Text-to-Video API Challenge

Objective

Build a text-to-video API (over HTTP) that accepts a text prompt and generates a video using the open-source Genmo Mochi-1 Preview model.

System Environment

The solution is expected to run on:
- 8 × H100 GPUs
- ~18TB cumulative NVMe storage (not mounted by default)
- 2 CPUs (~124 cores)
- Kubernetes (K8s) cluster with a worker node & a control plane (accessible via kubectl only)

Requirements

1. Asynchronous Job Management
- Submit Job API: Accept a new video generation request & return a job ID immediately.
- Get Job Status API: Query job status (pending, processing, completed, failed).
- List Jobs API: Retrieve all submitted jobs, with filtering & pagination support.
- Get Output File API: Download the generated video once a job completes.

2. Video Generation
- Use Genmo Mochi-1 Preview exclusively for text-to-video generation.
- Ensure support for concurrent video processing across all available GPUs.

3. Scalability & Concurrency
- The system should process multiple requests concurrently.
- Workloads must be distributed across GPUs for optimal throughput.

4. User Interface
- Provide a basic frontend that allows:
- Submitting text prompts
- Tracking job status
- Downloading completed videos

5. Deployment
- Deploy the entire service on Kubernetes.
- Ensure high availability & efficient GPU utilization:
- Minimum 2 replicas of the video generation service for redundancy.
- At least 2 GPUs per replica allocated for processing.

Constraints

- No reliance on employer resources.
- Use of public/open-source tools & references is allowed.

Assumptions (allowed)

- Video length
- Resolution/quality
- API schema (must be JSON in/out & sensibly structured)

Expected Deliverables

- A working demo of the service (MVP first, then expanded features).
- Documentation of:
- Planning & design decisions
- Debugging & troubleshooting
- Options considered & tools used