AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

AIM409-R3

Accelerate deep learning with cloud custom environments

Nabil Ezzarhouni (he/him)
Sr AI/ML Solutions Architect
AWS – Strategic Accounts

Qingwei Li (he/him)
Sr AI/ML Solutions Architect
AWS – Financial Services

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Challenges & opportunities

Deep learning environments on AWS

- AMIs for EC2
- Deep learning containers for ECS and EKS
- Deep learning on SageMaker

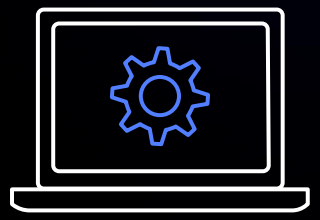
Demo

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

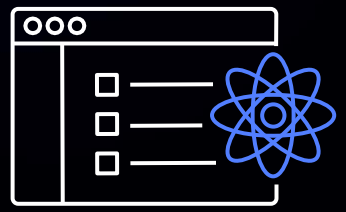
Challenges & opportunities

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

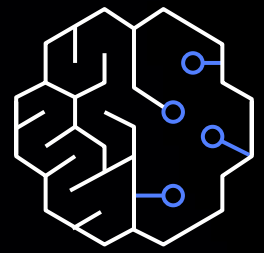
Why customization?



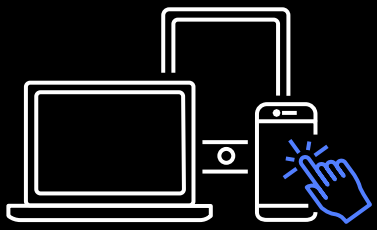
Performance



Experimentation



Pretrained model



Devices support



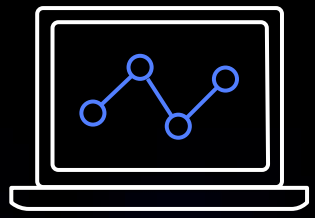
Launch to production



Elasticity



Cost optimization



Model velocity

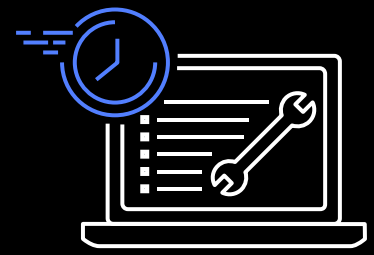
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Challenges and opportunities



- OS configuration
- GPU & CPU optimization
- Distributed training & inference



- Accelerate experimentations
- Faster response to changes
- Cost optimization

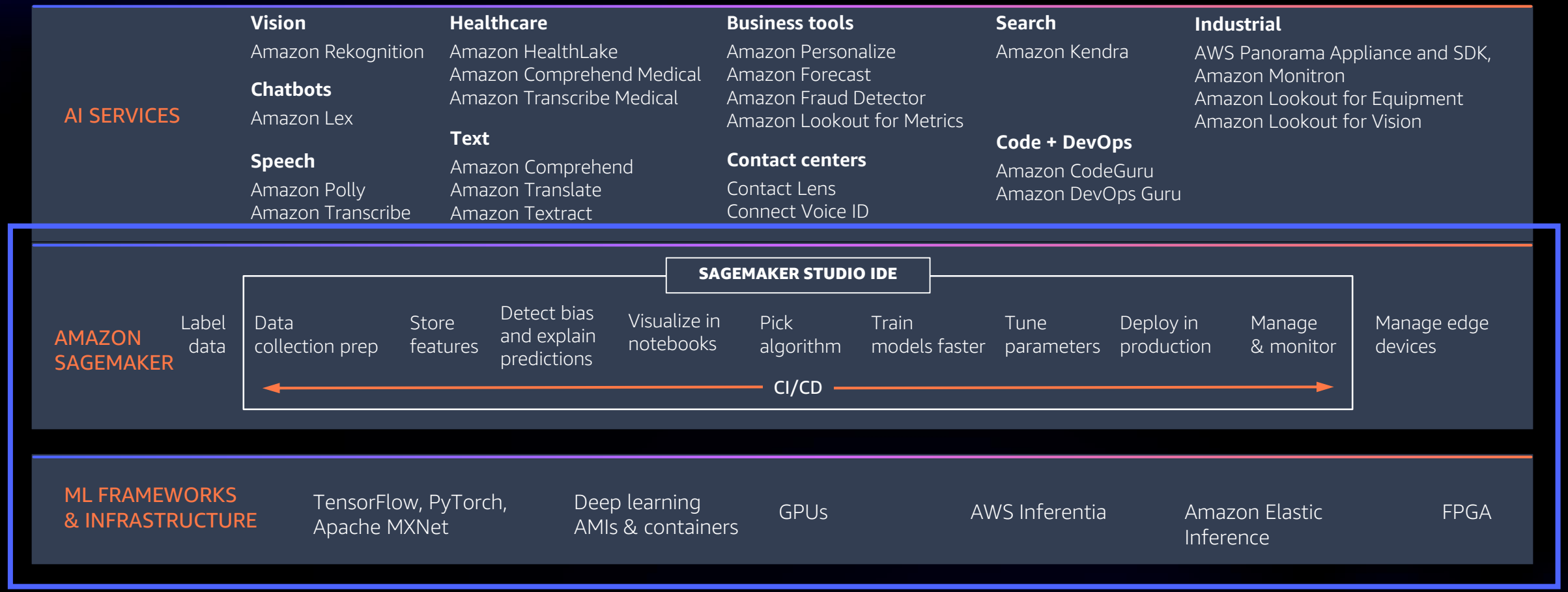
aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning on AWS

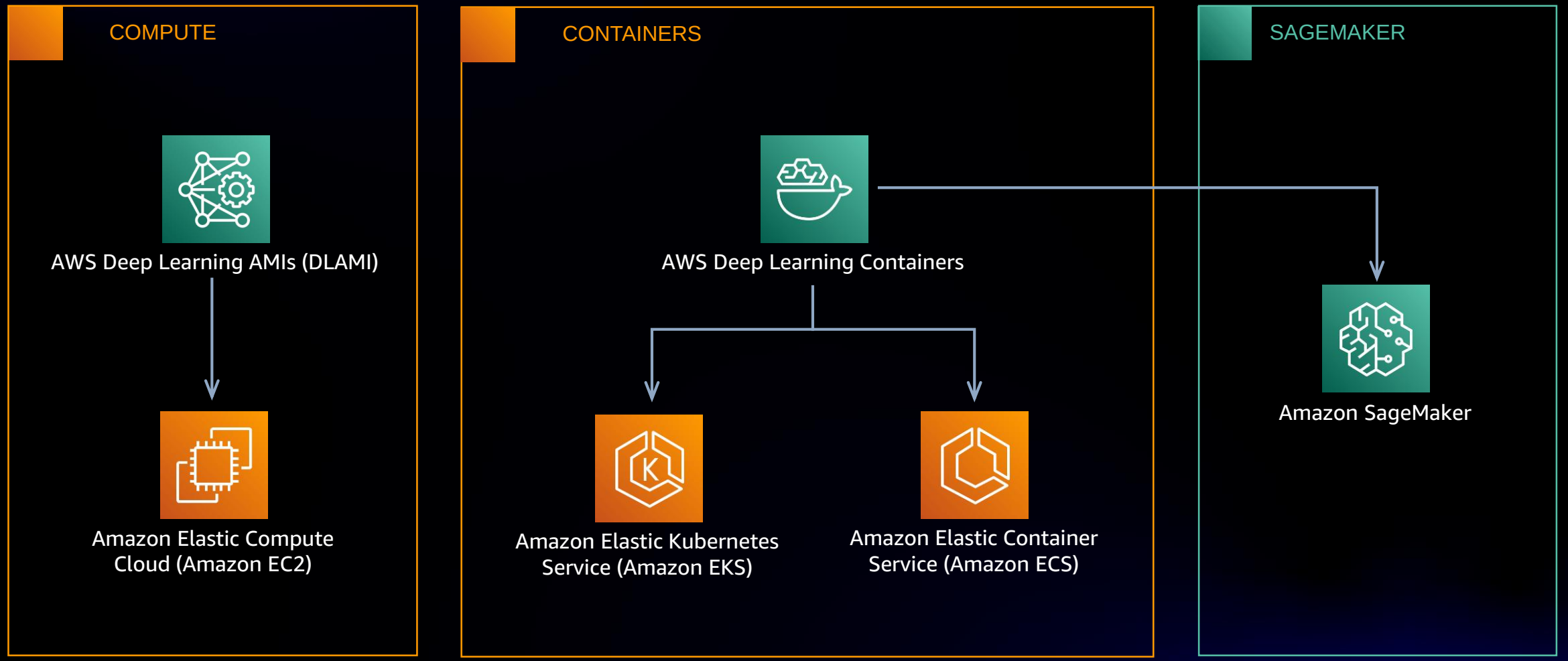
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The AWS ML stack



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning environments on AWS



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning AMIs

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning AMIs on EC2



AWS Deep Learning AMIs

Base

- Amazon Linux & Ubuntu
- Optimized CPU & GPU
- Clean slate for framework installation

Conda

- Amazon Linux, Ubuntu, & Windows
- Optimized virtual environment
- Iterative experimentation

aws

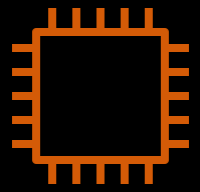
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning AMIs on EC2

EC2 instance



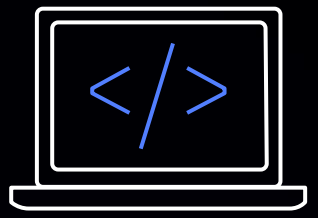
AWS Deep Learning AMIs



Instance



Snapshot



User data script

```
#!/bin/bash
yum update –y
pip install torch
```



Jupyter notebook
Conda



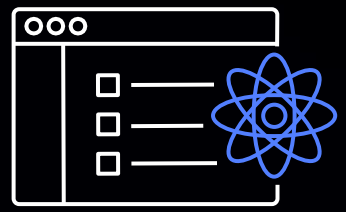
Amazon Elastic Block Store (Amazon EBS)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

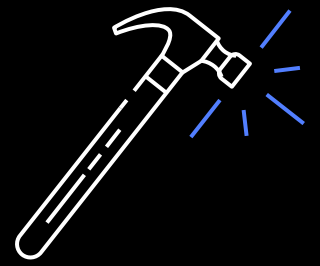
Deep learning AMIs on EC2



Compute & storage augmentation



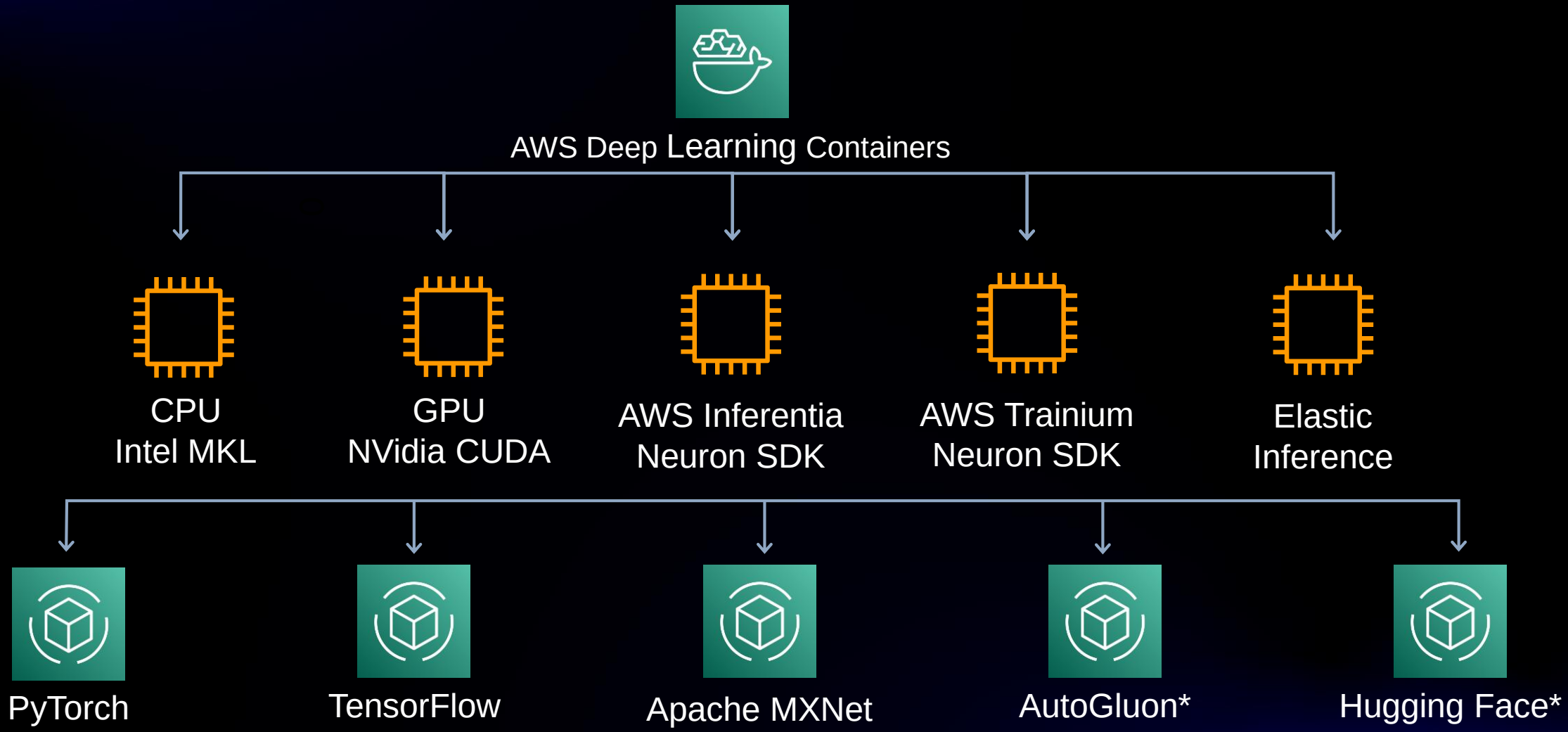
Frictionless experimentation



High level of control

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

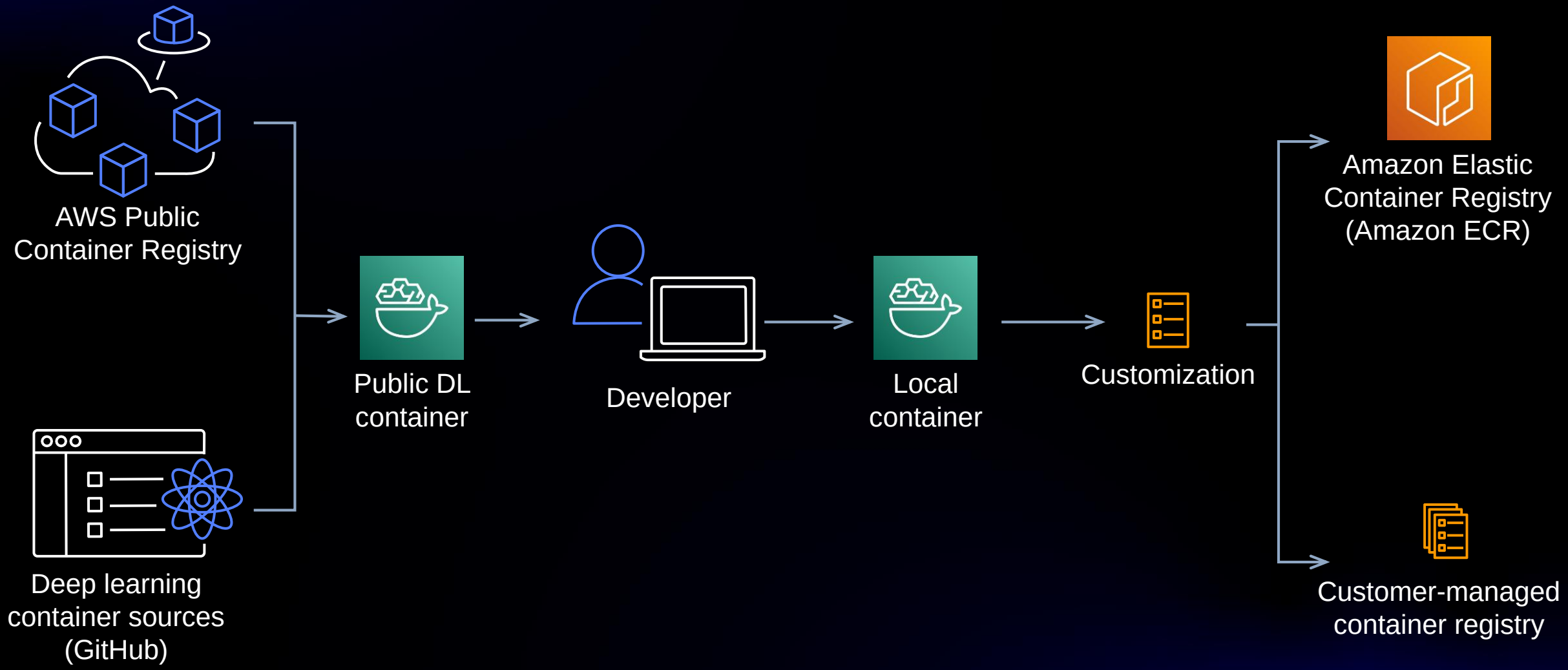
Deep learning containers



** Different inference & training containers*

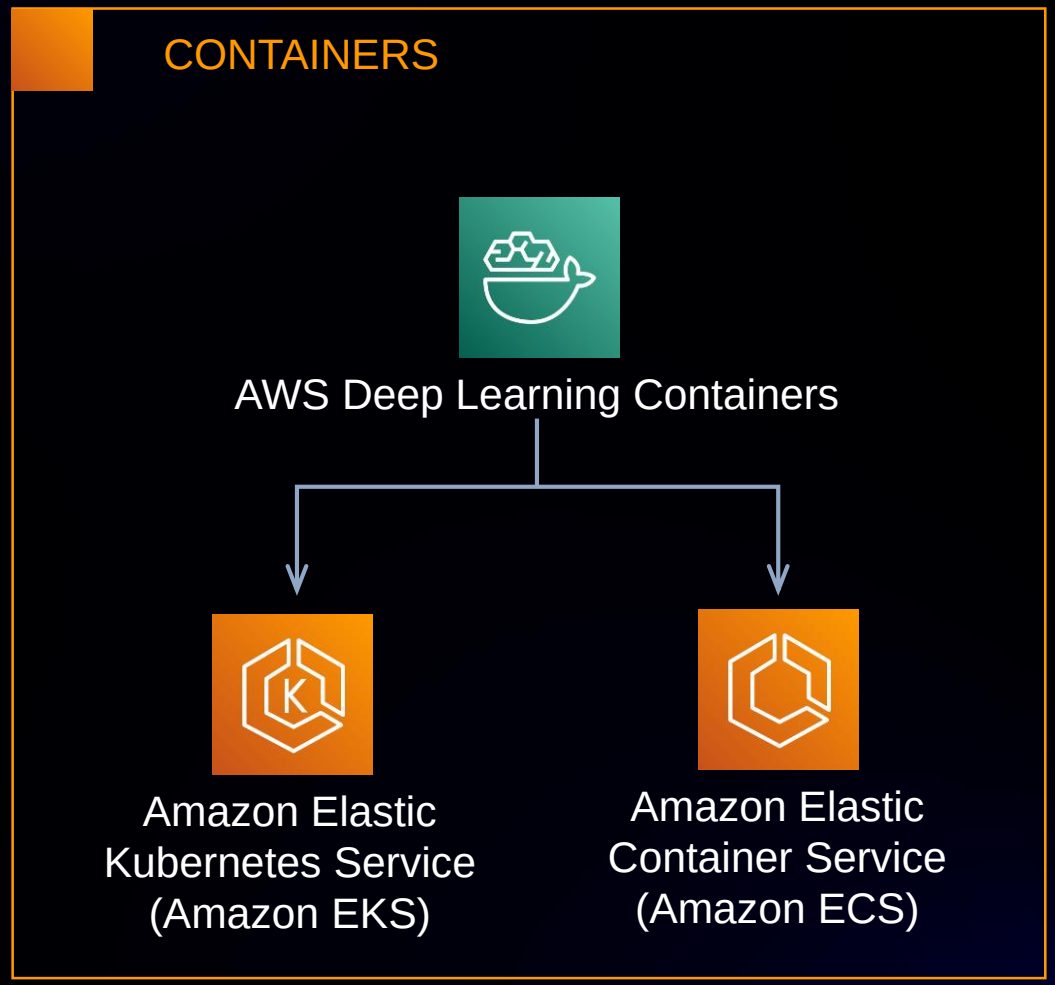
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning containers



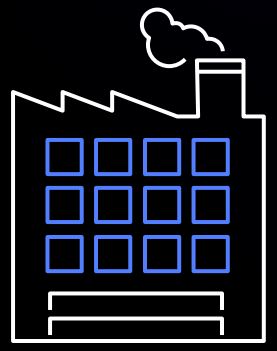
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning containers deployment



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

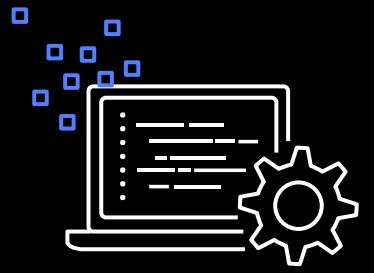
Deep learning containers on EC2



Existing container workload



Cloud elasticity



Agility

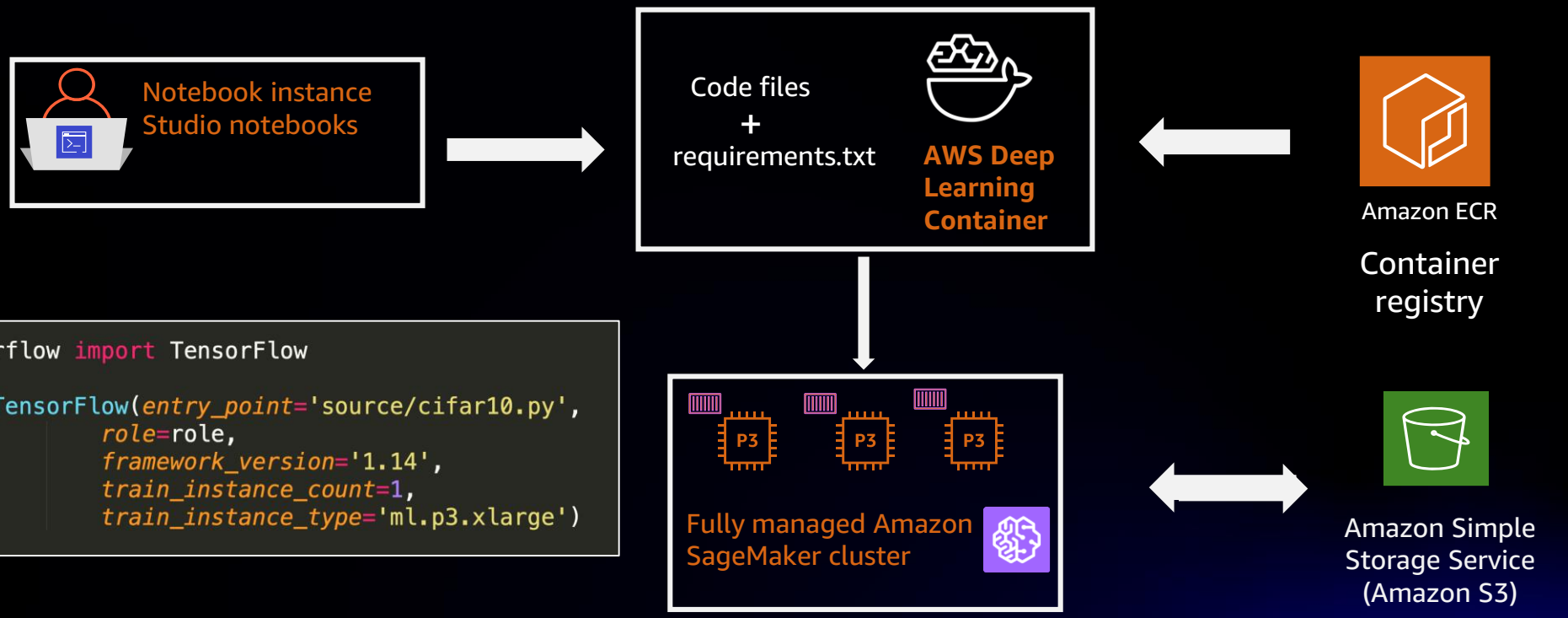
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deep learning on SageMaker

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker

- Amazon SageMaker is a fully managed machine learning service
- Customizing DL containers is NOT needed in most cases




```
from sagemaker.tensorflow import TensorFlow

cifar10_estimator = TensorFlow(entry_point='source/cifar10.py',
                               role=role,
                               framework_version='1.14',
                               train_instance_count=1,
                               train_instance_type='ml.p3.xlarge')
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Customizing DL containers on SageMaker

Extend a prebuilt container

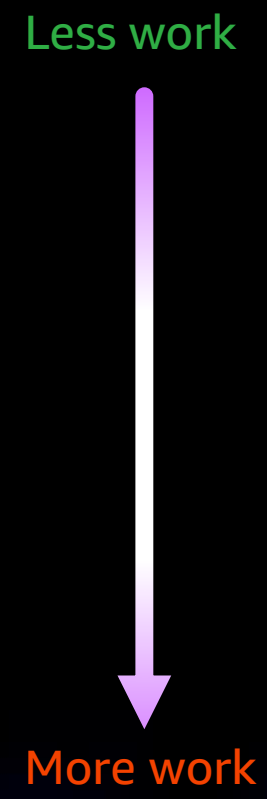
- You want to install additional dependencies

Adapt your own container to work with SageMaker

- You want to use an existing Docker image with SageMaker

Create a container with your own algorithm and models

- Neither the prebuilt nor your existing container meets your need



aws

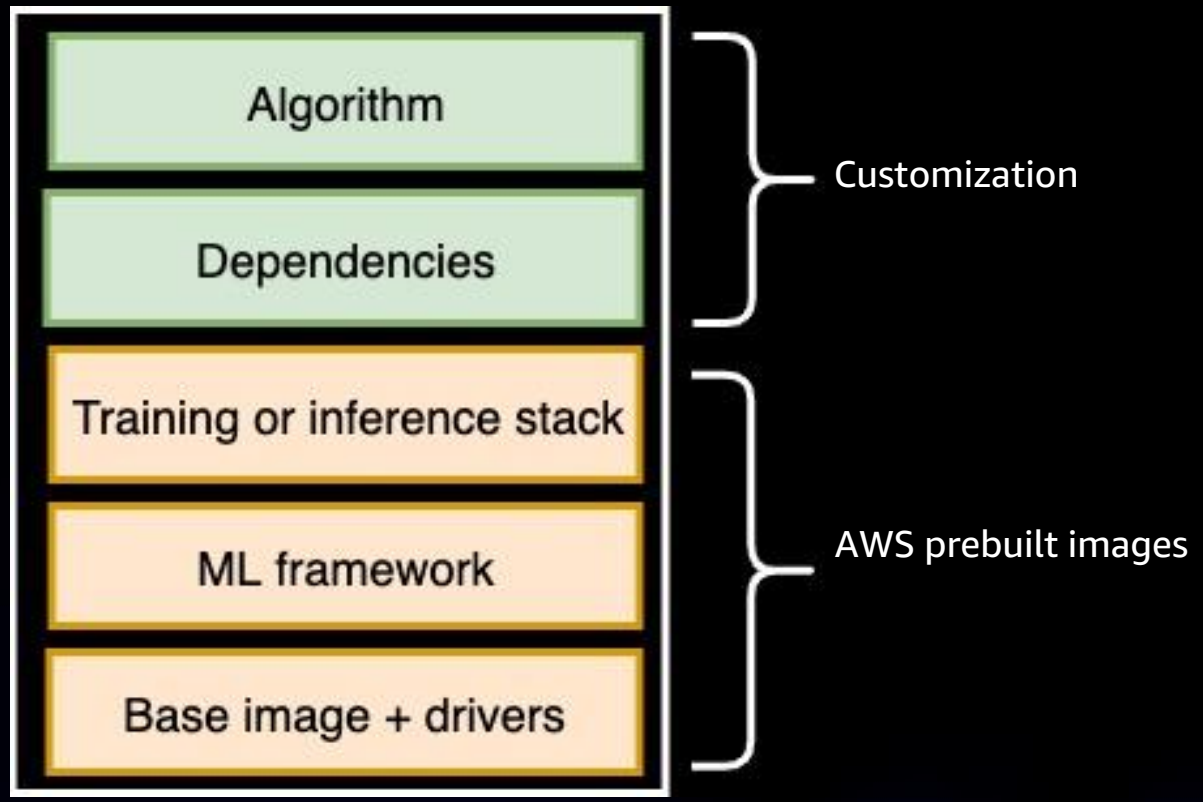
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Extend a prebuilt container

- Prebuilt containers by AWS

TensorFlow mxnet PyTorch Chainer

- Benefit of extending a prebuilt container
  - Leverage SageMaker provided deep learning libraries and settings



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

*PyTorch, the PyTorch logo and any related marks are trademarks of Facebook Inc.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How to customize prebuilt container

- Set required environment variables with SageMaker containers

```
ENV SAGEMAKER_SUBMIT_DIRECTORY /opt/ml/code
ENV SAGEMAKER_PROGRAM cifar10.py
```

- Install additional libraries and package your code

```
RUN apt-get install <library>
```

Remember pip install is supported in requirements.txt

- Build the container and push to Amazon ECR

```
docker build -t ${image_name} .
docker push ${fullname}
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

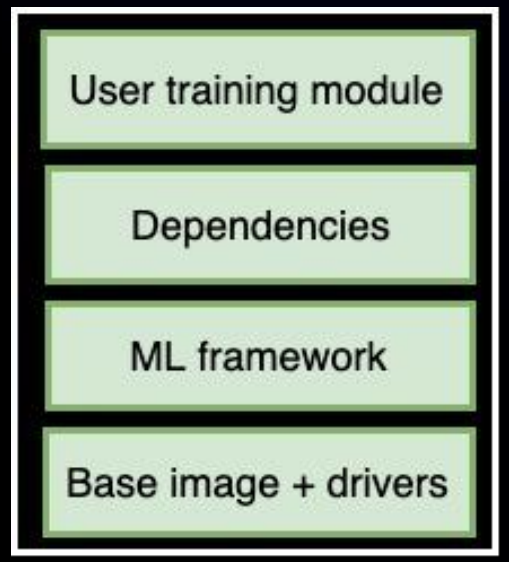
Customize your own container for SageMaker

- When
  - “I cannot use AWS deep learning containers for regulatory reasons”
  - “We use highly customized based image and deep learning frameworks”
  - “I want to reuse my existing/working container and bring it to SageMaker with little adaption”

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Customize your own container for training

The bare minimum

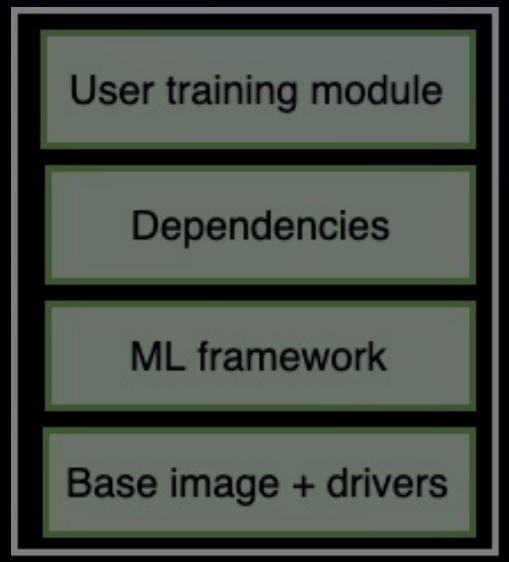



```
FROM <user image>
COPY train.py /
ENTRYPOINT ["python", "train.py"]
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

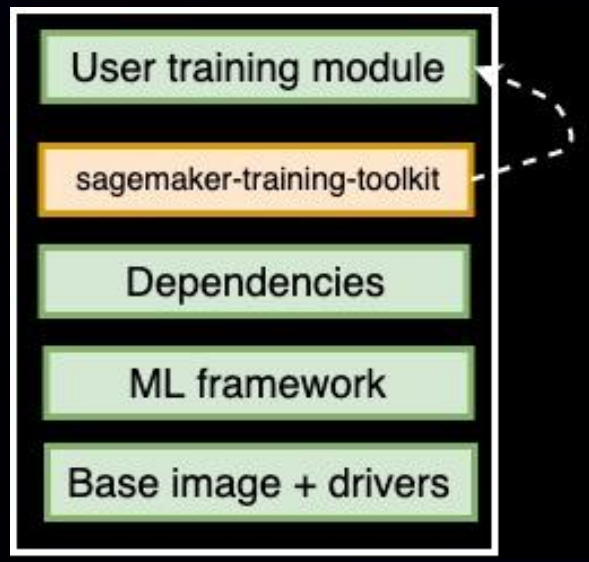
Customize your own container for training

The bare minimum




```
FROM <user image>
COPY train.py /
ENTRYPOINT ["python", "train.py"]
```

Script in container




```
FROM <user image>

RUN pip install sagemaker-training

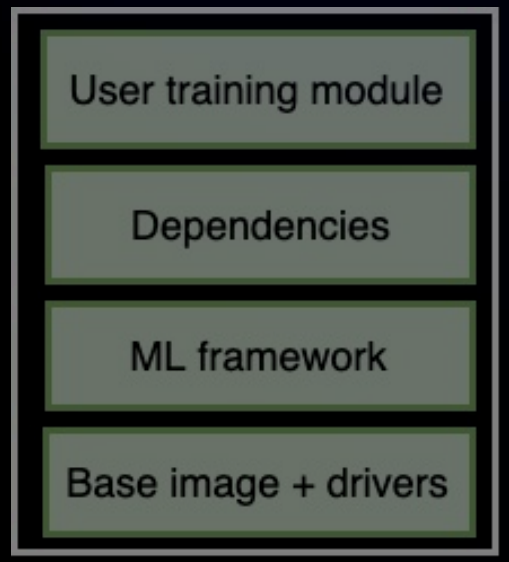
COPY code/* /opt/ml/code/

ENV SAGEMAKER_PROGRAM train.py
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

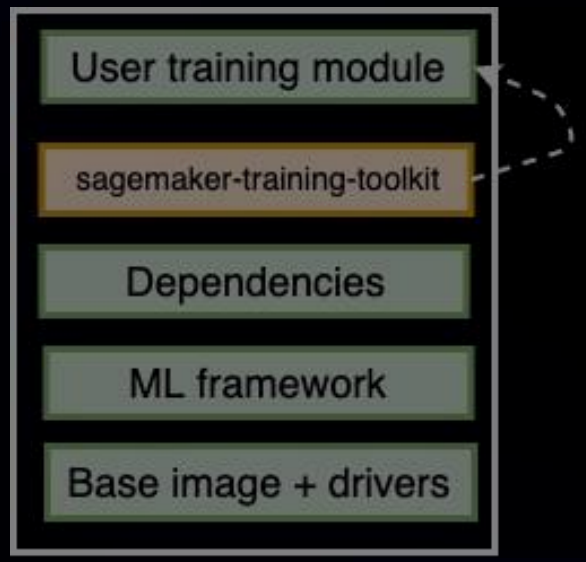
Customize your own container for training

The bare minimum




```
FROM <user image>
COPY train.py /
ENTRYPOINT ["python", "train.py"]
```

Script in container



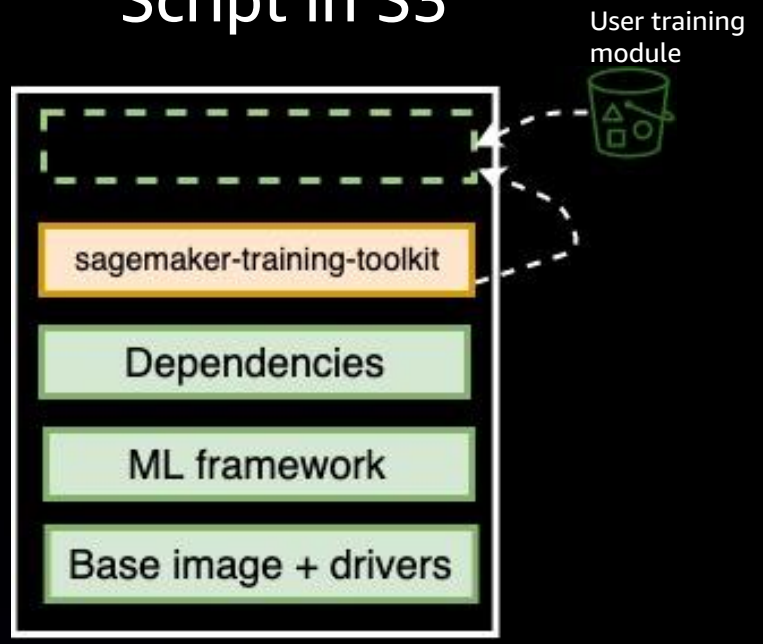

```
FROM <user image>

RUN pip install sagemaker-training

COPY code/* /opt/ml/code/

ENV SAGEMAKER_PROGRAM train.py
```

Script in S3



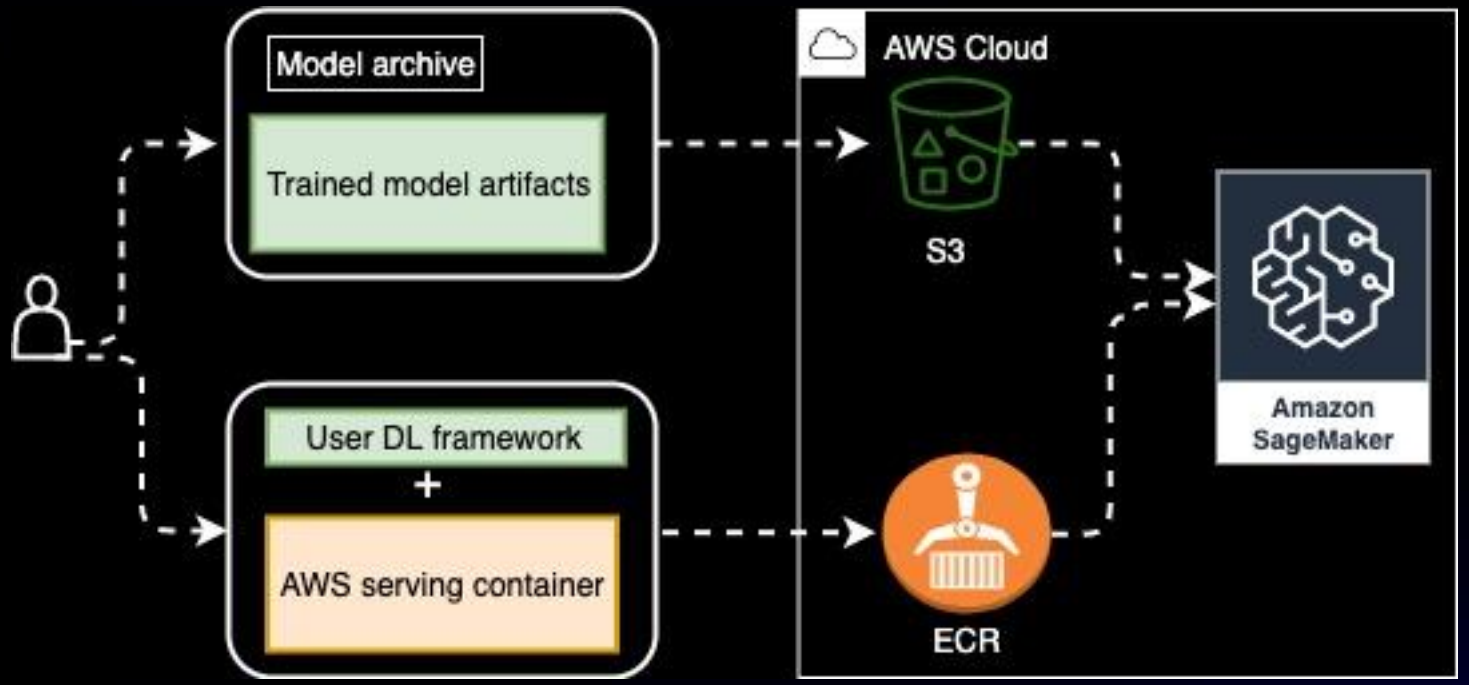

```
FROM <user image>
RUN pip install sagemaker-training
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Customize your own container for inferencing

Prior to your customization:

- Consider the shortcut: AWS serving container + user dependencies



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

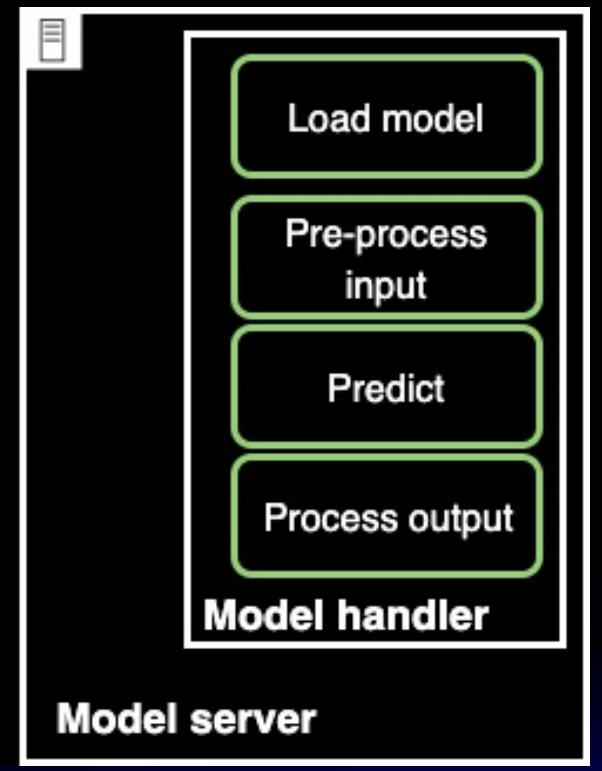
Customize your own container for inferencing

SageMaker Inference Toolkit

- Implements a model serving stack for any Docker container
- Framework specific toolkit available

How

- Create an inference handler
  - Load model
  - Pre-process input data
  - Get predictions
  - Process the output data
- Start model server



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Customize your own container for inferencing

handler.py

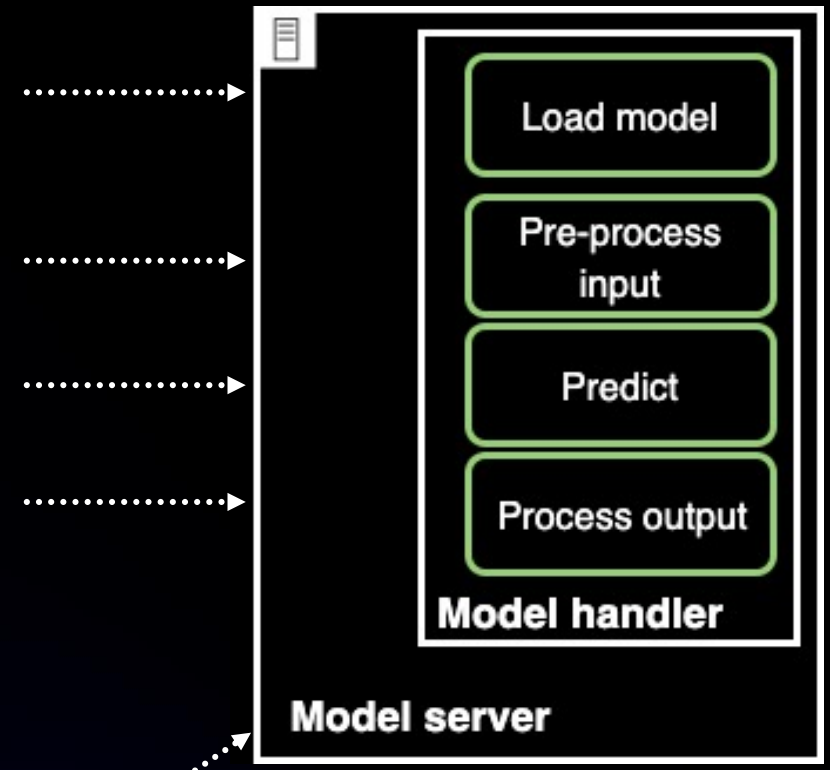
```
class ModelHandler(object):
    def initialize(self, context):
    # Initialize model. This will be called during model loading time

    def preprocess(self, request):
    # Take the input data and pre-process it make it inference ready

    def inference(self, model_input):
    # Do some inference call to engine here and return output

    def postprocess(self, inference_output):
    # Take output from network and post-process to desired format

    def handle(self, data, context):
        model_input = self.preprocess(data)
        model_out = self.inference(model_input)
        return self.postprocess(model_out)
```



serve.py

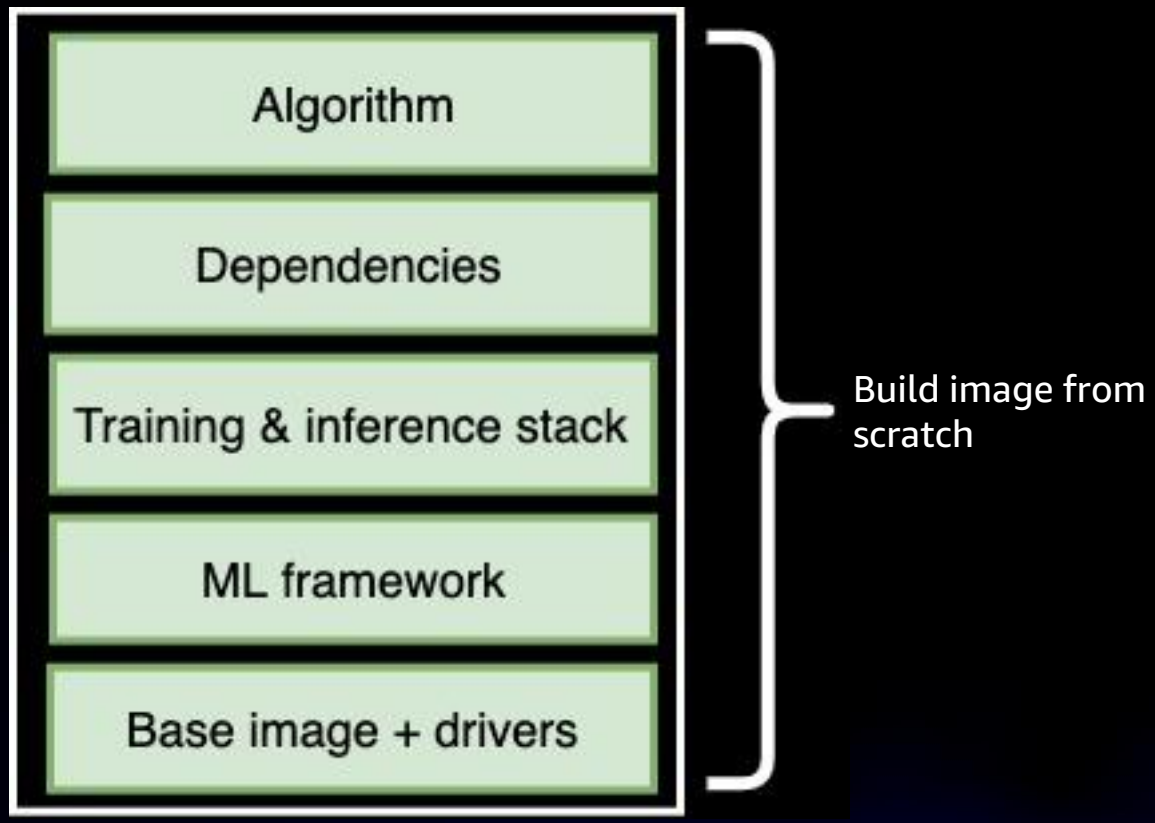
```
from sagemaker_inference import model_server
model_server.start_model_server(handler_service="handler.py:handle")
```

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Create a container with your own algorithm

- When?
  - “I don’t have an existing container”
  - “My ML model is not based on any of the frameworks supported on AWS”



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Create a container with your own algorithm

Training

- How SageMaker runs your image

```
docker run image train
```

- ENTRYPOINT

```
ENTRYPOINT ["python", ”train.py"]
```

- Reserved paths:

```
/opt/ml
|------/input
       |-----/config
       |------------/hyperparameters.json
       |------------/…
       |-----/data
|------/output
       |------/failure
       |------/data
|------/model
```

Inference

- How SageMaker runs your image

```
docker run image serve
```

- ENTRYPOINT

```
ENTRYPOINT ["python", ”serve.py"]
```

- Implement:
  - /invocations
  - /ping

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

reinvent2021.notebook.us-east-2.sagemaker.aws/notebooks/pytorch_extending_our_containers_2021-11-11/pytorch_extending_our_containers-demo.ipynb

Apps Sales-BD AI-ML-Ops d2l.ai draw SFDC Dive into Deep Le... AWS Market Intelli... fast.ai releases ne... PM SageMaker Reading List

jupyter pytorch_extending_our_containers-demo (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted conda_pytorch_p37

Run Markdown nbdiff

Notebook available at:

https://github.com/aws/amazon-sagemaker-examples/tree/master/advanced_functionality/pytorch_extending_our_containers

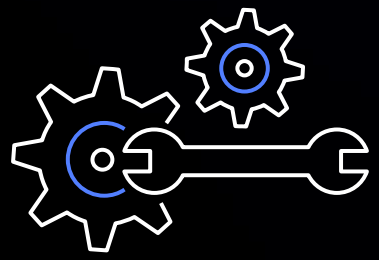
Extending our PyTorch containers ¶

When should I extend a SageMaker container?

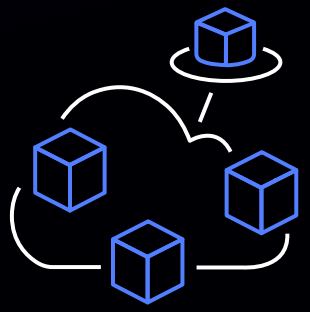
You may not need to create a container to bring your own code to Amazon SageMaker. When you are using a framework such as TensorFlow, MXNet, PyTorch or Chainer that has direct support in SageMaker, you can simply supply the Python code that implements your algorithm using the SDK entry points for that framework.

Some of the reasons to extend a SageMaker deep learning framework container are:

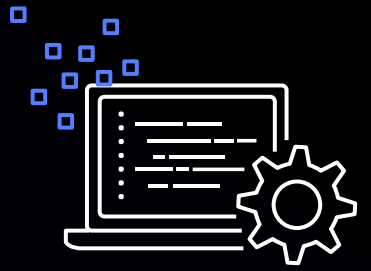
Conclusion



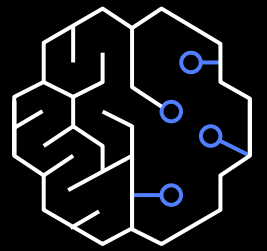
Build



Deploy



Develop



Any models

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Nabil Ezzarhouni
nabz@amazon.com

Qingwei Li
qqnl@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.