



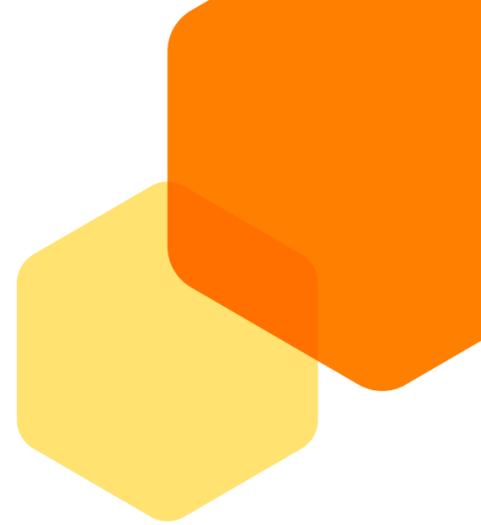
# CloudBees CD/RO Capability Guide

# Summary

CloudBees CD/RO is a platform designed to optimize your software development lifecycle. Automating deployment pipelines enables faster time to market without sacrificing security. Our robust set of integrations empowers developer tool choice while extracting contextual insights. As a result, you make more informed decisions in a repeatable and reliable manner to align software delivery to business outcomes. The following list is a summary of our key capabilities.

## Table of Contents

- 1 Release Orchestration
- 2 Deployment automation
- 3 Analytics Dashboards
- 4 Self-Service Catalog
- 5 Securing the SDLC
- 6 End user personalization
- 7 CI Integration



# Release Orchestration

Model and orchestrate your DevOps release pipelines across all tools

The screenshot displays a release orchestration tool interface for 'CloudBees SDA CD/RO'. The top navigation bar includes 'My work', 'Service catalog', and 'Release calendar'. The main interface is divided into four stages:

- Release Readiness (4 Tasks):**
  - Gather Evidence
    - Get Jira Tickets (EC-JIRA GetIssues)
    - Get frontend changelogs (EC-Git Clone)
    - Get backend changelogs (EC-Git Clone)
  - Quality Checks
    - Run Code Quality and Security Scan (EC-SonarQube Get Last)
    - Run Image Security Scan (Command)
  - Verify Feature Flag Experiment in FM (EC-FeatureFlags Get Experim)
  - Update JIRA tickets (EC-JIRA UpdateIssues)
- Quality Assurance (3 Tasks, 2 Exit Gate Rules):**
  - Deploy to QA (Deployer)
  - Enable Feature Flags for QA Env (EC-FeatureFlags Create or Up)
  - Run Functional Test (Command)

Exit Gate Rules:

  - 75% Passing QA Tests (Custom)
  - No P1s & P2s Issues (Custom)
- Pre-Production (5 Tasks):**
  - Deploy to Pre-prod (Deployer)
  - Enable Feature Flags for Pre-prod (EC-FeatureFlags Create or Up)
  - Run SIT Tests (SIT testing)
  - Manual validation (Manual)
  - Create ServiceNow ticket (EC-ServiceNow CreateChang)
- Production (5 Tasks):**
  - Deploy to Production (Deployer)
  - Run smoke tests (Smoke test)
  - Enable A/B Testing for new feature (EC-FeatureFlags Create or Up)
  - Determine winner of A/B test (Manual)
  - Set FF value for all users (EC-FeatureFlags Create or Up)

## **Model Pipelines:** with stages automating each task

- Manual and automated approvals between tasks (entry and exit gates).
- Flexibility and control in automation through error handling, parallel execution, automated and manual retries, concurrent execution of stages, parameter-driven automation, and more.

## **Evidence Links**

centralized access directly from the pipeline run to all evidence and reports associated with the stage: such as test and security scan results, build jobs, and changelogs.

## **Release Calendar**

aggregated view of all releases spanning ongoing and planned activities helps to identify where there may be conflicts between releases.

## **Release Portfolio View**

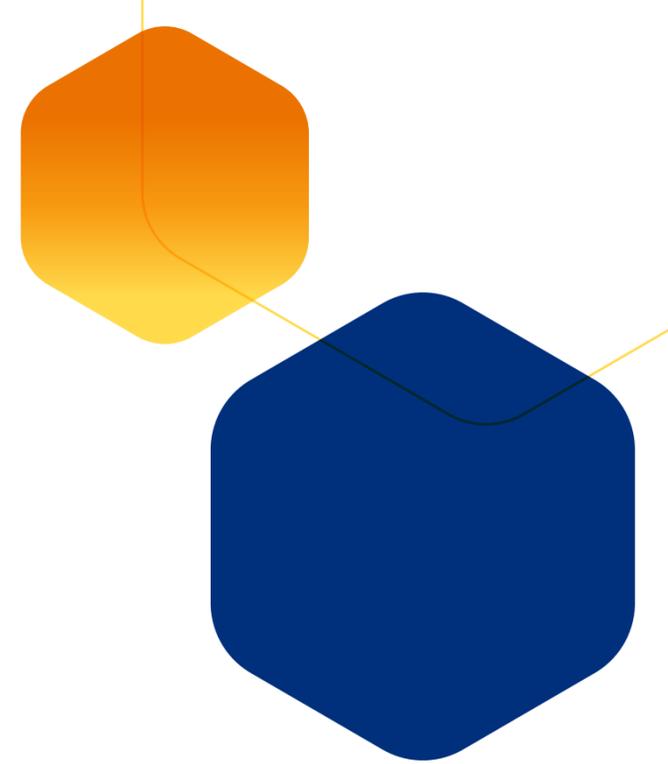
top-down visibility into the dependency hierarchy across all releases and pipelines. This view provides the status of all related releases at a glance, with drill-downs into dependencies to instantly understand progress and identify delays.

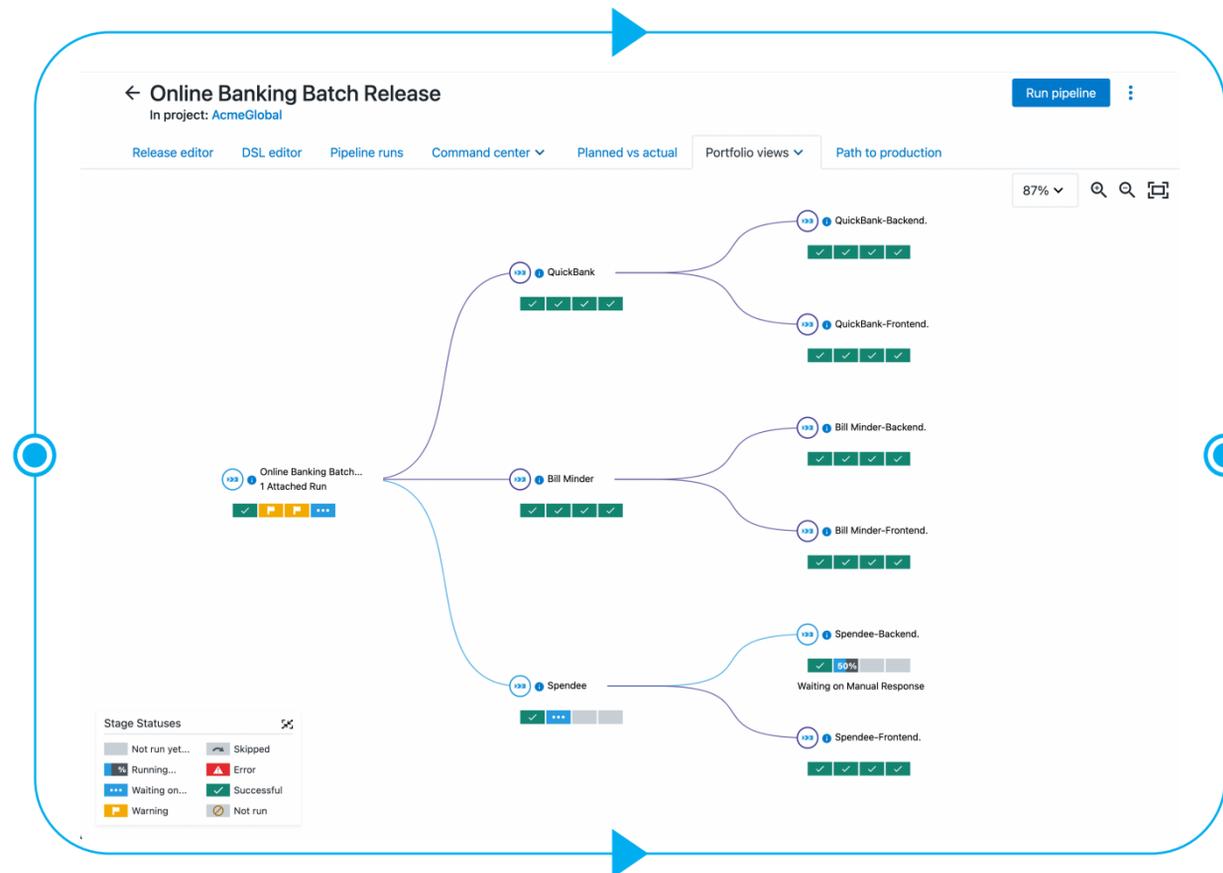
## **Planned vs Actual View**

analyze which phases of the release take the most time -- with breakdowns into manual/idle vs. automation to identify bottlenecks and improve release efficiency.

## **Release Command Center Dashboard**

provides a centralized view of all release activities and processes. Integrations pull critical metrics from the tools used across the end-to-end process.





## Path to Production Capability

review the entire payload delivered from the 1st stage (e.g., DEV) through the final stage (e.g., PROD). This feature includes the ability to click into the payload and understand the snapshot version, as well as the actual component(s) name(s) and version(s) delivered through the release.

## DSL Editor View

review the entire release pipeline in Domain Specific Language (DSL)/Groovy/ YAML and then store these pipelines in GitHub, Bitbucket, or another repository for testing and verification purposes.

## Dynamic and data-driven parameters

enable customers to create compelling end-user experiences to gather inputs and reduce errors.

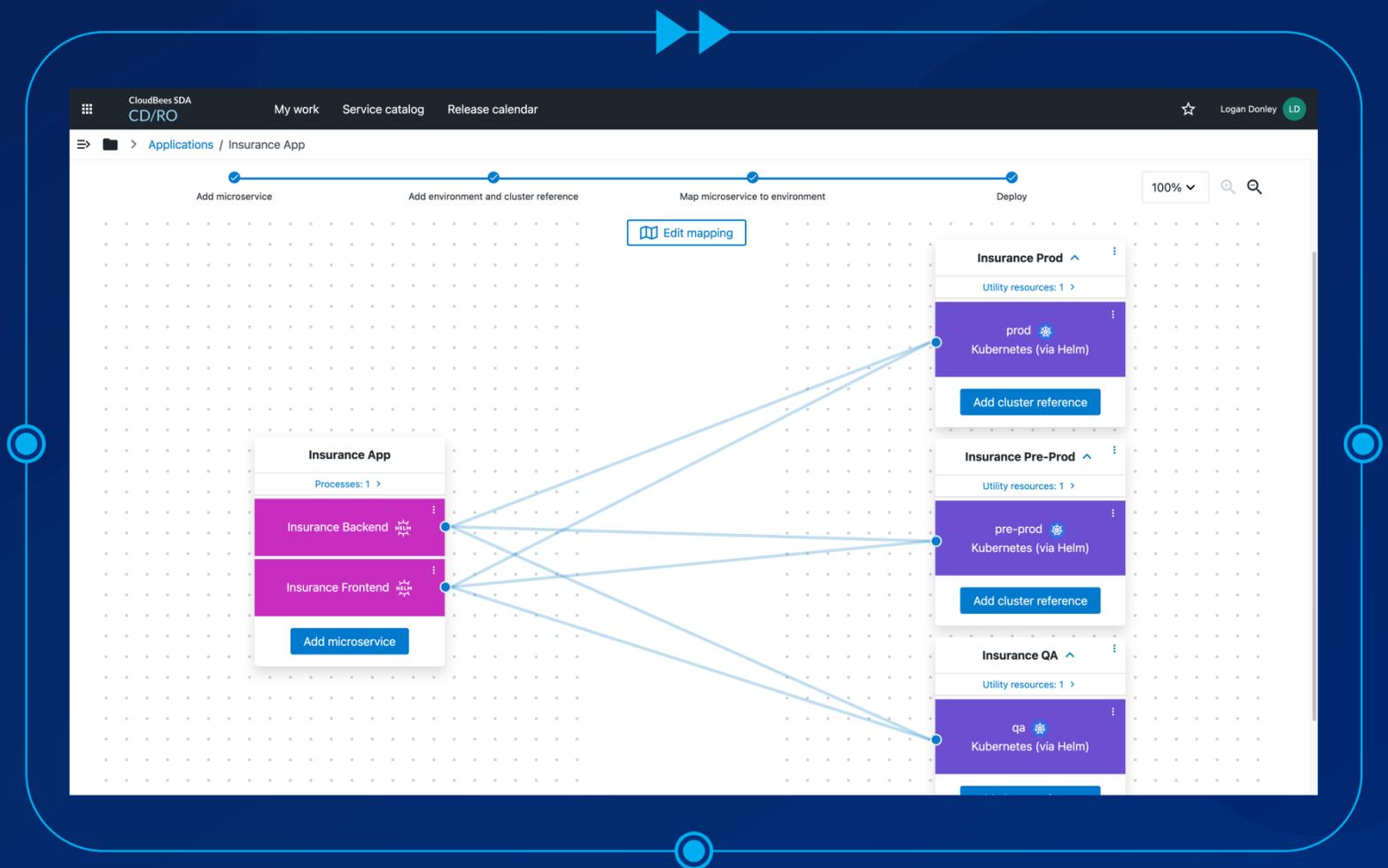
## Triggers

automatically kick off releases and pipelines on code commits using polling and webhooks.

# Deployment Automation

Model applications and environments and automate predictable deployments of the same application(s) across your environments at scale

- Flexible and customizable deployment processes can call any automation in the system and leverage our full plugin suite: support manual intervention, parallel steps, branching, and error handling.



## Deployment Features

- **Smart Deploy** – option to only deploy to changed artifacts.
- **Artifact Staging** – limit deployment time by staging artifacts on the environment in advance.
- **Master Components** – reuse best practices across your applications. Use by reference or by copy.
- **Snapshots** – capture the artifacts deployed to an environment and the exact process used to deploy them for consistency across environments.
- **Automated Rollback** – leverage integrations with APM vendors to analyze and roll back deployments to the last known good version when there is an error.
- **Advanced deployment strategies** – eliminate downtime and deploy with confidence across your environments and application types, with capabilities such as:
  - Blue/Green Deployments
  - Canary Deployments
  - Dark Launch Deployments
  - Hot Deployments
  - Partial Deployments
  - Combination Deployments (e.g., Canary Deployment + Dark Launch Deployment)
- **Third-Party Deployment** – manage deployments across third-party tools (Jenkins, Argo CD, Ansible Tower, etc.) directly within CloudBees CD/RO. Provide enhancements to these tools through automated repeatability, process visibility, environment inventory and reservation management, and analytics.

## Environment management

- Dynamic Environments – dynamically spin up and configure environments at deployment time, and automate destroying the environment when it is no longer needed.
- Environment inventory – visibility into what is deployed where with detailed traceability into exact component versions and processes used for deployment.
  - Inventory comparison to compare inventory in multiple environments and detect environment drift.
- Environment Reservation – equips the operations team with capabilities that allow it to impart control on environments or environment tiers, making application deployments more reliable and predictable. This includes the ability to reserve environments and create blackout periods to allow exclusive deployment to an environment and prevent conflicting activities.

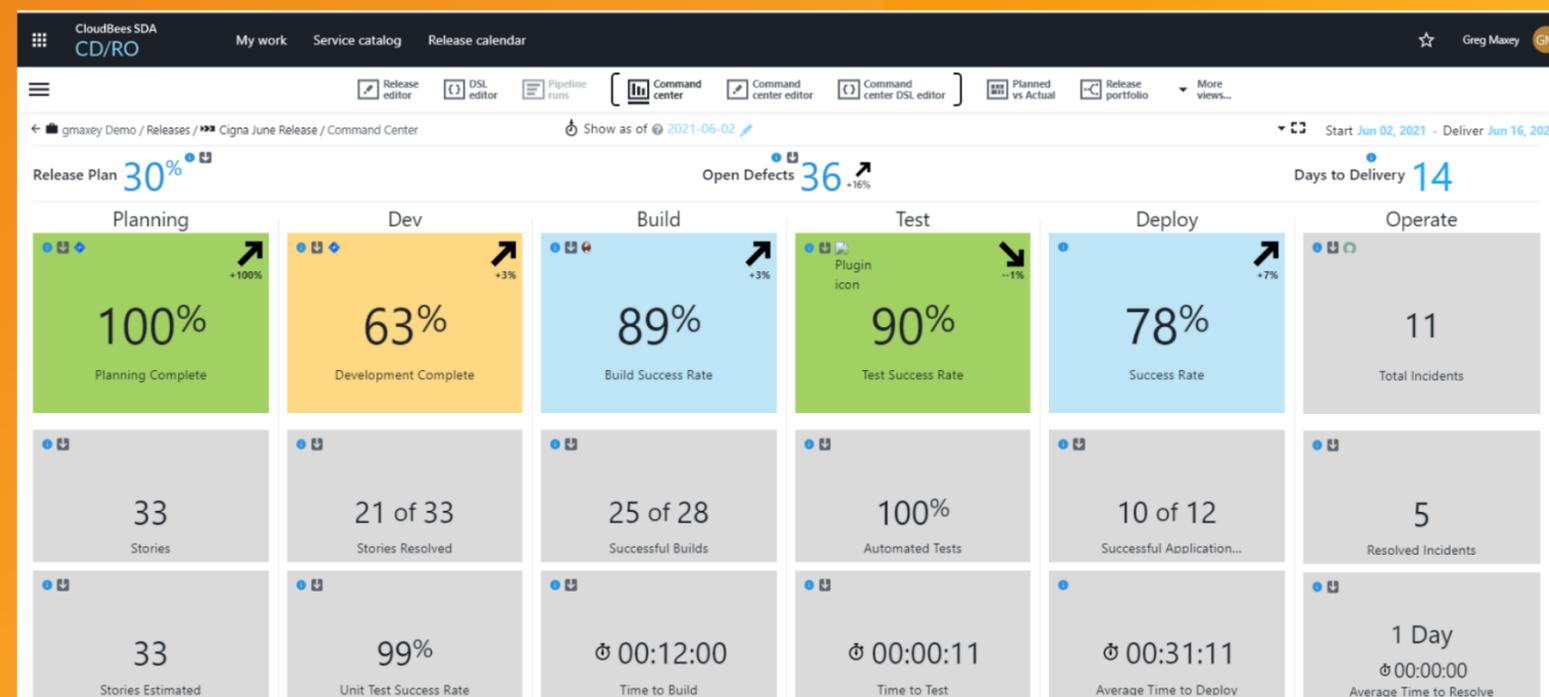
## Microservices modeling and deployment

native Helm and YAML support for modeling container-based microservices and clusters where they are deployed. Deep integrations with Kubernetes and all other leading container platforms to deploy your containers anywhere.

The screenshot displays a 'Compare' interface for a 'microservice-application' within a 'cluster-environment'. It compares two configurations side-by-side. The left configuration shows a 'dev-microservice' with artifact version '11.22' and a 'qa-microservice' with artifact versions '1.2 / 1.2.152'. The right configuration shows the same 'dev-microservice' but with artifact versions '10.2.5 / 10.2.5-beta' and the same 'qa-microservice' with artifact versions '1.2 / 1.2.152'. A 'Deploy Microservice Process' step is visible in both configurations. The interface includes navigation elements like '← Environments / Compare', 'microservice-application', and 'Compare: 2'.

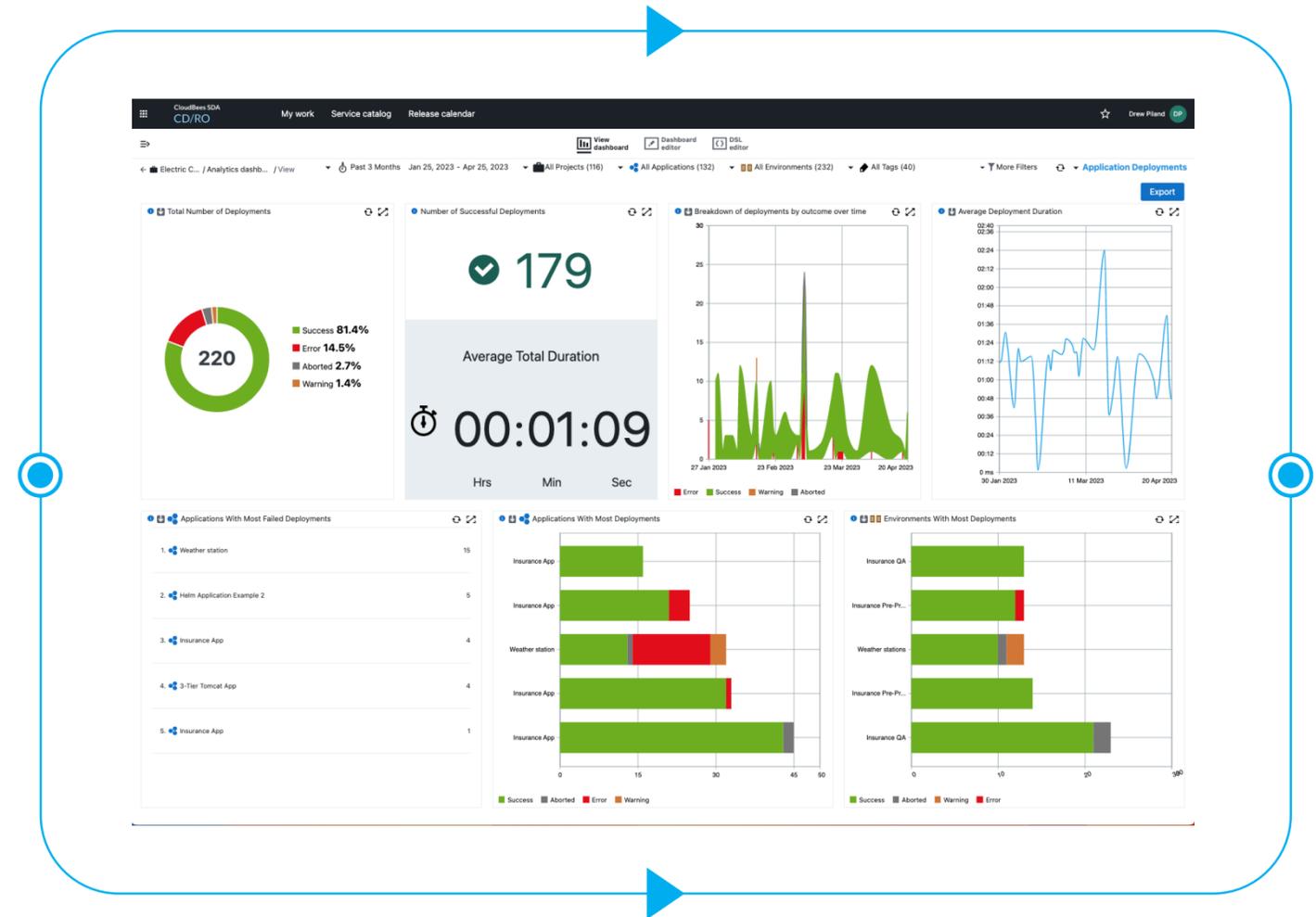
# Analytics Dashboard

Gain unprecedented insight into all your releases, pipelines, deployment, CI builds, and much more, all from a unified reporting and analytics dashboard



## Out of the box dashboards showing key metrics and patterns:

- Releases
  - Release Command Center
  - Microservices Deployments
  - Application Deployments
  - DORA Dashboard
  - Continuous Integration
  - Code Commit Trends
  - Workload Insights (Jenkins/CBCI Dashboard)
  - Plugin Usage (Jenkins/CBCI Dashboard)
- Environment inventory – visibility into what is deployed where with detailed traceability into exact component versions and processes used for deployment.
  - Persona dashboards – present only what is relevant for specific roles to help eliminate the noise.
  - CloudBees Analytics is based on Elastic Search – providing unprecedented insight into all reports contained and created within our system. Additionally, it can be leveraged to combine reports, searching, and rendering from objects outside the system. Customers can take full advantage of this powerful engine to report on any metrics and KPIs within their projects.



# Self-Service Catalog

Onboard and expose key DevOps capabilities to your entire team or organization at the click of a button

 <b>Migrate configurations</b> Migrate your old plugin configurations into plugin configuration objects. <a href="#">More...</a> <a href="#">Migrate</a>	 <b>Export DSL</b> Manage your automation as code in DSL. <a href="#">More...</a> <a href="#">Export</a>	 <b>AWS Dynamic Environment</b> Create a new environment by dynamically provisioning and configuring VMs on Amazon EC2. <a href="#">More...</a> <a href="#">Provision</a>	 <b>Multi-stage CD Release</b> Create a sample Release with Dev, QA and Production pipeline stages and approval gates to manage the path to production from Dev to Production. <a href="#">More...</a> <a href="#">Create</a>	 <b>Multi-stage CD Pipeline</b> Create a Pipeline with Dev, QA and Production pipeline stages with approval gates to manage path to production from Dev commit to Production. <a href="#">More...</a> <a href="#">Create</a>
 <b>Ansible tower model</b> Create a sample Ansible Tower application. <a href="#">More...</a> <a href="#">Create</a>	 <b>Artifact Usage and Remediation</b> Generate a report on where an artifact version is being used across the system, including environments, applications and master. <a href="#">More...</a> <a href="#">Execute</a>	 <b>Import DSL</b> Manage your automation as code in DSL. <a href="#">More...</a> <a href="#">Import</a>	 <b>Database Deployment to MySQL</b> Execute SQL script from file for MySQL. <a href="#">More...</a> <a href="#">Execute</a>	 <b>Import DSL from Git</b> Manage your automation as code in DSL. <a href="#">More...</a> <a href="#">Import</a>
 <b>Database Deployment to Oracle</b> Execute SQL script from file for Oracle. <a href="#">More...</a> <a href="#">Execute</a>	 <b>Database Deployment to SQL Server</b> Execute SQL script from file for SQL Server. <a href="#">More...</a> <a href="#">Execute</a>	 <b>Deployment Package Manager</b> Deployment package manager allows deployment of Java or .Net application quickly using CloudBees CD. <a href="#">More...</a> <a href="#">Create/Deploy</a>	 <b>IIS Application Model</b> Create a simple .Net application model to deploy to Microsoft IIS server. <a href="#">More...</a> <a href="#">Create</a>	 <b>JBoss Application Model</b> Create a simple Java application model to deploy to Redhat's JBoss Application Server. <a href="#">More...</a> <a href="#">Create</a>
 <b>Spring Boot Application Model</b> Create a simple Java application model to deploy to Spring Boot Application Server. <a href="#">More...</a> <a href="#">Create</a>	 <b>Tomcat Application Model</b> Create a simple Java application model to deploy to Tomcat Application Server. <a href="#">More...</a> <a href="#">Create</a>	 <b>WebLogic Application Model</b> Create a simple Java application model to deploy to Oracle's WebLogic Server. <a href="#">More...</a> <a href="#">Create</a>	 <b>WebSphere Application Model</b> Create a simple Java application model to deploy to IBM's WebSphere Application Server. <a href="#">More...</a> <a href="#">Create</a>	

## “Self Service” and ‘consumerizing’ DevOps:

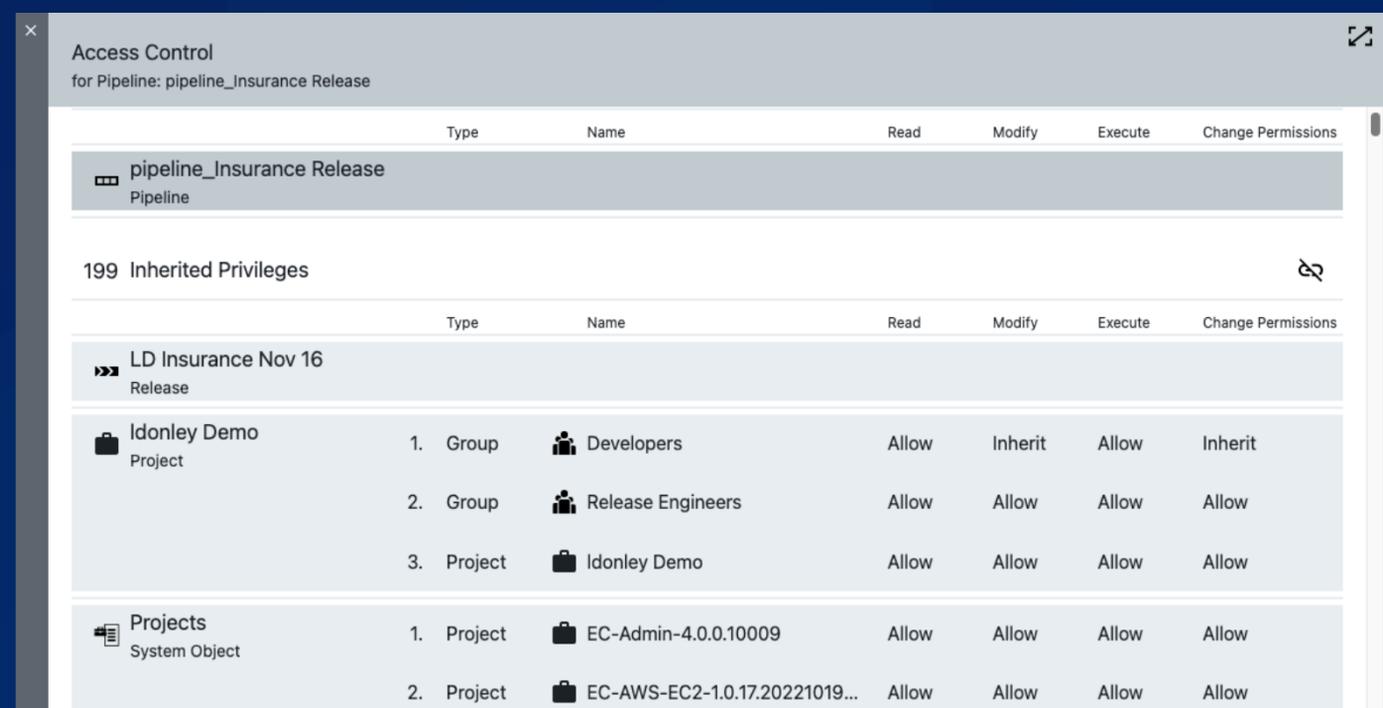
CloudBees CD/RO has differentiated capabilities like Self-Service Catalog (SSC) that dictate how end users can consume DevOps in their organization. The CloudBees CD/RO Self-Service Catalog empowers users to ‘self-serve’ to drastically reduce onboarding of our solution and increase usage. It significantly improves the success rate of DevOps projects, and there is no ‘tissue rejection’ by end users.

- Reuse best practices – quickly get started with numerous DevOps use cases with out-of-the-box catalog items.
- Customized catalog items – share automation across the organization. There are quite a few examples of the benefits of the Self Service Catalog (SSC) that include:
  - Need to POC software technology with another vendor? Leverage the CloudBees CD/RO Self-Service Catalog to spin up and tear down a complete Demo and POC environment.
  - Need to build a critical Dashboard and Report for your boss? Leverage the CloudBees CD/RO Self-Service Catalog to create any dashboards and reports needed to validate and show metrics!
  - Need to check system specs on a machine or series of machines? Leverage the CloudBees CD/RO Self-Service Catalog to execute these commands and collect the results you require
  - Need to enable team best practices templates for building applications, deploying applications, creating and executing pipelines and releases? Leverage the CloudBees CD/RO Self-Service Catalog to do all this and much more!
- One-stop-shop – centralized location for IT organizations to manage execution, creation, provisioning, discovery, and automation. Any company can create and customize self-service catalogs, and they can make them exposed to one person, a team of people, an entire department, or the entire company!

# Securing the SDLC

Empowers developers to create great code to drive value for their end customers while also providing the management tools needed to ensure a streamlined, secure, and governed process

- Fine-grained access controls – every object within CD/RO can have access control lists to view, run, and modify. Access is managed hierarchically for scalability and ease of management, ensuring users only have access to the objects they are authorized to view to help limit the blast radius.



Access Control for Pipeline: pipeline_Insurance Release						
Type	Name	Read	Modify	Execute	Change Permissions	
pipeline_Insurance Release Pipeline						
199 Inherited Privileges						
Type	Name	Read	Modify	Execute	Change Permissions	
LD Insurance Nov 16 Release						
Idonley Demo Project	1. Group	Developers	Allow	Inherit	Allow	Inherit
	2. Group	Release Engineers	Allow	Allow	Allow	Allow
	3. Project	Idonley Demo	Allow	Allow	Allow	Allow
Projects System Object	1. Project	EC-Admin-4.0.0.10009	Allow	Allow	Allow	Allow
	2. Project	EC-AWS-EC2-1.0.17.20221019...	Allow	Allow	Allow	Allow

- **Single Sign-On (SSO)** – delivers critical SSO capability, which in turn opens up additional use cases like approving from email and collaboration tools (such as Slack). CloudBees CD/RO supports this through the following protocols: SAML, OpenID Connect, and Kerberos.
- **Self-service catalog (SSC)** – promotes the reuse of best security practices with immutable components.
- **Entry & Exit gates** – ensure only good code makes it into production, and thus, dangerous/unsecured/unrecognized code should be shut down.
- **Traceability of the objects** – helps enforce governance to show that the agreed-upon process was followed.
- **Audit-ready pipelines** – capture all actions/evidence in real-time. As a result, teams always know the exact status of a release; and can generate single-click audit reports.

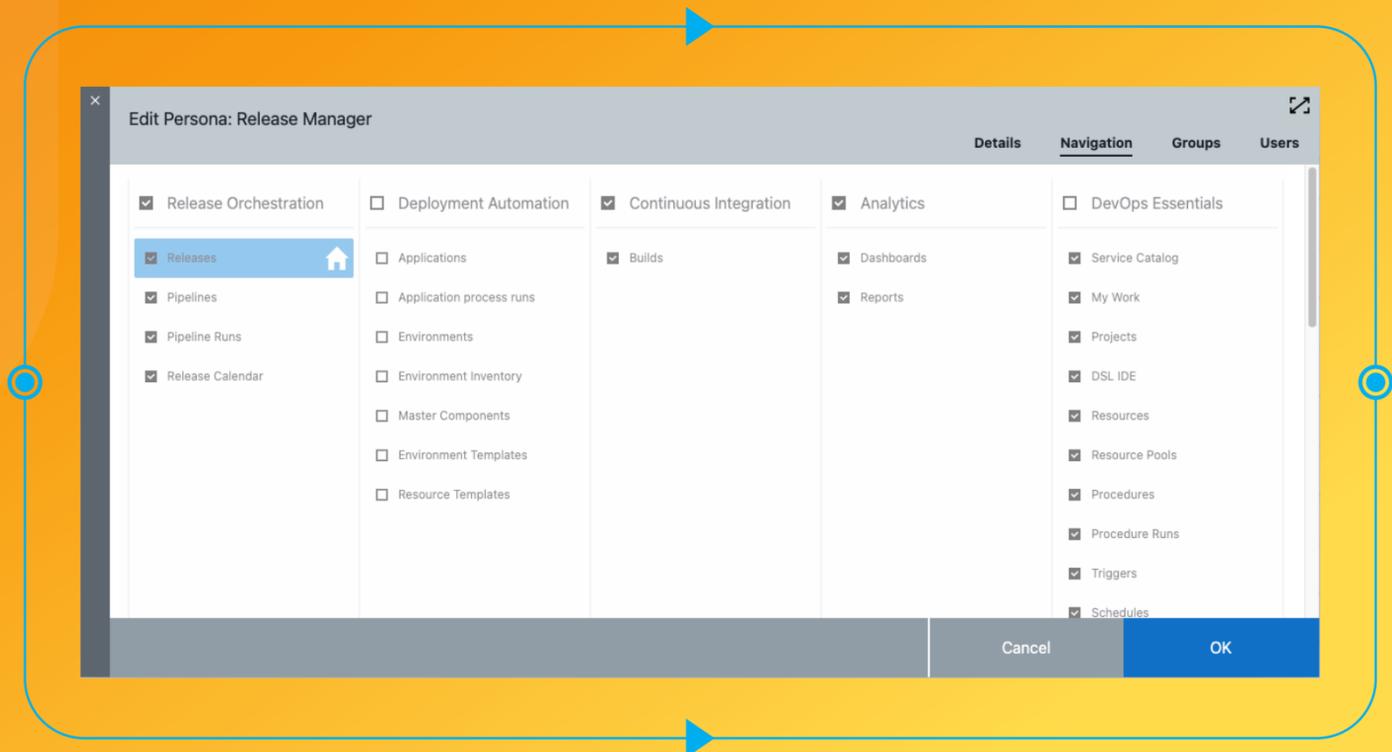
The screenshot displays the 'Payments API v3.12 audit report' in the CloudBees CD/RO interface. The report is marked as 'Successful' and started on Feb 08, 2023. It provides a detailed view of the pipeline run, including project and release information. The main part of the report is a table of evidence items, categorized by stage.

Stage	Evidence name	Evidence link
Release Readiness	JIRA Queried Issues Report	Download Report
Release Readiness	SonarQube Analysis (Run Code Quality and Security Scan)	Click here for Code Quality Report
Release Readiness	Feature management flags:	Feature Management flags report
Release Readiness	JIRA Updated Issues Report	Download Report
Quality Assurance	Insurance Frontend	link
Quality Assurance	Updated flag:	Flag report
Pre-Production	Insurance Frontend	link
Pre-Production	Updated flag:	Flag report
Pre-Production	ServiceNow Change Request:	Go to CHG0030818 on service-now.com
Production	Insurance Frontend	link
Production	Updated flag:	Flag report

# End User Personalization

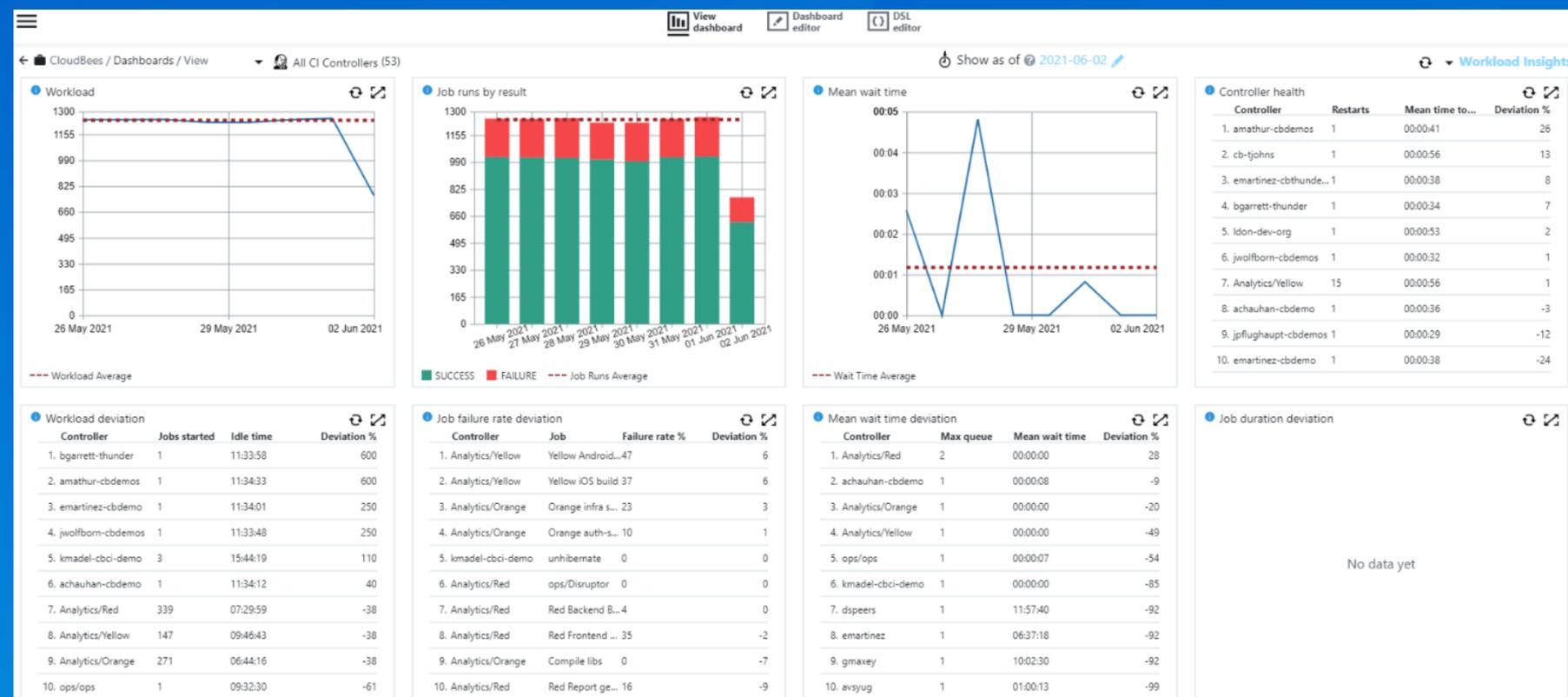
Simplify the user experience

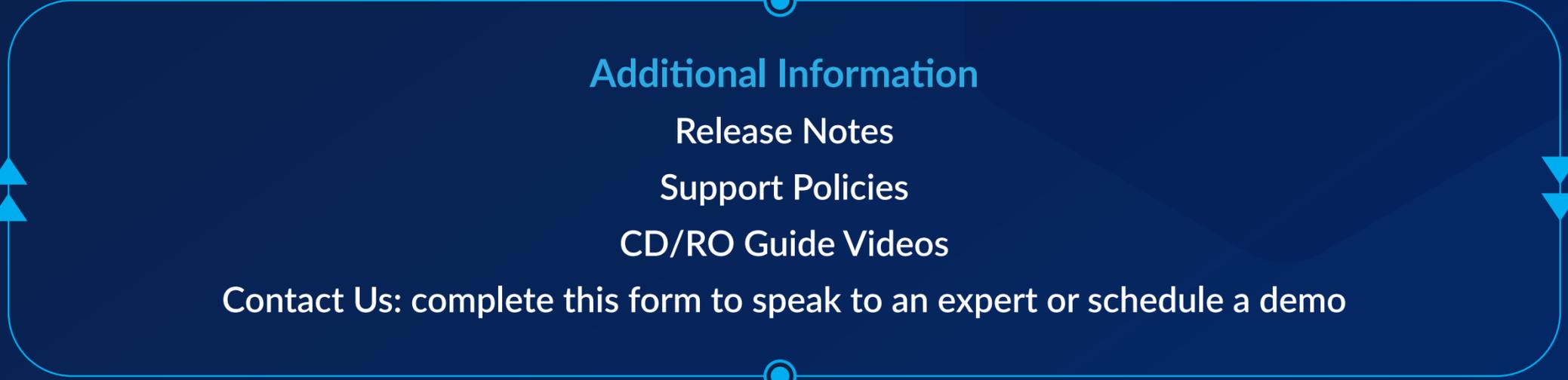
- **Persona-Based Navigation:** personas in CloudBees CD/RO allow users to create “tailored views” and navigation menus for each DevOps role (e.g., Approvers, DevOps Team, Release Managers, IT Operations, Application Support) so that users see only the things their role demands. Personas help separate duties for all DevOps roles without requiring multiple products.
- **Personalize user experience:** enables users to personalize their workflows through capabilities such as:
  - “MyWork” (for quick access to all the approvals waiting for a user)
  - Favorites (for quick access to most accessed pages)
  - Homepage personalization (to control the landing page when a user first logs into CloudBees CD/RO)



# CI Integration

- Native CD/RO ability to select and kick off CloudBees CI pipelines
- Jenkins plugin to kick off CD/RO pipelines, releases, etc.
- CBCI analytics data pushed to CD/RO Analytics engine (Plugin usage, Workload insights)





## Additional Information

Release Notes

Support Policies

CD/RO Guide Videos

Contact Us: complete this form to speak to an expert or schedule a demo



CloudBees, Inc., 4 North 2nd Street, Suite 1270 San José, CA 95113 United States  
www.cloudbees.com • info@cloudbees.com