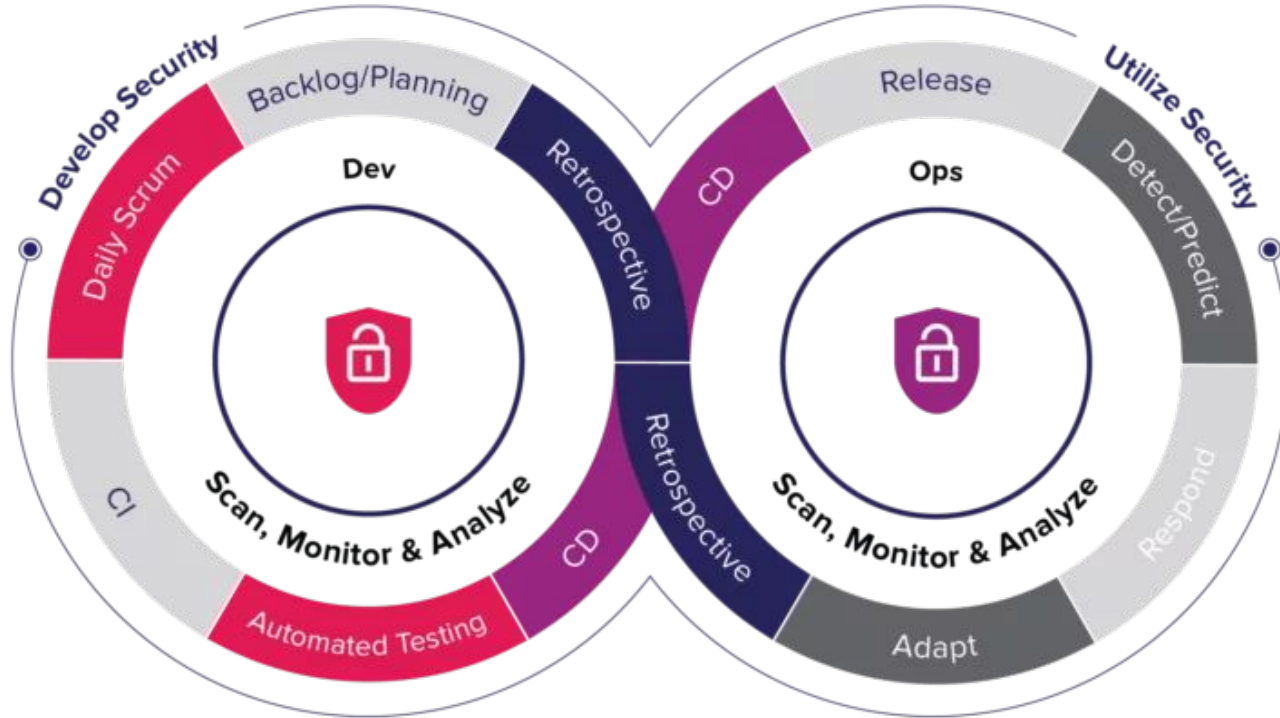




A Systems Approach to Software Supply Chain Security and Continuous ATO

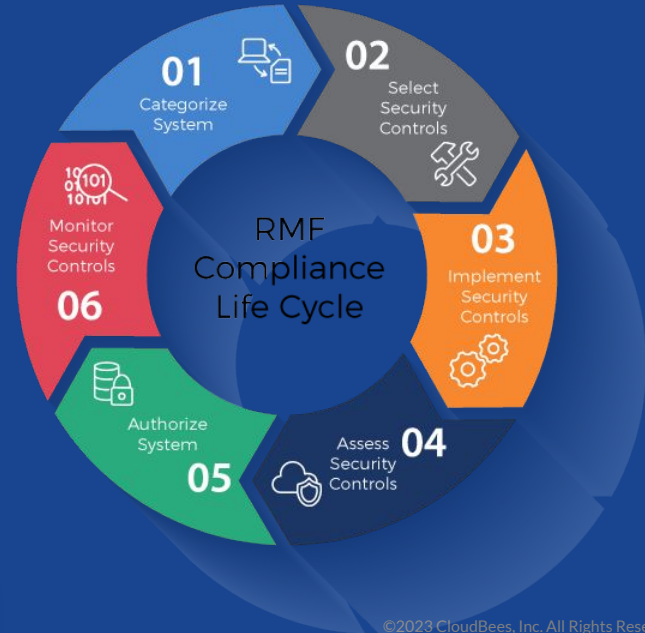
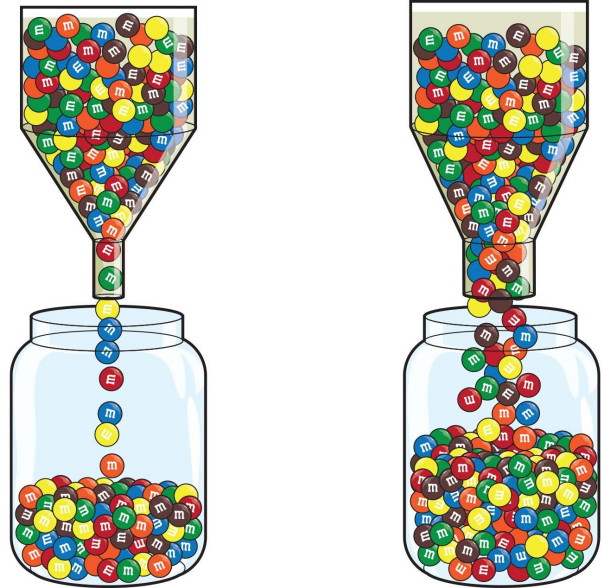
Tyler Johnson, Federal CTO

DevOps and cATO



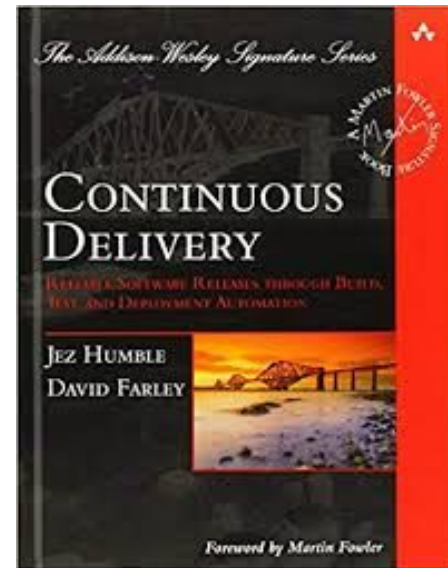
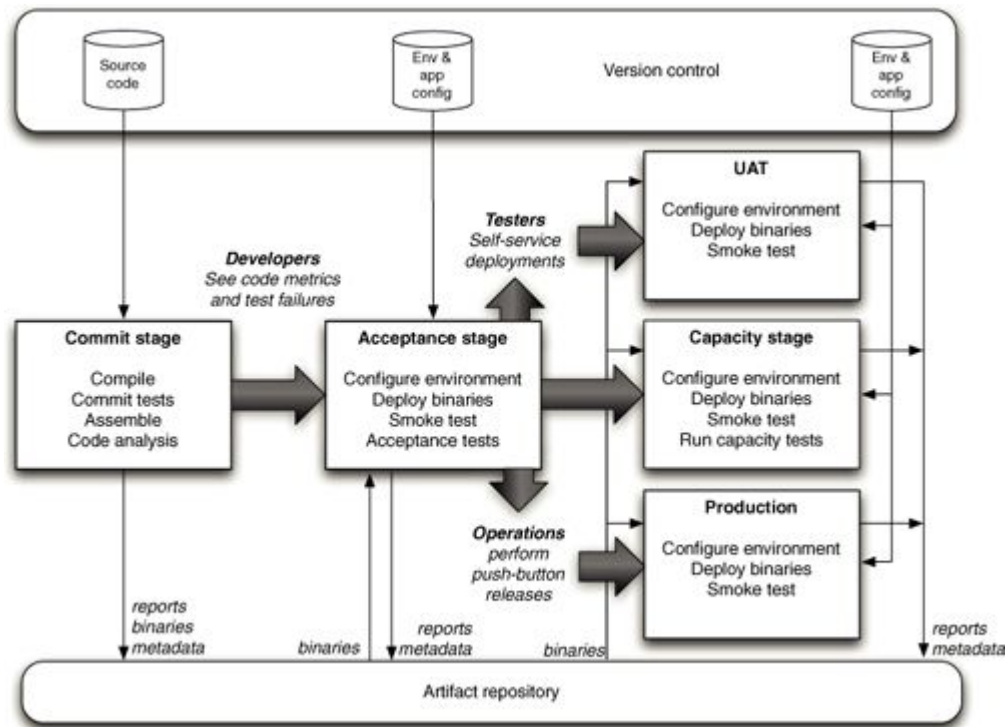
Maintain Velocity

How do agile processes maintain velocity through operations?



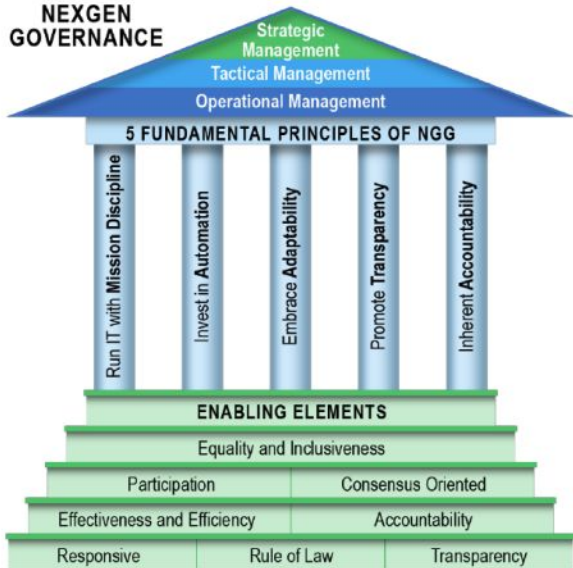
Increasing confidence in build's production readiness

Environments become more production like

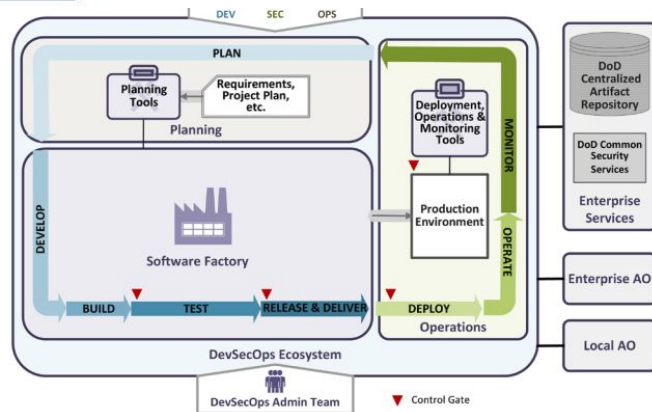
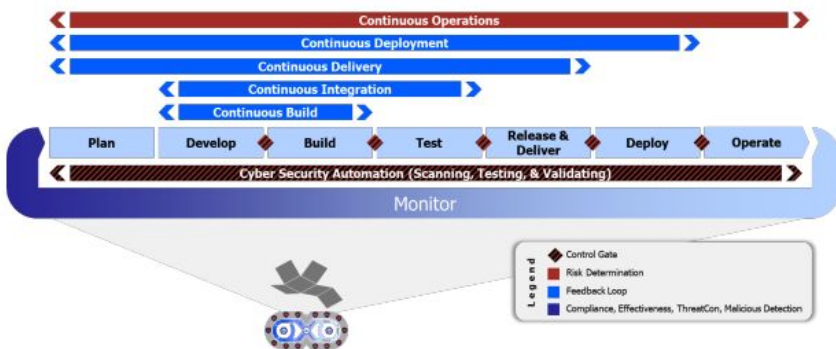
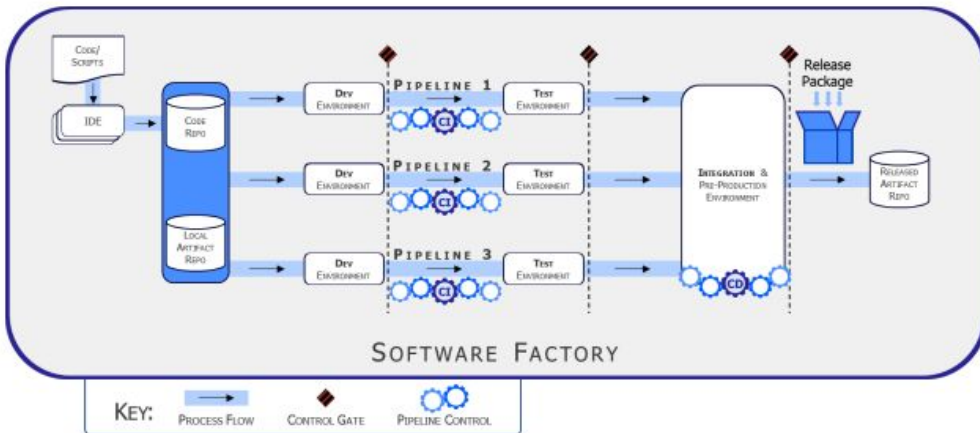


Basic Deployment Pipeline

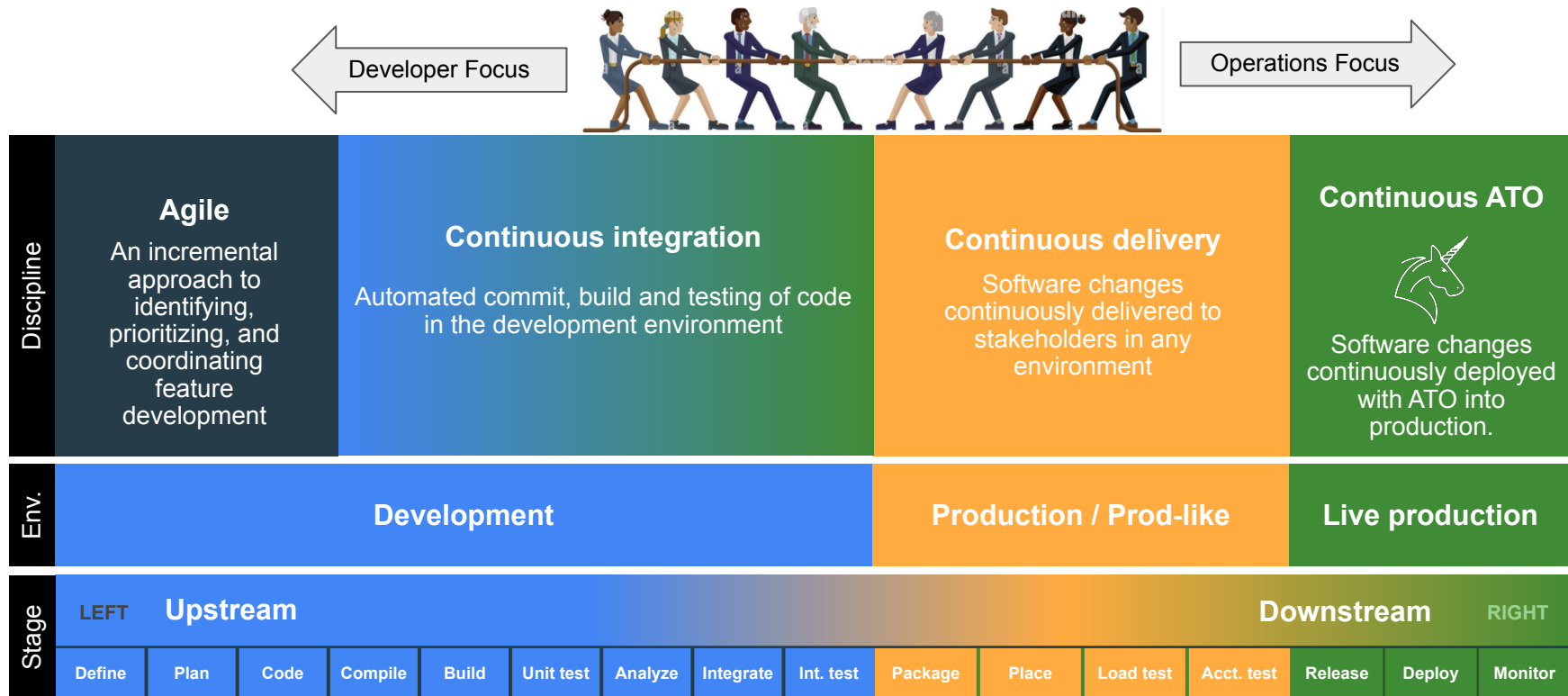
NEXGEN GOVERNANCE



HOW?



DevSecOps Maturity in context





OFFICE OF THE SECRETARY OF DEFENSE
1000 DEFENSE PENTAGON
WASHINGTON, DC 20301-1000

MEMORANDUM FOR SENIOR PENTAGON LEADERSHIP
DEFENSE AGENCY AND DOD FI

SUBJECT: Continuous Authorization To Operate (cATO)

The Risk Management Framework (RMF) establishes system cybersecurity risk. Current RMF implementation focuses on authorizations (ATO) but falls short in implementing continuous authorization. Continuous authorization has been reached. Efforts in the Department are to implement the continuous monitoring step of RMF to allow for continuous and near real-time data analytics for reporting security events is essential to the cybersecurity required to combat today's cyber threats. The purpose of this memo is to provide specific guidance to the Department to operate under a cATO state.

Current RMF implementation focuses on obtaining system authorizations (ATOs) but falls short in implementing continuous monitoring of risk once authorization has been reached.

Modeled Reference Design

Layer 1

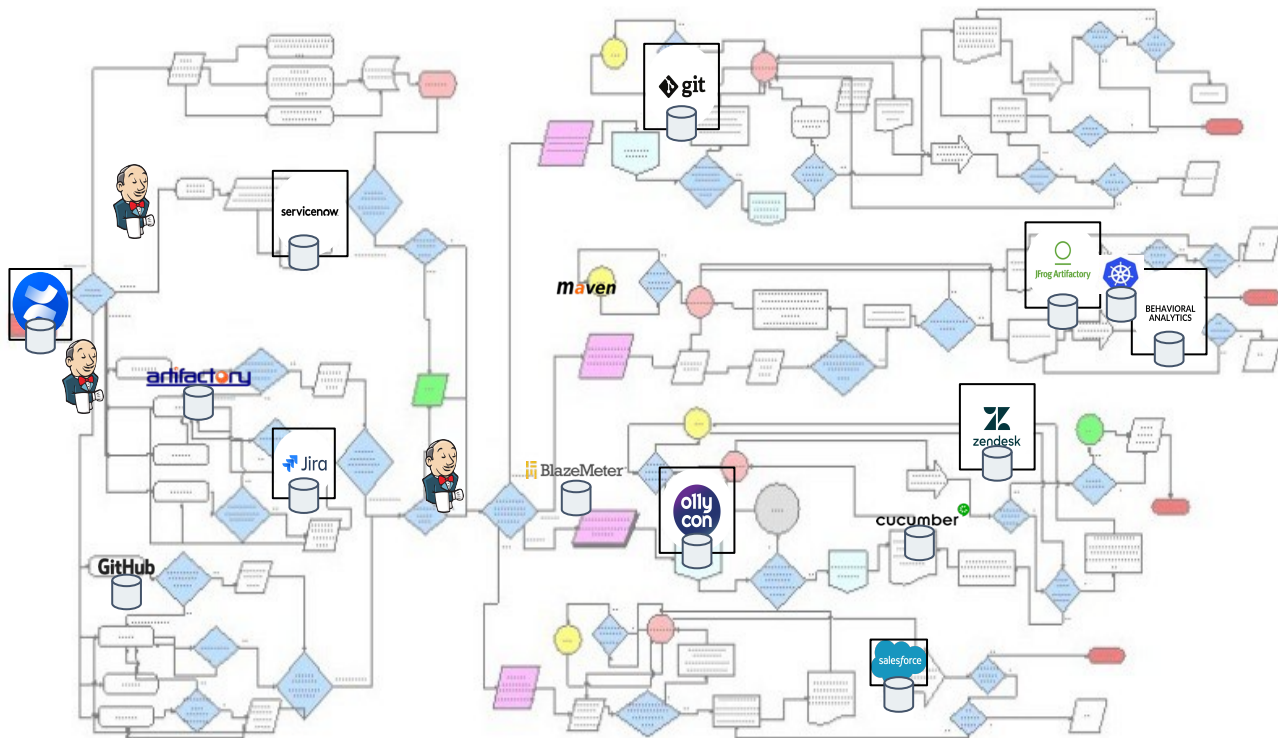
The screenshot displays a CI/CD pipeline interface with the following stages and tasks:

- Release Readiness** (9 Tasks):
 1. Enable Audit Reports (EC-AuditReports Audit Reports)
 2. Get Jira Tickets (EC-JIRA GetIssues)
 3. Get FE Change Logs (ECSCM-Git CheckoutCode)
 4. Get FE Build Details (EC-Jenkins GetBuildDetails)
 5. Get BE Build Details (EC-Jenkins GetBuildDetails)
 6. Verify Jenkins Build Status (EC-Jenkins GetBuildStatus)
 7. Quality Checks (EC-SonarQube Get Last SonarQube Metrics, EC-Run Image Security Scan)
- Quality Assurance (QA)** (4 Tasks):
 1. Deploy to QA (Deployer)
 2. Enable Feature Flag for QA Env (Command)
 3. Send Slack Message (EC-Slack Send Realtime Message)
 4. Run Functional Test (Command)
- Pre-Prod** (1 Entry Gate Rule):
 1. Wait for Release Manager Approval (Approval)
- Prod** (6 Tasks):
 1. Deploy to Pre-Prod (Deployer)
 2. Enable Feature Flag in Pre-Prod (Command)
 3. Run SIT Tests (SIT testing)
 4. Send Slack message (EC-Slack Send Realtime Message)
 5. Verify QA - Notify (Manual)
 6. Create ServiceNow ticket (EC-ServiceNow CreateChangeRequest)
- Exit Gate Rules** (3 Rules):
 1. 100% Passing Pre-Prod Tests (Custom)
 2. No P1s & P2s Issues (Custom)
 3. ATO Approval (Approval)
- Prod** (5 Tasks):
 1. Deploy to Prod (Deployer)
 2. Run smoke tests (Smoke test)
 3. Send Slack message (EC-Slack Send Realtime Message)
 4. Approve final deployment (Manual)
 5. Enable Feature Flag in Prod (Command)

Annotations and Callouts:

- Generate audit reports and build body of evidence for stakeholders & AO**: Points to the 'Quality Checks' task in the 'Release Readiness' stage.
- Steps embedded within CI/CD and release pipelines expedite approvals and ensure continuous authorization worthiness**: A general statement about the pipeline's purpose.
- AO can analyze evidence before authorizing releases to production**: Points to the 'ATO Approval' rule in the 'Exit Gate Rules' section.

The problem is... building and releasing software is complicated.



Distributed development teams can build confidently knowing integration is assured

Operations teams define templates for going to operations

Management can track metrics on all teams and progress against program closure

Operational Context

System Engineering teams can track configuration change over time

IA can aggregate evidence from DevSecOps tools and enforce compliance goals

New

Create Release from Template

Release Name: Blue and Green Release Address

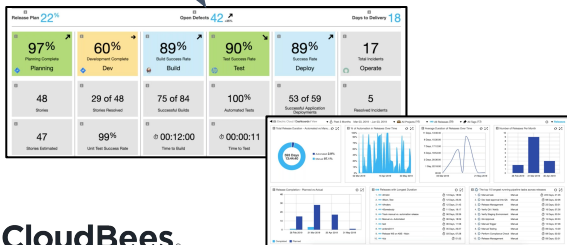
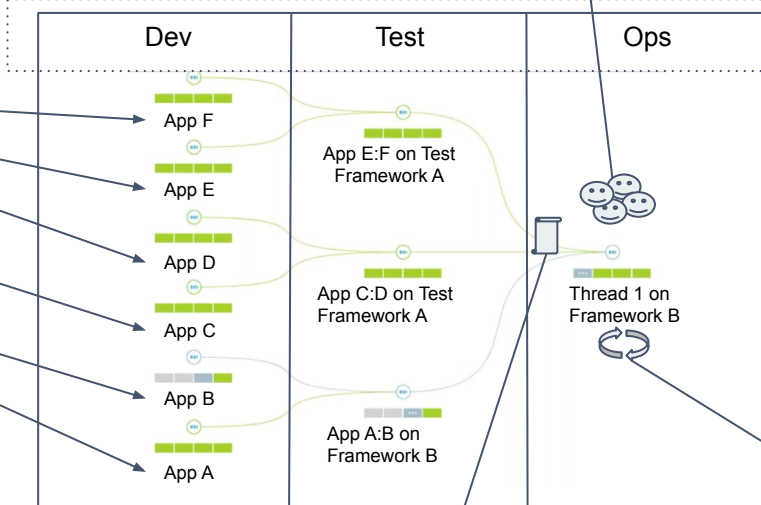
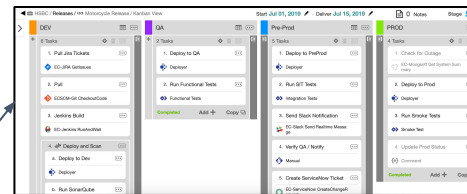
Template: Blue-Green release template

Instructions: Provide release name and template details

Acme Release

Create a new release from a list of standard release templates. Creates a full pipeline with stages, tasks, and stubs for environments and More...

Create



Approval Audit Report

General Information

Field	Value
Release Name	Motorcycle Release Pipeline
Release Start	2019-07-17T08:00:00.000Z
Release Complete	2019-08-19T08:00:00.000Z
Status	Success

Details

Step	Task	User	Status	Time	Comment
1	Build	John	Success	2019-07-17T08:00:00.000Z	Auto-approved by conditional: Server / No Critical Warnings
2	Test	John	Success	2019-07-17T08:00:00.000Z	Auto-approved by conditional: Code Coverage / 75%
3	Deploy	John	Success	2019-07-17T08:00:00.000Z	Auto-approved by conditional: 75% Passing QA Tests
4	Release	John	Success	2019-07-17T08:00:00.000Z	Auto-approved by conditional: Code / No Critical Warnings

Change History for Motorcycle Release

Task	Step	Time	User	Status	Comment
Build	1	2019-07-17T08:00:00.000Z	John	Success	Auto-approved by conditional: Server / No Critical Warnings
Test	2	2019-07-17T08:00:00.000Z	John	Success	Auto-approved by conditional: Code Coverage / 75%
Deploy	3	2019-07-17T08:00:00.000Z	John	Success	Auto-approved by conditional: 75% Passing QA Tests
Release	4	2019-07-17T08:00:00.000Z	John	Success	Auto-approved by conditional: Code / No Critical Warnings

Ongoing Visibility

Layer 2

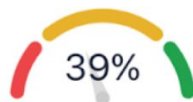


CloudBees Compliance

Search Assets, Controls, Organisations, Applications...



Compliance Health



Ineffective Controls

50

Technical Risk Position

30

Priority Open Risks



Risks Accepted

42

Control Effectiveness

Control Name	Effectiveness Rating
Avoid use of the 'root' account	20%
Ensure no root account access key exists	21%

Standards Conformity

Standard Name	Percentage of Controls Passed
FedRAMP	48%
CNSSI	31%

Important Mission Services at Risk

Service Name	Overall Risk	Confident...	Integrity	Availability
ISR	Very High	Very High	Medium	Low
Tanker Refueling	Very High	Very High	Medium	Low

Critical Business Applications at Risk

Application Name	Overall Risk	Confident...	Integrity	Availability
FMV Exploitation	Very High	Very High	Medium	Low
Pilot Availability	Very High	Very High	Medium	Low

Compensating Controls - Automated Mitigation

Mean Time to Mitigate

Layer 3



Feature flag as kill switch

- Everyone has to follow the same build method
- Kill features, not the experience



Automated roll back to known good state

- Rehearsed and integral to the system
- When a bigger hammer is required



118TH CONGRESS

1ST SESSION

S. 917

[Report No. 118-32]

IN THE SENATE OF THE UNITED STATES

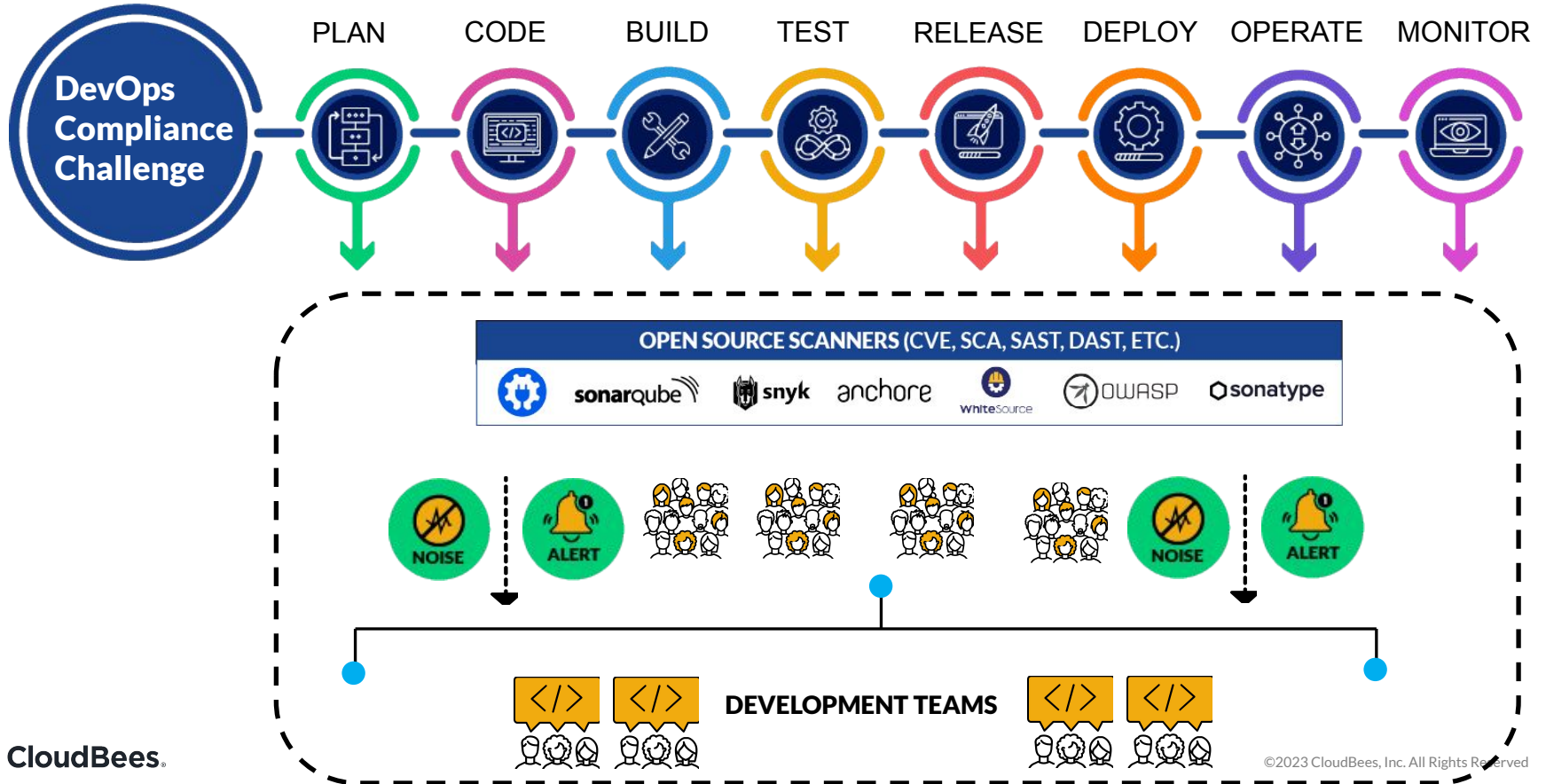
March 22, 2023

Mr. Peters (for himself and Mr. Hawley) introduced the following bill; which was read twice and referred to the Committee on Homeland Security and Governmental Affairs

Open Source Software Act



Enterprise Compliance Challenge



Real-time Events

anchore

sonatype



WhiteSource

sonarqube

OWASP

Open Source Scanners

Microsoft Azure



aws

OPA



Compliance Engine

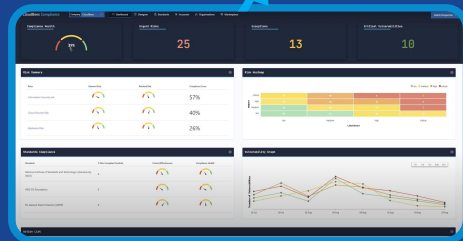
ISO27001
SOC2 HIPAA
FedRamp CCM NIST 800-53



CIS for AWS, GCP & Azure
CSA CCM NIST CSF
PCI DSS GDPR



Compliance Team



Automatically translate and dispatch compliance issues into team-focused and actionable tasks



PLAN



CODE



BUILD



TEST



RELEASE



DEPLOY



OPERATE



MONITOR

Keys to a Secure Software Supply Chain



Security/Dev Collab

- Security defines standards
- Improve developer experience
- Increase velocity
- Improve risk posture



Appropriate Tooling

- Low/No Code approach for security
- Consumable by different personas
- Enable rapid response and RT/NRT feedback



Automated Controls

- Continuously assess
- Evidence as a service
- Control gates prevent non-compliant deployment



Centralized Toolsets

- Enterprise view
- Project view
- Persona-specific dashboards
- Centralized authoritative security/compliance/defect tracking to reduce noise



Thanks!

Tyler Johnson - Federal CTO

tjohnson@cloudbees.com

(703)980-2777

<https://www.linkedin.com/in/tyler-johnson-rva/>

