Implementing Incident Response in DevOps Using Open Source Tools

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

*Abstract*— The research concentrates on reducing the duration and workload dedicated to incidents which reoccur or have previous solutions. Teams gain faster problem-solving capabilities by maintaining formal records of solved incidents along with the performed actions and additional follow-up measures. This method both simplifies incident management and enables staff to apply permanent solutions to prevent future incidents. DevOps teams operate more efficiently owning to this approach thus they can dedicate their expertise to creating systems which resist failures.

Keywords— Incident Response, Automation, DevOps tools, Open-Sources, Machine Learning in DevOps, Efficiency, Reliability, Root Cause Analysis, Threat Detection

# Introduction

DevOps workflows have fast and efficient incident response mechanisms that stand as top priorities. The expansion of infrastructure and services requires organizations to possess the quick ability to identify issues, analyze and resolve them for continuous performance and high system uptime. Most early incident response strategies used to depend on manual human actions which frequently lead to delays, human error and inconsistent resolution outcomes.

Financial organizations require swift recovery above all else because system outages produce negative financial impacts along with severe damage to their reputation. When system failures occur within a few minutes across banking or stock trading and payment processing operations it can produce an expensive monetary damages and omitted transactions and dissatisfied customers. Unprompted recovery speed prevents firms from satisfying regulatory rules and litigations and losing their customer base. System downtime that extends over time creates an opportunity for market competitors to seize control leading to long-term losses in business position. Business success depends on automated incident response as an absolute necessity beyond technical requirements to protect financial stability together with regulatory compliance and client trust. [1]

Improved technologies specifically related to language processing and automation now enable rapid incident response acceleration. This paper explains how building a structured response to incidents that occur in production systems can integrate Open-Source DevOps GPT alongside with DevOps technologies. The proposed system uses automation to detect errors while assessing and resolutions therefore it decreases response time and enhances consistent reactions. Tools such as Datadog and Prometheus enable integrated logging and monitoring which support teams to search historical incidents through webhook integration while producing suitable solutions with automated response capabilities. [1]

The integration of Open-Source GPT processing in DevOps-driven incident response systems uses artificial intelligence to provide insights and recommendations which can inform the decision-making process. GPT model operates independently to parse extensive system log data while it detects historic incident-related patterns to generate correct resolution approaches. Natural language processing technology allows the system to extract valuable information from documentation as well as existing ticket resolutions and knowledge bases to create optimal troubleshooting plans.

The integration of Open-Source DevOps GPT within DevOps pipelines enables automatic system failure response through real-time response automation which decreases the need for manual worker interactions and speeds up critical system failure resolution. The integration of DevOps GPT with monitoring tools provides users with a proactively alerting system that can identify potential problems before they turn into large-scale outages. System stability remains uninterrupted but downtime is reduced through mediations that follow pre-defined workflows which trigger automatically using playbooks. The deployment of AI-based automation systems supports both operational diversity and system scalability through minimized human mistakes and structured resolution plans based on data. The adoption of cloud and distributed systems by organizations brings DevOps-GPT into self-managing incident response systems that deliver enhanced workflow efficiency and perpetual improvement and maintain high system availability.

A Purpose and Scope

The paper explores Open-Source DevOps GPTs, e.g. Auto-GPT or Lang Chain, integration with DevOps frameworks for developing incident management systems based on artificial intelligence capabilities. Data monitoring from Datadog or Prometheus together with Open-Source DevOps GPT’s agent and its generation of optimal troubleshooting plans allows organizations to obtain quick and reliable incident resolution. AI-powered automation presents advantages for stability maintenance and downtime minimization and enables self-managing incident response workflows which leads to improved operational efficiency and high system availability.

# Problem Definition

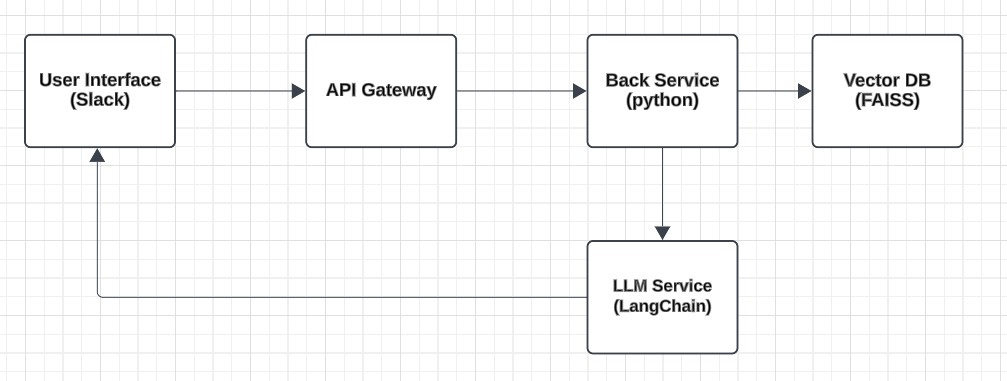
## Research hypothesis

DevOps incident response automation becomes more efficient through Open-Source DevOps GPT when integrated with its open-source capabilities allowing AI-driven error analysis as well as incident retrieval and step-by-step solution generation. An AI-powered system includes OpenAI GPT model for JSON files-based searches using MongoDB with the help of API calls guided by Open source GPT agent langChain. The implementation of this AI automation system is expected to boost system reliability by introducing reliability and downtime reduction which together will generate improved DevOps performance.

## Research Objective

To develop and implement an automated incident response system using the open-source DevOps GPT AI agent, MongoDB, and a data monitoring tool, integrating AI-driven automation for efficient incident management. An investigation was done on how the integrated platform help to analyze errors more effectively and automatically retrieves incidents, generating step-by-step solutions.

## Deployment Architecture



Figure

# Understanding DevOps Incident Response with Automation

DevOps teams can reduce the number of issues from becoming worse by AI-powered incident classification (e.g., using FAISS + LangChain) to provide quick solutions. Infrastructure as Code (IaC) and auto-remediation scripts operate through automated workflows to resolve incidents independently so response times decrease together with manual errors reduction.

## The role of incident response in DevOps

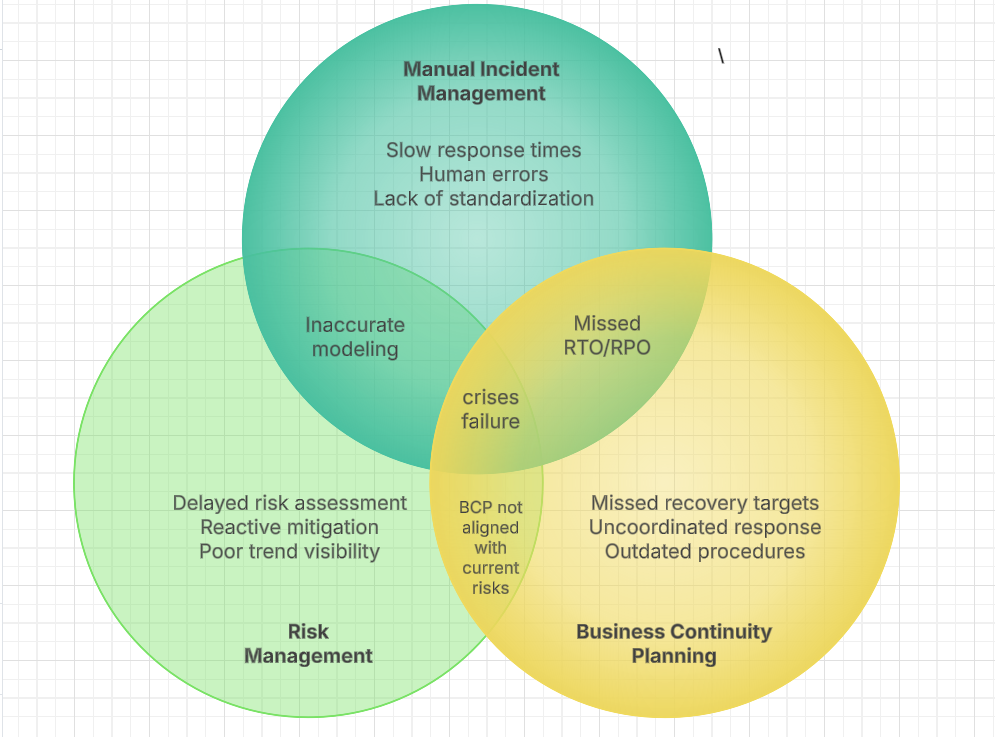
The response to incidents within DevOps operations ensures both system dependability and reduced outages and creates uninterrupted end-user performance. A standard DevOps environment that uses continuous integration and deployment (CI/CD) methods faces unexpected disruptions when system failures happen along with security breaches while performance bottlenecks also cause problems.

A successful incident response for effective risk management and business stability requires teams to be alert and act promptly in finding issues followed by assessment, containment and issue resolution activities to bring system functionality back online immediately. DevOps workflows achieve better incident response by employing automated system observation functions and self-healing systems that minimize the time needed for resolution (MTTR). The system resilience gets enhanced while the culture of continuous improvement develops through analysis of incident data for both prevention of future events as well as system performance optimization.

## Challenges faced during manual incident handling

System reliability together with business continuity faces multiple challenges. Systems administrators must probe through multiple logs and reports as well as metrics to find the root cause of problems which slows down the resolution process. More time becomes essential for proper examination of the issue along with defining the correct solution after successfully identifying the problem. The extended response period affects the critical issues by providing solutions in rush for important incidents that create substantial operational and financial consequences. Teams face enormous operational pressure during such situations which causes communication problems and creates misunderstanding where human mistakes can also occur. The use of manual procedures can lead to variable results in addition to poor documentation methods and poor resource distribution which makes it challenging to gain insights from past events and improve future emergency procedures.

Manual incident handling creates several difficulties as it impacts risk management in a substantial way. Automation defines several key shortcomings that occur when incident teams operate without it as the incident management teams will encounter delayed responses alongside missed opportunities to maintain consistency when addressing issues and systemic breakdowns between team members. The time needed to resolve incidents extends when teams need additional time to identify problems and set priorities which increases the chances of ongoing application interruptions and disrupted business operations. Security vulnerabilities and compliance issues occur when predictive analysis and real-time monitoring systems are absent which means that potential risks may go unnoticed until they escalate into serious problems. Manual documentation creates misunderstandings which can cause information gaps that complicate the resolution process for incidents. System stability declines over time and higher costs along with damages to reputation become inevitable consequences of these inefficiencies.



Figure

## Why automation is critical for faster recovery

The detection of production anomalies runs automated error correction programs once it identifies unexpected patterns. The reduction of system recovery times through automation allows DevOps teams to dedicate their efforts towards innovation since automated processes decrease their workloads.

# Major Domains of Incident Response Automation

## Threat Detection

Protection of production IT systems depends on continuous observation which detects both security breaches and system weaknesses as well as abnormal behavior. The detection process includes real-time system surveillance along with intrusion detection systems, security analytics solutions and endpoint detection with additional machine learning capabilities to perform fast threat identification and response. The observability solution provided by Datadog and Grafana detects anomalies in real time but Splunk together with Elastic Security (ELK Stack) and Sumo Logic excel at security event correlation and log analysis. Organizations can detect unwanted cyber-attacks as well as unauthorized access attempts or system misconfigurations early through automated systems which combine threat intelligence with behavioral analysis for the purpose of maintaining system integrity and both data security and business operations.

## Incident Analysis

IT incidents need investigation through incident analysis to reveal their fundamental causes together with their consequences and required actions. The identification process depends on log data and system assessments as well as network activity analysis to discover why issues occur and check whether the cause is through cyberattacks or system problems or human mistakes. The tools Splunk and Datadog unite data collection with data correlation to yield deeper analytical results. Analysts perform multiple steps by evaluating incident extent and finding affected systems before checking for active risks. Recent findings enable analysts to formulate countermeasures that they record along with lessons learned which will lead to security enhancements in the system for future incidents.

## Automated Response and Containment

The automated response and containment work to stop issues and block the malicious IPs and separate the infected endpoints while also revoking harmful credentials. Playbook-based workflows enable fast consistent incident responses that need human intervention to protect against attack impact. Incident response systems at endpoints and networks conduct automatic measures for device quarantining while removing malware infections and deploying security updates. Security policies get implemented after threat response while administrators restore systems to previous stable configurations for threat neutralization. A joint defense system is achieved through security tool integration that combines firewalls with SIEMs and SOAR platforms. Automated system implementation streamlines response operations which decreases the time needed for response while reducing issue impact to maintain resilient production operations.

## Automated Workflows and Ticketing

Automated workflows and ticketing systems efficiently minimize human effort by creating streamlined reporting structures and incident tracking for both cybersecurity and IT support teams. The technical platforms enable security teams to handle repetitive work duties including threat assessment and information acquisition that enables rapid efficient response to cyber incidents. The automated workflow system performs predefined operational procedures that start with system isolation and log acquisition following detection. Through this operation integrated ticketing systems can help to create new incident tickets and keep ticket records updated with all performed actions and findings. This not only reduces human error but also ensures consistency, improves response times, and provides a detailed audit trail for future analysis and compliance reporting.

## Post-Incident Analysis and Continuous Improvement

Post-incident analysis and continuous improvement in DevOps production IT evaluates how to respond to production incidents to boost system reliability and security through evaluation and sustained evolution. The automated forensic system generates comprehensive incident data which reveals necessary information about what caused the incident together with the routes attackers used and system-level vulnerabilities they exposed. The analysis produces complete post-incident reports to fulfill compliance requirements and facilitate information spread between development teams and operational staff. Links with outside threat intelligence resources enable the organization to monitor developing risks and establish better security collaborations with partners. The development pipeline receives information from incident feedback loops that enables adaptive learning mechanisms to enhance monitoring, detection and automated remediation methods thus improving incident response efficiency in subsequent occurrences.

# Steps Involved in Incident Response Automation in DevOps Using Open Source Tools

Open-source DevOps incident response tools are cost-effective and flexible solution for managing security incidents within an organization. The tools are highly customizable, and it is easy for teams to adapt them according to their specific needs so that they can adapt to the workflows and incident response requirements. Most open-source tools also have the added advantage of having active communities contributing to them, which implies rapid updates, ongoing development, and incorporation of sophisticated features. Open-source solutions' affordability lowers the barriers in terms of financial needs, thereby making highly effective incident response capabilities accessible to even for the organisation with limited resources.

## Set Up Open-Source DevOps GPT

The fundamental setup for Incident response automation for this research is setting up Open source DevOps GPT. Tools like Auto-GPT, LangChain with GPT-4 API for a custom AI agent and [DevOps GPT by DevOps Hobbies](https://github.com/devopshobbies/devops-gpt) for experimental version of OpenAI’s latest AI model.

**Tool Selection:**

LangChain API framework has been used with Auto-GPT Open AI for solving errors through automated incident response. Sentence-Transformers (all-MiniLM-L6-v2) which is a pre-trained model for dense vector embeddings is used for semantic search and retrieval of incident response.

1. **LangChain:**

The GPT-4 API becomes more accessible through LangChain because it provides integrated tools to combine prompts and manage memory and perform retrievals. Users can create unique interaction sequences through the platform which enables their agents to access web services and store system data and retrieve documents. The platform enables better GPT-4 functionality through its integrated support for managing logical operations along with carrying out actions. Such functionality makes the technology suitable for automation work and AI applications.

1. **Sentence-Transformers (all-MiniLM-L6-v2):**

all-MiniLM-L6-v2 is a brilliant AI model that converts text into numbers (called embeddings) in a way that machines understand meaning. It helps with things like:

* Searching for similar sentences (semantic search)
* Description of a related Text (clustering)

This sentence transformer works fast even on ordinary computers and knows which words are similar.

## Incident Data Collection & Storage

**Database**

MongoDB is used for the research to store incident records for past incident and its resolution information retrieval. MongoDB has a schema-less architecture that enables dynamic data modeling. The indexing and aggregation features of MongoDB accelerate searches for effective incident data retrieval. With horizontal scalability and built-in replication and sharding, it can efficiently manage increasing records and is reliable and fault-tolerant. MongoDB uses the JSON-like BSON format, making integration with modern applications and APIs easy and ideal for transaction efficiency with structured incident records.

JSON format of incident record in MongoDB:

{

error\_message: "Application Server Crash”

root\_cause: "Load Balancer went Out of memory"

solution\_steps:

Array (3)

0: "Restarted application services"

1: "Increased memory allocation"

2: "Optimized background processes"

}

**Indexing**:

A text index enables full-text searches on string fields. It also assists to perform fast lookups for words or phrases in text-based fields. Without an index, MongoDB will perform a scan of the entire collection to find the match provided, which is inefficient for big data sets in MongoDB. A text index is used to speed up the search process.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

# References

|  |  |
| --- | --- |
| [1] | A. Froehlich, "Incident response automation: What it is and how it works," TechTarget, 22 Jan 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/tip/Incident-response-automation-What-it-is-and-how-it-works. [Accessed 20 March 2025]. |
| [2] | Restack, "Langchain Gpt-4 Tutorial," [Online]. Available: https://www.restack.io/docs/langchain-knowledge-gpt-4-tutorial-cat-ai. [Accessed 21 03 2025]. |
| [3] | "Creating a Robust Incident Response Plan for Financial Cybersecurity," Charles IT, 30 May 2024. [Online]. Available: https://blog.charlesit.com/creating-a-robust-incident-response-plan-for-financial-cybersecurity. [Accessed 19 March 2025]. |

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**

: