Building Airflow Setups Resilient to Zonal/Regional Down Events

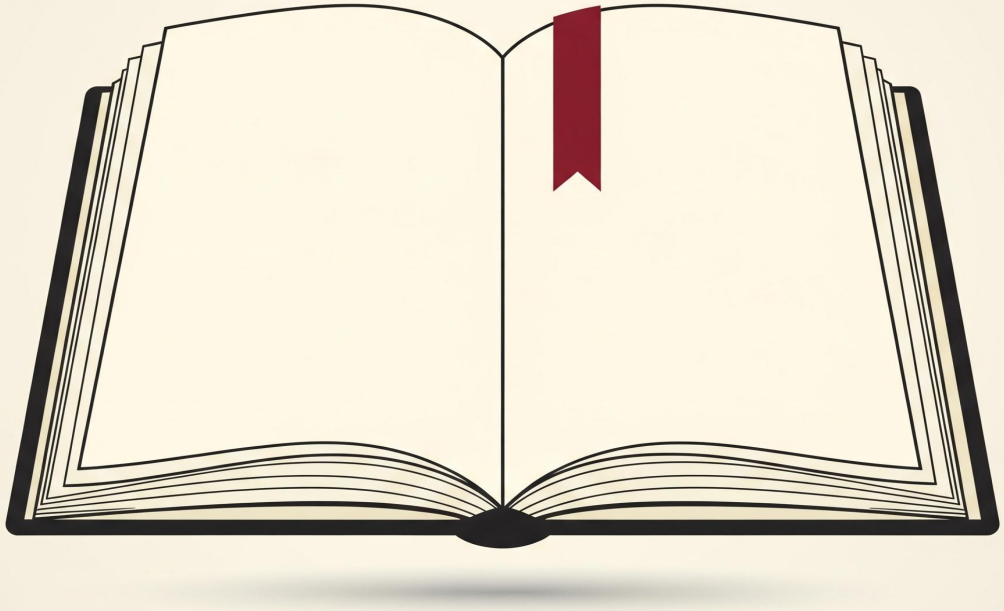
3.0

Khaled Hassan

AIRFLOW SUMMIT

Agenda

- The Resilience Problem
- Triggers of the problem
- Mitigation Strategies
- Conclusion
- QA

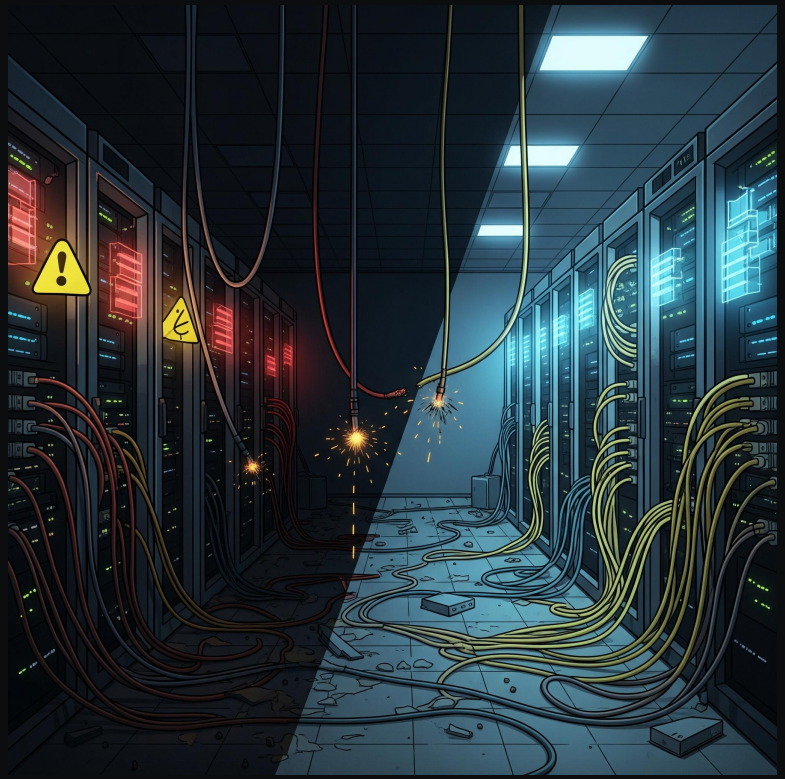


The Problem


ZONAL FAILURE DETECTED
SERVICE DISRUPTION
Benlar zone
ZONAL FAILURE DETECTED
Zonal zone
SERVICE DISRUPTION

The Problem: Vulnerability to Physical Failures

- Using Airflow as a solution for your business
- Physical failure (e.g. power/network outage)
- Running software behaviour during the outage in two aspects:
  - **Physically -** How and where are the critical infrastructure components located.
  - **Software -** How DAGs are prone to these failures



Mitigation Strategies

- **Retries should be enabled**

- **Infrastructure**
  - Deployments Replication
  - Deployments Spread
  - Data Replication

- **DAG Implementation**
  - DAGs statelessness
  - DAGs idempotency
  - DAGs dependencies replication

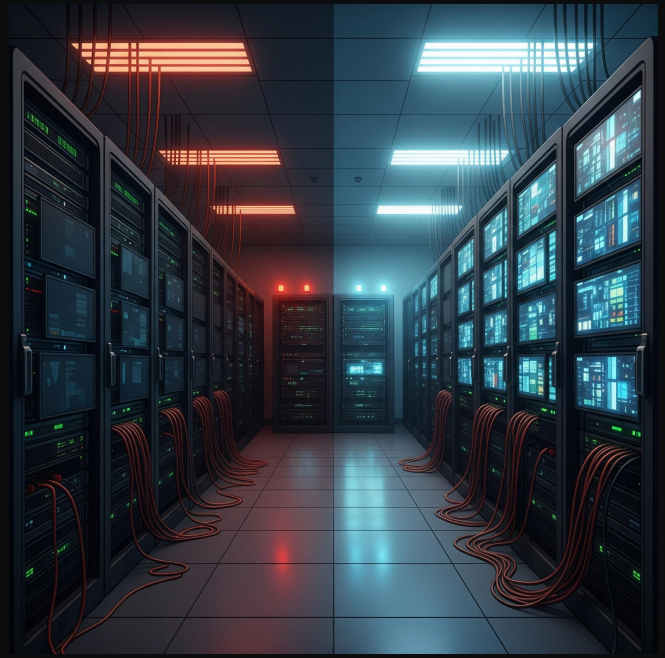
- **Complementary Strategies -** One on its own may not be sufficient.

Mitigation Strategy 1: Replication Of Critical Components

- **Deployments Replication**
  - Critical Deployments (e.g. Workers, Schedulers, Internal API in AF3)
  - Failovers to healthy replicas during failures

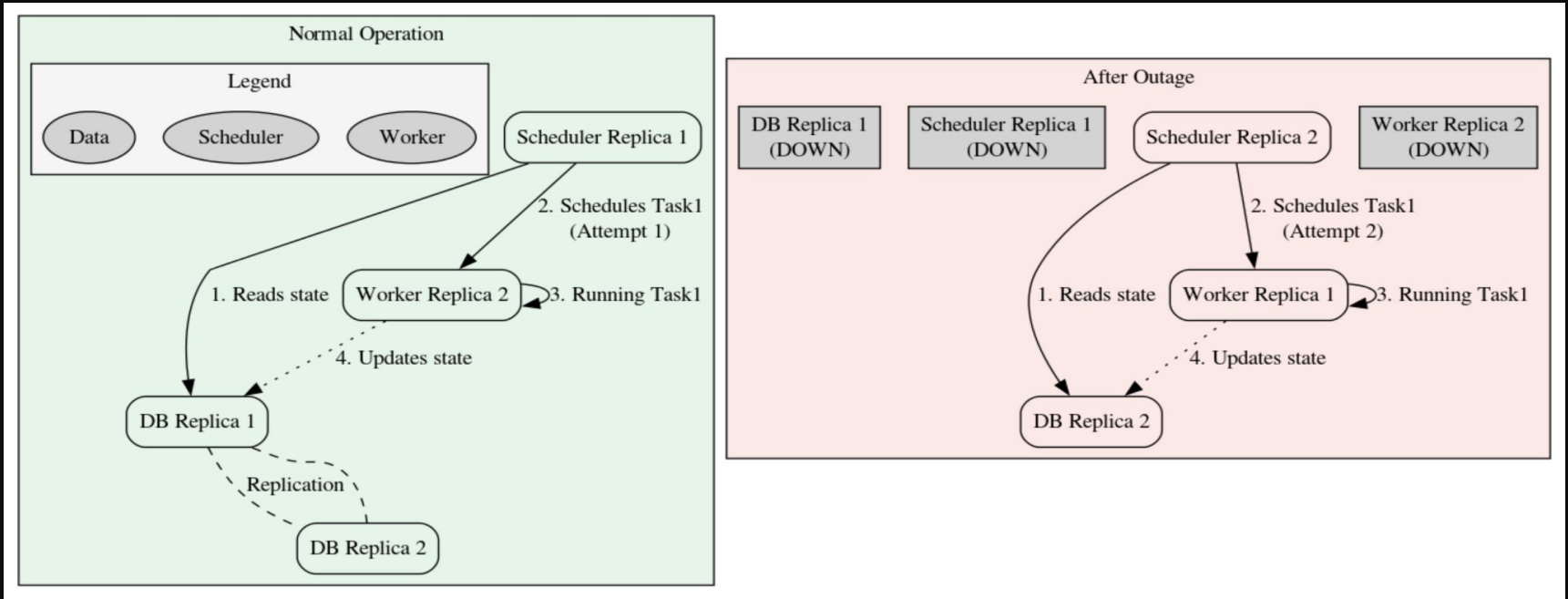
- **Data Replication -** Airflow DB
  - Ensure that healthy replicas can always access needed data
  - Replicating DBs might be more challenging than workloads

Real Life Scenario 1



- Identical server rooms (2 Failure Domains):
  - Failure Domain 1: Room A contains
    - Scheduler Replica 1
    - DB Replica 1
    - Worker Replica 2
    - AF3: Internal API Replica 2
  - Failure Domain 2: Room B contains
    - Scheduler Replica 2
    - DB Replica 2
    - Worker Replica 1
    - AF3: Internal API Replica 1
- Only Room A Loses Power
- Potential Problem: Building-wide Power Outage ?

Mitigation Strategy 1: Replication Of Critical Components



Potential Problem - What if both Rooms share the same infrastructure (e.g. power/network) ?

Mitigation Strategy 2: Spread of replicated components

- Mitigation of Zonal/Regional Outages
- Enhanced Availability
- Zonal/Regional Disaster Recovery

- Cloud solution example:
  - Components are workloads in Kubernetes
  - Defining topologySpreadConstraints in kubernetes

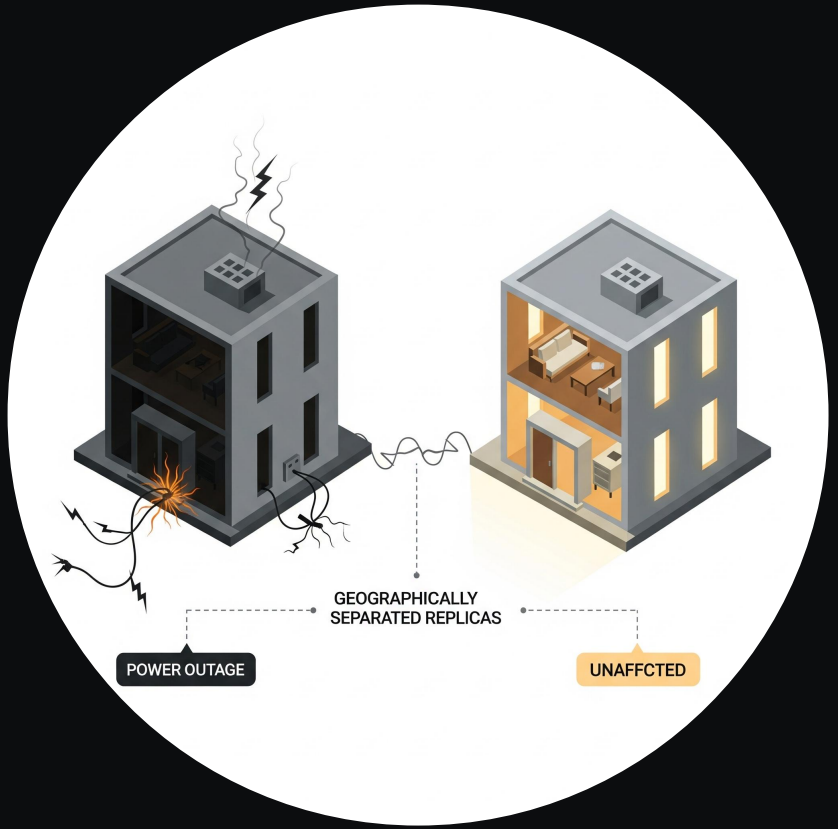
```
topologySpreadConstraints:
      - maxSkew: 1
        topologyKey: topology.kubernetes.io/zone
```

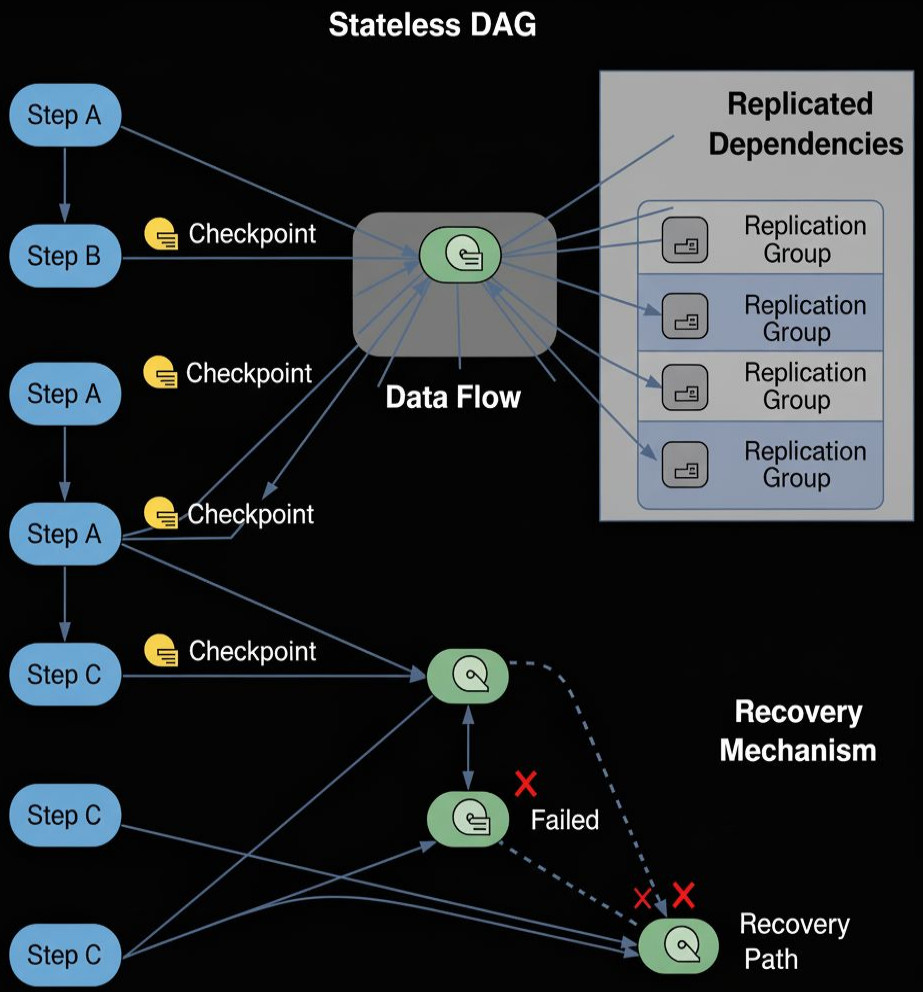
Real Life Scenario 2

- Room A is now in Building A and Room B is in Building B
- Each building now is a failure domain

- Whole Building A loses Power

- Continuity of processes hence continuity of business





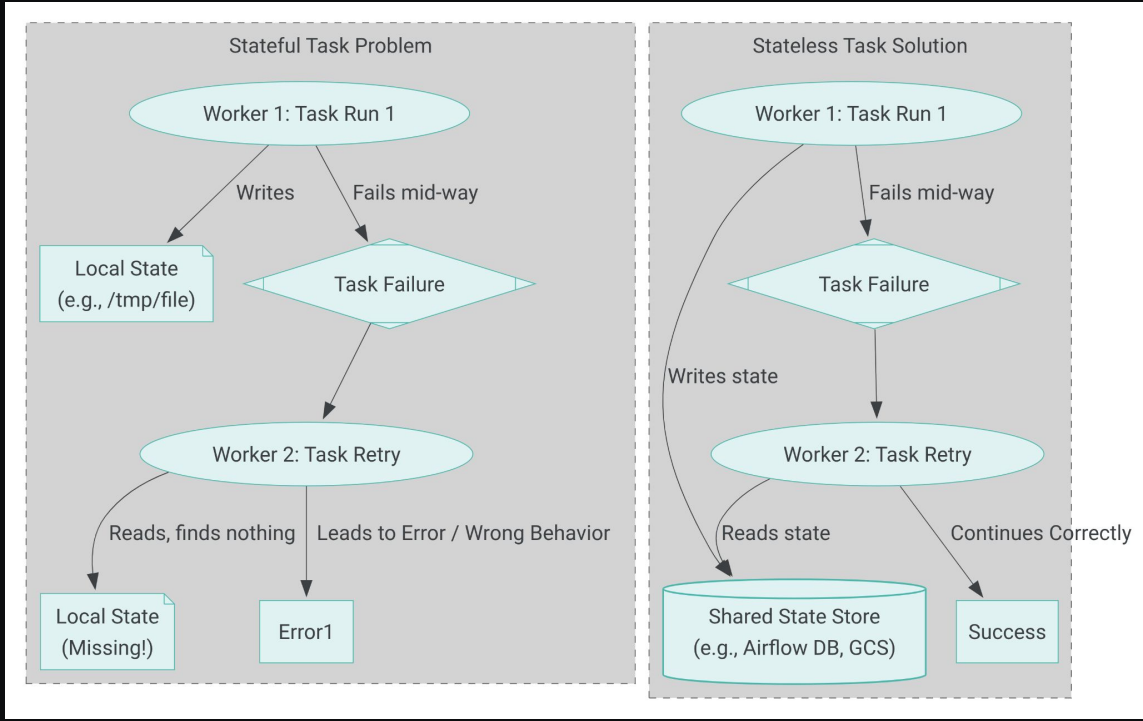
Mitigation Strategy 2: DAG Implementation

- **Stateless DAGs**
- **Idempotent Operations**
- **Replicated Dependencies**

DAG Strategy 1: Stateless DAGs

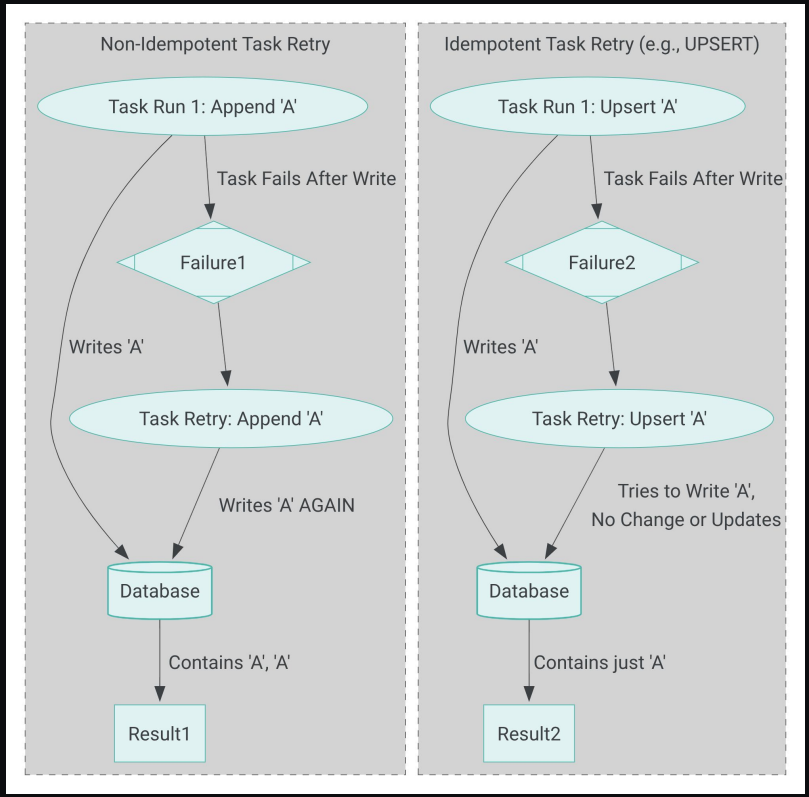
- **Stateful Definition**
- **Stateless Definition**

- **Why is Statefulness a Problem?**
  - **Lost Progress**
  - **Inconsistent State on retries**
  - **Failover Issues**



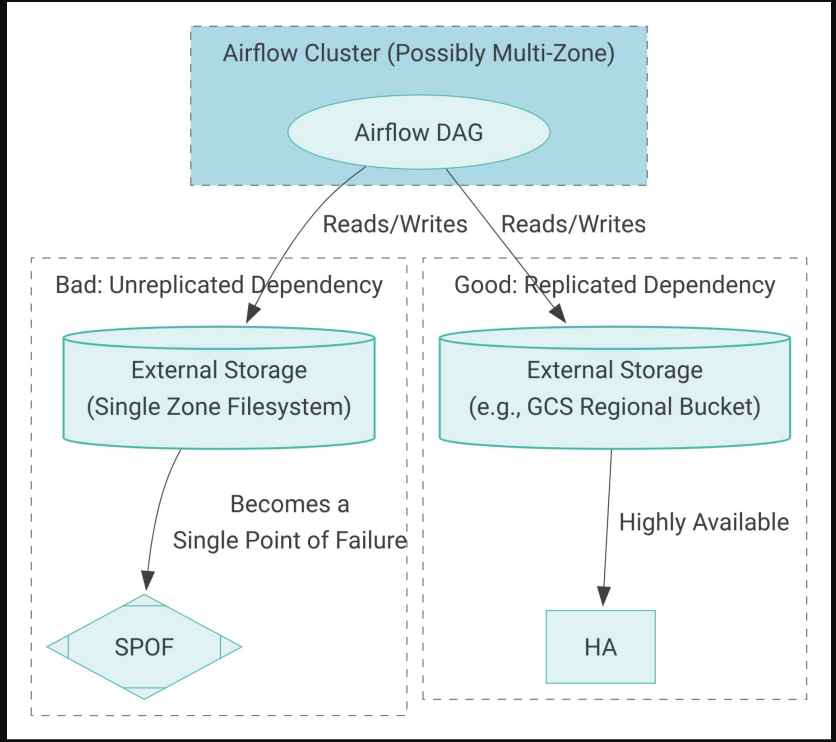
DAG Strategy 2: Idempotent Operations

- **What is Idempotency?**
- **Why Idempotency is Crucial for Reliability?**
  - **Preventing Duplication**
  - **Simplified Recovery after failures**



DAG Strategy 3: Replicating DAG Dependencies

- **External State & Single Points of Failure**
  - **DAGs often depend on external systems (storage, databases, services).**
  - **Replication of dependencies is crucial.**



Cloud Composer

- **Managed Airflow Environment**
- **High Resilience Configuration**
  - Minimum Number of Replicas
- **Underlying Infrastructure Handled automatically.**

Conclusions

- If availability is crucial for your business:
  - Replication and Spread is crucial for critical components
  - DAGs implementation analysis is crucial (Idempotency, Statelessness, etc)
- Resilience comes with a trade off :
  - Cost
  - Complexity
  - Sometimes latency
- Larger scale failures are sometimes inevitable

Questions?

hkhaled@google.com