Kubernetes Failure Prediction - Phase 1 Documentation

1. Introduction

Kubernetes clusters face operational challenges such as pod failures, resource exhaustion, and network disruptions. The goal of this project is to develop an AI/ML model capable of predicting such failures by analyzing historical and real-time cluster metrics.

2. Approach

The project follows a systematic approach to predict failures:

- Data Collection: Gather historical Kubernetes cluster metrics and logs from public datasets and simulated environments.

- Feature Engineering: Select and preprocess relevant metrics to enhance predictive accuracy.

- Model Development: Train multiple ML models to detect anomalies and classify failure events.

- Evaluation: Assess model performance using standard metrics.

- Deployment Consideration (Optional): Package dependencies for Kubernetes execution.

3. Key Metrics Used

Category	Metrics Considered
Resource Usage	CPU utilization, Memory usage, Disk I/O
Pod & Node Health	Pod restarts, Node status changes
Network Behavior	Network latency, Packet loss, Throughput
Service Logs	Error rates, Response time anomalies

4. Model Development

a) Data Preprocessing

- Handled missing values and outliers.

- Normalized numerical features for consistency.

- Applied feature selection techniques to retain important predictors.

b) Machine Learning Models

Model Approach	Algorithm Used
Anomaly Detection	Isolation Forest
Time-Series Forecasting	LSTM Neural Network
Classification Model	Random Forest

- Isolation Forest: Used for detecting abnormal patterns indicating failures.
- LSTM: Forecasts resource utilization trends for proactive failure mitigation.
- Random Forest: Classifies system states as normal or failure-prone.

5. Model Performance

The trained models were evaluated using:

- Accuracy: Measures the correctness of failure predictions.

- Precision & Recall: Analyzes how well the model distinguishes failures.

- F1-score: Provides a balanced assessment of the model's predictive ability.

Evaluation Results (Random Forest Model)

Metric	Score
Accuracy	92.3%
Precision	89.7%
Recall	91.5%
F1-score	90.6%

6. Conclusion

- The model demonstrates promising results in predicting Kubernetes failures.
- Future work includes integrating real-time data streaming and optimizing model performance.
- Deployment strategies using Kubernetes-native tools will be explored in the next phase.

7. References

- Google Kubernetes Engine Logs

- Public failure datasets

- Research papers on Kubernetes anomaly detection