CloudBolt software

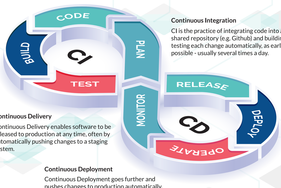
INFOGRAPHIC

A New Way for CI/CD

Standardized infrastructure, continuously tested

What is CI/CD?

Traditionally, development of applications involved siloed workloads with numerous teams working on numerous parts of development, only to merge them later to create a complete product. This often leads to "merge hell" and creates numerous difficulties as well as inefficiencies that **CI/CD (continuous integration/continuous deployment)** attempts to solve.



Continuous Integration

CI is the practice of integrating code into a shared repository (e.g. Github) and building/testing each change automatically, as early as possible - usually several times a day.

Continuous Delivery

Continuous Delivery enables software to be released to production at any time, often by automatically pushing changes to a staging system.

Continuous Deployment

Continuous Deployment goes further and pushes changes to production automatically.

CI/CD KEY BENEFITS

- Promote Developers and IT Operations to work together, breaking silos
- Identify bottlenecks proactively to increase productivity
- Faster app/software delivery, improved innovation and recovery process

CI/CD Can Still Fall Short

While CI/CD offers tremendous benefits and is quickly becoming the standard operating procedure for many companies, it's not without its challenges. Aside from the common challenges of implementing a completely new system (training, upfront costs, fine tuning, etc.), even the most well implemented CI/CD pipelines can have shortcomings.

SLOW RESOURCE PROVISIONING

Developers currently take days or weeks to ready the technology stack (storage, network, hypervisor, disaster recovery solutions, etc.) required for their testing environment, thus causing delays in application development and deployment.

LACK OF RELIABLE TESTING ENVIRONMENTS

Test environments are not as reliable as expected. They might fail often because of change of state in storage, network and server configurations, passwords, VM (Virtual Machine) images, etc. One test failure means the entire test process needs to be restarted: invested effort, time, and money is being wasted.

SELF-SERVICE IT

Only 24% of end users have been using self-service IT[1]

FAILURE RATE

Topic branches have an average failure rate of 31%[2]

CI/CD - A NEW WAY

Old Way vs The Intelligent Way

Luckily, there's now a new way to CI/CD that addresses the key challenges for all steps of the process. Below is how the new, intelligent way of implementing CI/CD compares to the old way.

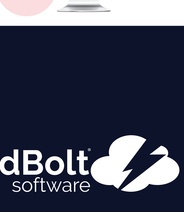
OLD WAY		INTELLIGENT WAY
Lengthy, Inconsistent Builds		**Fast, Standardized Builds**
Gen 1 Cloud Management Platform • Developed for technical admins only (~20% of users) • Lacks flexibility to orchestrate various steps • Does not provide extensibility to add modern technologies easily **Scripts or Infrastructure as Code (IaC)** • Does not scale as fast when the target environments change • Provides only limited environment visibility **Manual Machine Builds** • Time/labor intensive • Requires extensive knowledge/expertise	**DEV**	**CloudBolt Blueprints** • Deploy resources in minutes • Easily define how the environment will be orchestrated and who can provision resources • Give developers more visibility and control over their testing environments
Manual Testing CI = CD — MANUAL TESTING • Manual isolated test on the storage, networks, VM instances is a reactive process • Constant patches and upgrades make isolating the process difficult • Using change management and documentation or managing multiple tools has a steep learning curve	**TESTING**	**Continuous Infrastructure Testing** CIT CI CD • Proactively test your Continuous Integration (CI) environment end-to-end through automation • Automatically identify variance that would have stopped the CI process (password change, storage capacity available, etc.) • Get notified hours or days in advance if something isn't ready

THE BOTTOM LINE

Old Way		Intelligent Way
WEEKS TO PROVISION Builds take 30 minutes and the whole team is wasting time queuing.	DAYS/WEEKS vs MINUTES	**MINUTES TO PROVISION** Builds take minutes without wasted time.
MANUAL TESTING Testing requires multiple tools and constant attention.	REACTIVE vs PROACTIVE	**AUTOMATED TESTING** Testing is proactive and automatic.
DIFFICULT TO IMPLEMENT Requires numerous tools and hours of configuration in order to be efficient and secure.	NUMEROUS TOOLS vs SINGLE INTERFACE	**EASY TO IMPLEMENT AND RUN** Provides a single interface through which (via UI, API) pipelines are created, recreated and destroyed. CIT allows a secure and frictionless process.

See CloudBolt in Action!

REQUEST A DEMO



CloudBolt software

CloudBolt's hybrid cloud platform for enterprises helps IT admins provide simple to very complex IT resources to end users from a single portal.

1. FreeFormDynamics Survey
2. CircleCI blog