CloudBees Software Delivery Automation
Securing the Software Supply Chain

Bryan Guinn
Senior Account Executive - DoD
210.415.0330
bguinn@cloudbees.com

Tyler Johnson
Solution Architect
703.980.2777
tjohnson@cloudbees.com

Who We Are

The Company Behind Jenkins

We're the #1 corporate sponsor of Jenkins

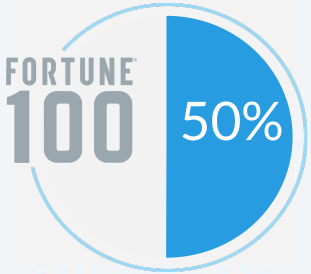
Industry Leaders

Our founders are open source veterans from Red Hat and JBOSS

Enterprise-**Grade** for the Mission

Government customers trust CloudBees with their critical delivery and

Founded in 2010
U.S. headquarters
~500 employees in 15+ countries
40+% YoY growth



Global 50

(Some) Federal Customers

Major Partnerships

Cloud Providers

Global Technology

CloudBees Customers are Faster

"Deployments from a couple per year to every two weeks."

13x

FASTER DEPLOYMENTS

"Three days to three hours."

8x

FASTER BUILDS

"Two months to one day"

40x

FASTER SET UP

Software delivery organizations encounter risk and inefficiency from disconnected groups, teams, and tools

> *Disconnected development and operations frequently breeds conflict and **inefficiency**, exacerbated by competing motivations and processes.* - **Wired**

Lack of integrated, end-to-end development, delivery and release

> *84% of respondents say inaccessibility of information gets in the way of their ability to do their jobs and/or make data-driven decisions. Respondents attributed this to **organizational and functional silos**.* - **ASG 2020 State of Software Delivery**

Lack of end-to-end visibility into the SDLC

> *59% of organizations **cannot keep an audit trail** of who changed what and when* - **DevSecOps 2019**

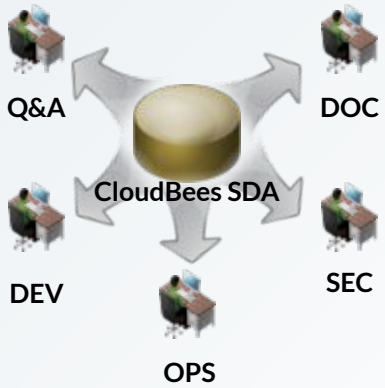
Lack of traceability and governance

Connect Fragmented Software Delivery Lifecycle across Groups, Teams and Tools

Software Delivery in Silos

- Each function uses disconnected point tools, processes and data.
- Inconsistencies across groups and teams creates friction and slows pace
- Errors, rework and redundancies escalate cost and audit risk

CloudBees Software Delivery Automation



- Formal enterprise scale connected CI/CD platform architecture provides governance and reuse.
- Entities process and assets readily available across CI/CD
- Shared orchestration across all groups, teams and tools provides full visibility across the SDLC

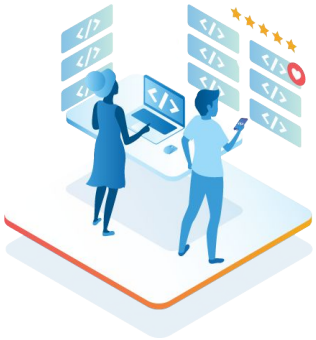
Modern Software Delivery Drives Mission Objectives

Increase Innovation

Drive Innovation Quickly

Enhance software development and delivery to increase speed, time to value, and getting capabilities into end users' hands.

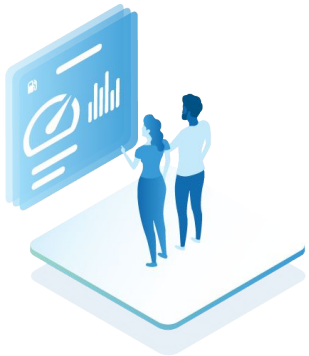
Reduce Risk



Reduce Risk by Ensuring Adherence to Internal and External Policies

Ensure a secure, compliant process and environment that produces safe, reliable outputs.

Increase Efficiency & Reduce Cost

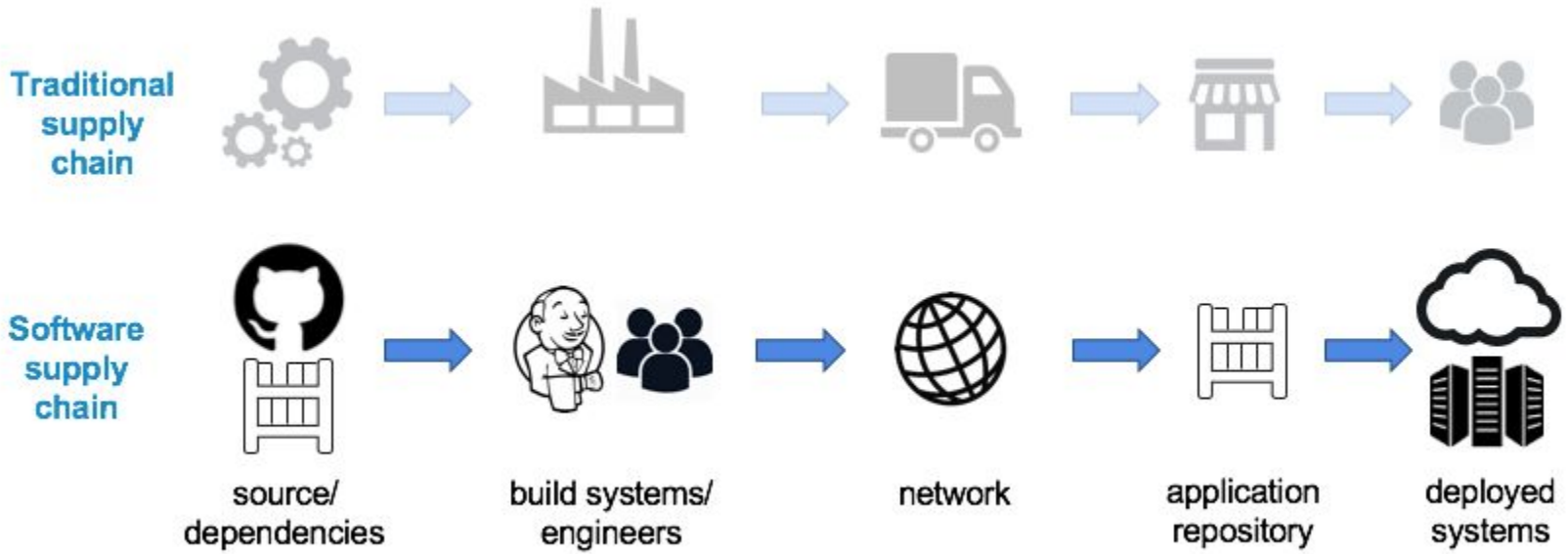


Optimize Software Development and Delivery Function

Eliminate wasted effort to free up staff to focus on mission-critical activities.

The Software Supply Chain

What is the Software Supply Chain?

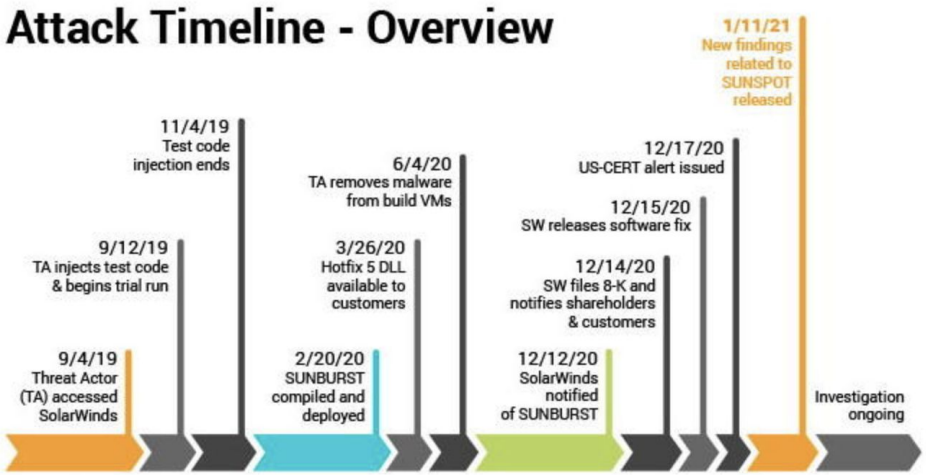


SolarWinds Incident

"Attackers may have leveraged malware at many points along the way as code moves through the software development lifecycle."

.

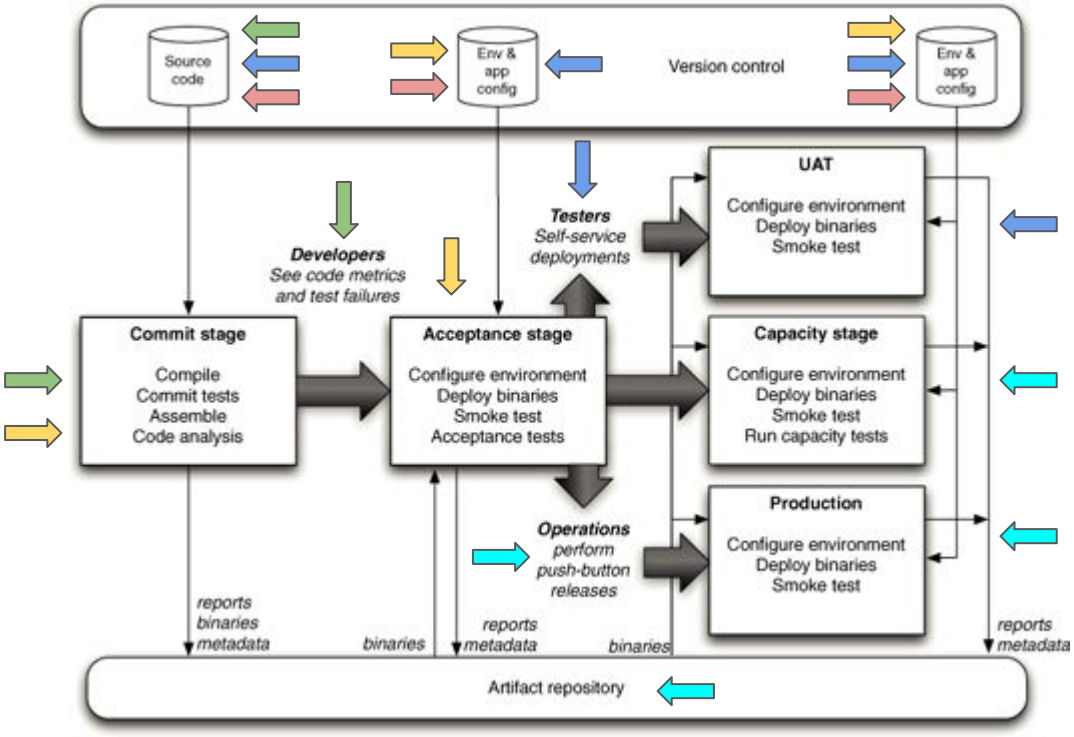
- Teri Radichel - @teriradichel © *2nd Sight Lab* *2020*



solarwinds

Increasing confidence in build's production readiness

Environments become more production like



Faster feedback

solarwinds

Possible Points of Entry

If the code got inserted on a developer machine, one would hope the change would be noticed in a review of source code changes, but some companies don't have such stringent reviews.

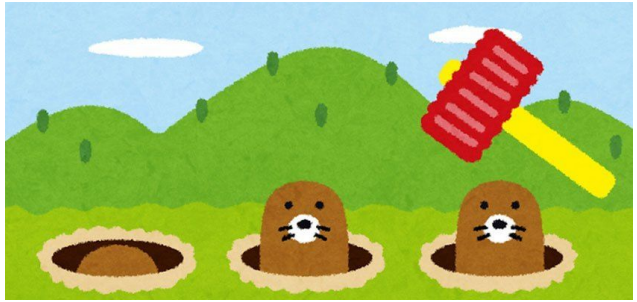
If the code was changed in the source control system perhaps it went unnoticed as it moved through the systems to production.

Automated processes may have injected the code if attackers obtained access to the servers that perform the automation.

If QA teams have access to change the code, any compromised QA system could have altered the outcome.

The attackers may have gotten access to the server that hosts the updates and simply replaced a good copy with a malicious copy of the software.

Fortify the Process! Don’t just plug the holes.



- Think of the Software Supply Chain like a bank.
- Securing the building is important but invest in a strong Vault
- CI/CD and Release processes are the “Vault”
- A strong internal process effectively mitigates risks like those from the Solarwinds incident.

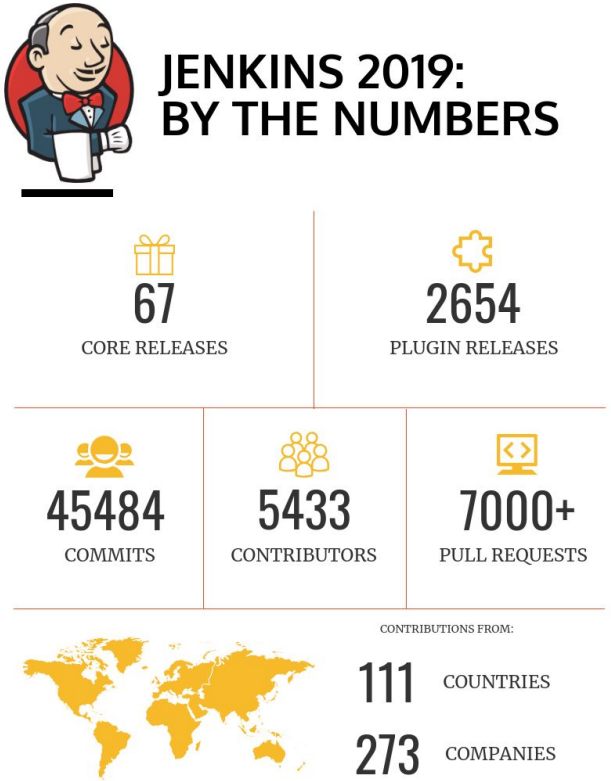
- Limiting the attack surface is important but should not be the focus.
- As you stop one issue another will inevitable arise.
- Securing the software supply chain shouldn’t be a game of whack-a-mole.
- Fortify the process!

CloudBees Software Delivery Automation

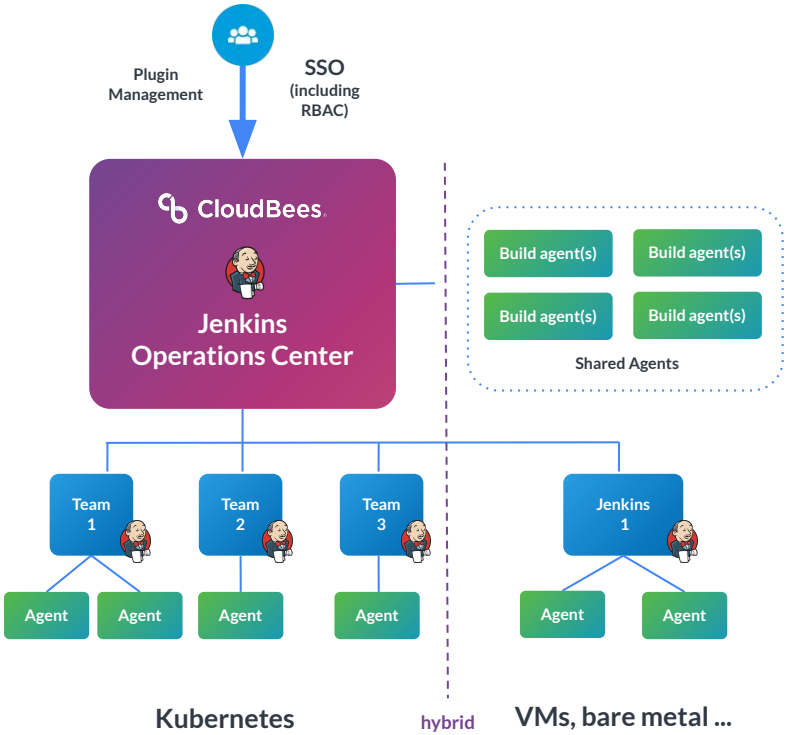
Securing the Software Supply Chain

Jenkins is the Way

- Open Source leadership for 15 years
- The leading open source automation server
- No vendor lock-in

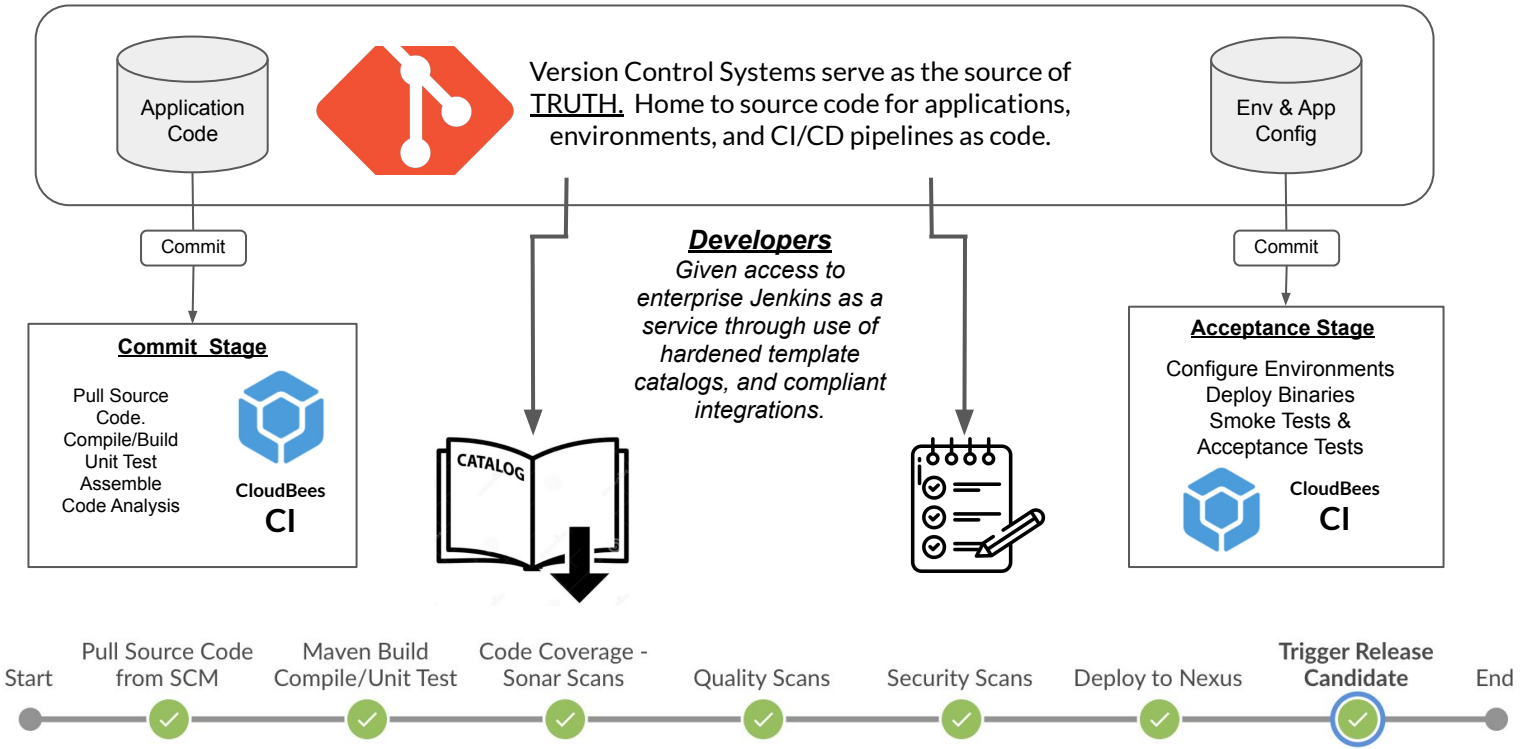


Centralized Control

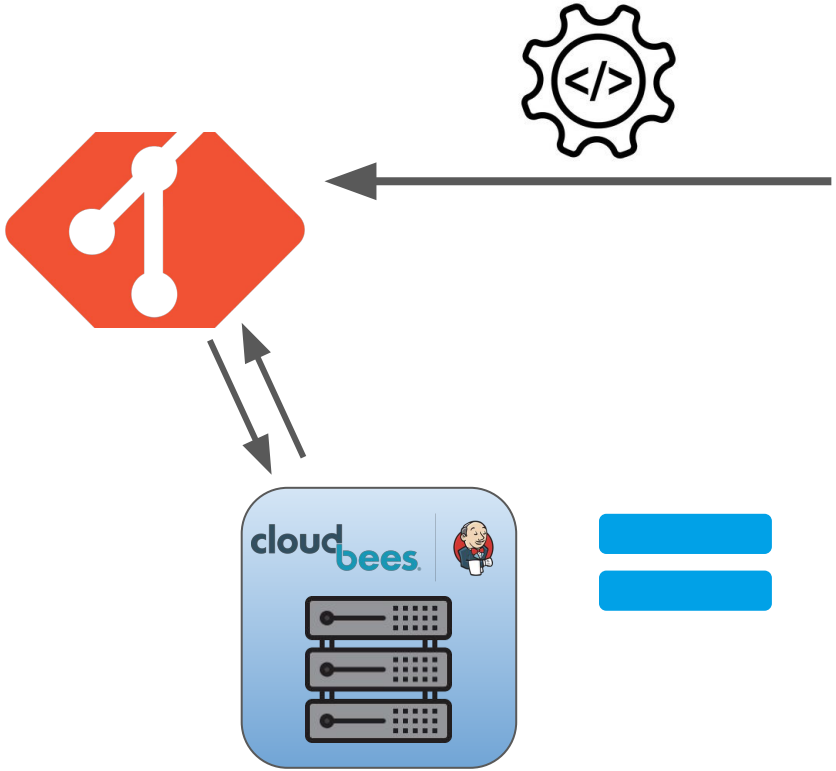


- **Stability**
  - Controlled use of plugins and use of Jenkins masters
- **Security**
  - Role-based access control to protect intellectual property
- **Compliance**
  - Ensure consistent CI/CD across all teams
- **Usability**
  - Shared agents to improve utilization and avoid bottlenecks
- **Scalability**
  - Manages multiple Jenkins masters

CloudBees CI with Compliant Pipeline Templates



Atlassian Bitbucket + CloudBees Configuration as Code (CasC)



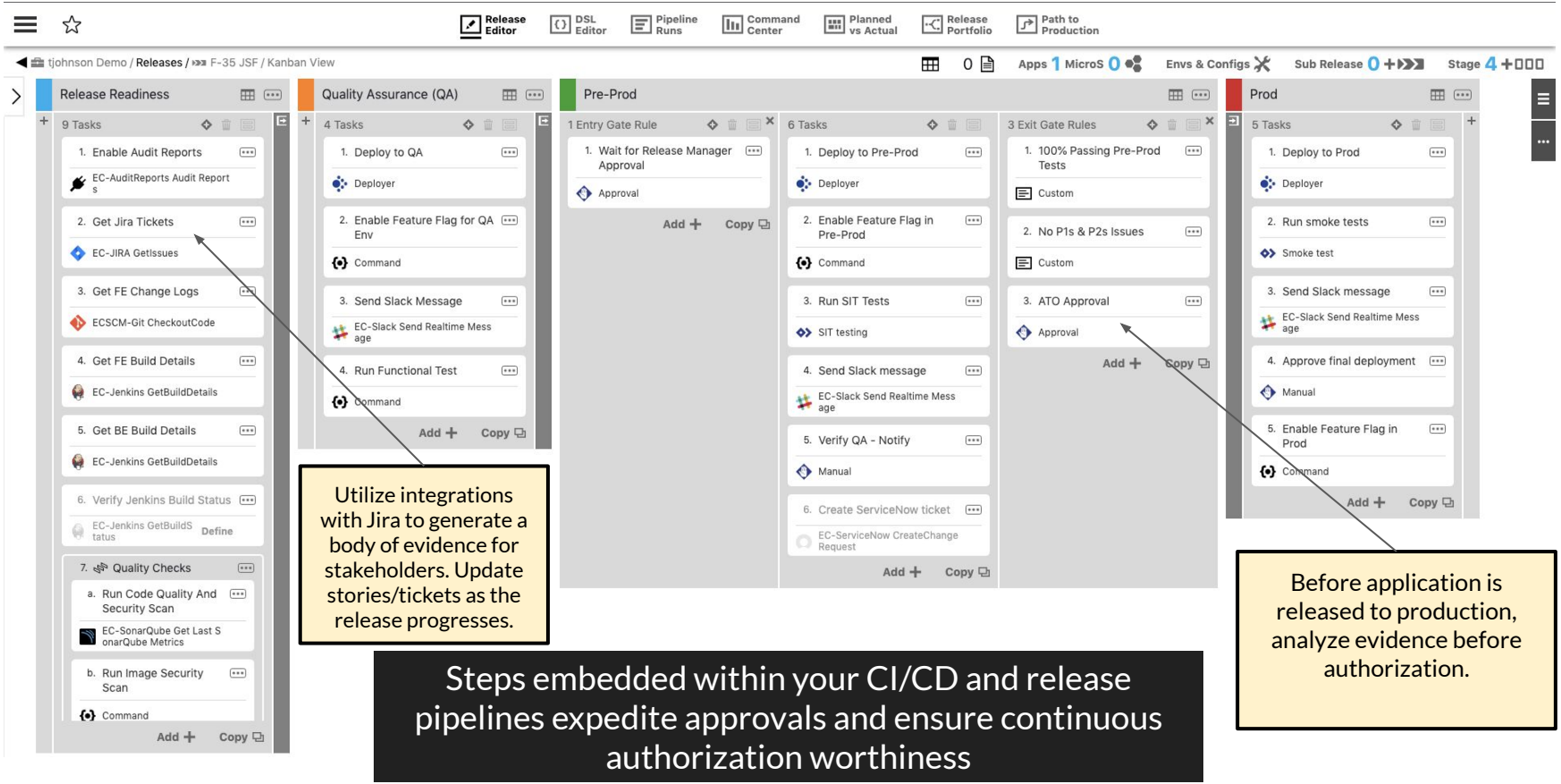
Defined as YAML Manifest

1. **`bundle.yaml`** - provides a version for the bundle (which must be incremented for bundle updates), and references the other files in the bundle.
2. **`jenkins.yaml`** - contains the Jenkins configuration as defined by the OSS Jenkins CasC plugin and supported CloudBees plugins.
3. **`plugin-catalog.yaml`** - provides a list of plugins that are ALLOWED to be installed on your *managed controllers* that are not already included as part of the CloudBees Assurance Program (CAP) Core plugins.
4. **`plugins.yaml`** - contains a list of all plugins that will be INSTALLED/UPDATED on the configured *managed controllers* - but plugins can only be installed if included via the `plugin-catalog.yaml`or if they are already included as CloudBees CAP plugins.
5. **`rbac.yaml`** - The purpose of the rbac.yaml file is to describe the role-based access control (RBAC) groups and roles defined at the root level.

CloudBees CD

Release Orchestration to Enable Continuous Authorization

Ensuring Trustworthy Releases



Utilize integrations with Jira to generate a body of evidence for stakeholders. Update stories/tickets as the release progresses.

Before application is released to production, analyze evidence before authorization.

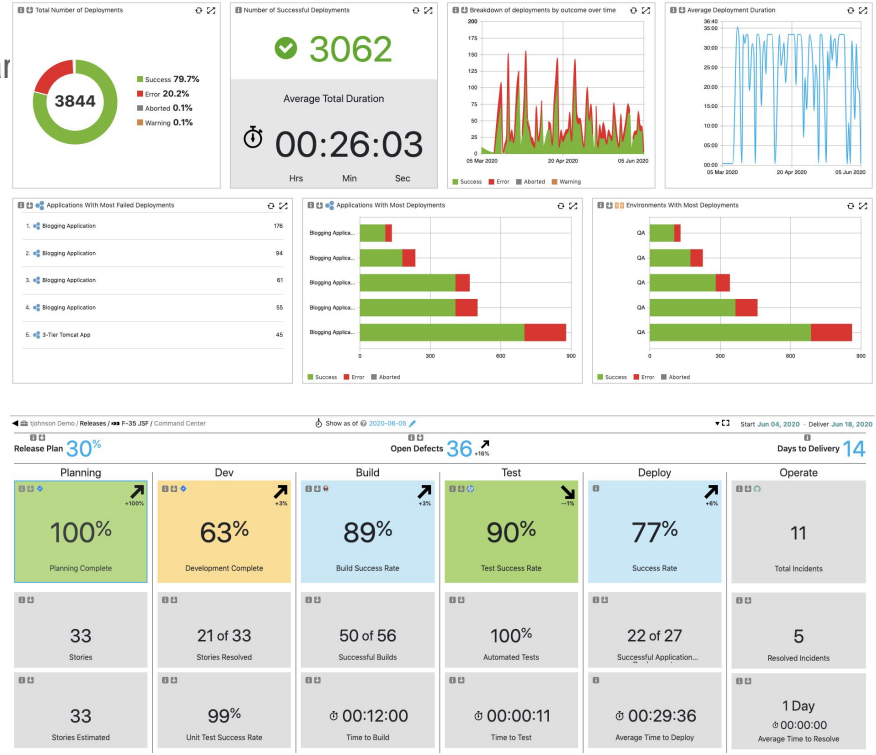
Steps embedded within your CI/CD and release pipelines expedite approvals and ensure continuous authorization worthiness

Value Stream Management

Insights to Measure and Improve Secure Deliveries

Improve the software delivery process and increase performance with realtime, actionable insights into the softwar value stream.

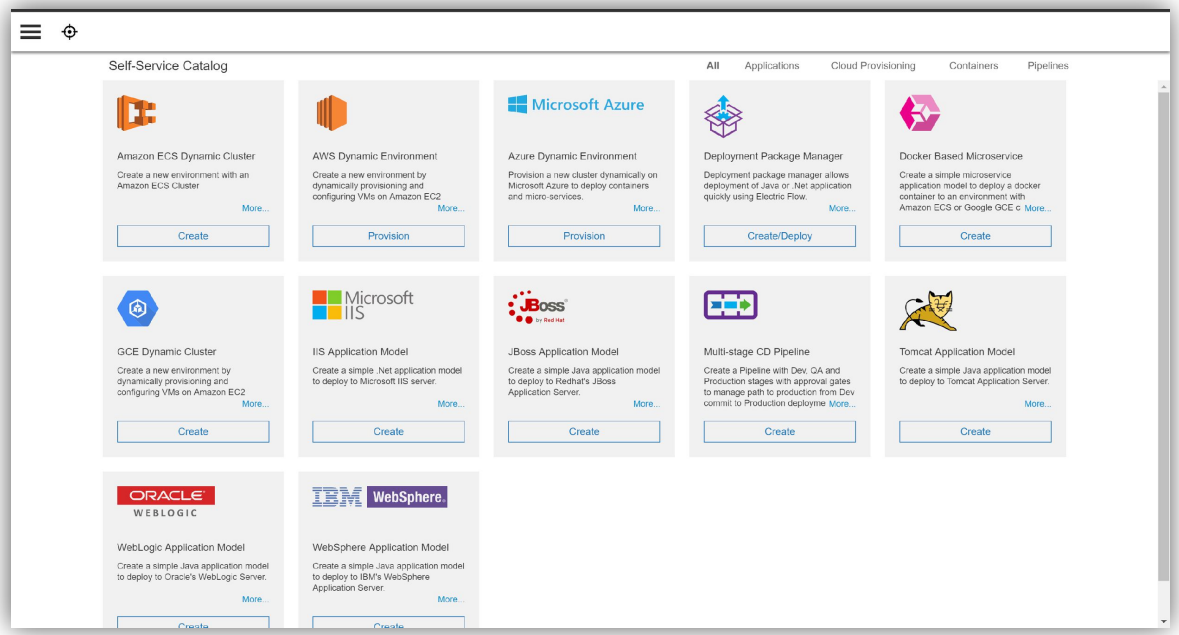
- **Track Performance:** Continuously monitor performance from commit to deploy
- **Continuously Improve:** Visually identify and eliminate blockers and bottlenecks across teams and tools
- **Drive DevOps Adoption:** Measure DevOps and DORA metrics within and across teams and releases
- **Demonstrate Value:** Quantify and communicate process and performance improvement to leadership



Self-Service Catalog
Facilitate Onboarding, Governance, Security and Best Practices

CloudBees CD has **prescriptive templates** and with flexibility and extension. Ready best practice content is exposed to users who select templates, provide parameters, and get started:

pipelines
containers
application models
environments
databases
processes
clusters
VMs



CloudBees Software Delivery Automation is a recognized market leader

Gartner

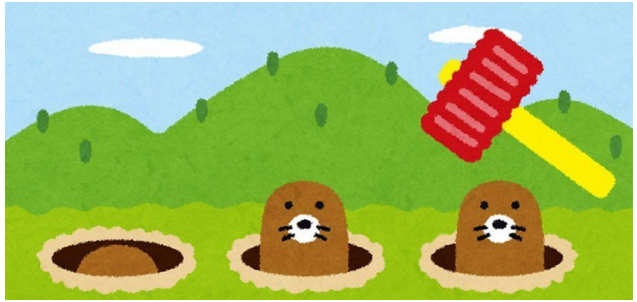
"An easy-to-use interface accelerates team and application onboarding; product enhancements and cadence demonstrate a strong ability to deliver on customer needs and requests."

FORRESTER®

"CloudBees has executed well on merging its products into an integrated platform that ranges from continuous integration and build management to release automation, with support for a number of platforms, from cloud-native to on-premises to mainframe and mobile.."

Thank You

Fortify the Process! Don’t just plug the holes.



Trying to block all potential entry points for you software delivery life cycle is like a game of whack-a-mole. As one issue is plugged another entry point springs a leak. While fortification of entry points is a critical piece to DevSecOps architecture, having a primary focus on hardening, standardizing, adopting CI/CD processes **Secure the Software Supply Chain.**

Instead think of the software supply chain as a bank. Banks don’t spend all their effort fortifying windows and doors. Instead they invest in uncrackable vaults. In this analogy CI/CD and release practices are the “vault”. If you have robust, compliant processes in place you mitigate vulnerabilities or malicious software being built into or deployed to secure environments. This means even if a breach like SolarWinds occurs again, a strong supply chain will detect and prevent the spread of any unwanted intrusions. Step 1 to securing the the supply chain starts with investing in a good vault!