CloudBees DAYS

Agenda

- Introduction
    - Kubernetes based apps
    - Jenkins X and Jenkins
    - GitOps
    - Library vs. Kubernetes Workloads
    - Build Packs

- Jenkins X and CloudBees Core for Kubernetes CD Workshop

Why should you want Native Kubernetes CD?

Kubernetes is an excellent platform for CD because it has core features that lend themselves to high-performing continuous delivery. These features include:

- Scalability: K8s allows you to easily scale your CD workload up and down.
- Resilience: K8s is fault tolerant. You can't execute CD if your CD platform is down.
- Built-in objects: Resource, Config and Credential management is built-in. These objects form the core of any CD platform.
- Extensibility: K8s provides a number of extension points to include Custom Resource Definitions - ensuring that K8s is capable of providing a robust solution for any number of specialized use cases - like CD.

How? - Best Practices for Native K8s CD

- Design cloud-native apps
- Adopt containers and schedulers
- Adopt Tekton as serverless solution for CI/CD pipelines in Kubernetes
- Adopt Prow for GitHub[1] automation with GitOps and ChatOps
- Adopt Helm as the standard Kubernetes packaging solution
- Join the tools and the processes into a single easy-to-use platform
- Create a CLI-first experience
- Define a prescriptive and easy to use process

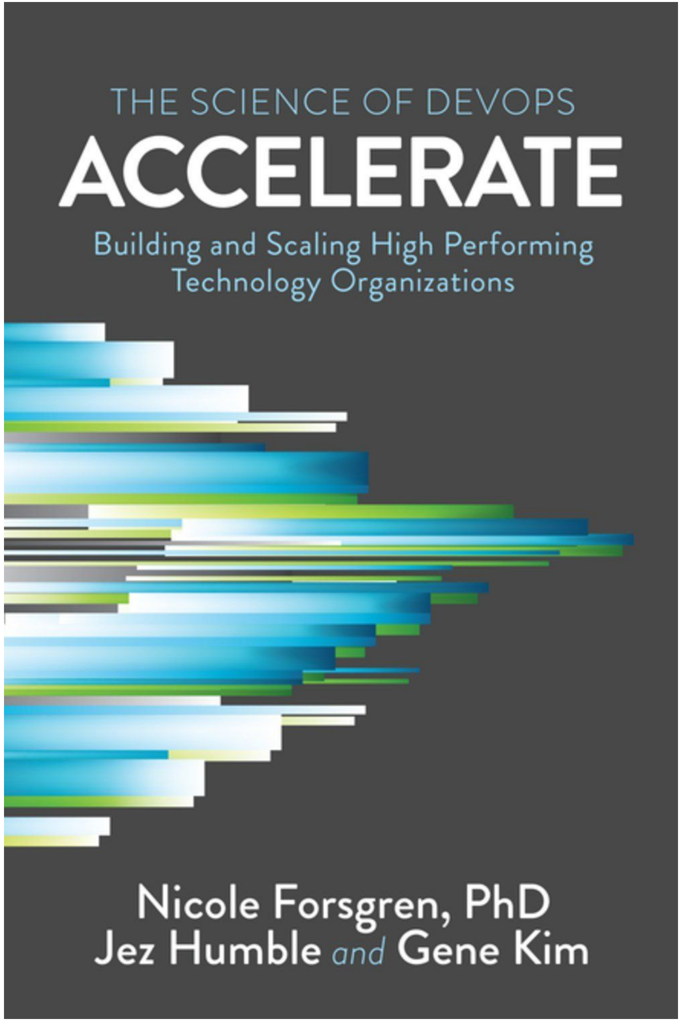
[1] Prow currently supports GitHub and GitHub Enterprise, there is an issue tracking support for other flavors of Git.

How do you do it?

- Are you an expert in kubernetes?
- Do you know how to create helm charts?
- Do you understand the intricacies of Prow and Tekton?
- How about skaffold, chartmuseum, ksync, ...?
- Do you know how to configure GitOps and GitChat?
- Do you expect everyone in your organization to know all that?
- Do all your projects employ continuous delivery?
- If the answer to any of those is no, how do you plan to be competitive?

Jenkins X

https://jenkins-x.io/

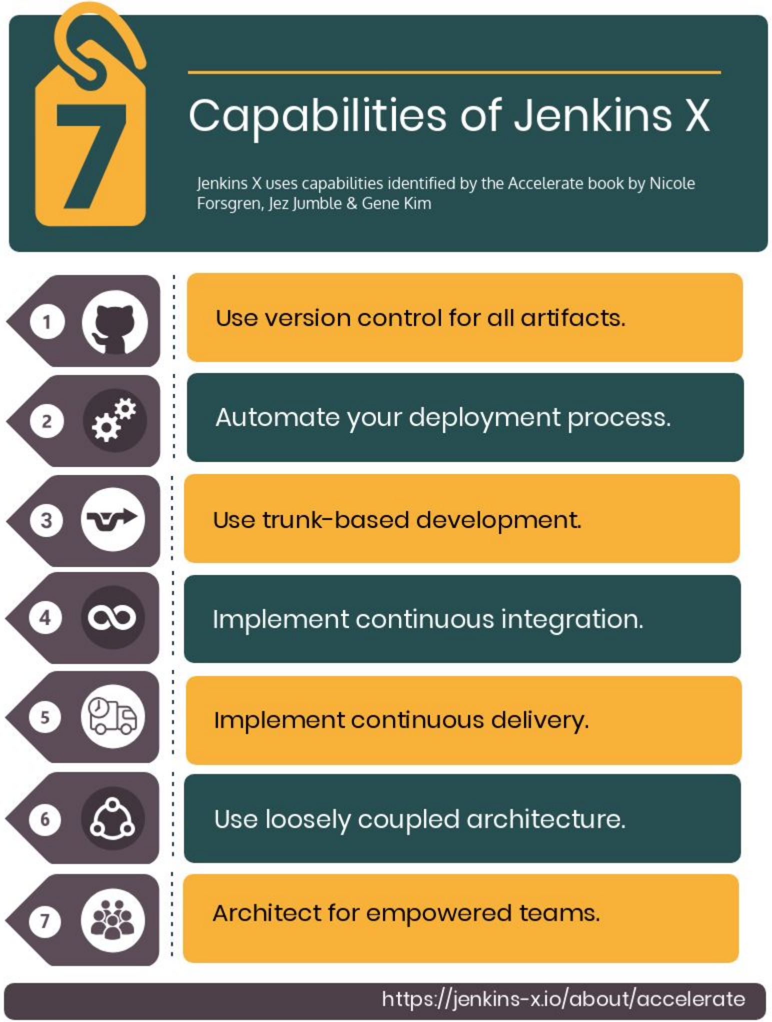


THE SCIENCE OF DEVOPS

ACCELERATE

Building and Scaling High Performing Technology Organizations

Nicole Forsgren, PhD
Jez Humble *and* Gene Kim



7 Capabilities of Jenkins X

Jenkins X uses capabilities identified by the Accelerate book by Nicole Forsgren, Jez Jumble & Gene Kim

1. Use version control for all artifacts.
2. Automate your deployment process.
3. Use trunk-based development.
4. Implement continuous integration.
5. Implement continuous delivery.
6. Use loosely coupled architecture.
7. Architect for empowered teams.

https://jenkins-x.io/about/accelerate

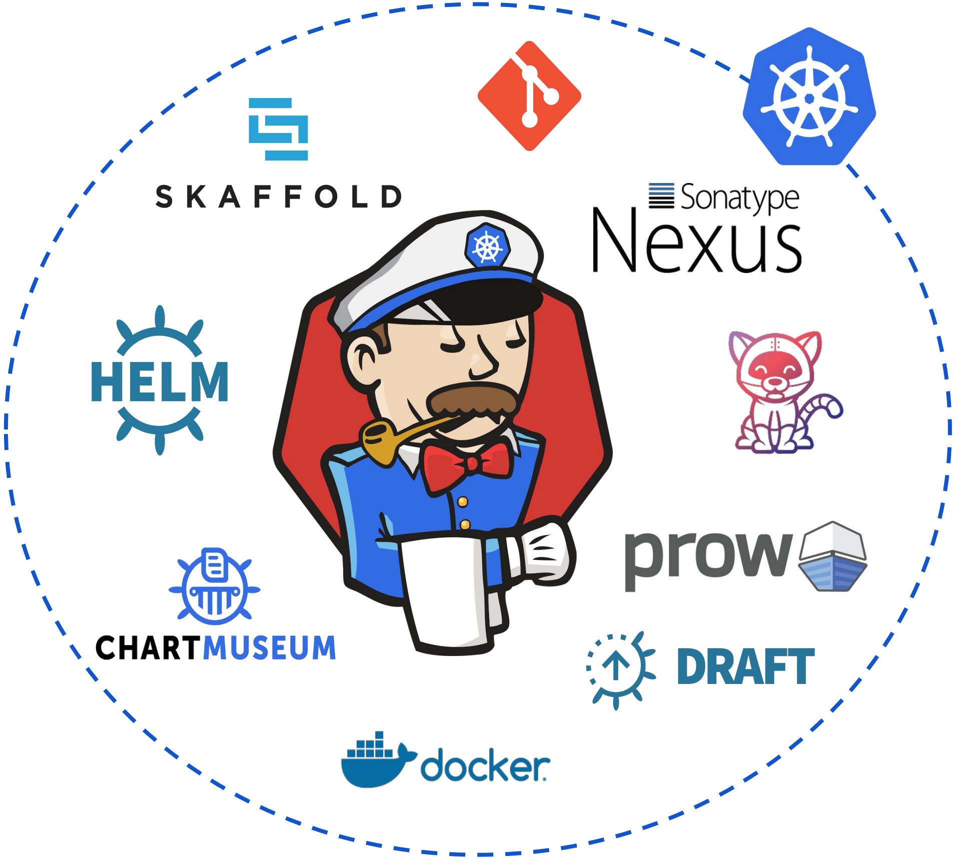
Developers can Concentrate on Developing

Do you know the difference?

- Must be installed on Kubernetes
  - **vs** Installed Anywhere
- Focused on K8s deployment
  - **vs** Deploy Anywhere
- CI/CD steps must run in containers
  - **vs** CI/CD steps can run anywhere
- Prescriptive best practices
  - **vs** Ultimate Flexibility

Do you know what we have in common?

- CloudBees is the main contributor
- Leading CI/CD solutions
- Fast growing open source projects
- Can both run on Kubernetes



CI/CD for Kubernetes powered by Jenkins X

CI/CD automation for Cloud

Jenkins Cloud Native implementation

Kubernetes only

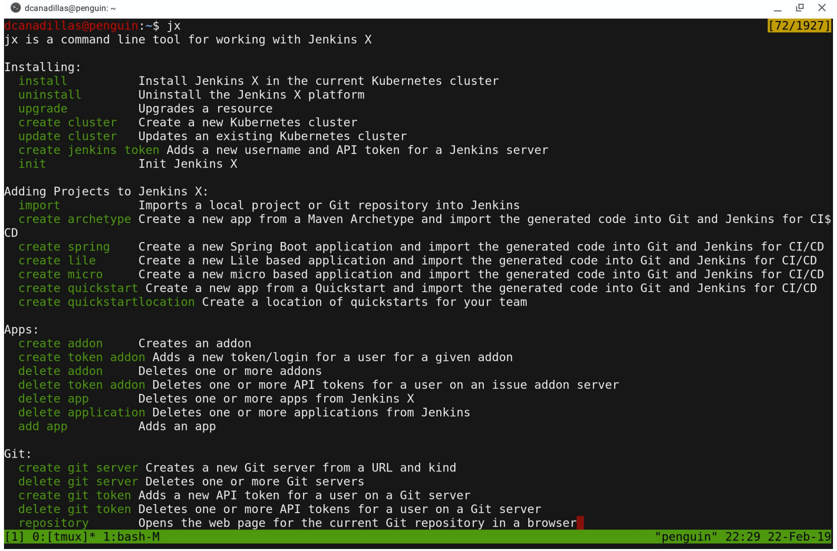
Extends K8s with CRDs

- Environments
- Pipeline Activities
- Releases
- Users
- Teams

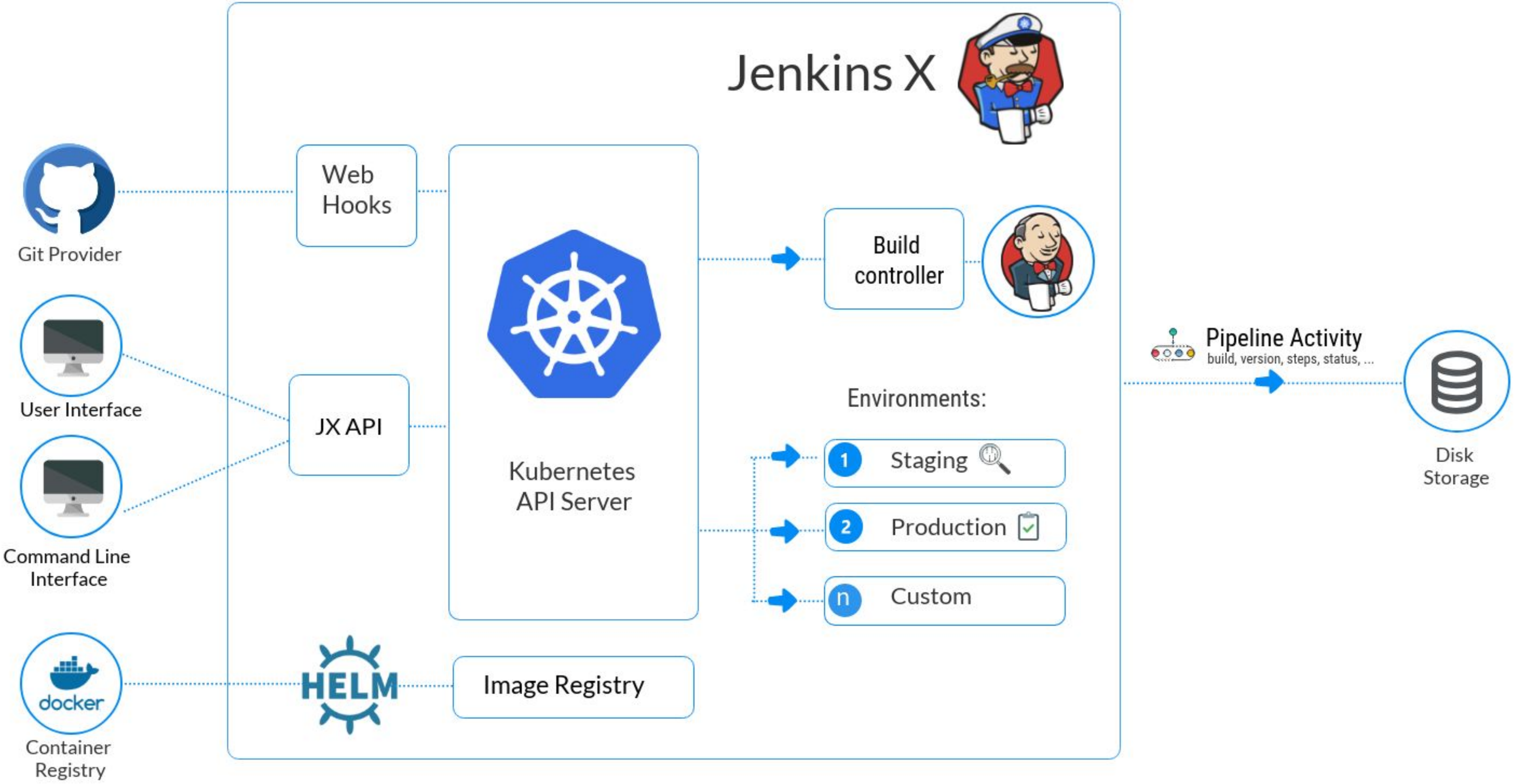
“One command to rule them all”

A CLI to start a real automation experience

```
$ jx create cluster gke --tekton --prow

$ jx get environments

$ jx import --url
https://github.com/dcanadillas/demo.git

$ jx get activity -f demo -w

$ jx get build logs <github-org>/demo/master

$ jx get applications

...
```



```
dcanadillas@penguin:~$ jx                                                                                   [72/1927]
jx is a command line tool for working with Jenkins X

Installing:
  install           Install Jenkins X in the current Kubernetes cluster
  uninstall         Uninstall the Jenkins X platform
  upgrade           Upgrades a resource
  create cluster    Create a new Kubernetes cluster
  update cluster    Updates an existing Kubernetes cluster
  create jenkins token Adds a new username and API token for a Jenkins server
  init              Init Jenkins X

Adding Projects to Jenkins X:
  import            Imports a local project or Git repository into Jenkins
  create archetype Create a new app from a Maven Archetype and import the generated code into Git and Jenkins for CI$
CD
  create spring     Create a new Spring Boot application and import the generated code into Git and Jenkins for CI/CD
  create lile       Create a new Lile based application and import the generated code into Git and Jenkins for CI/CD
  create micro      Create a new micro based application and import the generated code into Git and Jenkins for CI/CD
  create quickstart Create a new app from a Quickstart and import the generated code into Git and Jenkins for CI/CD
  create quickstartlocation Create a location of quickstarts for your team

Apps:
  create addon      Creates an addon
  create token addon Adds a new token/login for a user for a given addon
  delete addon      Deletes one or more addons
  delete token addon Deletes one or more API tokens for a user on an issue addon server
  delete app        Deletes one or more apps from Jenkins X
  delete application Deletes one or more applications from Jenkins
  add app           Adds an app

Git:
  create git server Creates a new Git server from a URL and kind
  delete git server Deletes one or more Git servers
  create git token Adds a new API token for a user on a Git server
  delete git token Deletes one or more API tokens for a user on a Git server
  repository        Opens the web page for the current Git repository in a browser
[1] 0:[tmux]* 1:bash-M                                                                   "penguin" 22:29 22-Feb-19
```



Jenkins X

Git Provider

Web Hooks

User Interface

JX API

Command Line Interface

Kubernetes API Server

Build controller

Environments:

1 Staging

2 Production

n Custom

Pipeline Activity

build, version, steps, status, ...

Disk Storage

Container Registry

HELM

Image Registry

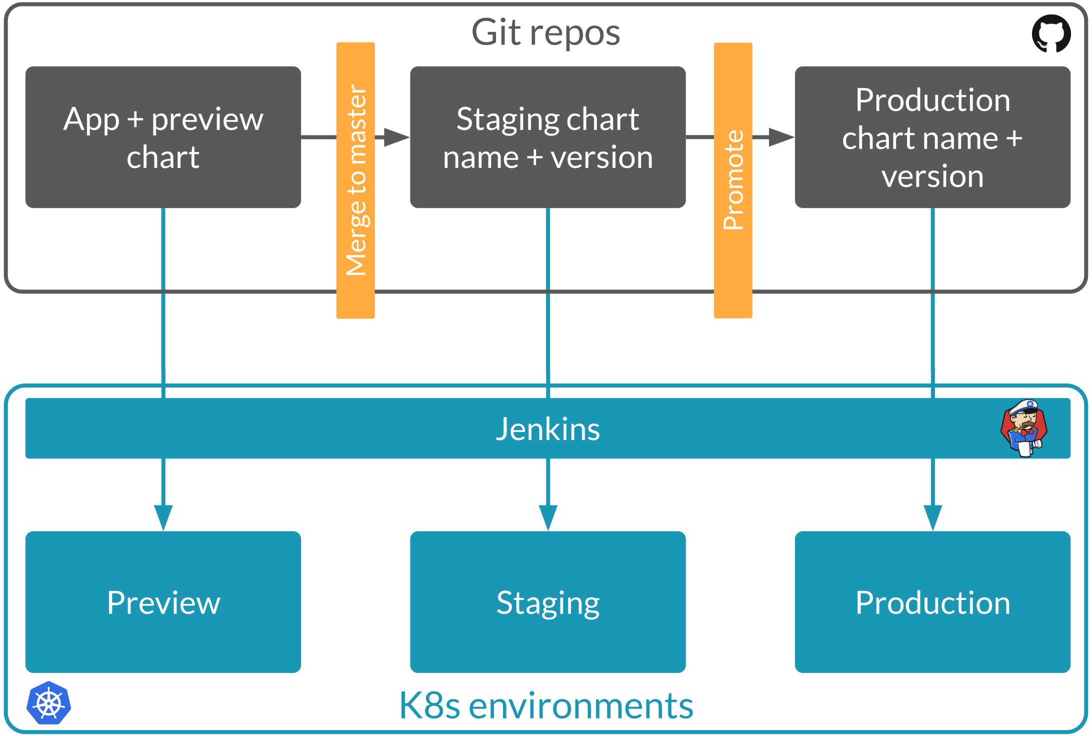
Promotion automation

Thinking about true DevOps…

“Continuous Delivery meets Cloud Native”

- ➔ IaC
- ➔ Automated promotions
- ➔ Git as source of truth

GitOps environments and promotion



Kubernetes Workloads
vs
Library Workloads

Library (classic) Workloads

- Jenkins X is not limited to applications targeting K8s for deployment
- Bring your own deployments and environments
- Library/Classic build pack supports CI+Releases but does not include CD.
  - e.g. do CI and release of your Java libraries or Node modules but don't deploy to Kubernetes
  - Extend the *Library* build pack with custom CD steps

Kubernetes Workloads

- This is what most people think of when they think of Jenkins X
- Includes deployment environments managed with GitOps with Staging and Production provided OOTB - modify or add additional environments
- Kubernetes Workload build packs support automated CI+CD with GitOps promotion and Preview Environments for kubernetes workloads

Build Packs

Build Packs

- Based on Azure Draft Packs
- Jenkins X provides two sets of build packs:
  - jenkins-x-kubernetes
  - jenkins-x-classic
- The K8s workload build packs extend the classic build packs
- Same build packs used for static and serverless
- Build packs are extensible

Custom Build Packs

- Fork https://github.com/jenkins-x-buildpacks/jenkins-x-kubernetes.git
- Add, modify, extend existing build packs in the fork
- Edit the build pack configuration for your team:

```
jx edit buildpack \
    -n kubernetes-workloads \
    -u https://github.com/{FORKED-ORG}/jenkins-x-kubernetes \
    -r master -b
```

Extending Build Packs

- import existing build packs via `imports.yaml`:

```
modules:
- name: classic
 gitUrl: https://github.com/jenkins-x-buildpacks/jenkins-x-classic.git
 gitRef: master
```

- extend:

```
extends:
 file: ../pipeline.yaml

extends:
 import: classic
 file: javascript/pipeline.yaml
```

  - add steps before:

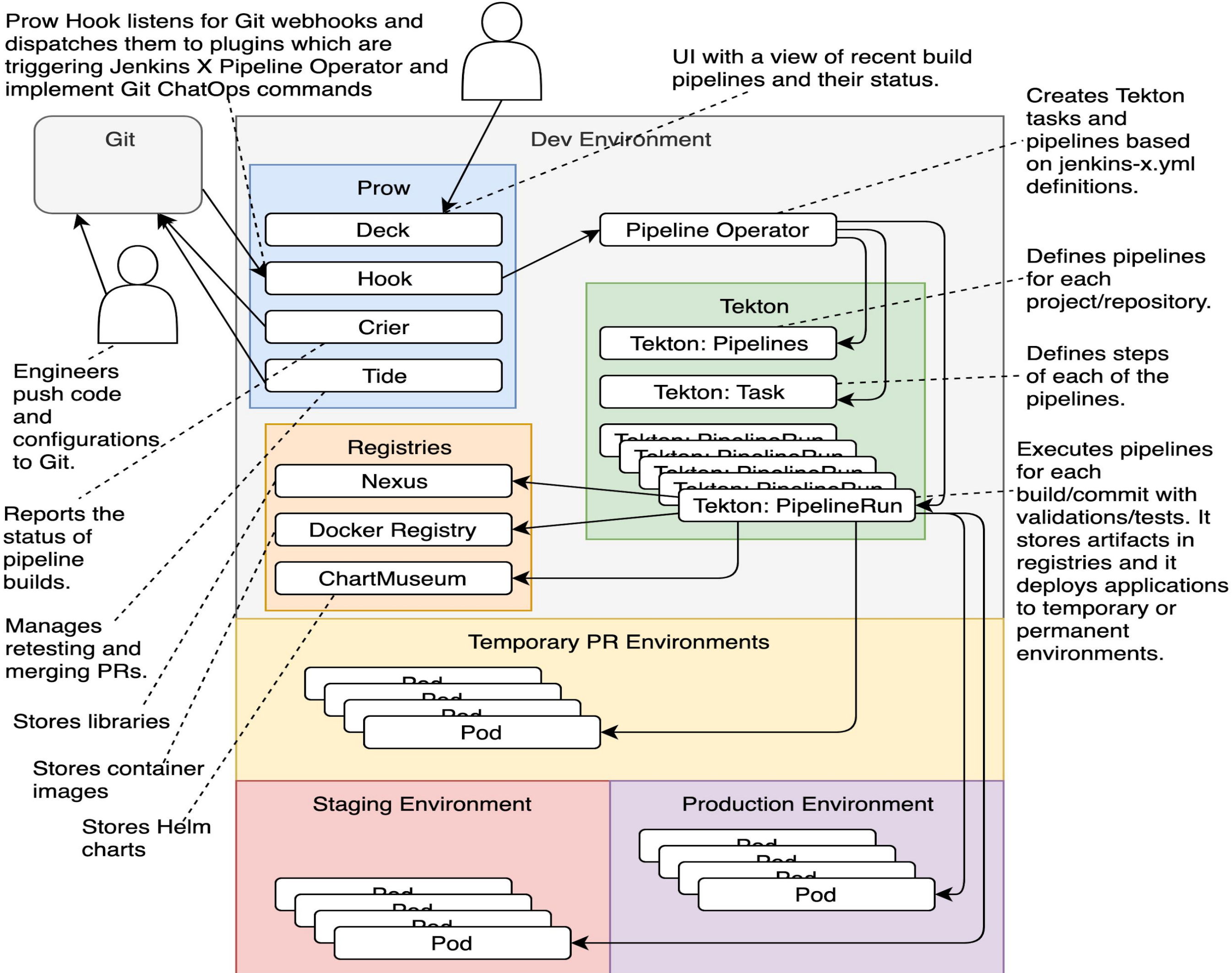
```
 release:
   setup:
     preSteps:
     - sh: echo BEFORE BASE SETUP
```

  - replace existing steps:

```
   build:
     replace: true
     steps:
     - sh: mvn clean deploy -Pmyprofile
       comment: this command is overridden from the base pipeline
```

Jenkins X/CloudBees Next Gen Pipeline

- A new Yaml based Declarative syntax that maps to the existing Declarative syntax
- Provide a more streamlined, easier to manage and comprehend Pipeline definition with purely K8s native Tekton Pipelines doing the heavy lifting behind the scenes
- Jenkinsfile Runner (already used by Serverless Jenkins X) will allow more complex Pipelines, either Declarative Pipelines utilizing shared libraries or fully Scripted Pipelines
- Stage reuse will be enabled through a new “stage library” concept. It will be similar to existing Scripted shared libraries, but will only allow definition of stages, and only stages which can be represented as Tekton Pipeline Tasks



Prow Hook listens for Git webhooks and dispatches them to plugins which are triggering Jenkins X Pipeline Operator and implement Git ChatOps commands

UI with a view of recent build pipelines and their status.

Creates Tekton tasks and pipelines based on jenkins-x.yml definitions.

Defines pipelines for each project/repository.

Defines steps of each of the pipelines.

Executes pipelines for each build/commit with validations/tests. It stores artifacts in registries and it deploys applications to temporary or permanent environments.

Engineers push code and configurations to Git.

Reports the status of pipeline builds.

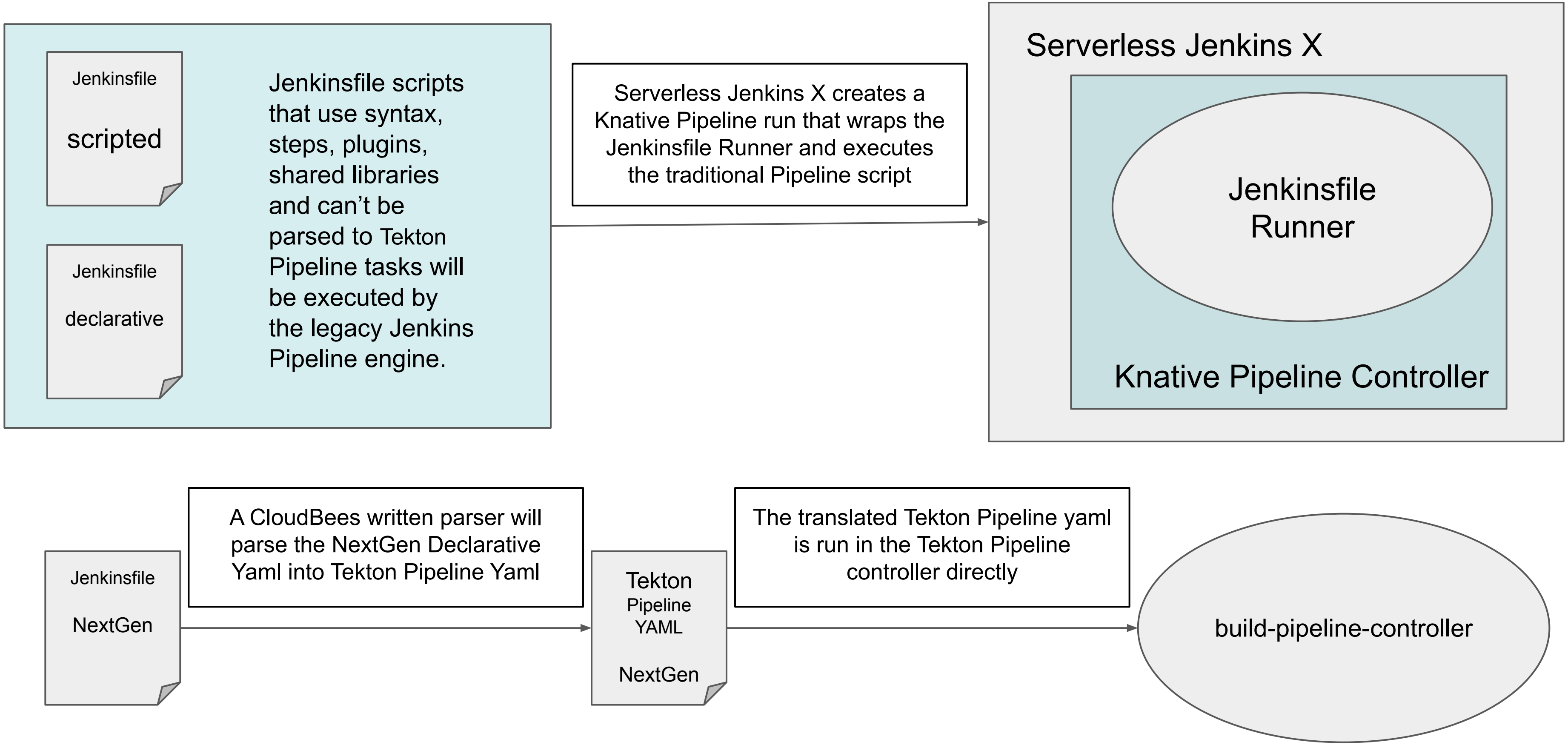
Manages retesting and merging PRs.

Stores libraries

Stores container images

Stores Helm charts

Next Gen Pipeline Flow



Jenkinsfile

scripted

Jenkinsfile

declarative

Jenkinsfile scripts that use syntax, steps, plugins, shared libraries and can’t be parsed to Tekton Pipeline tasks will be executed by the legacy Jenkins Pipeline engine.

Serverless Jenkins X creates a Knative Pipeline run that wraps the Jenkinsfile Runner and executes the traditional Pipeline script

Serverless Jenkins X

Jenkinsfile Runner

Knative Pipeline Controller

Jenkinsfile

NextGen

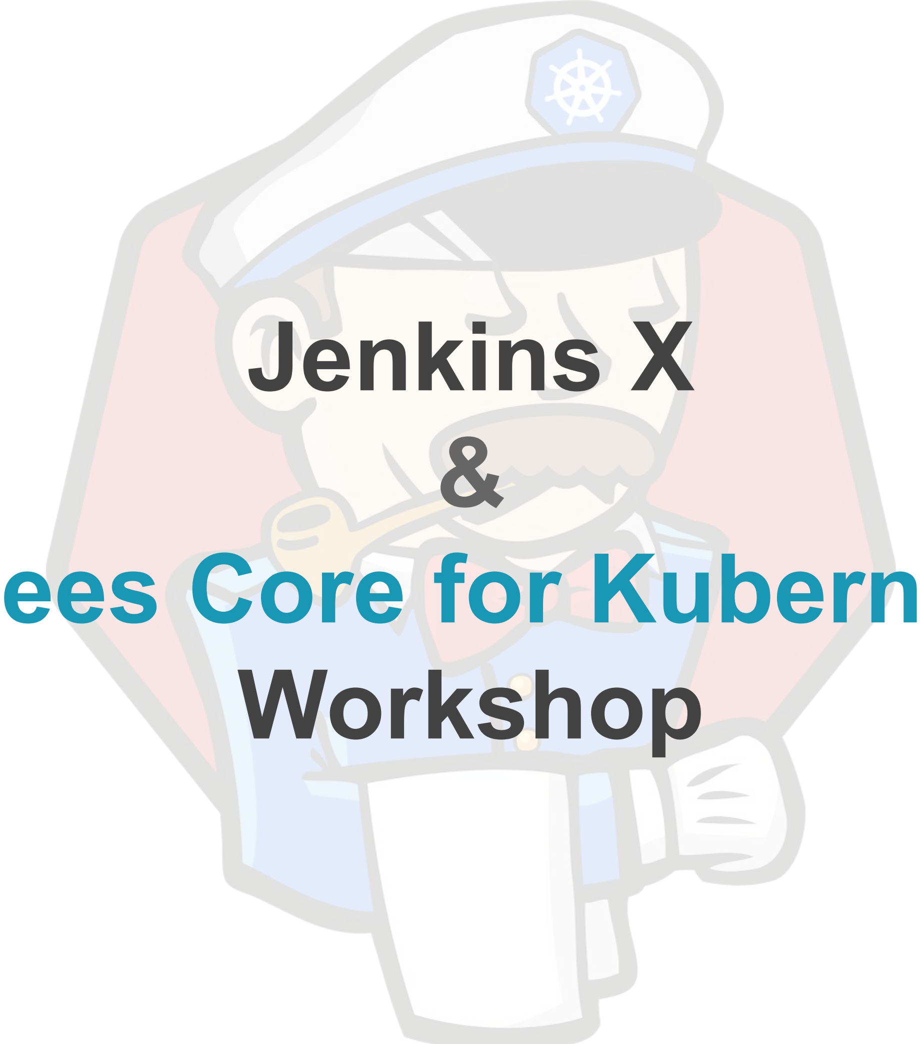
A CloudBees written parser will parse the NextGen Declarative Yaml into Tekton Pipeline Yaml

Tekton Pipeline YAML

NextGen

The translated Tekton Pipeline yaml is run in the Tekton Pipeline controller directly

build-pipeline-controller



Jenkins X
&
CloudBees Core for Kubernetes CD
Workshop

u: demo

p: cdxpreview