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 istallation required
    - available over browser
    - native scala spark notebooks
    - no local performance needed

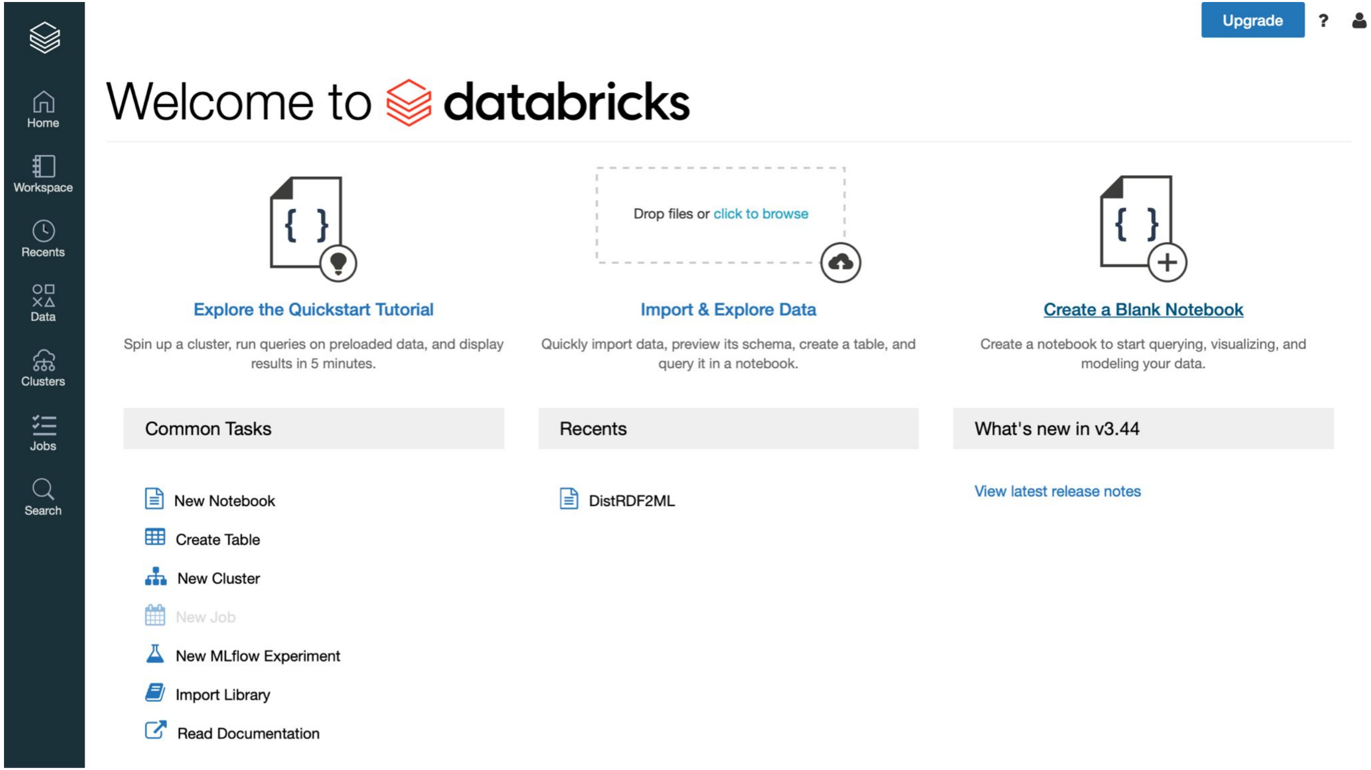
Databricks Registration

- 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



Welcome to databricks

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

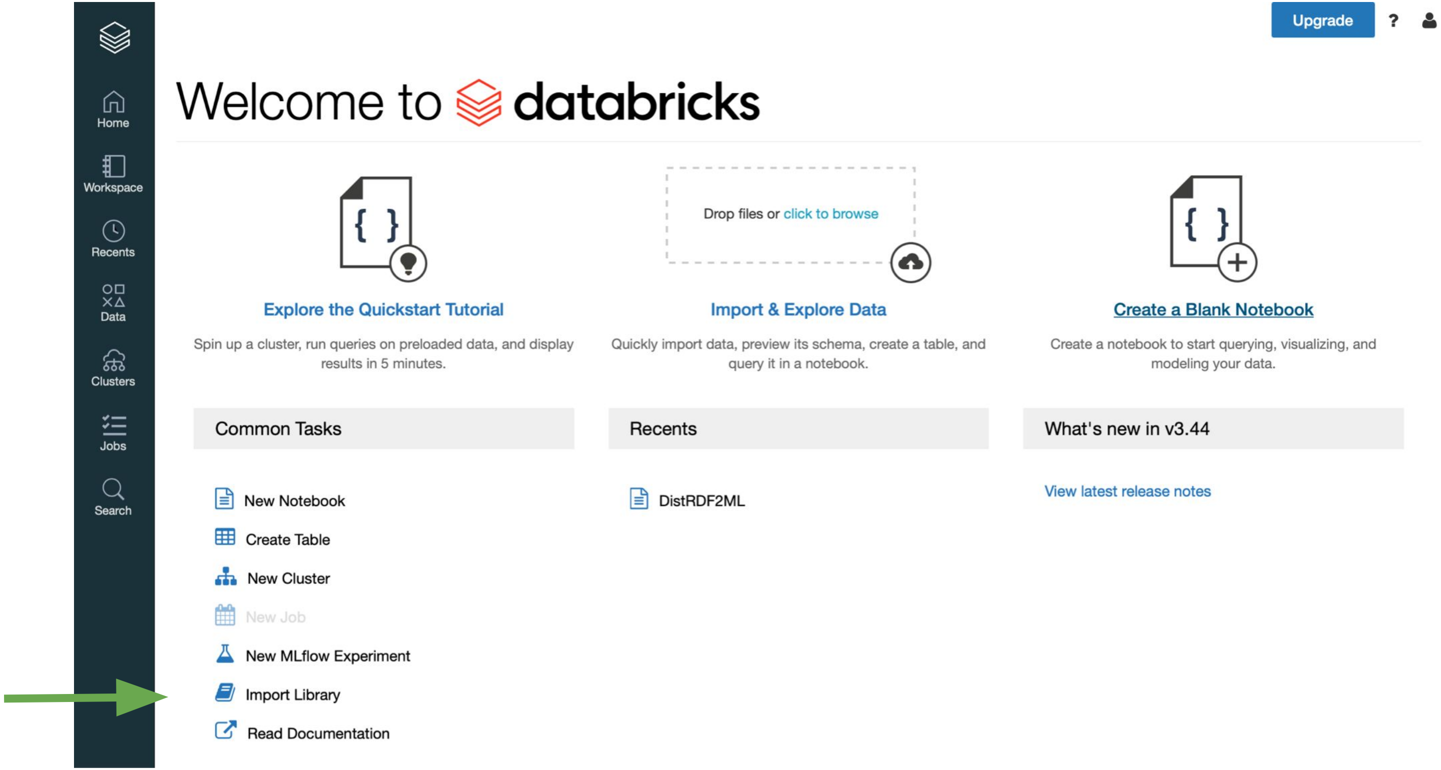
Recents

- DistRDF2ML

What's new in v3.44

View latest release notes

Setup Databricks - Upload needed Data - jar



Welcome to databricks

Drop files or click to browse

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

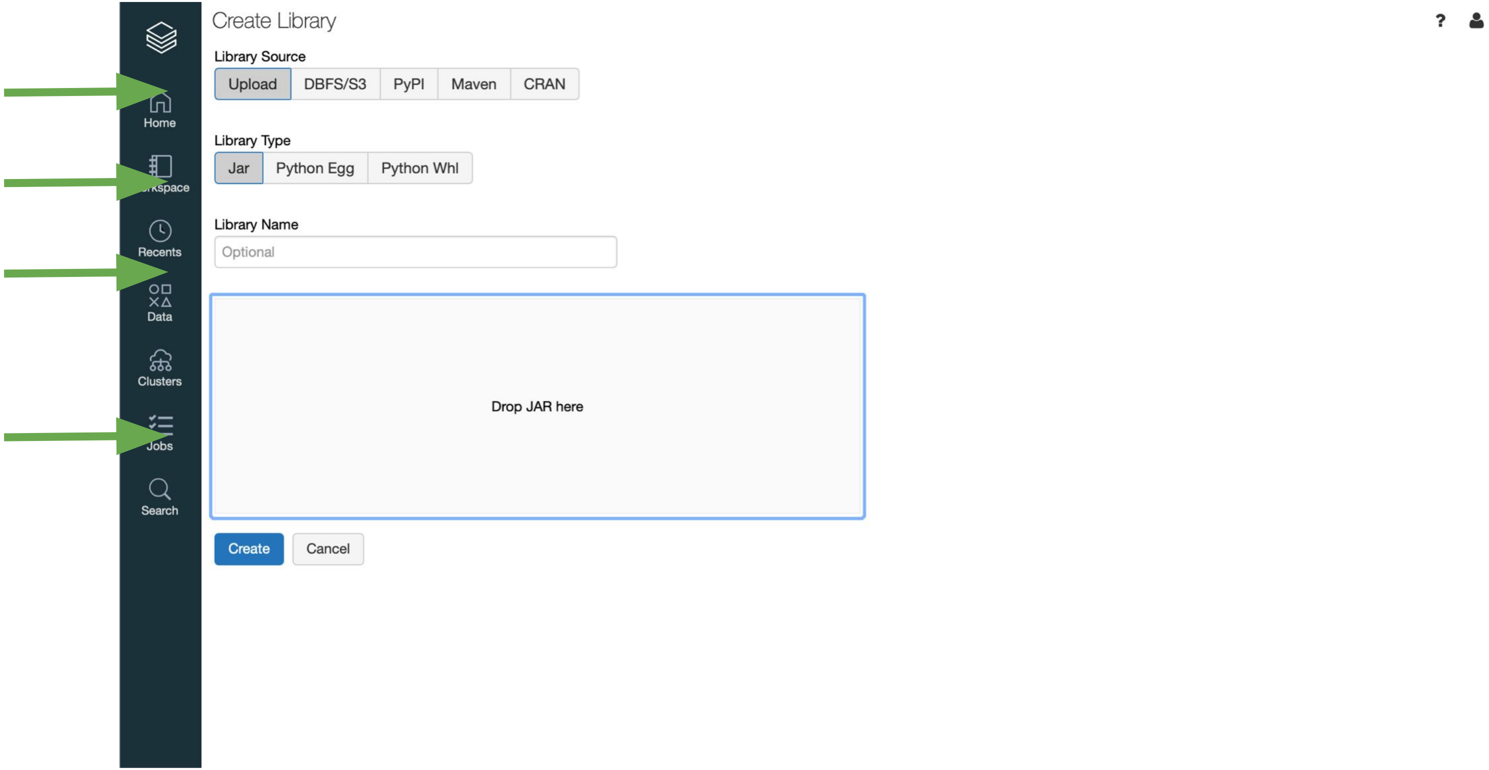
Recents

- DistRDF2ML

What's new in v3.44

View latest release notes

Setup Databricks - Upload needed Data - jar



Create Library

Library Source

Upload | DBFS/S3 | PyPI | Maven | CRAN

Library Type

Jar | Python Egg | Python Whl

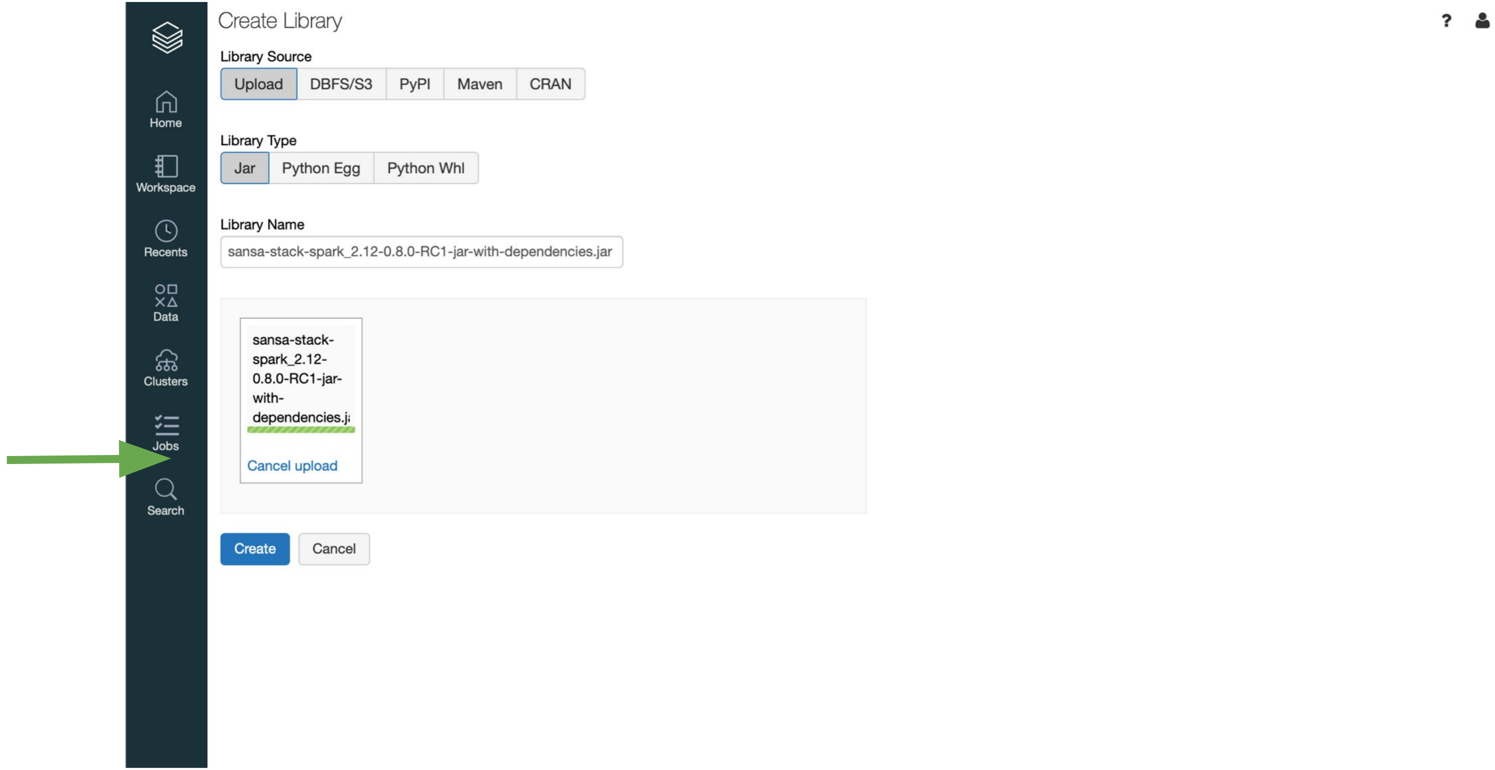
Library Name

Optional

Drop JAR here

Create | Cancel

Setup Databricks - Upload needed Data - jar



Create Library

Library Source: Upload | DBFS/S3 | PyPI | Maven | CRAN

Library Type: Jar | Python Egg | Python Whl

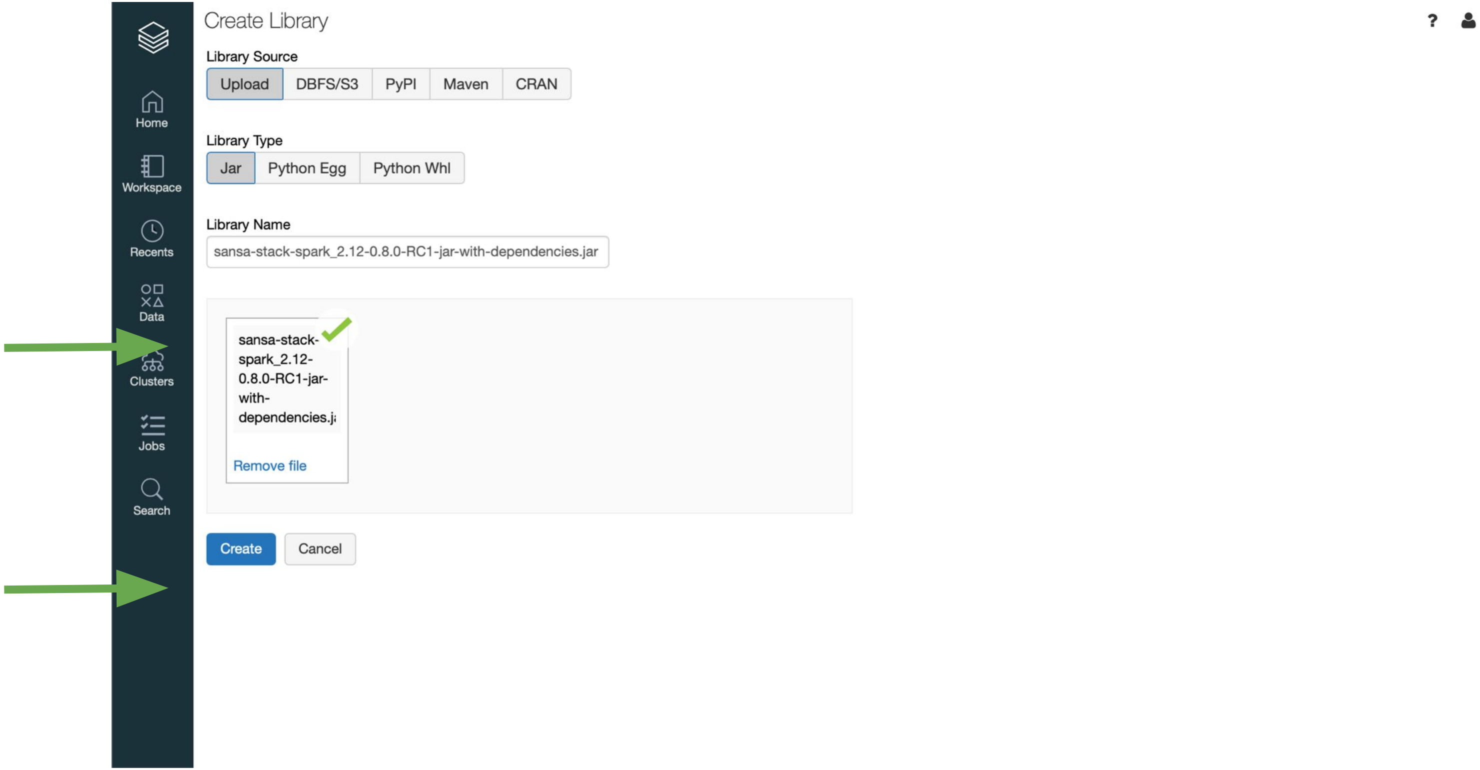
Library Name: sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.jar

sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.j

Cancel upload

Create | Cancel

Setup Databricks - Upload needed Data - jar



Create Library

Library Source

Upload | DBFS/S3 | PyPI | Maven | CRAN

Library Type

Jar | Python Egg | Python Whl

Library Name

sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.jar

sansa-stack-spark_2.12-0.8.0-RC1-jar-with-dependencies.j

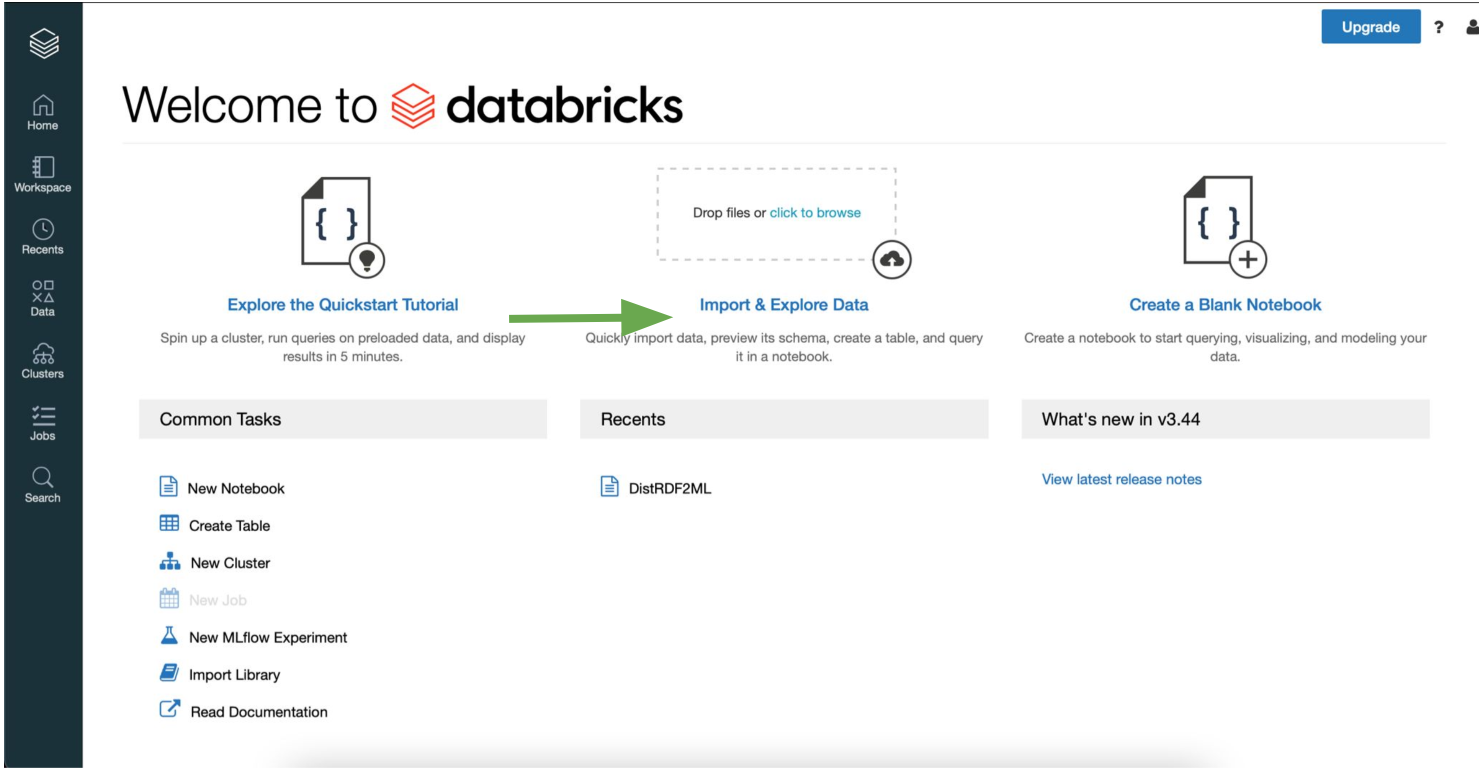
Remove file

Create | Cancel

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



Welcome to databricks

Drop files or click to browse

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

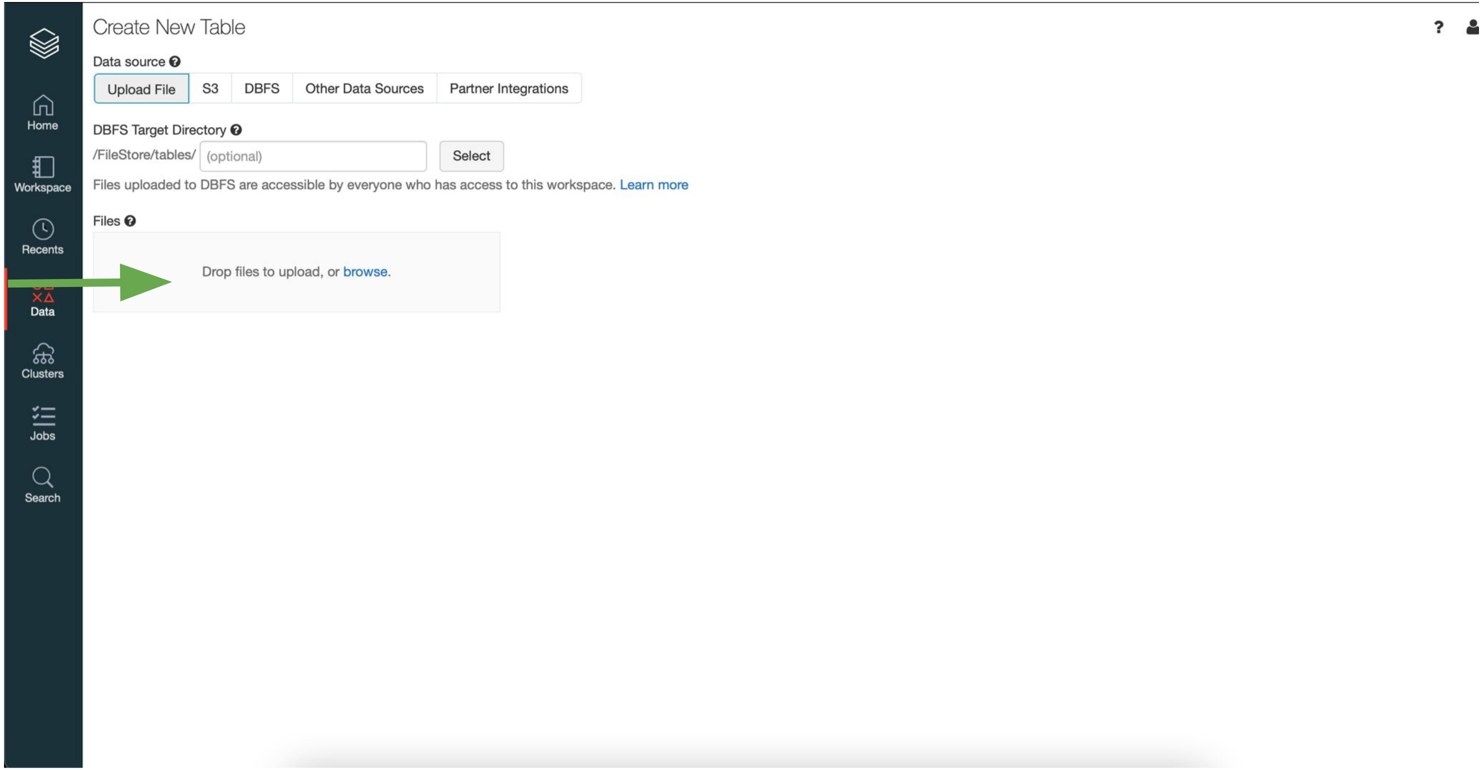
Recents

- 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

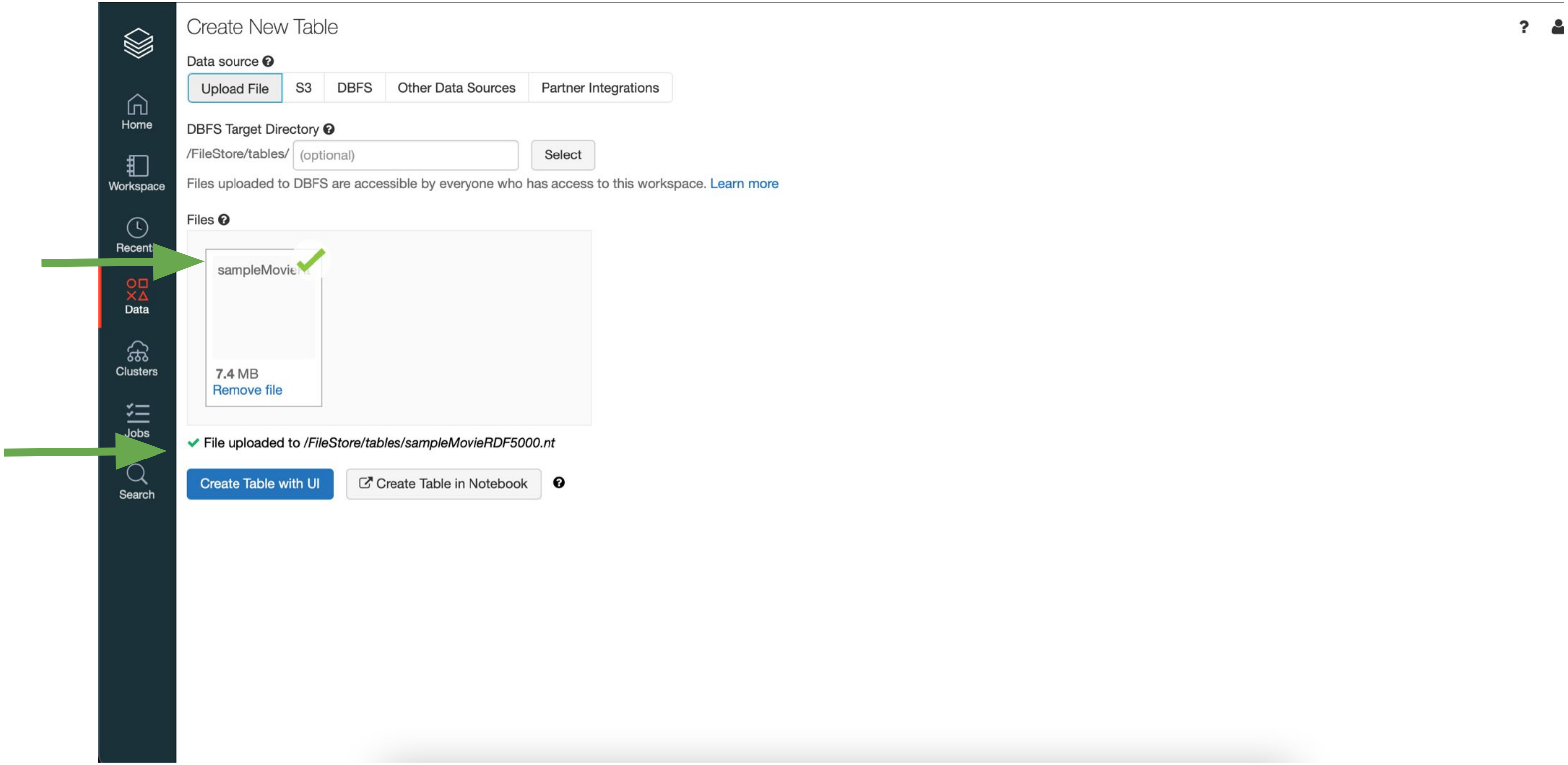
/FileStore/tables/ (optional) Select

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



Create New Table

Data source

DBFS Target Directory

/FileStore/tables/ (optional)

Files uploaded to DBFS are accessible by everyone who has access to this workspace. Learn more

Files

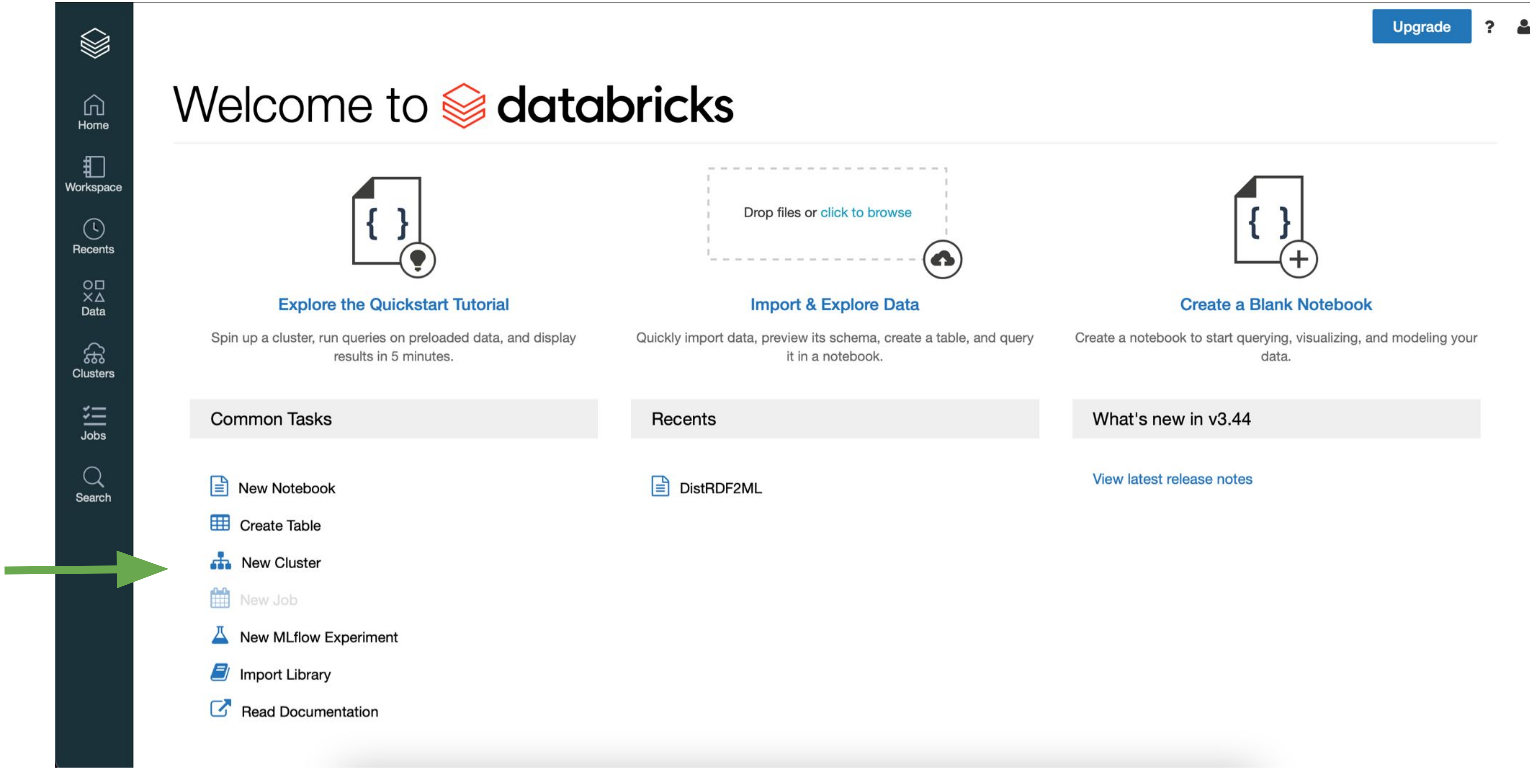
7.4 MB

✓ File uploaded to */FileStore/tables/sampleMovieRDF5000.nt*

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



Welcome to databricks

Drop files or click to browse

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

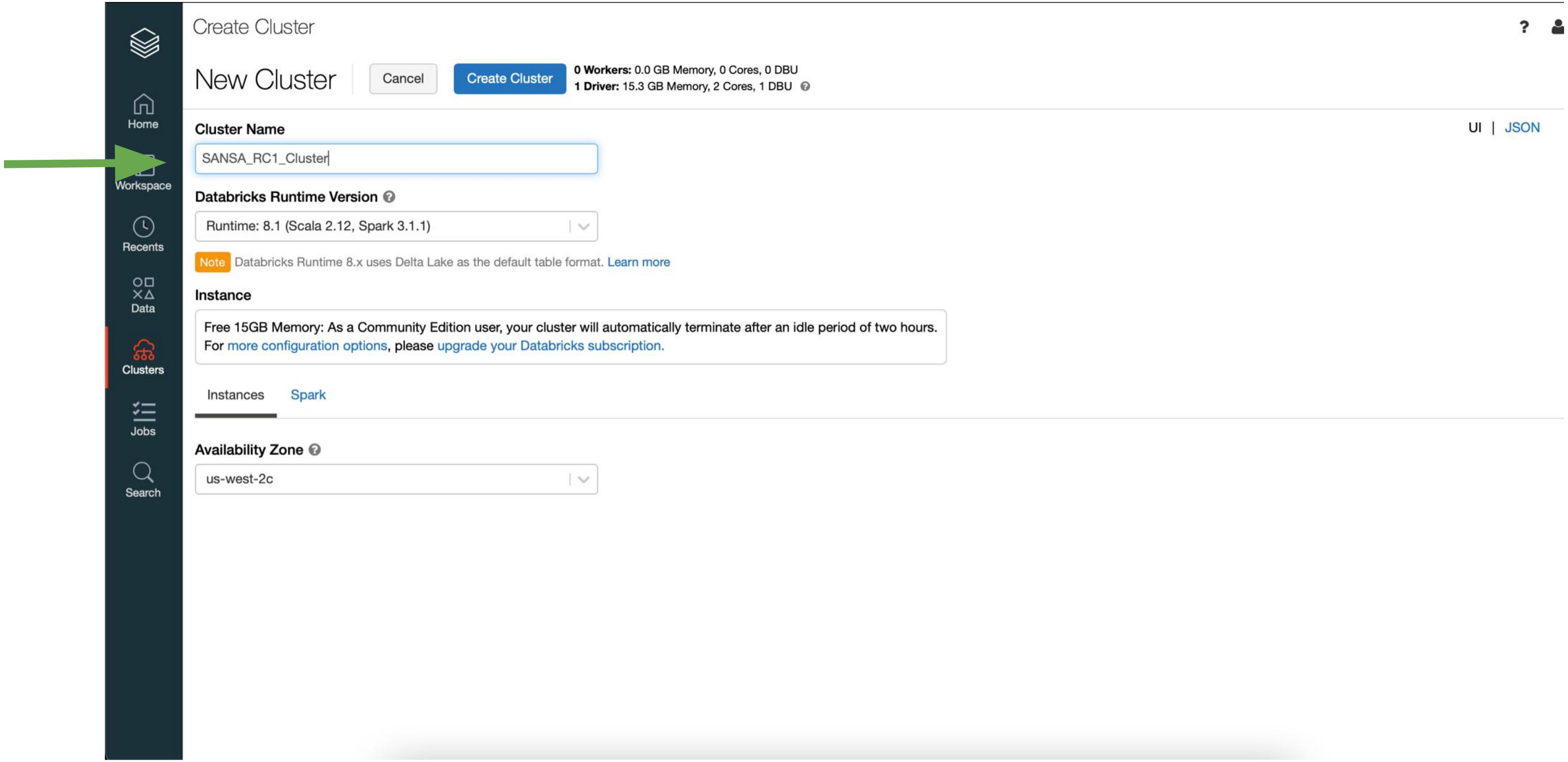
Recents

- DistRDF2ML

What's new in v3.44

View latest release notes

Setup Cluster - Give it a Name



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

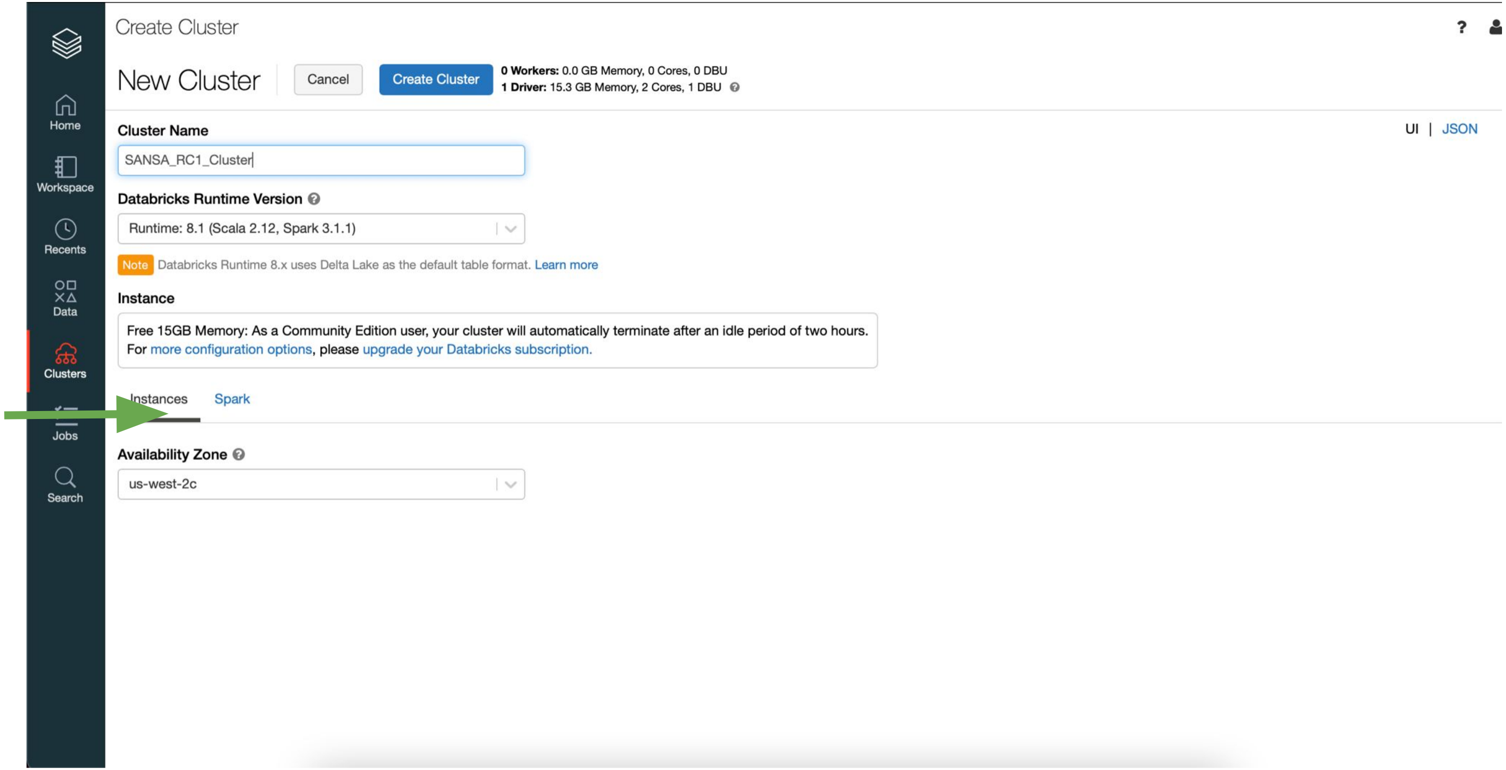
Instance

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.

Availability Zone

us-west-2c

Setup Cluster - Set Spark



New 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

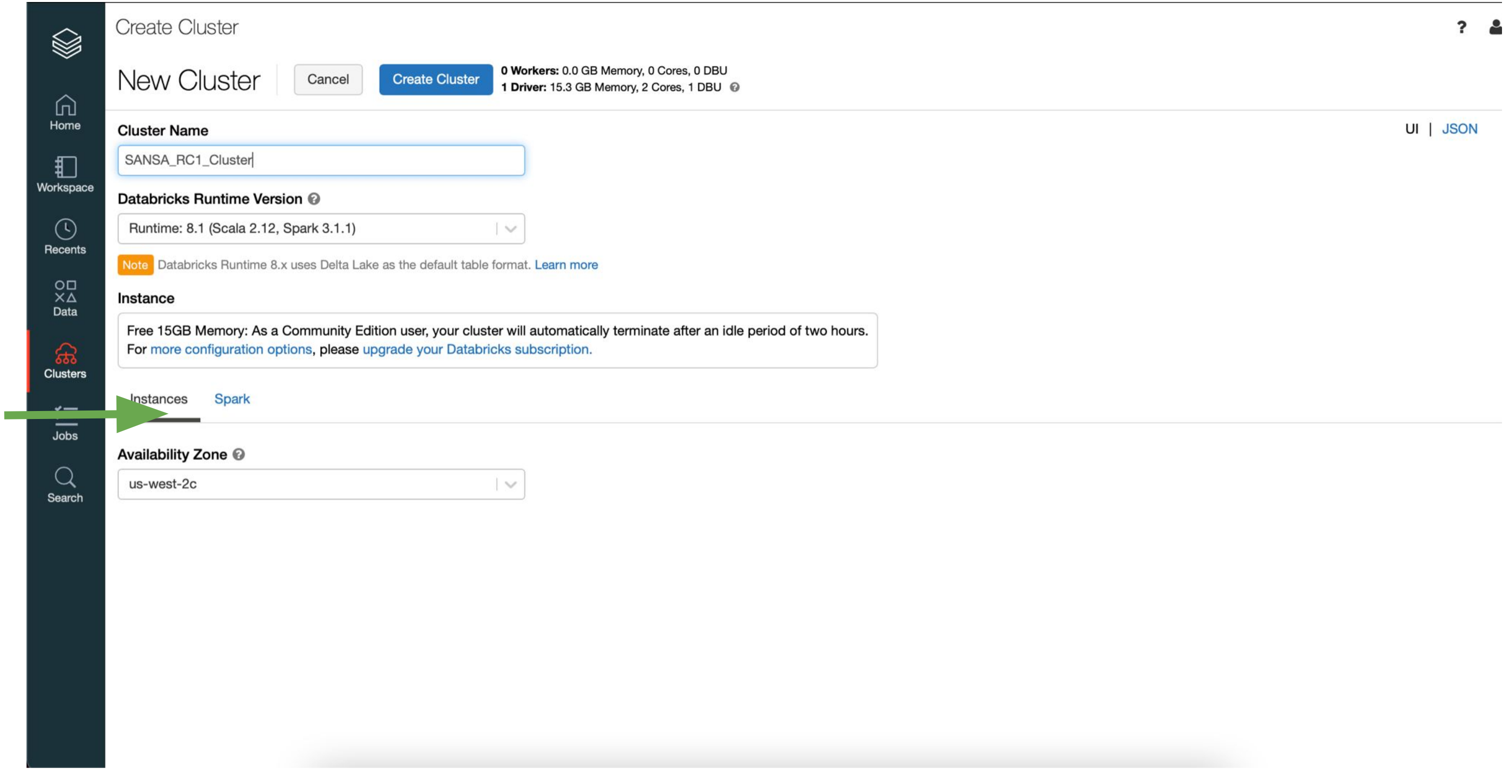
Instance

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.

Availability Zone

us-west-2c

Setup Cluster - Set Spark



New 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

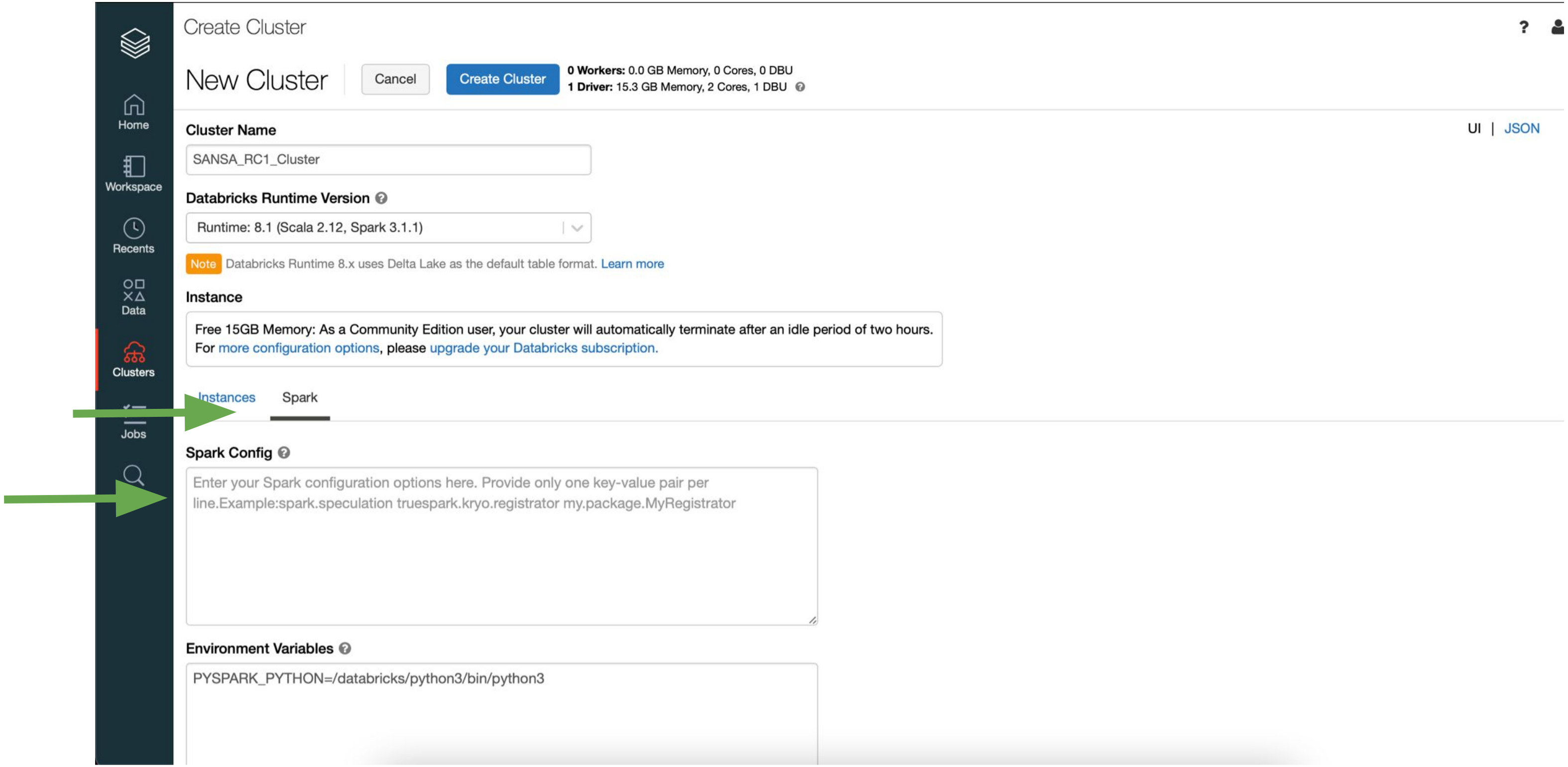
Instance

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.

Availability Zone

us-west-2c

Setup Cluster - Set Spark



Create Cluster

New 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.

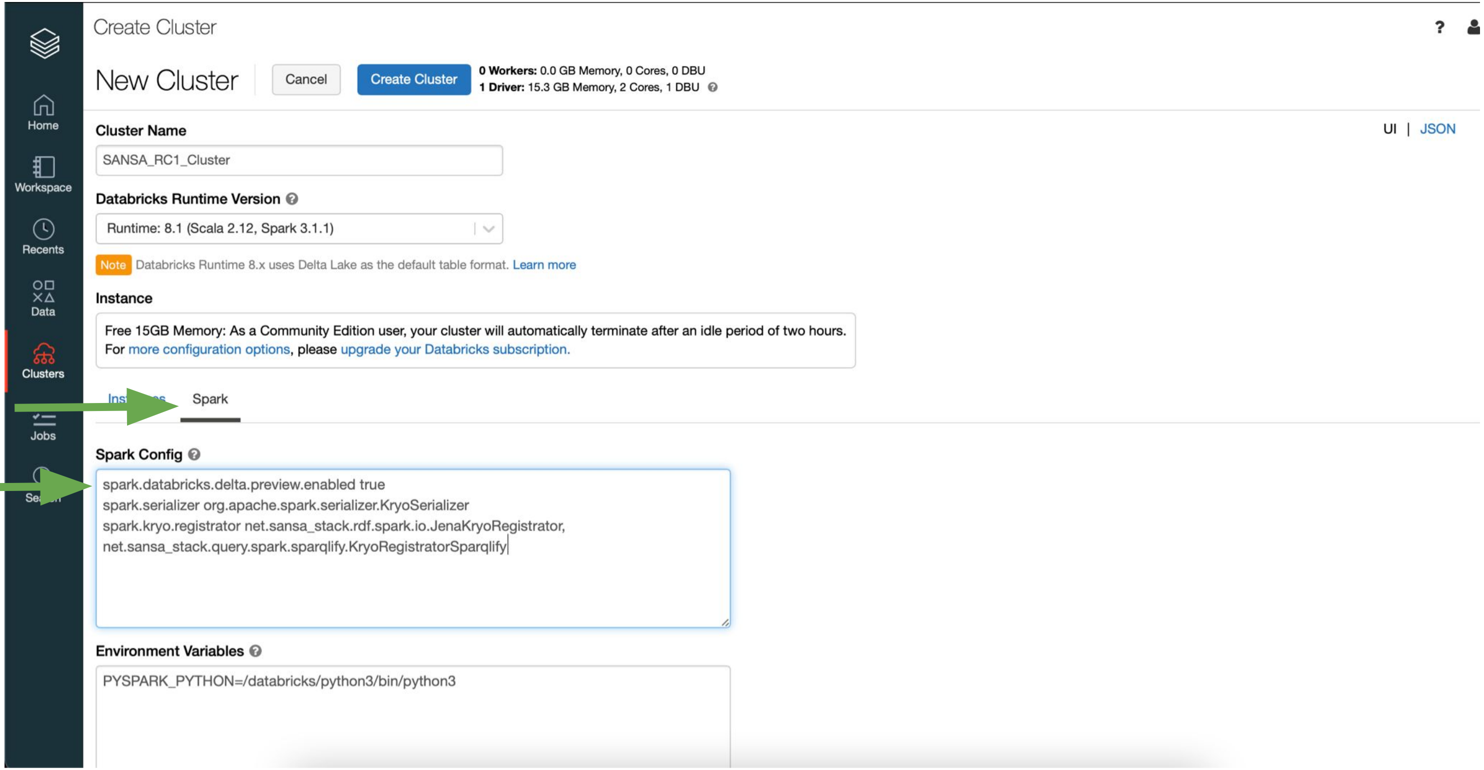
Spark Config

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

Environment Variables

PYSPARK_PYTHON=/databricks/python3/bin/python3

Setup Cluster - Set Spark



Create Cluster

New 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.

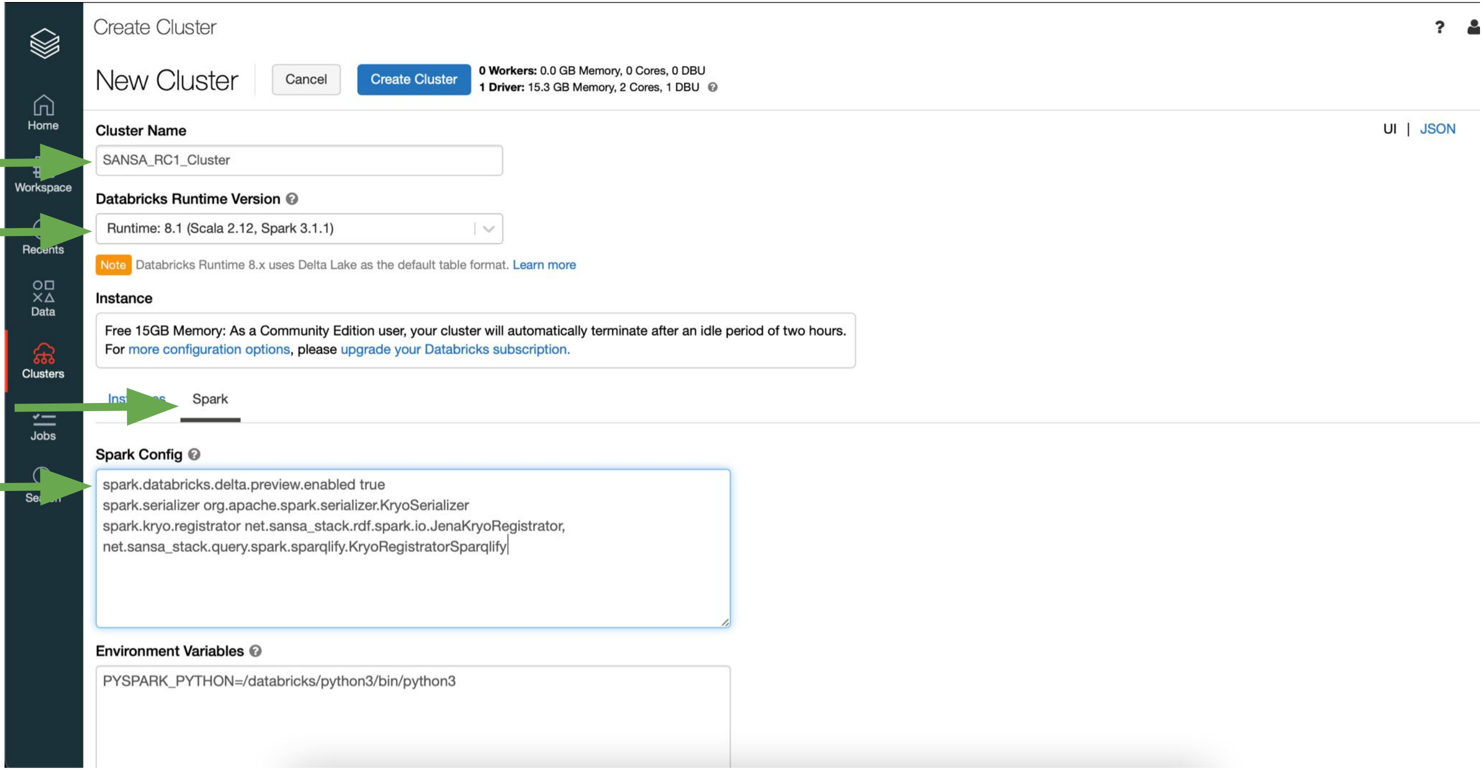
Spark Config

```
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



New 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.

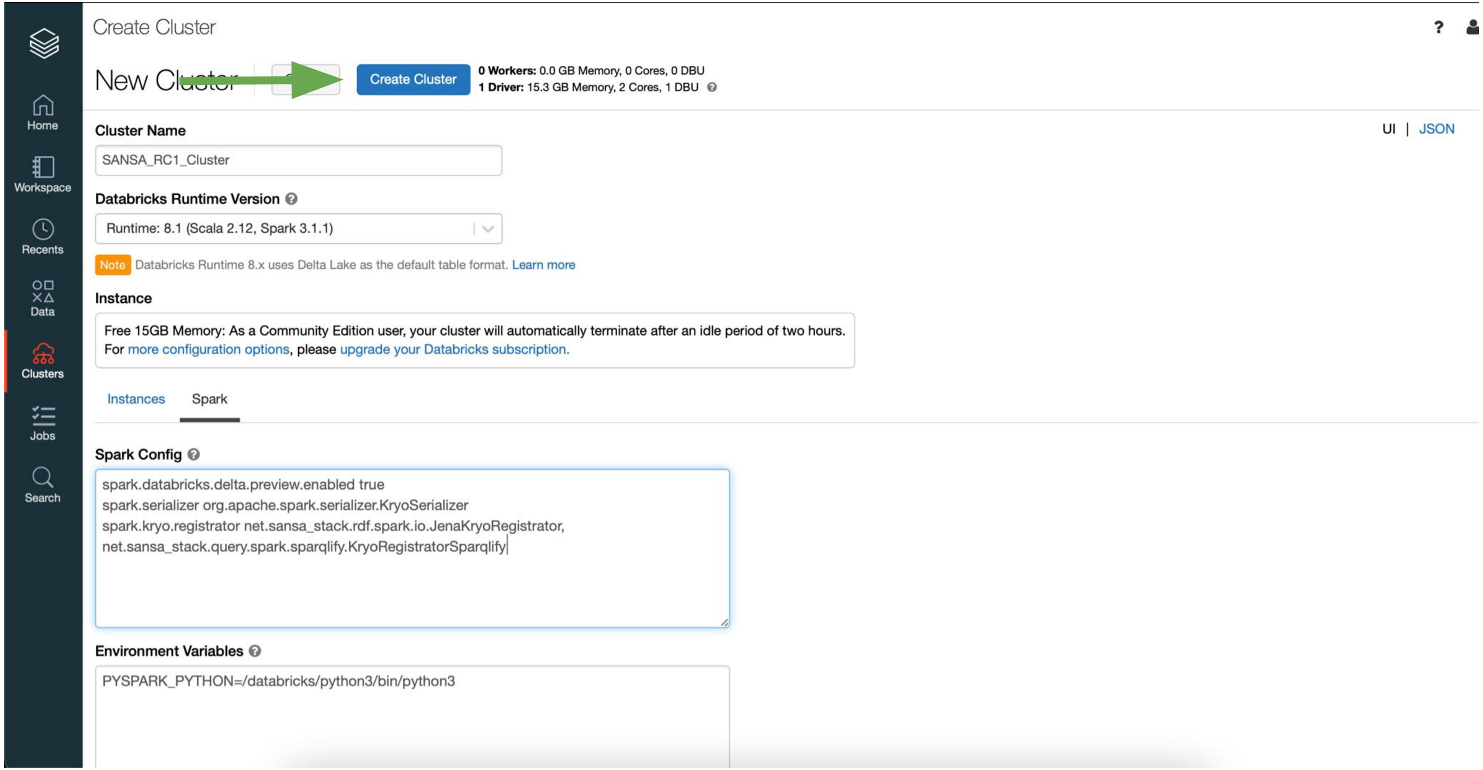
Spark Config

```
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 - 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.

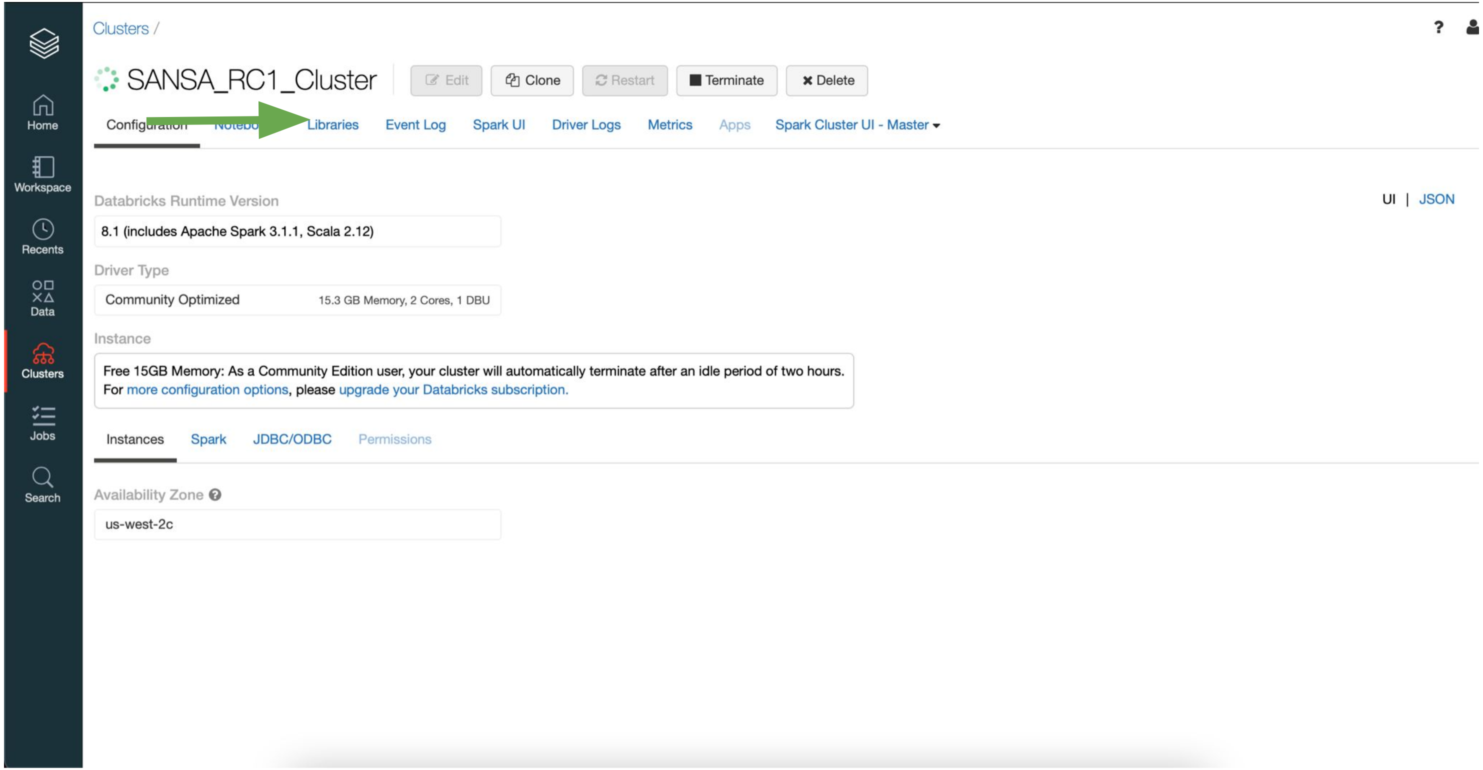
Spark Config

```
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 SANSA JAR



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

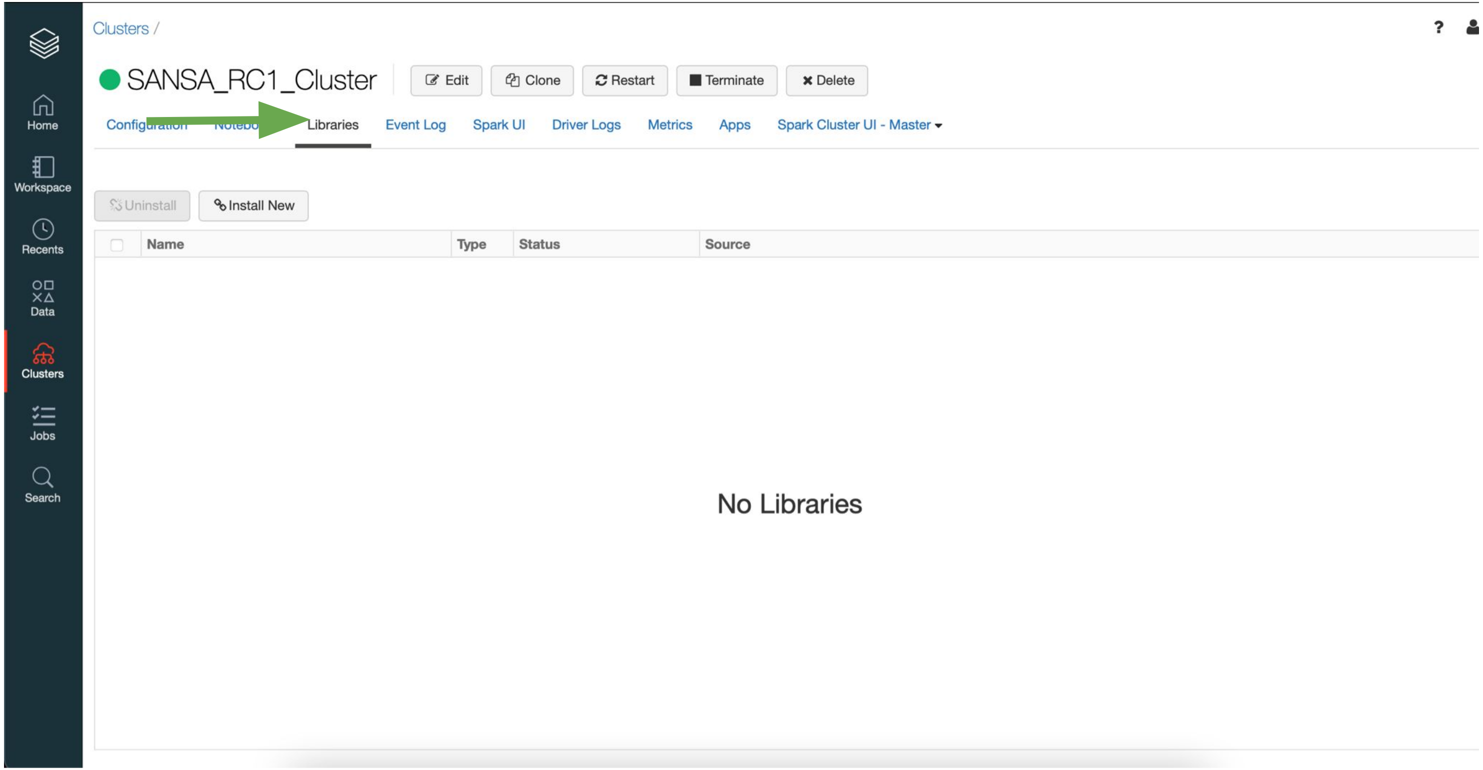
Instance

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.

Availability Zone

us-west-2c

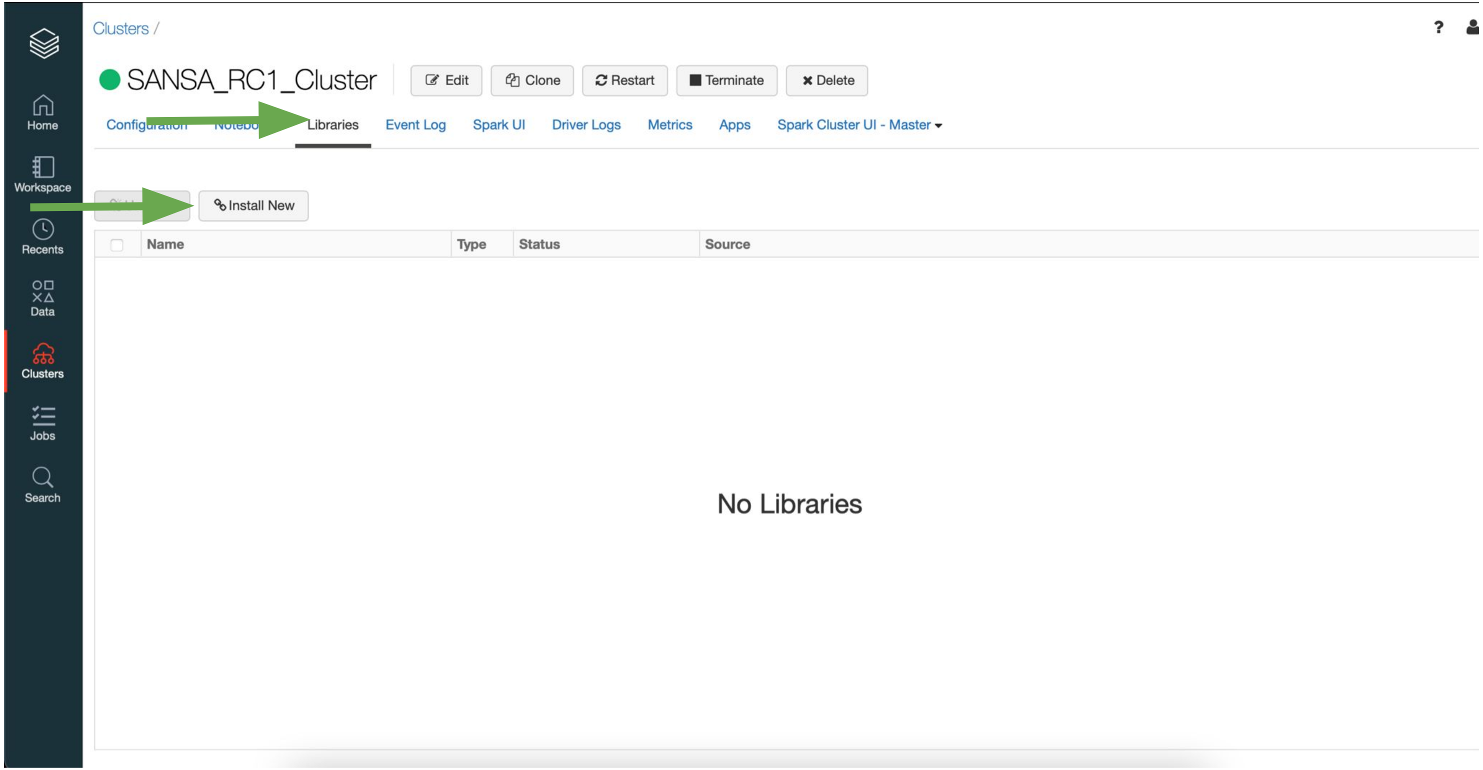
Setup Cluster - set SANSA JAR



□	Name	Type	Status	Source

No Libraries

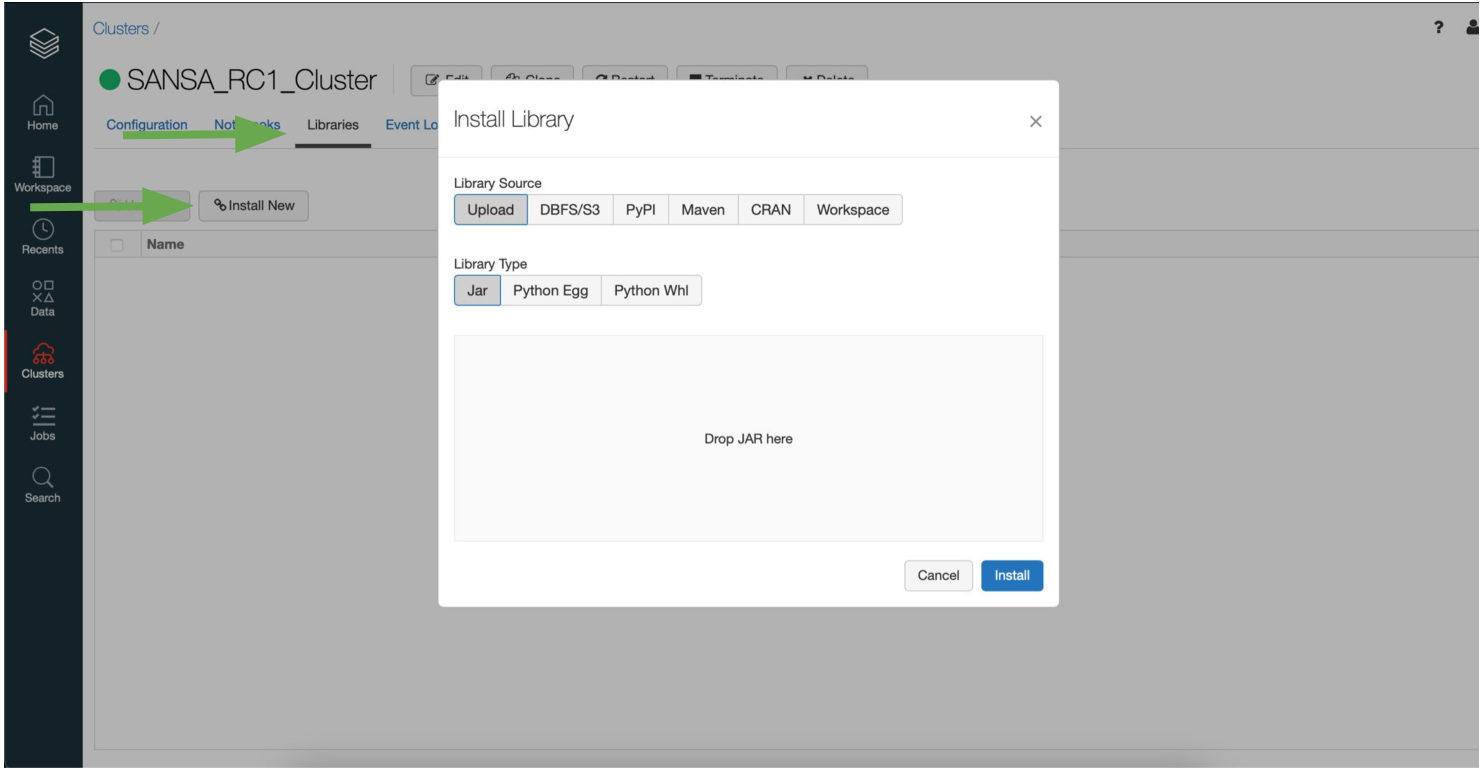
Setup Cluster - set SANSA JAR



	Name	Type	Status	Source

No Libraries

Setup Cluster - set SANSA JAR



Install Library

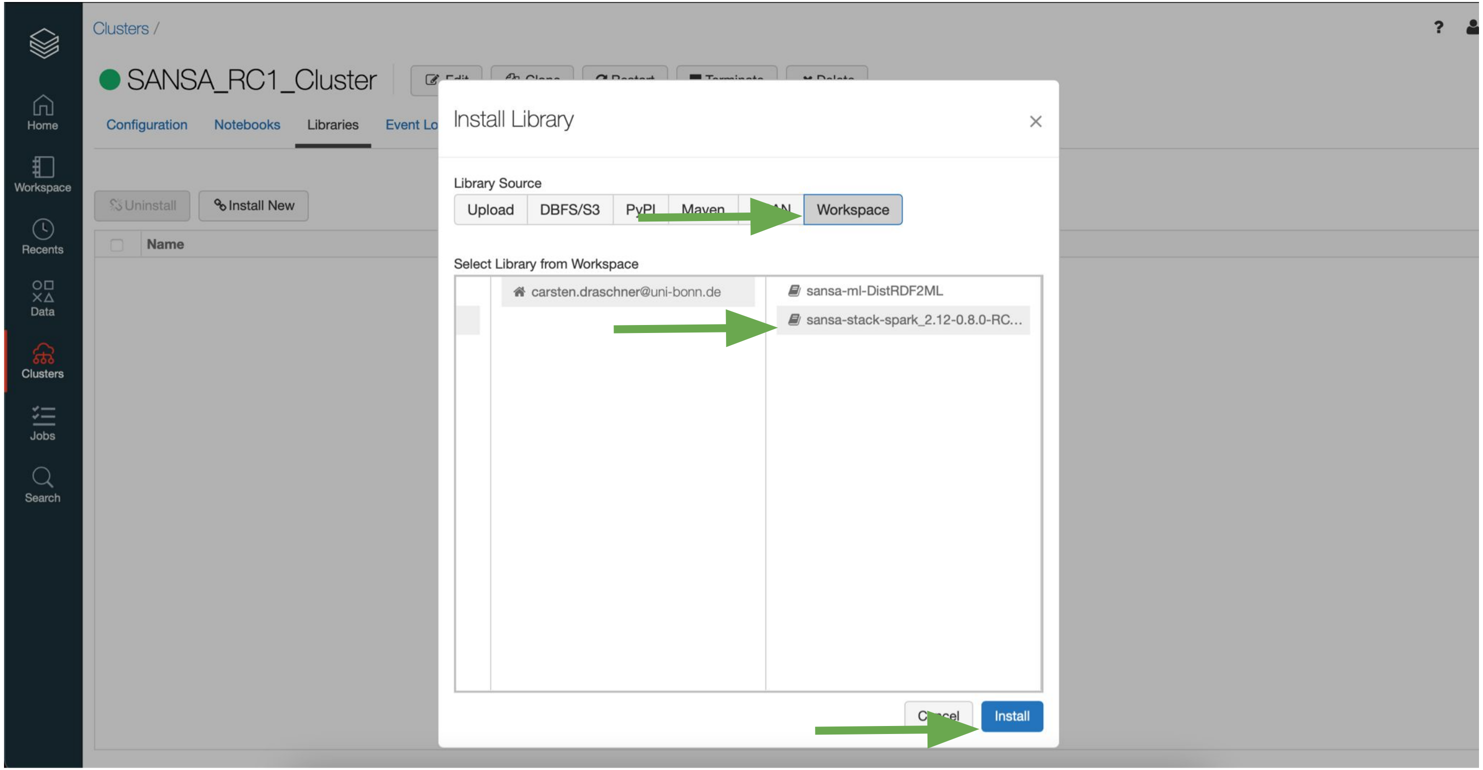
Library Source: Upload | DBFS/S3 | PyPI | Maven | CRAN | Workspace

Library Type: Jar | Python Egg | Python Whl

Drop JAR here

Cancel | Install

Setup Cluster - set SANSA JAR

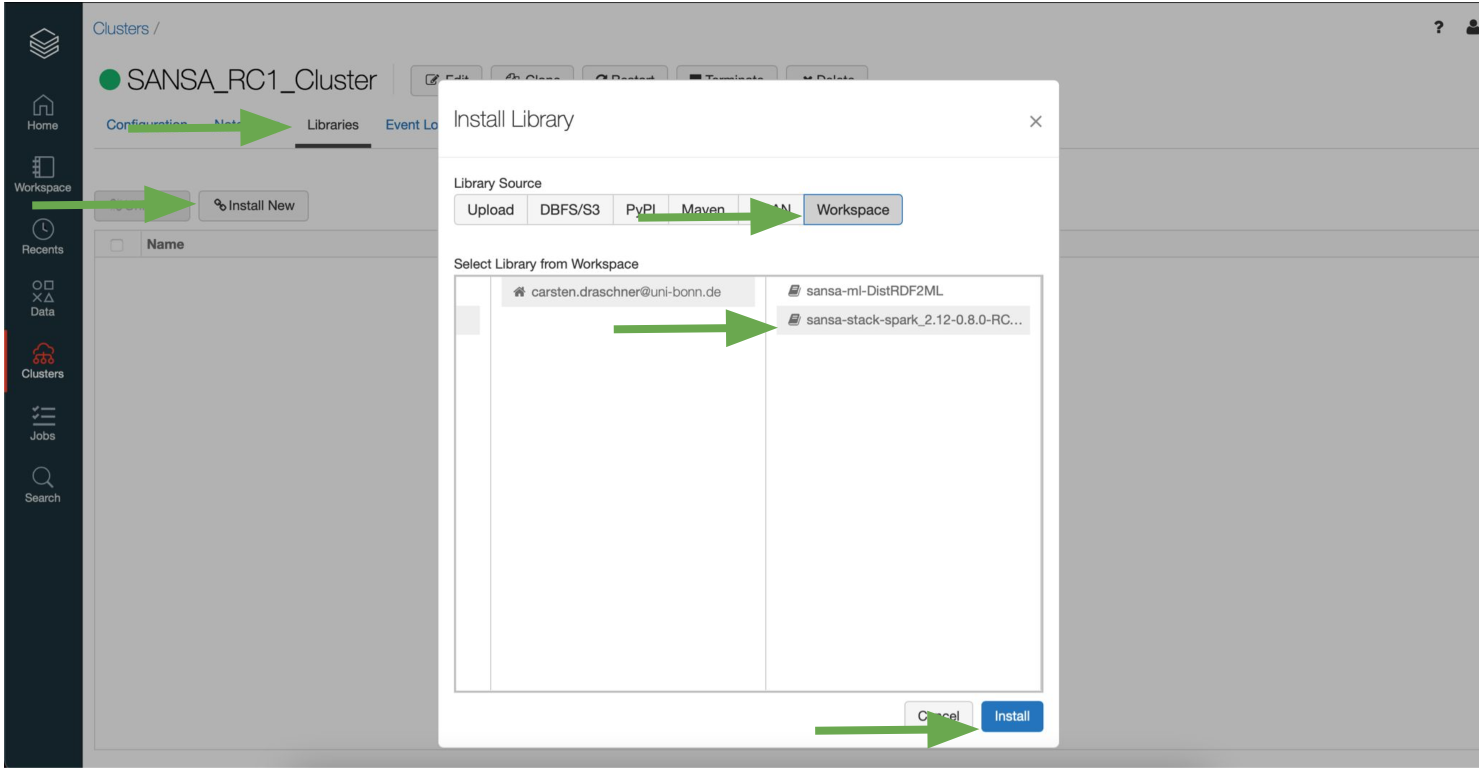


Library Source: Workspace

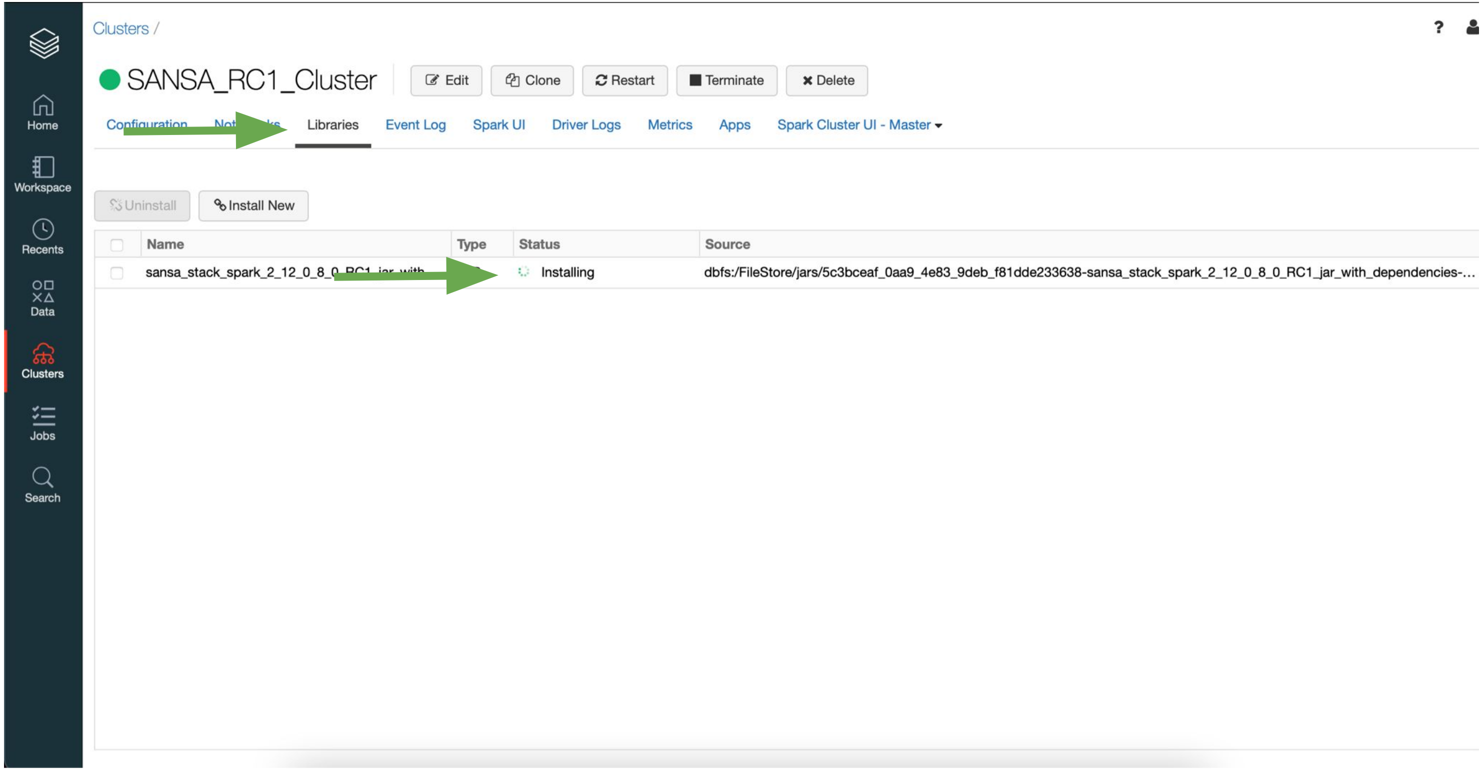
Select Library from Workspace

- carsten.draschner@uni-bonn.de
- sansa-ml-DistRDF2ML
- sansa-stack-spark_2.12-0.8.0-RC...

Setup Cluster - set SANSA JAR

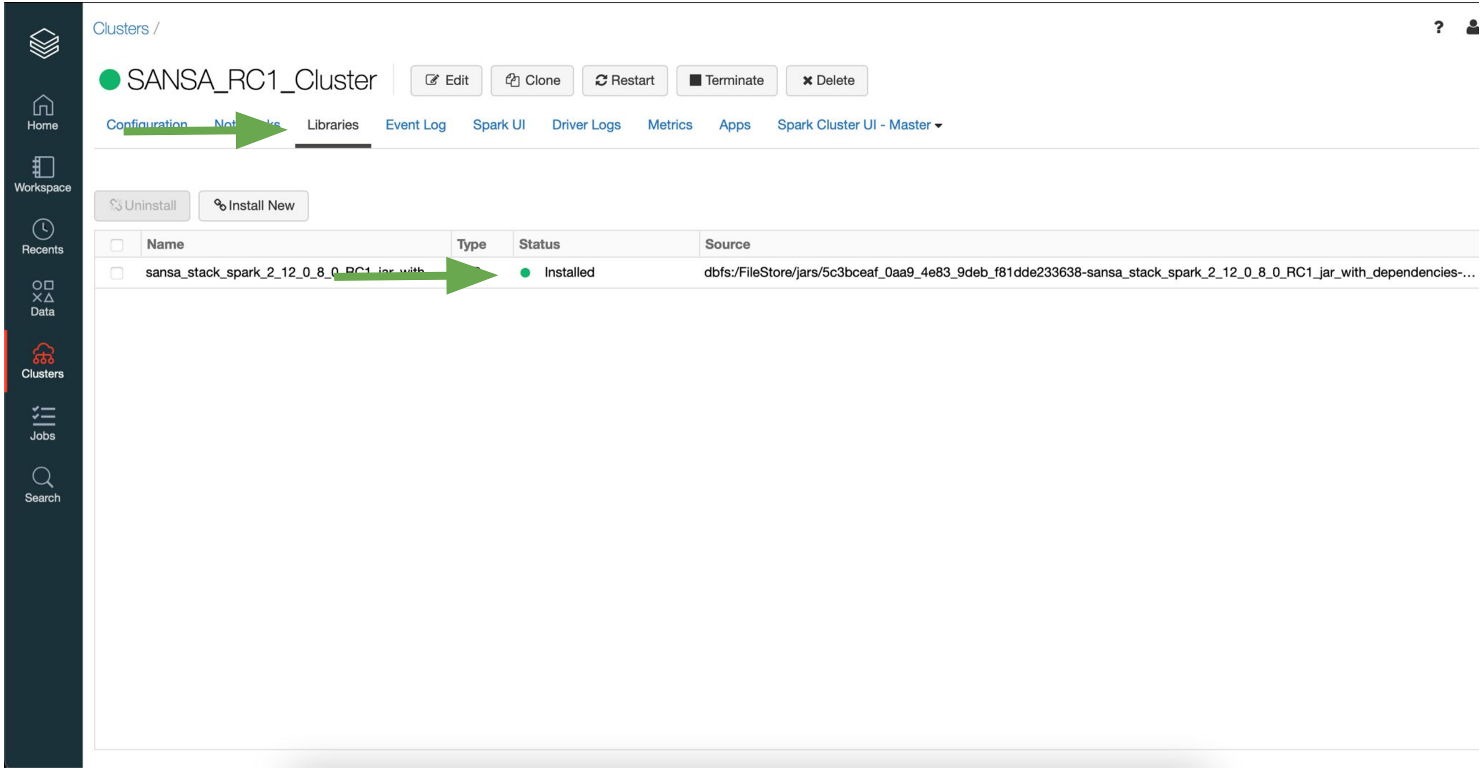


Setup Cluster - set SANSA JAR



☐	Name	Type	Status	Source
☐	sansa_stack_spark_2_12_0_8_0_RC1_jar_with	[illegible]	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

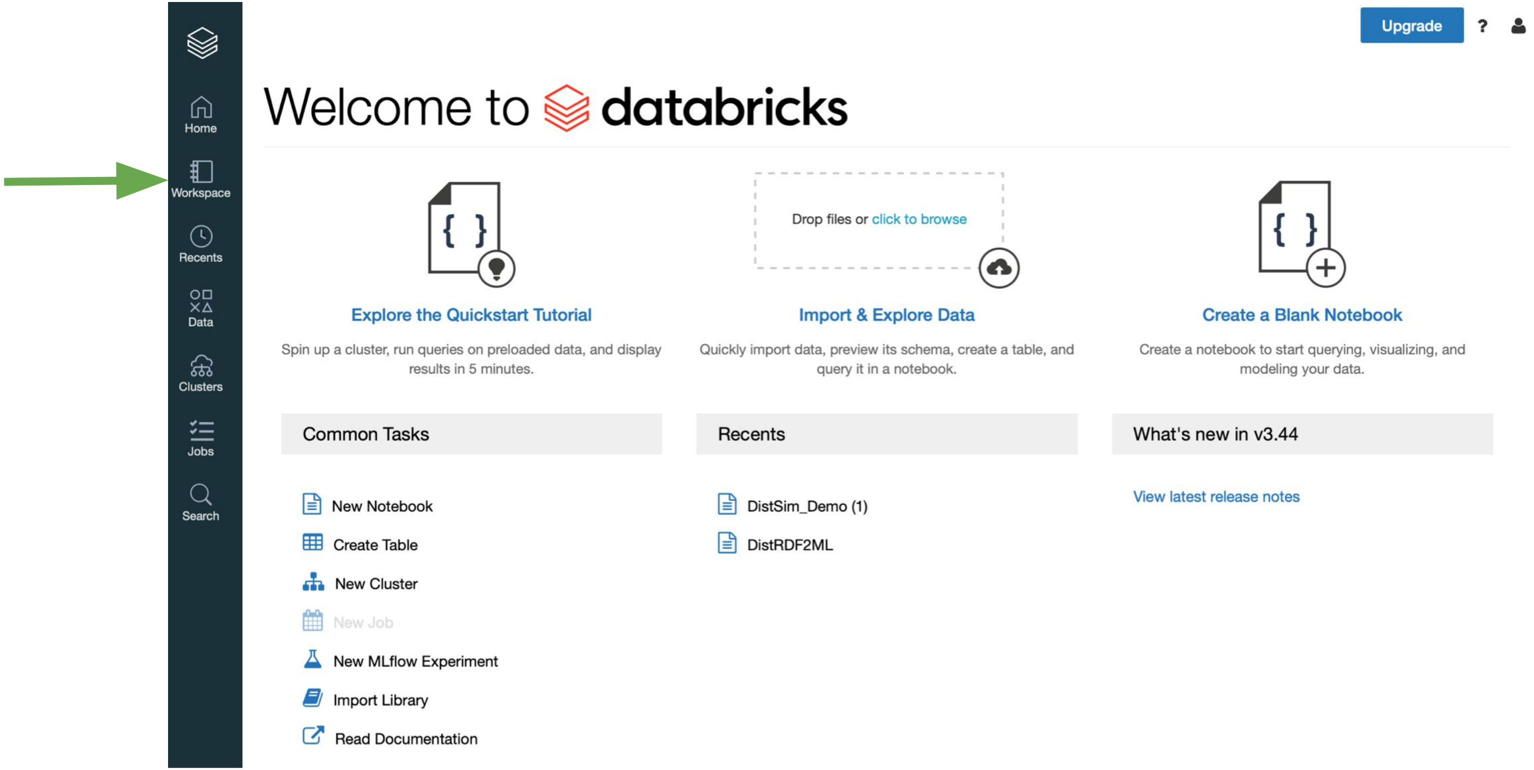


	Name	Type	Status	Source
□	sansa_stack_spark_2_12_0_8_0_RC1_jar_with		● Installed	dbfs:/FileStore/jars/5c3bceaf_0aa9_4e83_9deb_f81dde233638-sansa_stack_spark_2_12_0_8_0_RC1_jar_with_dependencies-…

Import Notebook

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

Import Notebook



Welcome to databricks

Drop files or click to browse

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

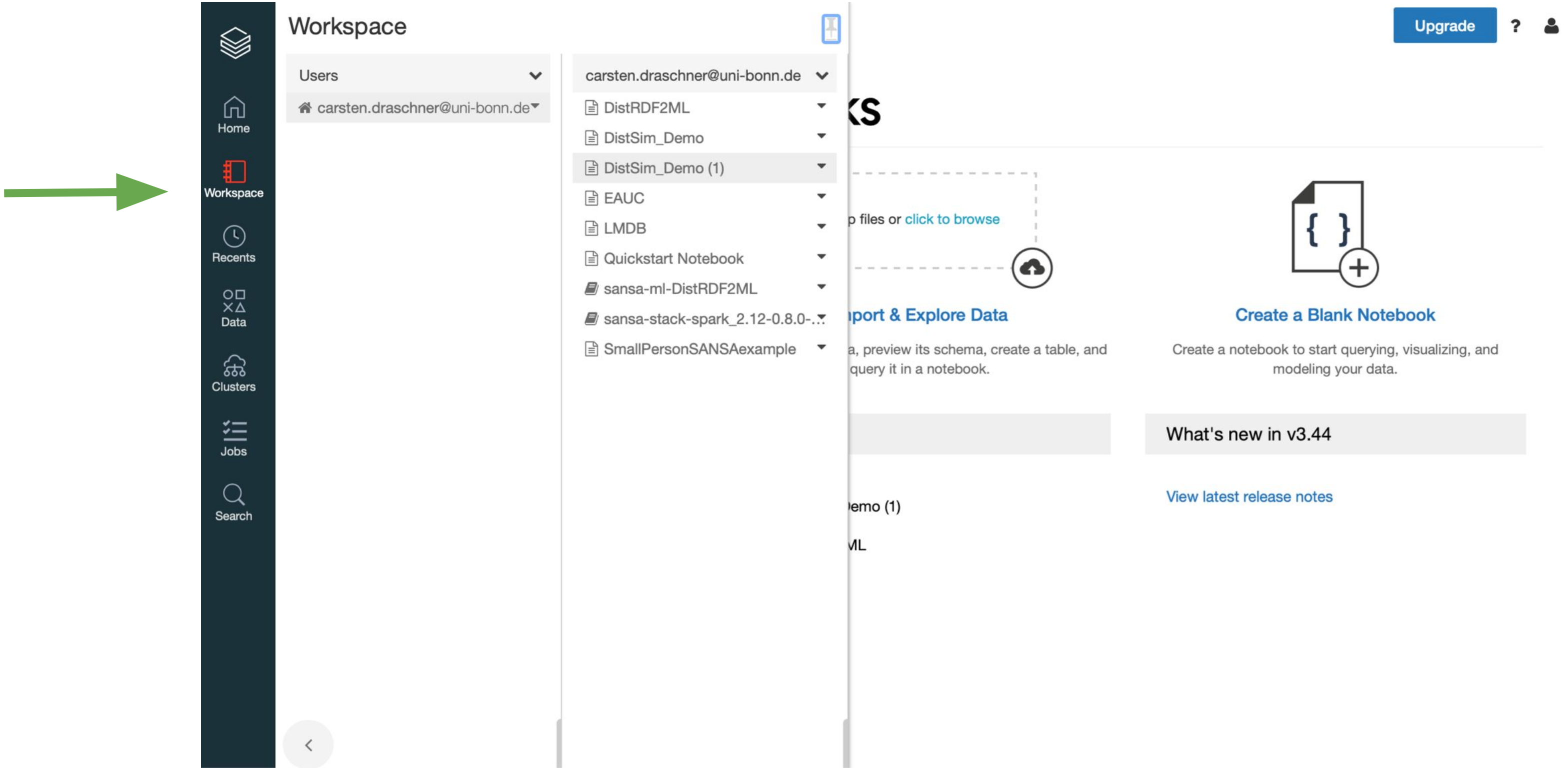
Recents

- DistSim_Demo (1)
- DistRDF2ML

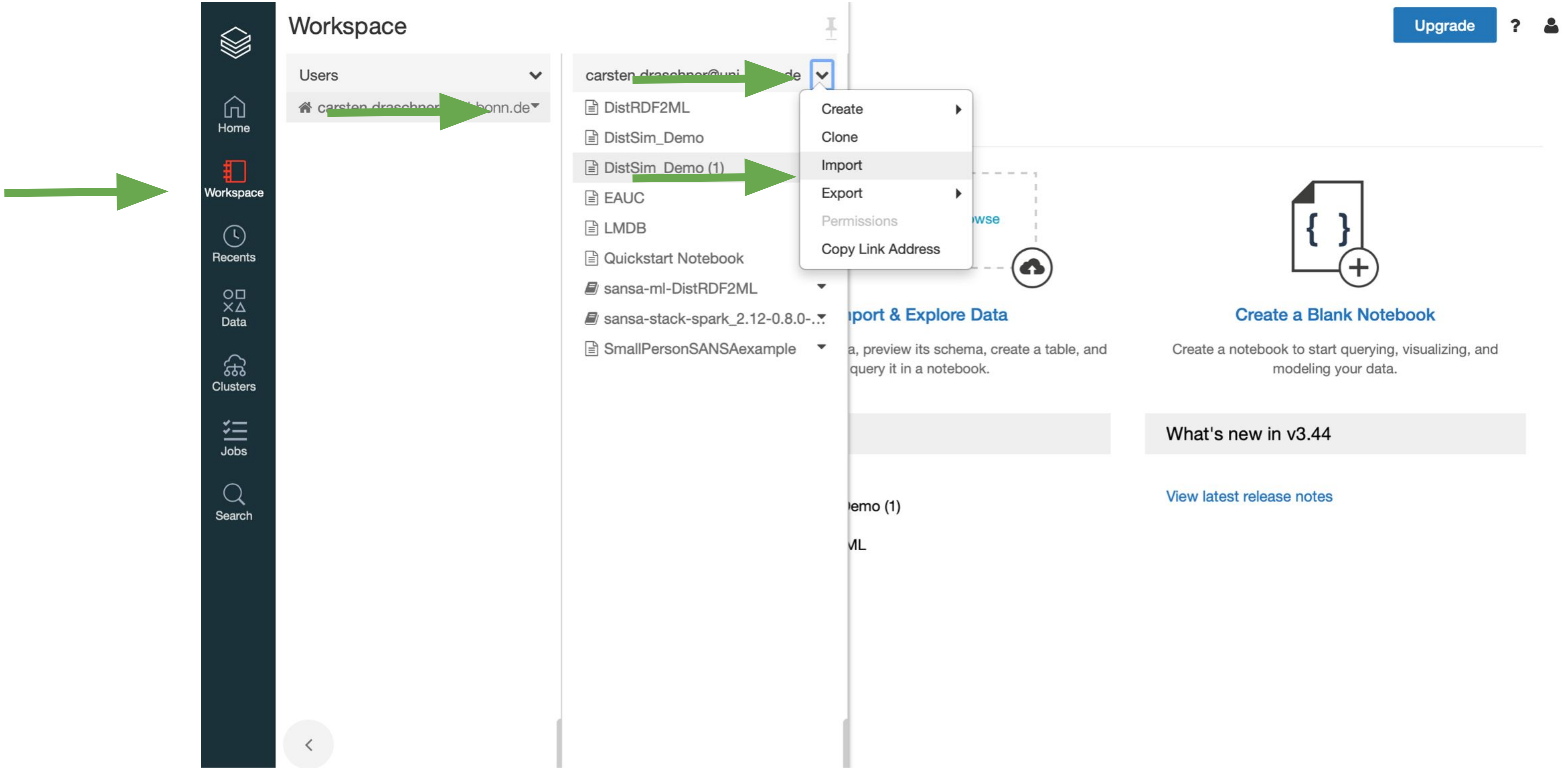
What's new in v3.44

View latest release notes

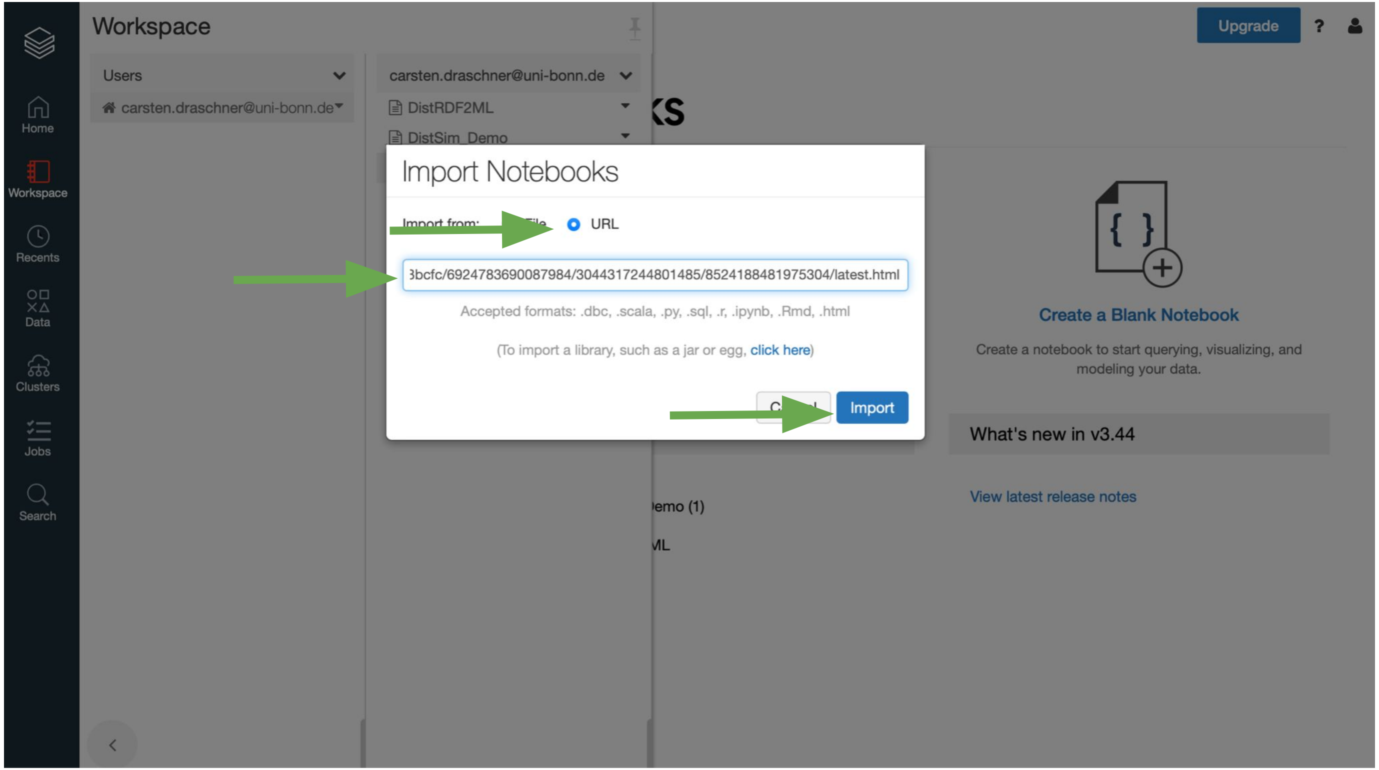
Import Notebook


Workspace
Users
carsten.draschner@uni-bonn.de
carsten.draschner@uni-bonn.de
DistRDF2ML
DistSim_Demo
DistSim_Demo (1)
EAUC
LMDB
Quickstart Notebook
sansa-ml-DistRDF2ML
sansa-stack-spark_2.12-0.8.0-...
SmallPersonSANSAexample
Home
Workspace
Recents
Data
Clusters
Jobs
Search
Upgrade
?
ks
p files or click to browse
nport & Explore Data
a, 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.
What's new in v3.44
View latest release notes
emo (1)
ML

Import Notebook



Import Notebook



Import Notebooks

Import from: File ● URL

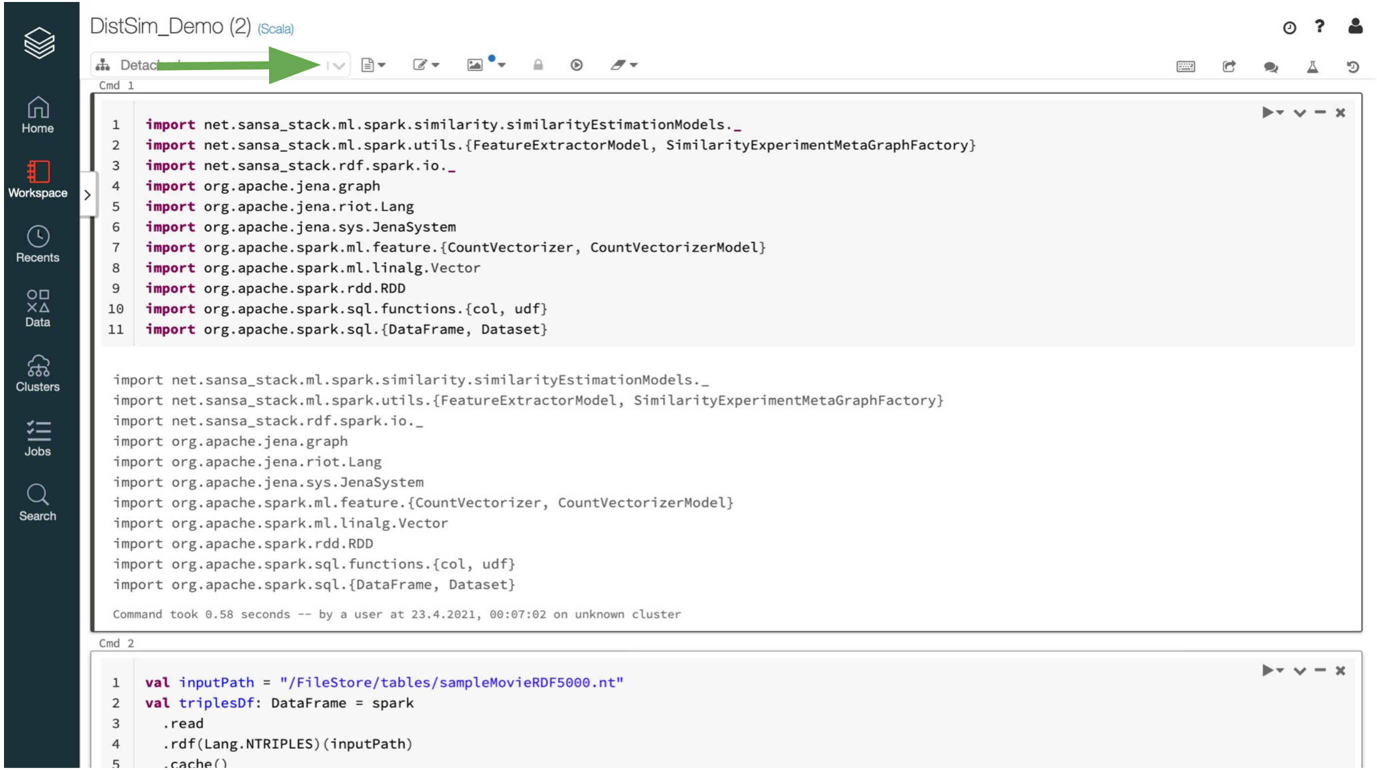
3bcfc/6924783690087984/3044317244801485/8524188481975304/latest.html

Accepted formats: .dbc, .scala, .py, .sql, .r, .ipynb, .Rmd, .html

(To import a library, such as a jar or egg, click here)

Workspace
Users
carsten.draschner@uni-bonn.de
carsten.draschner@uni-bonn.de
DistRDF2ML
DistSim_Demo
Upgrade
Home
Workspace
Recents
Data
Clusters
Jobs
Search
Create a Blank Notebook
Create a notebook to start querying, visualizing, and modeling your data.
What's new in v3.44
View latest release notes
Cancel
Import

Run Notebook - Select Cluster



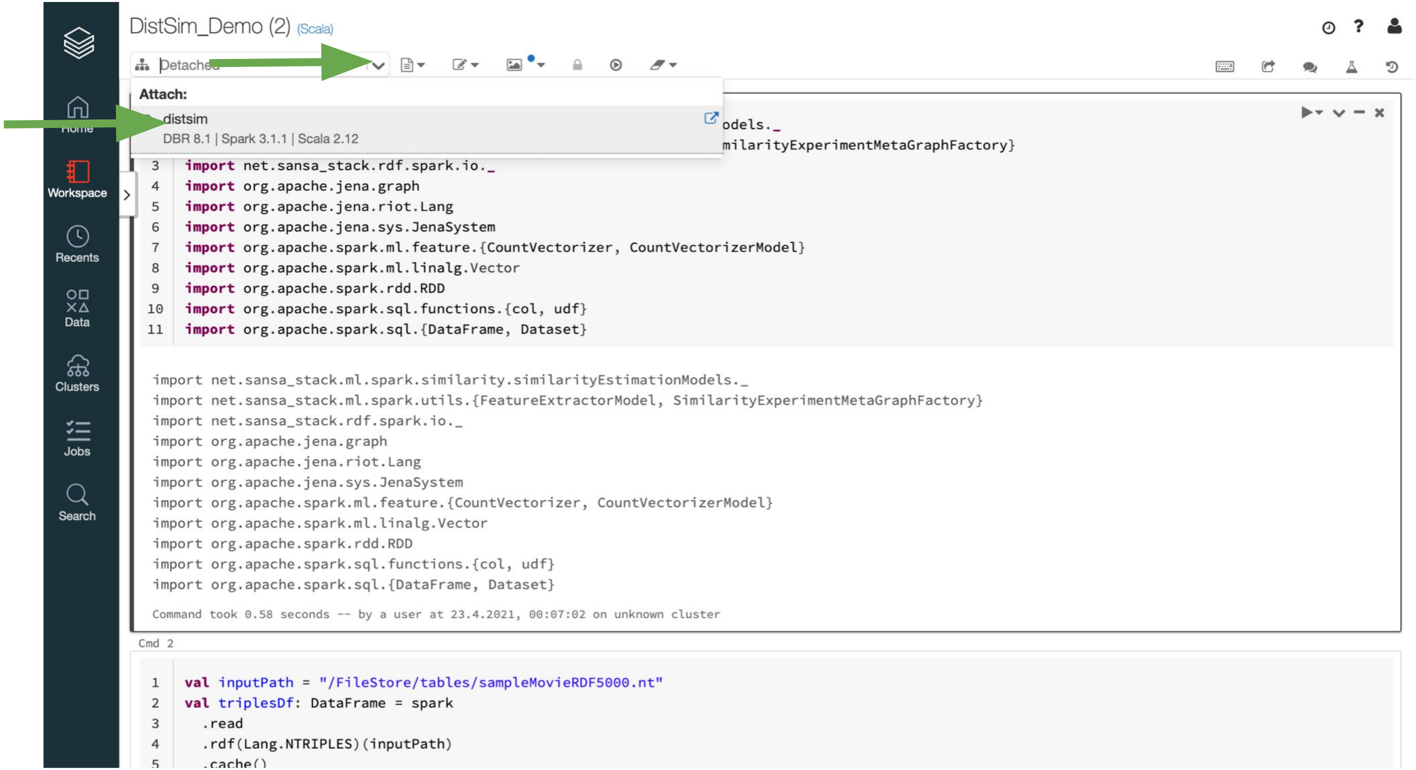
```
import net.sansa_stack.ml.spark.similarity.similarityEstimationModels._
import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
import net.sansa_stack.rdf.spark.io._
import org.apache.jena.graph
import org.apache.jena.riot.Lang
import org.apache.jena.sys.JenaSystem
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}
```

```
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
```

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

Run Notebook - Select Cluster



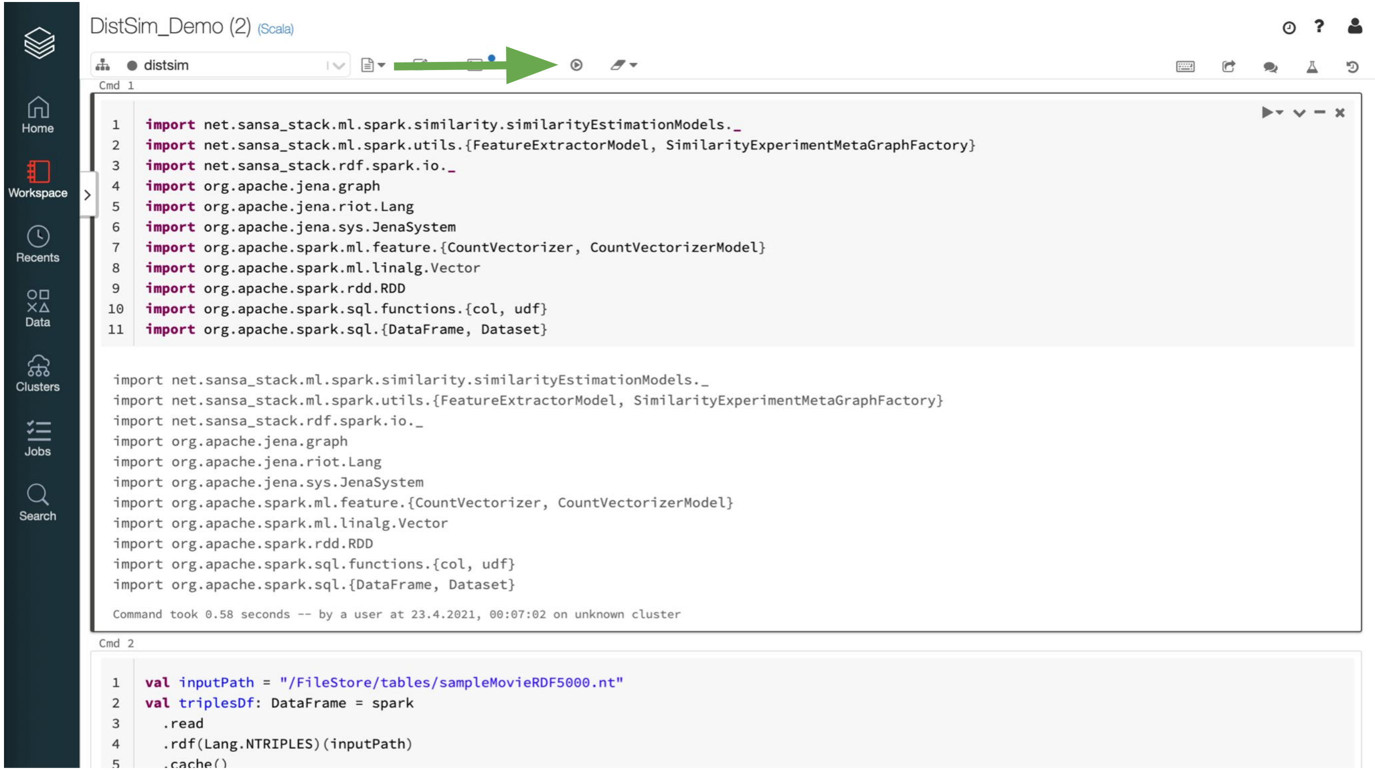
```
odels._
milarityExperimentMetaGraphFactory}
import net.sansa_stack.rdf.spark.io._
import org.apache.jena.graph
import org.apache.jena.riot.Lang
import org.apache.jena.sys.JenaSystem
import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
import org.apache.spark.ml.linalg.Vector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

```
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
```

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

Run Notebook - Run all cells



```
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}
```

```
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

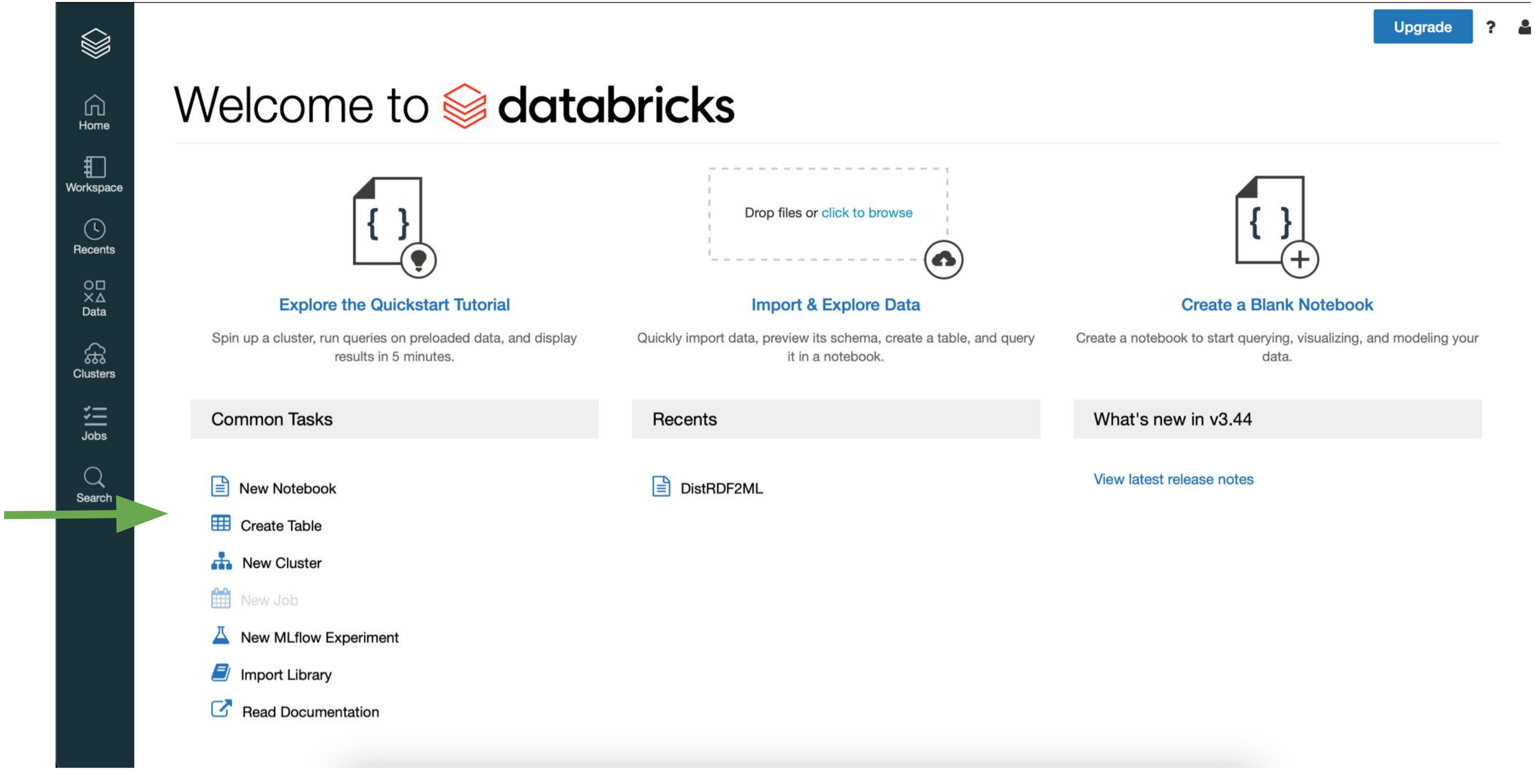
```
val inputPath = "/FileStore/tables/sampleMovieRDF5000.nt"
val triplesDf: DataFrame = spark
  .read
  .rdf(Lang.NTRIPLES)(inputPath)
  .cache()
```

You Made It!!!

Create Notebook

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

Create Notebook - Create Notebook



Upgrade ?

Welcome to databricks

Home
Workspace
Recents
Data
Clusters
Jobs
Search

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

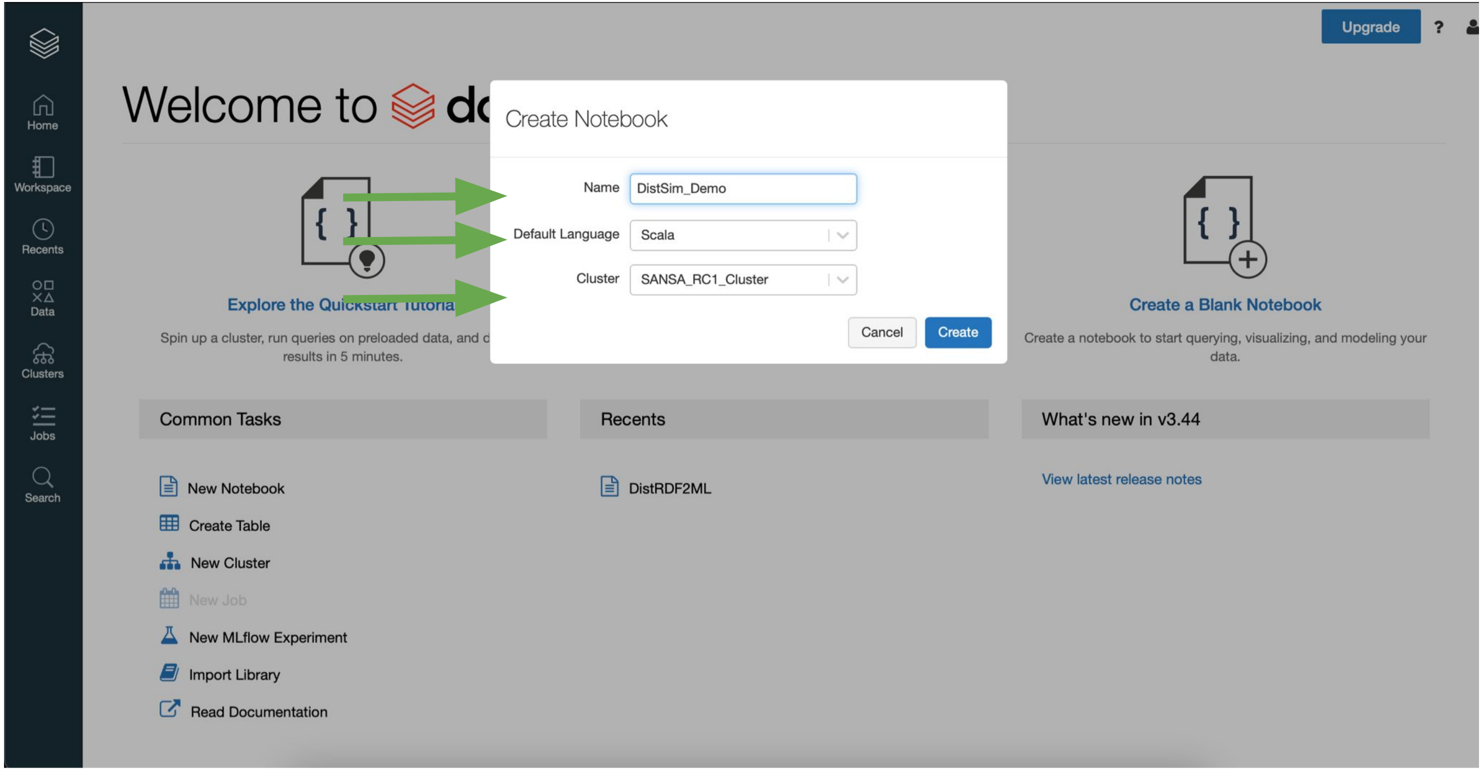
Recents

- DistRDF2ML

What's new in v3.44

View latest release notes

Create Notebook - Create Notebook



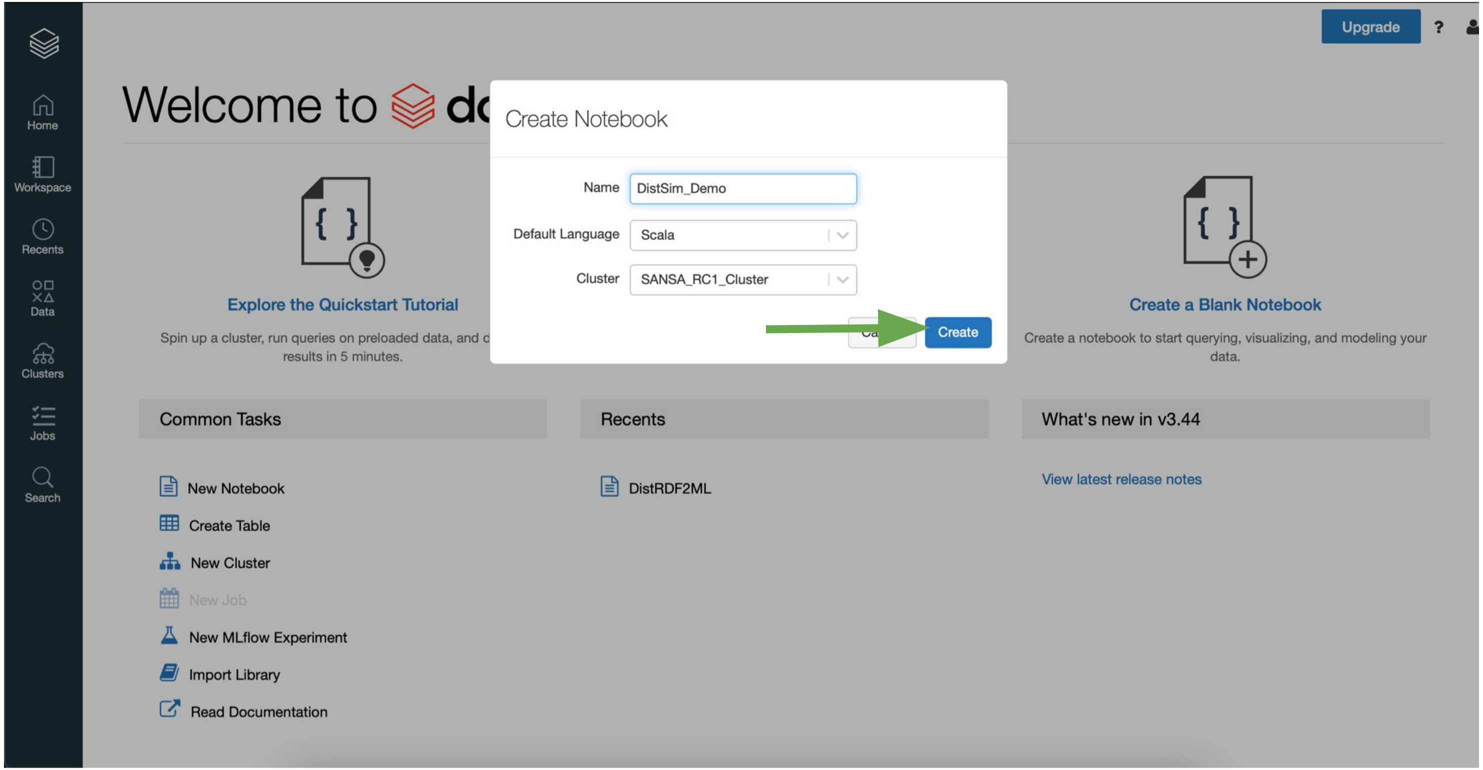
Create Notebook

Name: DistSim_Demo

Default Language: Scala

Cluster: SANSA_RC1_Cluster

Create Notebook - Create Notebook



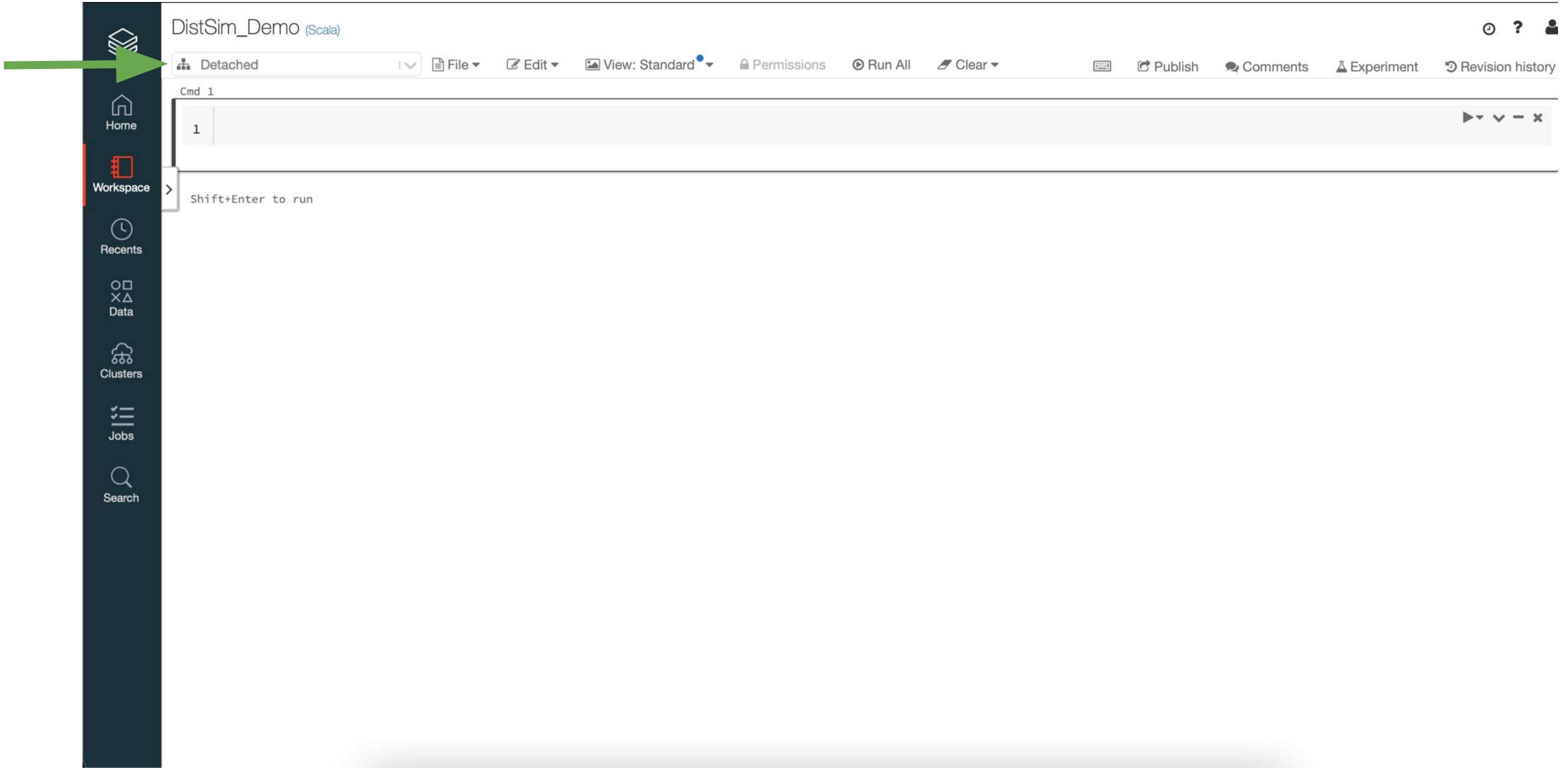
Create Notebook

Name: DistSim_Demo

Default Language: Scala

Cluster: SANSA_RC1_Cluster

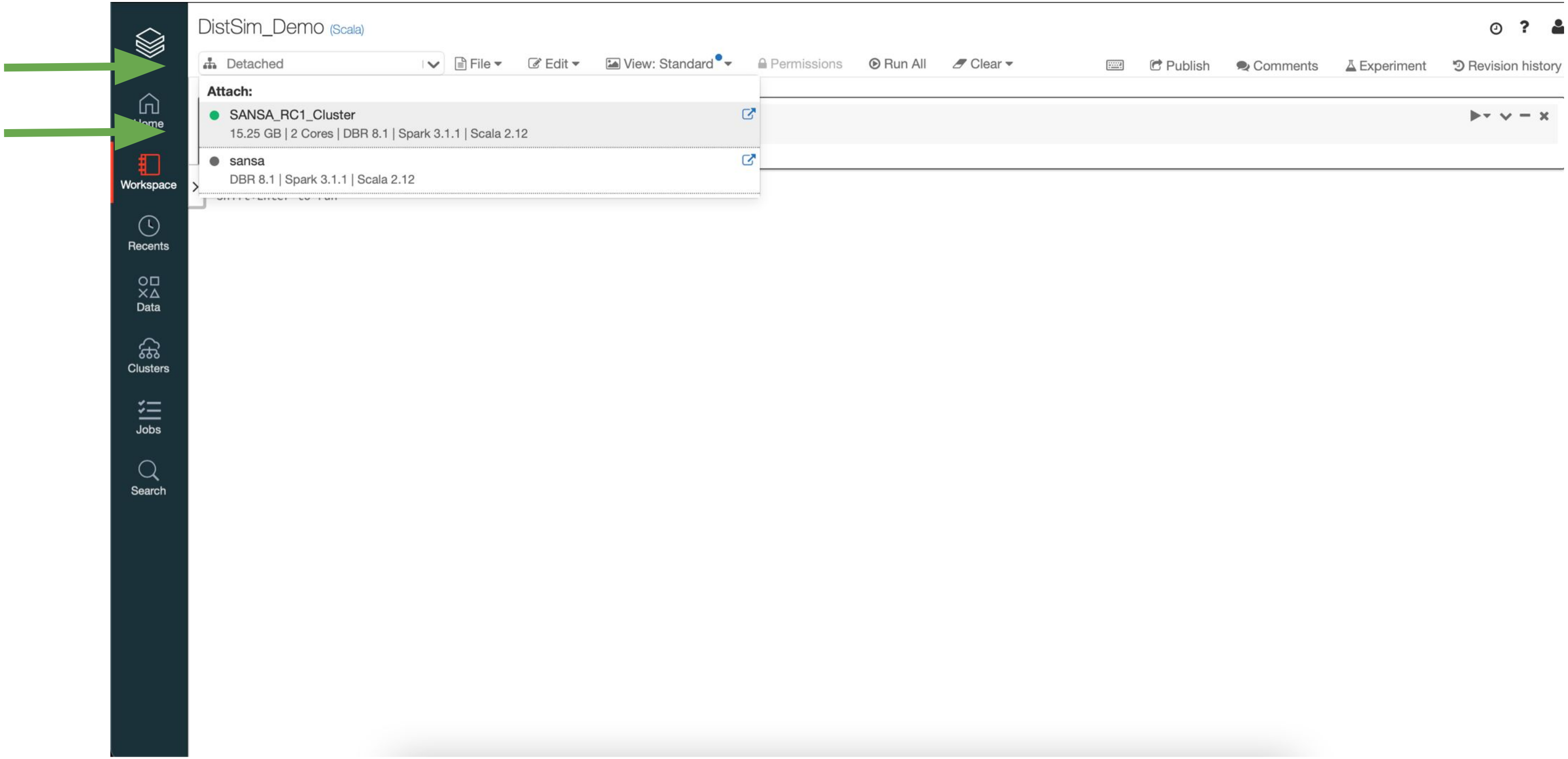
Create Notebook - Specify Cluster



```
Cmd 1
```

Shift+Enter to run

Create Notebook - Specify Cluster



Attach:

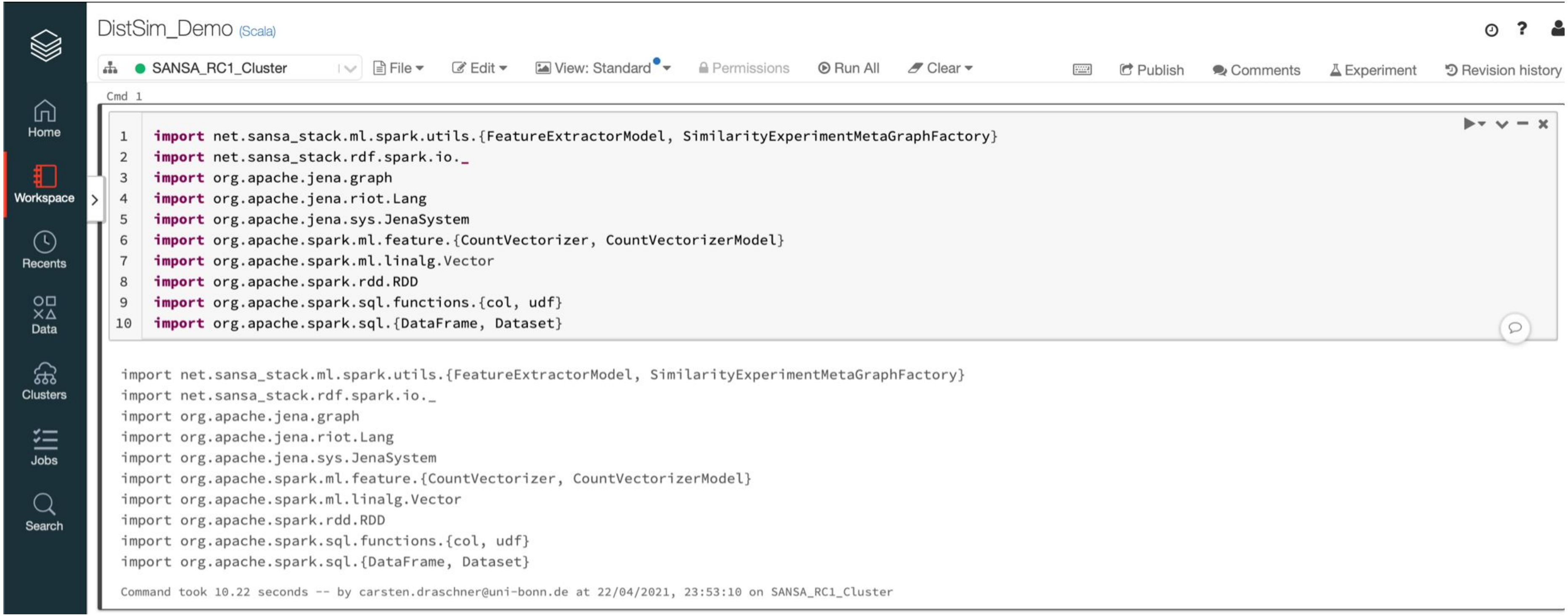
SANSA_RC1_Cluster
15.25 GB | 2 Cores | DBR 8.1 | Spark 3.1.1 | Scala 2.12

sansa
DBR 8.1 | Spark 3.1.1 | Scala 2.12

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

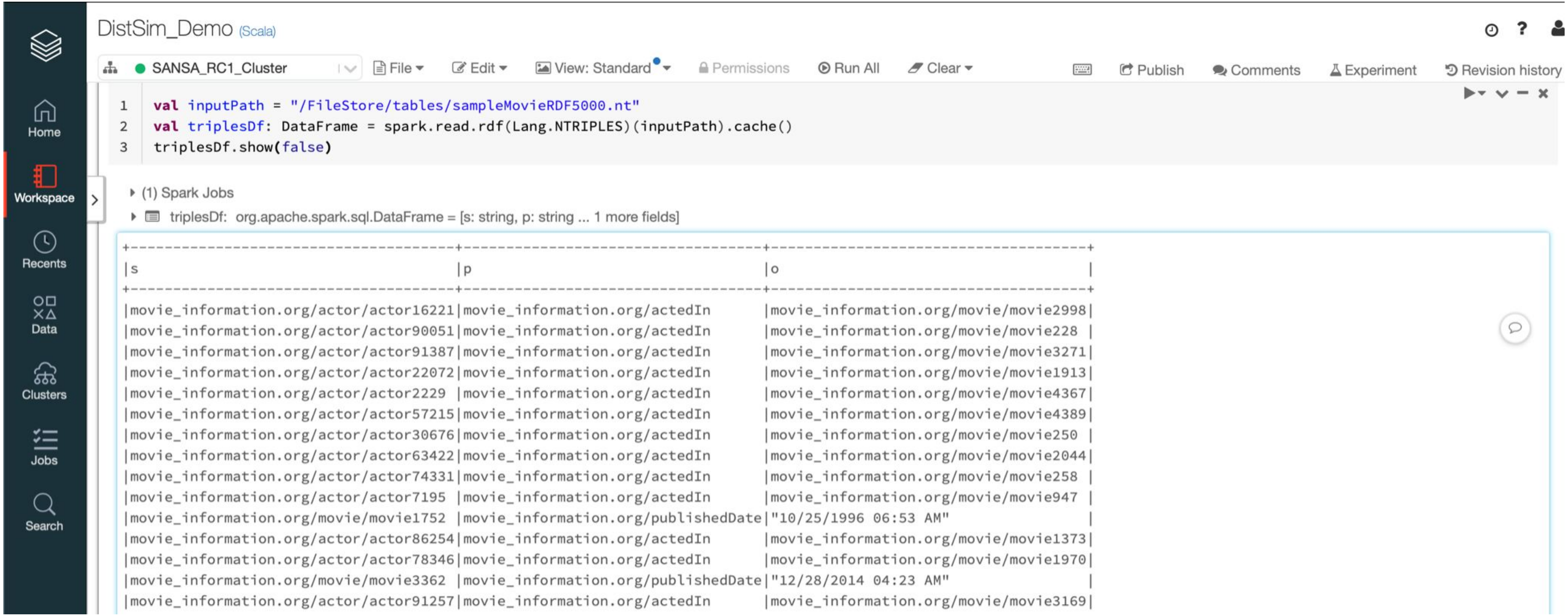


```scala
import net.sansa_stack.ml.spark.utils.{FeatureExtractorModel, SimilarityExperimentMetaGraphFactory}
import net.sansa_stack.rdf.spark.io._
import org.apache.jena.graph
import org.apache.jena.riot.Lang
import org.apache.jena.sys.JenaSystem
import org.apache.spark.ml.feature.{CountVectorizer, CountVectorizerModel}
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}
```

```
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



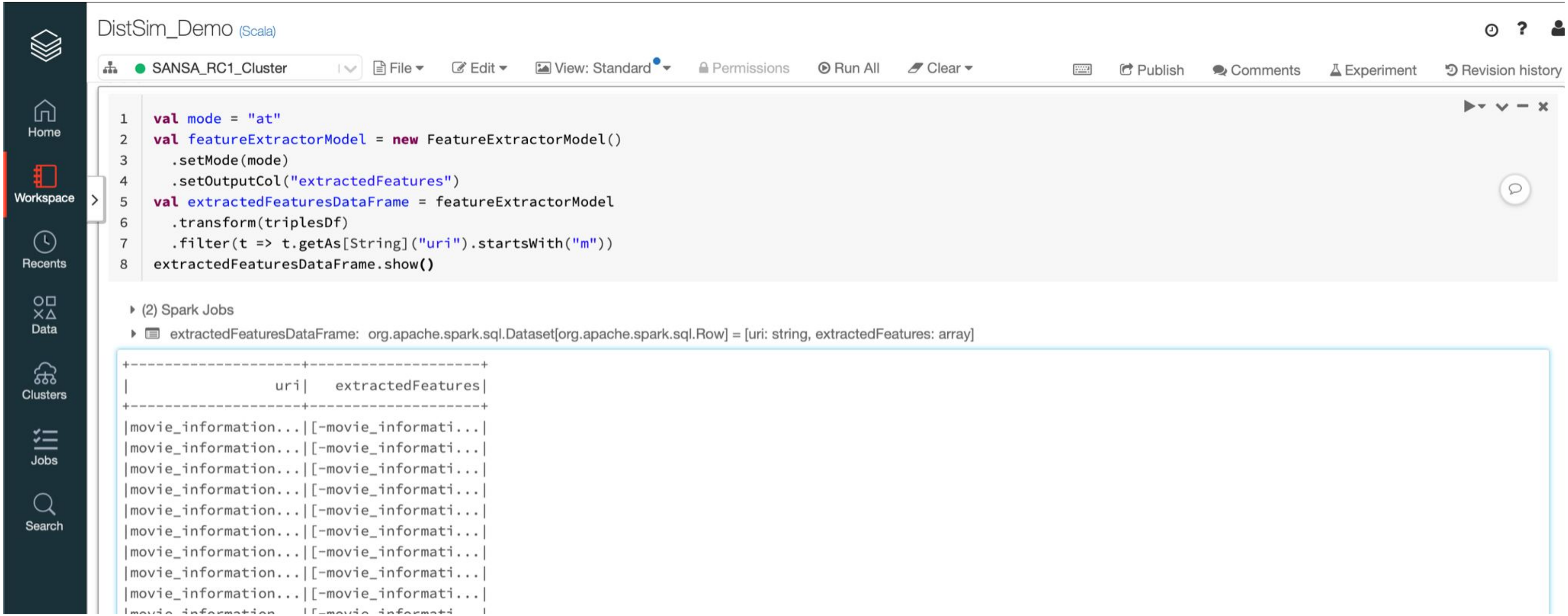
```scala
val inputPath = "/FileStore/tables/sampleMovieRDF5000.nt"
val triplesDf: DataFrame = spark.read.rdf(Lang.NTRIPLES)(inputPath).cache()
triplesDf.show(false)
```

(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



```
val mode = "at"
val featureExtractorModel = new FeatureExtractorModel()
  .setMode(mode)
  .setOutputCol("extractedFeatures")
val extractedFeaturesDataFrame = featureExtractorModel
  .transform(triplesDf)
  .filter(t => t.getAs[String]("uri").startsWith("m"))
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...|
```