

# Honorlock SageMaker Canvas and Inference Notes

## S3 Buckets

Create three s3 buckets, One to work with SageMaker Canvas, one for staging and one for tracking storing outputs.

## Sagemaker Canvas

Activate canvas if not already done

### Activating SageMaker Canvas

1. Go to Amazon SageMaker Console, select Canvas,
2. Click the drop down for the Default Execution Role and select Create a new role

3. **Amazon SageMaker** X

Amazon SageMaker > SageMaker Domain

### Setup SageMaker Domain

Use SageMaker Domain as the central store to manage the configuration of SageMaker for your organization.

**Quick setup**  
Let Amazon SageMaker configure your account, and set up permissions for your SageMaker Domain.

- ✓ Public internet access, and standard encryption
- ✓ SageMaker Studio integration
- ✓ Sharable SageMaker Studio Notebooks
- ✓ IAM Authentication

**Standard setup**  
Control all aspects of account

- ✓ Advanced network security
- ✓ SageMaker Studio, and RStudio
- ✓ SageMaker Studio Project
- ✓ IAM, or SSO authentication

#### User profile

Name

default-1643131720075

The name can have up to 63 characters. Valid characters: A-Z, a-z, 0-9, and - (hyphen)

**Default execution role**

SageMaker Domain requires permissions for its users to access other AWS services, such as Amazon SageMaker and Amazon S3. The execution role must have the [AmazonSageMakerFullAccess](#) policy attached. If you don't have a role with this policy attached, we can create one for you.

TeamRole

Create a new role

Enter a custom IAM role ARN

Use existing role

TeamRole

Cancel Submit

4. Ensure that the Any S3 Bucket radio button is checked and click the Create role button

5.

### 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**

Allows users that have access to your notebook instance access to any bucket and its contents in your account.

☐ **Specific S3 buckets**

Comma delimited. ARNs, "\*" and "/" are not supported.

☐ **None**

☒ Any S3 bucket with "sagemaker" in the name

☒ Any S3 object with "sagemaker" in the name

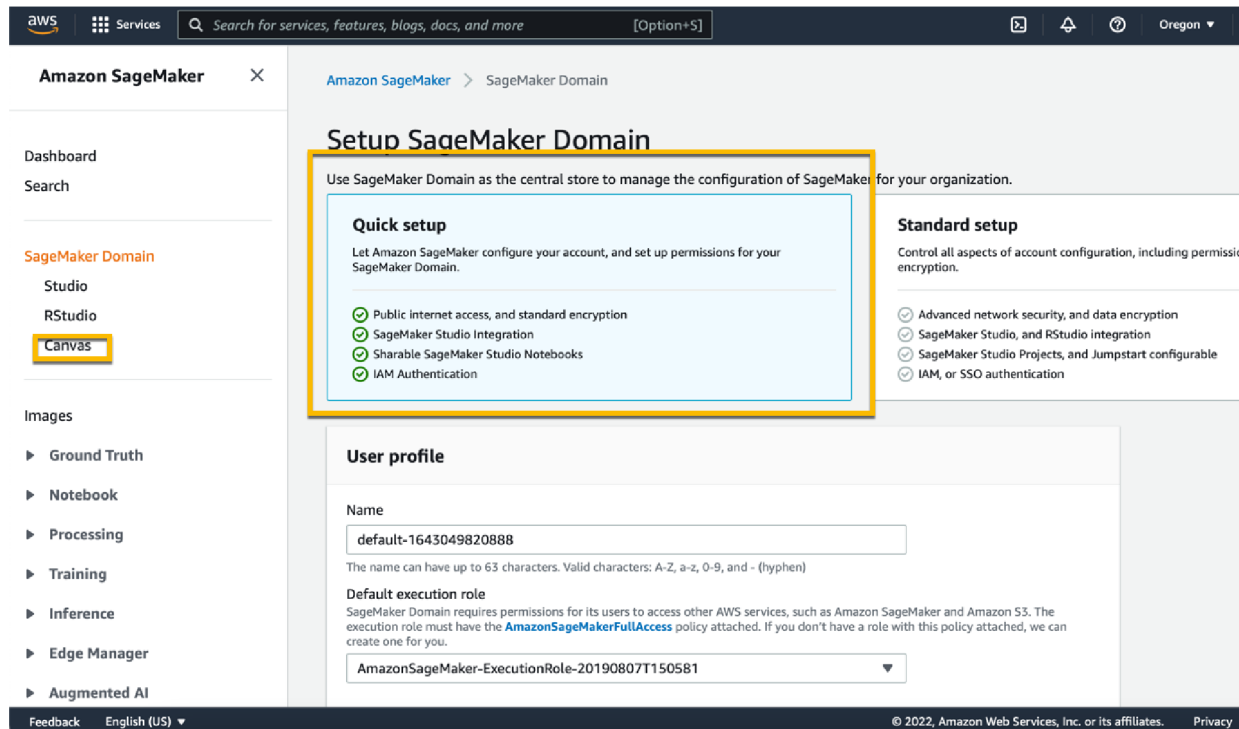
☒ Any S3 object with the tag "sagemaker" and value "true" [See Object tagging](#)

☒ S3 bucket with a Bucket Policy allowing access to SageMaker [See S3 bucket policies](#)

Cancel

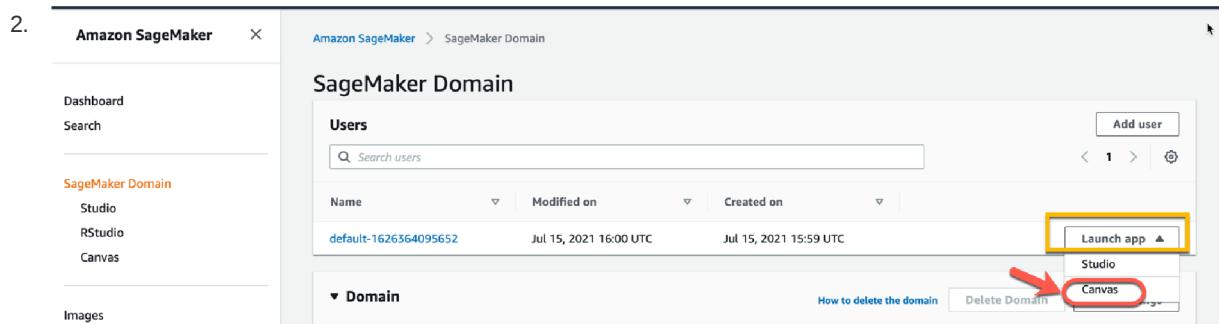
Create role

6. Follow Quick setup prompts or Standard setup if you so choose.



## Starting Canvas

1. Once set up is complete, you can launch Canvas from the SageMaker Console as shown below

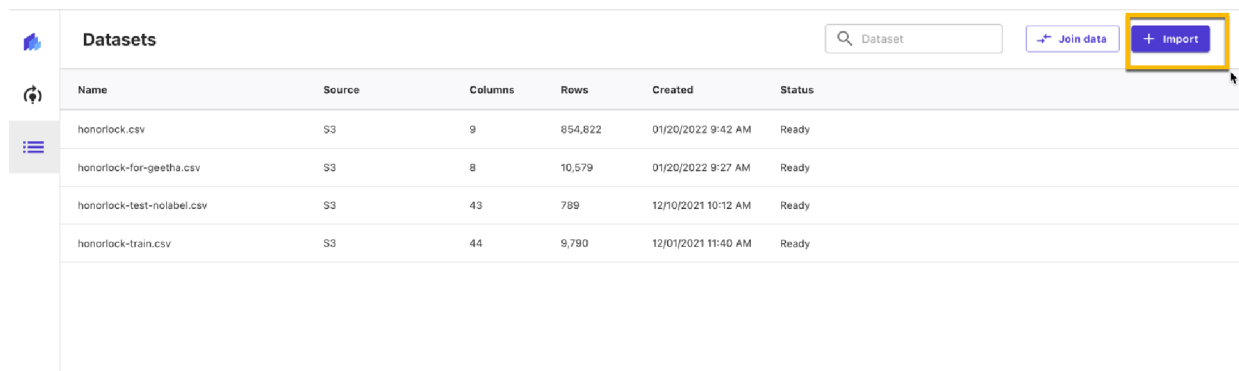


## Get started with ML on Canvas

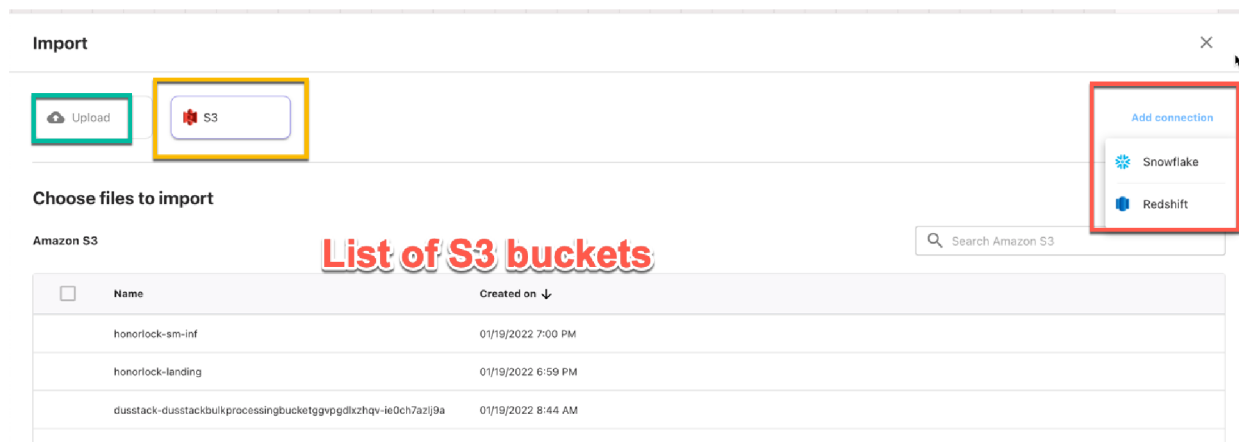
The process of ML with Canvas uses intuitive upload a dataset, examine and decide the columns and target, choose a ML model and then build and train a model followed by inference.

All steps can be carried out in Canvas, However this set up follows training models in Canvas and automating the inference using lambda functions.

1. Canvas allows you to upload datasets from S3 or local, It is recommended that you use S3 for staging your dataset



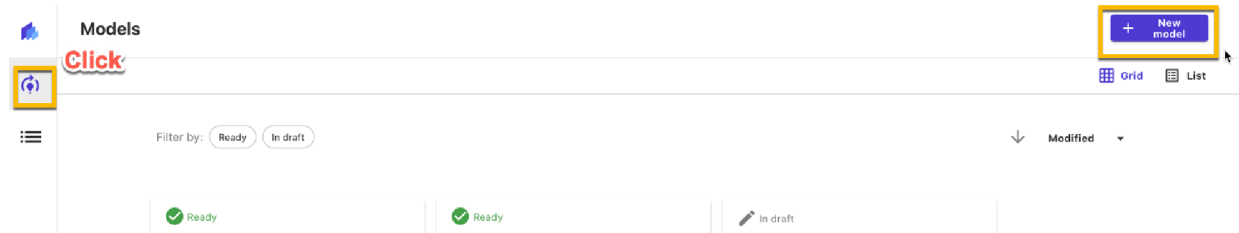
Name	Source	Columns	Rows	Created	Status
honorlock.csv	S3	9	854,822	01/20/2022 9:42 AM	Ready
honorlock-for-geetha.csv	S3	8	10,579	01/20/2022 9:27 AM	Ready
honorlock-test-nolabel.csv	S3	43	789	12/10/2021 10:12 AM	Ready
honorlock-train.csv	S3	44	9,790	12/01/2021 11:40 AM	Ready



**List of S3 buckets**

Name	Created on ↓
honorlock-sm-inf	01/19/2022 7:00 PM
honorlock-landing	01/19/2022 6:59 PM
dusstack-dusstackbulkprocessingbucketggvpgdkzhqv-le0ch7azj9a	01/19/2022 8:44 AM

2. Multiple datasets can be joined directly inside canvas, for example holiday dataset can be joined with your dataset on dates
3. the next step is examining the data and looking for any inconsistencies. we do this in model building stage b



Filter by: Ready In draft

Modified

4. click on New Model and create a new model, name it and click create.follow the prompts and select the dataset you want to use
5. The next screen is where you can examine the dataset, select a target, type of model and start a build

**testmodel10** V1 Draft Add version Share

Select **Build** Analyze Predict

Select a column to predict **Select a target**

Choose the target column. The model that you build predicts values for the column that you select.

Target column: test\_count

Model type: Standard build

Time series forecasting: Your model will forecast test\_count by using past data values to predict future data values. Change type Configure

Standard build: Preview model

honorlock.csv Sample

Column name	Data type	Missing	Mismatched	Unique	Mean / Mode
weekday	Numeric	0.00% (0)	0.00% (0)	7	1
top_school	Binary	0.00% (0)	0.00% (0)	2	0
test_count	Numeric	0.00% (0)	0.00% (0)	288	1
school_id	Numeric	0.00% (0)	0.00% (0)	318	125

**Model type**

☐ Time series forecasting

A time series forecasting model uses past data values to predict future data values.

Example business questions

- How will my sales be affected if I raise prices 10%?
- How much inventory should I order for the holiday season?

☒ **Numeric model type**

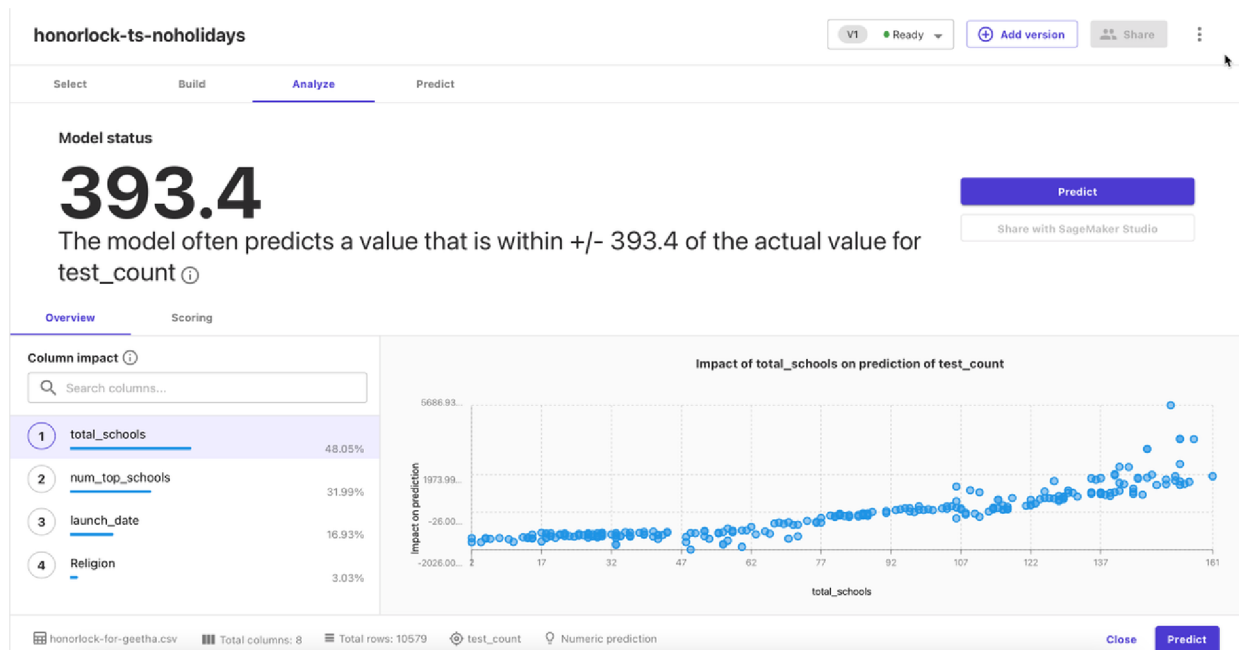
Estimate the value of the target column using the values of the other columns in the dataset.

Example business questions

- How many days will it take for a package to be delivered?
- How many days until a customer is likely to purchase again?

Cancel **Change type**

6. Once model type is selected, click on standard build and let the model build complete. You can see the model analysis such as influencing columns and their contribution allowing you to explain the model.



7. At this point we exit canvas and go to SageMaker Console. go to Training and Hyperparameter Tuning Jobs and find the Canvas job that just ran.

## Inference With SageMaker Batch.

We will create 2 lambda functions, a SNS topic and provision IAM permissions to lambda functions.

### Lambda Functions

#### Lambda Function for starting SageMaker Batch Inference

1. go to lambda console on aws console and create a new lambda function, select author your own and for language select python3.8

**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.

**Container image** ☐  
Select a container image to deploy for your function.

**Basic information**

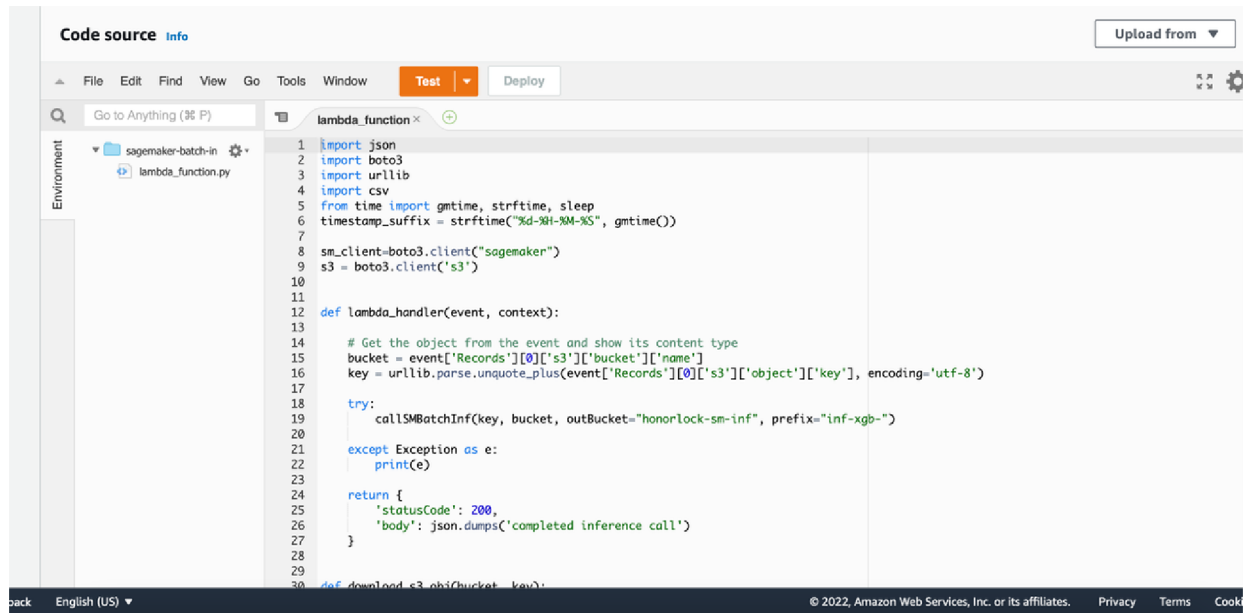
**Function name**  
Enter a name that describes the purpose of your function.  
  
Use only letters, numbers, hyphens, or underscores with no spaces.

**Runtime** [Info](#)  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  
☒ x86\_64  
☐ arm64

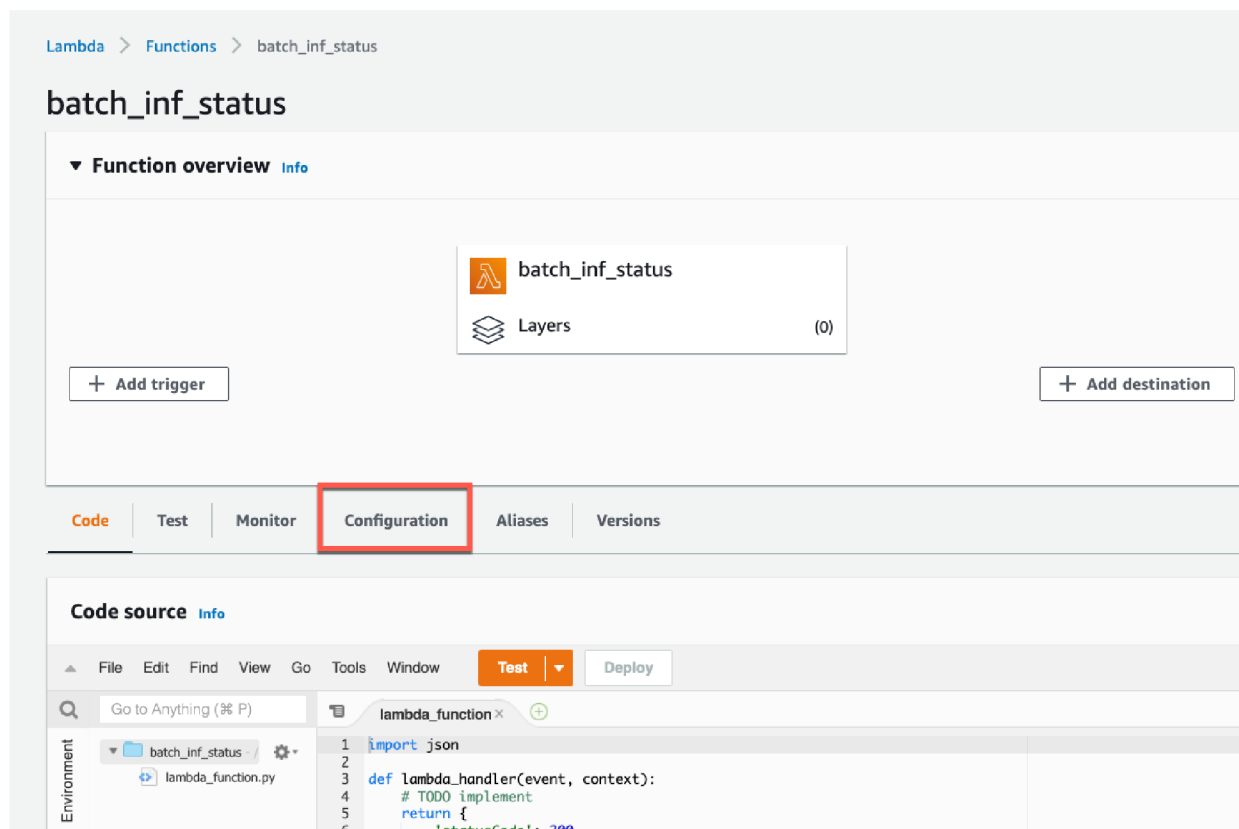
back English (US) © 2022, Amazon Web Services, Inc.

2. Once the function is created, go to code section of the lambda function and paste in the code sm-batch-inf.py
3. click File-->save.
4. Look at the code and change any setting , like naming conventions, logic to use the latest model as you see fit, One way to decouple that from code would be to have a config.json file stored in S3 that can be updated without touching the code. The config file can contain output file naming, model to use for inference or any other variables.



```
1 import json
2 import boto3
3 import urllib
4 import csv
5 from time import gmtime, strftime, sleep
6 timestamp_suffix = strftime("%d-%M-%Y", gmtime())
7
8 sm_client=boto3.client("sagemaker")
9 s3 = boto3.client('s3')
10
11
12 def lambda_handler(event, context):
13
14     # Get the object from the event and show its content type
15     bucket = event['Records'][0]['s3']['bucket']['name']
16     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
17
18     try:
19         callSMBatchInf(key, bucket, outBucket="honorlock-sm-inf", prefix="inf-xgb-")
20     except Exception as e:
21         print(e)
22
23     return {
24         'statusCode': 200,
25         'body': json.dumps('completed inference call')
26     }
27
28
29 def download_s3_obj(bucket, key):
```

- Next add a trigger to Lambda function, Choose one of the S3 buckets assigned as landing bucket for inference input.
- Next we will update permissions so lambda can access Sagemaker and S3.







Lambda > Functions > batch\_inf\_status

## batch\_inf\_status

▼ Function overview [Info](#)

 batch\_inf\_status

 Layers (0)

[+ Add trigger](#)

[-](#)


Code | Test | Monitor | **Configuration** | Aliases | Versions

**General configuration**  
Triggers  
**Permissions**  
Destinations  
Environment variables  
Tags

**General configuration** [Info](#)

Description -


Memory 128 MB


 **AWS Compute Optimizer**  
Opt in to see memory recommendations for your Lambda functions. [View details](#)

Lambda > Functions > batch\_inf\_status

## batch\_inf\_status

▼ Function overview [Info](#)

 batch\_inf\_status

 Layers (0)

[+ Add trigger](#)

[+ Add destination](#)

[-](#)


Code | Test | Monitor | **Configuration** | Aliases | Versions

General configuration  
Triggers  
**Permissions**  
Destinations  
Environment variables  
Tags  
VPC

**Execution role**

Role name  
[batch\\_inf\\_status-role-q3d2gl7v](#)

**Resource summary**

 **Amazon CloudWatch Logs**  
3 actions, 2 resources

**Identity and Access Management (IAM)**

**Dashboard**

- Access management
  - User groups
  - Users
  - Roles**
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

AWS account ID: 887240564678

**Summary**

**Role ARN** arn:aws:iam::887240564678:role/service-role/batch\_inf\_status-

**Role description** [Edit](#)

**Instance Profile ARNs**

**Path** /service-role/

**Creation time** 2022-01-24 09:55 MST

**Last activity** Not accessed in the tracking period

**Maximum session duration** 1 hour [Edit](#)

**Permissions** | Trust relationships | Tags | Access Advisor | Revoke sessions

▼ Permissions policies (1 policy applied)

**Attach policies**

Policy name ▼

▶ AWSLambdaBasicExecutionRole-72415c02-26a4-45d4-bcb2-7e435995dad2

▶ Permissions boundary (not set)

▼ Generate policy based on CloudTrail events

## Add permissions to batch\_inf\_status-role-q3d2gl7v

### Attach Permissions

**Create policy**


**Filter policies** ▼


Policy name ▼












- ☐ ▶ AmazonDMSRedshiftS3Role
- ☒ ▶ AmazonS3FullAccess
- ☐ ▶ AmazonS3ObjectLambdaExecutionRolePolicy
- ☐ ▶ AmazonS3OutpostsFullAccess
- ☐ ▶ AmazonS3OutpostsReadOnlyAccess
- ☐ ▶ AmazonS3ReadOnlyAccess
- ☐ ▶ QuickSightAccessForS3StorageManagementAnalyticsReadOnly


Add permissions to batch\_inf\_status-role-q3d2gl7v


Attach Permissions

Create policy 

Filter policies   Showing 11 results

	Policy name	Type	Used as
<input type="checkbox"/>	 AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	AWS managed	None
<input type="checkbox"/>	 AmazonSageMakerEdgeDeviceFleetPolicy	AWS managed	None
<input type="checkbox"/>	 AmazonSageMakerFeatureStoreAccess	AWS managed	None
<input checked="" type="checkbox"/>	 AmazonSageMakerFullAccess	AWS managed	None
<input type="checkbox"/>	 AmazonSageMakerGroundTruthExecution	AWS managed	None
<input type="checkbox"/>	 AmazonSageMakerMechanicalTurkAccess	AWS managed	None
<input type="checkbox"/>	 AmazonSageMakerPipelinesIntegrations	AWS managed	None
<input type="checkbox"/>	 AmazonSageMakerReadOnly	AWS managed	None
<input type="checkbox"/>	 AWSGlueConsoleSageMakerNotebookFullAccess	AWS managed	None
<input type="checkbox"/>	 AWSPanoramaSageMakerRolePolicy	AWS managed	None
<input type="checkbox"/>	 AWSQuickSightSageMakerPolicy	AWS managed	None



Cancel 

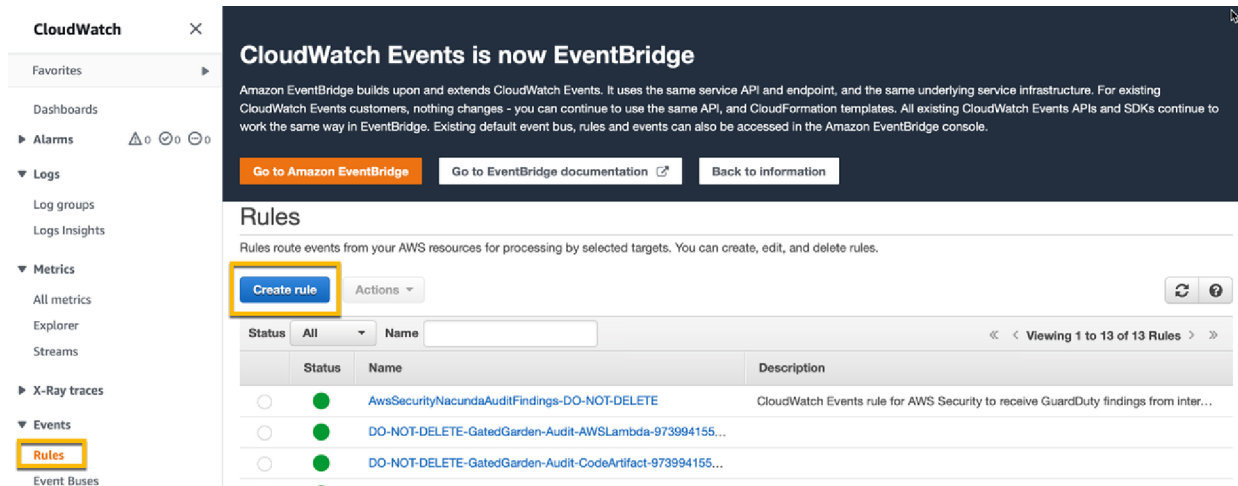
The Lambda function is now ready to react to a file drop in S, look at the file for headers, remove if there is any and call SageMaker Batch Inference with the file. We can do a few more things to the Lambda function to make it efficient.

In configurations, we can remove multiple retries in asynchronous section, we can also increase the memory to 1GB and change the timeout to 15min in General configuration sections.

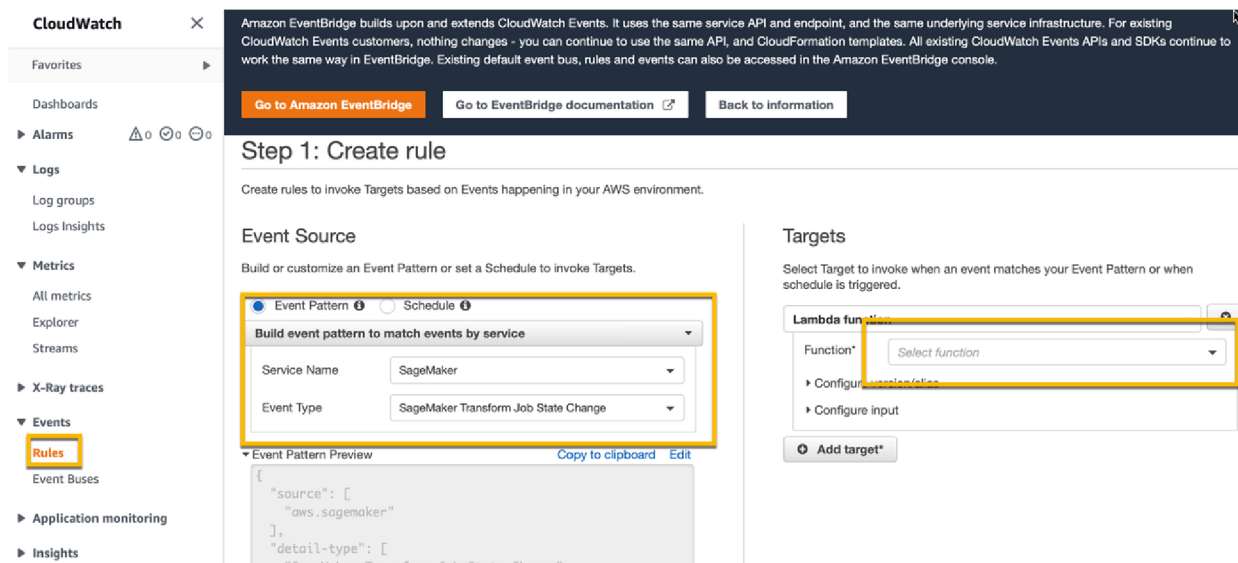
### Lambda Function for Checking completion status and re-assemble the output csv

This lambda function is created using the code from batch-ing-status.py code. The trigger for this is cloudwatch events.

1. Let's start with lambda function as above and add an addition permission for SNS to this lambda function.
2. For the Trigger, Let's create a cloudwatch event and associate with this lambda function.



3. Create a rule and select SageMaker as service and Transform Job State change as event type, on the right hand side select the second lambda function created above as your target and follow the rest of the prompts.



This lambda function will kick off after the first lambda function that initiated the inference. As a part of this, we have also provided code to create a SNS topic and subscribe to updates on Inference results. Whoever is on subscription list will get an email asking to confirm subscription and once confirmed they will get an email when the inference is successfully completed. In your case, your operations people will get an email saying new predictions are available.

## QuickSight and Q

We will configure the quicksight folder in your output bucket to sync with Amazon Quicksight.

Here, we will work with Quicksight setup I have walked you through in the past and we will add the new s3 location to the manifest.json file to be included in the dataset access.