

Making a Virtual War Room — the Journey to ChatOps - Robert Barron - Medium

Robert Barron : 8-10 minutes : 4/13/2020



The virtual war room described in a [previous article](#) is, to a certain extent, an ideal state — people co-operating seamlessly with the aid of automated bots that interject with just the right amount of information, not too little and not too much. As a reminder, ChatOps, in a nutshell, is *the integration of tools and processes into a collaboration platform so that teams can be more efficient.*

The question now becomes *How do I create my virtual war room? How do I start enjoying the fruits of implementing ChatOps?*



A physical war room — [Sailors visit the Churchill War Rooms Museum in London \(US Navy\)](#)

The simple truth is that achieving ChatOps, like most accomplishments, is a journey where you start at the bottom and slowly reach the level that is best for

you. In my experience, common triggers which launch the start of a ChatOps journey include the facts that:

- The organization as a whole wants to improve the way they communicate amongst themselves. They want something more easily accessible than WhatsApp groups or direct messaging solutions. They video-share, but it's difficult to record and retrieve the knowledge created during those online conversations. ChatOps enables communication, collaboration, and learning by watching others work.
- A Dev/DevOps team “shifts right” and wants to leverage ChatOps to keep up-to-date on the production status of their service. Often these teams will have end-to-end responsibility for their service and need ChatOps to primarily communicate amongst themselves before escalating to other teams using ChatOps.
- An Operations (Ops) team “shifts left” and needs to leverage ChatOps to manage multiple incidents over multiple services. An Ops team usually has availability responsibility for numerous services, applications, and infrastructure. They will use ChatOps to aid in triage and communicate with multiple other teams.

ChatOps lends itself to an agile kind of experimentation where new options are easily explored. Adopting ChatOps is a journey where the organization is constantly improving the way they operate, either by adding new capabilities or changing their process. Site Reliability Engineers often use ChatOps as an access point to their homegrown tools.

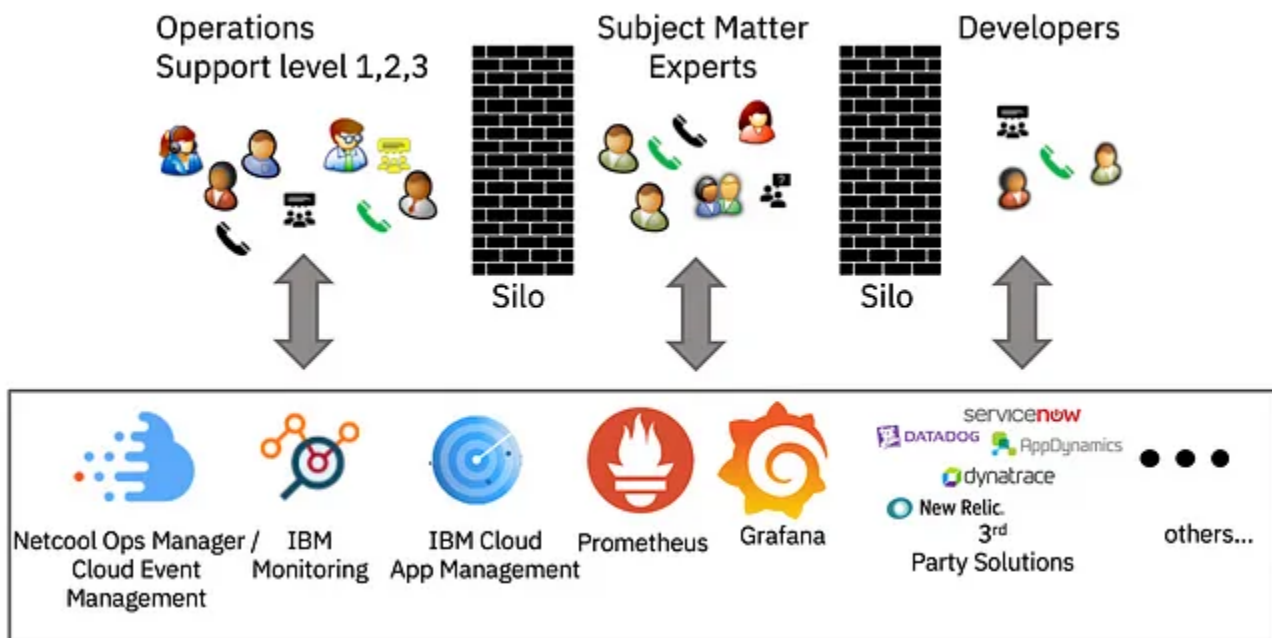
ChatOps is a Journey.

The following stages present a typical journey of an Operations team adopting ChatOps and thereby improving their internal and external collaboration in parallel with automating tasks.

Stage 0 — Ad-hoc collaboration (before ChatOps)

This is the pre-ChatOps stage. You use multiple technical tools to manage the production environment, collecting metrics and logs and alerting Operations and other subject matter experts (SMEs) when errors occur.

Different teams may or may not collaborate and, even when they do, this collaboration is often done either physically face-to-face or remotely using a technology that does not record their actions (phone calls, video conferences, phone messaging systems).



Stage 0 — Ad-hoc collaboration (before ChatOps)

Opportunities to learn from each other are limited, opportunities to cooperate across organizational lines are hampered. Solving problems is simply too time-consuming.

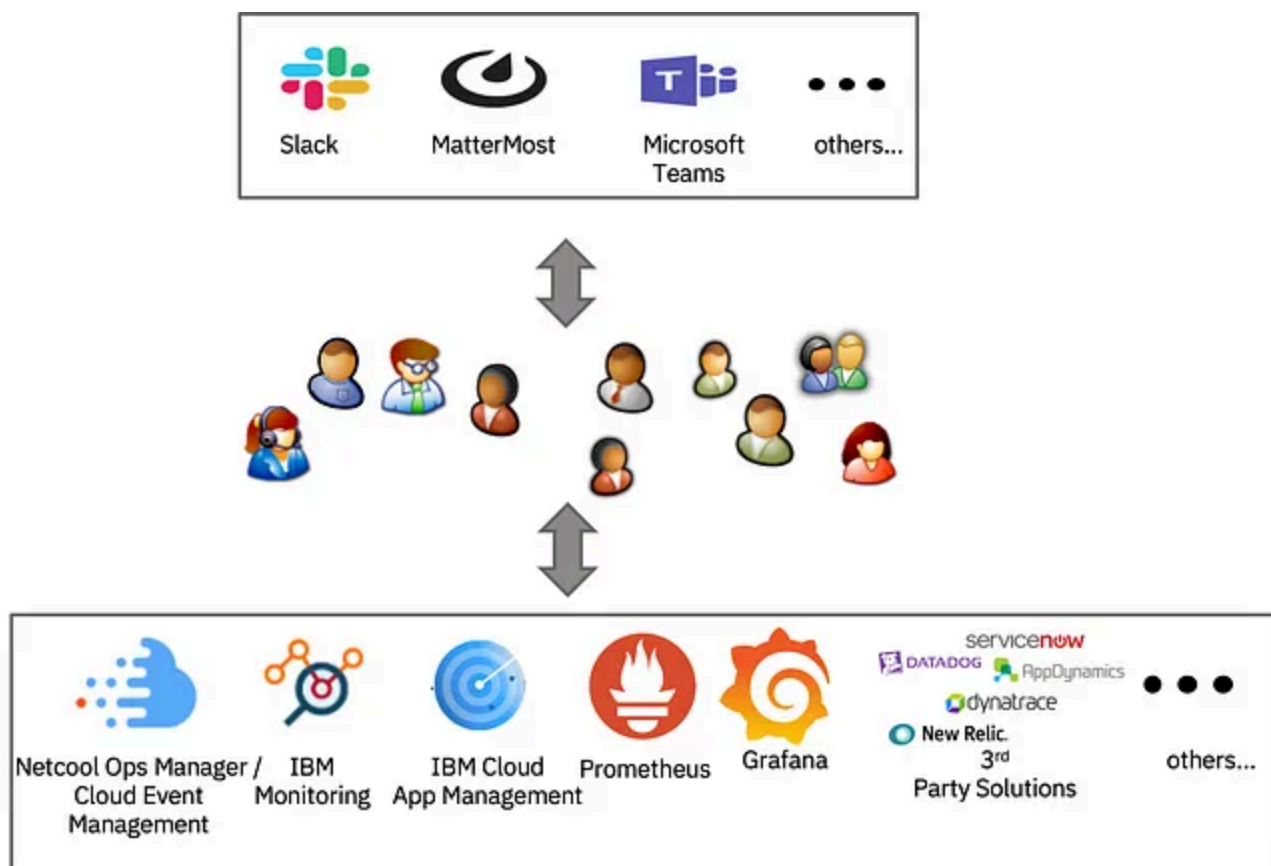
Stage 1 — Human-to-human collaboration

Once you've decided to improve communication and use a collaboration tool, you can use it to create a simple virtual war-room for communications and knowledge management.

While each expert will continue to use their individual tool, conversations can be held in dedicated chat rooms (or channels) and used for post-mortem and retrospective analysis. Common steps at this stage include:

- Creation of a dedicated room/channel for high severity incidents
- Manual documentation of technical tasks in the chatroom
- Upload of logs and configuration files
- Performing tasks in a terminal and copy/pasting the resulting commands in the chat window.
- Inserting screenshots of tools and dashboards into the chat window.

The value in doing all these tasks in the chat window is that you get a persistent record of how you dealt with the incident, communications are much clearer. For example, seeing that someone has made a typo while executing command is virtually impossible over the phone and even while screen-sharing in a conference call is easy to "blink and miss". On the other hand, adding the text of the command and response into the chat window makes it available forever.



Stage 1 — Human-to-human collaboration

Implementing Stage 1 reduces “Mean Time To Know” — clearer communications means you understand what’s going on that much faster.

Stage 2 — Single direction automations

The simplest automations available are uni-directional. You access external tools while trying to solve the incident without needing to “swivel”, “pivot” or “context switch”.

So instead of being alerted to a new incident by an email or seeing a red light on a dashboard, you are informed by an automation (a product like Pager Duty or [Cloud Event Management](#)) sending you a chat message, alerting you to the problem. Simple messages in emails, SMSs or WhatsApps are usually very limited in the information they send you. A collaboration platform can forward a full triage summary, complete with tables and graphs.

Other examples of single direction integrations include DevOps tools such as Github and Jenkins sending messages into the chat when something changes.

The incoming alerts let you know that something has happened. Once you start dealing with the problem, you might want to access a knowledge base to get more information — perhaps there are older trouble tickets with documented solutions you can use?

Instead of opening a browser window to search, you might want to query ServiceNow or [IBM Control Desk](#) directly from the chat window.

When investigating a performance problem, instead of opening a new dashboard

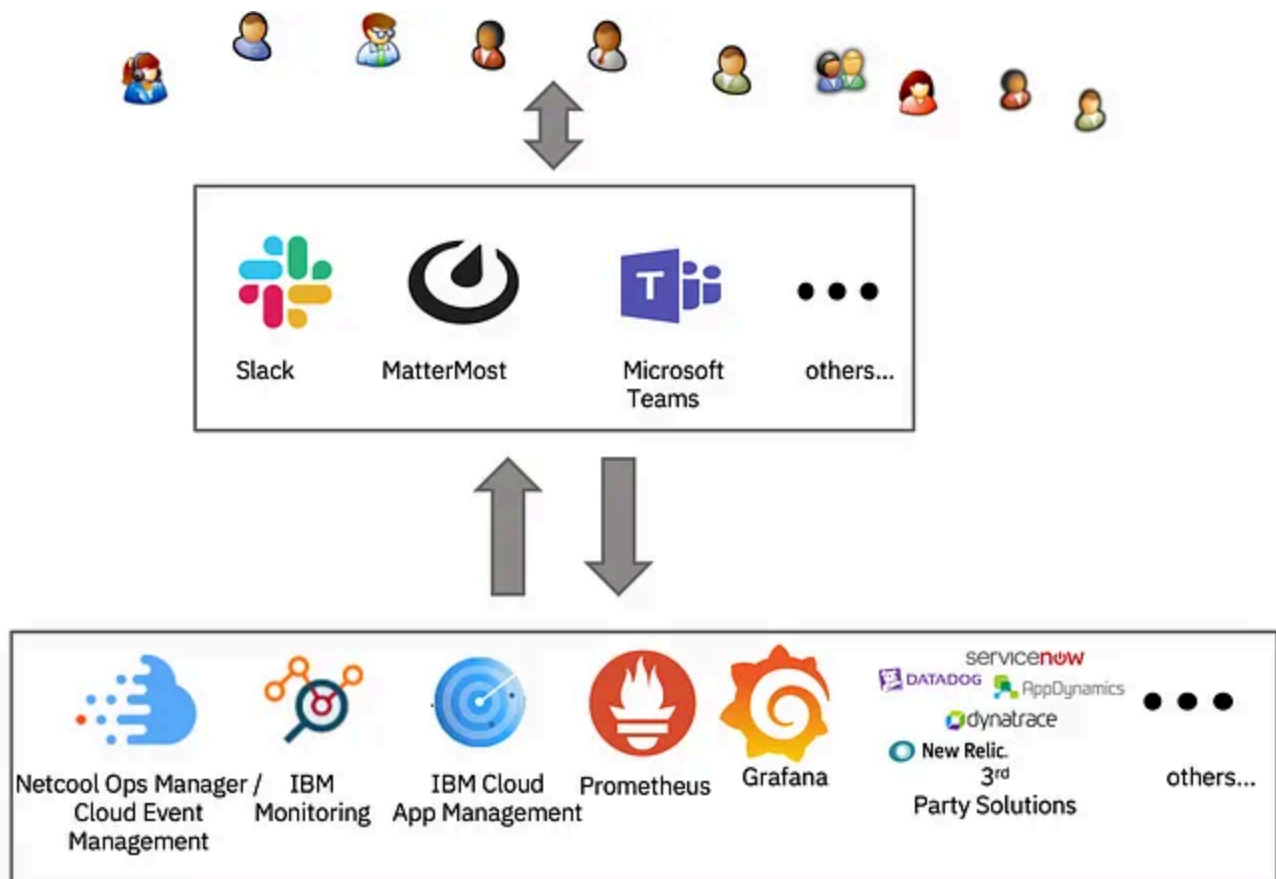
for yourself you can pull the graphs and other diagrams into the chat window so that everyone else in the chat can also see.

Implementing Stage 2 means that information is pushed to you more clearly and that your actions are that much faster.

Stage 3 — Bi-directional automation

While Stage 2 automations may be simply querying a database or performing a “fire and forget” task, Stage 3 makes the links between the humans typing commands (or pressing buttons) in the chat window to the underlying systems much tighter.

Instead of simply triggering a command remotely, you will see the result of your command in the chat window. Instead of merely creating a new ticket, you can update and edit the contents of tickets and other databases.



Stage 3 — Bi-directional automation

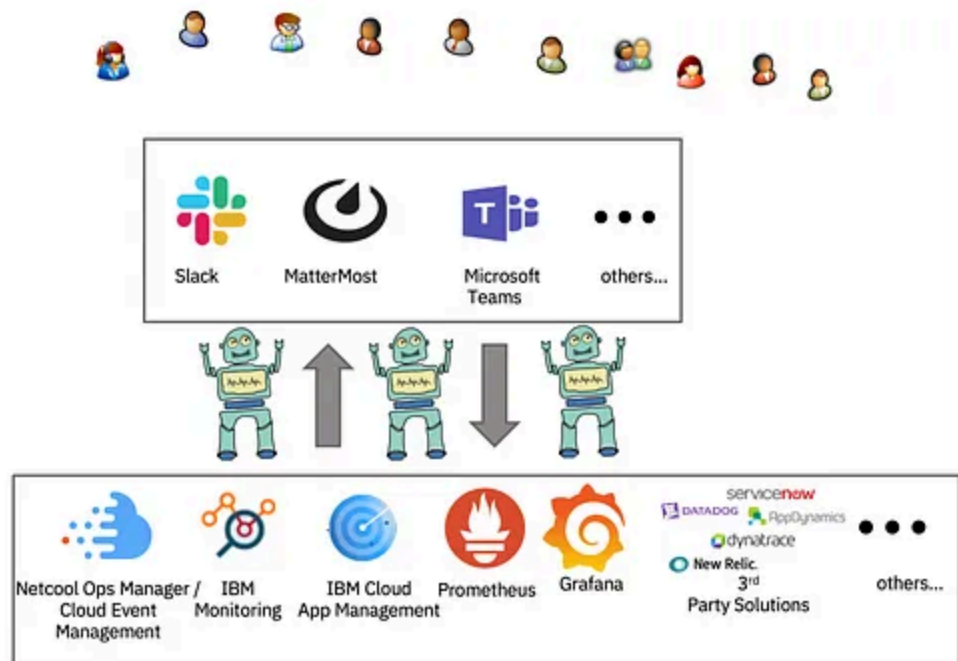
By having bi-directional integration, responses are even faster and more efficient.

Stage 4 — Bots added

Adding bots that participate in the conversation is a further level of automation. These bots might do things like copying key pieces of the conversation into

external tickets or audit trackers. This means that you no longer need a human to spend time on the manual tasks of documenting who-did-what-and-when. The human can spend time investigating or making decisions instead of toil.

The bot might listen for specific keywords and send a message to the human, interjecting with valuable information.



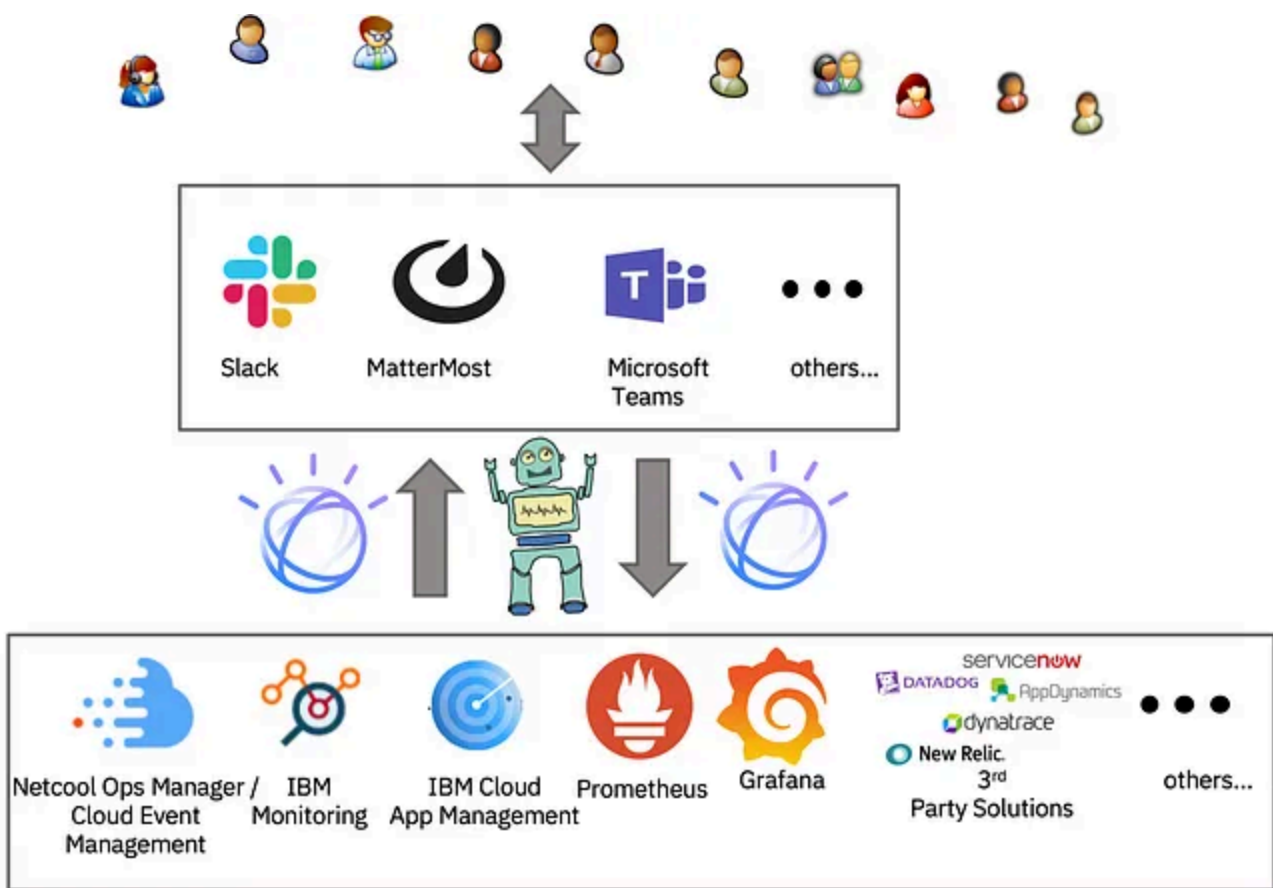
Stage 4 — Bots added

In short, these bots are not simply activated by people asking questions or entering commands but are actively listening in on the conversation and adding their own “opinions” when they see fit.

Stage 5 — Intelligent bots added

While waiting for specific keywords and acting on them is a clever stage 4 trick, the bots in stage 5 are intelligent. They’ll recognize a specific pattern in the conversation and realize that the humans are investigating a dead-end. They might suggest bringing specific experts in to help because they think the problem is taking longer than usual, or they might tell you that someone else is investigating a similar problem *right now* — and neither of you knows about the other.

A level 5 bot is a full participant in the war room conversation.



Stage 5 — Intelligent bots added

Reaching stage 5 integration with intelligent bots is not a goal in and of itself. If your team is made up of experts, with experience in the services they support, they may not need the extra helping hand that the cognitive team member brings — they already know what information they want and how to get it. But if you have someone new on the team, or you're on-boarding a new service, this cognitive helper might be the difference between solving the problem in minutes or hours.