Eliminate the gap between development and operations

How enabling developers to manage their own applications landscape infrastructure improves time to market.

codesphere

Software that runs on my laptop but not somewhere else provides little value

To comprehend the principles underlying Codesphere, it is essential to grasp the process of software development and the subsequent making applications available to users. This journey encompasses two phases: "development" and "deployment." In the pre-cloud era, developers managed both these aspects.

The cloud paradigm has shifted from dedicated servers to virtualization, transforming the deployment front-end from a simple FTP server to a containerization software stack, including technologies like Docker, Kubernetes, and Helm Scripts. This shift implies that software developers no longer hold complete end-to-end domain expertise over the software development process, needing support of specialized engineers for deployment.

Codesphere was established, to reinstate this comprehensive authority back to the developer by making deployments simple enough to handle without additional domain expertise.

Are DevOps teams a best practice or merely a case of “deceptive labeling”?

The ongoing debate in the software industry revolves around these questions: Should developers acquire skills in Docker and Kubernetes to reclaim authority and expedite their software's go-to-market strategy? Alternatively, should companies merge development teams with operational resources specializing in High-Performance Computing and Deployment, responsible for establishing automated CI/CD pipelines?

Many experts advocate for software developers to learn Docker and Kubernetes, particularly in the context of Microservices and Cloud Computing. With the growing reliance on cloud technologies, Docker and Kubernetes have become increasingly relevant, addressing key challenges in application deployment and scaling. These skills are not only in high demand but also command competitive compensation.

Alternatively numerous companies have recognized the improved efficiency of integrating operational resources into development teams, rather than expecting developers to manage operational tasks. This integration often leads to the formation of DevOps teams, offering several supposed advantages:

- **Enhanced Collaboration and Efficiency:** DevOps teams bridge the gap between development and operations, fostering improved collaboration, communication, and efficiency, leading to quicker problem-solving.
- **Improved Deployment and Scalability:** With expertise in Docker and Kubernetes, DevOps teams enhance deployment practices, effectively managing containerized applications for smooth, scalable, and reliable processes.
- **Increased Agility and Flexibility:** DevOps teams are adept at responding swiftly to market changes or new business requirements. Their agile approach, encompassing continuous integration and deployment (CI/CD), allows for rapid adaptation to business needs.

However, the question arises: Are these DevOps teams really best practice or merely a case of “deceptive labeling”? Establishing an automated CI/CD pipeline involves selecting appropriate tools, scripting automated workflows, integrating rigorous testing protocols, and meticulous planning for a streamlined workflow from code commit to deployment. The skill set required is advanced and quite different to that of regular development work. So while some improvements can be achieved by letting dev and ops work closely together in functional teams, individual developers are still not able to handle infrastructure in a self-service manner but need the help of their more experienced ops colleagues. You may end up just insourcing the bottleneck from a separate team into a shared team.

Enable developers to manage their own applications infrastructure with Codesphere

Codesphere is a self-service platform designed for developers, incorporating many of the best practices mentioned above. It simplifies the creation of virtual machines instantly using zero-config principles.
We set out to make the entire deployment orchestration easy enough so that no specialized operations knowledge is required. Individual developers are empowered to deploy complex application landscapes (those typically requiring Docker & Kubernetes) without the burden of learning how to set up, maintain and operate such systems:

- Deploy as if locally, from a terminal, without the need to learn Kubernetes, Docker, Serverless, or Buildpacks.
- Enable provisioning (production) workspaces from the UI, handle routing via domains - staging, progressive releases, or A/B testing
- Seamlessly integrates with all large Git providers (GitHub, GitLab and Bitbucket) with a strong focus on efficient developer experiences
- Proprietary deployment algorithm that coldstarts resources in seconds - cost savings for low traffic apps and during autoscaling
- Hardware agnostic - clients can run on our data centers, on (private) cloud subscriptions & hyperscalers or on bare metal on-premise hardware.

Furthermore, Codesphere has evolved to address coding needs, transforming into an Integrated Development Environment (IDE) that can deploy complex landscapes. This means,

- that developers can develop, review, and improve code directly on the platform, eliminating the need to context switch between code editors and Git hosts.
- that it automates the installation of dependencies, restarts in case of build failures, and deploys managed services like databases for preview deployments.
- that Codesphere is highly customizable, supporting any application that runs on Linux and can be installed without root access.

Gartner stated in its 2023 Market Guide for Cloud Management tooling: “Upper software stack activity (for example, containers), beyond just infrastructure as a service (IaaS), is causing a convergence of traditional cloud management tooling and newer container management offerings, as well as the addition of SaaS management capabilities.” And Codesphere is exactly that: An innovative combination and extension of state of the art container management software, an integrated development environment, a Platform as a Service offering that can be deployed on your cloud(s), too. The platform being technologically agnostic to avoid vendor lock-in effects.

Codesphere returns the end2end process authority to the development teams. And it allocates cloud resources smarter and faster, which leads to a higher density of virtual machines and reduces overprovisioning. In other words:

Bridging the deployment divide leads to quicker and more cost-efficient software innovations run in the Cloud.