



Deploy your ML model as a serverless API

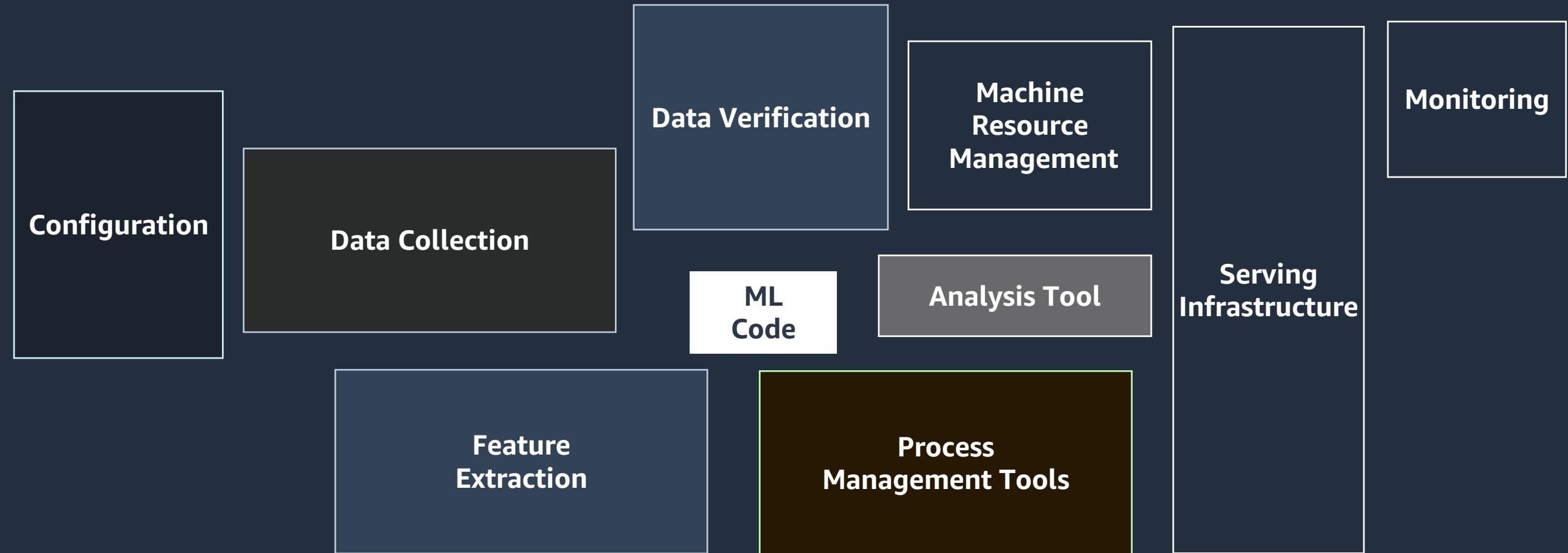
Turn your model into a cost effective and scalable API

Nicola Pietroluongo – *AWS Senior Solutions Architect*

29/07/2021

ML Code

...is one small part of the overall deployment picture

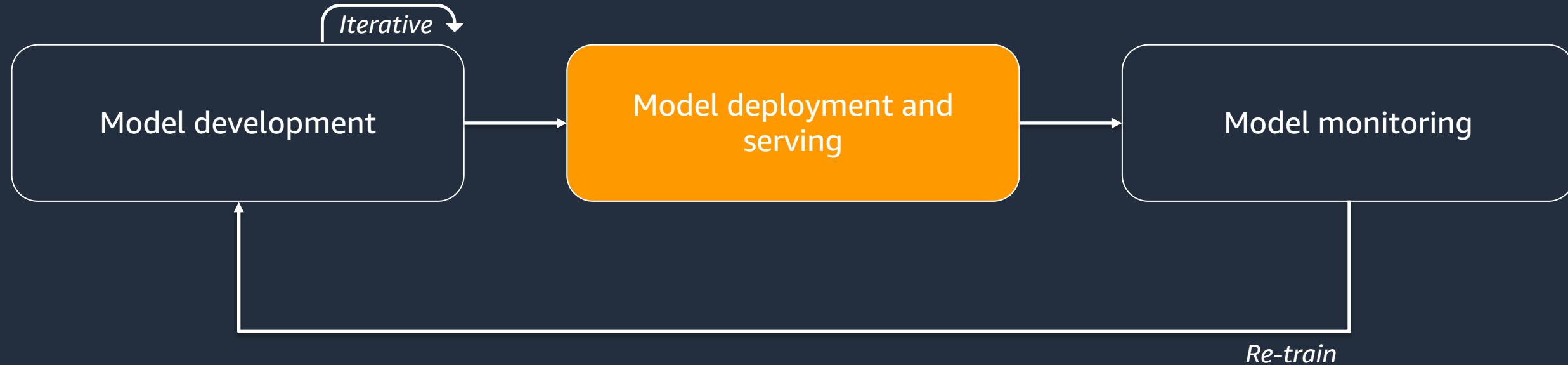


“Only a small fraction of real-world ML systems is composed of the ML code”

source: Hidden Technical Debt in Machine Learning Systems [D. Sculley, & al.] – 2015

<https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf>

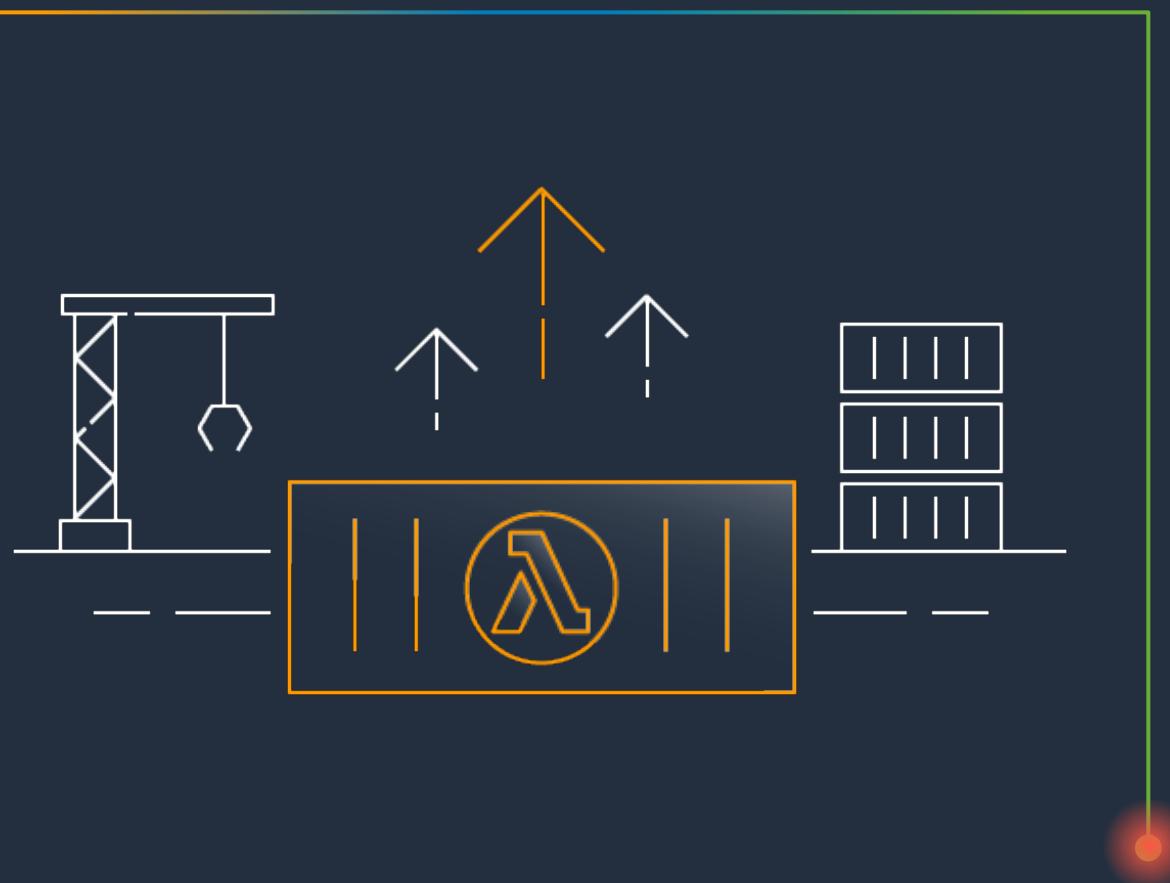
HL Machine learning Solution Lifecycle



*Where to host ?
How to host ?*

The Where: AWS Lambda

AWS Lambda supports packaging and deploying functions as container images



- › Use a consistent set of tools for containers and Lambda-based applications
- › Deploy large applications with AWS-provided or third-party images of up to 10 GB
- › Benefit from subsecond automatic scaling, high availability, 140 native service integrations, pay for use billing model

The How: AWS Serverless Application Model (SAM)

- CloudFormation extension **optimized for serverless**
- Shorthand syntax to express functions, APIs, databases, and event source mappings
- Simplifies IAM policy and Event trigger management
- Model with YAML, deploy using AWS CloudFormation
- Open source!

<https://aws.amazon.com/serverless/sam/>

<https://github.com/awslabs/serverless-application-model>



```
{  
    AWSTemplateFormatVersion: '2010-09-09'  
    Transform: AWS::Serverless-2016-10-31  
  
    Resources:  
        InferenceFunction:  
            Type: AWS::Serverless::Function  
            Properties:  
                PackageType: Image  
                Events:  
                    Inference:  
                        Type: Api  
                        Properties:  
                            Path: /classify_digit  
                            Method: post  
  
        Metadata:  
            Dockerfile: Dockerfile  
            DockerContext: ./app  
            DockerTag: python3.8-v1
```

SAM template transform

Creates:

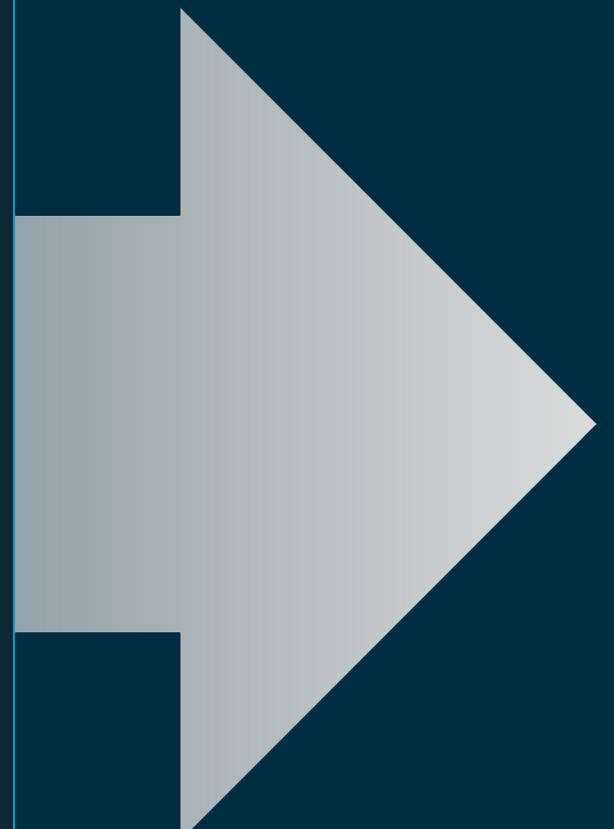
- Lambda function
 - Runtime
 - Execution Policy
 - Code
 - Handler
- API Gateway
 - API endpoint
 - Permissions

Define the container image

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

Globals:
Function:
  Timeout: 50
  MemorySize: 5000
Api:
  BinaryMediaTypes:
    - image/png
    - image/jpg
    - image/jpeg

Resources:
  InferenceFunction:
    Type: AWS::Serverless::Function
    Properties:
      PackageType: Image
  Events:
    Inference:
      Type: Api
      Properties:
        Path: /classify_digit
        Method: post
  Metadata:
    Dockerfile: Dockerfile
    DockerContext: ./app
    DockerTag: python3.8-v1
```



/classify_digit - POST

AWS SAM CLI



- CLI tool for local building, validating, testing of serverless apps
- Works with Lambda functions and “proxy-style” APIs
- Response object and function logs available on your local machine
- Mimic Lambda’s execution environment with Dockers images
 - Emulates timeout, memory limits, runtimes

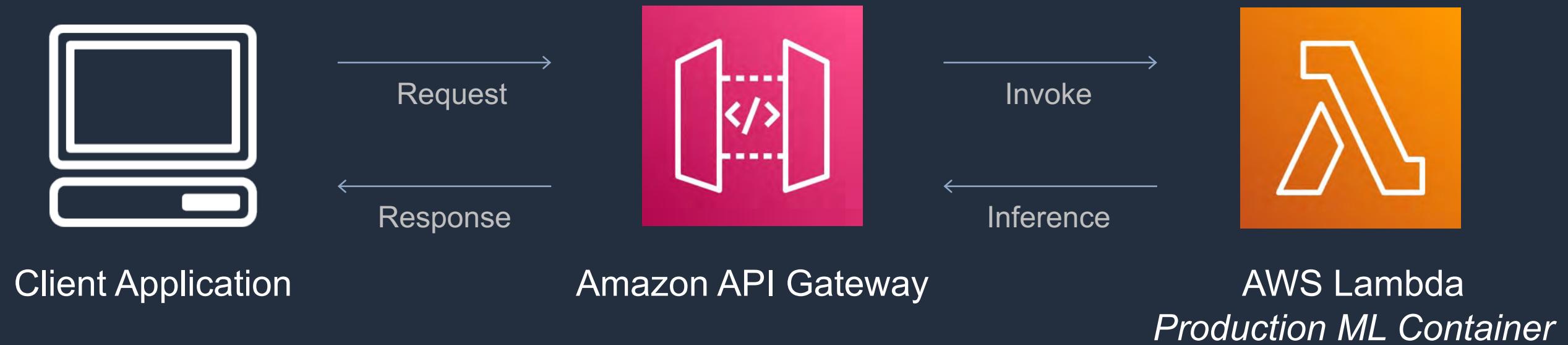
<https://github.com/aws/aws-sam-cli>

Getting Started with SAM CLI



- **sam init**
Generates a preconfigured AWS SAM template and example application code in the language that you choose
- **sam package**
Bundles your application code and dependencies into a "deployment package"
- **sam build**
Prepares it for subsequent steps like deploy or local testing
- **sam deploy**
Deploys your serverless application to the AWS Cloud
- **sam local**
Test your application code locally

Time for action



Steps

1. Generate the template with *sam init*
2. Build the solution with *sam build*
3. Test locally with *sam local*
4. Create a container registry (optional if already exists)
5. Deploy with *sam deploy*
6. Test the deployment

> demo\$

```
> demo$ sam init
```

```
> demo$ sam init
```

Which template source would you like to use?

- 1 - AWS Quick Start Templates
- 2 - Custom Template Location

Choice:

```
> demo$ sam init
```

Which template source would you like to use?

- 1 - AWS Quick Start Templates
- 2 - Custom Template Location

Choice: 1

```
> demo$ sam init  
Which template source would you like to use?  
  1 - AWS Quick Start Templates  
  2 - Custom Template Location  
Choice: 1  
What package type would you like to use?  
  1 - Zip (artifact is a zip uploaded to S3)  
  2 - Image (artifact is an image uploaded to an ECR im  
age repository)  
Package type: █
```

```
> demo$ sam init  
Which template source would you like to use?  
  1 - AWS Quick Start Templates  
  2 - Custom Template Location  
Choice: 1  
What package type would you like to use?  
  1 - Zip (artifact is a zip uploaded to S3)  
  2 - Image (artifact is an image uploaded to an ECR im  
age repository)  
Package type: 2
```

Which base image would you like to use?

- 1 - amazon/nodejs14.x-base
- 2 - amazon/nodejs12.x-base
- 3 - amazon/nodejs10.x-base
- 4 - amazon/python3.8-base
- 5 - amazon/python3.7-base
- 6 - amazon/python3.6-base
- 7 - amazon/python2.7-base
- 8 - amazon/ruby2.7-base
- 9 - amazon/ruby2.5-base
- 10 - amazon/go1.x-base
- 11 - amazon/java11-base
- 12 - amazon/java8.al2-base
- 13 - amazon/java8-base
- 14 - amazon/dotnet5.0-base
- 15 - amazon/dotnetcore3.1-base
- 16 - amazon/dotnetcore2.1-base

Base image:

Which base image would you like to use?

- 1 - amazon/nodejs14.x-base
- 2 - amazon/nodejs12.x-base
- 3 - amazon/nodejs10.x-base
- 4 - amazon/python3.8-base
- 5 - amazon/python3.7-base
- 6 - amazon/python3.6-base
- 7 - amazon/python2.7-base
- 8 - amazon/ruby2.7-base
- 9 - amazon/ruby2.5-base
- 10 - amazon/go1.x-base
- 11 - amazon/java11-base
- 12 - amazon/java8.al2-base
- 13 - amazon/java8-base
- 14 - amazon/dotnet5.0-base
- 15 - amazon/dotnetcore3.1-base
- 16 - amazon/dotnetcore2.1-base

Base image:

Which base image would you like to use?

- 1 - amazon/nodejs14.x-base
- 2 - amazon/nodejs12.x-base
- 3 - amazon/nodejs10.x-base
- 4 - amazon/python3.8-base
- 5 - amazon/python3.7-base
- 6 - amazon/python3.6-base
- 7 - amazon/python2.7-base
- 8 - amazon/ruby2.7-base
- 9 - amazon/ruby2.5-base
- 10 - amazon/go1.x-base
- 11 - amazon/java11-base
- 12 - amazon/java8.al2-base
- 13 - amazon/java8-base
- 14 - amazon/dotnet5.0-base
- 15 - amazon/dotnetcore3.1-base
- 16 - amazon/dotnetcore2.1-base

Base image: 4

2 - amazon/nodejs12.x-base
3 - amazon/nodejs10.x-base
4 - amazon/python3.8-base
5 - amazon/python3.7-base
6 - amazon/python3.6-base
7 - amazon/python2.7-base
8 - amazon/ruby2.7-base
9 - amazon/ruby2.5-base
10 - amazon/go1.x-base
11 - amazon/java11-base
12 - amazon/java8.al2-base
13 - amazon/java8-base
14 - amazon/dotnet5.0-base
15 - amazon/dotnetcore3.1-base
16 - amazon/dotnetcore2.1-base

Base image: 4

Project name [sam-app]:

5 - amazon/python3.7-base
6 - amazon/python3.6-base
7 - amazon/python2.7-base
8 - amazon/ruby2.7-base
9 - amazon/ruby2.5-base
10 - amazon/go1.x-base
11 - amazon/java11-base
12 - amazon/java8.al2-base
13 - amazon/java8-base
14 - amazon/dotnet5.0-base
15 - amazon/dotnetcore3.1-base
16 - amazon/dotnetcore2.1-base

Base image: 4

Project name [sam-app]:

Cloning from <https://github.com/aws/aws-sam-cli-app-templates>

12 - amazon/java8.al2-base
13 - amazon/java8-base
14 - amazon/dotnet5.0-base
15 - amazon/dotnetcore3.1-base
16 - amazon/dotnetcore2.1-base

Base image: 4

Project name [sam-app]:

Cloning from <https://github.com/aws/aws-sam-cli-app-templates>

AWS quick start application templates:

1 - Hello World Lambda Image Example
2 - PyTorch Machine Learning Inference API
3 - Scikit-learn Machine Learning Inference API
4 - Tensorflow Machine Learning Inference API
5 - XGBoost Machine Learning Inference API

Template selection:

12 - amazon/java8.al2-base
13 - amazon/java8-base
14 - amazon/dotnet5.0-base
15 - amazon/dotnetcore3.1-base
16 - amazon/dotnetcore2.1-base

Base image: 4

Project name [sam-app]:

Cloning from <https://github.com/aws/aws-sam-cli-app-templates>

AWS quick start application templates:

1 - Hello World Lambda Image Example
2 - PyTorch Machine Learning Inference API
3 - Scikit-learn Machine Learning Inference API
4 - Tensorflow Machine Learning Inference API
5 - XGBoost Machine Learning Inference API

Template selection: 2

- 2 - PyTorch Machine Learning Inference API
- 3 - Scikit-learn Machine Learning Inference API
- 4 - Tensorflow Machine Learning Inference API
- 5 - XGBoost Machine Learning Inference API

Template selection: 2

Generating application:

Name: sam-app

Base Image: amazon/python3.8-base

Dependency Manager: pip

Output Directory: .

Next steps can be found in the README file at ./sam-app/README.md

> demo\$

- 2 - PyTorch Machine Learning Inference API
- 3 - Scikit-learn Machine Learning Inference API
- 4 - Tensorflow Machine Learning Inference API
- 5 - XGBoost Machine Learning Inference API

Template selection: 2

Generating application:

Name: sam-app

Base Image: amazon/python3.8-base

Dependency Manager: pip

Output Directory: .

Next steps can be found in the README file at ./sam-app/README.md

```
> demo$ cd sam-app/
```

- 3 - Scikit-learn Machine Learning Inference API
- 4 - Tensorflow Machine Learning Inference API
- 5 - XGBoost Machine Learning Inference API

Template selection: 2

Generating application:

Name: sam-app

Base Image: amazon/python3.8-base

Dependency Manager: pip

Output Directory: .

Next steps can be found in the README file at ./sam-app/README.md

```
> demo$ cd sam-app  
> sam-app$ ]
```

```
> demo$ cd sam-app
```

```
> sam-app$ tree
```

```
.
```

- └── README.md
- └── __init__.py
- └── app
 - └── Dockerfile
 - └── __init__.py
 - └── app.py
 - └── model
 - └── requirements.txt
- └── events
 - └── event.json
- └── template.yml
- └── training.ipynb

```
2 directories, 10 files
```

```
> sam-app$
```

```
> demo$ cd sam-app
```

```
> sam-app$ tree
```

```
.
```

- |
 | | README.md
 | | __init__.py
 | | app
 | | | Dockerfile
 | | | __init__.py
 | | | app.py
 | | | model
 | | | requirements.txt
 | | events
 | | | event.json
 | | template.yml
 | | training.ipynb

```
2 directories, 10 files
```

```
> sam-app$
```

```
> demo$ cd sam-app
```

```
> sam-app$ tree
```

```
.
```

- └── README.md
- └── __init__.py
- └── app
 - └── Dockerfile
 - └── __init__.py
 - └── app.py
 - └── model
 - └── requirements.txt
- └── events
 - └── event.json
- └── template.yml
- └── training.ipynb

```
2 directories, 10 files
```

```
> sam-app$
```

```
> demo$ cd sam-app
> sam-app$ tree
.
├── README.md
├── __init__.py
└── app
    ├── Dockerfile
    ├── __init__.py
    ├── app.py
    └── model
        └── requirements.txt
├── events
│   └── event.json
└── template.yml
└── training.ipynb
```

2 directories, 10 files

```
> sam-app$ cat app/Dockerfile
```

```
└── model
    └── requirements.txt
── events
    └── event.json
── template.yml
└── training.ipynb
```

2 directories, 10 files

```
> sam-app$ cat app/Dockerfile
FROM public.ecr.aws/lambda/python:3.8
```

```
COPY app.py requirements.txt ./
```

```
COPY model /opt/ml/model
```

```
RUN python3.8 -m pip install -r requirements.txt -t .
```

```
CMD ["app.lambda_handler"]
```

```
> sam-app$
```

```
└── model
    └── requirements.txt
── events
    └── event.json
── template.yml
└── training.ipynb
```

2 directories, 10 files

```
> sam-app$ cat app/Dockerfile
FROM public.ecr.aws/lambda/python:3.8
```

```
COPY app.py requirements.txt ./
```

```
COPY model /opt/ml/model
```

```
RUN python3.8 -m pip install -r requirements.txt -t .
```

```
CMD ["app.lambda_handler"]
```

```
> sam-app$ less app/app.py
```

```
import torch
import torchvision
import base64
import json
import numpy as np

from PIL import Image
from io import BytesIO

# Preprocessing steps for the image
image_transforms = torchvision.transforms.Compose([torchvision.transforms.ToTensor()])

model_file = '/opt/ml/model'
model = torch.jit.load(model_file)

# Put model in evaluation mode for inferencing
app/app.py
```

```
# Put model in evaluation mode for inferencing
model.eval()

def lambda_handler(event, context):
    image_bytes = event['body'].encode('utf-8')
    image = Image.open(BytesIO(base64.b64decode(image_bytes)))
).convert(mode='L')
    image = image.resize((28, 28))

    probabilities = model.forward(image_transforms(np.array(i
mage)).reshape(-1, 1, 28, 28))
    label = torch.argmax(probabilities).item()

    return {
        'statusCode': 200,
        'body': json.dumps(
:
```

```
image_bytes = event['body'].encode('utf-8')
image = Image.open(BytesIO(base64.b64decode(image_bytes)))
).convert(mode='L')
image = image.resize((28, 28))

probabilities = model.forward(image_transforms(np.array(image)).reshape(-1, 1, 28, 28))
label = torch.argmax(probabilities).item()

return {
    'statusCode': 200,
    'body': json.dumps(
        {
            "predicted_label": label,
        }
    )
}
```

(END)

```
> sam-app$ sam build
```

---> 6cf7a3d1a828

Step 5/5 : CMD ["app.lambda_handler"]

---> Using cache

---> 32b80ac0879e

Successfully built 32b80ac0879e

Successfully tagged inferencefunction:python3.8-v1

Build Succeeded

Built Artifacts : .aws-sam/build

Built Template : .aws-sam/build/template.yaml

Commands you can use next

=====

[*] Invoke Function: sam local invoke

[*] Deploy: sam deploy --guided

> sam-app\$

---> 6cf7a3d1a828

Step 5/5 : CMD ["app.lambda_handler"]

---> Using cache

---> 32b80ac0879e

Successfully built 32b80ac0879e

Successfully tagged inferencefunction:python3.8-v1

Build Succeeded

Built Artifacts : .aws-sam/build

Built Template : .aws-sam/build/template.yaml

Commands you can use next

=====

[*] Invoke Function: sam local invoke

[*] Deploy: sam deploy --guided

> sam-app\$ cat events/event.json

```
jqEvQLET0Kw94DVhAQ5A9kXgGyB5IzEFCD7AZCtk4Qkno7EhtoLAmxhwUYmFg
QcSiooSa0oAdHO+QWVRZnpGSUKjsDQSvxwzEvW01EwMjAyZGAAhTVE9ecb4DB
kFONAiMVeYmDQnwjyN0IsX5yB4RAH AwNPMUJM8w0DA18aA8NRtYLEokS4Ax{i/
sRSnGRtB2NzbGRhYp/3//zmcgYFd k4Hh7/X//39v///7zIGBuZbDAwHvgEAq
4heIf06wrwAAABWZVhJZk1NACoAAAAIAAGHaQAEAAAAAQAAABoAAAAAAA0Shg
AHAAAAEgAAAESgAgAEAAAAAQAAA +gAwAEAAAAAQAAA IQAAAAAQVNDSUkAAAB
TY3J1ZW5zaG90j9MWGwAAAYpJREFUeJy1kM0rRGEUxp/7uu64w+iaQph p8jH1
IwkhH81IaiQLZSE2NpKtZGHh35CUP8AWOwuKQkwmmUz5GLIRjTAxd65zLG5j7
h2WnN3p1/0c3/sC/zLSryv/gPkurX+SAYjr5V0AgJxG7qa+msL6KgBAsxB7KW
vf7F5cJ2ZmMgxKrJVbk1LKo3E0LFNOpV81Z5lqb QtPB+XEcH Ct9AiP4Sf7KK
5eQ4BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFFQ0bEdiqJiJxUPD3vlz9uz
U9igozHQVWFodcT0tLp+Y+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe
5MxBucOzmiKyEoa0Rs2GMD9uMbIgoBuHvc pGR0zQa0rMFelf8GdgTIA1T/WNH
8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97BcHmc r0crh/d6+vU
AqqeDjQBAby/R/d0d/fumBEAuDU50qI+34avHi1Dc+m+mbdt oWSQUumP80XwB
dw yOoPfHcDkAAAAASUVORK5CYII="
}
> sam-app$
```

```
[E]
"resourcePath": "/{proxy+}",
"httpMethod": "POST",
"apiId": "1234567890",
"protocol": "HTTP/1.1"
},
"body": "iVBORw0KGgoAAAANSUhEUgAAABwAAAAAcAAAAABXZoBIAAABQG1DQ1BJQ0MgUHJvZm1
sZQAAeJxjYGDiSSwoyGFhYGDIZSspCnJ3UoiIjFJgf8rAyCDLIMQgxSCZmFxc4BgQ4ANUwgCjUcG3a
0DVQHBZF2TWwmPutxO7FLa+827S7Tlpdx1TPQrgSkktTgbSf4A4IbmqqISBgTEGyFYuLykAsRuAbJE
ioKOA7CkgdjqEvQLEToKw94DVhAQ5A9kXgGyB5IzEFCD7AZCtk4Qkn07EhtoLAmxhwUYmFgQcSiooS
a0oAdHO+QWVRZnpGSUKjsDQSVXwzEvW01EwMjAyZGAAhTVE9ecb4DBkFONAiMVeYmDQnwjyN0IsX5y
B4RAH AwNPMUJM8w0DA18aA8NRtYLEokS4Ax/sRSnGRtB2NzbGRhYp/3//zmcgYFd k4Hh7/X//39v/
//7zIGBuZbDAwHvgEAq4heIf06wrwAAABWZVhJZk1NACoAAAAIAAGHaQAEAAAAAQAABoAAAAAAAO
ShgAHAAAAEgAAAESgAgAEAAAAAQAAA+gAwAEAAAAAQAAAIQAAAAAQVNDSUkAAABTY3J1ZW5zaG90j
9MWGwAAAYpJREFUeJy1kM0rRGEUxp/7uu64w+iaQphp8jH1IwkhH81IaiQLZSE2NpKtZGHh35CUP8A
wOwuKQkwmmUz5GLIRjTAxd65zLG5j7h2WnN3p1/0c3/sC/zLSryv/gPkurX+SAYjr5V0AgJxG7qa+m
sL6KgBAsxB7KWvf7F5cJ2ZmMgxKrJVbk1LKo3E0LFNOpV81Z5lqbQtPB+XEchCt9AiP4Sf7KK5eQ4
BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFfQ0bEdiqJiJxUPD3v1z9uzU9igozHQVWFodcT0tLp+Y
+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe5MxBuc0zmiKyEoa0Rs2GMD9uMbIgoBuHvcpGRO
zQa0rMFel8GdgTIA1T/WNH8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97BcHmcr0crh
/d6+vUAqqeDjQBAby/R/d0d/fumBEAuDU50qI+34avHi1Dc+m+mbdt0WSQUumP80XwBdwyOoPfHcDk
AAAAASUVORK5CYII="
}
> sam-app$
```

sL6KgBAsxB7KWvf7F5cJ2ZmMgxKrJVbk1LKo3E0LFNOpV81Z5lqhbQtPB+XEcHCT9AiP4Sf7KK5eQ4
BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFFQ0bEdiqJiJxUPD3v1z9uzU9igozHQVWFodcT0tLp+Y
+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe5MxBucOzmiKyEoa0Rs2GMD9uMbIgoBuHvcpGRO
zQa0rMFel8GdgTIA1T/WNH8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97BcHmcr0crh
/d6+vUAqqeDjQBAby/R/d0d/fumBEAuDU50qI+34avHi1Dc+m+mbdtowSQUumP80XwBdwyOoPfHcDk
AAAAASUVORK5CYII=""

}

```
> sam-app$ echo "iVBORw0KGgoAAAANSUhEUgAAABwAAAAAcAAAAABXZoBIAAAABQG1DQ1BJQ0MgU  
HJvZmlsZQAAeJxjYGDiSSwoyGFhYGDIZSspCnJ3UoiIjFJgf8rAyCDLIMQgxscZmFxc4BgQ4ANUwgC  
jUcG3a0DVQHBZF2TwwmPutx07FLa+827S7Tlpdx1TPQrgSkktTgbSF4A4IbmqqISBgTEGyFYuLykAs  
RuAbJEioKO7CkgdjqEvQLET0Kw94DVhAQ5A9kXgGyB5IzEFCD7AZCtk4Qkno7EhtoLAmxhwUYmFgQ  
cSiooSa0oAdHO+QWVRZnpGSUKjsDQSVXwzEvW01EwMjAyZGAAhTVE9ecb4DBkFONAiMVeYmDQnwjyN  
0IsX5yB4RAHAwNPMUJM8w0DA18aA8NRtYLEokS4Ax/sRSnGRtB2NzbGRhYp/3//zmcgYFdk4Hh7/X  
//39v///7zIGBuZbDAwHvgEAq4heIf06wrwAAABWZVhJZk1NACoAAAAIAAGHaQAEAAAAAQAABoAA  
AAAAAOShgAHAAAAEgAAAESgAgAEAAAAAQAAAH+gAwAEAAAAAQAAAIQAAAAAQVNDsukAAABTY3J1ZW5  
zaG90j9MWGwAAAYpJREFUeJy1kM0rRGEUxp/7uu64w+iaQphp8jH1IwkhH81IaiQLZSE2NpKtZGHh3  
5CUP8AW0wuKQkwmmUz5GLIRjTAXd65zLG5j7h2WnN3p1/Oc3/sC/zLSryv/gPkurX+SAYjr5V0AgJx  
G7qa+msL6KgBAsxB7KWvf7F5cJ2ZmMgxKrJVbk1LKo3E0LFNOpV81Z5lqhbQtPB+XEcHCT9AiP4Sf7  
KK5eQ4BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFFQ0bEdiqJiJxUPD3v1z9uzU9igozHQVWFodcT  
0tLp+Y+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe5MxBucOzmiKyEoa0Rs2GMD9uMbIgoBuH  
vcpGROzQa0rMFel8GdgTIA1T/WNH8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97BcHm  
cr0crh/d6+vUAqqeDjQBAby/R/d0d/fumBEAuDU50qI+34avHi1Dc+m+mbdtowSQUumP80XwBdwyOo  
PfHcDkAAAAASUVORK5CYII=" | base64 -d > image.jpg && open image.jpg
```

```
BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFfQ0bEdiqJiJxUPD3vlz9uzU9igozHQVWFodcT0tLp+Y
+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe5MxBucOzmiKyEoa0Rs2GMD9uMbIgoBuHvcpGRO
zQa0rMFelf8GdgTIA1T/WNH8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97BcHmcr0crh
/d6+vUA
```

```
-34avHi1Dc+m+mbdt0WSQUumP80XwBdwyOoPfHcDk
AAAAASU
}
> sam-a
HJvZmls
jUcG3a0
RuAbJEi
cSiooS
0IsX5yB4RAHAwNPMUJM8w0DA18aA8NRtYLEokS4Ax/sRSnGRtB2NzbGRhYp/3//zmcgYFd
k4Hh7/X
//39v///7zIGBuZbDAwHvgEAq4heIf06wrwAAABWZVhJZk1NACoAAAAIAAGHaQAEAAAAAQAA
BoAA
AAAAAOShgAHAAAEGAAAESgAgAEAAAAAQAAA+gAwAEAAAAAQAAAIQAAAAAQVND
SUKAAABTY3J1ZW5
zaG90j9MWGwAAAYpJREFUeJy1kM0rRGEUxp/7uu64w+iaQphp8jH1IwkhH81IaiQLZSE2NpKtZG
h3
5CUP8AWOwuKQkwmmUz5GLIRjTAxd65zLG5j7h2WnN3p1/0c3/sC/zLSryv/gPkurX+SAYjr5V0AgJx
G7qa+msL6KgBAsxB7KWvf7F5cJ2ZmMgxKrJVbk1LKo3E0LFNOpV81Z5lqbQtPB+XEchCt9AiP4Sf7
KK5eQ4BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFfQ0bEdiqJiJxUPD3vlz9uzU9igozHQVWFodcT
0tLp+Y+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe5MxBucOzmiKyEoa0Rs2GMD9uMbIgoBuH
vcpGROzQa0rMFelf8GdgTIA1T/WNH8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97BcHm
cr0crh/d6+vUAqqeDjQBAby/R/d0d/fumBEAuDU50qI+34avHi1Dc+m+mbdt0WSQUumP80XwBdwyOo
PfHcDkAAAAASUVORK5CYII=" | base64 -d > image.jpg && open image.jpg
> sam-app$
```



```
> sam-app$ sam local invoke InferenceFunction --event events/event.json
```

```
> sam-app$ sam local invoke InferenceFunction --event events/event.json  
Invoking Container created from inferencefunction:python3.8-v1  
Building image.....  
Skip pulling image and use local one: inferencefunction:rapid-1.26.0.
```

```
> sam-app$ sam local invoke InferenceFunction --event events/event.json  
Invoking Container created from inferencefunction:python3.8-v1  
Building image.....  
Skip pulling image and use local one: inferencefunction:rapid-1.26.0.  
  
START RequestId: bb144c6e-ec50-46bf-8a18-9a3f54eb0e6e Version: $LATEST  
END RequestId: bb144c6e-ec50-46bf-8a18-9a3f54eb0e6e  
REPORT RequestId: bb144c6e-ec50-46bf-8a18-9a3f54eb0e6e Init Duration:  
0.57 ms Duration: 626.64 ms Billed Duration: 700 ms Memory Size: 50  
00 MB Max Memory Used: 5000 MB  
{"statusCode": 200, "body": "{\"predicted_label\": 3"}%  
> sam-app$ █
```

```
> sam-app$ sam local invoke InferenceFunction --event events/event.json
Invoking Container created from inferencefunction:python3.8-v1
Building image......
Skip pulling image and use local one: inferencefunction:rapid-1.26.0.

END RequestId: eebf4192-3a66-4d60-adc2-0ad0717d6e41
REPORT RequestId: eebf4192-3a66-4d60-adc2-0ad0717d6e41  Init Duration:
0.25 ms Duration: 590.06 ms      Billed Duration: 600 ms Memory Size: 50
00 MB  Max Memory Used: 5000 MB
{"statusCode": 200, "body": "{\"predicted_label\": 3"}%}
> sam-app$ clear
```

```
> sam-app$ aws --region <region> ecr get-login-password | docker login  
--username AWS --password-stdin <accountID>.dkr.ecr.<region>.amazonaws.  
com
```

```
> sam-app$ aws --region <region> ecr get-login-password | docker login  
--username AWS --password-stdin <accountID>.dkr.ecr.<region>.amazonaws.  
com  
> sam-app$ aws --region eu-west-1 ecr get-login-password | docker login  
--username AWS --password-stdin [REDACTED].dkr.ecr.eu-west-1.amazona  
ws.com
```

```
> sam-app$ aws --region <region> ecr get-login-password | docker login  
--username AWS --password-stdin <accountID>.dkr.ecr.<region>.amazonaws.  
com  
> sam-app$ aws --region eu-west-1 ecr get-login-password | docker login  
--username AWS --password-stdin [REDACTED].dkr.ecr.eu-west-1.amazona  
ws.com  
Login Succeeded  
> sam-app$ █
```

```
> sam-app$ aws --region <region> ecr get-login-password | docker login  
--username AWS --password-stdin <accountID>.dkr.ecr.<region>.amazonaws.  
com  
> sam-app$ aws --region eu-west-1 ecr get-login-password | docker login  
--username AWS --password-stdin [REDACTED].dkr.ecr.eu-west-1.amazona  
ws.com  
Login Succeeded  
> sam-app$ aws ecr create-repository \  
--repository-name ml-demo \  
--image-tag-mutability MUTABLE \  
--image-scanning-configuration scanOnPush=true
```

```
--image-tag-mutability MUTABLE \
--image-scanning-configuration scanOnPush=true
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:eu-west-1:[REDACTED]:repository/ml-demo",
        "registryId": "512562836817",
        "repositoryName": "ml-demo",
        "repositoryUri": "[REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo",
        "createdAt": "2021-07-16T09:59:43+01:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": true
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
> sam-app$
```

```
--image-tag-mutability MUTABLE \
--image-scanning-configuration scanOnPush=true
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:eu-west-1:[REDACTED]:repository/ml-demo",
        "registryId": "512562836817",
        "repositoryName": "ml-demo",
        "repositoryUri": "[REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo",
        "createdAt": "2021-07-16T09:59:43+01:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": true
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
> sam-app$
```

```
--image-tag-mutability MUTABLE \
--image-scanning-configuration scanOnPush=true
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:eu-west-1:[REDACTED]:repository/ml-demo",
        "registryId": "512562836817",
        "repositoryName": "ml-demo",
        "repositoryUri": "[REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo",
        "createdAt": "2021-07-16T09:59:43+01:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": true
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
> sam-app$ sam deploy --guided
```

```
ml-demo",
  "createdAt": "2021-07-16T09:59:43+01:00",
  "imageTagMutability": "MUTABLE",
  "imageScanningConfiguration": {
    "scanOnPush": true
  },
  "encryptionConfiguration": {
    "encryptionType": "AES256"
  }
}
```

```
> sam-app$ sam deploy --guided
```

Configuring SAM deploy

```
=====
```

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'

```
=====
```

Stack Name [sam-app]: █

```
"createdAt": "2021-07-16T09:59:43+01:00",
"imageTagMutability": "MUTABLE",
"imageScanningConfiguration": {
    "scanOnPush": true
},
"encryptionConfiguration": {
    "encryptionType": "AES256"
}
}
```

```
> sam-app$ sam deploy --guided
```

Configuring SAM deploy

```
=====
```

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'

```
=====
```

Stack Name [sam-app]:

AWS Region [eu-west-1]: █

```
"imageTagMutability": "MUTABLE",
"imageScanningConfiguration": {
    "scanOnPush": true
},
"encryptionConfiguration": {
    "encryptionType": "AES256"
}
}
```

```
> sam-app$ sam deploy --guided
```

Configuring SAM deploy

```
=====
```

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'

```
=====
```

Stack Name [sam-app]:

AWS Region [eu-west-1]:

Image Repository for InferenceFunction:

```
"imageScanningConfiguration": {
    "scanOnPush": true
},
"encryptionConfiguration": {
    "encryptionType": "AES256"
}
}
}

> sam-app$ sam deploy --guided

Configuring SAM deploy
=====
Looking for config file [samconfig.toml] : Not found
Setting default arguments for 'sam deploy'
=====
Stack Name [sam-app]:
AWS Region [eu-west-1]:
Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu
-west-1.amazonaws.com/ml-demo
```

}

```
> sam-app$ sam deploy --guided
```

Configuring SAM deploy

=====

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'

=====

Stack Name [sam-app]:

AWS Region [eu-west-1]:

Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo
inferencefunction:python3.8-v1 to be pushed to [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-python3.8-v1

#Shows you resources changes to be deployed and require a 'Y' to initiate deploy

Confirm changes before deploy [y/N]: [REDACTED]

```
}
```

```
> sam-app$ sam deploy --guided
```

Configuring SAM deploy
=====

Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'
=====

Stack Name [sam-app]:

AWS Region [eu-west-1]:

Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo

 inferencefunction:python3.8-v1 to be pushed to [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-pyton3.8-v1

 #Shows you resources changes to be deployed and require a 'Y' to initiate deploy

Confirm changes before deploy [y/N]: y

Configuring SAM deploy

```
Looking for config file [samconfig.toml] : Not found
```

```
Setting default arguments for 'sam deploy'
```

```
Stack Name [sam-app]:
```

```
AWS Region [eu-west-1]:
```

```
Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu  
-west-1.amazonaws.com/ml-demo
```

```
    inferencefunction:python3.8-v1 to be pushed to [REDACTED].d  
kr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-p  
ython3.8-v1
```

```
#Shows you resources changes to be deployed and require a 'Y' t  
o initiate deploy
```

```
Confirm changes before deploy [y/N]: y
```

```
#SAM needs permission to be able to create roles to connect to  
the resources in your template
```

```
Allow SAM CLI IAM role creation [Y/n]: [REDACTED]
```

```
Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'
=====
Stack Name [sam-app]: AWS Region [eu-west-1]: Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo
inferencefunction:python3.8-v1 to be pushed to [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-python3.8-v1

#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
Confirm changes before deploy [y/N]: y
#SAM needs permission to be able to create roles to connect to the resources in your template
Allow SAM CLI IAM role creation [Y/n]: Y
InferenceFunction may not have authorization defined, Is this okay? [y/N]: [REDACTED]
```

```
Looking for config file [samconfig.toml] : Not found
```

```
Setting default arguments for 'sam deploy'
```

```
=====
```

```
Stack Name [sam-app]:
```

```
AWS Region [eu-west-1]:
```

```
Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu
```

```
-west-1.amazonaws.com/ml-demo
```

```
inferencefunction:python3.8-v1 to be pushed to [REDACTED].d  
kr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-p  
ython3.8-v1
```

```
#Shows you resources changes to be deployed and require a 'Y' t  
o initiate deploy
```

```
Confirm changes before deploy [y/N]: y
```

```
#SAM needs permission to be able to create roles to connect to  
the resources in your template
```

```
Allow SAM CLI IAM role creation [Y/n]: Y
```

```
InferenceFunction may not have authorization defined, Is this o  
kay? [y/N]: y
```

```
Save arguments to configuration file [Y/n]: [REDACTED]
```

```
Looking for config file [samconfig.toml] : Not found
```

```
Setting default arguments for 'sam deploy'
```

```
=====
```

```
Stack Name [sam-app]:
```

```
AWS Region [eu-west-1]:
```

```
Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu
```

```
-west-1.amazonaws.com/ml-demo
```

```
inferencefunction:python3.8-v1 to be pushed to [REDACTED].d  
kr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-p  
ython3.8-v1
```

```
#Shows you resources changes to be deployed and require a 'Y' t  
o initiate deploy
```

```
Confirm changes before deploy [y/N]: y
```

```
#SAM needs permission to be able to create roles to connect to  
the resources in your template
```

```
Allow SAM CLI IAM role creation [Y/n]: Y
```

```
InferenceFunction may not have authorization defined, Is this o  
kay? [y/N]: y
```

```
Save arguments to configuration file [Y/n]: Y
```

```
Setting default arguments for 'sam deploy'  
=====
```

Stack Name [sam-app]:

AWS Region [eu-west-1]:

Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo

inferencefunction:python3.8-v1 to be pushed to [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-python3.8-v1

#Shows you resources changes to be deployed and require a 'Y' to initiate deploy

Confirm changes before deploy [y/N]: y

#SAM needs permission to be able to create roles to connect to the resources in your template

Allow SAM CLI IAM role creation [Y/n]: Y

InferenceFunction may not have authorization defined, Is this okay? [y/N]: y

Save arguments to configuration file [Y/n]: Y

SAM configuration file [samconfig.toml]: [REDACTED]

Setting default arguments for 'sam deploy'

=====

Stack Name [sam-app]:

AWS Region [eu-west-1]:

Image Repository for InferenceFunction: [REDACTED].dkr.ecr.eu

-west-1.amazonaws.com/ml-demo

inferencefunction:python3.8-v1 to be pushed to [REDACTED].d
kr.ecr.eu-west-1.amazonaws.com/ml-demo:inferencefunction-32b80ac0879e-p
ython3.8-v1

#Shows you resources changes to be deployed and require a 'Y' t
o initiate deploy

Confirm changes before deploy [y/N]: y

#SAM needs permission to be able to create roles to connect to
the resources in your template

Allow SAM CLI IAM role creation [Y/n]: Y

**InferenceFunction may not have authorization defined, Is this o
kay?** [y/N]: y

Save arguments to configuration file [Y/n]: Y

SAM configuration file [samconfig.toml]:

SAM configuration environment [default]: [REDACTED]

Saved arguments to config file

Running 'sam deploy' for future deployments will use the parameters saved above.

The above parameters can be changed by modifying samconfig.toml

Learn more about samconfig.toml syntax at

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-config.html>

The push refers to repository [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo]

```
5da5b0147be4: Pushing [>
c1fcc1c8e62d: Pushed
81d8cce4be4a: Pushed
92d02303a737: Pushing [=====>
5c7034f7ba65: Pushing [=>
    ] 4.358MB/201.2MB
a89d1ec7c980: Waiting
726a8c6c8737: Waiting
04e8d343a382: Waiting
```

Saved arguments to config file

Running 'sam deploy' for future deployments will use the parameters saved above.

The above parameters can be changed by modifying samconfig.toml

Learn more about samconfig.toml syntax at

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-config.html>

The push refers to repository [REDACTED].dkr.ecr.eu-west-1.amazonaws.com/ml-demo]

5da5b0147be4: Pushing [>

] 10.56MB/904.1MB

81d8cce4be4a: Pushed

92d02303a737: Pushing [=====

==>] 102.8MB Pushing [=====>

] 54.32MB/201.2MB [=====

==>] 8.173MB/8.167MB

726a8c6c8737: Pushed

04e8d343a382: Pushing [==>

] 11.83MB/294.7MB

Operation	LogicalResourceId	ResourceType
Replacement		
* Modify	InferenceFunction	AWS::Lambda::Function
False		
* Modify	ServerlessRestApi	AWS::ApiGateway::Rest
A False		pi

Changeset created successfully. arn:aws:cloudformation:eu-west-1:
[REDACTED]:changeSet/samcli-deploy1626426471/e05131a5-1bbd-4c23-bdb3-0a4902
daa27a

Previewing CloudFormation changeset before deployment

=====

Deploy this changeset? [y/N]: █

Operation	LogicalResourceId	ResourceType
Replacement		
*	InferenceFunction	AWS::Lambda::Function
False		
*	ServerlessRestApi	AWS::ApiGateway::Rest
A False		pi

Changeset created successfully. arn:aws:cloudformation:eu-west-1:
[REDACTED]:changeSet/samcli-deploy1626426471/e05131a5-1bbd-4c23-bdb3-0a4902
daa27a

Previewing CloudFormation changeset before deployment

=====

Deploy this changeset? [y/N]: y

Changeset created successfully. arn:aws:cloudformation:eu-west-1:
[REDACTED]:changeSet/samcli-deploy1626426471/e05131a5-1bbd-4c23-bdb3-0a4902
daa27a

Previewing CloudFormation changeset before deployment

=====

Deploy this changeset? [y/N]: y

2021-07-16 10:08:34 - Waiting for stack create/update to complete

CloudFormation events from changeset

ResourceStatus	ResourceType	LogicalResourceId
ResourceStatusReason		

=====

Description	Implicit IAM Role created for Inference function
Value	arn:aws:lambda:eu-west-1:[REDACTED]:function:sam-app-InferenceFunction-vXkR50pf5rbv
Key	InferenceFunction
Description	Inference Lambda Function ARN
Value	arn:aws:lambda:eu-west-1:[REDACTED]:function:sam-app-InferenceFunction-vXkR50pf5rbv

Successfully created/updated stack – sam-app in eu-west-1

> sam-app\$ █

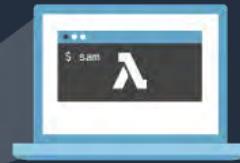
Key	InferenceApi
Description	API Gateway endpoint URL for Prod stage for Inference function
Value	https://████████.execute-api.eu-west-1.amazonaws.com/Prod/classify_digit/
Key	InferenceFunctionIamRole
Description	Implicit IAM Role created for Inference function
Value	arn:aws:lambda:eu-west-1:████████:function:sam-app-InferenceFunction-vXkR50pf5rbv
Key	InferenceFunction
Description	Inference Lambda Function ARN

Key	InferenceApi
Description	API Gateway endpoint URL for Prod stage for Inference function
Value	<code>https://████████.execute-api.eu-west-1.amazonaws.com/Prod/classify_digit/</code>
Key	InferenceFunctionIamRole
Description	Implicit IAM Role created for Inference function
Value	<code>arn:aws:lambda:eu-west-1:████████:function:sam-app-InferenceFunction-vXkR50pf5rbv</code>
Key	InferenceFunction
Description	Inference Lambda Function ARN

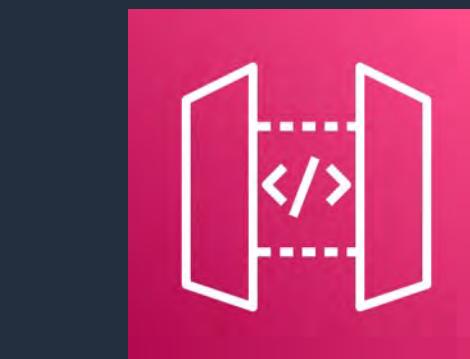
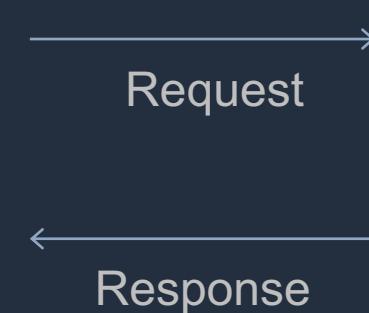
```
> sam-app$ curl -X POST https://████████.execute-api.eu-west-1.amazonaws.com/Prod/classify_digit/ --data "iVBORw0KGgoAAAANSUhEUgAAABwAAAACCAAAAABXZoBIAAABQG1DQ1BJQ0MgUHJvZm1sZQAAeJxjYGDiSSwoyGFhYGDIZSspCnJ3UoiIjFJgf8rAyCDLIMQgxsCZmFxc4BgQ4ANUwgCjUcG3a0DVQHBZF2TwmPutx07FLa+827S7Tlpdx1TPQrgSkktTgbSf4A4IbmqqISBgTEGyFYuLykAsRuAbJEioKOA7CkgdjqqEvQLEToKw94DVhAQ5A9kXgGyB5IzEFCD7AZCtk4Qkn07EhtoLAmxhwUYmFgQcSi0oSa0oAdHO+QWVRZnpGSUKjsDQSVXwzEvW01EwMjAyZGAAhTVE9ecb4DBkFONAiMVeYmDQnwjyN0IsX5yB4RAH AwNPMUJM8w0DA18aA8NRtYLEoks4Ax/sRSnGRtB2NzbGRhYp/3//zmcgYFd k4Hh7/X//39v///7zIGBuZbDAwHvgEAq4heIf06wrwAAABWZVhJZk1NACoAAAAIAAGHaQAEAAAAAQAABoAAAAAAAOShgAHAAAAEgAAAESgAgAEAAAAAQAAA H+gAwAEAAAAQAAAIQAAAAAQVNDSUkAAABTY3J1ZW5zaG90j9MWGwAAAYpJREFUeJy1kM0rRGEUxp/7uu64w+iaQphp8jH1IwkhH81IaiQLZSE2NpKtZGHh35CUP8AWOwuKQkwmmUz5GLIRjTAxd65zLG5j7h2WnN3p1/0c3/sC/zLSryv/gPkurX+SAYjr5V0AgJxG7qa+msL6KgBAsxB7Kwvf7F5cJ2ZmMgxKrJVbk1LKo3E0LFNOpV81Z5lqbQtPB+XEcHct9AiP4Sf7KK5eQ4BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFfQ0bEdiqJiJxUPD3v1z9uzU9igozHQVWFodcT0tLp+Y+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe5MxBuc0zmiKyEoa0Rs2GMD9uMbIgoBuHvcpGROzQa0rMFelf8GdgTIA1T/WNH8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97Bchmc r0crh/d6+vUAqqeDjQBAby/R/d0d/fumBEAuDU50qI+34avHi1Dc+m+mbdtowSQUumP80XwBdwyoOoPfHcDkAAAAASUVORK5CYII="
```

```
> sam-app$ curl -X POST https://████████.execute-api.eu-west-1.amazonaws.com/Prod/classify_digit/ --data "iVBORw0KGgoAAAANSUhEUgAAABwAAAAAcCAAAAABXZoBIAAABQG1DQ1BJQ0MgUHJvZm1sZQAAeJxjYGDiSSwoyGFhYGDIZSspCnJ3UoiIjFJgf8rAyCDLIMQgxsCZmFxc4BgQ4ANUwgCjUcG3a0DVQHBZF2TwmPutx07FLa+827S7Tlpdx1TPQrgSkktTgbSf4A4IbmqqISBgTEGyFYuLykAsRuAbJEioKOA7CkgdjqqEvQLEToKw94DVhAQ5A9kXgGyB5IzEFCD7AZCtk4Qkn07EhtoLAmxhwUYmFgQcSi0oSa0oAdHO+QWVRZnpGSUKjsDQSVXwzEvW01EwMjAyZGAAhTVE9ecb4DBkFONAiMVeYmDQnwjyN0IsX5yB4RAH AwNPMUJM8w0DA18aA8NRtYLEoks4Ax/sRSnGRtB2NzbGRhYp/3//zmcgYFd k4Hh7/X//39v///7zIGBuZbDAwHvgEAq4heIf06wrwAAABWZVhJZk1NACoAAAAIAAGHaQAEAAAAAQAABoAAAAAAAOShgAHAAAAEgAAAESgAgAEAAAAAQAAA H+gAwAEAAAAAQAAAIQAAAAAQVNDSUkAAABTY3J1ZW5zaG90j9MWGwAAAYpJREFUeJy1kM0rRGEUxp/7uu64w+iaQphp8jH1IwkhH81IaiQLZSE2NpKtZGHh35CUP8AWOwuKQkwmmUz5GLIRjTAxd65zLG5j7h2WnN3p1/0c3/sC/zLSryv/gPkurX+SAYjr5V0AgJxG7qa+msL6KgBAsxB7Kwvf7F5cJ2ZmMgxKrJVbk1LKo3E0LFNOpV81Z5lqbQtPB+XEcHct9AiP4Sf7KK5eQ4BAOVbST6ZtAshZSqUDFXKiEWyIABI3oa+gFfQ0bEdiqJiJxUPD3vlz9uzU9igozHQVWFodcT0tLp+Y+/zbySIiJnJOBjIgT1J7x+cgiSRQ2npCT/YIMcWe5MxBuc0zmiKyEoa0Rs2GMD9uMbIgoBuHvcpGROzQa0rMFelf8GdgTIA1T/WNH8BQHK3z3UrZJAF+pcCeSNXBCrtbKuVzrc3ny1Qay3Sp97Bchmc r0crh/d6+vUAqqeDjQBAby/R/d0d/fumBEAuDU50qI+34avHi1Dc+m+mbdtowSQUumP80XwBdwyoOoPfHcDkAAAAASUVORK5CYII=" {"predicted_label": 3}%> sam-app$
```

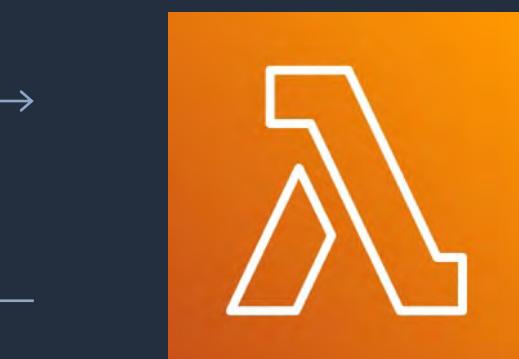
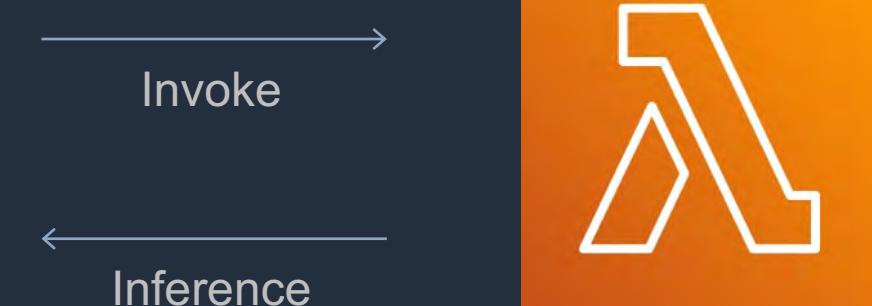
Recap



Client Application



Amazon API Gateway



AWS Lambda
Production ML Container

Conclusions

- Validate your use case
- Be aware of the service quotas
- Test, test, test

Thank you!

<https://aws.amazon.com/serverless/sam/>

<https://aws.amazon.com/lambda/>

<https://aws.amazon.com/api-gateway/>