direktiv

direktiv.io

5 reasons engineers are shifting to cloud native

Table of contents

Introduction 3

One

Cloud native is made highly scalable with microservices 4

Two

Engineers Save Valuable Time with Kubernetes 5

Three

Cloud native apps lead to automation opportunities 6

Four

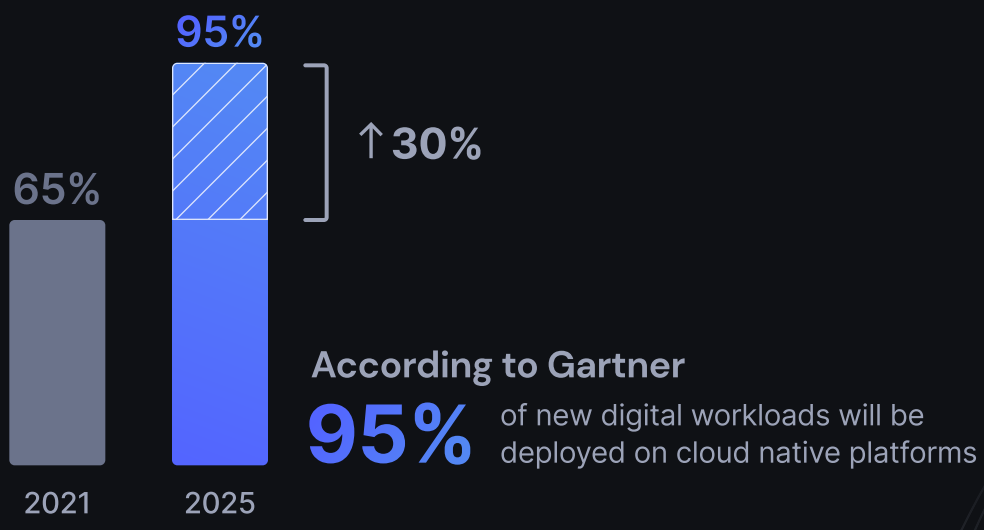
Avoiding vendor lock-in is easier than ever 8

Five

Enhanced ease of use simplifies engineers' workloads 9

Final thoughts 10

Cloud native platforms have grown leaps and bounds over the past few years — with no signs of slowing down. According to Gartner, by 2025, over 95% of new digital workloads will be deployed on cloud native platforms, increasing by 30% from 2021. Cloud native computing offers an easier approach to designing, constructing, and operating workflows that thrive within the cloud structure. These cloud native solutions take advantage of the scalability, elasticity, resiliency, and flexibility of the cloud, making cloud native an absolute dream for engineers. When creating and maintaining your workflows is so easy, it's no wonder engineers are flocking to cloud native solutions! Let's dive into just a few of the reasons engineers are shifting to cloud native.



One

Cloud native is made highly scalable with microservices

Most cloud native operations use microservices — an operational structure that develops functionality through a series of tiny services, often packaged in containers. Each microservice runs in its own isolated process, which is independently scaled and managed. These services loosely couple with other contained microservices to create a fully functioning application. Plenty of famous, large-scale enterprises use this type of computing (hi, Netflix!). With microservices, it's easy for engineers to scale apps with simple, packaged functions that can be used over and over to build apps rapidly. Cloud-native microservices even offer auto-scaling, meaning that as demand for an app increases, the app can be scaled by creating more specific microservices to take on additional functions. All happens in a matter of seconds.

For instance, let's say you're building out functionality for a ride-sharing app. The different app functions (mapping a route, selecting a location, typing in the search bar, etc.) all exist as separate microservices. You launch the app, and it begins growing rapidly, which indicates that you'll need additional functionality for ride-calling and driver tracking. Instead of spending weeks to months developing further functionality, your cloud-native microservices automatically create more unique, specific microservices to accommodate demand — allowing your functionality to grow with the app.

Two

Engineers save valuable time with Kubernetes

One of the greatest benefits of cloud native computing lies in Kubernetes, a container orchestrator that enables engineers and DevOps teams to deliver their product to market faster. As a platform, K8s automate the deployment, scaling, updating, and networking of various containers. Once containers are set up within Kubernetes orchestration, applications run with low downtime, great performance, and little support intervention. Just think of all the maintenance time you save with Kubernetes that you could spend building new, exciting code.

For example, let's say you begin using Kubernetes for container orchestration. Kubernetes regularly checks and ensures your entire container system is working properly, alerting you of any potential failures and fixing them before a break occurs. Where you used to spend hours upon hours manually fixing any node or pod failures within your container system, you can now start focusing on your more pressing tasks, like creating functionality for a new product your company is launching. And just like that, you never have to worry about the time-consuming tasks associated with container orchestration again!

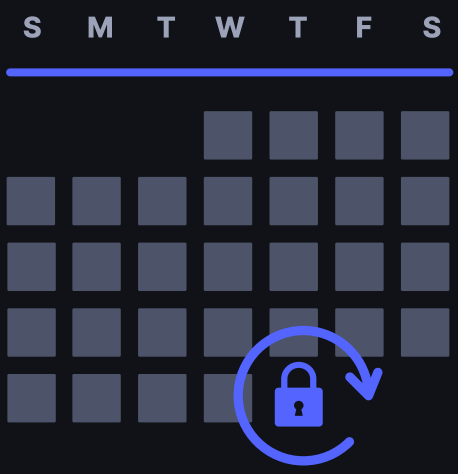
Three

Cloud native apps lead to automation opportunities

With the flexibility of containers and Kubernetes, creating automated, event-driven workflows using cloud native technology has become incredibly easy. You can build your workflow quickly and easily using the microservices and containers you need, then sit back and watch the workflow run effortlessly. This means that you're able to automate some of your most time-consuming and menial tasks, like password rotation (yeah, we just rolled our eyes too). Utilizing event-driven workflows within cloud native structures allows you to automate tasks and boost your efficiency. This enables you to focus more on your exciting tasks (like writing new code) instead of the menial tasks that cause your biggest headaches.

Consider this: instead of spending hours upon hours manually updating passwords every month... you construct a cloud native workflow that takes care of this task in seconds. By combining different microservices, you're able to create a workflow that generates new passwords, stores the new passwords in the correct location, and completes the entire rotation process in seconds, not hours. You set this workflow to trigger once every 30 days, so that you know your passwords are rotated consistently. Now, instead of eating up a full day of work, password rotation occurs automatically on your schedule.

Consider this: instead of spending hours upon hours manually updating passwords every month... you construct a cloud native workflow that takes care of this task in seconds. By combining different microservices, you're able to create a workflow that generates new passwords, stores the new passwords in the correct location, and completes the entire rotation process in seconds, not hours. You set this workflow to trigger once every 30 days, so that you know your passwords are rotated consistently. Now, instead of eating up a full day of work, password rotation occurs automatically on your schedule.



Set the workflow to trigger once every

30 days, so that you know your passwords are rotated consistently

Four

Avoiding vendor lock-in is easier than ever

We know what you're thinking... yeah, using containers provides all of these amazing advantages until you're locked in to a single vendor. Plenty of platforms require certain vendors for creating workflows using containers, which — we know — can be incredibly frustrating. Why would you want to opt for a platform that only allows one vendor, requiring you to reorganize your DevOps around the technology? Engineers have been flocking to cloud native, serverless solutions precisely because they let you avoid vendor lock-in. Serverless cloud native systems offer vendor-agnostic solutions, meaning you have the freedom to pull containers from multiple vendors, combine them, and even add in your own custom containers.

For instance, let's say you want to create a workflow using your own custom container, a Google container, and a few from GitHub. With the right serverless, cloud native platform, you're able to combine every container you need, no questions asked. You can finally work with the limitless possibilities offered by containers!

Five

Enhanced ease of use simplifies engineers' workloads

Aside from its flexibility, cloud native is just easy to use. And if you're anything like us, then anything that makes work easier is a cause for celebration. When you work within the cloud, it makes things like data backup, maintenance, development, and usage easier and quicker to manage. Since you aren't tasked with the chore of keeping up a server, your upkeep tasks — like data backup — become quicker and easier. Since cloud native servers are virtual, you can take care of these maintenance tasks with the simple click of a button.

If you're accustomed to using physical servers, you're likely familiar with the upkeep they require. Constant maintenance, updates, and check-ins are needed to ensure that the server is running smoothly and efficiently. Imagine, instead, that you're able to avoid all of that maintenance, and simply backup your data with the click of a button — no pre-checks to ensure the server works properly or that the backup will even run at all.

Final thoughts

Since working with cloud native computing allows you to save both money and time, it's no wonder that engineers are shifting to cloud native solutions. Say goodbye to server upkeep and vendor lock-in, and say hello to easy-to-use containers and microservices that build (and scale) your workflows quickly.

See for yourself why cloud native is growing in popularity and explore the possibilities of serverless, container-based workflows with Direktiv.

direktiv

Scaling the enterprise via event-driven orchestration

Contact us

Start for free

direktiv.io

Check our digest