Open for Innovation
KNIME

KNIME Große Datenerweiterungen Benutzer Leitfaden

KNIME AG, Zürich, Schweiz

Version 5.7 (letzte Aktualisierung auf)

Inhaltsverzeichnis

<a href="#page2" style="color: #000000; text-decoration: underline;">Überblick . . . . . <a href="#
<a href="#page2" style="color: #000000; text-decoration: underline;">Installation . . . . . <a href="#
<a href="#page3" style="color: #000000; text-decoration: underline;">. . . . . <a href="#
<a href="#page4" style="color: #000000; text-decoration: underline;">Hive Connector . . . <a href="#
<a href="#page5" style="color: #000000; text-decoration: underline;">Impala Connector . . <a href="#
<a href="#page7" style="color: #000000; text-decoration: underline;">Massendaten laden <a href="#
<a href="#page8" style="color: #000000; text-decoration: underline;">. . . . . <a href="#
<a href="#page9" style="color: #000000; text-decoration: underline;">Bevorzugte . . . . . <a href="#
<a href="#page10" style="color: #000000; text-decoration: underline;">Spark . . . . . <a href="#
<a href="#page10" style="color: #000000; text-decoration: underline;">Erstellen Sie Spark-Context <a href="#
<a href="#page15" style="color: #000000; text-decoration: underline;">Zerstören Spark Context <a href="#
<a href="#page15" style="color: #000000; text-decoration: underline;">Erstellen Sie Datarikes E <a href="#
<a href="#page16" style="color: #000000; text-decoration: underline;">Erstellen Sie H2O-Sparkling <a href="#
<a href="#page17" style="color: #000000; text-decoration: underline;">Proxy-Einstellungen <a href="#
<a href="#page17" style="color: #000000; text-decoration: underline;">Beispielworkflow <a href="#

Überblick

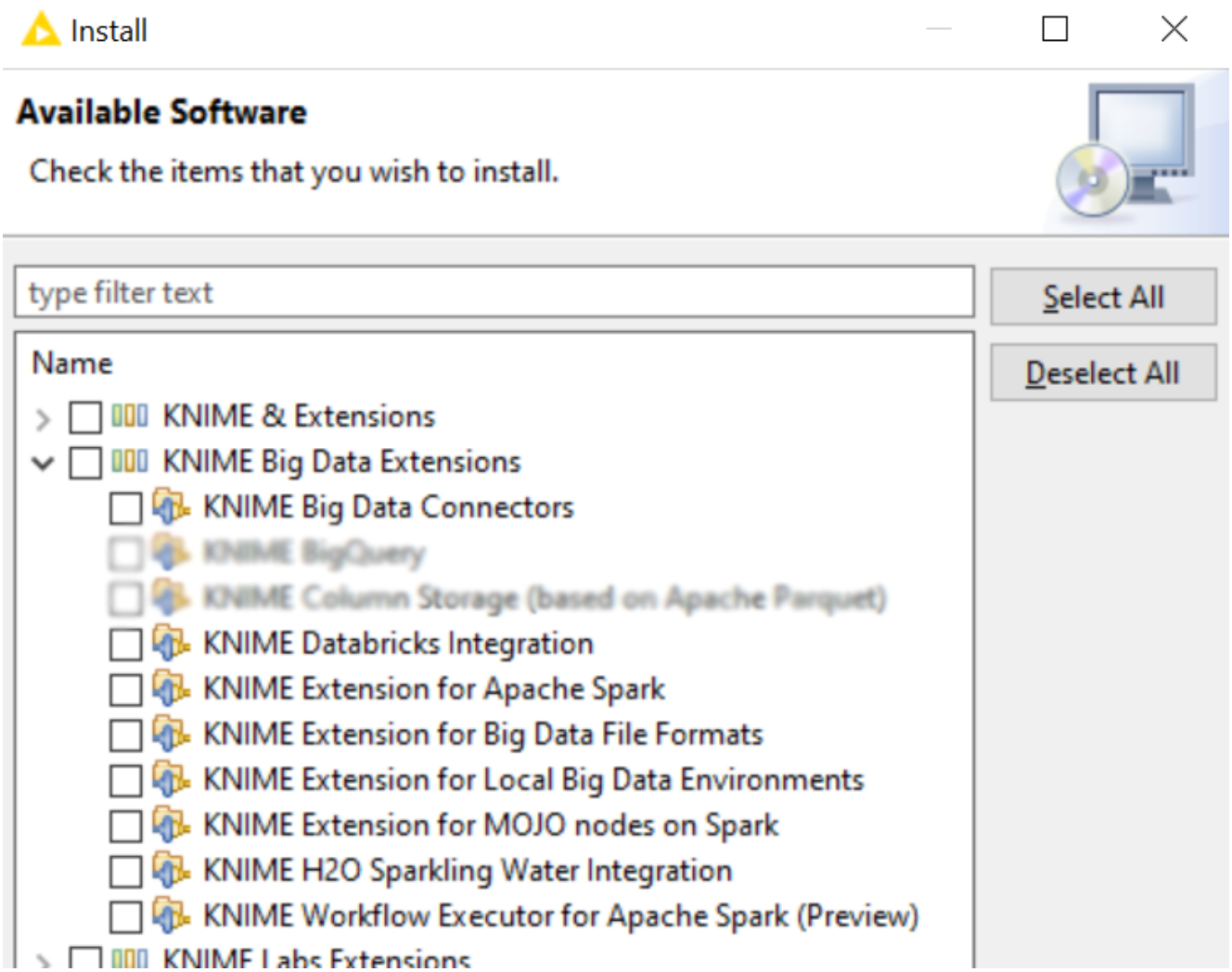
KNIME Big Data Extensions integrieren Apache Spark und das Apache Hadoop Ökosystem mit KNIME Analytics Platform.

Dieser Leitfaden richtet sich an Nutzer der KNIME Analytics Platform, die Workflows aufbauen möchten, die müssen große Datenmengen in einer großen Datenumgebung zugreifen, verarbeiten und analysieren.

Beachten Sie, dass zusätzliche Installations- und Konfigurationsschritte in Ihren Big Data erforderlich sein können Umwelt. Bitte konsultieren Sie die Big Data Extensions Admin Guide für Details.

Installation

Navigieren Datei → KNIME installieren Erweiterungen und öffnen KNIME Große Datenerweiterungen Kategorie. Überprüfen Sie die Boxen dieser Erweiterungen, die Sie installieren möchten.



KNIME Große Daten Erweiterungen sind eine Reihe von mehreren Erweiterungen, die aufeinander aufbauen:

- KNIME Big Data Connectors Bereitstellung von Verbindungsknoten zum Lesen/ Schreiben von Dateien in HDFS und Hive und Impala mit SQL abfragen.
- KNIME Erweiterung für Big Data Datei Formate erlaubt es, beliebte Dateiformate zu lesen/schreiben wie Parkett und ORC.
- KNIME Erweiterung für Apache Spark über 60 Knoten für den Datenzugriff und wrangling, sowie prädiktive Analytik in Spark. Die folgenden Erweiterungen ergänzen sogar mehr Funktionalität rund um Spark:
  - KNIME Integration von Databricks integriert Databricks mit KNIME. Bitte beachten Sie die KNIME Databricks Integration Benutzerhandbuch .
  - KNIME Erweiterung für lokale Big Data Umgebungen einen Knoten zur Erstellung eines komplett lokale Big Data Umgebung mit Spark und Hive, ohne zusätzliche Konfiguration oder Softwareinstallation.
  - KNIME Erweiterung für MOJO-Knoten auf Spark stellt Knoten bereit, mit H2O MOJOs in Spark.
  - KNIME H2O Sparking Water Integration integriert die KNIME H2O-Knoten mit Funken Sie H2O-Modelle über Daten in Spark zu lernen.
  - KNIME Workflow-Executor für Apache Spark (Voransicht) erlaubt nicht-Spark auszuführen KNIME-Knoten auf Apache Spark.

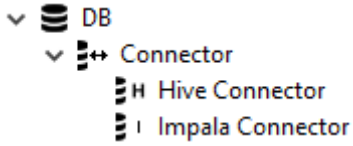
Sobald Sie die Erweiterung(en) installiert haben, starten Sie die KNIME Analytics Platform neu.

Wenn Sie keinen direkten Internetzugang haben, können Sie auch die Erweiterungen von eine Reißverschluss-Update-Seite. Folgen Sie den in Lokale Update-Sites hinzufügen .

Die Spark Erweiterungen unterstützen nur Apache Spark Versionen 3.4 - 3.5. Unterstützung für ältere Spark-Versionen (Spark 1.x, 2.x und 3.0 - 3.3) wurden getrennt verschoben Erweiterungen. Um sie zu installieren, navigieren, Datei → KNIME installieren Erweiterungen und Das ist die Gruppenposten nach Kategorie Box. Suchen Sie jetzt nach "Spark". Das Ergebnis Verlängerungen, die mit (Legali) Unterstützung für ältere Spark Versionen.

Hive und Impala

Die KNIME Big Data Connectors Erweiterung bietet Knoten zu Hive und Impala verbinden.



Hive Connector

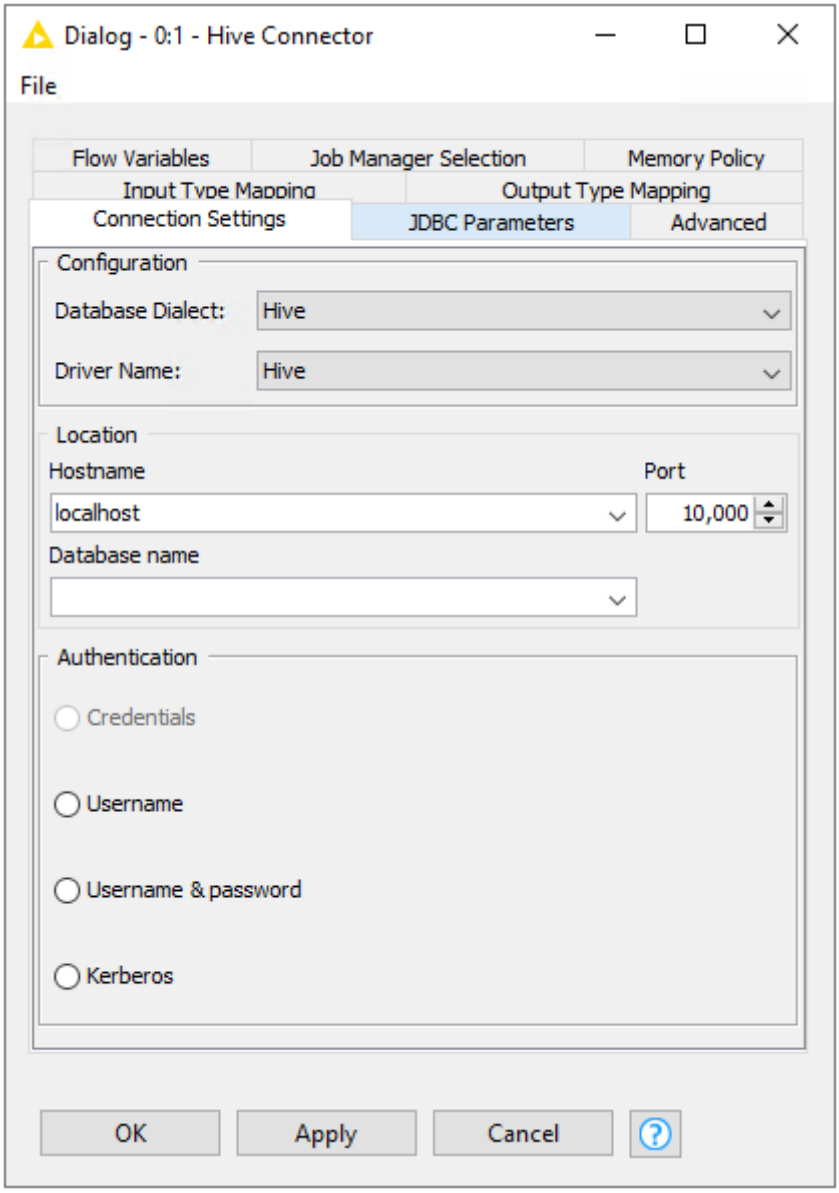


Abbildung 1. Hive Connector Konfiguration Dialog

Die Hive Connector node erstellt eine Verbindung zu Hive via JDBC. Sie müssen die folgende Angaben:

- Hostname (oder IP-Adresse) des Servers
- der Hafen
- einen Datenbanknamen.
- Eine Authentifizierungsmethode (wie von Hive gefordert):
  - Angaben wo Benutzername und Passwort über den Anmeldestrom geliefert werden Variable (siehe Anmeldeinformationen Node).
  - Benutzername wo der Benutzername im Dialog angezeigt wird.

- Benutzername und Passwort wo Benutzername und Passwort im Dialog geliefert werden.
- Kerberos, wo die Authentifizierung über Kerberos erfolgt.

Bei Verwendung von Kerberos Authentifizierung: Zusätzliche Parameter müssen sein in der Registerkarte JDBC-Parameter angegeben. Die genauen Parameter hängen von der JDBC Treiber ausgewählt in der Fahrername Einstellung.

Der eingebaute Treiber mit dem Namen "Hive" benötigt folgende Parameter für Kerberos:

- kerberosAuthTyp = vonSubject
- :Hive/@ , wo
  - ist der vollqualifizierte Hostname des Hive Service
  - ist das Kerberos Reich des Hive Service

Eigentum Hive Treiber (z.B. von Cloudera) benötigen folgende Parameter:

- AuthMech = 1
- Der Name des Unternehmens
- KrbHostFQDN = , wo ist der vollqualifizierte Hostname vom Hive-Service
- KrbRealm = , wo ist das Kerberos Reich der Hive Service

Bitte beachten Sie, dass die proprietären Fahrer wie in Registrieren Sie Ihre eigenen JDBC Treiber (KNIME Database Extension Guide) .

Impala Connector

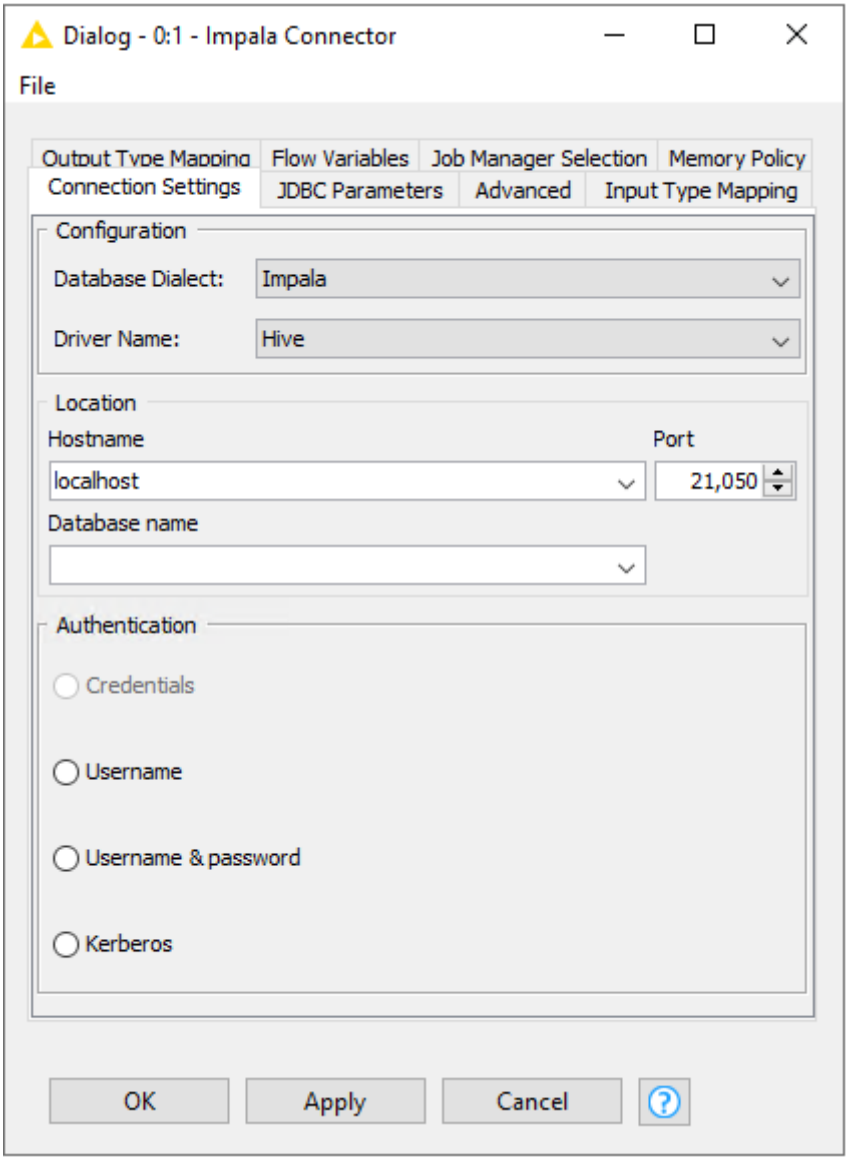


Abbildung 2. Impala Connector Konfiguration Dialog

Die Impala Connector node erstellt eine Verbindung mit Impala über JDBC. Sie müssen bereitstellen die folgenden Angaben:

- Hostname (oder IP-Adresse) des Impala-Dienstes
- der Hafen
- einen Datenbanknamen.
- Eine Authentifizierungsmethode (wie von Impala gefordert):
  - Angaben wo Benutzername und Passwort über den Anmeldestrom geliefert werden Variable (siehe Anmeldeinformationen Node).
  - Benutzername wo der Benutzername im Dialog angezeigt wird.
  - Benutzername und Passwort wo Benutzername und Passwort im Dialog geliefert werden.
  - Kerberos, wo die Authentifizierung über Kerberos erfolgt.

Bei Verwendung von Kerberos Authentifizierung: Zusätzliche Parameter müssen sein in der Registerkarte JDBC-Parameter angegeben. Die genauen Parameter hängen von der JDBC Treiber ausgewählt in der Fahrername Einstellung.

Der eingebaute Treiber mit dem Namen "Hive" benötigt folgende Parameter für Kerberos:

- kerberosAuthTyp = vonSubject
- = Imala/@ , wo
  - ist der vollqualifizierte Hostname des Hive Service
  - ist das Kerberos Reich des Hive Service

Eigentum Hive Treiber (z.B. von Cloudera) benötigen folgende Parameter:

- AuthMech = 1
- Der Name der Person
- KrbHostFQDN = , wo ist der vollqualifizierte Hostname vom Hive-Service
- KrbRealm = , wo ist das Kerberos Reich der Hive Service

Bulk-Datenbelastung

Die DB Loader node unterstützt das Bulkload von Daten von KNIME Analytics Platform in a Hive oder Impala Tisch. Beachten Sie, dass die Datenbanktabelle vor der Ausführung der DB Loader Knoten. Das folgende Beispiel verwendet DB Table Creator Node, um die Tabelle vor dem Laden zu erstellen die Daten in die Tabelle, dies ist jedoch nicht erforderlich, wenn die Tabelle bereits existiert.

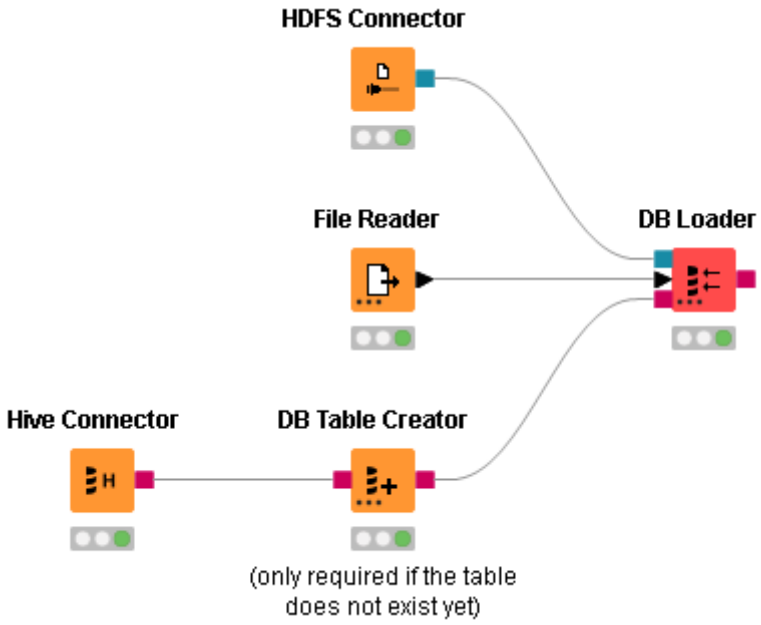
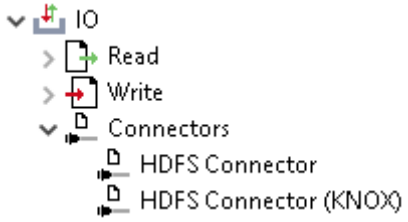


Abbildung 3. Workflow, der eine Hive-Tabelle erstellt und dann Daten in sie lädt.

HDFS

Die KNIME Big Data Connectors Erweiterung bietet mehrere Knoten zur Verbindung mit HDFS



- HDFS Connector kombiniert verschiedene HDFS kompatible Protokolle in einem Knoten:
  - HDFS kommuniziert direkt mit HDFS während der Datenübertragung, d.h. dem NameNode und alle DataNodes. Dieser Knoten erfordert **direkte Netzwerkverbindung (keine Proxies, nein Firewall)** zwischen KNIME Analytics Platform/KNIME Server und Hadoop Cluster, was oft nicht der Fall ist. Regelmäßig eingeschränkte Netzwerkverbindung führt zu Timout-Fehlern während der Datenübertragung.
  - WebHDFS Verwendung von HTTP zur direkten Verbindung mit HDFS, d.h. dem NameNode und allen DataNodes. Es ist möglich, über einen HTTP-Proxy zu verbinden, aber immer noch alle Cluster Knoten müssen durch den Proxy erreichbar sein.
  - WebHDFS mit SSL verwendet HTTPS (SSL verschlüsselt), um direkt mit HDFS zu verbinden.
  - HttpFS (**empfohlen**) verwendet HTTP zur Verbindung mit einem httpFS-Dienst in einem Cluster frontend/edge node. Der httpFS-Dienst dient als Vermittler zwischen internes Clusternetzwerk und KNIME Analytics Platform/KNIME Server.

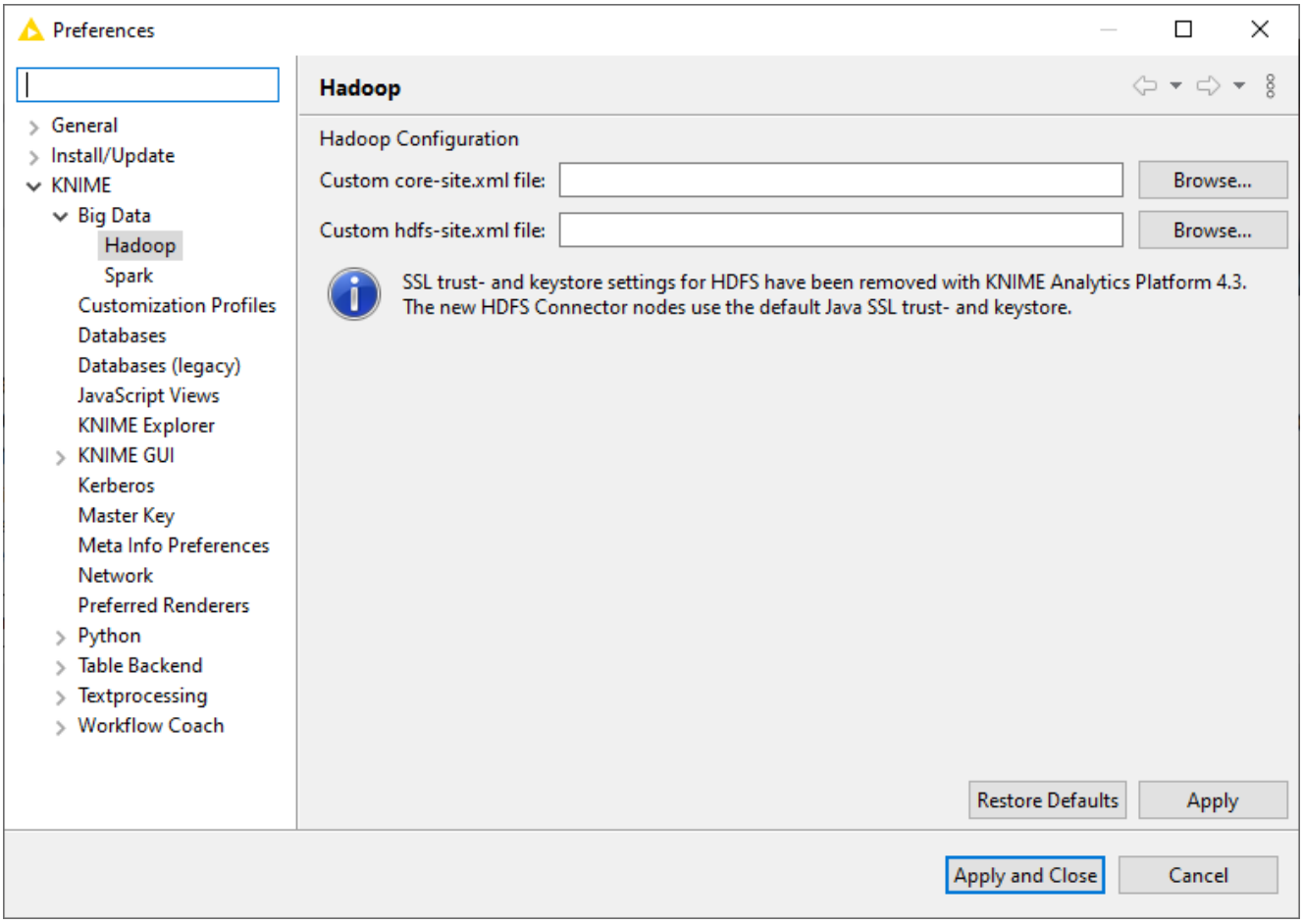
- HttpFS mit SSL (**empfohlen**) verwendet HTTPS (SSL verschlüsselt) um eine Verbindung zu a httpFS-Service auf einem Cluster-Frontend/edge-Knoten.

- HDFS Connector (KNOX) Verwendung von HTTP zur Verbindung mit einem Apache KNOX-Dienst in einem Cluster frontend/edge node. Der KNOX-Service dient als Vermittler zwischen dem internen Clusternetzwerk und KNIME Analytics Platform/KNIME Server.

Die oben genannten Verbindungsknoten können zusammen mit den KNIME File Handling Nodes bis Dateien hochladen, herunterladen oder auflisten und andere Dateioperationen ausführen.

Bitte konsultieren Sie die Beispiel-Workflows, die auf KNIME Hub im KNIME Beispiele Raum .

Vorlieben



Die Hadoop-Präferenzen erlauben typische Hadoop-Konfigurationsdateien (Core-Site.xml und hdfs-site.xml) falls erforderlich.

Weitere Informationen zur SSL-Verschlüsselung mit Firmen- oder selbstsignierten SSL-Zertifikaten können in der KNIME Leitfaden für die Verwaltung von Servern .

Spark

KNIME Erweiterung für Apache Spark bietet eine Reihe von über 60 Knoten zu erstellen und auszuführen Apache Spark Anwendungen.

Abbildung 4. Alle Funkknoten

Der erste Schritt in jedem Spark-Workflow besteht darin, einen Spark-Kontext zu erstellen, der die Verbindung zu einem Spark Cluster. Der Funkkontext behält sich auch Ressourcen vor (CPU-Kerne und Speicher) in Ihrem Cluster ausschließlich von Ihrem Workflow verwendet werden. Daher ein Spark-Kontext zu Beginn eines Workflows erstellt und am Ende zerstört werden soll, um zu lösen die Ressourcen.

Es gibt mehrere Knoten, um einen Spark-Kontext zu erstellen:

- <a href="#page10" style="color: #ff6600; text-decoration: underline;">Spark Context (Livy) erstelle (empfohlen)
- Lokale Big Data Environment erstellen (Anforderungen KNIME Erweiterung für lokale Big Data Umwelt)
- <a href="#page15" style="color: #ff6600; text-decoration: underline;">Databricks Umwelt erstellen< (Anforderungen KNIME Integration von Databricks)

Spark Context (Livy) erstellen

Die Spark Context (Livy) erstellen Knoten verbindet sich mit einem Apokalypse Server zum Erstellen eines neuen Funkkontext.

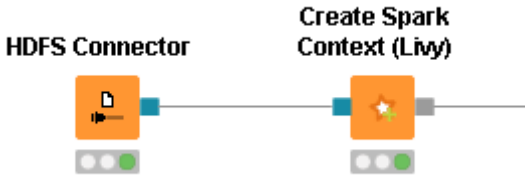


Abbildung 5. Erstellen Spark Context (Livy) Node

Anforderungen

- **Apache Livy Service** Livy muss als Service in Ihrem Cluster installiert werden. Bitte! berät. Apache Livy Setup Abschnitt für weitere Details.
- **Netzwerkverbindung:** Der Knoten initiiert eine HTTP(S)-Verbindung zu Livy (Standardport TCP/8998. Derzeit sind nur HTTP(S)-Proxies, die keine Authentifizierung erfordern, unterstützt.
- **Authentifizierung** : Wenn Livy Kerberos Authentifizierung erfordert, dann KNIME Analytics Platform muss entsprechend eingerichtet werden.
- **Remote-Dateisystem-Verbindung** : Der Knoten erfordert Zugriff auf ein Remote-Dateisystem, um temporäre Dateien zwischen der KNIME Analytics Platform und dem Spark-Kontext austauschen (auf den Cluster laufen). Unterstützte Dateisysteme Steckverbinder sind:
  - HDFS und HDFS (KNOX). Beachten Sie, dass der Knoten auf das Remote-Dateisystem zugreifen muss mit dem gleichen Benutzer wie der Spark-Kontext. Bei der Authentisierung mit Kerberos gegen HDFS/WebHDFS/HttpFS und Livy, dann wird der gleiche Benutzer verwendet. Andernfalls muss dies manuell gewährleistet werden.
  - Amazon S3, Azure Blob Store und Google Cloud Storage, die empfohlen wird bei Verwendung von Spark auf Amazon EMR/Azure HDInsight/Google Datapoc. Anmerkung: für diese Dateisysteme muss in der **Erweiterte** Registerkarte der Spark Context (Livy) erstellen Knoten.

Node Dialog

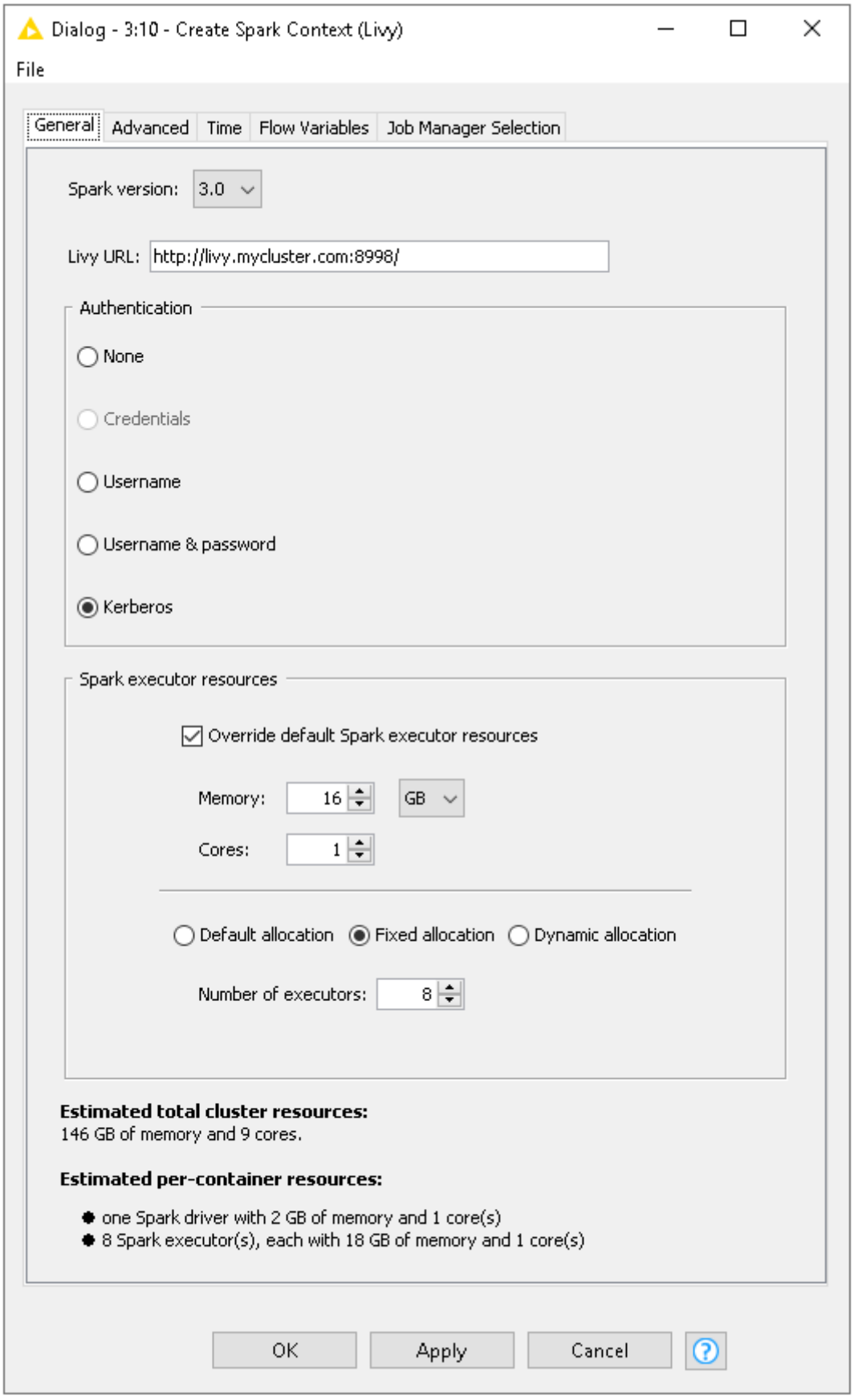


Abbildung 6. Spark Context erstellen (Livy): Register für allgemeine Einstellungen

Der Knotendialog hat zwei Registerkarten. Die erste Registerkarte bietet die am häufigsten verwendeten Einstellungen, wenn

Arbeiten mit Spark:

ANHANG **Funkversion:** Bitte wählen Sie die Spark-Version des Hadoop-Clusters aus, die Sie sind mit.

2. **Livy URL:** Die URL von Livy einschließlich Protokoll und Port z. http://localhost:8998/ .

3. **Authentication:** Wie man gegen Livy authentifiziert. Unterstützte Mechanismen sind Kerberos und Keine.

L 347 vom 20.12.2013, S. 1). **Spark Executor Ressourcen:** Stellt die Ressourcen fest, die für die Spark-Executors zu verlangen sind. wenn aktiviert, können Sie die Speichermenge und die Anzahl der Kerne für jede Executor. Darüber hinaus können Sie die Spark-Executor-Zuordnungsstrategie angeben.

5. **Geschätzte Ressourcen:** Eine Schätzung der Ressourcen, die in Ihrem Cluster zugewiesen werden durch den Spark-Kontext. Die Berechnung verwendet Standardeinstellungen für Speicherüberköpfe usw. und ist somit nur eine Schätzung. Die genauen Ressourcen können je nach Ihrer spezifische Clustereinstellungen.

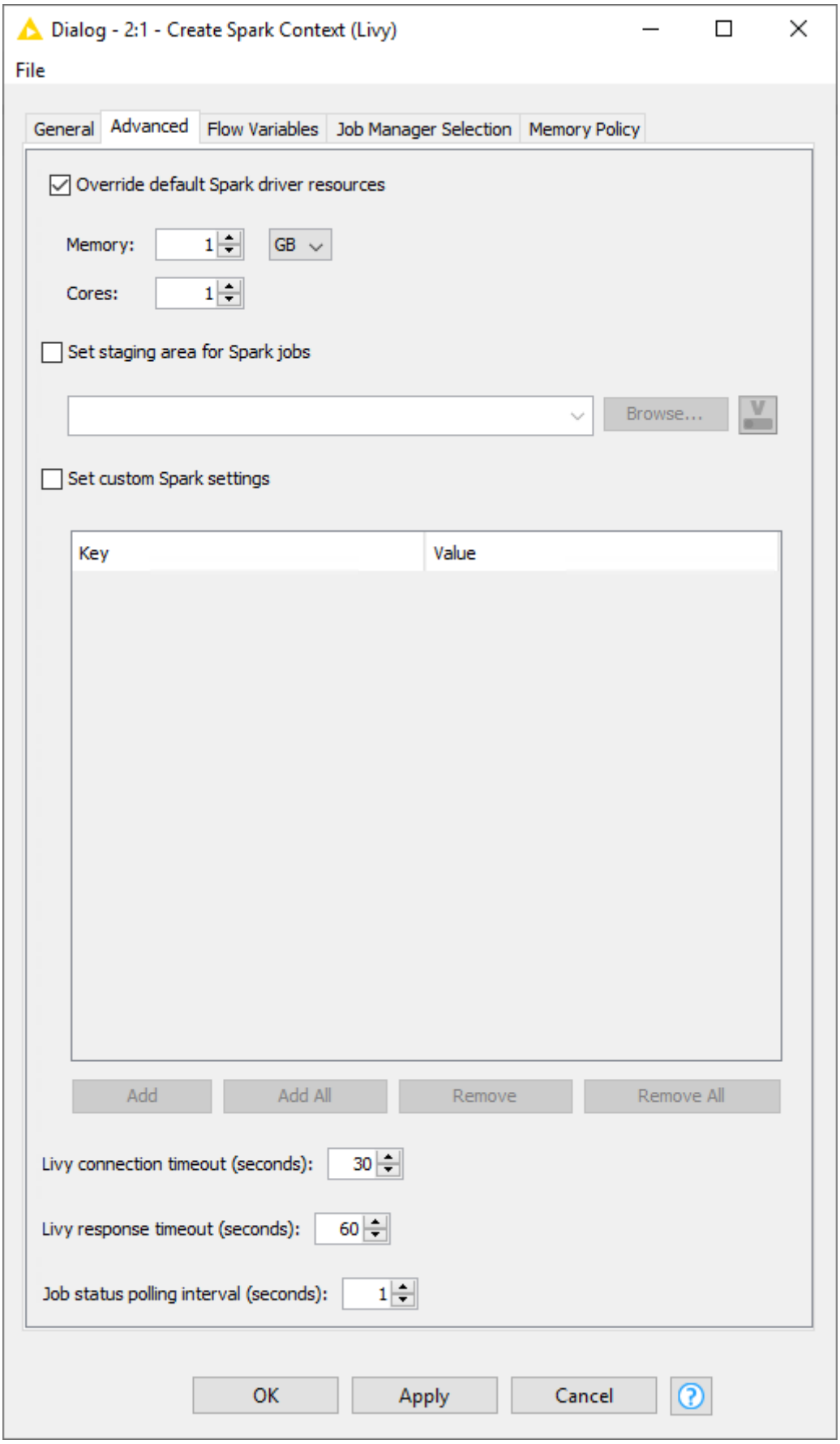


Abbildung 7. Spark Context erstellen (Livy): Register für erweiterte Einstellungen

Die zweite Registerkarte bietet die erweiterten Einstellungen, die manchmal nützlich sind, wenn Sie arbeiten

mit Spark:

ANH ANG **Standardüberschreitung Funk-Treiberressourcen:** Wenn aktiviert, können Sie den Betrag angeben Speicher und Anzahl der für den Spark-Treiberprozess zuzuordnenden Kerne.

2. **Stellplatz für Spark Jobs:** Wenn aktiviert, können Sie ein Verzeichnis im angeschlossenen Remote-Dateisystem, das verwendet wird, um temporäre Dateien zwischen KNIME und dem Funkkontext. Wenn kein Verzeichnis gesetzt wird, wird ein Standardverzeichnis gewählt, z.B. das HDFS Benutzer-Home-Verzeichnis. Wenn das Remote-Dateisystem Amazon S3 oder Azure ist Blob Store, dann muss ein Inserate-Verzeichnis bereitgestellt werden.
3. **Set benutzerdefinierte Funkeinstellungen:** Wenn aktiviert, können Sie zusätzliche Spark-Einstellungen angeben. A tooltip ist für die Schlüssel bereitgestellt, wenn verfügbar. Weitere Informationen zum Spark Einstellungen beziehen sich auf die Spark-Dokumentation.

Destroy Spark Context Node

Sobald Sie fertig sind Spark-Job, sollten Sie den erstellten Kontext zerstören, um die Ressourcen, die Ihr Spark Context im Cluster zugewiesen hat. So können Sie die **Zerstörung Spark Kontext** Knoten.

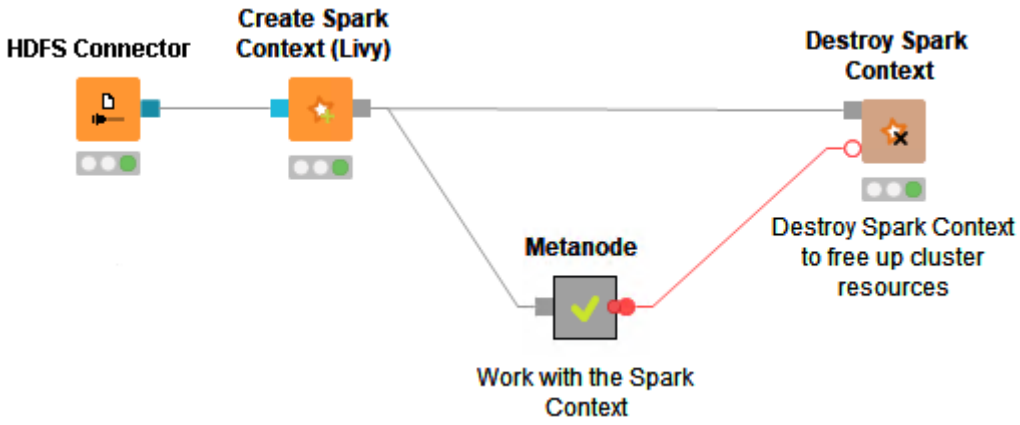
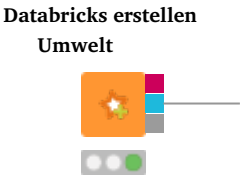


Abbildung 8. Wie verwendet man den Destroy Spark Context-Knoten

Databricks Umwelt erstellen

Dieser Knoten verbindet einen Databricks-Cluster innerhalb der KNIME Analytics Platform.



Für eine detailliertere Anleitung zur Konfiguration dieses Knotens und zur Erstellung eines Databricks Cluster bitte auf die KNIME Databricks Integration Benutzerhandbuch .

H2O Sparking Water Context erstellen

Unterstützung für H2O Sparking Water wird ausgeschaltet, und keine Unterstützung wird hinzugefügt für kommende Spark-Versionen. Ab jetzt ist Sparkling Water nicht unterstützt auf Clustern mit Spark 3.4 oder höher, z. Databricks. Nur die "Create Local Big Data Environment" node wird derzeit noch unterstützt, Diese Unterstützung wird jedoch auch in naher Zukunft entfernt werden.

Die H2O Sparking Water Context erstellen node erstellt einen H2O-Kontext innerhalb eines bestimmten Sparks inszeniert. Damit können Sie die KNIME H2O-Knoten auf Daten im Spark verwenden. Dieser Knoten erfordert KNIME H2O Sparking Water Integration . Weitere Informationen zur Installation der Integration können werden gefunden oben .

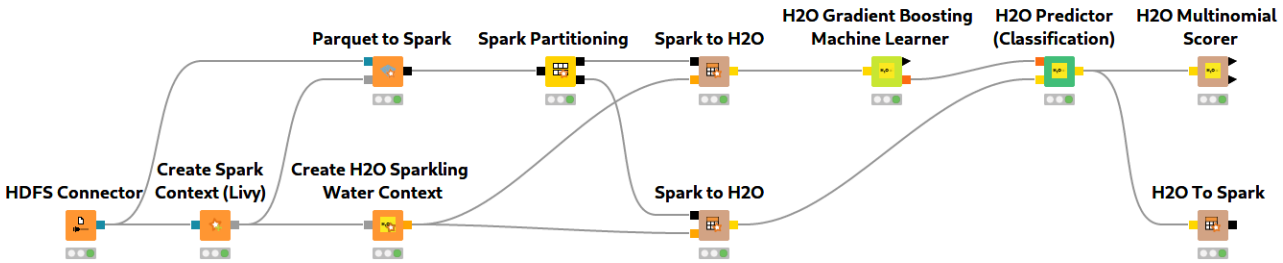


Abbildung 9. Erstellen H2O Sparkling Water Context Node

KNIME unterstützt nur einen begrenzten Satz von Spark- und H2O-Versionen. Der nächste Abschnitt enthält eine Tabelle mit allen kompatiblen Versionen. Öffnen Datei > Vorlieben > KNIME > H2O-3 und wählen Sie eine H2O-Version, die mit der Spark-Version arbeitet, die Sie sind Verwendung.

Die H2O Sparking Water Context erstellen node detektiert, wenn die benötigten H2O-Bibliotheken bereits vorhanden sind auf dem Cluster. Wenn nicht, es versucht, sie hochzuladen, was bedeutet, laden ~20MB von Bibliotheken jedes Mal, wenn der Knoten ausgeführt wird. Für eine bessere Leistung können die Bibliotheken direkt installiert von maven, wenn der Spark Kontext erstellt wird. Geben Sie dazu den richtigen Maven an Koordinaten in der Spark-Einstellung funker.jars.packages (siehe Benutzerdefinierte Spark-Einstellungen einstellen Option Hier.) Die folgende Tabelle gibt die Kombinationen von Spark- und H2O-Versionen an, die KNIME unterstützt sowie die jeweilige Sparkling Water maven-Koordinate.

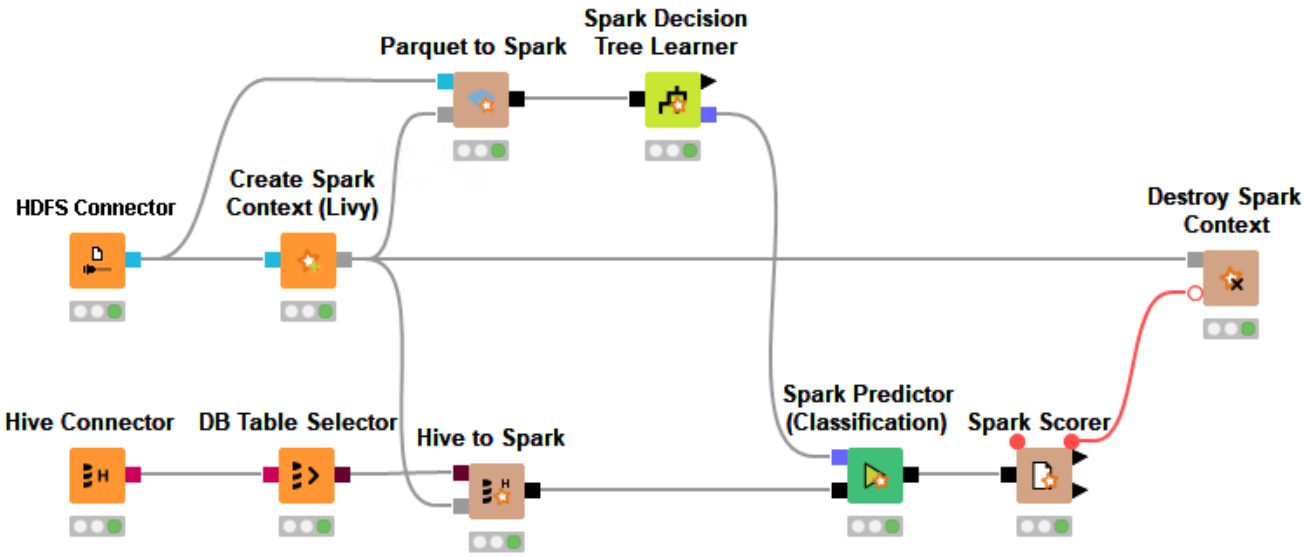
Spark Version	H2O Version	Sparking Water Maven Coordinate
3.5 (lokal groß Daten nur)	3.46.0.1	ai.h2o:sparkling-water-package_2.12:3.46.0.1-1-3,5
ANHANG	3.36.1.5	ai.h2o:sparkling-water-package_2.12:3.36.1.5-1-ANHANG

3.2.	3.36.1.2	ai.h2o:sparkling-water-package_2.12:3.36.1.2-1-3.2.
3.1.	3.32.1.2	ai.h2o:sparkling-water-package_2.12:3.32.12-1-3.1.
3	3.32.1.2	ai.h2o:sparkling-water-package_2.12:3.32.12-1-3
ANHANG	3.30.0.4	ai.h2o:sparkling-water-package_2.11:3.30.0.4-1-ANHANG
ANHANG	3.24.0.4	ai.h2o:sparkling-water-package_2.11:2.4.12
ANHANG	3.22.0.2	ai.h2o:sparkling-water-package_2.11:2.4.1
ANHANG	3.24.0.4	ai.h2o:sparkling-water-package_2.11:2.3.30
ANHANG	3.22.0.2	ai.h2o:sparkling-water-package_2.11:2.3.18
ANHANG	3.22.0.2	ai.h2o:sparkling-water-package_2.11:2.2.29

Proxy Einstellungen

Wenn Ihr Netzwerk Sie über einen Proxy mit Livy Server oder Databricks verbinden muss, öffnen Sie bitte Datei > Vorlieben > Netzwerkverbindungen . Hier können Sie die Details Ihrer HTTP/HTTPS/SOCKS-Proxies. Bitte konsultieren Sie den Beamten Eclipse Dokumentation wie man Proxies konfigurieren.

Beispiel-Workflow



Das obige Beispiel Workflow erstellt zunächst einen Spark-Kontext (Spark Context (Livy) erstellen) und dann liest Trainingsdaten aus einer in HDFS gespeicherten Parkettdatei (Parkett zum Spark) Dann trainiert es Entscheidung Baummodell (Entscheidung des Parks Tree Lernen) auf diesen Daten. Zusätzlich liest es ein Hive Tabelle mit Testdaten in Spark (Hive to Spark), verwendet das zuvor gelernte Modell zur Ausführung Vorhersagen (Spark Predictor), und bestimmt die Genauigkeit des Modells bei den Testdaten (Spark Scorer)

Weitere Beispiele finden Sie in KNIME Beispiele Raum auf KNIME Hub Sie finden eine Vielzahl von Beispiel-Workflows, die zeigen, wie man die Funkknoten.

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