weaveworks

Going Cloud Native: 6 essential things you need to know

WHITEPAPER

01. A BRIEF HISTORY OF CLOUD NATIVE

Depending on who you ask, cloud native can mean a lot of different things. Ten years ago it was coined by companies like Netflix who leveraged cloud technologies, going from a mail order company to one of the world's largest consumer on-demand content delivery networks. **Netflix pioneered** what we've come to call cloud native, reinventing, transforming and scaling how we all want to be doing software development.

With the phenomenal success of Netflix and their ability to deliver more features faster to their customers, companies want to know how Netflix used cloud native technology to gain such a huge competitive advantage.

At its core, the term cloud native is a way to increase the velocity of your business and a method to structure your teams to take advantage of the automation and scalability that cloud native technologies like Kubernetes offer.

02. BENEFITS FOR ENTERPRISES ADOPTING A CLOUD NATIVE STACK

The top benefits for enterprises adopting cloud native can be summarized as follows:

Increased Agility and Productivity

With GitOps and DevOps best practices, developers use fully automated continuous integration and **continuous delivery pipelines** (CICD) to rapidly test and push new code to production. Enterprises can bring new ideas to production within minutes or hours instead of weeks and months, resulting in a greater rate of innovation and competitiveness.

Improved Scalability and Reliability

On-demand elastic scaling or cloud bursting offers near limitless scaling of compute, storage and other resources. Enterprises can take advantage of built-in scalability to match any demand profile without the need for extra infrastructure planning or provisioning.

GitOps and DevOps best practices provide developers with a low risk method of reverting changes, clearing the way for innovation. With the ability to cleanly rollback, recovery from disaster in the case of a cluster meltdown is also faster. Higher uptime guarantees means businesses are more competitive and can offer more stringent service level agreements and a better quality of service.

Lower Costs

Because cloud native technology enables pay-per-use models, the economies of scale is passed through and shifts spending from CAPEX to OPEX. This lower barrier to entry for upfront CAPEX spending allows for more IT resources on development rather than on infrastructure. In addition, overall TCO/hosting costs will also be lower.

Attract and retain top talent

Working with cloud native and other cutting edge open source technology that lets you move faster and spend less time on infrastructure is appealing to developers. Hiring higher quality developers results in better products, and therefore more innovation for your business. An added bonus is that open source contributions can help establish your reputation as a technology leader.

Reduced vendor lock-in

Cloud native gives you a choice of tools without being stuck with legacy offerings. By taking advantage of multi-cloud compatible tooling wherever possible, your applications are more portable and beyond the reach of vendor predatory pricing. You can easily migrate to alternate public clouds with better product offerings or where compliance requires multi-cloud infrastructure.

03. CLOUD NATIVE ARCHITECTURE: WHAT DOES IT LOOK LIKE?

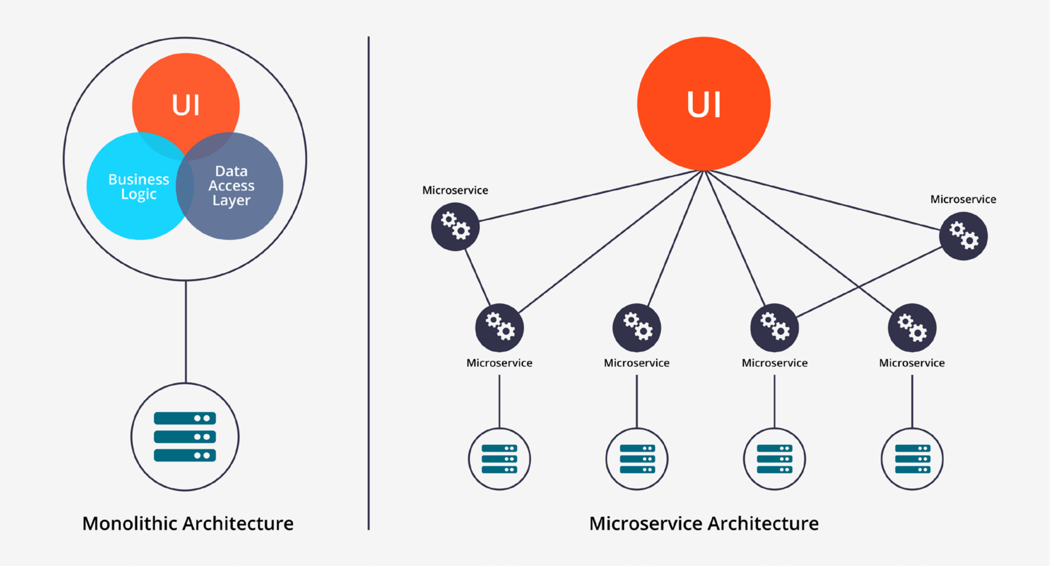
Monolith vs. microservices architectures

After **a disastrous release with a misplaced semicolon**, Adrian Cockcroft, former Netflix Cloud Architect shifted their entire architecture from a monolith to microservices.

The problem with a monolithic architecture is that when new features are developed and tested, it takes a considerable effort to deploy those changes to production:

- Multiple teams are needed to coordinate their code changes.
- Deploying several features all at once requires a lot of upfront integration and functional testing.
- Development teams are restricted to using one or two languages.

The shift to microservices empowered Netflix developers to deliver new features much faster to their customers.



Microservices results in a loosely coupled service-oriented architecture with bounded contexts. This means that if every service has to be updated simultaneously, it's not 'loosely coupled'; and along the same lines, if you have to know too much about the surrounding services then you don't have 'bounded contexts'. See also Martin Fowler and James Lewis original blog that discusses the definition: "**Microservices: a definition of this new architectural term**".

Microservices, Docker & Kubernetes

Docker containers lend themselves very well to microservices. By running your microservices in separate containers, they can all be deployed independently and even in different languages, if you wish. Containerization removes the risk of any friction or conflict between languages, libraries or frameworks. Because containers are portable and can operate in isolation from one another, it is very simple to create a microservices architecture with containers and move them to another environment if you need to.

Container orchestration

Once you have a large number of microservices all running in Docker containers, you need a way to manage or to orchestrate those containers so that they make sense as an application. This is where you need an orchestrator (cluster manager) like Kubernetes or Docker Swarm or others.

At one time in the past, you had to make an informed choice on which orchestrator to use, but now the orchestration wars are won and Google's Kubernetes came out on top. All of the major cloud providers support Kubernetes with easy-to-install solutions.

The gist of this discussion is that for most companies to be competitive, they have to architect their applications around microservices and run them in a Kubernetes cluster – although some companies do run Docker containers on other orchestrators as well. [link]

Deployment

With applications running in containers and orchestrated in Kubernetes, the next step is to automate deployments. A continuously automated flow of features is what distinguishes DevOps from other software development philosophies and practices like the waterfall model where development follows an orderly sequence of stages.

Continuous doesn't mean that engineers are working 24/7 updating code, or that they are deploying updates every time a line of code is changed. Continuous in this sense, refers to software changes and new features rolling out on a regular basis through an automated continuous integration and continuous deployment pipeline (CICD). More DevOps strategies for building CICD pipelines can be found in the **eBook: Building Continuous Delivery Pipelines.**

Monitoring

With containers and microservices, monitoring solutions have to manage more services and servers than ever before. Not only are there more objects to manage, but cloud native apps also generate a lot of extra data that needs to be kept track of.

Collecting data from an environment composed of so many moving parts is complex. Prometheus is the best modern solution for these dynamic cloud environments. It was built specifically to monitor applications and microservices running in containers at scale and is native to containerized environments. See the **whitepaper: Monitoring Cloud Native Applications**.

Cultural Changes

The success of implementing cloud native technology and DevOps best practices into your organization depends a lot on your existing company culture. Internal teams must not only learn to adopt cross-functional methods that ensure software is iterated on with a continuous cadence but that it also complements the business goals of the company. Making the actual switch to cloud native may be the simplest part in your journey; getting those changes to stick and propagating them throughout your organization could well be the most difficult part of the process.

04. CLOUD NATIVE IN PRACTICE

Invisible infrastructure means portability and speed

Most companies want to migrate applications to the cloud, but they may also want to keep some of their applications or their data behind a firewall and on premise. Some may want the ability to change cloud providers to take advantage of better pricing models or they may need to observe compliance regulations and span multiple cloud providers. For applications to be this easily portable, enterprises demand that their systems just work, so they can get back to building business value by releasing new applications and features instead of investing in infrastructure.

“We need to stop writing infrastructure... One day there will be cohorts of developers coming through that don’t write infrastructure code anymore. Just like probably not many of you build computers.” - Alexis Richardson, CEO, Weaveworks

Enterprises making the digital transformation and who need to move their business forward to be competitive are interested in creating what we call, an invisible infrastructure. To move faster, you need a way to simplify infrastructure changes so that developers can focus on innovation and on building new features without overhead. The ultimate goal is for developers to never write infrastructure code and instead focus on features. When infrastructure friction is reduced, businesses are more agile and competitive.

So, when we talk about applications being cloud native, what we’re fundamentally talking about is scalability, portability, and development speed.

Increased velocity means more agile businesses

From a business point of view, the cloud native pay off are apps that are always on, highly available that can be updated by your development team with zero downtime. Cloud native applications allow your development teams to address customer requests more or less as they come in, instead of waiting weeks. Increased velocity and agility are the main characteristics and benefits of this new breed of modern applications, architectures and practices.

Developers can practice continuous delivery

Companies who adopt cloud native have increased their mean time to deployment from 1 or 2 deployments a week to more than 150 deployments a day. If your site goes down, you can fix it in five minutes rather than keep your customers offline.

The ability to move fast is one of the main differences between those who change and update their applications continuously versus those who struggle to make small changes to their website. You can quantify continuous delivery, and that’s one reason people get so excited about unicorn companies such as Airbnb and Netflix, who have already figured this out.

Cloud Native is something that most companies are aware of and recognize is important. Of course the tricky part is democratizing and distributing that knowledge? How do we make this technology available to everybody, and not just to the elite technology companies in Silicon Valley?

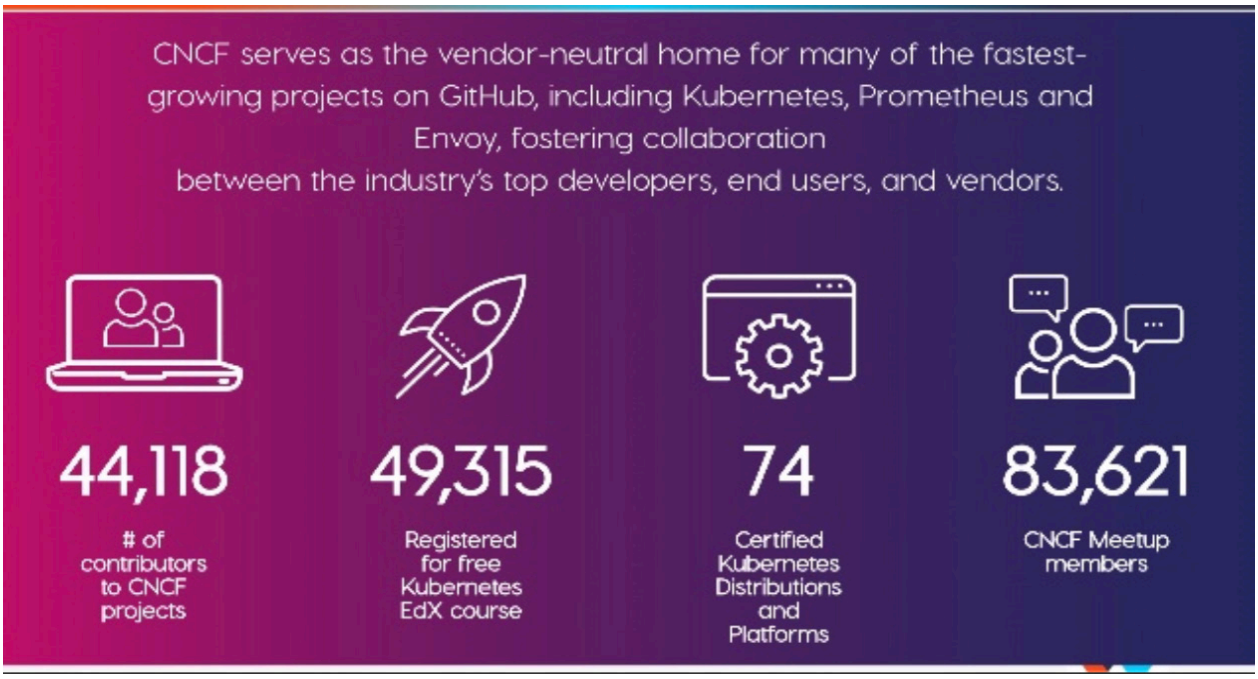
05. CNCF'S ROLE

The **Cloud Native Computing Foundation (CNCF)** was created four years ago and is the vendor-neutral home of Kubernetes -- an open source system for automating deployments as well as scaling and managing applications. Kubernetes was originally created by Google to run their search engine, but today it has contributions from Amazon, Microsoft and Cisco, as well as more than 300 other companies.

With Kubernetes, containers that make up an application are grouped into logical units for easy management and discovery. It scales with your app without you needing to add more resources to your Ops team.

In addition to this, automatic deployments as well as multiple simultaneous deployments can also be safely carried out. This new way of doing releases and product updates is for most people a very new concept.   And all of these ideas are part of what's been referred to as the cloud native revolution.

The main mission of the CNCF is to build sustainable ecosystems and community **around a constellation of high-quality projects** that support and manage containers for cloud native applications built on Kubernetes.



CNCF serves as the vendor-neutral home for many of the fastest-growing projects on GitHub, including Kubernetes, Prometheus and Envoy, fostering collaboration between the industry's top developers, end users, and vendors.

44,118 — # of contributors to CNCF projects

49,315 — Registered for free Kubernetes EdX course

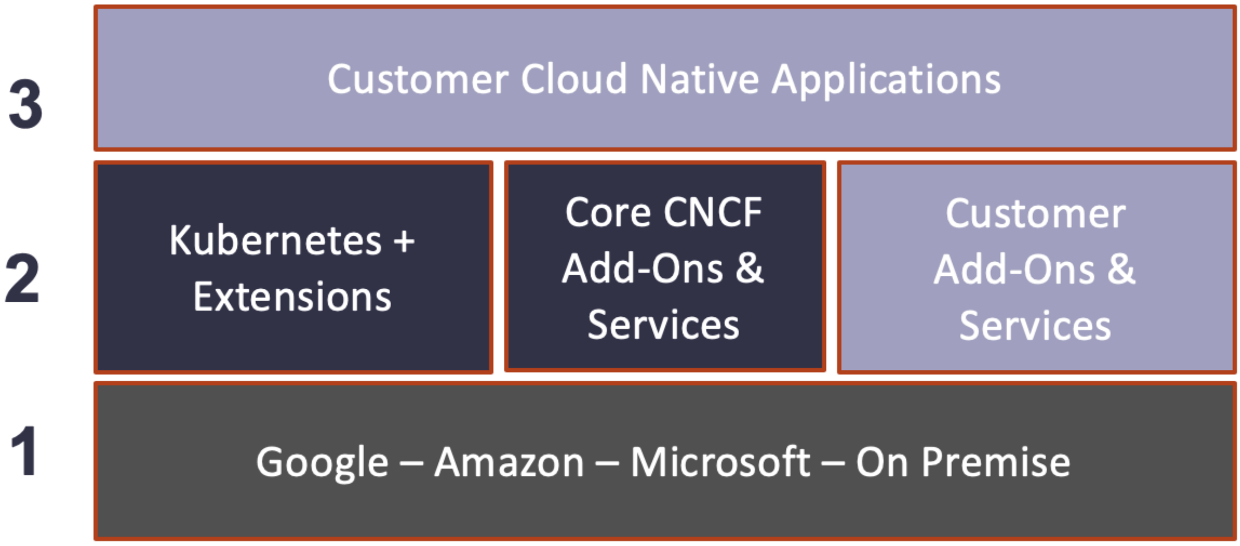
74 — Certified Kubernetes Distributions and Platforms

83,621 — CNCF Meetup members

In addition to hosting and supporting new cloud native projects, the CNCF provides training, a **Technical Oversight Board**, a Governing Board, a community infrastructure lab, and several certification programs.

A common cloud platform

In order to introduce digital solutions into different business environments, developers will need to stop worrying about the underlying infrastructure and focus instead on the applications and other features that add direct value to the bottom line. This leads us to an important goal of the CNCF which is building a common open cloud native platform and toolkit that enterprises can easily take and adapt within their organization.


3
Customer Cloud Native Applications
2
Kubernetes + Extensions
Core CNCF Add-Ons & Services
Customer Add-Ons & Services
1
Google – Amazon – Microsoft – On Premise

For this common platform to take shape we need the following:

1. Physical infrastructure that provides scalability and that also allows your app to run anywhere -- in a public cloud or on-premise or both.
2. A common cloud technology platform with a pluggable set of tools for this next generation of applications. A platform with pluggable tools that make it easy to run cloud native applications in the cloud.
3. The adoption and development of many modern cloud native architectures for new opportunities in data analysis, machine learning, finance, drones, cars, internet of things, medicine, communications and other business verticals.
4. Off the shelf cloud native components

By taking advantage of the many incubating projects available in the CNCF, you can now easily set up the infrastructure and lay the groundwork for your teams to innovate more. Before cloud native technology, adding a new business component to your monolithic platform meant hiring an army of consultants that took nine months to implement.

But now a lot of time is saved by using the **CNCF's landscape** of off the shelf, community supported components. This allows you to focus on the task at hand which could be introducing machine learning or other data science methods into your business to increase its innovation.

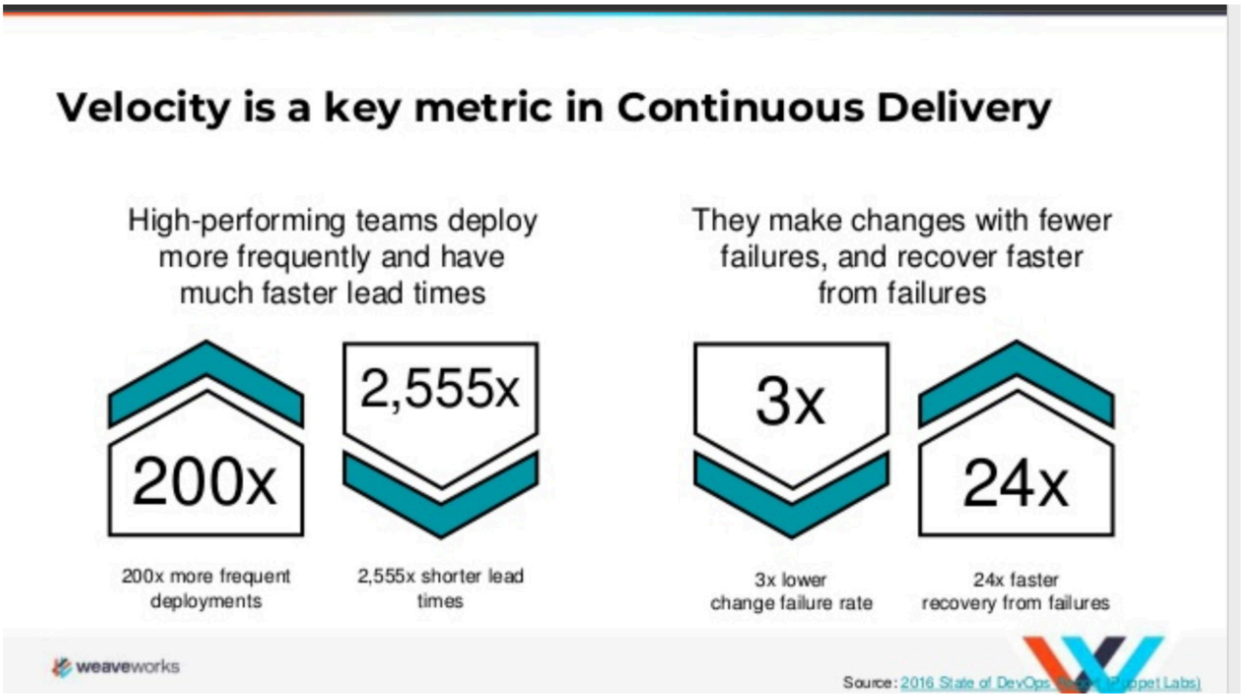
06. HOW CLOUD NATIVE RELATES TO DEVOPS

DevOps and Continuous delivery

Cloud native has led us to a whole new set of methods and philosophies on how to do software development, otherwise known as the cultural shift that is DevOps. With a new set of tools, teams will naturally figure out new and innovative ways of using them. This often occurs with new generations of developers who are coming with a fresh set of eyes and an untainted spin on an old set of problems. In particular, cloud native technologies have led to the implementation of new continuous delivery tools and methods that help you speed up development.

Velocity is the key to continuous delivery

Kubernetes platforms that provide continuous delivery components (among others) enable speed, and they also lower the barriers to entry. With continuous delivery in place, your team can deploy changes throughout the day, instead of quarterly or monthly. Continuous delivery also provides a mechanism to rollback changes whenever they need to. With a continuous delivery pipeline in place, development can make changes right from source code to production, but even more importantly, they also have the ability to revert and back out of the change just as easily.



Velocity is a key metric in Continuous Delivery

High-performing teams deploy more frequently and have much faster lead times

They make changes with fewer failures, and recover faster from failures

200x

200x more frequent deployments

2,555x

2,555x shorter lead times

3x

3x lower change failure rate

24x

24x faster recovery from failures

weaveworks

Source: 2016 State of DevOps Report (Puppet Labs)

The ability to continuously deploy changes means your team can deploy tests to subsets of customers or rollout specific customer requests more easily. And because a rollback is only a click away, developer can recover from failure much more quickly.

This is very different from software development 15 years ago when it took a monumental coordination effort to deploy a single change. With cloud native technology like Kubernetes, and other supporting projects, deploying changes on a continuous basis is trivial because this technology makes it possible.

Cloud native push code, not containers - GitOps

Cloud native lends itself well to GitOps style deployments. As an operating model for building Cloud Native applications, GitOps unifies Deployment, Monitoring and Management.

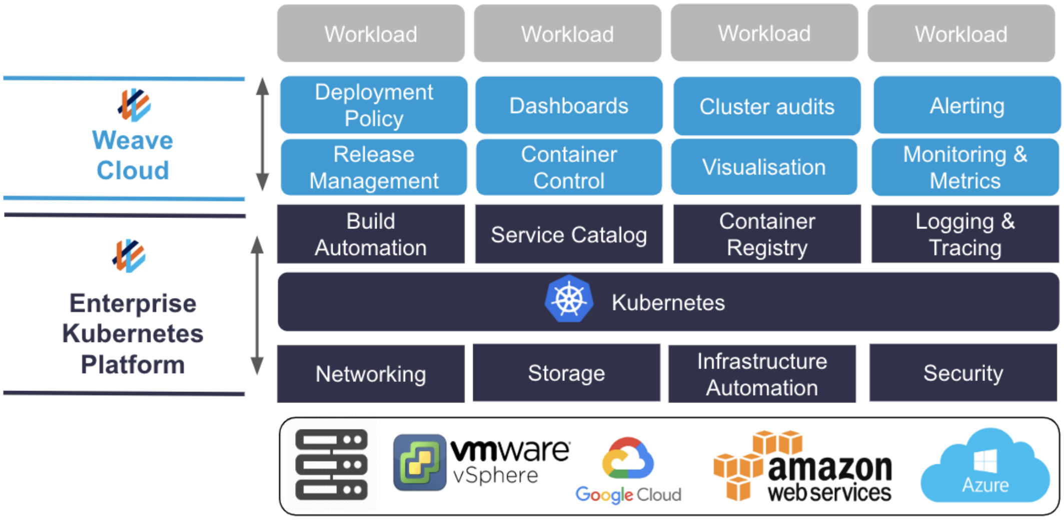
The goal of GitOps is to speed up development so that your team can make changes and updates safely and securely to complex applications running in Kubernetes. It does this using the tools and workflows that developers are familiar with. For more information see **GitOps - What you need to know.**

Weaveworks Enterprise Kubernetes Platform

Get up to speed quickly with the Weaveworks Enterprise Kubernetes Platform. Weave Kubernetes Manager combines the Kubernetes container management platform with Gitops and provides a flexible operating model for clusters, cluster components, and application workloads.

We provide an out-of-the-box production ready, full stack, experience that makes it simple to get into production with Kubernetes, and effectively manage changes:

- Single pane of glass across all your clusters
- A powerful set of tools for operations teams that allow them to manage clusters more efficiently.
- Production ready cluster with addons needed for secrets, CD, monitoring, etc.
- Professional training designed to reduce friction that impedes developer productivity



Contact us for more details or learn more about the **Weaveworks Enterprise Kubernetes Subscription**.