

DATA ENGINEERING

INTERMEDIATE

5 DAYS

Data Engineering on Google Cloud (GCP)

Build a strong foundation in Google Cloud Data Engineering, from data lake storage to batch and streaming delivery at scale. Learn how to build secure and observable pipelines using BigQuery, Composer, Pub/Sub, Dataflow, and governance practices aligned with real enterprise workloads.

Why GCP Data Engineering?

Faster analytics at scale

BigQuery enables high-performance querying without heavy infrastructure management.

Better AI integration

GCP aligns strongly with ML workflows using Vertex AI and data services.

Improved data streaming capability

Use Pub/Sub and Dataflow for real-time pipelines and event processing.

Scalable business intelligence

Enable self-service analytics with secure and centralized datasets.

Lower operational complexity

Managed services simplify cluster and pipeline maintenance.

What you'll be able to do

- ✓ Understand modern data engineering architectures on GCP and map services to batch and streaming workloads.
- ✓ Build batch and streaming data pipelines using GCP-native services.
- ✓ Design a scalable data lake on Cloud Storage (GCS) with raw/cleansed/curated zones.
- ✓ Build analytics-ready warehouse datasets using BigQuery with partitioning and clustering.
- ✓ Implement orchestration using Cloud Composer (Managed Airflow) with scheduling, monitoring, retries, and backfills.
- ✓ Build streaming pipelines using Pub/Sub and Dataflow (Apache Beam concepts) landing into BigQuery.
- ✓ Apply reliability best practices including idempotency, retries, backfills, and schema evolution handling.
- ✓ Implement data governance and security using IAM, service accounts, encryption, and access controls.
- ✓ Implement monitoring and observability practices for BigQuery, Dataflow, and Airflow pipelines.
- ✓ Deliver an end-to-end capstone pipeline that combines batch and streaming with validation and monitoring.

Who this is for

- Data Engineers
- Analytics Engineers
- Cloud Engineers supporting data platforms
- Data Platform Engineers

- DevOps engineers supporting GCP data services
- BI engineers transitioning into data engineering

Curriculum

01 GCP Data Engineering Architecture (Batch and Streaming Mapping)

- Modern GCP data platform reference architectures (lake + warehouse + streaming)
- Batch vs streaming workloads and selection criteria
- Service mapping overview (GCS, BigQuery, Pub/Sub, Dataflow, Composer)
- End-to-end lifecycle (ingest, store, transform, orchestrate, serve)
- **Hands-on:** Activity: Design a GCP reference architecture for analytics + real-time events

02 Data Lake Design on GCS (Raw, Cleansed, Curated Zones)

- GCS fundamentals for data lakes (buckets, prefixes, lifecycle basics)
- Zone-based storage layout (raw, cleansed, curated)
- Partition folder strategy (date, region, source) and naming conventions
- Columnar formats (Parquet) and small files considerations
- **Hands-on:** Lab: Create GCS lake zones and upload datasets with partitioned folder structure

03 BigQuery Fundamentals (Analytics Warehouse Foundations)

- BigQuery concepts (datasets, tables, views, costs model basics)
- Partitioning strategies (ingestion time vs column-based partitions)
- Clustering for performance improvements
- Warehouse modeling basics (facts/dimensions, analytics views)
- **Hands-on:** Lab: Create BigQuery datasets and build partitioned + clustered tables for analytics

04 Batch Pipelines on GCP (Ingestion + Transform into BigQuery)

- Batch ingestion patterns (file drops to GCS, scheduled loads)
- Loading data into BigQuery (load jobs, schema mapping concepts)
- Transformations using BigQuery SQL (CTEs, windows, incremental patterns overview)
- Publishing curated datasets and marts
- **Hands-on:** Lab: Build a batch pipeline from GCS raw → BigQuery staging → curated tables

05 Orchestration with Cloud Composer (Managed Airflow)

- Composer basics (DAGs, tasks, scheduling)
- Retries, timeouts, and failure handling patterns
- Backfills and catchup strategy for batch pipelines
- Monitoring and logs for Airflow operations
- **Hands-on:** Lab: Build a Composer DAG to orchestrate batch ingestion and BigQuery transforms with retries

06 Pub/Sub Fundamentals (Event-Driven Streaming Backbone)

- Pub/Sub concepts (topics, subscriptions, ack, delivery semantics)
- Designing event schemas and versioning strategy
- Ordering keys and throughput considerations (concept)
- Dead-letter patterns and failure isolation concepts
- **Hands-on:** Lab: Publish events to Pub/Sub and validate subscription consumption behavior

07 Streaming Pipelines with Dataflow (Apache Beam Concepts)

- Dataflow and Beam concepts (pipelines, transforms, windows concept)
- Streaming pipeline pattern (Pub/Sub → Dataflow → BigQuery)
- Handling late data and watermark awareness (concept)
- Streaming-to-warehouse design (deduplication, idempotent writes)
- **Hands-on:** Lab: Build a streaming Dataflow pipeline from Pub/Sub to partitioned BigQuery tables

08 Reliability Best Practices (Idempotency, Retries, Backfills, Schema Evolution)

- Idempotency patterns for batch loads and streaming writes
- Retries and backoff design (what to retry vs what not to retry)
- Backfill strategy for missed days or reprocessing
- Schema evolution handling (new columns, type changes, compatibility)
- **Hands-on:** Lab: Implement dedup/idempotency logic and simulate a schema change safely

09 Governance and Security (IAM, Service Accounts, Encryption, Access Controls)

- IAM fundamentals for data platforms (least privilege access)
- Service accounts and workload identity patterns (concept)
- Encryption concepts (at rest and in transit) and key management awareness
- Access controls for BigQuery datasets and GCS buckets
- **Hands-on:** Lab: Configure least-privilege access for pipelines and validate restricted permissions

10 Monitoring and Observability (BigQuery, Dataflow, Airflow)

- What to monitor (latency, errors, throughput, freshness)
- Airflow observability (DAG run status, retries, logs)
- Dataflow monitoring concepts (job health, backlogs, worker behavior)
- BigQuery monitoring concepts (slot usage, query costs, load errors)
- **Hands-on:** Lab: Create monitoring checklist + alerts for pipeline failures and data freshness

11 Capstone Project (End-to-End Batch + Streaming Pipeline)

- Capstone goal: Deliver an end-to-end GCP pipeline combining batch and streaming
- Batch: GCS zones → BigQuery staging → curated tables with validation
- Streaming: Pub/Sub → Dataflow → BigQuery with dedup/idempotency
- Orchestrate: Cloud Composer DAG with retries, monitoring, backfills
- Security + observability: IAM/service accounts + monitoring and runbook notes
- **Hands-on:** Capstone Lab: Deliver architecture diagram, pipeline code, DAG, validation results, and monitoring evidence