GitOps, Jenkins X & the Future of CI/CD

Tracy Miranda, Director of Open Source Community | CloudBees

tmiranda@cloudbees.com | @tracymiranda

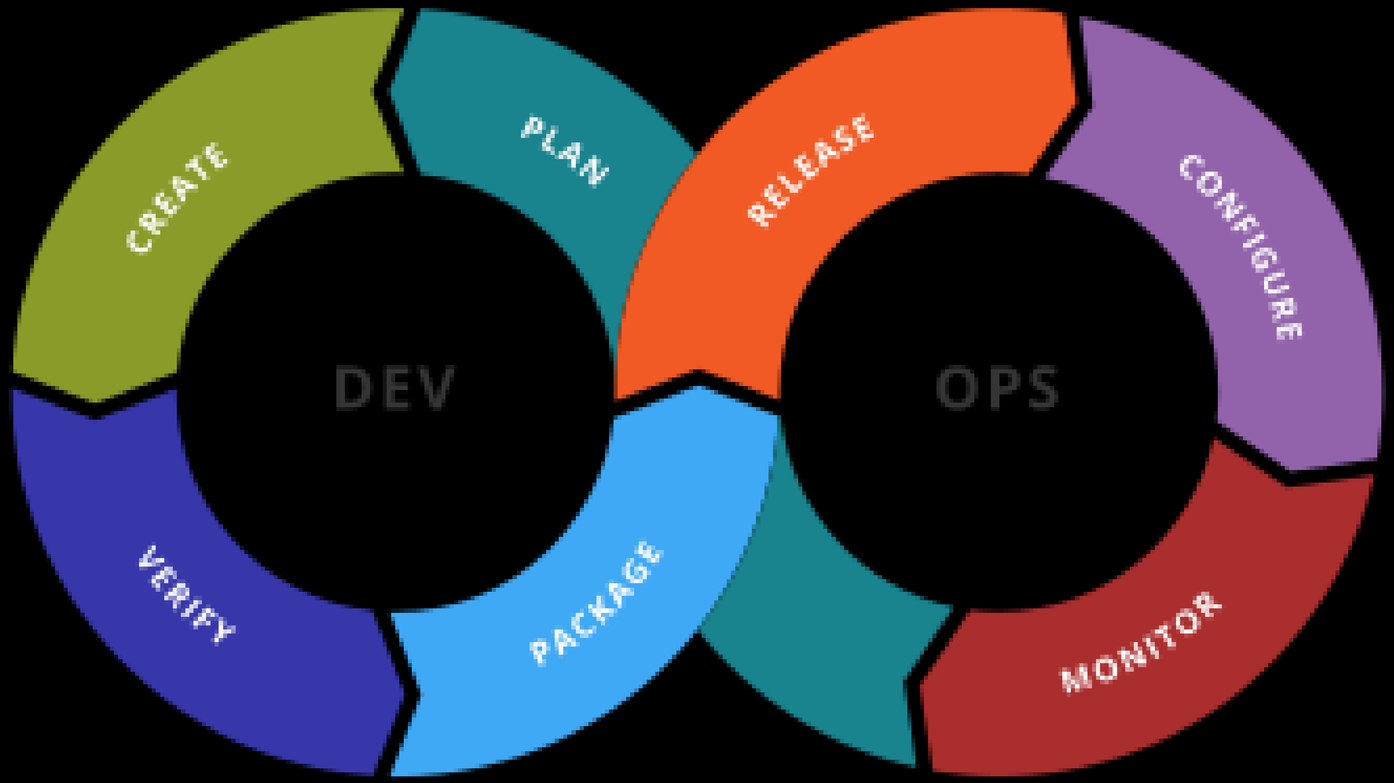
cloudbees

O'REILLY®
Velocity

So, DevOps…


Cindy Sridharan
@copyconstruct
Follow
Almost want to yell - Stop saying DevOps - it doesn't mean what you think it means.
2:33 AM - 13 Apr 2017
5 Retweets 10 Likes
1
5
10

DevOps is the new legacy



kubernetes

Cloud Native Technologies

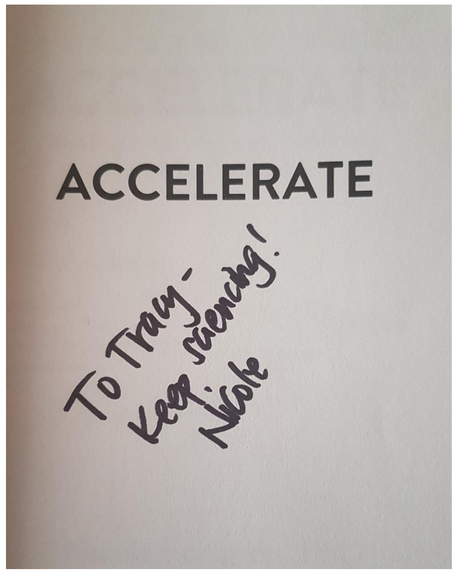
- On demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

DevOps → Cloud

GitOps → Cloud Native

What would CI/CD look like if we had unlimited, scalable resources?

DORA State of DevOps Reports



The Science of Lean Software and DevOps

“software delivery is an exercise in continuous improvement, and our research shows that year over year the best keep getting better, and those who fail to improve fall further and further behind.”

- Nicole Forsgren

Building and scaling high performance technology organisations

Throughput

Frequent deployments

Low lead time from commit to deploy

Stability

Fast mean time to recovery (MTTR)

Low change failure rate

GitOps

Operation by pull request

Git as the single place where we operate

All changes are observable

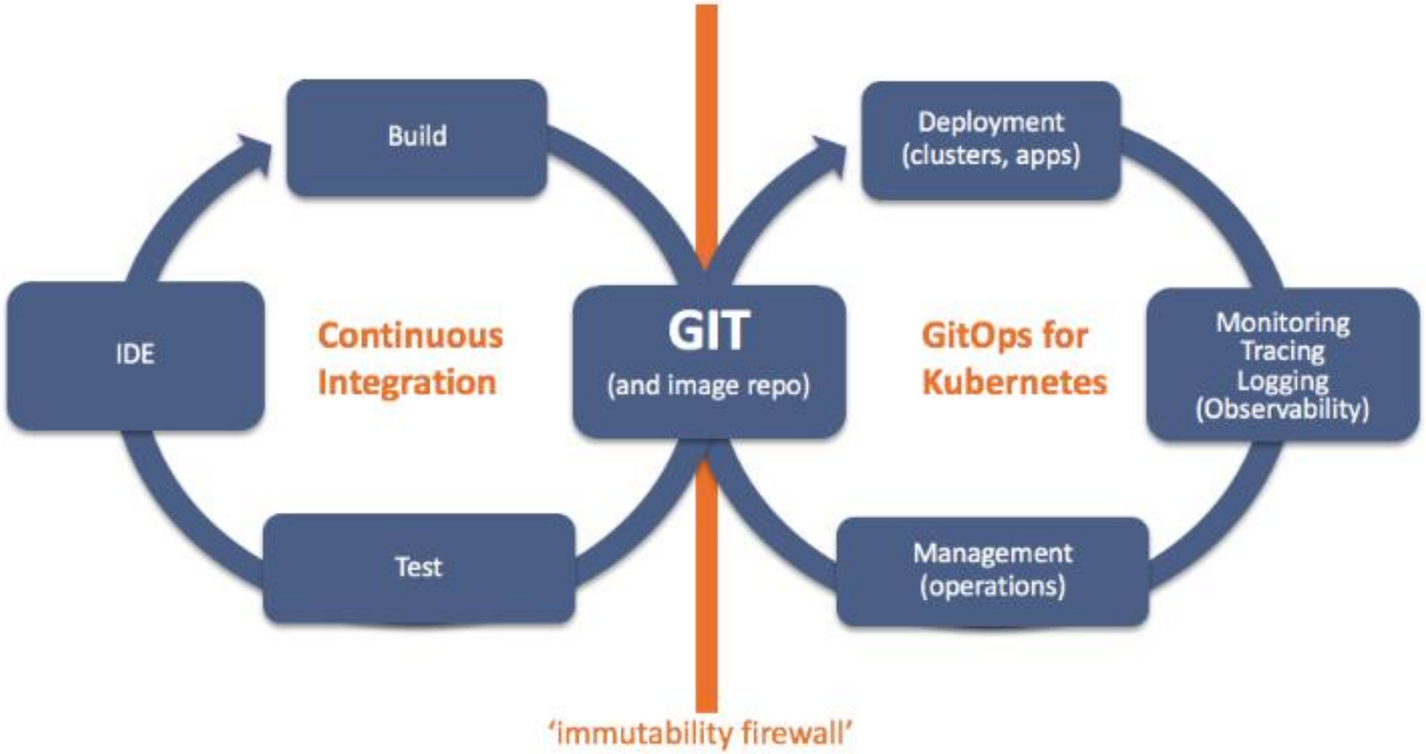
The GitOps Effect: Delivering Quality at Speed

2 → 150+

deployments per week

Via @fractallambda ref: https://twitter.com/tracymiranda/status/1045693846464729094

weaveworks https://www.weave.works/blog/what-is-gitops-really



Git as the single source of truth of a system's desired state

GitOps Diffs compare desired state with observed state (eg Kubediff, Terradiff, Canary..)

ALL intended operations are committed by pull request, for all environments

ALL diffs between GIT and observed state lead to (auto) convergence using tools like K8s

ALL changes are observable, verifiable and audited indisputably, with rollback & D/R

cloudbees

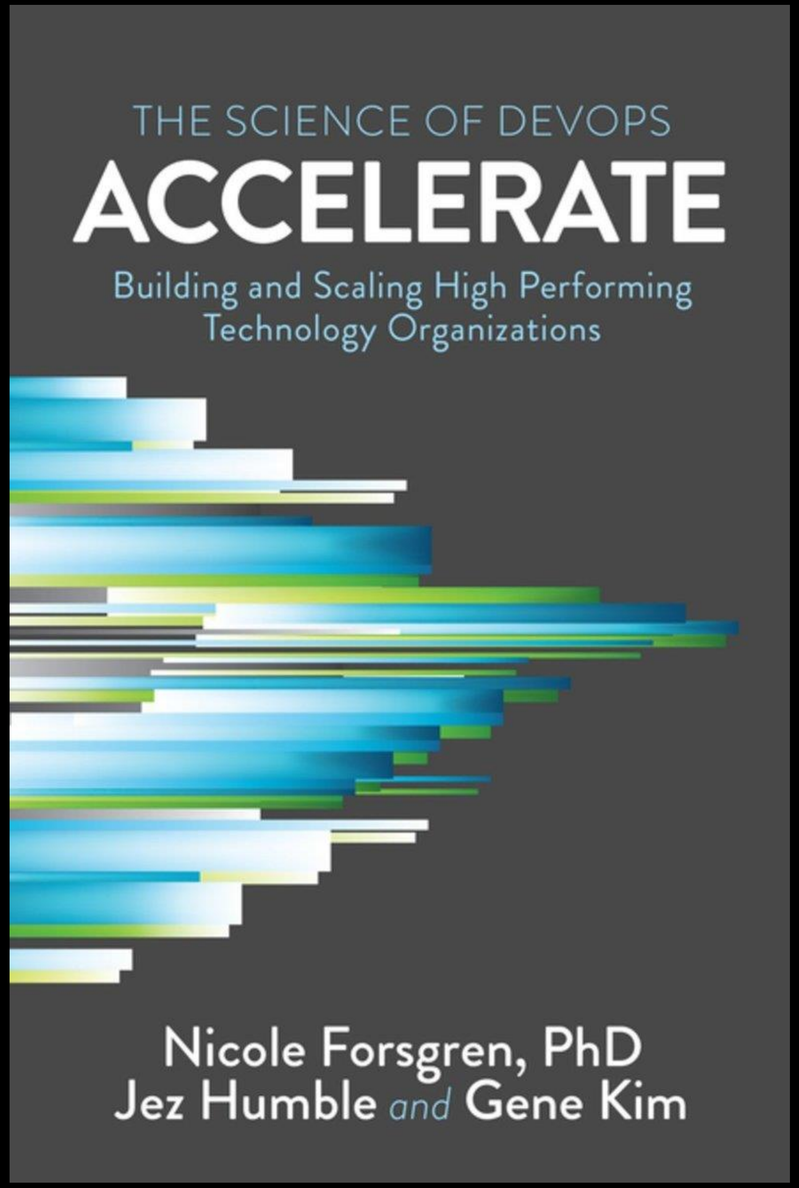

JENKINS X

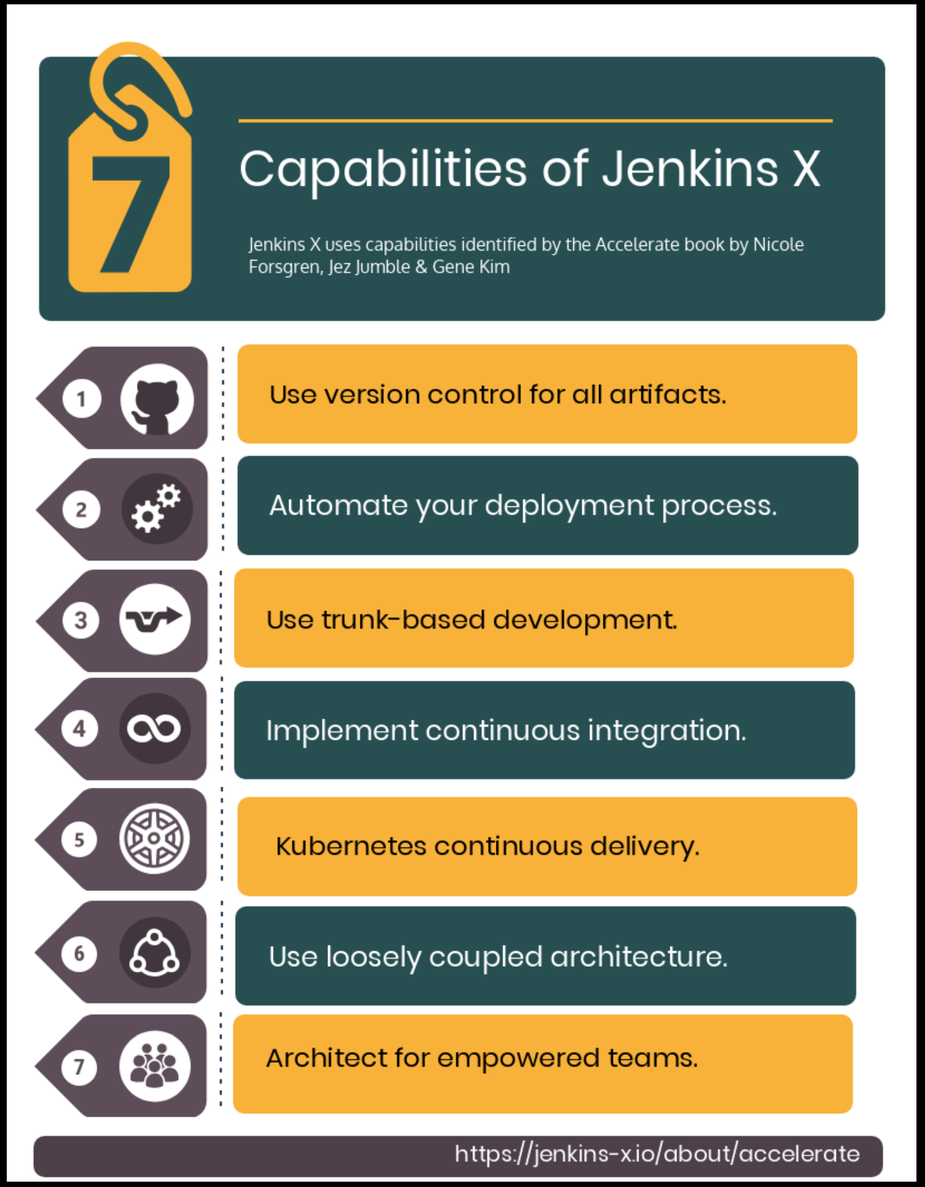
Best Practices for CI/CD for cloud native apps

1. Figure out the best practices of how to CD cloud native apps
   - Not just build and test, but review, promote, changelog, collaborate, etc.
2. Integrate best of breed software in this ecosystem to achieve it
3. Kubernetes as foundation & means to an end
4. Multicloud: Democratize it by building a high level, pleasant CLI
5. GitOps: deployments should be recorded and tracked in Git

Best Practices for CI/CD for cloud native apps

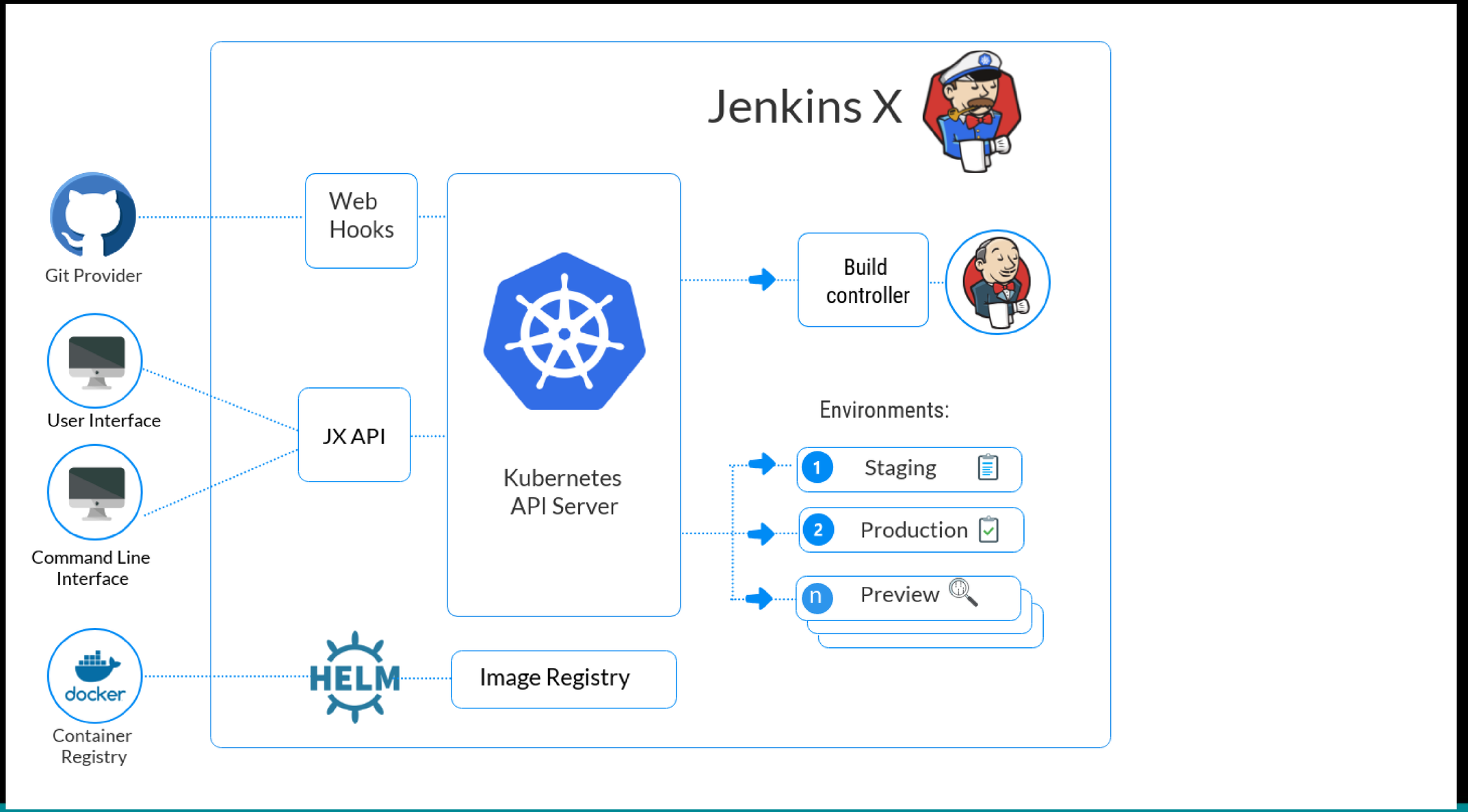
1. Figure out the best practices of how to CD cloud native apps
   - Not just build and test, but review, promote, changelog, collaborate, etc.
2. Integrate best of breed software in this ecosystem to achieve it
3. Kubernetes as foundation & means to an end
4. Multicloud: Democratize it by building a high level, pleasant CLI
5. GitOps: deployments should be recorded and tracked in Git


THE SCIENCE OF DEVOPS
ACCELERATE
Building and Scaling High Performing Technology Organizations
Nicole Forsgren, PhD
Jez Humble and Gene Kim


7
Capabilities of Jenkins X
Jenkins X uses capabilities identified by the Accelerate book by Nicole Forsgren, Jez Jumble & Gene Kim
1
Use version control for all artifacts.
2
Automate your deployment process.
3
Use trunk-based development.
4
Implement continuous integration.
5
Kubernetes continuous delivery.
6
Use loosely coupled architecture.
7
Architect for empowered teams.
https://jenkins-x.io/about/accelerate

Best Practices for CI/CD for cloud native apps

1. Figure out the best practices of how to CD cloud native apps
   - Not just build and test, but review, promote, changelog, collaborate, etc.
2. Integrate best of breed software in this ecosystem to achieve it
3. Kubernetes as foundation & means to an end
4. Multicloud: Democratize it by building a high level, pleasant CLI
5. GitOps: deployments should be recorded and tracked in Git


Jenkins X
Git Provider
Web Hooks
Kubernetes API Server
Build controller
User Interface
JX API
Environments:
1 Staging
2 Production
n Preview
Command Line Interface
docker
Container Registry
HELM
Image Registry

Best Practices for CI/CD for cloud native apps

1. Figure out the best practices of how to CD cloud native apps
   - Not just build and test, but review, promote, changelog, collaborate, etc.
2. Integrate best of breed software in this ecosystem to achieve it
3. Kubernetes as foundation & means to an end
4. Multicloud: Democratize it by building a high level, pleasant CLI
5. GitOps: deployments should be recorded and tracked in Git

Kubernetes Custom Resource Definitions (CRD)

Environment CRD

```
26  // EnvironmentSpec is the specification of an Environment
27  type EnvironmentSpec struct {
28          Label             string                `json:"label,omitempty" protobuf:"bytes,1,opt,name=label"`
29          Namespace         string                `json:"namespace,omitempty" protobuf:"bytes,2,opt,name=namespace"`
30          Cluster           string                `json:"cluster,omitempty" protobuf:"bytes,3,opt,name=cluster"`
31          PromotionStrategy PromotionStrategyType `json:"promotionStrategy,omitempty" protobuf:"bytes,4,opt,name=promotionStrategy"`
32          Source            EnvironmentRepository `json:"source,omitempty" protobuf:"bytes,5,opt,name=source"`
33          Order             int32                 `json:"order,omitempty" protobuf:"bytes,6,opt,name=order"`
34          Kind              EnvironmentKindType   `json:"kind,omitempty" protobuf:"bytes,7,opt,name=kind"`
35          PullRequestURL    string                `json:"pullRequestURL,omitempty" protobuf:"bytes,8,opt,name=pullRequestURL"`
36          TeamSettings      TeamSettings          `json:"teamSettings,omitempty" protobuf:"bytes,9,opt,name=teamSettings"`
37          PreviewGitSpec    PreviewGitSpec        `json:"previewGitInfo,omitempty" protobuf:"bytes,10,opt,name=previewGitInfo"`
38          WebHookEngine     WebHookEngineType     `json:"webHookEngine,omitempty" protobuf:"bytes,11,opt,name=webHookEngine"`
39  }
```

Kubernetes Custom Resource Definitions (CRD)

Pipeline Activity CRD

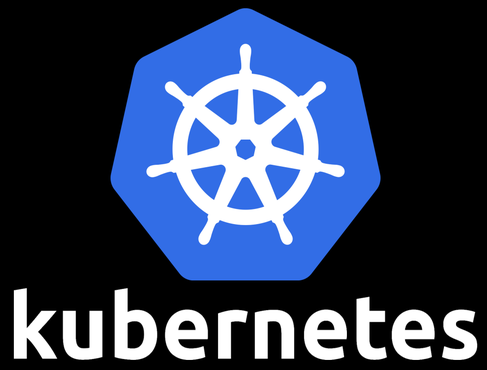
```
26  // PipelineActivitySpec is the specification of the pipeline activity
27  type PipelineActivitySpec struct {
28        Pipeline           string                `json:"pipeline,omitempty" protobuf:"bytes,1,opt,name=pipeline"`
29        Build              string                `json:"build,omitempty" protobuf:"bytes,2,opt,name=build"`
30        Version            string                `json:"version,omitempty" protobuf:"bytes,3,opt,name=version"`
31        Status             ActivityStatusType    `json:"status,omitempty" protobuf:"bytes,4,opt,name=status"`
32        StartedTimestamp   *metav1.Time          `json:"startedTimestamp,omitempty" protobuf:"bytes,5,opt,name=startedTimestamp"`
33        CompletedTimestamp *metav1.Time          `json:"completedTimestamp,omitempty" protobuf:"bytes,6,opt,name=completedTimestamp"`
34        Steps              []PipelineActivityStep `json:"steps,omitempty" protobuf:"bytes,7,opt,name=steps"`
35        BuildURL           string                `json:"buildUrl,omitempty" protobuf:"bytes,8,opt,name=buildUrl"`
36        BuildLogsURL       string                `json:"buildLogsUrl,omitempty" protobuf:"bytes,9,opt,name=buildLogsUrl"`
37        GitURL             string                `json:"gitUrl,omitempty" protobuf:"bytes,10,opt,name=gitUrl"`
38        GitRepository      string                `json:"gitRepository,omitempty" protobuf:"bytes,10,opt,name=gitRepository"`
39        GitOwner           string                `json:"gitOwner,omitempty" protobuf:"bytes,10,opt,name=gitOwner"`
40        ReleaseNotesURL    string                `json:"releaseNotesURL,omitempty" protobuf:"bytes,11,opt,name=releaseNotesURL"`
41        LastCommitSHA      string                `json:"lastCommitSHA,omitempty" protobuf:"bytes,12,opt,name=lastCommitSHA"`
42        LastCommitMessage  string                `json:"lastCommitMessage,omitempty" protobuf:"bytes,13,opt,name=lastCommitMessage"`
43        LastCommitURL      string                `json:"lastCommitURL,omitempty" protobuf:"bytes,14,opt,name=lastCommitURL"`
44        Workflow           string                `json:"workflow,omitempty" protobuf:"bytes,15,opt,name=workflow"`
45        WorkflowStatus     ActivityStatusType    `json:"workflowStatus,omitempty" protobuf:"bytes,16,opt,name=workflowStatus"`
46        WorkflowMessage    string                `json:"workflowMessage,omitempty" protobuf:"bytes,17,opt,name=workflowMessage"`
47        PostExtensions     []ExtensionExecution   `json:"postExtensions,omitempty" protobuf: "bytes,18,opt,name=postExtensions"`
48        Attachments        []Attachment          `json:"attachments,omitempty" protobuf: "bytes,19,opt,name=attachments"`
49        Facts              []Fact                `json:"facts,omitempty" protobuf: "bytes,20,opt,name=facts"`
50  }
```

Best Practices for CI/CD for cloud native apps

1. Figure out the best practices of how to CD cloud native apps
   - Not just build and test, but review, promote, changelog, collaborate, etc.
2. Integrate best of breed software in this ecosystem to achieve it
3. Kubernetes as foundation & means to an end
4. Multicloud: Democratize it by building a high level, pleasant CLI
5. GitOps: deployments should be recorded and tracked in Git

“Friends don’t let friends install Kubernetes”

- James Strachan, Jenkins X



You can have fun
tinkering

Or you can just get

productive

MultiCloud Support

jx create cluster minikube

```
* aks (Azure Container Service - https://docs.microsoft.com/en-us/azure/aks)
* aws (Amazon Web Services via kops - https://github.com/aws-samples/aws-workshop-for-kubernetes/blob/master/readme
* eks (Amazon Web Services Elastic Container Service for Kubernetes - https://docs.aws.amazon.com/eks/latest/usergu
* gke (Google Container Engine - https://cloud.google.com/kubernetes-engine)
* oke (Oracle Cloud Infrastructure Container Engine for Kubernetes - https://docs.cloud.oracle.com/iaas/Content/Con
# icp (IBM Cloud Private) - https://www.ibm.com/cloud/private
* iks (IBM Cloud Kubernetes Service - https://console.bluemix.net/docs/containers)
* oke (Oracle Cloud Infrastructure Container Engine for Kubernetes - https://docs.cloud.oracle.com/iaas/Content/Con
* kubernetes for custom installations of Kubernetes
* minikube (single-node Kubernetes cluster inside a VM on your laptop)
* minishift (single-node OpenShift cluster inside a VM on your laptop)
* openshift for installing on 3.9.x or later clusters of OpenShift
```

MultiCloud Create Cluster

Install kubectl, Helm, Jenkins X, etc

Install cloud provider CLI

- az, gcloud, kops, eksctl, oci

Create Kubernetes cluster

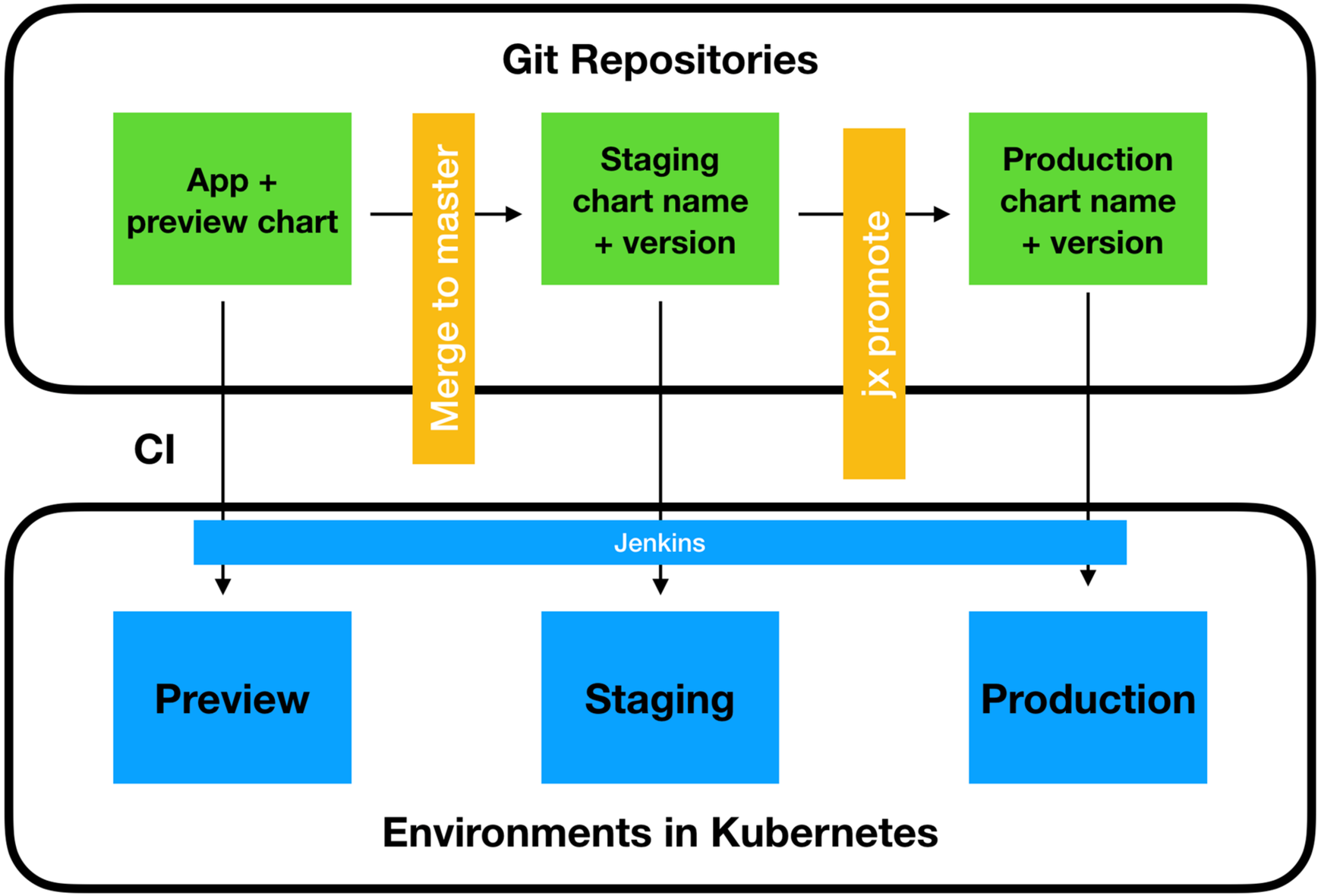
- RBAC,default storage class, enable docker registries, etc

Configure

- Create default namespace
- Setup the ingress controller
- Configure git source repository
- Create admin secrets

Best Practices for CI/CD for cloud native apps

1. Figure out the best practices of how to CD cloud native apps
   - Not just build and test, but review, promote, changelog, collaborate, etc.
2. Integrate best of breed software in this ecosystem to achieve it
3. Kubernetes is a means to an end
4. Multicloud: Democratize it by building a high level, pleasant CLI
5. GitOps: deployments should be recorded and tracked in Git


Git Repositories
App + preview chart
Merge to master
Staging chart name + version
jx promote
Production chart name + version
CI
Jenkins
Preview
Staging
Production
Environments in Kubernetes

Best Practices for CI/CD for cloud native apps

1. Figure out the best practices of how to CD cloud native apps
   - Not just build and test, but review, promote, changelog, collaborate, etc.
2. Integrate best of breed software in this ecosystem to achieve it
3. Kubernetes is a means to an end
4. Multicloud: Democratize it by building a high level, pleasant CLI
5. GitOps: deployments should be recorded and tracked in Git

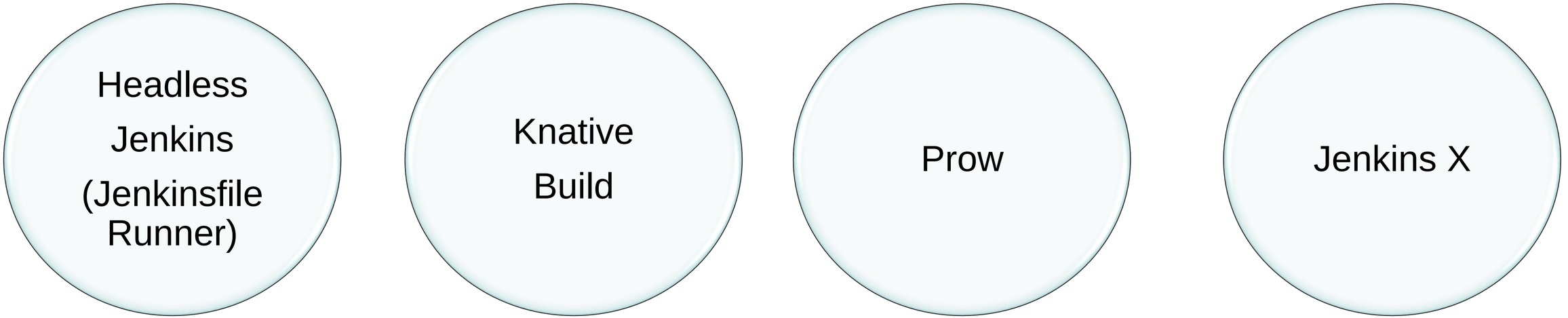
Future CI/CD

“Towards Progressive Delivery” James Governor, Redmonk

- ‘Blast Radius’
- Feature flags
- Canary
- ‘Built for failure’

Serverless Jenkins with Jenkins X

“By the time you merge it’s too late” – shifting left on CI/CD for pull requests



https://medium.com/@jdrawlings/serverless-jenkins-with-jenkins-x-9134cbfe6870

Wrap-up

Wrap Up

- Cloud native & Kubernetes are a major industry shift
- GitOps has essential techniques and practices to help you go faster without costing you stability.
- Jenkins X focuses on best practices for CI/CD for Kubernetes
- CI/CD is evolving forward quickly: serverless, progressive delivery
- We’re all on this journey, let’s drive things forward together