

Visual Flow

User Guide

October, 2021

Version 0.6

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
09/07/2021	0.5	Pipeline Operators, Job Operations, Storages
10/24/2021	0.6	Jobs and Pipelines statuses, Custom container, Storages

Table of Contents

1	Introduction	4
1.1	...Terminology	4
1.2	...Scope and Purpose.....	5
1.3	...Process Overview	5
2	Roles and Authorizations	6
3	Project Operations	7
3.1	...Create Project.....	7
3.2	...Project Overview	8
3.3	...Manage Project Settings	9
4	Job Operations	11
4.1	...Jobs Overview	11
4.2	...Create a Job.....	12
4.3	...Job Designer Functions Overview	18
4.4	...Job Execution	18
5	Pipeline Operations.....	20
5.1	...Pipelines Overview	20
5.2	...Create a Pipeline	21
5.3	...Pipeline Designer Functions Overview.....	25
5.4	...Pipeline Execution	25

1. Introduction

1.1. Terminology

ETL is an 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 a 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
- Write
- Join
- Union
- Filter
- Group By
- Remove Duplicates
- Transformer
- Change Data Capture

Pipeline is a compound of multiple jobs and can be run. In Visual Flow, user can use such stages as:

- Job
- Notification
- Container

1.2. Scope and Purpose

Visual Flow web application is an 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:
 - ✓ AWS S3
 - ✓ DB2
 - ✓ Elastic Search
 - ✓ IBM COS
 - ✓ MSSQL
 - ✓ MySQL
 - ✓ Oracle
 - ✓ PostgreSQL
- 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 an 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.

Pipeline designer is a graphical design interface aimed for managing pipelines. Designing a pipeline is similar to designing a job.

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
- ✓ Run custom code
- ✓ 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 a Super-admin user can create a workspace (project) and grant access to this project.

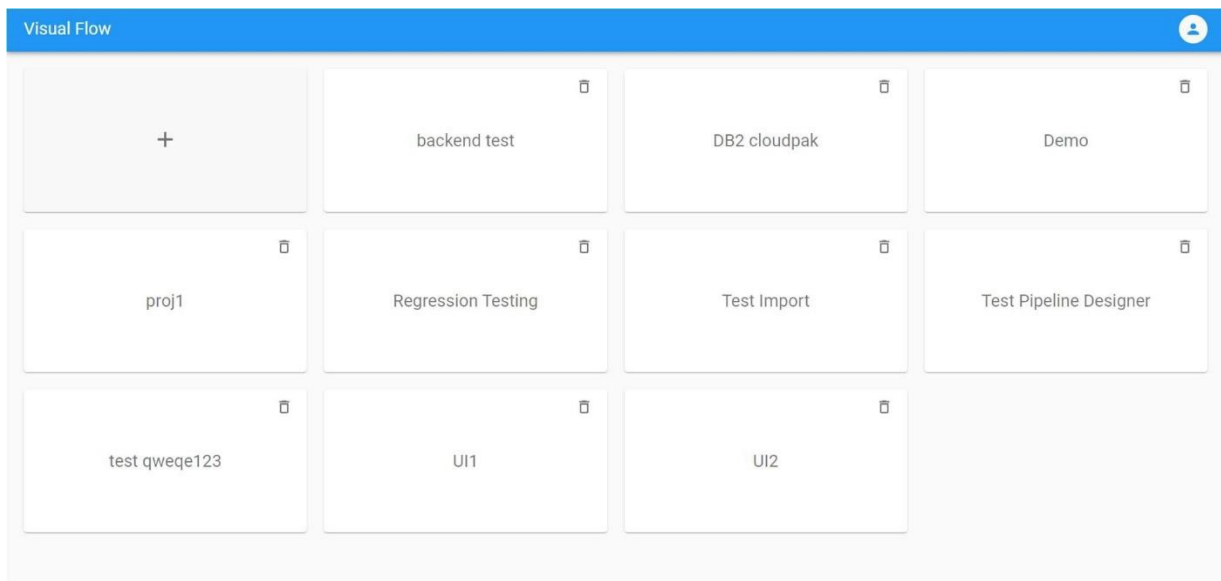
Role	Actions		
	Project Settings	Jobs	Pipelines
Viewer	View all	View all	View all
Operator	View all	View all / execute jobs	View all / execute pipelines
Editor	Edit all but Users and Roles	Edit / execute jobs	Edit / execute pipelines
Admin	Edit all	Edit / execute jobs	Edit / execute pipelines

3. Project operations

3.1. Create a Project

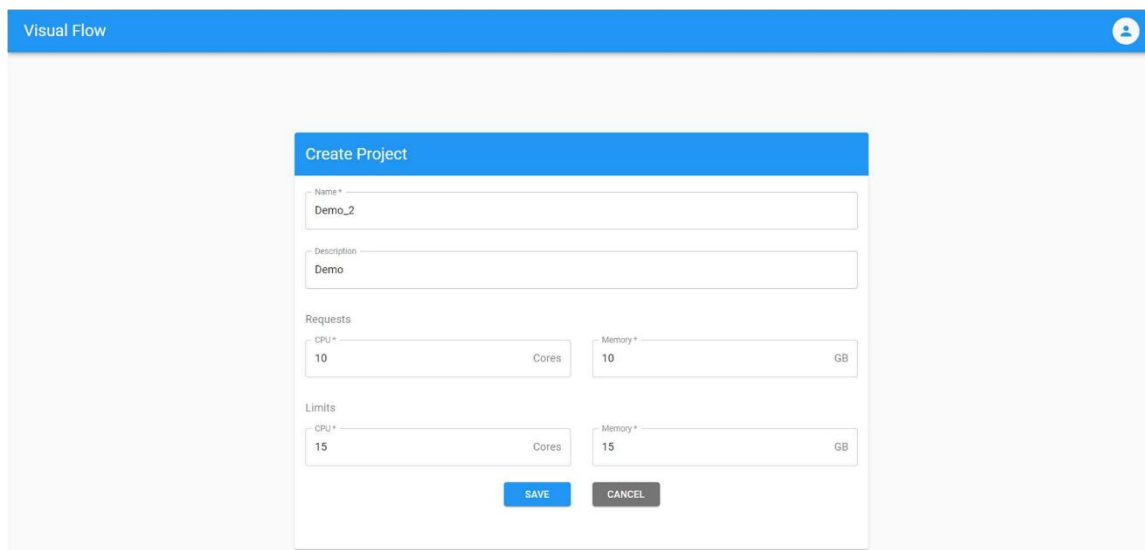
To create a project, you need to push “+” button on the initial screen.

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

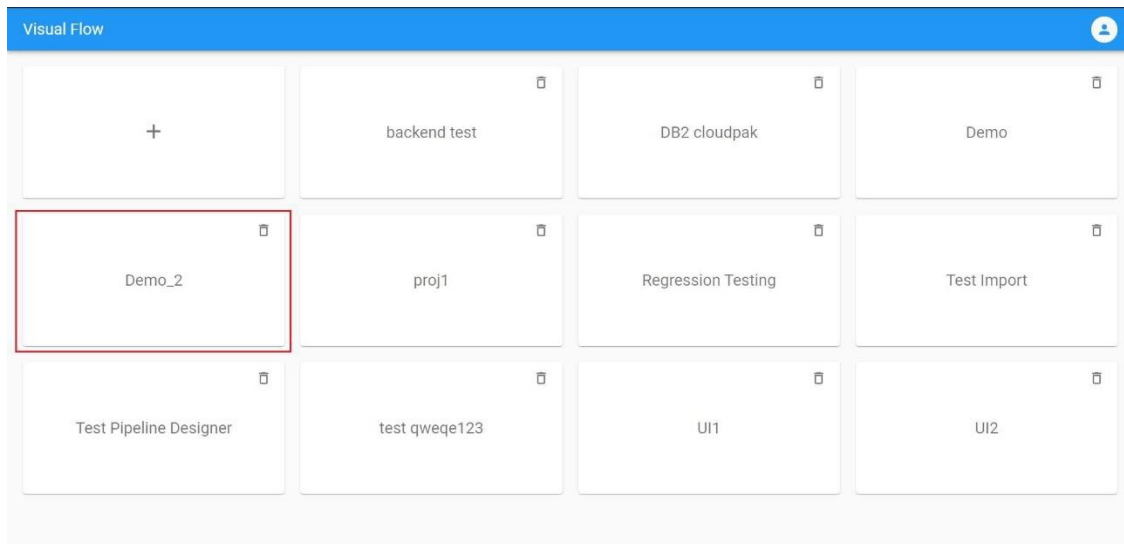


With “+” button pushed, you will get to *Create Project Form* to enter project basic settings:

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

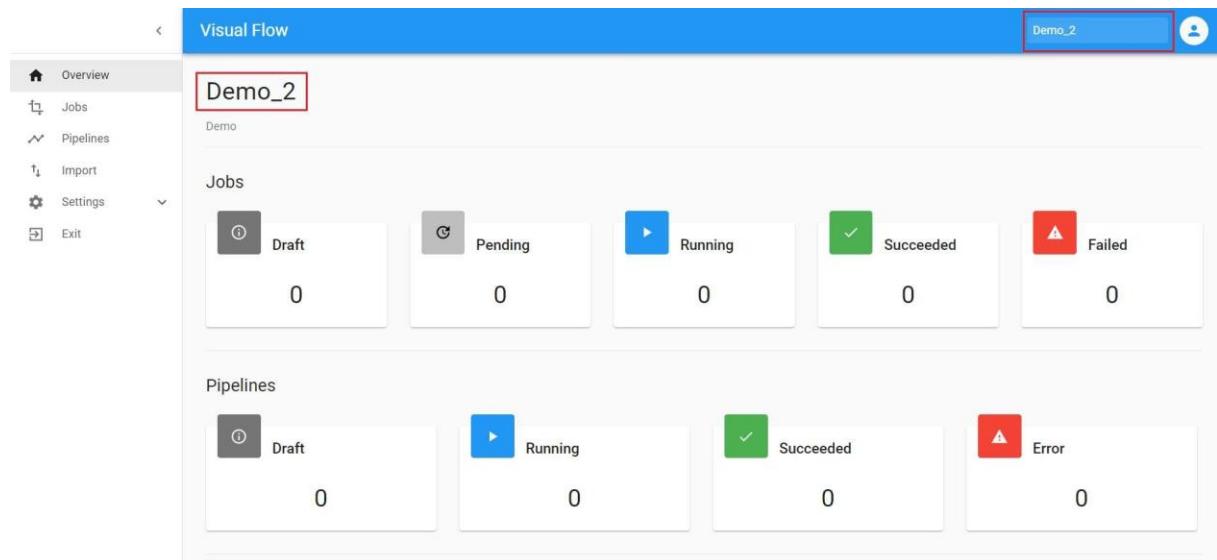
The screenshot shows the 'Create Project' form. It has a blue header bar with the text 'Create Project' and a user profile icon. The form contains several input fields: 'Name*' with the value 'Demo_2', 'Description' with the value 'Demo', 'Requests' section with 'CPU*' set to 10 (Cores) and 'Memory*' set to 10 (GB), and 'Limits' section with 'CPU*' set to 15 (Cores) and 'Memory*' set to 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 *ProjectOverview 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.

The screenshot shows the 'Visual Flow' interface with a sidebar on the left containing navigation options: Overview, Jobs, Pipelines, Import, Settings, Basic, Parameters, Users/Roles, and Exit. The 'Settings' menu is expanded, and the 'Basic' option is selected. The main area displays the 'View Project' dialog for a project named 'Demo_2'. The dialog includes fields for 'Name' (Demo_2) and 'Description' (Demo). It also has sections for 'Requests' and 'Limits', each with 'CPU' and 'Memory' settings. The 'Requests' section shows CPU at 10 Cores and Memory at 10 GB. The 'Limits' section shows CPU at 15 Cores and Memory at 15 GB. An 'Edit' button is visible in the top right corner of the dialog.

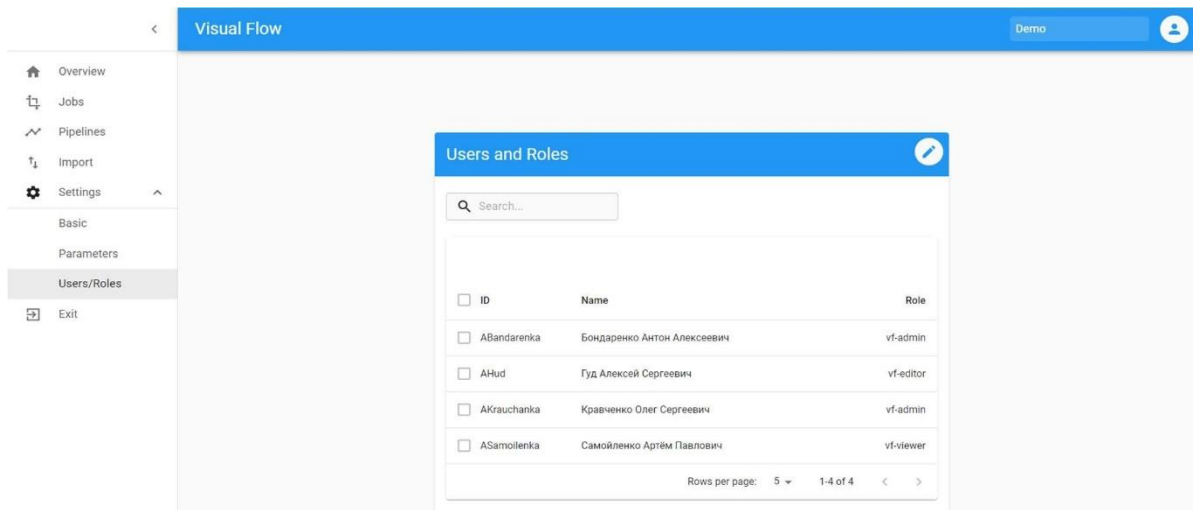
Parameters serve to store values required for the entire project, e.g. JDBC connection, DB2 credentials or table schemas can be the same for all jobs within the project and therefore stored at the project level. *Edit* button turns on the edit mode for updates.

The screenshot shows the 'Visual Flow' interface with the 'Parameters' option selected in the sidebar. The main area displays the 'View Project Parameters' dialog for a project named 'Demo'. The dialog includes a search bar and a list of parameters with their values. The parameters are: accessKey (1ae5ab46ec004860af18a9de3aa334c9), bucket (big-data-education), endpoint (s3.eu-de.cloud-object-storage.appdomain.cloud), index (vsw-test), jdbc (jdbc:db2://10.224.0.52:30100/EXAMPLE), nodes (23434e0d07a9405ca751a3a764027b69.us-east-1.aws.fo), and nodes1 (elastic.okd.comel.iba.bv). An 'Edit' button is visible in the top right corner of the dialog.

2) *User and Roles* allows user access management or view user access depending on authorization.

The user cannot change his role, this operation can be done by an Admin or a Super-admin. If the user tries to change his role, the error will occur «You cannot change your role».

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
- Resource Utilization (CPU/Memory)
- Available Actions (Run/Job Designer/Logs/Copy/Delete)

Job has a certain status at various phases of execution:

- Draft
- Pending
- Running
- Succeeded
- Failed
- Unknown (This status appears very rarely in the case of an undefined error)

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 also buttons for 'ADD JOB' and a search bar. The jobs listed are:

NAME	LAST RUN	STATUS	CPU	Memory
Demo1_COS_trans	Last Run: N/A; Last Finished: N/A; Last Edit: 2021-03-22 08:56:47	Draft	0%	0%
Demo2_union_TestOne	Last Run: N/A; Last Finished: N/A; Last Edit: 2021-02-26 11:24:55	Draft	0%	0%
Demo2_union_TestOne	Last Run: 2021-04-02 10:52:45; Last Finished: 2021-04-02 10:53:11; Last Edit: 2021-02-26 11:24:55	Failed	0%	0%
Job_CDC	Last Run: 2021-04-01 12:44:00; Last Finished: 2021-04-01 12:44:08; Last Edit: 2021-03-10 07:29:02	Failed	0%	0%
Job_CDC	Last Run: 2021-04-01 15:15:21; Last Finished: 2021-04-01 15:16:04; Last Edit: 2021-03-10 07:29:02	Succeeded	0%	0%

4.2. Create a Job

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

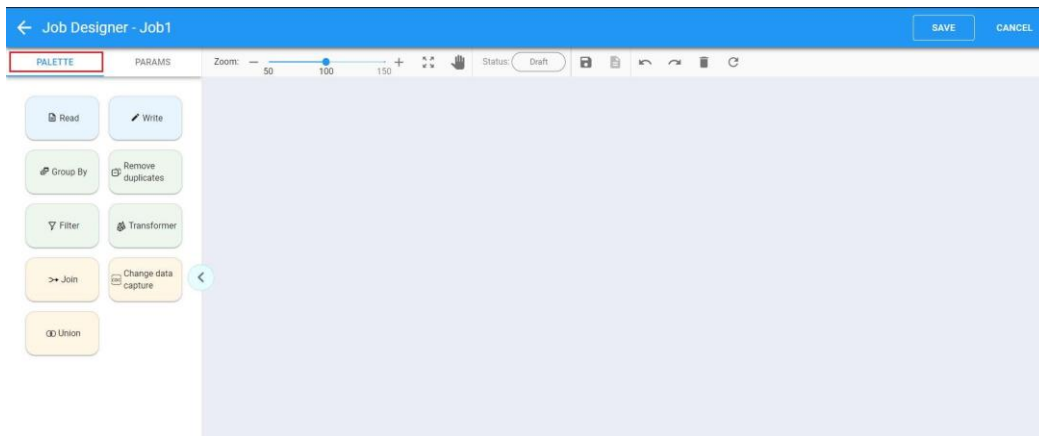
1) On the left configuration panel, you will need to give job a name, update parameters or keep their default values and then push *Confirm* on the panel:

This screenshot shows the 'Please enter name and save params' screen in the Job Designer. The left configuration panel contains the following fields: 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). The 'CONFIRM' button is highlighted in blue. The main area is a large light blue canvas with a zoom slider at the top (50, 100, 150) and a status bar at the bottom showing 'Draft'.

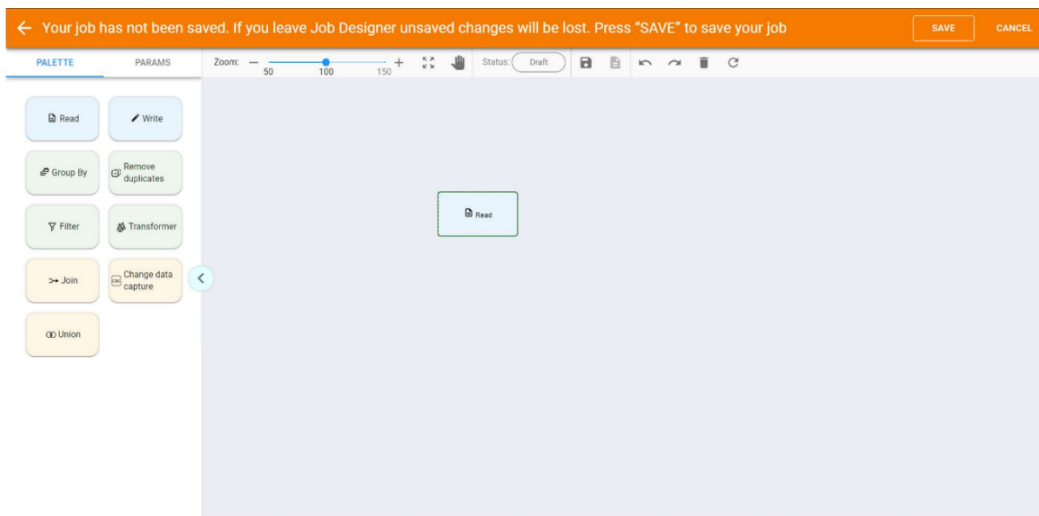
This screenshot shows the 'Your job has not been saved' screen in the Job Designer. The left configuration panel contains the same fields as the previous screenshot, but the 'CONFIRM' button is now greyed out. The main area is a large light blue canvas with a zoom slider at the top (50, 100, 150) and a status bar at the bottom showing 'Draft'. A warning message at the top states: 'Your job has not been saved. If you leave Job Designer unsaved changes will be lost. Press "SAVE" to save your job'.

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

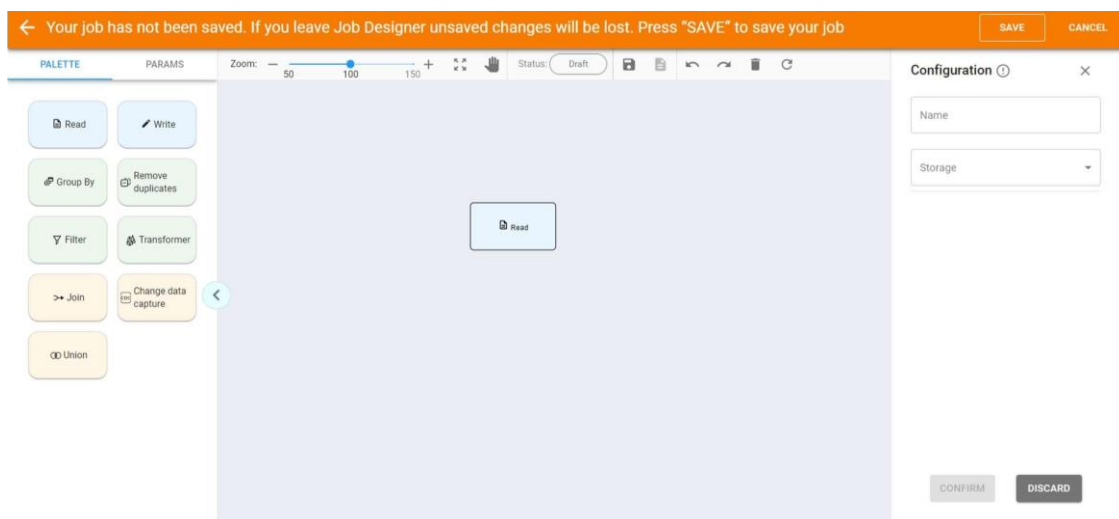
3) Go to *Palette* tab to see all available stages:



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:




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

The image shows a 'Configuration' dialog box with a title bar containing a warning icon and a close button. The dialog has several fields: 'Name' with the value 'Read_stage_DB2', 'Storage' with a dropdown menu showing 'DB2', 'JDBC URL', 'User', 'Password', 'Custom SQL' (dropdown), and 'CertData (optional)'. Each field has a 'Parameters' icon (three horizontal lines with a vertical line) and a 'Close' icon (an 'X' in a circle).

Available *Storage* values for Read stage are:

- ✓ AWS S3
- ✓ DB2
- ✓ Elastic Search
- ✓ IBM COS
- ✓ MSSQL
- ✓ MySQL
- ✓ Oracle
- ✓ PostgreSQL

6) 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 image shows a 'Configuration' dialog box similar to the one above, but with the 'Name' field set to 'Read_stage_1'. The 'Storage' dropdown is still 'DB2'. The 'JDBC URL' field is highlighted with a red rectangle, and its 'Parameters' icon is also highlighted with a red rectangle.

For the DB2 storage, you can use *Custom SQL* only Read stage (e.g. *select * from table where field = value*). Displays the schema and the table fields, if you choose false. If you choose true, you will be able to write your own SQL code in the provided field.

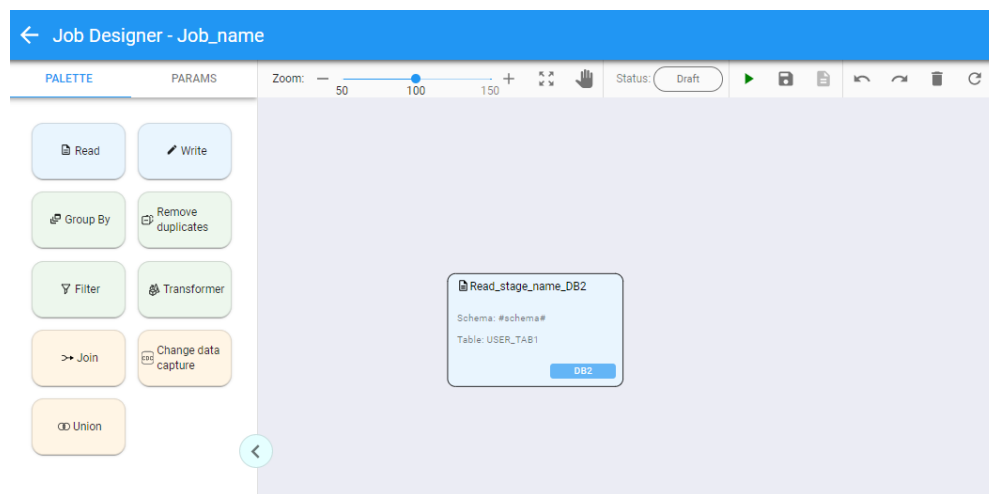
Custom SQL

True

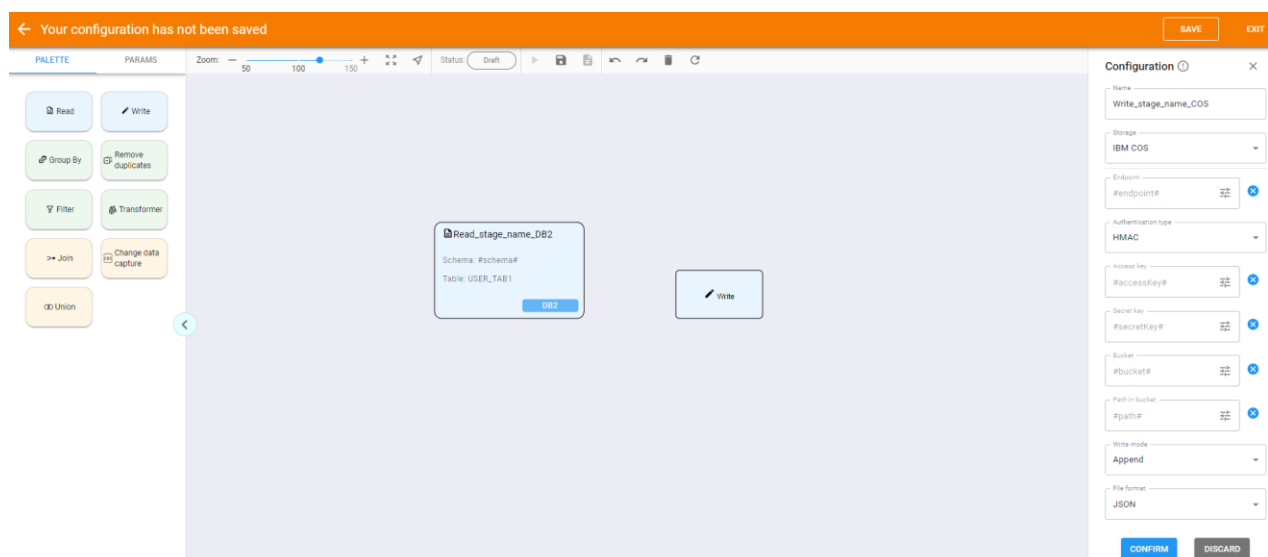
SQL statement

7) 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:



8) Now drag another stage, e.g. Write stage:



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

Available *Storage* values for Write stage are:

- ✓ AWS S3
- ✓ DB2
- ✓ Elastic Search
- ✓ IBM COS
- ✓ MSSQL
- ✓ MySQL
- ✓ Oracle
- ✓ PostgreSQL
- ✓ STDOUT

For IBM COS Storage, you can use *Authentication type*. Authentication type displays accessKey and secretKey, if you choose HMAC, or iamApiKey and iamServiceId, if you choose IAM.

<div>Authentication type HMAC</div>	<div>Authentication type IAM</div>
<div>Access key</div>	<div>IAM api key</div>
<div>Secret key</div>	<div>IAM service id</div>

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

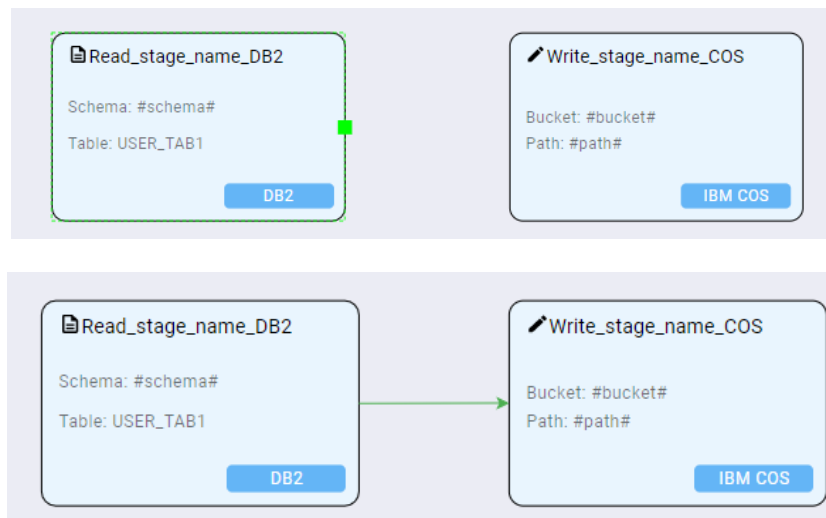
10) Save the stage by pushing *Save* on the panel.

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



Important:

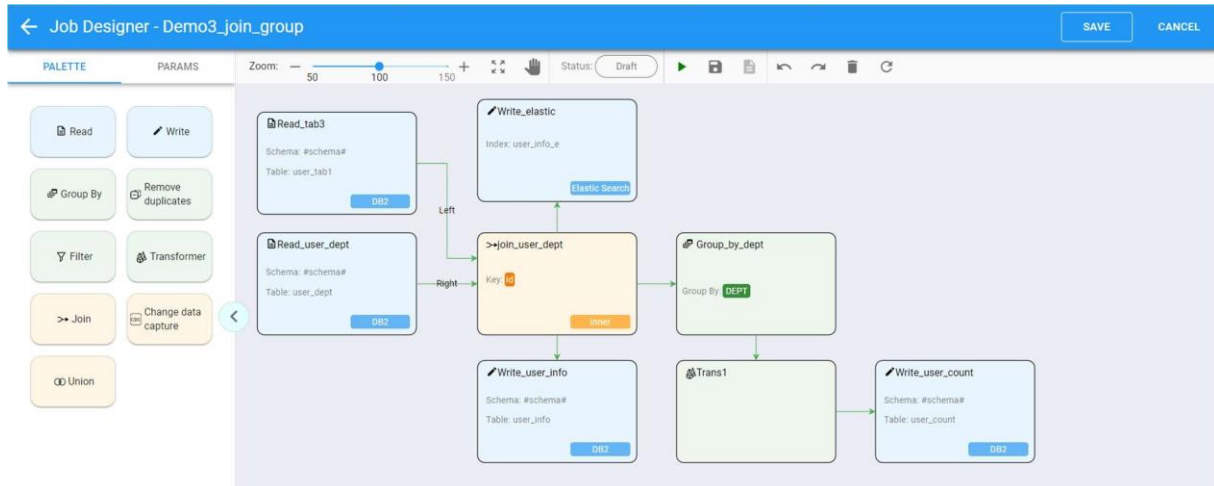
To connect stages, hover your mouse on a stage edge until you see a green rectangle. Click it and drag it to the border of another stage and its green rectangle. When you reach it, a green arrow should appear.



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 IBM 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

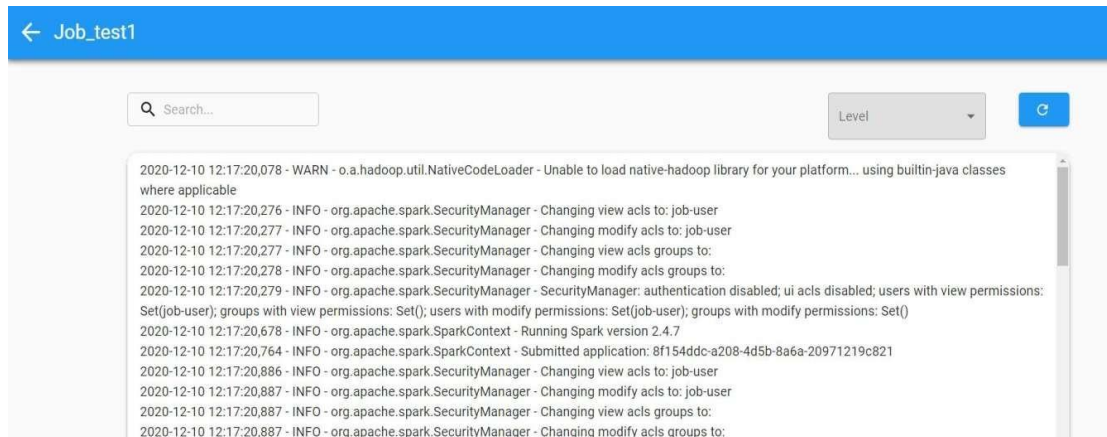
Push *Play* button to run the job:

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

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

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*:



Logs Screen has several levels:

- ✓ WARNING
- ✓ INFO
- ✓ ERROR
- ✓ DEBUG

5. Pipeline Operations

5.1. Pipelines Overview

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

It displays the following information:

- Pipeline Name
- Checkbox for deleting/exporting the pipeline
- Pipeline Last run/Last finished/Last edit
- Pipeline Status
- Pipeline Progress
- Available Actions (Run/Pipeline Designer/Copy/Delete)

Pipeline has a certain status at various phases of execution:

- Draft
- Running
- Succeeded
- Error (This status appears, e.g., due to incorrectly entered data)
- Terminated
- Suspended (This status can be reproduced via the API)
- Stopped
- Failed

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

Overview

Jobs

Pipelines

Import

Settings

Basic

Parameters

Users/Roles

Exit

Visual Flow

Test Pipeline Designer

Pipelines

Search...

ADD PIPELINE

NAME

LAST RUN

STATUS

Status

Last Run

1-5 of 26

Pipeline_1

Last Run: 2021-08-22 11:52:45; Last Finished: 2021-08-22 11:53:05; Last Edit: 2021-07-23 18:45:57

Status

Terminated

Progress

100%

Demo_test1

Last Run: 2021-08-02 06:00:48; Last Finished: 2021-08-02 06:01:09; Last Edit: 2021-07-26 17:51:55

Status

Terminated

Progress

100%

test_pipe1

Last Run: 2021-09-02 11:56:18; Last Finished: 2021-09-02 11:56:28; Last Edit: 2021-09-02 11:55:36

Status

Succeeded

Progress

100%

test_pipe2

Last Run: 2021-09-02 12:13:38; Last Finished: 2021-09-02 12:24:55; Last Edit: 2021-09-02 12:12:59

Status

Succeeded

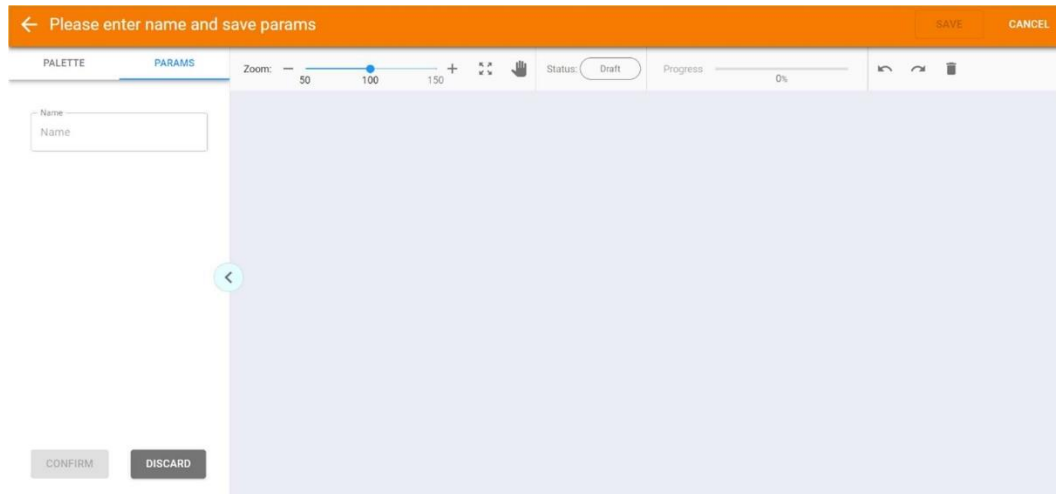
Progress

100%

5.2. Create a Pipeline

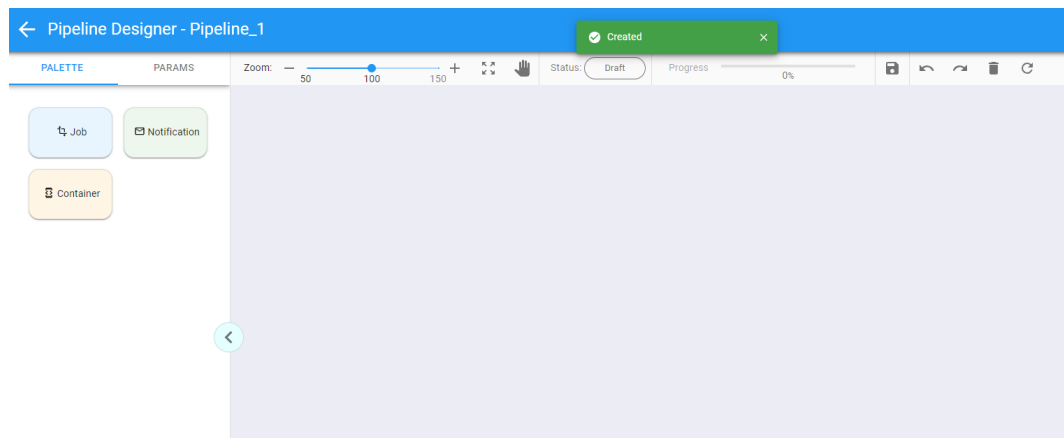
With *Add Pipeline* button pushed, you will get to *Pipeline Designer* for creating a pipeline.

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

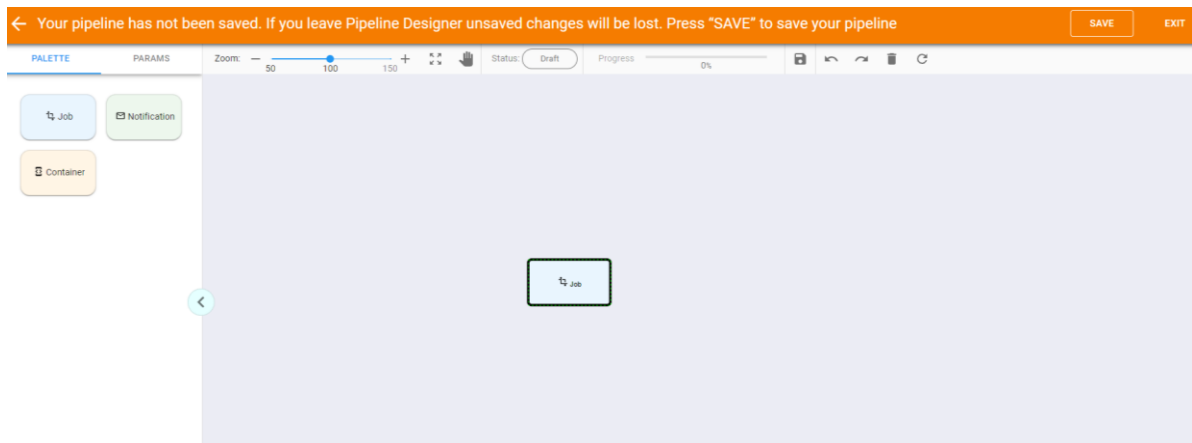


2) Save the pipeline by pushing *Save* button on the *Pipeline Designer* header.

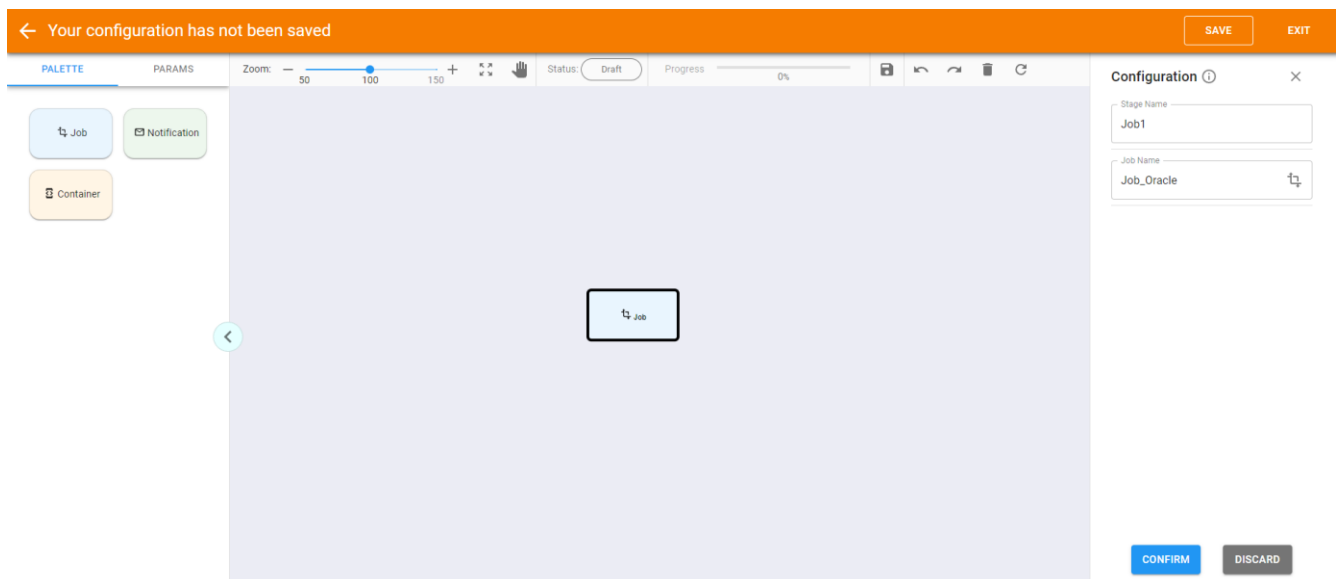
3) After saving the pipeline, *Palette* tab is opened by default, at this tab you can see all available stages:



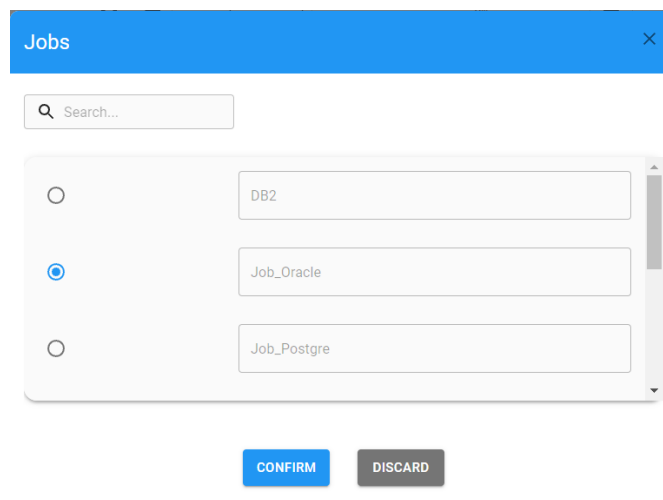
4) Pipeline is a combination of existing jobs stages and/or notification stages and container stages. Notification stage most often added to configuration in the case of job stage failure/success. Start creating a pipeline by dragging *Job* stage to the canvas:



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

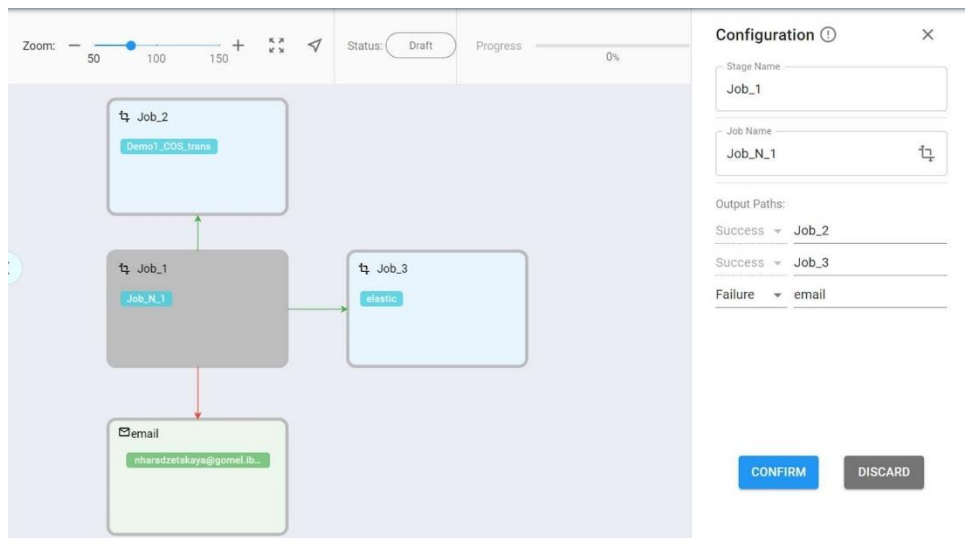


Enter a name for the stage and select a job from the list by pushing *Job* button.



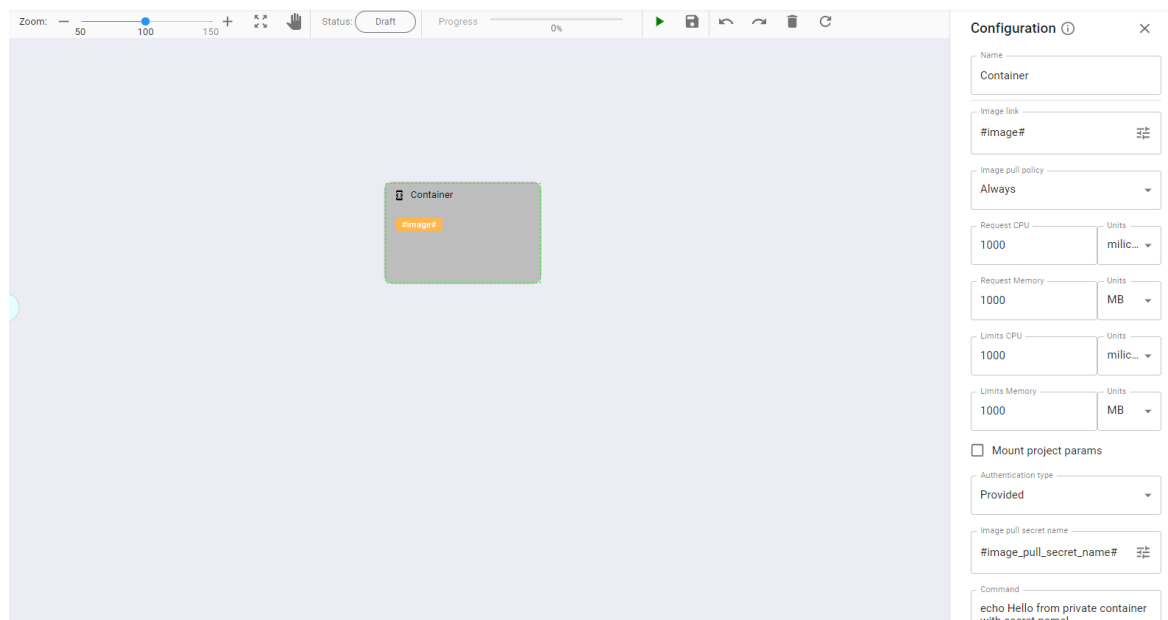
6) Save the stage by pushing *Confirm* button on the panel. If you want to save your pipeline at this step, you should press *Save* button on the header.

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 configured pipeline:



A custom container stage is required to run custom commands to execute any logic in the pipeline. Instead of custom commands, can use the created docker image.


1) Start creating a pipeline by dragging *Container* stage to the canvas and enter parameters in Configuration panel:

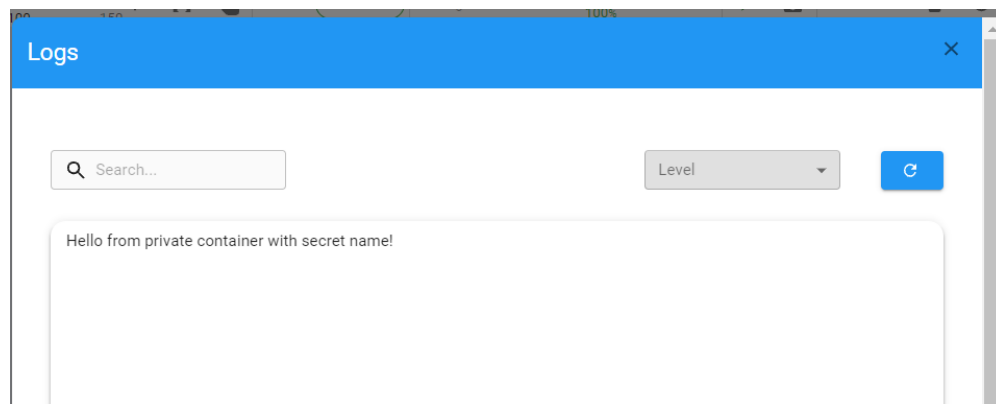



The Container stage has the following fields in the Configuration:

- ✓ Image link. Docker image path (Examples: mysql, mysql:latest, bitnami/argo-cd:2.1.2, localhost:5000/bitnami/argo-cd:2.1.2, registry.redhat.io/rhel7:latest.)
- ✓ Image pull policy. Defines when the image will be pulled(downloaded). Possible values:
 - *If not present* - download only if not exist locally;
 - *Always* - download before each start;
 - *Never* - do not download use only local copy.
- ✓ Requests and Limits CPU
- ✓ Requests and Limits memory
- ✓ Mount project params. Defines whether to mount all project params as environment variables inside the Pod.
- ✓ Authentication type
- ✓ Authentication mode that could be one of these:
 - *Not applicable* - image pull secrets are not needed, as the image is pulled from the public registry;
 - *New* - create a new image pull secret on the fly by providing all necessary information;
 - *Provided* - use existing image pull secret by providing it's name (Image pull secret name).
- ✓ Image pull secret name. Name of the secret to pull the image. Note that it must exist within the same k8s namespace as the current pipeline.
- ✓ Username
- ✓ Password
- ✓ Registry. Name of the registry for authentication.
- ✓ Command. Command that will be executed once Pod is be created.

Important:












Container stage has a  **Logs** button. In Logs window, provided that the pipeline is successfully completed, the text of the command that was previously registered in the Configuration of Container stage will be displayed.



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

5.3. Pipeline Designer Functions Overview

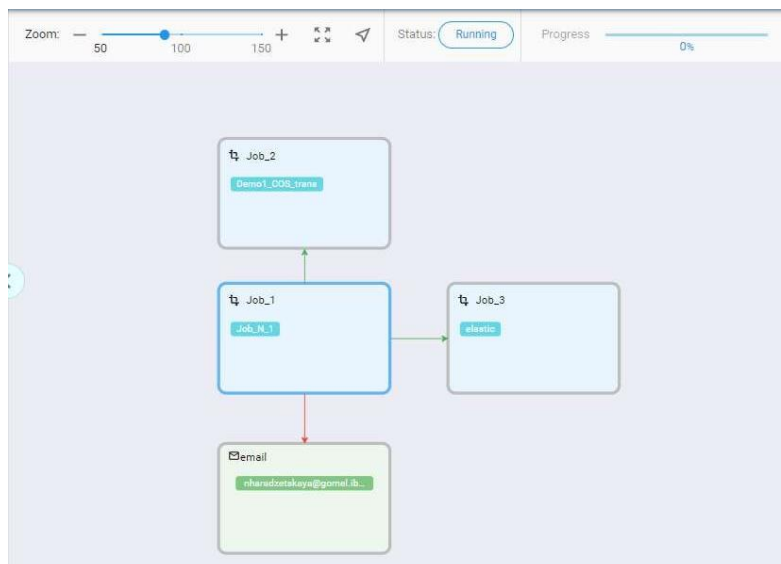
The following functions are available in *Pipeline Designer*:

- ✓ 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*. Push Refresh to update the status.

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

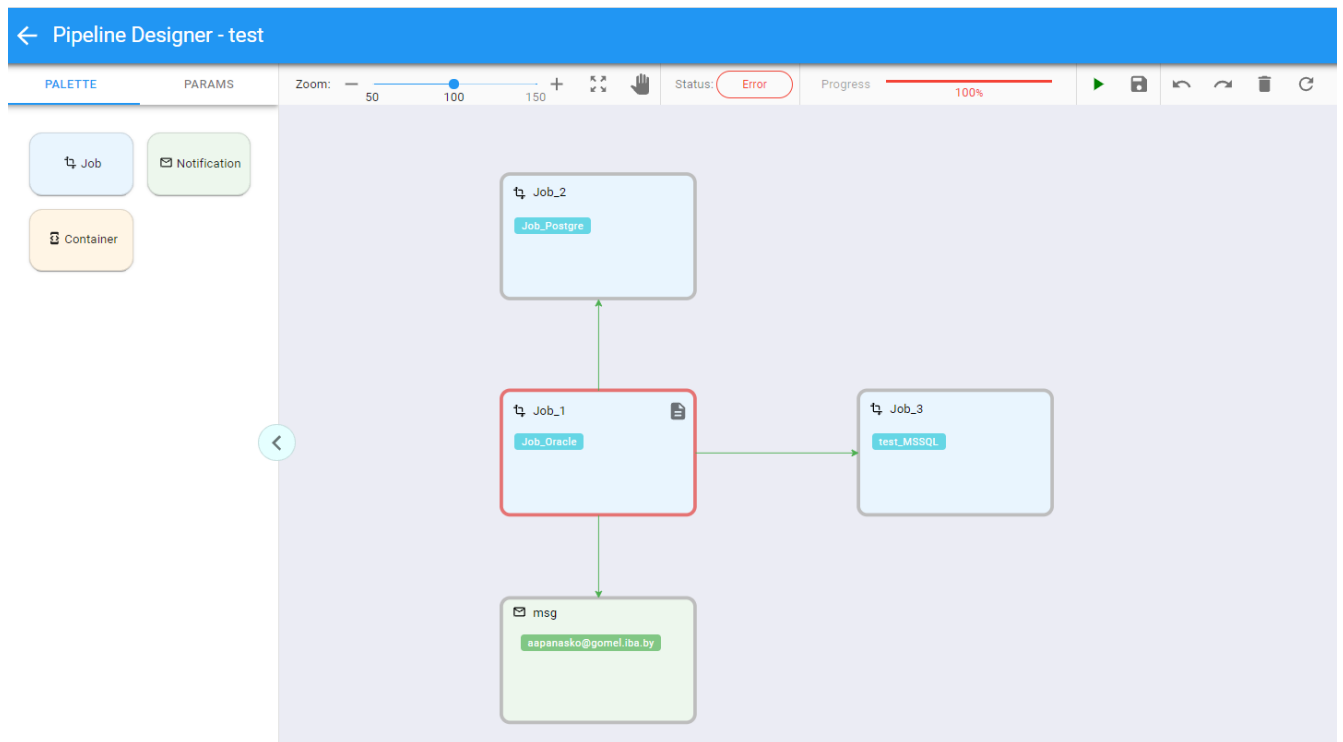



If a pipeline succeeded, all completed stages will be painted in *Green* indicating success.

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:

Job stage has a *Logs* button  for analyzing logs of a certain job.