



Leonardo Cyber & Security Solutions

5 Nov 2025  
01.00

Secure Cloud Management Platform

NUMERO DOCUMENTO: **C000CMP01STP01**

REVISIONE: **01.00**

DATA: **04/11/2025**

CAGE CODE: **A0069**

## **Leonardo Services Documentation**



Leonardo Cyber & Security Solutions

5 Nov 2025

01.00

Secure Cloud Management Platform

## Firme

Autore: <b>Digital Proposal &amp; Pre Sales</b> Digital Proposal & Pre Sales  Digital Systems & Engineering Technologies   Engineering	..... Elio Arena
Verifica: <b>PEM IPT di Prodotto</b> R. Digital Systems & Engineering Technologies   Engineering	..... Andrea Giorgio Busà
Verifica: <b>PAM IPT Sviluppo</b> Quality Cyber Security, Intelligence & Digital Solutions	..... Simonetta De Biase
Approvazione: <b>IPT Leader IPT di Sviluppo</b> R. Digital Platform   Digital Systems & Engineering Technologies   Engineering	..... Daniele Leone
Approvazione: <b>Technical Authority</b> Solution Architects   LoB Public Admin., Defence & Inter. Agencies	..... Susanna Fortunato
Autorizzazione: <b>Product Manager IPT Prodotto</b> Product Management Digital Trasformation   Product Management	..... Fabio Russo

## Contatti

Elio Arena <b>Digital Proposal &amp; Pre Sales</b> Digital Proposal & Pre Sales   Digital Systems & Engineering Technologies   Engineering	<b>Leonardo S.p.A.</b> Via A. Agosta SNC 95121 Catania
--	---



## **Lista delle Revisioni**

<b>Rev.</b>	<b>Numero Modifiche</b>	<b>Data</b>	<b>Descrizione</b>	<b>Autore</b>
01.00	-	24/01/2022	Prima emissione	D. Leone
02.00	DCN222372	29/07/2022	Integrazione Rilascio SCMP 2.0.0	D. Leone
03.00	DCN222981	20/12/2022	Integrazione Rilascio SCMP 3.0.0	D. Leone
04.00	DCN230550	30/06/2023	Integrazione Rilascio SCMP 4.0.0	D. Leone
05.00	DCN231199	22/12/2023	Integrazione Rilascio SCMP 5.0.0	D. Leone
06.00	DCN240480	28/07/2024	Integrazione Rilascio SCMP 6.0.0	D. Leone
07.00	DCN240891	20/12/2024	Integrazione Rilascio SCMP 7.0.0	D. Leone



Leonardo Cyber & Security Solutions

5 Nov 2025

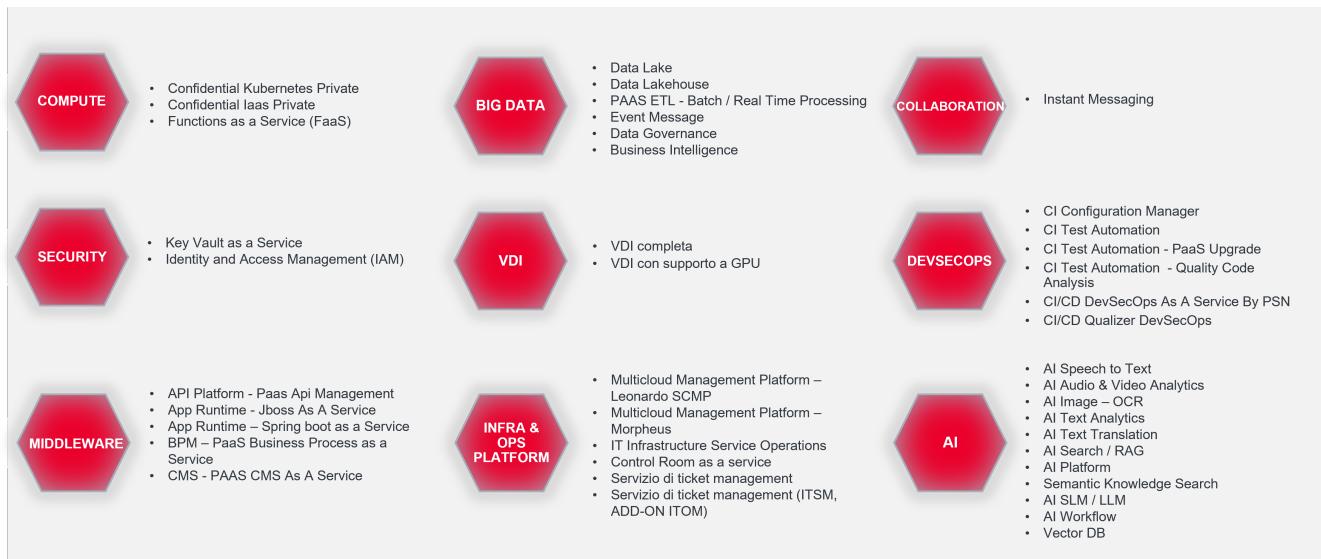
01.00

Secure Cloud Management Platform

# Leonardo Services Documentation

# 1 Leonardo Services

Leonardo provides a collection of managed services which are represented in the following figure by type and sub-type (technically called "Family" and "Sub-family" respectively).



*Figura 1 – Leonardo Services Overview*

From a logical-functional perspective, these services can be divided into three macro-categories:

- Infrastructure as a Service (IaaS)
- Container as a Service (CaaS)
- Platform as a Service (PaaS)

The IaaS and CaaS categories include some services from the "Compute" family. The PaaS category includes services from all other families.

The above macro-categories are described below, and for each of them, we provide a description of the services for each family and subfamily.



## 2 Infrastructure as a Service (IaaS)

The following table lists the services falling under the Infrastructure as a Service (IaaS) category.

FAMILY	SUB-FAMILY	LIST OF SERVICES
Compute	Confidential - IaaS - Private	Pool Small (Confidential) Pool Medium (Confidential) Pool Large (Confidential) Pool X-Large (Confidential)

### 2.1 Compute Family

Below are the sub-families belonging to the Compute family:

- Confidential - IaaS – Private

#### 2.1.1 Confidential - IaaS - Private Sub-Family

Below is the list of services within the Confidential - IaaS - Private Sub-family:

- Pool Small (Confidential)
- Pool Medium (Confidential)
- Pool Large (Confidential)
- Pool X-Large (Confidential)

##### 2.1.1.1 Pool Confidential Services

###### > Services Description

These services enable the provision of Private virtual computing environments (IaaS), i.e., on a pool of physical resources, dedicated and isolated for each individual customer, based on the use of bare metal computing instances. Data from physical resources is encrypted and kept secure throughout all phases of use (at rest, in transit, and in use), leveraging the Confidential Computing paradigm.

Depending on the pool of computing resources required for each individual Administration, the most suitable service from the four available types can be selected.

###### > Features and Advantages



Private Cloud resources are dedicated exclusively to each customer.

The services use secure enclaves based on Trusted Execution Environments (TEEs) based on Confidential Hardware, which offer an advanced level of security for data in use, protecting it during processing.

They support advanced encryption of data at rest, in transit, and in use.

They use advanced remote attestation systems to verify the correctness of the TEE environment, isolating virtual machine memory from the host operating system and other malicious guests.

The services offer the following advantages:

- *Data security and confidentiality in dedicated environments.*
- *Workload isolation* through advanced virtualization.
- *Dedicated firewalls and network micro-segmentation.*
- *Automated provisioning and rapid resource management.*
- *Comprehensive control and centralized governance:* centralized monitoring and auditing for traceability.



## 3 Container as a Service (CaaS)

The following table lists the services included in the Container as a Service (CaaS) category.

FAMILY	SUB-FAMILY	LIST OF SERVICES
Compute	Confidential - Kubernetes - Private	Kubernetes Confidential Computing

### 3.1 Compute Family

Below are the sub-families belonging to the Compute family:

- Confidential - Kubernetes - Private

#### 3.1.1 Confidential - Kubernetes - Private Sub-Family

Below is the list of services within the Confidential - Kubernetes - Private Sub-family:

- Kubernetes Confidential Computing

##### 3.1.1.1 Kubernetes Confidential Computing Service

###### > Service Description

This service provides a platform for orchestrating private and secure containers, designed to manage containerized applications in highly regulated environments or with confidentiality requirements.

It offers a secure and controlled Kubernetes environment where security is a key aspect of the solution.

The operating system on which the solution is based is hardened to minimize the attack surface and potential vulnerabilities.

The solution's architectural components utilize mechanisms that ensure data security, including during communication (via encryption mechanisms applied by default to communications between platform components) and for data stored within the platform itself. The platform can be customized to adapt to the specific needs of each organization, ensuring integration with existing enterprise systems and applications.

###### > Features and Advantages

Implementation requires a combination of hardware certified for Confidential Computing, a private, security-hardened Kubernetes infrastructure, and a suite of observability and governance tools to maintain complete control over the container lifecycle.

Features included:



- *Data protection* → The operating system is configured to ensure protection at all stages: data in memory, through full disk encryption and key rotation; data in transit, using secure and encrypted communication protocols; and data in use, adopting Confidential Computing practices and secure execution environments.
- *Secure enclaves* → Enforces isolation and encryption, ensuring that only authorized parties can access data.
- *Trusted execution environments (TEEs)* → Adds a secure computing environment, protecting data from external threats.
- As a managed Kubernetes solution, the customer does not have to worry about managing the infrastructure and its complexity, as the infrastructure layer is managed by Leonardo throughout the service lifecycle.

The service offers the following advantages:

- *Security and confidentiality of containerized applications* → end-to-end encryption, confidential computing for workloads, container isolation on dedicated nodes with hardware-based protection, integrated security policies, and advanced RBAC.
- *Centralized cluster control and governance*.
- *Scalability and flexibility*.
- *Integration with multicloud and legacy environments*.



## 4 Platform as a Service (CaaS)

The following table lists the services included in the Platform as a Service (CaaS) category.

FAMILY	SUB-FAMILY	LIST OF SERVICES
Compute	FAAS	Functions as a Service
Security	IAM	Identity & Access Management Service
Security	Key Management	Key Vault as a Service
Middleware	API Platform	PaaS API Management
Middleware	APP Runtime	Jboss as a Service
Middleware	APP Runtime	Spring boot as a Service
Middleware	BPM	PaaS Business Process as a Service
Middleware	CMS	PaaS CMS as a Service
Middleware	ETL	PaaS ETL - Batch / Real Time Processing - 1 worker
Infra & Ops Platform	Multicloud Management	Multicloud Management Platform-Leonardo SCMP
Infra & Ops Platform	Multicloud Management	Multicloud Management Platform-Morpheus
Infra & Ops Platform	Observability-Infra	Control Room as Service
Infra & Ops Platform	Observability-Infra	IT infrastructure Service Operations (Logging & Monitoring)
Infra & Ops Platform	TTM	PaaS Ticket Management Service
Infra & Ops Platform	TTM	PaaS Ticket Management Service (ITSM)
Infra & Ops Platform	TTM	PaaS Ticket Management Service (ADD-ON ITOM)
DevSecOps	CI	Configuration Manager
DevSecOps	CI	Test Automation
DevSecOps	CI	Quality Code Analysis



FAMILY	SUB-FAMILY	LIST OF SERVICES
DevSecOps	CI/CD	DevSecOps As A Service By PSN
DevSecOps	CI/CD	Qualizer DevSecOps
Big Data	Data Lake	Data Lake - 1TB
Big Data	Data Lakehouse	Data Lakehouse
Big Data	Business Intelligence	Business Intelligence
Big Data	ETL	Batch/Real time Processing - 1 Worker
Big Data	Event Platform	Event Message
Big Data	Data Governance	Data Governance
AI	AI - Audio & Conversazioni	Speech to Text
AI	AI - Image	OCR
AI	AI - Text	AI Search - AI Search - RAG - 10 GB - 1 worker
AI	AI - Text	Text Analytics
AI	AI - Text	Translation
AI	AI - Generativa	AI SLM/LLM
AI	AI - Tools	AI workflow
AI	AI - Tools	Vector DB
AI	AI - Tools	AI Platform
VDI	Virtual Desktop	VDI
VDI	Virtual Desktop	VDI with GPU support
Collaboration	Communication	Instant Messaging

## 4.1 Compute Family



Below are the sub-families belonging to the Compute family:

- FAAS

#### 4.1.1 FAAS Sub-Family

Below is the list of services within the FAAS Sub-family:

- Functions as a Service

##### 4.1.1.1 Functions as a Service

###### > Service Description

FaaS (Function as a Service) is an event-driven system design model running on stateless containers, where developers create, deploy, and execute small, independent functions to perform specific tasks without worrying about the underlying infrastructure.

Adopting FaaS allows for standardization of application development and execution by centralizing cross-functional capabilities such as orchestration, automatic provisioning, monitoring, integrated service management, and event-driven flow control.

It offers tools to:

- centrally manage serverless functions;
- automate component lifecycle management;
- enable multi-cloud and hybrid cloud portability;
- support innovation with GPU runtimes and dedicated AI tools.

The FaaS platform provisions and scales the underlying resources based on demand. It is ideal for highly dynamic scenarios with variable workloads and integrates seamlessly with microservices and event-based architectures.

###### > Features and Advantages

The service goes beyond simply providing an execution engine; it also offers a complete ecosystem, consisting of:

- Serverless execution → stateless functions and event-driven workflows, scalable and available in various programming languages.
- Portability and independence → can run on any Kubernetes cluster, across multiple environments, without lock-in constraints.
- Security and compliance → data protection and centralized access management.
- The solution enables organizations to adopt a modern and flexible model, reducing operational complexity and



benefiting from a standardized and easily accessible service.

The service is delivered through Apache OpenServerless, an open-source, cloud-agnostic serverless platform based on Apache OpenWhisk as a Function-as-a-Service (FaaS) engine.

The service offers the following advantages:

- *Reduced operating costs* → you only pay for the actual use of features.
- *Flexibility and scalability* → resources adapt to demand.
- *Operational efficiency* → eliminating the need to directly manage servers, patches, and updates.
- *High availability* → built-in redundancy and fault tolerance, ensuring high availability of features even in the event of hardware failures or other interruptions.
- *Accelerated time-to-market* → rapid release of new features without worrying about the infrastructure.
- *Agile development* → focus on code and business logic, not server management.
- *Continuous innovation* → rapid experimentation with new, low-cost services. Competitive advantage in cost and speed compared to traditional hosting models.

#### 4.1.2 IAM Sub-Family

Below is the list of services within the IAM Sub-family:

- Identity & Access Management Service

##### 4.1.2.1 Identity & Access Management Service

###### > Service Description

The Service provides an essential level of security for identity and access management, ensuring basic protection against unauthorized access.

It manages single sign-on access to guarantee access to all protected resources with a single authentication. It supports standard OIDC/OAUTH and SAML protocols for easy integration with applications and products.

It enables first-level authentication with username/password and second-level authentication with multi-factor authentication based on Time-based One-Time Password (TOTP) protocols.

It manages access authorization to system-protected resources only for users with rights to use them according to the Role-based Access Control (RBAC) and Attribute-based Access Control (ABAC) paradigms. Integration with external user repositories (LDAP or Active Directory) is also available.

It manages the user lifecycle and related authorizations via the console.

###### > Features and Advantages



The main features and functionalities of the service are:

- *Identity Management*

- User Management → creation, modification, and deletion of users; management of user profiles (name, email, custom attributes, roles, etc.); import/export of users from external directories (LDAP, Active Directory).
- Identity Federation → integration with external providers via LDAP or Active Directory; two-way or one-way synchronization of users and roles.
- Account Management UI → self-service portal for users to update profiles and passwords, manage devices and active sessions, and view permissions.

- *Access Management*

- Single Sign-On (SSO) / Single Logout (SLO).
- Multi-Factor Authentication (MFA).
- Delegated Authentication (Identity Brokering).
- Role-Based Authorization (RBAC) and policies.

- *Protocol and Integration*

- Support for standard protocols, such as OpenID Connect (OIDC), OAuth 2.0, and SAML 2.0.
- Official adapters for Java, Spring Boot, WildFly, Node.js, and other applications.
- Ability to integrate with API Gateways, microservices, and web frontends.

- *Security and Management*

- Session and Token Management.
- Password Policies.
- Events and Auditing.
- Scalability and High Availability → distributed architecture, with support for clustering and replication.

- *Extensibility*

- REST API for automated user, role, and client management.
- SPI (Service Provider Interfaces) for extending authentication, validation, or provisioning capabilities.
- Ability to implement custom authenticators or connect to external systems.

The service offers the following advantages:

- *Improved overall security* → Centralizing authentication reduces the risk of vulnerabilities distributed across applications.
- *Reduced maintenance and development costs* → A single, centralized platform reduces the complexity and



duplication of authentication code across applications.

- *Agility and Scalability* → Increased speed of onboarding new applications thanks to the use of standard protocols (OIDC, SAML, OAuth2).
- *Maintainability and Standardization* → Use of standard protocols (OIDC, SAML, OAuth2) that eliminate proprietary implementations and facilitate interoperability.

#### 4.1.3 Key Management Sub-Family

Below is the list of services within the Key Management Sub-family:

- Key Vault as a Service

##### 4.1.3.1 Key Vault as a Service

###### > Service Description

The service provides a secure cloud repository (Vault) for storing and managing credentials and passwords used by cloud applications without having to manually install and manage dedicated IaaS machines. The service consists of a software platform that enables centralized and automated management of encryption keys, secrets, and certificates, with access controlled by identity-based authentication and authorization methods.

It also allows organizations to significantly simplify key lifecycle management, ensuring centralized control while leveraging the native cryptographic capabilities of KMS providers.

###### > Features and Advantages

The main features and functionalities of the service are:

- *Secure Secret Storage* → Key/value secrets are stored in Key Vault As A Service in encrypted form, ensuring their integrity in the event of unauthorized access to raw storage.
- *Dynamic Secrets* → Key Vault As A Service can generate secrets on demand to allow users and/or applications to access different systems.
- *Data Encryption* → Key Vault As A Service can encrypt and decrypt workloads running on the PA infrastructure without archiving them, managing the entire lifecycle of the cryptographic material used in the encryption process.
- *Leasing and Renewal* → Key Vault As A Service associates a lease with each key or secret managed, which will result in its automatic revocation upon expiration and which can be renewed by clients through the integrated APIs provided by the platform.
- *Revocation* → Key Vault As A Service has integrated support for revoking keys and secrets, which can be revoked individually or in bulk (e.g., all keys of a specific user), for example in case of compromise.



The service is provided using Hashicorp Vault technology. The service offers high availability and geographic replication.

The main workflow of Key Vault as a Service consists of four phases:

- *Authentication* → The process by which a client provides information that Key Vault as a Service uses to determine the authenticity of the requester. Once the client is authenticated, the system generates a token that is associated with the relevant policy.
- *Validation* → Validation occurs through trusted third-party sources, such as Active Directory, LDAP, and Okta.
- *Authorization* → The client is then associated with the Key Vault as a Service security policy, which consists of a set of rules that define which API endpoints a user, machine, or application is allowed or denied access to with its token.
- *Access* → Key Vault as a Service then grants access to keys and encryption features, secrets, and certificates.

The service offers the following advantages:

- *Risk reduction* → thanks to automatic key rotation and secret lifecycle management, it increases the protection of sensitive data, simplifies regulatory compliance and reduces the risk of human errors.
- *Operational efficiency and cost reduction* → less internal management, automation and standardization, scalability without hardware investment.
- *Optimized time-to-market* → developers focus on code, not key management; also enables secure applications to be delivered faster, improving agility and innovation.
- *Improved trust and reputation* → audit and traceability to demonstrate secure secret management to stakeholders or customers.
- *Cryptographic and standardized compliance* → can be configured to use FIPS (Federal Information Processing Standards) validated cryptographic modules, ensuring that all encryption, signing, HMAC and key derivation operations comply with the standards.

## 4.2 Middleware Family

Below are the sub-families belonging to the Compute family:

- API Platform
- APP Runtime
- BPM
- CMS
- ETL



#### 4.2.1 API Platform Sub-Family

Below is the list of services within the API Platform Sub-family:

- PaaS API Management

##### 4.2.1.1 PaaS API Management

###### > Service Description

It is a platform of tools and services that facilitates the management, control, monitoring, and protection of APIs (Application Programming Interfaces) without having to manually implement all the components. The service typically offers:

- API gateways to route and secure traffic;
- Authentication and authorization: Rate limiting and throttling to control consumption;
- Logging and observability: Integration with security and DevOps systems.

The API manager facilitates API lifecycle management, including aspects such as creation, version management, deprecation, and retirement, to ensure backward compatibility, allowing developers to gradually migrate to new versions without disrupting existing applications.

The API manager allows you to define and enforce policies, such as usage limits, quota management, custom authentication, data transformations, and caching. These policies allow you to control API behavior and ensure compliance with security requirements and guidelines.

The API Manager can integrate with other systems and tools, such as identity and access management (IAM) systems, performance monitoring systems, data analytics systems, and security gateways. This integration expands the API Manager's functionality and integrates it into the ecosystem of existing applications and services.

###### > Features and Advantages

The main features and functionalities of the service are:

- *API Publishing* → the API Manager offers tools for publishing APIs, allowing developers or authorized users to access them. For optimal use, clear and comprehensive documentation is provided describing how to use the APIs, which endpoints are available, which parameters are requested, and how to interpret the responses.
- *Access Control* → the API Manager manages the authentication and authorization of users who wish to use the APIs. This allows you to control who can access the APIs and with what permission levels. The API Manager can adopt authentication mechanisms such as access tokens, API keys, or digital certificates to ensure API security.
- *Monitoring and Analytics* → the API Manager offers tools for monitoring API performance, such as the number of requests, response times, and errors. This information allows developers and administrators to monitor API usage, identify any performance issues, and take corrective action.



The architecture, based on Kong technology, is divided into several key components that interact to provide comprehensive functionality to users:

- *Front-end* → administration clients and graphical interfaces (Admin GUI, Dev Portal) accessible via browser or dedicated applications, which allow users to configure services, manage users, and monitor metrics in real time.
- *Back-end Kong Control Plane* → manages configurations, policies, plugins, and API orchestration.
- *Back-end Data Plane* → routes user requests to back-end services, applying security rules, transformations, caching, and rate limiting. - *Database* → stores configurations, users, roles, statistics, and logs. Supports replication and high availability capabilities to ensure resilience and business continuity
- *Integrations* → supports integrations with development tools, CI/CD, monitoring systems, and project management platforms, allowing Kong to be incorporated into existing enterprise workflows.
- *Security and Authentication* → offers advanced security options, including multi-factor authentication, support for enterprise protocols (OIDC, SAML, LDAP), and granular access control, ensuring data protection and compliance with corporate standards.

The service offers the following advantages:

- *Reduced time to market* → APIs can be published and managed quickly without building the infrastructure from scratch.
- *Flexibility and scalability* → the platform grows with business needs, supporting traffic spikes or new integrations without disruption.
- Reduced operating costs → no hardware or maintenance investments: infrastructure management is delegated to the PaaS provider.
- *API monetization* → ability to create API-driven business models (e.g., exposing APIs to partners or customers with pricing plans).
- *Enhanced security and compliance* → secure management of APIs and traffic between services, with authentication, authorization, and rate limiting policies, protecting the infrastructure from unauthorized access.
- *Open ecosystem* → Facilitates partnerships and innovation thanks to an API-ready and standardized infrastructure.

#### 4.2.2 APP Runtime Sub-Family

Below is the list of services within the APP Runtime Sub-family:

- Jboss as a Service
- Spring boot as a Service



#### 4.2.2.1 Jboss as a Service

##### > Service Description

The service is based on an open source platform for running and managing Enterprise Java applications, designed to offer reliability, scalability, and flexibility in modern environments. It allows to run Java EE/Jakarta EE applications and microservices, providing a robust environment for business logic, data persistence, and transaction management.

It allows to manage the application lifecycle, including deployment, updates, rollbacks, and centralized configuration, ensuring secure and repeatable processes.

Thanks to its modular architecture, compatibility with cloud environments, and rich integration with automation and security tools, it represents a strategic solution for companies seeking efficiency, innovation, and operational control.

##### > Features and Advantages

JBoss offers a robust, high-performance, and secure environment for developing and managing enterprise applications, providing a stable foundation for the growth and evolution of enterprise systems.

The main features and functionalities of the service are:

- *Security and Compliance* → manages security, authentication, authorization, and data protection.
- *Web Services* → JAX-RS, JAX-WS, creation and management of RESTful and SOAP APIs for service integration.
- *Microservices Management* → MicroProfile, a set of specifications optimized for developing microservices-based applications. Includes features such as configuration, resiliency, monitoring, and metrics.

The architectural components of the service are as follows:

- *Front-end* → administration interfaces (Web Console, CLI) accessible via browser or terminal, which allow administrators to manage configurations, deployment, resources, and monitoring.
- *Back-end* → the server core manages application execution, request processing, resource management (datasources, JMS queues, batch, etc.), and integration with external systems via resource adapters and connectors.
- *Database* → integrates with relational and NoSQL databases via configurable datasources, used by applications for data persistence.
- *Security and Authentication* → offers an advanced security subsystem for authentication, authorization, encryption, and auditing. It supports authentication via LDAP, Kerberos, SSO, and integration with external identity providers, ensuring secure access that complies with corporate standards.

The service offers the following advantages:

- *Reduced time to market* → application lifecycle automation, centralized management, and easy integration with DevOps pipelines reduce development and release times, accelerating response to market needs.



- *Reduced operating costs* → centralized resource management and the platform's modularity optimize the use of existing infrastructure, reducing waste and operating costs.
- *Security posture* → security policies can be defined and applied consistently across all applications, reducing risk and ensuring regulatory compliance.
- *Faster innovation* → management tools (CLI, Web Console, REST API) and automated deployment and configuration processes reduce the operational burden on IT teams.
- *DevOps integration* → integrated CI/CD pipelines for build and deployment.

#### 4.2.2.2 Spring boot as a Service

##### > Service Description

This service allows you to use Spring Boot, an open-source framework for Java application development, as a managed service.

It is designed to simplify the development of production-ready Java applications by providing a platform that eliminates much of the manual configuration required by the traditional Spring framework and reduces the need for server provisioning and dependency management.

With a preconfigured environment optimized for the Spring Boot framework, the service allows teams to focus on developing business features, reducing release times and costs.

It integrates with DevOps tools and leading cloud services, offering scalability, managed updates, and continuous monitoring.

##### > Features and Advantages

The main features and functionalities of the service are:

- *Automatic environment provisioning* → automatic configuration of Java runtime (JDK), integrated application server, and Spring Boot framework. No need to manually configure build environments or containers. Simplified deployment → ability to directly upload a JAR or source code (e.g., via Git, API, or CI/CD pipeline).
- *Scalability* → horizontal (replication) and vertical (CPU/RAM resources) scaling managed by the PaaS based on load.
- *Integrated monitoring and logging* → access to runtime metrics (CPU, memory, latency, throughput); centralized logs (stdout/stderr) accessible via console or API; integration with BI tools (Prometheus, Grafana, etc.).
- *Configuration and secret management* → centralized configuration (environment variables, Spring Cloud Config, or Vault); secure management of credentials, tokens, and keys. Integrated support services → easy connection to managed databases (PostgreSQL, MySQL, MongoDB); support for messaging (RabbitMQ, Kafka), caching (Redis), and storage; automatic service binding via environment variables or injection.
- *DevOps integration* → support for CI/CD pipelines; continuous deployment (Continuous Deployment) and



automatic rollbacks; compatibility with tools such as GitHub Actions, Jenkins, GitLab CI.

- *Security and isolation* → each application is isolated (namespace, container, or dedicated VM); HTTPS/TLS by default, identity management, and integration with authentication systems (OAuth2, SSO).

The solution is based on the following architectural layers:

- *Infrastructure layer* → provides the hardware and virtual resources needed to run application containers (Compute nodes, Storage, Networking, Security layer); automatic provisioning via IaC (Infrastructure as Code).
- *Orchestration layer* (Platform Runtime) → manages the lifecycle of Spring Boot containers, from deployment to monitoring, ensuring availability, replication, and load balancing
- *Application layer* (Spring Boot Runtime) → Spring Boot runs within a container; supports Actuator endpoints for health checks and metrics; exposes HTTP/REST APIs on predefined and configurable ports
- *Management layer and PaaS services* → web dashboard or CLI to manage applications, versions, and resources. REST API for automation (deployment, scale, logs, metrics). Integration with external logging and monitoring systems.

The service offers the following advantages:

- *Reduced time to market* → Deployment automation and simplified environment management allow applications to be brought into production more quickly.
- *Reduced operating costs* → No hardware or maintenance investments: infrastructure management is handled for the customer.
- *Observability and monitoring* → Preconfigured tools to track performance, errors, and response times.
- *Guaranteed security* → Automatic patch and update management.
- *Environment consistency* → Same environments for development, testing, and production.
- *Microservices support* → Simplified management of distributed architectures.

#### 4.2.3 BPM Sub-Family

Below is the list of services within the BPM Sub-family:

- PaaS Business Process as a Service

##### 4.2.3.1 PaaS Business Process as a Service

###### > Service Description



It is a comprehensive Business Process Management (BPM) platform that helps companies model and automate complex processes, improve productivity and service quality, and ensure control, traceability, and flexibility in an integrated and scalable environment.

It combines workflow automation, application integration, and performance monitoring in a single solution. The goal is to improve operational efficiency, reduce execution times, and ensure process consistency across the organization.

It facilitates collaboration between business users and IT during the creation, management, validation, and deployment of customized process and decision automation solutions. Business users can modify business logic and business processes without requiring assistance from IT staff.

### > Features and Advantages

The main features and functionalities of the service are:

- *Process Modeling & Simulation* → allows business analysts and developers to collaborate on process definition using a standard language (BPMN 2.0) with drag-and-drop tools.
- *Process Automation & Orchestration* → allows for the automation of repetitive tasks and decision rules.
- *Human Workflow Management* → automatic assignment of tasks based on roles, priorities, and workloads. Intuitive user portal for completing, delegating, or commenting on tasks.
- *Monitoring, Reporting & Optimization* → real-time dashboard for performance analysis based on KPIs and SLAs, reporting, optimization recommendations through predictive analytics, and historical data.
- *Security & Governance* → integrated authentication with LDAP/Active Directory. Granular roles for users and groups (process owner, approver, admin). Complete audit trail for compliance and traceability. Version control and approvals prior to deployment.
- *Cloud & DevOps Integration* → offered as a managed cloud service. Integration with CI/CD pipelines and DevOps tools.

The service, based on IBM technology, is organized into the following integrated modules that cover the entire process lifecycle—from modeling to performance measurement.

- *Process Designer* → Visual process modeling tool.
- *Process Center* → Centralized repository and collaborative environment, allows you to manage multiple versions of processes, reuse common components, and collaborate across multiple teams.
- *Process Server* → Process execution engine. Manages both human and automated tasks.
- *Process Portal* → User portal for receiving, executing, or approving tasks.
- *Performance Data Warehouse (PDW)* → Performance collection and analysis system, stores process execution data and enables historical analysis and real-time monitoring.



The service offers the following advantages:

- *Operational efficiency and cost reduction* → automation and reduction of manual and repetitive tasks, resulting in reduced personnel costs, errors, and inefficiencies - *Transparency and control* → end-to-end visibility. Each process is tracked in real time. Increases accountability and control. - *Quality and standardization* → consistent and compliant processes. Ensures processes are always executed consistently, reducing deviations and variability. - *Compliance and auditability* → complete traceability for audits and regulatory compliance. Every step and decision is documented, facilitating internal controls and regulatory compliance. - *Monitoring and observability* → integrated dashboards and analytics

#### 4.2.4 CMS Sub-Family

Below is the list of services within the CMS Sub-family:

- PaaS CMS as a Service

##### 4.2.4.1 PaaS CMS as a Service

> **Service Description**

> **Features and Advantages**

#### 4.2.5 ETL Sub-Family

Below is the list of services within the ETL Sub-family:

- PaaS ETL - Batch / Real Time Processing - 1 worker

##### 4.2.5.1 PaaS ETL - Batch / Real Time Processing - 1 worker

> **Service Description**

> **Features and Advantages**