CI 000

000

Fast track cloud development with JenkinsX

Juan Peredo linkedin.com/in/juanperedotech @JuanPeredoTech



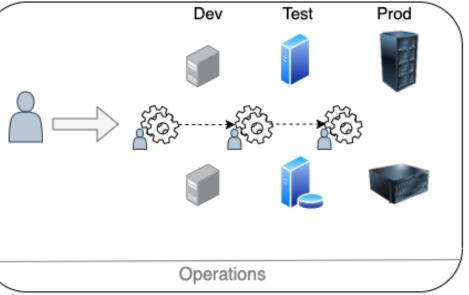
Traditional deployments

Custom scripts nightmare

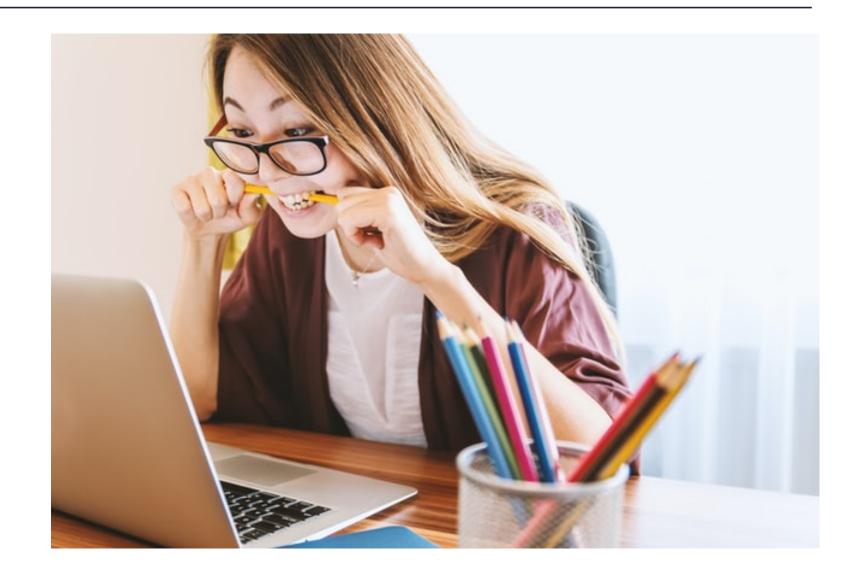
- Multiple custom scripts used for deployments
- Multiple teams involved
- Hard to update and debug
- No visibility

Development	1	





Process may have caused a little stress...





Classic CI/CD

- General purpose tools
- Automated
- Standardized pipeline steps
- Better visibility





Cloud CI/CD

- Automated
- Great traceability and visibility
- Built for re-usability
- Cloud focus



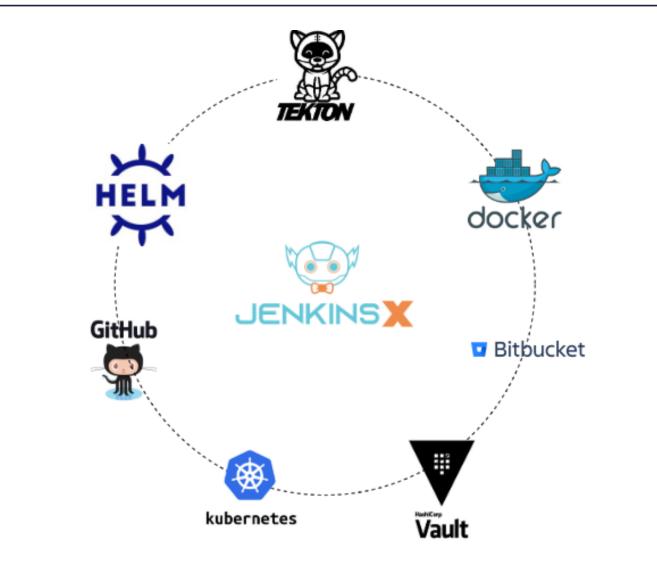






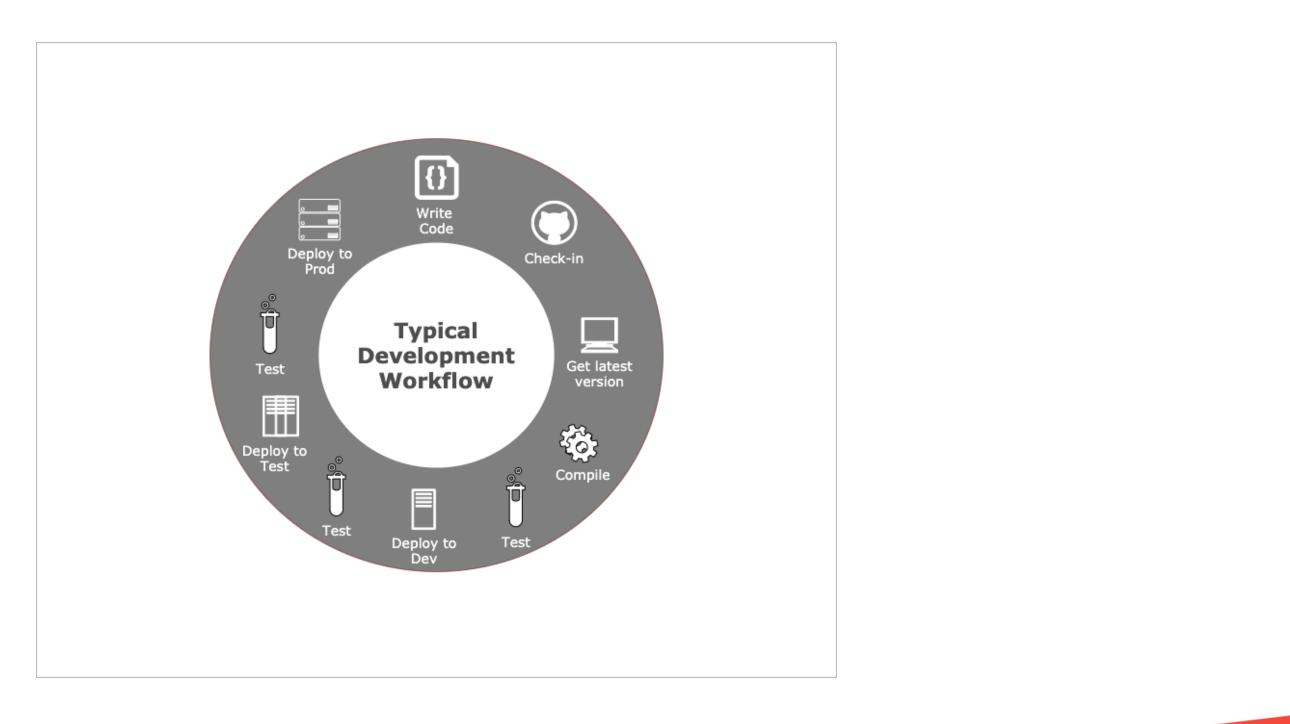
JX add milk to shopping list...

- Jenkins X is more than just pipelines:
 - Ecosystem to simplify everyday tasks
 - Opinionated (In a good way)
 - Cloud Native
 - Flexible and customizable
 - Combine with Terraform for cluster as code



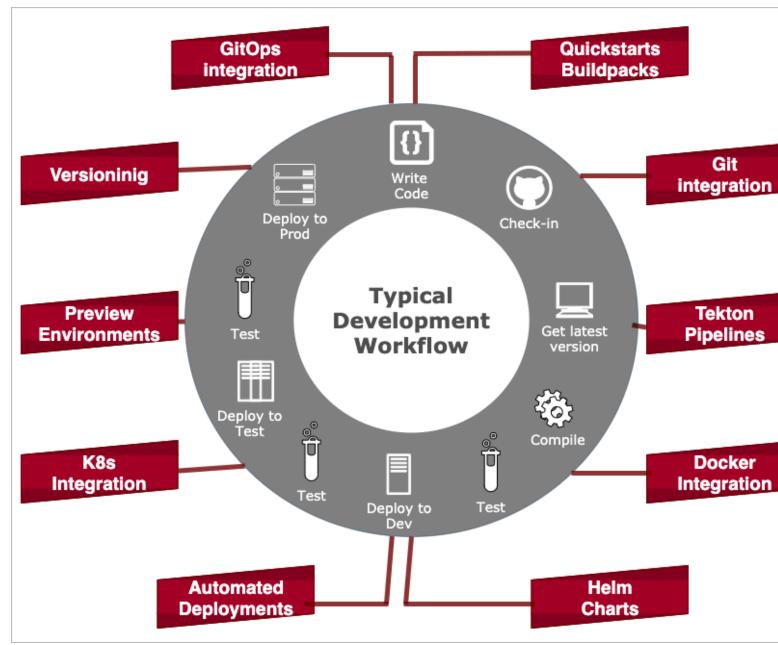


Classic dev workflow





Jenkins benefits







Setup Jenkins X

Git Provider	GCP	Terraform	Jenkins X	Domain & TLS
- Create an account (or use existing account)	 Create new project Setup billing Enable Kubernetes API 	 Install Terraform Create a project folder Create empty main.tf file 	- Install Jenkins X CLI	 Optional but highly recommended step TLS will secure your internet communications
- Create organization to hold Jenkins X repos	 Install Kubectl component Add bucket to hold Terraform state 	 Get jx gcp mmodule from terraform registry Use module instructions to update main.tf 	- Install Helm v2 - Clone the JX Boot repo	 Obtain top level domain from internet registrar Configure cloud provider to use the domain
- Create account to perform Jenkins X actions ("bot")	 Set default user credentials Add storage Admin role to user 	 Initialize Terraform Generate terraform requirements.yml file Create the Cluster 	 Copy requirements.yml to JX Boot repo Run JX Boot to configure Jenkins X 	 Configure Terraform to use the domain Configure Terraform to use the certificates

* Illustrative, refer to official install procedures for full install instructions. Setup on Mac OS, using GCP as cloud provider





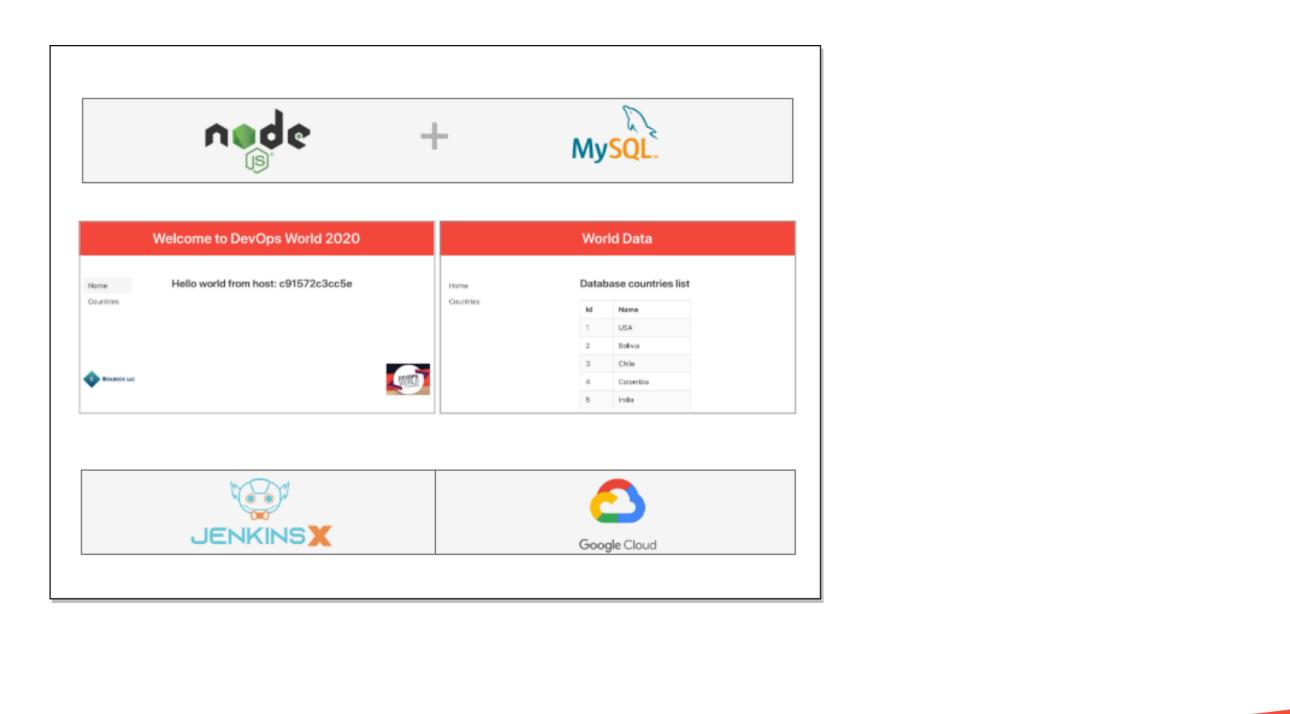
아버렸다는 것은 것은 것을 많이 했다.	Spalanty Security 5 (6-74) Section 5.
And	v2.6.0-beta.2) build: release 2.6.0-beta 2
the second	build: build 2.6.0-beta.2 feat: dynamic directive arguments for v-on, v-bind and custom directive
n flow • packages	origin/dynamic-directive-arguments) feat: dynamic args for custom perfection accuracy (#9371)
 scripts 	test: test cases for v-on/v-bind dynamic arguments
▶ SIC	refactor: v-bind dynamic arguments use bind helper
test	test: fix tests, resolve helper conflict
types	tix: fix middle modifier
.babelrc.js .editorconfig .eslintignore	feat: handle dynamic argument for v-bind.sync feat: handle dynamic argument for v-bind.sync feat: dynamic directive arguments for v-bind and v-on
eslintrc.js flowconfig	feat: dynamic directive digundate refactor: extend dom-props update skip to more fix: fix checkbox event edge case in Firefox
.gitignore BACKERS.md	test: fix tests in IE/Edge refactor: simplify timestance
LICENSE package joon	chore: update contracts

- Goals:
 - Zero to Production
 - Automate every day tasks
 - Everything as code



Demo

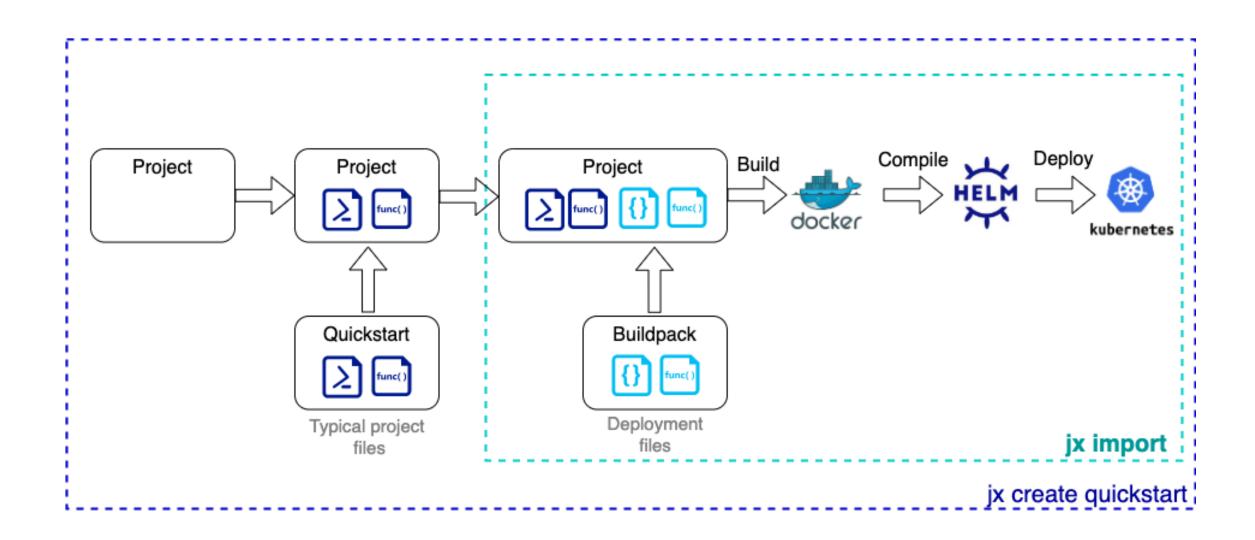
Our application





Initialize project

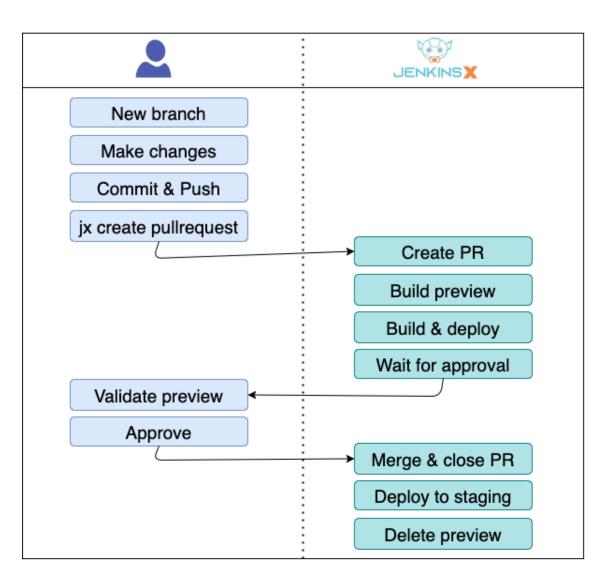
jx create and jx import





Change Management

jx create pullrequest





Deployment management

jx promote

- Deploy manually to prod by default
- Promote a version of app to any environment with jx promote
- Rollback by promoting previous version
- Can be configured to do different types of deployments



Summary

App Code		Chille	
MySQL. Dockerfile	JENKINSX	GitHub	Google Cloud
Initial application			
Init project & first deploy	jx import	Create / update repo	Dev Stage
Pull request	jx create pullrequest	 Create PR Run pipeline Deploy Preview Wait for approval Merge PR Delete Preview 	Dev Stage Prev
Push to production	jx promote	 Create & close PR Create release tag 	Dev Stage Pr







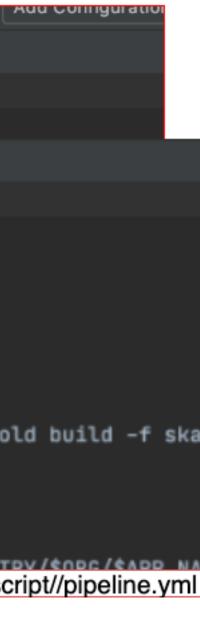


Inherited pipelines

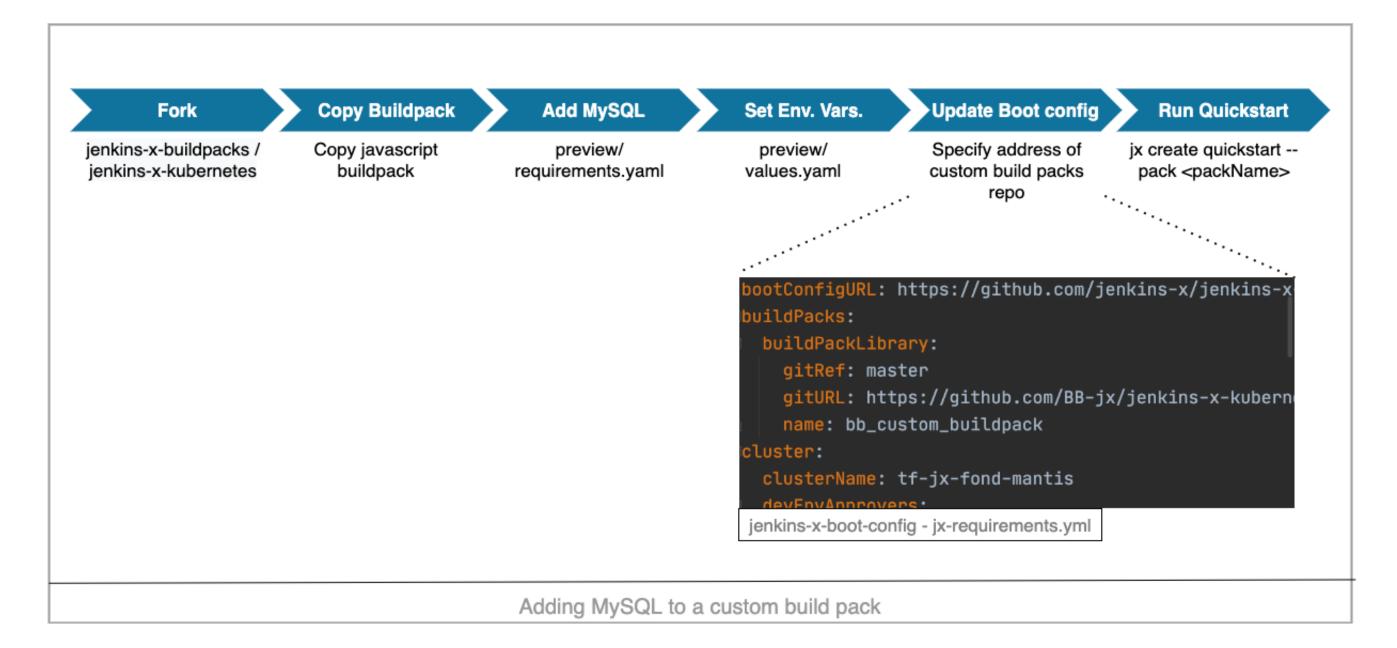
our-project/jenkins-x.yml

kins-x.yml	×
buildF	Pack: javascript
🚛 pipel	ine.yaml ×
1 (extends:
	import: classic
	file: javascript/pipeline.yaml
	pipelines:
	pullRequest:
6 (build:
7 (steps:
8 (- sh: export VERSION=\$PREVIEW_VERSION && skaffold build -f ska
9 (name: container-build
10 (postBuild:
11 (steps:
12 (jenkins-x.kunernetes/javascript//pipeline.yml
	1 (1 2 (1 3 (1 5 (1 7 (1 8 (1 9 (1 10 (1)





Custom build packs





Managing your cloud cost



Cloud deployments are priced per resource usage, time and storage

- Removing an application from Jenkins X
- jx delete application
- Removing nodes from cluster

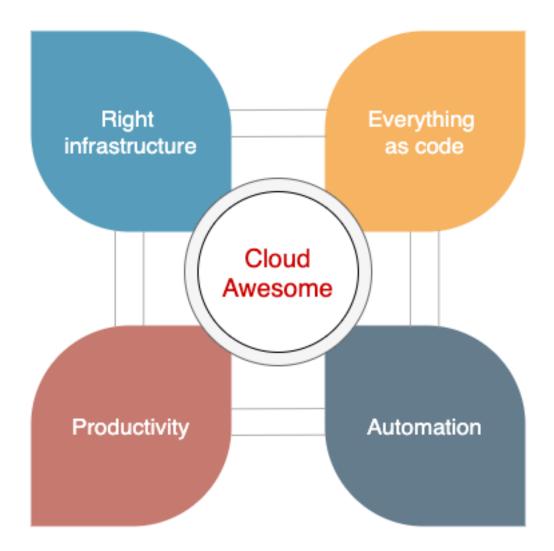
• Destroying the cluster

Terraform destroy



gcloud container clusters resize

Key take aways







Questions ? Thanks to CloudBees & all the conference sponsors



Fast track cloud development with JenkinsX

000

Juan Peredo linkedin.com/in/juanperedotech @JuanPeredoTech



Appendix



Photos

- Photo by Andre Guerra on Unsplash
- Photo by JESHOOTS.COM on Unsplash
- Photo by Yancy Min on Unsplash

