

Images haven't loaded yet. Please exit printing, wait for images to load, and try to print again.

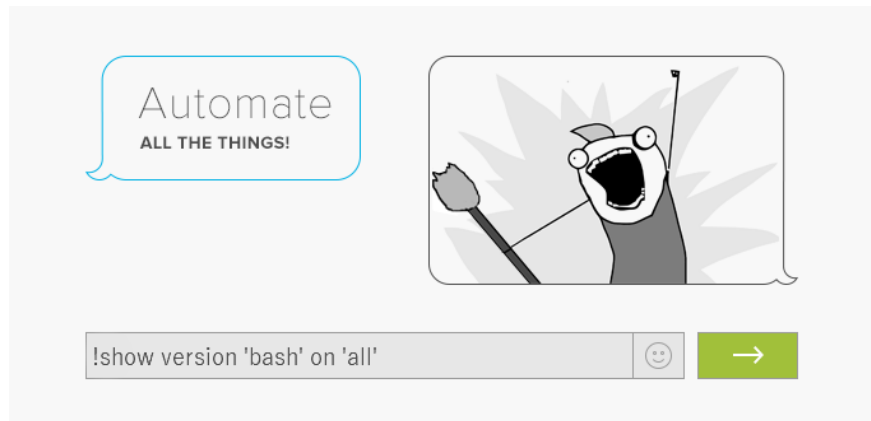
DevOps Better



Rania ZYANE

Follow

Mar 18, 2017 · 5 min read



When people hear DevOps, they often relate it to “automation”, “team work” and many “tools”. DevOps is all about CAMS: a culture of automation, measurement and sharing. The purpose of this article is to show how can ChatOps boost DevOps by bringing CAMS to everyday practice.

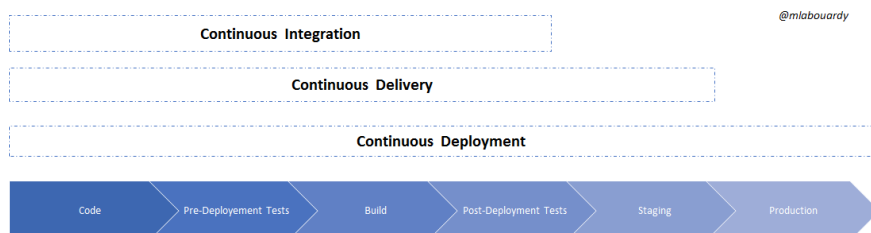
What is DevOps ?

DevOps is a set of practices that emphasize the collaboration and communication of both Software Engineers and IT & Infrastructure Operations to reduce the time to market of a Product. Devops mainly aim to deploy features into production quickly, detect and correct problems when they occur, without disrupting other services (Self-healing, Blue/Green deployment, Canary updates ...)

There are several guidelines in order to achieve DevOps maturity, here are a few you need to know:



- **Continuous Integration** is the practice of integrating, building and testing code within the development environment. It requires developers team to integrate code into a shared repository (Version Control System). Checks out the code and runs all the pre-deployment tests (Do not require code to be deployed to a server). Where applicable, next steps are to compile and package the code into Artifacts (JAR, Docker Image, gem ...), then to push it to an Artifact Repository Manager. The Artifact is deployed then inside an immutable container to a test environment (Quality Assurance). Once deployed post-deployment tests (Functional, integration & performance tests) are executed.
- **Continuous Delivery** is an extension of continuous integration pipeline. It aims to make any change to the system releasable and requires a person or business rule to decide when the final push to production should be occurred.
- **Continuous Deployment** is an advanced evolution of continuous delivery. It's the practice of deploying all the way into production without any human intervention.



In addition to those practices discussed above. Today, most DevOps teams embrace collaborative messaging platforms, such as Slack, to communicate with each other. And get real time updates about the system through online chat. And that's certainly the spirit behind ChatOps.

What is ChatOps ?

“Placing tools directly in the middle of the conversation”—*Jesse Newland, GitHub*

Collaboration and conversation are a power that let teams work and learn together to produce better. Today, this is happening in an exponential way that accelerates every year.

ChatOps (an amalgamation of chat and operations) is an emerging movement that led to ease the integration between teams and various tools/platforms of DevOps. It is about conversation driven development by bringing the tools into conversations. Robots are today members of your team to whom you can send a request and get an instant response.

ChatOps is a model where people, tools, process and automation are connected in a transparent flow. It also helps collaborate and control pipelines in one window.

Today, DevOps toolchain puts in many tools for development software, networks and servers management, tests, monitoring, etc. Collaborating and controlling DevOps pipelines in one window has helped developers teams work in more efficient and agile way.

Three main components in ChatOps :

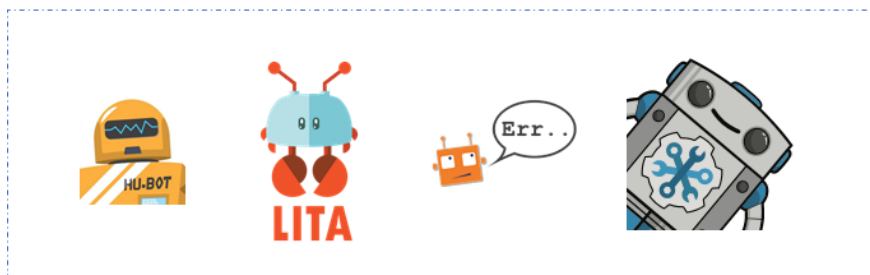
Collaboration tool: It's the Chat Client where stakeholders and teams are connected between them and to the systems they work on. There are several chat platforms :

- **Slack** : a leading chat platform for teams which accumulated more than 4 million daily active users. It is also one of the first platforms that integrated bots into its system.
- **HipChat** by Atlassian is a group chat, file sharing, video chat & screen sharing built for teams & business.

Bot: It is the core of chatops methodology. The Bot comes in the middle of the collaboration tool and the Devops tools. The Bot receives requests from team members, then retrieves information from integrated systems by executing set of commands (scripts).

- **Hubot**, a leading bot tool for ChatOps. It is a valuable open source robot (CoffeeScript) for automating chatrooms made back in 2013 by [Github](#). Hubot is useful and powerful via Scripts. They define the skills of your Hubot instance. Hundreds of them are written and maintained by the community. It mainly helps to automate most of ops-related tasks
- **Lita** is a framework for bots dedicated to company chat rooms written in Ruby. It is heavily inspired from Hubot. This framework can be used to build operational task automations and has a very comprehensive list of plugins which means that it can be integrated to many chat platforms as : Slack, Facebook Messenger and others.
- **Cog**, made by Operable, is another chatbot framework to help automate DevOps workflows. It's designed to be chat platform and language agnostic, and uses a Unix-style pipeline to activate complex functionality.
- **ErrBot** is a chatbot daemon that generates bots that are in the middle of a chat platform and DevOps tools. It is written in Python and aims to make integrating and tool that provides an API, easily to a chat platform via commands.

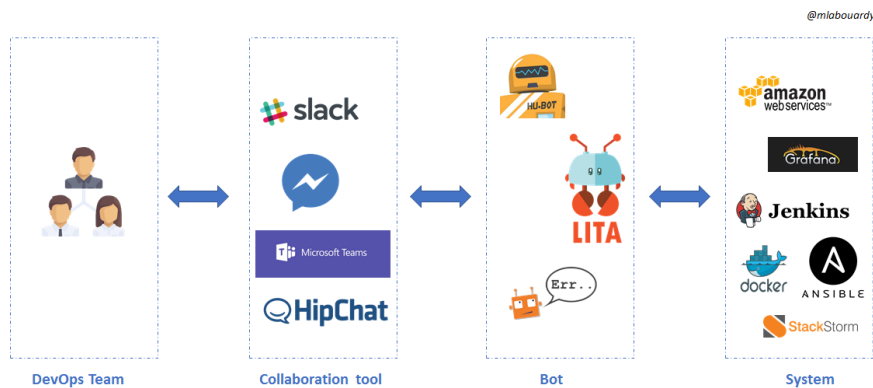
@Raniazy



System integration: Third key element in ChatOps. Simply put, it is the DevOps tools allowing more productivity. Such as :

- **Issue tracking:** JIRA, OTRS, TeamForge ...
- **Version Control Systems:** Github, Gitlab, Bitbucket ...

- **Infrastructure as Code (IaC):** Terraform, Vagrant, Packer, Swarm, Kubernetes, Docker, AWS CloudFormation ...
- **Configuration Management tools (Provisioning):** Ansible, Salt, Chef, Puppet ...
- **Continuous Integration Servers:** Jenkins, Travis CI, Bamboo ...
- **Monitoring:** Grafana, Kibana, Prometheus ...



Nowadays ChatOps are operational. Several teams around the world already connected their chat platforms to their build systems to get notifications, query and execute processes on their continuous integration servers. Same thing happens in QA (Quality Assurance) teams, support teams and the rest.

With ChatOps, trust is build among the team especially that work is shared and brought into the foreground by putting it all in one place. Your chat platform is your new command line.

Conversation-driven collaboration is not new, but with ChatOps, the oldest form of collaboration is harmonized with the newest techies. We are not surprised that this combination have changed the way staff members work. This should make people often think of making software that allows this collaboration to be more contributory, soft and secure.

For further Information on DevOps and ChatOps approaches, check out our [DevOps Wiki](#) on Github.

Thanks for reading,

With ♥ by [Rania Zyane](#) and [Mohamed Labouardy](#).

