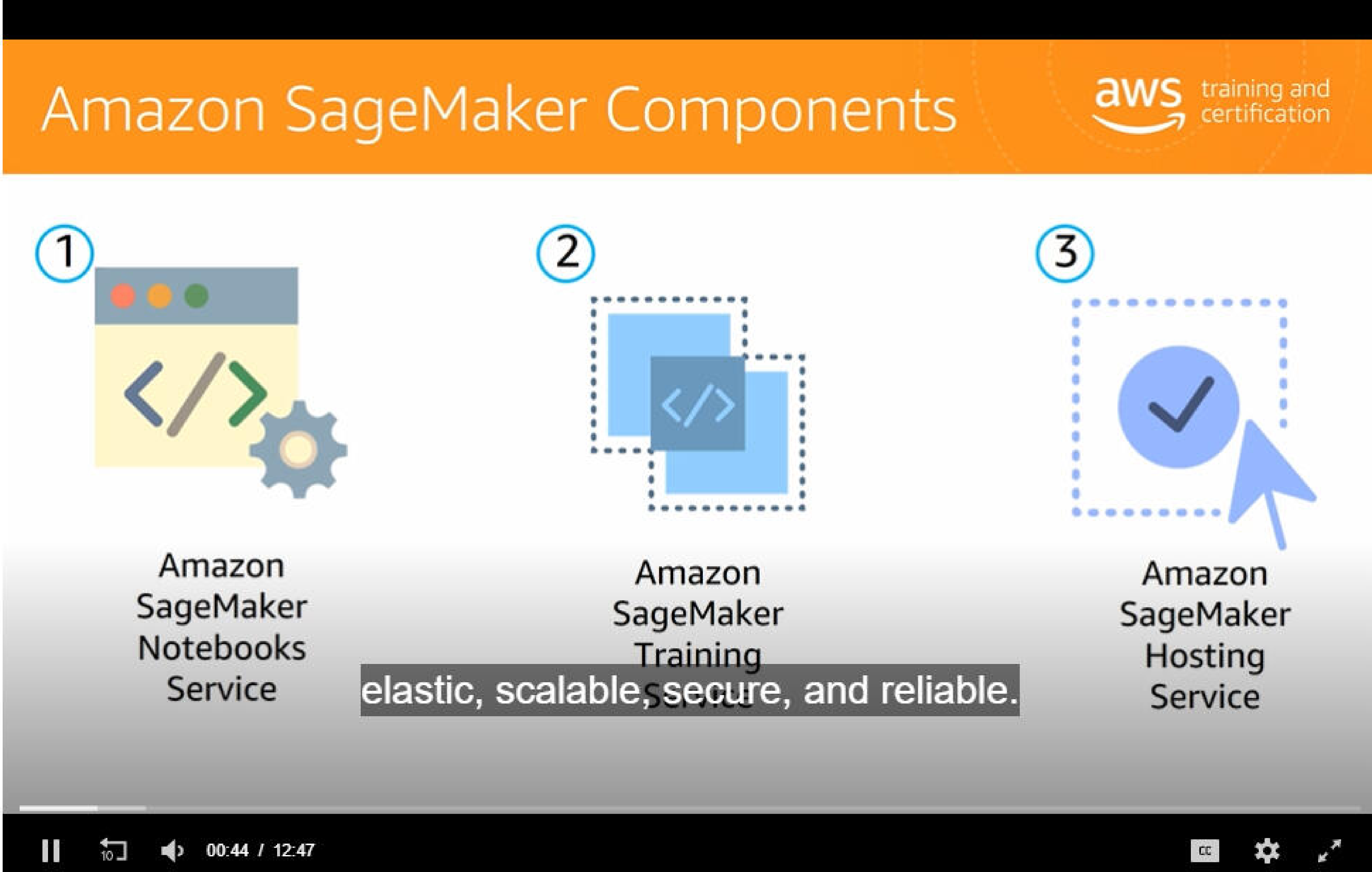

Amazon SageMaker Components
aws training and certification
1
Amazon SageMaker Notebooks Service
2
Amazon SageMaker Training Service
3
Amazon SageMaker Hosting Service
elastic, scalable, secure, and reliable.
00:44 / 12:47

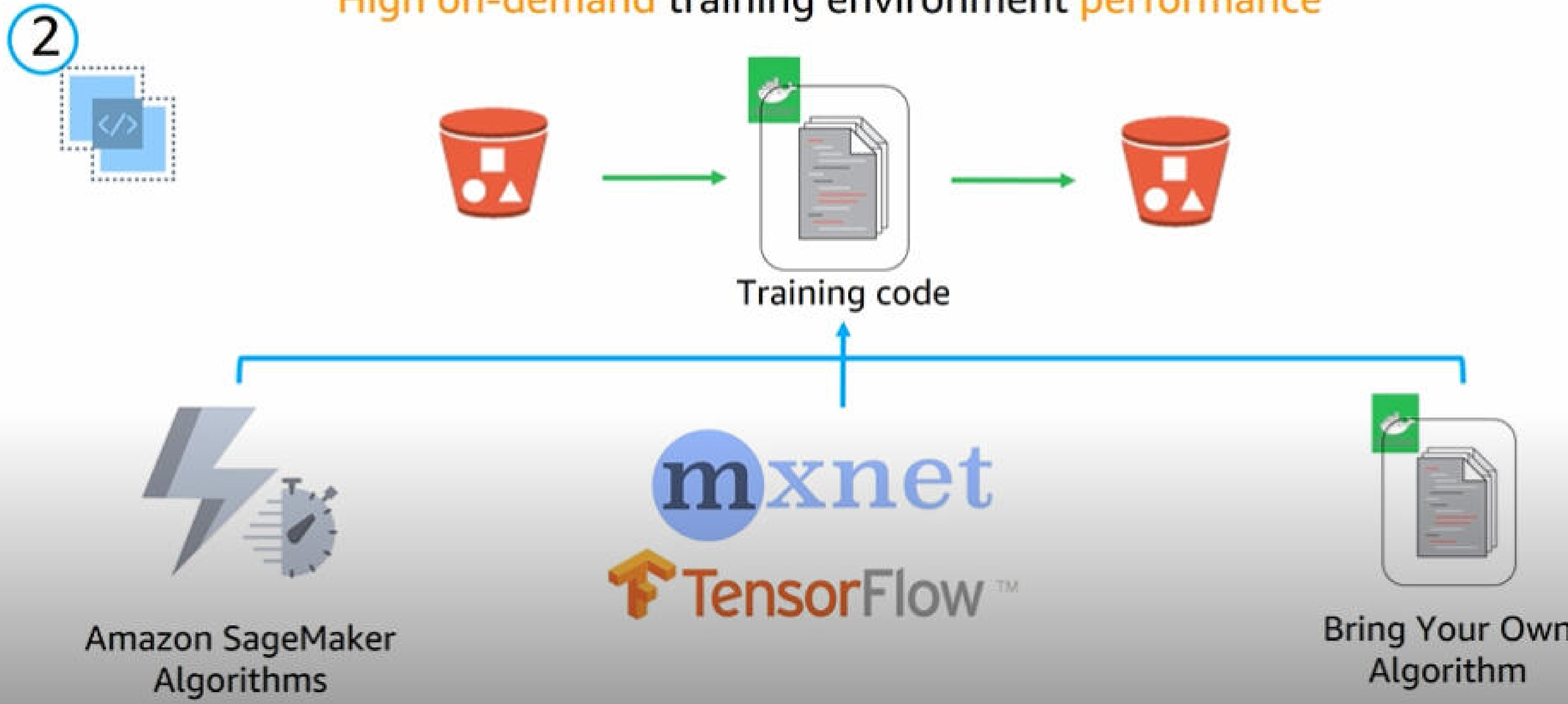
Zero Setup for Exploratory Data Analysis

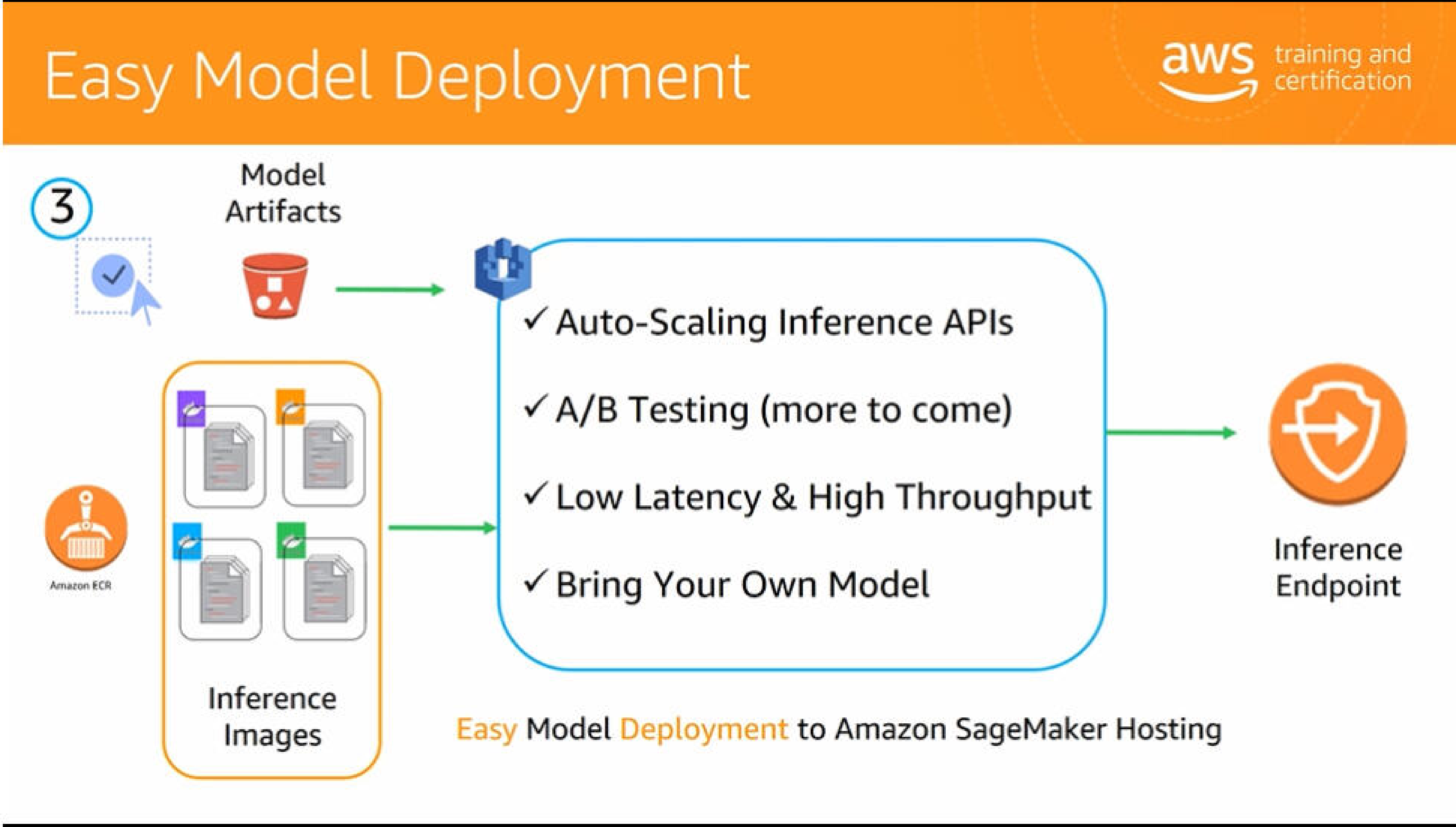
aws training and certification

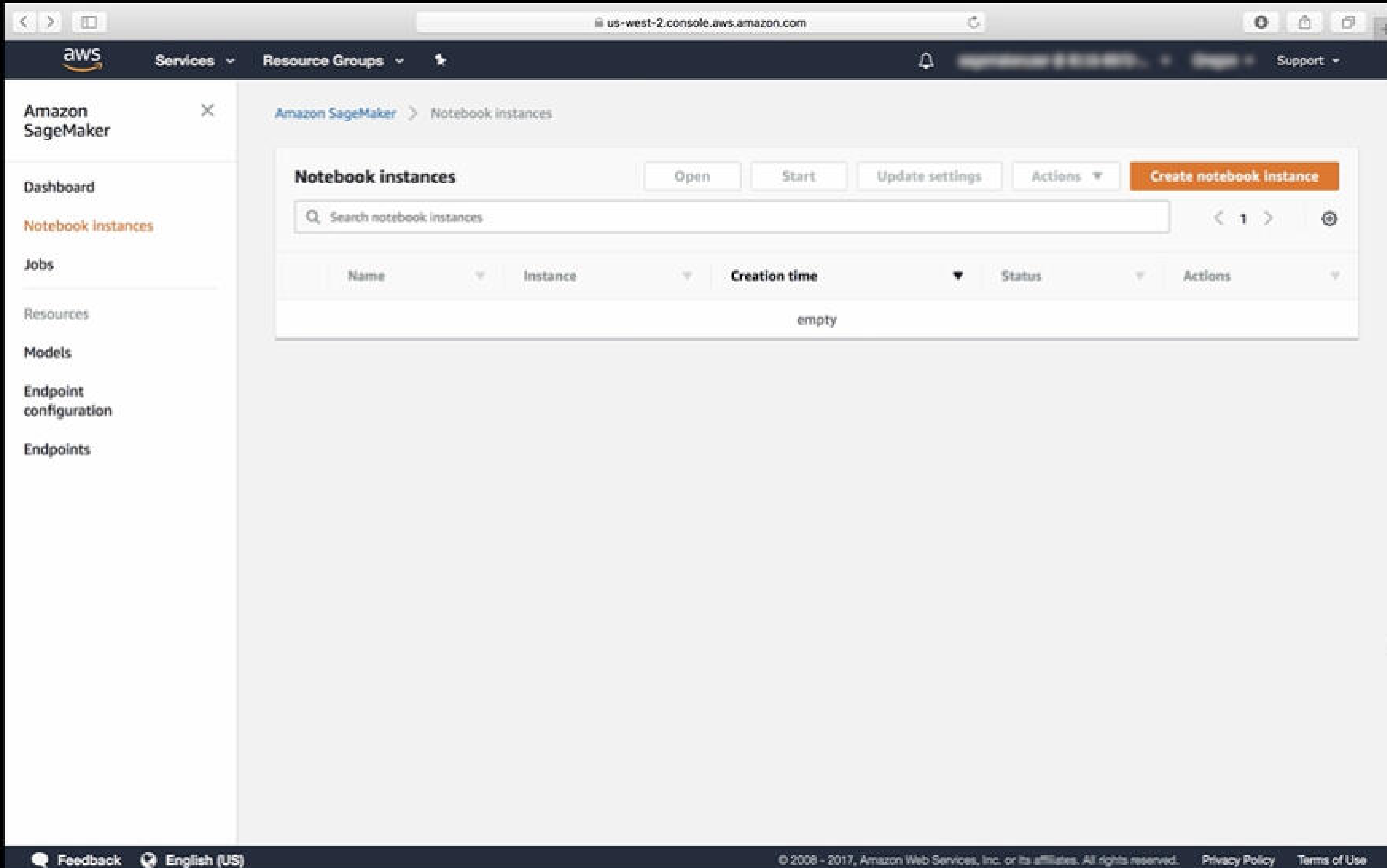
01:04 / 12:47

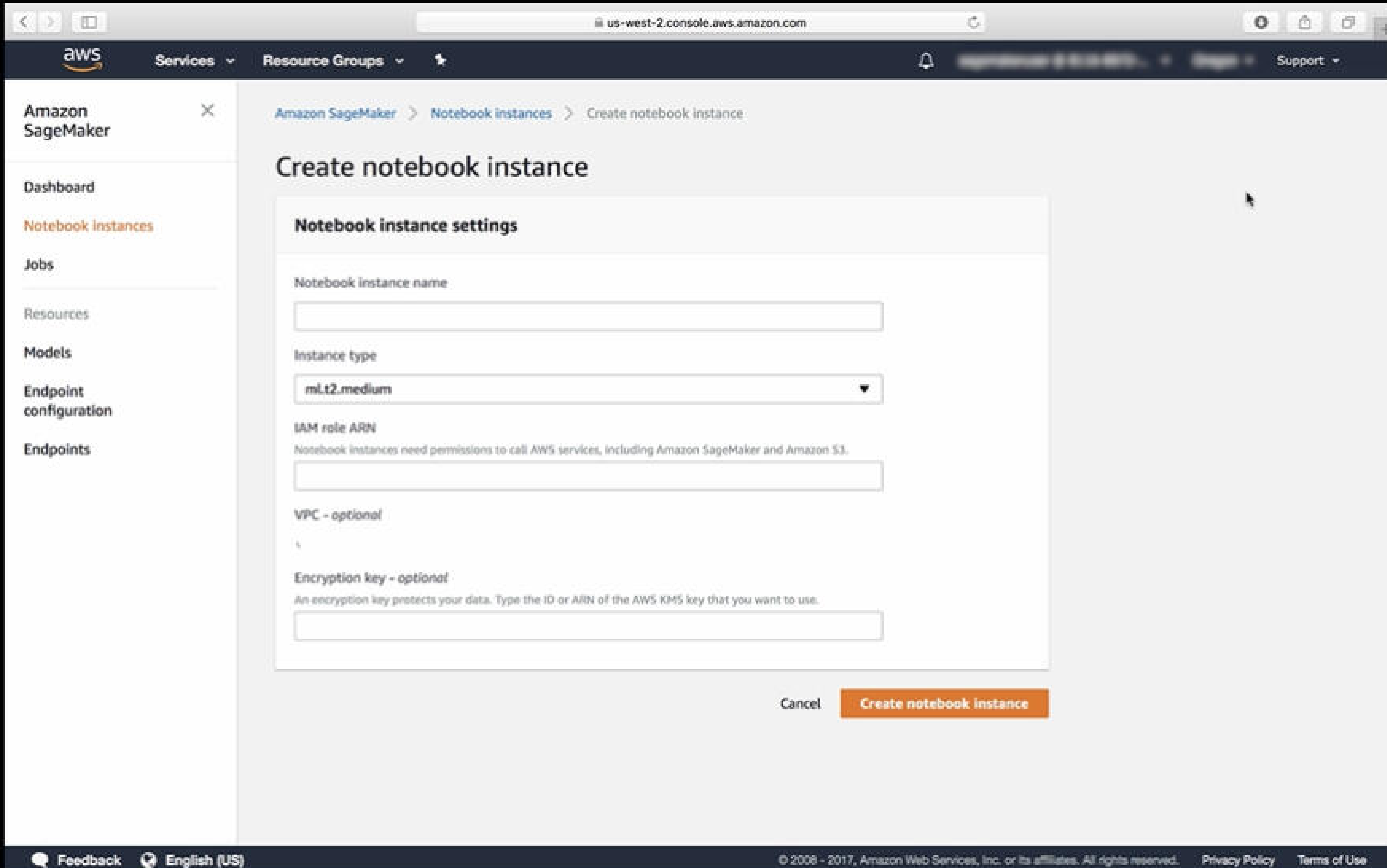
High Performance, On-Demand

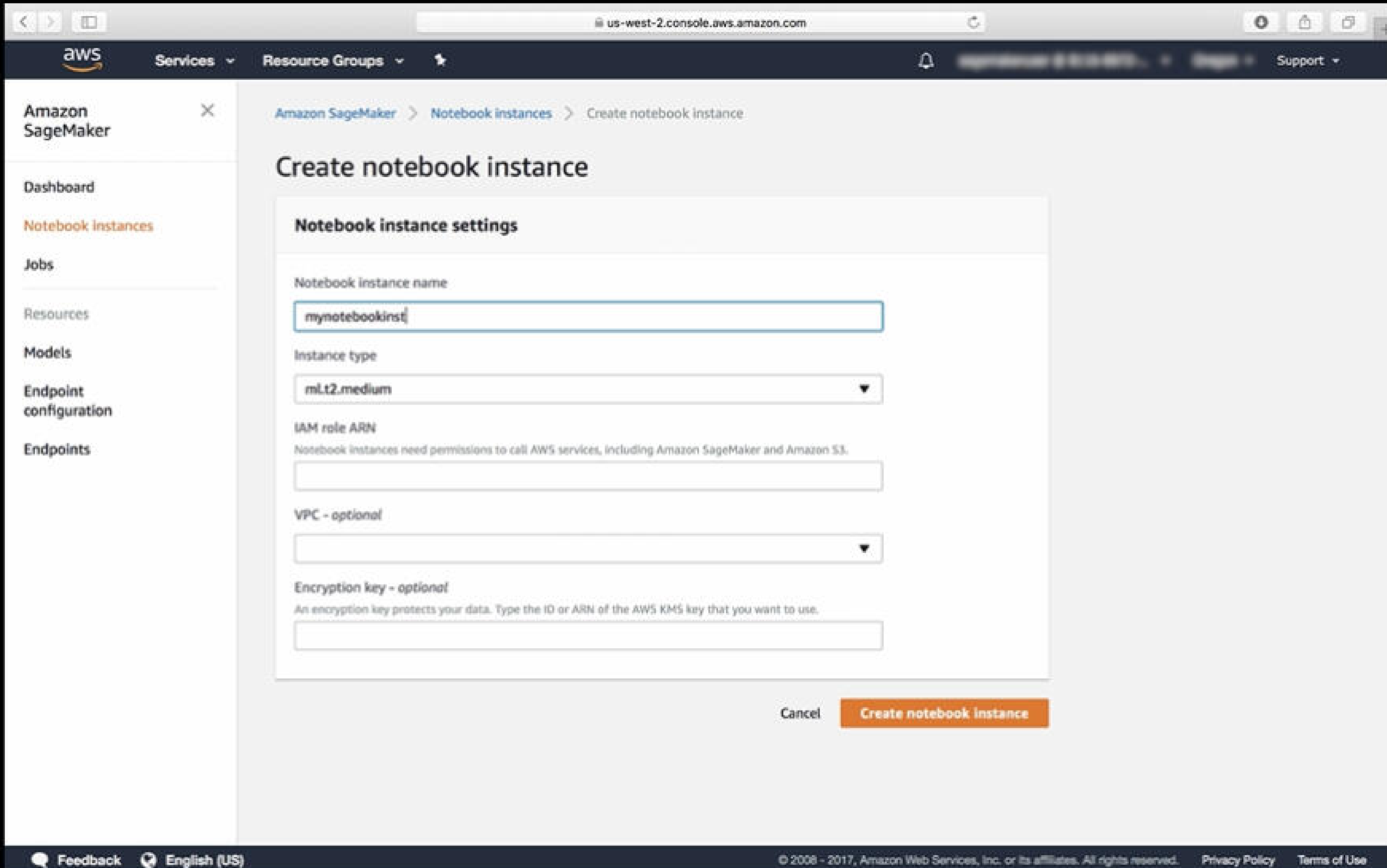
High on-demand training environment performance

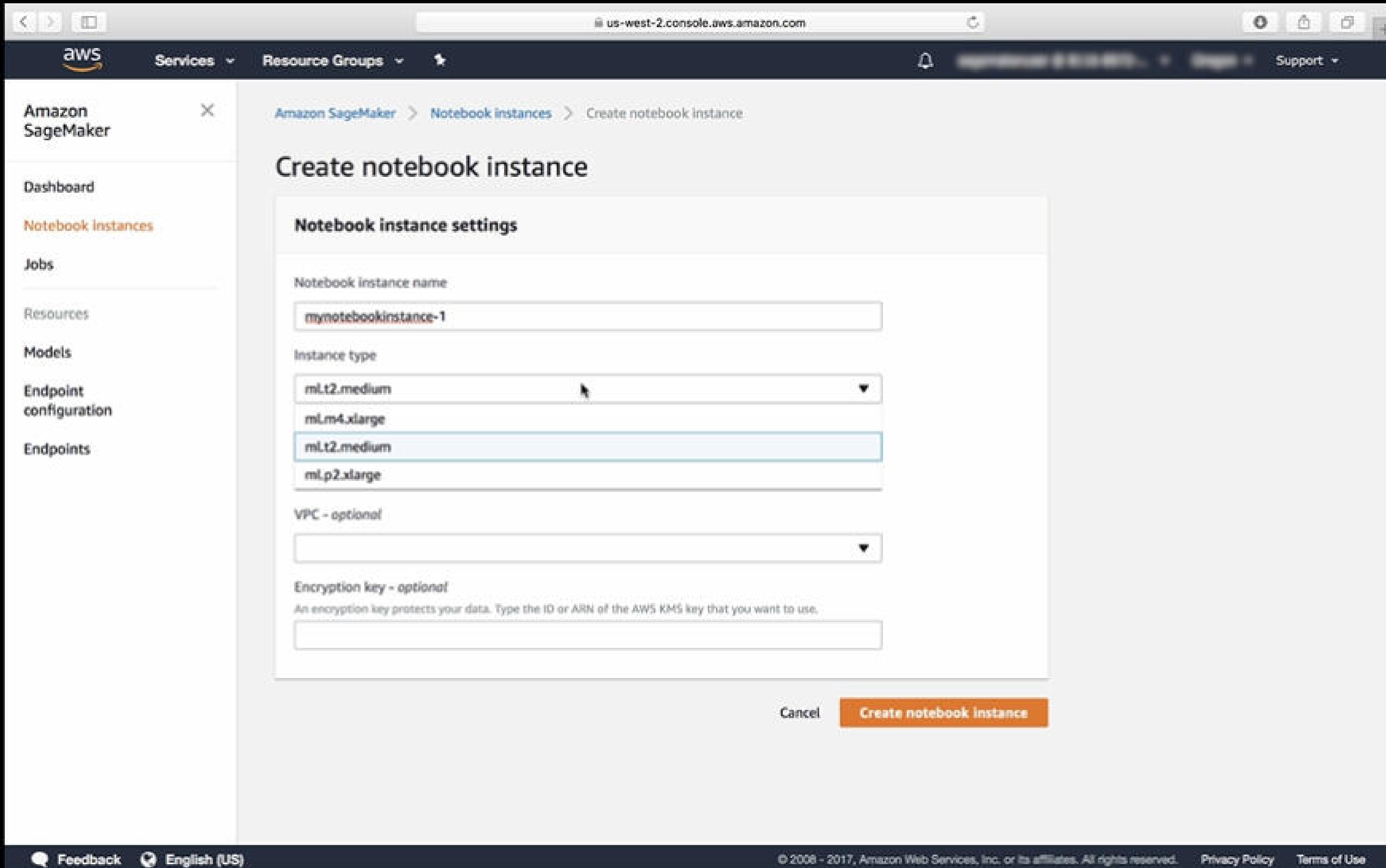


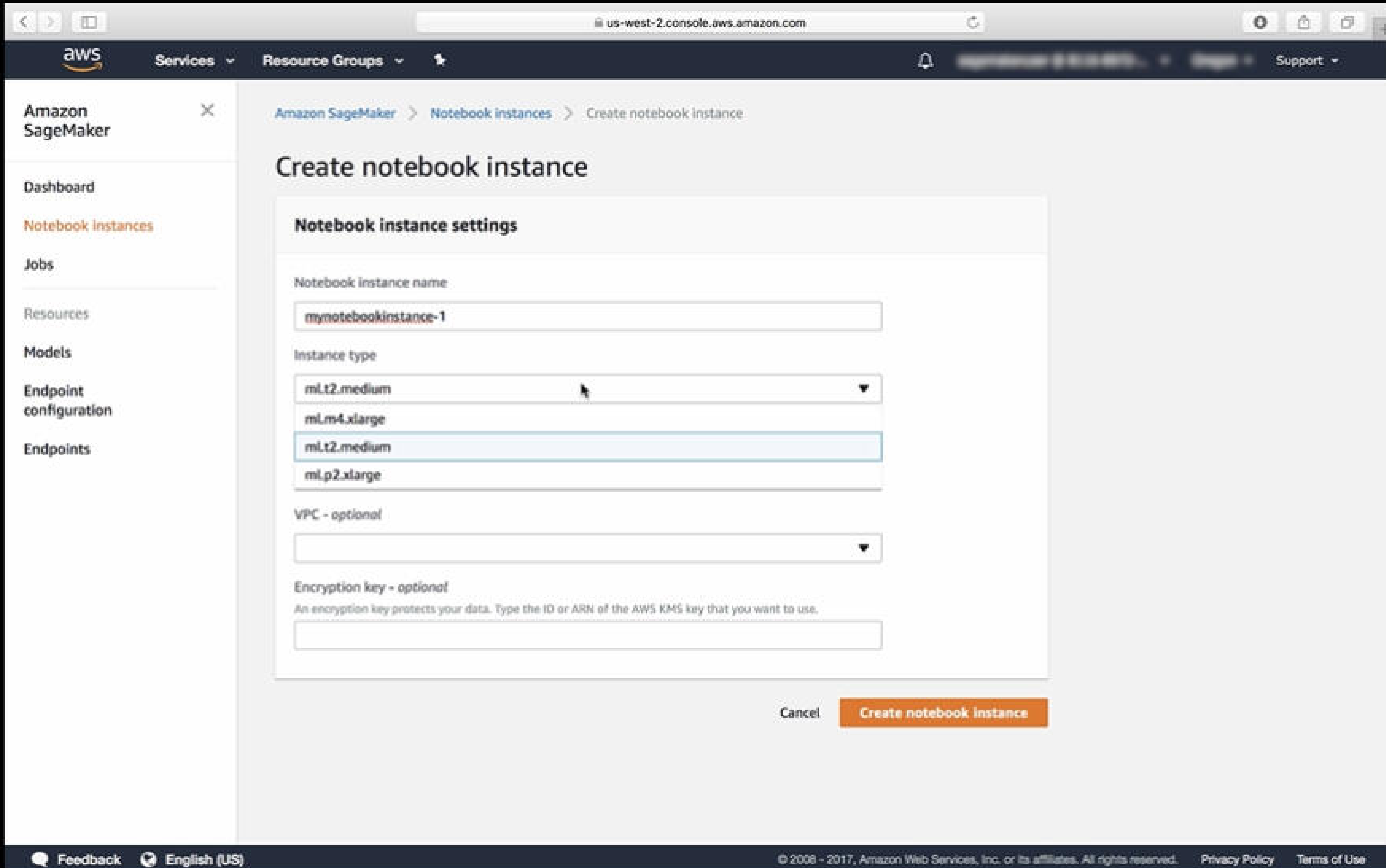

Easy Model Deployment
aws training and certification
3
Model Artifacts
✓ Auto-Scaling Inference APIs
✓ A/B Testing (more to come)
✓ Low Latency & High Throughput
✓ Bring Your Own Model
Amazon ECR
Inference Images
Inference Endpoint
Easy Model Deployment to Amazon SageMaker Hosting

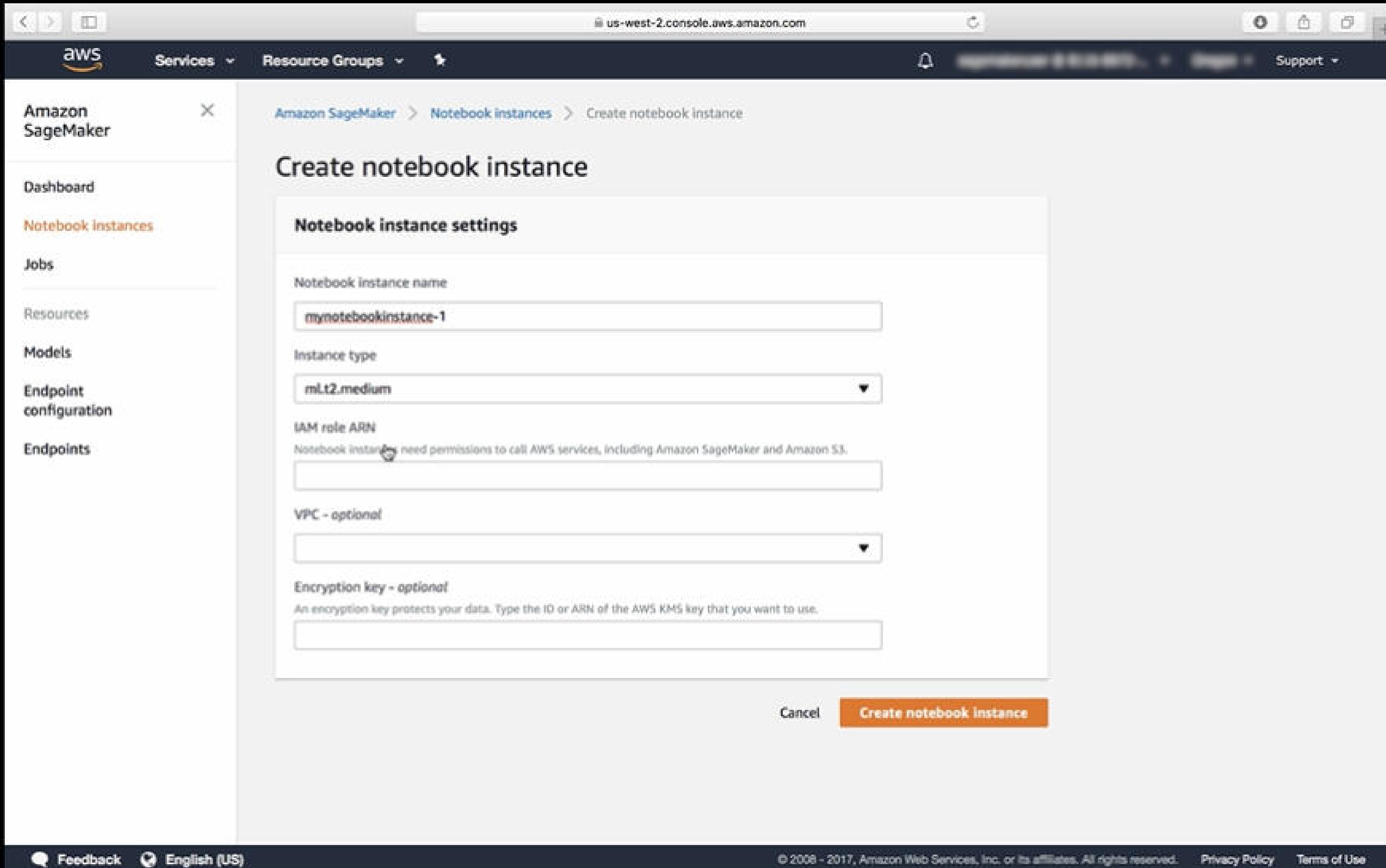

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances
Notebook instances
Open
Start
Update settings
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
empty
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

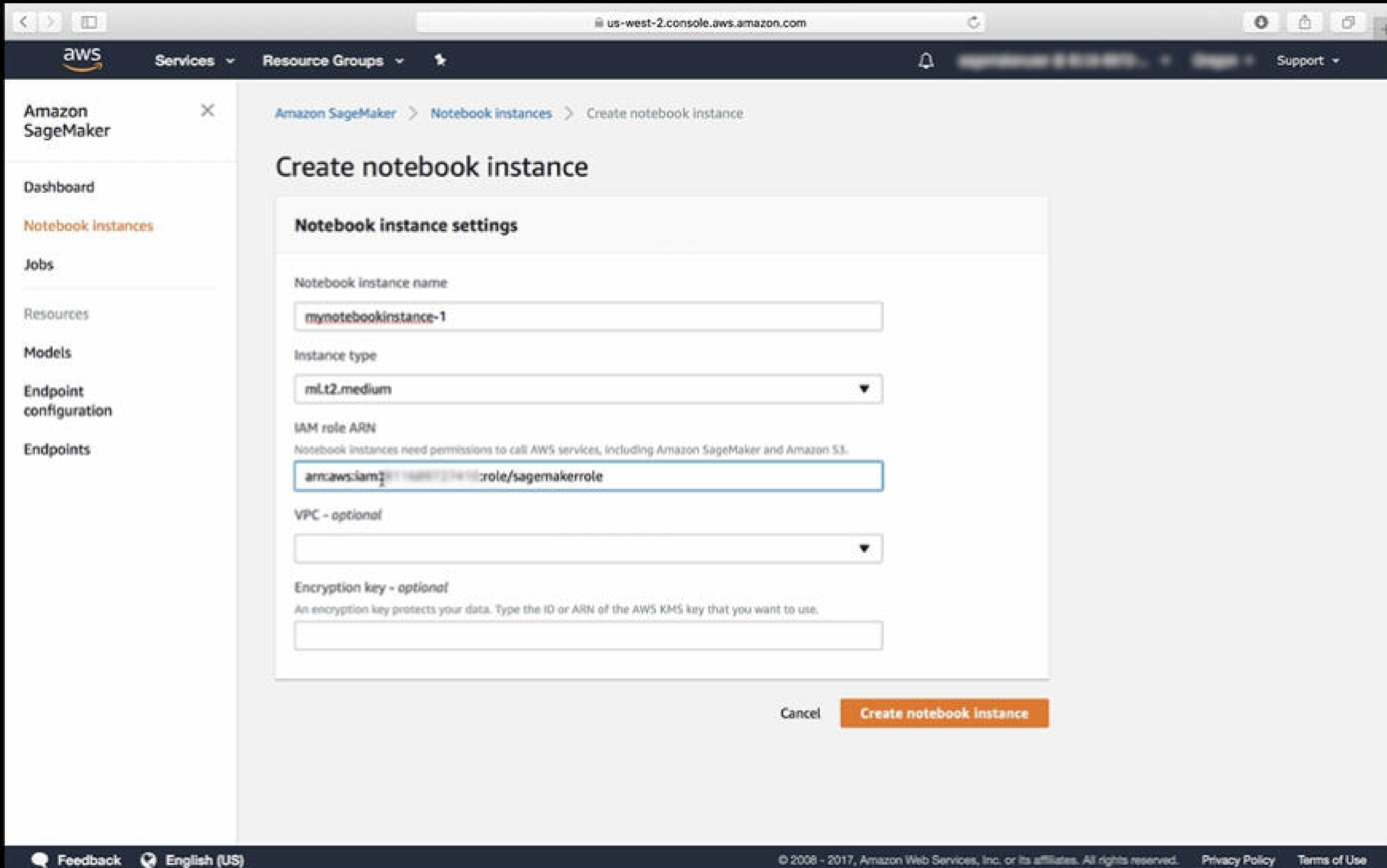

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
VPC - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

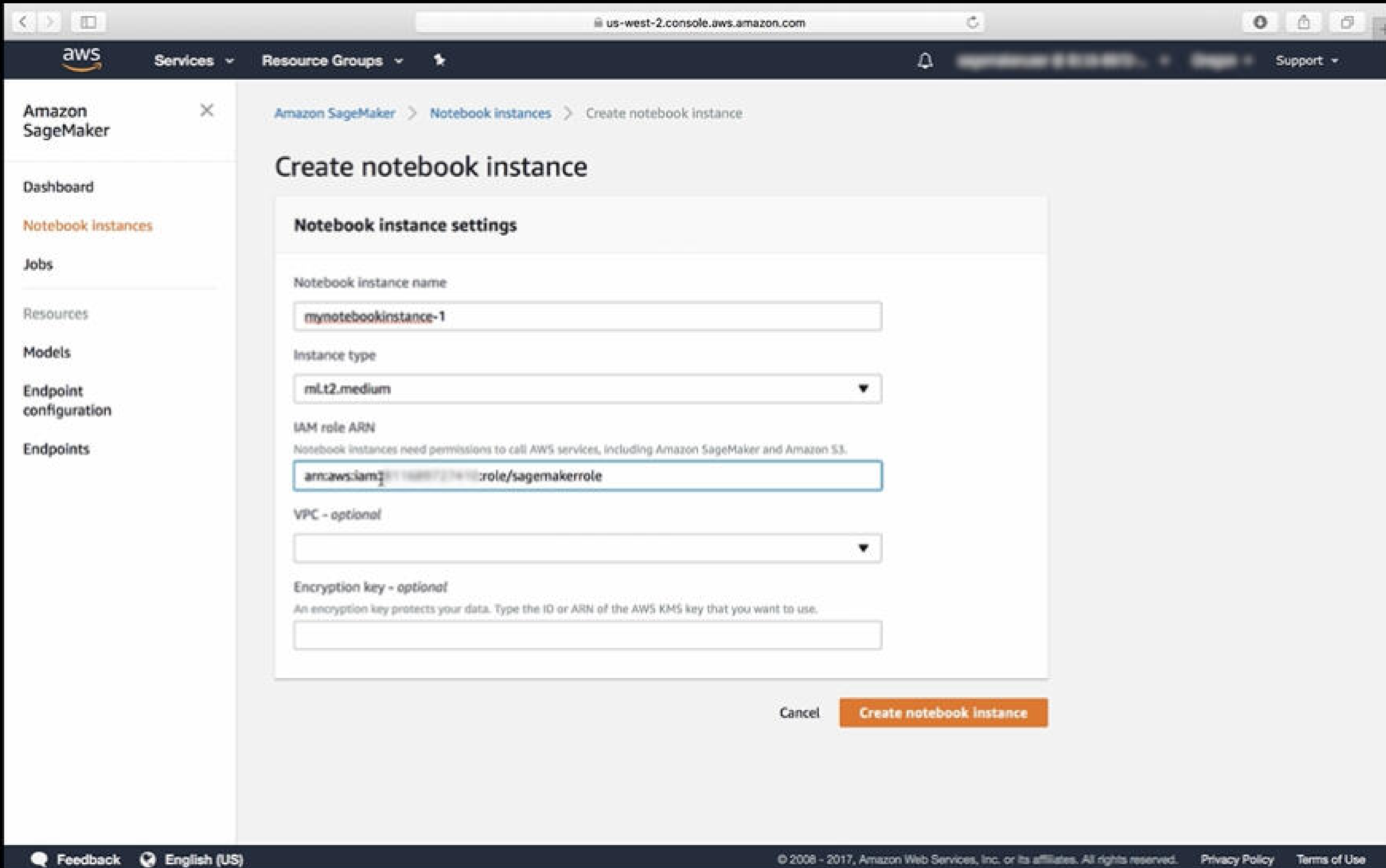

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinst
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
VPC - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

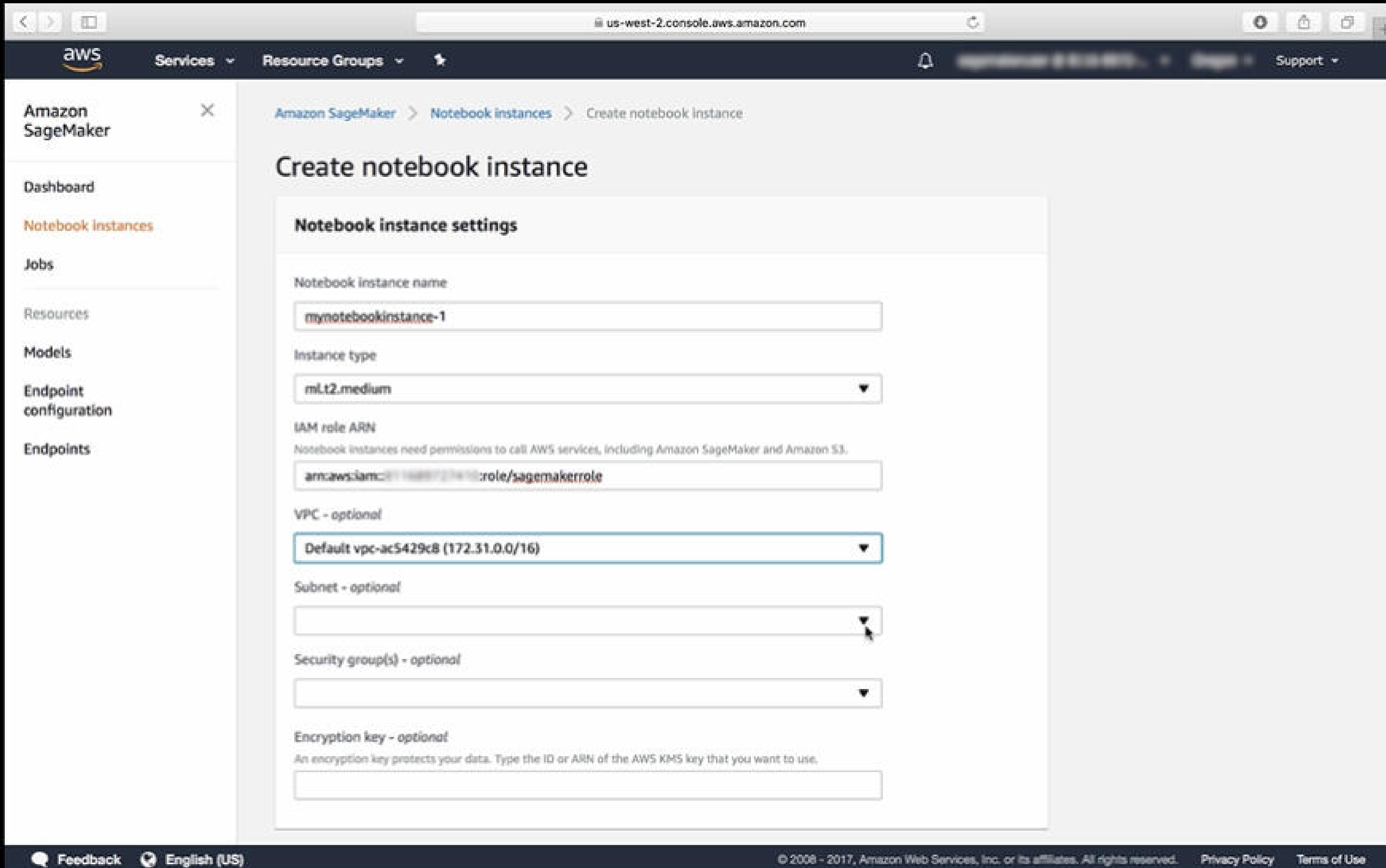

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
ml.m4.xlarge
ml.t2.medium
ml.p2.xlarge
VPC - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
ml.m4.xlarge
ml.t2.medium
ml.p2.xlarge
VPC - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

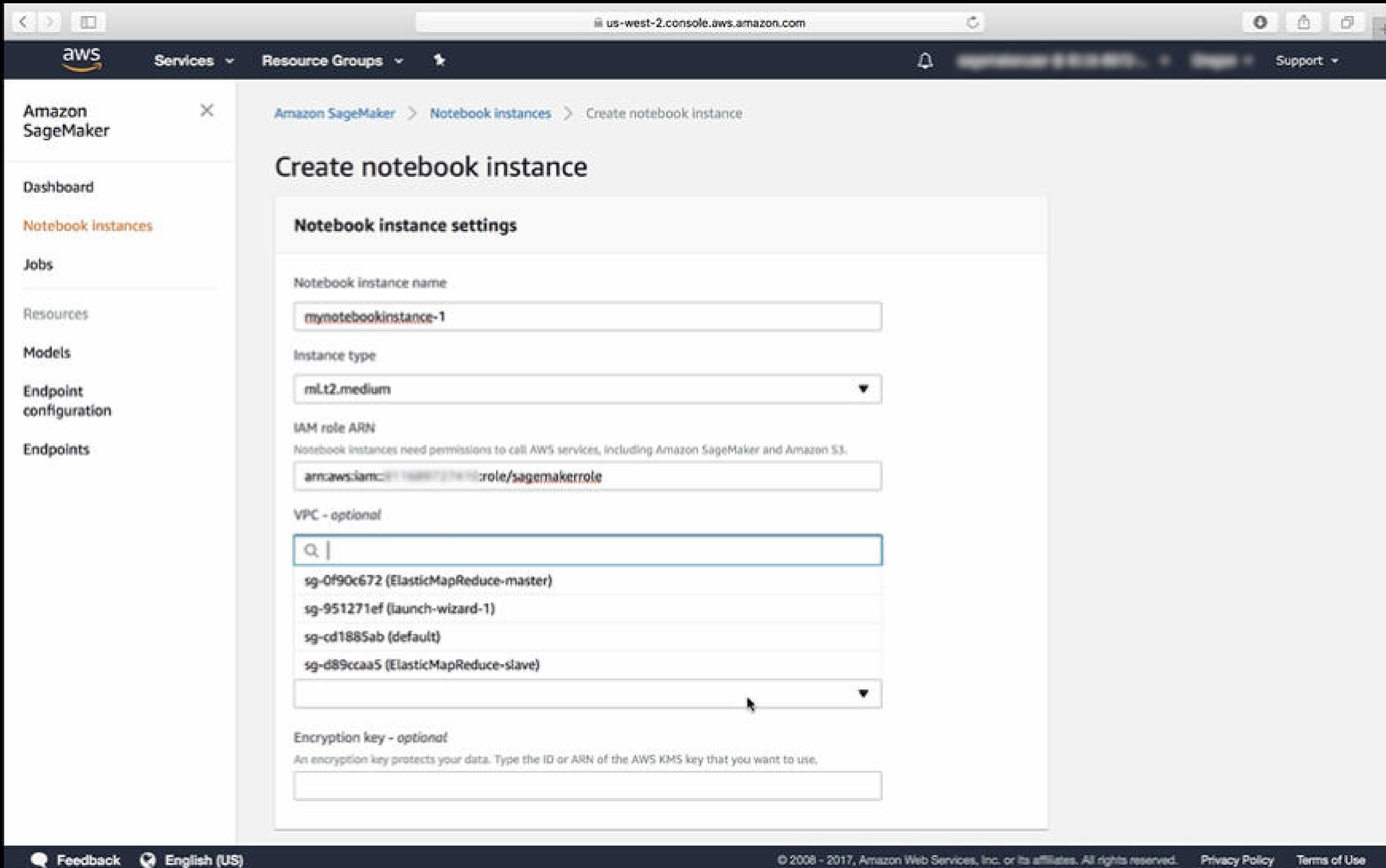

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
VPC - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

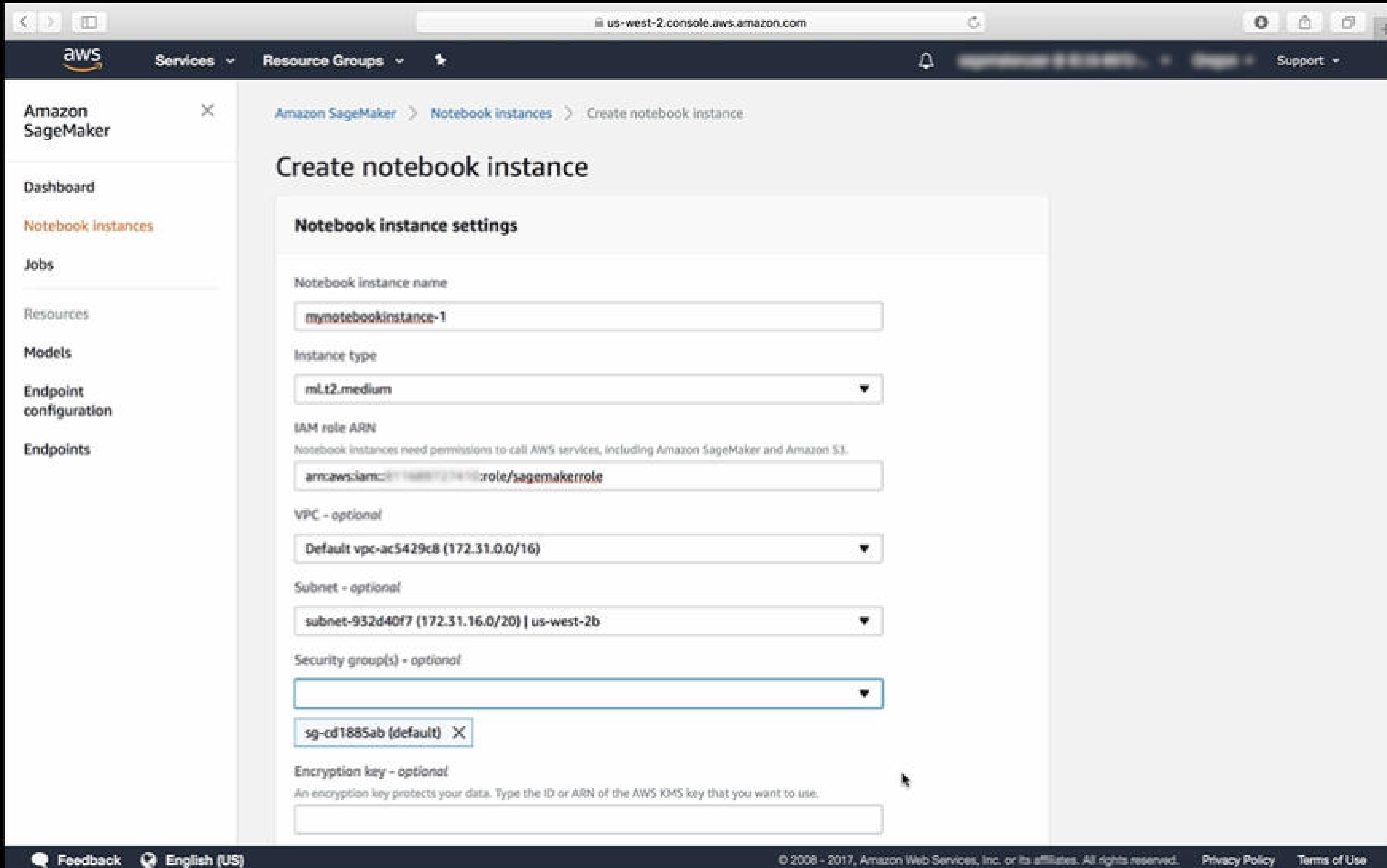

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
role/sagemakerrole
VPC - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

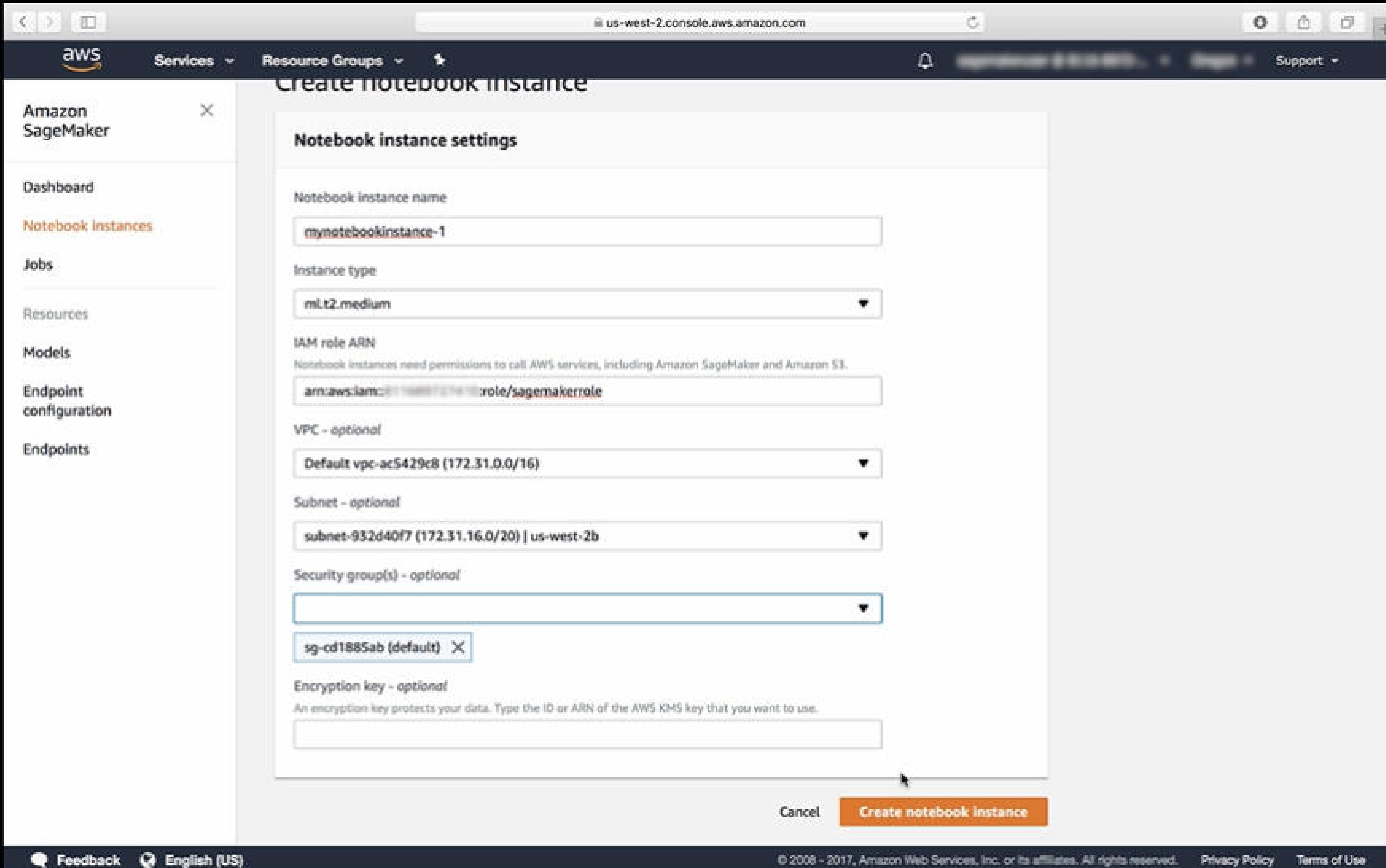

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
role/sagemakerrole
VPC - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



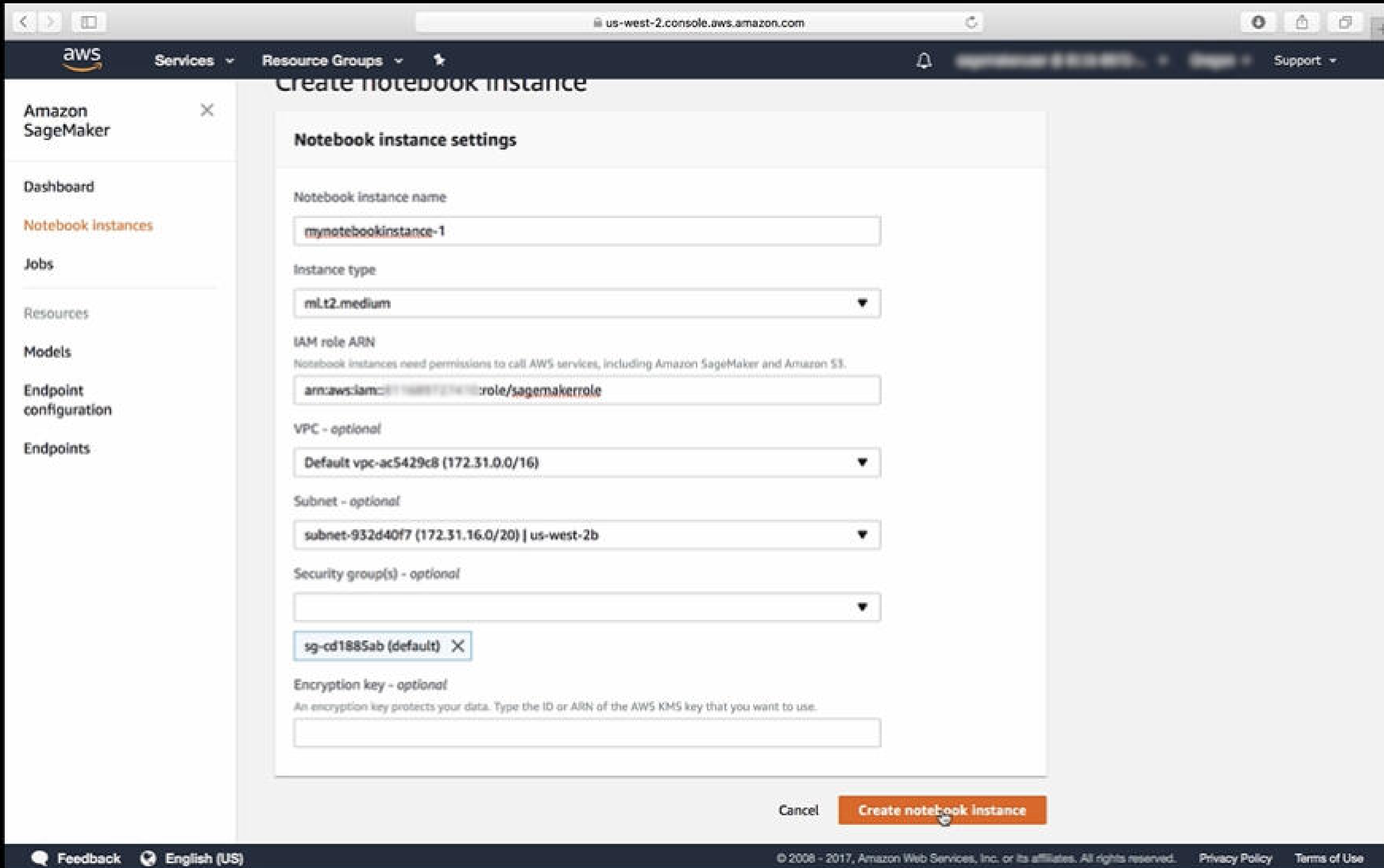
us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
arn:aws:iam:
:role/sagemakerrole
VPC - optional
Default vpc-ac5429c8 (172.31.0.0/16)
Subnet - optional
Security group(s) - optional
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

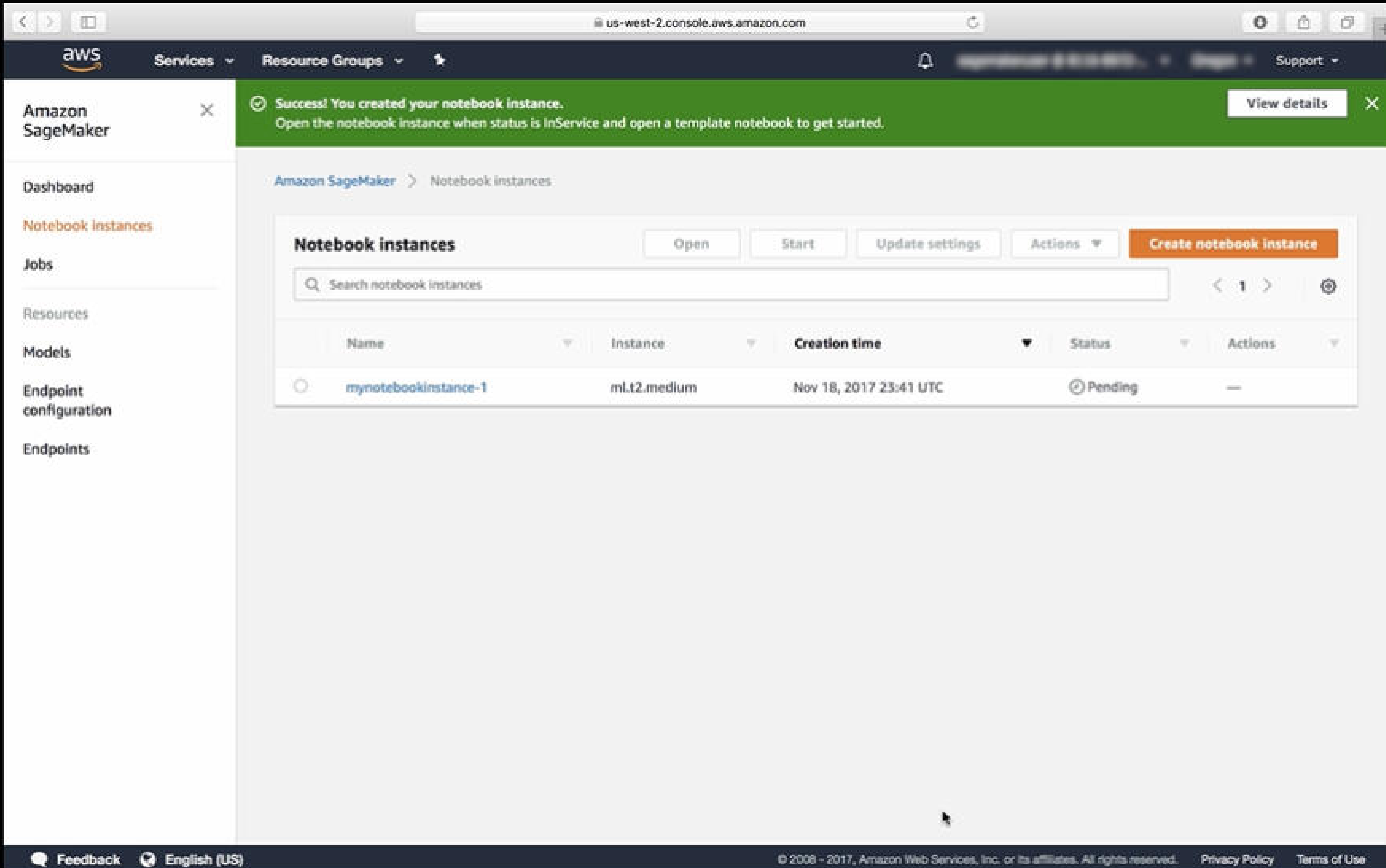

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
role/sagemakerrole
VPC - optional
sg-0f90c672 (ElasticMapReduce-master)
sg-951271ef (launch-wizard-1)
sg-cd1885ab (default)
sg-d89ccaa5 (ElasticMapReduce-slave)
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances > Create notebook instance
Create notebook instance
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
arn:aws:iam: :role/sagemakerrole
VPC - optional
Default vpc-ac5429c8 (172.31.0.0/16)
Subnet - optional
subnet-932d40f7 (172.31.16.0/20) | us-west-2b
Security group(s) - optional
sg-cd1885ab (default)
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

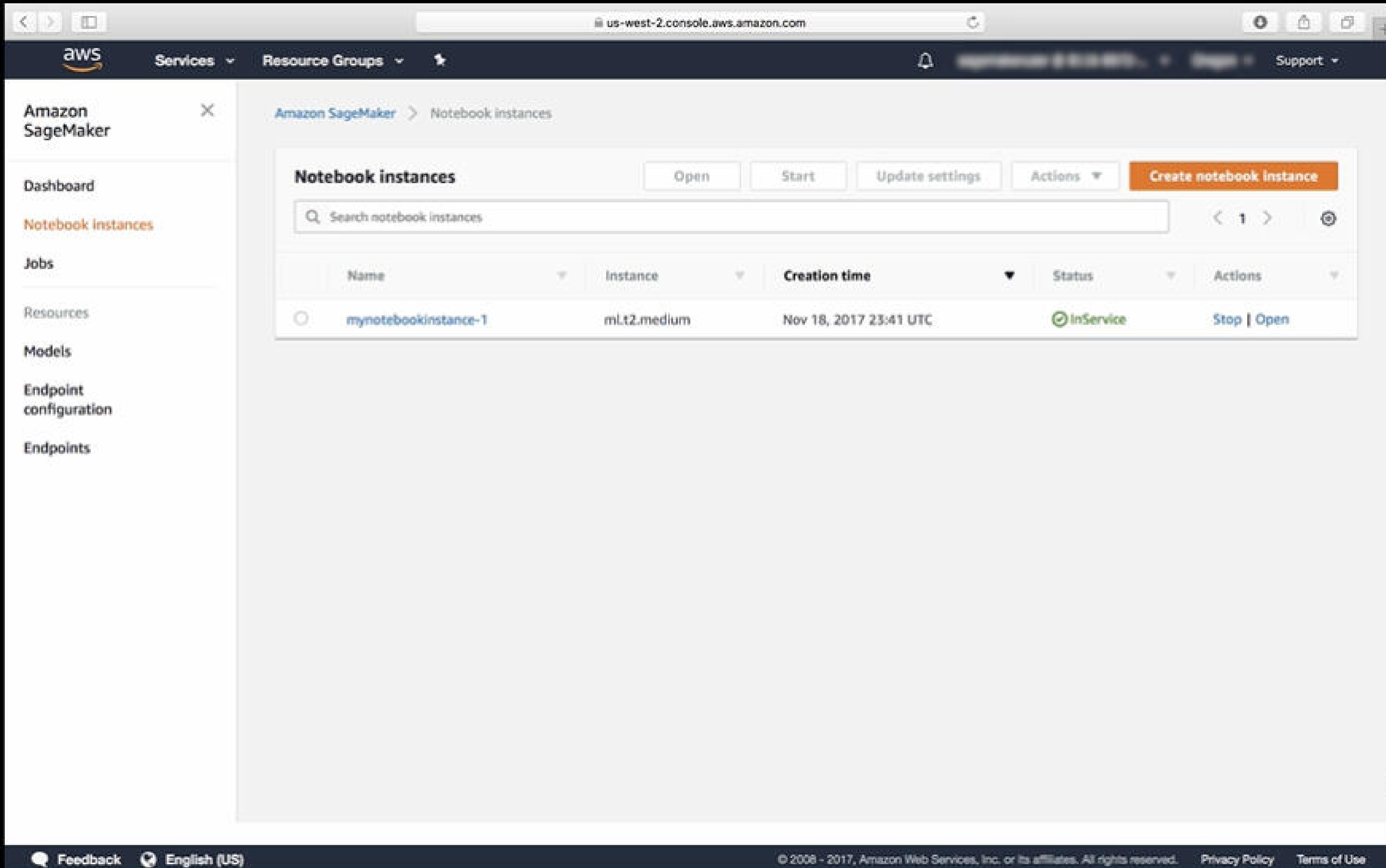


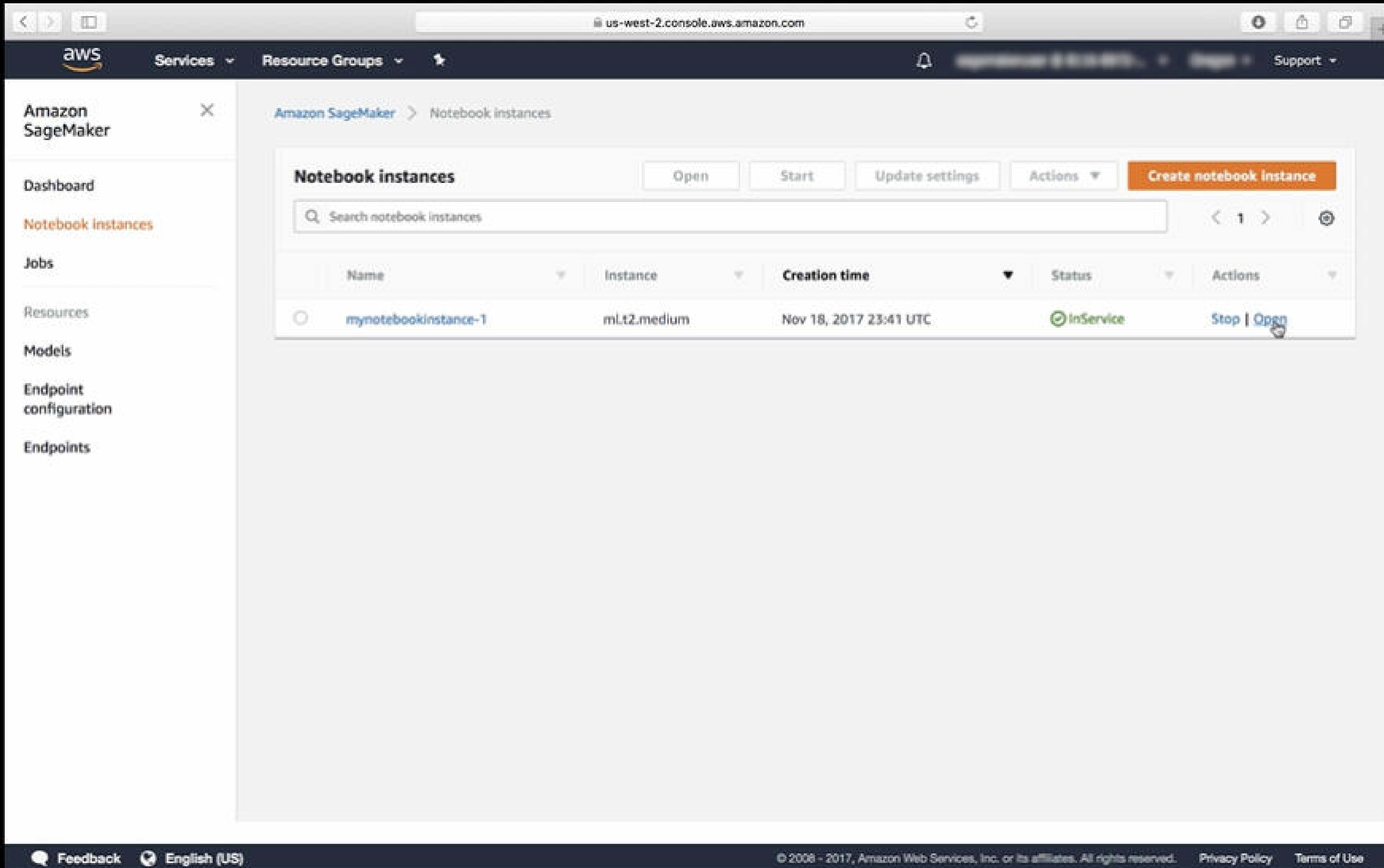
us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Create notebook instance
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
arn:aws:iam: :role/sagemakerrole
VPC - optional
Default vpc-ac5429c8 (172.31.0.0/16)
Subnet - optional
subnet-932d40f7 (172.31.16.0/20) | us-west-2b
Security group(s) - optional
sg-cd1885ab (default)
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

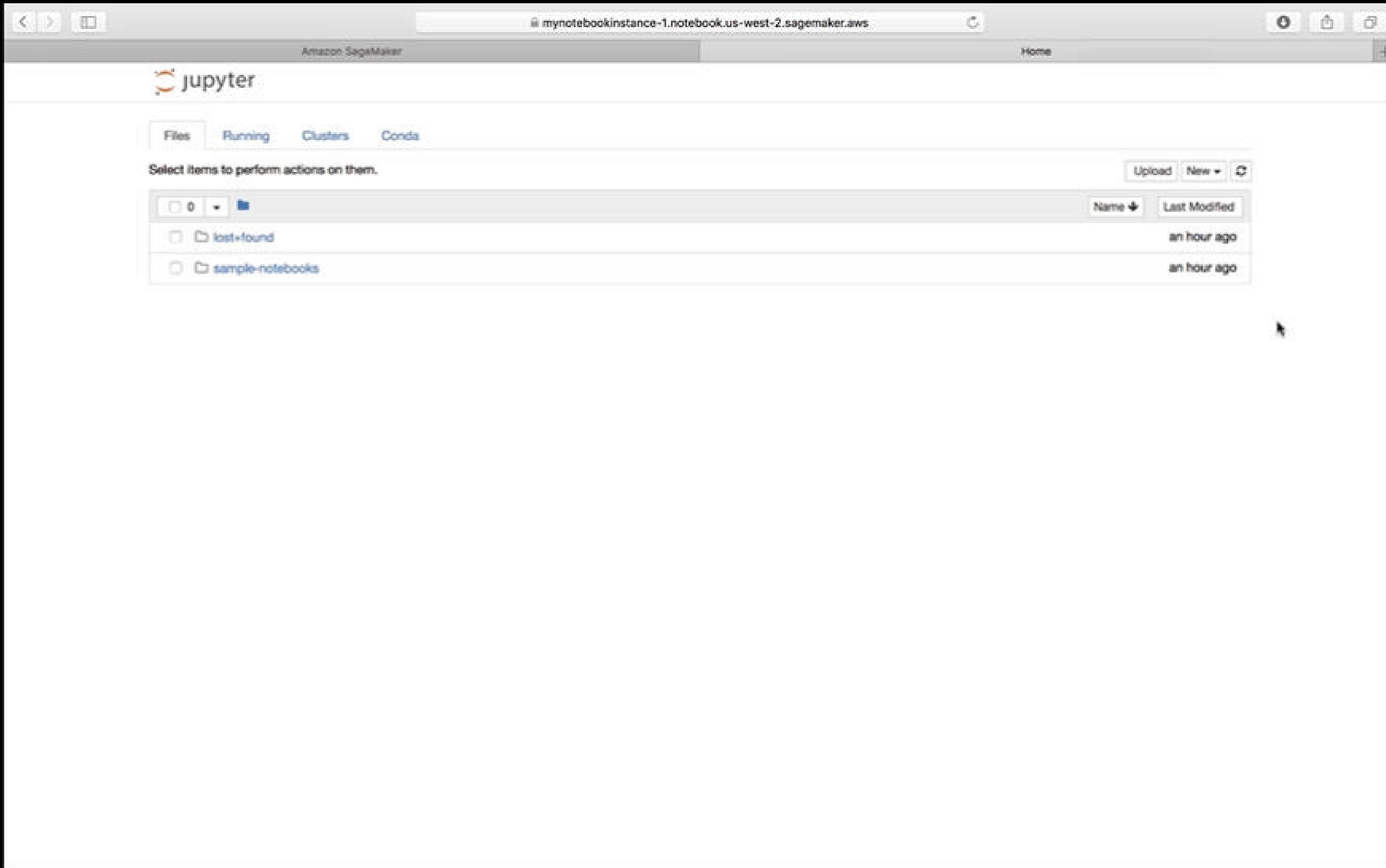

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Create notebook instance
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Notebook instance settings
Notebook instance name
mynotebookinstance-1
Instance type
ml.t2.medium
IAM role ARN
Notebook instances need permissions to call AWS services, including Amazon SageMaker and Amazon S3.
arn:aws:iam: :role/sagemakerrole
VPC - optional
Default vpc-ac5429c8 (172.31.0.0/16)
Subnet - optional
subnet-932d40f7 (172.31.16.0/20) | us-west-2b
Security group(s) - optional
sg-cd1885ab (default)
Encryption key - optional
An encryption key protects your data. Type the ID or ARN of the AWS KMS key that you want to use.
Cancel
Create notebook instance
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

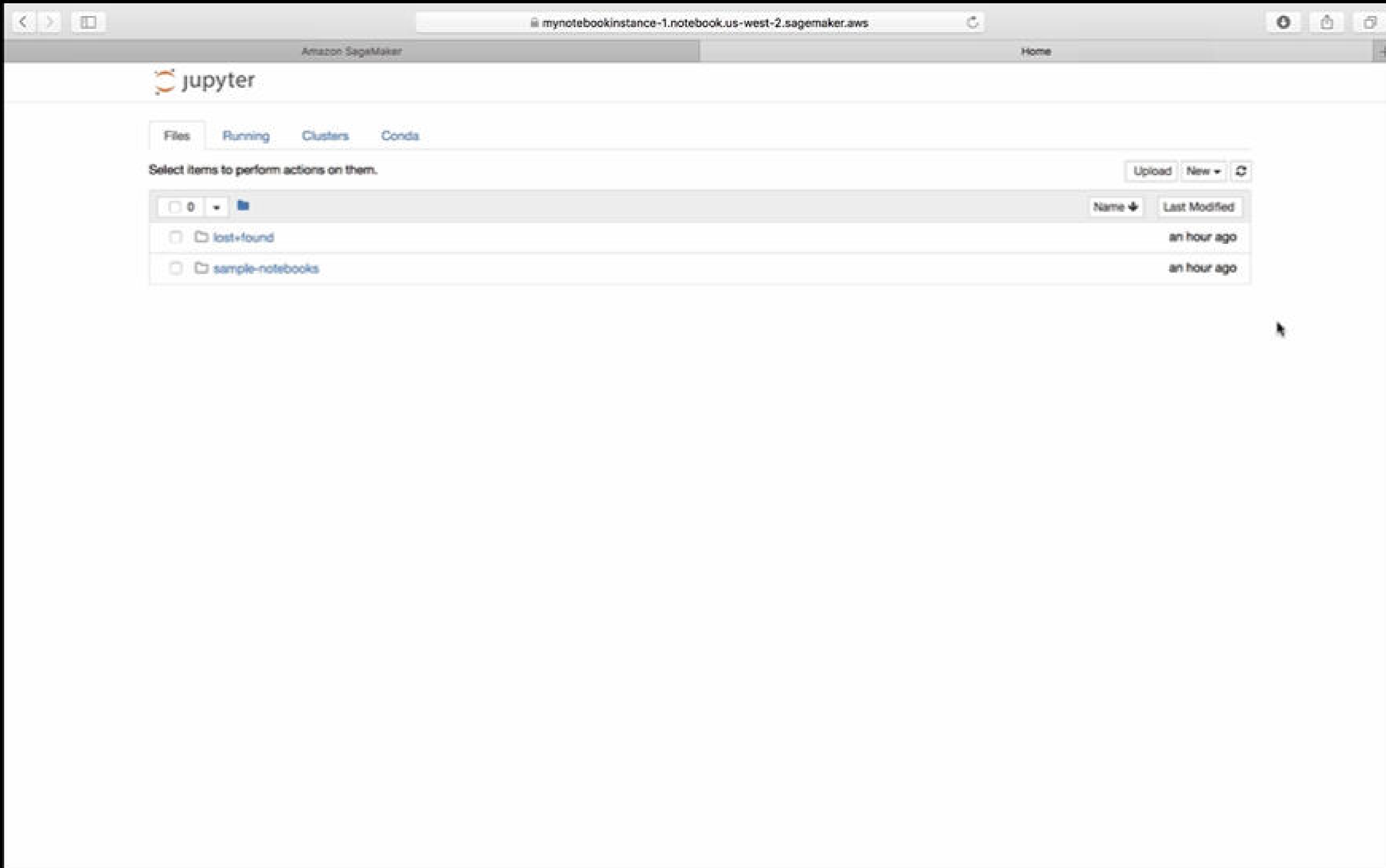


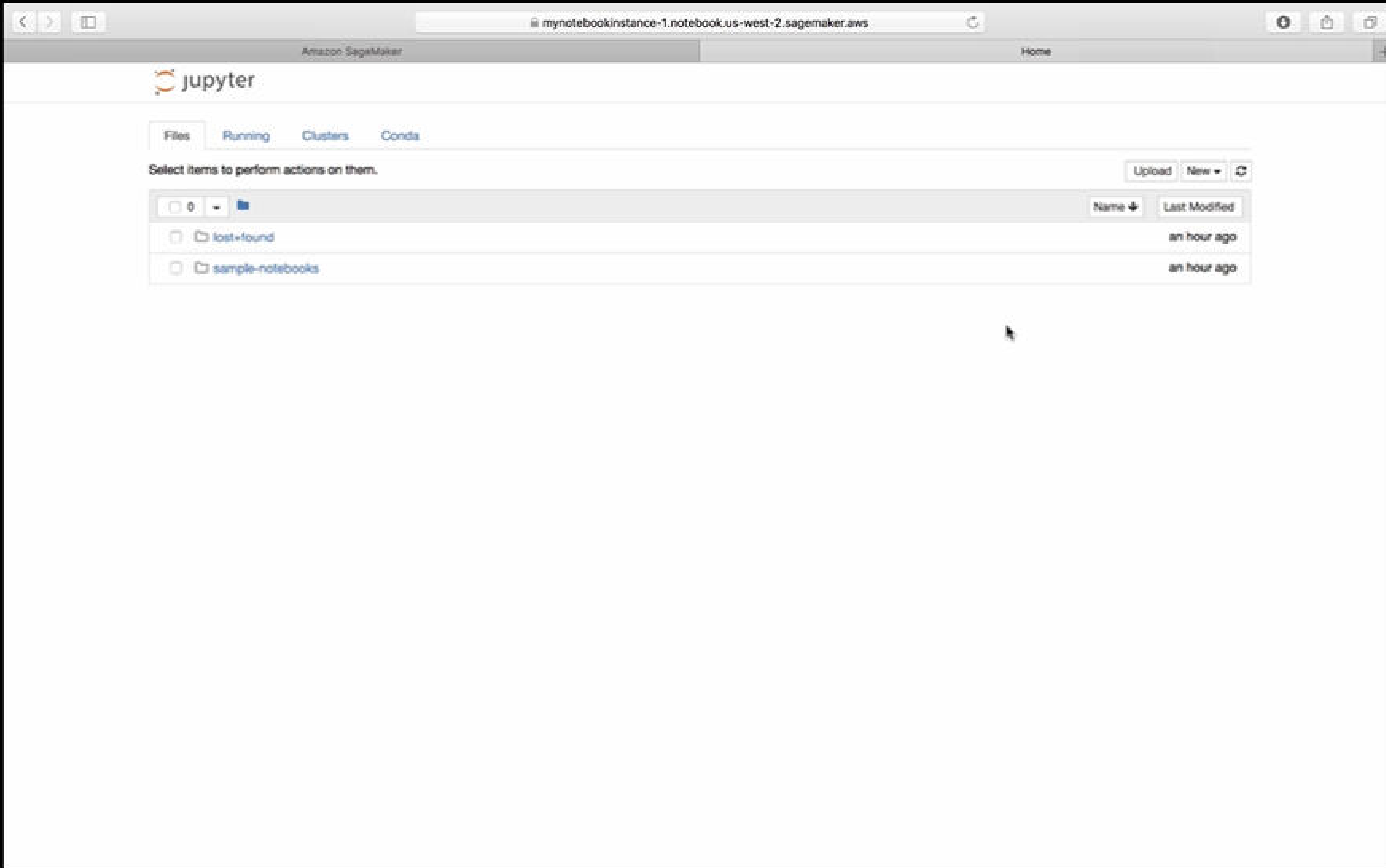
us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Success! You created your notebook instance.
Open the notebook instance when status is InService and open a template notebook to get started.
View details
Amazon SageMaker > Notebook instances
Notebook instances
Open
Start
Update settings
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
mynotebookinstance-1
ml.t2.medium
Nov 18, 2017 23:41 UTC
Pending
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

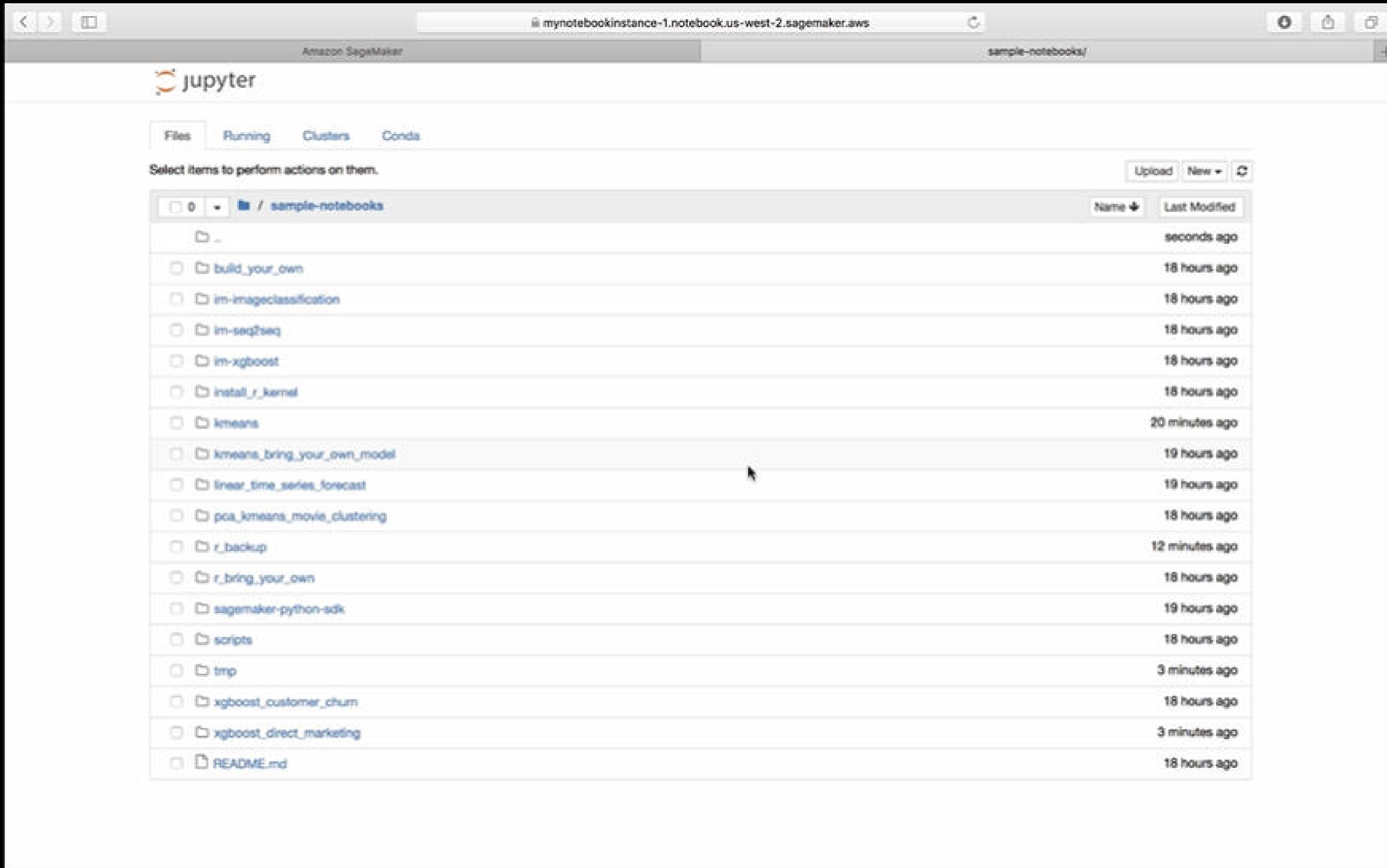

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances
Notebook instances
Open
Start
Update settings
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
mynotebookinstance-1
ml.t2.medium
Nov 18, 2017 23:41 UTC
InService
Stop | Open
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

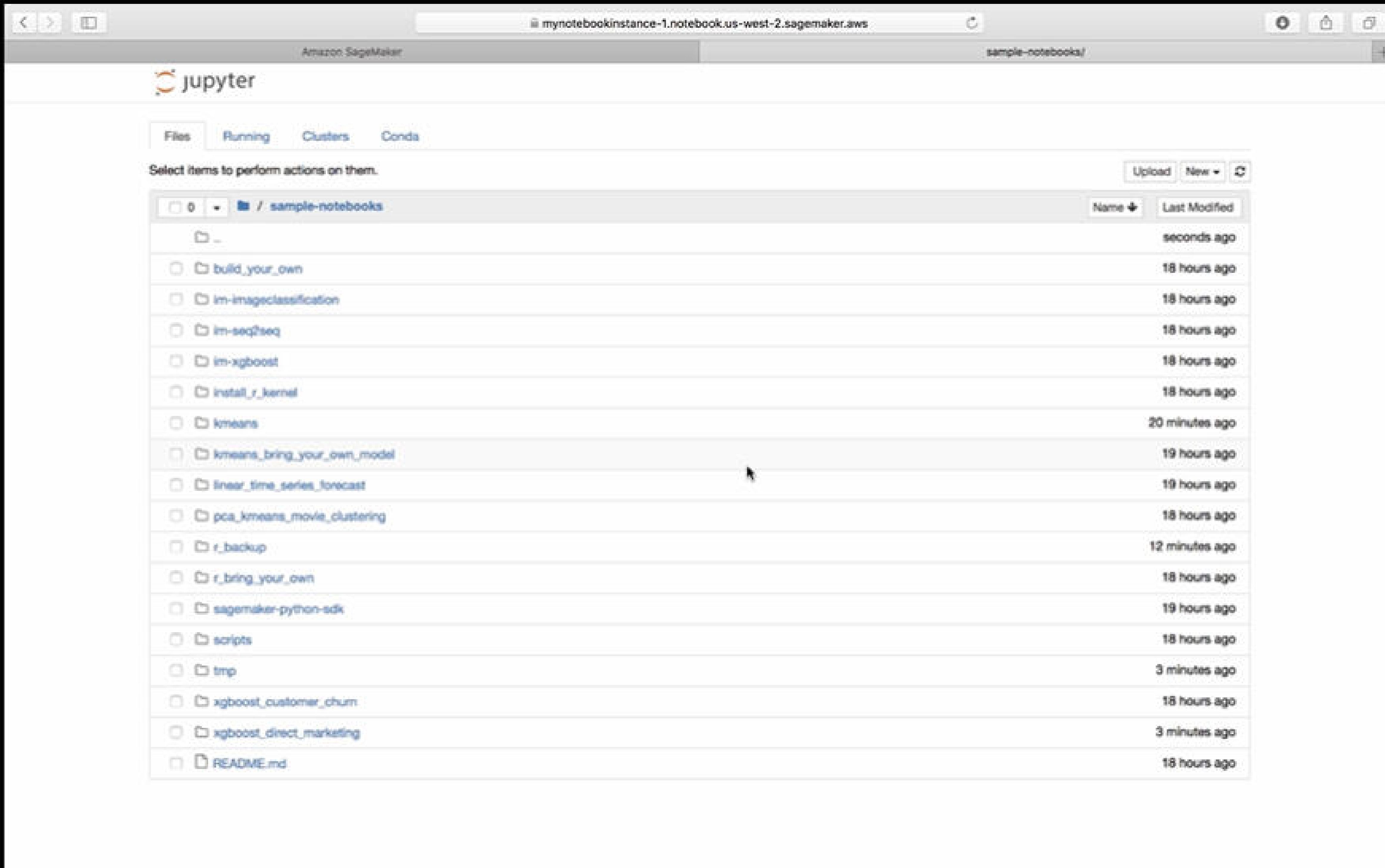

us-west-2.console.aws.amazon.com
aws
Services
Resource Groups
Support
Amazon SageMaker
Dashboard
Notebook instances
Jobs
Resources
Models
Endpoint configuration
Endpoints
Amazon SageMaker > Notebook instances
Notebook instances
Open
Start
Update settings
Actions
Create notebook instance
Search notebook instances
1
Name
Instance
Creation time
Status
Actions
mynotebookinstance-1
ml.t2.medium
Nov 18, 2017 23:41 UTC
InService
Stop | Open
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

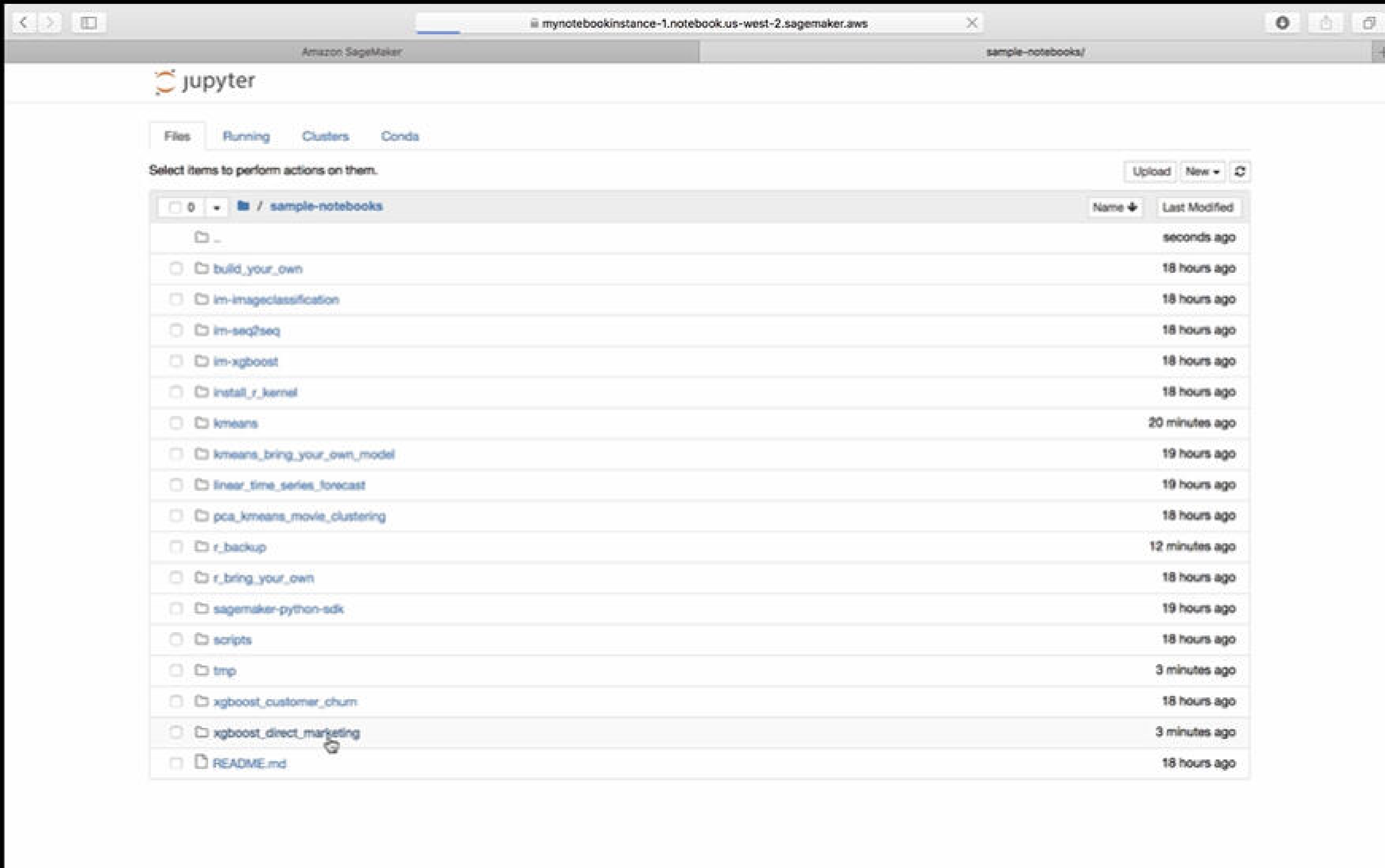

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
Home
jupyter
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
0
Name
Last Modified
lost+found
an hour ago
sample-notebooks
an hour ago

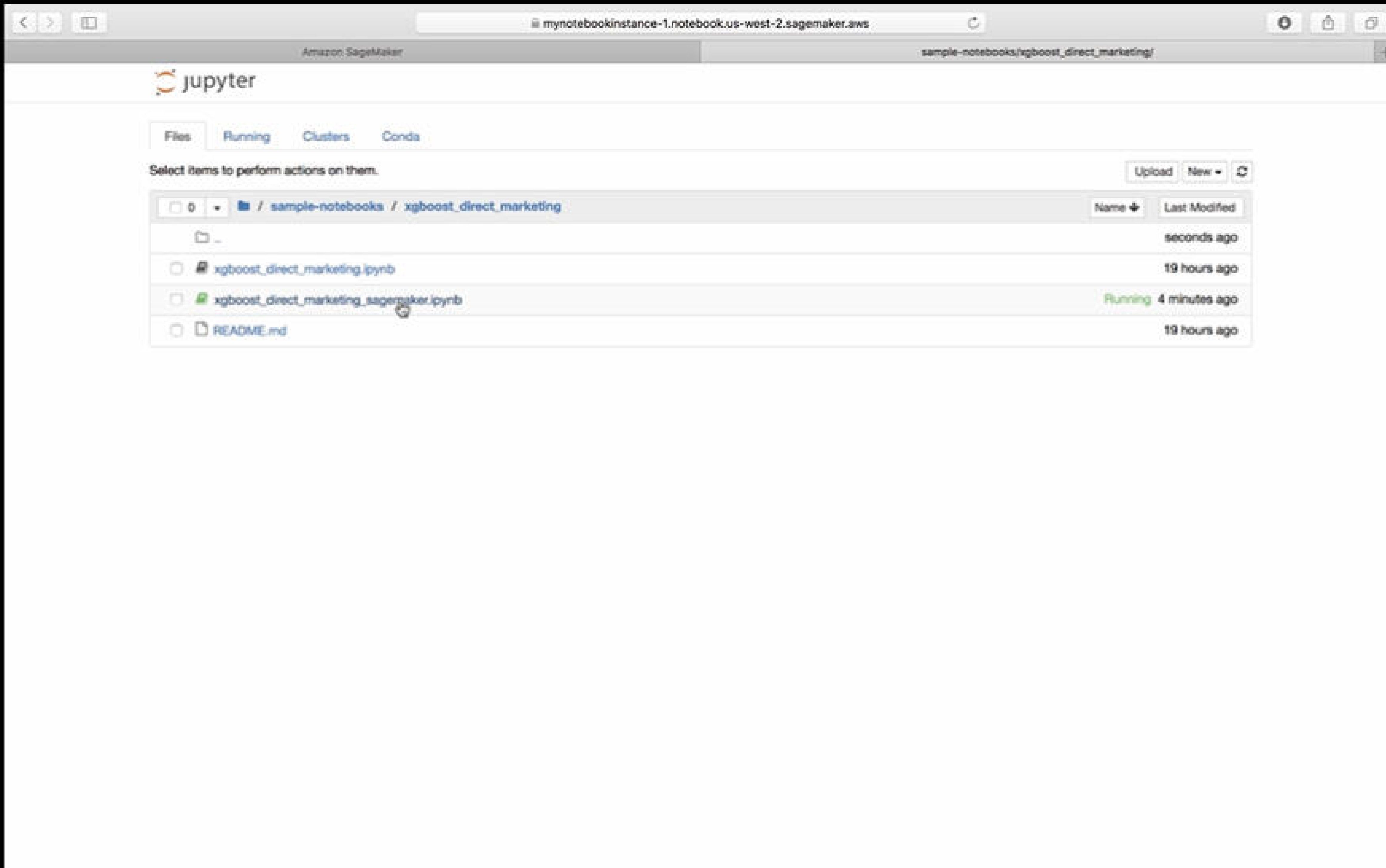

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
Home
jupyter
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
0
Name
Last Modified
lost+found
an hour ago
sample-notebooks
an hour ago

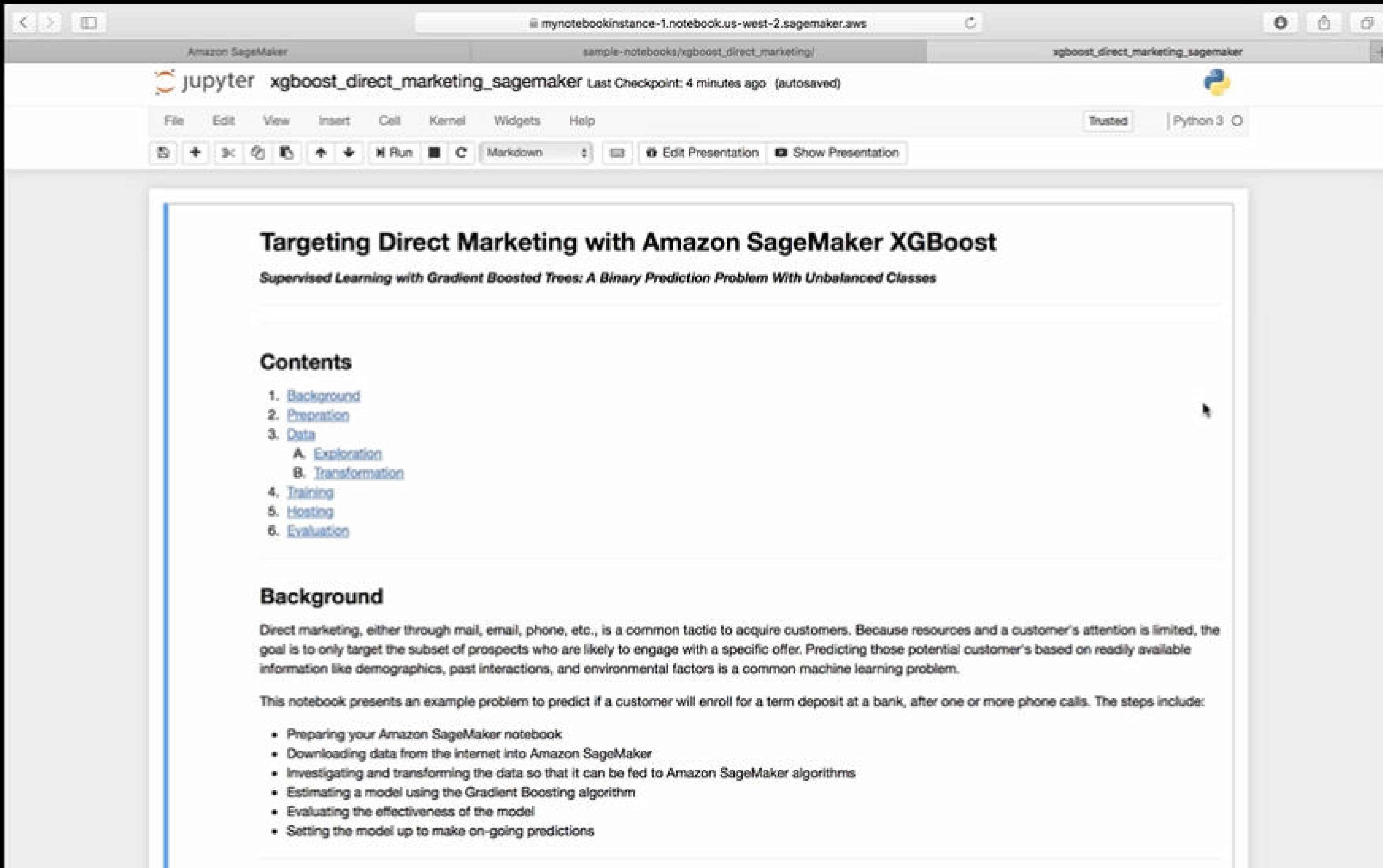

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
Home
jupyter
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
0
Name
Last Modified
lost+found
an hour ago
sample-notebooks
an hour ago

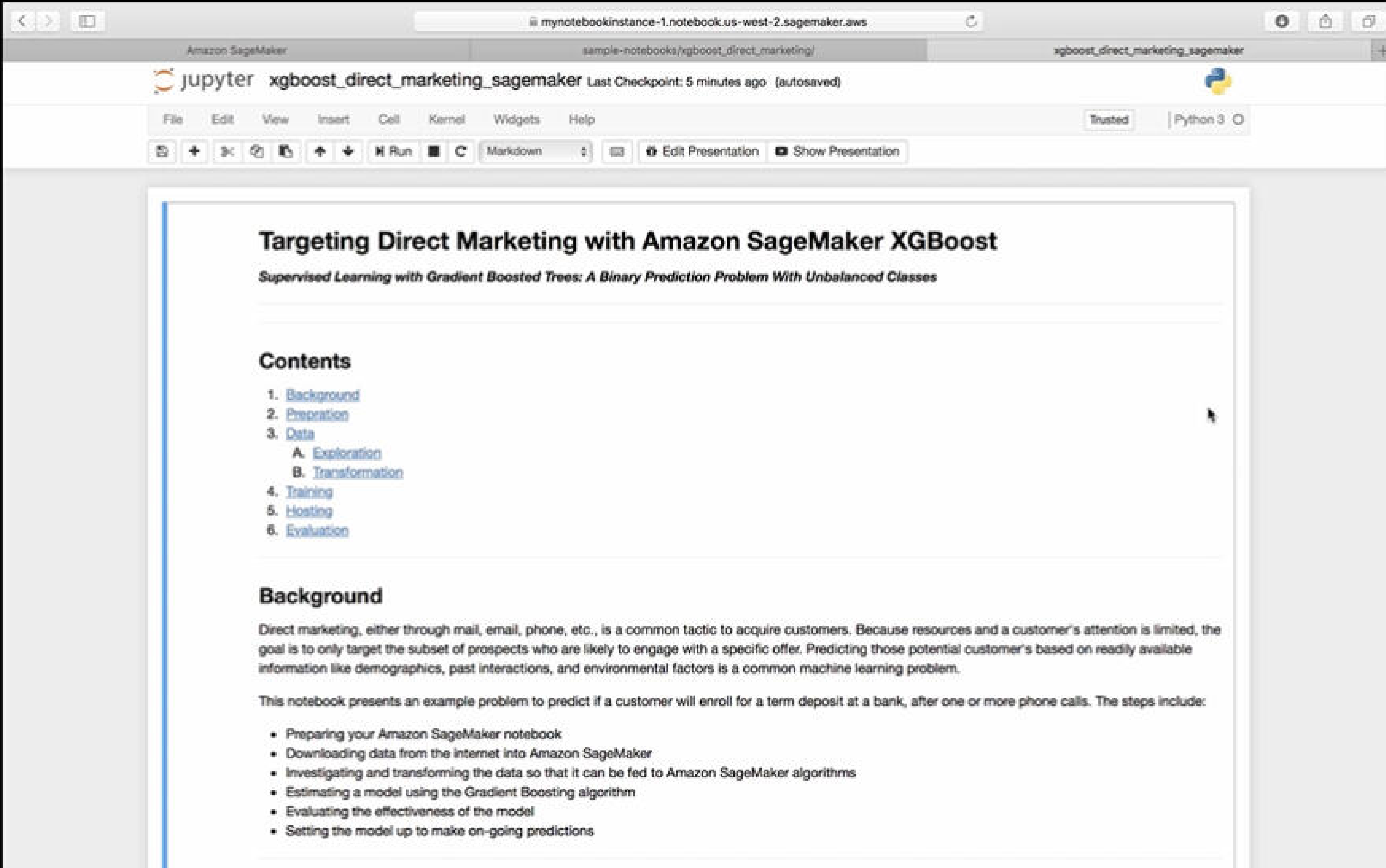

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/
jupyter
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
0
/ sample-notebooks
Name
Last Modified
..
seconds ago
build_your_own
18 hours ago
im-imageclassification
18 hours ago
im-seq2seq
18 hours ago
im-xgboost
18 hours ago
install_r_kernel
18 hours ago
kmeans
20 minutes ago
kmeans_bring_your_own_model
19 hours ago
linear_time_series_forecast
19 hours ago
pca_kmeans_movie_clustering
18 hours ago
r_backup
12 minutes ago
r_bring_your_own
18 hours ago
sagemaker-python-sdk
19 hours ago
scripts
18 hours ago
tmp
3 minutes ago
xgboost_customer_churn
18 hours ago
xgboost_direct_marketing
3 minutes ago
README.md
18 hours ago


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/
jupyter
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
0
/ sample-notebooks
Name
Last Modified
..
seconds ago
build_your_own
18 hours ago
im-imageclassification
18 hours ago
im-seq2seq
18 hours ago
im-xgboost
18 hours ago
install_r_kernel
18 hours ago
kmeans
20 minutes ago
kmeans_bring_your_own_model
19 hours ago
linear_time_series_forecast
19 hours ago
pca_kmeans_movie_clustering
18 hours ago
r_backup
12 minutes ago
r_bring_your_own
18 hours ago
sagemaker-python-sdk
19 hours ago
scripts
18 hours ago
tmp
3 minutes ago
xgboost_customer_churn
18 hours ago
xgboost_direct_marketing
3 minutes ago
README.md
18 hours ago


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/
jupyter
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
0
/ sample-notebooks
Name
Last Modified
..
seconds ago
build_your_own
18 hours ago
im-imageclassification
18 hours ago
im-seq2seq
18 hours ago
im-xgboost
18 hours ago
install_r_kernel
18 hours ago
kmeans
20 minutes ago
kmeans_bring_your_own_model
19 hours ago
linear_time_series_forecast
19 hours ago
pca_kmeans_movie_clustering
18 hours ago
r_backup
12 minutes ago
r_bring_your_own
18 hours ago
sagemaker-python-sdk
19 hours ago
scripts
18 hours ago
tmp
3 minutes ago
xgboost_customer_churn
18 hours ago
xgboost_direct_marketing
3 minutes ago
README.md
18 hours ago


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
jupyter
Files
Running
Clusters
Conda
Select items to perform actions on them.
Upload
New
0
/ sample-notebooks / xgboost_direct_marketing
Name
Last Modified
..
seconds ago
xgboost_direct_marketing.ipynb
19 hours ago
xgboost_direct_marketing_sagemaker.ipynb
Running
4 minutes ago
README.md
19 hours ago


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
jupyter xgboost_direct_marketing_sagemaker Last Checkpoint: 4 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Edit Presentation
Show Presentation
Targeting Direct Marketing with Amazon SageMaker XGBoost
Supervised Learning with Gradient Boosted Trees: A Binary Prediction Problem With Unbalanced Classes
Contents
1. Background
2. Prepration
3. Data
A. Exploration
B. Transformation
4. Training
5. Hosting
6. Evaluation
Background
Direct marketing, either through mail, email, phone, etc., is a common tactic to acquire customers. Because resources and a customer's attention is limited, the goal is to only target the subset of prospects who are likely to engage with a specific offer. Predicting those potential customer's based on readily available information like demographics, past interactions, and environmental factors is a common machine learning problem.
This notebook presents an example problem to predict if a customer will enroll for a term deposit at a bank, after one or more phone calls. The steps include:
• Preparing your Amazon SageMaker notebook
• Downloading data from the internet into Amazon SageMaker
• Investigating and transforming the data so that it can be fed to Amazon SageMaker algorithms
• Estimating a model using the Gradient Boosting algorithm
• Evaluating the effectiveness of the model
• Setting the model up to make on-going predictions



mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
jupyter xgboost_direct_marketing_sagemaker Last Checkpoint: 5 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Edit Presentation
Show Presentation
Targeting Direct Marketing with Amazon SageMaker XGBoost
Supervised Learning with Gradient Boosted Trees: A Binary Prediction Problem With Unbalanced Classes
Contents
1. Background
2. Prepration
3. Data
A. Exploration
B. Transformation
4. Training
5. Hosting
6. Evaluation
Background
Direct marketing, either through mail, email, phone, etc., is a common tactic to acquire customers. Because resources and a customer's attention is limited, the goal is to only target the subset of prospects who are likely to engage with a specific offer. Predicting those potential customer's based on readily available information like demographics, past interactions, and environmental factors is a common machine learning problem.
This notebook presents an example problem to predict if a customer will enroll for a term deposit at a bank, after one or more phone calls. The steps include:
Preparing your Amazon SageMaker notebook
Downloading data from the internet into Amazon SageMaker
Investigating and transforming the data so that it can be fed to Amazon SageMaker algorithms
Estimating a model using the Gradient Boosting algorithm
Evaluating the effectiveness of the model
Setting the model up to make on-going predictions

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

jupyter xgboost_direct_marketing_sagemaker Last Checkpoint: 5 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Code | Edit Presentation | Show Presentation

Preparation

To start, let's define:

- IAM role for training/hosting to access resources
- The name of the S3 bucket we want to use
- An S3 prefix

In [1]:
```
import os
import boto3

role = 'arn:aws:iam::[redacted]:role/sagemakerrole'
bucket = 'sagemaker-pdx1'
prefix = 'xgboost'
```

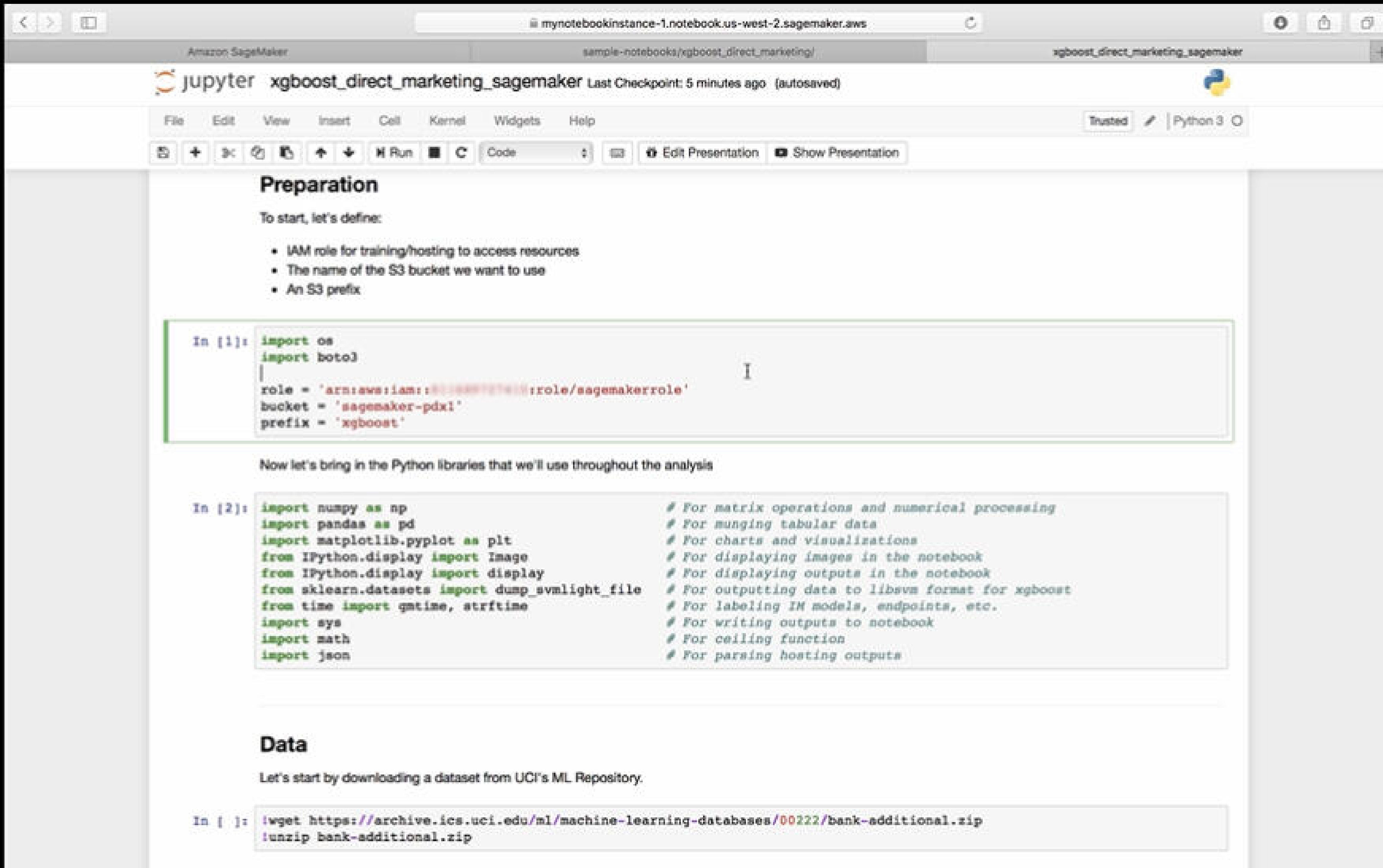
Now let's bring in the Python libraries that we'll use throughout the analysis

In [2]:
```
import numpy as np                                # For matrix operations and numerical processing
import pandas as pd                               # For munging tabular data
import matplotlib.pyplot as plt                   # For charts and visualizations
from IPython.display import Image                 # For displaying images in the notebook
from IPython.display import display               # For displaying outputs in the notebook
from sklearn.datasets import dump_svmlight_file   # For outputting data to libsvm format for xgboost
from time import gmtime, strftime                 # For labeling SM models, endpoints, etc.
import sys                                        # For writing outputs to notebook
import math                                       # For ceiling function
import json                                       # For parsing hosting outputs
```

Data

Let's start by downloading a dataset from UCI's ML Repository.

In [ ]:
```
!wget https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip
!unzip bank-additional.zip
```


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
jupyter xgboost_direct_marketing_sagemaker Last Checkpoint: 5 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Edit Presentation
Show Presentation
Preparation
To start, let's define:
• IAM role for training/hosting to access resources
• The name of the S3 bucket we want to use
• An S3 prefix
In [1]:
import os
import boto3
role = 'arn:aws:iam:: :role/sagemakerrole'
bucket = 'sagemaker-pdx1'
prefix = 'xgboost'
Now let's bring in the Python libraries that we'll use throughout the analysis
In [2]:
import numpy as np # For matrix operations and numerical processing
import pandas as pd # For munging tabular data
import matplotlib.pyplot as plt # For charts and visualizations
from IPython.display import Image # For displaying images in the notebook
from IPython.display import display # For displaying outputs in the notebook
from sklearn.datasets import dump_svmlight_file # For outputting data to libsvm format for xgboost
from time import gmtime, strftime # For labeling ML models, endpoints, etc.
import sys # For writing outputs to notebook
import math # For ceiling function
import json # For parsing hosting outputs
Data
Let's start by downloading a dataset from UCI's ML Repository.
In [ ]:
!wget https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip
!unzip bank-additional.zip

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

jupyter xgboost_direct_marketing_sagemaker Last Checkpoint: 5 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code Edit Presentation Show Presentation

Preparation

To start, let's define:

- IAM role for training/hosting to access resources
- The name of the S3 bucket we want to use
- An S3 prefix

In [12]:
```
import os
import boto3

role = 'arn:aws:iam::[illegible]:role/sagemakerrole'
bucket = 'sagemaker-pdx1'
prefix = 'xgboost'
```

Now let's bring in the Python libraries that we'll use throughout the analysis

In [2]:
```
import numpy as np                                  # For matrix operations and numerical processing
import pandas as pd                                 # For munging tabular data
import matplotlib.pyplot as plt                     # For charts and visualizations
from IPython.display import Image                   # For displaying images in the notebook
from IPython.display import display                 # For displaying outputs in the notebook
from sklearn.datasets import dump_svmlight_file     # For outputting data to libsvm format for xgboost
from time import gmtime, strftime                   # For labeling ML models, endpoints, etc.
import sys                                          # For writing outputs to notebook
import math                                         # For ceiling function
import json                                         # For parsing hosting outputs
```

Data

Let's start by downloading a dataset from UCI's ML Repository.

In [ ]:
```
!wget https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip
!unzip bank-additional.zip
```

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

jupyter xgboost_direct_marketing_sagemaker Last Checkpoint: 5 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code Edit Presentation Show Presentation

Preparation

To start, let's define:

- IAM role for training/hosting to access resources
- The name of the S3 bucket we want to use
- An S3 prefix

In [12]:

```
import os
import boto3

role = 'arn:aws:iam::[illegible]:role/sagemakerrole'
bucket = 'sagemaker-pdx1'
prefix = 'xgboost'
```

Now let's bring in the Python libraries that we'll use throughout the analysis

In [2]:

```
import numpy as np                                # For matrix operations and numerical processing
import pandas as pd                               # For munging tabular data
import matplotlib.pyplot as plt                   # For charts and visualizations
from IPython.display import Image                 # For displaying images in the notebook
from IPython.display import display               # For displaying outputs in the notebook
from sklearn.datasets import dump_svmlight_file   # For outputting data to libsvm format for xgboost
from time import gmtime, strftime                 # For labeling ML models, endpoints, etc.
import sys                                        # For writing outputs to notebook
import math                                       # For ceiling function
import json                                       # For parsing hosting outputs
```

Data

Let's start by downloading a dataset from UCI's ML Repository.

In [ ]:

```
!wget https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip
!unzip bank-additional.zip
```

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

jupyter xgboost_direct_marketing_sagemaker Last Checkpoint: 5 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Markdown | Edit Presentation | Show Presentation

```
import math                                        # For ceiling function
import json                                        # For parsing hosting outputs
```

Data

Let's start by downloading a dataset from UCI's ML Repository.

In [*]:

```
!wget https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip
!unzip bank-additional.zip
```

```
--2017-11-19 00:42:39--  https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank-additional.zip
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.249
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.249|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 444572 (434K) [application/zip]
Saving to: 'bank-additional.zip'

bank-additional.zip   0%[                    ]       0  --.-KB/s
```

Now lets read this into a Pandas data frame and take a look.

In [ ]:

```
data = pd.read_csv('./bank-additional/bank-additional-full.csv', sep=';')
pd.set_option('display.max_columns', 500)     # Make sure we can see all of the columns
pd.set_option('display.max_rows', 20)         # Keep the output on one page
data
```

Let's talk about the data. At a high level, we can see:

- We have a little over 40K customer records, and 20 features for each customer
- The features are mixed; some numeric, some categorical
- The data appears to be sorted, at least by `time` and `contact`, maybe more

Specifics on each of the features:

Demographics:

- `age`: Customer's age (numeric)

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Out[15]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	261	1	999	0	nonexistent
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	149	1	999	0	nonexistent
2	37	services	married	high.school	no	yes	no	telephone	may	mon	226	1	999	0	nonexistent
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	151	1	999	0	nonexistent
4	56	services	married	high.school	no	no	yes	telephone	may	mon	307	1	999	0	nonexistent
5	45	services	married	basic.9y	unknown	no	no	telephone	may	mon	198	1	999	0	nonexistent
6	59	admin.	married	professional.course	no	no	no	telephone	may	mon	139	1	999	0	nonexistent
7	41	blue-collar	married	unknown	unknown	no	no	telephone	may	mon	217	1	999	0	nonexistent
8	24	technician	single	professional.course	no	yes	no	telephone	may	mon	380	1	999	0	nonexistent
9	25	services	single	high.school	no	yes	no	telephone	may	mon	50	1	999	0	nonexistent
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
41178	62	retired	married	university.degree	no	no	no	cellular	nov	thu	483	2	6	3	success
41179	64	retired	divorced	professional.course	no	yes	no	cellular	nov	fri	151	3	999	0	nonexistent
41180	36	admin.	married	university.degree	no	no	no	cellular	nov	fri	254	2	999	0	nonexistent
41181	37	admin.	married	university.degree	no	yes	no	cellular	nov	fri	281	1	999	0	nonexistent
41182	29	unemployed	single	basic.4y	no	yes	no	cellular	nov	fri	112	1	9	1	success
41183	73	retired	married	professional.course	no	yes	no	cellular	nov	fri	334	1	999	0	nonexistent
41184	46	blue-collar	married	professional.course	no	no	no	cellular	nov	fri	383	1	999	0	nonexistent
41185	56	retired	married	university.degree	no	yes	no	cellular	nov	fri	189	2	999	0	nonexistent
41186	44	technician	married	professional.course	no	no	no	cellular	nov	fri	442	1	999	0	nonexistent
41187	74	retired	married	professional.course	no	yes	no	cellular	nov	fri	239	3	999	1	failure

41188 rows × 21 columns

Let's talk about the data. At a high level, we can see:

- We have a little over 40K customer records, and 20 features for each customer

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Out[15]:

loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
no	telephone	may	mon	261	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	149	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	226	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	151	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
yes	telephone	may	mon	307	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	198	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	139	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	217	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	380	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	50	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
no	cellular	nov	thu	483	2	6	3	success	-1.1	94.767	-50.8	1.031	4963.6	yes
no	cellular	nov	fri	151	3	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	254	2	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	281	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	112	1	9	1	success	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	334	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	383	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	189	2	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	442	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	239	3	999	1	failure	-1.1	94.767	-50.8	1.028	4963.6	no

Let's talk about the data. At a high level, we can see:

- We have a little over 40K customer records, and 20 features for each customer

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Out[15]:

loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
no	telephone	may	mon	261	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	149	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	226	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	151	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
yes	telephone	may	mon	307	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	198	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	139	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	217	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	380	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	50	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
no	cellular	nov	thu	483	2	6	3	success	-1.1	94.767	-50.8	1.031	4963.6	yes
no	cellular	nov	fri	151	3	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	254	2	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	281	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	112	1	9	1	success	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	334	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	383	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	189	2	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	442	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	239	3	999	1	failure	-1.1	94.767	-50.8	1.028	4963.6	no

Let's talk about the data. At a high level, we can see:

- We have a little over 40K customer records, and 20 features for each customer

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Out[15]:

loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
no	telephone	may	mon	261	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	149	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	226	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	151	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
yes	telephone	may	mon	307	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	198	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	139	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	217	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	380	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
no	telephone	may	mon	50	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
no	cellular	nov	thu	483	2	6	3	success	-1.1	94.767	-50.8	1.031	4963.6	yes
no	cellular	nov	fri	151	3	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	254	2	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	281	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	112	1	9	1	success	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	334	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	383	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	189	2	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	no
no	cellular	nov	fri	442	1	999	0	nonexistent	-1.1	94.767	-50.8	1.028	4963.6	yes
no	cellular	nov	fri	239	3	999	1	failure	-1.1	94.767	-50.8	1.028	4963.6	no

Let's talk about the data. At a high level, we can see:

- We have a little over 40K customer records, and 20 features for each customer

Present | Slides | Themes | Help

Let's talk about the data. At a high level, we can see:

- We have a little over 40K customer records, and 20 features for each customer
- The features are mixed; some numeric, some categorical
- The data appears to be sorted, at least by `time` and `contact`, maybe more

Specifics on each of the features:

Demographics:

- `age`: Customer's age (numeric)
- `job`: Type of job (categorical: 'admin.', 'services', ...)
- `marital`: Marital status (categorical: 'married', 'single', ...)
- `education`: Level of education (categorical: 'basic.4y', 'high.school', ...)

Past customer events:

- `default`: Has credit in default? (categorical: 'no', 'unknown', ...)
- `housing`: Has housing loan? (categorical: 'no', 'yes', ...)
- `loan`: Has personal loan? (categorical: 'no', 'yes', ...)

Past direct marketing contacts:

- `contact`: Contact communication type (categorical: 'cellular', 'telephone', ...)
- `month`: Last contact month of year (categorical: 'may', 'nov', ...)
- `day_of_week`: Last contact day of the week (categorical: 'mon', 'fri', ...)
- `duration`: Last contact duration, in seconds (numeric). Important note: If duration = 0 then `y` = 'no'.

Campaign information:

- `campaign`: Number of contacts performed during this campaign and for this client (numeric, includes last contact)
- `pdays`: Number of days that passed by after the client was last contacted from a previous campaign (numeric)
- `previous`: Number of contacts performed before this campaign and for this client (numeric)
- `poutcome`: Outcome of the previous marketing campaign (categorical: 'nonexistent','success', ...)

External environment factors:

Target variable:

- `y`: Has the client subscribed a term deposit? (binary: 'yes','no')

Exploration

Let's start exploring the data. First, let's understand how the features are distributed.

In [ ]:
```
# Frequency tables for each categorical feature
for column in data.select_dtypes(include=['object']).columns:
    display(pd.crosstab(index=data[column], columns='% observations', normalize='columns'))

# Histograms for each numeric features
display(data.describe())
%matplotlib inline
hist = data.hist(bins=30, sharey=True, figsize=(10, 10))
```

Notice that:

- Almost 90% of the values for our target variable `y` are "no", so most customers did not subscribe to a term deposit.
- Many of the predictive features take on values of "unknown". Some are more common than others. We should think carefully as to what causes a value of "unknown" (are these customers non-representative in some way?) and how we that should be handled.
  - Even if "unknown" is included as it's own distinct category, what does it mean given that, in reality, those observations likely fall within one of the other categories of that feature?
- Many of the predictive features have categories with very few observations in them. If we find a small category to be highly predictive of our target outcome, do we have enough evidence to make a generalization about that?
- Contact timing is particularly skewed. Almost a third in May and less than 1% in December. What does this mean for predicting our target variable next December?
- There are no missing values in our numeric features. Or missing values have already been imputed.
  - `pdays` takes a value near 1000 for almost all customers. Likely a placeholder value signifying no previous contact.
- Several numeric features have a very long tail. Do we need to handle these few observations with extremely large values differently?
- Several numeric features (particularly the macroeconomic ones) occur in distinct buckets. Should these be treated as categorical?

Next, let's look at how our features relate to the target that we are attempting to predict.

In [ ]:
```
display(data.corr())
```

admin.	0.253035
blue-collar	0.224677
entrepreneur	0.035350
housemaid	0.025736
management	0.070992
retired	0.041760
self-employed	0.034500
services	0.096363
student	0.021244
technician	0.163713

Notice that:

- Almost 90% of the values for our target variable `y` are "no", so most customers did not subscribe to a term deposit.
- Many of the predictive features take on values of "unknown". Some are more common than others. We should think carefully as to what causes a value of "unknown" (are these customers non-representative in some way?) and how we that should be handled.
  - Even if "unknown" is included as it's own distinct category, what does it mean given that, in reality, those observations likely fall within one of the other categories of that feature?
- Many of the predictive features have categories with very few observations in them. If we find a small category to be highly predictive of our target outcome, do we have enough evidence to make a generalization about that?
- Contact timing is particularly skewed. Almost a third in May and less than 1% in December. What does this mean for predicting our target variable next December?
- There are no missing values in our numeric features. Or missing values have already been imputed.
  - `pdays` takes a value near 1000 for almost all customers. Likely a placeholder value signifying no previous contact.
- Several numeric features have a very long tail. Do we need to handle these few observations with extremely large values differently?
- Several numeric features (particularly the macroeconomic ones) occur in distinct buckets. Should these be treated as categorical?

Next, let's look at how our features relate to the target that we are attempting to predict.

```
In [ ]: display(data.corr())
```

Notice that:

- Features vary widely in their relationship with one another. Some with highly negative correlation, others with highly positive correlation.

- Many of the predictive features take on values of "unknown". Some are more common than others. We should think carefully as to what causes a value of "unknown" (are these customers non-representative in some way?) and how we that should be handled.
  - Even if "unknown" is included as it's own distinct category, what does it mean given that, in reality, those observations likely fall within one of the other categories of that feature?
- Many of the predictive features have categories with very few observations in them. If we find a small category to be highly predictive of our target outcome, do we have enough evidence to make a generalization about that?
- Contact timing is particularly skewed. Almost a third in May and less than 1% in December. What does this mean for predicting our target variable next December?
- There are no missing values in our numeric features. Or missing values have already been imputed.
  - `pdays` takes a value near 1000 for almost all customers. Likely a placeholder value signifying no previous contact.
- Several numeric features have a very long tail. Do we need to handle these few observations with extremely large values differently?
- Several numeric features (particularly the macroeconomic ones) occur in distinct buckets. Should these be treated as categorical?

Next, let's look at how our features relate to the target that we are attempting to predict.

In [17]:
```
display(data.corr())
```

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
age	1.000000	-0.000866	0.004594	-0.034369	0.024365	-0.000371	0.000857	0.129372	0.010767	-0.017725
duration	-0.000866	1.000000	-0.071699	-0.047577	0.020640	-0.027968	0.005312	-0.008173	-0.032897	-0.044703
campaign	0.004594	-0.071699	1.000000	0.052584	-0.079141	0.150754	0.127836	-0.013733	0.135133	0.144095
pdays	-0.034369	-0.047577	0.052584	1.000000	-0.587514	0.271004	0.078889	-0.091342	0.296899	0.372605
previous	0.024365	0.020640	-0.079141	-0.587514	1.000000	-0.420489	-0.203130	-0.050936	-0.454494	-0.501333
emp.var.rate	-0.000371	-0.027968	0.150754	0.271004	-0.420489	1.000000	0.775334	0.196041	0.972245	0.906970
cons.price.idx	0.000857	0.005312	0.127836	0.078889	-0.203130	0.775334	1.000000	0.058986	0.688230	0.522034
cons.conf.idx	0.129372	-0.008173	-0.013733	-0.091342	-0.050936	0.196041	0.058986	1.000000	0.277686	0.100513
euribor3m	0.010767	-0.032897	0.135133	0.296899	-0.454494	0.972245	0.688230	0.277686	1.000000	0.945154
nr.employed	-0.017725	-0.044703	0.144095	0.372605	-0.501333	0.906970	0.522034	0.100513	0.945154	1.000000

Notice that:

- Features vary widely in their relationship with one another. Some with highly negative correlation, others with highly positive correlation.

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

- Handling missing values: Some machine learning algorithms are capable of handling missing values, but most would rather not. Options include:
  - Removing observations with missing values: This works well if only a very small fraction of observations have incomplete information.
  - Remove features with missing values: This works well if there are a small number of features which have a large number of missing values.
  - Imputing missing values: Entire books have been written on this topic, but common choices are replacing the missing value with the mode or mean of that column's non-missing values.
- Converting categorical to numeric: The most common method is one hot encoding, which for each feature maps every distinct value of that column to its own feature which takes a value of 1 when the categorical feature is equal to that value, and 0 otherwise.
- Oddly distributed data: Although for non-linear models like Gradient Boosted Trees, this has very limited implications, parametric models like regression can produce wildly inaccurate estimates when fed highly skewed data. In some cases, simply taking the natural log of the features is sufficient to produce more normally distributed data. In others, bucketing values into discrete ranges is helpful. These buckets can then be treated as categorical variables and included in the model when one hot encoded.
- Handling more complicated data types: Mainpulating images, text, or data at varying grains is left for other notebook templates.

Luckily, some of these aspects have already been handled for us, and the algorithm we are showcasing tends to do well at handling sparse or oddly distributed data. Therefore, let's keep pre-processing simple.

In [10]:
```
data['no_previous_contact'] = np.where(data['pdays'] == 999, 1, 0)                                  # Indicator variable
data['not_working'] = np.where(np.in1d(data['job'], ['student', 'retired', 'unemployed']), 1, 0)    # Indicator for indi
model_data = pd.get_dummies(data)                                                                   # Convert categorica
```

Another question to ask yourself before building a model is whether certain features will add value in your final use case. For example, if your goal is to deliver the best prediction, then will you have access to that data at the moment of prediction? Knowing it's raining is highly predictive for umbrella sales, but forecasting weather far enough out to plan inventory on umbrellas is probably just as difficult as forecasting umbrella sales without knowledge of the weather. So, including this in your model may give you a false sense of precision.

Following this logic, let's remove the economic features and `duration` from our data as they would need to be forecasted with high precision to use as inputs in future predictions.

Even if we were to use values of the economic indicators from the previous quarter, this value is likely not as relevant for prospects contacted early in the next quarter as those contacted later on.

In [8]:
```
model_data = model_data.drop(['duration', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed'
```

When building a model whose primary goal is to predict a target value on new data, it is important to understand overfitting. Supervised learning models are designed to minimize error between their predictions of the target value and actuals, in the data they are given. This last part is key, as frequently in their quest for greater accuracy, machine learning models bias themselves toward picking up on minor idiosyncrasies within the data they are shown. These idiosyncrasies then don't repeat themselves in subsequent data, meaning those predictions can actually be made less accurate, at the expense of more accurate predictions

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

Following this logic, let's remove the economic features and `duration` from our data as they would need to be forecasted with high precision to use as inputs in future predictions.

Even if we were to use values of the economic indicators from the previous quarter, this value is likely not as relevant for prospects contacted early in the next quarter as those contacted later on.

In [19]:
```
model_data = model_data.drop(['duration', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed'
```

When building a model whose primary goal is to predict a target value on new data, it is important to understand overfitting. Supervised learning models are designed to minimize error between their predictions of the target value and actuals, in the data they are given. This last part is key, as frequently in their quest for greater accuracy, machine learning models bias themselves toward picking up on minor idiosyncrasies within the data they are shown. These idiosyncrasies then don't repeat themselves in subsequent data, meaning those predictions can actually be made less accurate, at the expense of more accurate predictions in the training phase.

The most common way of preventing this is to build models with the concept that a model shouldn't only be judged on its fit to the data it was trained on, but also on "new" data. There are several different ways of operationalizing this, holdout validation, cross-validation, leave-one-out validation, etc. For our purposes, we'll simply randomly split the data into 3 uneven groups. The model will be trained on 70% of data, it will then be evaluated on 20% of data to give us an estimate of the accuracy we hope to have on "new" data, and 10% will be held back as a final testing dataset which will be used later on.

In [9]:
```
train_data, validation_data, test_data = np.split(model_data.sample(frac=1, random_state=1729), [int(0.7 * len(model_da
```

Amazon SageMaker's XGBoost container expects data in the libSVM data format. This expects features and the target variable to be provided as separate arguments. Let's split these apart. Notice that although repetitive it's easiest to do this after the train|validation|test split rather than before. This avoids any misalignment issues due to random reordering.

In [10]:
```
dump_svmlight_file(X=train_data.drop(['y_no', 'y_yes'], axis=1), y=train_data['y_yes'], f='train.libsvm')
dump_svmlight_file(X=validation_data.drop(['y_no', 'y_yes'], axis=1), y=validation_data['y_yes'], f='validation.libsvm'
dump_svmlight_file(X=test_data.drop(['y_no', 'y_yes'], axis=1), y=test_data['y_yes'], f='test.libsvm')
```

Now we'll copy the file to S3 for Amazon SageMaker's managed training to pickup.

In [ ]:
```
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/train.libsvm')).upload_file('train.lib
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/validation.libsvm')).upload_file('vali
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/test.libsvm')).upload_file('test.libsv
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Following this logic, let's remove the economic features and `duration` from our data as they would need to be forecasted with high precision to use as inputs in future predictions.

Even if we were to use values of the economic indicators from the previous quarter, this value is likely not as relevant for prospects contacted early in the next quarter as those contacted later on.

In [19]:
```
model_data = model_data.drop(['duration', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed'
```

When building a model whose primary goal is to predict a target value on new data, it is important to understand overfitting. Supervised learning models are designed to minimize error between their predictions of the target value and actuals, in the data they are given. This last part is key, as frequently in their quest for greater accuracy, machine learning models bias themselves toward picking up on minor idiosyncrasies within the data they are shown. These idiosyncrasies then don't repeat themselves in subsequent data, meaning those predictions can actually be made less accurate, at the expense of more accurate predictions in the training phase.

The most common way of preventing this is to build models with the concept that a model shouldn't only be judged on its fit to the data it was trained on, but also on "new" data. There are several different ways of operationalizing this, holdout validation, cross-validation, leave-one-out validation, etc. For our purposes, we'll simply randomly split the data into 3 uneven groups. The model will be trained on 70% of data, it will then be evaluated on 20% of data to give us an estimate of the accuracy we hope to have on "new" data, and 10% will be held back as a final testing dataset which will be used later on.

In [20]:
```
train_data, validation_data, test_data = np.split(model_data.sample(frac=1, random_state=1729), [int(0.7 * len(model_da
```

Amazon SageMaker's XGBoost container expects data in the libSVM data format. This expects features and the target variable to be provided as separate arguments. Let's split these apart. Notice that although repetitive it's easiest to do this after the train|validation|test split rather than before. This avoids any misalignment issues due to random reordering.

In [10]:
```
dump_svmlight_file(X=train_data.drop(['y_no', 'y_yes'], axis=1), y=train_data['y_yes'], f='train.libsvm')
dump_svmlight_file(X=validation_data.drop(['y_no', 'y_yes'], axis=1), y=validation_data['y_yes'], f='validation.libsvm'
dump_svmlight_file(X=test_data.drop(['y_no', 'y_yes'], axis=1), y=test_data['y_yes'], f='test.libsvm')
```

Now we'll copy the file to S3 for Amazon SageMaker's managed training to pickup.

In [ ]:
```
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/train.libsvm')).upload_file('train.lib
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/validation.libsvm')).upload_file('vali
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/test.libsvm')).upload_file('test.libsv
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

Following this logic, let's remove the economic features and `duration` from our data as they would need to be forecasted with high precision to use as inputs in future predictions.

Even if we were to use values of the economic indicators from the previous quarter, this value is likely not as relevant for prospects contacted early in the next quarter as those contacted later on.

In [19]:
```
model_data = model_data.drop(['duration', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed'
```

When building a model whose primary goal is to predict a target value on new data, it is important to understand overfitting. Supervised learning models are designed to minimize error between their predictions of the target value and actuals, in the data they are given. This last part is key, as frequently in their quest for greater accuracy, machine learning models bias themselves toward picking up on minor idiosyncrasies within the data they are shown. These idiosyncrasies then don't repeat themselves in subsequent data, meaning those predictions can actually be made less accurate, at the expense of more accurate predictions in the training phase.

The most common way of preventing this is to build models with the concept that a model shouldn't only be judged on its fit to the data it was trained on, but also on "new" data. There are several different ways of operationalizing this, holdout validation, cross-validation, leave-one-out validation, etc. For our purposes, we'll simply randomly split the data into 3 uneven groups. The model will be trained on 70% of data, it will then be evaluated on 20% of data to give us an estimate of the accuracy we hope to have on "new" data, and 10% will be held back as a final testing dataset which will be used later on.

In [20]:
```
train_data, validation_data, test_data = np.split(model_data.sample(frac=1, random_state=1729), [int(0.7 * len(model_da
```

Amazon SageMaker's XGBoost container expects data in the libSVM data format. This expects features and the target variable to be provided as separate arguments. Let's split these apart. Notice that although repetitive it's easiest to do this after the train|validation|test split rather than before. This avoids any misalignment issues due to random reordering.

In [10]:
```
dump_svmlight_file(X=train_data.drop(['y_no', 'y_yes'], axis=1), y=train_data['y_yes'], f='train.libsvm')
dump_svmlight_file(X=validation_data.drop(['y_no', 'y_yes'], axis=1), y=validation_data['y_yes'], f='validation.libsvm'
dump_svmlight_file(X=test_data.drop(['y_no', 'y_yes'], axis=1), y=test_data['y_yes'], f='test.libsvm')
```

Now we'll copy the file to S3 for Amazon SageMaker's managed training to pickup.

In [ ]:
```
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/train.libsvm')).upload_file('train.lib
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/validation.libsvm')).upload_file('vali
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/test.libsvm')).upload_file('test.libsv
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Following this logic, let's remove the economic features and `duration` from our data as they would need to be forecasted with high precision to use as inputs in future predictions.

Even if we were to use values of the economic indicators from the previous quarter, this value is likely not as relevant for prospects contacted early in the next quarter as those contacted later on.

In [19]:
```
model_data = model_data.drop(['duration', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed'
```

When building a model whose primary goal is to predict a target value on new data, it is important to understand overfitting. Supervised learning models are designed to minimize error between their predictions of the target value and actuals, in the data they are given. This last part is key, as frequently in their quest for greater accuracy, machine learning models bias themselves toward picking up on minor idiosyncrasies within the data they are shown. These idiosyncrasies then don't repeat themselves in subsequent data, meaning those predictions can actually be made less accurate, at the expense of more accurate predictions in the training phase.

The most common way of preventing this is to build models with the concept that a model shouldn't only be judged on its fit to the data it was trained on, but also on "new" data. There are several different ways of operationalizing this, holdout validation, cross-validation, leave-one-out validation, etc. For our purposes, we'll simply randomly split the data into 3 uneven groups. The model will be trained on 70% of data, it will then be evaluated on 20% of data to give us an estimate of the accuracy we hope to have on "new" data, and 10% will be held back as a final testing dataset which will be used later on.

In [20]:
```
train_data, validation_data, test_data = np.split(model_data.sample(frac=1, random_state=1729), [int(0.7 * len(model_da
```

Amazon SageMaker's XGBoost container expects data in the libSVM data format. This expects features and the target variable to be provided as separate arguments. Let's split these apart. Notice that although repetitive it's easiest to do this after the train|validation|test split rather than before. This avoids any misalignment issues due to random reordering.

In [*]:
```
dump_svmlight_file(X=train_data.drop(['y_no', 'y_yes'], axis=1), y=train_data['y_yes'], f='train.libsvm')
dump_svmlight_file(X=validation_data.drop(['y_no', 'y_yes'], axis=1), y=validation_data['y_yes'], f='validation.libsvm'
dump_svmlight_file(X=test_data.drop(['y_no', 'y_yes'], axis=1), y=test_data['y_yes'], f='test.libsvm')
```

Now we'll copy the file to S3 for Amazon SageMaker's managed training to pickup.

In [ ]:
```
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/train.libsvm')).upload_file('train.lib
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/validation.libsvm')).upload_file('vali
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/test.libsvm')).upload_file('test.libsv
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

The most common way of preventing this is to build models with the concept that a model shouldn't only be judged on its fit to the data it was trained on, but also on "new" data. There are several different ways of operationalizing this, holdout validation, cross-validation, leave-one-out validation, etc. For our purposes, we'll simply randomly split the data into 3 uneven groups. The model will be trained on 70% of data, it will then be evaluated on 20% of data to give us an estimate of the accuracy we hope to have on "new" data, and 10% will be held back as a final testing dataset which will be used later on.

In [20]:
```
train_data, validation_data, test_data = np.split(model_data.sample(frac=1, random_state=1729), [int(0.7 * len(model_da
```

Amazon SageMaker's XGBoost container expects data in the libSVM data format. This expects features and the target variable to be provided as separate arguments. Let's split these apart. Notice that although repetitive it's easiest to do this after the train|validation|test split rather than before. This avoids any misalignment issues due to random reordering.

In [21]:
```
dump_svmlight_file(X=train_data.drop(['y_no', 'y_yes'], axis=1), y=train_data['y_yes'], f='train.libsvm')
dump_svmlight_file(X=validation_data.drop(['y_no', 'y_yes'], axis=1), y=validation_data['y_yes'], f='validation.libsvm'
dump_svmlight_file(X=test_data.drop(['y_no', 'y_yes'], axis=1), y=test_data['y_yes'], f='test.libsvm')
```

Now we'll copy the file to S3 for Amazon SageMaker's managed training to pickup.

In [ ]:
```
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/train.libsvm')).upload_file('train.lib
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/validation.libsvm')).upload_file('vali
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/test.libsvm')).upload_file('test.libsv
```

Training

Now we know most of our features have skewed distributions, some are highly correlated with one another, and some appear to have non-linear relationships with our target variable. Also, for targeting future prospects, good predictive accuracy is preferred to being able to explain why that prospect was targeted. Taken together, these aspects make gradient boosted trees a good candidate algorithm.

There are several intricacies to understanding the algorithm, but at a high level, gradient boosted trees works by combining predictions from many simple models, each of which tries to address the weaknesses of the previous models. By doing this the collection of simple models can actually outperform large, complex models. Other Amazon SageMaker notebooks elaborate on gradient boosting trees further and how they differ from similar algorithms.

`xgboost` is an extremely popular, open-source package for gradient boosted trees. It is computationally powerful, fully featured, and has been successfully used in many machine learning competitions. Let's start with a simple `xgboost` model, trained using Amazon SageMaker's managed, distributed training framework.

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

In [21]:
```
dump_svmlight_file(X=train_data.drop(['y_no', 'y_yes'], axis=1), y=train_data['y_yes'], f='train.libsvm')
dump_svmlight_file(X=validation_data.drop(['y_no', 'y_yes'], axis=1), y=validation_data['y_yes'], f='validation.libsvm'
dump_svmlight_file(X=test_data.drop(['y_no', 'y_yes'], axis=1), y=test_data['y_yes'], f='test.libsvm')
```

Now we'll copy the file to S3 for Amazon SageMaker's managed training to pickup.

In [22]:
```
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/train.libsvm')).upload_file('train.lib
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/validation.libsvm')).upload_file('vali
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train/test.libsvm')).upload_file('test.libsv
```

Training

Now we know most of our features have skewed distributions, some are highly correlated with one another, and some appear to have non-linear relationships with our target variable. Also, for targeting future prospects, good predictive accuracy is preferred to being able to explain why that prospect was targeted. Taken together, these aspects make gradient boosted trees a good candidate algorithm.

There are several intricacies to understanding the algorithm, but at a high level, gradient boosted trees works by combining predictions from many simple models, each of which tries to address the weaknesses of the previous models. By doing this the collection of simple models can actually outperform large, complex models. Other Amazon SageMaker notebooks elaborate on gradient boosting trees further and how they differ from similar algorithms.

`xgboost` is an extremely popular, open-source package for gradient boosted trees. It is computationally powerful, fully featured, and has been successfully used in many machine learning competitions. Let's start with a simple `xgboost` model, trained using Amazon SageMaker's managed, distributed training framework.

First we'll need to specify training parameters. This includes:

1. The role to use
2. Our training job name
3. The `xgboost` algorithm container
4. Training instance type and count
5. S3 location for training data
6. S3 location for output data
7. Algorithm hyperparameters
8. Stopping conditions

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Training

Now we know most of our features have skewed distributions, some are highly correlated with one another, and some appear to have non-linear relationships with our target variable. Also, for targeting future prospects, good predictive accuracy is preferred to being able to explain why that prospect was targeted. Taken together, these aspects make gradient boosted trees a good candidate algorithm.

There are several intricacies to understanding the algorithm, but at a high level, gradient boosted trees works by combining predictions from many simple models, each of which tries to address the weaknesses of the previous models. By doing this the collection of simple models can actually outperform large, complex models. Other Amazon SageMaker notebooks elaborate on gradient boosting trees further and how they differ from similar algorithms.

`xgboost` is an extremely popular, open-source package for gradient boosted trees. It is computationally powerful, fully featured, and has been successfully used in many machine learning competitions. Let's start with a simple `xgboost` model, trained using Amazon SageMaker's managed, distributed training framework.

First we'll need to specify training parameters. This includes:

1. The role to use
2. Our training job name
3. The `xgboost` algorithm container
4. Training instance type and count
5. S3 location for training data
6. S3 location for output data
7. Algorithm hyperparameters
8. Stopping conditions

In [ ]:

```
job_name = 'xgboost-dm-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print("Training job", job_name)

create_training_params = \
{
    "RoleArn": role,
    "TrainingJobName": job_name,
    "AlgorithmSpecification": {
        "TrainingImage": "032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest",
        "TrainingInputMode": "File"
    },
```

Present | Slides | Themes | Help

Training

Now we know most of our features have skewed distributions, some are highly correlated with one another, and some appear to have non-linear relationships with our target variable. Also, for targeting future prospects, good predictive accuracy is preferred to being able to explain why that prospect was targeted. Taken together, these aspects make gradient boosted trees a good candidate algorithm.

There are several intricacies to understanding the algorithm, but at a high level, gradient boosted trees works by combining predictions from many simple models, each of which tries to address the weaknesses of the previous models. By doing this the collection of simple models can actually outperform large, complex models. Other Amazon SageMaker notebooks elaborate on gradient boosting trees further and how they differ from similar algorithms.

`xgboost` is an extremely popular, open-source package for gradient boosted trees. It is computationally powerful, fully featured, and has been successfully used in many machine learning competitions. Let's start with a simple `xgboost` model, trained using Amazon SageMaker's managed, distributed training framework.

First we'll need to specify training parameters. This includes:

1. The role to use
2. Our training job name
3. The `xgboost` algorithm container
4. Training instance type and count
5. S3 location for training data
6. S3 location for output data
7. Algorithm hyperparameters
8. Stopping conditions

In [ ]:

```
job_name = 'xgboost-dm-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print("Training job", job_name)

create_training_params = \
{
    "RoleArn": role,
    "TrainingJobName": job_name,
    "AlgorithmSpecification": {
        "TrainingImage": "032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest",
        "TrainingInputMode": "File"
    },
```

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

3. The `xgboost` algorithm container
4. Training instance type and count
5. S3 location for training data
6. S3 location for output data
7. Algorithm hyperparameters
8. Stopping conditions

In [ ]:

```
job_name = 'xgboost-dm-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print("Training job", job_name)

create_training_params = \
{
    "RoleArn": role,
    "TrainingJobName": job_name,
    "AlgorithmSpecification": {
        "TrainingImage": "032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest",
        "TrainingInputMode": "File"
    },
    "ResourceConfig": {
        "InstanceCount": 2,
        "InstanceType": "ml.c4.2xlarge",
        "VolumeSizeInGB": 50
    },
    "InputDataConfig": [
        {
            "ChannelName": "train",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": "s3://{}/{}/train".format(bucket, prefix),
                    "S3DataDistributionType": "FullyReplicated"
                }
            },
            "ContentType": "libsvm",
            "CompressionType": "None"
        }
    ],
    "OutputDataConfig": {
        "S3OutputPath": "s3://{}/{}/output".format(bucket, prefix)
    },
    "HyperParameters": {
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

In [ ]:

```
job_name = 'xgboost-dm-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print("Training job", job_name)

create_training_params = \
{
    "RoleArn": role,
    "TrainingJobName": job_name,
    "AlgorithmSpecification": {
        "TrainingImage": "032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest",
        "TrainingInputMode": "File"
    },
    "ResourceConfig": {
        "InstanceCount": 2,
        "InstanceType": "ml.c4.2xlarge",
        "VolumeSizeInGB": 50
    },
    "InputDataConfig": [
        {
            "ChannelName": "train",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": "s3://{}/{}/train".format(bucket, prefix),
                    "S3DataDistributionType": "FullyReplicated"
                }
            },
            "ContentType": "libsvm",
            "CompressionType": "None"
        }
    ],
    "OutputDataConfig": {
        "S3OutputPath": "s3://{}/{}/output".format(bucket, prefix)
    },
    "HyperParameters": {
        "max_depth":"5",
        "eta":"0.2",
        "gamma":"4",
        "min_child_weight":"6",
        "subsample":"0.8",
        "silent":"0",
        "objective":"binary:logistic",
        "num_class":"1",
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

In [ ]:
```
job_name = 'xgboost-dm-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print("Training job", job_name)

create_training_params = \
{
    "RoleArn": role,
    "TrainingJobName": job_name,
    "AlgorithmSpecification": {
        "TrainingImage": "032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest",
        "TrainingInputMode": "File"
    },
    "ResourceConfig": {
        "InstanceCount": 2,
        "InstanceType": "ml.c4.2xlarge",
        "VolumeSizeInGB": 50
    },
    "InputDataConfig": [
        {
            "ChannelName": "train",
            "DataSource": {
                "S3DataSource": {
                    "S3DataType": "S3Prefix",
                    "S3Uri": "s3://{}/{}/train".format(bucket, prefix),
                    "S3DataDistributionType": "FullyReplicated"
                }
            },
            "ContentType": "libsvm",
            "CompressionType": "None"
        }
    ],
    "OutputDataConfig": {
        "S3OutputPath": "s3://{}/{}/output".format(bucket, prefix)
    },
    "HyperParameters": {
        "max_depth":"5",
        "eta":"0.2",
        "gamma":"4",
        "min_child_weight":"6",
        "subsample":"0.8",
        "silent":"0",
        "objective":"binary:logistic",
        "num_class":"1",
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

```
                "ContentType": "libsvm",
                "CompressionType": "None"
            }
        ],
        "OutputDataConfig": {
            "S3OutputPath": "s3://{}/{}/output".format(bucket, prefix)
        },
        "HyperParameters": {
            "max_depth":"5",
            "eta":"0.2",
            "gamma":"4",
            "min_child_weight":"6",
            "subsample":"0.8",
            "silent":"0",
            "objective":"binary:logistic",
            "num_class":"1",
            "num_round":"100",
            "train_file_name":"train.libsvm",
            "val_file_name":"validation.libsvm"
        },
        "StoppingCondition": {
            "MaxRuntimeInSeconds": 60 * 60
        }
}
```

In [ ]:

```
client = boto3.client('sagemaker')
client.create_training_job(**create_training_params)

status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print(status)
client.get_waiter('TrainingJob_Created').wait(TrainingJobName=job_name)
status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print("Training job ended with status: " + status)
if status == 'Failed':
    message = client.describe_training_job(TrainingJobName=job_name)['FailureReason']
    print('Training failed with the following error: {}'.format(message))
    raise Exception('Training job failed')
```

Hosting

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

```
        "max_depth":"5",
        "eta":"0.2",
        "gamma":"4",
        "min_child_weight":"6",
        "subsample":"0.8",
        "silent":"0",
        "objective":"binary:logistic",
        "num_class":"1",
        "num_round":"100",
        "train_file_name":"train.libsvm",
        "val_file_name":"validation.libsvm"
    },
    "StoppingCondition": {
        "MaxRuntimeInSeconds": 60 * 60
    }
}
```

In [ ]:

```
client = boto3.client('sagemaker')
client.create_training_job(**create_training_params)

status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print(status)
client.get_waiter('TrainingJob_Created').wait(TrainingJobName=job_name)
status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print("Training job ended with status: " + status)
if status == 'Failed':
    message = client.describe_training_job(TrainingJobName=job_name)['FailureReason']
    print('Training failed with the following error: {}'.format(message))
    raise Exception('Training job failed')
```

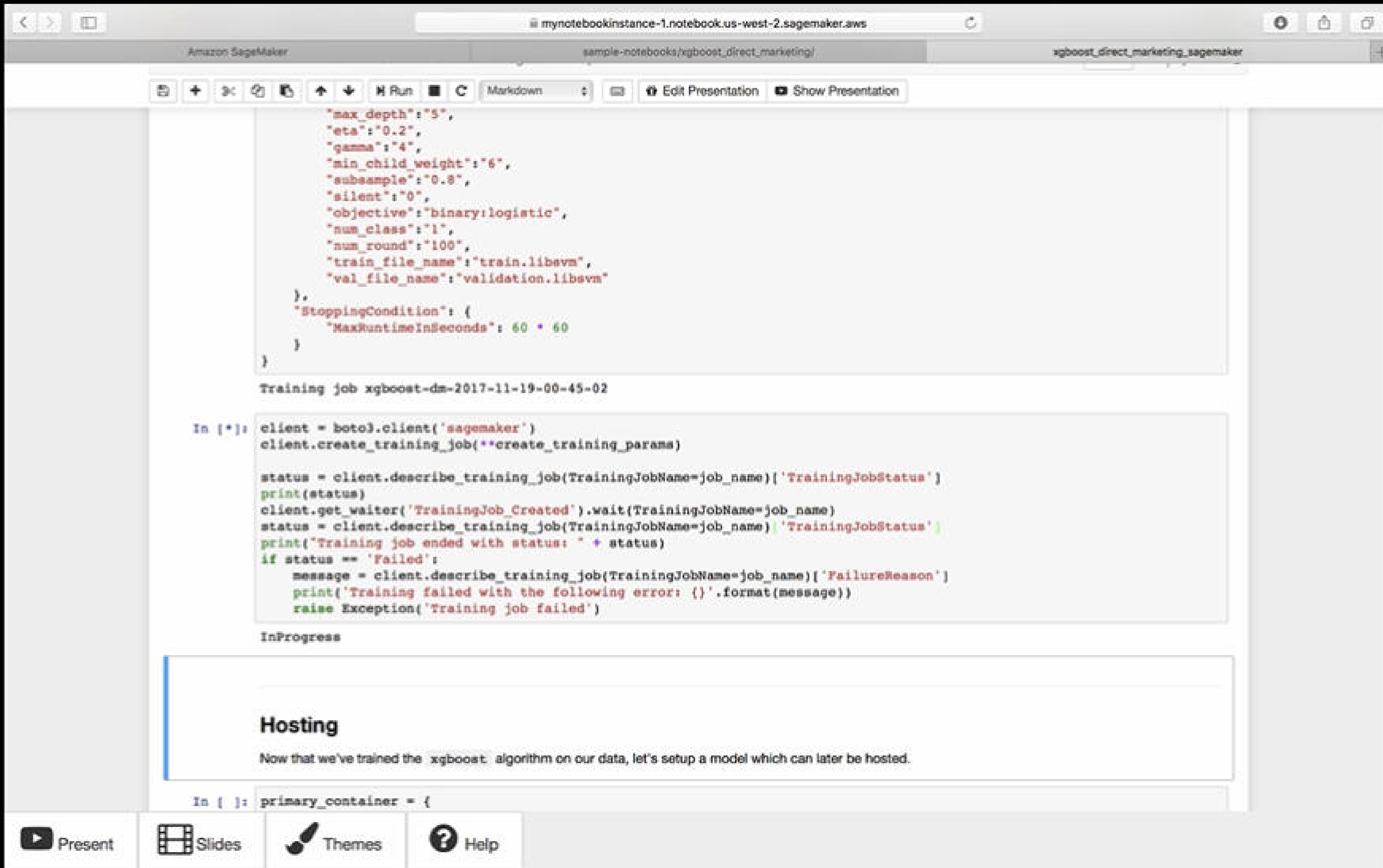
Hosting

Now that we've trained the `xgboost` algorithm on our data, let's setup a model which can later be hosted.

In [ ]:

```
primary_container = {
    'Image': "032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest",
    'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}
```

Present | Slides | Themes | Help


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
Run
Markdown
Edit Presentation
Show Presentation
"max_depth":"5",
"eta":"0.2",
"gamma":"4",
"min_child_weight":"6",
"subsample":"0.8",
"silent":"0",
"objective":"binary:logistic",
"num_class":"1",
"num_round":"100",
"train_file_name":"train.libsvm",
"val_file_name":"validation.libsvm"
},
"StoppingCondition": {
"MaxRuntimeInSeconds": 60 * 60
}
}
Training job xgboost-dm-2017-11-19-00-45-02
In [*]:
client = boto3.client('sagemaker')
client.create_training_job(**create_training_params)
status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print(status)
client.get_waiter('TrainingJob_Created').wait(TrainingJobName=job_name)
status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print("Training job ended with status: " + status)
if status == 'Failed':
message = client.describe_training_job(TrainingJobName=job_name)['FailureReason']
print('Training failed with the following error: {}'.format(message))
raise Exception('Training job failed')
InProgress
Hosting
Now that we've trained the xgboost algorithm on our data, let's setup a model which can later be hosted.
In [ ]:
primary_container = {
Present
Slides
Themes
Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

```
            "eta":"0.2",
            "gamma":"4",
            "min_child_weight":"6",
            "subsample":"0.8",
            "silent":"0",
            "objective":"binary:logistic",
            "num_class":"1",
            "num_round":"100",
            "train_file_name":"train.libsvm",
            "val_file_name":"validation.libsvm"
        },
        "StoppingCondition": {
            "MaxRuntimeInSeconds": 60 * 60
        }
}
```

```
Training job xgboost-dm-2017-11-19-00-45-02
```

In [24]:

```
client = boto3.client('sagemaker')
client.create_training_job(**create_training_params)

status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print(status)
client.get_waiter('TrainingJob_Created').wait(TrainingJobName=job_name)
status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print("Training job ended with status: " + status)
if status == 'Failed':
    message = client.describe_training_job(TrainingJobName=job_name)['FailureReason']
    print('Training failed with the following error: {}'.format(message))
    raise Exception('Training job failed')
```

```
InProgress
Training job ended with status: Completed
```

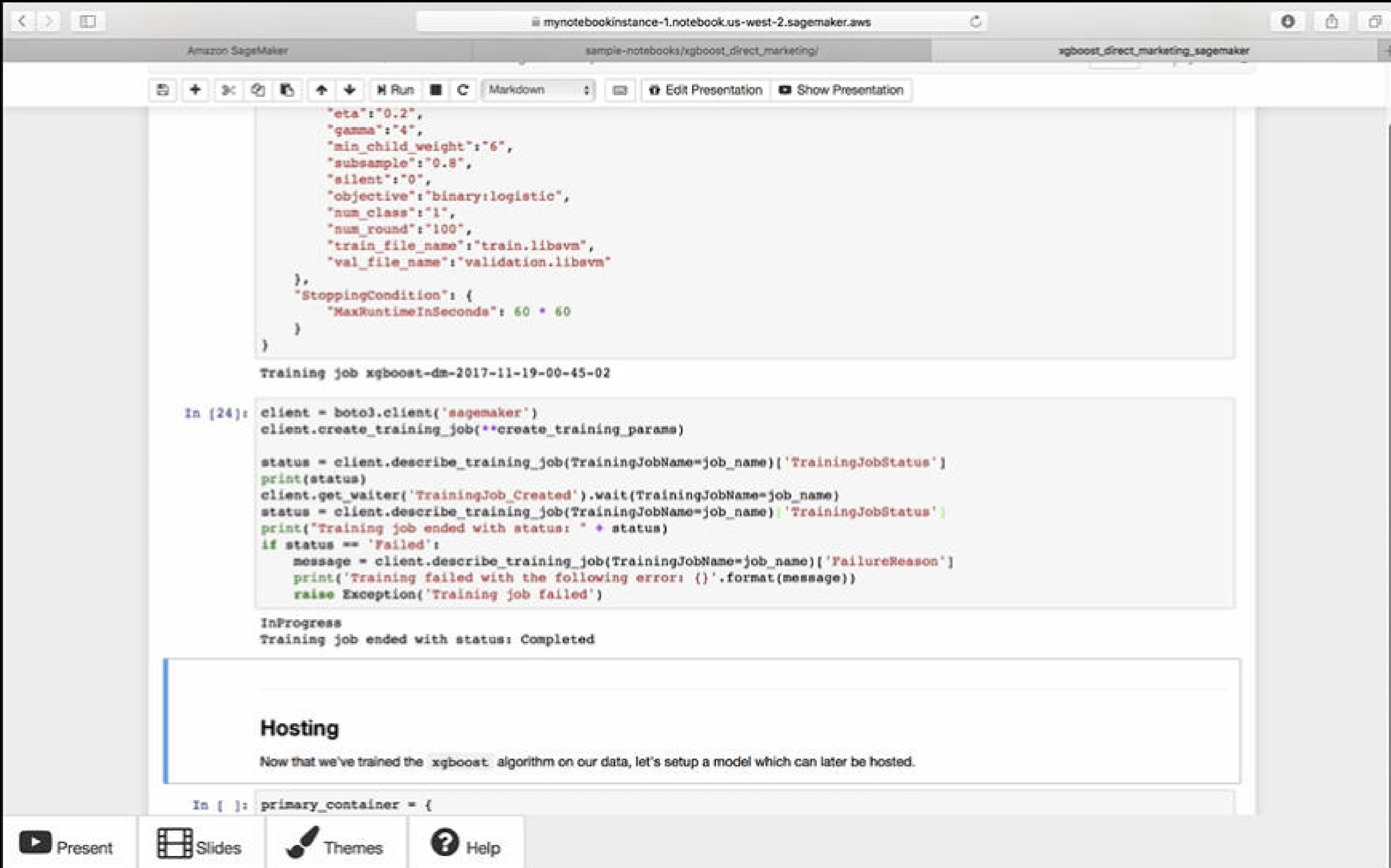
Hosting

Now that we've trained the `xgboost` algorithm on our data, let's setup a model which can later be hosted.

In [ ]:

```
primary_container = {
```

Present | Slides | Themes | Help


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
Run
Markdown
Edit Presentation
Show Presentation
"eta":"0.2",
"gamma":"4",
"min_child_weight":"6",
"subsample":"0.8",
"silent":"0",
"objective":"binary:logistic",
"num_class":"1",
"num_round":"100",
"train_file_name":"train.libsvm",
"val_file_name":"validation.libsvm"
},
"StoppingCondition": {
"MaxRuntimeInSeconds": 60 * 60
}
}
Training job xgboost-dm-2017-11-19-00-45-02
In [24]:
client = boto3.client('sagemaker')
client.create_training_job(**create_training_params)
status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print(status)
client.get_waiter('TrainingJob_Created').wait(TrainingJobName=job_name)
status = client.describe_training_job(TrainingJobName=job_name)['TrainingJobStatus']
print("Training job ended with status: " + status)
if status == 'Failed':
message = client.describe_training_job(TrainingJobName=job_name)['FailureReason']
print('Training failed with the following error: {}'.format(message))
raise Exception('Training job failed')
InProgress
Training job ended with status: Completed
Hosting
Now that we've trained the xgboost algorithm on our data, let's setup a model which can later be hosted.
In [ ]:
primary_container = {
Present
Slides
Themes
Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

Hosting

Now that we've trained the `xgboost` algorithm on our data, let's setup a model which can later be hosted.

In [ ]:

```
primary_container = {
    'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
    'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}

create_model_response = client.create_model(
    ModelName = job_name,
    ExecutionRoleArn = role,
    PrimaryContainer = primary_container)

print(create_model_response['ModelArn'])
```

In [ ]:

```
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType': 'ml.c4.xlarge',
        'InitialInstanceCount': 1,
        'ModelName': job_name,
        'VariantName': 'AllTraffic'}])

print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
```

Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.

In [ ]:

```
endpoint_name = 'xgboost-endpoint-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_name)
create_endpoint_response = client.create_endpoint(
    EndpointName=endpoint_name,
    EndpointConfigName=endpoint_config_name)
print(create_endpoint_response['EndpointArn'])
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

Hosting

Now that we've trained the `xgboost` algorithm on our data, let's setup a model which can later be hosted.

In [25]:

```
primary_container = {
    'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
    'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}

create_model_response = client.create_model(
    ModelName = job_name,
    ExecutionRoleArn = role,
    PrimaryContainer = primary_container)

print(create_model_response['ModelArn'])
```

```
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
```

In [ ]:

```
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType': 'ml.c4.xlarge',
        'InitialInstanceCount': 1,
        'ModelName': job_name,
        'VariantName': 'AllTraffic'}])

print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
```

Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.

In [ ]:

```
endpoint_name = 'xgboost-endpoint-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_name)
create_endpoint_response = client.create_endpoint(
    EndpointName=endpoint_name,
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

Hosting

Now that we've trained the `xgboost` algorithm on our data, let's setup a model which can later be hosted.

In [25]:
```
primary_container = {
    'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
    'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}

create_model_response = client.create_model(
    ModelName = job_name,
    ExecutionRoleArn = role,
    PrimaryContainer = primary_container)

print(create_model_response['ModelArn'])
```

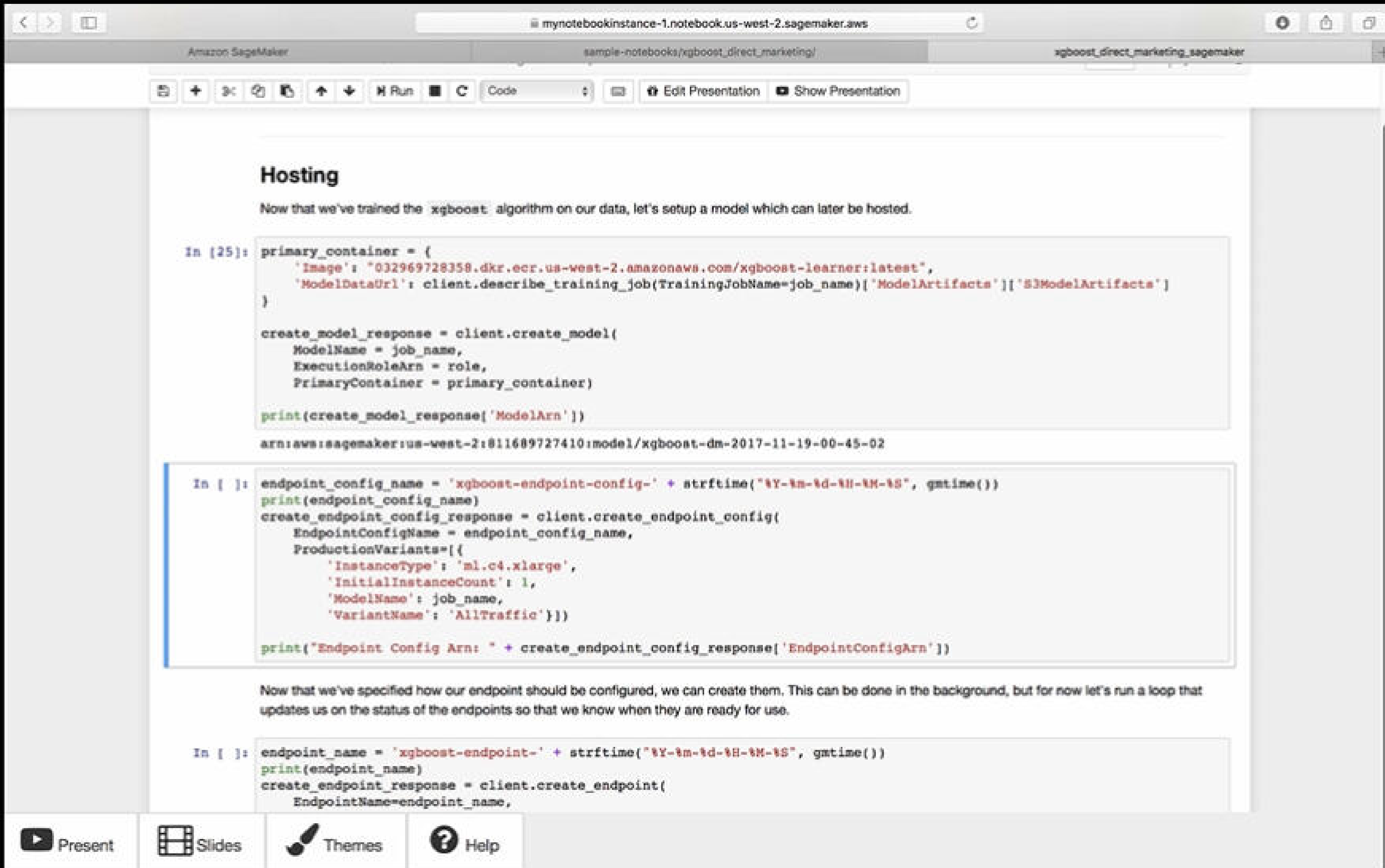
```
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
```

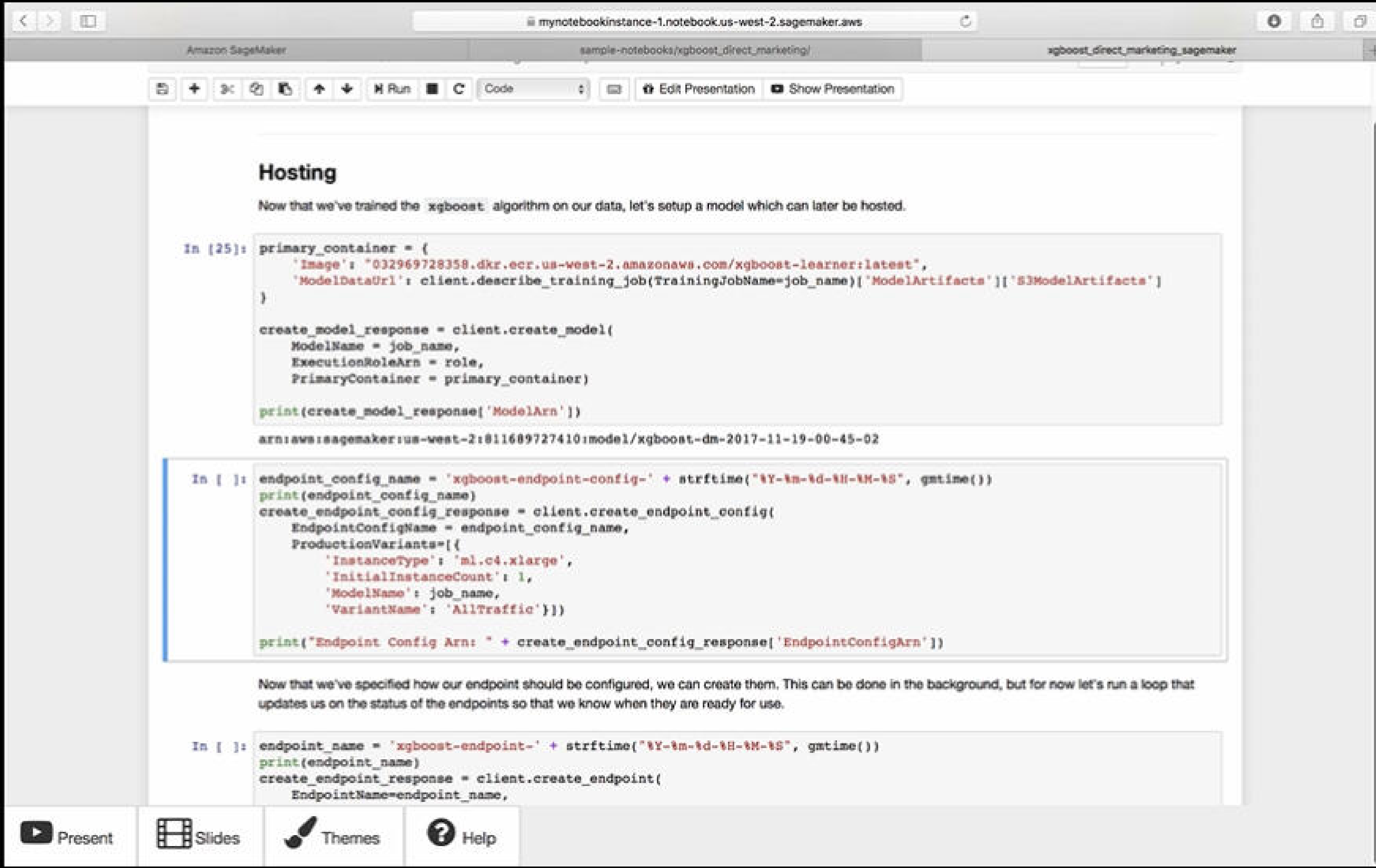
In [ ]:
```
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType': 'ml.c4.xlarge',
        'InitialInstanceCount': 1,
        'ModelName': job_name,
        'VariantName': 'AllTraffic'}])

print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
```

Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.

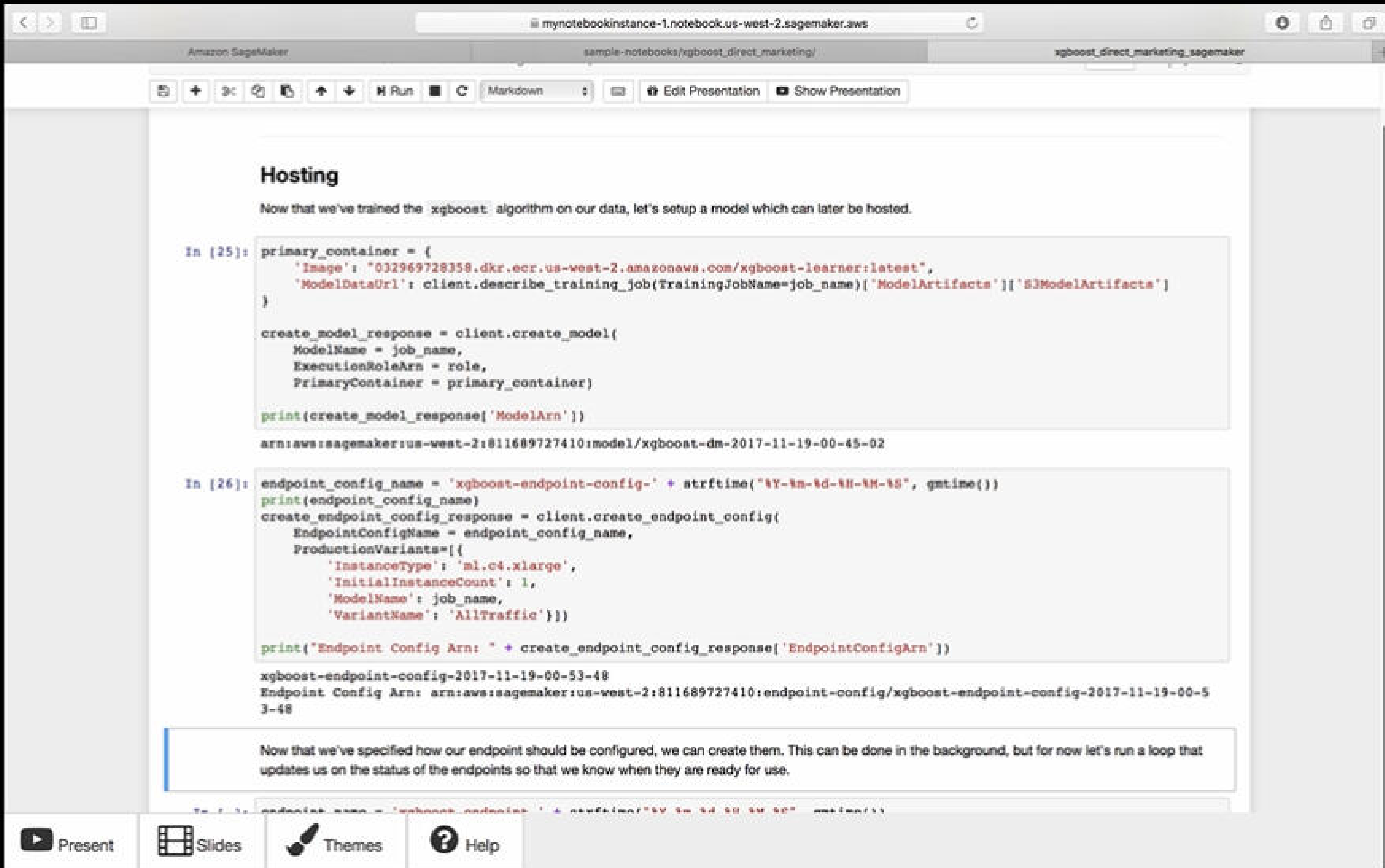
In [ ]:
```
endpoint_name = 'xgboost-endpoint-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_name)
create_endpoint_response = client.create_endpoint(
    EndpointName=endpoint_name,
```

Present | Slides | Themes | Help


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
Run
Code
Edit Presentation
Show Presentation
Hosting
Now that we've trained the xgboost algorithm on our data, let's setup a model which can later be hosted.
In [25]:
primary_container = {
'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}
create_model_response = client.create_model(
ModelName = job_name,
ExecutionRoleArn = role,
PrimaryContainer = primary_container)
print(create_model_response['ModelArn'])
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
In [ ]:
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
EndpointConfigName = endpoint_config_name,
ProductionVariants=[{
'InstanceType': 'ml.c4.xlarge',
'InitialInstanceCount': 1,
'ModelName': job_name,
'VariantName': 'AllTraffic'}])
print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.
In [ ]:
endpoint_name = 'xgboost-endpoint-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_name)
create_endpoint_response = client.create_endpoint(
EndpointName=endpoint_name,
Present
Slides
Themes
Help



mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
Run
Code
Edit Presentation
Show Presentation
Hosting
Now that we've trained the xgboost algorithm on our data, let's setup a model which can later be hosted.
In [25]:
primary_container = {
'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}
create_model_response = client.create_model(
ModelName = job_name,
ExecutionRoleArn = role,
PrimaryContainer = primary_container)
print(create_model_response['ModelArn'])
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
In [ ]:
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
EndpointConfigName = endpoint_config_name,
ProductionVariants=[{
'InstanceType': 'ml.c4.xlarge',
'InitialInstanceCount': 1,
'ModelName': job_name,
'VariantName': 'AllTraffic'}])
print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.
In [ ]:
endpoint_name = 'xgboost-endpoint-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_name)
create_endpoint_response = client.create_endpoint(
EndpointName=endpoint_name,
Present
Slides
Themes
Help



mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
Run
Markdown
Edit Presentation
Show Presentation
Hosting
Now that we've trained the xgboost algorithm on our data, let's setup a model which can later be hosted.
In [25]:
primary_container = {
'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}
create_model_response = client.create_model(
ModelName = job_name,
ExecutionRoleArn = role,
PrimaryContainer = primary_container)
print(create_model_response['ModelArn'])
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
In [26]:
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
EndpointConfigName = endpoint_config_name,
ProductionVariants=[{
'InstanceType': 'ml.c4.xlarge',
'InitialInstanceCount': 1,
'ModelName': job_name,
'VariantName': 'AllTraffic'}])
print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
xgboost-endpoint-config-2017-11-19-00-53-48
Endpoint Config Arn: arn:aws:sagemaker:us-west-2:811689727410:endpoint-config/xgboost-endpoint-config-2017-11-19-00-5
3-48
Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.
Present
Slides
Themes
Help

Hosting

Now that we've trained the `xgboost` algorithm on our data, let's setup a model which can later be hosted.

In [25]:

```
primary_container = {
    'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
    'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}

create_model_response = client.create_model(
    ModelName = job_name,
    ExecutionRoleArn = role,
    PrimaryContainer = primary_container)

print(create_model_response['ModelArn'])
```

```
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
```

In [26]:

```
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType': 'ml.c4.xlarge',
        'InitialInstanceCount': 1,
        'ModelName': job_name,
        'VariantName': 'AllTraffic'}])

print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
```

```
xgboost-endpoint-config-2017-11-19-00-53-48
Endpoint Config Arn: arn:aws:sagemaker:us-west-2:811689727410:endpoint-config/xgboost-endpoint-config-2017-11-19-00-5
3-48
```

Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Markdown | Edit Presentation | Show Presentation

Hosting

Now that we've trained the `xgboost` algorithm on our data, let's setup a model which can later be hosted.

In [25]:
```
primary_container = {
    'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
    'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}

create_model_response = client.create_model(
    ModelName = job_name,
    ExecutionRoleArn = role,
    PrimaryContainer = primary_container)

print(create_model_response['ModelArn'])
```

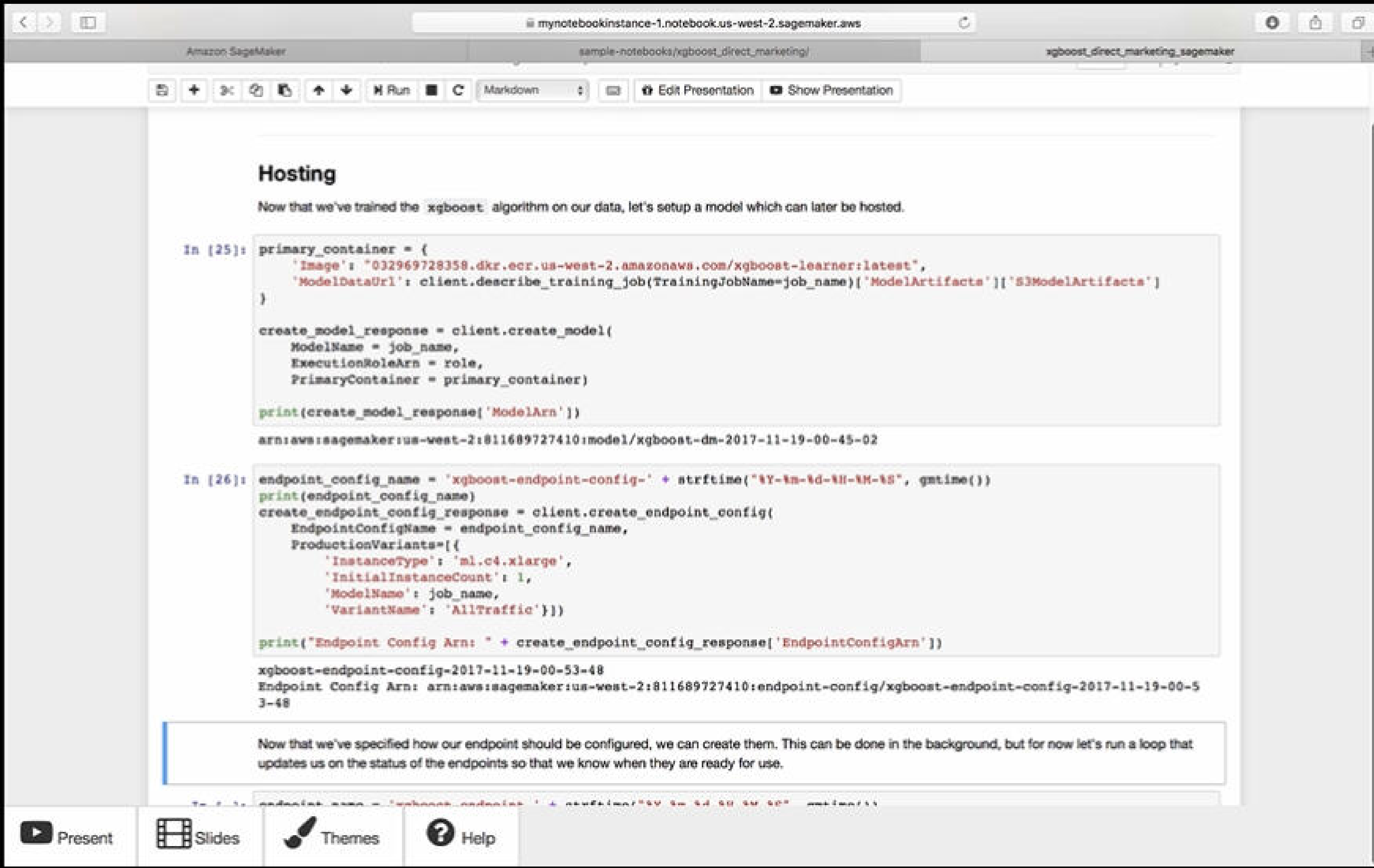
```
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
```

In [26]:
```
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
    EndpointConfigName = endpoint_config_name,
    ProductionVariants=[{
        'InstanceType': 'ml.c4.xlarge',
        'InitialInstanceCount': 1,
        'ModelName': job_name,
        'VariantName': 'AllTraffic'}])

print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
```

```
xgboost-endpoint-config-2017-11-19-00-53-48
Endpoint Config Arn: arn:aws:sagemaker:us-west-2:811689727410:endpoint-config/xgboost-endpoint-config-2017-11-19-00-5
3-48
```

Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.

Present | Slides | Themes | Help


mynotebookinstance-1.notebook.us-west-2.sagemaker.aws
Amazon SageMaker
sample-notebooks/xgboost_direct_marketing/
xgboost_direct_marketing_sagemaker
Run
Markdown
Edit Presentation
Show Presentation
Hosting
Now that we've trained the xgboost algorithm on our data, let's setup a model which can later be hosted.
In [25]:
primary_container = {
'Image': '032969728358.dkr.ecr.us-west-2.amazonaws.com/xgboost-learner:latest',
'ModelDataUrl': client.describe_training_job(TrainingJobName=job_name)['ModelArtifacts']['S3ModelArtifacts']
}
create_model_response = client.create_model(
ModelName = job_name,
ExecutionRoleArn = role,
PrimaryContainer = primary_container)
print(create_model_response['ModelArn'])
arn:aws:sagemaker:us-west-2:811689727410:model/xgboost-dm-2017-11-19-00-45-02
In [26]:
endpoint_config_name = 'xgboost-endpoint-config-' + strftime("%Y-%m-%d-%H-%M-%S", gmtime())
print(endpoint_config_name)
create_endpoint_config_response = client.create_endpoint_config(
EndpointConfigName = endpoint_config_name,
ProductionVariants=[{
'InstanceType': 'ml.c4.xlarge',
'InitialInstanceCount': 1,
'ModelName': job_name,
'VariantName': 'AllTraffic'}])
print("Endpoint Config Arn: " + create_endpoint_config_response['EndpointConfigArn'])
xgboost-endpoint-config-2017-11-19-00-53-48
Endpoint Config Arn: arn:aws:sagemaker:us-west-2:811689727410:endpoint-config/xgboost-endpoint-config-2017-11-19-00-5
3-48
Now that we've specified how our endpoint should be configured, we can create them. This can be done in the background, but for now let's run a loop that updates us on the status of the endpoints so that we know when they are ready for use.
Present
Slides
Themes
Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

Evaluation

There are many ways to compare the performance of a machine learning model, but let's start by simply by comparing actual to predicted values. In this case, we're simply predicting whether the customer subscribed to a term deposit (`1`) or not (`0`), which produces a simple confusion matrix.

In [28]:
```
runtime = boto3.client('sagemaker-runtime')
```

In [29]:
```
def do_predict(data, endpoint_name, content_type):
    payload = '\n'.join(data)
    response = runtime.invoke_endpoint(EndpointName=endpoint_name,
                                       ContentType=content_type,
                                       Body=payload)
    result = response['Body'].read()
    result = result.decode("utf-8")
    result = result.split(',')
    preds = [float((num)) for num in result]
    preds = [round(num) for num in preds]
    return preds

def batch_predict(data, batch_size, endpoint_name, content_type):
    items = len(data)
    arrs = []

    for offset in range(0, items, batch_size):
        if offset+batch_size < items:
            results = do_predict(data[offset:(offset+batch_size)], endpoint_name, content_type)
            arrs.extend(results)
        else:
            arrs.extend(do_predict(data[offset:items], endpoint_name, content_type))
        sys.stdout.write('.')
    return(arrs)
```

In [ ]:
```
%%time
import json

with open('test.libsvm', 'r') as f:
    payload = f.read().strip()

labels = [int(line.split(' ')[0]) for line in payload.split('\n')]
test_data = [line for line in payload.split('\n')]
preds = batch_predict(test_data, 1000, endpoint_name, 'text/x-libsvm')
```

Present | Slides | Themes | Help

mynotebookinstance-1.notebook.us-west-2.sagemaker.aws

Amazon SageMaker | sample-notebooks/xgboost_direct_marketing/ | xgboost_direct_marketing_sagemaker

Run | Code | Edit Presentation | Show Presentation

```python
                                   Body=payload)
    result = response['Body'].read()
    result = result.decode("utf-8")
    result = result.split(',')
    preds = [float((num)) for num in result]
    preds = [round(num) for num in preds]
    return preds

def batch_predict(data, batch_size, endpoint_name, content_type):
    items = len(data)
    arrs = []

    for offset in range(0, items, batch_size):
        if offset+batch_size < items:
            results = do_predict(data[offset:(offset+batch_size)], endpoint_name, content_type)
            arrs.extend(results)
        else:
            arrs.extend(do_predict(data[offset:items], endpoint_name, content_type))
        sys.stdout.write('.')
    return(arrs)
```

In [34]:

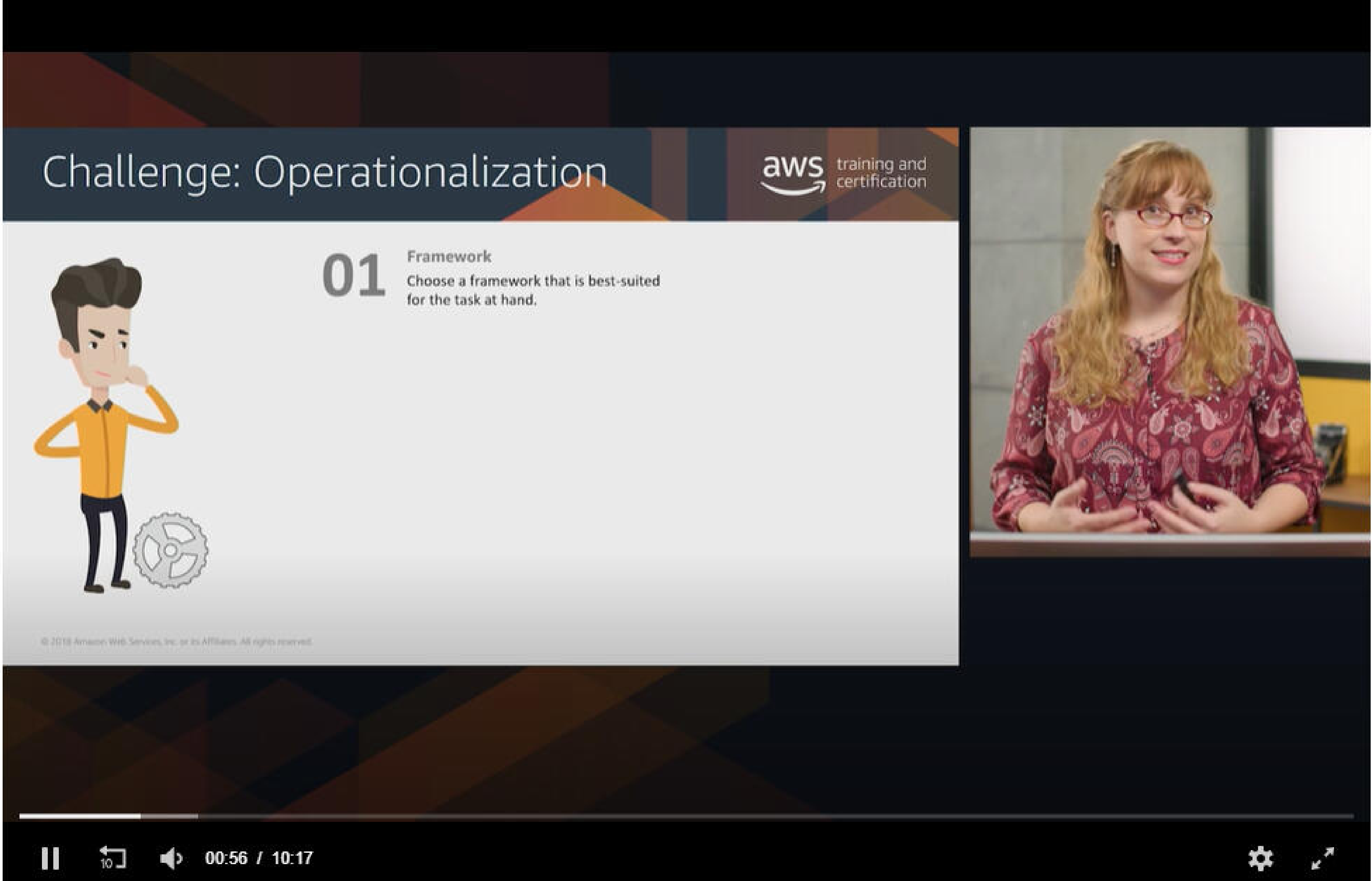
```python
%%time
import json

with open('test.libsvm', 'r') as f:
    payload = f.read().strip()

labels = [int(line.split(' ')[0]) for line in payload.split('\n')]
test_data = [line for line in payload.split('\n')]
preds = batch_predict(test_data, 1000, endpoint_name, 'text/x-libsvm')

print ('\nerror rate=%f' % ( sum(1 for i in range(len(preds)) if preds[i]!=labels[i]) /float(len(preds))))
```

```
.....
error rate=0.095897
CPU times: user 32 ms, sys: 0 ns, total: 32 ms
Wall time: 650 ms
```

In [ ]:

Present | Slides | Themes | Help


00:41 / 10:17


Challenge: Operationalization
aws training and certification
01
Framework
Choose a framework that is best-suited for the task at hand.
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.
00:56 / 10:17

Challenge: Operationalization

01 Framework

Choose a framework that is best-suited for the task at hand.

02 Models

Build the models using the chosen framework.

03 Train Models to Make Predictions

Train a model using sample data to make accurate predictions on bigger data sets.

Challenge: Operationalization

01 Framework

Choose a framework that is best-suited for the task at hand.

02 Models

Build the models using the chosen framework.

03 Train Models to Make Predictions

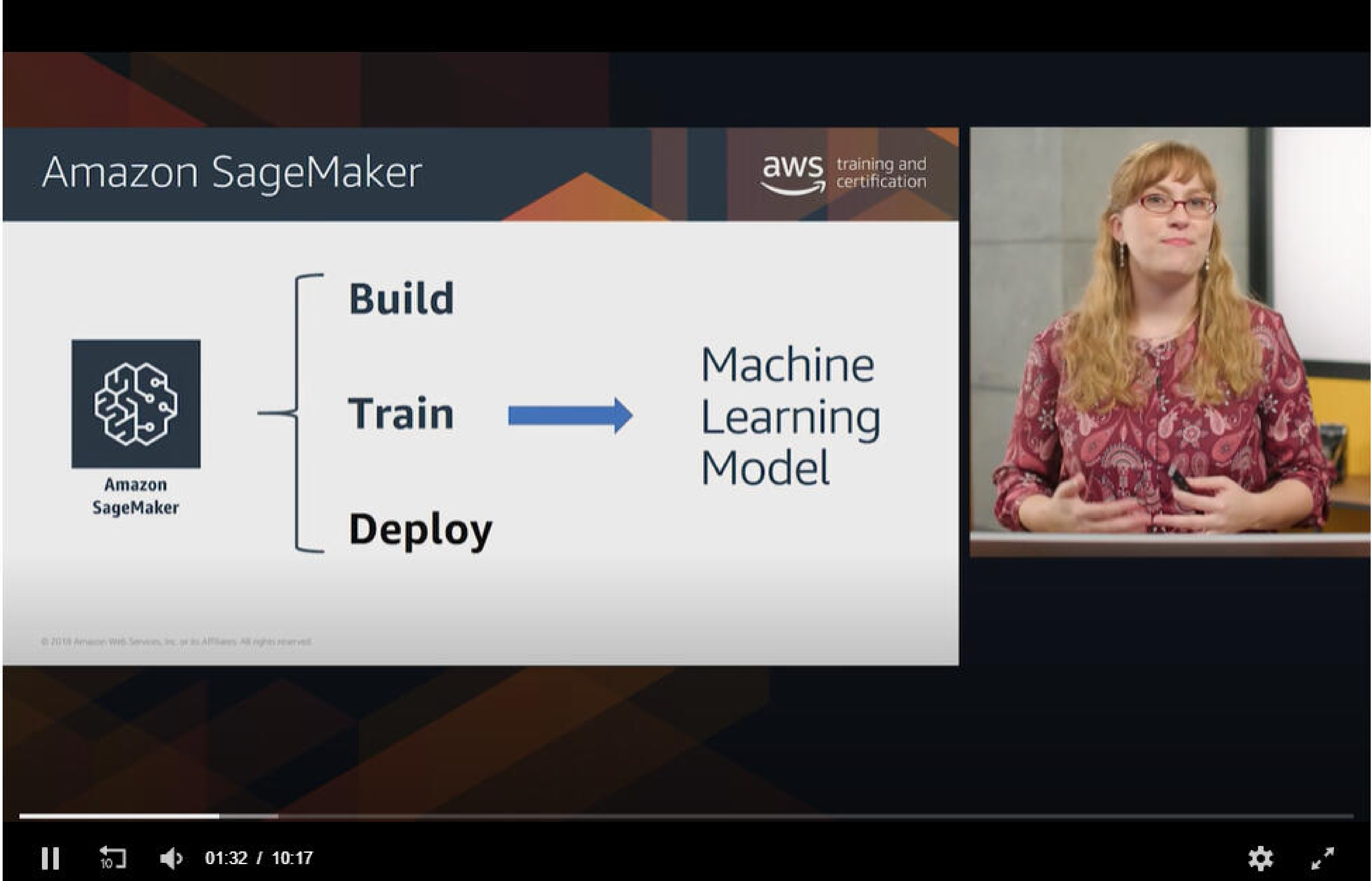
Train a model using sample data to make accurate predictions on bigger data sets.

04 Integrate

Integrate the model with the application.

05 Deploy

Deploy the application, the model, and the framework on a platform.



Amazon SageMaker
aws training and certification
Amazon SageMaker
Build
Train
Deploy
Machine Learning Model
01:32 / 10:17

Amazon SageMaker

aws training and certification

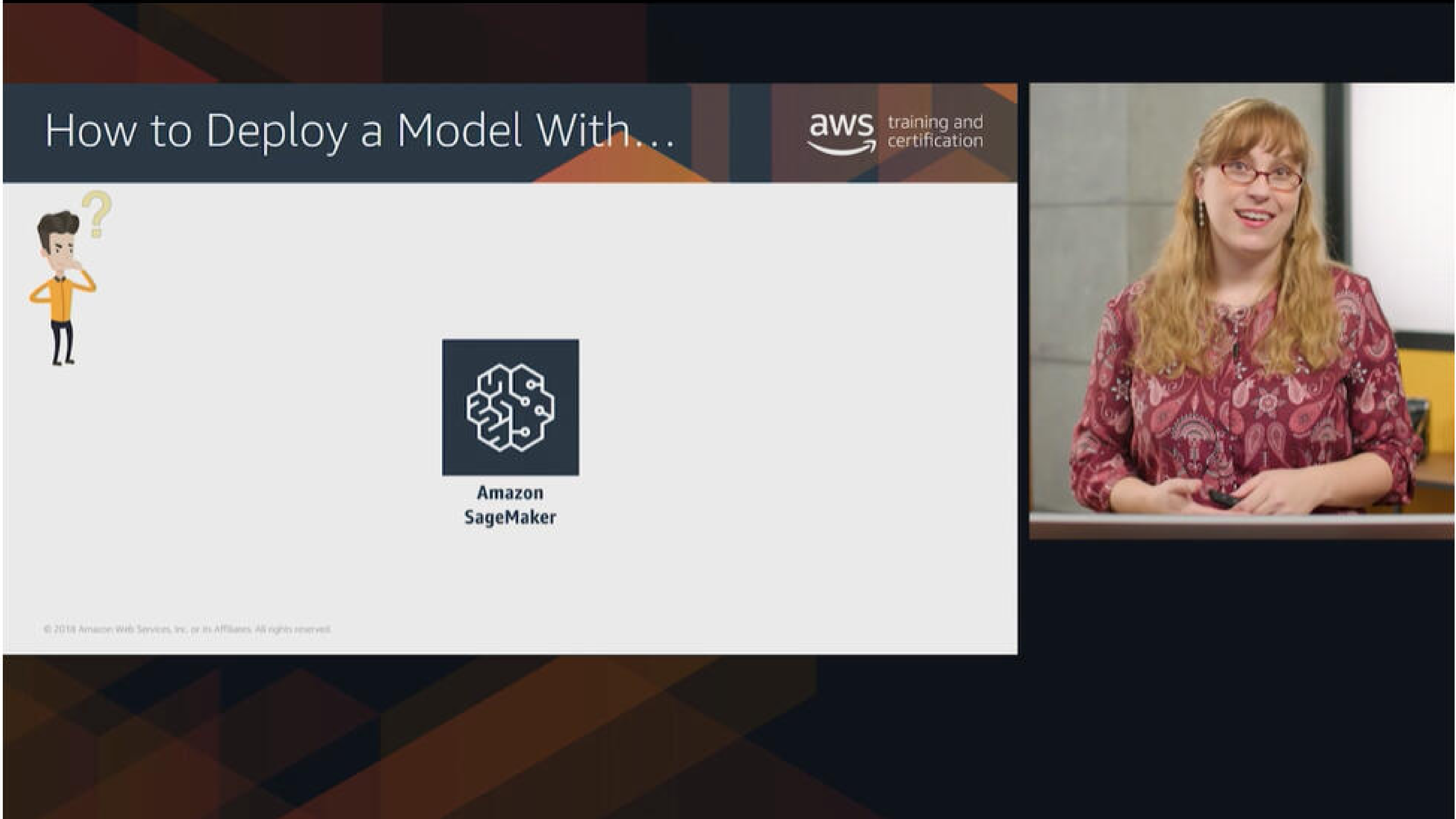
Amazon SageMaker

Build

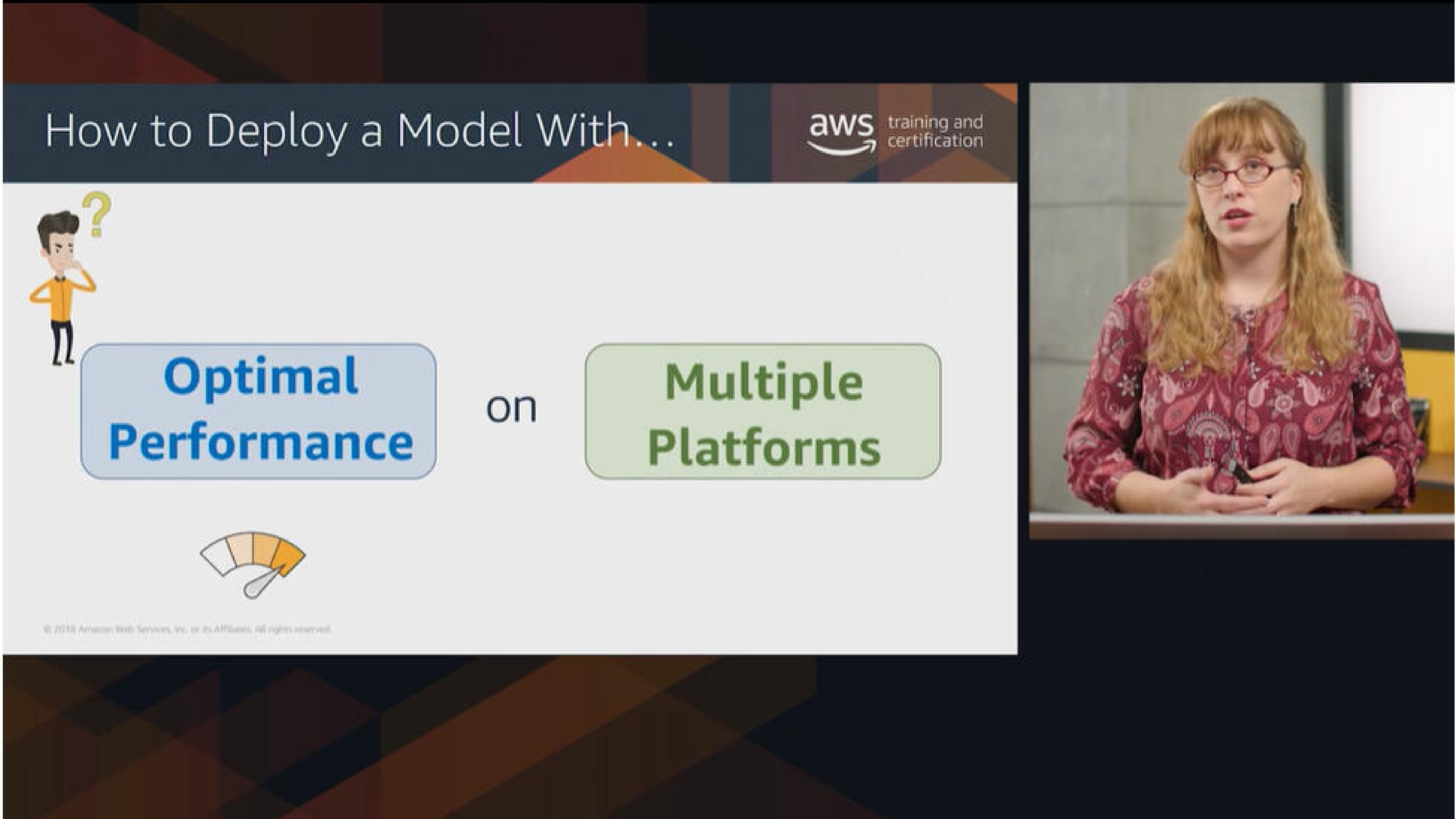
Train

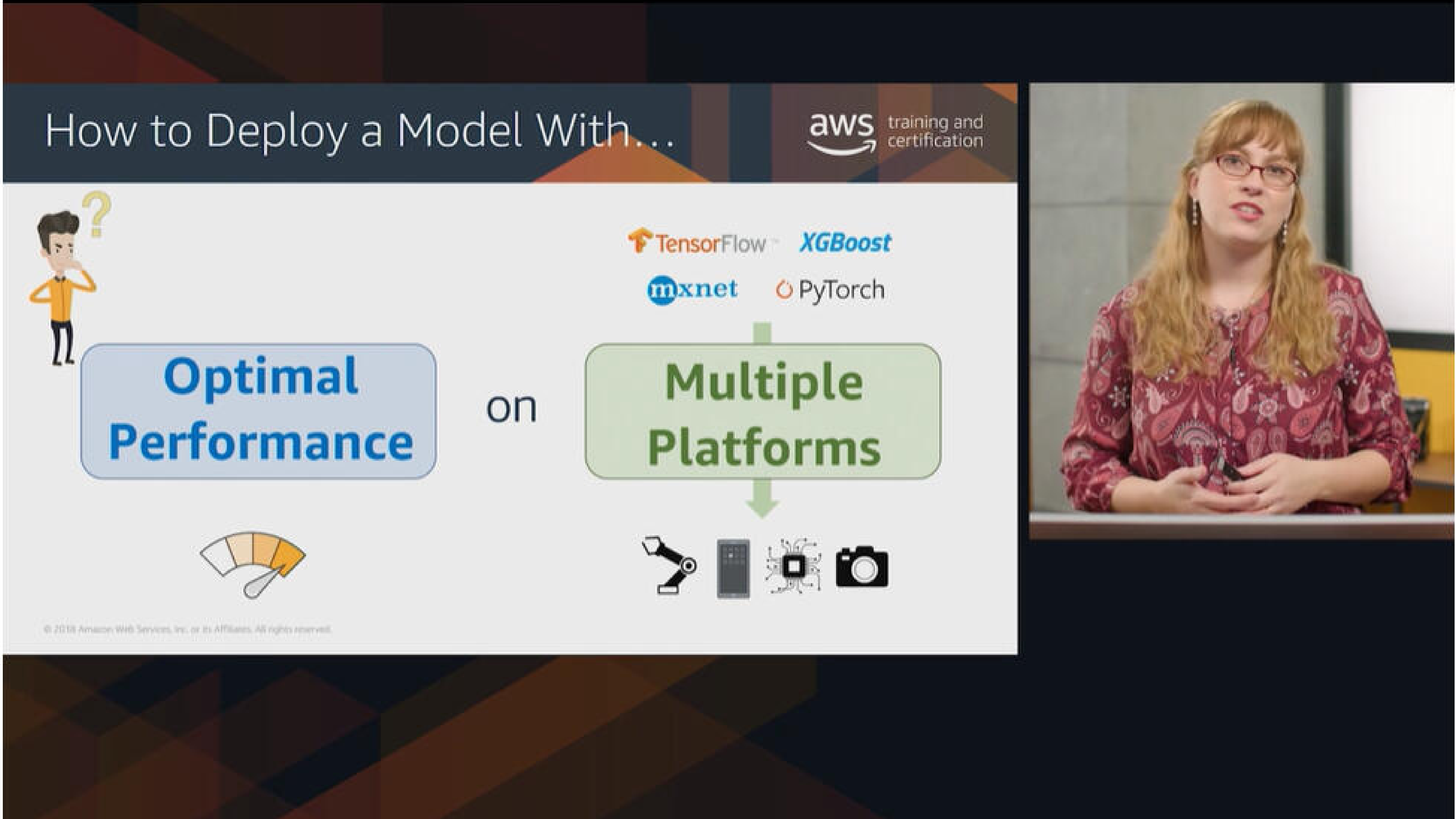
Deploy

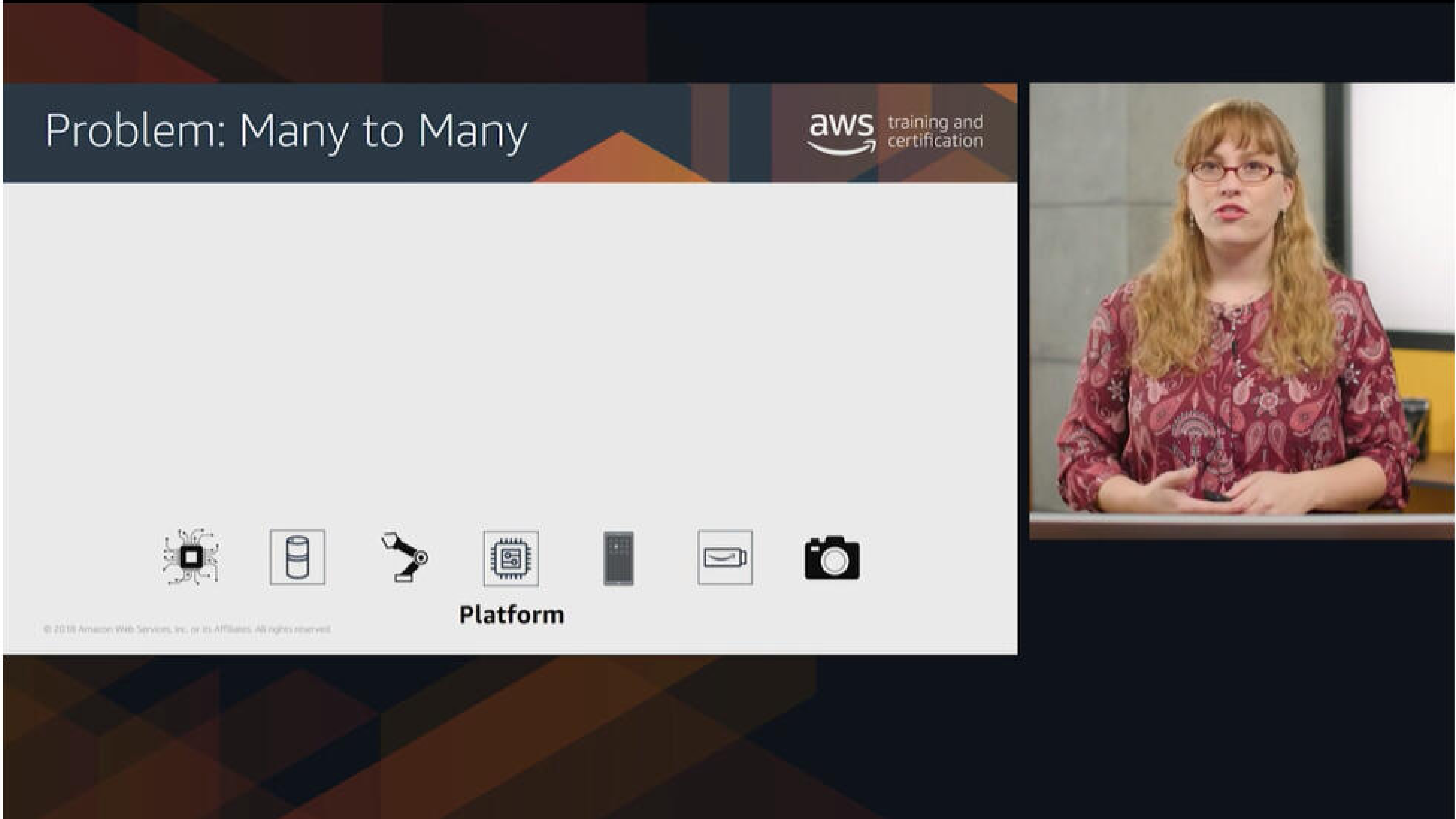
→ Machine Learning Model



How to Deploy a Model With…
aws training and certification
Amazon SageMaker


How to Deploy a Model With…
aws training and certification
Optimal Performance
on
Multiple Platforms
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.


How to Deploy a Model With...
aws training and certification
TensorFlow
XGBoost
mxnet
PyTorch
Optimal Performance
on
Multiple Platforms
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

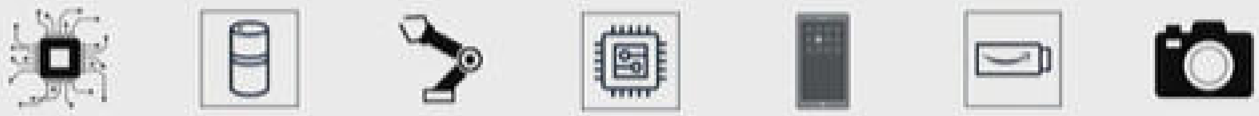

Problem: Many to Many
aws training and certification
Platform
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Problem: Many to Many

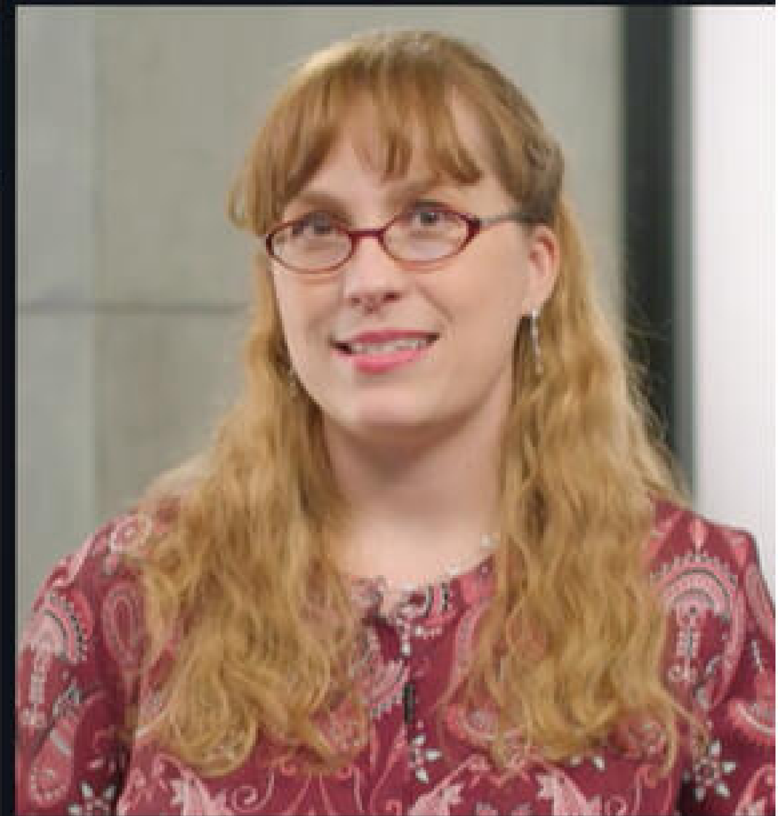
aws training and certification

Framework

TensorFlow mxnet PYTORCH XGBoost

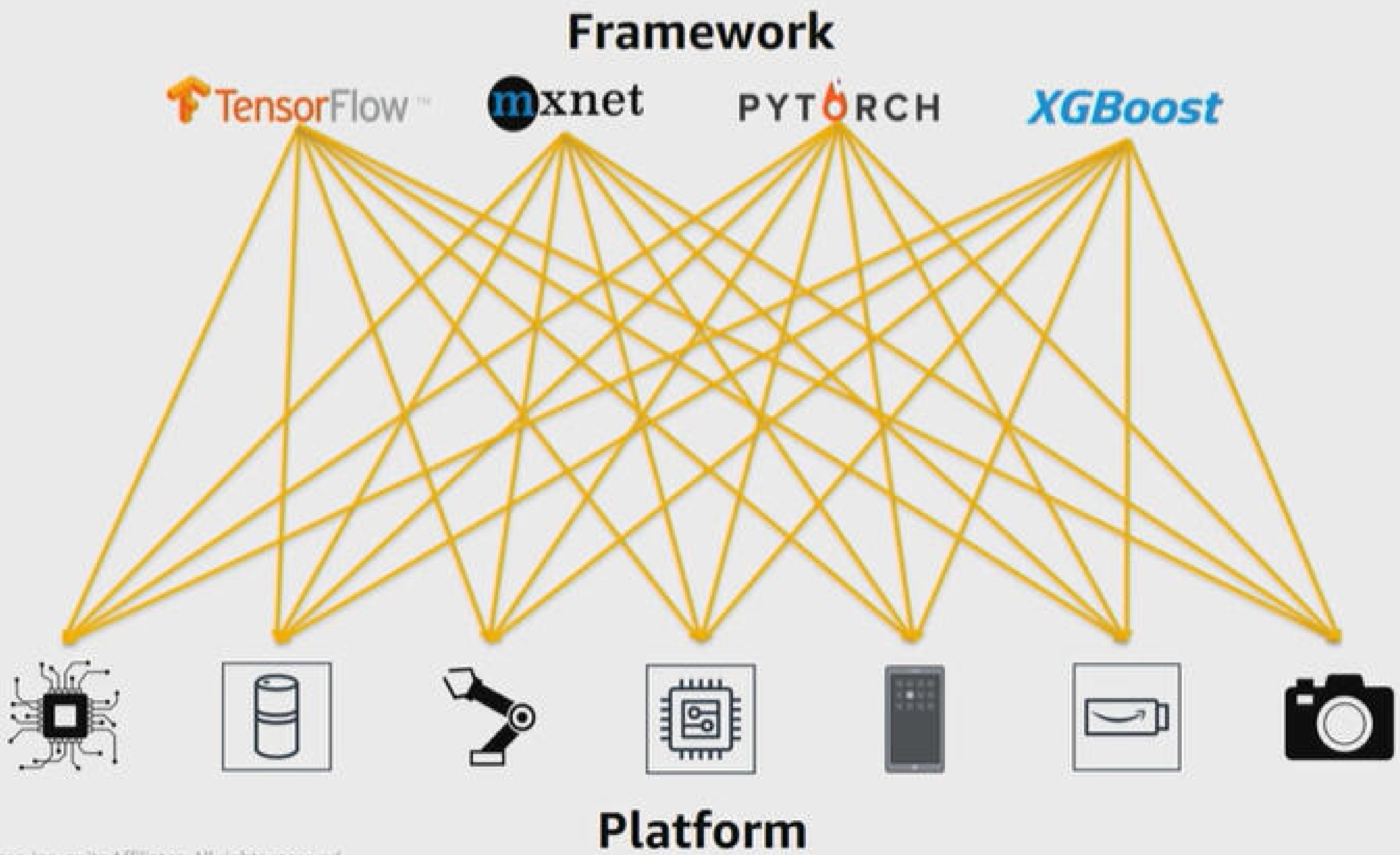


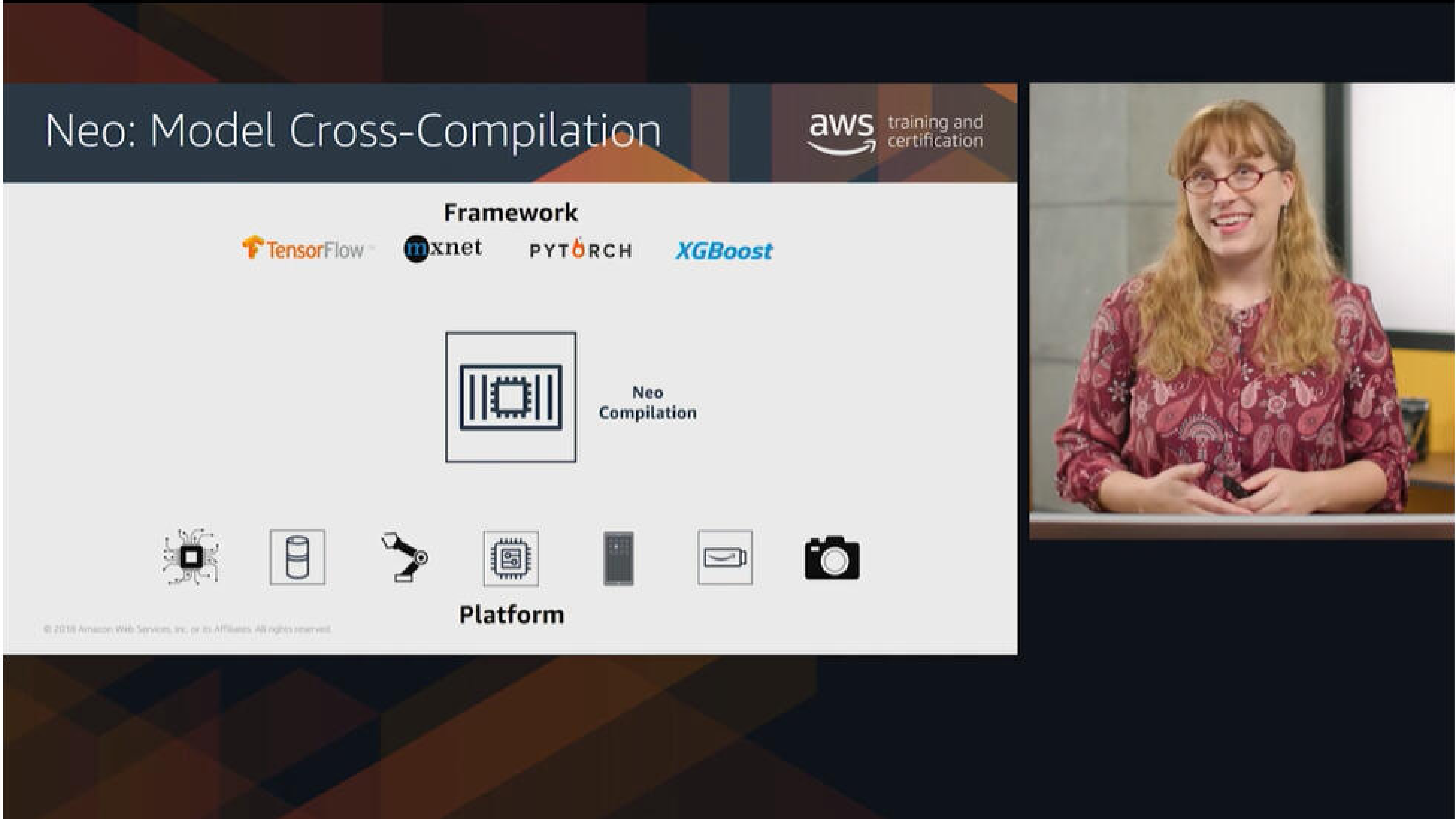
Platform

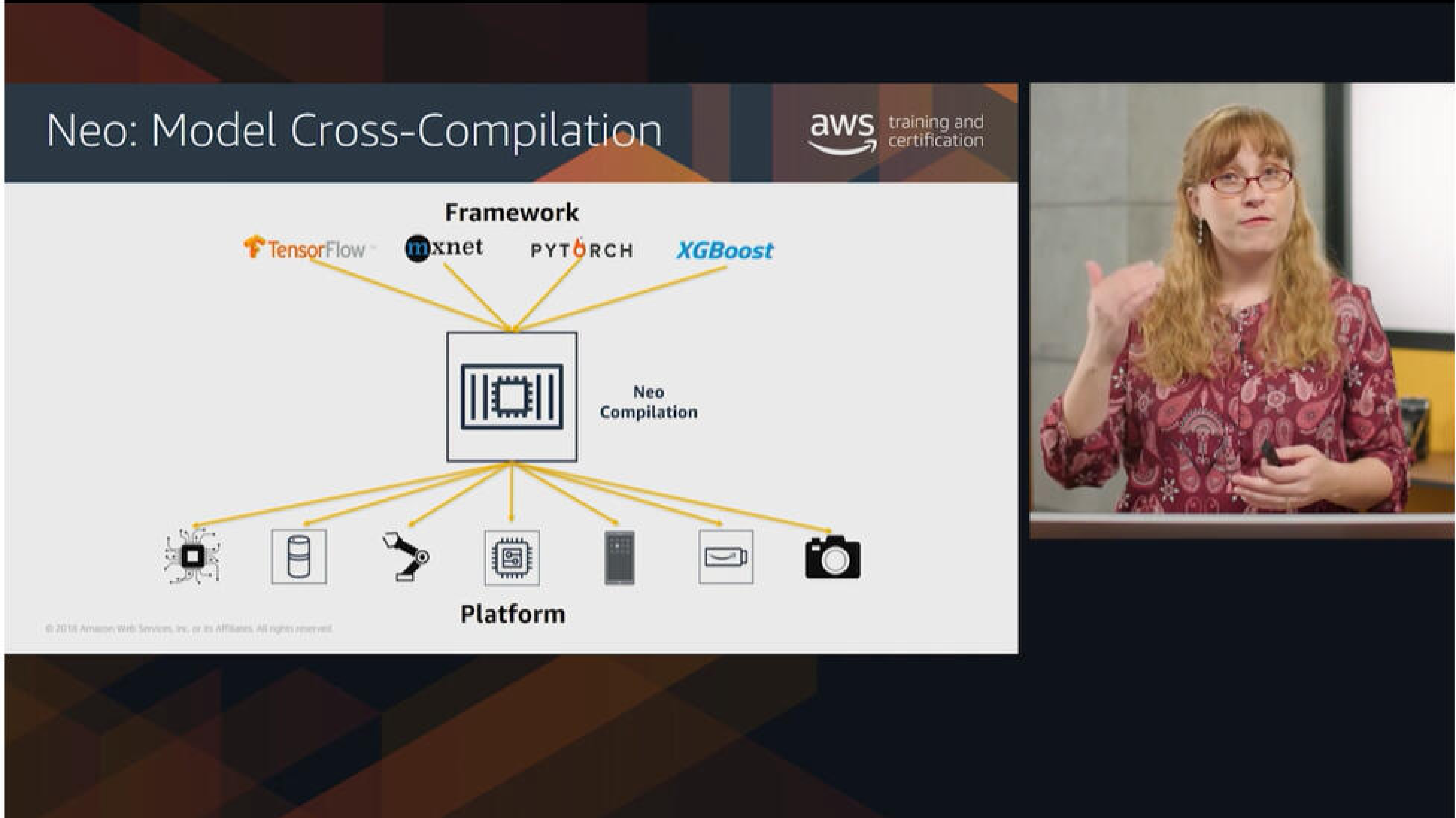


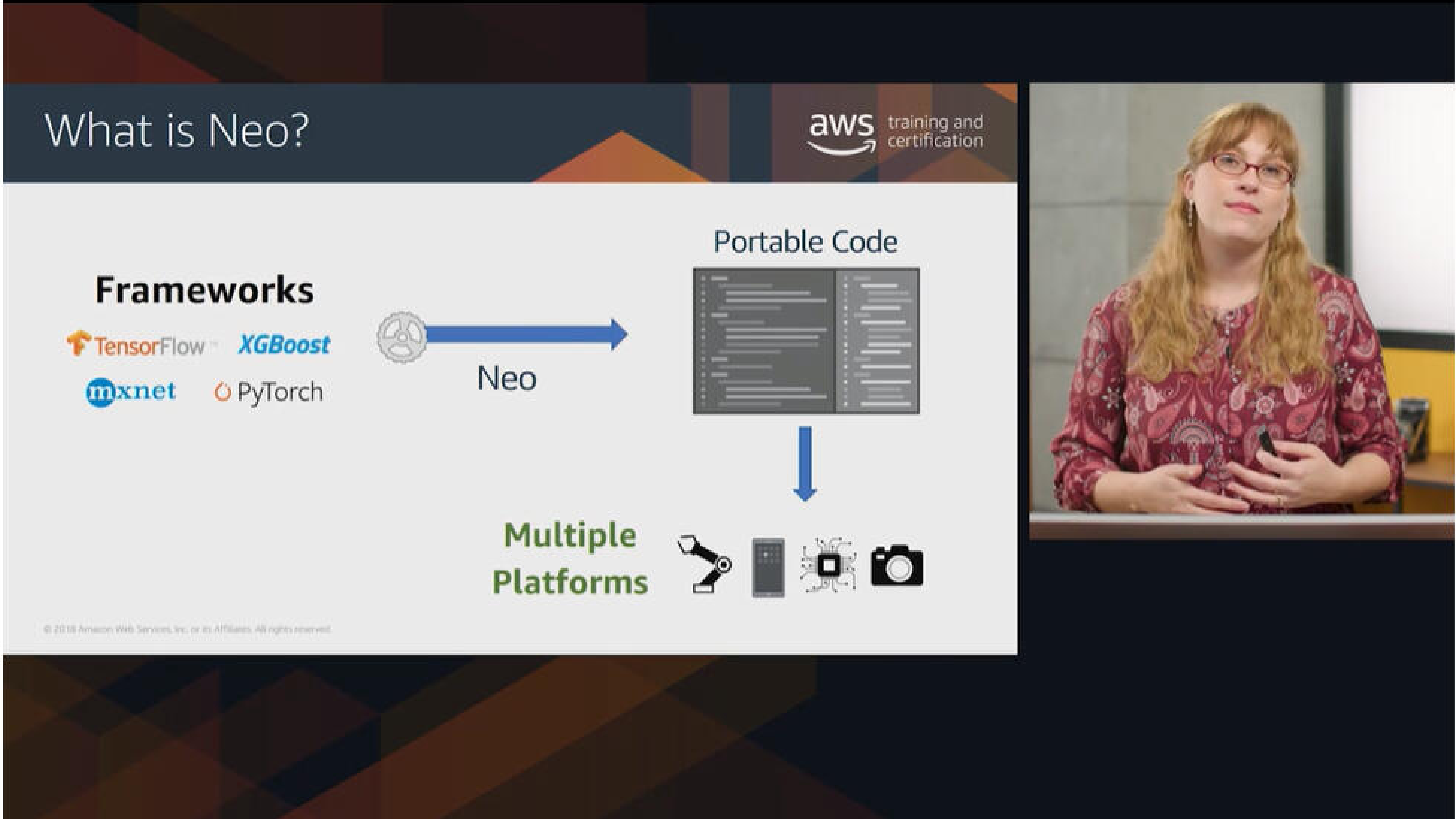
Problem: Many to Many

aws training and certification



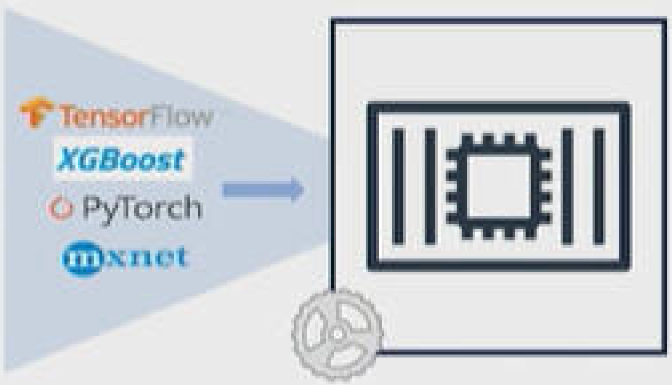

Neo: Model Cross-Compilation
aws training and certification
Framework
TensorFlow
mxnet
PYTORCH
XGBoost
Neo Compilation
Platform
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.


Neo: Model Cross-Compilation
aws training and certification
Framework
TensorFlow
mxnet
PYTORCH
XGBoost
Neo Compilation
Platform

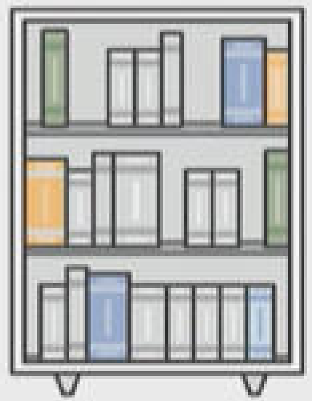

What is Neo?
aws training and certification
Frameworks
TensorFlow
XGBoost
mxnet
PyTorch
Neo
Portable Code
Multiple Platforms
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Neo Components

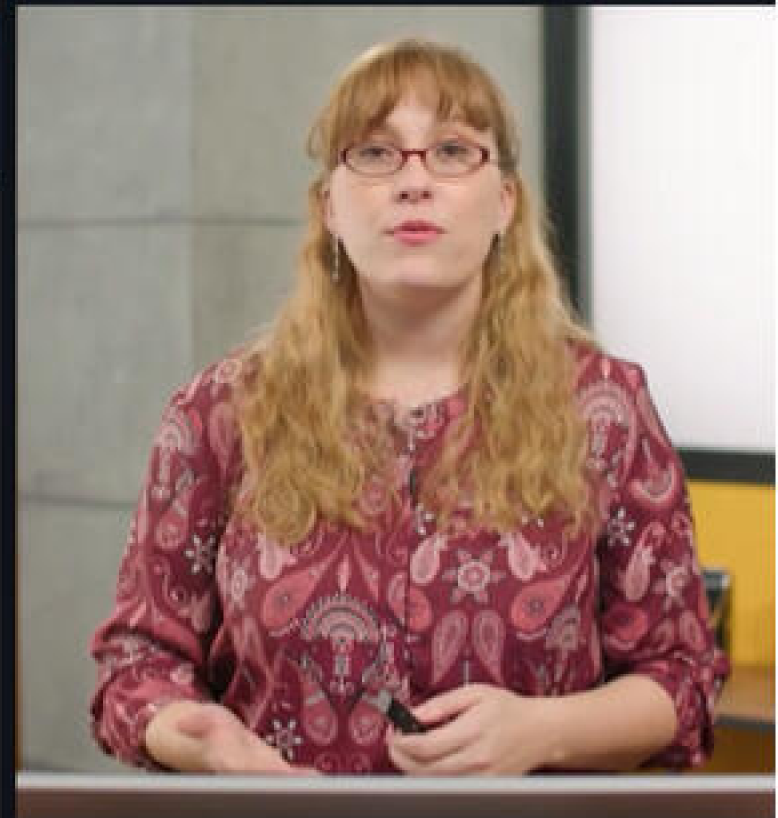
aws training and certification



Neo Compiler Container

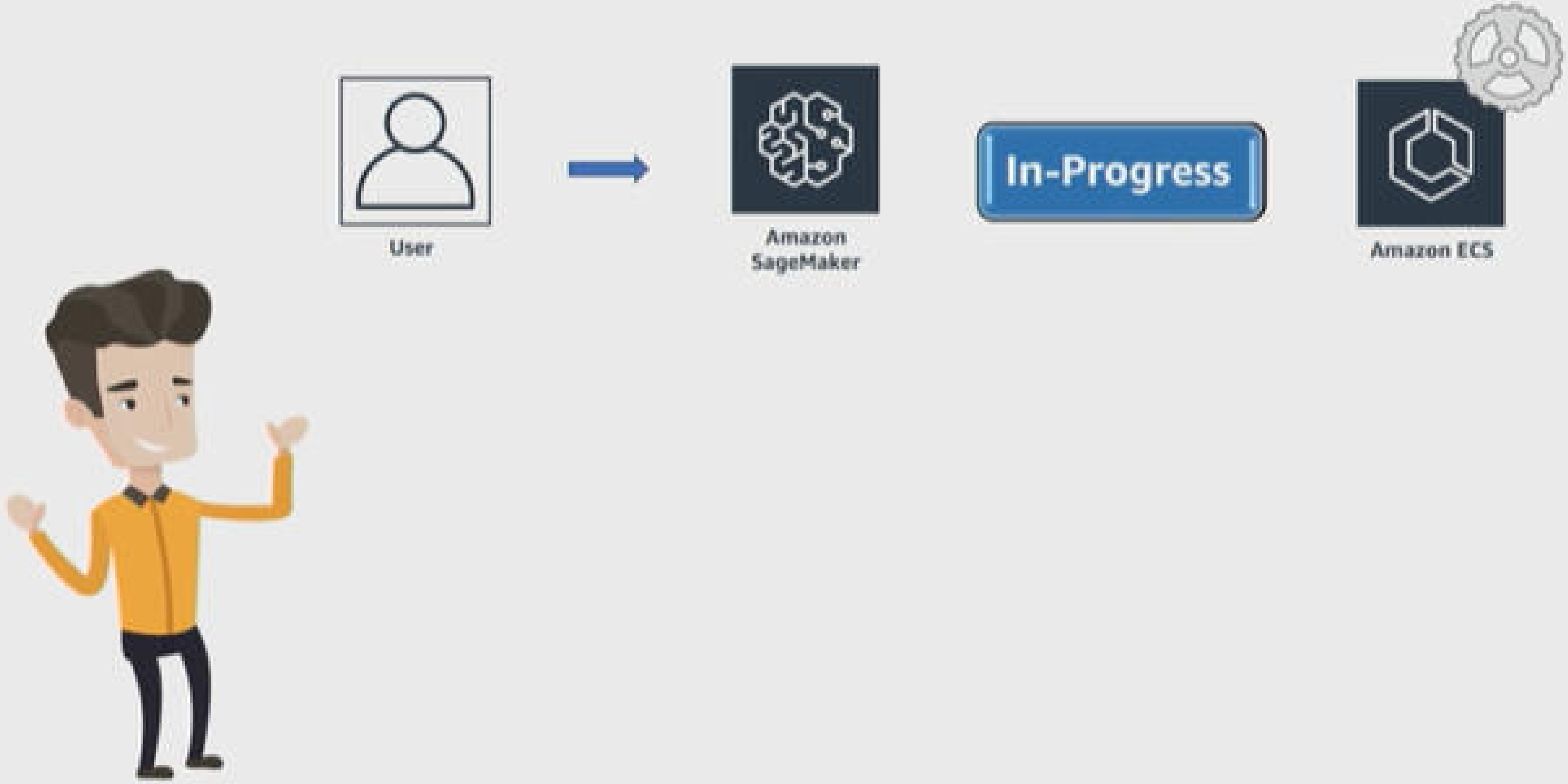


Shared Object Library

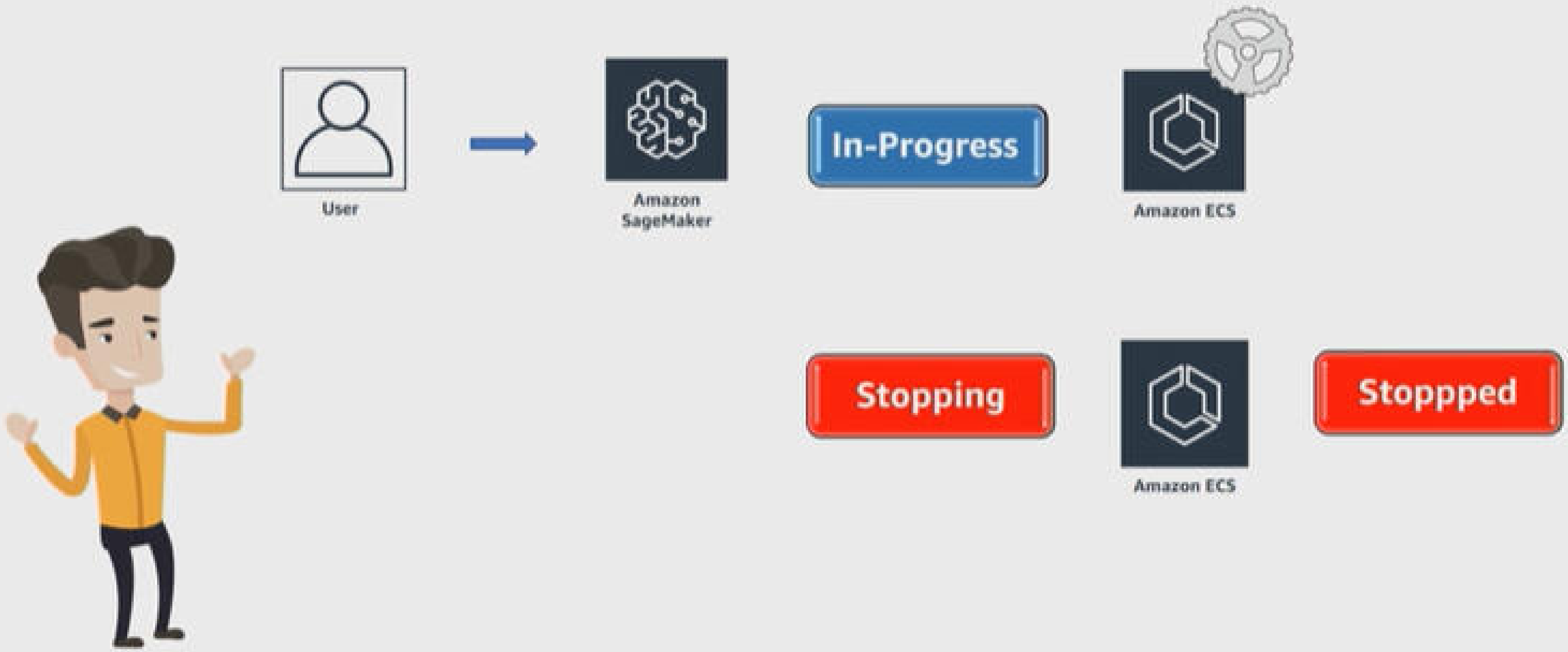


How Does Neo Work?

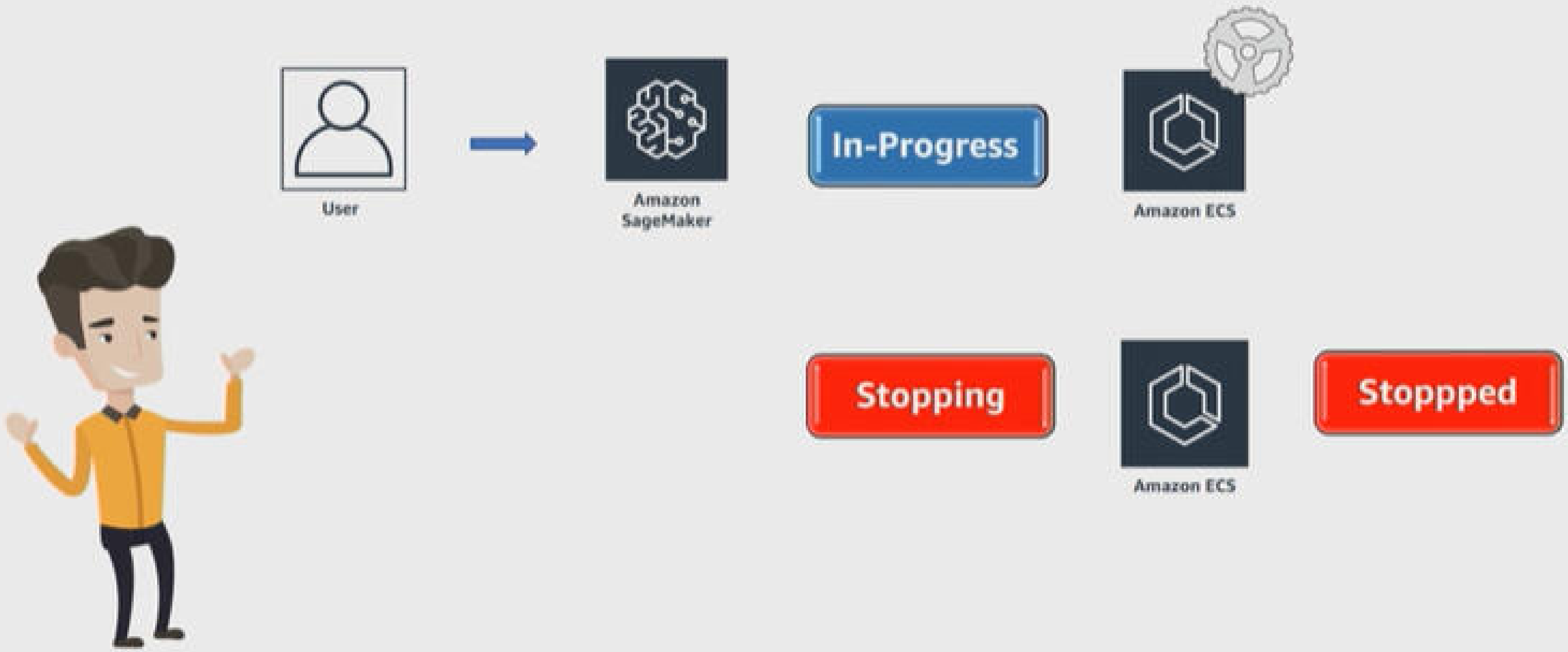
How Does Neo Work?



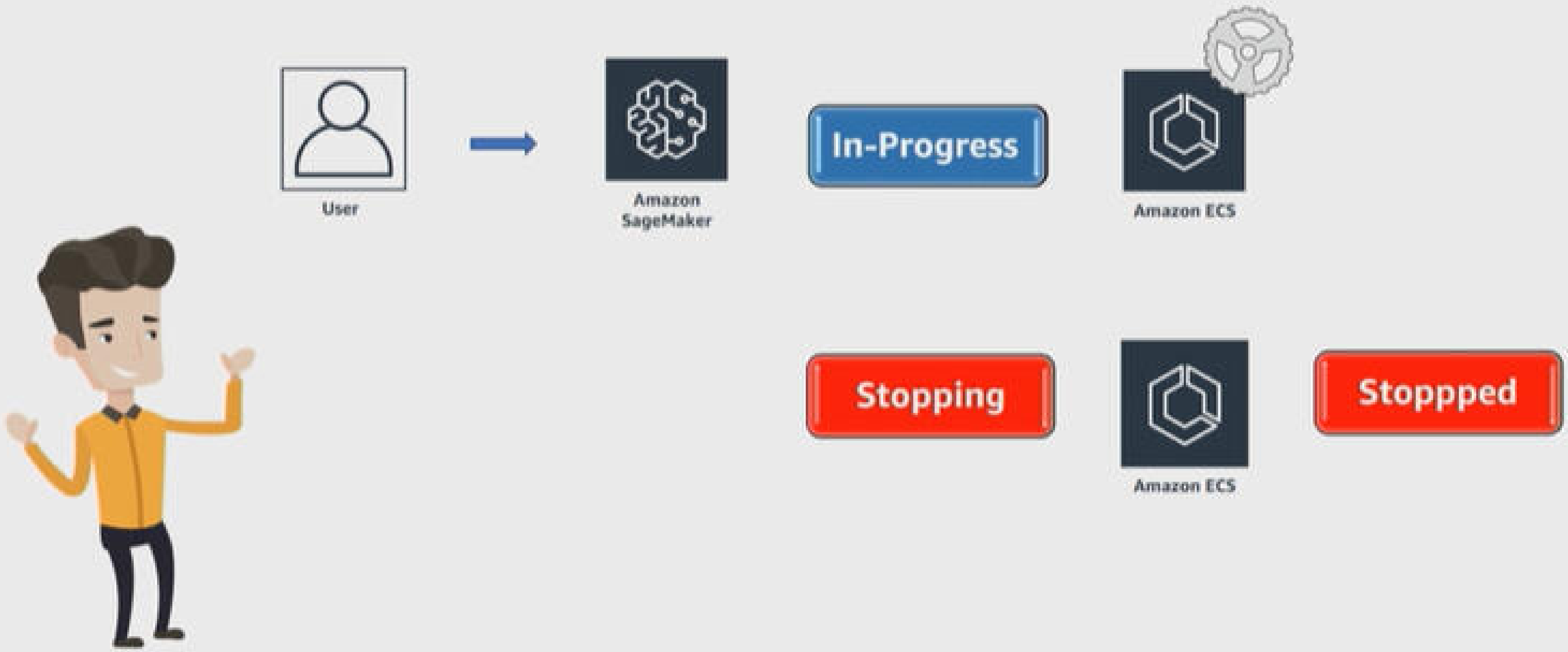
How Does Neo Work?



How Does Neo Work?

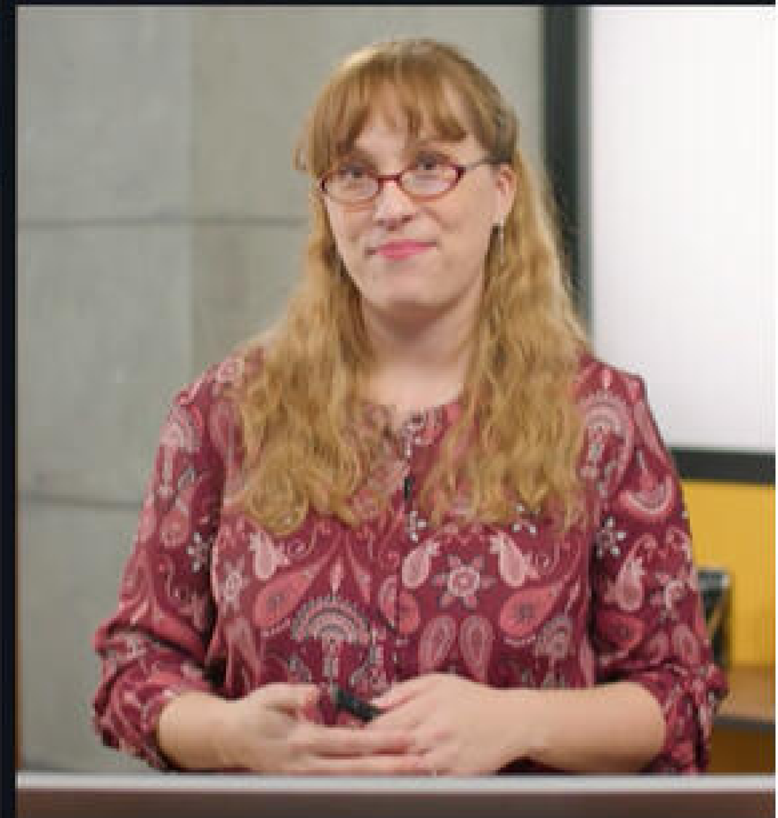


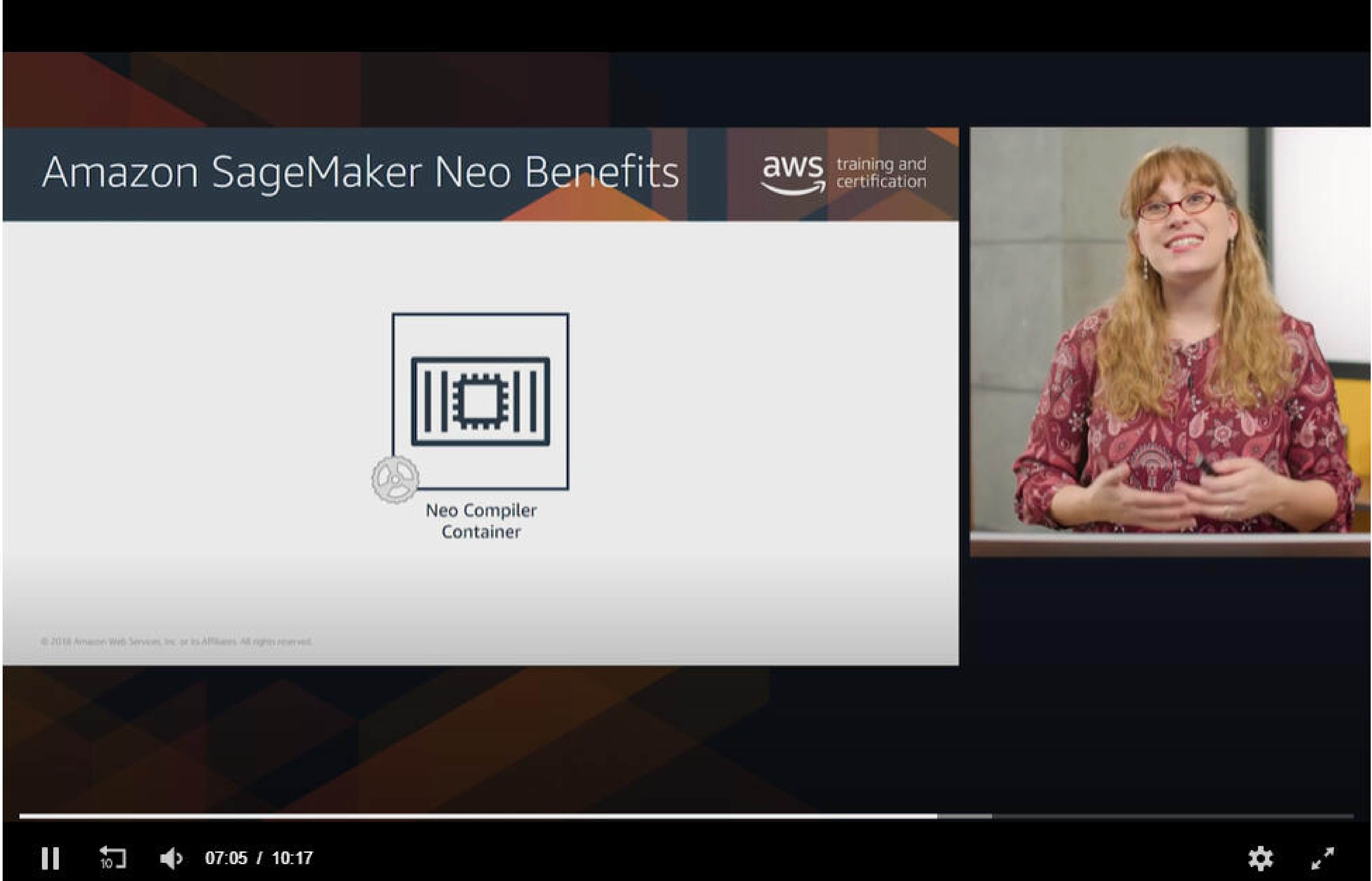
How Does Neo Work?



How Does Neo Work?

aws training and certification



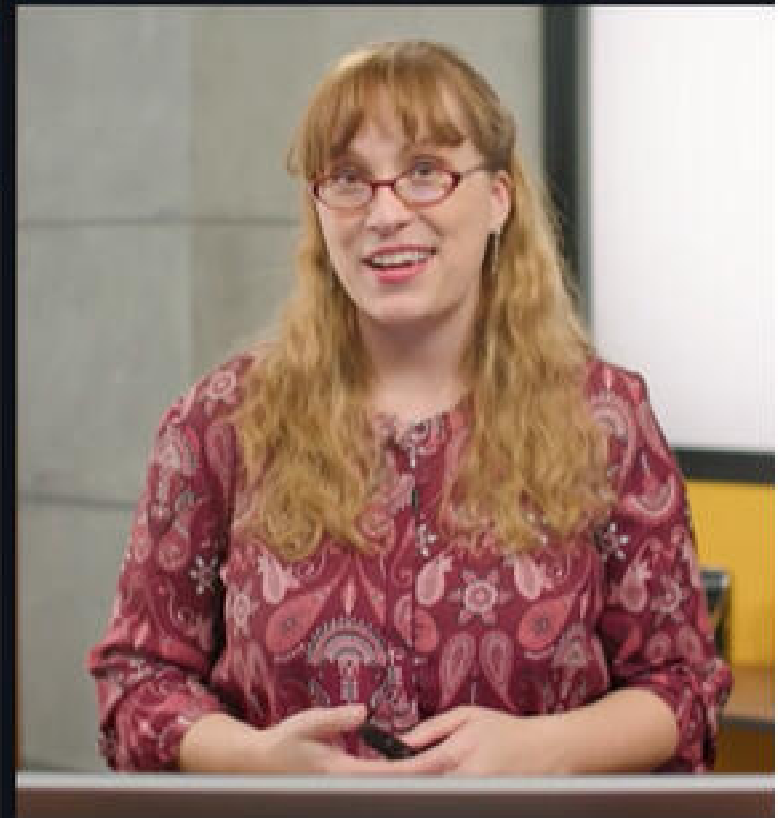

Amazon SageMaker Neo Benefits
aws training and certification
Neo Compiler Container
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.
07:05 / 10:17

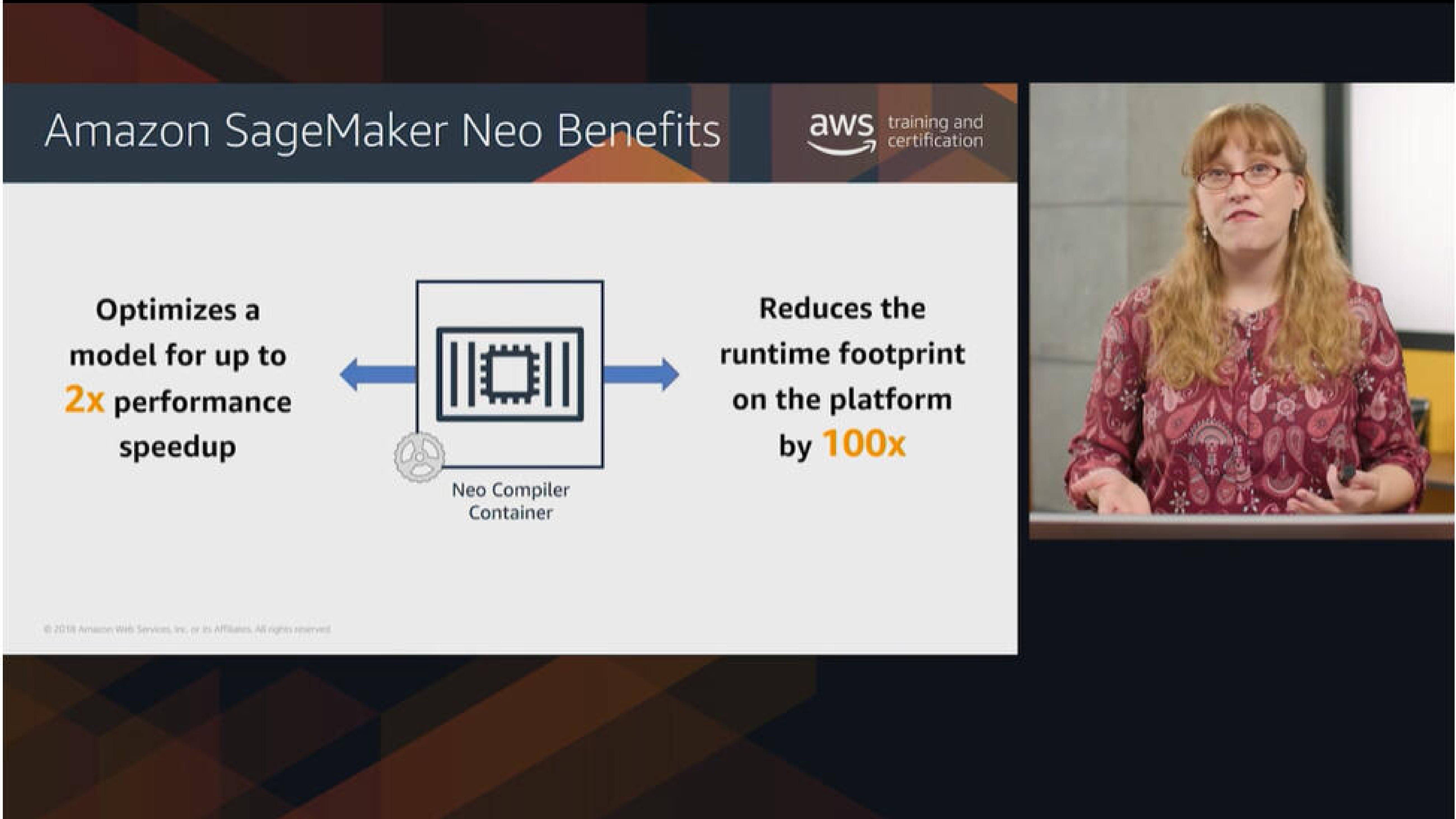
Amazon SageMaker Neo Benefits

aws training and certification

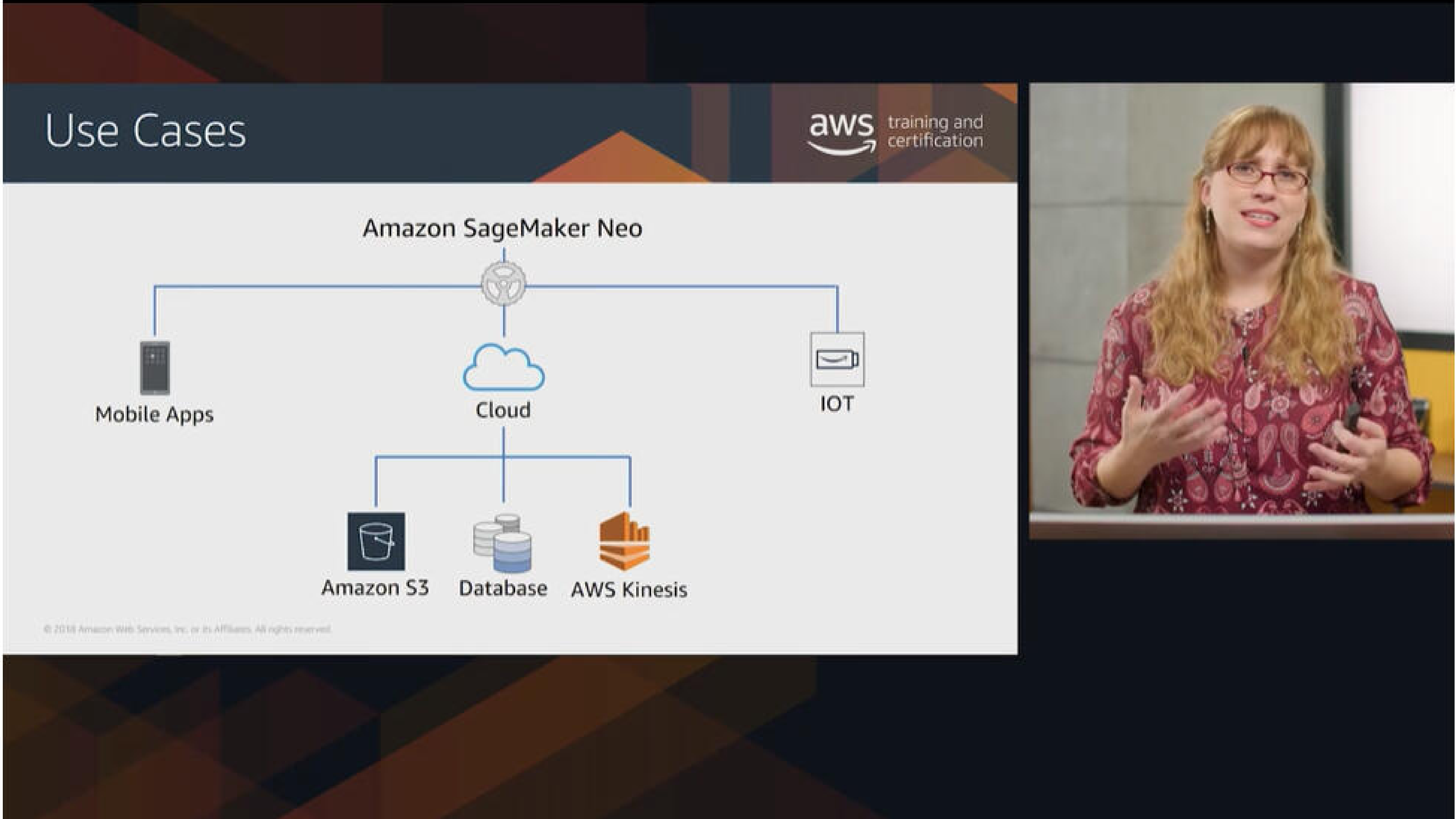
Optimizes a model for up to 2x performance speedup

Neo Compiler Container





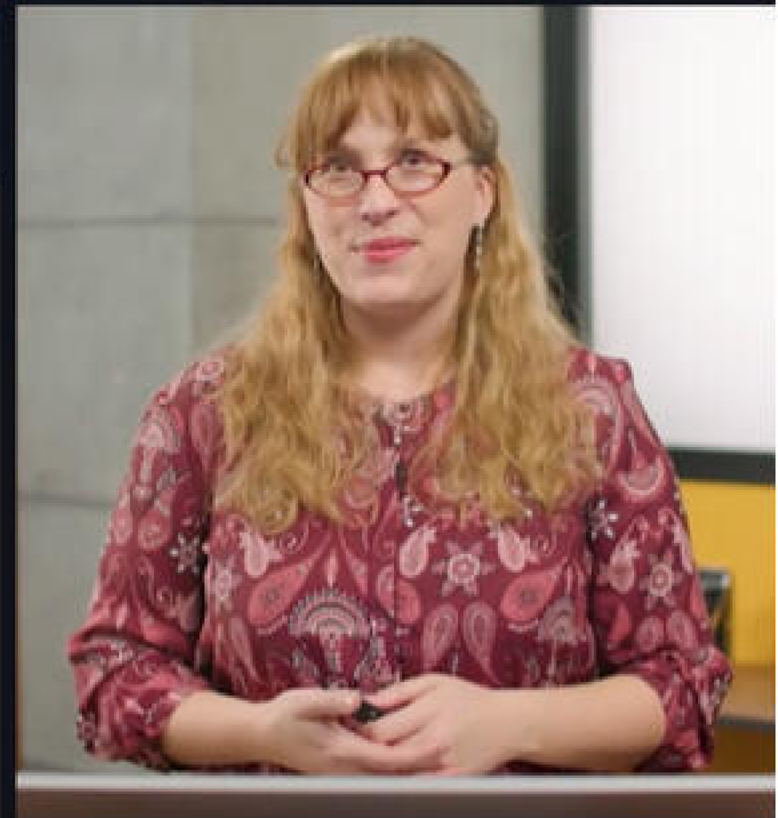
Amazon SageMaker Neo Benefits
aws training and certification
Optimizes a model for up to 2x performance speedup
Neo Compiler Container
Reduces the runtime footprint on the platform by 100x
© 2018 Amazon Web Services, Inc. or its Affiliates. All rights reserved.


Use Cases
aws training and certification
Amazon SageMaker Neo
Mobile Apps
Cloud
IOT
Amazon S3
Database
AWS Kinesis

Key Takeaways

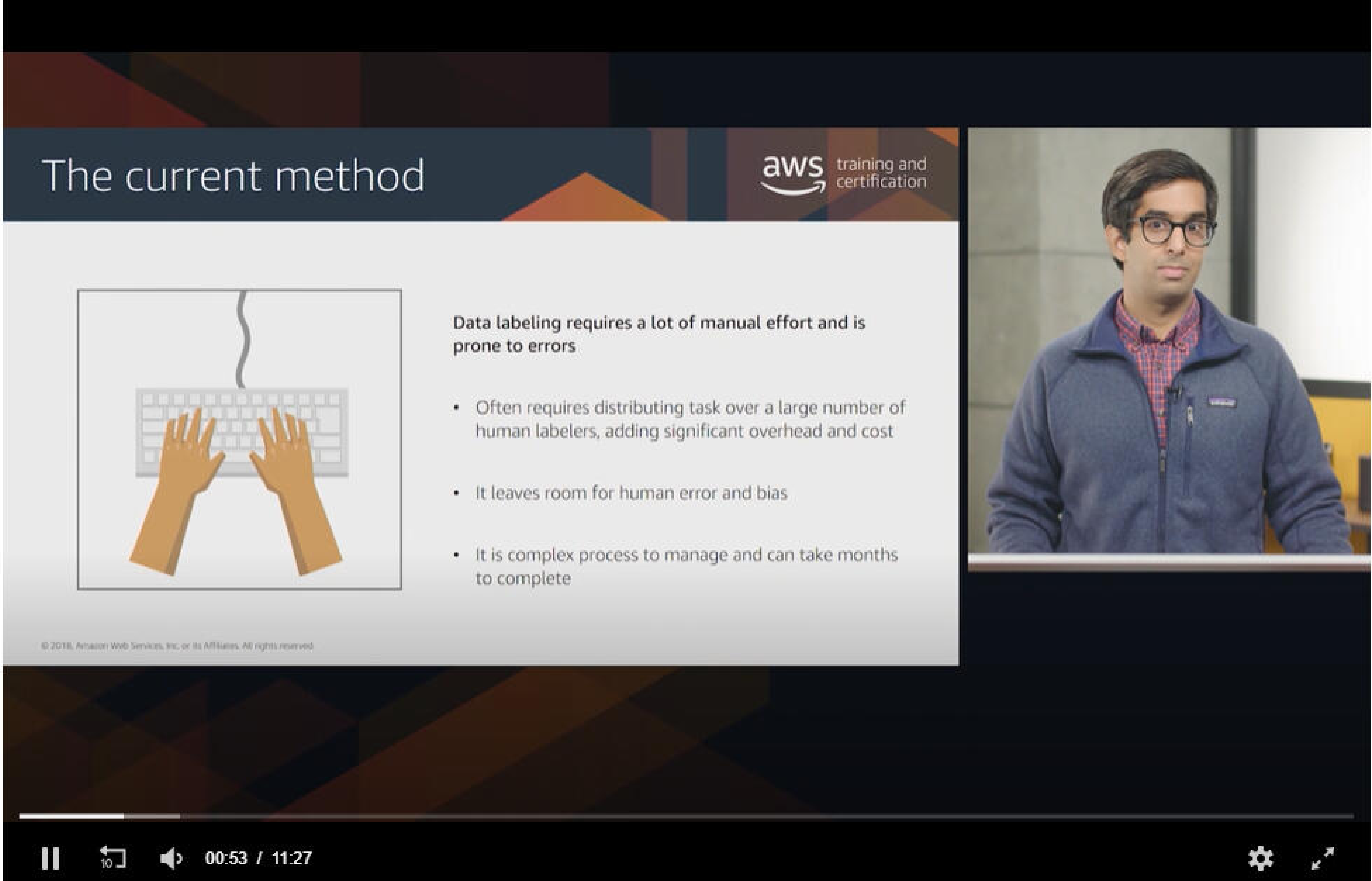
aws training and certification

- Popular deep learning and decision tree models.
- Apache MXNet, TensorFlow, PyTorch, XGBoost.
- Various Amazon EC2 instances and edge devices.
- Up to 2x performance speedup and 100x memory footprint reduction at no additional charge.



What is Ground Truth?

00:17 / 11:27


The current method
aws training and certification
Data labeling requires a lot of manual effort and is prone to errors
• Often requires distributing task over a large number of human labelers, adding significant overhead and cost
• It leaves room for human error and bias
• It is complex process to manage and can take months to complete
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
00:53 / 11:27

Ground Truth fixes these pain points

aws training and certification

Amazon Simple Storage Service (S3)

Ground Truth Labeling Jobs

- Managed experience where customers can set up a labeling job in a just a few steps
- Templates for common labeling tasks
- Built-in features to improve labeling accuracy
- Selection from multiple labeling workforces

Human labeling

You have three options when selecting a workforce to perform your labeling

Public crowdsourced workforce

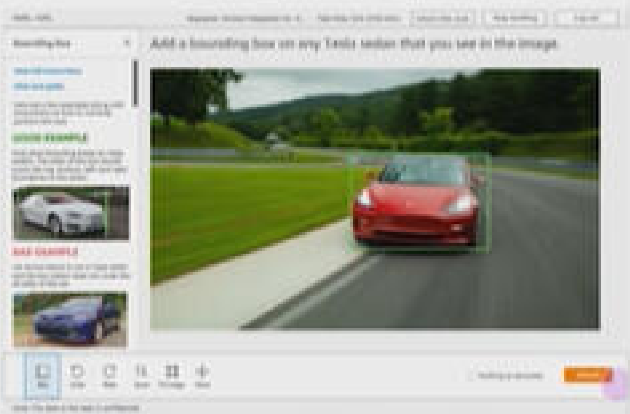
Pre-approved third-party vendors

Your own internal workforce

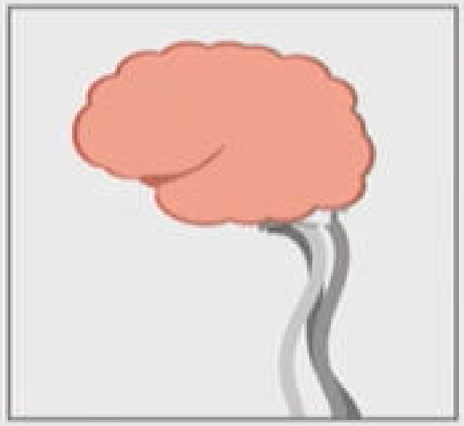
Better labeling accuracy

aws training and certification

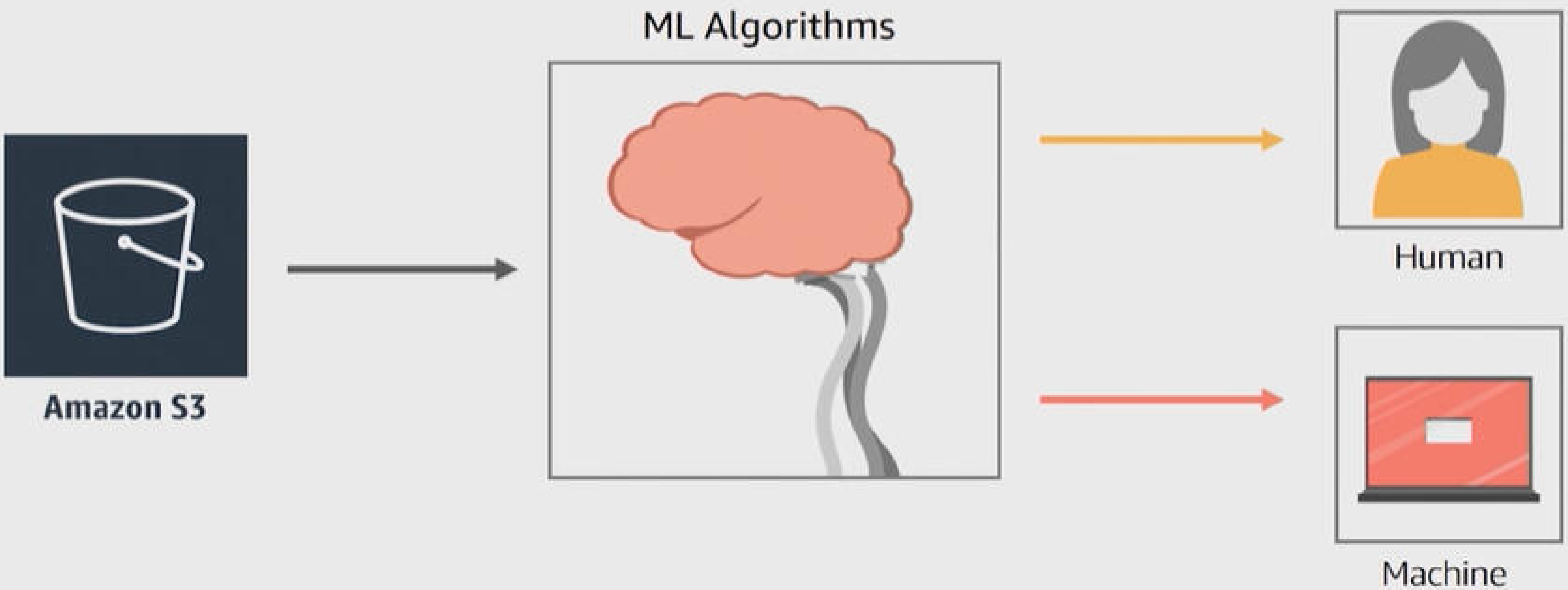
Innovative UX Techniques



Built-in Algorithms

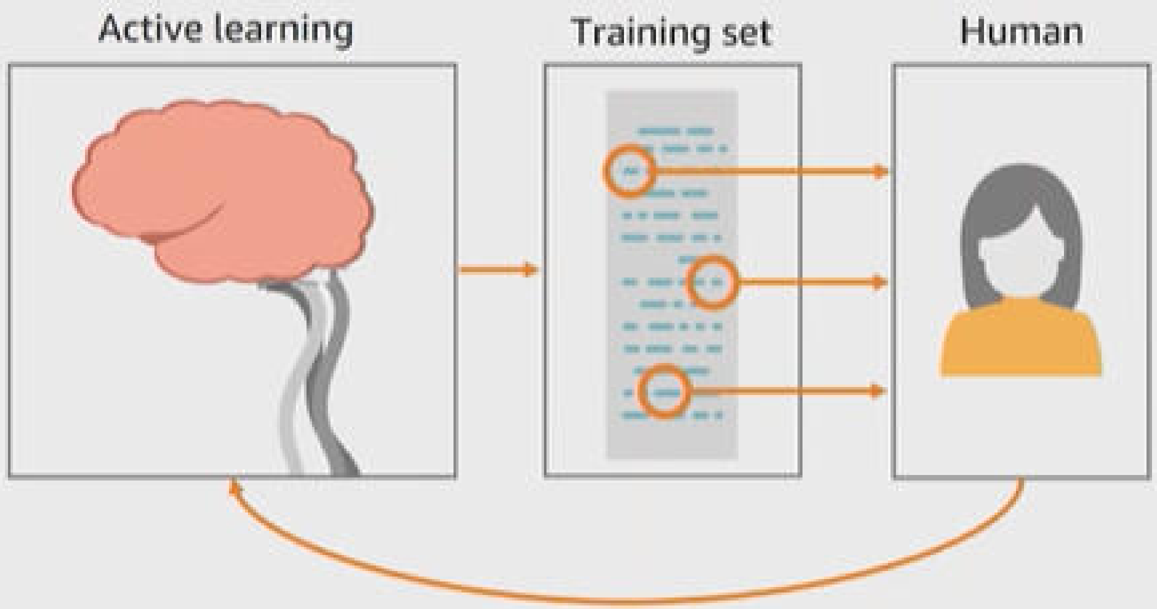


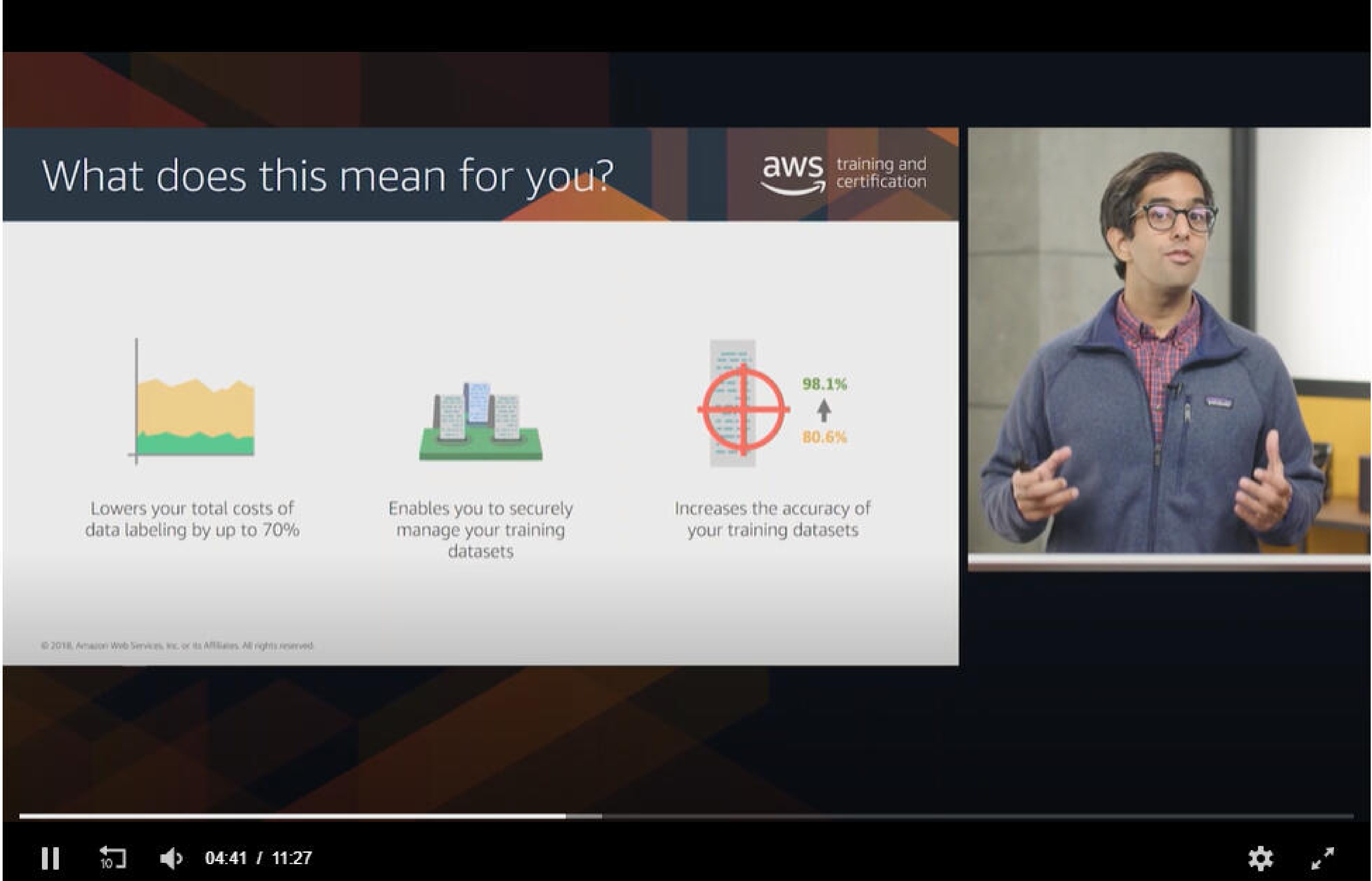
Automatic labeling

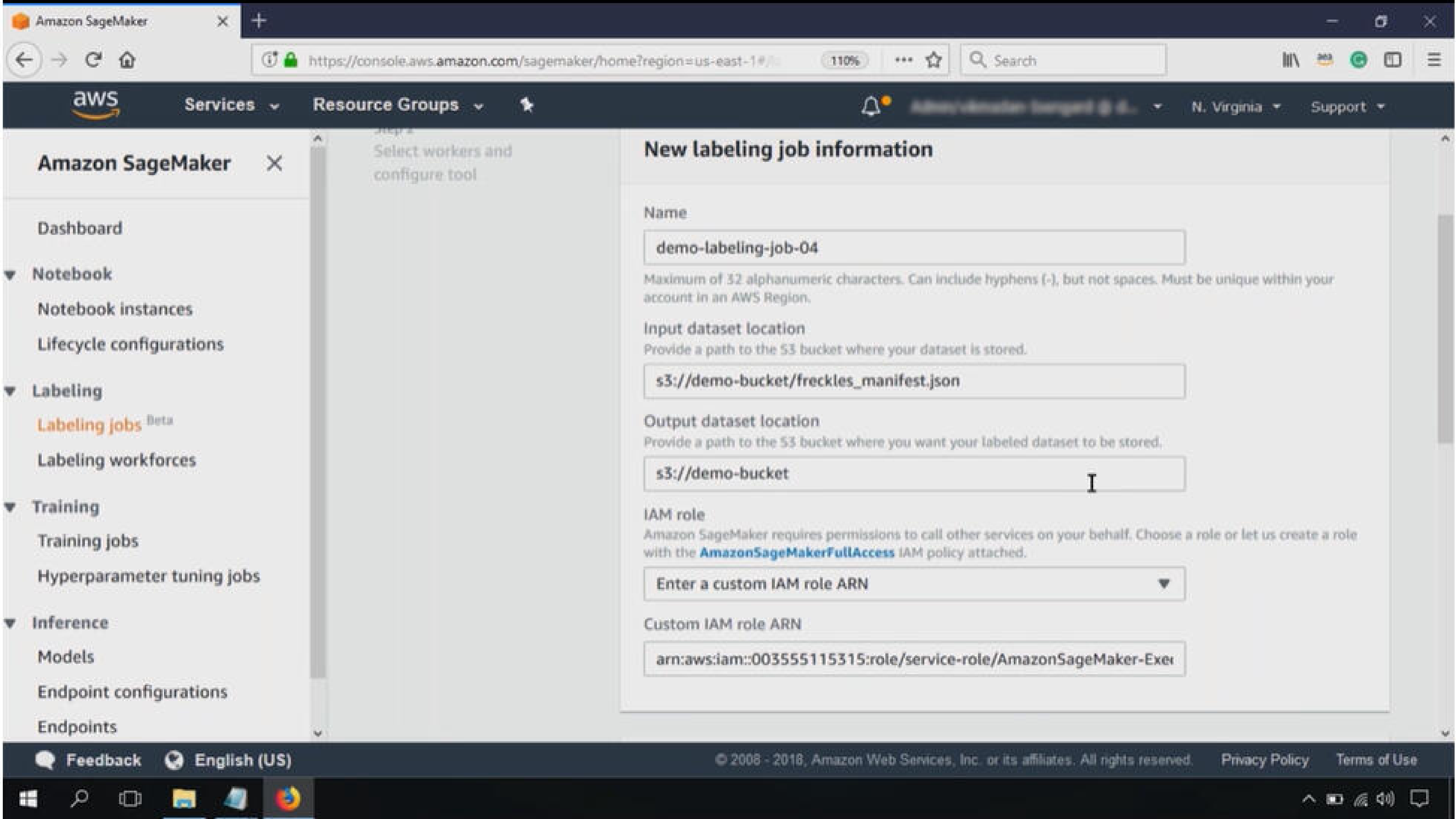


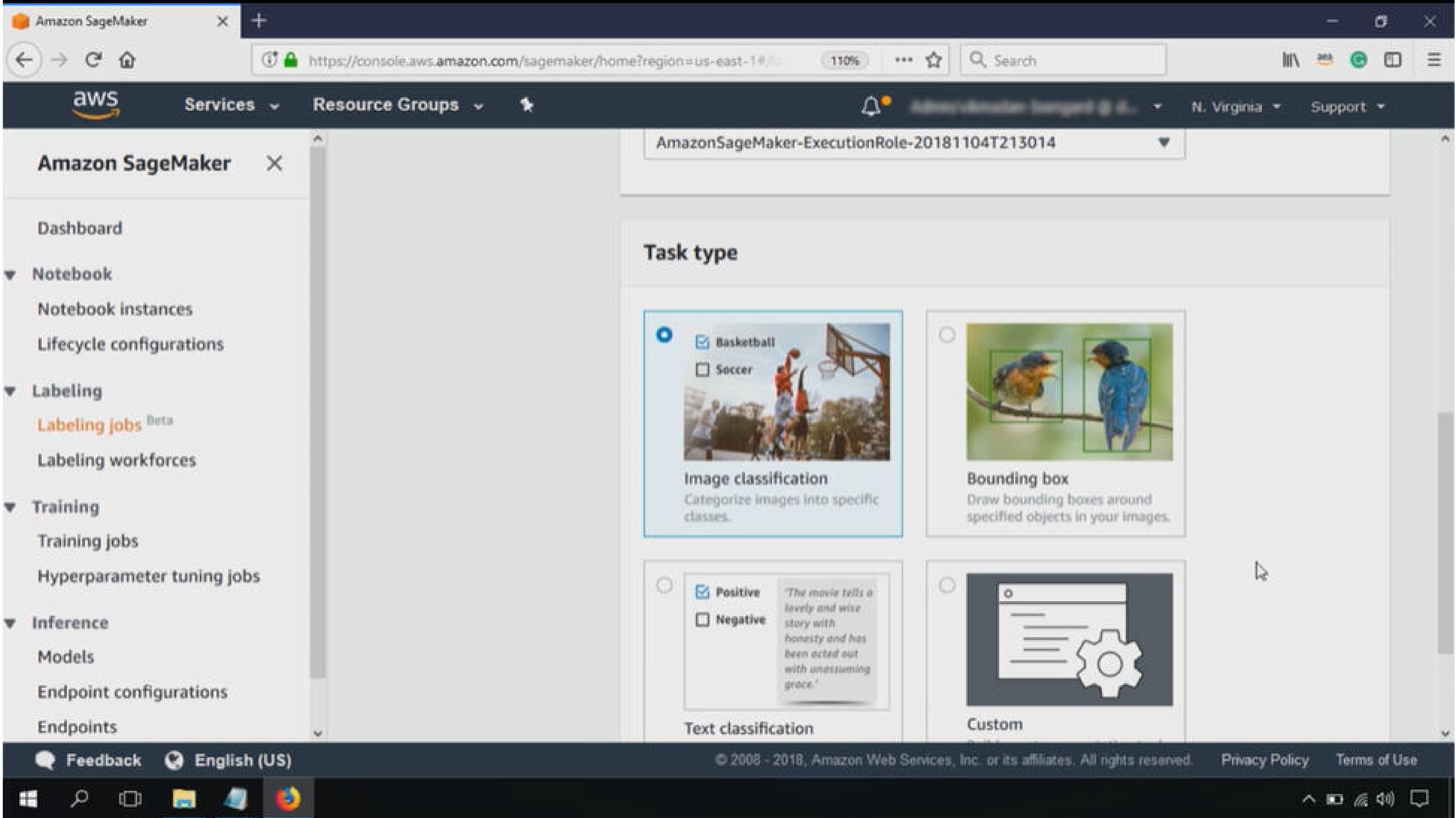
How automatic labeling works

aws training and certification

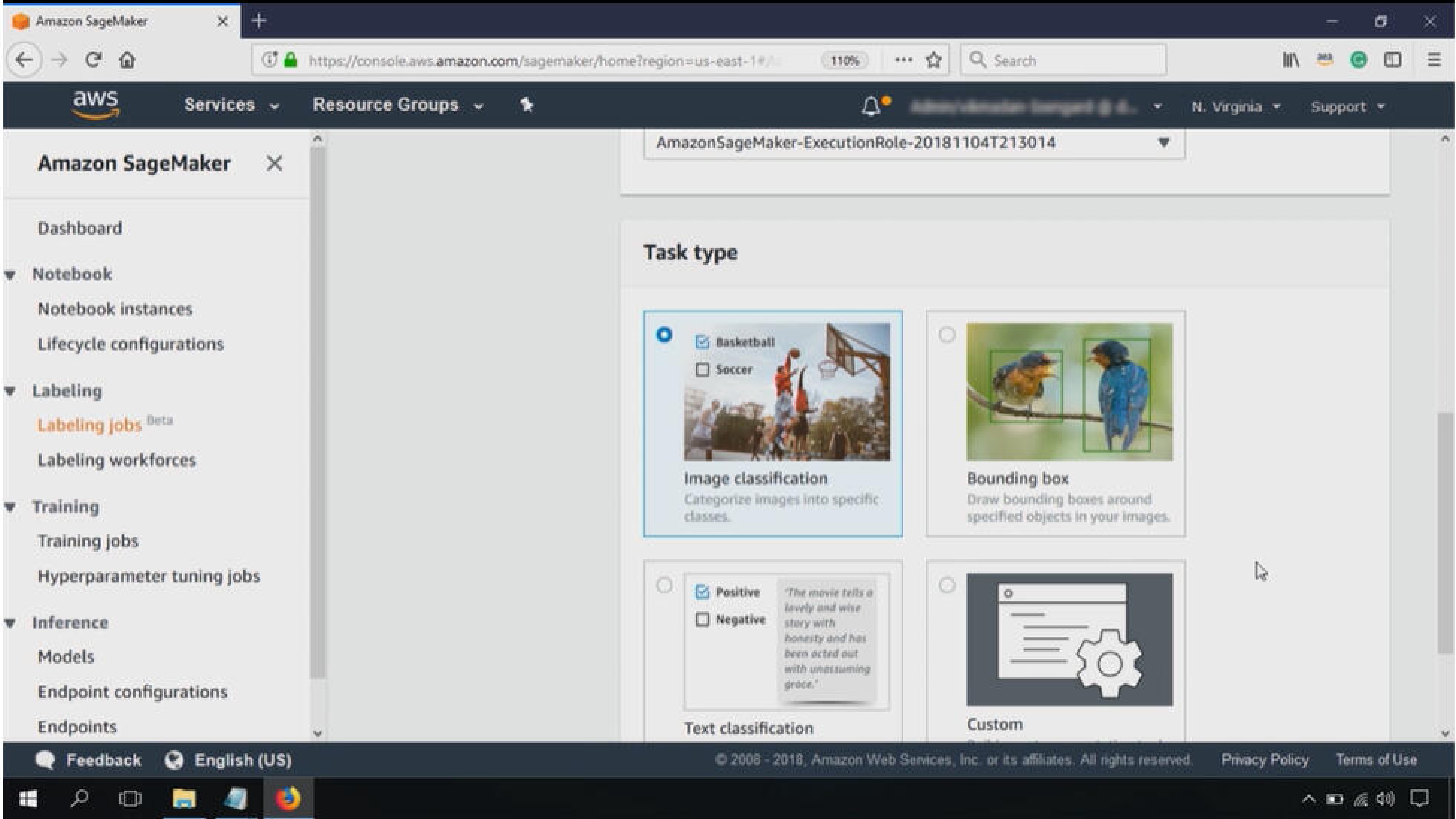


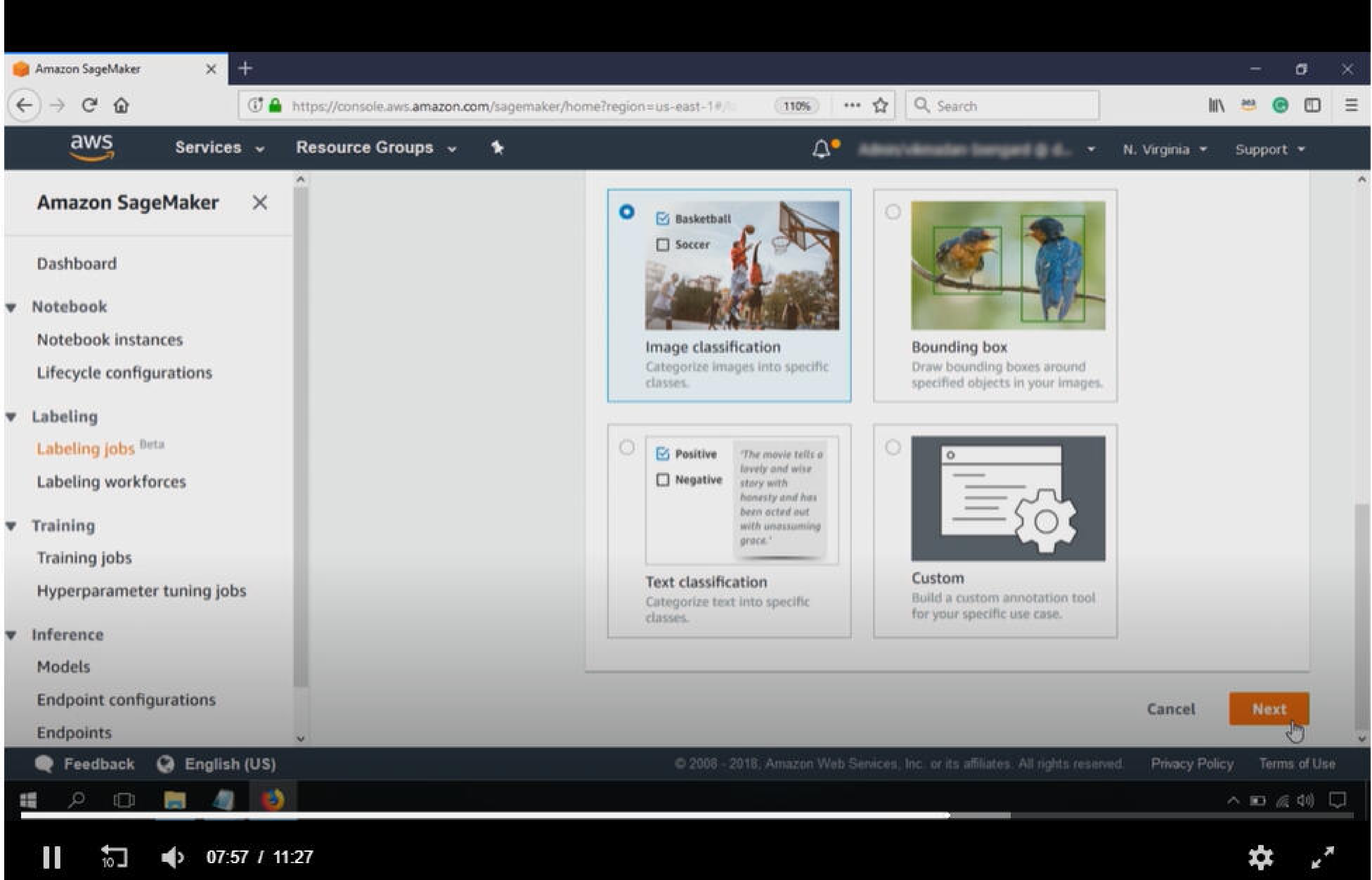

What does this mean for you?
aws training and certification
98.1%
80.6%
Lowers your total costs of data labeling by up to 70%
Enables you to securely manage your training datasets
Increases the accuracy of your training datasets
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
04:41 / 11:27


Amazon SageMaker
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
Services
Resource Groups
N. Virginia
Support
Select workers and configure tool
New labeling job information
Name
demo-labeling-job-04
Maximum of 32 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
Input dataset location
Provide a path to the S3 bucket where your dataset is stored.
s3://demo-bucket/freckles_manifest.json
Output dataset location
Provide a path to the S3 bucket where you want your labeled dataset to be stored.
s3://demo-bucket
IAM role
Amazon SageMaker requires permissions to call other services on your behalf. Choose a role or let us create a role with the AmazonSageMakerFullAccess IAM policy attached.
Enter a custom IAM role ARN
Custom IAM role ARN
arn:aws:iam::003555115315:role/service-role/AmazonSageMaker-Exe
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

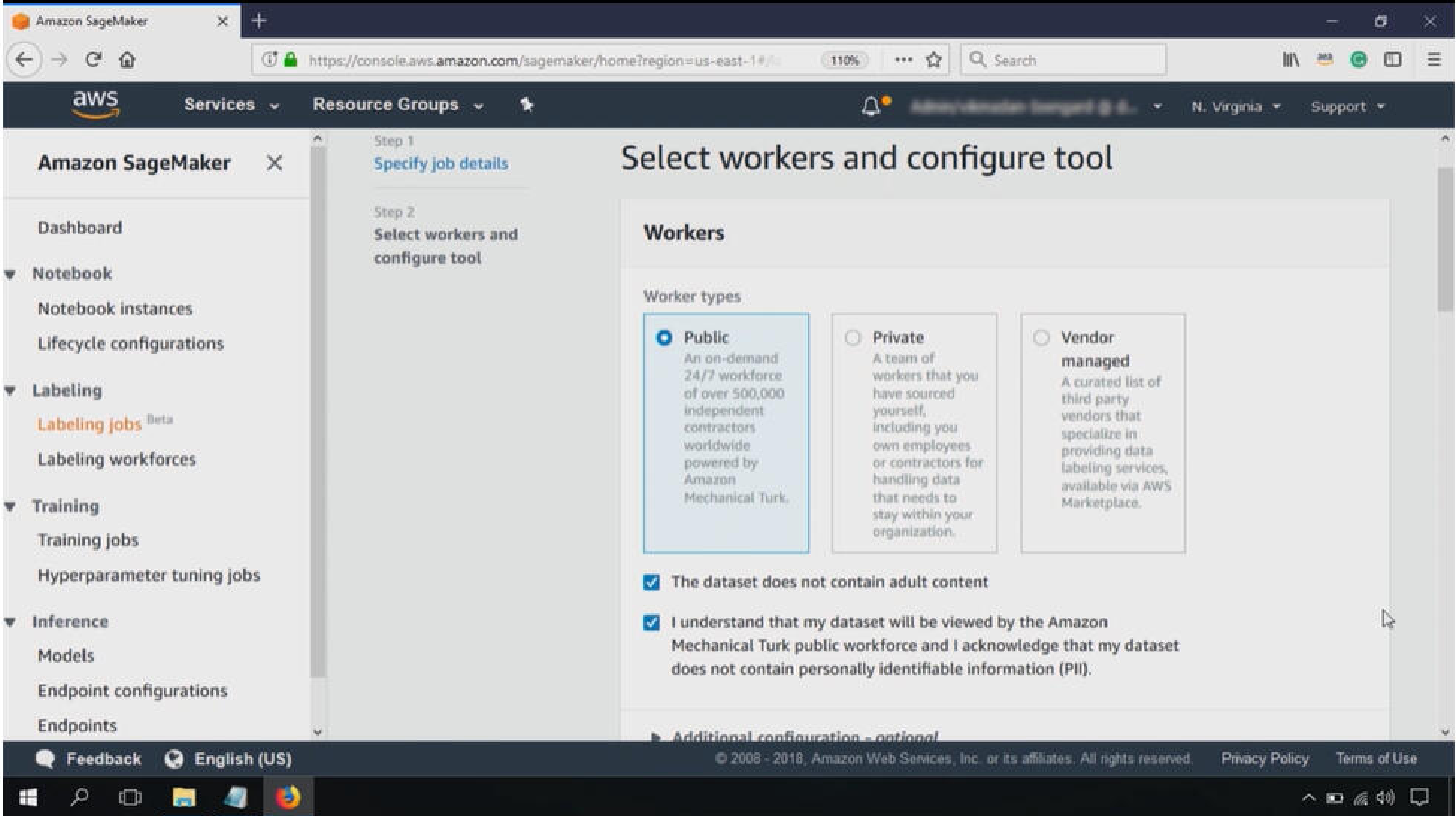


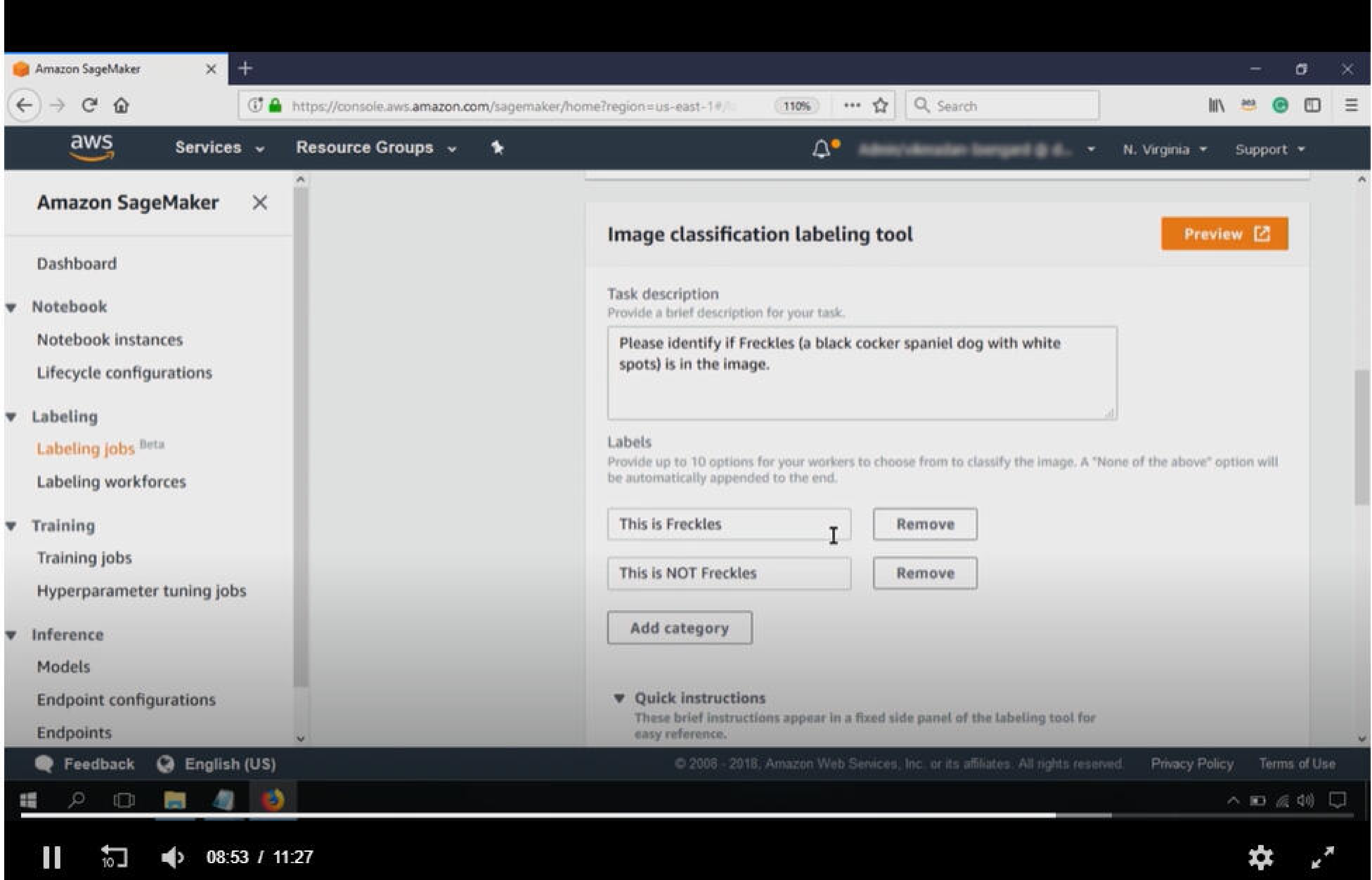
Amazon SageMaker
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
AmazonSageMaker-ExecutionRole-20181104T213014
Task type
Basketball
Soccer
Image classification
Categorize images into specific classes.
Bounding box
Draw bounding boxes around specified objects in your images.
Positive
Negative
"The movie tells a lovely and wise story with honesty and has been acted out with unassuming grace."
Text classification
Custom
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

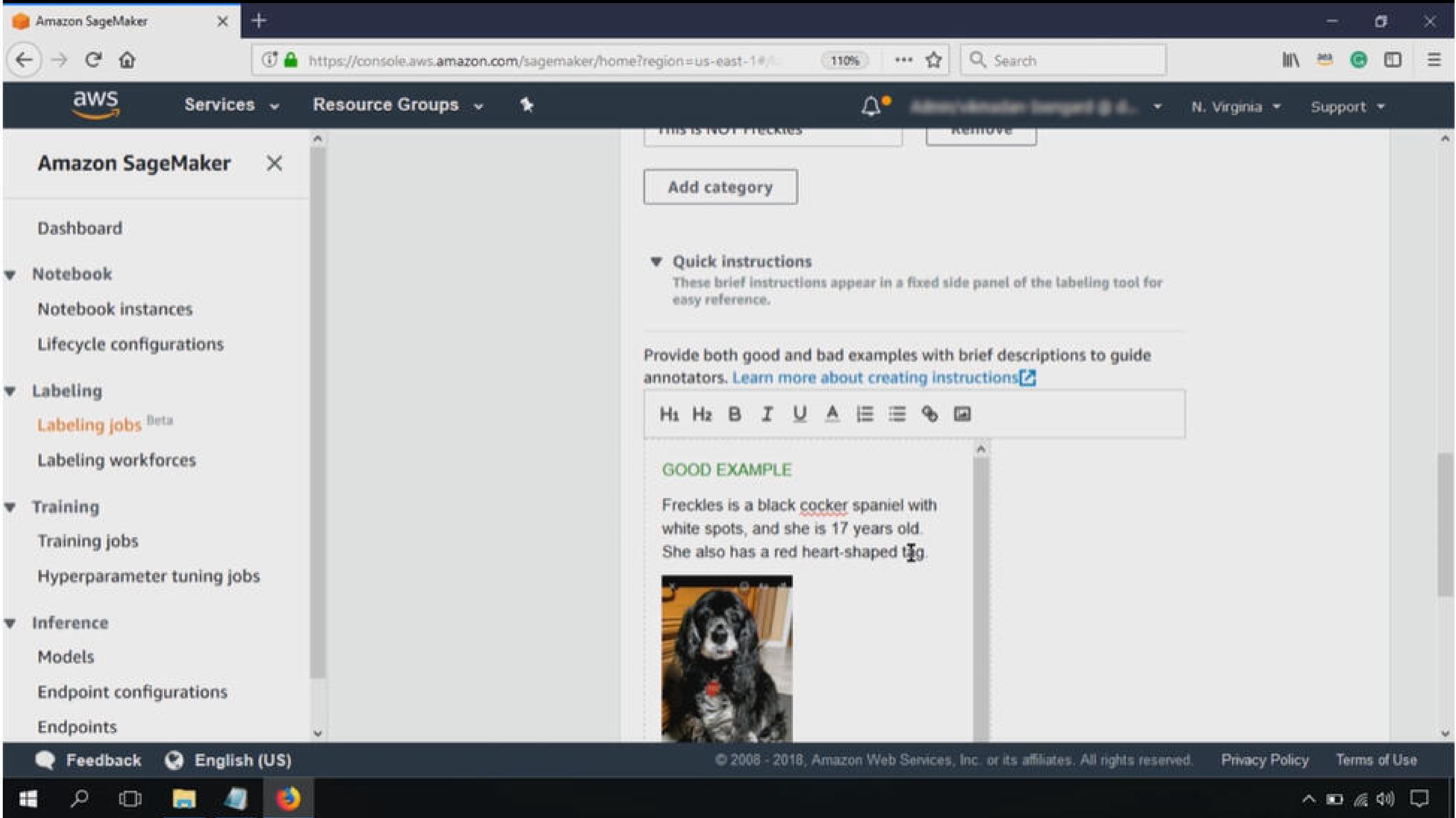

Amazon SageMaker
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
AmazonSageMaker-ExecutionRole-20181104T213014
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
Task type
Basketball
Soccer
Image classification
Categorize images into specific classes.
Bounding box
Draw bounding boxes around specified objects in your images.
Positive
Negative
"The movie tells a lovely and wise story with honesty and has been acted out with unassuming grace."
Text classification
Custom
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Amazon SageMaker
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
Basketball
Soccer
Image classification
Categorize images into specific classes.
Bounding box
Draw bounding boxes around specified objects in your images.
Positive
Negative
"The movie tells a lovely and wise story with honesty and has been acted out with unassuming grace."
Text classification
Categorize text into specific classes.
Custom
Build a custom annotation tool for your specific use case.
Cancel
Next
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
07:57 / 11:27


Amazon SageMaker
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
Step 1
Specify job details
Step 2
Select workers and configure tool
Select workers and configure tool
Workers
Worker types
Public
An on-demand 24/7 workforce of over 500,000 independent contractors worldwide powered by Amazon Mechanical Turk.
Private
A team of workers that you have sourced yourself, including you own employees or contractors for handling data that needs to stay within your organization.
Vendor managed
A curated list of third party vendors that specialize in providing data labeling services, available via AWS Marketplace.
The dataset does not contain adult content
I understand that my dataset will be viewed by the Amazon Mechanical Turk public workforce and I acknowledge that my dataset does not contain personally identifiable information (PII).
Additional configuration - optional
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Amazon SageMaker
https://console.aws.amazon.com/sagemaker/home?region=us-east-1
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
Image classification labeling tool
Preview
Task description
Provide a brief description for your task.
Please identify if Freckles (a black cocker spaniel dog with white spots) is in the image.
Labels
Provide up to 10 options for your workers to choose from to classify the image. A "None of the above" option will be automatically appended to the end.
This is Freckles
Remove
This is NOT Freckles
Remove
Add category
Quick instructions
These brief instructions appear in a fixed side panel of the labeling tool for easy reference.
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
08:53 / 11:27


Amazon SageMaker
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
Add category
Quick instructions
These brief instructions appear in a fixed side panel of the labeling tool for easy reference.
Provide both good and bad examples with brief descriptions to guide annotators. Learn more about creating instructions
H1 H2 B I U
GOOD EXAMPLE
Freckles is a black cocker spaniel with white spots, and she is 17 years old. She also has a red heart-shaped tag.
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Amazon SageMaker

https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/ 110% Search

aws Services Resource Groups N. Virginia Support

Amazon SageMaker

- Dashboard
- Notebook
  - Notebook instances
  - Lifecycle configurations
- Labeling
  - Labeling jobs Beta
  - Labeling workforces
- Training
  - Training jobs
  - Hyperparameter tuning jobs
- Inference
  - Models
  - Endpoint configurations
  - Endpoints

Quick instructions

These brief instructions appear in a fixed side panel of the labeling tool for easy reference.

Provide both good and bad examples with brief descriptions to guide annotators. Learn more about creating instructions

H1 H2 B I U A

GOOD EXAMPLE

Freckles is a black cocker spaniel with white spots, and she is 17 years old. She also has a red heart-shaped tag.

BAD EXAMPLE

Please make sure to mark dogs that are

Amazon SageMaker

https://console.aws.amazon.com/sagemaker/home?region=us-east-1

Services | Resource Groups | N. Virginia | Support

Amazon SageMaker

Dashboard

Notebook
- Notebook instances
- Lifecycle configurations

Labeling
- Labeling jobs Beta
- Labeling workforces

Training
- Training jobs
- Hyperparameter tuning jobs

Inference
- Models
- Endpoint configurations
- Endpoints

Quick instructions

These brief instructions appear in a fixed side panel of the labeling tool for easy reference.

Provide both good and bad examples with brief descriptions to guide annotators. Learn more about creating instructions

GOOD EXAMPLE

Freckles is a black cocker spaniel with white spots, and she is 17 years old. She also has a red heart-shaped tag.

BAD EXAMPLE

Feedback | English (US)

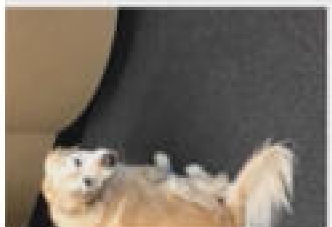
1. **Read** the task carefully and inspect the image.
2. **Read** the options and review the examples provided to understand more about the labels.
3. **Choose the** appropriate label that best suits the image.

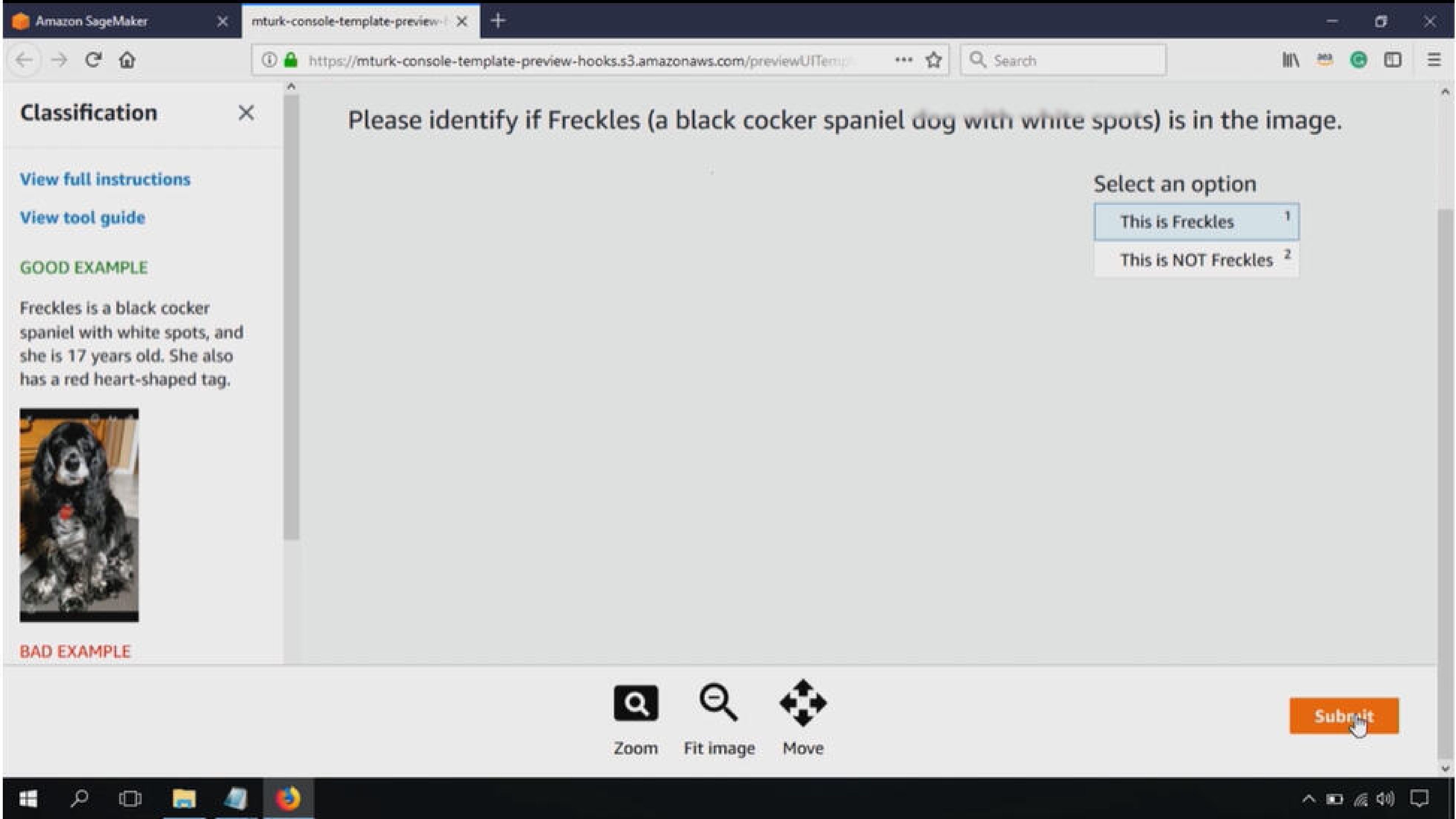
GOOD EXAMPLE

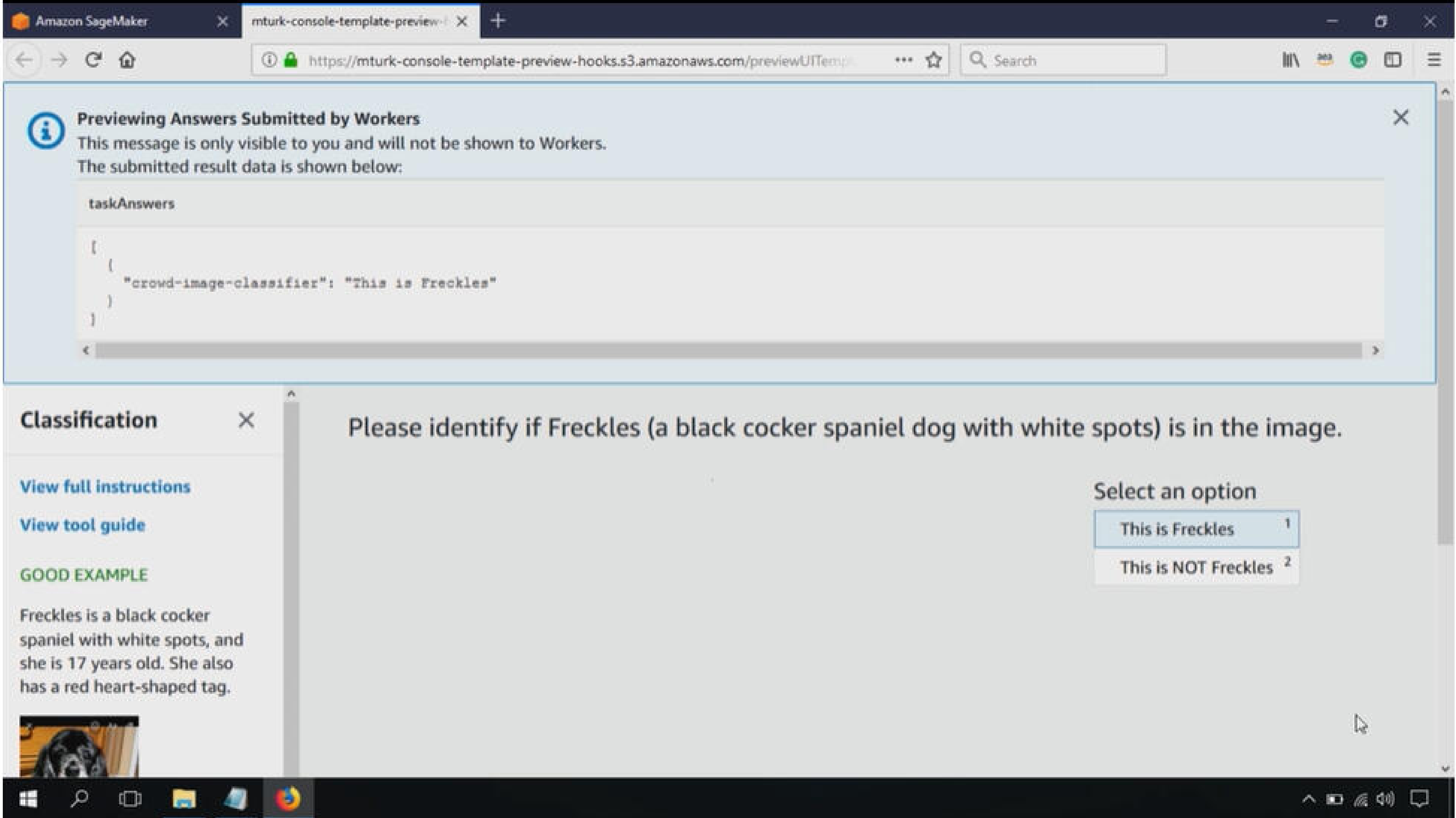
Freckles is a black cocker spaniel with white spots, and she is 17 years old. She also has a red heart-shaped tag.

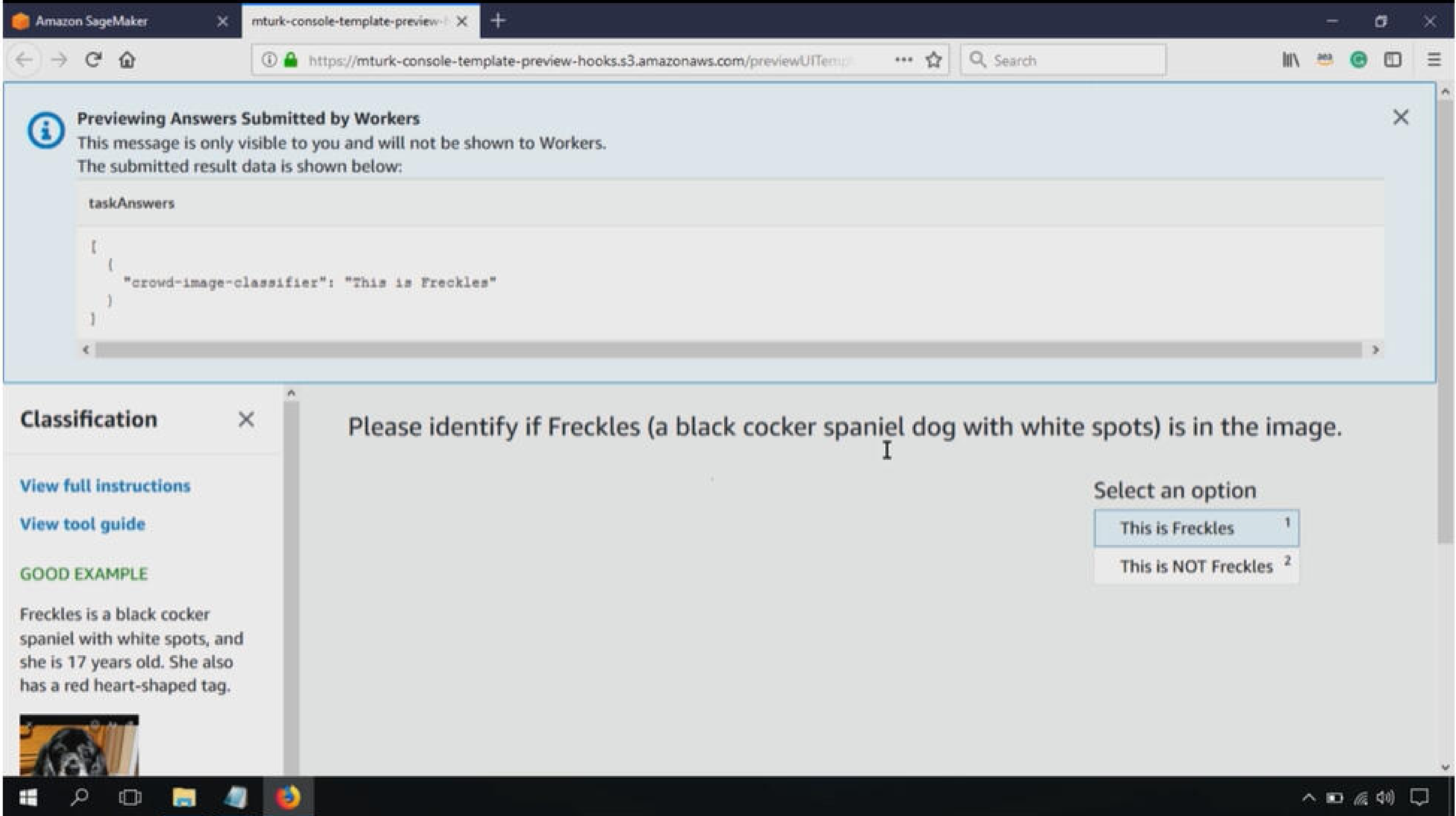
BAD EXAMPLE

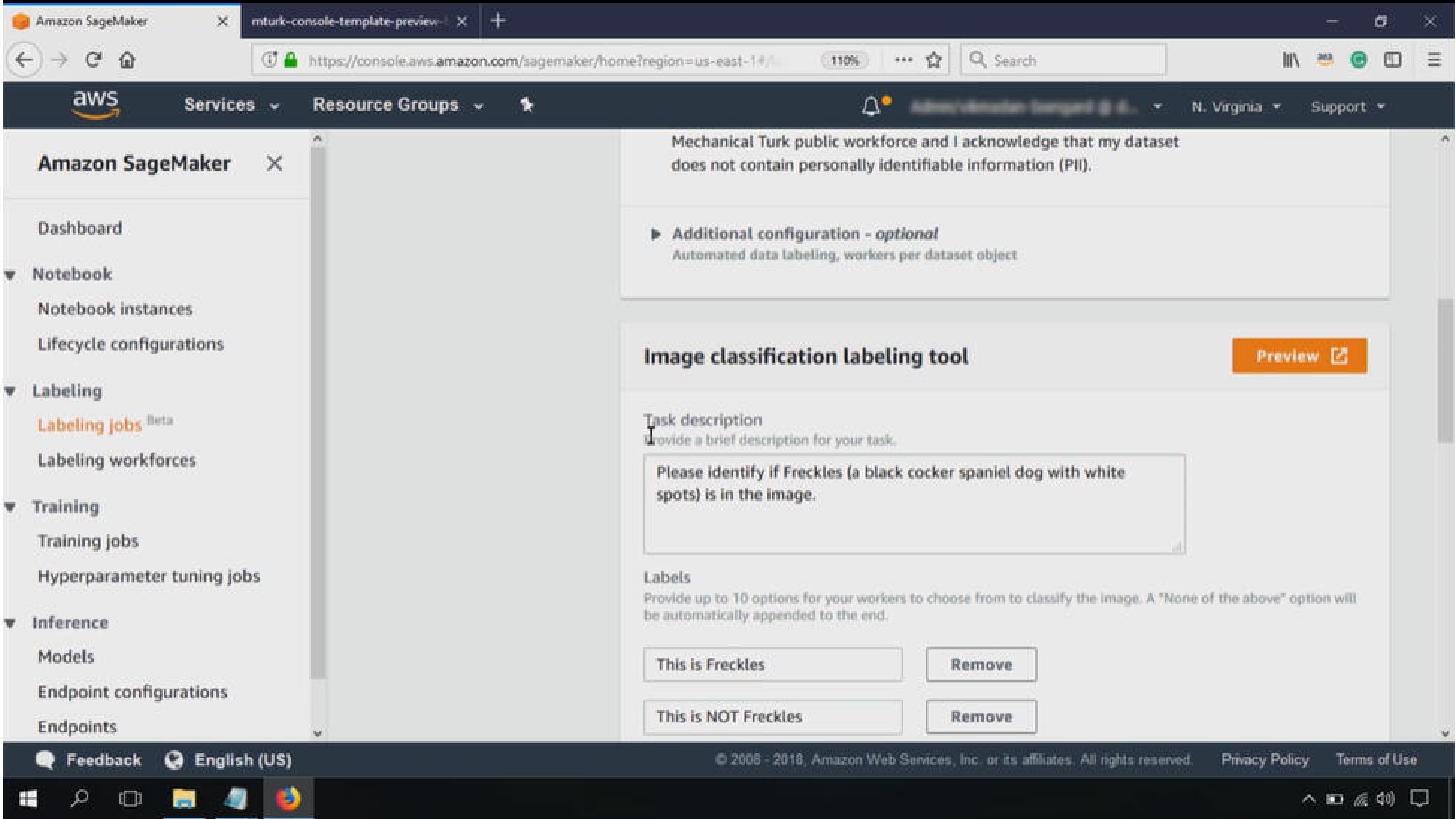
Please make sure to mark dogs that are not Freckles as "This is NOT Freckles".

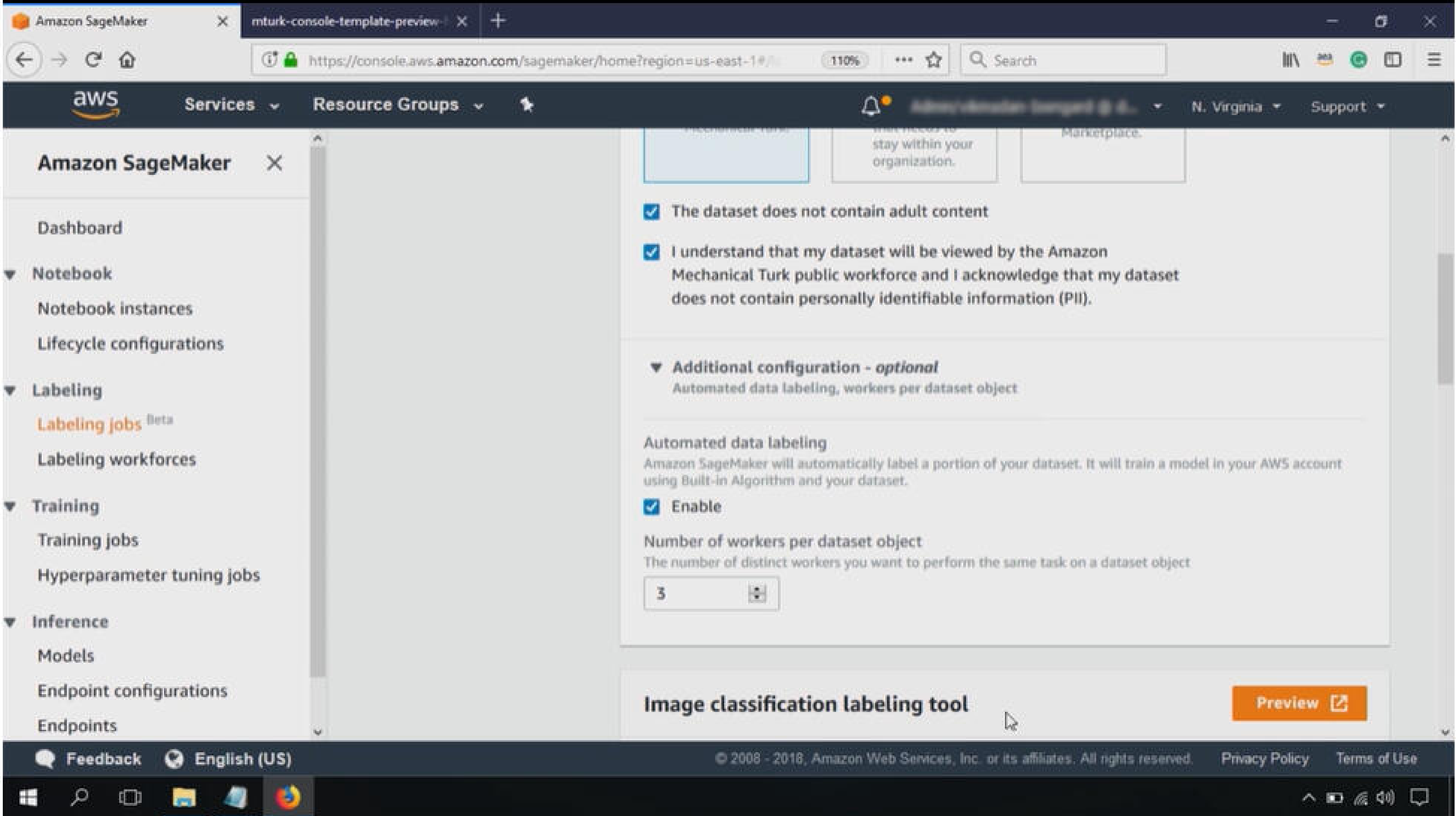



Amazon SageMaker
mturk-console-template-preview-
https://mturk-console-template-preview-hooks.s3.amazonaws.com/previewUITem
Search
Classification
View full instructions
View tool guide
GOOD EXAMPLE
Freckles is a black cocker spaniel with white spots, and she is 17 years old. She also has a red heart-shaped tag.
BAD EXAMPLE
Please identify if Freckles (a black cocker spaniel dog with white spots) is in the image.
Select an option
This is Freckles 1
This is NOT Freckles 2
Zoom
Fit image
Move
Submit

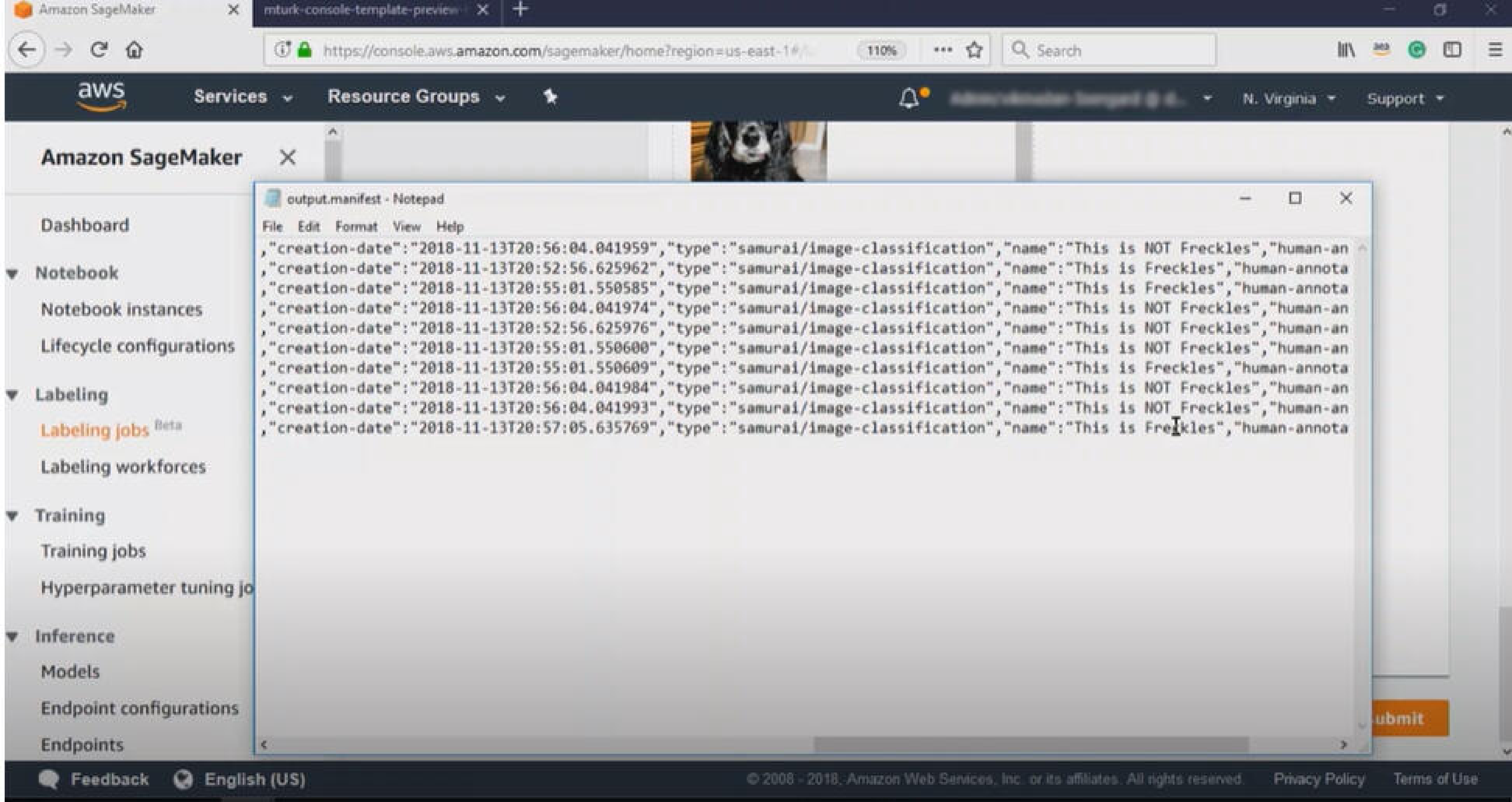

Amazon SageMaker
mturk-console-template-preview-
https://mturk-console-template-preview-hooks.s3.amazonaws.com/previewUITem
Search
Previewing Answers Submitted by Workers
This message is only visible to you and will not be shown to Workers.
The submitted result data is shown below:
taskAnswers
[
{
"crowd-image-classifier": "This is Freckles"
}
]
Classification
View full instructions
View tool guide
GOOD EXAMPLE
Freckles is a black cocker spaniel with white spots, and she is 17 years old. She also has a red heart-shaped tag.
Please identify if Freckles (a black cocker spaniel dog with white spots) is in the image.
Select an option
This is Freckles 1
This is NOT Freckles 2


Amazon SageMaker
mturk-console-template-preview-
https://mturk-console-template-preview-hooks.s3.amazonaws.com/previewUITem
Search
Previewing Answers Submitted by Workers
This message is only visible to you and will not be shown to Workers.
The submitted result data is shown below:
taskAnswers
[
{
"crowd-image-classifier": "This is Freckles"
}
]
Classification
View full instructions
View tool guide
GOOD EXAMPLE
Freckles is a black cocker spaniel with white spots, and she is 17 years old. She also has a red heart-shaped tag.
Please identify if Freckles (a black cocker spaniel dog with white spots) is in the image.
Select an option
This is Freckles 1
This is NOT Freckles 2


Amazon SageMaker
mturk-console-template-preview-
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
Mechanical Turk public workforce and I acknowledge that my dataset does not contain personally identifiable information (PII).
Additional configuration - optional
Automated data labeling, workers per dataset object
Image classification labeling tool
Preview
Task description
Provide a brief description for your task.
Please identify if Freckles (a black cocker spaniel dog with white spots) is in the image.
Labels
Provide up to 10 options for your workers to choose from to classify the image. A "None of the above" option will be automatically appended to the end.
This is Freckles
Remove
This is NOT Freckles
Remove
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Amazon SageMaker
mturk-console-template-preview-
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jobs
Inference
Models
Endpoint configurations
Endpoints
stay within your organization.
Marketplace.
The dataset does not contain adult content
I understand that my dataset will be viewed by the Amazon Mechanical Turk public workforce and I acknowledge that my dataset does not contain personally identifiable information (PII).
Additional configuration - optional
Automated data labeling, workers per dataset object
Automated data labeling
Amazon SageMaker will automatically label a portion of your dataset. It will train a model in your AWS account using Built-in Algorithm and your dataset.
Enable
Number of workers per dataset object
The number of distinct workers you want to perform the same task on a dataset object
3
Image classification labeling tool
Preview
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Amazon SageMaker
mturk-console-template-preview
https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/
110%
Search
aws
Services
Resource Groups
N. Virginia
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Labeling
Labeling jobs Beta
Labeling workforces
Training
Training jobs
Hyperparameter tuning jo
Inference
Models
Endpoint configurations
Endpoints
output.manifest - Notepad
File Edit Format View Help
,"creation-date":"2018-11-13T20:56:04.041959","type":"samurai/image-classification","name":"This is NOT Freckles","human-an
,"creation-date":"2018-11-13T20:52:56.625962","type":"samurai/image-classification","name":"This is Freckles","human-annota
,"creation-date":"2018-11-13T20:55:01.550585","type":"samurai/image-classification","name":"This is Freckles","human-annota
,"creation-date":"2018-11-13T20:56:04.041974","type":"samurai/image-classification","name":"This is NOT Freckles","human-an
,"creation-date":"2018-11-13T20:52:56.625976","type":"samurai/image-classification","name":"This is NOT Freckles","human-an
,"creation-date":"2018-11-13T20:55:01.550600","type":"samurai/image-classification","name":"This is NOT Freckles","human-an
,"creation-date":"2018-11-13T20:55:01.550609","type":"samurai/image-classification","name":"This is Freckles","human-annota
,"creation-date":"2018-11-13T20:56:04.041984","type":"samurai/image-classification","name":"This is NOT Freckles","human-an
,"creation-date":"2018-11-13T20:56:04.041993","type":"samurai/image-classification","name":"This is NOT Freckles","human-an
,"creation-date":"2018-11-13T20:57:05.635769","type":"samurai/image-classification","name":"This is Freckles","human-annota
ubmit
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Amazon Polly

Presented by Jorge Negron

aws training and certification

Introduction

- Text to Speech Service Introduction
- Amazon Polly Overview
- Amazon Polly Use Cases
- AWS Console Demonstration

Artificial Intelligence Services

aws training and certification

- Amazon Lex: Build Voice and Text Chatbots
- **Amazon Polly: Turn Text into Lifelike Speech**
- Amazon Rekognition: Search and Analyze Images
- Amazon Machine Learning: Build Smart Applications

Amazon Polly

- Create applications that talk and increase accessibility
- Provides an easy to use and device independent solution
- Voices are high quality and as good as natural human speech
- Supports 24 languages with multiple voices per language
- Store and distribute the generated speech as an audio file

Amazon Polly

- Create applications that talk and increase accessibility
- Provides an easy to use and device independent solution
- Voices are high quality and as good as natural human speech
- Supports 24 languages with multiple voices per language
- Store and distribute the generated speech as an audio file

Amazon Polly

aws training and certification

- Accurate text processing.
- Highly Intelligible.
- Supports Speech Synthesis Markup Language (SSML).
- Supports dictionaries (Lexicons) for custom pronunciation.
- Natural sounding speech generation from text.

Amazon Polly and SSML

- Speech Synthesis Markup Language (SSML)
- XML based markup language. Tags comply with SSML 1.1
- SSML start with **<speak>**, end with **</speak>**
- Modify aspects of the speech output:
  - Expansion of abbreviations and acronyms
  - Control of pitch, volume, and speed of speech

Amazon Polly and SSML

- Speech Synthesis Markup Language (SSML)
- XML based markup language. Tags comply with SSML 1.1
- SSML start with **<speak>**, end with **</speak>**
- Modify aspects of the speech output:
  - Expansion of abbreviations and acronyms
  - Control of pitch, volume, and speed of speech

Amazon Polly Use Cases

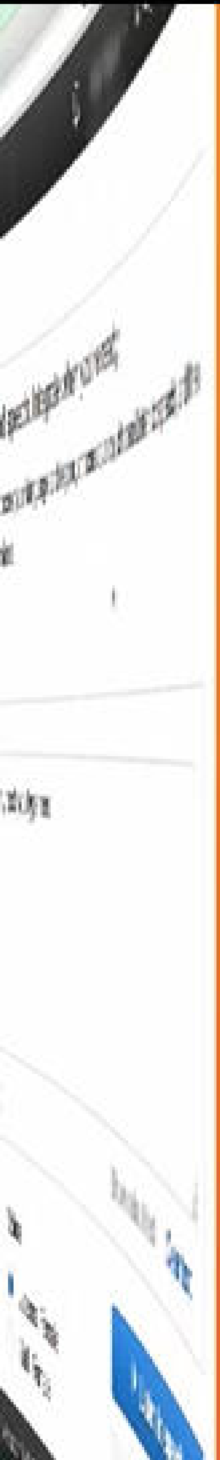
Amazon Polly Use Cases

- Education: Language learning applications (24 languages)
- Gaming: Test in-game dialogs without needing a voice actor
- Content Creation: Read news channel content aloud
- Telephony: Generated speech is used as Voice Response

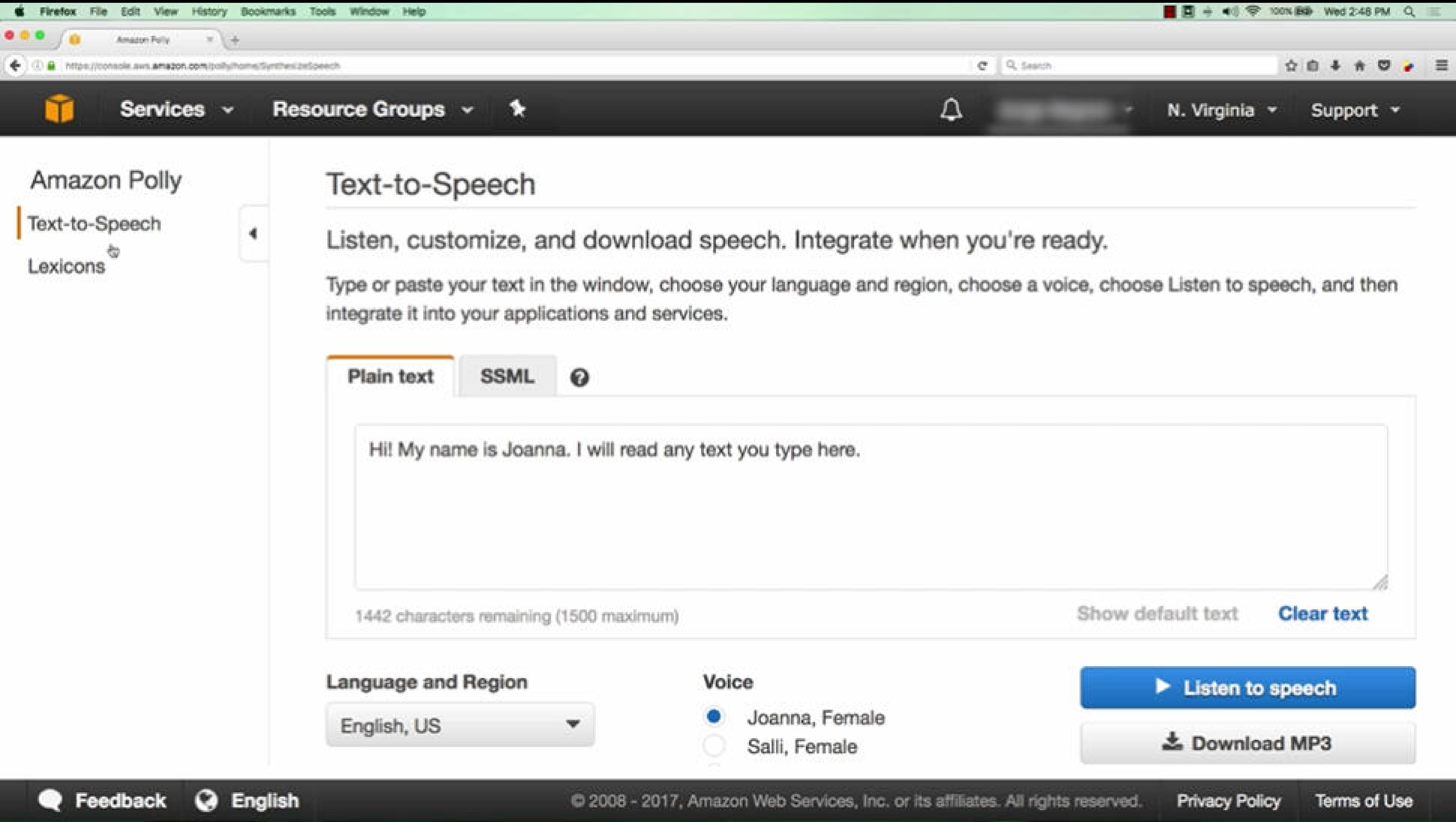
Amazon Polly Use Cases

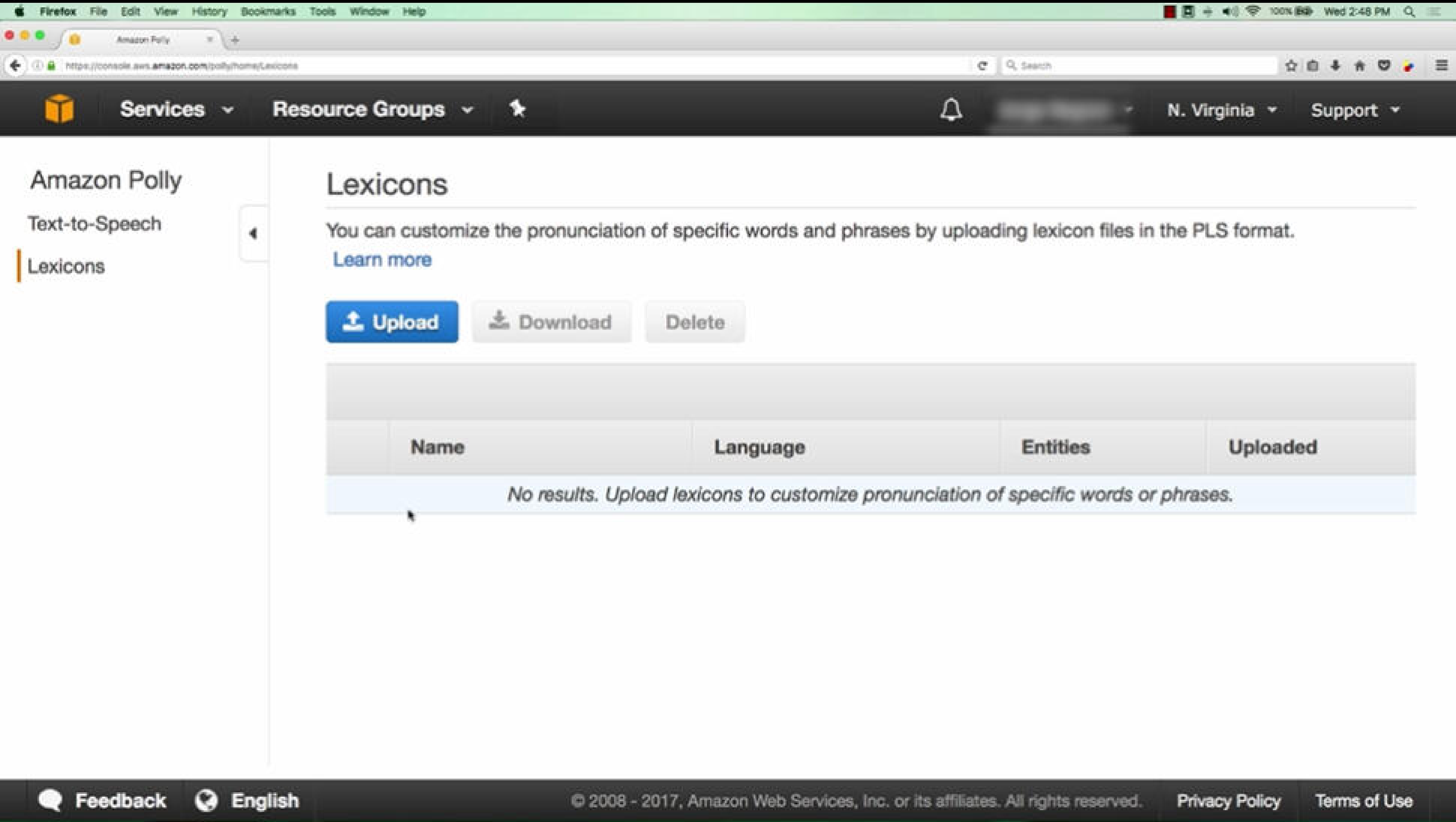
aws training and certification

- Education: Language learning applications (24 languages)
- Gaming: Test in-game dialogs without needing a voice actor
- Content Creation: Read news channel content aloud
- Telephony: Generated speech is used as Voice Response

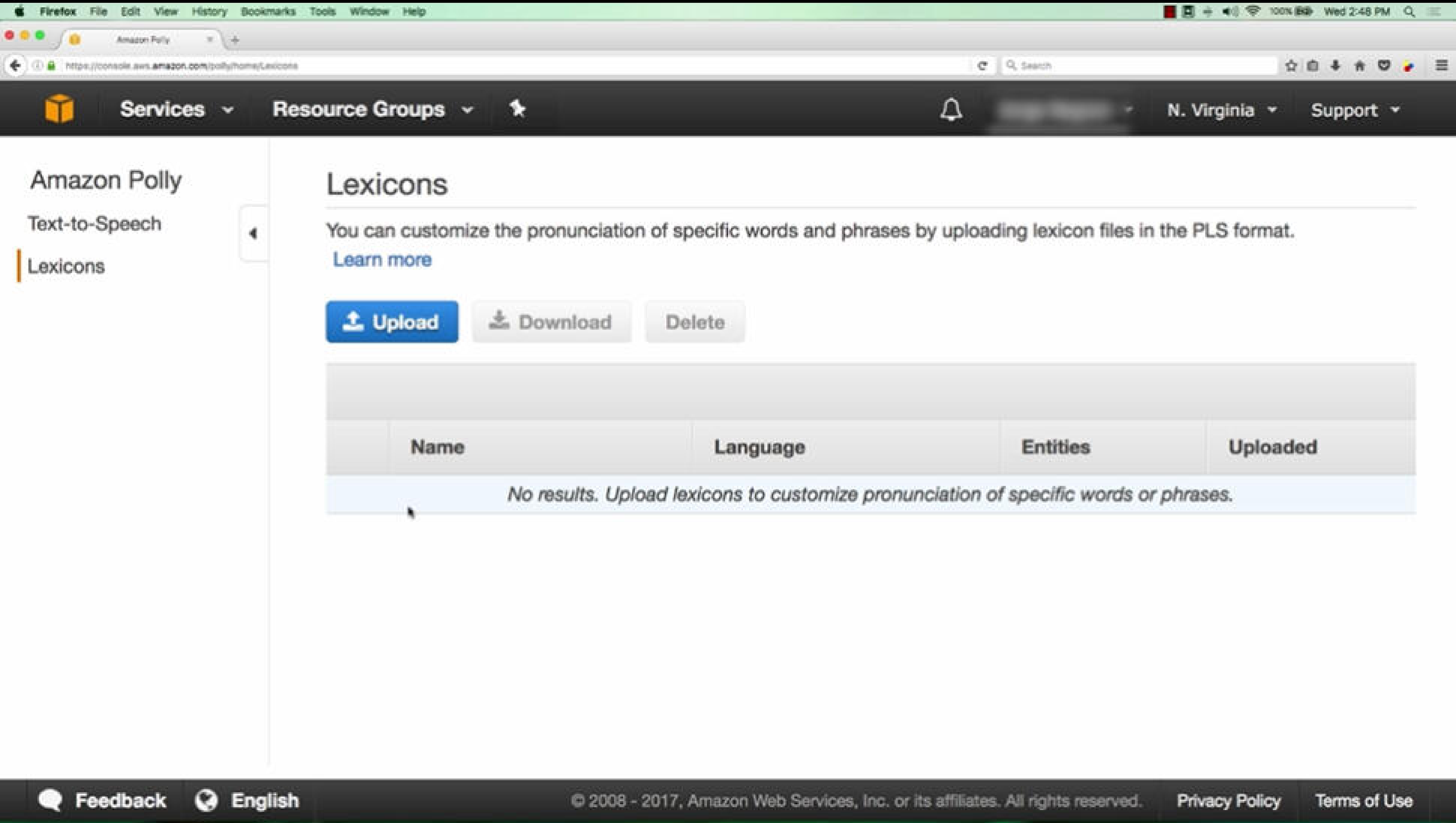


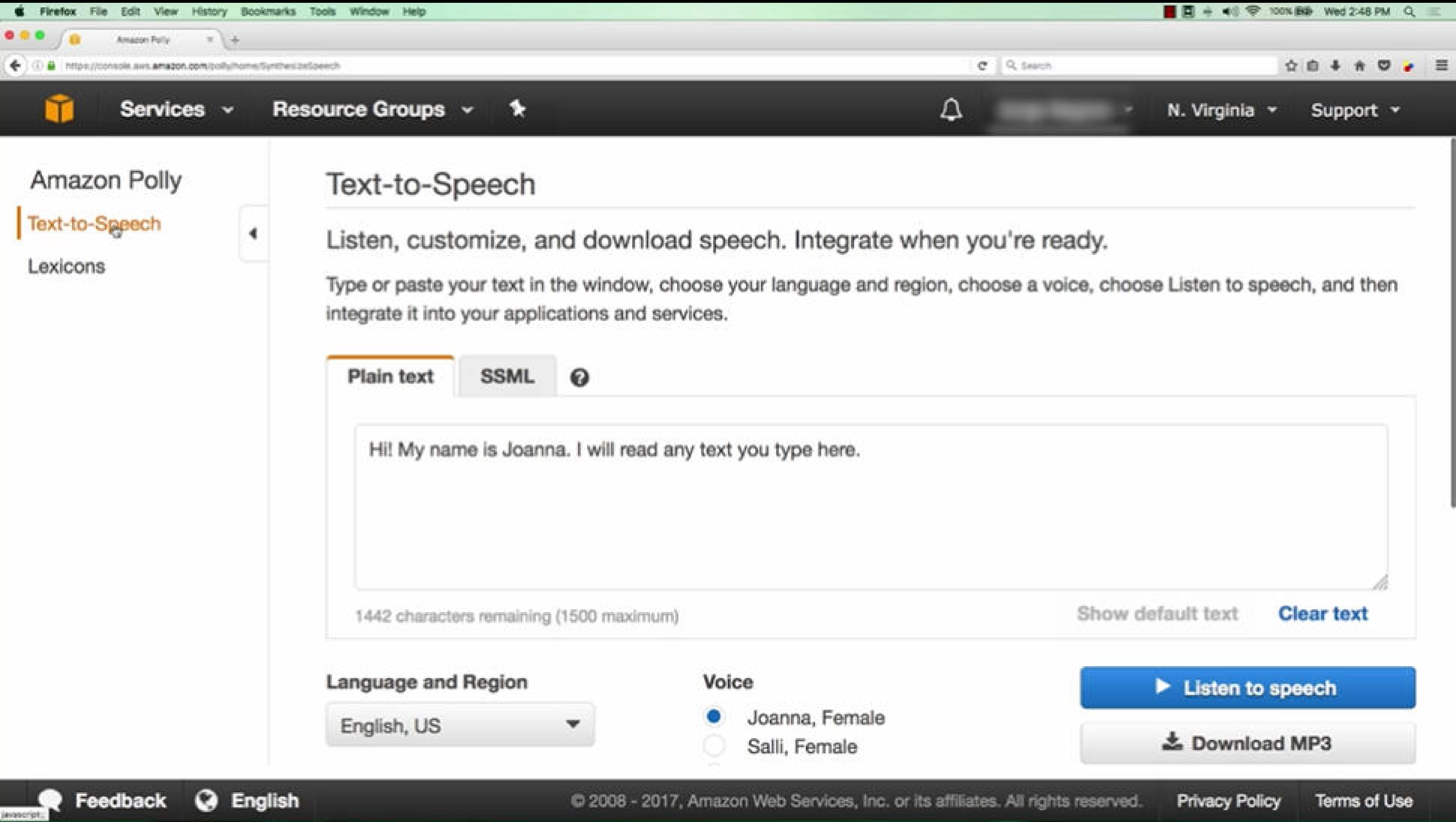
Amazon Polly Demonstration

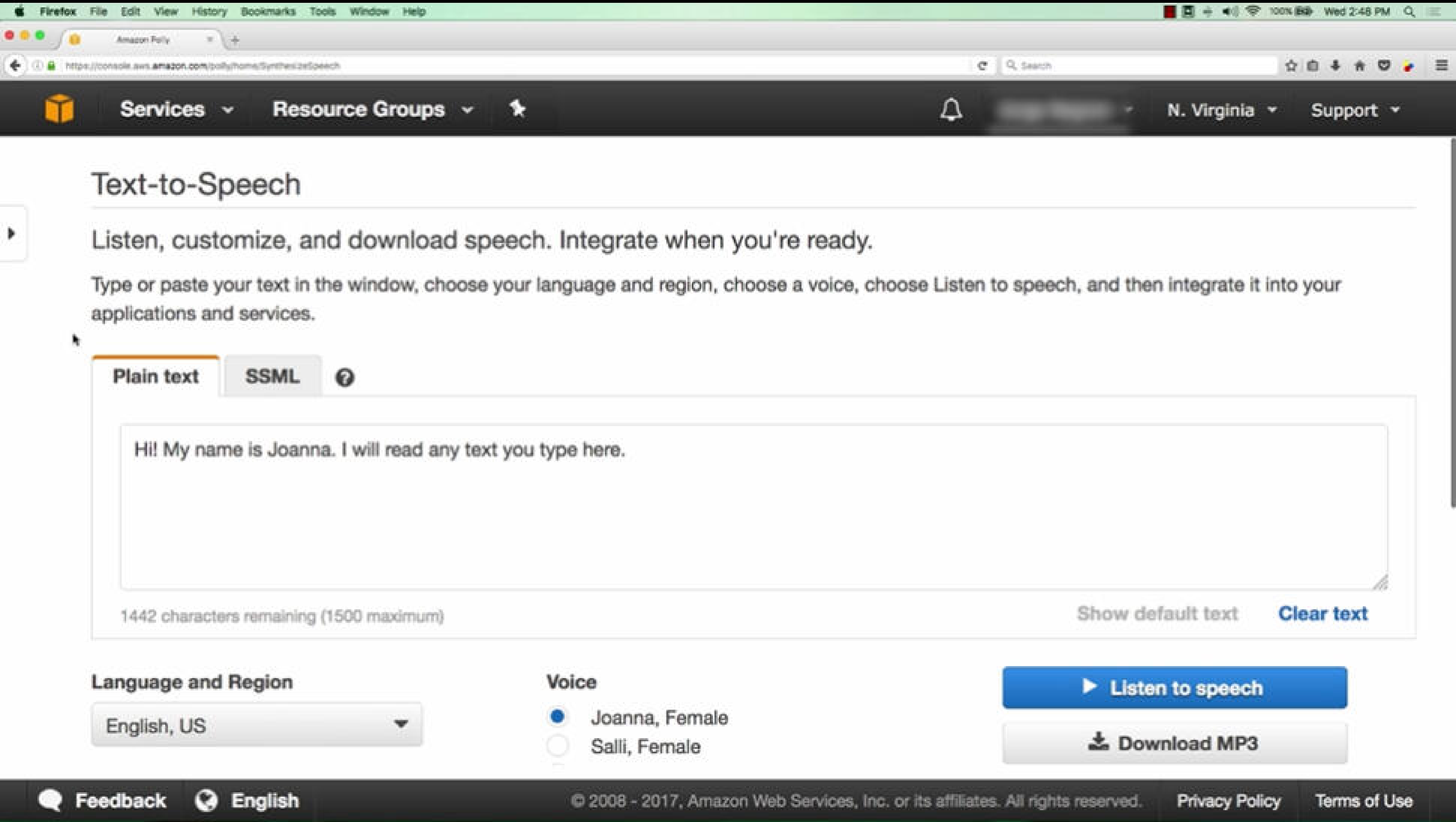

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:48 PM
Amazon Polly
Search
Services
Resource Groups
N. Virginia
Support
Amazon Polly
Text-to-Speech
Lexicons
Text-to-Speech
Listen, customize, and download speech. Integrate when you're ready.
Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Listen to speech
Download MP3
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

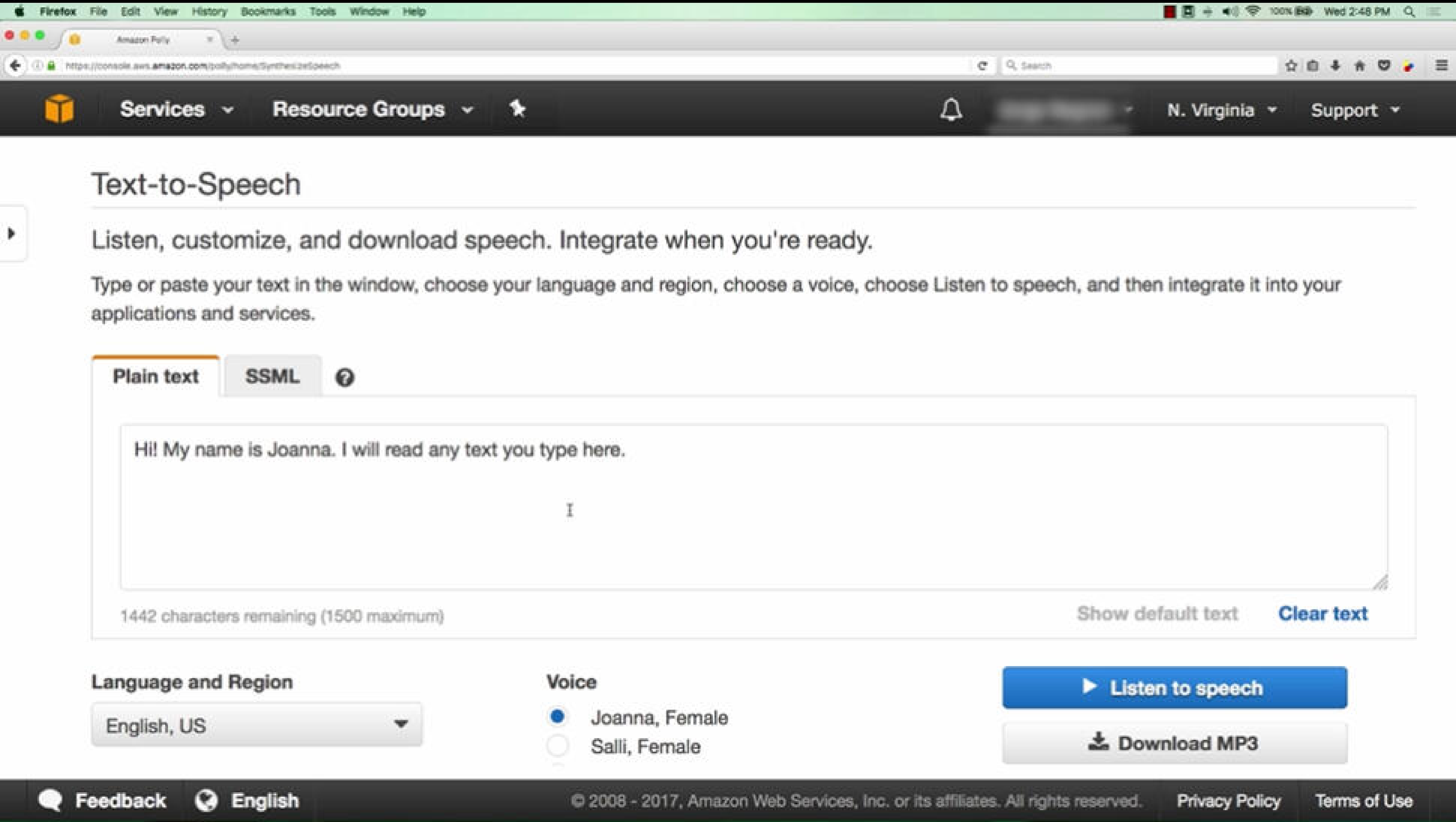


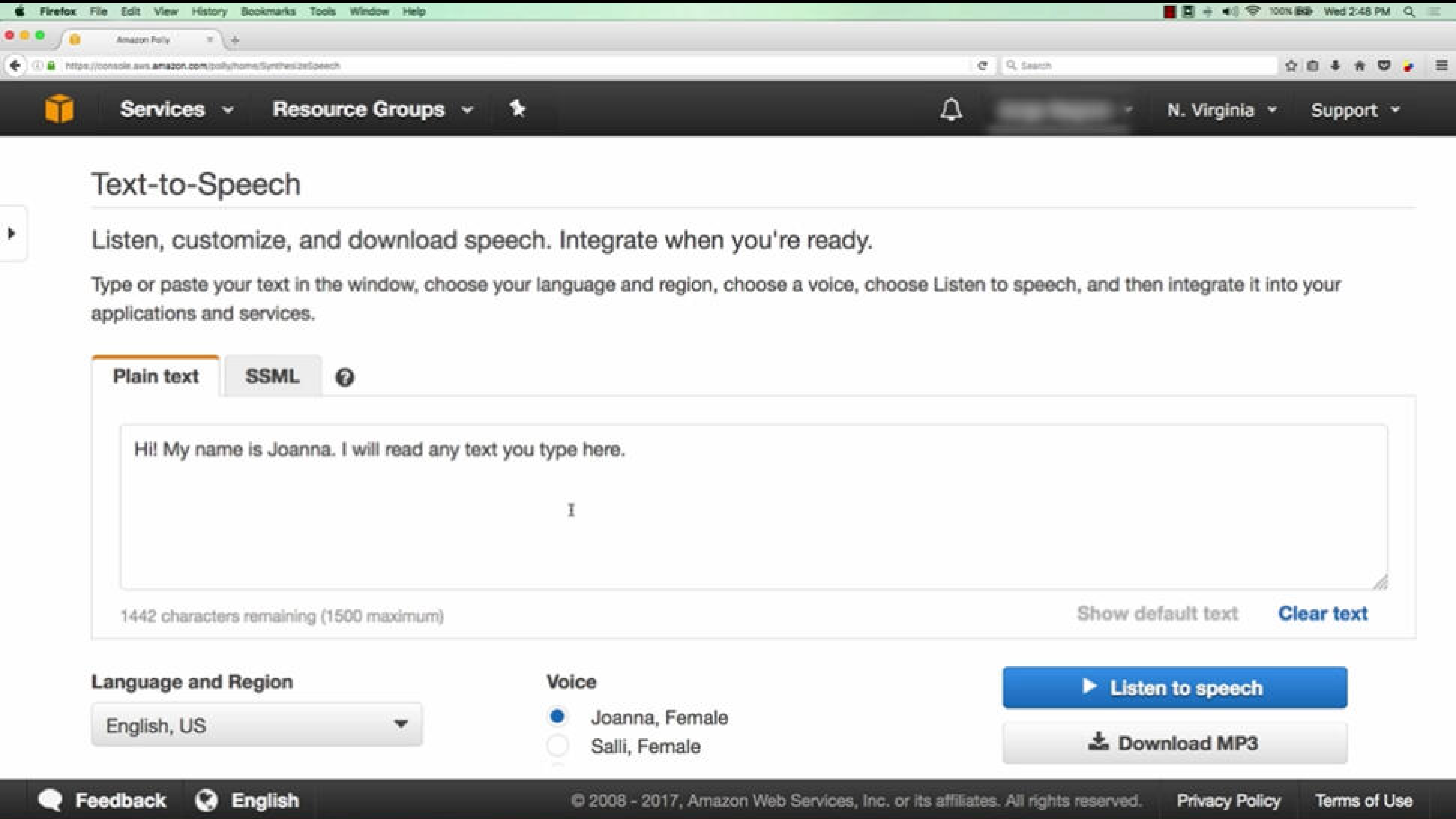
Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:48 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/Lexicons
Search
Services
Resource Groups
N. Virginia
Support
Amazon Polly
Text-to-Speech
Lexicons
Lexicons
You can customize the pronunciation of specific words and phrases by uploading lexicon files in the PLS format.
Learn more
Upload
Download
Delete
Name
Language
Entities
Uploaded
No results. Upload lexicons to customize pronunciation of specific words or phrases.
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

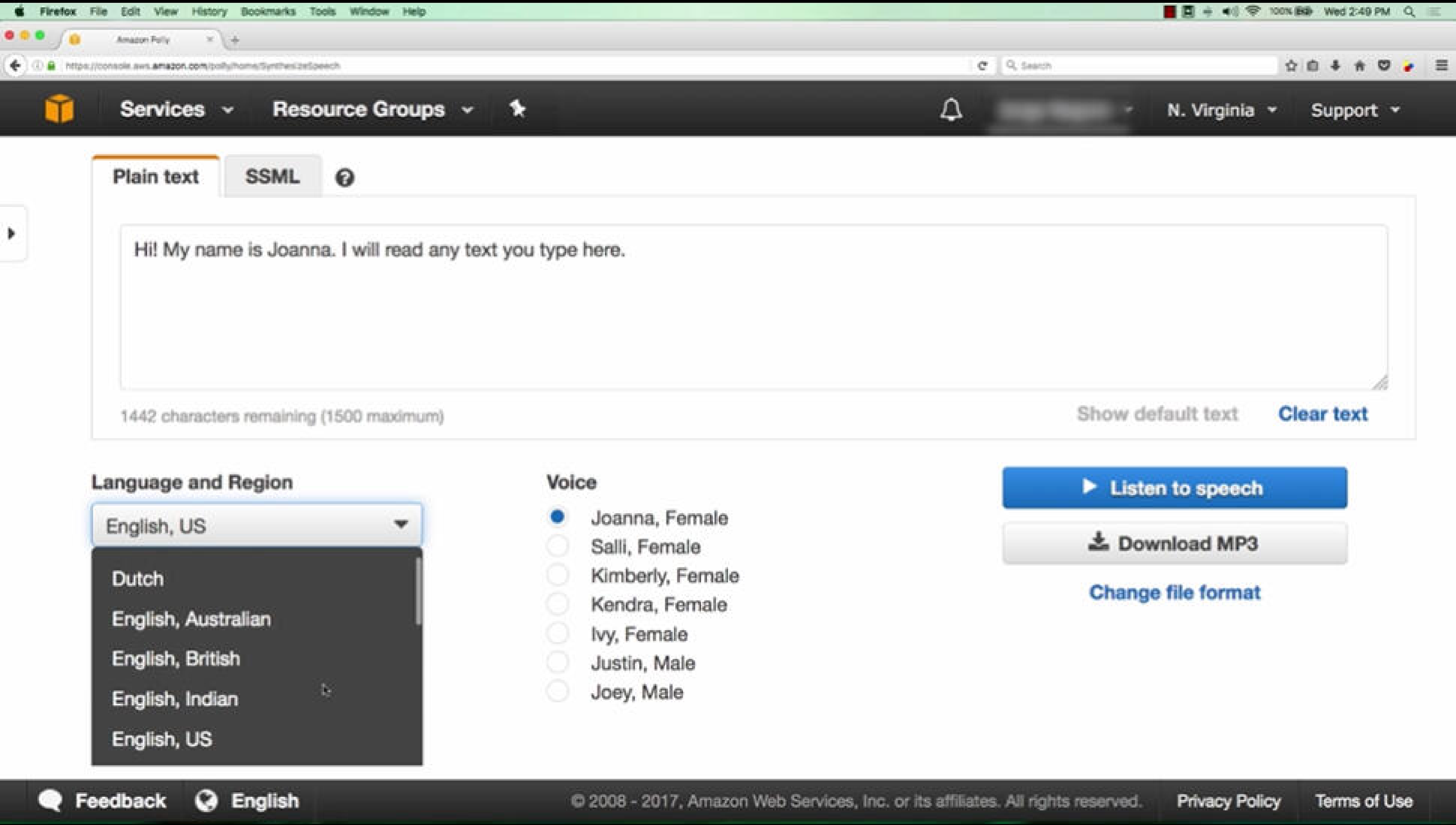

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:48 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/Lexicons
Search
Services
Resource Groups
N. Virginia
Support
Amazon Polly
Text-to-Speech
Lexicons
Lexicons
You can customize the pronunciation of specific words and phrases by uploading lexicon files in the PLS format.
Learn more
Upload
Download
Delete
Name
Language
Entities
Uploaded
No results. Upload lexicons to customize pronunciation of specific words or phrases.
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

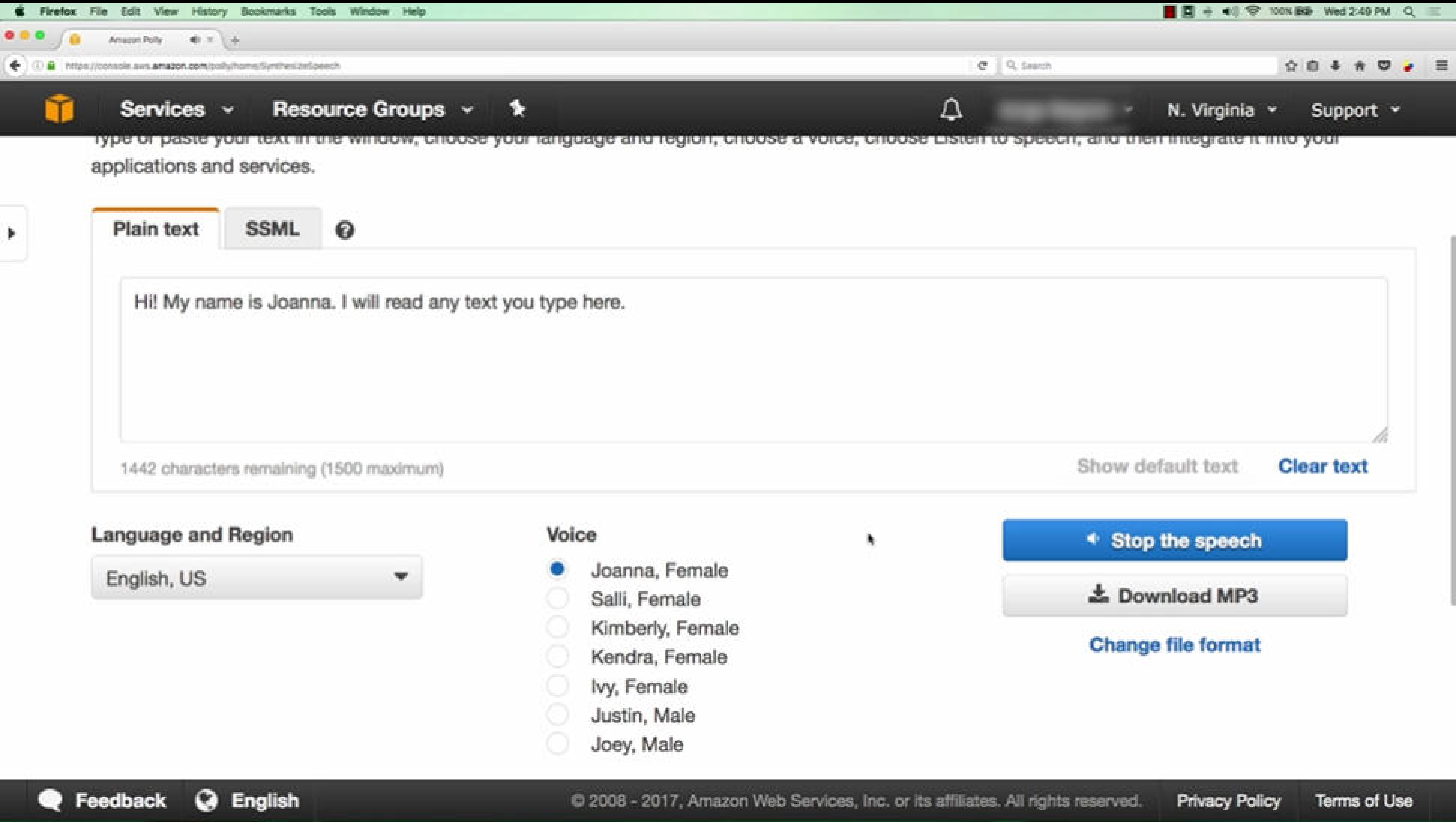

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:48 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
Amazon Polly
Text-to-Speech
Lexicons
Text-to-Speech
Listen, customize, and download speech. Integrate when you're ready.
Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Listen to speech
Download MP3
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

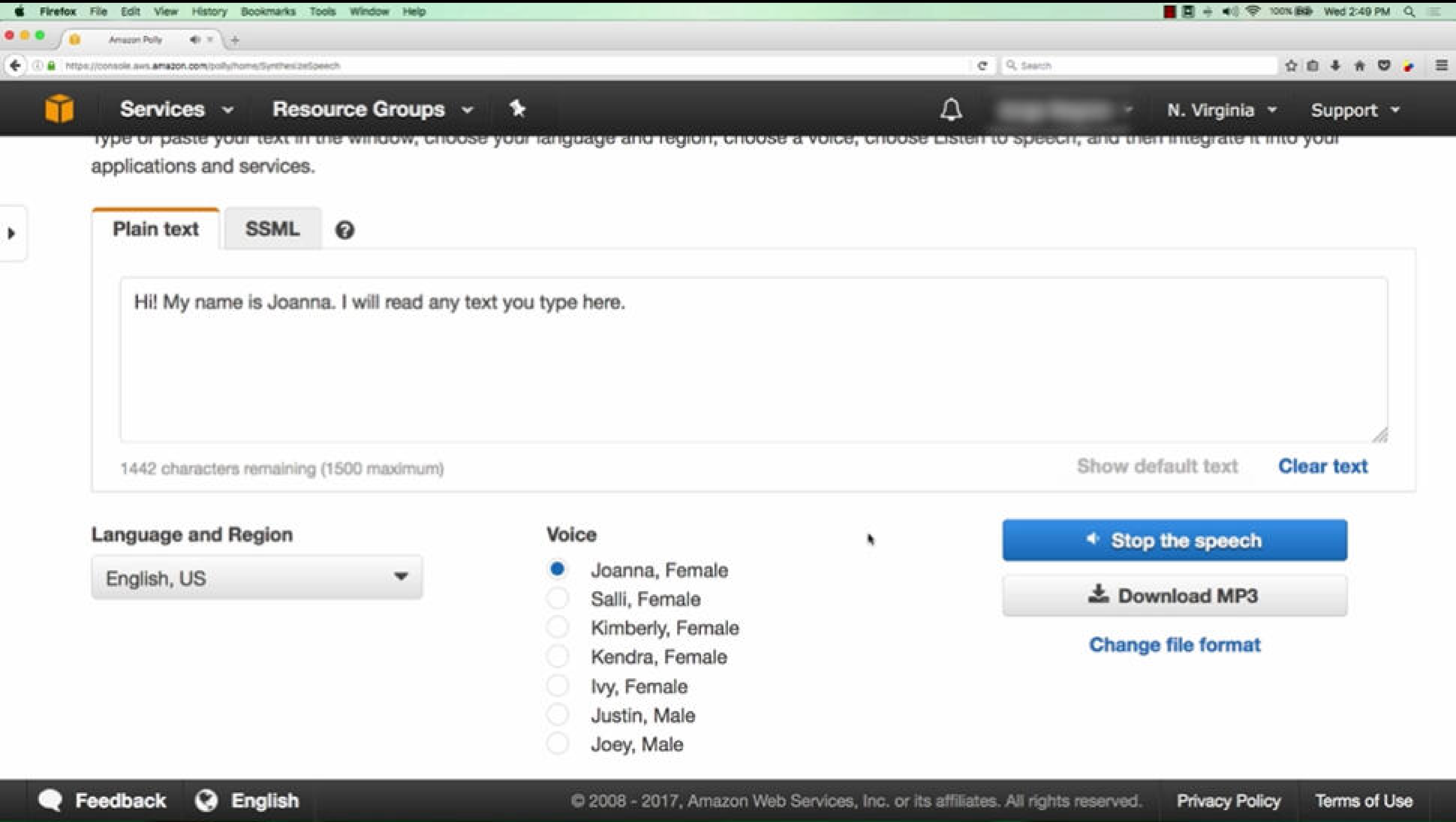

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:48 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
Text-to-Speech
Listen, customize, and download speech. Integrate when you're ready.
Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Listen to speech
Download MP3
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

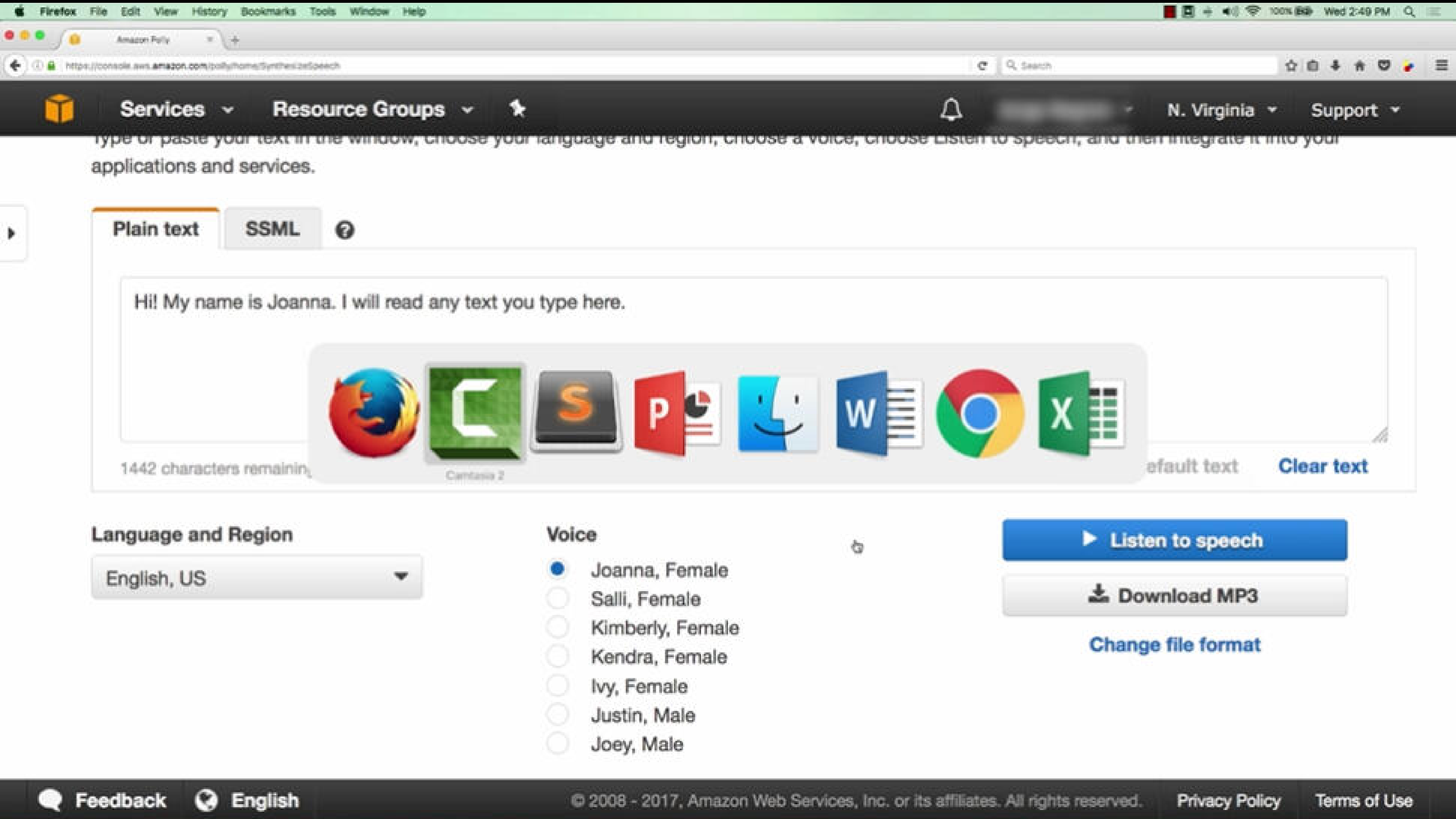

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:48 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Search
Services
Resource Groups
N. Virginia
Support
Text-to-Speech
Listen, customize, and download speech. Integrate when you're ready.
Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Listen to speech
Download MP3
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

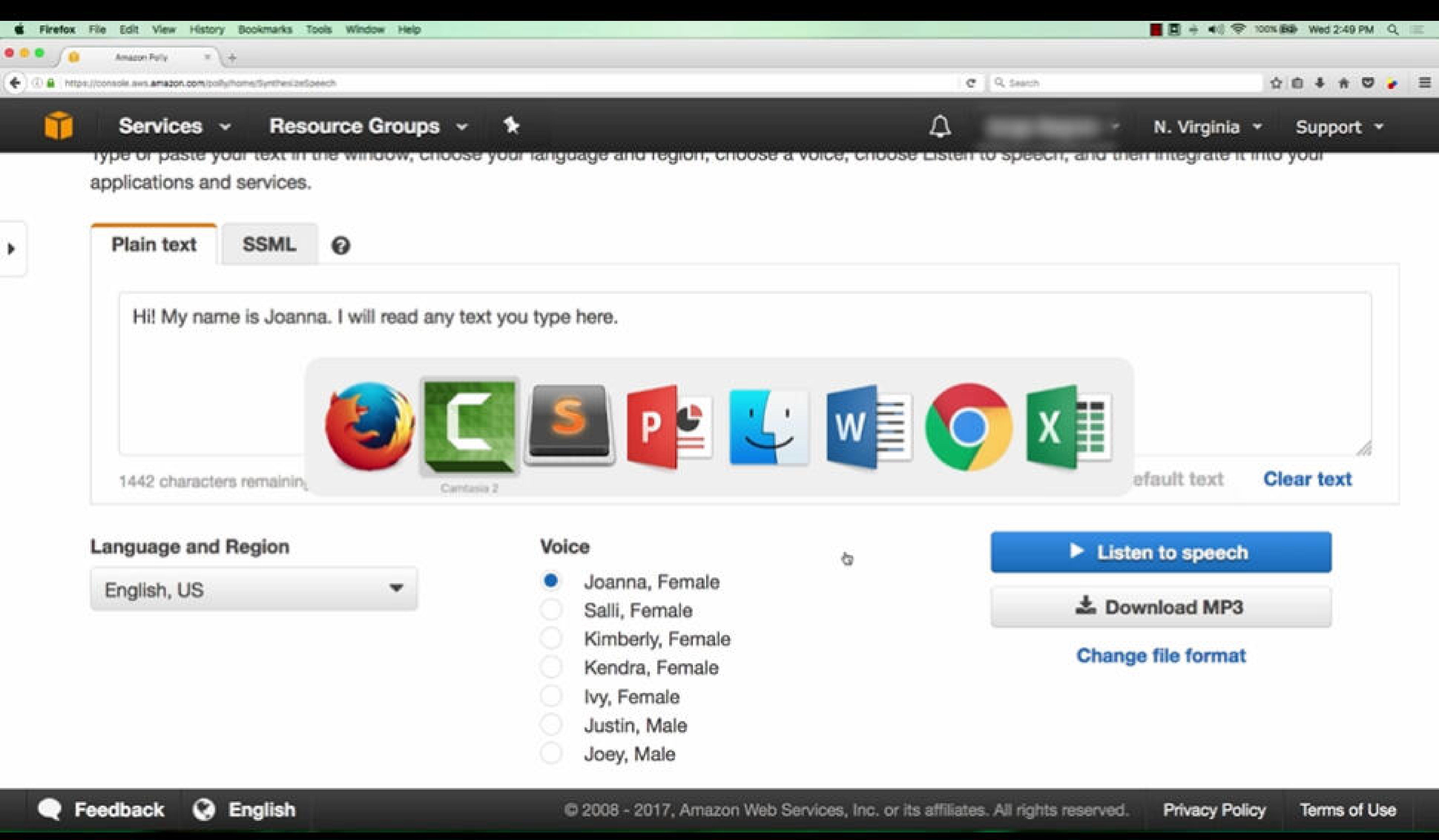

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:48 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
Text-to-Speech
Listen, customize, and download speech. Integrate when you're ready.
Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Listen to speech
Download MP3
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:49 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Search
Services
Resource Groups
N. Virginia
Support
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Dutch
English, Australian
English, British
English, Indian
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

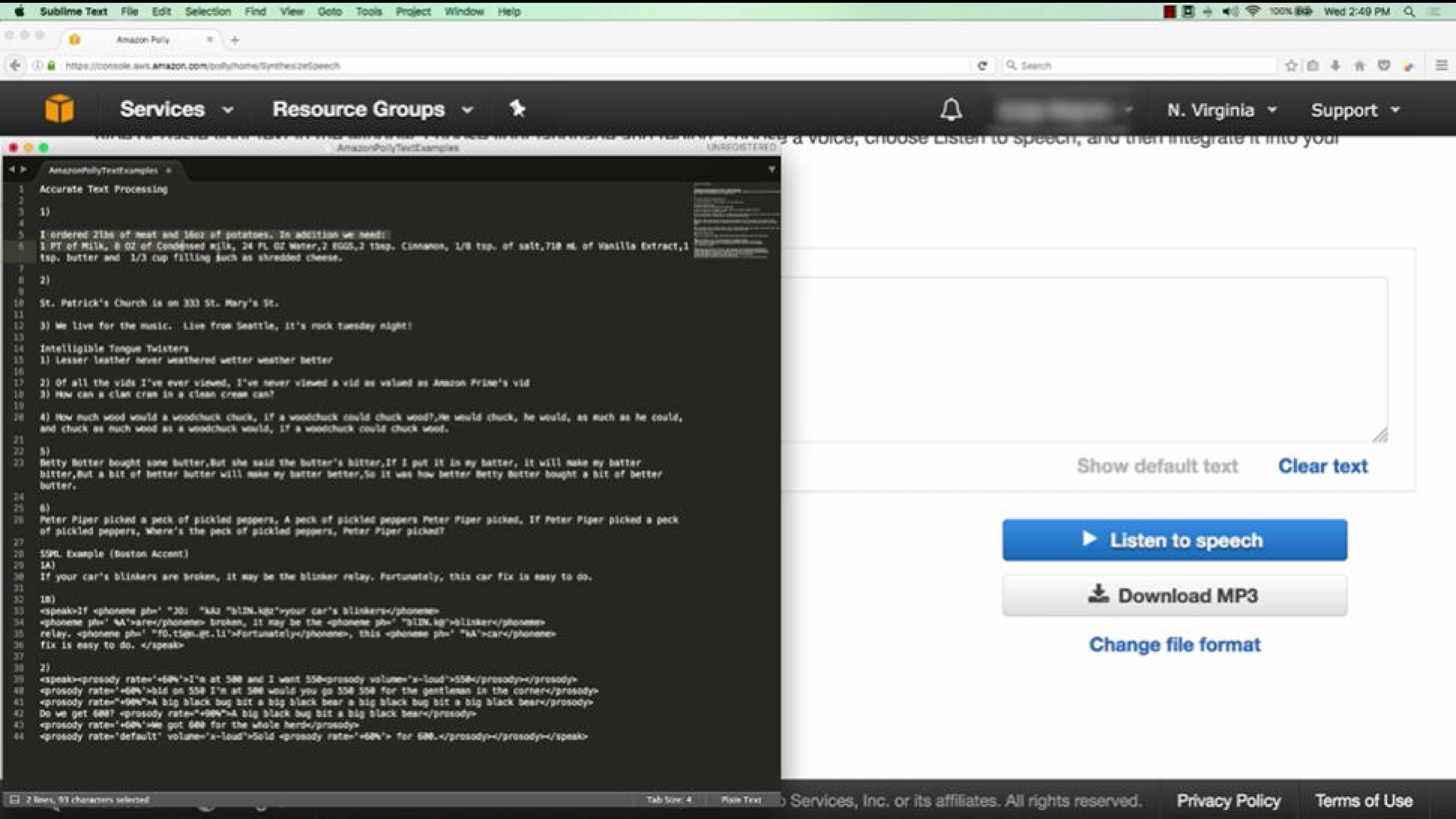

Firefox File Edit View History Bookmarks Tools Window Help
Amazon Polly
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Stop the speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:49 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
1442 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Stop the speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

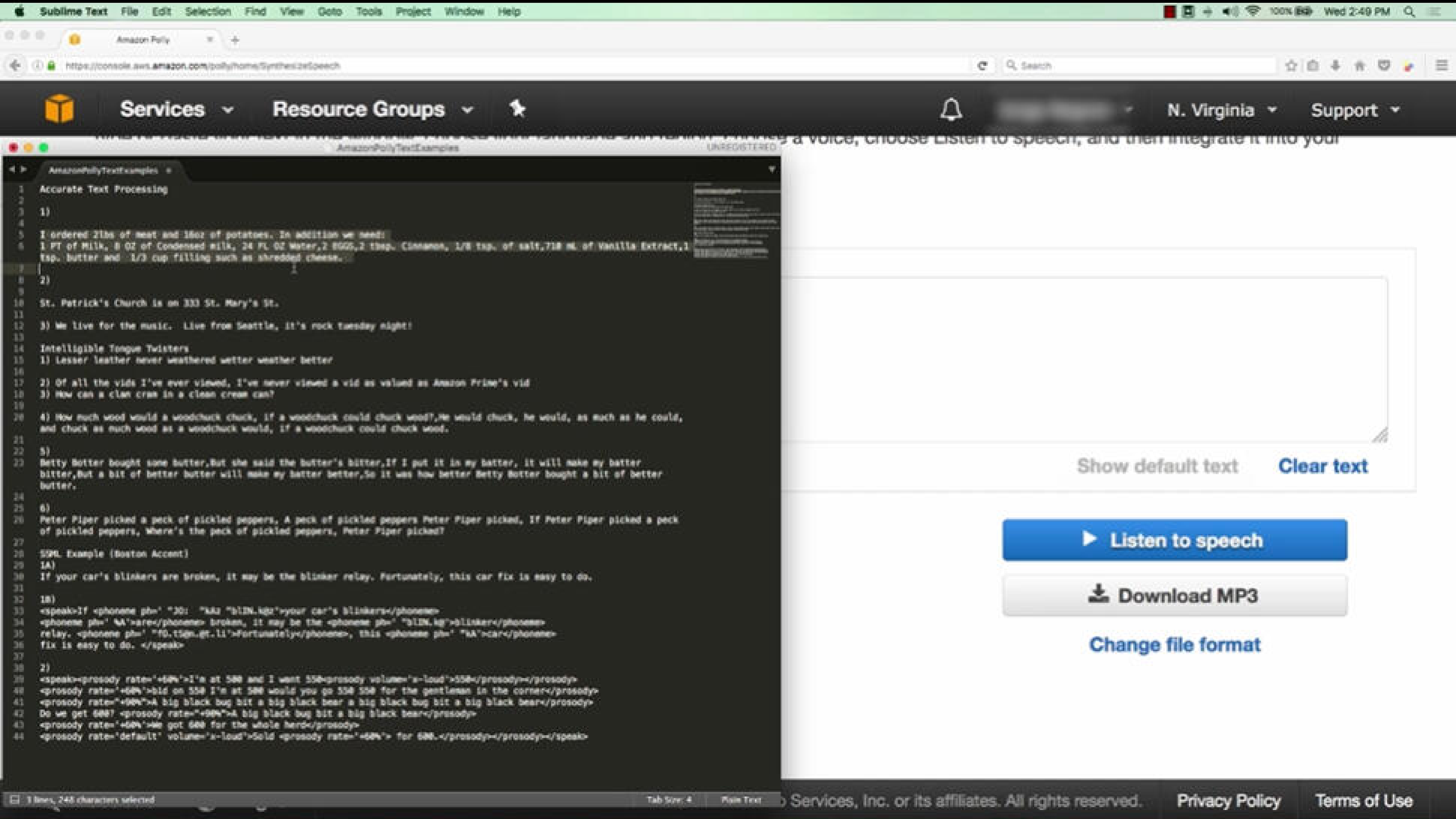

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:49 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
Camtasia 2
1442 characters remainin
efault text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:49 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
Hi! My name is Joanna. I will read any text you type here.
Camtasia 2
1442 characters remainin
efault text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Sublime Text File Edit Selection Find View Goto Tools Project Window Help
Wed 2:49 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
AmazonPollyTextExamples
UNREGISTERED
Accurate Text Processing
1)
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
2)
St. Patrick's Church is on 333 St. Mary's St.
3) We live for the music. Live from Seattle, it's rock tuesday night!
Intelligible Tongue Twisters
1) Lesser leather never weathered wetter weather better
2) Of all the vids I've ever viewed, I've never viewed a vid as valued as Amazon Prime's vid
3) How can a clam cram in a clean cream can?
4) How much wood would a woodchuck chuck, if a woodchuck could chuck wood?,He would chuck, he would, as much as he could, and chuck as much wood as a woodchuck would, if a woodchuck could chuck wood.
5)
Betty Botter bought some butter,But she said the butter's bitter,If I put it in my batter, it will make my batter bitter,But a bit of better butter will make my batter better,So it was how better Betty Botter bought a bit of better butter.
6)
Peter Piper picked a peck of pickled peppers, A peck of pickled peppers Peter Piper picked, If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers, Peter Piper picked?
SSML Example (Boston Accent)
1A)
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1B)
<speak>If <phoneme ph=' "JO: "kAz "blIN.k@z'>your car's blinkers</phoneme>
<phoneme ph=' %A'>are</phoneme> broken, it may be the <phoneme ph=' "blIN.k@'>blinker</phoneme>
relay. <phoneme ph=' "fO.tS@n.@t.li'>Fortunately</phoneme>, this <phoneme ph=' "kA'>car</phoneme>
fix is easy to do. </speak>
2)
<speak><prosody rate='+60%'>I'm at 500 and I want 550<prosody volume='x-loud'>550</prosody></prosody>
<prosody rate='+60%'>bid on 550 I'm at 500 would you go 550 550 for the gentleman in the corner</prosody>
<prosody rate="+90%">A big black bug bit a big black bear a big black bug bit a big black bear</prosody>
Do we get 600? <prosody rate="+90%">A big black bug bit a big black bear</prosody>
<prosody rate='+60%'>We got 600 for the whole herd</prosody>
<prosody rate='default' volume='x-loud'>Sold <prosody rate='+60%'> for 600.</prosody></prosody></speak>
2 lines, 93 characters selected
Tab Size: 4
Plain Text
Show default text
Clear text
Listen to speech
Download MP3
Change file format
Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
Wed 2:49 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
AmazonPollyTextExamples
Accurate Text Processing
1)
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
2)
St. Patrick's Church is on 333 St. Mary's St.
3) We live for the music. Live from Seattle, it's rock tuesday night!
Intelligible Tongue Twisters
1) Lesser leather never weathered wetter weather better
2) Of all the vids I've ever viewed, I've never viewed a vid as valued as Amazon Prime's vid
3) How can a clam cram in a clean cream can?
4) How much wood would a woodchuck chuck, if a woodchuck could chuck wood?,He would chuck, he would, as much as he could, and chuck as much wood as a woodchuck would, if a woodchuck could chuck wood.
5)
Betty Botter bought some butter,But she said the butter's bitter,If I put it in my batter, it will make my batter bitter,But a bit of better butter will make my batter better,So it was how better Betty Botter bought a bit of better butter.
6)
Peter Piper picked a peck of pickled peppers, A peck of pickled peppers Peter Piper picked, If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers, Peter Piper picked?
SSML Example (Boston Accent)
1A)
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1B)
fix is easy to do. </speak>
2)
<speak><prosody rate='+60%'>I'm at 500 and I want 550<prosody volume='x-loud'>550</prosody></prosody>
<prosody rate='+60%'>bid on 550 I'm at 500 would you go 550 550 for the gentleman in the corner</prosody>
<prosody rate="+90%">A big black bug bit a big black bear a big black bug bit a big black bear</prosody>
Do we get 600? <prosody rate="+90%">A big black bug bit a big black bear</prosody>
<prosody rate='+60%'>We got 600 for the whole herd</prosody>
<prosody rate='default' volume='x-loud'>Sold <prosody rate='+60%'> for 600.</prosody></prosody></speak>
3 lines, 248 characters selected
Tab Size: 4
Plain Text
Show default text
Clear text
Listen to speech
Download MP3
Change file format
Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Firefox File Edit View History Bookmarks Tools Window Help

Amazon Polly

https://console.aws.amazon.com/polly/home/SynthesizeSpeech

Services Resource Groups N. Virginia Support

Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.

Plain text | SSML

I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.

1252 characters remaining (1500 maximum)

Show default text Clear text

Language and Region

English, US

Voice

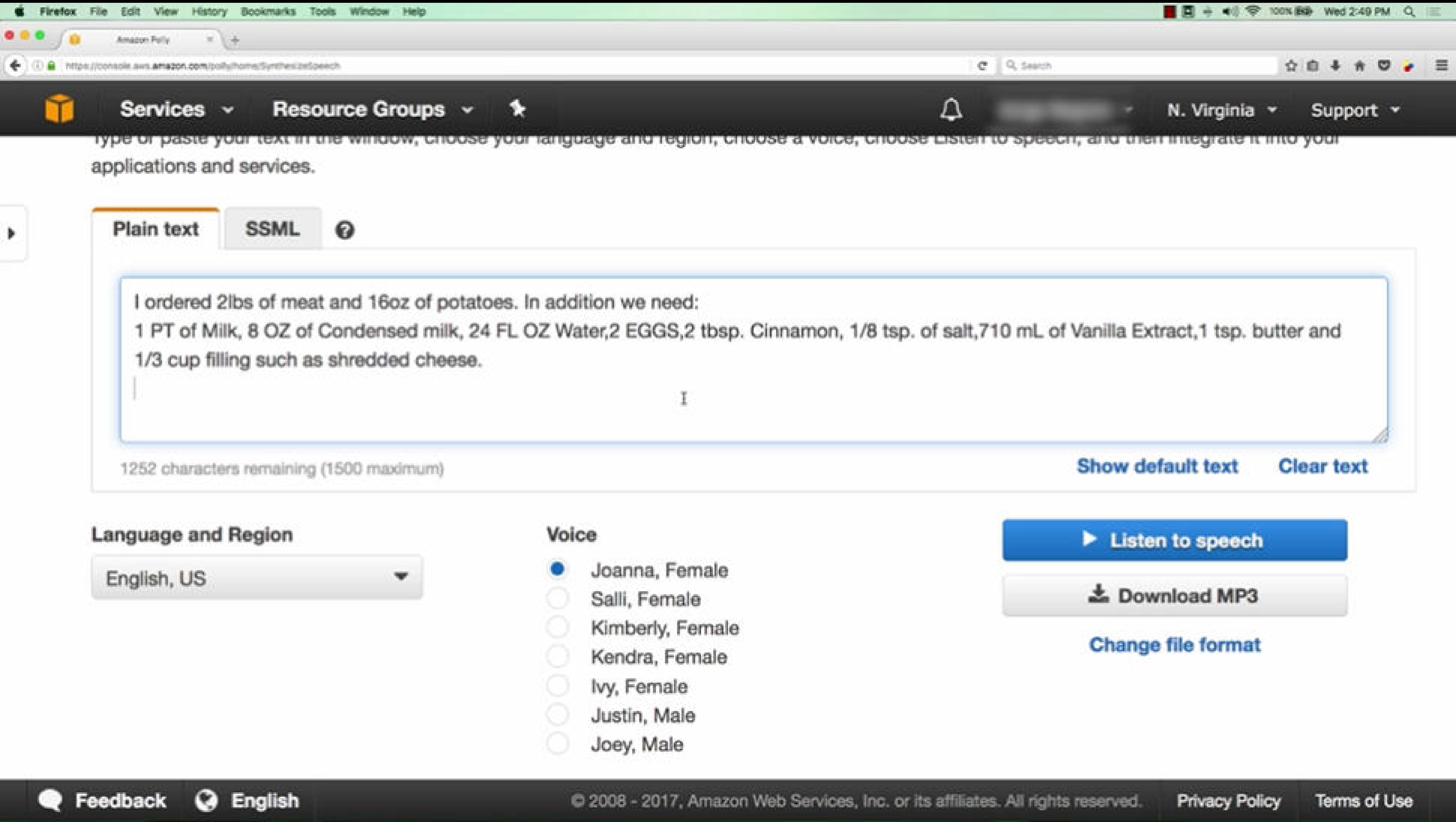
- Joanna, Female
- Salli, Female
- Kimberly, Female
- Kendra, Female
- Ivy, Female
- Justin, Male
- Joey, Male

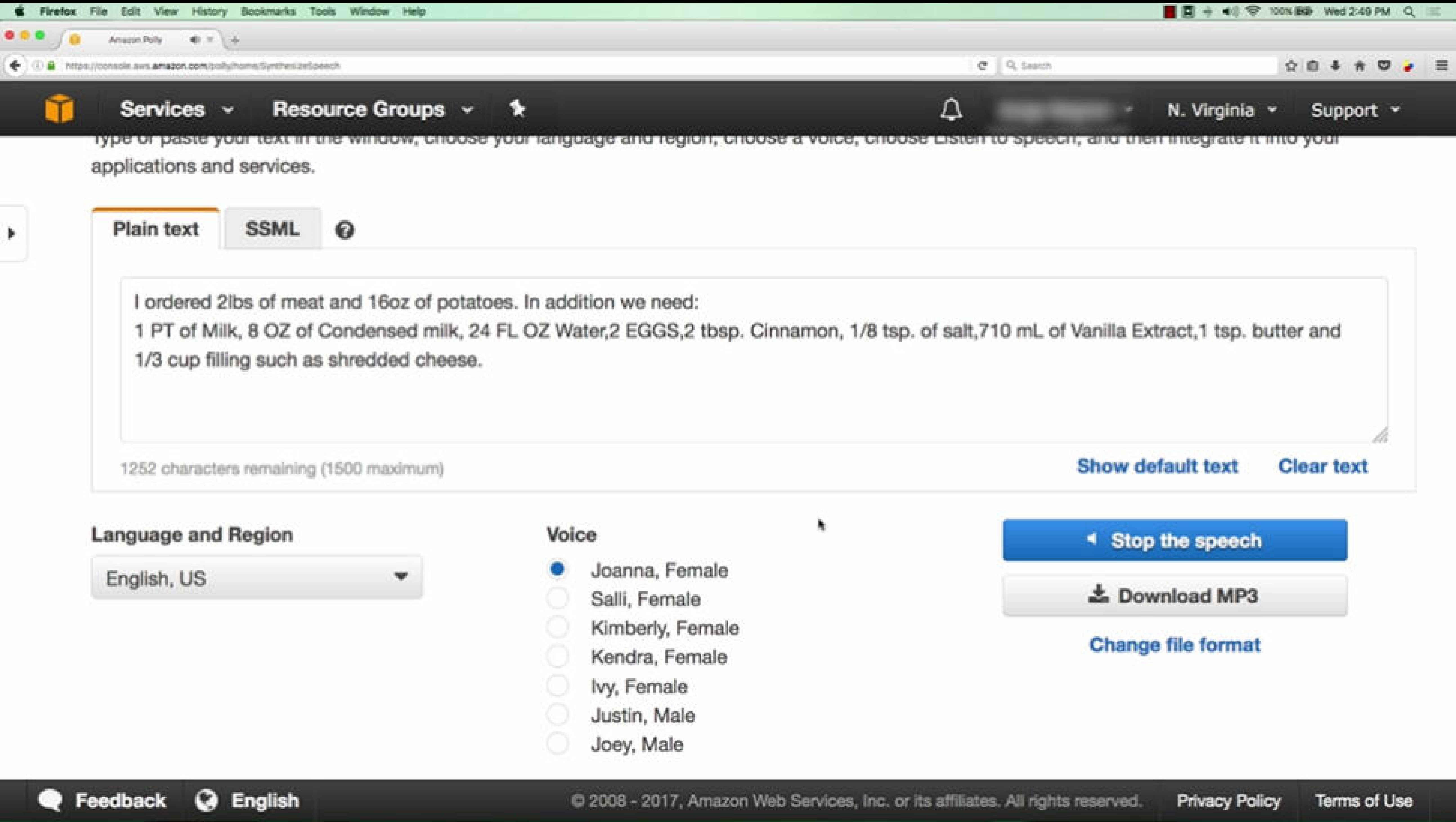
Listen to speech

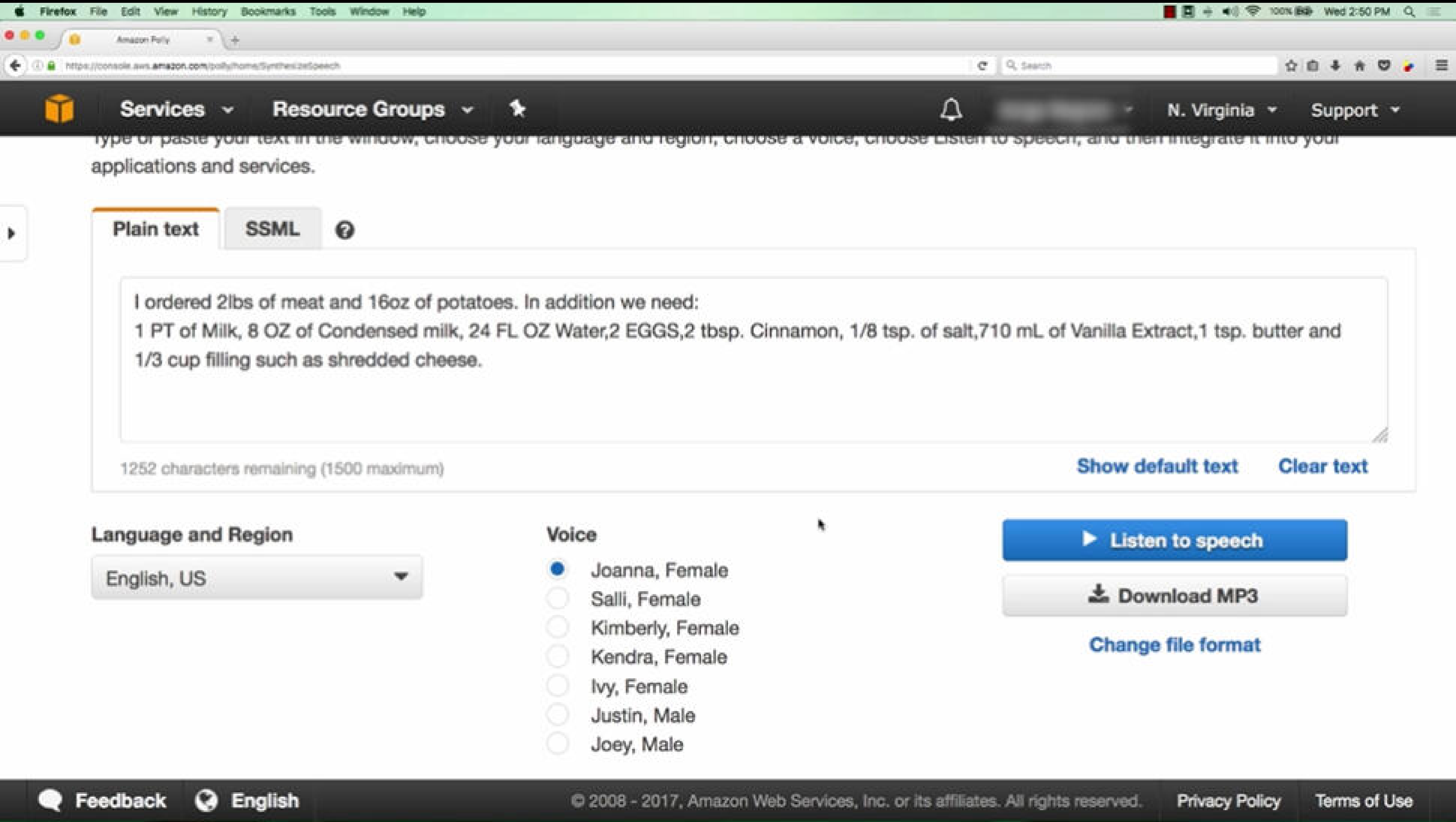
Download MP3

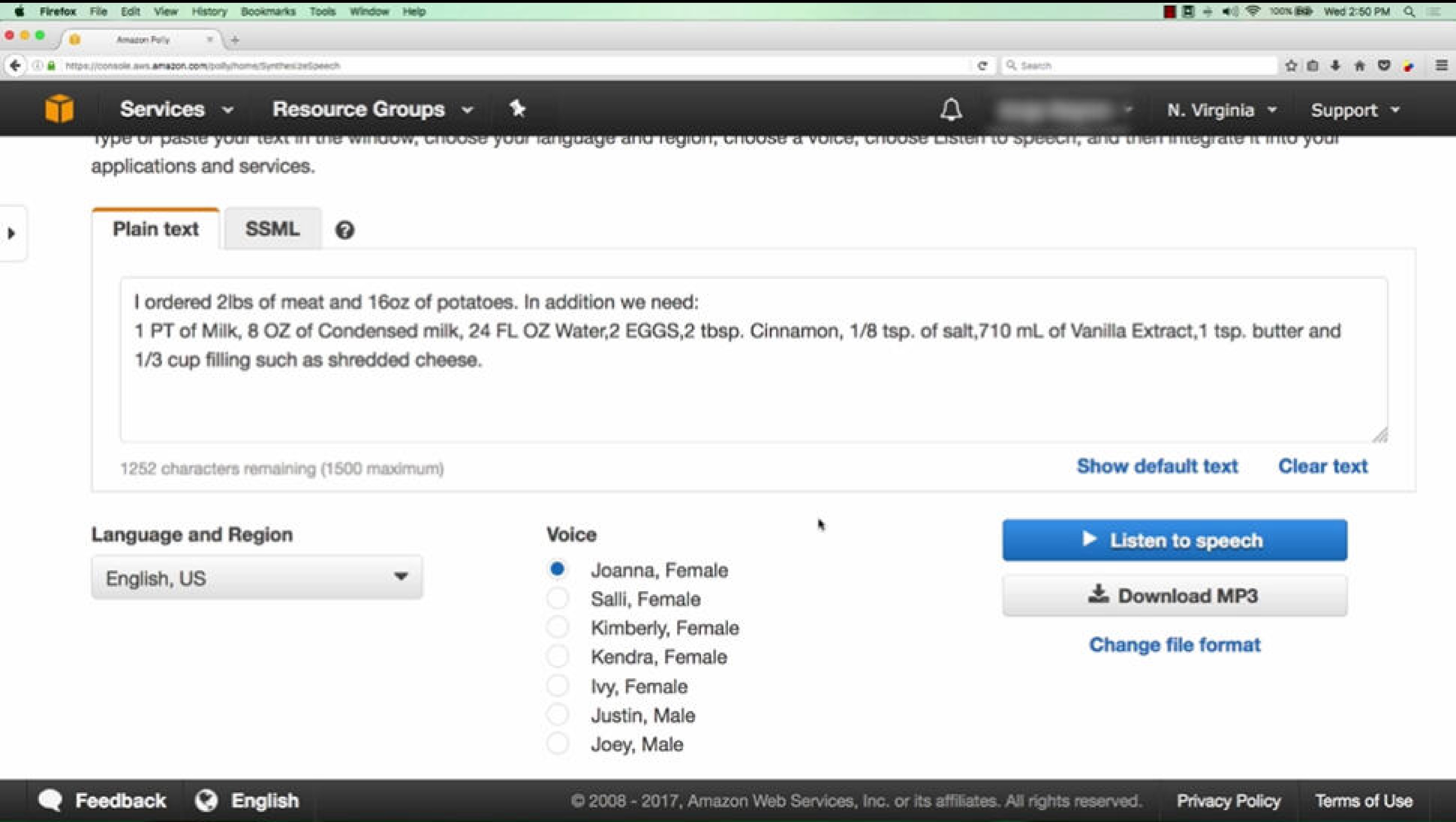
Change file format

Feedback English Privacy Policy Terms of Use

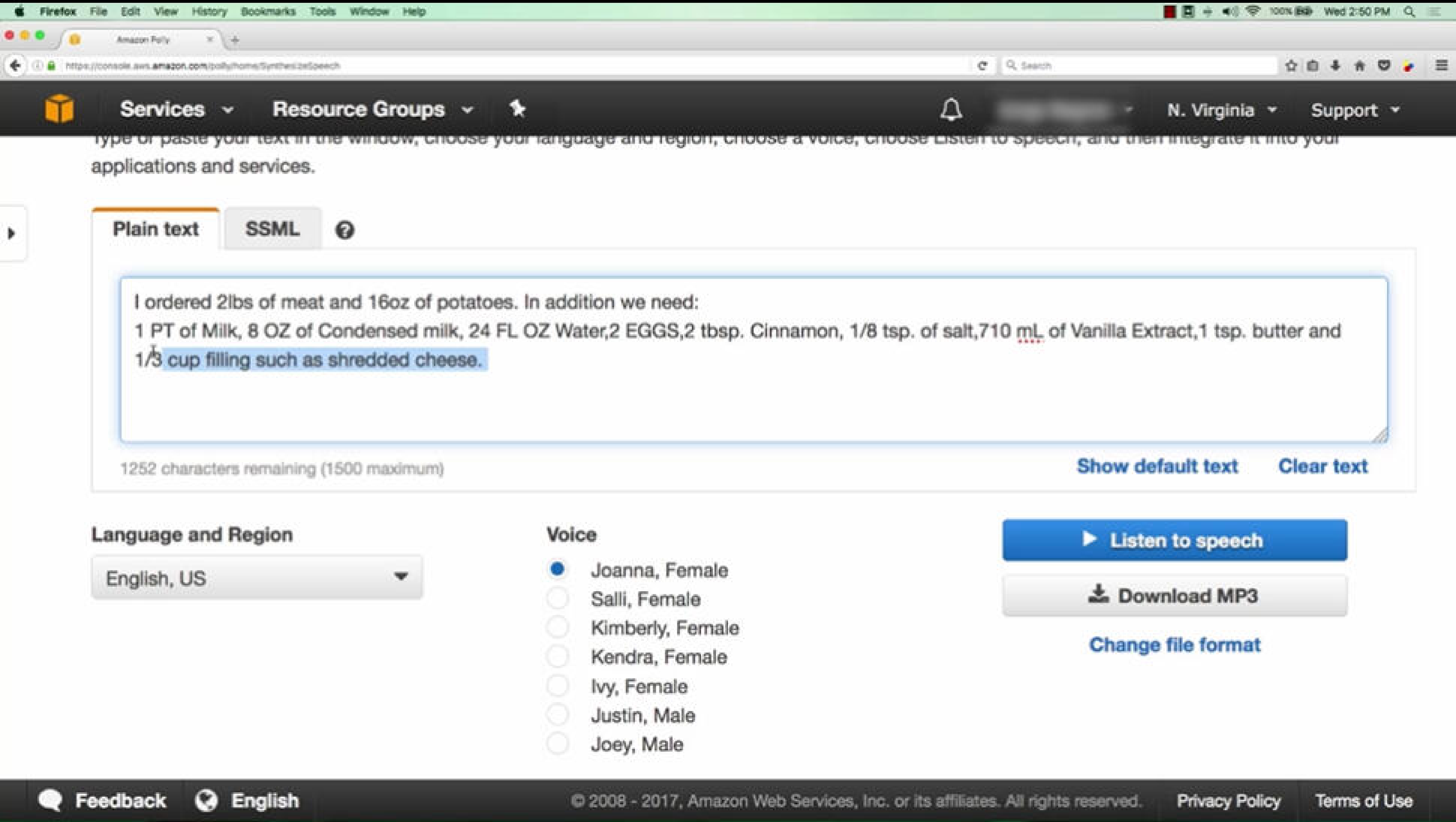

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:49 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Search
Services
Resource Groups
N. Virginia
Support
Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.
Plain text
SSML
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
1252 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

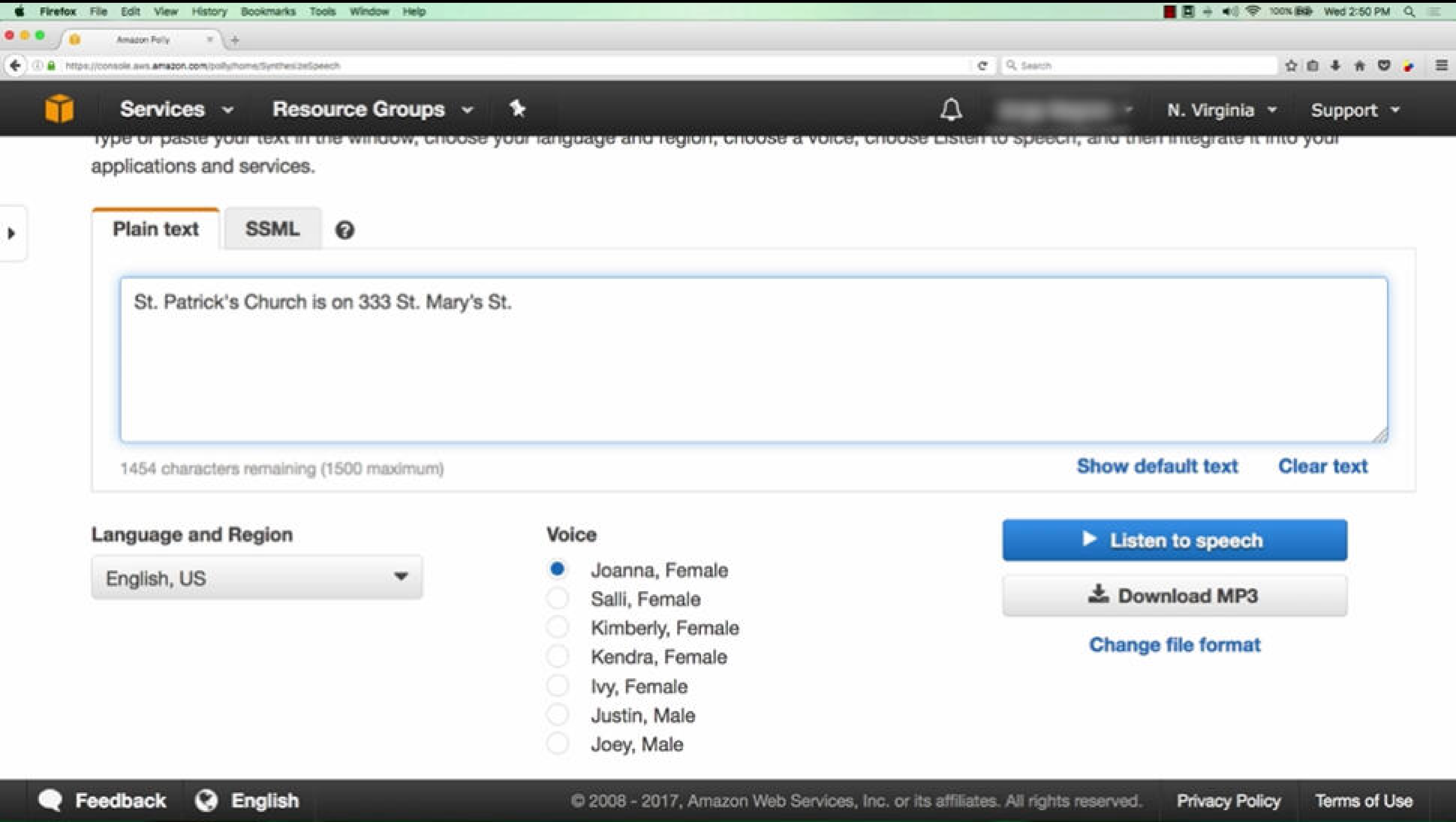

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:49 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
1252 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Stop the speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

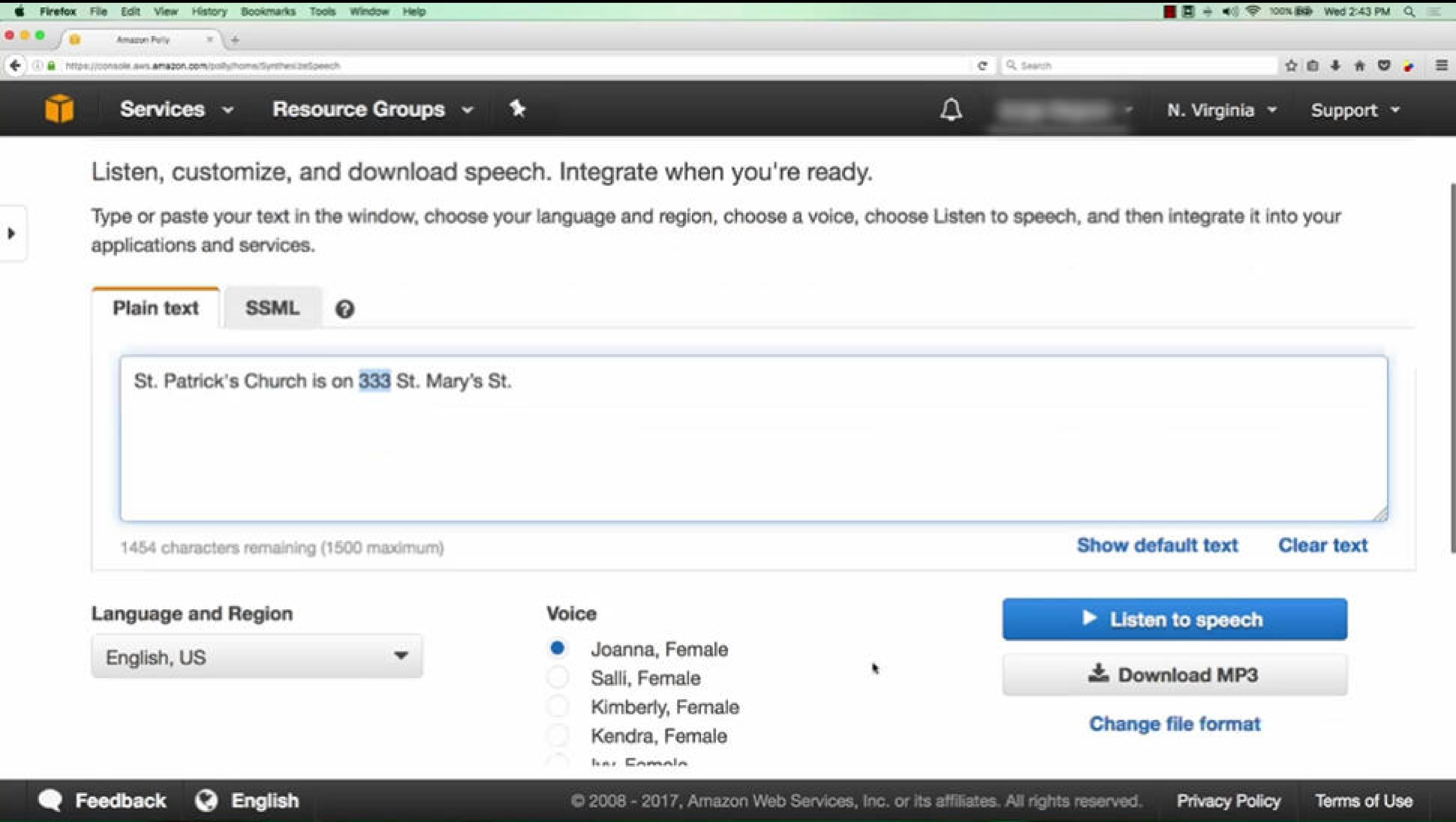

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:50 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
1252 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:50 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
1252 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:50 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
1252 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Services
Resource Groups
N. Virginia
Support
applications and services.
Plain text
SSML
St. Patrick's Church is on 333 St. Mary's St.
1454 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:43 PM
Amazon Polly
Search
Services
Resource Groups
N. Virginia
Support
Listen, customize, and download speech. Integrate when you're ready.
Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.
Plain text
SSML
St. Patrick's Church is on 333 St. Mary's St.
1454 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Listen to speech
Download MP3
Change file format
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Firefox File Edit View History Bookmarks Tools Window Help Wed 2:43 PM

Amazon Polly

https://console.aws.amazon.com/polly/home/SynthesizeSpeech

Services Resource Groups [redacted] N. Virginia Support

Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.

Plain text | SSML

St. Patrick's Church is on 333 St. Mary's St.

1454 characters remaining (1500 maximum)

Show default text Clear text

Language and Region

English, US

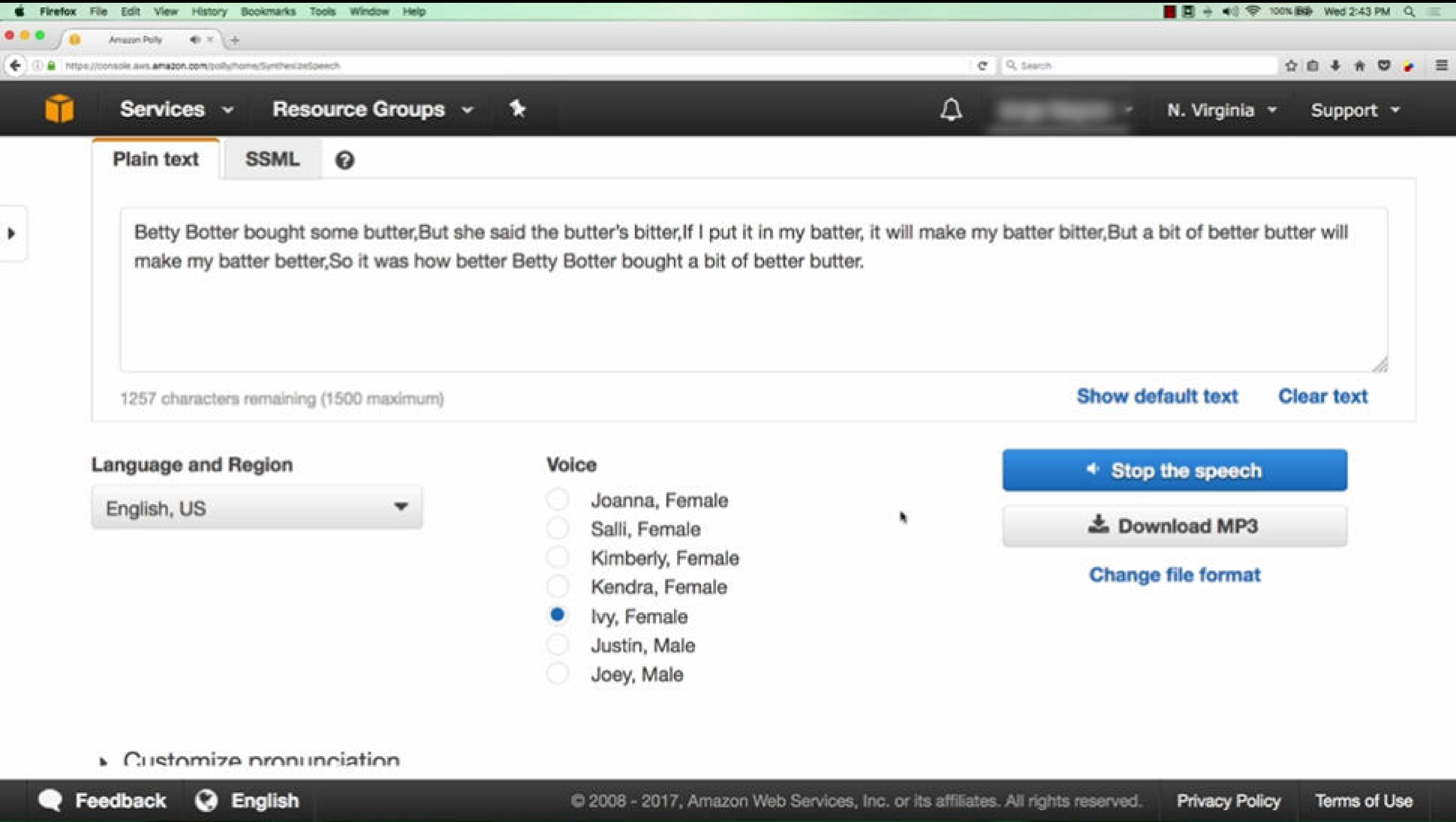
Voice

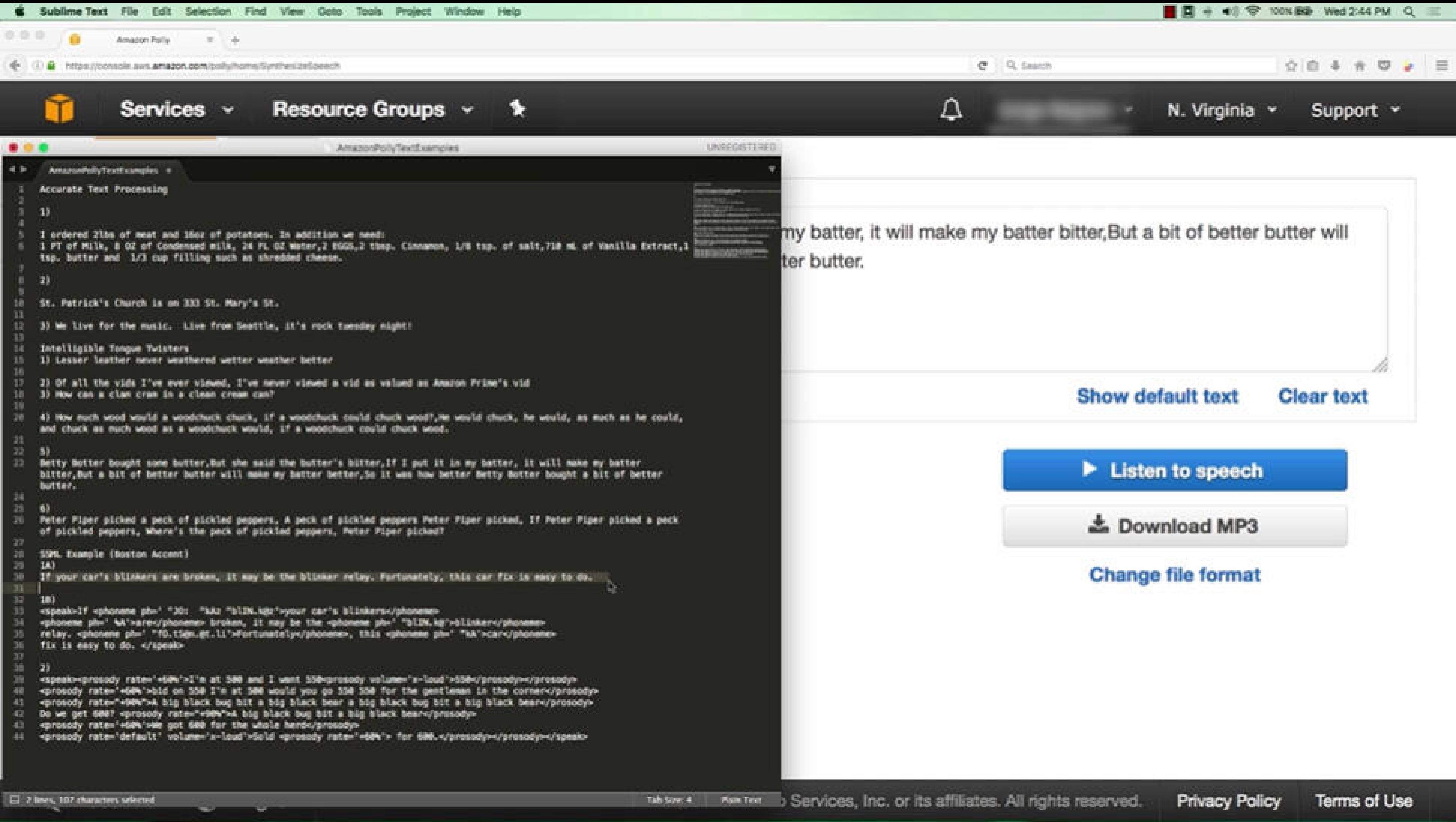
- Joanna, Female
- Salli, Female
- Kimberly, Female
- Kendra, Female
- Ivy, Female
- Justin, Male

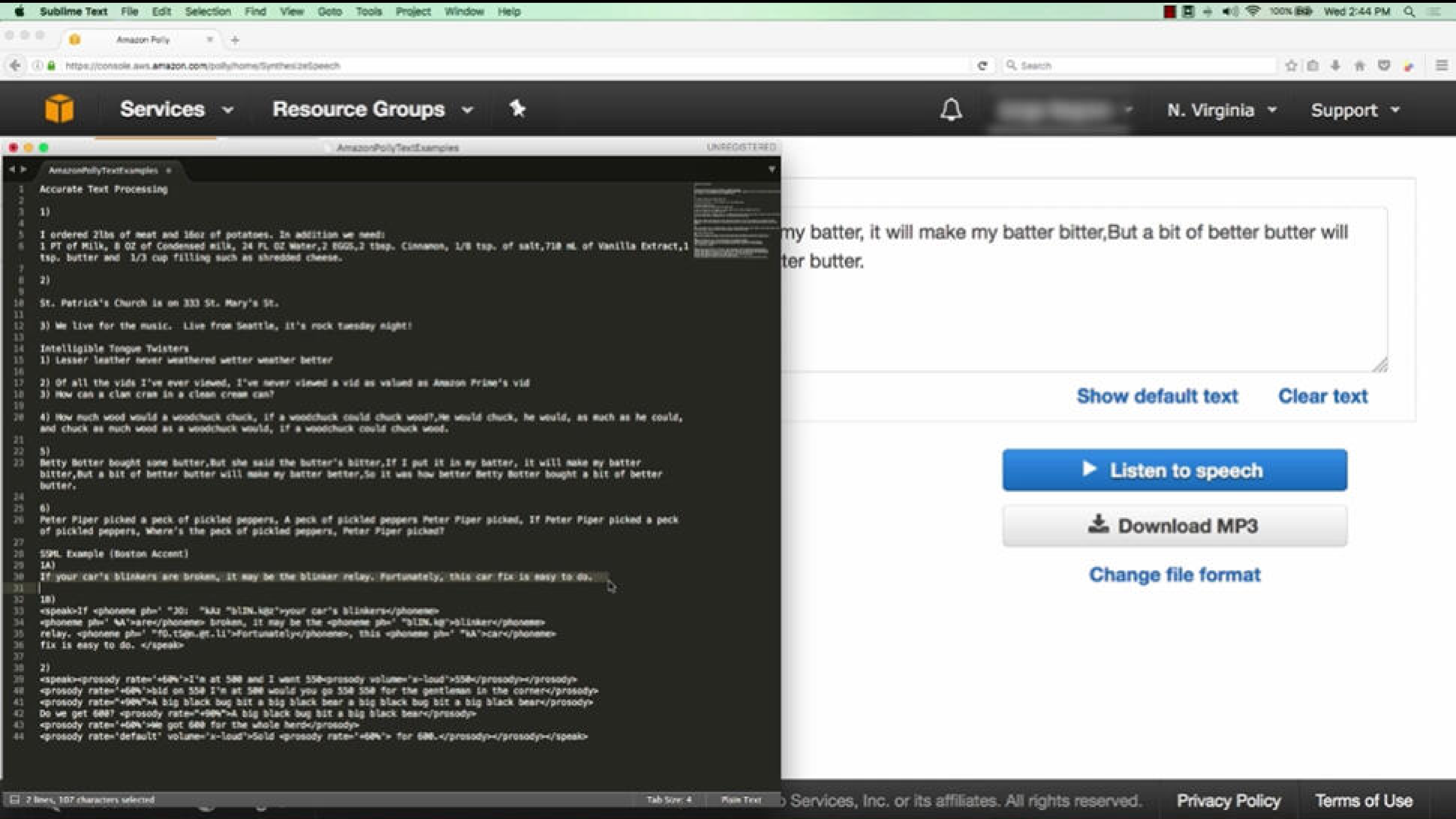
Listen to speech

Download MP3

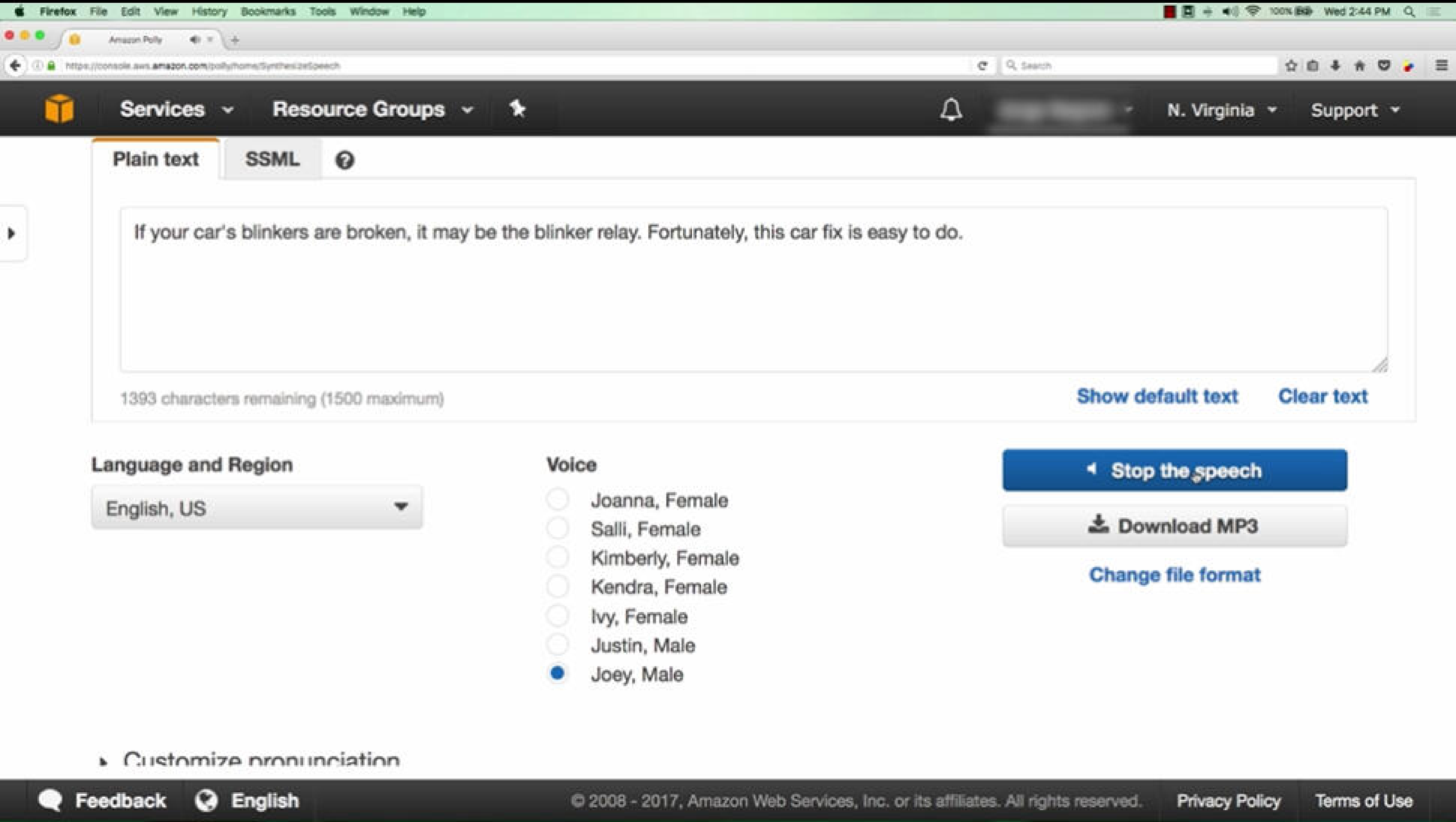
Change file format

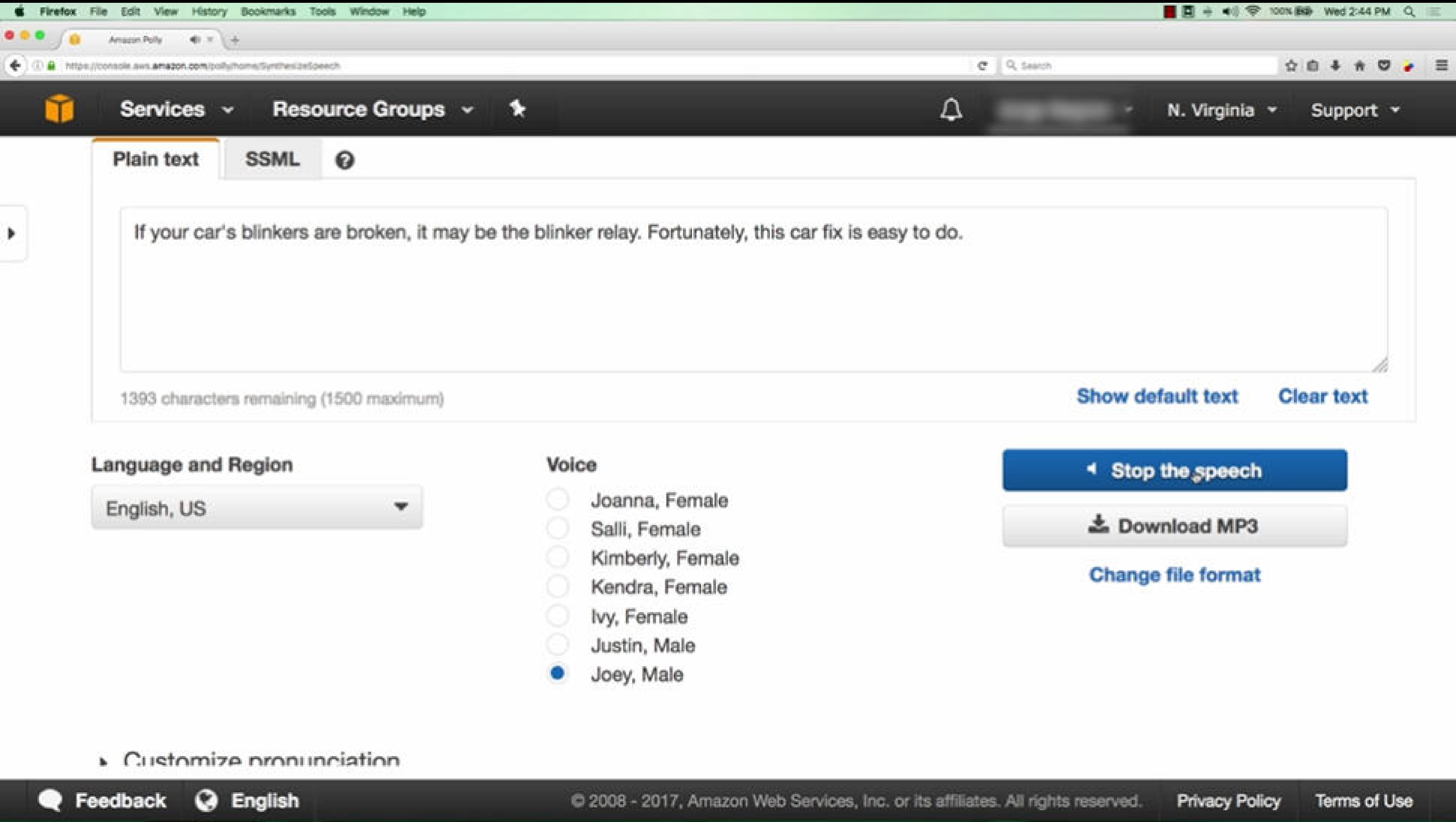

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:43 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Search
Services
Resource Groups
N. Virginia
Support
Plain text
SSML
Betty Botter bought some butter,But she said the butter's bitter,If I put it in my batter, it will make my batter bitter,But a bit of better butter will make my batter better,So it was how better Betty Botter bought a bit of better butter.
1257 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Stop the speech
Download MP3
Change file format
Customize pronunciation
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
Wed 2:44 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
AmazonPollyTextExamples
UNREGISTERED
Accurate Text Processing
1)
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
2)
St. Patrick's Church is on 333 St. Mary's St.
3) We live for the music. Live from Seattle, it's rock tuesday night!
Intelligible Tongue Twisters
1) Lesser leather never weathered wetter weather better
2) Of all the vids I've ever viewed, I've never viewed a vid as valued as Amazon Prime's vid
3) How can a clam cram in a clean cream can?
4) How much wood would a woodchuck chuck, if a woodchuck could chuck wood?,He would chuck, he would, as much as he could, and chuck as much wood as a woodchuck would, if a woodchuck could chuck wood.
5)
Betty Botter bought some butter,But she said the butter's bitter,If I put it in my batter, it will make my batter bitter,But a bit of better butter will make my batter better,So it was how better Betty Botter bought a bit of better butter.
6)
Peter Piper picked a peck of pickled peppers, A peck of pickled peppers Peter Piper picked, If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers, Peter Piper picked?
SSML Example (Boston Accent)
1A)
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1B)
fix is easy to do. </speak>
2)
<prosody rate='default' volume='x-loud'>Sold <prosody rate='+60%'> for 600.</prosody></prosody></speak>
2 lines, 107 characters selected
Tab Size: 4
Plain Text
my batter, it will make my batter bitter,But a bit of better butter will
tter butter.
Show default text
Clear text
Listen to speech
Download MP3
Change file format
Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

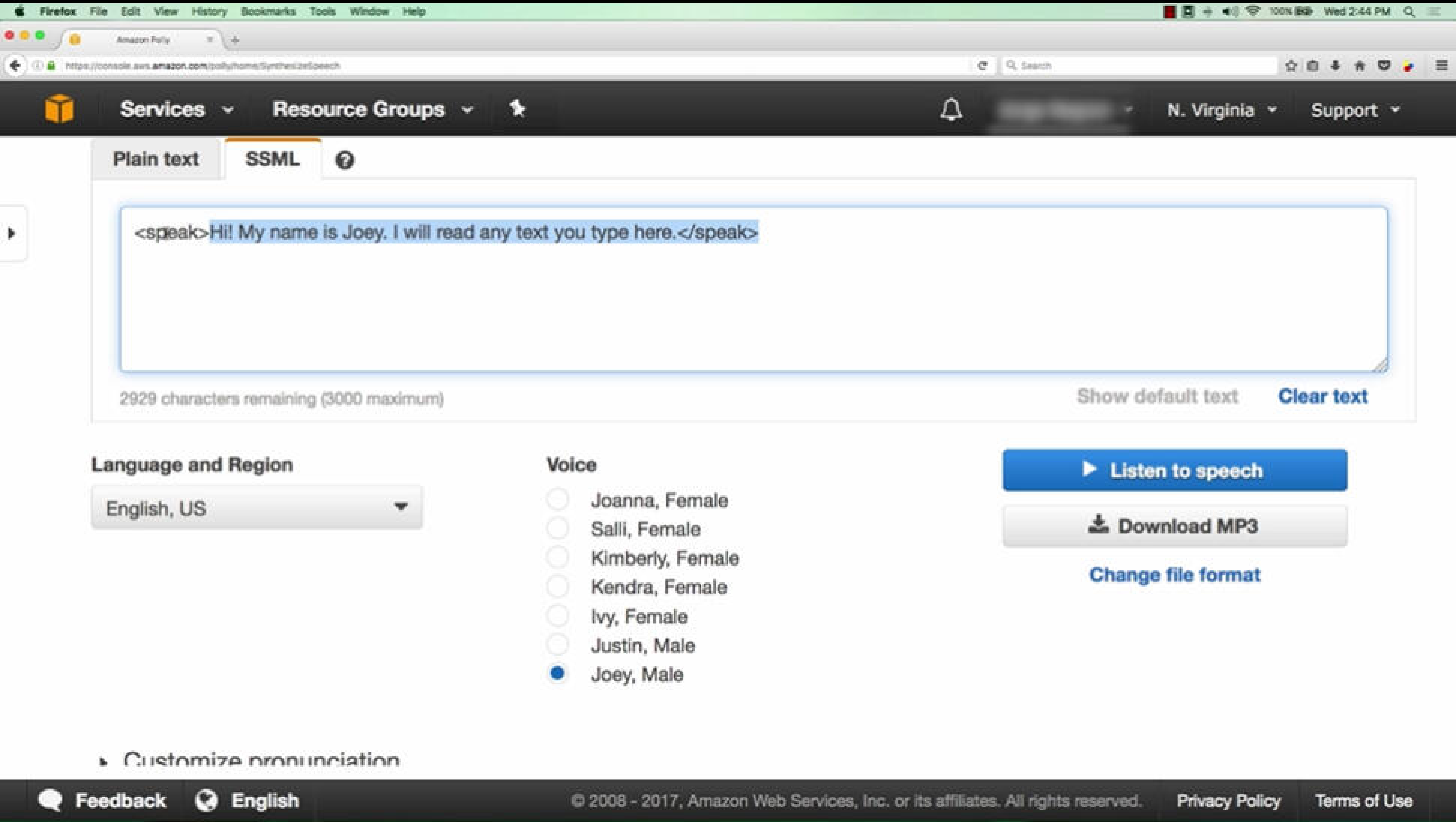


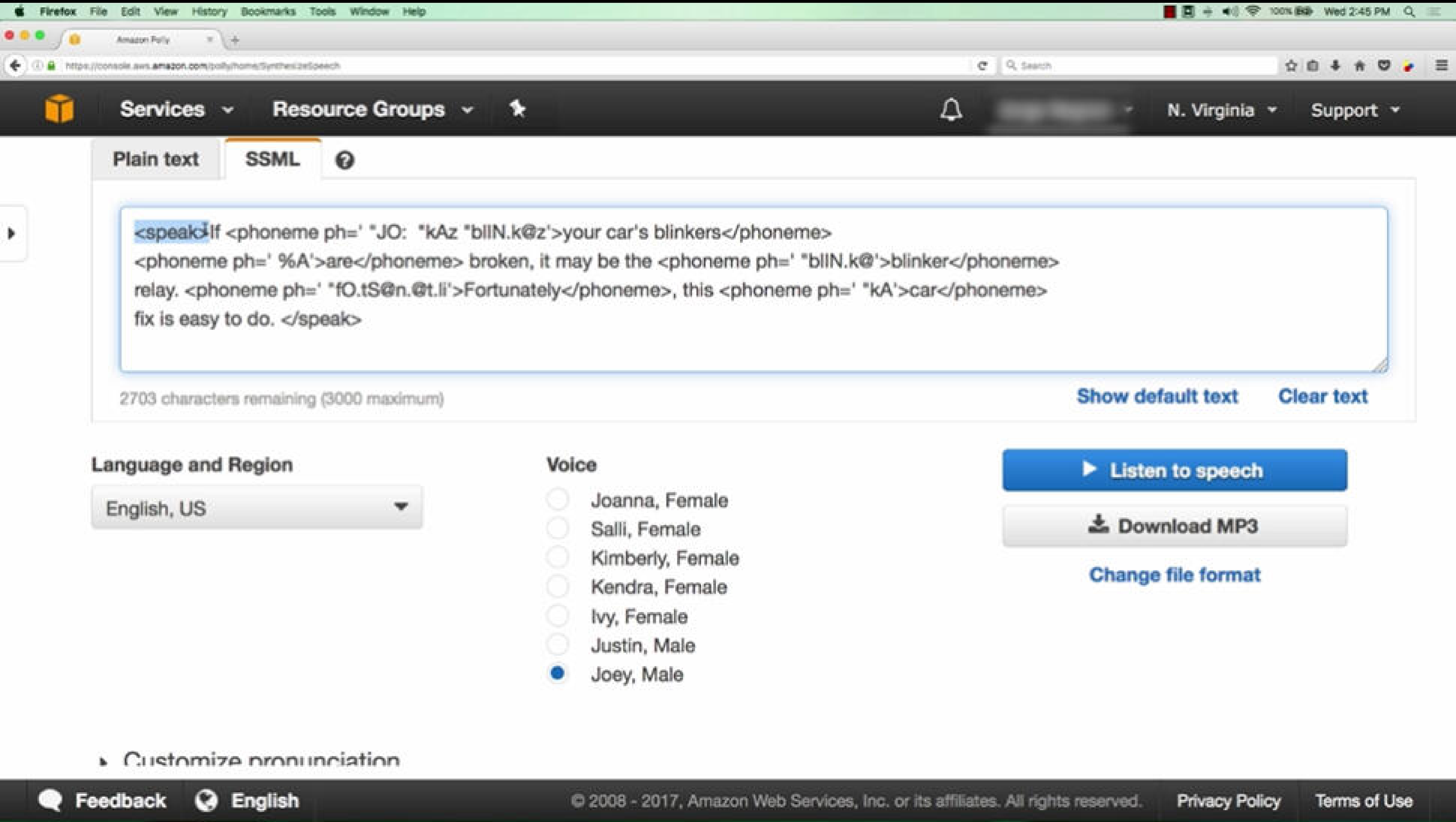
Sublime Text File Edit Selection Find View Goto Tools Project Window Help
Wed 2:44 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
AmazonPollyTextExamples
UNREGISTERED
Accurate Text Processing
1)
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
2)
St. Patrick's Church is on 333 St. Mary's St.
3) We live for the music. Live from Seattle, it's rock tuesday night!
Intelligible Tongue Twisters
1) Lesser leather never weathered wetter weather better
2) Of all the vids I've ever viewed, I've never viewed a vid as valued as Amazon Prime's vid
3) How can a clam cram in a clean cream can?
4) How much wood would a woodchuck chuck, if a woodchuck could chuck wood?,He would chuck, he would, as much as he could, and chuck as much wood as a woodchuck would, if a woodchuck could chuck wood.
5)
Betty Botter bought some butter,But she said the butter's bitter,If I put it in my batter, it will make my batter bitter,But a bit of better butter will make my batter better,So it was how better Betty Botter bought a bit of better butter.
6)
Peter Piper picked a peck of pickled peppers, A peck of pickled peppers Peter Piper picked, If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers, Peter Piper picked?
SSML Example (Boston Accent)
1A)
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1B)
<speak>If <phoneme ph='...'>your car's blinkers</phoneme>
<phoneme ph='...'>are</phoneme> broken, it may be the <phoneme ph='...'>blinker</phoneme>
relay. <phoneme ph='...'>Fortunately</phoneme>, this <phoneme ph='...'>car</phoneme>
fix is easy to do. </speak>
2)
<speak><prosody rate='+60%'>I'm at 500 and I want 550<prosody volume='x-loud'>550</prosody></prosody>
<prosody rate='+60%'>bid on 550 I'm at 500 would you go 550 550 for the gentleman in the corner</prosody>
<prosody rate="+90%">A big black bug bit a big black bear a big black bug bit a big black bear</prosody>
Do we get 600? <prosody rate="+90%">A big black bug bit a big black bear</prosody>
<prosody rate='+60%'>We got 600 for the whole herd</prosody>
<prosody rate='default' volume='x-loud'>Sold <prosody rate='+60%'> for 600.</prosody></prosody></speak>
2 lines, 107 characters selected
Tab Size: 4
Plain Text
my batter, it will make my batter bitter,But a bit of better butter will
tter butter.
Show default text
Clear text
Listen to speech
Download MP3
Change file format
Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

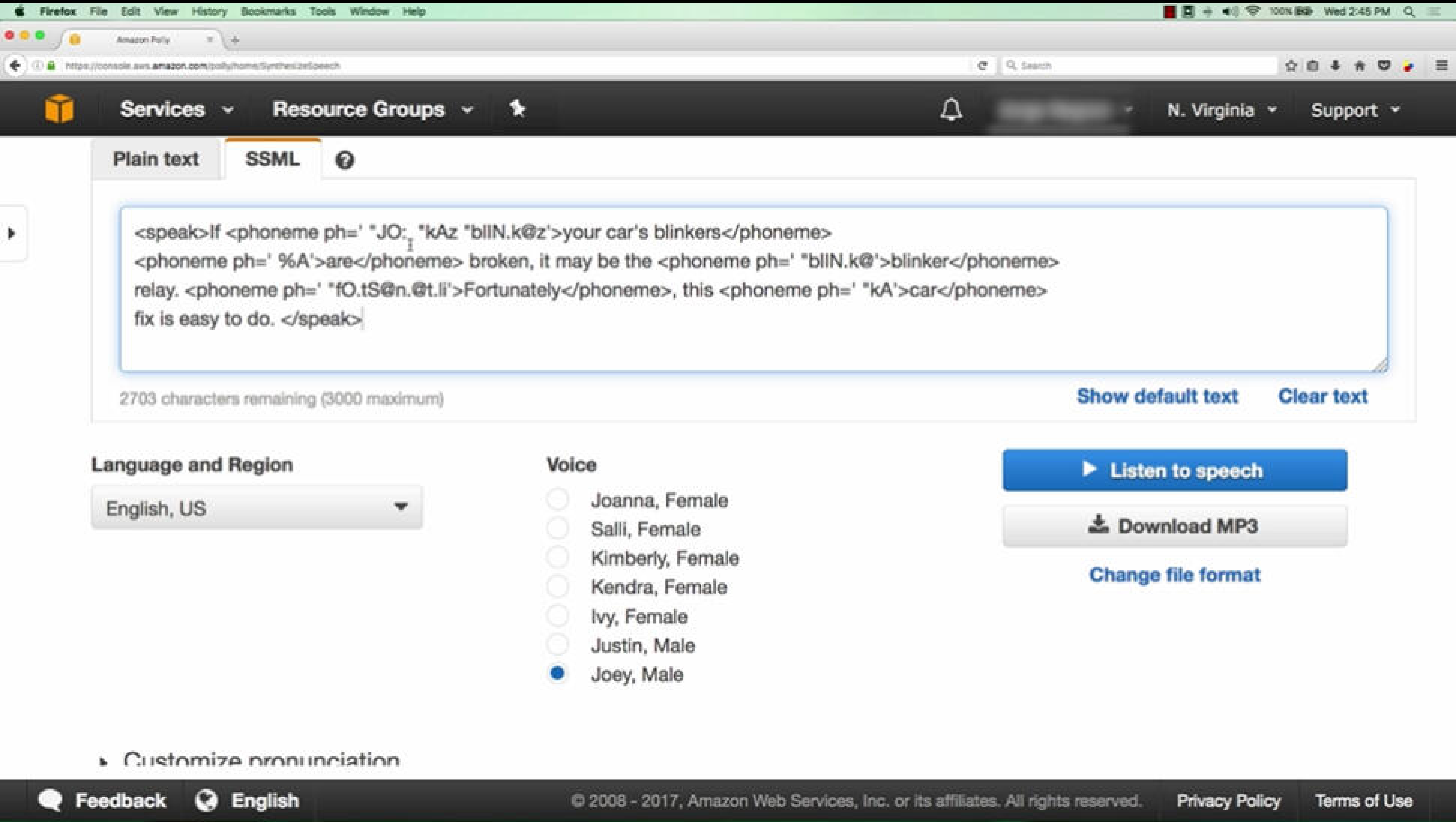

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:44 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
Plain text
SSML
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1393 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Stop the speech
Download MP3
Change file format
Customize pronunciation
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

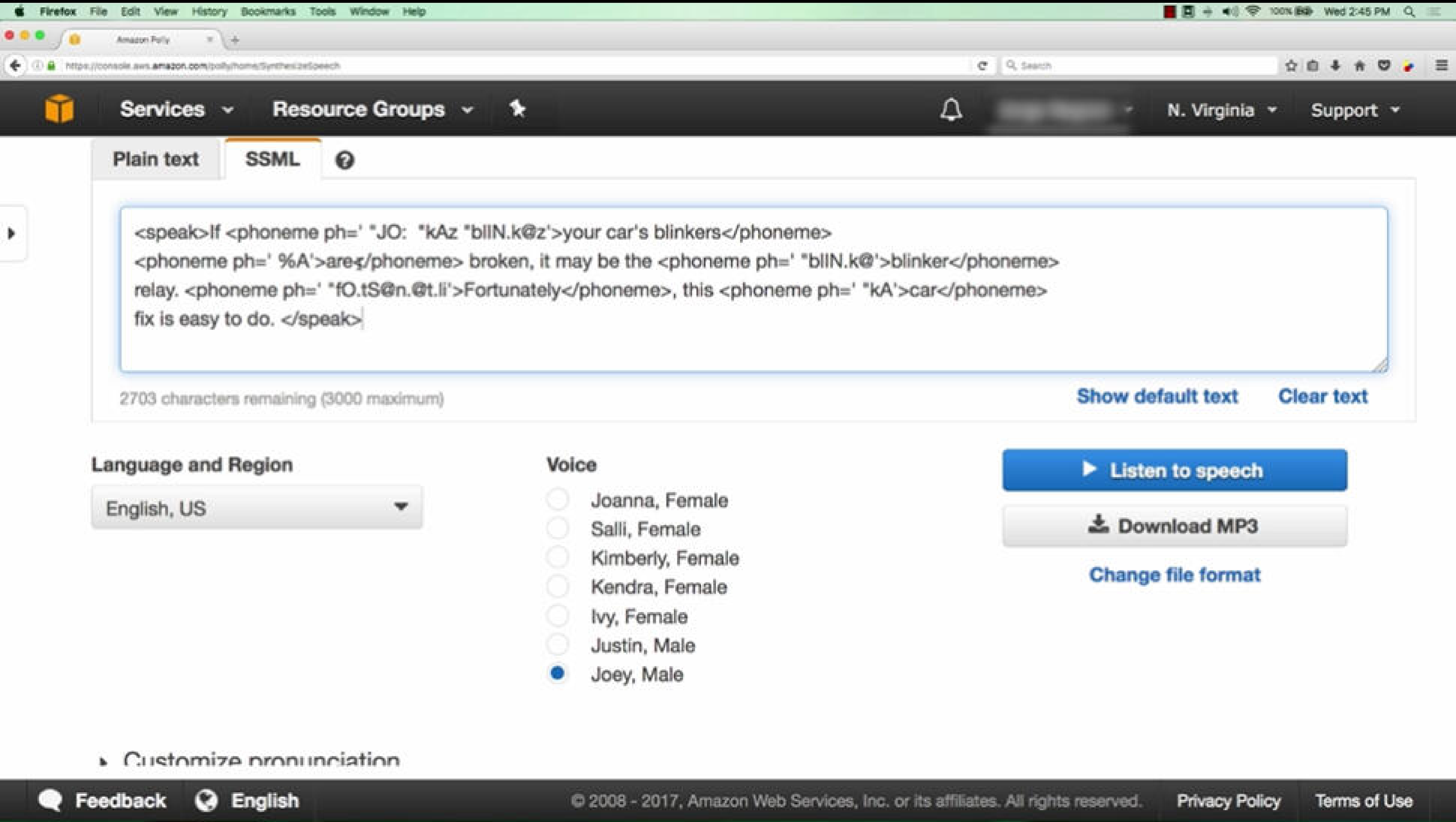


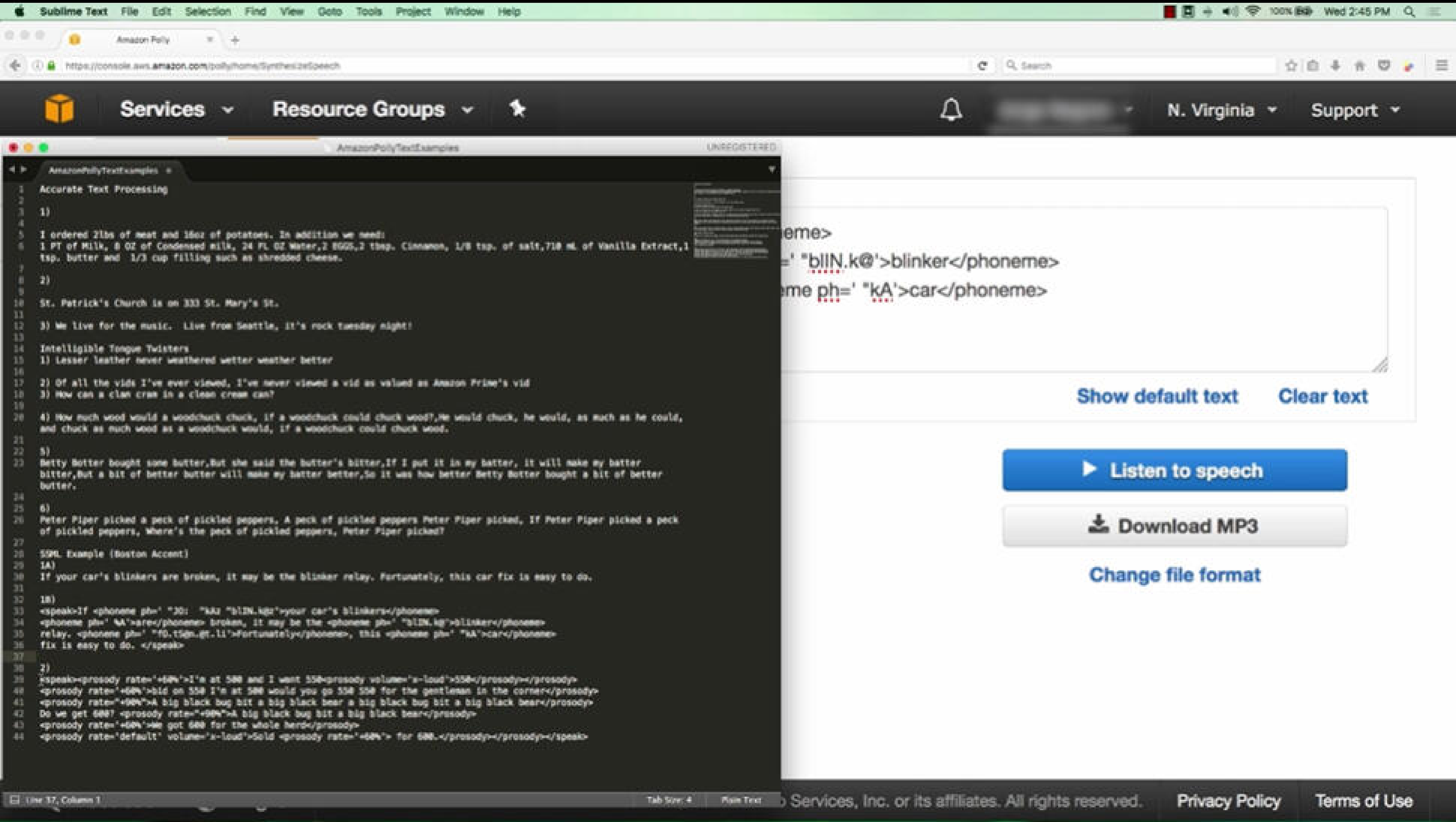
Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:44 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
Plain text
SSML
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1393 characters remaining (1500 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Stop the speech
Download MP3
Change file format
Customize pronunciation
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

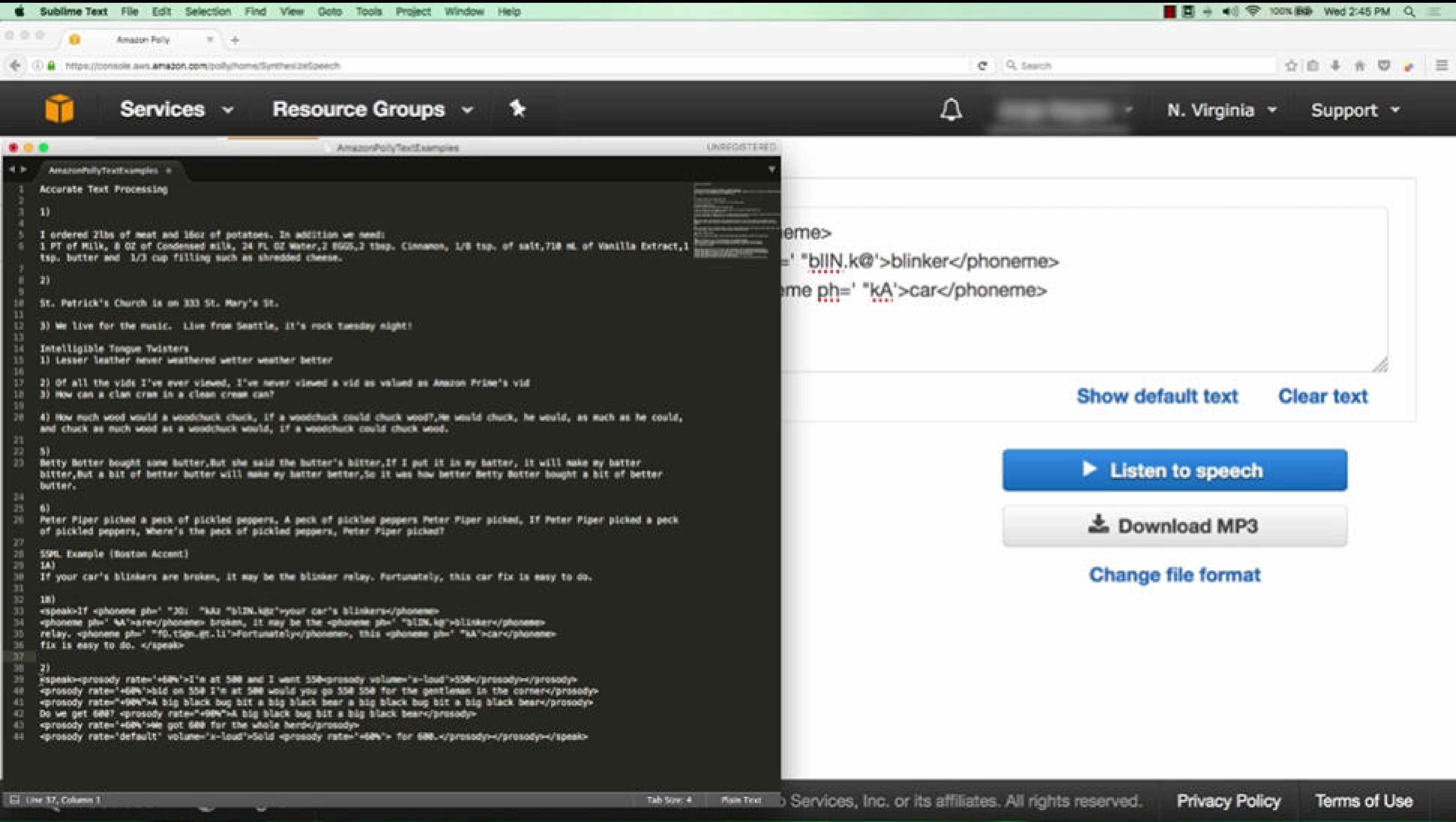

Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:44 PM
Amazon Polly
Search
Services
Resource Groups
N. Virginia
Support
Plain text
SSML
<speak>Hi! My name is Joey. I will read any text you type here.</speak>
2929 characters remaining (3000 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Customize pronunciation
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:45 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Services
Resource Groups
N. Virginia
Support
Plain text
SSML
<speak>If <phoneme ph=' "JO: "kAz "blIN.k@z'>your car's blinkers</phoneme>
<phoneme ph=' %A'>are</phoneme> broken, it may be the <phoneme ph=' "blIN.k@'>blinker</phoneme>
relay. <phoneme ph=' "fO.tS@n.@t.li'>Fortunately</phoneme>, this <phoneme ph=' "kA'>car</phoneme>
fix is easy to do. </speak>
2703 characters remaining (3000 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Customize pronunciation
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Services
Resource Groups
N. Virginia
Support
Plain text
SSML
<speak>If <phoneme ph=' "JO:, "kAz "blIN.k@z'>your car's blinkers</phoneme>
<phoneme ph=' %A'>are</phoneme> broken, it may be the <phoneme ph=' "blIN.k@'>blinker</phoneme>
relay. <phoneme ph=' "fO.tS@n.@t.li'>Fortunately</phoneme>, this <phoneme ph=' "kA'>car</phoneme>
fix is easy to do. </speak>
2703 characters remaining (3000 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Customize pronunciation
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Firefox File Edit View History Bookmarks Tools Window Help
Wed 2:45 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
Plain text
SSML
<speak>If <phoneme ph=' "JO: "kAz "blIN.k@z'>your car's blinkers</phoneme>
<phoneme ph=' %A'>are</phoneme> broken, it may be the <phoneme ph=' "blIN.k@'>blinker</phoneme>
relay. <phoneme ph=' "fO.tS@n.@t.li'>Fortunately</phoneme>, this <phoneme ph=' "kA'>car</phoneme>
fix is easy to do. </speak>
2703 characters remaining (3000 maximum)
Show default text
Clear text
Language and Region
English, US
Voice
Joanna, Female
Salli, Female
Kimberly, Female
Kendra, Female
Ivy, Female
Justin, Male
Joey, Male
Listen to speech
Download MP3
Change file format
Customize pronunciation
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
Wed 2:45 PM
Amazon Polly
https://console.aws.amazon.com/polly/home/SynthesizeSpeech
Search
Services
Resource Groups
N. Virginia
Support
AmazonPollyTextExamples
UNREGISTERED
Accurate Text Processing
1)
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
2)
St. Patrick's Church is on 333 St. Mary's St.
3) We live for the music. Live from Seattle, it's rock tuesday night!
Intelligible Tongue Twisters
1) Lesser leather never weathered wetter weather better
2) Of all the vids I've ever viewed, I've never viewed a vid as valued as Amazon Prime's vid
3) How can a clam cram in a clean cream can?
4) How much wood would a woodchuck chuck, if a woodchuck could chuck wood?,He would chuck, he would, as much as he could, and chuck as much wood as a woodchuck would, if a woodchuck could chuck wood.
5)
Betty Botter bought some butter,But she said the butter's bitter,If I put it in my batter, it will make my batter bitter,But a bit of better butter will make my batter better,So it was how better Betty Botter bought a bit of better butter.
6)
Peter Piper picked a peck of pickled peppers, A peck of pickled peppers Peter Piper picked, If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers, Peter Piper picked?
SSML Example (Boston Accent)
1A)
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1B)
<speak>If <phoneme ph=' "JO: "kAz "blIN.k@z'>your car's blinkers</phoneme>
<phoneme ph=' %A'>are</phoneme> broken, it may be the <phoneme ph=' "blIN.k@'>blinker</phoneme>
relay. <phoneme ph=' "fO.tS@n.@t.li'>Fortunately</phoneme>, this <phoneme ph=' "kA'>car</phoneme>
fix is easy to do. </speak>
2)
<speak><prosody rate='+60%'>I'm at 500 and I want 550<prosody volume='x-loud'>550</prosody></prosody>
<prosody rate='+60%'>bid on 550 I'm at 500 would you go 550 550 for the gentleman in the corner</prosody>
<prosody rate="+90%">A big black bug bit a big black bear a big black bug bit a big black bear</prosody>
Do we get 600? <prosody rate="+90%">A big black bug bit a big black bear</prosody>
<prosody rate='+60%'>We got 600 for the whole herd</prosody>
<prosody rate='default' volume='x-loud'>Sold <prosody rate='+60%'> for 600.</prosody></prosody></speak>
Line 37, Column 1
Tab Size: 4
Plain Text
eme>
=' "blIN.k@'>blinker</phoneme>
eme ph=' "kA'>car</phoneme>
Show default text
Clear text
Listen to speech
Download MP3
Change file format
Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
Wed 2:45 PM
Amazon Polly
Services
Resource Groups
N. Virginia
Support
AmazonPollyTextExamples
UNREGISTERED
Accurate Text Processing
1)
I ordered 2lbs of meat and 16oz of potatoes. In addition we need:
1 PT of Milk, 8 OZ of Condensed milk, 24 FL OZ Water,2 EGGS,2 tbsp. Cinnamon, 1/8 tsp. of salt,710 mL of Vanilla Extract,1 tsp. butter and 1/3 cup filling such as shredded cheese.
2)
St. Patrick's Church is on 333 St. Mary's St.
3) We live for the music. Live from Seattle, it's rock tuesday night!
Intelligible Tongue Twisters
1) Lesser leather never weathered wetter weather better
2) Of all the vids I've ever viewed, I've never viewed a vid as valued as Amazon Prime's vid
3) How can a clam cram in a clean cream can?
4) How much wood would a woodchuck chuck, if a woodchuck could chuck wood?,He would chuck, he would, as much as he could, and chuck as much wood as a woodchuck would, if a woodchuck could chuck wood.
5)
Betty Botter bought some butter,But she said the butter's bitter,If I put it in my batter, it will make my batter bitter,But a bit of better butter will make my batter better,So it was how better Betty Botter bought a bit of better butter.
6)
Peter Piper picked a peck of pickled peppers, A peck of pickled peppers Peter Piper picked, If Peter Piper picked a peck of pickled peppers, Where's the peck of pickled peppers, Peter Piper picked?
SSML Example (Boston Accent)
1A)
If your car's blinkers are broken, it may be the blinker relay. Fortunately, this car fix is easy to do.
1B)
fix is easy to do. </speak>
2)
Do we get 600?
eme>
=' "blIN.k@'>blinker</phoneme>
me ph=' "kA'>car</phoneme>
Show default text
Clear text
Listen to speech
Download MP3
Change file format
Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Line 37, Column 1
Tab Size: 4
Plain Text

Firefox File Edit View History Bookmarks Tools Window Help

Amazon Polly

https://console.aws.amazon.com/polly/home/SynthesizeSpeech

Services Resource Groups N. Virginia Support

Text-to-Speech

Listen, customize, and download speech. Integrate when you're ready.

Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.

Plain text **SSML**

```
<prosody rate='+60%'>bid on 550 I'm at 500 would you go 550 550 for the gentleman in the corner</prosody>
<prosody rate="+90%">A big black bug bit a big black bear a big black bug bit a big black bear</prosody>
Do we get 600? <prosody rate="+90%">A big black bug bit a big black bear</prosody>
<prosody rate='+60%'>We got 600 for the whole herd</prosody>
<prosody rate='default' volume='x-loud'>Sold <prosody rate='+60%'> for 600.</prosody></prosody></speak>
```

2439 characters remaining (3000 maximum)

Show default text Clear text

Language and Region

English, US

Voice

Joanna, Female

Salli, Female

Listen to speech

Download MP3

Feedback English

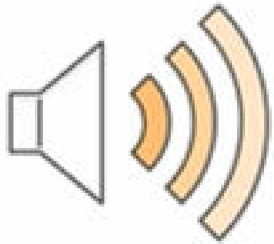
Privacy Policy Terms of Use

Thanks for watching!

aws training and certification

aws training and certification

What is Amazon Lex?

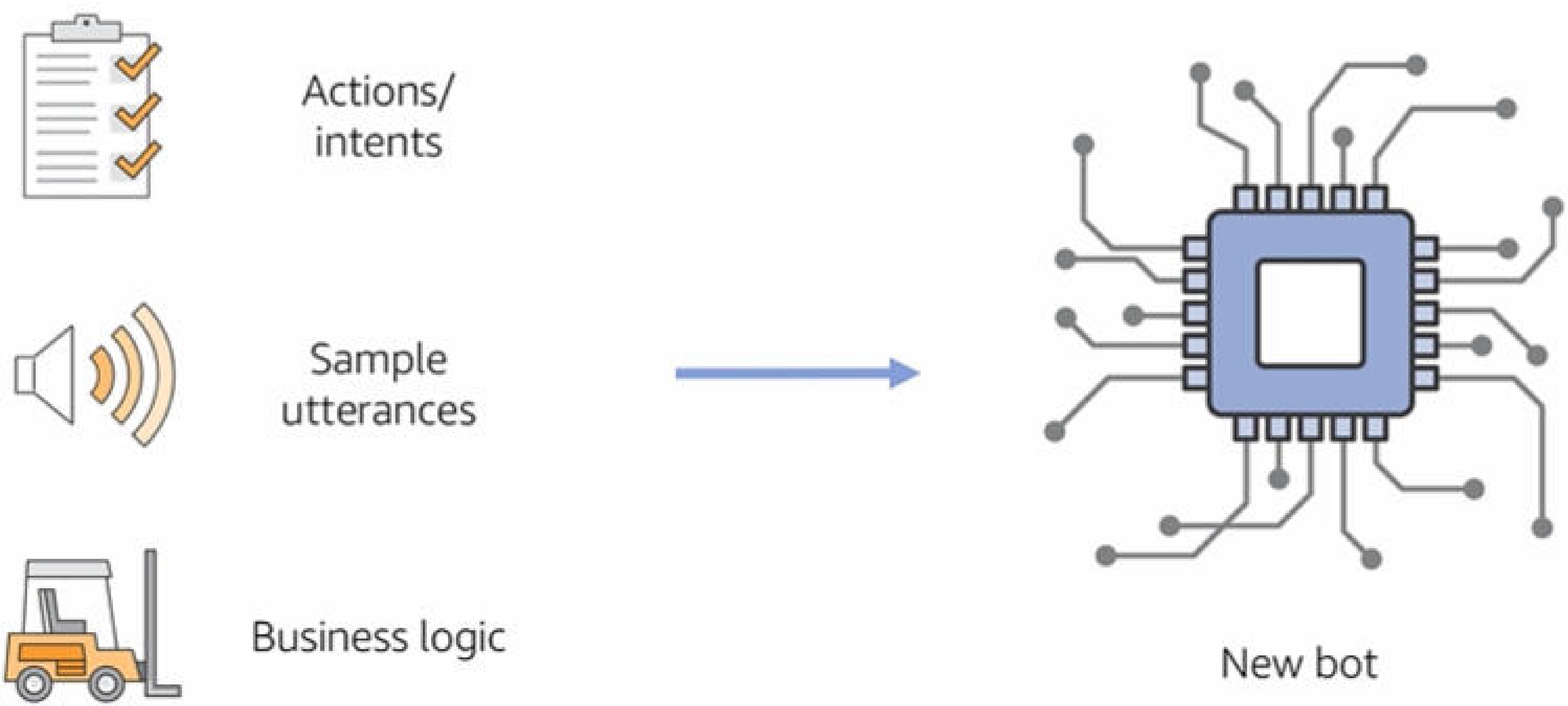


Conversational interfaces using voice and text

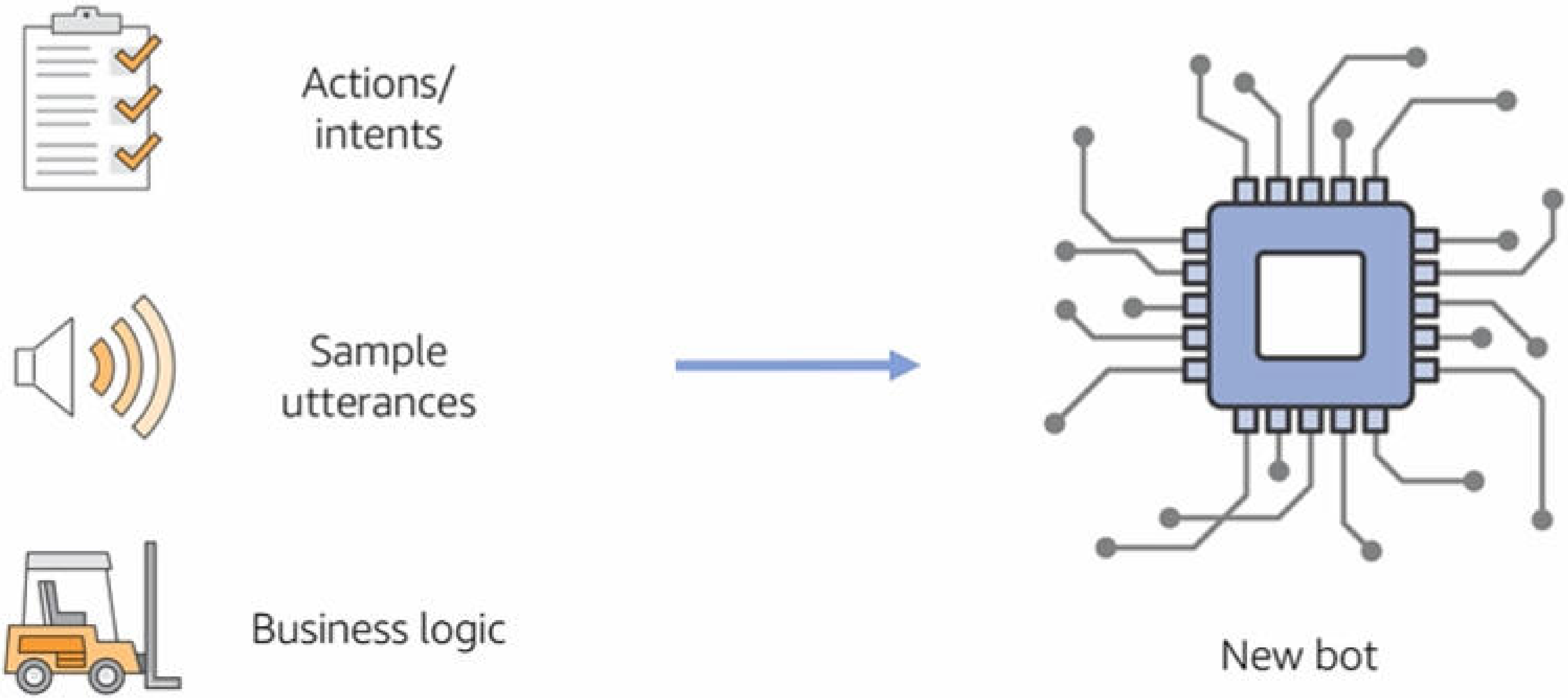
Automatic speech recognition (ASR) and Natural Language Understanding (NLU)

Build applications with highly engaging user experiences

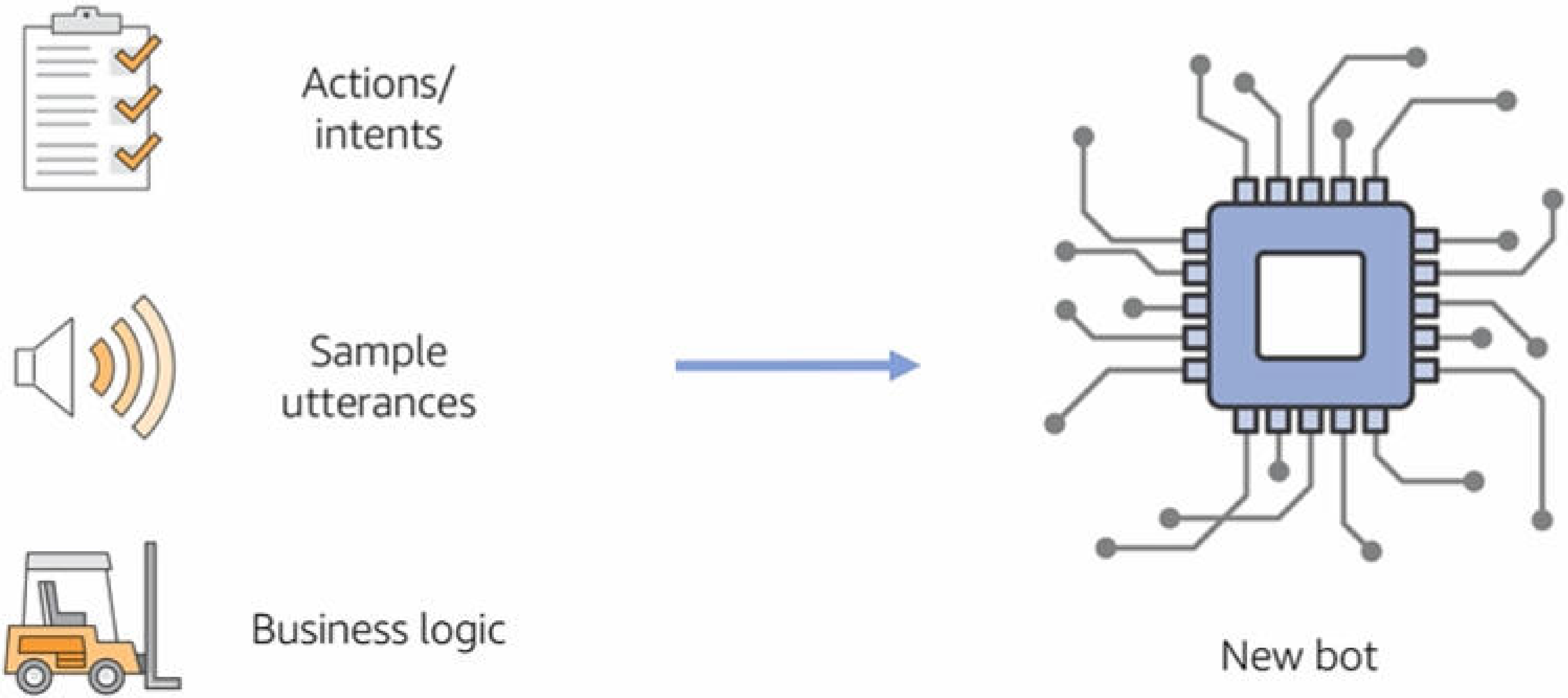
What Goes Into a Bot?



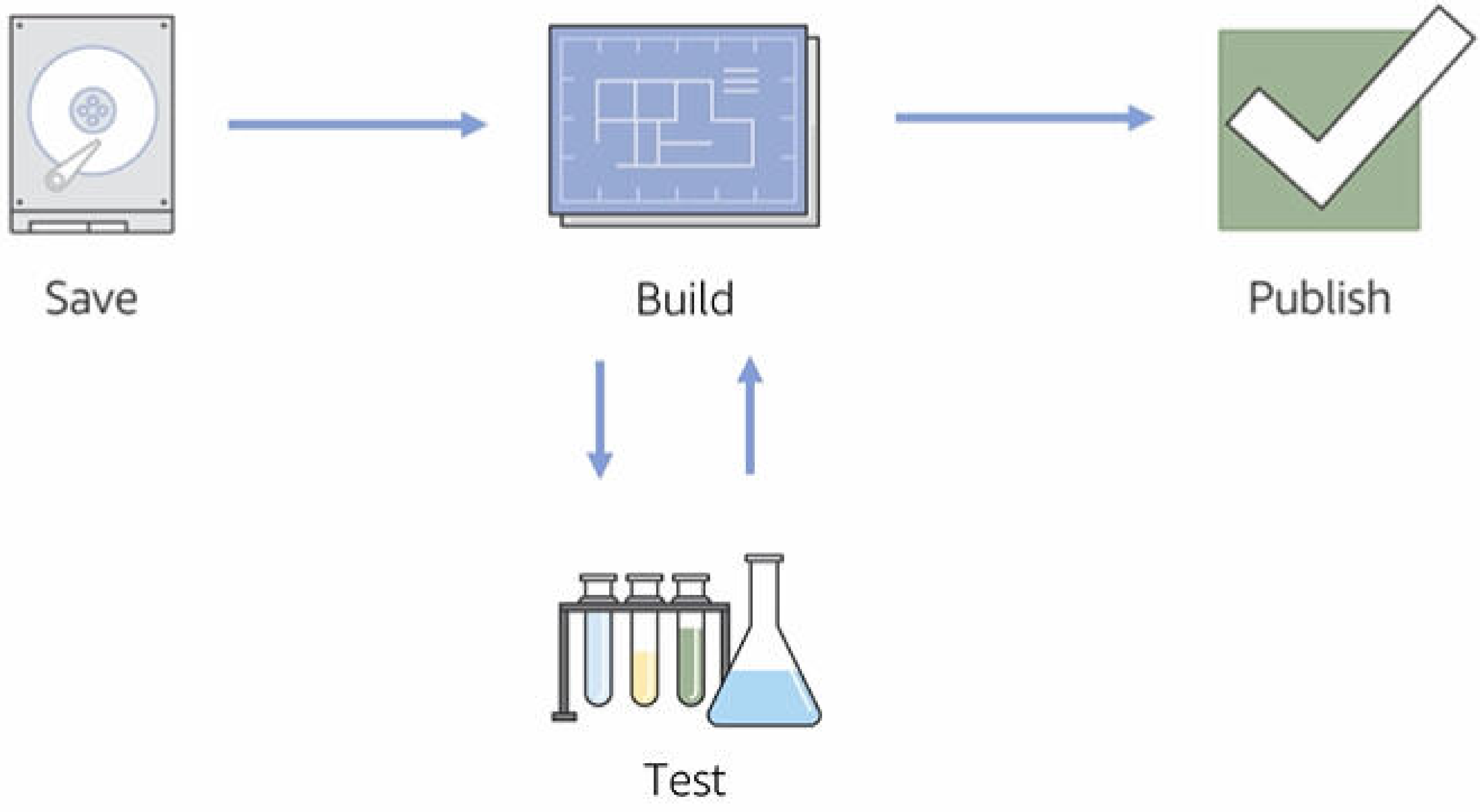
What Goes Into a Bot?



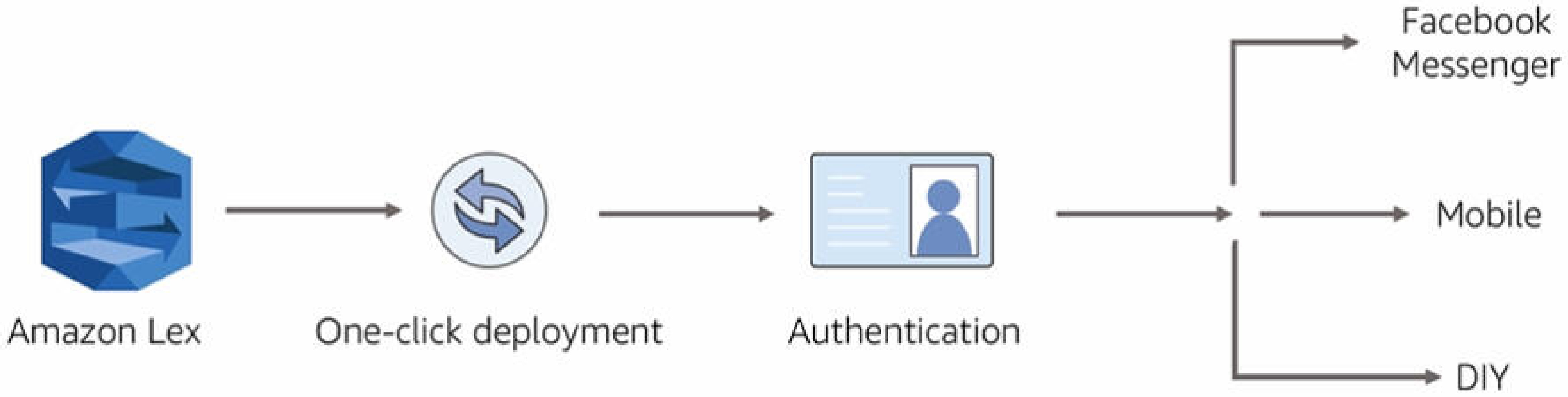
What Goes Into a Bot?



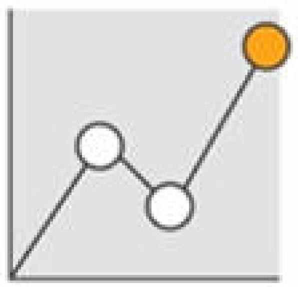
Amazon Lex Workflow



Deployment to Chat Services



Monitoring



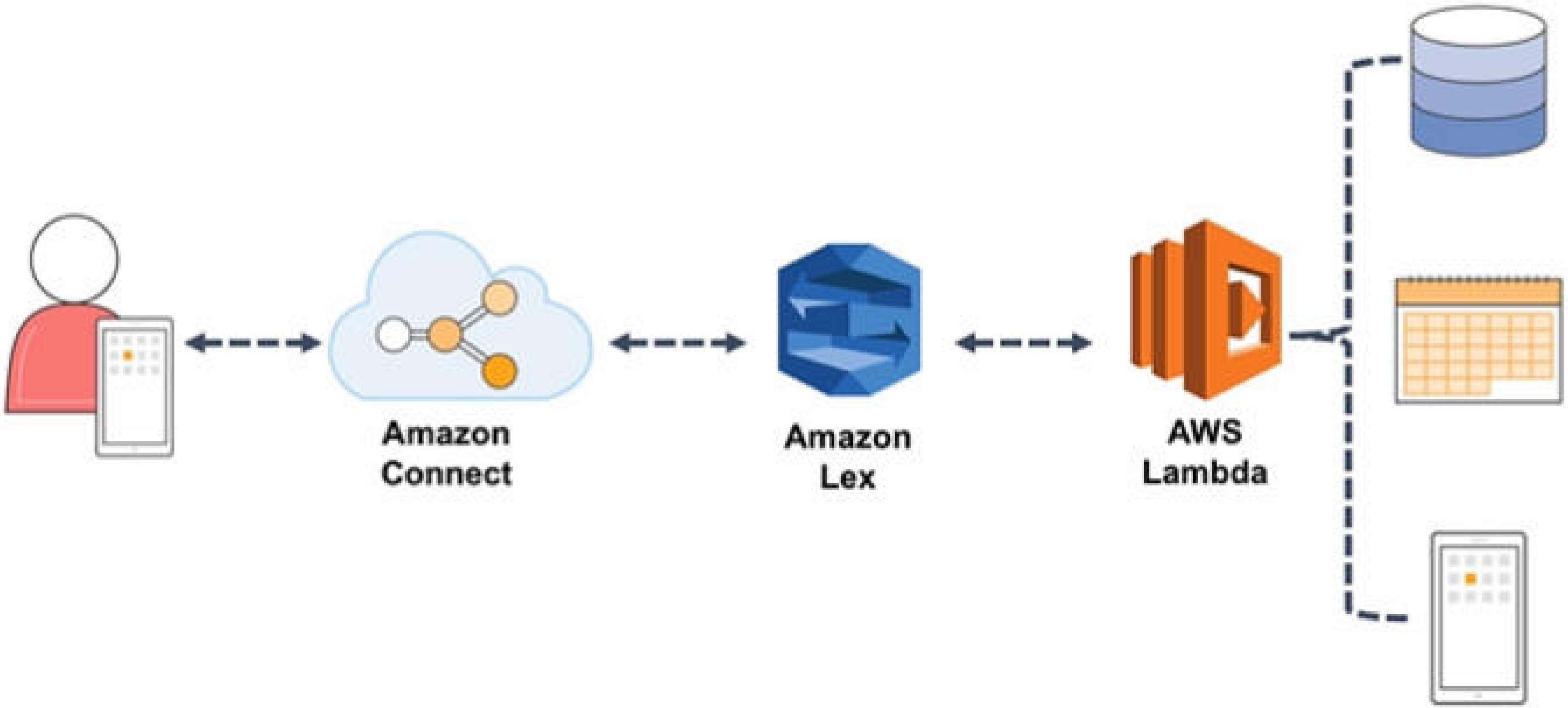
Number of requests

Latency

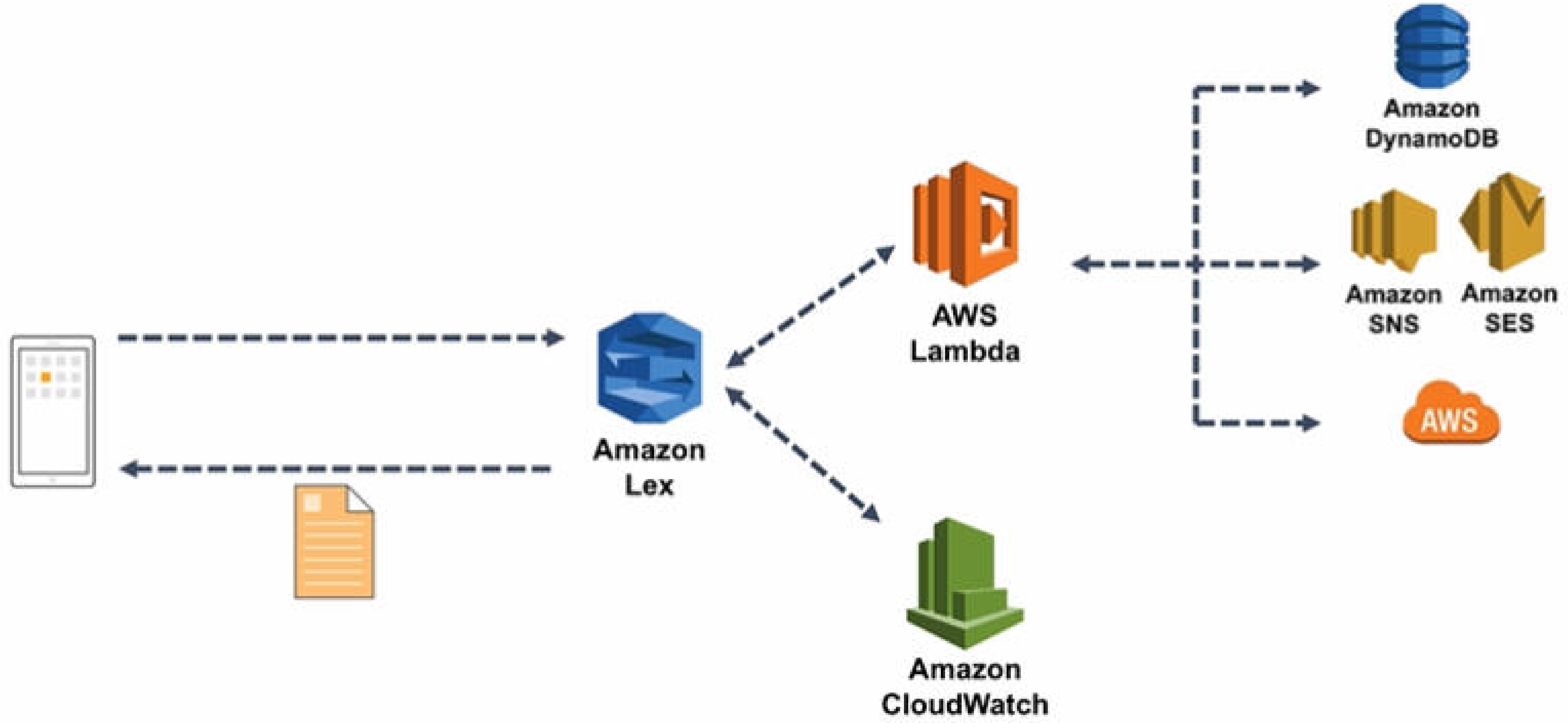
Errors

Use Cases

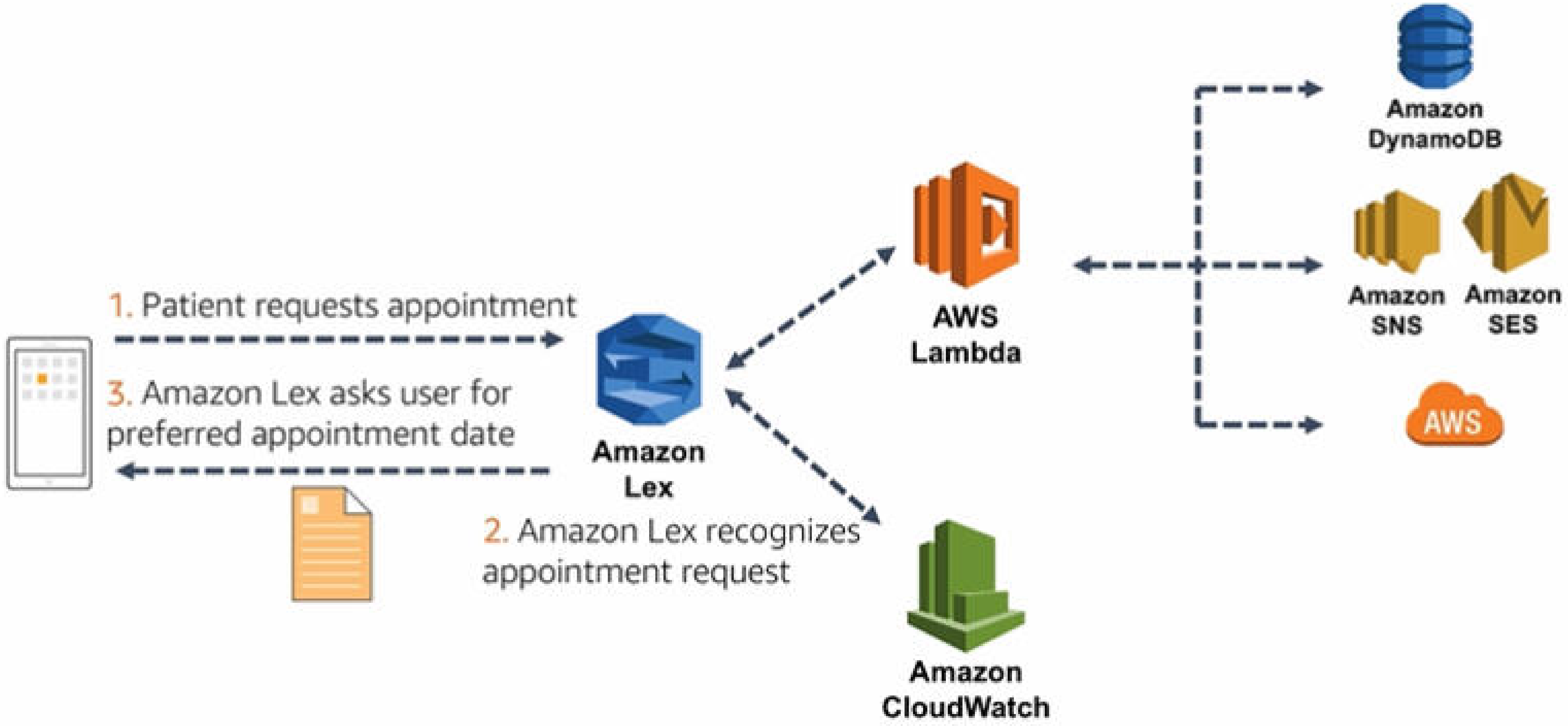
Use Case: Call Center Bots



Use Case: Informational Bots



Use Case: Informational Bots



Time-series forecasting

Science of predicting future points in a time series based on historical data.

Product demand

Workforce demand

Financial metrics

Inventory control

Forecasting accuracy

aws training and certification

- Historical data
- Holidays and vacations
- Economic indicators
- Weather conditions

The challenge:

aws training and certification

Traditional forecasting methods struggle with forecast accuracy.

- Spiky or intermittent data
- Additional variables
- External factors
- No history available

Introducing Amazon Forecast

aws training and certification

Amazon Forecast

- Fully-managed accuracy forecasting solution
- Uses deep learning models
- From over 10 years of machine learning experience
- Comes with multiple deep learning algorithms

Benefits

Accurate

Improves accuracy by up to 50 percent compared to traditional models

Benefits

Accurate
Improves accuracy by up to 50 percent compared to traditional models

Automated
Automates feature engineering, algorithm selection, and model tuning

Benefits

aws training and certification

Accurate
Improves accuracy by up to 50 percent compared to traditional models

Automated
Automates feature engineering, algorithm selection, and model tuning

Private and Secure
Creates a custom model for each customer without any data sharing

Customizable
Provides the ability to create your own algorithms or customize existing ones

How does it work?

Import data from Amazon S3 buckets

How does it work?

aws training and certification

Import data from Amazon S3 buckets

Manually or automatically select algorithms

Evaluate accuracy metrics and deploy to production

How does it work?

aws training and certification

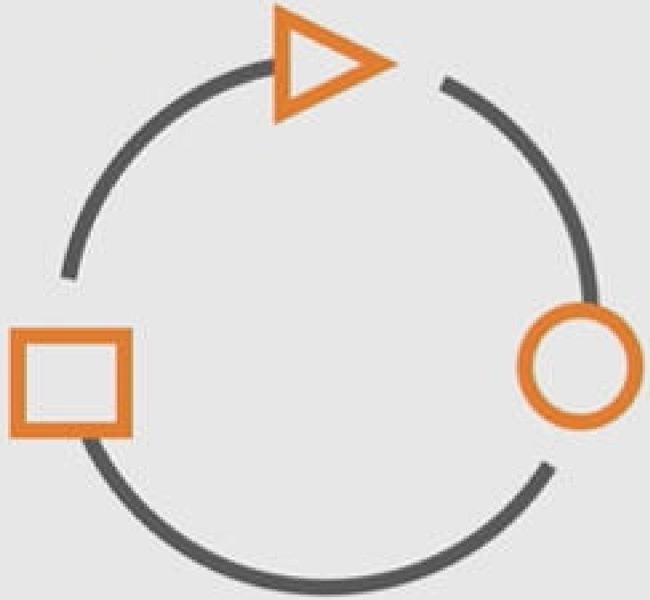
Import data from Amazon S3 buckets

Manually or automatically select algorithms

Evaluate accuracy metrics and deploy to production

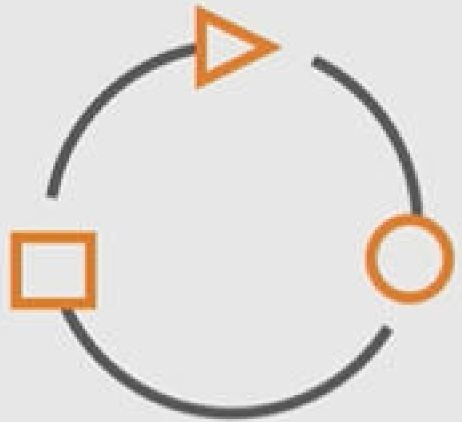
Visualize or export forecasts

Simplifying your data pipeline



- Automatically ingests new data when it's available and uses it for re-training
- Ability to schedule periodic model re-training and re-forecasting
- Allows for values and features to be overridden

Simplifying your data pipeline



- Automatically ingests new data when it's available and uses it for re-training
- Ability to schedule periodic model re-training and re-forecasting
- Allows for values and features to be overridden

Machine learning automation

aws training and certification

Machine learning automation

aws training and certification

Amazon SageMaker and Amazon Forecast

aws training and certification

Amazon SageMaker

- Data needs to be converted for model training.
- Algorithm needs to be manually selected.
- Parameters require manual tuning for better accuracy.

Amazon Forecast

- No machine learning expertise needed
- Algorithm can be automatically selected
- Parameters are automatically tuned for maximum accuracy

Use cases

aws training and certification

Amazon Forecast works best with many related time-series and at least 1,000 data points, but it can still work for smaller data-sets.

Use cases

aws training and certification

Amazon Forecast works best with many related time-series and at least 1,000 data points, but it can still work for smaller data-sets.

Amazon Forecast is ideal for:

- Implementing tighter inventory control
- Web traffic forecasting
- Profit, sales, and expenses forecasting
- Resource and equipment forecasting

Use cases

Amazon Forecast works best with many related time-series and at least 1,000 data points, but it can still work for smaller data-sets.

Amazon Forecast is ideal for:

- Implementing tighter inventory control
- Web traffic forecasting
- Profit, sales, and expenses forecasting
- Resource and equipment forecasting

Use cases

aws training and certification

Amazon Forecast works best with many related time-series and at least 1,000 data points, but it can still work for smaller data-sets.

Amazon Forecast is ideal for:

- Implementing tighter inventory control
- Web traffic forecasting
- Profit, sales, and expenses forecasting
- Resource and equipment forecasting

Use cases

Amazon Forecast works best with many related time-series and at least 1,000 data points, but it can still work for smaller data-sets.

Amazon Forecast is ideal for:

- Implementing tighter inventory control
- Web traffic forecasting
- Profit, sales, and expenses forecasting
- Resource and equipment forecasting

Use cases

aws training and certification

Amazon Forecast works best with many related time-series and at least 1,000 data points, but it can still work for smaller data-sets.

Amazon Forecast is ideal for:

- Implementing tighter inventory control
- Web traffic forecasting
- Profit, sales, and expenses forecasting
- Resource and equipment forecasting

Key Takeaways

- Amazon Forecast leverages over a decade of experience in forecasting at Amazon.
- Comes with pre-built deep learning and traditional algorithms and automates the entire cleaning and model training process.
- Amazon Forecast incorporates external data and can generate forecasts for new time series where historical data is limited.

Key Takeaways

aws training and certification

- Amazon Forecast leverages over a decade of experience in forecasting at Amazon.
- Comes with pre-built deep learning and traditional algorithms and automates the entire cleaning and model training process.
- Amazon Forecast incorporates external data and can generate forecasts for new time series where historical data is limited.

Key Takeaways

- Amazon Forecast leverages over a decade of experience in forecasting at Amazon.
- Comes with pre-built deep learning and traditional algorithms and automates the entire cleaning and model training process.
- Amazon Forecast incorporates external data and can generate forecasts for new time series where historical data is limited.

Key Takeaways

aws training and certification

- Amazon Forecast leverages over a decade of experience in forecasting at Amazon.
- Comes with pre-built deep learning and traditional algorithms and automates the entire cleaning and model training process.
- Amazon Forecast incorporates external data and can generate forecasts for new time series where historical data is limited.

Key Takeaways

- Amazon Forecast leverages over a decade of experience in forecasting at Amazon.
- Comes with pre-built deep learning and traditional algorithms and automates the entire cleaning and model training process.
- Amazon Forecast incorporates external data and can generate forecasts for new time series where historical data is limited.
- Visualizations may be accessed via the console, retrieved via API, or exported in CSV format.

In this course

- Service introduction
- Overview
- Demonstration
- Key takeaways

ML

aws training and certification

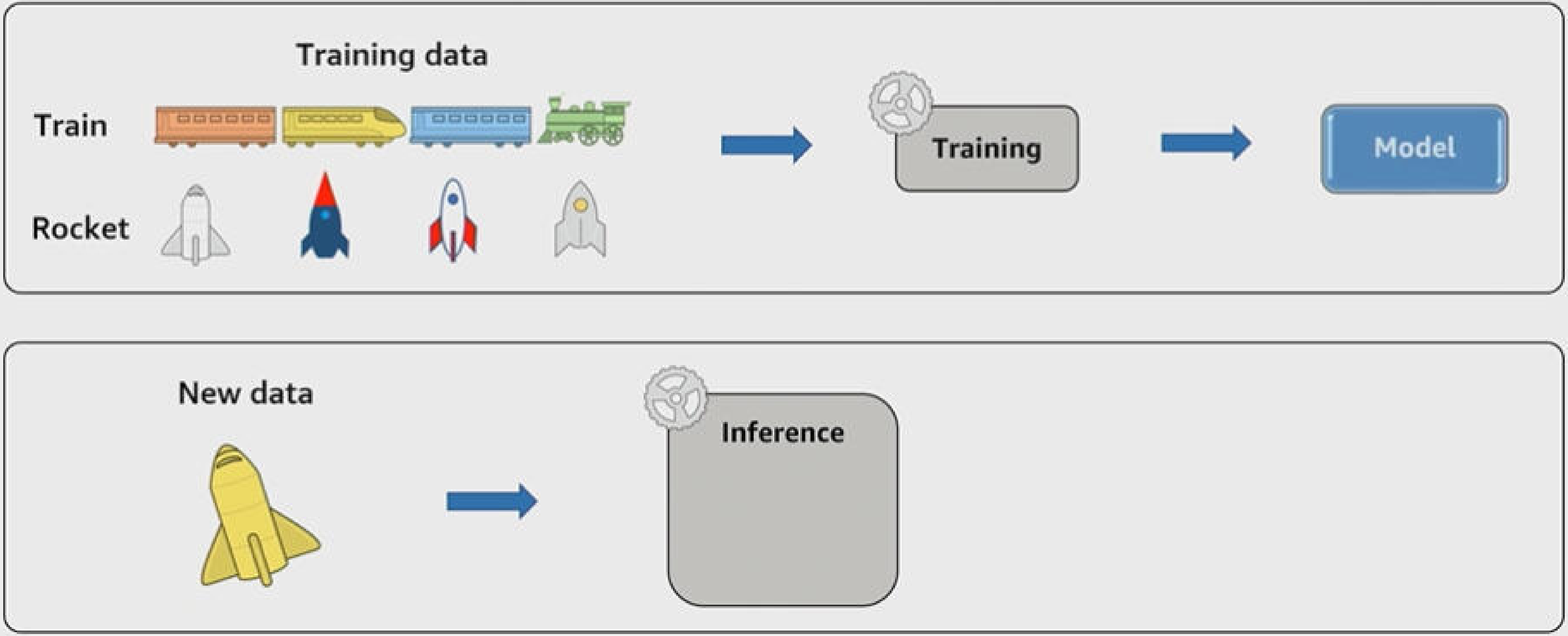
Methods and programs that:

Extract | Predict | Summarize | Optimize | Adapt

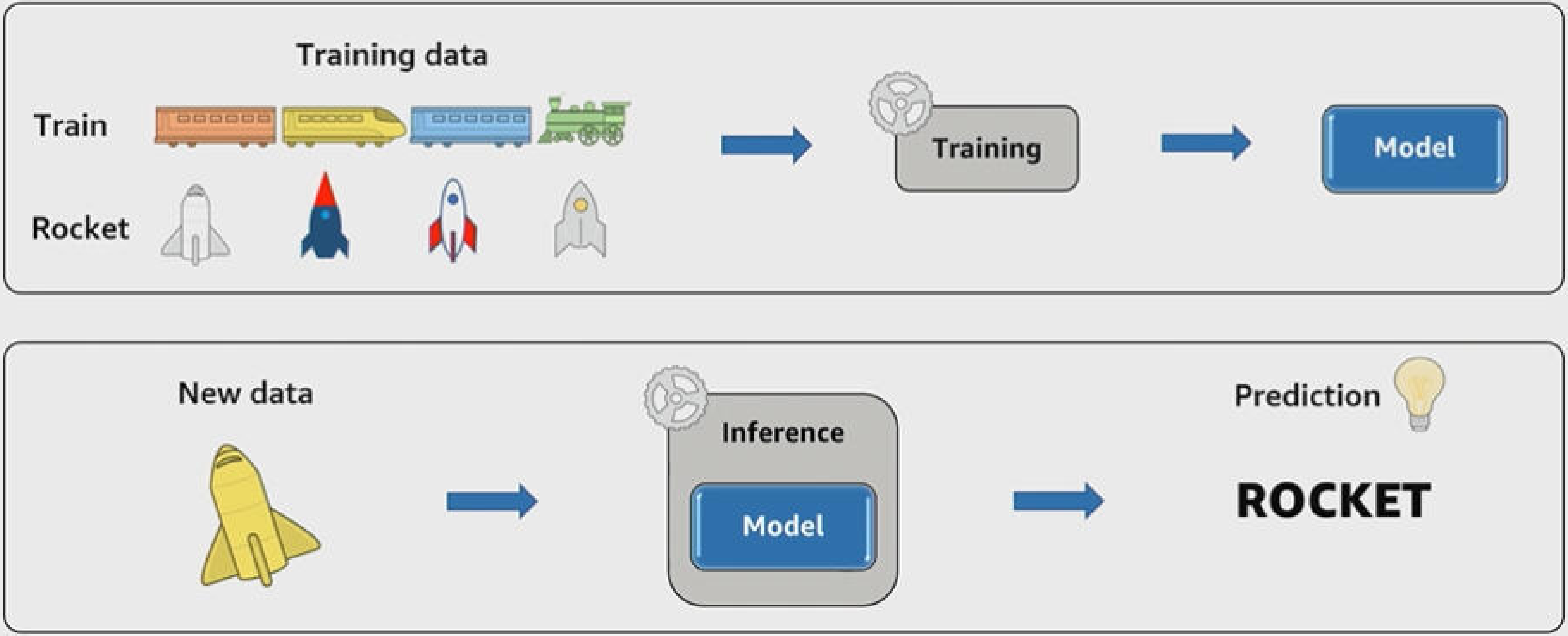
Your data + machine learning = smart applications

10:27

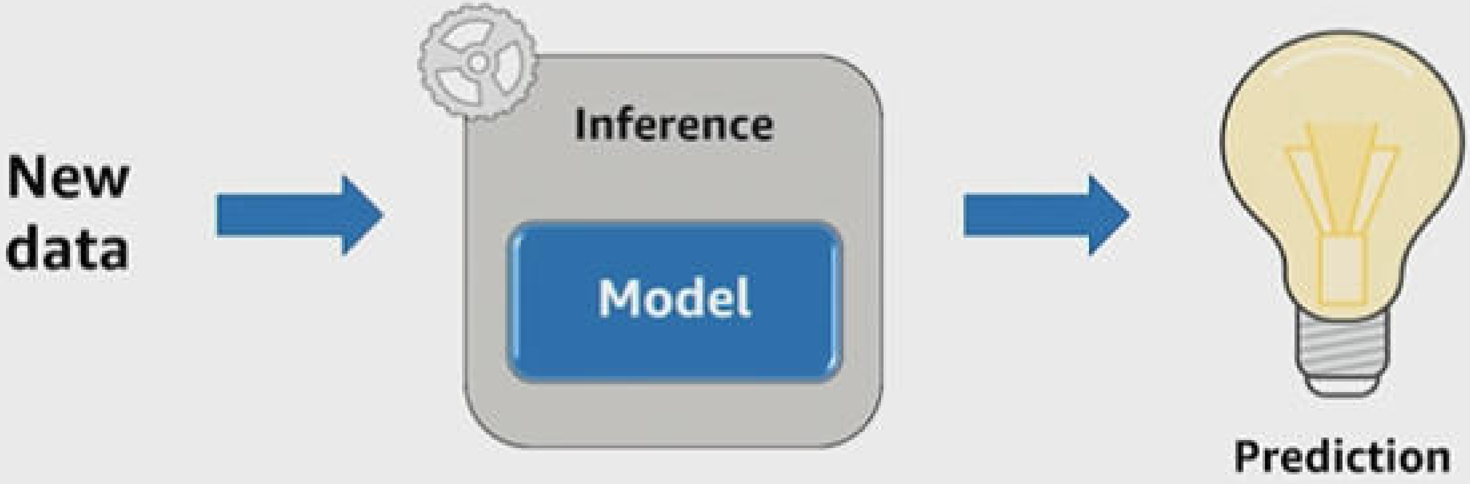
Training and inference



Training and inference



Inference



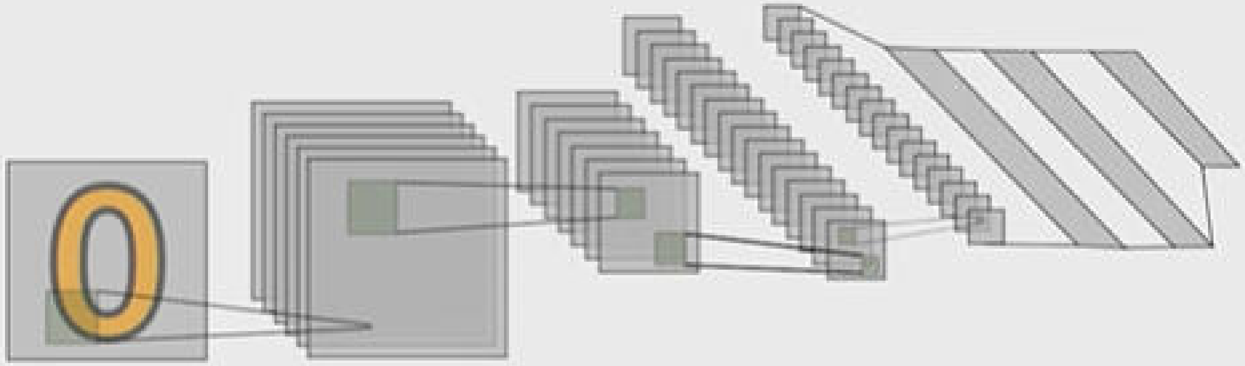
ML Challenge: Lower the Cost

aws training and certification

ML Challenge: Lower the Cost

aws training and certification

Convolutional neural network



ML Challenge: Lower the Cost

Model

Model

Model

10:27

ML Challenge: Lower the Cost

aws training and certification

– GPU
+ CPU
+ Memory

Model

Language understanding

Model

Model

Computer vision

ML Challenge: Lower the Cost

– GPU
+ CPU
+ Memory

Model

Language understanding

Model

+ GPU
– CPU
– Memory

Model

Computer vision

ML Challenge: Lower the Cost

– GPU
+ CPU
+ Memory

Model — Language understanding

Model

+ GPU
– CPU
– Memory

Model — Computer vision

ML Challenge: Lower the Cost

aws training and certification

– GPU
+ CPU
+ Memory

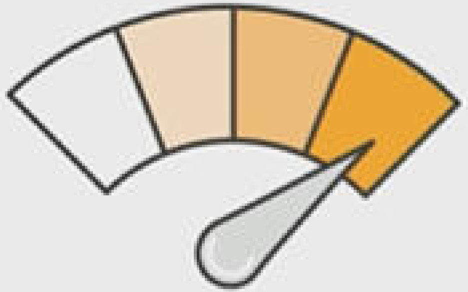
Model — Language understanding

Model

+ GPU
– CPU
– Memory

Model — Computer vision

Amazon EI



GPU-powered deep learning inference acceleration

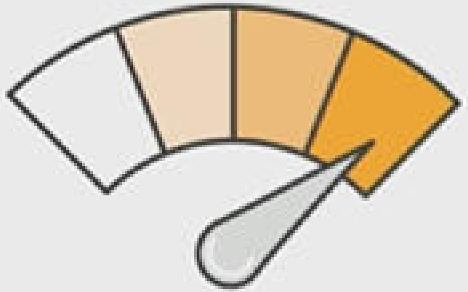
Fraction of the cost of standalone GPU instances

Amazon EI

GPU-powered deep learning inference acceleration

Fraction of the cost of standalone GPU instances

Amazon EI



GPU-powered deep learning inference acceleration

Fraction of the cost of standalone GPU instances

Amazon EI

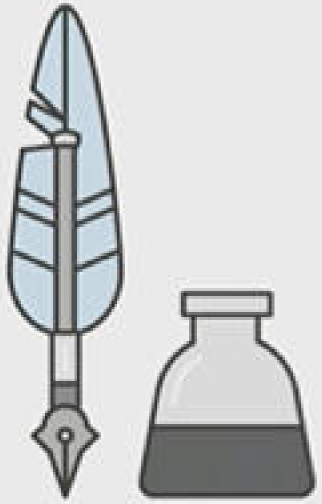
aws training and certification

Amazon EC2

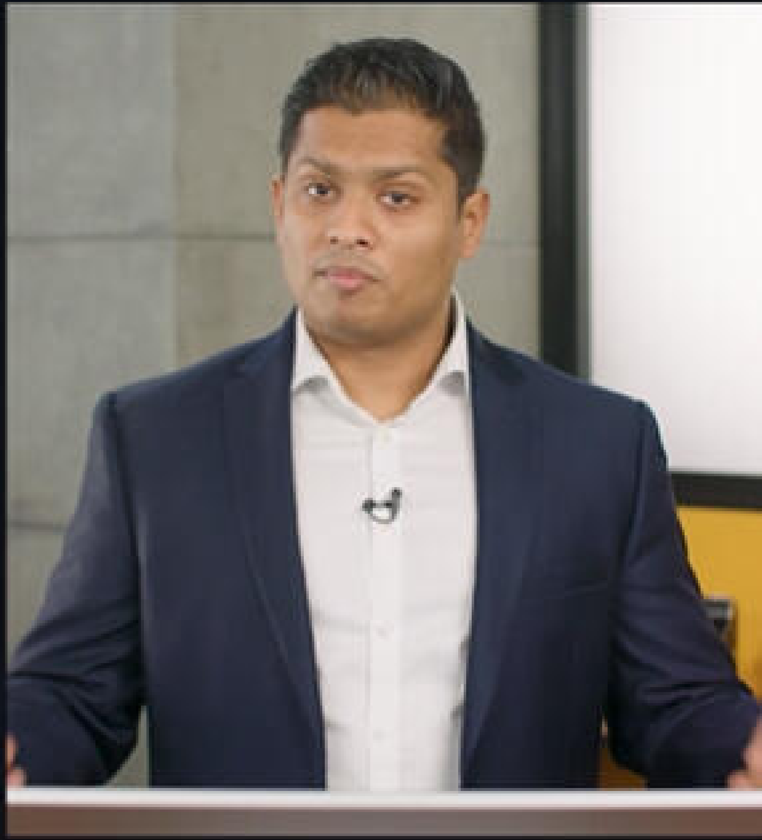
Amazon SageMaker

Overview

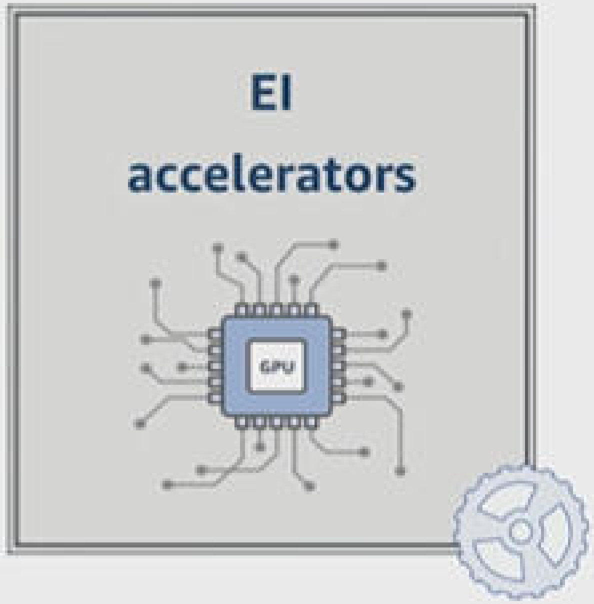
What is Amazon EI?

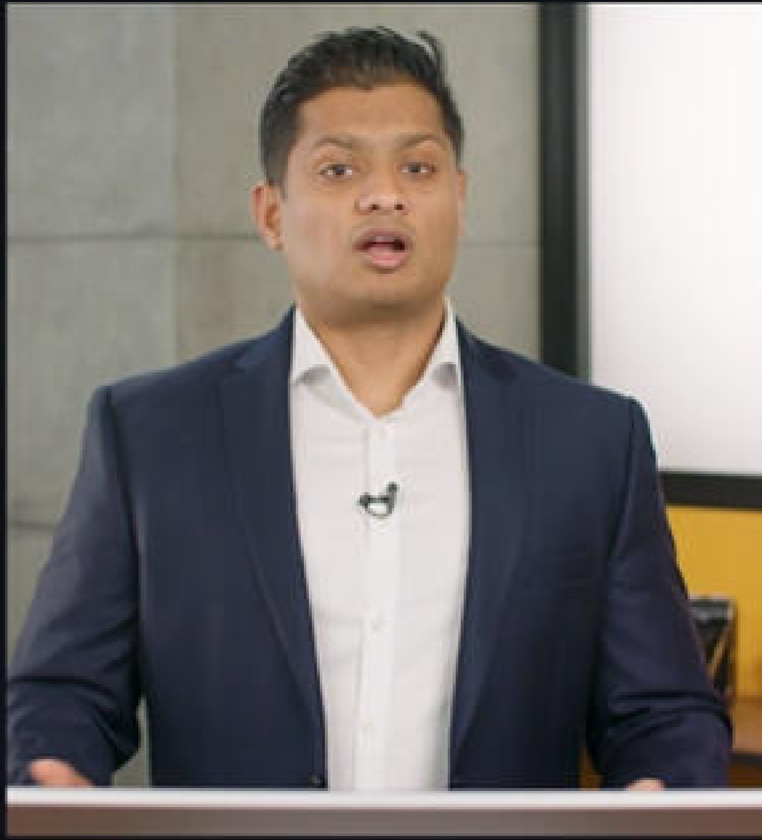


Accelerated compute service for Amazon SageMaker and Amazon EC2

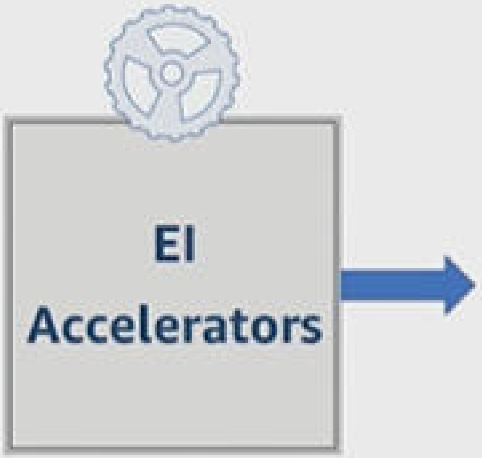


EI accelerators



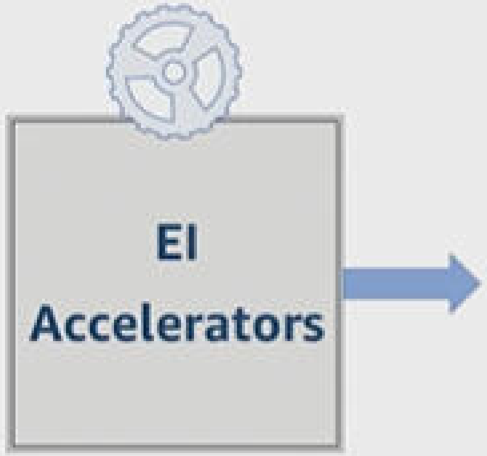


Benefits



Amazon Elastic Interference

Benefits



Amazon Elastic Interference

Reduce interference costs by up to 75%.

Benefits

aws training and certification

EI
Accelerators

Amazon Elastic
Interference

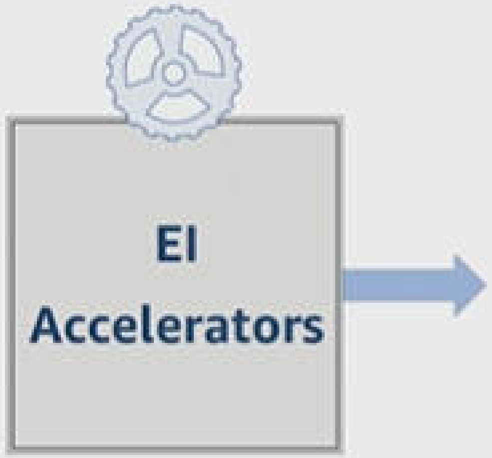
Reduce interference costs by up to **75%.**

Get exactly what you need.

TFLOPS: Tera Floating Operations Per Second

Benefits

aws training and certification



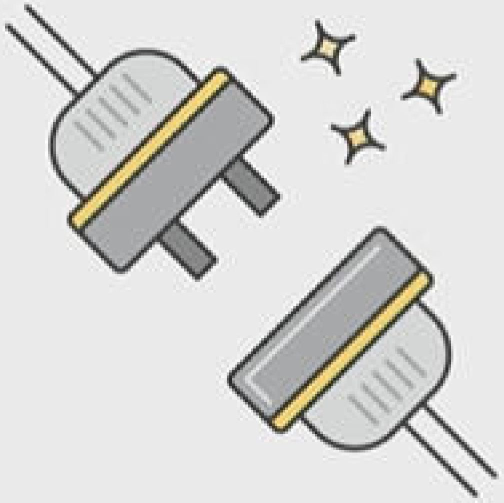
Amazon Elastic Interference

Reduce interference costs by up to **75%.**

Get exactly what you need.

Respond to changes in demand.

Key features



Integrated with Amazon SageMaker and Amazo

Key features

aws training and certification

TensorFlow™ ONNX mxnet

Model support

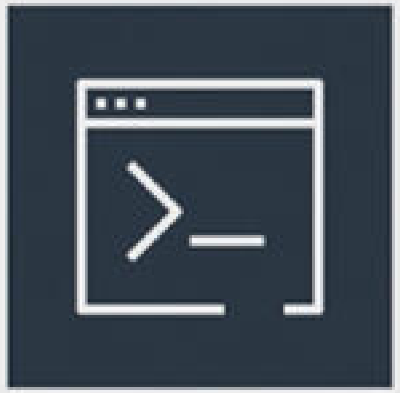
05:42

How does Amazon EI work?

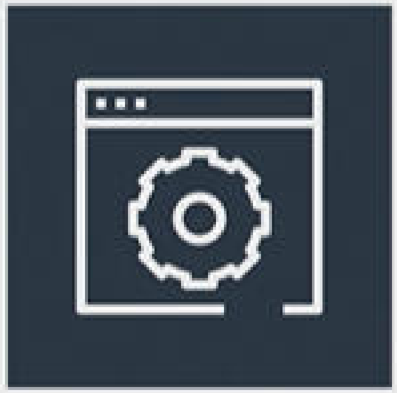
aws training and certification

Instances	Amazon-enhanced libraries	Models loaded	Inference calls
Instances communicate with accelerators	TensorFlow, mxnet: Model serving platforms for these frameworks automatically use EI accelerators	TensorFlow, MXNet, ONNX	Inference: Model

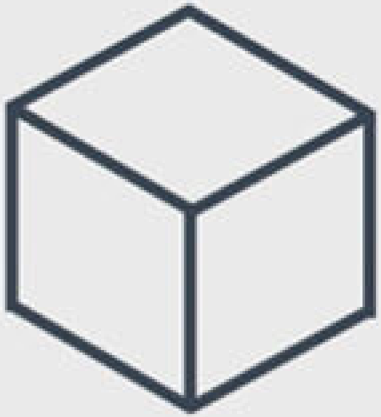
Amazon EI configuration



AWS Command Line Interface (AWS CLI)



AWS Management Console



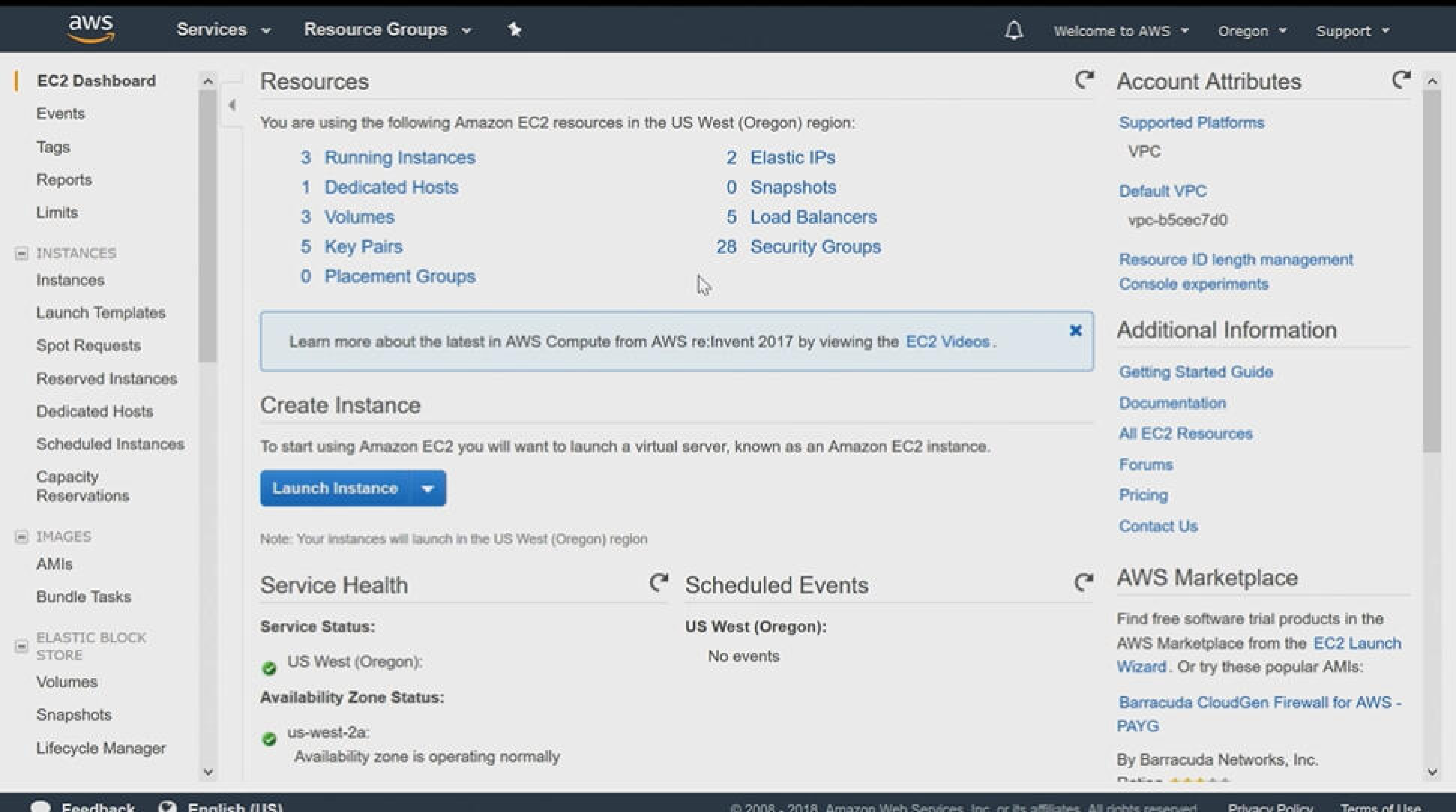
AWS SDK

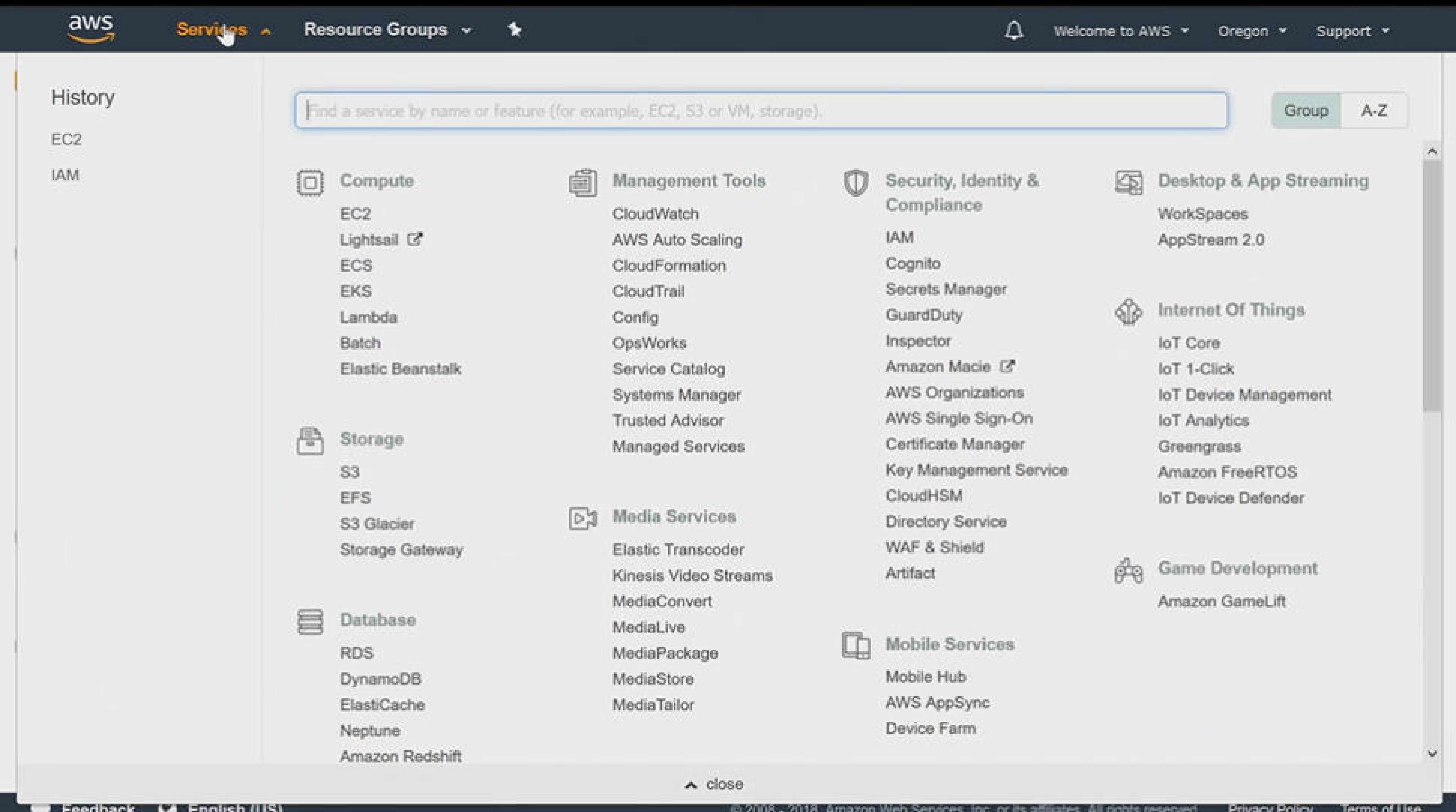
Amazon EI accelerator sizes

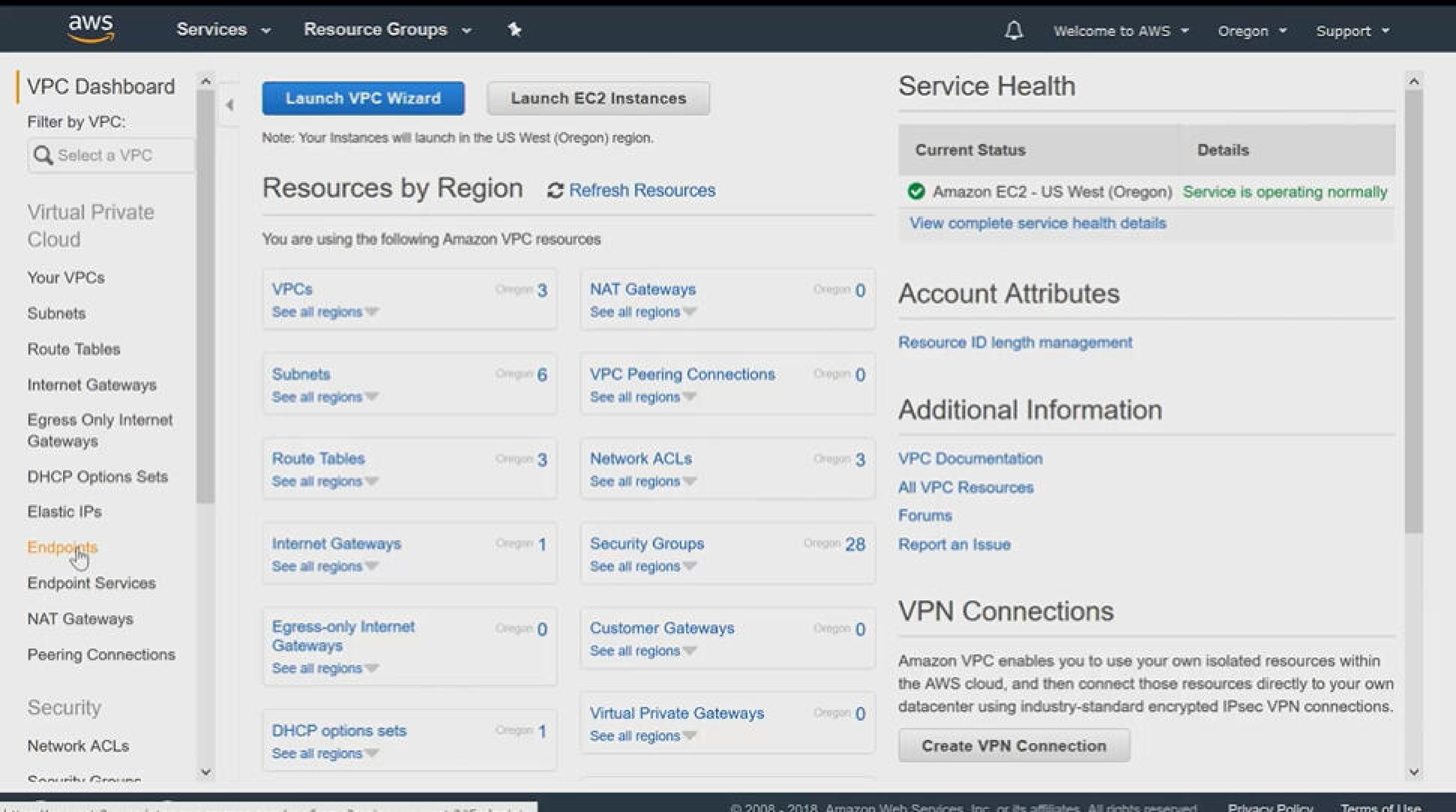
Accelerator Type	FP32 Throughput (TOPS)	FP16 Throughput (TOPS)	Memory (GB)
eia1.medium	1	8	1
eia1.large	2	16	2
eia1.xlarge	4	32	4

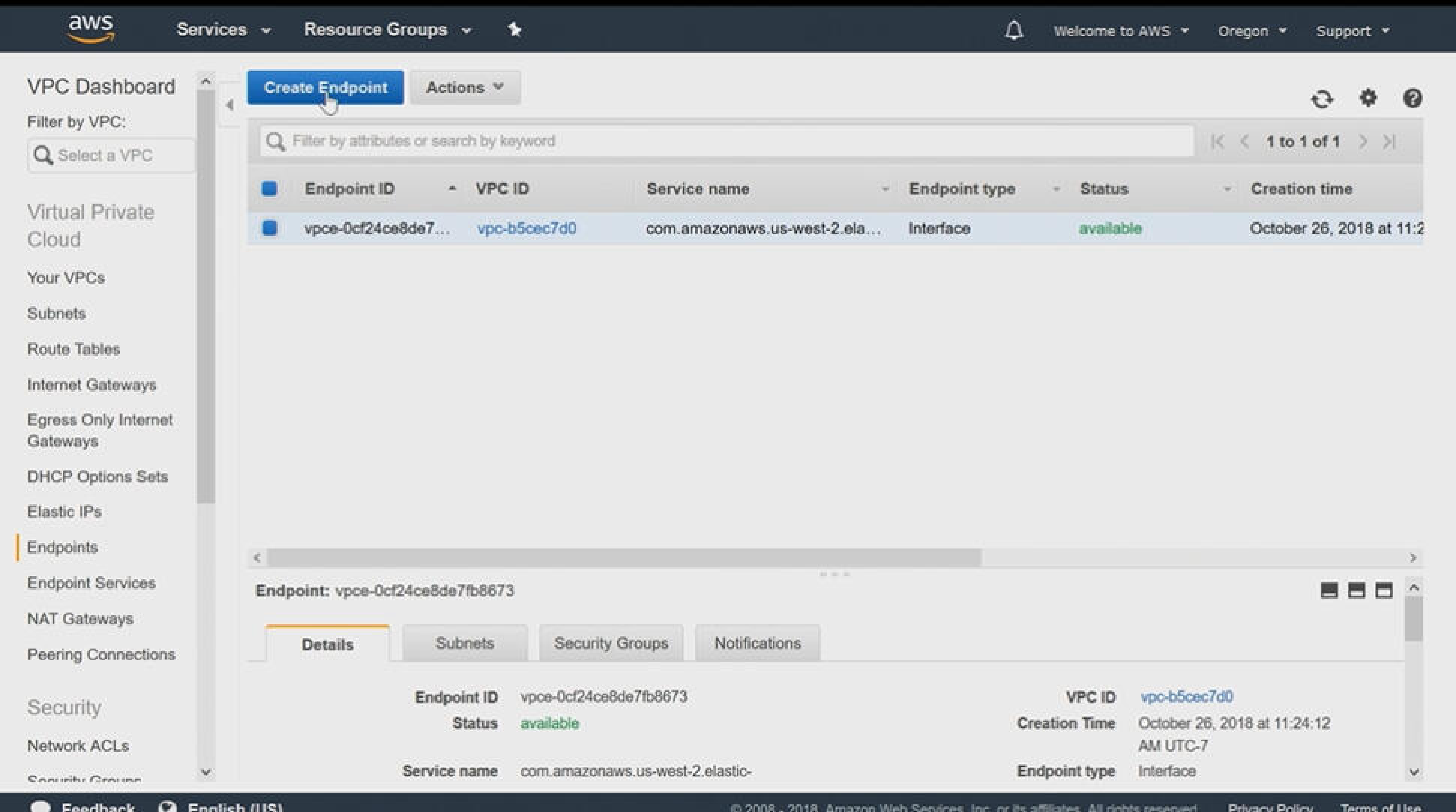
Demo

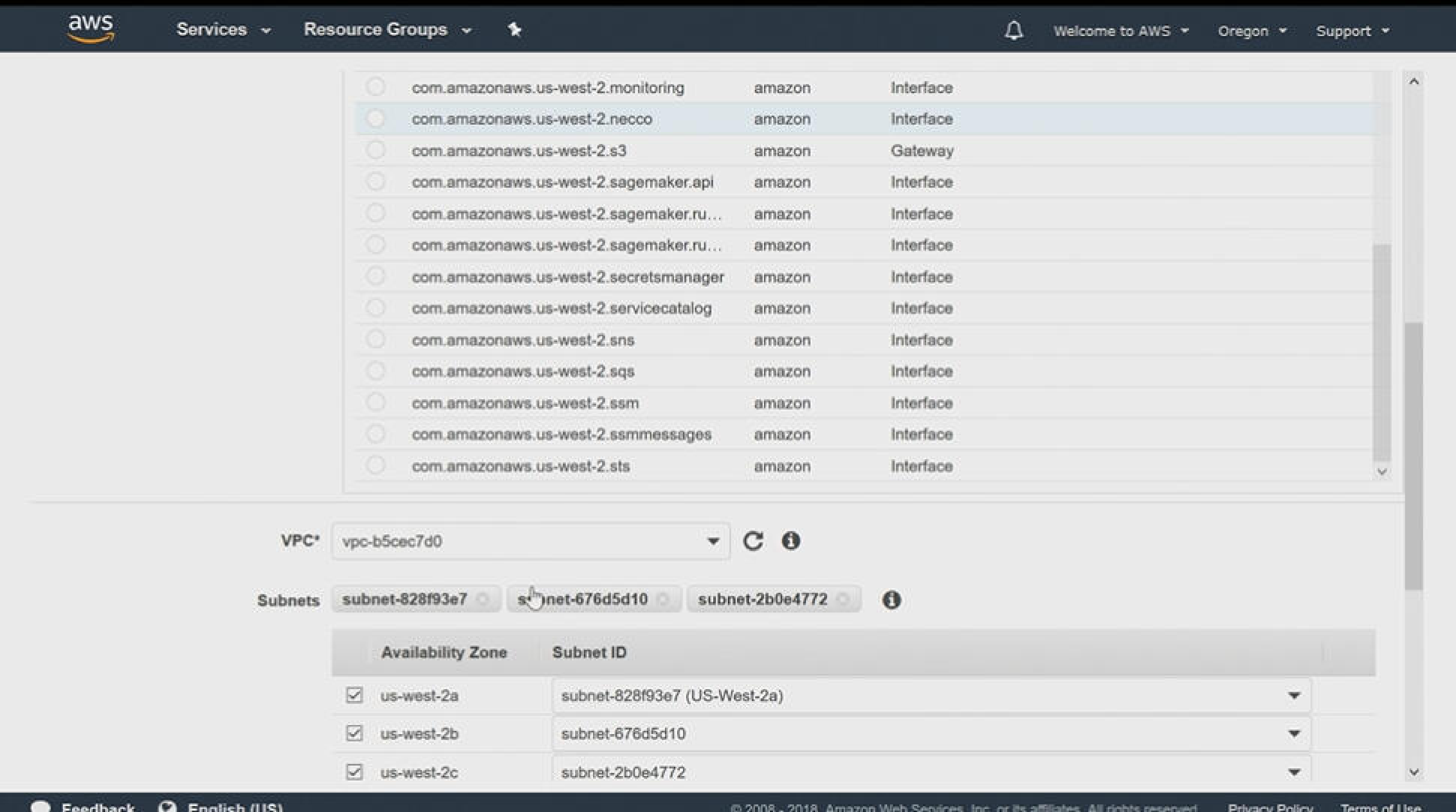
05:42

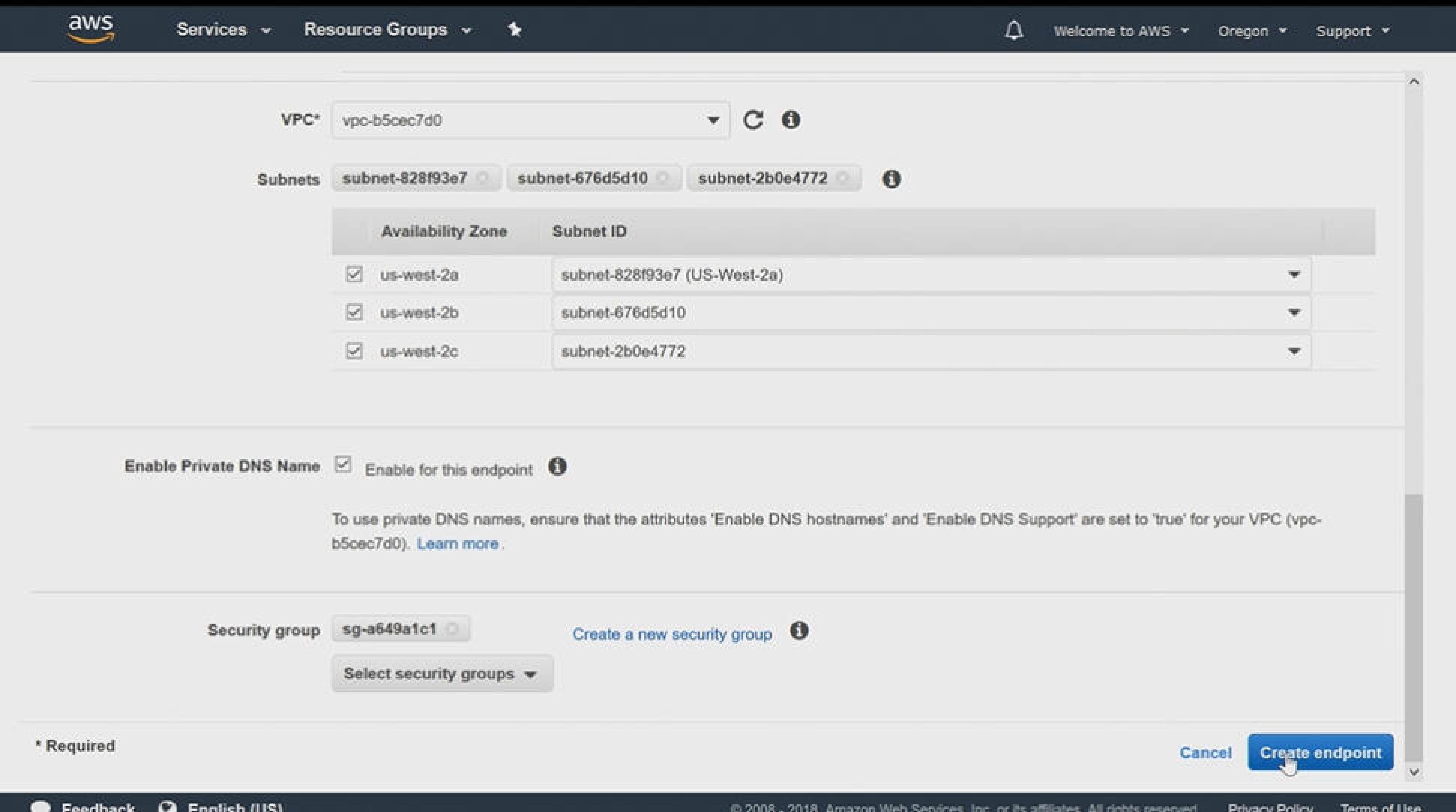

aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
Resources
You are using the following Amazon EC2 resources in the US West (Oregon) region:
3 Running Instances
1 Dedicated Hosts
3 Volumes
5 Key Pairs
0 Placement Groups
2 Elastic IPs
0 Snapshots
5 Load Balancers
28 Security Groups
Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.
Create Instance
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
Launch Instance
Note: Your instances will launch in the US West (Oregon) region
Service Health
Service Status:
US West (Oregon):
Availability Zone Status:
us-west-2a:
Availability zone is operating normally
Scheduled Events
US West (Oregon):
No events
Account Attributes
Supported Platforms
VPC
Default VPC
vpc-b5cec7d0
Resource ID length management
Console experiments
Additional Information
Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us
AWS Marketplace
Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:
Barracuda CloudGen Firewall for AWS - PAYG
By Barracuda Networks, Inc.
Feedback
English (US)
Privacy Policy
Terms of Use

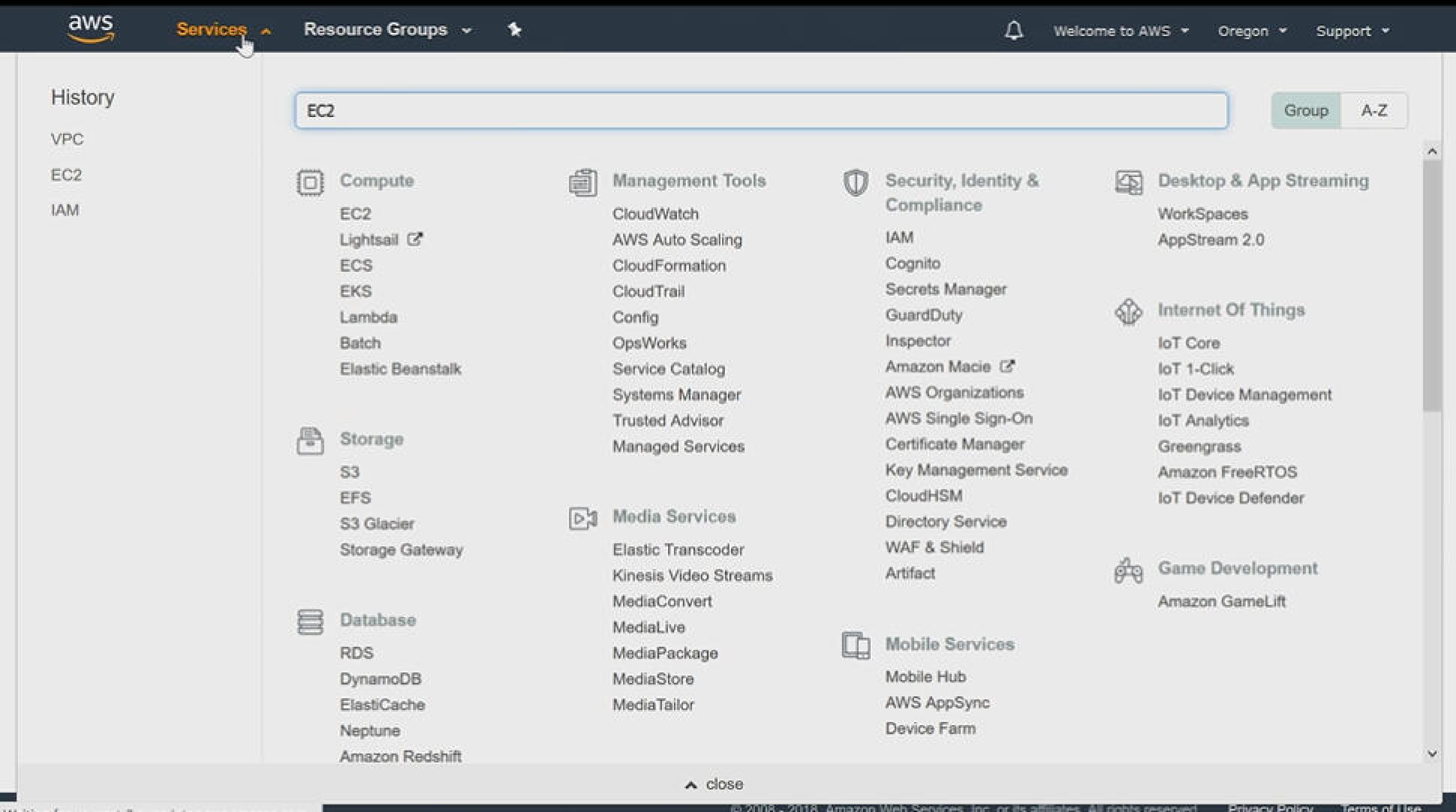

aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
History
EC2
IAM
Find a service by name or feature (for example, EC2, S3 or VM, storage).
Group
A-Z
Compute
EC2
Lightsail
ECS
EKS
Lambda
Batch
Elastic Beanstalk
Storage
S3
EFS
S3 Glacier
Storage Gateway
Database
RDS
DynamoDB
ElastiCache
Neptune
Amazon Redshift
Management Tools
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services
Media Services
Elastic Transcoder
Kinesis Video Streams
MediaConvert
MediaLive
MediaPackage
MediaStore
MediaTailor
Security, Identity & Compliance
IAM
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Organizations
AWS Single Sign-On
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
Artifact
Mobile Services
Mobile Hub
AWS AppSync
Device Farm
Desktop & App Streaming
WorkSpaces
AppStream 2.0
Internet Of Things
IoT Core
IoT 1-Click
IoT Device Management
IoT Analytics
Greengrass
Amazon FreeRTOS
IoT Device Defender
Game Development
Amazon GameLift
close

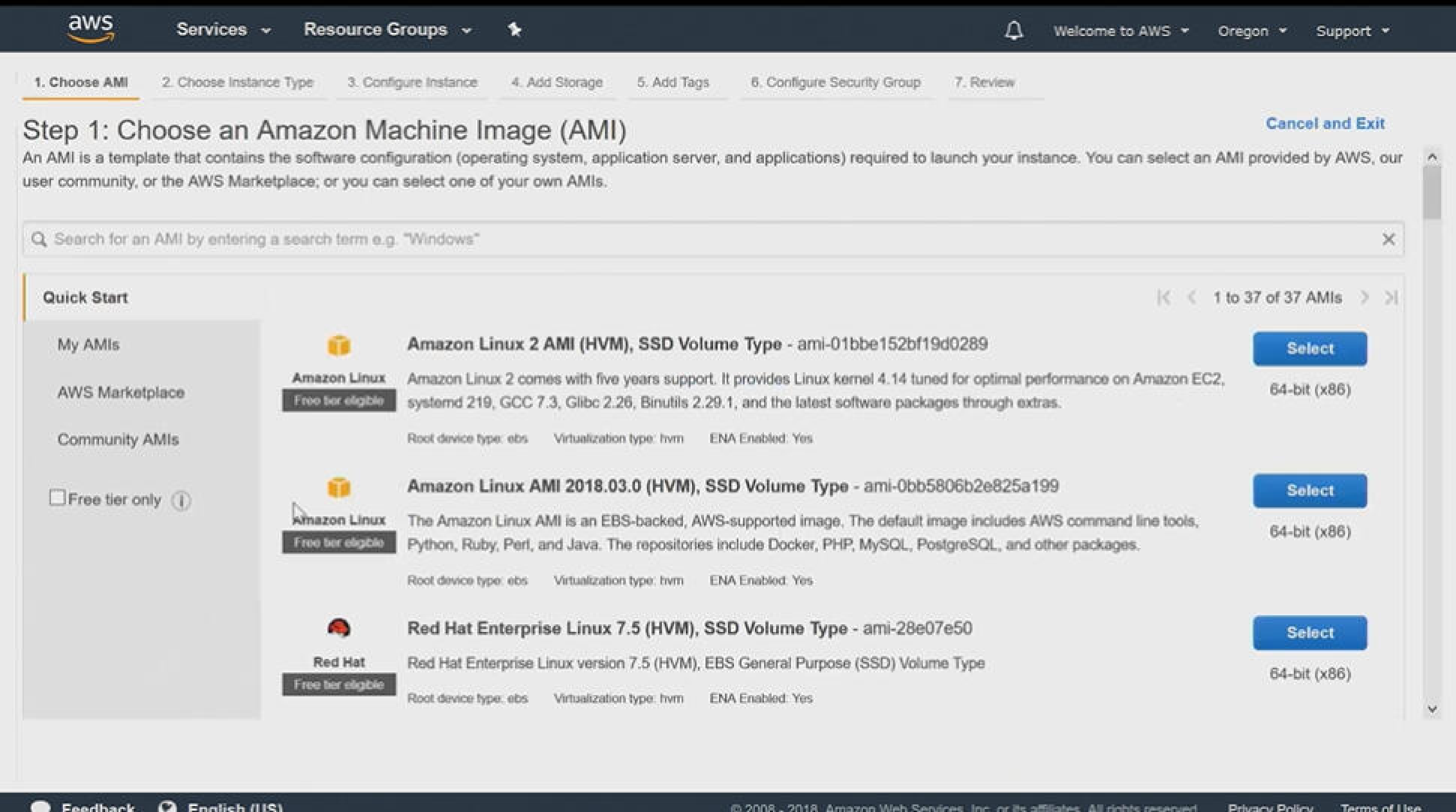

aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Security
Network ACLs
Launch VPC Wizard
Launch EC2 Instances
Note: Your Instances will launch in the US West (Oregon) region.
Resources by Region
Refresh Resources
You are using the following Amazon VPC resources
VPCs Oregon 3 See all regions
NAT Gateways Oregon 0 See all regions
Subnets Oregon 6 See all regions
VPC Peering Connections Oregon 0 See all regions
Route Tables Oregon 3 See all regions
Network ACLs Oregon 3 See all regions
Internet Gateways Oregon 1 See all regions
Security Groups Oregon 28 See all regions
Egress-only Internet Gateways Oregon 0 See all regions
Customer Gateways Oregon 0 See all regions
DHCP options sets Oregon 1 See all regions
Virtual Private Gateways Oregon 0 See all regions
Service Health
Current Status
Details
Amazon EC2 - US West (Oregon)
Service is operating normally
View complete service health details
Account Attributes
Resource ID length management
Additional Information
VPC Documentation
All VPC Resources
Forums
Report an Issue
VPN Connections
Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.
Create VPN Connection
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

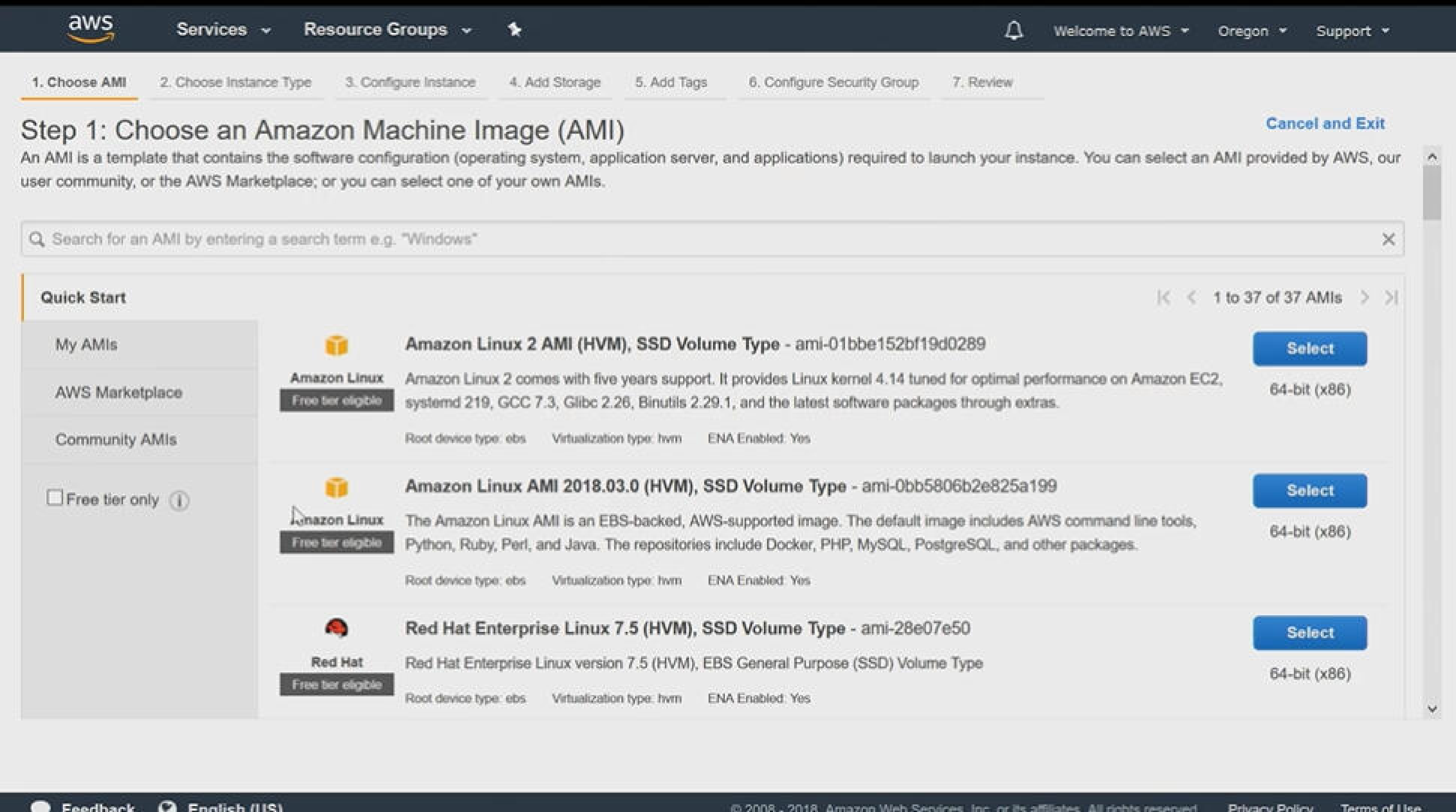

aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Security
Network ACLs
Create Endpoint
Actions
Filter by attributes or search by keyword
1 to 1 of 1
Endpoint ID
VPC ID
Service name
Endpoint type
Status
Creation time
vpce-0cf24ce8de7...
vpc-b5cec7d0
com.amazonaws.us-west-2.ela...
Interface
available
October 26, 2018 at 11:2
Endpoint: vpce-0cf24ce8de7fb8673
Details
Subnets
Security Groups
Notifications
Endpoint ID
vpce-0cf24ce8de7fb8673
Status
available
Service name
com.amazonaws.us-west-2.elastic-
VPC ID
vpc-b5cec7d0
Creation Time
October 26, 2018 at 11:24:12 AM UTC-7
Endpoint type
Interface
Feedback
English (US)
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
com.amazonaws.us-west-2.monitoring amazon Interface
com.amazonaws.us-west-2.necco amazon Interface
com.amazonaws.us-west-2.s3 amazon Gateway
com.amazonaws.us-west-2.sagemaker.api amazon Interface
com.amazonaws.us-west-2.sagemaker.ru... amazon Interface
com.amazonaws.us-west-2.sagemaker.ru... amazon Interface
com.amazonaws.us-west-2.secretsmanager amazon Interface
com.amazonaws.us-west-2.servicecatalog amazon Interface
com.amazonaws.us-west-2.sns amazon Interface
com.amazonaws.us-west-2.sqs amazon Interface
com.amazonaws.us-west-2.ssm amazon Interface
com.amazonaws.us-west-2.ssmmessages amazon Interface
com.amazonaws.us-west-2.sts amazon Interface
VPC*
vpc-b5cec7d0
Subnets
subnet-828f93e7
subnet-2b0e4772
Availability Zone
Subnet ID
us-west-2a
subnet-828f93e7 (US-West-2a)
us-west-2b
subnet-676d5d10
us-west-2c
subnet-2b0e4772
Feedback
English (US)
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
VPC*
vpc-b5cec7d0
Subnets
subnet-828f93e7
subnet-676d5d10
subnet-2b0e4772
Availability Zone
Subnet ID
us-west-2a
subnet-828f93e7 (US-West-2a)
us-west-2b
subnet-676d5d10
us-west-2c
subnet-2b0e4772
Enable Private DNS Name
Enable for this endpoint
To use private DNS names, ensure that the attributes 'Enable DNS hostnames' and 'Enable DNS Support' are set to 'true' for your VPC (vpc-b5cec7d0). Learn more.
Security group
sg-a649a1c1
Create a new security group
Select security groups
* Required
Cancel
Create endpoint
Feedback
English (US)
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
History
VPC
EC2
IAM
EC2
Group
A-Z
Compute
EC2
Lightsail
ECS
EKS
Lambda
Batch
Elastic Beanstalk
Storage
S3
EFS
S3 Glacier
Storage Gateway
Database
RDS
DynamoDB
ElastiCache
Neptune
Amazon Redshift
Management Tools
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services
Media Services
Elastic Transcoder
Kinesis Video Streams
MediaConvert
MediaLive
MediaPackage
MediaStore
MediaTailor
Security, Identity & Compliance
IAM
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Organizations
AWS Single Sign-On
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
Artifact
Mobile Services
Mobile Hub
AWS AppSync
Device Farm
Desktop & App Streaming
WorkSpaces
AppStream 2.0
Internet Of Things
IoT Core
IoT 1-Click
IoT Device Management
IoT Analytics
Greengrass
Amazon FreeRTOS
IoT Device Defender
Game Development
Amazon GameLift
close


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
Search for an AMI by entering a search term e.g. "Windows"
Quick Start
1 to 37 of 37 AMIs
My AMIs
AWS Marketplace
Community AMIs
Free tier only
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-01bbe152bf19d0289
Amazon Linux
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0bb5806b2e825a199
Amazon Linux
Free tier eligible
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-28e07e50
Red Hat
Free tier eligible
Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
Search for an AMI by entering a search term e.g. "Windows"
Quick Start
1 to 37 of 37 AMIs
My AMIs
AWS Marketplace
Community AMIs
Free tier only
Amazon Linux
Free tier eligible
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-01bbe152bf19d0289
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
Amazon Linux
Free tier eligible
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0bb5806b2e825a199
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
Red Hat
Free tier eligible
Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-28e07e50
Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs
Virtualization type: hvm
ENA Enabled: Yes
Select
64-bit (x86)
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Services Resource Groups Welcome to AWS Oregon Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3a.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3a.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
	General purpose	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details aws Services Resource Groups

Welcome to AWS Oregon Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

General purpose	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	m5ad.large	2	8	1 x 75 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.xlarge	4	16	1 x 150 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.2xlarge	8	32	1 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.4xlarge	16	64	2 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.12xlarge	48	192	2 x 900 (SSD)	Yes	10 Gigabit	Yes
General purpose	m5ad.24xlarge	96	384	4 x 900 (SSD)	Yes	20 Gigabit	Yes

Cancel Previous **Review and Launch** **Next: Configure Instance Details**

Feedback English (US) Privacy Policy Terms of Use

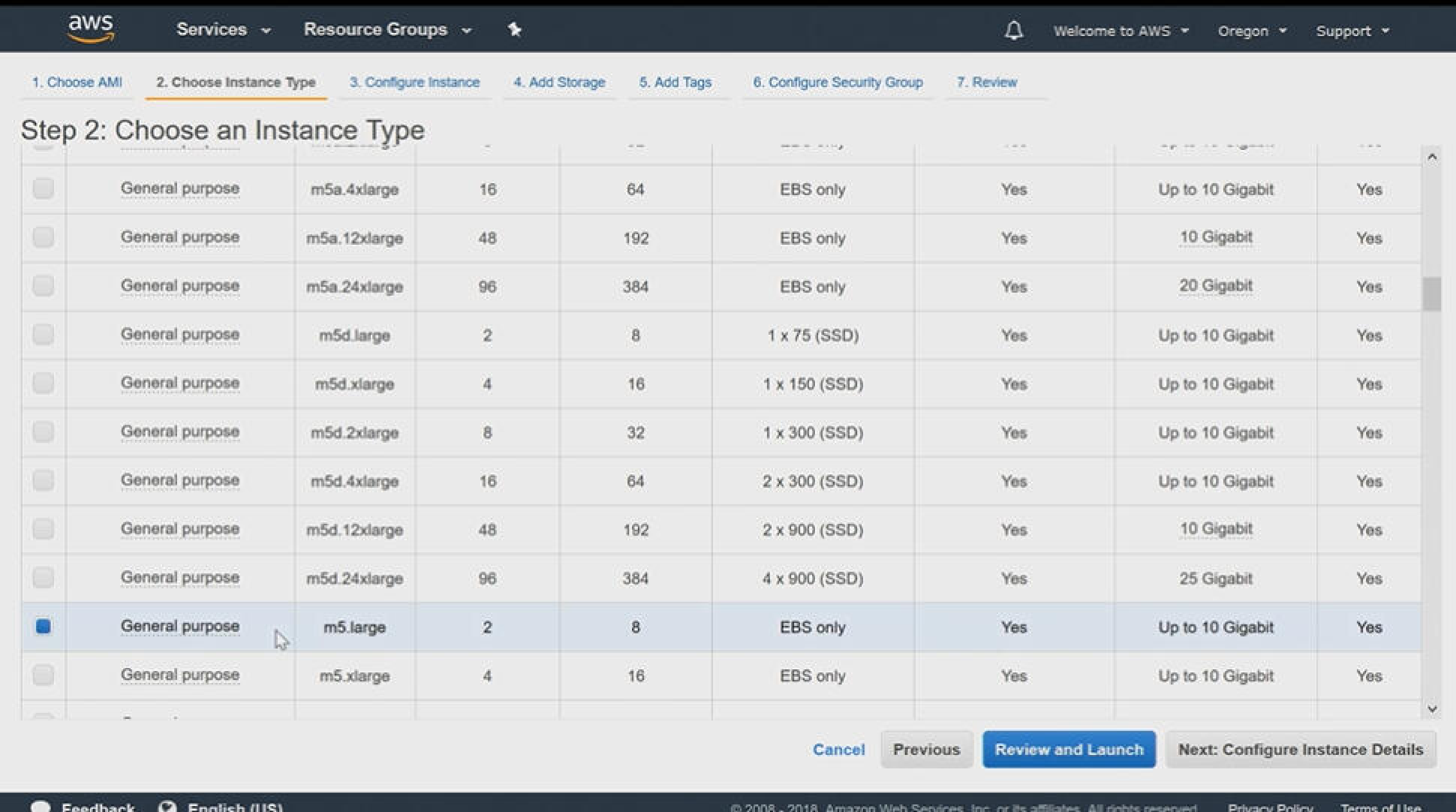
aws Services Resource Groups Welcome to AWS Oregon Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 2: Choose an Instance Type

General purpose	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	t3.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes
General purpose	m5ad.large	2	8	1 x 75 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.xlarge	4	16	1 x 150 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.2xlarge	8	32	1 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.4xlarge	16	64	2 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
General purpose	m5ad.12xlarge	48	192	2 x 900 (SSD)	Yes	10 Gigabit	Yes
General purpose	m5ad.24xlarge	96	384	4 x 900 (SSD)	Yes	20 Gigabit	Yes
General purpose	m5a.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
General purpose	m5a.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes
General purpose	m5a.2xlarge	8	32	EBS only	Yes	Up to 10 Gigabit	Yes
General purpose	m5a.4xlarge	16	64	EBS only	Yes	Up to 10 Gigabit	Yes

Cancel | Previous | Review and Launch | Next: Configure Instance Details


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
General purpose m5a.4xlarge 16 64 EBS only Yes Up to 10 Gigabit Yes
General purpose m5a.12xlarge 48 192 EBS only Yes 10 Gigabit Yes
General purpose m5a.24xlarge 96 384 EBS only Yes 20 Gigabit Yes
General purpose m5d.large 2 8 1 x 75 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.xlarge 4 16 1 x 150 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.2xlarge 8 32 1 x 300 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.4xlarge 16 64 2 x 300 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.12xlarge 48 192 2 x 900 (SSD) Yes 10 Gigabit Yes
General purpose m5d.24xlarge 96 384 4 x 900 (SSD) Yes 25 Gigabit Yes
General purpose m5.large 2 8 EBS only Yes Up to 10 Gigabit Yes
General purpose m5.xlarge 4 16 EBS only Yes Up to 10 Gigabit Yes
Cancel
Previous
Review and Launch
Next: Configure Instance Details
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

aws Services Resource Groups Welcome to AWS Oregon Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 2: Choose an Instance Type

☐	General purpose	m5a.4xlarge	16	64	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5a.12xlarge	48	192	EBS only	Yes	10 Gigabit	Yes
☐	General purpose	m5a.24xlarge	96	384	EBS only	Yes	20 Gigabit	Yes
☐	General purpose	m5d.large	2	8	1 x 75 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.xlarge	4	16	1 x 150 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.2xlarge	8	32	1 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.4xlarge	16	64	2 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.12xlarge	48	192	2 x 900 (SSD)	Yes	10 Gigabit	Yes
☐	General purpose	m5d.24xlarge	96	384	4 x 900 (SSD)	Yes	25 Gigabit	Yes
☑	General purpose	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes

Cancel | Previous | Review and Launch | Next: Configure Instance Details

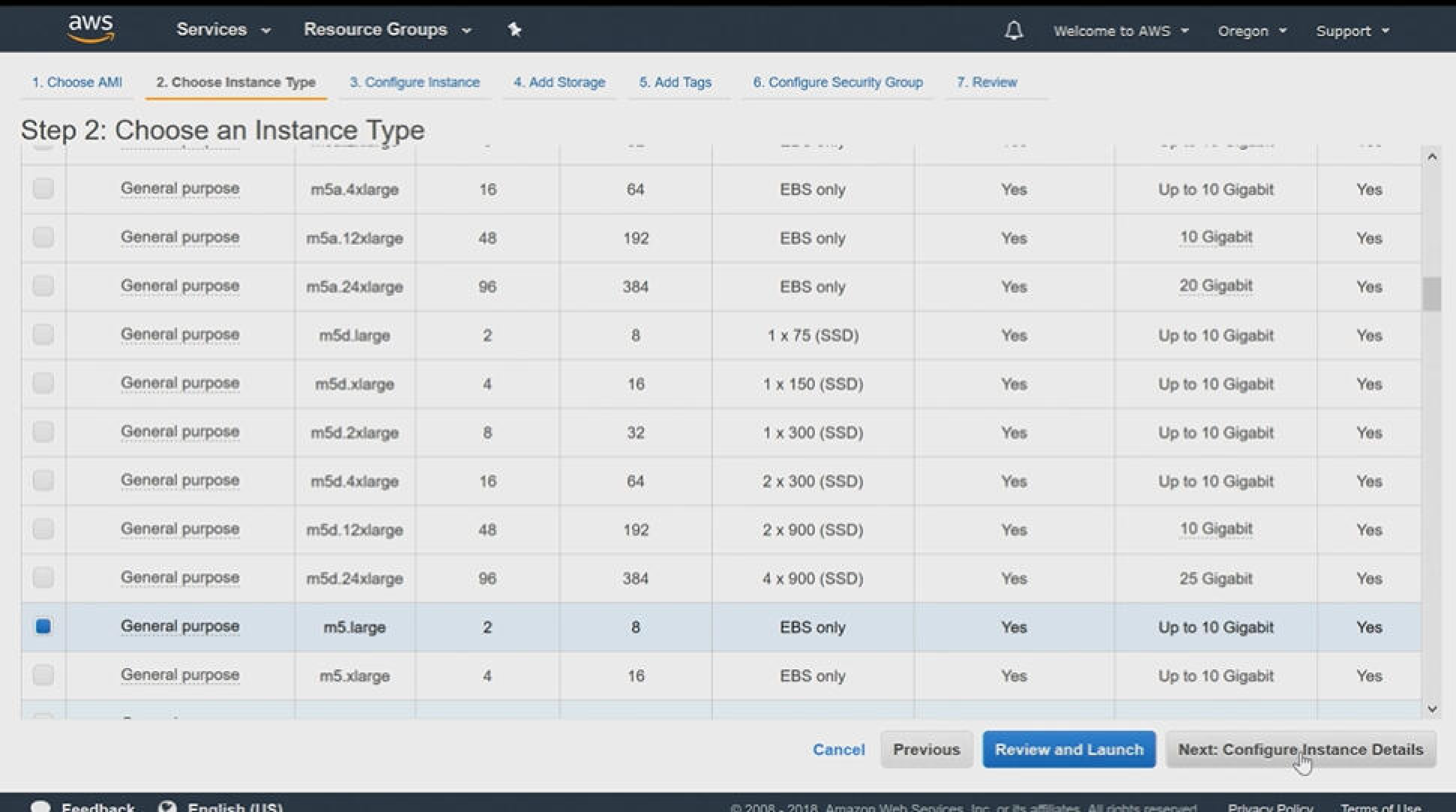
aws Services Resource Groups Welcome to AWS Oregon Support

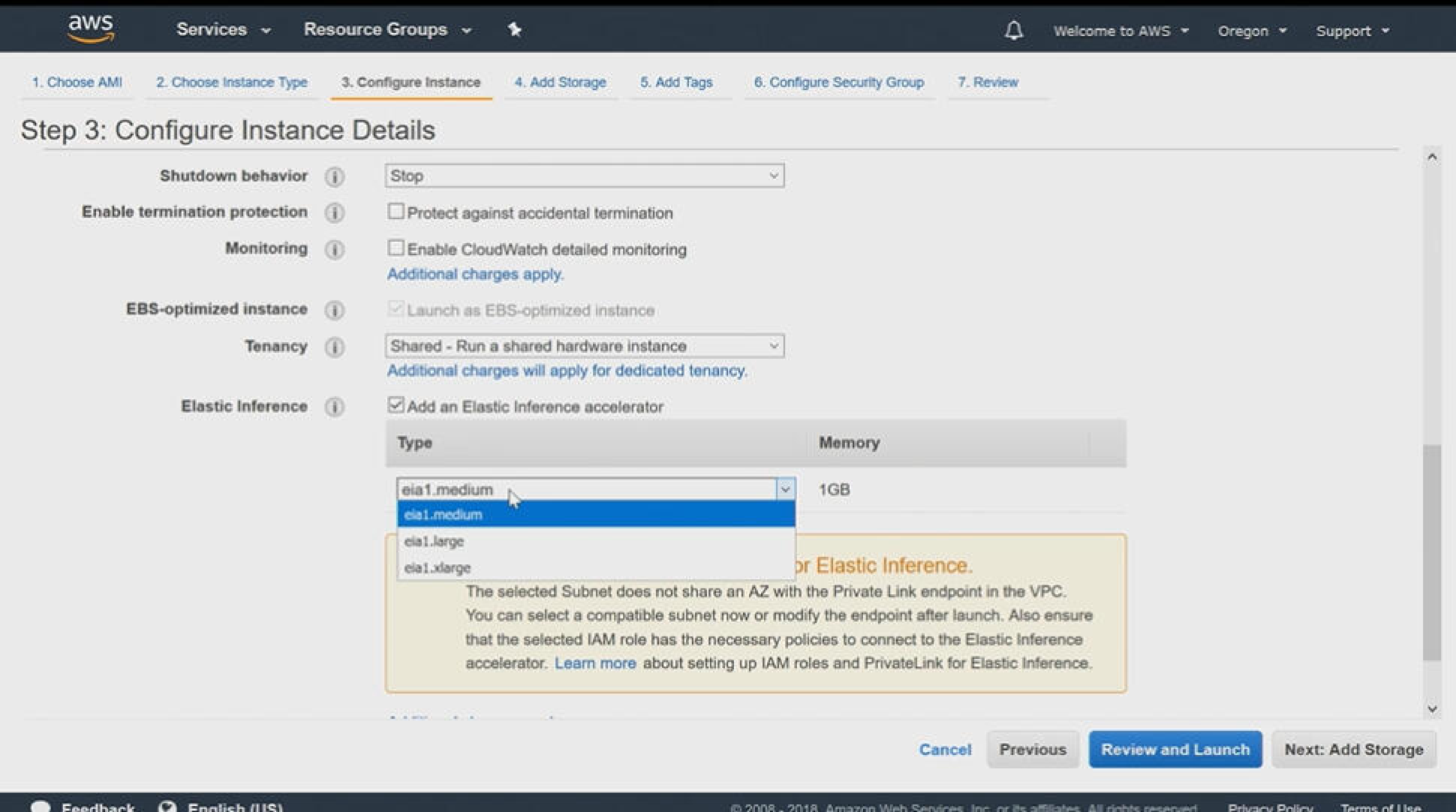
1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

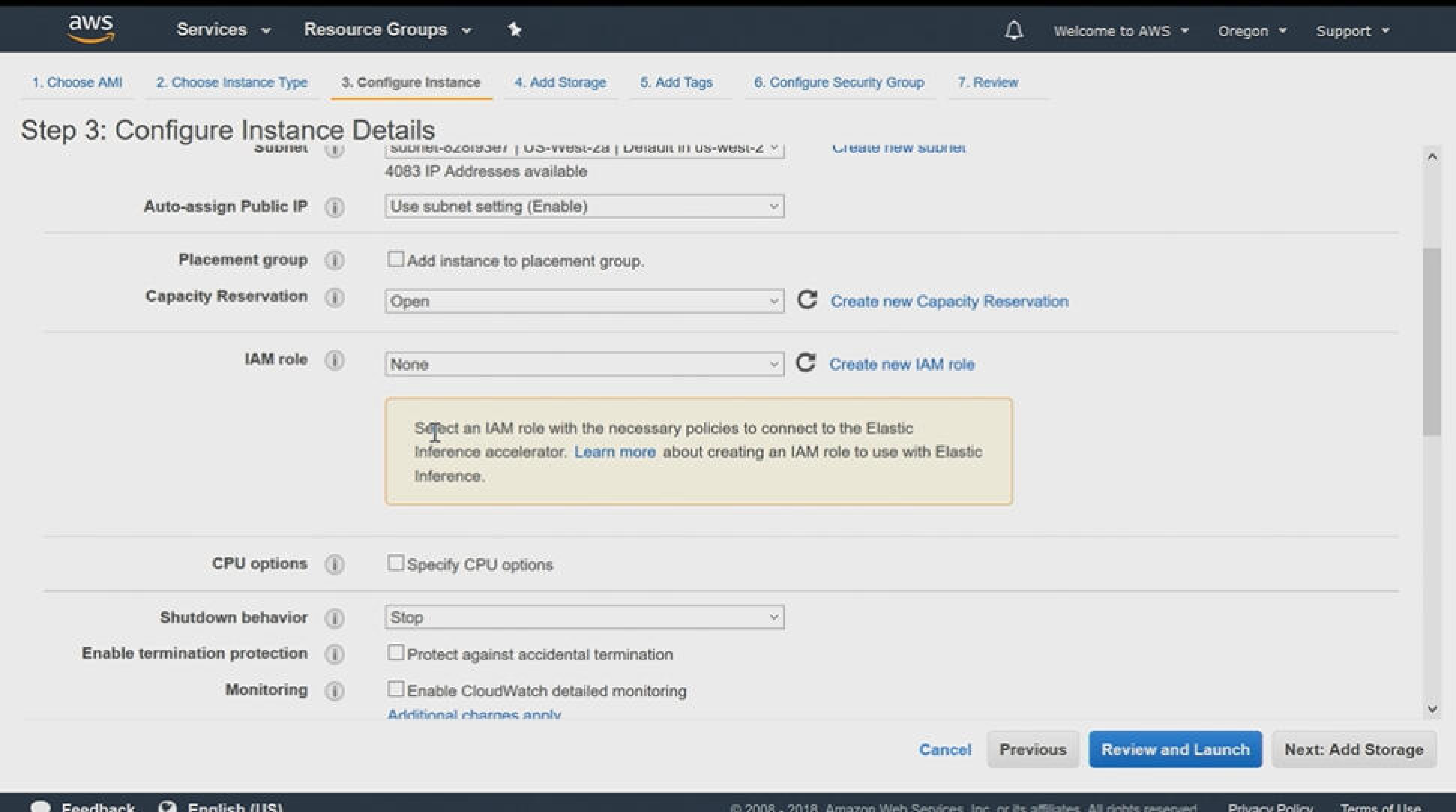
Step 2: Choose an Instance Type

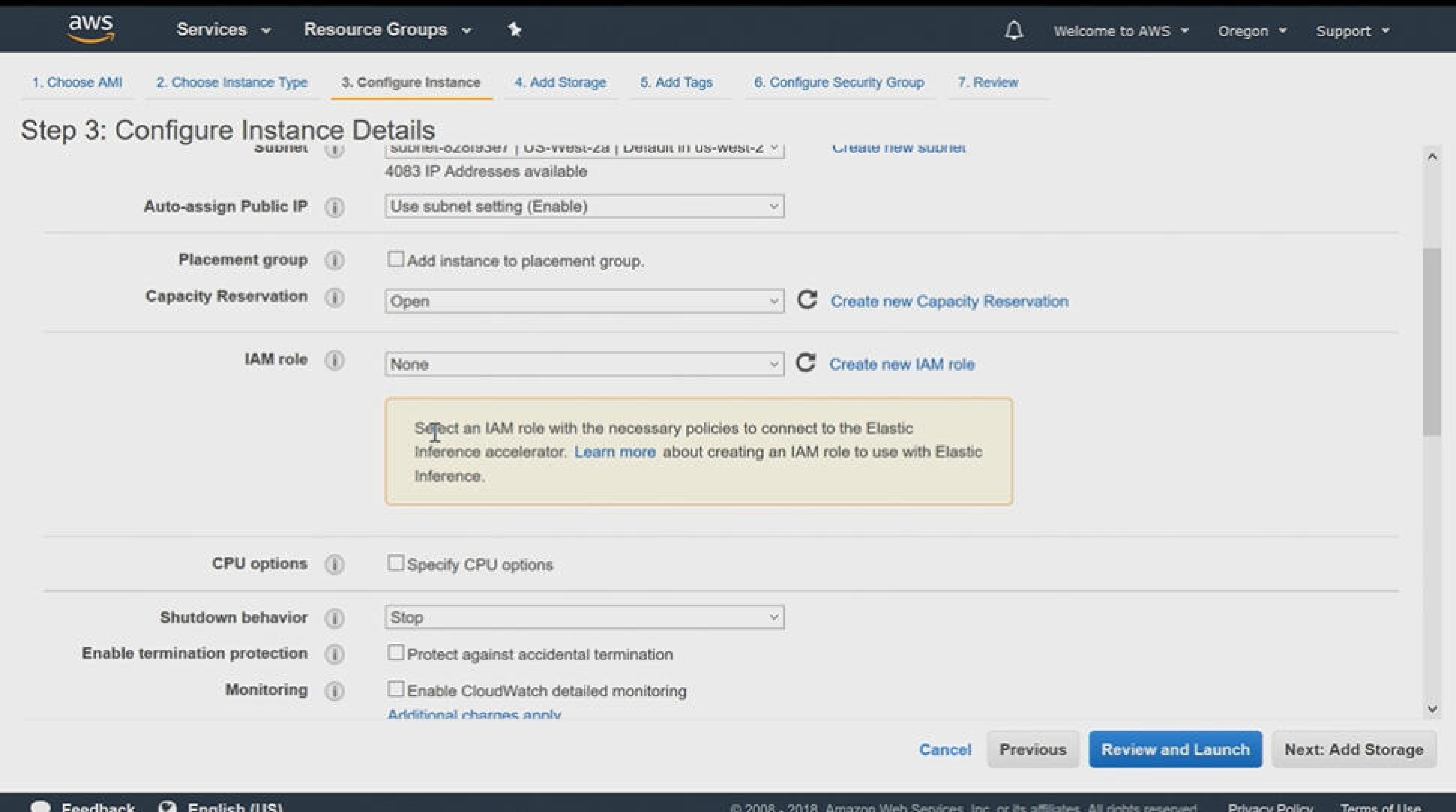
☐	General purpose	m5a.4xlarge	16	64	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5a.12xlarge	48	192	EBS only	Yes	10 Gigabit	Yes
☐	General purpose	m5a.24xlarge	96	384	EBS only	Yes	20 Gigabit	Yes
☐	General purpose	m5d.large	2	8	1 x 75 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.xlarge	4	16	1 x 150 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.2xlarge	8	32	1 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.4xlarge	16	64	2 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5d.12xlarge	48	192	2 x 900 (SSD)	Yes	10 Gigabit	Yes
☐	General purpose	m5d.24xlarge	96	384	4 x 900 (SSD)	Yes	25 Gigabit	Yes
■	General purpose	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes

Cancel | Previous | Review and Launch | Next: Configure Instance Details

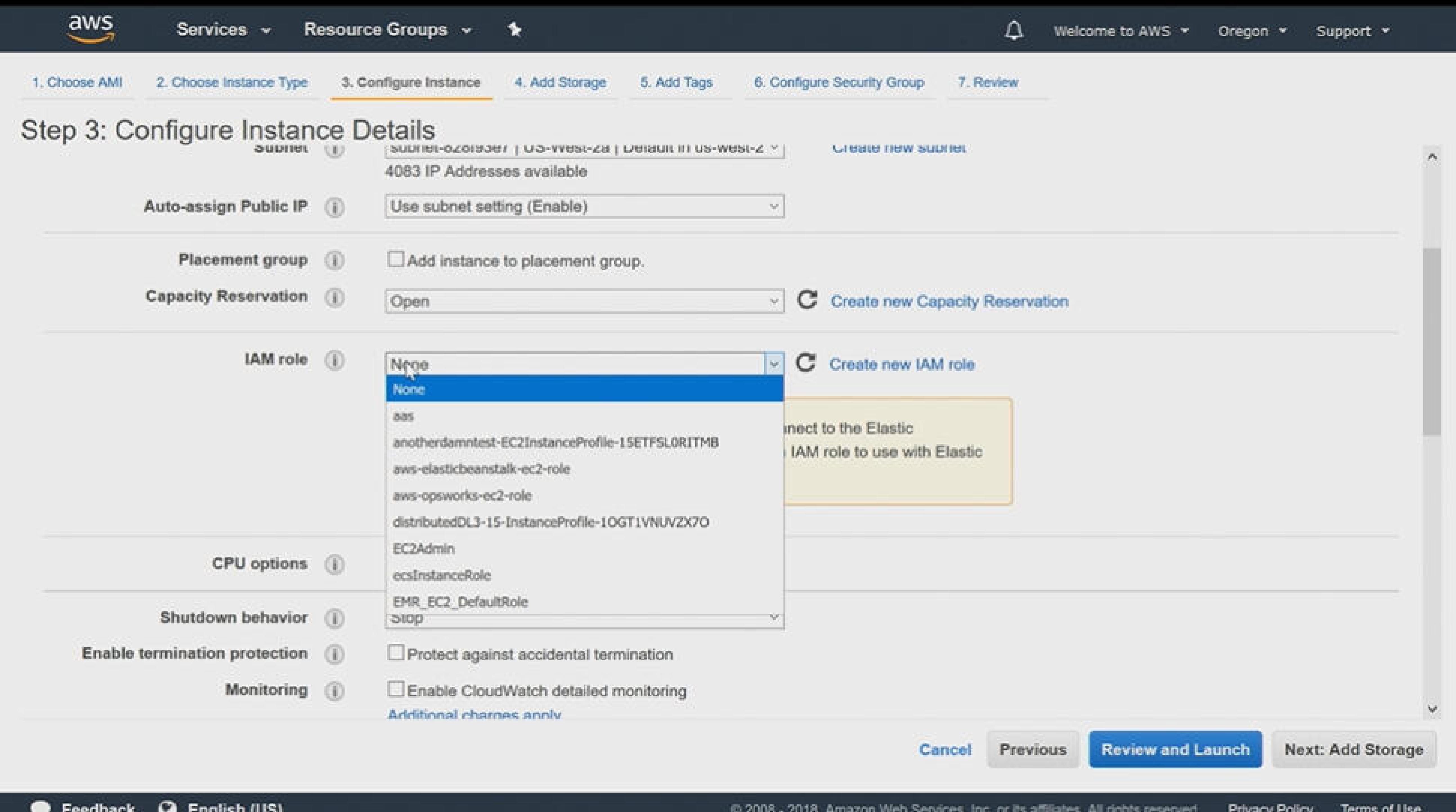

aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
General purpose m5a.4xlarge 16 64 EBS only Yes Up to 10 Gigabit Yes
General purpose m5a.12xlarge 48 192 EBS only Yes 10 Gigabit Yes
General purpose m5a.24xlarge 96 384 EBS only Yes 20 Gigabit Yes
General purpose m5d.large 2 8 1 x 75 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.xlarge 4 16 1 x 150 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.2xlarge 8 32 1 x 300 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.4xlarge 16 64 2 x 300 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5d.12xlarge 48 192 2 x 900 (SSD) Yes 10 Gigabit Yes
General purpose m5d.24xlarge 96 384 4 x 900 (SSD) Yes 25 Gigabit Yes
General purpose m5.large 2 8 EBS only Yes Up to 10 Gigabit Yes
General purpose m5.xlarge 4 16 EBS only Yes Up to 10 Gigabit Yes
Cancel
Previous
Review and Launch
Next: Configure Instance Details


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
Shutdown behavior
Stop
Enable termination protection
Protect against accidental termination
Monitoring
Enable CloudWatch detailed monitoring
Additional charges apply.
EBS-optimized instance
Launch as EBS-optimized instance
Tenancy
Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.
Elastic Inference
Add an Elastic Inference accelerator
Type
Memory
eia1.medium
1GB
eia1.medium
eia1.large
eia1.xlarge
r Elastic Inference.
The selected Subnet does not share an AZ with the Private Link endpoint in the VPC.
You can select a compatible subnet now or modify the endpoint after launch. Also ensure
that the selected IAM role has the necessary policies to connect to the Elastic Inference
accelerator. Learn more about setting up IAM roles and PrivateLink for Elastic Inference.
Cancel
Previous
Review and Launch
Next: Add Storage
Feedback
English (US)
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
Subnet
Create new subnet
4083 IP Addresses available
Auto-assign Public IP
Use subnet setting (Enable)
Placement group
Add instance to placement group.
Capacity Reservation
Open
Create new Capacity Reservation
IAM role
None
Create new IAM role
Select an IAM role with the necessary policies to connect to the Elastic Inference accelerator. Learn more about creating an IAM role to use with Elastic Inference.
CPU options
Specify CPU options
Shutdown behavior
Stop
Enable termination protection
Protect against accidental termination
Monitoring
Enable CloudWatch detailed monitoring
Additional charges apply
Cancel
Previous
Review and Launch
Next: Add Storage
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
4083 IP Addresses available
Auto-assign Public IP
Use subnet setting (Enable)
Placement group
Add instance to placement group.
Capacity Reservation
Open
Create new Capacity Reservation
IAM role
None
Create new IAM role
Select an IAM role with the necessary policies to connect to the Elastic Inference accelerator. Learn more about creating an IAM role to use with Elastic Inference.
CPU options
Specify CPU options
Shutdown behavior
Stop
Enable termination protection
Protect against accidental termination
Monitoring
Enable CloudWatch detailed monitoring
Cancel
Previous
Review and Launch
Next: Add Storage


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
4083 IP Addresses available
Auto-assign Public IP
Use subnet setting (Enable)
Placement group
Add instance to placement group.
Capacity Reservation
Open
Create new Capacity Reservation
IAM role
None
Create new IAM role
None
aas
anotherdamntest-EC2InstanceProfile-15ETFSL0RITMB
aws-elasticbeanstalk-ec2-role
aws-opsworks-ec2-role
distributedDL3-15-InstanceProfile-1OGT1VNUVZX7O
EC2Admin
ecsInstanceRole
EMR_EC2_DefaultRole
nect to the Elastic
IAM role to use with Elastic
CPU options
Shutdown behavior
Enable termination protection
Protect against accidental termination
Monitoring
Enable CloudWatch detailed monitoring
Cancel
Previous
Review and Launch
Next: Add Storage

aws Services Resource Groups Welcome to AWS Oregon Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Your instance configuration is not eligible for the free usage tier

To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about free usage tier eligibility and usage restrictions.

Don't show me this again

AMI Details

Edit AMI

Deep Learning AMI (Amazon Linux) Version 18.0 - ami-0454f6c5e35766c6e

With latest deep learning frameworks pre-installed: MXNet, TensorFlow, PyTorch, Keras, Chainer, Caffe/2, Theano & CNTK, configured with NVIDIA CUDA, cuDNN, NCCL & Intel MKL-DNN. For a fully managed experience, check: https://aws.amazon.com/sagemaker

Root Device Type: ebs Virtualization type: hvm

Instance Type

Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
m5.large	10	2	8	EBS only	Yes	Up to 10 Gigabit

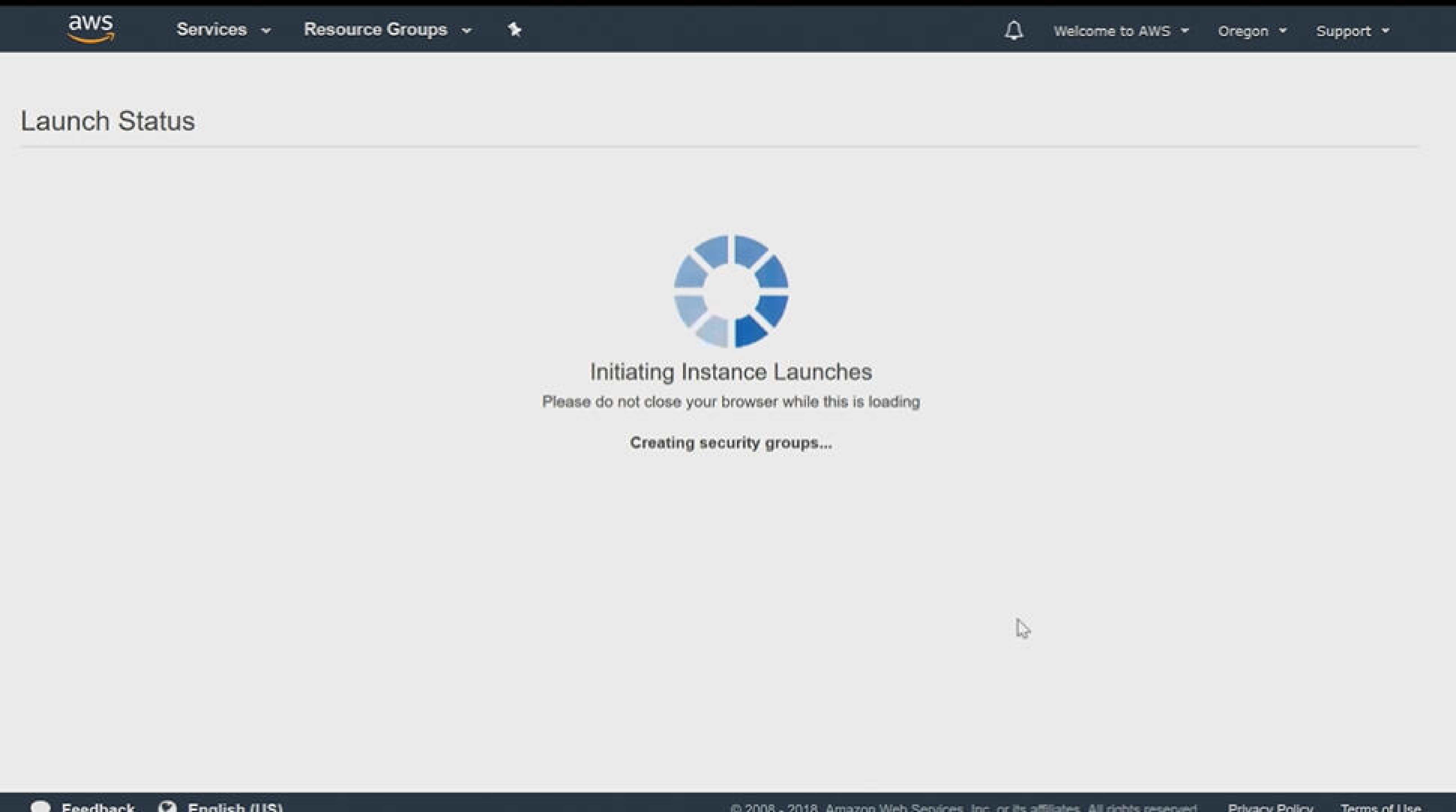
Security Groups

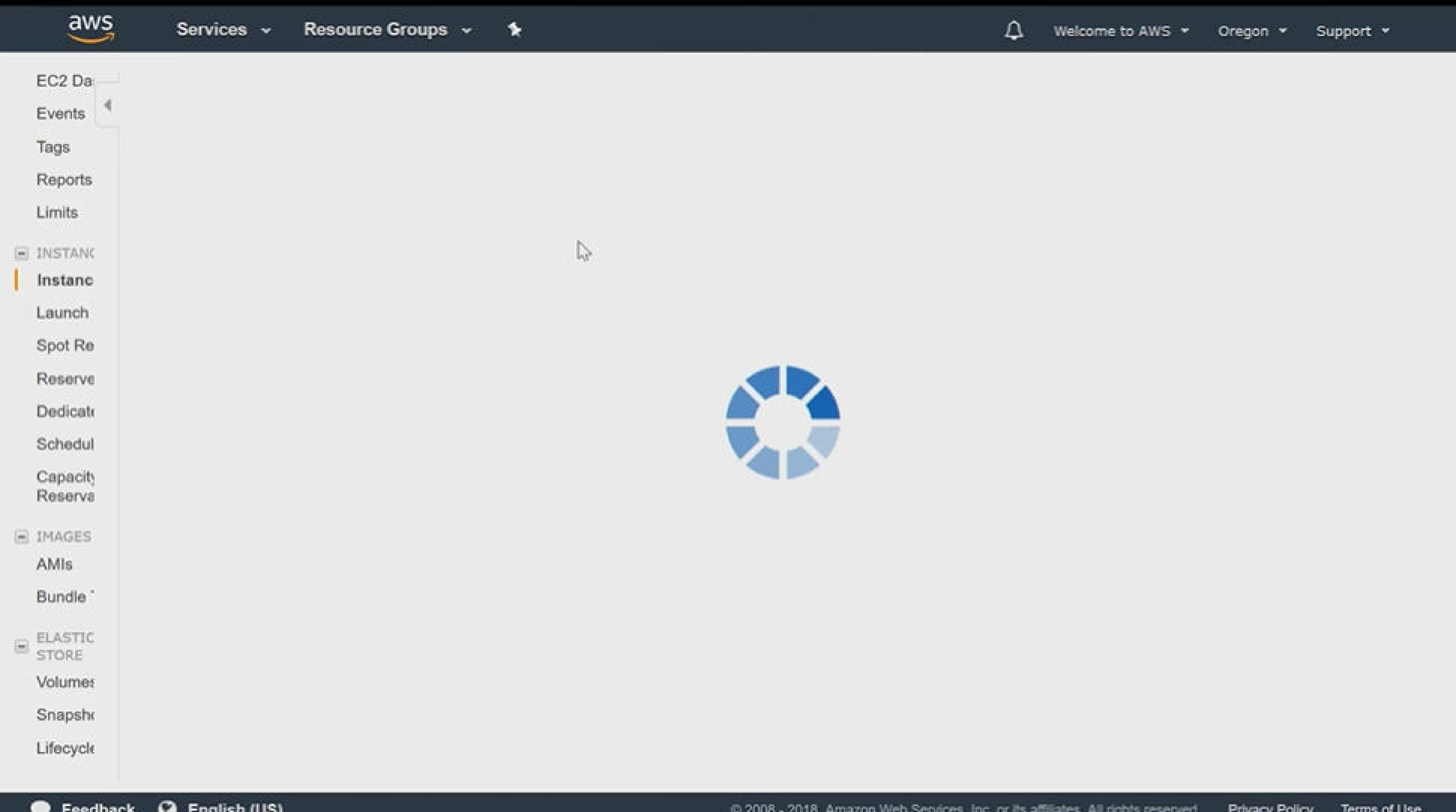
Edit security groups

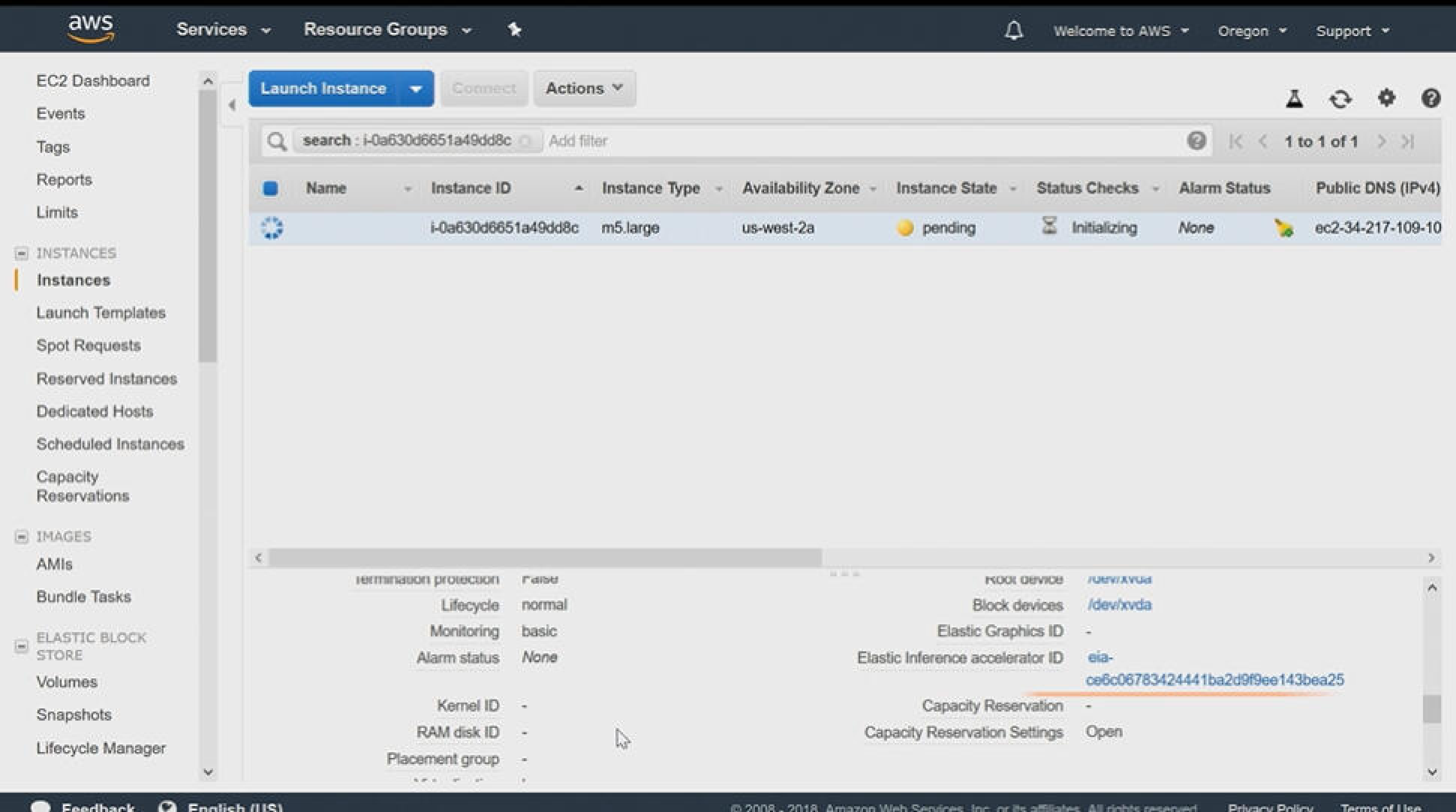
Cancel Previous Launch

Feedback English (US)

 Privacy Policy Terms of Use


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
Launch Status
Initiating Instance Launches
Please do not close your browser while this is loading
Creating security groups...
Feedback
English (US)
Privacy Policy
Terms of Use


aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
EC2 Da
Events
Tags
Reports
Limits
INSTANC
Instanc
Launch
Spot Re
Reserve
Dedicat
Schedul
Capacity
Reserva
IMAGES
AMIs
Bundle
ELASTIC
STORE
Volume
Snapsh
Lifecycl
Feedback
English (US)
Privacy Policy
Terms of Use



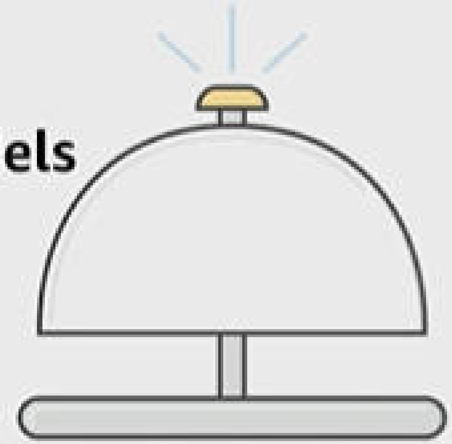
aws
Services
Resource Groups
Welcome to AWS
Oregon
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
Launch Instance
Connect
Actions
search : i-0a630d6651a49dd8c
Add filter
1 to 1 of 1
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
i-0a630d6651a49dd8c
m5.large
us-west-2a
pending
Initializing
None
ec2-34-217-109-10
Lifecycle
normal
Monitoring
basic
Alarm status
None
Kernel ID
-
RAM disk ID
-
Placement group
-
Block devices
/dev/xvda
Elastic Graphics ID
-
Elastic Inference accelerator ID
eia-ce6c06783424441ba2d9f9ee143bea25
Capacity Reservation
-
Capacity Reservation Settings
Open
Feedback
English (US)
Privacy Policy
Terms of Use

Key Takeaways

aws training and certification

Lowers inference costs by up to 75%

Supports TensorFlow, MXNet & ONNX models

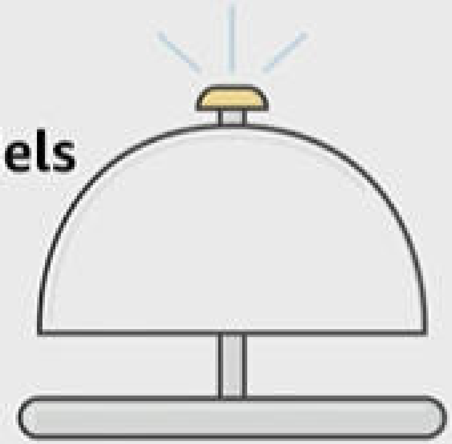


Key Takeaways

aws training and certification

Lowers inference costs by up to 75%

Supports TensorFlow, MXNet & ONNX models

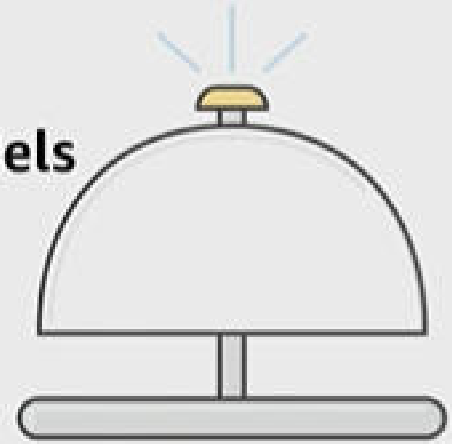


Available to all EC2 instance types.

Key Takeaways

Lowers inference costs by **up to 75%**

Supports **TensorFlow, MXNet & ONNX** models



Available to **all EC2 instance** types.

Key Takeaways

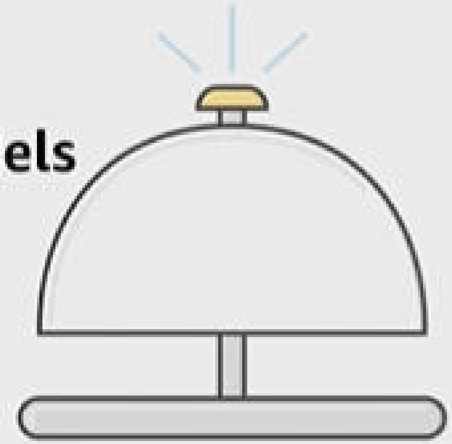
aws training and certification

Lowers inference costs by up to 75%

Supports TensorFlow, MXNet & ONNX models

Available to all EC2 instance types.

Integrated with Amazon SageMaker

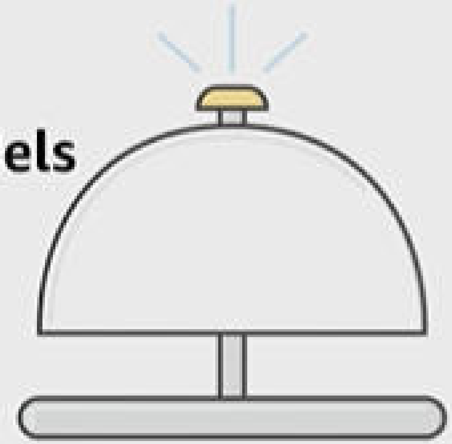


Key Takeaways

aws training and certification

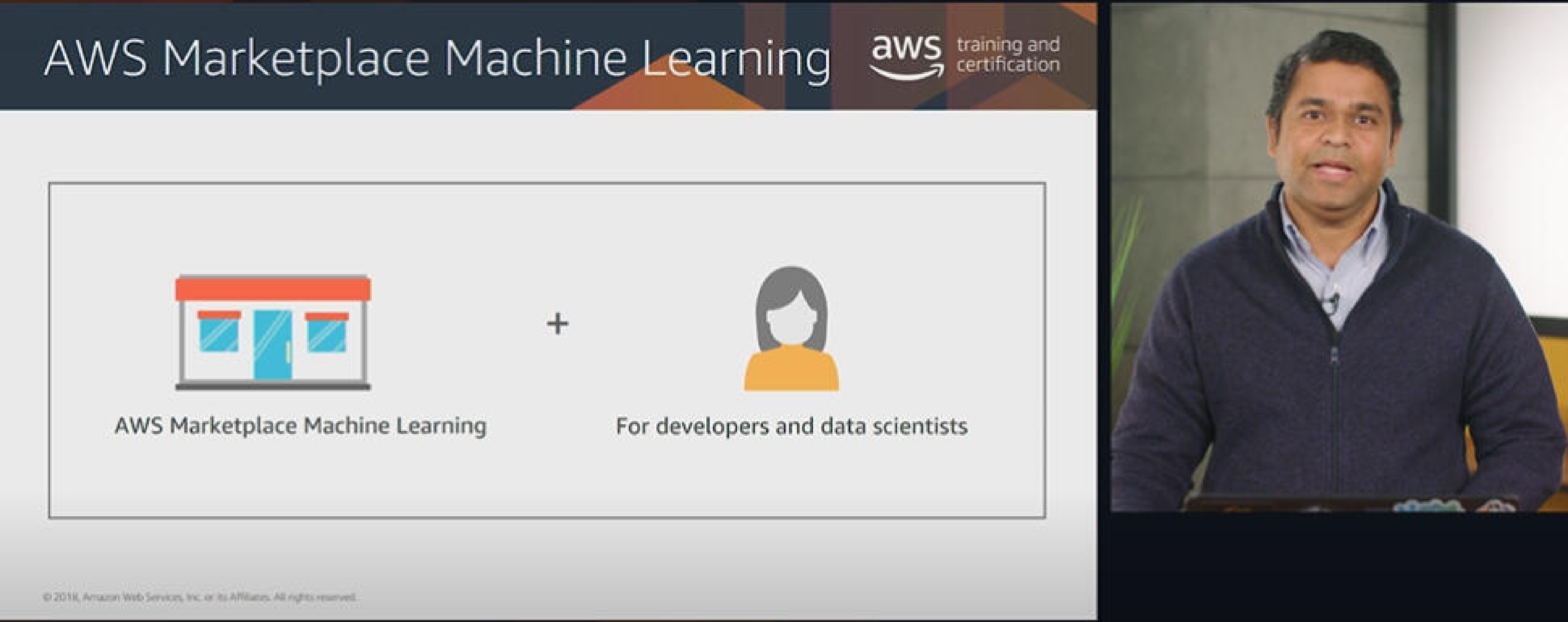
Lowers inference costs by up to 75%

Supports TensorFlow, MXNet & ONNX models



Available to all EC2 instance types.

Integrated with Amazon SageMaker

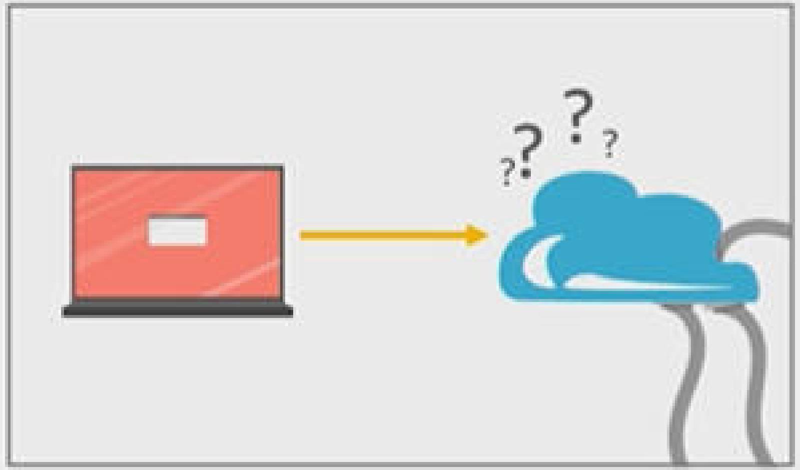

AWS Marketplace Machine Learning
aws training and certification
AWS Marketplace Machine Learning
+
For developers and data scientists

What was the status quo?

Significant time developing

What was the status quo?

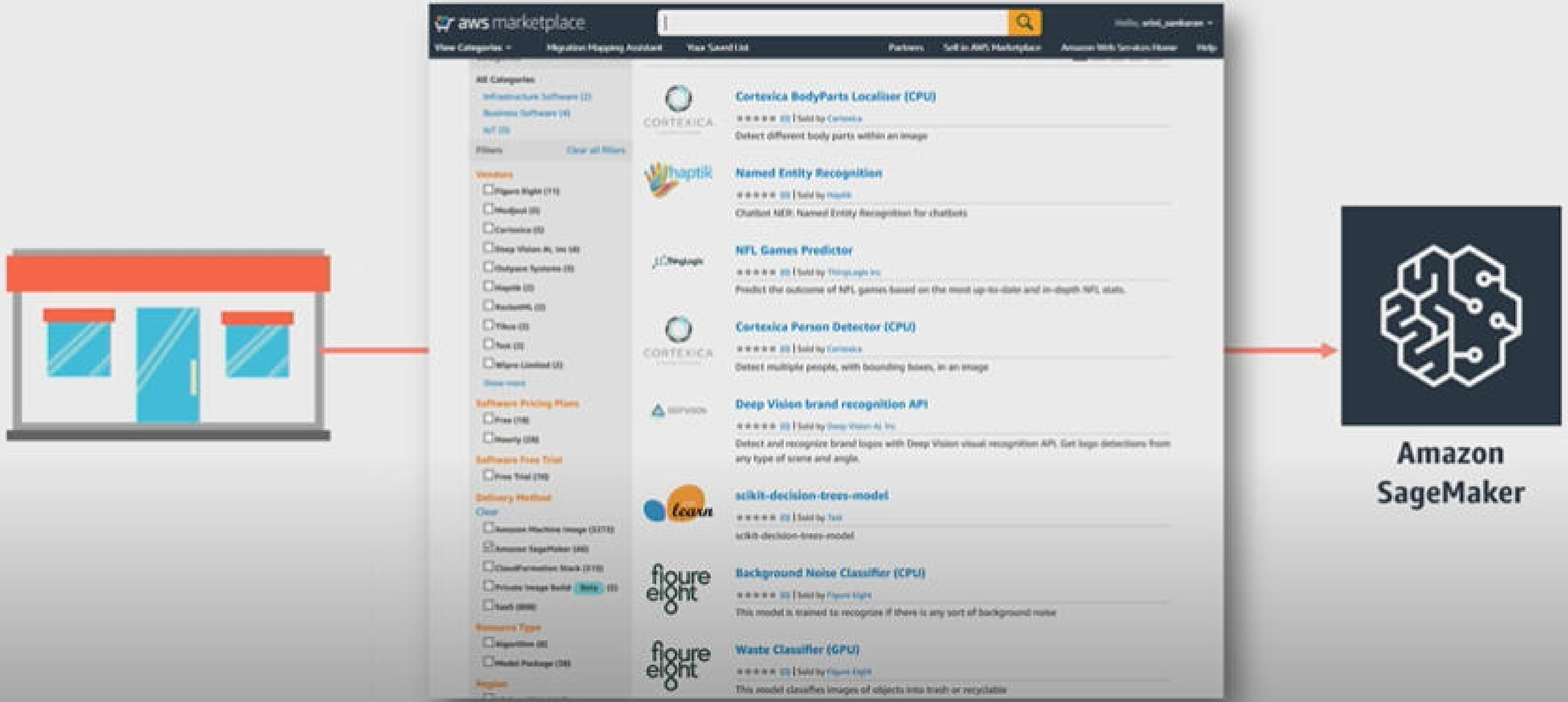
Significant time developing



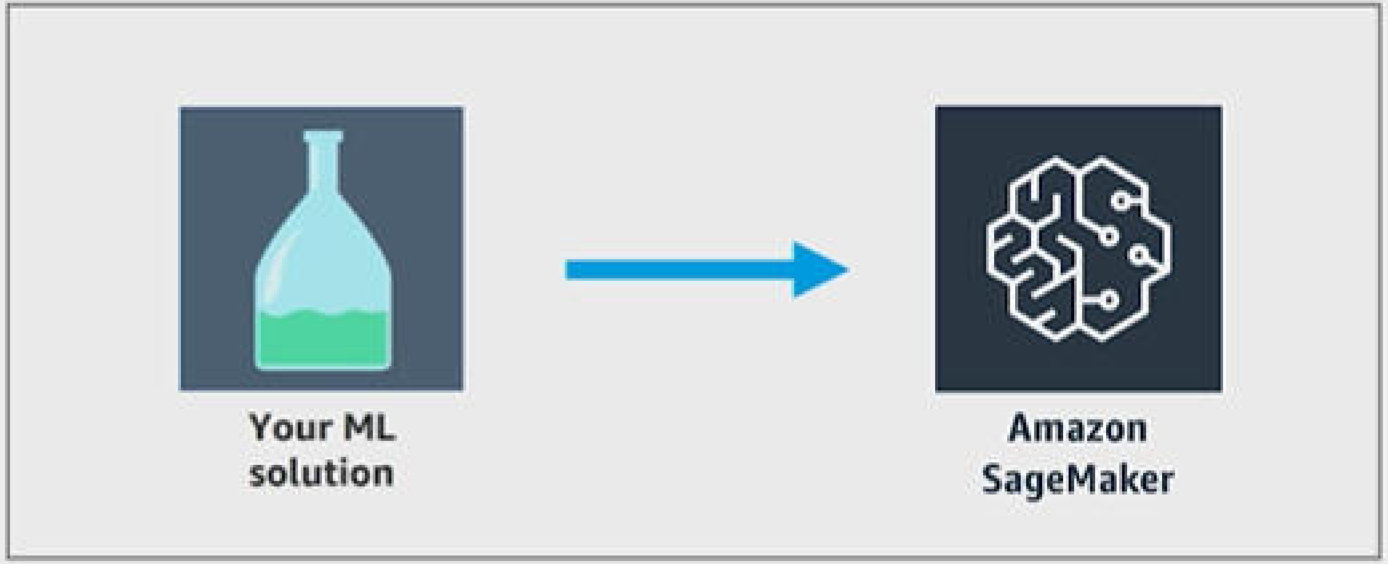
Searching the internet

How does AWS Marketplace fix this?

aws training and certification

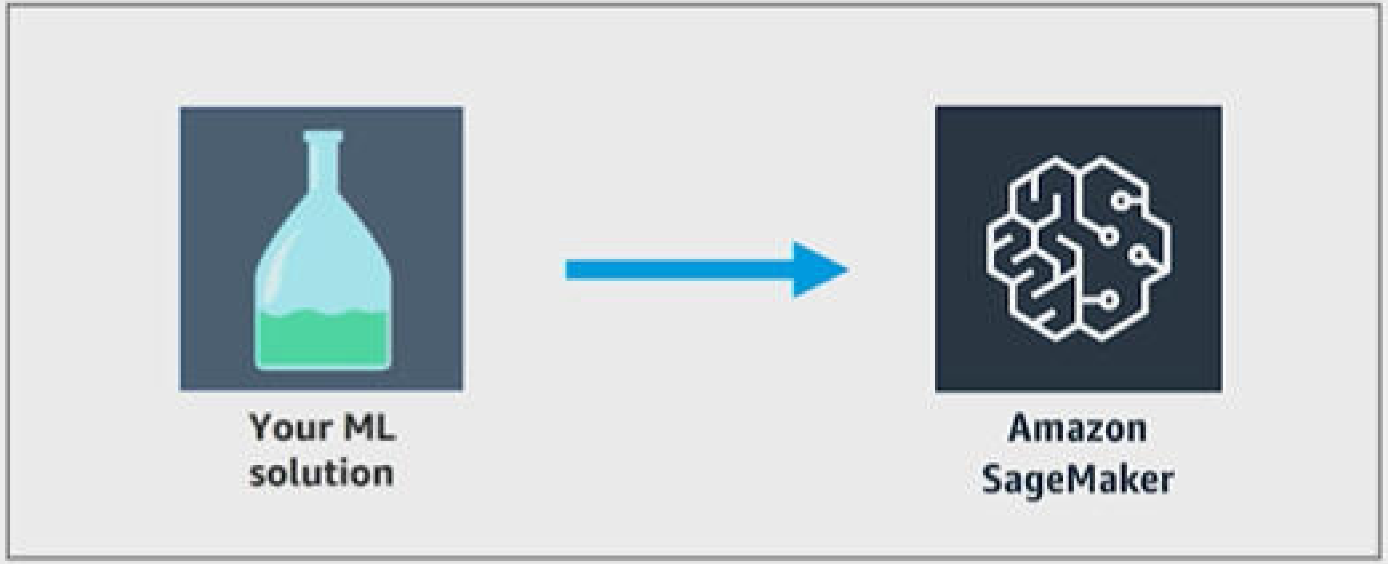


Looking to share your work?



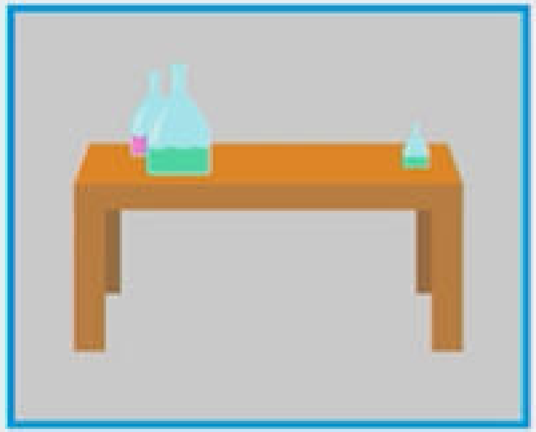
Looking to share your work?

aws training and certification

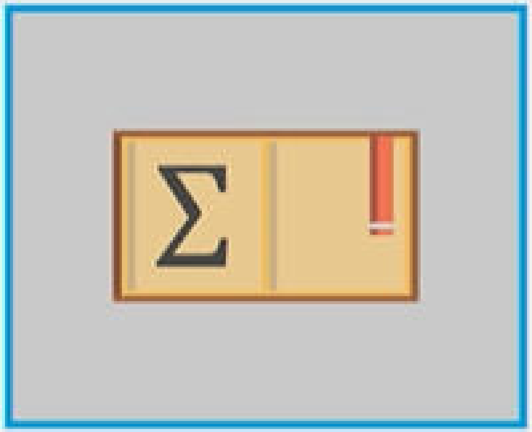


What does this mean for you?

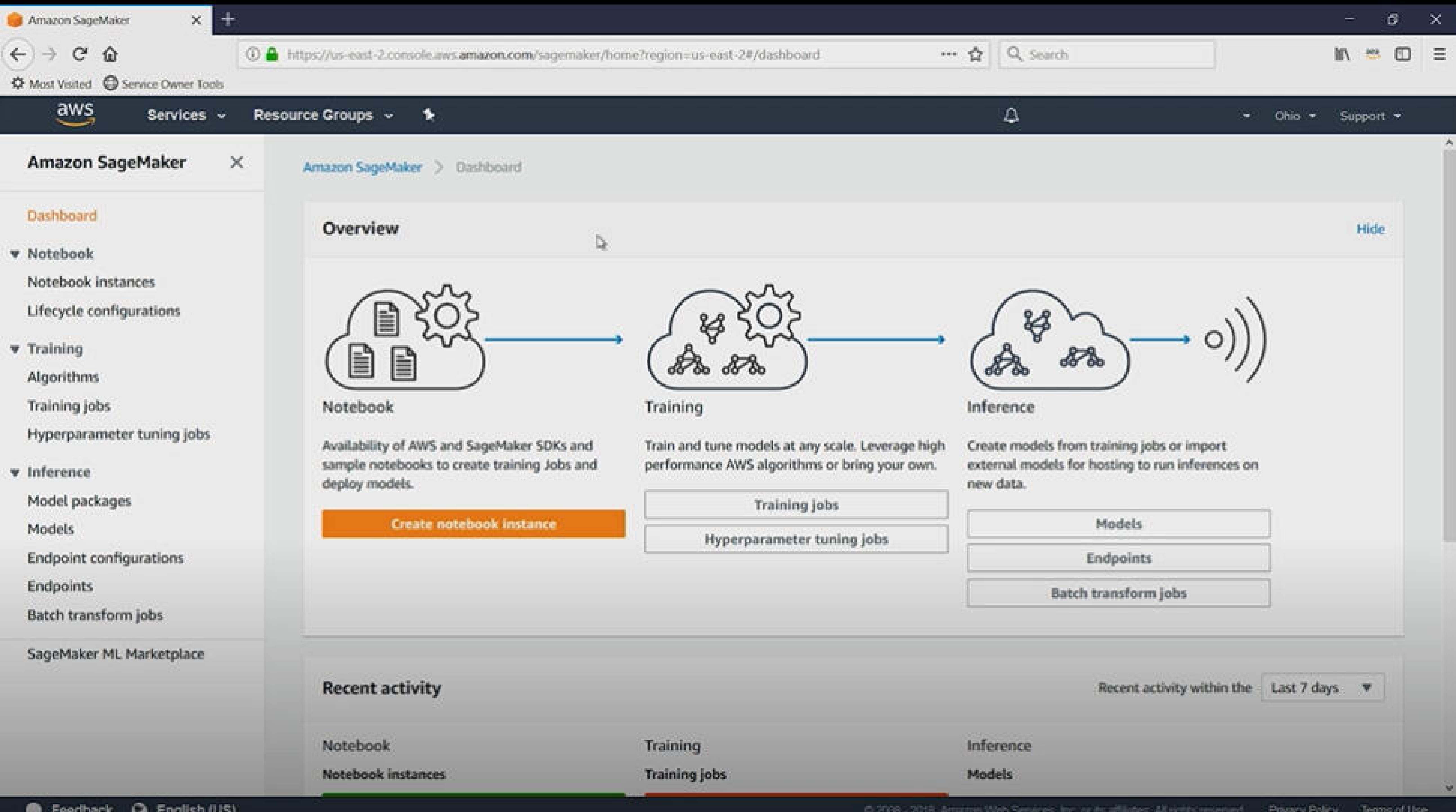
aws training and certification

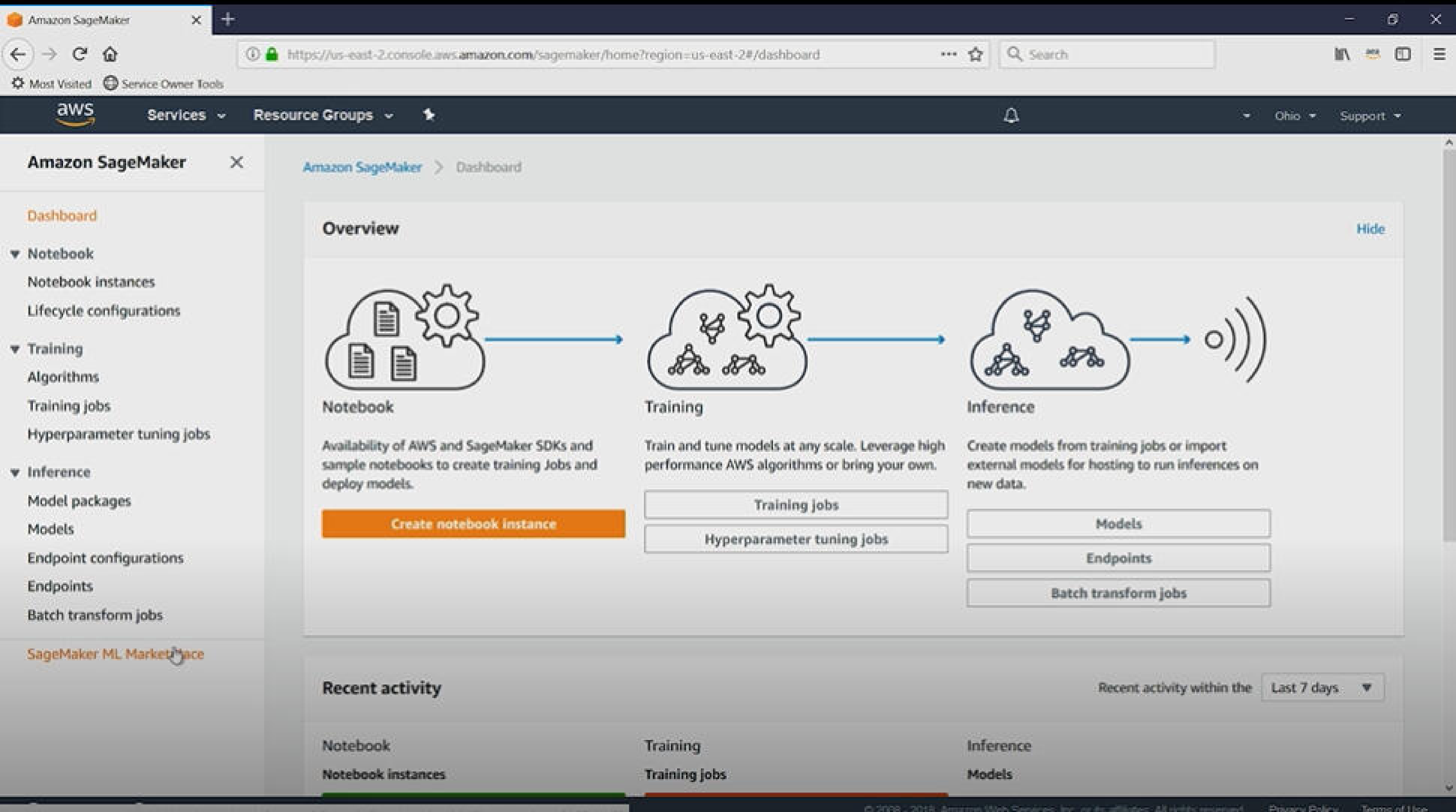


You can avoid undifferentiated heavy lifting



Large catalog of trusted and secure algorithms and models


Amazon SageMaker
https://us-east-2.console.aws.amazon.com/sagemaker/home?region=us-east-2#/dashboard
Search
Most Visited
Service Owner Tools
aws
Services
Resource Groups
Ohio
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Training
Algorithms
Training jobs
Hyperparameter tuning jobs
Inference
Model packages
Models
Endpoint configurations
Endpoints
Batch transform jobs
SageMaker ML Marketplace
Amazon SageMaker > Dashboard
Overview
Hide
Notebook
Availability of AWS and SageMaker SDKs and sample notebooks to create training Jobs and deploy models.
Create notebook instance
Training
Train and tune models at any scale. Leverage high performance AWS algorithms or bring your own.
Training jobs
Hyperparameter tuning jobs
Inference
Create models from training jobs or import external models for hosting to run inferences on new data.
Models
Endpoints
Batch transform jobs
Recent activity
Recent activity within the
Last 7 days
Notebook
Notebook instances
Training
Training jobs
Inference
Models
Feedback
English (US)


Amazon SageMaker
https://us-east-2.console.aws.amazon.com/sagemaker/home?region=us-east-2#/dashboard
Search
Most Visited
Service Owner Tools
aws
Services
Resource Groups
Ohio
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Training
Algorithms
Training jobs
Hyperparameter tuning jobs
Inference
Model packages
Models
Endpoint configurations
Endpoints
Batch transform jobs
SageMaker ML Marketplace
Amazon SageMaker > Dashboard
Overview
Hide
Notebook
Availability of AWS and SageMaker SDKs and sample notebooks to create training Jobs and deploy models.
Create notebook instance
Training
Train and tune models at any scale. Leverage high performance AWS algorithms or bring your own.
Training jobs
Hyperparameter tuning jobs
Inference
Create models from training jobs or import external models for hosting to run inferences on new data.
Models
Endpoints
Batch transform jobs
Recent activity
Recent activity within the
Last 7 days
Notebook
Notebook instances
Training
Training jobs
Inference
Models

Amazon SageMaker | AWS Marketplace: Search Res

https://aws.amazon.com/marketplace/search/results?page=1&filters=fulfillment_options&fulfillment_opt

Most Visited | Service Owner Tools

aws marketplace

Hello, srini_sankaran

View Categories | Migration Mapping Assistant | Your Saved List | Partners | Sell in AWS Marketplace | Amazon Web Services Home | Help

IoT (5)

Filters | Clear all filters

Vendors

- Figure Eight (11)
- Modjoul (5)
- Cortexica (5)
- Deep Vision AI, Inc (4)
- Outpace Systems (3)
- Haptik (2)
- RocketML (2)
- Tibco (2)
- Test (2)
- Wipro Limited (2)

Show more

Software Pricing Plans

- Free (18)
- Hourly (28)

Software Free Trial

- Free Trial (10)

Delivery Method

Clear

- Amazon Machine Image (3271)
- [x] Amazon SageMaker (46)
- CloudFormation Stack (314)
- Private Image Build Beta (5)
- SaaS (808)

(0) | Sold by Cortexica

Detect different body parts within an image

Named Entity Recognition

(0) | Sold by Haptik

Chatbot NER: Named Entity Recognition for chatbots

NFL Games Predictor

(0) | Sold by ThingLogix Inc

Predict the outcome of NFL games based on the most up-to-date and in-depth NFL stats.

Cortexica Person Detector (CPU)

(0) | Sold by Cortexica

Detect multiple people, with bounding boxes, in an image

Deep Vision brand recognition API

(0) | Sold by Deep Vision AI, Inc

Detect and recognize brand logos with Deep Vision visual recognition API. Get logo detections from any type of scene and angle.

scikit-decision-trees-model

(0) | Sold by Test

scikit-decision-trees-model

Background Noise Classifier (CPU)

(0) | Sold by Figure Eight

This model is trained to recognize if there is any sort of background noise

CORTEXICA

Cortexica Person Detector (CPU)

Continue to Subscribe

SageMaker Realtime Inference	**$0.134/host/hr** *running on ml.m5.large*
SageMaker Batch Transform	**$0.134/host/hr** *running on ml.m5.large*

○	ml.m5.12xlarge	$0.5	$3.226	$3.726
○	ml.m5.24xlarge	$0.5	$6.451	$6.951
○	ml.m4.xlarge	$0.5	$0.28	$0.78
○	ml.m4.2xlarge	$0.5	$0.56	$1.06

Usage Information

Fulfillment Methods

Amazon SageMaker

Usage Instructions:
Supported content types are image/jpeg, image/png and image/bmp.

Supported response types are application/json (default), and image/jpeg

After creating an endpoint, the AWS APIs can be used to invoke the model. For, example, you can use the AWS cli:

aws sagemaker-runtime invoke-endpoint --endpoint-name

Additional Resources

Cortexica - AI Solutions for Business

Amazon SageMaker

Amazon SageMaker is a fully-managed platform that enables developers and data scientists to quickly and easily build, train, and deploy machine learning models at any scale. With Amazon SageMaker, all the barriers and complexity that typically slow down developers who want to use machine learning are removed. The service includes

Subscribe to this software

Review the pricing terms and End User License Agreement (EULA) below. By clicking Accept Offer, you will be subscribed to this software and agree that you use of this software is subject to the pring terms and the seller's End User License Agreement (EULA) and your use of AWS services is subject to the AWS Customer Agreement

Cortexica Offer

By: Cortexica

Accept Offer

Pricing Terms

Batch Transform usage

Instance type	Batch Transform/hr
ml.c5.xlarge	$1
ml.c5.2xlarge	$1
ml.c5.4xlarge	$1
ml.c5.9xlarge	$1
ml.c5.18xlarge	$1
ml.c4.xlarge	$1

Amazon SageMaker | AWS Marketplace: Cortexica Pr

https://aws.amazon.com/marketplace/ai/procurement?productId=e7762a1b-6226-43dd-a50b-60999d870

Most Visited | Service Owner Tools

aws marketplace

Hello, srini_sankaran

View Categories | Migration Mapping Assistant | Your Saved List | Partners | Sell in AWS Marketplace | Amazon Web Services Home | Help

Product Detail > Subscribe

Subscribe to this software

Review the pricing terms and End User License Agreement (EULA) below. By clicking Accept Offer, you will be subscribed to this software and agree that you use of this software is subject to the pring terms and the seller's End User License Agreement (EULA) and your use of AWS services is subject to the AWS Customer Agreement

Your subscription request is in progress.

Cortexica Offer

Subscribing

By: Cortexica

Pricing Terms

Batch Transform usage

Instance type	Batch Transform/hr
ml.c5.xlarge	$1
ml.c5.2xlarge	$1
ml.c5.4xlarge	$1
ml.c5.9xlarge	$1

Subscribe to this software

Review the pricing terms and End User License Agreement (EULA) below. By clicking Accept Offer, you will be subscribed to this software and agree that you use of this software is subject to the pring terms and the seller's End User License Agreement (EULA) and your use of AWS services is subject to the AWS Customer Agreement

Thank you for subscribing. You can now view your product.

Cortexica Offer

Already Subscribed

By: Cortexica

Pricing Terms

Batch Transform usage

Instance type	Batch Transform/hr
ml.c5.xlarge	$1
ml.c5.2xlarge	$1
ml.c5.4xlarge	$1
ml.c5.9xlarge	$1

Amazon SageMaker | AWS Marketplace: Cortexica P...

https://aws.amazon.com/marketplace/ai/procurement?productId=e7762a1b-6226-43dd-a50b-60999d870...

Most Visited   Service Owner Tools

aws marketplace

Hello, srini_sankaran

View Categories   Migration Mapping Assistant   Your Saved List   Partners   Sell in AWS Marketplace   Amazon Web Services Home   Help

CORTEXICA

Cortexica Person Detector (CPU)

Continue to configuration

Product Detail > Subscribe

Subscribe to this software

Review the pricing terms and End User License Agreement (EULA) below. By clicking Accept Offer, you will be subscribed to this software and agree that you use of this software is subject to the pring terms and the seller's End User License Agreement (EULA) and your use of AWS services is subject to the AWS Customer Agreement

Cortexica Offer

By: Cortexica

Already Subscribed

Pricing Terms

Batch Transform usage

Instance type	Batch Transform/hr
ml.c5.xlarge	$1
ml.c5.2xlarge	$1
ml.c5.4xlarge	$1

CORTEXICA

Cortexica Person Detector (CPU)

View in SageMaker

Product Detail > Subscribe > Configure

Configure this Software

Select the Fulfillment Option, Version, and Region that you want to launch this software, then view in SageMaker to begin using it.

Configure

Fulfillment Option

Amazon Sagemaker

Software Version

1.1

Region

Select a region

Product Arn:

View in SageMaker

Amazon SageMaker | Configuration

https://aws.amazon.com/marketplace/ai/configuration?productId=e7762a1b-6226-43dd-a50b-60999d870

Most Visited | Service Owner Tools

aws marketplace

Hello, srini_sankaran

View Categories | Migration Mapping Assistant | Your Saved List | Partners | Sell in AWS Marketplace | Amazon Web Services Home | Help

CORTEXICA

Cortexica Person Detector (CPU)

View in SageMaker

Product Detail > Subscribe > Configure

Configure this Software

Select the Fulfillment Option, Version, and Region that you want to launch this software, then view in SageMaker to begin using it.

Configure

Fulfillment Option

Amazon Sagemaker

Software Version

1.1

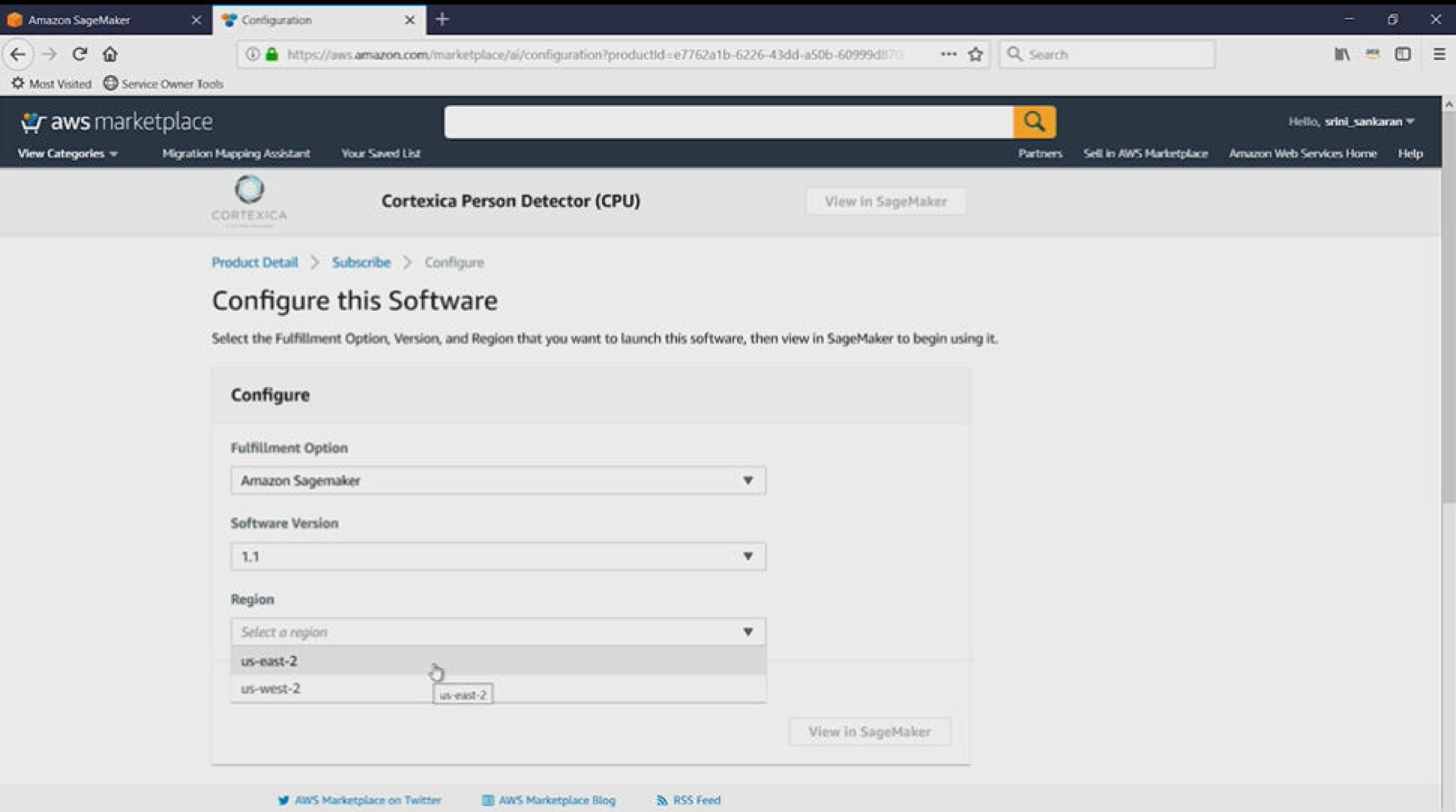
Region

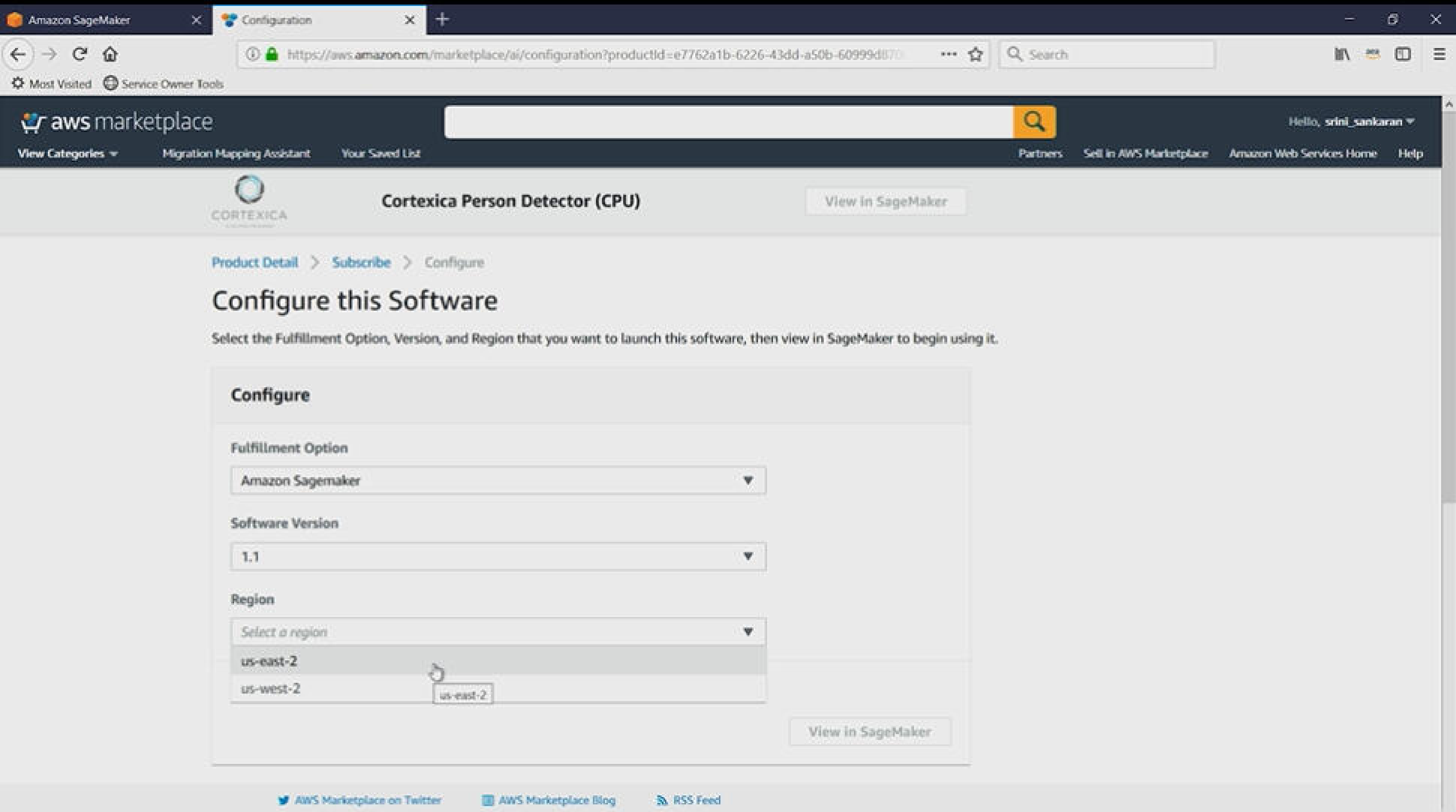
Select a region

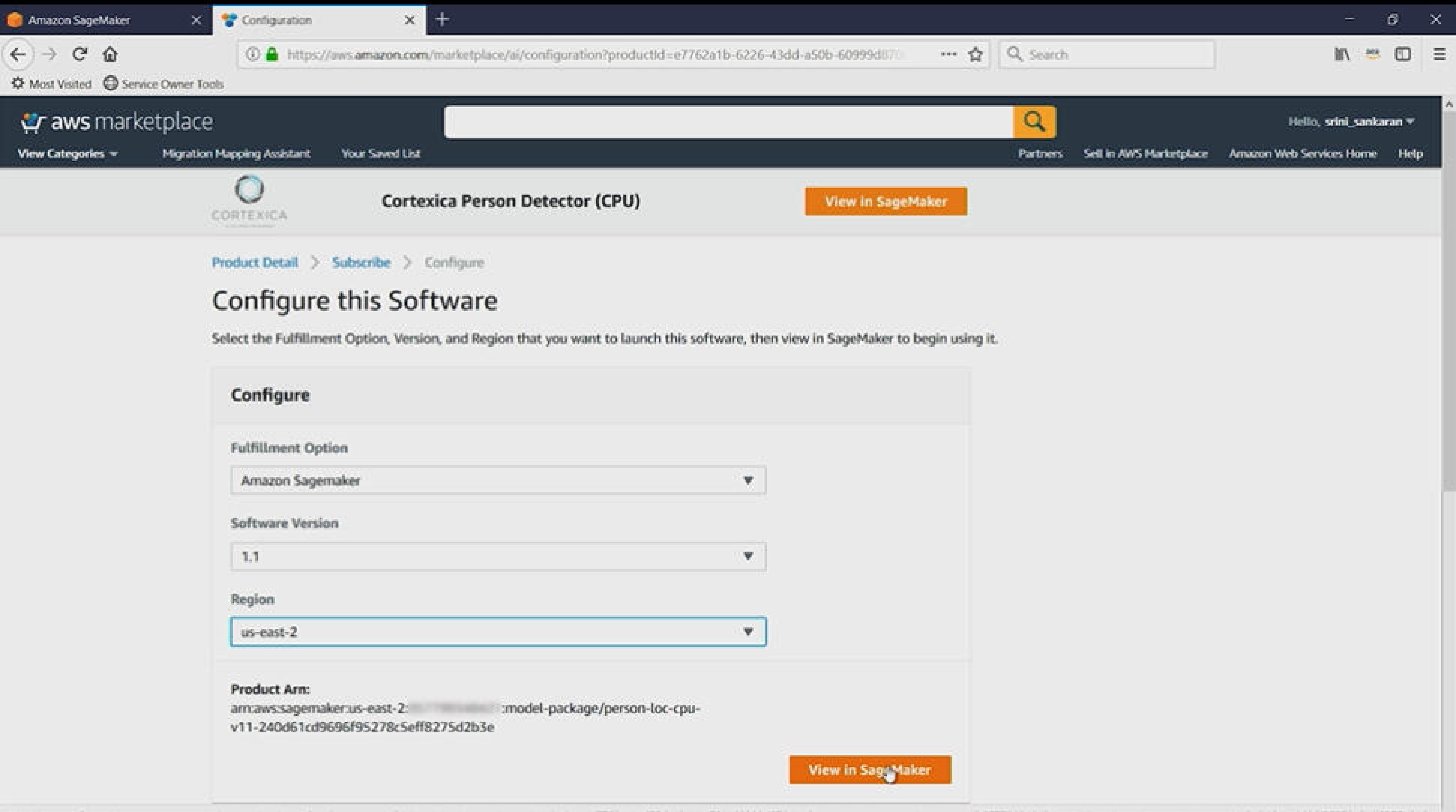
Product Arn:

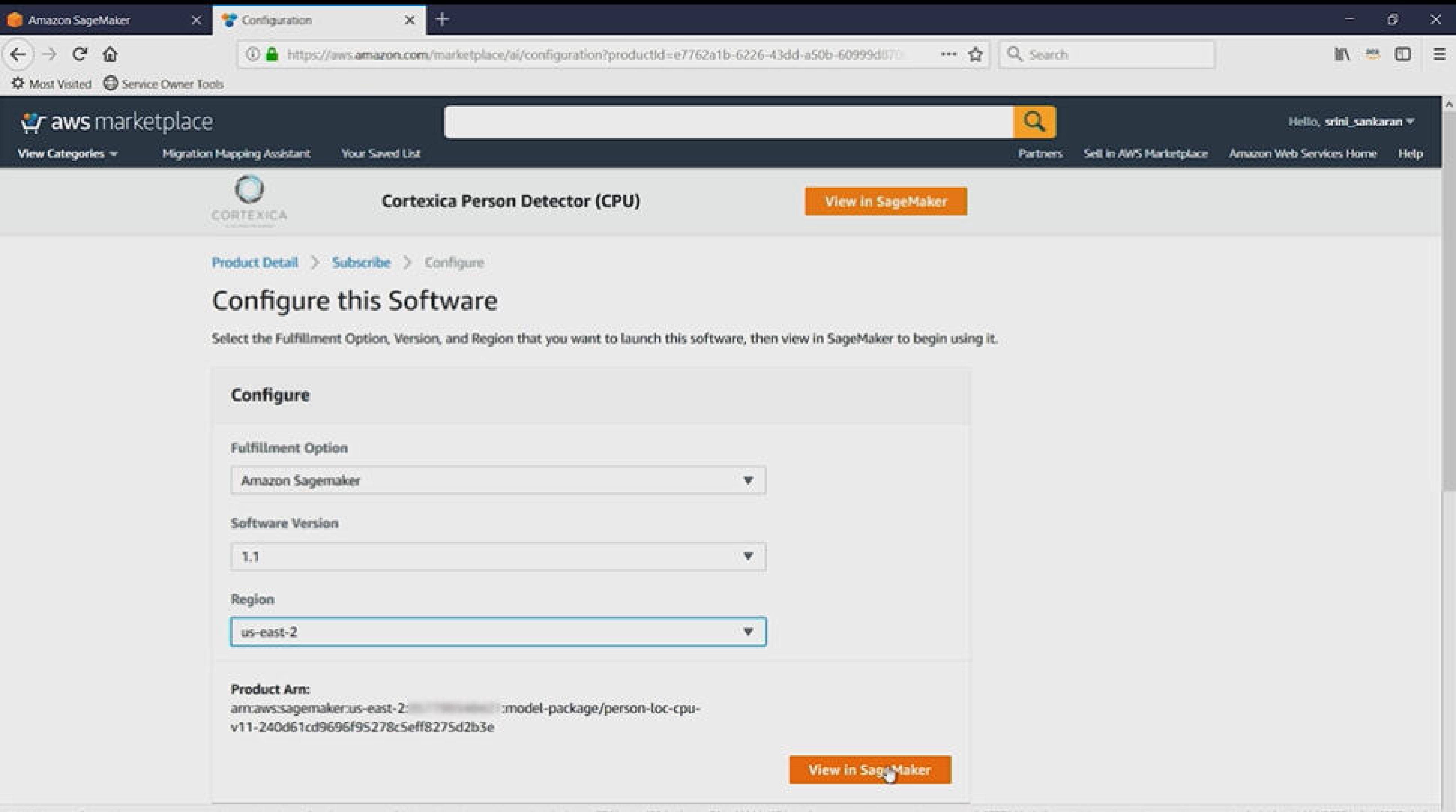
View in SageMaker

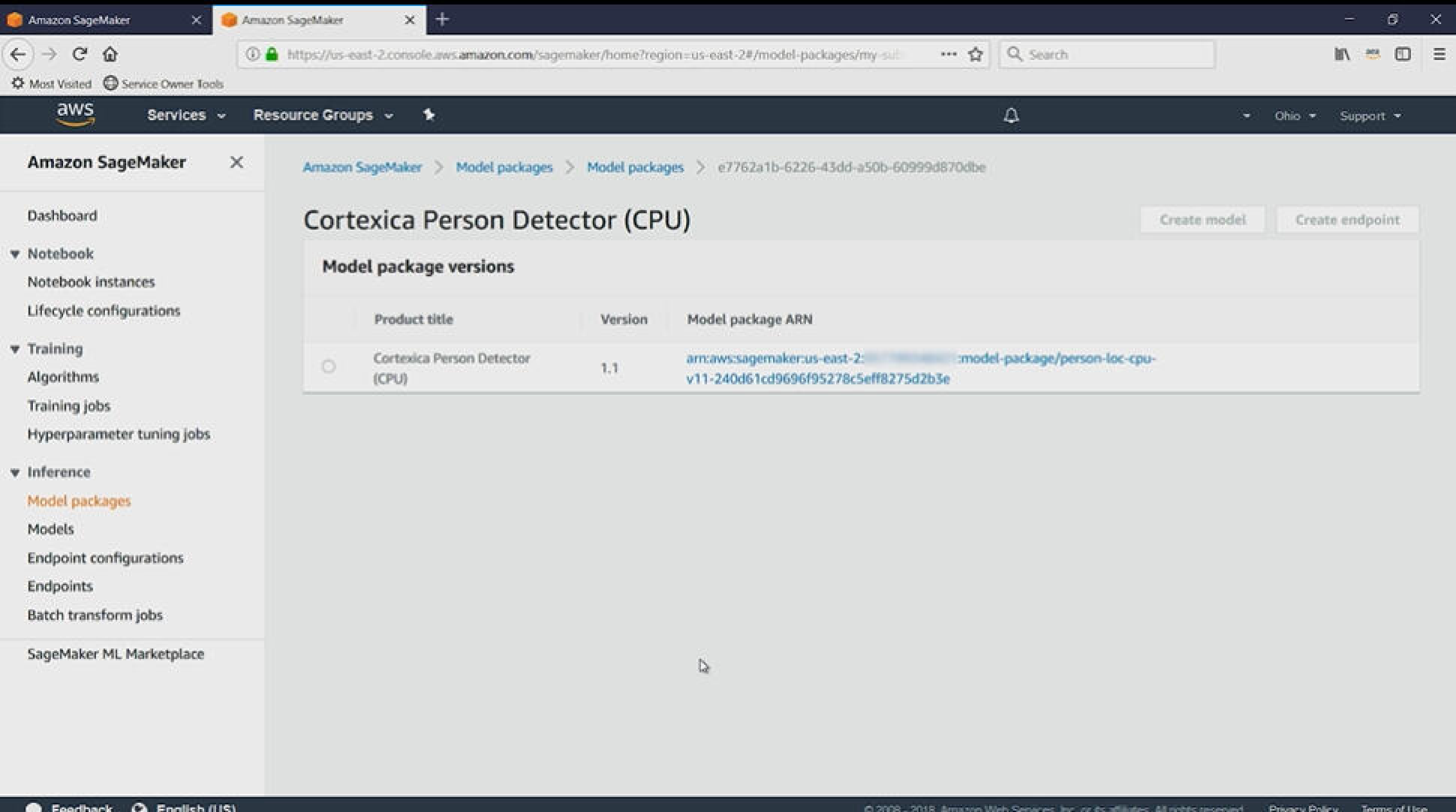
AWS Marketplace on Twitter | AWS Marketplace Blog | RSS Feed


Amazon SageMaker
Configuration
https://aws.amazon.com/marketplace/ai/configuration?productId=e7762a1b-6226-43dd-a50b-60999d8
Search
Most Visited
Service Owner Tools
aws marketplace
Hello, srini_sankaran
View Categories
Migration Mapping Assistant
Your Saved List
Partners
Sell in AWS Marketplace
Amazon Web Services Home
Help
CORTEXICA
Cortexica Person Detector (CPU)
View in SageMaker
Product Detail
Subscribe
Configure
Configure this Software
Select the Fulfillment Option, Version, and Region that you want to launch this software, then view in SageMaker to begin using it.
Configure
Fulfillment Option
Amazon Sagemaker
Software Version
1.1
Region
Select a region
us-east-2
us-west-2
us-east-2
View in SageMaker
AWS Marketplace on Twitter
AWS Marketplace Blog
RSS Feed


Amazon SageMaker
Configuration
https://aws.amazon.com/marketplace/ai/configuration?productId=e7762a1b-6226-43dd-a50b-60999d
Search
Most Visited
Service Owner Tools
aws marketplace
Hello, srini_sankaran
View Categories
Migration Mapping Assistant
Your Saved List
Partners
Sell in AWS Marketplace
Amazon Web Services Home
Help
CORTEXICA
Cortexica Person Detector (CPU)
View in SageMaker
Product Detail > Subscribe > Configure
Configure this Software
Select the Fulfillment Option, Version, and Region that you want to launch this software, then view in SageMaker to begin using it.
Configure
Fulfillment Option
Amazon Sagemaker
Software Version
1.1
Region
Select a region
us-east-2
us-west-2
us-east-2
View in SageMaker
AWS Marketplace on Twitter
AWS Marketplace Blog
RSS Feed


Amazon SageMaker
Configuration
https://aws.amazon.com/marketplace/ai/configuration?productId=e7762a1b-6226-43dd-a50b-60999d
Search
Most Visited
Service Owner Tools
aws marketplace
Hello, srini_sankaran
View Categories
Migration Mapping Assistant
Your Saved List
Partners
Sell in AWS Marketplace
Amazon Web Services Home
Help
CORTEXICA
Cortexica Person Detector (CPU)
View in SageMaker
Product Detail
Subscribe
Configure
Configure this Software
Select the Fulfillment Option, Version, and Region that you want to launch this software, then view in SageMaker to begin using it.
Configure
Fulfillment Option
Amazon Sagemaker
Software Version
1.1
Region
us-east-2
Product Arn:
arn:aws:sagemaker:us-east-2: :model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d2b3e
View in SageMaker


Amazon SageMaker
Configuration
https://aws.amazon.com/marketplace/ai/configuration?productId=e7762a1b-6226-43dd-a50b-60999d87
Search
Most Visited
Service Owner Tools
aws marketplace
Hello, srini_sankaran
View Categories
Migration Mapping Assistant
Your Saved List
Partners
Sell in AWS Marketplace
Amazon Web Services Home
Help
CORTEXICA
Cortexica Person Detector (CPU)
View in SageMaker
Product Detail > Subscribe > Configure
Configure this Software
Select the Fulfillment Option, Version, and Region that you want to launch this software, then view in SageMaker to begin using it.
Configure
Fulfillment Option
Amazon Sagemaker
Software Version
1.1
Region
us-east-2
Product Arn:
arn:aws:sagemaker:us-east-2: :model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d2b3e
View in SageMaker


Amazon SageMaker
Amazon SageMaker
https://us-east-2.console.aws.amazon.com/sagemaker/home?region=us-east-2#/model-packages/my-sub
Most Visited
Service Owner Tools
Services
Resource Groups
Ohio
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Training
Algorithms
Training jobs
Hyperparameter tuning jobs
Inference
Model packages
Models
Endpoint configurations
Endpoints
Batch transform jobs
SageMaker ML Marketplace
Amazon SageMaker > Model packages > Model packages > e7762a1b-6226-43dd-a50b-60999d870dbe
Cortexica Person Detector (CPU)
Create model
Create endpoint
Model package versions
Product title
Version
Model package ARN
Cortexica Person Detector (CPU)
1.1
arn:aws:sagemaker:us-east-2: :model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d2b3e
Feedback
English (US)
Privacy Policy
Terms of Use

Amazon SageMaker
Amazon SageMaker

https://us-east-2.console.aws.amazon.com/sagemaker/home?region=us-east-2#/model-packages/my-su

Most Visited · Service Owner Tools

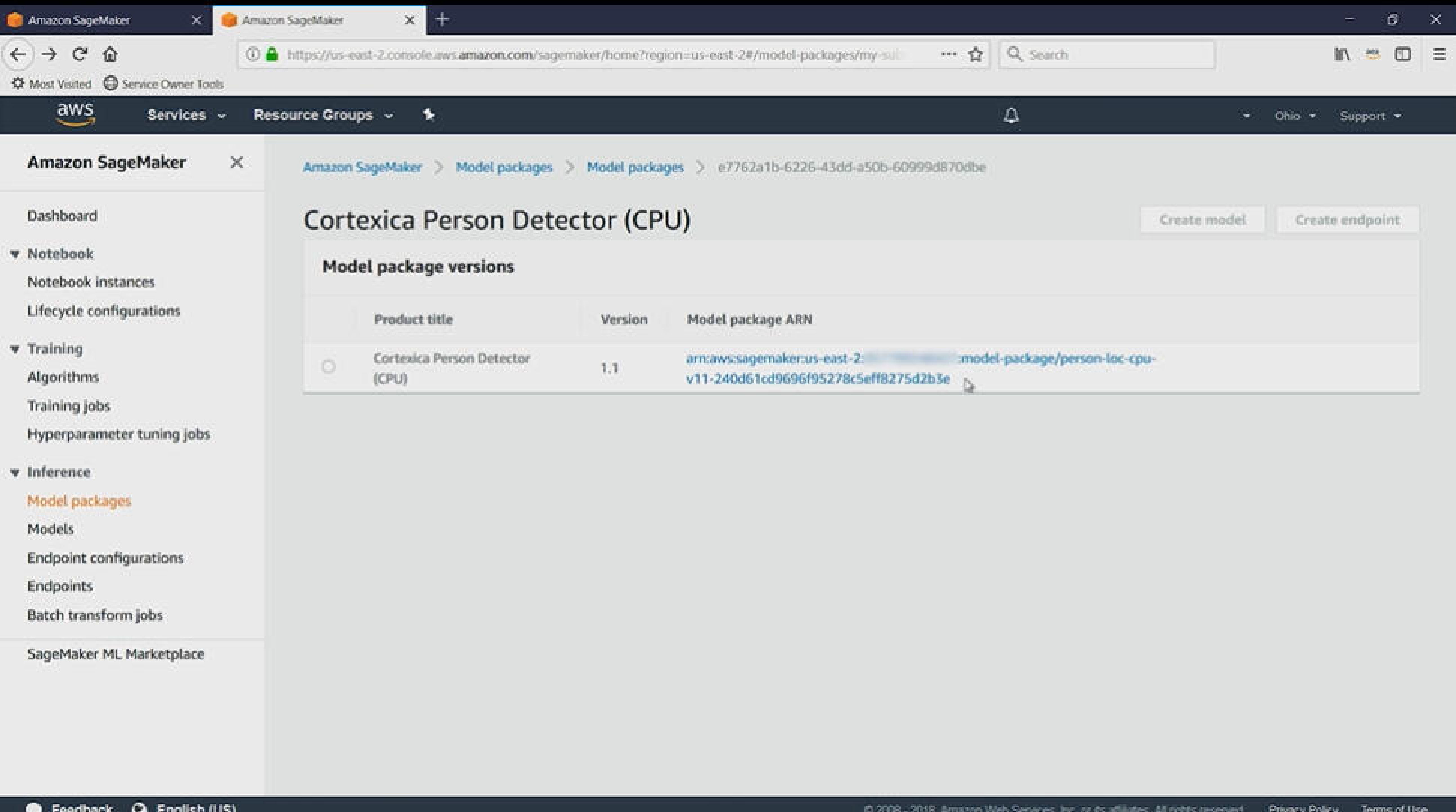
aws · Services · Resource Groups · Ohio · Support

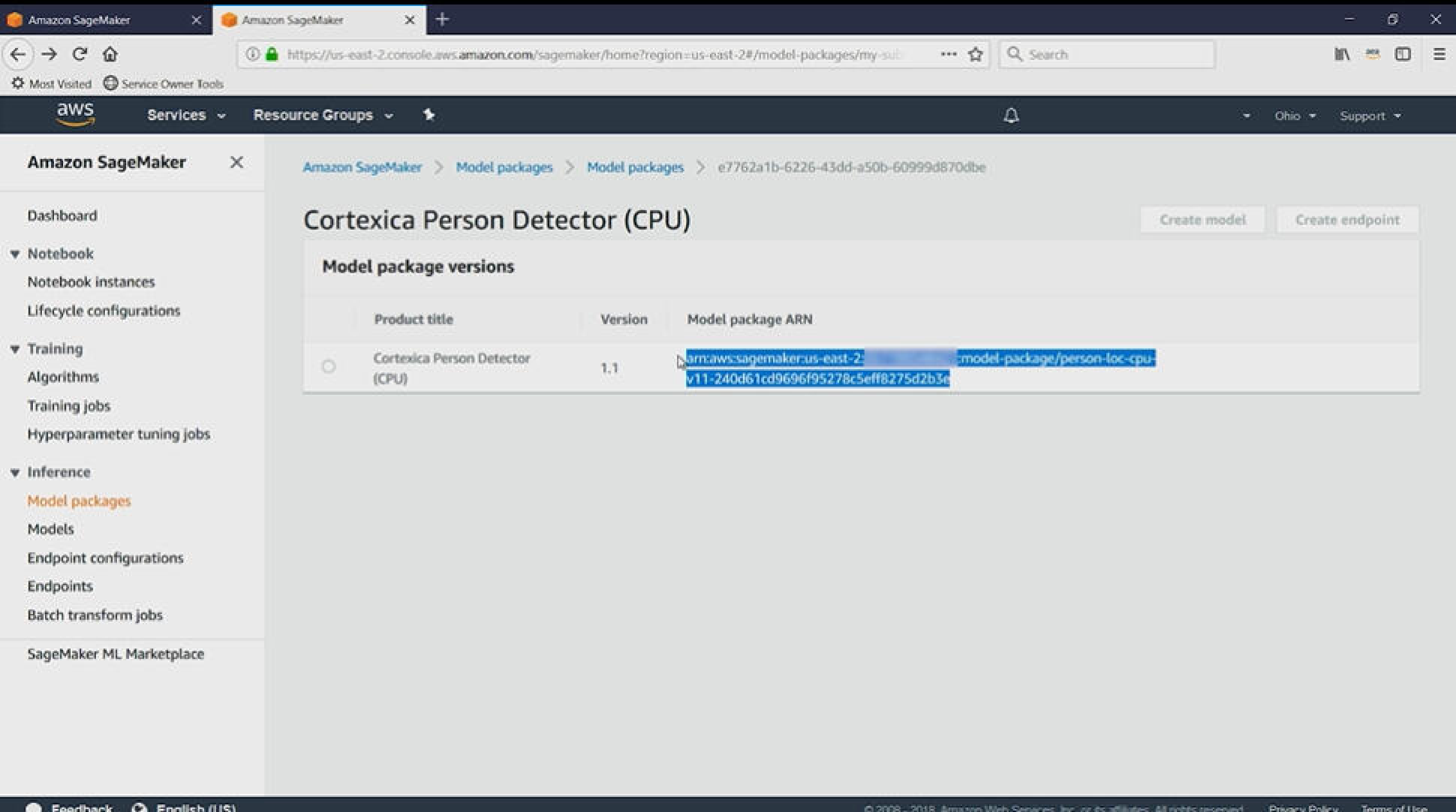
Amazon SageMaker

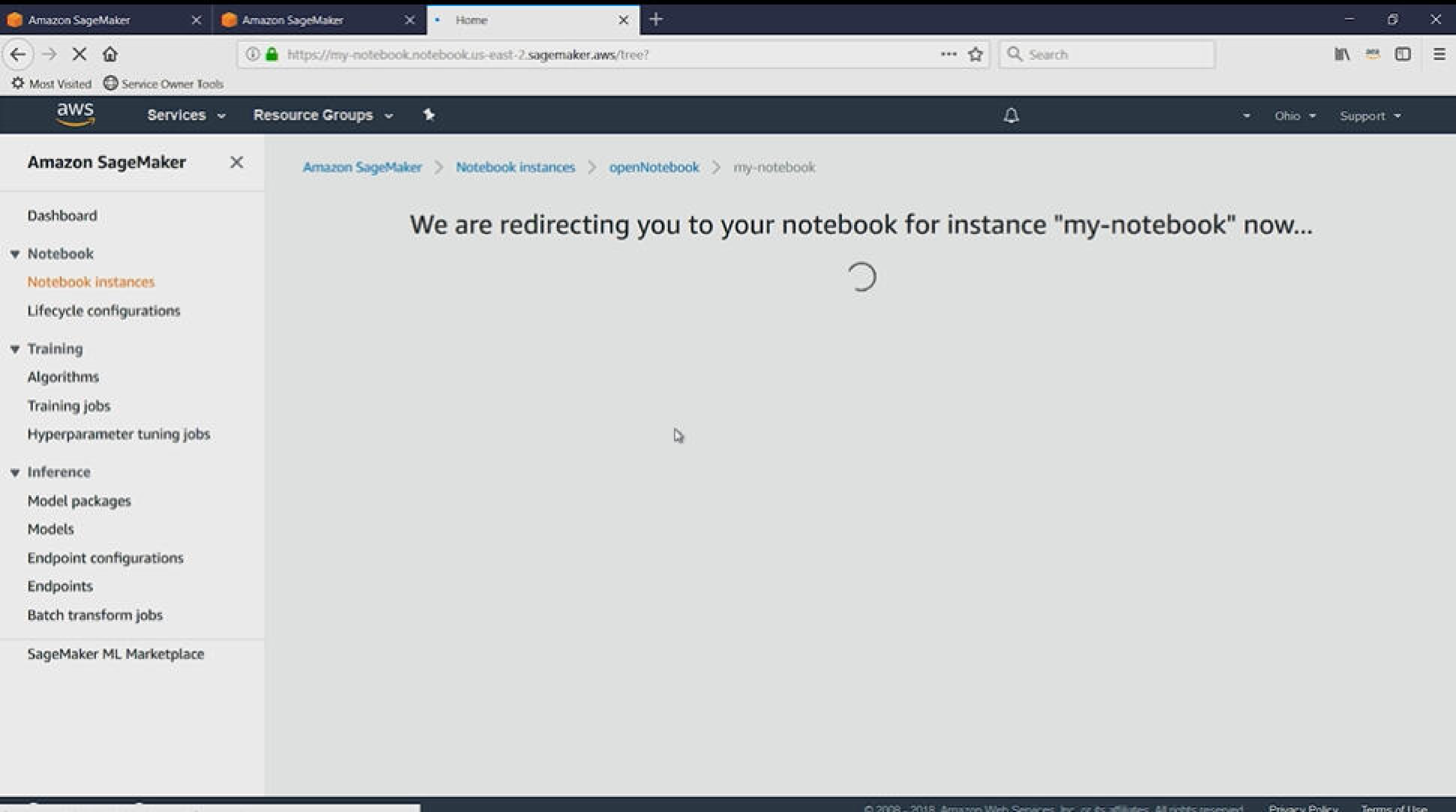
- Dashboard
- ▼ Notebook
  - Notebook instances
  - Lifecycle configurations
- ▼ Training
  - Algorithms
  - Training jobs
  - Hyperparameter tuning jobs
- ▼ Inference
  - Model packages
  - Models
  - Endpoint configurations
  - Endpoints
  - Batch transform jobs
- SageMaker ML Marketplace

Amazon SageMaker > Model packages > Model packages > e7762a1b-6226-43dd-a50b-60999d870dbe > arn%3Aaws%3Asagemaker%3Aus-east-2%3A %3Amodel-package%2Fperson-loc-cpu-v11-240d61cd9696f95278c5eff8275d2b3e

Feedback · English (US) · Privacy Policy · Terms of Use


Amazon SageMaker
Amazon SageMaker
https://us-east-2.console.aws.amazon.com/sagemaker/home?region=us-east-2#/model-packages/my-su
Search
Most Visited
Service Owner Tools
aws
Services
Resource Groups
Ohio
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Training
Algorithms
Training jobs
Hyperparameter tuning jobs
Inference
Model packages
Models
Endpoint configurations
Endpoints
Batch transform jobs
SageMaker ML Marketplace
Amazon SageMaker > Model packages > Model packages > e7762a1b-6226-43dd-a50b-60999d870dbe
Cortexica Person Detector (CPU)
Create model
Create endpoint
Model package versions
Product title
Version
Model package ARN
Cortexica Person Detector (CPU)
1.1
arn:aws:sagemaker:us-east-2:
:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d2b3e
Feedback
English (US)
Privacy Policy
Terms of Use


Amazon SageMaker
Amazon SageMaker
https://us-east-2.console.aws.amazon.com/sagemaker/home?region=us-east-2#/model-packages/my-sub
Search
Most Visited
Service Owner Tools
aws
Services
Resource Groups
Ohio
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Training
Algorithms
Training jobs
Hyperparameter tuning jobs
Inference
Model packages
Models
Endpoint configurations
Endpoints
Batch transform jobs
SageMaker ML Marketplace
Amazon SageMaker > Model packages > Model packages > e7762a1b-6226-43dd-a50b-60999d870dbe
Cortexica Person Detector (CPU)
Create model
Create endpoint
Model package versions
Product title
Version
Model package ARN
Cortexica Person Detector (CPU)
1.1
arn:aws:sagemaker:us-east-2: :model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d2b3e
Feedback
English (US)
Privacy Policy
Terms of Use


Amazon SageMaker
Amazon SageMaker
Home
https://my-notebook.notebook.us-east-2.sagemaker.aws/tree?
Search
Most Visited
Service Owner Tools
aws
Services
Resource Groups
Ohio
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Training
Algorithms
Training jobs
Hyperparameter tuning jobs
Inference
Model packages
Models
Endpoint configurations
Endpoints
Batch transform jobs
SageMaker ML Marketplace
Amazon SageMaker > Notebook instances > openNotebook > my-notebook
We are redirecting you to your notebook for instance "my-notebook" now...
Privacy Policy
Terms of Use

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter **Cortexica-person-detector** Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

In [ ]:
```
#import necessary libraries
import boto3
import urllib.request
import base64
import json
import os
from sagemaker import get_execution_role
import time
import sagemaker as sagemaker
```

In [ ]:
```
%%sh

aws s3 sync s3://buyer-sm-opt-prm-prvw-1022 ./
chmod +x ./sdk/install-cli.sh
./sdk/install-cli.sh
```

In [ ]:
```
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initialising other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[illegible]:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

In [ ]:
```
#import necessary libraries
import boto3
import urllib.request
import base64
import json
import os
from sagemaker import get_execution_role
import time
import sagemaker as sagemaker
```

In [ ]:
```
%%sh

aws s3 sync s3://buyer-sm-opt-prm-prvw-1022 ./
chmod +x ./sdk/install-cli.sh
./sdk/install-cli.sh
```

In [ ]:
```
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initialising other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[redacted]:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

In [ ]:

```python
#import necessary libraries
import boto3
import urllib.request
import base64
import json
import os
from sagemaker import get_execution_role
import time
import sagemaker as sagemaker
```

In [ ]:

```
%%sh

aws s3 sync s3://buyer-sm-opt-prm-prvw-1022 ./
chmod +x ./sdk/install-cli.sh
./sdk/install-cli.sh
```

In [ ]:

```python
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initialising other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[redacted]:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited   Service Owner Tools

jupyter **Cortexica-person-detector** Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help   Trusted   conda_python3

Run   Code

In [ ]:
```python
#import necessary libraries
import boto3
import urllib.request
import base64
import json
import os
from sagemaker import get_execution_role
import time
import sagemaker as sagemaker
```

In [ ]:
```
%%sh

aws s3 sync s3://buyer-sm-opt-prm-prvw-1022 ./
chmod +x ./sdk/install-cli.sh
./sdk/install-cli.sh
```

In [ ]:
```python
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initialising other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[illegible]:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter **Cortexica-person-detector** Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

In [1]:
```
#import necessary libraries
import boto3
import urllib.request
import base64
import json
import os
from sagemaker import get_execution_role
import time
import sagemaker as sagemaker
```

In [ ]:
```
%%sh

aws s3 sync s3://buyer-sm-opt-prm-prvw-1022 ./
chmod +x ./sdk/install-cli.sh
./sdk/install-cli.sh
```

In [ ]:
```
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initialising other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[redacted]:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted conda_python3

Run Code

```
import json
import os
from sagemaker import get_execution_role
import time
import sagemaker as sagemaker
```

In [2]:

```
%%sh

aws s3 sync s3://buyer-sm-opt-prm-prvw-1022 ./
chmod +x ./sdk/install-cli.sh
./sdk/install-cli.sh
```

```
SageMaker with Marketplace features is now installed to use from boto3 and AWS CLI
```

```
+ aws configure add-model --service-model file:///home/ec2-user/SageMaker/sdk/sagemaker-2017-07-24.normal.json --serv
ice-name sagemakermp
+ echo 'SageMaker with Marketplace features is now installed to use from boto3 and AWS CLI'
```

In [ ]:

```
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initializing other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[illegible]:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
SageMaker with Marketplace features is now installed to use from boto3 and AWS CLI
```

```
+ aws configure add-model --service-model file:///home/ec2-user/SageMaker/sdk/sagemaker-2017-07-24.normal.json --serv
ice-name sagemakermp
+ echo 'SageMaker with Marketplace features is now installed to use from boto3 and AWS CLI'
```

In [ ]:

```
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initializing other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[redacted]:model-package/person-loc-cpu-v11-240d61cd9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

In [ ]:

```
#First create a model from the model-package

create_model_response = sagemakermp.create_model(
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
+ echo 'SageMaker with Marketplace features is now installed to use from boto3 and AWS CLI'
```

In [3]:

```
#Initializing BOTO3 client for Amazon SageMaker Service and Amazon SageMaker Runtime, initializing other variables

timestamp = time.strftime('-%Y-%m-%d-%H-%M-%S', time.gmtime())
runtime = boto3.client('sagemaker-runtime')
sagemakermp = boto3.client('sagemakermp', region_name='us-east-2', endpoint_url="https://sagemaker.us-east-2.amazonaws.c
session = sagemaker.Session()

#Specify model package from which end-point needs to be created
modelPackageName='arn:aws:sagemaker:us-east-2:[redacted]:model-package/person-loc-cpu-v11-240d61od9696f95278c5eff8275d
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

In [ ]:

```
#First create a model from the model-package

create_model_response = sagemakermp.create_model(
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
)
create_model_response
```

In [ ]:

```
#First create an Endpoint configuration
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```python
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

In [ ]:

```python
#First create a model from the model-package

create_model_response = sagemakermp.create_model(
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
)
create_model_response
```

In [ ]:

```python
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
            'ModelName': modelName,
            'InitialInstanceCount': 1,
            'InstanceType': 'ml.c5.xlarge',
            'InitialVariantWeight': 1
        },
    ],
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
bucket='sm-testing-ml'
#sm-testing-ml

role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

In [ ]:

```
#First create a model from the model-package

create_model_response = sagemakermp.create_model(
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
)
create_model_response
```

In [ ]:

```
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
            'ModelName': modelName,
            'InitialInstanceCount': 1,
            'InstanceType': 'ml.c5.xlarge',
            'InitialVariantWeight': 1
        },
    ],
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter **Cortexica-person-detector** Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

In [4]:

```
#First create a model from the model-package

create_model_response = sagemakermp.create_model(
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
)
create_model_response
```

Out[4]:

```
{'ModelArn': 'arn:aws:sagemaker:us-east-2:[illegible]:model/cortexicapersondetector-model-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '290c6ce2-1513-428f-8727-4b74ac2a3866',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '290c6ce2-1513-428f-8727-4b74ac2a3866',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '111',
   'date': 'Wed, 07 Nov 2018 19:20:20 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:

```
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

```
role = get_execution_role()
modelName='CortexicaPersonDetector-model'+timestamp
endPointConfigName='CortexicaPD-endpoint-config'+timestamp
endPointName='CortexicaPersonDetector-endpoint'+timestamp
batch_job_name='CortexicaPersonDetector-batch-job'+timestamp
```

In [4]:

```
#First create a model from the model-package

create_model_response = sagemakermp.create_model(
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
)
create_model_response
```

Out[4]:

```
{'ModelArn': 'arn:aws:sagemaker:us-east-2:[redacted]:model/cortexicapersondetector-model-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '290c6ce2-1513-428f-8727-4b74ac2a3866',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '290c6ce2-1513-428f-8727-4b74ac2a3866',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '111',
   'date': 'Wed, 07 Nov 2018 19:20:20 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:

```
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```python
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
)
create_model_response
```

Out[4]:

```
{'ModelArn': 'arn:aws:sagemaker:us-east-2:[illegible]:model/cortexicapersondetector-model-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '290c6ce2-1513-428f-8727-4b74ac2a3866',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '290c6ce2-1513-428f-8727-4b74ac2a3866',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '111',
   'date': 'Wed, 07 Nov 2018 19:20:20 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:

```python
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
            'ModelName': modelName,
            'InitialInstanceCount': 1,
            'InstanceType': 'ml.c5.xlarge',
            'InitialVariantWeight': 1
        },
    ],
    Tags=[
        {
            'Key': 'costcenter',
            'Value': '115'
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
    ModelName=modelName,
    Containers=[
        {
            'ModelPackageName': modelPackageName
        }
    ],
    ExecutionRoleArn=role,
    EnableNetworkIsolation=True
)
create_model_response
```

Out[4]:


```
{'ModelArn': 'arn:aws:sagemaker:us-east-2:[redacted]:model/cortexicapersondetector-model-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '290c6ce2-1513-428f-8727-4b74ac2a3866',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '290c6ce2-1513-428f-8727-4b74ac2a3866',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '111',
   'date': 'Wed, 07 Nov 2018 19:20:20 GMT'},
  'RetryAttempts': 0}}
```


In [ ]:

```
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
            'ModelName': modelName,
            'InitialInstanceCount': 1,
            'InstanceType': 'ml.c5.xlarge',
            'InitialVariantWeight': 1
        },
    ],
    Tags=[
        {
            'Key': 'costcenter',
            'Value': '115'
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
'content-type': 'application/x-amz-json-1.1',
'content-length': '111',
'date': 'Wed, 07 Nov 2018 19:20:20 GMT'},
'RetryAttempts': 0}}
```

In [ ]:

```
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
            'ModelName': modelName,
            'InitialInstanceCount': 1,
            'InstanceType': 'ml.c5.xlarge',
            'InitialVariantWeight': 1
        },
    ],
    Tags=[
        {
            'Key': 'costcenter',
            'Value': '115'
        }
    ])
create_endpoint_config_response
```

In [ ]:

```
#First create an Endpoint
#endPointName='my-endpoint'+timestamp

create_endpoint_response=sagemakermp.create_endpoint(
    EndpointName=endPointName,
    EndpointConfigName=endPointConfigName,
    Tags=[{
            'Key': 'test-type',
            'Value': 'cortexica-model-testing'
        }])
create_endpoint_response
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
  'content-type': 'application/x-amz-json-1.1',
  'content-length': '111',
  'date': 'Wed, 07 Nov 2018 19:20:20 GMT'},
 'RetryAttempts': 0}}
```

In [5]:

```
#First create an Endpoint configuration

create_endpoint_config_response = sagemakermp.create_endpoint_config(
    EndpointConfigName=endPointConfigName,
    ProductionVariants=[
        {
            'VariantName': 'default-variant-name',
            'ModelName': modelName,
            'InitialInstanceCount': 1,
            'InstanceType': 'ml.c5.xlarge',
            'InitialVariantWeight': 1
        },
    ],
    Tags=[
        {
            'Key': 'costcenter',
            'Value': '115'
        }
    ])
create_endpoint_config_response
```

Out[5]:

```
{'EndpointConfigArn': 'arn:aws:sagemaker:us-east-2:[illegible]:endpoint-config/cortexicapd-endpoint-config-2018-11-0
7-19-20-02',
 'ResponseMetadata': {'RequestId': 'a2be6669-f1d9-4209-bdf6-804d9325fd36',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': 'a2be6669-f1d9-4209-bdf6-804d9325fd36',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '128',
   'date': 'Wed, 07 Nov 2018 19:20:55 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:

```
#First create an Endpoint
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited   Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help   Trusted   conda_python3

Run   Code


```
 'ResponseMetadata': {'RequestId': 'a2be6669-f1d9-4209-bdf6-804d9325fd36',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': 'a2be6669-f1d9-4209-bdf6-804d9325fd36',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '128',
   'date': 'Wed, 07 Nov 2018 19:20:55 GMT'},
  'RetryAttempts': 0}}
```


In [ ]:
```
#First create an Endpoint
#endPointName='my-endpoint'+timestamp

create_endpoint_response=sagemakermp.create_endpoint(
    EndpointName=endPointName,
    EndpointConfigName=endPointConfigName,
    Tags=[{
            'Key': 'test-type',
            'Value': 'cortexica-model-testing'
        }])
create_endpoint_response
```

In [ ]:
```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomgb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [ ]:
```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

In [ ]:
```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
 'ResponseMetadata': {'RequestId': 'a2be6669-f1d9-4209-bdf6-804d9325fd36',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': 'a2be6669-f1d9-4209-bdf6-804d9325fd36',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '128',
   'date': 'Wed, 07 Nov 2018 19:20:55 GMT'},
  'RetryAttempts': 0}}
```

In [6]:

```
#First create an Endpoint
#endPointName='my-endpoint'+timestamp

create_endpoint_response=sagemakerrmp.create_endpoint(
    EndpointName=endPointName,
    EndpointConfigName=endPointConfigName,
    Tags=[{
            'Key': 'test-type',
            'Value': 'cortexica-model-testing'
        }])
create_endpoint_response
```

Out[6]:

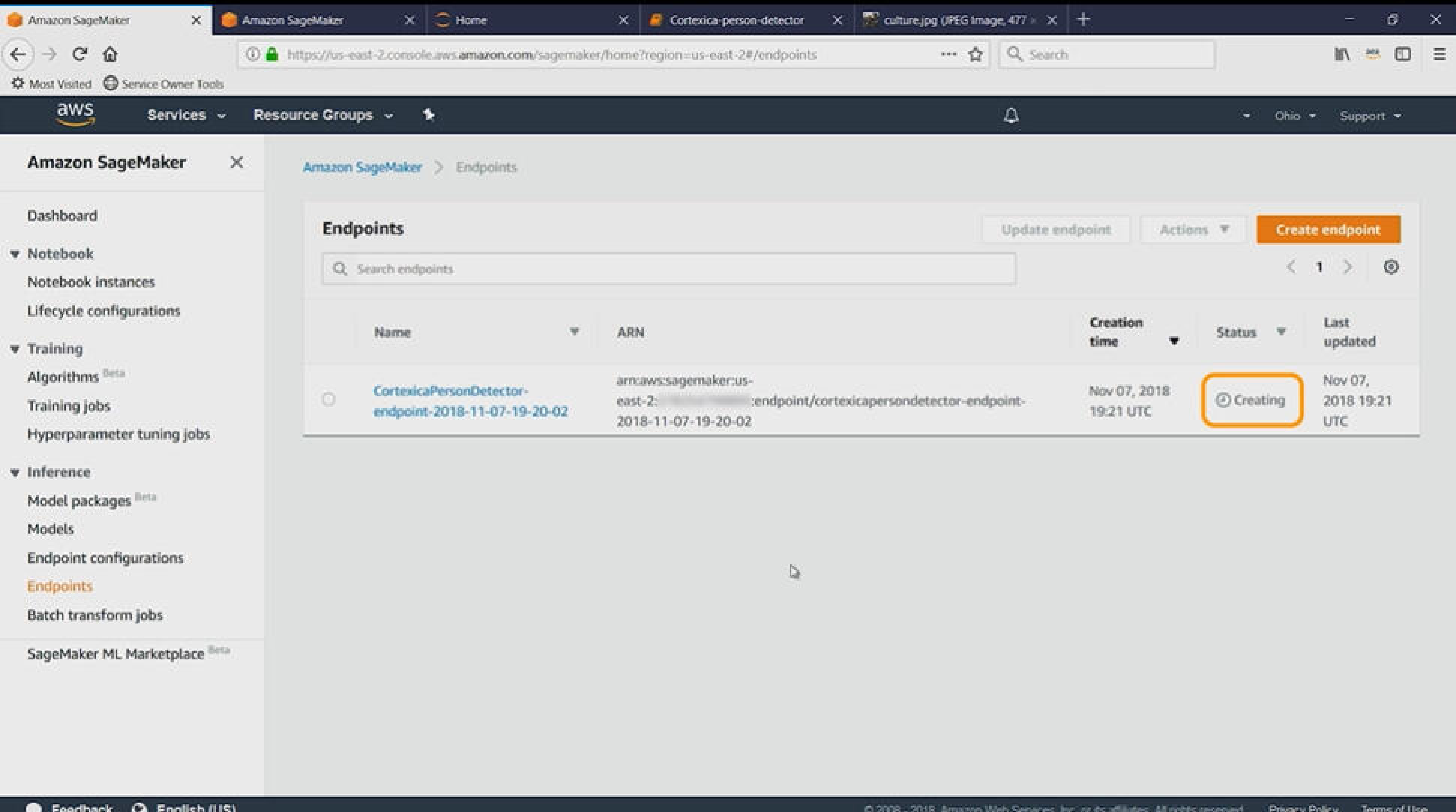
```
{'EndpointArn': 'arn:aws:sagemaker:us-east-2:[redacted]:endpoint/cortexicapersondetector-endpoint-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '120',
   'date': 'Wed, 07 Nov 2018 19:21:09 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:

```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomqb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [ ]:

```
#Download the input image
```


Amazon SageMaker
Amazon SageMaker
Home
Cortexica-person-detector
culture.jpg (JPEG Image, 477
https://us-east-2.console.aws.amazon.com/sagemaker/home?region=us-east-2#/endpoints
Search
Most Visited
Service Owner Tools
aws
Services
Resource Groups
Ohio
Support
Amazon SageMaker
Dashboard
Notebook
Notebook instances
Lifecycle configurations
Training
Algorithms Beta
Training jobs
Hyperparameter tuning jobs
Inference
Model packages Beta
Models
Endpoint configurations
Endpoints
Batch transform jobs
SageMaker ML Marketplace Beta
Amazon SageMaker > Endpoints
Endpoints
Update endpoint
Actions
Create endpoint
Search endpoints
1
Name
ARN
Creation time
Status
Last updated
CortexicaPersonDetector-endpoint-2018-11-07-19-20-02
arn:aws:sagemaker:us-east-2: :endpoint/cortexicapersondetector-endpoint-2018-11-07-19-20-02
Nov 07, 2018 19:21 UTC
Creating
Nov 07, 2018 19:21 UTC
Feedback
English (US)
Privacy Policy
Terms of Use

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
    EndpointName=endPointName,
    EndpointConfigName=endPointConfigName,
    Tags=[{
            'Key': 'test-type',
            'Value': 'cortexica-model-testing'
        }])
create_endpoint_response
```

Out[6]:

```
{'EndpointArn': 'arn:aws:sagemaker:us-east-2:[illegible]:endpoint/cortexicapersondetector-endpoint-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '120',
   'date': 'Wed, 07 Nov 2018 19:21:09 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:

```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomqb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [ ]:

```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

In [ ]:

```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter **Cortexica-person-detector** Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
    EndpointName=endPointName,
    EndpointConfigName=endPointConfigName,
    Tags=[{
            'Key': 'test-type',
            'Value': 'cortexica-model-testing'
        }])
create_endpoint_response
```

Out[6]:
```
{'EndpointArn': 'arn:aws:sagemaker:us-east-2:[illegible]:endpoint/cortexicapersondetector-endpoint-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '120',
   'date': 'Wed, 07 Nov 2018 19:21:09 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:
```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomqb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [ ]:
```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

In [ ]:
```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ...

https://www.amazondelivers.jobs/media/16938/culture.jpg

Search

Most Visited | Service Owner Tools

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ×

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter **Cortexica-person-detector** Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
    EndpointName=endPointName,
    EndpointConfigName=endPointConfigName,
    Tags=[{
            'Key': 'test-type',
            'Value': 'cortexica-model-testing'
        }])
create_endpoint_response
```

Out[6]:

```
{'EndpointArn': 'arn:aws:sagemaker:us-east-2:[illegible]:endpoint/cortexicapersondetector-endpoint-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '120',
   'date': 'Wed, 07 Nov 2018 19:21:09 GMT'},
  'RetryAttempts': 0}}
```

In [ ]:

```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomqb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [ ]:

```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

In [ ]:

```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ×

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Search

Most Visited   Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help   Trusted   conda_python3

Run   Code

```
    EndpointName=endPointName,
    EndpointConfigName=endPointConfigName,
    Tags=[{
            'Key': 'test-type',
            'Value': 'cortexica-model-testing'
        }])
create_endpoint_response
```

Out[6]:

```
{'EndpointArn': 'arn:aws:sagemaker:us-east-2:[illegible]:endpoint/cortexicapersondetector-endpoint-2018-11-07-19-20-02',
 'ResponseMetadata': {'RequestId': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amzn-requestid': '39a3ec54-269a-4870-8c32-c0ce90c0bc04',
   'content-type': 'application/x-amz-json-1.1',
   'content-length': '120',
   'date': 'Wed, 07 Nov 2018 19:21:09 GMT'},
  'RetryAttempts': 0}}
```

In [7]:

```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomqb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [ ]:

```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

In [ ]:

```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ×

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

```
  'content-length': '120',
  'date': 'Wed, 07 Nov 2018 19:21:09 GMT'},
 'RetryAttempts': 0}}
```

In [7]:
```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomgb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [8]:
```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

Out[8]: `('input.jpg', <http.client.HTTPMessage at 0x7f2793f8bf98>)`

In [ ]:
```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
data= output['Body'].read()

fh = open("output" + timestamp + ".jpeg", "wb")
fh.write(data)
fh.close()
```

In [ ]:
```
#Lets upload data to S3

batch_job_name='Cortexica-batchJob-'+timestamp

batch_input = session.upload_data("input.jpg", bucket, "data()/Cortexica-batch_input".format(timestamp))
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ×

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted conda_python3

Run Code

```
  'content-length': '120',
  'date': 'Wed, 07 Nov 2018 19:21:09 GMT'},
 'RetryAttempts': 0}}
```

In [7]:
```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomgb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [8]:
```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

Out[8]: ('input.jpg', <http.client.HTTPMessage at 0x7f2793f8bf98>)

In [ ]:
```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
data= output['Body'].read()

fh = open("output" + timestamp + ".jpeg", "wb")
fh.write(data)
fh.close()
```

In [ ]:
```
#Lets upload data to S3

batch_job_name='Cortexica-batchJob-'+timestamp

batch_input = session.upload_data("input.jpg", bucket, "data{}/Cortexica-batch_input".format(timestamp))
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ×

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help   Trusted   conda_python3

Run   Code

```
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomqb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [8]:

```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

Out[8]:

```
('input.jpg', <http.client.HTTPMessage at 0x7f2793f8bf98>)
```

In [ ]:

```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
data= output['Body'].read()

fh = open("output" + timestamp + ".jpeg", "wb")
fh.write(data)
fh.close()
```

In [ ]:

```
#Lets upload data to S3

batch_job_name='Cortexica-batchJob-'+timestamp

batch_input = session.upload_data("input.jpg", bucket, "data{}/Cortexica-batch_input".format(timestamp))
"uploaded training data file to {}".format(batch_input)
```

In [ ]:

```
%%time

batch_output = 's3://{}/{}/output'.format(bucket, batch_job_name)
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ×

https://my-notebook.notebook.us-east-2.sagemaker.aws/notebooks/Cortexica-person-detector.ipynb

Most Visited | Service Owner Tools

jupyter Cortexica-person-detector Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Code

In [7]:

```
#specify input image
url='https://www.amazondelivers.jobs/media/16938/culture.jpg'

#url='https://d2908q01vomqb2.cloudfront.net/cb4e5208b4cd87268b208e49452ed6e89a68e0b8/2018/02/08/smallpdf-webteam-1024x76
```

In [8]:

```
#Download the input image
urllib.request.urlretrieve(url,'input.jpg')
```

Out[8]: ('input.jpg', <http.client.HTTPMessage at 0x7f2793f8bf98>)

In [10]:

```
#Perform real-time prediction

#endPointName='my-endpoint-2018-11-07-02-10-30'

runtime = boto3.client('sagemaker-runtime')
with open("input.jpg", "rb") as imageFile:
  f = imageFile.read()
  b = bytearray(f)
output=runtime.invoke_endpoint(EndpointName=endPointName,Body=b,Accept="image/jpeg", ContentType='image/jpeg')
print(output)
data= output['Body'].read()

fh = open("output" + timestamp + ".jpeg", "wb")
fh.write(data)
fh.close()
```

{'ResponseMetadata': {'RequestId': 'cc368f31-e61f-4dac-a25f-9529cb7b941a', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'cc368f31-e61f-4dac-a25f-9529cb7b941a', 'x-amzn-invoked-production-variant': 'default-variant-name', 'date': 'Wed, 7 Nov 2018 19:28:05 GMT', 'content-type': 'image/jpeg', 'content-length': '78369'}, 'RetryAttempts': 0}, 'ContentType': 'image/jpeg', 'InvokedProductionVariant': 'default-variant-name', 'Body': <botocore.response.StreamingBody object at 0x7f2793fee240>}

In [ ]:

```
#Lets upload data to S3

batch_job_name='Cortexica-batchJob-'+timestamp
```

Amazon SageMaker | Amazon SageMaker | Home | Cortexica-person-detector | culture.jpg (JPEG Image, 477 ×

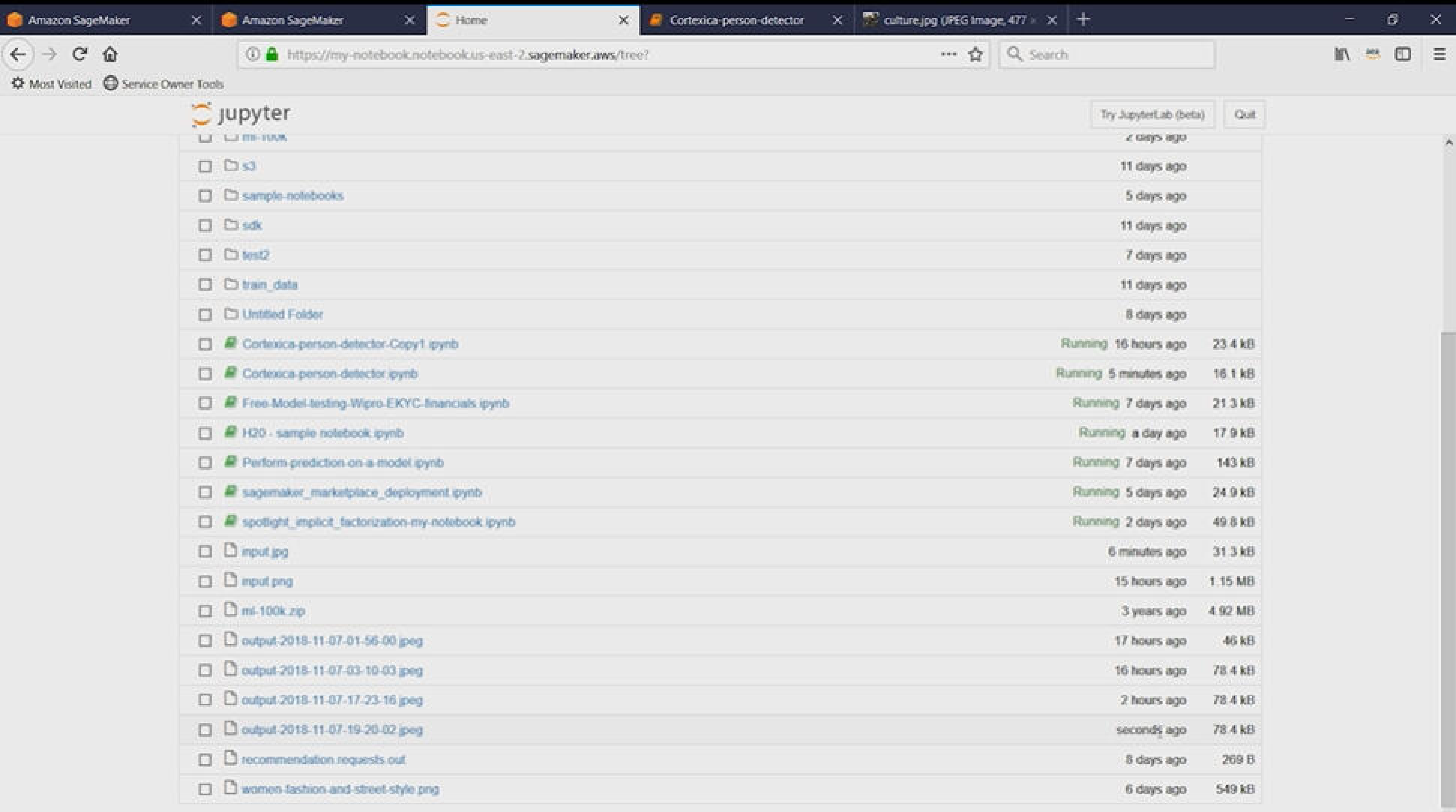
https://my-notebook.notebook.us-east-2.sagemaker.aws/tree?

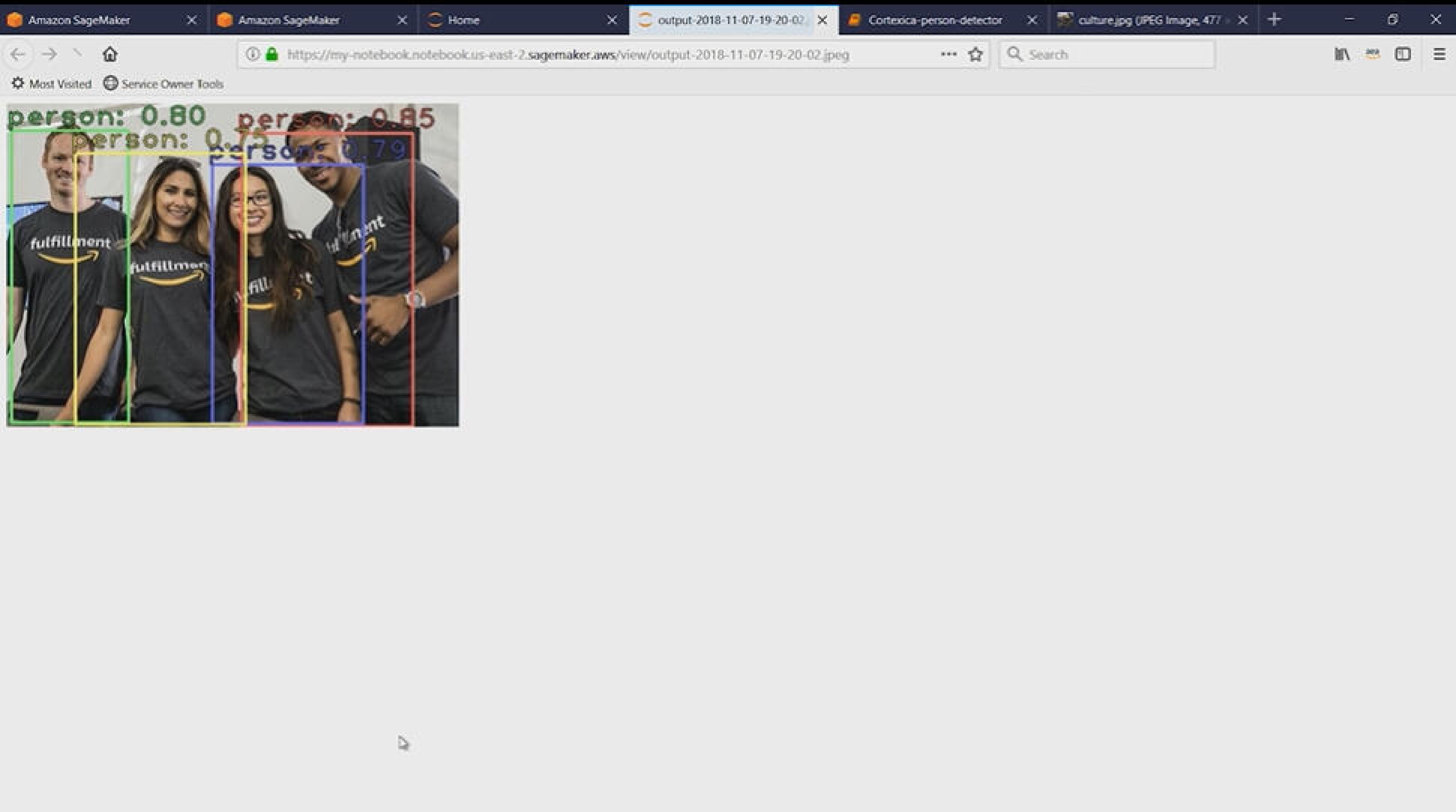
Most Visited Service Owner Tools

jupyter

Try JupyterLab (beta) Quit

Name	Status	Last Modified	File size
ml-100k		2 days ago	
s3		11 days ago	
sample-notebooks		5 days ago	
sdk		11 days ago	
test2		7 days ago	
train_data		11 days ago	
Untitled Folder		8 days ago	
Cortexica-person-detector-Copy1.ipynb	Running	16 hours ago	23.4 kB
Cortexica-person-detector.ipynb	Running	5 minutes ago	16.1 kB
Free-Model-testing-Wipro-EKYC-financials.ipynb	Running	7 days ago	21.3 kB
H20 - sample notebook.ipynb	Running	a day ago	17.9 kB
Perform-prediction-on-a-model.ipynb	Running	7 days ago	143 kB
sagemaker_marketplace_deployment.ipynb	Running	5 days ago	24.9 kB
spotlight_implicit_factorization-my-notebook.ipynb	Running	2 days ago	49.8 kB
input.jpg		6 minutes ago	31.3 kB
input.png		15 hours ago	1.15 MB
ml-100k.zip		3 years ago	4.92 MB
output-2018-11-07-01-56-00.jpeg		17 hours ago	46 kB
output-2018-11-07-03-10-03.jpeg		16 hours ago	78.4 kB
output-2018-11-07-17-23-16.jpeg		2 hours ago	78.4 kB
output-2018-11-07-19-20-02.jpeg		seconds ago	78.4 kB
recommendation.requests.out		8 days ago	269 B
women-fashion-and-street-style.png		6 days ago	549 kB


Amazon SageMaker
Amazon SageMaker
Home
Cortexica-person-detector
culture.jpg (JPEG Image, 477
https://my-notebook.notebook.us-east-2.sagemaker.aws/tree?
Search
Most Visited
Service Owner Tools
jupyter
Try JupyterLab (beta)
Quit
s3 11 days ago
sample-notebooks 5 days ago
sdk 11 days ago
test2 7 days ago
train_data 11 days ago
Untitled Folder 8 days ago
Cortexica-person-detector-Copy1.ipynb Running 16 hours ago 23.4 kB
Cortexica-person-detector.ipynb Running 5 minutes ago 16.1 kB
Free-Model-testing-Wipro-EKYC-financials.ipynb Running 7 days ago 21.3 kB
H20 - sample notebook.ipynb Running a day ago 17.9 kB
Perform-prediction-on-a-model.ipynb Running 7 days ago 143 kB
sagemaker_marketplace_deployment.ipynb Running 5 days ago 24.9 kB
spotlight_implicit_factorization-my-notebook.ipynb Running 2 days ago 49.8 kB
input.jpg 6 minutes ago 31.3 kB
input.png 15 hours ago 1.15 MB
ml-100k.zip 3 years ago 4.92 MB
output-2018-11-07-01-56-00.jpeg 17 hours ago 46 kB
output-2018-11-07-03-10-03.jpeg 16 hours ago 78.4 kB
output-2018-11-07-17-23-16.jpeg 2 hours ago 78.4 kB
output-2018-11-07-19-20-02.jpeg seconds ago 78.4 kB
recommendation.requests.out 8 days ago 269 B
women-fashion-and-street-style.png 6 days ago 549 kB


Amazon SageMaker
Amazon SageMaker
Home
output-2018-11-07-19-20-02
Cortexica-person-detector
culture.jpg (JPEG Image, 477
https://my-notebook.notebook.us-east-2.sagemaker.aws/view/output-2018-11-07-19-20-02.jpeg
Search
Most Visited
Service Owner Tools
person: 0.80
person: 0.85
person: 0.75
person: 0.79
fulfillment
fulfillmen

Amazon SageMaker | Amazon SageMaker | Home | output-2018-11-07-19-20-02 | Cortexica-person-detector | culture.jpg (JPEG Image, 477

https://my-notebook.notebook.us-east-2.sagemaker.aws/view/output-2018-11-07-19-20-02.jpeg

Search

Most Visited | Service Owner Tools

Amazon SageMaker | Amazon SageMaker | Home | output-2018-11-07-19-20-02 | Cortexica-person-detector | culture.jpg (JPEG Image, 477

https://my-notebook.notebook.us-east-2.sagemaker.aws/view/output-2018-11-07-19-20-02.jpeg

Most Visited | Service Owner Tools