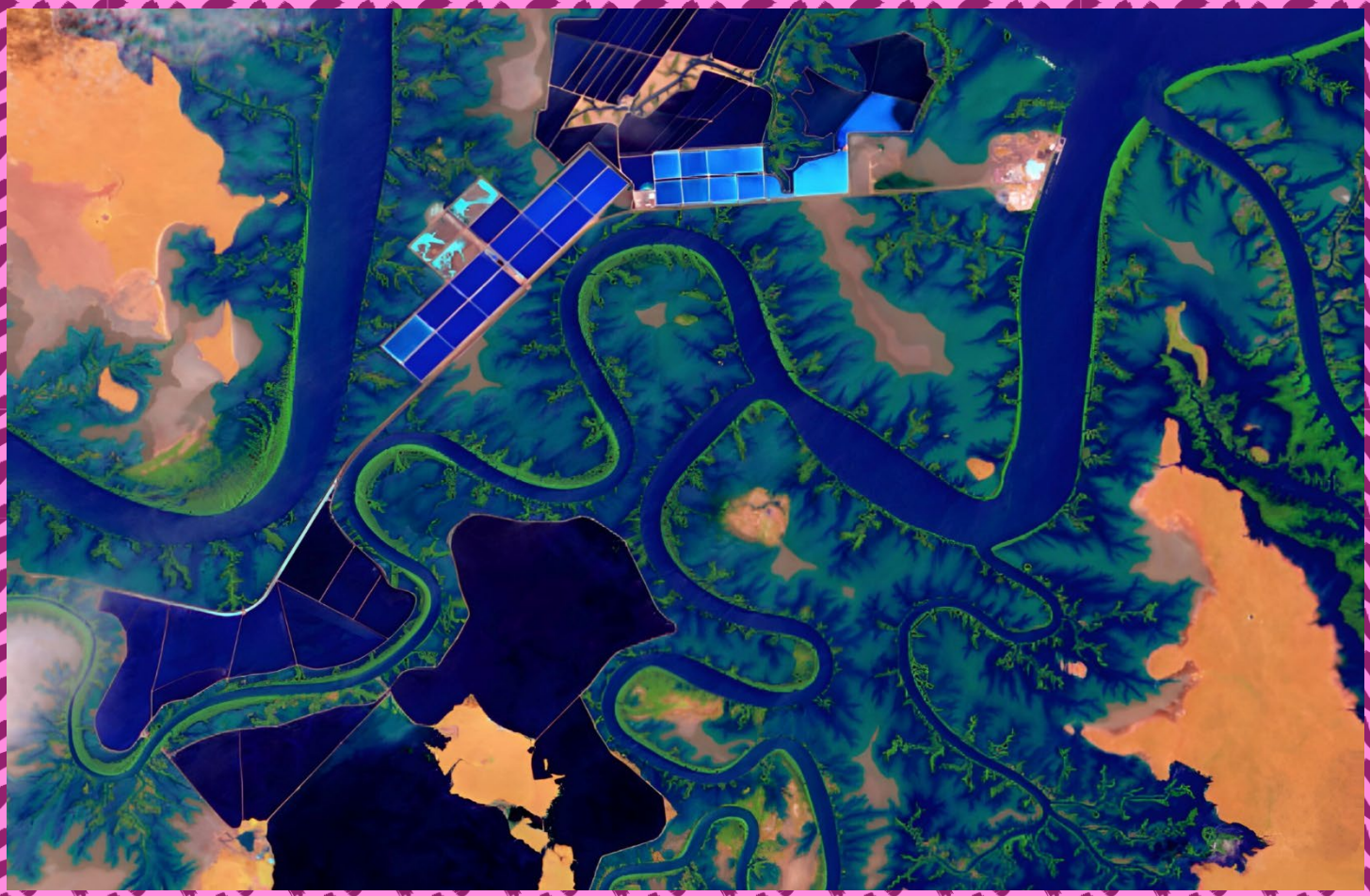
CGIAR
SCIENCE FOR A FOOD-SECURE FUTURE

Report

Evolving the Limpopo Digital Twin Open Data Cube: From Legacy Catalogs to Spatiotemporal Asset Catalog-Native, User Managed and Containerized Workflows

Abdul Afham, Zolo Kiala, Surajit Ghosh and Mariangel Garcia Andarcia

December 2025



Authors

Abdul Afham, Junior Data Engineer, Consultant, International Water Management Institute (IWMI), Colombo, Sri Lanka
Zolo Kiala, Postdoctoral Fellow – Water Data Science, IWMI, South Africa
Surajit Ghosh, Regional Researcher – Water Data Science, IWMI, Colombo, Sri Lanka
Mariangel Garcia Andarcia, Research Group Leader - Water Futures Data & Analytics (WFDA), IWMI, Colombo, Sri Lanka

Acknowledgments

This work was conducted as part of the CGIAR Accelerator for Digital Transformation. We would like to thank all funders who supported this research through their contributions to the CGIAR Trust Fund (www.cgiar.org/funders).

This study was funded by the CGIAR Initiative on Digital Innovation, which advances sustainable agrifood systems through digital solutions and the International Water Management Institute's Digital Innovations for Water Secure Africa (DIWASA) project. We also wish to thank the Leona M. and Harry B. Helmsley Charitable Trust for their financial support for the DIWASA project and the Limpopo Watercourse Commission (LIMCOM) for their support to this project.

Citation

Afham, A.; Kiala, Z.; Ghosh, S.; Garcia Andarcia, M. 2025. *Evolving the Limpopo Digital Twin Open Data Cube: from legacy catalogs to spatiotemporal asset catalog-native, user managed and containerized workflows.* Colombo, Sri Lanka: International Water Management Institute (IWMI). CGIAR Accelerator for Digital Transformation. 31p.

Front cover photo: Aerial image (*photo*: Geoscience Australia)
Back cover photo: Aerial survey map(*photo*: apemltd)

Disclaimer

This publication has been prepared as an output of the CGIAR Accelerator for Digital Transformation and has not been independently peer reviewed. Responsibility for editing, proofreading, and layout, opinions expressed, and any possible errors lies with the authors and not the institutions involved. been independently peer reviewed.

Contents

List of Figures

List of Tables

Executive Summary

The Limpopo River Basin, shared by Botswana, Mozambique, South Africa, and Zimbabwe, requires a robust, interoperable data infrastructure to support the Limpopo Digital Twin and inform water- and irrigation-related decision-making. This report covers recent updates to the Limpopo Digital Twin Open Data Cube (ODC) implementation, concentrating on modernizing its metadata, management tools, and deployment architecture. We first position ODC within the emerging SpatioTemporal Asset Catalog (STAC) ecosystem, recognizing that while STAC increasingly replaces ODC's original cataloging role, ODC remains useful as part of a broader cloud-native stack. We then describe two core technical changes: a shift from EO3 to STAC Item metadata and a metadata-only workflow that avoids raster duplication, reducing processing time and storage overhead. In parallel, the new ODC Manager web application enables authenticated users to create, update, index, and delete products without developer intervention. Finally, we outline the migration from a manually managed AWS setup to a fully containerized Docker deployment, improving reproducibility, scalability, and maintainability of the Limpopo Digital Twin data infrastructure.

Keywords: Open Data Cube (ODC); SpatioTemporal Asset Catalog (STAC); Earth observation; metadata management; ODC Manager; containerization.

Introduction

The Limpopo River Basin, spanning four countries, such as Botswana, South Africa, Mozambique, and Zimbabwe, faces significant challenges related to water scarcity and climate change, which in turn affect irrigation practices and increase the risk of drought across the region (Mafuta et al., 2021). To better manage these resources and support forward-looking planning, existing data must be carefully analyzed. A crucial first step is to ensure that the data are catalogued and organized in a way that makes them easy to curate and reuse. The datasets discussed in this report are raster datasets: grids of pixels or cells that hold spatial and temporal information, along with additional geospatial metadata such as geometry and projection. For the Limpopo Digital Twin data catalog, which contains many raster datasets, the Open Data Cube (ODC) was selected as the core framework.

This document provides an update to the first ODC technical report (Afham et al., 2024). It details several major changes made to the existing workflow to improve the system, including making it easier for users to manage their products, defining a smoother deployment strategy using Docker, and introducing ODC-specific refinements in the metadata creation process that significantly enhance the usability and transparency of the system.

In brief, ODC is a software framework that allows users to catalog raster data, provides tools to explore these data via a dashboard-style user interface (Explorer), and exposes Web Map Services (WMS) for raster visualization. At its core, ODC relies on a PostgreSQL database that is managed through data cube command-line utilities.

The updates made to the system include:

- Addition of new products and expansion of datasets for existing products
- Changes in the metadata creation process
- Introduction of the ODC Manager application
- A revised deployment strategy

The current state of our ODC index is summarized in Table 1.

Table 1 Current raster products indexed in the Limpopo Digital Twin Open Data Cube

Product	Description
vegetation_condition_index_limpopo	Monthly Vegetation Condition Index (VCI) calculated using MODIS data to visualize drought extent, produced by IWMI under the Digital Twin project (Ghosh et al., 2024).
eflow_warnings_limpopo	Rasterized average monthly e-flow warnings, derived from SWAT-simulated discharges, implemented by IWMI as part of the LIMCOM Digital Twin project.

incremental_channel_contributions_limpopo	Rasterized average monthly incremental channel contributions, simulated by SWAT and implemented by IWMI under the LIMCOM Digital Twin project.
irrigated_areas_limpopo	Monthly irrigated area extent map produced by IWMI under the Digital Twin project (Kiala and Karthikeyan, 2024).
land_use_limpopo	Land use map of the Limpopo Basin.
soil_map_limpopo	Soil map of the Limpopo Basin.
dem_limpopo	Digital Elevation Model (DEM) of the Limpopo Basin.
et_monthly_limpopo	Monthly Actual Evapotranspiration and Interception for the Limpopo River Basin
blue_et_monthly_africa	Monthly Incremental Evapotranspiration (ET) for Africa from the Water Accounting Plus framework.
Green_et_africa	Green ET for Africa from the Water Accounting Plus framework.

The two major updates to the metadata creation pipeline are:

a. Metadata-only processing

The process creates only metadata for each dataset; no raster data are copied or duplicated into our AWS S3 (Simple Storage Service) bucket. This reduces the processing time and avoids unnecessary storage overhead.

b. Migration to SpatioTemporal Asset Catalog (STAC)

The metadata format has been switched from EO3 (https://pypi.org/project/eodatasets3/) to a STAC Item format, making the datasets more interoperable and easier to integrate with external tools and platforms.

The ODC Manager application (https://odc-manager.iwmi.org/odc_manager) was created to enable users to index their own products without needing to contact the system developers.

It is an easy-to-use application that supports four main operations: Create, Update, Index, and Delete.

Regarding deployment, the previous system ran ODC, and all related applications directly on an Amazon Web Services (AWS) server. This has now been switched to a containerized setup using Docker, making it much easier to replicate the system on another server if the need arises.

The remainder of this document is structured as follows. Section 2 provides background on the ODC and related initiatives, situating the Limpopo implementation within the broader EO data landscape. Section 3 examines the role of ODC in the STAC era, comparing core concepts and illustrating alternative, cloud-native access patterns. Section 4 describes the methodology underlying the recent updates, including the revised metadata pipeline, the ODC Manager application, and the containerized deployment. Section 5 presents key system outputs and the ODC Manager user interface. The report then concludes with a brief synthesis and outlook, followed by an annex that provides a full example of a STAC Item generated for the Limpopo Digital Twin.

Background on ODC and Related Initiatives

The ODC was established to provide users with a free and open data architecture for exploiting Earth observation (EO) data, addressing the challenges associated with processing the rapidly growing volume of satellite imagery. It is based on the Australian Geoscience Data Cube (AGDC) initiative (Lewis et al., 2017).

A related effort, the Committee on Earth Observation Satellites (CEOS) ODC, focuses on providing Analysis Ready Data (ARD) products to improve access to, and analysis of, satellite data. In early 2017, CDC set a goal of supporting operational data cubes in 20 countries by 2022. After the first 18 months of implementation, four operational data cubes had been established in Australia, Colombia, Switzerland, and Taiwan, with an additional eleven data cubes under development in Georgia, Moldova, Uganda, the United Kingdom, Vietnam, China, Kenya, Tanzania, Ghana, Senegal, and Sierra Leone (Kopp et al., 2019).

Major current implementations of ODC include:

- Digital Earth Australia (https://www.ga.gov.au/scientific-topics/dea) – continental-scale implementation for Australia.
- Digital Earth Africa (https://digitalearthafrica.org/en_za/) – continental-scale implementation for Africa.
- Swiss Data Cube (https://www.swissdatacube.org/) – Earth observation Analysis Ready Data for Switzerland.
- Brazil Data Cube (https://data.inpe.br/bdc/en/home-page-2/) – covering the entire Brazilian territory.

Open Data Cube (ODC) in the STAC Era

Building on the background above, this section reviews the core functions of the ODC and examines whether, in light of newer SpatioTemporal Asset Catalog (STAC)-based technologies (https://stacspec.org/en/), its original architecture remains the most appropriate choice for cataloguing and serving Earth Observation (EO) data.

Historically, accessing EO data was extremely difficult. Much of it was stored on magnetic tapes, and a single quarter-scene could cost more than 1,000 USD. As these archives were gradually digitized onto spinning disks and Analysis Ready Data (Levels 1 and 2) became

available, Geoscience Australia developed the ODC as an infrastructure layer to help users work more efficiently with this growing volume of data (Kopp et al., 2019).

The ODC has two primary functions. First, it creates and maintains a catalog of data scenes in a PostgreSQL database. Second, it provides a Python API that queries this index and returns the results as an xarray dataset. Xarray (https://docs.xarray.dev/en/stable/) offers a high-level abstraction over raw NumPy arrays, which can be cumbersome to handle directly; by contrast, xarray is more intuitive, concise, and less error-prone for multidimensional geospatial data.

However, working with an ODC index requires a direct PostgreSQL connection, which can be inconvenient in practice, as users need database read credentials simply to access the data. This is where STAC comes in. Introduced in 2017, STAC provides a simple, cloud-friendly standard for describing and discovering geospatial data. Instead of relying on a custom PostgreSQL index, data can be exposed through lightweight JSON catalogs and APIs, making it much easier to share, search, and integrate across different platforms and services. Table 2 summarizes how the core entities in ODC correspond to their STAC equivalents.

Table 2 Conceptual mapping between core Open Data Cube (ODC) entities and their SpatioTemporal Asset Catalog (STAC) equivalents.

ODC	STAC
ODC database	Catalog
Product	Collection
Dataset	Item

The STAC API is a RESTful interface, widely used for its simplicity and flexibility when searching and accessing geospatial assets.

Below is a short example of how data are accessed using the ODC Python API:

```
from datacube import Datacube
# Connect to a running ODC instance (requires Postgres + indexed data)
dc = Datacube(app="odc_example")
# Load Sentinel-2 data for a region and time
ds = dc.load(
    product="s2_l2a",
    x=(149.0, 149.1),    # longitude bounds
    y=(-35.3, -35.2),    # latitude bounds
    time=("2020-01-01", "2020-01-31"),
    measurements=["red", "green", "blue"]
)
print(ds)
```

By contrast, accessing the same type of data through a STAC API does not require a direct database connection. The example below shows how a public STAC endpoint can be

queried to discover Sentinel-2 items and load an individual asset directly into an xarray-compatible object using rioxarray.

```
from pystac_client import Client
import xarray as xr
import rioxarray
# Connect to a public STAC API (no Postgres, just JSON API)
catalog = Client.open("https://earth-search.aws.element84.com/v1")
# Search Sentinel-2 items in the same region & time
search = catalog.search(
    collections=["sentinel-2-l2a"],
    bbox=[149.0, -35.3, 149.1, -35.2],
    datetime="2020-01-01/2020-01-31"
)
items = search.get_all_items()
# Grab the first item asset (e.g. red band) and open as xarray
asset_href = items[0].assets["B04"].href  # Red band
ds = rioxarray.open_rasterio(asset_href, masked=True)
print(ds)
```

For the sole purpose of cataloguing data, the ODC is increasingly being superseded by newer, more interoperable solutions such as STAC. This view is echoed by Alex Leith, a prominent contributor in the ODC community, in his talk “The Death of ODC and the Transition to STAC” at the Cloud Native Geospatial Conference 2025 (available at: https://youtu.be/WIeNwMhNWfk?si=wZs5_J4uvvX3zhSI).

While STAC can effectively replace ODC’s cataloging function, cataloging is not the only service offered within the ODC ecosystem. The ODC Explorer service, for example, is a dashboard-style user interface used to browse and inspect data stored in an ODC index; this role can now be fulfilled by tools such as STAC Browser (https://radiantearth.github.io/stac-browser/#/?.language=en) or other custom front ends built on top of STAC APIs.

Similarly, the OWS service, which exposes data indexed in an ODC via standard OGC web services (WMS, WMTS, WCS), can be substituted with modern, cloud-native tools such as TiTiler (https://developmentseed.org/titiler/), which can serve imagery directly from STAC assets. Taken together, these STAC-based tools and services can replace much of the traditional ODC stack, offering a more modular and modern approach to handling geospatial Earth observation data.

In this context, our work does not discard ODC outright, but instead adapts it: we retain ODC where it still adds value, while aligning our implementation with STAC and cloud-native patterns for metadata, access, and visualization. The next section therefore describes the methodology behind these adaptations, including the revised metadata pipeline, the ODC Manager application, and the containerized deployment architecture.

Methodology

This chapter describes the methodology underlying the latest developments in our ODC implementation.

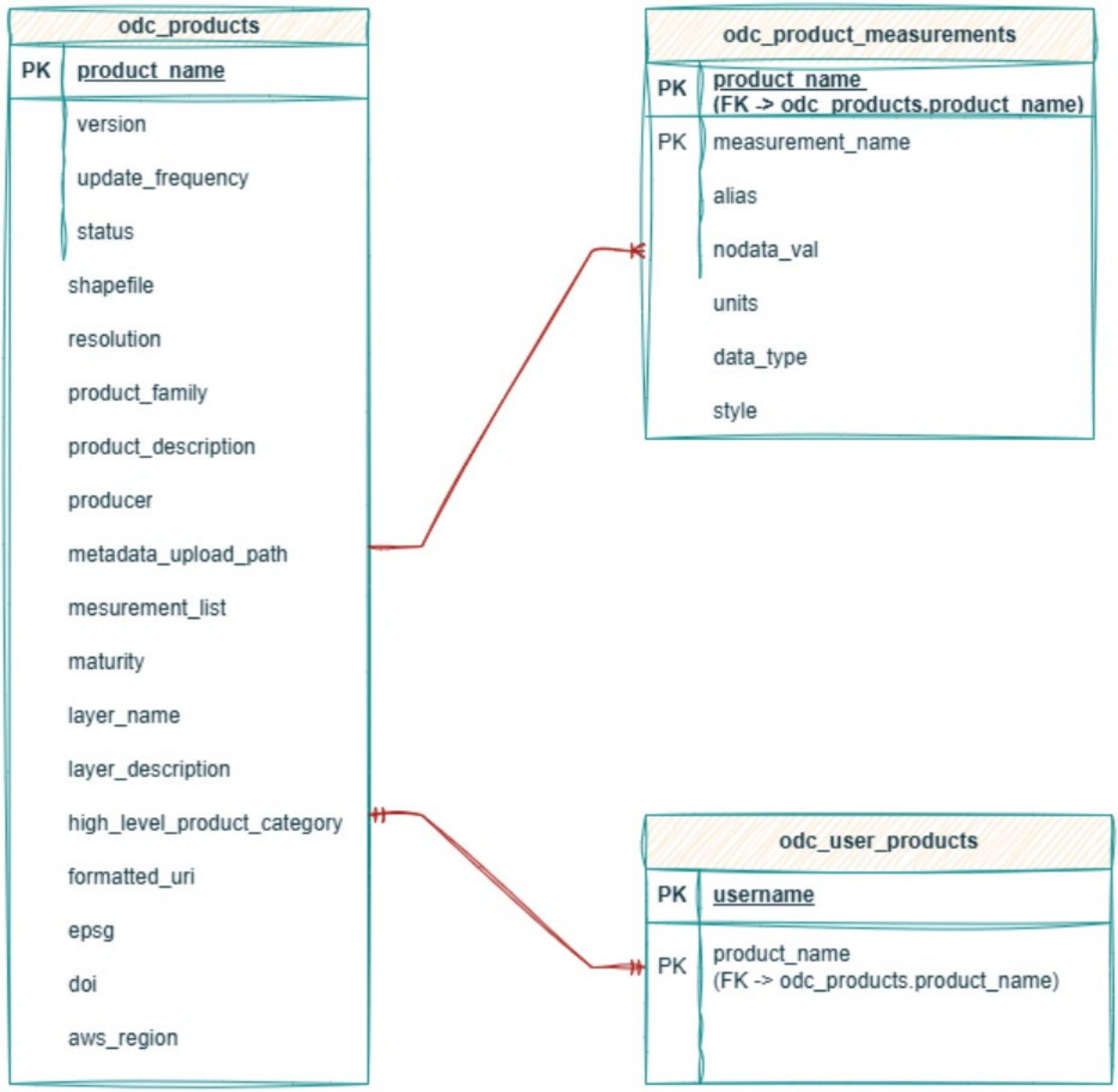
Metadata development

The previous metadata format used in our system was EO3, which is the recommended format for ODC metadata. However, with many geospatial metadata implementations now moving toward the SpatioTemporal Asset Catalog (STAC) standard, we adopted STAC for our system as well. This transition makes our datasets more interoperable and easier for users to integrate with other tools and platforms that support STAC.

A second major change in the metadata pipeline was to avoid duplicating raster datasets during metadata creation. In the updated workflow, only a stac-item.json file is generated in the output folder; no additional copy of the dataset itself is created. This significantly reduces the runtime of the metadata creation pipeline, as the previous processing step that rewrote the dataset has been removed.

ODC Manager application

To streamline this process, we developed the ODC Manager application with a dedicated product-level metadata database and updated the codebase to retrieve product fields directly from it. Figure 1 shows how product-level metadata, measurement definitions, and user–product permissions are stored and linked in the ODC Manager database.



1

Figure 1 Entity–relationship diagram for the ODC Manager catalog database, illustrating the odc_products table (core product metadata), the odc_product_measurements table (per-measurement attributes), and the odc_user_products table (mapping users to the product. [Source: Authors]

Building on this foundation, the ODC Manager now provides an easy-to-use interface that supports four core operations that users need to manage their products:

- Create new product definitions
- Update existing product configurations
- Index datasets into the ODC

1

- Delete products or entries when they are no longer needed

Together, these changes reduce developer overhead, empower end users to manage their own products, and make the ODC implementation more maintainable and scalable.

The Create function contains all the fields required by ODC to register a new product. These fields are grouped into three categories:

1. Product Information includes

- Product name
- Description
- Producer
- Dataset maturity
- Product version
- Product family
- High-level product category
- Layer name
- Layer description
- Digital Object Identifier (DOI)
- Dataset update frequency
- Region (shapefile)

2. Dataset Path includes

- Dataset URI
- Dataset formatted URI

3. Measurement Information includes

- Measurement name
- Alias
- NoData value
- Units
- Data type
- Style

The Update function is intentionally restricted compared to the Create function, allowing users to modify only selected fields of an existing product. These include product description, product version, layer name, layer description, DOI, formatted URI, NoData value, data type, alias, and style.

The Index operation is connected to the metadata creation pipeline, allowing users to specify the dataset fields for the datasets they wish to index into ODC.

The Delete function allows users to remove either specific datasets from the ODC index (for example, by specifying dates) or, if required, to delete an entire product.

From an implementation perspective, the frontend of the application was developed using Vite (https://vite.dev/), while Flask (https://flask.palletsprojects.com/en/stable/) was used as the backend framework to build the APIs for each of these functionalities, linking the frontend, the external ODC metadata database, and the ODC PostgreSQL database. Keycloak (https://www.keycloak.org/) is used as the authentication system for the platform, and role-based access control was implemented for product management.

Deployment

In the initial deployment, all applications were run directly on a single AWS server, including the ODC-managed PostgreSQL database, the Explorer application, and the OWS application. The Explorer and OWS services were kept alive using the screen command, which allowed them to continue running after the terminal session was closed. However, this approach was not easily reproducible or scalable. To address these limitations, we adopted a containerized architecture using Docker.

The resulting container layout is shown in Figure 2, which illustrates the ODC-managed PostgreSQL database, the Explorer and Datacube OWS applications, the ODC Manager frontend and backend with their MySQL catalog database, and the Airflow service used to orchestrate the creation of three products: irrigated areas, vegetation indices, and ET Limpopo datasets.

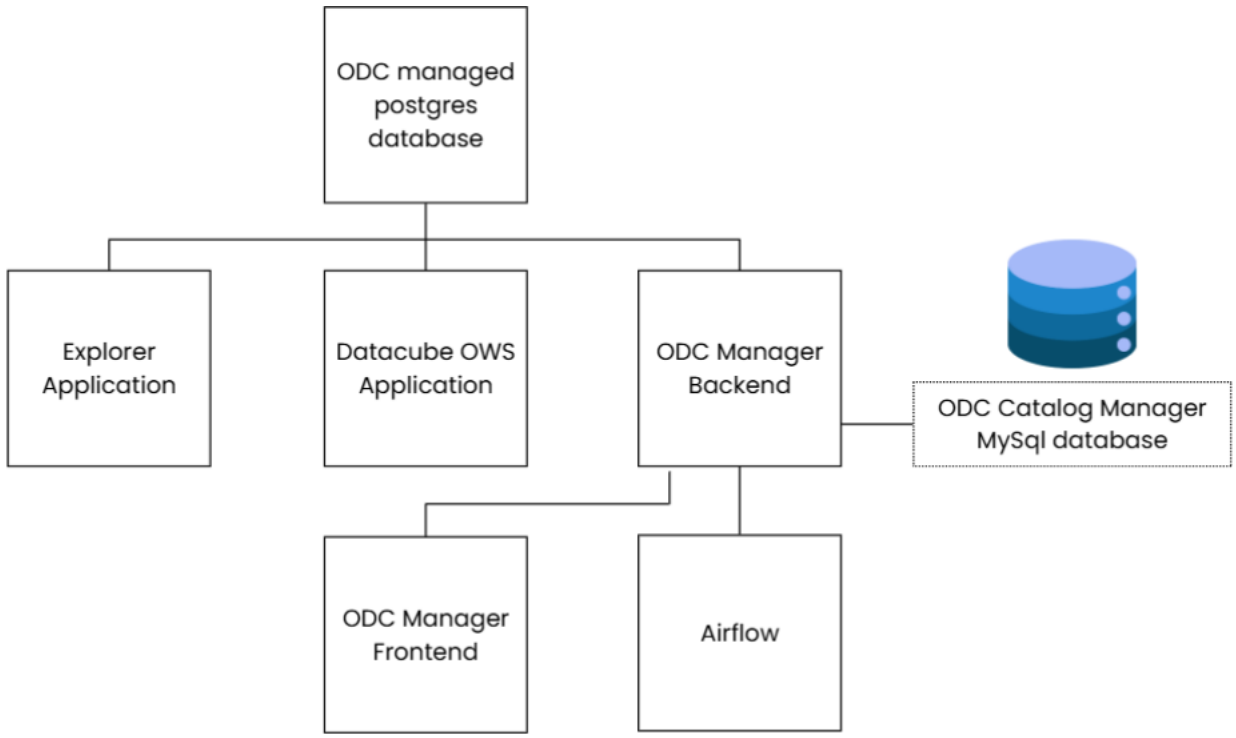


Figure 2 Container-based architecture of the Open Data Cube deployment, showing the ODC-managed PostgreSQL database, Explorer and Datacube OWS applications, the ODC Manager frontend and backend with its MySQL catalog database, and the Airflow orchestra service. [Source: Authors]

In the final stage of the migration, all Docker images were pushed to AWS Amazon Elastic Container Registry (ECR), creating two deployment environments: staging and production. All ODC applications now run on AWS Elastic Container Service (ECS) using a rolling deployment strategy, each with its own load balancer and target group. The ODC PostgreSQL database was also moved to an external instance on AWS Relational Database Service (RDS), ensuring consistency and reproducibility across the staging and production environments.

The ECR repositories used are:

- Iwmi-odc-explorer
- Iwmi-odc-ows
- Iwmi-odc-manager-frontend
- Iwmi-odc-manager-backend
- Iwmi-odc-airflow

Each repository contains Docker images tagged either staging or prod, depending on the target environment. The workflow for promoting new versions of the applications through staging and into production is summarized in Figure 3 (AWS deployment workflow), while Figure 4 presents a high-level AWS architecture diagram of the overall deployment.

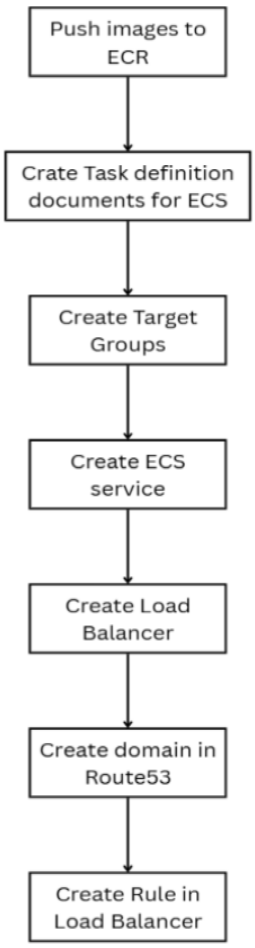


Figure 3 AWS deployment workflow for the ODC applications: push images to ECR, create ECS task definitions, configure target groups and ECS services, set up the load balancer, create the domain in Route 53, and define the load balancer routing rule. [Source: Authors]

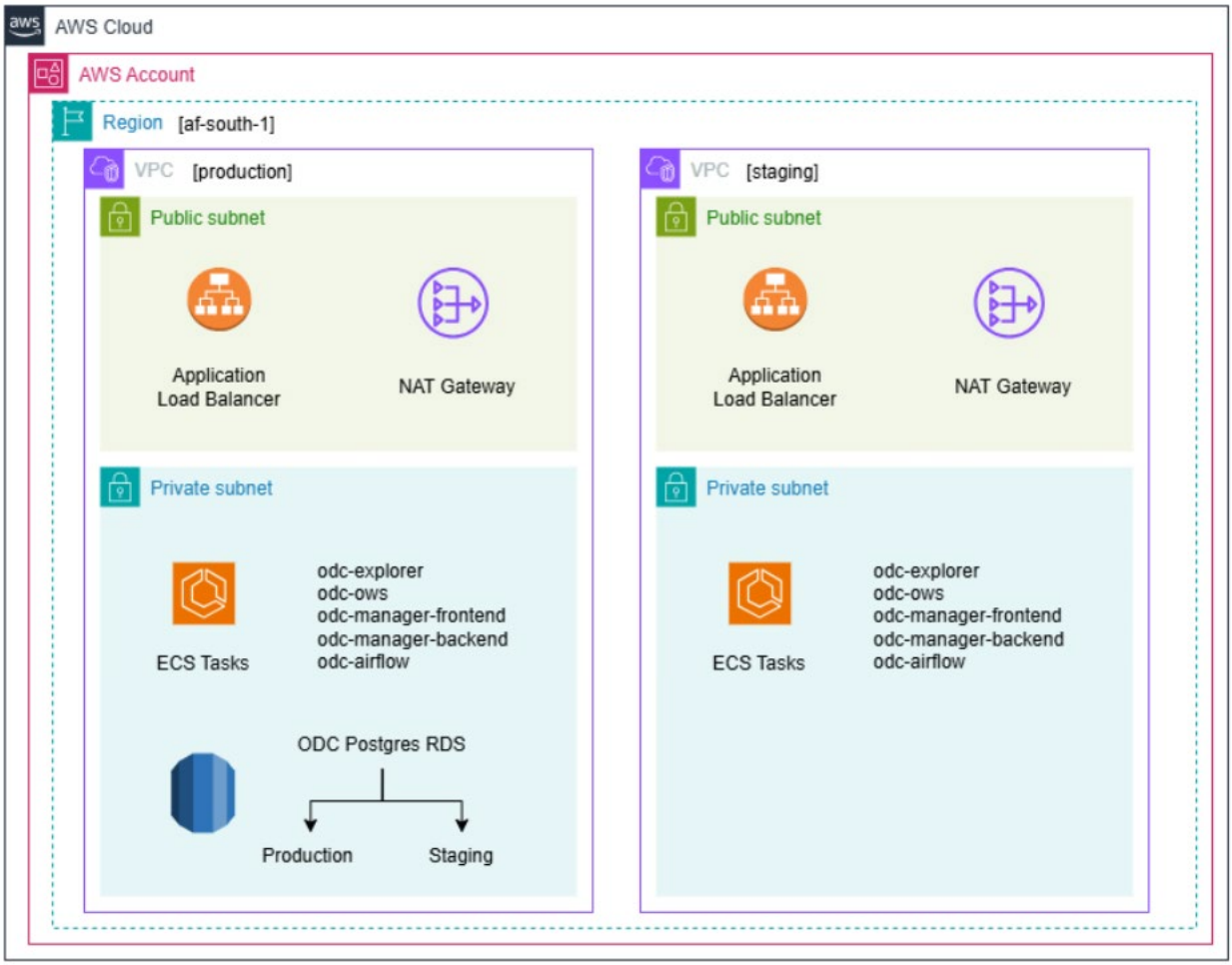


Figure 4 AWS high-level architecture for the Limpopo Digital Twin ODC deployment, showing production and staging VPCs with load balancers, NAT gateways, ECS tasks (odc-explorer, odc-ows, odc-manager-frontend, odc-manager-backend, odc-airflow), and a shared ODC Postgres RDS database. [Source: Authors]

Implementation of ODC manager application

Frontend

The frontend of the ODC Manager application was built using Vite. The components of the ODC Manager frontend are shown in Figure 5.

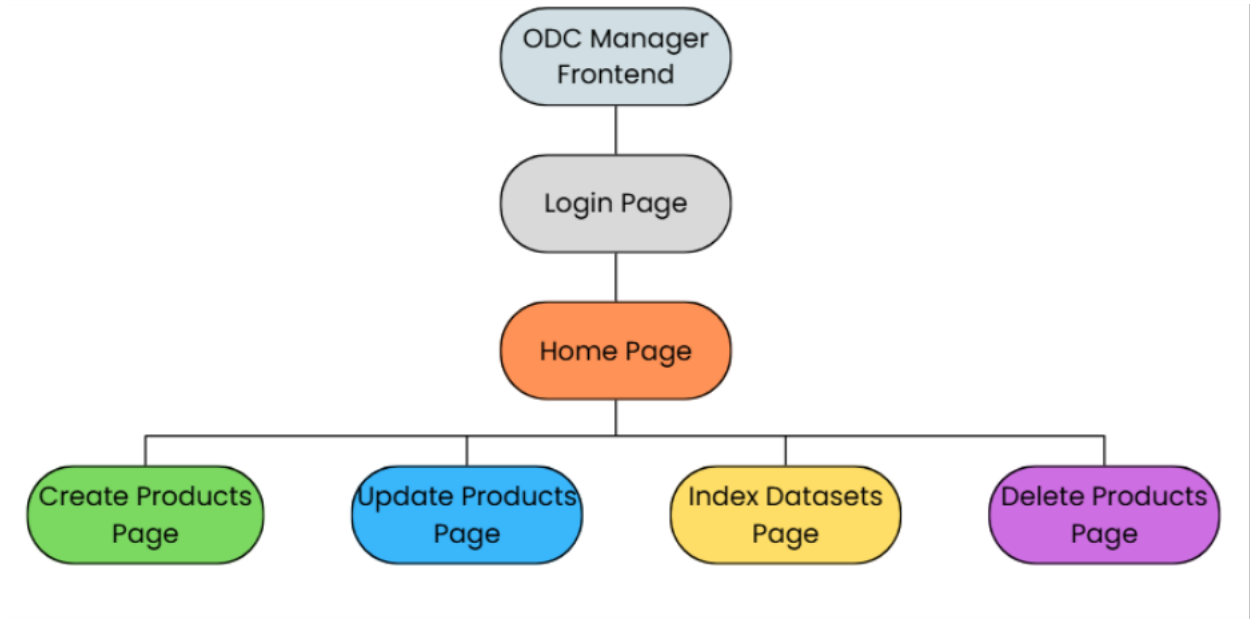


Figure 5 Structure of the ODC Manager frontend, showing the navigation flow from the login page to the home page and the four main product-management pages (Create Products, Update Products, Index Datasets, and Delete Products). [Source: Authors]

For login and authentication, Keycloak was set up, with the appropriate realm and the dt-odc-manager client configured (Figure 6).

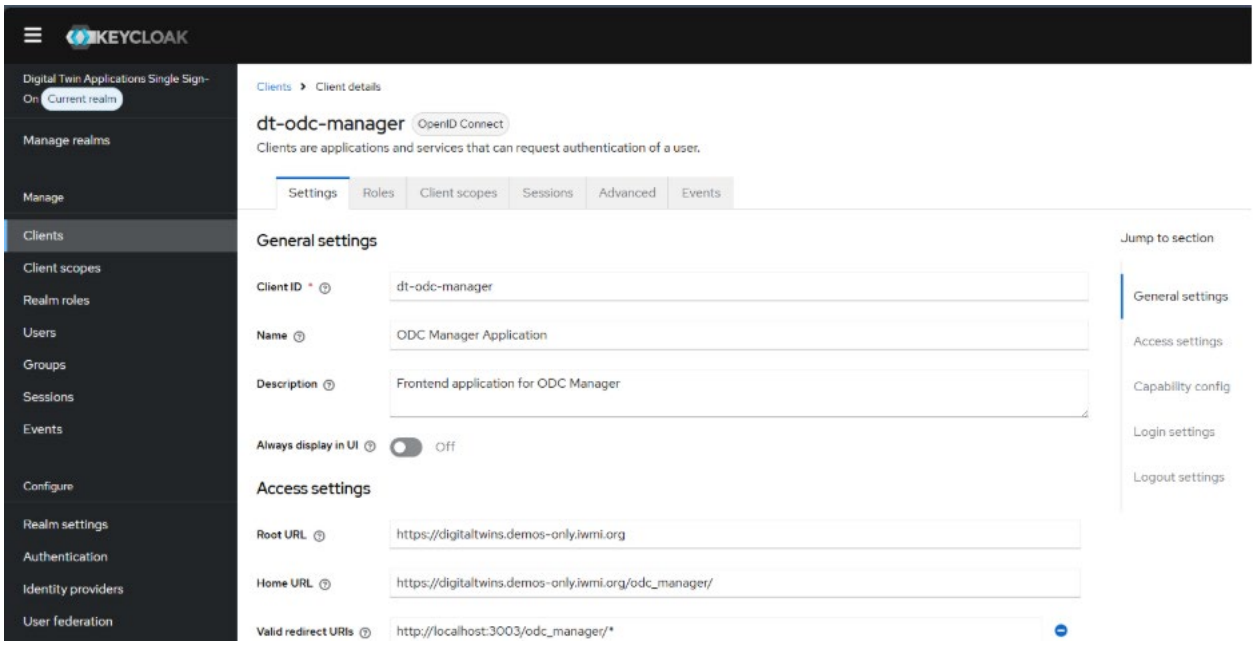


Figure 6 Keycloak configuration for the dt-odc-manager client in the Digital Twin Applications realm, used for authentication in the ODC Manager frontend. [Soucre: Authors]

The front-end code was then updated to integrate the new login system. It now includes logic to distinguish between administrative and regular users, paving the way for additional role-based UX improvements.

Backend

The backend is written in Python using Flask. The main API endpoints defined in app.py are listed below:

- GET / – Returns a welcome message to confirm that the backend is running.
- GET /logs/years – Lists available years of Airflow logs for a selected product.
- GET /logs/timestamps – Lists available log timestamps for a given product and year.
- GET /logs – Retrieves log lines for a specific product and timestamp.
- POST /login – Authenticates a user with a username and password.
- POST /check_username – Checks whether a username is available for registration.
- GET /products – Lists products available to the user (all products for admins, assigned products for other users).
- POST /check-product-name – Checks whether a product name already exists.
- POST /create-product – Creates a new product and updates related configurations.
- GET /product/<product_name> – Retrieves details and measurements for a specific product.
- PUT /product/<product_name> – Updates details and measurements for a specific product.
- GET /get-preview-uris – Generates preview URIs for a selected product and date range.
- POST /index-datasets – Indexes datasets for a product and updates ODC services.
- GET /preview-dataset-ids – Retrieves dataset IDs for a product and date range.
- POST /delete-datasets – Deletes selected datasets or an entire product.
- POST /upload-shapefiles – Uploads custom shapefiles for use in the system.

Docker-based Deployment of the ODC System

Docker (https://www.docker.com/) was used to containerize the running applications of the system, replacing the previous approach of running everything directly on the server and making the deployment more reproducible and easier to manage.

Containerization process

As part of the containerization process, the following steps were carried out:

Created individual Docker files for each service (backend Flask API, frontend application, Airflow, and supporting services).

Developed a comprehensive docker-compose.yml file to orchestrate all containers and their dependencies.

Configured proper networking between containers and volume mounts to ensure data persistence.

Set up environment variable management to support different deployment configurations.

Benefits achieved

This Docker-based setup brought several benefits:

- Consistency – the application now runs identically across development, testing, and production environments.

- Simplified deployment – the entire stack can be started with a single docker-compose up command.
- Isolation – each service runs in its own container, preventing dependency conflicts.
- Scalability – individual services can be scaled independently based on demand.
- Portability – the containerized application can run on any system with Docker installed.

Services containerized

The following services are now fully containerized:

- PostgreSQL database with PostGIS extensions.
- DataCube Explorer for data visualization.
- DataCube OWS for web mapping services.
- Flask backend API.
- Frontend application.
- Apache Airflow for workflow management.

Overall, the Docker-based approach has significantly reduced deployment complexity and improved system reliability.

System Outputs and ODC Manager Interface

Metadata Output

An example stac-item.json produced by the metadata pipeline is provided in Annex 1.

ODC Manager Application

Login

Access to the ODC Manager is handled through the Digital Twin Applications Single Sign-On page (Figure 7), where users authenticate with their institutional credentials or via federated providers (Google or Microsoft), as illustrated below.

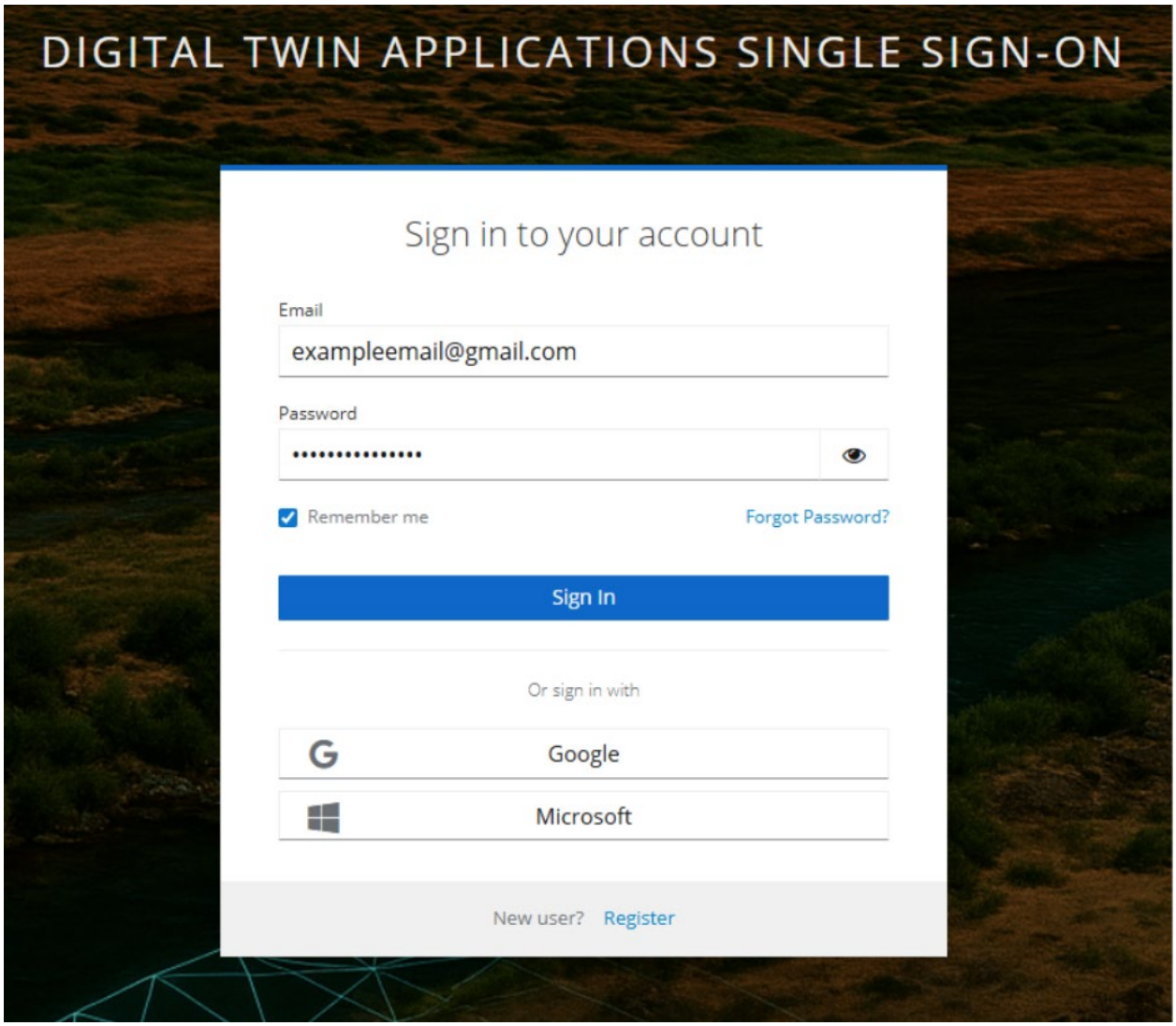


Figure 7 Digital Twin Applications Single Sign-On interface used to authenticate users before accessing the ODC Manager Interface. [Soucre: Authors]

Home page

The ODC Manager home page (Figure 8) provides a simple control panel where authenticated users can select one of their assigned products and perform key management actions such as creating, updating, indexing, or removing datasets.

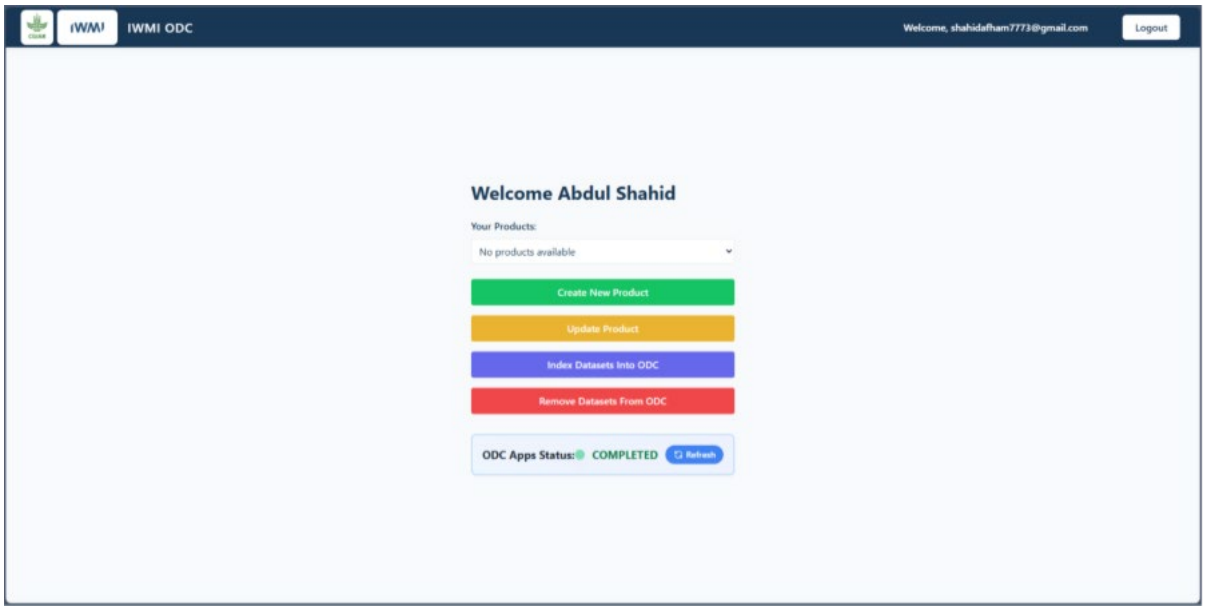


Figure 8 ODC Manager home page showing the product selection dropdown and the main management actions (Create New Product, Update Product, Index Datasets into ODC, and Remove Datasets From ODC). [Source: Authors]

Create product

The Create Product interface (Figure 9) guides users through entering all required high-level metadata for a new ODC product, including producer, maturity, version, family, category, DOI, update frequency, and region.

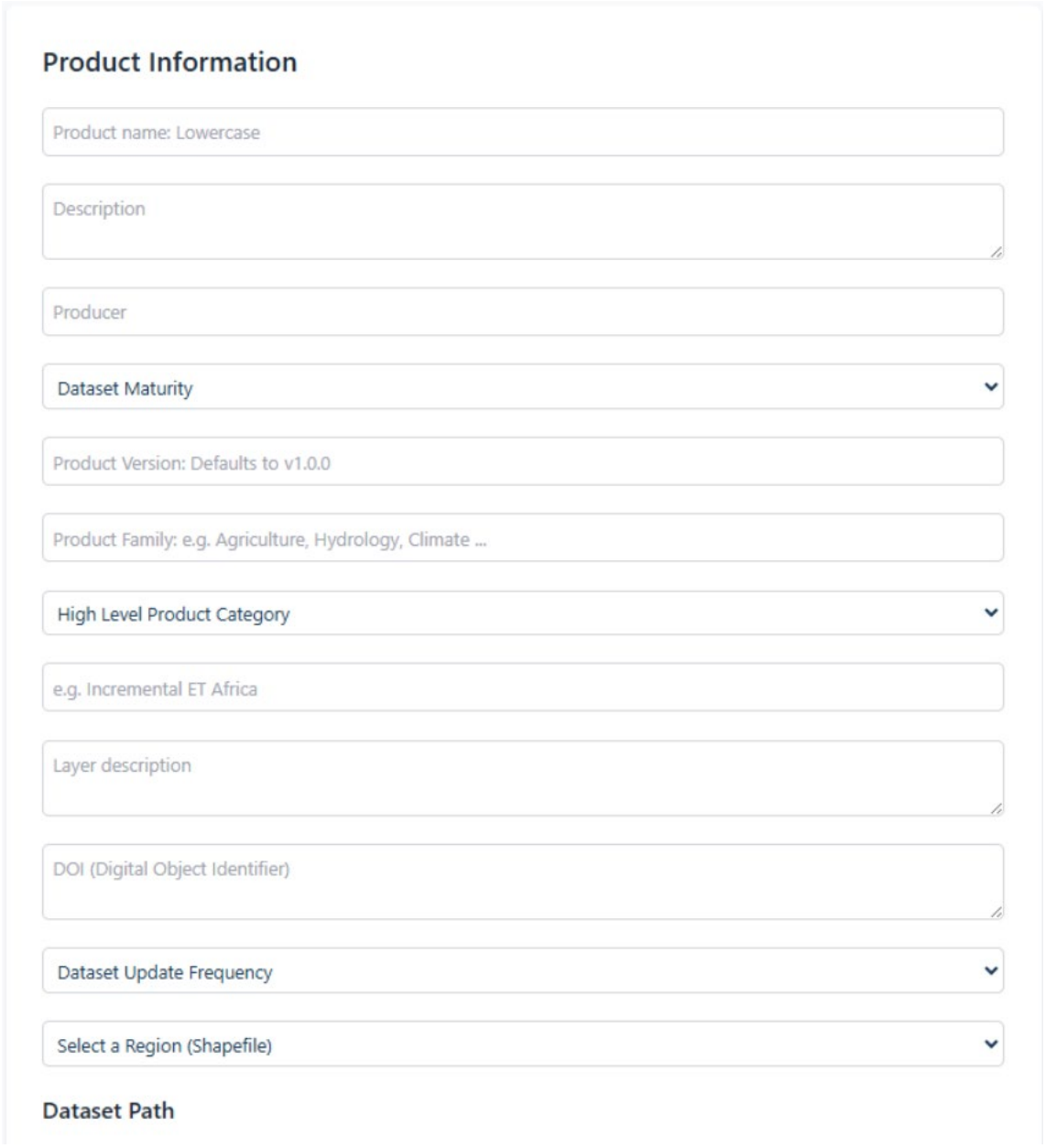


Figure 9 ODC Manager Create Product page showing the Product Information section with fields for core product-level metadata. [Source: Authors]

Create dataset path

This section of the Create Product form (Figure 10) captures the dataset path and measurement-level details, including example and formatted URIs as well as per-band metadata such as name, units, NoData value, data type, and style.

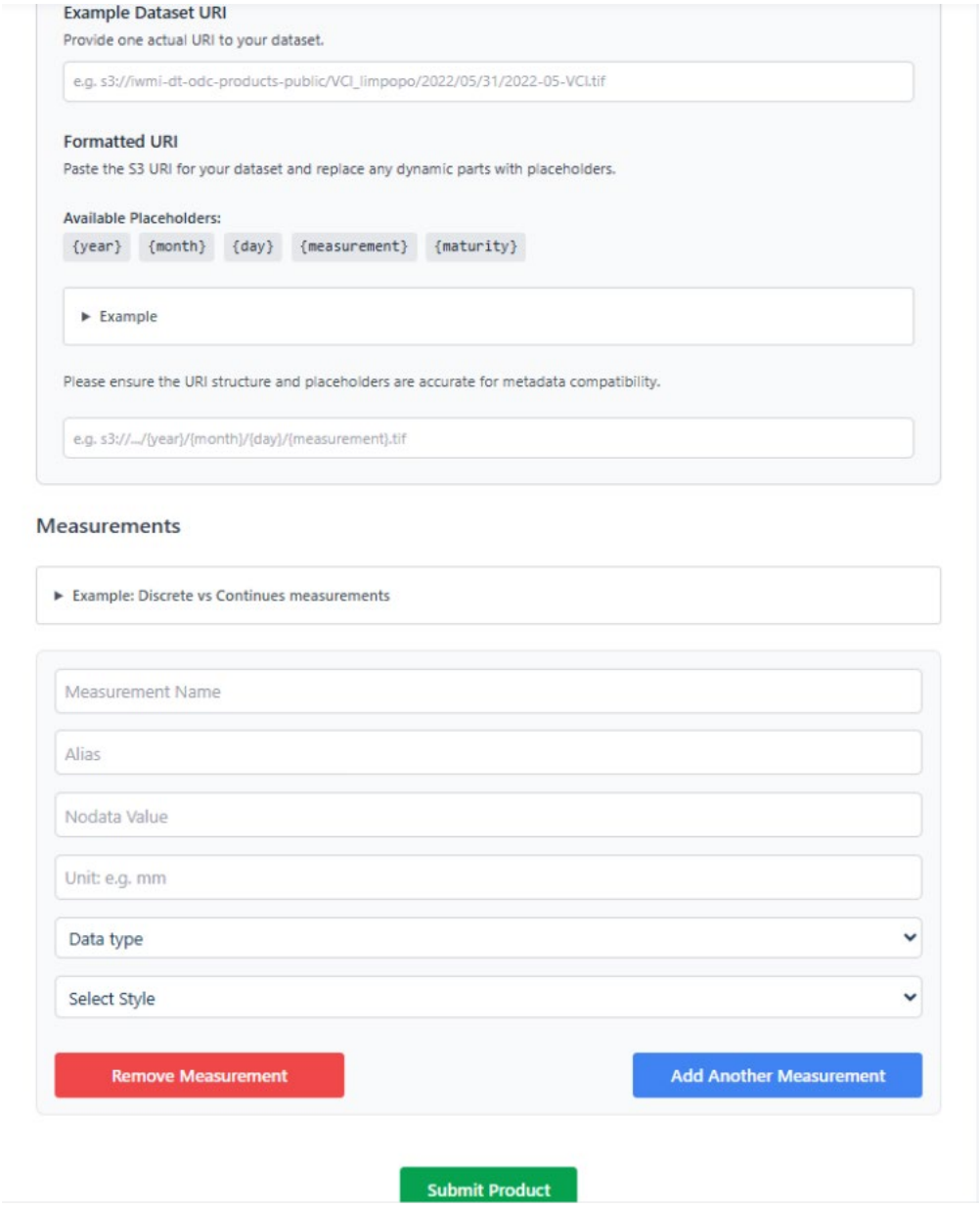


Figure 10 ODC Manager Create Product page showing the Dataset Path and Measurements sections used to define S3 URIs and measurement-specific metadata. [Source: Authors]

Index datasets

The Index Datasets page (Figure 11) allows users to select a product, specify one or more dates (via a date range or explicit list), preview the expected S3 URLs, and then trigger indexing of the corresponding datasets into ODC.

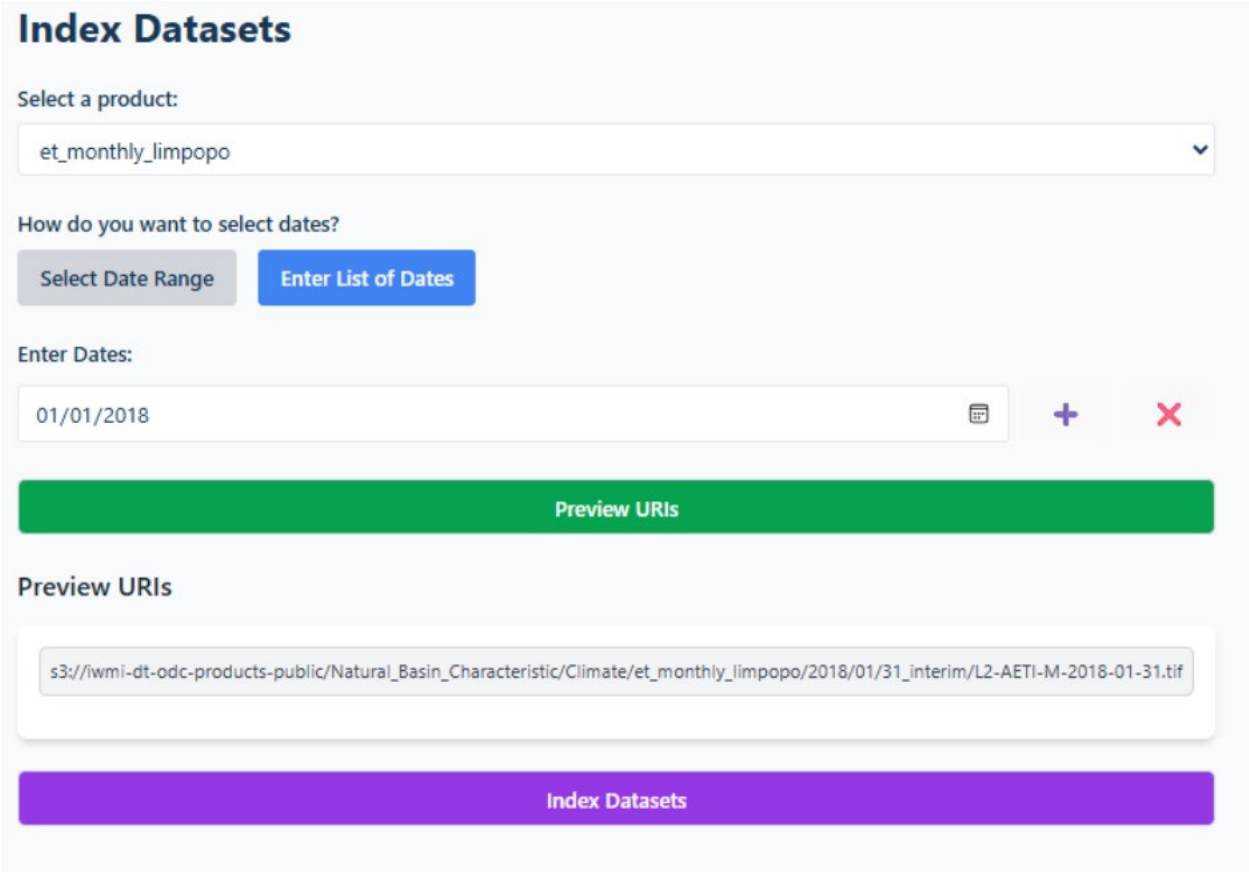


Figure 11. ODC Manager Index Datasets interface showing product selection, date input options, previewed dataset URIs, and the button to index datasets into the ODC. [Source: Authors]

Update product

The Update Product page (Figure 12) lets users revise key metadata for an existing product, including its description, version, layer name and description, DOI, formatted URI, and measurement-level attributes such as NoData value, data type, and style.

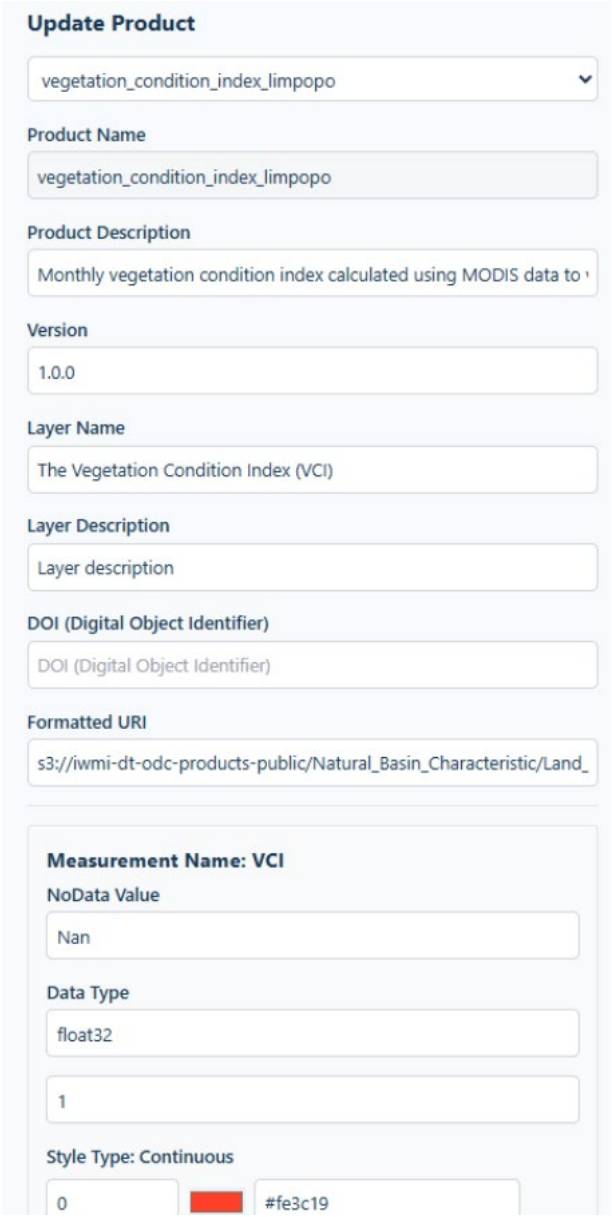


Figure 12 ODC Manager Update Product interface showing editable product- and measurement-level fields for the vegetation_condition_index_limpopo product. [Source: Authors]

Delete product

The Delete Datasets page (Figure 13) provides controlled removal of data from the cube, allowing users to delete either an entire product or selected datasets for specific dates after first loading and reviewing the corresponding dataset IDs.

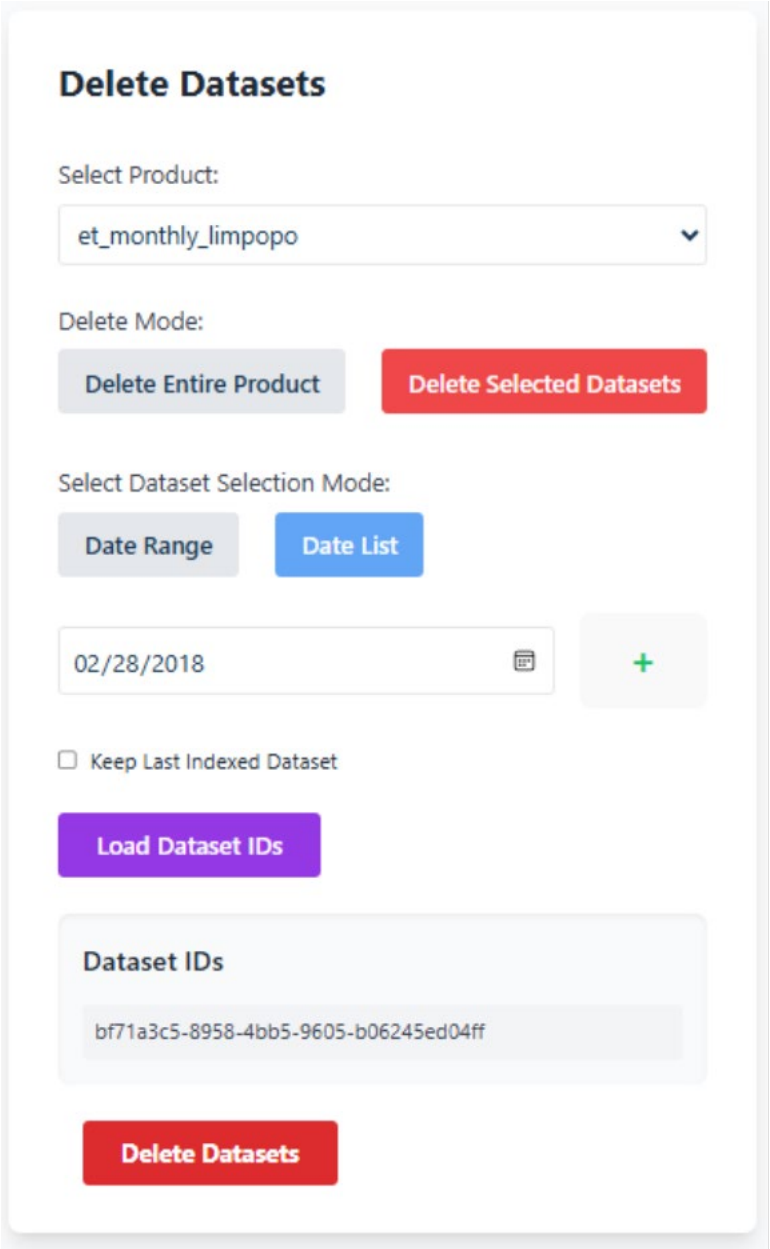


Figure 13 ODC Manager Delete Datasets interface showing product selection, delete mode (entire product or selected datasets), date-based selection options, loaded dataset IDs and the final delete action . [Source: Authors]

Conclusion

This report has outlined how the Limpopo Digital Twin ODC has been modernized to better support water-related decision-making in the Limpopo River Basin. While ODC remains the core framework for managing key raster products (irrigated areas, vegetation condition, evapotranspiration, land use, soils, and elevation), the implementation has been updated to align with cloud-native and STAC-based practices.

Three main improvements were introduced. First, the metadata pipeline was enhanced by moving from EO3 to STAC Items and adopting a metadata-only workflow, reducing processing time, storage overhead, and improving interoperability. Second, the ODC Manager application now gives authenticated users a structured way to create, update, index, and delete products, reducing reliance on developers and making catalog

management more transparent. Third, full containerization with Docker has made the system easier to deploy, replicate, and scale across environments.

Overall, the updated architecture treats ODC as part of a broader ecosystem rather than a monolithic solution: ODC is retained where it adds value, while STAC, containerization, and modern web tools prepare the Limpopo Digital Twin for future integration with emerging cloud-native geospatial services and new basin-relevant products.

References

Afham, A., Silva, P., Ghosh, S., Kiala, Z., Retief, H., Dickens, C. and Garcia Andarcia, M., 2024. Limpopo River Basin Digital Twin Open Data Cube Catalog.

Kopp, S., Becker, P., Doshi, A., Wright, D.J., Zhang, K. and Xu, H., 2019. Achieving the full vision of earth observation data cubes. *Data*, *4*(3), p.94.

Lewis, A., Oliver, S., Lymburner, L., Evans, B., Wyborn, L., Mueller, N., Raevksi, G., Hooke, J., Woodcock, R., Sixsmith, J. and Wu, W., 2017. The Australian geoscience data cube—foundations and lessons learned. *Remote Sensing of Environment*, *202*, pp.276-292.

Mafuta, C., Skaalvik, J., Sevaldsen, P., RESILIM, U. and SA, G., 2021. Limpopo River Basin-Changes, challenges and opportunities.

Acknowledgements

This study was funded by the CGIAR Initiative on Digital Innovation, which advances sustainable agrifood systems through digital solutions and the International Water Management Institute's Digital Innovations for Water Secure Africa (DIWASA) project. We also wish to thank all funders who supported this research through their contributions to the CGIAR Trust Fund (https://www.cgiar.org/funders/) and the Leona M. and Harry B. Helmsley Charitable Trust for their financial support for the DIWASA project. We also wish to thank the Limpopo Watercourse Commission (LIMCOM) for their support to this project.

Annexure

Annex 1. Example of STAC Item (stac-item.json)

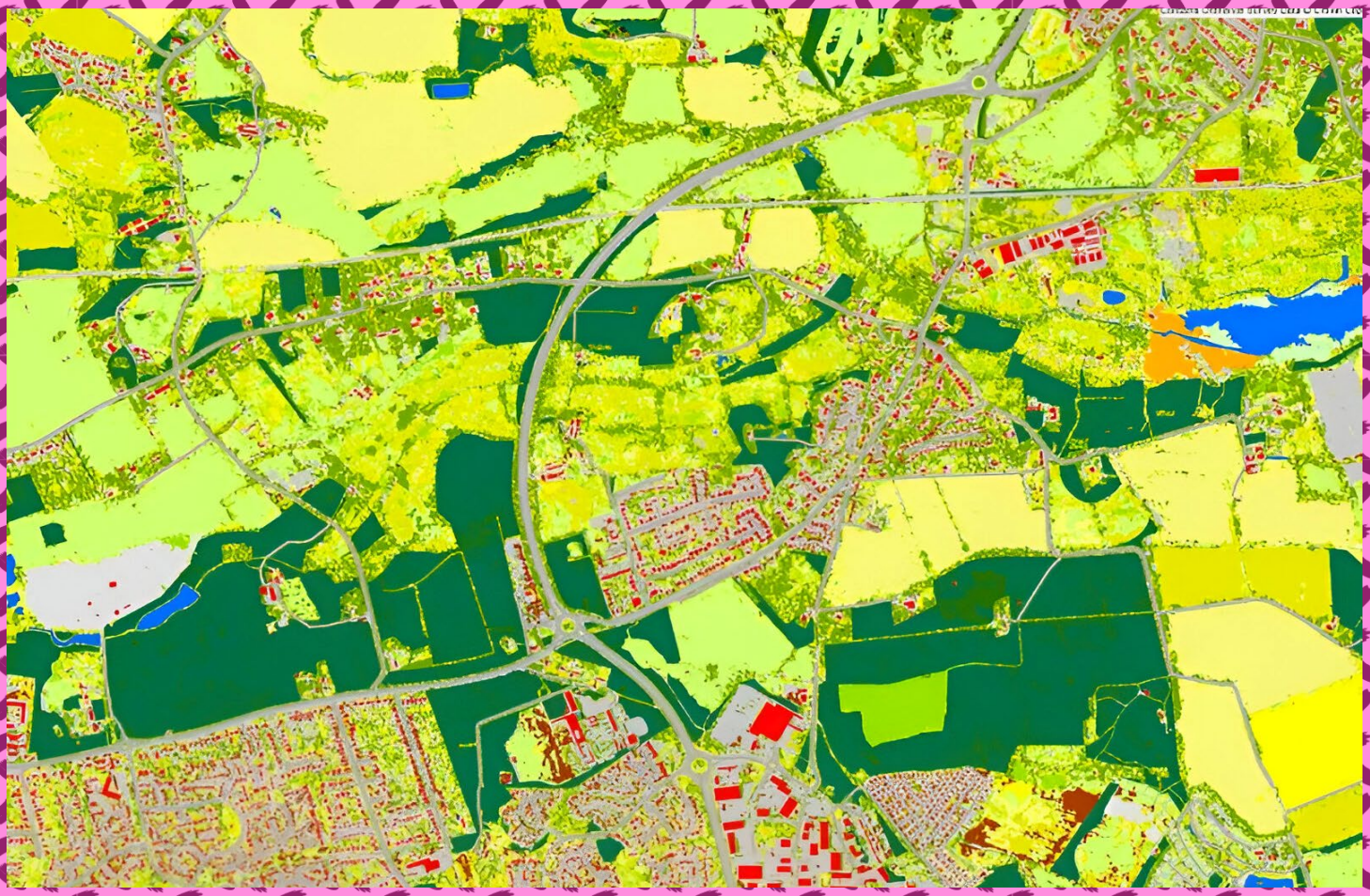
The example below shows a STAC Item JSON document generated for the irrigated_areas_limpopo product (August 2025, interim version):


```
{
    "type": "Feature",
    "stac_version": "1.0.0",
    "id": "005f1b16-8890-4cf3-ac7d-c6fddb3a7aa5",
    "properties": {
        "title": "irrigated_areas_limpopo-v1-0-0_2025-08-31_interim",
        "start_datetime": "2025-08-01T00:00:00Z",
        "end_datetime": "2025-08-31T23:59:59Z",
        "created": "2025-09-09T14:12:42.563983Z",
```

```
        "odc:producer": "IWMI",
        "dea:dataset_maturity": "interim",
        "odc:dataset_version": "1.0.0",
        "odc:product_family": "Agriculture",
        "proj:epsg": 4326,
        "proj:shape": [
            78917,
            110039
        ],
        "proj:transform": [
            0.00008983152841195215,
            0,
            24.970829278783988
        ],
        "datetime": "2025-08-31T00:00:00Z"
    },
    "geometry": {
        "type": "Polygon",
        "coordinates": []
    },
    "links": [
        {
            "rel": "self",
            "href":
"/app/src/odc_pipeline/irrigated_areas_limpopo/2025/08/31_interim/irrigated_ar
eas_limpopo-v1-0-0_2025-08-31_interim.stac-item.json",
            "type": "application/json"
        }
    ],
    "assets": {
        "filtered": {
            "href": "s3://iwmi-dt-odc-products-
public/Natural_Basin_Characteristic/Land_cover_and_Land_use/irrigated_areas_li
mpopo/2025/08/31_interim/2025-08-filtered.tif",
            "type": "image/tiff; application=geotiff; profile=cloud-
optimized",
            "title": "filtered",
            "eo:bands": [
                {
                    "name": "filtered"
                }
            ],
            "proj:epsg": 4326,
            "proj:shape": [
                78917,
                110039
            ],
            "proj:transform": [
```

```
                0.00008983152841195215,
                0,
                24.970829278783988,
                0
            ],
            "roles": [
                "data"
            ]
        },
        "map": {
            "href": "s3://iwmi-dt-odc-products-
public/Natural_Basin_Characteristic/Land_cover_and_Land_use/irrigated_areas_li
mpopo/2025/08/31_interim/2025-08-map.tif",
            "type": "image/tiff; application=geotiff; profile=cloud-
optimized",
            "title": "map",
            "eo:bands": [
                {
                    "name": "map"
                }
            ],
            "proj:epsg": 4326,
            "proj:shape": [
                78917,
                110039
            ],
            "proj:transform": [
                0.00008983152841195215,
                0,
                24.970829278783988
            ],
            "roles": [
                "data"
            ]
        },
        "prob": {
            "href": "s3://iwmi-dt-odc-products-
public/Natural_Basin_Characteristic/Land_cover_and_Land_use/irrigated_areas_li
mpopo/2025/08/31_interim/2025-08-prob.tif",
            "type": "image/tiff; application=geotiff; profile=cloud-
optimized",
            "title": "prob",
            "eo:bands": [
                {
                    "name": "prob"
                }
            ],
            "proj:epsg": 4326,
```

```
            "proj:shape": [
                78917,
                110039
            ],
            "proj:transform": [
                0.00008983152841195215,
                0,
                24.970829278783988,
                0
            ],
            "roles": [
                "data"
            ]
        }
    },
    "bbox": [
        24.970877311247406,
        -26.713572781949228,
        34.85579428909439,
        -19.624796531218106
    ],
    "stac_extensions": [
        "https://stac-extensions.github.io/eo/v1.1.0/schema.json",
        "https://stac-extensions.github.io/projection/v1.1.0/schema.json"
    ],
    "collection": "irrigated_areas_limpopo"
}
```



CGIAR is a global research partnership for a food-secure future. CGIAR science is dedicated to transforming food, land, and water systems in a climate crisis. Its research is carried out by 13 CGIAR Centers/Alliances in close collaboration with hundreds of partners, including national and regional research institutes, civil society organizations, academia, development organizations and the private sector. www.cgiar.org
To learn more about this and other Science Programs and Accelerators in the CGIAR Research Portfolio 2025–2030, please visit www.cgiar.org/cgiar-research-portfolio-2025-2030/

Contact
Mariangel Garcia Andarcia, Research Group Leader - Water Futures Data & Analytics (WFDA), International Water Management Institute (IWMI), Colombo, Sri Lanka (M.GarciaAndarcia@cgiar.org)

CGIAR DIGITAL TRANSFORMATION IWMI International Water Management Institute