

Article

Prototyping of Utilization Model for KOMPSAT-3/3A Analysis Ready Data Based on the Open Data Cube Platform in Multi-Cloud Computing Environment: A Case Study

Kwangseob Kim [†]  and Kiwon Lee ^{*} 

Department of Electronics and Information Engineering, Hansung University, Seoul 02876, Republic of Korea; engkwangseob@gmail.com or kwangseob@kyungmin.ac.kr

^{*} Correspondence: kilee@hansung.ac.kr

[†] Current address: Department of Computer Software, Kyungmin University, Uijeongbu-City 11618, Republic of Korea.

Abstract: This study introduces a multi-cloud model that combines private and public cloud services for processing and managing satellite images. The multi-cloud service is established by incorporating private clouds within organizations and integrating them with external public cloud services to utilize the data. Private clouds can maintain data security within an organization or between organizations, while public clouds offer easy processing options for general users with access accounts. The model for the private cloud service utilizes open-source OpenStack software to create virtual machines, allowing users to manage analysis ready data (ARD) of the Korea Multi-Purpose Satellite (KOMPSAT)-3/3A images simultaneously. The public cloud service through Amazon Web Services (AWS) offers four services and uses the Open Data Cube (ODC) to manage data and provide web-based time-series visualization and processing. The model utilizes OpenStack to create virtual machines, and the public cloud service through AWS offers various services using ODC to manage data. A system that handles large amounts of satellite imagery in a multi-cloud environment has benefits such as improved availability, cost savings through open-source, and enhanced scalability. We present a prototyped utilization model that can be used with the ODC user interface (UI) that applies the proposed multi-cloud model. The multi-cloud model of this study can be applied to constructing a country-scale data cube system, that deals with large-scale satellite image data. It can also be applied to systems that need to be built with data that is tailored to a specific user's needs at any institution.

Keywords: analysis-ready data; Amazon Web Services; KOMPSAT-3/3A; multi-cloud; open data cube; OpenStack



Citation: Kim, K.; Lee, K. Prototyping of Utilization Model for KOMPSAT-3/3A Analysis Ready Data Based on the Open Data Cube Platform in Multi-Cloud Computing Environment: A Case Study. *Appl. Sci.* **2023**, *13*, 10478. <https://doi.org/10.3390/app131810478>

Academic Editors: Christos Bouras and Dimitris Mourtzis

Received: 16 July 2023

Revised: 24 August 2023

Accepted: 18 September 2023

Published: 20 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The emergence of cloud computing is rooted in resource rental, rather than resource ownership, of computer resources such as hardware and software. Typically, the cloud service model is classified as software as a service (SaaS), platform as a service (PaaS), which provides the foundation for running applications and infrastructure as a service (IaaS) which offers memory, a central processing unit (CPU), storage, and networking resources. Additionally, cloud computing can be classified according to deployment models such as private, public, community, or hybrid cloud [1].

The reasons why a cloud-based system approach is necessary for managing and analyzing earth observation (EO) satellite information are summarized as follows [2,3]: First, it enables easy processing of large-scale data. Data collected from EO satellites has a substantial capacity. A considerable amount of computing resources is needed to process such a large volume of data. Using a cloud system allows resources to be expanded as needed, making it possible to process large-scale data. Second, it allows for flexible system configurations. Cloud systems can use virtualization technology to flexibly adapt the

required resources. This makes it possible to quickly change the system configuration and add the necessary resources to build a system optimized for EO satellite data. Third, it has high security. Cloud service providers are very interested in security and offer various security solutions. In addition, cloud systems provide functions such as data backup, restoration, and disaster recovery, ensuring data stability. Fourth, it is cost-effective. Using a cloud system can reduce costs for purchasing or maintaining resources such as servers or storage. In addition, cloud services adopt a pay-as-go model, meaning that only the necessary resources are used, making cost management more efficient.

With these advantageous points of the cloud computing platform, many attempts and achievements have been made in developing remote sensing platforms or frameworks based on cloud computing environments for EO applications. Yue et al. [4] proposed a machine learning approach to predict the computational intensity by dividing the spatial domain into balanced subdivisions for better performance. Yan et al. [5] developed a product generation system using multi-source remote sensing data in cloud computing. Yao et al. [6] described how EO data can be utilized by using cloud computing in discrete global grid systems. Huang et al. [7] used an elastic computing scheme with a Spark engine on Kubernetes for real-time fusion processing of remotely sensed big data. Kline [8] summarized the state of Landsat migration in the cloud environment at the United States Geological Survey (USGS). Astsatryan et al. [9] proposed a scalable EO data processing platform interoperable with high-performance computing (HPC) cloud and EO data repositories, capable of connecting any data repository that supports web coverage service. In addition to the research on applying cloud technology for data management and distribution of satellite imagery as described above, there is also active research being conducted on SaaS that provides practical analysis capabilities in various fields such as land cover mapping, disaster monitoring, and wetland monitoring [10–13].

As mentioned above, there are various approaches to research on the application of cloud computing for EO data. This study aims to explore research that specifically focuses on integrating topics such as ARD and ODC with cloud computing, which can be operated in OpenStack and AWS environments. This is because ODC is typically operated on AWS, and ODC inherently provides ARD as a fundamental feature. In this study, the OpenStack cloud environment is employed in conjunction with AWS as the cloud environment.

ARD enables users to easily access data without complex processing by generating data for areas of interest on a regular basis. Research on the types and generation of ARD in satellite information has consistently received attention as one of the topics of interest. Over the years, efforts have been made to provide ARD for Landsat satellites. Dwyer et al. [14] defined the contents of Landsat ARD, and the United States Geological Survey (USGS) provided the ARD Data Format Control Book for all Landsat products [15,16]. Producing data in the form of ARD is an important issue.

In addition, it is critical for future applications to explore approaches for distributing data at a similar level. In particular, ARD is processed in large time series rather than individual datasets. Difficulties may arise when providing this through conventional techniques. Cloud computing is the technology that can address this issue, with Google Earth Engine (GEE) and AWS being prominent commercial services. The most convenient way to use satellite image ARD, such as Landsat, Moderate Resolution Imaging Spectroradiometer (MODIS), and Sentinel, is to utilize GEE. The GEE platform provides these satellite data and offers preprocessed ARD data that users can directly utilize by performing various processing steps [17]. The GEE platform offers satellite information processing algorithms that support various time series satellite big data analysis applications [18]. Users can upload their own processed results to be used as ARD data. GEE has the potential to support both artificial intelligence (AI) methods and infrastructure to utilize satellite information as a type of big data, but a smoother integration of the two technologies is needed to achieve this, according to Yang et al. [19].

As for an ARD guideline for remote sensing products, the Committee on EO Satellites (CEOS) has pioneered a basic definition and specification [20]. The CEOS-ARD can be

defined as attributes of actual measurement products for global remote sensing users with remote sensing applications such as surface reflectance, surface temperature, and normalized backscatter, and it means the minimum level for time series analysis and multi-typed data interoperability. Especially, CEOS Analysis Ready Data for Land (CARD4L) products are processed to minimum requirements and organized into a guideline that allow for immediate analysis with a minimum of additional user-side efforts. They can be resampled as a common geometric grid and would provide reference data for increased interoperability over time and with other datasets.

Kopp et al. [21] discussed data cubes linked to a cloud computing environment for EO data sets. Giuliani et al. [22] tested Earth Observation Data Cube (EODC) to facilitate the production of essential variables at different scales and to benefit from the spatial and temporal dimensions of satellite EO data for enhanced environmental monitoring. Sudmanns et al. [23] implemented data cube of Sen2Cube for Sentinel-2 MultiSpectral Instrument (MSI) images covering Austria, with knowledge-based large EO data analysis capabilities. The practical benefits of applying the data cube approach as an infrastructure for managing and analyzing large-scale satellite imagery have been demonstrated by initiatives such as Euro Data Cube [24].

Geoscience Australia (GA) and Digital Earth Australia (DEA) initiated the Open Data Cube project intending to develop tools for processing and analyzing large-scale geospatial data. Initially, the development focused on Landsat satellite image data. In 2015, Geoscience Australia released the Australian Geoscience Data Cube. The overall goal of the ODC is to facilitate the management of extensive data collection, eliminating the need for data to adhere to specific storage formats or locations. In essence, this entails the ability to direct the ODC to data repositories and index the data in its existing location, thereby abstracting the complexities associated with monitoring large and distributed data holdings [25–27].

ODC operates in a cloud environment due to its handling of large-scale EO data, and the ODC application services developed so far are deployed in an AWS-based environment. Ferreira et al. [28,29] developed the Brazil Data Cube on AWS, to provide services to create, integrate, discover, access, and process the data sets to produce land use and cover maps using time series analysis and machine learning schemes, with ARD generation capabilities. The Japan Aerospace Exploration Agency (JAXA) provides that the public release of the ScanSAR products are provided as CEOS-ARD normalized radar backscatter (NRB) on AWS data exchange [30]. Digital Earth Africa provides Sentinel-1 backscatter products developed to comply with the CARD4L specifications on AWS [31,32]. Cao et al. [33,34] implemented the China Data Cube (CDC) framework for Landsat 8 and Gaofen-1 (GF-1) ARD based on ODC. The ODC-Spatio Temporal Asset Catalogs (STAC) API provides the functionality to access Sentinel-2 imagery on AWS [35].

Such as these trends, the data cube system for satellite imagery that has been developed or built is mainly based on AWS. The CEOS-ARD data cube application model is also advantageous for integration with AWS. Of course, although AWS is currently the most widely used and versatile cloud environment, there may be situations where it is built on other types of clouds. In addition, application models that integrate open-source cloud environments with data cubes can have many advantages in terms of both economic and technological aspects. Of course, the economic advantage of open-source cloud platforms is that there are no license costs. The technological advantage is that only the necessary elements can be applied according to the target service, making the system lightweight and optimization easier. However, such advantages are based on prior knowledge or experience of open-source cloud technology specifications and development environments. With such technological backgrounds, there are advantages such as the ease of extending the system to multiple clouds.

OpenStack is a popular cloud platform for implementing application services such as IaaS, an open-source software cloud computing platform [36]. As expected, there are some essential cases in which a service system has been developed to manage, analyze, and distribute EO satellite images using OpenStack as an IaaS environment. OpenStack

is a fundamental technology used in the NASA Earthdata program to support the Earthdata Cloud [37]. The Earthdata Cloud is a cloud-based platform that hosts and provides access to many Earth science data, including remote sensing imagery. OpenStack is also a core technology in the EODC program, a part of the European Space Agency (ESA) Earth Observation Exploitation Platform, which promotes the development of data exploitation platforms to enable efficient access, processing, and analysis of Earth observation data [38]. Semlali and Freitag [39] proposed a distributed system for remote sensing data pre-processing and ingestion using OpenStack to integrate data into the Hadoop storage system, notably the Hadoop Distributed File System (HDFS), HBase, and Hive.

Lee et al. [40] implemented a cloud-based satellite image processing service using the open-source stack for geo-based data processing (<https://ksatdb.kari.re.kr/main/main.do> (accessed on 26 June 2023)) at the Korea Aerospace Research Institute (KARI). Lee et al. tested the performance of remote sensing image processing services based on the PaaS of Cloud Foundry and OpenStack [41]. Kim and Lee implemented OpenStack-based SaaS to produce top-of-atmospheric reflectance (TOAR) and top-of-canopy reflectance (TOCR) for KOMPSAT-3/3A Satellite Image [42].

In this study, we first described the design and components of the multi-cloud system configuration. We also presented a sequence diagram executed according to this design strategy. As a result of the system implementation, we presented the ODC processing steps in a multi-cloud environment using the TOAR/TOCR ARD of KOMPSAT-3/3A bundled images and a simple example application in the ODC UI.

2. System Design and Its Components

The AWS provides infrastructure and services for satellite ground station operations, data storage, and distribution. The collected satellite information can be stored using AWS Elastic Compute Cloud (EC2) and AWS Simple Storage Service (S3) systems. It enables further processing of stored satellite information using EC2, and pre-processed data or ARD can be made available to users through the AWS marketplace, which is a catalog that hosts various products such as software, data, and services.

The Landsat, Sentinel, and MODIS satellite images provide features to process data with platforms such as GEE and publicly available AWS S3. These platforms are easily accessible and available for anyone to download and use through an open policy. However, commercial satellite images have limited data access and cannot be directly verified before purchase. In Korea, the high-resolution KOMPSAT series has been developed, launched, and operated by the KARI. KOMPSAT image sets can be searched and easily viewed by users through utilization support services of satellite images that provide image ordering services and primary index and metadata. However, the actual data must be purchased to use. Furthermore, the service operates in a cloud environment and offers a feature that enables users to experience the image analysis service by providing test images in some regions.

The development of various information technology (IT) technologies has expanded the scope of satellite information utilization. However, most active utilization is observed for open satellites, while there needs to be more consideration for ARD generation and utilization of high-resolution commercial satellites. Therefore, this study proposes a utilization approach for the KOMPSAT series using a multi-cloud model. The multi-cloud model utilizes a combination of proprietary and open cloud computing technologies. Since most commercial satellites require the purchase of original data, security is crucial, and data management is typically handled by the organization that launched the satellite. Multi-cloud and hybrid cloud refer to the use of multiple cloud services, but significant differences exist. Multi-cloud refers to the independent use of services from multiple cloud providers. On the other hand, hybrid cloud refers to using services from one or more cloud providers in a single, unified environment. The advantage of multi-cloud is greater flexibility and choice. Users can select the cloud provider that best meets their specific requirements for an application, and they can change their cloud-based environment as needed. On the

other hand, a hybrid cloud is easier to secure and manage because users must use a single, unified environment.

There are many cases of both uses of OpenStack and AWS for cloud computing. For example, an organization might use AWS for its public-facing web applications and OpenStack for its internal infrastructure and data processing tasks. It allows the organization to take advantage of both cloud platforms' benefits, such as AWS's high availability and scalability and OpenStack's security and data control. Tools and services also allow organizations to manage and orchestrate workloads across multiple cloud platforms, such as AWS and OpenStack. For example, the OpenStack cloud management platform supports integration with AWS through tools such as an OpenStack EC2 API, which enables OpenStack users to manage and deploy AWS resources from within the OpenStack dashboard.

2.1. Overview of KOMPSAT-3/3A

The multi-spectral band specification of KOMPSAT-3 launched on 17 May 2012, is as follows: 450–900 μm for the panchromatic band, 450–520 μm for blue, 520–600 μm for green, 630–690 μm for red, and 760–900 μm for near-infrared (NIR). The GSD, which stands for ground sample distance at the nadir, is 0.70 m for a panchromatic image. The GSD value for multi-spectral bands and NIR data is 2.8 m and 5.5 m, respectively. KOMPSAT-3A was launched on March 25, 2015, as a successor to this satellite. This has the same multi-spectral band specifications as KOMPSAT-3. The GSD of a panchromatic image is 0.55 m at the nadir. The value for multi-spectral bands and NIR data are 2.2 m and 5.5 m, respectively. The swath width at the nadir for KOMPSAT-3 and KOMPSAT-3A is 15 and 12 km, respectively.

The average size of the KOMPSAT-3/3A image data is approximately 2.1 GB and 2.5 GB, respectively, based on the Level-1-Geometric corrected (Level-1G) bundled product.

KOMPSAT-3 and KOMPSAT-3A image data are archived at a rate of 270 TB and 160 TB per year, respectively, including other types of products, including Level-1G Bundled products. It is estimated that the KOMPSAT-3/3A images accumulated by 2024 will be approximately 100 petabytes, based on an estimate, although this is not official data from the distribution agency or sales company.

In this study, ARD refers to pre-processed and corrected data that can be used for various purposes, as it is not yet in the self-assessment and peer review stage of CEOSCARD4L-ARD, which is the official ARD for KOMPSAT-3/3A images according to CEOS's Product Family Specifications. The KARI is currently developing a dedicated system for KOMPSAT ARD production in accordance with the CARD4L guidelines. The prototype model proposed in this study focuses on creating ARD-formatted data based on open source and providing data cube services that can utilize this ARD separately from KARI's KOMPSAT ARD.

2.2. Concept of Multi-Cloud Service Based on ODC

The multi-cloud service proposed in this study is based on installing and operating private clouds within organizations, which are then integrated with external public cloud services to utilize the data. The cloud application model proposed in this study is a model that integrates a private cloud for commercial satellite images and a public cloud service for processing free satellite images. Private clouds can manage original data to maintain data security and can be expanded internally for various research purposes. Public clouds can include SaaS that allows anyone to access and perform basic processing on data prior to purchase. While these services can be applied to private clouds, public clouds have the access and expansion, making them easy to apply and expand services quickly.

Therefore, the multi-cloud model proposed in this study includes processing systems for providing data on the private cloud, as shown in Figure 1, and additional services that internal users can use for various processing in the future. As an example of the model that processes ARD data, including the surface reflectance SaaS designed and constructed in [42], this study added a flow that can be connected to public cloud services. In particular, we propose a model that can efficiently add and analyze multiple satellites in large time-

series form, even in the internal cloud environment, by adding the ODC platform, which can manage and visualize diverse data. To index data in ODC, the satellite sensor model indexing process needs to be added.

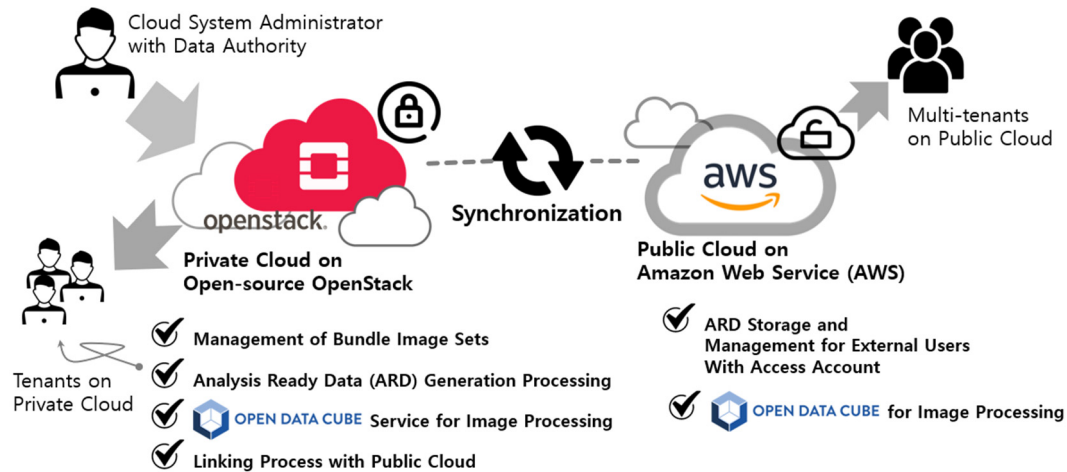


Figure 1. Conceptual diagram of a multi-cloud service based on the ODC platform.

2.3. System Configuration and Component Deployment

In this study, we propose a prototype model for a private cloud service, as shown in Figure 2. We directly build and use the cloud environment using OpenStack. We designed the system with scalability in mind by using a micro-service approach to build each function with separate virtual machines. The private cloud service consists of six independent virtual machines that allow users to upload and manage KOMPSAT images and process them simultaneously.

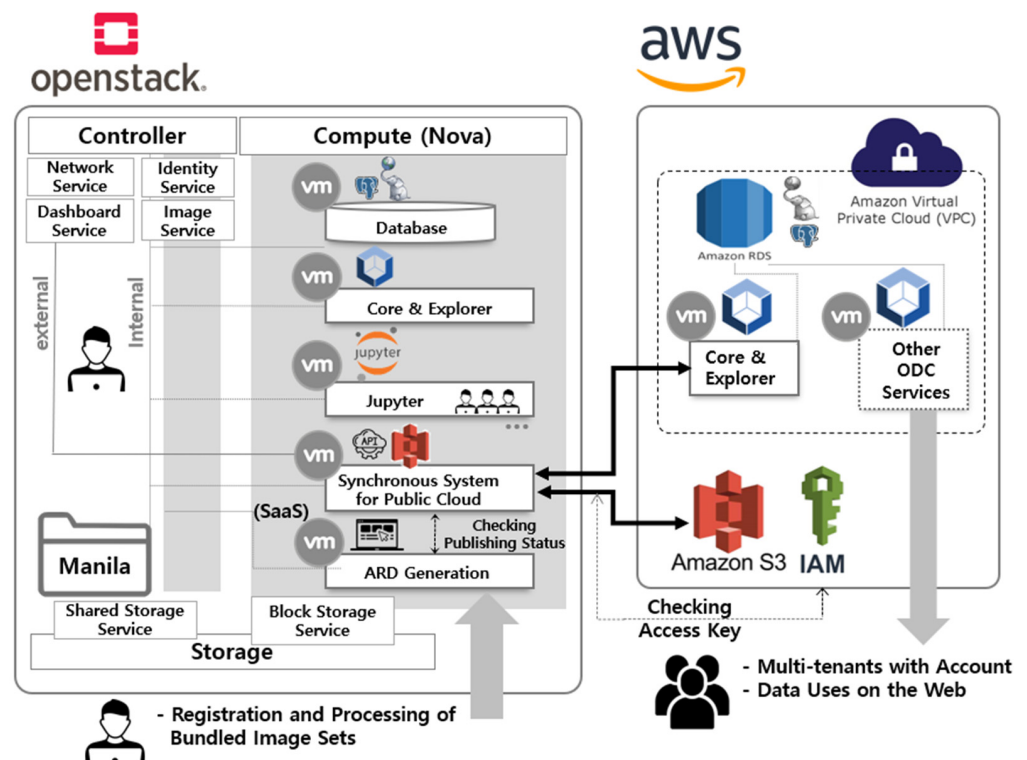


Figure 2. Microservice design of the multi-cloud environment for the integrated management and utilization of KOMPSAT based on ODC.

The Database service manages all data using PostgreSQL [43] and PostGIS [44] as relational database services. On the other hand, the Core & Explorer service stores and manages data in a time series format as an ODC platform service. The Jupyter service, which allows internal users to process and use the data in conjunction with ODC, provides each user with a virtual environment. These virtual machines can be replicated based on the initial build, allowing for rapid deployment to multiple users within the acceptable limits. The Shared Storage service is a file-sharing service provided within the private cloud, and satellite images are stored and shared with internal systems through communication that is disconnected from external systems. The ARD generation process service is a SaaS with modules for uploading and processing data. Data management users upload and manage data through SaaS, and the interface establishes data linkage with the public cloud. Finally, the Synchronous System links data to AWS S3 and requests data registration with ODC provided by the public cloud service. To create a unified system, the database and data must be provided in a shareable format over the internal network. Therefore, an OpenStack Self-Service Network was established to create an internal network that allows all services to communicate with each other.

The public cloud service provided by AWS offers various services and comprises four services. The Relational Database Service (RDS) and Core & Explorer services were configured similarly to the private cloud service, as ODC is also used in the public cloud. This study performed indexing data through Core & Explorer for simple web browsing, but the ODC platform supports additional functions. Other ODC services provide further explanation and offer various services that provide web-based time-series visualization and processing for external users. AWS provides a number of services that use the latest technology to store large amounts of data quickly and reliably. Among these services, S3 provides storage for large data such as LANDSAT and Sentinel. The ODC is designed for cloud-based services, enabling data management through object storage access. Therefore, this study utilized the storage and sharing method to store and share satellite images connected to the private cloud through S3. Only administrators who synchronize data through the AWS Identity and Access Management (IAM) service can access S3. As KOMPSAT data is not free like LANDSAT data, the service is limited to external users who download the data directly.

ODC Core is an open-source data cube framework for storing, managing, and analyzing large-scale geospatial data. It stores geospatial data in a multidimensional array format, allowing for fast retrieval and analysis. These multidimensional arrays are designed to match the characteristics of geospatial data with arbitrary dimensions such as x, y, time, and others. ODC Core supports various geospatial data formats, enabling efficient data management from different sources. It also supports a distributed computing architecture for processing large amounts of data. ODC Core is written in Python and uses various open-source libraries and frameworks such as NumPy, PostgreSQL, and Apache Spark. It is distributed under the Apache 2.0 license, allowing free use. ODC Core is used in various geospatial data analysis and visualization fields, such as disaster response, environmental monitoring, agriculture, and geographic information systems.

ODC Explorer is a visualization and analysis tool for the ODC database, an open-source data cube framework for storing and analyzing large-scale geospatial data. The ODC Explorer provides a web interface for users to visualize and manipulate various layers of the ODC database. Users can render ODC data on a map, adjust time series using a slider bar, filter the range of data, and query data using various analysis tools. It enables users to utilize ODC data in various fields, such as terrain analysis, crop detection, water resource management, and environmental monitoring. Additionally, the ODC Explorer can be easily expanded through a plugin system, allowing users to add new features and tools available through the web interface. The ODC Explorer is written in Python and JavaScript.

2.4. Applied Computing Resources

Table 1 presents a detailed description of the service created to test the potential of the model proposed in this study. For the private cloud, the Wallaby, the 22nd release of the OpenStack cloud computing platform released in April 2021, was used. OpenStack is an IaaS platform that combines several projects. It includes various enhancements and new features for IaaS cloud deployments, including improved security, networking, and integration with other open-source projects. OpenStack Wallaby aims to provide a more robust and secure cloud infrastructure platform while improving integration with other open-source tools and platforms.

Table 1. Service configuration and service components of the multi-cloud platform system.

System	Service	Name
OpenStack (Private Cloud)	Platform	Wallaby
	Controller	Identity Image Networking
	Dashboard	Horizon
	Compute	Compute
	Storage	Block storage Shared file systems
Amazon Web Service (Public Cloud)	Virtual machine Database Storage	EC2 RDS S3

In this study, we built three essential services for configuring IaaS: Controller, Compute, and Storage. The Controller is composed of authentication, operating system images for virtual machine operation, network, and access dashboard services. The Identity service of the Controller is responsible for authentication and authorization when operating all services, such as virtual machines and networks. Keystone is an OpenStack package that performs API client authentication. The Image service manages the operating system that virtual machines run on and can provide scalability by creating virtual machines that can be as snapshots. Glance is an image package that affords a suitable way to copy and launch instances. Besides, it allows users to upload, register, and retrieve virtual machine images easily and rapidly. The Networking service is responsible for communication with the cloud service. It can create external and internal communication services in the form of virtual networks, and in this study, we built a service that enables internal network configuration. Neutron provides functionalities for Networking as a Service (NaaS) between interface devices managed by other types of OpenStack services. It is a significant chunk of the OpenStack platform. The Dashboard is a service that allows users to check the status on the web and makes it easy to use. Horizon is a dashboard service with a graphical interface to access, provide, and automate cloud-based resources for administrators and users.

The Compute is a service that manages the space where virtual machines are created. There are two types of storage services: Nova is a cloud computing controller that is crucial in an IaaS. It allows users to create and manage virtual servers through machine images. Block storage is a service which creates space used as the main storage when virtual machines are operated. A Shared File System is a service that can share storage space on the network independently of virtual machines. Cinder provides a Block Storage as a Service (BaaS), which offers a persistent block-level storage device. It is responsible for managing the creation, attachment, and detachment of block devices to clusters. Manila is one of the projects in OpenStack that provides a file-sharing service. It offers a storage-sharing service that allows users to share files with other users and manage permissions for file sharing. It supports various storage backends and enables file sharing using protocols such as

Common Internet File System (CIFS) and Network File System (NFS). Additionally, Manila allows users to easily create, modify, and delete file shares through OpenStack Dashboard, Command-line Interface (CLI), and Representational State Transfer (REST) API. Using Manila makes it easier to manage file sharing with other OpenStack services. For example, users can mount Manila shares on Nova instances to access file systems. In addition, Manila uses Keystone for authentication and authorization, enabling consistent security policies to be applied across the OpenStack infrastructure. Manila is distributed under the Apache 2.0 license as one of OpenStack's projects. It provides an open-source-based file-sharing service and is very useful for setting up an environment for sharing large-scale data storage.

The AWS is a public cloud service with worldwide data centers offering more than 200 services. This study used four services at the Asia Pacific (Seoul) data center (ap-northeast-2): EC2, RDS, S3, and IAM. EC2 provides virtual machines with over 500 instances, offering the latest processors, storage, and more, and was used to build and deploy the ODC. The RDS database service provides fast storage and backup services. It offers representative DBMSs such as MySQL, PostgreSQL, MariaDB, and SQL Server for direct use, making it ideal as the database for the ODC. S3 is an object storage service that offers cutting-edge storage technology, which was used to store actual KOMPSAT data. IAM is a service used to manage authentication and authorization and was used to control access to KOMPSAT data.

The ODC is open-source software that manages and analyzes satellite images in various forms and was designed to be scalable. There are 45 related projects, such as core, web-based data retrieval, Open Geospatial Consortium Web Services (OWS) spatial standard visualization, and time-series analysis system (<https://github.com/orgs/pendatacube/repositories> (accessed on 26 June 2023)). In this study, we applied three projects: core, Jupyter-based ODC-linked data processing, and web-based data retrieval. The installation results of each module are shown in Table 2, and they can be built independently for flexible expansion. Private cloud virtual machines are all built on the Ubuntu server operating system to provide services. ODC Core & Explorer is a core feature, which is a service that allows users to quickly check data on the web.

Table 2. Detailed configuration of the integrated ODC environment for the multi-cloud platform.

Virtual Machine or Service Name		Applied Environment, Open-Source Projects, Execution Modules, and Version
OpenStack	Core & Explorer [VM]	Ubuntu server 20.04 LTS (cloud image) Open Data Cube Core v1.8.11
		Open Data Cube Explorer v2.10.0
	Database [VM]	Ubuntu server 20.04 LTS (cloud image) PostgreSQL 14
	SaaS of ARD [VM]	Ubuntu server 20.04 LTS (cloud image) OpenJRE 11.0.18 Apache Tomcat 9.0.73 GDAL 3.0.4 Apache HTTP Server 2.4
	Jupyter [VM]	Ubuntu server 20.04 LTS (cloud image) Anaconda 3 Python 3.8 Jupyter notebook server 1.18.1
	Synchronous [VM]	Ubuntu server 20.04 LTS (cloud image) aws-cli 1.27.74
	Manila [NFS Service]	Network File System (500 GB)

Table 2. Cont.

Virtual Machine or Service Name		Applied Environment, Open-Source Projects, Execution Modules, and Version
AWS	Core & Explorer [EC2 Service]	Ubuntu server 20.04 LTS (cloud image) Open Data Cube Core v1.8.11 Open Data Cube Explorer v2.10.0
	Database [RDS Service]	PostgreSQL 14
	Storage [S3 Service]	Bucket: KOMPSAT-public

SaaS of ARD is a separate service from ODC that allows users to upload and manage KOMPSAT data. This service is a cloud-based satellite image processing SaaS designed and built on Java, using OpenJRE for Java Runtime Environment, Tomcat, and Geospatial Data Abstraction Library (GDAL) [45] responsible for the real-time input/output of the satellite image. Jupyter is a web platform for writing and executing code in various programming languages, including Python. ODC supports access and processing of time-series data through the Python API. Therefore, internal users can search for and process the desired data by receiving the assigned machine. Finally, the Synchronous service connects to SaaS as a publicly configured service and synchronizes data on S3 through AWS-CLI. All virtual machines can be expanded in the future after optimization. In particular, the Jupyter service provides individual virtual machines for users to utilize immediately.

AWS has configured ODC Core and Explorer as virtual machine services and PostgreSQL as a shared resource, following the same approach as for a private cloud. S3 is used as the data-sharing resource. ODC is satellite image processing software designed for cloud services and supports accessing and processing time-series data via S3. The top-level directory name in the S3 bucket is set to 'KOMPSAT-public', and the data shared in the private cloud is synchronized within the bucket.

3. Implemented Results of Multi-Cloud Services

Figures 3 and 4 depict the steps for creating a proposed model using KOMPSAT-3/3A image ARD services. Figure 3 illustrates the process of uploading KOMPSAT data to a private cloud environment and synchronizing it with a public cloud environment. Meanwhile, Figure 4 shows how synchronized data from a private cloud was made available to users through the ODC framework environment in a public cloud.

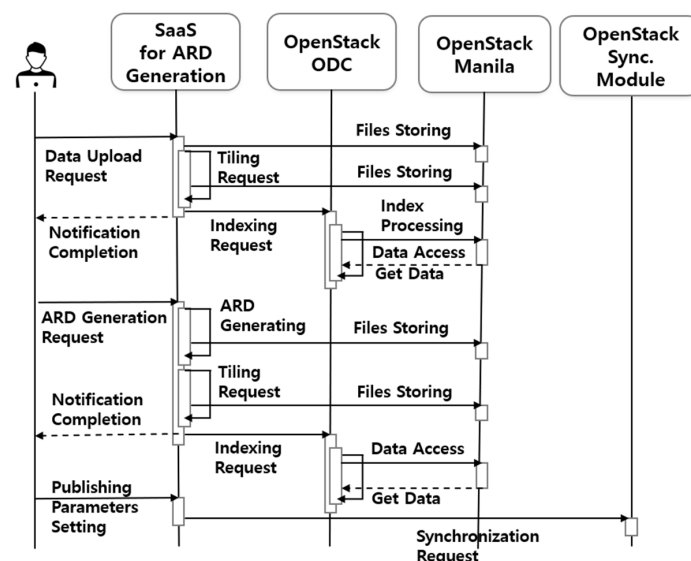


Figure 3. Sequence diagram for uploading KOMPSAT data to the private cloud and ODC indexing.

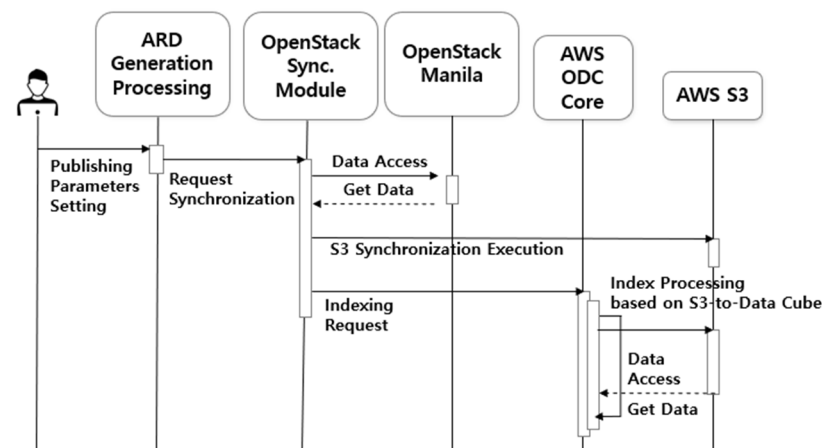


Figure 4. Sequence diagram for ODC indexing of KOMPSAT data in conjunction with public and private clouds.

3.1. Sequential Workflow of Multi-Cloud Services

Figure 3 depicts a sequence diagram of data uploading and management in the private cloud, with the data manager as the actor. The data manager first uploads KOMPSAT-3/3A Level-1G data through the SaaS system, which stores the original data in the Manila service storage. Then, the Tiling process is performed for web mapping visualization, with processing results sent to the system as an alert when completed. Simultaneously, the ODC indexing process registers the data in the ODC Core system, which already has a module for indexing through the auxiliary files of the KOMPSAT series. The SaaS system includes a calibration function that processes TOAR/TOCR and stores them in Manila. Finally, the data manager can select data for public access through the SaaS management function. The synchronization module will synchronize data to AWS S3 once the public setting is changed to ‘On’.

It describes the process after sharing settings are made in the private cloud SaaS, as shown in the sequence diagram in Figure 4. AWS ODC Core has a separate system for external users, managed through the private cloud Synchronous module. The synchronization process is initiated once the sharing settings are made on the private cloud SaaS management page. When synchronization is performed through the sharing settings, the synchronous system can communicate with AWS S3 because it is connected to an external network, and accepted data is synchronized. After synchronization, the KOMPSAT data processed by the SaaS is stored in the API S3. At the same time, the synchronous system requests indexing from the AWS ODC core service. Access to the data stored in S3 is required to perform indexing, which differs from the block storage method used for storing KOMPSAT data in the private cloud. However, since the ODC project is software that already takes the cloud into account, it provides a Python API for indexing through S3. In this study, we installed this API in the AWS ODC Core service to process it. The processed data can be checked through Explorer, and additional ODC extension projects can be added to provide them to external users.

3.2. Implemented Results

Figures 5 and 6 illustrate the operational results of the utilization model presented in this study. The interface for processing ARD data and sharing settings in SaaS is shown in Figure 5. Step 1 in Figure 5 displays the screen for uploading KOMPSAT-3A bundled data. SaaS consists of a web mapping interface, and users can upload auxiliary files in Blue, Green, Red, Near Infrared, and Extensible Markup Language (XML) using the Data Adding function. Internal users perform ODC indexing for the stored bundled data to enable ODC utilization. The SaaS utilized in this study offers various internal/external satellite processing algorithms. It includes atmospheric and surface reflectance processing

functions, as shown in Step 2 of Figure 5. The processed results are also visualized for each blue, green, red, and near infrared band, and ODC indexing is performed for the processed data, similar to the bundled data. Step 3 displays the indexed results via the ODC Explorer, which allows users to view data registered in the ODC on the web, including bundled images (KOMPSAT3A_Level-1G), atmospheric reflectance images (KOMPSAT3A_TOAR), and surface reflectance images (KOMPSAT3A_TOCR). By uploading and processing data in the private cloud, internal users can utilize this data via tools such as Jupyter Notebook. In Step 4, a data-sharing settings page is designed to make data available to external users as a public cloud service to external users. Users can choose to make uploading data (Data List) or processed data (Process List) public by setting the 'is public' option to 'On'. Once set, the data is copied to the public cloud service via an internal synchronization system.

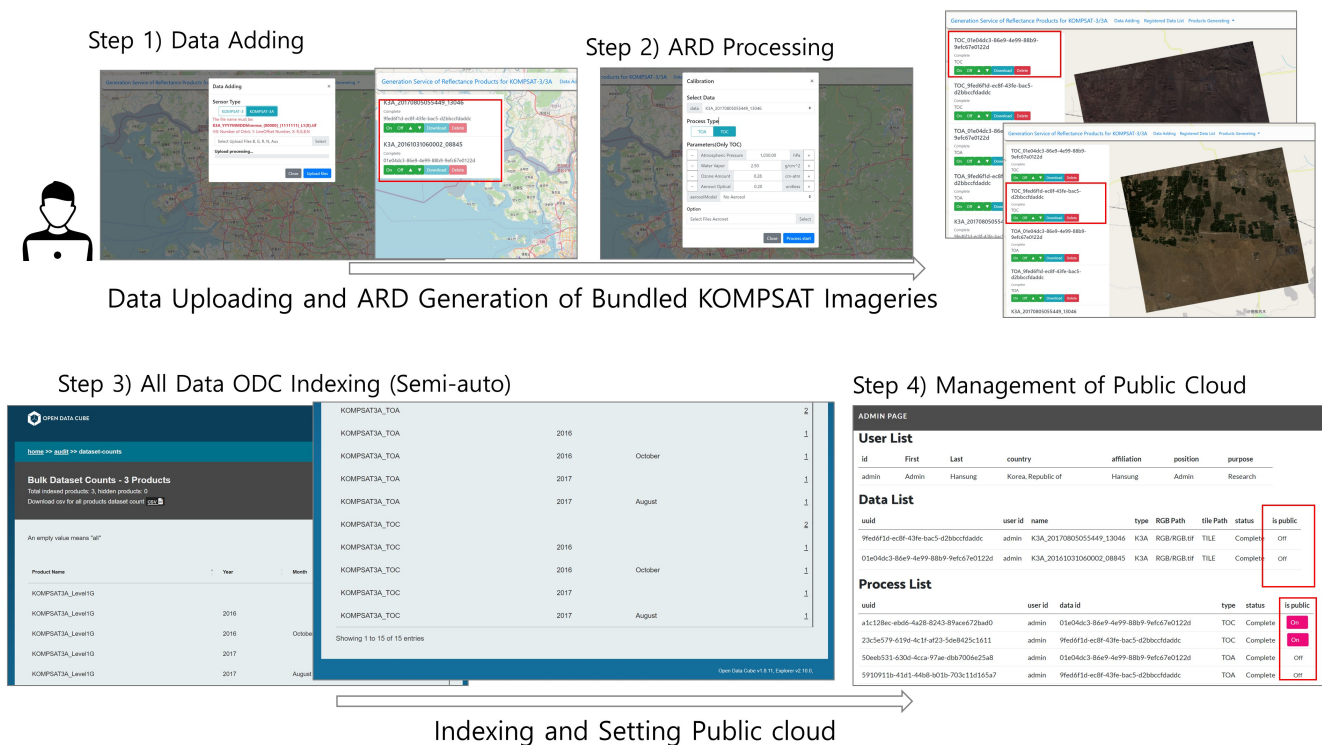


Figure 5. Examples of the step processes on ODC and data sharing settings for private cloud KOMPSAT bundled data. Base map is OpenStreetMap [46] in Korean.

Figure 6 shows data synchronization to AWS S3 when the 'is public' option is set to 'On' in Step 4 of Figure 5. The synchronous system in the private cloud utilizes the S3 API to copy data, and selecting 'Off' deletes the data. The KOMPSAT-public bucket is created in the ap-northeast-2 region of Amazon S3 to store KOMPSAT data. However, KOMPSAT data is not freely accessible, so only data managers have permission to access it through IAM, and the synchronous system proceeds with synchronization once authenticated. The Yet Another Markup Language (YAML) file required for ODC indexing is also synchronized during this process. Finally, synchronization involves copying the data to S3. Once completed, the synchronous system requests indexing processing from AWS ODC Core to add KOMPSAT data to the final public cloud ODC.

Figure 7 displays the resulting process, where the synchronous system requests indexing from AWS ODC Core and Python S3 to DC on the AWS ODC Core virtual machine performs indexing. Figure 7 also confirms that the two TOC data sets that were set to public in the private cloud are served from the location of S3 (s3://kompsat-public (accessed on 26 June 2023)).

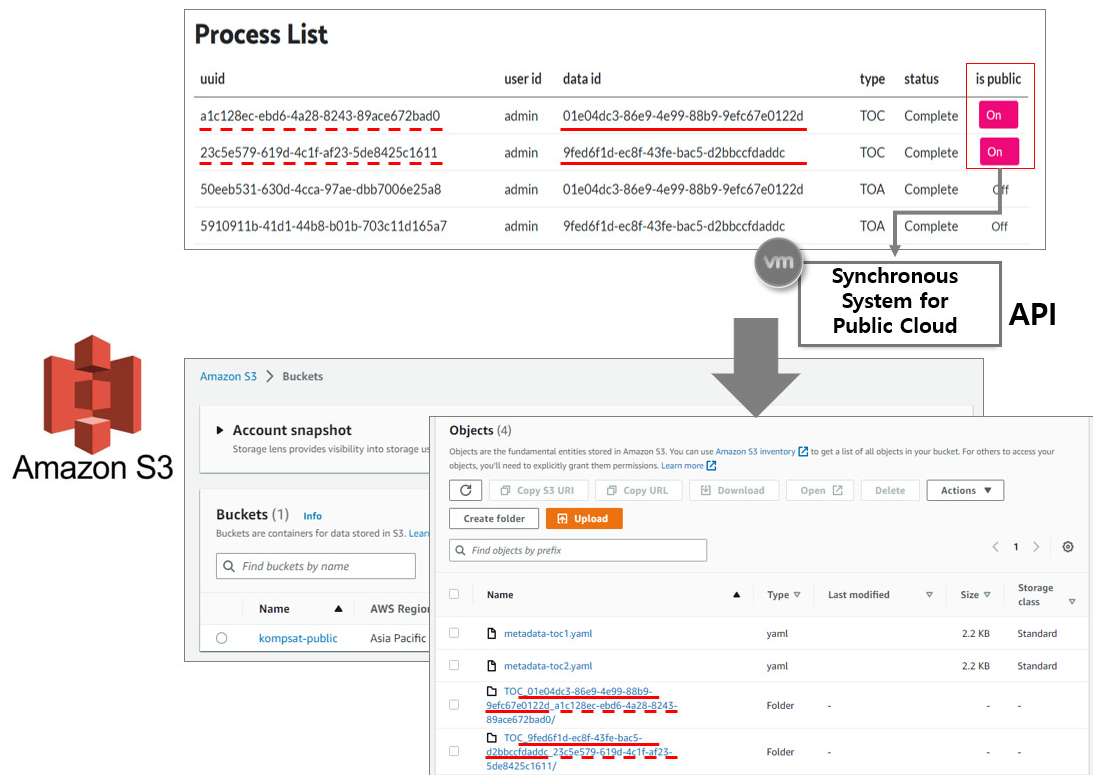


Figure 6. Results of synchronization of KOMPSAT-3/3A reflectance data on AWS S3.

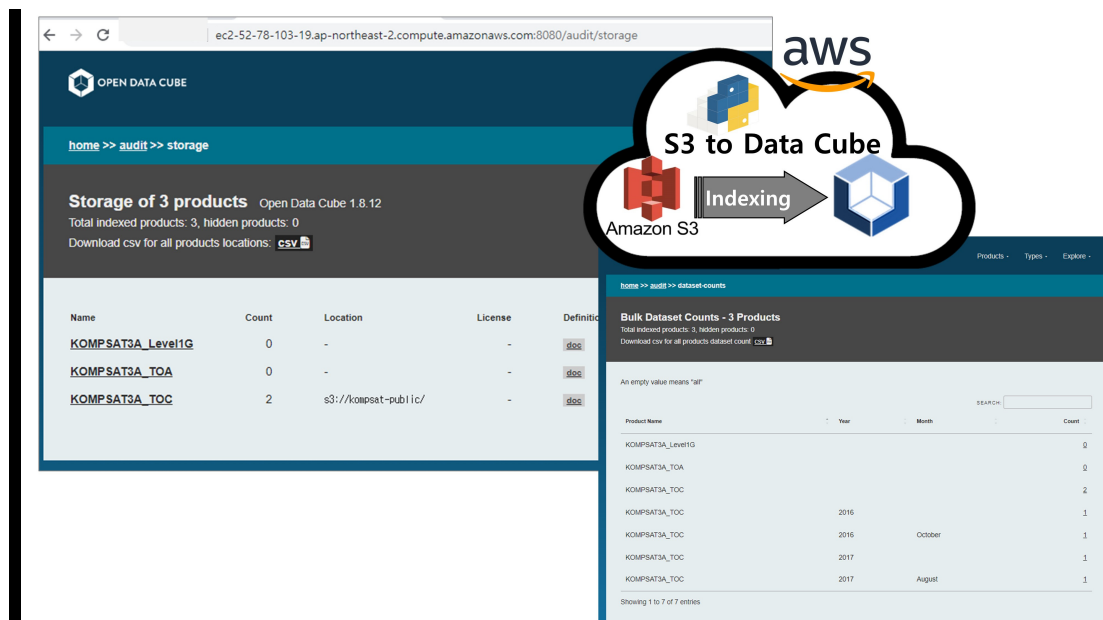


Figure 7. Web search results for ODC KOMPSAT TOCR data indexing on AWS cloud.

3.3. Application Cases Using ODC UI

The ODC Services in the architecture diagram shown in Figure 2 refer to various projects that utilize ODC. These include the ODC Explorer for data exploration, the Web Map Service (WMS), the Web Map Tile Service (WMTS), and Web Catalog Service (WCS), which are handled by the ODC Open Web Services (OWS), as well as the ODC UI, which is capable of processing spatiotemporal Network Common Data Form (netCDF) data. This study presents the registration of processed TOAR/TOCR data in ODC UI and showcases

the processing results. In ODC UI, the ingesting process incorporated Level-1G and TOCR data for processing. Direct processing of the ingested data in ODC UI is not possible. The functionalities integrated into the UI are called ‘apps’ and are designed as independent modules. Registration is carried out through the administrator page, where the content of Satellites, Area, Area products maps, and Applications, as depicted in Figure 8, is registered for use. Specifically, it is necessary to register the data with the apps responsible for the processing in the Applications section. Figure 8a illustrates the configuration of the region. Figure 8b,c represent the area for each function and the satellite image settings, respectively.

Figure 8 displays three user interfaces for ODC data processing:

- (a) Change area:** A form for configuring a region. It includes fields for ID (baotou-data), Name (Baotou), Latitude min (40.805083), Latitude max (40.952725), Longitude min (105.470735), Longitude max (106.589019), Thumbnail imagery (/static/assets/images/black.png), and Satellites (KOMPSAT3A_TOCR, KOMPSAT3A_Level1G).
- (b) Change application:** A form for configuring an application. It includes fields for ID (spectral_indices), Name (Spectral Indices), and Areas (baotou-data). It also has a Satellites field with KOMPSAT3A_TOCR and KOMPSAT3A_Level1G.
- (c) Change satellite:** A form for configuring a satellite. It includes fields for Database platform (KOMPSAT3A_TOCR), Name (KOMPSAT3A_TOCR), Date min (2018-01-01), Date max (2017-12-31), Data min (0.0), Data max (1.0), Measurements (blue.green.red.nv), No data value (0.0), Platform (KOMPSAT3A_TOCR), Collection (c1), and Level (TOCR).

Figure 8. User interfaces for ODC data processing: (a) Areas; (b) Application type; and (c) Satellite sensor type.

Figure 9 presents the results of normalized difference vegetation index (NDVI) processing using the registered data and designated areas with sample image sets. Figure 9a shows the outcome of the ingested dataset. It displays the registered data areas on top of an OpenStreetMap background map using web mapping. It shows all ingested data in ODC. ODC UI is equipped with eight core functionalities, including five in the land category and three in the water category. It is built using the Python Django framework, allowing for the addition of user data processing capabilities in the future. Figure 9b represents the user interface of the spectral indices processing page. Users can select the pre-registered data through the Satellite option and choose the spectral indices. They have the option to select from seven different processing options, including NDVI, normalized difference water index (NDWI) to estimate the leaf water content at a canopy level, and enhanced vegetation index (EVI) to quantify vegetation greenness. By selecting the processing area on the map through geospatial bounds and specifying the time range, users can monitor the processing status asynchronously as shown in Figure 9c. Once the processing with KOMPSAT 3A TOCR is complete, the results of NDVI are displayed within the predefined color scale from -1.0 to 1.0 , as shown in Figure 9d. Figure 9e,f show the results of EVI and NDWI processing, respectively, in this ODC prototyping model. When KOMPSAT ARDs of multiple time periods are built into this system, it will be possible to perform time series analysis using these high-resolution image sets.

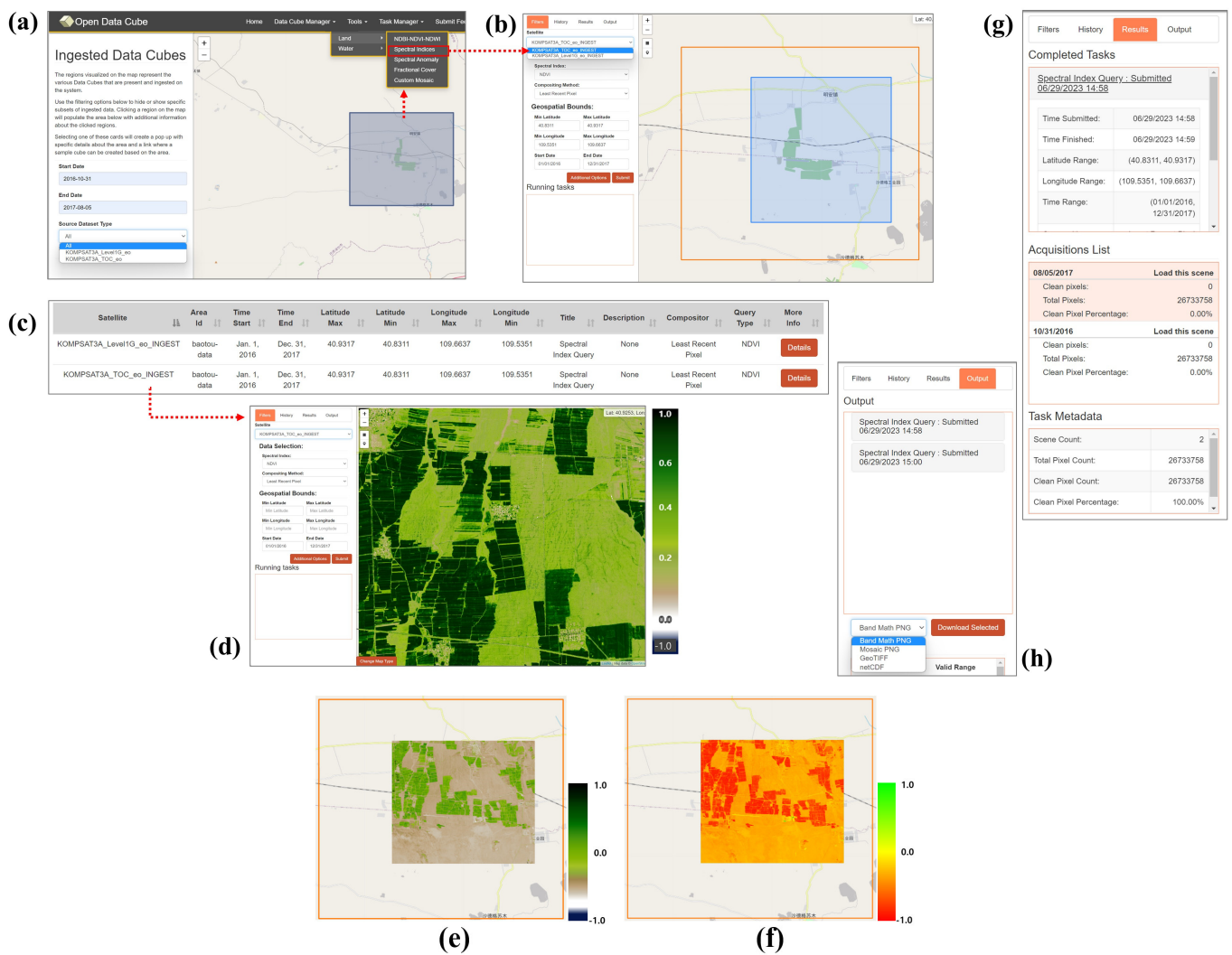


Figure 9. Processing results of registered data visualization, spectral indices using KOMPSAT-3A TOCR, and downloading tab: (a) Web mapping of ingested data; (b) Interface for spectral indices function selection; (c) Interface for checking processing status; (d) Processing results of NDVI; (e) Processing results of EVI; (f) Processing results of NDWI; (g) Information on processing results in the result tab; and (h) Downloading processing results in the output tab.

Furthermore, the processed results can be accessed through the Results and Output tabs for various information and data downloads. In the Results tab Figure 9g, users can review the processing area and duration, as well as the list of processed data. Additionally, in the output tab Figure 9h, data can be downloaded in PNG, GeoTiff, or netCDF format.

4. Discussion and Conclusions

There are various data cube services being developed or operated based on the data cube paradigm. However, each data cube service differs in terms of target regions, scope of data provision, and types of satellite images provided. The data processing capabilities and sizes of archived ARD are also different. Currently, most ODC-based services utilize the AWS cloud, but the underlying cloud environment for the data cube can naturally vary due to these factors. This study presents a prototype model that can be implemented using a multi-cloud approach.

The demand for ARD products for satellite imagery is increasing because users can use the data directly without additional calibration work. ARD and data cubes are not separate entities, but it is desirable to handle them together because data cube systems

based on cloud computing environments are the best way to utilize ARD. In this study, we proposed an application model that can handle ARD in a multi-cloud environment. Multi-cloud services refer to the use of several cloud services in combination.

OpenStack enables the virtualization of computing, storage, and networking resources to build a private cloud, while AWS offers a range of cloud services to build a public cloud. Companies or organizations can benefit from improved security, efficiency, and flexibility by configuring a hybrid cloud architecture that combines OpenStack and AWS. Although these platforms serve different purposes, they can be used together to build a powerful hybrid cloud. AWS and OpenStack are platforms that provide cloud computing services, but each has its own characteristics, advantages, and disadvantages. AWS provides high availability and stability while providing an easy and user-friendly experience. While OpenStack is a platform that allows users to build a personalized cloud environment by combining components.

There are several advantages to using these two platforms together. First, there is flexibility. Each AWS and OpenStack offer unique functionalities. AWS has predefined functionalities, making it easier for users to get started, while OpenStack allows users to freely configure and customize functionalities. Using them together can combine flexible functionality configuration with ease of use. OpenStack is an open-source project that allows customers to configure a personalized cloud environment as they want. AWS provides public cloud services. A private cloud environment can be configured by using these two platforms together. Second, there are cost savings: AWS rents out large-scale infrastructure, so the initial cost is very high. OpenStack is open-source, so the initial cost is low. Therefore, by using both platforms together, users can build a cloud infrastructure while reducing the initial costs. Third, there is stability. AWS provides very high availability and stability. By using these two platforms together, the stability can be increased.

On the other hand, using AWS and OpenStack together can have the following disadvantages or issues. The first is complexity. The two systems may have compatibility and integration issues when using AWS and OpenStack together. These issues can make configuration and management difficult. Second, because OpenStack is an open-source project, customer and technical support may need to be improved. This can make it difficult to resolve problems when they arise. Third, there are security issues. AWS provides strict control and auditing for security. However, since OpenStack is an open-source, security vulnerabilities may be discovered. It can lead to potential security issues.

This study focused on KOMPSAT-3/3A satellite images, which can also be applied to Landsat, Sentinel, and other sensor data without any problems. The performance of the processing speed of the multi-cloud system proposed in this study may vary depending on the physical computing environment in which the system is implemented. It will perform well even if the same system design content is built in a high-performance computing environment. In a low-level computing environment, the processing speed will naturally slow down.

Author Contributions: Conceptualization, K.L.; methodology, K.L.; software, K.L. and K.K.; investigation and analysis, K.L.; writing, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the Hansung University for Kiwon Lee. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. RS-2023-00241346) for Kwangseob Kim.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Choosing the Right Cloud Service: IaaS, PaaS, or SaaS. Available online: <https://rubygarage.org/blog/iaas-vs-paas-vs-saas> (accessed on 26 June 2021).
2. Yang, C.; Huang, Q. *Spatial Cloud Computing: A Practical Approach*; CRC Press: Boca Raton, FL, USA, 2014.
3. Wang, L.; Yan, J.; Ma, Y. *Cloud Computing in Remote Sensing*; CRC Press: Boca Raton, FL, USA, 2020.
4. Yue, P.; Zhou, H.; Gong, J.; Hu, L. Geoprocessing in cloud computing platforms—A comparative analysis. *Int. J. Digit. Earth* **2013**, *6*, 404–425. [\[CrossRef\]](#)
5. Yan, J.; Ma, Y.; Wang, L.; Choo, K.-K.R.; Jie, W. A cloud-based remote sensing data production system. *Future Gener. Comput. Syst.* **2018**, *86*, 1154–1166. [\[CrossRef\]](#)
6. Yao, X.; Li, G.; Xia, J.; Ben, J.; Cao, Q.; Zhao, L.; Ma, Y.; Zhang, L.; Zhu, D. Enabling the Big Earth Observation Data via Cloud Computing and DGGS: Opportunities and Challenges. *Remote Sens.* **2020**, *12*, 62. [\[CrossRef\]](#)
7. Huang, W.; Zhou, J.; Zhang, D. On-the-Fly Fusion of Remotely-Sensed Big Data Using an Elastic Computing Paradigm with a Containerized Spark Engine on Kubernetes. *Sensors* **2021**, *21*, 2971. [\[CrossRef\]](#)
8. Kline, K. USGS Landsat Migration to the Cloud, Presentation Material in CEOS WGISS-49 Meeting. 2021. Available online: <https://ceos.org/meetings/wgiss-51/> (accessed on 26 June 2023).
9. Astsatryan, H.; Lalayan, A.; Giuliani, G. Scalable Data Processing Platform for Earth Observation Data Repositories. *Scalable Comput. Pract. Exper.* **2023**, *24*, 35–44. [\[CrossRef\]](#)
10. Shaharum, N.S.N.; Shafri, H.Z.M.; Ghani, W.A.W.A.K.; Samsatli, S.; Prince, H.M.; Yusuf, B.; Hamud, A.M. Mapping the spatial distribution and changes of oil palm land cover using an open-source cloud-based mapping platform. *Int. J. Remote Sens.* **2019**, *40*, 7459–7476. [\[CrossRef\]](#)
11. Antunes, R.R.; Blaschke, T.; Tiede, D.; Bias, E.S.; Costa, G.A.O.P.; Happ, P.N. Proof of concept of a novel cloud computing approach for object-based remote sensing data analysis and classification. *GIScience Remote Sens.* **2019**, *56*, 536–553. [\[CrossRef\]](#)
12. Zou, Q.; Li, G.; Yu, W. Cloud Computing Based on Computational Characteristics for Disaster Monitoring. *Appl. Sci.* **2020**, *10*, 6676. [\[CrossRef\]](#)
13. Abdelmajeed, A.Y.A.; Albert-Saiz, M.; Rastogi, A.; Juszczak, R. Cloud-Based Remote Sensing for Wetland Monitoring—A Review. *Remote Sens.* **2023**, *15*, 1660. [\[CrossRef\]](#)
14. Dwyer, J.L.; Roy, D.P.; Sauer, B.; Jenkerson, C.B.; Zhang, H.K.; Lymburner, L. Analysis Ready Data: Enabling Analysis of the Landsat Archive. *Remote Sens.* **2018**, *10*, 1363. [\[CrossRef\]](#)
15. United States Geological Survey (USGS). G.S. Landsat Collection 1(C1) Analysis Ready Data (ARD) Data Format Control Book (DFCB). Available online: <https://www.usgs.gov/media/files/landsat-collection-1-us-analysis-ready-data-dfcb> (accessed on 20 May 2023).
16. United States Geological Survey (USGS). G.S. Landsat Collection 2(C2) Analysis Ready Data (ARD) Data Format Control Book (DFCB). Available online: <https://www.usgs.gov/media/files/landsat-collection-2-us-analysis-ready-data-dfcb> (accessed on 20 May 2023).
17. Earth Engine Data Catalog. Available online: <https://developers.google.com/earth-engine/datasets/catalog> (accessed on 26 June 2023).
18. Zhao, Q.; Yu, L.; Li, X.; Peng, D.; Zhang, Y.; Gong, P. Progress and Trends in the Application of Google Earth and Google Earth Engine. *Remote Sens.* **2021**, *13*, 3778. [\[CrossRef\]](#)
19. Yang, L.; Driscoll, J.; Sarigai, S.; Wu, Q.; Chen, H.; Lippitt, C.D. Google Earth Engine and Artificial Intelligence (AI): A Comprehensive Review. *Remote Sens.* **2020**, *14*, 3253. [\[CrossRef\]](#)
20. CEOS Analysis Ready Data. Available online: <https://ceos.org/ard/> (accessed on 26 June 2023).
21. Kopp, S.; Becker, P.; Doshi, A.; Wright, D.J.; Zhang, K.; Xu, H. Achieving the Full Vision of Earth Observation Data Cubes. *Data* **2019**, *4*, 94. [\[CrossRef\]](#)
22. Giuliani, G.; Egger, E.; Italiano, J.; Poussin, C.; Richard, J.-P.; Bruno Chatenoux, B. Essential Variables for Environmental Monitoring: What Are the Possible Contributions of Earth Observation Data Cubes? *Data* **2020**, *5*, 100. [\[CrossRef\]](#)
23. Sudmanns, M.; Augustin, H.; van der Meer, L.; Baraldi, A.; Tiede, D. The Austrian Semantic EO Data Cube Infrastructure. *Remote Sens.* **2021**, *13*, 4807. [\[CrossRef\]](#)
24. Euro Data Cube A Data Cube Service as Earth Observation Factory. Available online: <https://eo4society.esa.int/resources/euro-data-cube/> (accessed on 26 June 2023).
25. Killough, B.; Cherry, A.; Gowda, S. Lessons Learned and Cost Analysis of Hosting a Full Stack Open Data Cube (ODC) Application on the Amazon Web Services (AWS). In Proceedings of the 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018. [\[CrossRef\]](#)
26. Killough, B.; Rizvi, S.; Lubawy, A. Advancements in the Open Data Cube and the Use of Analysis Ready Data in the Cloud. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019. [\[CrossRef\]](#)
27. Killough, B.; Lubawy, A.; Dyke, G. The Open Data Cube Sandbox: A Tool to Support Flood Disaster Response and Recovery. In Proceedings of the 2022 IEEE International Geoscience and Remote Sensing Symposium, Virtual Symposium, 26 September–2 October 2020. [\[CrossRef\]](#)

28. Ferreira, K.R.; Queiroz, G.R.; Vinhas, L.; Marujo, R.F.B.; Simoes, R.E.O.; Picoli, M.C.A.; Camara, G.; Cartaxo, R.; Gomes, V.C.F.; Santos, L.A.; et al. Earth Observation Data Cubes for Brazil: Requirements, Methodology and Products. *Remote Sens.* **2020**, *12*, 4033. [CrossRef]
29. Ferreira, K.R.; Queiroz, G.R.; Marujo, R.F.B.; Raphael, W.; Costa, R.W. Building Earth Observation Data Cubes On AWS. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *43*, 597–602. [CrossRef]
30. ALOS-2 PALSAR-2 ScanSAR Products. Available online: https://www.eorc.jaxa.jp/ALOS/en/dataset/palsar2_l22_e.htm (accessed on 26 June 2023).
31. Digital Earth Africa Sentinel-1 Radiometrically Terrain Corrected. Available online: <https://registry.opendata.aws/deafrica-sentinel-1/> (accessed on 26 June 2023).
32. Yuan, F.; Repse, M.; Leith, A.; Rosenqvist, A.; Milcinski, G.; Moghaddam, N.F.; Dhar, T.; Burton, C.; Hall, L.; Jorand, C.; et al. An Operational Analysis Ready Radar Backscatter Dataset for the African Continent. *Remote Sens.* **2022**, *14*, 351. [CrossRef]
33. Cao, Q.; Li, G.; Yao, X.; Jia, T.; Yu, G.; Zhang, L.; Xu, D.; Zhang, H.; Shan, X. GF-1 Satellite Imagery Data Service and Application Based on Open Data Cube. *Appl. Sci.* **2022**, *12*, 7816. [CrossRef]
34. Cao, Q.; Li, G.; Yao, X.; Ma, Y. China Data Cube (CDC) for Big Earth Observation Data: Practices and Lessons Learned. *Information* **2022**, *13*, 407. [CrossRef]
35. Access Sentinel 2 Data from AWS. Available online: <https://odc-stac.readthedocs.io/en/latest/notebooks/stac-load-e84-aws.html> (accessed on 26 June 2023).
36. The Most Widely Deployed Open-source Cloud Software in the World. Available online: <https://www.openstack.org/> (accessed on 26 July 2022).
37. Earthdata Cloud Evolution. Available online: <https://www.earthdata.nasa.gov/eosdis/cloud-evolution> (accessed on 26 June 2023).
38. EODC Cloud Launcher. Available online: <https://eodc.eu/services/cloudlauncher/> (accessed on 26 June 2023).
39. Semlali, B.-E.B.; Freitag, F. SAT-Hadoop-Processor: A Distributed Remote Sensing Big Data Processing Software for Earth Observation Applications. *Appl. Sci.* **2021**, *11*, 10610. [CrossRef]
40. Lee, K.; Kang, S.; Kim, K.; Chae, T.-B. Cloud-based Satellite Image Processing Service by Open-source Stack: A KARI Case. *Korean J. Remote Sens.* **2017**, *33*, 339–350. [CrossRef]
41. Lee, K.; Kim, K. A Performance Evaluation of a Geo-Spatial Image Processing Service Based on Open-source PaaS Cloud Computing Using Cloud Foundry on OpenStack. *Remote Sens.* **2018**, *10*, 1274. [CrossRef]
42. Kim, K.; Lee, K. An Implementation of Open-source-Based Software as a Service (SaaS) to Produce TOA and TOC Reflectance of High-Resolution KOMPSAT-3/3A Satellite Image. *Remote Sens.* **2021**, *13*, 4550. [CrossRef]
43. PostgreSQL. PostgreSQL: The World's Most Advanced Open Source Relational Database. Available online: <https://www.postgresql.org/> (accessed on 9 March 2023).
44. PostGIS. About PostGIS. Available online: <http://postgis.net/> (accessed on 9 March 2023).
45. GDAL. GDAL—Geospatial Data Abstraction Library; 2.2.2; Open-Source Geospatial Foundation: Chicago, IL, USA, 2017; Available online: <http://www.gdal.org> (accessed on 9 March 2023).
46. OpenStreetMap. Available online: <https://www.openstreetmap.org> (accessed on 1 July 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.