

Visual Flow User Guide

May 2021

Version 0.4

Document Revisions

Date	Version Number	Document Changes
08/12/2020	0.1	Initial Draft
04/22/2021	0.2	Pipeline Operators
04/26/2021	0.2	Job Operators
05/07/2021	0.3	Project Name, Project Operations
05/25/2021	0.4	Project Name in document

Table of Contents

1	Introduction	4
1.1	...Terminology	4
1.2	...Scope and Purpose.....	4
1.3	...Process Overview	5
2	Roles and Authorizations	5
3	Project Operations	6
3.1	...Create Project	6
3.2	...Project Overview.....	8
3.3	...Manage Project Settings.....	8
4	Job Operations	10
4.1	...Jobs Overview	10
4.2	...Create a Job	11
4.3	...Job Designer Functions Overview.....	16
4.4	...Job Execution	17
5	Pipeline Operations.....	17
5.1	...Pipelines Overview.....	17
5.2	...Create a Pipeline	18
5.3	...Pipeline Designer Functions Overview	21
5.4	...Pipeline Execution.....	22

1. Introduction

1.1. Terminology

ETL is abbreviation for *extract, transform, load*, three database functions combined into one tool to pull data out of one database, transform it and place it into another database.

- **Extract** is the process of *reading data* from a database. In this stage, the data is collected, often from multiple and different types of sources.
- **Transform** is the process of *converting the extracted data* from its previous form into the form needed to place it into another database.
- **Load** is the process of *writing the data* into the target database.

Job is a chain of individual stages linked together. It describes the flow of data from a data source to a data target. Usually, a stage has minimum of one data input and/or one data output. However, some stages can accept more than one data input and output to more than one stage.

In Visual Flow various stages you can use are:

- Read stage
- Write stage
- Join stage
- Union stage
- Filter stage
- Group By stage
- Remove Duplicates stage
- Transformer stage
- Change Data Capture stage

Pipeline is a compound of multiple jobs and can run on schedule.

1.2. Scope and Purpose

Visual Flow web application is ETL tool designed for effective data manipulation via convenient and user-friendly interface.

The tool has the following capabilities:

- Can integrate data from heterogeneous sources:
 - ✓ DB2
 - ✓ COS
 - ✓ Elastic Search
- Leverage direct connectivity to enterprise applications as sources and targets
- Perform data processing and transformation

- Leverage metadata for analysis and maintenance

1.3. Process Overview

Visual Flow jobs and pipelines exist within a certain namespace (project) so the first step in the application would be to create a project or enter existing project. Then you need to enter Job Designer to create a job.

Job Designer is a graphical design interface used to create, maintain, execute and analyze jobs. Each job determines the data sources, the required transformations and destination of the data. Designing a pipeline is similar to designing a job.

Pipeline designer is a graphical design interface aimed for managing pipelines.

Visual Flow key functions include but not limited to

- ✓ Create project which serves as a namespace for jobs and/or pipelines
- ✓ Manage project settings
- ✓ User access management
- ✓ Create/maintain a job in Job Designer
- ✓ Job execution and logs analysis
- ✓ Create/maintain a pipeline in Pipeline Designer
- ✓ Pipeline execution
- ✓ Import/Export jobs and pipelines

2. Roles and authorizations

The following roles are available in the application:

- ✓ Viewer
- ✓ Operator
- ✓ Editor
- ✓ Administrator

They can perform the below operations within the namespaces they are authorized to.
Only Super-admin user can create a workspace (project) and grant access to this project.

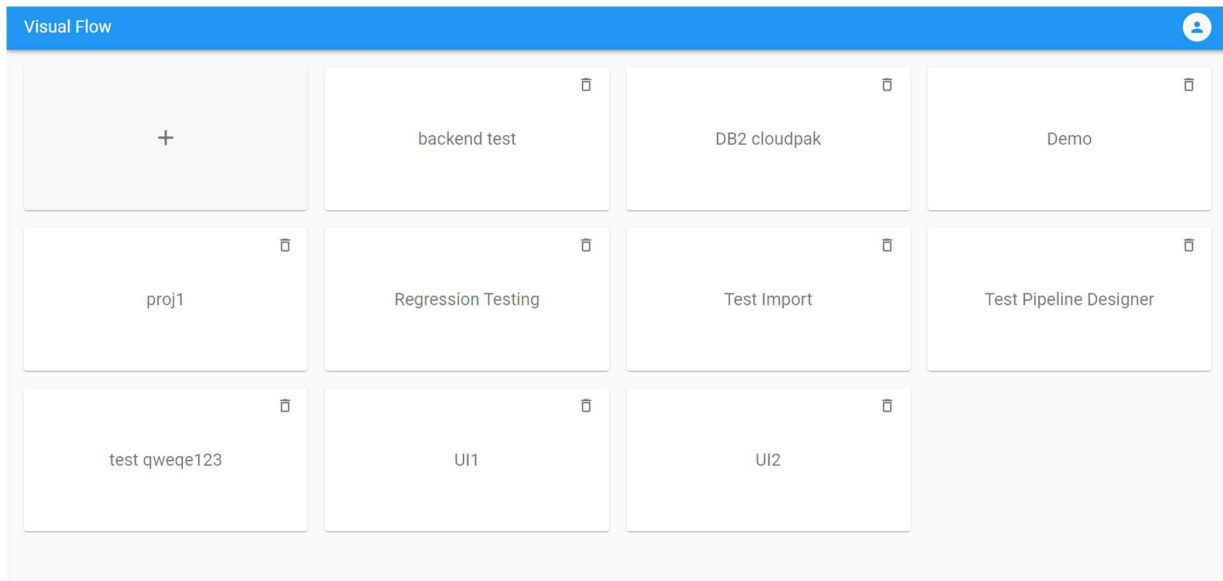
Role	Actions		
	Project Settings	Jobs	Pipelines
Viewer	View all	View all	View all
Operator	View all	View all / execute jobs	View all / execute pipelines
Editor	Edit all but Users and Roles	Edit / execute jobs	Edit / execute pipelines
Admin	Edit all	Edit / execute jobs	Edit / execute pipelines

3. Project operations

3.1. Create a Project

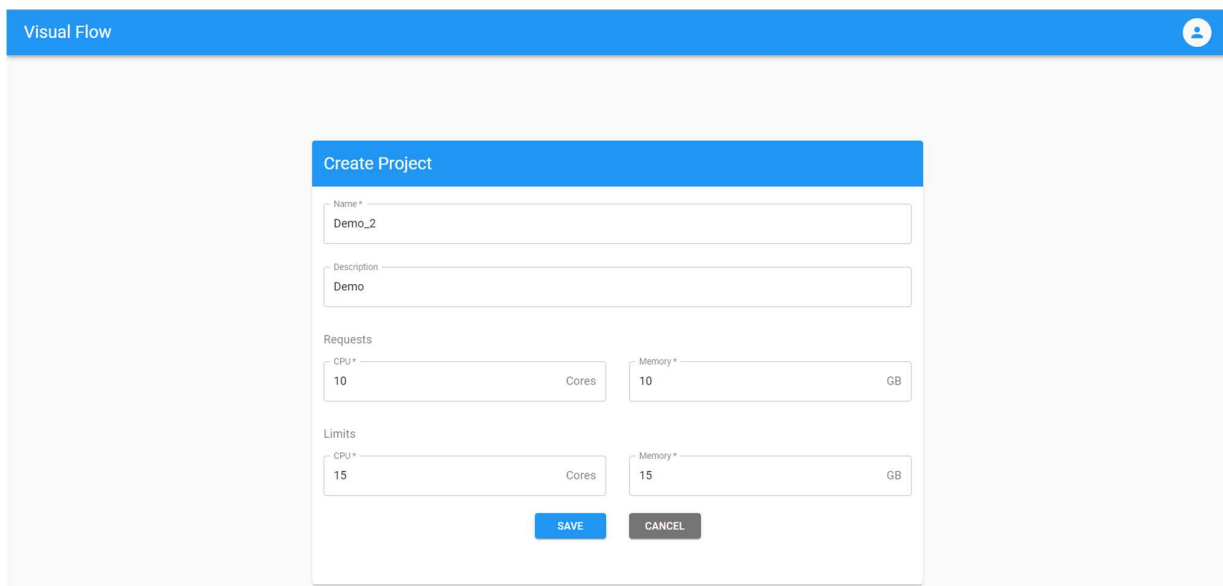
To create a project you need to push “+” button on initial screen.

Note: this is an action of super-admin user only. The button is not visible for the application roles (Viewer, Operator, Editor, Admin) .

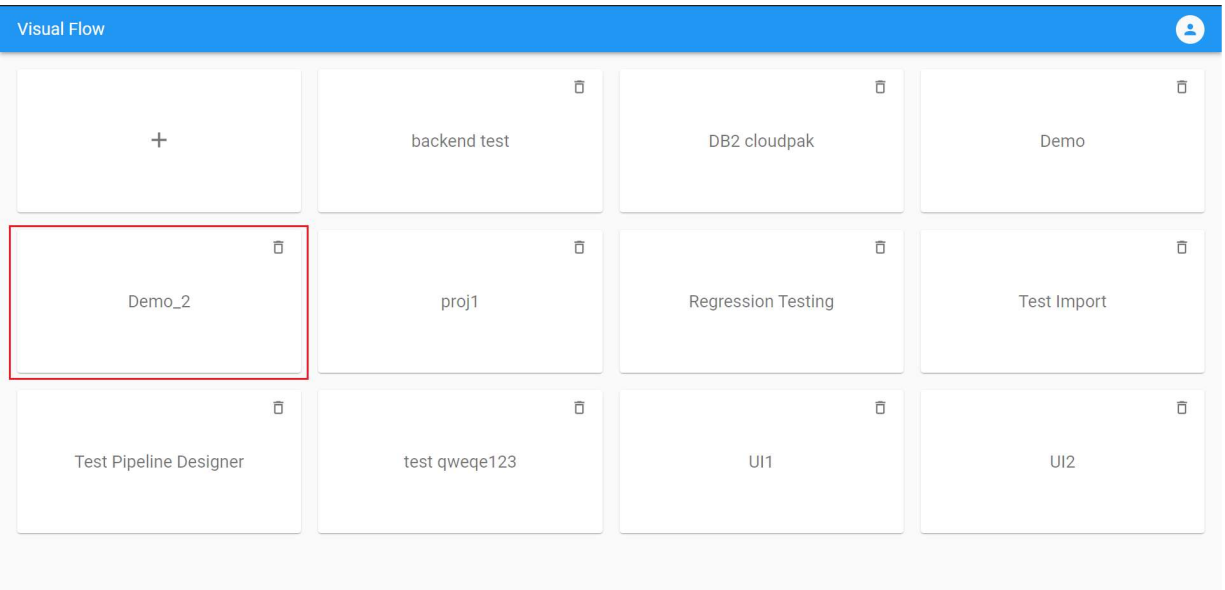


With “+” button pushed you will get to *Create Project Form* to enter project basic settings:

- Project Name
- Project Description
- Requests (CPU/Memory)
- Limits (CPU/Memory)

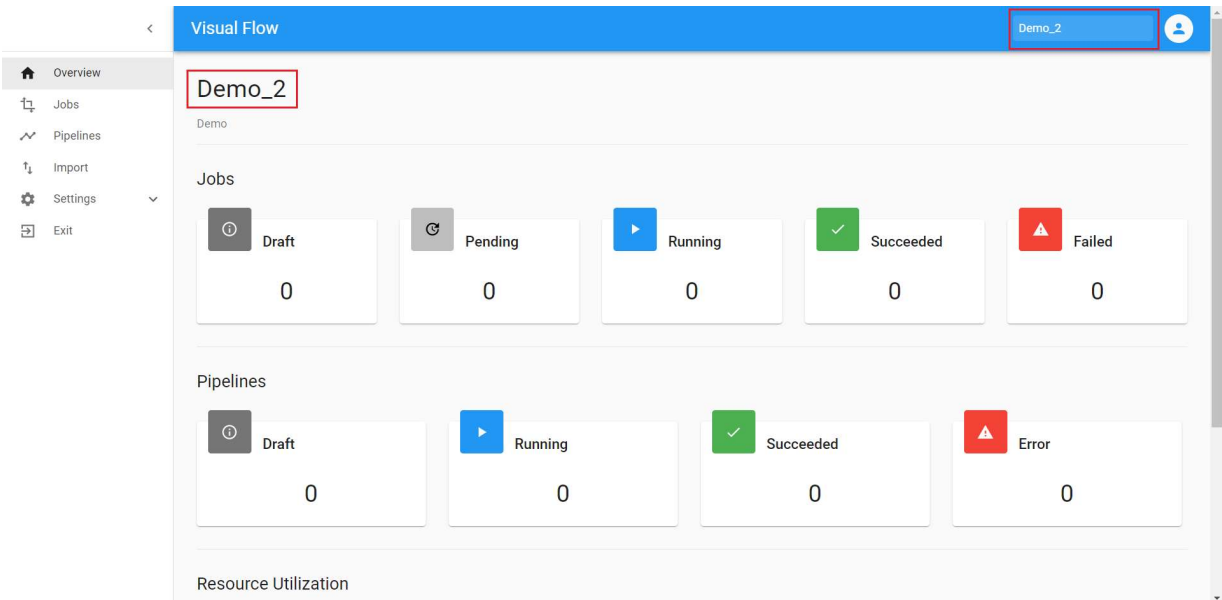
The screenshot shows the 'Create Project' form. It has a blue header with the text 'Create Project'. Below the header are several input fields: 'Name *' with the value 'Demo_2', 'Description' with the value 'Demo', 'Requests' section with 'CPU *' (10 Cores) and 'Memory *' (10 GB), and 'Limits' section with 'CPU *' (15 Cores) and 'Memory *' (15 GB). At the bottom are two buttons: 'SAVE' (blue) and 'CANCEL' (gray).

After saving *Create Project Form* the project created under the given name and then can be found on the initial screen:



3.2. Project Overview

Click the project card to enter the newly created project and you will get to the *Project Overview Screen*:

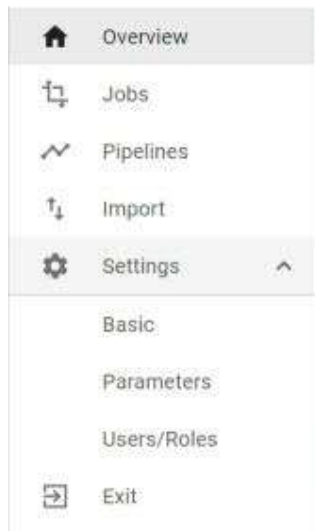


The screen contains project left menu and displays information about the project jobs, pipelines and their resource utilization (applicable for running jobs).

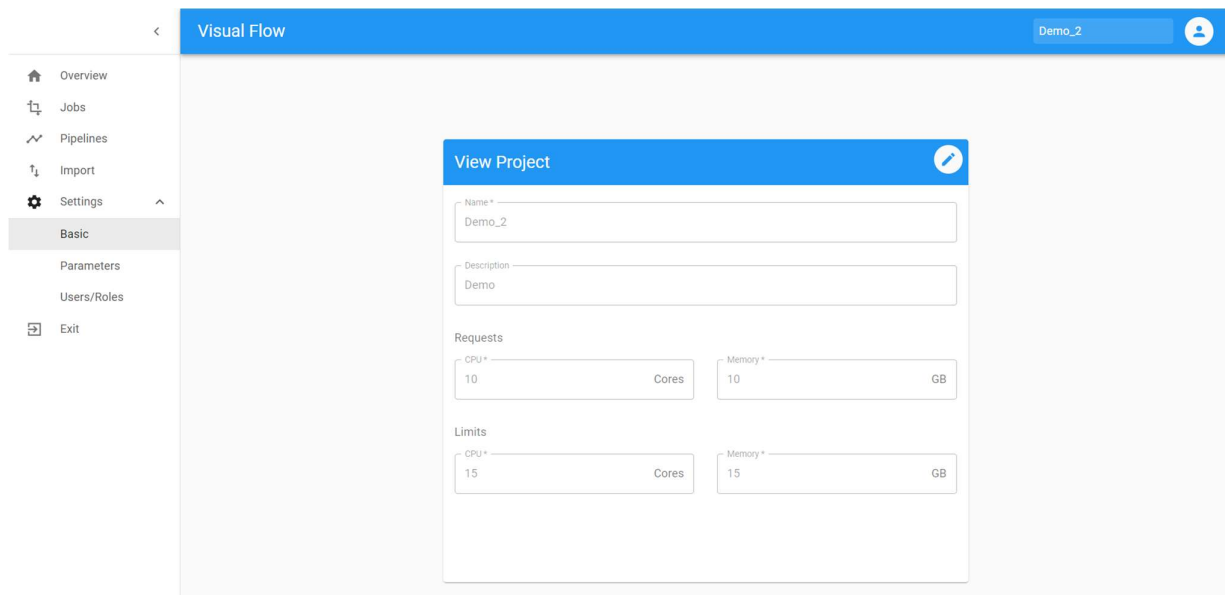
3.3. Manage Project Settings

Settings submenu contains:

- Basic
- Parameters
- Users and Roles

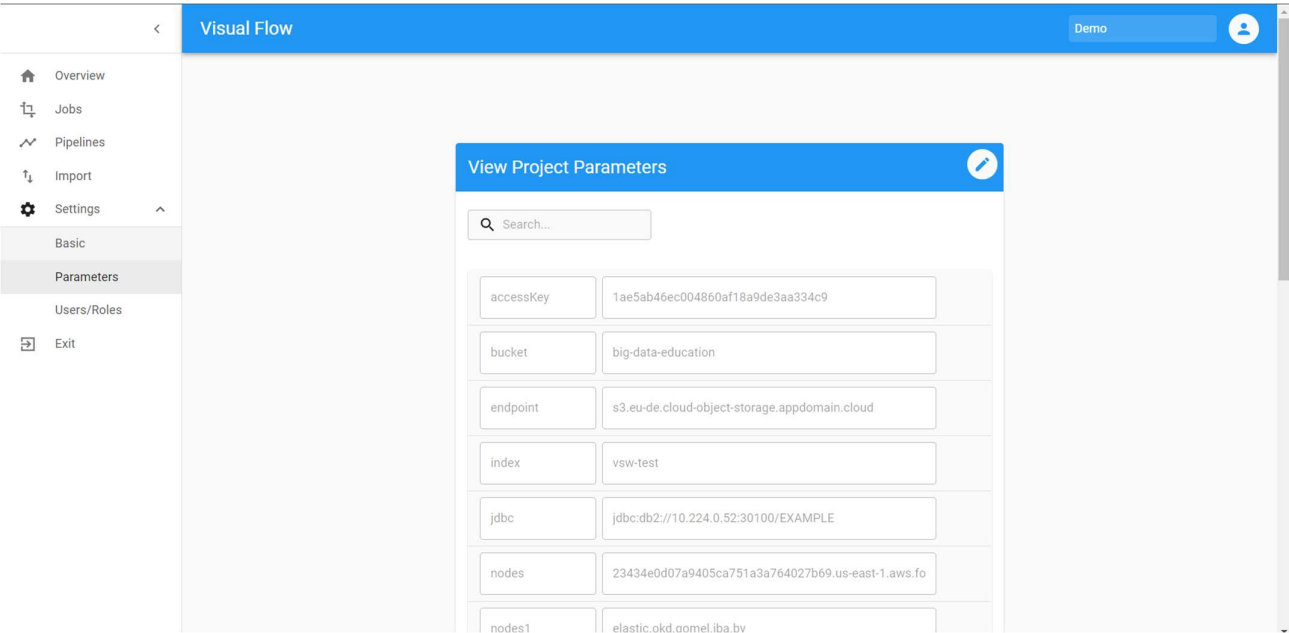


- 1) *Basic* is already there after project creation. *Edit* button turns on the edit mode for updates.

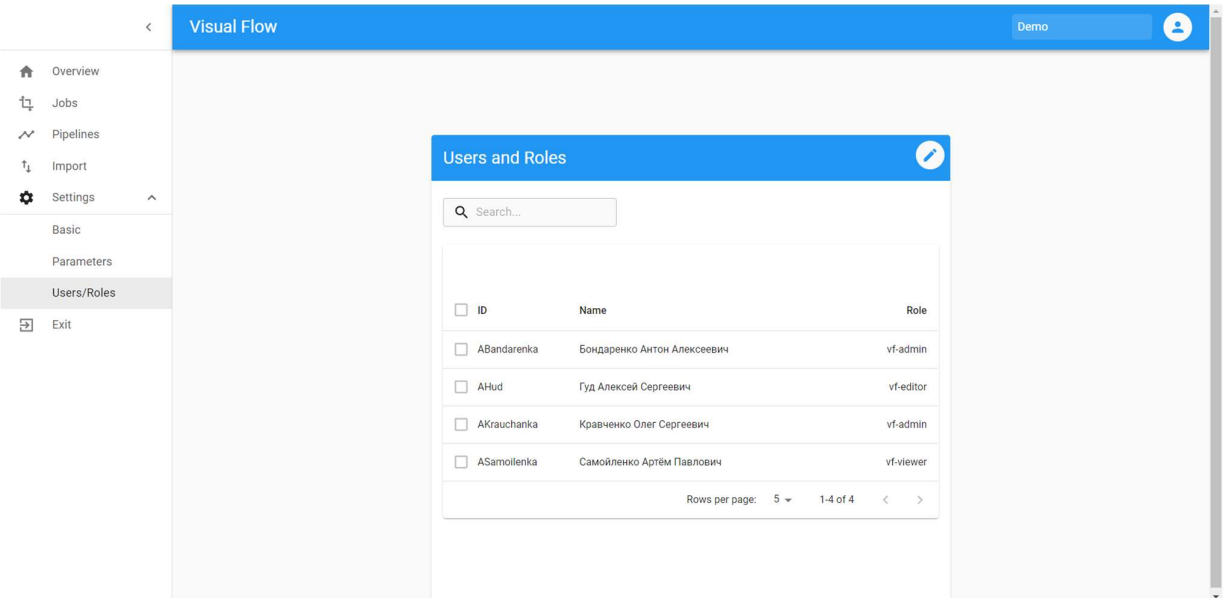


- 2) *Parameters* serve to store values required for the entire project, e.g. JDBC connection, DB2 credentials or table schemas can be the same for all jobs within the

project and therefore stored at the project level. *Edit* button turns on the edit mode for updates.




- 3) *User and Roles* allows user access management or view user access depending on authorization.
Edit button and therefore Edit mode is only available for admin within the project or super-admin.



4. Job Operations

4.1. Jobs Overview

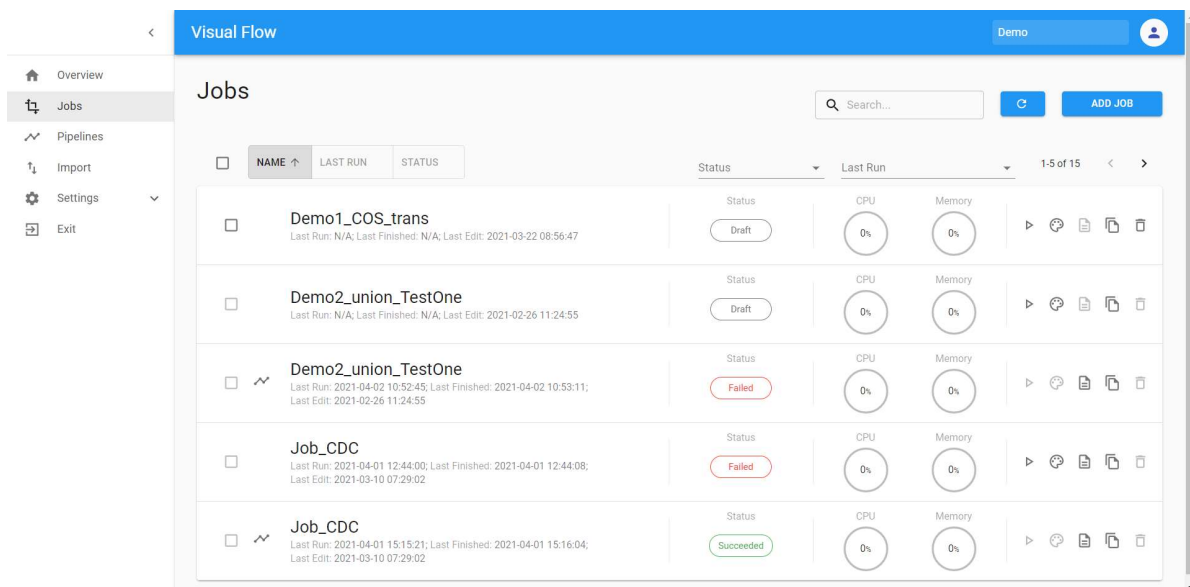
Clicking *Jobs* menu item will lead you to *Jobs Overview Screen*, which allows you to see a list of jobs existing within a project. Some of the jobs can be used in pipelines, this is indicated by the  icon.

Jobs Overview Screen displays the following information:

- Job Name
- Job Last run/Last finished/Last edit
- Job Status (Draft/Running/Succeeded/Failed)
- Resource Utilization (CPU/Memory)
- Available Actions (Run/Job Designer/Logs/Copy/Delete)

Notes:

- The actions availability and therefore visibility is depending on user authorizations
- You cannot delete job that is used in pipeline



The screenshot shows the 'Visual Flow' interface with a sidebar on the left containing 'Overview', 'Jobs', 'Pipelines', 'Import', 'Settings', and 'Exit'. The 'Jobs' section is active, displaying a table of jobs. The table has columns for 'NAME', 'LAST RUN', 'STATUS', 'CPU', and 'Memory'. There are five jobs listed, each with a checkbox, a status button, and resource utilization gauges. The first two jobs are in 'Draft' status, the third is 'Failed', and the last two are 'Succeeded'. The third job has a wavy icon next to its name, indicating it is used in a pipeline.

<input type="checkbox"/>	NAME	LAST RUN	STATUS	CPU	Memory	Actions
<input type="checkbox"/>	Demo1_COS_trans <small>Last Run: N/A; Last Finished: N/A; Last Edit: 2021-03-22 08:56:47</small>		Draft	0%	0%	▶ ⚙️ 📄 🗑️
<input type="checkbox"/>	Demo2_union_TestOne <small>Last Run: N/A; Last Finished: N/A; Last Edit: 2021-02-26 11:24:55</small>		Draft	0%	0%	▶ ⚙️ 📄 🗑️
<input type="checkbox"/>	Demo2_union_TestOne <small>Last Run: 2021-04-02 10:52:45; Last Finished: 2021-04-02 10:53:11; Last Edit: 2021-02-26 11:24:55</small>		Failed	0%	0%	▶ ⚙️ 📄 🗑️
<input type="checkbox"/>	Job_CDC <small>Last Run: 2021-04-01 12:44:00; Last Finished: 2021-04-01 12:44:08; Last Edit: 2021-03-10 07:29:02</small>		Failed	0%	0%	▶ ⚙️ 📄 🗑️
<input type="checkbox"/>	Job_CDC <small>Last Run: 2021-04-01 15:15:21; Last Finished: 2021-04-01 15:16:04; Last Edit: 2021-03-10 07:29:02</small>		Succeeded	0%	0%	▶ ⚙️ 📄 🗑️

4.2. Create a Job

With *Add Job* button pushed you will get to *Job Designer* for creating a new job.

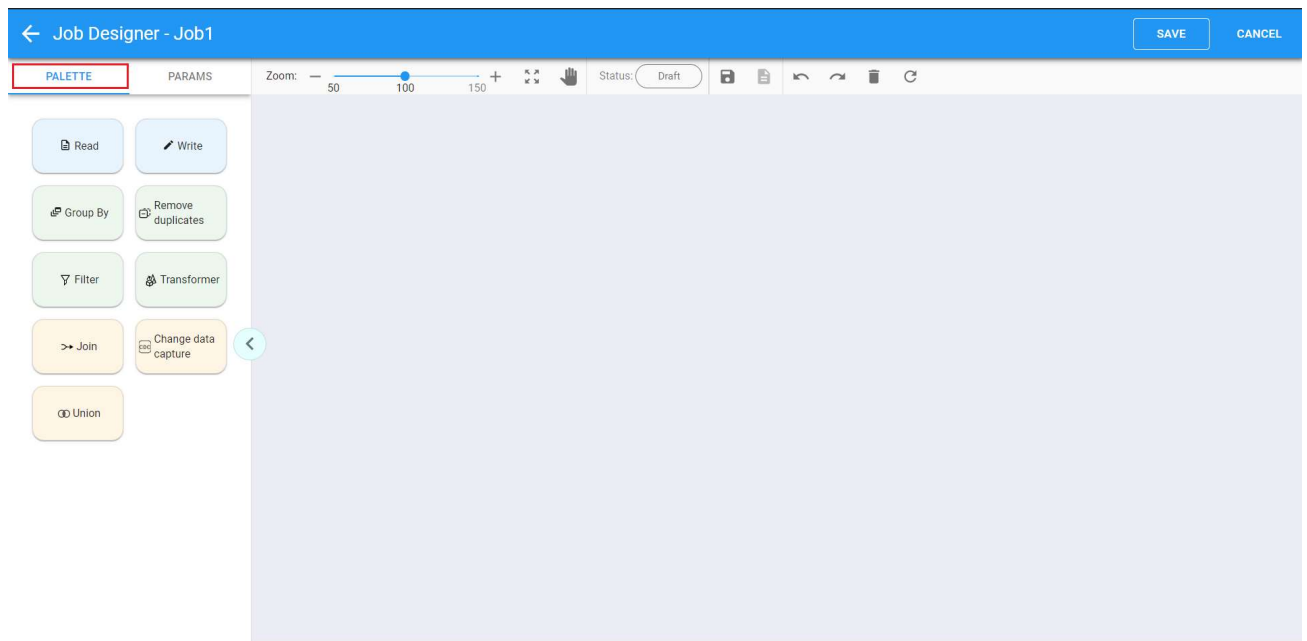
- 1) On the left configuration panel, you will need to give job a name, update parameters or keep their default values and then push *Confirm* on the panel:

The image displays two screenshots of the Job Designer interface, illustrating the process of saving a job.

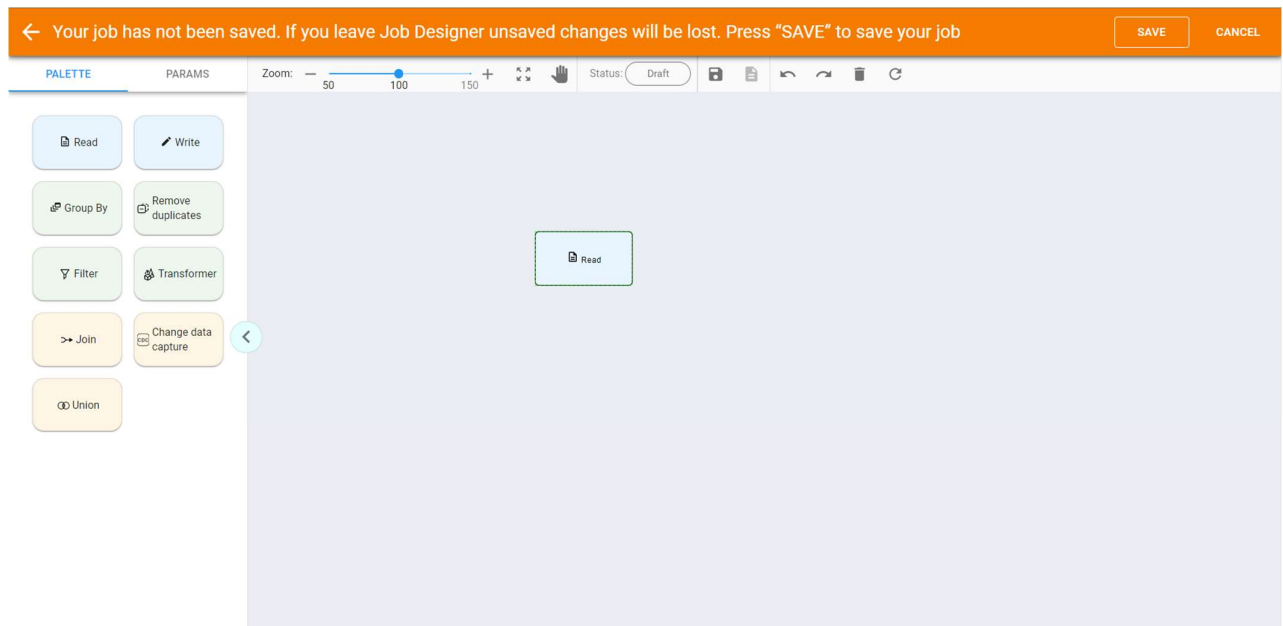
Top Screenshot: The header bar is orange and contains the text "Please enter name and save params" on the left and "SAVE" and "CANCEL" buttons on the right. The main area shows a configuration sidebar on the left with fields for "Name" (Job1), "Driver Request Cores" (0,1), "Driver Cores" (1), "Driver Memory" (1 GB), "Executor Request Cores" (0,1), "Executor Cores" (1), "Executor Memory" (1 GB), "Executor Instances" (2), and "Shuffle Partitions" (10). Below these fields are "CONFIRM" and "DISCARD" buttons. The main canvas area is empty, showing a zoom slider (50 to 150) and a status bar (Draft).

Bottom Screenshot: The header bar is orange and contains the text "Your job has not been saved. If you leave Job Designer unsaved changes will be lost. Press 'SAVE' to save your job" on the left and "SAVE" and "CANCEL" buttons on the right. The main area shows the same configuration sidebar and main canvas area as the top screenshot, but the "CONFIRM" and "DISCARD" buttons are now disabled (grayed out).

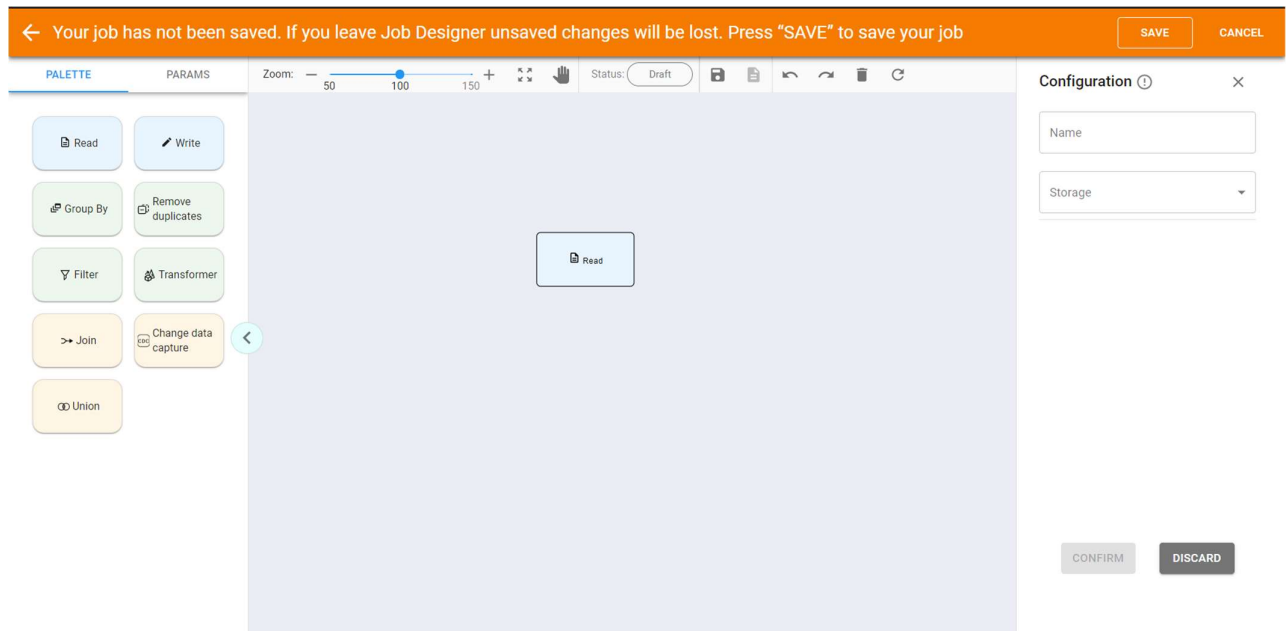
- 2) Save the job by pushing *Save* button on the *Job Designer* header.
- 3) Go to *Palette* tab to see all available stages:



- 4) You can start creating a job by dragging a stage to the canvas, e.g. you can drag *Read* stage:



- 5) Double-click on the stage will open the configuration panel on the right:




- 6) Enter name for the stage and select *Storage* DB2 if you want to read data from DB2 table.

Available *Storage* values for read stage are:



- ✓ DB2
- ✓ COS
- ✓ Elastic Search

- 7) Fill required parameters for DB2 *Storage*.

Important: you can pick up a parameter value with *Parameters*  button on the right panel if you have it previously created as project parameters.

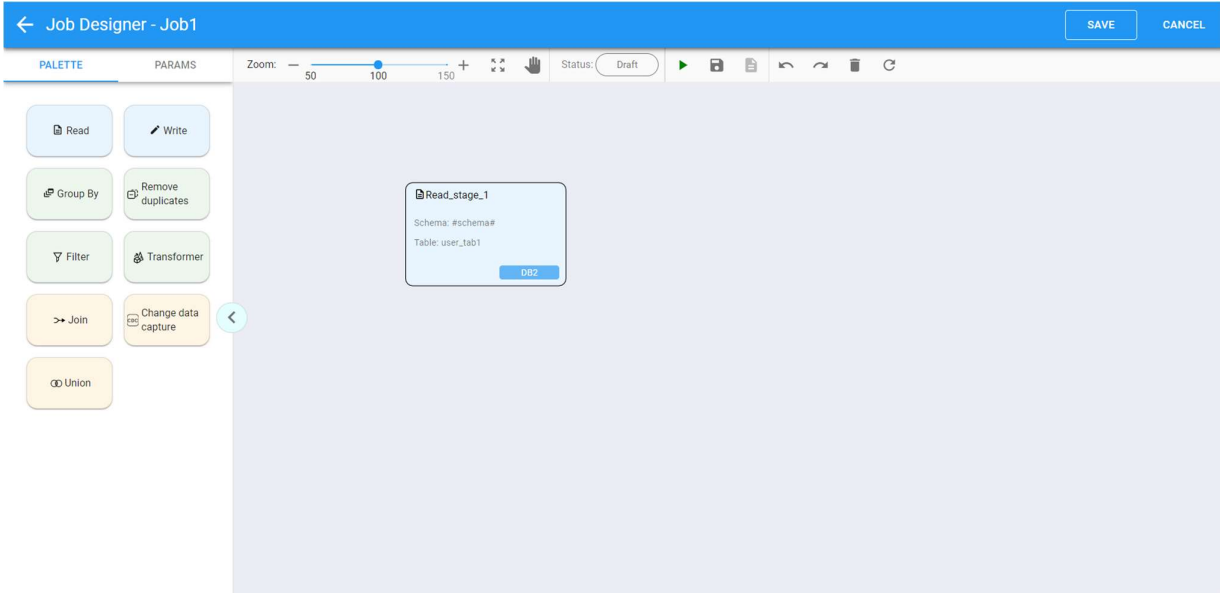


The Configuration panel for 'Read_stage_1' shows the following fields:

- Name: Read_stage_1
- Storage: DB2 (selected in a dropdown)
- JDBC URL:  (highlighted with a red box)
- User: 
- Password: 
- Schema: 

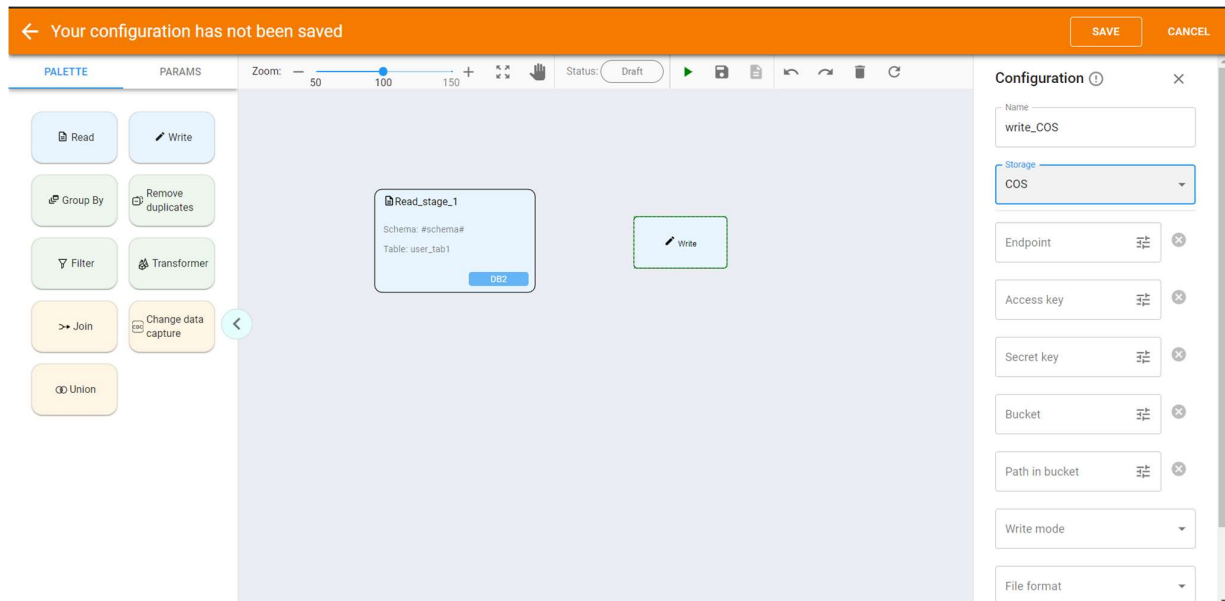
- 8) Save the stage by pushing Confirm button on the configuration panel. If you want to save your job at this step, you should press *Save* button on the header.

You have configured the first stage of the job and it now looks like this:



The Job Designer interface shows a job named 'Job1' in 'Draft' status. The left palette contains various stages: Read, Write, Group By, Remove duplicates, Filter, Transformer, Join, Change data capture, and Union. The main canvas displays a single stage named 'Read_stage_1' with the configuration: Schema: #schema#, Table: user_tab1, and Storage: DB2.

- 9) Now drag another stage, e.g. *Write* stage:



10) Enter a name a name for the stage and select *Storage* COS if you want to post data from the DB2 table to Cloud Object Storage file. Fill required parameters for COS *Storage*.

Available *Storage* values for write stage are:

- ✓ DB2
- ✓ COS
- ✓ Elastic Search
- ✓ STDOUT

Important:

Write mode field defines how data will be posted to its destination.

Available values are:

- ✓ Overwrite
- ✓ Append
- ✓ Error if Exists

File format is to choose a format of destination file.

Available formats are:

- ✓ CSV
- ✓ JSON
- ✓ Parquet
- ✓ ORC
- ✓ Text

11) Save the stage by pushing *Save* on the panel.

12) Now you have two stages to connect to each other.



Important:

To connect stages hover your mouse on a stage edge until you see green rectangle. Click it and drag it to the border of another stage and its green rectangle. When you reach it a green arrow should appear.



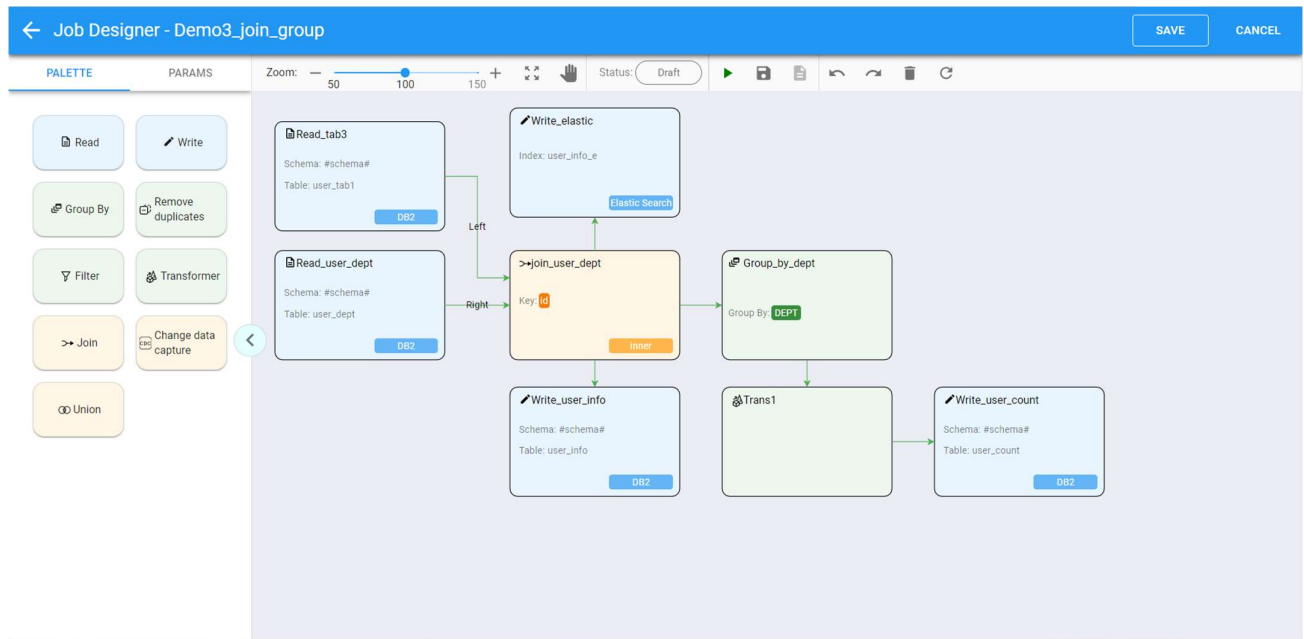
13) Save the job by pushing *Save* on the *Job Designer* header.

You have created a job reading data from the DB2 table and posting it to the COS file. For newly created job before you run it the status will be *Draft*:

Status: Draft

Drag other stages according to the flow of your job from source to destination.

See the job with more stages as the example:



4.3. Job Designer functions overview

The following functions are available in *Job Designer*:

- ✓ Zoom operations:
- ✓ Show job status:
- ✓ Run job / Stop job (for running)
- ✓ Save job
- ✓ See job logs
- ✓ Undo / Redo operation on canvas
- ✓ Remove element from canvas
- ✓ Refresh

4.4. Job Execution

Push *Play* button to run the job:


You will see its status changed from *Draft* to *Pending*.

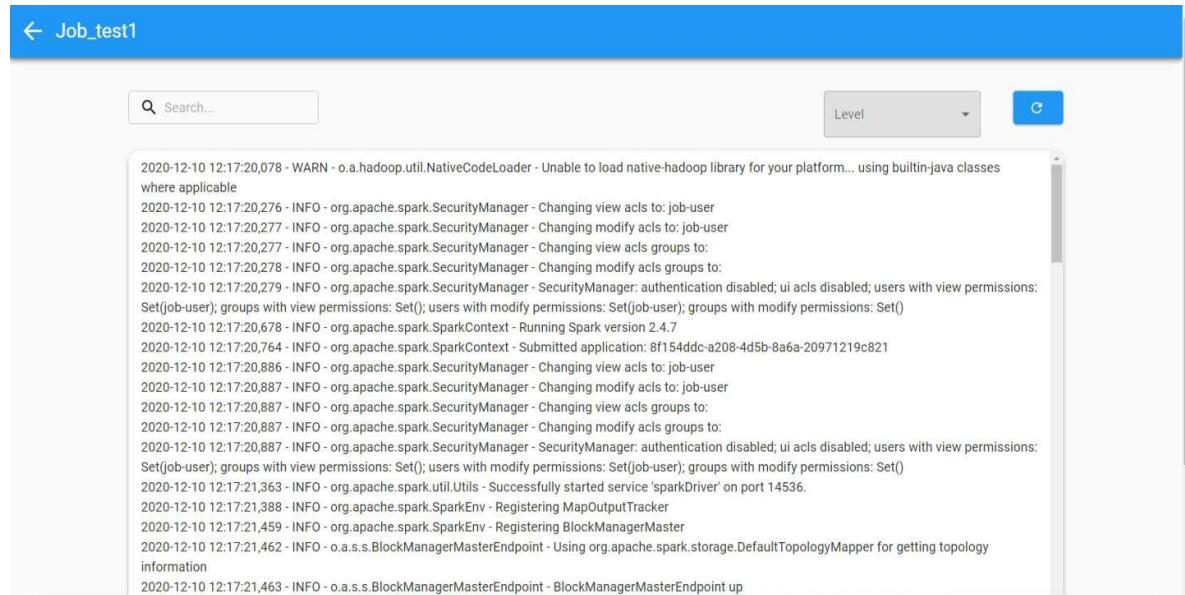
Push *Refresh* to update the status. It should turn to *Running*

Status:

While running it can be interrupted with *Stop* button.

When job completed the status will be *Succeeded* or *Failed*.

Use *Logs* button  to analyze job logs. You will get to *Logs Screen*:



5. Pipeline Operations

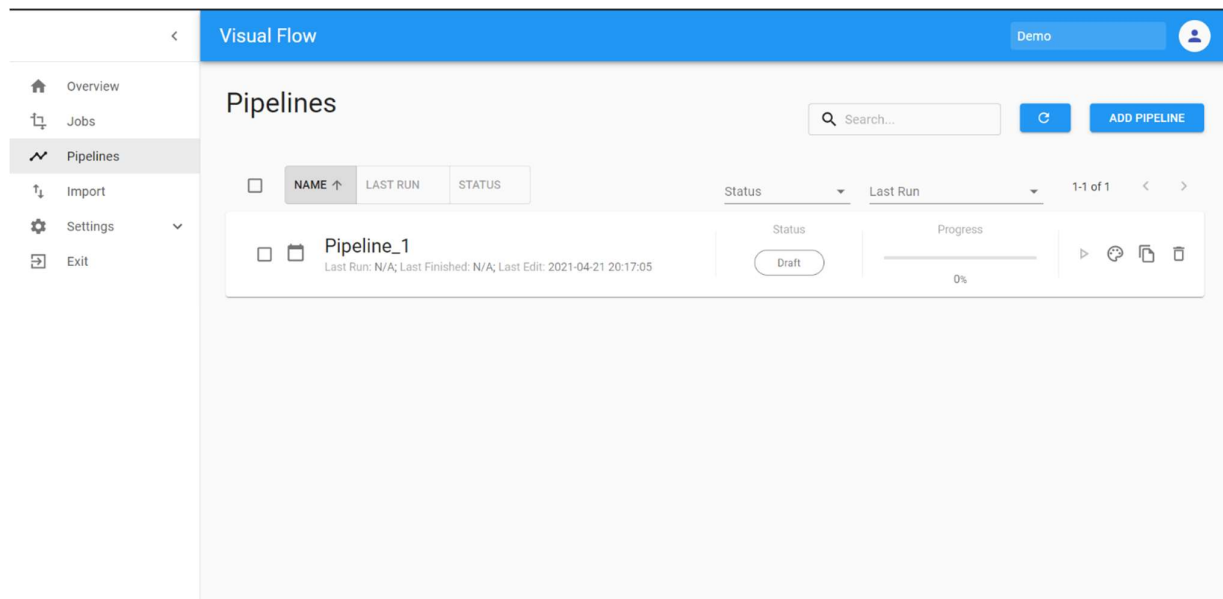
5.1. Pipelines Overview

Clicking *Pipelines* menu item will take you to *Pipelines Overview Screen*, which allows you to see list of pipelines existing within a project.

It displays the following information:

- Pipeline Name
- Checkbox for deleting the pipeline
- CRON schedule icon
- Pipeline Last run/Last finished/Last edit
- Pipeline Status (Draft/Running/Succeeded/Error)
- Pipeline Progress
- Available Actions (Run/Pipeline Designer/Copy/Delete)

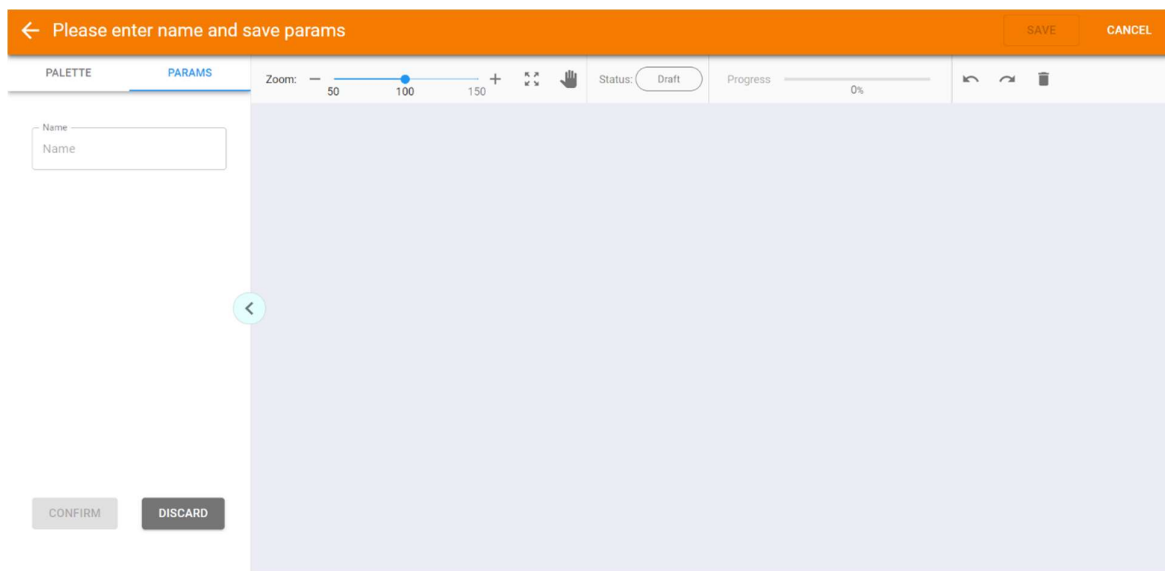
Note: the actions availability and therefore visibility is depending on user authorizations.



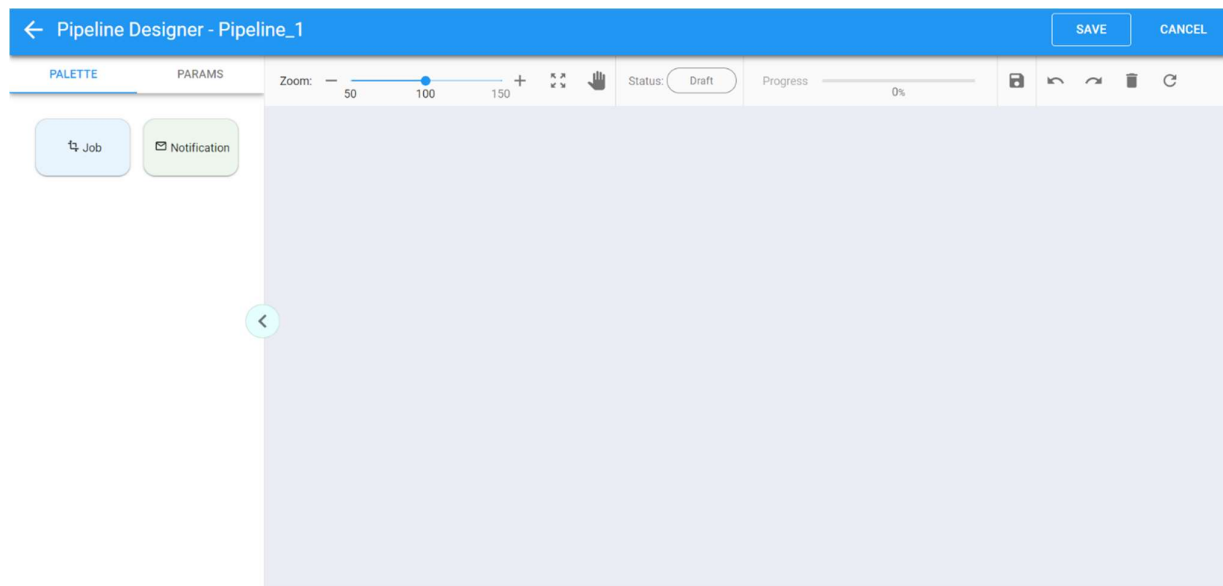
5.2. Create a Pipeline

With *Add Pipeline* button pushed you will get to *Pipeline Designer* for creating a pipeline.

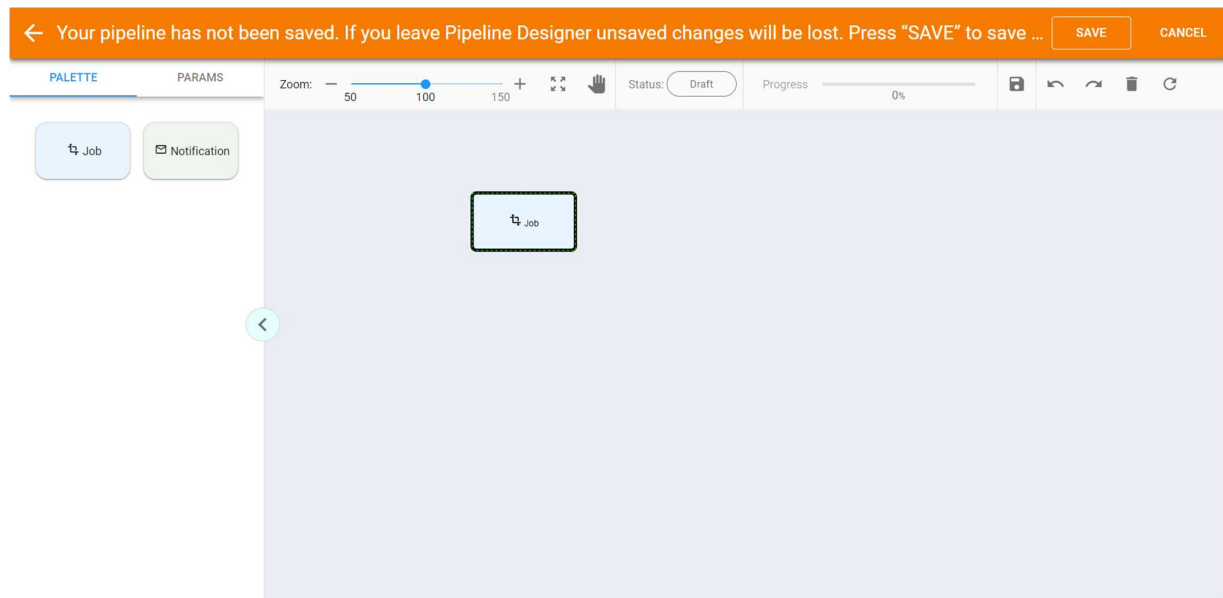
- 1) On the left configuration panel *Params* tab is opened by default, you can enter pipeline name and push *Confirm* button on the panel:



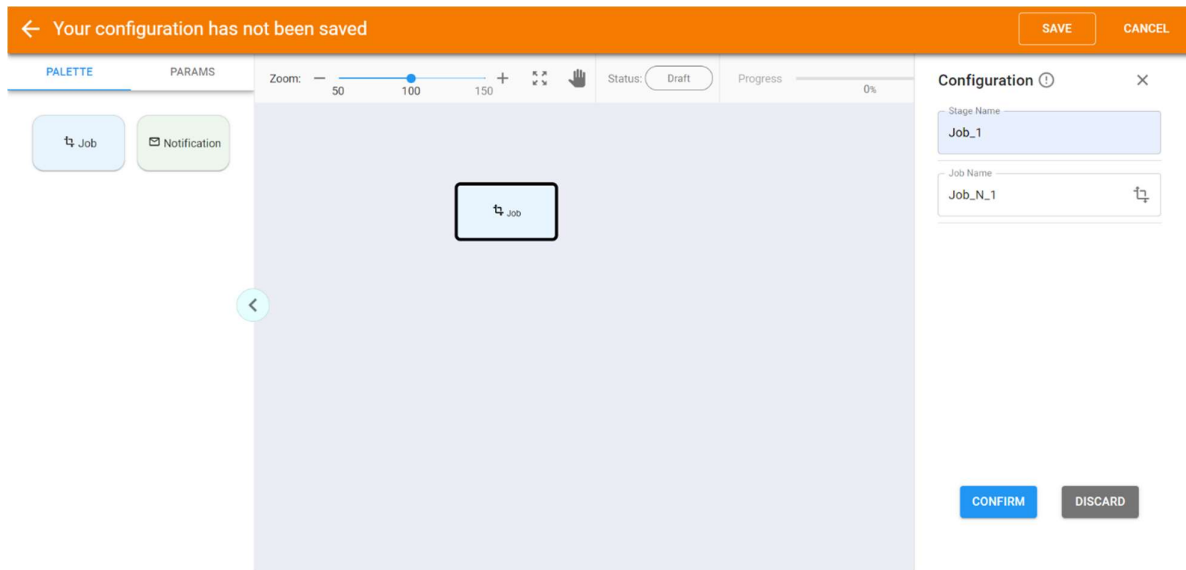
- 2) Save the job by pushing *Save* button on the *Pipeline Designer* header.
- 3) After saving the pipeline *Palette* tab is opened by default, at this tab you can see all available stages:




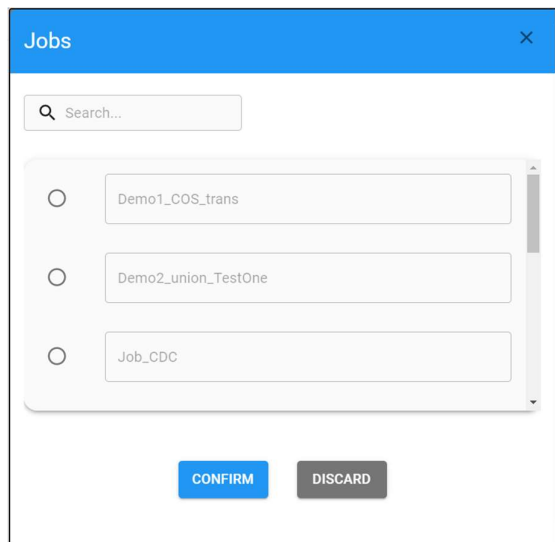
- 4) Pipeline is a combination of existing jobs stages and/ or notification stages. Notification stage most often added to configuration for the case of job stage failure/success. Start creating a pipeline by dragging *Job* stage to the canvas:



- 5) Double-click on the stage will open the configuration panel on the right:

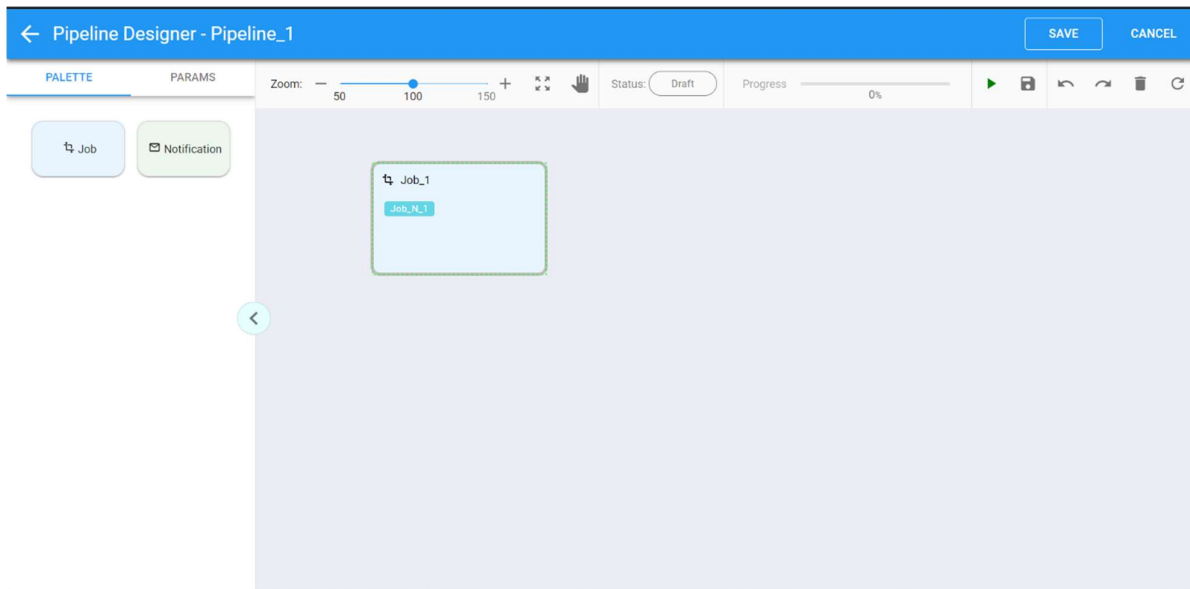


Enter a name for the stage and select job from the list by pushing *Job* button. 

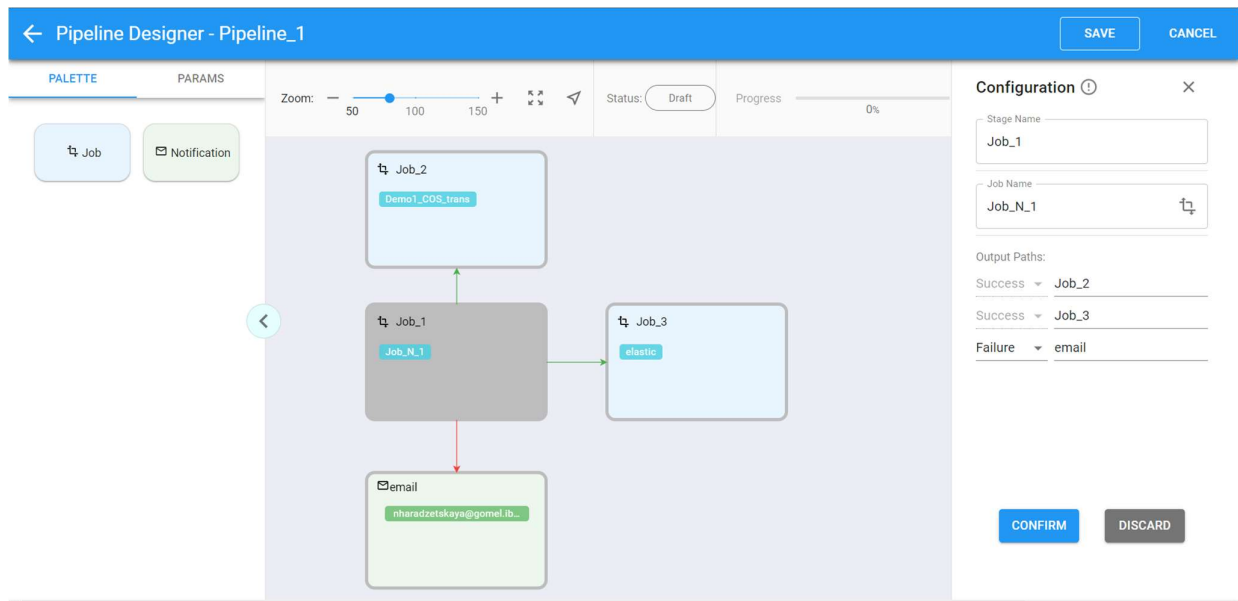


- 6) Save the stage by pushing *Confirm* button on the panel. If you want to save your pipeline at this step, you should press *Save* button on the header.

You have configured the first stage of the pipeline and it now looks like this:



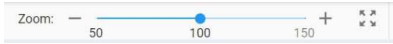










- 7) Drag and configure other stages. Connect them with the same manner you did in Job Designer. You can link your stages based on the success or failure of each stage. After connecting stages between themselves you can choose Success or Failure link on configuration panel. There can be only one connection for failure. See the example of fully configured pipeline:



Before the first run its status will be *Draft*. See each stage border painted in *Grey* color, which stands for *Draft*.

5.3. Pipeline Designer Functions Overview

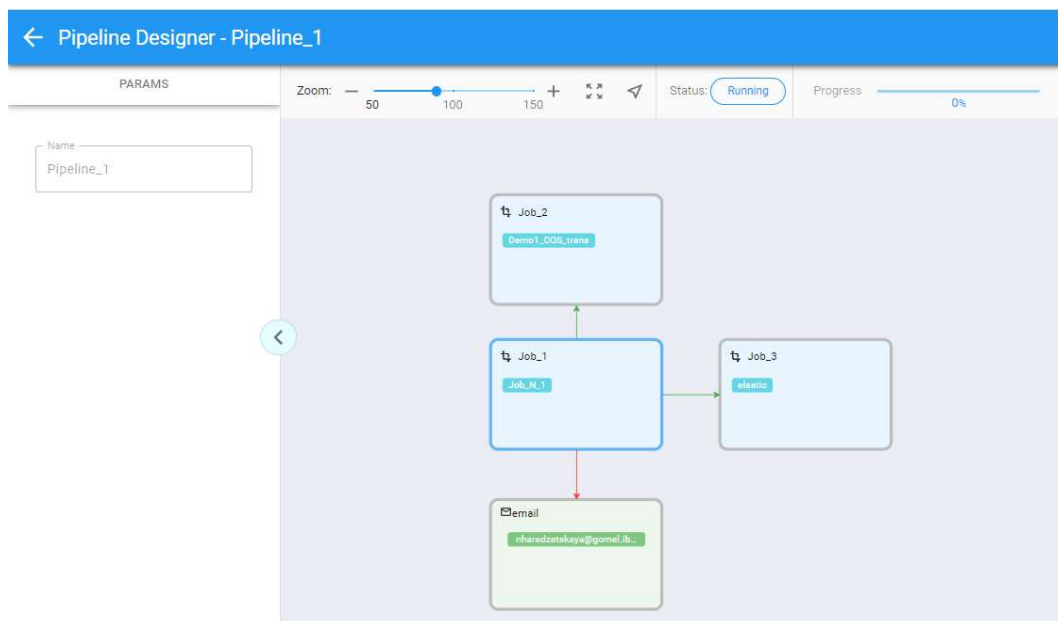
The following functions are available in *Pipeline Designer*:

- ✓ Zoom functions: 
- ✓ Move elements: 
- ✓ Move elements/screen: 
- ✓ Show pipeline status: 
- ✓ Show pipeline progress: 
- ✓ Run pipeline  / Stop pipeline  (for running)
- ✓ Save pipeline 
- ✓ Undo / Redo operation on canvas 
- ✓ Remove element from canvas 
- ✓ Refresh 

5.4. Pipeline Execution

If you run a pipeline e.g. from the above example its status will change from *Draft* to *Pending* and then to *Running*.

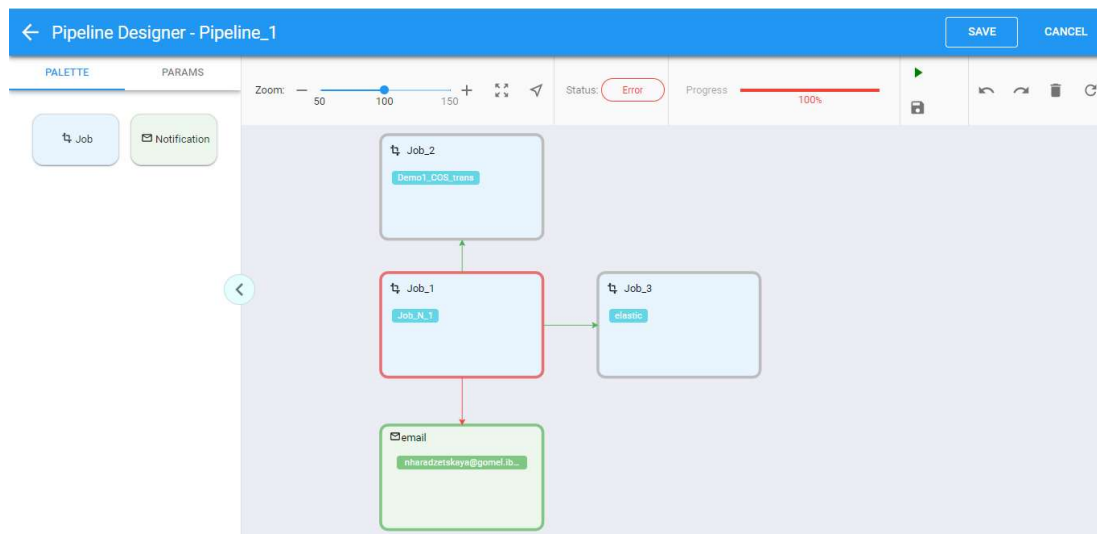
The border of the stage currently running will be painted in *Blue*:

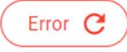


If a pipeline succeeded all completed stages will be painted in *Green* indicating success.

The ones configured for failure scenario (red arrow) of the previous stage will remain *Grey* as *Draft* as they have not been executed.

If a pipeline failed then *Red* border will indicate the failed stage:



Failed pipeline can be re-run from the point of failure with button  located on the Pipelines Overview Screen.

Important:

If *Job* stage succeeded or failed it has *Logs* button  available to get to *Logs Screen* for analyzing logs of a certain job.