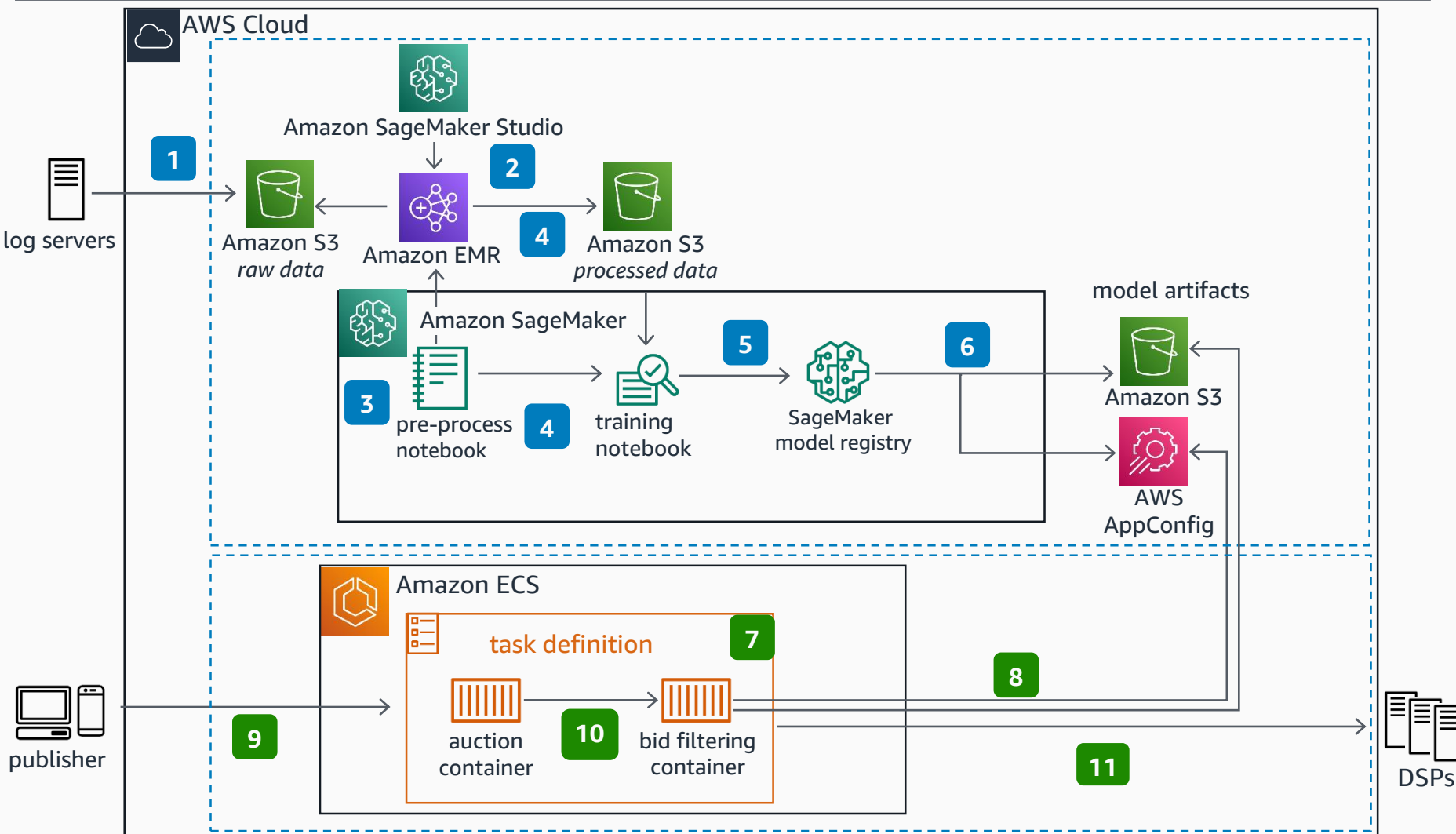


Guidance for Machine Learning for Near Real-Time Advertising on AWS

This architecture shows how AdTech users can build, train, and deploy machine learning models to an ad auction server application.



- 1 Copy raw OpenRTB log data into an **Amazon Simple Storage Service** (Amazon S3) bucket.
- 2 **Amazon SageMaker Studio** deploys an **Amazon EMR** cluster to process raw OpenRTB data.
- 3 **Amazon SageMaker Pipelines** is triggered by a user to launch the pre-processing and training steps.
- 4 In the pre-processing step, the raw data is transformed in **Amazon EMR** and stored in another **Amazon S3** bucket, with the transformation steps defined in a **SageMaker** preprocessing notebook.
- 5 A machine learning (ML) training job uses the pre-processed data, and outputs trained models to the **SageMaker** model registry.
- 6 Register the last version, save the trained model to an **Amazon S3** bucket, and register metadata in **AWS AppConfig**.
- 7 In a separate AWS environment, the ad auction server and the bid filtering applications are deployed to **Amazon Elastic Container Service** (Amazon ECS) as individual containers.
- 8 The bid filtering application pulls the latest version from the model registry and then downloads the model from **Amazon S3**.
- 9 The publisher issues auction requests to the ad auction server.
- 10 The auction server calls the bid filter. The bid filter predicts demand side platform's (DSPs) likelihood of bidding.
- 11 Bids to DSP below the threshold are filtered, eliminating the associated data transfer cost.

