

NEXUSTRACE: MULTI-AGENT COLLABORATION SYSTEM FOR KUBERNETES ERROR LOG ANALYSIS

Presented by:

Najai Alarcon
Chalssa Jaculina
Mrunali Katta
Devesh Singh
Aksah Thiagarajan

Project Advisors:

Dr. Simon Shim
Dr. Lee C. Chang

Introduction

Kubernetes environments generate massive volumes of logs and error events requiring deep expertise in diagnosis. Single LLM systems frequently hallucinate and produce unsafe remediation. NexusTrace coordinates four specialized LLM agents using a Blackboard shared memory architecture to parallelize RCA agents. A Reconciliation agent, and a Validation agent covering 19 Kubernetes failure categories with a strict human-in-the-loop approval gate.



- 9,500 balanced synthetic Kubernetes incident records across 19 failure categories.
- Models: Qwen 3.5 5B, DeepSeek-R1 68, Develan-Small-2.21B, Llama 3.2 3B
- Deployed on AWS EKS via Docker containers with LoRA fine-tuned adapters.

Methodology

Multi-Agent Orchestration Pipeline

NexusTrace uses four LLM agents: parallel RCAs, a reconciliation step for remediation, and a validator for rollback/safety all gated by human approval. Adapters were fine-tuned and run on AWS EKS.

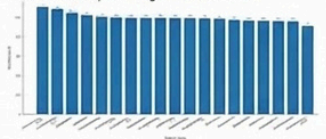


- Data: 9,500 synthetic K8s incidents → transformed into 0 self-specific SFT corpora
- Training: GLaRA 4-B4 finetuning with LoRA adapters (v18) on Google Colab A100/H100
- Deployment: Docker images pushed to Docker Hub → deployed on AWS EKS (povav trace namespace)

Analysis and Results

NexusTrace was evaluated across two profiles, end-to-end orchestration on the deployed EKS etwiss using llama3.2 5B, and standalone SFT evaluation of the offline/aimed Qwen2.5-TB-Instruct soRn adapter on a 406-example held-out set.

The SFT evaluation reported scenario-match accuracy of 0.715, section-coverage of 0.853, and BSRTScore 71 of 6.641. Five scenarios reached perfect accuracy including OOMKilled, QuotaExceeded, LivenessProbeFailed, PVCNotBound, and CurlHttpError. Manner scenarios such as FailedScheduling and ResourceQuotaExceeded will benefit from dedicated specialist agents in future work.



All five themed errorization payloads: storage failures, image-sell errors, scheduling errors, runtime crashes, and RBAC details which produce avoidance-wise diagnoses and streamlined remediation plans with warm inference latency averaging 10-25 seconds per agent call.

Evaluation Metrics

Schema validation based on all 400 SFT evaluation examples with zero mismatch errors. The Pydantic-validated RCA structure covering diagnosis, likely causes, remediation steps, rollback plan, and verification completion round-tripped through MongoDB and two frontend renderers without failures.

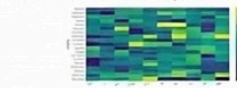
The end-to-end ingestion pipeline ran continuously without message loss across deliberate pod restarts, confirming to in-flight Kafka delivery. All orchestration evaluation suites passed with consistent behavior in both stub-wsile and K8s model testing.

Per-scenario accuracy ranged from 1.00 on OOMKilled, QuotaExceeded, and LivenessProbeFailed to 0.24-0.47 on FailedScheduling and NucleusRestart. Falsives with stereotyped event text were easiest to diagnose while scheduling and registry-auth scenarios were hardest. Diverse specialist agents for networking, storage, and security will address these gaps.

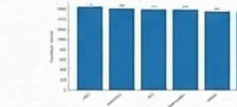


The Blackboard shared-memory architecture ensured fail-safe multi-LLMs across every reasoning stage. Each incident in MongoDB carries Kafka offset metadata and an ingested at timestamp, as any RCA output can be traced back to the exact alert that triggered it.

Two RCA agents analyzed each incident independently using different models and reasoning strategies. The Reconciliation agent prioritized disagreements and produced a unified remediation plan with kubect commands, withack instructions, and validation steps.

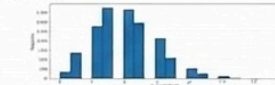


The human-in-the-loop approval gate prevented any infrastructure-changing commits from executing without explicit operator review, aligning with AI generalization best practices and ensuring operational safety throughout.

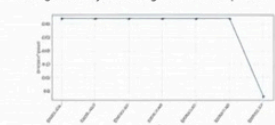


NexusTrace demonstrates that collaborative LLM reasoning on real Kubernetes infrastructure materially improves incident diagnosis compared to single-model approaches while maintaining strict operator-in-the-loop safety through human oversight.

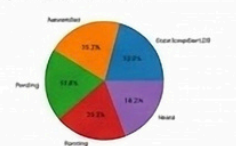
The GKCPs-managed deployment surface ensured full reproducibility where every cluster change is materialized from Git by Flux, on the etcd slaves from Kafka through Otama through the analyser: version controlled and auditable.



Kastart points peaked between 2-5, consistent with real Kubernetes CrashLoopBackOff behavior. The timeline confirms stable data generation with 1,400+ incidents daily through January 6 before generation completed.



Resource usage was broadly distributed across of CPU and memory ranges, confirming diverse operational coverage across failure scenarios.



Summary/Conclusions

NexusTrace shows has collaborative multi-agent LLM reasoning improves Kubernetes incident diagnosis compared with single-model workflows. By committing parallel RCA, reconciliation, and validation, the system produces clearer and safer remediation guidance with reliable results. The platform is deployment ready through Docker and AWS EKS, and supports both lightweight and larger models. Overall, this work shows a practical, trustworthy framework for AI-assisted cloud approvals.

Key References

- [1] Burns, B., Burnt, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes.
- [5] Burns, B., & Beda, J. (2018). *Kubernetes: Up and Running* (2nd ed.).
- [3] Bai, J., et al. (2024). Qwen2.5 Technical Report. arXiv:2412.15115.
- [4] Tian, R., et al. (2023). *DeepSeek-R1 Technical Report*.
- [6] Hu, E. J., et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.25665.
- [8] Dettmers, T., et al. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314.

Acknowledgements

We thank Dr. Simon Shim & Dr. Lee C. Chang for all the guidance and support. We also thank the Applied Data Science Department at San Jose State University and our team members for collaboration throughout this project.

Interested in partnering with us on a capstone project or exploring other ways to collaborate with the College of IDS? Contact CIDS_CAP@sjsu.edu to discuss potential projects and partnership opportunities.