

SANSA

in the Palm of your Browser

Carsten Felix Draschner - SDA Research

Overview

- get Databricks
- setup Databricks
 - get SANSA Jar
 - get sample files
 - create cluster
 - set spark enviroment
 - use SANSA jar
 - create notebook
 - use sansa modules

Databricks

- Similar concepts
 - jupyter notebooks
 - google colab
 - Scala
 - Apache Spark
 - AWS
- Advantages
 - No installation required
 - available over browser
 - native scala spark notebooks
 - no local performance needed









Databricks Registration


- Databricks FAQ
 - <https://databricks.com/de/product/faq/community-edition>
- Login
 - <https://community.cloud.databricks.com/login.html>
- Or create and use for free
 - <https://databricks.com/try-databricks>
 - 15GB Ram, 2 Core Cluster SAmple Cluster


Setup Databricks - Upload needed Data - jar

- Most recent SANSA Release Jar Available on Github Page:
 - <https://github.com/SANSA-Stack/SANSA-Stack/releases>
- Or through this link directly: 234mb fat jar
 - [sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.jar](#)


Setup Databricks - Upload needed Data - jar




Home

Workspace

Recents

Data

Clusters

Jobs

Search

**data**bricks









[Explore the Quickstart Tutorial](#)
Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

Drop files or [click to browse](#)



[Import & Explore Data](#)
Quickly import data, preview its schema, create a table, and query it in a notebook.


[Create a Blank Notebook](#)
Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

 New Notebook
 Create Table
 New Cluster
 New Job
 New MLflow Experiment
 Import Library
 Read Documentation

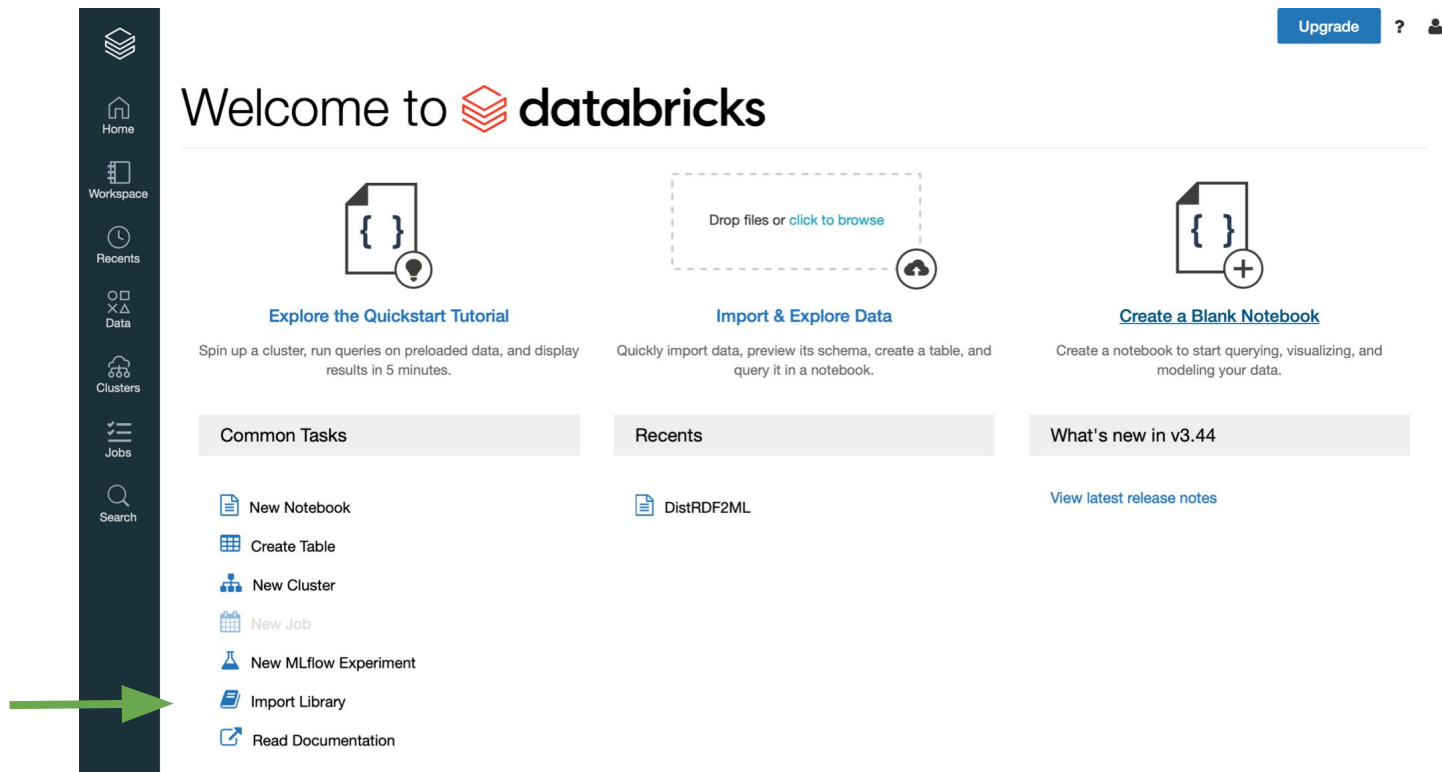
Recents

 DistRDF2ML

What's new in v3.44

[View latest release notes](#)


Setup Databricks - Upload needed Data - jar



The screenshot shows the Databricks web interface. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. A green arrow points to the Search icon. In the top right corner, there is an 'Upgrade' button and user icons. The main header says 'Welcome to databricks'. Below this are three main action cards: 'Explore the Quickstart Tutorial', 'Import & Explore Data', and 'Create a Blank Notebook'. Each card has a brief description of the action. Below these cards are three sections: 'Common Tasks' with links to 'New Notebook', 'Create Table', 'New Cluster', 'New Job', 'New MLflow Experiment', 'Import Library', and 'Read Documentation'; 'Recents' with a link to 'DistRDF2ML'; and 'What's new in v3.44' with a link to 'View latest release notes'.

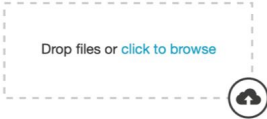
Upgrade ?

Welcome to databricks




Explore the Quickstart Tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



Import & Explore Data

Quickly import data, preview its schema, create a table, and query it in a notebook.



Create a Blank Notebook

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

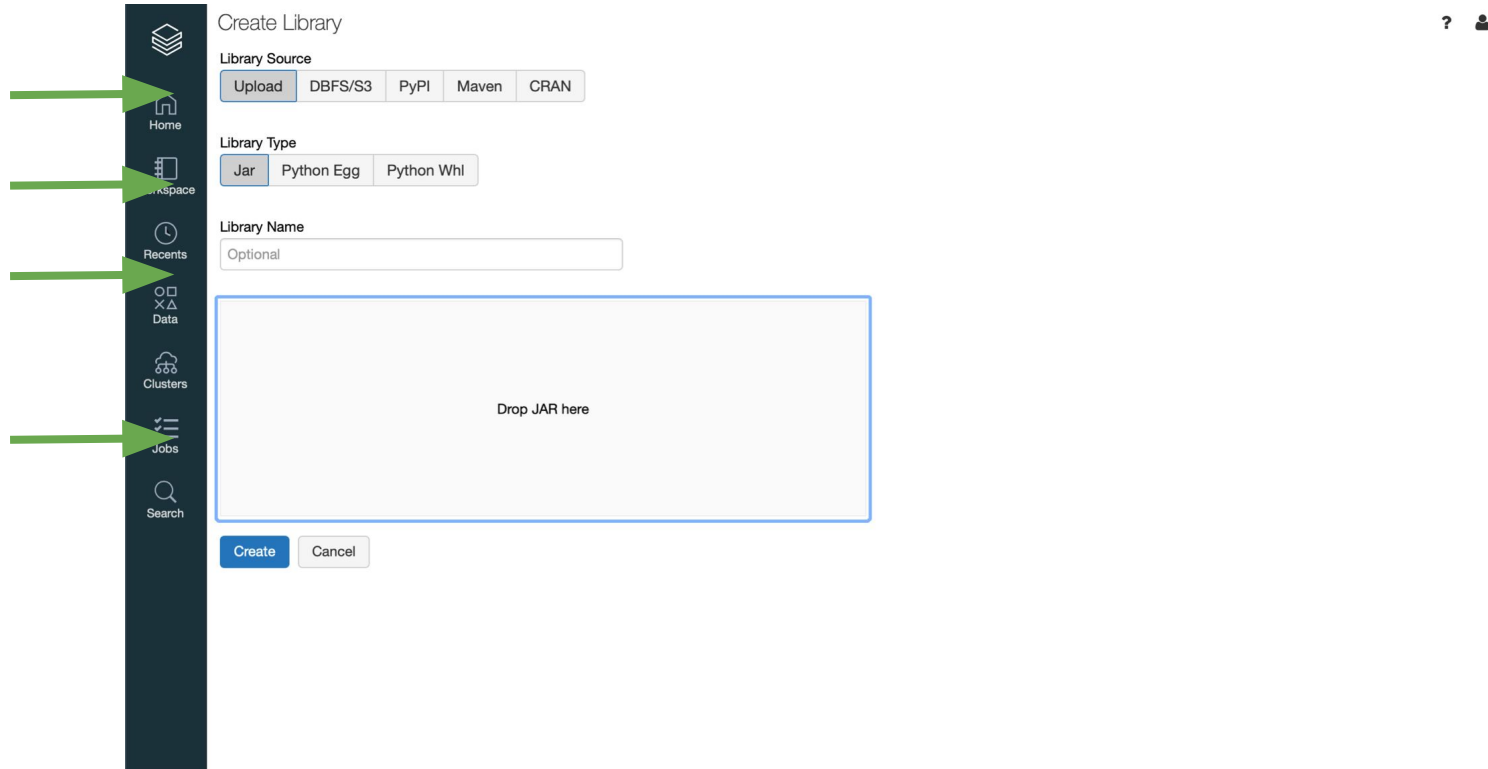
Recents

- DistRDF2ML

What's new in v3.44

[View latest release notes](#)

Setup Databricks - Upload needed Data - jar



The screenshot shows the Databricks 'Create Library' page. On the left is a dark navigation sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. Green arrows point to each of these icons. The main content area is titled 'Create Library' and includes a 'Library Source' section with buttons for 'Upload', 'DBFS/S3', 'PyPI', 'Maven', and 'CRAN'. Below this is the 'Library Type' section with buttons for 'Jar', 'Python Egg', and 'Python Whl'. The 'Library Name' section has a text input field with the placeholder 'Optional'. A large light gray box with a blue border contains the text 'Drop JAR here'. At the bottom are 'Create' and 'Cancel' buttons. In the top right corner, there are help and user icons.

Create Library

Library Source

Upload DBFS/S3 PyPI Maven CRAN

Library Type

Jar Python Egg Python Whl

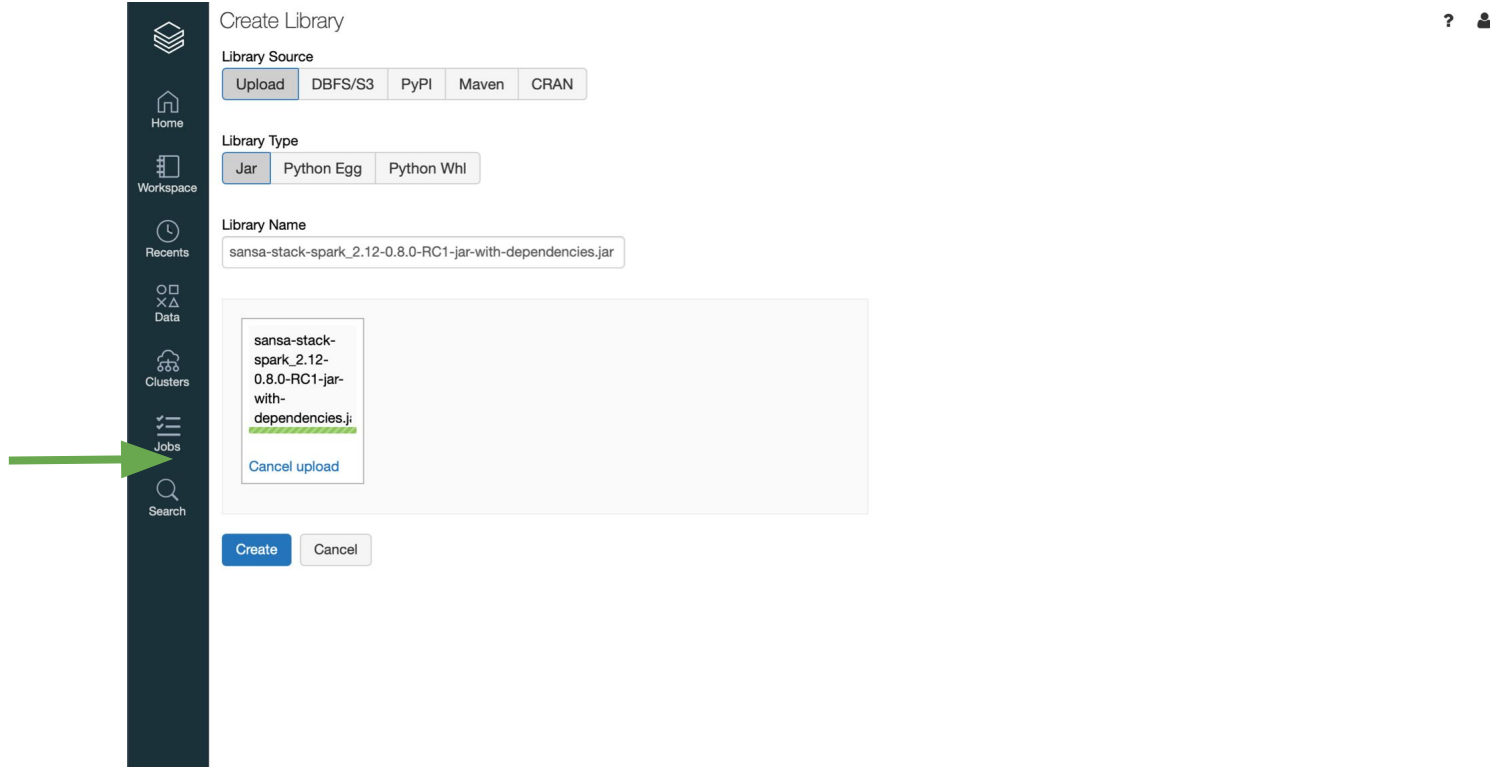
Library Name

Optional

Drop JAR here

Create Cancel

Setup Databricks - Upload needed Data - jar

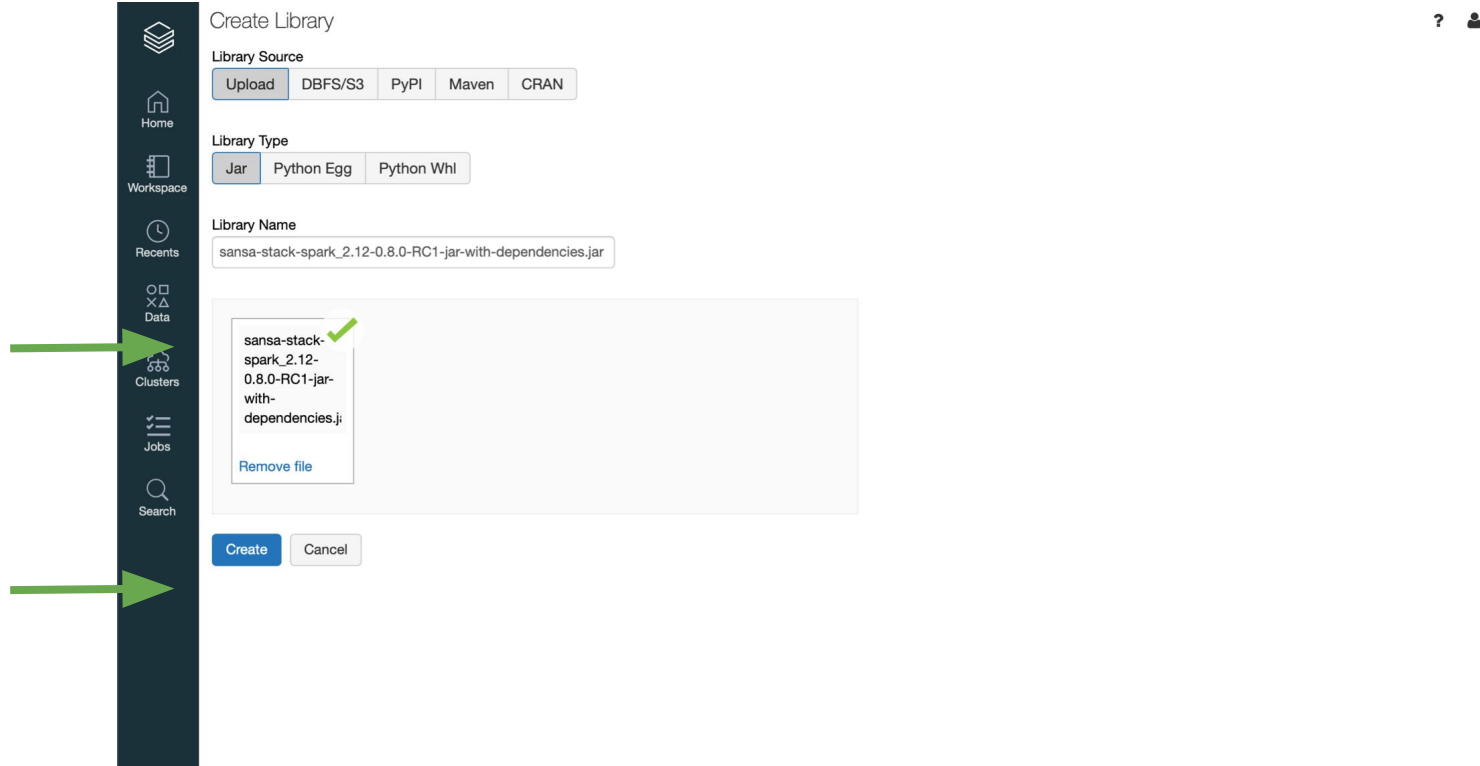


The screenshot shows the Databricks 'Create Library' page. The left sidebar contains navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs (highlighted with a green arrow), and Search. The main content area is titled 'Create Library' and includes the following sections:

- Library Source:** Buttons for Upload, DBFS/S3, PyPI, Maven, and CRAN. 'Upload' is selected.
- Library Type:** Buttons for Jar, Python Egg, and Python Whl. 'Jar' is selected.
- Library Name:** A text input field containing 'sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.jar'.

A modal dialog is open, displaying the library name 'sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.jar' and a 'Cancel upload' link. At the bottom of the page, there are 'Create' and 'Cancel' buttons.

Setup Databricks - Upload needed Data - jar



The screenshot displays the Databricks 'Create Library' page. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. Two green arrows point to the 'Data' and 'Clusters' icons. The main content area is titled 'Create Library' and includes a 'Library Source' section with 'Upload' selected, 'Library Type' section with 'Jar' selected, and a 'Library Name' field containing 'sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.jar'. Below this is a file upload area showing a preview of the selected file with a green checkmark and a 'Remove file' link. At the bottom are 'Create' and 'Cancel' buttons. In the top right corner, there are help and user icons.

Setup Databricks - Upload needed Data - data

- Sample data

- LMDB

- <http://www.cs.toronto.edu/~oktie/linkedmdb/linkedmdb-18-05-2009-dump.nt>

- Sample Data

- [E4toE5.zip](#)

Setup Databricks - Upload needed Data - data

The screenshot shows the Databricks web interface. On the left is a dark sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area has a top navigation bar with an 'Upgrade' button, a help icon, and a user profile icon. Below the navigation bar, the heading 'Welcome to databricks' is displayed. The central area features three main action cards: 'Explore the Quickstart Tutorial' (with a document icon and a lightbulb), 'Import & Explore Data' (with a dashed box icon and a cloud upload icon), and 'Create a Blank Notebook' (with a document icon and a plus sign). A green arrow points from the 'Explore the Quickstart Tutorial' card to the 'Import & Explore Data' card. Below these cards are three sections: 'Common Tasks' (listing New Notebook, Create Table, New Cluster, New Job, New MLflow Experiment, Import Library, and Read Documentation), 'Recents' (showing a file named 'DistRDF2ML'), and 'What's new in v3.44' (with a link to 'View latest release notes').

Upgrade ?

Welcome to databricks

Explore the Quickstart Tutorial
Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

Import & Explore Data
Quickly import data, preview its schema, create a table, and query it in a notebook.

Create a Blank Notebook
Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

Recents

- DistRDF2ML

What's new in v3.44

[View latest release notes](#)

Setup Databricks - Upload needed Data - data

Create New Table

Data source ?

Upload File S3 DBFS Other Data Sources Partner Integrations

DBFS Target Directory ?

/FileStore/tables/ (optional) Select

Files uploaded to DBFS are accessible by everyone who has access to this workspace. [Learn more](#)

Files ?

Drop files to upload, or [browse](#).

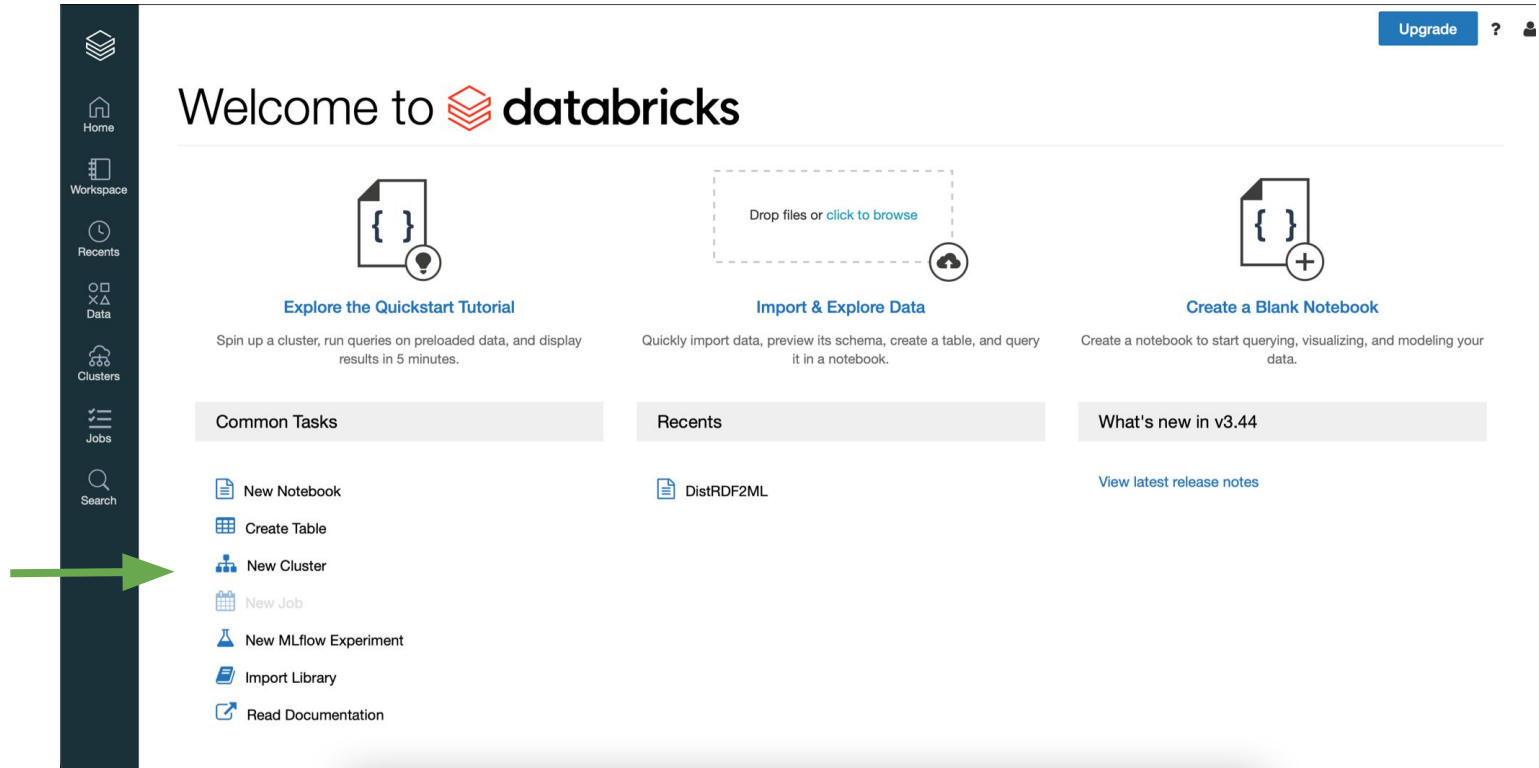
Setup Databricks - Upload needed Data - data

The screenshot displays the 'Create New Table' interface in Databricks. On the left is a dark sidebar with navigation icons: Home, Workspace, Recent (highlighted with a green arrow), Data (highlighted with a red arrow), Clusters, Jobs (highlighted with a green arrow), and Search. The main content area is titled 'Create New Table' and includes a 'Data source' section with tabs for 'Upload File' (selected), 'S3', 'DBFS', 'Other Data Sources', and 'Partner Integrations'. Below this is the 'DBFS Target Directory' section with a text input field containing '/FileStore/tables/' and '(optional)', followed by a 'Select' button. A note states: 'Files uploaded to DBFS are accessible by everyone who has access to this workspace. [Learn more](#)'. The 'Files' section shows a single file named 'sampleMovie' with a green checkmark, a size of '7.4 MB', and a 'Remove file' link. A status message at the bottom reads: '✓ File uploaded to /FileStore/tables/sampleMovieRDF5000.nt'. At the bottom of the interface are two buttons: 'Create Table with UI' and 'Create Table in Notebook'.

Setup Cluster

- set spark (each one line)
 - `spark.databricks.delta.preview.enabled true`
 - `spark.serializer org.apache.spark.serializer.KryoSerializer`
 - `spark.kryo.registrator net.sansa_stack.rdf.spark.io.JenaKryoRegistrator,`
`net.sansa_stack.query.spark.sparqlify.KryoRegistratorSparqlify`
- set jar


Setup Cluster



The screenshot shows the Databricks web interface. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. A green arrow points to the 'Clusters' icon. The main content area has a header 'Welcome to databricks' and three primary action cards: 'Explore the Quickstart Tutorial', 'Import & Explore Data', and 'Create a Blank Notebook'. Below these are three sections: 'Common Tasks' (listing New Notebook, Create Table, New Cluster, New Job, New MLflow Experiment, Import Library, and Read Documentation), 'Recents' (showing 'DistRDF2ML'), and 'What's new in v3.44' (with a link to 'View latest release notes'). The top right corner features an 'Upgrade' button and user icons.

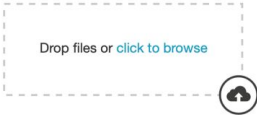
Upgrade ?

Welcome to databricks




Explore the Quickstart Tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



Import & Explore Data

Quickly import data, preview its schema, create a table, and query it in a notebook.



Create a Blank Notebook

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

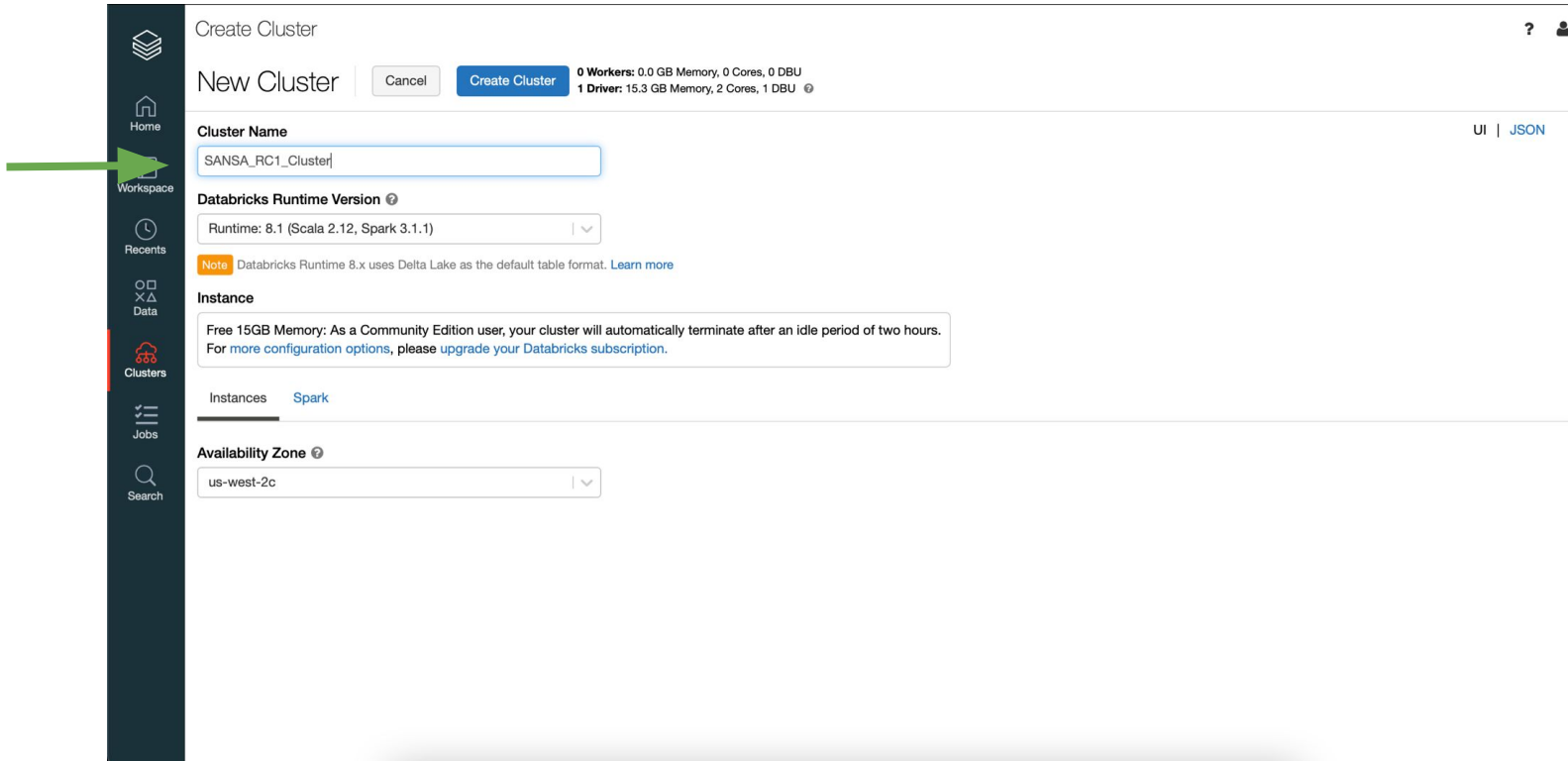
Recents

- DistRDF2ML

What's new in v3.44

[View latest release notes](#)

Setup Cluster - Give it a Name



Create Cluster

New Cluster Cancel Create Cluster **0 Workers:** 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU ⓘ

Cluster Name UI | JSON

SANSА_RC1_Cluster

Databricks Runtime Version ⓘ

Runtime: 8.1 (Scala 2.12, Spark 3.1.1) | v

Note Databricks Runtime 8.x uses Delta Lake as the default table format. [Learn more](#)

Instance

Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances Spark

Availability Zone ⓘ

us-west-2c | v

Setup Cluster - Set Spark

Create Cluster

New Cluster Cancel Create Cluster **0 Workers:** 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU ⓘ

Cluster Name
SANSА_RC1_Cluster

Databricks Runtime Version ⓘ
Runtime: 8.1 (Scala 2.12, Spark 3.1.1) | v

Note Databricks Runtime 8.x uses Delta Lake as the default table format. [Learn more](#)

Instance
Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours.
For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances Spark

Availability Zone ⓘ
us-west-2c | v

Setup Cluster - Set Spark

Create Cluster

New Cluster Cancel Create Cluster **0 Workers:** 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU ⓘ

Cluster Name
SANSА_RC1_Cluster

Databricks Runtime Version ⓘ
Runtime: 8.1 (Scala 2.12, Spark 3.1.1) | v

Note Databricks Runtime 8.x uses Delta Lake as the default table format. [Learn more](#)

Instance
Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours.
For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances **Spark**

Availability Zone ⓘ
us-west-2c | v

Setup Cluster - Set Spark

Create Cluster ?

New Cluster Cancel Create Cluster **0 Workers:** 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU

Cluster Name UI | [JSON](#)

Databricks Runtime Version

Runtime: 8.1 (Scala 2.12, Spark 3.1.1)

Note Databricks Runtime 8.x uses Delta Lake as the default table format. [Learn more](#)

Instance

Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances **Spark**

Spark Config

Enter your Spark configuration options here. Provide only one key-value pair per line. Example: spark.speculation true spark.kryo.registrator my.package.MyRegistrator

Environment Variables

PYSPARK_PYTHON=/databricks/python3/bin/python3

Setup Cluster - Set Spark

Create Cluster

New Cluster

Cancel Create Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU

Cluster Name

SANSA_RC1_Cluster

Databricks Runtime Version

Runtime: 8.1 (Scala 2.12, Spark 3.1.1)

Note Databricks Runtime 8.x uses Delta Lake as the default table format. [Learn more](#)

Instance

Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For more configuration options, please [upgrade your Databricks subscription](#).

Spark Config

```
spark.databricks.delta.preview.enabled true
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryo.registrator net.sansa_stack.rdf.spark.io.JenaKryoRegistrator,
net.sansa_stack.query.spark.sparqlify.KryoRegistratorSparqlify|
```

Environment Variables

PYSPARK_PYTHON=/databricks/python3/bin/python3

Setup Cluster - Set Spark

The screenshot shows the 'Create Cluster' page in Databricks. The left sidebar contains navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Recent. Green arrows point to the 'Workspace' icon, the 'Runtime: 8.1 (Scala 2.12, Spark 3.1.1)' dropdown, the 'Spark' tab under the 'Instance' section, and the 'Spark Config' text area. The main content area includes a 'New Cluster' header with 'Cancel' and 'Create Cluster' buttons, and resource specifications: '0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU' and '1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU'. The 'Cluster Name' field contains 'SANSА_RC1_Cluster'. The 'Databricks Runtime Version' is set to 'Runtime: 8.1 (Scala 2.12, Spark 3.1.1)'. A note states: 'Databricks Runtime 8.x uses Delta Lake as the default table format. Learn more'. The 'Instance' section has a warning: 'Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For more configuration options, please upgrade your Databricks subscription.' The 'Spark Config' field contains the following code:

```
spark.databricks.delta.preview.enabled true
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryo.registrator net.sansa_stack.rdf.spark.io.JenaKryoRegistrator,
net.sansa_stack.query.spark.sparqlify.KryoRegistratorSparqlify|
```

 The 'Environment Variables' field contains:

```
PYSPARK_PYTHON=/databricks/python3/bin/python3
```

Setup Cluster - Create Cluster

Home

Workspace

Recents

Data

Clusters

Jobs

Search

Create Cluster

?

Create Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU

UI | JSON

Cluster Name

SANSA_RC1_Cluster

Databricks Runtime Version

Runtime: 8.1 (Scala 2.12, Spark 3.1.1)

Note

Databricks Runtime 8.x uses Delta Lake as the default table format. [Learn more](#)

Instance

Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For more configuration options, please upgrade your Databricks subscription.

Instances

Spark

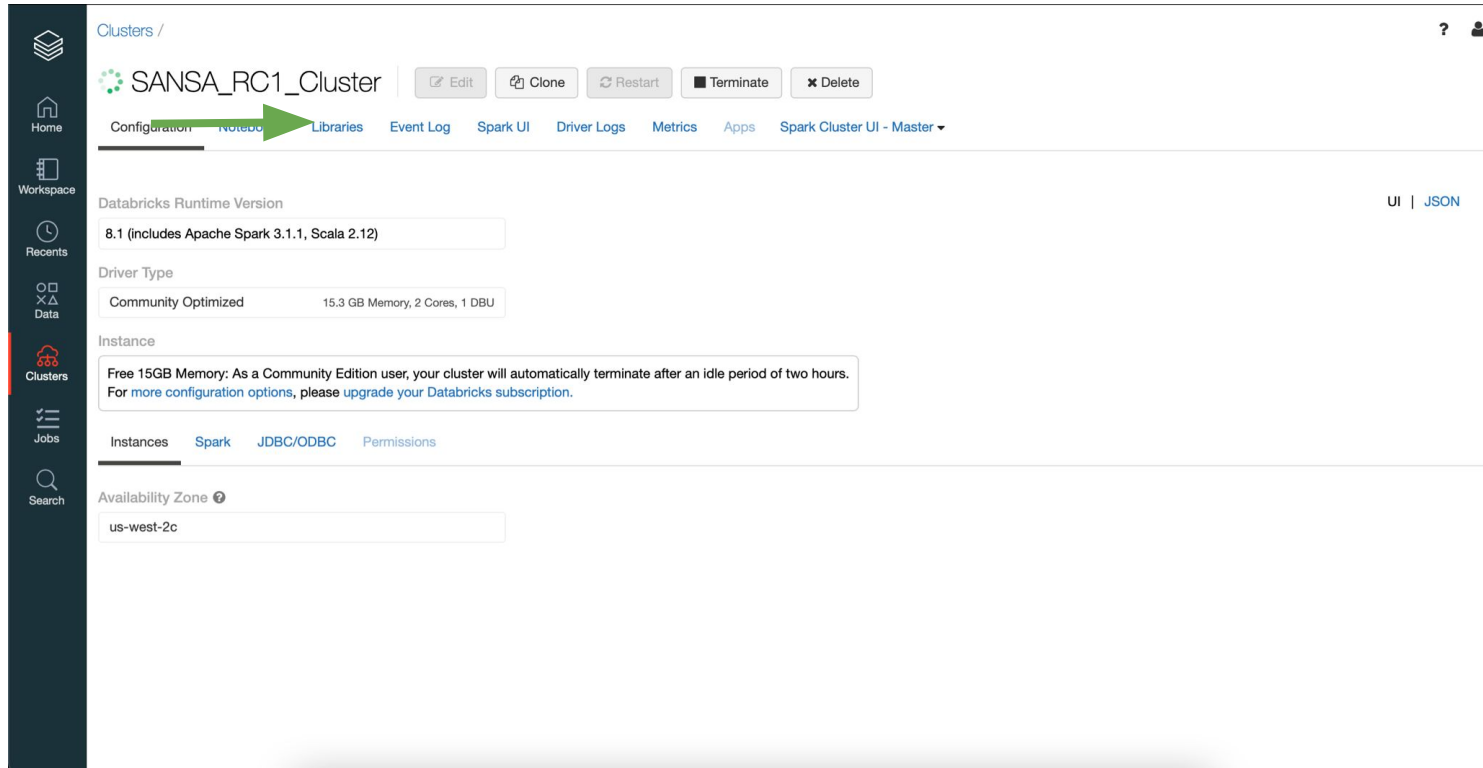
Spark Config

spark.databricks.delta.preview.enabled true
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryo.registrator net.sansa_stack.rdf.spark.io.JenaKryoRegistrator,
net.sansa_stack.query.spark.sparqlify.KryoRegistratorSparqlify

Environment Variables

PYSPARK_PYTHON=/databricks/python3/bin/python3

Setup Cluster - set SANSA JAR



The screenshot shows the Databricks Clusters management interface. On the left is a dark sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters (highlighted), Jobs, and Search. The main content area is titled 'Clusters /' and shows a cluster named 'SANSA_RC1_Cluster'. Above the cluster name are buttons for Edit, Clone, Restart, Terminate, and Delete. Below the name is a horizontal tab bar with 'Configuration', 'Libraries', 'Event Log', 'Spark UI', 'Driver Logs', 'Metrics', 'Apps', and 'Spark Cluster UI - Master'. A green arrow points to the 'Libraries' tab. The 'Configuration' tab is active, showing settings for 'Databricks Runtime Version' (8.1), 'Driver Type' (Community Optimized), and 'Instance' (Free 15GB Memory). At the bottom, the 'Availability Zone' is set to 'us-west-2c'.

Clusters /

SANSA_RC1_Cluster

Edit Clone Restart Terminate Delete

Configuration Libraries Event Log Spark UI Driver Logs Metrics Apps Spark Cluster UI - Master

Databricks Runtime Version

8.1 (includes Apache Spark 3.1.1, Scala 2.12)

Driver Type

Community Optimized 15.3 GB Memory, 2 Cores, 1 DBU

Instance

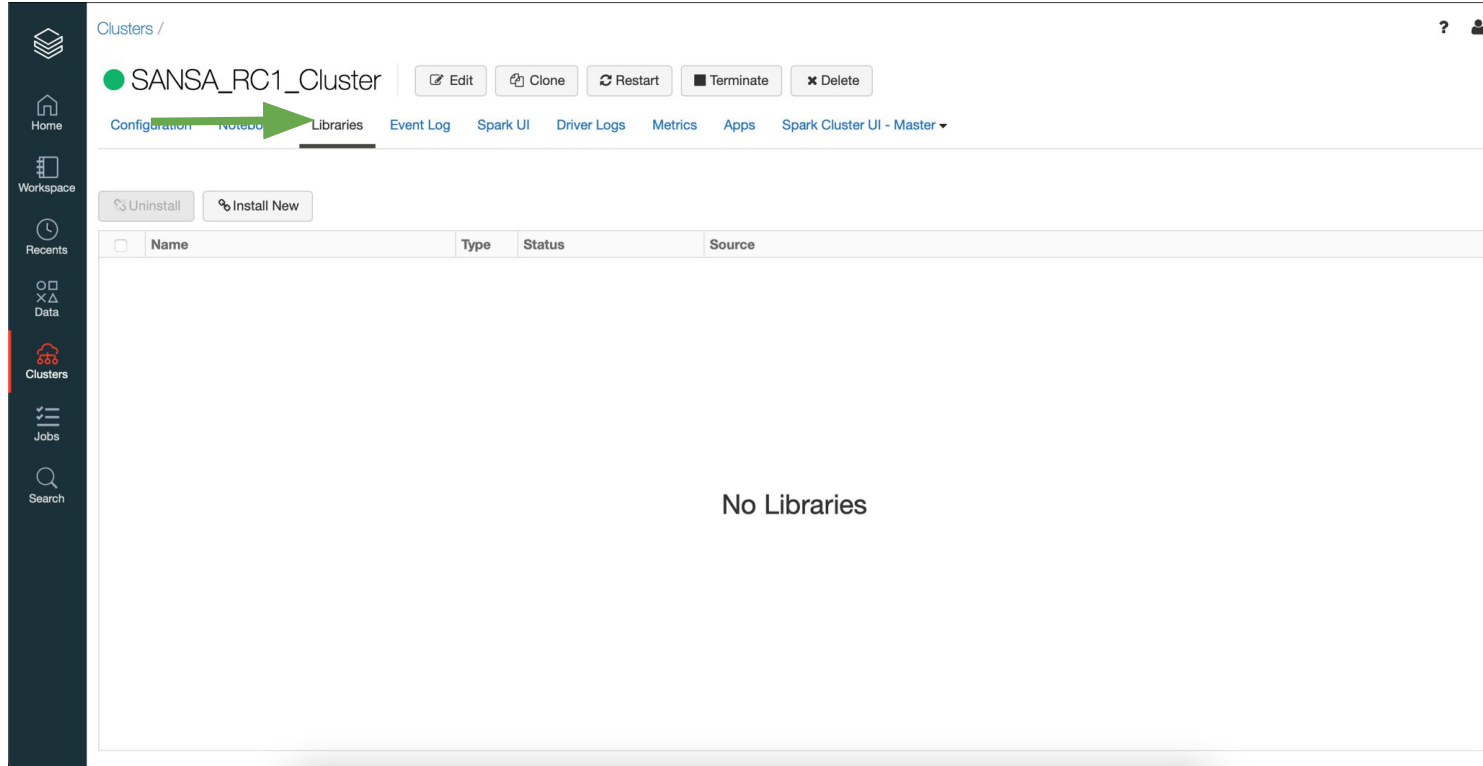
Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For more configuration options, please upgrade your Databricks subscription.

Instances Spark JDBC/ODBC Permissions

Availability Zone

us-west-2c

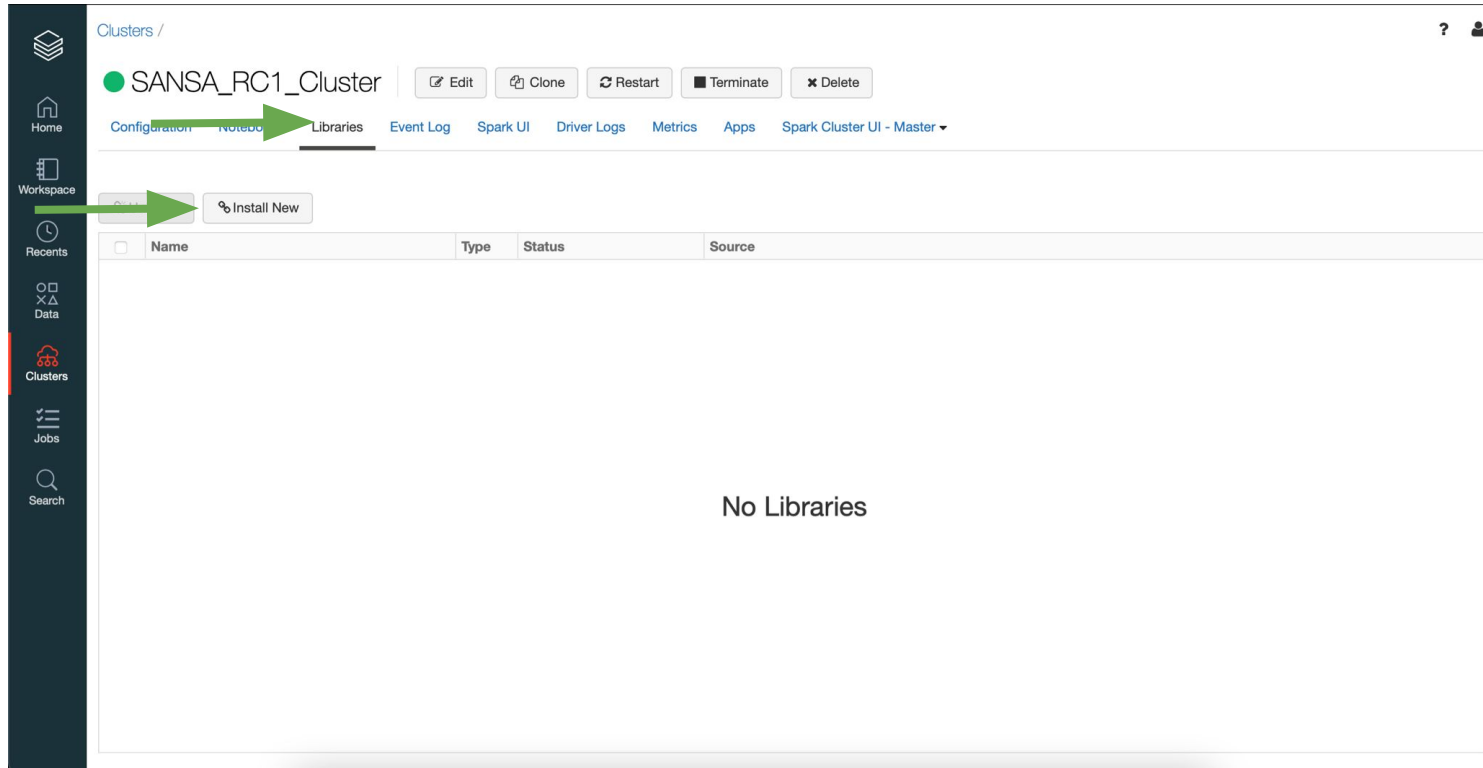
Setup Cluster - set SANSA JAR



The screenshot shows the Databricks Clusters management interface. On the left is a dark sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters (highlighted), Jobs, and Search. The main content area is titled 'Clusters /' and shows a cluster named 'SANSA_RC1_Cluster' with a green status dot. Above the cluster name are buttons for Edit, Clone, Restart, Terminate, and Delete. Below the cluster name is a horizontal menu with tabs: Configuration, Notebook (highlighted with a green arrow), Libraries, Event Log, Spark UI, Driver Logs, Metrics, Apps, and Spark Cluster UI - Master. Under the Libraries tab, there are 'Uninstall' and 'Install New' buttons. Below these buttons is a table with columns: Name, Type, Status, and Source. The table is currently empty, and the text 'No Libraries' is displayed in the center of the table area.

Name	Type	Status	Source
No Libraries			

Setup Cluster - set SANSA JAR



The screenshot shows the Databricks Clusters management interface. On the left is a dark sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters (highlighted), Jobs, and Search. The main content area is titled 'Clusters /' and shows a cluster named 'SANSA_RC1_Cluster' with a green status dot. Above the cluster name are buttons for Edit, Clone, Restart, Terminate, and Delete. Below the cluster name is a tabbed interface with tabs for Configuration, Libraries (selected), Event Log, Spark UI, Driver Logs, Metrics, Apps, and Spark Cluster UI - Master. A green arrow points from the 'Libraries' tab to the 'Install New' button. Below this is a table with columns: Name, Type, Status, and Source. The table is currently empty, and the text 'No Libraries' is displayed in the center.

Clusters /

SANSA_RC1_Cluster

Edit Clone Restart Terminate Delete

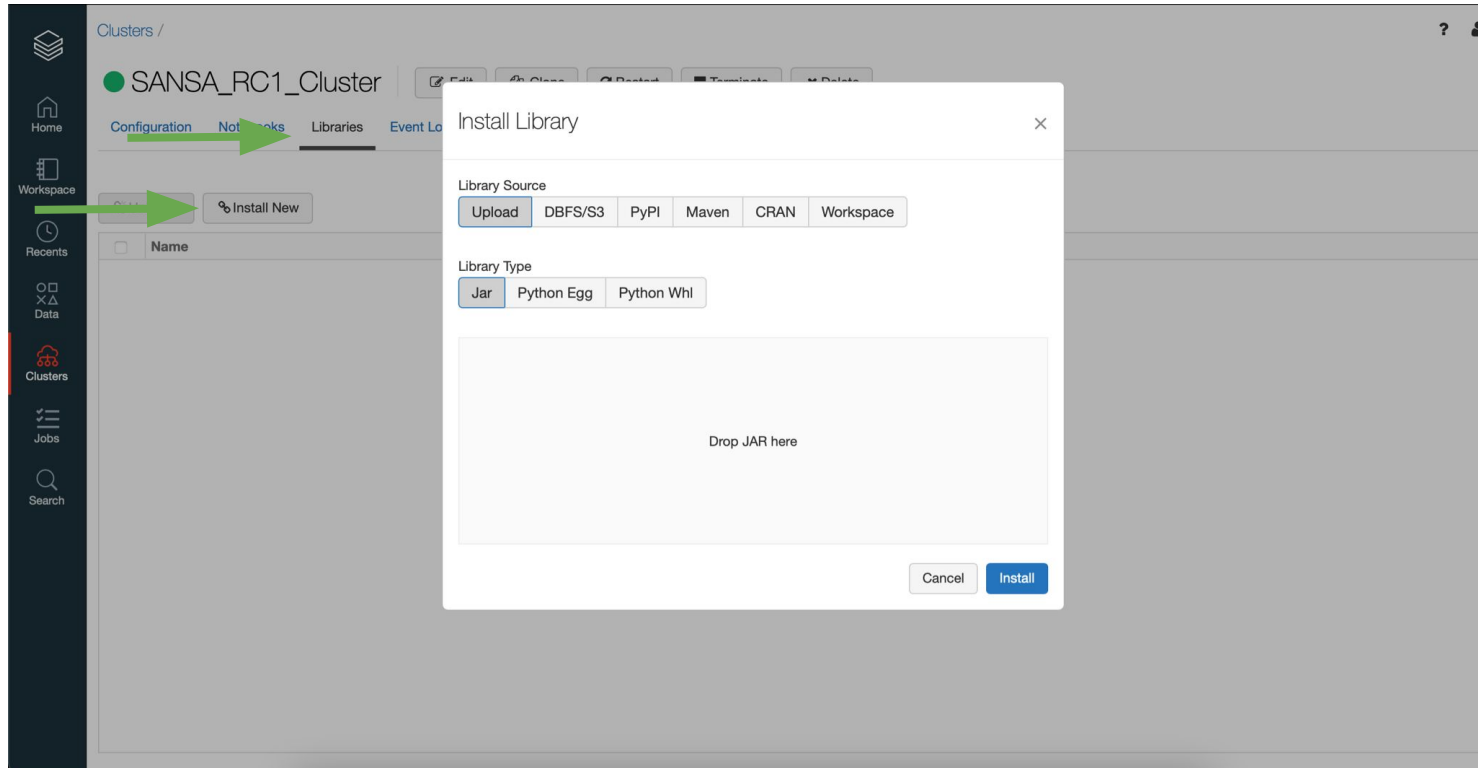
Configuration Libraries Event Log Spark UI Driver Logs Metrics Apps Spark Cluster UI - Master

Install New

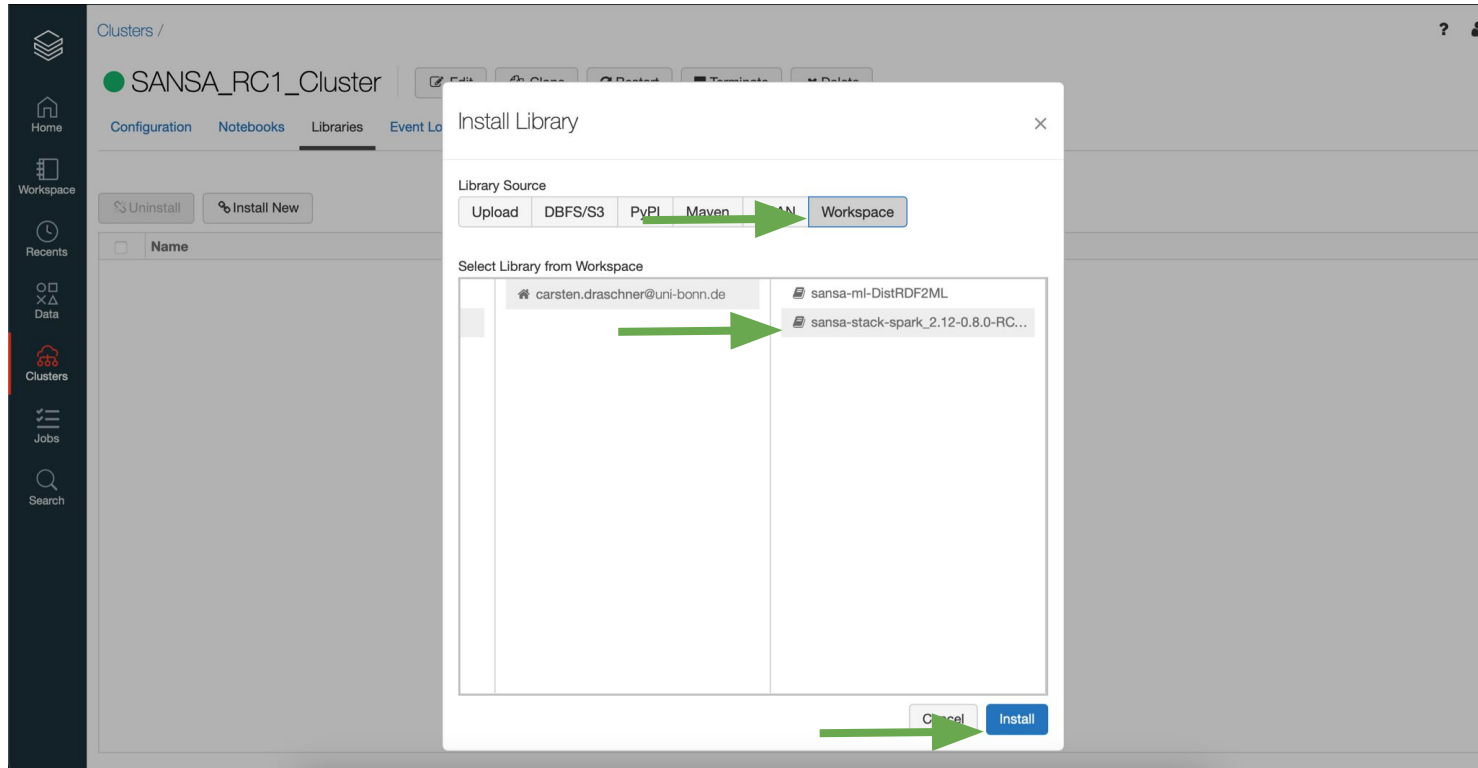
Name	Type	Status	Source
------	------	--------	--------

No Libraries

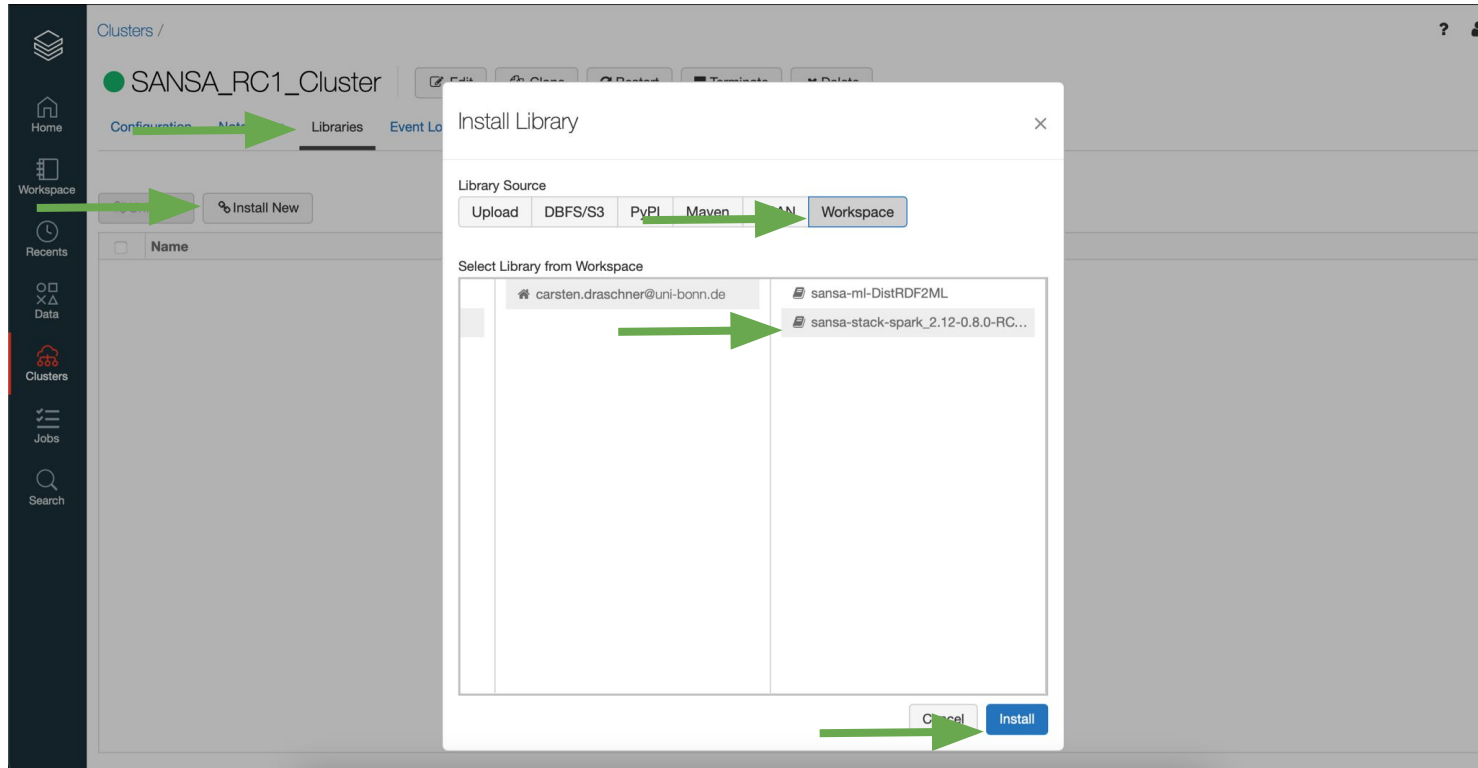
Setup Cluster - set SANSA JAR



Setup Cluster - set SANSA JAR



Setup Cluster - set SANSA JAR



Setup Cluster - set SANSA JAR

The screenshot shows the Databricks Clusters management interface. On the left is a dark sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters (highlighted), Jobs, and Search. The main content area is titled 'Clusters /' and features a cluster named 'SANSA_RC1_Cluster' with a green status dot. Above the cluster name are buttons for Edit, Clone, Restart, Terminate, and Delete. Below the name is a tabbed interface with 'Configuration' selected, indicated by a green arrow. Other tabs include Libraries, Event Log, Spark UI, Driver Logs, Metrics, Apps, and Spark Cluster UI - Master. Under the Configuration tab, there are 'Uninstall' and 'Install New' buttons. Below these is a table with columns: Name, Type, Status, and Source. A green arrow points to the first row of the table, which contains the text 'sansa_stack_spark_2_12_0_8_0_RC1_jar_with' in the Name column, 'Installing' in the Status column, and a long file path in the Source column.

Name	Type	Status	Source
sansa_stack_spark_2_12_0_8_0_RC1_jar_with		Installing	dbfs:/FileStore/jars/5c3bceaf_0aa9_4e83_9deb_f81dde233638-sansa_stack_spark_2_12_0_8_0_RC1_jar_with_dependencies-...

Setup Cluster - set SANSA JAR

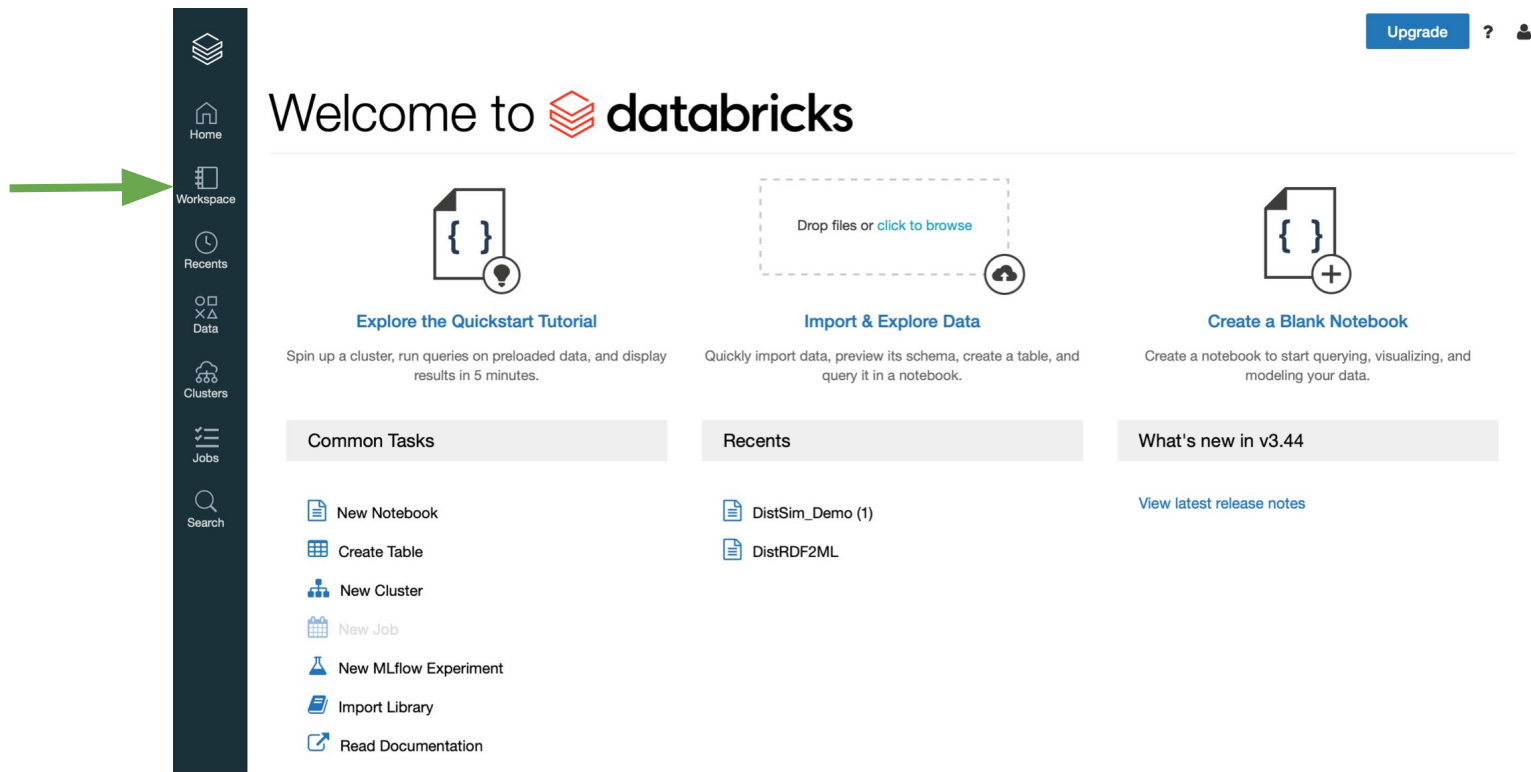
The screenshot shows the Databricks Clusters management interface. The cluster 'SANSA_RC1_Cluster' is selected, and the 'Configuration' tab is active. The 'Libraries' section displays a table of installed libraries. A green arrow points to the 'Configuration' tab, and another green arrow points to the 'sanse_stack_spark_2_12_0_8_0_RC1_jar_with_dependencies' entry in the libraries table.

<input type="checkbox"/>	Name	Type	Status	Source
<input type="checkbox"/>	sanse_stack_spark_2_12_0_8_0_RC1_jar_with_dependencies		Installed	dbfs:/FileStore/jars/5c3bceaf_0aa9_4e83_9deb_f81dde233638-sansa_stack_spark_2_12_0_8_0_RC1_jar_with_dependencies-...

Import Notebook

- Notebook
 - <https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/6924783690087984/3044317244801485/8524188481975304/latest.html>

Import Notebook



The screenshot shows the Databricks web interface. On the left is a dark sidebar with icons for Home, Workspace (highlighted with a green arrow), Recents, Data, Clusters, Jobs, and Search. The main content area has a header 'Welcome to databricks' with the Databricks logo. Below the header are three main action cards: 'Explore the Quickstart Tutorial', 'Import & Explore Data', and 'Create a Blank Notebook'. Each card has an icon and a brief description. At the bottom, there are three sections: 'Common Tasks' with links to New Notebook, Create Table, New Cluster, New Job, New MLflow Experiment, Import Library, and Read Documentation; 'Recents' with a list of recent notebooks; and 'What's new in v3.44' with a link to the latest release notes.

Upgrade ?

Welcome to databricks

Explore the Quickstart Tutorial
Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

Import & Explore Data
Quickly import data, preview its schema, create a table, and query it in a notebook.

Create a Blank Notebook
Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

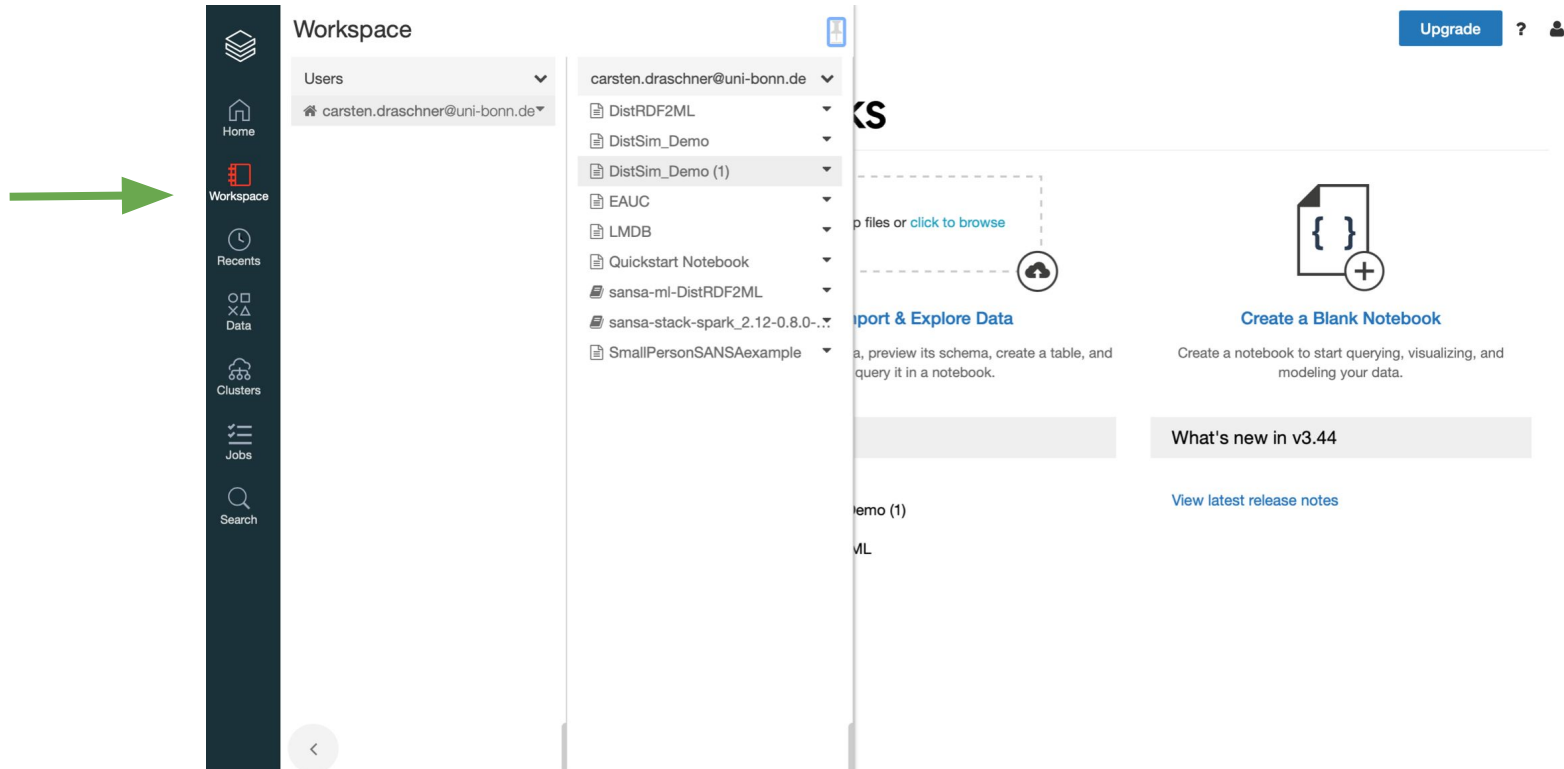
Recents

- DistSim_Demo (1)
- DistRDF2ML

What's new in v3.44

[View latest release notes](#)

Import Notebook

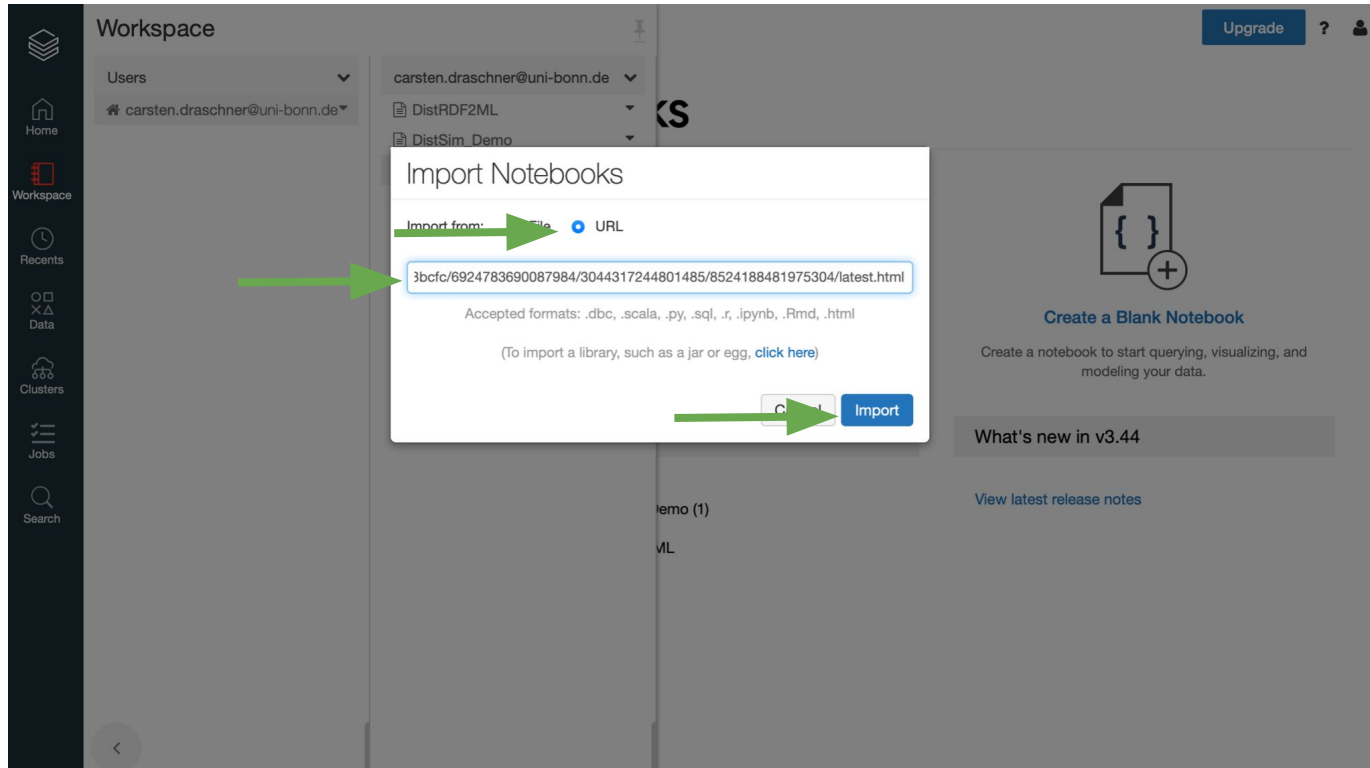


The screenshot shows the Databricks workspace interface. On the left, a dark sidebar contains navigation icons: Home, Workspace (highlighted with a green arrow), Recents, Data, Clusters, Jobs, and Search. The main workspace area is titled 'Workspace' and shows a list of notebooks under the user 'carsten.draschner@uni-bonn.de'. The notebooks listed are 'DistSim_Demo', 'DistSim_Demo (1)', 'EAUC', 'LMDB', 'Quickstart Notebook', 'sansa-ml-DistRDF2ML', 'sansa-stack-spark_2.12-0.8.0-...', and 'SmallPersonSANSAXample'. Below the list, there are two main sections: 'Import & Explore Data' and 'Create a Blank Notebook'. The 'Import & Explore Data' section includes a dashed box with a cloud icon and the text 'Import files or click to browse'. The 'Create a Blank Notebook' section includes a notebook icon with a plus sign and the text 'Create a notebook to start querying, visualizing, and modeling your data.'.

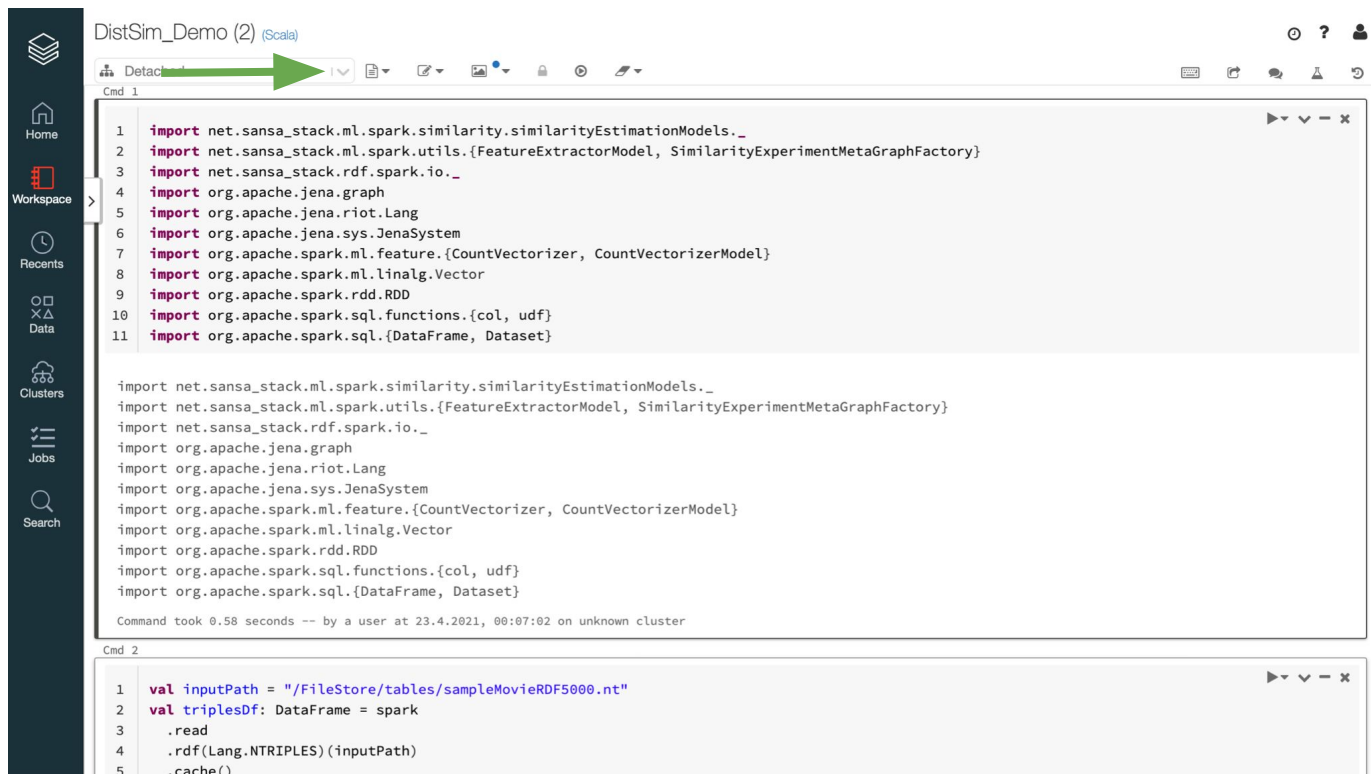
Import Notebook

The screenshot displays the SANSANSA web interface. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. A green arrow points to the 'Workspace' icon. The main area is titled 'Workspace' and shows a list of users under a 'Users' dropdown. The user 'carsten.draschner@uni-bonn.de' is selected, with a green arrow pointing to their name. Below the user list is a list of notebooks: DistRDF2ML, DistSim_Demo, DistSim_Demo (1), EAUC, LMDB, Quickstart Notebook, sansa-ml-DistRDF2ML, sansa-stack-spark_2.12-0.8.0-..., and SmallPersonSANSAAexample. A green arrow points to 'DistSim_Demo (1)'. A context menu is open over this notebook, showing options: Create, Clone, Import (highlighted), Export, Permissions, and Copy Link Address. To the right of the notebook list is a section titled 'Import & Explore Data' with a description: 'a, preview its schema, create a table, and query it in a notebook.' Further right is a 'Create a Blank Notebook' section with a description: 'Create a notebook to start querying, visualizing, and modeling your data.' Below this is a 'What's new in v3.44' section with a link to 'View latest release notes'. In the top right corner, there is an 'Upgrade' button and user profile icons.

Import Notebook



Run Notebook - Select Cluster



The screenshot shows a web-based notebook interface for "DistSim_Demo (2) (Scala)". On the left is a dark sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main area displays Scala code in two command blocks. A green arrow points to the "Detached" dropdown menu in the top toolbar.

Command 1:

```
1 import net.sansa_stack.ml.spark.similarity.similarityEstimationModels._
2 import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
3 import net.sansa_stack.rdf.spark.io._
4 import org.apache.jena.graph
5 import org.apache.jena.riot.Lang
6 import org.apache.jena.sys.JenaSystem
7 import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
8 import org.apache.spark.ml.linalg.Vector
9 import org.apache.spark.rdd.RDD
10 import org.apache.spark.sql.functions.{col, udf}
11 import org.apache.spark.sql.{DataFrame, Dataset}
```

Command 2:

```
1 val inputPath = "/FileStore/tables/sampleMovieRDF5000.nt"
2 val triplesDf: DataFrame = spark
3   .read
4   .rdf(Lang.NTRIPLES)(inputPath)
5   .cache()
```

Run Notebook - Select Cluster

The screenshot shows a web-based notebook interface for 'DistSim_Demo (2) (Scala)'. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The 'Clusters' icon is highlighted with a green arrow. At the top of the notebook area, a dropdown menu is open, showing the 'Attach:' section with a list of clusters. The 'distsim' cluster is selected, with a green arrow pointing to it. The dropdown also shows 'DBR 8.1 | Spark 3.1.1 | Scala 2.12'. Below the dropdown, the notebook code is visible, consisting of two sections of Scala code. The first section contains imports for various libraries like 'net.sansa_stack.rdf.spark.io._', 'org.apache.jena.graph', 'org.apache.jena.riot.Lang', 'org.apache.jena.sys.JenaSystem', 'org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}', 'org.apache.spark.ml.linalg.Vector', 'org.apache.spark.rdd.RDD', 'org.apache.spark.sql.functions.{col, udf}', and 'org.apache.spark.sql.{DataFrame, Dataset}'. The second section contains imports for 'net.sansa_stack.ml.spark.similarity.similarityEstimationModels._', 'net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}', and 'net.sansa_stack.rdf.spark.io._'. Below the code, a status message indicates 'Command took 0.58 seconds -- by a user at 23.4.2021, 00:07:02 on unknown cluster'. At the bottom, a command prompt shows the execution of a Scala command to read a dataset from a file store.

```
DistSim_Demo (2) (Scala)
```

Attach:

- distsim
DBR 8.1 | Spark 3.1.1 | Scala 2.12

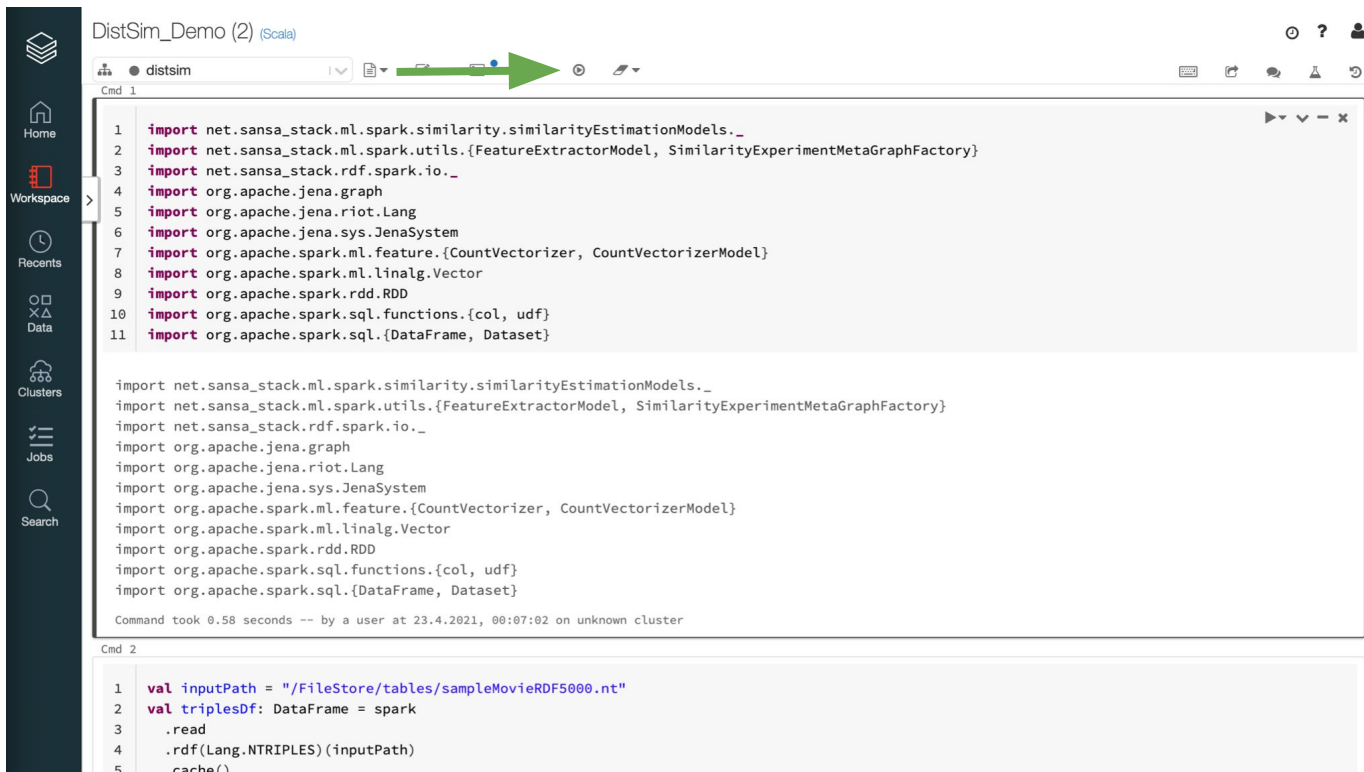
```
3 import net.sansa_stack.rdf.spark.io._
4 import org.apache.jena.graph
5 import org.apache.jena.riot.Lang
6 import org.apache.jena.sys.JenaSystem
7 import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
8 import org.apache.spark.ml.linalg.Vector
9 import org.apache.spark.rdd.RDD
10 import org.apache.spark.sql.functions.{col, udf}
11 import org.apache.spark.sql.{DataFrame, Dataset}

import net.sansa_stack.ml.spark.similarity.similarityEstimationModels._
import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
import net.sansa_stack.rdf.spark.io._
import org.apache.jena.graph
import org.apache.jena.riot.Lang
import org.apache.jena.sys.JenaSystem
import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
import org.apache.spark.ml.linalg.Vector
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.functions.{col, udf}
import org.apache.spark.sql.{DataFrame, Dataset}

Command took 0.58 seconds -- by a user at 23.4.2021, 00:07:02 on unknown cluster
```

```
Cmd 2
1 val inputPath = "/FileStore/tables/sampleMovieRDF5000.nt"
2 val triplesDf: DataFrame = spark
3   .read
4   .rdf(Lang.NTRIPLES)(inputPath)
5   .cache()
```

Run Notebook - Run all cells



The screenshot shows a web-based notebook interface for 'DistSim_Demo (2) (Scala)'. On the left is a dark sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main area displays two code cells. The first cell, 'Cmd 1', contains 11 import statements for various libraries including sansa_stack, jena, and spark. A green arrow points to the 'Run all cells' button (a circle with a play icon) in the toolbar above the code editor. The second cell, 'Cmd 2', contains Scala code for loading a dataset from a file store. The status bar at the bottom of the first cell indicates the command took 0.58 seconds.

```
Cmd 1

1 import net.sansa_stack.ml.spark.similarity.similarityEstimationModels._
2 import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
3 import net.sansa_stack.rdf.spark.io._
4 import org.apache.jena.graph
5 import org.apache.jena.riot.Lang
6 import org.apache.jena.sys.JenaSystem
7 import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
8 import org.apache.spark.ml.linalg.Vector
9 import org.apache.spark.rdd.RDD
10 import org.apache.spark.sql.functions.{col, udf}
11 import org.apache.spark.sql.{DataFrame, Dataset}

import net.sansa_stack.ml.spark.similarity.similarityEstimationModels._
import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
import net.sansa_stack.rdf.spark.io._
import org.apache.jena.graph
import org.apache.jena.riot.Lang
import org.apache.jena.sys.JenaSystem
import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
import org.apache.spark.ml.linalg.Vector
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.functions.{col, udf}
import org.apache.spark.sql.{DataFrame, Dataset}

Command took 0.58 seconds -- by a user at 23.4.2021, 00:07:02 on unknown cluster
```

```
Cmd 2

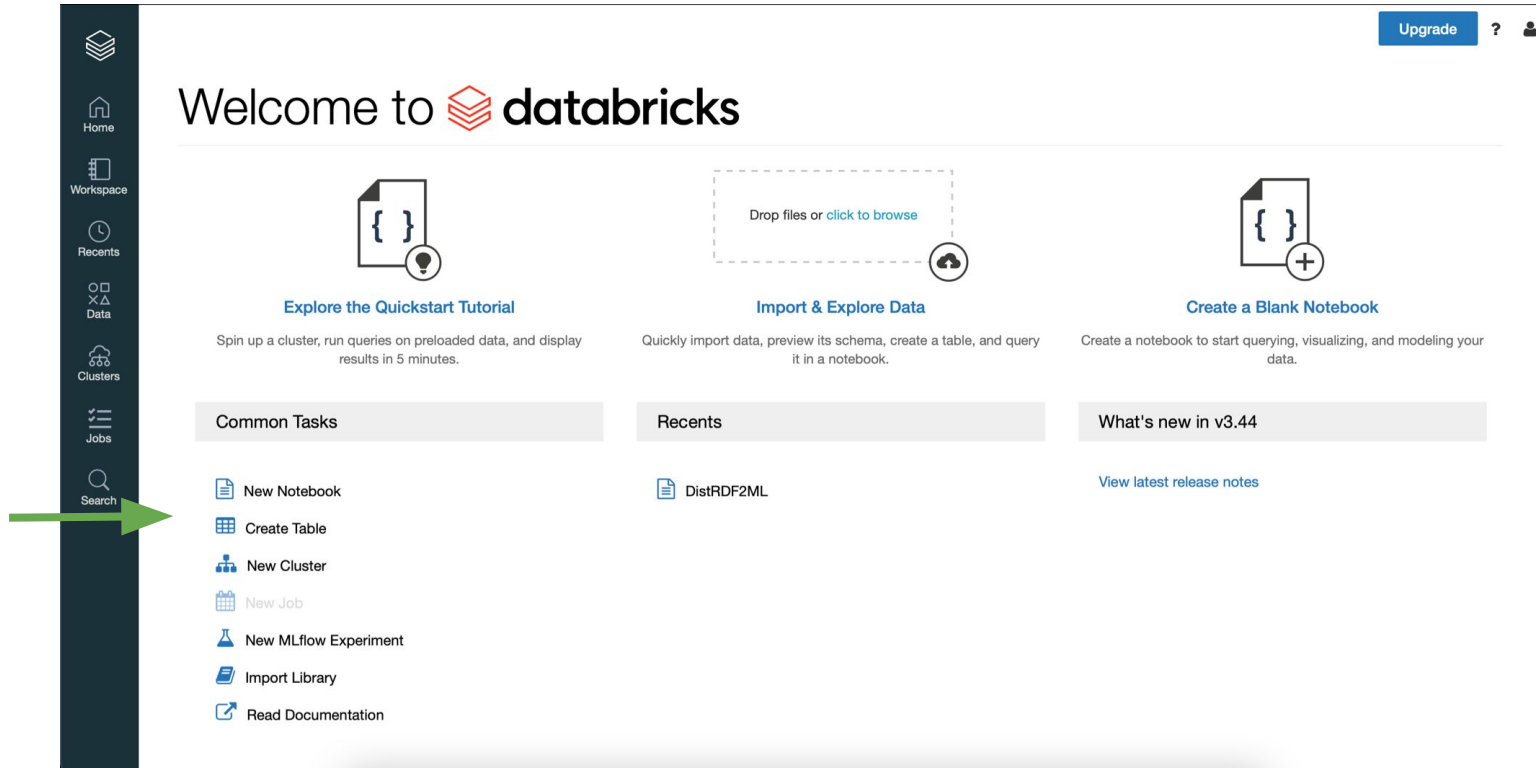
1 val inputPath = "/FileStore/tables/sampleMovieRDF5000.nt"
2 val triplesDf: DataFrame = spark
3   .read
4   .rdf(Lang.NTRIPLES)(inputPath)
5   .cache()
```

You Made It!!!

Create Notebook

- Create Notebook
- Attach Cluster
- Read In Data
- Perform DistSim Modules


Create Notebook - Create Notebook



The screenshot shows the Databricks web interface. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. A green arrow points to the Search icon. The main content area has a header with the Databricks logo and a 'Welcome to databricks' message. Below this are three large cards: 'Explore the Quickstart Tutorial', 'Import & Explore Data', and 'Create a Blank Notebook'. At the bottom, there are three sections: 'Common Tasks' with links to New Notebook, Create Table, New Cluster, New Job, New MLflow Experiment, Import Library, and Read Documentation; 'Recents' showing 'DistRDF2ML'; and 'What's new in v3.44' with a link to 'View latest release notes'. In the top right corner, there is an 'Upgrade' button and user icons.


Upgrade ?

Welcome to databricks




Explore the Quickstart Tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.



Import & Explore Data

Quickly import data, preview its schema, create a table, and query it in a notebook.



Create a Blank Notebook

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

Recents

- DistRDF2ML

What's new in v3.44

[View latest release notes](#)

Create Notebook - Create Notebook

The screenshot shows the Databricks web interface. A 'Create Notebook' modal dialog is open in the center. The dialog has three input fields: 'Name' with the value 'DistSim_Demo', 'Default Language' with a dropdown set to 'Scala', and 'Cluster' with a dropdown set to 'SANSA_RC1_Cluster'. At the bottom of the dialog are 'Cancel' and 'Create' buttons. In the background, the Databricks home page is visible. On the left is a dark sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area has a 'Welcome to Databricks' header, a 'Create a Blank Notebook' button, and a 'Common Tasks' section with links like 'New Notebook', 'Create Table', 'New Cluster', 'New Job', 'New MLflow Experiment', 'Import Library', and 'Read Documentation'. A 'Recents' section shows a notebook named 'DistRDF2ML'. A 'What's new in v3.44' section is also present.

Upgrade ?

Welcome to Databricks

Create Notebook

Name: DistSim_Demo

Default Language: Scala

Cluster: SANSA_RC1_Cluster

Cancel Create

Explore the Quickstart tutorial

Spin up a cluster, run queries on preloaded data, and get results in 5 minutes.

Create a Blank Notebook

Create a notebook to start querying, visualizing, and modeling your data.

Common Tasks

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

Recents

- DistRDF2ML

What's new in v3.44

[View latest release notes](#)

Create Notebook - Create Notebook

Welcome to Databricks

Create Notebook

Name:

Default Language:

Cluster:

Common Tasks

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

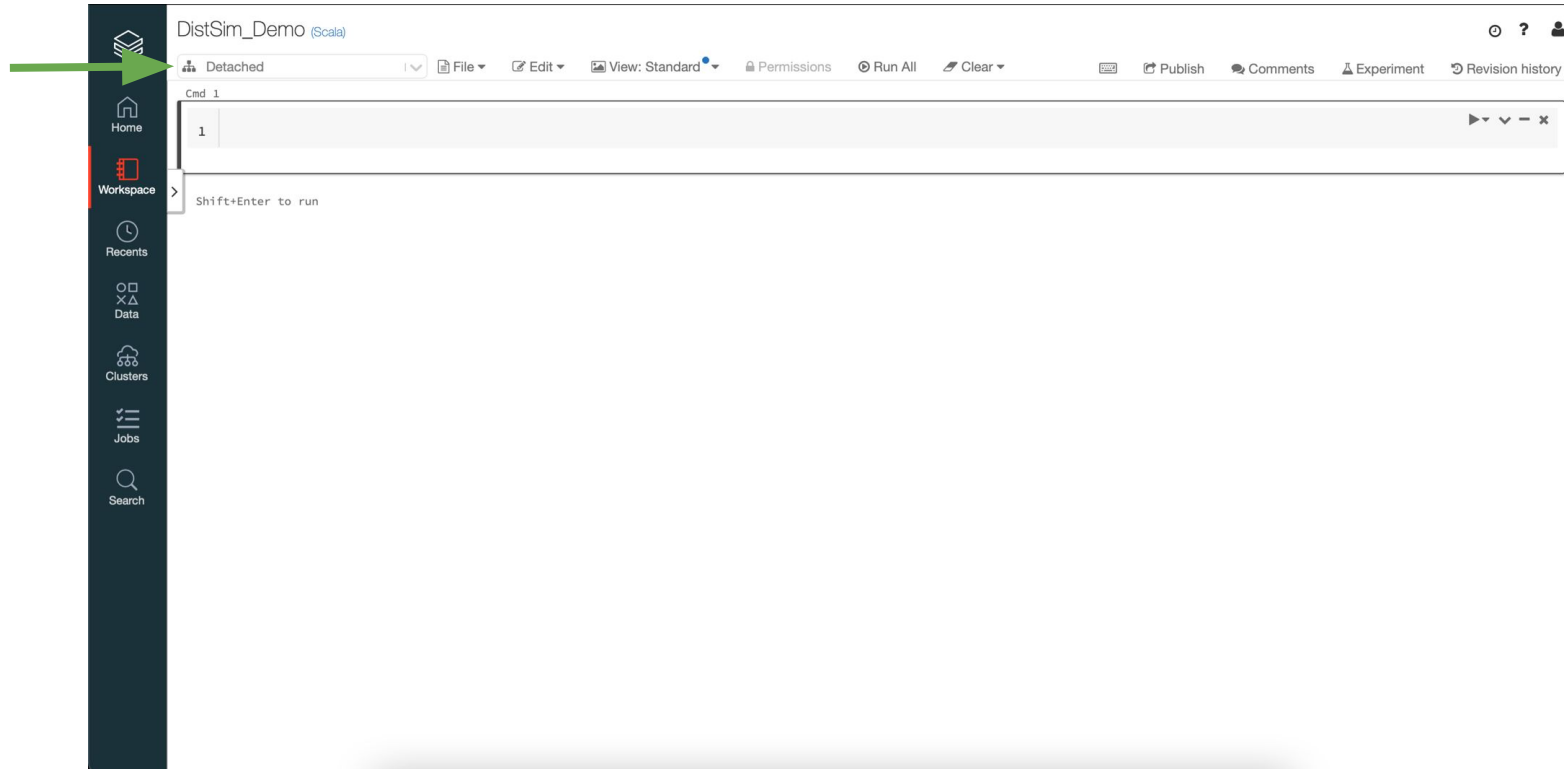
Recents

- DistRDF2ML

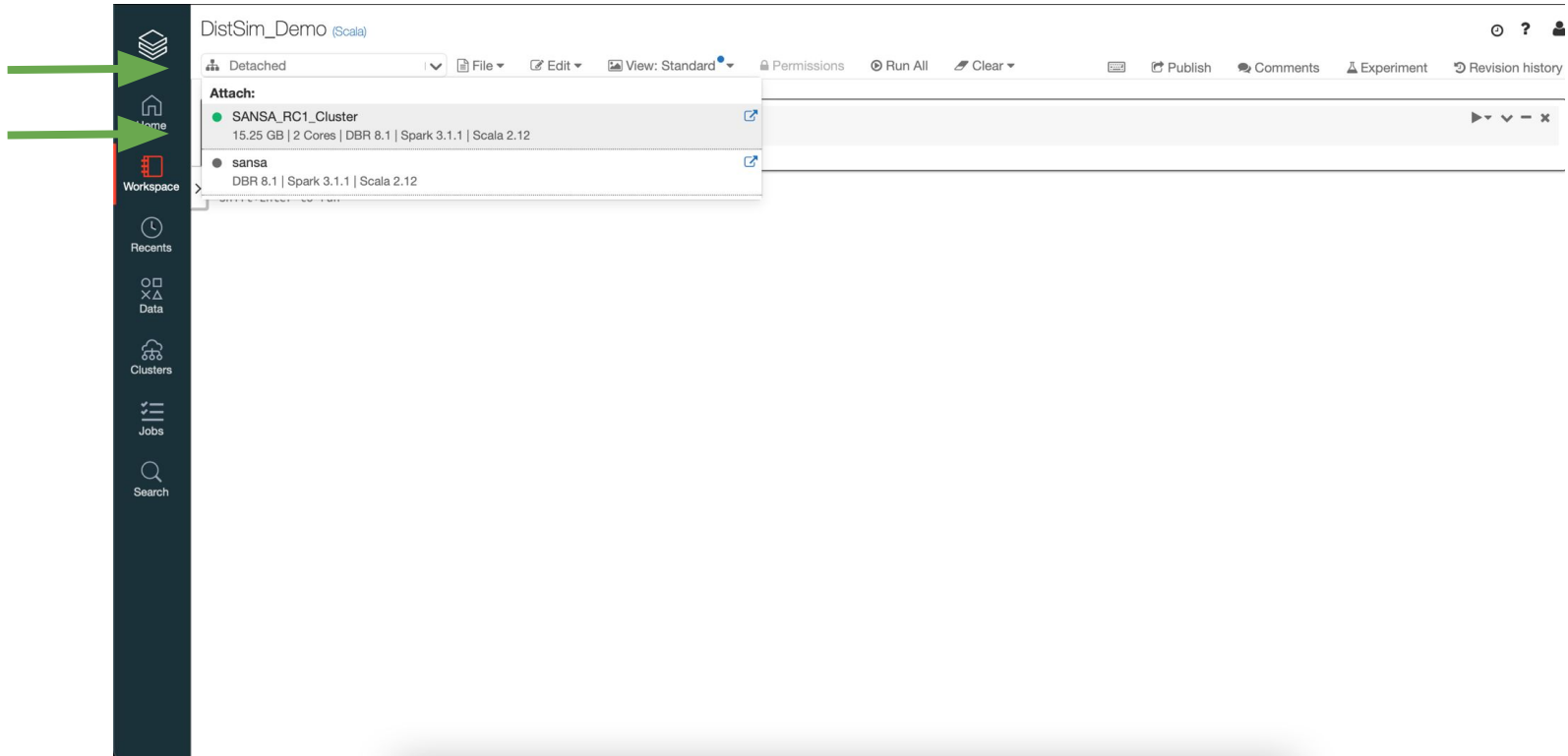
What's new in v3.44

[View latest release notes](#)

Create Notebook - Specify Cluster



Create Notebook - Specify Cluster



Code in Notebook - Imports

```
import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
import net.sansa_stack.rdf.spark.io._
import org.apache.jena.graph
import org.apache.jena.riot.Lang
import org.apache.jena.sys.JenaSystem
import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
import org.apache.spark.ml.linalg.Vector
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.functions.{col, udf}
import org.apache.spark.sql.{DataFrame, Dataset}
```

Code in Notebook - Imports

The screenshot shows a web-based notebook interface. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main area has a header bar with the title 'DistSim_Demo (Scala)' and various tool icons. Below the header, the code editor shows a list of imports for Scala code. The code is as follows:


```
1 import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
2 import net.sansa_stack.rdf.spark.io._
3 import org.apache.jena.graph
4 import org.apache.jena.riot.Lang
5 import org.apache.jena.sys.JenaSystem
6 import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
7 import org.apache.spark.ml.linalg.Vector
8 import org.apache.spark.rdd.RDD
9 import org.apache.spark.sql.functions.{col, udf}
10 import org.apache.spark.sql.{DataFrame, Dataset}
```


Below the code editor, the output of the command is displayed:


```
import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
import net.sansa_stack.rdf.spark.io._
import org.apache.jena.graph
import org.apache.jena.riot.Lang
import org.apache.jena.sys.JenaSystem
import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
import org.apache.spark.ml.linalg.Vector
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.functions.{col, udf}
import org.apache.spark.sql.{DataFrame, Dataset}
```


Command took 10.22 seconds -- by carsten.draschner@uni-bonn.de at 22/04/2021, 23:53:10 on SANSa_RC1_Cluster


Code in Notebook - Imports



Home



Workspace


Recents


Data


Clusters


Jobs


Search

DistSim_Demo (Scala)

SANSA_RC1_Cluster

File

Edit

View: Standard

Permissions

Run All

Clear

Publish

Comments

Experiment

Revision history

```
1 val inputPath = "/FileStore/tables/sampleMovieRDF5000.nt"
2 val triplesDf: DataFrame = spark.read.rdf(Lang.NTRIPLES)(inputPath).cache()
3 triplesDf.show(false)
```

(1) Spark Jobs

triplesDf: org.apache.spark.sql.DataFrame = [s: string, p: string ... 1 more fields]

s	p	o
movie_information.org/actor/actor16221	movie_information.org/actedIn	movie_information.org/movie/movie2998
movie_information.org/actor/actor90051	movie_information.org/actedIn	movie_information.org/movie/movie228
movie_information.org/actor/actor91387	movie_information.org/actedIn	movie_information.org/movie/movie3271
movie_information.org/actor/actor22072	movie_information.org/actedIn	movie_information.org/movie/movie1913
movie_information.org/actor/actor2229	movie_information.org/actedIn	movie_information.org/movie/movie4367
movie_information.org/actor/actor57215	movie_information.org/actedIn	movie_information.org/movie/movie4389
movie_information.org/actor/actor30676	movie_information.org/actedIn	movie_information.org/movie/movie250
movie_information.org/actor/actor63422	movie_information.org/actedIn	movie_information.org/movie/movie2044
movie_information.org/actor/actor74331	movie_information.org/actedIn	movie_information.org/movie/movie258
movie_information.org/actor/actor7195	movie_information.org/actedIn	movie_information.org/movie/movie947
movie_information.org/movie/movie1752	movie_information.org/publishedDate	"10/25/1996 06:53 AM"
movie_information.org/actor/actor86254	movie_information.org/actedIn	movie_information.org/movie/movie1373
movie_information.org/actor/actor78346	movie_information.org/actedIn	movie_information.org/movie/movie1970
movie_information.org/movie/movie3362	movie_information.org/publishedDate	"12/28/2014 04:23 AM"
movie_information.org/actor/actor91257	movie_information.org/actedIn	movie_information.org/movie/movie3169

Code in Notebook - Imports

The screenshot displays a Databricks notebook titled "DistSim_Demo (Scala)". The interface includes a left sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The top toolbar contains various controls like cluster selection (SANSA_RC1_Cluster), file operations, editing tools, view modes, permissions, and execution buttons (Run All, Clear). The main area shows Scala code for creating a FeatureExtractorModel and applying it to a dataset, with a filter for URIs starting with "m". Below the code, the output of the job is shown as a table with two columns: "uri" and "extractedFeatures".

```
1 val mode = "at"
2 val featureExtractorModel = new FeatureExtractorModel()
3   .setMode(mode)
4   .setOutputCol("extractedFeatures")
5 val extractedFeaturesDataFrame = featureExtractorModel
6   .transform(triplesDf)
7   .filter(t => t.getAs[String]("uri").startsWith("m"))
8 extractedFeaturesDataFrame.show()
```

▶ (2) Spark Jobs

▶ extractedFeaturesDataFrame: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [uri: string, extractedFeatures: array]

uri	extractedFeatures
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...
movie_information...	[-movie_informati...