To design a production and disaster recovery (DR) Azure architecture diagram with Visio/draw.io for your Confluent Kafka and Azure integration, here's a structured approach:

Production Architecture (East US)

**Core Components:**

* **Confluent Kafka Cloud** (Vendor-managed)
  + Connected via Azure Private Link service[3](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview)
  + Azure-side private endpoints in prd-kafka-pe-subnet[3](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview)
* **Azure Compute:**
  + Function Apps in prd-func-subnet
  + AKS cluster in prd-aks-subnet
* **Data Layer:**
  + Cosmos DB with multi-region writes[3](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview)
* **Networking:**

text

| Component | Configuration |

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

| DMZ | Azure Firewall Premium + NSGs [4] |

| Private Link Connection | East US VNet ↔ Confluent Cloud [3] |

| Subnets | Segregated tiers with NSG flow logs |

Production Architecture *(Conceptual flow: Confluent Kafka → Private Link → Azure PE → Functions/AKS → Cosmos DB)*

Disaster Recovery Architecture (South Central US)

**Replication Strategy:**

1. **Data Replication:**
   * Cosmos DB geo-redundancy with automatic failover[3](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview)
   * ACR geo-replication for container images
2. **Network Mirroring:**
   * Secondary VNet with peered connection to production
   * Warm standby private endpoints in dr-kafka-pe-subnet
3. **DR Components:**
   * Scaled-down AKS cluster in dr-aks-subnet
   * Azure Functions in standby mode

**Cross-Region Connectivity:**

* Virtual Network peering between regions
* Azure Traffic Manager for DNS failover
* Private Link maintained through global VNet peering[3](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview)

Critical Networking Elements

**Security Implementation:**

text

| Layer | Controls |

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

| Perimeter (DMZ) | Azure Firewall + WAF policies [4] |

| Private Connectivity | Private Endpoints (10+ services) [3] |

| Internal Segmentation | NSGs with zero-trust rules |

**Connection Flow:**

1. Confluent Kafka → Private Link Service
2. Azure Private Endpoint (East US)
3. Through Hub-Spoke architecture[4](https://learn.microsoft.com/en-us/answers/questions/1441423/how-to-create-a-virtual-network-in-dmz-in-azure)
4. To Function Apps/AKS via internal load balancers

Diagram Implementation Guide

**Visio Steps:**

1. Use *Azure Networking* stencil for Private Link/Endpoint shapes[5](https://support.microsoft.com/en-us/office/create-azure-diagrams-in-visio-efbb25e7-c80e-42e1-b1ad-7ef630ff01b7)
2. Apply *Cloud Infrastructure* template for DR layout[5](https://support.microsoft.com/en-us/office/create-azure-diagrams-in-visio-efbb25e7-c80e-42e1-b1ad-7ef630ff01b7)
3. Add annotation layers for failover paths

**draw.io Method:**

text

1. Insert Azure Architecture Library [1][2]

2. Use "Disaster Recovery" template (Cloud section)

3. Add custom Private Link connection lines

4. Export as .vsdx for team collaboration [1]

**Key Symbols to Include:**

* Azure Firewall (Dual deployment in DMZ)
* Private Link/Endpoint pairs (Color-coded per environment)
* Geo-redundancy icons for Cosmos DB
* Traffic flow arrows with failover notation

DR Activation Process

1. **Manual Failover Steps:**
   * Redirect Private Link to DR endpoints[3](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview)
   * Scale up DR AKS cluster
   * Update Traffic Manager priority
2. **Automated Checks:**
   * Private Link connection status monitoring[3](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview)
   * Cosmos DB failover validation
   * Network security group audit

This architecture provides RPO < 5 mins and RTO < 1 hour through:

* Active-passive regional deployment
* Geo-zone-redundant storage
* Pre-configured network paths
* Automated health probes for critical components

**How to Make it Active-Active**

To implement an Active-Active deployment for your Confluent Kafka and Azure infrastructure across East US and South Central US, follow this architecture:

Core Architecture Principles

1. Confluent Kafka Active-Active Setup

* Deploy two Confluent Cloud clusters (East US and South Central US) with Cluster Linking for bidirectional replication[2](https://docs.confluent.io/cloud/current/multi-cloud/cluster-linking/dr-failover.html)4.
* Configure writable topics with identical names in both regions (e.g., events) to simplify client logic[2](https://docs.confluent.io/cloud/current/multi-cloud/cluster-linking/dr-failover.html).
* Use offset synchronization via Cluster Linking to maintain consumer progress across regions[6](https://stackoverflow.com/questions/75075271/kafka-3-3-1-active-active-consumers-and-producers).

2. Azure Component Configuration

* Function Apps/AKS:
  + Deploy instances in both regions, subscribed to their local Kafka cluster.
  + Enable idempotent processing to handle duplicate events from cross-region replication[6](https://stackoverflow.com/questions/75075271/kafka-3-3-1-active-active-consumers-and-producers).
* Cosmos DB:
  + Use multi-region writes for active-active data synchronization.

3. Networking

text

| Component | Configuration |

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

| Private Link | Two-way Private Endpoints (East US + South Central US) [3] |

| VNet Peering | Connect East US and South Central US VNets |

| Azure Firewall | Stateful inspection for cross-region traffic [4] |

| NSGs | Zero-trust rules for inter-subnet communication |

Implementation Steps

1. Cluster Linking Setup

bash

*# Create bidirectional cluster links*

confluent kafka link create east-to-dr --source-cluster <east-cluster-id> --destination-cluster <dr-cluster-id>

confluent kafka link create dr-to-east --source-cluster <dr-cluster-id> --destination-cluster <east-cluster-id>

Configure topic replication policies to mirror all critical topics[2](https://docs.confluent.io/cloud/current/multi-cloud/cluster-linking/dr-failover.html)4.

2. Producer/Consumer Logic

* Producers: Write to the local Kafka cluster’s topic (e.g., events). Cluster Linking replicates data to the paired region automatically.
* Consumers:
  + Read from both the local topic and mirrored topic (e.g., events-dr in DR region)[6](https://stackoverflow.com/questions/75075271/kafka-3-3-1-active-active-consumers-and-producers).
  + Use consumer group synchronization to avoid duplicate processing[8](https://stackoverflow.com/questions/47466514/kafka-multi-node-set-up-active-active).

3. Disaster Recovery Activation

* Automatic Failover: Azure Traffic Manager routes traffic to the healthy region.
* Manual Override: Update Private Link endpoints if regional outages affect Cluster Linking.

Key Considerations

* Data Consistency:
  + Cosmos DB’s multi-region writes ensure low-latency synchronization.
  + Kafka exactly-once semantics apply *per cluster* but not cross-region[6](https://stackoverflow.com/questions/75075271/kafka-3-3-1-active-active-consumers-and-producers).
* Performance:
  + Latency between East US and South Central US averages <20ms, suitable for active-active workloads.
* Security:
  + Encrypt inter-region traffic with Azure Private Link + TLS 1.3[3](https://www.kai-waehner.de/blog/2024/07/29/apache-kafka-cluster-type-deployment-strategies/).
  + Audit replication paths via Azure Network Watcher.

Validation Checklist

1. Test cross-region failover by disabling one cluster’s Private Link.
2. Verify offset continuity using kafka-consumer-groups CLI after failover[6](https://stackoverflow.com/questions/75075271/kafka-3-3-1-active-active-consumers-and-producers).
3. Monitor replication lag with Confluent Cloud Metrics API4.

This design achieves RTO < 5 minutes and RPO ≈ 0 for Kafka workloads, with Azure components maintaining active-active readiness.

