

plusYOU
personal. career. management.

meets



Spring Boot empowered K8s & Exploring Tekton

Lennart Gamradt
Christopher Karow

Wer sind wir



Lennart Gamradt

✉ lennart.gamradt@legital.org

✎ [xing.to/lgamradt](https://www.xing.to/lgamradt)

in [linkedin.com/in/lgamradt](https://www.linkedin.com/in/lgamradt)

Christopher Karow

✉ christopher.karow@legital.org

✎ [xing.to/ckarow](https://www.xing.to/ckarow)

in [linkedin.com/in/ckarow](https://www.linkedin.com/in/ckarow)



Spring Boot empowered K8s

Getting started with Spring Boot on Kubernetes

- Spring Boot Docker Image
- Application Health Check & K8s Probes

Dockerfile für Spring Boot

- Dockerfile definiert Docker Image Aufbau

```
FROM openjdk:8-jdk-alpine
ARG JAR_FILE=target/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

- Nachteile dieser Variante:
 - Keine User Rechte
 - Applikationen im Docker Container laufen als root User
 - JAR verschlechtert Performance durch verpackte Sourcen

Optimiertes Dockerfile für Spring Boot

- Kompilierte Klassen direkt in Docker Image mit User

```
FROM openjdk:8-jdk-alpine
RUN addgroup -S spring && adduser -S spring -G spring
USER spring:spring
ARG DEPENDENCY=target/dependency
COPY ${DEPENDENCY}/BOOT-INF/lib /app/lib
COPY ${DEPENDENCY}/META-INF /app/META-INF
COPY ${DEPENDENCY}/BOOT-INF/classes /app
ENTRYPOINT ["java", "-cp", "app:app/lib/*", "hello.Application"]
```

spring-boot:build-image

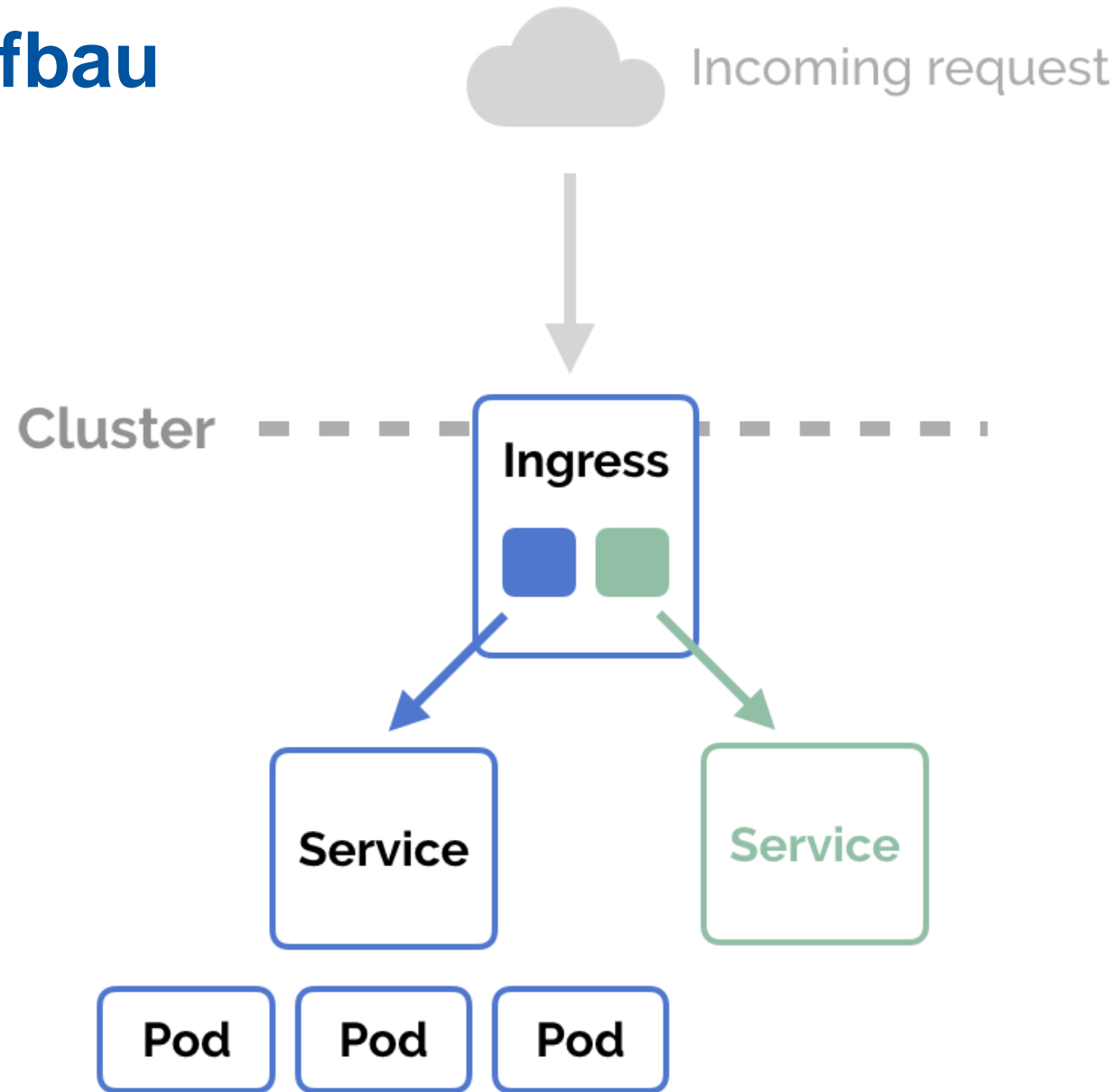
- Mit Spring Boot 2.3 wurde das direkte Bauen eines Docker Images für Spring Boot Applikationen eingeführt
- Bei Nutzung vom bereitgestellten Spring Boot Maven Plugin werden die Cloud Native Buildpacks herangezogen und ein OCI Image erstellt

```
./mvnw spring-boot:build-image  
-Dspring-boot.build-image.imageName=legital/k8s-probes-demo
```

■ Gradle Befehl

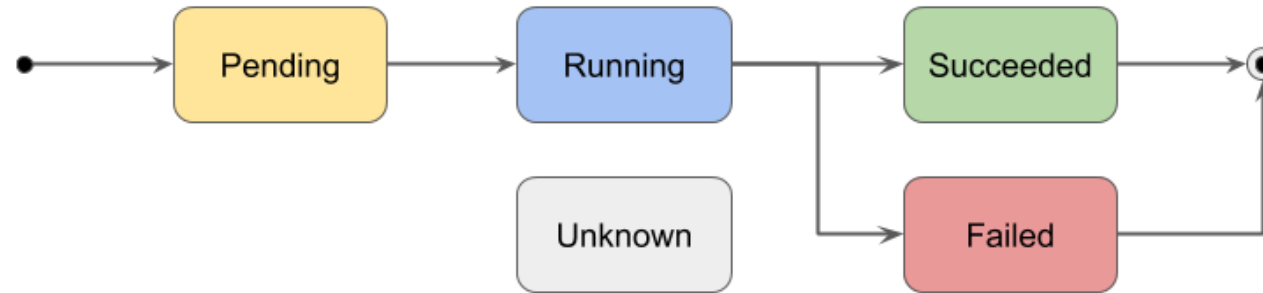
```
./gradlew bootBuildImage
```

Kubernetes Aufbau



Zustände während Laufzeit

Pod



Container

Waiting

Running

Terminated

`kubectl describe pod <name-of-pod>`

Spring Boot Application im Docker Image



Kubernetes Container Probes

- Probe ist eine Diagnose, die periodisch vom Kubelet auf Container durchgeführt wird
- Arten von Handlern
 - **ExecAction**: Befehl im Container mit Erwartung Exit mit Status 0 ausführen
 - **TCPSocketAction**: TCP Check auf IP-Adresse vom Pod mit Erwartung, dass Port offen ist
 - **HTTPGetAction**: HTTP Get Request auf IP-Adresse vom Pod mit Erwartung, dass HTTP Status ≥ 200 und < 400

Arten von Probes

- Ergebnis der Probes: Success, Failure, Unknown
- Arten von Probes
 - **livenessProbe:** Signalisiert, ob App läuft. Bei Fehlschlag killt Kubelet den Container. Die „Restart Policy“ greift.
 - **readinessProbe:** Signalisiert, ob App antwortet. Bei Fehlschlag entfernt Endpunkt-Controller IP-Adresse des Pods von Endpunkten aller Dienste, die mit Pod übereinstimmen.
 - **startupProbe:** Signalisiert, ob App im Container läuft. Alle anderen Probes werden deaktiviert, wenn Startup Probe bereitgestellt ist. Wenn Probe fehlschlägt, killt Kubelet den Container. Die „Restart Policy“ greift. (K8s v. 1.20)

Spring Boot Actuator

- Spring Boot Actuator wird als MVN-Dependency hinzugefügt

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-actuator</artifactId>  
</dependency>
```

- Seit Spring Boot 2.3: Liveness & Readiness Probe Unterstützung
Manuelle Aktivierung in Application Configuration:

```
management:  
  health:  
    probes:  
      enabled: true  
endpoint:  
  health:  
    show-details: always
```

Actuator – Liveness & Readiness

- Wenn Anwendung auf K8s läuft, sind die Actuator Endpunkte automatisch aktiviert
- ApplicationAvailability Interface für “Liveness” und “Readiness”
- LivenessStateHealthIndicator und ReadinessStateHealthIndicator als dedizierte Health Indikatoren

/actuator/health/liveness HTTP-Endpunkt

```
livenessProbe:  
  httpGet:  
    path: /actuator/health/liveness  
    port: <actuator-port>  
  failureThreshold: ...  
  periodSeconds: ...
```

/actuator/health/readiness HTTP-Endpunkt

```
readinessProbe:  
  httpGet:  
    path: /actuator/health/readiness  
    port: <actuator-port>  
  failureThreshold: ...  
  periodSeconds: ...
```

Fragen & Diskussion



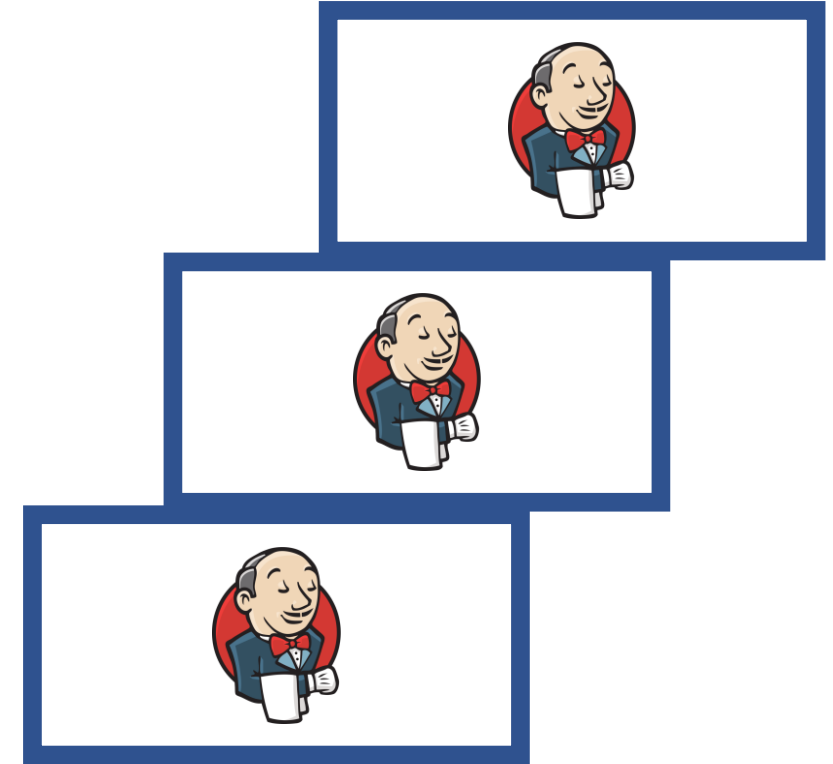
- Wie sind eure Erfahrungen mit readiness und liveness Probes?
- Welche Erfahrung habt ihr mit der in Kubernetes 1.20 eingeführten startup Probes gemacht?

Exploring Tekton

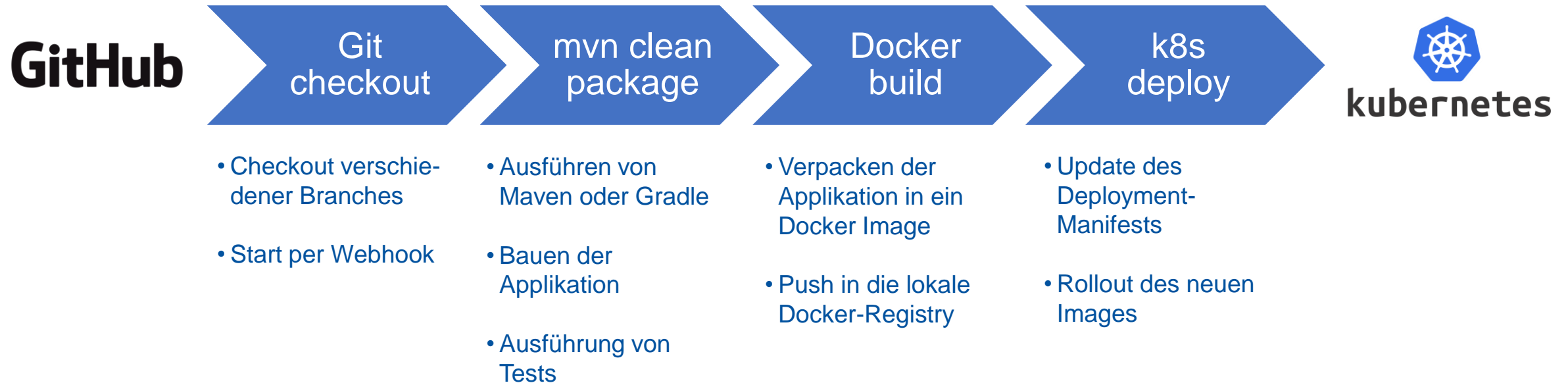
CI with Kubernetes

Meine Probleme mit Jenkins in Containern

- Speicher basiert komplett auf Dateien
- Verwendet überhaupt Speicher (nicht stateless)
- Verbraucht viele Ressourcen (RAM & CPU)
- Jeder baut seine eigenen Pipelines
- Kein Caching
- Weitere Punkte?



Die Pipeline



Tekton is a powerful and flexible open-source framework for creating CI/CD systems, allowing developers to build, test, and deploy across cloud providers and on-premise systems.

<https://tekton.dev>



Was ist Tekton?

Cloud Native	Pipeline as Code	Wiederverwendbar	Open Source
<ul style="list-style-type: none">• Läuft direkt im K8s Cluster• Definition durch Custom Resources• Basiert vollständig auf Containern• Kein zusätzliches CI/CD System notwendig	<ul style="list-style-type: none">• Definition als yaml oder json• Pipeline wird direkt mit dem Code verwaltet• Spielt hervorragend mit Tools wie z.B. helm & git zusammen• Unterstützt GitOps	<ul style="list-style-type: none">• Bausteine komplett parametrisiert• Erst bei der Ausführung werden umgebungs-spezifische Daten (z. B. Branch-Name, Stage etc.) übergeben.• Admins können standardisierte Pipelines bauen, die von Devs verwendet werden.	<ul style="list-style-type: none">• Aktiv entwickelt von einer großen Community• Bausteine können beliebig verändert und erweitert werden• Kompletter Quellcode in github

Mehr Infos hier:

- <https://www.cncf.io/>
- <https://landscape.cncf.io/>

Kubernetes Custom Resources Definitions (CRDs)

- Erstellung von eigenen Manifesttypen
- Erweiterungspunkt zur bestehenden K8s API
- Stark genutzt von K8s-Operators und Tekton

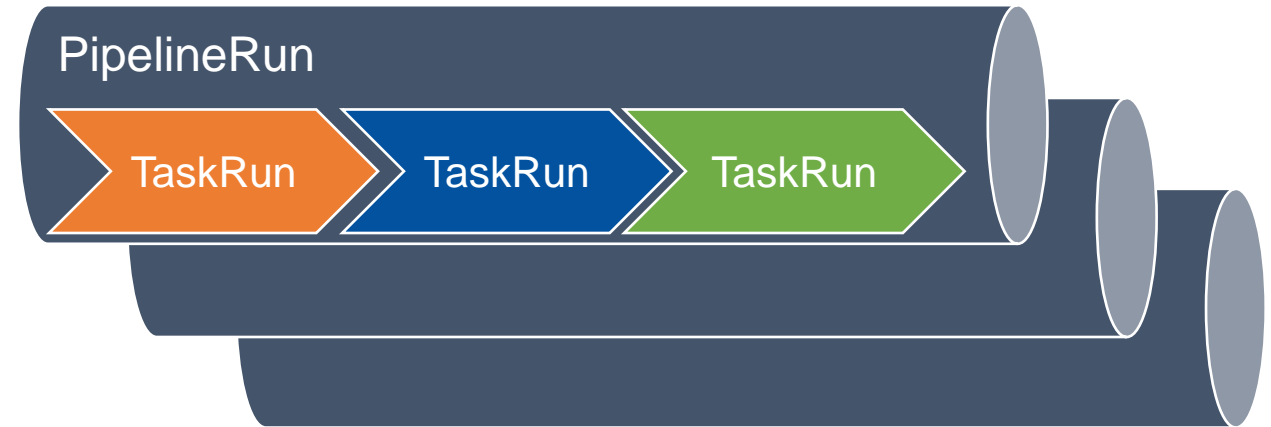
Mehr Infos hier:

- <https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/>
- <https://operatorhub.io/>

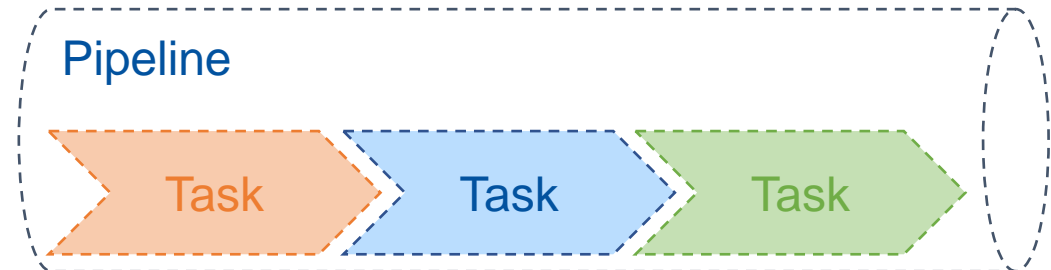
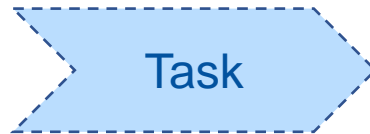
```
apiVersion: tekton.dev/v1beta1
kind: Pipeline
metadata:
  name: mvn-build
  namespace: playground
spec:
  params:
    - description: URL of the git repository
      name: gitUrl
      type: string
    - description: git revision to checkout
      name: gitRevision
      type: string
      default: develop
    - description: Name of the docker image
      name: imageName
      type: string
    - description: Context path to deployment
      name: deploymentTemplateContext
      type: string
  tasks:
    - name: checkout
      params:
        - name: url
          value: $(params.gitUrl)
```

Zentrale Bestandteile

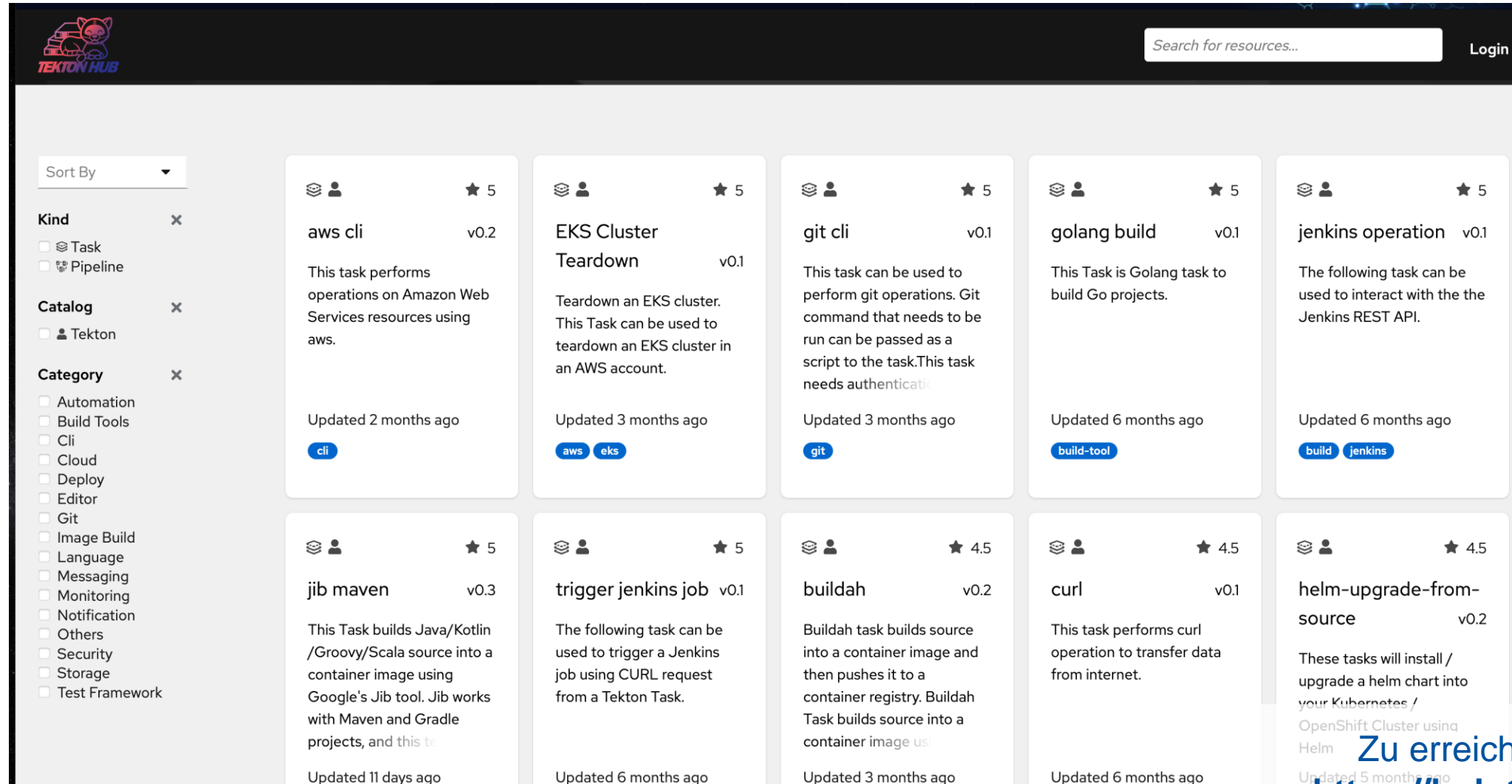
Ausführung



Definition



Tekton Hub



The screenshot shows the Tekton Hub interface. At the top, there's a search bar labeled "Search for resources..." and a "Login" button. On the left, there's a sidebar with filters: "Sort By", "Kind" (Task, Pipeline), "Catalog" (Tekton), and "Category" (Automation, Build Tools, Cli, Cloud, Deploy, Editor, Git, Image Build, Language, Messaging, Monitoring, Notification, Others, Security, Storage, Test Framework). The main area displays a grid of task cards. Each card shows the task name, version, a star rating, a brief description, and when it was last updated. Some cards also have tags indicating the tool or platform they use.

Task Name	Version	Rating	Description	Updated	Tags
aws cli	v0.2	5	This task performs operations on Amazon Web Services resources using aws.	2 months ago	cli
EKS Cluster Teardown	v0.1	5	Teardown an EKS cluster. This Task can be used to teardown an EKS cluster in an AWS account.	3 months ago	aws, eks
git cli	v0.1	5	This task can be used to perform git operations. Git command that needs to be run can be passed as a script to the task. This task needs authentication.	3 months ago	git
golang build	v0.1	5	This Task is Golang task to build Go projects.	6 months ago	build-tool
jenkins operation	v0.1	5	The following task can be used to interact with the the Jenkins REST API.	6 months ago	build, jenkins
jib maven	v0.3	5	This Task builds Java/Kotlin /Groovy/Scala source into a container image using Google's Jib tool. Jib works with Maven and Gradle projects, and this task...	11 days ago	
trigger jenkins job	v0.1	5	The following task can be used to trigger a Jenkins job using CURL request from a Tekton Task.	6 months ago	
buildah	v0.2	4.5	Buildah task builds source into a container image and then pushes it to a container registry. Buildah Task builds source into a container image using...	3 months ago	
curl	v0.1	4.5	This task performs curl operation to transfer data from internet.	6 months ago	
helm-upgrade-from-source	v0.2	4.5	These tasks will install / upgrade a helm chart into your Kubernetes / OpenShift Cluster using Helm	5 months ago	

Zu erreichen unter
<https://hub.tekton.dev/>

Installation & Benutzung

■ kubectl

```
kubectl apply -f https://storage.googleapis.com/tekton-releases/pipeline/latest/release.yaml
```

```
kubectl create -f k8s/tekton/pipeline-run.yaml
```

```
kubectl get pr
```

■ Tekton CLI

```
tkn pr list
```

```
tkn hub search git
```

```
tkn pr describe probes-run-m6q7l
```

Walkthrough & Demo

Verwandte Projekte & Alternativen



Fragen & Diskussion

- Welche CI/CD Tools nutzt ihr?
- Habt ihr Erfahrungen gemacht mit CI/CD in der Cloud?

Die Quellen und Links zum Talk findet ihr hier:



<https://github.com/legital-gmbh/k8s-probes-demo>