

# ESA Datalabs – PIPEMAN

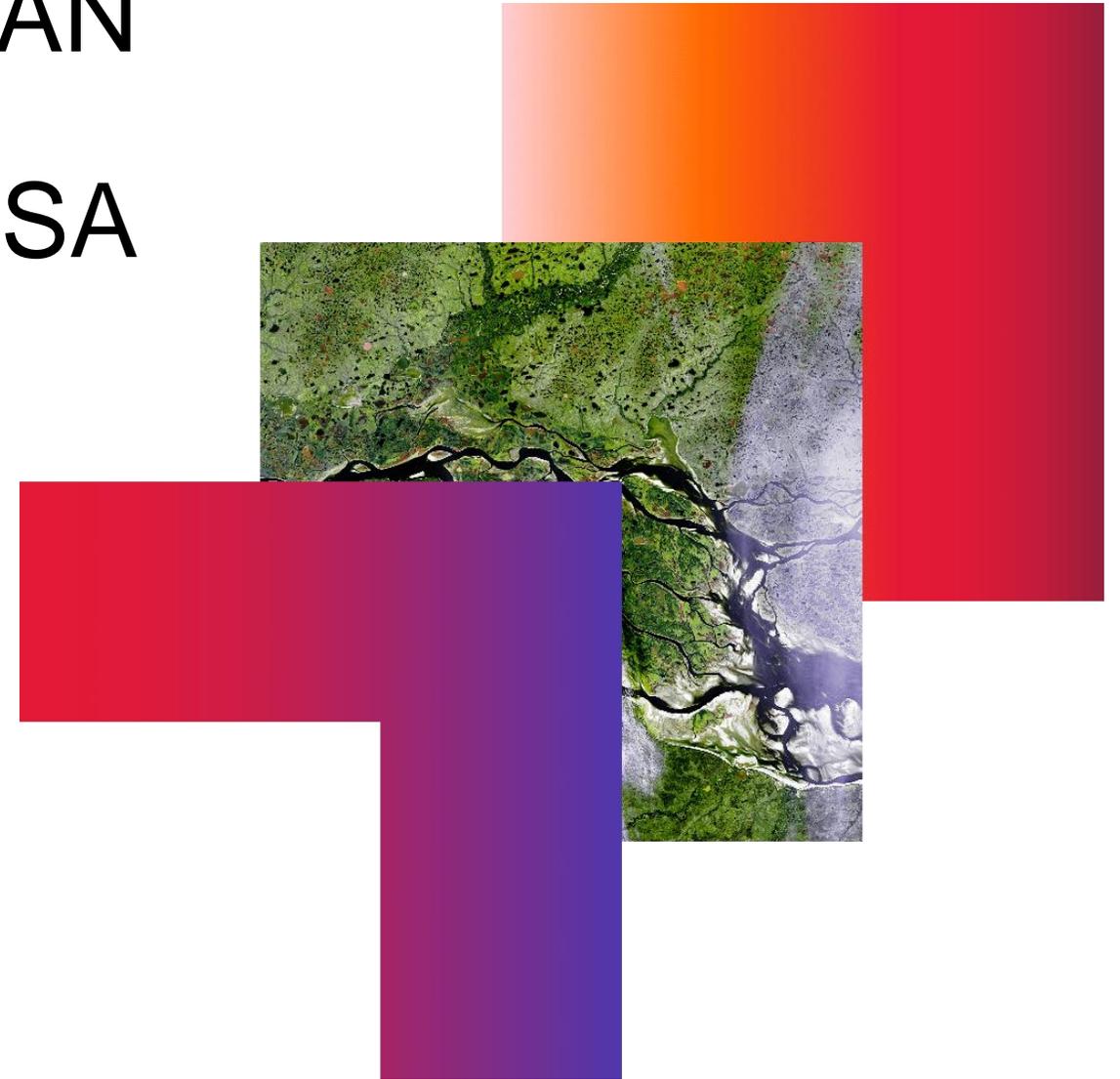
## Pipeline computing in ESA

### Datalabs

Brief overview – 24.11.2022

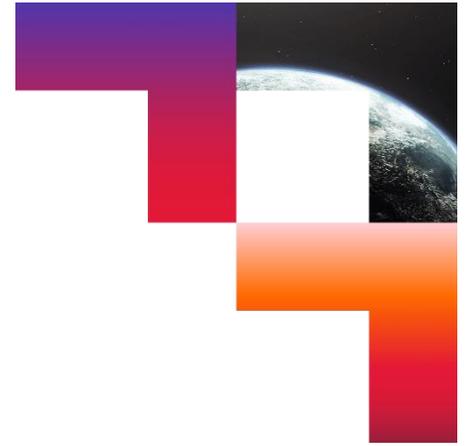
Alo Joosepson

CGI Estonia



# Agenda

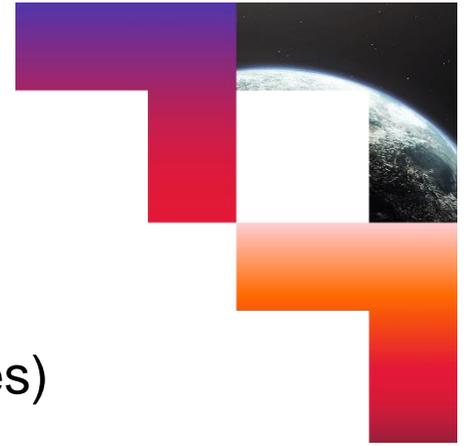
1. What is a pipeline?
2. User profiles
3. Logical architecture
4. Main functions
5. Release plans
6. Hands-on session invitation



# What is a pipeline?

## **A predefined data processing workflow**

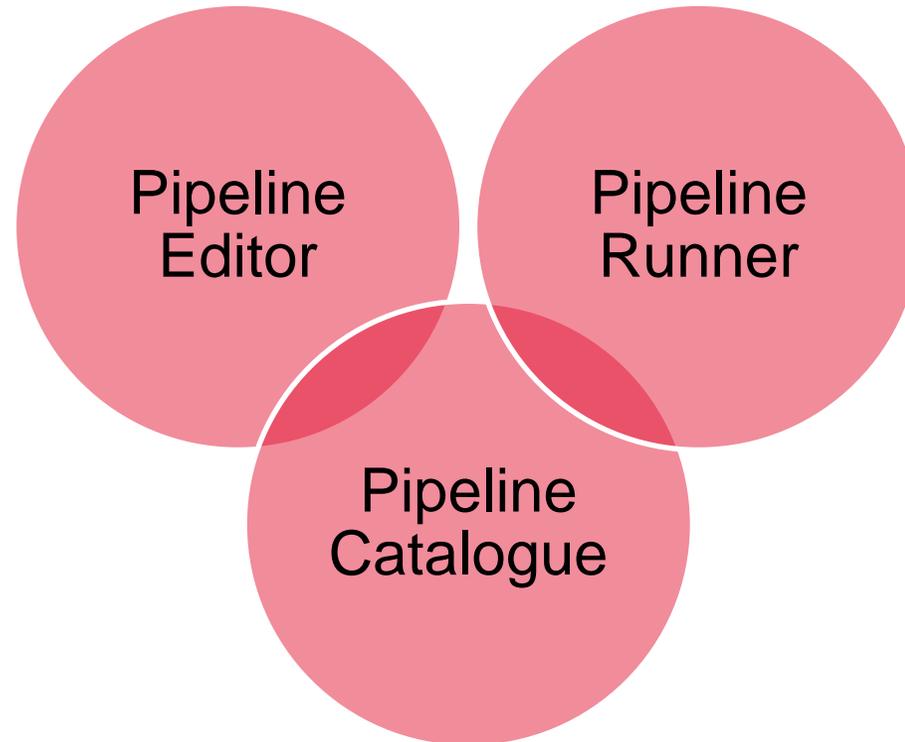
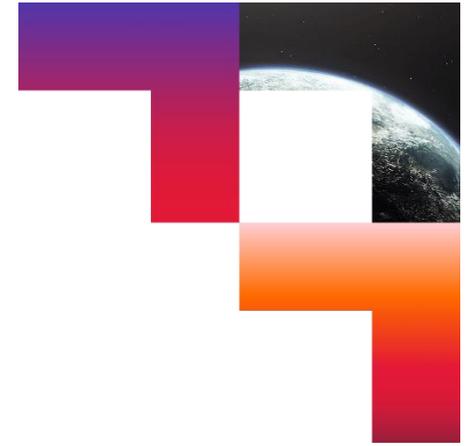
- Takes one or more inputs (files, parameters, messages, data from databases)
- Processes them in a series of one or more steps.
- Produces one or more outputs (files, messages, data into databases).
- Repeatable
- Re-usable
- Versionable
- Can do the heavy lifting of massive processing of typical data



# User profiles



# Logical architecture



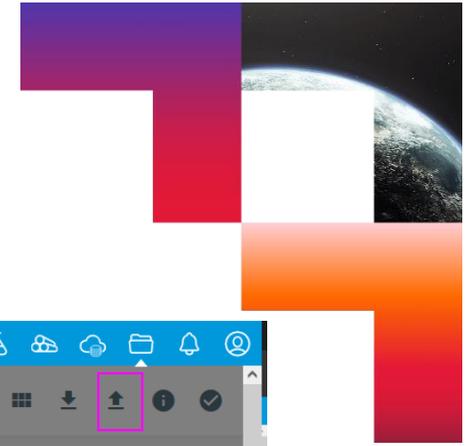
**Authentication and user rights**

**Data Discovery**

**Data Storage**

**Datalabs**

# Pipeline Runner – Create, find or upload input data

A screenshot of the ESA Datalabs interface. The main window shows a file browser with a list of files. A 'File Upload' dialog box is open, showing the selected file 'alo\_input\_file\_for\_HELLO\_PIPEMAN\_pipeline.txt'. Below the dialog, an 'Upload' modal is displayed with two options: 'File' and 'Folder'. The 'File' option is highlighted with a pink box. In the background, the 'Open' button in the 'File Upload' dialog is also highlighted with a pink box. The interface includes a search bar, navigation icons, and a list of files with their last modified dates.

Name	Date	Type
2022-11-Datalabs_workshop-program.PNG	18.11.2022 15:46	PNG File
alo_input_file_for_HELLO_PIPEMAN_pipeline.txt	19.11.2022 18:08	TXT File
ESA Datalabs.txt	18.11.2022 15:00	TXT File

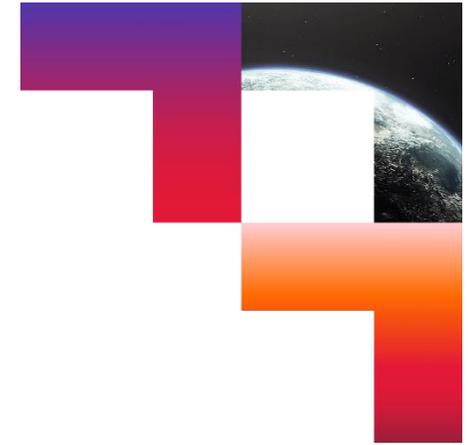
File name: alo\_input\_file\_for\_HELLO\_PIPEMAN\_pipeline.txt

Upload

Select an option to upload.

File Folder

# Pipeline Runner – Find a pipeline



esa | datalabs

## Pipelines

+ Launch new pipeline ? Help

Search pipelines Sort by newest first Select all

Currently you have no Pipeline Runs. Start a Pipeline Run by clicking "Launch new pipeline" button above.

Previous Next

## Pipeline launch

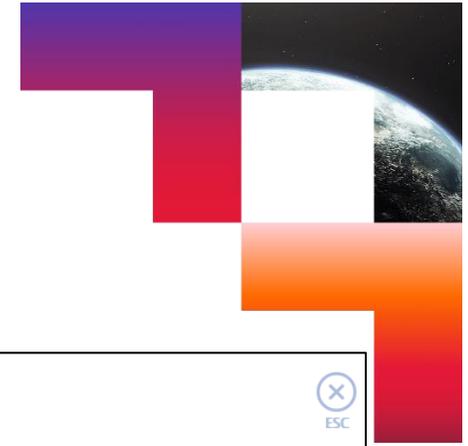
Find a pipeline in pipelines catalog

Filter results Sort by last modified Steps Pipelines Launch User pipeline Browse

### System Pipelines

 <b>Hello_PIPEMAN</b> <p>This pipeline has two steps. The input to the pipeline is a text file. The two steps append "Hello" and "PIPEMAN" in block letters to the copy of the input file. The output of the pipeline is a copy of the input file with the appended "Hello PIPEMAN". This example uses a "figlet" pipeline step that needs to exist as a step in Pipeline Catalogue. "figlet" is a unix command that writes text in block letters. The "figlet" pipeline step uses a container from registry (Docker) with the "figlet" unix command preinstalled.</p> <p>example figlet</p> <p>jcahi Fri Nov 11 2022</p>	 <b>BC_MCAM</b> <p>No description provided.</p> <p>BC</p> <p>jcahi Fri Feb 04 2022</p>	 <b>Timestamp</b> <p>This one-step pipeline has two inputs. The first input is a text file. The output of the pipeline is the copy of the input file with current time appended. The second input to the pipeline is an integer that specifies sleep time. The pipeline step sleeps the specified number of seconds.</p> <p>example</p> <p>jcahi Mon Jan 31 2022</p>
--	---	---

# Pipeline Runner – Launch parameters



## Pipeline launch

Hello\_PIPEMAN

This pipeline has two steps. The input to the pipeline is a text file. The two steps append "Hello" and "PIPEMAN" in block letters to the copy of the input file. The output of the pipeline is a copy of the input file with the appended "Hello PIPEMAN". This example uses a 'figlet' pipeline step that needs to exist as a step in Pipeline Catalogue. 'figlet' is a unix command that writes text in block letters. The 'figlet' pipeline step uses a container from registry (Nexus) with the 'figlet' unix command preinstalled.

Version 0.0.4

Now three steps

[Change version](#)

**Parameters** Graph Code

### Input(s)

[1] text (string)

[2] text\_1 (string)

[3] file (File) [Browse](#)

[4] text\_2 (string)

### Output

output (Directory) Path for all pipeline output (default: '/home/ajooseps/out')

out [Browse](#)

### Identifier

Hello\_PIPEMAN

### Schedule

[Add schedule](#)

### Notification

[Add notification](#)

### Keywords

example × figlet ×

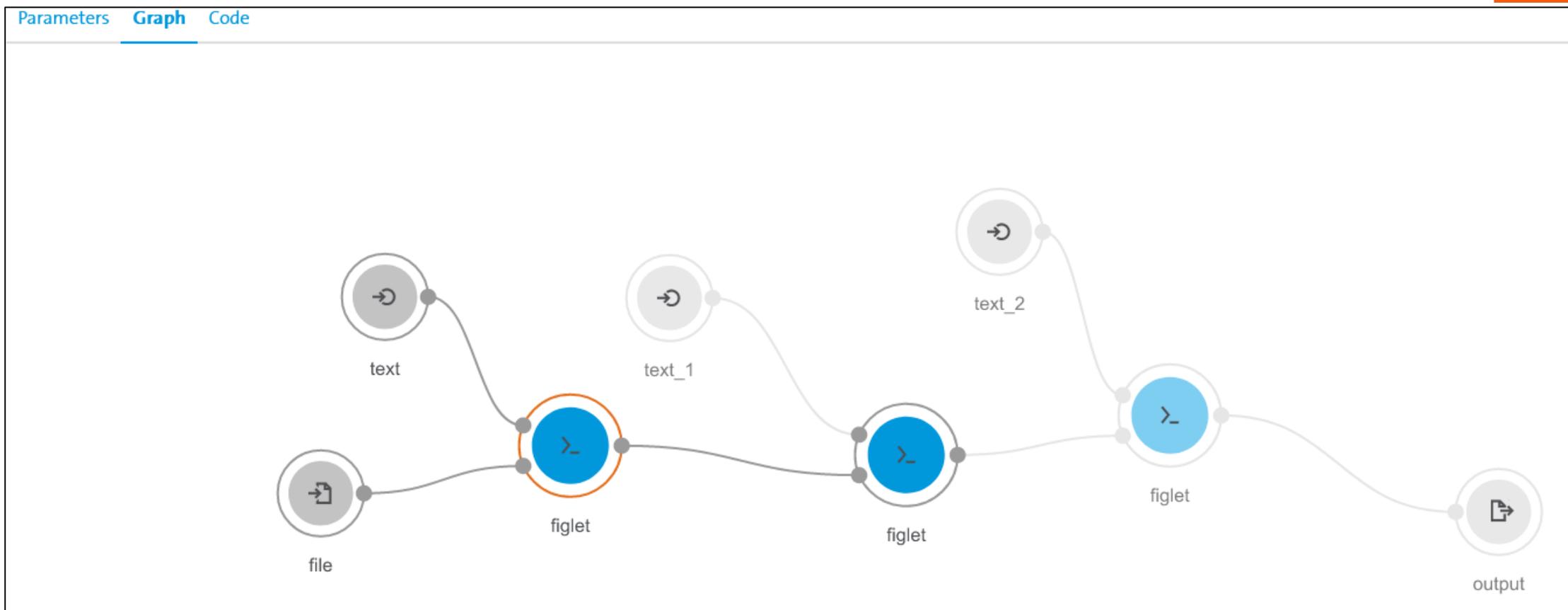
New keyword [Add keyword](#)

### Default image

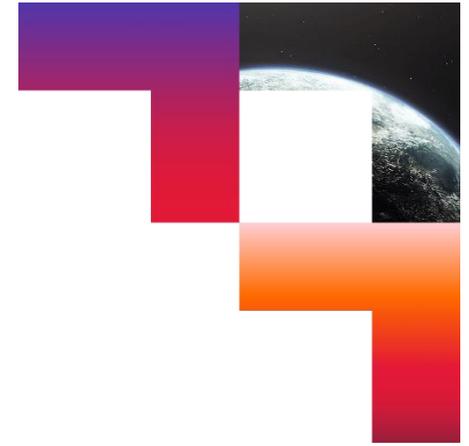
Default image tag [Browse](#)

[Launch pipeline](#)

# Pipeline Runner – Graph

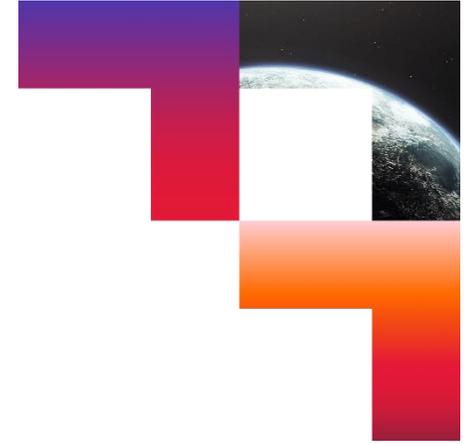


# Pipeline Runner – Pipeline Run Code



```
Parameters Graph Code
1 class: Workflow
2 cwlVersion: v1.0
3 id: _hello_p_i_p_e_m_a_n
4 label: Hello_PIPEMAN
5 $namespaces:
6   sepp: 'https://localhost/'
7   sbg: 'https://www.sevenbridges.com/'
8 inputs:
9   - id: text
10     type: string
11     'sbg:x': -1083.9599609375
12     'sbg:y': -397.1409912109375
13   - id: text_1
14     type: string
15     'sbg:x': -839
16     'sbg:y': -396
17   - id: data
18     type: File
19     label: file
20     'sbg:x': -1138
21     'sbg:y': -228.1409912109375
22   - id: text_2
23     type: string
24     'sbg:x': -556
25     'sbg:y': -453
26 outputs:
27   - id: output
```

# Pipeline Runner – Pipeline Run List



## Pipelines

[+ Launch new pipeline](#) [↶ Open pipeline editor](#) [? Help](#)

Search pipelines

☐ ▶ 🗑️

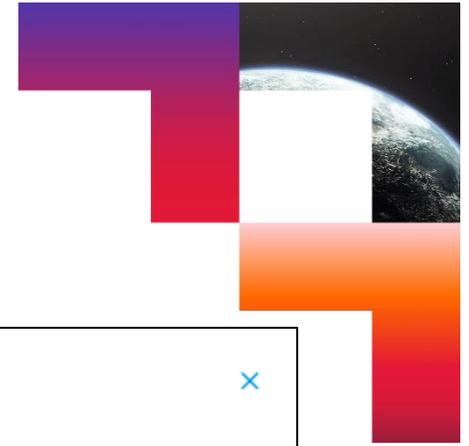
 **Hello\_PIPEMAN\_alo\_test\_4 [0.0.4]**

 This pipeline has two steps. The input to the pipeline is a text file. The two steps append "Hello" and "PEPEMAN" in block letters to the copy of the input file. The output of the pipeline is a copy of the input file with the appended "Hello PIPEMAN". This example uses a 'figlet' pipeline step that needs to exist as a step in Pipeline Catalogue. 'figlet' is a unix command that writes text in block letters. The 'figlet' pipeline step uses a container from registry (Nexus) with the 'figlet' unix command preinstalled.

[example](#) [figlet](#) [alo\\_test](#)

🕒 00:00:12 📅 4 minutes ago

# Pipeline Runner – Pipeline Run Execution Statistics



Pipeline run details - Hello\_PIPEMAN\_alo\_test\_4 [0.0.4] ×

[Execution](#) [Logs](#) [Parameters](#) [Graph](#) [Code](#)

---

**Status**  
Progress: Step 3 of 3

**Started**  
11/18/2022, 3:19:10 PM

**Finished**  
11/18/2022, 3:19:23 PM

**Schedule**  
No schedule set

**Notifications**  
Notifying **alo.joosepson@cgi.com** when **pipeline completes**

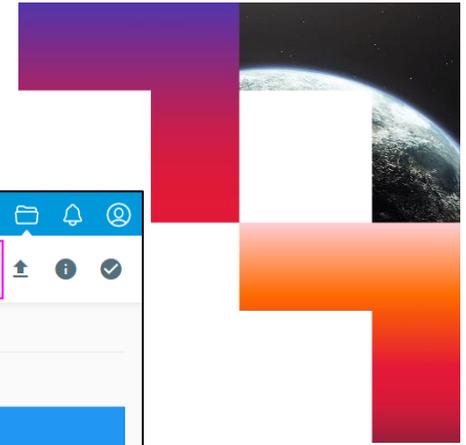
**Keywords**  
example figlet alo\_test

# Pipeline Runner – Pipeline Run Execution Logs



```
Pipeline run details - Hello_PIPEMAN_alo_test_4 [0.0.4]
Execution Logs Parameters Graph Code
Pipeline logs
1 [2022-11-18 15:19:10] INFO calrissian 0.10.0 (cwltool 3.0.20210124104916)
2 [2022-11-18 15:19:10] INFO Resolved '/cwl/tmp/cwl/Hello_PIPEMAN.pipeline.cwl' to 'file:///cwl/tmp/cwl/H
3 [2022-11-18 15:19:12] INFO [workflow ] starting step figlet
4 [2022-11-18 15:19:12] INFO [step figlet] start
5 [2022-11-18 15:19:12] INFO [workflow ] start
6 [figlet] logs start
7 [figlet] logs end
8 [2022-11-18 15:19:15] INFO [step figlet] completed success
9 [2022-11-18 15:19:15] INFO [workflow ] starting step figlet_1
10 [2022-11-18 15:19:15] INFO [step figlet_1] start
11 [figlet_1] logs start
12 [figlet_1] logs end
13 [2022-11-18 15:19:19] INFO [step figlet_1] completed success
14 [2022-11-18 15:19:19] INFO [workflow ] starting step figlet_2
15 [2022-11-18 15:19:19] INFO [step figlet_2] start
16 [figlet_2] logs start
```

# Pipeline Runner – Download results file(s)



cesalabs

Search...

My files

out > alo\_test\_2

Name ↑	Size	Last modified
final.txt	1.3 KB	a day ago
output.json	297 B	a day ago
pipeline_logs.log	1.19 KB	a day ago
usage.json	2.14 KB	a day ago

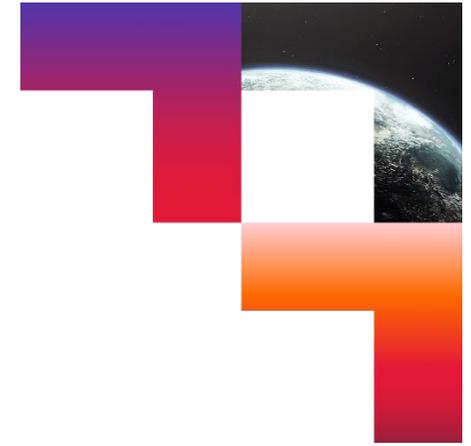
File Browser 2.11.0  
Help

### Download files

Choose the format you want to download.

- zip
- tar
- tar.gz
- tar.bz2
- tar.xz
- tar.lz4
- tar.sz

# Pipeline Runner – Re-run



 Hello\_PIPEMAN\_alo\_test\_3 [0.0.3] Re-run pipeline

 This pipeline has two steps. The input to the pipeline is a text file. The two steps append "Hello" and "PEPEMAN" in block letters to the copy of the input file. The output of the pipeline is a copy of the input file with the appended "Hello PIPEMAN". This example uses a 'figlet' pipeline step that needs to exist as a step in Pipeline Catalogue. 'figlet' is a unix command that writes text in block letters. The 'figlet' pipeline step uses a container from registry (Nexus) with the 'figlet' unix command preinstalled.

[example](#) [figlet](#) [alo\\_test](#)

00:00:16

### Pipeline launch

Parameters Graph Code

#### Hello\_PIPEMAN

This pipeline has two steps. The input to the pipeline is a text file. The two steps append "Hello" and "PEPEMAN" in block letters to the copy of the input file. The output of the pipeline is a copy of the input file with the appended "Hello PIPEMAN". This example uses a 'figlet' pipeline step that needs to exist as a step in Pipeline Catalogue. 'figlet' is a unix command that writes text in block letters. The 'figlet' pipeline step uses a container from registry (Nexus) with the 'figlet' unix command preinstalled.

Version 0.0.3  
No description provided.

[Change version](#)

**Input(s)**

[1] text (string)  
input\_text0\_

[2] text\_1 (string)  
input\_text1\_

[3] file (File)  
alo\_test\_input\_file.txt [Browse](#)

[4] text\_2 (string)  
input\_text2\_

[5] text\_3 (string)  
input\_text3\_

**Output**

output (Directory) Path for all pipeline output (default: /home/ajoseps/out)  
out/alo\_test\_4 [Browse](#)

**Identifier**  
Hello\_PIPEMAN\_alo\_test\_4

**Schedule**  
[Add schedule](#)

**Notification**  
Notify ajoosep@cgi.com when pipeline completes [Add notification](#)

**Keywords**  
[example](#) [figlet](#) [alo\\_test](#)  
New keyword [Add keyword](#)

**Default image**  
Default image tag [Browse](#)

[Launch pipeline](#)

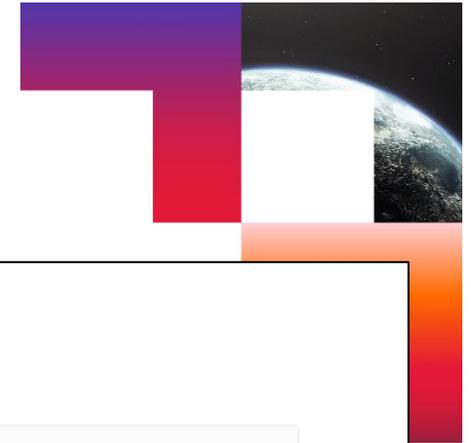
# Pipeline Runner – Stop and delete a pipeline run



A screenshot of a pipeline runner interface. The main panel shows a pipeline named 'Hello\_PIPEMAN\_alo\_test\_4 [0.0.3]' with a description and three tags: 'example', 'figlet', and 'alo\_test'. A red square stop button is circled in red, and a 'Stop pipeline run' dialog box is overlaid on top. The dialog box contains the text: 'datalabs.datpipesys.ebe.lan Are you sure you want to kill "Hello\_PIPEMAN\_alo\_test\_4"? Pipeline will stop immediately and cannot be resumed.' with 'OK' and 'Cancel' buttons.

A screenshot of a pipeline runner interface showing a pipeline run that has been stopped. The pipeline is 'Hello\_PIPEMAN\_alo\_test\_5 [0.0.3]'. A red square stop button is circled in red, and a 'Stopped' label is overlaid on it. The pipeline description and tags ('example', 'figlet', 'alo\_test') are visible. At the bottom, it shows a clock icon, '00:00:03', and a document icon, '41 seconds ago'.

A screenshot of a pipeline runner interface showing a pipeline run that is ready to be deleted. The pipeline is 'Hello\_PIPEMAN\_alo\_test\_5 [0.0.3]'. A green checkmark icon is circled in green, and a 'Delete pipeline run' dialog box is overlaid on top. The dialog box contains the text: 'datalabs.datpipesys.ebe.lan Are you sure you want to delete "Hello\_PIPEMAN\_alo\_test\_5"? This action cannot be reversed.' with 'OK' and 'Cancel' buttons. The pipeline description and tags ('example', 'figlet', 'alo\_test') are visible. At the bottom, it shows a clock icon, '00:00:03', and a document icon, '2 minutes ago'.



# Pipeline Runner – Scheduled runs and notifications

## Schedule

Schedule execution based on following **cron** expression

25-30 12 25 11 FRI 2022

Custom



Every minute between 12:25 and 12:30, on day 25 of the month, and on Friday, only in November, only in 2022 (UTC)

## Notification

Notify

when



Add notification

## Keywords

×

×

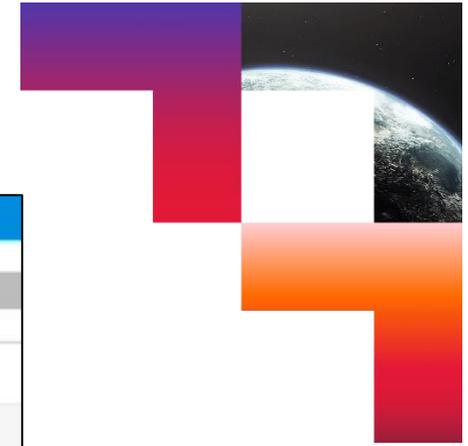
Select condition

pipeline fails

pipeline completes

pipeline exceeds time limit of

# Pipeline Runner – Run in a Jupyterlab notebook



The screenshot shows a JupyterLab notebook titled 'getting\_started.ipynb' in the 'Launcher' environment. The left sidebar displays a file browser with a table of files:

Name	Last Modified
/ notebooks /	
python_lib_help	9 months ago
cdr_demo.ipynb	9 months ago
getting_started.ipynb	9 months ago
hello_pipeman.ipynb	9 months ago
hwst_demo.ipynb	9 months ago
LICENSE.txt	9 months ago

The main notebook area contains the following content:

When we execute next cell, it creates a new pipeline run at system.

```
[11]: run = hello_pipeman.run(run_inputs)
run
```

PipelineRun(identifier="Hello\_PIPEMAN\_1111141759\_7366", id="3f62e78f-113e-4808-a449-803716920ff7", status="SUBMITTED")

We can use methods `get_logs()` and `get_status()` to query updates from system.

```
[12]: print(run.get_logs())
'get_status() ', run.get_status()
```

```
[12]: ('get_status() ', ('RUNNING', {}))
```

We find where pipeline output were generated using `run.output_path`

```
[13]: run.output_path
```

```
[13]: '/home/ajooseps/pipeline_output/Hello_PIPEMAN/1111141759_7366'
```

On pipeline completion file output will contain file `output.json` which has metadata about pipeline outputs.

**Note!** Here we use `await run.monitor()` to ensure that pipeline has completed execution.

`run.monitor()` is asynchronous python coroutine that runs while pipeline is also running.

```
[ ]: import json

if await run.monitor() != 'COMPLETED':
    raise Exception('Pipeline has failed.')

out_json_path = Path(run.output_path) / 'output.json'

with open(out_json_path, 'r') as f:
    out_json = json.loads(f.read())

out_json
```

Now lets read the output file.

# Pipeline Editor – Open, Personal Workspace



esa | datalabs

## Pipelines

+ Launch new pipeline

Open pipeline editor

? Help

esa | datalabs

Workspace Catalogue

### Workspace

- > .team\_areas
- > data
- > example\_pipeline\_input
- > my\_pipelines
- > out
- > pipeline\_output
- > pulled\_system\_pipelines
- alo\_test\_input\_file1.txt
- alo\_uploaded\_input\_file\_for\_HELLO\_PIPEMAN\_pipeline.txt
- getting\_started.ipynb
- notebooks
- test1.txt
- test2.txt
- Timestamp\_scheduled\_run1.txt

### My Pipelines in Workspace

- > alotestpipeline1
- > aloteststep1

### System Pipelines in Workspace

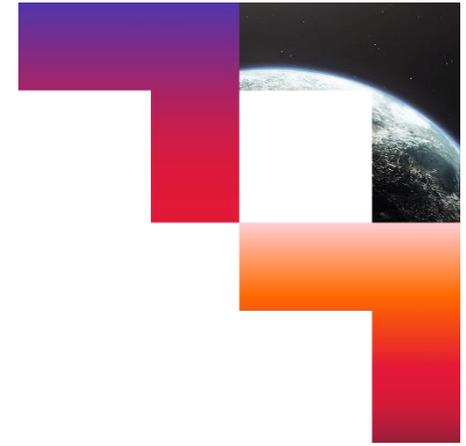
- > hello\_pipeman

Pull

## Pipeline Editor

Help

# Pipeline Catalogue – Find, Pull, Modify



**Pull from the Catalogue** ✕

System pipelines ○ Steps ○ Pipelines

Filter results Sort by last modified ▾



### Hello\_PIPEMAN

This pipeline has two steps. The input to the pipeline is a text file. The two steps append "Hello" and "PEPEMAN" in block letters to the copy of the input file. The output of the pipeline is a copy of the input file with the appended "Hello PIPEMAN". This example uses a 'figlet' pipeline step that needs to exist as a step in Pipeline Catalogue. 'figlet' is a unix command that writes text in block letters. The 'figlet' pipeline step uses a container from registry (Nexus) with the 'figlet' unix command preinstalled.

[example](#) [figlet](#)

 jkuhi  Fri Nov 11 2022



### BC\_MCAM

[BC](#)

 jkuhi  Fri Feb 04 2022

**Pipeline version**  
**Version 0.0.3:** Now four steps

**Destination path**  
  [Browse](#)

Overwrite if exists [Pull all filtered \(3\)](#) [Pull](#)

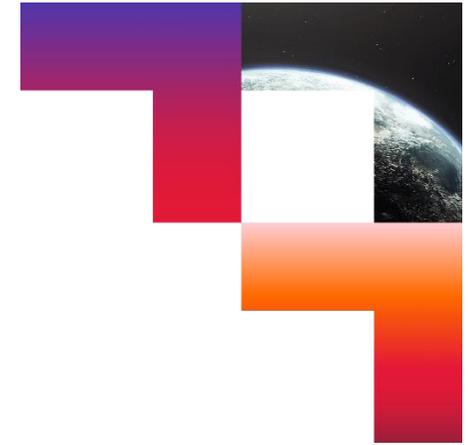
# Pipeline Editor – Edit, Code, Test, Details

The screenshot displays the Pipeline Editor interface for a pipeline named 'alotest1.pipeline.cwl'. The interface is divided into several sections:

- Workspace:** A sidebar on the left showing a file tree with folders like 'team\_areas', 'data', 'example\_pipeline\_input', 'my\_pipelines', and 'out'. The 'out' folder is expanded to show 'alotest1\_1' with files like 'output.json', 'pipeline\_logs.log', and 'usage.json'. Other folders include 'bc\_mcam1', 'test1', 'test2', 'test3', 'timestamp1', 'pulled\_system\_pipelines', 'notebooks', 'test1.txt', and 'test2.txt'. Below this are 'My Pipelines in Workspace' and 'System Pipelines in Workspace'.
- Graph:** The central area shows a workflow graph with nodes: 'sleep in seconds', 'text', 'file', 'figlet', 'simple', 'final.txt', and 'output\_1'. The 'figlet' and 'simple' nodes are highlighted in blue.
- Input(s):** A panel on the right for configuring inputs:
  - [1] text (string): text to be appended to the end of file
  - [2] file (File): test2.txt (with a 'Browse' button)
  - [3] sleep in seconds (int) Optional: 1
- Output:** A panel on the right for configuring outputs:
  - final.txt (Directory): Path for all pipeline output (default: /home/ajoseps/out) (with a 'Browse' button)
  - output\_1 (Directory): Path for all pipeline output (default: /home/ajoseps/out) (with a 'Browse' button)
- Default image:** A panel on the right for configuring the default image tag (with a 'Browse' button).
- Execution logs:** A panel at the bottom showing the pipeline logs:

```
1 [2022-11-11 13:29:40] INFO calrissian 0.10.0 (cwltool 3.0.20210124104916)
2 [2022-11-11 13:29:40] INFO Resolved '/cwl/tmp/cwl/alotest1.pipeline.cwl'
3 [2022-11-11 13:29:43] INFO [workflow ] completed success
4 [2022-11-11 13:29:43] INFO [workflow ] start
5 [2022-11-11 13:29:43] INFO Final process status is success
```
- Buttons:** At the top right, there are icons for workspace, catalogue, and pipeline. Below the graph, there are tabs for 'Graph', 'Code', 'Test', 'Details', 'Push', and 'Pipeline'. A 'Test pipeline' button is located at the bottom right.

# Pipeline Catalogue – Access, Version, Keywords, Push



**Pipeline Step details** [x]

**Name**  
figlet

**Description**  
Runs text input through 'figlet' command and appends it to end of the input file.

**Keywords**  
example x

New keyword  **Add keyword**

**Allowed roles**  
user-management-operator-operator-role x pipeline-services-pipeline-examples-r x user-management-user-user x

**Select roles**

**Update Pipeline Step**

Graph Code Test Details **Push** Pipeline

**Name**  
alotest1

**Description**  
Short system pipeline description

**Keywords**  
No keywords specified.

New keyword  **Add keyword**

**Allowed roles**  
No roles specified. At least one role required.

**Select roles**

**Version**  
0.0.1

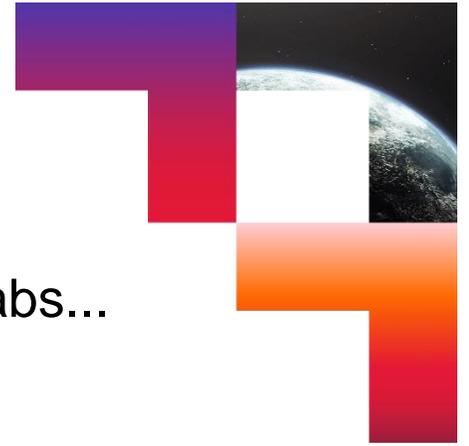
**Version description**  
Changes made in the new version

**Default Execution Engine**  
Calrissian

**Push as System Pipeline**

# Release plan

Version 1.0 of pipelines module available for public use in ESA datalabs...



**v1.0 January 2023**  
and quarterly releases after that

# Benefits of using ESA Datalabs Pipelines



1. Easier access to data which is close to the processing location
2. Easier to share with team(s) and (if needed) also with wider researcher community
3. Easier to define input-output changes
4. Easier to schedule, monitor, stop etc runs
5. Easier to get and modify notifications about run events
6. Easier to set-up and run your code (installation of environment lower levels and dependencies are taken care of by professional IT)
7. Easier to parallelize the runs

# Ideas for additional features for ESA Datalabs pipelines



1. Easier uploading for step code or step configuration in ESA Datalabs pipelines.
2. Clearer way to define whose intellectual property the pipeline consists of and how to refer to it when it was used in your work.
3. Ordering the launching of several pipeline instances in parallel with access to a lot of memory and high performance computing power.
4. Ability to take input from and save output to external cloud data sources (Google Drive, Dropbox, Microsoft Onedrive).
5. More intelligent and dynamic input and output location selection using some pattern.
6. Automatic run when new data is available (e.g. listening to the folder).
7. Ability to obtain input data from databases and save output data to databases.
8. Step code log event generated e-mail notification.
9. Conditional (i.e. branching) pipeline definition.

**Your feature wishes and priorities are very welcome for planning the next releases.**

# Pipelines Hands-On session 25.11.22 11:30-12:30

1. Please register **TODAY** to become a user of the pipelines demo environment for tomorrow's hands-on session – the background processing of registration is slow.
2. Connect to wireless network „ESA“
3. URL to start registration: <https://datalabs-pipeman.situk8spro.ebe.lan>
4. NB! this is a separate environment from the other hands-on sessions, so separate registration is necessary.

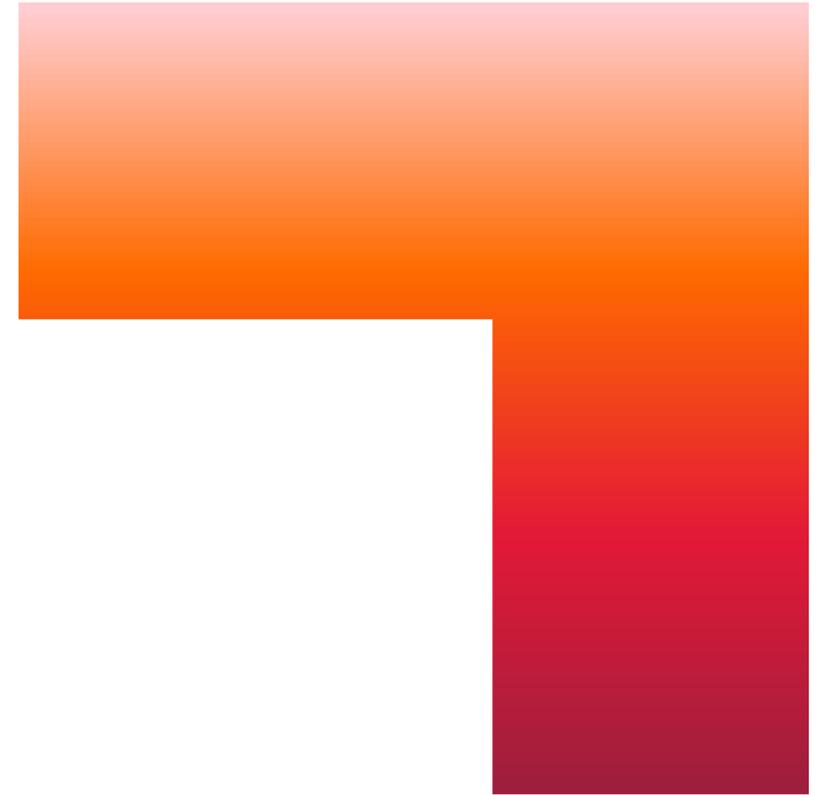


Your feedback,  
questions and  
feature requests  
about pipelines in  
ESA Datalabs are  
welcome!

Write to [alo.joosepson@cgi.com](mailto:alo.joosepson@cgi.com)

Thank you! Gracias!

[cgi.com](http://cgi.com)



**CGI**