

White Paper

Doing the Cloud Right: 5 Keys to Unlocking Business Value

Table of Contents

Introduction	3
The Cloud Adoption Divide – Haves and Have-Nots	5
The Complex Systems Dilemma	10
Pipeline Strategies to Enable 5 Key Characteristics of Cloud	14
Doing Cloud Right – Final Considerations	17
CloudBees and Release Orchestration	20

Introduction

There's no longer any debate about the merits of cloud computing. Optimally, adopting the cloud provides organizations in every industry the opportunity to increase responsiveness, improve resilience and deliver a superior customer experience. Yet this opportunity is far greater for companies that realize it's not just about adopting cloud; it's about using it in essential ways. For these select few companies, a myopic focus on rehosting existing legacies and technical encumbrances gives way to architecting new cloud patterns that consistently meet and surpass the goal of faster software delivery and improved operational performance.

Failing to fully embrace all the essential characteristics of cloud computing makes it difficult for organizations to gain positive returns. Old ways of working mean developers and testers cannot use cloud capabilities to rapidly deliver software that delights customers and exceeds business needs. Worse still, executives become frustrated when they aren't able to reap the promised benefits of increased speed and stability.

In part, this is due to rehosting traditional systems without having a migration plan to continuously modernize and take advantage of cloud native capabilities.

Elite performers recognize that cloud is a journey, not a destination, which means taking a more holistic approach to adoption and embracing all five essential characteristics of the cloud computing paradigm. This is especially important within the software delivery management process itself, in which teams can exploit the cloud patterns needed to accelerate application architectures that help future-proof businesses by using advanced automated tools.

The Cloud Adoption Divide – Haves and Have Nots

According to a recent report by Forrester, cloud adoption continues to grow. In 2019, 65% of North American enterprises relied on public cloud platforms, and 66% ran internal private clouds.¹ Irrespective of industry vertical, it's no longer a case of when to move to the cloud, but how to execute a cloud adoption strategy to gain and retain a competitive advantage. But adopting modern technology is only half the battle. Companies must also incorporate new methods to use these technologies to meet the accelerated delivery of stable software services.

[*The 2019 Accelerate State of DevOps Report*](#) from DevOps Research and Assessment (DORA) continues to strongly validate that increasing deployment frequency, reducing lead time, restoring service faster and lowering change failure rates are the best critical indicators of software delivery and operational performance.



In 2019, 65% of North American enterprises relied on public cloud platforms, and 66% ran internal private clouds.

FORRESTER®

¹ Forrester: [Predictions 2020 - Cloud Computing Sees New Alliances and New Security Concerns](#), November 4, 2019, Dave Bartoletti, Vice President, Principal Analyst.

And, for organizations in the “best of the best” – or elite group, when compared to low performers, this means:

- Executing 208 times more frequent code deployments
- Recovering from change failures 2,604 times faster
- Maintaining lead times (code commit to deploy) that are 106 times faster
- Lowering change failure rates by 7x

As impressive and important as these advantages appear, elite performer groups are actually business differentiators too; they are twice as likely than low performers to exceed their organizational performance goals. Importantly, the DORA report also confirms that while increasing software delivery speed is possible without compromising stability, this acceleration is being fueled by transformational technology shifts – especially cloud computing.

In terms of cloud adoption, the DORA report also suggests many organizations are not as fully invested in the cloud as they believe they are. By relying on quick wins associated with rehosting legacy systems (often called “lift and shift”) and failing to take advantage of important cloud characteristics and capabilities, teams can never elevate their performance to those levels exhibited by elite performers.

The DORA report also identified that 80% of organizations use some form of cloud platform to host a primary application or service.



However, and as the findings further suggest, what really matters is not that organizations are just using the cloud, but how they implement the technology to truly drive business value.

So, while many organizations can report they've adopted the cloud, only 29% of the DORA survey respondents who claim to be using cloud computing agreed or strongly agreed that they had implemented the five essential characteristics of cloud computing (see Table 1) as described in National Institute of Standards and Technology (NIST) special publication 800-145.

Importantly, elite performers are 24x more likely to have met all essential cloud computing characteristics than low performers. This can partially explain the increasing satisfaction gap between the cloud 'haves,' who implement essential practices to gain sustainable value, and the 'have nots,' who persist with traditional mismatched processes and fail to achieve realize the promise of increased delivery speed and operational stability.

Clearly, those organizations that can fully use cloud services are better positioned to gain an advantage over those that don't. This is verified by the DORA survey results, which strongly suggest that by leveraging all essential characteristics of cloud, companies are better able to achieve elite levels of performance and make massive improvements in speed, responsiveness, scalability and customer experience.

By fully embracing the cloud and moving beyond basic lift and shift migrations, teams become better positioned to attain elite levels of performance. This involves moving towards a cloud native approach, adopting new architectural approaches that are synergistic with cloud and implementing tools and processes engineered and optimized for cloud environments.

The Five Essential Characteristics of Cloud Computing (percentage agreed or strongly agreed)

On-demand self service	Consumers can automatically provision computing resources as needed, with human interaction from the provider.	57%
Broad network access	Capabilities can be accessed through heterogeneous platforms such as mobile phones, tablets, laptops, PC's and workstations.	60%
Resource pooling	Provider resources are pooled in a multi-tenant model, with physical and virtual resources assigned on-demand. The customer may specify location at a higher level of abstraction such as country, state or datacenter.	58%
Rapid elasticity	Capabilities can be elastically provisioned and released to rapidly scale outward or inward, appearing to be unlimited and able to be appropriated in any quantity at any time.	58%
Measured service	Cloud systems automatically control, optimize, resource use based on the type of service such as storage, processing, bandwidth and active user accounts.	62%

Table 1: NIST's Five Essential Characteristics of Cloud Computing

The Complex Systems Dilemma



There are incredible challenges facing organizations as they strive to achieve the elite levels of performance described in the DORA report. These include:

Application architecture

The industry is moving from traditional monolithic systems to architectures that comprise loosely coupled microservice elements. While these architectures promise greater team independence and are an important enabler of continuous delivery, they will introduce many new operational overheads. For example, shipping an increased number of application components into production will require extremely robust and scalable release orchestration.

Different release cadences

Organizations are gradually moving from delivery architectures that support infrequent cycles associated with traditional applications, towards modern architectures that promise the nirvana of continuous delivery. This suggests that teams will require solutions that can support the existing release workloads, along with the ability to quickly model more complex application pipelines and coordinate major, cross-product releases combining traditional and cloud native applications.

Operational architecture

At the same time, the operational architectures that teams are deploying are also changing dramatically and adding more complexity. Organizations are quickly moving from virtual machine mainstays towards technologies that introduce further levels of abstraction, including containers, serverless computing and various functions -as a Service.

Orchestration architecture

Software pipeline tools themselves are also changing rapidly. Manual methods and scripting are being supplanted with declarative abstract models, where pipeline processes may be defined as code and stored and versioned in source repositories.

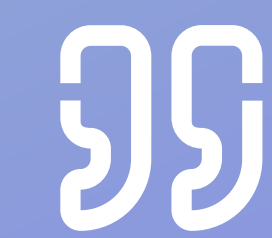
All of these dynamic architectural shifts introduce incredible opportunities for driving massive improvements in the way software is developed and delivered. However, as we've learned over time, the emergence of modern technologies doesn't necessarily displace (or need to displace) existing applications and services. Therefore, the challenge isn't managing releases across old and new technologies, but in supporting a truly hybrid world.

The net-net of these dynamic shifts is that organizations are rarely lucky enough to manage only one application. In reality, large enterprises deal with thousands, even tens of thousands, of applications and components. These systems are running across all kinds of runtime environments and architectures, so teams must now manage and coordinate releases across multiple applications, platforms and cloud services.

Adding further release complexity is the fact that teams must now manage their traditional releases side-by-side with DevOps-style initiatives that have much faster release cycles.

All this results in the greater need to control and track from a single pane of glass the exact nature of release payloads and application versions deployed in any cloud environment, at any point in time. Also essential is top down visibility into the precise status of a release across the pipeline, including: which stakeholders are waiting for it, as well as any constraints, bottlenecks or elements of waste that are impeding the smooth and uninterrupted flow of value to customers.

Importantly, and as illustrated in Table 2 (below), superior enterprise release tools should support those lean practices that consistently help identify and eliminate eight damaging elements of waste across the pipeline.



Superior enterprise release tools should support those lean practices that consistently help identify and eliminate eight damaging elements of waste across the pipeline.

Waste	Elimination Tips
D efects	Reduce software bugs and security vulnerabilities by incorporating automated testing within and across the release pipeline.
O verproduction	Prevent costly overprovisioning by optimizing resource pooling using loosely coupled architectures.
W aiting	Reduce wait times by providing fully automated self-service provisioning capabilities to developers.
N on value added processing	Avoid building distinct release systems for every cloud platform with a model-based approach that decouples environments, apps and release processes.
T ransportation	Reduce the cost of acquiring (transporting) resources by dynamically managing pipeline and deployment workloads at extreme scale.
I nventory	Identify redundant systems and components through complete visibility into all application release artifacts.
M otion	Eliminate manual handoffs and human-centric bottlenecks to increase delivery speed.
E xperience (unused)	Give staff complete visibility and control across the entire pipeline to identify areas and opportunities for improvement.

Table 2: Eliminating the Eight Elements of Waste (DOWNTIME) Across the Release Pipeline

Pipeline Strategies to Enable the Five Key Characteristics of Cloud

Achieving the elite levels of performance described earlier in this paper involves using advanced tooling that incorporates the five essential characteristics of cloud computing within release pipelines. In the context of supporting speed and stability goals, these strategies include:

1 On-demand self-service

One of the best and easiest ways to use the cloud is to give developers and testers single-click access to pre-configured cloud environments, with quotas and tear down processes automatically built into the process.

A self-service catalog includes built-in support for release process best practices, like gating and security integrations (e.g., static code analysis). In essence, granular and loosely coupled approaches enable self-service, with more advanced pipeline strategies allowing applications and environments to be specified independently.

2 Broad access

Beyond only supporting major cloud platforms (such as AWS, Azure, Google), a pure model-based approach that decouples cloud infrastructure, environments, applications and release processes/pipelines is necessary. This allows organizations to target any multi-cloud (or on-premise) environment without changing the pipeline, release process or application configurations.

A model-based approach also allows environments (like QA, test, etc.) to be built on-demand and removed when not in use. This reduces the cost associated from moving from one cloud to another or from on-premise to the cloud (and vice versa).

3 Resource pooling

The ability for production systems to optimize resource usage is critical, but it's also important that the software pipeline (which can and should be considered a production system itself) is fully optimized to ensure the continuous delivery of important updates, security patches and new releases to production systems.

Optimized resource pooling again involves decoupling applications from environments plus offering the ability whenever possible to dynamically provision (and de-provision) full pre-production environments.

Other best practices include allowing the scheduling of static resources to minimize blocking, and ensuring environmental configuration consistency to reduce the need for developers and testers to hoard costly and underutilized infrastructure.

4

Rapid elasticity

Release pipeline infrastructure must be elastic in terms of operating at extreme scale to support constantly changing release requirements and workloads. This could involve deploying ten different applications to a handful of applications one day, or responding to a CVE (Common Vulnerabilities and Exposures) entry by deploying a patch to thousands of systems the next day.

5

Measured service

In the context of release pipeline infrastructure, measured service involves gaining complete visibility from code check-in to production deployment in order to drive continuous improvement in people, process and technology. With iteration frequency increasing exponentially, so does the opportunity to quickly and efficiently analyze data to identify hidden bottlenecks and constraints. However, this means being able to constantly measure release performance against business outcomes.

Doing the Cloud Right: Important Conclusions

A modern release pipeline is incredibly flexible and dynamic. It can accommodate many different commit pipelines with numerous microservices that are independently developed. These microservices can be deployed to production as needed, as each has its own independent lifecycle.

Additionally, the modern pipeline will include specific release pipelines. The release pipelines will include processes for provisioning environments, integration testing and methods to control execution flow in the event of failures. These modern pipelines increase the number of moving parts and complexity, but that's a good thing.

More moving parts means services can be developed, moved and fixed independently without compromising the application or service as a whole.



A modern release pipeline is incredibly flexible and dynamic.

Exploiting these advantages and doing the cloud right requires maturing capabilities around the essential concepts discussed in this paper.

To summarize, this involves:

Setting the right goals

Understand that it's not sufficient to move to the cloud, it's about how to optimally move to gain technology and business advantage. Lift and shift migrations can quickly help organizations gain a tick in the cloud strategy box, but hosting old monolithic technology on new systems is no way to exploit the value that cloud architectures deliver.

Starting fast and adapting

Give stakeholders access to all the resources they need as quickly as possible by supporting on-demand, self-service. Always consider that any pipeline solution should support both legacy and cloud native during periods of transition.

Never forgetting control

Always ensure release pipelines provide flexible governance so that problems can be detected and remediated quickly.

Increasing visibility

With so many moving parts, cloud systems demand extreme visibility. Focus on centralized system visibility by integrating data from all cloud deployments and monitoring tools.

Delivering speed and scale

Recognize that moving to the cloud will require more flexibility and scale from the pipeline and release processes.



As *The 2019 Accelerate State of DevOps Report* indicates, the cloud drives DevOps success and helps teams achieve elite levels of performance. But while the cloud provides massively scalable, flexible and versatile platforms for building and deploying software, true success comes when organizations use the five key characteristics of the cloud. Nowhere is this more important than in building effective release orchestration and the strategies now needed to build pipelines that result in *doing the cloud right*.

CloudBees and Application Release Orchestration



Building a successful business – and maintaining and expanding that business in a sustainable way – is about coming up with novel ideas and developing products and services to meet customer demand.

But standing in the way is the challenge of coordinating release approvals, environments and deployment activities across multiple teams and shared infrastructure.

Organizations want a way to manage their releases and CD pipelines in a scalable, auditable and predictable way.

Predictability goes a long way towards eliminating the stress associated with the software release process. This is where CloudBees can help.

Widely recognized as the leader in CI/CD, CloudBees offers industry-leading solutions like CloudBees Core and CloudBees Flow. CloudBees Core is a CI/CD automation engine with the flexibility to support diverse software portfolios and the unified governance required by growing organizations. CloudBees Flow is an application release orchestration platform that enables teams to implement fast and adapt easily, with unprecedented insight and control of all types of releases – at any scale.

A core component of a Software Delivery Management strategy, CloudBees Flow provides a way to orchestrate the entire software delivery lifecycle and tie together all the software development tools you already use, automating the repetitive parts of the process.

CloudBees Flow can also be part of a collaboration platform for the organization that is based on shared common data. This facilitates more high-value collaboration from the creative, highly-trained individuals that make up most software engineering teams, while providing transparency to other departments like marketing, finance and customer support.



Widely recognized as the leader in CI/CD, CloudBees offers industry-leading solutions like CloudBees Core and CloudBees Flow.

Everyone in the business is on the same page and has visibility as a feature or application moves from research and ideation through development, production and deployment. The business has more insights into the process and therefore the ability to evaluate an application's success.

The end result? Software that engages customers and delivers on business KPIs.

Learn more

Optimize DevOps with Application Release Orchestration

- Download the eBook [Optimize DevOps with Application Release Orchestration](#)
- Read the White Paper [DevOps Performance: The Importance of Measuring Throughput and Stability](#)
- Read the White Paper [Continuous Everything: Why Software Delivery Management Matters](#)

Learn more cloudbees.com/get-started



CloudBees, Inc.
4 North Second Street, Suite 1270
San Jose, CA 95113
United States
cloudbees.com
info@cloudbees.com

Jenkins® is a registered trademark of LF Charities Inc.
Read more about Jenkins at: cloudbees.com/jenkins/about

© 2025 CloudBees, Inc., CloudBees® and the Infinity® logo are registered trademarks of CloudBees, Inc. in the United States and may be registered in other countries. Other products or brand names may be trademarks or registered trademarks of CloudBees, Inc. or their respective holders.