

SEARCH

[News](#) [Data & Insight](#) [Sectors](#) [Topics](#) [Tech M&A](#) [Careers](#) [Jobs](#)
[Regions](#)

IPOs

M&A

PE & VC

The most important DevOps tools for tech workers

by **Antony Savvas**

9 August 2022

There are so many DevOps tools out there to help you deliver, it can feel overwhelming. Github or Gitlab? Openshift or Kubernetes? Antony Savas talks to DevOps experts about their favourites

DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development lifecycle and provide continuous delivery with high quality software. *Information Age* looks at the most important tools for tech workers.

RELATED STORIES



SPONSC

**Find**
Info

Most important DevOps tools for tech workers

“It’s important to remember that DevOps is a culture, and getting it right is more about processes and systems, over specific tools,” says David Huntley, continuous improvement director at workforce platform **Distributed**. “With that in mind, there are three things that are very important to nurturing an effective DevOps culture.”

First, he says, is the infrastructure, and thus the ability to build stable and repeatable environments with code. Second, is the Linux command line and the ability to execute commands in the build system, he maintains. Finally, proper instrumentation and visualisation of data from the production side are crucial, “so that everyone knows not just what’s happening in development, but in production too,” says Huntley.

‘There are too many tools which essentially do the same job as each other’

Chris Astley, head of connected engineering, KPMG UK

BizDevOps x metadata

Ales Zeman, senior manager for pre-sales and professional services for EMEA at **Quest Software**, says: “Across organisations we’re increasingly seeing teams that were once separated by organisational structures, budgets and mandates, being brought together under the cultural umbrella of BizDevOps.

“This helps ensure that the organisation is solving actual problems, not creating software that doesn’t align with business processes – or worse,

Establishing a strong DevOps pipeline



The most vital roles to fill in DevOps



DevOps vs Agile: distinguishing and combining the two

creating solutions for problems that don't exist.”

Zeman says one of the biggest challenges in undertaking the shift to BizDevOps is finding common terminology and understanding among the teams and bridging the divide between business stakeholders and developer teams.

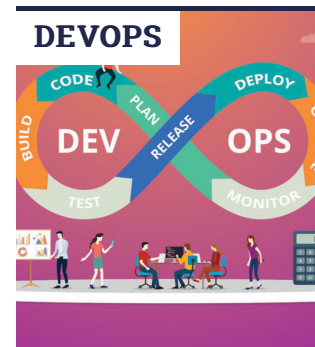
A business process modelling tool can detail the tasks, responsible parties, information elements involved in processes, and the interactions which can occur across systems, procedures and organisational hierarchies. All this information is stored in a shared, central repository, giving business process professionals, enterprise architects, IT teams and developers greater understanding of how software changes and deployments affect the business.

Another part of forming the common language among BizDevOps participants revolves around metadata – or the information the enterprise has about its data elements, says Zeman. Confusion would rein if there were different assumptions and conflicting definitions.

Metadata management tools help harvest, store and administer the data that describes other data. These tools also track how data elements are related and explain their lineage and, by centralising metadata in a data catalogue, both business and technical users can see how data flows through their business processes and they can refer to that data with “consistency and confidence”, Zeman says.

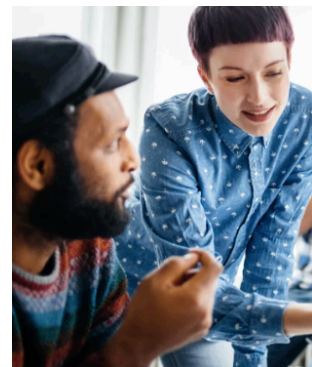
Security

Ryan Sheldrake, field CTO for EMEA at cloud security platform [Lacework](#), and ambassador for the [DevOps Institute](#), says that ensuring



The ultimate guide to DevOps

SPONSORED CONTENT



Find all the best Information Age

By Jol

DevOps teams can work efficiently without sacrificing security is fundamental for safe innovation in cloud environments.

“Automated cloud security tools help developers to identify mis-configurations inherent in code before they have even committed, rather than at build time, or worse, in production,” says Sheldrake.

“Not only does this reduce the burden of manual work, but it reduces friction between developers and stretched security teams. It’s easier, faster, safer and cheaper to fix earlier in the DevOps pipeline.”

Low code

Lila Dorato, senior director of solution engineering at **MuleSoft**, says low-code and no-code solutions are among the most important components of modern DevOps toolchains.

“With ‘drag and drop’ approaches, these tools enable developers to easily create new digital solutions, integrations and automations by reusing existing IT capabilities,” says Dorato. “The composable enterprise strategy that emerges from this eliminates the need for developers to write every line of code or start each new project from scratch, which increases the agility that DevOps brings.”

She says developers can also use low- and no-code solutions to implement RPA (robotic process automation) bots that intelligently process data from any system or legacy user interface to streamline DevOps workflows.

For example, they could use this capability to automate time-consuming, manual or static tasks in the integration, development, testing,

deployment and monitoring phases. By doing this, they can reduce “manual toil”, enabling them to release new code faster, says Dorato.

Too many DevOps tools?

Chris Astley, head of connected engineering at KPMG UK, says the market has become “flooded” with DevOps tools. “The market has got to the point where which tools are the most important is the less critical problem to solve.”

He says: “In my experience, the issue is that there are too many tools which essentially do the same job as each other. In a DevOps culture, the tooling is all geared around collaboration and these tools lose value when everyone isn’t using them.

“To give a simple example of source control – GitHub vs GitLab is a common conversation between technologists. Inevitably everyone will have an opinion. However, it’s much more important that the business chooses only one tool and uses it, and not waste time arguing over which is the best.”

Multiple DevOps tools

Arun Chandrasekaran, product owner for APM solutions at **ManageEngine**, says the continuous integration and continuous deployment (CI/CD) cycle is the indispensable part of the software development process.

In this process, code is typically built, tested, deployed, observed and measured. When issues are encountered, the code is modified and the process begins all over again. He says that when you have a high functioning CI/CD pipeline that comprises of multiple steps, one tool might not be able to provide holistic and continuous insight

into the entire cycle.

Chandrasekaran says: “That is why organisations often opt for multiple tools. Now, there are two approaches to this.”

In the first approach, he says, organisations can use cloud-native CI/CD pipeline tools that work with software deployed in containers and which are managed by container orchestrators like **Docker Swarm**, **OpenShift** or **Kubernetes**.

The second approach involves choosing cloud-based CI/CD pipeline tools that are meant for applications deployed in cloud infrastructures.

“**Jenkins** and **GitHub** are the two tools that we use for review and code management in the cloud,” Chandrasekaran says.

“We use our in-house tool Applications Manager for application performance management. Apart from streamlining workflows and increasing efficiency across departments, organisations should ensure that their tools provide analytical functions that are enhanced when teams implement observable CI/CD pipelines,” adds Chandrasekaran.

Open source

Open source tech is being deployed to build solid DevOps pipelines, but what tools are required to get your next modern data stack out of the door, on time and on budget?

DoubleCloud CTO Vladimir Borodin says: “To get reliable and efficient DevOps tooling you have only two options: Buy a ready-to-go solution – expensive today – or build it yourself with best-in class open source technologies.

“In our experience of building observability solutions for 1GB of inserted data per second, building it with technologies like **Kafka** and **ClickHouse** is the most efficient option. We have heard similar things from our customers and partners.”

Related:

The ultimate guide to DevOps: everything an enterprise needs to know – *A strong DevOps strategy can help to break down the silos between business and IT, necessary to maintain productivity, efficiency and security in this new age of remote working, which is why Information Age has produced this DevOps guide*

DevOps vs Agile: distinguishing and combining the two – *Agile development means fast, practical delivery of software while DevOps handles its practical rollout to users. Put together, they're both sides of the same coin*

Establishing a strong DevOps pipeline – *Having a strong DevOps pipeline is increasingly important for business creating software inhouse. What can CTOs do to ensure a steady flow?*

The most vital roles to fill in DevOps – *What are the vital roles when it comes to DevOps? With a growing IT skills shortage, Antony Savvas considers the most important jobs in the sector*

Related Topics

DevOps DevOps Engineers

Privacy