recently visited AWS services appear here.') and 'All services'. At the bottom of the sidebar, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. A copyright notice at the very bottom states: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

2. Select Lambda from Compute.

The screenshot shows the AWS Management Console with the 'Compute' service selected. On the left, there is a sidebar with 'Favorites' (Resource Groups & Tags) and 'Recently visited' (Console Home, IAM). The main area is titled 'All services' and lists various compute-related services. The 'Lambda' service is highlighted with a red box and a yellow arrow pointing to it. Other listed services include Compute, EC2, Lightsail, Batch, Elastic Beanstalk, Serverless, Application Repository, AWS Outposts, EC2 Image Builder, Machine Learning (Amazon SageMaker), Amazon Augmented AI, Amazon CodeGuru, Amazon DevOps Guru, Amazon Comprehend, Amazon Forecast, and Amazon Fraud Detector. At the bottom, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

3. Click on run-step-functions-lambda.

name		size	modified
transcribe-audio-lambda	-	Zip Python 3.8	497.0 byte 25 min ago
transcribe-status-lambda	-	Zip Python 3.8	377.0 byte 25 min ago
run-step-functions-lambda	-	Zip Python 3.8	426.0 byte 25 min ago
categorize-data-lambda	-	Zip Python 3.8	854.0 byte 25 min ago

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

- Under Function code, click on the **index.py**.
- In the Python file, note that it contains a placeholder at the end called **STATEMACHINEARN**.

```

import boto3
import json
import os
stepfunctions_client = boto3.client('stepfunctions')
def lambda_handler(event, context):
    s3event = event['Records'][0]['s3']
    s3bucket = s3event['bucket']['name']
    s3key = s3event['object']['key']
    step_state = {
        "s3_bucket": s3bucket,
        "s3_audio_key": s3key
    }
    response = stepfunctions_client.start_execution(
        stateMachineArn=os.environ['STATEMACHINEARN'])

```

- Click on the **Configuration** tab.

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with 'Services' dropdown, search bar, and user info ('cloud_user @ 1885-6494-7610'). Below the navigation is a breadcrumb trail: 'Lambda > ... > run-step-functions-lambda'. On the right side, there's a sidebar with 'Application' details: 'cfst-3269-8b6f01d374c4ae8e067af01a c1bd'. The main content area has tabs: 'Code' (highlighted in orange), 'Test', 'Monitor', 'Configuration' (highlighted with a red box and a yellow arrow), and 'Versions'. Below the tabs, there's a 'Code source' section with an 'Upload from' button. A toolbar at the bottom includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (highlighted in orange), 'Deploy', and a green 'Changes deployed' button. The footer contains links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

7. Click on Environment variables.

The screenshot shows the AWS Lambda console interface. The left sidebar lists several configuration sections: 'Permissions', 'Destination', 'Environment variables' (highlighted with a red box and a yellow arrow), 'Tags', 'VPC', and 'Monitoring and operations tools'. The main content area displays 'Timeout' settings: '0 min 3 sec'. A toolbar at the bottom includes 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Test' (highlighted in orange), 'Deploy', and a green 'Changes deployed' button. The footer contains links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

8. Click on the Edit button.

The screenshot shows the AWS Lambda function configuration page for 'run-step-functions-lambda'. The left sidebar has tabs for General configuration, Triggers, Permissions, Destinations, Environment variables (which is selected and highlighted in orange), and Tags. The main content area is titled 'Environment variables (0)' with an 'Edit' button. Below it, there's a table with columns 'Key' and 'Value', and a message stating 'No environment variables'. At the bottom right of the main area is another 'Edit' button, which is also highlighted with a red box and an orange arrow pointing to it.

9. Click on the **Add environment variable** button.

The screenshot shows the 'Edit environment variables' page for the same Lambda function. It includes a note about defining environment variables as key-value pairs and a link to learn more. Below that, it says 'There are no environment variables on this function.' At the top left is an 'Add environment variable' button, which is highlighted with a red box and an orange arrow pointing to it. At the bottom right are 'Cancel' and 'Save' buttons.

10. Add an environment variable by entering **STATEMACHINEARN** under Key and pasting your state machine's ARN under **Value**.

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key
STATEMACHINEARN

Value
arn:aws:states:us-east-1:188564947610:stateMachine:Categorize-Audio-Data-Pipeline

Remove

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11. Click on the Save button.

Value
arn:aws:states:us-east-1:188564947610:stateMachine:Categorize-Audio-Data-Pipeline

Remove

Add environment variable

► Encryption configuration

Save

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 4: Update S3 to Work with Step Functions

1. Click on Services.

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with the AWS logo, a 'Services' dropdown menu (highlighted with a red box and an orange arrow), a search icon, a refresh icon, the user information 'cloud_user @ 5128-3147-4478', the region 'N. Virginia', and a 'Support' link. Below the navigation bar, the main title 'AWS Management Console' is displayed. On the left, there is a sidebar titled 'AWS services' with sections for 'Recently visited services' (listing 'Your recently visited AWS services appear here.') and 'All services'. At the bottom of the page, there is a footer with links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences', along with a copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'

2. Select S3 from Storage.

The screenshot shows the AWS Management Console with the 'Services' dropdown menu expanded. The 'Storage' section is highlighted with a red box and an orange arrow pointing to the 'S3' service. Other services listed under Storage include EFS, FSx, S3 Glacier, Storage Gateway, and AWS Backup. The rest of the page includes a 'Favorites' sidebar with 'Resource Groups & Ta...', 'Recently visited' items like 'Console Home' and 'IAM', and a large 'All services' list on the right side.

3. Click on the created bucket.

Buckets (1) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

[Create bucket](#)

Find buckets by name

Name	AWS Region	Access	Creation date
meeting-audio-21e062b0	US East (N. Virginia) us-east-1	Bucket and objects not public	October 13, 2021, 23:58:33 (UTC-07:00)

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

4. Click on the **Properties** tab.

Amazon S3 > meeting-audio-21e062b0

meeting-audio-21e062b0 [Info](#)

Objects [Properties](#) Metrics Management Access Points

Bucket overview

AWS Region	Amazon Resource Name (ARN)
US East (N. Virginia) us-east-1	arn:aws:s3:::meeting-audio-21e062b0

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Click on the **Create event notification** button.

Screenshot of the AWS CloudWatch Events console showing the 'Event notifications' page. The page displays a table with one row, indicating 'No event notifications'. A red arrow points to the 'Create event notification' button at the bottom of the page.

Name	Event types	Filters	Destin type
No event notifications			

Choose Create event notification to be notified when a specific event occurs.

Create event notification

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Under General configuration, set the following values:
- In the **Event** name, enter the name **trigger-audio-processing-event**.
 - In Prefix - optional, enter **upload/**.
 - In Suffix - optional, enter **.mp3**.

Screenshot of the 'General configuration' page. The 'Event name' field contains 'trigger-audio-processing-event', the 'Prefix - optional' field contains 'upload/', and the 'Suffix - optional' field contains '.mp3'. Each input field has a red border and a red arrow pointing to it from the left.

General configuration

Event name
trigger-audio-processing-event
Event name can contain up to 255 characters.

Prefix - optional
Limit the notifications to objects with key starting with specified characters.
upload/

Suffix - optional
Limit the notifications to objects with key ending with specified characters.
.mp3

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

7. Select the checkbox next to **All object create events**.



All object create events
s3:ObjectCreated:"



- Put
s3:ObjectCreated:Put
- Post
s3:ObjectCreated:Post
- Copy
s3:ObjectCreated:Copy
- Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload

All object delete events
s3:ObjectRemoved:"

Feedback English (US) ▾

Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. In the dropdown menu under the Lambda function, select the **run-step-functions-lambda** function.



Lambda function

Run a Lambda function script based on S3 events.

SNS topic

Send notifications to email, SMS, or an HTTP endpoint.

SQS queue

Send notifications to an SQS queue to be read by a server.

Specify Lambda function

Choose from your Lambda functions

Enter Lambda function ARN

Lambda function

run-step-functions-lambda



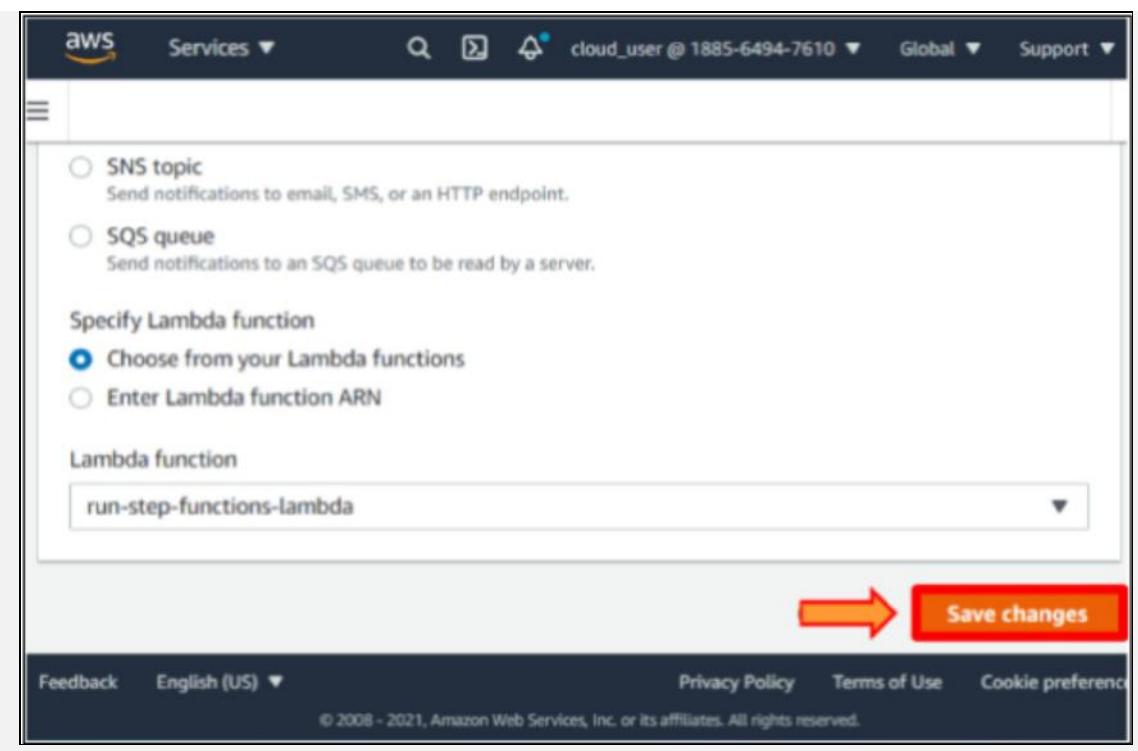
Feedback English (US) ▾

Privacy Policy Terms of Use

Cookie preferences

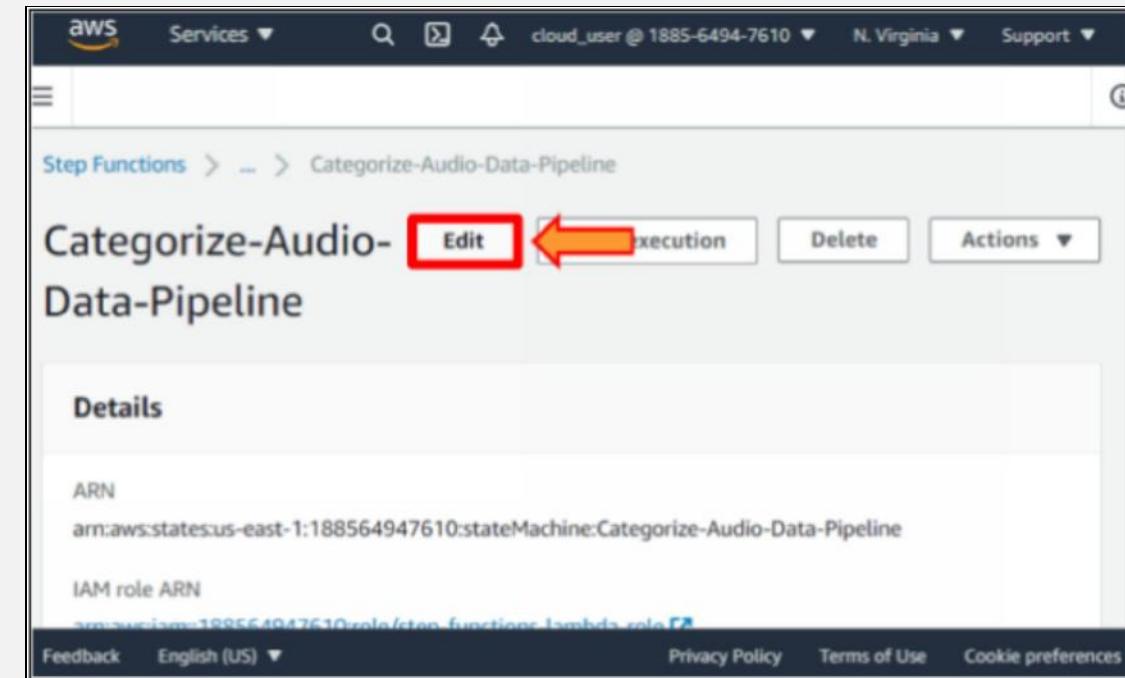
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Click on the **Save changes** button.

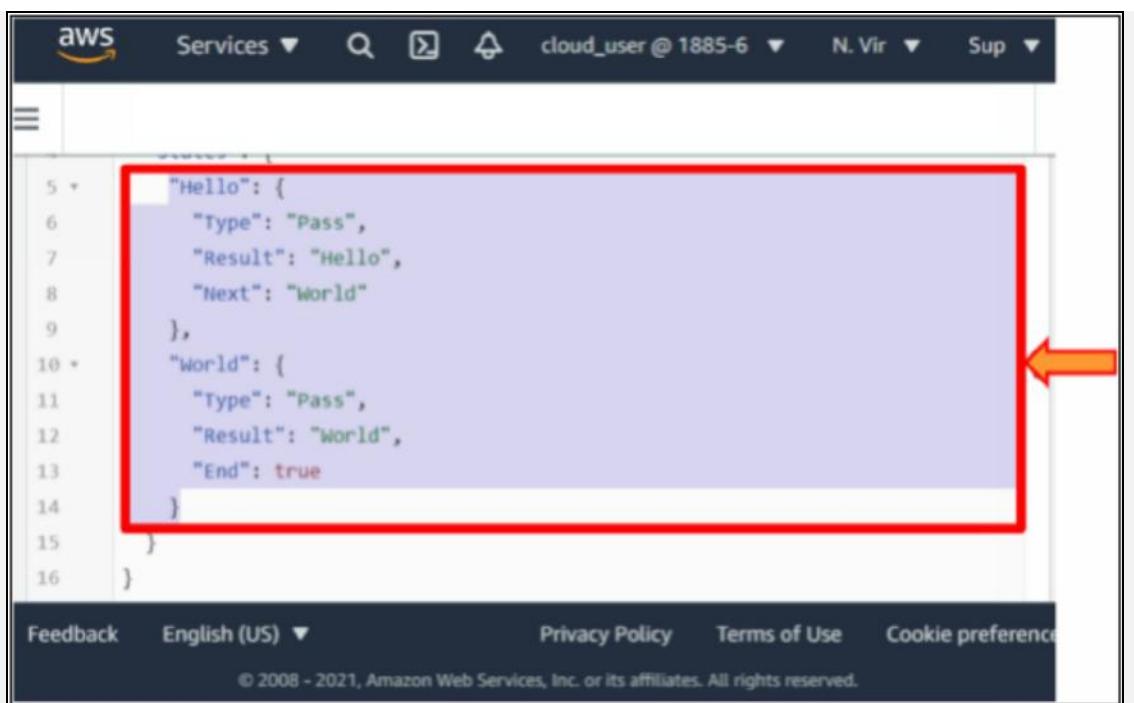


Step 5: Create Step Function Flow

1. Go back to the **Step Function** dashboard.
2. Click on the **Edit** button.



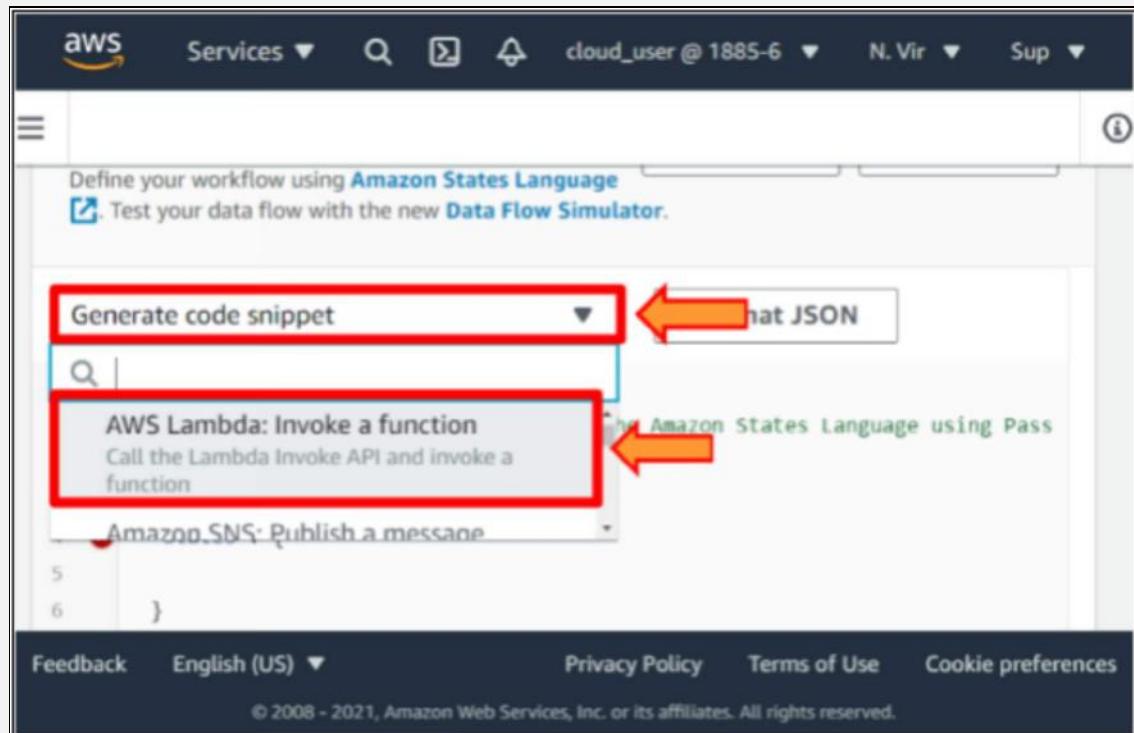
3. Create a Transcribe Job.
4. From the code snippet, delete the 'Hello' 'World' states after "States": {.



```
5 +     "Hello": {  
6         "Type": "Pass",  
7         "Result": "Hello",  
8         "Next": "World"  
9     },  
10    "World": {  
11        "Type": "Pass",  
12        "Result": "World",  
13        "End": true  
14    }  
15}  
16}
```

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. In the **Generate code snippet** dropdown, select the **AWS Lambda: Invoke a function**.



Define your workflow using [Amazon States Language](#)
 Test your data flow with the new [Data Flow Simulator](#).

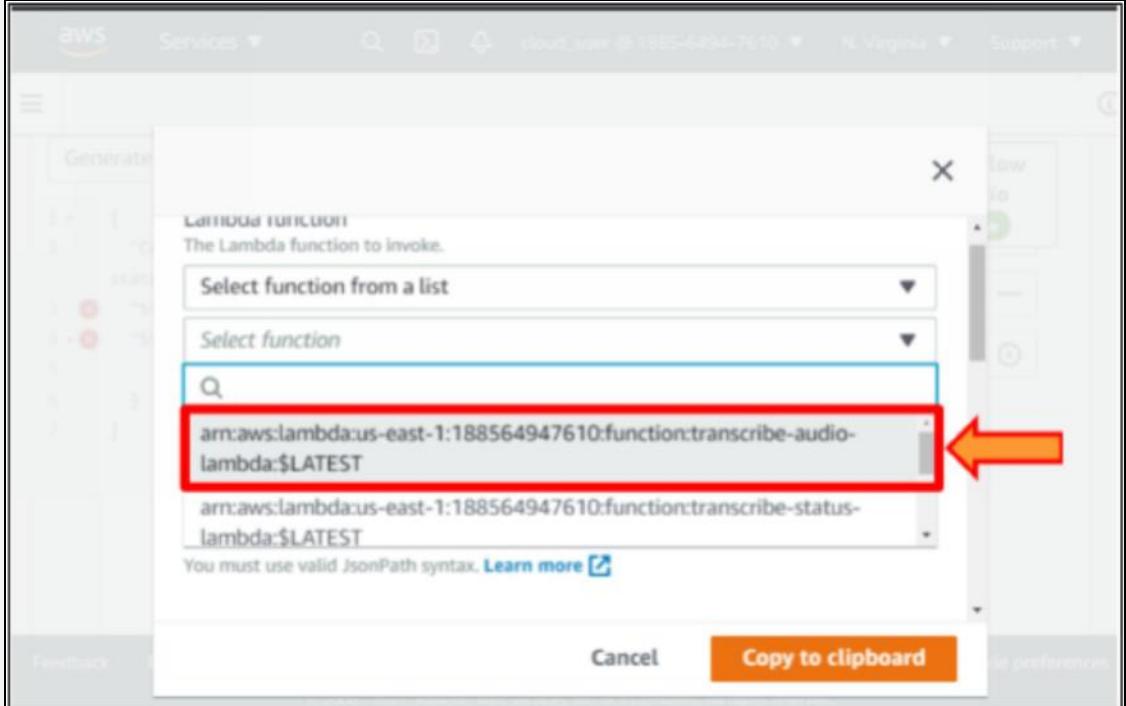
Generate code snippet ▾ that JSON

AWS Lambda: Invoke a function Call the Lambda Invoke API and invoke a function

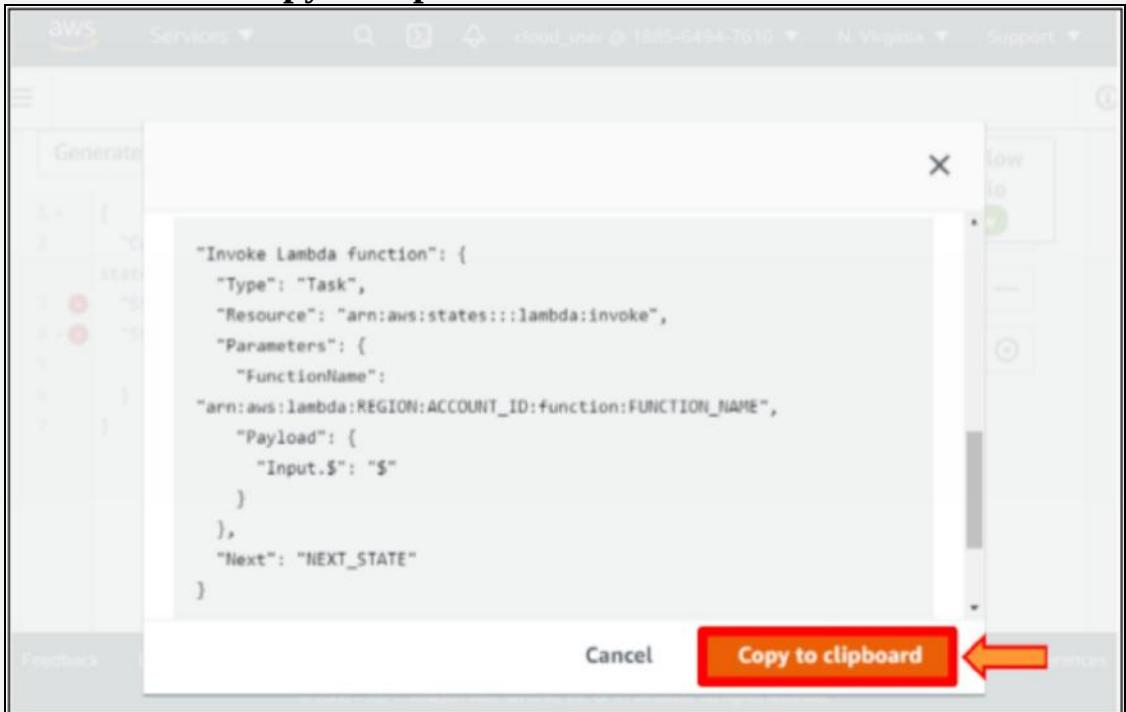
Amazon SNS: Publish a message

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Select the **transcribe-audio-lambda** function.



7. Click on the **Copy to clipboard** button.



8. Paste the code into the code snippet after "**States**": {.

AWS Services ▾ Q cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

Generate code snippet ▾ Format JSON

```
1+ {
2     "Comment": "A Hello World example of the Amazon States Language using Pass
states",
3     "StartAt": "Hello",
4     "States": {
5         "Hello": {
6             "Type": "Task",
7             "Resource": "arn:aws:lambda:us-east-1:123456789012:function:HelloWorld"
9         }
10    }
11 }
```

Workflow Studio New

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



9. Click on the **Format JSON** button.

AWS Services ▾ Q cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

Definition

Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Export ▾ Layout ▾

Generate code snippet ▾ Format JSON

```
1+ {
2     "Comment": "A Hello World example of the Amazon States Language using Pass
states",
3     "StartAt": "Hello",
4     "States": {
5         "Hello": {
6             "Type": "Task",
7             "Resource": "arn:aws:states:::lambda:invoke",
8             "Parameters": {
9                 "FunctionName": "HelloWorld"
10            }
11        }
12    }
13 }
```

Workflow Studio New

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



10. Rename **Invoke lambda function** to **Start Transcription**.

AWS Services Q cloud_user @ 1885-6494-7610 N. Virginia Support

```
1 * {
2     "Comment": "A Hello World example of the Amazon States Language using Pass
states",
3     "StartAt": "Hello", ←
4     "States": {
5         "Invoke Lambda function": {
6             "Type": "Task",
7             "Resource": "arn:aws:states:::lambda:invoke",
8             "Parameters": {
9                 "FunctionName": "arn:aws:lambda:us-east-1:188564947610:function:transcribe
audio-lambda:$LATEST",
10            "Payload": {
11                "Input.$": "$"
12            }
13        },
14        "Next": "NEXT_STATE"
}
```

Feedback English (US) Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11. At StartAt: Hello, replace Hello with: Start Transcription.

AWS Services Q cloud_user @ 1885-6494-7610 N. Virginia Support

```
1 * {
2     "Comment": "A Hello World example of the Amazon States Language using Pass
states",
3     "StartAt": "Start Transcription", ←
4     "States": {
5         "Start Transcription": {
6             "Type": "Task",
7             "Resource": "arn:aws:states:::lambda:invoke",
8             "Parameters": {
9                 "FunctionName": "arn:aws:lambda:us-east-1:188564947610:function:transcribe
audio-lambda:$LATEST",
10            "Payload": {
11                "Input.$": "$"
12            }
13        },
14        "Next": "NEXT_STATE"
}
```

Feedback English (US) Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

12. Create a Wait State.

13. In the Generate code snippet dropdown, select the Wait state.

The screenshot shows the AWS Lambda State Machine Configuration interface. A green notification bar at the top left says "Code snippet copied to clipboard." Below it, a modal window displays a JSON code snippet. The "Generate code snippet" button and the JSON preview area are highlighted with red boxes and arrows pointing to them from the left. The JSON code is as follows:

```
1 {  
2     "Comment": "A Hello World example of the Amazon States Language using Pass  
3     states",  
4     "StartAt": "Start Transcription",
```

At the bottom right of the modal is a red "Copy to clipboard" button.

14. Click on the **Copy to clipboard** button.

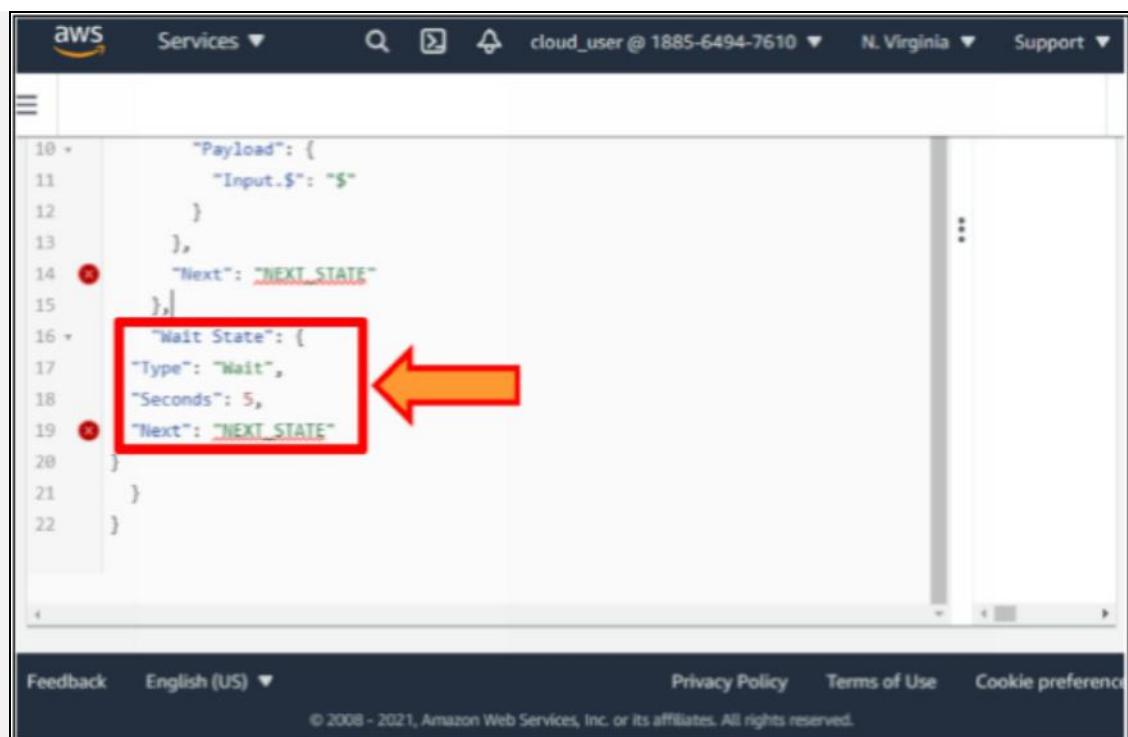
The screenshot shows the same AWS Lambda State Machine Configuration interface as the previous step. The modal window now has the "Copy to clipboard" button highlighted with a large red arrow pointing to it.

15. Paste it under the }, that is under **Next: NEXT_STATE**.

AWS Services cloud_user @ 1885-6494-7610 N. Virginia Support

```
10 +     "Payload": {  
11         "Input.$": "$"  
12     }  
13 },  
14     "Next": "NEXT_STATE"  
15 },  
16 +     "Wait State": {  
17         "Type": "Wait",  
18         "Seconds": 5,  
19         "Next": "NEXT_STATE"  
20     }  
21 }  
22 }
```

Feedback English (US) Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



16. Click on the Format JSON button.

AWS Services cloud_user @ 1885-6494-7610 N. Virginia Support

Definition

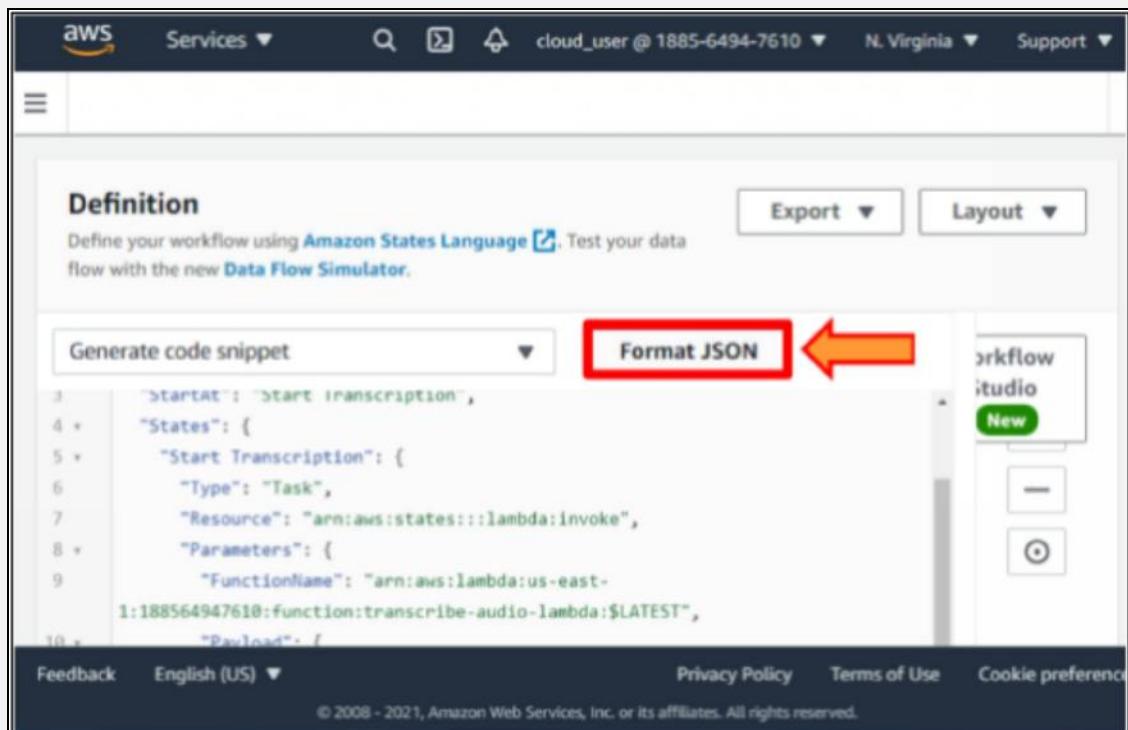
Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Generate code snippet **Format JSON** Export Layout

```
3     "StartAt": "Start Transcription",  
4 *     "States": {  
5     "Start Transcription": {  
6         "Type": "Task",  
7         "Resource": "arn:aws:states:::lambda:invoke",  
8         "Parameters": {  
9             "FunctionName": "arn:aws:lambda:us-east-  
10            1:188564947610:function:transcribe-audio-lambda:$LATEST",  
11            "Payload": {  
12        }}
```

Workflow Studio New

Feedback English (US) Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



17. Replace **Wait State** with: **Wait for Transcribe**.
18. Change **Seconds: 5**, to **Seconds": 30**.
19. In the Start Transcription state, change **NEXT_STATE** to: **Wait for Transcribe**.

Sales

Services ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

```
7     "Resource": "arn:aws:states:::lambda:invoke",
8     "Parameters": {
9       "FunctionName": "arn:aws:lambda:us-east-
10      1:188564947610:function:transcribe-audio-lambda:$LATEST",
11      "Payload": {
12        "Input.$": "$"
13      }
14      ],
15      "Next": "Wait for Transcribe"
16    },
17    "Wait for Transcribe": [
18      {
19        "Type": "Wait",
20        "Seconds": 30,
21        "Next": "NEXT_STATE"
22      }
23    ]
24  }
```

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

20. In the **Generate code snippet** dropdown, select **AWS Lambda: Invoke a function**.

Sales

Services ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

Definition

Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Export ▾ Layout ▾

Generate code snippet ▾ JSON

Call any API in the AWS SDK

AWS Lambda: Invoke a function

Call the Lambda Invoke API and invoke a function

Amazon States Language using Pass

Amazon SNS: Publish a message

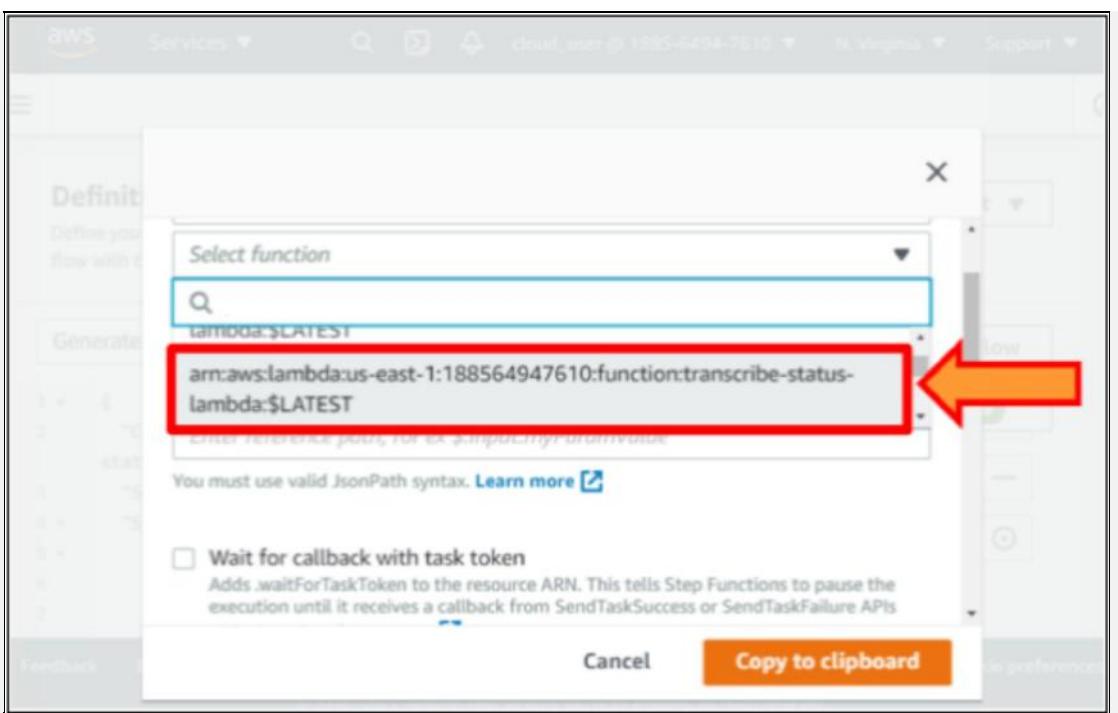
```
6   "Type": "Task",
7   "Resource": "arn:aws:states:::lambda:invoke",
8   "Parameters": {
9     "FunctionName": "arn:aws:lambda:us-east-
```

Workflow Studio New

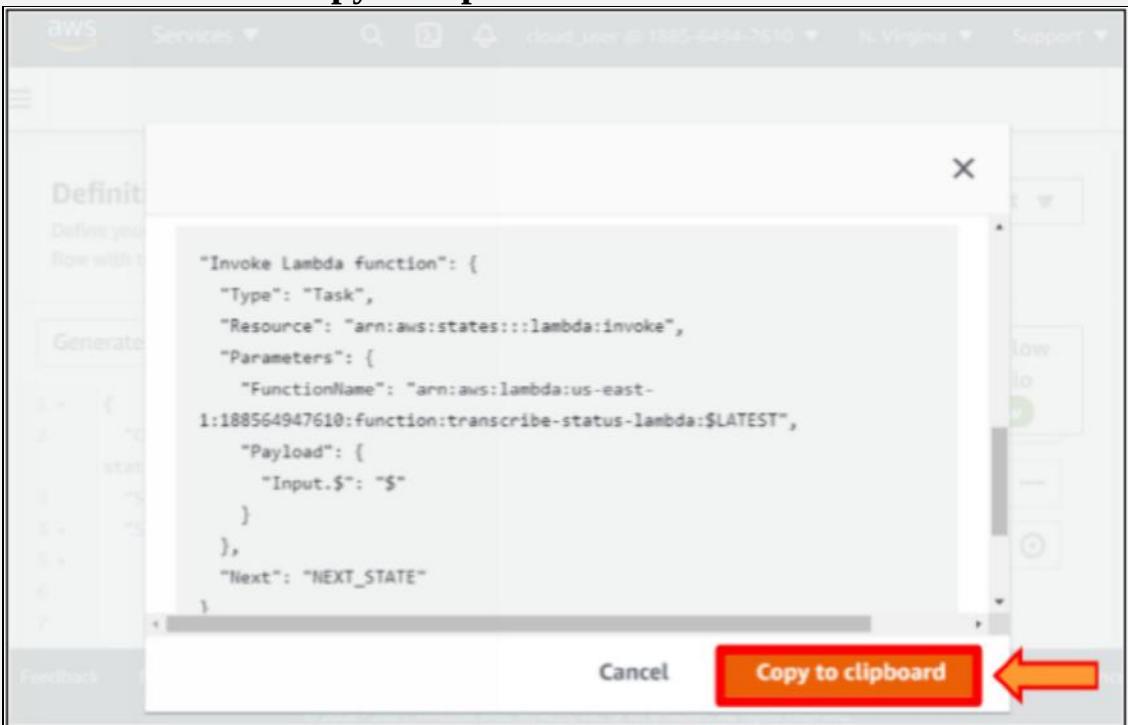
Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

21. Select the **transcribe-status-lambda** function.



22. Click on the **Copy to clipboard** button.

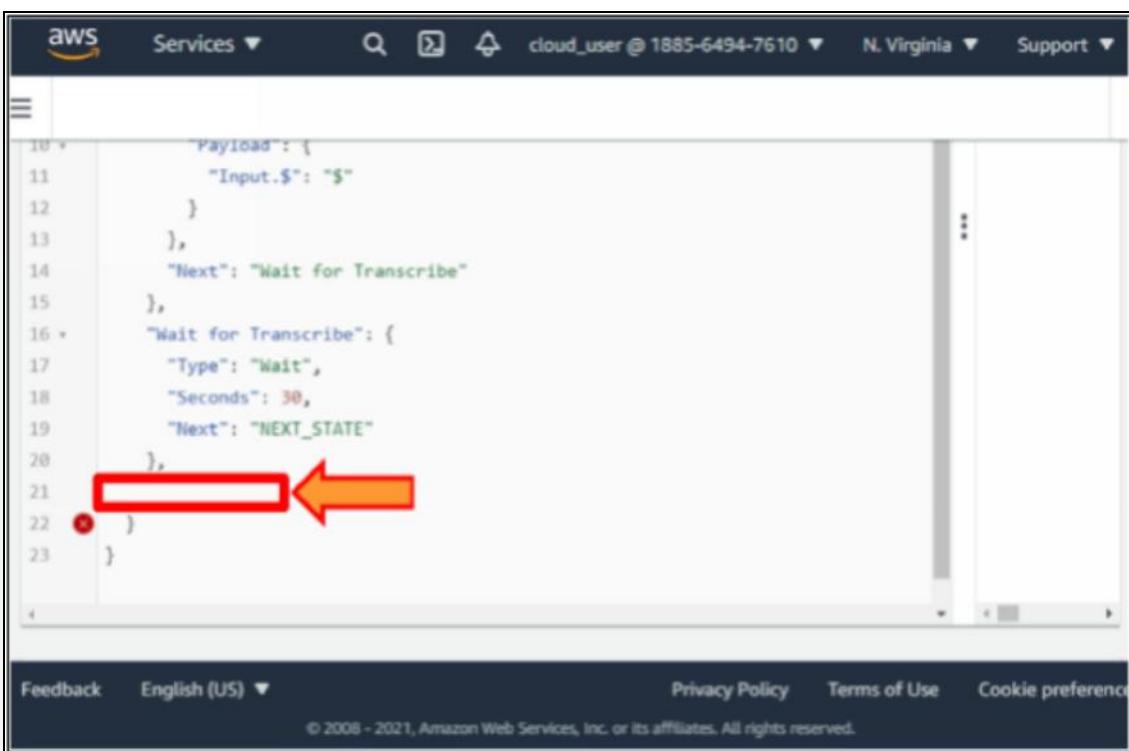


23. Paste it under the }, that is under **Next: NEXT_STATE**.

AWS Services Search cloud_user @ 1885-6494-7610 N. Virginia Support

```
10 *     "Payload": {
11         "Input.$": "$"
12     },
13     "Next": "Wait for Transcribe"
14 },
15 "Wait for Transcribe": {
16     "Type": "Wait",
17     "Seconds": 30,
18     "Next": "NEXT_STATE"
19 },
20 },
21 }
```

Feedback English (US) Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



24. Click on the Format JSON button.

AWS Services Search cloud_user @ 1885-6494-7610 N. Virginia Support

Definition

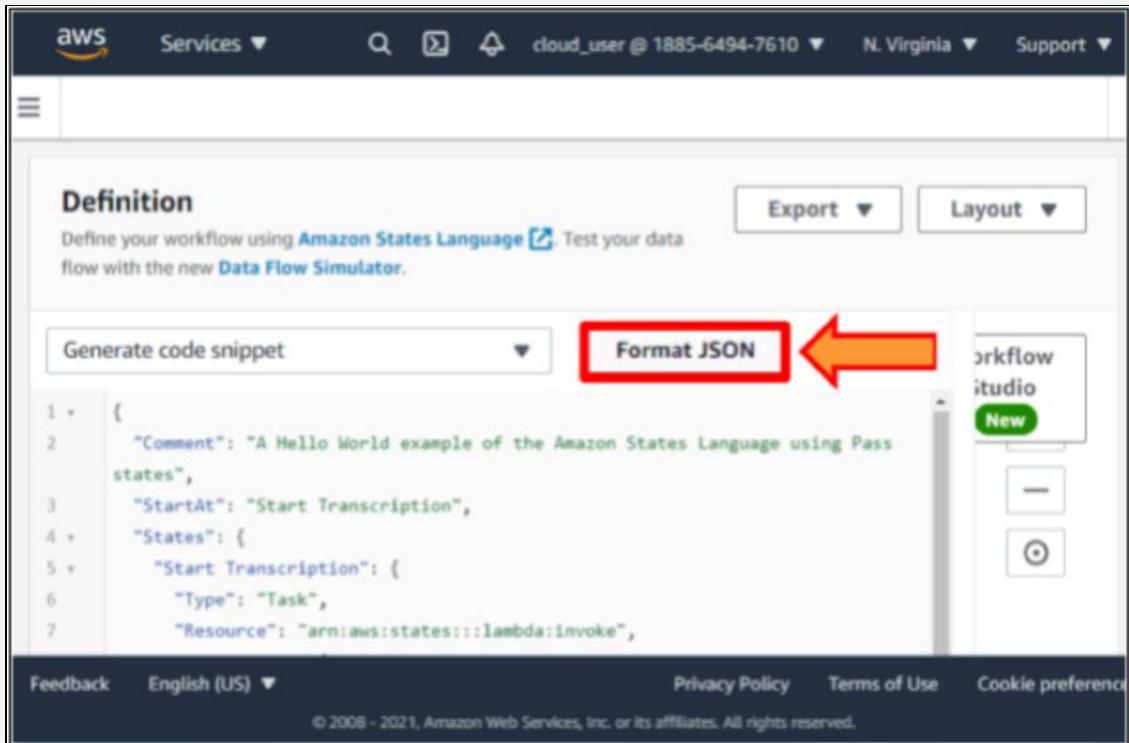
Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Generate code snippet Export Layout

```
1 * {
2     "Comment": "A Hello World example of the Amazon States Language using Pass states",
3     "StartAt": "Start Transcription",
4     "States": {
5         "Start Transcription": {
6             "Type": "Task",
7             "Resource": "arn:aws:states:::lambda:invoke",
```

Workflow Studio New

Feedback English (US) Privacy Policy Terms of Use Cookie preferences
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



25. Rename Invoke lambda function to Check Transcribe Status.

Screenshot of the AWS Lambda function code editor showing the state machine configuration. The code is as follows:

```
15     },
16     "Wait for Transcribe": {
17         "Type": "Wait",
18         "Seconds": 30,
19         "Next": "NEXT_STATE" ←
20     },
21     "Invoke Lambda function": {
22         "Type": "Task",
23         "Resource": "arn:aws:states:::lambda:invoke",
24         "Parameters": {
25             "FunctionName": "arn:aws:lambda:us-east-
1:188564947610:function:transcribe-status-lambda:$LATEST",
26             "Payload": {
27                 "Input.$": "$"
28             }
29         },
30         "Next": "NEXT_STATE"
31     }
32 }
```

The code is annotated with two red arrows pointing to the "Next" fields of the "Wait for Transcribe" and "Invoke Lambda function" states, which are highlighted in red.

26. In the Wait State, change NEXT_STATE to Check Transcribe Status.

Screenshot of the AWS Lambda function code editor showing the state machine configuration after modification. The code is as follows:

```
11     "Input.$": "$"
12     }
13     },
14     "Next": "Wait for Transcribe" ←
15     },
16     "Wait for Transcribe": {
17         "Type": "Wait",
18         "Seconds": 30,
19         "Next": "Check Transcribe Status"
20     },
21     "Check Transcribe Status": {
22         "Type": "Task",
23         "Resource": "arn:aws:states:::lambda:invoke",
24         "Parameters": {
25             "FunctionName": "arn:aws:lambda:us-east-
1:188564947610:function:transcribe-status-lambda:$LATEST",
26             "Payload": {
27                 "Input.$": "$"
28             }
29         },
30         "Next": "NEXT_STATE"
31     }
32 }
```

The code is annotated with two red arrows pointing to the "Next" fields of the "Wait for Transcribe" and "Check Transcribe Status" states, which are highlighted in red.

27. Create a Completed and Failed State
28. In the Generate code snippet dropdown, select Choice state.

Definition

Export ▾

Layout ▾

Define your workflow using [Amazon States Language](#) . Test your data flow with the new [Data Flow Simulator](#).

Generate code snippet

JSON



Parallel state

Create parallel branches of execution

Choice state

Adds branching logic to a state machine

Pass state

6

```
  "Type": "Task",
  "Resource": "arn:aws:states:::lambda:invoke",
```

Feedback

English (US) ▾

Privacy Policy

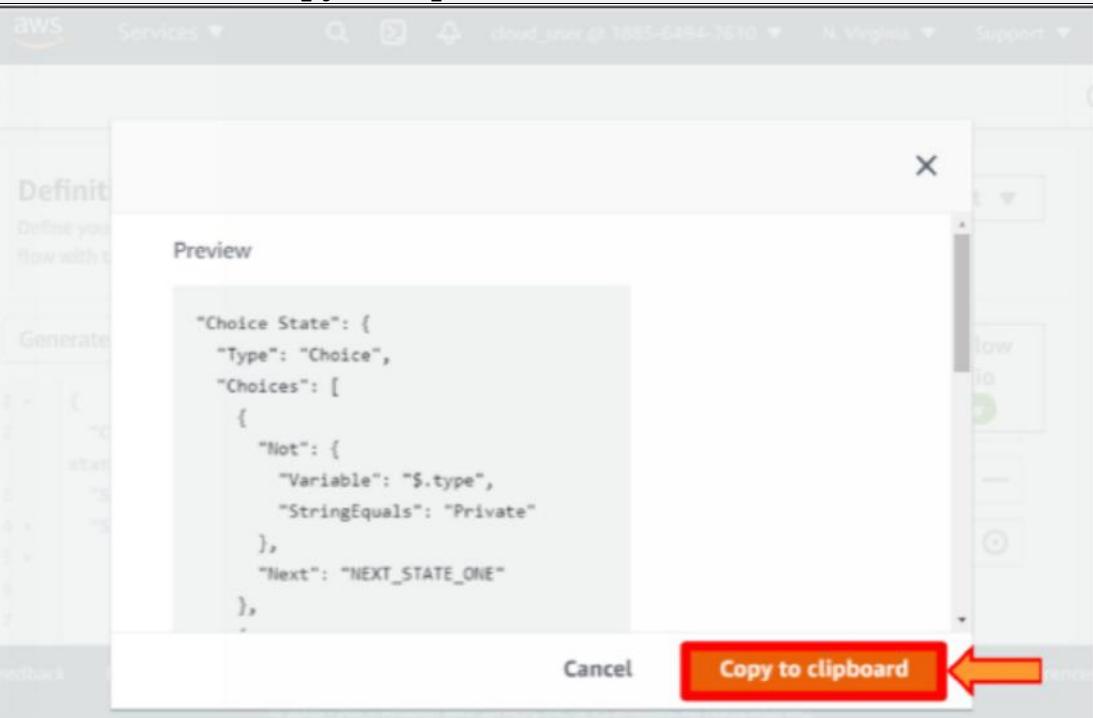
Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Workflow studio

New

29. Click on the Copy to clipboard button.

The screenshot shows the 'Definition' section of the AWS Lambda State Machine Definition tool. In the 'Generate code snippet' dropdown, 'JSON' is selected. Below it, a 'Choice state' is chosen, which is highlighted with a red box. The preview window displays the JSON configuration for this state. At the bottom right of the preview window, there is a red box around the 'Copy to clipboard' button, which is also highlighted with a red arrow.

```
  "Choice State": {
    "Type": "Choice",
    "Choices": [
      {
        "Not": {
          "Variable": "$.type",
          "StringEquals": "Private"
        },
        "Next": "NEXT_STATE_ONE"
      }
    ]
  }
```

Cancel **Copy to clipboard**

30. Paste it under } found under Next: NEXT_STATE.

Servicess ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

```
28     }
29     },
30     "Next": "NEXT_STATE"
31     },
32     ██████████ ←
33   }
34 }
```

Permissions

Execution role

The IAM role that defines which resources your state machine has permission to access during execution. To create a custom role, go to the [IAM console](#) ↗

Create new role

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

31. Replace Choice State with: Is Transcribe Complete.

Servicess ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

Definition

Define your workflow using [Amazon States Language](#) ↗. Test your data flow with the new [Data Flow Simulator](#).

Export ▾ Layout ▾

Generate code snippet ▾ Format JSON

Workflow Studio New

```
26 +
27   "Payload": {
28     "Input.$": "$"
29   },
30   ██████████ ← "Next": "NEXT_STATE"
31   },
32   ██████████ ← "Choice State": {
33     "Type": "Choice",
34     "Choices": [
```

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

32. In the Check Transcribe Status state, change NEXT_STATE to: Is Transcribe Complete.



Services ▾



cloud_user @ 1885-6494-7610 ▾

N. Virginia ▾

Support ▾



Definition

Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

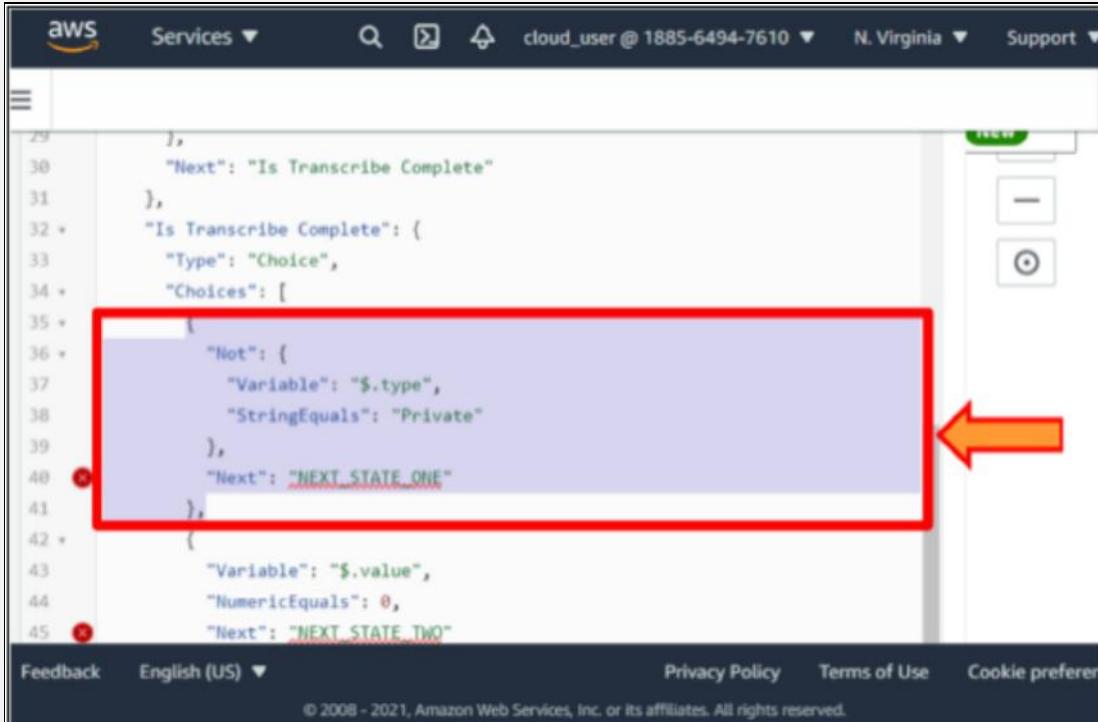
[Export ▾](#)[Layout ▾](#)[Generate code snippet](#) ▾[Format JSON](#)

```
28      }
29    },
30    "Next": "Is Transcribe Complete" ←
31  },
32 + "Is Transcribe Complete": {
33   "Type": "Choice",
34   "Choices": [
35     {
36       "Not": {
37         "Variable": "$.type",
38         "StringEquals": "Private"
39       },
40       "Next": "NEXT_STATE_ONE" ←
41     },
42     {
43       "Variable": "$.value",
44       "NumericEquals": 0,
45       "Next": "NEXT_STATE_TWO"
46     }
47   ]
48 }
```

[Workflow studio](#)[New](#)[Feedback](#) [English \(US\) ▾](#)[Privacy Policy](#) [Terms of Use](#)[Cookie preference](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

33. Under **Choices: [**, delete the **Not: {** block.



```
29      },
30    },
31    "Next": "Is Transcribe Complete"
32  },
33  "Is Transcribe Complete": {
34    "Type": "Choice",
35    "Choices": [
36      {
37        "Not": { ←
38          "Variable": "$.type",
39          "StringEquals": "Private"
40        },
41        "Next": "NEXT_STATE_ONE" ←
42      },
43      {
44        "Variable": "$.value",
45        "NumericEquals": 0,
46        "Next": "NEXT_STATE_TWO"
47      }
48    ]
49 }
```

[Feedback](#) [English \(US\) ▾](#)[Privacy Policy](#) [Terms of Use](#)[Cookie preference](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

34. Delete the **And: {** block.

```
38     "Next": "NEXT_STATE_TWO"
39   },
40   {
41     "And": [
42       {
43         "Variable": "$.value",
44         "NumericGreaterThanOrEqualTo": 20
45       },
46       {
47         "Variable": "$.value",
48         "NumericLessThan": 30
49       }
50     ],
51     "Next": "NEXT_STATE_THREE"
52   },
53   ],
54   "Default": "DEFAULT_STATE"
```

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

35. Copy the block under **Choices**: [and paste it under the existing block.

```
30   "Next": "Is Transcribe Complete"
31 },
32 "Is Transcribe Complete": {
33   "Type": "Choice",
34   "Choices": [
35     {
36       "Variable": "$.value",
37       "NumericEquals": 0,
38       "Next": "NEXT_STATE_TWO"
39     },
40   ],
41   "Default": "DEFAULT_STATE"
42 }
43 }
44 }
```

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

36. In the first Choices block, replace **NumericEquals** with **StringEquals** and replace **0** with **COMPLETED**.



```
32 "Is Transcribe Complete": {
33     "Type": "Choice",
34     "Choices": [
35         {
36             "Variable": "$.value",
37             "NumericEquals": 0, ←
38             "Next": "NEXT_STATE_TWO"
39         },
40         {
41             "Variable": "$.value",
42             "NumericEquals": 0, ←
43             "Next": "NEXT_STATE_TWO"
44         }
45     ],
46     "Default": "DEFAULT_STATE"
47 }
48 }
```

[Feedback](#)

English (US) ▾

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

37. In the second Choices block, replace **NumericEquals** with **StringEquals** and replace **0** with **FAILED**.

```
32 "Is Transcribe Complete": {
33     "Type": "Choice",
34     "Choices": [
35         {
36             "Variable": "$.value",
37             "StringEquals": "COMPLETED", ←
38             "Next": "NEXT_STATE_TWO"
39         },
40         {
41             "Variable": "$.value",
42             "StringEquals": "FAILED", ←
43             "Next": "NEXT_STATE_TWO"
44         }
45     ],
46     "Default": "DEFAULT_STATE"
47 }
48 }
```

[Feedback](#)

English (US) ▾

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

38. In the first Choices block, replace **\$.value** with: **\$.Payload.TranscriptionJobStatus**.
39. In the second Choices block, replace **\$.value** with: **\$.Payload.TranscriptionJobStatus**.



```
32 "Is Transcribe Complete": {  
33     "Type": "Choice",  
34     "Choices": [  
35         {  
36             "Variable": "$.Payload.TranscriptionJobStatus",  
37             "StringEquals": "COMPLETED",  
38             "Next": "NEXT_STATE_TWO"  
39         },  
40         {  
41             "Variable": "$.Payload.TranscriptionJobStatus",  
42             "StringEquals": "FAILED",  
43             "Next": "NEXT_STATE_TWO"  
44         }  
45     ],  
46     "Default": "DEFAULT_STATE"  
47 }  
48 }
```

[Feedback](#)

English (US) ▾

[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

40. Replace DEFAULT_STATE with: Wait for Transcribe

```
40     {  
41         "Variable": "$.Payload.TranscriptionJobStatus",  
42         "StringEquals": "FAILED",  
43         "Next": "NEXT_STATE_TWO"  
44     }  
45 ],  
46     "Default": "Wait for Transcribe"  
47 }  
48 }  
49 }
```

Permissions

[Feedback](#)

English (US) ▾

[Privacy Policy](#)

[Terms of Use](#)

[Cookie preferences](#)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

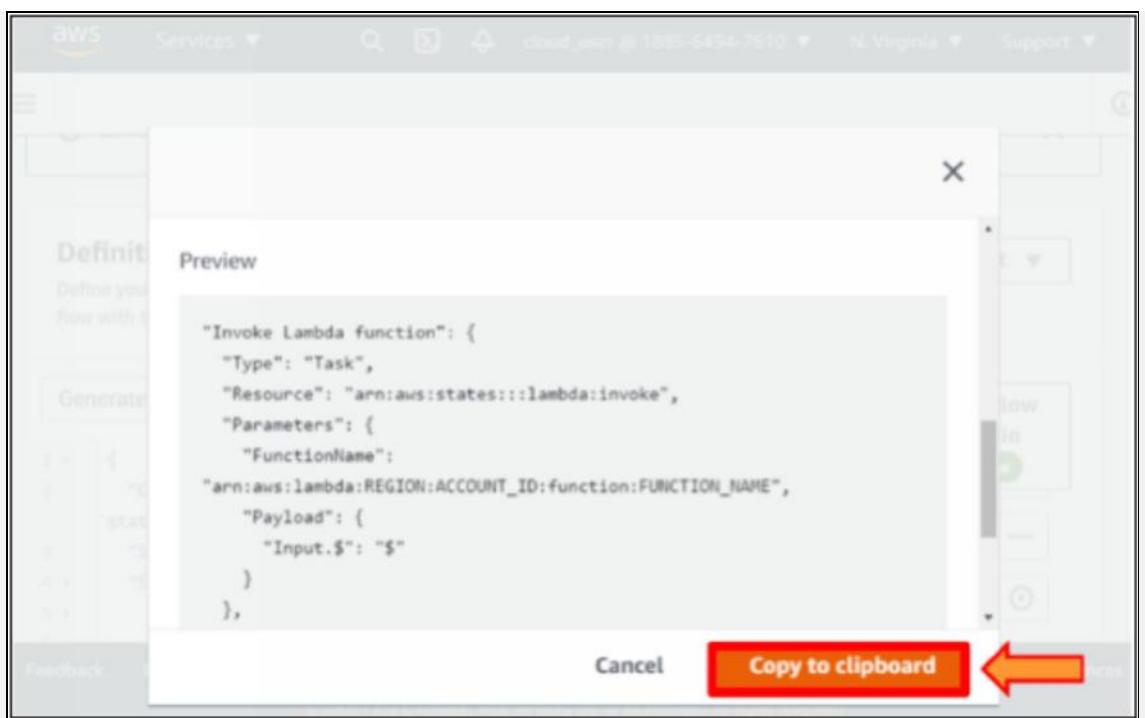
41. In the Generate code snippet dropdown, select AWS Lambda: Invoke a function.

The screenshot shows the AWS Lambda Invoke a function interface. At the top, there's a search bar and navigation links for Services, cloud_user@1885-6494-7610, N. Virginia, and Support. Below the search bar, there are buttons for 'Export' and 'Layout'. A red box highlights the 'AWS Lambda: Invoke a function' button, which has an orange arrow pointing to it. Another red box highlights the 'Generate code snippet' button, also with an orange arrow pointing to it. The main area contains a JSON code editor with some sample code. At the bottom, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences, along with a copyright notice: © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

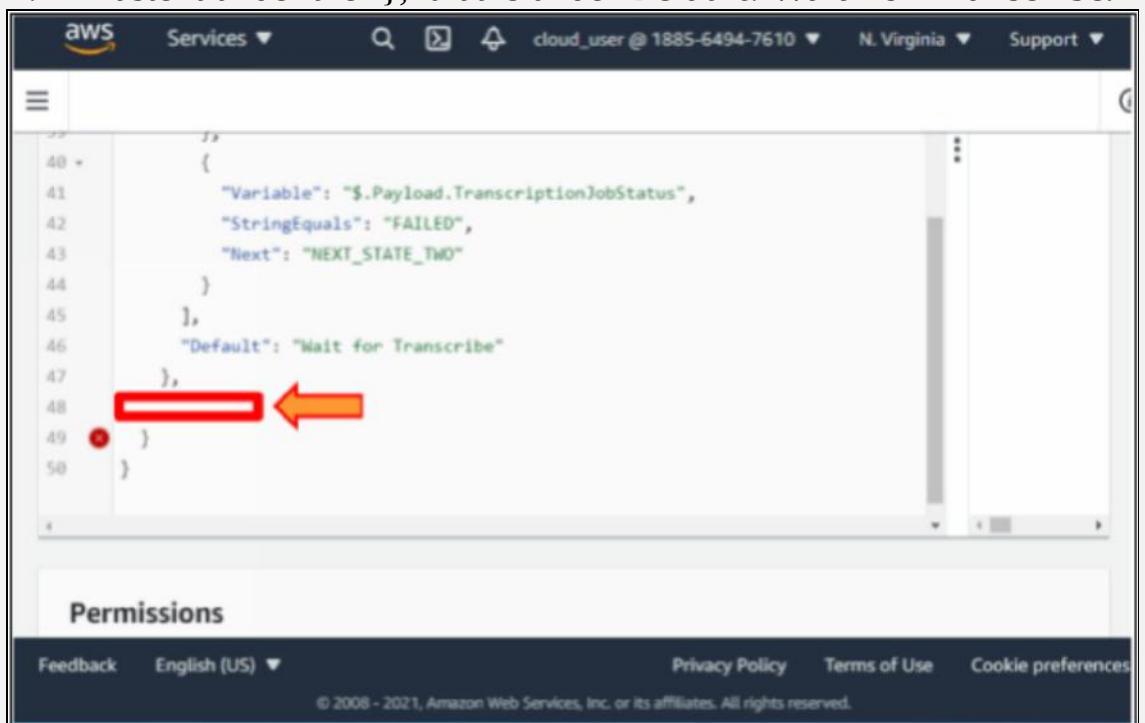
42. In the Select function from a list dropdown menu. Select the **categorize-data-lambda** function.

The screenshot shows a modal dialog for selecting a Lambda function. The title is 'Lambda function'. It says 'The Lambda function to invoke.' Below that is a dropdown menu labeled 'Select function from a list'. Underneath is another dropdown menu labeled 'Select function' with a search bar containing 'lambda:\$LATEST'. A red box highlights the text 'arn:aws:lambda:us-east-1:188564947610:function:categorize-data-lambda:\$LATEST' in the search bar, with an orange arrow pointing to it. Below the search bar is a note: 'Enter reference path, for ex: \$.input.myParam.value'. At the bottom of the dialog are 'Cancel' and 'Copy to clipboard' buttons, with the 'Copy to clipboard' button being orange.

43. Click on the **Copy to clipboard** button.



44. Paste it under the '},' that is under Default: Wait For Transcribe.



45. Click on the Format JSON button.

Salesforce

Services ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

Definition

Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Generate code snippet ▾ Format JSON

```
1 <  {
2     "Comment": "A Hello World example of the Amazon States Language using Pass states",
3     "StartAt": "Start Transcription",
4     "States": {
5         "Start Transcription": {
6             "Type": "Task",
```

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

46. Rename Invoke Lambda function to Categorize Data.

Salesforce

Services ▾

cloud_user @ 1885-6494-7610 ▾ N. Virginia ▾ Support ▾

```
35 <  {
36     "Variable": "$.Payload.TranscriptionJobStatus",
37     "StringEquals": "COMPLETED",
38     "Next": "NEXT_STATE_TWO"
39 },
40 {
41     "Variable": "$.Payload.TranscriptionJobStatus",
42     "StringEquals": "FAILED",
43     "Next": "NEXT_STATE_TWO"
44 }
45 ],
46 "Default": "Wait for Transcribe"
47 },
48 "Invoke Lambda function": (
49     "Type": "Task",
50     "Resource": "arn:aws:states:::lambda:invoke",
```

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

47. In the Completed block for Choices, replace **NEXT_STATE_TWO** with **Categorize Data**.

Screenshot of the AWS Lambda function editor showing the state machine code. Two specific lines are highlighted with red boxes and arrows:

```
35 *     {
36         "Variable": "$.Payload.TranscriptionJobStatus",
37         "StringEquals": "COMPLETED",
38         "Next": "Categorize Data" ←
39     },
40 *
41     {
42         "Variable": "$.Payload.TranscriptionJobStatus",
43         "StringEquals": "FAILED",
44         "Next": "NEXT_STATE_TWO" ←
45     },
46     "Default": "Wait for Transcribe"
47 },
48 *
49     "Categorize Data": {
50         "Type": "Task",
51         "Resource": "arn:aws:states:::lambda:invoke",
52     }
53 }
```

The first highlighted line is "Next": "Categorize Data". The second highlighted line is the opening brace of the "Categorize Data" block: "Categorize Data": {.

48. In the Categorize Data block, replace **Next: NEXT_STATE** with:
End: true

Screenshot of the AWS Lambda function editor showing the state machine code after modification. The "End: true" line is highlighted with a red box and arrow:

```
45     ],
46     "Default": "Wait for Transcribe"
47 },
48 *
49     "Categorize Data": {
50         "Type": "Task",
51         "Resource": "arn:aws:states:::lambda:invoke",
52         "Parameters": {
53             "FunctionName": "arn:aws:lambda:us-east-
1:188564947610:function:categorize-data-lambda:$LATEST",
54             "Payload": {
55                 "Input.$": "$"
56             },
57             "End": true ←
58         }
59 }
```

The highlighted line is "End": true.

49. In the **Generate code snippet** dropdown, select the **Fail state**.



Services ▾



cloud_user @ 1885-6494-7610 ▾

N. Virginia ▾

Support ▾



Definition

Export ▾

Layout ▾

Define your workflow using Amazon States Language [↗](#). Test your data flow with the new [Data Flow Simulator](#).

Generate code snippet ▾

JSON



Stops an execution successfully

Fail state

Stops an execution and mark it as a failure

Intrinsic functions

```
9   "FunctionName": "arn:aws:lambda:us-east-  
10  1:188564947610:function:transcribe-audio-lambda:$LATEST",  
11  100% 1 sec 100% 1 sec 100%
```

Feedback

English (US) ▾

Privacy Policy

Terms of Use

Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

50. Click on the Copy to clipboard button.

The screenshot shows the AWS Lambda function configuration page. The 'Code' tab is selected. A large orange arrow points to the 'Edit' button, which is located next to the 'Upload' and 'Lambda@Edge' buttons. Below the code editor, there is a preview window showing the JSON configuration for a fail state. At the bottom right of the preview window, there is a red box around the 'Copy to clipboard' button, with an orange arrow pointing to it.

Preview

```
"Fail State": {  
    "Type": "Fail",  
    "Error": "ErrorCode",  
    "Cause": "Caused By Message"  
}
```

Cancel **Copy to clipboard**

51. Paste it under the }, found under End: true.

AWS Services Search cloud_user @ 1885-6494-7610 N. Virginia Support

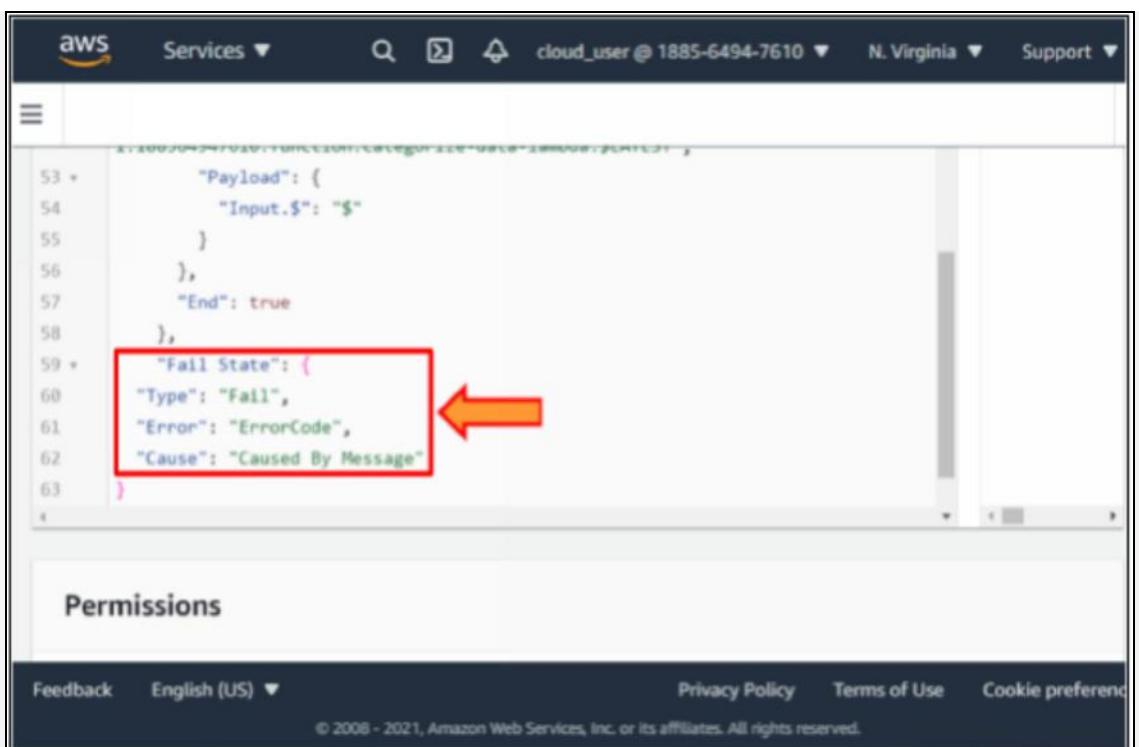
```
53 +
  "Payload": {
    "Input.$": "$"
  }
},
"End": true
),
59 +
  "Fail State": {
    "Type": "Fail",
    "Error": "ErrorCode",
    "Cause": "Caused By Message"
}

```

Permissions

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



52. Click on the **Format JSON** button.

AWS Services Search cloud_user @ 1885-6494-7610 N. Virginia Support

Definition

Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Export ▾ Layout ▾

Generate code snippet ▾ Format JSON

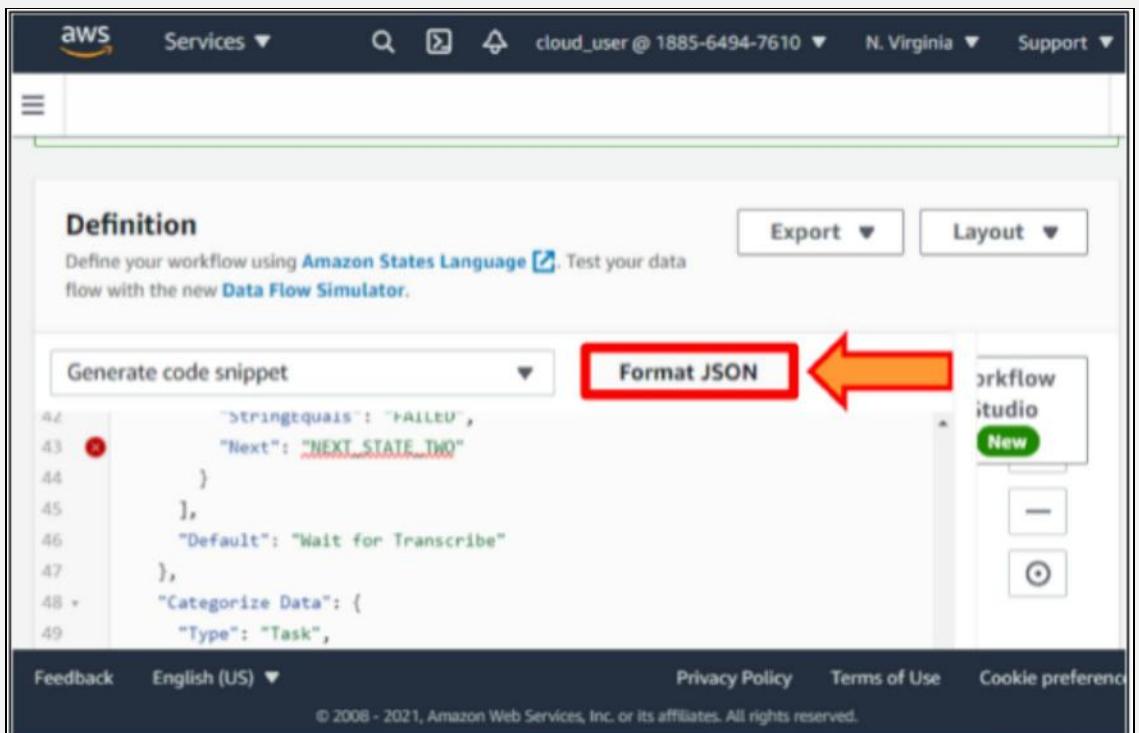
```
42
43   "Stringequals": "FAILED",
44     "Next": "NEXT_STATE_TWO"
45   },
46   "Default": "Wait for Transcribe"
47 },
48 +
  "Categorize Data": {
    "Type": "Task",

```

Workflow Studio New

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



53. Replace **Fail State** with **Transcribe Failed**.
54. Delete the **Error** and **Cause** lines.



Services ▾



cloud_user @ 1885-6494-7610 ▾

N. Virginia ▾

Support ▾



```
    "End": true
  },
  "Transcribe Failed": {
    "Type": "Fail",
    "Error": "ErrorCode",
    "Cause": "Caused By Message"
  }
}
```

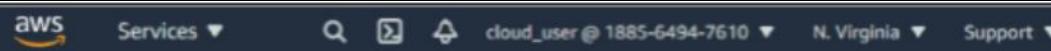


Permissions

[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

55. In the Failed block for Choices, replace **NEXT_STATE_TWO** with: **Transcribe Failed**.

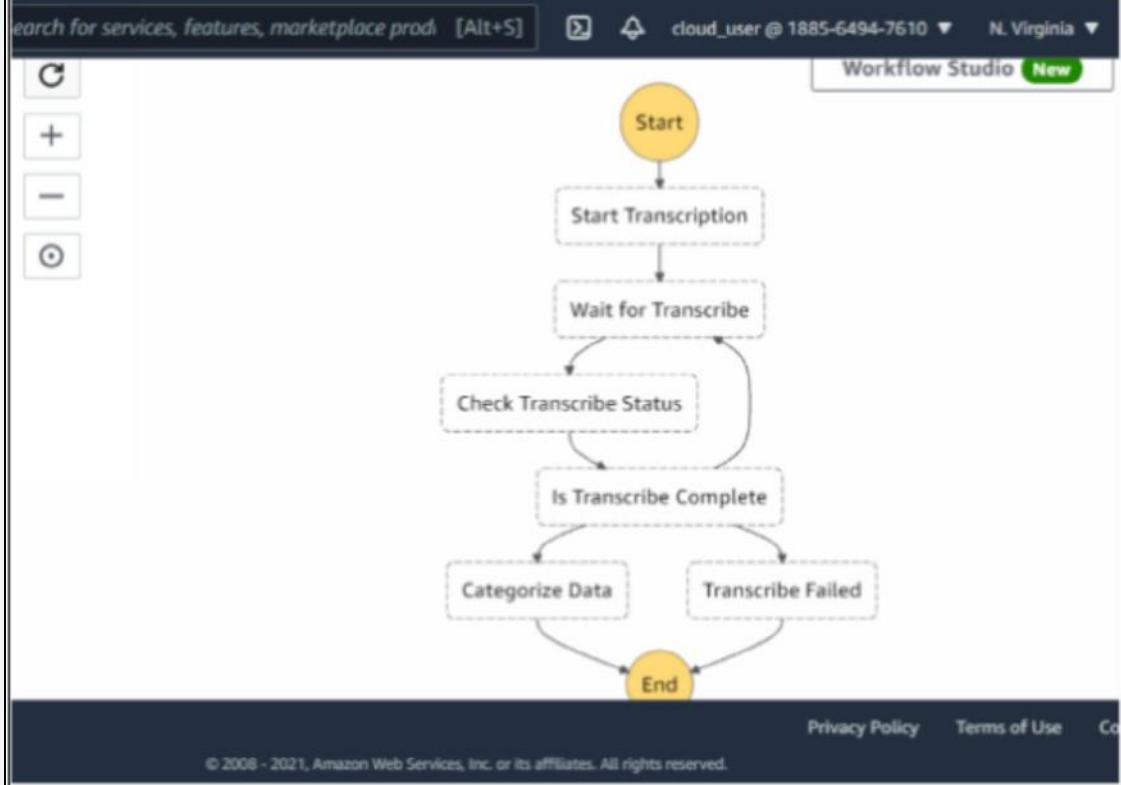


Permissions

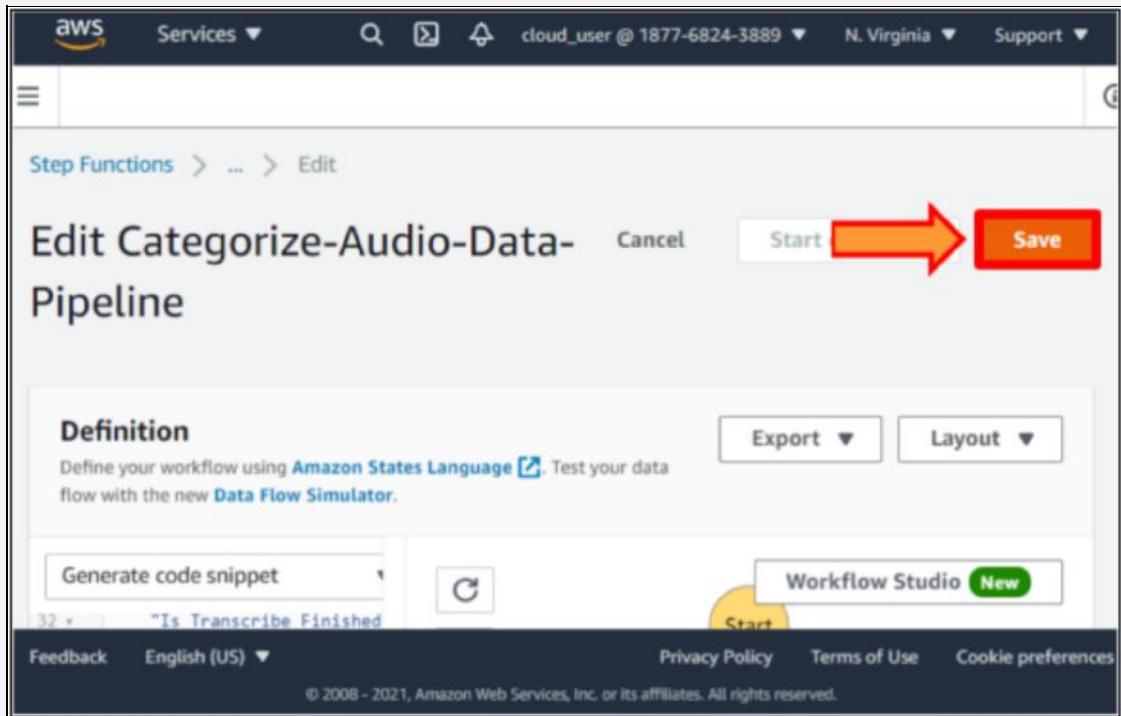
[Feedback](#)[English \(US\) ▾](#)[Privacy Policy](#)[Terms of Use](#)[Cookie preferences](#)

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

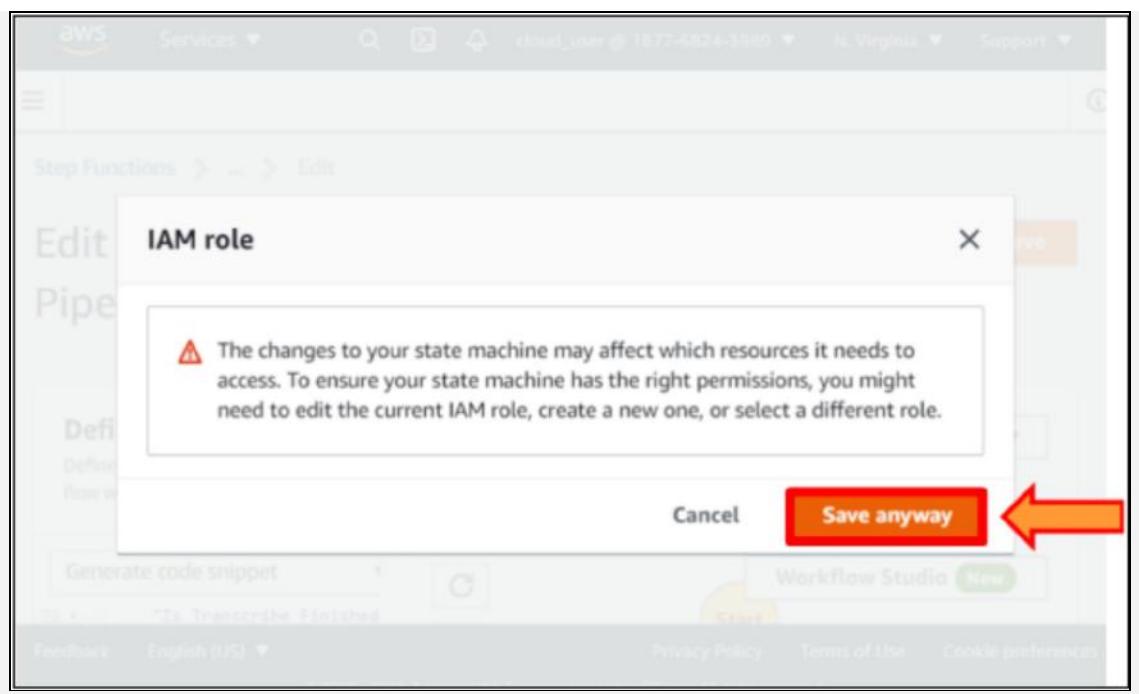
56. After that, you will see the block graph of the model.



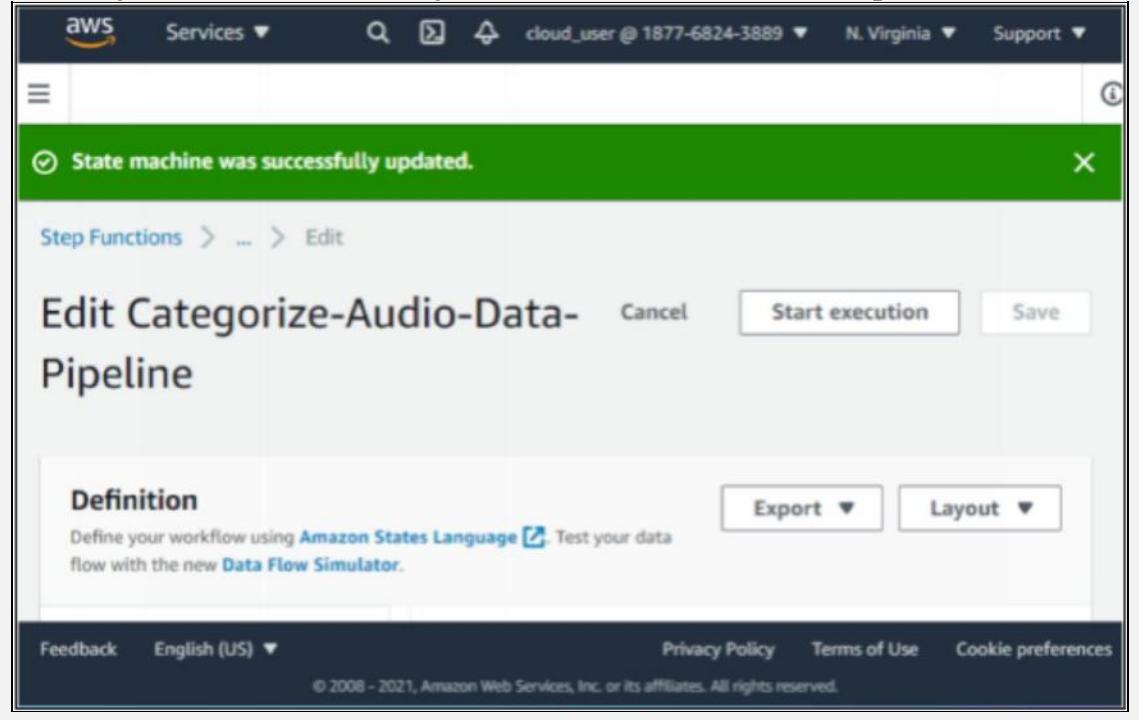
57. Click on the **Save** button.



58. Click on the **Save anyway** button.



Hence, you have successfully created the model with Step Function.



Step 6: Create the Lambda Business Logic

1. Go back to the **Lambda Function** dashboard.
2. Write the Transcribe Audio Function.
3. Click on the **transcribe-audio-lambda**.

The screenshot shows the AWS Lambda Functions dashboard. A red box highlights the row for the 'transcribe-audio-lambda' function. An orange arrow points from the right towards the highlighted row. The table columns include name, size, and last modified time. The 'transcribe-audio-lambda' row is the fourth one from the top.

name	size	last modified
transcribe-status-lambda	377.0 byte	32 min ago
run-step-functions-lambda	426.0 byte	5 min ago
categorize-data-lambda	854.0 byte	32 min ago
transcribe-audio-lambda	497.0 byte	32 min ago

4. Click on the **index.py**.

The screenshot shows the AWS Lambda function editor for the 'transcribe-audio-lambda' function. A red box highlights the 'index.py' file tab. An orange arrow points from the left towards the highlighted file tab. The code editor displays the contents of the index.py file, which includes imports for boto3 and uuid, and a lambda_handler function that starts a transcription job.

```
1 import boto3
2 import uuid
3
4 transcribe_client = boto3.client('transcribe')
5
6 def lambda_handler(event, context):
7     #Retrieve the state information
8     step_state = event['Input']
9     s3_bucket = '' #TODO
10    s3_audio_key = '' #TODO
11
12    #Prepare the parameters needed for the Transcribe job
13    meeting_audio_URI = '' #TODO
14    jobName = '' #TODO
15    transcript_key = '' #TODO
16
17    #Start the transcription job
18    response = transcribe_client.start_transcription_job(
```

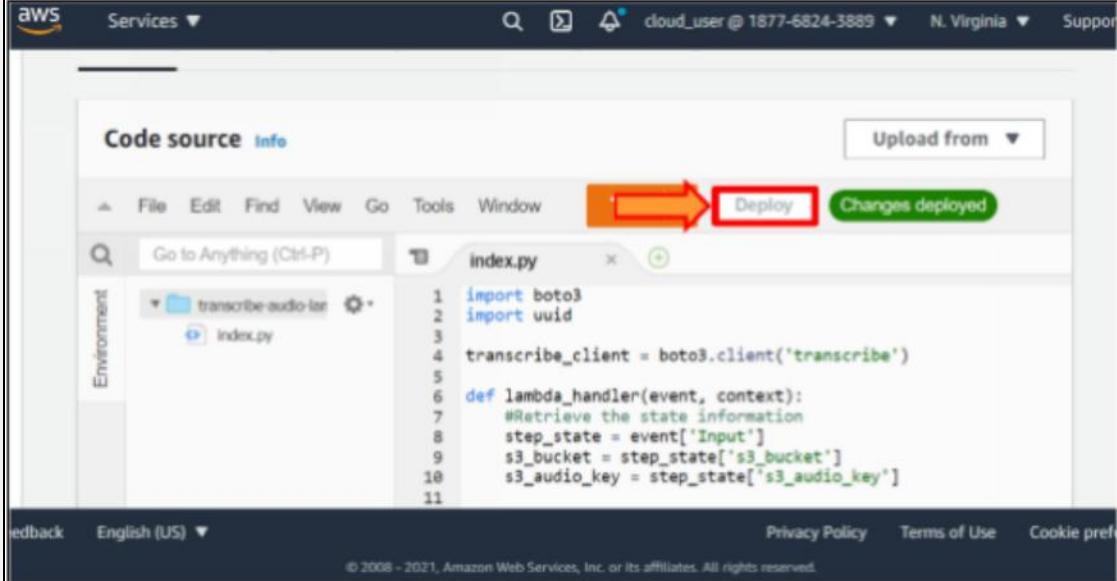
5. Copy the code from the following Github link:
<https://github.com/linuxacademy/content-aws-mls-c01/blob/master/CategorizeDataUploadsUsingStepFunctions/transcribe-audio-lambda.py>

30 lines (24 sloc) | 1.01 KB

...

```
1 import boto3
2 import uuid
3
4 transcribe_client = boto3.client('transcribe')
5
6 def lambda_handler(event, context):
7     #Retrieve the state information
8     step_state = event['Input']
9     s3_bucket = step_state['s3_bucket']
10    s3_audio_key = step_state['s3_audio_key']
11
12    #Prepare the parameters needed for the Transcribe job
13    meeting_audio_URI = f's3://{s3_bucket}/{s3_audio_key}'
14    jobName = f'{s3_audio_key}-{str(uuid.uuid4())}'.replace('/', '-')
15    transcript_key = f'transcripts/{s3_audio_key}-transcript.json'
16
17    #Start the transcription job
18    response = transcribe_client.start_transcription_job(
19        TranscriptionJobName=jobName,
20        LanguageCode='en-US',
```

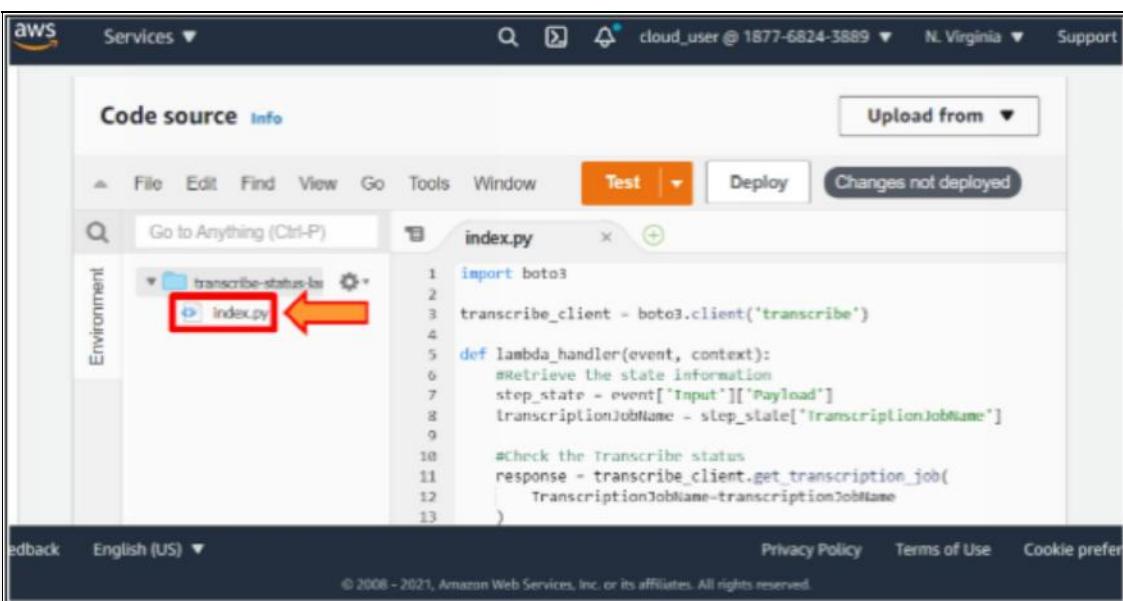
6. Paste the code into the Lambda Function code editor.
7. Click on the **Deploy** button to save the changes.



8. Write the Transcribe Status Function.
9. Click on the **transcribe-status-lambda**.

Function name	Description	Package type	Runtime	Code size	Last modified
transcribe-status-lambda	-	Zip	Python 3.8	377.0 byte	39 minutes ago
run-step-functions-lambda	-	Zip	Python 3.8	426.0 byte	13 minutes ago
categorize-data-lambda	-	Zip	Python 3.8	854.0 byte	39 minutes ago
transcribe-audio	-	Zip	Python 3.8	560.0 byte	1 minute

10. Click on the **index.py**.



11. Copy the code from the following Github link:
<https://github.com/linuxacademy/content-aws-mls-c01/blob/master/CategorizeDataUploadsUsingStepFunctions/transcribe-status-lambda.py>

The screenshot shows a GitHub code viewer displaying the 'transcribe-status-lambda.py' file. The code is identical to the one shown in the AWS Lambda editor. It contains 18 lines of Python code, totaling 560 bytes. The code is as follows:

```
1 import boto3
2
3 transcribe_client = boto3.client('transcribe')
4
5 def lambda_handler(event, context):
6     #Retrieve the state information
7     step_state = event['Input']['Payload']
8     transcriptionJobName = step_state['TranscriptionJobName']
9
10    #Check the Transcribe status
11    response = transcribe_client.get_transcription_job(
12        TranscriptionJobName=transcriptionJobName
13    )
14
15    #Add the Transcribe job status back to the state
16    step_state['TranscriptionJobStatus'] = response['TranscriptionJob']['TranscriptionJobStatus']
17
18    return step_state
```

12. Paste the code into the Lambda Function code editor.
13. Click on the Deploy button to save the changes.

The screenshot shows the AWS Lambda code editor interface. At the top, there's a navigation bar with 'Services', a search bar, and account information ('cloud_user @ 1877-6824-3889', 'N. Virginia', 'Support'). Below the navigation is a toolbar with 'File', 'Edit', 'Find', 'View', 'Go', 'Tools', 'Window', 'Upload from', 'Deploy' (which has a red arrow pointing to it), and 'Changes deployed'. A status bar at the bottom indicates 'Feedback English (US)' and copyright information ('© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'). The main area shows a file named 'index.py' with the following code:

```
import boto3
transcribe_client = boto3.client('transcribe')
def lambda_handler(event, context):
    #Retrieve the state information
    step_state = event['Input']['Payload']
    transcriptionJobName = step_state['TranscriptionJobName']
    #Check the Transcribe status
    response = transcribe_client.get_transcription_job(
        TranscriptionJobName=transcriptionJobName
    )
```

14. Write the Categorize Data Function.
15. Click on the **categorize-data-lambda**.

The screenshot shows a list of AWS Lambda functions in the AWS Lambda console. The functions listed are:

name	size	modified
transcribe-status-lambda	367.0 byte	1 minute ago
run-step-functions-lambda	426.0 byte	15 minutes ago
categorize-data-lambda	854.0 byte	41 minutes ago
transcribe-audio-lambda	560.0 byte	3 minutes ago

16. Click on the **index.py**.

The screenshot shows the AWS Lambda code editor interface again. The 'Deploy' button is now greyed out ('Changes not deployed'). The 'Test' button is highlighted. The file list shows 'index.py' selected (indicated by a red box and a red arrow). The code in 'index.py' is:

```
import boto3
import json
import os
import uuid
from datetime import date
s3_client = boto3.client('s3')
def lambda_handler(event, context):
    #Retrieve the state information
    step_state = event["Input"]["Payload"]
```

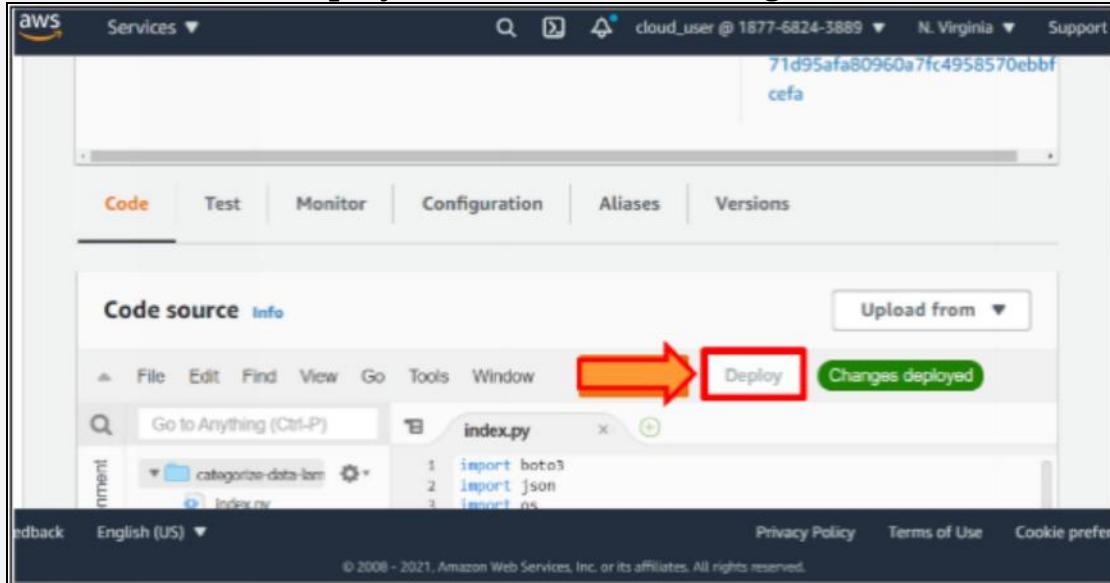
17. Copy the code from the following Github link:
<https://github.com/linuxacademy/content-aws-mls-c01/blob/master/CategorizeDataUploadsUsingStepFunctions/categorize-data-lambda.py>

66 lines (54 sloc) | 2.36 KB

...

```
1 import boto3
2 import json
3 import os
4 import uuid
5 from datetime import date
6
7 s3_client = boto3.client('s3')
8
9 def lambda_handler(event, context):
10     #Retrieve the state information
11     step_state = event['Input']['Payload']
12
13     #Get the S3 information for the audio file and transcript
14     s3_bucket = step_state['s3_bucket']
15     s3_audio_key = step_state['s3_audio_key']
16     s3_transcript_key = step_state['transcript_key']
17
18     base_audio_key = os.path.basename(s3_audio_key)
19     base_transcript_key = os.path.basename(s3_transcript_key)
20
```

18. Paste the code into the Lambda Function code editor.
19. Click on the **Deploy** button to save the changes.



20. Click on the **Configuration** tab.
21. Click on the **Environment variables**.
22. Click on the **Edit** button.

The screenshot shows the AWS Lambda Configuration page. The top navigation bar includes 'Services' (dropdown), 'Code', 'Test', 'Monitor', 'Configuration' (highlighted with a red box and arrow), and 'Versions'. On the left sidebar, 'General configuration', 'Triggers', 'Permissions', 'Destinations', and 'Environment variables' (highlighted with a red box and arrow) are listed. The main content area is titled 'Environment variables (0)' with a 'Edit' button. It displays a table with columns 'Key' and 'Value', stating 'No environment variables' and 'No environment variables associated with this function.' A second red arrow points to the 'Edit' button in the table row.

23. Click on the Add environment variable button.

The screenshot shows the 'Edit environment variables' page. The title is 'Edit environment variables'. Below it is a section titled 'Environment variables' with a descriptive text about defining environment variables as key-value pairs. A large red arrow points to the 'Add environment variable' button at the bottom left of the page.

24. Under Key, enter **KEYWORDS**, and under Value, enter **important**.

The screenshot shows the 'Edit environment variables' page with two red arrows. One arrow points to the 'Key' input field containing 'KEYWORDS', and another arrow points to the 'Value' input field containing 'important'.

25. Click on the Save button.

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key	Value	Remove
KEYWORDS	important	

Add environment variable

Encryption configuration

Save

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 7: Categorize Audio Data

1. Go back to the S3 Bucket console.
2. Click on the bucket.

Buckets (1) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

C	Copy ARN	Empty	Delete	Create bucket
---	----------	-------	--------	-------------------------------

Find buckets by name

< 1 > ⌂

Name	AWS Region	Access	Creation date
meeting-audio-21e062b0	US East (N. Virginia) us-east-1	Bucket and objects not public	October 13, 2021, 23:58:33 (UTC-07:00)

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

3. Click on the **Create folder** button.

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

< 1 > ⌂

Name	Type	Last modified	Size	Storage class
No objects				

You don't have any objects in this bucket.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4. Name the folder **Upload**.

appropriate settings:

Folder

Folder name /

Folder names can't contain "/". See [rules for naming](#)

Server-side encryption

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Click on the **Create folder** button.

aws Services ▾

cloud_user @ 1877-6824-3889 ▾ Global ▾ Support ▾

Server-side encryption

The following settings apply only to the new folder object and not to the objects contained within it.

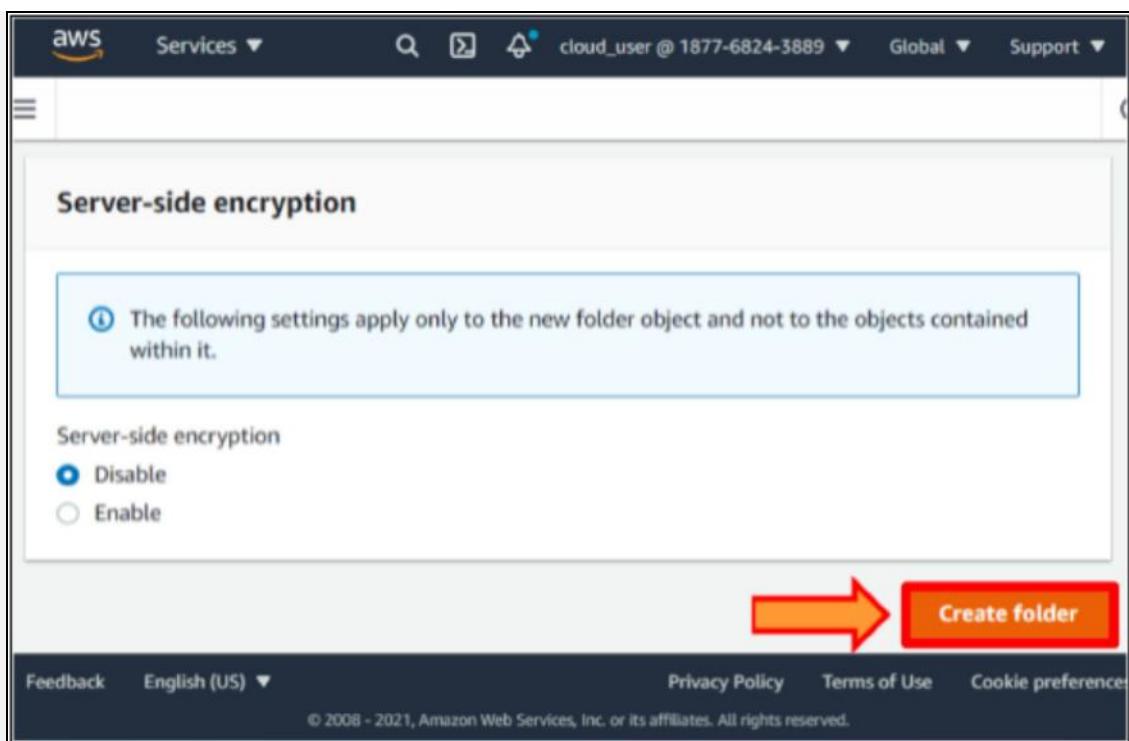
Server-side encryption

Disable
 Enable

 **Create folder**

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



6. Click on the **upload/** folder.

aws Services ▾

cloud_user @ 1877-6824-3889 ▾ Global ▾ Support ▾

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

 **upload/** Folder

Name ▲ **Type** ▼ **Last modified** ▼ **Size** ▼ **Storage class** ▼

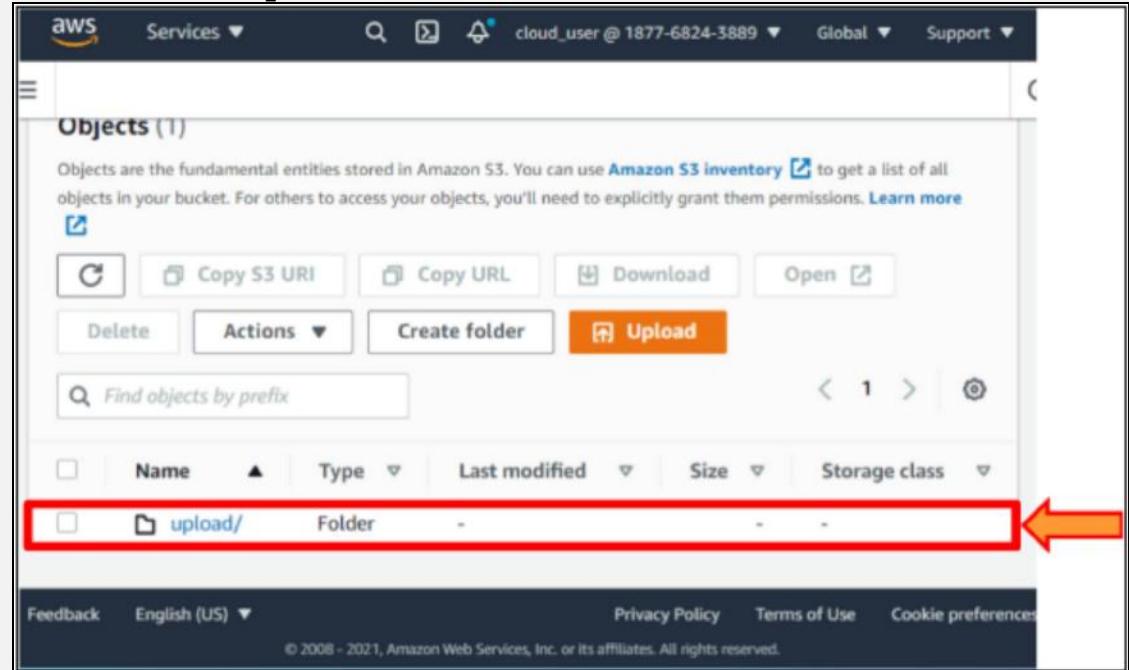
 Copy S3 URI  Copy URL  Download Open  Upload

Find objects by prefix

< 1 > 

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



7. Click on the **Upload** button.

Screenshot of the AWS S3 console showing a folder named "ImportantBusiness". The "Upload" button is highlighted with a red box and a large orange arrow pointing to it.

No objects

You don't have any objects in this folder.

Upload

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. Click on the **Add files** button.

Screenshot of the AWS S3 "Add files" page. The "Add files" button is highlighted with a red box and a large orange arrow pointing to it.

Files and folders (0)

All files and folders in this table will be uploaded.

Add files

Find by name

No files or folders

You have not chosen any files or folders to upload.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Select an mp3 file to upload. Download the `ImportantBusiness.mp3` file from the GitHub link: <https://github.com/linuxacademy/content-aws-mls-c01/tree/master/CategorizeDataUploadsUsingStepFunctions>

The screenshot shows the AWS S3 'Files and folders' page. A single file, 'CategorizeDataUploadsUsingStepFunctions_ImportantBusiness.mp3', is listed in the table. This file is highlighted with a red box and has a yellow arrow pointing to its row. The file details are: Name: CategorizeDataUploadsUsingStepFunctions_ImportantBusiness.mp3, Folder: -, Type: audio/mpeg, Size: 32.5 KB.

	Name	Folder	Type	Size
<input type="checkbox"/>	CategorizeDataUploadsUsingStepFunctions_ImportantBusiness.mp3	-	audio/mpeg	32.5 KB

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10. Click on the **Upload** button.

The screenshot shows the 'Upload' configuration page. It includes sections for 'Destination' (set to s3://meeting-audio-c768fab0/upload/) and 'Permissions'. A large yellow arrow points to the 'Upload' button at the bottom right of the page.

Destination
s3://meeting-audio-c768fab0/upload/

▶ Destination details
Bucket settings that impact new objects stored in the specified destination.

▶ Permissions
Grant public access and access to other AWS accounts.

▶ Properties
Specify storage class, encryption settings, tags, and more.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11. Go back to the **Step Function** dashboard. The step function should now be running.

Sales

Services ▾

cloud_user @ 1877-6824-3889 N. Virginia Support

Executions (1)

C View details Stop execution Start execution

Search for executions Filter by status

< 1 > ⚙

Name	Status	Started	End Time
5d3daa6c-7271-463a-9de7-78f51e07f6e0	Running	Oct 14, 2021 02:19:37.213 AM	-

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5d3daa6c-7271-463a-9de7-78f51e07f6e0

Running

Oct 14, 2021 02:19:37.213 AM

12. Click on the step function name. You will see the model graph.

Sales

Services ▾

cloud_user @ 1877-6824-3889 N. Virginia Support

Step Functions X

State machines Activities Data flow simulator Feature spotlight Local Development Join our feedback panel

```
graph TD; Start((Start)) --> Transcribe[Transcribe Audio]; Transcribe --> Wait[Wait for Transcribe]; Wait --> Check[Check Transcribe Status]; Check --> Is{Is Transcribe Finished}; Is -- No --> Check; Is -- Yes --> Categorize[Categorize Data]; Categorize --> End((End)); TranscribeFailed[Transcribe Failed] --> End;
```

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

13. Click Step input to view the input.

Sales

Services ▾

cloud_user @ 1877-6824-3889 N. Virginia Support

Step Functions X

Start Transcribe Audio Wait for Transcribe Check Transcribe Status Is Transcribe Finished Categorize Data Transcribe Failed End

Details Step input

```
1: {
2:   "ExecutedVersion": "$LATEST",
3:   "Payload": {
4:     "s3_bucket": "meeting-audio-c768fab0",
5:     "s3_audio_key": "upload/CategorizeDataUploads_ImportantBusiness.mp3",
6:     "transcript_key": "transcripts/upload/CategorizeDataUsingStepFunctions_ImportantBusiness.mp3-transcript",
7:     "TranscriptionJobName": "upload-CategorizeDataUsingStepFunctions_ImportantBusiness.mp3-27926ee0-994df3d73d9",
8:     "TranscriptionJobStatus": "COMPLETED"
9:   },
10:  "SdkHttpMetadata": {
```

Feedback English (US) Privacy Policy Terms of Use Cookie preferences

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Start

Transcribe Audio

Wait for Transcribe

Check Transcribe Status

Is Transcribe Finished

Categorize Data

Transcribe Failed

End

14. Click Step output to view the output of the step function.



15. Go back to the S3 Bucket dashboard.
16. Click on the **important/** folder.

The screenshot shows the AWS S3 Bucket dashboard. In the "Objects (3)" section, there is a single folder named "important/". A red arrow points to this folder.

Name	Type	Last modified	Size	Storage class
important/	Folder	-	-	-

17. Click the **year, month, and day** folder and confirm the mp3 and transcript are saved correctly.

The screenshot shows the AWS S3 Bucket dashboard. A specific folder is highlighted with a red box. Inside this folder, two files are listed: "CategorizeDataUploadsUsingStepFunctions_ImportantBusiness.mp3" and "CategorizeDataUploadsUsingStepFunctions_ImportantBusiness.mp3-transcript.json". A red arrow points to this folder.

18. Go back to the Step Function tab.
19. Scroll down the page to find the execution event history.

The screenshot shows the AWS Step Functions Execution event history. A red box highlights the last three rows of the table, which represent the execution steps. An orange arrow points to the timestamp column of the third row, specifically to the date and time "Oct 14, 2021 02:19:37.252 AM".

ID	Type	Step	Resource	Elapsed Time (ms)	Timestamp
1	ExecutionStarted	-	-	0	Oct 14, 2021 02:19:37.213 AM
2	TaskStateEntered	Transcribe Audio	-	39	Oct 14, 2021 02:19:37.252 AM
3	TaskScheduled	Transcribe Audio	Lambda CloudWatch Logs	39	Oct 14, 2021 02:19:37.252 AM

Execution event history

Services ▾ Search Support

cloud_user @ 1877-6824-3889 N. Virginia

Privacy Policy Terms of Use Cookie preferences

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lab 16: Coordinating AI Services with Step Functions

Certifications	AWS Machine Learning - Specialty
Cloud Provider	AWS
Time Required	1 Hr.
Tags	Lambda, AWS

Lab Prerequisites

- Familiarity with basic AWS Cloud Computing concepts and terminology
- An AWS account with an active subscription

Service Introduction

AWS Lambda

AWS Lambda allows you to run code without creating or managing servers. There is no charge when your code is not executing; you only pay for the compute time you use. You can run code for nearly any application or backend service with Lambda, and you do not have to worry about administration. Upload your code, and Lambda will handle everything necessary to run and grow it with high availability. You may configure your code to be automatically triggered by other AWS services, or you can access it directly from any computer or smartphone app.

AWS Simple Storage S3

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 provides a simple web service interface for storing and retrieving any quantity of data from any place at any time. You may quickly create projects that integrate cloud-native storage using this service. Because Amazon S3 is easily customizable and you only pay for what you use, you can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is also built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. They can build a simple FTP system or a complex web application like the amazon.com retail website; read the same piece of data a million times or only for emergency disaster recovery; store whatever type and amount of data desired.

AWS Translate

Amazon Translate is a text translation service that uses Neural Machine Translation (MT) to translate text between supported languages. The service, powered by deep learning technologies, delivers high-quality, inexpensive, and configurable language translation, allowing developers to solve business and

user-authored content and construct applications that require multilingual support. The service can be accessed using an API, which allows for real-time or batch translation of the text from one language to another.

AWS Transcribe

Conduct a (text-based) content analysis on audio/video content using Automatic Speech Recognition (ASR) technology. Amazon Transcribe is an AWS service that makes converting speech to text simple for customers. Customers may utilize Amazon Transcribe for various commercial applications, such as voice-based customer care call transcription and the production of subtitles for audio/video material.

AWS Comprehend

Amazon Comprehend is a Natural Language Processing (NLP) service that searches the text for meaning and insights using machine learning. You can use Amazon Comprehend to extract key terms, places, people, brands, or events from a library of documents, understand sentiment about products or services, and identify the primary topics. This text could have come from various places, including online pages, social media feeds, emails, and newspapers. You can also input Amazon Comprehend, a collection of text documents. It will pick subjects (or groups of words) that best reflect the data. The output of Amazon Comprehend may be used to evaluate consumer comments better, provide a search experience via search filters, and arrange themes.

AWS Polly

Polly is an Amazon service that converts text into natural-sounding speech. Amazon Polly enables existing apps to speak as a first-class feature and new categories of speech-enabled goods, such as mobile apps, automobiles, gadgets, and appliances. Amazon Polly comes with dozens of lifelike voices and multilingual capabilities, allowing you to pick the perfect voice and distribute your speech-enabled apps across many locations. Amazon Polly is simple to use: transmit the text you want to be translated into speech to the Amazon Polly API. Amazon Polly will return the audio stream to your application. You can play or save it in a regular audio file format like MP3. You can change the speech pace, pitch, or volume using Amazon Polly because it supports Speech Synthesis Markup Language (SSML) tags like prosody. Amazon Polly is a safe service that provides these advantages at low latency and on a large scale. Amazon Polly's generated speech can be cached and replayed at no extra charge. When you sign up for Amazon Polly, you can convert millions of characters per month for free for the first year. Amazon Polly is a cost-effective solution to enable speech synthesis everywhere because of its pay-as-you-go pricing, low cost per request, and lack of constraints on storage and reuse of voice output. You can utilize Amazon Polly to provide high-quality spoken work for your app. This low-cost service provides quick response times and may be used for almost any application, with no constraints on storing and reusing created speech.

Amazon Step Function

AWS Step Functions is a fully managed solution that makes it simple to coordinate the components of distributed applications and microservices using visual workflows. You may easily grow and alter applications by creating them from separate components that execute a specific function. Step Functions is a dependable method for coordinating components and stepping through your application's functions. Step Functions provides a graphical interface for organizing and visualizing your application's components as a series of stages. It makes creating and running multi-step apps a breeze. Step Functions automates the triggering and tracking of each step and retries when problems

occur, ensuring that your application runs smoothly. Step Functions records the condition of each step so that you can immediately detect and troubleshoot problems if something goes wrong. You can alter and add stages without writing code, allowing you to grow and innovate your application quickly.

Case Study Enterprise Internet Television Network – Netflix

Business Challenge

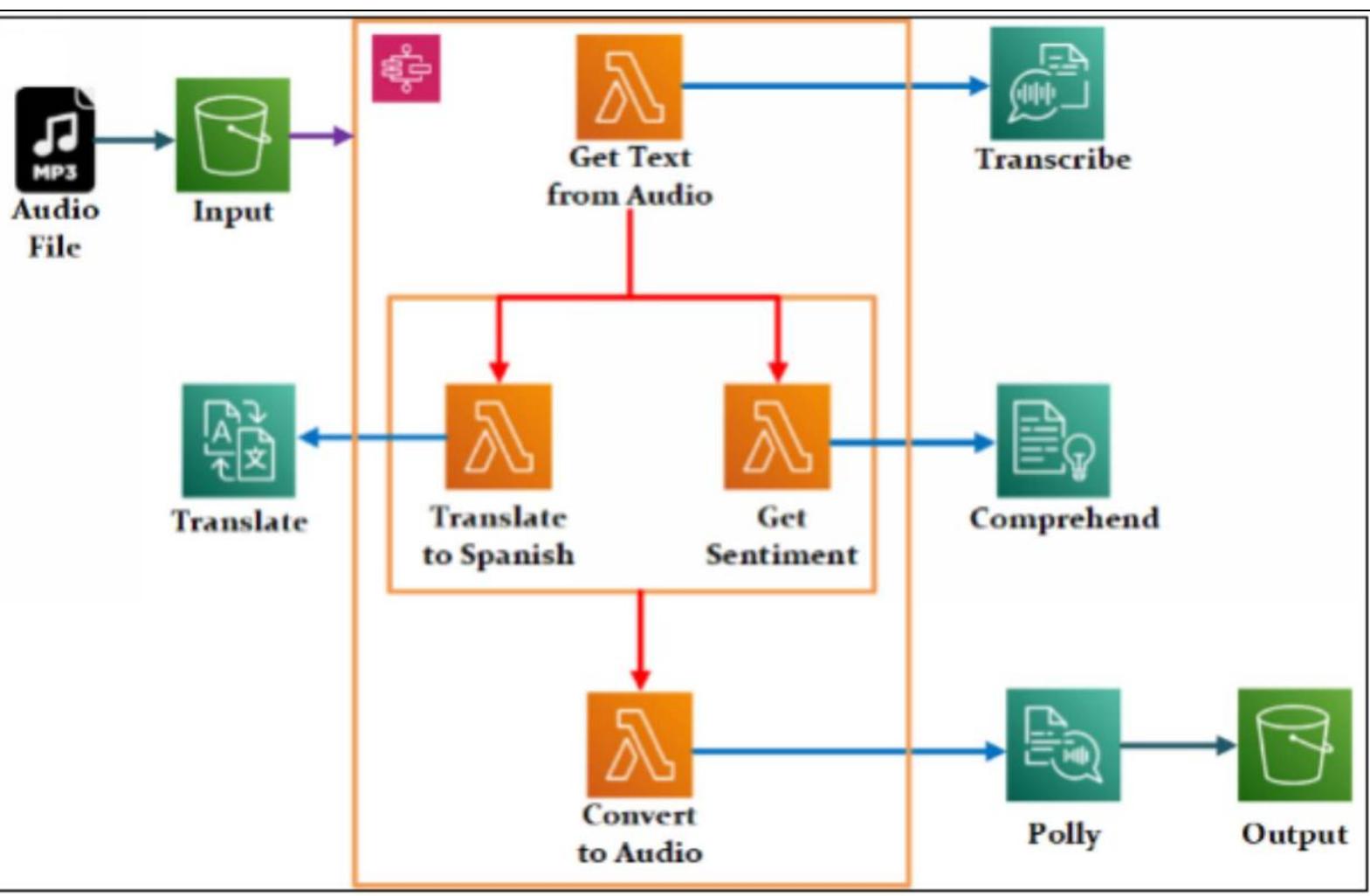
You are a Machine Learning developer at Netflix, developing content on different cloud computing courses. They want their content to be translated into the Spanish language. But there is an issue in translating the whole book of 300 to 500 pages by a content developer. The organization's requirement to complete the book in one week is impossible. They gave you the task of building a Natural Language Processing (NLP) model from scratch using Tensorflow or Pytorch framework. It requires lots of data on the Spanish and English languages. It also needs expensive hardware like GPU to train the model. Building the model from scratch also requires extensive time and cost. Hence, your task is to develop a Neural Machine Translation model with less time and a low budget. How will you automate this task?

Proposed Solution

The solution is simply to design a Machine Learning pipeline using AWS Lambda functions in the background to run the logic of the channel. But you can also use AWS S3 for input and output of data. You also use other Amazon helping services to automate this task. You use AWS Translate, AWS Transcribe, AWS Comprehend, and AWS Polly, as shown in the figure below.

Before deep-diving into the lab, create Lambda Function, Step Function, and S3 Bucket, as discussed in the above lab sections.

Lab Diagram



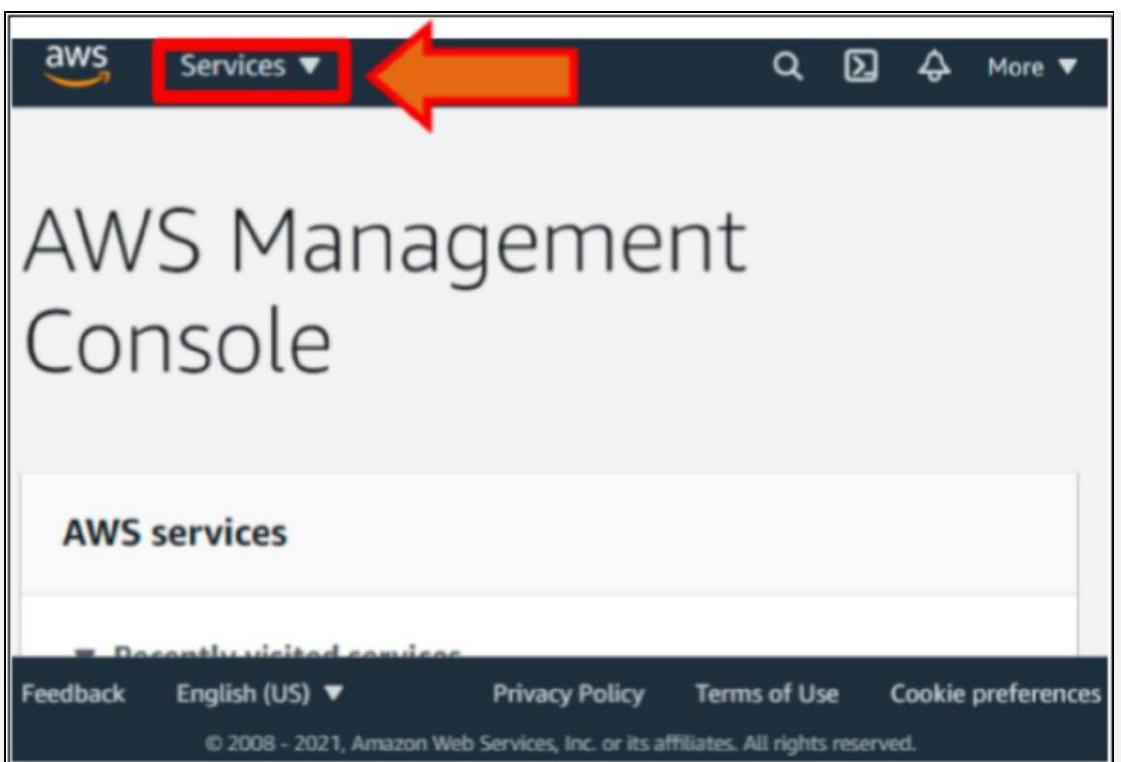
Implementation Steps

1. Create a Translation Pipeline.
2. Upload an Audio File on S3 Bucket
3. Translate Text on AWS Transcribe.

Solution

Step 1: Create Translation Pipeline

1. Log in to AWS Console.
2. Click on Services.



AWS Management Console

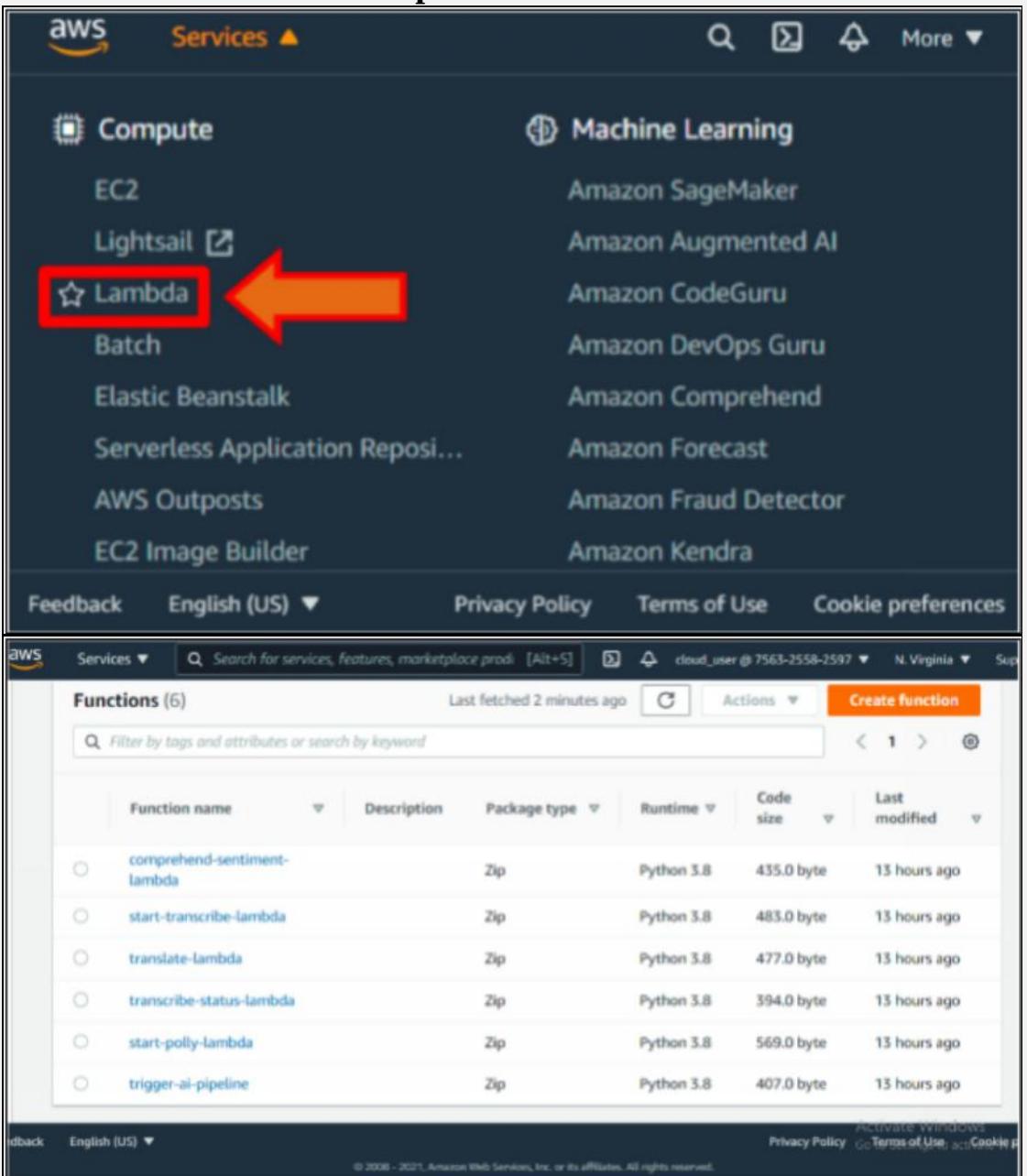
AWS services

Recently visited services

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

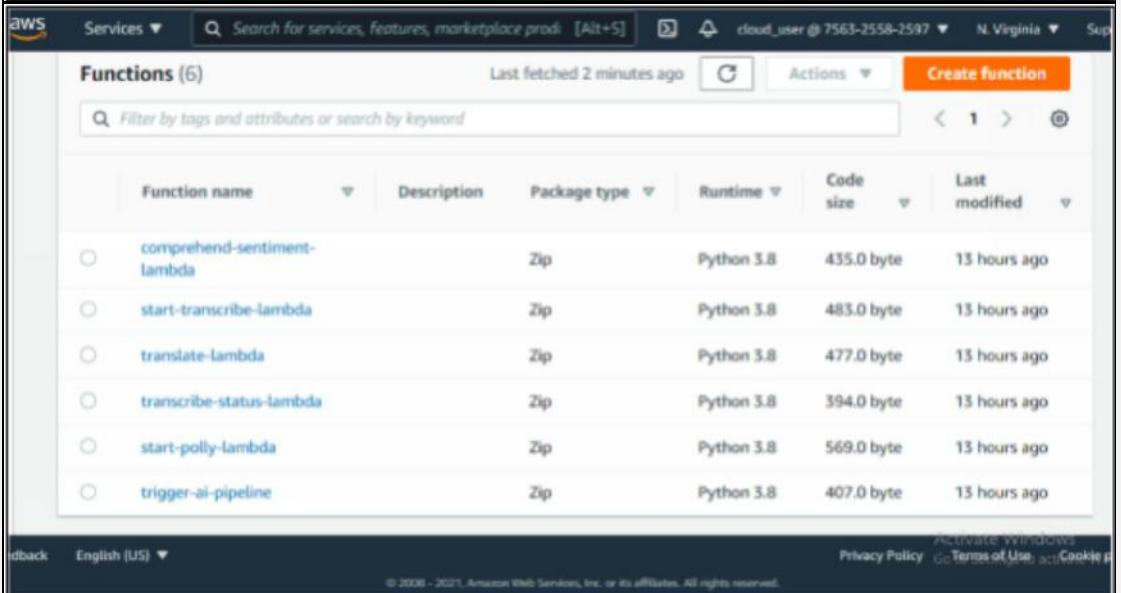
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Select Lambda from Compute.



Compute	Machine Learning
EC2	Amazon SageMaker
Lightsail	Amazon Augmented AI
Lambda	Amazon CodeGuru
Batch	Amazon DevOps Guru
Elastic Beanstalk	Amazon Comprehend
Serverless Application Repository	Amazon Forecast
AWS Outposts	Amazon Fraud Detector
EC2 Image Builder	Amazon Kendra

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences



Function name	Description	Package type	Runtime	Code size	Last modified
comprehend-sentiment-lambda		Zip	Python 3.8	435.0 byte	13 hours ago
start-transcribe-lambda		Zip	Python 3.8	483.0 byte	13 hours ago
translate-lambda		Zip	Python 3.8	477.0 byte	13 hours ago
transcribe-status-lambda		Zip	Python 3.8	394.0 byte	13 hours ago
start-polly-lambda		Zip	Python 3.8	569.0 byte	13 hours ago
trigger-ai-pipeline		Zip	Python 3.8	407.0 byte	13 hours ago

Search for services, features, marketplace products [Alt+5] cloud_user @ 7563-2558-2597 N. Virginia Support

Filter by tags and attributes or search by keyword

Functions (6) Last fetched 2 minutes ago Actions Create function

Function name Description Package type Runtime Code size Last modified

comprehend-sentiment-lambda Zip Python 3.8 435.0 byte 13 hours ago

start-transcribe-lambda Zip Python 3.8 483.0 byte 13 hours ago

translate-lambda Zip Python 3.8 477.0 byte 13 hours ago

transcribe-status-lambda Zip Python 3.8 394.0 byte 13 hours ago

start-polly-lambda Zip Python 3.8 569.0 byte 13 hours ago

trigger-ai-pipeline Zip Python 3.8 407.0 byte 13 hours ago

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences

4. Now open the Step Function tab.

The screenshot shows the AWS Step Functions console. On the left, a sidebar lists 'Step Functions', 'State machines' (which is highlighted in red), 'Activities', 'Data flow simulator', 'Feature spotlight', 'Local Development', and 'Join our feedback panel'. The main area is titled 'State machines (1)'. It displays a table with one row for 'Polyglot-Pipeline'. The columns are 'Name', 'Type', 'Creation date', 'Status', 'Logs', 'Total', and 'Running'. The 'Polyglot-Pipeline' row is selected, indicated by a blue circle icon. At the top of the main area are buttons for 'View details', 'Edit', 'Copy to new', and 'Delete'. Below these are 'Create state machine' and search/filter controls. The bottom of the page includes standard AWS footer links like 'Privacy Policy', 'Terms of Use', and 'Cookie policy'.

5. Add translation to the pipeline.

6. Click Polyglot-Pipeline.

This screenshot is identical to the one above, showing the Step Functions console with the 'State machines' tab selected. The 'Polyglot-Pipeline' row in the table is now highlighted with a red rectangular box, and an orange arrow points to it from the left side of the screen.

7. Click the Edit button.

This screenshot shows the 'Polyglot-Pipeline' details page. The top navigation bar shows the path: 'Step Functions > State machines > Polyglot-Pipeline'. The main title is 'Polyglot-Pipeline' with an 'Edit' button highlighted by a red arrow. Below the title is a 'Details' section containing ARN, Type, Creation date, and IAM role ARN information. At the bottom right of the page is another red arrow pointing to the 'Edit' button.

Salesforce

Services ▾ Search for services, features, marketplace prod [Alt+S] cloud_user @ 7563-2558-2597 N. Virginia ▾

Definition

Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

[Generate code snippet](#) [Format JSON](#)

```

1 + {
2   "StartAt": "Start Transcribe",
3   "States": [
4     "Start Transcribe": {
5       "Type": "Task",
6       "Resource": "arn:aws:states::lambda:invoke",
7       "Parameters": {
8         "FunctionName": "arn:aws:lambda:us-east-1:756325582597:Function:start-transcribe-lambda:$LATEST",
9         "Payload": {
10           "Input.$": "$"
11         }
12       },
13       "Next": "Wait for Transcribe"
14     }
  
```

Workflow Studio [New](#)

Start → Start Transcribe → Wait for Transcribe → Check Transcribe Status → Is Transcribe Complete → Wait for Transcribe

[Feedback](#) English (US) ▾ Privacy Policy [Go](#) [Terms of Use](#) [Activate Windows](#) [Cookie Policy](#)

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. From Generate a code snippet, select AWS Lambda: Invoke a function.

Salesforce

Services ▾ Search for services, features, marketplace prod [Alt+S] cloud_user @ 7563-2558-2597 N. Virginia ▾

Definition

Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

[Generate code snippet](#) [JSON](#)

Task states

AWS Lambda: Invoke a function
Call the Lambda Invoke API and invoke a function

Amazon SNS: Publish a message
Call the SNS Publish API to send a message to a destination

Amazon ECS: Manage a task
Call the ECS RunTask API

Next: "Wait for Transcribe"

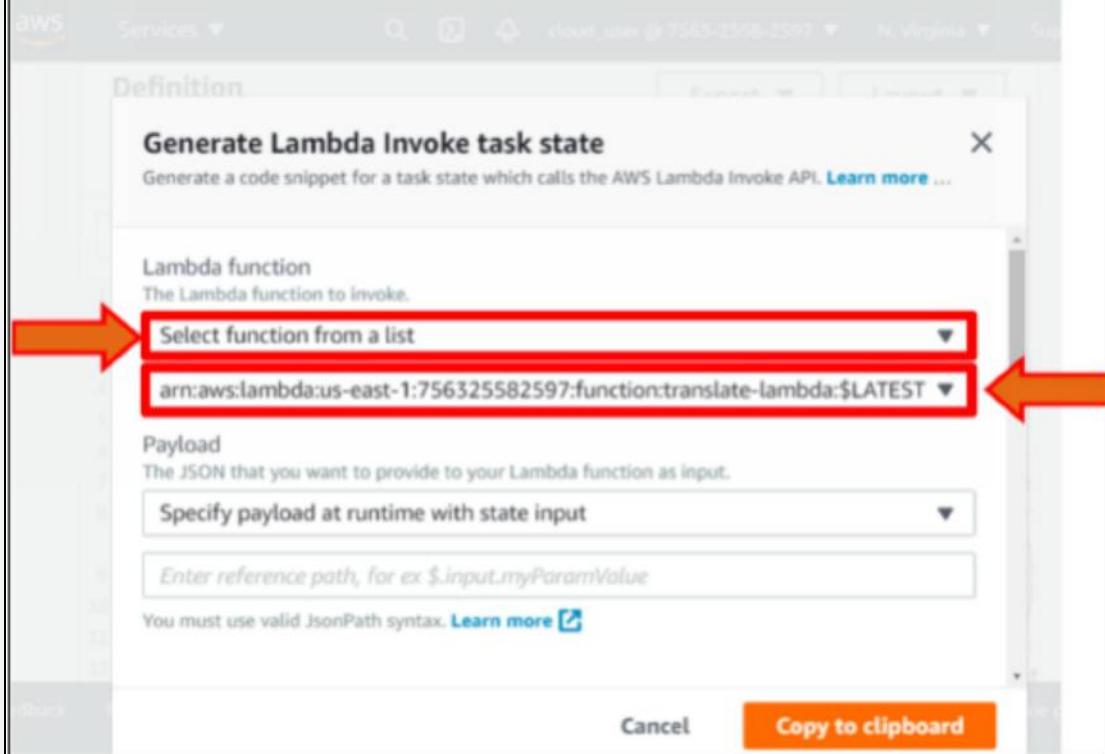
Workflow Studio [New](#)

Start → Start Transcribe → Wait for Transcribe → Check Transcribe Status → Is Transcribe Complete → Wait for Transcribe

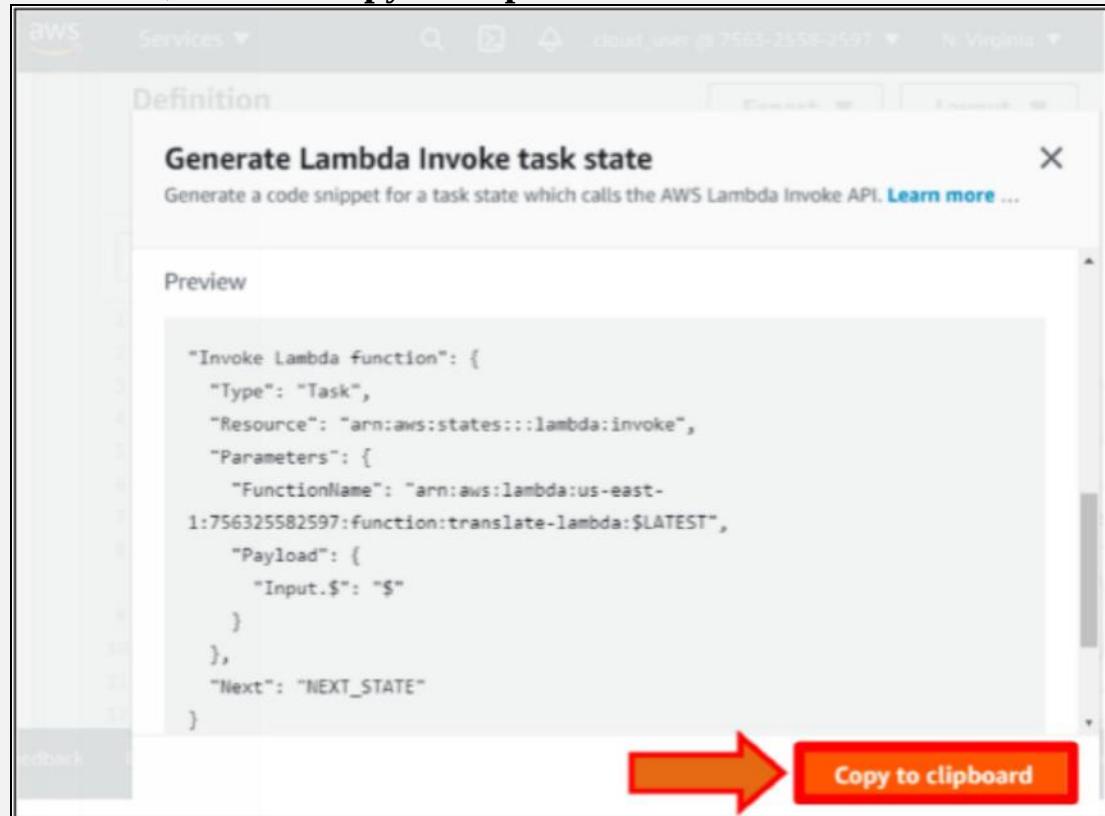
[Feedback](#) English (US) ▾ Privacy Policy [Go](#) [Terms of Use](#) [Activate Windows](#) [Cookie Policy](#)

© 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. From the Lambda function's dropdown menu, select a function from a list and then Select the **translate-lambda** function from the submenu.



10. Then, click on **Copy to Clipboard**.



11. Paste the code snippet on line 51.

Services ▾ Q Search for services, features, marketplace prod [Alt+S] cloud_user @ 2857-1548-9524 N. Virginia

```

45     "Default": "Wait For Transcribe"
46   },
47   "Transcript Available": {
48     "Type": "Pass",
49     "End": true
50   },
51   "Invoke Lambda Function": {
52     "Type": "Task",
53     "Resource": "arn:aws:states:::lambda:invoke",
54     "Parameters": {
55       "FunctionName": "arn:aws:lambda:us-east-1:285715489524:function:translate-lambda:$LATEST",
56       "Payload": {
57         "Input.$": "$"
58       }
59     },
60     "Next": "NEXT_STATE"
61   },
62   "Transcribe Failed": {
63     "End": true
64   }
65 }
```

Feedback English (US) ▾ Privacy Policy Go Terms of Use activate Cookies © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

12. Replace Invoke Lambda Function with Translate Text.

Services ▾ Q Search for services, features, marketplace prod [Alt+S] cloud_user @ 2857-1548-9524 N. Virginia

```

49     "End": true
50   },
51   "Translate Text": {
52     "Type": "Task",
53     "Resource": "arn:aws:states:::lambda:invoke",
54     "Parameters": {
55       "FunctionName": "arn:aws:lambda:us-east-1:285715489524:function:translate-lambda:$LATEST",
56       "Payload": {
57         "Input.$": "$"
58       }
59     },
60     "Next": "NEXT_STATE"
61   },
62   "Invoke Lambda function": {
63     "Type": "Task",
64     "Resource": "arn:aws:states:::lambda:invoke",
65     "Parameters": {
66       "FunctionName": "arn:aws:lambda:us-east-1:285715489524:function:comprehend-sentiment-lambda:$LATEST"
67     },
68     "Payload": {
69       "End": true
70     }
71 }
```

Feedback English (US) ▾ Privacy Policy Go Terms of Use activate Cookies © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

13. Now add Sentiment Analysis to the pipeline.

14. In the **Step Function** tab, from **Generate a code snippet**, select **AWS Lambda: Invoke a function**.

Services ▾ Q Search for services, features, marketplace prod [Alt+S] cloud_user @ 7563-2558-2597 N. Virginia

Definition Export Layout

Define your workflow using Amazon States Language Test your data flow with the new Data Flow Simulator.

Generate code snippet ▾ ON

AWS Lambda: Invoke a function Call the Lambda Invoke API and invoke a function

Amazon SNS: Publish a message Call the SNS Publish API to send a message to a destination

Amazon ECS: Manage a task Call the ECS RunTask API

```

45     "Default": "Wait For Transcribe"
46   },
47   "Transcript Available": {
48     "Type": "Pass",
49     "End": true
50   },
51   "Invoke Lambda Function": {
52     "Type": "Task",
53     "Resource": "arn:aws:states:::lambda:invoke",
54     "Parameters": {
55       "FunctionName": "arn:aws:lambda:us-east-1:285715489524:function:translate-lambda:$LATEST",
56       "Payload": {
57         "Input.$": "$"
58       }
59     },
60     "Next": "NEXT_STATE"
61   },
62   "Transcribe Failed": {
63     "End": true
64   }
65 }
```

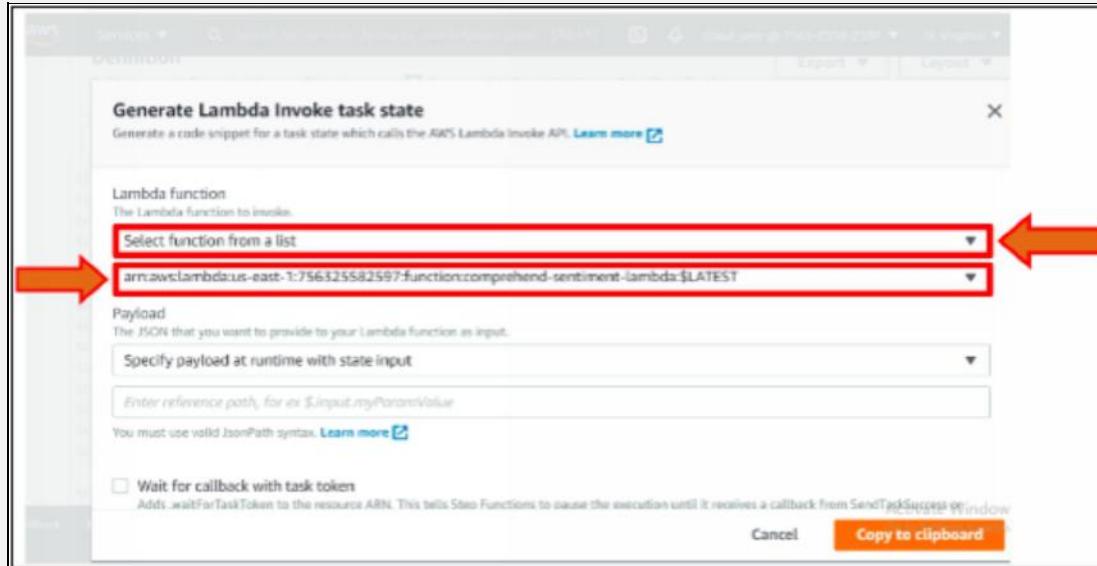
Workflow Studio New

Feedback English (US) ▾ Privacy Policy Go Terms of Use activate Cookies © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

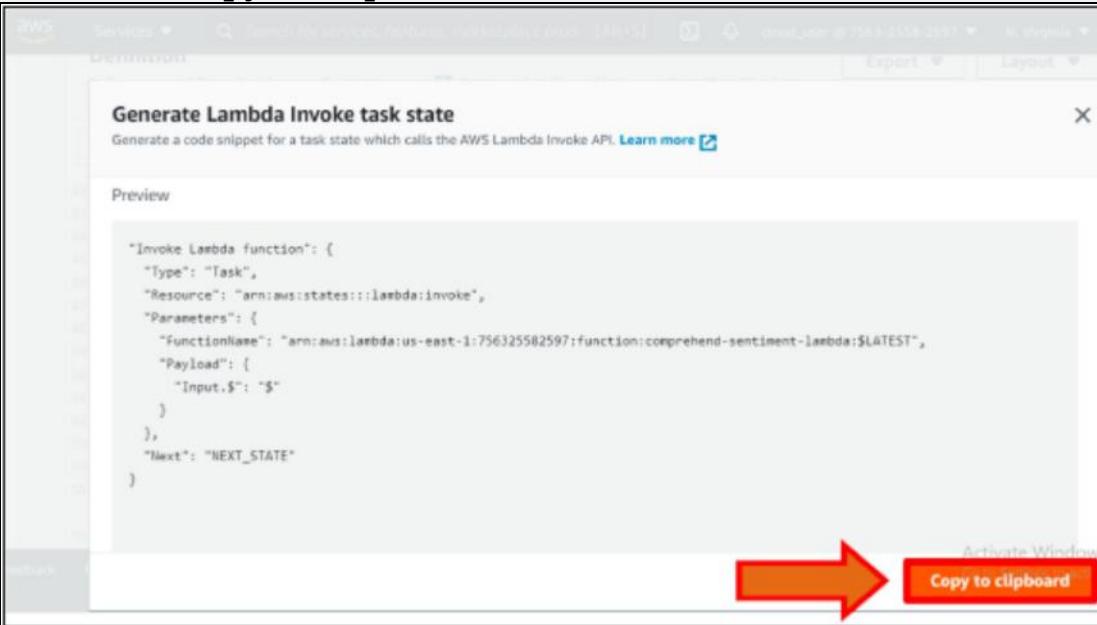
15. From the Lambda Function dropdown menu, select **function from**

a list.

16. Then select the **Comprehend-Sentiment** function from the sub-menu.



17. Click **Copy to Clipboard**.



18. Paste the **Comprehend-Sentiment** code snippet on line 62.
19. Replace **Invoke Lambda Function** with **Comprehend Sentiment**.

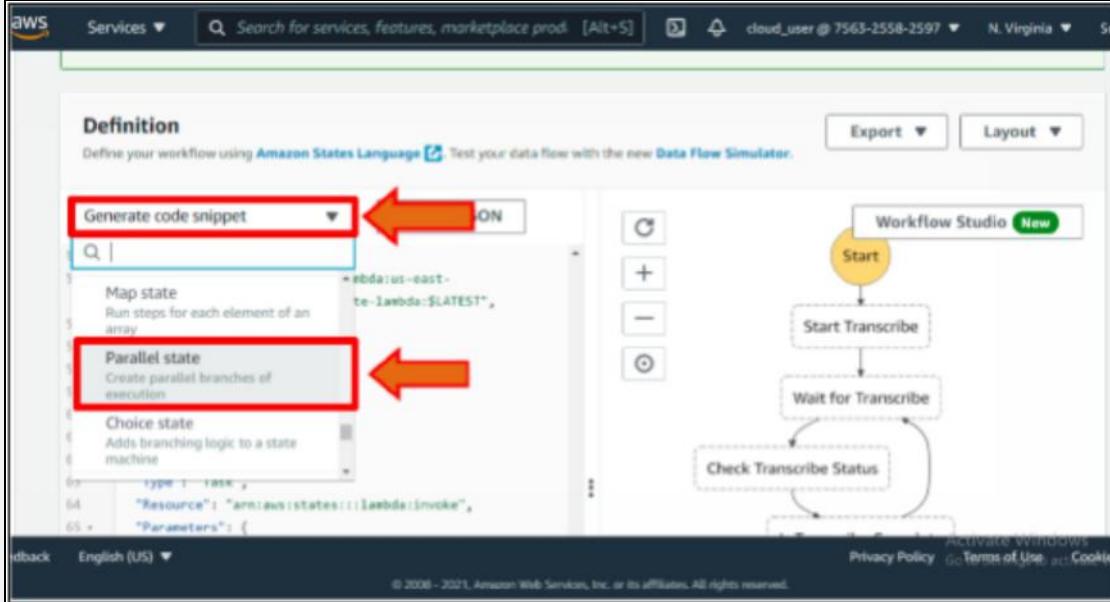
aws Services ▾ Search for services, features, marketplace products

```

49         "End": true
50     },
51     "Translate Text": {
52         "Type": "Task",
53         "Resource": "arn:aws:states:::lambda:invoke",
54         "Parameters": {
55             "FunctionName": "arn:aws:lambda:us-east-
56             1:285715489524:function:translate-lambda:$LATEST",
57             "Payload": {
58                 "Input.$": "$"
59             }
60         },
61         "Next": "NEXT_STATE"
62     },
63     "Comprehend Sentiment": { ←
64         "Type": "Task",
65         "Resource": "arn:aws:states:::lambda:invoke",
66         "Parameters": {
67             "FunctionName": "arn:aws:lambda:us-east-
68             1:285715489524:function:comprehend-sentiment-lambda:$LATEST"
69             "Payload": {
70
    
```

dback English (US) © 2008 - 2021, Amazon Web Services

20. Under Generate code snippet, select Parallel State.



21. Click Copy to Clipboard.

Generate a Parallel state example

Use a Parallel state to create parallel branches of execution in your state machine. [Learn more](#)

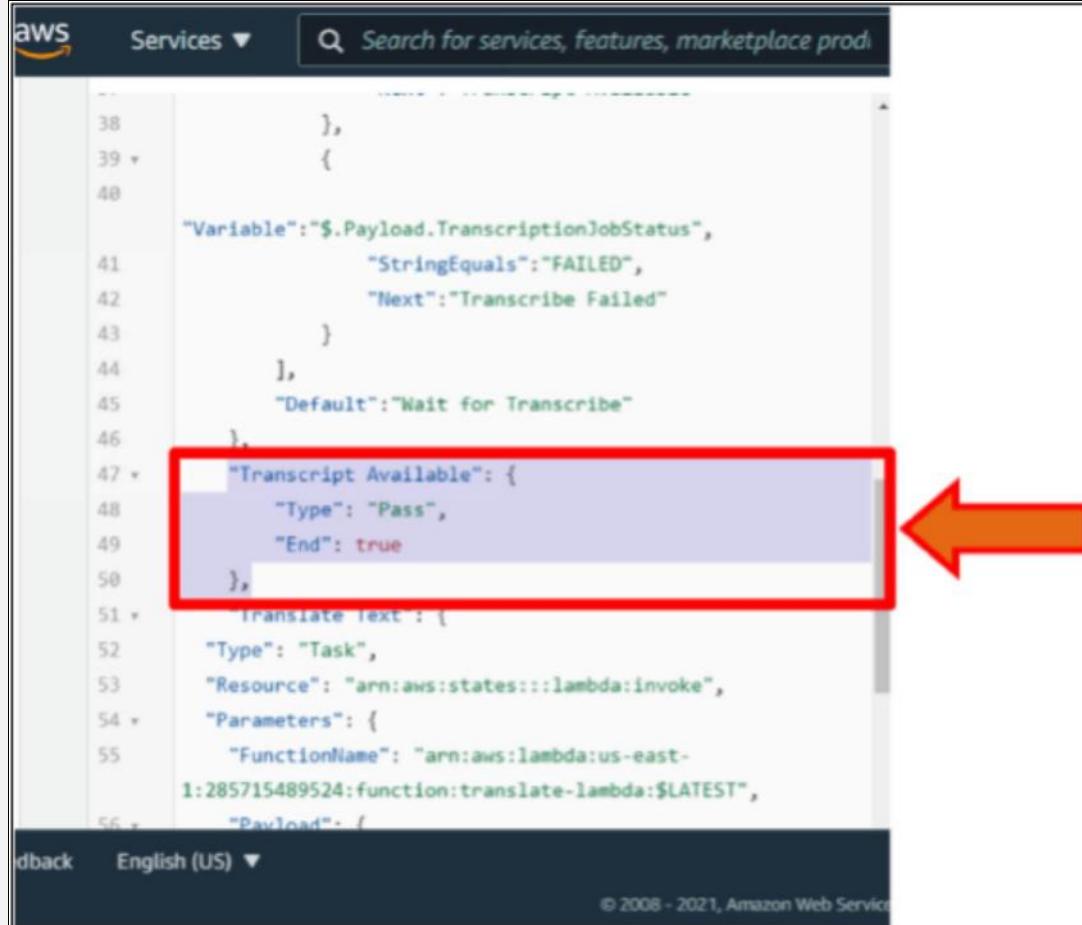
```
"Parallel State": {  
    "Type": "Parallel",  
    "Branches": [  
        {  
            "StartAt": "Pass State 1",  
            "States": {  
                "Pass State 1": {  
                    "Type": "Pass",  
                    "End": true  
                }  
            }  
        },  
        {  
            "StartAt": "Pass State 2",  
            "States": {  
                ...  
            }  
        }  
    ]  
},  
...  
}
```

Activate Windows

 Copy to clipboard



22. Remove the code from lines 47 to 50 and paste the **Parallel State** block code.



The screenshot shows the AWS Lambda function editor with the state machine configuration. A red box highlights the code block from line 47 to 50, which is the Parallel State block. An orange arrow points from this highlighted area to the corresponding code in the AWS CloudFormation template below.

```
38     ],  
39     {  
40         "Variable": "$.Payload.TranscriptionJobStatus",  
41             "StringEquals": "FAILED",  
42             "Next": "Transcribe Failed"  
43         }  
44     ],  
45     "Default": "Wait for Transcribe"  
46 },  
47     "Transcript Available": {  
48         "Type": "Pass",  
49         "End": true  
50     },  
51     "Translate Text": {  
52         "Type": "Task",  
53         "Resource": "arn:aws:states:::lambda:invoke",  
54         "Parameters": {  
55             "FunctionName": "arn:aws:lambda:us-east-  
1:285715489524:function:translate-lambda:$LATEST",  
56             "Payload": /  
...  
dback English (US) ▾
```

© 2006 - 2021, Amazon Web Services

AWS Services ▾ Search for services, features, marketplace products

```
42             "Next":"Transcribe Failed"
43         }
44     ],
45     "Default":"Wait for Transcribe"
46 },
47 "Parallel State": {
48     "Type": "Parallel",
49     "Branches": [
50         {
51             "StartAt": "Pass State 1",
52             "States": {
53                 "Pass State 1": {
54                     "Type": "Pass",
55                     "End": true
56                 }
57             }
58         },
59         {
60             "StartAt": "Pass State 2",
61             "States": {
62                 "Pass State 2": {
```

dback English (US) ▾ © 2008 - 2021, Amazon Web Services

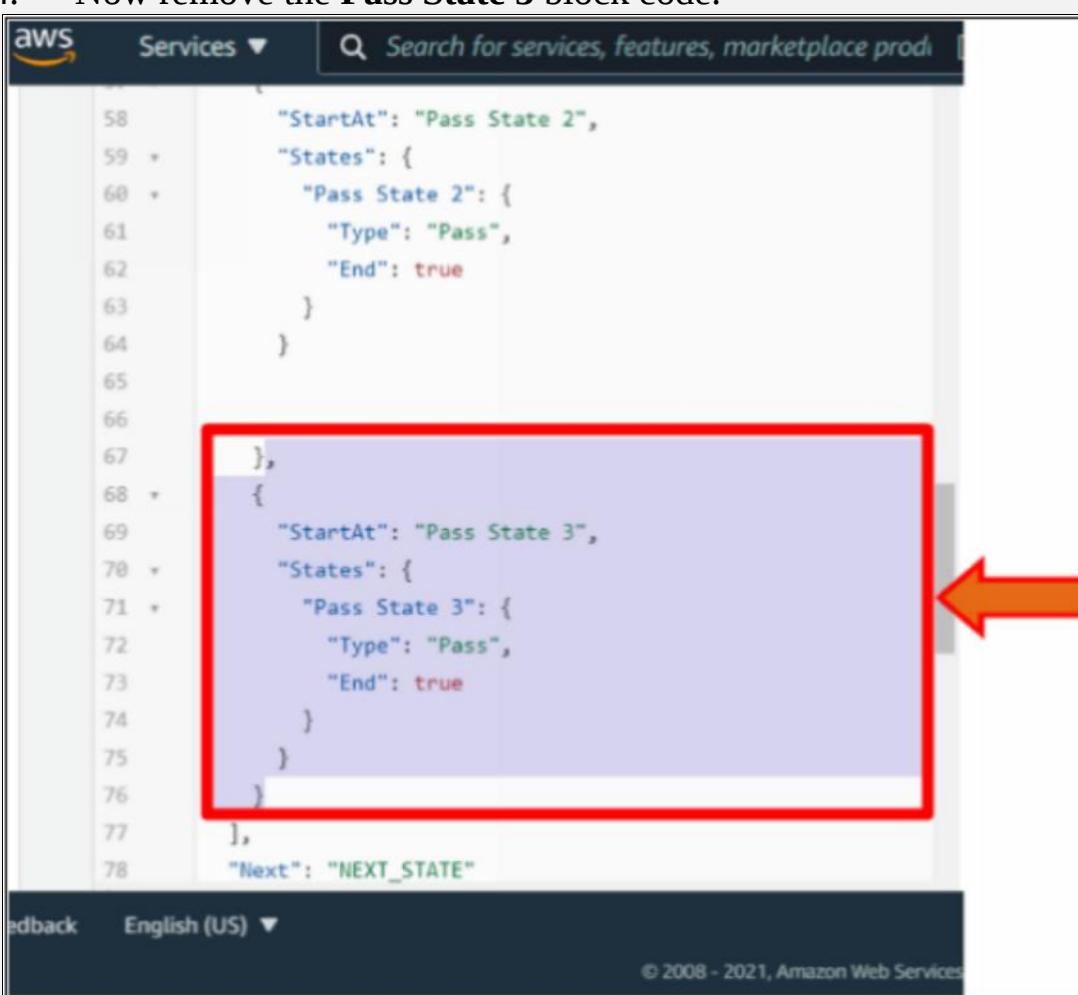
23. Replace Parallel State with Transcript Available.

AWS Services ▾ Search for services, features, marketplace products

```
42             "Next":"Transcribe Failed"
43         }
44     ],
45     "Default":"Wait for Transcribe"
46 },
47 "Transcript Available": {
48     "Type": "Parallel",
49     "Branches": [
50         {
51             "StartAt": "Pass State 1",
52             "States": {
53                 "Pass State 1": {
54                     "Type": "Pass",
55                     "End": true
56                 }
57             }
58         },
59         {
60             "StartAt": "Pass State 2",
61             "States": {
62                 "Pass State 2": {
```

dback English (US) ▾ © 2008 - 2021, Amazon Web Services

24. Now remove the **Pass State 3** block code.



```
aws Services ▼ Search for services, features, marketplace products
58     "StartAt": "Pass State 2",
59     "States": {
60       "Pass State 2": {
61         "Type": "Pass",
62         "End": true
63       }
64     }
65
66   ],
67   {
68     "StartAt": "Pass State 3",
69     "States": {
70       "Pass State 3": {
71         "Type": "Pass",
72         "End": true
73       }
74     }
75   }
76 }
77 ],
78 "Next": "NEXT_STATE"
```

Feedback English (US) © 2008 - 2021, Amazon Web Services

25. Copy the **Translate Text** block code. Remove it from its current position.

aws Services ▾ Search for services, features, marketplace products

```
64      }
65
66
67    }
68  ],
69  "Next": "NEXT_STATE"
70 }
71
72 +   "Translate Text": {
73  "Type": "Task",
74  "Resource": "arn:aws:states:::lambda:invoke",
75  "Parameters": {
76    "FunctionName": "arn:aws:lambda:us-east-1:285715489524:function:translate-lambda:$LATEST",
77  "Payload": {
78    "Input.$": "$"
79  }
80 },
81 "Next": "NEXT_STATE"
82 },
83 +   "Comprehend Sentiment": {
```

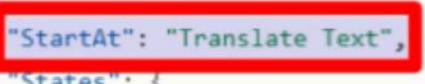
Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc.

26. Remove Pass state 1 from lines 53 to 56 and paste the Translate Text block code.

aws Services ▾ Search for services, features, marketplace products [Alt]

```
44      ],
45      "Default": "Wait for Transcribe"
46  ],
47 +   "Parallel State": {
48  "Type": "Parallel",
49  "Branches": [
50 +    {
51    "StartAt": "Pass State 1",
52  "States": {
53 +      "Pass State 1": {
54        "Type": "Pass",
55        "End": true
56      }
57    }
58  ],
59  {
60    "StartAt": "Pass State 2",
61  "States": {
62 +      "Pass State 2": {
63        "Type": "Pass",
64        "End": true
65      }
66    }
67  ],
68  {
69    "StartAt": "Pass State 3",
70  "States": {
71 +      "Pass State 3": {
72        "Type": "Pass",
73        "End": true
74      }
75    }
76  ],
77  {
78    "StartAt": "Pass State 4",
79  "States": {
80 +      "Pass State 4": {
81        "Type": "Pass",
82        "End": true
83      }
84    }
85  ],
86  {
87    "StartAt": "Pass State 5",
88  "States": {
89 +      "Pass State 5": {
90        "Type": "Pass",
91        "End": true
92      }
93    }
94  ],
95  {
96    "StartAt": "Pass State 6",
97  "States": {
98 +      "Pass State 6": {
99        "Type": "Pass",
100       "End": true
101      }
102    }
103  ],
104  {
105    "StartAt": "Pass State 7",
106  "States": {
107 +      "Pass State 7": {
108        "Type": "Pass",
109        "End": true
110      }
111    }
112  ],
113  {
114    "StartAt": "Pass State 8",
115  "States": {
116 +      "Pass State 8": {
117        "Type": "Pass",
118        "End": true
119      }
120    }
121  ],
122  {
123    "StartAt": "Pass State 9",
124  "States": {
125 +      "Pass State 9": {
126        "Type": "Pass",
127        "End": true
128      }
129    }
130  ],
131  {
132    "StartAt": "Pass State 10",
133  "States": {
134 +      "Pass State 10": {
135        "Type": "Pass",
136        "End": true
137      }
138    }
139  ],
140  {
141    "StartAt": "Pass State 11",
142  "States": {
143 +      "Pass State 11": {
144        "Type": "Pass",
145        "End": true
146      }
147    }
148  ],
149  {
150    "StartAt": "Pass State 12",
151  "States": {
152 +      "Pass State 12": {
153        "Type": "Pass",
154        "End": true
155      }
156    }
157  ],
158  {
159    "StartAt": "Pass State 13",
160  "States": {
161 +      "Pass State 13": {
162        "Type": "Pass",
163        "End": true
164      }
165    }
166  ],
167  {
168    "StartAt": "Pass State 14",
169  "States": {
170 +      "Pass State 14": {
171        "Type": "Pass",
172        "End": true
173      }
174    }
175  ],
176  {
177    "StartAt": "Pass State 15",
178  "States": {
179 +      "Pass State 15": {
180        "Type": "Pass",
181        "End": true
182      }
183    }
184  ],
185  {
186    "StartAt": "Pass State 16",
187  "States": {
188 +      "Pass State 16": {
189        "Type": "Pass",
190        "End": true
191      }
192    }
193  ],
194  {
195    "StartAt": "Pass State 17",
196  "States": {
197 +      "Pass State 17": {
198        "Type": "Pass",
199        "End": true
200      }
201    }
202  ],
203  {
204    "StartAt": "Pass State 18",
205  "States": {
206 +      "Pass State 18": {
207        "Type": "Pass",
208        "End": true
209      }
210    }
211  ],
212  {
213    "StartAt": "Pass State 19",
214  "States": {
215 +      "Pass State 19": {
216        "Type": "Pass",
217        "End": true
218      }
219    }
220  ],
221  {
222    "StartAt": "Pass State 20",
223  "States": {
224 +      "Pass State 20": {
225        "Type": "Pass",
226        "End": true
227      }
228    }
229  ],
230  {
231    "StartAt": "Pass State 21",
232  "States": {
233 +      "Pass State 21": {
234        "Type": "Pass",
235        "End": true
236      }
237    }
238  ],
239  {
240    "StartAt": "Pass State 22",
241  "States": {
242 +      "Pass State 22": {
243        "Type": "Pass",
244        "End": true
245      }
246    }
247  ],
248  {
249    "StartAt": "Pass State 23",
250  "States": {
251 +      "Pass State 23": {
252        "Type": "Pass",
253        "End": true
254      }
255    }
256  ],
257  {
258    "StartAt": "Pass State 24",
259  "States": {
260 +      "Pass State 24": {
261        "Type": "Pass",
262        "End": true
263      }
264    }
265  ],
266  {
267    "StartAt": "Pass State 25",
268  "States": {
269 +      "Pass State 25": {
270        "Type": "Pass",
271        "End": true
272      }
273    }
274  ],
275  {
276    "StartAt": "Pass State 26",
277  "States": {
278 +      "Pass State 26": {
279        "Type": "Pass",
280        "End": true
281      }
282    }
283  ],
284  {
285    "StartAt": "Pass State 27",
286  "States": {
287 +      "Pass State 27": {
288        "Type": "Pass",
289        "End": true
290      }
291    }
292  ],
293  {
294    "StartAt": "Pass State 28",
295  "States": {
296 +      "Pass State 28": {
297        "Type": "Pass",
298        "End": true
299      }
300    }
301  ],
302  {
303    "StartAt": "Pass State 29",
304  "States": {
305 +      "Pass State 29": {
306        "Type": "Pass",
307        "End": true
308      }
309    }
310  ],
311  {
312    "StartAt": "Pass State 30",
313  "States": {
314 +      "Pass State 30": {
315        "Type": "Pass",
316        "End": true
317      }
318    }
319  ],
320  {
321    "StartAt": "Pass State 31",
322  "States": {
323 +      "Pass State 31": {
324        "Type": "Pass",
325        "End": true
326      }
327    }
328  ],
329  {
330    "StartAt": "Pass State 32",
331  "States": {
332 +      "Pass State 32": {
333        "Type": "Pass",
334        "End": true
335      }
336    }
337  ],
338  {
339    "StartAt": "Pass State 33",
340  "States": {
341 +      "Pass State 33": {
342        "Type": "Pass",
343        "End": true
344      }
345    }
346  ],
347  {
348    "StartAt": "Pass State 34",
349  "States": {
350 +      "Pass State 34": {
351        "Type": "Pass",
352        "End": true
353      }
354    }
355  ],
356  {
357    "StartAt": "Pass State 35",
358  "States": {
359 +      "Pass State 35": {
360        "Type": "Pass",
361        "End": true
362      }
363    }
364  ],
365  {
366    "StartAt": "Pass State 36",
367  "States": {
368 +      "Pass State 36": {
369        "Type": "Pass",
370        "End": true
371      }
372    }
373  ],
374  {
375    "StartAt": "Pass State 37",
376  "States": {
377 +      "Pass State 37": {
378        "Type": "Pass",
379        "End": true
380      }
381    }
382  ],
383  {
384    "StartAt": "Pass State 38",
385  "States": {
386 +      "Pass State 38": {
387        "Type": "Pass",
388        "End": true
389      }
390    }
391  ],
392  {
393    "StartAt": "Pass State 39",
394  "States": {
395 +      "Pass State 39": {
396        "Type": "Pass",
397        "End": true
398      }
399    }
400  ],
401  {
402    "StartAt": "Pass State 40",
403  "States": {
404 +      "Pass State 40": {
405        "Type": "Pass",
406        "End": true
407      }
408    }
409  ],
410  {
411    "StartAt": "Pass State 41",
412  "States": {
413 +      "Pass State 41": {
414        "Type": "Pass",
415        "End": true
416      }
417    }
418  ],
419  {
420    "StartAt": "Pass State 42",
421  "States": {
422 +      "Pass State 42": {
423        "Type": "Pass",
424        "End": true
425      }
426    }
427  ],
428  {
429    "StartAt": "Pass State 43",
430  "States": {
431 +      "Pass State 43": {
432        "Type": "Pass",
433        "End": true
434      }
435    }
436  ],
437  {
438    "StartAt": "Pass State 44",
439  "States": {
440 +      "Pass State 44": {
441        "Type": "Pass",
442        "End": true
443      }
444    }
445  ],
446  {
447    "StartAt": "Pass State 45",
448  "States": {
449 +      "Pass State 45": {
450        "Type": "Pass",
451        "End": true
452      }
453    }
454  ],
455  {
456    "StartAt": "Pass State 46",
457  "States": {
458 +      "Pass State 46": {
459        "Type": "Pass",
460        "End": true
461      }
462    }
463  ],
464  {
465    "StartAt": "Pass State 47",
466  "States": {
467 +      "Pass State 47": {
468        "Type": "Pass",
469        "End": true
470      }
471    }
472  ],
473  {
474    "StartAt": "Pass State 48",
475  "States": {
476 +      "Pass State 48": {
477        "Type": "Pass",
478        "End": true
479      }
480    }
481  ],
482  {
483    "StartAt": "Pass State 49",
484  "States": {
485 +      "Pass State 49": {
486        "Type": "Pass",
487        "End": true
488      }
489    }
490  ],
491  {
492    "StartAt": "Pass State 50",
493  "States": {
494 +      "Pass State 50": {
495        "Type": "Pass",
496        "End": true
497      }
498    }
499  ],
500  {
501    "StartAt": "Pass State 51",
502  "States": {
503 +      "Pass State 51": {
504        "Type": "Pass",
505        "End": true
506      }
507    }
508  ],
509  {
510    "StartAt": "Pass State 52",
511  "States": {
512 +      "Pass State 52": {
513        "Type": "Pass",
514        "End": true
515      }
516    }
517  ],
518  {
519    "StartAt": "Pass State 53",
520  "States": {
521 +      "Pass State 53": {
522        "Type": "Pass",
523        "End": true
524      }
525    }
526  ],
527  {
528    "StartAt": "Pass State 54",
529  "States": {
530 +      "Pass State 54": {
531        "Type": "Pass",
532        "End": true
533      }
534    }
535  ],
536  {
537    "StartAt": "Pass State 55",
538  "States": {
539 +      "Pass State 55": {
540        "Type": "Pass",
541        "End": true
542      }
543    }
544  ],
545  {
546    "StartAt": "Pass State 56",
547  "States": {
548 +      "Pass State 56": {
549        "Type": "Pass",
550        "End": true
551      }
552    }
553  ],
554  {
555    "StartAt": "Pass State 57",
556  "States": {
557 +      "Pass State 57": {
558        "Type": "Pass",
559        "End": true
560      }
561    }
562  ],
563  {
564    "StartAt": "Pass State 58",
565  "States": {
566 +      "Pass State 58": {
567        "Type": "Pass",
568        "End": true
569      }
570    }
571  ],
572  {
573    "StartAt": "Pass State 59",
574  "States": {
575 +      "Pass State 59": {
576        "Type": "Pass",
577        "End": true
578      }
579    }
580  ],
581  {
582    "StartAt": "Pass State 60",
583  "States": {
584 +      "Pass State 60": {
585        "Type": "Pass",
586        "End": true
587      }
588    }
589  ],
590  {
591    "StartAt": "Pass State 61",
592  "States": {
593 +      "Pass State 61": {
594        "Type": "Pass",
595        "End": true
596      }
597    }
598  ],
599  {
600    "StartAt": "Pass State 62",
601  "States": {
602 +      "Pass State 62": {
603        "Type": "Pass",
604        "End": true
605      }
606    }
607  ],
608  {
609    "StartAt": "Pass State 63",
610  "States": {
611 +      "Pass State 63": {
612        "Type": "Pass",
613        "End": true
614      }
615    }
616  ],
617  {
618    "StartAt": "Pass State 64",
619  "States": {
620 +      "Pass State 64": {
621        "Type": "Pass",
622        "End": true
623      }
624    }
625  ],
626  {
627    "StartAt": "Pass State 65",
628  "States": {
629 +      "Pass State 65": {
630        "Type": "Pass",
631        "End": true
632      }
633    }
634  ],
635  {
636    "StartAt": "Pass State 66",
637  "States": {
638 +      "Pass State 66": {
639        "Type": "Pass",
640        "End": true
641      }
642    }
643  ],
644  {
645    "StartAt": "Pass State 67",
646  "States": {
647 +      "Pass State 67": {
648        "Type": "Pass",
649        "End": true
650      }
651    }
652  ],
653  {
654    "StartAt": "Pass State 68",
655  "States": {
656 +      "Pass State 68": {
657        "Type": "Pass",
658        "End": true
659      }
660    }
661  ],
662  {
663    "StartAt": "Pass State 69",
664  "States": {
665 +      "Pass State 69": {
666        "Type": "Pass",
667        "End": true
668      }
669    }
670  ],
671  {
672    "StartAt": "Pass State 70",
673  "States": {
674 +      "Pass State 70": {
675        "Type": "Pass",
676        "End": true
677      }
678    }
679  ],
680  {
681    "StartAt": "Pass State 71",
682  "States": {
683 +      "Pass State 71": {
684        "Type": "Pass",
685        "End": true
686      }
687    }
688  ],
689  {
690    "StartAt": "Pass State 72",
691  "States": {
692 +      "Pass State 72": {
693        "Type": "Pass",
694        "End": true
695      }
696    }
697  ],
698  {
699    "StartAt": "Pass State 73",
700  "States": {
701 +      "Pass State 73": {
702        "Type": "Pass",
703        "End": true
704      }
705    }
706  ],
707  {
708    "StartAt": "Pass State 74",
709  "States": {
710 +      "Pass State 74": {
711        "Type": "Pass",
712        "End": true
713      }
714    }
715  ],
716  {
717    "StartAt": "Pass State 75",
718  "States": {
719 +      "Pass State 75": {
720        "Type": "Pass",
721        "End": true
722      }
723    }
724  ],
725  {
726    "StartAt": "Pass State 76",
727  "States": {
728 +      "Pass State 76": {
729        "Type": "Pass",
730        "End": true
731      }
732    }
733  ],
734  {
735    "StartAt": "Pass State 77",
736  "States": {
737 +      "Pass State 77": {
738        "Type": "Pass",
739        "End": true
740      }
741    }
742  ],
743  {
744    "StartAt": "Pass State 78",
745  "States": {
746 +      "Pass State 78": {
747        "Type": "Pass",
748        "End": true
749      }
750    }
751  ],
752  {
753    "StartAt": "Pass State 79",
754  "States": {
755 +      "Pass State 79": {
756        "Type": "Pass",
757        "End": true
758      }
759    }
760  ],
761  {
762    "StartAt": "Pass State 80",
763  "States": {
764 +      "Pass State 80": {
765        "Type": "Pass",
766        "End": true
767      }
768    }
769  ],
770  {
771    "StartAt": "Pass State 81",
772  "States": {
773 +      "Pass State 81": {
774        "Type": "Pass",
775        "End": true
776      }
777    }
778  ],
779  {
780    "StartAt": "Pass State 82",
781  "States": {
782 +      "Pass State 82": {
783        "Type": "Pass",
784        "End": true
785      }
786    }
787  ],
788  {
789    "StartAt": "Pass State 83",
790  "States": {
791 +      "Pass State 83": {
792        "Type": "Pass",
793        "End": true
794      }
795    }
796  ],
797  {
798    "StartAt": "Pass State 84",
799  "States": {
800 +      "Pass State 84": {
801        "Type": "Pass",
802        "End": true
803      }
804    }
805  ],
806  {
807    "StartAt": "Pass State 85",
808  "States": {
809 +      "Pass State 85": {
810        "Type": "Pass",
811        "End": true
812      }
813    }
814  ],
815  {
816    "StartAt": "Pass State 86",
817  "States": {
818 +      "Pass State 86": {
819        "Type": "Pass",
820        "End": true
821      }
822    }
823  ],
824  {
825    "StartAt": "Pass State 87",
826  "States": {
827 +      "Pass State 87": {
828        "Type": "Pass",
829        "End": true
830      }
831    }
832  ],
833  {
834    "StartAt": "Pass State 88",
835  "States": {
836 +      "Pass State 88": {
837        "Type": "Pass",
838        "End": true
839      }
840    }
841  ],
842  {
843    "StartAt": "Pass State 89",
844  "States": {
845 +      "Pass State 89": {
846        "Type": "Pass",
847        "End": true
848      }
849    }
850  ],
851  {
852    "StartAt": "Pass State 90",
853  "States": {
854 +      "Pass State 90": {
855        "Type": "Pass",
856        "End": true
857      }
858    }
859  ],
860  {
861    "StartAt": "Pass State 91",
862  "States": {
863 +      "Pass State 91": {
864        "Type": "Pass",
865        "End": true
866      }
867    }
868  ],
869  {
870    "StartAt": "Pass State 92",
871  "States": {
872 +      "Pass State 92": {
873        "Type": "Pass",
874        "End": true
875      }
876    }
877  ],
878  {
879    "StartAt": "Pass State 93",
880  "States": {
881 +      "Pass State 93": {
882        "Type": "Pass",
883        "End": true
884      }
885    }
886  ],
887  {
888    "StartAt": "Pass State 94",
889  "States": {
890 +      "Pass State 94": {
891        "Type": "Pass",
892        "End": true
893      }
894    }
895  ],
896  {
897    "StartAt": "Pass State 95",
898  "States": {
899 +      "Pass State 95": {
900        "Type": "Pass",
901        "End": true
902      }
903    }
904  ],
905  {
906    "StartAt": "Pass State 96",
907  "States": {
908 +      "Pass State 96": {
909        "Type": "Pass",
910        "End": true
911      }
912    }
913  ],
914  {
915    "StartAt": "Pass State 97",
916  "States": {
917 +      "Pass State 97": {
918        "Type": "Pass",
919        "End": true
920      }
921    }
922  ],
923  {
924    "StartAt": "Pass State 98",
925  "States": {
926 +      "Pass State 98": {
927        "Type": "Pass",
928        "End": true
929      }
930    }
931  ],
932  {
933    "StartAt": "Pass State 99",
934  "States": {
935 +      "Pass State 99": {
936        "Type": "Pass",
937        "End": true
938      }
939    }
940  ],
941  {
942    "StartAt": "Pass State 100",
943  "States": {
944 +      "Pass State 100": {
945        "Type": "Pass",
946        "End": true
947      }
948    }
949  ],
950  {
951    "StartAt": "Pass State 101",
952  "States": {
953 +      "Pass State 101": {
954        "Type": "Pass",
955        "End": true
956      }
957    }
958  ],
959  {
960    "StartAt": "Pass State 102",
961  "States": {
962 +      "Pass State 102": {
963        "Type": "Pass",
964        "End": true
965      }
966    }
967  ],
968  {
969    "StartAt": "Pass State 103",
970  "States": {
971 +      "Pass State 103": {
972        "Type": "Pass",
973        "End": true
974      }
975    }
976  ],
977  {
978    "StartAt": "Pass State 104",
979  "States": {
980 +      "Pass State 104": {
981        "Type": "Pass",
982        "End": true
983      }
984    }
985  ],
986  {
987    "StartAt": "Pass State 105",
988  "States": {
989 +      "Pass State 105": {
990        "Type": "Pass",
991        "End": true
992      }
993    }
994  ],
995  {
996    "StartAt": "Pass State 106",
997  "States": {
998 +      "Pass State 106": {
999        "Type": "Pass",
1000       "End": true
1001      }
1002    }
1003  ],
1004  {
1005    "StartAt": "Pass State 107",
1006  "States": {
1007 +      "Pass State 107": {
1008        "Type": "Pass",
1009       "End": true
1010      }
1011    }
1012  ],
1013  {
1014    "StartAt": "Pass State 108",
1015  "States": {
1016 +      "Pass State 108": {
1017        "Type": "Pass",
1018       "End": true
1019      }
1020    }
1021  ],
1022  {
1023    "StartAt": "Pass State 109",
1024  "States": {
1025 +      "Pass State 109": {
1026        "Type": "Pass",
1027       "End": true
1028      }
1029    }
1030  ],
1031  {
1032    "StartAt": "Pass State 110",
1033  "States": {
1034 +      "Pass State 110": {
1035        "Type": "Pass",
1036       "End": true
1037      }
1038    }
1039  ],
1040  {
1041    "StartAt": "Pass State 111",
1042  "States": {
1043 +      "Pass State 111": {
1044        "Type": "Pass",
1045       "End": true
1046      }
1047    }
1048  ],
1049  {
1050    "StartAt": "Pass State 112",
1051  "States": {
1052 +      "Pass State 112": {
1053        "Type": "Pass",
1054       "End": true
1055      }
1056    }
1057  ],
1058  {
1059    "StartAt": "Pass State 113",
1060  "States": {
1061 +      "Pass State 113": {
1062        "Type": "Pass",
1063       "End": true
1064      }
1065    }
1066  ],
1067  {
1068    "StartAt": "Pass State 114",
1069  "States": {
1070 +      "Pass State 114": {
1071        "Type": "Pass",
1072       "End": true
1073      }
1074    }
1075  ],
1076  {
1077    "StartAt": "Pass State 115",
1078  "States": {
1079 +      "Pass State 115": {
1080        "Type": "Pass",
1081       "End": true
1082      }
1083    }
1084  ],
1085  {
1086    "StartAt": "Pass State 116",
1087  "States": {
1088 +      "Pass State 116": {
1089        "Type": "Pass",
1090       "End": true
1091      }
1092    }
1093  ],
1094  {
1095    "StartAt": "Pass State 117",
1096  "States": {
1097 +      "Pass State 117": {
1098        "Type": "Pass",
1099       "End": true
1100      }
1101    }
1102  ],
1103  {
1104    "StartAt": "Pass State 118",
1105  "States": {
1106 +      "Pass State 118": {
1107        "Type": "Pass",
1108       "End": true
1109      }
1110    }
1111  ],
1112  {
1113    "StartAt": "Pass State 119",
1114  "States": {
1115 +      "Pass State 119": {
1116        "Type": "Pass",
1117       "End": true
1118      }
1119    }
1120  ],
1121  {
1122    "StartAt": "Pass State 120",
1123  "States": {
1124 +      "Pass State 120": {
1125        "Type": "Pass",
1126       "End": true
1127      }
1128    }
1129  ],
1130  {
1131    "StartAt": "Pass State 121",
1132  "States": {
1133 +      "Pass State 121": {
1134        "Type": "Pass",
1135       "End": true
1136      }
1137    }
1138  ],
1139  {
1140    "StartAt": "Pass State 122",
1141  "States": {
1142 +      "Pass State 122": {
1143        "Type": "Pass",
1144       "End": true
1145      }
1146    }
1147  ],
1148  {
1149    "StartAt": "Pass State 123",
1150  "States": {
1151 +      "Pass State 123": {
1152        "Type": "Pass",
1153       "End": true
1154      }
1155    }
1156  ],
1157  {
1158    "StartAt": "Pass State 124",
1159  "States": {
1160 +      "Pass State 124": {
1161        "Type": "Pass",
1162       "End": true
1163      }
1164    }
1165  ],
1166  {
1167    "StartAt": "Pass State 125",
1168  "States": {
1169 +      "Pass State 125": {
1170        "Type": "Pass",
1171       "End": true
1172      }
1173    }
1174  ],
1175  {
1176    "StartAt": "Pass State 126",
1177  "States": {
1178 +      "Pass State 126": {
1179        "Type": "Pass",
1180       "End": true
1181      }
1182    }
1183  ],
1184  {
1185    "StartAt": "Pass State 127",
1186  "States": {
1187 +      "Pass State 127": {
1188        "Type": "Pass",
1189       "End": true
1190      }
1191    }
1192  ],
1193  {
1194    "StartAt": "Pass State 128",
1195  "States": {
1196 +      "Pass State 128": {
1197        "Type": "Pass",
1198       "End": true
1199      }
1200    }
1201  ],
1202  {
1203    "StartAt": "Pass State 129",
1204  "States": {
1205 +      "Pass State 129": {
1206        "Type": "Pass",
1207       "End": true
1208      }
1209    }
1210  ],
1211  {
1212    "StartAt": "Pass State 130",
1213  "States": {
1214 +      "Pass State 130": {
1215        "Type": "Pass",
1216       "End": true
1217      }
1218    }
1219  ],
1220  {
1221    "StartAt": "Pass State 131",
1222  "States": {
1223 +      "Pass State 131": {
1224        "Type": "Pass",
1225       "End": true
1226      }
1227    }
1228  ],
1229  {
1230    "StartAt": "Pass State 132",
1231  "States": {
1232 +      "Pass State 132": {
1233        "Type": "Pass",
1234       "End": true
1235      }
1236    }
1237  ],
1238  {
1239    "StartAt": "Pass State 133",
1240  "States": {
1241 +      "Pass State 133": {
1242        "Type": "Pass",
1243       "End": true
1244      }
1245    }
1246  ],
1247  {
1248    "StartAt": "Pass State 134",
1249  "States": {
1250 +      "Pass State 134": {
1251        "Type": "Pass",
1252       "End": true
1253      }
1254    }
1255  ],
1256  {
1257    "StartAt": "Pass State 135",
1258  "States": {
1259 +      "Pass State 135": {
1260        "Type": "Pass",
1261       "End": true
1262      }
1263    }
1264  ],
1265  {
1266    "StartAt": "Pass State 136",
1267  "States": {
1268 +      "Pass State 136": {
1269        "Type": "Pass",
1270       "End": true
1271      }
1272    }
1273  ],
1274  {
1275    "StartAt": "Pass State 137",
1276  "States": {
1277 +      "Pass State 137": {
1278        "Type": "Pass",
1279       "End": true
1280      }
1281    }
1282  ],
1283  {
1284    "StartAt": "Pass State 138",
1285  "States": {
1286 +      "Pass State 138": {
1287        "Type": "Pass",
1288       "End": true
1289      }
1290    }
1291  ],
1292  {
1293    "StartAt": "Pass State 139",
1294  "States": {
1295 +      "Pass State 139": {
1296        "Type": "Pass",
1297       "End": true
1298      }
1299    }
1300  ],
1301  {
1302    "StartAt": "Pass State 140",
1303  "States": {
1304 +      "Pass State 140": {
1305        "Type": "Pass",
1306       "End": true
1307      }
1308    }
1309  ],
1310  {
1311    "StartAt": "Pass State 141",
1312  "States": {
1313 +      "Pass State 141": {
1314        "Type": "Pass",
1315       "End": true
1316      }
1317    }
1318  ],
1319  {
1320    "StartAt": "Pass State 142",
1321  "States": {
1322 +      "Pass State 142": {
1323        "Type": "Pass",
1324       "End": true
1325      }
1326    }
1327 
```

```
42             "Next":"Transcribe Failed"
43         }
44     ],
45     "Default":"Wait for Transcribe"
46 ),
47 "Transcript Available": {
48     "Type": "Parallel",
49     "Branches": [
50     {
51         "StartAt": "Translate Text", 
52         "States": {
53             "Translate Text": {
54                 "Type": "Task",
55                 "Resource": "arn:aws:states:::lambda:invoke",
56                 "Parameters": {
57                     "FunctionName": "arn:aws:lambda:us-east-
58 1:756325582597:function:translate-lambda:$LATEST",
59                     "Payload": {
60                         "Input.$": "$"
61                     }
62                 },
63             },
64         }
65     }
66   }
67 }
```



Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc.

28. In line 62, replace **Next: NEXT_STATE** with **End: true**.

```
48     "Type": "Parallel",
49     "Branches": [
50       {
51         "StartAt": "Translate Text",
52         "States": {
53           "Translate Text": {
54             "Type": "Task",
55             "Resource": "arn:aws:states:::lambda:invoke",
56             "Parameters": {
57               "FunctionName": "arn:aws:lambda:us-east-
1:756325582597:function:translate-lambda:$LATEST",
58             "Payload": {
59               "Input.$": "$"
60             }
61           },
62           "End": true
63         }
64       }
65     ],
66   {
67     "StartAt": "Comprehend Text",
```

Feedback English (US) ▾

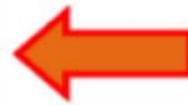
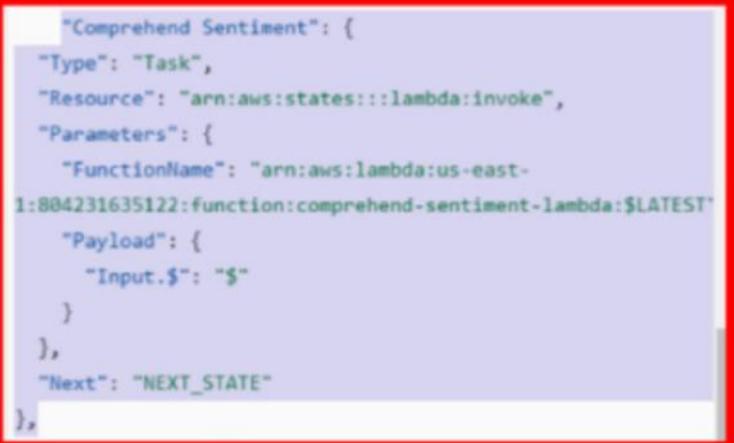
© 2008 - 2021, Amazon Web Services

29. Copy the **Comprehend Sentiment** block code. Remove it from its current position.

aws Services ▾ Search for services, features, marketplace products

```
72     }
73   }
74 }
75 ],
76 "Next": "NEXT_STATE"
77 },
78 *   "Comprehend Sentiment": {
79   "Type": "Task",
80   "Resource": "arn:aws:states:::lambda:invoke",
81   "Parameters": {
82     "FunctionName": "arn:aws:lambda:us-east-
1:804231635122:function:comprehend-sentiment-lambda:$LATEST"
83   "Payload": {
84     "Input.$": "$"
85   }
86 },
87 "Next": "NEXT_STATE"
88 },
89 *   "Transcribe Failed": {
90   "Type": "Fail"
91 }
```

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services



30. Remove **Pass State 2** from lines 69 to 72. Paste the **Comprehend Sentiment** block code.

aws Services ▾ Search for services, features, marketplace products

```
63   ],
64   }
65   ],
66   {
67     "StartAt": "Pass State 2",
68   "States": [
69 *     "Pass State 2": {
70       "Type": "Pass",
71       "End": true
72     }
73   ],
74   ]
75 },
76 "Next": "NEXT_STATE"
77 },
78 *   "Comprehend Sentiment": {
79   "Type": "Task",
80   "Resource": "arn:aws:states:::lambda:invoke",
81   "Parameters": {
82     "FunctionName": "arn:aws:lambda:us-east-
1:804231635122:function:comprehend-sentiment-lambda:$LATEST"
83 }
```

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services



31. Replace **StartAt** with **Comprehend Sentiment**.



Services ▾

Search for services, features, marketplace products

```
58    "Payload": {  
59        "Input.$": "$"  
60    }  
61 },  
62 "End": true  
63 }  
64 }  
65 },  
66 {  
67     "StartAt": "Comprehend Text",  
68     "States": {  
69         "Comprehend Text": {  
70             "Type": "Task",  
71             "Resource": "arn:aws:states:::lambda:invoke",  
72             "Parameters": {  
73                 "FunctionName": "arn:aws:lambda:us-east-  
1:756325582597:function:comprehend-sentiment-lambda:$LATEST  
74         "Payload": {  
75             "Input.$": "$"  
76         }  
77     },  
78 }
```



Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc.

32. In line 78, replace **Next: NEXT_STATE** with **End: true**.

AWS Services Search for services, features, marketplace products

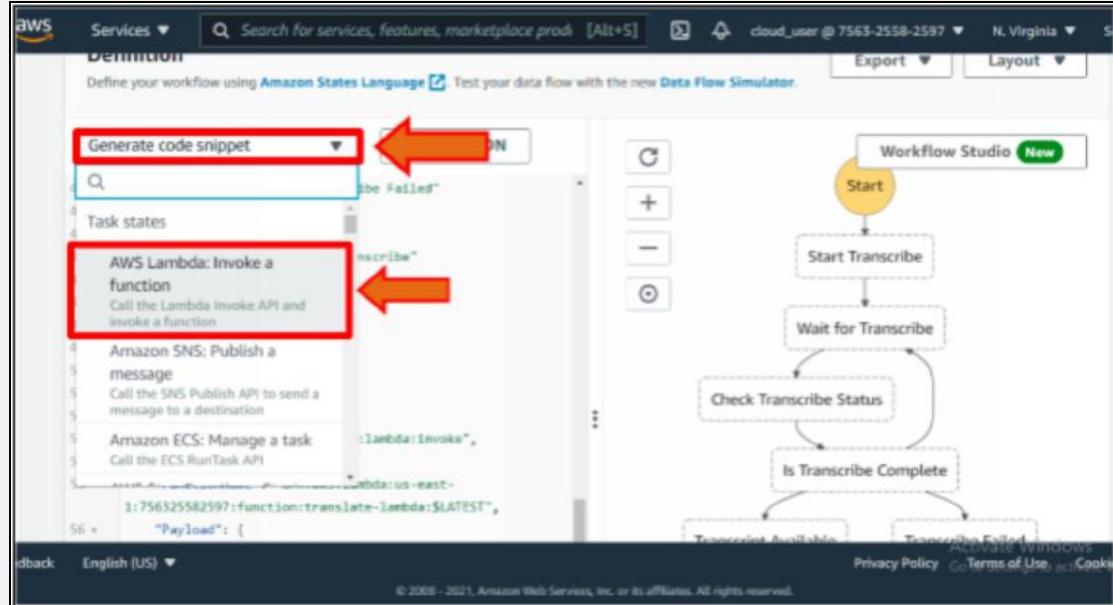
```

63     }
64     }
65     },
66     {
67         "StartAt": "Comprehend Text",
68     +
69         "States": {
70             "Comprehend Text": {
71                 "Type": "Task",
72                 "Resource": "arn:aws:states:::lambda:invoke",
73                 "Parameters": {
74                     "FunctionName": "arn:aws:lambda:us-east-
75                         1:756325582597:function:comprehend-sentiment-lambda:$LATEST"
76                     "Payload": {
77                         "Input.$": "$"
78                     }
79                 }
80             }
81         }
82     ],

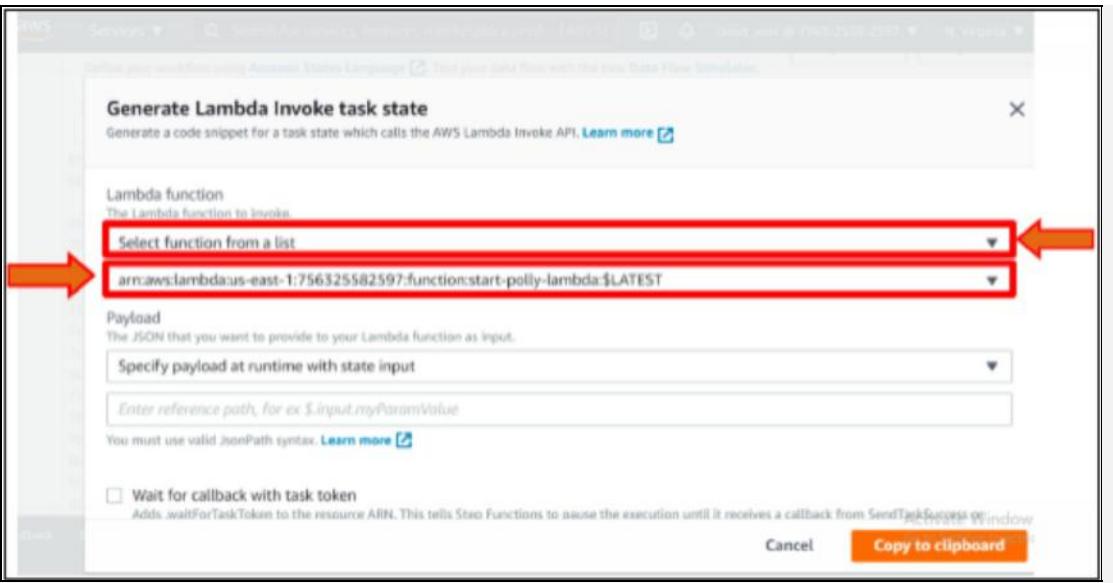
```

edbck English (US) © 2008 - 2021, Amazon Web Services

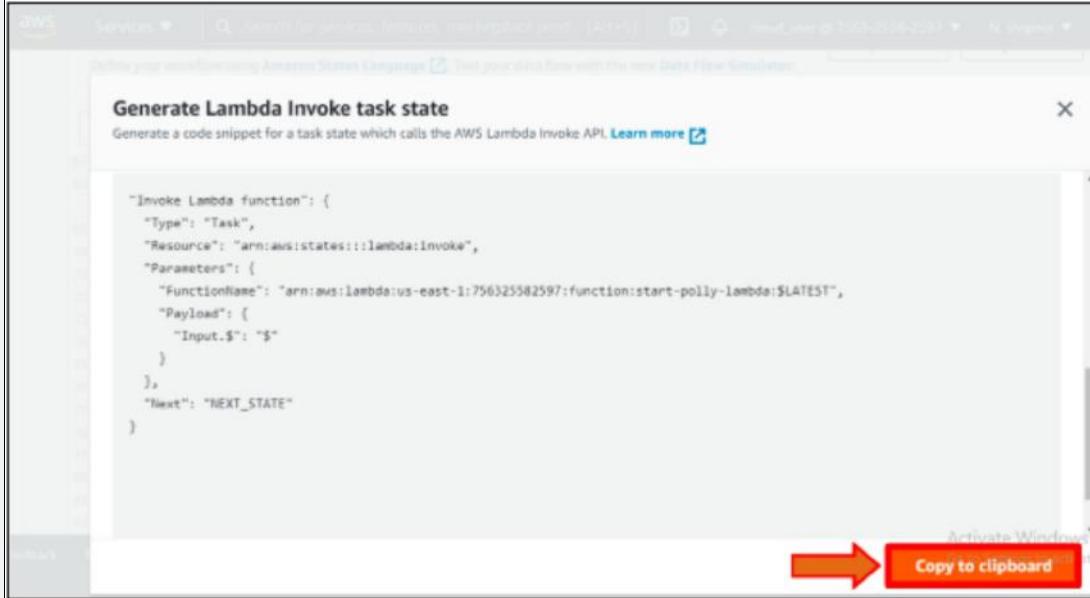
33. Now convert the Translated Text to Audio.
34. In the **Step Function** tab, from **Generate a code snippet**, select **AWS Lambda: Invoke a function**.



35. From the Lambda function dropdown menu, select **Start-Polly Lambda Function**.



36. Click **Copy to Clipboard**.

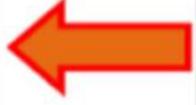


37. Paste the **Start-Polly** code block on line 85.
38. Replace **Invoke Lambda Function** with **Convert Text to Speech**.

aws Services ▾ Search for services, features, marketplace products

```
81     }
82   ],
83   "Next": "Convert Text to Speech"
84 },
85 "Convert Text to Speech": {
86   "Type": "Task",
87   "Resource": "arn:aws:states:::lambda:invoke",
88   "Parameters": {
89     "FunctionName": "arn:aws:lambda:us-east-
1:756325582597:function:start-polly-lambda:$LATEST",
90   "Payload": {
91     "Input.$": "$"
92   }
93 },
94 "End": true
95 },
96   "Transcribe Failed": [
97     {
98       "Type": "Fail"
99     }
100 ]}
```

Feedback English (US) ▾ © 2006 - 2021, Amazon Web Services



39. Replace the **Next: NEXT_STATE** with **Next: Convert Text to Speech**.

aws Services ▾ Search for services, features, marketplace products

```
81     }
82     },
83     "Next": "Convert Text to Speech" 
84   },
85   "Convert Text to Speech": {
86     "Type": "Task",
87     "Resource": "arn:aws:states:::lambda:invoke",
88     "Parameters": {
89       "FunctionName": "arn:aws:lambda:us-east-
1:756325582597:function:start-polly-lambda:$LATEST",
90       "Payload": {
91         "Input.$": "$"
92       }
93     },
94     "End": true
95   },
96   "Transcribe Failed": {
97     "Type": "Fail"
98   }
99 }
100 }
```

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services

40. In the end, the block replaces **Next: NEXT_STATE** with **End: true**.

aws Services ▾ Q Search for services, features, marketplace products

```
81      }
82    ],
83    "Next": "Convert Text to Speech"
84  },
85  "Convert Text to Speech": {
86    "Type": "Task",
87    "Resource": "arn:aws:states:::lambda:invoke",
88    "Parameters": {
89      "FunctionName": "arn:aws:lambda:us-east-1:756325582597:function:start-polly-lambda:$LATEST",
90      "Payload": {
91        "Input.$": "$"
92      }
93    },
94    "End": true // Red box and arrow here
95  },
96  "Transcribe Failed": {
97    "Type": "Fail"
98  }
99 }
100 }
```

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc.

41. Click Format JSON.

aws Services ▾ Q Search for services, features, marketplace products [Alt+5] cloud_user @ 2857-1548-9524 N. Virginia ⓘ

Edit Polyglot Pipeline

Code snippet copied to clipboard. X

Definition
Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Generate code snippet Format JSON

Export ▾ Layout ▾

```
68  "States": {
69    "Comprehend Text": {
70      "Type": "Task",
71      "Resource": "arn:aws:states:::lambda:invoke",
72      "Parameters": {
73        "FunctionName": "arn:aws:lambda:us-east-1:756325582597:function:comprehend-sentiment-lambda:$LATEST",
74        "Payload": {
```

Workflow Studio

START
Start Transcribe
Wait for Transcribe
Check Transcribe Status

ACTIVATE WINDOWS
Privacy Policy Go to Terms of Use Go to Cookie

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

42. On the right-hand side of the screen, you will observe the change in the block diagram.

The screenshot shows the AWS Lambda function code on the left and the Workflow Studio diagram on the right. A large red arrow points from the code area towards the diagram.

```

    "States": {
      "Comprehend Text": {
        "Type": "Task",
        "Resource": "arn:aws:states:::lambda:invoke",
        "Parameters": {
          "FunctionName": "arn:aws:lambda:us-east-1:1756325582597:function:translate"
        },
        "Payload": {
          "Input.$": "$"
        }
      }
    }
  }
}
{
  "Next": "Convert Text to Speech"
},
"Convert Text to Speech": {

```

Workflow Studio Diagram:

```

graph TD
    Start((Start)) --> StartTranscribe[Start Transcribe]
    StartTranscribe --> WaitForTranscribe[Wait for Transcribe]
    WaitForTranscribe --> CheckTranscribeStatus[Check Transcribe Status]
    CheckTranscribeStatus --> IsTranscribeComplete{Is Transcribe Complete?}
    IsTranscribeComplete -- No --> TranscribeFailed[Transcribe Failed]
    IsTranscribeComplete -- Yes --> TranslateText[Translate Text]
    TranslateText --> ComprehendText[Comprehend Text]
    ComprehendText --> ConvertTextToSpeech[Convert Text to Speech]
    ConvertTextToSpeech --> End((End))
    TranscribeFailed --> End

```

43. Click the **Save** button.

The screenshot shows the Step Functions 'Edit Polyglot-Pipeline' page. A red arrow points to the 'Save' button.

Definition:
Define your workflow using [Amazon States Language](#). Test your data flow with the new [Data Flow Simulator](#).

Code snippet copied to clipboard.

```

    "States": {
      "Comprehend Text": {
        "Type": "Task",
        "Resource": "arn:aws:states:::lambda:invoke",
        "Parameters": {
          "FunctionName": "arn:aws:lambda:us-east-1:1756325582597:function:translate"
        },
        "Payload": {
          "Input.$": "$"
        }
      }
    }
  }
}
{
  "Next": "Convert Text to Speech"
},
"Convert Text to Speech": {

```

44. After that, upload the Audio file and watch the magic.

Step 2: Upload an Audio File on S3 Bucket

1. Go to the S3 tab.
2. Select the bucket whose Name starts with **input**.

The screenshot shows the Amazon S3 console. A red arrow points to the 'input-clip-3280...' bucket in the list.

Buckets:

- input-clip-3280-5d7efcbeff79c033d87deecf13cafa (Selected)
- output-clip-3280-5d7efcbeff79c033d87deecf13cafa

Name	AWS Region	Access	Creation date
input-clip-3280-5d7efcbeff79c033d87deecf13cafa	US East (N. Virginia) us-east-1	Objects can be public	August 3, 2021, 14:57:20 (UTC+05:00) Windows
output-clip-3280-5d7efcbeff79c033d87deecf13cafa	US East (N. Virginia) us-east-1	Objects can be public	August 3, 2021, 14:56:56 (UTC+05:00) Windows

3. Click the **Upload** button.

The screenshot shows the AWS S3 console with a single bucket named 'input-cfst-3280-5d7efc8efcf79d033df7decedf13cafa'. The 'Objects' tab is selected. At the top, there are several actions: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create Folder, and Upload. A search bar below the actions contains the placeholder 'Find objects by prefix...'. Below the search bar is a table header with columns: Name, Type, Last modified, Size, and Storage class. A message in the center of the table says 'No objects' and 'You don't have any objects in this bucket.' An orange arrow points to the 'Upload' button at the bottom right of the table area.

4. Click the **Add files** button.
5. Select the **GreatDayToBeYou.mp3** file and upload it. The files are provided in the link below Github link:
<https://github.com/linuxacademy/content-aws-mls-c01/tree/master/CoordinateAIServicesWithStepFunctions>

The screenshot shows the 'Upload' dialog box for AWS S3. It has a central area for dragging and dropping files or choosing files/folders. Below this is a table titled 'Files and folders' with one item: 'GreatDayToBeYou.mp3' (1 Total, 92.9 KB). The 'Add files' button is highlighted with a red arrow. Another red arrow highlights the row for 'GreatDayToBeYou.mp3' in the file list, which includes columns for Name, Folder, Type, and Size. The 'Destination' section shows the target bucket as 's3://input-cfst-3280-5d7efc8efcf79d033df7decedf13cafa'. The 'Destination details' section provides more information about bucket settings. The footer includes standard AWS links like Feedback, English (US), and Copyright information.

6. Click on the **Upload** button.

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
GreatDayToBeYou.mp3	-	audio/mpeg	92.9 KB

Destination

Destination
s3://input-cfst-3280-5d7efc8efcf79d033df7decedf13cafa

► Destination details
Bucket settings that impact new objects stored in the specified destination.

► Permissions
Grant public access and access to other AWS accounts.

► Properties
Specify storage class, encryption settings, tags, and more.

Cancel **Upload** 

Activate Windows
Go to Settings to activate Windows

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

7. Select Make Public from the dropdown to publicize the file.

Amazon S3 > input-cfst-3280-5d7efc8efcf79d033df7decedf13cafa 

Objects Properties Permissions Metrics Management Access Points

Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. [Learn more](#)

 **Actions**  Create folder  **Upload**

 **GreatDayToBeYou.mp3** mp3 August 3, 2021, 17:02:08 (UTC+05:00) 92.9 KB Standard

Activate Windows
Go to Settings to activate Windows

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

8. Click the Make Public button.

Amazon S3 > input-cfst-3280-5d7efc8efcf79d033df7decedf13cafa > Make public

Make public

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

 When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.

Specified objects

Name	Type	Last modified	Size
GreatDayToBeYou.mp3	mp3	August 3, 2021, 17:02:08 (UTC+05:00)	92.9 KB

 **Make public**

Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

9. Also, listen to the file audio by clicking on the object URL.

Lab+LabServices-Prod-5189
AWS Region: US East (N. Virginia) us-east-1
Last modified: August 3, 2021, 17:02:08 (UTC+05:00)
Size: 92.9 KB
Type: mp3
Key: GreatDayToBeYou.mp3

Object URL: <https://input-csh-3280-5d7efcbeefcf79d033cf7deceff13cafa.s3.amazonaws.com/GreatDayToBeYou.mp3>

10. Click the Polyglot- Pipeline link at the top.

Step Functions > State machines > Polyglot-Pipeline

Edit Polyglot-Pipeline

Code snippet copied to clipboard.

Definition

Define your workflow using Amazon States Language. Test your data flow with the new Data Flow Simulator.

Generate code snippet Format JSON

Workflow Studio New

Start

Starts Transcribe

11. Click the link under Name.

12. The pipeline status is Running.

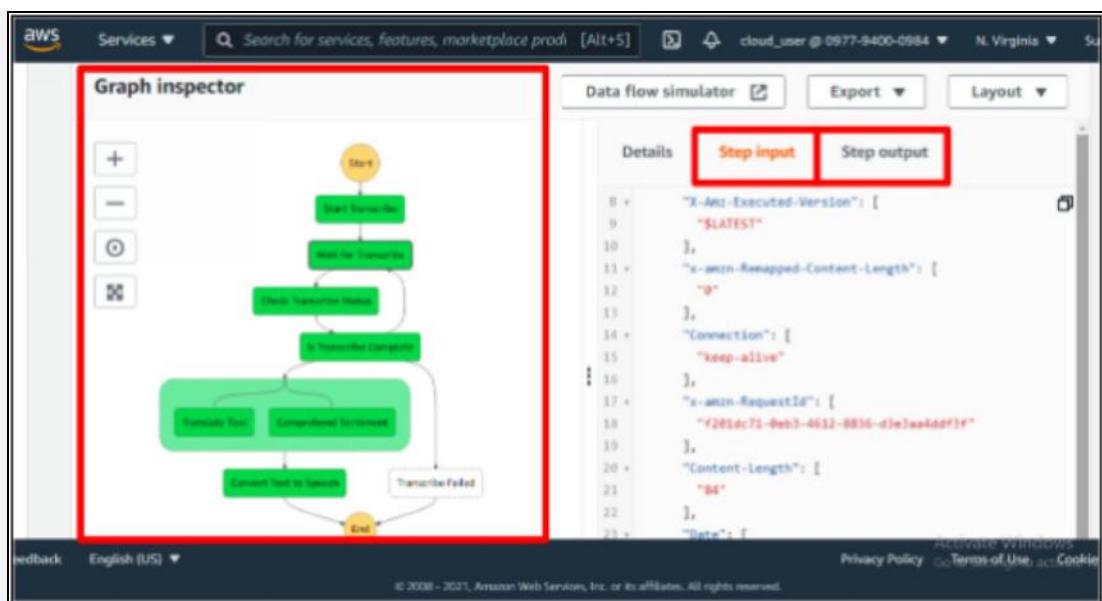
Step Functions

State machines Activities Data flow simulator Feature spotlight Local Development Join our feedback panel

Executions (1)

Name	Status	Started	End Time
e302a01c-fc49-460d-a0d4-7d4d8b6b273e	Running	Aug 3, 2021 05:31:32.766 PM	-

13. View the Step Input and Step Output tabs to view the pipeline's progress.



Step 3: Translate Text on AWS Transcribe

1. Now, go to Amazon Transcribe. Click Launch Amazon Transcribe.

The screenshot shows the Amazon Transcribe landing page. It has a dark header with the service name. Below it is a main section with the heading 'Amazon Transcribe Automatic Speech Recognition'. It includes a brief description of the service and a large 'Create a transcript' button highlighted with a red arrow. There are also links for 'Create a medical transcript' and another 'Create a transcript' button.

2. In the left-hand menu, click Transcription Jobs.
3. Click the link under Name.

The screenshot shows the 'Transcription jobs' page within the Amazon Transcribe service. On the left, there's a sidebar with links like 'Real-time transcription', 'Transcription jobs' (which is highlighted with a red box and has a red arrow pointing to it), 'Custom language model', etc. The main area shows a table of transcription jobs. The first job listed is highlighted with a red box and has a red arrow pointing to it. The table columns include 'Name', 'Status', and 'Language'.

4. Under Transcription preview, view the translated text.

Transcription preview
Select download to save a local copy of the transcription

Text | Audio identification

It is a great day to be you seize the day and believe that you can do anything you set your mind to.

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2006–2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Go back to the AWS S3 tab.
6. Now select the bucket name that starts with the word **Output**.

Amazon S3

Buckets

Access Points Object Lambda Access Points Batch Operations Access analyzer for S3

Block Public Access settings for this account

Storage Lens Dashboards AWS Organizations settings Feature spotlight

AWS Marketplace for S3

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2006–2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon S3

▶ Account snapshot View Storage Lens dashboard

Buckets (2) info Buckets are containers for data stored in S3. Learn more

Name AWS Region Access Creation date

input-cfn-3280-fe71e55563bdfb4d30ba63143cd83568	US East (N. Virginia) us-east-1	Objects can be public	August 3, 2021, 16:32:52 UTC+00:00
output-cfn-3280-fe71e55563bdfb4d30ba63143cd83568	US East (N. Virginia) us-east-1	Objects can be public	August 3, 2021, 16:32:07 UTC+00:00

7. Click on the folder in the **Output** bucket, which should reflect the sentiment of the audio file.

Amazon S3

Buckets

Access Points Object Lambda Access Points Batch Operations Access analyzer for S3

Block Public Access settings for this account

Storage Lens Dashboards AWS Organizations settings Feature spotlight

AWS Marketplace for S3

Feedback English (US) ▾ Privacy Policy Terms of Use Cookie preferences © 2006–2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon S3 > output-cfn-3280-fe71e55563bdfb4d30ba63143cd83568

output-cfn-3280-fe71e55563bdfb4d30ba63143cd83568 info

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name Type Last modified Size Storage class

POSITIVE/	Folder	-	-	-
------------------	--------	---	---	---

8. Click on Actions to make the file public.

Amazon S3 > output-cfst-3280-fe71e55563bdfb4d30ba63143cd83568 > POSITIVE/

POSITIVE/

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. You'll need to explicitly grant them permission. [Learn more](#)

Actions

Query with S3 Select
Edit actions
Rename object
Edit storage class
Edit server-side encryption
Edit metadata
Edit tags
Make public

Name	Type	Last modified	Size	Storage class
GreatDayToBeYou.mp3-7d8477aa-d4fa-40b0-b511-f9793dedc07a.a1c4146d-f600-45f0-a668-5342e0153753.mp3	mp3	August 3, 2021, 17:45:45 (UTC+05:00)	31.4 KB	Standard

9. Click **Make Public** to the public the object.

Amazon S3 > output-cfst-3280-fe71e55563bdfb4d30ba63143cd83568 > POSITIVE/ > Make public

Make public [Info](#)

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

Specified objects

When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.

Name	Type	Last modified	Size
GreatDayToBeYou.mp3-7d8477aa-d4fa-40b0-b511-f9793dedc07a.a1c4146d-f600-45f0-a668-5342e0153753.mp3	mp3	August 3, 2021, 17:45:45 (UTC+05:00)	31.4 KB

Make public

10. Click on the object URL to listen to the translated audio in the Spanish Language.

AWS Services Search for services, features, marketplace products, and docs [Alt+S] cloud_user @ 0377-9400-0394 Gmail

17942
mp3
Key
POSITIVE/GreatDayToBeYou.mp3-7db477aa-d4fa-40b0-b511-49146d-f600-45f0-a668-5342e0153753.mp3

Object URL: <https://output-cfst-3280-fe71e5563bfb4d30be63143cd83.amazonaws.com/POSITIVE/GreatDayToBeYou.mp3-7db477aa-d4fa-40b0-b511-49146d-f600-45f0-a668-5342e0153753.mp3>

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties

Bucket Versioning
When enabled, multiple variants of an object can be stored in the bucket to easily recover from unintended user actions and application failures.
Disabled

Bucket "output-cfst-3280-fe71e5563bfb4d30be63143cd83" doesn't have Bucket Versioning enabled.
We recommend that you enable Bucket Versioning to help protect against unintentionally overwriting or deleting objects. [Learn more](#)

Enable Bucket Versioning

Management configurations

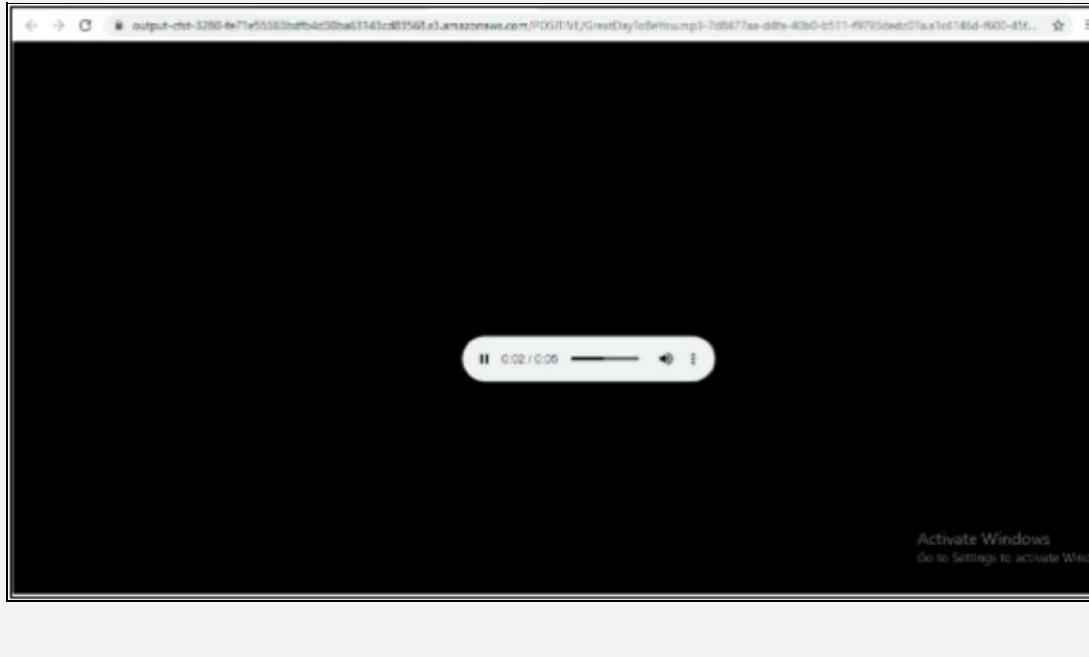
Replication status
When a replication rule is applied to an object the replication status indicates the progress of the operation.
-

[View replication rules](#)

Expiration rule
You can use a lifecycle configuration to define expiration rules to schedule the removal of this object after a pre-defined time period.
-

Expiration date
The object will be permanently deleted on this date.
Activate Windows
Go to Settings to activate Windows

11. The output translates audio.



Github Link: <https://github.com/linuxacademy/content-aws-mls-c01/tree/master/CoordinateAIWithStepFunctions>

Note: If there is trouble executing the code, you also take help from the below-given code.

Code:

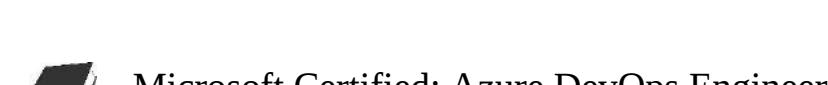
```
{  
  "StartAt": "Start Transcribe",  
  "States": {  
    "Start Transcribe": {  
      "Type": "Task",  
      "Resource": "arn:aws:states:::lambda:invoke",  
      "Parameters": {  
        "FunctionName": "arn:aws:lambda:us-east-  
1:756325582597:function:start-transcribe-lambda:$LATEST",  
        "Payload": {  
          "Input.$": "$"  
        }  
      },  
      "Next": "Wait for Transcribe"  
    },  
    "Wait for Transcribe": {  
      "Type": "Wait",  
      "Seconds": 45,  
      "Next": "Check Transcribe Status"  
    },  
    "Check Transcribe Status": {  
      "Type": "Task",  
      "Resource": "arn:aws:states:::lambda:invoke",  
      "Parameters": {  
        "FunctionName": "arn:aws:lambda:us-east-  
1:756325582597:function:transcribe-status-lambda:$LATEST",  
        "Payload": {  
          "Input.$": "$"  
        }  
      },  
      "Next": "Is Transcribe Complete"  
    },  
    "Is Transcribe Complete": {  
      "Type": "Choice",  
      "Choices": [  
        {  
          "Variable": "$.Payload.TranscriptionJobStatus",  
          "StringEquals": "COMPLETED",  
          "Next": "Transcript Available"  
        },  
        {  
          "Variable": "$.Payload.TranscriptionJobStatus",  
          "StringEquals": "FAILED",  
          "Next": "Transcript Failed"  
        }  
      ]  
    }  
  }  
}
```

```
{  
    "Variable":("$.Payload.TranscriptionJobStatus",  
    "StringEquals":"FAILED",  
    "Next":"Transcribe Failed"  
}  
,  
    "Default":"Wait for Transcribe"  
,  
    "Transcript Available": {  
        "Type": "Parallel",  
        "Branches": [  
            {  
                "StartAt": "Translate Text",  
                "States": {  
                    "Translate Text": {  
                        "Type": "Task",  
                        "Resource": "arn:aws:states:::lambda:invoke",  
                        "Parameters": {  
                            "FunctionName": "arn:aws:lambda:us-east-  
1:756325582597:function:translate-lambda:$LATEST",  
                            "Payload": {  
                                "Input.$": "$"  
                            }  
},  
                    "End": true  
}  
            }  
        },  
        {  
            "StartAt": "Comprehend Text",  
            "States": {  
                "Comprehend Text": {  
                    "Type": "Task",  
                    "Resource": "arn:aws:states:::lambda:invoke",  
                    "Parameters": {  
                        "FunctionName": "arn:aws:lambda:us-east-  
1:756325582597:function:comprehend-sentiment-lambda:$LATEST",  
                        "Payload": {  
                            "Input.$": "$"  
                        }  
},  
                "End": true  
}  
        }  
    }  
}
```

```
],
  "Next": "Convert Text to Speech"
},
"Convert Text to Speech": {
  "Type": "Task",
  "Resource": "arn:aws:states:::lambda:invoke",
  "Parameters": {
    "FunctionName": "arn:aws:lambda:us-east-1:756325582597:function:start-polly-lambda:$LATEST",
    "Payload": {
      "Input.$": "$"
    }
  },
  "End": true
},
"Transcribe Failed": {
  "Type": "Fail"
}
}
```

About Our Products

Other products from IPSpecialist LTD regarding CSP technology are:



 Microsoft Certified: Azure Security Engineer Microsoft Certified: Azure Data Fundamentals Microsoft Certified: Azure AI Fundamentals Microsoft Certified: Azure Database Administrator Associate Google Certified: Associate Cloud Engineer Google Certified: Professional Cloud Developer Microsoft Certified: Azure Data Engineer Associate Microsoft Certified: Azure Data Scientist Ansible Certified: Advanced Automation Oracle Certified: OCI Foundations Associate Oracle Certified: OCI Developer Associate Oracle Certified: OCI Architect Associate



Oracle Certified: OCI Operations Associate



Kubernetes Certified: Application Developer

Other Network & Security related products from IPSpecialist LTD are:

- CCNA Routing & Switching Study Guide
- CCNA Security Second Edition Study Guide
- CCNA Service Provider Study Guide
- CCDA Study Guide
- CCDP Study Guide
- CCNP Route Study Guide
- CCNP Switch Study Guide
- CCNP Troubleshoot Study Guide
- CCNP Security SENSS Study Guide
- CCNP Security SIMOS Study Guide
- CCNP Security SITCS Study Guide
- CCNP Security SISAS Study Guide
- CompTIA Network+ Study Guide
- Certified Blockchain Expert (CBEv2) Study Guide
- EC-Council CEH v10 Second Edition Study Guide
- Certified Blockchain Expert v2 Study Guide