STS DAG Design Scenarios: New Files Only

Goal:

Transfer only new files from one GCS bucket to another using Storage Transfer Service (STS) via Composer DAG.

SCENARIO 1: Polling via Composer (No Pub/Sub or Cloud Function)

- Composer DAG runs every 15 min
- PythonOperator checks for new files
- If found, it creates a new one-time STS job

Pros:

- Works in restricted orgs
- Fully controlled logic

Cons:

- Creates new job every time
- Build-up of jobs in STS console

Use When:

- Pub/Sub not allowed
- You want simple control over timing

SCENARIO 2: Pub/Sub + Cloud Function + DAG Trigger

- GCS bucket sends event to Pub/Sub
- Cloud Function triggers DAG (via API or Pub/Sub-to-Airflow)
- DAG uses pre-created or dynamic STS job

Pros:

- Truly event-driven

- Executes only when needed

Cons:

- Needs Cloud Function and Pub/Sub setup

- More IAM complexity

Use When:

- Real-time transfer is needed

- Organization allows event infra

SCENARIO 3: Pre-Created STS Job + RunJobOperator

- STS job created once in console

- Composer DAG reuses same job via CloudDataTransferServiceRunJobOperator

Pros:

- No job build-up

- Simple DAG

Cons:

- Less flexibility per run

- Requires pre-defined filters in job

Use When:

- You want minimal DAG logic

- You can manually create/manage STS job

STS Limitations with Excess Job Creation:

- STS quotas may be exceeded (e.g., jobs/day, transfer operations)
- Console becomes cluttered with completed jobs
- No auto-cleanup for old jobs
- Small transfer jobs may lead to inefficiencies and higher cost

Best Practices per Scenario:

Polling (Scenario 1):

- Use prefix filters like YYYY/MM/DD/ to limit transferred files
- Add timestamps to job names for traceability
- Limit to one job per hour/day if not urgent
- Consider a cleanup DAG if jobs accumulate

Pub/Sub + Function (Scenario 2):

- Use a centralized notification bucket
- Use batching or buffer window before triggering DAG
- Add conditional logic in Cloud Function to reduce noise

Pre-Created Job (Scenario 3):

- Use scheduleStartDate + includePrefixes smartly to limit scope
- Monitor job success/failure in STS UI or logs
- Rotate job if prefix scheme changes

Comparison Table:

| Scenario | STS Job | Trigger | Infra | Handles New Files | Complexity |

|----------|---------|---------|-------|-------------------|------------|

| 1. Polling | New job | DAG schedule | Composer only | Yes | Medium |

| 2. Pub/Sub | Reuse/Create | File upload | Full infra | Yes | High |

| 3. Pre-created | Reuse | Schedule | Composer | Yes (with filter) | Low |