Model Monitoring Pipeline and Drift Detection for ASR Microservice on AWS

Deploying the `wav2vec2-large-960h` Automatic Speech Recognition (ASR) model as a containerized microservice in the cloud introduces the need for robust monitoring to ensure reliability, detect performance degradation, and identify model drift over time. Given the project's EC2- and Docker-based architecture, a custom monitoring pipeline on Amazon Web Services (AWS) can be built using native AWS tools and open-source libraries to address these requirements without relying on SageMaker Model Monitor.

Monitoring Objectives

The monitoring pipeline must address three core concerns:

- **Operational health**: API uptime, latency, and resource usage.
- **Model performance**: Quality and reliability of transcriptions.
- **Model drift**: Detecting changes in input data or model behavior over time.

Architecture Overview

The monitoring pipeline integrates with the existing system:

- ASR microservice containerized and hosted on **AWS EC2**,
- **Elasticsearch** backend,
- **Search UI** as the end-user interface,
- Audio data stored in **Amazon S3** for auditability and retraining.

Monitoring with Cloudwatch

Each EC2 instance will be configured with the **CloudWatch Agent** to push logs and metrics, including API response times, status codes, memory and CPU usage.

CloudWatch Dashboards visualize these metrics, while **CloudWatch Alarms** notify engineers via **Amazon SNS** of anomalies (e.g., spike in failures, increased latency).

Custom Model Drift Detection

Given that wav2vec2 processes unstructured audio, drift monitoring requires custom analysis.

a. Data Drift Detection

Audio files received by the ASR API will be preprocessed using `librosa` or `pydub` to extract audio-level features such as Duration, loudness (RMS), silence ratio.

These are stored in **Amazon Timestream** or **S3** for temporal analysis. Weekly or daily jobs (triggered by **AWS Lambda**) compare current distributions with historical baselines using statistical tests to detect significant deviations.

b. Concept Drift Detection

Where ground-truth transcriptions are available, **Word Error Rate (WER)** and **Levenshtein distance** are calculated. For production data, drift is inferred using:

- Transcription entropy,
- Out-of-vocabulary token frequency,
- Change in common n-gram patterns.

These metrics are logged and visualized to flag semantic inconsistencies.

Audit Logging and Feedback Loop

All transcriptions, audio metadata, and extracted features are stored in **Amazon S3**, **Athena** or **Glue** can be used to query this data. This supports both manual review and retraining pipelines.

Model Retraining

If drift or degradation is confirmed, a retraining pipeline can be initiated manually or scheduled. Audio samples flagged for drift or poor performance are re-labeled and added to an updated dataset for finetuning wav2vec2, with retraining occurring either locally or later migrated to **SageMaker Training Jobs**.

Conclusion

A well-integrated AWS-native monitoring pipeline, using EC2, CloudWatch, S3, and open-source audio libraries, enables proactive drift detection and robust maintenance of ASR service quality. This custom approach aligns with the containerized, low-overhead architecture while ensuring scalability, auditability, and long-term performance.