Guide to the cloud

How to navigate the multi-cloud future

GitLab

Organizations are under immense pressure to increase development speed but are often limited by the boundaries of infrastructure. Whether it's the number of servers, or a lack of knowledge, or outdated architectures, organizations can hit a productivity wall that prevents them keeping up with their competitors. The widespread adoption of cloud computing in response to these challenges is changing the way we deliver software.

Organizations don't have to rely on their own infrastructure or data centers anymore. Cloud computing is all about access to computing services on demand such as servers, storage, databases, networking, software, analytics, and intelligence over the cloud. From a cost perspective, organizations only pay for what they use, which should translate into more resources to fostering innovation and cost efficiency in the long term.

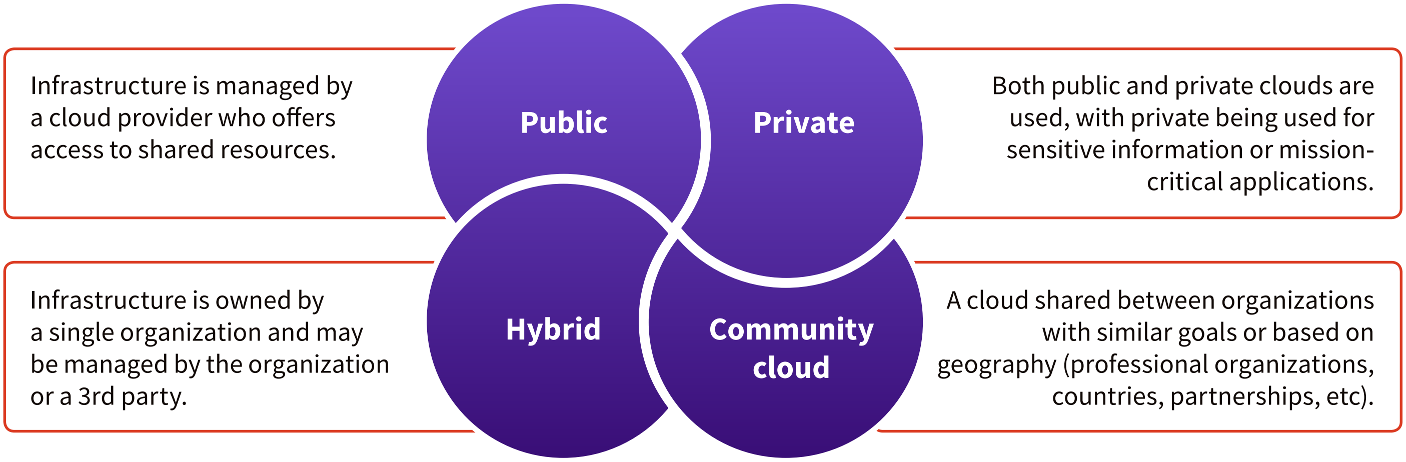
Even with the seemingly limitless possibilities of cloud computing, leaders still need to put these pieces together. A cohesive cloud strategy is how we can harness the power of the cloud, and that requires discipline as well as investments in solutions that fit with a long-term vision. Forrester has gone a step further by saying that cloud is no longer a tactical solution but rather a strategic enabler of connected economies. Businesses functioning at cloud speed and fluidity will dominate their markets. These new powerhouse "tech companies" will take aim at competitors with "focused" cloud, and choosing the right partners is vital in cloud success.

The cloud is only as efficient as we are. For the cloud of tomorrow, a cohesive multi-cloud approach is going to help teams optimize their cloud computing strategy for faster delivery and greater productivity.

The cloud of today

Cloud computing can be broken down into deployment models and service models. The deployment model refers to access and ownership of the environment. They describe where the software runs while service models describe *how* the cloud gets used. Most companies use multiple combinations of these models, sometimes with multiple vendors, and it's this flexibility that sometimes causes complexity in processes.

There are four main cloud deployment models.



Having covered the “where” of cloud models the next step is to look at the “how”. Services models describe how businesses use the cloud. Cloud service models have evolved over time, with new services popping up continually, but the most popular have traditionally been SaaS, IaaS, and PaaS.

Software-as-a-Service

SaaS is a software distribution and licensing model in which a third-party provider makes applications available to users through a subscription. It is the largest category of cloud computing and accounted for more than 50% of all public cloud spending in 2019. A majority of SaaS applications run directly through a web browser and don’t require any downloads or installations on the client side.

Infrastructure-as-a-Service

IaaS provides an instant computing infrastructure provisioned and managed over the internet and the second-largest category of public cloud spending with a five-year CAGR of 33.7%. While a cloud service provider provides the infrastructure, organizations manage the operating systems, middleware, and applications.

Platform-as-a-Service

PaaS is an on-demand environment for developing, testing, delivering, and managing software applications. PaaS is the third-largest category of cloud spend with 29.8% CAGR.

_______-as-a-Service

As businesses demonstrate needs, more services are becoming available to service those needs. Storage, Database, Information, Process, Application, Integration, Security, Management, Testing—all of these services are becoming widely available for organizations wanting to solve specific problems.

Cloud services come packaged as ready-to-use, but in order to use them effectively you need the proper application architecture.

Running a cloud architecture

The cloud has enabled new ways of designing software for immense resilience and scale. Taking a traditional application running in an on-prem server and simply doing a "lift and shift" to a VM on the cloud can give some benefits, but it doesn't truly take advantage of what's possible with on-demand cloud infrastructure and services. Designing an application with a cloud architecture in mind is cloud native. Cloud native architectures take advantage of containers, microservices, service meshes, and other technologies native to cloud environments.

It makes a lot of sense to containerize applications because it allows them to be bundled with their dependencies so that they can be operated independently on the platform of your choosing. Large organizations can have a variety of architectures for different use cases, such as serverless, monoliths, and microservices. Using containerization with these architecture choices allows organizations to have the flexibility they need to run applications anywhere.

With a monolithic architecture, all of the components are part of a single unit: Everything is developed, deployed, and scaled together. In comparison, microservices have each component broken out and deployed individually as services, and these services communicate with each other via API calls. For complex applications that need to run at scale, microservices can offer greater flexibility, reliability, and a faster pace of innovation than monoliths.

The oldest argument for monoliths has always been their simplicity: They're easy to build and easy to run. While it was once difficult to develop applications with a microservices architecture, over the past five years it has become considerably easier with container orchestration tools like Kubernetes, comprehensive CI/CD tools that automate testing and deployments, and APIs that update automatically. Organizations that adopt microservices get their simplicity through automated processes.

Enterprises turn to the cloud to deploy more quickly and frequently, and automation is where enterprises make the most of their cloud resources. A container orchestration tool like Kubernetes automates the deployment, scaling, and management of your containerized applications. You can respond quickly and efficiently to customer demand while limiting hardware usage and minimizing disruption to feature rollouts.

Cloud's big three: AWS, Microsoft Azure, and GCP

Gartner estimated IT spending to increase by 3.2% to $3.76 trillion in 2019 with as-a-service models fueling everything from data center spending to enterprise software. Organizations continue to go all-in on their cloud strategies with investments in infrastructure to support cloud computing estimated at more than a third of all IT spending.

Adoption of IaaS often begins with modern workloads, emphasizing developer productivity and business agility. But an increasing amount of cloud IaaS is being bought for traditional IT, with an emphasis on cost reduction, safety, and security. Infrastructure and operations (I&O) leaders typically lead the sourcing when cloud IaaS is to be used for traditional IT. By contrast, sourcing for modern workloads is typically driven by enterprise architects, application development leaders, and digital business leaders.

IaaS is just one part of cloud computing. While the overall cloud market is massive, the largest cloud providers – Amazon Web Services, Microsoft Azure, and Google Cloud Platform—deliver over 57% of all cloud services.

Service	AWS	Azure	GCP
Compute	Elastic Cloud Compute	Virtual MAchines	Compute Engine
App Hosting	Elastic Beanstalk	Cloud Services	App Engine
Serverless Computing	AWS Lambda	Azure Functions	Cloud Functions
Container Support	ECS/EKS Containers	AKS Container Service	Kubernetes Engine
File Storage	S3 Storage Service	Azure Storage	Cloud Storage
Block Storage	Elastic Block Storage	Azure Blob	Persistent Disc

Backup Options	AWS Glacier	Azure Backup	Cloud Storage
Data Orchestration	Data Pipeline	Data Factory	Cloud DataFlow
Data Management	AWS Redshift	SQL Data Warehouse	Google BigQuery
NoSQL Database	DynamoDB	Cosmos DB	Cloud DataStore

The expanding role of cloud service providers in software and app development.

AWS offers 90 different services, as does GCP. In comparison, Microsoft lists over 160 services on its Azure product page, many of them integrations with other Microsoft products. Google, for example, has products like G Suite, AdWords, Google Analytics, Chrome, and others that play a role in every facet of an organization. Serverless is one architecture pattern making a major impact. Serverless takes advantage of many cloud services to bypass in-house operations altogether.

The serverless uprising?

Serverless architecture refers to applications that scale on demand up to infinity and then down to zero when not in use. They depend on third-party services (Backend-as-a-Service or BaaS) or on custom code written as discrete functions that's run on ephemeral compute (Function-as-a-Service or FaaS), the biggest vendor being AWS with their Lambda FaaS. Serverless architecture has an annual growth rate of over 700% and shows no signs of slowing down. Its popularity is due to the operational efficiency and lower operating expense it promises. While all cloud services offer the ability to offload some amount of operational overhead, serverless completely outsources the provisioning and scaling of infrastructure to the cloud provider. Additionally, paying only for compute time while the application is in use makes the model cost effective.

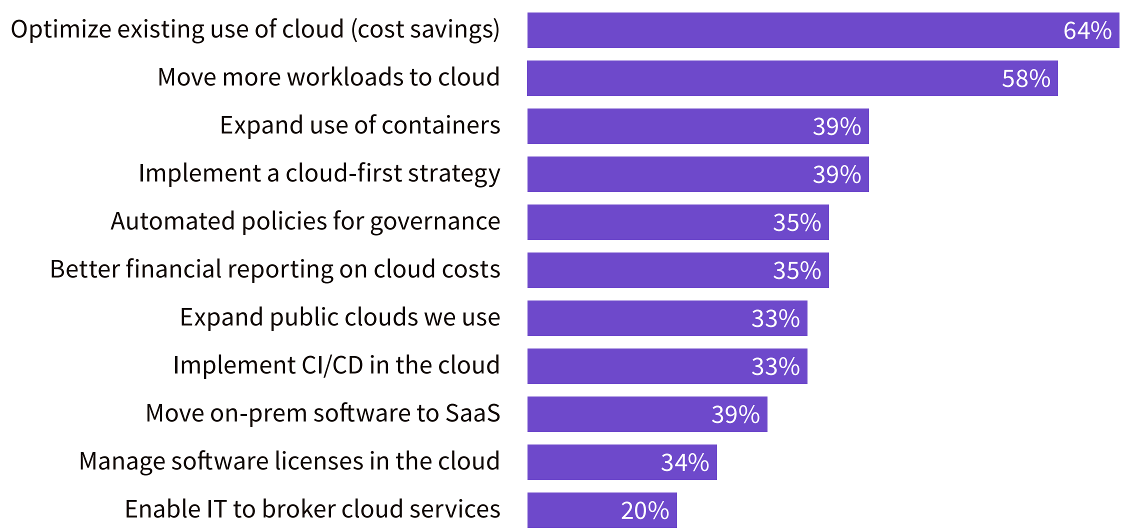
With the advent of Kubernetes a new form of serverless has emerged. Using technologies like Knative, developers can deploy serverless containers that scale up and down on demand just like FaaS functions. The advantage of containerized serverless is the ability to run full applications rather than only functions. FaaS requires new architectures that break applications down into their smallest atom components. With containerized serverless, developers can use technologies and patterns they are already familiar with while taking advantage of serverless scaling.

The dream of serverless computing is pretty simple: Developers deploy into infrastructures they don't have to manage, set up, or maintain. Once they upload a simple cloud function it just works. It scales on demand and shuts down when not in use.

Cloud challenges

Cloud computing, even with all of the flexibility, scalability, and efficiency it promises, is also immensely complicated. I&O and application development leaders have to consider not only business objectives but the people who work within these cloud environments. Migrating to the cloud, and managing the complexities of the cloud once it's implemented, takes significant resources. A cloud migration doesn't always ensure that problems get solved—it can just create newer, more expensive problems.

Top Cloud Initiatives in 2019



Source: RightScale 2019 State of the Cloud Report from Flexera

Cloud costs

Cloud computing ***can*** be cheaper, but that doesn't mean it always is. The "pay for what you use model" is a little bit deceptive in that, yes, you can avoid the costs of running your own servers all the time, but if cloud usage isn't monitored closely, costs can end up a lot higher than if they'd just managed the servers themselves. Often using the cloud unblocks projects that were stuck waiting on manually provisioned infrastructure. This means more projects are being completed leading to a surge in cost.

Flexera in its State of the Cloud report found that cost optimization was a huge priority for enterprises. What the survey found is that many respondents weren't doing basic cost-optimization, and with cloud usage being very much a micro-transcation world, costs can creep higher quickly.

Cloud sprawl

The idea of sprawl is nothing new in the computing world. Historically, the biggest example of sprawl was storage. Physical servers are a bit harder to sprawl because anyone can walk into a data center or network closet and count servers. When VMs became more popular, compute resources didn't have to physically exist and virtual machines had seemingly unlimited CPU and RAM available. Businesses didn't worry too much about abandoned workloads.

Cloud, in comparison, doesn't give anything free. Forgotten or abandoned workloads that didn't matter on a VM can bust budgets in the cloud, and teams end up paying for resources they didn't use—the very reason they didn't want to manage their own servers in the first place.

In addition to forgotten or abandoned workloads, Shadow IT can quite often lead to cloud sprawl. When employees order and spin up cloud servers, or bring in outside SaaS tools without IT's knowledge, those additions build up over time. Bringing in unauthorized tools can also cause potentially serious risks to the security and integrity of data, and result from a lack of communication between the central IT department and the business unit.

Cloud complexity

Cloud computing is managing more data than ever before: Storing data, organizing data, and ultimately using data to release faster software or deliver applications. In the past, moving small amounts of data was relatively easy. When we start talking about exabytes of data, rather than terabytes of data, the process of migration becomes herculean. According to Jean-Luc Valente, the VP for product management in the cloud platforms and solutions group at Cisco, egressing that kind of data to a public cloud could cost as much as $30 million dollars. Because of these migration costs, data continues to be kept in different repositories scattered across accounting, CRM, HR, and other cloud services, and will continue to operate with infrastructure and data distributed across multiple providers. Everything from security to business processes and workflow are impacted.

Adding to this complexity are layers of processes, people, software development lifecycles, and toolchains which make it harder to establish workflows that actually work. Then add containers and pipelines. Then add to that knowledge gaps. Leaders have to navigate these challenges, but they can't do it alone.

I&O and application development leaders want development and IT operations teams working together to meet business objectives. They need processes that facilitate meaningful collaboration, visibility across teams, systems that are highly-adoptable and scalable, and architecture that enables cloud native application development.

As organizations continue to adopt cloud infrastructure and cloud services, they need cloud-agnostic partners to help them:

- » Optimize multi-cloud
- » Encourage DevOps collaboration
- » Increase operational efficiency

Harnessing the cloud

> "Our customers want workflow portability, no matter what cloud that application is deployed on." - Sid Silbrandij, co-founder & CEO of GitLab

85% of enterprises currently operate in multiple clouds. Mitchell Hashimoto, founder and CTO of HashiCorp, described the importance of multi-cloud for future growth:

> *I think the #1 value of multi-cloud is organizational: You build your core infra/app lifecycle processes (dev, build, deploy, monitor, etc.) around a technology-agnostic stance. As technologies shift, other clouds become important, new paradigms emerge…your organization is likely more prepared to experience that change.*

Tools like Terraform are often seen as "multi-cloud" because it allows teams to manage anything with an API as code. Even if 100% of your compute is on one provider, you could easily use that tool with another provider, or over multiple providers.

Why organizations are choosing multi-cloud

When teams aren't tied down to vendor-specific services (e.g. Lambda, DynamoDB, and BigQuery) they have the freedom to use the platforms they need and deploy anywhere they see fit. Being solely dependent on a single cloud vendor is not necessarily bad, but it can limit the flexibility of an organization and leave it susceptible to vendor lock-in. Workflow portability is one of the benefits of multi-cloud, and it's important to prioritize it as early as possible (transitioning will be much harder later on).

There are several reasons why most businesses have adopted multi-cloud, and why more will continue to use this approach:

» **Better acquisitions**: Whether an organization wants to grow through acquisitions (or be acquired itself), existing systems can work within another company's infrastructure, even if both are using separate cloud providers.

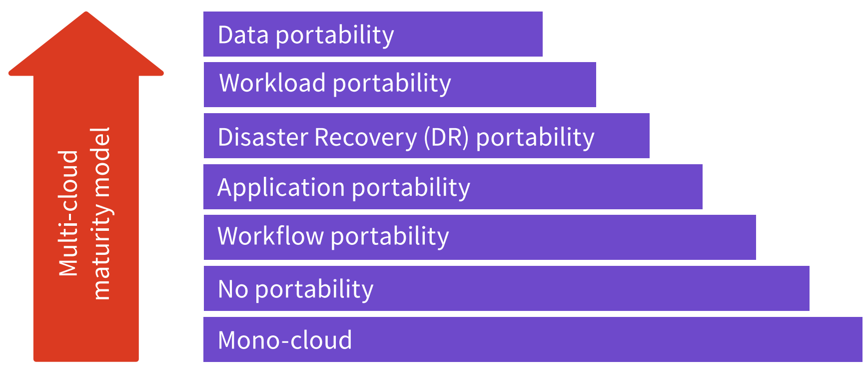
» **Deploy anywhere**: Have the same workflow across clouds.

» **Greater flexibility**: Each cloud vendor shines in some areas and is weak in others. Using multiple vendors lets you use the right tool for the job.

» **Increased resilience**: Architecting failover between multiple cloud providers lets you stay up even if one of your vendors is down.

» **Improved cloud negotiations**: If another cloud vendor offers better terms or significant credits, businesses can have better leverage because their DevOps processes are not tied to vendor-specific services.

» **Fewer conflicts of interest**: With cloud service providers offering so many different services, you may find yourself in conflict with customers competing in those same spaces (such as what happened with Walmart and Amazon).



Even though an organization may be using multiple cloud providers, that does not imply that an organization is multi-cloud. If each team is dependant on one cloud, then each is operating in a mono-cloud environment. The ability for information to move between clouds is what separates multi-cloud from multi-provider, with portability and interoperability key factors. As teams work through the multi-cloud maturity model, they increase portability by insulating cloud services from underlying infrastructure like processors, operating system and virtualization software, through layers of abstraction.

Mono-cloud

All applications are in one cloud. With this strategy, a company goes "all-in" with one cloud provider for the ease of use, or because the services offered meet current business needs. The organization is locked in.

No portability

There may be separate teams within the same organization, and each are working out of different cloud providers, but each team is working in its own mono-cloud environment. This structure uses multiple clouds but is not technically multi-cloud.

Workflow portability

Workflow portability is what makes deploying anywhere possible. Instead of having to tailor certain workflows to certain clouds, developers can have one workflow with cloud-independent DevOps processes and frameworks for making deployment decisions.

Application portability

The ability to move apps/data to any cloud is an aspiration for many IT teams. This can also apply to organizations that want to move applications back and forth from Private to Public clouds. In this scenario, applications can run on any cloud, and cloud-specific services are abstracted. Application portability is hard to attain because it requires engineering interfaces as abstractions. It also leaves organizations using only the features that are common to all clouds so they miss out on any specialty capabilities that could improve their processes.

Disaster Recovery Portability

In this scenario, applications can fail over to another cloud with limited downtime. Even if you are not managing architecture in-house, that doesn't mean that your applications aren't running somewhere. If a cloud provider's data center should go down, organizations have the ability to switch to another provider.

Workflow portability

The goal of workload portability is for organizations to shift application workloads between multiple clouds dynamically (for example: autoscaling servers for background jobs). Workload portability makes it possible to migrate elements of a business service, such as an instance of an application, OS, and data, to the appropriate infrastructure so that it can service the needs of the user.

Data portability

Data portability is a feature that lets users take their data from a service and transfer or "port" it elsewhere, typically through an API.

Multi-cloud on Kubernetes

Containers, microservices, and container orchestration tools like Kubernetes play an important role in the automation organizations need to optimize cloud infrastructure.

Kubernetes has helped companies turn the idea of multi-cloud into a reality. Nodes from multiple clouds can be added to a single cluster. Since Kubernetes abstracts applications running in containers from the infrastructure they run on, it enables true workload portability. By being able to run the same container images across multiple cloud platforms, IT teams can control their environment while maintaining leverage for pricing. Google open-sourced the Kubernetes project in 2014, and today Kubernetes is hosted by the Cloud Native Computing Foundation (CNCF). Kubernetes has been a valuable asset for cloud native development because it combines the Google knowledge of running production workloads at scale with best practices from the larger open source community, and can be run in any container environment.

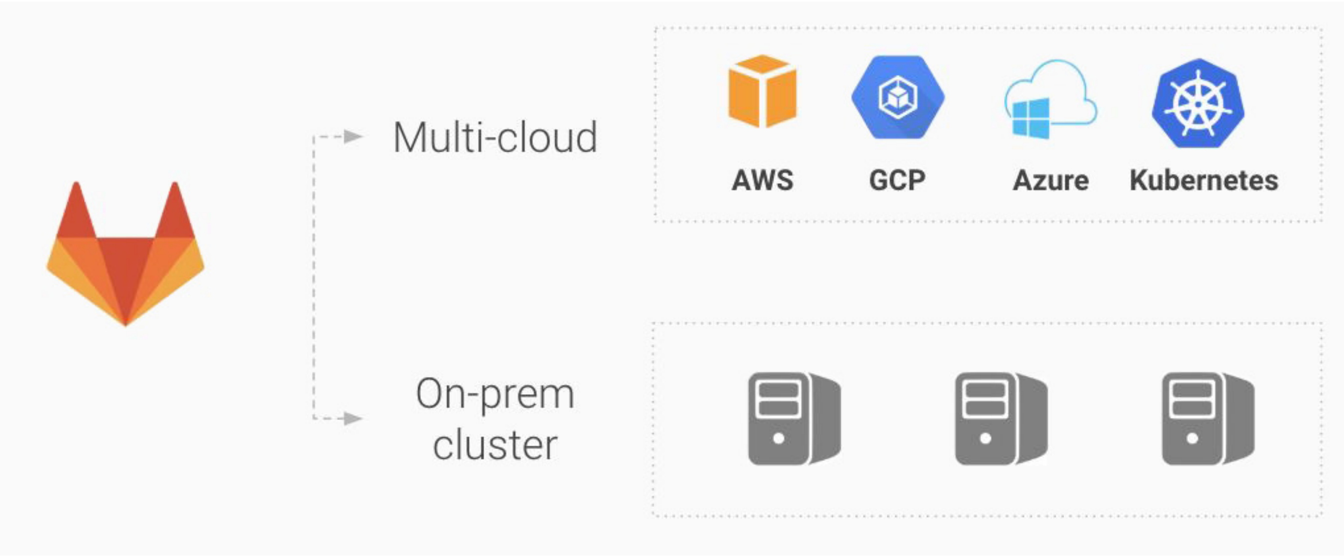
Becoming truly cloud native: Delta

Large companies know that building in silos becomes all too easy as needs grow. What happens is enterprises end up with unwieldy infrastructures, with different teams having workloads in different clouds. Sometimes they want to shift those workloads to different clouds, but this reasonable goal becomes difficult due to cloud providers optimizing for vendor lock-in with their tools, partnerships, and timing.

The DevOps team at Delta wanted to avoid the perils of enterprise cloud lock-in to create a first-class experience for their 5,000 developers. They decided to revitalize their process so they could focus on improving their cloud computing infrastructure and commit to Kubernetes-driven development. See how they did it.

Watch the presentation

How GitLab enables multi-cloud



Businesses want to choose cloud providers for their inherent value and use the services that best meet their needs. A multi-cloud future gives organizations the flexibility to deploy anywhere and run workloads across multiple clouds. Partnering with cloud-agnostic vendors will be imperative to this mission, and avoiding reliance on cloud-dependent processes will give teams consistent workflows. Instead of prioritizing infrastructures or working within the confines of a certain cloud, development teams can spend more time creating applications that add real business value to their users.

GitLab provides a complete DevOps platform that allows teams to have the same productivity metrics, the same governance, irrespective of what cloud you use.

“Choosing a cloud provider should depend on the company’s business objectives, it should not be constrained by technology, and GitLab wants to enable every one of our customers to have this freedom,” says Sid Silbrandij, co-founder and CEO at GitLab. Having the right partners in place to minimize the learning curve to set up, maintain, and use clusters can increase the odds of cloud native development success. GitLab invested in Kubernetes to give teams the ability to spin up clusters in the cloud of their choosing, and provides a complete DevOps platform delivered as a single application, ensuring visibility across the entire SDLC.

The future of cloud computing

In the cloud of tomorrow, enterprises large and small are able to function at “cloud speed” thanks to a cohesive cloud strategy. In the multi-cloud future:

» Organizations of the future are embracing multi-cloud, supported by solutions that are truly cloud-agnostic and offer all of the flexibility and visibility they need to move seamlessly between services.
» IT infrastructure is elastic and scales up or down automatically; teams can build and test applications for web, mobile and API without worrying about their architecture.
» Data is secure, accessible across any network, and optimized for machine learning and artificial intelligence.
» Teams can deploy to any environment in any cloud with a single deployment workflow.
» Cloud providers compete with each other to provide best-in-class infrastructure and services, and software companies have the support from independent vendors to use the tools, platforms, and services that contribute to long-term growth.

About GitLab

GitLab is a complete DevOps platform, delivered as a single application. Only GitLab enables Concurrent DevOps, unlocking organizations from the constraints of today’s toolchain. GitLab provides unmatched visibility, radical new levels of efficiency and comprehensive governance to significantly compress the time between planning a change and monitoring its effect. This makes the software lifecycle 200% faster, radically improving the speed of business.

GitLab and Concurrent DevOps collapse cycle times by driving higher efficiency across all stages of the software development lifecycle. For the first time, Product, Development, QA, Security, and Operations teams can work concurrently in a single application. There’s no need to integrate and synchronize tools, or waste time waiting for handoffs. Everyone contributes to a single conversation, instead of managing multiple threads across disparate tools. And only GitLab gives teams complete visibility across the lifecycle with a single, trusted source of data to simplify troubleshooting and drive accountability. All activity is governed by consistent controls, making security and compliance first-class citizens instead of an afterthought.

Built on Open Source, GitLab leverages the community contributions of thousands of developers and millions of users to continuously deliver new DevOps innovations. More than 100,000 organizations, including Ticketmaster, ING, NASDAQ, Alibaba, Sony, and Intel trust GitLab to deliver great software at new speeds.

GitLab