

Advancing Earth Science

EO Apps in Exploitation Platforms

EOEPCA Demonstration

- million and a second second



TERRA)UE

	Advancing Earth Science	
● ● ● ● OGC Testbe	d=13: EP Applicatii x + ecure docs.opengeospatial.org/per/17-023.html 💐 🛠 🖉 Q & ≕ 🔘 :	
OGC Tes	tbed-13: EP Application Package ER	
$p_{\rm u} \rightarrow c_{\rm o}$	Testbed-13: Application: x + Not Secure docs.opengeospatial.org/per/17-024.html	
ef Execut	tion Service ER	
at Pul 8	OGC Testibed-14: Application : × + O Not Secure docs.opengeospatial.org/per/18-049r1.html	
Ref OG Ref Eng	GC Testbed-14: Application Package gineering Report	
VA Edi App		陶 ☆ 🕖
el OG Refe COI Refe COF Cate httl Edite WA Title	OGC Earth Observations Applications Pilot: Terradue Engineering Report	
ОС	Publication Date: 2020-10-22 Approval Date: 2020-09-23	
WAI This OGC is su	Submission Date: 2020-09-03 Reference number of this document: OGC 20-042 Reference URL for this document: <u>http://www.opengis.net/doc/PER/EOAppsP</u> J	<u>ilot-Terradue</u>
	Category: OGC Public Engineering Report Editor: Pedro Gonçalves	
	Title: OGC Earth Observations Applications Pilot: Terradue Engineering Repo	ırt
	OGC Public Engineering Report	

A 0 0

Background

- Previous OGC Testbeds initiated the design of an application package for Earth Observation **Applications in distributed Cloud Platforms**
- It provides information about the software item, metadata and dependencies
- Deployed and executed within an Exploitation Platform in a service compliant with the OGC **API Process specification.**
- Active development on OGC Testbeds and ESA project: https://github.com/EOEPCA



Advancing Earth Science

Alice

- EO data stage-in
- Application development/build/test with staged data
- Eric
 - Application package deployment
 - Application execution, monitoring and results

Alice is an EO data scientist tasked to create an application that detects lava flows in eruptive events.





The EO data stage-in is an Operation that

takes as input:

- a reference to a catalog entry (e.g. Opensearch, STAC Item)
 produces:
- a local STAC Catalog with a STAC Item whose Assets have an accessible href (either local or remote e.g. COG)

returns:

• the path to a local folder containing the STAC Catalog file (catalog.json)

The staged local STAC Catalog is the input manifest for the application.



Alice uses a CWL document with two steps workflow:

- a stage-in step that invokes a stage-in tool in a Docker image
- a stage-out step that invokes a stage-out tool in a Docker image

CWL:

- Two tools to stage the EO data: docker and a CWL runner (e.g. cwltool)
- No other tools required for stage-in

STAC as input manifest:

- Easy access to EO metadata
- Easy access to asset location
- Easy access to EO data via common band names (e.g. red, nir, swir16)



Alice is an EO data scientist tasked to create an application to detect lava flows using Sentinel-2 Level 1C data.

The application:

- takes one Sentinel-2 Level 1C data
- accesses the nir, swir16 and swir22 bands
- generates two normalized hotspot indexes
- generates two bitmasks based on a defined threshold

The application generates a local STAC catalog as the output manifest describing the results metadata and assets' location.



Alice includes in her project:

- a **YAML file** describing the Python environment required to run the application (conda environment.yml)
- a **Dockerfile** that:
 - configures that python environment
 - installs the application
 - generates a docker image that "knows" how to run the application
- a **CWL document** that:
 - references the docker image
 - knows how to invoke the application in the docker





STAC

- STAC local catalog as input manifest for applications
- STAC local catalog as output manifest describing the results
- The application supports implicitly EO data from different sensors and missions

Environment manifest

- Application has an environment manifest

Container image

- Application has a Dockerfile to create a docker image

Application Package

- Application is packaged as an CWL document

Application development in a PDE

Advancing Earth Science

Terra)Ue



Alice uses a PDE to:

- access test EO data
- implement, test and package her application

The PDE provides:

- JupyterLab
- Theia Cloud IDE
- Jenkins CI/CD
- Local S3 Object Storage
- Access to the Workspace



Application Package Exploitation

Eric takes Alice's Application Package then, using the OGC API - Process,

- Deploys it on an ADES
- Executes and monitors a job
- Gets the results.
- The Application Package is a CWL document :
 - that Alice uses to test on application
 - that Eric deploys on an Exploitation Platform
- A single Application can yield several Application Packages that expose parameters and inputs in different flavours and execution patterns.

TERRAJUE Take away messages about the process

- Advancing Earth Science
- Build/test with staged data using docker and CWL
 - Develop an Application Package and test it as an Application Package
- Scripted execution using the Application Package
 - The Application Package can be executed locally
- Deployment, execution, monitoring and results with the OGC API Process.

OGC 20-089 OGC Best Practice for Earth Observation Application Package https://gitlab.ogc.org/ogc/eoap-best-practice



Advancing Earth Science



Questions?