

cloudbees®





Cojan van Ballegooijen
EMEA Senior Solutions Architect

"I love technology and more...."



Agenda

Introduction

- Kubernetes based apps
- Jenkins X and Jenkins
- GitOps
- Jenkins X Serverless

Demo (live / video)



Why should you want Native Kubernetes CD?

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

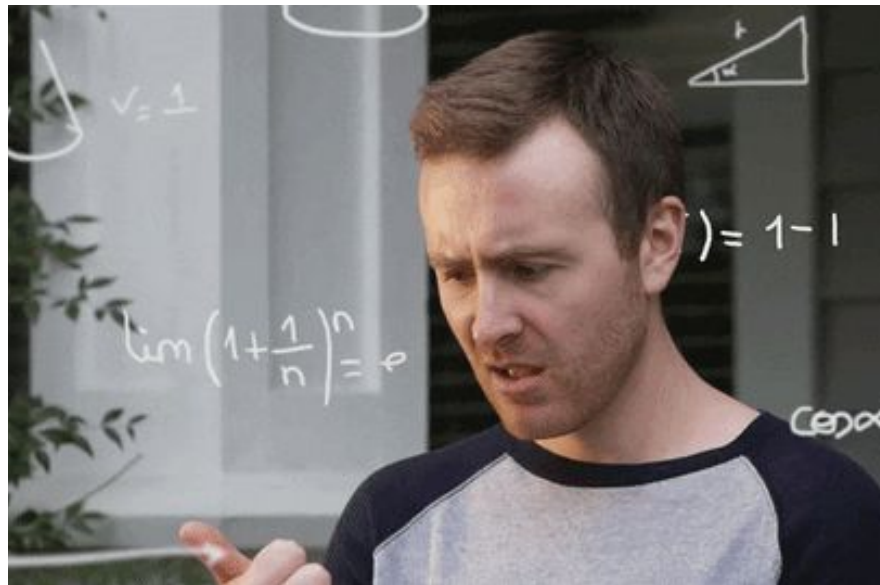
- Design cloud-native apps
- Adopt containers and schedulers
- Adopt Tekton as serverless solution for CI/CD pipelines in Kubernetes
- Adopt Prow for GitHub¹ automation with GitOps and ChatOps
- Adopt Kaniko to securely build and push container images
- Adopt Helm as the standard Kubernetes packaging 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 Kaniko, 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



Capabilities of Jenkins X

Jenkins X uses capabilities identified by the Accelerate book by Nicole Forsgren, Jez Humble & 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.

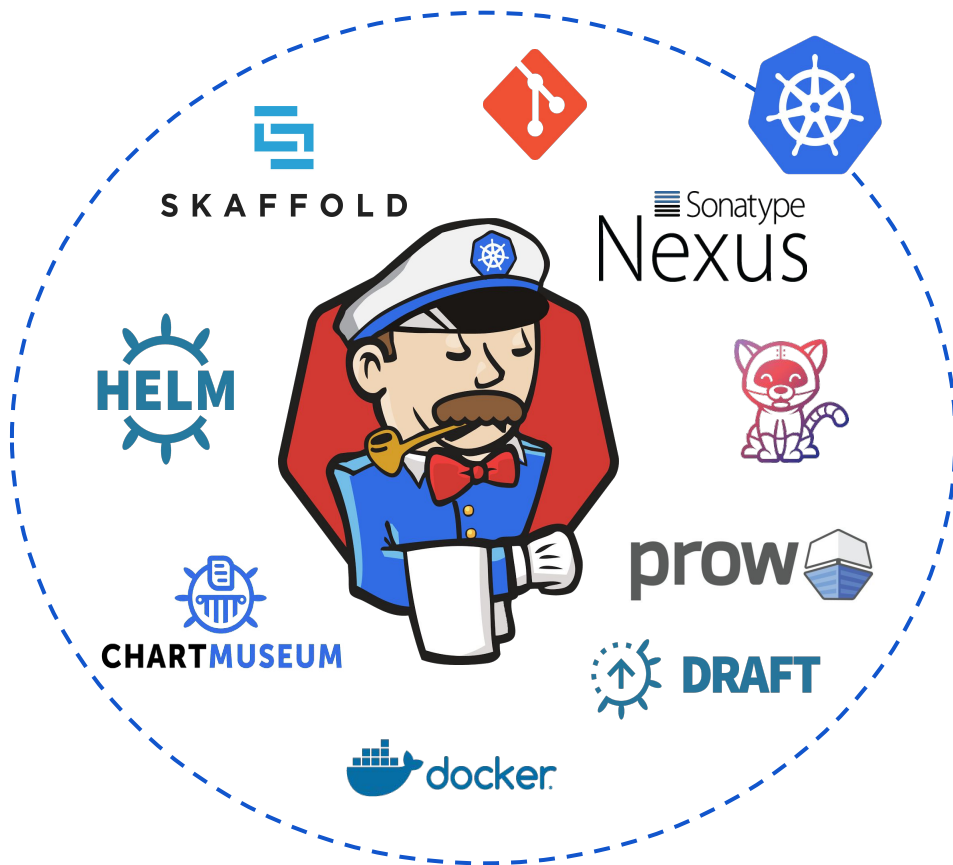
7



Architect for empowered teams.

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

red.



CI/CD for Kubernetes powered by Jenkins X

CI/CD automation for Cloud

Jenkins Cloud Native implementation

Kubernetes only

Extends K8s with CRDs

- Environment
- Pipeline Activities
- Releases
- Users
- Teams



You mean this is not even similar to Jenkins?



There are some differences



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



But they share some things



- CloudBees is the main contributor
- Leading CI/CD solutions
- Fast growing open source projects
- Can both run on Kubernetes
- Both are initial Continuous Delivery Foundation (CDF) projects



“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 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
```

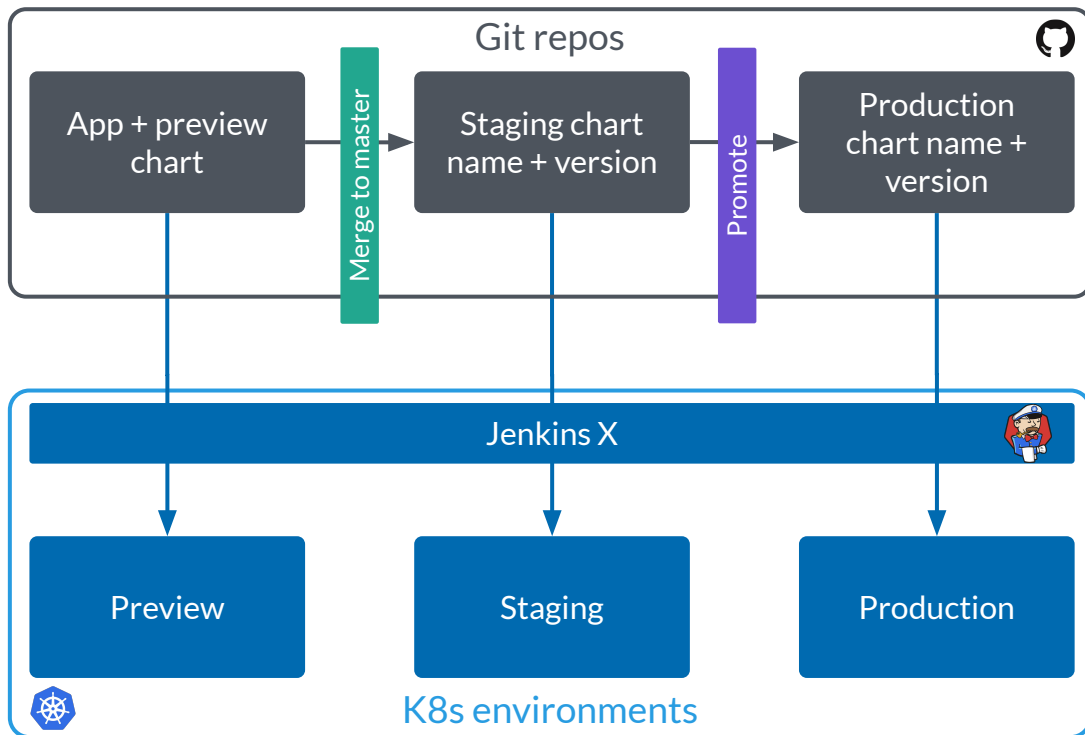


Promotion automation

Thinking about true DevOps...
“*Continuous Delivery meets Cloud Native*”

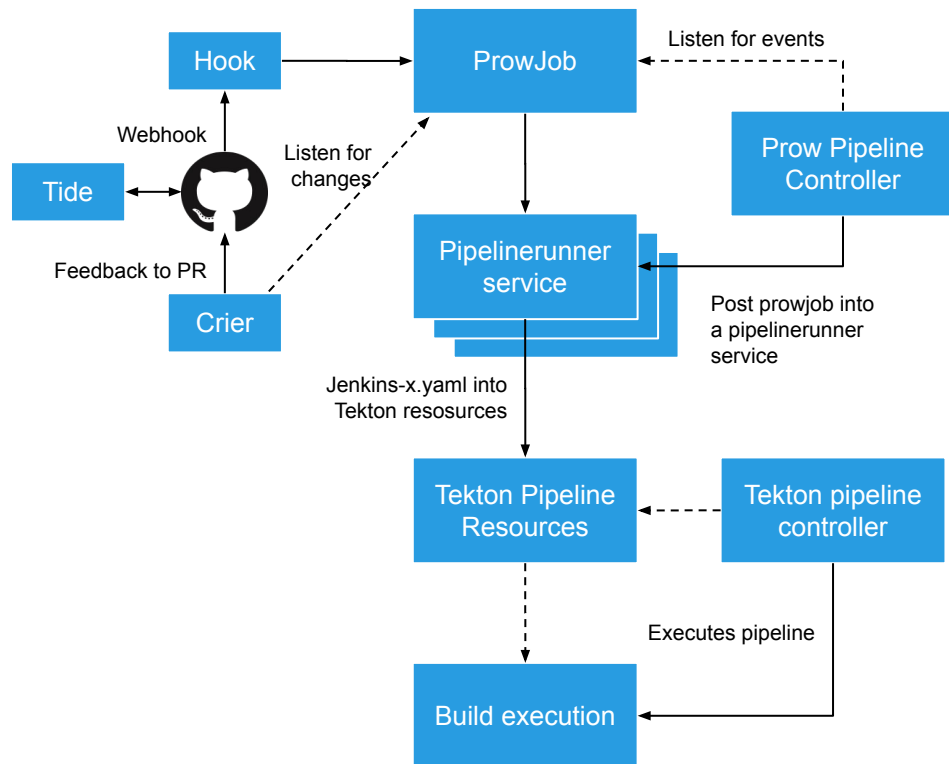
- IaC
- Automated promotions
- Git as source of truth

GitOps environments and promotion



Jenkins X Serverless Experience

Tekton based solution



Tekton pipelines

No Jenkins

Prow Webhook handler





Thank You!

cvanballegooijen@cloudbees.com

@cvanball