



ENVRI
FAIR



EOSC ENVRI DevOps framework -- Progress report

Zhiming Zhao¹, Andrea Manzi²

¹University of Amsterdam ENVRI-FAIR WP7 leader

²EGI foundation



ENVRI-FAIR has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824068



Time line

- Use case 1: workflow automation
- Use case 2: Jupyter notebook for data intensive science
- Use case 3: EOSC DevOps framework

Phase 1

Describe the key use case: scenario, scope, KPI, steps etc.

Get familiar with the EOSC services, following training and practices from the other projects etc.

Get the requested resource provisioned
Setup OLA with resource providers and agreement with Jelastic.

Phase 2

DevOps pipeline configured, including Git, automated testing, integration and **deployment demonstrate** in at least via two service development

Phase 3

Demonstrate the initial version of the workflow from ENVRI-FAIR, with automated workflow execution in Cloud;
Demonstrate the other common data services identified in the ENVRI communities (optional)

Phase 4

Exploit the results to the development activities in ENVRI-FAIR sub domain
Sustain the development by finding new opportunities, e.g., new EOSC projects etc.



Resource provisioning

1. EGI OLAs agreed with the resource providers (CESGA and INFN-CATANIA) and cloud resources provisioned
1. Installation of the Jelastic platform performed by Jelastic engineers on CESGA VMs
 - a. Admin interface: <https://jca.j.fedcloud.eu/>
 - b. User interface: <https://app.j.fedcloud.eu/>
1. Onedata Oneprovider for EGI DataHub deployed at CESGA with 10TB storage (ENVRI-FAIR space)
 - a. <https://oneprovider-envri.datahub.egi.eu>
1. EGI community Notebooks instance deployed at INFN-CATANIA K8s cluster (access via EGI Checkin)
 - a. <https://envri-notebooks.fedcloud-tf.fedcloud.eu>



Activities during the last phase

- Get familiar with Jelastic
- Integrate the tools developed in the previous phase
 - IaaS automation engine with OpenStack interface, and authentication interface
 - Work on the use case of ecology Lidar data processing

```
vr_ demo Last Checkpoint: a minute ago (unsaved changes)
File Edit View Insert Cell Kernel Help
Not Trusted Python 3.0

In [42]:
tosca = get_tosca(provisioned_tosca_id)

tosca_dict = yaml.safe_load(tosca)
graph = build_graph(tosca_dict['topology_template']['node_templates'])
nx.draw(graph, with_labels=True)

for node_name in toasca_dict['topology_template']['node_templates']:
    if toasca_dict['topology_template']['node_templates'][node_name]['type'] == 'tosca.nodes.OC.VM.Compute':
        print(node_name)
        print(tosca_dict['topology_template']['node_templates'][node_name]['attributes']['public_ip'])

/opt/conda/lib/python3.7/site-packages/urllib3/connectionpool.py:986: InsecureRequestWarning: Unverified HTTPS request is being made to host 'lifewatch.lab.uvalight.net'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
InsecureRequestWarning:
compute: 3.128.289.252
compute_1: 3.126.139.56
```

IaaS Planner | **Planner** | **Optimizer** | **Robin**

Parameter optimizer

1 Select application type

- Regular Workflow
- Regular Workflow**
- Time-Constrained Workflow
- Microservices
- IOT

2

3

4 Configure QoS demands

Select the workflow of your application

Insert workflow file in cwl format

0 files (0 B in total)

Select the performance models that you wish to compare

Insert performance models in yaml format

0 files (0 B in total)

COMPARE **CLEAR**

Lowest makespan	Lowest total costs + makespan
File name: input_pcp- _Copy.yaml	File name: input_pcp- _Copy.yaml
Total costs: 27	Total costs: 27
Makespan: 27	Makespan: 27



Running Jupyter in distributed cloud

- FAIR-Cells: Customize Jupyter environment
- Collecting use cases from ENVRI community

FAIR-Cells

About

FAIR-Cells is developed by the research team for Quality Critical Distributed Computing (QCDC) in the Multi Scale Networking System Group (MNS) of the University of Amsterdam. The development of FAIR-Cells is coordinated by Dr. Zhiming Zhao: z.zhao@uva.nl

Developers:
W. Kruijerwilco@uva.nl
S. Koulouzis@uva.nl

MNS UvA

FAIR-Cells

Build Run About

Build

Image name: nb_helper

Base image: Data Science Notebook

Cell: Cell 5

Preview:

line = [0.25, 1.5]
mean_line = 0.875

Conda environment.yml:

```
# Autogenerated by FAIR-Cells, please adjust.
channels:
- conda-forge
dependencies:
- pip:
- matplotlib
```

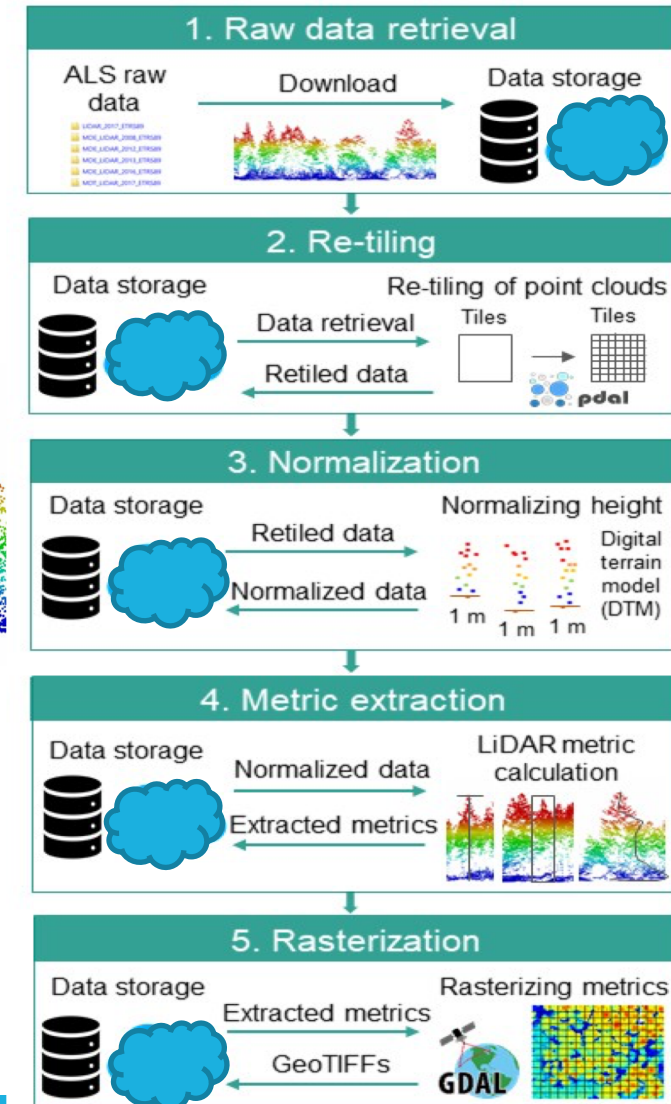
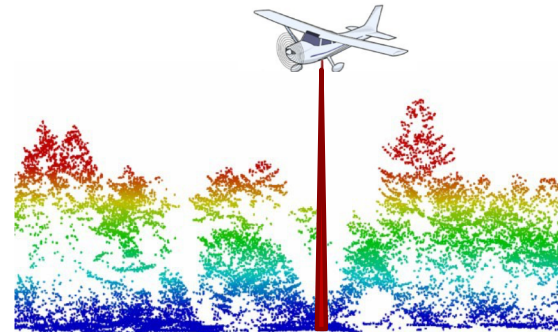
Variables:

Variable	Query	Post	Disabled
sum	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
len	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
line	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
mean_line	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
plt	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>



Processing Lidar data on Cloud

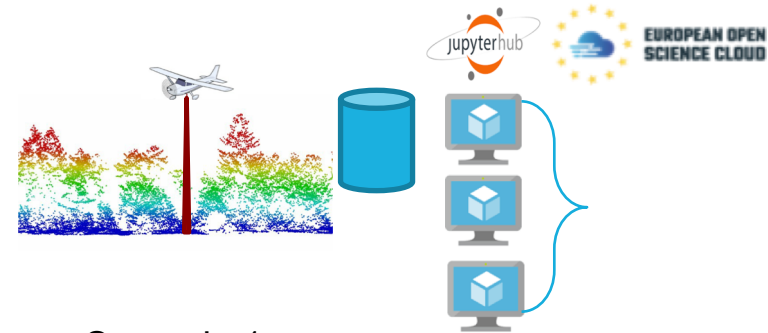
- Running the current Jupyter notebook with bigger data sources on larger infrastructure
- Laserchichek is a program to process the Lidar data via 5 steps
- The program will be executed on
 - Jupyter Hub, and
 - On distributed EOSC VMs using the FAIR –CELLs tool developed



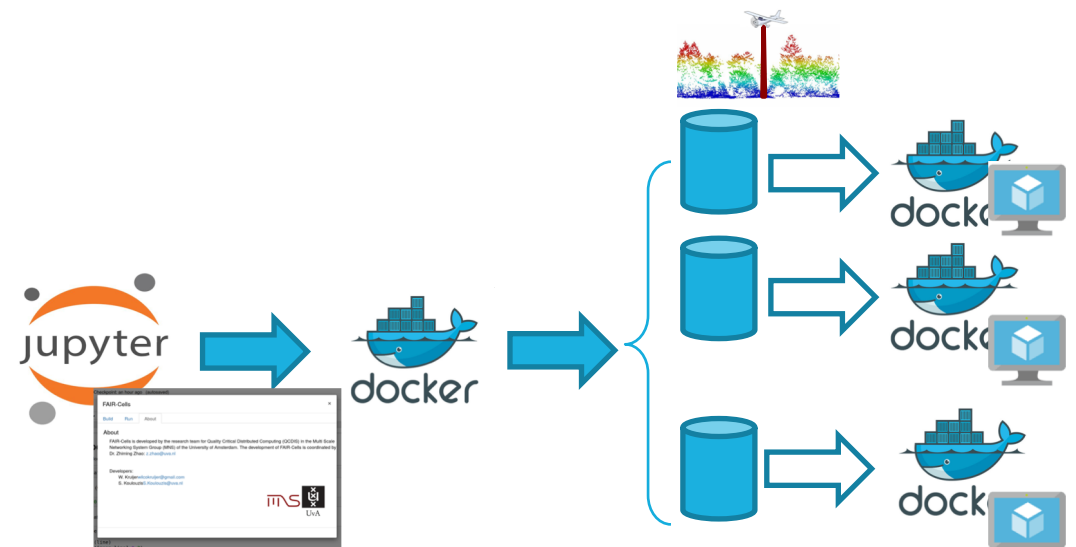


Demonstrator

- Scenario 1: running the laserchilcheck on the JupyterHub
- Scenario 2: containerizing the code using FAIR-CELLS, and then distributed containers on different VMs
- Submitted an abstract to the EGI conference.



Scenario 1



Scenario 2



DevOps pipeline comparison

Action:

1. Establish DevOps pipeline for a software we developed in ENVRI
2. Compare Jalestic and other DevOps options (Azure and Open source choices)
3. Create a practice report



Summary

Phase 1

Describe the key use case: scenario, scope, KPI, steps etc.

Get familiar with the EOSC services, following training and practices from the other projects etc.

Get the requested resource provisioned
Setup OLA with resource providers and agreement with Jelastic.

Phase 2

DevOps pipeline configured, including Git, automated testing, integration and deployment
demonstrate in at least via two service development

Phase 3

Demonstrate the initial version of the workflow from ENVRI-FAIR, with automated workflow execution in Cloud;
Demonstrate the other common data services identified in the ENVRI communities (optional)

Phase 4

Exploit the results to the development activities in ENVRI-FAIR sub domain
Sustain the development by finding new opportunities, e.g., new EOSC projects etc.



Next phase

- **Demonstrator: scaling LidarData notebook to large scale data and infrastructure (by Nov.)**
 - From notebook to REST services/containers
 - Describing distributed workflows
 - Automated the execution using EOSC IaaS
- **Demonstrator 2: DevOps frameworks demonstrator (by Dec).**
 - Demonstrator 3.1: demonstrate the automation of testing, integration, deployment
- **Exploitation to ENVRI-FAIR in the last three months.**



Sustainability

1. The use case is jointly with the LifeWatch ERIC VL innovation center; the output will be taken by LifeWatch ERIC to further develop
2. Exploitation of the results to the ENVRI communities (via the common development plan of RIs in ENVRI).
3. Getting support from other ongoing relevant projects, e.g. ARTICONF, BlueCloud and CLARIFY
4. Other opportunities



ENVRI
FAIR

EOSC Earlier Adopter Program

www.envri.eu



envri.eu/envri-fair



[@ENVRIcommunity](https://twitter.com/ENVRIcommunity)



[ENVRI community](https://www.linkedin.com/company/ENVRIcommunity)



facebook.com/ENVRIcommunity