Good question — this is the **most important part** of making the pipeline work.

You must provide an **ECR container image URI** that contains:

- Python 3.10+
- `mlflow`
- `sagemaker-mlflow`
- `transformers`
- `torch`
- `boto3`

There is **no prebuilt AWS container that includes MLflow + Transformers + SageMaker MLflow plugin together**, so you must build a small custom image once and push it to ECR.

Below is the exact production-ready way to do it.

✅ Recommended Approach (Clean & Production Safe)

Step 1 — Create a Dockerfile

Create a folder locally:

```
hf-mlflow-container/
```

Inside it, create:

Dockerfile

```
FROM python:3.10-slim

# System deps
RUN apt-get update && apt-get install -y \
    build-essential \
    git \
    curl \
    && rm -rf /var/lib/apt/lists/*

# Install Python packages
RUN pip install --no-cache-dir \
```

```
    mlflow==2.13.2 \
    sagemaker-mlflow \
    transformers \
    torch \
    boto3

# SageMaker expects this working directory
WORKDIR /opt/ml/processing

ENV PYTHONUNBUFFERED=TRUE
```

Step 2 — Build the Docker Image

From inside that folder:

```
docker build -t hf-mlflow-container .
```

Step 3 — Push to Amazon ECR

1️⃣ Create an ECR repository

```
aws ecr create-repository --repository-name hf-mlflow-container
```

2️⃣ Authenticate Docker to ECR

```
aws ecr get-login-password --region <your-region> | \
docker login --username AWS --password-stdin <account-id>.dkr.ecr.
<region>.amazonaws.com
```

3️⃣ Tag the image

```
docker tag hf-mlflow-container:latest \
<account-id>.dkr.ecr.<region>.amazonaws.com/hf-mlflow-container:latest
```

4️⃣ Push it

```
docker push <account-id>.dkr.ecr.<region>.amazonaws.com/hf-mlflow-container:latest
```

✅ Final Value to Put in ScriptProcessor

Now replace:

```
image_uri="<your-mlflow-transformers-container>"
```

with:

```
image_uri="123456789012.dkr.ecr.us-east-1.amazonaws.com/hf-mlflow-container:latest"
```

Format:

```
<account-id>.dkr.ecr.<region>.amazonaws.com/hf-mlflow-container:latest
```

That's it.

🧠 Why We Cannot Use a Built-in Image

Built-in Image	Problem
HuggingFace Training container	Does NOT include mlflow
SageMaker Processing container	Does NOT include transformers
MLflow standalone image	Does NOT include sagemaker-mlflow

So we build a minimal one.

💡 Instance Type Recommendation

For your classification packaging job:

```
instance_type="ml.m5.large"
```

is fully sufficient.

No GPU required.

🚀 After This

Your pipeline will:

1. Start Processing job using this container
2. Download HF `model.tar.gz`
3. Convert to MLflow model
4. Register in MLflow Registry
5. Save `mlflow-model.tar.gz` to S3

If you'd like, I can now give you:

- A **no-Docker alternative** (using SKLearn Processor hack)
- A **CI/CD ready enterprise setup**
- Or a **fully automated Model Registry promotion workflow**

Just tell me your environment constraints.