

# The Trip Planner.

## Cloud Application Development

Team name: Kenting 

Presenter: Po-Chun Lu

Professor: Dr. Markus Eilsperger

# Architecture Overview

# Architecture Overview

- **Frontend:**  
Built with Vue 3 for a responsive UI.
- **Backend:**  
Split into Core API Service (Itinerary & AI) and Ad Service (Ads).
- **Database:**  
Combines MySQL (Core data) and Firestore (AI & Real-time data).
- **Infrastructure:**  
Fully deployed on GKE.

# Overview

Evaluation Item	Description
Wow Factors Count	Dynamic Feed + AI Recommendation + Ad Service
Microservice Architecture	Frontend + Backend + Ad-Service + MySQL
Separate Microservice Wow	Ad-Service deployed independently on Port 3002
Kubernetes Deployment	Complete k8s on Google Cloud Kubernetes Engine
IaC Automation	Terraform + kubectl apply
Load Testing Scripts	locust/locustfile.py

# SaaS B2B Plan Architecture

Item	Implementation
Free Plan	<code>membership_tier = 'Free' (default)</code>
Standard Plan	<code>membership_tier = 'Premium'</code>
Tier Badge UI	ProfileCard.vue displays tier badge
Admin User Upgrade	AdminDashboard.vue
Premium Exclusive Features	Private trips, hide ads, Gemini 3 AI

# Itinerary Management

Based on Milestone 1, added functions

Item	Implementation
Create Itinerary	POST /api/itineraries
Edit Itinerary	PUT /api/itineraries/:id
Delete Itinerary	DELETE /api/itineraries/:id
View Itinerary Details	GET /api/itineraries/detail/:id
Itinerary List	GET /api/itineraries/by-email/:email
Destination	destination field
Date Range	start_date, end_date
Short Description	short_description (80 char limit)
Detail Description	detail_description
Private Itinerary	is_private field (Premium only)
Clone Itinerary	ItineraryManager.vue → cloneTrip()

# Social Interaction

**Based on Milestone 1, added functions.  
Deployed on Firebase Firestore**

Item	Implementation
Like Feature	POST /api/itineraries/:id/like/toggle
Like Count	GET /api/itineraries/:id/like/count
Like List	GET /api/itineraries/:id/like/list
Comment Feature	POST /api/itineraries/:id/comments
Comment List	GET /api/itineraries/:id/comments
Delete Own Comment	DELETE /api/itineraries/:id/comments/:commentId
View Other User Profiles	?profile=email@example.com URL parameter
Guest Mode Browsing	View public content & feed without login
Share Link	ItineraryManager.vue → shareTrip()

# Wow Factors

## Overview

Wow Factor	Microservice	Category
✓ Personalized Live Feed	Frontend	Social Interaction
✓ AI Recommendation Engine	Backend (Gemini)	Social Interaction
✓ Destination Ad Placements	Ad-Service	Destination Management
✓ Special Offers/Discounts	Ad-Service	Destination Management
✓ External Link Marketing	Ad-Service	Destination Management

# Wow Factors

## Destination Management

Item	Implementation
★ Buy Ad Placements	POST /api/ads
★ Special Offers Management	discount_code field
★ Attraction Discount Marketing	external_url external link redirect
Ad CRUD	Complete Create/Read/Update/Delete
Filter Ads by Destination	GET /api/ads?destination=Kyoto
Ad Images	image_url field
Merchant Dashboard UI	MerchantDashboard.vue

# Wow Factors

## Destination Management

Item	Implementation
Ad Carousel Display	AdBanner.vue in itinerary details & feed
Default Seed Data	Sample ads (Kyoto, Tokyo, Paris, Kenting, etc.)
External Link Redirect	target="_blank" opens in new window

## Travel Information

Item	Implementation
Weather Information Processing	AI suggestions include seasonal/weather tips

# Async Workflows

## AI Recommendation Engine

Item	Implementation
Soft Timeout Mechanism	server.js:500 - 4 sec foreground timeout
Background Fallback	server.js:525-546 - Background processing after timeout
Firestore State Storage	aiSuggestions/{id} collection
Frontend Polling	ItineraryManager.vue:47-62 - Poll every 2 sec
Max Tries Control	Max 8 polling attempts (16 sec)
User Toggle Control	enable_ai: false to disable AI

# Async Workflows

## Social Features

Item	Implementation
Optimistic UI Update	Immediate UI update for likes/comments
Parallel Loading	<code>loadLikesForVisibleTrips()</code> parallel load
Background Cleanup	Background Firestore cleanup on delete

## Ad Service

Item	Implementation
Fallback Strategy	Random ads when no destination match
Watch Reactive	Auto-reload on destination change
Seed Data Init	Async init on service startup

# Technical Requirements

## Microservice Architecture & 12-Factor

Item	Description
Microservice Architecture	4 services: Frontend, Backend, Ad-Service, MySQL
Codebase	Git version control
Dependencies	Explicitly declared in package.json
Config	Environment variables (.env, K8s Secrets)
Backing Services	MySQL, Firebase, GCS, Gemini
Build/Release/Run Separation	Docker build → K8s apply → rollout
Stateless Processes	Pods are stateless, data in MySQL/Firestore
Port Binding	3000, 3002, 80
Concurrency	K8s replicas adjustable

# Technical Requirements

## Microservice Architecture & 12-Factor

Item	Implementation
Fast Startup/Graceful Shutdown	Container starts in seconds
Dev/Prod Parity	Docker ensures environment consistency
Logs	JSON structured logs to stdout
Admin Processes	AdminDashboard management interface

# Technical Requirements

## Kubernetes Deployment

Item	Description
Core Components on K8s	All 4 services deployed to K8s
Local K8s (OrbStack)	k8s/ directory
GKE Deployment	k8s-gke/ directory
PersistentVolume	MySQL data persistence
ConfigMap	mysql-init-scripts
Secrets	backend-secrets (API Keys)
LoadBalancer Service	Frontend external service
HPA Autoscaling	YES

# Technical Requirements

## IaC Automation

Item	Description
Deployment Automation	kubectl apply -f k8s/
Dockerfile	All three services have Dockerfile
Deployment Scripts	local_deploy.sh, local_deploy.ps1
Terraform GKE	terraform/main.tf (GKE Cluster + Node Pool)
Redeploy Workflow	.agent/workflows/redeploy_k8s.md

# Technical Requirements

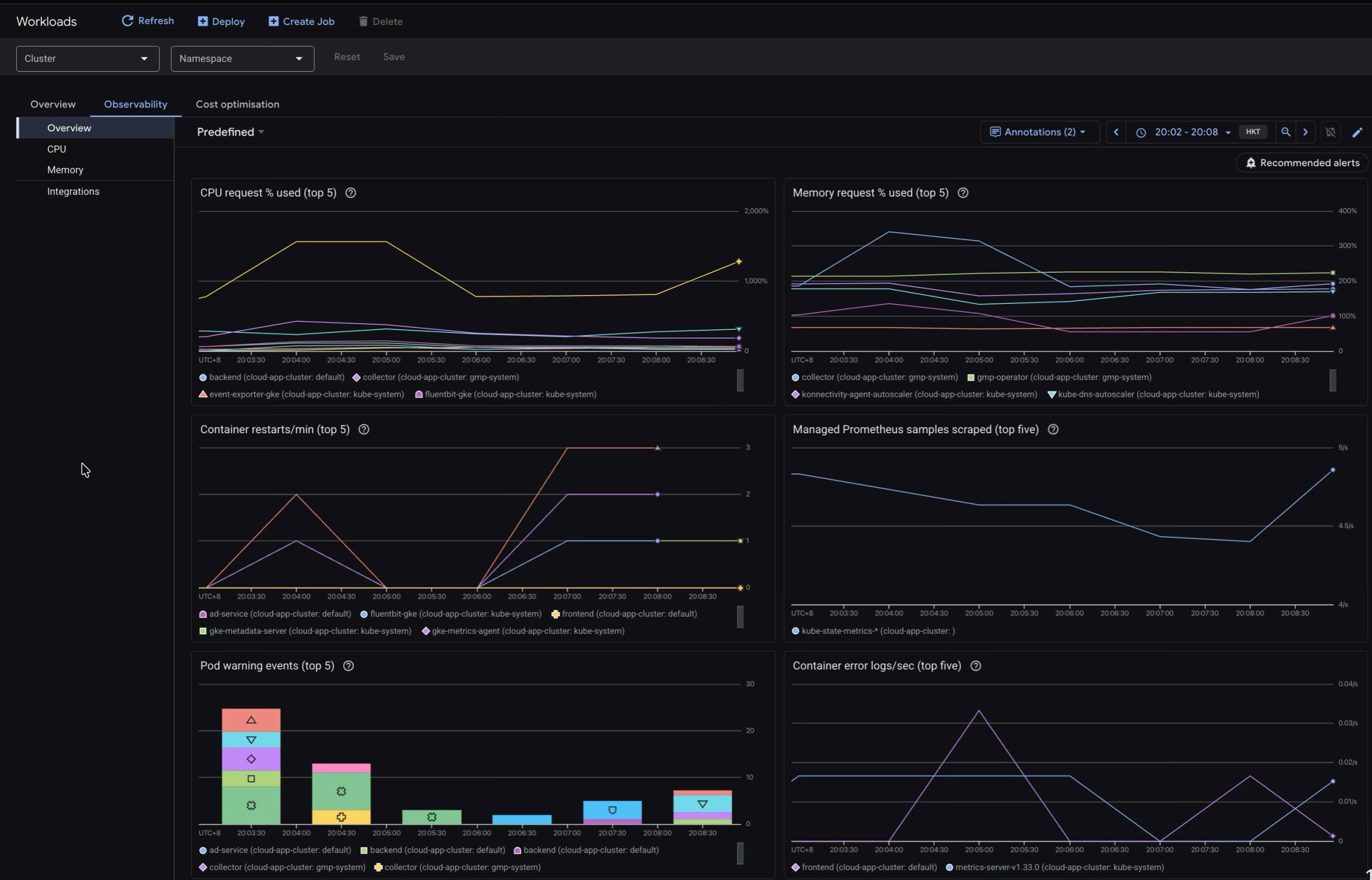
## Performance Testing

Item	Description
Performance Test Scripts	locust/locustfile.py (Locust)
Test Coverage	Feed, Detail, Likes, Comments, Health
Performance Test Reports	See below
Test Data Sets	dummy_data.sql + Ad-Service seed data

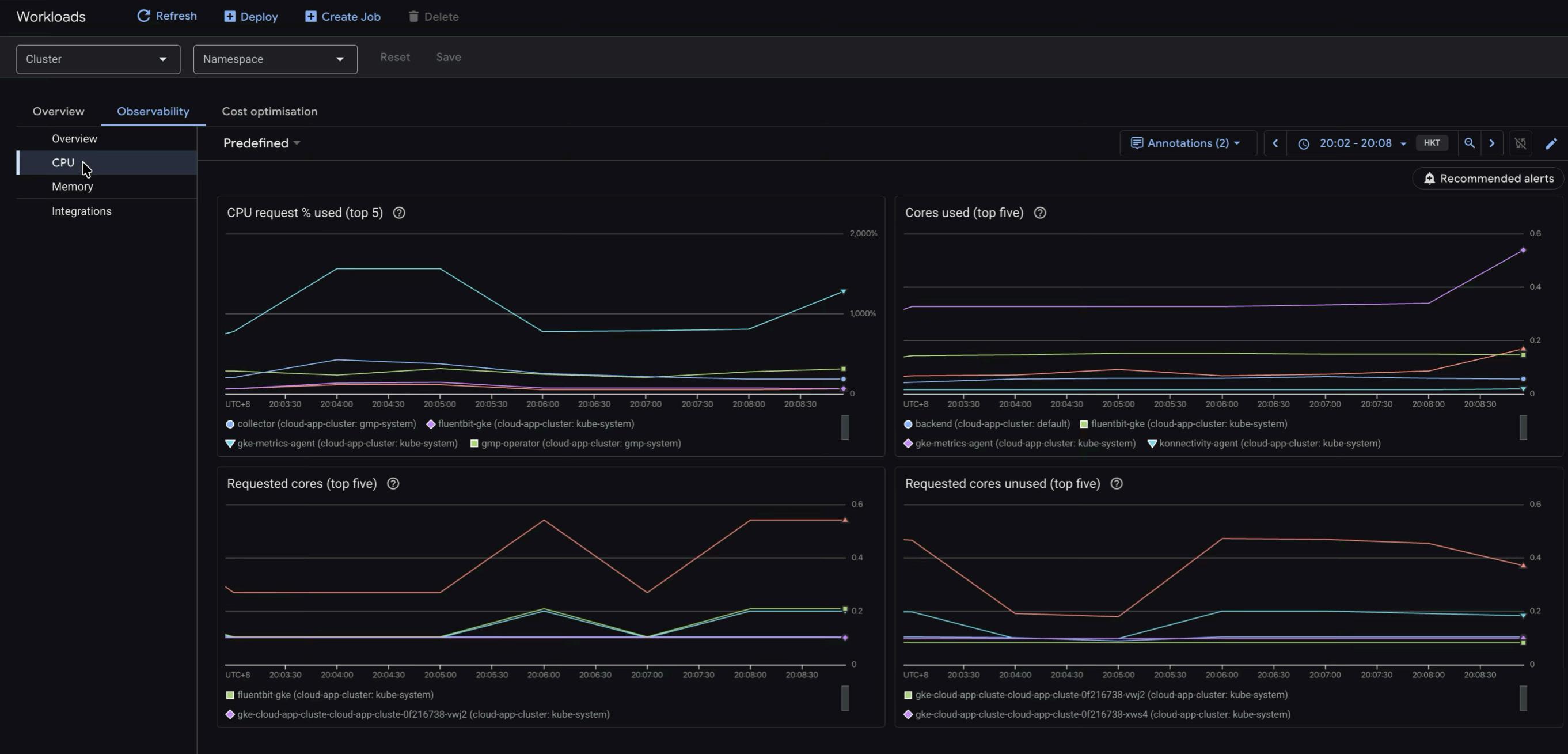
# Locust Load Test

## Video (5m)

# After Load Test



# After Load Test



# After Load Test





**Demo**  
Video



# Thanks For Listening