

# Table of Contents

## Data Explorer

<a href="#">Containers</a>	2
<a href="#">Datasets</a>	5
<a href="#">Dataset Catalog browsing</a>	8
<a href="#">Resultsets</a>	11
<a href="#">adectl - Miscellaneous commands</a>	14

## Data Explorer > Getting Started

<a href="#">Overview</a>	15
<a href="#">Pre-requisites</a>	20
<a href="#">User registration</a>	21
<a href="#">Local mode setup</a>	24

## Data Explorer > Data Ingestion

<a href="#">Preparation</a>	26
<a href="#">Ingesting data and catalog</a>	27

## Data Explorer > Explore and refine

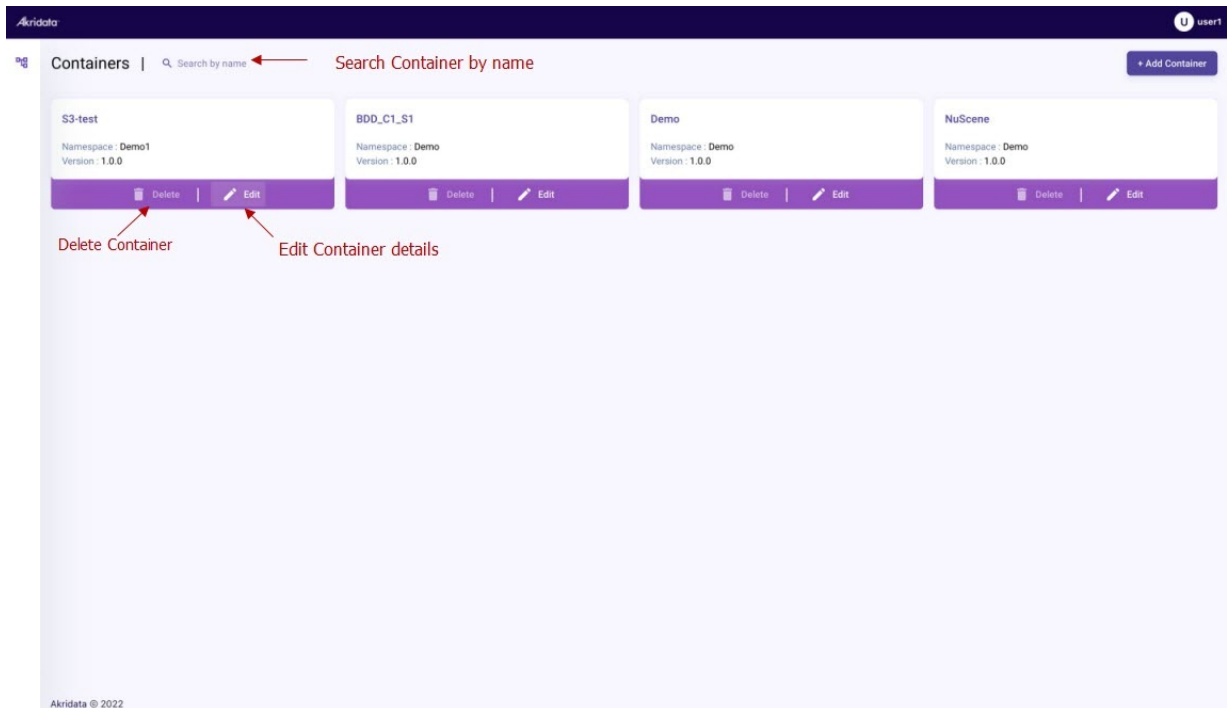
<a href="#">Jobs</a>	29
<a href="#">Job creation and visualization</a>	31
<a href="#">Select and Refine</a>	34
<a href="#">Similarity search</a>	39

# Containers

A **container** describes a storage location from where data is intended to be ingested into the system. A container can be an S3 bucket, Azure blob store or a directory on the local file system. The container is registered through the UI with user providing the details like the end point URL, credentials etc.

If you plan to ingest data from your local machine where the adectl setup was run, then skip this step.

The Containers page is displayed with the options as seen in the below image,



## Containers

For S3/Azure blob store etc, create a container by clicking **Add Container** button. Following fields are displayed to add a container,

## Add Container

+ Add Container ×

Name  Namespace

**Add Data Store**

Data Store Name  Select Store Type  URI

## Add Container

Fill the required information.

1. **Name** - An identifier for the container.
2. **Namespace** - Namespace can be used to group related containers into a logical collection. By default a namespace 'default' is assigned.

## Add Data Store

1. **Data Store Name** - Enter the data store name.
2. **Select Store Type** - Select a store type from the drop down list.
  1. If S3 is selected, below options are displayed,

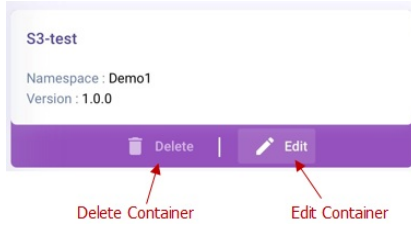
The screenshot shows a form titled "Add Data Store" with a "Save" button in the top right corner. The form contains the following fields:

- Data Store Name**: A text input field with a red border and a "Mandatory Field" label below it.
- Select Store Type**: A dropdown menu with "S3" selected.
- URI**: A text input field with a copy icon (Ⓔ) to its right.
- Credentials**: A section header for the following fields.
  - Access Key**: A text input field.
  - Secret Key**: A text input field.
- Region**: A text input field.

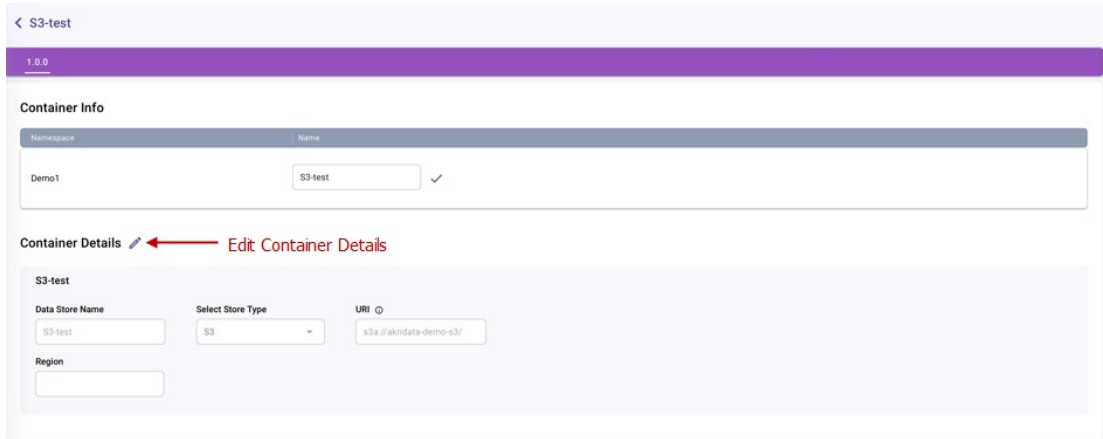
Data store Enter the data store name, URI, Access key, Secretkey and Region.  
The URI field should specify the path to folder with objects to be ingested(example: s3://my-bucket/d1/d2)Click **Save** button.

## Edit Container Details

1. Click **Edit** as seen in the below image to edit Container details,

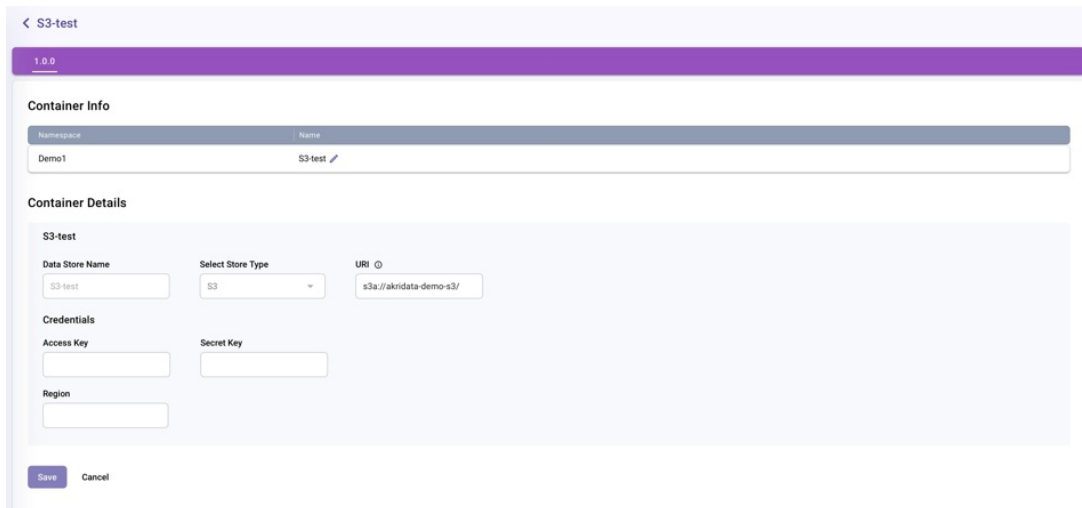


**Edit Container** Following options are displayed along with the container details,



### Container Details

2. Click **Edit** to edit container details.  
Following options are displayed,



### Edit Container Details

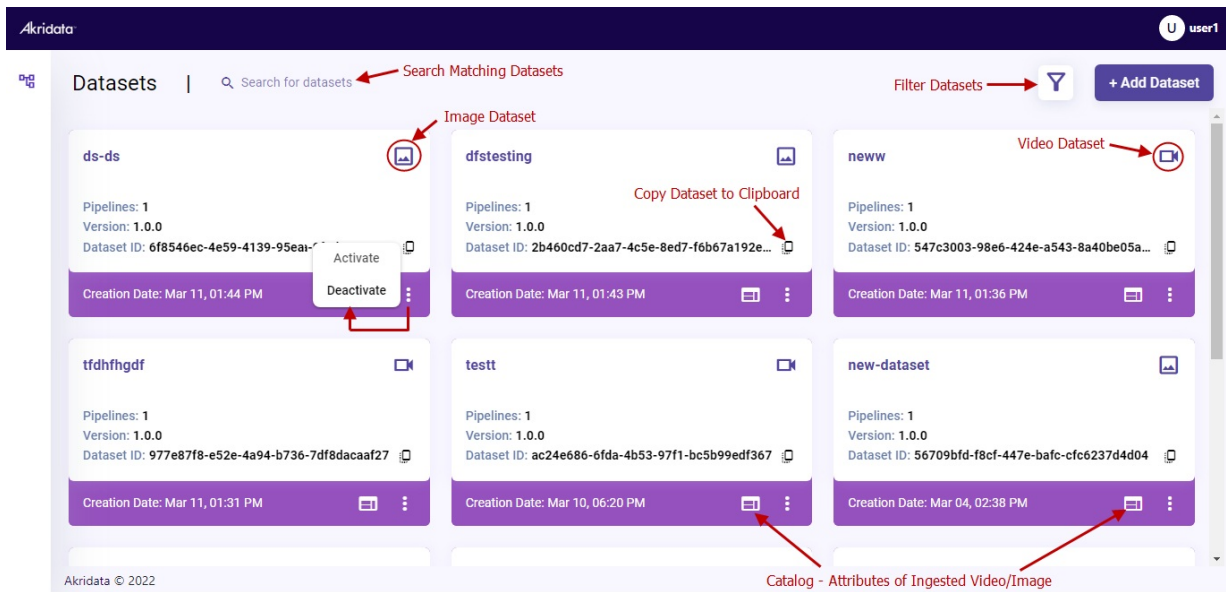
3. Enter the required information and click **Save** button to save the changes.  
The Container with the latest information is displayed in the Containers page.

# Datasets

A **dataset** is an entity that specifies a selector on the contents of the container. A dataset can be of Image or Video type.

For example, An S3 bucket has two directories CAMERA-FRONT and CAMERA-BACK with images from front and back cameras respectively and each of these camera images have different feature extraction model that is most appropriate. For such a case, you can define two datasets with glob pattern CAMERA-FRONT/\*\*/\*.\*.jpg and CAMERA-BACK/\*\*/\*.\*.jpg respectively to logically group the images from two cameras.

The various fields and controls on the datasets page is shown below,



## Dataset listing page

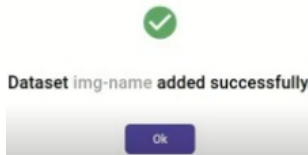
- **Search for datasets** - It is a search box where user can enter dataset name prefix to search for matching datasets (e.g. entering test should list 5 datasets starting with test).
- **Filter** - This option allows the user to view the criteria on which the datasets can be filtered.
- **Add Dataset** - The user can add a dataset by clicking on **Add Dataset** button. Following fields are displayed to add a dataset,

The screenshot shows a vertical form titled "Add Dataset". It contains the following fields from top to bottom:

- Enter Dataset Name**: A text input field.
- Enter Dataset Namespace**: A text input field containing the value "default".
- Select Data Type**: A dropdown menu with "Select" and a downward arrow.
- Select Source Container**: A dropdown menu with "Select" and a downward arrow.
- Glob**: A text input field with a help icon (ⓘ) to its right.
- Add**: A purple button with white text.

### Add Dataset

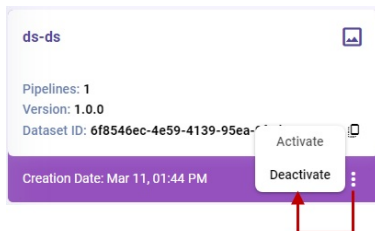
1. Fill the required information and click **Add** button.
  1. **dataset name** - An identifier for the dataset.
  2. **Namespace** - Namespace can be used to group related datasets into a logical collection. For example, all datasets used for test purposes can be grouped into a namespace called 'test' and datasets used for production can be grouped into namespace called 'prod'. By default a namespace 'default' is assigned.
  3. **Data Type** - Can be Image or Video type depending on the type of the data in dataset.
  4. **Source container** - Select a container(registered on Containers page) where data objects for this dataset are present. In case of ingesting data from a local file system directory, please use the '**Create Local Container**' option.
  5. **Glob** - The glob is pre-filled based on the data type selected and edit it based on file name pattern of files to be assigned to the dataset. For example, a glob pattern "\*\*(png|jpg|gif|jpeg|tiff)" captures all file with png, jpg, gif, jpeg, tiff file extensions.
2. The dataset gets added and is displayed in the **Datasets** page. Following message is displayed after successful addition of dataset.



### Dataset-creation-success

The added dataset can be activated or deactivated by selecting the options as seen in the below image,

Note: A dataset in deactivated state implies that no more data is ingested against the dataset.



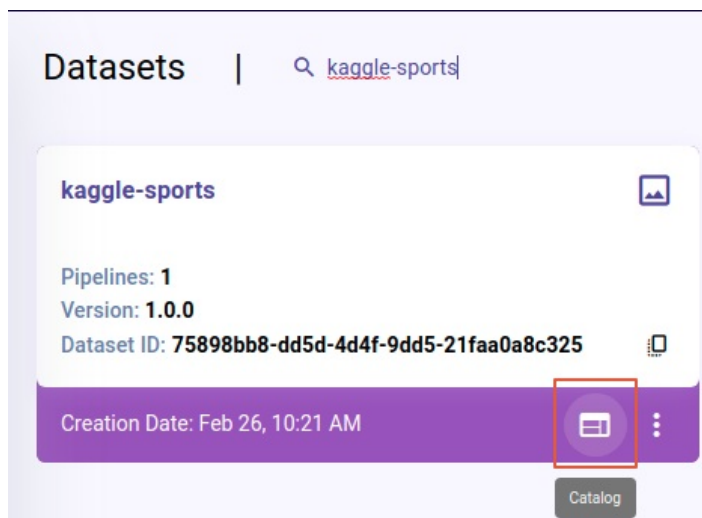
### Dataset activation/deactivation

## Dataset Catalog browsing

Each dataset has a catalog consisting of one or more tables that store metadata associated with ingested data. The catalog tables can be of two types,

- **Internal tables** - Tables that are auto-created and populated as part of data ingestion through 'adectl run' command.
- **Imported tables** - Tables that are created and populated when user imports external catalog information in a CSV file through 'adectl import' command.

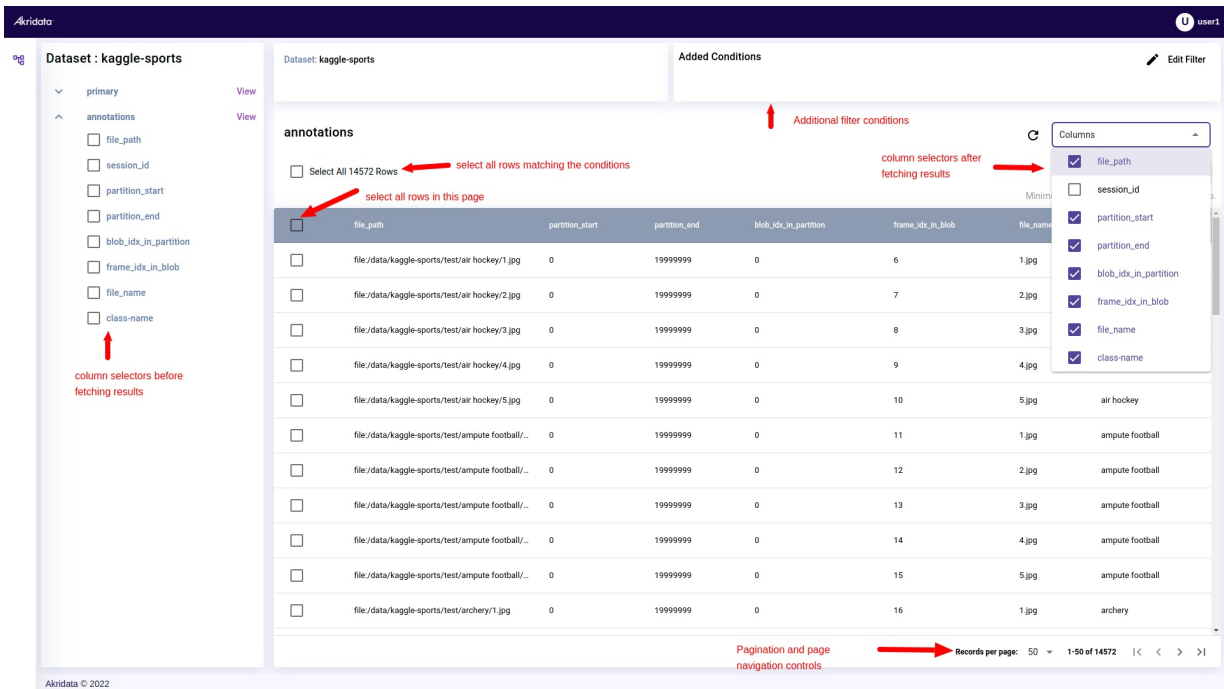
The catalog table is accessible from the catalog button on the dataset card in 'DataSets' page as shown below,



### Dataset Catalog button

The various fields and controls on the catalog page is shown below and includes column selectors, row selectors, pagination related controls and ability to specify filter conditions to filter the catalog rows of interest.



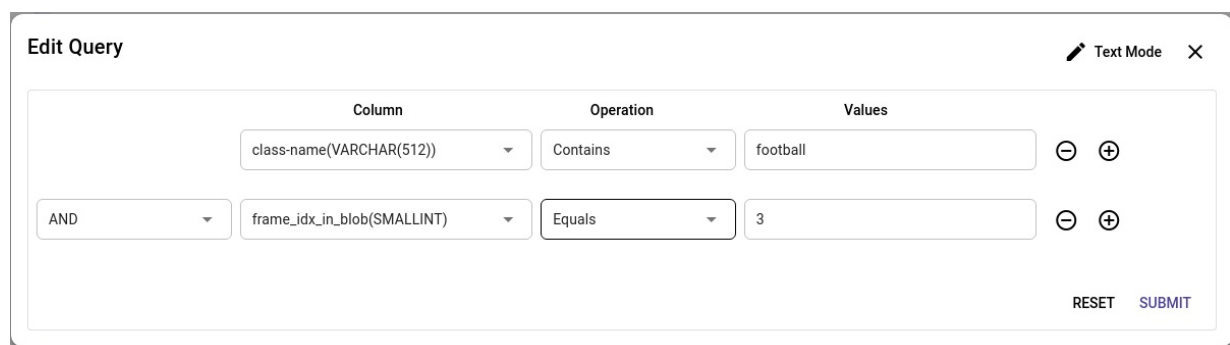


## Dataset Catalog page Filtering catalog rows:

The 'Edit filter' has 2 modes,

- **Basic mode** - A UI based selection of filtering conditions.
- **Text mode** (recommended for advanced users) - A text box where the filtering condition is specified as a SQL WHERE clause.

## Basic mode



## Catalog edit filter - basic mode

## Text mode

### Edit Query

Basic Mode✕

(Ctrl + Space) for suggestions

```
['class-name' LIKE '%football%' AND 'frame_idx_in_blob' = 3] OR (class-name=cricket AND frame_idx_in_blob = 2]
```

RESET SUBMIT

**Catalog edit filter - text mode** In text mode, entering (Ctrl + Space) will show a list of columns to select from.

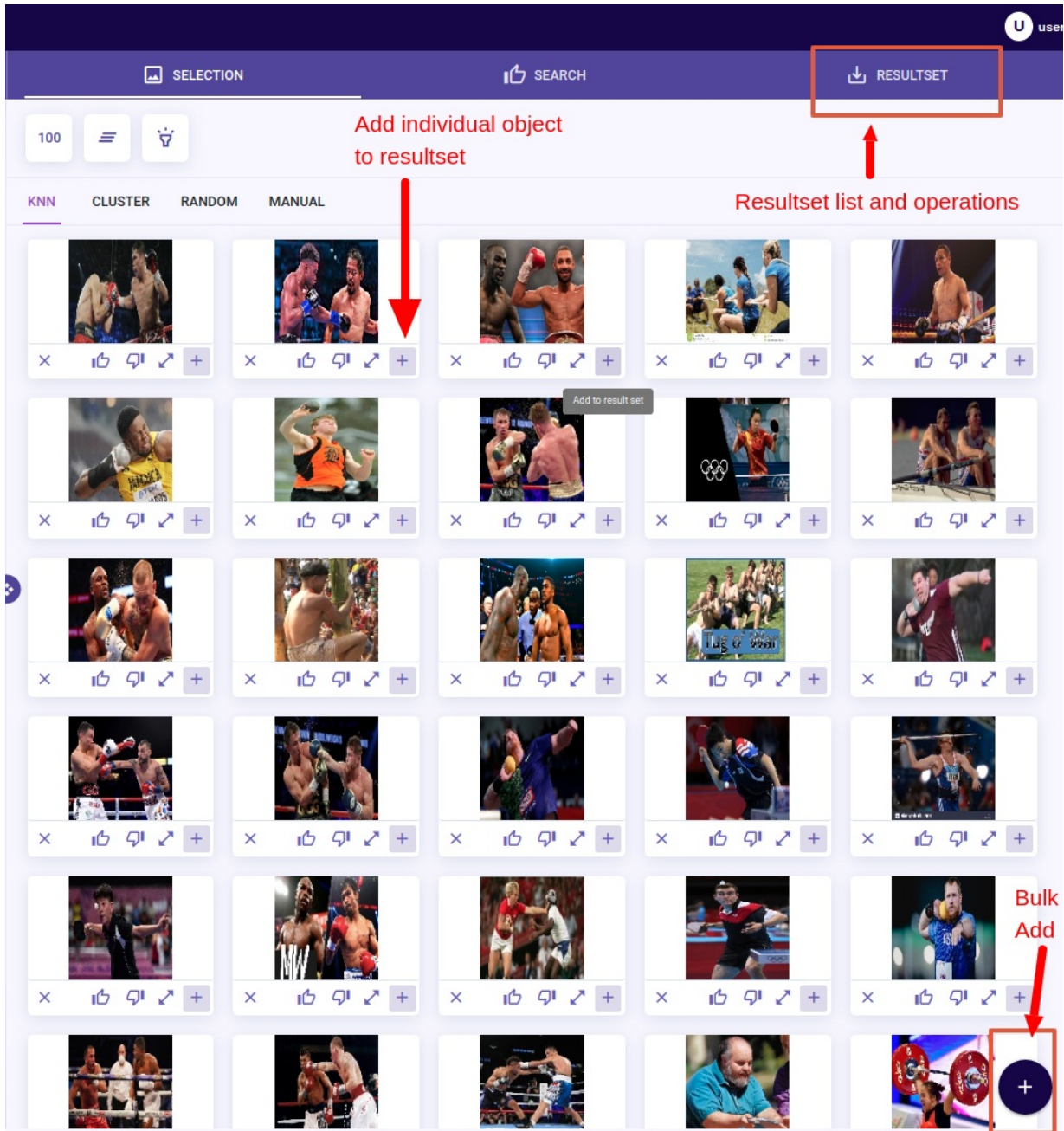
#### Switching from text mode to basic mode

Switching from text mode to basic mode will reset the query in basic mode since the text mode query may not be representable through UI based specification in basic mode.

# Resultsets

A resultset is a collection of curated data objects that the user has selected through different explore and refine capabilities provided by Data Explorer. Using an `adectl` command, user can upload the resultset objects into a S3 bucket, Azure blob store or a local file system.

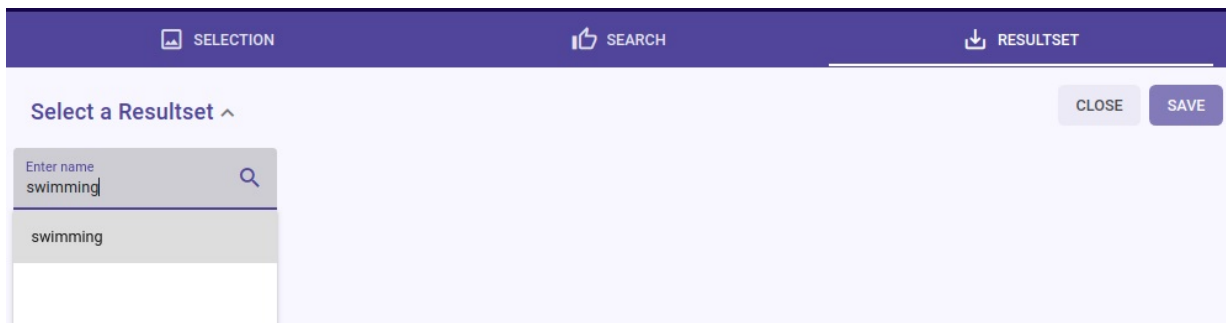
## Adding objects to resultset



**Resultset operations** The '+' button against each thumbnail and '+' button for bulk add on bottom right are used to add objects into resultset. If there is no resultset in opened state (when adding to resultset for the first time), a form to enter the name of the resultset will be presented.

## Opening an existing resultset

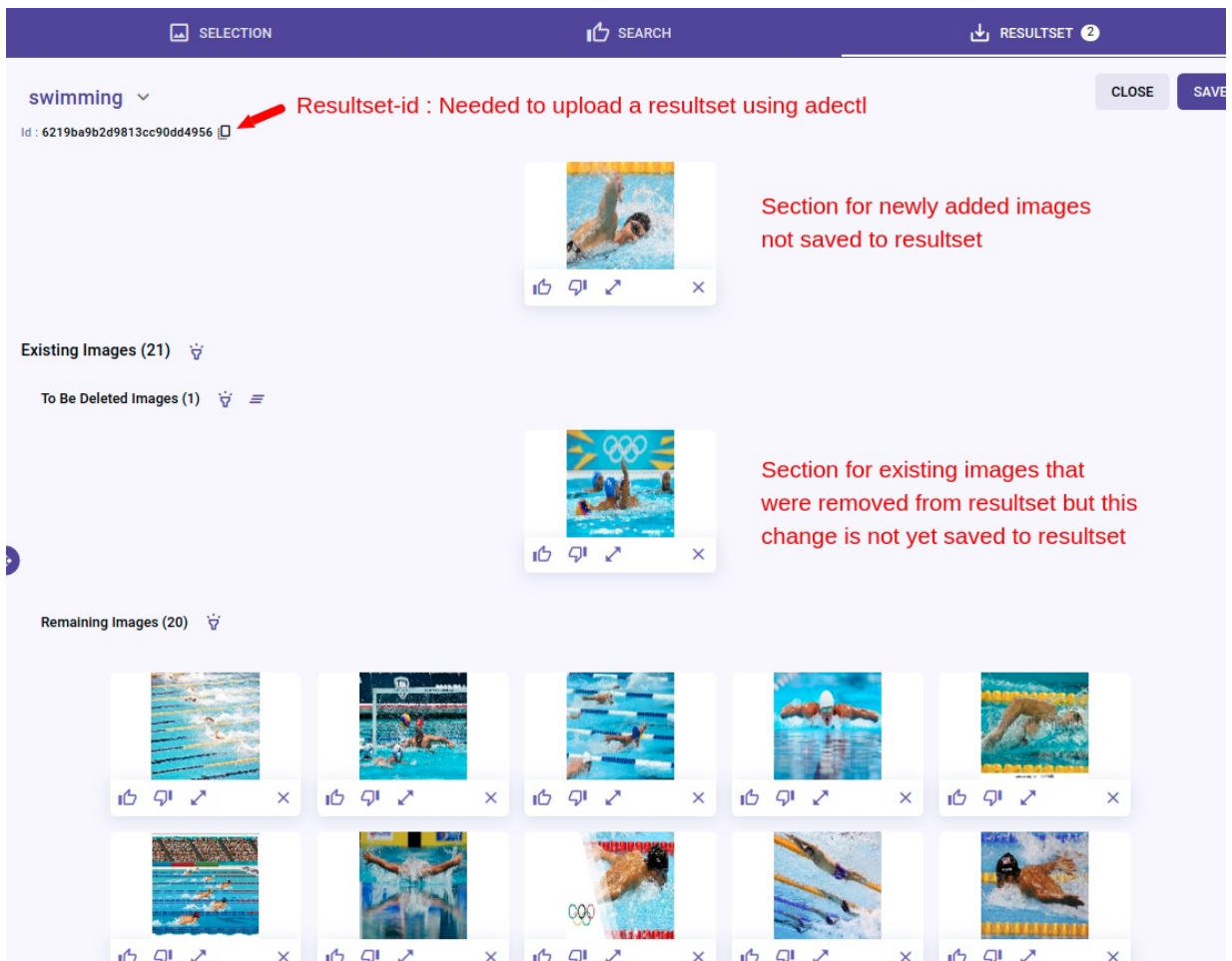
In the resultset tab, click **Select a Resultset** drop down arrow as shown in the below image and choose a resultset,



Open a resultset

## Edit resultset

Once a resultset is opened and in context, you can add objects to the resultset using the '+' buttons on the selection tab and similarity search tabs. You can delete the existing objects from resultset using the 'X' button against thumbnails presented in the 'RESULTSET' tab. The number 2 in the below diagram on the 'RESULTSET' tab indicates the number of objects that are part of the edit done on the resultset in a draft state that is not yet saved. Click **Save** button to save these draft edits to the resultset.



**Resultset edit operations**

### Adding to a new resultset when a resultset is in open state

If a resultset A is in open state and you want to add some objects to a new resultset B, then click 'close' in resultset tab to close the current open resultset A before adding object to new resultset. The 'close' operation will prompt if you want to save the changes made to resultset A or discard the changes.

## Resultset upload

The resultset upload operation allows the user to materialize the objects in a resultset onto a target S3 bucket, Azure blob store or a local file system for further analysis or connecting to downstream pipelines (e.g. training pipeline).

## Configure the target location

Bash	Copy
<pre>adectl resultset config</pre>	

This command will prompt for selection on the type of destination location (S3/Azure/. etc.) and necessary credentials and configuration information. As an example, the fields captured for S3 destination are listed below,

None	Copy
<pre>Select store type [s3   azure   file   hdfs] : s3 Enter S3 Bucket Name: bucket Enter S3 Access Key: xxxxx Enter S3 Secret Key: yyyy Enter S3 Endpoint [default: https://s3.amazonaws.com]: Configured S3 Store Successfully</pre>	

For Azure blob store as destination, storage account and storage key fields are needed.

## Upload resultset objects

Bash	Copy
<pre>adectl resultset upload -r &lt;resultsetid&gt; -t &lt;target-location&gt;</pre>	

- **resultsetid** - Available under 'RESULTSET' tab on the UI.
- **target-location** - Location relative to the configured location using `adectl resultset config` command. For e.g., if S3 bucket name configured is `s3://bucket` and `-t` is specified as `/rsupload` then the resultset objects will be uploaded to `s3://bucket/rsupload`.

The command starts the upload operation as an asynchronous operation and the status of this operation is available by running the following command.

Bash	Copy
<pre>adectl show</pre>	

# adectl - Miscellaneous commands

## Getting help

Bash	Copy
<pre>adectl help -&gt; shows top level commands adectl help &lt;sub-command&gt; e.g. adectl help setup -&gt; shows help about adectl setup</pre>	

## Collecting logs

In case an issue is seen that is to be reported to Akridata support, please collect the logs using the following command.

Bash	Copy
<pre>adectl logs</pre>	

This will create a log file by default in /tmp with a name dataexplorer-logs-`<timestamp>`.tar.gz

## Updating software version

Bash	Copy
<pre>adectl update</pre>	

This command will check for a newer version of software is available and prompt for confirmation to go ahead with the update.

## Teardown the setup

Bash	Copy
<pre>adectl teardown</pre>	

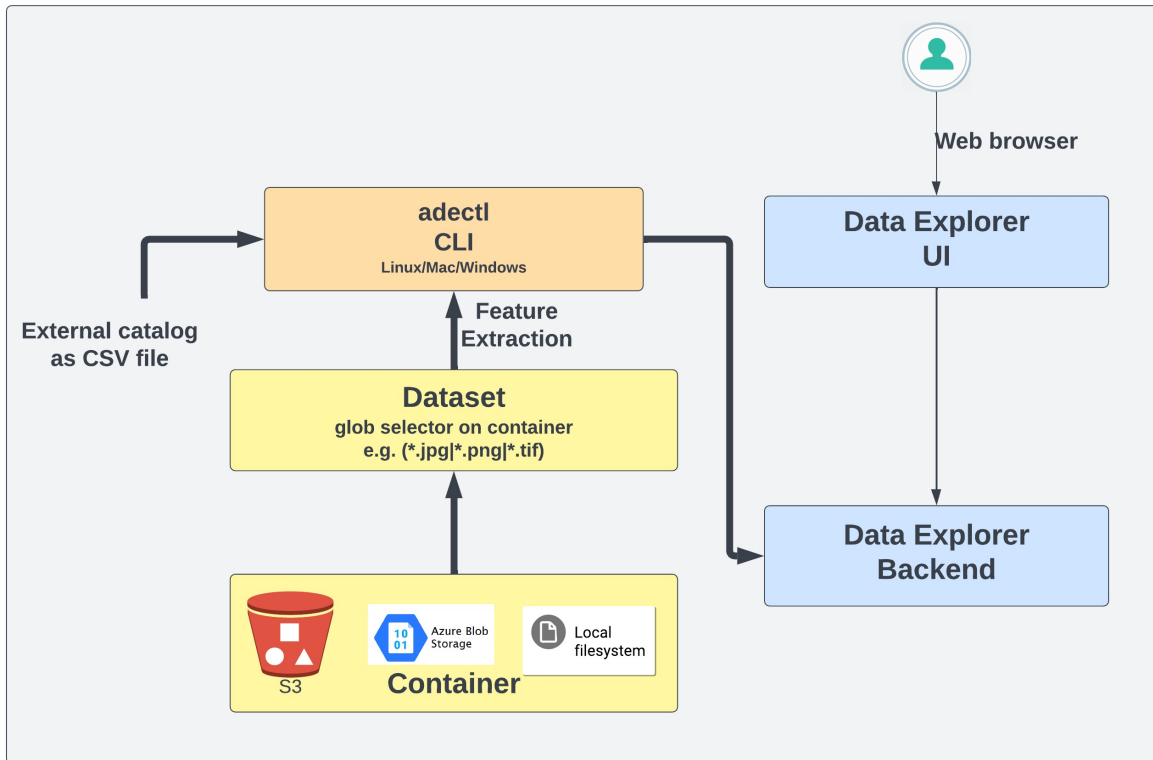
This command will delete the local mode setup.

### Teardown operation

A teardown operation will cleanup the state and hence all Data explorer entities like Datasets, containers, jobs, catalog etc will be lost and cannot be recovered. The data that was source of the ingestion is unaffected by a teardown operation.

# Overview

## System architecture



**Data Explorer - System Architecture** The system consists of following components,

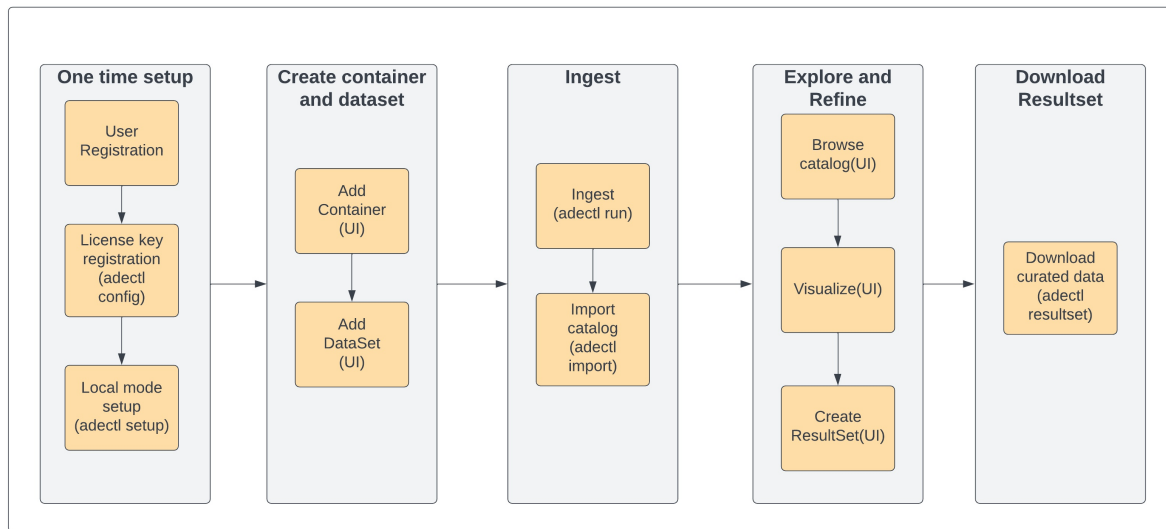
- adectl CLI tool to ingest new data and catalog into the system. The data can be ingested from multiple sources like S3 bucket, Azure blob store and a local file system. The catalog is ingested as a comma-separated-values(CSV) file.
- A web browser based application for browsing the catalog and curating the ingested data.

There are following modes of deployment,

- adectl and web application running on the same machine - This mode is for evaluation purposes only.
- adectl on a user's machine with full access to source data and web application hosted on cloud in SaaS(software-as-a-service) mode - The data ingest is user triggered through adectl CLI and hence suitable for pilot and small scale production.
- Web application hosted on the cloud in SaaS mode with auto-managed data ingestion - In this mode, the user registers the data source with Data explorer as an one time operation. Data explorer will automatically provision compute resources in the region close to the data and run ingestion process on existing and new data as a fully automated operation without any user involvement.

**Only mode 1 is supported at this point. Other modes are on the roadmap, so please stay tuned.**

## User interaction flow chart



## User operations Flow Chart

### Container

A **container** describes a storage location from where data is intended to be ingested into the system. A container can be an S3 bucket, Azure blob store or a directory on the local file system. The container is registered through the UI with user providing the details like the end point URL, credentials etc.

#### Local Container

If data to be ingested is present on the local file system, then explicit container creation is not required.

### Dataset

A **dataset** is an entity that specifies a selector on the contents of the container. A dataset can be of Image or Video type.

For example, an S3 bucket has two directories CAMERA-FRONT and CAMERA-BACK with images from front and back cameras respectively and each of these camera images have different feature extraction model that is most appropriate. For such a case, you can define two datasets with glob pattern CAMERA-FRONT/\*\*/\*.\*.jpg and CAMERA-BACK/\*\*/\*.\*.jpg respectively to logically group the images from two cameras.

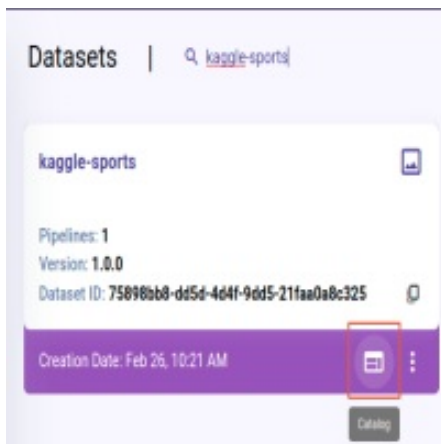
### Pipelines

A pipeline is an abstraction that captures the ingest processing routines. A pipeline typically has feature extraction, thumbnail generation and feature summarization stages. In the current release, a default provided pipeline is attached to the dataset when dataset is registered. This pipeline is triggered through the following command.

Bash	Copy
<pre>adectl run -d &lt;dataset-id&gt; -i &lt;directory-with-input-objects&gt;</pre>	



The output of the above processing is accessible through the 'Catalog' button on the dataset card as shown below.



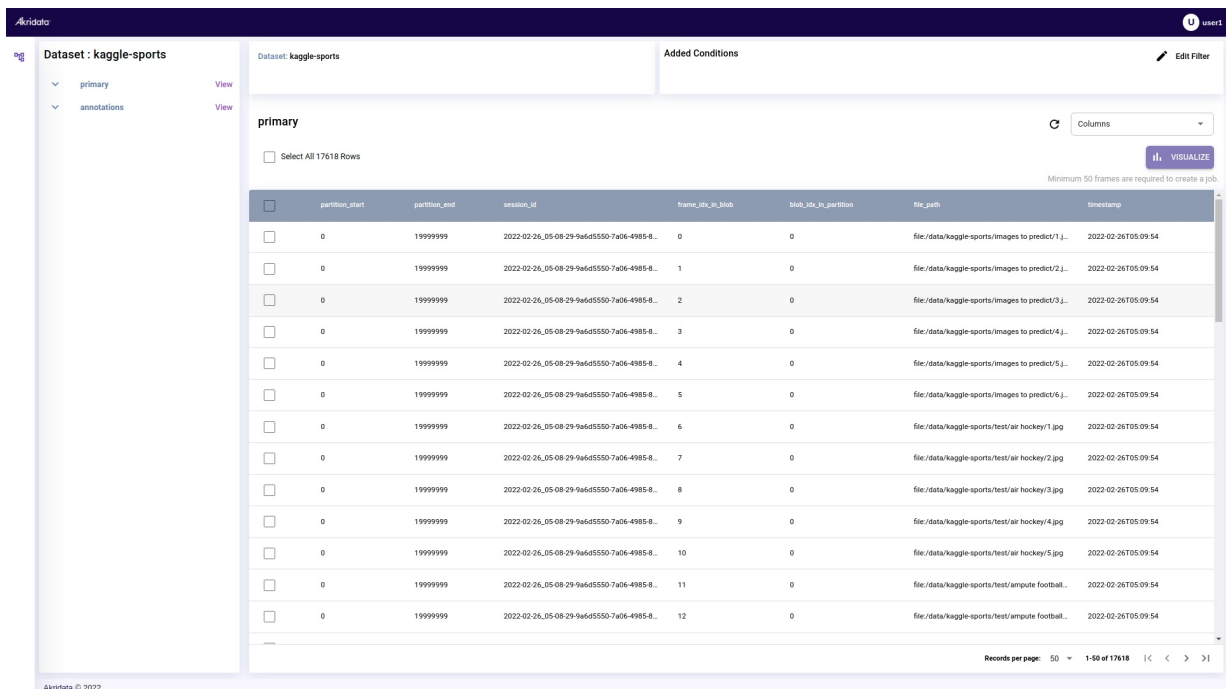
### dataset-catalog-access

#### Coming Soon - Pipeline customization

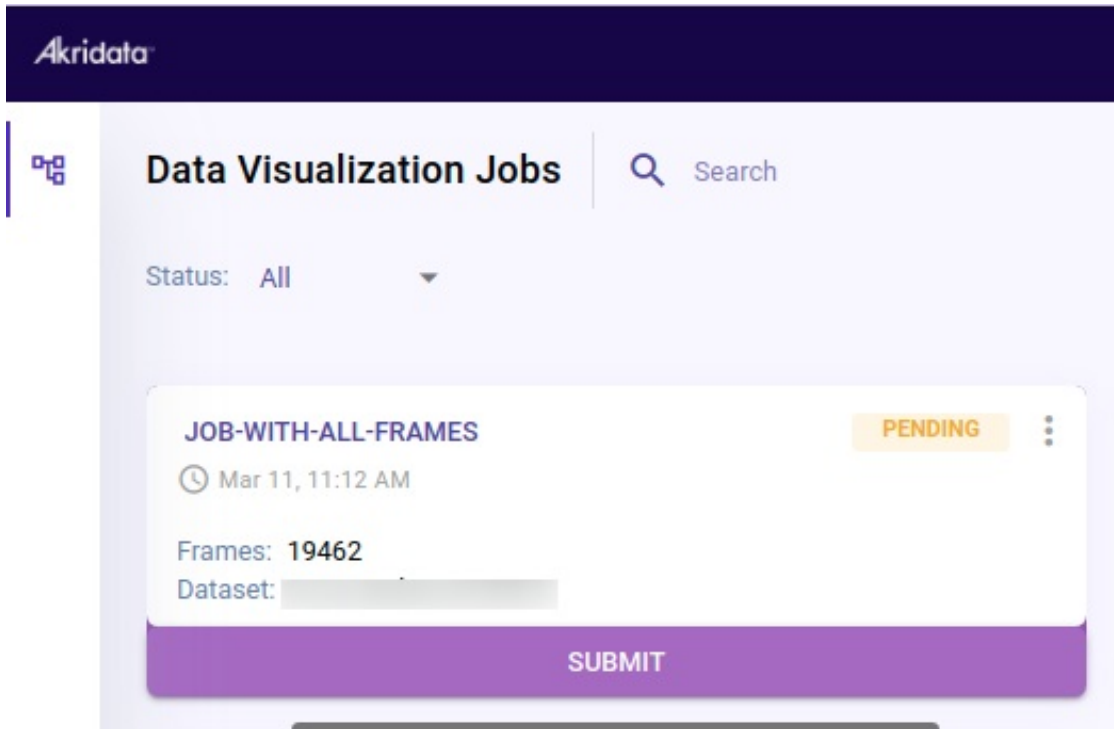
The default pipeline may not be best fit for data across all domains. The pipeline abstraction will be extended to support user provided featurizer in upcoming releases.

## Data Visualization Job

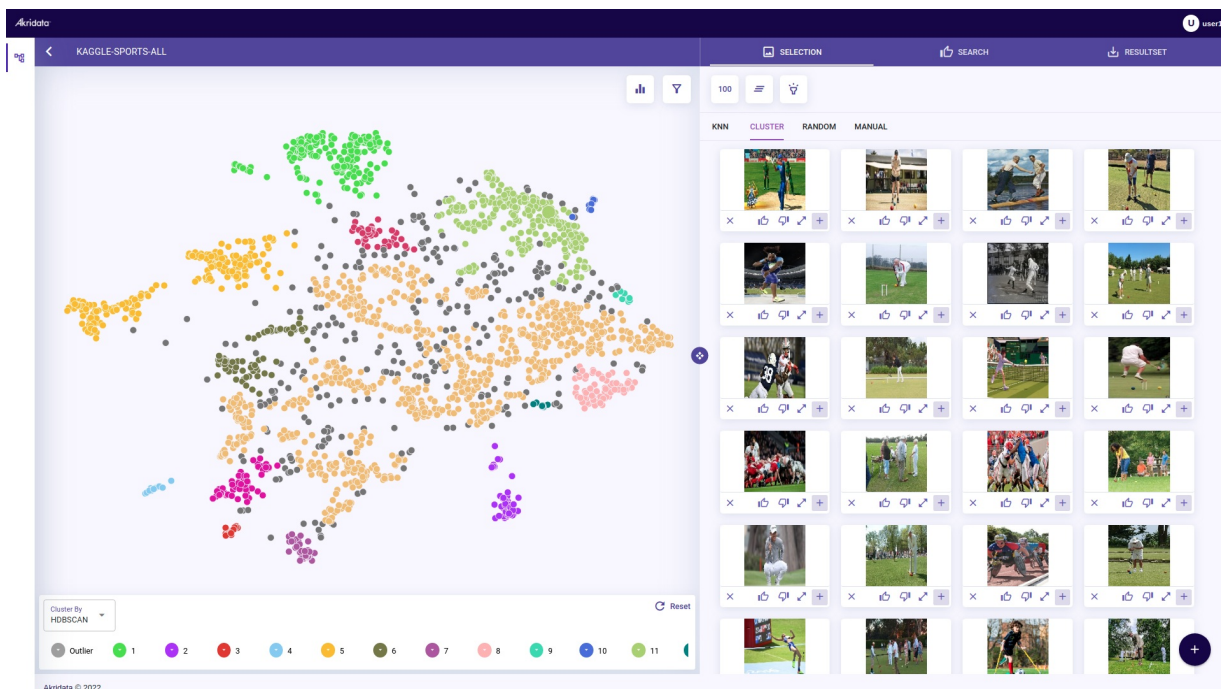
Once data is ingested, a data visualization job can be created by browsing the catalog by clicking on the 'Catalog' button in the dataset card as shown in the previous section.



### Catalog browsing to create a data visualization job



Visualization job before submission



Visualization Job

## Resultset

Data visualization UI provides capabilities to explorer, drill down and curate the data using cluster views, nearest neighbour searches and similarity searches. The curated subset of the data objects is referred to as a **resultset**. A resultset can be downloaded to a local directory or exported to a S3 bucket or Azure blob store for downstream processing (e.g. machine learning training pipeline) on the curated data.

## Partitions

During ingestion, the dataset is divided into multiple partitions. Each object in a partition is featurized using a deep neural network. The features from all objects in partition are used to generate lower dimensional representations for every object and coresets which act as representative subset for objects in a partition.

## Clustering and Embedding

When a job is submitted, the low dimensional representations and coresets are used to cluster the data objects to enable exploration and curation.

## Local mode setup(For evaluation purpose only)

For evaluation purposes, Data Explorer provides a **local mode** deployment where entire software is deployed on a single machine in a scaled down mode. All data and control interactions stay within the single machine except the following.

- Downloading of software from Akridata repositories as part of local mode setup.
- License validation

Hence, this mode provides complete data privacy for user data.

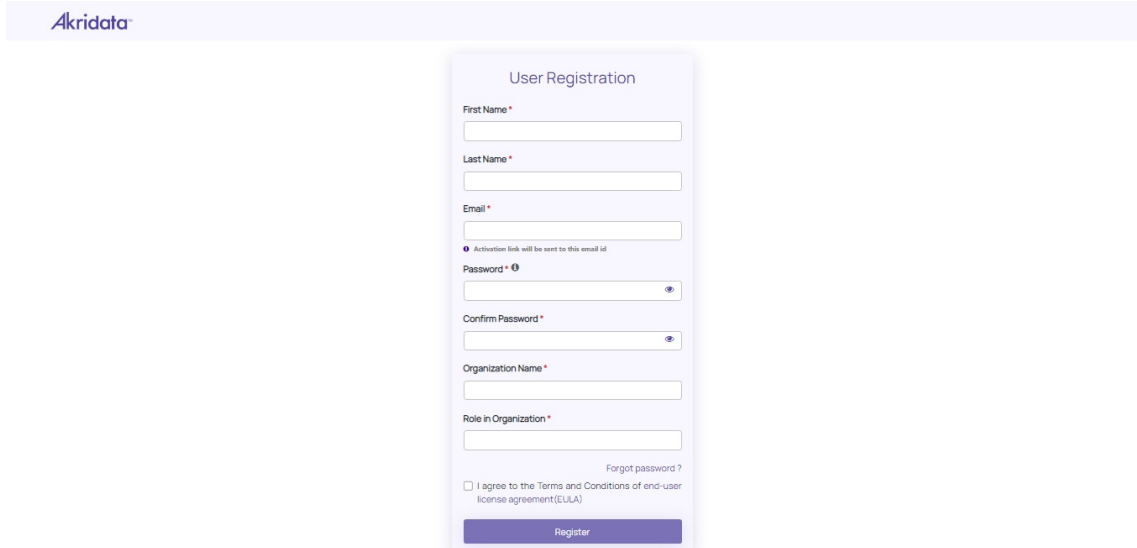
## Pre-requisites

- Linux or Mac machine with 8 GB RAM, 2 CPUs and 50GB storage. Both x86-64 and ARM-64 architectures are supported.
- docker v20.10.4+
- The user or UID that runs adectl CLI tool should have permission to launch docker containers (part of 'docker' group).
- If NVIDIA GPU available, then NVIDIA driver version 450.50.0+ and nvidia docker runtime - NVIDIA GPUs are supported only on Linux.

# User registration

Follow the below instructions to register new user,

1. Enter the URL, <https://de.akridata.ai> in the web browser and click Enter. The user registration page is displayed as seen below,

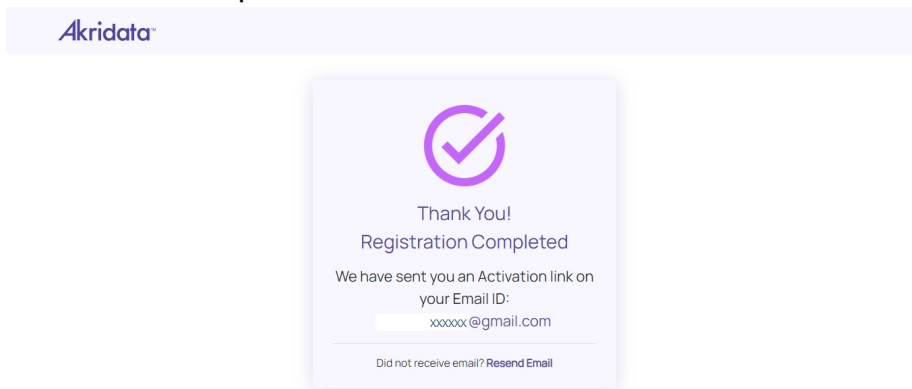


The screenshot shows the Akridata user registration page. At the top left is the Akridata logo. The main content is a white registration form with a purple border. The form is titled "User Registration" and contains the following fields: "First Name \*", "Last Name \*", "Email \*", "Password \*" (with an eye icon for visibility), "Confirm Password \*", "Organization Name \*", and "Role in Organization \*". Below the "Email" field, there is a note: "Activation link will be sent to this email id". Below the "Password" field, there is a link: "Forgot password?". At the bottom of the form, there is a checkbox: "I agree to the Terms and Conditions of end-user license agreement (EULA)". A purple "Register" button is located at the bottom of the form.

## User Registration Details

2. Fill in the required information, accept the terms and conditions of end-user license agreement (EULA) and click Register.

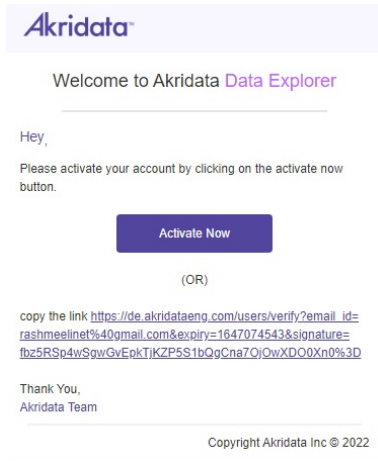
Below screen is displayed with an intimation that registration is complete and the activation link is sent to the email provided.



Copyright Akridata Inc © 2022

## Registration Completed

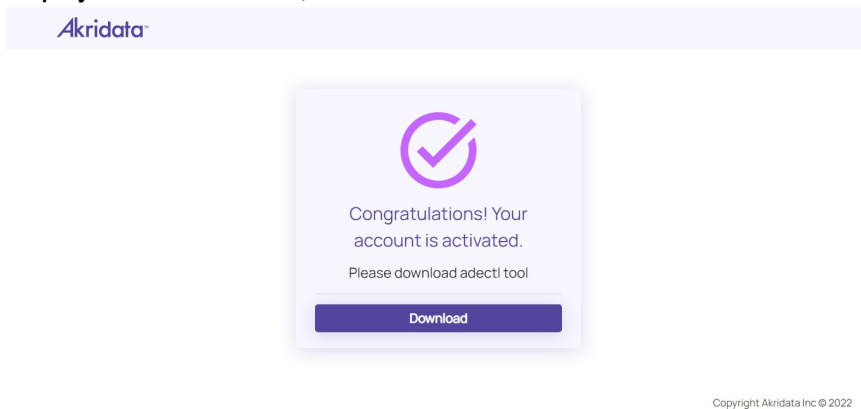
3. Open the mail containing the activation link.



**Activate**

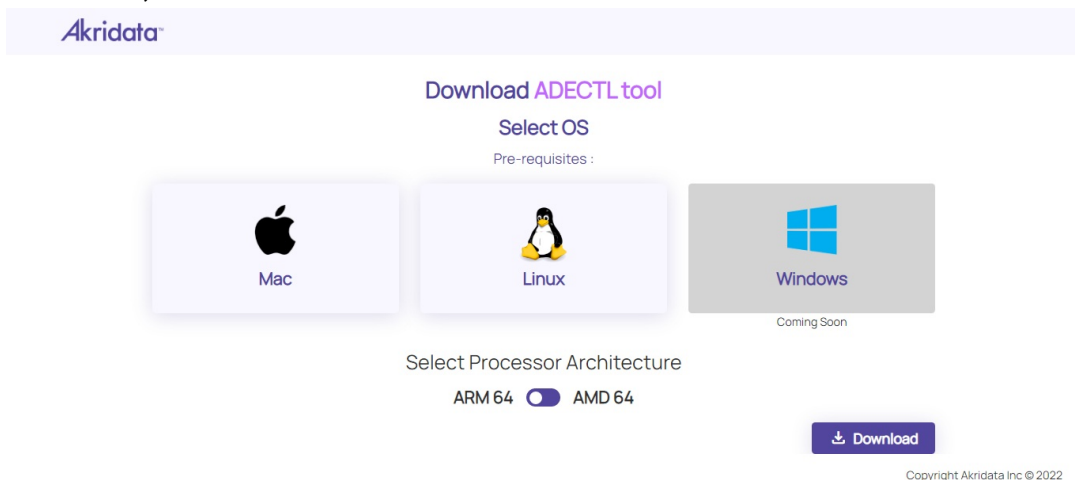
4. Click **Activate Now** button or copy the link provided to the web browser to proceed with the activation.

The account is activated and a screen with successful activation and **Download** button is displayed as seen below,



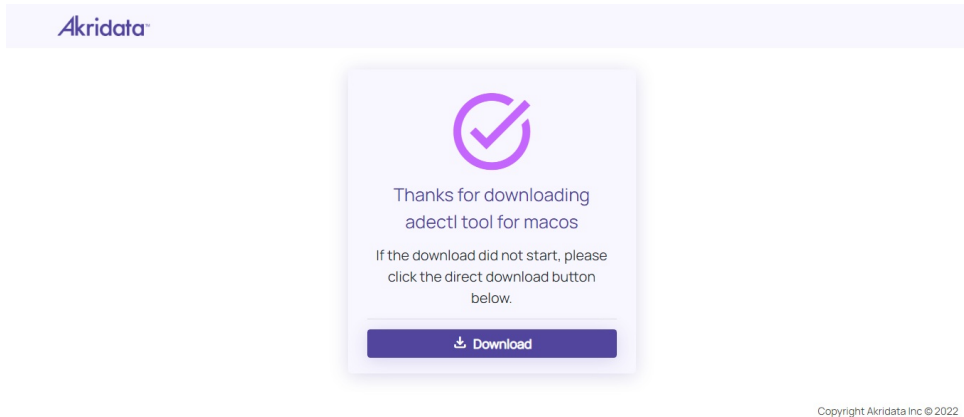
**Account Activated**

5. Click **Download** button, a screen with pre-requisites to download aeductl tool is displayed as seen below,



**Select OS**

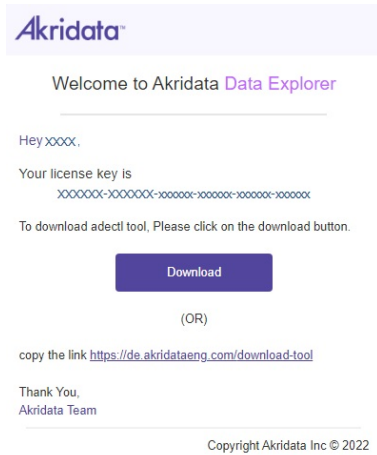
6. Select the preferred OS and the processor architecture (ARM/AMD) and click **Download** button. adectl tool is downloaded and the below screen is displayed,



Copyright Akridata Inc © 2022

### Download Successful

The license key is sent to the registered email address which can be used at the time of installation.



### License Key

## Local mode setup

Once the user registration is complete, you will receive a license key and link to download the CLI tool for your platform. Once adectl is downloaded, copying the tool to a directory accessible on system path is recommended.

### Docker installation check

Check docker is installed with correct permission for the user by running a 'docker ps' command

## Configure adectl

adectl configuration requires the following information,

- User email id and password used during user registration.
- License key
- If data to be ingested is on the local machine, then a top level directory where data is present.

Shell

Shell

Shell

Bash	Copy
adectl config	

Text

Text

Text

None	Copy
Enter email id : a@b.com Enter password : Enter license key : <license-key> Local data root directory(mandatory if ingesting data that is on this machine) : /data	

### Local data root directory

In the above example with /data configured as local data root directory, all local file system data for ingestion is expected to be within /data directory tree.

!! This directory cannot be set or changed later without tearing down the local setup !!

**Note:** For red hat/centos OS, to run at setup/reboot of machine:

```
sudo modprobe ip_tables
```

## Setup adectl

Shell

Shell

Shell



Bash	Copy
<pre>adectl setup</pre>	

- The above command will download all the required software and setup a local mode setup. The detailed logs from the setup are stored at `<user_home>/.adectl/logs/setup.log`.
- In case of any failures, the command can be rerun.
- Depending on the network speed for download, the setup time can vary from 10-30 minutes.

## Check connectivity

If your web browser is on the same machine as the local mode setup, then enter <http://localhost:58086> to access the UI. If web browser is on a remote machine, then create a SSH tunnel to port 58086 before accessing the UI.

Shell

Shell

Shell

Bash	Copy
<pre>ssh -L 58086:localhost:58086 user@host ssh -L 58086:localhost:58086 -i &lt;pem-file&gt; user@host -&gt; if machine is setup for access with a PEM file</pre>	

# Preparation

## Preparing data

1. If you are using S3 or Azure blob store to ingest data, then there is no specific preparation step required.
2. If you are using local filesystem directory to ingest data, then please download the data into a sub directory within the data directory configured using 'adectl config'. Refer to [local mode setup](#) for details.

## Preparing catalog

To ingest catalog, a comma separated values(CSV) file must be prepared in the following format

None	Copy
<code>file_path(string), field1(type), field2(type)</code>	
An example CSV file	
<code>file_path(string), class(string), quality-score(int)</code>	
<code>a.jpg, person, 85</code>	
<code>b.jpg, vehicle, 50</code>	

The first line should be the header line with

1. `file_path(string)` being a mandatory column.
2. Name and data type for other columns must be provided. The supported data types are
  1. float
  2. int
  3. bool - For this type of a column, the valid values are 0 and 1.
  4. string

### CSV file location

Irrespective of whether data is present on S3, Azure or local file system, this CSV file should be present on the local file system wherever adectl CLI is installed.

The `file_path` should be specified relative

1. URI specified in container specification for S3 bucket and Azure blob store. As an example if container URI is `s3://my-bucket` and the file path is `s3://my-bucket/vehicle/a.jpg`, then the `file_path` should be specified as `/vehicle/a.jpg`
2. For directory on local filesystem, the file path should be relative to the data directory configured during 'adectl config'.

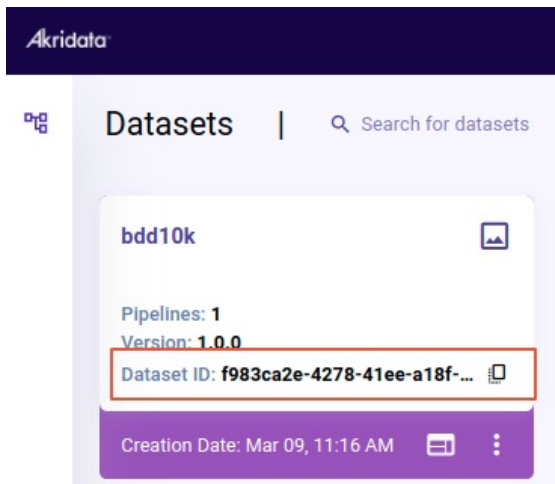
# Ingesting data and catalog

## Ingesting data

Run the following command

```
Bash Copy
adectl run -d <dataset-id> -i <path-with-images-or-videos-which-is-subdir-of-setup-dir>
```

Please get the dataset-id from the 'Datasets' page on Data Explorer UI



### Get dataset-id

The command starts ingestion process in the background and a message similar to below message is displayed.

```
Submitting the dataset process pipeline: /Submitted successfully.
Session ID: 2022-03-09_05-47-12-05af3635-ee57-401d-a98c-181ab431514a
Please run adectl show to monitor process status
```

## Monitor ingest progress

To monitor the progress of the background process, run the following command

```
Bash Copy
adectl show
```

Dataset ID	Name	Version	Type	Resultset ID	Progress	Status	Runtime	Data Path
f983ca2e-4278-41ee-a18f-24c89e17daa9	bdd10k	1.0.0	Image		100% (100/100 partitions)	COMPLETED	6m:17s	/data/bdd10k

The output show the following information

1. Dataset name and other details
2. Status
  1. RUNNING - In progress
  2. COMPLETED - Completed successfully
  3. FAILED - There was some error
3. Progress - Shows a percentage progress based on number of partitions processed. The processing divides the source data into multiple partitions for parallel processing.

In case of a FAILED status, run the following command to get more information on the error.

Bash	Copy
<pre>adectl show -e</pre>	

## Importing catalog

If there is external catalog information to be imported then prepare a CSV(comma-separated-values) file as per [Preparation](#). Run the following command to import the CSV file.

Bash	Copy
<pre>adectl import -f &lt;path-to-csv-file&gt; -d &lt;dataset-id&gt; -t &lt;table-name&gt;</pre>	

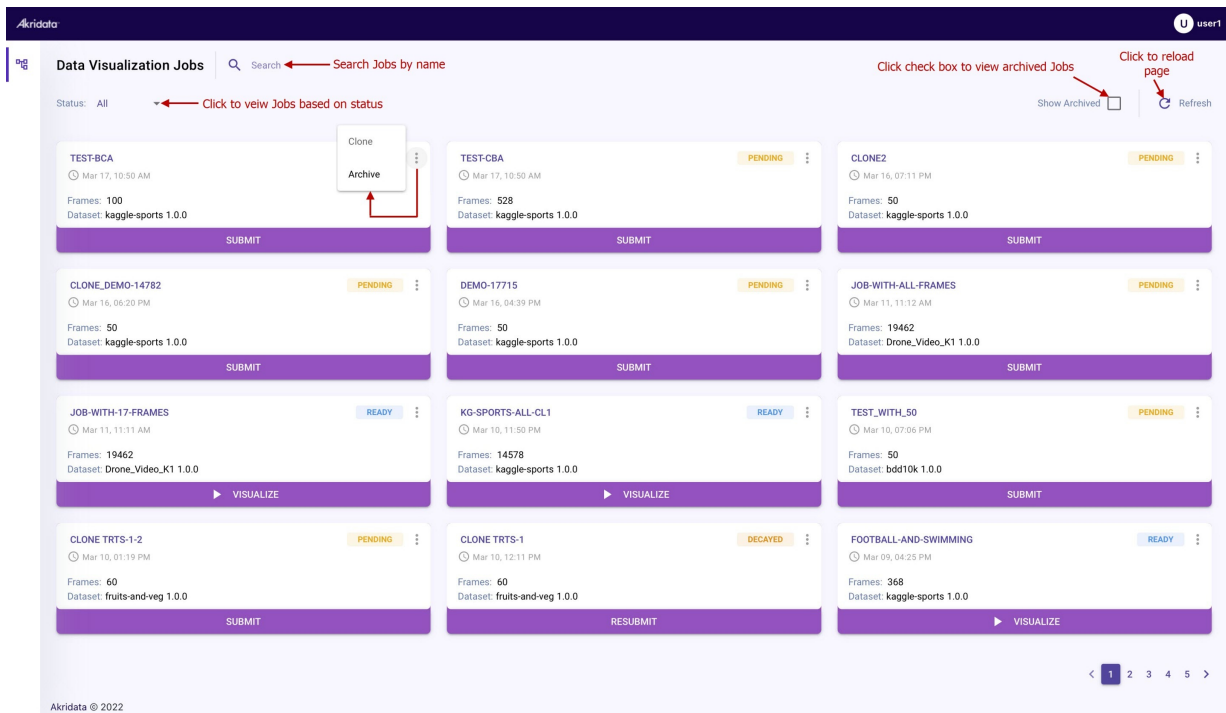
**<dataset-id>** - Please get this id from 'Datasets' page on Data Explorer UI

**<table-name>** - Name of catalog table to which the entries must be imported. This can be an existing table matching the schema of the CSV or if table does not exist, then the table is automatically created.

On successful import, the catalog entries are visible in the Dataset Catalog page.

# Jobs

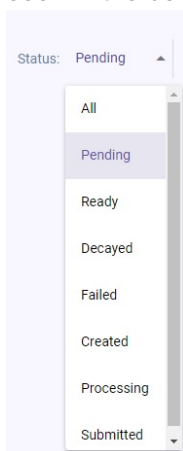
Jobs page is displayed with the options as seen in the below image,



## Jobs

Options displayed in the Data Visualization Jobs page:

- **Search** - This option is used to search jobs by name.
- **Show Archived** - This check box can be clicked to view archived jobs.
- **Refresh** - This option is clicked to reload the page.
- **Status** - This drop down is clicked to view jobs based on the status. It consists of options as seen in the below image.

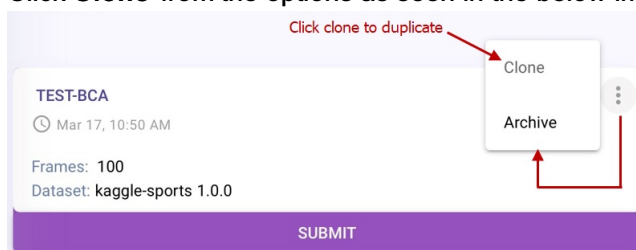


## Status

Options available for the jobs which are in 'Ready' state:

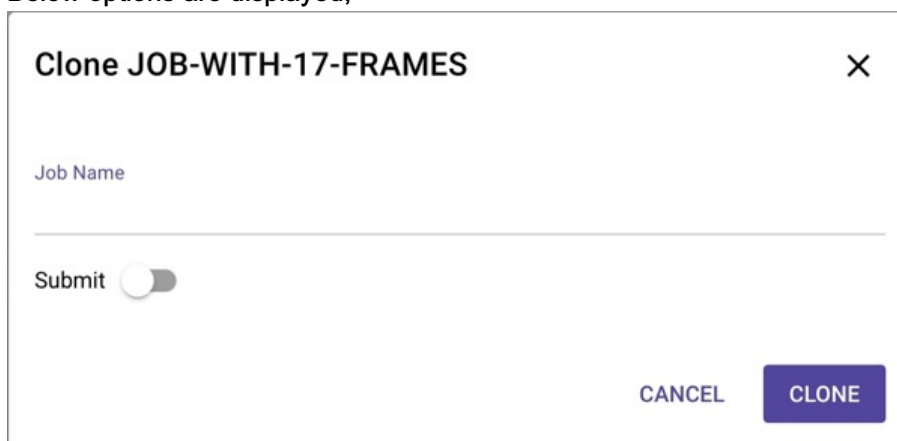
- **Clone** - This option is used to duplicate an existing job and submit the job with a different set of clustering and embedding parameters.

1. Click **Clone** from the options as seen in the below image,



### Ready Job

Below options are displayed,



### Clone

2. The job can be only cloned or cloned and submitted. Click on **Submit** slide bar to clone and submit the job.
  3. Enter a new Job name and click **Clone**.  
The Job is cloned and displayed with the new name in the Jobs page.
- **Archive** - This option is used to archive a job which is no longer used.
    1. Click **Archive** from the options as seen in the **Ready Job** image.  
The Job is archived and can be viewed by clicking on **Show Archived** checkbox in the Jobs page.

# Job creation and visualization

## Overview

The below animation provides an overview of the steps involved in browsing the catalog to create a job, submitting the visualization job and visualizing the job for exploration and further operations.



## Overview

## Job Creation

Follow the below instructions to create a job,

1. Navigate to the catalog as described in [Dataset catalog](#).
2. Edit filter to filter out results and make a selection of individual rows, multiple pages or all results on which visualization job needs to be created.
3. Click on the **Visualize** or **Visualize all rows** button depending on whether individual rows are selected or all rows are selected.
4. Enter job name and click **Create**.

Dataset: kaggle-sports Added Conditions ✎ Edit Filter

---

**annotations** Columns ↻

Select All 14572 Rows

Create a visualization job with all selected rows →

<input checked="" type="checkbox"/>	file_path	session_id	partition_start	partition_end	blob_idx_in_partition	frame_idx_in_blob	file_name	class_name
<input checked="" type="checkbox"/>	file/data/kaggle-sports/test/air hoc...	2022-02-26_05-08-29-9a6d5550-7a0...	0	19999999	0	6	1.jpg	air hockey
<input checked="" type="checkbox"/>	file/data/kaggle-sports/test/air hoc...	2022-02-26_05-08-29-9a6d5550-7a0...	0	19999999	0	7	2.jpg	air hockey
<input checked="" type="checkbox"/>	file/data/kaggle-sports/test/air hoc...	2022-02-26_05-08-29-9a6d5550-7a0...	0	19999999	0	8	3.jpg	air hockey

**VISUALIZE**

## Job Creation

## Job Submission

Navigate to 'Jobs' page and click **Submit** button.

**Data Visualization Jobs** cricket ✕

Status: All ▼

**CRICKET-ALL** PENDING ⋮

🕒 Mar 09, 11:43 PM

Frames: 140  
Dataset: kaggle-sports 1.0.0

**SUBMIT**

1. Click on Submit

**Submit CRICKET-ALL** ✕

Job Name  
CRICKET-ALL

Clusterer ▼

Embedder ▼

Advanced Options ^

Clustering Order  
Cluster After Embed ▼

Visualization Quality  
Medium ▼

3. Click Submit →

CANCEL **SUBMIT**

2. Select choices for various job parameters

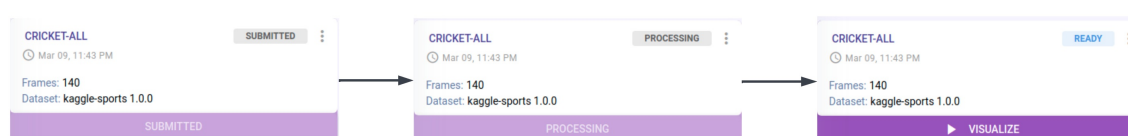
**Job Submit**The following options must be selected during **Submit** operation.



- **Clusterer** - Following clustering algorithms are available for selection,
  - K-means - This mode requires the number of clusters to be provided by the user.
  - HDBSCAN - This mode clusters the data as per the Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm. This generates clusters of non-spherical nature. The user isn't required to specify any initial number of clusters to work upon; rather, the appropriate number of clusters is computed implicitly.
  - K-segmentation - The K-Segmentation is a method of segmenting the data by considering the time dimension in addition to the feature data. This provides insights into what are the interesting divisions of the data and allows users to look at the data in a different perspective - boundaries where a segment transitions, the core of each segment amongst others. This is most suitable for video data.
- **Embedder** - The following embedding algorithms are available for selection,
  - UMAP
  - Principal Component Analysis(PCA)
  - Locally linear embedding(LLE)
 Based on the clustering algorithm selected a recommendation on the embedding algorithm is automatically provided which can be overridden by the user.
- **Advanced Options-**
  - Clustering before embedding - The default choice is to cluster after embedding the features into 2D space. This provides good visual separation of clusters and hence allows easier navigation on the UI. The default can be overridden which may give better assignment of objects to clusters due to additional features being used at the expense of clean visual separation on the UI.
  - Visualization quality - This is a tunable that provides a trade-off between visualization speed and quality. In general the better the visualization quality selected, the slower is the visualization speed.

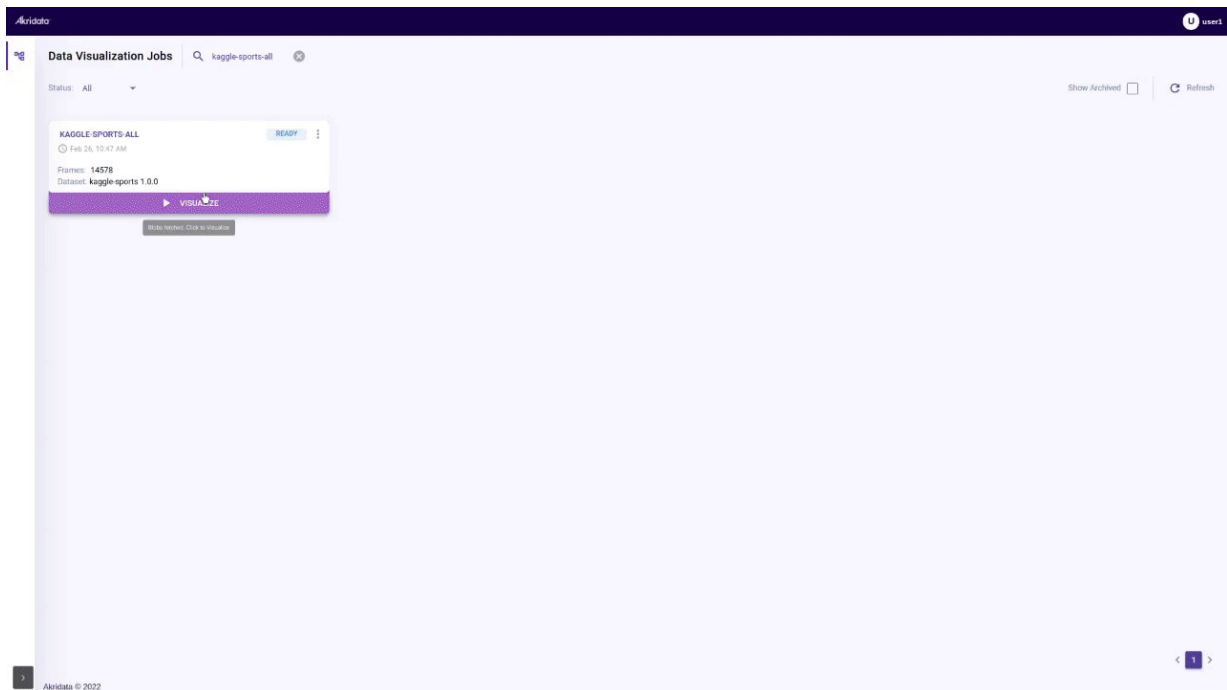
## Job Visualization

Once submitted, the job is processed in the background and goes through following state transitions,



**Job State transitions when processing** Once the job transits to 'READY' state, click the **VISUALIZE** button.

# Select and Refine



## Select and refine operations

The 'Visualize' operation shows the visualization view that has following elements and controls.

### Plot view and sampling modes

The plot view shows the distribution of points with colors representing the clusters and an outlier category of points that don't belong to any cluster. Each point in the plot view is clickable and this action populates the right 'selection' panel with sampled points in relation to the clicked point. The plot view supports zooming in/out and panning to get to the most relevant area representing the data points of interest. The points can be sampled using the following modes,

- **KNN** - Nearest neighbours around the clicked point are sampled.
- **Cluster** - Uniform distribution of points from the cluster to which the clicked point belongs to are sampled.
- **Random** - A random set of points are sampled.
- **Manual** - The clicked point is sampled.

### Selection(thumbnail) panel

The right panel shows the sampled thumbnails and has following controls,

- **Sample size setting** - Controls the number of points to be sampled in response to a click in plot view.
- **Clear points** - Clear points from selection panel.
- **Highlight points** - Highlight points in plot view corresponding to thumbnails in selection panel.



### Thumbnail actions

Each thumbnail has an action bar with following actions,

- Remove this point from selection.
- Add to similarity search as a positive sample.
- Add to similarity search as a negative sample.
- Show full resolution image for this thumbnail.
- Add to resultset

### Detailed view



### Detailed view

### Cluster-by(Color-by)

The bottom left part of the view has a Cluster-by option to color the points in plot view based on the following attributes,

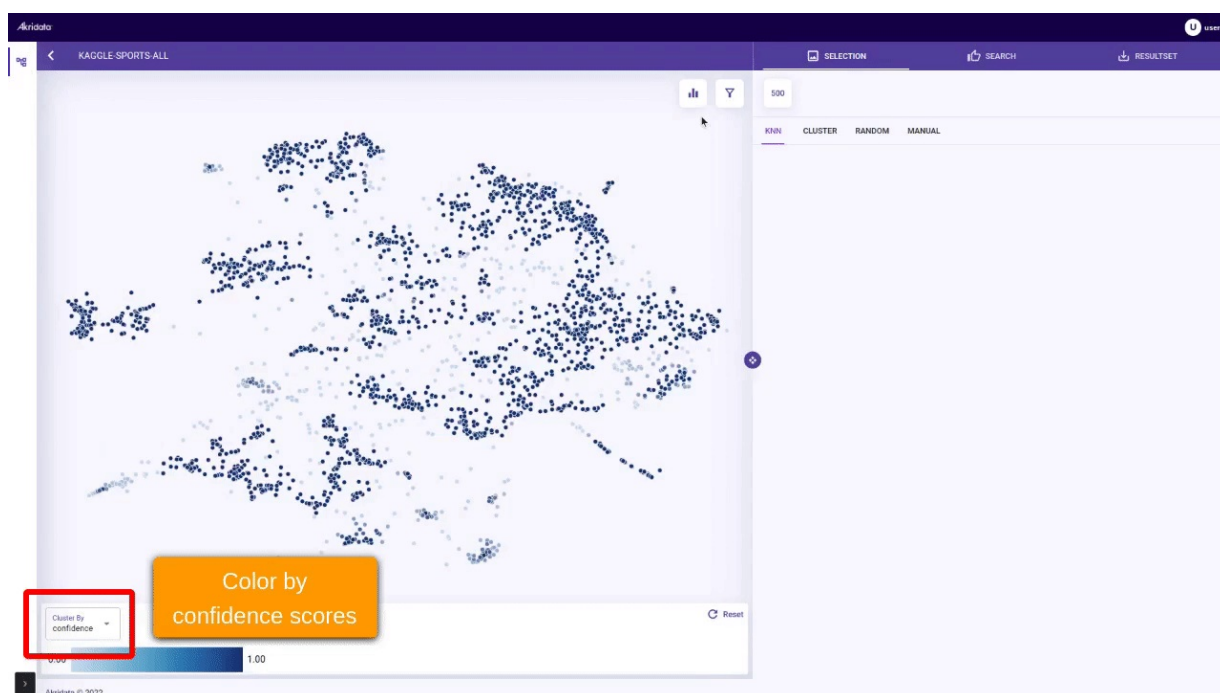
- **partition-id** - Refer to [partition](#). This mode colors the plot view based on partition to which point belongs to. This is useful for cases where ingested data is sorted based on timestamp or other attributes which results in a partition holding neighbouring points in the sorted order. If colors are spread across the plot view, then it indicates that each partition has a lot of variety versus colors grouping together indicates that objects in a partitions are very similar to each other.
- **weight** - Data explorer uses 'coresets' to keep a subset of points as representatives and each representative is assigned a 'weight' based on how many other points it represents.
- **Cluster(HDBSCAN)** - Cluster id to which point belongs to.
- **confidence** - The algorithm's confidence on it's cluster assignment to a point.

## Tunables

The **tunables** button provides the following controls to filter the points in plot view,

- **Number of clusters** - Change the number of clusters based on intent towards fine grained or coarse grained grouping of points.
- **Sampling modes**- The following probabilistic sampling modes are available,
  - a. **Inlier** - Points that are strong inliers to some cluster.
  - b. **Outlier** - Points that don't belong to any cluster.
  - c. **Bimodal** - Points that are either strong inliers or outliers.
  - d. **Normal** - Sample points using a normal distribution.
- **Sampling fraction** - Fraction of total points to be sampled.
- **Sampling weight** - Extent of preference to be given to the selected sampling mode. The higher the number, stronger is the preference towards selected sampling mode.

The below picture shows sampling in action with Cluster-by option selected to color points by confidence scores. Since the sampling weight was set to highest allowed value (and hence indicating strong preference to sampling mode), it can be seen that inlier sampling chooses points with high confidence scores and outlier sampling chooses points with low confidence scores.



## Ksegmentation specific sampling modes

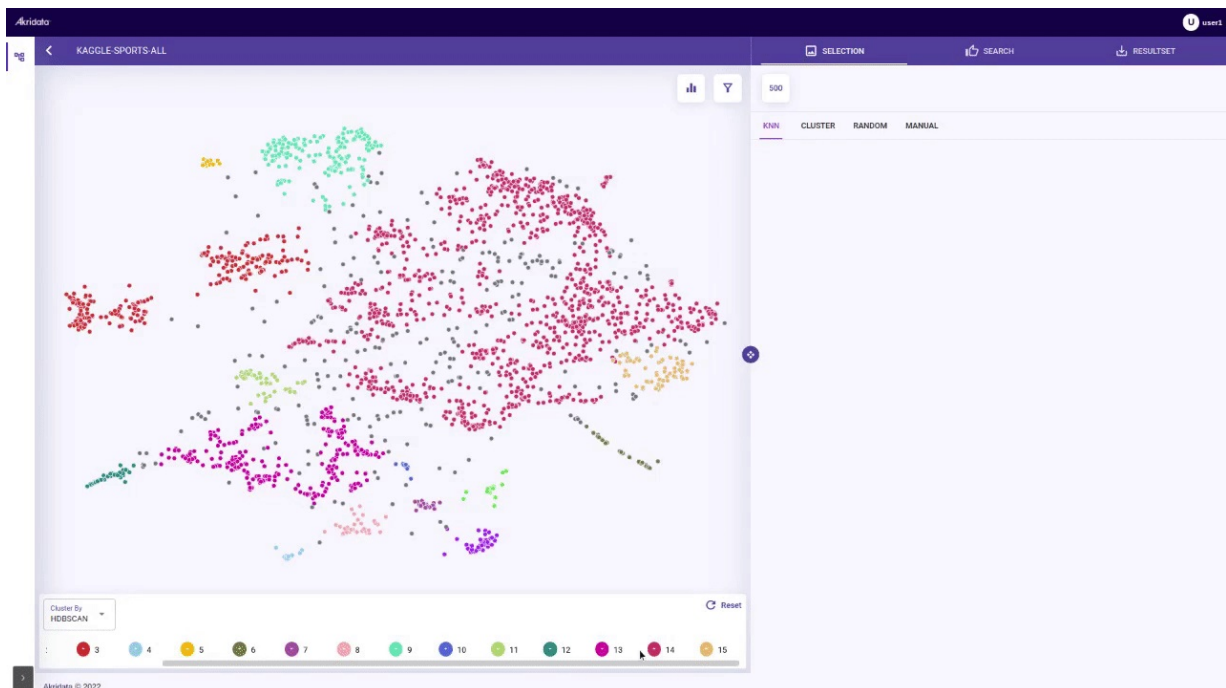
- **Edge** - Points that are at cluster boundaries representing transitions.
- **Line** - A regression line(trendline) for each cluster is drawn and this sampling mode prefers points that are close to this line.
- **Core** - Points that are away from the cluster edges.

## Filters

The 'Filter' button provides the following filtering criteria selection,

- **Partition ID**
- **Cluster** - Choose a subset of clusters to be displayed.
- **Confidence** - Choose only those points that have a clustering confidence within the selected range.
- **Weight** - Data explorer uses 'coresets' to keep a subset of points as representatives and each representative is assigned a 'weight' based on how many other points it represents.

## Splitting and merging clusters



**Cluster split and merge** If there is a large cluster with many points, it might help to split the cluster into sub-clusters for sampling and refinement. The above graphic shows the steps to split the cluster. The reverse operation of merging a cluster with rest of the clusters is also supported.

## Adding sampled points to a resultset

A resultset represents a curated set of points. From the selection panel, the points can be added to a resultset using controls highlighted in the below picture.

SELECTION SEARCH RESULTSET 1

500

Number of objects selected for adding to resultset

KNN CLUSTER RANDOM MANUAL add/remove individual object

add/remove individual object

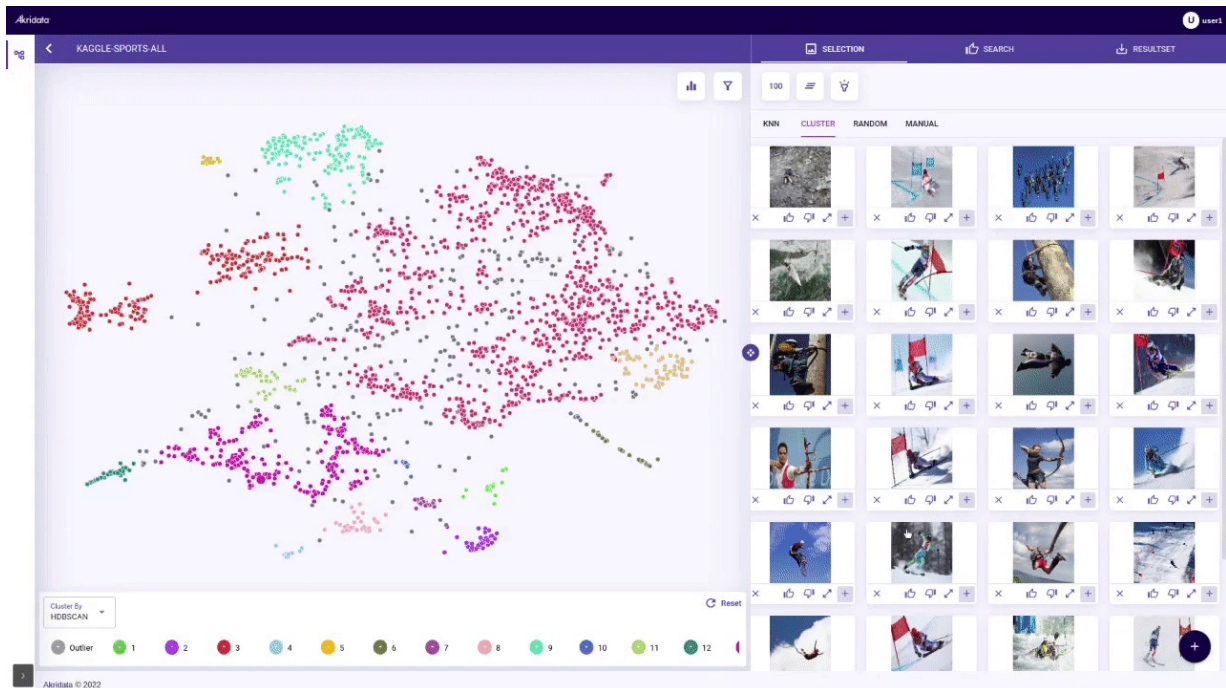
bulk add

Cluster All 500 Points Top 25

add-objects-to-resultset

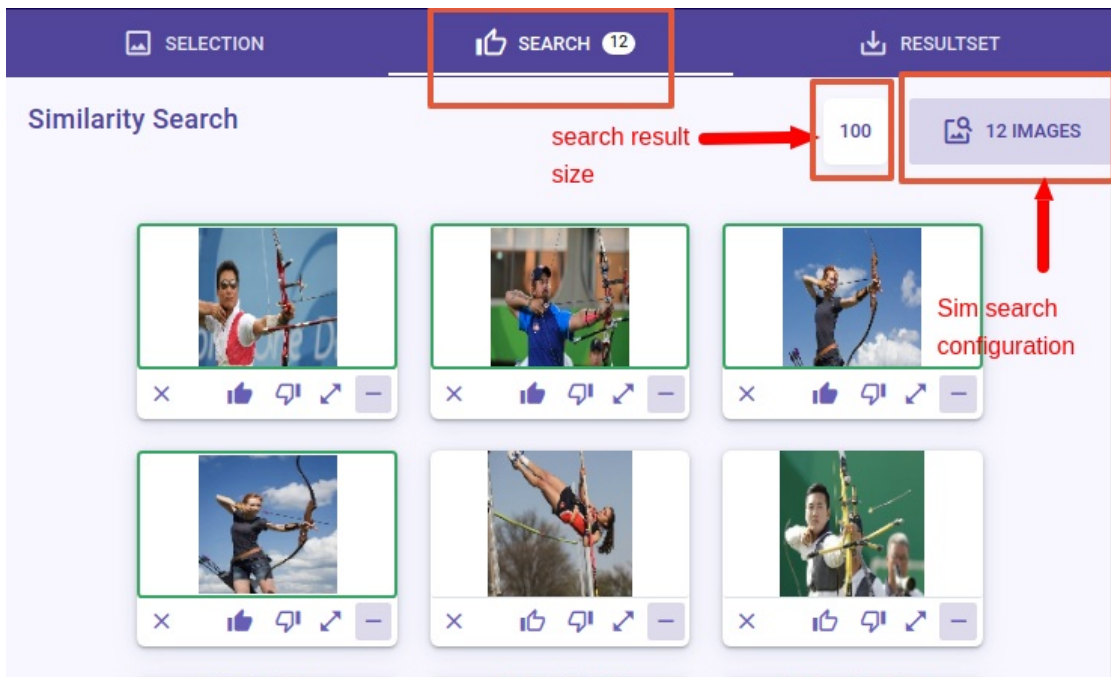


# Similarity search



**Similarity-search** Similarity search allows refining the objects of interest by expressing like and dislike inputs on a small subset of objects. The above image shows similarity search in action to find 'archery' images through iterative refinement of results and user feedback on like and dislikes.

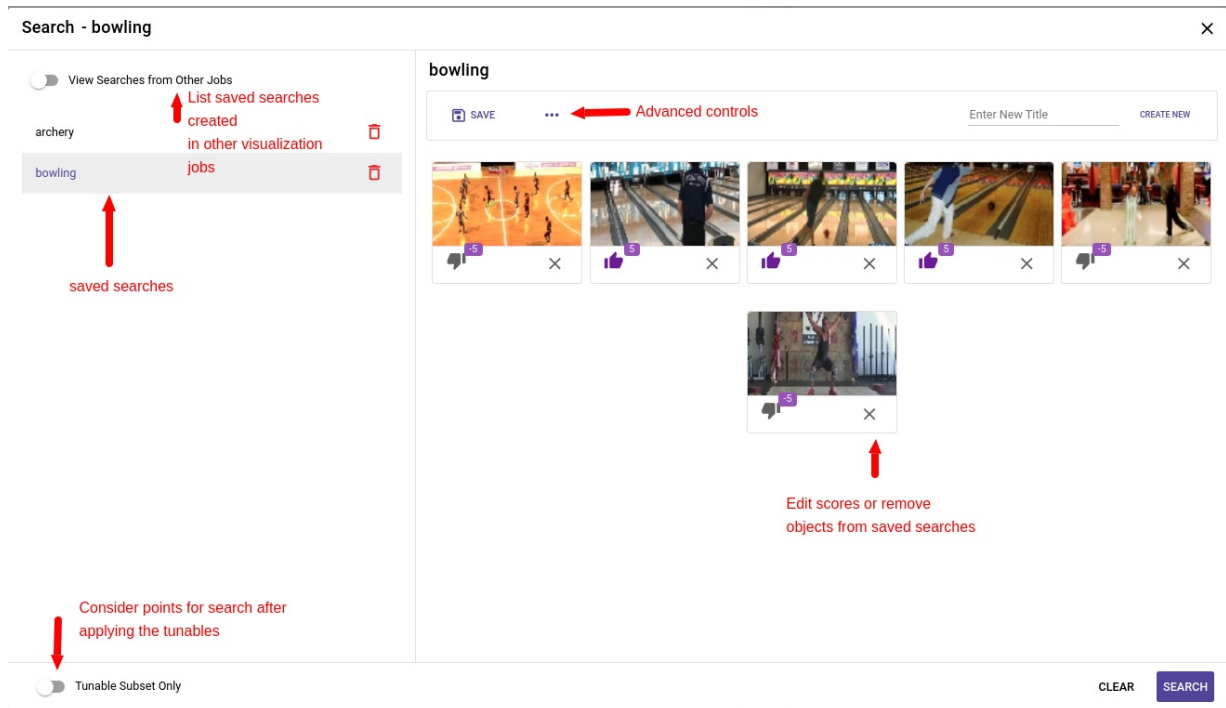
## Similarity search tab



similarity-search-controls-on-main-view

## Similarity search configuration

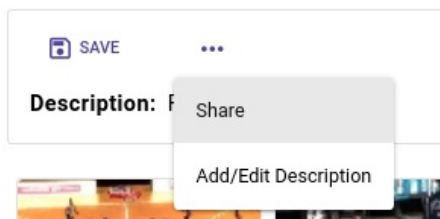
The similarity search configuration page allows detailed configuration of similarity search request. It also allows loading existing similarity searches and saving current similarity search as shown in the below image,



Similarity search configuration page

The advanced controls provide the following options,

- **Share** - Make this similarity search available to all users on the system. This is not relevant in local mode deployment since there is only one user supported in local mode.
- **Add/Edit Description** - Enter a descriptive text as a note about the similarity search request.



Similarity search advanced configuration

## Adding results to a resultset

After iteratively refining the search results, the curated list of objects can be added to an existing or a new [resultset](#) using the controls shown in the below image,



The screenshot displays the Akridata interface for a dataset named "KAGGLE-SPORTS-ALL". On the left, a scatter plot shows data points clustered into 8 groups, color-coded from 1 (green) to 8 (red). A legend below the plot indicates the clustering method is "DBSCAN" and lists the clusters. On the right, a "Similarity Search" panel shows a grid of 100 image thumbnails. Each thumbnail has a set of interaction icons: a close button (x), a thumbs-up icon, a thumbs-down icon, a magnifying glass, and a plus sign. Two red arrows point from the scatter plot to the similarity search grid. One arrow points to a specific image thumbnail, with the text "add individual objects to resultset" next to it. The other arrow points to the close button (x) of the same thumbnail, with the text "Manually remove elements from result" next to it. At the bottom right of the similarity search grid, there are buttons for "All 100 Points" and "Top 74", along with a close button (x).

## Add results to a resultset