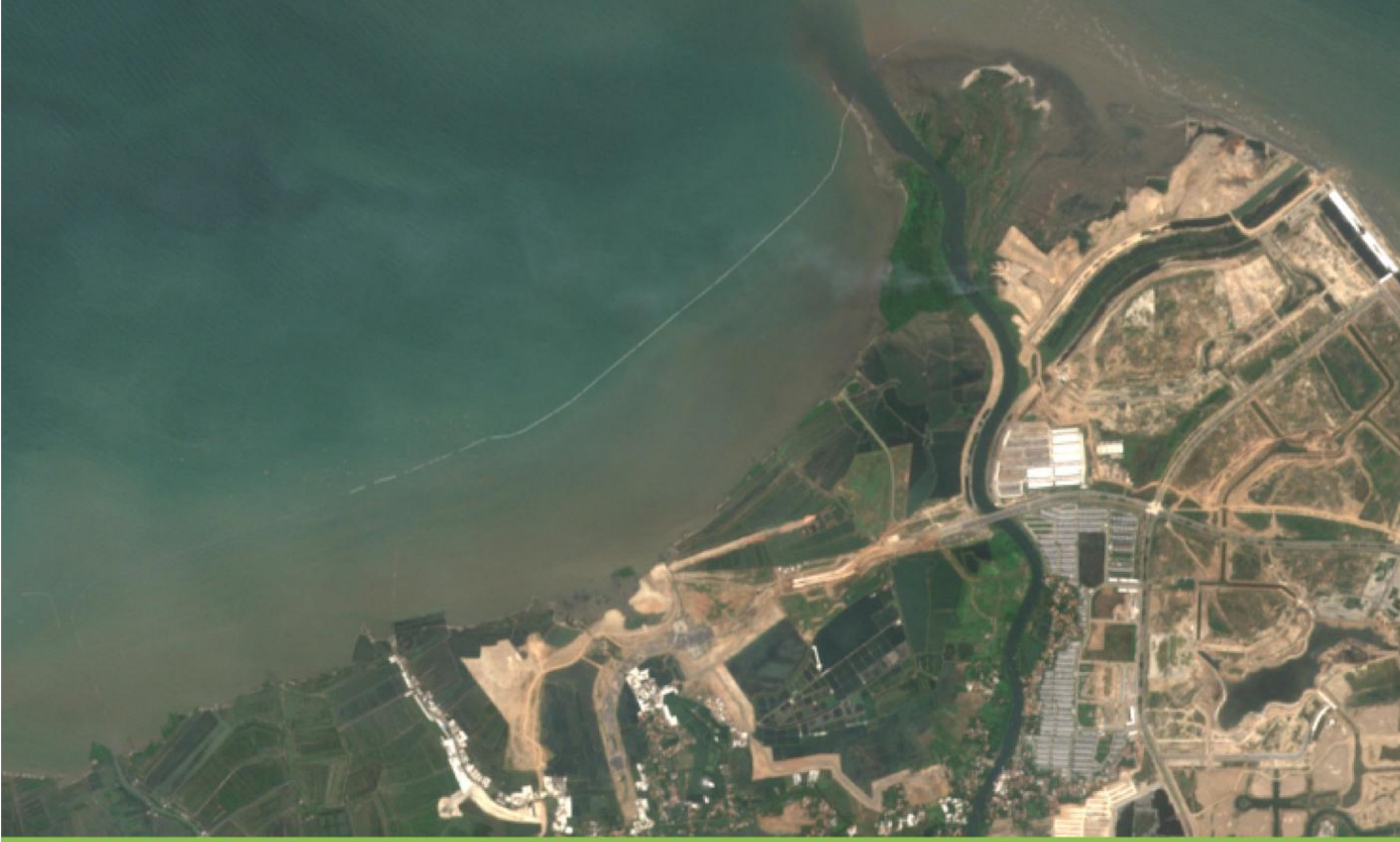

PIKSEL PROJECT PLAN

Prepared by

Badan Informasi Geospasial & Geoscience Australia

piksel

BADAN INFORMASI GEOSPASIAL

Australian Government
Geoscience Australia

Version 1.2
Apr 23, 2025

CONTENTS

EXECUTIVE SUMMARY

A cloud-based platform called Piksel is designed and developed to harness Earth Observation (EO) data particularly satellite imagery to address various challenges in Indonesia. Piksel provides a standardized, accessible, and scalable solution for processing, analyzing, and visualizing large volumes of satellite data. The objectives of Piksel are to support government decision-making, empower businesses and industries, advance research and innovation, and ensure open and scalable access to EO data.

Piksel is developed by collaboration between Badan Informasi Geospasial (BIG) and Geoscience Australia (GA). GA has succeded in developing and maintaining Digital Earth Australia and Digital Earth Africa. GA will assist the development of the system. The cloud will be implemented at Amazon Web Service (AWS) Indonesia. The potential users of Piksel are government institutions, university, and private sector.

Piksel will enhance food security by enabling real-time crop monitoring while supporting sustainable land use and urban planning. It will improve coastal and marine resource management by tracking ecosystems like mangroves and coral reefs, and aid disaster management through early warnings and damage assessments. The system will also support environmental protection by monitoring deforestation, and carbon sequestration. By providing data-driven insights, Piksel will empower policymakers, foster economic growth, and strengthen Indonesia’s resilience to climate change and natural disasters.

1. PROJECT OVERVIEW

1. Purpose

Piksel is a cloud-based platform designed to harness Earth observation (EO) data—particularly satellite imagery—to address various challenges in Indonesia. aims to provide a standardized, accessible, and scalable solution for processing, analyzing, and visualizing large volumes of satellite data. By leveraging open-source technologies such as the SpatioTemporal Asset Catalog (STAC) and Open Data Cube, it ensures interoperability and ease of access for stakeholders across government, industry, and research sectors.

Currently, stakeholders face significant challenges in utilizing EO data, including:

- Data accessibility issues: The large volume and complexity of satellite imagery make it difficult to access and use effectively.
- Inconsistent formats: Different data sources provide imagery in various formats, complicating integration and analysis.
- Processing complexity: Transforming raw satellite data into meaningful insights requires significant technical expertise and infrastructure.

Pixel addresses these challenges by offering a comprehensive platform that enables efficient data ingestion, processing, and visualization, allowing users to monitor environmental changes, track land use, support disaster management, and improve agricultural productivity.

This initiative is a collaboration between Badan Informasi Geospasial (BIG) and Geoscience Australia (GA), funded by the Marine Resources Initiative (MRI). Inspired by the success of Digital Earth Australia and Digital Earth Africa, Piksel leverages their experience and best practices to build a robust EO data platform for Indonesia.

2. Objectives

Figure 1. Objectives of Piksel

The are four primary objectives of Piksel development as follows:

2.1. Support Government Decision-Making

- Enhance the efficiency and effectiveness of government programs and policies by providing high-quality, timely EO data.
- Enable better resource allocation, environmental monitoring, and disaster response capabilities.

2.2. Empower Businesses and Industries

- Provide open and reliable access to EO data to improve operational efficiency, productivity, and innovation.
- Support sectors such as agriculture, forestry, marine resources, and urban planning with actionable insights.

2.3. Advance Research and Innovation

- Facilitate research institutions and universities in conducting geospatial analysis for various scientific studies.

- Enable collaborative research efforts with an open-data approach to address national and global challenges.
- Leverage EO data for coastal monitoring, marine resource management, and environmental protection. This platform will serve as the main tool for creating intertidal coastlines products, which will be one of the primary outputs of the collaboration.

2.4. Ensure Open and Scalable Access to EO Data

- Standardize and simplify access to EO data through open-source technologies, ensuring interoperability across platforms.
- Develop user-friendly tools to visualize, analyze, and download data effortlessly.

3. Key Components

The Piksel platform consists of several core components that enable users to analyze, visualize, and interact with Earth observation (EO) data efficiently:

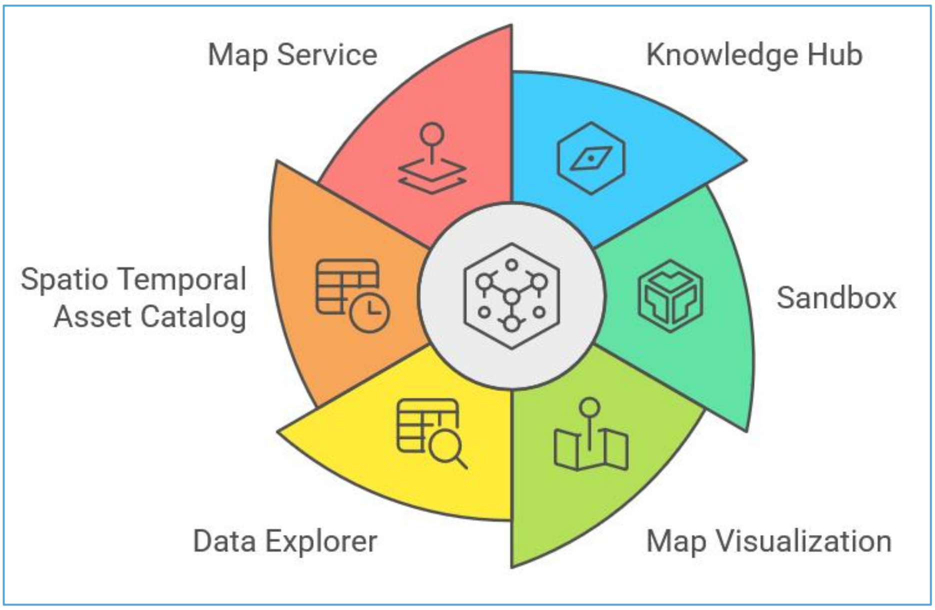

Figure 2. Component of Piksel

3.1. Knowledge Hub

A centralized platform for accessing educational and informational resources related to Earth observation and geospatial data. It provides tutorials, case studies, documentation, and other learning materials to support users.

3.2. Sandbox (Cloud-based Analysis Environment)

The sandbox will be built using **Docker containers** and **Kubernetes**, taking advantage of cloud computing capabilities such as autoscaling and resource management. This environment will provide users with dedicated CPU and RAM resources, allowing them to analyze EO products without requiring local infrastructure. The sandbox will support popular programming languages such as Python, enabling seamless integration with geospatial libraries and data science workflows.

3.3. Piksel Maps (Visualization Platform)

Piksel Maps will be powered by **Terria.IO**, a platform designed for browser-based geospatial data exploration and visualization. This will allow users to interact with vast amounts of geospatial data directly through their web browsers.

3.4. Data Explorer

Data Explorer will provide an intuitive interface for browsing available EO products, visualizing metadata, and understanding product coverage. Users can search, filter, and preview data before analysis.

3.5. STAC (SpatioTemporal Asset Catalog)

Piksel will implement STAC standards to ensure interoperability and easy access to EO data, making it possible to query and retrieve data based on spatial, temporal, and thematic criteria.

3.6. Web Services (Data Access and Delivery)

There are two map services provided:

- **Web Map Service (WMS):** Allows users to visualize EO data layers in mapping applications.
- **Web Coverage Service (WCS):** Provides access to raw EO data for further analysis and processing.

4. Collaboration

Figure 3. Collaborative initiative

Piksel is a collaborative initiative driven by a shared vision for innovation. The collaboration between **Badan Informasi Geospasial (BIG)** and **Geoscience Australia (GA)** aims to create an open and accessible EO platform tailored for Indonesia. This collaboration is driven by a collective commitment to leveraging EO data for national development and innovation.

4.1. Geoscience Australia (GA)

- Provides initial funding to kickstart the project.

- Shares expertise, technical guidance, and lessons learned from Digital Earth Australia and Digital Earth Africa.
- Supports knowledge transfer and best practices for platform development.

4.2. Badan Informasi Geospasial (BIG)

- Leads the adaptation of Piksel to Indonesia's specific geospatial needs.
- Ensures alignment with national geospatial policies and data standards.
- Facilitates local stakeholder engagement and outreach.
- Provides cloud infrastructure and technical support.

4.3. AWS Indonesia

- Provides cloud infrastructure and technical support.
- Assists with the deployment and optimization of AWS services for the piksel platform.
- Supports training and capacity-building initiatives for local teams.

2. INFRASTRUCTURE & DEPLOYMENT

The Piksel platform will be deployed using a cloud-first approach, leveraging AWS cloud services to ensure scalability, security, and efficient management of Earth observation (EO) data. The deployment strategy is designed to support high-performance processing, seamless data access, and user-friendly interactions with EO data.

1. Core Cloud Infrastructure

AWS services will serve as the building blocks for implementing Piksel's services, which include sandboxes, mapping applications, knowledge hubs, and web services. The infrastructure mentioned here will focus on the cloud services directly related to providing Piksel's services.

1.1. Sandboxes

- **AWS EKS (Kubernetes)**: Kubernetes is a tool used to manage groups of servers. AWS EKS is a service that helps run Kubernetes on AWS. It will manage clusters (groups) of On-Demand EC2 Instances and EC2 Spot Instances.
- **Autoscaling:** This means that the system can automatically adjust the number of servers based on user demand. For example, if more users start using the sandbox for data analysis, more servers will be added to handle the load.
- **Resource Management**: Dedicated CPU and RAM resources will be allocated to ensure smooth performance for users' analysis needs.

1.2. Maps

The platform will leverage the capabilities of Terria.IO, a mature external geospatial platform.

- **API-Driven Data Access:** Terria.IO will interact with the Piksel platform programmatically via standardized APIs. The Web Coverage Service (WCS) and Web Map Service (WMS) standards are used to request specific image tiles or data layers for visualization within the Terria.IO map interface.
- **Secure S3 Storage:** The actual imagery data will reside in an S3 bucket within the Piksel platform. S3 offers scalability and cost-effective storage for large geospatial datasets. This approach ensures that Terria.IO, and potentially other authorized applications, can access imagery data through the defined APIs.
- **API Gateway**: API Gateway will serve as the secure entry point for external applications like Terria.IO to access Piksel's data through the WCS/WMS APIs.

1.3. Knowledge Hub

- **Amazon S3 Bucket:** This is a storage service provided by AWS where static website files (like HTML, CSS, images) will be stored. It's cost-effective and highly durable.
- **Static Website:** The Knowledge Hub will provide information, landing pages, and documentation for users to learn about the project and access resources.

1.4. Explorer & Database

- **EC2 Instance:** A virtual server where the Explorer service and the database will be hosted.

- **RDS PostgreSQL + PostGIS Database:** RDS PostgreSQL is a powerful database system, and PostGIS is an extension that adds support for geographic objects. This database will store information about satellite imagery and user data. Automatic backups will be performed to ensure data protection and recovery in case of any failures.

1.5. STAC and Web Services

- **AWS API Gateway:** This service helps manage APIs (Application Programming Interfaces) which allow different software systems to communicate with each other. It will facilitate data access and interoperability, making it easier for users to access geospatial data and services.
- **AWS Lambda:** AWS Lambda can handle various backend processes triggered by API requests. It allows custom code execution without the need for server management, making it ideal for tasks such as data transformation, validation, and real-time analysis. AWS Lambda ensures scalable and cost-effective execution of tasks based on interactions with the API Gateway and integrates seamlessly with other AWS services.

2. Supporting Cloud Infrastructure

To support the Piksel services, the following AWS components will be utilized to ensure everything runs smoothly and efficiently:

2.1. Satellite Imagery Storage

- **Amazon S3 (Simple Storage Service):** This is a storage service provided by AWS that is highly reliable and durable. It will be used to store derived products from raw satellite imagery. The raw satellite imagery available from Sentinel and Landsat Servers might not be moved, but the derived products will need storage in Amazon S3.

-

2.2. Database Solutions and User Management

- **PostgreSQL + PostGIS:** This database system will RDS PostgreSQL. PostgreSQL handles general database tasks, while PostGIS adds support for geographic objects. It will store both satellite imagery index information and user data.
- **AWS Backup:** This service will be used to automate routine backups of the database to ensure data safety and recovery in case of any issues.

2.3. Development Pipeline

- **AWS CodePipeline:** The development pipeline uses AWS CodePipeline to orchestrate the Continuous Integration/Continuous Delivery (CI/CD) process. Code is committed to GitHub, triggering CodePipeline. CodeBuild builds and tests the application, leveraging cost-effective EC2 Spot Instances for testing. Built artifacts are stored in ECR and deployed to the appropriate environments.

2.4. Security & Monitoring

- **AWS IAM (Identity and Access Management):** This service manages access to AWS resources by assigning roles and permissions to users. It ensures that only authorized individuals can access certain data and services.
- **AWS Web Application Firewall (WAF):** AWS WAF protects web applications from common web exploits and attacks. It adds a layer of security to safeguard the Piksel services.

- **AWS CloudWatch:** This monitoring service collects and tracks metrics, collects log files, and sets alarms. It helps in logging and monitoring the performance of the services to ensure they run smoothly.
- **AWS VPC (Virtual Private Cloud):** AWS VPC allows the creation of a private network within AWS. This secure network connects all the services, ensuring that data transfers are safe and protected.

3. Deployment, Scalability, Reliability, and Security

The following strategies are designed to ensure that the Piksel infrastructure not only supports the immediate needs of the services provided but also remains robust, scalable, and secure over time. These strategies integrate various AWS components to deliver automated deployment, performance monitoring, redundancy, and compliance.

3.1. Deployment Strategies

- **Automated Deployment:** Using AWS CodePipeline, automated deployment ensures that new code changes are tested and deployed efficiently without manual intervention. This minimizes downtime and ensures continuous integration and deployment.

3.2. Scalability and Performance

- **Autoscaling:** AWS services such as EKS, ASG, and Lambda ensure that the infrastructure can automatically scale to meet varying demands. This ensures high performance and responsiveness even during peak usage.
- **Performance Monitoring:** AWS CloudWatch is used to monitor the performance of all services, with alerts set up to detect and respond to performance issues promptly.

3.3. Reliability and Redundancy

- **Routine Backups:** AWS Backup ensures that all critical data, including databases and configurations, are regularly backed up. This enables quick recovery in case of data loss.
- **Failover Mechanisms:** High availability is achieved through redundancy, ensuring that services remain operational even if some components fail. Key services are deployed across multiple availability zones.

3.4. Compliance and Security

- **Access Control:** AWS IAM is used to implement role-based access control, ensuring that only authorized personnel have access to sensitive data and services.
- **Data Protection:** AWS WAF and encryption measures protect against common web threats and ensure the security of data in transit and at rest.
- **Compliance:** The infrastructure complies with relevant laws and regulations, with regular security audits and updates to maintain compliance.

4. Infrastructure Diagram

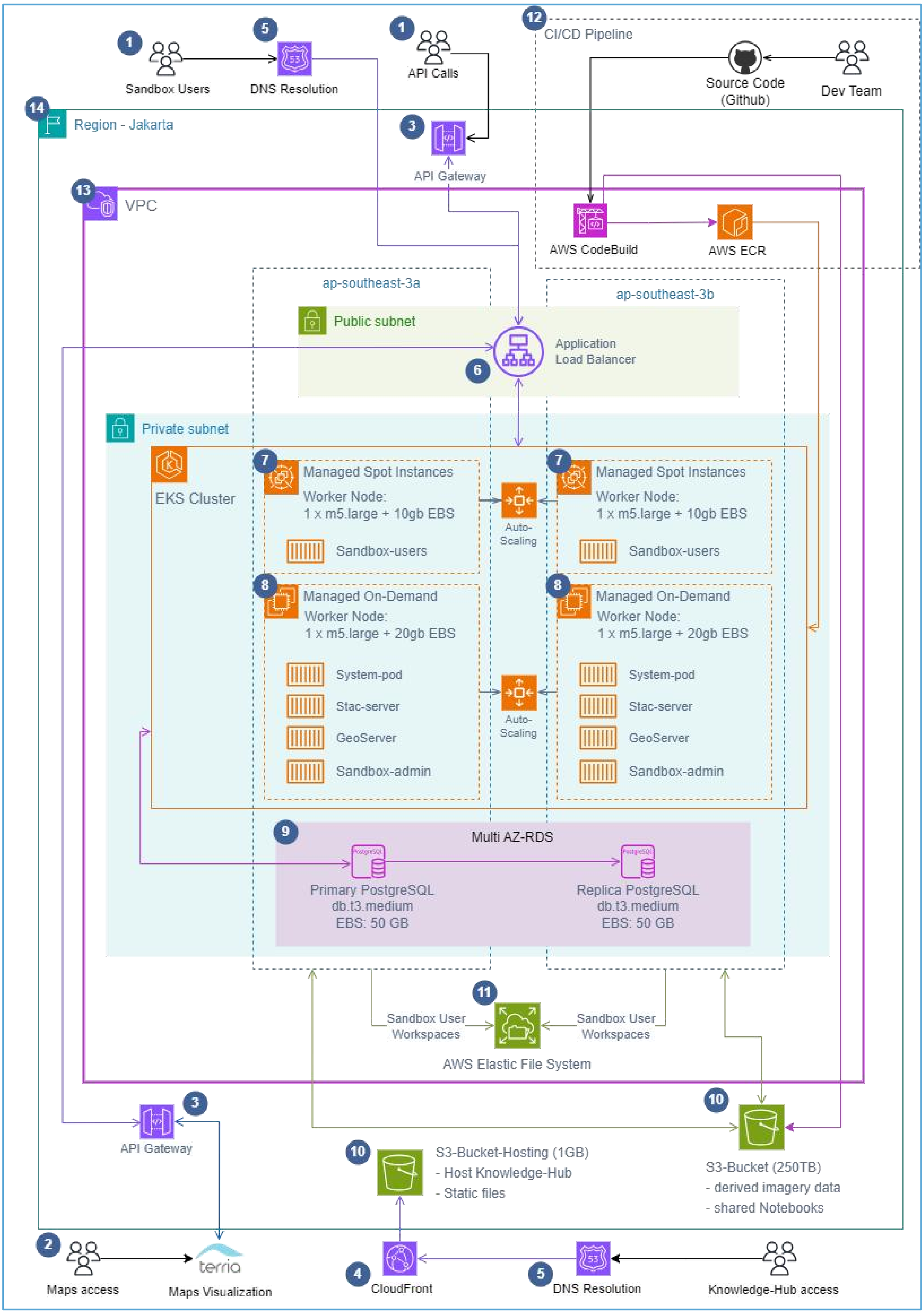

Figure 4. Infrastructure Diagram

Key Points of Piksel Architecture with Detailed Explanations:

1. **Users** (Sandbox/API/Knowledge Hub): Users interact with the Piksel platform through three distinct access points:
    - JupyterLab Sandboxes: Users access isolated JupyterLab environments for experimentation and analysis.
    - STAC/WCS/WMS APIs: Users or external applications can programmatically query metadata (STAC) and request imagery (WCS/WMS).
    - Knowledge Hub: Users can browse and access information about the platform and its resources.
2. **Terria.IO Integration**: Terria.IO, an external geospatial platform, integrates with Piksel by making programmatic requests to the STAC/WCS/WMS APIs. This allows Terria.IO to access and visualize the imagery data managed by Piksel.
3. **API Gateway (External):** The API Gateway, positioned outside the VPC, acts as a gatekeeper for external access to the STAC/WCS/WMS APIs. It handles:
    - Authentication: Verifies the identity of the user or application making the request.
    - Authorization: Controls what resources the authenticated user or application can access.
    - Request Routing: Directs API requests to the appropriate backend services (STAC Server, GeoServer).
    - Throttling: Prevents abuse by limiting the number of requests.
4. **CloudFront (CDN)**: CloudFront, a global Content Delivery Network (CDN), accelerates delivery of the Knowledge Hub website to users. It caches static content (HTML, CSS, JavaScript, images, fonts) closer to users geographically, reducing latency and improving load times. It sits outside the VPC.

5. **Route 53 (DNS)**: Route 53 is the DNS service that translates domain names (e.g., knowledgehub.piksel.com) into IP addresses of the appropriate entry points: API Gateway for API access, CloudFront for the Knowledge Hub, and the Application Load Balancer (ALB) for JupyterLab.
6. **Application Load Balancer (ALB):** The ALB, located in the public subnet, distributes incoming traffic across multiple JupyterLab pods running on the Spot Instances. This improves scalability and fault tolerance. It receives authenticated traffic from Cognito.
7. **EKS Cluster (Spot Instances - Autoscaling)**: This EKS cluster hosts the JupyterLab pods used for user sandboxes. Spot Instances are used for cost-effectiveness. Autoscaling is configured (e.g., min 2, max 10 instances) to dynamically adjust the number of running instances based on user demand, ensuring responsiveness while managing costs. Each instance is an m5.large with 10GB EBS volumes for application data.
8. **EKS Cluster (On-Demand Instances - Autoscaling)**: This EKS cluster hosts the core services: STAC Server, GeoServer, a dedicated JupyterLab Admin pod for managing shared notebooks, and other system components. On-Demand instances are used for reliability and stability, as they are not subject to interruption like Spot Instances. Autoscaling is also enabled (min 2, max 3 instances) to ensure these critical services can handle varying loads. Each instance is an m5.large with 20GB EBS volumes for application data.
9. **RDS PostgreSQL (Multi-AZ - Primary/Replica):** RDS PostgreSQL stores metadata about satellite imagery. The multi-AZ deployment creates a primary and a replica database in different Availability Zones. This ensures high availability; if one Availability Zone fails, the replica can be quickly promoted to become the primary, minimizing downtime.
10. **S3 (Storage - Imagery, Notebooks, Website)**: S3 provides scalable object storage. Two separate buckets are used:

- Imagery & Shared Notebooks (250TB estimated): Stores the raw satellite imagery data and the shared Jupyter notebooks used across the platform. Connections are made from the EKS cluster (for data access) and Terria.IO (for imagery access via APIs).
- Knowledge Hub Website: Stores static files (HTML, CSS, JavaScript, images, fonts, etc.) that make up the Knowledge Hub website. CloudFront connects to this bucket to serve the website content.

11. **EFS (Elastic File System)**: EFS provides a shared file system that is accessible by all pods in the EKS cluster. This is used for user workspaces, allowing users to store temporary files, install packages, and manage their JupyterLab environment. It's needed for data persistence across pod restarts and for shared access.
12. **CI/CD (CodePipeline, CodeBuild, ECR)**: The Continuous Integration/Continuous Delivery (CI/CD) pipeline automates the process of building, testing, and deploying updates to the Piksel platform. Developers commit code to GitHub. CodePipeline triggers the pipeline, CodeBuild builds Docker images and pushes them to ECR. These images are then used to update the deployments running on EKS.
13. **VPC (Virtual Private Cloud)**: Piksel utilizing Availability Zones 3a and 3b for redundancy. The VPC provides network isolation and security for all Piksel resources.
14. **Region Jakarta**: The entire Piksel infrastructure resides within the ap-southeast-3 region (Jakarta).

3. PHASES & TIMELINE

The Piksel project will be executed in multiple phases to ensure a structured and efficient development process. Each phase focuses on specific deliverables and milestones to achieve the overall project objectives. The stages are:

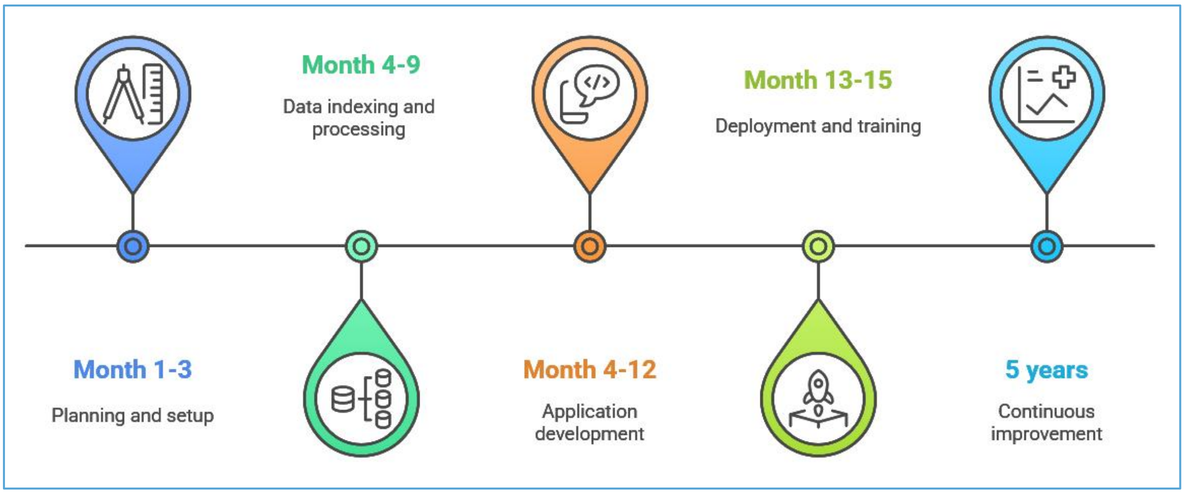

Figure 5. Project Timeline

Phase 1 - Planning and Setup

Duration	3 Months; January – March 2025
Objectives	○ Define system requirements and architecture. ○ Establish cloud infrastructure and security measures. ○ Collaborate with stakeholders to align goals and expectations.

Key Activites	○ **Regulation Compliance:** Ensuring the project complies with all relevant regulations and obtaining necessary approvals. ○ **Designing Cloud Architecture:** Selecting the appropriate AWS services and designing how they will work together to support Piksel services. The cloud design outlined in this document will be reviewed and curated with the input of technical architects to finalize the design. Detailed architecture diagrams and documentation will be created to guide the implementation. ○ **Setting Up AWS VPC, IAM Policies, and Security Groups:** Establishing a secure network environment and defining access controls are critical to protecting project resources. Activities include configuring a Virtual Private Cloud (VPC) to securely host Piksel services, defining IAM policies to manage user permissions, and setting up security groups to control inbound and outbound traffic to AWS resources. ○ **Resources Allocatin:** Estimating the costs associated with the project and allocating necessary financial resources form the basis of this activity. This involves conducting cost estimation for AWS services, personnel, and other

	expenses, allocating AWS credits and financial resources accordingly, and creating a detailed budget plan. ◦ **Setting Up the Development Environment and CI/CD Pipeline:** Configuring tools and processes for efficient development and deployment is crucial. This involves setting up development tools and environments for coding and testing, establishing a CI/CD pipeline using GitHub Action to automate code testing and deployment.
Milestones	◦ Approval of system requirements and architecture. ◦ Deployment of initial cloud environment. ◦ Completion of security assessments. ◦ Establishment of clear roles and responsibilities within the project team. ◦ Initial budgeting plan approved, and resources allocated.

Phase 2 - Data Ingestion and Processing

Duration	6 Months; April – September 2025

Objectives	○ Acquire, ingest, and index satellite imagery. ○ Create high-quality derived products and add them to the storage. ○ Ensure efficient data storage and retrieval.
Key Activites	○ **Acquiring and Indexing Satellite Imagery:** Indexing the data using the stac-to-dc tool, and ensuring compatibility with the datacube product scheme. ○ **Automating Data Ingestion:** setup to ensure that the ingestion pipelines efficiently transfer, transform, and load data into the systems automatically. ○ **Creating Derived Products:** Creating high-quality derived products such as geo-median images and Analysis Ready Data (ARD) satellite imagery. ○ **Storing and Managing Derived Products in AWS S3:** The derived products will be stored in Amazon S3, organized efficiently to ensure easy access and management. ○ **Conducting Performance Testing of Data Retrieval:** Performance testing ensures that data retrieval processes perform efficiently under different load conditions. This activity involves simulating various query loads, identifying bottlenecks, and optimizing the system to handle varying volumes of

	data queries. ◦ **Implementing Open Data Cube (ODC) Features:** This includes setting up and configuring the core ODC features to ensure the system can handle the required data processing tasks.
Milestones	◦ Successful acquisition and ingestion of initial data sets. ◦ Implementation of basic ODC features. ◦ Creation and storage of high-quality derived products. ◦ Validation of data accuracy and completeness. ◦ Completion of the indexing framework. ◦ Optimized performance of data retrieval processes.

Phase 3 - Application Development

Duration	9 Months; April – December 2025
Objectives	◦ Develop core Piksel services including Maps, Explorer, and Sandbox. ◦ Ensure interoperability with STAC, WMS, and WCS.

	◦ Conduct internal testing and validation.
Key Activites	◦ **Developing Piksel Maps for Geospatial Visualization Using Terria.IO:** Creating an interactive mapping tool using Terria.IO to visualize geospatial data. ◦ **Implementing Explorer for Metadata and Product Browsing:** Developing a web-based interface that allows users to browse and search for Metadata and derived products. Using the open-source datacube-explorer repository. ◦ **Setting Up Sandbox Environment with Kubernetes and Docker:** Creating a sandbox environment for users to perform data analysis and experimentation. This involves using Kubernetes and Docker to manage and isolate user sessions, ensuring each user has a dedicated environment to work in. ◦ **Developing APIs for Interoperability:** Ensure the Piksel platform can interact with other systems and services using APIs. These APIs will follow industry standards and be well-documented to facilitate easy integration. This step is essential for enabling interoperability with STAC (SpatioTemporal Asset Catalog), WMS (Web Map

	Service), and WCS (Web Coverage Service). ○ **Integrating Security and User Authentication (Amazon Cognito):** Integrating user authentication and authorization using Amazon Cognito. ○ **Conducting Functional and Performance Testing:** Functional testing ensures that the developed applications work as intended, while performance testing evaluates how the applications perform under different load conditions. Identifying and fixing any issues during testing helps deliver a robust and reliable platform.
Milestones	○ Deployment of functional prototypes for Piksel Maps, Explorer, and Sandbox. ○ Completion of internal testing and validation. ○ Ensured interoperability with STAC, WMS, and WCS. ○ Successful integration of security and user authentication. ○ Implementation of ODC using Kubernetes and odc-stats orchestration.

Phase 4 - Deployment and Training

Duration	3 Months; January – March 2026
Objectives	◦ Deploy the Piksel platform for public access. ◦ Conduct training sessions for users and stakeholders. ◦ Establish a support and maintenance framework.
Key Activites	◦ **Deploying the Final Version to Production AWS Environment:** configuring the production environment, ensuring all settings and configurations are optimized for public access, and conducting final deployment tests to verify that everything is working as expected. ◦ **Conducting Training Sessions for Government Agencies and Businesses:** organizing and conducting training sessions to educate users on how to navigate and utilize the platform. Training sessions will include hands-on demonstrations, tutorials, and Q&A sessions to address any questions or concerns. ◦ **Developing Documentation and User Guides:** Comprehensive documentation and user guides will be created to assist users in navigating and

	utilizing the Piksel platform. These resources will include step-by-step instructions, best practices, and troubleshooting tips. The goal is to provide users with the information they need to use the platform effectively and independently. ○ **Setting Up Ongoing Monitoring and Support Mechanisms:** Setting up monitoring tools and processes, defining support channels for users to report issues and receive assistance, and implementing incident response procedures to address any issues promptly. ○ **Collecting User Feedback and Optimizing Platform Performance:** collecting user feedback through surveys, focus groups, and direct interactions, analyzing the feedback to identify common themes and issues, and implementing changes and optimizations based on the feedback to enhance the platform's performance and user experience.
Milestones	○ Successful deployment of the final version to the production AWS environment. ○ Completion of training sessions for government agencies and businesses. ○ Publication of comprehensive documentation and user guides.

	○ Establishment of monitoring and support mechanisms. ○ Collection and analysis of user feedback. ○ Initial optimization of platform performance based on user feedback.

Phase 5 - Continuous Improvement

Duration	Ongoing; March 2026 ++
Objectives	○ Continuously enhance the platform based on user feedback. ○ Expand capabilities and integrate additional datasets.
Key Activites	○ **Monitoring System Performance and User Feedback:** This activity involves using monitoring tools such as AWS CloudWatch to track key performance indicators (KPIs) and collecting feedback from users through surveys, focus groups, and direct interactions. ○ **Introducing New Features Based on Demand:** Prioritizing feature requests, designing and

	developing new features, and conducting testing to ensure the new features meet user needs and quality standards. Regular updates and enhancements will be made to the platform based on user demand. ◦ **Expanding Satellite Imagery Sources and Datasets:** Identifying additional data sources, integrating new datasets into the existing infrastructure, and ensuring compatibility with the platform's data management and retrieval systems. ◦ **Ensuring Security and Compliance Updates:** Regularly reviewing and applying security updates, conducting security audits, and ensuring compliance with relevant regulations and standards. Proactive security measures will help protect the platform from emerging threats.
Milestones	◦ Regular release of updates and new features. ◦ Expansion of user base and datasets. ◦ Achievement of performance optimization targets. ◦ Maintenance of high security and compliance standards. ◦ Continuous collection and analysis of user feedback and system performance data.

4. SPECIFICATIONS & REQUIREMENTS

This section provides detailed technical specifications and requirements for each service within the platform. It includes core features, potential enhancements, and the minimum skillsets needed to implement these services.

1. Specifications

This section provides detailed specifications for each service within the Piksel platform, outlining the core features and potential enhancements.

1.1. Sandbox

Core Features	Potential Enhancements
○ Provide a cloud-based analysis environment using Docker and Kubernetes (EKS). ○ Dedicated CPU/RAM allocation.	○ Customizable resource allocation. ○ Advanced monitoring and usage analytics.

Technical Specifications:

- Containerization: Docker
- Orchestration: Kubernetes (Amazon EKS)
- Instance Type:
  - m5.large (2 vCPU, 8GB RAM) – On-Demand Instances for Administrators with 20GB EBS Volume per instance

- m5.large (2 vCPU, 8GB RAM) – Spot Instances for Users with 10GB EBS Volume per Sandbox Instance

Workflow:

- Request:
  - A user requests a Sandbox instance through the Piksel platform's interface.
  - An administrator requests a Sandbox instance through the Piksel platform's interface.
- Provisioning:
  - The platform provides a new Sandbox environment by deploying a JupyterLab pod on the EKS cluster designated for Sandbox instances (Spot Instances)
  - The platform provisions a new Sandbox environment by deploying a JupyterLab pod on the EKS cluster designated for Sandbox instances (On-Demand Instances).
- Resource Allocation: The pod is configured with the specified CPU, RAM, and storage resources.
- Access:
  - The user accesses their Sandbox environment through a secure connection (e.g., via the Application Load Balancer).
  - The administrator can utilize the JupyterLab environment for administrative tasks, such as managing users, configuring the platform, or troubleshooting issues.
- Usage: The user can utilize the JupyterLab environment to analyze data, develop code, and conduct experiments.
- Monitoring: The platform monitors resource usage and activity within the Sandbox.

- Termination: Once the work is complete, the Sandbox environment can be terminated, freeing up the resources

1.2. Maps

Core Features	Potential Enhancements
- Browser-based geospatial visualization with Terria.IO. - Basic map layers (raster).	- Vector visualization support. - Advanced analytical tools.

Specifications:

- Platform: Terria.IO
- API Integration: WCS & WMS (for imagery access)
- Data Storage: S3 (dedicated bucket for imagery data)
- Access Control: API Gateway (for secure API access)

Workflow:

- User Interaction: Users interact with maps through Terria.IO's interface.
- Data Discovery: Terria.IO queries the Piksel platform's API to find relevant imagery.
- Imagery Access: Terria.IO requests imagery data from the Piksel platform using WCS/WMS APIs.
- Visualization: Terria.IO visualizes the imagery data on the map.
- Analysis: Users can utilize Terria.IO's tools for geospatial analysis.

1.3. Explorer

Core Features	Potential Enhancements
○ Browse Metadata and product locations. ○ Simple search and filtering options.	○ Prevent activities that might burden the servers

Specifications:

- Component: datacube-explorer (part of the Open Data Cube project)
- Dedicated Pod in EKS
- Data Source: PostgreSQL database (RDS) containing the ODC metadata catalog
- User Interface: Web-based interface provided by datacube-explorer

Workflow:

- User Access: Users access the datacube-explorer web service through a web browser.
- Metadata Browsing: Users can browse the available imagery metadata.
- Visualization: Users can visualize the location of imagery products on a map.
- Search & Filter: Users can search and filter the data based on specific criteria.

1.4. Knowledge Hub

Core Features	Potential Enhancements
◦ Static web pages providing product knowledge, tutorials, and documentation. ◦ Hosted on AWS S3. ◦ Supporting multiple languages.	◦ Interactive guides, video tutorials and training materials. ◦ The website includes a search function enabling users to quickly find the information they need.

Specifications:

- Hosting: S3 (Static Website Hosting)
- Content Delivery: CloudFront (CDN)
- Storage: S3 Bucket dedicated to Knowledge Hub content.

Workflow:

- Content Creation/Update: Content creators develop and update website content using the chosen content management system.
- Build : The content is built into static HTML files.
- Deployment: The static website files are uploaded to the designated S3 bucket.
- CloudFront Distribution: CloudFront is configured to serve the content from the S3 bucket.

- User Access: Users access the Knowledge Hub website through a web browser via the CloudFront URL.

1.5. STAC & Web Services

Core Features	Potential Enhancements
○ API gateway for data interoperability. ○ Basic STAC implementation.	○ Advanced WMS/WCS integration. ○ Real-time data access optimization.

Specifications:

- Software Components:
  - STAC Server
  - Web services with GeoServer
- Deployment: Kubernetes (Amazon EKS)
- Data Storage: S3 (dedicated bucket for imagery data)
- API Access: API Gateway (for external access), internal service discovery within EKS.
- Database: RDS PostgreSQL (for STAC metadata catalog)

Workflow:

- API Request: A user or application (e.g., Terria.IO) makes a request to the STAC API (for metadata) or WCS/WMS endpoints (for imagery).

- Authentication/Authorization: API Gateway (for external requests) or internal Kubernetes mechanisms (for internal requests) handle authentication and authorization.
- STAC Query (if applicable): If the request is for metadata, the STAC Server queries the RDS PostgreSQL database.
- Data Retrieval: GeoServer retrieves the requested imagery data from the S3 bucket.
- Processing (if applicable): GeoServer processes the data (e.g., reprojection, format conversion).
- Response: The STAC Server returns metadata, or GeoServer returns imagery data or rendered map tiles to the user or application.

1.6. Storage & Database

Core Features	Potential Enhancements
○ S3 Bucket 1: stores the raw satellite imagery data and the shared Jupyter notebooks used by users across the platform. ○ S3 Bucket 2: stores static website content for the Knowledge Hub, including HTML, CSS, JavaScript, images, fonts, and other static assets ○ RDS stores metadata about satellite imagery products, including, Spatial and	○ Automated failover and disaster recovery.

temporal extent, Resolution, Acquisition date, Sensor information, links to the actual imagery data ◦ EFS provides persistent storage for user workspaces	

Specifications:

- **S3 Bucket 1**: Imagery Data & Shared Notebooks
  - Estimated Size: 250 TB.
  - Data Types: Satellite imagery data, Jupyter Notebook files (.ipynb)
  - Access: EKS Cluster (Pods):
  - Direct access for data processing and analysis.
  - Terria.IO: Access via API Gateway for imagery visualization.
  - JupyterLab Admin Pod: Access for managing shared notebooks.
  - Lifecycle Policies: (Define any lifecycle policies, e.g., archiving older data to Glacier after a certain period).
  - Security: Access controlled through IAM policies, potentially including bucket policies and Access Control Lists (ACLs).
- **S3 Bucket 2**: Knowledge Hub Website
  - Data Types: HTML files, CSS files, JavaScript files, Image files (various formats), Font files, Other static assets
  - CloudFront: Access for content delivery to users.
  - Lifecycle Policies: cache invalidation
  - Security: Access controlled through IAM policies and potentially bucket policies.
- **Database (RDS PostgreSQL)**

- Instance Type: db.t3.medium
- Storage: 50 GB EBS
- Multi-AZ Deployment: Primary and replica instances in different Availability Zones for high availability.
- Backup and Recovery: Automated backups enabled, point-in-time recovery available.
- Security: Access controlled through security groups and IAM authentication.

○ **EFS (Elastic File System)**

- Store temporary files and data.
- Install and manage software packages within user's Sandbox environments.
- Share files and collaborate with other users (if configured)..
- Access: Accessible by all pods within the EKS cluster.
- Security: Access controlled through security groups and IAM authorization.
- Mount Points: EFS file systems are mounted to specific directories within the user's Sandbox environments.

1.7. Security & Monitoring

Core Features	Potential Enhancements
○ IAM roles and policies for access control. ○ Basic monitoring with CloudWatch.	○ Web Application Firewall (WAF) setup. ○ Security audits and compliance checks.

- Implementing a centralized log management system to provide a consolidated view of logs from all components of the platform.
- Integrating threat detection services
- Implementing an intrusion detection/prevention system (IDS/IPS) for deep packet inspection and real-time threat response.
- Vulnerability Scanning: Regular vulnerability scanning of EC2 instances, containers, and other components

Specifications:

- Identity and Access Management: AWS IAM
- Monitoring: Amazon CloudWatch
- Web Application Firewall (Potential): AWS WAF
- Security Audits (Potential): AWS Config, AWS Inspector
- Log Management (Potential): Amazon Elasticsearch Service, CloudWatch Logs Insights
- Threat Detection (Potential): Amazon GuardDuty

1.8. User Management

Core Features	Potential Enhancements
○ Users can log in using a combination of username, email, and password. ○ Access to platform resources and functionalities is controlled through authorization mechanisms. ○ Users can save code snippets within their Sandbox environments. ○ Username, email, and password-based login system. ○ Authentication tools and database setup. ○ Provide code snippets saving in the sandbox and Piksel Maps.	○ Customizable resource allocation based on request and approval. ○ Implementing user activity monitoring and reporting would provide insights into user behavior, resource usage, and potential security risks. ○ Users visualize maps in Piksel Maps (Terria.IO).

Specifications:

- Authentication Provider: Amazon Cognito
- Authorization: Amazon Cognito, IAM
- User Interface: Integrated with the Piksel platform's web interface.
- Database: Within the existing RDS PostgreSQL instance.

- Code Snippet Storage: user's Sandbox environment (EFS) and back up in S3 bucket.

Workflow:

- Registration: New users can register for an account by providing the required information.
- Login: Users can log in using their credentials.
- Code Snippet Saving: Users can save code snippets within their Sandbox environments.
- Resource Request (Potential): Users can request customized resource allocation for their Sandboxes.
- Activity Monitoring (Potential): User activity is monitored and reported

2. Skillset Requirements

This section outlines the minimum skillsets needed to implement and maintain the services within the Piksel platform.

Implementation Area	Minimum Skillset
Sandbox	○ Cloud infrastructure (AWS) ○ DevOps Practices with Docker ○ Kubernetes (EKS) and container orchestration ○ Python programming ○ Version control - GitHub

Implementation Area	Minimum Skillset
Maps	○ Web development (JavaScript, HTML, CSS) ○ Geospatial Information Systems (GIS) ○ Terria.IO Platform Experience ○ AWS Services (S3, CloudFront - for potential custom UI elements
Explorer	○ Web development (JavaScript, HTML, CSS) ○ Geospatial data handling ○ Experience with PostGIS ○ AWS services (EC2, S3) ○ Version control using GitHub
Knowledge Hub	○ Web development (JavaScript, HTML, CSS) ○ Experience with AWS S3 for static website hosting ○ Documentation and content creation ○ Version control using GitHub

Implementation Area	Minimum Skillset
Data Ingestion Pipelines	○ Python programming ○ AWS Lambda for serverless computing ○ PostgreSQL/PostGIS for database management ○ Handling and processing geospatial data ○ Version control using GitHub
STAC & Web Services	○ API development (RESTful, GraphQL) ○ Experience with AWS API Gateway ○ Handling geospatial services ○ Version control using GitHub
Storage & Database	○ Database management (PostgreSQL/PostGIS) ○ Experience with AWS S3 ○ Backup strategies and implementation
Security & Monitoring	○ Cloud security best practices ○ AWS IAM for access control ○ AWS CloudWatch for monitoring

Implementation Area	Minimum Skillset
	◦ Security auditing
User Management	◦ Web security best practices ◦ Database management for user data ◦ Authentication and Authorization tools (OAuth, JWT) ◦ User activity monitoring and reporting ◦ Version control using GitHub
CI/CD	◦ CI/CD Principles & Best Practices ◦ AWS CodePipeline ◦ AWS CodeBuild ◦ AWS ECR ◦ Infrastructure as Code (IaC) (Terraform/CloudFormation)

5. SUCCESS METRICS & EVALUATION

1. Success Metrics and Evaluation

The success of the Piksel project will be evaluated based on the achievement of specific goals and milestones across different phases. The criteria for success are detailed below:

- **Data Accessibility:** Measure the number of unique users accessing EO data through DE Indonesia using web analytics tools.
- **Performance:** Monitor the average response time of the Explorer and Sandbox services using performance monitoring tools.
- **Usage Growth:** Track the number of new users from government agencies, businesses, and researchers registering on the platform monthly.
- **Training Effectiveness:** Evaluate user feedback through post-training surveys and measure competency improvement via pre- and post-training assessments.
- **Data Coverage:** Track the increase in the number of indexed EO products over time through database records.

2. Overall Project Success

The success indicators of Piksel project are as follows:

- **Achievement of Objectives:** Assess the completion of all defined objectives for each phase through milestone reviews.
- **User Satisfaction:** Collect and analyze feedback surveys and usage metrics to ensure high levels of satisfaction.

- **Impact on Coastal Monitoring and Management:** Document the creation and utilization of intertidal coastlines products, demonstrating their impact.
- **Sustainability and Scalability:** Evaluate the platform's ability to accommodate future growth and additional datasets through system audits.

6. RISKS & MITIGATION STRATEGIES

Identifying potential risks and implementing effective mitigation strategies is crucial to the success of the Piksel project. Below are the key risks and their corresponding mitigation strategies:

Potential Risks	Mitigation Strategy
Data Overload	Implement efficient indexing and cloud storage policies.
Security Breaches	Enforce strict IAM roles and conduct regular security audits
Performance Bottlenecks	Optimize infrastructure and enable auto-scaling.
Low User Adoption	Conduct outreach and training programs to increase awareness.
Data Inaccuracy	Implement robust data validation and quality control processes.
Budget Overruns	Regularly monitor expenses and adjust plans as

Potential Risks	Mitigation Strategy
	needed.
Integration Issues	Thoroughly test API integrations and ensure compatibility.
Compliance Violations	Stay updated with regulations and conduct compliance checks.
User Experience Issues	Gather user feedback and continuously improve the interface.

7. CONCLUSION

The Piksel project represents a significant step forward in leveraging Earth Observation (EO) data. By following the detailed phases and implementing robust technical specifications and skill sets, we aim to create a powerful platform that meets the needs of diverse stakeholders.

Our success metrics and evaluation criteria will ensure that the project remains on track and achieves its objectives. Additionally, the identified risks and mitigation strategies will help us navigate potential challenges effectively.

We look forward to the successful execution and continuous improvement of the Piksel platform. And great appreciation to all the stakeholders, team members, and partners who are contributing to this important project. Together, we will make a lasting impact.