Octopus Deploy

GitHub

Fast and Compliant CI/CD Pipelines in the financial industry

Bob Walker | Field CTO | Octopus Deploy

Philip Holleran | Field CTO | GitHub

Hello from Octonauts and Octocats

Bob Walker

Field CTO

Octonaut since 2018

bob.walker@octopus.com

Philip Holleran

Field CTO

GitHubber since 2015

pholleran@github.com

Outline

What we hear as the current challenges with audit and compliance

How GitHub and Octopus Deploy impact auditing and compliance

Demo of CI/CD pipeline with GitHub + Octopus

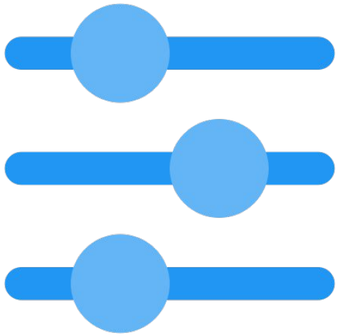
Outline

What we hear as the current challenges with audit and compliance

How GitHub and Octopus Deploy impacts auditing and compliance

Demo of CI/CD pipeline using GitHub + Octopus

What is compliance —aligning our understanding



CONTROLS

Implementing appropriate internal controls

REPORTS

The provenance - who, what, when, where, and why changes happened

AUDIT

Passing regular audits

Challenges we see with audit and compliance

✅	Reduced auditability and increased risk though a DIY CI/CD pipeline.
✅	Manual steps that reduce efficiency.
✅	Lower traceability and manual reporting.
✅	Lack of enterprise-wide standardization of processes and visibility.
✅	Poor DevEx from higher friction due to lack of trust.

Audit—the dirty word in development

SOME COMMON RESPONSES WE HEAR FROM DEVELOPERS AND ORGANIZATIONS ABOUT AUDIT!

DID I FOLLOW THE COMPANY PROCESS?

I NEED CONTROLS ON WHO CAN DO WHAT

WHY ISN'T THIS AUTOMATED?

WHERE ARE THE GUARDRAILS TO FOLLOW SO I AM COMPLIANT?

EXTERNAL AUDIT HAS MORE EYES ON IT

EVERYTHING STOPS WHEN AUDITS HAPPEN

WHAT IS THE COMPANY PROCESS?

Visibility into the "last mile" of development is difficult

STOP EVERYTHING AND GO PUT TOGETHER SOME MANUAL REPORTS!

SOX requires a granular and painstaking scrutiny

THIS IMPACTS MY SCHEDULE

WHY DON'T ALL THESE SYSTEMS WORK TOGETHER

HOW DO I REQUEST EXCEPTIONS?

I NEED EXCEPTIONS FOR ME TO COMPLETE MY WORK

THIS PROBLEM IS HAPPENING MORE FREQUENTLY

I DON'T TRUST THE PROCESS

WHY ISN'T THERE A REPORT THAT CAN CAPTURE EVERYTHING?

WAS THE INTERNAL AUDIT FINDING MY FAULT?

Outline

What we hear as the current challenges with audit and compliance

How GitHub and Octopus Deploy impacts auditing and compliance

Demo of CI/CD pipeline using GitHub + Octopus

The ideal CI/CD pipeline

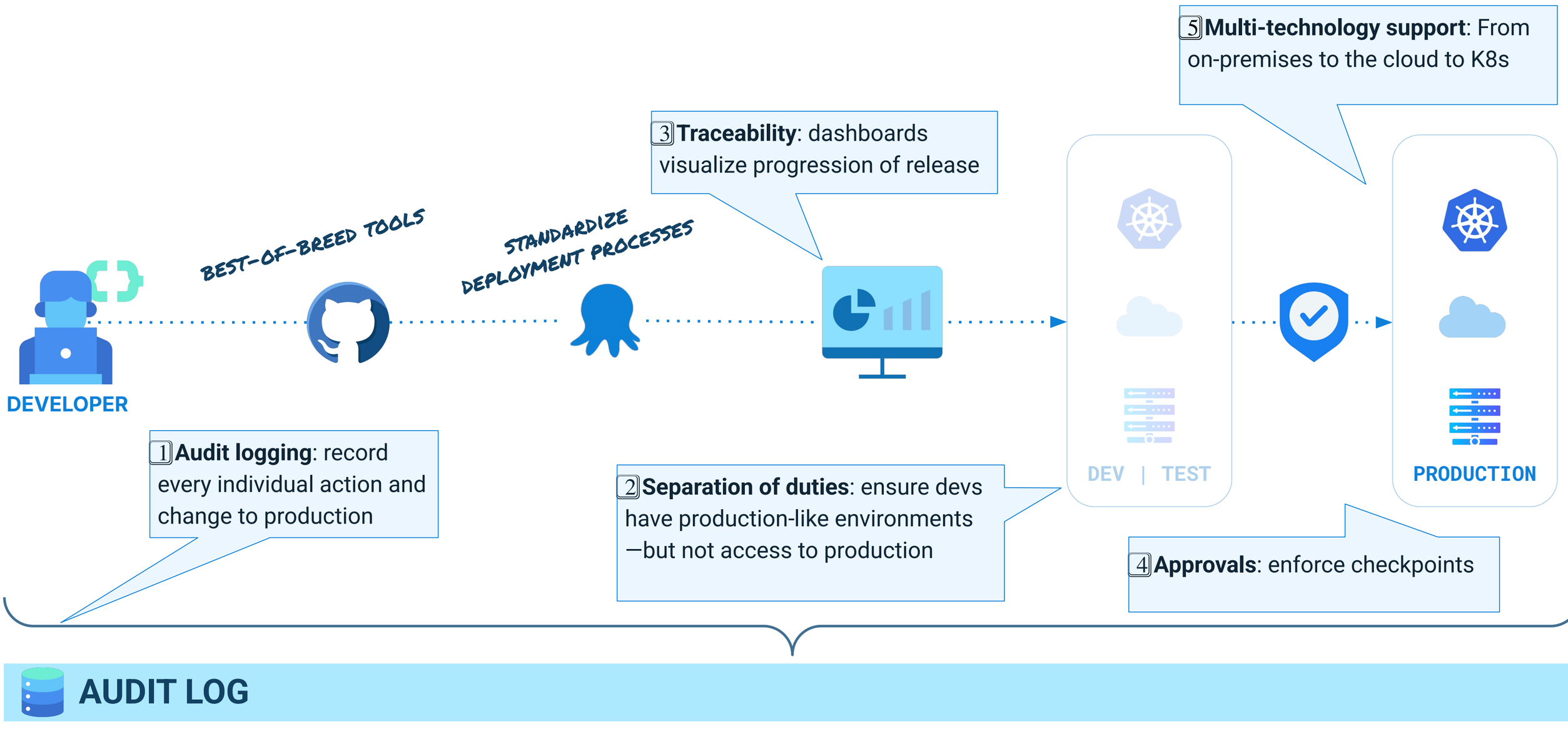
✅	Supply chain security (Vulnerability Scanning, SBOMs, etc.)
✅	All components (including database schema) is included
✅	Build once, deploy anywhere
✅	RBAC controls and separation of duties
✅	Testing in prod-like (ephemeral) environments
✅	Separate deploying new versions from releasing new functionality

More critical because of Executive Order 14028

Surprising number of people exclude the database

De-risking new features

Compliance with GitHub and Octopus Deploy



Compliant CI with GitHub Actions

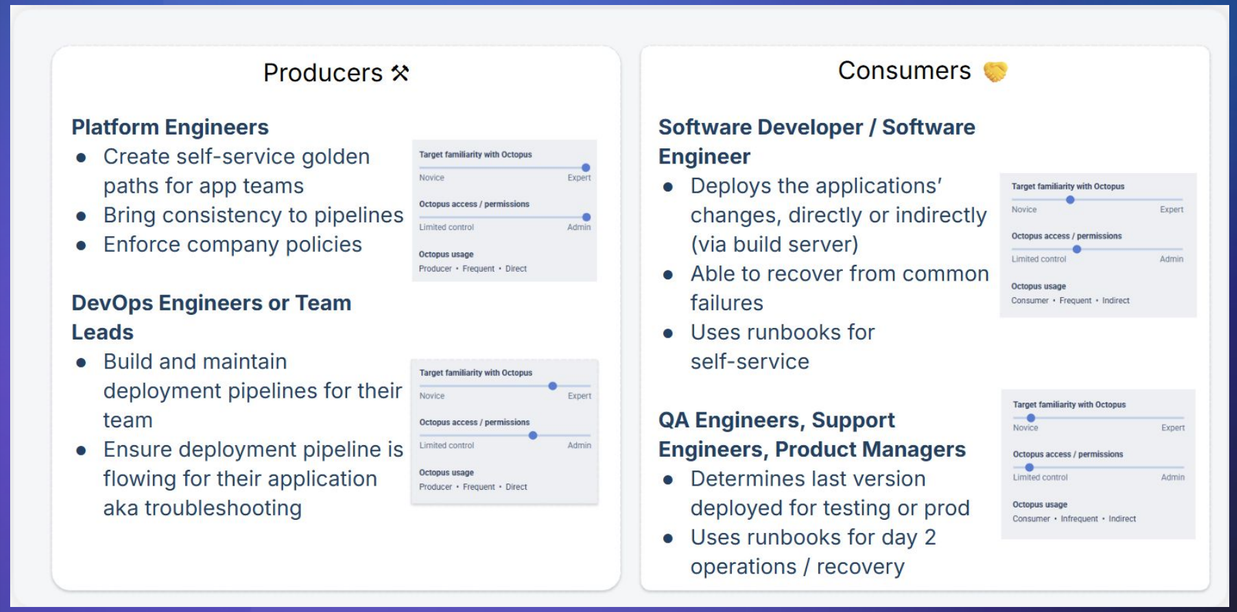
✅	**Least privilege**: Require approval for workflow changes via CODEOWNERS
✅	**Required jobs (workflows):** Repository Rules
✅	**Auditability:** via Git, Pull Request activity, audit log
✅	**Security:** GitHub Code Scanning, SLSA Level 3 & Custom Networking (VNET)
✅	**Traceability:** via artifact attestations

Platform Hub

Democratize CI/CD Pipelines

Promoting collaboration between producers and consumers and eliminate hard walls.

- **Producers** (platform engineers or team leads) are the experts in the tools used in the deployment pipeline and company policies.
- **Consumers** (developers) are the experts on their applications.
- Better collaboration between the two groups via shared tools.
- Support both ends of the spectrum from “I give you the artifact, you deploy it” to “I want to build and own my pipeline.”



Platform Hub— four key components

Process Templates

Create reusable multi-step templates to use for deployments and runbooks.

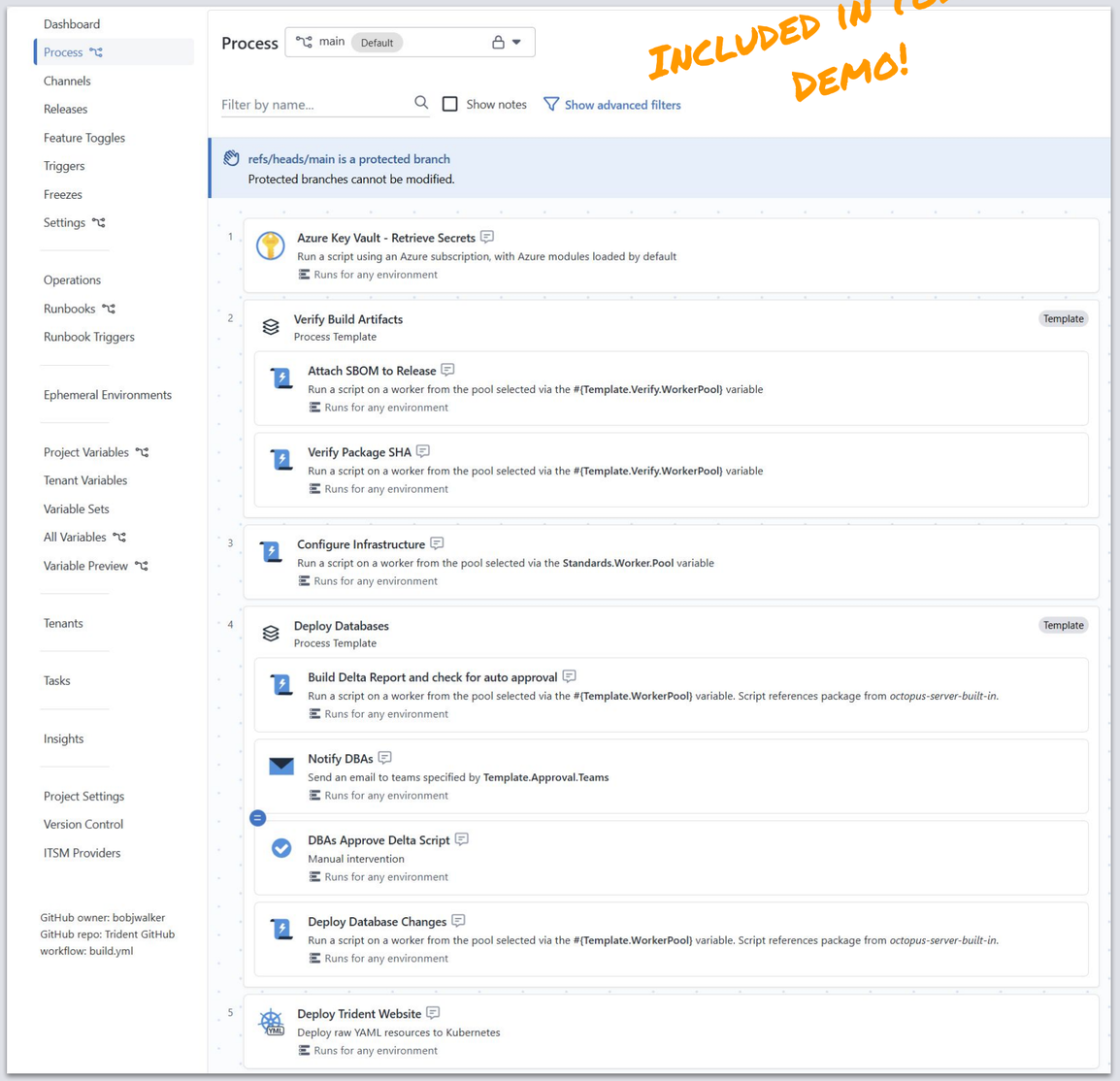
A process can be composed of one or many templates.

App teams can inject custom steps specific to their application before or after the template.

Policy Engine

Project Templates

Compliance Dashboard



Platform Hub— four key components

Process Templates

Policy Engine

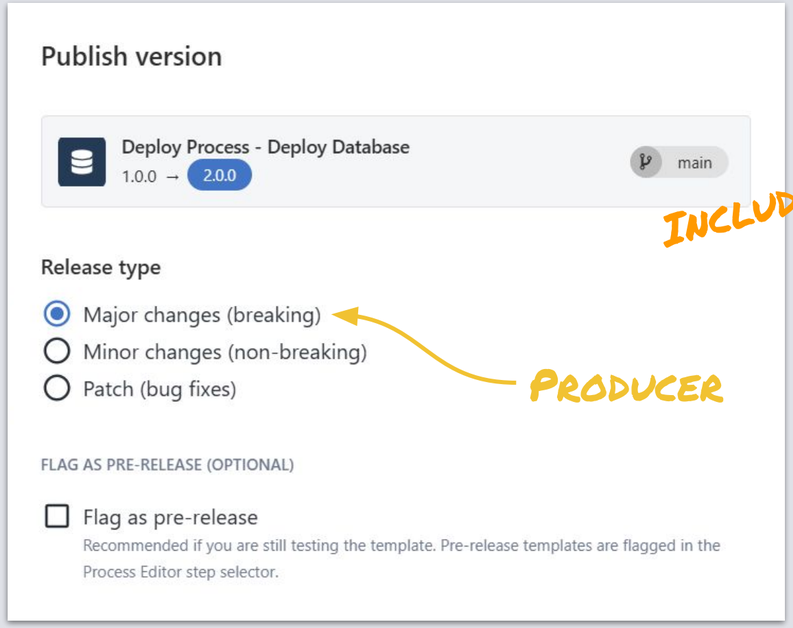
Project Templates

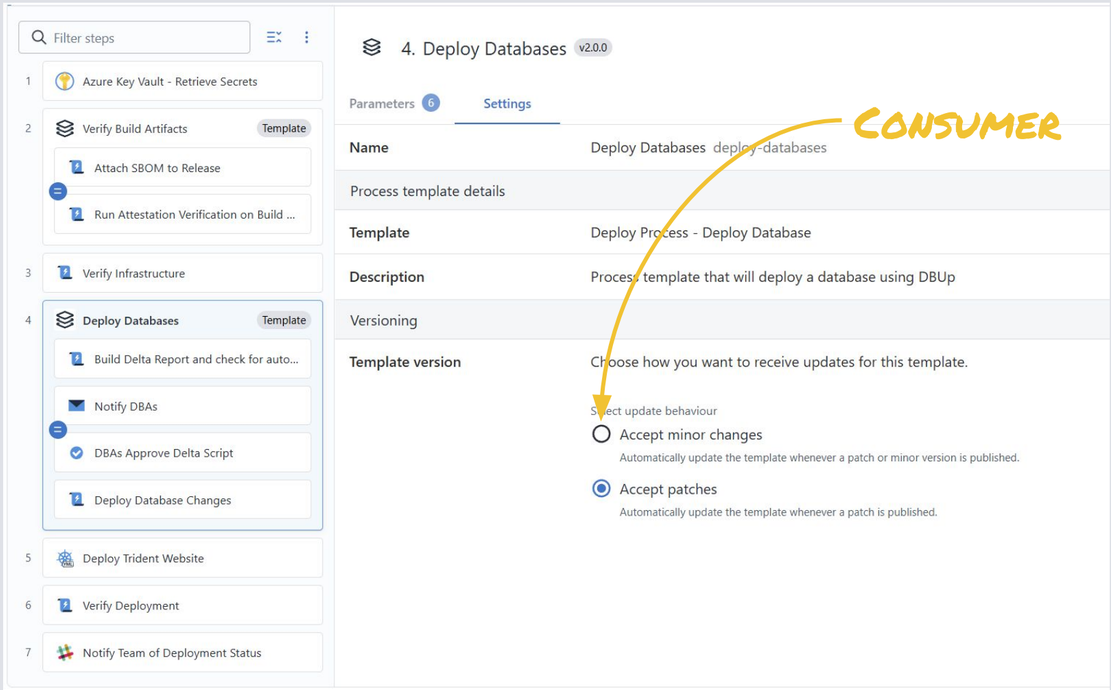
Compliance Dashboard

Versioning

Producers get an easy to use mechanism to publish major, minor, and patch changes to consumers.

Consumers can automatically accept minor changes while controlling when to bring-in breaking changes.





Platform Hub— four key components

Process Templates

☑️ **Policy Engine**

Project Templates

Compliance Dashboard

Create rules for prod vs non-prod environments.

Create deploy time rules to make ITSM, step usage, and security scans mandatory.

Teams are informed when they don’t meet a policy, either via a warning or by blocking deployments.

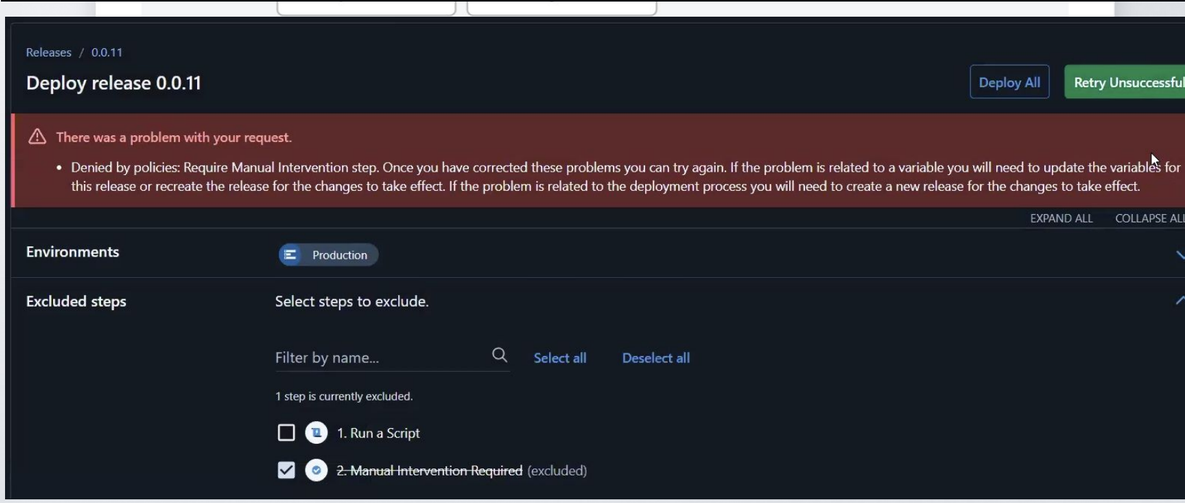
```
name = "Require Manual Intervention step"
description = "Require Manual Intervention step"
content = <<-EOT
package mypolicy

default result := {"allowed": false}

result := {"allowed": true} if {
        is_production
        some step in input.Steps
        not step.Id in input.SkippedSteps
        step.ActionType == "Octopus.Manual"
}

result := {"allowed": true} if not is_production

is_production if {
        some env in data.Environments
        env.Slug == "production"
        input.EnvironmentId == env.Id
}
EOT
```



Platform Hub— four key components

Process Templates

Policy Engine

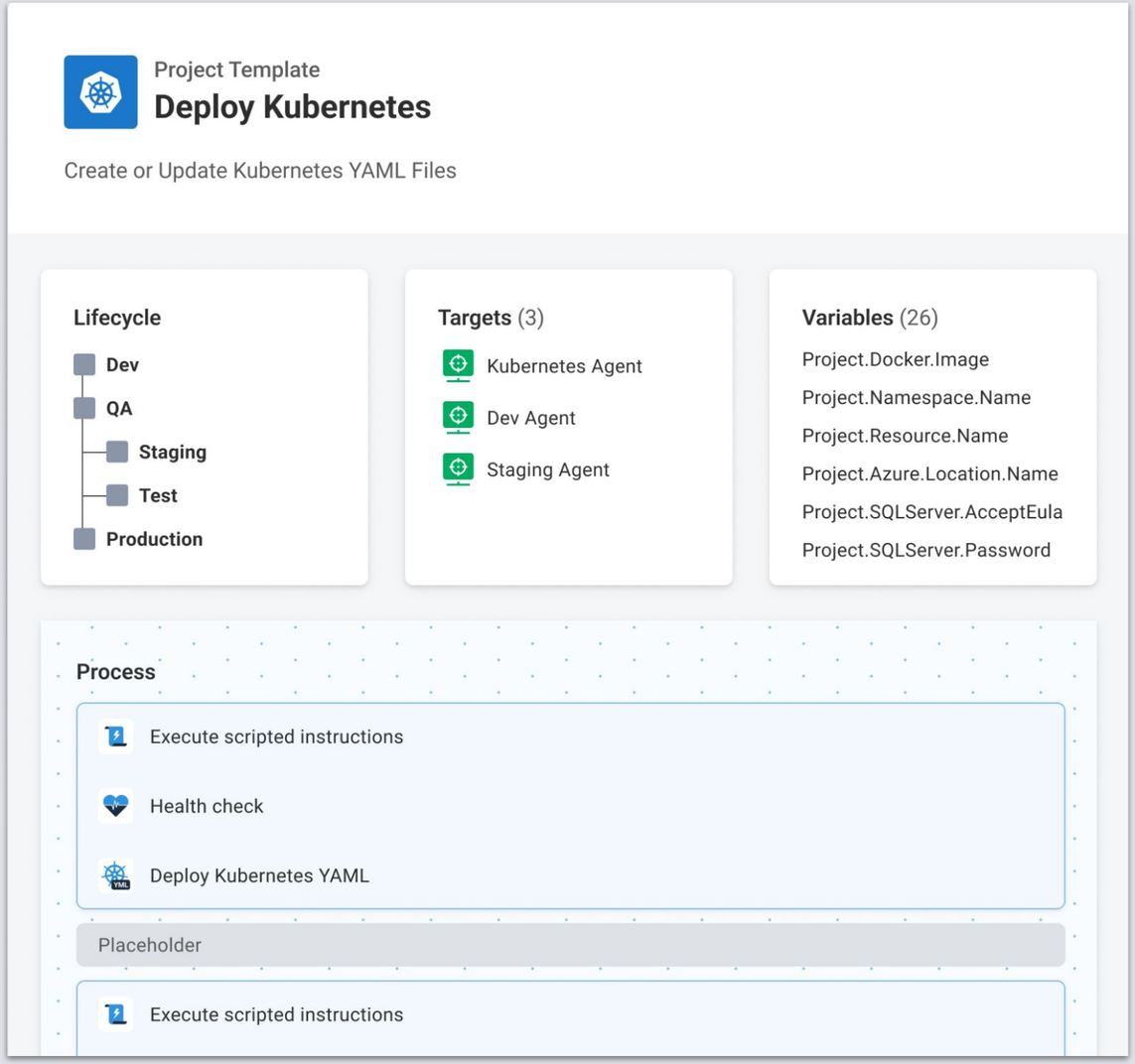
📖 **Project Templates**

Compliance Dashboard

Template of entire project to go from zero to deploying to production with standards and compliance built in. Template is chosen at project creation time.

Includes all the resources needed for a project—e.g.: Deployment Process, Runbooks, Lifecycles, Environments, and Variables.

Same versioning and syncing capabilities as process templates.



Platform Hub— four key components

Process Templates

Surface process and project template usage in a centralized location.

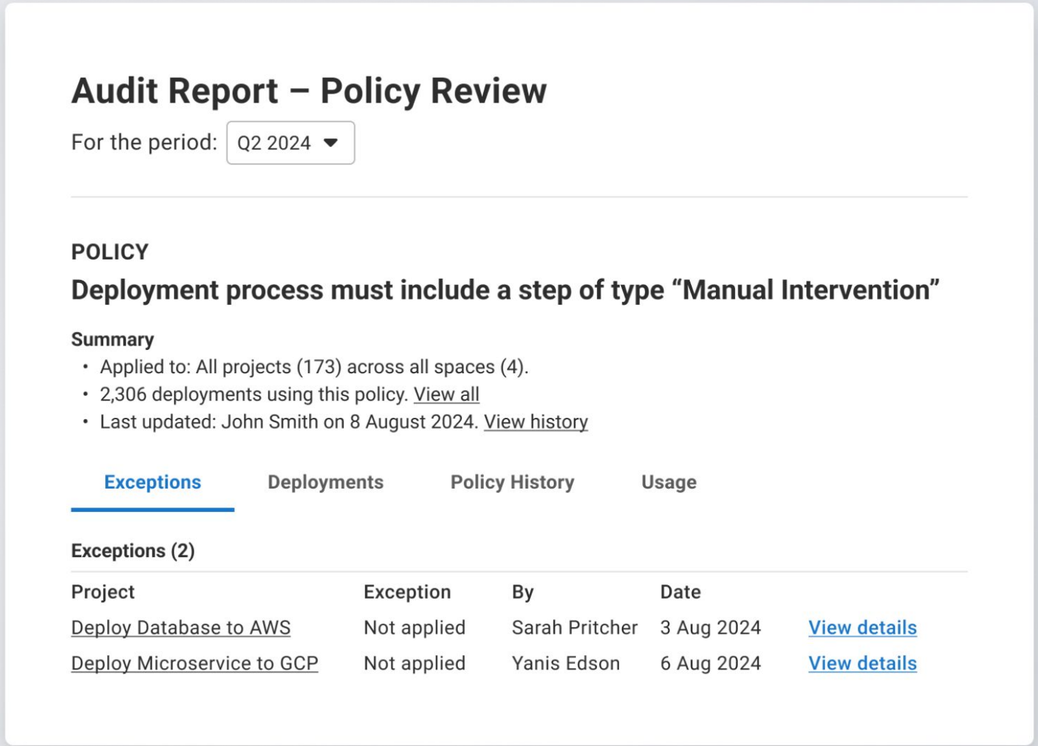
Policy Engine

Add exceptions and approve log exceptions with a single click.

Project Templates

Audit reports to auditing of template and policy usage.

Compliance Dashboard



Outline

What we hear as the current challenges with audit and compliance

How GitHub and Octopus impacts auditing and compliance

Demo of CI/CD pipeline using GitHub + Octopus

Tooling Responsibilities in the pipeline

GitHub

✅	Branch protection policies
✅	Pull request workflow
✅	Linting, static code analysis, and vulnerability scanning
✅	Automated testing (unit tests, integration tests, etc.)
✅	Create and publish build artifacts (packages, containers, etc.)
✅	Generating SBOMs and Attestations
✅	Calculating version numbers
✅	Creating releases in Octopus Deploy

Octopus Deploy

✅	Environmental progression and release orchestration
✅	Centralized dashboard for visibility of deployment and version status
✅	Getting deployment approvals via ITSM tooling
✅	RBAC and separation of duties for production deployments
✅	Ingesting SBOMs and verifying Attestations
✅	Environmental modeling of infrastructure
✅	Authentication and authorization to deployment targets (K8s Clusters, Azure Web Apps, etc.) and cloud providers
✅	Creating and destroying ephemeral environments

Demo

How we help audit and compliance

THE KEY SLIDE TO REMEMBER FROM THIS PRESENTATION

✅	Improve auditability and reduce risk though a best in breed CI/CD pipeline
✅	Provide enterprise-wide standardization of processes and increased visibility
✅	Increase traceability and reduce manual reporting
✅	Remove manual steps to increase efficiency.
✅	Improves DevEx by removing friction from the process and building trust

Octopus Deploy

GitHub

Thank you!

Bob Walker | Field CTO | Octopus Deploy

Philip Holleran | Field CTO | GitHub

The ideal CI/CD pipeline

✅	Supply chain security (Vulnerability Scanning, SBOMs, etc.)
✅	All components (including database schema) is included
✅	Build once, deploy anywhere
✅	RBAC controls and separation of duties
✅	Testing in prod-like (ephemeral) environments
✅	Trunk-based development with feature toggles

Common challenges —at enterprise scale

Focus	Challenge	Why this is important
Standardization and Scalability	**As a Platform Engineer**, how can I standardize the tooling (IE Kustomize vs. Helm) and establish deployment pipelines that can be quickly created for any team across the enterprise **As a DevOps Engineer**, how can I know which tooling is preferred and best practices to leverage them?	When each application team chooses its own tools, the result is fragmentation. Best practices slip away, leaving confusion. Without standards, **costs rise and risks multiply**.
Audit and Compliance	**As a Platform Engineer**, how can I ensure compliance across all your deployments? **As an Auditor**, how can I discover which applications are in compliance vs non-compliance?	Every industry has its rules. Deployments need clear steps to ensure they meet company standards and regulations. If you don't build this into the process, **you risk missing and breaking the rules**.
DevEx and Innovation	**As a Platform Engineer**, what is the process for updating all your applications' pipelines when a new process or tool is introduced? **As a DevOps engineer**, what is the process for introducing new tooling or deployment strategies and contributing to improving the processes? **As the VP of DevEx**, how can encourage innovation and make it easy for devs to focus on developing and get visibility into their deployments —while integrating automated auditability for SOX	No process stays the same. Companies **change tools and deployment strategies over time**. The users must be part of this, able to suggest and shape the changes. When the standard shifts, it should be simple to share with the application teams.
Ease of Use and Efficiency	**As a DevOps engineer**, what steps are involved when creating a new pipeline or adding an existing application? **As a Application Engineer**, how can I quickly get my application changes to production? **As the VP of DevEx**, how can I encourage quick adoption?increase speed from development to production deployment—while also building in SOX compliance and security	The higher the friction, the slower the adoption. New and existing applications must find it easy to embrace the new processes. Adding unnecessary steps makes it harder, causing delays. In that time, teams miss the benefits and **risk falling out of compliance**.
Risk Reduction	Automate application deployments to meet regulatory requirements and mitigate compliance risks consistently	

LET'S ZOOM ON ON THIS

Common Compliance Tasks

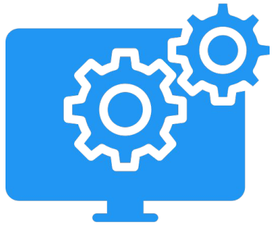
Focus	How
1 **Audit Logging**	Every action is logged providing a chain of custody from initial creation to production deployment.
2 **Traceability**	View the progression of each release across various environments.
3 **Separation of duties**	Only authorized users can access specific projects, environments, and actions.
4 **Approvals**	Enforce and automate compliance checkpoints right in the workflow.
5 **Multi-Technology Support**	Support complex deployments across Kubernetes, VMs, Cloud PaaS, and serverless platforms.

Continuous Delivery is more than delivering code

DEVELOPER EXPERIENCE

Make it easy for developers to deploy their code as it is to write it—by removing **auditing burden** and reducing cognitive load through integrating compliance policies into the deployment pipeline.

DRIVE DEVELOPER TRUST INTO A CONSISTENT AND RELIABLE DEPLOYMENT PROCESS



PLATFORM ENGINEERING

Make it easy for platform engineers to provide a centrally managed deployment platform—with templates and **enforceable guardrails** to remove friction for developers in their deployment process.

REDUCE FRICTION AND RISKY BEHAVIOR TO SOLVE URGENT ISSUES

CD at scale —with speed and safety

Standardization

= Templates
= Decrease configuration chaos + increase innovation
= **Blueprints** to repeat and scale best practices
= **Speed**

BUILD STANDARDIZATION, AUTOMATION, AND AUDITING, WITH COMPLIANCE AND SECURITY

Compliance

= Policies
= Increase governance + reduce risk
= **Guardrails** to enforce adaptable policies
= **Safety**

CD AT SCALE REDUCES RISK, TIME, AND COSTS REQUIRED TO MANAGE PROCESSES WHILE STILL ENCOURAGING INNOVATION