Machine Learning Demo / Tutorial

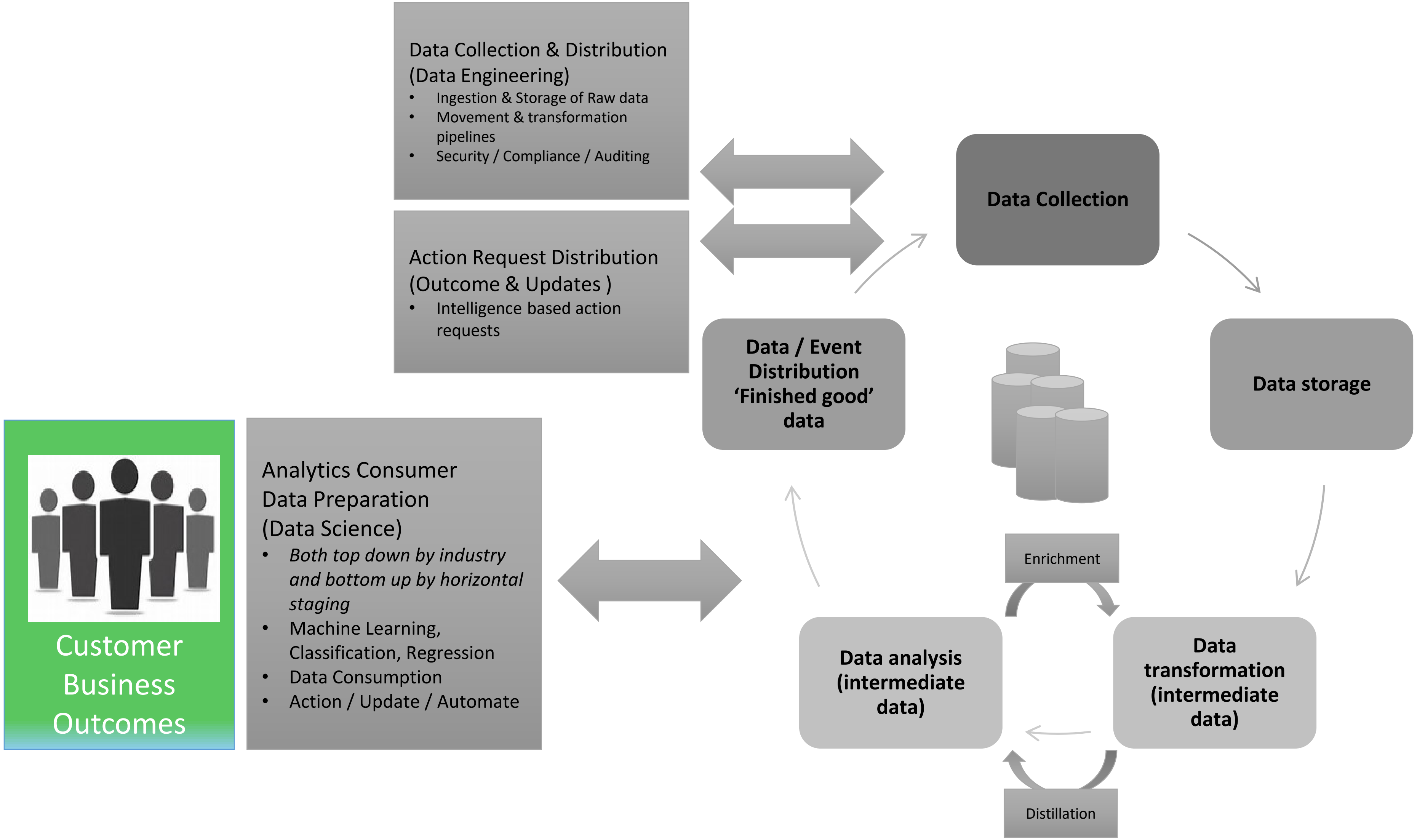
Big Data Management
Big Data Analytics

Ken Cottrell
Cottrell.ken@gmail.com
(214) 546-100

agenda

- Problem Domain / Intro to the data set (in this case Public Healthcare Data)
- Demo & Tools discussion
- Summary, Lessons Learned

Goal of this demo / Tutorial: Describe a phased Approach to a Data-Driven Architecture



Data Collection & Distribution (Data Engineering)

- Ingestion & Storage of Raw data
- Movement & transformation pipelines
- Security / Compliance / Auditing

Action Request Distribution (Outcome & Updates)

- Intelligence based action requests

Customer Business Outcomes

Analytics Consumer
Data Preparation
(Data Science)

- *Both top down by industry and bottom up by horizontal staging*
- Machine Learning, Classification, Regression
- Data Consumption
- Action / Update / Automate

Data Collection

Data storage

Data transformation (intermediate data)

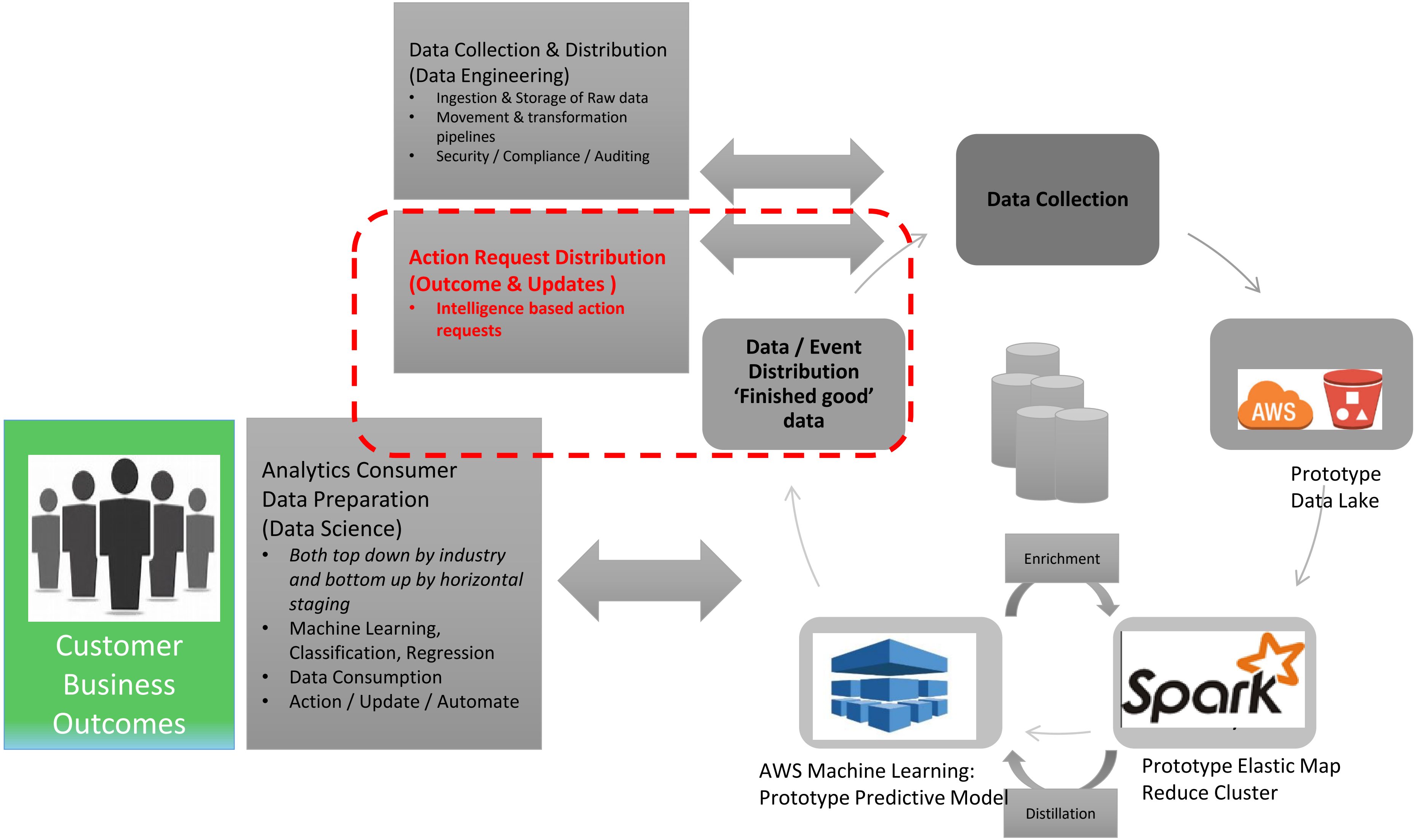
Data analysis (intermediate data)

Data / Event Distribution 'Finished good' data

Enrichment

Distillation

Goal of this demo / Tutorial: Describe a phased Approach to a Data-Driven Architecture



Data Collection & Distribution (Data Engineering)

- Ingestion & Storage of Raw data
- Movement & transformation pipelines
- Security / Compliance / Auditing

Action Request Distribution (Outcome & Updates)

- Intelligence based action requests

Customer Business Outcomes

Analytics Consumer Data Preparation (Data Science)

- *Both top down by industry and bottom up by horizontal staging*
- Machine Learning, Classification, Regression
- Data Consumption
- Action / Update / Automate

Data Collection

Data / Event Distribution 'Finished good' data

Prototype Data Lake

Enrichment

Distillation

AWS Machine Learning: Prototype Predictive Model

Prototype Elastic Map Reduce Cluster

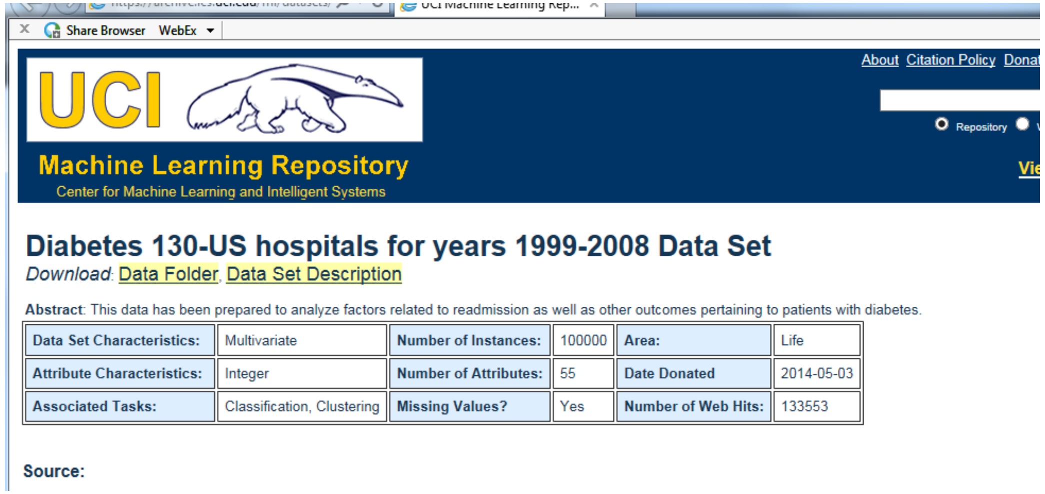
Problem domain (Healthcare)

Functional: *Providers who increasingly need to move to a Value-based care model, away from per-encounter payment model.*

- Providers want to predict & prevent Readmissions within 30 days, that treat the same condition within a 30 day window. Otherwise face non-reimbursements.
- Need to make use of multiple data sources: Electronic health records, Lab systems, Claims systems, Population Health evidence-based data (for baseline criteria), etc.
- Massive number of Attributes from many data sources need to be analyzed: difficult to manually create the business rules needed to predict

Technical

- Massive capacity required for Data ingest, preparation, staging, analysis, refinement
- Time consuming to Find out (a) which measures ("features") have the most predictive power
- Have to iteratively experiment with and measure different algorithms to find the best predictive model


UCI Machine Learning Rep...
Share Browser
WebEx
About
Citation Policy
Donat
Repository
Vie
UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Diabetes 130-US hospitals for years 1999-2008 Data Set

Download: Data Folder, Data Set Description

Abstract: This data has been prepared to analyze factors related to readmission as well as other outcomes pertaining to patients with diabetes.

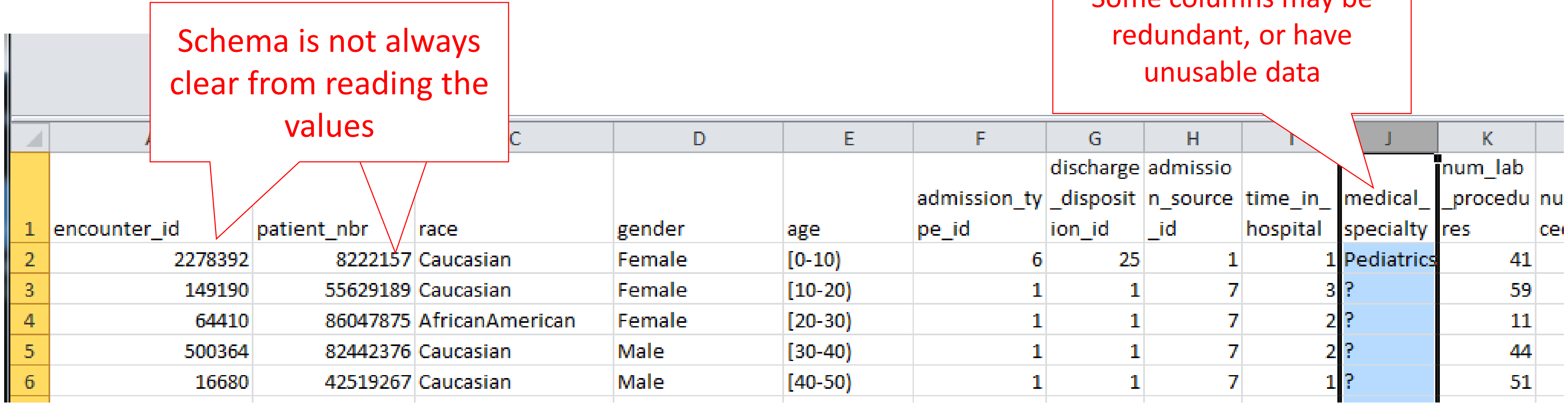
Data Set Characteristics:	Multivariate	**Number of Instances:**	100000	**Area:**	Life
Attribute Characteristics:	Integer	**Number of Attributes:**	55	**Date Donated**	2014-05-03
Associated Tasks:	Classification, Clustering	**Missing Values?**	Yes	**Number of Web Hits:**	133553

Source:

Problem domain (All verticals)

“Featurization” of data (Features in raw data may become Columns / Attributes in BI / DW)

- Features often need to be extracted from lots of input data examples
- What is best tool to transform raw data into types that lend themselves to regression / classification? Target use cases (binary Classification , multi-class Classification, Regression) work best with different tools
- Which columns are just noise (depends on other columns or have useless data) and can be removed to improve throughput?
- Often Requires many cycles to find the best models, and therefore benefits from parallelization and elastic compute and storage services



Schema is not always clear from reading the values

Some columns may be redundant, or have unusable data

	A		C	D	E	F	G	H	I	J	K	
1	encounter_id	patient_nbr	race	gender	age	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	medical_specialty	num_lab_procedures	nu ce
2	2278392	8222157	Caucasian	Female	[0-10)	6	25	1	1	Pediatrics	41	
3	149190	55629189	Caucasian	Female	[10-20)	1	1	7	3	?	59	
4	64410	86047875	AfricanAmerican	Female	[20-30)	1	1	7	2	?	11	
5	500364	82442376	Caucasian	Male	[30-40)	1	1	7	2	?	44	
6	16680	42519267	Caucasian	Male	[40-50)	1	1	7	1	?	51	

Problem domain: Data Analysis today is constrained by (a) inbound Data pipelines and (b) speed of analytics workflow

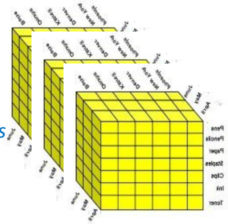
Capture data from **carefully defined** Enterprise sources

Financial Management
Customer Relationship Management
Supply Chain Management
ERP SYSTEM
Human Resource Management
Manufacturing Resource Planning

- *New sources constantly added*
- *Constantly growing in volume, variety, velocity*

Traditional ETL processes into BI/DW repository

Stores only carefully formatted data



Reporting processes

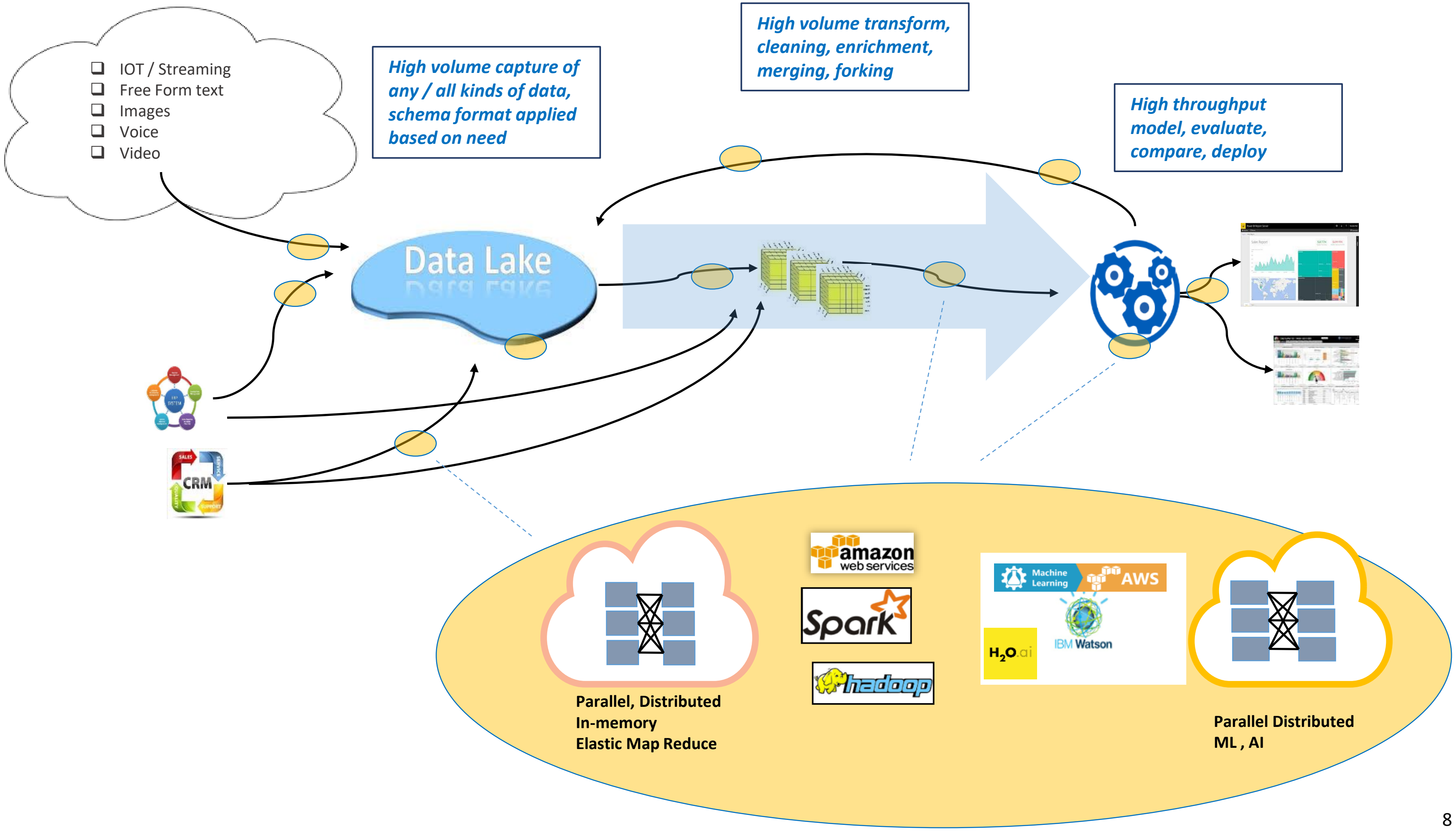
- *Existing Data transform & repository is Server-based, Static, Schema-first designs*
- *Expensive to reformat new data at scale: volume, velocity, variety requirements*
- *Don't really support self-service pipelines*

Predefined analytics processes and information

Existing analytics tools can't model, evaluate, retest, deploy new insights fast enough for new business objectives

- ***Current architecture:** historical Analysis of structured data using hard-coded business rules*
- ***New architecture : Predictive** analysis that learns business rules with massive amounts of training data*

Removing the Constraints: Let Data Science / Data Analysts better utilize the upstream data pipeline



- ❑ IOT / Streaming
- ❑ Free Form text
- ❑ Images
- ❑ Voice
- ❑ Video

High volume capture of any / all kinds of data, schema format applied based on need

High volume transform, cleaning, enrichment, merging, forking

High throughput model, evaluate, compare, deploy

Parallel, Distributed In-memory Elastic Map Reduce

Parallel Distributed ML , AI

Demo tools discussion : Elastic Map Reduce (EMR) & Machine Learning (ML)

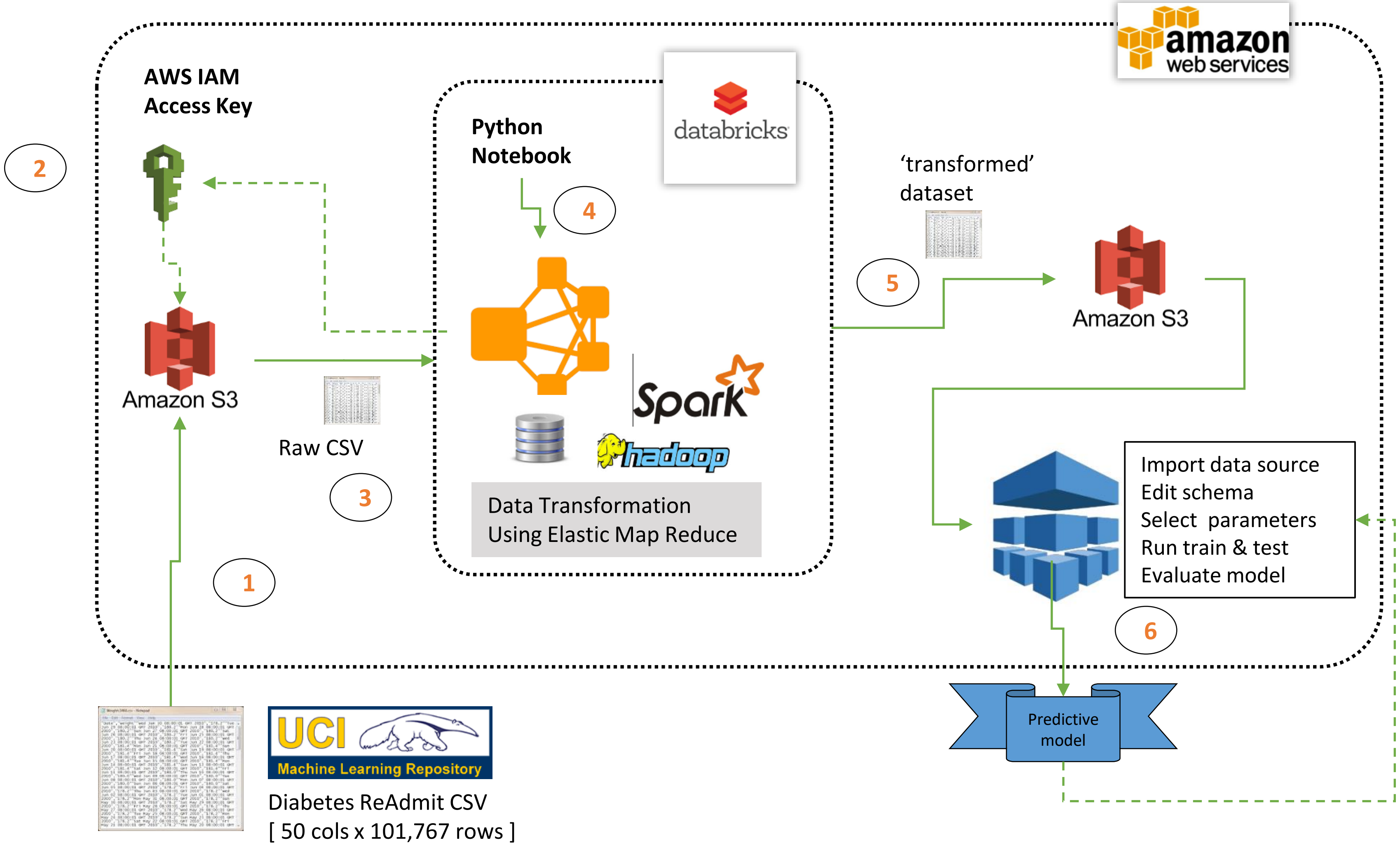
What is EMR and how does it augment "traditional" ETL, BI, DW?

- ETL, BI, DW still fit into the overall data pipeline but EMR provides the elastic compute and storage capacity to handle massive intermediate data processing via parallelization over large clusters
- For example, AWS provides ETL and DW systems for source and target stages in the EMR pipeline
- EMR used to mean Hadoop framework for disk-based batch, but now has evolved to include in-memory Apache Spark and other in-memory distributed process frameworks

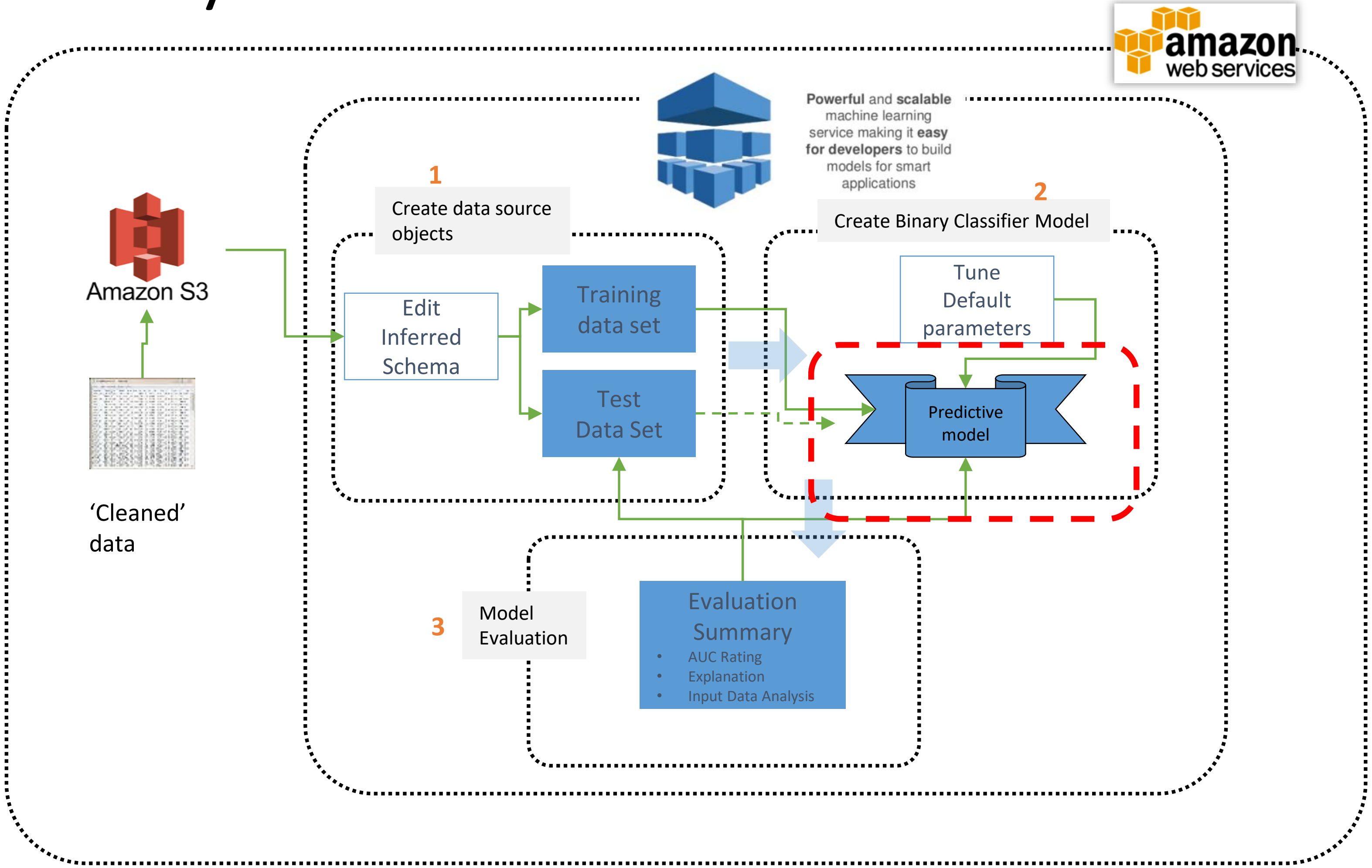
Why is it called Machine Learning?

- Automates many of the feature extraction, model creation, model evaluation, and model refinement steps by using parallel processing
- The business rules for prediction are inferred automatically, without explicit coding, based on training examples that contain the correct answer.
- It is able to infer the "rules" from data that is either too voluminous, too variable, or has too many attributes to specify in a rules table. Some examples of data include free text, voice, image, video, or tables with a large number of attributes such as health records.
- Algorithms include Classification such as Binary , Multi-class, Regression, Neural nets, etc

Demo Data Workflow



Demo analytics workflow



Demo

Summary / Lessons Learned

Demo Summary / lessons learned (cont.)

The ML Predictive model created by my demo is considered poor (not much better than random guessing)

- *AWS ML suggests that these iterations may improve Model.*
- *Collect more data: Increase the number of training examples*
- *Feature processing: Add more variables and better feature processing*
- *Model parameter tuning: Consider alternate values for the training parameters used by your learning algorithm*

Ken's ideas about provided data set

- *for HIPAA compliance, had to strip out needed diagnostic, procedure codes and the like that fall under PHI*
- *Additional demo graphic data (such as Zip codes) or Census Block data, would likely add predictive value to a model. Again, too specific for public data.*

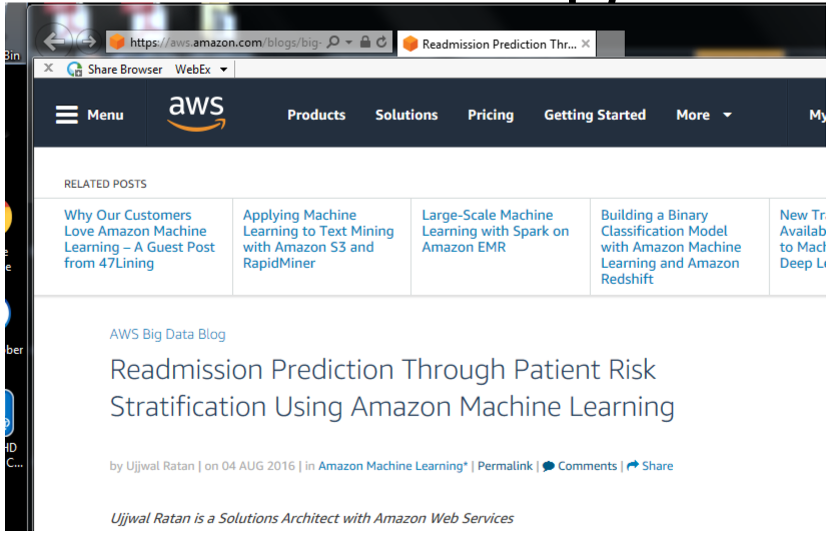
For Healthcare and other complex needs, a more specialized ML not provided by AWS may be required.

- *Note that you can still use AWS as an infrastructure platform in these cases. For example, vendors like H20.ai or DataRobot may be a good fit.*

Demo tools: AWS Machine Learning

Provides a ready-to-use Diabetes example, with instructions and sample data

1. AWS ML Low cost
2. Automatically divide the data into Train and Test sets (70%/30% split)
3. Ask analyst to refine the schema as needed from the inferred schema
4. Ask analyst to specify the output (Binary Classification) value if present
5. Create a Model & Evaluation of Model's predictive quality0



RELATED POSTS

Why Our Customers Love Amazon Machine Learning – A Guest Post from 47Lining

Applying Machine Learning to Text Mining with Amazon S3 and RapidMiner

Large-Scale Machine Learning with Spark on Amazon EMR

Building a Binary Classification Model with Amazon Machine Learning and Amazon Redshift

New Tr Availab to Mach Deep L

AWS Big Data Blog

Readmission Prediction Through Patient Risk Stratification Using Amazon Machine Learning

by Ujjwal Ratan | on 04 AUG 2016 | in Amazon Machine Learning* | Permalink | Comments | Share

Ujjwal Ratan is a Solutions Architect with Amazon Web Services

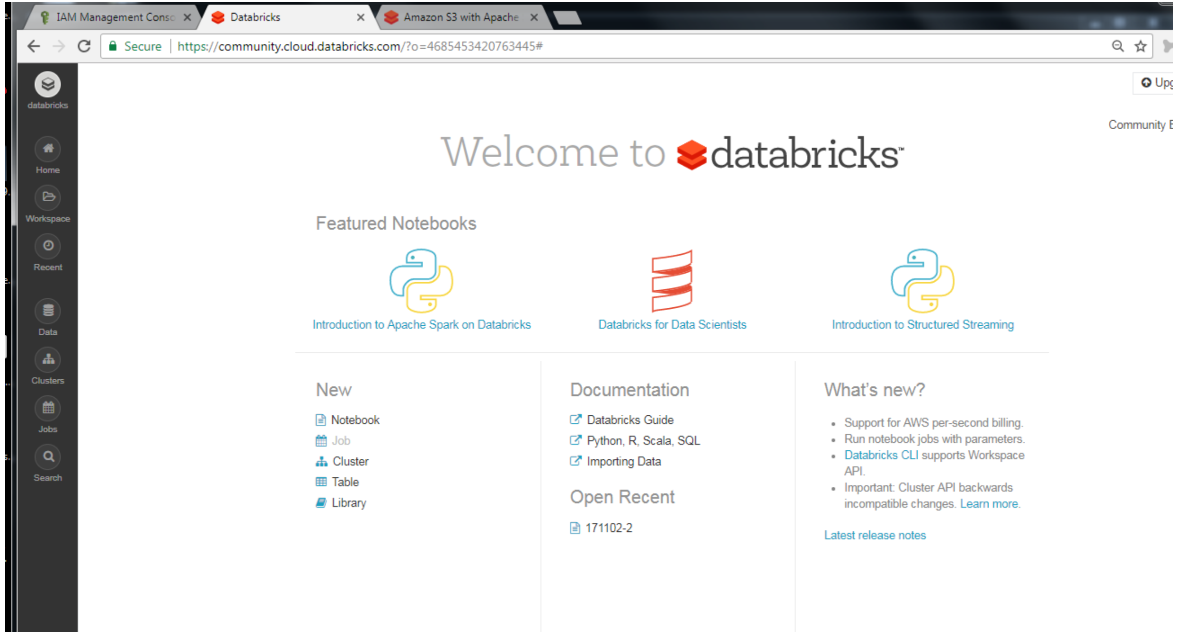
I tailored portions of the example

- *AWS Tutorial Example uses Redshift, but I just used S3 in my account*
- *Enabled remote read / write into S3 from an external EMR cluster, to simulate a more complex Data Pipeline*
- *In the external EMR cluster (actually runs on AWS itself, but not in my account) added some Python "data engineering" DataFrame code, very simple*
- *Omitted a couple of the Redshift SQL steps (omitted Joins of Admit, Discharge codes as Numerical Categories instead of String Categories)*

Demo tools: Apache Spark

Elastic Map Reduce (EMR)

- de-facto standard for in-memory elastic processing
- Platform, API, SDK standards
- Improves throughput via integration with Hadoop framework
- Rish Support for Parallel distributed processing
- Rich language and ML library support (Python seems to be most popular but Scala and R also have their following)



Welcome to databricks

Featured Notebooks

Introduction to Apache Spark on Databricks

Databricks for Data Scientists

Introduction to Structured Streaming

New

- Notebook
- Job
- Cluster
- Table
- Library

Documentation

- Databricks Guide
- Python, R, Scala, SQL
- Importing Data

Open Recent

- 171102-2

What's new?

- Support for AWS per-second billing.
- Run notebook jobs with parameters.
- Databricks CLI supports Workspace API.
- Important: Cluster API backwards incompatible changes. Learn more.

Latest release notes

AWS has core EMR / Spark services, but I used Databricks as an integrated Front-end to AWS for convenience

- Free community edition (includes Notebooks to run interactive code, file-import features, but you need licensed version for Jobs management on a larger cluster)
- Fully supports Apache Spark standard as a major contributor
- Convenient UI with Notebook code execution on Clusters, for Data Engineering and Data Science
- Built-in support for AWS clusters and Data import from S3 and other repositories

appendix

Demo Summary / lessons learned

- Created S3 bucket as a prototype Data Lake
- *stores raw data of any format, scales to almost infinite size and provides tools to overlay schemas for extraction*
- Python
- *Why Python versus Scala or R? Python seems more (Big) Data Scientist friendly than R. R has been used a lot for workstation-level statistics but not as common in EMR. Scala may have better performance for some intensive data transformation processes.*
- Importance of a Data Pipeline approach to feed Advanced Analytics
- *Key components: EMR clusters, Micro-services & Containers (Kubernetes seems to be gaining traction)*
- *Need to have flexibility for Ad-hoc, agile way to insert, merge, split, serialize, parallelize data streams*
- *Tradeoff between batch and interactive needed for ML. we should be conversant in both Hadoop and Apache Spark architectures. (They're actually converging anyway)*

How I setup ML and EMR accounts

- S3 account, Free tier

1. Create buckets / folders
2. Upload the Diabetes dataset
3. Setup Spark Users, permissions, access keys

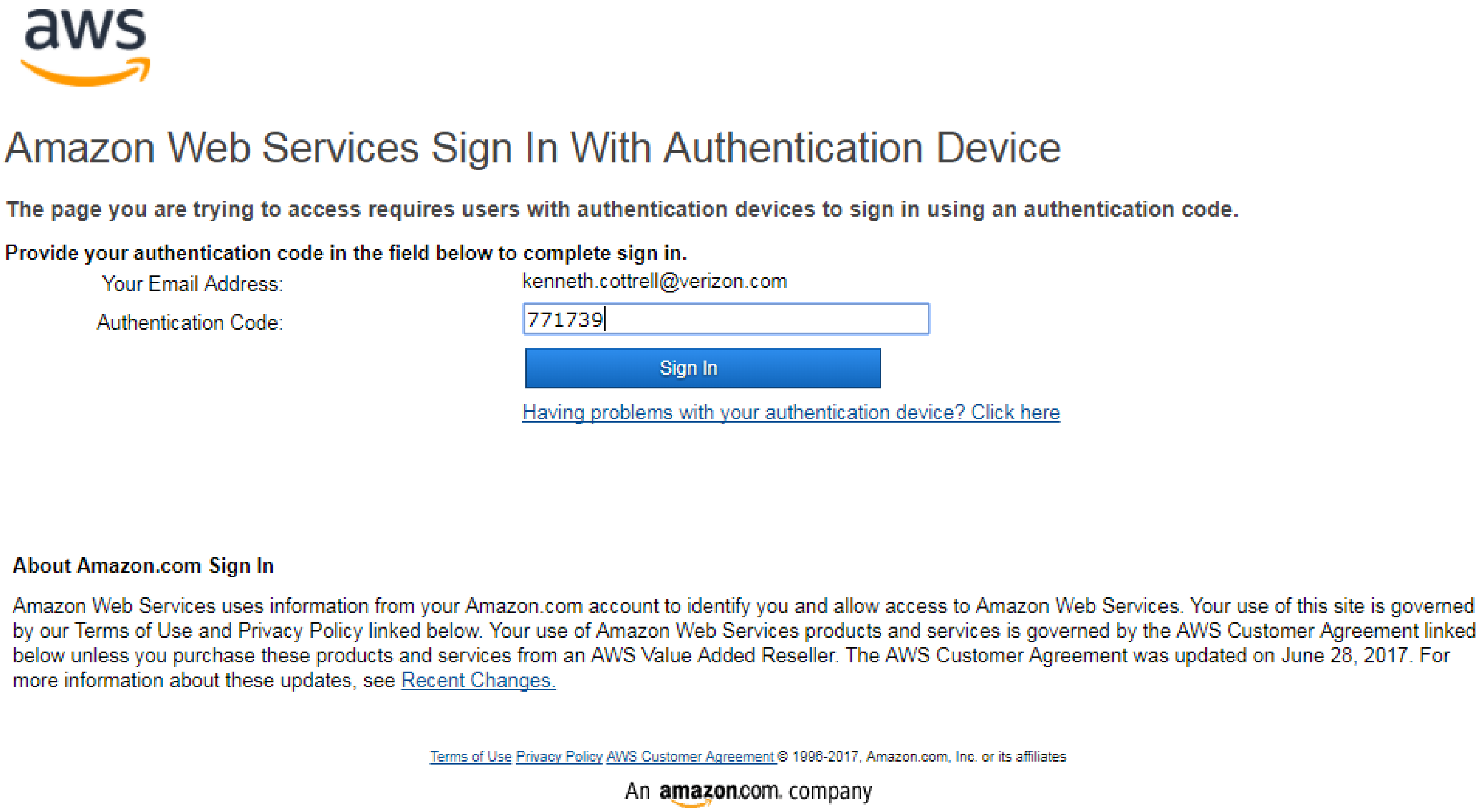
- Databricks account (free community edition)

1. Run a Cluster, create a Notebook to run in cluster
2. In Notebook key in Python Code (Shift-Return for each command cell)
3. Insert my Access Key and Secret Key from a special Spark user in IAM
4. code to Setup Read access to S3
5. code to transform data
6. code to Write cleaned Dataframe to S3

AWS ML

1. Set ML DataSource to S3 bucket folder with Cleaned from external EMR
2. Change some of the inferred column schemas
3. Select default Train/test mix (70/30)
4. Select binary classification output label (Readmitted = yes / no)
5. Run the train / evaluation session
6. Refresh screen to see when completed
7. Look at evaluation of model quality (ability to predict Positives / negatives
8. (not tried – run a sample record for a sample prediction)

MFA for your account



Amazon Web Services Sign In With Authentication Device

The page you are trying to access requires users with authentication devices to sign in using an authentication code.

Provide your authentication code in the field below to complete sign in.

Your Email Address: kenneth.cottrell@verizon.com

Authentication Code: 771739

Having problems with your authentication device? Click here

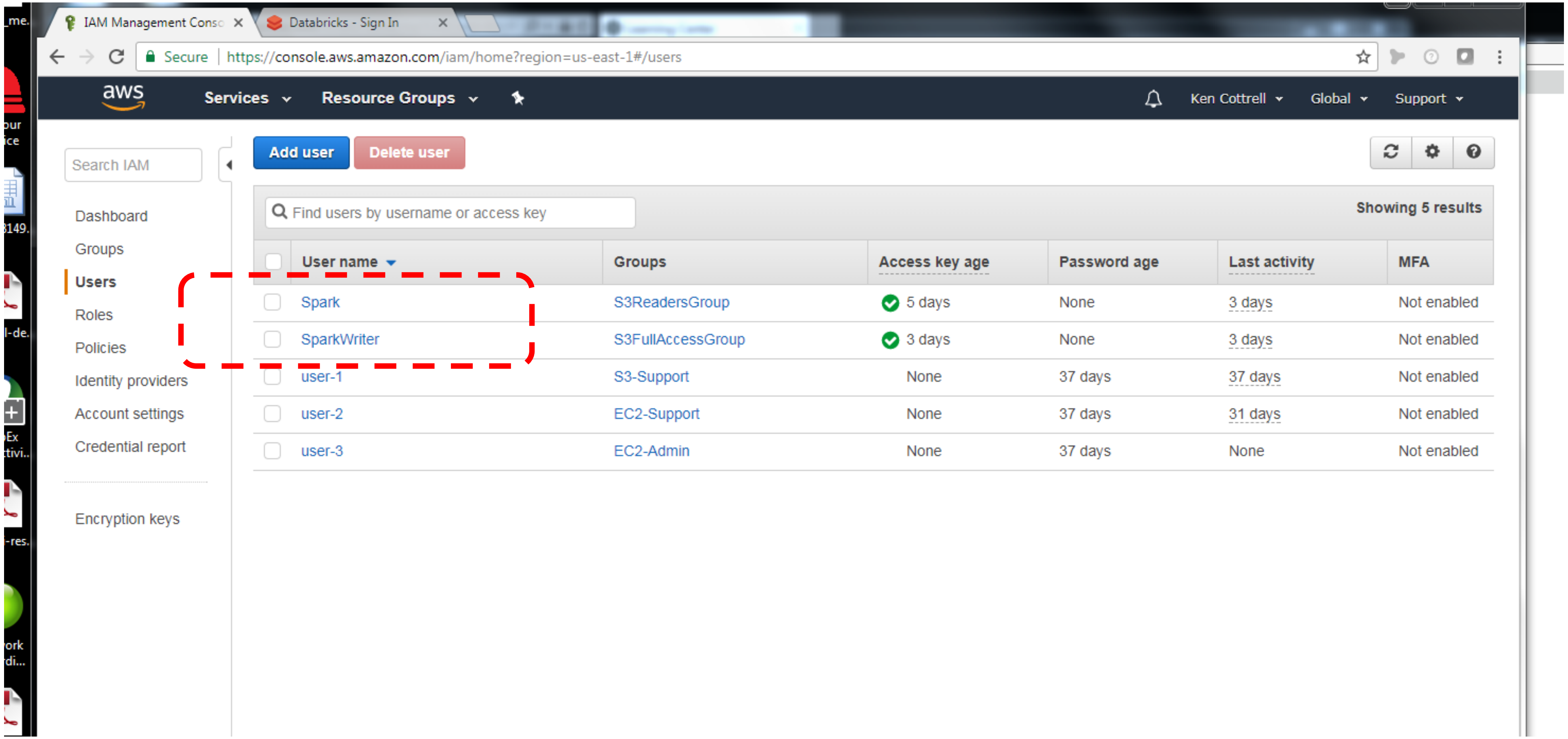
About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our Terms of Use and Privacy Policy linked below. Your use of Amazon Web Services products and services is governed by the AWS Customer Agreement linked below unless you purchase these products and services from an AWS Value Added Reseller. The AWS Customer Agreement was updated on June 28, 2017. For more information about these updates, see Recent Changes.

Terms of Use Privacy Policy AWS Customer Agreement © 1996-2017, Amazon.com, Inc. or its affiliates

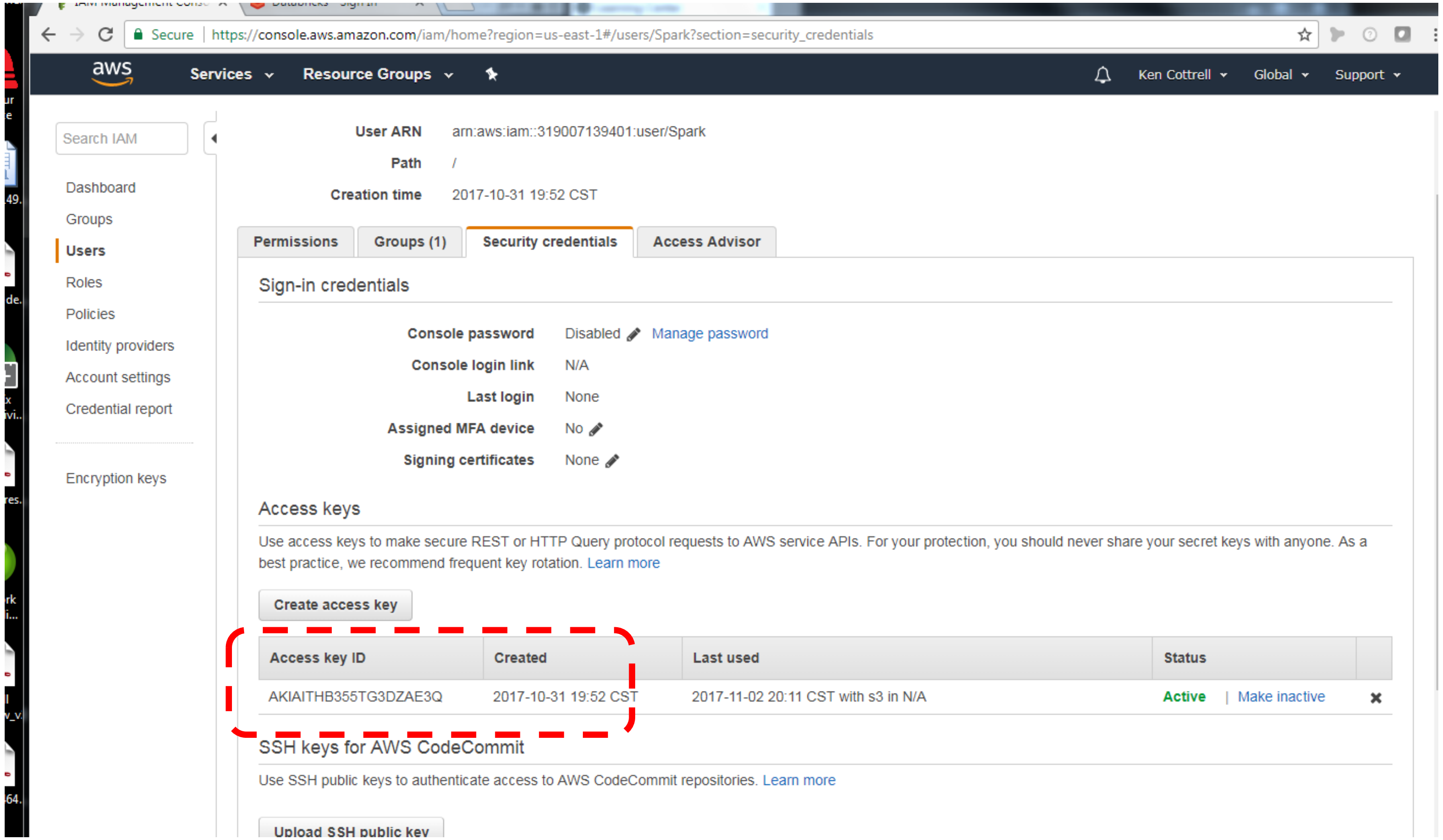
An amazon.com. company

AWS IAM to create Spark users from external EMR



User name	Groups	Access key age	Password age	Last activity	MFA
Spark	S3ReadersGroup	✓ 5 days	None	3 days	Not enabled
SparkWriter	S3FullAccessGroup	✓ 3 days	None	3 days	Not enabled
user-1	S3-Support	None	37 days	37 days	Not enabled
user-2	EC2-Support	None	37 days	31 days	Not enabled
user-3	EC2-Admin	None	37 days	None	Not enabled

IAM Spark user Access Key



User ARN arn:aws:iam::319007139401:user/Spark

Path /

Creation time 2017-10-31 19:52 CST

Sign-in credentials

Console password Disabled Manage password

Console login link N/A

Last login None

Assigned MFA device No

Signing certificates None

Access keys

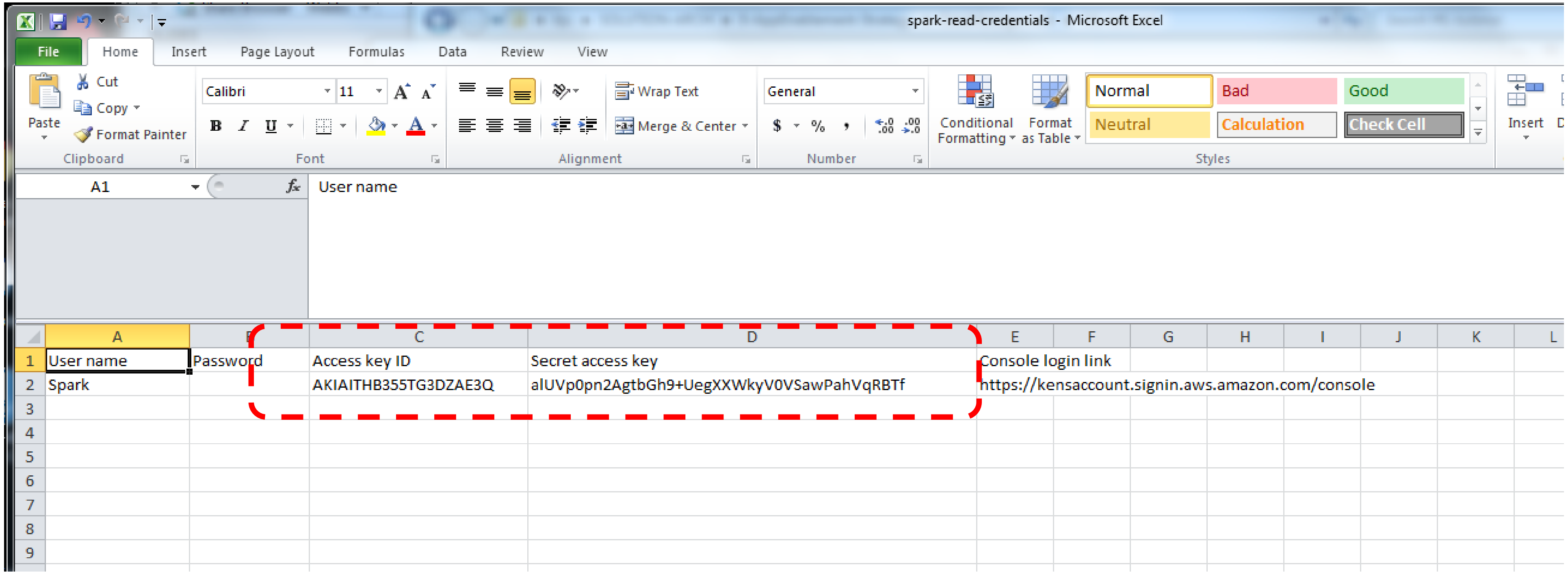
Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

Access key ID	Created	Last used	Status	
AKIAITHB355TG3DZAE3Q	2017-10-31 19:52 CST	2017-11-02 20:11 CST with s3 in N/A	Active | Make inactive	✖

SSH keys for AWS CodeCommit

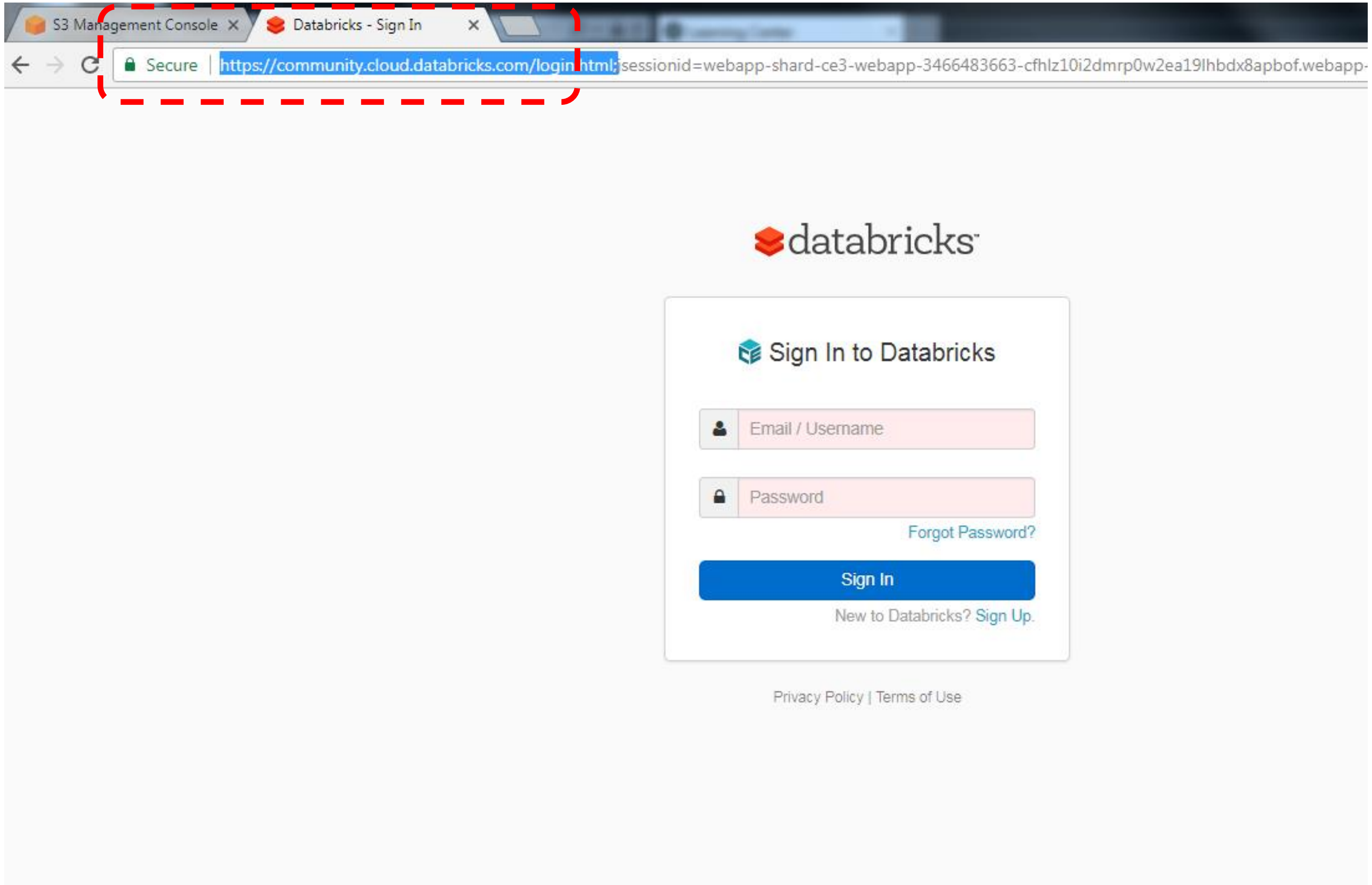
Use SSH public keys to authenticate access to AWS CodeCommit repositories. Learn more

AWS Spark user (in IAM) Credentials to use in Python code



	A	B	C	D	E
1	User name	Password	Access key ID	Secret access key	Console login link
2	Spark		AKIAITHB355TG3DZAE3Q	alUVp0pn2AgtbGh9+UegXXWkyV0VSawPahVqRBTf	https://kensaccount.signin.aws.amazon.com/console

DateBricks Community Edition



S3 Management Console

Databricks - Sign In

Secure | https://community.cloud.databricks.com/login.html;jsessionid=webapp-shard-ce3-webapp-3466483663-cfhlz10i2dmrp0w2ea19lhbdx8apbof.webapp-

databricks

Sign In to Databricks

Email / Username

Password

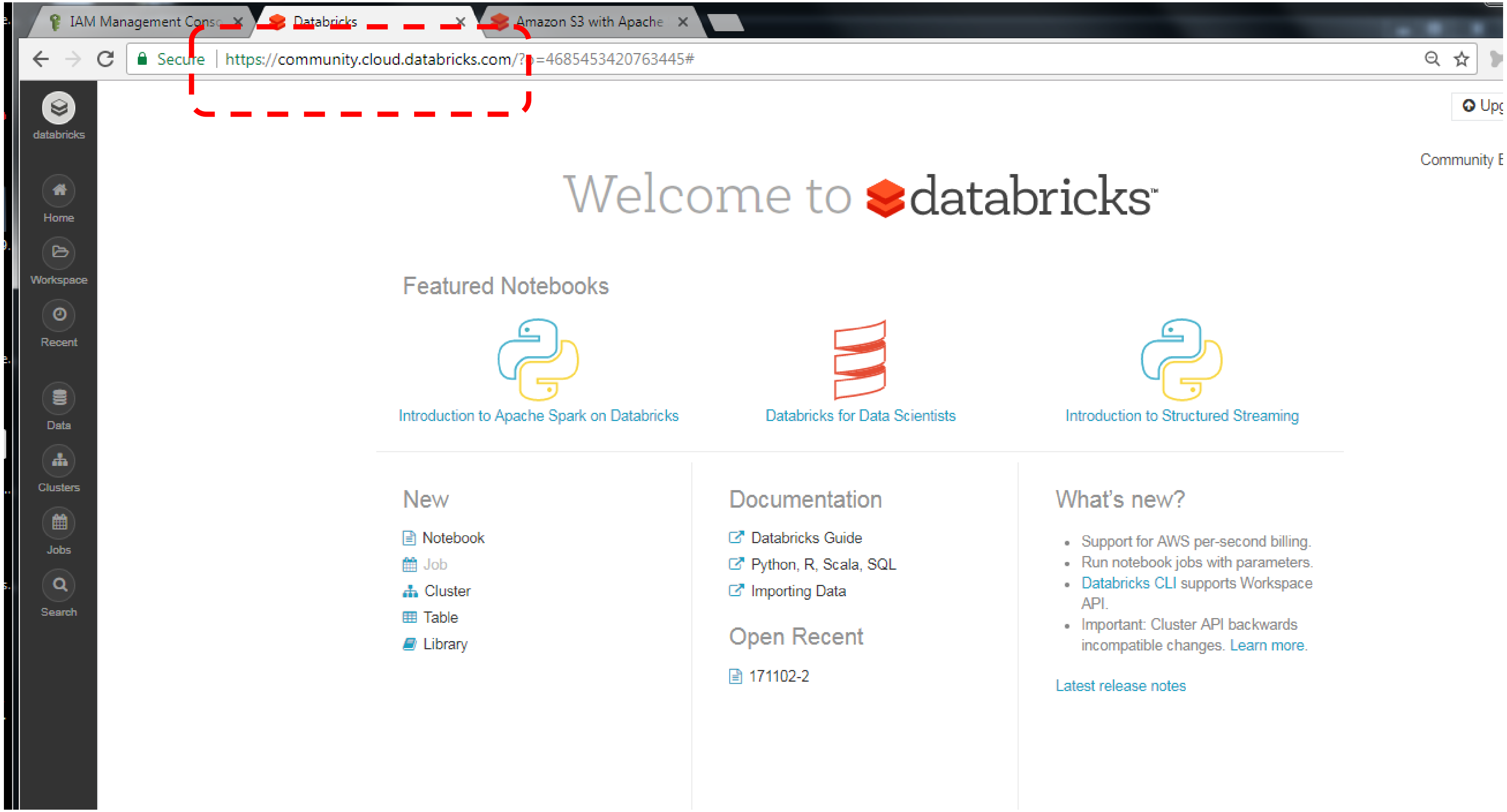
Forgot Password?

Sign In

New to Databricks? Sign Up.

Privacy Policy | Terms of Use

Data Bricks landing page



Welcome to databricks™

Featured Notebooks

Introduction to Apache Spark on Databricks

Databricks for Data Scientists

Introduction to Structured Streaming

New

- Notebook
- Job
- Cluster
- Table
- Library

Documentation

- Databricks Guide
- Python, R, Scala, SQL
- Importing Data

Open Recent

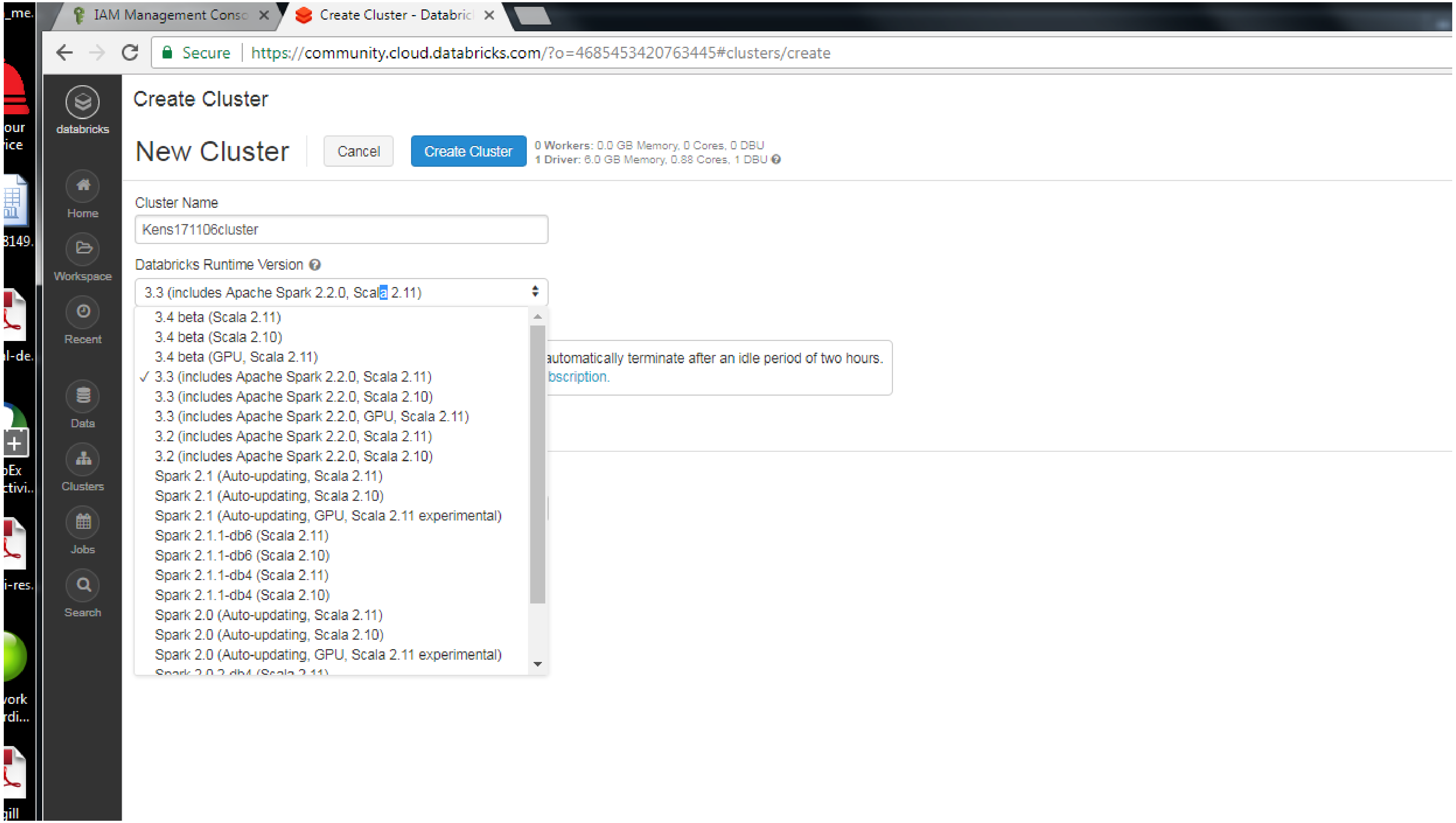
- 171102-2

What's new?

- Support for AWS per-second billing.
- Run notebook jobs with parameters.
- Databricks CLI supports Workspace API.
- Important: Cluster API backwards incompatible changes. Learn more.

Latest release notes

Starting a cluster for the Notebook


IAM Management Conso
Create Cluster - Databric
Secure
https://community.cloud.databricks.com/?o=4685453420763445#clusters/create
databricks
Home
Workspace
Recent
Data
Clusters
Jobs
Search
Cancel
Create Cluster

Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU

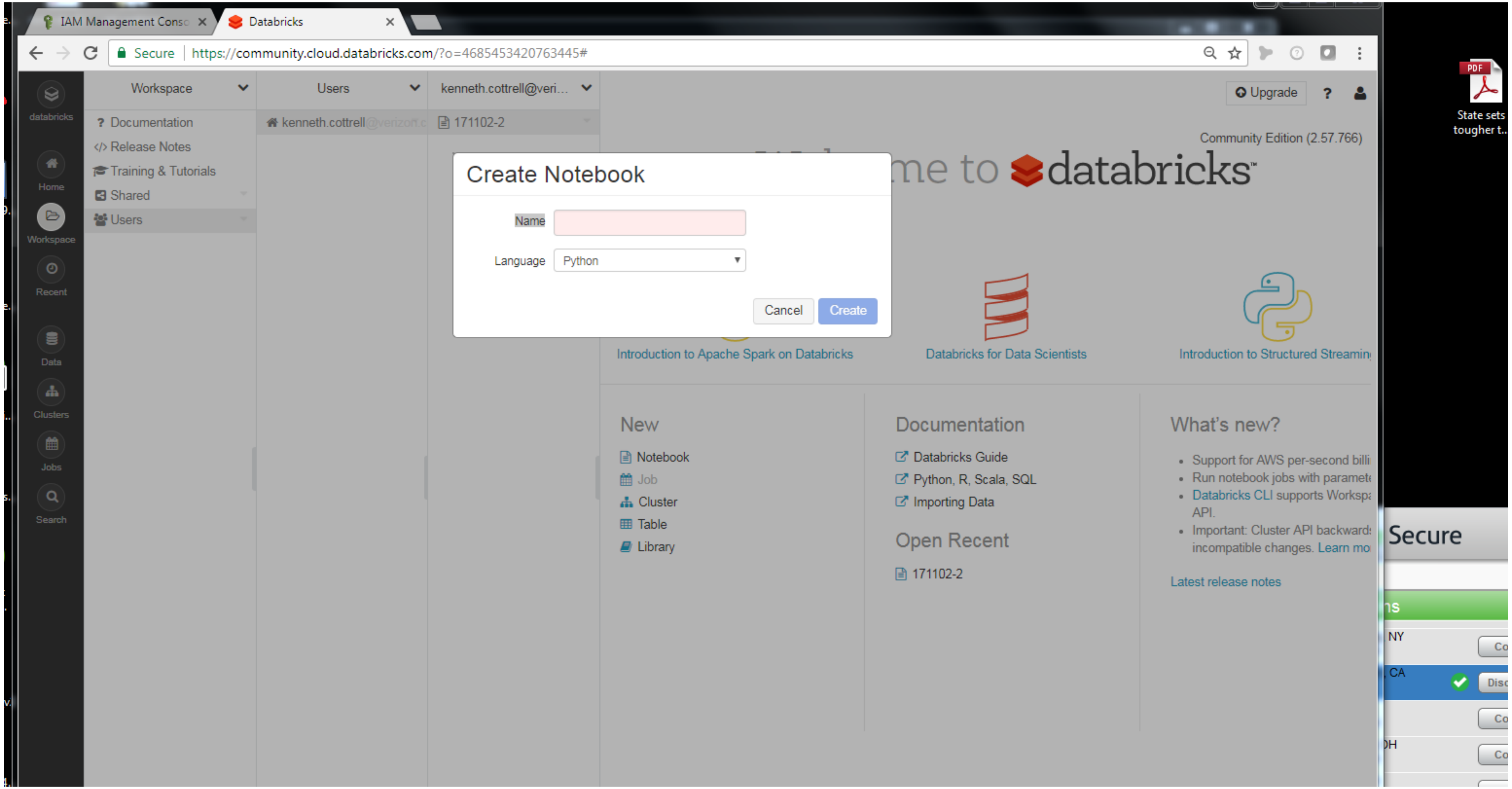
Cluster Name: Kens171106cluster

Databricks Runtime Version: 3.3 (includes Apache Spark 2.2.0, Scala 2.11)

- 3.4 beta (Scala 2.11)
- 3.4 beta (Scala 2.10)
- 3.4 beta (GPU, Scala 2.11)
- ✓ 3.3 (includes Apache Spark 2.2.0, Scala 2.11)
- 3.3 (includes Apache Spark 2.2.0, Scala 2.10)
- 3.3 (includes Apache Spark 2.2.0, GPU, Scala 2.11)
- 3.2 (includes Apache Spark 2.2.0, Scala 2.11)
- 3.2 (includes Apache Spark 2.2.0, Scala 2.10)
- Spark 2.1 (Auto-updating, Scala 2.11)
- Spark 2.1 (Auto-updating, Scala 2.10)
- Spark 2.1 (Auto-updating, GPU, Scala 2.11 experimental)
- Spark 2.1.1-db6 (Scala 2.11)
- Spark 2.1.1-db6 (Scala 2.10)
- Spark 2.1.1-db4 (Scala 2.11)
- Spark 2.1.1-db4 (Scala 2.10)
- Spark 2.0 (Auto-updating, Scala 2.11)
- Spark 2.0 (Auto-updating, Scala 2.10)
- Spark 2.0 (Auto-updating, GPU, Scala 2.11 experimental)
- Spark 2.0.2-db4 (Scala 2.11)

automatically terminate after an idle period of two hours.
bscription.

Create a Notebook



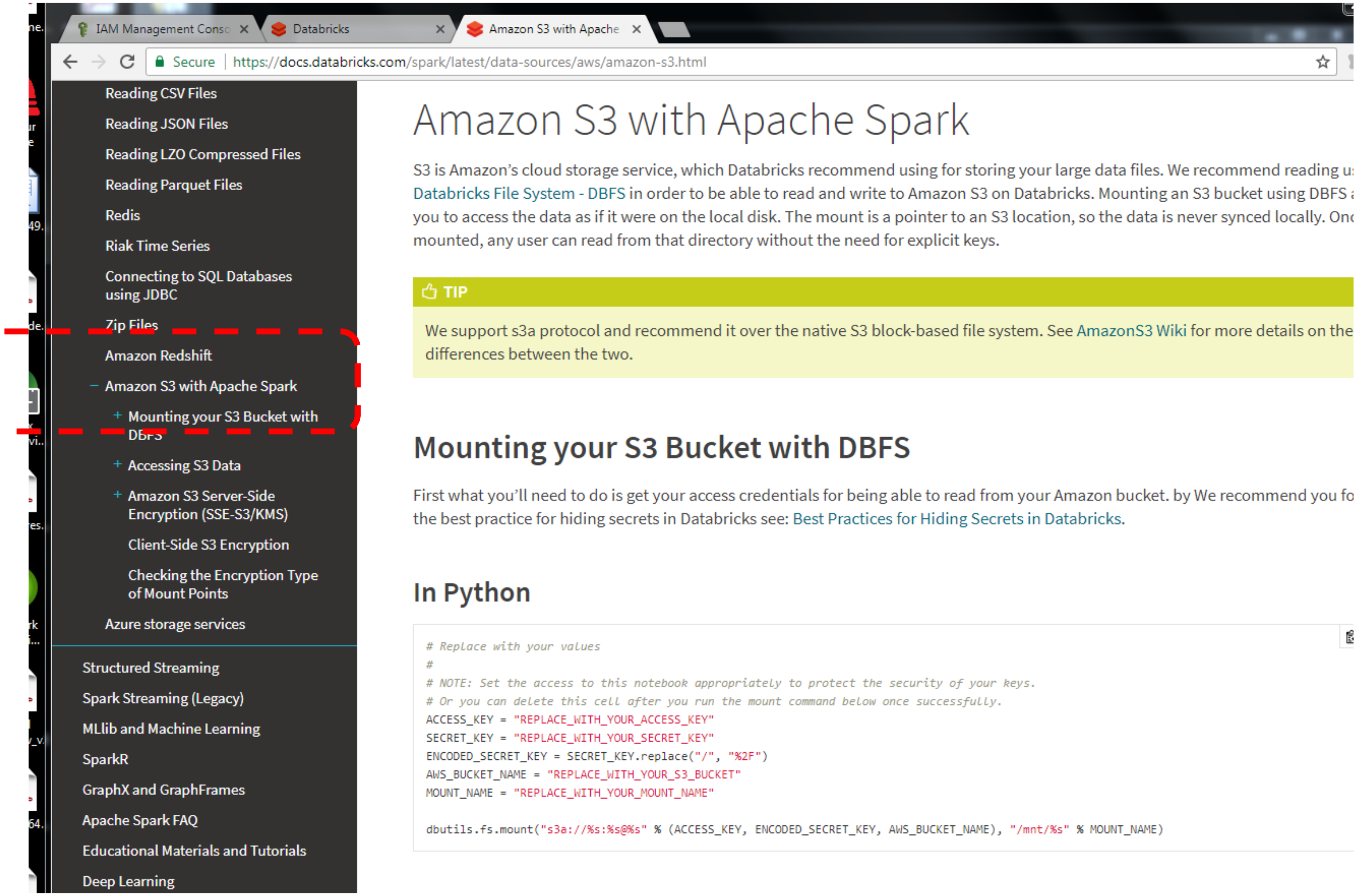
Create Notebook

Name

Language: Python

Cancel | Create

Data Bricks to S3 connection: many options, I used Boto framework


IAM Management Conso
Databricks
Amazon S3 with Apache
Secure
https://docs.databricks.com/spark/latest/data-sources/aws/amazon-s3.html
Reading CSV Files
Reading JSON Files
Reading LZO Compressed Files
Reading Parquet Files
Redis
Riak Time Series
Connecting to SQL Databases using JDBC
Zip Files
Amazon Redshift
Amazon S3 with Apache Spark
Mounting your S3 Bucket with DBFS
Accessing S3 Data
Amazon S3 Server-Side Encryption (SSE-S3/KMS)
Client-Side S3 Encryption
Checking the Encryption Type of Mount Points
Azure storage services
Structured Streaming
Spark Streaming (Legacy)
MLlib and Machine Learning
SparkR
GraphX and GraphFrames
Apache Spark FAQ
Educational Materials and Tutorials
Deep Learning

Amazon S3 with Apache Spark

S3 is Amazon's cloud storage service, which Databricks recommend using for storing your large data files. We recommend reading u Databricks File System - DBFS in order to be able to read and write to Amazon S3 on Databricks. Mounting an S3 bucket using DBFS a you to access the data as if it were on the local disk. The mount is a pointer to an S3 location, so the data is never synced locally. Onc mounted, any user can read from that directory without the need for explicit keys.

TIP

We support s3a protocol and recommend it over the native S3 block-based file system. See AmazonS3 Wiki for more details on the differences between the two.

Mounting your S3 Bucket with DBFS

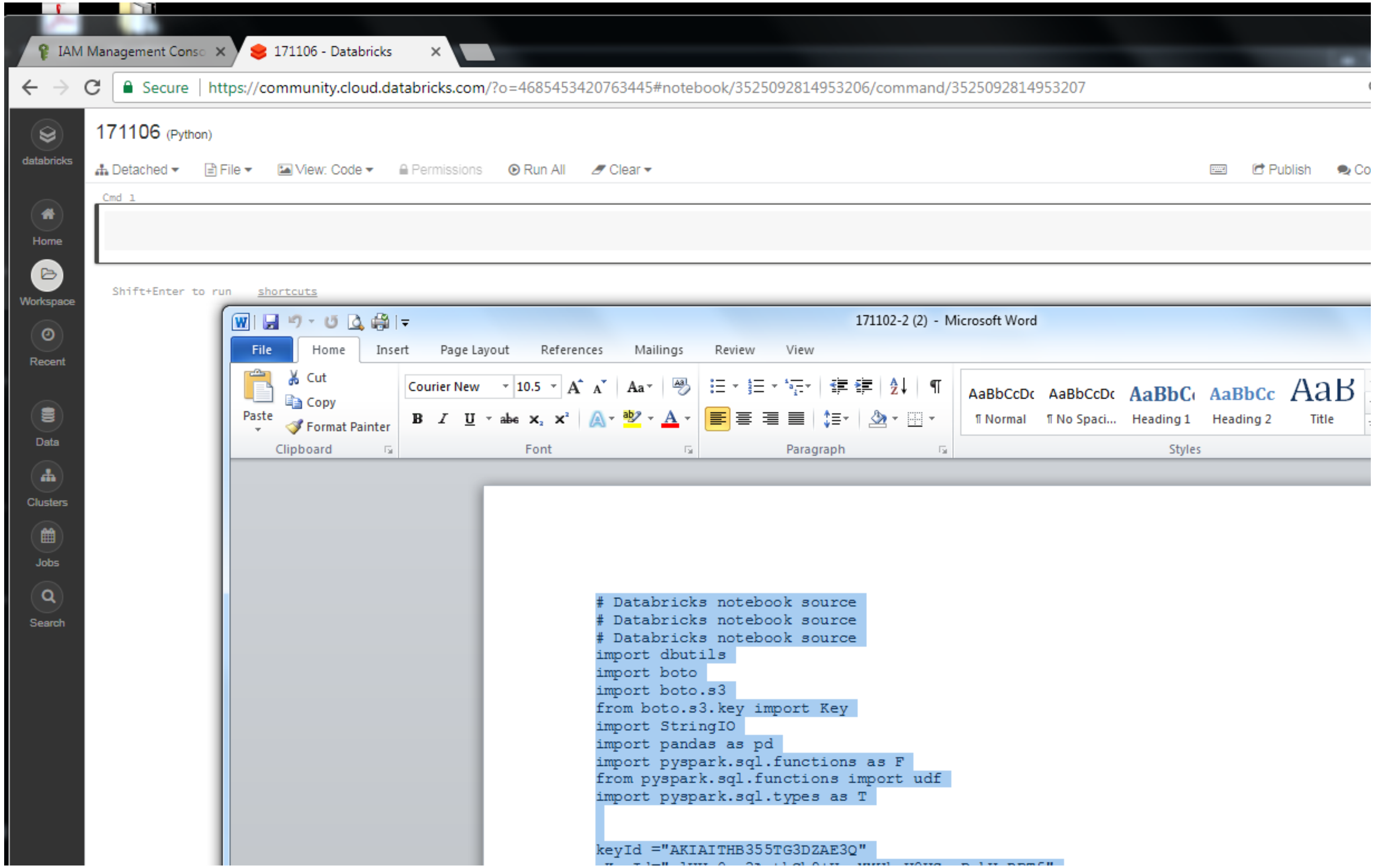
First what you'll need to do is get your access credentials for being able to read from your Amazon bucket. by We recommend you fo the best practice for hiding secrets in Databricks see: Best Practices for Hiding Secrets in Databricks.

In Python

```
# Replace with your values
#
# NOTE: Set the access to this notebook appropriately to protect the security of your keys.
# Or you can delete this cell after you run the mount command below once successfully.
ACCESS_KEY = "REPLACE_WITH_YOUR_ACCESS_KEY"
SECRET_KEY = "REPLACE_WITH_YOUR_SECRET_KEY"
ENCODED_SECRET_KEY = SECRET_KEY.replace("/", "%2F")
AWS_BUCKET_NAME = "REPLACE_WITH_YOUR_S3_BUCKET"
MOUNT_NAME = "REPLACE_WITH_YOUR_MOUNT_NAME"

dbutils.fs.mount("s3a://%s:%s@%s" % (ACCESS_KEY, ENCODED_SECRET_KEY, AWS_BUCKET_NAME), "/mnt/%s" % MOUNT_NAME)
```

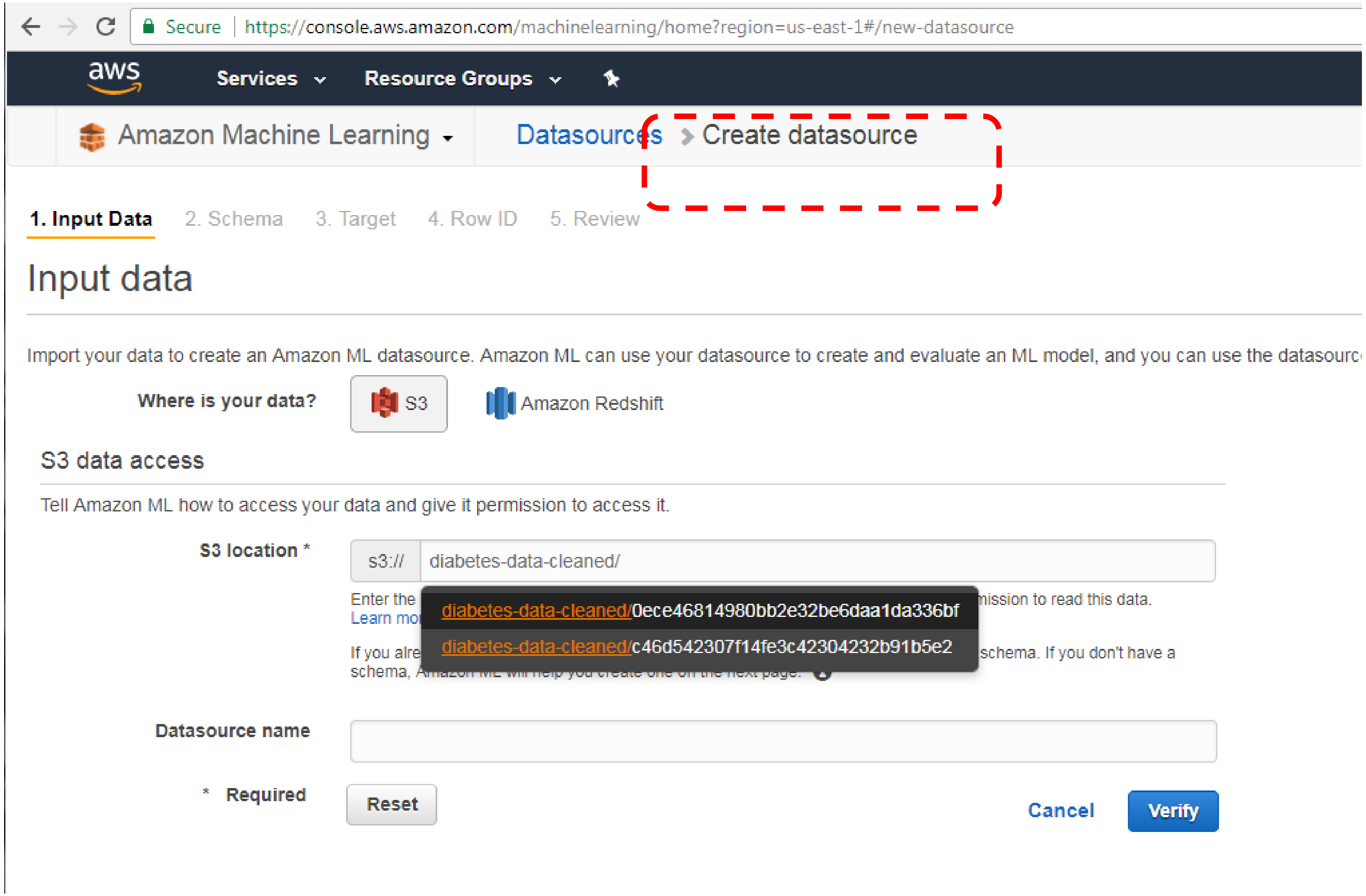
Python coding , you can cut and paste from my Textfile into Notebook. Run one cell at a time or all together. At the end Upload cleaned data to S3



```
# Databricks notebook source
# Databricks notebook source
# Databricks notebook source
import dbutils
import boto
import boto.s3
from boto.s3.key import Key
import StringIO
import pandas as pd
import pyspark.sql.functions as F
from pyspark.sql.functions import udf
import pyspark.sql.types as T


keyId ="AKIAITHB355TG3DZAE3Q"
```

AWS ML screen shots – load cleaned data (sent from EMR) from S3



Input data

Import your data to create an Amazon ML datasource. Amazon ML can use your datasource to create and evaluate an ML model, and you can use the datasourc

Where is your data? S3 Amazon Redshift

S3 data access

Tell Amazon ML how to access your data and give it permission to access it.

S3 location * s3:// diabetes-data-cleaned/

diabetes-data-cleaned/0ece46814980bb2e32be6daa1da336bf

diabetes-data-cleaned/c46d542307f14fe3c42304232b91b5e2

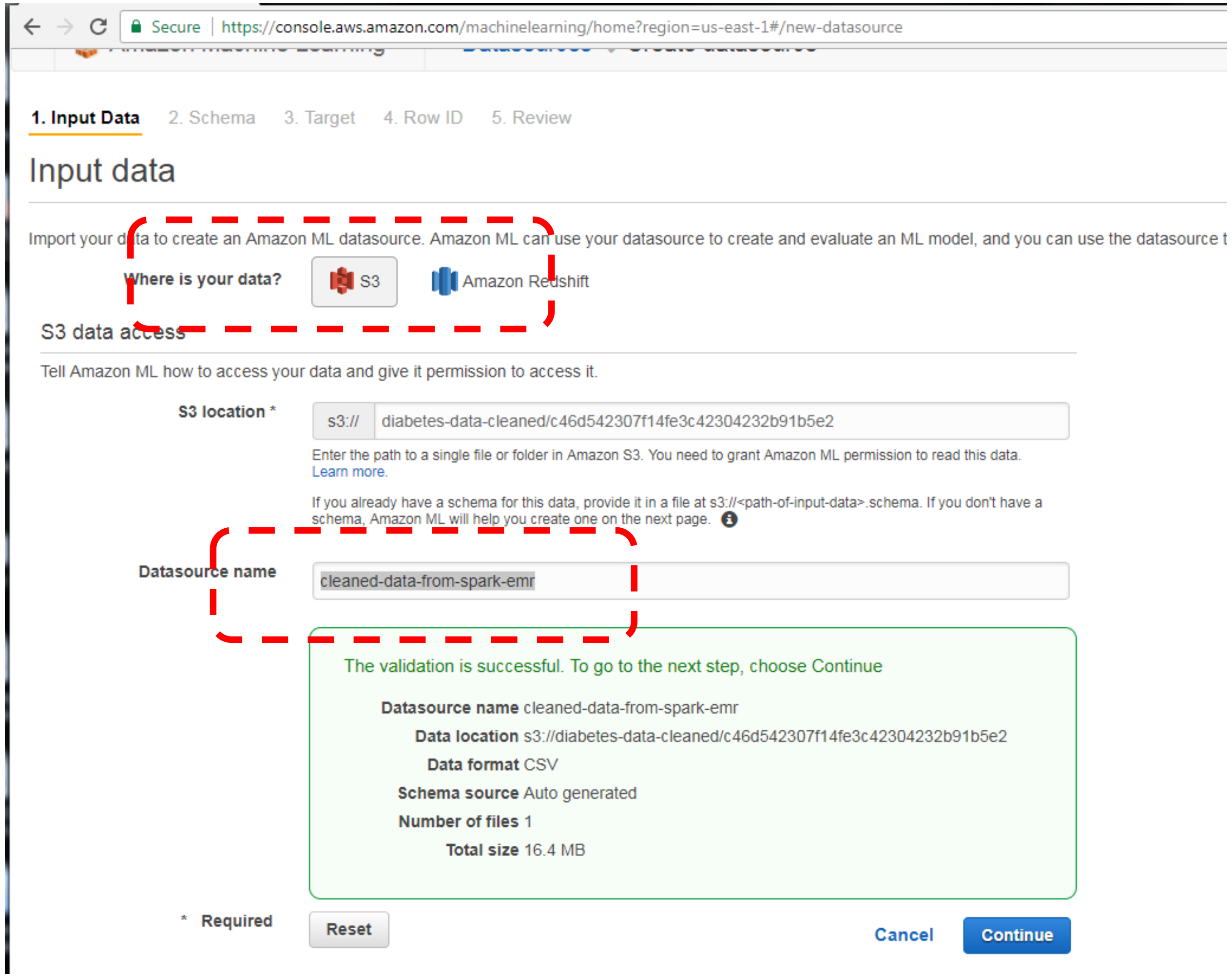
Enter the mission to read this data.
Learn mo

If you alre schema. If you don't have a schema, Amazon ML will help you create one on the next page.

Datasource name

* **Required**

AWS ML asks for permission to use S3, then asks you to proceed



Secure
https://console.aws.amazon.com/machinelearning/home?region=us-east-1#/new-datasource
1. Input Data
2. Schema
3. Target
4. Row ID
5. Review

Input data

Import your data to create an Amazon ML datasource. Amazon ML can use your datasource to create and evaluate an ML model, and you can use the datasource t

Where is your data? S3 Amazon Redshift

S3 data access

Tell Amazon ML how to access your data and give it permission to access it.

S3 location * s3:// diabetes-data-cleaned/c46d542307f14fe3c42304232b91b5e2

Enter the path to a single file or folder in Amazon S3. You need to grant Amazon ML permission to read this data. Learn more.

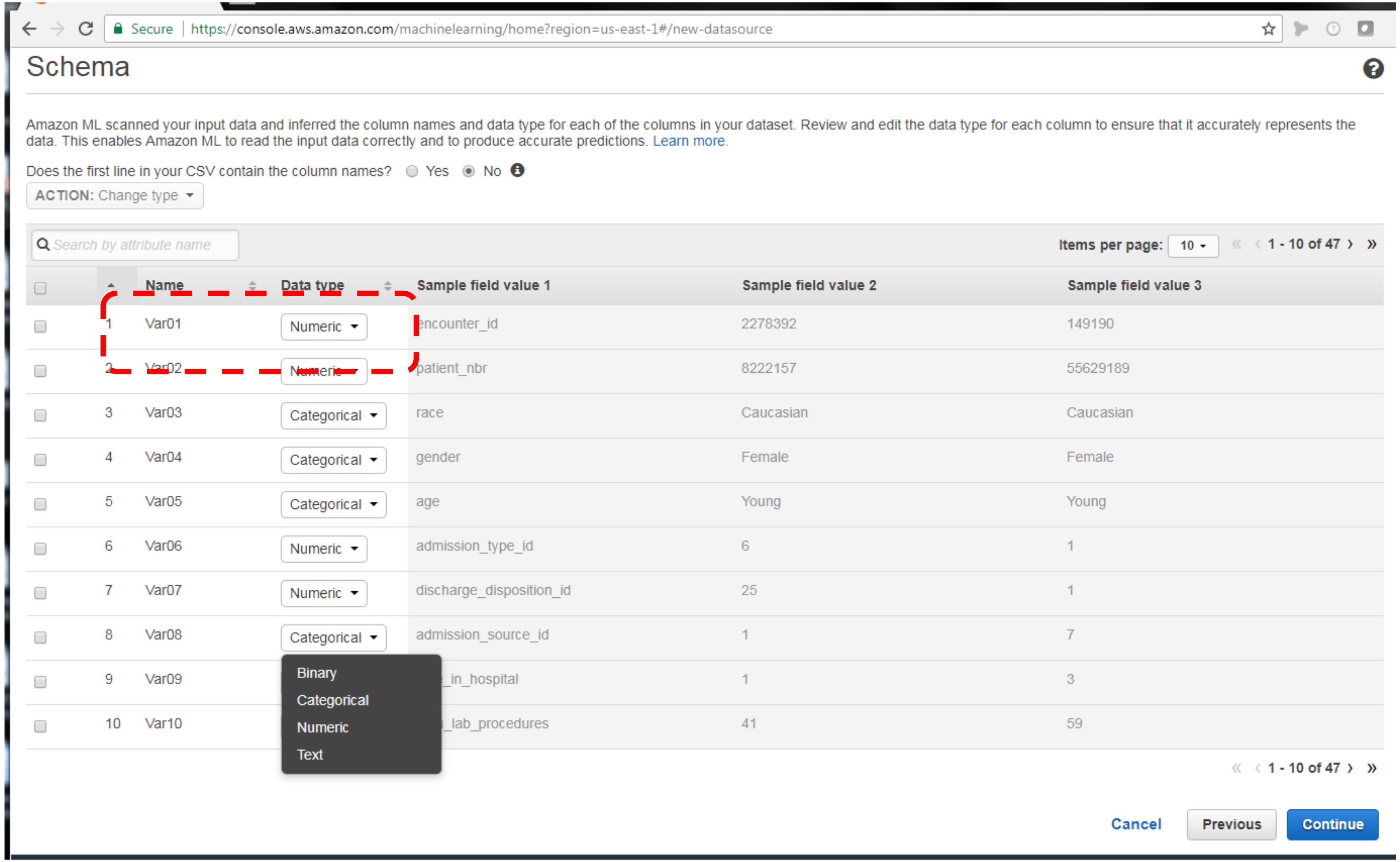
If you already have a schema for this data, provide it in a file at s3://<path-of-input-data>.schema. If you don't have a schema, Amazon ML will help you create one on the next page.

Datasource name cleaned-data-from-spark-emr

The validation is successful. To go to the next step, choose Continue

Datasource name cleaned-data-from-spark-emr
Data location s3://diabetes-data-cleaned/c46d542307f14fe3c42304232b91b5e2
Data format CSV
Schema source Auto generated
Number of files 1
Total size 16.4 MB

Some of the datatypes need to be reset from the inferred schema



Secure | https://console.aws.amazon.com/machinelearning/home?region=us-east-1#/new-datasource

Schema

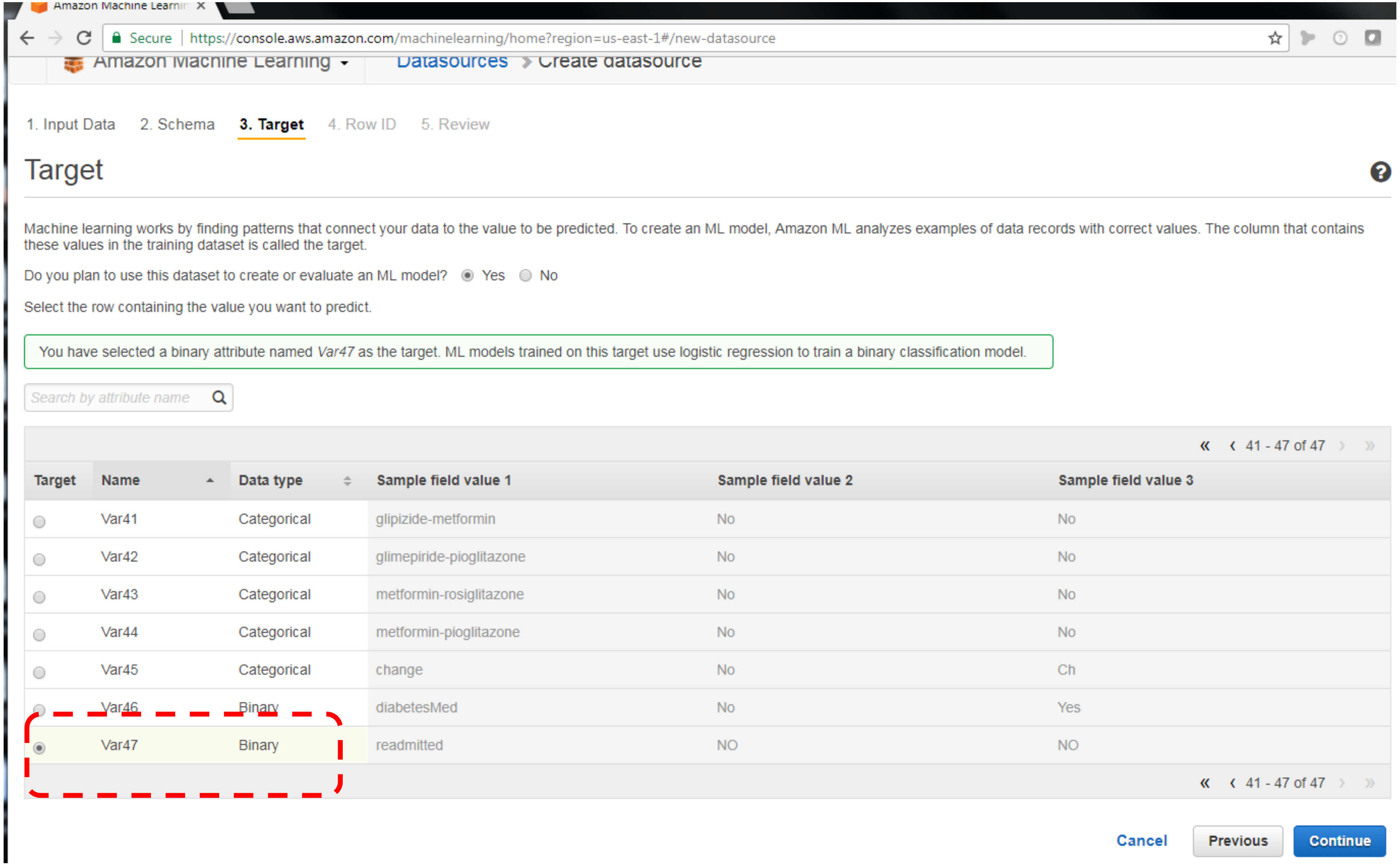
Amazon ML scanned your input data and inferred the column names and data type for each of the columns in your dataset. Review and edit the data type for each column to ensure that it accurately represents the data. This enables Amazon ML to read the input data correctly and to produce accurate predictions. Learn more.

Does the first line in your CSV contain the column names? ○ Yes ● No

ACTION: Change type

□		Name	Data type	Sample field value 1	Sample field value 2	Sample field value 3
□	1	Var01	Numeric	encounter_id	2278392	149190
□	2	Var02	Numeric	patient_nbr	8222157	55629189
□	3	Var03	Categorical	race	Caucasian	Caucasian
□	4	Var04	Categorical	gender	Female	Female
□	5	Var05	Categorical	age	Young	Young
□	6	Var06	Numeric	admission_type_id	6	1
□	7	Var07	Numeric	discharge_disposition_id	25	1
□	8	Var08	Categorical	admission_source_id	1	7
□	9	Var09		_in_hospital	1	3
□	10	Var10		_lab_procedures	41	59

Set the last column (readmitted) as the binary classification output (i.e. the “Label”



Amazon Machine Learnir ×
Secure
https://console.aws.amazon.com/machinelearning/home?region=us-east-1#/new-datasource
Amazon Machine Learning
Datasources > Create datasource
1. Input Data
2. Schema
3. Target
4. Row ID
5. Review

Target

Machine learning works by finding patterns that connect your data to the value to be predicted. To create an ML model, Amazon ML analyzes examples of data records with correct values. The column that contains these values in the training dataset is called the target.

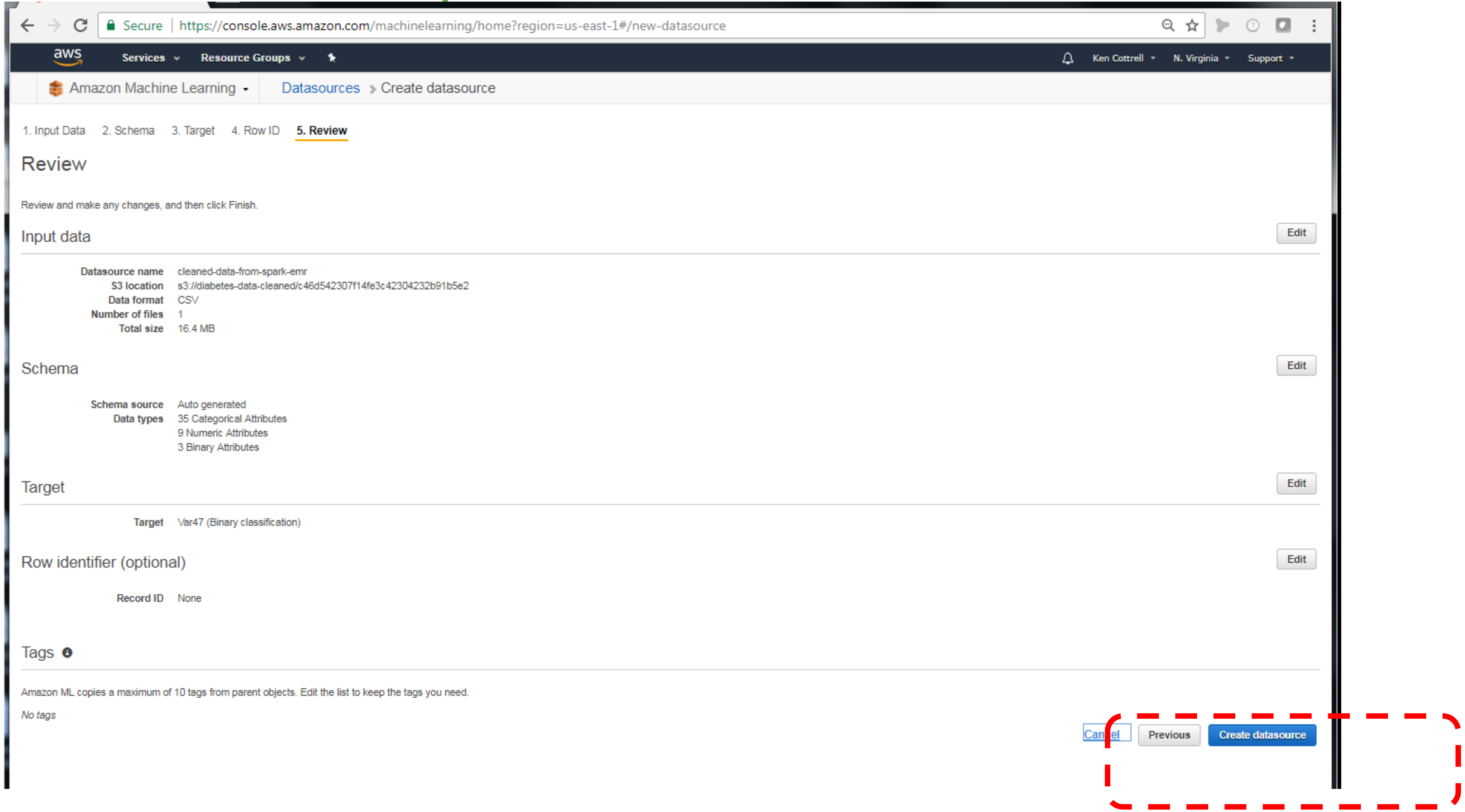
Do you plan to use this dataset to create or evaluate an ML model? ● Yes ○ No

Select the row containing the value you want to predict.

You have selected a binary attribute named *Var47* as the target. ML models trained on this target use logistic regression to train a binary classification model.

Target	Name	Data type	Sample field value 1	Sample field value 2	Sample field value 3
○	Var41	Categorical	glipizide-metformin	No	No
○	Var42	Categorical	glimepiride-pioglitazone	No	No
○	Var43	Categorical	metformin-rosiglitazone	No	No
○	Var44	Categorical	metformin-pioglitazone	No	No
○	Var45	Categorical	change	No	Ch
○	Var46	Binary	diabetesMed	No	Yes
●	Var47	Binary	readmitted	NO	NO

Review and complete datasource creation



Review

Review and make any changes, and then click Finish.

Input data

Datasource name	cleaned-data-from-spark-emr
S3 location	s3://diabetes-data-cleaned/c46d542307f14fe3c42304232b91b5e2
Data format	CSV
Number of files	1
Total size	16.4 MB

Schema

Schema source	Auto generated
Data types	35 Categorical Attributes 9 Numeric Attributes 3 Binary Attributes

Target

Target Var47 (Binary classification)

Row identifier (optional)

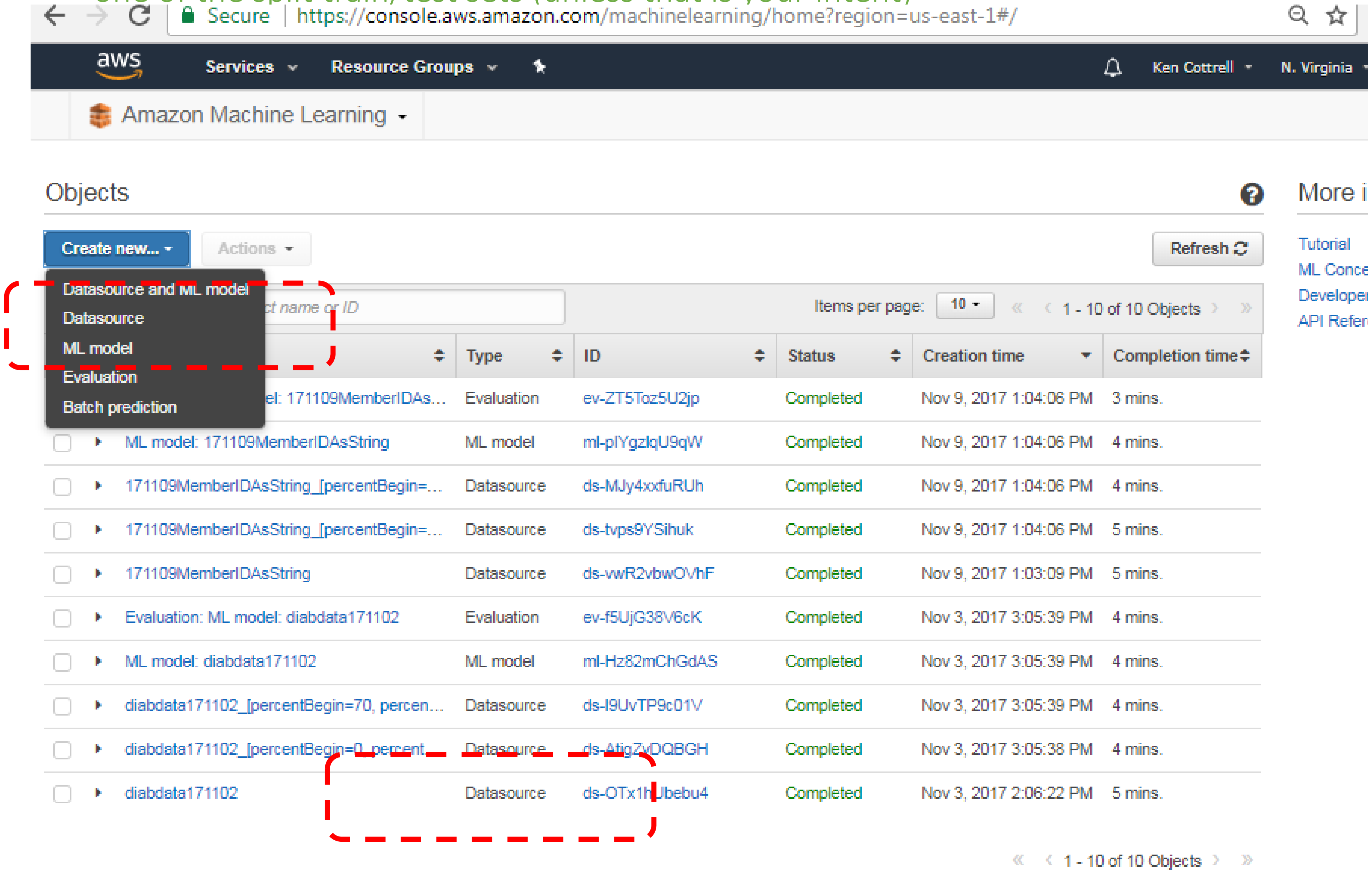
Record ID None

Tags

Amazon ML copies a maximum of 10 tags from parent objects. Edit the list to keep the tags you need.

No tags

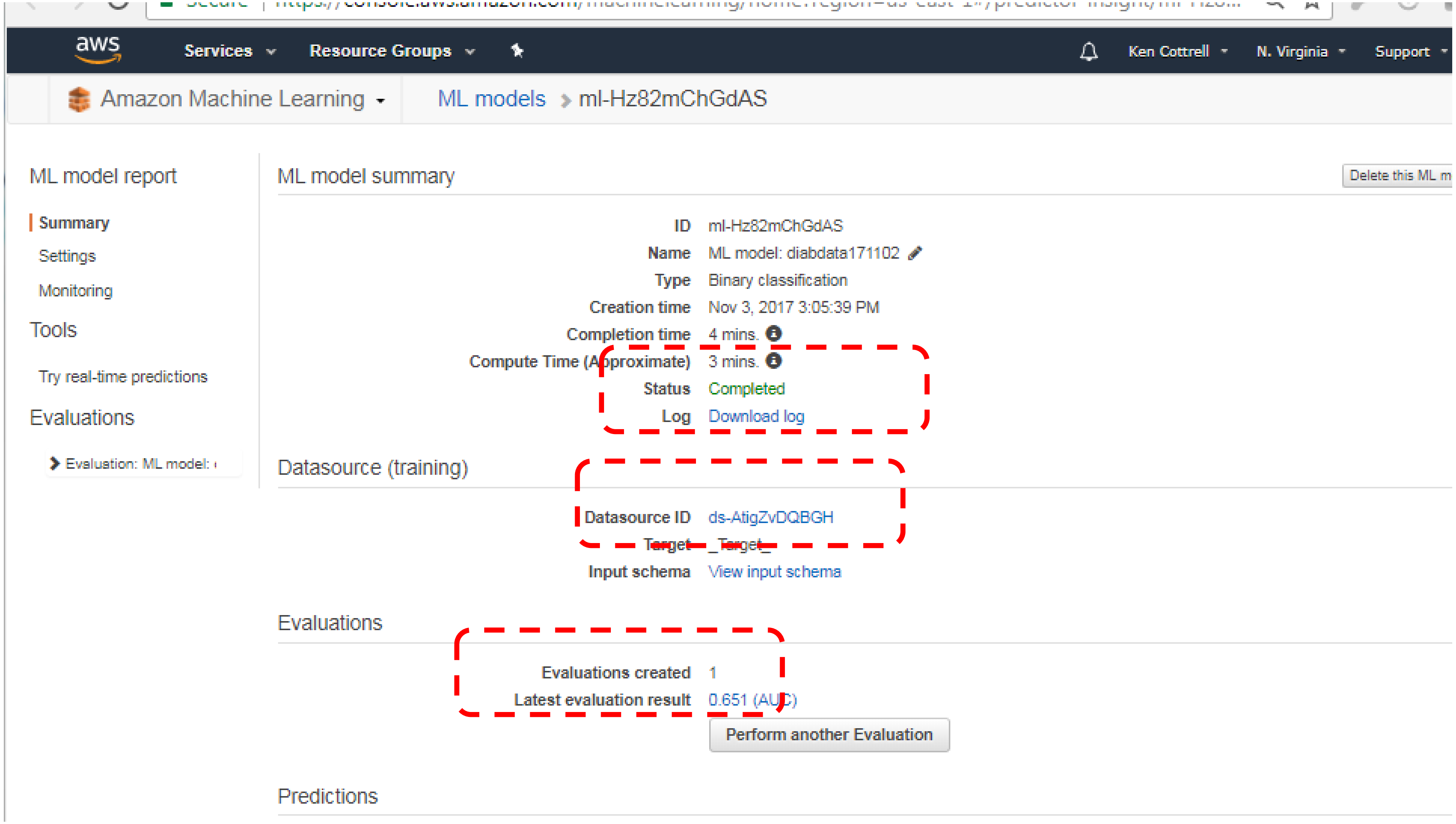
To create just the ML model, using an existing datasource - Make sure you don’t select one of the split train/test sets (unless that is your intent)



	Type	ID	Status	Creation time	Completion time
el: 171109MemberIDAs…	Evaluation	ev-ZT5Toz5U2jp	Completed	Nov 9, 2017 1:04:06 PM	3 mins.
ML model: 171109MemberIDAsString	ML model	ml-pIYgzlqU9qW	Completed	Nov 9, 2017 1:04:06 PM	4 mins.
171109MemberIDAsString_[percentBegin=…	Datasource	ds-MJy4xxfuRUh	Completed	Nov 9, 2017 1:04:06 PM	4 mins.
171109MemberIDAsString_[percentBegin=…	Datasource	ds-tvps9YSihuk	Completed	Nov 9, 2017 1:04:06 PM	5 mins.
171109MemberIDAsString	Datasource	ds-vwR2vbwOVhF	Completed	Nov 9, 2017 1:03:09 PM	5 mins.
Evaluation: ML model: diabdata171102	Evaluation	ev-f5UjG38V6cK	Completed	Nov 3, 2017 3:05:39 PM	4 mins.
ML model: diabdata171102	ML model	ml-Hz82mChGdAS	Completed	Nov 3, 2017 3:05:39 PM	4 mins.
diabdata171102_[percentBegin=70, percen…	Datasource	ds-I9UvTP9c01V	Completed	Nov 3, 2017 3:05:39 PM	4 mins.
diabdata171102_[percentBegin=0, percent…	Datasource	ds-AtigZvDQBGH	Completed	Nov 3, 2017 3:05:38 PM	4 mins.
diabdata171102	Datasource	ds-OTx1hUbebu4	Completed	Nov 3, 2017 2:06:22 PM	5 mins.

1 - 10 of 10 Objects

When you run the ML model, or ML Model and Evaluation (which does Train and test) – you may need to refresh screen to see if completed



ML model summary

ID	ml-Hz82mChGdAS
Name	ML model: diabdata171102
Type	Binary classification
Creation time	Nov 3, 2017 3:05:39 PM
Completion time	4 mins.
Compute Time (Approximate)	3 mins.
Status	Completed
Log	Download log

Datasource (training)

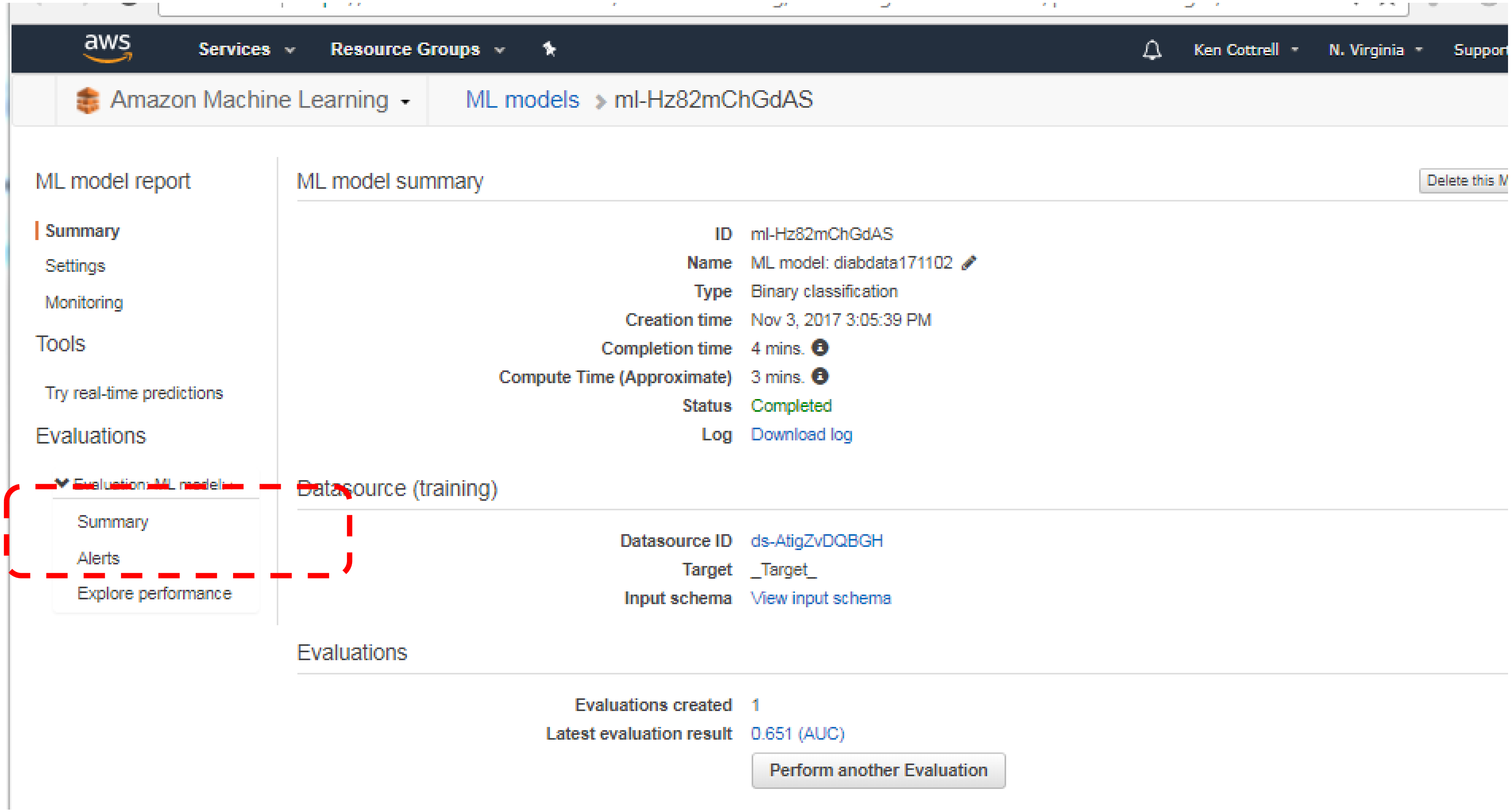
Datasource ID	ds-AtigZvDQBGH
Target	_Target_
Input schema	View input schema

Evaluations

Evaluations created	1
Latest evaluation result	0.651 (AUC)

Predictions

View the evaluation. This demo’s evaluation says our model is not much better than random (=50%) so considered a poor predictor



ML model summary

ID	ml-Hz82mChGdAS
Name	ML model: diabdata171102
Type	Binary classification
Creation time	Nov 3, 2017 3:05:39 PM
Completion time	4 mins.
Compute Time (Approximate)	3 mins.
Status	Completed
Log	Download log

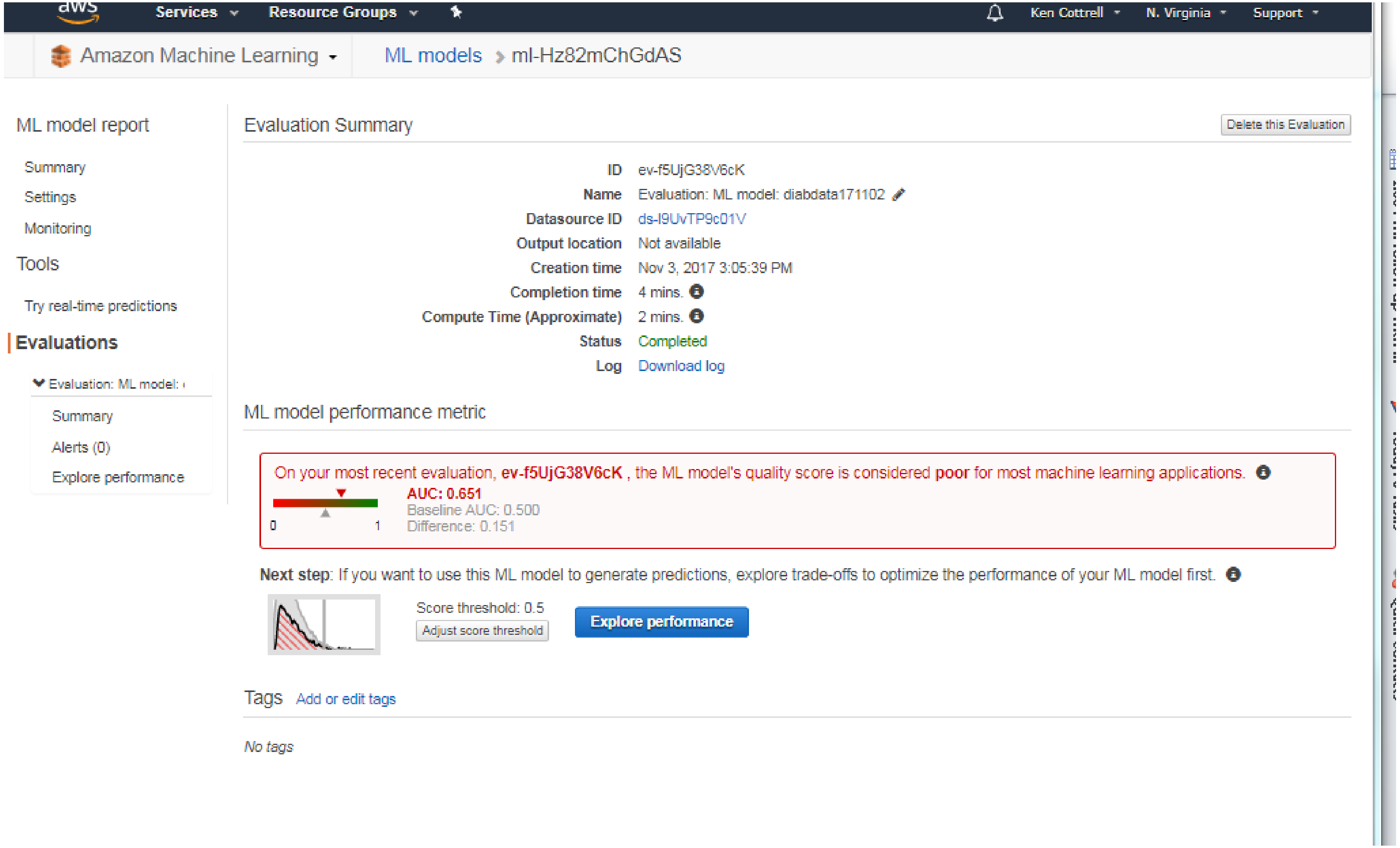
Datasource (training)

Datasource ID	ds-AtigZvDQBGH
Target	_Target_
Input schema	View input schema

Evaluations

Evaluations created	1
Latest evaluation result	0.651 (AUC)

Model is considered poor – reason is AUC of 6.5 isn't much better than random (.5). Should be greater than .9 to be considered a good model



Evaluation Summary

ID	ev-f5UjG38V6cK
Name	Evaluation: ML model: diabdata171102
Datasource ID	ds-I9UvTP9c01V
Output location	Not available
Creation time	Nov 3, 2017 3:05:39 PM
Completion time	4 mins.
Compute Time (Approximate)	2 mins.
Status	Completed
Log	Download log

ML model performance metric

On your most recent evaluation, **ev-f5UjG38V6cK** , the ML model's quality score is considered **poor** for most machine learning applications.

AUC: 0.651
Baseline AUC: 0.500
Difference: 0.151

Next step: If you want to use this ML model to generate predictions, explore trade-offs to optimize the performance of your ML model first.

Score threshold: 0.5

Tags

No tags

Appendix

Demo / tools discussion

- Why not use “traditional” ETL, BI, DW systems?
- Traditional centralized (server-based) tools can’t handle capacity and unstructured data
- Need new parallel elastic scaling approaches to handle Volume, Velocity, Variety of data
- Enforce schema layout / data model too soon in the data factory process
- BI and DW are still good for “finished good” data for specific use cases, but not as suitable for “work in process” data engineering and data science stages
- Better approach: *New elastic cloud architectures* coming to market, coupled with a rich ecosystem of data management and data science tools that run in cloud
- Intermediate data: Elastic Map Reduce, both batch-to-disk (Hadoop) and interactive in-memory (Apache Spark)
- Movement of traditional statistical languages like R and Python to a cloud parallel processing capability
- Machine Learning and AI are now able to leverage cloud capacity to automate many of the analytical workflows to calculate, compare, refine, and deploy the predictive models
- Pay only for temporary capacity needed to run huge jobs, then free it up when you get the insight needed

Demo / tools discussion

“push” data model

- Why not use “traditional” ETL, BI, DW systems?
- Traditional tools can’t handle capacity and unstructured data
- Need new parallel elastic scaling approaches to handle Volume, Velocity, Variety of data
- yout / data model too soon in the data factory process
- l good for “finished good” data for specific use cases, but not as suitable for “work ngineering and data science stages
- etter approach: *New elastic cloud architectures* available, coupled with a rich ecosystem of data management and data science tools that run in cloud
- Intermediate data: Elastic Map Reduce, both batch-to-disk (Hadoop) and interactive in-memory (Apache Spark)
- Movement of traditional statistical languages like R and Python to a cloud parallel processing capability
- Machine Learning and AI are now able to leverage cloud capacity to automate many of the analytical workflows to calculate, compare, refine, and deploy the predictive models
- Pay only for temporary capacity needed to run huge jobs, then free it up when you get the insight needed

“pull”, self-service data model for ad-hoc constituencies