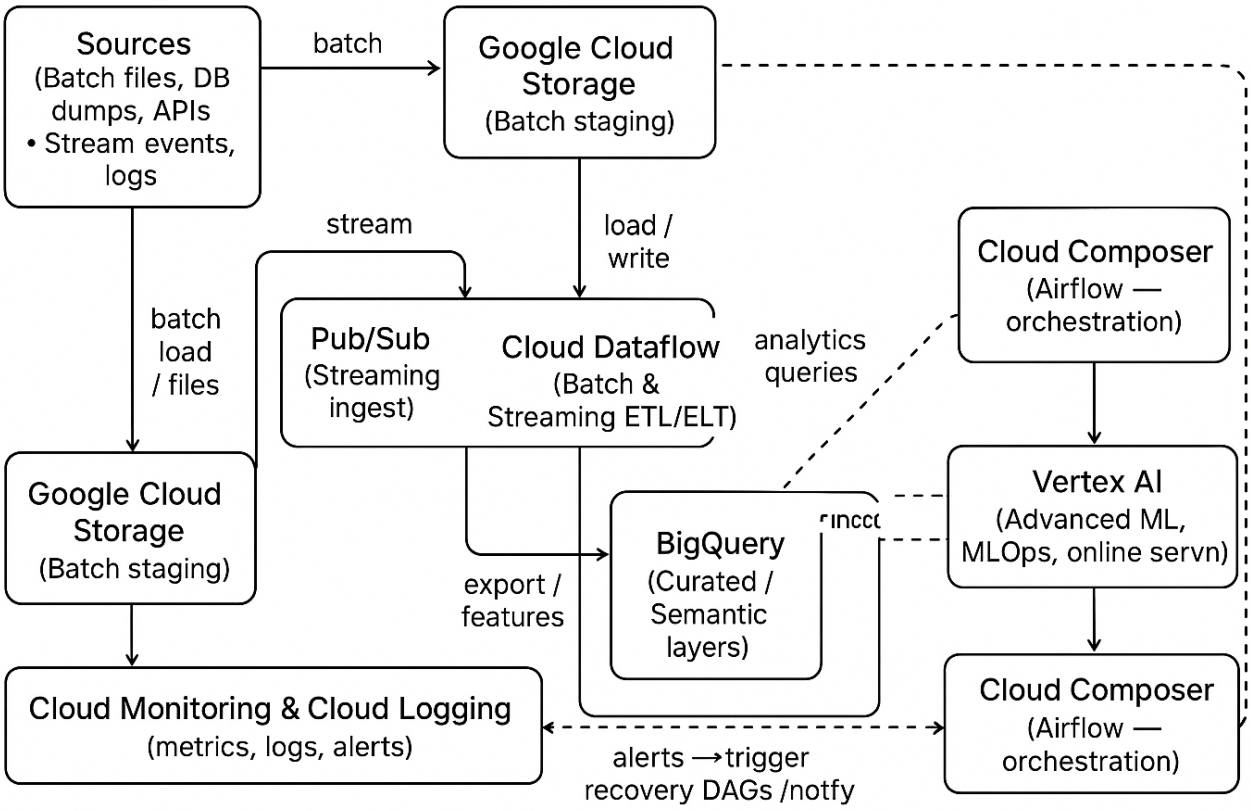
МАСШТАБИРОВАНИЕ ETL-ПАЙПЛАЙНА

Расширенное объяснение в рамках техзадания

Цель: Данный документ объясняет, как ETL-пайплайн может масштабироваться для обработки больших объёмов данных (100+ миллионов строк) с использованием облачных решений.

Рассмотрены два облачных стека: Google Cloud Platform (GCP) и Amazon Web Services (AWS).

1. Google Cloud Platform (GCP) Stack



Хранилище:

- Исходные данные: Сохраняются в бакетах Google Cloud Storage (GCS).
- Агрегированные результаты: Хранятся в BigQuery для быстрых SQL-запросов и аналитики.

Оркестрация ETL:

- Airflow (Cloud Composer): Управление DAG'ами для последовательности extract →transform → load, планирование и мониторинг ETL.
- Опционально: Prefect для более лёгкой оркестрации, если необходимо.

Обработка:

- Dataflow (Apache Beam) или Dataproc (Spark): Распределённая обработка данныхдля трансформации и агрегации.

- BigQuery: Возможность обработки больших объёмов агрегированных данных без необходимости разворачивать отдельные VM.

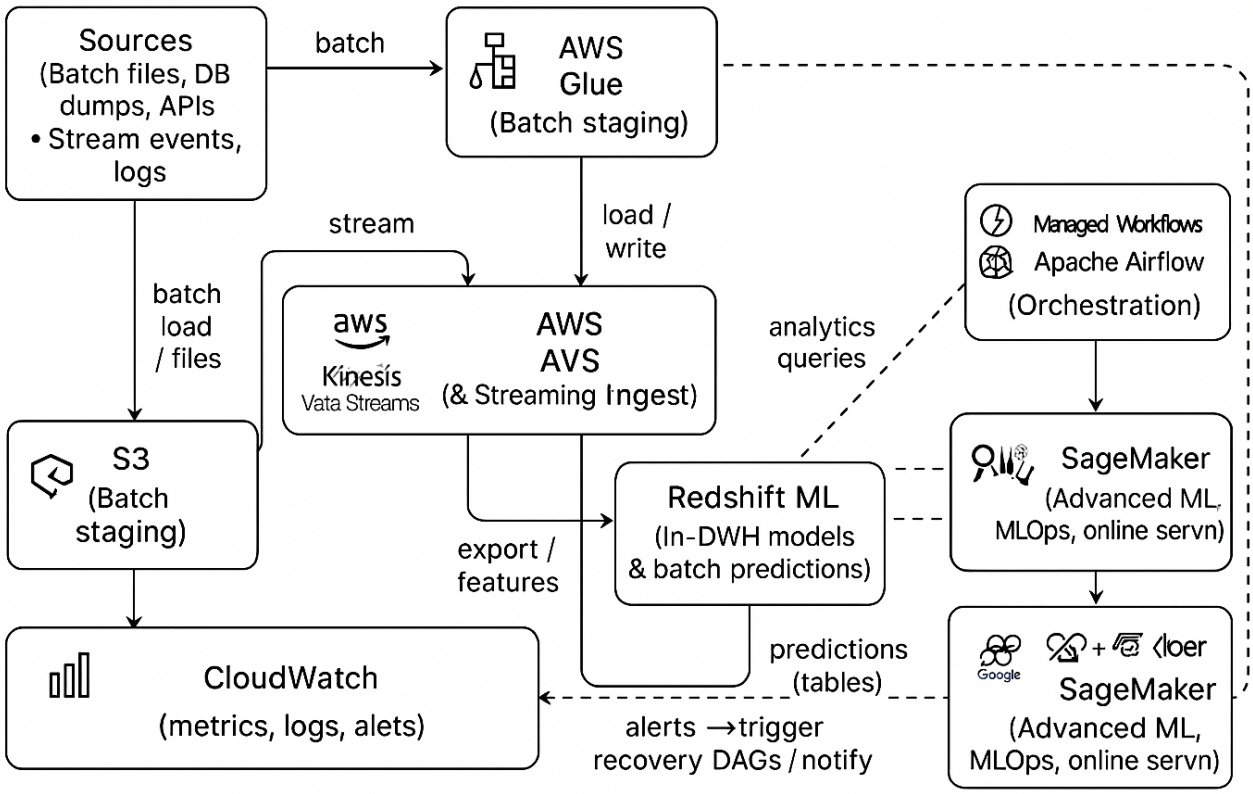
Мониторинг:

- Stackdriver / Cloud Monitoring: Отслеживание производительности ETL, ошибок и метрик качества данных.
- Кастомные дэшборды: Метрики обработанных строк, времени выполнения, пропусков и ошибок.

Контейнеризация:

- Docker + Cloud Run / GKE: Обеспечивает воспроизводимость ETL-запусков даже при резком увеличении объёма данных.

2. Amazon Web Services (AWS) Stack



Хранилище

- Исходные данные: Хранятся в бакетах S3.
- Агрегированные результаты: Хранятся в Amazon Redshift, Athena или AuroraPostgreSQL.

Оркестрация ETL

- AWS Managed Airflow (MWAA) или Step Functions: Управление ETL-воркфлоу с зависимостями, повторными попытками и расписанием.

Обработка

- AWS Glue (на базе Spark): Распределённая обработка данных для больших объёмов.

- Athena: Возможность запросов напрямую к данным в S3 для ad-hoc агрегаций.

Мониторинг

- CloudWatch Metrics & Logs: Отслеживание выполнения ETL, ошибок и производительности.
- SNS / EventBridge: Оповещения о сбоях или пропущенных данных.

Контейнеризация

- Docker + ECS / EKS: Запуск ETL-пайплайнов в любом окружении с возможностью масштабирования.

3. Основные рекомендации для обоих стеков

- **Партиционирование (partitioning) данных:** Разделение по неделе / клиенту / символу для ускорения обработки и запросов.
- **Инкрементальные загрузки:** Избегать полной переработки данных; использовать дельты для сокращения времени обработки.
- **Повторно используемые Docker-контейнеры:** Гарантируют запуск ETL в dev, staging и production без изменений кода.
- **Интерактивность:** Очищенные после EDA данные могут использоваться для интерактивных дэшбордов (Tableau / QuickSight / Looker) для аналитики в реальном времени.
- **Прогнозирование и моделирование:** основные текущие метрики идут на «запитывание» стандартизированных ML-моделей (регрессия, классификация и т. д.)

4. Мониторинговые метрики и алерты: GCP

Cloud Dataflow

- **Метрики:** job status (running/failed), job latency, throughput (rows/s), backlog, worker count, CPU/Memory per worker, error rates.
- **Алерты:** job failed, throughput < threshold, worker autoscaling failed.

Pub/Sub

- **Метрики:** message backlog, publish/ack latency, acked messages/s, subscription errors.
- **Метрики:** backlog > threshold, ack latency high.

GCS

- **Метрики:** object count, bytes ingested per time, transfer errors, lifecycle actions.
- **Метрики:** failed uploads, sudden drop/increase in bytes.

BigQuery

- **Метрики:** query latency, long-running queries, slot utilization, load job failures, storage bytes, streaming insert error rates.
- **Метрики:** load job failures, slot usage > threshold, query failures spike.

BigQuery ML

- **Метрики:** training job status, model evaluation metrics (RMSE, AUC), training time.
- **Метрики:** training failed, degradation of evaluation metrics.

Vertex AI

- **Метрики:** training/deploy job status, endpoint latency, error rate, prediction count.
- **Метрики:** endpoint latency > SLA, prediction error spikes.

Cloud Composer

- **Метрики:** DAG run success/failure rate, task retries, scheduler errors, env health.
- **Метрики:** DAG failure count > threshold, scheduler down.

Cloud Monitoring & Logging

- **Метрики:** alerting policies, notification delivery success, log ingestion lag.

5. Мониторинговые метрики и алерты: AWS

AWS Glue (ETL)

- **Метрики:** job status (running/failed), job duration, DPU usage, succeeded/failed runs, error messages.
- **Алерты:** job failed, job duration > threshold, DPU limit reached.

Amazon Kinesis / Kinesis Data Streams

- **Метрики:** incoming/outgoing records, iterator age, put/processing latency, shard count, throttling errors.
- **Алерты:** iterator age > threshold, throttling errors spike, shard count mismatch.

Amazon S3

- **Метрики:** object count, bytes ingested per time, request count, 4xx/5xx errors, lifecycle transitions.
- **Алерты:** failed uploads, sudden drop/increase in bytes, 5xx error spike.

Amazon Redshift

- **Метрики:** query latency, long-running queries, concurrency slots usage, load/unload errors, disk space utilization.
- **Алерты:** load failures, query failures spike, disk usage > threshold.

Amazon SageMaker / SageMaker Studio / SageMaker Endpoints

- **Метрики:** training job status, model metrics (RMSE, AUC), training time, endpoint latency, invocation errors, invocations per second.
- **Алерты:** training failed, model performance degradation, endpoint latency > SLA, invocation error spikes.

AWS Step Functions

- **Метрики:** execution status (succeeded/failed), execution duration, retries, throttling errors.
- **Алерты:** execution failure count > threshold, retries > threshold.

Amazon CloudWatch

- **Метрики:** custom metrics, alarm state changes, log ingestion latency, metric publishing errors.
- **Алерты:** alarm triggered, log ingestion lag, metric publishing failures.

AWS Lambda

- **Метрики:** invocation count, duration, error count, throttles, iterator age (for stream-based triggers).
- **Алерты:** errors > threshold, throttling > threshold, duration > SLA.