



CCP: A Cloud Computing Platform for FAIR, Open Earth Science

Alfredo Oliviero ⁽¹⁾, Marco Lettere ⁽²⁾, Andrea Dell'Amico ⁽¹⁾, Pasquale Pagano ⁽¹⁾

1. Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" - Consiglio Nazionale delle Ricerche, Pisa, Italia 2. Nubisware S.r.l.



Wien, 01/05/2025

The Cloud Computing Platform



A platform for running scientific methods in a scalable, cloud-based environment (VREs)

> Provides on-demand, scalable computing through a web interface or REST APIs

Supports running methods imported by users, available in the platform, or shared by colleagues

Enables collaboration by sharing data, tools, and results in the same environment

Built on containerization and JSON-based APIs for flexibility and seamless integration



What is D4Science





A digital infrastructure for collaborative scientific research, combining over 500 software components into a coherent and centrally managed system of hardware, software, and data resources

D4Science is owned and managed by CNR

 Pisa
 1.624

 • Dec. '22
 ●

 6.976
 ●

 • June '23
 ●

- Federated digital infrastructure:
- 4 sites (1 Pisa, 3 @ GARR)
- 11.100 CPUs core
- 29 TB RAM
- 1.400 TB Storage

It supports ESFRI RIs, national and European projects, national and international initiatives

Virtual Research Environments (VREs)





- Secure, web-based platforms tailored to scientific workflows
- Integrate both local resources and external services (e.g. cloud storage, distributed computing)
- All accessible via a unified, ondemand workspace in the browser
- Enable researchers to:
 - Organize and store data
 - Run and monitor computational methods
 - Share tools and results
 - Collaborate across teams and disciplines

VRE - Collaborative Data Analysis



CCP: A Cloud Computing Platform for FAIR, Open Earth Science by D4Science – EGU25 Wien - 01/05/2005

5

Sharing is

encouraged



What is CCP

<u>Method import</u> <u>support</u> via a dedicated tool supporting any prog. language

"out-of-the-box" methods **as-a-Service** and automatic code generation for Python, R, Julia, Jupyter Per-method Web UI and REST API access via <u>Standard</u> (OGC API – Processes)

> Provenance management



Defining Methods

SimpleImageCla A simple image clas

D4Science infrastrue Inputs
Runtime ?
nubisware/simpi
Annotations for ex

> Annotations for Input picture ? https://as1.ftcdn

Outputs

Input imag



Method Implementation: Script, Algorithm, library



Importing phase





Resulting Web UI & REST API

er with parametrizable url to input picture compatible with the re	Execution Monitor
	Search
	SimpleImageClassifier ImageClassifier
classifier:latest	T 10.1 running 🖨 P 🗊 Re-submit 🚺
,	Accepted 05/03/2024 @ 13:54:48. Last update 05/03/2024 @ 13:58:33: Execution completed: Initialization
n	completed Difference executivation interactionality multiwarehimpleimanelisatifier latest
(jpg/00/85/32/68/1000_F_85326806_k3nKFIDnL78KZZpgplit	<pre>init - :# init is the appendix is if if it is it</pre>
Curbor mago	
Generate code for:	1.0.1 running Compared 04/02/2024 @ 15:10:26
Python 3	Last update 01/03/2024 @ 15:22:33: Execution completed: Initialization



Method Executions



Web UI / Rest API execution

StorageHub/Email results



Use Case: Geospatial Processing



ITINERIS_Training-Platform ✿ Administration 중 ITINERIS_Training-Platform	tics Engine 📀 🛱 JupyterLab RStudio 💿 🧾 Catalogue 🚢 Merr	
vtics Engine (CCP)	Method execution form	
your Methods/Algorithms on the Cloud Learn more	GDAL - Reproject Clip Resample Raster TIFF - Workspace	
ods List	 Reproject a raster file to a different spatial reference system (SRS) (if indicated) optionally clip the file to a selectable bounding box (if selected) 	
ethods C	optionally resample to a specific reso Inputs	
Search ▼GDAL Geospatial 05€	 Runtime The image of the runtime to use for method execution. This depends on the infrastructure specific protocol for interacting with registries. 	
GDAL - Raster TIFF Details - Workspace 1.0. Alfredo Oliviero	hub.dev.d4science.org/test/python-gdal:latest	
show details of a raster file reads the file from workspace	Raster tiff Public url of the raster tiff from workspace	
(bython) (gdal) (raster) (tiff) D4Science production Infrastructure	Raster tiff	
GDAL - Reproject Raster TIFF - Workspace 1.0.1 Attrade Oliviero Reproject a raster file to a different spatial reference system (SRS) read the file from workspace	SRS target Target spatial reference system in EPSG format examples: 3003, 4326, etc SRS target	
(python) gdal (raster) tiff) D4Science production infrastructure GDAL - Clin Raster TIFE to boundingbox - Workspace 500	bounding box for clipping (optionally) select a rectangle area from the map, the selected bounding box has to be inside the area of the tiff image please select "WKT" format	
Alfredo Oliviero Cilp a raster file to a bounding box selectable from the map	bounding box for clipping	
(python) (gdal) (raster) (tiff) D4Science production Infrastructure		
GDAL - Resample Raster TIFF - Workspace 10.0 Alfredo Oliviero	Resample resolution	
► Resample a raster file to a to a specific resolution	(optional) resample resolution. used for x and y example: 20	
(python) (gdal) (raster) (tiff) D4Science production Infrastructure	Resample resolution	

- A Python-based Docker image with tools like GDAL and the Copernicus Climate Data Store (CDS) CDSAPI is provided
- Users can extend it or build their own containers.
- Example methods:
 - Raster handling: extract metadata, convert formats, reproject raster
 - Terrain analysis: compute slope and aspect
 - Vector operations: buffering, spatial joins, overlays
 - Advanced workflows: spatial interpolation, network analysis

Use Case: Modeling and Simulation







- CCP supports CPU-intensive simulations within asynchronous, containerized workflows.
- Compatible with models implemented in:
 - Python, R, Java, Fortran, C/C++, and other scripting languages
 - Any containerizable framework or simulation engine •
- Example workflows:
 - Groundwater flow and contaminant transport
 - Slope stability and landslide modeling •
 - Seismic hazard and crustal deformation •
 - Basin evolution and stratigraphic modeling •
- Execution is reproducible and scalable across different infrastructures.

FAIR Principles



CCP is designed for Open Science and it fully aligns with the FAIR principles:



Provenance is tracked using standards like **PROV-O**, ensuring full documentation and reproducibility.



Why It Matters for scientists

Reproducibility across methods and data

Access to significant computational resources

Simplified execution without technical overhead

Integration into a robust infrastructure supported by CNR, widely adopted in European research Driver for Open Science, making outputs transparent, traceable, and shareable

Sharing is encouraged

Conclusion



CCP brings scalable computation into your research, without the burden of system configuration

Define once, run anywhere: from quick tests to complex research

Designed for **Open Science**, built for **reproducibility**, and ready for **collaboration**

Already supporting major initiatives like **Blue-Cloud** and **ITINERIS**



Thank you

Any questions?



https://www.d4science.org/



