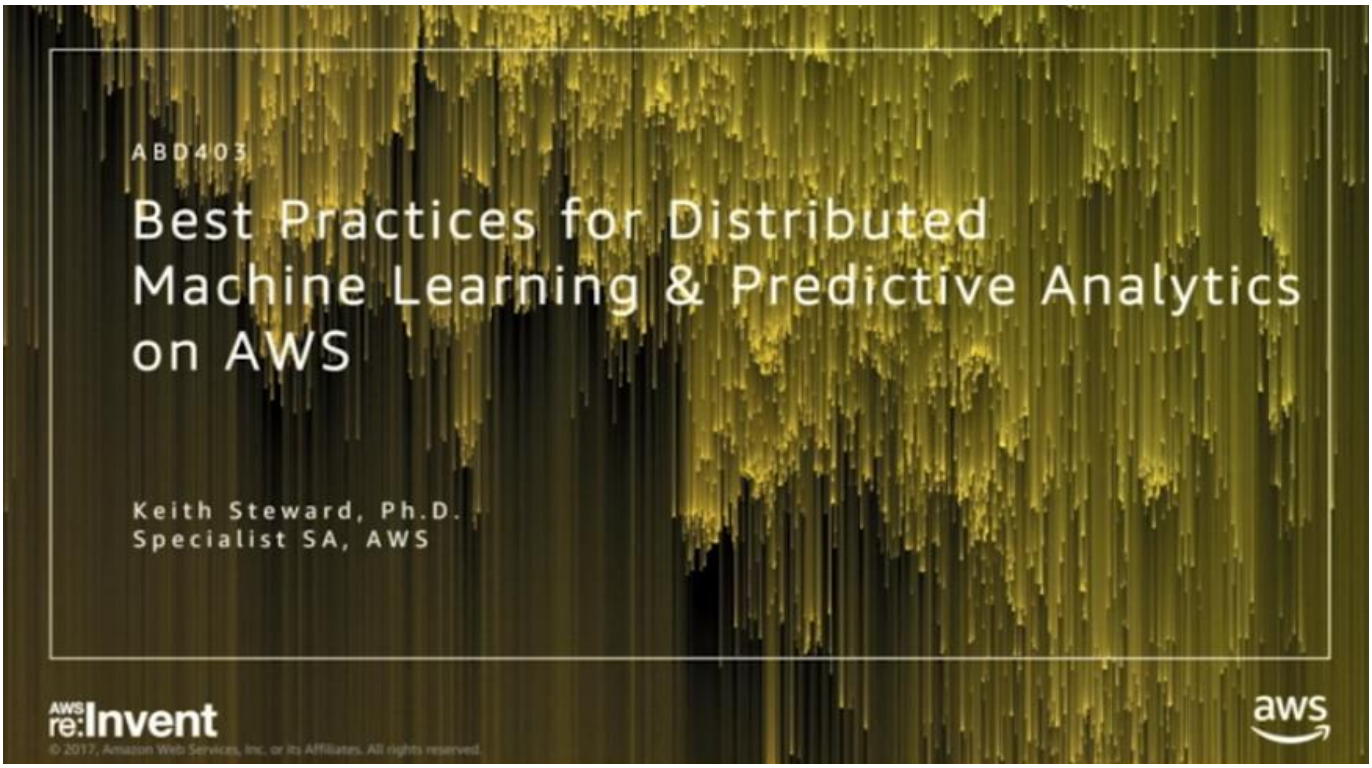
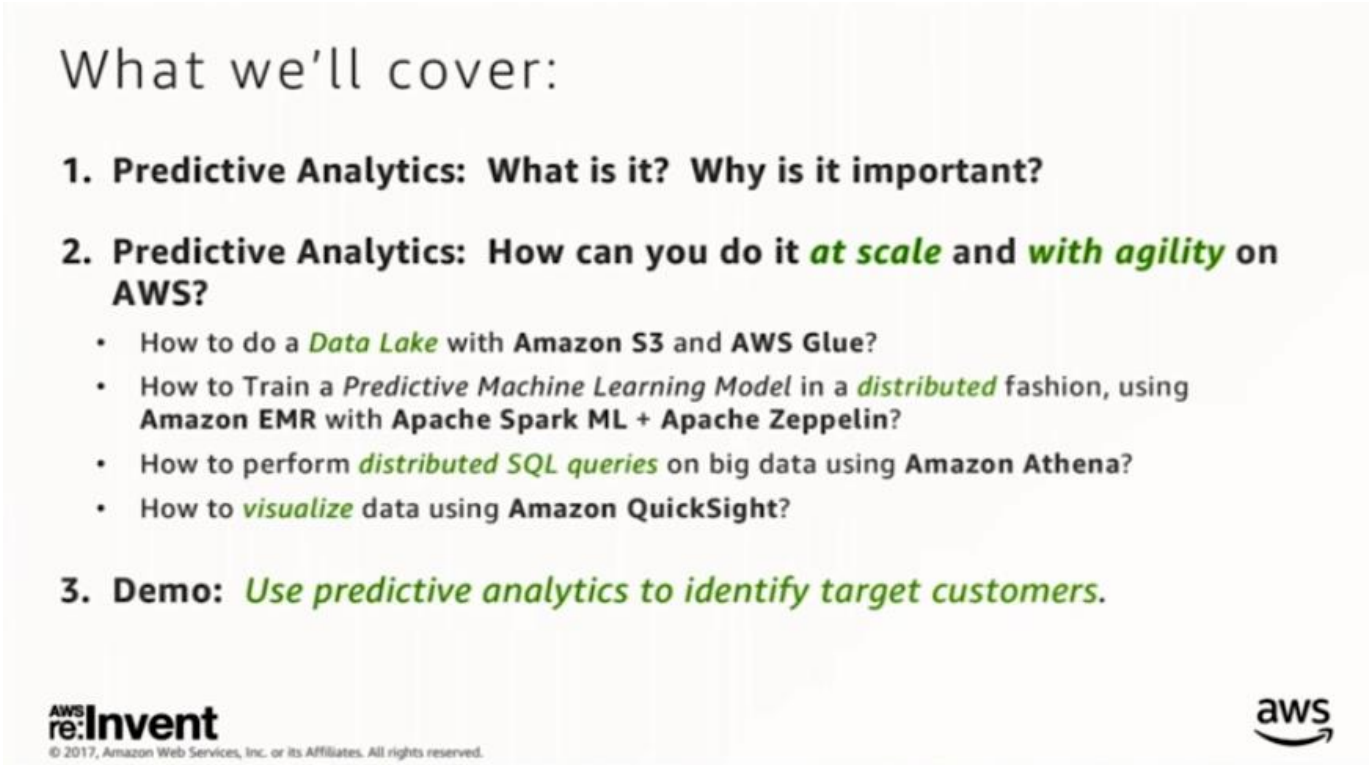



This session, we focus on common use cases and design patterns for predictive analytics using Amazon EMR. We address accessing data from a data lake, extraction and preprocessing with Apache Spark, analytics and machine learning code development with notebooks (Jupyter, Zeppelin), and data visualization using Amazon QuickSight. We cover other operational topics, such as deployment patterns for ad hoc exploration and batch workloads using Spot and multi-user notebooks. The intended audience for this session includes technical users who are building statistical and data analytics models for the business using tools, such as Python, R, Spark, Presto, Amazon EMR, Notebooks

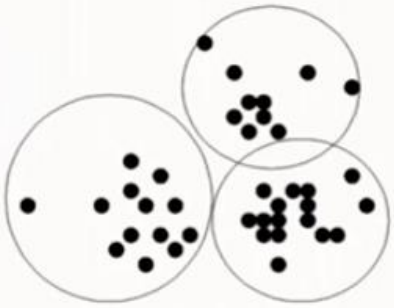


# 1. Predictive Analytics:

What is it?

Why is it important?

## Wikipedia Definitions:



**Analytics**: the discovery, interpretation, and communication of meaningful patterns in data.

**Predictive Analytics:** encompasses a variety of statistical techniques from predictive modeling, machine learning, and data mining that analyze current and historical facts to make predictions about future or otherwise unknown events.

## Predictive Analytics Use Cases:


**Customer predictions**

- *"Which customers are likely to be the most profitable?"*
- *"How much revenue should I expect this customer to generate?"*
- *"Which customers are likely to churn?"*
- *"Among all of our customers, which are likely to respond to a given offer?"*
- *"What's the probability that a given customer X will respond to a given offer Y"*

**Product or Service predictions**

- *"What products should we offer or develop?"*
- *"What items are likely to be purchased together?"* (basket analysis)

**Business Operations predictions**

- *"Are the metrics for service X nominal or anomalous?"*
- *"Is equipment X likely to fail within time Y?"*

# Why is Predictive Analytics Important?

- Companies have been accumulating "Big Data" about customers, products/services, and operations for many years now.
- Big Data technologies have provided proven solutions to address this *data management* need.
- But… Increasing pressure to turn data into insights about trends, classifications, detect anomalies, and provide feedback loops to improve their business

# Why is Predictive Analytics Important?

- Desire to evolve from backwards-looking monthly or quarterly reports to real-time alerts, and now to predicting the future

Business' Evolving Needs for Data



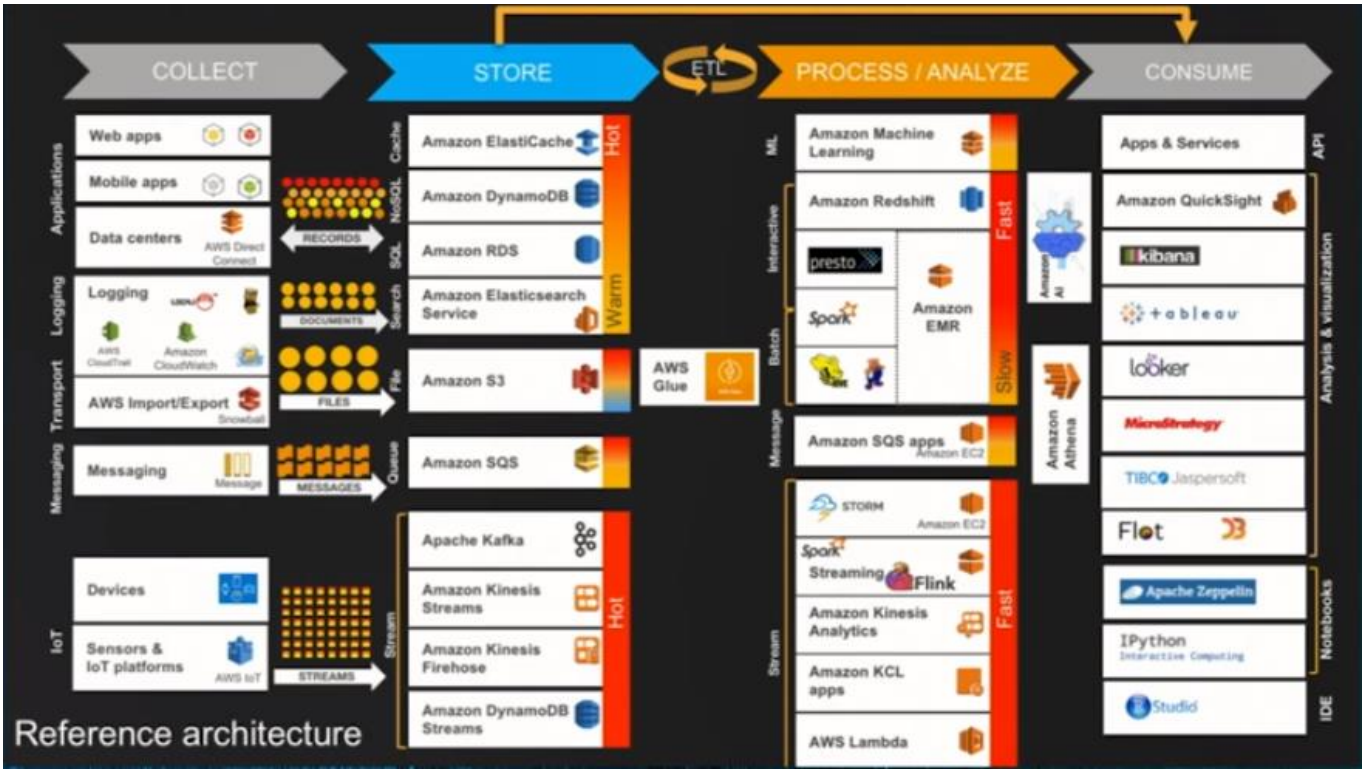


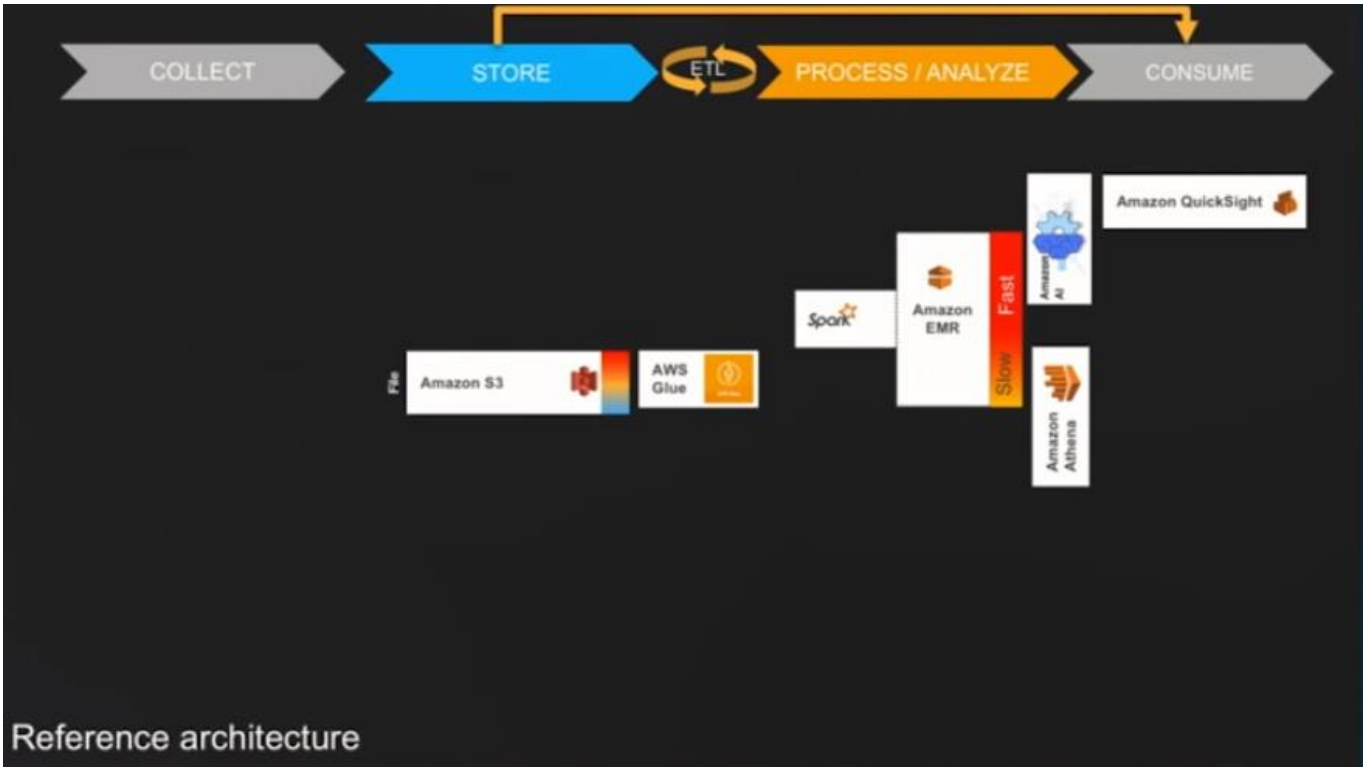
aws

## 2. Predictive Analytics: How can you do it at *scale* and with *agility* on AWS?

# Predictive Analytics Architecture


# Plethora of Tools


COLLECT
STORE
ETL
PROCESS / ANALYZE
CONSUME
Applications
Web apps
Mobile apps
Data centers
AWS Direct Connect
RECORDS
Logging
Logging
AWS CloudTrail
Amazon CloudWatch
DOCUMENTS
Transport
AWS Import/Export Snowball
FILES
Messaging
Messaging
Message
MESSAGES
IoT
Devices
Sensors & IoT platforms
AWS IoT
STREAMS
Cache
NoSQL
SQL
Search
File
Queue
Stream
Amazon ElastiCache
Amazon DynamoDB
Amazon RDS
Amazon Elasticsearch Service
Amazon S3
Amazon SQS
Apache Kafka
Amazon Kinesis Streams
Amazon Kinesis Firehose
Amazon DynamoDB Streams
Hot
Warm
Hot
AWS Glue
ML
Amazon Machine Learning
Interactive
Amazon Redshift
presto
Batch
Spark
Amazon EMR
Fast
Slow
Amazon AI
Amazon Athena
Message
Amazon SQS apps
Amazon EC2
Stream
STORM
Amazon EC2
Spark Streaming
Flink
Amazon Kinesis Analytics
Amazon KCL apps
AWS Lambda
Fast
Apps & Services
API
Amazon QuickSight
kibana
tableau
looker
MicroStrategy
TIBCO Jaspersoft
Flot
Analysis & visualization
Apache Zeppelin
IPython
Interactive Computing
Notebooks
Studio
IDE
Reference architecture


COLLECT
STORE
ETL
PROCESS / ANALYZE
CONSUME
File
Amazon S3
AWS Glue
Spark
Amazon EMR
Fast
Slow
Amazon AI
Amazon Athena
Amazon QuickSight
Reference architecture

Amazon S3

**Store anything (object storage)**

**Scalable / Elastic**

**99.999999999% durability**

**Effectively infinite inbound bandwidth**

**Extremely low cost:** $0.023/GB-Mo; $23/TB-Mo

**Data layer for virtually all AWS services**

# Why Amazon S3 for data lake?

**Durable**

Designed for 11 9s of durability

**Available**

Designed for **99.99%** availability

**High performance**

- Multiple upload
- Range GET

**Easy to use**

- Simple REST API
- AWS SDKs
- Read-after-create consistency
- Event notification
- Lifecycle policies

**Scalable**

- Store as much as you need
- Scale storage and compute independently
- No minimum usage commitments

**Integrated**

- Amazon EMR
- Amazon Redshift
- Amazon DynamoDB

AWS Glue

**Data Catalog**

**Managed ETL Engine**

**Job Scheduler**

**Built on Apache Spark**

**Integrated with S3, RDS, Redshift**

**Integrates with any JDBC data store**

QuickSight

## Glue data catalog

Discover and organize your data sets

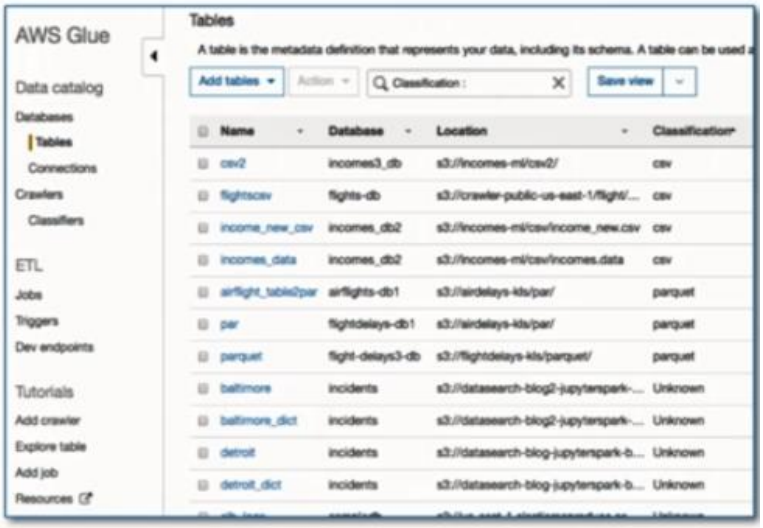
## Glue data catalog

Manage **table metadata** through a **Hive metastore API** or **Hive SQL**. Supported by tools such as Hive, Presto, Spark, etc.

We added a few extensions:

- **Search** metadata for data discovery
- **Connection info** – JDBC URLs, credentials
- **Classification** for identifying and parsing files
- **Versioning** of table metadata as schemas evolve and other metadata are updated

Populate using Hive DDL, bulk import, or automatically through **crawlers**.






## Crawlers: auto-populate data catalog

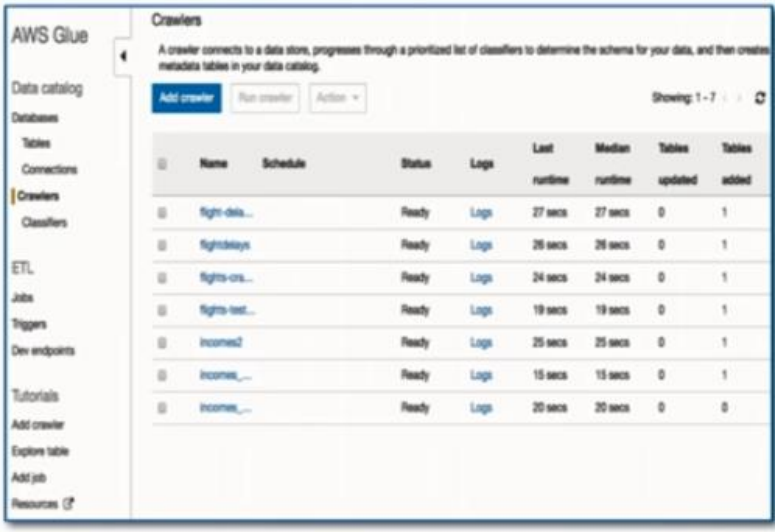
Automatic schema inference:

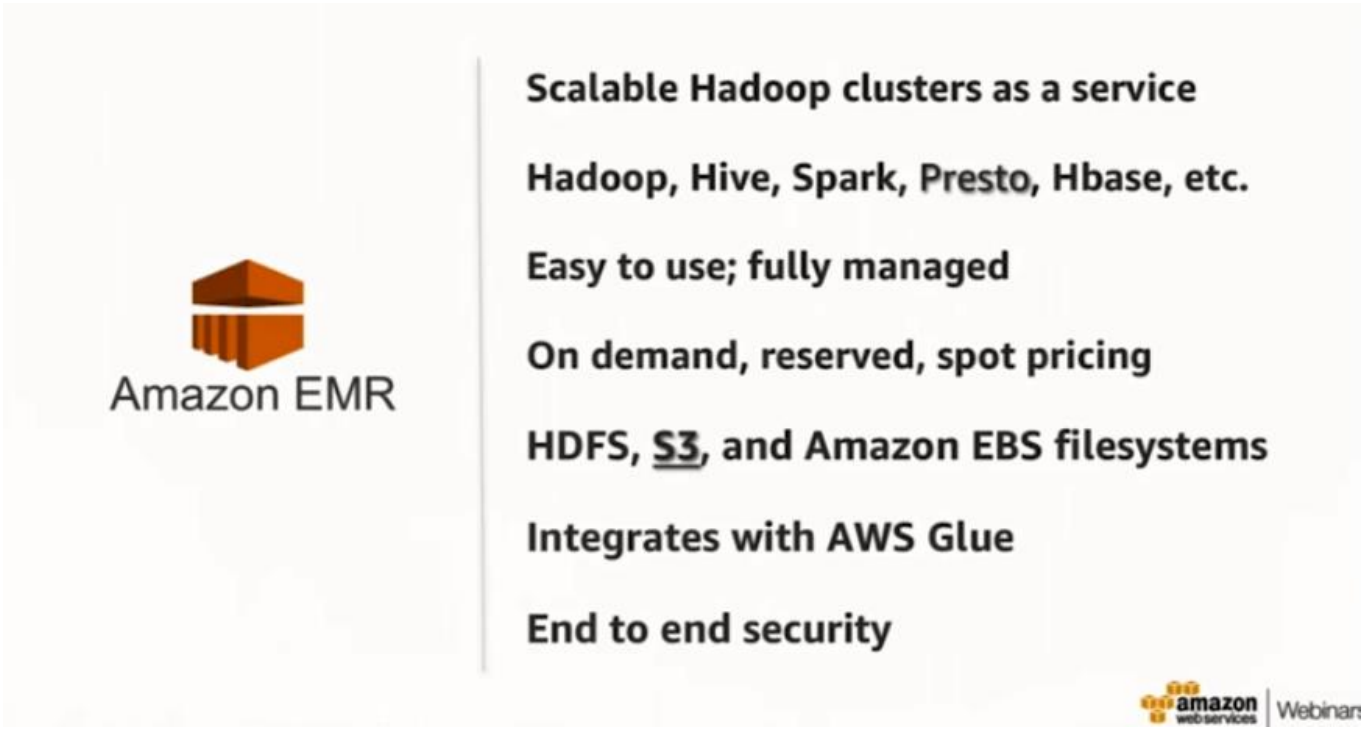
- Built-in **classifiers** detect file type and **extract schema:** record structure and data types.
- Add your own or share with others in the Glue community - It's all Grok and Python.

Auto-detects Hive-style partitions, grouping similar files into one table.

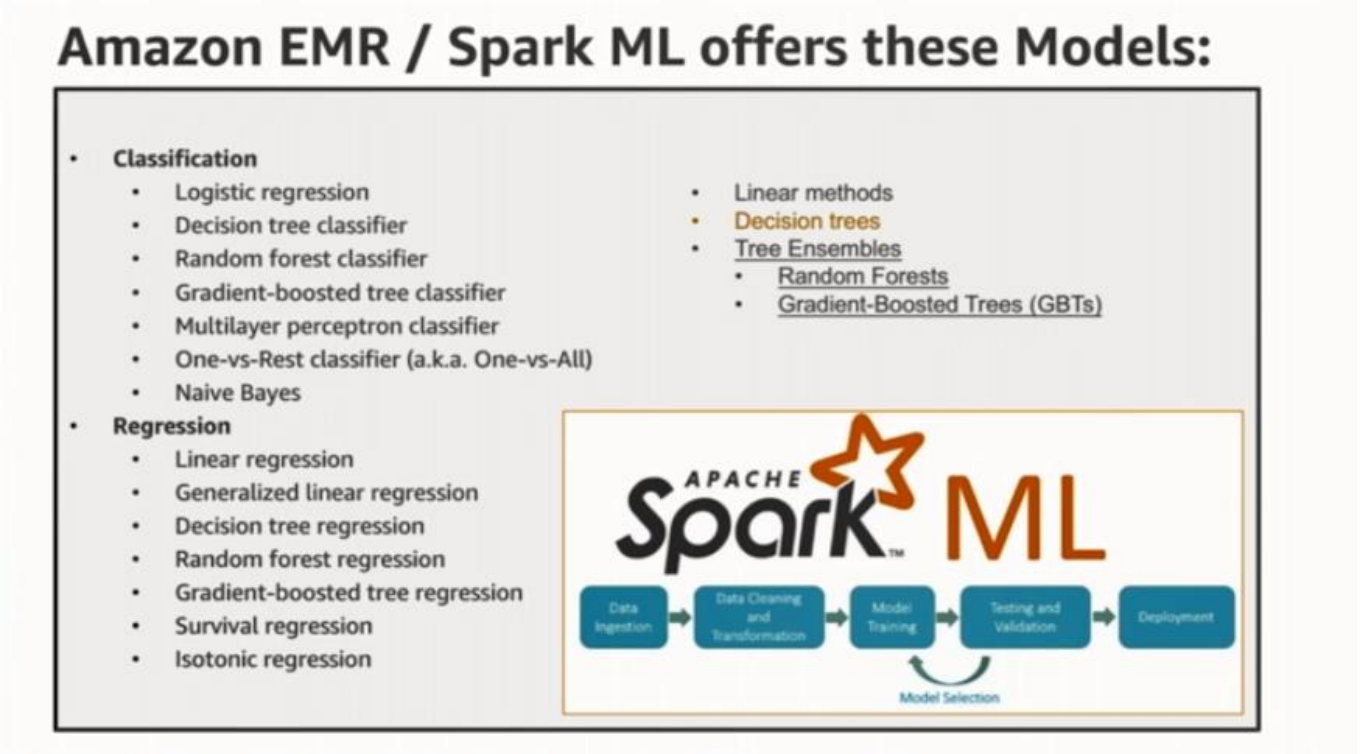
Run crawlers on schedule to **discover** new data and **schema changes.**

Serverless – only pay when crawls run.

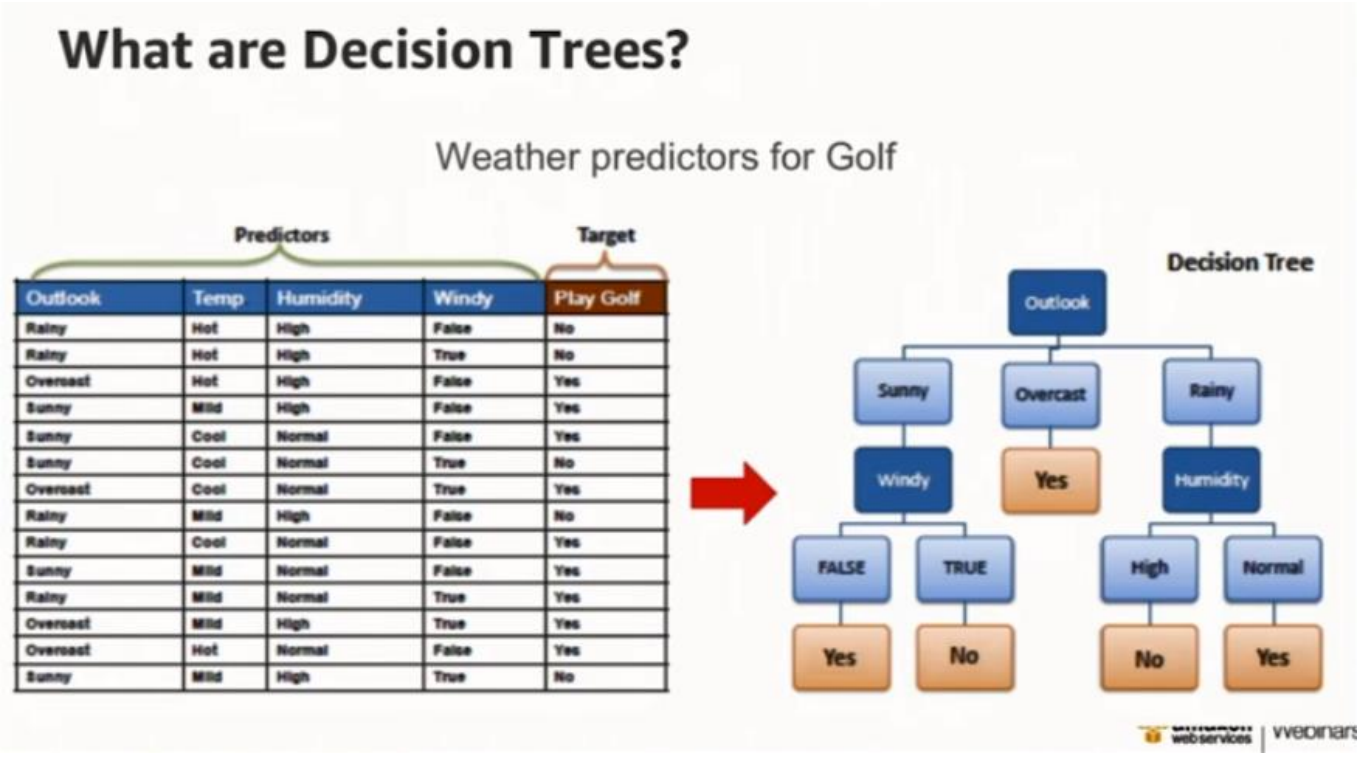



Amazon EMR
Scalable Hadoop clusters as a service
Hadoop, Hive, Spark, Presto, Hbase, etc.
Easy to use; fully managed
On demand, reserved, spot pricing
HDFS, S3, and Amazon EBS filesystems
Integrates with AWS Glue
End to end security
amazon web services | Webinars


EMR: Just some of the organizations using
The New York Times
Adobe
NASDAQ OMX
airbnb
slack
ROVIO
AMGEN
DataXu
AUTODESK.
BrightRoll
washingtonpost.com
FINRA
has offers
bloomreach
THE CLIMATE CORPORATION
amazon.com.
yelp
Expedia
4
NASA JPL
NOKIA
razorfish.
AdRoll
NETFLIX
MediaMath
gumgum
reddit
HEARST digital media
Swipely
amazon web services | Webinars

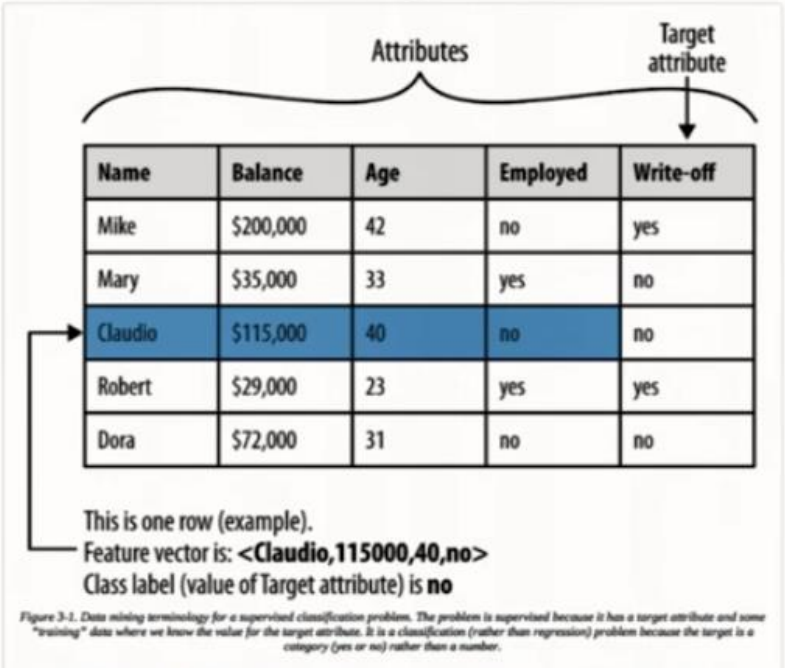



***SparkML*** allows us to implement a pipeline using data frame transformations, we can persist the pipeline and reuse models that were earlier trained




It builds a decision tree in-memory for the prediction problem

# Decision Trees: Training Data




| Name | Balance | Age | Employed | Write-off |
|---|---|---|---|---|
| Mike | $200,000 | 42 | no | yes |
| Mary | $35,000 | 33 | yes | no |
| Claudio | $115,000 | 40 | no | no |
| Robert | $29,000 | 23 | yes | yes |
| Dora | $72,000 | 31 | no | no |

*Figure 3-1. Data mining terminology for a supervised classification problem. The problem is supervised because it has a target attribute and some "training" data where we know the value for the target attribute. It is a classification (rather than regression) problem because the target is a category (yes or no) rather than a number.*

Bank loan write-off predictions

Amazon QuickSight

**Fast Business Analytics / Data Visualization Service running in AWS**

**Access multiple AWS data sources (S3, Athena, Aurora, Redshift, etc) or data you upload**

**Perform ad hoc analyses to gain insights from your data**

**Supports 100's of thousands of users**

**Smart visualizations dynamically optimized for your data**

**Generate fast, interactive visualizations on large datasets**


Amazon QuickSight is a Business Analytics Service that lets business users quickly and easily visualize, explore, and share insights from their data.

QuickSight

amazon web services | Webinars

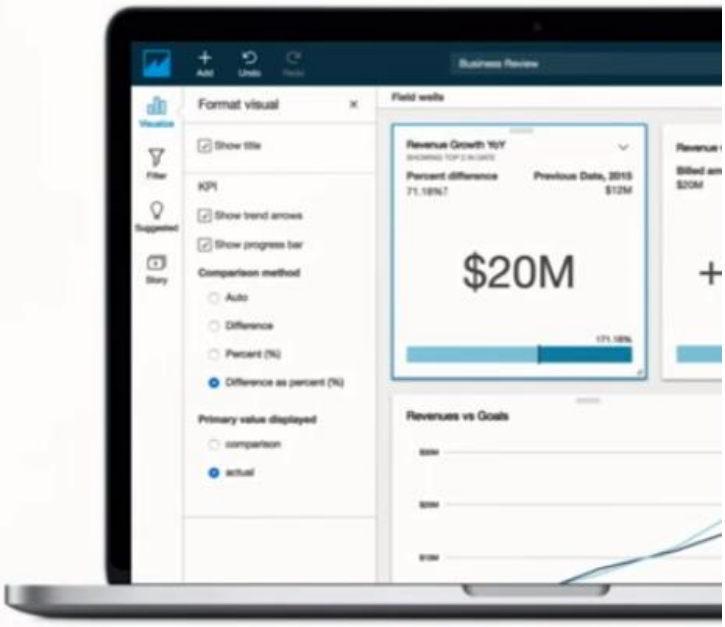
## QuickSight Overview

**Integrated with AWS**
Redshift, RDS, Athena, S3, IAM, Roles, etc…

**Cloud Native**
Fully managed, serverless analytics at scale

**Super Fast and Easy to Use**
Backed by SPICE and a beautiful UI

**Cost Effective**
Starts at $9 per user per month

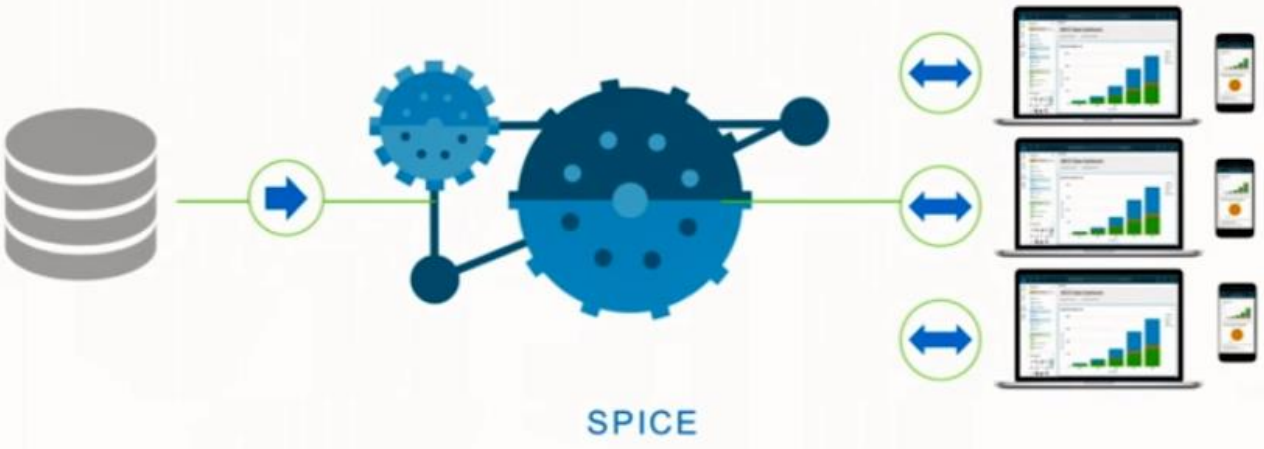


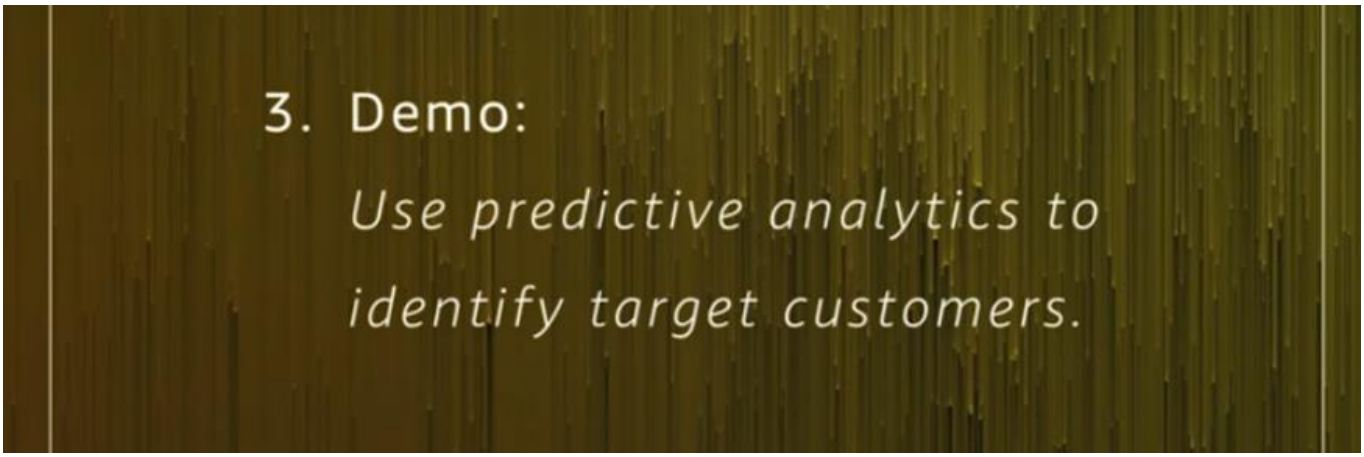
**Deep Integration with AWS Data Sources**

QuickSight is **deeply integrated** with AWS data sources like Redshift, RDS, S3, Athena and others, as well as third-party sources like Excel, Salesforce, as well as on-premises databases.


## SPICE

QuickSight is powered by SPICE, a super-fast calculation engine that delivers unprecedented performance and scale, delivering insights at the speed of thought.






## Our Scenario: We are LuxCars, Inc.

### The Challenge:

- ❑ We're bringing an expensive new luxury car (the "*Luxmobile*") to market.
- ❑ Want to focus marketing on customers likely to be *wealthy* enough to afford our luxury automobile.
- ❑ Have demographic data for our target geography, but we don't have income data.
- ❑ Need to *predict* salary from the demographic data.

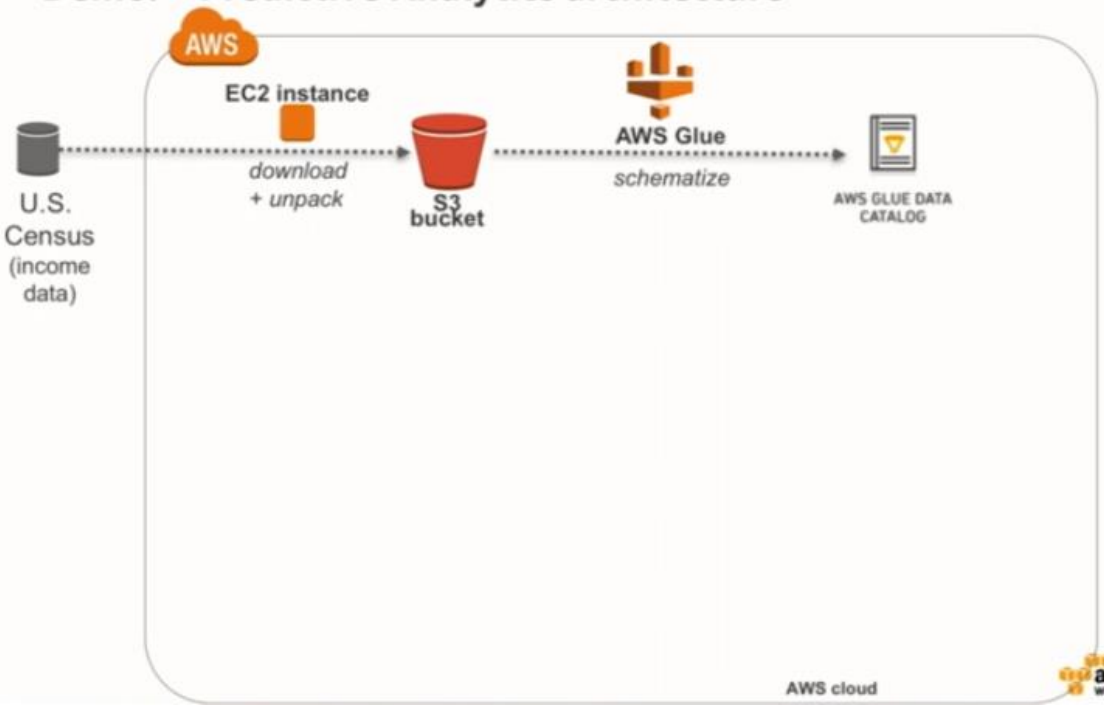
# Our Scenario: We are LuxCars Inc.

## Strategy:

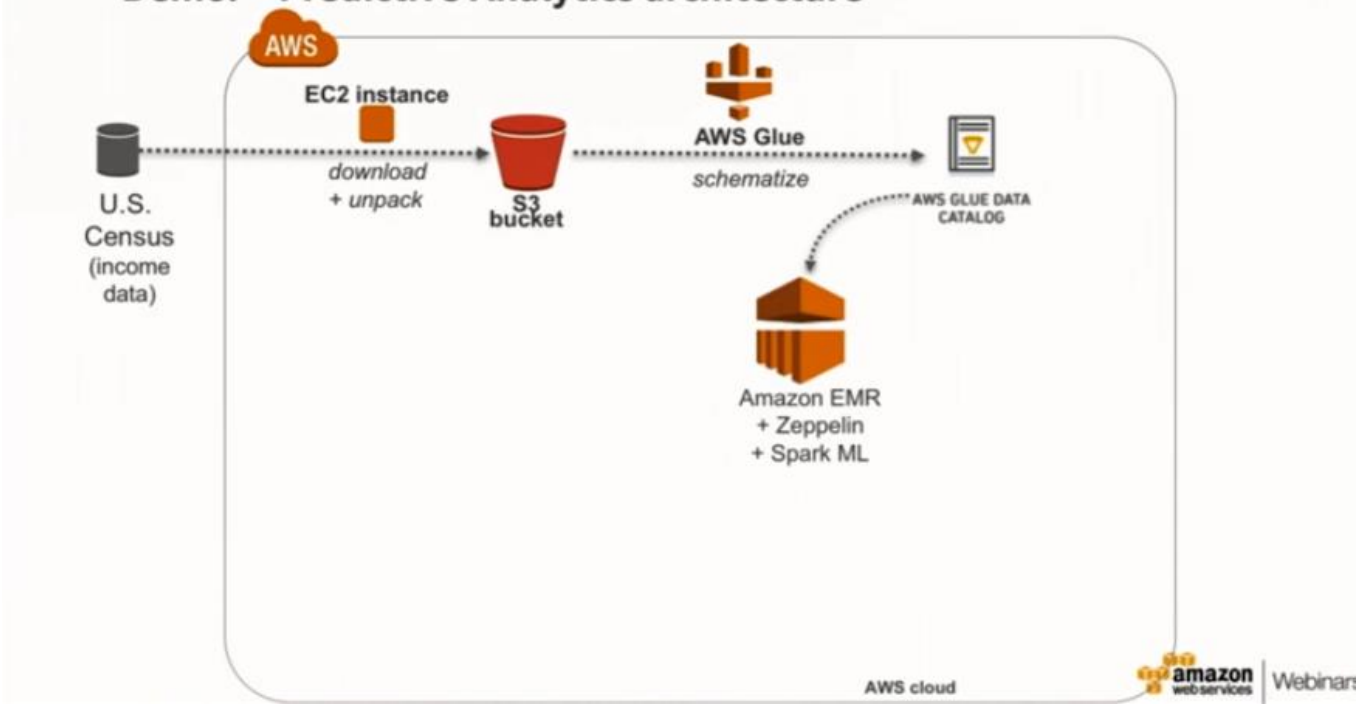
1. Collect U.S. Census Income data & put into the Cloud.
2. *Schematize* the data.
3. Train a Spark ML model on the data.
4. Do batch predictions of income for people in the target geography.
5. Compute mean incomes per sub-geography ("income density")
6. Visualize those predictions.
7. Share results with our Marketing Director.



## Demo: Predictive Analytics architecture

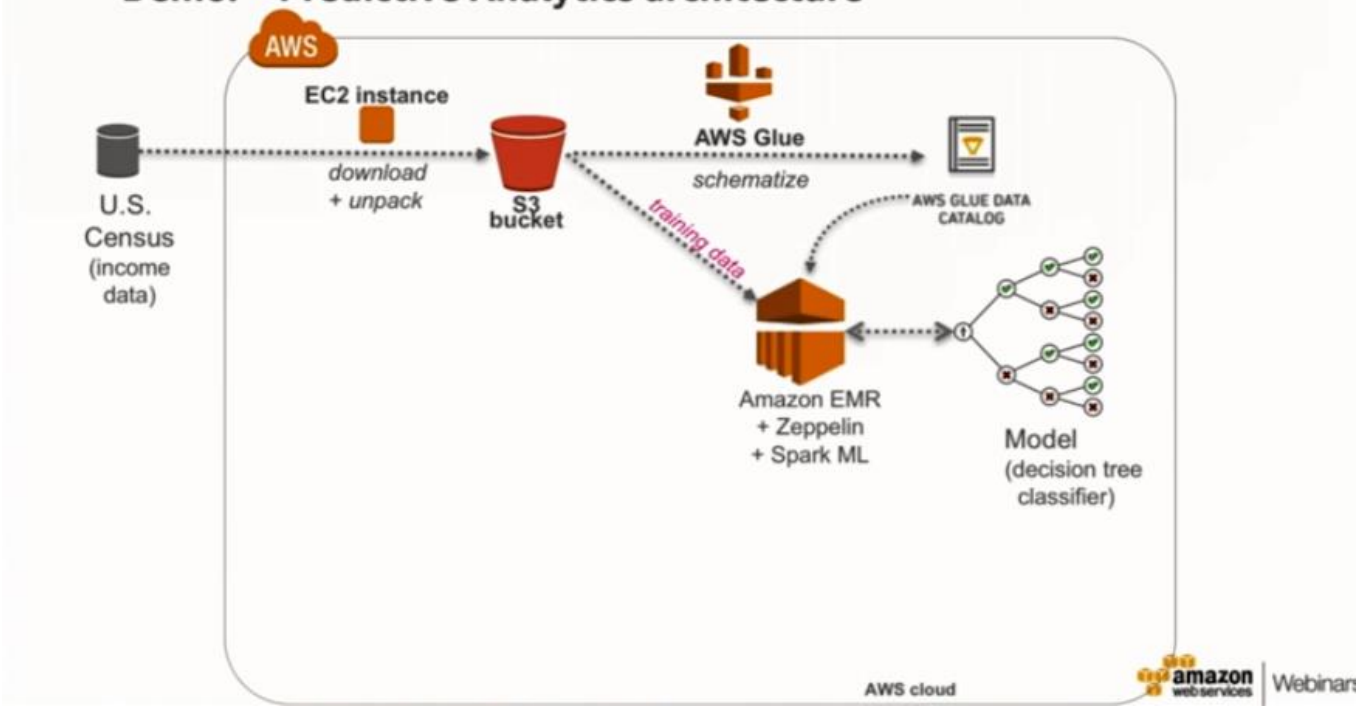


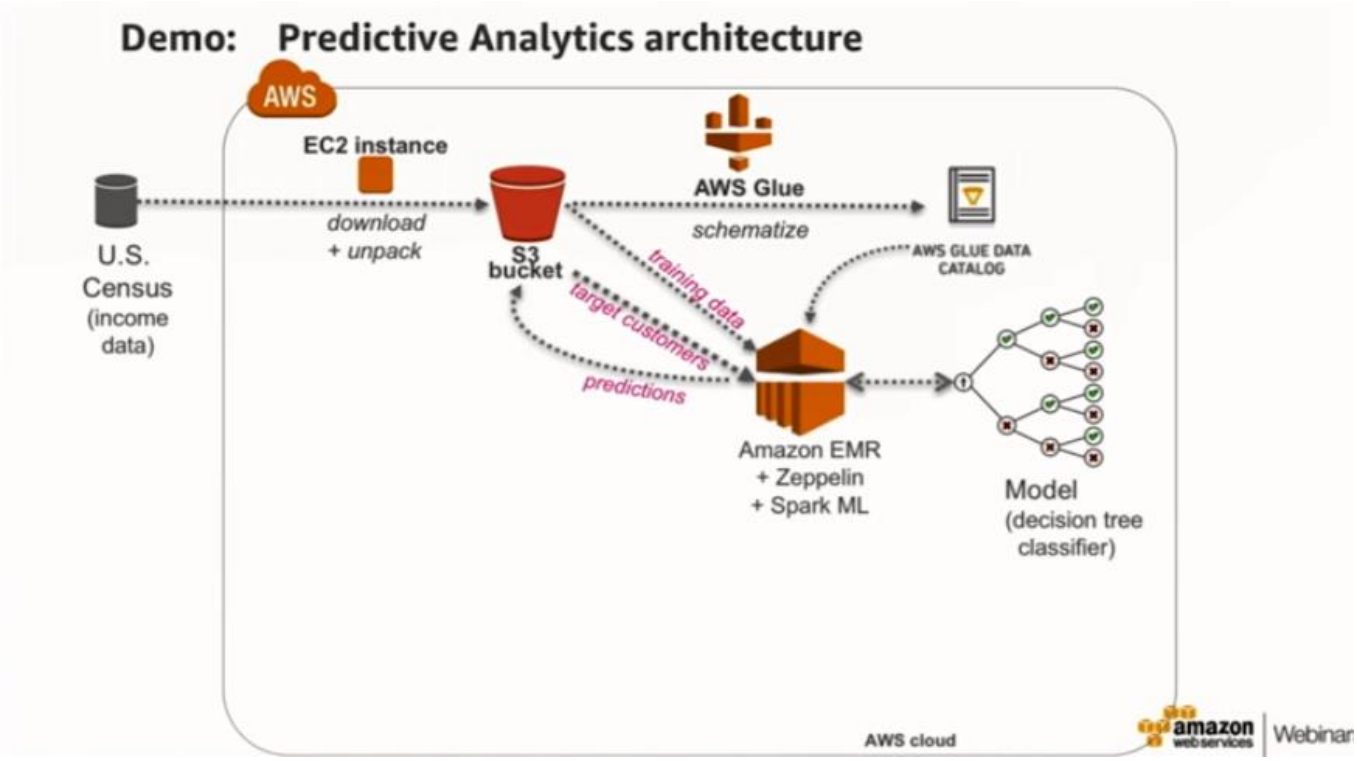
## Demo: Predictive Analytics architecture



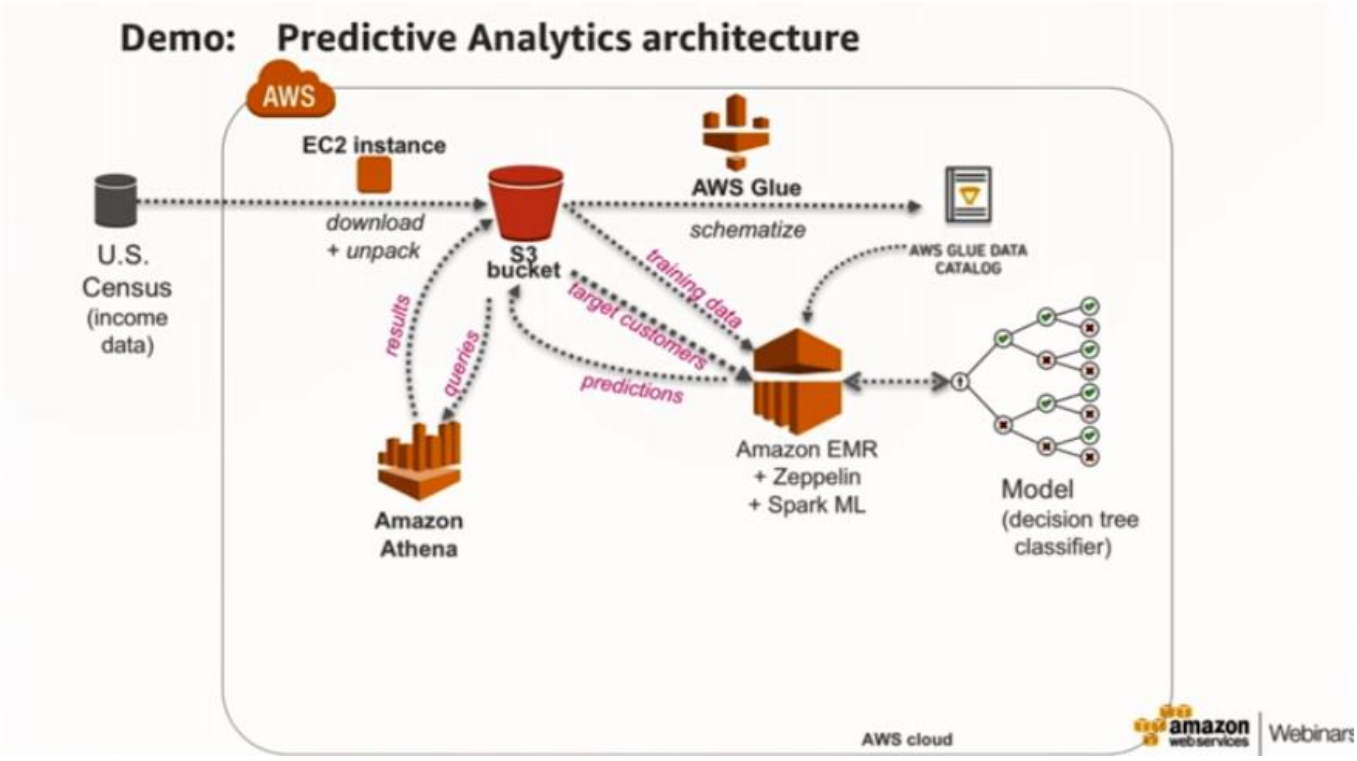



## Demo: Predictive Analytics architecture

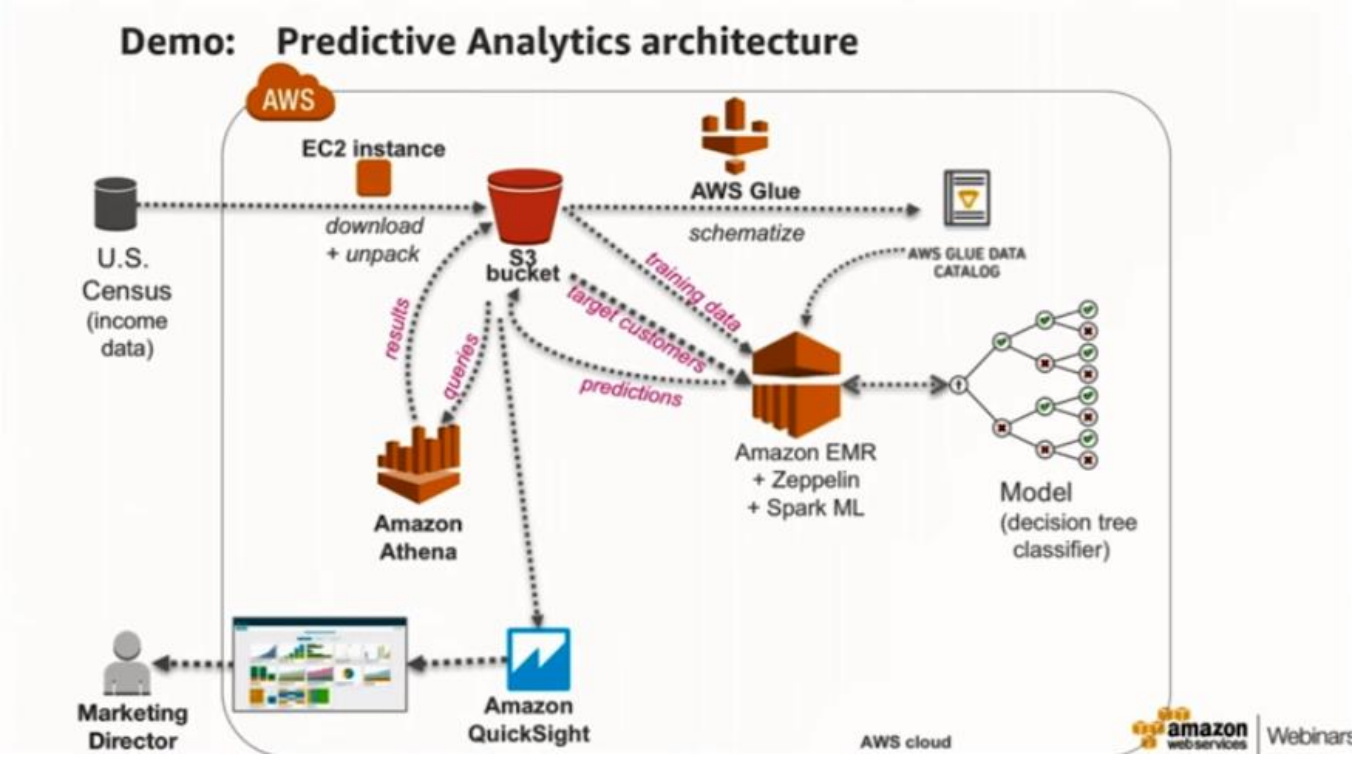




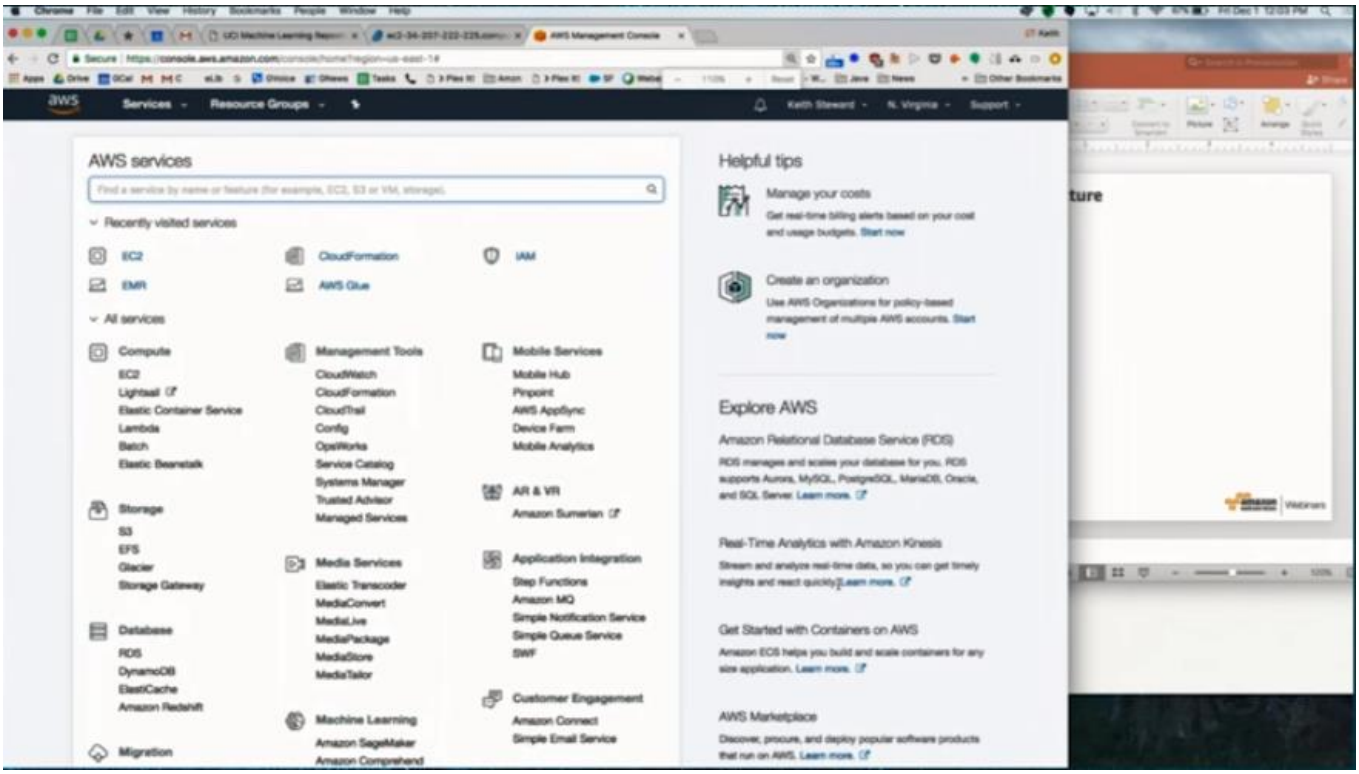

We then train a decision tree using ***SparkML***, then we will take customers that we don't know their incomes and use that dataset to make predictions and store the results

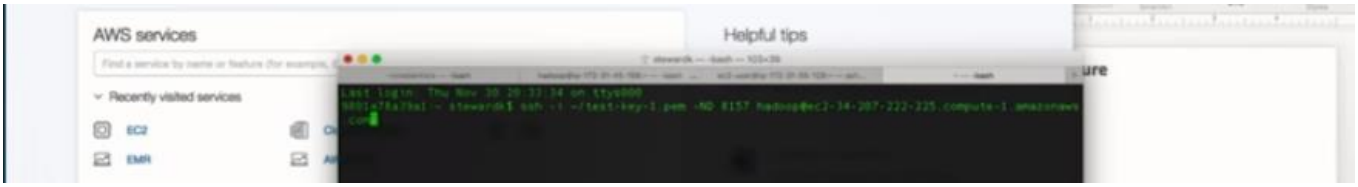



We then query the predictions data with Athena and store the query results

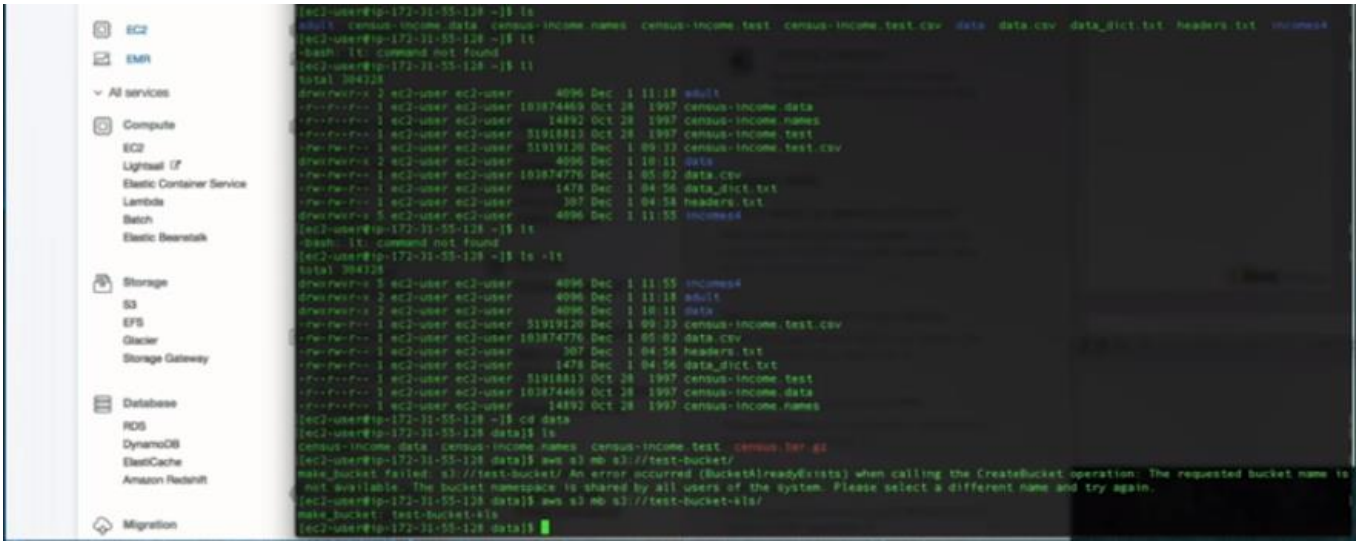



Then we can visualize our query results and create reports to share

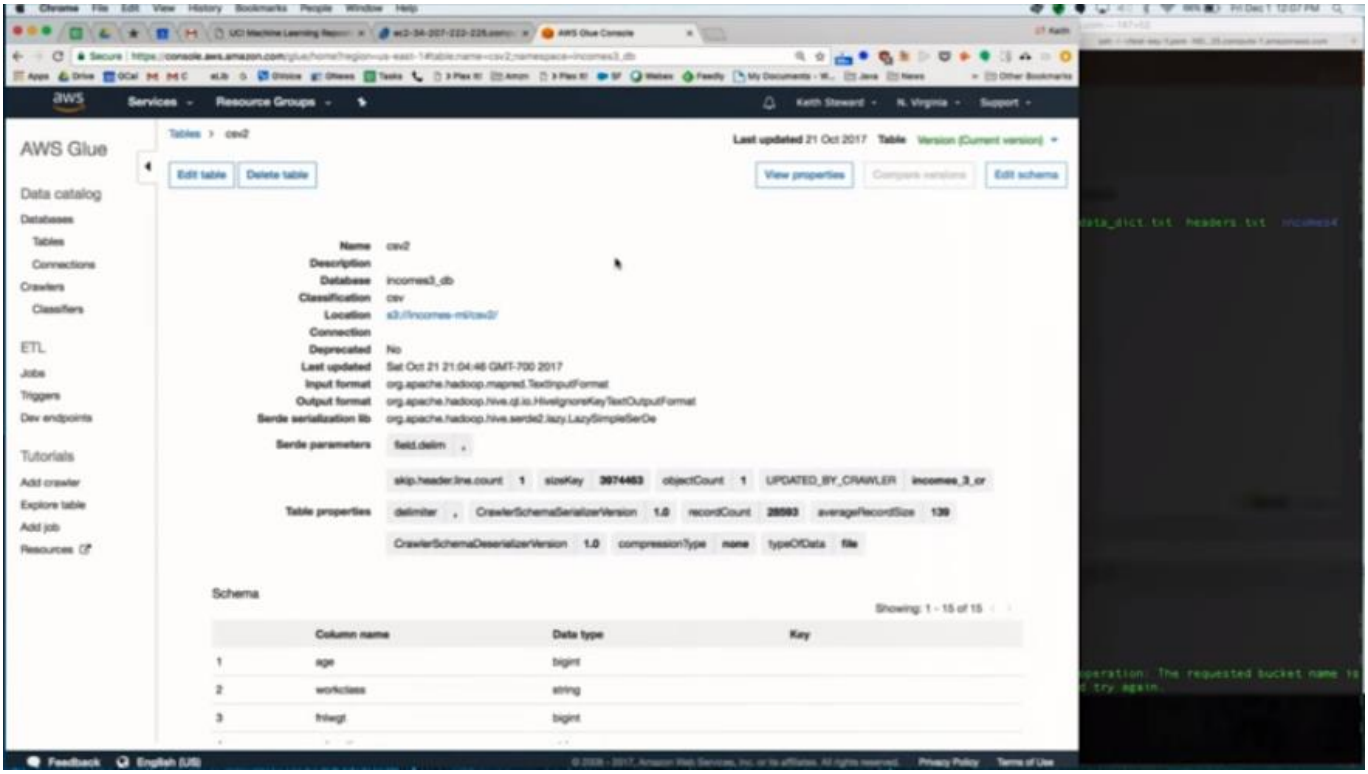


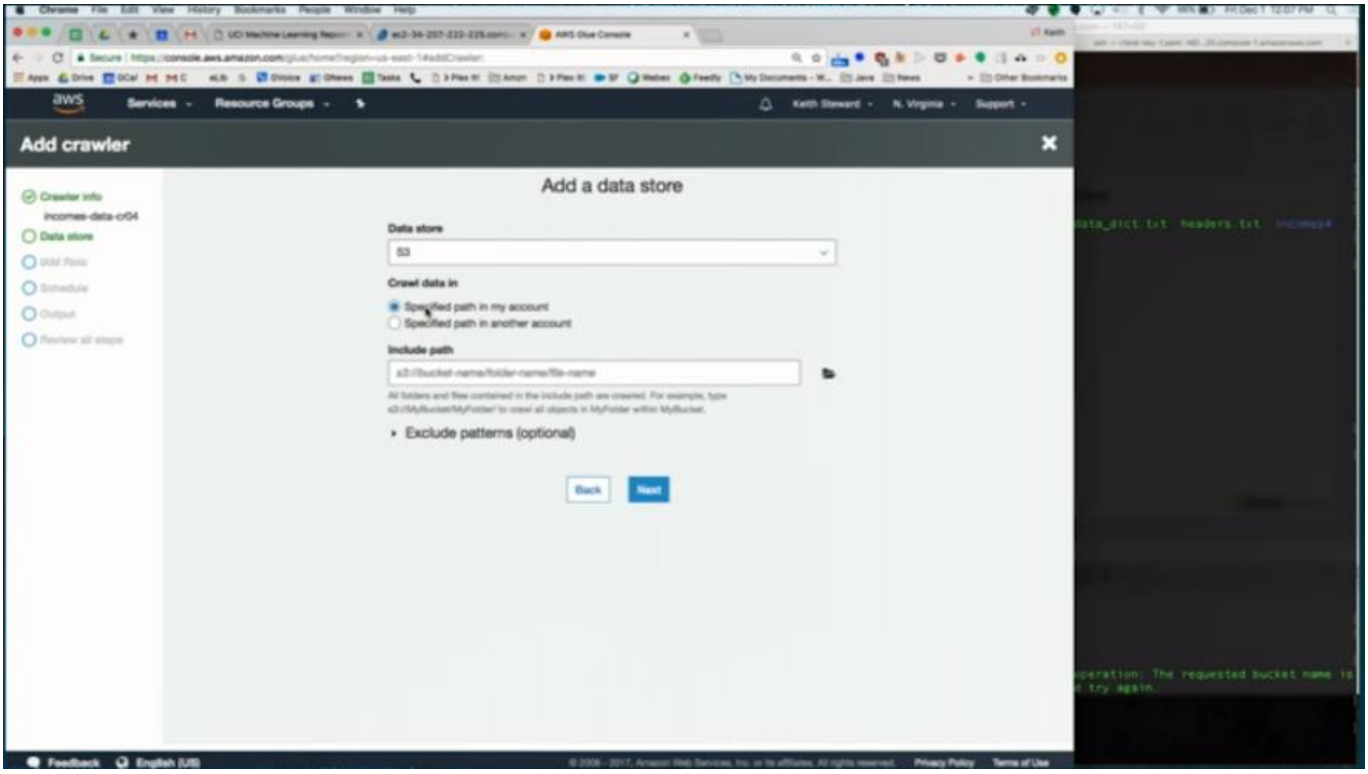

Spin up an EC2 instance an SSH into it

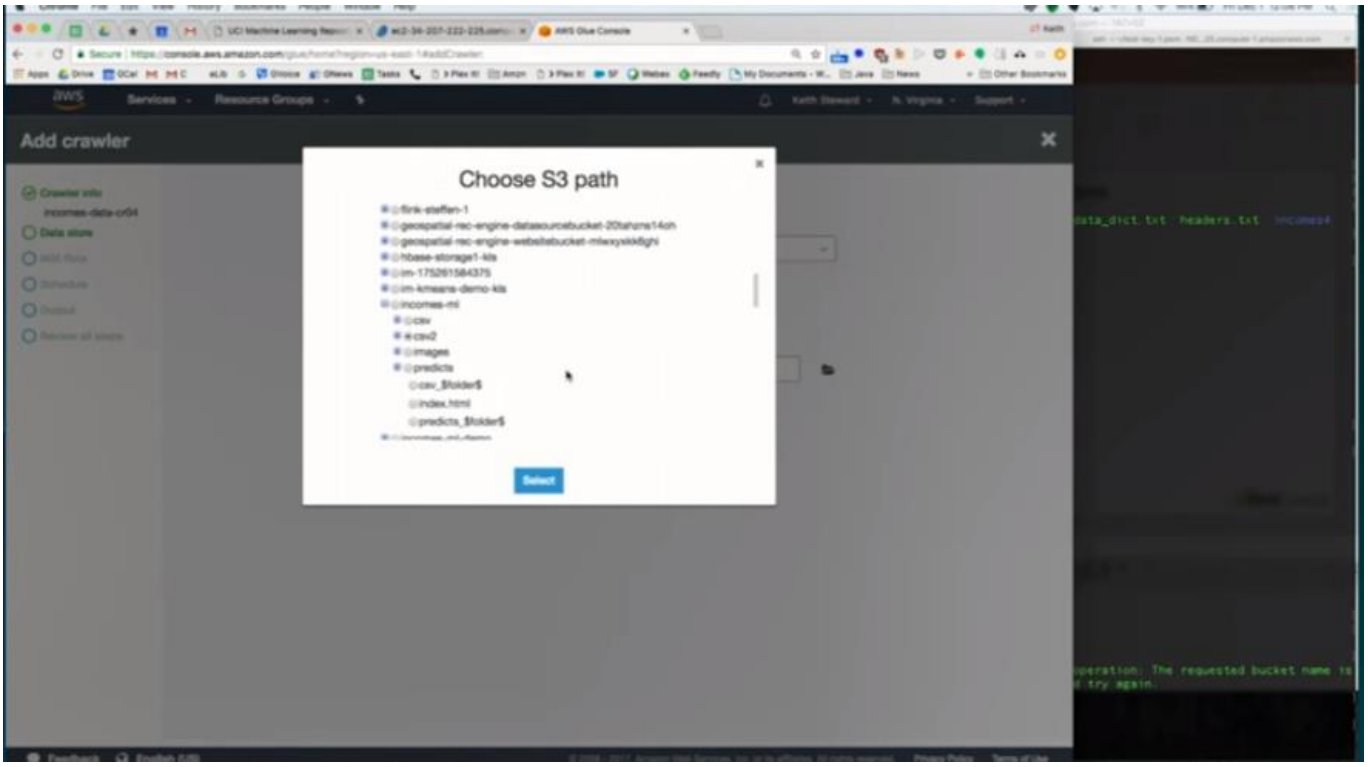


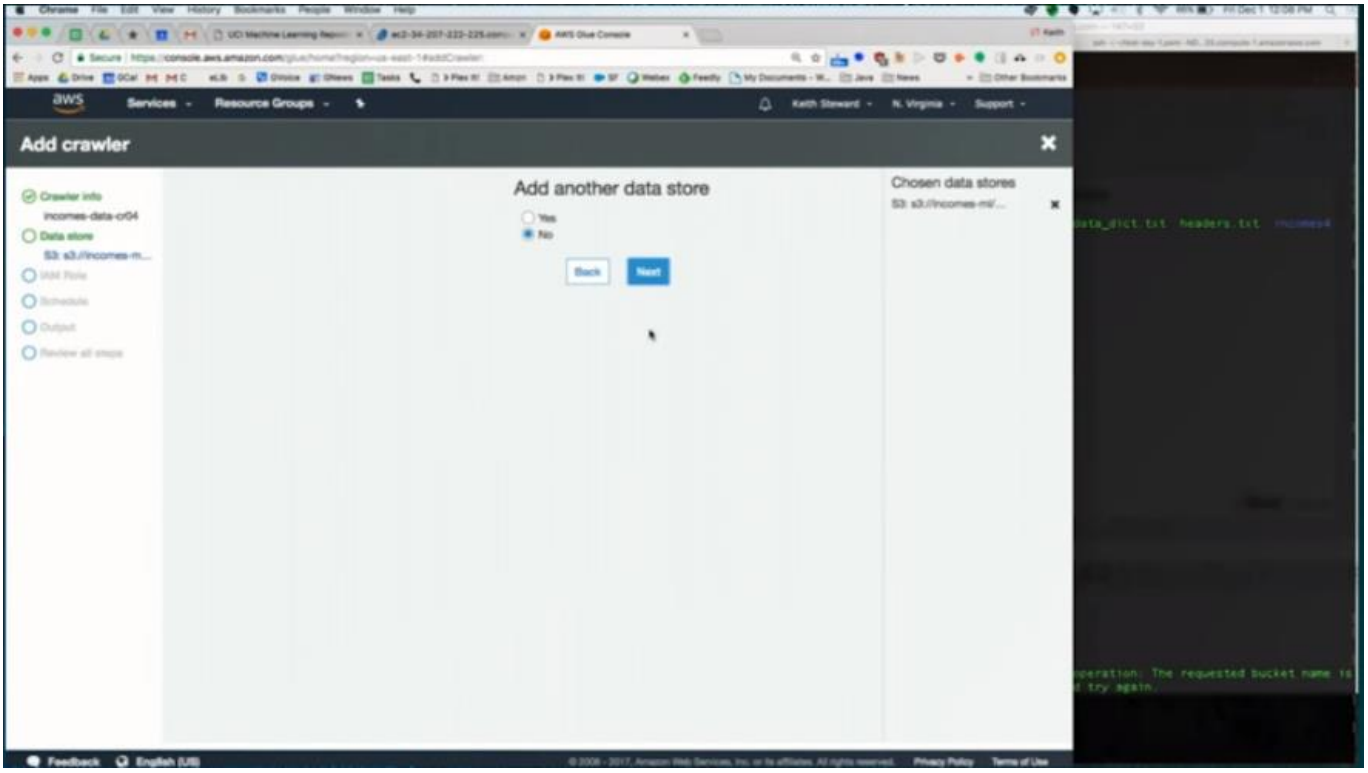

Use ***wget*** to pull down the census data and store it in an S3 bucket using the AWS S3 CLI with the ***$ aws s3 mb s3://test-bucket-kls/training*** command and then use the ***$ aws s3 cp census-income-data s3://test-bucket-kls/training/*** to copy the data into the bucket.

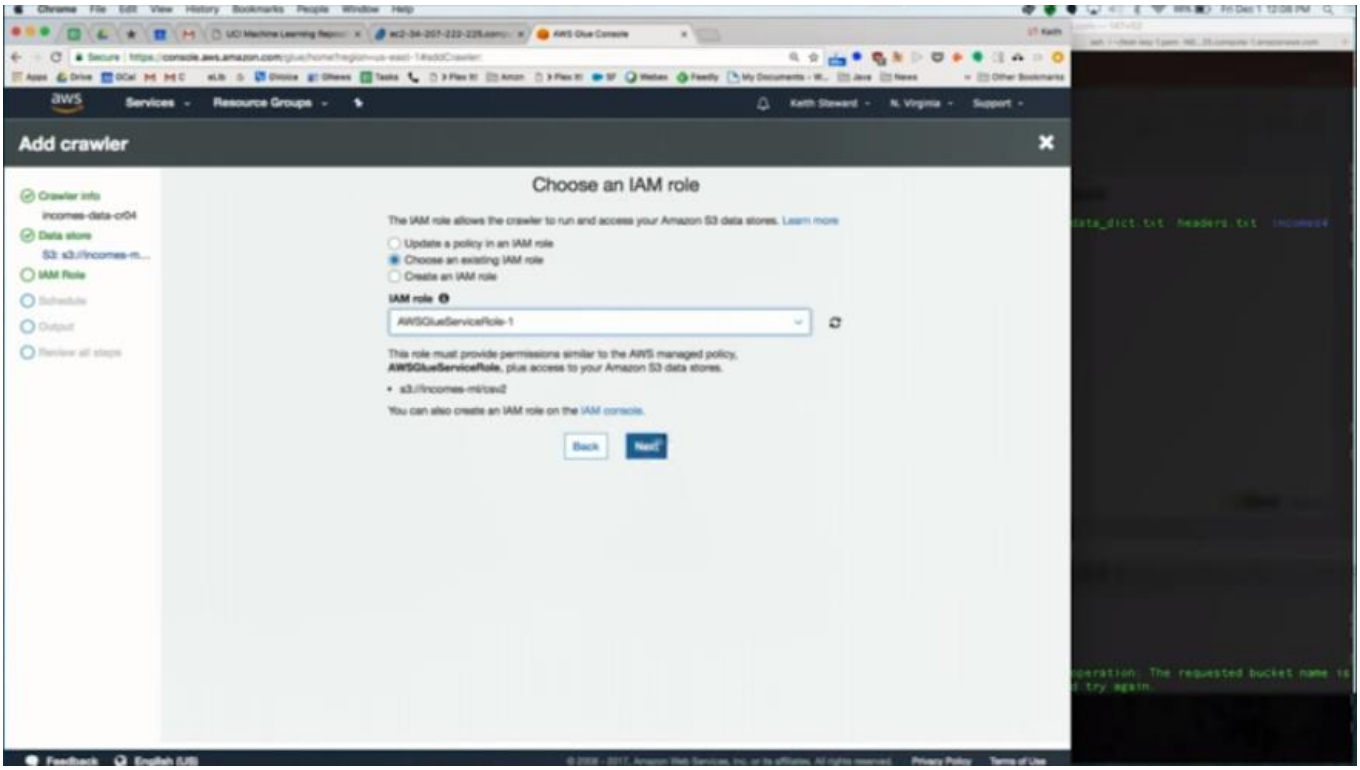


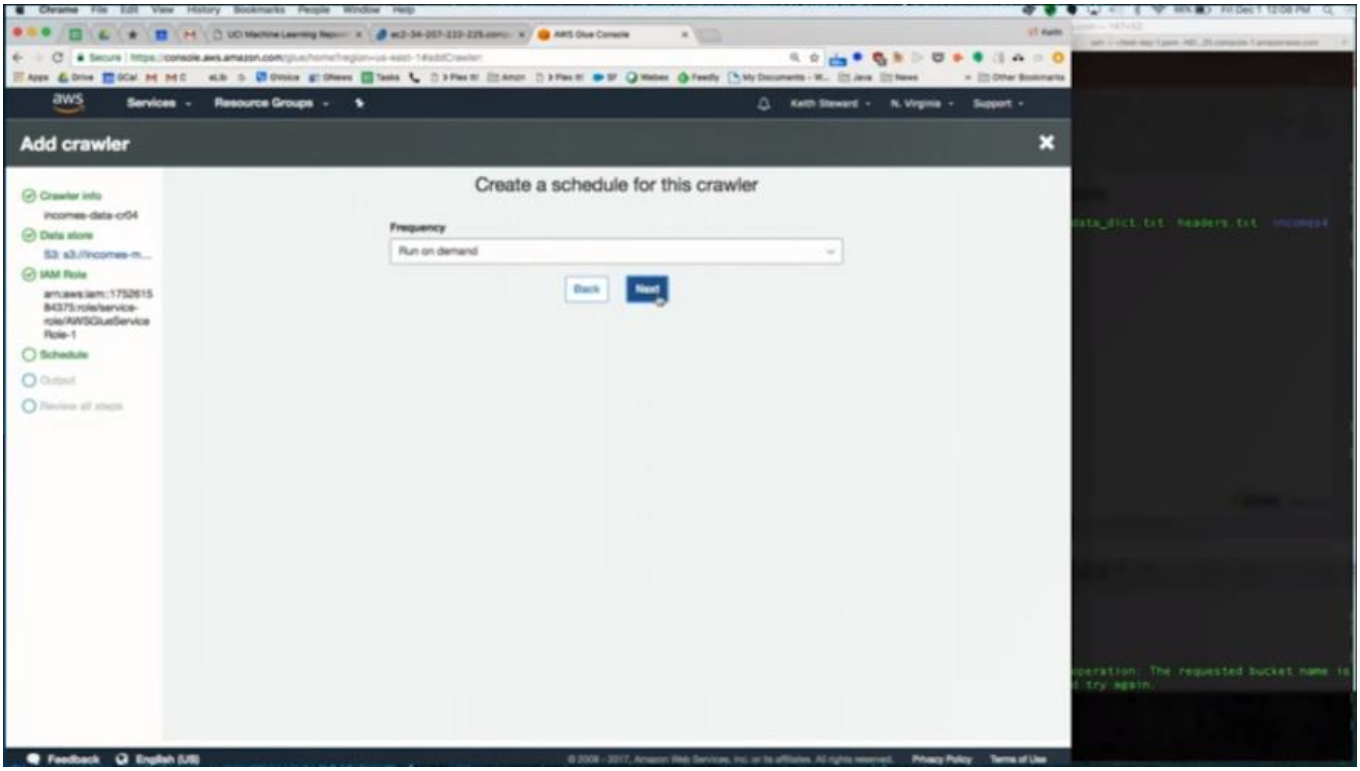

We then run AWS Glue on the data in S3 to schematize it and get us a table with all the data attributes as columns

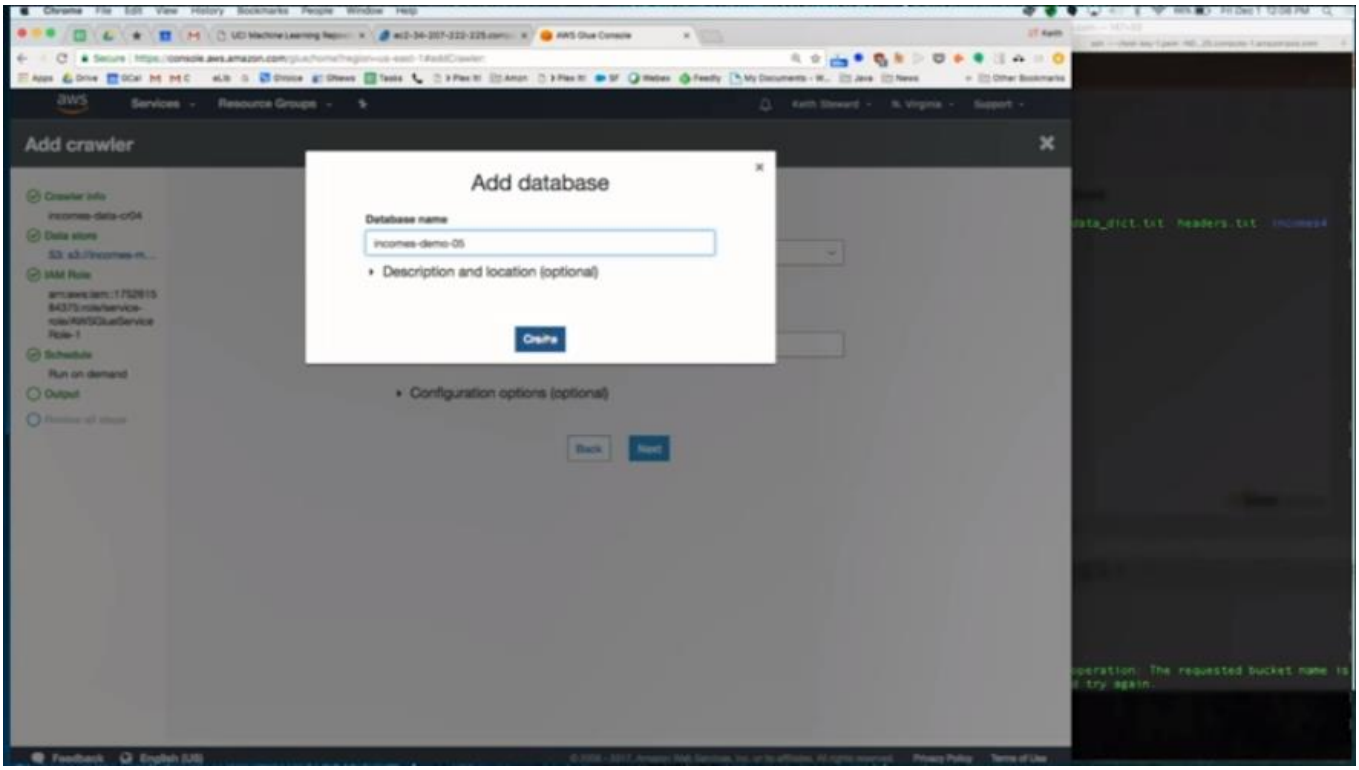


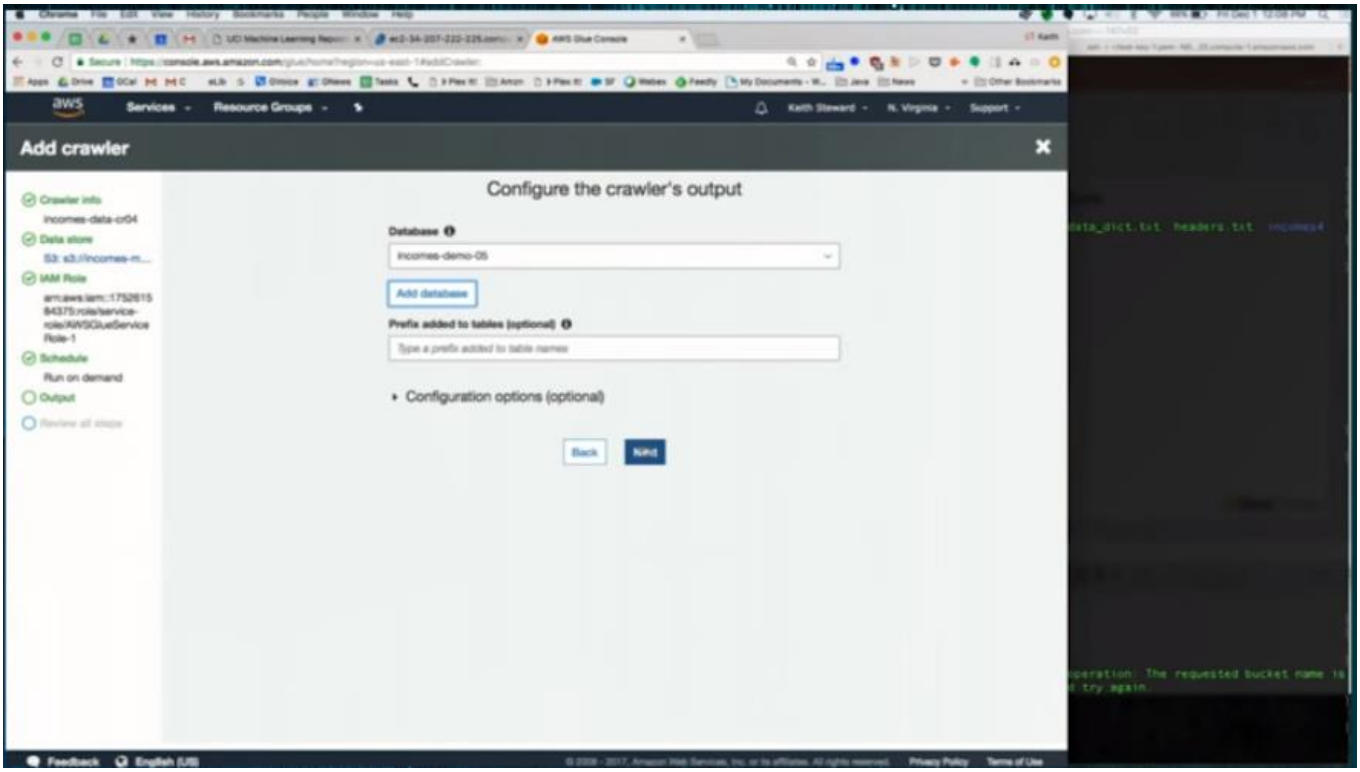

Add crawler
Choose S3 path
Select

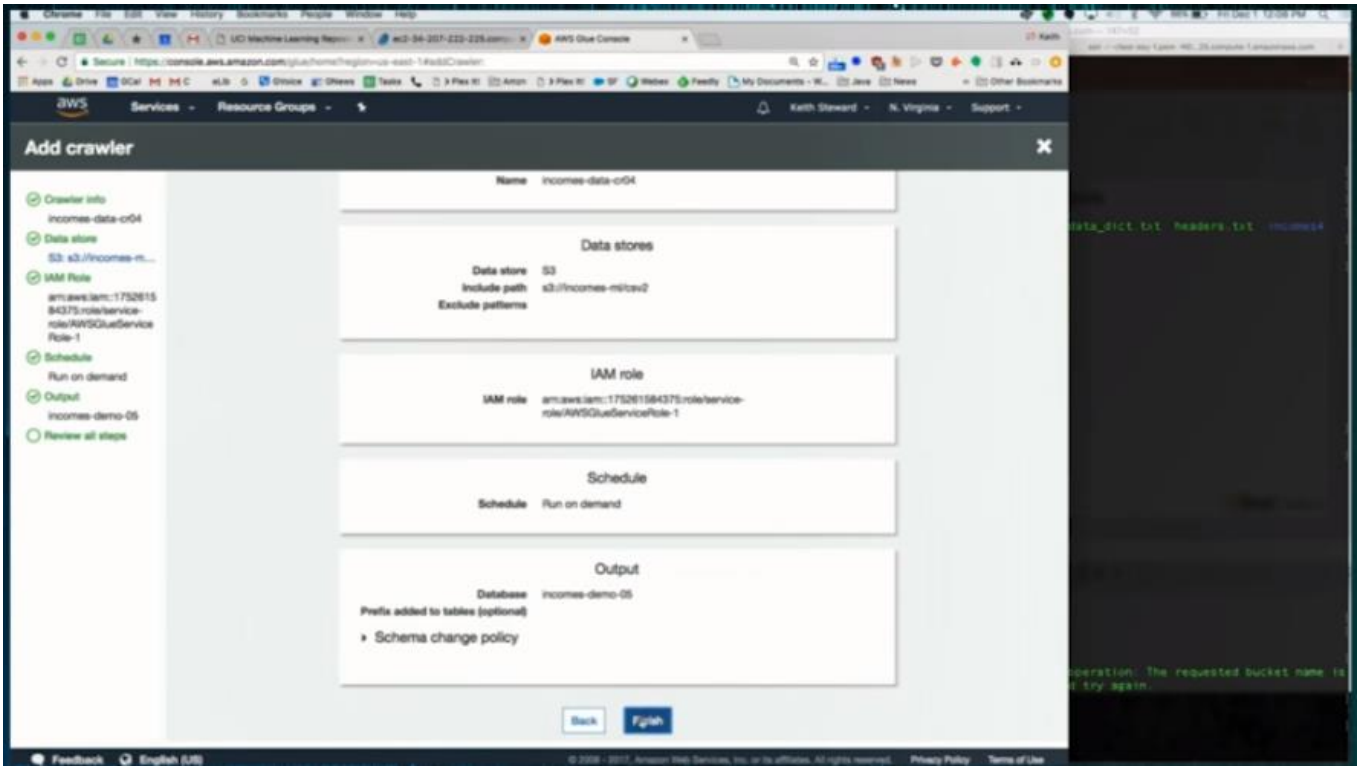

Add crawler
Add another data store
Yes
No
Back
Next
Chosen data stores
Feedback
English (US)

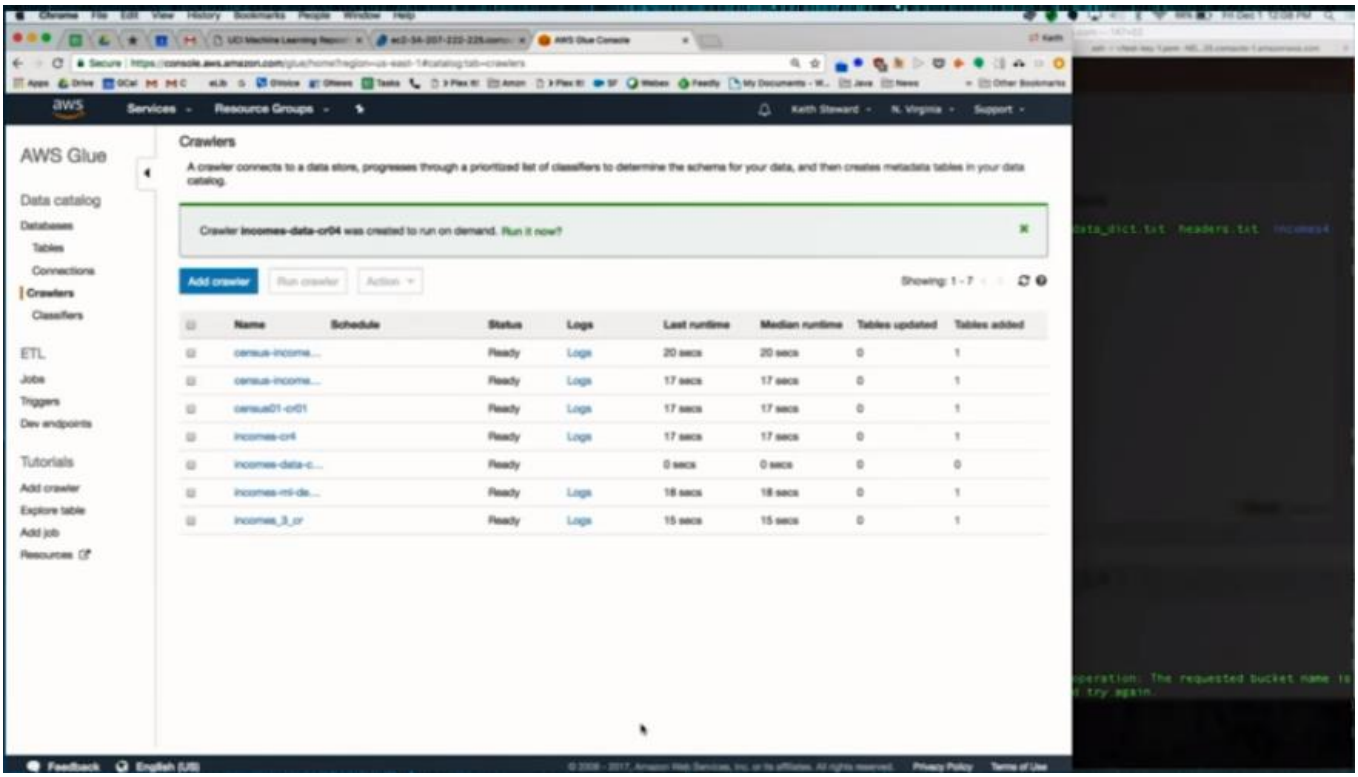

Add crawler
Crawler info
incomes-data-cr04
Data store
S3: s3://incomes-m...
IAM Role
Schedule
Output
Review all steps
Choose an IAM role
The IAM role allows the crawler to run and access your Amazon S3 data stores. Learn more
Update a policy in an IAM role
Choose an existing IAM role
Create an IAM role
IAM role
AWSGlueServiceRole-1
This role must provide permissions similar to the AWS managed policy, AWSGlueServiceRole, plus access to your Amazon S3 data stores.
s3://incomes-ml/csv2
You can also create an IAM role on the IAM console.
Back
Next

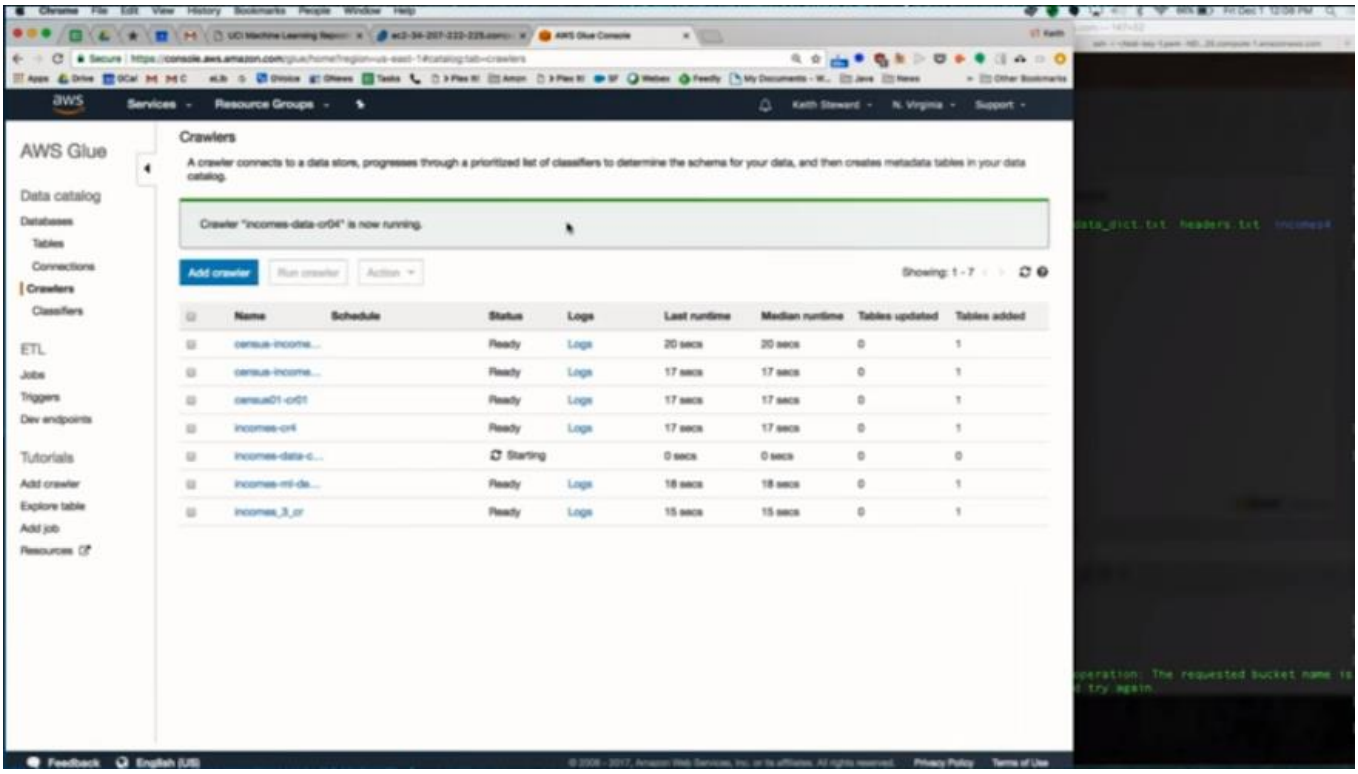

Add crawler
Crawler info
incomes-data-cr04
Data store
S3: s3://incomes-m...
IAM Role
arn:aws:iam::175261584375:role/service-role/AWSGlueServiceRole-1
Schedule
Output
Review all steps
Create a schedule for this crawler
Frequency
Run on demand
Back
Next

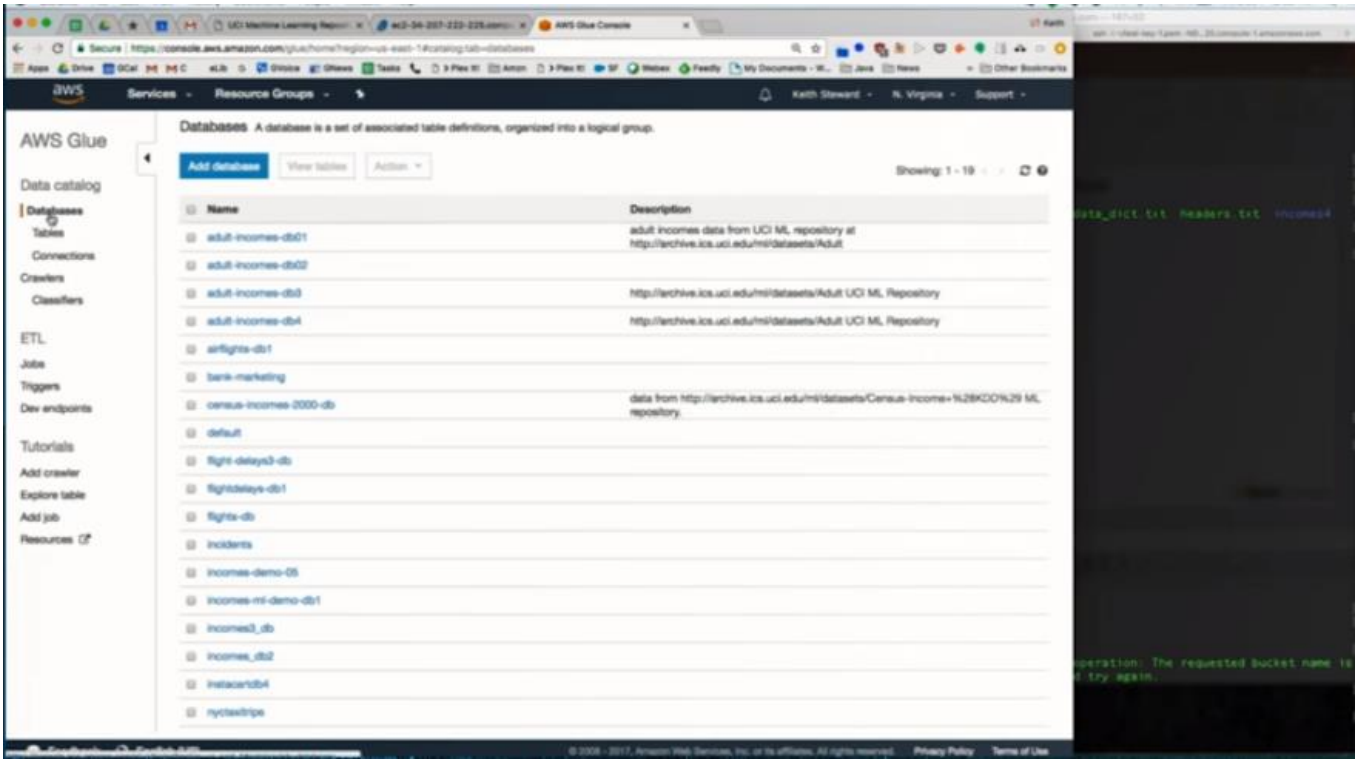

Add crawler
Add database
Database name
incomes-demo-05
Description and location (optional)
Configuration options (optional)
Back
Next


Add crawler
Configure the crawler's output
Database
incomes-demo-05
Add database
Prefix added to tables (optional)
Configuration options (optional)
Back
Next

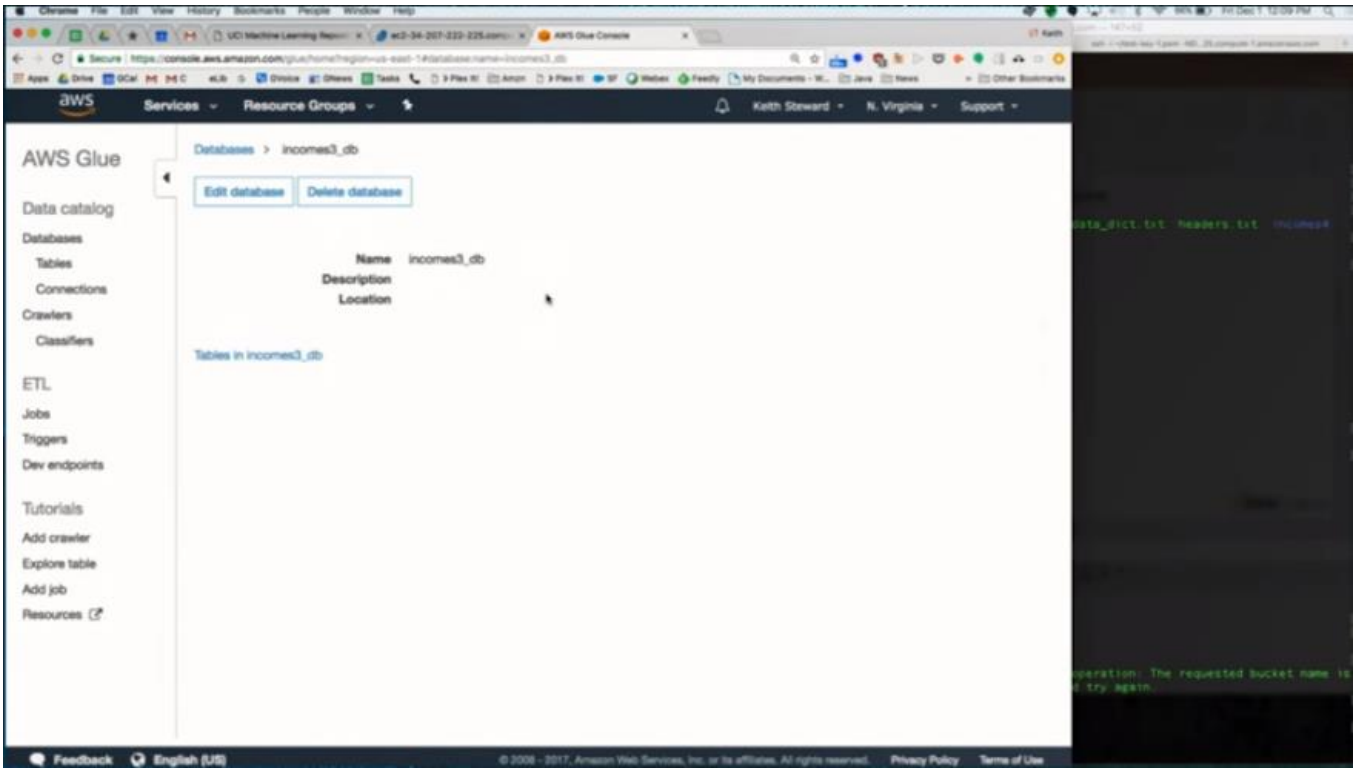

Add crawler
Crawler info
incomes-data-cr04
Data store
S3: s3://incomes-m...
IAM Role
arn:aws:iam::175261584375:role/service-role/AWSGlueServiceRole-1
Schedule
Run on demand
Output
incomes-demo-05
Review all steps
Name
incomes-data-cr04
Data stores
Data store
S3
Include path
s3://incomes-ml/csv2
Exclude patterns
IAM role
IAM role
arn:aws:iam::175261584375:role/service-role/AWSGlueServiceRole-1
Schedule
Schedule
Run on demand
Output
Database
incomes-demo-05
Prefix added to tables (optional)
Schema change policy
Back
Finish

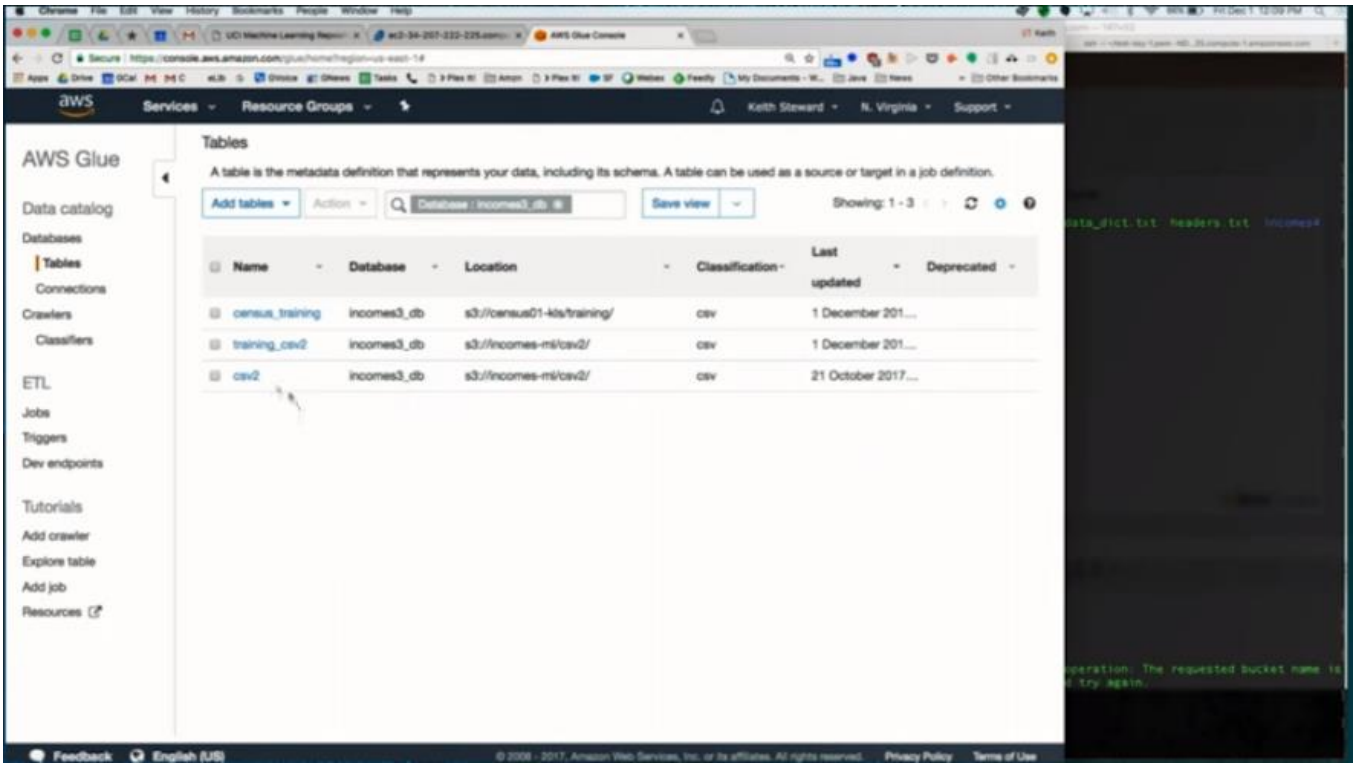

AWS Glue
Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
ETL
Jobs
Triggers
Dev endpoints
Tutorials
Add crawler
Explore table
Add job
Resources
Crawlers
A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.
Crawler incomes-data-cr04 was created to run on demand. Run it now?
Add crawler
Run crawler
Action
Showing: 1 - 7
Name
Schedule
Status
Logs
Last runtime
Median runtime
Tables updated
Tables added
census-income...
Ready
Logs
20 secs
0
1
census-income...
Ready
Logs
17 secs
17 secs
0
1
census01-cr01
Ready
Logs
17 secs
17 secs
0
1
incomes-cr4
Ready
Logs
17 secs
17 secs
0
1
incomes-data-c...
Ready
0 secs
0 secs
0
0
incomes-ml-da...
Ready
Logs
18 secs
18 secs
0
1
incomes_3_cr
Ready
Logs
15 secs
15 secs
0
1

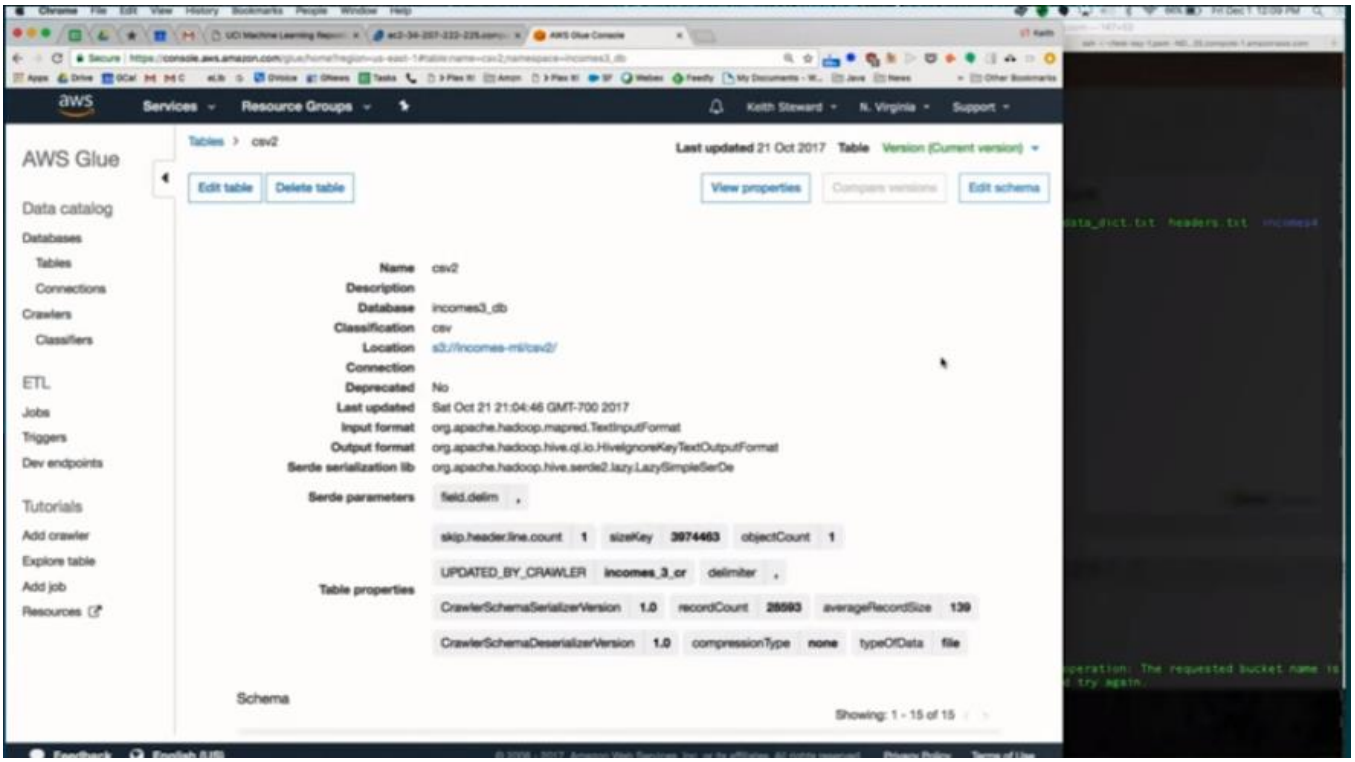


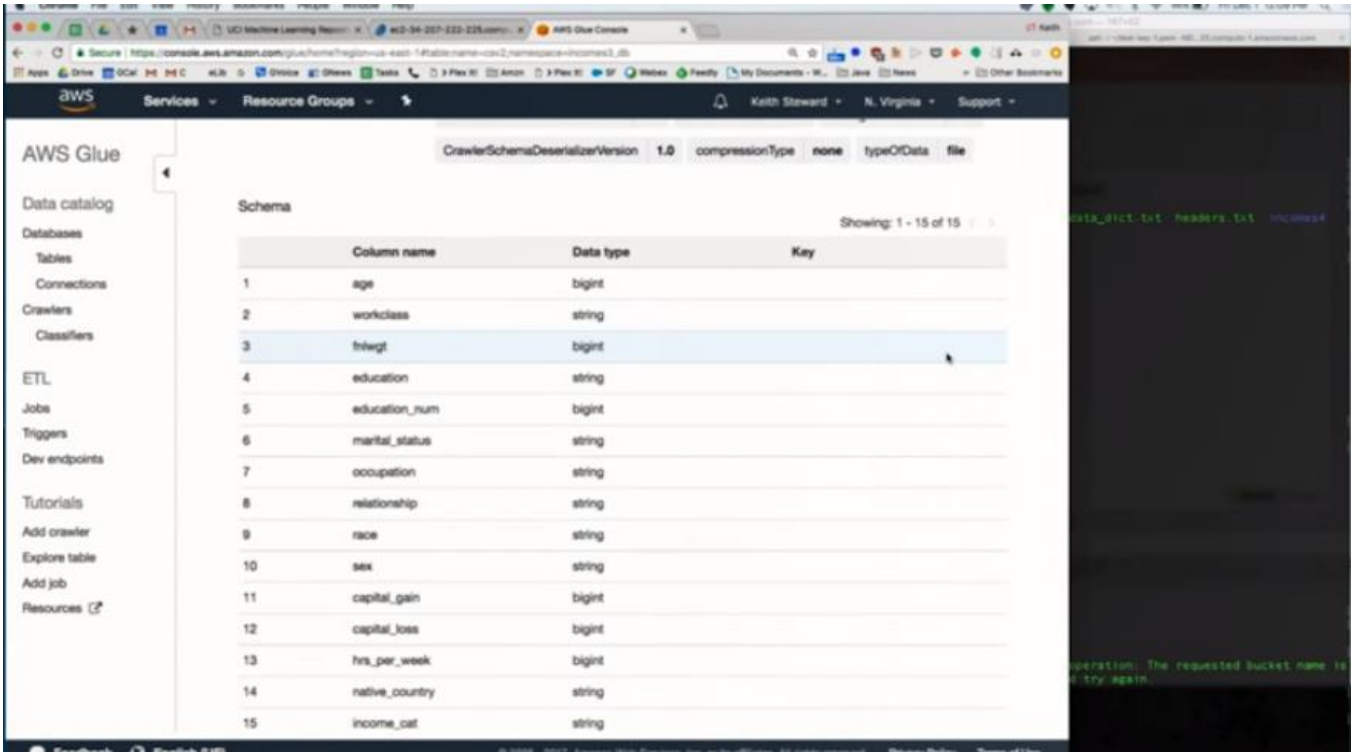




When the AWS Glue crawler finishes, it creates a database with a bunch of tables to store the results in for us

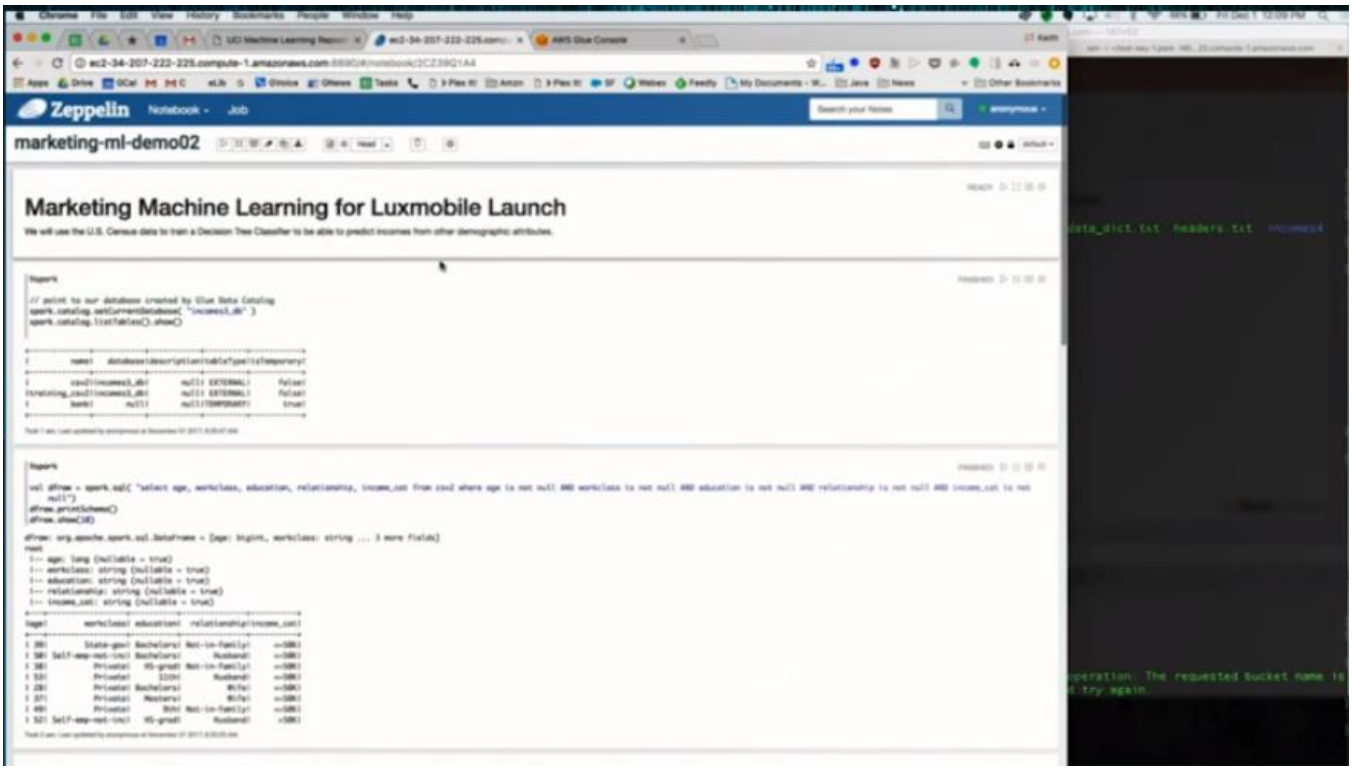

Services
Resource Groups
Keith Steward
N. Virginia
Support
AWS Glue
Databases > incomes3_db
Edit database
Delete database
Name incomes3_db
Description
Location
Tables in incomes3_db
Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
ETL
Jobs
Triggers
Dev endpoints
Tutorials
Add crawler
Explore table
Add job
Resources
Feedback
English (US)
Privacy Policy
Terms of Use


Services
Resource Groups
Keith Steward
N. Virginia
Support
AWS Glue
Tables
A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.
Add tables
Action
Save view
Showing: 1 - 3
Name
Database
Location
Classification
Last updated
Deprecated
census_training incomes3_db s3://census01-kls/training/ csv 1 December 201...
training_csv2 incomes3_db s3://incomes-ml/csv2/ csv 1 December 201...
csv2 incomes3_db s3://incomes-ml/csv2/ csv 21 October 2017...
Data catalog
Databases
Tables
Connections
Crawlers
Classifiers
ETL
Jobs
Triggers
Dev endpoints
Tutorials
Add crawler
Explore table
Add job
Resources
Feedback
English (US)
Privacy Policy
Terms of Use

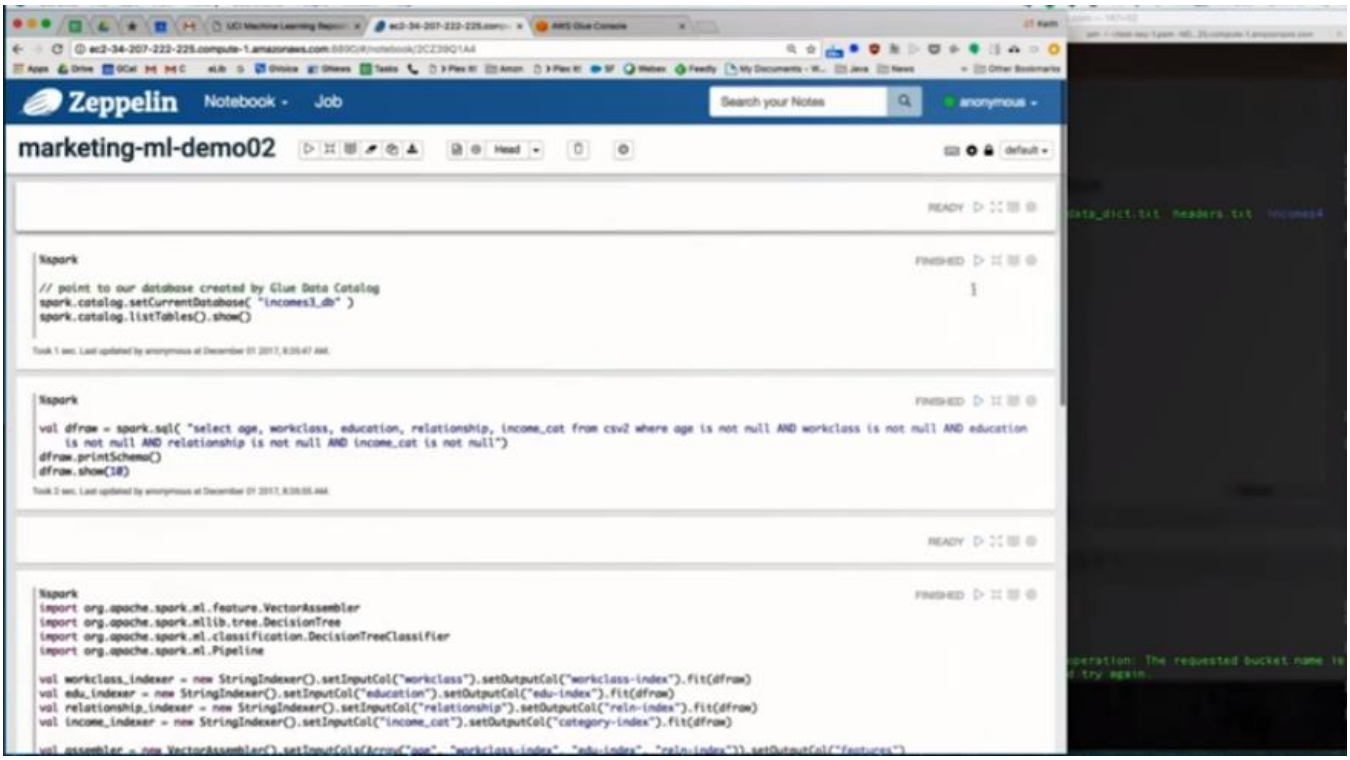


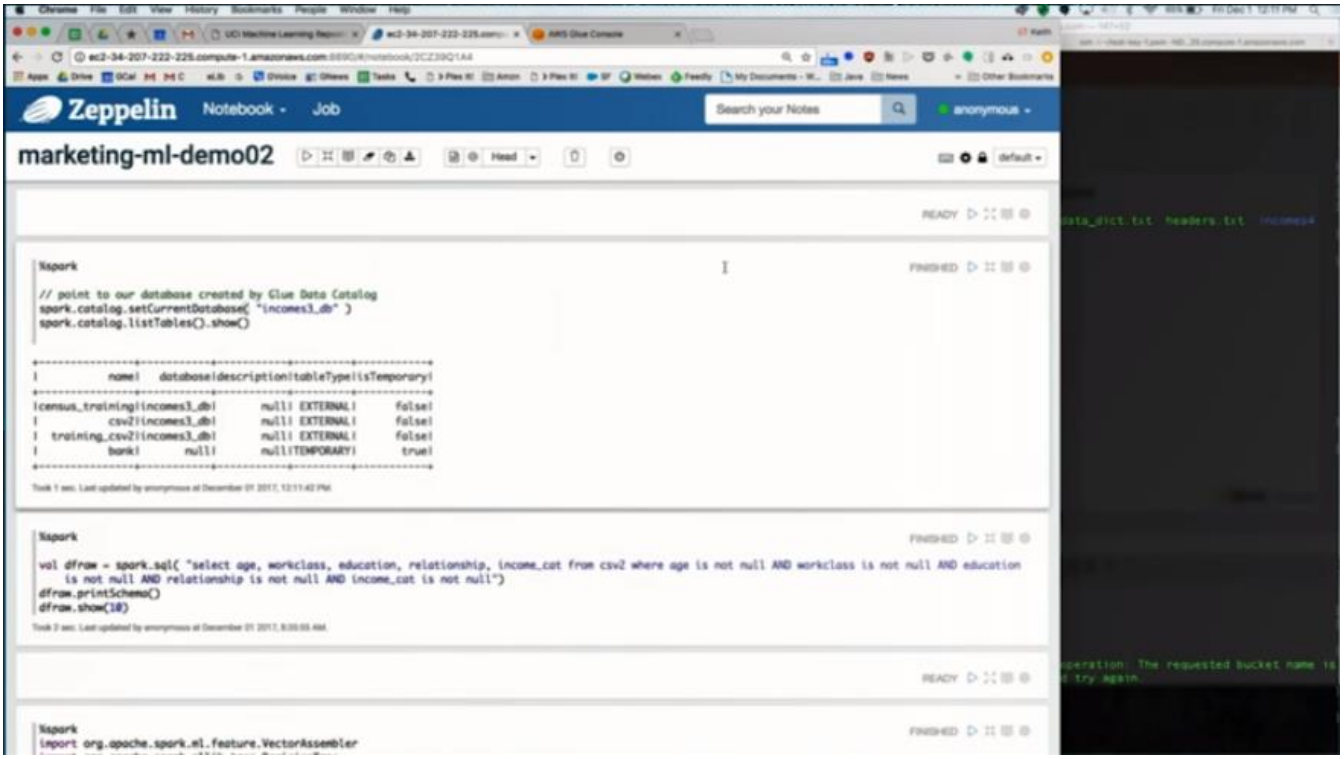

It has now converted the CSV data into a suitable schema for us

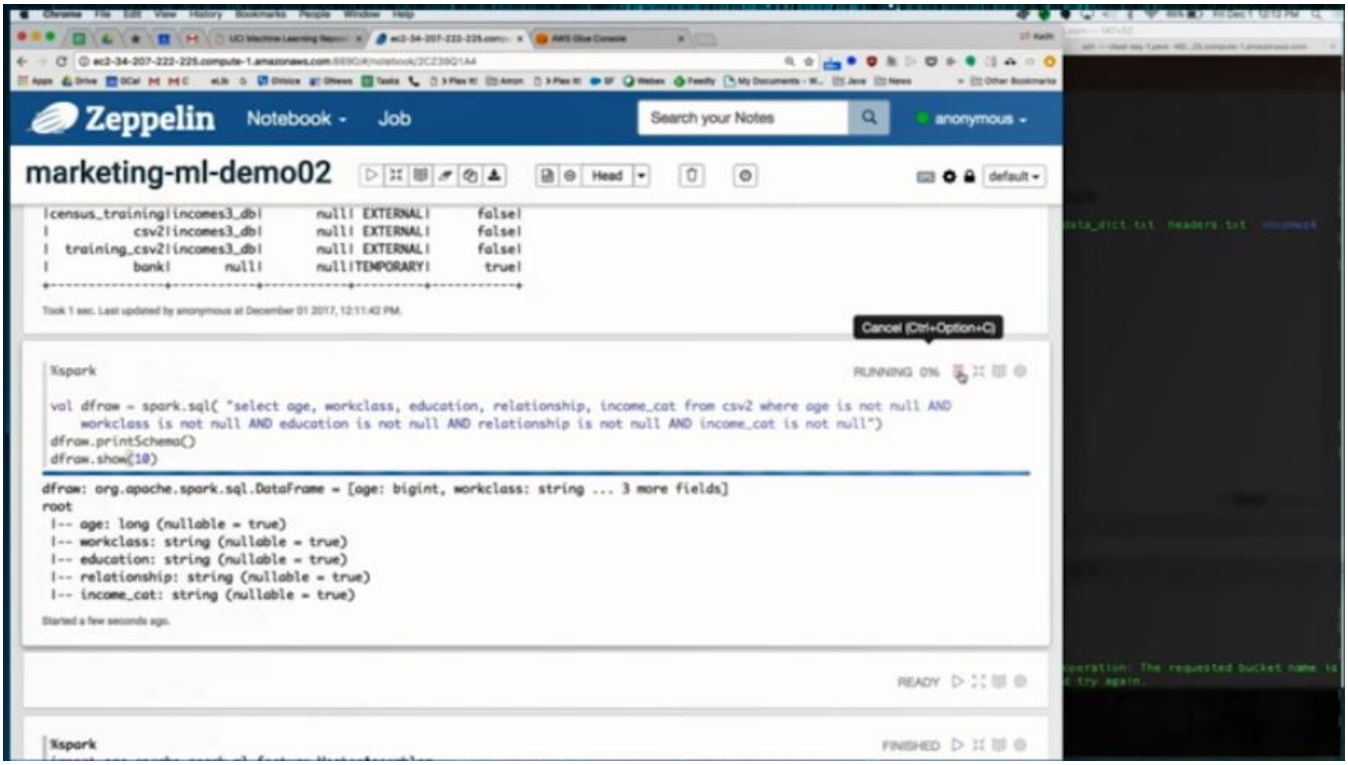



We then spin up an EMR cluster for our ML process with Spark and Zeppelin using the Web UI to have the applications running, we can then open up the Zeppelin notebook that is running on the master node in the EMR cluster as above

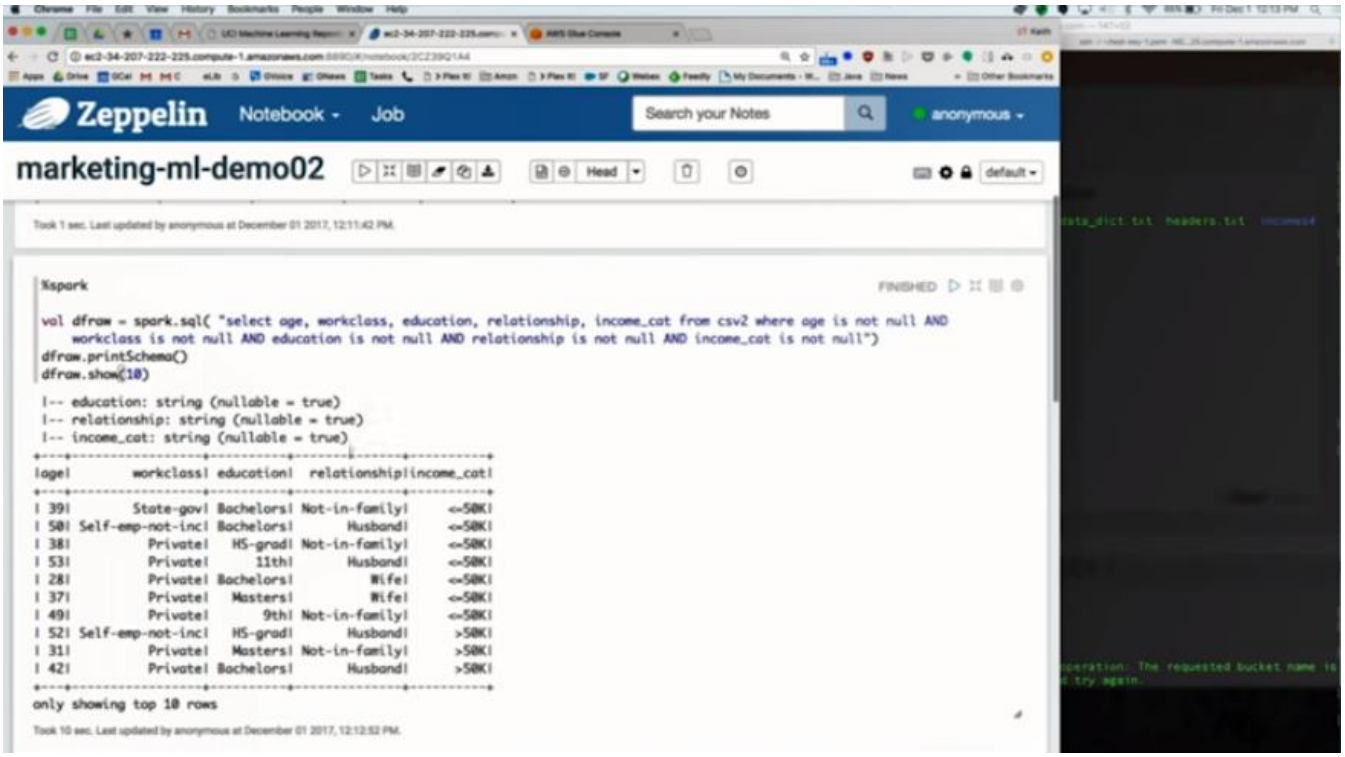



We are then running Spark with the notebook to run Spark code, this will distribute all the processing across all the nodes within the 10-node cluster we have. We can then start running Spark code as above

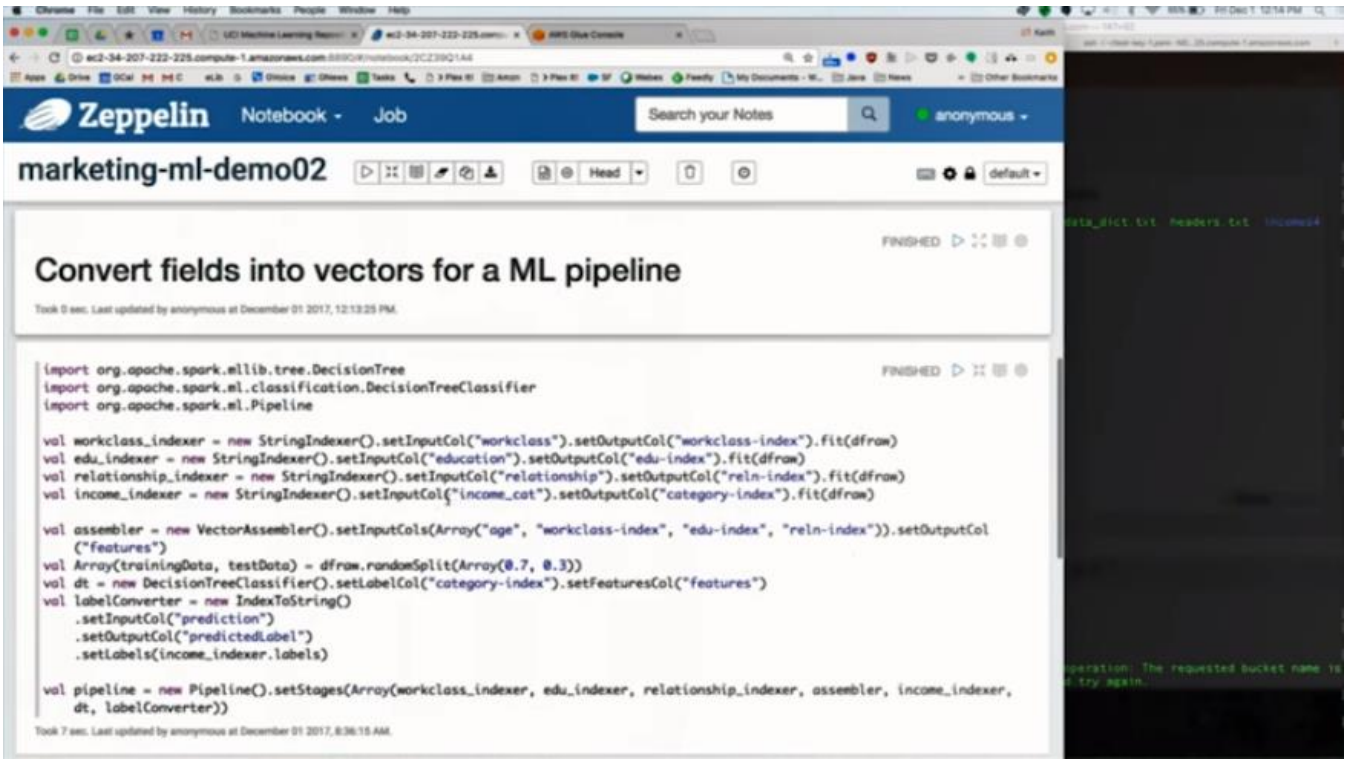


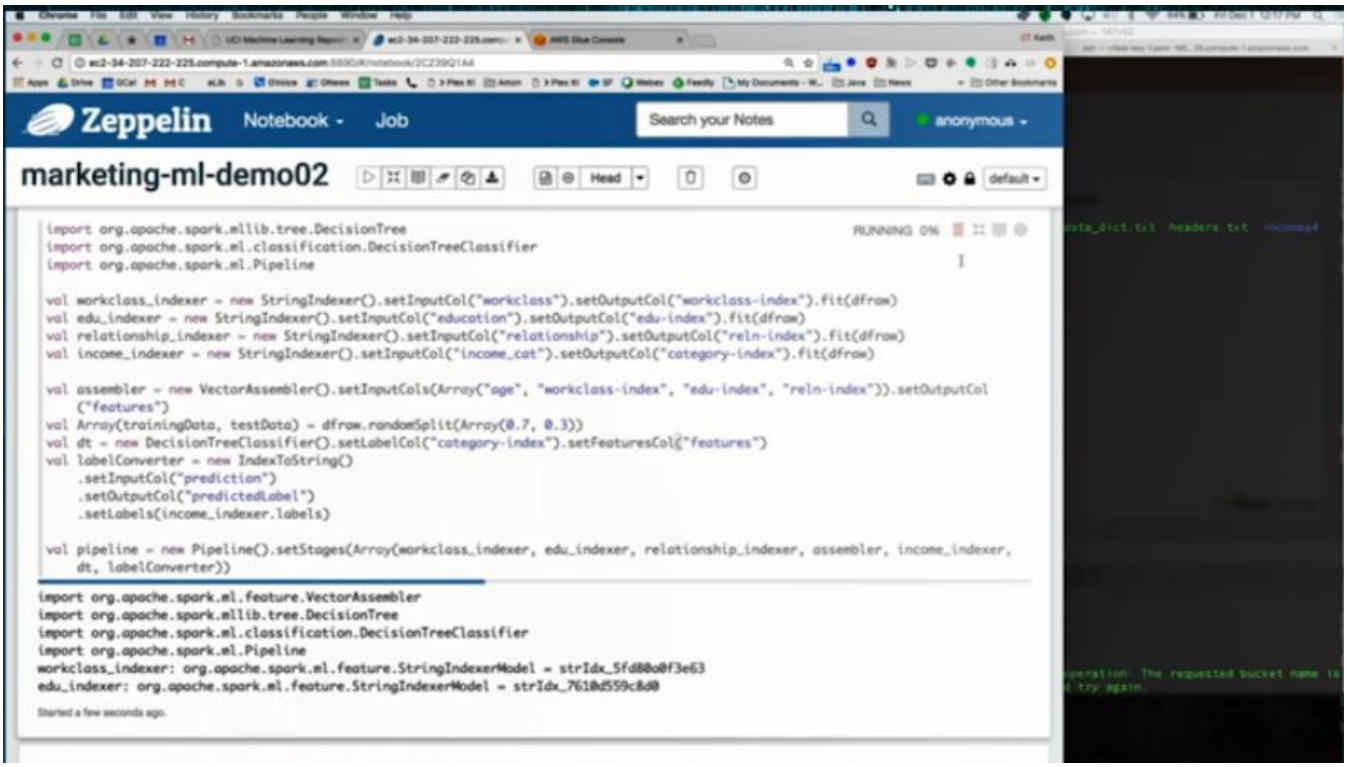

Next, we run another query to create a data frame by running Spark SQL across the Glue data catalog table called csv2 and pull out specific fields we need.



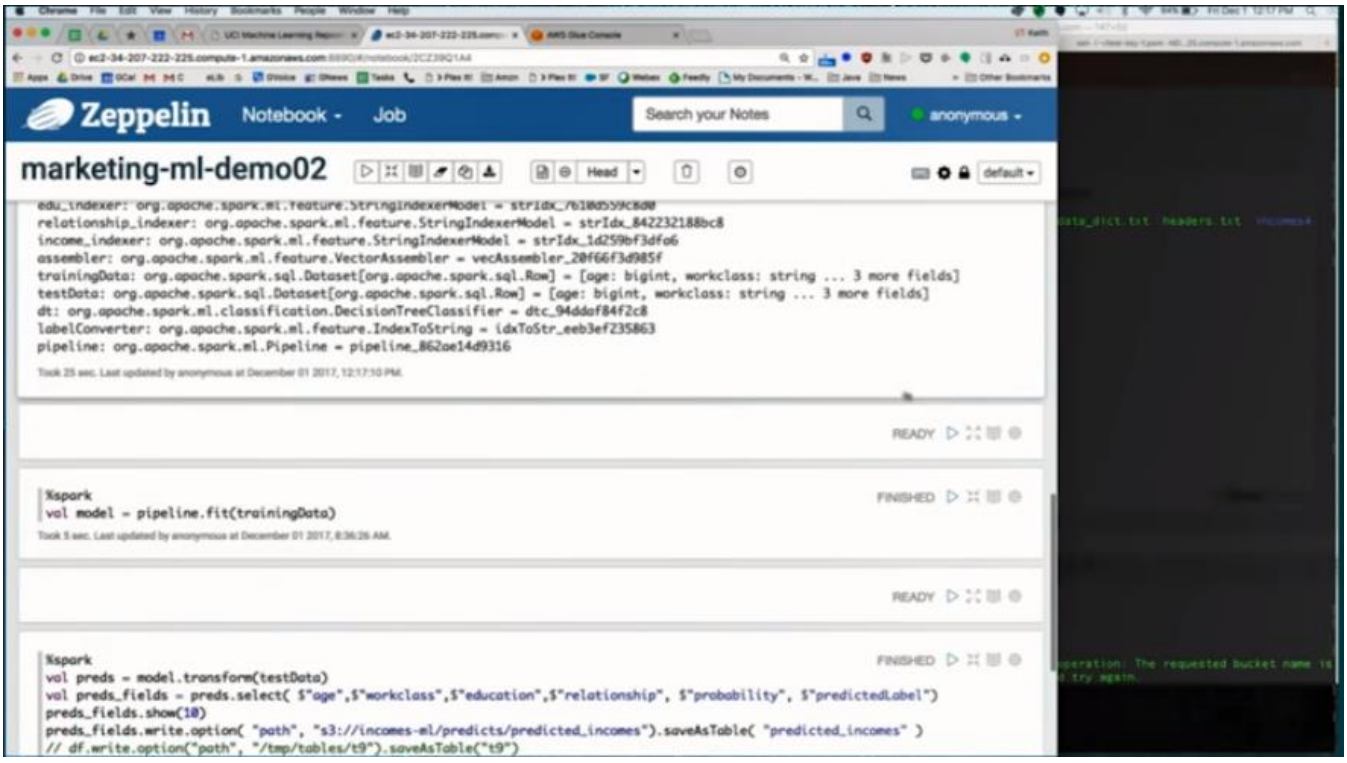

Using the data frame and running the query will put the result data in-memory for us to do further work with.

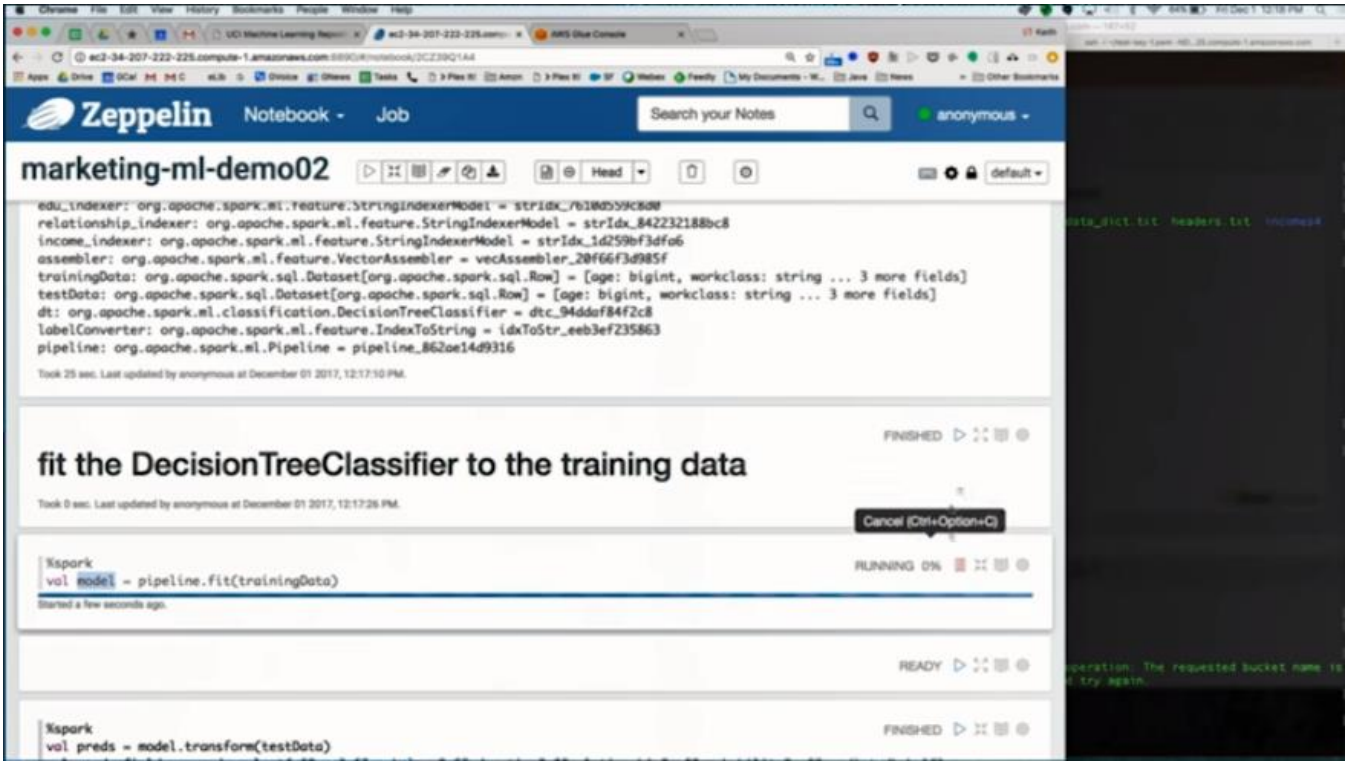



We can now start to do the training of the model in this step.

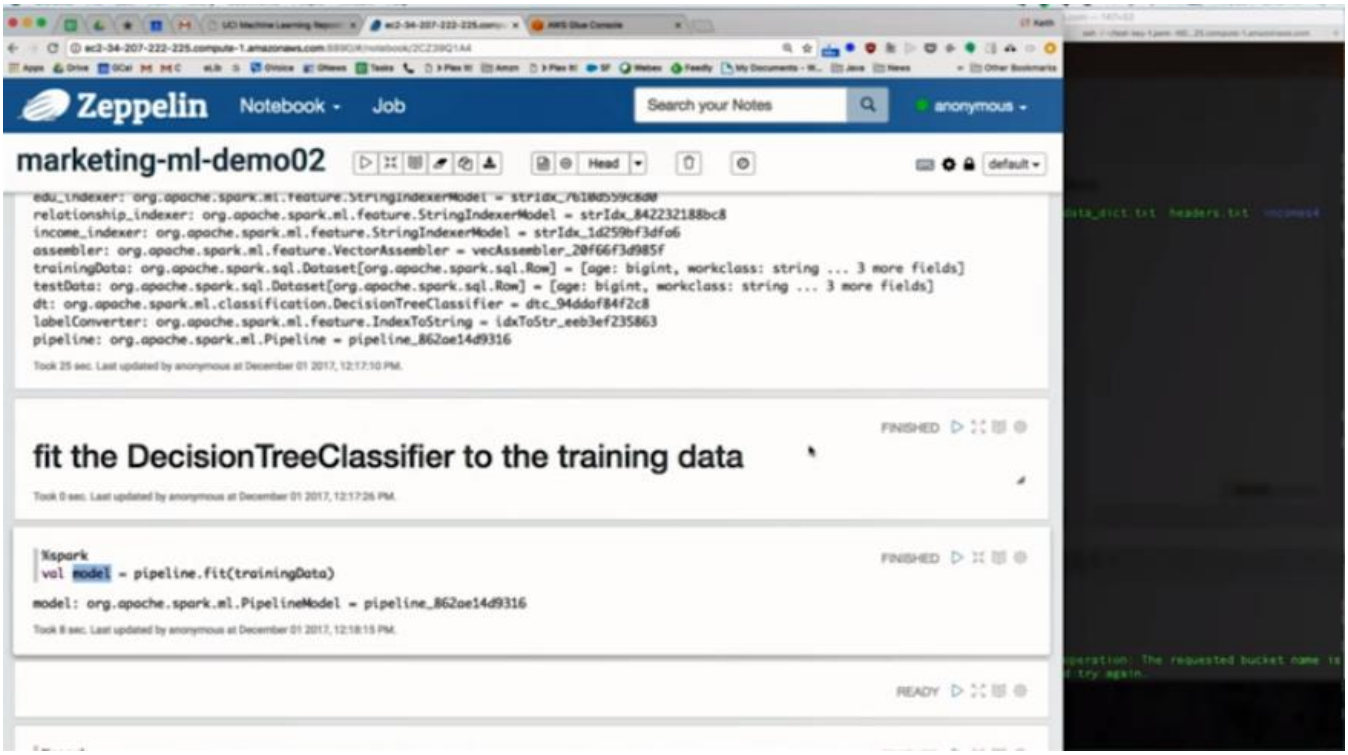



We then run the pipeline query again

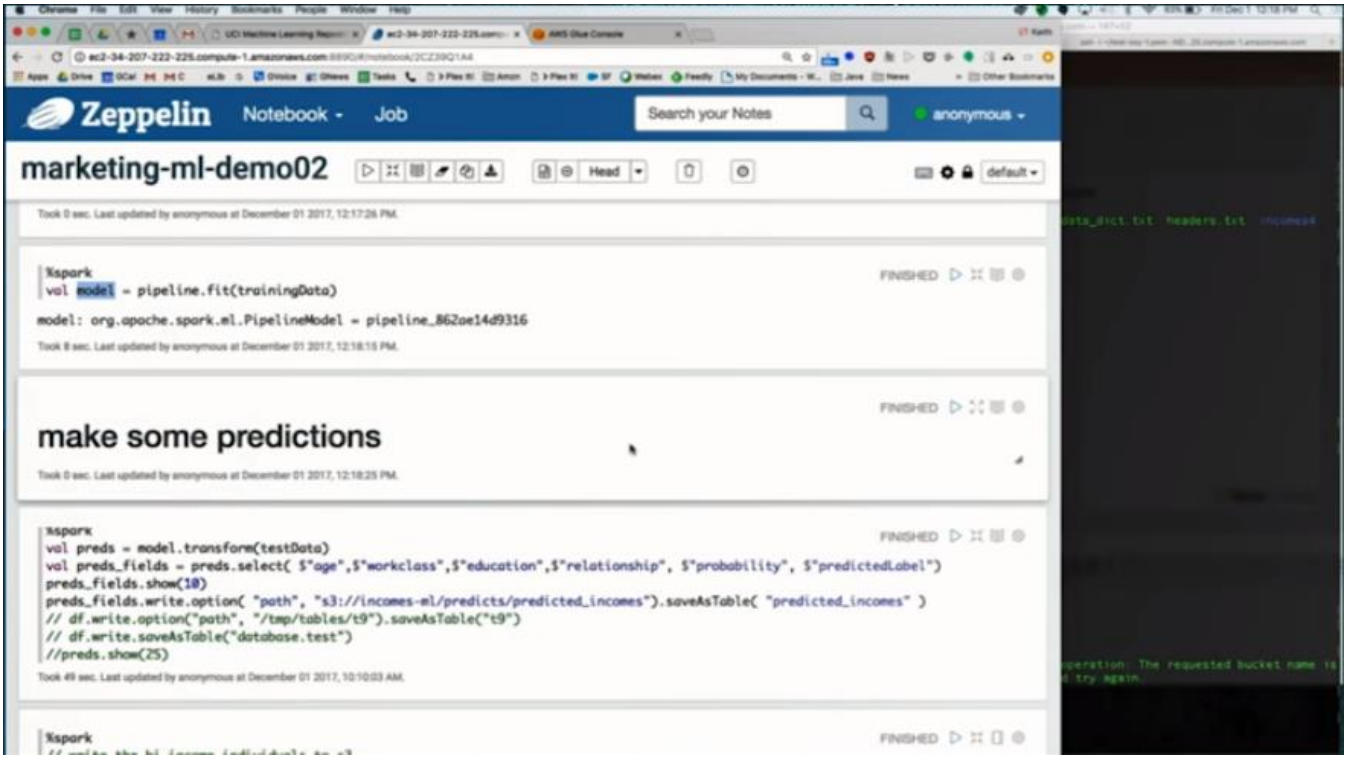




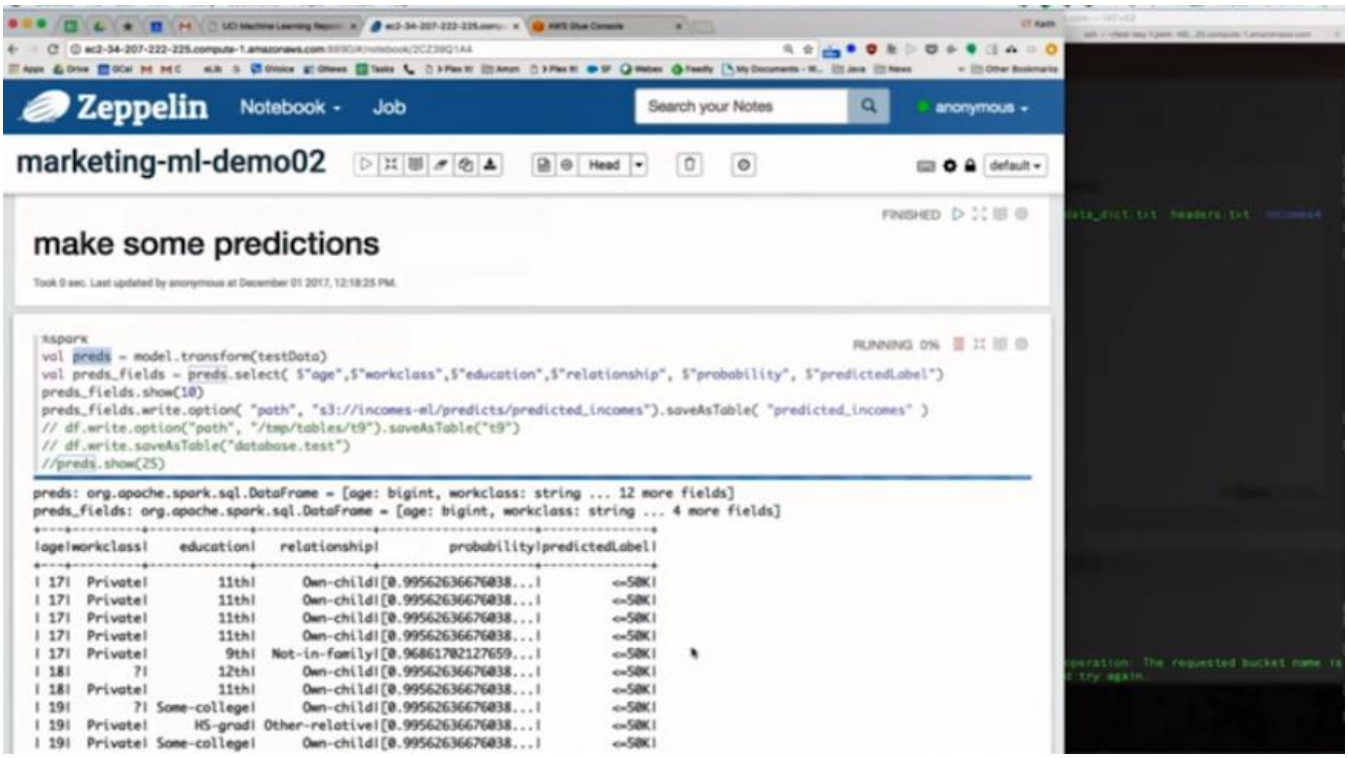

Now that we have the new data columns, we now need to fit the pipeline to the training model

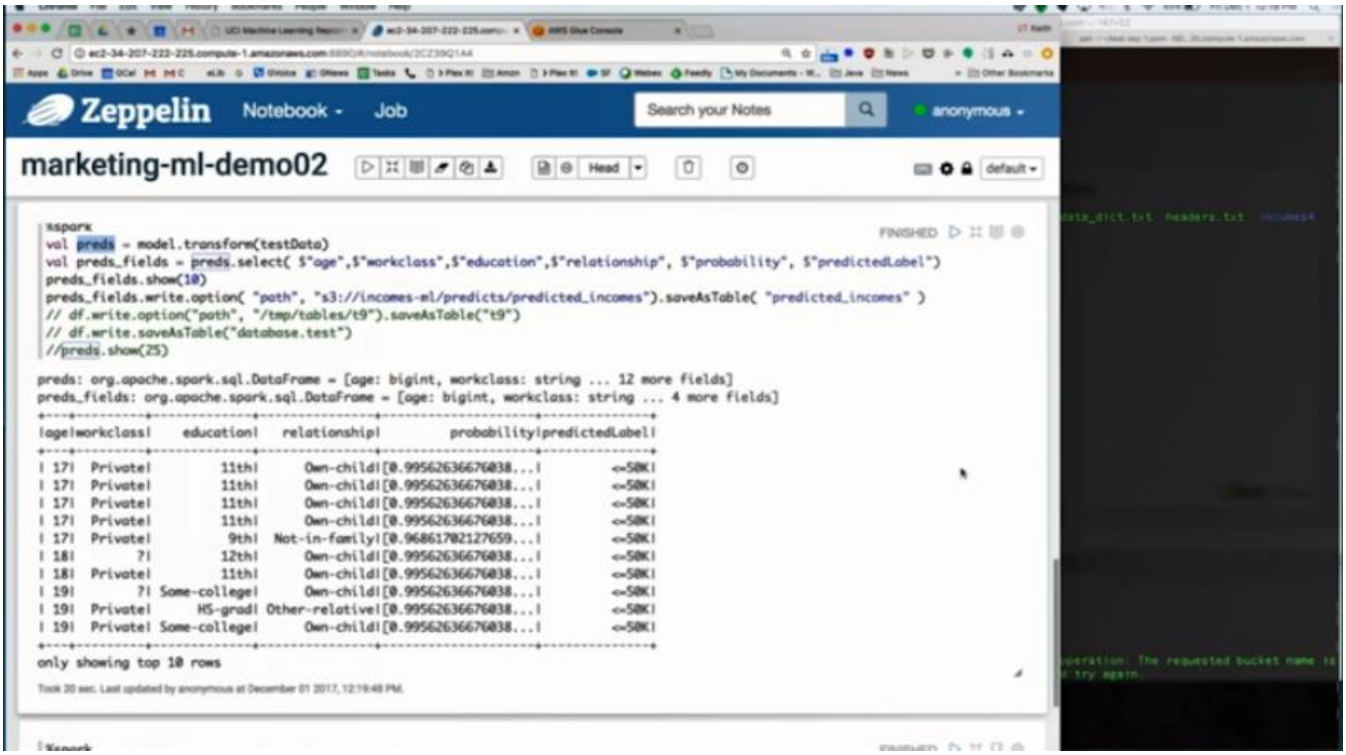


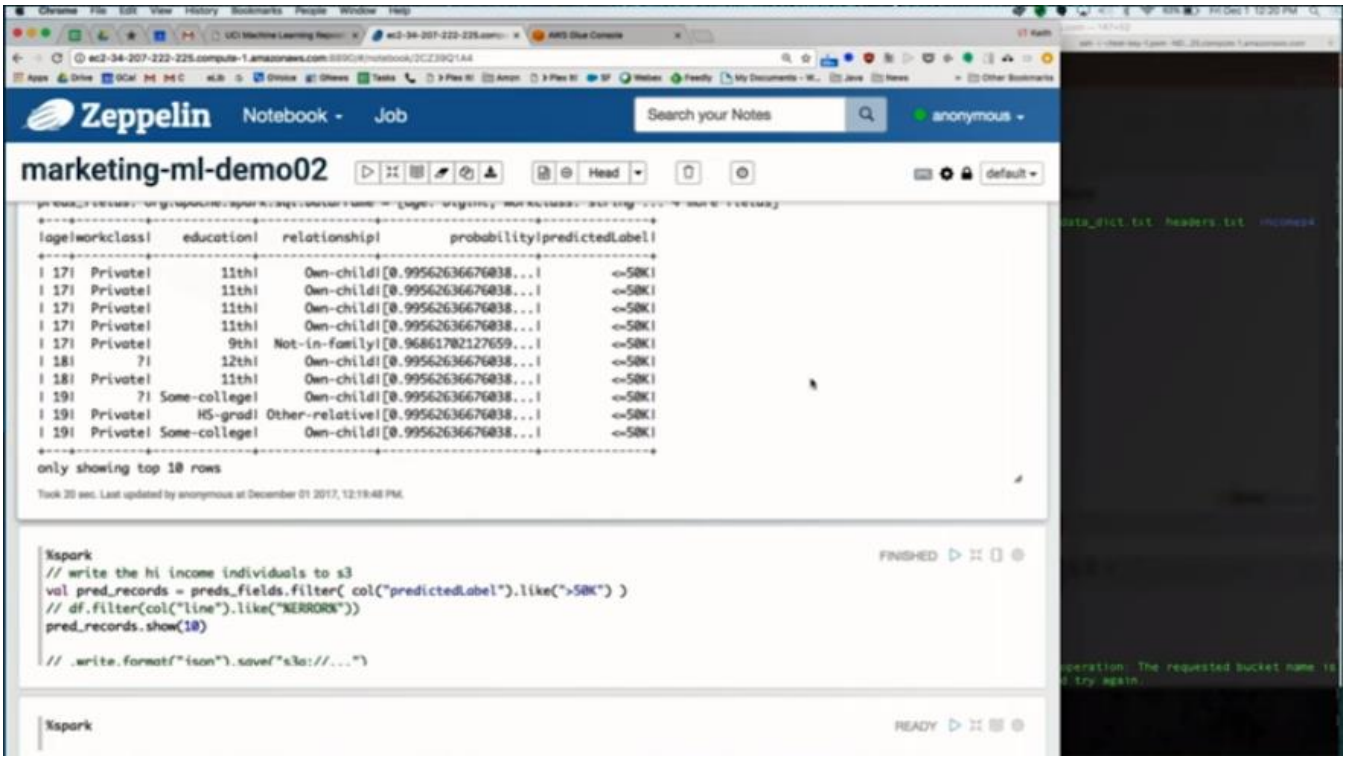

We now have a trained model

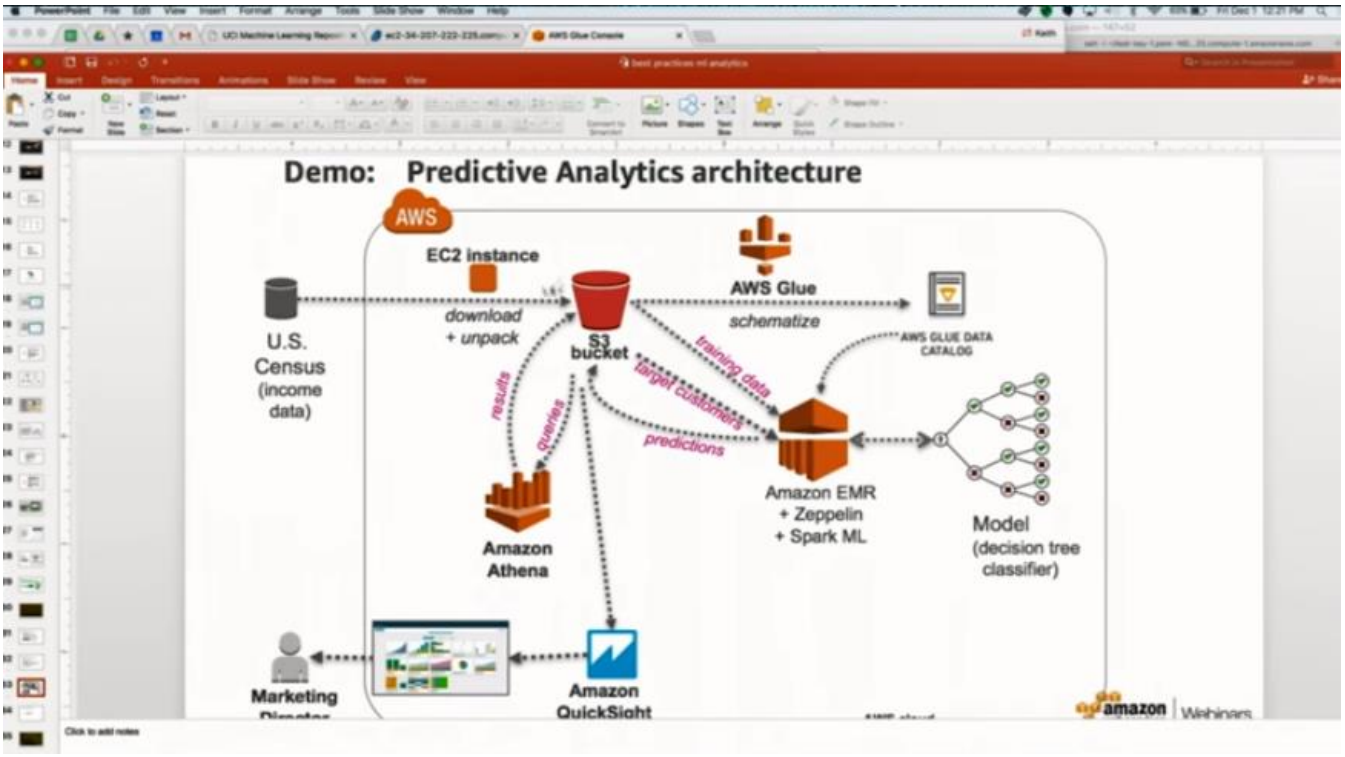



We can now make some predictions on a different set of data to infer the income level of the samples using the trained model, we then write the results back to an S3 bucket using a Hive table with the results

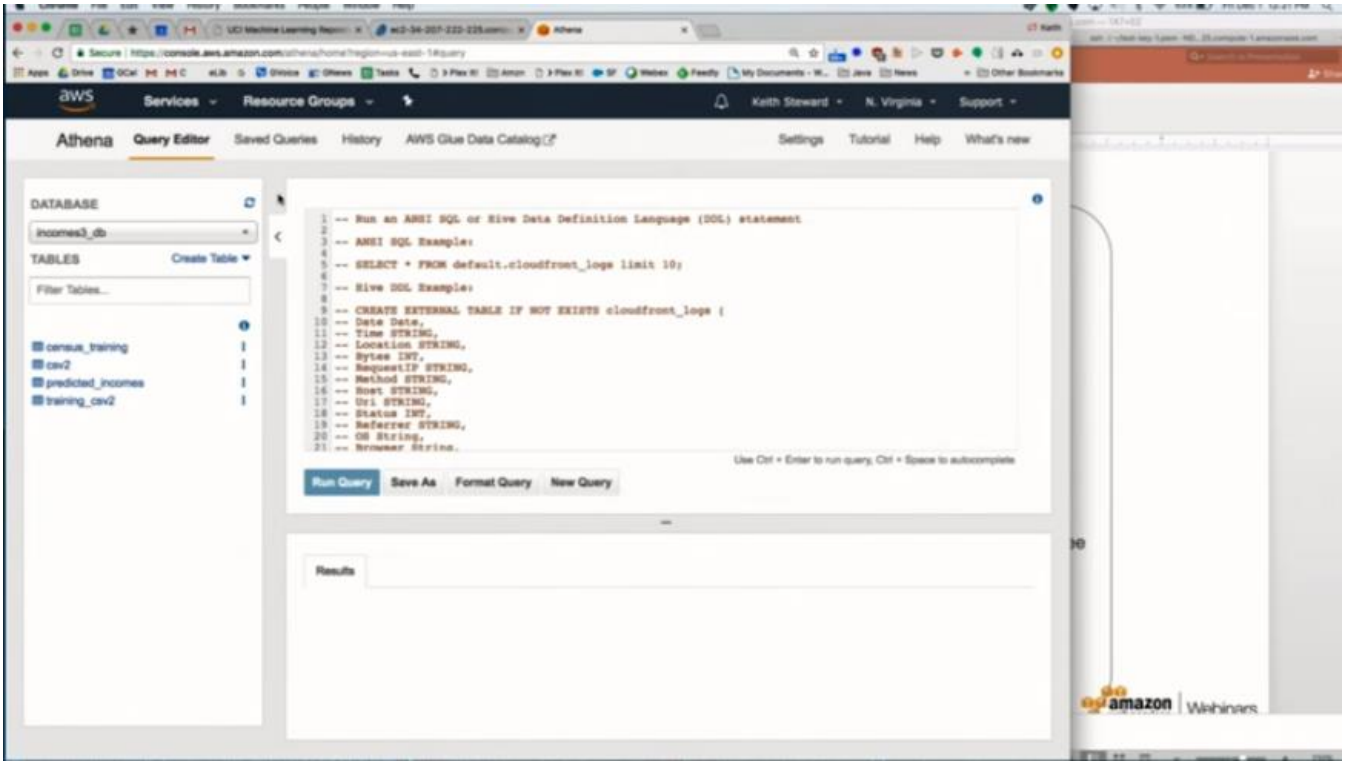


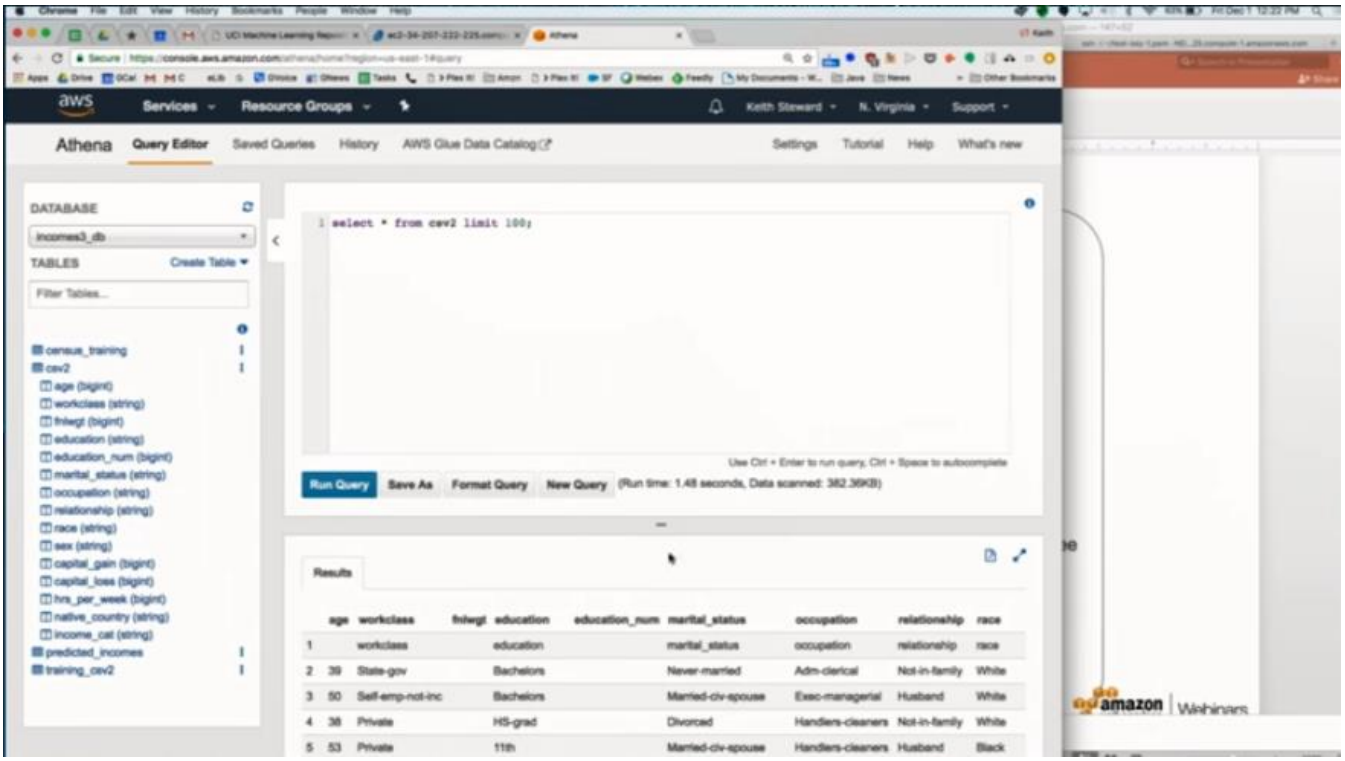

Zeppelin
Notebook
Job
Search your Notes
anonymous
marketing-ml-demo02
Head
default
%spark
val preds = model.transform(testData)
val preds_fields = preds.select( $"age",$"workclass",$"education",$"relationship", $"probability", $"predictedLabel")
preds_fields.show(10)
preds_fields.write.option( "path", "s3://incomes-ml/predicts/predicted_incomes").saveAsTable( "predicted_incomes" )
// df.write.option("path", "/tmp/tables/t9").saveAsTable("t9")
// df.write.saveAsTable("database.test")
//preds.show(25)
FINISHED
preds: org.apache.spark.sql.DataFrame = [age: bigint, workclass: string ... 12 more fields]
preds_fields: org.apache.spark.sql.DataFrame = [age: bigint, workclass: string ... 4 more fields]
+---+---------+------------+------------------+--------------------+--------------+
|age|workclass|   education|      relationship|         probability|predictedLabel|
+---+---------+------------+------------------+--------------------+--------------+
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|         9th|  Not-in-family|[0.96861782127659...|         <=50K|
| 18|        ?|        12th|         Own-child|[0.99562636676038...|         <=50K|
| 18|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 19|        ?|Some-college|         Own-child|[0.99562636676038...|         <=50K|
| 19|  Private|     HS-grad| Other-relative|[0.99562636676038...|         <=50K|
| 19|  Private|Some-college|         Own-child|[0.99562636676038...|         <=50K|
+---+---------+------------+------------------+--------------------+--------------+
only showing top 10 rows
Took 20 sec. Last updated by anonymous at December 01 2017, 12:19:48 PM.


Zeppelin
Notebook
Job
Search your Notes
anonymous
marketing-ml-demo02
Head
default
+---+---------+------------+------------------+--------------------+--------------+
|age|workclass|   education|      relationship|         probability|predictedLabel|
+---+---------+------------+------------------+--------------------+--------------+
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 17|  Private|         9th|  Not-in-family|[0.96861782127659...|         <=50K|
| 18|        ?|        12th|         Own-child|[0.99562636676038...|         <=50K|
| 18|  Private|        11th|         Own-child|[0.99562636676038...|         <=50K|
| 19|        ?|Some-college|         Own-child|[0.99562636676038...|         <=50K|
| 19|  Private|     HS-grad| Other-relative|[0.99562636676038...|         <=50K|
| 19|  Private|Some-college|         Own-child|[0.99562636676038...|         <=50K|
+---+---------+------------+------------------+--------------------+--------------+
only showing top 10 rows
Took 20 sec. Last updated by anonymous at December 01 2017, 12:19:48 PM.
%spark
// write the hi income individuals to s3
val pred_records = preds_fields.filter( col("predictedLabel").like(">50K") )
// df.filter(col("line").like("%ERROR%"))
pred_records.show(10)
// .write.format("json").save("s3a://...")
FINISHED
%spark
READY

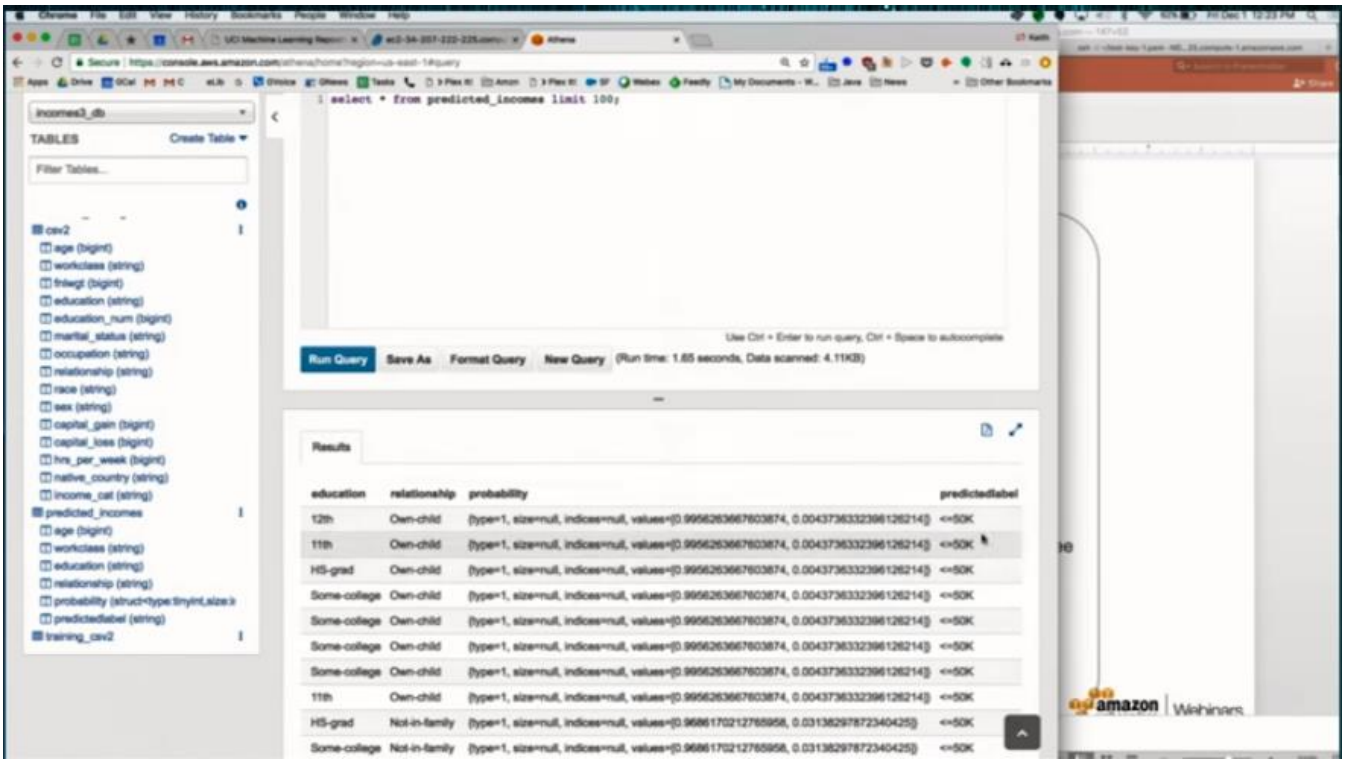



We can now use Athena to query the result data in S3

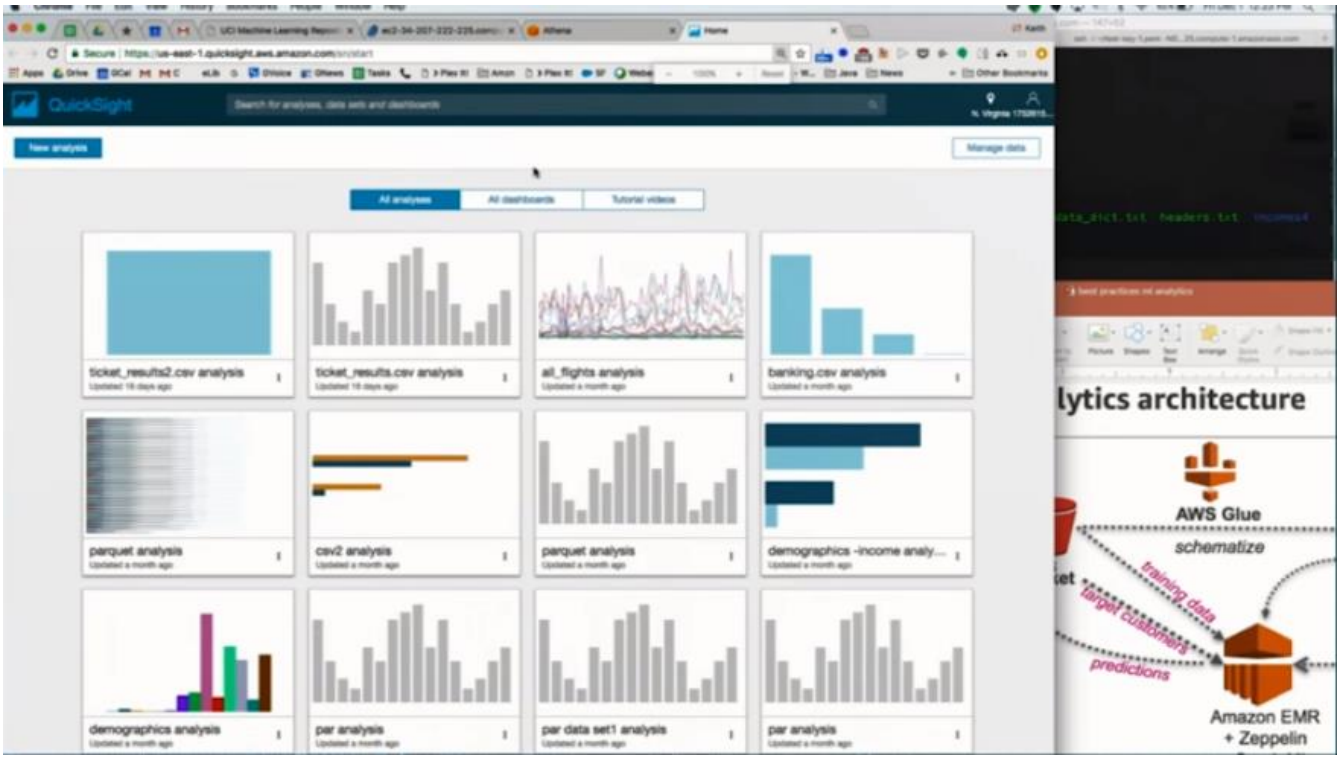


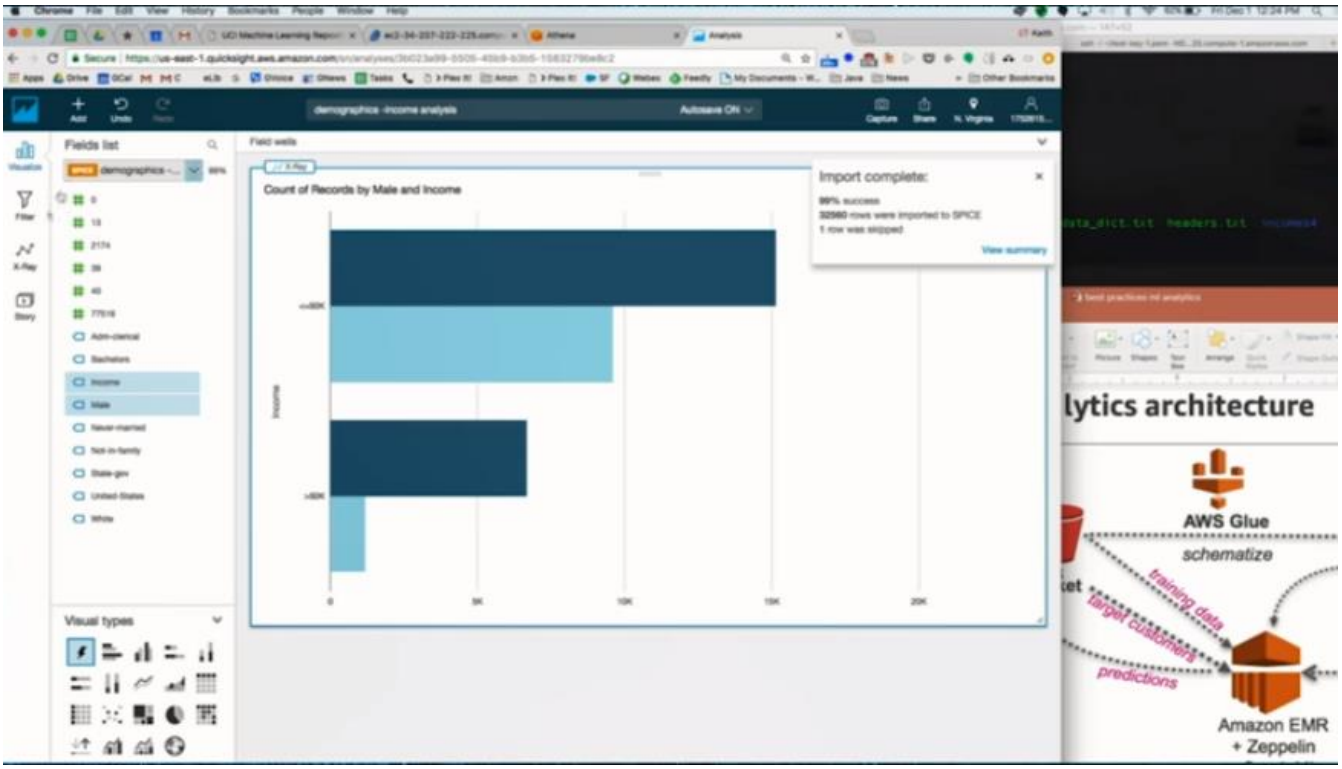



We can now write a query against the original CSV data from the census site

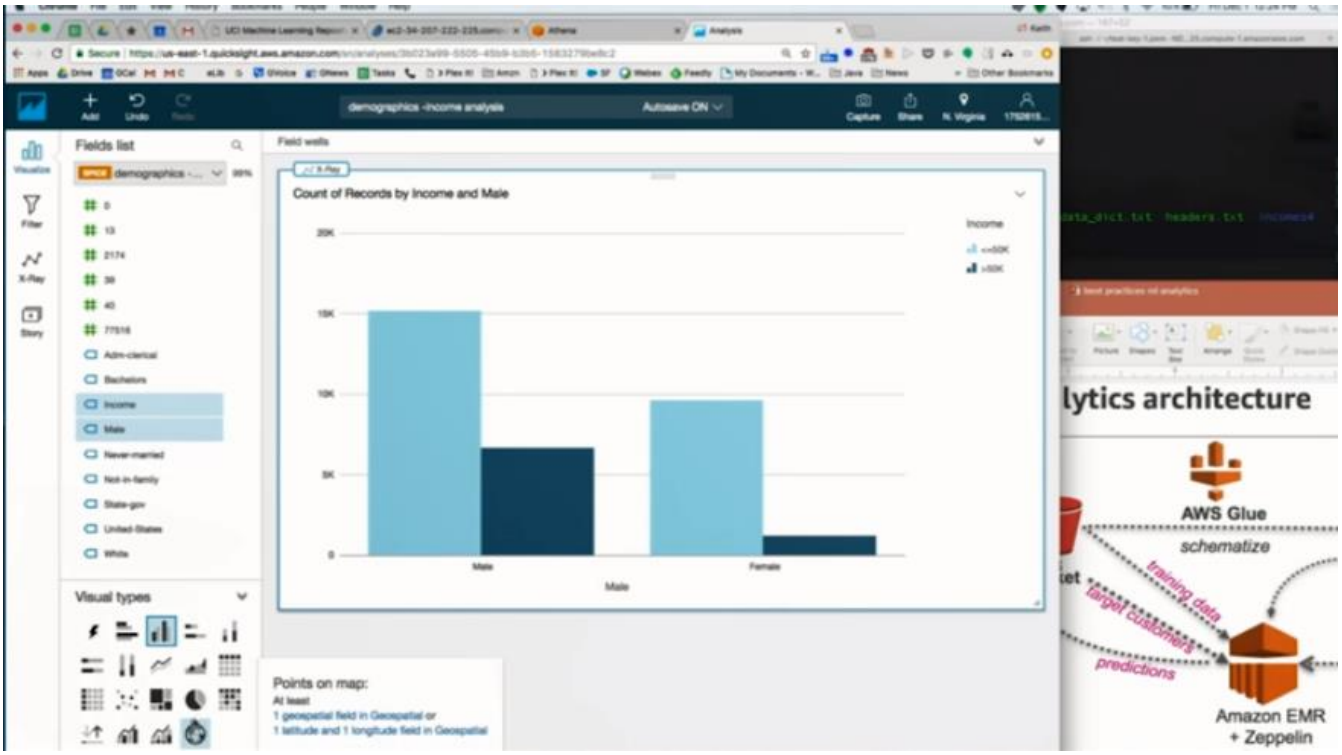


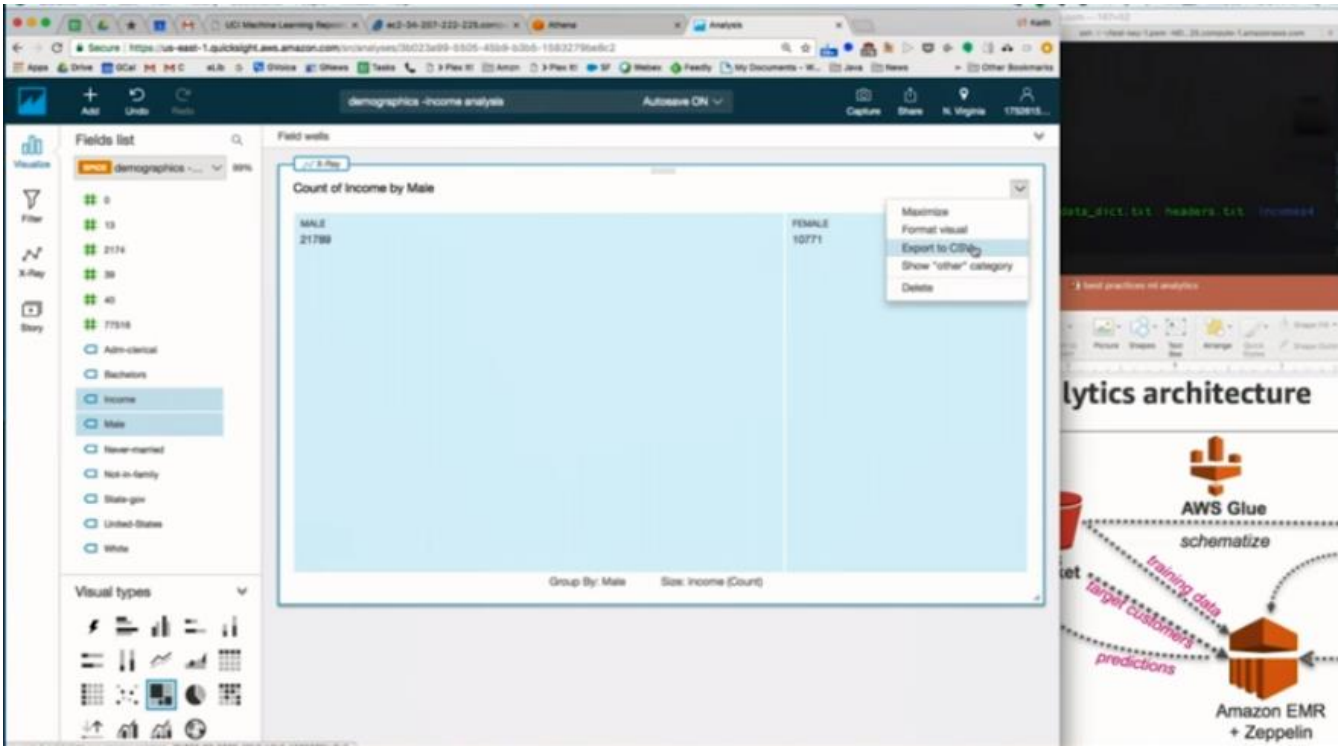

We then run a query against the new predicted_table data from the ML process too as above




We then use QuickSight to pull the S3 data into the SPICE engine for visualizations like above to share


demographics -income analysis
Autosave ON
Fields list
Count of Records by Income and Male
Income
Male
Female
Visual types
Points on map:
At least
1 geospatial field in Geospatial or
1 latitude and 1 longitude field in Geospatial
lytics architecture
AWS Glue
schematize
training data
target customers
predictions
Amazon EMR
+ Zeppelin


demographics -income analysis
Autosave ON
Fields list
Count of Income by Male
MALE
21789
FEMALE
10771
Maximize
Format visual
Export to CSV
Show "other" category
Delete
Group By: Male
Size: Income (Count)
Visual types
lytics architecture
AWS Glue
schematize
training data
target customers
predictions
Amazon EMR
+ Zeppelin