Revolution of Cloud Technology in Software Development

Bhanuprakash Madupati

United Airlines, Il, Dec 2019

Abstract
Software development has been fundamentally re-imagined by cloud technology through scalable, available-on-demand services and resources. Enter Infrastructure as a Service (IaaS), followed by Platform as a Service (PaaS) and now Software as a Service (SaaS); these models have disrupted the traditional development landscape enabling organizations to concentrate on their primary applications without investing heavily in infrastructure. And integration of DevOps and Continuous Integration/Continuous Deployment (CI/CD) practices helped to unify the cycle even more by speeding up updates and automating deployments. Cloud technology may offer many benefits, but also brings with it some challenges including data security and privacy both at stake as well as vendor lock-in translates in the form of integration cost impacting legacy infrastructure. In this paper, we will discuss the changes that have occurred in software development with cloud technology and how it works.

Keywords: Cloud Computing, DevOps, Continuous Integration, Continuous Deployment, IaaS, PaaS, SaaS, Automation, Software Development

I. Introduction

Software development has been one of the areas most transformed by Cloud computing, because its scalable on demand resources released developers hand from infrastructure management complexity. Before cloud services were available, businesses had to heavily invest in on-premises hardware which required capital expenditure up front for expensive hardware and months or years before it could be commissioned as well as maintenance costs ongoing. The arrival of cloud technology in the forms of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software etc, Scaled on demand with virtualized environments which can scale according to workload demands [2]. The models are not only an economical way of using infrastructure but also provide flexibility to businesses by choosing the service model that best suits their needs, IaaS for control at infra level and SaaS as a complete managed software. This is what saw the wide spread implementation of DevOps practices, due to the adoption of cloud computing. DevOps Brings Operations and Development Teams Together — By blending development teams requests with operations team requirements, DevOps helps to make the wall come down. By integrating automated provisioning and configuration tools with cloud platforms in delivery pipelines, DevOps methodologies have become much faster to deploy updates as well as new features without the previous overheads from manual provisioning. Instead it spawned a new wave of CI/CD pipelines that automate testing and deployment tasks resulting in quicker delivery times with increased quality software [1],[3]. On the other hand, tools like Jenkins for automation, Kubernetes and Docker (packaging) are now cornerstones of a cloud platform that can help developers to automate desired processes across all stages from source code control up to production without needing any manual intervention.

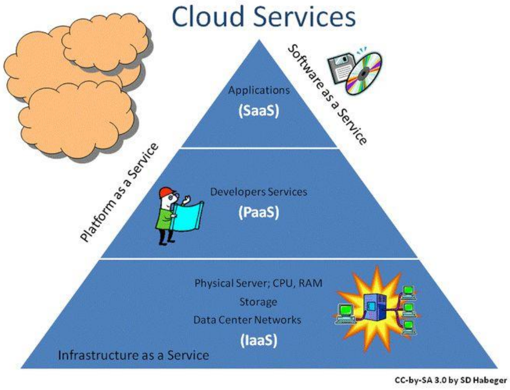
While cloud computing has several benefits it brings with its set of challenges that organizations need to solve. Escrow of information is critical, and never more so than in a multi-tenant environment such as the data center or public cloud where multiple customers share the same physical infrastructure. On the other hand, provide data isolation and encryption within cloud-native systems is hard, with maintaining compliance of same industry regulations where GDPR & HIPAA are key compliant standards [2]. In addition, there is a substantial risk of vendor lock-in which means that if an organization becomes too dependent on one cloud provider then it could very hard and expensive to switch providers or return back to an on-premises infrastructure. Moreover, connecting cloud services with legacy fleets bring about technology and operational challenges that translate into a significant amount of re-engineering on both domains [5].

This paper explains how the movement to cloud changed the Software Development landscape for ever. It looks at the way cloud service models such as IaaS, PaaS and SaaS have modified conventional construction practices in addition to how DevOps availability for CI-CD has automated enhanced automation efficacy. The paper also looks at the major challenges of cloud computing from data security through vendor lock-in to integration with existing legacy systems. And finally, looking at what's next in cloud technologies like Artificial Intelligence (AI), the role of edge computing and why more organizations care about sustainability when it comes to their cloud infrastructure.

2. Cloud Service Models

Now days cloud computing provides mainly three kind of services which help in development part IaaS, PaaS & SaaS. Model abstraction and control- These models offer a choice of level of abstraction, which any organization can pick as the best solution based on their requirements. These models have changed the way software is being developed, which in turn has helped developers innovate quickly and concentrate on writing applications instead of handling infrastructure.

The image below are the 3 different types of cloud service models visually. IaaS is the ground floor — it provides the most control over infrastructure and third-party management in general, while PaaS abstracts you further by removing much of that same operating burden. At the very top, SaaS offers are a pre-configured solution to say: “Here is something that you can use right away without managing any platform or infrastructure. Every model here is based on the one under it — with different levels of abstraction and flexibility. Offering varying levels of abstraction and flexibility.



2.1. Infrastructure as a Service (IaaS)

It is called infrastructure as a Service because it provides the virtualized computer resources over internet where clients has there full control on their own facilities. IaaS allows businesses to rent virtual machines, storage and networking components on a per-use basis without having to pay for/ maintain physical IT hardware [2]. IaaS platforms like Amazon EC2 and Google Compute Engine give developers the ability to configure their servers by hand. Because IaaS allows you to increase your infrastructure with changing workloads -- which is vital for applications that receive variable demand during the day.

One of the main benefits IaaS provides is cost saving with less capital spending. These businesses replace expensive hardware that could be underutilized with flexible infrastructure which they can scale up and down as required, while only paying for what's used at each moment. Furthermore, IaaS platforms provide capabilities like auto-scaling, load balancing and disaster recovery to maintain high availability & ensure peak performance even at scale. IaaS is what powers modern software development, as its scalability designed to handle variable traffic patterns makes it a perfect solution for unpredictable applications.

Still, with the greater control IaaS allows for comes increased responsibility. Developers must administer and control both operating systems, middleware, and applications in addition to securing that environment through proper security practice implementation including performance tuning. It involves a serious back end infrastructure management, which is almost the reason why it is perfect for organizations with professional IT teams [2].

2.2. Platform as a Service (PaaS)

PaaS abstracts the hardware infrastructure, and provides developers with a platform instead that also includes operating systems, databases, and even development frameworks. PaaS abstracts infrastructure management away from developers to concentrate on writing applications and deploying them, without the intricacies of server configuration. Google App Engine, Microsoft Azure App Service are the most popular PaaS platforms which provide pre-configured environment to rapidly develop and test an application & deploy it in no time [2].

PaaS does an excellent job of smoothing the development process, this is especially beneficial in agile and devops environments that needs to iterate extremely fast. PaaS comes with CI/CD (Continuous Integration / Continuous Deployment) tools, and it can automate every part of your software development lifecycle. Besides, it provides support for some popular software development tools like GitHub, Jenkins and Docker which promotes smooth collaboration between dispersed teams.

Best suited approach for small to medium-size enterprises (SMEs) and startups, PaaS platforms offer businesses solutions with less complexities. Their main advantage is that they allow companies without a dedicated IT team to get their foot in the door with managed platforms. But PaaS can also be constraint where customization and configuration are concern as developers will not have access to the system environment due to systems pre-configuration nature which would limit most of the application that may need certain functionalities [3].

2.3. Software as a Service (SaaS)

In simpler terms, SaaS provides on-demand software applications via the internet and users can interact with them in a web browser or through an API. SaaS, on the other hand takes abstraction to it greatest because when we truly use a cloud application service no infrastruction of any kind is needed. SaaS operates over the internet and most of these applications can be used without downloading or installing. Salesforce, Google app suite (Gmail), Microsoft 365 are few key examples that provide cloud productivity tools for personal and business use [2].

SaaS has transformed how software is consumed and delivered. Businesses do not need to buy and install software on their local machines, instead they can subscribe SaaS services over the internet on a pay-as-you-go basis and save upfront cost as well as make maintenance easy. The SaaS provider manages everything such as infrastructure, platform, and software management from updates to security or scaling providing businesses with the ability to concentrate on their core operations.

An obvious advantage of SaaS is how it can scale to a much greater extent than traditional, monolithic software — where we would have thousands (if not millions) individual users all sharing the same infrastructure. SaaS is best suited for applications that are used by a wide user base, such as customer relationship management (CRM) systems, collaboration tools and enterprise resource planning (ERP). However, the downside to SaaS applications is that they are managed by third-party vendors and therefore organizations will be limited in terms of customization and data security [2].

Deployment Models	Holder	Security	Scalability	Cost
Private Cloud	Single private organization	Higher than other deployment models	Limited	High
Community Cloud	Two or more private organizations with identical requirements	Lower than Private Cloud and higher than Public and Hybrid Cloud	Limited	Medium
Public Cloud	Cloud Service Provider (CSP)	Lower than other deployment models	Very High	Pay-per-use
Hybrid Cloud	CSP and private organizations	Lower than Private and Community Cloud and higher than Public Cloud	High	Pay-per-use

3. DevOps and Continuous Integration/Continuous Deployment (CI/CD)

The adoption of DevOps principles into the cloud has created a sea change in how software is developed and released, an acceleration previously unknown to most enterprises through improved collaboration, faster automation — among other things. DevOps — a superior Agile approach that focuses on collaboration between software developers and IT operations to automate the development, testing, and deployment process. The importance of DevOps has increased with the advent of cloud computing which provides unprecedented scalability and flexibility.

Continuous Deployment (CD) is the process of every code change going to production after all tests have passed. To facilitate the deployment of these applications across different environments, one way to achieve greater consistency and reduce downtime is through containerization by tools like Docker and Kubernetes on cloud platforms [3].

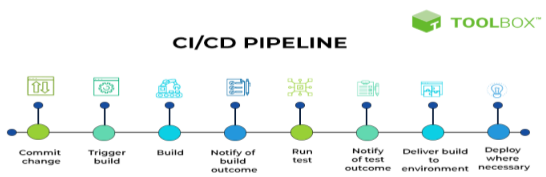


Figure 1: A Typical CI/CD Pipeline in Cloud Environments, demonstrating the sequence from code commitment to deployment.

3.1. DevOps in Cloud Environments

At the heart of DevOps is breaking down wrd separating development and operations teams and encouraging them to function as a single unit, where both parties take responsibility for delivering software more with better efficiency. Some of the cloud platforms that have infrastructure most suitable for DevOps practices are Amazon Web Services (AWS) and Google Cloud, with tools also supporting automation, monitoring and continuous feedback loops [2]. Cloud environments provide a dynamic provisioning mechanism where DevOps teams can deploy your applications and infrastructure in real time eliminating the manual interventions.

DevOps in the cloud You cannot talk about DevOps in the cloud without mentioning Infrastructure as Code (IaC). With IaC you can manage the cloud resources in code which allows developers to automatically create and configure infrastructure. Tools such as Chef, Puppet and Ansible are common in cloud-based DevOps workflows to automate infrastructure management which usually contribute to consistency while reducing human errors [4]. As a result, development teams can rapidly deploy applications and scale resources as end users demand more application computing power—all in real time. Monitoring tools — Cloud platforms are also packed with monitoring tools, which is one of the key aspects for a DevOps practices. These services do not only provide real-time analytics to application delivery and performance but also helping operation teams catch problems before they become user-facing. Such a feedback loop is critical to ensuring cloud-native applications continue to be available and perform well [1].

3.2. Continuous Integration (CI)

It is the practice where multiple developers work in a shared repository and are expected to frequently merge code updates into that helping reduce problems when deploying, automated testing process against their merged codes. Code integrated into multiple code branches typically took days, or even weeks to be included back in the traditional development environments creating massive integration conflicts and release delays. CI became more integral to the development process and could be executed many times a day with its use in cloud environments [4]

CI Cloud platforms have also made it possible for the uptake of CI practices in which scalable infrastructure is provided that powers the automation frameworks. See: frequent tools used in CI pipelines for automated testing, integration and deployment processes are Jenkins [8}, Travis CI [5], They are tied in with cloud platforms so you can spawn a new environment on-the-fly, execute tests concurrently and auto-scale resources based on the test suite size. It lessens the time needed to find and debug a given issue which allows faster growth in development.

Developers experience faster feedback loops and lower integration risks when targeted at cloud-based CI environments. All of those things are ensured by automated testing, which ensures that code changes have validation and prevent from introducing bugs into the codebase. Therefore, CI pipelines help development teams ship better software in less time for accelerating the productivity and speed-to-market overall [3].

3.3. Continuous Deployment (CD)

Continuous Deployment (CD) takes CI further by automatically deploying the software to production systems, once it passes all tests in an incremental and automated way. In CD, when you pass your automated testing then everything is deployed to production cut without any intermediate step or manual approval which reduces the time of releasing new features and bug fixes [3].

The automatic up-down features, balancing the load, automating infrastructure management all provided by cloud platforms are perfect for CD. Containerisation of applications, to run them in consistent manner across different environments is well known property and CD pipelines often use tools like Docker & Kubernetes for the same[4]. Containers allow packaging applications along with all the dependencies, configuration files and libraries making them lightweight enabling rapid deployment from development to testing and till production.

These shipments from the market also facilitate auto-scaling and self-healing infrastructure for high availability and resilience security in CD pipelines in cloud environments. This is more critical for applications that deal with varying traffic loads as cloud provides automatic scaling up and down, guaranteeing services to be available without any manual operation [3].

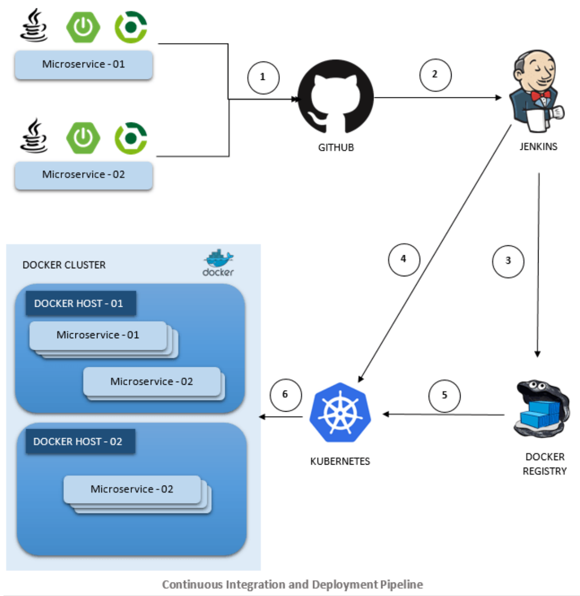


Figure 3: CI/CD pipeline with Docker & Kubernetes

Continue with the diagram to see how code getting committed in GitHub, testing automatically via Jenkins, and containerized through Docker and Kubernetes orchestration are all part of Continuous Integration and Deployment.

4. Limitations and Challenges

Cloud technology provides tremendous advantages such as scalability, flexibility, and efficiency. However, it also comes with many challenges that organizations must address. With the growing digitization of businesses, more and more enterprises are moving their operations to the Cloud. A global survey recently found that 95% of organizations have adopted some form of cloud service. Yet concerns such as data security constraints, vendor lock-in challenges, legacy system integration drawbacks and increasing costs associated with maintaining on-premises back-ends hinder the successful conduction proliferation towards business-critical assets hosted in commercial off-the-shelf (COTS) or homegrown white-boxes built running bare-metal hypervisor software stacks adapted out offerings from server-centric firms VMware Inc., Citrix Systems etc How do these factors impact upon your company's pursuit after having embarked full steam ahead into reinventing everything around new build chain products development approaches based not only around virtualized environments but also leveraging specialized storage arrays using scale-out NASs equipped with optimization techniques like deduplication underpinning other optimizations for parallel support overall improving along vertical-scalable servers hosting priority workloads handle lowering low-level suboptimal runtime conditions adjusted appropriately according means testing configuration management tooling setups whether automation facilitates assumptions needed fulfill general requirements requested validate proper functioning ongoing when intentions transitioning various applications zones express included within broader dynamic multi-provider fulfilling stack credentials; yet over time verification still proves necessary route destined success going forward? Failing to understand these limitations could weaken any organization between employing the advantages of cloud technology and moderating its drawbacks.

4.1 Data Security and Privacy

Data security is often considered the most critical issue related to cloud technology. Sensitive data are off-premises in managed third-party cloud providers' data centres, and there is a risk of a data breach and unauthorized access since the organization lacks control over the business's possessed data [2]. Multi-tenancy is a common trait of the cloud platform, and it allows multiple users to share infrastructure on which data can leak between tenants if tenant isolation rules are not strong enough.

They deploy security measures, e.g. encryption, firewalls and access controls, to secure customers' data. After all, there is evidence of potential security breaches — remember when top cloud providers had data leaks and hacking attempts? Furthermore, using data privacy rules like the General Data Protection Regulation (GDPR) and Health Insurance Portability & Responsibility Act HIPAA might not comply, which can be a challenge for multi-regional businesses since Cloud vendors can store your information in destination [2]

In addition to a ransom, organizations organize a deployment environment by releasing their data and accepting higher exposure than physical deployments. To solve these struggles, organizations are turning towards hybrid Cloud or multi-cloud design that aids in keeping sensitive information on-premises and using non-sensitive workloads in the Cloud [2].

4.2 Vendor Lock-In

Vendor lock-in describes how hard it is to move workloads and apps from one cloud provider to another because of the incompatibility or high cost of jumping ship. Because Cloud service users are typically offered a set of propriety tools and APIs, it may be easier for an organization heavily invested in one vendor's ecosystem to transition to that provided by another with substantial reconfiguration or loss of some functionality [2].

For instance, transferring AWS-built applications to Microsoft Azure may require rewriting or optimizing the codebase to tap into Azure's unique services. However, heavy reliance on one cloud provider is inherently inflexible and locks organizations into an approach that may not be best practice if another supplier offers the same capabilities, she said. In addition, the charges that can drive data growth costs many times to move these volumes of information across cloud environments prevent vendor lock-in.

To minimize the dangers of vendor lock-in, businesses are increasingly using multi-cloud strategies, utilizing regular economic cloud vendors so as not to be dependent on just one supplier. In addition to the increased range of options, this method enables organizations to balance performance and cost by choosing the right provider for specific workloads [4].

4.3 Compatibility with Existing Systems

Organizations face the problem of seamlessly integrating cloud solutions with their legacy systems—another big challenge when they adopt a new technology trend. Most companies have a large amount of on-prem hardware and traditional applications that are not built for Cloud. Cloud migration of these legacy systems frequently incurs costly re-engineering efforts [5], increasing the overall cost and complexity.

CES use cases like legacy system integration can be challenging, especially for products that rely on custom hardware and software (like Point-of-Sale terminals or billing systems), which cloud-native services cannot necessarily replace. Similarly, there may be issues related to the consistency and latency of data and compatibility between systems in place today (on-premises legacy) vis-a-vis applications based on cloud technologies [17]. For this reason, numerous businesses often take the hybrid cloud path — keeping consequential legacy systems on-premise and progressively transferring non-core workloads into the Cloud.

Cloud providers provide various tools and services to help organizations tackle the issues they encounter as part of their journey towards transitioning legacy systems. Examples of services that help organizations assess and migrate their workloads with minimal downtime are AWS Migration Hub (source) and Migrate for Compute Engine by Google Cloud. However, industry observers said those companies are still concerned about migration's complexity [4].

5. The Future Of Cloud Technology

Traditional GDP Aggregation and the Future of Cloud Computing: These trends will evolve to characterize what we view as cloud computing today.

5.1. Integration of AI and Machine Learning

Cloud platforms are giving AI and Machine Learning (ML) services, enabling organizations to use data-based insights with scalability measures [4]. The services will likely further embed within cloud platforms and provide faster, more innovative applications.

5.2. Multi-Cloud Strategies

Many businesses are adopting multi-cloud strategies to prevent vendor lock-in by splitting workloads across different providers for cost and performance efficiency [2]. At the same time, this will be more common with the maturation of multi-cloud orchestration tools.

5.3. Edge Computing

With edge computing, data processing occurs closer to the source and reduces latency, making real-time applications such as autonomous vehicles or IoT-based devices possible. It is growing so much that cloud providers are beginning to broaden their edge services to improve this increasing demand. [5]

5.4. Sustainability Green Cloud computing

Cloud providers are committing significant finances to their sustainable and green cloud computing projects using renewable energy for data centres. SaaS has allowed organizations to optimize and optimize technology practices, leading to a front [5].

6. Conclusions

1. Cloud technology has transformed the software development ecosystem with its scalable and on-demand provisioning of resources.
2. IaaS, PaaS and SaaS wont the organizations have allowed optimization of their Software Development practise leading to reduction in Infrastructure complexity.
3. DevOps combined with CI/CD pipelines have streamlined software delivery, making companies far nimbler and decreasing time-to-market.
4. Cloud computing has challenges like data security and vendor lock-in legacy system integration.
5. AI, edge computing, and sustainability are the avenues along which Cloud will march in its future steps to unleash software innovation further.

References:

1. L. E. Lwakatare, “DevOps adoption and implementation in software development practice: concept, practices, benefits and challenges,” *Urn.fi*, 2017. Available: https://urn.fi/URN:ISBN:9789526217116.
2. P. Brebner and A. Liu, "Performance and Cost Assessment of Cloud Services," *Service-Oriented Computing*, pp. 39–50, 2011. doi: https://doi.org/10.1007/978-3-642-19394-1_5.
3. M. Shahin, M. Ali Babar, and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017. doi: https://doi.org/10.1109/access.2017.2685629.
4. L. Satyanarayana, “Cloud Computing, Revolution in New Web Technology: Application and Challenges Survey,” *InternatIonal Journal of Computer SCIenCe and teChnology*, vol. 4, 2013, Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1c60b1d17a94cccf3b1bec126bb8a8e9f15b992f. [Accessed: Sep. 12, 2024]
5. H. Zhou, The Internet of Things in the Cloud. CRC Press, 2013.
6. J. Wettinger, V. Andrikopoulos, and F. Leymann, "Automated Capturing and Systematic Usage of DevOps Knowledge for Cloud Applications," *Proceedings of the IEEE International Conference on Cloud Engineering*, 2015. Available: https://www.iaas.uni-stuttgart.de/publications/INPROC-2015-01-Automated-Capturing-and-Systematic-Usage-of-DevOps-Knowledge-for-Cloud-Applications.pdf.
7. B. Shestakofsky, "Working Algorithms: Software Automation and the Future of Work," *Work and Occupations*, vol. 44, no. 4, pp. 376–423, 2017. doi: https://doi.org/10.1177/0730888417726119.