

 **Welcome!**

## Contributing to EDITO Datalab

Learn how to contribute your knowledge to [\*\*EDITO Tutorials\*\*](#).

Presented by **Samuel Fooks**  
*Flanders Marine Institute (VLIZ)*

For all the PDFs and code, check out the workshop [\*\*GitHub repository\*\*](#)

# Adding our tutorial to the EDITO Tutorials

► Tutorials



# What We'll Go Over

- How to become a contributor to EDITO tutorials
- Create a shareable tutorial
- Share it publicly via **GitHub**
- Launch it on **EDITO Datalab**
- Register it using `tutorials.json`
- Submit a **merge request**

All this is also covered in [EDITO Datalab Documentation.](#)

# Get an Account on EDITO

🌐 Become a Beta Tester:  
[Sign up here](#)

✉️ Receive an Email:  
You will receive an email from the developer team with further instructions.

🔑 Sign up to Mercator Ocean GitLab:  
[Create your account](#)



# Contribute to the Tutorials Content Repository

## 🔗 Access Repositories:

Once your account is created, you will be added as a developer to the following repositories:

- [Service Playground Repository](#)
- [Process Playground Repository](#)
- [Tutorial Content Repository](#)



# I have a new tool/script to share

✓ For example, I've written a tutorial in `.Rmd` :

- It explains how to use a tool or perform a task
- Includes **Markdown** text and **R code chunks**
- Shows plots, tables, or results inline
- Has some interactivity/user interaction

# Example: My Tutorial on Accessing EDITO STAC

Here in this repository

[`/add\_tutorial/my\_stac\_rTutorial/stac\_rTutorial.Rmd`](#)

```
45  
46 ## Querying the STAC API  
47  
48 This script demonstrates how to query a STAC API and download data.  
49  
Run Chunk | Run Above  
50 ```{r stac-query-collections}  
51 # Define the root STAC API endpoint  
52 stac_endpoint <- "https://api.dive.edito.eu/data/"  
53  
54 # Query the root STAC API to get collections  
55 collections_query <- stac(stac_endpoint) %>%  
56 | rstac::collections() %>%  
57 | get_request()  
58  
59 # Display the number of collections  
60 cat("Number of collections:", length(collections_query$collections), "\n")  
61  
62  
Run Chunk | Run Above  
63 ```{r stac-query-occurrence}  
64 # Filter collections with 'occurrence' in their name or description directly  
65 occurrence_collections <- Filter(function(collection) {  
66   grep("occurrence", collection$title, ignore.case = TRUE)  
67 }, collections_query$collections)  
68  
69 # Display the occurrence collections
```

The screenshot shows a Jupyter Notebook interface. On the left, there is a code cell containing R code for querying a STAC API. On the right, the output pane shows a table with two rows: 'contributing-edito' and 'DTO-Bioflow-M13-technical-'. The 'contributing-edito' row has a status of 'AB' and 'No results'. The 'DTO-Bioflow-M13-technical-' row also has a status of 'AB' and 'No results'. The interface includes standard Jupyter Notebook controls like up and down arrows, a search bar, and a close button.

# Recommended Folder Structure

- Not mandatory but on EDITO we need applications that are resilient and understandable for everyone
- Include a good README.md [makeareadme.com](https://makeareadme.com)
- Data and other assets separate

```
my_stac_r_tutorial/
├── stac-r-tutorial.Rmd
├── data/
└── docker-compose.yml(*optional)
    README.md
```

# Create a Repository on your GitHub

- Go to [github.com](https://github.com)
- Click **New repository**
- Set it to **Public**

A demonstration on YouTube

Creating Your First GitHub Repository and Pushing Code [Youtube](#)



# Push Your Local Code to Your Github

```
# Initialize Git in your local directory (if not already initialized)
git init
# Add all files to the staging area
git add .
git config user.name username
git config user.email usermail@mail.com
# Commit the changes
git commit -m "Initial commit"
# Add the remote origin
git remote add origin https://github.com/username/stac-r-tutorial.git
# Push the changes to GitHub
git branch -M main
git push -u origin main
```

Creating Your First GitHub Repository and Pushing Code [Youtube](#)

# ⚙️ Make your deployment URL

## EDITO Services

### Access the Service Configuration

- Choose a service from the Service Catalog appropriate for your Tutorial
- ex. R Studio, Jupyter-python

### Add Your GitHub Repository

- In the `GIT` section add the url to your tutorial's github repository in the Repository field

### Set Resource Limits

- In the resources section, adjust CPU and memory limits as needed
- e.g., `1600m` for CPU, `5Gi` for memory

# Save Configuration and Test your tutorial

## Save the Configuration

- Click **Save** to store your settings.

## Copy the URL in your browser [deployment\\_url](#)

- This is the link used to deploy your service and clone your github into the service

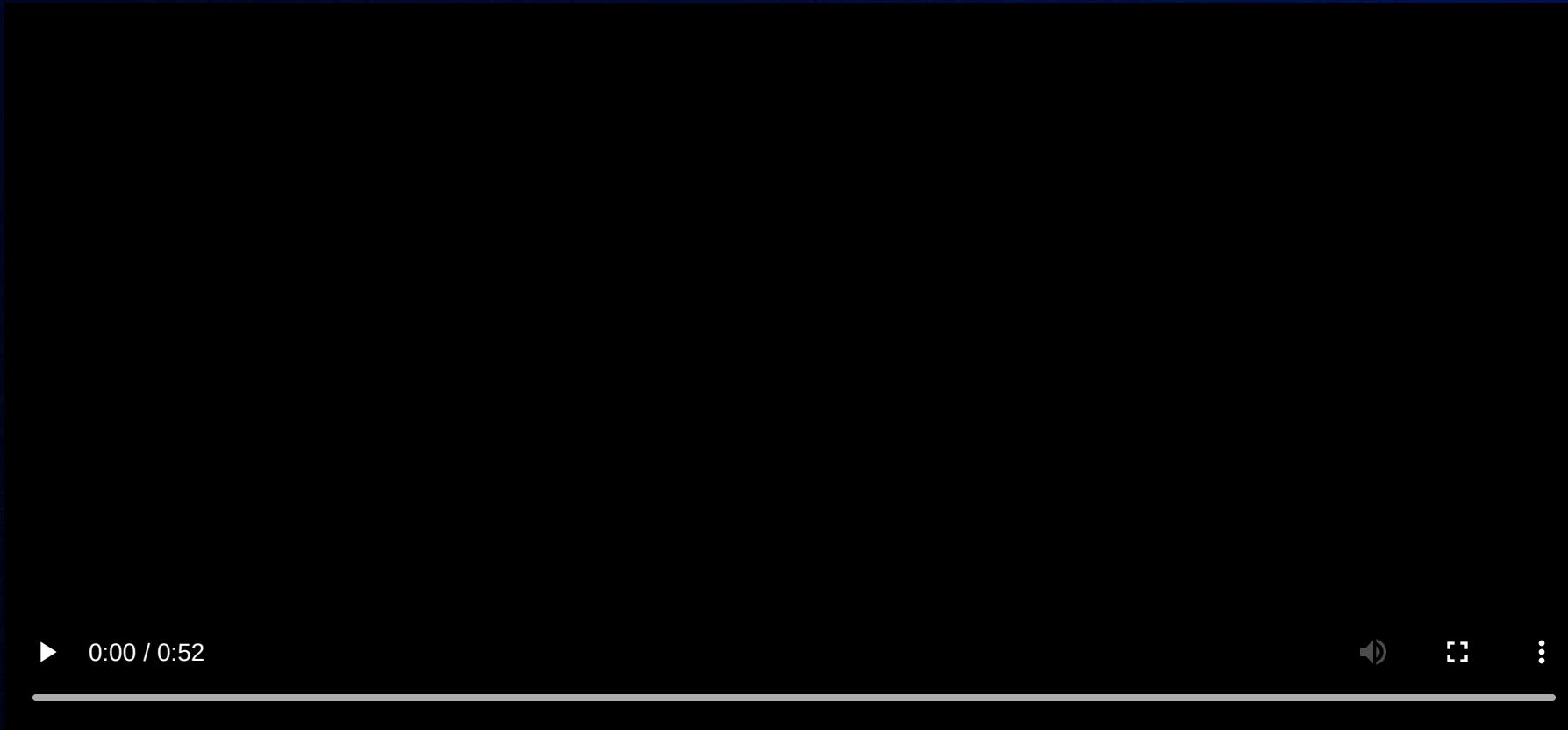
## Launch the Service

- Use the **Launch** button to start the service with your configuration.

## Test Your Tutorial

- Verify that the service clones your github, and your tutorial can be run
- Does it install the right packages?

# Configuring EDITO Service



# Tutorials repository and the tutorials.json

In order to add our tutorial to the EDITO tutorials we need to add it to the  
tutorials.json list

<https://gitlab.mercator-ocean.fr/pub/edito-infra/edito-tutorials-content/tutorials.json>

We will clone this repository and add our tutorial to this list, using the template provided in the README

# Clone the Tutorials Repository to your Local PC

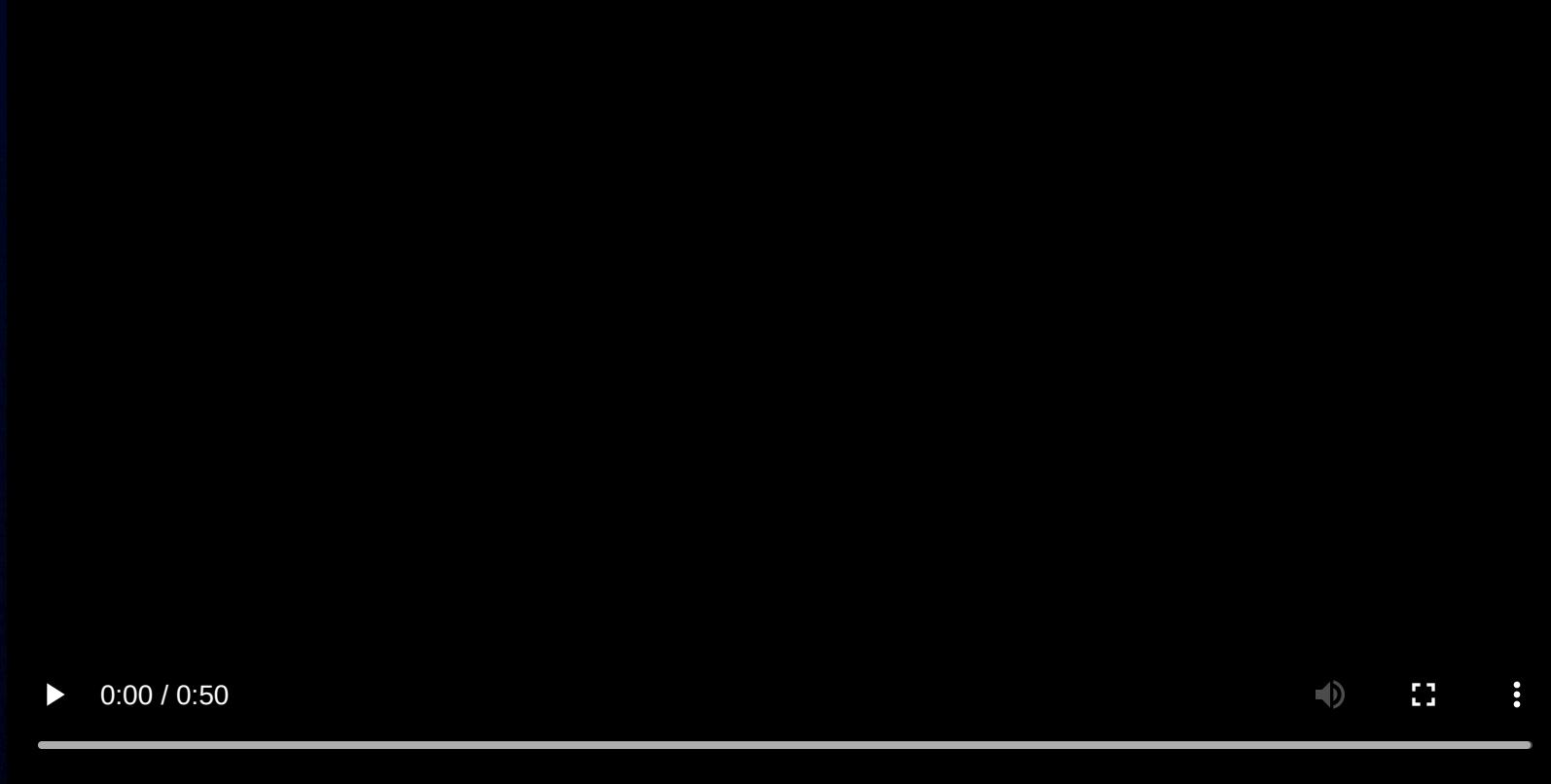
## EDITO GitLab Tutorials:

<https://gitlab.mercator-ocean.fr/pub/edito-infra/edito-tutorials-content>

## Clone the Repo:

```
git clone https://gitlab.mercator-ocean.fr/pub/edito-infra/edito-tutorials-content.git
```

# Cloning EDITO Tutorials Content



# 🛠 Make a new branch

🌿 \*\*Create a New Branch\*\*:

```
git checkout -b my-new-tutorial-branch
```

⬆ Push the New Branch:

```
git push origin my-new-tutorial-branch
```

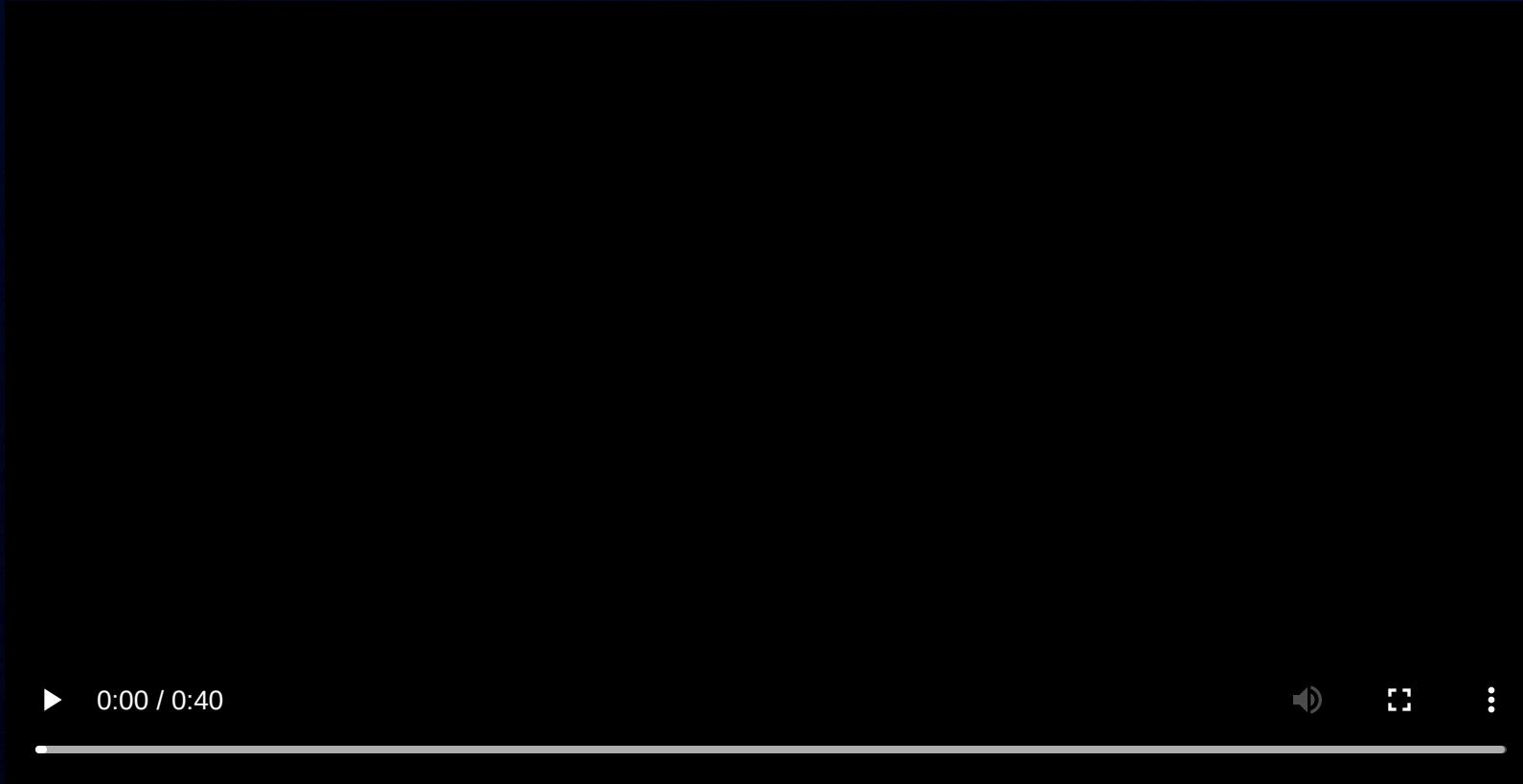


European Digital  
Twin Ocean

supported by



# Make branch



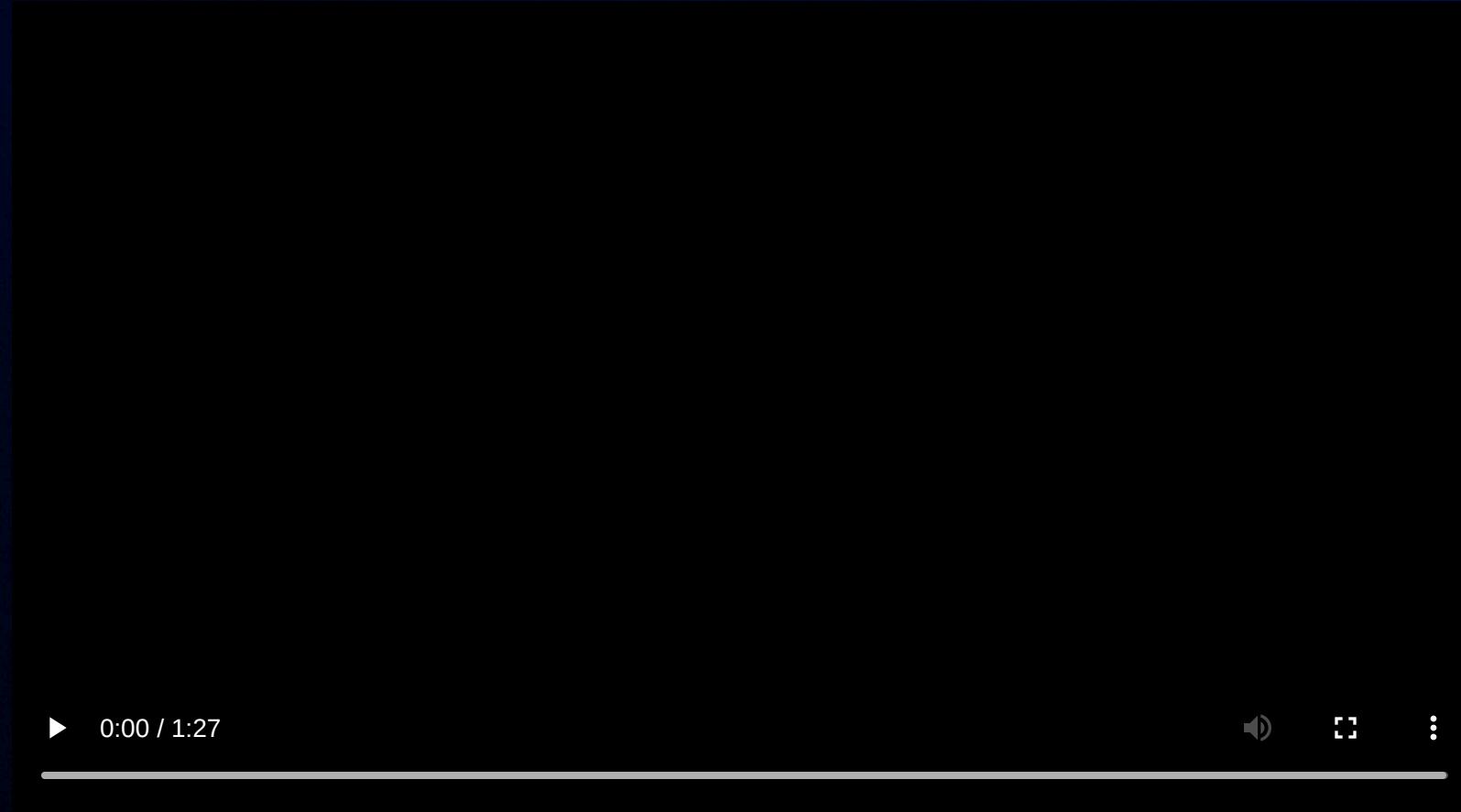
# Add your tutorial to 'tutorials.json'

## Deployment URL from previous step

```
{  
  "name": {  
    "en": "My New Tutorial"  
  },  
  "abstract": {  
    "en": "A short description of your tutorial"  
  },  
  "authors": [  
    "The authors and contributors"  
  ],  
  "types": [  
    "types": [  
      {  
        "en": "Tutorial"  
      }  
    ],  
    "tags": [  
      "create",  
    ],  
    "category": "training courses in data science", // "What-If applications", "Focus applications", "training courses in data science"  
    "imageUrl": "https://www.edito.eu/wp-content/uploads/2023/09/favicon.png",  
    "articleUrl": {  
      "en": "https://github.com/username/stac-r-tutorial", // Your github  
    }  
    "deploymentUrl": "https://datalab.dive.edito.eu/mydeployment.configuration.git.resources.etc"// DEPLOYMENT URL FROM PREVIOUS STEP  
    // parts: []  
}
```

## Adding to tutorials.json

Ex. A tutorial on accessing data via STAC in R and how to subset ARCO data.



# Push your updates onto your branch

```
# Stage all changes  
git add .  
# Commit the changes with a descriptive message  
git commit -m "Added my awesome tutorial to tutorials.json"  
# Push the changes to your branch  
git push origin my-new-tutorial-branch
```

# Create a Merge Request

- Check the gitlab <https://gitlab.mercator-ocean.fr/pub/edito-infra/edito-tutorials-content>
- See if your commit is in a pipeline and if it passes or not
- If it passes, create a Merge Request
- In your merge request, '@pub/edito-infra/codeowners' to request code owners to review your proposal.

# ✓ Final Review Checklist

- ✓ Tutorial .Rmd created and runs
- ➡ GitHub repo is public and clean
- 🔗 Launch link tested
- ✓ tutorials.json updated
- ✓ Committed to Gitlab and passes Pipeline
- ✓ Merge Request submitted

# 🙌 Done!

Once your Merge Request is approved

🎉 You've contributed to EDITO Datalab!

Your tutorial is now one click away from reproducible research!

💬 Issues? Email [edito-infra-dev@mercator-ocean.eu](mailto:edito-infra-dev@mercator-ocean.eu)

🔗 [Contribution Docs](#)

