



A DEVOPS RESET FOR A MULTI-CLOUD WORLD

IN THE WORLD OF DEVOPS, ONLY TWO THINGS REALLY MATTER:

- To release applications and updates as quickly as possible
- To have access to resources (processing, storage, networks) able to run those apps, as cost-effectively as possible

The measures used to achieve these twin goals should work together seamlessly – otherwise you're inefficient. But software development and IT operations have been on separate journeys, and it now takes layers of tools, specialized expertise, and lots of precious time to distribute and manage applications where they need to run. Organizations tolerate all this complexity (and the delays and costs that come with it) since they see no alternative. But what if there was an easier way?

Let's examine the two fundamental shifts driving software development and IT operations, and see why current processes are so cumbersome. Then we'll explore how to bridge the gap.

Shift #1: From Monolithic Applications to Micro Services

In the beginning, applications were monolithic, with all functions and code packaged together. That approach was manageable when apps were updated once or twice a year, but not when updates are expected multiple times each day. Finally someone realized that applications would be much easier to build, improve, and maintain if they were divided into smaller units called microservices – independent, loosely-coupled,

individually deployable services, each one addressing a specific function.

To support microservices, a technology stack has emerged, similar to steps on a ladder:

To make them portable, microservices are packaged in containers – a lightweight, standard way for applications to move between environments and run independently. Everything needed to run the microservice such as settings, system tools, libraries, and dependencies is packaged inside the container.

A single application could have hundreds of containers associated with it. How do you manage them? Along came Kubernetes (aka K8s), open-source software for deploying and orchestrating containers within a cluster of virtual machines. (Note the word cluster – the significance will be discussed momentarily.)

Since microservices are updated constantly, how do you make sure the correct version is running? Along came GitOps to provide version control for all updates to all those micro-services. Git repositories provide a single source of truth for the desired application state, as well as a record of every code change.



NETHOPPER DEVOPS RESET BRIEF

Finally, we need to set up continuous integration/continuous delivery (CI/CD) for rolling out application improvements and new features. Along came ArgoCD, a GitOps continuous delivery tool for Kubernetes that continuously monitors running applications, compares the current, live state against the target state specified in a Git repository, and resolves discrepancies.

Shift #2: From On-Prem to Multi-Cloud

Not long ago, IT departments were having hot debates about this strange new concept called the public cloud and whether it could possibly play a role in their operations. But once they discovered how easy it was to spin up resources, that opened the floodgates.

The first tentative steps typically involved hybrid clouds combining an on-site private cloud with spillover into a public cloud for workload spikes. These initial forays mushroomed to dozens or hundreds of clouds, in different locations and with different providers. It's a mind shift happening in enterprises globally – where they once had a small number of clouds, now their cloud operations are huge, growing, and boundless. Companies are also discovering that the cloud is not just an easier, more cost-effective way to run their applications, it unlocks new opportunities.

Now there are core clouds, edge clouds for hyper-local applications, regional clouds, clouds for high availability and resilience, and more.

Edge computing is another, where low latency is required for operations at the customer-facing frontend cloud while backend processing happens in the primary cloud. But this is just the beginning.

Continuous Delivery Is Now Way Too Complicated

Today an application may have dozens of microservices distributed across different clouds that need to work together – perhaps some in cloud 1, a few in cloud 2, others in cloud 40, etc. The scale gets huge very quickly; for example, if 100 microservices are being delivered to 100 clouds, there are 10,000 mutations that have to be configured and managed.

This is the new normal, and DevOps needs to adapt to it. Processes should mesh transparently, but connecting microservices across multiple clouds and cloud instances requires inter-cluster communication, which Kubernetes (and GitOps/ArgoCD on top of it) does not natively support.

“...APPLICATION MAY HAVE DOZENS OF MICROSERVICES DISTRIBUTED ACROSS DIFFERENT CLOUDS THAT NEED TO WORK TOGETHER – PERHAPS SOME IN CLOUD 1, A FEW IN CLOUD 2, OTHERS IN CLOUD 40, ETC. THE SCALE GETS HUGE VERY QUICKLY...IF 100 MICROSERVICES ARE BEING DELIVERED TO 100 CLOUDS, THERE ARE 10,000 MUTATIONS THAT HAVE TO BE CONFIGURED AND MANAGED.”

NETHOPPER DEVOPS RESET BRIEF

To make it work requires specialized skill sets, so organizations hire teams with expertise in continuous-delivery, multi-cloud, networking, security, and automation. The first step involves distributing K8s containers to clouds where the microservices are going to run. Some enterprises create their own distribution system, others choose to download, install, and manage third party software such as open-source ArgoCD.

The second step is more complex: securely connecting those microservices across different clouds and K8s clusters. This involves integrating complex service meshes, load balancers, API gateways, and network/firewall automation tools, all feeding into a centralized console for configuring and monitoring.

The result is inevitable: lots of extra costs, inefficiency, potential security risk, and delayed time to market with resulting risks to the business. Development teams update microservices daily but depend on CloudOps, NetOps, SecOps, and other flavors of Ops to deliver them to the different clouds and clusters where they need to run. Despite the best efforts of these different stakeholders, it can days or weeks to update cloud networks and security for each change.

Simplifying Multi-Cloud Operations with an Overlay Network

When hundreds of microservices are being distributed across more and more clouds and updated constantly, you need an alternative – one that streamlines current operations without sacrificing security or application performance. Let's set aside traditional methods and think about how it should be done.

- First, the solution should be easy to use, so DevOps people don't have to be proficient with the underlying technology or be dependent on NetOps, SecOps, and/or CloudOps people who are. Instead, just provide some basic information about the Git repository and the clouds where you want the microservices to run, and an overlay network handles the rest.
- It should be comprehensive – integrating ArgoCD, the service mesh, the K8s ingress gateway, Zero-Trust security, resilience / failover, load balancing, and multi-cloud networking – all those elements that normally require assembling and configuring individual tools.
- It should make multi-cloud Kubernetes application operations easy to manage, with a single dashboard where you can securely deploy, connect, and monitor microservices in all your clouds and clusters.
- Finally, the solution should be available as a service, rather than having to install an ArgoCD server and try to manage it yourself (or hiring a team to support it). And if you have questions or need help, competent people should be available to give you a hand.

Disruptive – in a Good Way

Get ready for self-service multi-cloud Kubernetes, decoupled from the underlying networking and security details. It's an approach that's about to upend how microservices are deployed and managed, so DevOps can get back to what really matters: delivering apps as fast as possible, on the resources where they need to run. The ability to deliver new features in a fraction of previous times while significantly reducing costs and without compromising security will mean a competitive advantage and stronger bottom line.

NETHOPPER DEVOPS RESET BRIEF



Nethopper.io is a software company leading the evolution of the cloud stack from single to multi-cloud. We enable application teams to deploy, operate, and connect their containers seamlessly and securely across clouds. Nethopper has deep roots in Boston-area network innovations from Digital Equipment Corp, Cascade Communications, Alcatel, Ciena, Web.com, RedHat, Ericsson, and others.

Visit <https://www.nethopper.io> or email multicloud@nethopper.io.