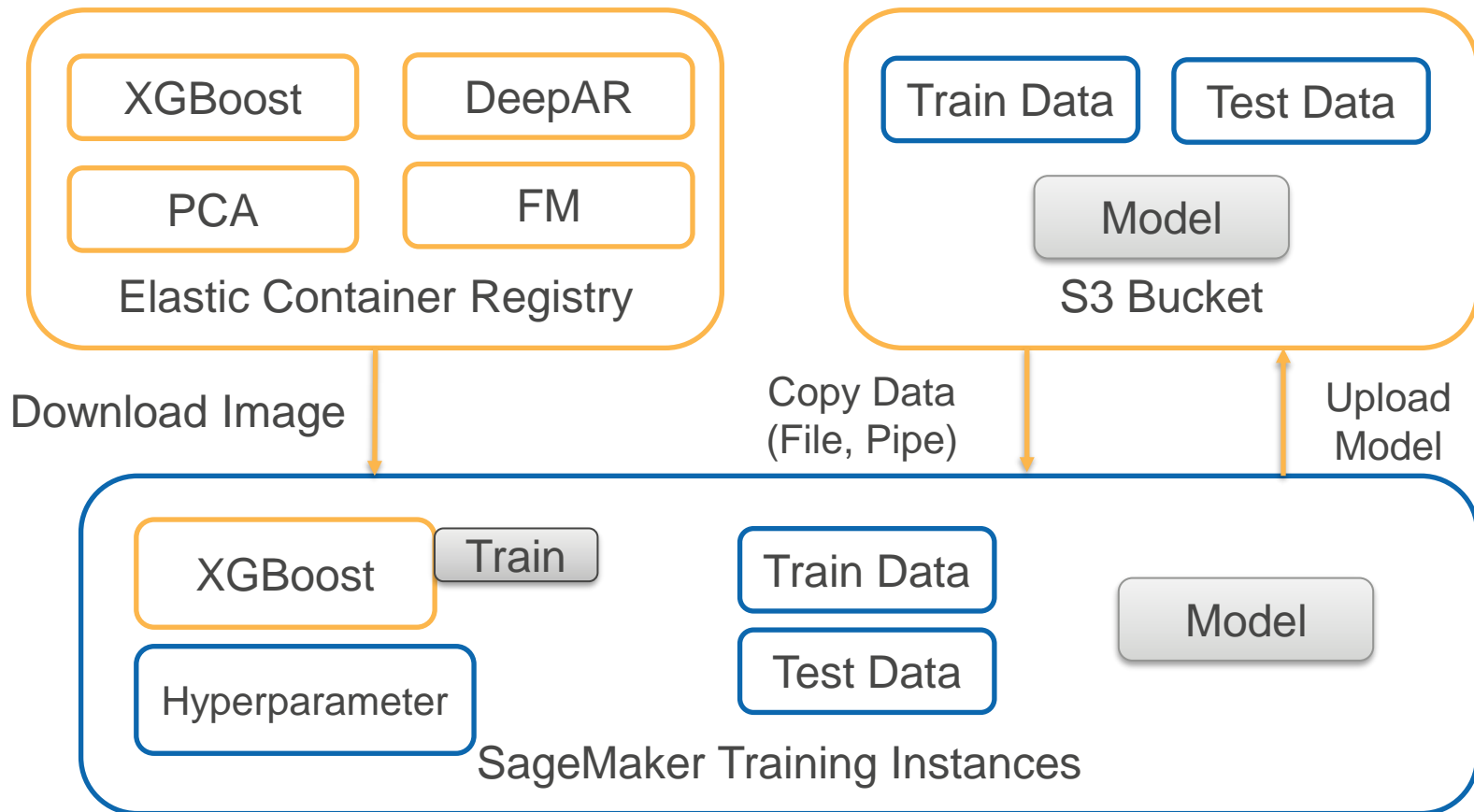# Bring Your Own Algorithms

AWS SageMaker
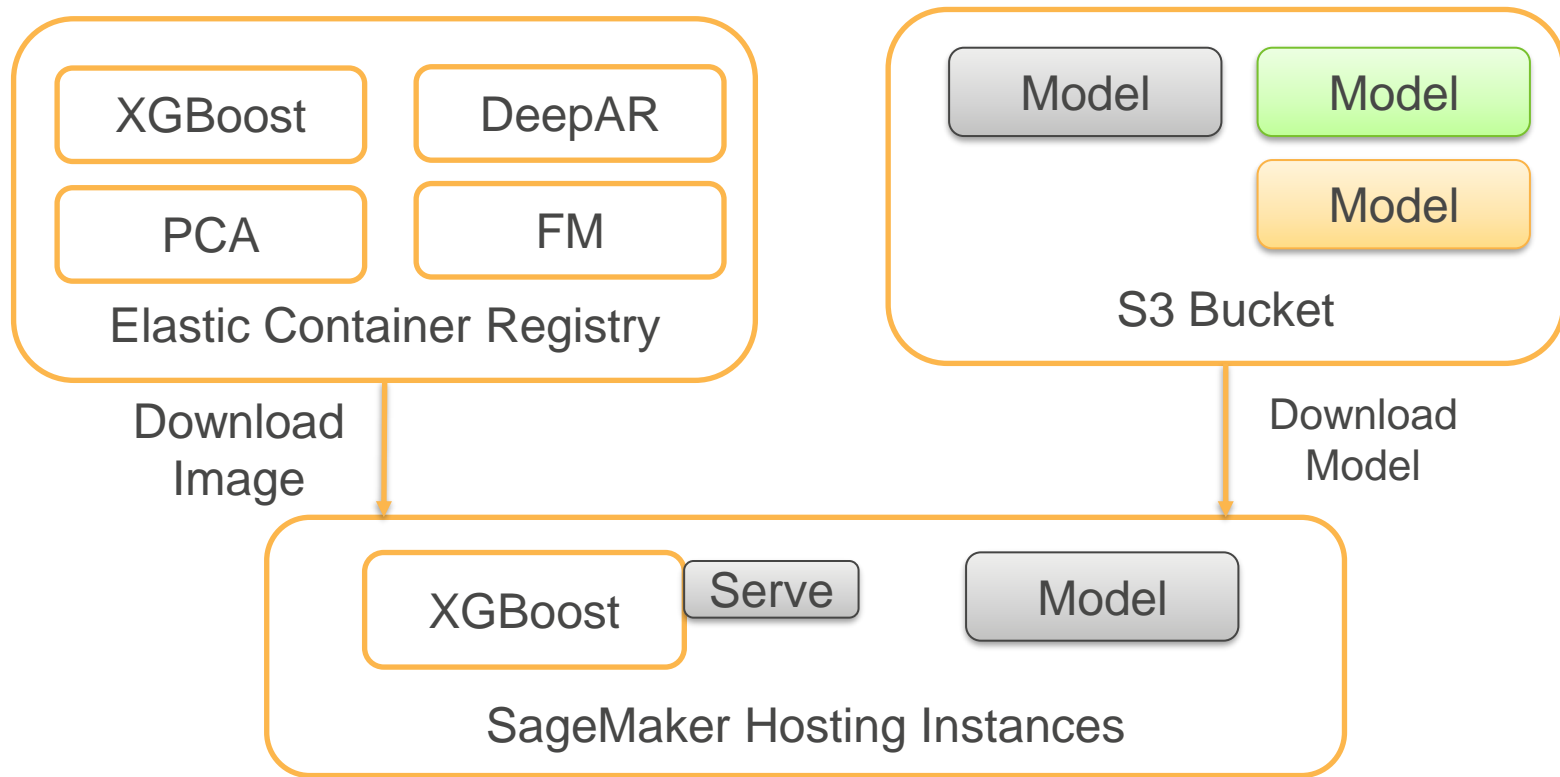
SageMaker – Training and Hosting

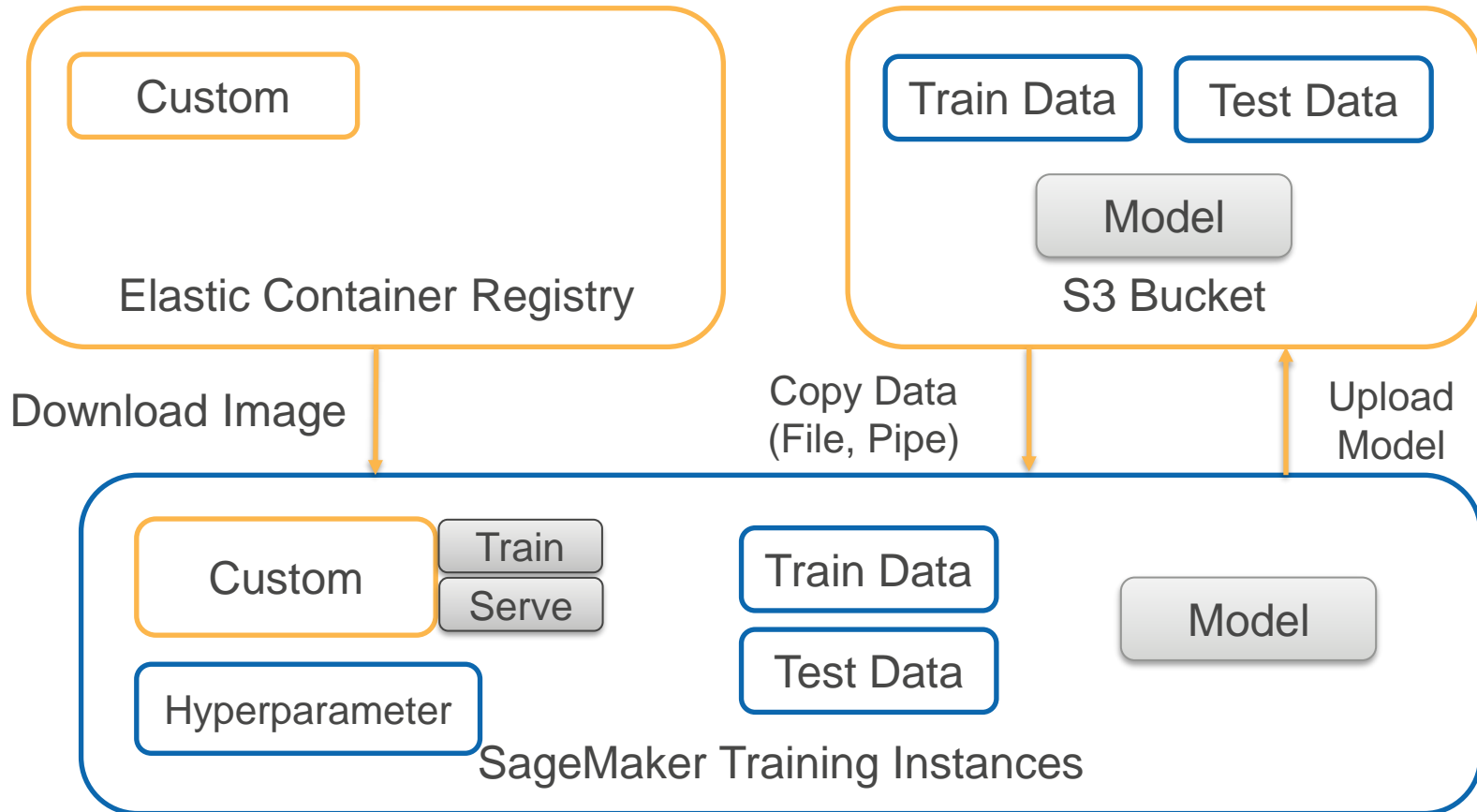| Options	Usage Scenario
Built-in Algorithms	Training algorithms provided by SageMaker Easy to use and scale Optimized for AWS Cloud
Pre-built Container Images	Supports popular frameworks like MxNet, TensorFlow, scikit-learn, PyTorch Flexibility to use wide selection of algorithms
Extend Pre-built Container Images	Extend pre-built container images to your needs
Custom Container Images	Use different language and framework

Built-in Algorithms - Training

Elastic Container Registry

- XGBoost
- DeepAR
- PCA
- FM

S3 Bucket

- Train Data
- Test Data
- Model

Download Image

Copy Data (File, Pipe)

Upload Model

SageMaker Training Instances

- XGBoost — Train
- Hyperparameter
- Train Data
- Test Data
- Model

Built-in Algorithms – Hosting (Realtime, Batch)

XGBoost | DeepAR

PCA | FM

Elastic Container Registry

Model | Model

Model

S3 Bucket

Download Image

Download Model

XGBoost | Serve | Model

SageMaker Hosting Instances

Custom Image – Training, Hosting

Custom

Elastic Container Registry

Train Data   Test Data

Model

S3 Bucket

Download Image

Copy Data
(File, Pipe)

Upload
Model

Custom   Train
Serve

Hyperparameter

Train Data

Test Data

Model

SageMaker Training Instances

Framework - Training

Framework - Hosting

SKLearn

TensorFlow

PyTorch

MxNet

Elastic Container Registry

Model

S3 Bucket

Download Image

Download Model

TensorFlow

Serve Script File

Model

Local Mode

SageMaker Hosting Instances
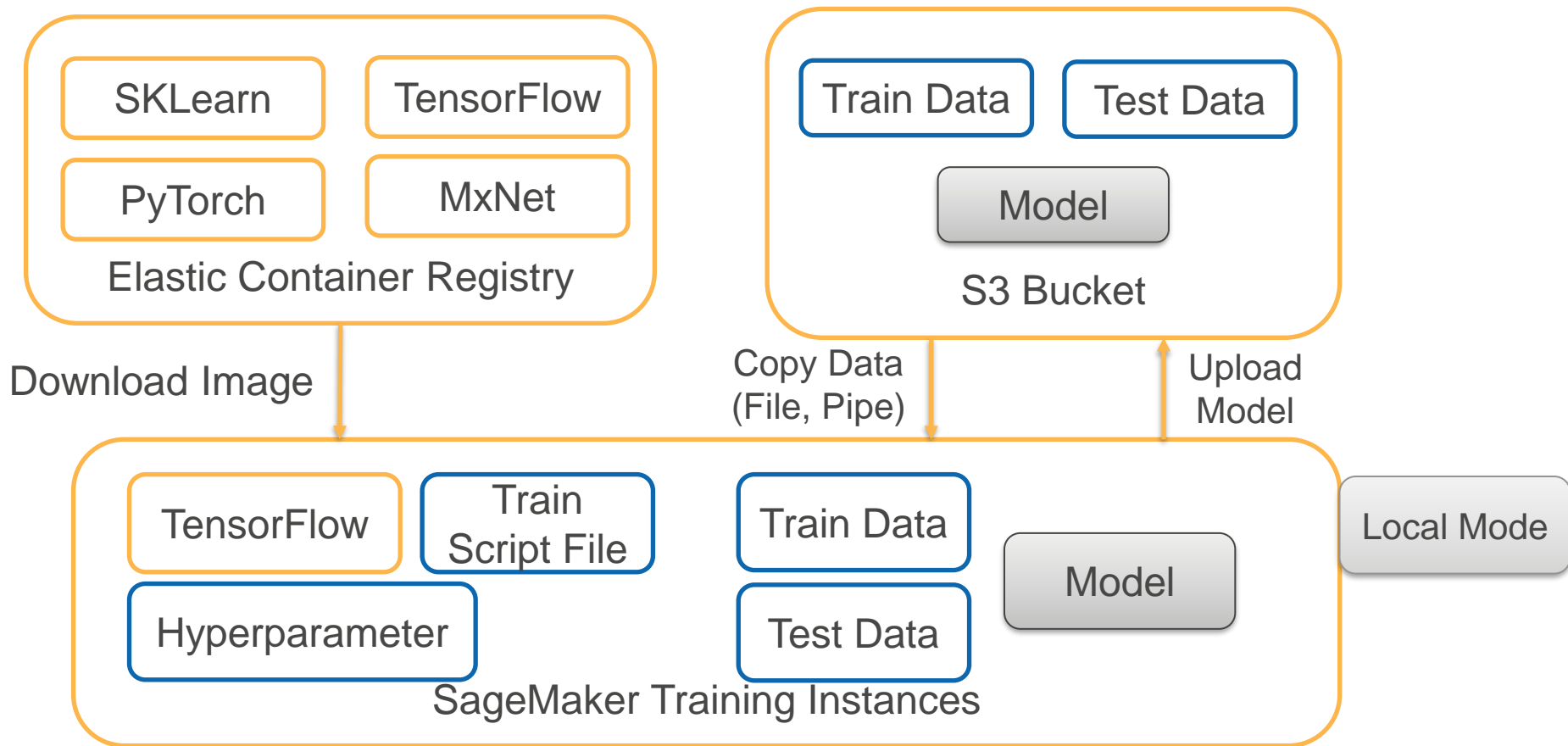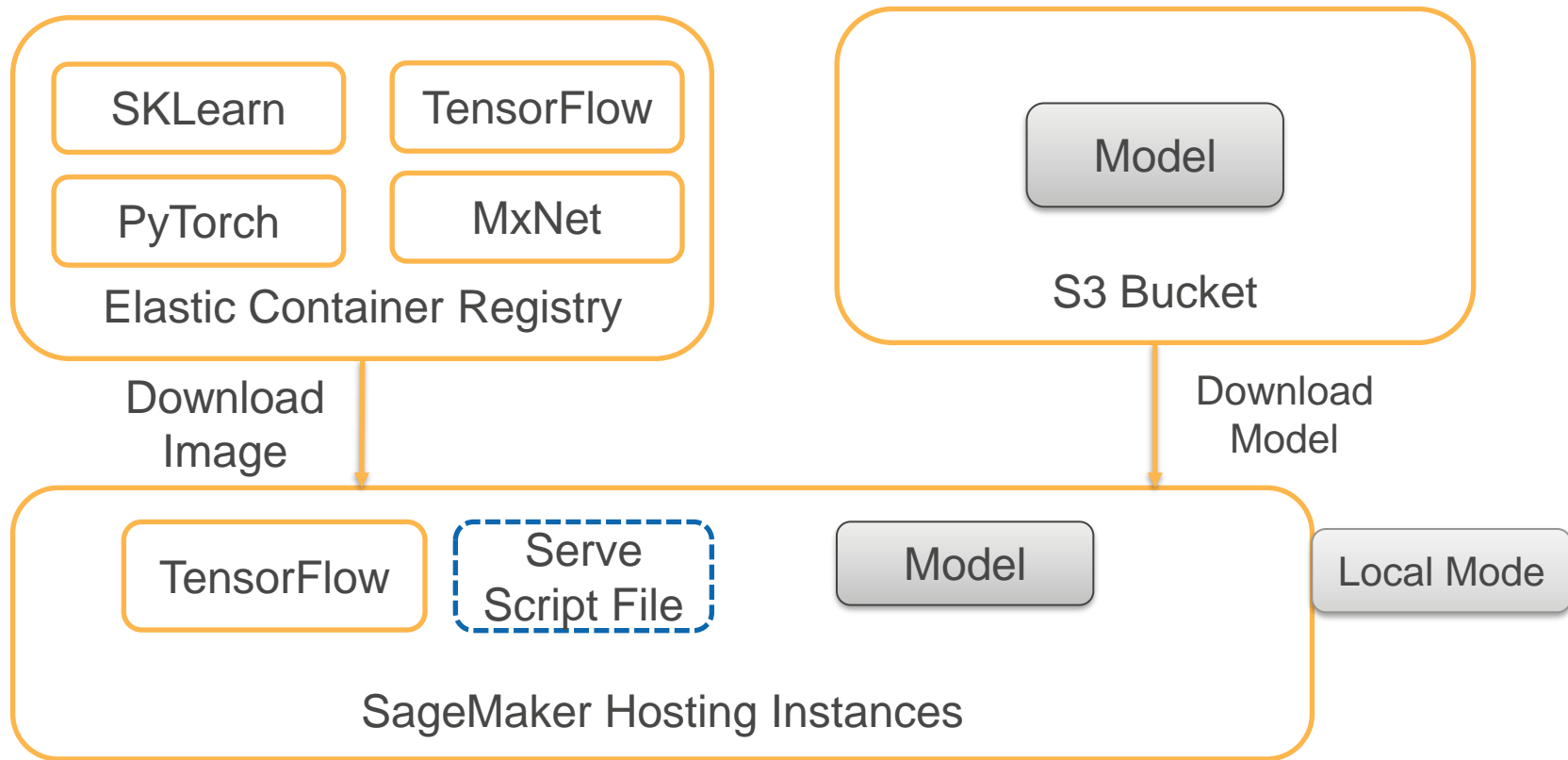
Bring Your Own Algorithm Training and Hosting

"Amazon SageMaker has certain contractual requirements that a container must satisfy to be used with it."

- Standard folder structure for reading data and resources

- Entry point that contains the code to run when container is started

- Instrumentation – Use StdOut, StdErr. SageMaker sends these message to CloudWatch log

- Metric Capture – Log metrics and define regex patterns to capture values from log

- One image for training and hosting (or) separate images (when compute resource requirements are substantially different)

Reference: https://docs.aws.amazon.com/sagemaker/latest/dg/amazon-sagemaker-containers.html

Container Folder Structure
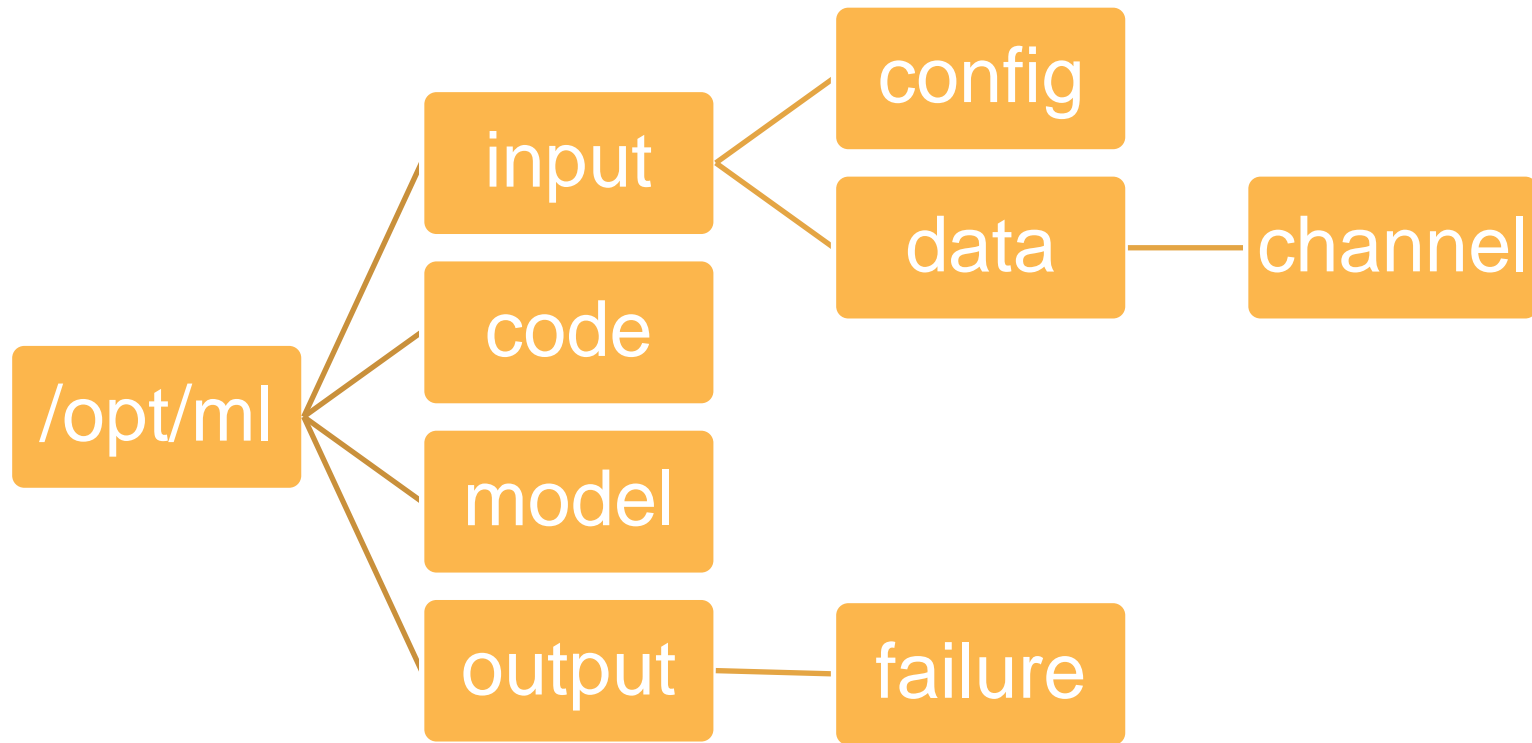
Container Folder Structure - Training

Folder	Purpose
/opt/ml/input/config/	• hyperparameters.json for training • resourceConfig.json - Container network layout for distributed training

/opt/ml/input/data/channel/	• channel = training, testing, …
	• Contains files for each channel
	    /opt/ml/input/data/training/
	    /opt/ml/input/data/testing/

/opt/ml/input/data/channel_epoch/	• Channel = training, test, eval, … • Epoch = 0,1,2,… • Read the pipe to stream data from S3 for each epoch
/opt/ml/code/	• Scripts to run from container

Container Folder Structure – Training Output

Folder	Purpose
/opt/ml/model/	• Script should write the generated model to this directory • Store your model checkpoints and final output. • SageMaker uploads the content of model folder to your S3 bucket
/opt/ml/output/failure	• If the training fails, your script should write the error description to the failure file • SageMaker returns the first 1024 characters from this file as Failure Reason in the job description • SageMaker uploads content of output folder to your S3 bucket

Container Folder Structure – Hosting

Folder	Purpose
/opt/ml/model/	• Model files to use for inference
/opt/ml/code/	• Scripts to run from container

Important Environment Variables –Your script can use

Variable	Value & Purpose
SM_MODEL_DIR	/opt/ml/model – use this to store your model checkpoints and final output. SageMaker uploads this to your S3 bucket
SM_CHANNELS	Contains the list of input data channels in the container. Example: ["training", "testing"]
SM_CHANNEL_{channel_name}	Directory containing channel data files Example: SM_CHANNEL_TRAINING='/opt/ml/input/data/training' SM_CHANNEL_TESTING='/opt/ml/input/data/testing'

Reference & Usage Examples: https://github.com/aws/sagemaker-containers#how-a-script-is-executed-inside-the-container

Important Environment Variables –Your script can use

Variable	Value & Purpose
SM_HPS	Contains a JSON encoded dictionary with the user provided hyperparameters Example: SM_HPS='{"batch-size": "256", "learning-rate": "0.0001","communicator": "pure_nccl"}'
SM_HP_{hyperparameter_name}	Contains value of the hyperparameter Example: SM_HP_LEARNING-RATE=0.0001 SM_HP_BATCH-SIZE=256 SM_HP_COMMUNICATOR=pure_nccl
NOTE: Hyperparameters are also provided as arguments to your script	

Reference & Usage Examples: https://github.com/aws/sagemaker-containers#how-a-script-is-executed-inside-the-container

Important Environment Variables –Your script can use

Variable	Value & Purpose
SM_HOSTS	JSON encoded list containing all the containers that are used for training Example: SM_HOSTS=["algo-1", "algo-2"]
SM_CURRENT_HOST	Name of the current container Example: SM_CURRENT_HOST=algo-1
SM_NUM_GPUS	The number of gpus available in the current container Example: SM_NUM_GPUS=1

Reference & Usage Examples: https://github.com/aws/sagemaker-containers#how-a-script-is-executed-inside-the-container

Lab – Bring Your Own Algorithm with SKLearnEstimator

- Develop scikit-learn model using scripts
- Train and host using SageMaker SKLearnEstimator
- Test using local mode
- Train and deploy on cloud Instance

Modified version of AWS Example: https://github.com/awslabs/amazon-sagemaker-examples/blob/master/sagemaker-python-sdk/scikit_learn_iris/Scikit-learn%20Estimator%20Example%20With%20Batch%20Transform.ipynb

Lab – Bring Your Own Algorithm TensorFlow Estimator

- Develop TensorFlow model using scripts
- Train and host using SageMaker TensorFlow Estimator
- Test using local mode
- Deploy to cloud instance

Modified version of AWS Example: https://github.com/awslabs/amazon-sagemaker-examples/blob/master/sagemaker-python-sdk/tensorflow_script_mode_training_and_serving/tensorflow_script_mode_training_and_serving.ipynb

Optional Lab – Built your own container

Most complex requires knowledge of Docker Containers, Web Stack for hosting

Walk through the code example here:

https://github.com/awslabs/amazon-sagemaker-examples/blob/master/advanced_functionality/scikit_bring_your_own/scikit_bring_your_own.ipynb