Develop, Build, and Test AWS Lambda Python Functions using Docker

Michael Wehrle - “mDub”
Lead Cloud Solutions Engineer - Scripps Networks

github.com/sirmdub/pythonLambdaDevWithDocker

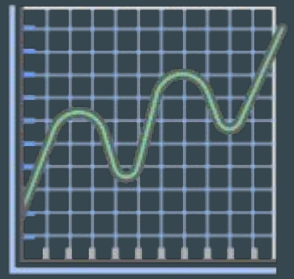
AWS Lambda

Run code without thinking about servers.

Pay for only the compute time you consume.

No Servers To Manage

AWS Lambda automatically runs your code without requiring you to provision or manage servers. Just write the code and upload it to Lambda.



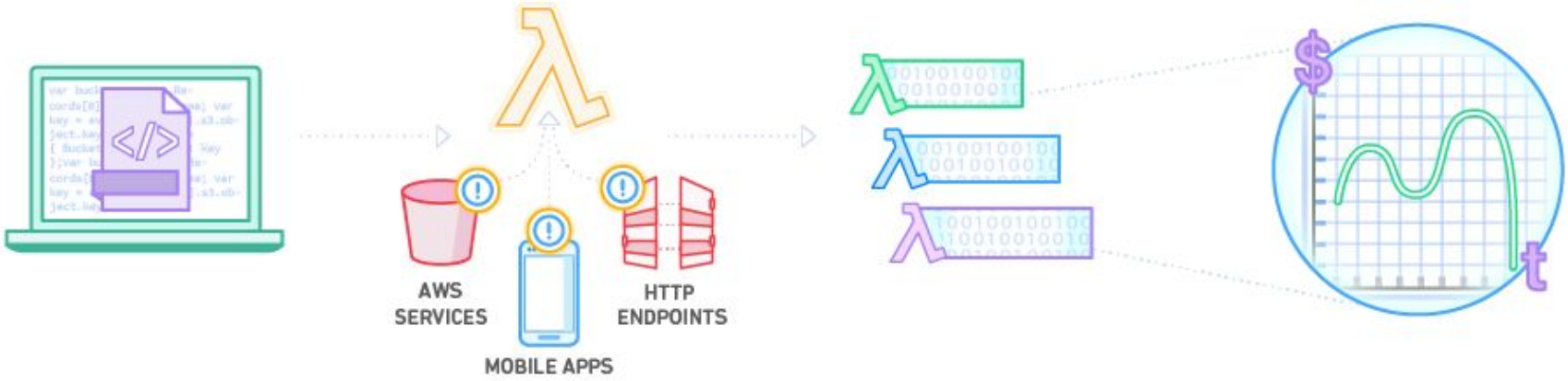
Continuous Scaling

AWS Lambda automatically scales your application by running code in response to each trigger. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload.

Sub-second Metering

With AWS Lambda, you are charged for every 100ms your code executes and the number of times your code is triggered. You don't pay anything when your code isn't running.

How It Works



Upload your code to AWS Lambda

Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity

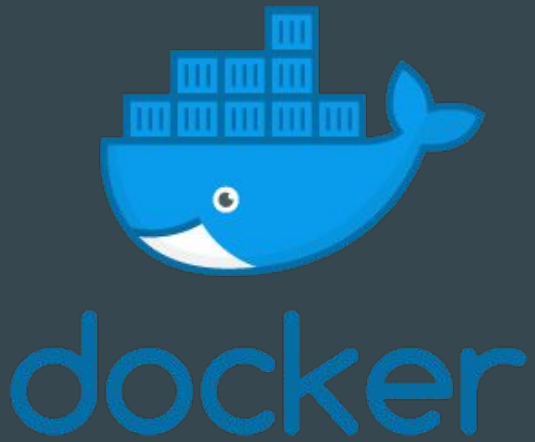
Lambda runs your code only when triggered, using only the compute resources needed

Pay just for the compute time you use

What can you build with AWS Lambda?

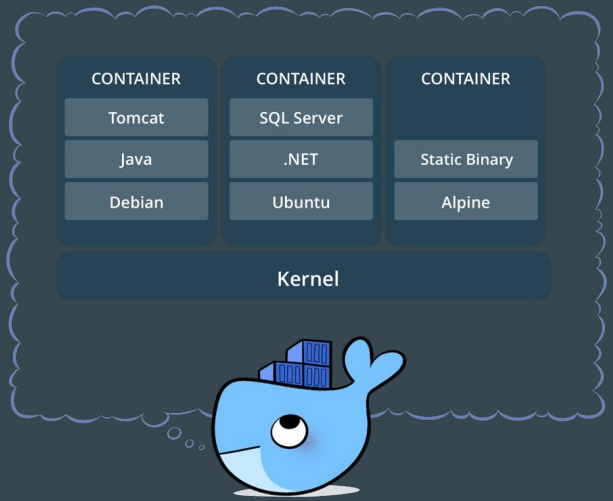
- Data Processing
  - Real-time File Processing
  - Real-time Stream Processing
  - Extract, Transform, Load
- Backends
  - IoT Backends
  - Mobile Backends
  - Web Applications

Docker



What is a container?...

Package software into **a standardized unit** for development, shipment and deployment



...everything required to make a piece of software run is packaged into isolated containers.

...only libraries and settings required to make the software work are needed.

...efficient, lightweight, self-contained systems guarantee that software will always run the same, regardless of where it's deployed.

Docker for Developers

ANY APP, LANGUAGE, OR STACK

Build, test, debug and deploy Linux and Windows Server container apps written in any programming language without risk of incompatibilities or version conflicts.

Docker for Developers

AWESOME DEVELOPER EXPERIENCE

Reduce onboarding time by 65%: Quickly build, test and run complex multi-container apps and stop wasting time installing and maintaining software on servers and developer machines. All dependencies run in containers, eliminating “works on my machine” problems.

“anyone with Docker and an editor installed”

Docker for Developers

BUILT-IN CONTAINER ORCHESTRATION

Docker comes with built-in swarm clustering that's easy to configure. Test and debug apps in environments that mimic production with minimal setup.

AWS Lambda Development with Python

AWS Lambda Development with Python

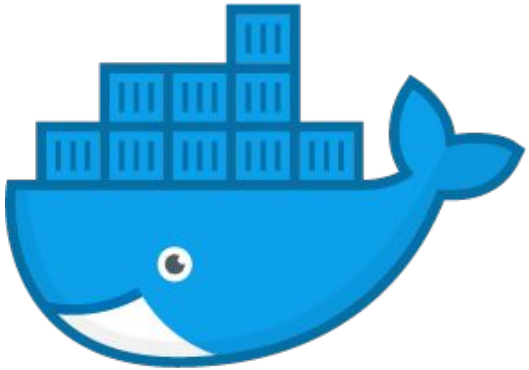
1. Install Python 2.7 and/or 3.6
    a. System default? PATH? Install pyenv?
2. Install pip ‘python get-pip.py’
    a. “Be cautious if you're using a Python install that's managed by your operating system or another package manager. get-pip.py does not coordinate with those tools, and may leave your system in an inconsistent state.”
3. virtualenv?
4. AWS CLI Install (If you want to automate )

OK... I think I’m ready!

AWS Lambda Development with Python

1. Write profitable CODE!
2. Install dependencies in project dir
    a. pip install -r requirements.txt -t .
3. Zip project
    a. zip -r pythonLambdaDevWithDocker.zip . -x *.git* *tests\/* super_secrets.txt
4. Deploy to AWS!
    a. AWS Console Point & Click
        i. OR
    b. AWS CLI / API's

AWS Lambda Development with Python + Docker!



1. ~~Install Python 2.7 and/or 3.6~~
    a. ~~System default? PATH? Install pyenv?~~
2. ~~Install pip 'python get-pip.py'~~
    a. ~~"Be cautious if you're using a Python install that's managed by your operating system or another package manager. get-pip.py does not coordinate with those tools, and may leave your system in an inconsistent state."~~
3. ~~Virtualenv?~~
4. AWS CLI Install

```
$ cat Dockerfile
FROM python:2-alpine

RUN apk add --no-cache zip

RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

COPY requirements.txt /usr/src/app
RUN pip install -r requirements.txt -t .

COPY . /usr/src/app
RUN chmod +x *.py

RUN zip -r pythonLambdaDevWithDocker.zip . -x *.git*
*tests\/* aws_secrets.txt

ENTRYPOINT [ "python" ]
CMD [ "./run.py" ]
```

Which Python would you like?

3.6.2-slim, 3.6-slim, 3-slim, slim (3.6/jessie/slim/Dockerfile)
3.6.2-onbuild, 3.6-onbuild, 3-onbuild, onbuild (3.6/jessie/onbuild/Dockerfile)
3.6.2-alpine3.6, 3.6-alpine3.6, 3-alpine3.6, alpine3.6 (3.6/alpine3.6/Dockerfile)
3.6.2-alpine3.4, 3.6-alpine3.4, 3-alpine3.4, alpine3.4, 3.6.2-alpine, 3.6-alpine, 3-alpine, alpine (3.6/alpine3.4/Dockerfile)
3.6.2-windowsservercore, 3.6-windowsservercore, 3-windowsservercore, windowsservercore (3.6/windows/windowsservercore/Dockerfile)
3.5.3-jessie, 3.5-jessie, 3.5.3, 3.5 (3.5/jessie/Dockerfile)
3.5.3-slim, 3.5-slim (3.5/jessie/slim/Dockerfile)
3.5.3-onbuild, 3.5-onbuild (3.5/jessie/onbuild/Dockerfile)
3.5.3-alpine3.4, 3.5-alpine3.4, 3.5.3-alpine, 3.5-alpine (3.5/alpine3.4/Dockerfile)
3.5.3-windowsservercore, 3.5-windowsservercore (3.5/windows/windowsservercore/Dockerfile)
3.4.6-jessie, 3.4-jessie, 3.4.6, 3.4 (3.4/jessie/Dockerfile)
3.4.6-slim, 3.4-slim (3.4/jessie/slim/Dockerfile)
3.4.6-onbuild, 3.4-onbuild (3.4/jessie/onbuild/Dockerfile)
3.4.6-wheezy, 3.4-wheezy (3.4/wheezy/Dockerfile)
3.4.6-alpine3.4, 3.4-alpine3.4, 3.4.6-alpine, 3.4-alpine (3.4/alpine3.4/Dockerfile)
3.3.6-jessie, 3.3-jessie, 3.3.6, 3.3 (3.3/jessie/Dockerfile)
3.3.6-slim, 3.3-slim (3.3/jessie/slim/Dockerfile)
3.3.6-onbuild, 3.3-onbuild (3.3/jessie/onbuild/Dockerfile)
3.3.6-wheezy, 3.3-wheezy (3.3/wheezy/Dockerfile)
3.3.6-alpine3.4, 3.3-alpine3.4, 3.3.6-alpine, 3.3-alpine (3.3/alpine3.4/Dockerfile)
2.7.13-stretch, 2.7-stretch, 2-stretch (2.7/stretch/Dockerfile)
2.7.13-jessie, 2.7-jessie, 2-jessie, 2.7.13, 2.7, 2 (2.7/jessie/Dockerfile)
2.7.13-slim, 2.7-slim, 2-slim (2.7/jessie/slim/Dockerfile)
2.7.13-onbuild, 2.7-onbuild, 2-onbuild (2.7/jessie/onbuild/Dockerfile)
2.7.13-wheezy, 2.7-wheezy, 2-wheezy (2.7/wheezy/Dockerfile)
2.7.13-alpine3.6, 2.7-alpine3.6, 2-alpine3.6 (2.7/alpine3.6/Dockerfile)
2.7.13-alpine3.4, 2.7-alpine3.4, 2-alpine3.4, 2.7.13-alpine, 2.7-alpine, 2-alpine (2.7/alpine3.4/Dockerfile)
2.7.13-windowsservercore, 2.7-windowsservercore, 2-windowsservercore (2.7/windows/windowsservercore/Dockerfile)

1. ~~Install Python 2.7 and/or 3.6~~
    a. ~~System default? PATH? Install pyenv?~~
2. ~~Install pip 'python get-pip.py'~~
    a. ~~"Be cautious if you're using a Python install that's managed by your operating system or another package manager. get-pip.py does not coordinate with those tools, and may leave your system in an inconsistent state."~~
3. ~~Virtualenv?~~
4. ~~AWS CLI Install~~

```
$ cat Dockerfile.deploy
FROM python:2-alpine

RUN pip install awscli
COPY --from=pldwd:build
/usr/src/app/pythonLambdaDevWithDocker.zip /

CMD aws lambda update-function-code --function-name
pythonLambdaDevWithDocker --zip-file
fileb://pythonLambdaDevWithDocker.zip
```

1. Write profitable CODE!
2. ~~Install dependencies in project dir~~
   a. ~~pip install -r requirements.txt -t .~~
3. ~~Zip project~~
   a. ~~zip -r pythonLambdaDevWithDocker.zip . -x *.git* *tests\/* super_secrets.txt~~
4. Deploy to AWS!
   a. AWS Console Point & Click
      i. OR
   b. AWS CLI / API's

```
$ cat Dockerfile
FROM python:2-alpine

RUN apk add --no-cache zip

RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

COPY requirements.txt /usr/src/app
RUN pip install -r requirements.txt -t .

COPY . /usr/src/app
RUN chmod +x *.py

RUN zip -r pythonLambdaDevWithDocker.zip . -x *.git*
*tests\/* aws_secrets.txt

ENTRYPOINT [ "python" ]
CMD [ "./run.py" ]
```

1. Write profitable CODE!
2. ~~Install dependencies in project dir~~
    a. ~~pip install -r requirements.txt -t .~~
3. ~~Zip project~~
    a. ~~zip -r pythonLambdaDevWithDocker.zip . -x *.git* *tests\/* super_secrets.txt~~
4. ~~Deploy to AWS!~~
    a. ~~AWS Console Point & Click~~
        i. ~~OR~~
    b. ~~AWS CLI / API's~~

$ cat Dockerfile.deploy
FROM python:2-alpine

RUN pip install awscli
COPY --from=pldwd:build
/usr/src/app/pythonLambdaDevWithDocker.zip /

CMD aws lambda update-function-code --function-name pythonLambdaDevWithDocker --zip-file fileb://pythonLambdaDevWithDocker.zip

github.com/sirmdub/pythonLambdaDevWithDocker

Michael Wehrle - “mDub”

Lead Cloud Solutions Engineer - Scripps Networks

mdub@snilabs.com