

Amazon SageMaker and Teradata Vantage

As many Teradata customers have interest in integrating Teradata Vantage with AWS First Party Services, this Getting Started Guide document will help you to integrate Amazon SageMaker with Teradata Vantage.

The approach this guide explains is one of many potential approaches to integrate with the service and is offered on an as-is basis. Although the approach has been implemented and tested internally, there is no formal support from either Teradata or AWS on the approach.

That said, your feedback is very desired and appreciated – what worked, what didn’t, how can this be improved, or etc. Please send your feedback to [wenjie.tehan@teradata.com](mailto:wenjie.tehan@teradata.com).

# Overview

Amazon SageMaker provides a fully managed Machine Learning Platform. There are two use cases for Amazon SageMaker and Teradata:

1. Data resides on Teradata Vantage and Amazon SageMaker will be used for both the Model definition and subsequent scoring. Under this use case Teradata will provide data into the Amazon S3 environment so that Amazon SageMaker can consume training and test data sets for the purpose of model development. Teradata would further make data available via Amazon S3 for subsequent scoring by Amazon SageMaker. Under this model Teradata is a data repository only.
2. Data resides on Teradata Vantage and Amazon SageMaker will be used for the Model definition, and Teradata for the subsequent scoring. Under this use case Teradata will provide data into the Amazon S3 environment so that Amazon SageMaker can consume training and test data sets for the purpose of model development. Teradata will need to import the Amazon SageMaker model into a Teradata table for subsequent scoring by Teradata Vantage. Under this model Teradata is a data repository and a scoring engine.

The first use case is discussed in this document.

Amazon SageMaker consumes training and test data from an Amazon S3 bucket. This article describes how you can load Teradata analytics data sets into an Amazon S3 bucket. The data can then available to Amazon SageMaker to build and train machine learning models and deploy them into a production environment.

Prerequisites:

* IAM permission to access Amazon S3 bucket, and to use Amazon SageMaker service
* An Amazon S3 bucket to store training data

This article leverages the existing Amazon SageMaker online example to build, train and deploy a machine learning model using data from a Vantage system. The example uses [Bank Marketing Data Set](https://archive.ics.uci.edu/ml/datasets/bank+marketing) to predict whether a customer will enroll for a CD account. The python code and instructions for the example is located [here](https://aws.amazon.com/getting-started/tutorials/build-train-deploy-machine-learning-model-sagemaker/).

### About Teradata Vantage

Vantage is the modern cloud platform that unifies data warehouses, data lakes, and analytics into a single connected ecosystem.

Vantage combines descriptive, predictive, prescriptive analytics, autonomous decision-making, ML functions, and visualization tools into a unified, integrated platform that uncovers real-time business intelligence at scale, no matter where the data resides.

Vantage enables companies to start small and elastically scale compute or storage, paying only for what they use, harnessing low-cost object stores and integrating their analytic workloads.

Vantage supports R, Python, Teradata Studio, and any other SQL-based tools. You can deploy Vantage across public clouds, on-premises, on optimized or commodity infrastructure, or as-a-service.

# Getting Started

## Load Data

Amazon SageMaker trains data from an Amazon S3 bucket. Following are the steps to load training data from Vantage to an Amazon S3 bucket:

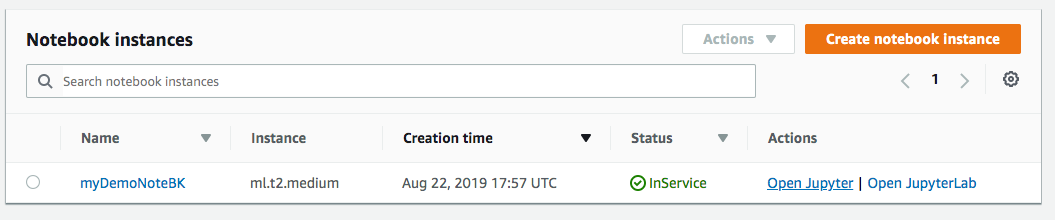
* Go to Amazon SageMaker console and create a Notebook Instance.

A screenshot of a cell phone

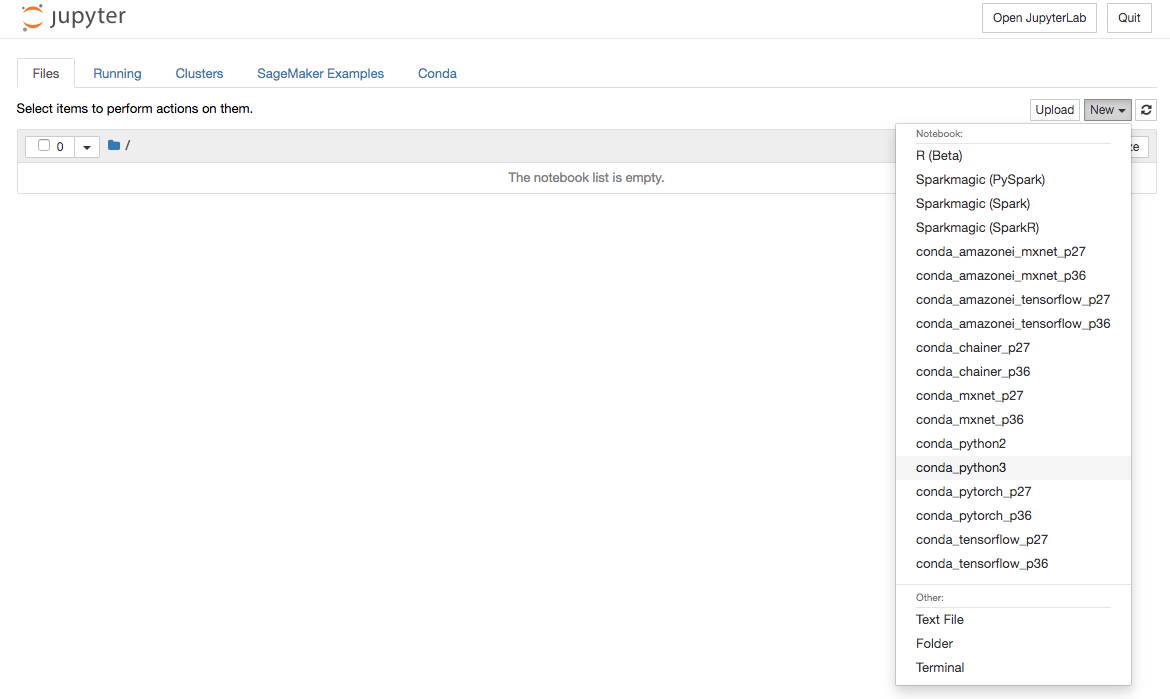
Description automatically generated

Instructions on how to create a Notebook Instance can be found [here](https://docs.aws.amazon.com/sagemaker/latest/dg/gs-setup-working-env.html).

* Open your notebook instance



* Start a new file by clicking on New -> conda\_python3



* Install Teradata Python library using “!pip install teradataml”

A picture containing newspaper, text

Description automatically generated

* Import following libraries

import teradataml as tdml

from teradataml import create\_context, get\_context, remove\_context

from teradataml.dataframe.dataframe import DataFrame

import pandas as pd

import boto3, os

A screenshot of a cell phone

Description automatically generated

* Connect to Teradata using command: create\_context(host = ‘<hostname>’, username = ‘<database user name>’, password = '<database password>')

A close up of a device

Description automatically generated

* Retrieve Data using Teradata Python DataFrame module from the table where the training dataset resides using tdml.DataFrame(‘<table name>’). i.e. tmdl.DataFrame(‘train’)



* Convert Teradata DataFrame to Panda DataFrame

A close up of a logo

Description automatically generated

* Write data to a local file

A close up of a logo

Description automatically generated

* Assign Variables for Amazon S3 bucket

A screenshot of a social media post

Description automatically generated

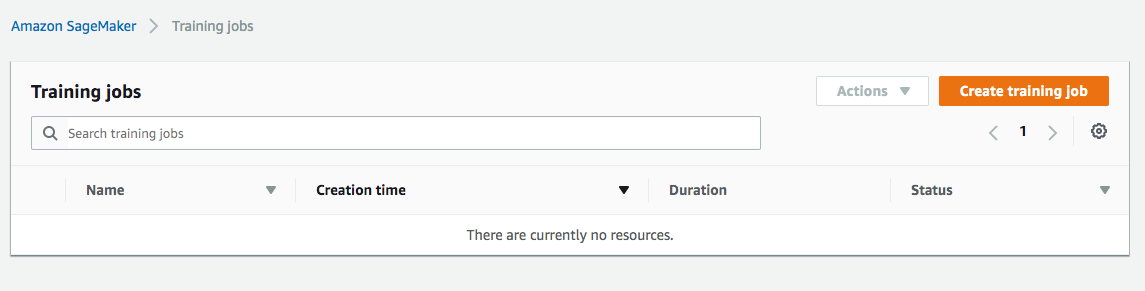
* Upload file to Amazon S3:



## Train the Model

### Create a training job

* Select “Training jobs” on the left menu under “Training”, then click on “Create training job”



* At the “Create training job” window, fill in the “Job name” (i.e. xgboost-bank) and “Create a new role” for the IAM role. Choose “Any S3 bucket” for the Amazon S3 buckets and “Create role”

A screenshot of a social media post

Description automatically generated

* Back at the “Create training job” window, use “XGBoost” as the algorithm

A screenshot of a social media post

Description automatically generated

* Use default ml.m4.xlarge instance type, and 30GB of additional storage volume per instance. This is a short training job, shouldn’t take more than 10 minutes.

A screenshot of a cell phone

Description automatically generated

* Fill in following hyperparameters and leave everything else as default:

num\_round=100

silent=0

eta=0.2

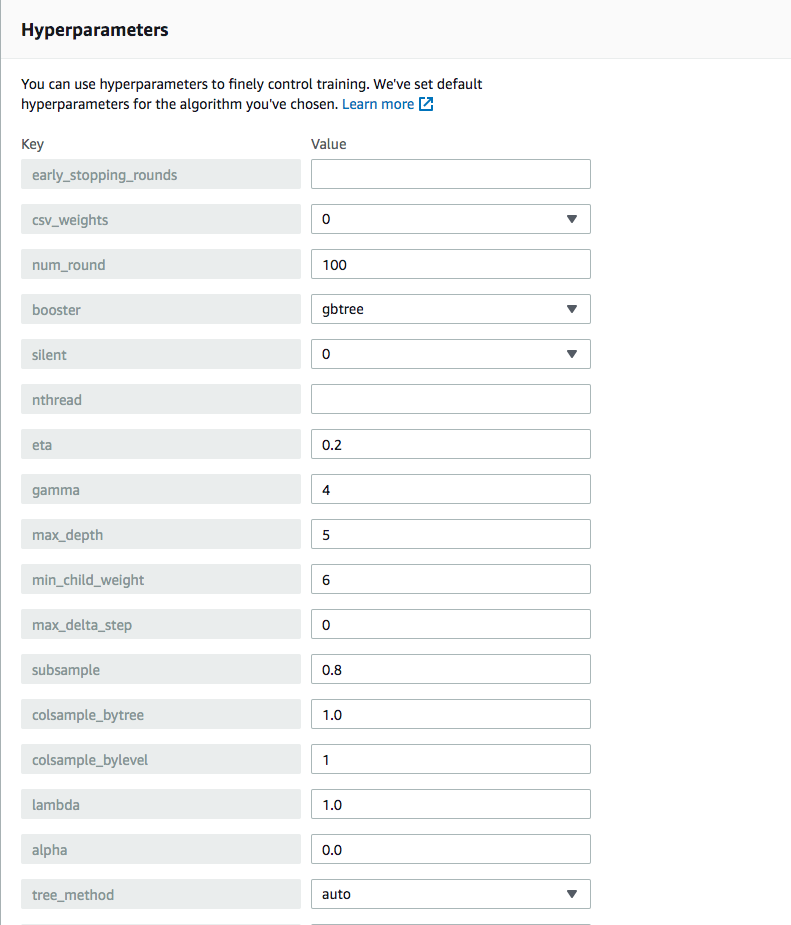
gamma=4

max\_depth=5

min\_child\_weight=6

subsample=0.8

objective='binary:logistic'



A screenshot of a computer

Description automatically generated

For “Input data configuration”, enter the Amazon S3 bucket where you stored your training data. Input mode is “File”. Content type is “csv”. “S3 location” is where the file uploaded to.

A screenshot of a cell phone

Description automatically generated

* For “Output data configuration”, enter path where the output data will be stored

A screenshot of a cell phone

Description automatically generated

Leave everything else as default, and click on “Create training job”. Detail instructions on how to configure the training job can be found on [AWS site here](https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-mkt-algo-train.html#sagemaker-mkt-algo-train-console)**.**

Once the training job’s created, Amazon SageMaker launches the ML instances to train the model, and stores the resulting model artifacts and other output in the “Output data configuration” path/<training job name>/output by default.

## Deploy the Model

After you train your model, deploy it using a persistent endpoint

### Create a Model

* Select “Models” under “Inference” from the left panel, then “Create model”. Fill in Model name (i.e. xgboost-bank), and choose the IAM role you created from previous step
* For Container definition 1, use “433757028032.dkr.ecr.us-west-2.amazonaws.com/xgboost:latest” as “Location of inference code image”. “Location of model artifacts” is the output path of your training job

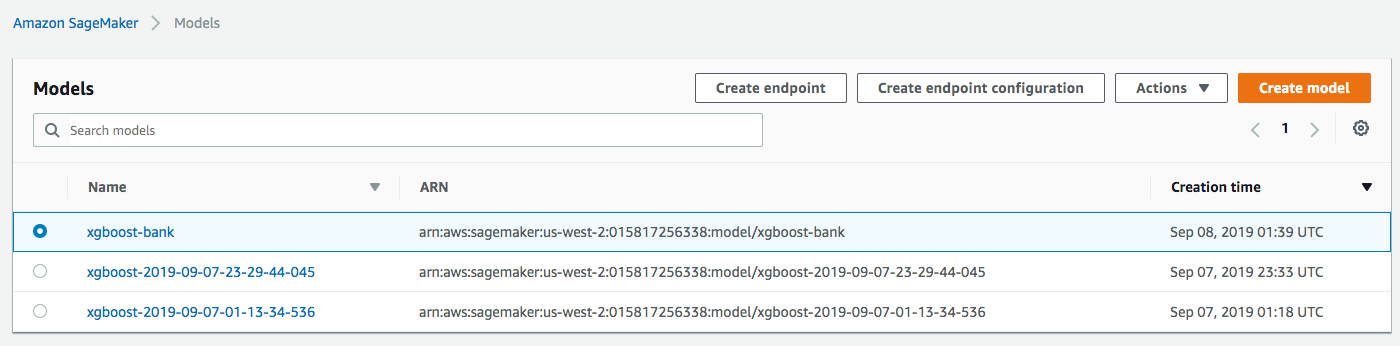
A screenshot of a cell phone

Description automatically generated

Leave everything else as default, then “Create model”.

### Create and endpoint configuration

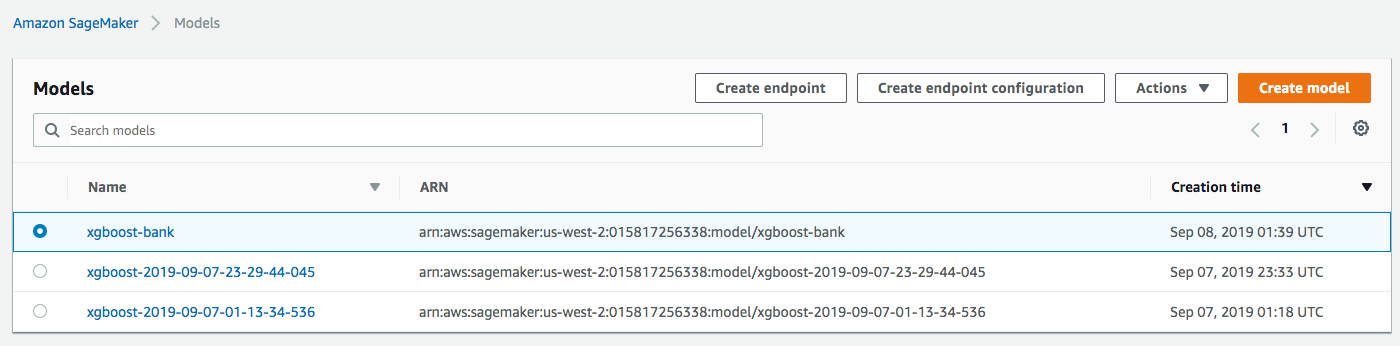
* Select the model you just created, then click on “Create endpoint configuration”



* Fill in the name (i.e. xgboost-bank) and use default for everything else. The Model name and Training Job should be automatically entered out for you. Click on “Create endpoint configuration”

### Create endpoint

* Select “Inference” -> “Models” from the left panel, select the model again, and click on “Create endpoint” this time



* Fill in the name (i.e. xgboost-bank), and “Use an existing endpoint configuration”

A screenshot of a cell phone

Description automatically generated

* Select the endpoint configuration created from last step, and click on “Select endpoint configuration”

A screenshot of a cell phone

Description automatically generated

Leave everything else as default and click on “Create endpoint”.

Now the model is deployed to the endpoint and can be used by client applications.

17095 Via Del Campo, San Diego, CA 92127   [Teradata.com](http://www.teradata.com)

Teradata and the Teradata logo are registered trademarks of Teradata Corporation and/or its affiliates in the U.S. and worldwide. Teradata continually improves products as new technologies and components become available. Teradata, therefore, reserves the right to change specifications without prior notice. All features, functions, and operations described herein may not be marketed in all parts of the world. Consult your Teradata representative or Teradata.com for more information.

Copyright © 2021 by Teradata Corporation    All Rights Reserved.    Produced in U.S.A.

7.21 EBxxxx