# ADIKAVI NANNAYA UNIVERSITY

## UNIVERSITY COLLEGE OF ENGINEERING

RAJAMAHENDRAVARAM

# PNEUMONIA DETECTION FROM X RAY USING AWS REKOGNITION

Under the esteemed guidance of:
**Professor. P.SURESH VARMA**
Department of CSE

PRESENTED BY:
VAISHNAVI BANDARU
REG.NO: 218297601003
B.TECH

# Index

# Abstract

This project leverages the power of Amazon Rekognition's custom labels feature to develop a machine learning model capable of detecting pneumonia from X-ray images. The model is trained using a dataset of labeled X-ray images, categorized into two distinct classes: "normal" and "pneumonic." By utilizing AWS Cloud services, the project explores the potential of cloud-based image recognition technologies to improve diagnostic accuracy and speed in medical imaging. The ultimate goal of this project is to create an efficient, scalable model that can assist healthcare professionals in identifying pneumonia in patients based on X-ray scans, aiding in early diagnosis and treatment decisions.

# Introduction

- This project harnesses the robust infrastructure of AWS Cloud to create and train a model that distinguishes between two key labels: "normal" and "pneumonic”.
- Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. I am focusing on the “on demand delivery” of the services, Machine Learning services to be exact in this project.
- I am using AWS recognition, a built in ML service in AWS cloud, to train my model
- The suggested approach, makes use of machine learning algorithms to enable precise and prompt detection of whether the X-Ray suggests pneumonia or not.

# Existing System

- Numerous systems have been developed to assist in the detection of diseases such as pneumonia through the use of image recognition and machine learning models.
- The main objective of such systems has been to provide automated support to healthcare professionals by processing X-ray images to determine whether they are normal or show signs of pneumonia.
- Many of them use techniques like convolutional neural networks (CNNs) for analyzing medical images and identifying patterns indicative of certain medical conditions.

## Limitations:

- Dependence on high computational power, limiting usability on standard hardware
- Small level data-set
- Complex setup and maintenance requirements for non-cloud-based models

# Proposed System

- The proposed system focuses on the detection of pneumonia using Amazon Rekognition's custom labels feature combined with the scalable and powerful AWS Cloud infrastructure
- This approach uses Rekognition's built-in image analysis capabilities and custom training features, the system is designed to simplify model-building process.
- This system processes and analyzes a substantial collection of labeled X-ray images to enhance model training and accuracy.

## Advantages:

- Enhanced accuracy
- Accessible and maintainable with minimal on-premise hardware requirements
- Reduced time for training and implementation due to cloud-based resources

# System Analysis

**Functional Requirement**

1. Data Collection and Input: The system should be able to accept and process high-resolution X-ray images.
2. Model Training and Custom Labeling: The system should train a custom model using labeled data sets with "normal" and "pneumonic" categories.
3. Prediction and Classification: The system should analyze input images and classify them accurately into "normal" or "pneumonic" labels.
4. Data Storage and Management: The system should store image securely in the AWS S3 buckets.
5. Reporting and Visualization: The system should generate reports in AWS Cloud Shell

**Non-Functional Requirement**

1. Performance: The system should provide predictions within a reasonable time frame.
2. Accuracy: The prediction accuracy of the system should meet defined thresholds to ensure reliable pneumonia prediction.
3. Usability: The system should have an intuitive and user-friendly.
4. Scalability: The system should seamlessly scale to accommodate large datasets.
5. Reliability: The system should have high availability and be capable of functioning with minimal downtime.

# System Requirements

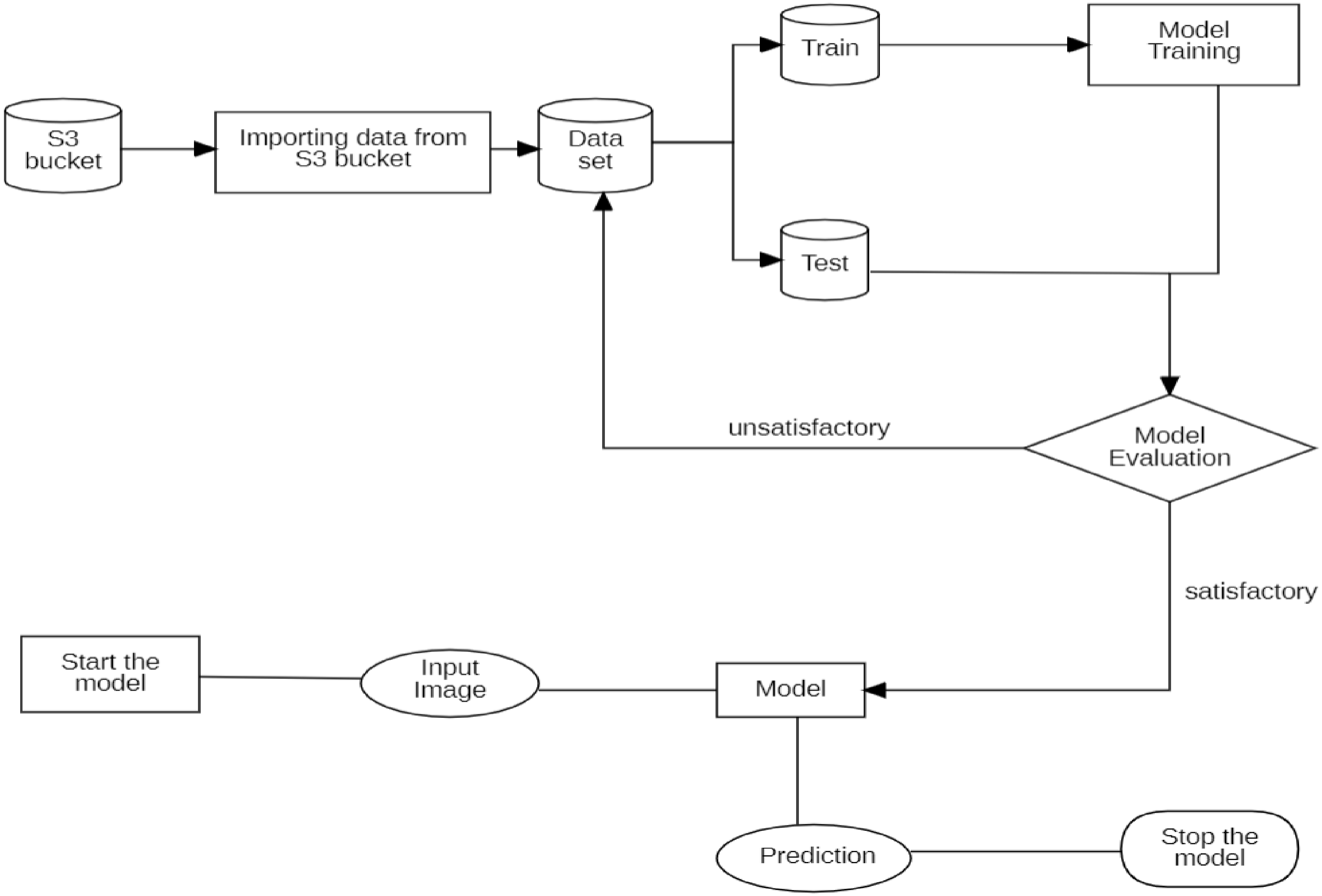
**Hardware:**

- Processor (CPU) -  Intel Core i3 or higher
- Memory (RAM) - 8GB or more
- Storage (HDD/SSD) -512GB SSD or higher

**Software:**

- Windows 10/11, macOS, or Linux for local development
- Any web browser for accessing AWS Management Console
- Access to AWS Rekognition feature
- Access to AWS S3 buckets

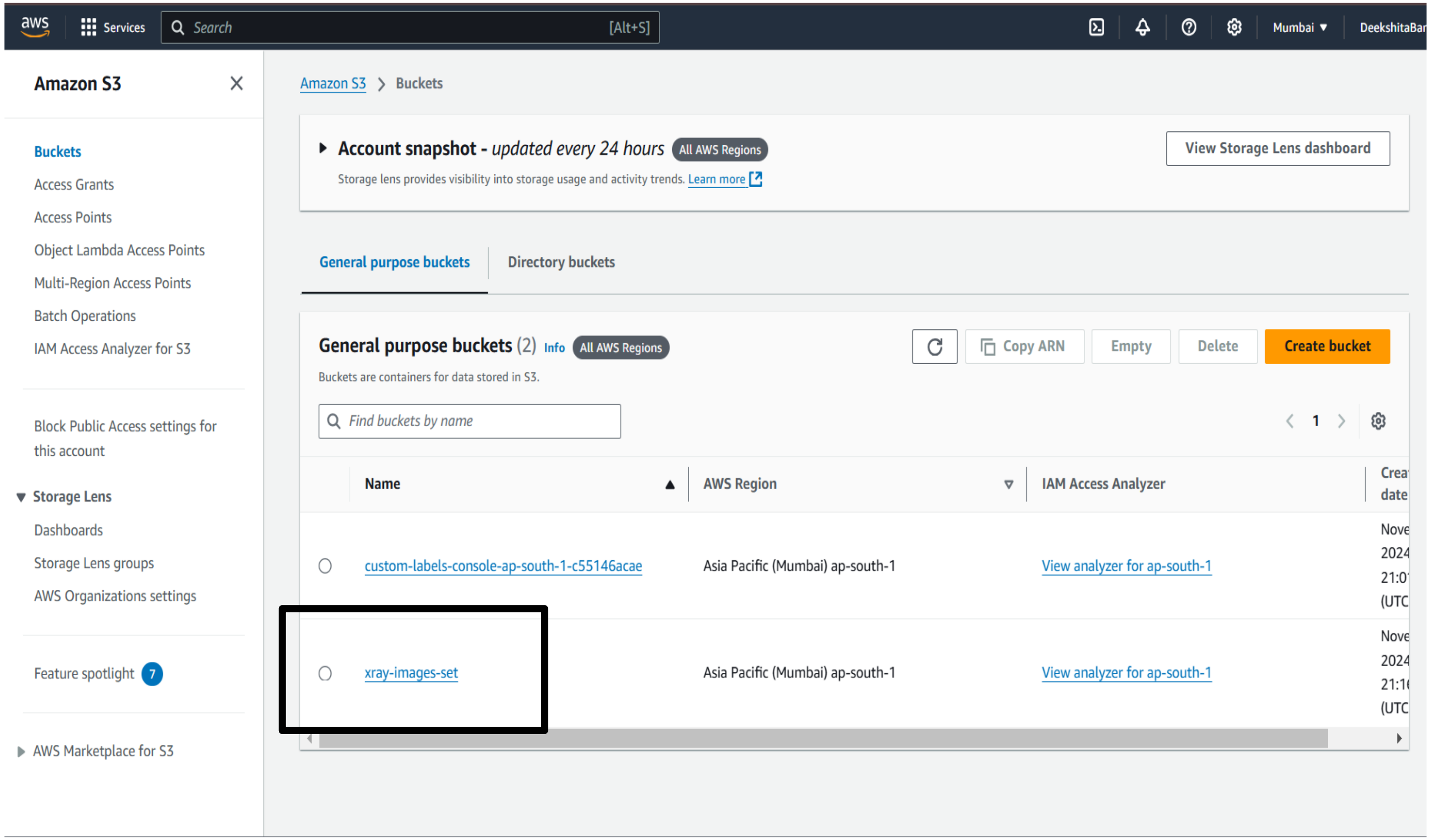
# System Architecture



# Implementation

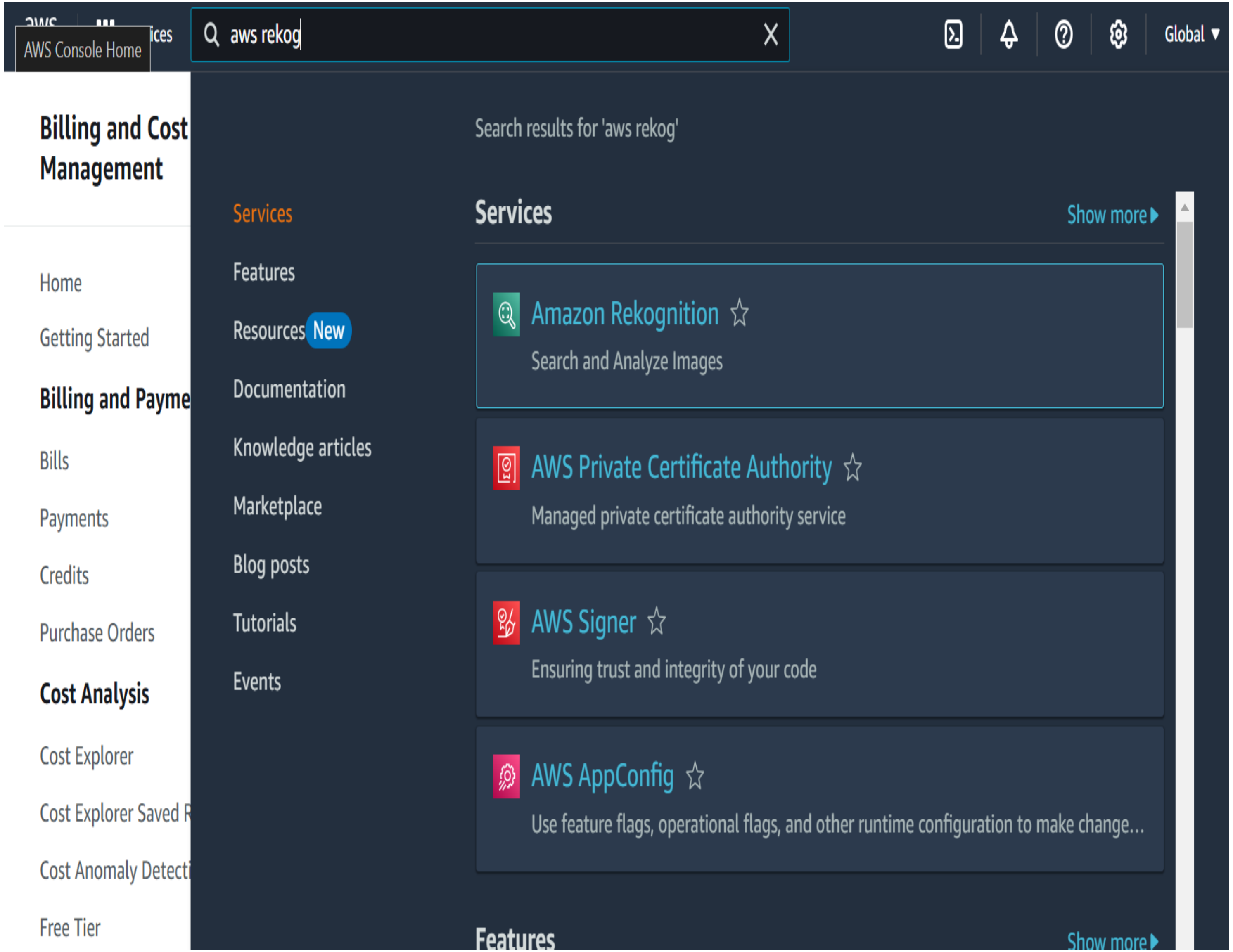
1. Creating an S3 bucket
2. Creating the project
3. Importing dataset
4. Training the model
5. Model Evaluation
6. Deploying the model in cloud shell

# Creating an S3 bucket and uploading data



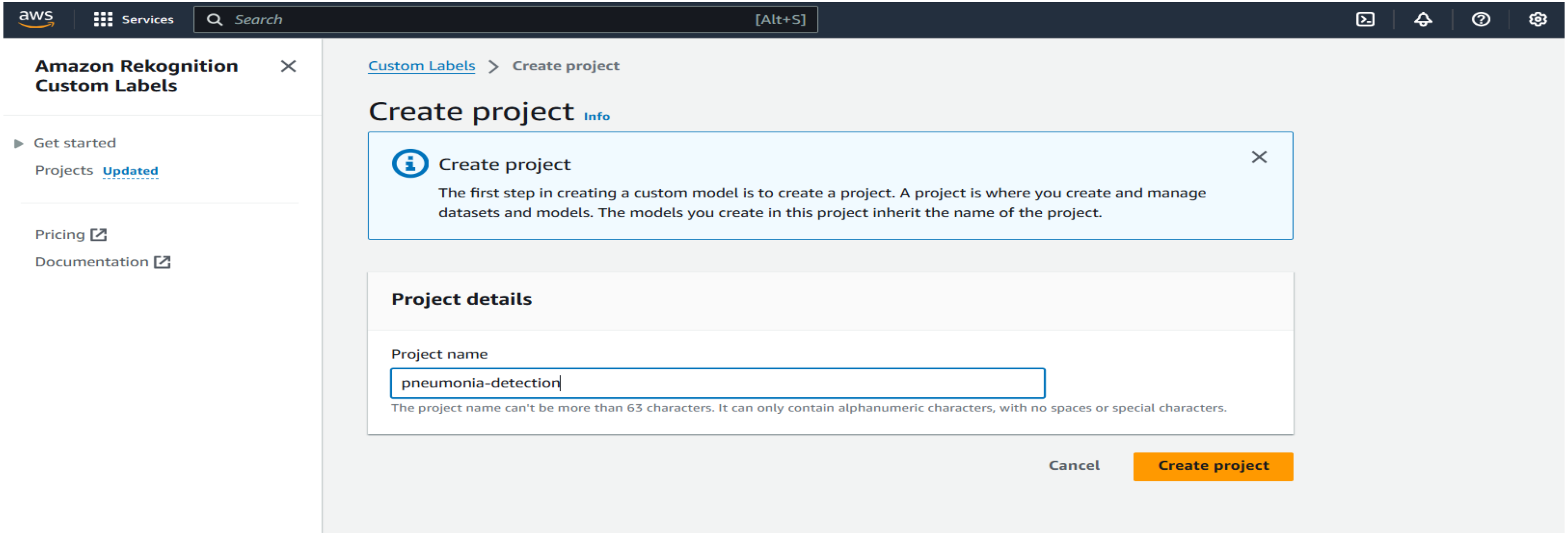

Create an S3 bucket in your AWS management console by navigating to the S3 bucket feature and using "Create Bucket" option. I named my S3 bucket- "Xray-image-set", which will store the Xray images sourced from Kaggle. Ensure that the data is uploaded in labelled folders to make it easier during classification. In this system the data is collected from the Kaggle.

# AWS Rekognition




With Amazon Rekognition Custom Labels, you can identify the objects, logos, and scenes in images that are specific to your business needs. Generating this data can take months to gather, and can require large teams of labelers to prepare it for use in machine learning. Using Amazon Rekognition Custom Labels you can upload a small set of training that are specific to your use case. You can do this by using the easy-to-use console. If your images are already labeled training can begin in a short time. Behind the scenes, Amazon Rekognition Custom Labels automatically loads and inspects the training data, selects the right machine learning algorithms, trains a model, and provides model performance metrics. You can then use your custom model through the Amazon Rekognition Custom Labels API and integrate it into your applications.

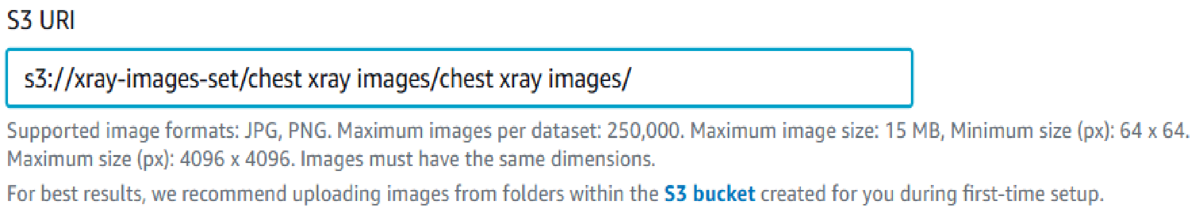
# Creating the project

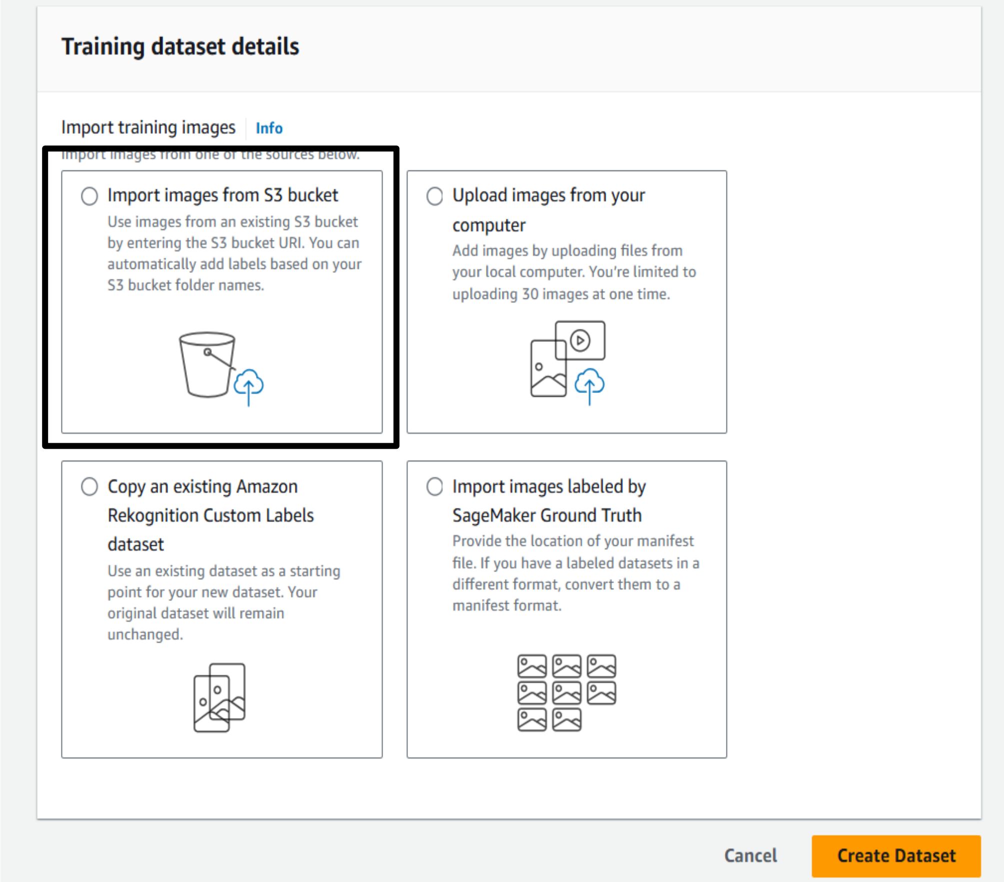


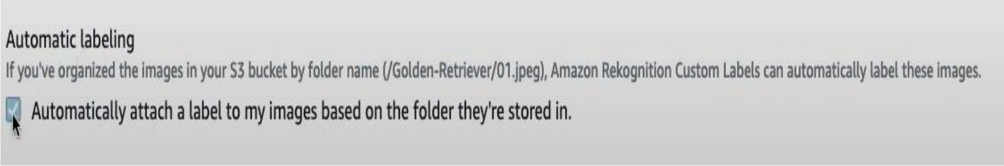

Next step is to navigate to the AWS recognition from search bar and use “Create project” button to create our project with the name “pneumonia-detection”. We need to import dataset and train the model in this project.

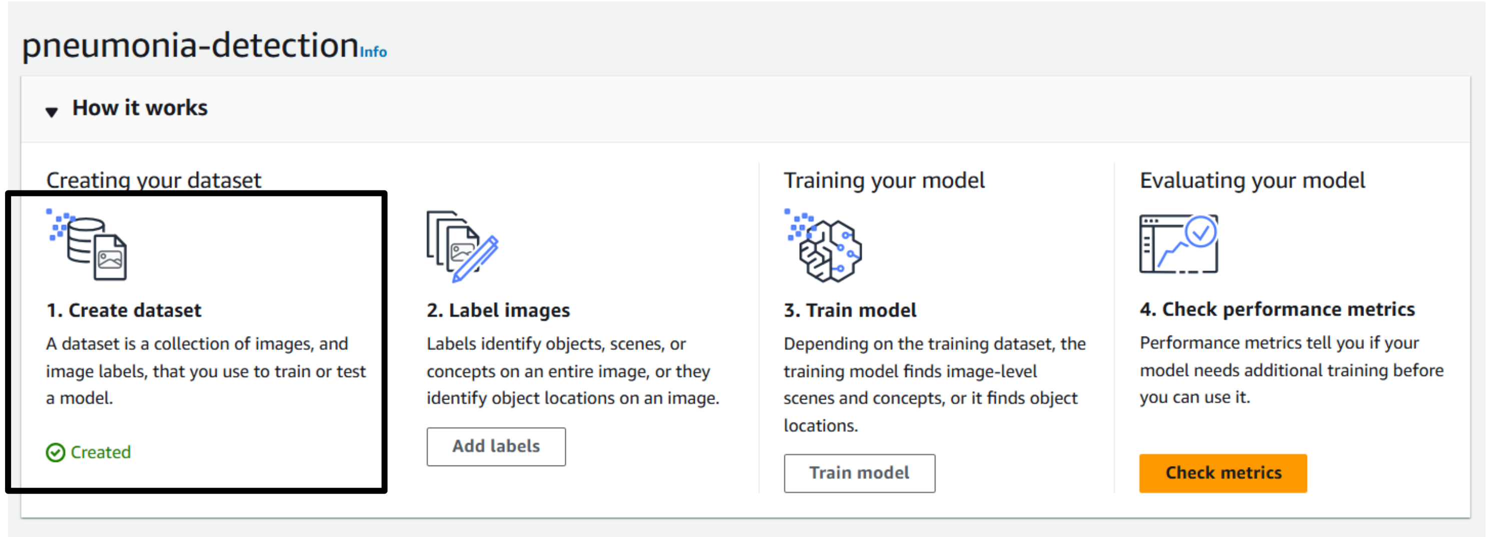
# Importing dataset

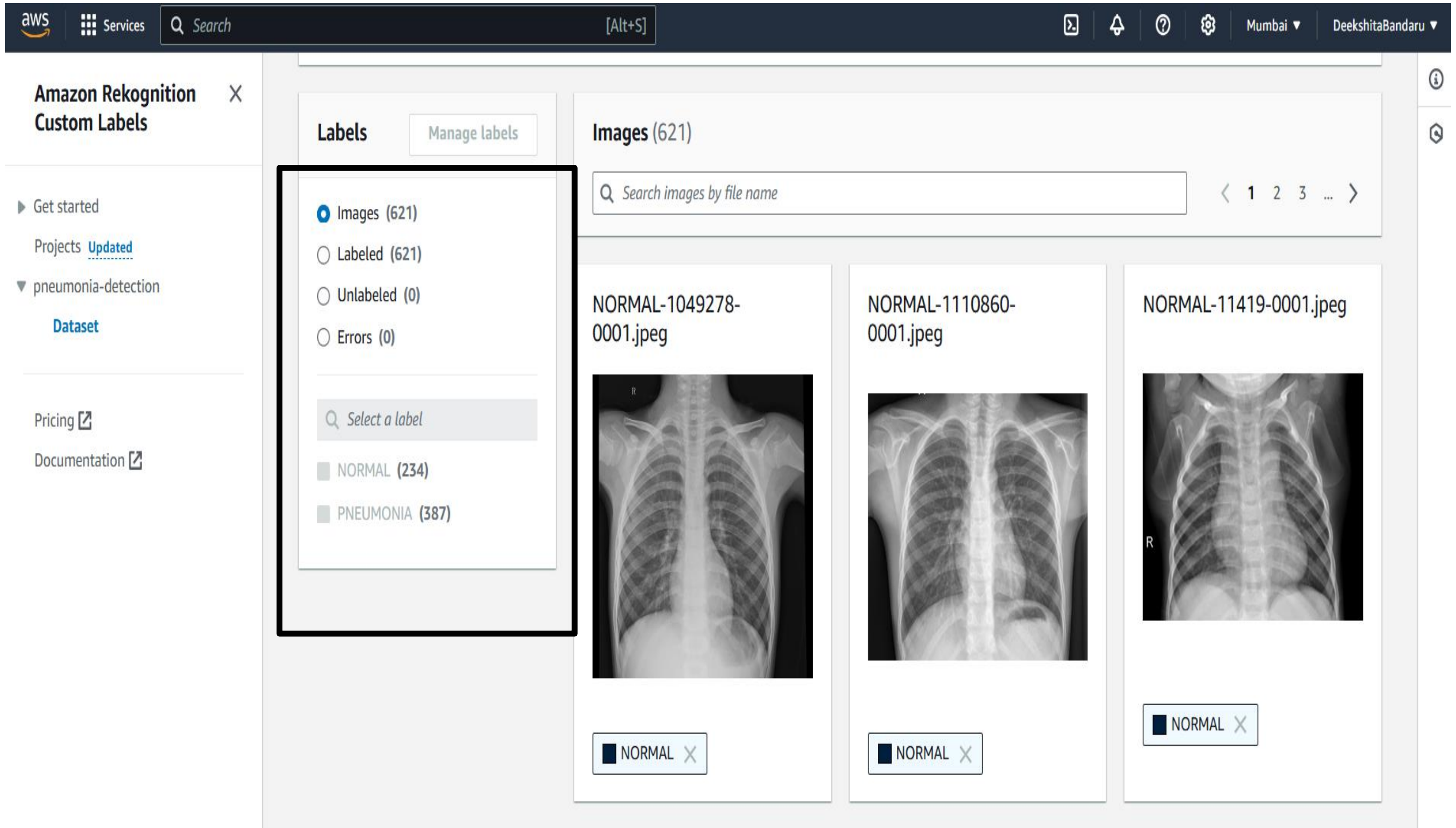
- Next step is to import the data from the S3 bucket to our project “pneumonia-detection”
- Navigate to the “create dataset” feature to create a dataset
- Select “import images from S3 bucket” option
- Provide the S3 URI which can be copied from the S3 bucket properties so that the data from the S3 bucket can be imported.
- Make sure the “automatically attach the labels based on the folders they are stored in” option is enabled for automatic labelling.
- Press “create dataset” and the dataset will be created in a few minutes.



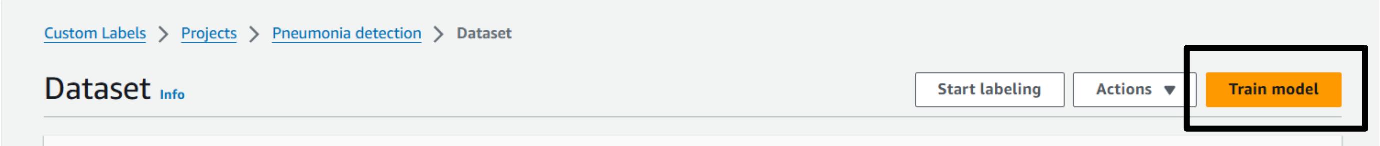



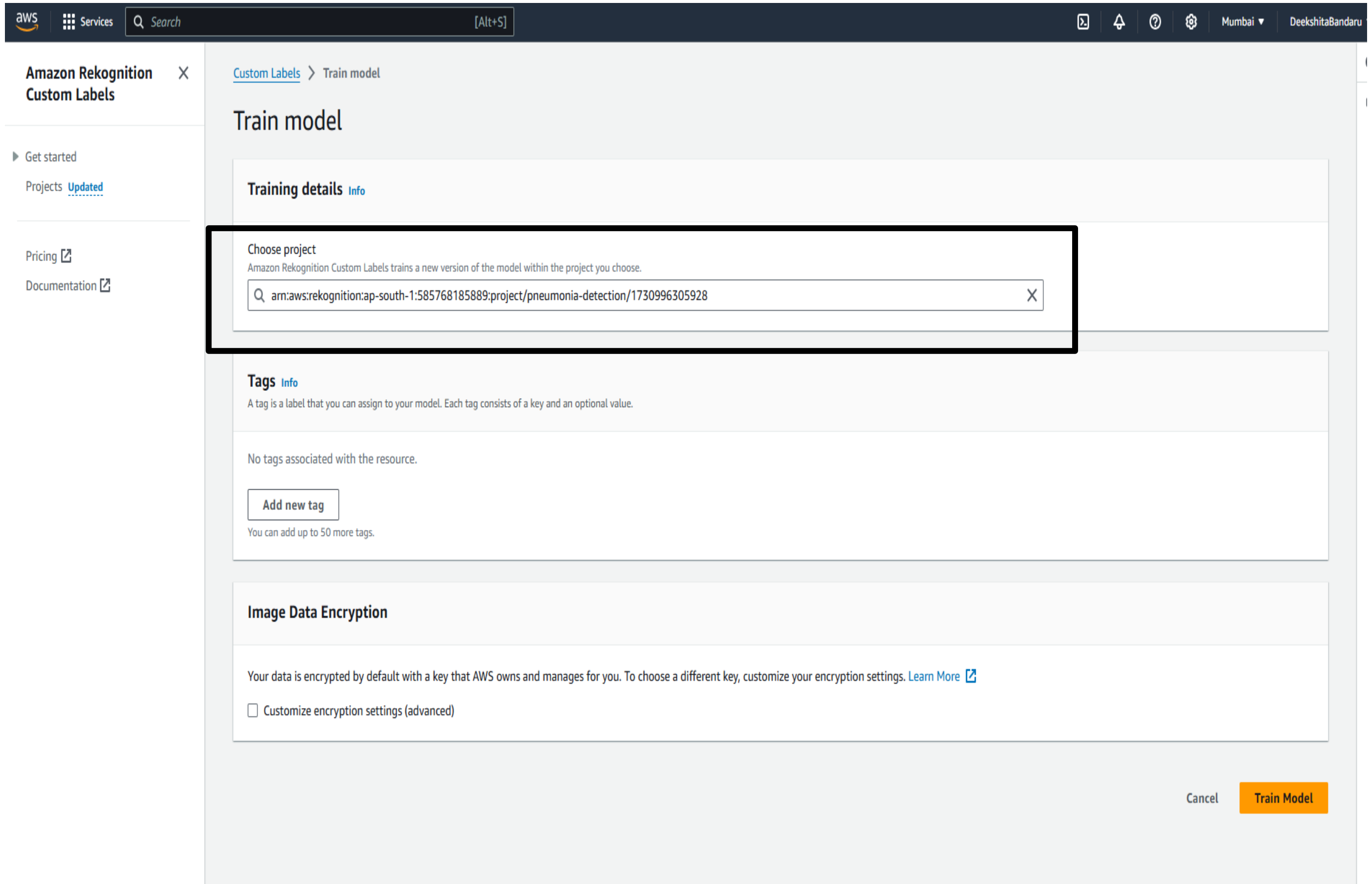


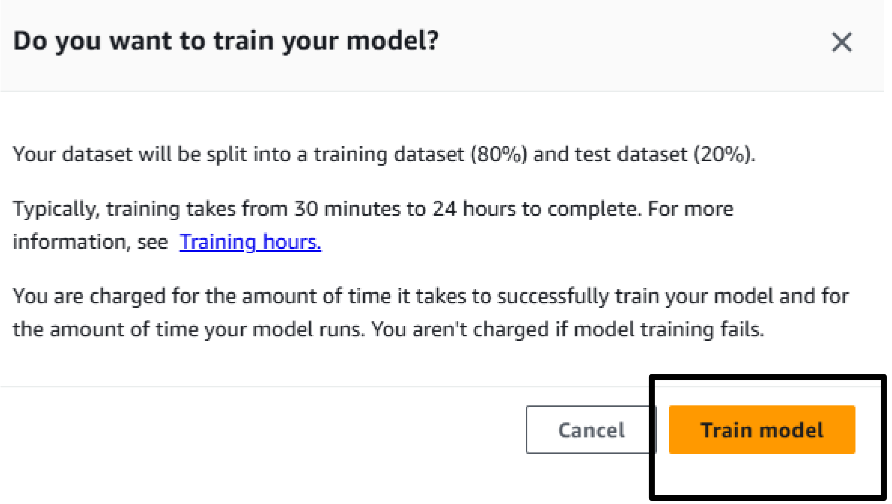




- The dataset is successfully created
- All the data is labeled either as “pneumonia” or “normal” automatically.
- This is because we enabled the “automatic labeling” feature while creating the dataset.
- The entire dataset consists of 621 images
- Out of which 234 are labeled “Normal” and 387 are labeled “pneumonia”

# Training the model

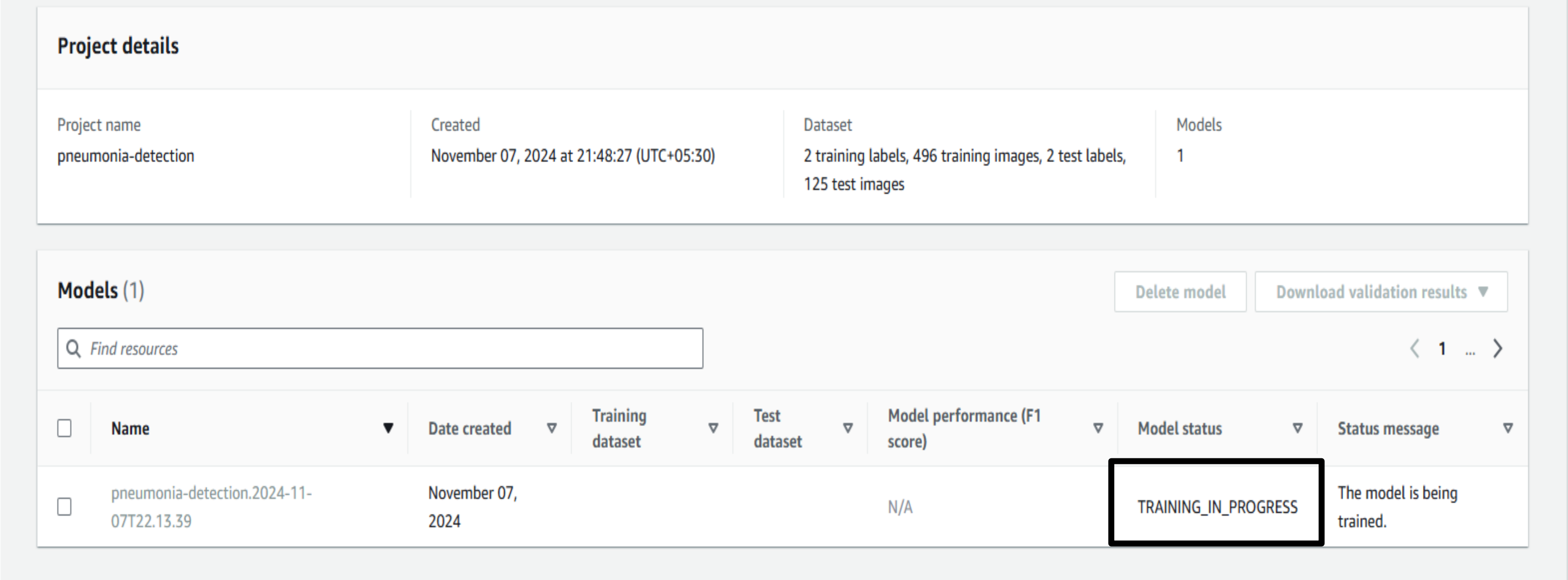


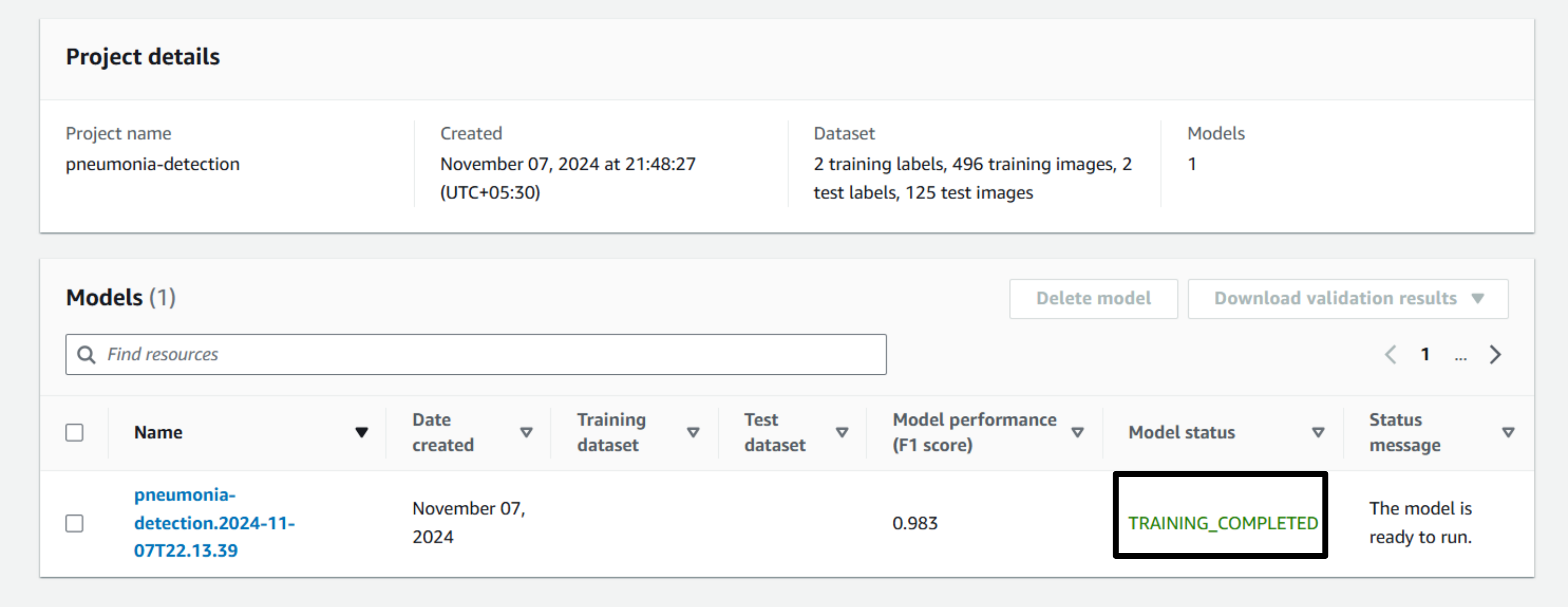



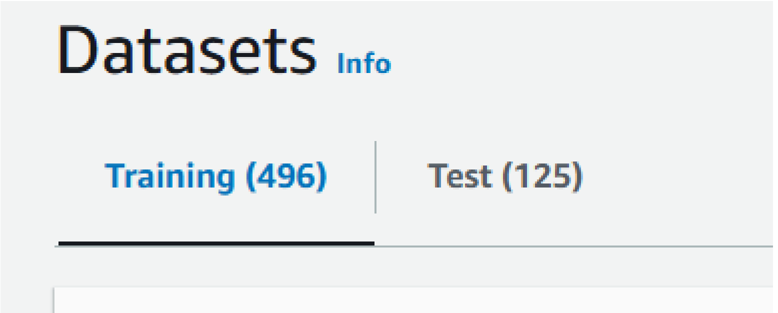

Choose the “train model” option to start the training process. Choose the project in which the model to be trained is present.

The dataset in the project will be split into training dataset (80%) and test dataset (20%).

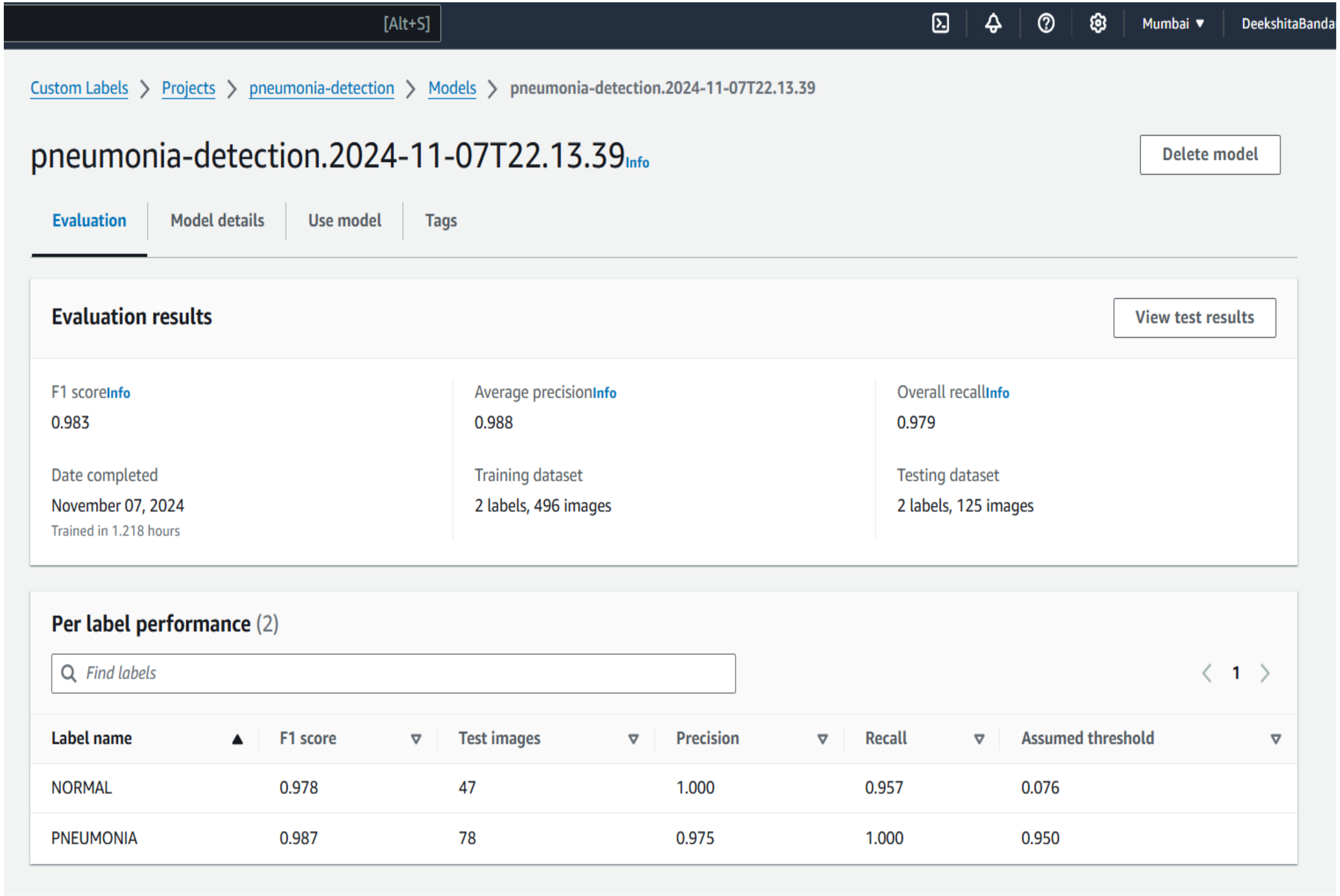


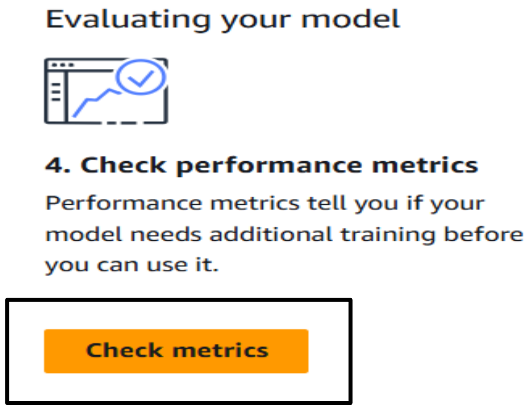




- Once the model training process starts, the model status will be set to “TRAINING_IN_PROGRESS” state.
- After the model is done being trained, the model status changes to “TRAINING_COMPLETED”
- The dataset is automatically split into test and train datasets.
- Now the model is ready to be evaluated.



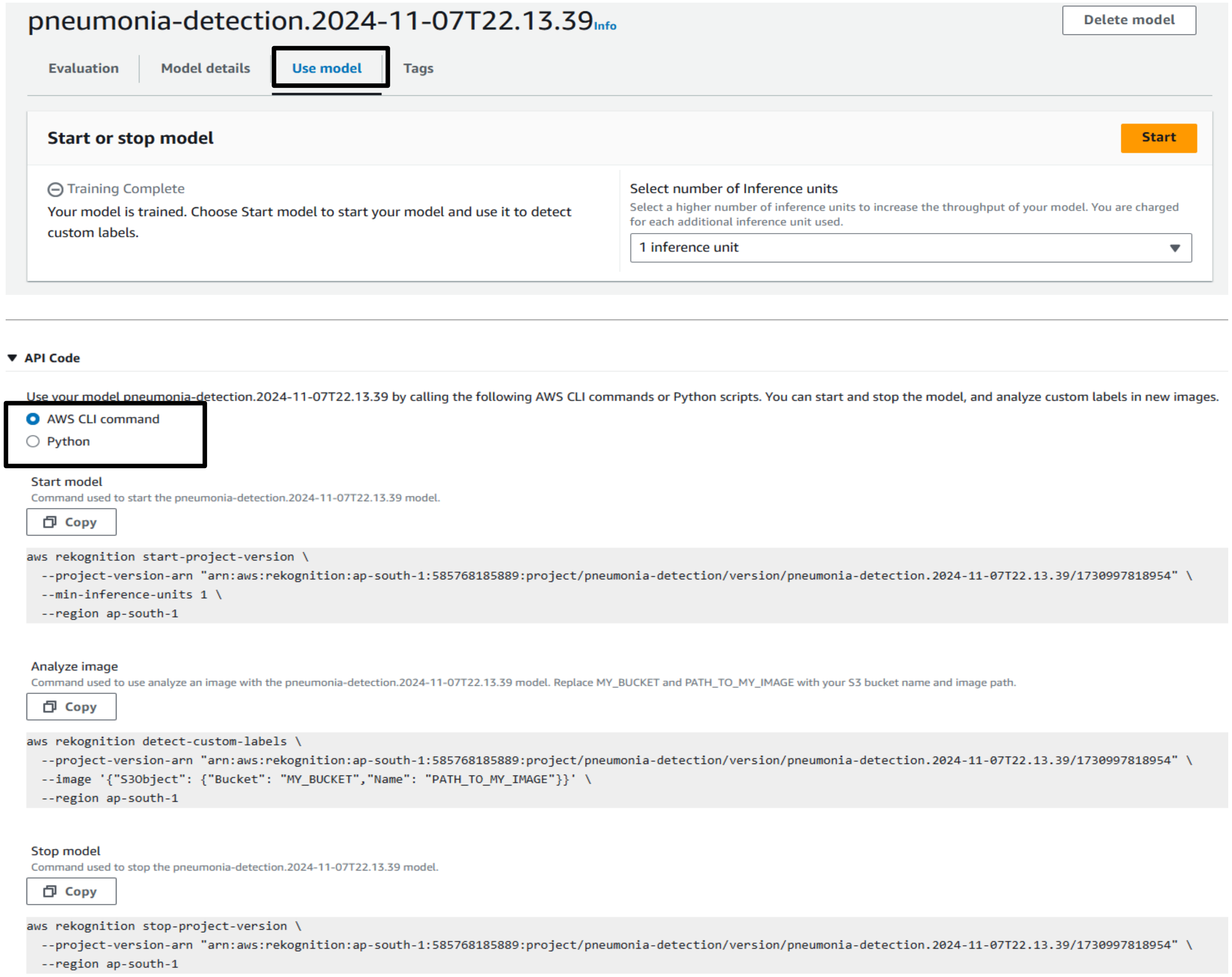
# Evaluating the model



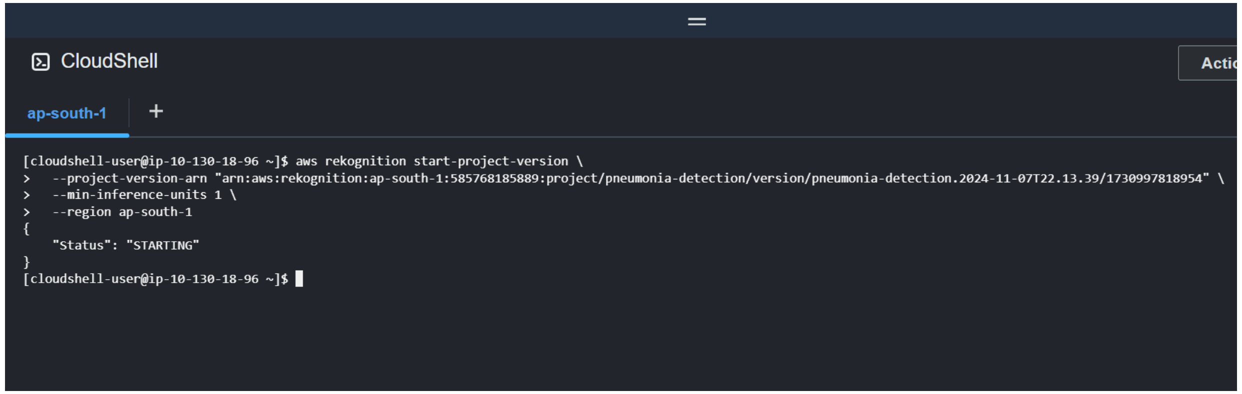




- AWS recognition has a built in feature to evaluate your model.
- Navigate to “check metrics” to get a detailed evaluation report of your model based on the test data.
- The evaluation measures used are F1 score, Precision and Recall.

# Deploying the model in cloud shell

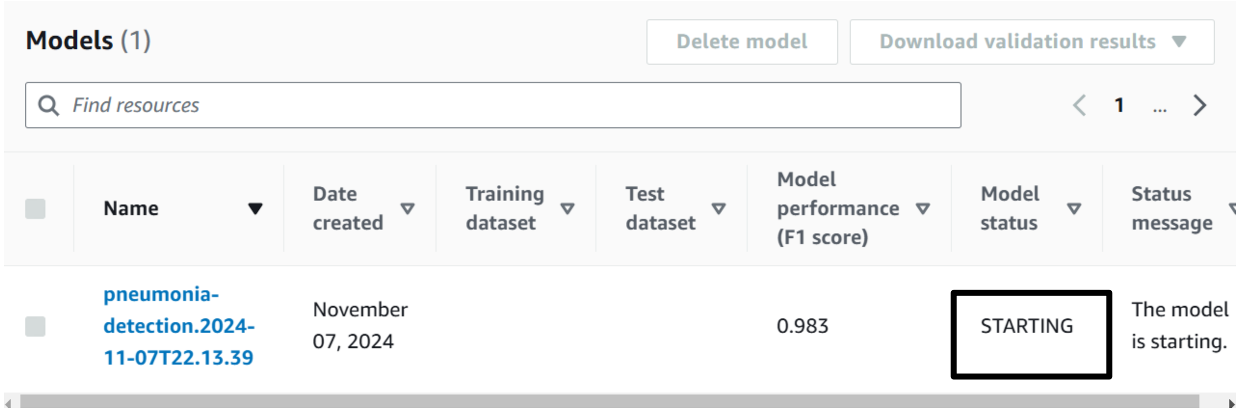



- Navigate to the “Use Model” field.
- Here we have API code available to start the model. Stop the model and analyze image.
- All we have to do is select whether you want to implement the project in python environment or AWS Command line interface (CLI).
- Copy the API codes and paste them in the necessary environments to implement them.
- I am using the AWS CLI to deploy my model.
- For analyzing the image, the bucket name and the path to the image to be tested should be given to the API code, these modifications can be made in any text editor

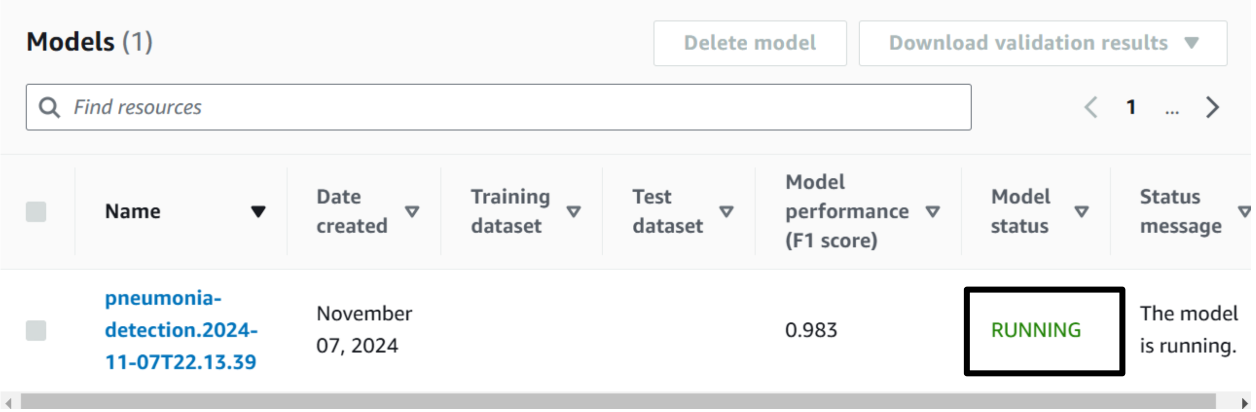


CloudShell
Acti
ap-south-1
+
[cloudshell-user@ip-10-130-18-96 ~]$ aws rekognition start-project-version \
> --project-version-arn "arn:aws:rekognition:ap-south-1:585768185889:project/pneumonia-detection/version/pneumonia-detection.2024-11-07T22.13.39/1730997818954" \
> --min-inference-units 1 \
> --region ap-south-1
{
"Status": "STARTING"
}
[cloudshell-user@ip-10-130-18-96 ~]$

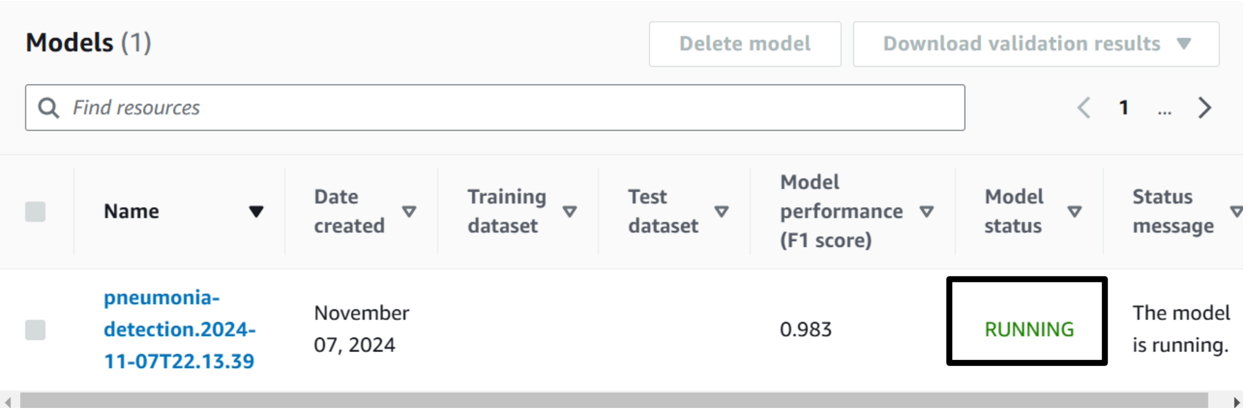

> --region ap-south-1
{
"CustomLabels": [
{
"Name": "PNEUMONIA",
"Confidence": 100.0
}
]
}
[cloudshell-user@ip-10-130-18-96 ~]$ aws rekognition stop-project-version \
> --project-version-arn "arn:aws:rekognition:ap-south-1:585768185889:project/pneumonia-detection/version/pneumonia-detection.2024-11-07T22.13.39/1730997818954" \
> --region ap-south-1
{
"Status": "STOPPING"
}
[cloudshell-user@ip-10-130-18-96 ~]$

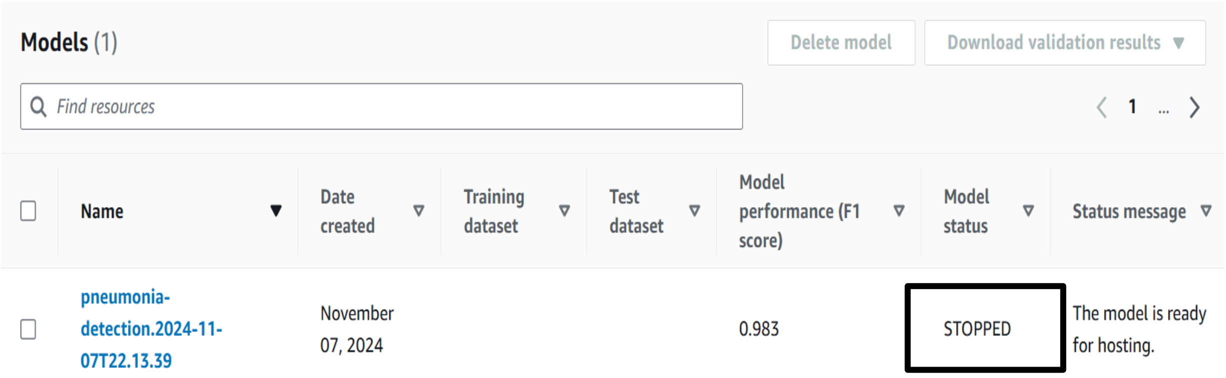
Models (1)
Delete model
Download validation results
Find resources
1 ...
Name
Date created
Training dataset
Test dataset
Model performance (F1 score)
Model status
Status message
pneumonia-detection.2024-11-07T22.13.39
November 07, 2024
0.983
STARTING
The model is starting.

Models (1)
Delete model
Download validation results
Find resources
1 ...
Name
Date created
Training dataset
Test dataset
Model performance (F1 score)
Model status
Status message
pneumonia-detection.2024-11-07T22.13.39
November 07, 2024
0.983
RUNNING
The model is running.

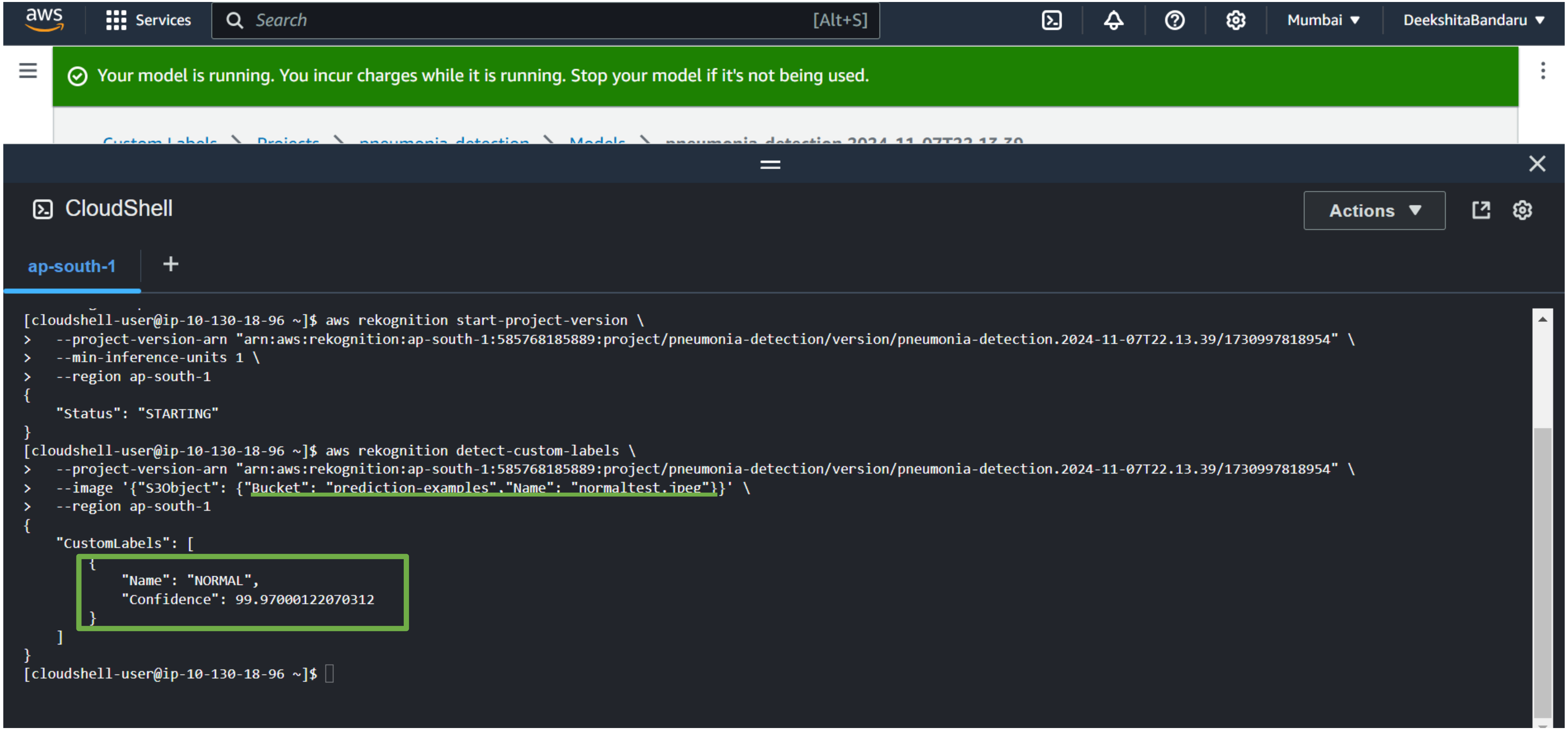



Models (1)
Delete model
Download validation results
Find resources
1 ...
Name
Date created
Training dataset
Test dataset
Model performance (F1 score)
Model status
Status message
pneumonia-detection.2024-11-07T22.13.39
November 07, 2024
0.983
RUNNING
The model is running.

Models (1)
Delete model
Download validation results
Find resources
1 ...
Name
Date created
Training dataset
Test dataset
Model performance (F1 score)
Model status
Status message
pneumonia-detection.2024-11-07T22.13.39
November 07, 2024
0.983
STOPPED
The model is ready for hosting.

# Output Screens



```
[cloudshell-user@ip-10-130-18-96 ~]$ aws rekognition detect-custom-labels \
>   --project-version-arn "arn:aws:rekognition:ap-south-1:585768185889:project/pneumonia-detection/version/pneumonia-detection.2024-11-07T22.13.39/1730997818954" \
>   --image '{"S3Object": {"Bucket": "prediction-examples","Name": "pneumono-test.jpeg"}}' \
>   --region ap-south-1
{
    "CustomLabels": [
        {
            "Name": "PNEUMONIA",
            "Confidence": 100.0
        }
    ]
}
[cloudshell-user@ip-10-130-18-96 ~]$
```

- For prediction purposes, I created another S3 bucket with the name “prediction-examples”, into which I uploaded test images from a different source (unseen data) for the prediction to be made.
- The name of the bucket will be passed to the “Bucket” field in the API code
- The path to the image in the bucket is passed to the “Name” field in the API code
- The output given is the label of the predicted image and the confidence for evaluation measure.

# Conclusion

The developed system for detecting pneumonia using chest X-ray images demonstrated the effectiveness of utilizing cloud-based machine learning tools, specifically Amazon Rekognition's custom labels feature. By training the model with labeled data categorized as "normal" and "pneumonia," the system successfully identified cases of pneumonia with good accuracy. Testing on sample data validated the system's reliability to distinguish between normal and pneumonic images. The project showed that AWS Rekognition, combined with data preprocessing and effective cloud resource management, can serve as a powerful tool for medical image analysis and how cloud can provide on demand delivery of IT services.

Although the current system delivers promising results, there are opportunities for future enhancements. Integrating the model with a back end and front end can be explored to turn this project into a full blown application.

# Thank you