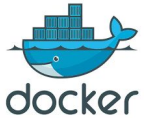
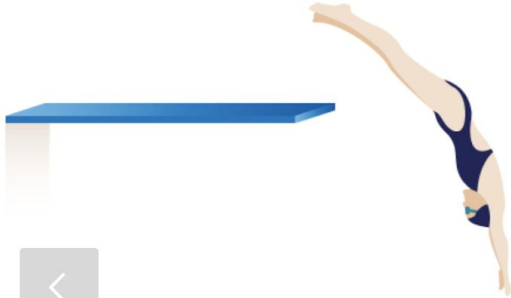


JENKINS CI/CD PIPELINES

WITH KUBERNETES AND HELM

Go2Group

Products & Services | Solutions | Resources | Contact Us



Jumpstart Your **DevOps** Transformation With Us

#DevOpsMatters

cloudbees | Go2Group

EXPERT DEVOPS SERVICES FOR ORGANIZATIONS IN 70 COUNTRIES

The Workshop

- Intro and General Overview
- A Step By Step Approach To Installing And Setting Up The Frameworks
- Learn The Command Line Interfaces
- Review Of Our Example Configuration (Nostromo) And Important Files
- Pipeline Setup On Our Local Cluster (Minikube)
- Introduction To Jenkins X
- Pipeline And Ci/Cd Examples Using Jenkins X

Disclaimers

- I Am Not A Powerpoint Guru
- I Am Not Responsible For Those Who Fall Out Of Their Seats
- We Are Entering The Devops Danger Zone
- So Buckle Up

HOW DID WE GET HERE?

A Series Of Fortunate Events

- Agile - Kicked Into Gear Mid 2000
- Git - Started To Show Up 2006
- Github - Social Coding 2008
- The Jenkins Project - 2011
- Cloud
- Ansible
- Docker
- Kubernetes And Helm

WHAT WE LEFT BEHIND

A Series Of Not So Fortunate Events

- Waterfall
- CVS, SVN, Perforce, Sourcesafe
- Cruise Control, Anthill, And Others
- Bare-metal And Vmware
- Operations Controlled Environments

The Struggle Was Real

- Could Not Deliver On Time
- Long Development Cycles
- QA Pushed To The Wall At The End
- Server Gatekeepers
- Data Refreshes And Environment Issues
- Requirements Not Broken Down
- No Continuous Feedback On Delivered Code
- Lack Of Test Automation And Test Coverage
- Could Not Adjust For Change
- Lots Of Meetings To Determine What Happened And Why We Sucked

IT’S A BRAND NEW DAY

Tools

- Jenkins
- Kubernetes
- Helm
- Docker
- Github
- Google Cloud
- Minikube
- Jenkins X

Code

Build

Test

Docker Image

Staging Deployment

Manual Verification

Production Deployment

Kubernetes

The Name **kubernetes** originates From Greek, Meaning *helmsman* or *pilot*, And Is The Root Of *governor* and cybernetic. *k8s* is An Abbreviation Derived By Replacing The 8 Letters “Ubernete” With “8”.

What Is K8s ?

Kubernetes is An Open-source System For Automating Deployment,

Scaling, And Management Of Containerized Applications.

K8s Components

- kubectl and the K8s API
- Containers
- Pods
- Volumes
- Secrets
- Ingress

Helm

- The package manager for Kubernetes
- Helm helps you manage kubernetes applications
- Helm charts help you define, install, and upgrade even the most complex Kubernetes applications
- Charts are easy to create, version, share, and publish -- so start using Helm and stop the copy-and-paste madness

Docker

Docker Is The World's Leading Software Containerization Platform

Docker Containers And Tooling Make Building And Shipping Applications

Dramatically Easier And Faster. On Average Companies Using Docker Experience

A 7x Improvement In How Frequently They Are Able To Ship Software. More

Innovation Delivers Value To Their Customers Faster.

Today's Clouds

01

Minikube - Our local cluster for dev and play

02

Google Cloud Platform - An example using Jenkins X

LET’S GET STARTED

Why Local?

- Easy to develop and tear down
- We will learn more about K8s networking and other aspects such as Ingress
- You can try thing before you roll them out to your cloud

Workshop Repositories

https://github.com/mmaheu/modern-pipelines

https://github.com/mmaheu/nostromo

https://github.com/mmaheu/jenkins-k8s

Minikube Requirements

- Kubectl
- Virtual Box
- Docker

Install Kubectl

https://github.com/mmaheu/modern-pipelines/blob/master/kubernetes/kubectl.md

Install Minikube

If You Are On Mac And Have Homebrew Installed

brew cask install minikube

https://Github.Com/Mmaheu/Modern-pipelines/Blob/Master/Minikube/Setup.Md

Installing Helm

https://github.com/mmaheu/modern-pipelines/blob/master/helm/setup.md

Let’s Start Our K8s Cluster

01

minikube start / stop

02

minikube ip

03

minikube dashboard

Let’s Run Some Commands

- kubectl cluster-info
- kubectl get pods
- kubectl logs $podname
- kubectl get pods
- kubectl get services --namespace
- kubectl get secrets $podname -o yaml

CONGRATS ON YOUR K8S CLUSTER

Helm Intro

- Helm Charts
- Commands
  - helm search
  - helm install stable/owncloud
  - helm upgrade

Jenkins K8s Chart

Review The Chart On Github:

https://github.com/mmaheu/jenkins-k8s/blob/master/helm/jenkins-k8s/values.yaml

Jenkins Docker

- Build The Image Or Use The Image
- Review Dockerfile
- Install Jenkins K8s Using Helm
- https://cloud.docker.com

https://github.com/mmaheu/modern-pipelines/blob/master/jenkins/setup.md

Verify

01
Kubectl get pods

02
Kubectl get services

03
NodePort for local dev

04
ngrok http for easy URLs (local dev)

Let’s Take A Look At Our Jenkins Master

- HTTP://192.168.99.100:$PORT
- NGROK DNS

How To Connect To Our Pod

Kubectl Exec -it $Nodename — /Bin/Bash

Local Workarounds

- Docker.Io Auth Issue
  - Docker Login
- Running Tiller Locally On Our Jenkins Pod
  - We Have Helm Repo Set To Localhost
  - Helm Serve
  - We Are Not Using Ingress For Networking And Dns – Just Minikube Ip And Port
  - Ngroc

Nostromo

Https://Github.Com/Mmaheu/Nostromo

- Chart
- Jenkinsfile
- Build.Sh
- Dockerfile

Let’s Create The Pipeline

- This Time We Will Create The Pipeline Using Blue Ocean
- Setup The Webhook In Github
- Verify Helm Server Is Running – Local Only
- Initial Builds Will Fail As They Conflict

Manual Run

1. Run Narcissus development branch
2. Run Master branch

Lets Make Some Changes

1. Make Code Change On Narcissus
2. Commit And Push
3. Watch Build
4. Pull Request To Merge Narcissus Into Master
5. Watch Build

Review

- Minikube As Single Node K8s Cluster
- Command Line Tools Kubectl And Helm
- Nodeport vs Loadbalancer
- Github Integration
- Docker Io For Image Storage
- Jenkins And Blue Ocean
- Manual Creation Of Pipeline Based On Jenkinsfile

JENKINS X

LET'S DO THIS ON GOOGLE CLOUD

Jenkins X

- Automate Jenkins Ci/Cd Pipelines In The Cloud
- Rethinks How Developers Should Interact With CI/CD In The Cloud With A Focus On Making Development Teams Productive Through Automation, Tooling And Devops Best Practices.
- Is Open Source And We Invite You To Give Us Feedback And To Contribute To The Project.

The Big Deal

THE BIG DEAL ABOUT JENKINS X IS AS A DEVELOPER YOU CAN TYPE ONE COMMAND JX CREATE OR JX IMPORT AND GET YOUR SOURCE CODE, GIT REPOSITORY AND APPLICATION CREATED, AUTOMATICALLY BUILT AND DEPLOYED TO KUBERNETES ON EACH PULL REQUEST OR GIT PUSH WITH FULL CI/CD COMPLETE WITH ENVIRONMENTS AND PROMOTION VIA GITOPS!

What's In The Box

- JENKINS MASTER
- MONOCULAR
- NEXUS
- HELM
- KUBERNETES
- GITHUB WEBHOOK
- COMMAND LINE

Google Cloud

01
Create a project

02
Install Google Cloud SDK

03
./google-cloud-sdk/install.sh

04
Jenkins X will auth during the first k8s cluster create

Jenkins X Setup

HTTPS://GITHUB.COM/MMAHEU/MODERN-PIPELINES/BLOB/MASTER/JENKINSX/SETUP.MD

General Commands

- jx create cluster gke
- jx get environments
- jx get apps
- jx get urls
- jx rsh

Lets Create A K8s Cluster On Google Cloud

kubernetes

Google Cloud Platform

jx create cluster gke

CONTEXTS

Let’s Check It Out

- kubectl get pods and kubectl get nodes
- jx get applications
- jx get pipelines
- helm ls
- kubectl get namespaces
- jx console

https://console.cloud.google.com

https://console.cloud.google.com

Spring Boot

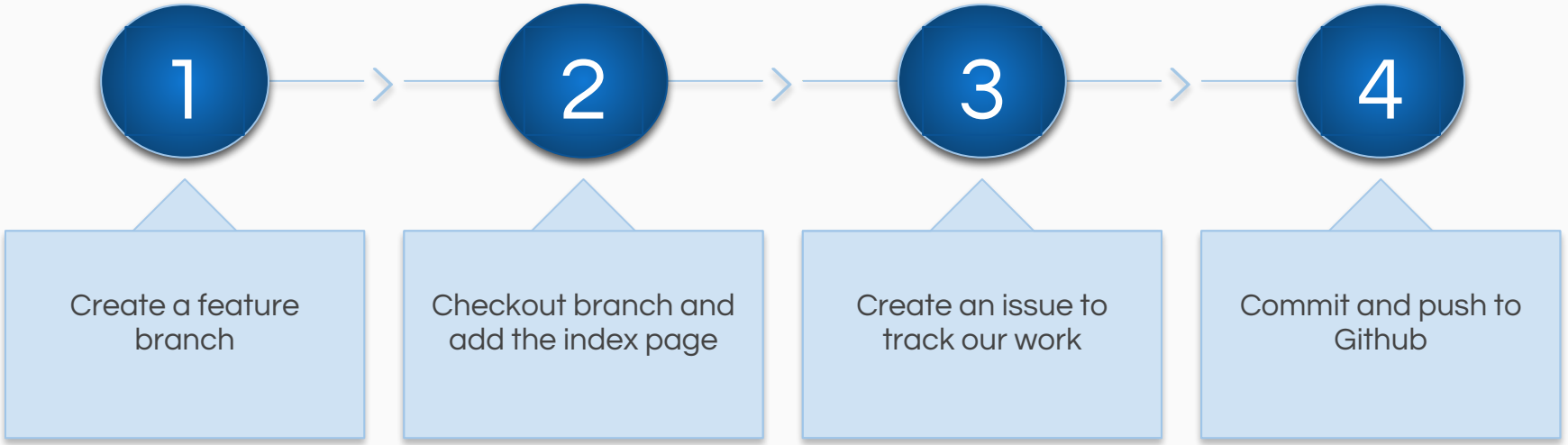
jx create spring

Or

jx create spring -d web -d actuator

Pipeline Review

Let’s Add An Index Page To Our App



Review

- Github Has An Issue And A Commit
- A Ci Build Got Kicked Off
- A Preview Environment Was Also Created
- Maven Artifacts Were Stored In Nexus
- Helm Chart Got Version And Stored - Monocular

MONITOR PIPELINE

CREATE A PULL REQUEST

Pull Request

- Review Pull Request
- Notice Issue Update
- Comment And Merge
- A CD Build Will Kick Off
- Our App Will Deploy To Staging

MONITOR PIPELINE

WHAT ABOUT PROD

Promote To Production

- JX PROMOTE --VERSION V0.0.X --ENV PRODUCTION --TIMEOUT 1H
- JX GET APPS
- REVIEW THE DEPLOYED VERSIONS
- COOL

ONE LAST THING

JX Import

1. jx import from project from Github
2. Dockerfile, Jenkinsfile, and Helm Chart get auto generated
3. Pipeline gets created
4. Initial Build

Import

1. jx import --url git@your_repo.git
2. Review The Pipeline and Builds
3. Review the generated configuration files in Github
4. Make changes
5. Monitor
6. Review

Q&A