# **Vertex AI Predication**

### **1. Cross-region Model Replication**

Vertex AI models are bound to a single region.

To prepare for disasters, regularly export your trained models to a **GCS bucket configured with multi-region or dual-region replication**. Then, **import the model into your DR region** using gcloud, Terraform, or Python SDK. This ensures the model is available in case the primary region goes down.

### **2. Endpoint & Deployment Recovery**

Endpoints that serve models for prediction are also regional.

**define endpoints in the DR region ahead of time**, and **deploy the replicated model** there. Use **Terraform** to script this so you can quickly spin up endpoints without manual effort during recovery.

### **3. Backup Frequency & Synchronization**

Automate regular backups of model versions using **Cloud Scheduler + Cloud Functions or Cloud Build**. Trigger a model export after each training job or at scheduled intervals (e.g., daily). These backups are saved in GCS and kept in sync with the DR region to avoid stale deployments during failover.

### **4. Infrastructure as Code (Terraform)**

Manage all your model and endpoint infrastructure using **Terraform modules**, parameterizing the region to support both primary and DR deployments. Store Terraform state securely (e.g., Terraform Cloud or GCS backend) and ensure changes in primary are reproducible in DR.

### **5. Failover & Traffic Switching**

Failover requires your application to be aware of both regions. Implement **a traffic-switching mechanism**, like a config-driven endpoint selection (from Firestore or Secret Manager), or use a proxy/load balancer to route requests. You must plan how to **update clients to use the DR endpoint** with minimal downtime.