Getting started with Incident Command and PagerDuty bots

Streamlining customer incident management

January 18th, 2024

Contents

Organizations are constantly challenged with efficiently managing incidents and minimizing service downtime. In response to that universal problem, Cisco has harnessed the robust capabilities of the Webex developer platform to create its own incident command bot. Combined with the PagerDuty Bot, this solution offers a powerful toolset to streamline incident management processes. This white paper provides an overview of the capabilities of incident management bots, guides on how they can be effectively implemented using the Webex developer platform and shares best practices for their use. The aim is to equip Webex developers with the knowledge to build their own bots, tailored to their specific incident management needs. It also expands on the advantages of integrating them for automating incident management.

1. Introduction

In any organization, incidents, whether minor or major, are a typical occurrence. However, their impact can be significantly mitigated when they are effectively managed. Traditionally, incident management has largely been a manual process, often leading to slower response times, increased downtime, and potential human error. That not only hampers productivity but can also negatively affect the customer experience and result in financial losses.

In the quest to address those challenges, automation has emerged as a game-changer. Specifically, solutions like the incident command bot and the PagerDuty Bot, have become key players in transforming the landscape of incident management. Developed by Cisco using the Webex Developer Platform, their custom incident command bot, along with the PagerDuty Bot, can automate, orchestrate, and expedite the incident response process.

These bots not only augment efficiency but also enhance the customer experience by minimizing downtime. This whitepaper delves into the functionalities of these two bots, the benefits they offer, their implementation process, and the best practices for their use. The aim is to provide a comprehensive guide for leveraging these tools to optimize incident management in your organization.

2. Cisco’s incident command bot solution

2.1 Purpose and benefits

Cisco uses its Incident Command Bot as a central hub for managing incidents. Its primary purposes include:

- Incident Detection: Automated incident detection.
- Incident Engagement: Automated incident engagement.
- Incident Triage: Quickly assess the severity of incidents.
- Automated Actions: Trigger predefined actions in response to incidents.
- Communication Capabilities: Notify relevant teams and stakeholders.
- Documentation Tools: Maintain detailed incident logs and reports.

The benefits of using an Incident Command Bot include:

- Faster Response: Swiftly identify and respond to incidents.
- Reduced Downtime: Minimize the impact on customers.
- Improved Accountability: Easily track incident ownership and resolution progress.
- Data-Driven Insights: Analyze incident data for continuous improvement.
- Timeline Tracking: Timeline data tracked for all events as they occur.
- Inquiries and Actions: Real time automated inquiries and actions based on inputs.
- Seamless Integration with Webex: For Cisco users, the Incident Command Bot integrates effortlessly with Webex – the platform already in use for daily collaboration.

2.2 Implementation

To implement an Incident Command Bot:

1. Define Incident Workflows: Map out incident response workflows, including detection, classification, escalation, and resolution steps.
2. Bot Integration: Integrate the bot with incident management tools, monitoring systems, and Webex messaging spaces and meetings.
3. User Training: Train relevant teams on how to use the bot effectively.
4. Testing and Refinement: Conduct tests and refine workflows as needed.

2.2.1 Integrate an incident command bot with Webex

Integrating an Incident Command Bot with Webex involves a few steps, but it's achievable. Here's a step-by-step guide to help you with the integration:

1. Define the Bot's Functionality:

   Before you begin, determine the specific functionality and purpose of your incident command bot. What kind of information and actions should it provide in the context of incidents? Where does that information reside? Are there APIs available for your bot to access this information?

2. Choose a Bot Framework:

   You'll need a bot framework to create and host your incident command bot. Popular choices include Webex Node Bot Framework, Dialogflow, or custom development using a programming language like Python or Node.js.

3. Create a Webex Developer Account:

   To integrate with Webex, you'll need a Webex developer account. You can use your Webex account at the Webex for Developers portal at developer.webex.com.

4. Create a Bot:

   In your Webex developer account, create a new bot integration. You'll receive a bot token that you'll use to authenticate your bot with the Webex API.

5. Configure Webhooks:

   Set up webhooks in Webex to allow your bot to receive messages. You'll need to define the endpoint URL where your bot will listen for incoming messages. This URL should point to your bot's server. (Note that the Webex Node Bot Framework automatically creates these webhooks for you!)

6. Integrate with Your Bot Framework:

   In your bot's code, use the Webex SDK or API to handle incoming and outgoing messages. Ensure your bot can respond to commands and questions related to incident management. Add your bot token to the framework here.

7. Test the Integration:

   Test your incident command bot within the Webex platform to ensure it can receive and respond to messages effectively. Verify that the bot understands and executes incident-related commands.

8. Iterate and Improve:

   Continuously gather feedback from users and stakeholders to improve the bot's functionality and user experience. Iterate your bot's capabilities based on real-world usage.

9. Create Documentation and Support Resources:

   Create documentation for your incident command bot and offer support to users who may have questions or encounter issues.

10. Handle User Training:

    Educate your users on how to interact with the bot effectively. Provide user documentation or training materials as needed.

Remember that the exact steps and tools may vary depending on the bot framework you choose and the specific requirements of your incident command system. Be sure to refer to the documentation provided by your chosen bot framework and the Webex Developer Portal for detailed instructions and resources.

2.3 Best practices

Keep in mind the following best practices during design and development:

- Clear Workflow Documentation: Document incident response workflows clearly, so all team members understand their roles and responsibilities.
- Regular Updates: Keep the bot and workflows up to date to reflect changes in your organization's systems and processes. Monitor Webex Developer blogs and webinars for new features that your bot could implement.
- Post-Incident Analysis: After an incident, conduct post-mortem analyses to identify areas for improvement. The bot can even be configured to automate the post-mortem collection and report generation itself.

3. PagerDuty bot

The PagerDuty Bot is one of many integrations that can be used as a critical component of your incident management process, focused on alerting and on-call scheduling. Unlike the custom Incident Command Bot, the PagerDuty Bot is available for all Webex users on the Webex App Hub and requires no in-house development to utilize. Since Cisco has been using PagerDuty as its incident notification platform, the bot can be configured with its existing configuration. Its key purposes include:

- Incident Alerting: PagerDuty allows for the automatic and real-time alerting of on-call teams when incidents occur.
- Incident Tracking: It provides a centralized platform for tracking incidents, enabling teams to monitor their status and resolution progress.
- Escalation Policies: PagerDuty enables the creation of escalation policies to ensure that incidents are addressed promptly and that responsibilities are clear.
- Response Automation: It offers automation and workflows to streamline incident response, reducing manual effort.
- Incident Reporting: Teams can analyze and report on incident data to identify patterns and areas for improvement.
- Integration: Integrate with monitoring and alerting tools.

PagerDuty is essential for incident response and resolution for several reasons:

- Real-Time Incident Management: PagerDuty ensures that incidents are promptly detected and reported in real-time, minimizing downtime and service disruptions.
- Efficient Team Coordination: It centralizes incident management, allowing for efficient coordination among teams and ensuring the right people are alerted and involved.
- Reduced MTTR (Mean Time To Resolution): PagerDuty helps organizations reduce MTTR by automating incident response processes, improving response times, and ultimately minimizing the impact of incidents on customers and operations.
- Enhanced Visibility: PagerDuty provides visibility into incident data, which helps organizations analyze past incidents, identify root causes, and implement preventive measures.
- Improving Customer Experience: Quick and effective incident resolution provided by PagerDuty leads to improved customer satisfaction and trust.

3.1 Implementation

To implement a PagerDuty Bot:

1. Choose PagerDuty: Sign up for a PagerDuty account if you haven't already.
2. Define On-Call Schedules: Create and manage on-call schedules for teams and individuals.
3. Configure Alerting Rules: Define rules for alerting, escalation, and notification.
4. User Training: Train relevant teams on how to acknowledge and resolve alerts.
5. Bot Integration: Integrate the PagerDuty Bot with your Incident Command bot.

3.2 Best practices

Consider the following best practices when planning and implementing your integration:

- Well-Defined Escalation Policies: Establish clear escalation policies to ensure that critical incidents are promptly escalated to the right personnel.
- Regular Testing: Regularly test alerting and on-call processes to ensure they work as intended.
- Feedback Loops: Encourage teams to provide feedback on the effectiveness of alerting and escalation policies.

4. Integrating your incident command and PagerDuty bots

To maximize the benefits of both bots, consider integrating them together. Benefits include:

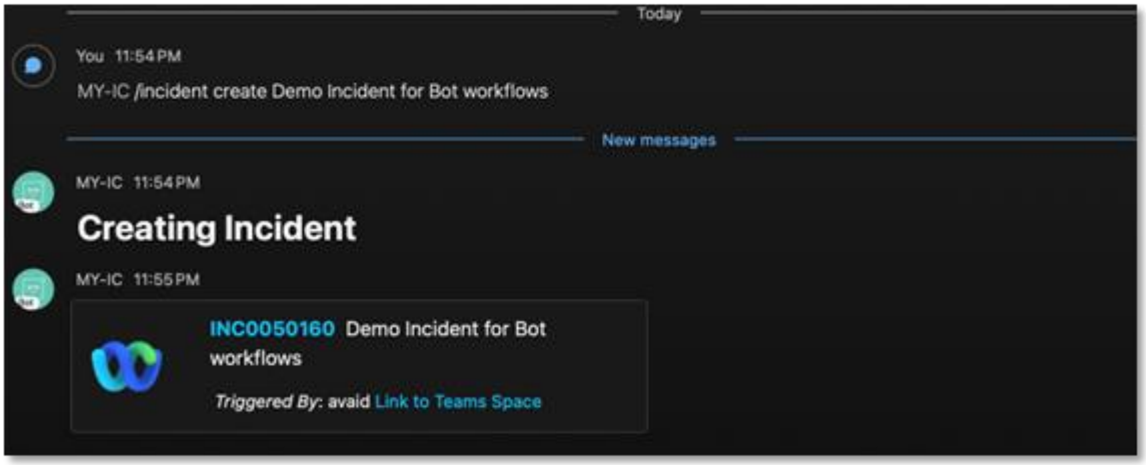
- Alert to Incident: Use PagerDuty Bot to alert on-call personnel, and automatically create incidents in the Incident Command Bot for immediate response.
- Data Sharing: Share incident data between the bots for a comprehensive incident management experience.
- Feedback Loop: Enable feedback mechanisms between the bots to continuously improve incident management processes.
- Code Reduction: There's no need to build notification and escalation processes into your Incident Command Bot if you are utilizing the PagerDuty Bot.

5. Cisco incident command and Pager Duty Use cases

The following section provides examples of how Cisco itself is using the Incident Command and PagerDuty bots to streamline its own workflows.

5.1 Dynamically create an incident

When triggered by custom integration logic, an incident is created using the Incident Command bot:



Today

You 11:54 PM

MY-IC /incident create Demo Incident for Bot workflows

New messages

MY-IC 11:54 PM

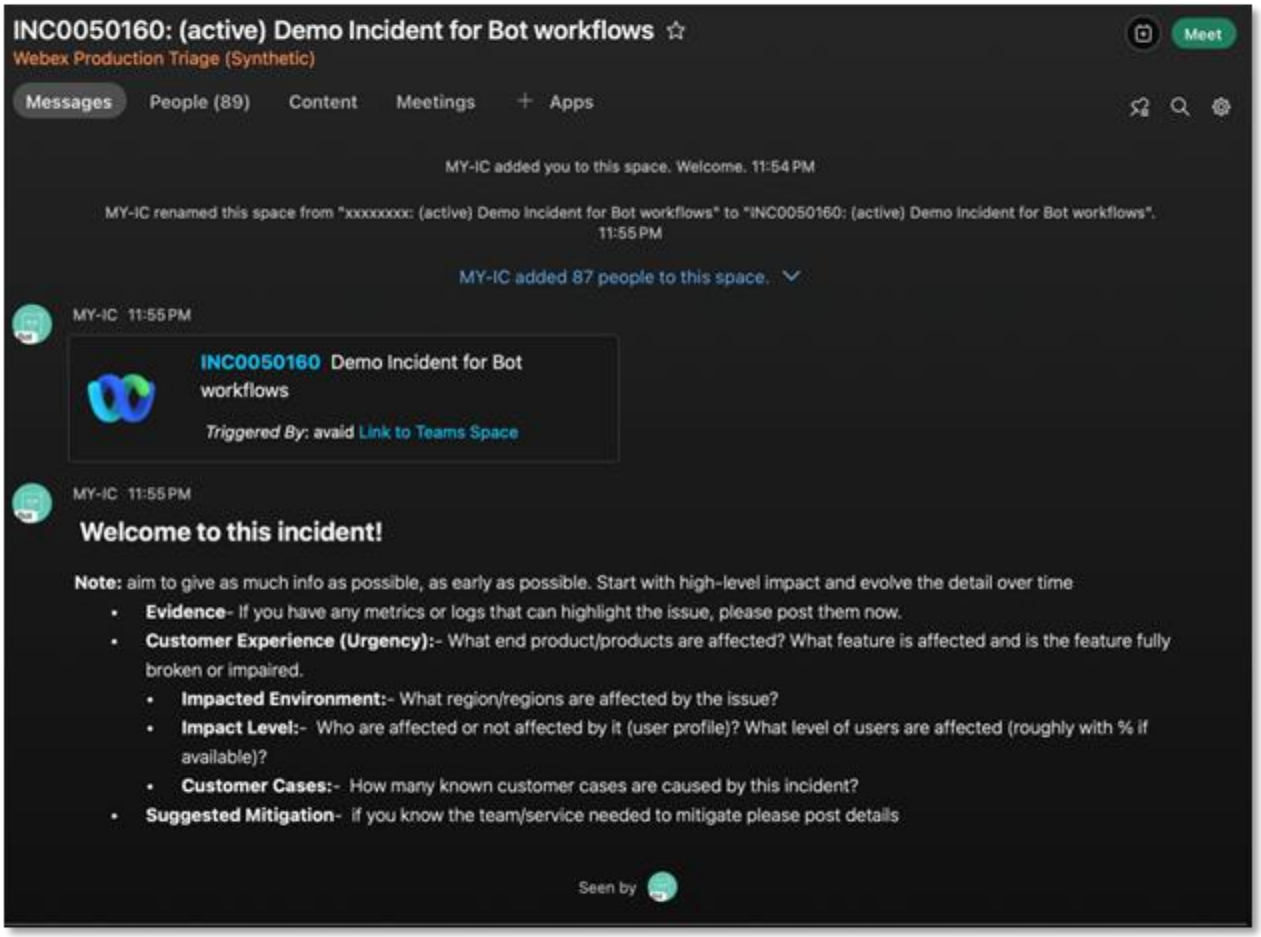
Creating Incident

MY-IC 11:55 PM

INC0050160 Demo Incident for Bot workflows

Triggered By: avaid Link to Teams Space

Based upon the space being used, specific rules can be applied to include responders and other interested collaborators, as well as important details on the incident itself:



INC0050160: (active) Demo Incident for Bot workflows ☆

Webex Production Triage (Synthetic)

MY-IC added you to this space. Welcome. 11:54 PM

MY-IC renamed this space from "xxxxxxxx: (active) Demo Incident for Bot workflows" to "INC0050160: (active) Demo Incident for Bot workflows". 11:55 PM

MY-IC added 87 people to this space.

MY-IC 11:55 PM

INC0050160 Demo Incident for Bot workflows

Triggered By: avaid Link to Teams Space

MY-IC 11:55 PM

Welcome to this incident!

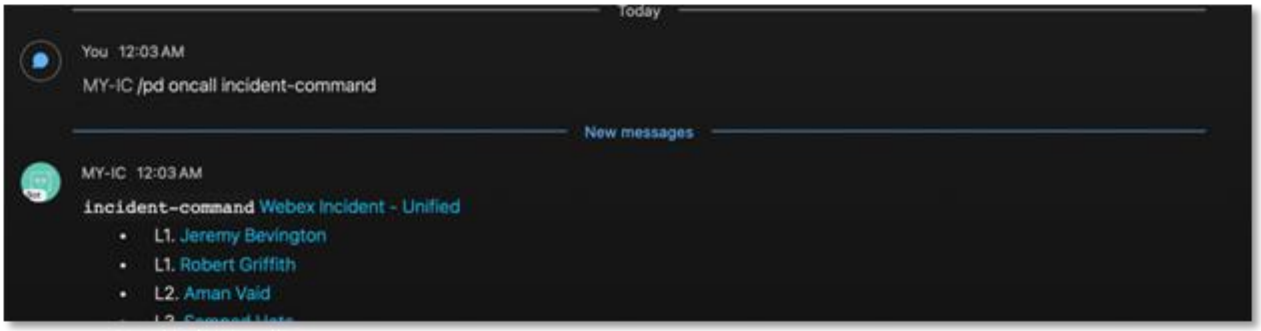
Note: aim to give as much info as possible, as early as possible. Start with high-level impact and evolve the detail over time

- **Evidence**- If you have any metrics or logs that can highlight the issue, please post them now.
- **Customer Experience (Urgency):**- What end product/products are affected? What feature is affected and is the feature fully broken or impaired.
  - **Impacted Environment:**- What region/regions are affected by the issue?
  - **Impact Level:**- Who are affected or not affected by it (user profile)? What level of users are affected (roughly with % if available)?
  - **Customer Cases:**- How many known customer cases are caused by this incident?
- **Suggested Mitigation**- if you know the team/service needed to mitigate please post details

Seen by

5.2 Engage relevant teams using the PagerDuty bot

Using the PagerDuty bot, response teams can be easily notified and rallied:



You 12:03 AM

MY-IC /pd oncall incident-command

MY-IC 12:03 AM

`incident-command` Webex Incident - Unified

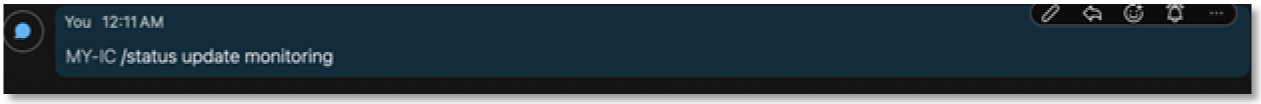
- L1. Jeremy Bevington
- L1. Robert Griffith
- L2. Aman Vaid

Useful PagerBot duty commands include:

- `oncall`: return a list of on-call resources.
- `page`: page a particular service team included in the space.
- `sp`: (soft page) level one engineers are added to the space but are **not directly paged** via the bot.
- `service list`: list all configured services.

5.3 Integrate with ServiceNow

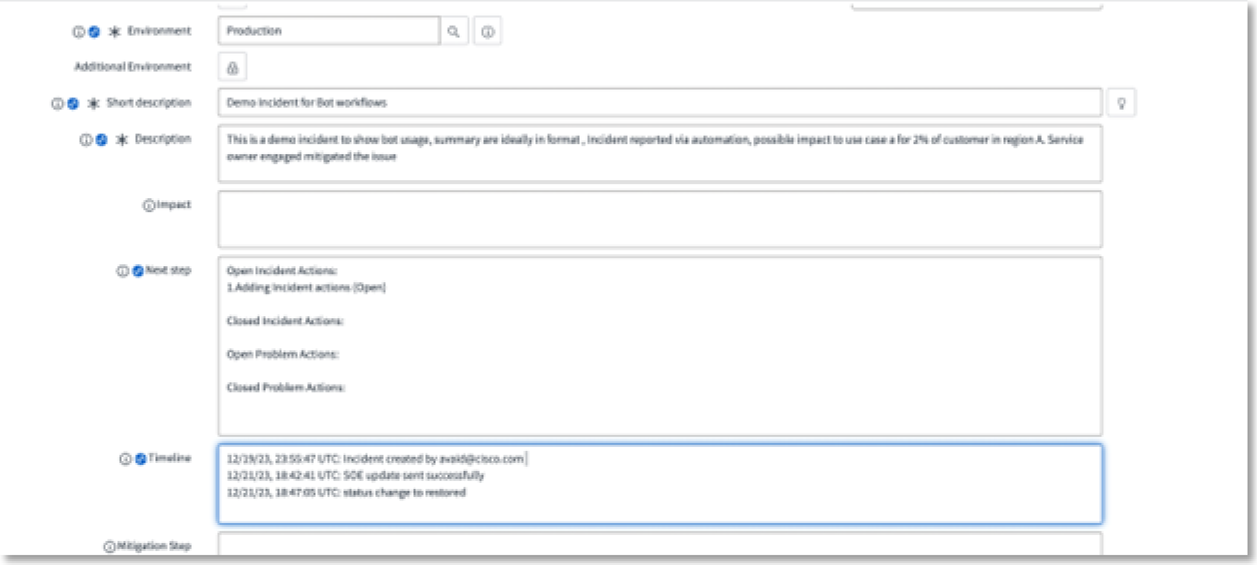
Incident states can be easily managed and updated in ServiceNow right from within the incident space:



You 12:11 AM

MY-IC /status update monitoring

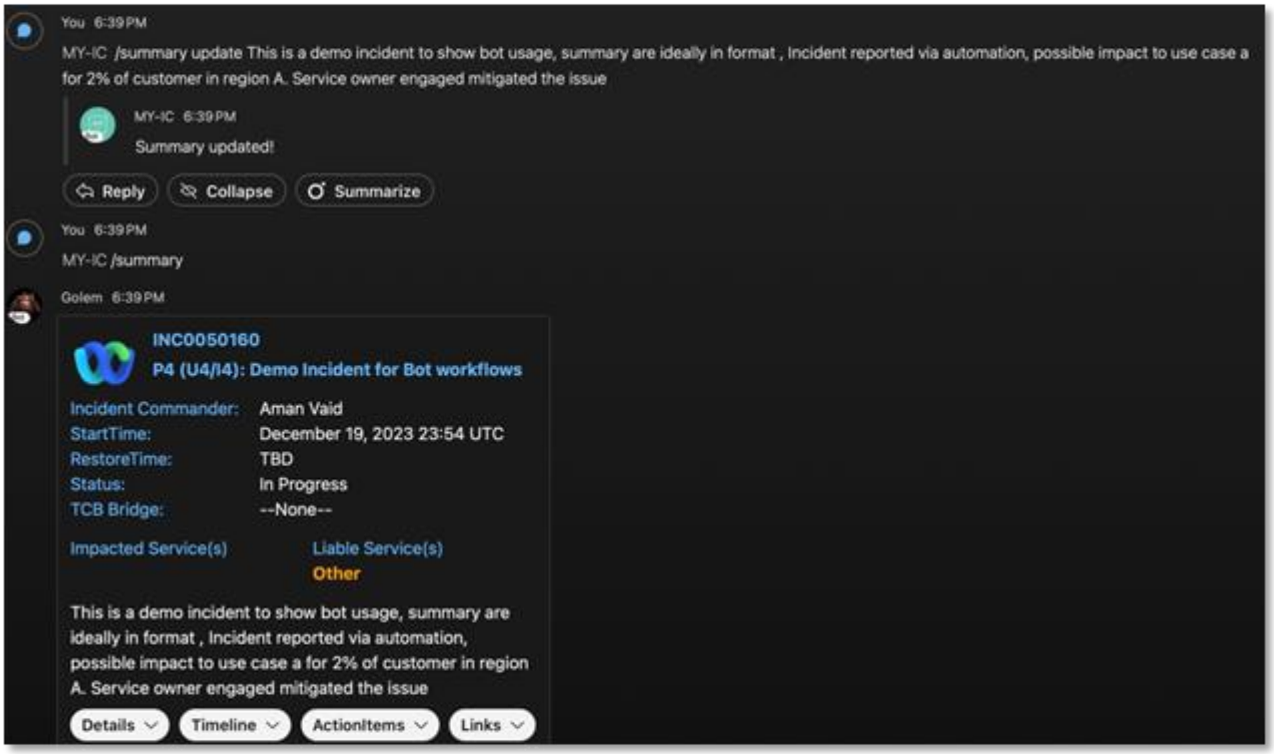
The incident information is available within ServiceNow for further workflow tracking:



✱ Environment	Production
Additional Environment	
✱ Short description	Demo incident for Bot workflows
✱ Description	This is a demo incident to show bot usage, summary are ideally in format , incident reported via automation, possible impact to use case a for 2% of customer in region A. Service owner engaged mitigated the issue
Impact	
Next step	Open Incident Actions: 1.Adding incident actions [Open] Closed Incident Actions: Open Problem Actions: Closed Problem Actions:
Timeline	12/19/23, 23:55:47 UTC: Incident created by awaid@cisco.com 12/21/23, 18:42:41 UTC: SOE update sent successfully 12/21/23, 18:47:05 UTC: status change to restored
Mitigation Step	

5.4 Update incident summaries

Incident summaries can be updated and recalled into the incident space at any time, ensuring that context is never lost:



You 6:39 PM

MY-IC /summary update This is a demo incident to show bot usage, summary are ideally in format , Incident reported via automation, possible impact to use case a for 2% of customer in region A. Service owner engaged mitigated the issue

MY-IC 6:39 PM

Summary updated!

You 6:39 PM

MY-IC /summary

Golem 6:39 PM

INC0050160

P4 (U4/I4): Demo Incident for Bot workflows

Incident Commander: Aman Vaid

StartTime: December 19, 2023 23:54 UTC

RestoreTime: TBD

Status: In Progress

TCB Bridge: --None--

Impacted Service(s)

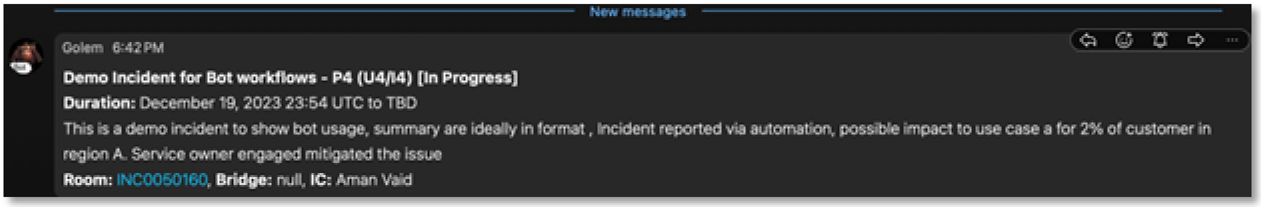
Liable Service(s)

Other

This is a demo incident to show bot usage, summary are ideally in format , Incident reported via automation, possible impact to use case a for 2% of customer in region A. Service owner engaged mitigated the issue

5.5 Post to other relevant spaces

Incident descriptions can be posted to other spaces. For instance, a space containing executive management can be provided with a concise description and status for severe, high-priority incidents if necessary:



Golem 6:42 PM

Demo Incident for Bot workflows - P4 (U4/I4) [In Progress]

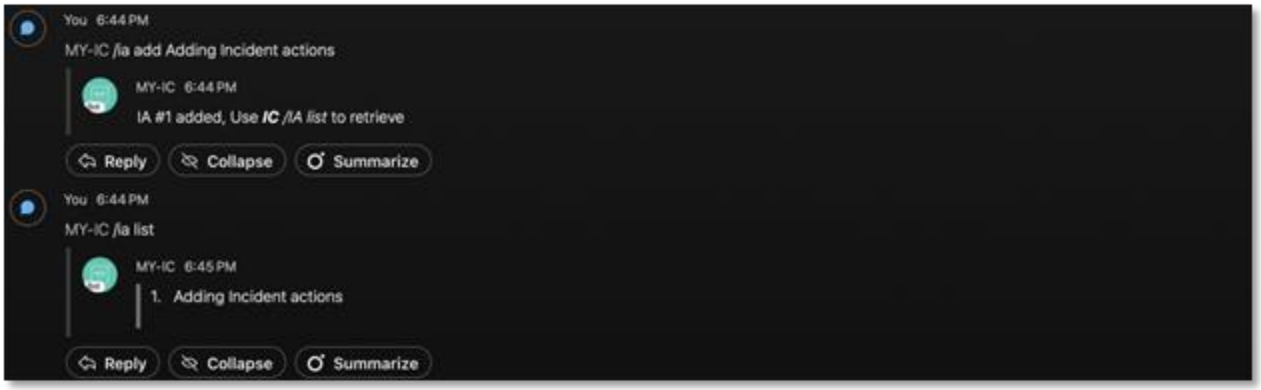
Duration: December 19, 2023 23:54 UTC to TBD

This is a demo incident to show bot usage, summary are ideally in format , Incident reported via automation, possible impact to use case a for 2% of customer in region A. Service owner engaged mitigated the issue

Room: INC0050160, **Bridge:** null, **IC:** Aman Vaid

5.6 Create incident monitors

Incidents and problem actions can be monitored, and reminders can be set to take action:



You 6:44 PM

MY-IC /ia add Adding Incident actions

MY-IC 6:44 PM

IA #1 added, Use ***IC*** */IA list* to retrieve

You 6:44 PM

MY-IC /ia list

MY-IC 6:45 PM

1. Adding Incident actions

6. Other real-world use cases

Additional real-world examples of organizations successfully using Incident Command and PagerDuty Bots include:

- E-commerce: Swiftly respond to website outages and payment gateway issues.
- SaaS Providers: Manage server outages and performance incidents efficiently.
- Healthcare: Ensure rapid response to patient data system downtimes.

One Webex customer, Sutherland Global, recently shared on stage at Webex One 2023 that they have built their own Incident Command Bot to successfully automate incident management.

7. Conclusion

The utilization and integration of Incident Command and PagerDuty Bots represent a substantial step forward in the field of incident management. These innovative tools enable organizations to navigate the challenges of incident management effectively, leading directly to improved operational efficiency and customer satisfaction. By leveraging these bots, businesses can not only react swiftly to incidents but also proactively prevent potential issues, thereby reducing downtime significantly.

As we move forward, the importance of automated incident management will only continue to grow in our increasingly digital world. Tools like the Incident Command and PagerDuty Bots will be at the forefront of this evolution, aiding businesses in their pursuit of operational excellence and superior customer experience.

Hence, the time to embrace this transformation is now. The guidance provided in this whitepaper will assist you in implementing these powerful tools and maximizing their potential benefits. As you embark on this journey, remember that adaptation and continuous improvement will be key to achieving the full potential of automated incident management.

8. Resources

Use the following resources to begin your own incident management transformation.

- Webex Developer Platform: Access the Webex Developer Platform to create a Webex developer account and explore tools and resources for building incident management bots.
- PagerDuty: Sign up for PagerDuty to enhance your incident alerting and on-call scheduling capabilities.
- Webex App Hub: The PagerDuty Bot is available for all Webex users on the Webex App Hub and requires no in-house development to utilize.
- Webex Node Bot Framework: Choose the Webex Node Bot Framework for creating and hosting your own incident command bot.