



# eFlows4HPC

## HPC WORKFLOWS AS A SERVICE AND SOFTWARE STACK HANDS-ON

Jedrzej Rybicki ([j.rybicki@fz-juelich.de](mailto:j.rybicki@fz-juelich.de)), Albertin, Loïc ([loic.albertin@atos.net](mailto:loic.albertin@atos.net))

Jorge Ejarque ([jorge.ejarque@bsc.es](mailto:jorge.ejarque@bsc.es))

14th September 2022



This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

# Hands-on setup

For this hands-on you need:

- A computer
- An internet connection
- A user in an HPC cluster and eFlow4HPC services

# First things first ensure that you can reach BSC platforms

```
ssh -o PreferredAuthentications=password -o  
PubkeyAuthentication=no <user_name>@nord3.bsc.es
```

```
# Create some directories that will be used  
later
```

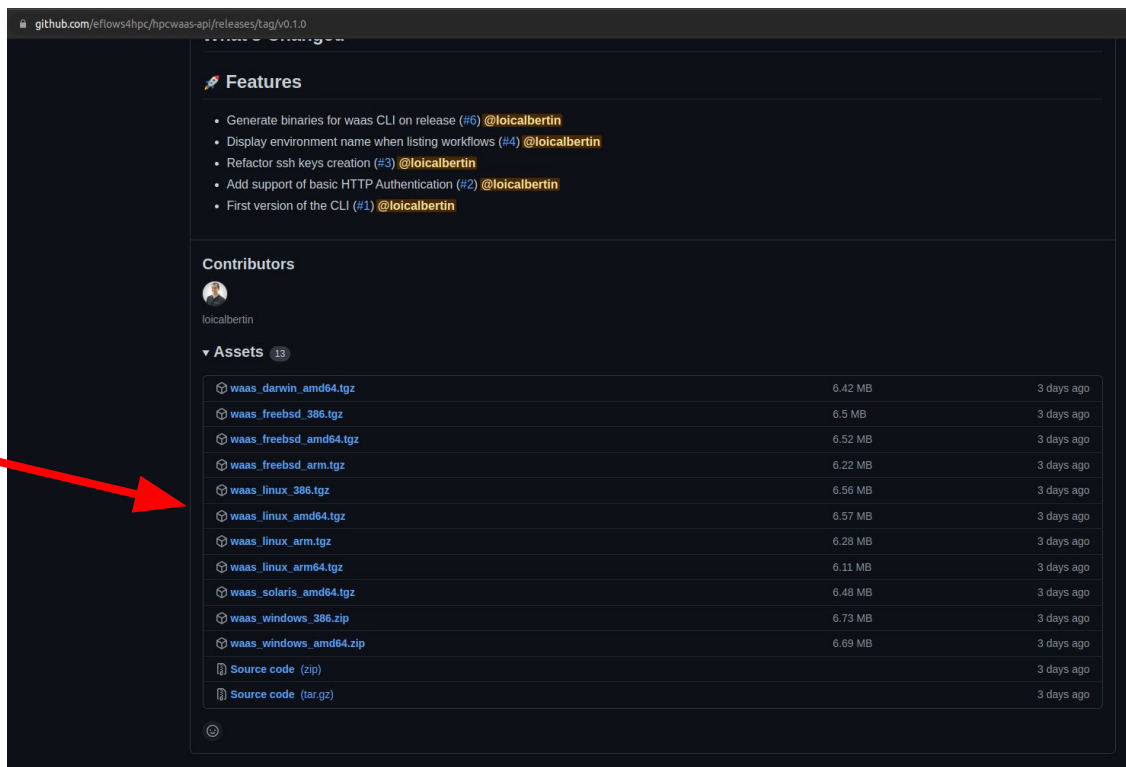
```
mkdir -p /gpfs/projects/nct01/<user_id>
```

```
mkdir -p /home/nct01/<user_id>/data/inputs
```

```
mkdir -p /home/nct01/<user_id>/data/results
```

# Get the HPCWaaS CLI from github

Explore <https://github.com/eflows4hpc/hpcwaas-api/releases/tag/v0.1.0> to get a binary that matches your computer.



The screenshot shows the GitHub release page for the repository `eflows4hpc/hpcwaas-api` at tag `v0.1.0`. The page is dark-themed and displays the following sections:

- Features:**
  - Generate binaries for waas CLI on release (#6) @loicalbertin
  - Display environment name when listing workflows (#4) @loicalbertin
  - Refactor ssh keys creation (#3) @loicalbertin
  - Add support of basic HTTP Authentication (#2) @loicalbertin
  - First version of the CLI (#1) @loicalbertin
- Contributors:**
  - loicalbertin
- Assets (13):**

Asset	Size	Time
<a href="#">waas_darwin_amd64.tgz</a>	6.42 MB	3 days ago
<a href="#">waas_freebsd_386.tgz</a>	6.5 MB	3 days ago
<a href="#">waas_freebsd_amd64.tgz</a>	6.52 MB	3 days ago
<a href="#">waas_freebsd_arm.tgz</a>	6.22 MB	3 days ago
<a href="#">waas_linux_386.tgz</a>	6.56 MB	3 days ago
<a href="#">waas_linux_amd64.tgz</a>	6.57 MB	3 days ago
<a href="#">waas_linux_arm.tgz</a>	6.28 MB	3 days ago
<a href="#">waas_linux_arm64.tgz</a>	6.11 MB	3 days ago
<a href="#">waas_solaris_amd64.tgz</a>	6.48 MB	3 days ago
<a href="#">waas_windows_386.zip</a>	6.73 MB	3 days ago
<a href="#">waas_windows_amd64.zip</a>	6.69 MB	3 days ago
<a href="#">Source code (zip)</a>		3 days ago
<a href="#">Source code (tar.gz)</a>		3 days ago

# Alternatively get the docker version

Simply run:

```
docker run -ti --rm ghcr.io/eflows4hpc/hpcwaas-api:main-cli help
```

this is equivalent to

```
./waas help
```

We will use the later form in this hands-on.

# Generate an SSH keypair

```
./waas --api_url https://eflows4hpc.bsc.es/waas -u <user>:<password> ssh_keys  
key-gen
```

INFO: Below is your newly generated SSH public key.

INFO: Take note of it as you will not see it again.

INFO: You are responsible for adding it to the `authorized_keys` file on the systems you want to run your workflows.

INFO: SSH key ID: 31...3f

INFO: SSH Public key: ssh-rsa AAA...mH

**Credentials are the same as for your bsc account.**

**The private key is stored in Vault and never provided to you.**

**Copy the public key on nord3 in `~/.ssh/authorized_keys` file to allow access to your account by the stack.**

**Also copy the SSH Key ID that identifies your keypair and which is required to deploy and run workflows.**

## Now let's create, configure and deploy a workflow

The goal in this hands-on is not to learn how to develop an application in Tosca.

For today we will focus on how to use the stack.

We did a Tosca tutorial during the first year of eFlows4HPC.

This tutorial is publically available on github:  
<https://github.com/eflows4hpc/tosca-tutorial>

# Now let's create, configure and deploy a workflow

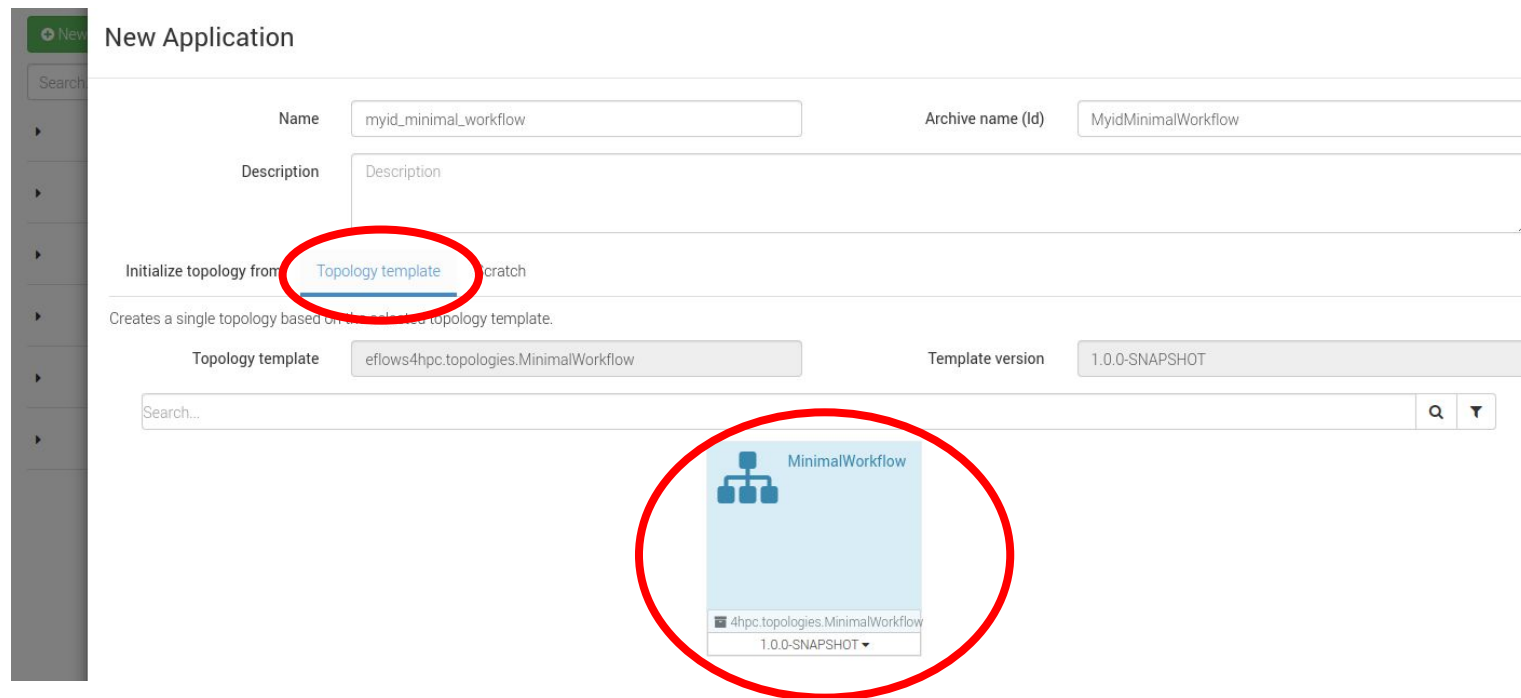
Navigate to <https://eflows4hpc.bsc.es/alien/#/>

And log into Alien4Cloud. Credentials are the same as your BSC account.  
Click on “Applications” on top left tab, then click on “New Application”.



# Now let's create, configure and deploy a workflow

Give a name (should be unique) to your application. And initialize it from the minimal workflow template



New Application

Name: myid\_minimal\_workflow

Archive name (Id): MyidMinimalWorkflow

Description: Description

Initialize topology from: Topology template | scratch

Creates a single topology based on the selected topology template.

Topology template: eflows4hpc.topologies.MinimalWorkflow

Template version: 1.0.0-SNAPSHOT


Search...

MinimalWorkflow

4hpc.topologies.MinimalWorkflow

1.0.0-SNAPSHOT

# Now let's create, configure and deploy a workflow

 Applications Catalog Administration

myid\_minimal\_workflow

Drop an image file, or [browse](#).

## myid\_minimal\_work...

[description](#)

ID MyidMinimalWorkflow

Creation date Thu, Sep 8, 2022 3:37 PM

Update date Thu, Sep 8, 2022 3:37 PM

Versions


Environm  
ents

Variables

Users and  
Groups

Delete

Tags



### Work on an environment

Prepare next deployment for an environment: associate a version, edit it's topology, configure deployment options and deploy.  
Manage the current deployment for the environment.

Search...

Name	Type	Next version	Deployed version
● Environment	PRODUCTION	0.1.0-SNAPSHOT	

### Edit a version's topology

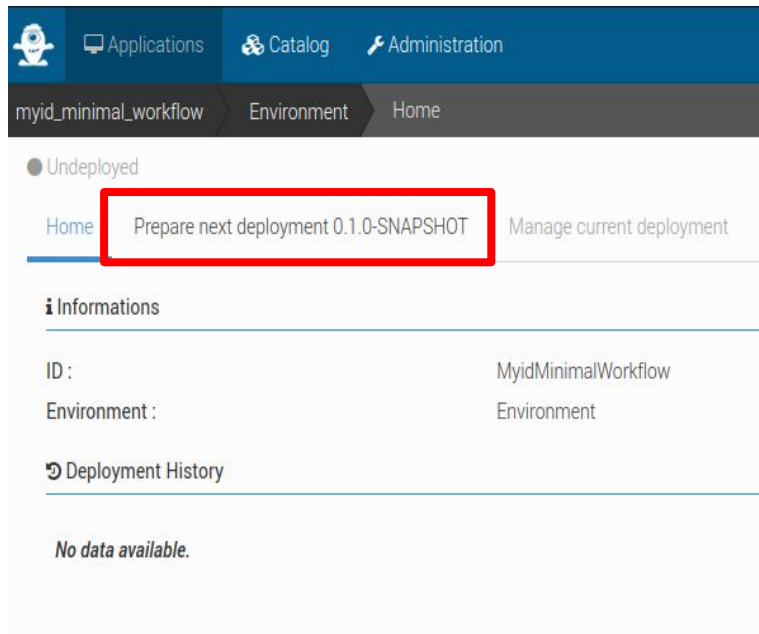
Edit a version's topology out of the scope of an environment (allow the edition of a version that is not currently selected for deployment on any environment)

Search...

Version	Description
</> 0.1.0-SNAPSHOT	
0.1.0-SNAPSHOT	default topology

10

# Now let's create, configure and deploy a workflow



myid\_minimal\_workflow Environment Home

● Undeployed

Home **Prepare next deployment 0.1.0-SNAPSHOT** Manage current deployment

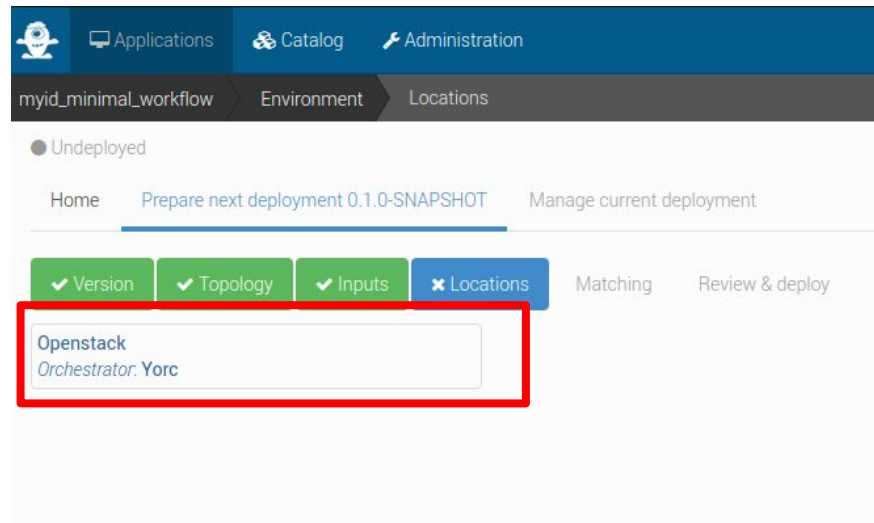
**i** Informations

ID : MyidMinimalWorkflow

Environment : Environment

**D** Deployment History

*No data available.*



myid\_minimal\_workflow Environment Locations

● Undeployed

Home Prepare next deployment 0.1.0-SNAPSHOT Manage current deployment

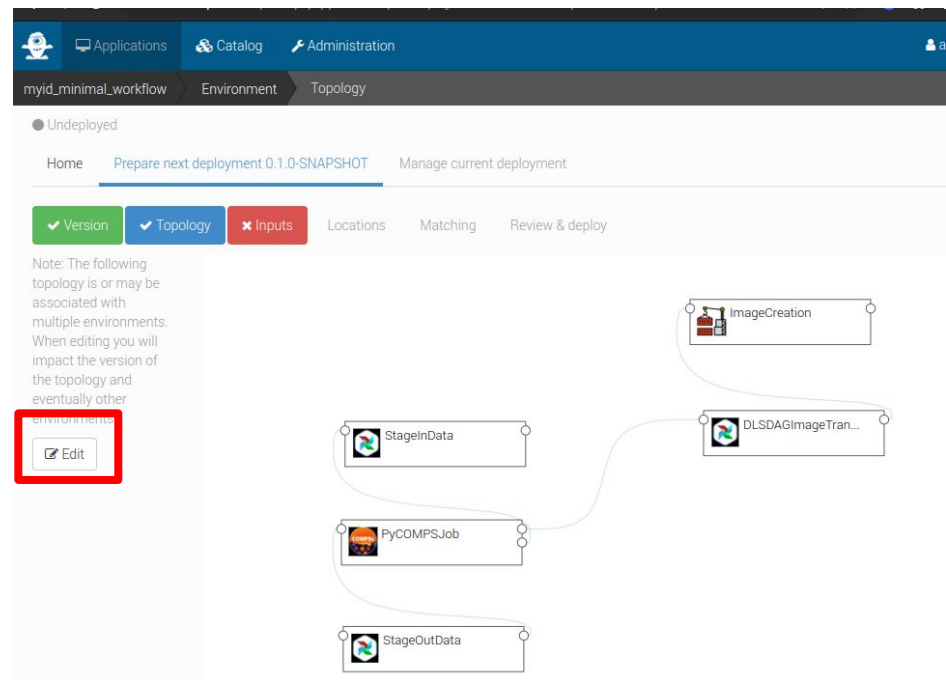
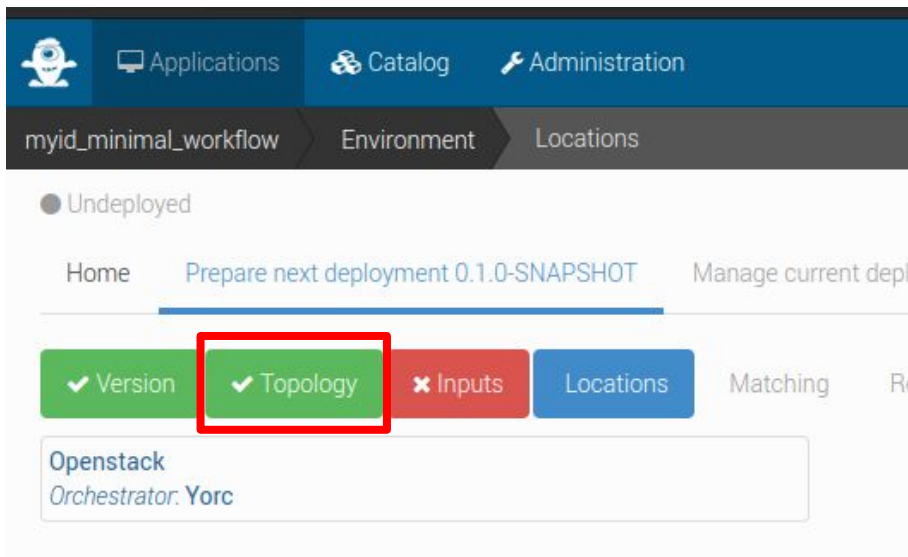
✓ Version ✓ Topology ✓ Inputs **✗ Locations** Matching Review & deploy

**Openstack**  
Orchestrator: Yorc

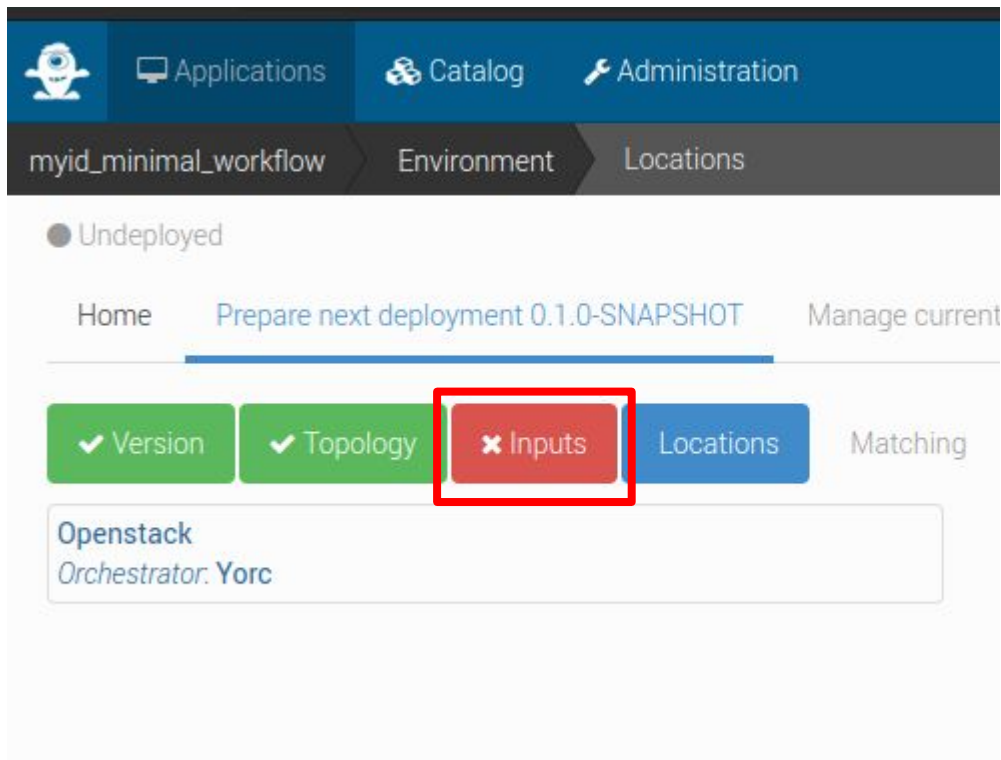
# Now let's create, configure and deploy a workflow

At this point you can have a break to explore the topology of your application.

But make sure to not change it ;)

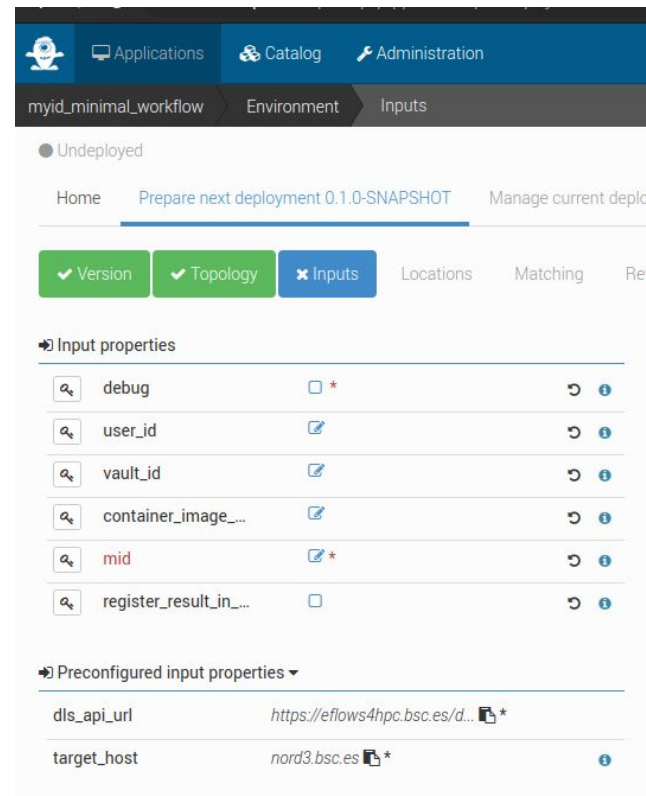


# Now let's create, configure and deploy a workflow



The screenshot shows the eFlows4HPC interface with the following elements:

- Top navigation bar: Applications, Catalog, Administration.
- Sub-navigation bar: myid\_minimal\_workflow, Environment, Locations.
- Status: Undeployed.
- Home button and links: Prepare next deployment 0.1.0-SNAPSHOT, Manage current.
- Buttons: Version (green), Topology (green), **Inputs (red, highlighted with a red box)**, Locations (blue), Matching.
- Text: Openstack, Orchestrator: Yorc.



The screenshot shows the eFlows4HPC interface with the following elements:

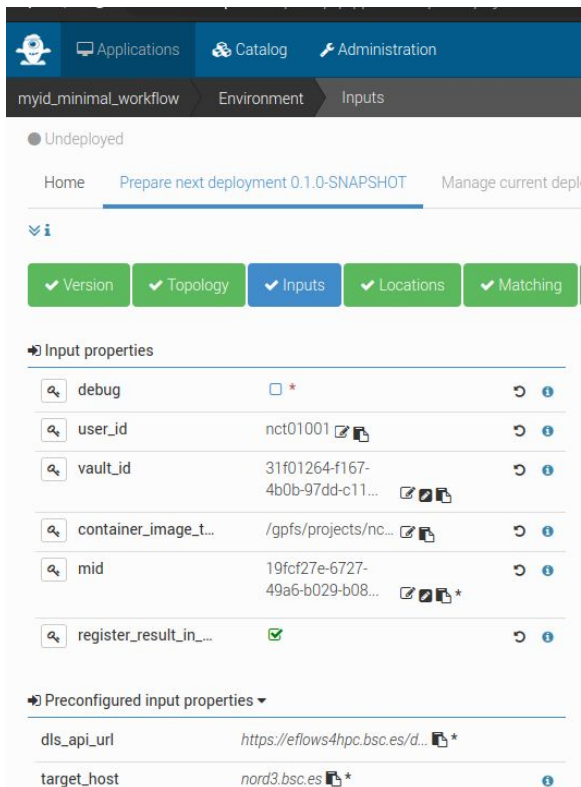
- Top navigation bar: Applications, Catalog, Administration.
- Sub-navigation bar: myid\_minimal\_workflow, Environment, Inputs.
- Status: Undeployed.
- Home button and links: Prepare next deployment 0.1.0-SNAPSHOT, Manage current.
- Buttons: Version (green), Topology (green), **Inputs (blue, active)**, Locations, Matching, Re.
- Section: Input properties
- Table of input properties:

Property	Value	Required	Info
debug	<input type="checkbox"/>	*	Info
user_id	<input type="text"/>		Info
vault_id	<input type="text"/>		Info
container_image_...	<input type="text"/>		Info
mid	<input type="text"/>	*	Info
register_result_in_...	<input type="checkbox"/>		Info









- Section: Preconfigured input properties
- Table of preconfigured input properties:

Property	Value	Required	Info
dls_api_url	https://eflows4hpc.bsc.es/d...	*	Info
target_host	nord3.bsc.es	*	Info



# Now let's create, configure and deploy a workflow



The screenshot shows the eFlows4HPC web interface. The top navigation bar includes 'Applications', 'Catalog', and 'Administration'. Below it, the 'myid\_minimal\_workflow' is selected, with tabs for 'Environment' and 'Inputs'. The 'Inputs' tab is active, showing a list of input properties. The 'Input properties' section includes a search bar and a table of inputs. The 'Preconfigured input properties' section is also visible.

Input	Value	Actions
debug	<input type="checkbox"/> *	🔍 ⓘ
user_id	nct01001  	🔍 ⓘ
vault_id	31f01264-f167-4b0b-97dd-c11...  	🔍 ⓘ
container_image_t...	/gpfs/projects/nc...  	🔍 ⓘ
mid	19fcf27e-6727-49a6-b029-b08...   *	🔍 ⓘ
register_result_in...	<input checked="" type="checkbox"/>	🔍 ⓘ

Property	Value	Actions
dls_api_url	https://eflows4hpc.bsc.es/d...  *	
target_host	nord3.bsc.es  *	ⓘ

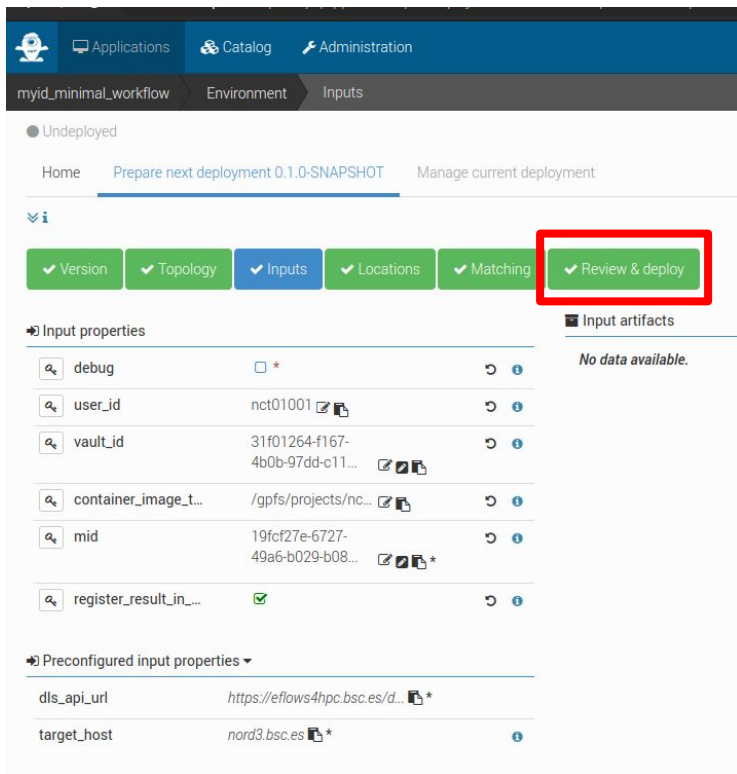
user id is the user account that will be used to connect to nord3 to transfer the generated container image (it's your BSC account)

vault id is the SSH keypair id you generated using waas CLI - remember we asked to keep it next to you ;)

container image transfer directory is where the image will be transferred. Keep in mind that you are the developer for this workflow but not necessarily the unique end-user. So you should store it in a shared folder. On nord3 the path `/gpfs/projects/nct01/<user_id>` will do. The directory **SHOULD EXIST** before deploying.

mid is a metadata id that is used to store the results of the workflow execution to the data catalogue. For today it should be 19fcf27e-6727-49a6-b029-b08c59e8e38b

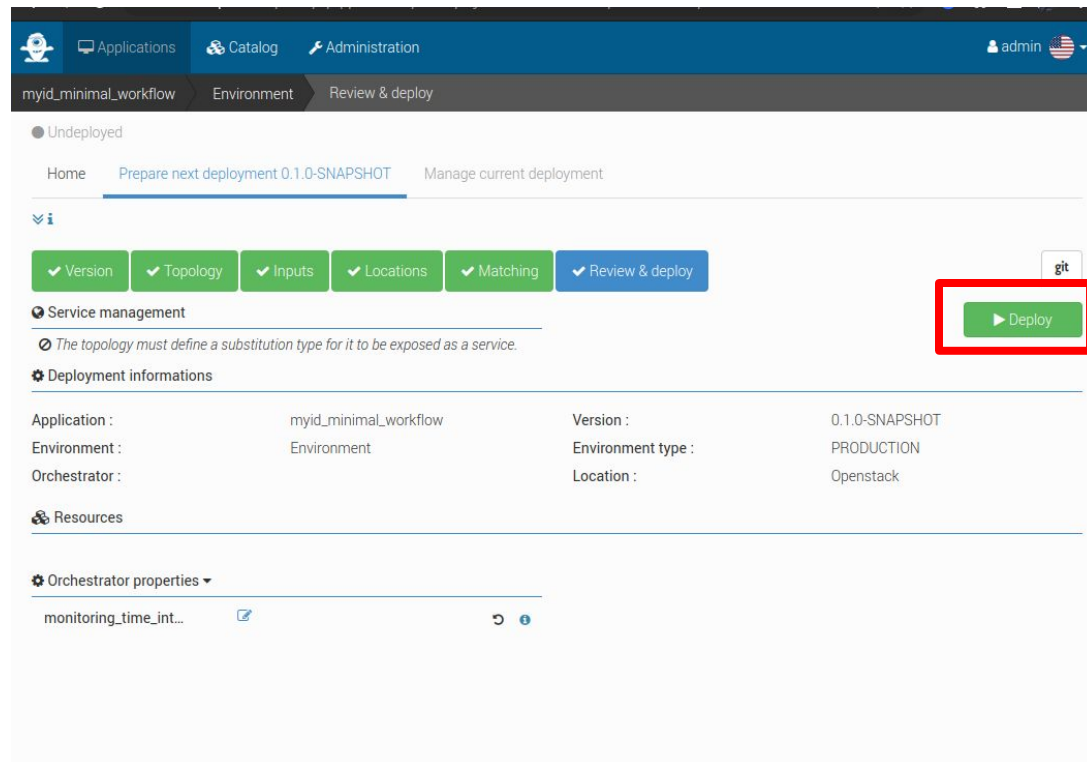
# Now let's create, configure and deploy a workflow



The screenshot shows the 'Inputs' tab of the 'myid\_minimal\_workflow' configuration page. The top navigation bar includes 'Applications', 'Catalog', and 'Administration'. The breadcrumb trail is 'myid\_minimal\_workflow' > 'Environment' > 'Inputs'. Below the navigation, there are tabs for 'Version', 'Topology', 'Inputs', 'Locations', 'Matching', and 'Review & deploy'. The 'Review & deploy' tab is highlighted with a red box. The main content area is divided into 'Input properties' and 'Preconfigured input properties'. The 'Input properties' section lists several inputs with their values and icons for editing or deleting. The 'Preconfigured input properties' section lists two inputs with their values and icons.

Input	Value	Actions
debug	<input type="checkbox"/> *	🔍 ⚙️
user_id	nct01001 📄	🔍 ⚙️
vault_id	31f01264-f167-4b0b-97dd-c11... 📄 📄 📄	🔍 ⚙️
container_image_t...	/gifs/projects/nc... 📄 📄	🔍 ⚙️
mid	19fcf27e-6727-49a6-b029-b08... 📄 📄 📄 *	🔍 ⚙️
register_result_in...	✅	🔍 ⚙️

Property	Value	Actions
dls_api_url	https://eflows4hpc.bsc.es/d... 📄 *	🔍
target_host	nord3.bsc.es 📄 *	🔍



The screenshot shows the 'Review & deploy' tab of the 'myid\_minimal\_workflow' configuration page. The top navigation bar includes 'Applications', 'Catalog', and 'Administration'. The breadcrumb trail is 'myid\_minimal\_workflow' > 'Environment' > 'Review & deploy'. Below the navigation, there are tabs for 'Version', 'Topology', 'Inputs', 'Locations', 'Matching', and 'Review & deploy'. The 'Review & deploy' tab is highlighted with a red box. The main content area includes a 'Deploy' button (highlighted with a red box), 'Service management' section, 'Deployment informations' section, and 'Orchestrator properties' section.

**Deploy** (highlighted with a red box)

**Service management**

The topology must define a substitution type for it to be exposed as a service.

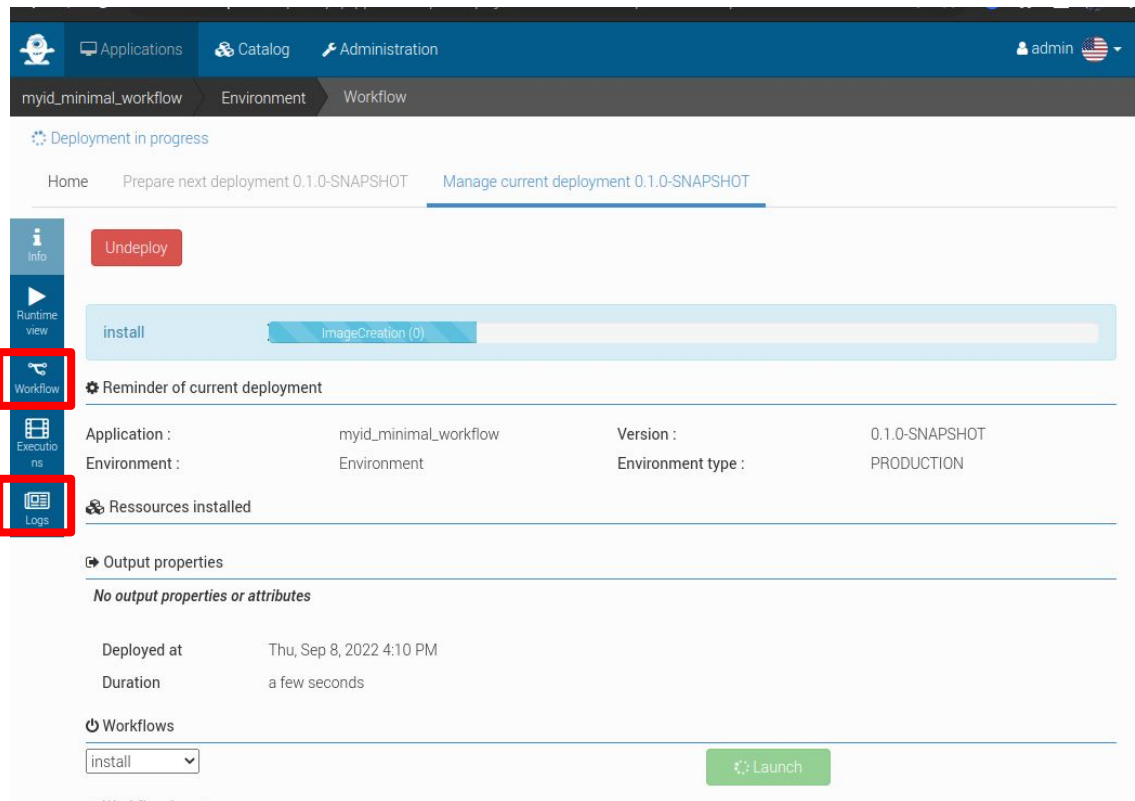
**Deployment informations**

Property	Value
Application :	myid_minimal_workflow
Environment :	Environment
Orchestrator :	
Version :	0.1.0-SNAPSHOT
Environment type :	PRODUCTION
Location :	Openstack

**Orchestrator properties**

Property	Value	Actions
monitoring_time_int...	📄	🔍 ⚙️

# Now let's create, configure and deploy a workflow



The screenshot shows the eFlows4HPC interface with the 'Workflow' tab selected. The left sidebar has 'Workflow' and 'Logs' highlighted with red boxes. The main content area shows the deployment progress of the 'myid\_minimal\_workflow' in the 'Environment'.

Deployment in progress

Home Prepare next deployment 0.1.0-SNAPSHOT **Manage current deployment 0.1.0-SNAPSHOT**

Undeploy

install ImageCreation (0)

**Reminder of current deployment**

Application :	myid_minimal_workflow	Version :	0.1.0-SNAPSHOT
Environment :	Environment	Environment type :	PRODUCTION

**Ressources installed**

**Output properties**

No output properties or attributes

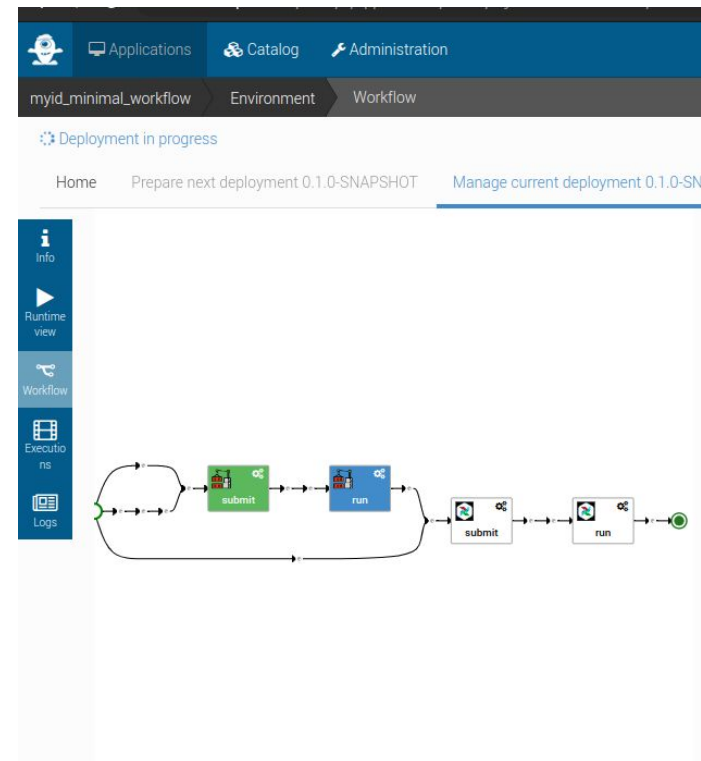
Deployed at Thu, Sep 8, 2022 4:10 PM

Duration a few seconds

**Workflows**


install

Launch





# Now let's create, configure and deploy a workflow

 Applications Catalog Administration admin

myid\_minimal\_workflow Environment Logs

Deployment in progress

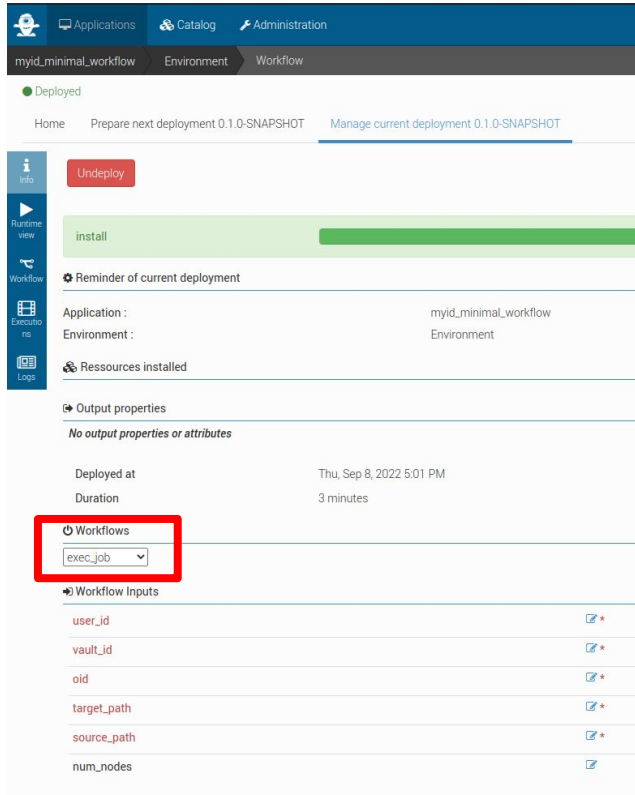
Home Prepare next deployment 0.1.0-SNAPSHOT Manage current deployment 0.1.0-SNAPSHOT

From To Search 2 s

- Info
- Runtime view
- Workflow
- Execution
- Logs

```
[2022-09-08 16:10:06] [install] [StageOutData] Status for node "StageOutData", instance "0" changed to "initial"
[2022-09-08 16:10:06] [] [StageOutData] Attribute "state" value for node "StageOutData", instance "0" changed to "initial"
[2022-09-08 16:10:06] [install] [] DeploymentID: "myid_minimal_workflow-Environment", Workflow: "install", step: "StageOutData_initial" ended successfully
[2022-09-08 16:10:06] [install] [] Start processing workflow step install:ImageCreation_submitting
[2022-09-08 16:10:06] [install] [ImageCreation] Status for node "ImageCreation", instance "0" changed to "submitting"
[2022-09-08 16:10:06] [] [ImageCreation] Attribute "state" value for node "ImageCreation", instance "0" changed to "submitting"
[2022-09-08 16:10:06] [install] [] DeploymentID: "myid_minimal_workflow-Environment", Workflow: "install", step: "ImageCreation_submitting" ended successfully
[2022-09-08 16:10:07] [install] [] Start processing workflow step install:ImageCreation_submit
[2022-09-08 16:10:07] [install] [ImageCreation] Status for workflow step "ImageCreation_submit" changed to "INITIAL"
[2022-09-08 16:10:07] [install] [ImageCreation] Status for task execution "2a561e47-e04e-4e34-8ff1-f8ac8094b9b4-0" changed to "INITIAL"
[2022-09-08 16:10:07] [install] [ImageCreation] Status for workflow step "ImageCreation_submit" changed to "RUNNING"
[2022-09-08 16:10:07] [install] [ImageCreation] Status for task execution "2a561e47-e04e-4e34-8ff1-f8ac8094b9b4-0" changed to "RUNNING"
[2022-09-08 16:10:07] [install] [ImageCreation] Start the ansible execution of: ImageCreation with operation : tosca.interfaces.node.lifecycle.runnable.submit
[2022-09-08 16:10:11] [install] [ImageCreation] Ansible task output:
PLAY [Image Creation Submission] TASK [Gathering Facts]
ok: [ImageCreation_0]
[2022-09-08 16:10:11] [install] [ImageCreation] Ansible task output:
PLAY [Image Creation Submission] TASK [Set payload]
ok: [ImageCreation_0] => changed=false
ansible_facts:
payload:
force: 'false'
machines:
```

# Now as a Developer let's test our workflow



The screenshot shows the eFlows4HPC web interface. The top navigation bar includes 'Applications', 'Catalog', and 'Administration'. The main content area shows the 'myid\_minimal\_workflow' deployment page. The 'Workflows' dropdown menu is open, and 'exec\_job' is selected. Below the dropdown, the 'Workflow Inputs' section lists several inputs: 'user\_id', 'vault\_id', 'oid', 'target\_path', 'source\_path', and 'num\_nodes'. Each input has a red asterisk icon next to it, indicating it is required.

Select the “exec\_job” workflow and fill inputs:

user\_id & vault\_id are the account credential used to run the job just use the same values as just before deploying

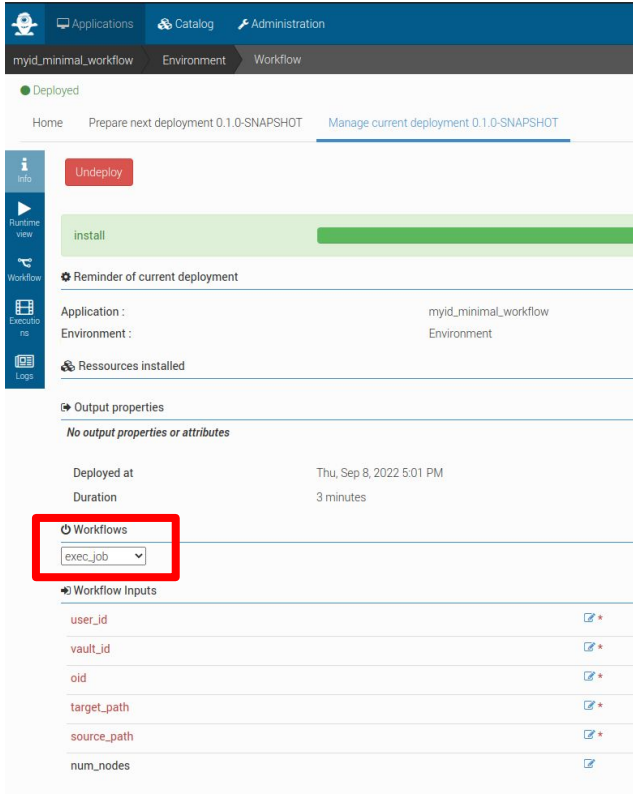
oid is an object id in the data catalogue referencing the workflow input data. For today it should be 2c2463377aac4aa59381c6b06fe800f3

target\_path is the path to a directory where the DLS will store input data and will be used as input for the PyCOMPSs job. Typically use some directory in your home, like /home/nct01/<user\_id>/data/inputs . This directory SHOULD exist before running the workflow.

source\_path is the path to a directory where the PyCOMPSs job produces results and which is uploaded to the data catalogue by the DLS. Typically use some directory in your home, like /home/nct01/<user\_id>/data/results . This directory SHOULD exist before running the workflow.

num\_nodes is the number of compute nodes used to run the PyCOMPSs job. Use “2” for this input.

# Now as a Developer let's test our workflow



The screenshot shows the eFlows4HPC web interface. The top navigation bar includes 'Applications', 'Catalog', and 'Administration'. The main content area displays the 'myid\_minimal\_workflow' environment. A sidebar on the left contains icons for 'info', 'Runtime view', 'Workflow', 'Executions', and 'Logs'. The 'Workflow' section is active, showing a list of workflows. The 'exec\_job' workflow is selected, and its inputs are listed below. The 'Workflows' dropdown menu is highlighted with a red box.

Workflow
exec_job

Workflow Inputs
user_id
vault_id
oid
target_path
source_path
num_nodes

Select the “exec\_job” workflow and fill inputs:

user\_id & vault\_id are the account credential used to run the job just use the same values as just before deploying

oid is an object id in the data catalogue referencing the workflow input data. For today it should be 2c2463377aac4aa59381c6b06fe800f3

target\_path is the path to a directory where the DLS will store input data and will be used as input for the PyCOMPSs job. Typically use some directory in your home, like /home/nct01/<user\_id>/data/inputs . This directory SHOULD exist before running the workflow.

source\_path is the path to a directory where the PyCOMPSs job produces results and which is uploaded to the data catalogue by the DLS. Typically use some directory in your home, like /home/nct01/<user\_id>/data/results . This directory SHOULD exist before running the workflow.

num\_nodes is the number of compute nodes used to run the PyCOMPSs job. Use “2” for this input.

# Now as a Developer let's test our workflow

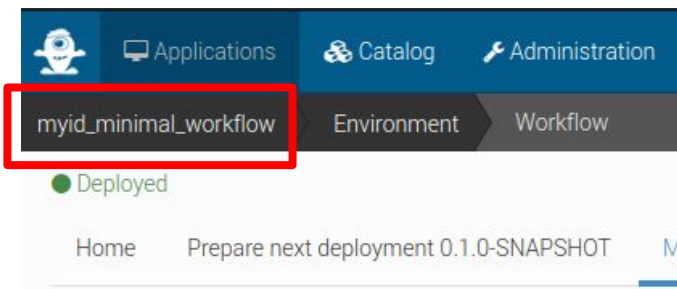
Finally, click on the “Launch” button to run the workflow.

You can follow the workflow execution exactly like we did for the workflow deployment (workflow & logs tabs).

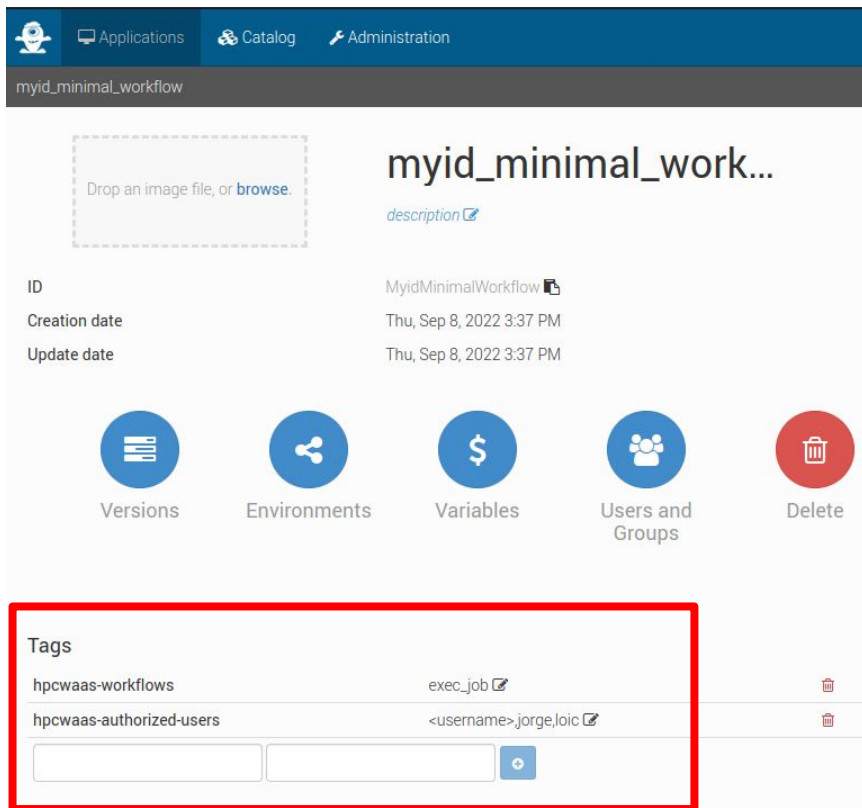
# Expose your workflow to the HPCWaaS API

Once you tested that everything works properly, you could choose to expose your workflow to end-users.

Go back to the main page of your application by clicking on its name on the top left corner



# Expose your workflow to the HPCWaaS API



The screenshot shows the configuration page for a workflow named 'myid\_minimal\_workflow'. The interface includes a navigation bar with 'Applications', 'Catalog', and 'Administration' tabs. Below the navigation bar, there is a description field with a 'Drop an image file, or browse.' prompt. The workflow details section shows the ID 'MyidMinimalWorkflow', creation date 'Thu, Sep 8, 2022 3:37 PM', and update date 'Thu, Sep 8, 2022 3:37 PM'. Below this, there are five circular icons representing 'Versions', 'Environments', 'Variables', 'Users and Groups', and 'Delete'. At the bottom, there is a 'Tags' section with a table of tags. The table has two rows: one for 'hpcwaas-workflows' with the value 'exec\_job' and one for 'hpcwaas-authorized-users' with the value '<username>jorge,loic'. Below the table, there are two empty input fields and a plus button to add more tags.

Tags	
hpcwaas-workflows	exec_job
hpcwaas-authorized-users	<username>jorge,loic

Use the tags section to configure the way the HPCWaaS API will interact with your application.

**hpcwaas-workflows** is a comma-separated list of workflows of your application that should be exposed to the API. In the case of the minimal workflow use "exec\_job" as value.

**hpcwaas-authorized-users** is a comma-separated list of users that are authorized to use this workflow. If this tag is not set every authenticated user can access your workflow. To keep things readable please specify at least your username here.



**Now we are done with the Developer role,  
let's switch to the end-user role**



## Interacting with the HPCWaaS API with the CLI

For the end-user the first thing to do is normally to generate an SSH keypair just like you did on slide 6.

You can do it again or just use the one you previously generated.



## Let list available workflows

```
./waas --api_url https://eflows4hpc.bsc.es/waas -u <user>:<password> workflows  
list
```

**As a result you will get various information about workflows. The most important one is the Workflow ID as it uniquely identifies a workflow and allow to trigger an execution.**

## Trigger an execution on a given workflow

```
./waas --api_url https://eflows4hpc.bsc.es/waas -u <user>:<password> workflows  
trigger -f -i user_id=<username> -i vault_id=<SSH_KEYPAIR_ID> -i  
oid=2c2463377aac4aa59381c6b06fe800f3 -i  
target_path=/home/nct01/<username>/data/inputs -i  
source_path=/home/nct01/<username>/data/results -i num_nodes=2 <workflow_id>
```

**Remember that target\_path & source\_path SHOULD EXIST before running the workflow.**

## Monitor a workflow an execution

The **-f** flag on the “trigger” command allows to follow the execution by regularly refreshing its status from the HPCWaaS API.

Another way to do it is to use the “execution status” command on the Execution ID returned by the “trigger” command.

```
./waas --api_url https://eflows4hpc.bsc.es/waas -u <user>:<password> executions  
status <Execution_ID>
```

“execution status” command also have its own **-f** flag

# Cancel a workflow an execution

Cancelling a running execution can be done using the “executions cancel” command

```
./waas --api_url https://eflows4hpc.bsc.es/waas -u <user>:<password> executions  
cancel <Execution_ID>
```

# That's all folks!



**eFlows4HPC**

Enabling dynamic and Intelligent workflows  
in the future EuroHPC ecosystem

[www.eFlows4HPC.eu](http://www.eFlows4HPC.eu)



@eFlows4HPC



eFlows4HPC Project



This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.