



KNIME Edge Installation Anleitung

KNIME AG, Zürich, Schweiz

Version 1.3 (letzte Aktualisierung am 2024-05-15)



Inhaltsverzeichnis

Einleitung
Komponenten von KNIME
 KNIME Kantenkontrolle
 KNIME Edge Operator
 KNIME Edge Inference
 Inference Deployment
 Kong API Gateway. MinIO.
KNIME Kantenarchitektur
 Verwendung von KNIME Bus
Installationsplanung . . .
 Installationsmethoden
 KNIME Business Hub
 Kubernetes
 Softwarevoraussetzungen
 Netzwerkvoraussetzungen
 Kapazitätsplanung & Erwägungen
KNIME konfigurieren Business Hub
 Das KNIME hochladen
 Konfigurieren Sie den KNIME Edge
 Eine Notiz über die Autoreferenz
Installation von KNIME Randdiagramm
 Auswahl eines Kubernetes
 Einloggen in die KNIME Edge
 Konfigurieren von Helm
 Hinzufügen eines Pod
 Anwendung eines KNIME Edge
 KNIME Randdiagramm
 Fetching der Helm
 Werte Dateikonfiguration
 Installation des KNIME Edge
Installation von KNIME Randdiagramm
 Verbinden Sie mit dem Helm
 Installieren Sie den Edge

[Zugang zur KOTS Admin Control Panel](#page16)

[Übertragung der Daten aus dem KNIME Edge zu KNIME Katalyst](#page18)

[Konfigurieren Sie die Helm- und Edge-Host-Systeme](#page19)

[Bestätigung eines KNIME Edge Host Systems](#page21)

[Installation von KNIME Edge Host Systemen](#page21)

[Dolmetschen der Podcasts des KNIME Edge Host Systems](#page22)

[Prüfung des KNIME Edge Host Systems](#page23)

[Fortgeschrittene Operationsfunktionen](#page24)

[Fehler: Verfehlt, Bildschirm nicht erreichbar](#page24)

[Fehler: Unable to access Helm](#page26)

[Aktualisierung eines KNIME Edge Host Systems](#page27)

[Fehler: Helm Upgrade failed](#page27)

Einleitung

Diese Anleitung beschreibt die Anforderungen, Überlegungen und Schritte zur Erstellung eines KNIME Edge Cluster.

Komponenten von KNIME Rand

Kurze Beschreibungen der hochrangigen architektonischen Komponenten, die KNIME Edge umfassen.

KNIME Kantenkontrollebene

Im Rahmen von Kubernetes bezieht sich der Begriff "Kontrollebene" auf eine Gruppe von Komponenten dass mit der Verwaltung des aktuellen Zustands eines Clusters umgehen. Bei KNIME Edge-Clustern, diese Cluster können an verschiedenen Orten wie Rechenzentren installiert werden, Herstellung Einrichtungen, Cloud-Anbieter oder Umgebungen mit eingeschränktem Internetzugang. Um die Kante zu gewährleisten Cluster können richtig funktionieren, sie müssen periodische Kommunikation mit einer Steuerung aufbauen eine Ebene, um die gewünschte Konfiguration für ihre Bereitstellungen abzurufen.

KNIME Business Hub hat die Möglichkeit, als Kontrollebene für einen KNIME Edge Cluster zu fungieren. Sie erreichen dies durch eine Sammlung von KNIME Workflows, die als interaktive Daten dienen Apps. Diese Workflows ermöglichen die Erstellung, Aktualisierung und Löschung von Kanteneinsätzen. Die Der Zustand des Clusters wird durch die Nutzung einer PostgreSQL-Datenbank, die in KNIME Business Hub.

KNIME Edge Operator

Der KNIME Edge Operator hat zwei Aufgaben. Erstens übernimmt sie die Verantwortung für Initialisierung des KNIME Edge-Stacks, um sicherzustellen, dass alle notwendigen Komponenten ordnungsgemäß sind eingerichtet. Zweitens spielt es eine entscheidende Rolle bei der Vereinbarkeit von Bereitstellungen, die konfiguriert wurden durch die Steuerebene. Dies bedeutet, dass, wenn eine Bereitstellung erstellt, aktualisiert oder gelöscht, der Bediener übernimmt die Aktion, indem er die relevanten Kubernetes-Ressourcen innerhalb der Cluster, um die gewünschten Änderungen zu reflektieren.

KNIME Edge Inference Agent

Das KNIME Edge Inference Agent bietet eine leichte REST API auf KNIME Analytics Plattform, die es KNIME ermöglicht, Workflows in Containerbilder zu bauen, so dass sie gehostet in einem Kubernetes-Cluster und skaliert nach Bedarf hinter einem Netzlastausgleich. Nur

ein Workflow kann pro Containerbild gehostet werden, aber mehrere Instanzen des Workflows können in den Speicher geladen werden. Dieser Ansatz hilft, den Durchsatz und die Leistung für KNIME Workflow Ausführung.

Beschäftigung

Ein Inferenzeinsatz (Beschäftigung) ein [Kubernets benutzerdefinierte Ressourcen](#) dass enthält die Definition für eine KNIME Edge-Bereitstellung, einschließlich Konfigurationsoptionen wie als:

- Der KNIME-Workflow zum Einsatz
- Ressourcenzuweisung
- Skalierungskonfiguration

Kongo API Gateway

Kong ist ein Open-Source-, Drittanbieter-Stack, der bietet KNIME Mehr Flexibilität und Funktionalität, wie externe API-Anfragen bearbeitet und auf Inferenzeinsätze proxidiert werden.

MinIO

MinIO ist ein eingebetteter Objektspeicher (der die gleiche API wie der AWS S3 Service nutzt). KNIME Edge verwendet MinIO, um Workflows, Protokolle und andere Artefakte zu speichern, um den Betrag zu optimieren vom KNIME Business Hub verlangt.

KNIME Architektur

Im Folgenden finden Sie hochrangige Diagramme der architektonischen Komponenten in KNIME Edge. In KNIME Rand 1.3+, KNIME Business Hub kann als Steuerebene für KNIME Edge verwendet werden.

Verwendung von KNIME Business Hub als Kontrollflugzeug

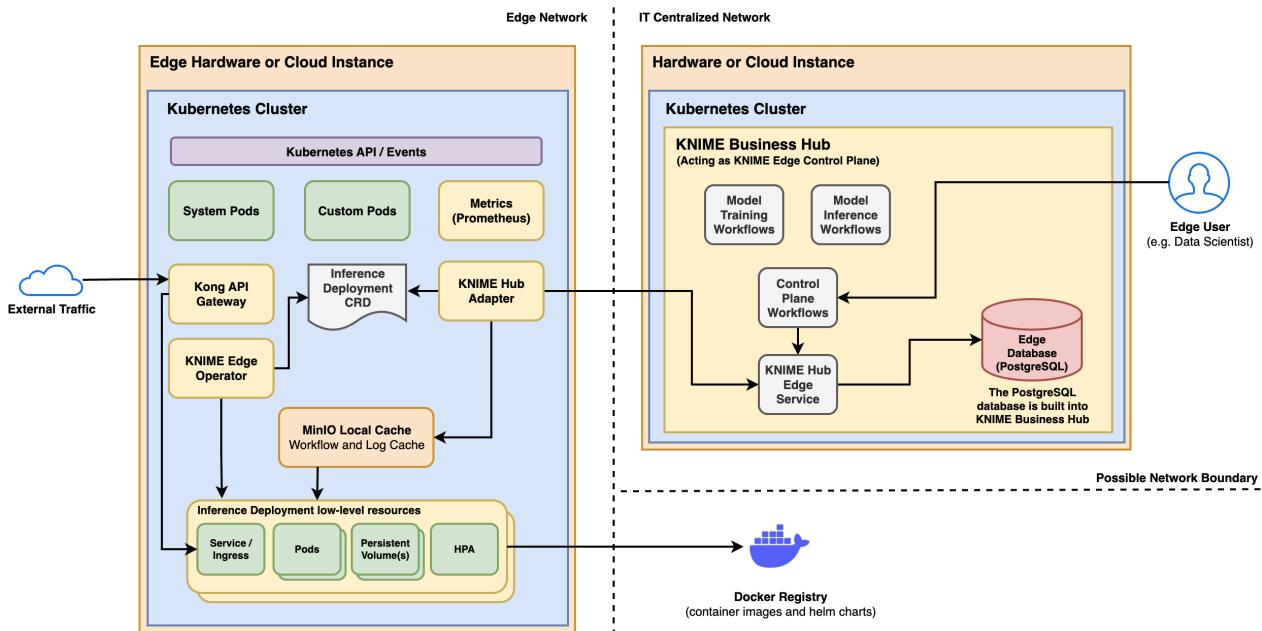
Es gibt mehrere Vorteile, die KNIME Business Hub als Steuerungsebene für KNIME nutzen Rand:

- Installation und Konfiguration der KNIME Edge-Steuerungsebene ist durch zur verbesserten Architektur des KNIME Business Hubs.

- Die PostgreSQL-Datenbank zum Speichern der KNIME Edge-Konfiguration ist im Lieferumfang enthalten KNIME Business Hub und erfordert keine manuelle Konfiguration.
- Die KNIME-Workflows, die als Kontrollebene wirken, sind einfacher zu implementieren und zu verwenden.

KNIME Edge Architecture Diagram

(Using KNIME Business Hub as the control plane)



Installationsplanung

Kontaktieren Sie Ihren Kundenbetreuer, wenn Sie Hilfe bei der Installationsplanung benötigen.

Installationsmethode: Kurl gegen Helm

KNIME Edge unterstützt zwei Arten von Kubernetes Cluster-Installationen, **eingebetteter Cluster** und **bestehendes Cluster**. Für eingebettete Cluster ist Kurl die empfohlene Installationsmethode. Für bestehende Cluster, Helm ist die empfohlene Installationsmethode.

• Embedded Cluster Installation über kurl

- Bietet eine geführte Installation eines Kubernetes-Clusters mit [Kurl](#).
- Future Kubernetes-Cluster-Versionsaktualisierungen werden durch Neulauf des Kurls durchgeführt Installer. Die Versionskompatibilität wird vom Installer vollständig erkannt und verwaltet.
- Best für Kunden, die neu sind, um Kubernetes-Anwendungen zu verwalten, aber auch ideal für alle Kunden, die von automatisiertem Cluster profitieren möchten Installation/Upgrades.

- Vereinfacht den Cluster-Installations- und Upgrade-Prozess.
- Neue Knoten können mit Kurl, aber Cluster, angehängt oder aus dem Cluster entfernt werden
autoscaling wird nicht unterstützt.

• **vorhandene Cluster-Installation über Helm**

- Erfordert den Kunden, einen eigenen Cluster bereitzustellen und aufrechtzuerhalten, der
in der [Repliziert bestehende Cluster-Installationsanforderungen](#) zusätzlich zu jeder
Anforderungen in dieser Anleitung.
- Erfordert den Kunden, die Kompatibilität der Clusterversion mit KNIME Edge sicherzustellen.
- Beste für Kunden, die in Kubernetes erfahren und mehrere Cluster haben
heute in der Produktion. Insbesondere diese Option ist ideal für Kunden
deren IT-Abteilungen etablierte Politiken und Praktiken für
Pflege von Kubernetes Clustern.
- Diese Option ist in Bezug auf die erste Konfiguration schwieriger und erfordert mehr
vom Kunden in Bezug auf laufende Wartung, aber ermöglicht fortgeschrittene
Konfigurationen.
- Mit einer bestehenden Cluster-Installation ist es möglich, Cluster zu nutzen
autoscaling Funktionen von Cloud-Anbietern wie EKS, GKE oder AKS.

KNIME Business Hub



Unterstützte Versionen: KNIME Business Hub 1.4.0 +

KNIME Business Hub Standard (oder höher) kann als Steuerungsebene für KNIME Edge verwendet werden.

Wenn Sie an KNIME Edge interessiert sind und mehr über die Lizenzierung verstehen möchten, bitte

[Kontaktieren Sie uns](#)[Kunden Pflegeteam](#).

Kubernetes



Unterstützte Versionen: v1.21 bis v1.30

Ein Kubernetes-Cluster muss für die Funktion von KNIME Edge verfügbar sein. In einem eingebetteten Cluster
Installationen mit kurl, der Cluster ist im Rahmen des Installationsprozesses vorgesehen. In allen
anderen Konfigurationen, muss der Cluster vom Kunden bereitgestellt (und aktualisiert) werden. Vgl.
die [Kubernetes Dokumentation](#) für Details zur Clusterinstallation und -nutzung.

Softwarevoraussetzungen

- **kubectl** : nur bei der Installation in einen bestehenden Cluster oder bei der Fernverwaltung einer Cluster. Beim Installieren des eingebetteten Clusters mit kURL kubectl automatisch installiert auf der Host-Maschine.
- **Helm** : nur bei der Deinstallation von KNIME Edge erforderlich.
- **Empfohlen:** eine Kubernetes integrierte Entwicklungsumgebung (IDE) einfacher Inspektion von Kubernetes Ressourcen innerhalb des KNIME Edge Stacks. Es gibt viele Optionen auf dem Markt wie Lens und OpenLens.

Netzwerkvoraussetzungen

Trusted Host	Gebiet
KNIME	https://registry.hub.knime.com , https://registry.hubdev.knime.com ,
Replika	Siehe Firewall-Eröffnung für Online Installationen Führung.

Kapazitätsplanung & Erwägungen



Speicher pro Pod ist der primäre Treiber der Infrastrukturkosten in einem KNIME Edge Cluster.

Schlussfolgerung

Warnung : Wird einem Inference Agent pod ein unzureichender Speicher zugewiesen, wird der Pod wird wahrscheinlich abstürzen und Kubernetes wird eine **OOMKilled** Veranstaltung für diesen Pod.

Bei der Berechnung der benötigten Kapazität in Kubernetes für einen KNIME Edge Cluster, die **größter Faktor** wird die erwartete Anzahl von Inference Deployment Pods sein, die laufen wird . In Replizierte Installationen, es gibt auch [Mindestsystemanforderungen](#) dass hinzugefügt werden sollte über die erwartete Kapazität.

Die Basiskomponenten von KNIME Edge erfordern minimale Ressourcen, typischerweise weniger als 1 CPU und ca. 1 GB Speicher. Der KNIME Edge Inference Agent pod(s), der eingesetzt wird für jeden Inference Deployment benötigen zusätzliche Ressourcen. Für die meisten Inferenz-Workflows, es wird empfohlen, jedem entsprechenden KNIME Edge Inference Agent pod eine obere Grenze **1 bis 2 CPUs** und eine obere Grenze **2 GB RAM**.

Beachten Sie, dass jedes Pod, einmal vollständig gestartet, in der Regel rund **1,5 bis 2 GB aktiv**

Speicherauslastung . Diese Speicherauslastung ist meist statisch und nimmt typischerweise nicht zu oder Spike mit Anfragelast.

Die CPU wird im Vergleich stark durch den Anforderungsdurchsatz beeinflusst. CPU-Nutzung wird in der Regel spike während Anträge bearbeitet werden und sich auf ein geringeres Auslastungsniveau einstellen, während auf neue Anfragen warten.

Eine nützliche Methode für die Kapazitätsanforderungen ist die Prüfung eines einzelnen Pods, die verwendet den beabsichtigten Inferenz-Workflow und die Anzahl der bearbeiteten und latenten Anträge messen pro Anfrage. Durch die Bewertung der erwarteten Zahl der gleichzeitigen Inferenzausgaben und Anzahl der in jeder Bereitstellung laufenden Pods, es ist möglich, die Gesamtkapazitätsanforderungen zu schätzen.

Es wird empfohlen, einen ~20% Puffer auf der Oberseite der geschätzten Kapazität für Skalierbarkeit hinzuzufügen. Wenn Großformat KNIME Rand-Cluster wird geplant, die [Überlegungen für große Cluster](#) Seite aus der Kubernetes-Dokumentation kann nützlich sein.

KNIME konfigurieren Business Hub

Dieser Abschnitt zeigt, wie man KNIME Business Hub Standard 1.4.0 + -Instanz konfiguriert (oder höher) als Kontrollebene für KNIME Edge.

Das KNIME hochladen Edge Control Plan Workflows zu KNIME Business Hubraum

Um mit KNIME Edge und KNIME Business Hub zu beginnen, folgen Sie diesen Schritten, um die KNIME Kantenkontrollebene Workflows am Hub. Die Steuerebene besteht aus einem Datensatz Apps, die die Steuerung und Interaktion mit dem entfernten KNIME Edge Cluster ermöglichen.

Bevor Sie fortfahren, stellen Sie sicher, dass Sie ein KNIME Community Hub Konto haben. Wenn Sie nicht haben [Sie können sich bei](#) <https://hub.knime.com/> . Sobald Sie ein Konto haben, können Sie auf Das KNIME Kantenkontrollebenen-Workflows in der [KNIME Rand](#) Raum.

Es gibt zwei Arten von KNIME Edge-Steuerungsebenen-Workflows:

- **Verwaltungsabläufe**

- Diese Workflows finden Sie im Ordner "Administration".
- Dazu gehören Datenapps für die Verwaltung von Ausführungsbildern und die Überwachung des Status von verbundenen Clustern.

- **Inferenz-Einsatz-Workflows**

- Diese Workflows befinden sich an der Wurzel der KNIME Edge Steuerebene Workflows.
- Sie stellen Daten-Apps für das Erstellen, Lesen, Aktualisieren und Löschen von Inference zur Verfügung Bereitstellungen.

Wenn Sie Administrator des KNIME Business Hub sind, folgen Sie den folgenden Schritten, um die KNIME Edge Control Plan Workflows und stellen Sie sie Ihrem Team zur Verfügung:

• **Erstellen Sie einen neuen Raum und/oder Team im KNIME Business Hub.**

- Zum Beispiel können Sie ein Team namens "KNIME Edge" und einen Raum namens "KNIME Edge Control Plane".

• **Das KNIME hochladen Kantensteuerungsebenen-Workflows zum bezeichneten Raum.**

- Achten Sie darauf, die Standard-Ordner/Workflow-Struktur aufrechtzuerhalten, um jegliche unerwartete Fehler bei der Ausführung.
- Der Ordner "Inference Deployments" sollte sowohl eingesetzte Workflows als auch die bereit für den Einsatz.
- Dieser Ordner dient als zentraler Standort für die Verwaltung aller Workflows im Zusammenhang mit Inferenzeinsätze.
- Sobald die Workflows hochgeladen sind, ist keine zusätzliche Konfiguration erforderlich, um die Daten-Apps verwenden. Sie können direkt zugreifen und nutzen sie von der benannten Raum.

Vor der Bereitstellung eines Workflows als Inferenz-Bereitstellung in KNIME Edge, es sollte in den Ordner "Inference Deployments" im "KNIME" hochgeladen werden Edge Control Plane" Raum. Einmal hochgeladen, wird es als wählbare sichtbar sein Workflow in der Erstellung von Inference Deployment-Daten-App.

The screenshot shows the KNIME Business Hub interface. At the top, there is a navigation bar with the KNIME logo, a search bar containing 'Search workflows, nodes and more...', and a user profile icon. Below the navigation bar, a breadcrumb trail shows the path: KNIME Business Hub > KNIME Edge > Spaces > KNIME Edge Control Plane > Edge Control Plane. The main content area is titled 'KNIME Edge Control Plane' and includes a 'Working area' section with a list of deployment-related options: Administration, Inference Deployments, Create Inference Deployment, Delete Inference Deployment, Update Inference Deployment, and View Inference Deployments. There are also icons for Space history, heart, copy link, and help.

Konfigurieren Sie den KNIME Hub Edge Service auf dem KNIME Business Hub

Wenn Sie KNIME Business Hub 1.4.0+ nutzen und KNIME Edge auf Ihrem Konto aktiviert haben

Repliziert (.yaml) Lizenz, eine Option für die Freigabe von KNIME Edge sollte im KNIME angezeigt werden

Business Hub KOTS Admin Console.



Kontaktieren Sie Ihren Kundenbetreuer, wenn KNIME Edge aktiviert werden muss auf Ihre Replica-Lizenz.



Diese Einstellung ist in der KOTS Admin Console von KNIME Business Hub sichtbar und wird wahrscheinlich vom Systemadministrator aktiviert werden müssen. Einmal aktiviert, bereitstellen die Änderungen, um den KNIME Hub Edge Service online zu bringen.

KNIME Edge

Configuration for KNIME Edge.

Enable KNIME Edge

Enabling this option turns on the KNIME Hub Edge Service, which provides an integration point for the KNIME Edge Hub Adapter.

Ein Hinweis auf die Autorisierung

Die Kommunikation zwischen der KNIME Edge Steuerungsebene und jedem gegebenen KNIME Edge

Cluster erfolgt über REST API Anrufe in den KNIME Hub Edge Service. Der KNIME Edge Hub

Adapter behandelt die Extraktion des Berechtigungs-Tokens, der als Anfrage-Header verwendet wird jeden Anruf beim KNIME Edge Service durch die Steuerungsebenen-Workflows zu autorisieren.

- Es ist wichtig zu beachten, dass diese Autorisierungsmethode nur funktioniert, wenn sie läuft Daten-Apps zum KNIME Business Hub. Die Datenapps können nicht lokal ausgeführt werden in der KNIME Analytics Platform und noch erfolgreich authentifizieren.

Installation von KNIME Edge mit KNIME Business Hub und Helm

Helm ist eine fortschrittliche Installationsmethode im Vergleich zum kurl installer und ist die empfohlene Installationsmethode für KNIME Edge, wenn Sie KNIME Edge in ein bestehender Kubernetes-Cluster. Installation über Helm gibt Ihnen mehr Konfigurationsoptionen gegenüber der Installation über Kurl.

Wahl eines Kubernetes Namespace

- | | |
|---|------------------------|
| Installation von KNIME Rand in einen anderen Namensbereich | Standard Kubernetes |
| Namespace ist sehr empfehlenswert , aber optional. In Kurlanlagen, KNIME | |
| Edge wird in der Rand Namespace. Für weitere Informationen siehe | |
| Kubernetes Namespace Dokumentation | |

Mit einem Kubernetes Namensraum wird empfohlen, da es alle KNIME Edge verkapselt Ressourcen in einem Bereich und isoliert sie von anderen Ressourcen, die innerhalb der Kubernetes-Cluster. Die KNIME Edge-Installation in einen Namensraum zu schotten macht auch jede potenzielle manuelle Wartung sicherer seit `kubectl` Befehle können eingeschränkt werden auf einen bestimmten Namensraum über den optionalen `[-n]` Flagge.

Dieser Leitfaden wird sich auf diesen optionalen, aber empfohlenen Namensraum beziehen, gegebenenfalls mit `[-n]`.

Wenn gewünscht, erstellen Sie einen Namensraum mit dem folgenden Befehl:

```
kubectl erstellen namespace
```

Einloggen in die KNIME Artifact Registry

Eine gültige KNIME Community Hub-Benutzer (mit erhöhten Berechtigungen) ist erforderlich, um Zugang zu knime-edge Projekt, Docker-Bilder und Bereitstellungsdiagramme in Harbor. Kontakt Support@knime.com wenn Hilfe benötigt wird, um Projekte in Harbor.

Die für einen KNIME Edge Cluster benötigten Docker-Bilder sind im KNIME Artifact erhältlich

Registrierung: <https://registry.hub.knime.com>, und in einigen Fällen

<https://registry.hubdev.knime.com>

Die knime-edge Bilder und Helmkarten [finden Sie hier](#).

Melden Sie sich mit einem gültigen KNIME Community Hub-Nutzer über die **OIDC Anmelden Knopf**.

Helm konfigurieren

Um einen KNIME Edge Cluster direkt von der [KNIME Art und Weise](#), Sie müssen Fügen Sie das KNIME Edge-Diagramm-Repository hinzu.

Authentisierung mit der [KNIME Art und Weise](#) und [KNIME Randdiagramm-Repository](#) erfordert Benutzernamew^{und} Passwort vergessen? Die Benutzernamew^{ist} das gleiche wie Ihr Hub-Benutzername. Die Passwort vergessen?^{bis} angeben ist ein CLI Geheimnis von Harbor bereitgestellt. Siehe [Verwendung von OIDC aus dem Docker oder Helm CLI](#) Informationen zum Abrufen des CLI-Geheimnisses (das sowohl den Helm als auch Docker Passwort).

Um das Helm-Diagramm-Repository hinzuzufügen, führen Sie den folgenden Befehl aus (mit kubectl bereits verfügbar und konfiguriert), und ersetzen die Benutzernamew^{und} Passwort vergessen?.

```
helm repo add --username --password knime-edge \
https://registry.hub.knime.com/chartrepo/knime-edge
```

Hinzufügen eines Registrierungsgeheimnisses zu Kubernetes

Um Docker Bilder direkt von der [KNIME Art und Weise](#), a **Kubernetes Geheimnis** mit dem Namen Regcred muss im KNIME Edge Cluster erstellt werden. Das Geheimnis zu schaffen ist durchgeführt mit kubectl CLI.

Stellen Sie sicher, dass das Geheimnis in der gleichen Namespace wenn **KNIME Rand** ist zur Bereitstellung.

```
kubectl [-n] erstellen geheime Docking-Registry regcred \
--docker-server=
--docker-username= \
--docker-password=
```

Überprüfen Sie durch Laufen:

```
kubectl [-n] get secret regcred
```

Anwendung eines KNIME Edge-Lizenz

Für die KNIME Edge-Cluster zum Arbeiten ist eine Lizenz erforderlich. Weitere Informationen zum

Lizenzierung in KNIME Edge kann in der

[KNIME Rand-Licensing-Abschnitt](#)

vom KNIME

Edge User Guide.

Sobald eine Lizenz erhalten wurde, **geheimnis** mit dem Namen **knime-edge-license** enthält die Lizenzdatei muss im Kubernetes-Cluster erstellt werden.

Stellen Sie sicher, dass das Geheimnis in der gleichen **Namespace** wenn **KNIME Rand** ist zur Bereitstellung.

```
kubectl [-n] kreieren geheime Generika knime-edge-license --from
-file=license=
```

KNIME Randdiagramm

Das KNIME Rand Diagramm ist ein Steuerdiagramm, das die CRDs und Abhängigkeiten definiert, die von ein KNIME Edge Cluster.

Die Parameter, die zur Konfiguration eines KNIME Edge Clusters erforderlich sind, werden in einem Standard angegeben.

Helm Werte Datei, z. **Werte.yaml**, die aus der **knime-edge** Helm Repository.

Die folgenden Schritte beschreiben den Prozess der Gewinnung, des Verständnisses, der Anpassung und Installieren Sie dieses Diagramm.

Starten der Helm-Werte-Datei für das KNIME Edge-Diagramm

Verfügbare Diagramme aktualisieren

```
helm repo update
```

Bestimmen der Version :

Sie können alle veröffentlichten Versionen des KNIME Edge anzeigen Helmkarten in der [Harbor Registrierung](#).

Siehe das README in jeder Chartversion für die für diese Version spezifische Wertedeidefinition. Eine Übersicht über YAML-Techniken, die in den Werten verwendet werden können. yaml-Datei kann in der [Heldokumentation](#).

(Erweitert) Alle Diagrammversionen können mit dem folgenden Befehl aufgelistet werden:

```
helm search repo knime-edge/knime-edge-operator --versionen --devel
```

Werte Dateikonfiguration

Die Datei generieren Werte.yaml um Parameter für den KNIME Edge Cluster zu konfigurieren:

Für die neueste stabile Veröffentlichung:

```
helm show value knime-edge/knime-edgeoperator > Kantenwerte. Yaml
```

(Erweitert) Für eine bestimmte Version laufen:

```
helm show value knime-edge/knime-edge-operator -Version > Rand  
Werte.yaml
```

Jetzt öffnen Sie die Rand-Werte.yaml in einem Texteditor und folgen Sie den Anweisungen in der Datei:

Kommentarzeichen in der Rand-Werte.yaml Datei angeben, welche Felder müssen vor der zu installieren und welche Werte für erweiterte Konfigurationen übergeordnet werden können.

Installation des KNIME Rand Helm Diagramm

Sobald Sie die Wertedei mit der für Ihre Umgebung relevanten Konfiguration aktualisiert haben, haben Sie kann KNIME installieren Edge zu Ihrem Kubernetes-Cluster mit dem folgenden Befehl.

```
helm install [-n ] knime-edge knime-edge/knime-edgeoperator -f edge-  
Werte.yaml
```

(Erweitert) Wenn Sie eine bestimmte Chart-Version verwenden, installieren Sie mit:

```
helm install [-n ] knime-edge knime-edge/knime-edge-operator --version
-f Kantenwerte. Yaml
```

Installation von KNIME Edge mit KNIME Business Hub und Kurl

[Kurl](#) ist die empfohlene Installationsmethode für KNIME Edge, wenn Sie KNIME installieren möchten Rand in ein eingebettetes Kubernetes-Cluster. Es wird ein Kubernetes Cluster einrichten und installieren KNIME In den Rand.

Für die unten gezeigten Befehle ersetzen Sie alles, was in Werte. mit real

Kontaktieren Sie Ihren Host VM

Der nächste Schritt ist die Verbindung mit Ihrem Host VM und die Aktualisierung. Wenn Sie über SSH verbinden, sicherstellen, dass die von Ihnen verwendete Maschine mit dem Port 22 der Instanz verbunden werden kann. Auch sicherstellen, dass der Benutzer, den Sie mit der Instanz verbinden, über Berechtigungen verfügt, Befehle auszuführen, der Superuser (d.h. Sudo)

```
Verbinden Sie mit Ihrer Instanz. Dieser Prozess/Befehl kann je nach Betriebssystem unterschiedlich sein.
ssh -i "some-identity-key.pem" ubuntu@
```

```
# Aktualisieren Sie die Instanz. Dieser Prozess/Befehl kann je nach Betriebssystem unterschiedlich sein.
sudo apt-get update && sudo apt-get upgrade
```

Installieren Sie den eingebetteten Cluster für KNIME Edge

Der folgende Befehl führt eine Freisprechanlage aller unterstützenden Kubernetes aus Architektur erforderlich, um KNIME Edge zu hosten. Es dauert etwa 10-15 Minuten, um in seiner Gesamtheit zu laufen und wird eine beträchtliche Anzahl von Protokollen ausgeben, da der Prozess alle notwendigen Installationen installiert Abhängigkeiten.

```
curl -sSL https://kurl.sh/knime-edge | sudo bash
```

Für erweiterte Installationsoptionen mit kURL konsultieren Sie bitte [kURL Dokumentation](#).
Bitte beachten Sie: Wenn Sie diesen Befehl mit zusätzlichen Flaggen oder Umgebungsvariablen ausführen
Bitte beachten Sie sie in einem Dokument. Die gleichen Fähnen und Umgebungsvariablen

muss wieder vorhanden sein, wenn Sie die kubernetes Cluster-Version oder KOTS Admin aktualisieren Konsole.

Sobald der Prozess abgeschlossen ist, sollten Sie etwas ähnlich wie die folgende Ausgabe sehen. Das Ausgabe enthält sehr wichtige URLs, Benutzernamen, Passwörter und Befehle für Ihre Ein Beispiel. Stellen Sie sicher, dass Sie diese Ausgabe irgendwo sicher speichern, bevor Sie fortfahren.

```
Installation
Complete ✓

Kotsadm: http://[REDACTED]:8800
Login with password (will not be shown again): [REDACTED]
This password has been set for you by default. It is recommended that you change this password; this can be done with the following command: kubectl kots reset-password default

The UIs of Prometheus, Grafana and Alertmanager have been exposed on NodePorts 30900, 30902 and 30903 respectively.
To access Grafana use the generated user:password of [REDACTED] .

To access the cluster with kubectl, reload your shell:
bash -l

Node join commands expire after 24 hours.
To generate new node join commands, run curl -fsSL https://kurl.sh/version/v2022.08.25-0/knime-hub/tasks.sh | sudo bash -s join_token on this node.
To add worker nodes to this installation, run the following script on your other nodes:
[REDACTED]
```

Zugang zur KOTS Admin Console

Navigieren Sie zur KOTS Admin Console URL in der eingebetteten Cluster-Installation ausgeben und beachten Sie das Passwort.

```
Installation
Complete ✓

Kotsadm: http://[REDACTED]:8800
Login with password (will not be shown again): [REDACTED]
```

Die erste Seite, die angezeigt wird, ist eine Warnung bezüglich Transport Layer Security (TLS) Konfiguration. Folgen Sie den Anweisungen auf dem Bildschirm.

Bypass browser TLS warning

We use a self-signed SSL/TLS Certificate to secure the communication between your local machine and the Admin Console during setup. You'll see a warning about this in your browser, but you can be confident that this is secure.

Chrome

On the next screen, click "Advanced", then click "Proceed" to continue to the Admin Console.

Verifying the certificate's authenticity

SHA Fingerprint

Continue to Setup or visit <https://app.mycompany.com:8800/tls> to proceed

Sie werden dann aufgefordert, Ihren eigenen TLS-Zertifikat zur Sicherung des Verkehrs an den Admin bereitzustellen Konsole, falls gewünscht.

HTTPS for KNIME Edge the admin console

Certificate type

Self-signed Upload your own

A self-signed TLS certificate is currently used to secure communication between your browser and the admin console. You will see a warning in your browser every time you access the admin console unless you upload your own TLS certificate.

Hostname (optional)

Ensure this domain is routable on your network.

app.yourdomain.com

Continue

Sie sollten dann eine Aufforderung für ein Passwort sehen. Geben Sie das Admin-Konsole-Passwort von der Eingebundene Cluster-Installationsausgabe (dieses Passwort kann später geändert werden).



Log in to KNIME Edge

Enter the password to access the KNIME Edge admin console.

Log in

Bieten Sie ein Replikat

.yaml Lizenzdatei

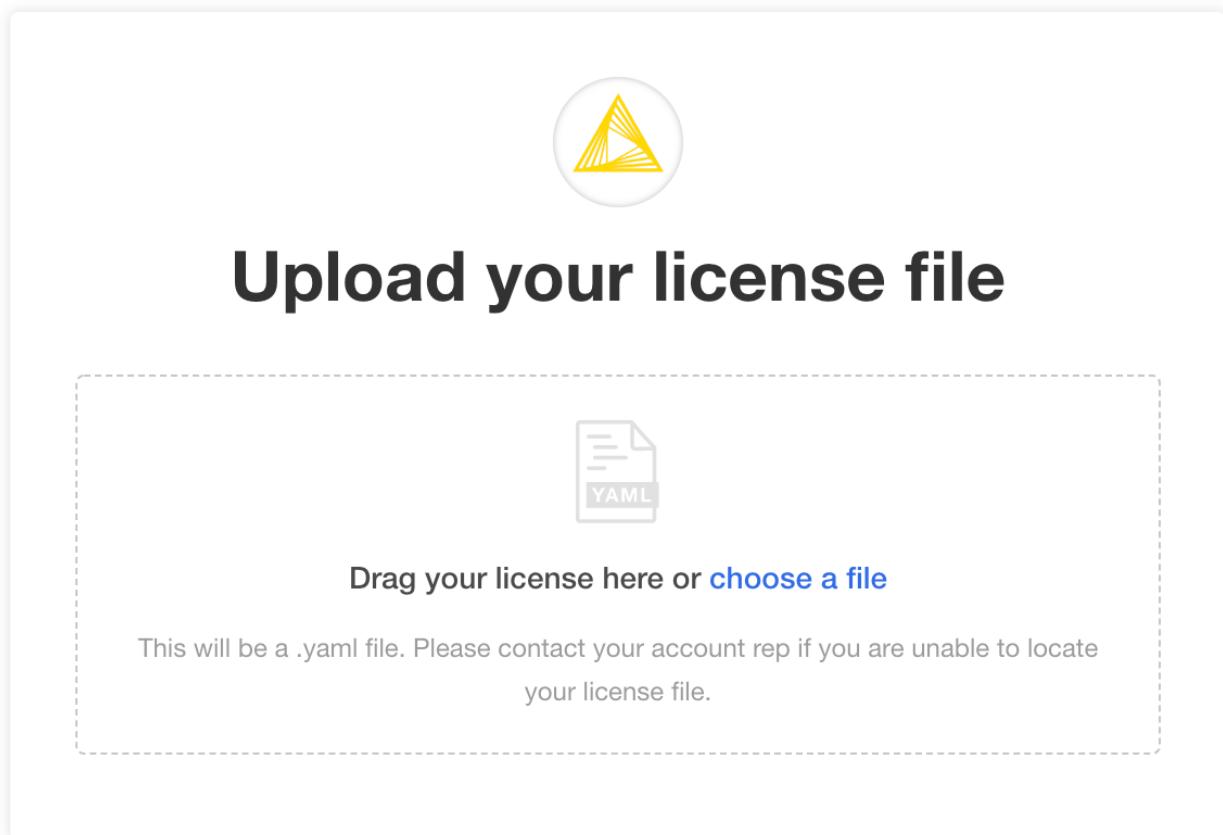
Nach dem Login sollten Sie für eine Lizenzdatei aufgefordert werden. Dies ist die Replicat Lizenz-Datei

dass Ihr KNIME Kundenbetreuer Ihnen zur Verfügung gestellt hat und

.yaml

Erweiterung. Bitte kontaktieren Sie Ihren Kundenbetreuer, wenn Sie Hilfe benötigen

Ihre Lizenz.



Konfigurieren der Installation

Wenn alle vorherigen Schritte erfolgreich waren, sollten Sie nun aufgefordert werden, Ihren KNIME Edge zu konfigurieren. Eine Reihe von Einstellungen wird angezeigt, um anzupassen.

Global

KNIME Edge Cluster Name
KNIME Edge Cluster Description
KNIME Edge Cluster Location (e.g. "us-east-1")
Image Pull Policy
Node Selection ▾
Inference Deployments ▾
Ingress Configuration ▾
KNIME Hub Adapter ▾

Global

High-level configuration for the KNIME Edge deployment.

KNIME Edge Cluster Name Required

The name of the KNIME Edge deployment. Should be lowercase with letters, numbers, and hyphens only (no spaces). Displays in the KNIME Hub Control Plane workflows.

`edge`

Default value: `edge`

KNIME Edge Cluster Description Required

The description of the KNIME Edge deployment. Displays in the KNIME Hub Control Plane workflows.

`KNIME Edge`

Default value: `KNIME Edge`

KNIME Edge Cluster Location (e.g. "us-east-1") Required

The location (i.e. region) of the KNIME Edge deployment. Displays in the KNIME Hub Control Plane workflows.

`us-east-1`

Image Pull Policy

Determines the image pull policy for KNIME Edge components. See the [Kubernetes Image Pull Policy documentation](#) for more details.

If Not Present Always Never

Bitte achten Sie auf die

KNIME Zubehör und ZubehörAbschnitt, der Konfiguration enthält

erforderlich für KNIME Edge, um mit dem KNIME Business Hub Steuerungsflugzeug zu verbinden.

Das KNIME Business Hubinstanz, die als Steuerung verwendet werden soll

Flugzeug für KNIME Edge sollte DNS konfiguriert und TLS aktiviert haben, bevor

Installation von KNIME Rand. Bitte beachten Sie, dass KNIME Business Hub Standard der

Mindestwert für KNIME Edge.

Die KNIME Hub URL sollte mit `api.` , z.

<https://api.hub.example.com/> . Dies ist die Adresse, die KNIME Edge verwendet

bei der Kommunikation mit der KNIME Business Hub API.

Der Benutzername und das Passwort bei der Verbindung mit KNIME Business Hub sollten sein Anwendungspasswort aus dem KNIME Business Hub.

Der Benutzer, der das App-Passwort generiert, sollte Admin-Zugriff auf den Raum haben

in denen KNIME Edge-Workflows werden eingesetzt. Siehe [Anwendungspasswörter](#)

Abschnitt der **KNIME Benutzerhandbuch des Business Hub** für weitere Informationen

Erstellung von App-Passwörtern.

KNIME Hub Adapter

The KNIME Edge Hub Adapter is responsible for all communication between KNIME Edge and the KNIME Hub Control Plane.

Enable KNIME Hub Adapter Recommended

The KNIME Edge Hub Adapter must be enabled for KNIME Edge to communicate with KNIME Hub.

Skip TLS Verify

Determines whether Transport Layer Security (TLS) verification is enabled for all communication between KNIME Edge and KNIME Hub.

KNIME Hub URL Required

`https://api.hub.example.com/`

Default value: `https://api.hub.example.com/`

KNIME Hub Username Required

`9HfuV8DSIuVbXWzeFLuR9iTjJfE7qHilRgReVhYuleF`

Default value: `knimeadmin`

KNIME Hub Password Required

.....

Bestätigung eines KNIME Edge Cluster ist Betrieb

Installation von KNIME Edge Cluster überprüfen

Bei der Erstinstallation sollte der KNIME Edge Operator und KNIME Edge Hub Adapter Pods sein und laufen. Die Kubectl CLI kann verwendet werden, um den Status zu überprüfen (die Pod-ID wird anders in Ihrem Cluster:

```
> kubectl [-n namespace] get pod -knime-edge-operator-6574d6fd79-stlss

NAME READY STATUS RESTARTS
# KNIME Edge Operator
-knime-edge-operator-6574d6fd79-stlss 1/1 Laufen 0 7m43
```

```
> kubectl [-n namespace] get pod hub-adapter-697768bc8c-g9zjw

NAME READY STATUS RESTARTS
# KNIME Auf den Seiten
Nabenadapter-697768bc8c-g9zjw 1/1 Laufen 0 7m40s
```

Wenn der KNIME Edge Hub Adapter nicht existiert, könnte es ein Problem im KNIME Edge geben Betreiber. Um die Protokolle des KNIME Edge Operator Pod zu überprüfen, führen Sie:

```
> kubectl [-n namespace] logs -knime-edge-operator-6574d6fd79-stlss
```

Zusätzlich zu überprüfen, ob die Pods laufen, sollte die Abfrage des Endpunktes eine leere Liste für inferenceBeschäftigte :

```
% curl -sL http://localhost:8081/ | jq
{
  "inferenceDeployments": []
}
```

Das KNIME Edge Operator erstellt im KNIME Edge Cluster zusätzliche Pods, die alle sind entweder im laufenden oder abgeschlossenen Zustand. Weitere Informationen zu diesen Pods finden Sie in der Abschnitt unten.

Interpretieren der Pods im Cluster

Mit dem Kubectl CLI können die Pods in einem KNIME Edge Cluster untersucht werden. Unten ist Ausgeben der Probe aus der kubectl [-n] get pods Befehl (wobei -n definiert den Namenspace) mit Kommentaren hinzugefügt, um die verschiedenen Pods zu identifizieren und ihre Rollen:

```
> kubectl [-n ] get pods

NAME READY STATUS RESTARTS
# Register Inference Deployments as Kubernetes Endpoints
Endpunkt-Entdeckung-depl-6457b745b7-pkb24 1/1 Laufen 0 7m40s

# Logging Service
fluentd-ds-6q8l6 1/1 Laufen 0 7m41s
fluentd-ds-562zj 1/1 Laufen 0 7m41s

# KNIME Edge Operator
knime-edge-knime-edge-operator-6574d6fd79-stlss 1/1 Laufen 0 25m

# Kong
knime-edge-kong-5bb658458f-znwsg 2/2 Laufen 0 25m

# Workflow und Log Cache
minio-b9b9547dd-xbvs2 1/1 Laufen 0 7m41s

# Inference Deployment
sentiment-predictor-7d56b6cd47-lp8gp 1/1 Laufen 0 6m31s

# KNIME Auf den Seiten
Nabenadapter-697768bc8c-g9zjw 1/1 Laufen 0 7m40s

# Kong Gateway Loadbalancer
svclb-knime-edge-kong-proxy-bgqzc 2/2 Laufen 0 25m
svclb-knime-edge-kong-proxy-jg8fg 2/2 Laufen 0 25m

# Gestartet von einem Cronjob, um alte Protokolle zu löschen
minio-log-cleaner-cronjob-1632700800-ssqsx 0/1 Fertiggestellt 0 19m
```

Prüfung des KNIME Auf den Seiten

Der Hub Adapter ist ein wesentliches Stück in der KNIME Edge Architektur, da er kommuniziert mit KNIME Hub. Wenn z.B. Inference Deployments in der Kontrollebene auf KNIME Hub erstellt wurden werden nicht im KNIME Edge Cluster angezeigt, ein nicht richtig funktionierender Hub Adapter könnte das Problem.

Zuerst testen Sie, ob der Hub Adapter Pod richtig läuft (die Pod ID wird in Ihrem

Cluster:

```
> kubectl [-n namespace] get pod hub-adapter-697768bc8c-g9zjw
```

Es sollte den Status "Running" zurückgeben.

```
NAME READY STATUS RESTARTEN
Nabenadapter-697768bc8c-g9zjw 1/1 Laufen 0 8m2
```

Um die neuesten Protokolle des Hub Adapter Pod zu überprüfen, führen Sie:

```
> kubectl [-n namespace] logs --since = 30s hub-adapter-697768bc8c-g9zjw
```

Es gibt ein paar Dinge, die in den Logs suchen:

ANHANG Eine Zeile, die "Registrierung akzeptiert" sagt Ihnen, dass die KNIME Edge Lizenz gültig ist und korrekt eingerichtet. Nur wenn KNIME Hub die Registrierung akzeptiert, ist der Hub Adapter Informationen zu Inference Deployments von KNIME Hub ziehen.

2. Der Hub Adapter macht Anfragen an verschiedene API-Endpunkte auf KNIME Hub. Für jeden anfordern, es protokolliert den HTTP-Antwortstatus. a) 200: Die Anfrage war erfolgreich. b) 404: der Endpunkt wurde nicht gefunden. Stellen Sie sicher, dass der KNIME Edge Service auf KNIME Hub richtig laufen. Siehe [KNIME konfigurieren Business Hub](#). c) 5xx: Serverfehler. Stellen Sie sicher, dass der KNIME Edge Service auf KNIME Hub richtig läuft. Siehe [KNIME konfigurieren Business Hub](#).

Erweiterte Operationen und Fehlerbehebung

Fehler: Versäumt, Bild zu ziehen

Unterschrift des Fehlers

```
"registry.hub.knime.com/knime-edge/knime-edge-Operator:0.0.1-beta-20210802-161605-000061-4fa252d":\
rpc-Fehler: code = Unknown desc = hat das Bild nicht gezogen und ausgepackt
"registry.hub.knime.com/knime-edge/knime-edge-Operator:0.0.1-beta-20210802-161605-000061-4fa252d":\
"registry.hub.knime.com/knime-edge/knime-edge-
Betreiber:0.0.1-beta-20210802-161605-000061-4fa252d":\
unerwarteter Statuscode [manifests 0.0.1-beta-20210802-161605-000061-4fa252d]: 401
Unbefugt
```

Auslegung :

- 401 Unbefugt
- Bild URI
Registry.hub.knime.com/knime-edge/knime-edge-Operator:0.0.1-beta-20210802-

161605-000061-4fa252d

□ Web-Seiten

https://registry.hub.knime.com/harbor/projects/2/repositories	→ knime-edge	→
Strickmaschinen	→ Suche nach tag	0,0,1-beta-20210802-161605-000061-
4fa252		

ANHANG Anmelden zu KNIME Artifact Registry

Einloggen <https://registry.hub.knime.com> die Gültigkeit des Zugriffstokens zu aktualisieren.

2. Überprüfen Sie, ob die Geheimnisse korrekt konfiguriert sind

2a. Verifizieren Sie die BildPullSecrets werden konfiguriert

Suchen Sie in der Bereitstellung von Bild PullSecrets; überprüfen, ob Regcred ist anwesend:

```
% kubectl [-n] erhalten Bereitstellungen knime-edge-knime-edge-operator -o json | jq
'.spec.template.spec.imagePullSecrets'
(
  {
    "Name": "regcred"
  }
)
```

2b. Verifizieren Sie das Geheimnis im richtigen Namensraum

```
% kubectl [-n] get secret regcred
NAME TYPE DATEN
Regcred kubernetes.io/dockerconfigjson 1 50m
```

2c. Überprüfen Sie den Wert des Geheimnisses

Schlussfolgerung

Warnung : Ausführung des Befehls unten druckt den geheimen Wert zu stdout

Prüfung Geheime Werte:

```
% kubectl [-n ] get secret regcred -o json | jq -r
'.data.".dockerconfigjson" | base64 -d | jq
{
  "Authen": {\cHFFFF}
  "Registry.hub.knime.com": {\cHFFFF}
  "username": "",
  "Passwort": "",
  "email": "",
  "Auth": ""
}
}
}
```

(Erweitert) Prüfung Geheime Werte ohne Druck auf Standard out:

```
% cd
% diff <(kubectl [-n ] get secret regcred -o json | jq -r
'.data.".dockerconfigjson" | base64 -d | jq -r '.auths[].password') <(catcat
./harbor_secret); Echo $?
()
```

Fehler: Unable to Access Host

Mögliche Ursachen:

- Misconfigured Ingress, d.h. Kong wurde nicht aktiviert oder Traefik wurde aktiviert

Fehler Signatur: HTTP - Leere Antwort vom Server

```
% curl -L http://localhost:8081/
curl: (52) Leere Antwort vom Server
```

Zur Referenz würde ein vollständig falsch konfigurierter Cluster zurückkehren:

Curl: (7)

Nicht angeschlossen an localhost port 8081: Verbindung verweigert

ANHANG Überprüfen Sie die Kubernetes Cluster Networking-Einstellungen

```
kubectl [-n ] erhalten ingress
kubectl [-n ] Dienste erhalten
kubectl [-n ] erhalten Endpunkte
```

2. Überprüfen Sie die Kong Ingress Controller-Einstellungen

Richtige Ingress-Konfiguration erfordert eine Helm-Installation mit

kong.enabled: wahr

Überprüfen Sie die folgenden in der Werte.yaml verwendet, um den Cluster (oder erkunden Kubernetes Konfiguration, um festzustellen, ob der Wert eingestellt wurde:

```
...
englisch:
wahr
```

Alternativ können Sie das von Helm verwendete Manifest bei Cluster-Installationszeit überprüfen:

```
helm status -o yaml [-n ] knime-edge | weniger
```

Entschließung :

Falls erforderlich, auf wahr mit:

```
helm upgrade [-n ] knime-edge knime-edge/knime-edge-operator --version
--Reuse-Werte --set kong.enabled=true
```

Aktualisierung eines bestehenden KNIME Randdiagramm

Um eine neue Chart-Version zu aktualisieren, folgen Sie den Anweisungen, um eine aktualisierte Wertedatei zu erhalten und dann laufen:

```
helm upgrade [-n ] knime-edge knime-edge/knime-edge-operator --version
-f Kantenwerte. Yaml
```

Wenn Fehler auftreten, beziehen Sie sich auf die erweiterte Fehlersuche in dieser Anleitung.

Fehler: Helm Upgrade ist wegen Erforderlichen Wertes ungültig

Unterschrift: INSTALLATION FAILED: EdgeDeployment.edge.knime.com "edge" ist

Invalid: ... Erforderlicher Wert

```
% helm upgrade [-n
-f Kantenwerte. Yaml
...
Fehler: INSTALLATION FAILED: EdgeDeployment.edge.knime.com "edge" ist ungültig: \
[spec.knimeServerDeleteLogRequestWF: Erforderlicher Wert, \
spec.knimeServerDeployWF: Erforderlicher Wert, \
spec.knimeServerGetLogRequestsWF: Erforderlicher Wert, \
spec.knimeServerPushMetricsWF: Erforderlicher Wert, \
spec.knimeServerRegisterWF: Erforderlicher Wert, \
spec.knimeServerUploadLogFileWF: Erforderlicher Wert]
```

Auslegung : Dies sind alle Schlüssel, die von den derzeit installierten CRDs benötigt werden, die vom Betreiber genutzt. Die folgenden Fragen untersuchen mögliche Ursachen.

ANHANG Sind diese Werte in dieser Version neu? Schauen Sie sich das Diagramm an

Es ist möglich, dass eine neuere Version aktualisierte Werte haben kann;

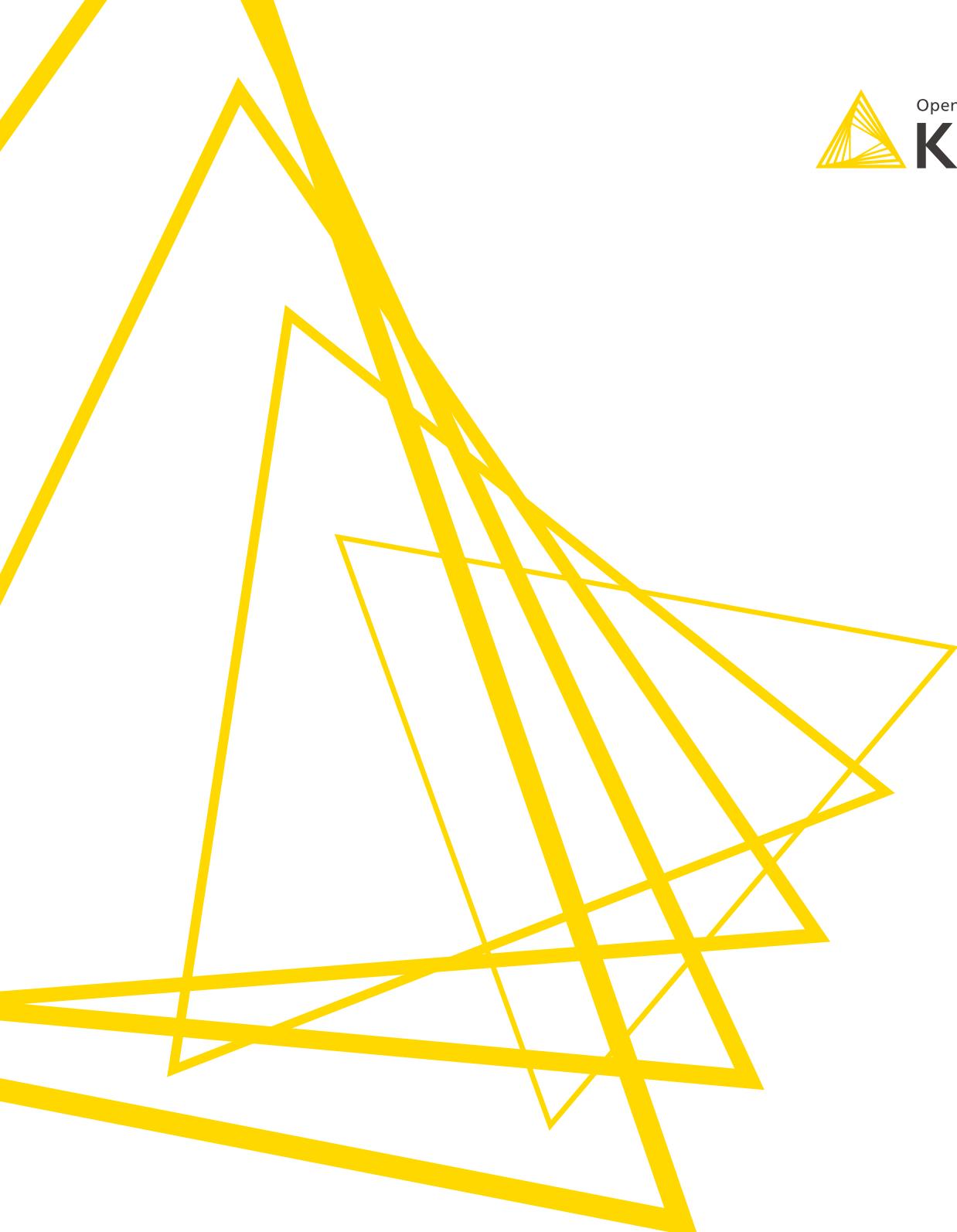
Schritte, um eine neue Kopie der `Werte.yaml` Datei und Suche nach den genannten Werten.

2. Sind diese Werte nicht im Diagramm für diese Version vorhanden? Löschen der CRDs

Der folgende Befehl wird alle KNIME Edge CRDs entfernen:

```
kubectl delete customresourcedefinition $(kubectl erhalten customresourcedefinition
-o=jsonpath='{.items[?(@.spec.group == "edge.knime.com")].metadata}'')
```

Erläuterung: Helm bewusst [wird nicht modifiziert](#), also wenn sich die erforderlichen Werte ändern zwischen Versionen, Helm kann das Update nicht automatisch verwalten.



KNIME AG
Talacker 50
8001 Zürich, Schweiz
www.knime.com
Info@knime.com