cloudbees®

CloudBees DAYS

David Cañadillas
EMEA Solutions Architect and Technologist

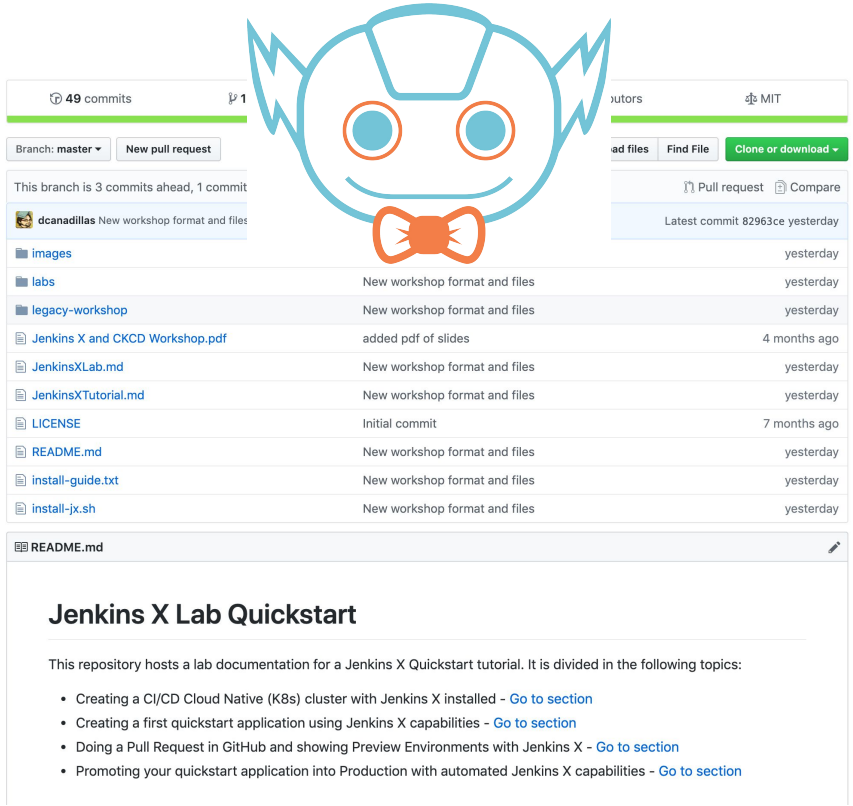
“I love software tech and riding my Harley”

Agenda

Introduction

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

Jenkins X Workshop



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 GitHub[1]) or Lighthouse for Git 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?

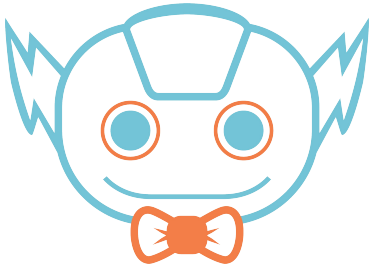
- Are you an expert in kubernetes?
- Do you know how to create helm charts?
- Do you understand the intricacies of Prow and Tekton?
- How about Kaniko, Skaffold, Chartmuseum, Ksync, ...?
- Do you know how to configure GitOps and GitChat?
- Do you expect everyone in your organization to know all that?
- 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/

A Next Generation Cloud Native CI/CD

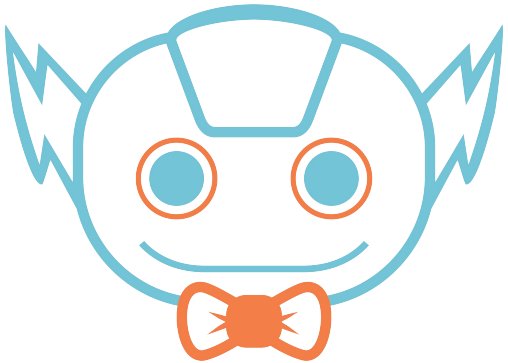
- *Kubernetes native*
- *GitOps promotions*
- *Vault*
- *Tekton CI/CD engine*
- *Multi-cluster*
- *Extensible pipelines*
- *Build packs*



- *ChatOps integration*
- *Serverless*
- *CloudBees Distribution*
- *Preview Environments*
- *Multi-Cloud*
- *Easy on-boarding*
- *Jenkins Apps*

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

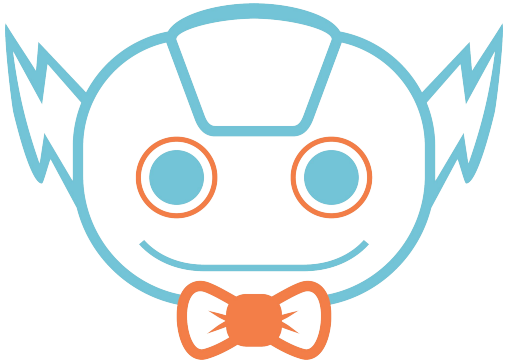
CI/CD steps can run anywhere

Prescriptive best practices

vs

Ultimate Flexibility

But they share some things

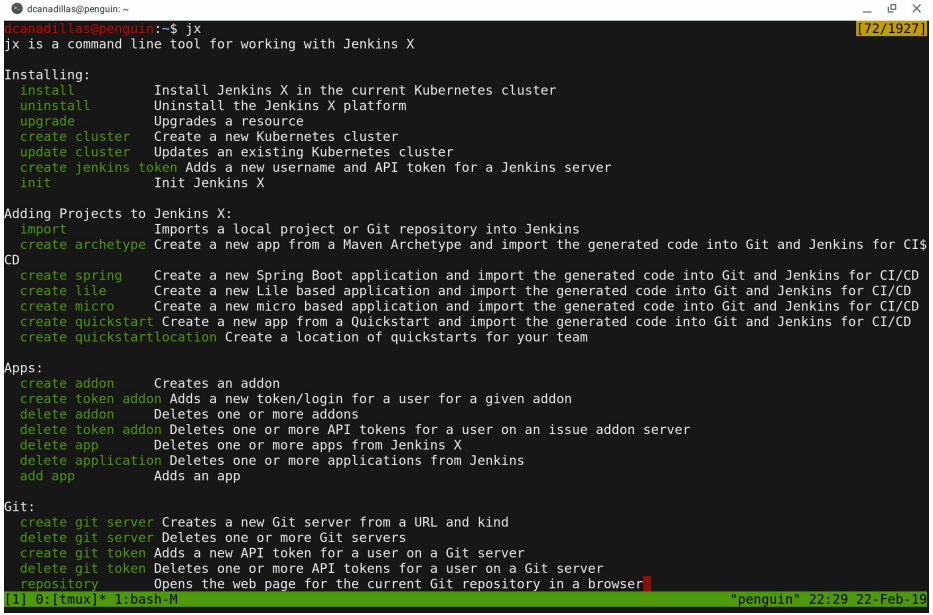


- **CloudBees** is the main contributor
- **Leading CI/CD** solutions
- Fast growing **open source** projects
- Can both run on **Kubernetes**

“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

...
```

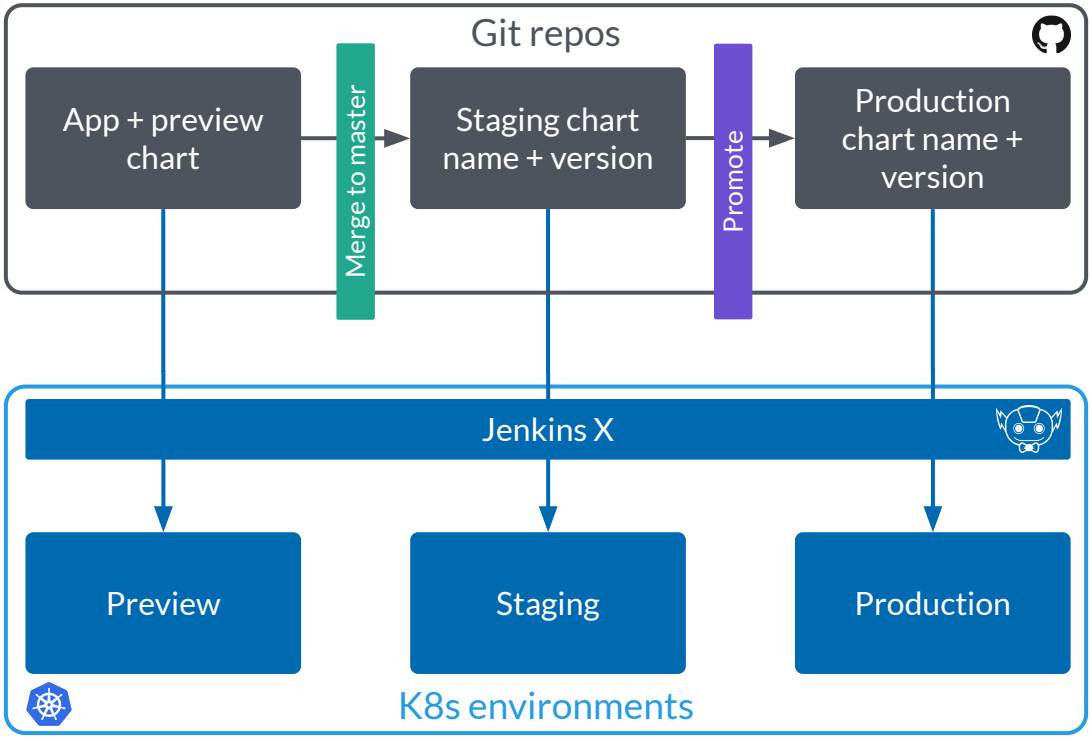


Promotion automation

Thinking about true DevOps…
"***Continuous Delivery meets Cloud Native***"

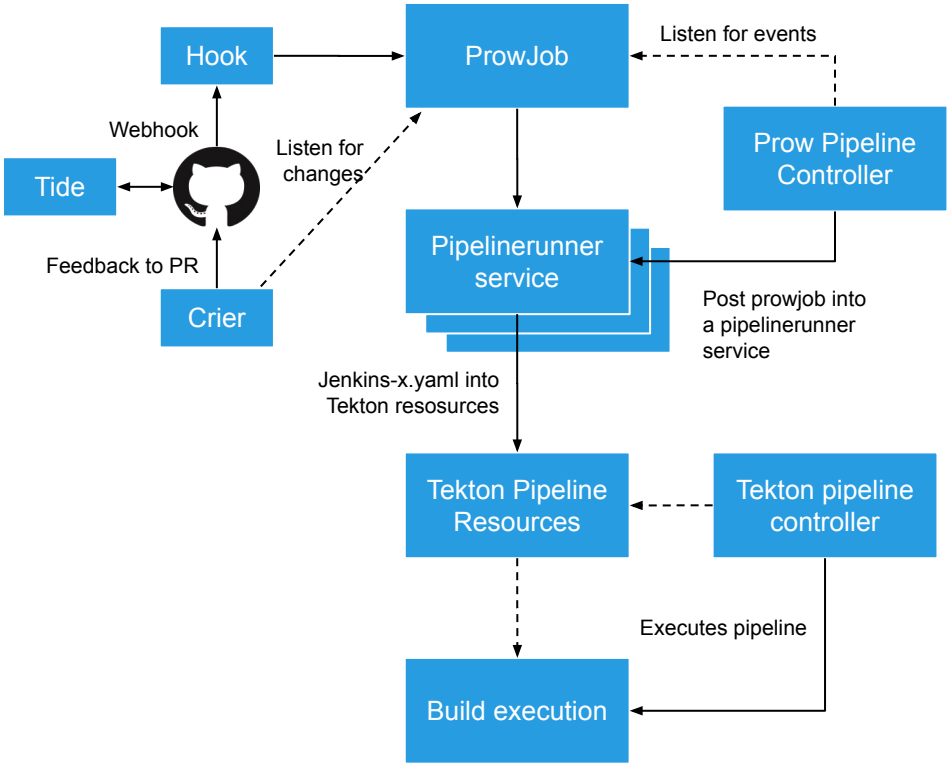
- ➔ IaC
- ➔ Automated promotions
- ➔ Git as source of truth

GitOps environments and promotion



Jenkins Serverless experience

Tekton based solution



Tekton pipelines

No Jenkins

Prow Webhook handler

Kubernetes Workloads
vs
Library Workloads

CI/CD or CI + Release

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

Developers to develop

How? - Best Practices for Native K8s CD

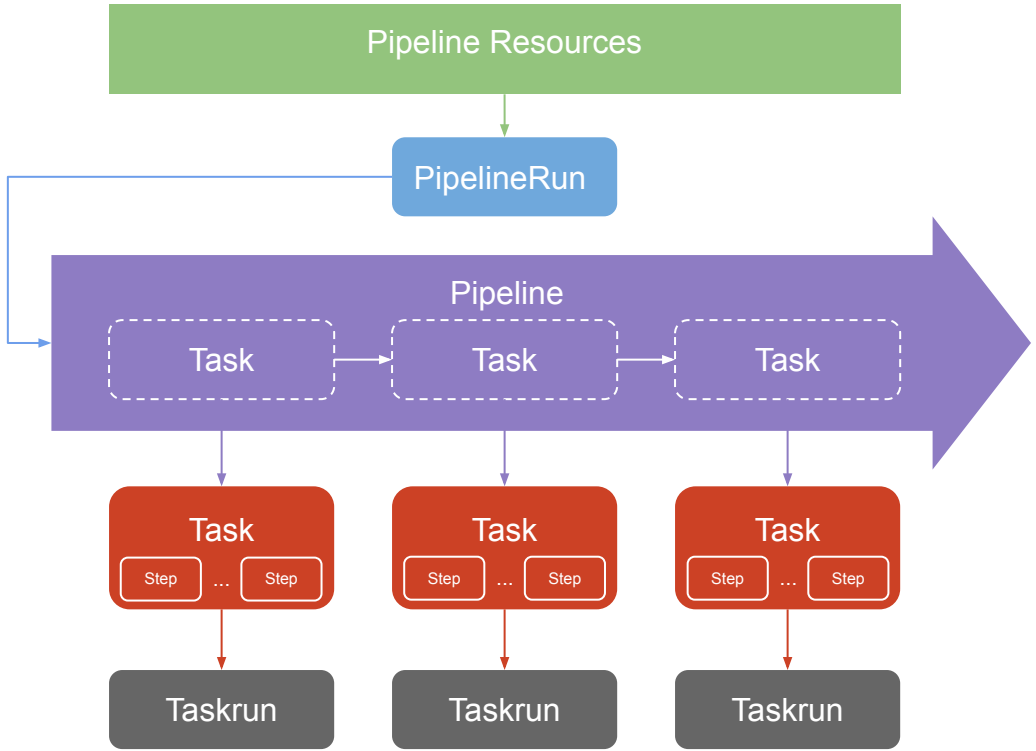
- Based on Azure Draft Packs
- Jenkins X provides two sets of build packs:
  - jenkins-x-kubernetes
  - jenkins-x-classic
- The K8s workload build packs extend the classic build packs
- Same build packs used for static and serverless
- Build packs are extensible

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:

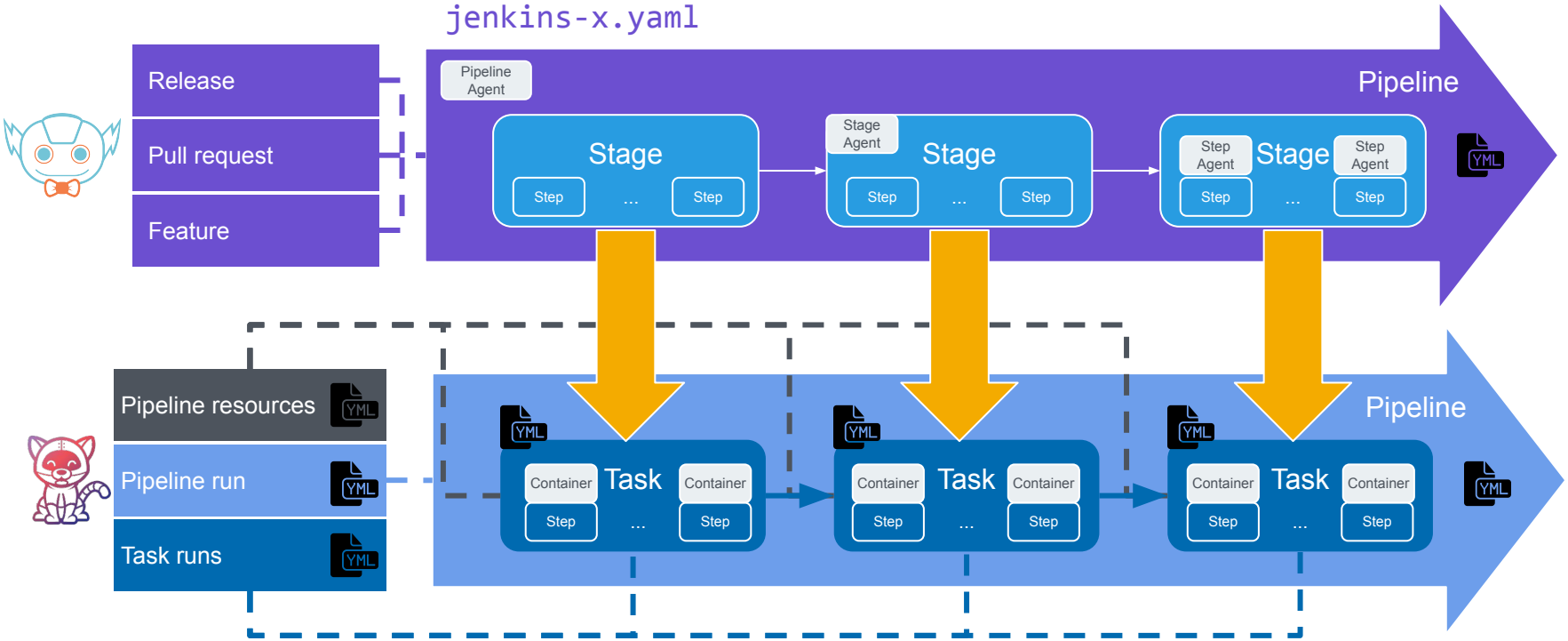
```
jx edit buildpack \
    -n kubernetes-workloads \
    -u https://github.com/{FORKED-ORG}/jenkins-x-kubernetes \
    -r master -b
```

What are Tekton Pipelines



- Containers as building blocks
- Deploy on any K8s cluster
- Decouples stages into tasks
- Decouples pipeline resources
- Cloud Native CRDs

Decoupling Jenkins X Pipelines into Tekton



*Technologists post: http://bit.ly/2kzusFz

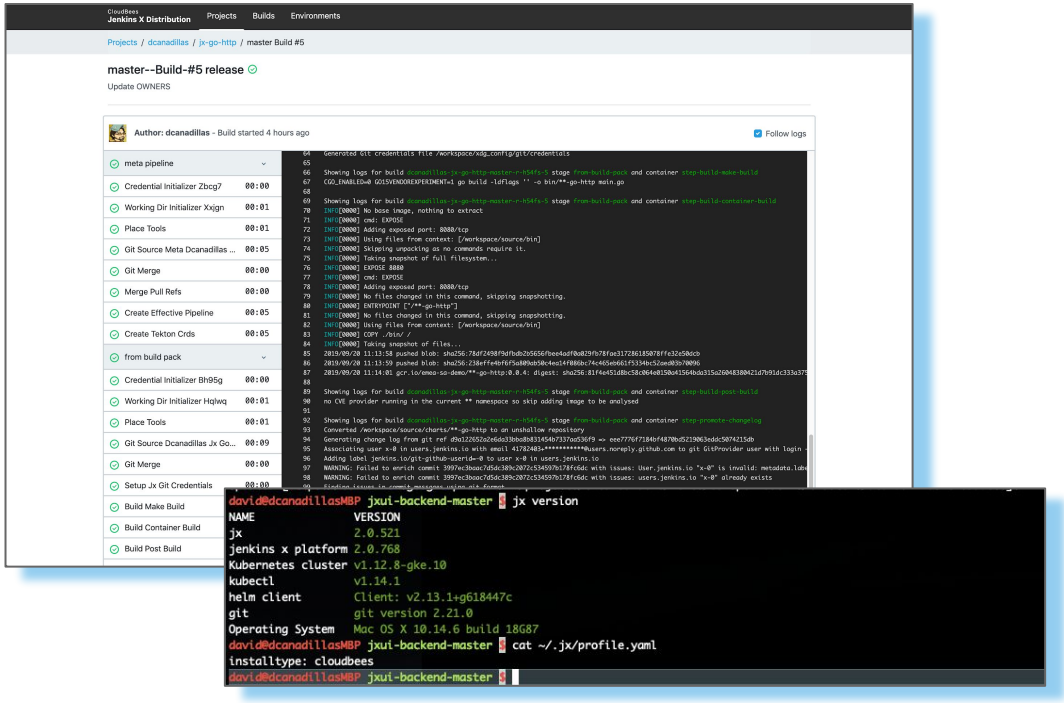
One more thing...

But is this production ready?

CloudBees Jenkins X Distribution

https://www.cloudbees.com/products/cloudbees-jenkins-x-distribution

- *Integrated Vault security*
- *Certified distribution*
- *Optional CloudBees support*
- *Stable upgrades*
- *GKE certified and more coming*
- *Releasing a Jenkins X UI*



The Workshop

Jenkins X Quickstart in GKE

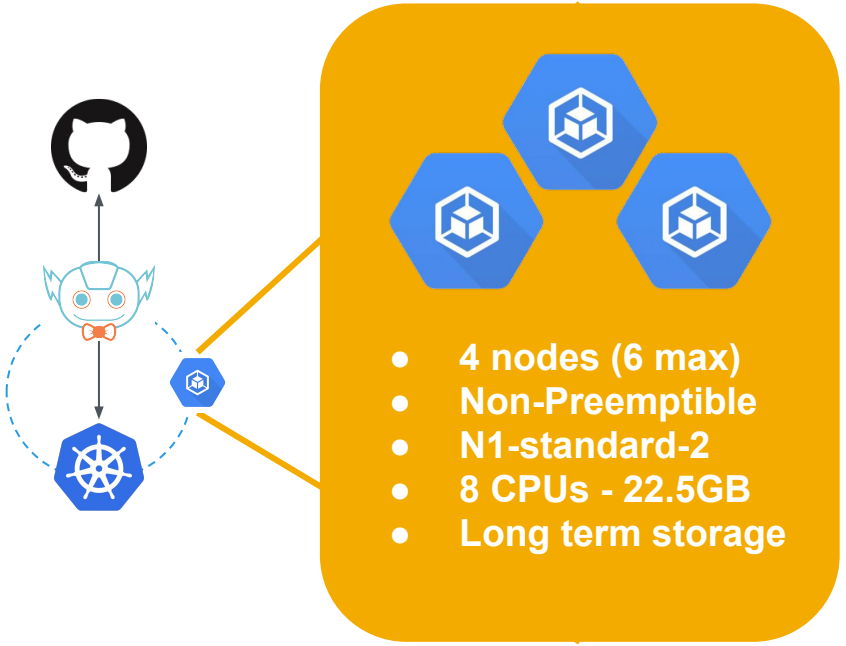
Start developing with Jenkins X

What you'll do

- Create a GKE cluster and install Jenkins X
- Use K8's GKE to deploy your application
- Create a Quickstart Project from a build pack
- Leverage Preview Environments for Pull Requests
- Promote your changes into Production

What you'll need

- GCP account (Free Tier can apply)
- Create a GitHub user and Org
- Internet access (Chrome recommended)



GitHub Workshop*: https://github.com/dcanadillas/jenkins-x-workshop

* The CloudBees original repo is outdated, and this one is an updated work in progress forked from the original

The goal

This workshop goal is to **understand** Jenkins X components as well as the starting default workflow.

So:

- Follow every step of the lab
- Think about every cli command and type it
- Use "`jx <command> --help`"
- Work with your teammates
- Ask any question you have
- Share > Ask > Share

1 hour to complete
But... You can follow!

Do you have your own GCP account?

Be my guest!!

Forked repo *: https://github.com/dcanadillas/jenkins-x-workshop

* This GitHub repo is a work in progress than may be different from the CloudBees Workshop

CloudBees DAYS

Thank You!

dcanadillas@cloudbees.com