
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

BOA322

Build and deploy a live, ML-powered music genre classifier

Sohan Maheshwar
Developer Advocate
Amazon Web Services

Linda Haviv
Developer Advocate
Amazon Web Services

Suman Debnath
Developer Advocate
Amazon Web Services

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Imagine this



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This is what we are going to build in the next 30 minutes

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This is what we are going to build in the next 30 minutes

- Building a music classifier
  - By genre

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This is what we are going to build in the next 30 minutes

- Building a music classifier
  - By genre

- Train on AWS
  - Build and train the machine learning model on AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This is what we are going to build in the next 30 minutes

- Building a music classifier
  - By genre
- Train on AWS
  - Build and train the Machine Learning Model on AWS
- Deploy on AWS
  - Using a serverless endpoint
  - Fully automated

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training a ML model

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GTZAN dataset

DATA IN AUDIO AND TEXT FORMATS



Audio
(.wav)

	A	B	C	D	E	F	G	H	I	J	K
1	filename	length	chroma_stft_	chroma_stft_	rms_mean	rms_var	spectral_cen	spectral_cen	spectral_ban	spectral_ban	rolloff_mear
2	blues.00000.	661794	0.35008812	0.08875657	0.13022792	0.0028267	1784.16585	129774.065	2002.44906	85882.7613	3805.83961
3	blues.00001.	661794	0.34091359	0.09498025	0.09594781	0.00237274	1530.17668	375850.074	2039.03652	213843.755	3550.5221
4	blues.00002.	661794	0.36363718	0.0852752	0.17557041	0.00274592	1552.81186	156467.643	1747.70231	76254.1923	3042.26023
5	blues.00003.	661794	0.40478471	0.09399904	0.141093	0.00634635	1070.10661	184355.942	1596.41287	166441.495	2184.7458
6	blues.00004.	661794	0.30852604	0.08784098	0.09152871	0.0023034	1835.00427	343399.939	1748.17212	88445.209	3579.75763
7	blues.00005.	661794	0.30245629	0.08753238	0.10349364	0.00398051	1831.99394	1030482.37	1729.65329	201910.509	3481.51759
8	blues.00006.	661794	0.29132798	0.09398133	0.14187415	0.00880319	1459.36647	437859.398	1389.00913	185023.24	2795.61096
9	blues.00007.	661794	0.30795479	0.09290288	0.13182218	0.00553114	1451.66707	449568.162	1577.27094	168211.939	2954.83676
10	blues.00008.	661794	0.40887922	0.08651248	0.14241649	0.00150716	1719.36895	163282.845	2031.74038	105542.718	3782.31629
11	blues.00009.	661794	0.2739504	0.09231596	0.08131366	0.00434686	1817.15086	298236.141	1973.77331	114070.113	3943.49057
12	blues.00010.	661794	0.30399331	0.09470294	0.14286506	0.00920028	1409.95526	205666.839	1512.31027	145008.805	2765.90081
13	blues.00011.	661794	0.36715207	0.10232875	0.06574069	0.00249991	1352.65967	512135.09	1756.7798	220626.792	2880.77786
14	blues.00012.	661794	0.26939112	0.09390165	0.11904605	0.0040528	1360.6412	257065.39	1567.56523	45236.5114	2738.63874
15	blues.00013.	661794	0.26471281	0.09117574	0.11316983	0.00418892	1324.29322	171474.993	1828.03925	78278.5668	2710.2692
16	blues.00014.	661794	0.32903633	0.10811146	0.06705453	0.00271697	1171.84823	147278.546	1705.5193	96584.4434	2344.24638
17	blues.00015.	661794	0.26969042	0.09498938	0.08062402	0.00282897	1420.36517	199958.978	1731.16428	81354.2615	2929.96449
18	blues.00016.	661794	0.30418557	0.09849298	0.08147175	0.00264104	1454.96394	152317.124	1825.79796	116774.736	3009.46082
19	blues.00017.	661794	0.30212399	0.09334568	0.09330608	0.00215077	1088.84423	231640.778	1410.98426	69795.4429	2134.99294

Text
(.csv)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

GTZAN dataset

DATA IN AUDIO AND TEXT FORMATS



Audio
(.wav)

	A	B	C	D	E	F	G	H	I	J	K
1	filename	length	chroma_stft_	chroma_stft_	rms_mean	rms_var	spectral_cen	spectral_cen	spectral_ban	spectral_ban	rolloff_mear
2	blues.00000.	661794	0.35008812	0.08875657	0.13022792	0.0028267	1784.16585	129774.065	2002.44906	85882.7613	3805.83961
3	blues.00001.	661794	0.34091359	0.09498025	0.09594781	0.00237274	1530.17668	375850.074	2039.03652	213843.755	3550.5221
4	blues.00002.	661794	0.36363718	0.0852752	0.17557041	0.00274592	1552.81186	156467.643	1747.70231	76254.1923	3042.26023
5	blues.00003.	661794	0.40478471	0.09399904	0.141093	0.00634635	1070.10661	184355.942	1596.41287	166441.495	2184.7458
6	blues.00004.	661794	0.30852604	0.08784098	0.09152871	0.0023034	1835.00427	343399.939	1748.17212	88445.209	3579.75763
7	blues.00005.	661794	0.30245629	0.08753238	0.10349364	0.00398051	1831.99394	1030482.37	1729.65329	201910.509	3481.51759
8	blues.00006.	661794	0.29132798	0.09398133	0.14187415	0.00880319	1459.36647	437859.398	1389.00913	185023.24	2795.61096
9	blues.00007.	661794	0.30795479	0.09290288	0.13182218	0.00553114	1451.66707	449568.162	1577.27094	168211.939	2954.83676
10	blues.00008.	661794	0.40887922	0.08651248	0.14241649	0.00150716	1719.36895	163282.845	2031.74038	105542.718	3782.31629
11	blues.00009.	661794	0.2739504	0.09231596	0.08131366	0.00434686	1817.15086	298236.141	1973.77331	114070.113	3943.49057
12	blues.00010.	661794	0.30399331	0.09470294	0.14286506	0.00920028	1409.95526	205666.839	1512.31027	145008.805	2765.90081
13	blues.00011.	661794	0.36715207	0.10232875	0.06574069	0.00249991	1352.65967	512135.09	1756.7798	220626.792	2880.77786
14	blues.00012.	661794	0.26939112	0.09390165	0.11904605	0.0040528	1360.6412	257065.39	1567.56523	45236.5114	2738.63874
15	blues.00013.	661794	0.26471281	0.09117574	0.11316983	0.00418892	1324.29322	171474.993	1828.03925	78278.5668	2710.2692
16	blues.00014.	661794	0.32903633	0.10811146	0.06705453	0.00271697	1171.84823	147278.546	1705.5193	96584.4434	2344.24638
17	blues.00015.	661794	0.26969042	0.09498938	0.08062402	0.00282897	1420.36517	199958.978	1731.16428	81354.2615	2929.96449
18	blues.00016.	661794	0.30418557	0.09849298	0.08147175	0.00264104	1454.96394	152317.124	1825.79796	116774.736	3009.46082
19	blues.00017.	661794	0.30212399	0.09334568	0.09330608	0.00215077	1088.84423	231640.778	1410.98426	69795.4429	2134.99294

Text
(.csv)

{ blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, rock }

(10 genres)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model training

TOOLS USED

- Scikit learn
  - For ML data analysis and training
- Librosa
  - For music and audio analysis.
- TensorFlow
  - ML training with the image file dataset
- k-Nearest neighbor, SVM, etc.
  - Algorithm used for training the model









aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.

How does the data look?

Audio features

- Mel spectrogram
- Zero crossing rate
- Spectral centroid
- Mel-frequency cepstral coefficients
- Spectral rolloff

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How does the data look?

Audio features

- Mel spectrogram
- Zero crossing rate
- Spectral centroid
- Mel-frequency cepstral coefficients
- Spectral rolloff

```
1 y, s = librosa.load(f'music_file_1.wav')
2
3 print('y:', y, '\n')
4 print('y shape:', np.shape(y), '\n')
5 print('Sample Rate (KHz):', s, '\n')
```

librosa

Sound

- Sequence of vibrations in varying pressure strengths (y)

Sample rate

- Number of samples of audio carried per second, measured in Hz or KHz (sr)

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How does the data look?

Audio features

- Mel spectrogram
- Zero crossing rate
- Spectral centroid
- Mel-frequency cepstral coefficients
- Spectral rolloff

```
1 y, s = librosa.load(f'music_file_1.wav')
2
3 print('y:', y, '\n')
4 print('y shape:', np.shape(y), '\n')
5 print('Sample Rate (KHz):', s, '\n')
```

librosa

Sound

- Sequence of vibrations in varying pressure strengths (y)

Sample rate

- Number of samples of audio carried per second, measured in Hz or KHz (sr)

```
y: [0.02072144 0.04492188 0.05422974 ...0.06912231 0.08303833 0.08572388]

y shape: (661794,)

Sample Rate (KHz): 22050
```

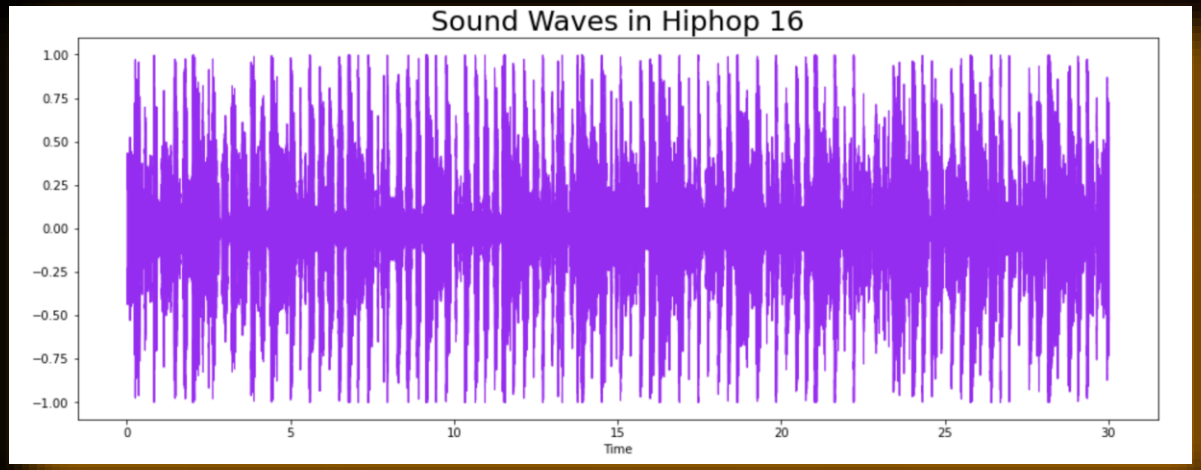
Output

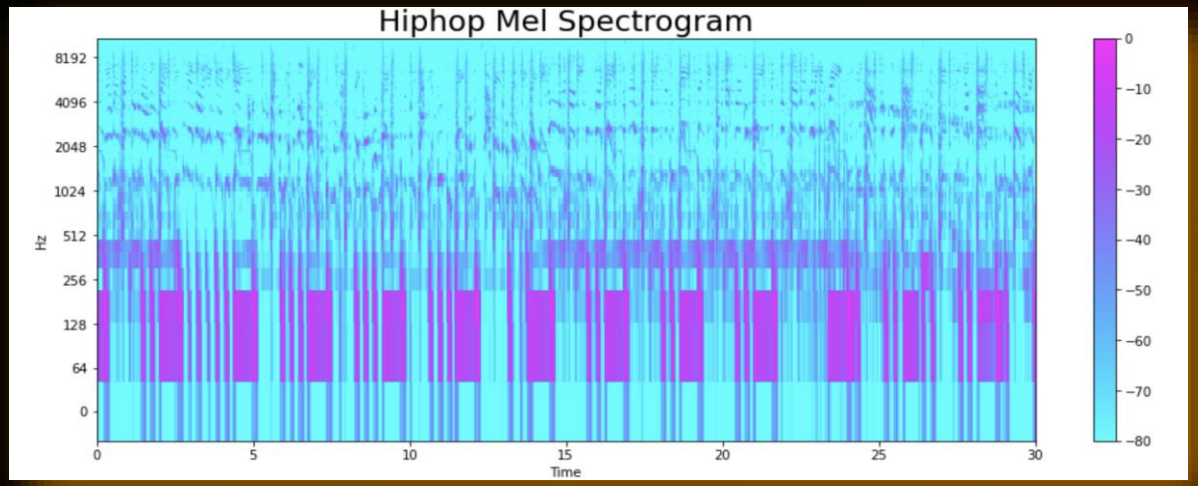
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

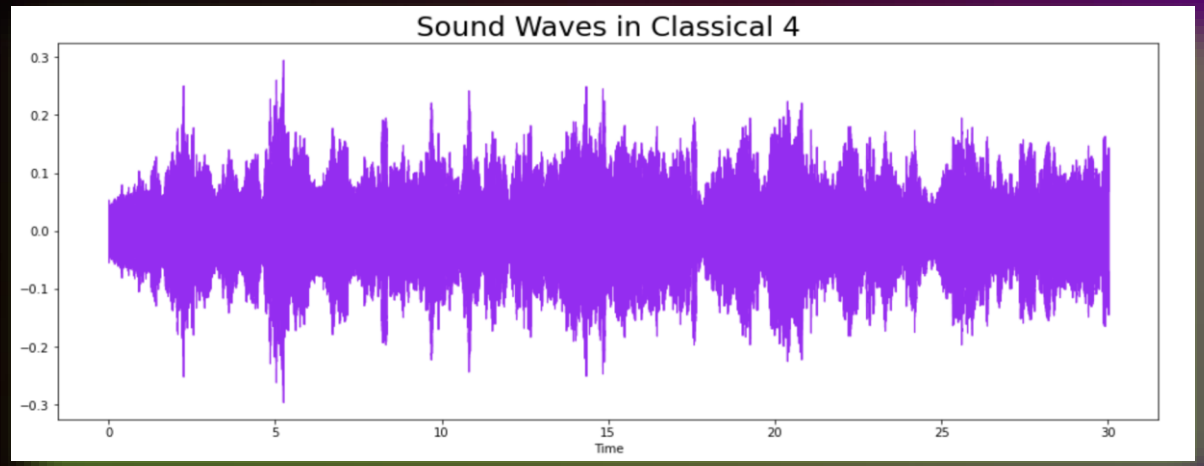
A comparison

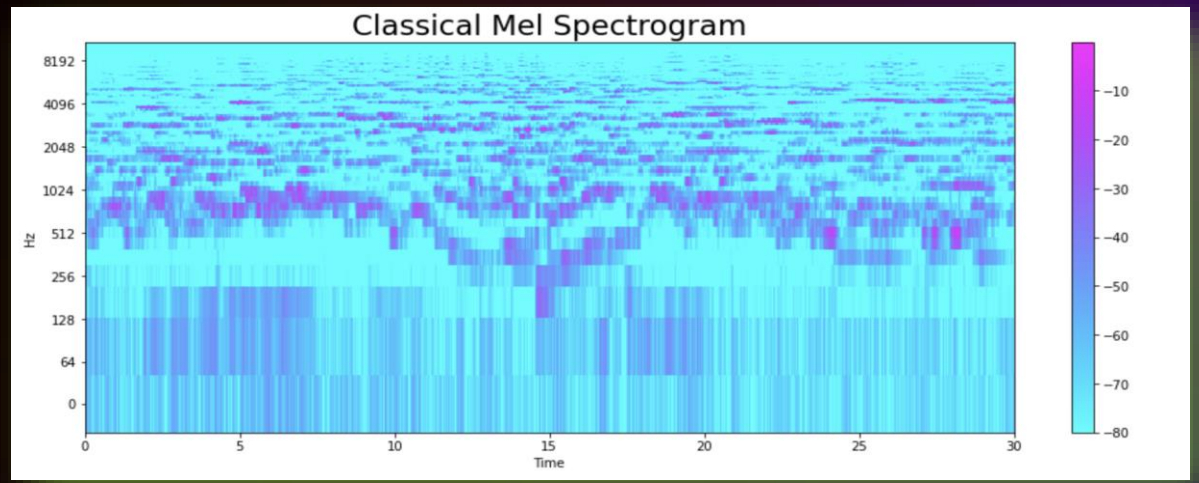
Hip-hop track





Classical track

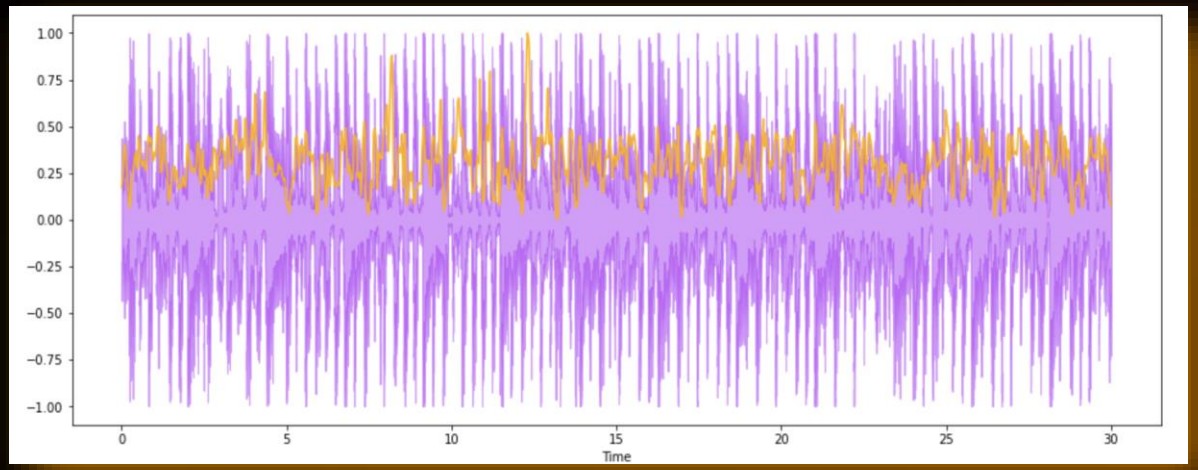


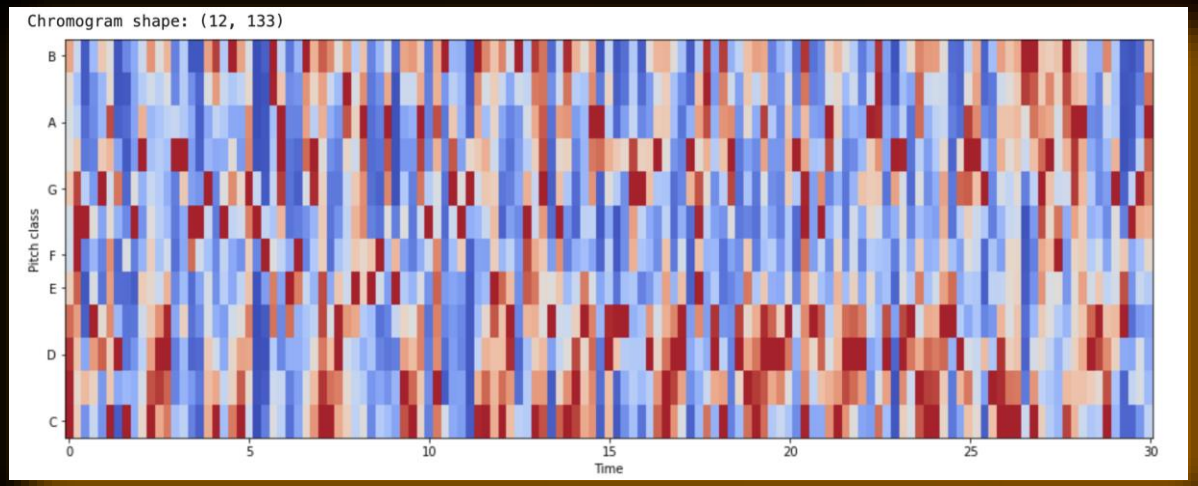


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

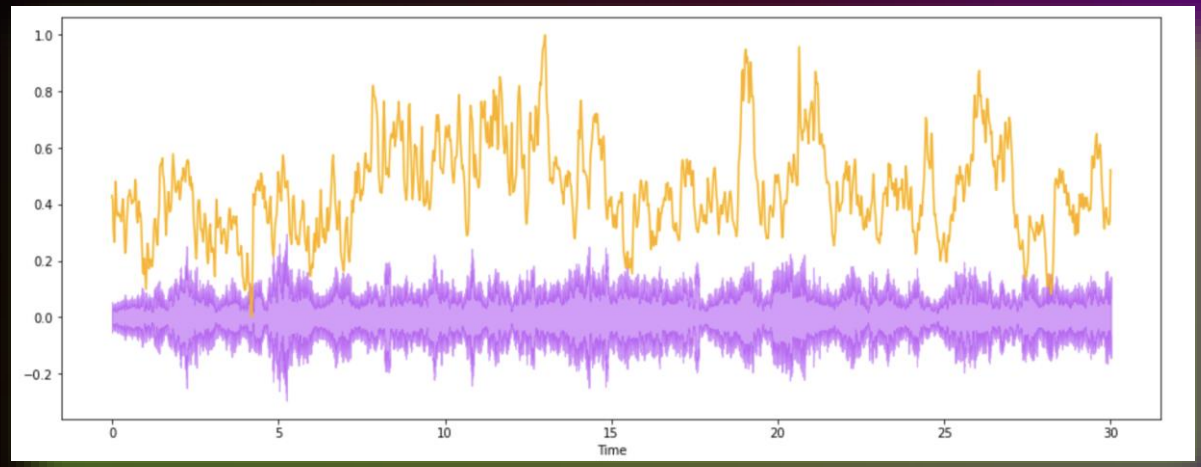
A comparison

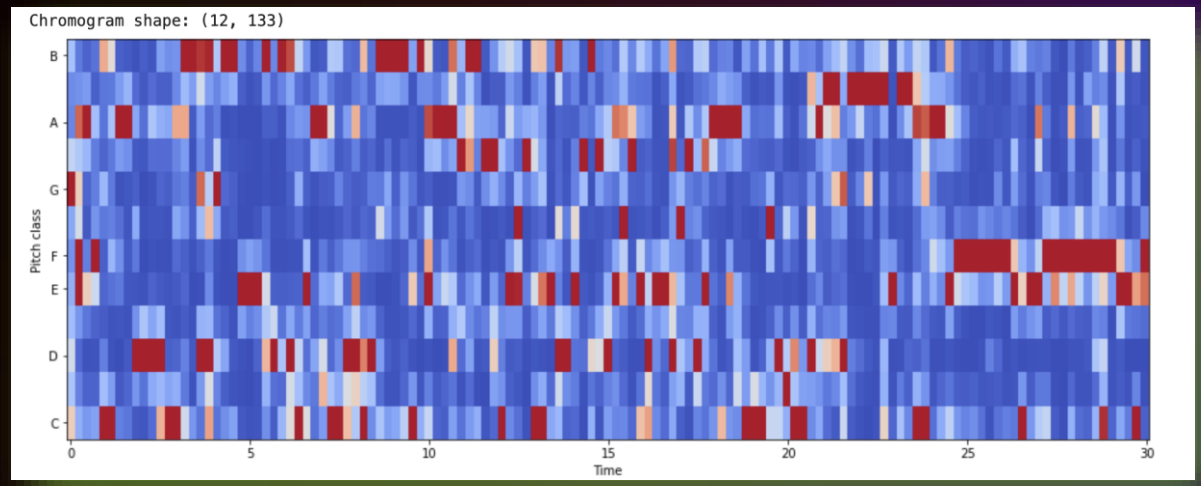
Hip-hop track





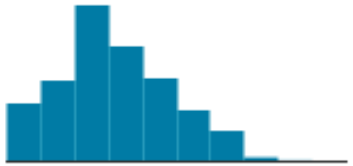
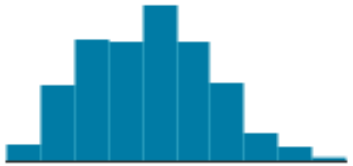
Classical track





aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Exploratory data analysis (EDA)

filename	length	chroma_stft_mean	rms_mean	spectral_centroid...
1000 unique values	 660k 676k	 0.17 0.66	 0.01 0.4	 570 4.44k
blues.00000.wav	661794	0.35008811950683594	0.1302279233932495	1784.165849538755
blues.00001.wav	661794	0.3409135937690735	0.09594780951738358	1530.1766787460795
blues.00002.wav	661794	0.36363717913627625	0.17557041347026825	1552.8118647610036
blues.00003.wav	661794	0.4047847092151642	0.14109300076961517	1070.1066149971282
blues.00004.wav	661794	0.30852603912353516	0.09152871370315552	1835.0042655120715
blues.00005.wav	661794	0.3024562895298004	0.10349363833665848	1831.993939606849

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Exploratory data analysis (EDA)

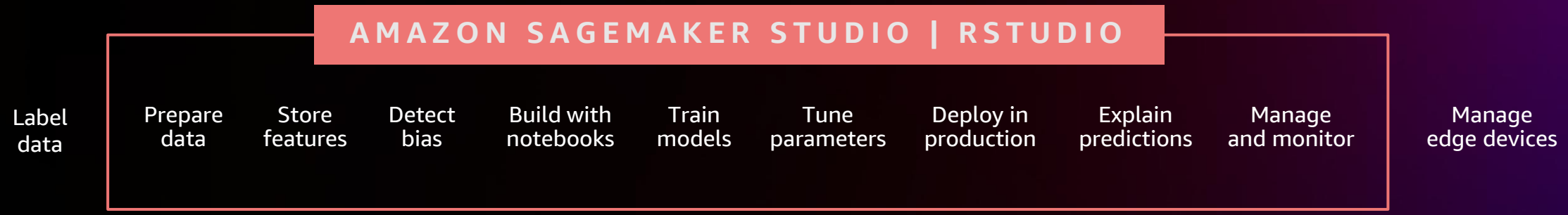
filename	length	chroma_stft_mean	rms_mean	spectral_centroid...
1000 unique values	660k – 676k	0.17 – 0.66	0.01 – 0.4	570 – 4.44k
blues.00000.wav	661794	0.35008811950683594	0.1302279233932495	1784.165849538755
blues.00001.wav	661794	0.3409135937690735	0.09594780951738358	1530.1766787460795
blues.00002.wav	661794	0.36363717913627625	0.17557041347026825	1552.8118647610036
blues.00003.wav	661794	0.4047847092151642	0.14109300076961517	1070.1066149971282
blues.00004.wav	661794	0.30852603912353516	0.09152871370315552	1835.0042655120715
blues.00005.wav	661794	0.3024562895298004	0.10349363833665848	1831.993939606849

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker brings tools for every step of the ML lifecycle under one unified visual user interface



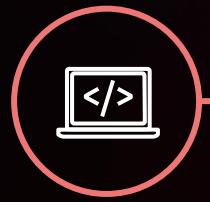
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Build ML models

Fully managed shareable notebooks on Amazon EC2



Fully managed, sharable Jupyter notebooks
Run notebooks on elastic compute resources



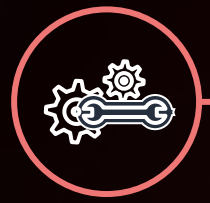
Built-in algorithms
15 built-in algorithms available in prebuilt container images



Prebuilt solutions and open-source models
More than 150 popular open-source models



AutoML
Automatically create ML models with full visibility



Support for major frameworks and toolkits
Optimized for popular deep learning (DL) frameworks such as TensorFlow, PyTorch, Apache MXNet, and Hugging Face

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Train ML models

Fast and cost-effective ML model training



Experiment management and model tuning

Save weeks of effort by automatically tracking training runs and tuning hyperparameters



Debug and profile training runs

Use real-time metrics to correct performance problems



Distributed training

Complete distributed training up to 40% faster



Training compiler

Accelerate training times by up to 50% through more efficient use of GPUs



Managed spot training

Reduce the costs of training by up to 90%

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are we going to build?



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are we going to build?



User request

Inference

Build and train model

AWS Cloud

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are we going to build?



User request

Inference

Build and train model

AWS Cloud

Prediction

```
{
  "predicted_label" : "Disco"
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What are we going to build?



User request

Inference

Build and train model

AWS Cloud

Prediction

```
{
  "predicted_label" : "Disco"
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Servers

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What size servers are
right for my budget?

Servers

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What size servers are
right for my budget?

Servers

How many servers
should I budget for?

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What size servers are right for my budget?

Servers

How can I control **access from** my servers?

How will the application handle server **hardware failure**?

How many servers should I budget for?

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Servers

How should my app withstand a server **failing**?

How can I tell if a server has been **compromised**?

How can I increase **utilization** of my servers?

Which **OS** should my servers run?

How much remaining **capacity** do my servers have?

When should I decide to **scale up** my servers?

What size servers are right for my budget?

How should I implement dynamic **configuration changes** on my servers?

How will I keep my server OS **patched**?

How can I control **access from** my servers?

Which packages should be baked into my **server images**?

(Arghhhhhhhh!)

How will new code be **deployed** to my servers?

How will the application handle server **hardware failure**?

What size server is right for my **performance**?

How many users create **too much load** for my servers?

Which users should have **access to** my servers?

Should I **tune OS settings** to optimize my application?

How many servers should I budget for?

When should I decide to **scale out** my servers?

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How should my app withstand a server **failing**?

How can I tell if a server has been **compromised**?

How can I increase **utilization** of my servers?

Which **OS** should my servers run?

How much remaining **capacity** do my servers have?

How should I implement dynamic **configuration changes** on my servers?

When should I decide to **scale up** my servers?

What size servers are right for my budget?

How will I keep my server OS **patched**?

In most cases, a developer's time is spent on the **operation and maintenance of applications**, rather than the actual **business problem**

(Arghhhhhhhh!)

How will the application handle server **hardware failure**?

How will new code be **deployed** to my servers?

What size server is right for my **performance**?

How many users create **too much load** for my servers?

Which users should have **access to** my servers?

Should I **tune OS settings** to optimize my application?

How many servers should I budget for?

When should I decide to **scale out** my servers?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hello serverless!

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless !== No Servers

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless !== No Servers



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless !== No Servers





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda



Event-driven compute

Function as a service (FaaS)

Serverless FaaS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

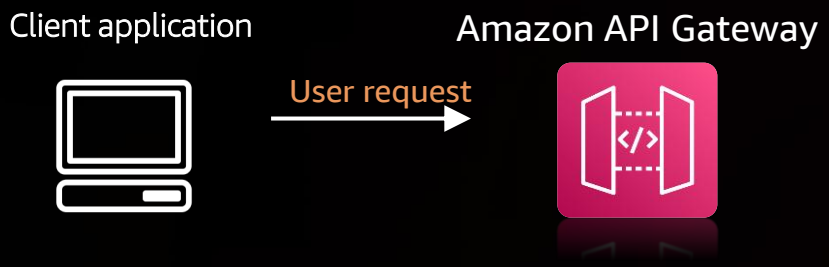
Architecture

Client application



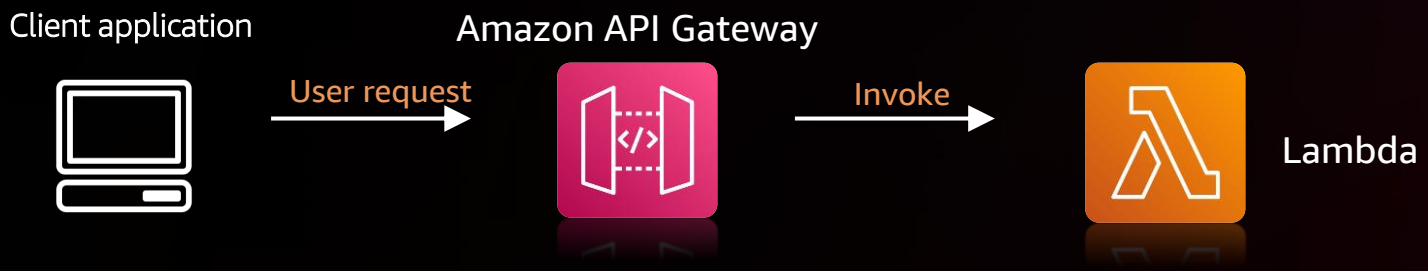
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



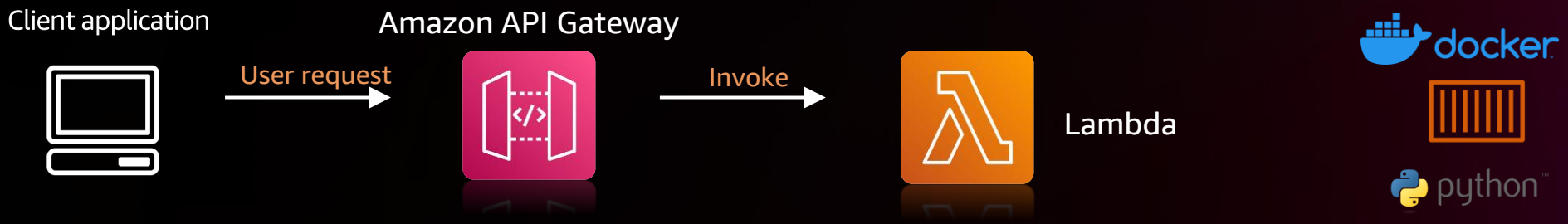
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



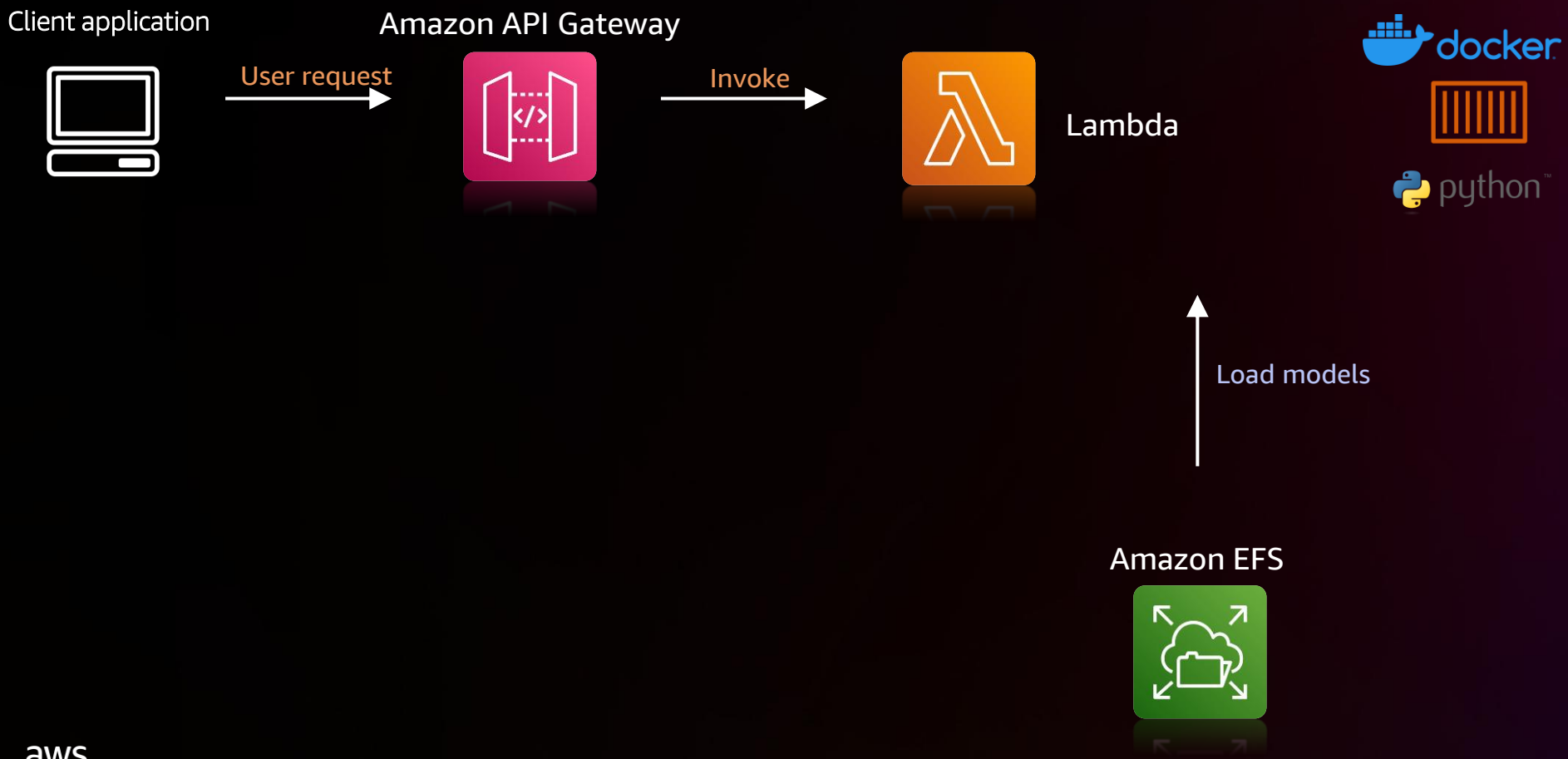
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



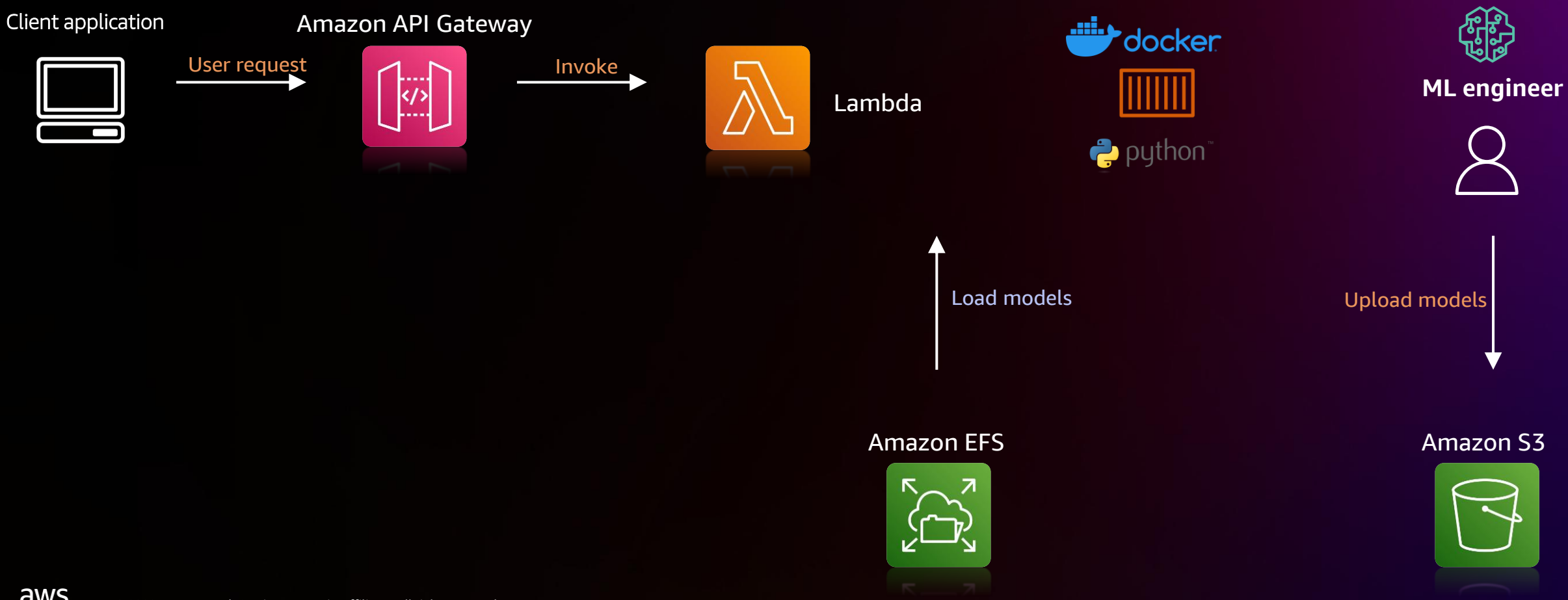
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



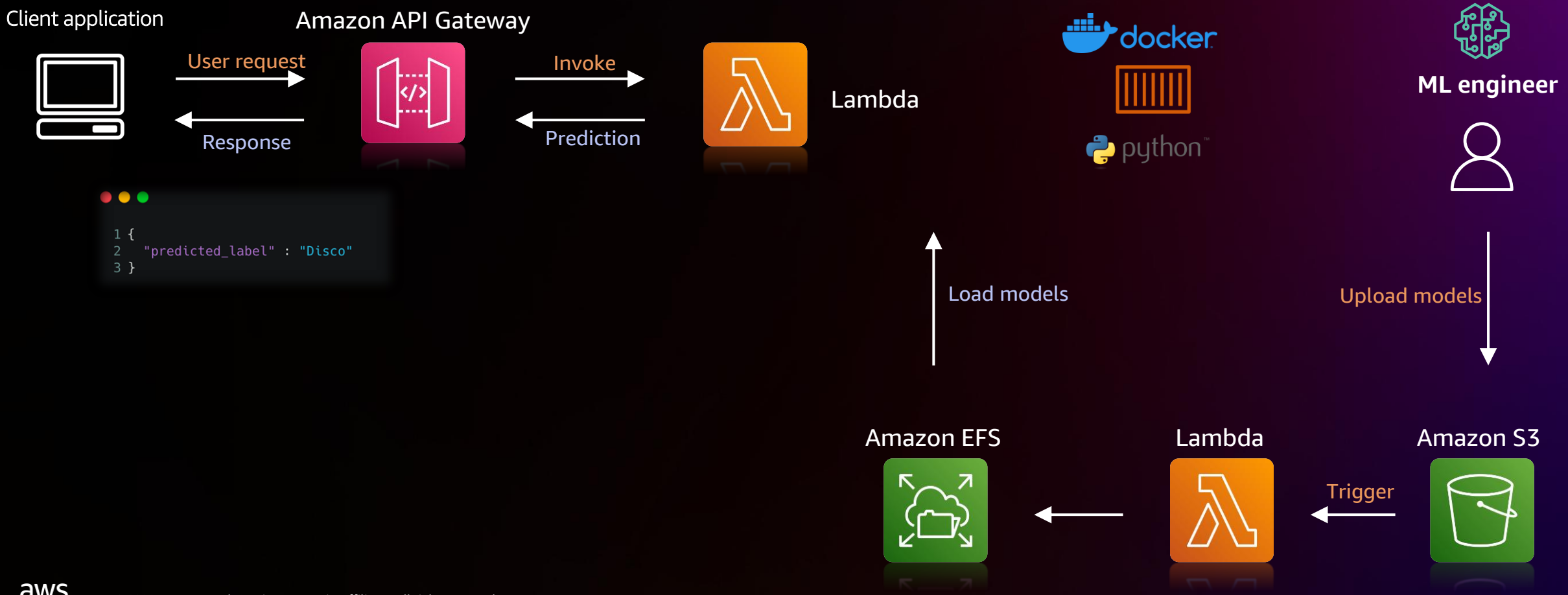
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



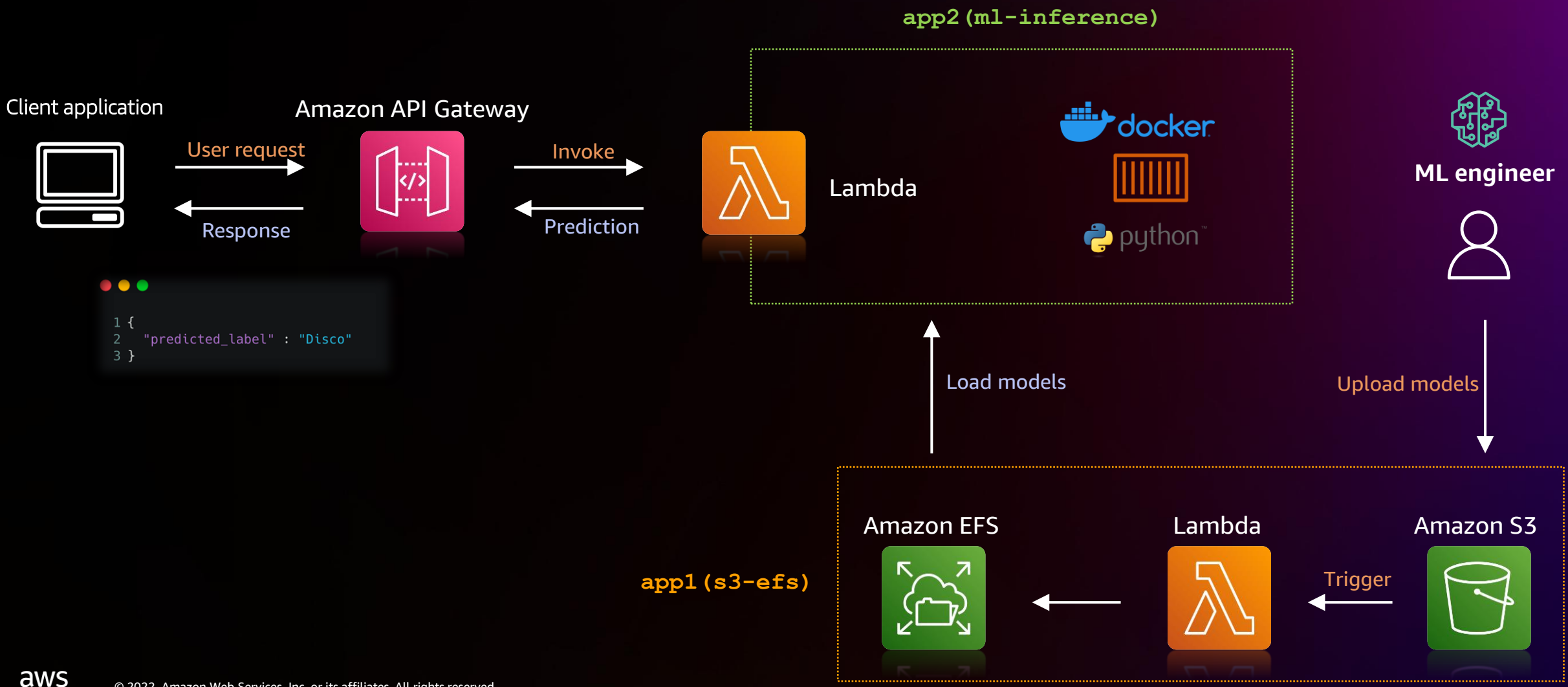
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless Application Model (SAM)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless Application Model (SAM)

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >

Resources:

  MyS3Bucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: !Ref SrcBucket

  EfsFileSystem:
    Type: AWS::EFS::FileSystem

  MLModelUploadFunction:
    Type: AWS::Serverless::Function
    DependsOn:
      - MountTargetA
    Properties:
      CodeUri: s3-efs/
      Handler: app.lambda_handler
      Runtime: python3.8
      Events:
        UploadMLModelEvent:
          Type: S3
          Properties:
            Bucket: !Ref MyS3Bucket
            Events: s3:ObjectCreated:*
  InferenceFunction:
    Type: AWS::Serverless::Function
    DependsOn:
      - MountTargetA
    Properties:
      PackageType: Image
      VpcConfig:
        SecurityGroupIds:
          - !Ref EfsLambdaSecurityGroup
        SubnetIds:
          - !Ref SubnetAPrivate
      FileSystemConfigs:
        - Arn: !GetAtt AccessPoint.Arn
          LocalMountPath: /mnt/ml
      Events:
        Inference:
          Type: Api
          Properties:
            Path: /classify_song
            Method: POST
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless Application Model (SAM)

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >

Resources:

  MyS3Bucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: !Ref SrcBucket

  EfsFileSystem:
    Type: AWS::EFS::FileSystem

  MLModelUploadFunction:
    Type: AWS::Serverless::Function
    DependsOn:
      - MountTargetA
    Properties:
      CodeUri: s3-efs/
      Handler: app.lambda_handler
      Runtime: python3.8
      Events:
        UploadMLModelEvent:
          Type: S3
          Properties:
            Bucket: !Ref MyS3Bucket
            Events: s3:ObjectCreated:*
  InferenceFunction:
    Type: AWS::Serverless::Function
    DependsOn:
      - MountTargetA
    Properties:
      PackageType: Image
      VpcConfig:
        SecurityGroupIds:
          - !Ref EfsLambdaSecurityGroup
        SubnetIds:
          - !Ref SubnetAPrivate
      FileSystemConfigs:
        - Arn: !GetAtt AccessPoint.Arn
          LocalMountPath: /mnt/ml
      Events:
        Inference:
          Type: Api
          Properties:
            Path: /classify_song
            Method: POST
```

The SAM template creates

- Amazon S3 bucket
- Amazon EFS file system
- Lambda function
- API Gateway

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless Application Model (SAM)

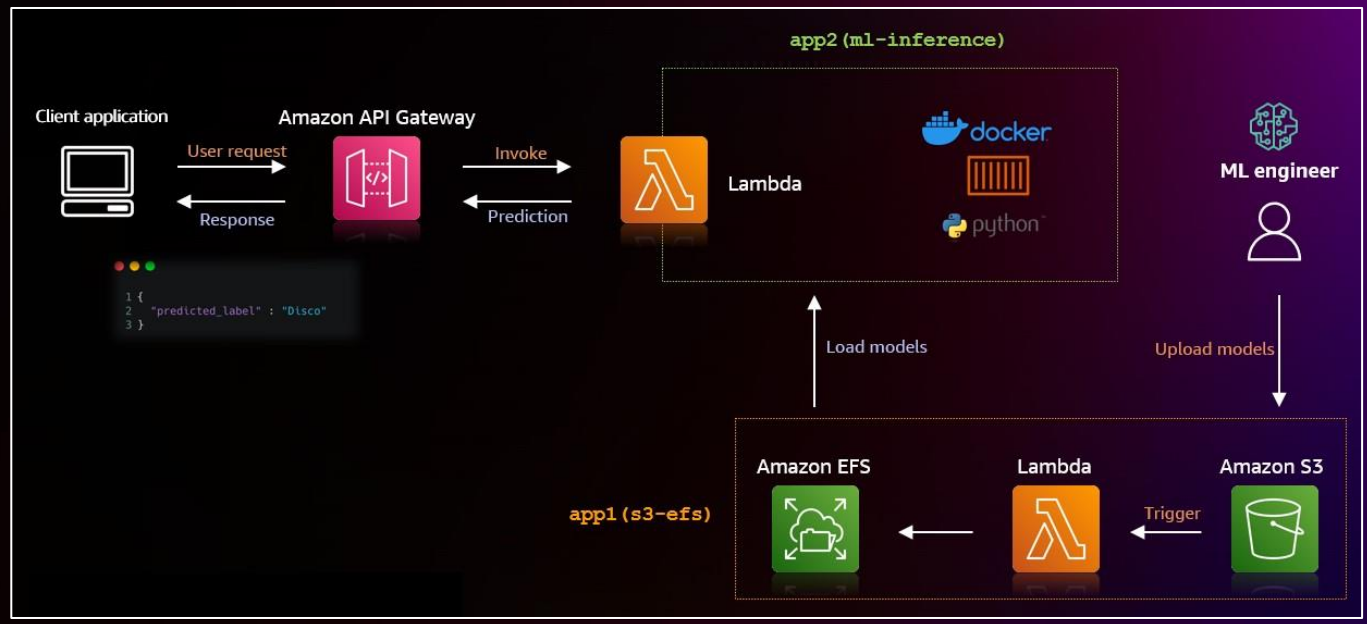
```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >

Resources:

  MyS3Bucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: !Ref SrcBucket

  EfsFileSystem:
    Type: AWS::EFS::FileSystem

  MLModelUploadFunction:
    Type: AWS::Serverless::Function
    DependsOn:
      - MountTargetA
    Properties:
      CodeUri: s3-efs/
      Handler: app.lambda_handler
      Runtime: python3.8
      Events:
        UploadMLModelEvent:
          Type: S3
          Properties:
            Bucket: !Ref MyS3Bucket
            Events: s3:ObjectCreated:*
  InferenceFunction:
    Type: AWS::Serverless::Function
    DependsOn:
      - MountTargetA
    Properties:
      PackageType: Image
      VpcConfig:
        SecurityGroupIds:
          - !Ref EfsLambdaSecurityGroup
        SubnetIds:
          - !Ref SubnetAPrivate
      FileSystemConfigs:
        - Arn: !GetAtt AccessPoint.Arn
          LocalMountPath: /mnt/ml
      Events:
        Inference:
          Type: Api
          Properties:
            Path: /classify_song
            Method: POST
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What next

- To **explore more** check out the GitHub Repo
  - GTZAN dataset: https://bit.ly/3GrPIEW
  - Model training
  - SAM deployment
- **SageMaker example** notebooks
  https://bit.ly/3UKPtJa
- **Hands-on workshops**
  - Machine learning: Amazon SageMaker workshop
    https://sagemaker-workshop.com/
  - Deployment: Building Serverless Apps with SAM
    https://bit.ly/3tHBYOs



https://github.com/debnsuma/boa322-master

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Sohan Maheshwar
in/sohanmaheshwar/
@soganmageshwar

Linda Haviv
in/lindahaviv/
@lindavivah

Suman Debnath
in/suman-d/
@_sumand



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.