



# 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](mailto:bob.walker@octopus.com)



**Philip Holleran**

Field CTO

GitHubber since 2015

[pholleran@github.com](mailto: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

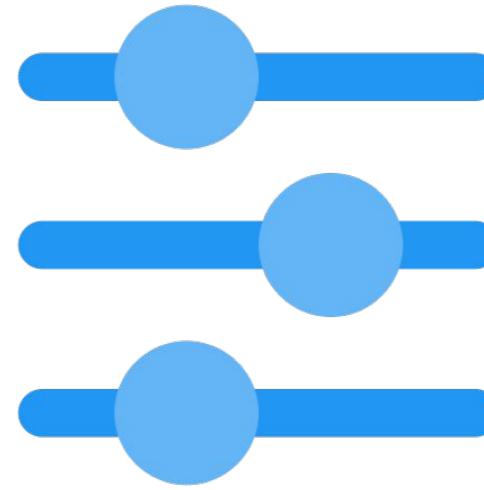
What we hear as the current **challenges** with audit and compliance

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 through 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?

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?

I NEED CONTROLS ON  
WHO CAN DO WHAT

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

WAS THE INTERNAL AUDIT  
FINDING MY FAULT?

THIS PROBLEM IS  
HAPPENING MORE  
FREQUENTLY

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

I DON'T TRUST  
THE PROCESS



# 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

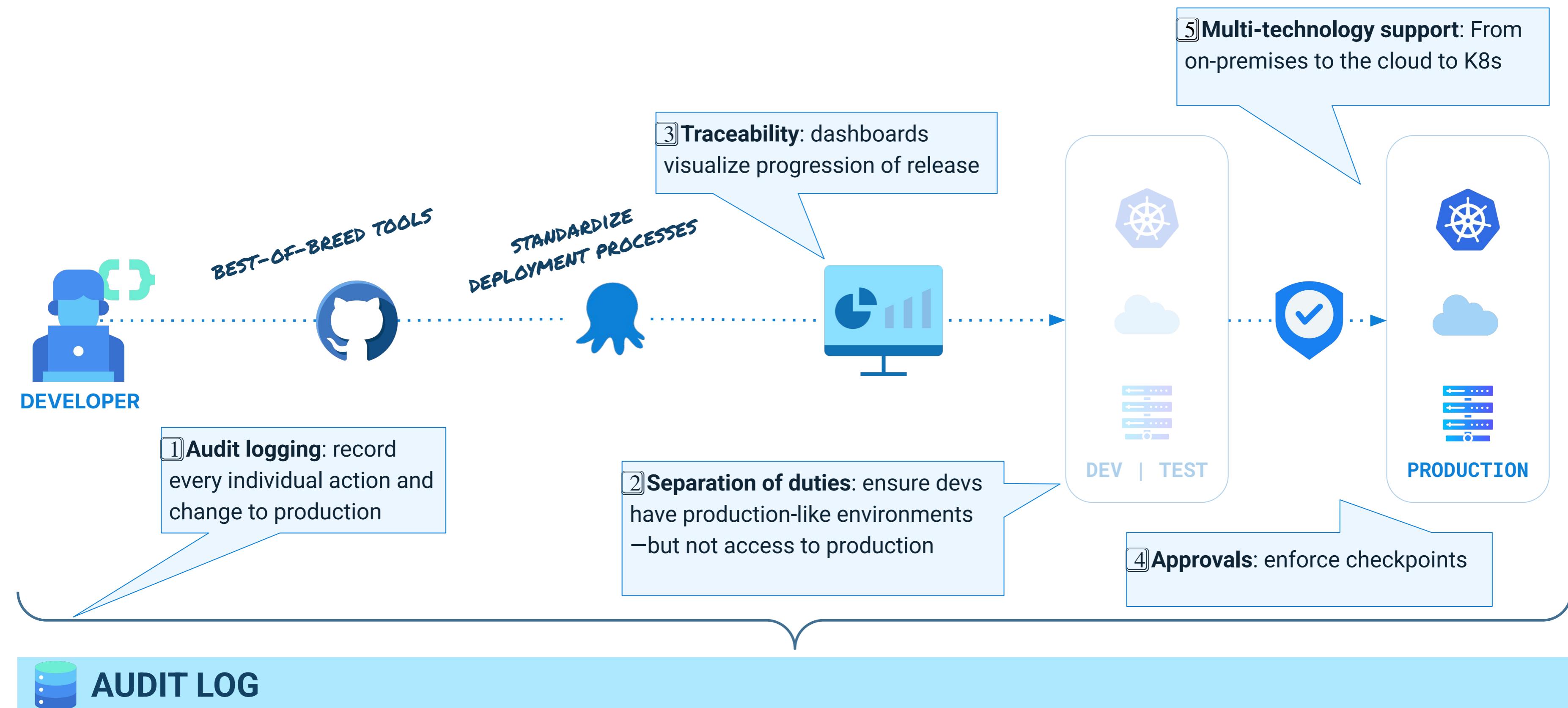
MORE CRITICAL BECAUSE OF  
EXECUTIVE ORDER 14028

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."

## Producers ✖

### Platform Engineers

- Create self-service golden paths for app teams
- Bring consistency to pipelines
- Enforce company policies



### DevOps Engineers or Team Leads

- Build and maintain deployment pipelines for their team
- Ensure deployment pipeline is flowing for their application aka troubleshooting



## Consumers 🤝

### Software Developer / Software Engineer

- Deploys the applications' changes, directly or indirectly (via build server)
- Able to recover from common failures
- Uses runbooks for self-service



### QA Engineers, Support Engineers, Product Managers

- Determines last version deployed for testing or prod
- Uses runbooks for day 2 operations / recovery



# 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.

## Compliance Dashboard

The screenshot shows the Octopus Platform Hub interface. On the left, a sidebar lists various settings: Channels, Releases, Feature Toggles, Triggers, Freezes, Settings, Operations, Runbooks, Runbook Triggers, Ephemeral Environments, Project Variables, Tenant Variables, Variable Sets, All Variables, Variable Preview, Tenants, Tasks, Insights, Project Settings, Version Control, and ITSM Providers. The main area is titled "Process" and shows a "main" workflow. A note at the top states: "refs/heads/main is a protected branch. Protected branches cannot be modified." The workflow consists of five steps:

1. Azure Key Vault - Retrieve Secrets: Run a script using an Azure subscription, with Azure modules loaded by default. Runs for any environment.
2. Verify Build Artifacts: Process Template. Contains: Attach SBOM to Release (Run a script on a worker from the pool selected via the #[Template.Verify.WorkerPool] variable. Runs for any environment), Verify Package SHA (Run a script on a worker from the pool selected via the #[Template.Verify.WorkerPool] variable. Runs for any environment).
3. Configure Infrastructure: Run a script on a worker from the pool selected via the Standards.Worker.Pool variable. Runs for any environment.
4. Deploy Databases: Process Template. Contains: Build Delta Report and check for auto approval (Run a script on a worker from the pool selected via the #[Template.WorkerPool] variable. Script references package from octopus-server-built-in. Runs for any environment), Notify DBAs (Send an email to teams specified by Template.Approval.Teams. Runs for any environment), DBAs Approve Delta Script (Manual intervention. Runs for any environment).
5. Deploy Database Changes: Run a script on a worker from the pool selected via the #[Template.WorkerPool] variable. Script references package from octopus-server-built-in. Runs for any environment.

At the bottom of the main area, it says: "GitHub owner: bobjwalker GitHub repo: Trident GitHub workflow: build.yml". A yellow annotation in the top right corner says "INCLUDED IN TODAY'S DEMO!".

# Platform Hub— four key components



## Process Templates

### 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.

### Publish version

Deploy Process - Deploy Database  
1.0.0 → 2.0.0

main

#### Release type

- Major changes (breaking)
- Minor changes (non-breaking)
- Patch (bug fixes)

#### FLAG AS PRE-RELEASE (OPTIONAL)

Flag as pre-release

Recommended if you are still testing the template. Pre-release templates are flagged in the Process Editor step selector.

PRODUCER

INCLUDED IN TODAY'S  
DEMO!

The screenshot shows the 'Deploy Databases' process template in the Platform Hub. The template consists of the following steps:

1. Azure Key Vault - Retrieve Secrets
2. Verify Build Artifacts
  - Attach SBOM to Release
  - Run Attestation Verification on Build ...
3. Verify Infrastructure
4. Deploy Databases
  - Build Delta Report and check for auto...
  - Notify DBAs
  - DBAs Approve Delta Script
  - Deploy Database Changes
5. Deploy Trident Website
6. Verify Deployment
7. Notify Team of Deployment Status

The template is currently at version v2.0.0. The 'Settings' tab is selected, showing the following details:

- Name:** Deploy Databases deploy-databases
- Template:** Deploy Process - Deploy Database
- Description:** Process template that will deploy a database using DBUp
- Versioning:** Choose how you want to receive updates for this template.
- Select update behaviour:**
  - Accept minor changes**: Automatically update the template whenever a patch or minor version is published.
  - Accept patches**: Automatically update the template whenever a patch is published.

Yellow arrows from the 'PRODUCER' and 'CONSUMER' labels point to the 'Major changes (breaking)' radio button and the 'Accept minor changes' radio button respectively.

CONSUMER

# 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.

```
1  name = "Require Manual Intervention step"
2  description = "Require Manual Intervention step"
3  content = <<-EOT
4  package mypolicy
5
6  default result := {"allowed": false}
7
8  result := {"allowed": true} if {
9      is_production
10     some step in input.Steps
11     not step.Id in input.SkippedSteps
12     step.ActionType == "Octopus.Manual"
13   }
14
15  result := {"allowed": true} if not is_production
16
17  is_production if {
18      some env in data.Environments
19      env.Slug == "production"
20      input.EnvironmentId == env.Id
21   }
22 EOT
```

Releases / 0.0.11

Deploy release 0.0.11

⚠ There was a problem with your request.

- Denied by policies: Require Manual Intervention step. Once you have corrected these problems you can try again. If the problem is related to a variable you will need to update the variables for this release or recreate the release for the changes to take effect. If the problem is related to the deployment process you will need to create a new release for the changes to take effect.

EXPAND ALL COLLAPSE ALL

Environments Production

Excluded steps Select steps to exclude.

Filter by name...  Select all Deselect all

1 step is currently excluded.

1. Run a Script

2. Manual Intervention Required (excluded)

Deploy All Retry Unsuccessful

# Platform Hub— four key components

Process  
Templates

Policy  
Engine

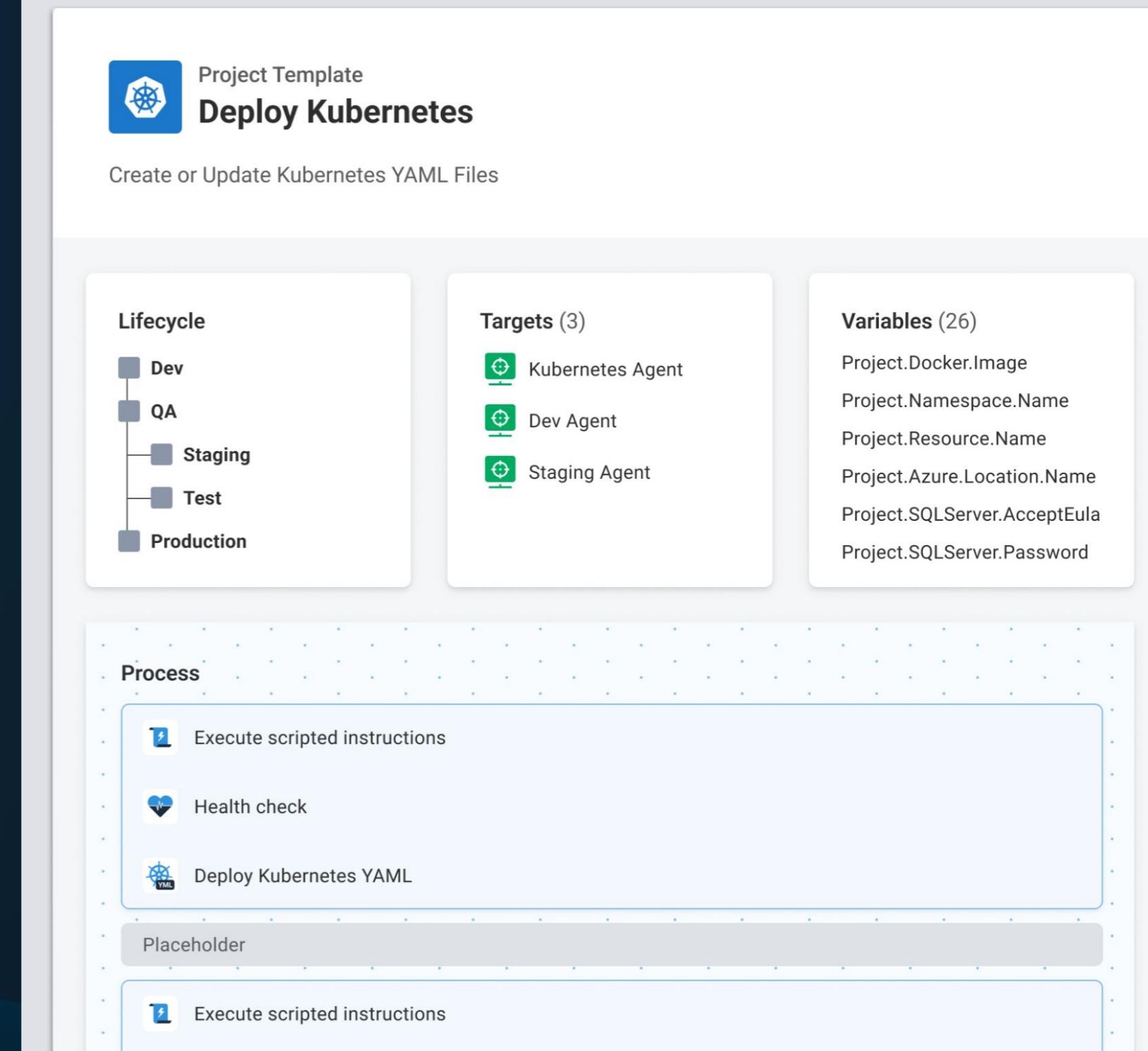
 Project  
Templates

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

## Audit Report – Policy Review

For the period: Q2 2024 ▾

### POLICY

**Deployment process must include a step of type “Manual Intervention”**

#### Summary

- Applied to: All projects (173) across all spaces (4).
- 2,306 deployments using this policy. [View all](#)
- Last updated: John Smith on 8 August 2024. [View history](#)

[Exceptions](#) [Deployments](#) [Policy History](#) [Usage](#)

#### Exceptions (2)

Project	Exception	By	Date	
<a href="#">Deploy Database to AWS</a>	Not applied	Sarah Pritch	3 Aug 2024	<a href="#">View details</a>
<a href="#">Deploy Microservice to GCP</a>	Not applied	Yanis Edson	6 Aug 2024	<a href="#">View details</a>

# 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



- 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



- 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	<p><b>As a Platform Engineer</b>, 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</p> <p><b>As a DevOps Engineer</b>, how can I know which tooling is preferred and best practices to leverage them?</p>	When each application team chooses its own tools, the result is fragmentation. Best practices slip away, leaving confusion. Without standards, <b>costs rise and risks multiply</b> .
Audit and Compliance	<p><b>As a Platform Engineer</b>, how can I ensure compliance across all your deployments?</p> <p><b>As an Auditor</b>, how can I discover which applications are in compliance vs non-compliance?</p>	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, <b>you risk missing and breaking the rules</b> .
DevEx and Innovation	<p><b>As a Platform Engineer</b>, what is the process for updating all your applications' pipelines when a new process or tool is introduced?</p> <p><b>As a DevOps engineer</b>, what is the process for introducing new tooling or deployment strategies and contributing to improving the processes?</p> <p><b>As the VP of DevEx</b>, 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</p>	No process stays the same. Companies <b>change tools and deployment strategies over time</b> . 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	<p><b>As a DevOps engineer</b>, what steps are involved when creating a new pipeline or adding an existing application?</p> <p><b>As a Application Engineer</b>, how can I quickly get my application changes to production?</p> <p><b>As the VP of DevEx</b>, how can I encourage quick adoption? increase speed from development to production deployment—while also building in SOX compliance and security</p>	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 <b>risk falling out of compliance</b> .
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 <b>Audit Logging</b>	Every action is logged providing a chain of custody from initial creation to production deployment.
② <b>Traceability</b>	View the progression of each release across various environments.
③ <b>Separation of duties</b>	Only authorized users can access specific projects, environments, and actions.
④ <b>Approvals</b>	Enforce and automate compliance checkpoints right in the workflow.
⑤ <b>Multi-Technology Support</b>	Support complex deployments across Kubernetes, VMs, Cloud PaaS, and serverless platforms.



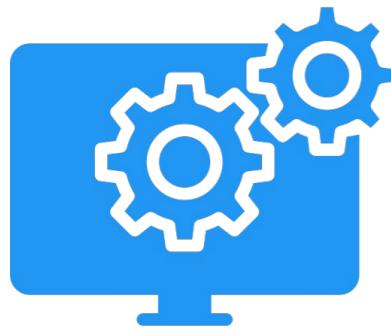
# Continuous Delivery is more than delivering code



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

