

Jenkins X

Continuous Integration, Automated Testing & Continuous Delivery for Kubernetes



Jenkins is al vele jaren in staat vrijwel alles te doen op het gebied van CI/CD. De uitdaging is altijd wel geweest om uit te zoeken hoe je de juiste plug-ins, configuratie en code kunt verzamelen en combineren om samen te werken in je pipeline.

In de afgelopen jaren zagen we grote verschuivingen binnen de software-industrie. We zien de opkomst van immutable containers voor het distribueren van de applicaties. Kubernetes is hard op weg om de defacto standaard te worden op het gebied van het installeren, upgraden, onderhouden en managen van containers op elke schaalgrootte zowel op de public als private cloud. Een steeds grotere acceptatie en implementatie van microservices en cloud-native applicaties zorgt voor een grote toename in het aantal componenten die allen CI/CD behoeven en een hogere releasefrequentie veroorzaken.

Deze verschuivingen zorgde ervoor dat de tooling welke we gebruikte voor CI/CD mee moesten veranderen. Deze tooling is grofweg in te delen in twee groepen:

- Continuous Integration tooling welke ook Kubernetes kunnen ondersteunen;
- Continuous Deployment tools voor deployment naar Kubernetes.

De CI tooling zijn de tools welke vooral zijn gericht op het unit testen en het integreren van wijzigingen in je code. Hieronder vallen onder andere: Jenkins, Travis, Circle CI en GitLab. Deze CI tools kunnen meestal ook geschikt worden gemaakt voor het doen van Continuous Deployment maar zijn van oudsher hier niet op gericht. Bij CD kunnen we denken aan onder andere: Shippable, Weave Cloud, CodeFresh en Harness. Deze tools besteden het CI gedeelte uit aan andere tooling en richting zich specifiek op deployments.

In maart 2018 verscheen Jenkins X op het toneel als dé Jenkins-tegenhanger voor het

geautomatiseerd uitvoeren van zowel Continuous Integration als Continuous Deployment op Kubernetes-omgevingen. Sindsdien hoeven ontwikkelaars en teams geen tijd meer te besteden aan het packagen van docker images, het schrijven van Kubernetes YAML-files om hun applicaties op Kubernetes te draaien of zelfs maar te leren hoe je declaratieve pipelines-as-code Jenkinsfiles gebruikt voor het implementeren van CI/CD.

Tegelijkertijd verbergt Jenkins X niets. Als je de Dockerfile, Jenkinsfile of Helm-diagrams voor je applicaties of omgevingen wil aanpassen, dan zijn deze allemaal beschikbaar in Git.

Als ontwikkelaar kun je met een commando je broncode, git repositories en applicaties aanmaken, automatisch gebouwd en gedeployed op Kubernetes bij elke pull request of git push met volledige CI/CD compleet volgens GitOps (zie kader).



Arjan Gelderblom is een Security Software Engineer bij First8. Hij specialiseert zich in alle aspecten van het beveiligen van applicaties, van ontwerp tot end-of-life.

GITOPS

GitOps is een manier om Kubernetes-clusterbeheer en applicatie delivery te doen. Het werkt door Git te gebruiken als een enkele bron van waarheid voor declaratieve infrastructuur en applicaties. Met Git in het midden van de delivery pipeline kunnen ontwikkelaars pull requests doen om deployments van applicaties en operations taken voor en naar Kubernetes te versimpelen en te versnellen. Het betekent dat elke Environment een git-repository heeft om alle omgevingsspecifieke configuratie samen met een lijst van alle applicaties en hun versie en configuratie op te slaan. Hierdoor leidt een promotie van een applicatie naar een omgeving tot een pull request voor de configuratiewijzigingen die een pipeline triggert voor test, code review en acceptatie. Zodra de pull request is gemerged wordt de Helm Chart metadata toegepast door de release-pipeline.[1]

Wat is Jenkins X?

Jenkins X biedt pipeline-automatisering, ingebouwde GitOps en preview-omgevingen om teams te helpen samenwerken en hun software delivery op elke schaal te kunnen versnellen. Om dit te kunnen doen gebruikt het verschillende open source projecten.



Jenkins

Jenkins is de belangrijkste CI/CD engine binnen Jenkins X. Jenkins X positioneert zich als een subproject van het vertrouwde Jenkins project maar met een focus op het automatiseren van CI/CD voor de cloud. De bekende CI/CD functionaliteiten worden dus nog steeds vanuit Jenkins gebruikt.



Helm

Helm helpt bij het beheren van applicaties binnen Kubernetes. Helm is een applicatie package manager die op Kubernetes draait. Helm maakt het mogelijk om een applicatiestructuur te beschrijven via zogeheten Helm Charts en deze te beheren. Voor de opslag van de Helm Charts wordt gebruik gemaakt van Chartmuseum, een open source repository voor Helm Charts. Voor het zoeken van Helm Charts is Monocular beschikbaar, een webapplicatie die in staat is om in meerdere Helm-repositories te zoeken.

Draft

Jenkins X gebruikt tooling van Draft zodat taal- en frameworkspecifieke "build packs" kunnen worden onderhouden die de stan-



daard Dockerfile, Jenkinsfile en Helm Charts bevatten die nodig zijn om verschillende soorten applicaties te bouwen, te testen, vrij te geven en te implementeren.



Artifacts

Voor het opslaan van artifact gebruikt Jenkins X Nexus, de bij de meeste mensen wel bekende (open source) artifact repository. Daarnaast wordt Docker Registry geïnstalleerd voor het opslaan van de docker images.

Jenkins X belangrijkste features

Jenkins brengt enkele nieuwe features in de pipeline die het leven van een ontwikkelaar een stuk makkelijker maken en volledige cloud-native CI/CD mogelijk maken.

Geautomatiseerde CI/CD pipelines

Of je nu een nieuw Spring Boot project, een quickstart of bestaande code importeert, Jenkins X creëert automatisch een pipeline die de best practice CI/CD standaarden implementeert. Deze pipeline bestaat uit een Jenkinsfile, een Dockerfile en Helm Charts. Er wordt een git-repository aangemaakt voor de code, met de juiste webhooks. Na dit alles wordt de eerste release pipeline gestart voor de promotie van de applicatie naar de Staging omgeving.

KUBERNETES IS HARD OP WEG OM DE DEFACTO STANDAARD TE WORDEN OP HET GEBIED VAN HET INSTALLEREN, UPGRADEN, ONDERHOUDEN EN MANAGEN VAN CONTAINERS OP ELKE SCHAAL-GROOTTE

BOUWSTENEN

- Jenkins X is opgebouwd uit de volgende bouwstenen:
- Jenkins 2.0 - De oude vertrouwde Jenkins - jenkins.io
- Helm - Package Manager - helm.sh
- Draft - Development environment build tool - draft.sh
- Monocular - UI voor Helm repositories - github.com/helm/monocular
- Chartmuseum - Helm repository - chartmuseum.com
- Nexus - Artifact repository - sonatype.com/nexus-repository-oss

Bij elke pull request wordt een CI pipeline gestart die je unit tests draait en de applicatie bouwt om er voor te zorgen dat de master branch altijd in een status is waarin deze klaar is voor een release. Gelijktijdig wordt er een Preview Environment opgezet waarin de applicatie wordt gedeployed.

Wanneer de pull request gemerged wordt naar de master branch, dan wordt de release pipeline gestart om een nieuwe release te bouwen. Dit houdt in dat een new Semantic Version Number wordt gecreëerd en de source code (bijvoorbeeld pom.xml) wordt aangepast met dit nieuwe versienummer. Artifacts (bijvoorbeeld jar-files), docker images en helm charts worden gepubliceerd. En deze worden gepromoot naar de juiste Environment.

Promotie via GitOps

In Jenkins X krijgt elk team zijn eigen omgevingen genaamd Environments. Standaard zijn de omgevingen Staging en Production. Maar teams kunnen zoveel omgevingen, met eigen gekozen namen, maken als ze zelf maar willen.

Een Environment is een plaats waarin de applicaties gedeployed worden en elke Environment is gekoppeld aan een namespace in Kubernetes. Hierdoor is elke omgeving geïsoleerd en kunnen ze onafhankelijk worden beheerd.

Jenkins X gebruikt GitOps voor het managen van omgevingen en het doorvoeren van promoties tussen omgevingen. Omgevingen kunnen worden geconfigureerd om ofwel automatisch te promoten als onderdeel van een release-pipeline of ze kunnen worden geconfigureerd voor handmatige promotie. Momenteel zijn de standaardinstellingen dat de Staging-omgeving automatisch gepromoot wordt. De Production-omgeving is geconfigureerd om handmatig promotie te gebruiken, zodat kan worden gekozen wanneer de promotie naar productie plaatsvindt. Het is echter mogelijk om de configuratie op elk moment aan te passen.

Preview omgevingen

Jenkins X laat je preview-omgevingen maken voor pull requests. Standaard gebeurt dit automatisch wanneer een pull request wordt aangemaakt, maar dit kan ook handmatig. Wanneer een preview-omgeving wordt gemaakt wordt er een Environment van het type

Preview gemaakt met daaraan gekoppeld een Kubernetes namespace. Daarna wordt de pull request gebouwd als een preview docker image en chart en gedeployed in de nieuwe omgeving. En er wordt een comment geplaatst bij de pull request om te laten weten dat de Preview omgeving klaar staat. Zodra een update op de pull request wordt gedaan (een nieuwe commit) zal de preview omgeving worden geupdate met deze wijziging. Voor het verwijderen van gesloten en gemergde pull requests loopt er elke drie uur een cron job.

Door middel van deze preview omgevingen kun je je wijzigingen testen alsof ze al gemerged zijn. Hiermee maak je de kans op "But it works on my machine" een stuk kleiner omdat je zelfs al voor de merge een mogelijkheid hebt om je wijzigingen te testen.

Switch

De belangrijkste vraag is nu natuurlijk: Moeten we nu switchen? Zoals altijd het antwoord is: Dat hangt af van een heleboel. Allereerst moet je natuurlijk Kubernetes gebruiken om te kunnen switchen naar Jenkins X.

Heb je het geluk om net op dit moment een nieuw project te starten dan is mijn antwoord volmondig JA. Jenkins X levert een hoop features out-of-the-box die een verrijking zijn voor de CI/CD pipeline.

Voor degene die van Jenkins afkomen en gebruik maken van Jenkinsfiles is de overstap eenvoudiger te realiseren doordat Jenkinsfiles ook ondersteund worden door Jenkins X. Dit kan dan als tussenstap worden gebruikt naar het volledige model van Jenkins X.

Voor alle andere pakketten zal er afgewogen moeten worden of de features welke Jenkins X tegen de kosten van de migratie opwegen.

Zelf experimenteren?

Wil je zelf eens experimenteren met Jenkins X? Op de website van Jenkins X (<https://jenkins-x.io>) staan tutorials. Onder andere één om zelf Jenkins X te installeren op bijvoorbeeld Minikube (een lokaal draaiend kubernetes cluster in een Virtuele Machine). ■

**JENKINS X
LEVERT EEN
HOOP FEATURES
OUT-OF-THE-
BOX DIE EEN
VERRIJKING ZIJN
VOOR DE CI/CD
PIPELINE**

BRONNEN

[1] <https://www.weave.works/technologies/gitops/>