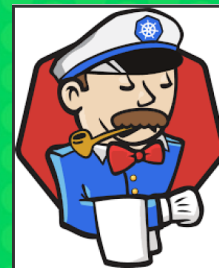


DevOps Pipelines with Jenkins X on SUSE CaaS Platform

Best Practices Basics

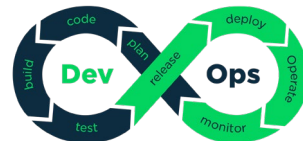
Jean Marc Lambert & Stephane Nguyen
EMEA SUSE Consulting

JENKINS X



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

Landscape for DevOps and Containers



DevOps teams:

- Love containers
- Start to adopt « micro-services » architectures
- Need automated CI/CD pipelines and more...
- Require multiple isolated environments (Dev/Stag/Prod)
- Dream about developing in a « iso-prod » environment
- Need global configuration management of their deployed workloads

Kubernetes is today's best-of-breed container orchestration ecosystem (especially SUSE CaaS Platform).

Jenkins X answers most of the DevOps requirements above.

Jenkins X Introduction

A reimagined **CI/CD** implementation for the cloud

Based on the **state of the art** in DevOps

And the “Accelerate” book recommendations

STARTED BY JAMES STRACHAN

(ex RedHat, now CloudBees)

Evolution of Fabric8 (2015-2017)

JX 1st Release by March 2018



Capabilities of Jenkins X

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



Use version control for all artifacts.



Automate your deployment process.



Use trunk-based development.



Implement continuous integration.



Implement continuous delivery.



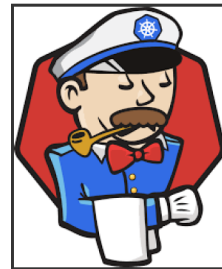
Use loosely coupled architecture.



Architect for empowered teams.

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

How Jenkins X Solves All of These



Jenkins X provides:

- JX CLI tool to interact with
- Kubernetes deployment and configuration tool (multi-clouds)
- « CI/CD/DevOps » tooling using Git, Jenkins, Skaffold, Helm...
- Simple/efficient continuous delivery and promotion workflow
- Automation setup of various environments (Dev/Stag/Prod...) in Git and Jenkins, with relevant pipelines and integration points
- Multiple languages pipelines and quickstarts (Go, Java, Node, Python...)
- Environments to develop in an Iso-Prod mode
- *Based on open source components*



kubernetes

Which Open Source Components Are in Jenkins X?

Jenkins

CI/CD

CI/CD pipeline solution

Git family

Config

Code Versioning/Config

Draft (draft.sh)

Build

Applications build packs

Scaffold

Docker images builder
on Kubernetes

Helm

Deploy

Package manager for Kubernetes

Chartmuseum

Helm chart repository

Monocular

Web UI for Helm charts

Kubernetes

Container orchestrator

Nexus

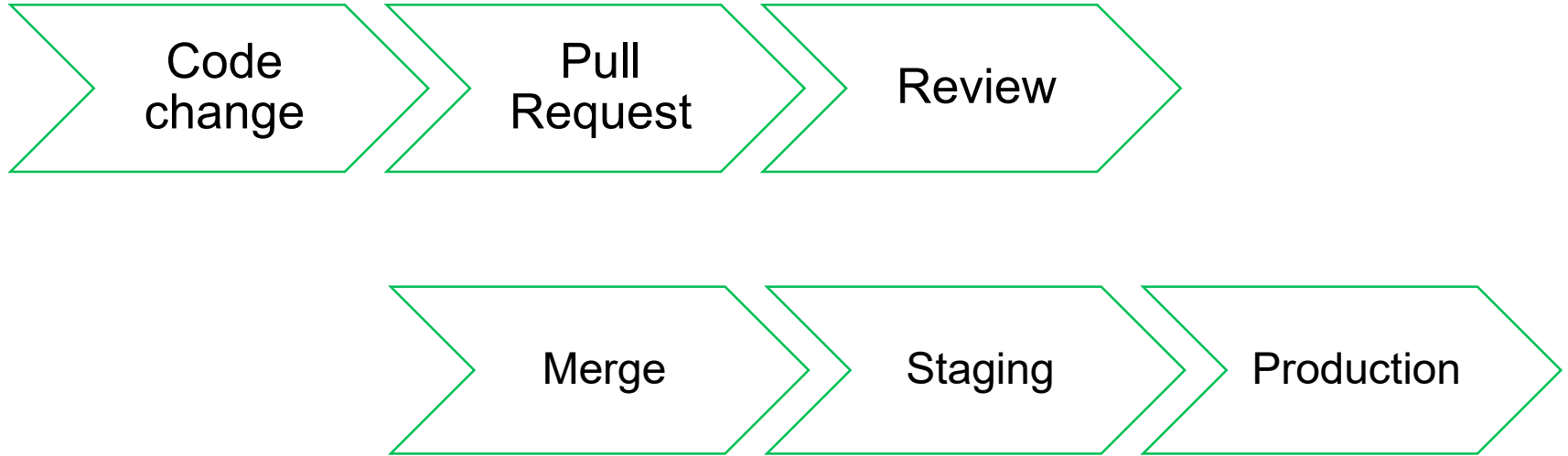
Store

Artifact repository

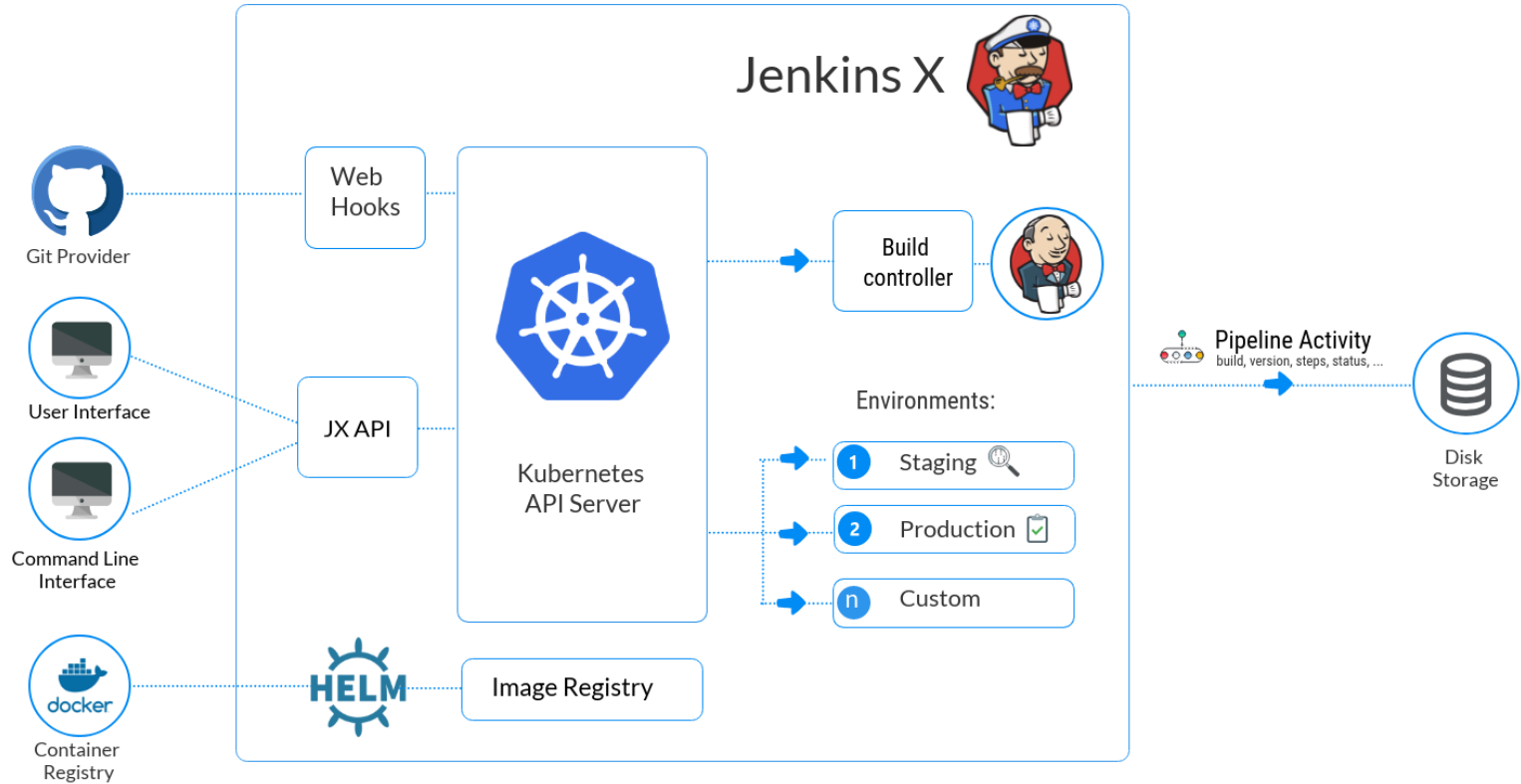


kubernetes

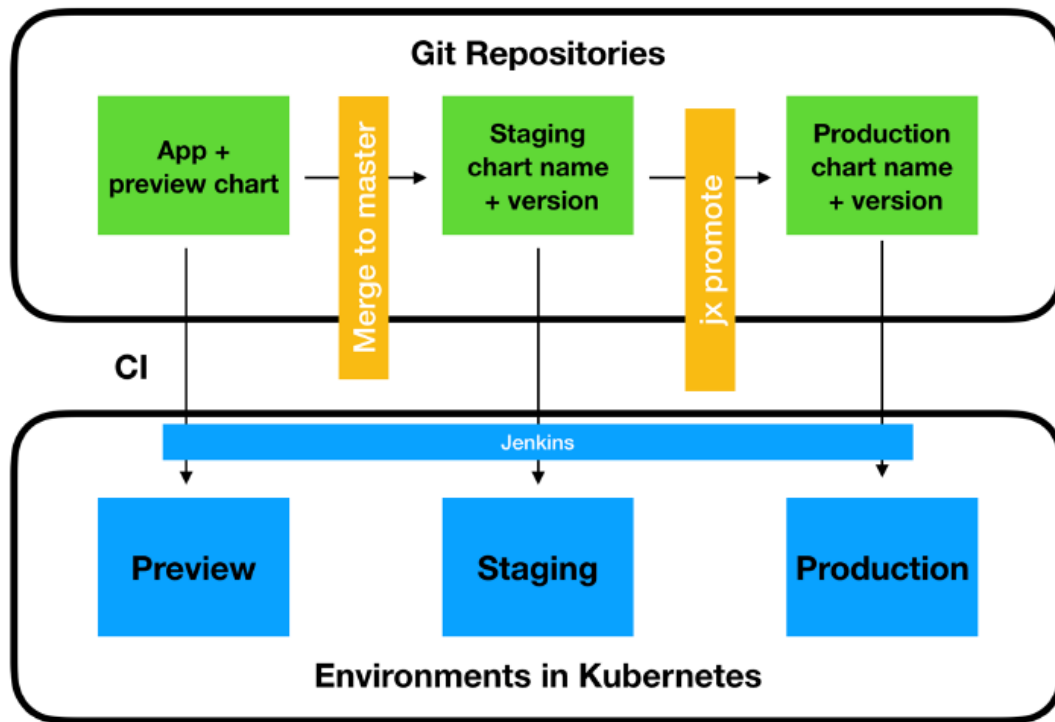
Jenkins X Supports What We Do!



Architecture of Jenkins X

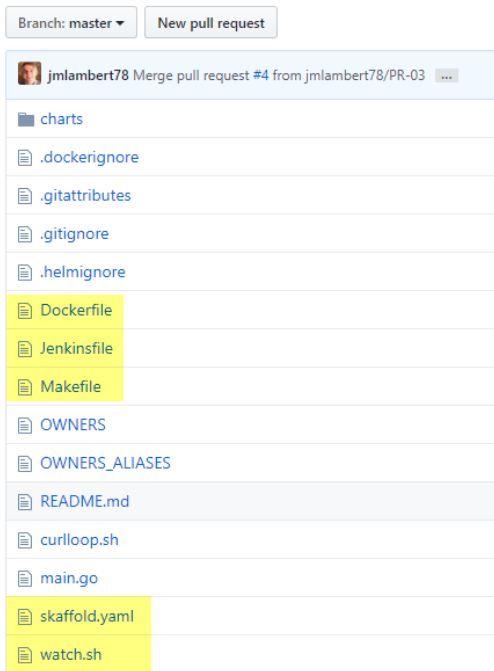


Overview of Git + Jenkins X environments



Structure of a Jenkins X Supported Project

In the Application Project Git



Files used by JX

Dockerfile: image specification

Jenkinsfile: pipeline code

Skaffold.yaml: how to build

Makefile: actions used in the pipelines

Charts: Helm charts for app

JX Deployment Environment Stored in Git

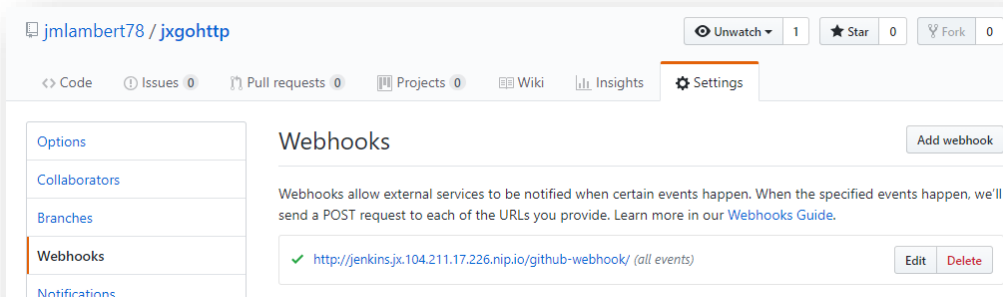
Example: Staging environment

The screenshot displays a Git repository interface for the 'environment-bellychocolate-staging' repository. The main view shows the file structure of the 'env' directory, which includes files like .gitignore, Jenkinsfile, LICENSE, Makefile, and README.md. A pull request is visible, titled 'Merge pull request #9 from jmlambert78:env', with a commit message 'Initial commit'. A green bracket highlights the 'env' directory and the pull request. To the right, a detailed view of the 'requirements.yaml' file is shown, listing dependencies for the staging environment. The file is titled 'environment-bellychocolate-staging / env / requirements.yaml' and shows a commit by 'jenkins-x-bot' with the message 'Promote node-http to version 0.0.1'. The file content is as follows:

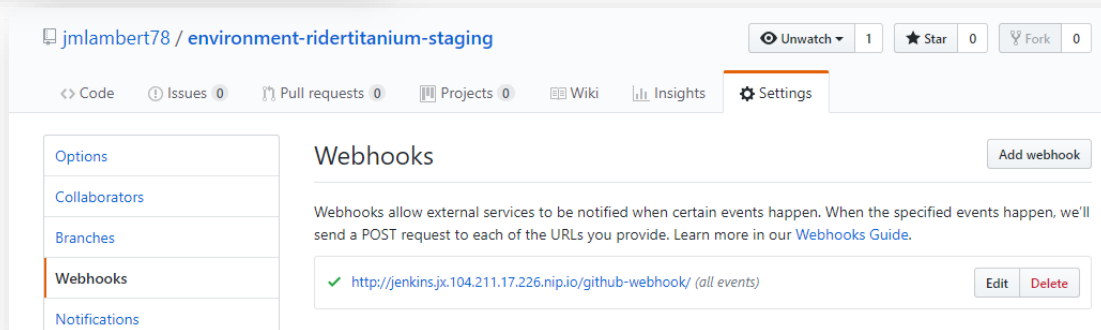
```
1 dependencies:
2   - alias: expose
3     name: exposecontroller
4     repository: http://chartmuseum.jenkins-x.io
5     version: 2.3.89
6   - alias: cleanup
7     name: exposecontroller
8     repository: http://chartmuseum.jenkins-x.io
9     version: 2.3.89
10  - name: golang-kubedok
11    repository: http://jenkins-x-chartmuseum:8080
12    version: 0.0.9
13  - name: node-http
14    repository: http://jenkins-x-chartmuseum:8080
15    version: 0.0.1
16  - name: node-http-watch-pipeline-activity
17    repository: http://jenkins-x-chartmuseum:8080
18    version: 0.0.1
```

Git Webhooks to Jenkins Instances

App side



Env(s) side



Jenkins Deployed and Pipelines Set Up

The screenshot shows the Jenkins web interface. At the top, the Jenkins logo is on the left, a red box with the number '4' in the center, and a search bar on the right. Below the header, the breadcrumb 'Jenkins > jmlambert78' is visible. On the left sidebar, there are navigation links: Up, Status, Configure, New Item, Delete Folder, People, Build History, Project Relationship, Check File Fingerprint, Open Blue Ocean, Rename, Config Files, Credentials, and New View. The main content area displays the 'jmlambert78' folder. Below the folder name is a table of builds. The table has columns: S (Status), W (Weather icon), Name, Last Success, Last Failure, Last Duration, and Fav (Favorite). There are four builds listed, all with a 'Success' status (green sun icon). Below the table, there is a 'Legend' section with RSS links for all builds, failures, and latest builds. At the bottom left, there are two panels: 'Build Queue' (empty) and 'Build Executor Status' (showing a build 'go-s77lp' with a progress bar).

Jenkins

4

search

admin | log out

Jenkins > jmlambert78

ENABLE AUTO REFRESH

Up

Status

Configure

New Item

Delete Folder

People

Build History

Project Relationship

Check File Fingerprint

Open Blue Ocean

Rename

Config Files

Credentials

New View

jmlambert78

add description

All +

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		environment-jayscratch-production	9 min 9 sec - log	N/A	1.8 sec	
		environment-jayscratch-staging	8 min 9 sec - log	N/A	1.3 sec	
		go-kube-http	38 sec - log	N/A	1.7 sec	
		node-kube-http	9.4 sec - log	N/A	0.91 sec	

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Build Queue

No builds in the queue.

Build Executor Status

go-s77lp









1 jmlambert78 » go-kube-http » #1
master (Build Release)

Jenkins Pipelines Views




Jenkins


Pipelines Administration  Logout


Pipelines  Search pipelines... [New Pipeline](#)




NAME	HEALTH	BRANCHES	PR
jmlambert78 / environment-jayscratch-production		1 passing -	
jmlambert78 / environment-jayscratch-staging		1 passing -	
jmlambert78 / go-kube-http		- -	
jmlambert78 / node-kube-http		1 passing -	

1.9.0

✓ **jmlambert78 / node-kube-http 1** Pipeline Changes Tests Artifacts    Logout X

Branch: master  24m 26s No changes
Commit: 48a9e9f 9 minutes ago Branch indexing



Promote to Environments - 22m 5s  [Restart Promote to Environments](#)  

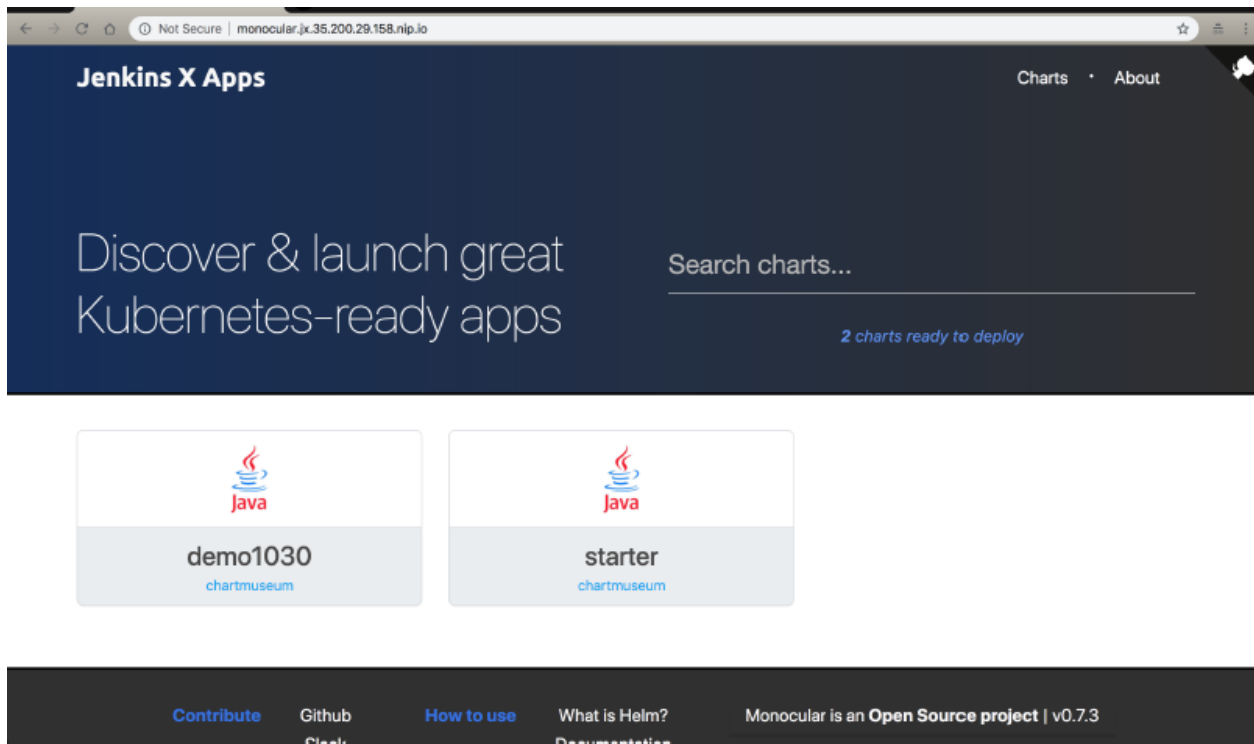
✓ > jx step changelog --batch-mode --version v\$(cat ../VERSION) — Shell Script 7s

✓ > jx step helm release — Shell Script 8s

✓ > jx promote -b --all-auto --timeout 1h --version \$(cat ../VERSION) — Shell Script 21m 50s

✓ > Delete workspace when build is done <1s

Monocular GUI: Helm Charts Repository



How Does That Work on the SUSE CaaS Platform?

What Is the SUSE CaaS Platform?

- A Kubernetes cluster with an additional layer of management
- Currently at version 3



kubernetes

CaaS Platform – Kubernetes

- Supports Docker and CRI-O as container runtimes



- Provides a “vanilla” Kubernetes

- Supports all features that Kubernetes 1.10 supports



kubernetes

- Can run on-premises, on physical servers or virtual machines, and also in the Public Cloud
- RBAC is the de facto authorization mechanism
- Can integrate with external LDAP/Active Directories for users and groups management

CaaS Platform – The Management Layer

- **Supports deployment and ongoing CaaS Platform management**
 - Automated cluster bootstrapping
 - Patch management with no service interruption
 - Certificate distribution, private registries addition, external LDAP integration, etc.
 - User tokens management
 - Orchestration performed by Salt
- **Provides a graphical user interface to perform these tasks: Velum**
- **Runs on a dedicated server: the Administration Server**

The screenshot displays the SUSE CaaS Platform Velum web interface. The top navigation bar includes 'SUSE CaaS Platform', 'Home', 'Settings', and a 'Logout' button. The main section is titled 'Cluster Status' and contains a 'Summary' box with the following data:

Summary	
Total nodes	3
Master nodes	1
New nodes	0
Updates	Manual
# of nodes w/ outdated software	0

Below the summary is a 'Nodes' section with a 'kubeconfig' download button. It contains a table of cluster nodes:

Status	ID	Hostname	Role	Actions
✓	2967ad5a38264351ba5ef0cc275f152a	node1	<input type="checkbox"/> worker	Remove
✓	4dc8659c7fc749f1a80aa65ca47967df	master	<input checked="" type="checkbox"/> master	
✓	4a86697219eb4c14a2afa46182aa4bfb	node2	<input type="checkbox"/> worker	Remove

CaaS Platform – Typical Configurations

- Minimal

- 1 Administration Server
- 1 Master node
- 2 Worker nodes

- Typical Configuration

- 1 Administration Server
- 3 Master nodes
- 2..n Worker nodes

- Persistent Storage

- Dynamically or statically provisioned NFS PVs
- Dynamically provisioned Ceph RBD PVs

Jenkins X Prerequisites – On-Premises

- **Kubernetes cluster (SUSE CaaS Platform) with RBAC enabled**
 - Size, 3 VMs: 60Gb disk, 2-3v CPU, 8Gb RAM
- **INGRESS router**
 - Deployed with Helm on the Kube
- **Fixed IP for the Ingress**
- **Wildcard DNS entry for this Ingress IP**
 - Reachable from github to allow the webhooks to operate
- **GITHUB account (or git family)**
- **Storage-class: to create on-demand Persistent Volumes (PV)**
 - « NFS Client Provider » can make the job or « **SUSE Enterprise Storage** » RBD
- **Docker Setup: (if using jx deployed registry)**
 - `DOCKER_OPTS="--insecure-registry <clusterRange.0.0/16"`

Deployment of Jenkins X on SUSE CaaS Platform

Install the command line tool:

jx



<http://jenkins-x.io/>

If you have a K8S cluster already

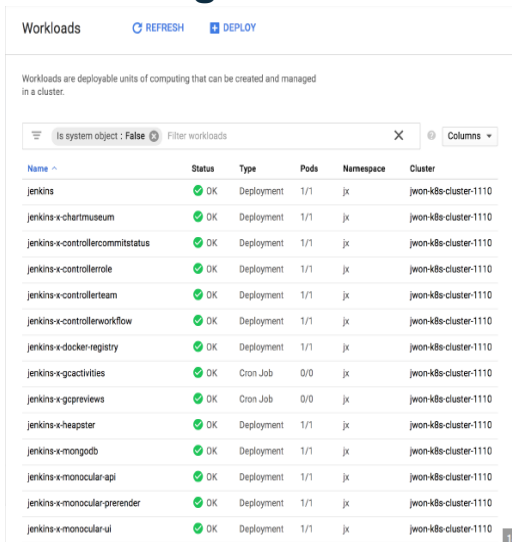
- Ensure RBAC is enabled and then:

```
jx install --provider=kubernetes  
--on-premise --skip-ingress
```

What Do You Get Out of That?

- Automates the **installation/upgrade of tools**
 - Helm, Scaffold, Kaniko, Jenkins, KSync, Monocular, Nexus, etc.
 - All configured + optimized for Kubernetes OOTB

Jenkins X components running in the CaaS



The screenshot shows the 'Workloads' page in a Kubernetes dashboard. At the top, there are 'REFRESH' and 'DEPLOY' buttons. Below them is a description: 'Workloads are deployable units of computing that can be created and managed in a cluster.' A filter bar shows 'Is system object: False' and 'Filter workloads'. A table lists various Jenkins X components, all with a status of 'OK' and located in the 'jx' namespace across the 'jwon-k8s-cluster-1110' cluster.

Name	Status	Type	Pods	Namespace	Cluster
jenkins	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-chartmuseum	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-controllercommitstatus	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-controllerrole	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-controllerteam	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-controllerworkflow	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-docker-registry	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-gcactivities	OK	Cron Job	0/0	jx	jwon-k8s-cluster-1110
jenkins-x-gcpreviews	OK	Cron Job	0/0	jx	jwon-k8s-cluster-1110
jenkins-x-heapster	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-mongodb	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-monocular-api	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-monocular-prerender	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110
jenkins-x-monocular-ui	OK	Deployment	1/1	jx	jwon-k8s-cluster-1110

Staging + Prod Envs in Github

Git staging and production repos

- Ready to welcome new apps

Jenkins setup

- Pipelines set up for these repos

WebHooks

- Set in Github to trigger Jenkins on change or PR

What Does That Give Me?

- The Jenkins X CLI tool integrates a lot of features (deploy, logs, etc.)
- Ability to work with multiple k8s contexts (clusters)

Each team gets its own:

- **Development Tools Environment**
 - Jenkins master
 - Elastic pool of Kubernetes build pods
 - Nexus for artifacts
 - ChartMuseum + Monocular (Helm application store + API + UI)
 - Private Docker registry
- **Staging + Production Environment (default)**
 - Both environments configured in Git for version control and triggers to Jenkins on changes

Jenkins X Components

Environments / Kubernetes Namespaces

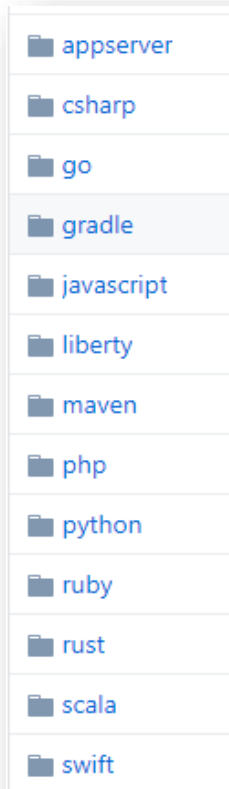
- 1 Dev Env per team/project
- 0-N Permanent Environments (e.g.: staging, production)
Each permanent env is managed within Git & PR to allow traceable changes / Rollbacks
- Optional preview environments allocated to preview PRs

Components deployed by Jenkins X

- Jenkins
- Nexus
- Docker registry
- Chartmuseum (registry for helm chart)
- Monocular: UI for discovering/running Helm charts

Jenkins X Provides Languages Buildpacks

- **For each supported framework, a buildpack includes:**
 - Dockerfile
 - Jenkinsfile
 - Helm chart (used to deploy it on k8s)
 - Chart/preview Helm chart used to deploy in a preview environment
- **You may write your own if it's not yet available**
- **A set of pod templates is provided**
 - To execute the Jenkins pipeline itself
 - Including all the tools required by a standard pipeline of the framework



Jenkins X CLI: Create/Import Projects

- **Available commands:**
 - `jx create spring`
 - `jx create quickstart`
 - `jx import`
- **Automatically set up CI/CD pipelines for new + imported projects**
 - Set up git repository
 - Adds webhooks on git provider to trigger Jenkins pipelines on PR / master
 - Triggers first pipeline

Jenkins X: On Pull Request

- **Compiles + runs tests**
- **Creates preview Docker image + Helm chart**
- **Creates a preview environment**
- **Comments on the PR in Git with a link to the running app**



test branch #1

[Open](#) jmlambert78 wants to merge 1 commit into `master` from `tb`

Conversation 1 Commits 1 Checks 0 Files changed 1



jmlambert78 commented 5 minutes ago

Owner + 👤 ...

launch the pr

test branch

✓ d8f502e



jmlambert78 commented 2 minutes ago

Owner + 👤 ...

★ PR built and available in a preview environment [jmlambert78-gkejxgohttp2310-pr-1 here](#)

Add more commits by pushing to the `tb` branch on [jmlambert78/gkejxgohttp2310](#).



✓ All checks have passed
1 successful check

[Show all checks](#)

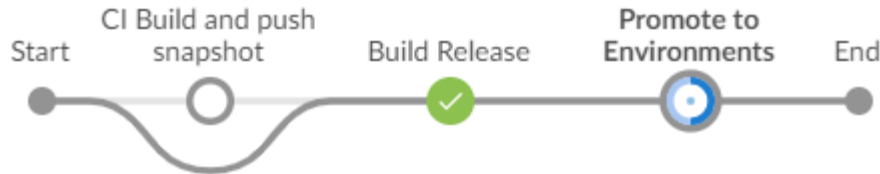
✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

[Merge pull request](#)

You can also open [this in GitHub Desktop](#) or view [command line instructions](#).

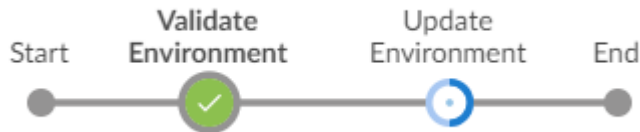
Jenkins X: On Release (Push to Master)

- **Compiles + runs tests**
- **Creates semantic release version**
- **Publishes versioned artifacts, Docker image and Helm chart**
- **Promotes through all *automatic* environments, e.g., Staging**



Jenkins X: Promotion via GitOps

- **Each environment stores its configuration as Helm charts in a Git repository**
 - Configuration as code
 - All changes audited and easy to revert
 - Reuse the Pull Request workflow for changes
- **To promote a version to Production, for example, Jenkins X submits a Pull Request**
 - The Promote step waits for the Pull Request CI build to complete and merge, and for the environments pipeline to finish applying the change



Example: Staging Environment History (Git)

Search or jump to... Pull requests Issues Marketplace Explore

jmlambert78 / environment-jayscratch-staging

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master

Commits on Feb 11, 2019

Merge pull request #3 from jmlambert78/promote-go-kube-http-0.0.2

jmlambert78 committed an hour ago

1

Verified



e3f6621



Promote go-kube-http to version 0.0.2

jenkins-x-bot committed an hour ago



e02e000



Merge pull request #2 from jmlambert78/promote-go-kube-http-0.0.1

jmlambert78 committed an hour ago

Verified



3b82d6f



Promote go-kube-http to version 0.0.1

jenkins-x-bot committed 2 hours ago



74d7758



Merge pull request #1 from jmlambert78/promote-node-kube-http-0.0.1

jmlambert78 committed 2 hours ago

Verified



33b17aa



Add environment configuration

jmlambert78 committed an hour ago



ea8fa57



Use correct namespace for environment

jmlambert78 committed an hour ago



974bcf7



Promote node-kube-http to version 0.0.1

jenkins-x-bot committed 2 hours ago



3a8f49c



Branch: master

environment-jayscratch-staging / env / requirements.yaml

jenkins-x-bot Promote go-kube-http to version 0.0.2

2 contributors

Executable File 16 lines (15 sloc) 399 Bytes

```
1 dependencies:
2   - alias: expose
3     name: exposecontroller
4     repository: http://chartmuseum.jenkins-x.io
5     version: 2.3.89
6   - alias: cleanup
7     name: exposecontroller
8     repository: http://chartmuseum.jenkins-x.io
9     version: 2.3.89
10  - name: go-kube-http
11    repository: http://jenkins-x-chartmuseum:8080
12    version: 0.0.2
13  - name: node-kube-http
14    repository: http://jenkins-x-chartmuseum:8080
15    version: 0.0.1
```

The Cherry on Top: DevPod to Speed up the Development Cycle

- Allows the developer to use the same environment/tooling as the CI/CD pipelines on Kubernetes
- Lets you build, run tests or redeploy apps before you commit to Git, safe in the knowledge that you are using the same tools as the CI/CD pipelines!
- **Developer experience**
 - Works on its own IDE (Eclipse, Vscode, etc.)
 - Each file save is reflected (**ksync**) within the DevPod and triggers a local rebuild and Docker image production through **scaffold** and deployment
 - Dev may also ~ssh to the DevPod to run his investigation or debug
 - No need to install the exact same tools on the Dev workstation



Environment:

SUSE CaaS Platform already deployed

Jenkins X installed with *jx install --provider=kubernetes*

Workstation

jx client available

kubectl credential available

DEMO TIME

Jenkins X Usage Demo

- **Create a new project**
 - `jx create quickstart -f go`
- **Check the activities triggered automatically**
 - `jx get activity -f <projectname> -w`
- **Get the Apps urls**
 - `jx get urls`
- **Connect Jenkins GUI to check progress**
 - Jx console <http://jenkins.jx.35.224.16.222.nip.io/admin/shieldtyphoon>
- **Branches and PR auto review**
- **Update code and perform a full process from source to production**
- **Demo the development process within Kubernetes (faster cycles) and devpod**



Unpublished Work of SUSE LLC. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC. in the United States and other countries. All third-party trademarks are the property of their respective owners.