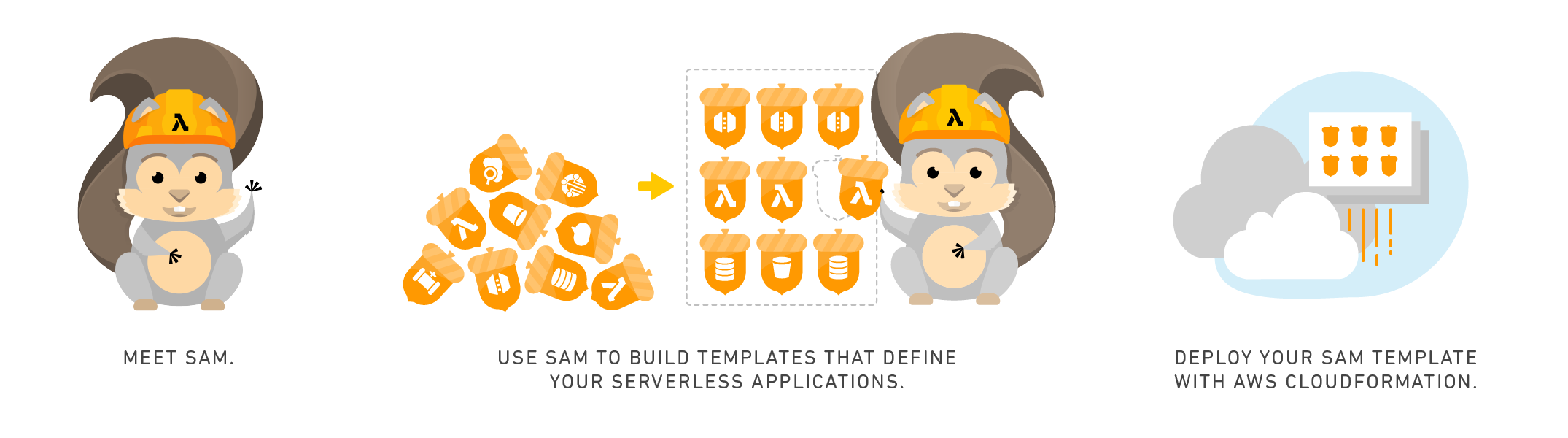
**S3-Image-Monitoring-Detection-Using-Rekognition-Publish-Data-on-SNS**



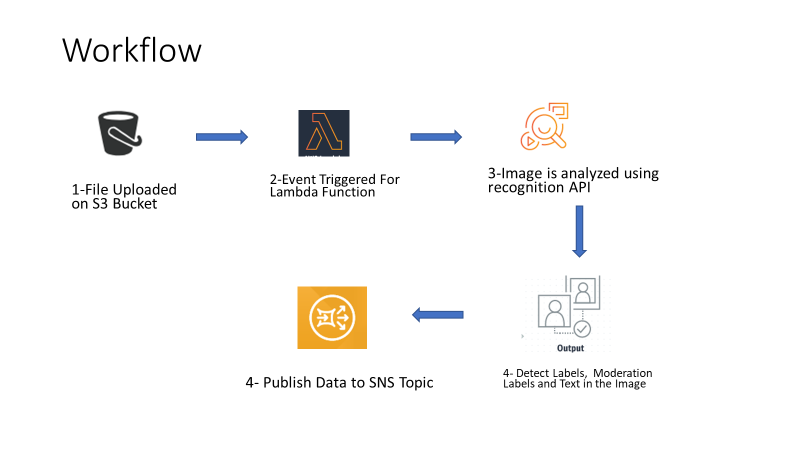
Serverless App for Social Good

AWS Serverless Application Model (SAM)

# S3-Image-Monitoring-Detection-Using-Rekognition-Publish-Data-on-SNS

The Objective of this application is to Detect and Analyze Images uploaded on **S3 Bucket**. When a user uploads any file on S3 Bucket created by application, an event is triggered for **Lambda** function and if the uploaded file is an image then Image labels, Text and Moderation Labels are detected using Amazon **Rekognition API**. Then all the data is published in a message to **SNS Topic**. If any user has subscribed the SNS Topic then user shall be notified using SMS , Email etc. User can review the data and if image content is inappropriate or unsafe then can take necessary action.

# Workflow



# Resources Used

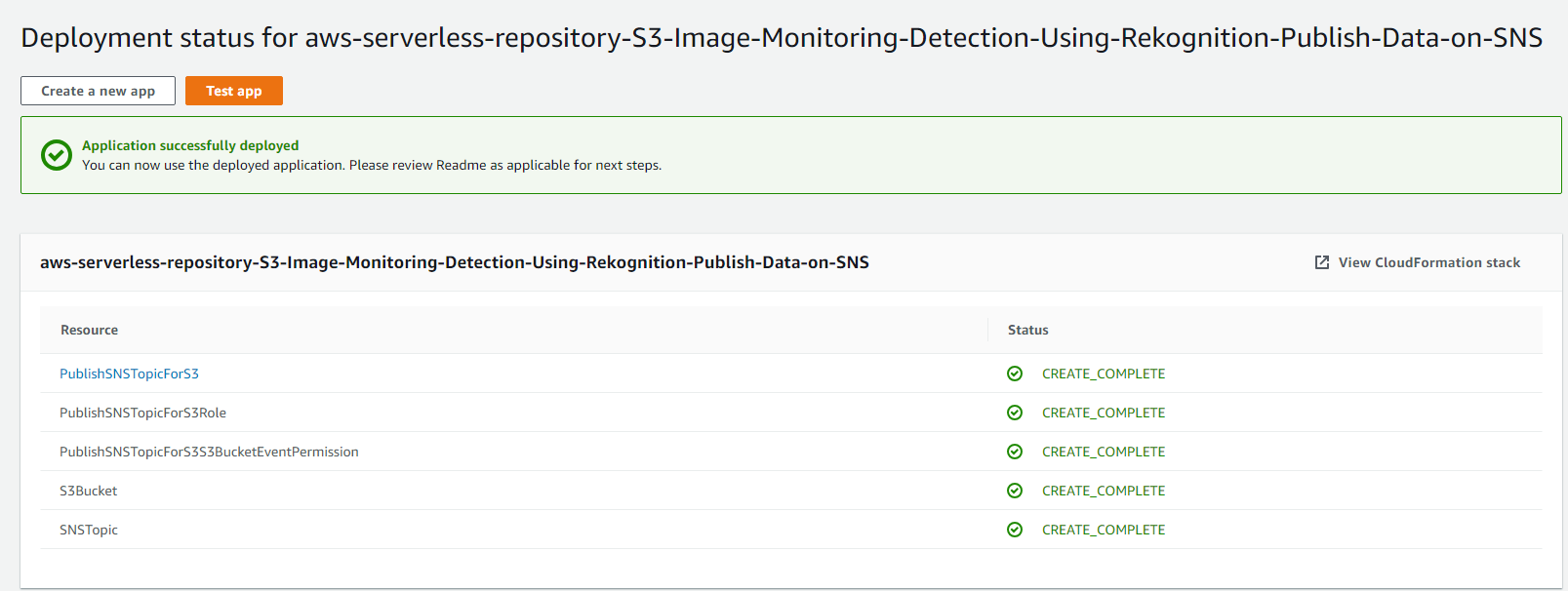
* S3 Bucket
* SNS Topic
* Lambda Function
* Rekognition API

# Application Link

<https://serverlessrepo.aws.amazon.com/#/applications/arn:aws:serverlessrepo:us-east-1:647087826666:applications~S3-Image-Monitoring-Detection-Using-Rekognition-Publish-Data-on-SNS>

# How to Use

* On AWS SAM Repository find application ‘[S3-Image-Monitoring-Detection-Using-Rekognition-Publish-Data-on-SNS’](https://serverlessrepo.aws.amazon.com/)
* Click on “Deploy” Next page appears to review and configure application
* Default Application Name is given, you may change application name if required
* Click on “Deploy”
* On Next step wait until application is fully deployed
* Message is displayed when Application is deployed fully see screenshot



* Deployment of application creates an S3 Bucket which shall be used for Image Uploading. We need to give Read Access of this S3 Bucket to the Lambda Function Role ‘PublishSNSTopicForS3Role’. To do this
* Go to CloudFormation stack by clicking on the button on Top Right corner
* Note Down all the resources given on the CloudFormation Stack e.g. S3 Bucket

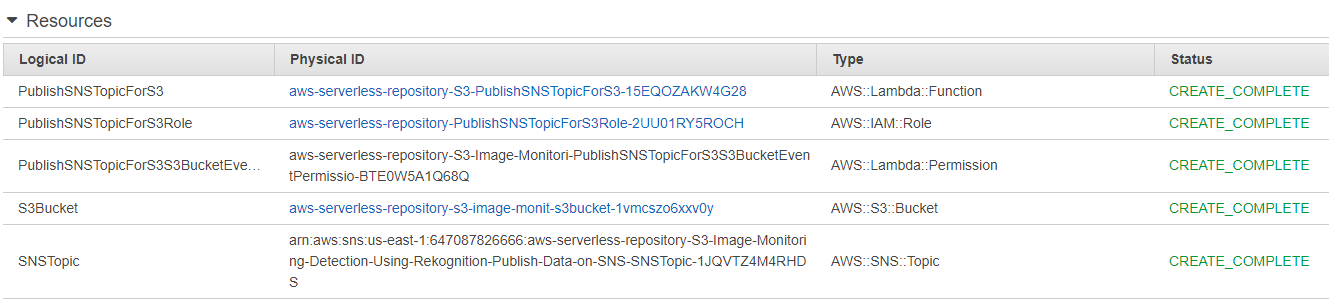
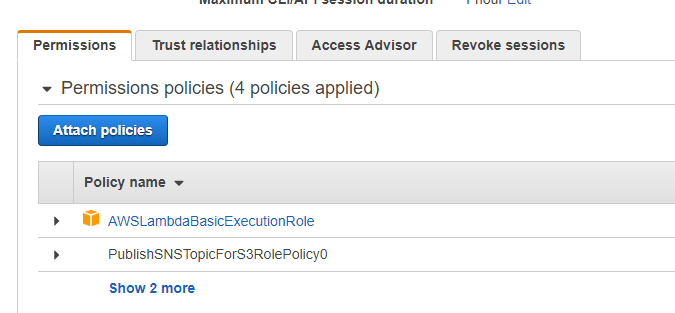
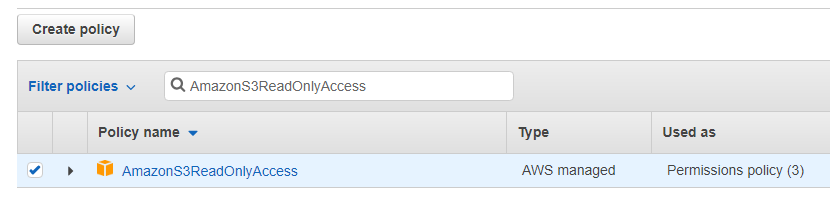


Figure Application Resources

* Click on the link for the PublishSNSTopicForS3Role’ under the column ‘Physical ID’
* On the ‘Role Summery’ Click on “Attach Policies” button



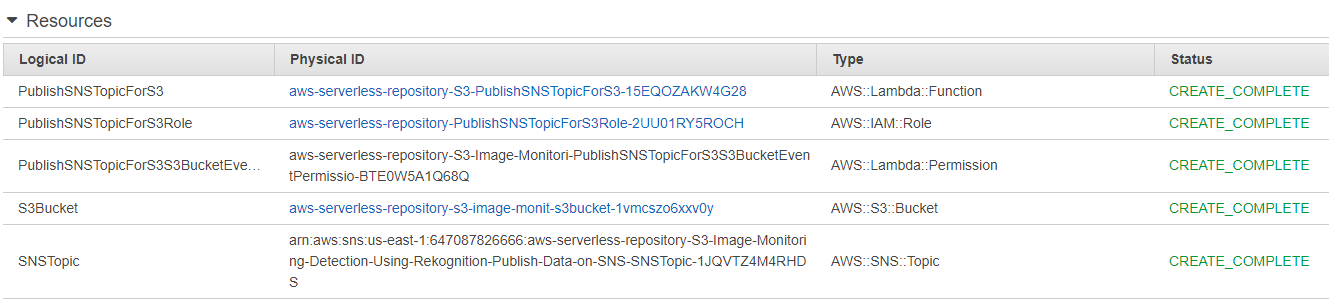
* On the next screen for “Attach Permission” , search for “AmazonS3ReadOnlyAccess” and select the policy



* Then attach selected policy by clicking on the button “Attach Policy” available on Right Bottom corner



* Policy shall be attached to the role and message appears e.g. Policy AmazonS3ReadOnlyAccess has been attached for the aws-serverless-repository 797sddyshshsoo$FFF.
* Now your application is ready for use .
* Subscribe SNS Topic ( ARN of SNS Topic is available on Cloud Formation Stack) see AWS documentation for how to subscribe an SNS Topic <https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html>
* To test your app upload an Image on the S3 Bucket ( Link of bucket is given on Cloud Formation Stack

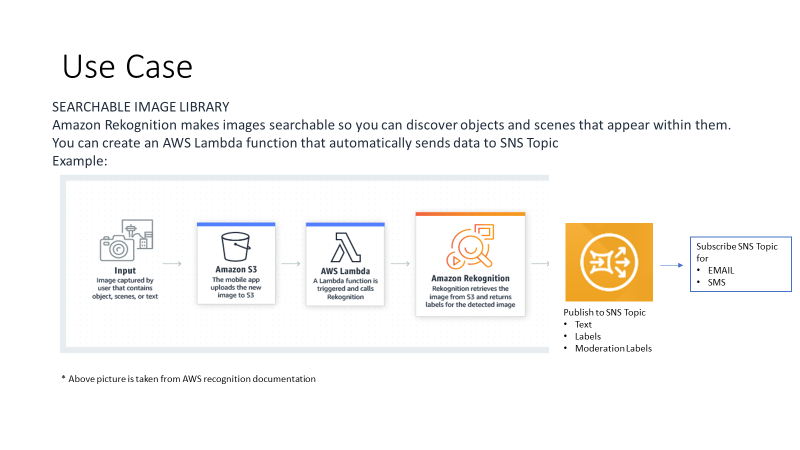


Once image is uploaded then SNS Topic shall receive message which shall be sent to all subscriber.

# GitHub Link For Source Code

<https://github.com/Kitemaker/S3-Image-Monitoring-Detection-Using-Rekognition-Publish-Data-on-SNS>

# USE CASE



# SAM Template:

{

"AWSTemplateFormatVersion": "2010-09-09",

"Transform": "AWS::Serverless-2016-10-31",

"Resources": {

"SNSTopic": {

"Type": "AWS::SNS::Topic",

"Properties" : {

"DisplayName" : "S3\_Image\_Labels\_and\_Text"

}

},

"S3Bucket": {

"Type": "AWS::S3::Bucket"

},

"PublishSNSTopicForS3": {

"Type": "AWS::Serverless::Function",

"Properties": {

"Handler": "SNS\_Image\_Labels\_Rekognition\_S3.lambda\_handler",

"Runtime": "python3.6",

"CodeUri": "s3://sjalexaskillsvirginea/Code/SNS\_Image\_Labels\_Rekognition\_S3.zip",

"Environment": {

"Variables": {

"TOPIC\_ARN": { "Ref" : "SNSTopic" }

}

},

"MemorySize": 128,

"Timeout": 60,

"Policies": [

{

"SNSPublishMessagePolicy": {

"TopicName": { "Fn::GetAtt" : [ "SNSTopic", "TopicName" ] }

}

},

{

"SNSCrudPolicy": {

"TopicName": { "Fn::GetAtt" : [ "SNSTopic", "TopicName" ] }

}

},

{

"RekognitionDetectOnlyPolicy":{}

}

],

"Events": {

"S3BucketEvent": {

"Type": "S3",

"Properties": {

"Bucket": {

"Ref": "S3Bucket"

},

"Events": [

"s3:ObjectCreated:\*"

]

}

}

}

} }

}

}

Enjoy the app !!!!!!!!!!!!!