**Disaster Recovery in Confluent Cloud Across AWS, GCP, and Azure**

**Disaster Recovery in Confluent Cloud Across AWS, GCP, and Azure**

Confluent Cloud provides **disaster recovery (DR) solutions** to ensure data availability, business continuity, and fault tolerance across multiple cloud providers. Here are the key methods for **AWS, GCP, and Azure**:

**1. Multi-Region and Multi-Cloud Cluster Linking**

* **Cluster Linking** allows real-time replication between Kafka clusters across different regions or cloud providers.
* Supports **AWS, GCP, and Azure** for cross-cloud disaster recovery.
* Ensures **low-latency replication** and automatic failover capability.

**Steps:**

1. Create a **Cluster Link** from the source to the target Confluent Cloud cluster.
2. Enable **replication** for critical topics.
3. Configure failover **producers and consumers** to switch to the backup cluster.

**Example CLI Command:**

confluent cluster link create --source <source-cluster> --destination <destination-cluster>

**2. Backup & Restore with Object Storage**

* Backup Kafka data to **Amazon S3 (AWS), Google Cloud Storage (GCS), or Azure Blob Storage**.
* **Confluent Tiered Storage** helps offload older messages for long-term retention.
* Enables data recovery if the primary cluster fails.

**Steps:**

1. Configure **Tiered Storage** to store Kafka topics in object storage.
2. In case of failure, restore messages to a new Confluent Cloud cluster.

**3. Disaster Recovery with Confluent Replicator**

* **Confluent Replicator** syncs Kafka topics between on-prem and cloud or across different cloud providers.
* Can be used for **real-time replication** and failover.

**Steps:**

1. Deploy Confluent Replicator on AWS, GCP, or Azure.
2. Configure replication from the primary to the DR cluster.
3. Set up **consumer failover** to switch to the backup cluster if the primary fails.

**4. Cross-Cloud Disaster Recovery Using Kafka MirrorMaker 2**

* **MirrorMaker 2 (MM2)** replicates Kafka topics between different Confluent Cloud clusters.
* Supports **AWS, GCP, and Azure** for multi-cloud resilience.
* Useful for **high-availability Kafka deployments**.

**Steps:**

1. Deploy MM2 in a separate cloud provider (AWS, GCP, Azure).
2. Configure **source and target clusters** for replication.
3. Ensure **consumers can switch** to the backup cluster automatically.