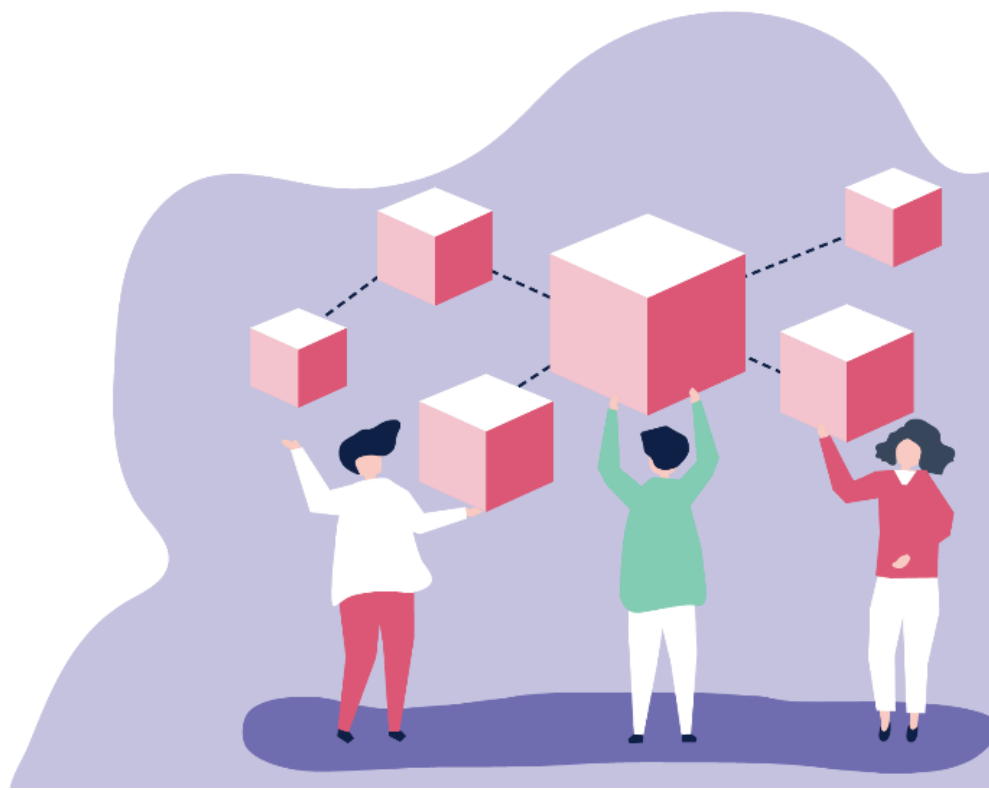




D4.1 Microservices Containerization & Deployment

Deliverable No.	D4.1	Due Date	31/03/2020
Description	Microservices Containerization & Deployment		
Type	Other	Dissemination Level	PU
Work Package No.	WP4	Work Package Title	GATEKEEPER Things Management Infrastructure & Development
Version	1.0	Status	Final



Authors

Name and surname	Partner name	e-mail
Claudio Caimi	HPE	claudio.caimi@hpe.com
Mirko Manea	HPE	mirko.manea@hpe.com
Patrizia Ciampoli	HPE	patrizia.ciampoli@hpe.com
Sabino Minervini	HPE	sabino.minervini@hpe.com

History

Date	Version	Change
01/02/2021	0.1	Table of content and initial content
16/02/2021	0.2	Revision of ToC and contributions
28/02/2021	0.3	Integration of content up to the date
12/03/2021	0.4	Integration of 1 st round of contributions
22/03/2021	0.5	Integration of contributions
24/03/2021	0.6	Incorporated comments from internal reviewers
31/03/2021	1.0	Revision of content after quality review

Key data

Keywords	Data Centre Infrastructure
Lead Editor	<i>See Authors</i> (HPE)
Internal Reviewer(s)	Carlo Allocca (SAM), Alessio Antonini (OU)

Abstract

This deliverable contains a description of the GATEKEEPER Infrastructure that will host the GATEKEEPER Platform to allow the Pilots executions.

It describes the provision and setup of the Cloud Services, their access mechanisms and security measures, including the Continuous Integration and Continuous Deployment

tools, and the Containerization Techniques and software that will help the GATEKEEPER partners to develop the GATEKEEPER Platform.

The GATEKEEPER Infrastructure will be complemented with the Big Data Infrastructure (Task 4.3), that will be released as part of Deliverable D4.3. A new updated version of this document is expected at M30 and finally at M40, describing the final version of the infrastructure.

This deliverable accompanies the release of the actual hardware and software release of cloud service infrastructure for the GATEKEEPER platform.

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Acronyms

Table 1: List of acronyms

Acronym	Description
2FA	Two-Factor-Authentication
CA	Certification Authority
CE	Community Edition
CPE	Common Platform Enumeration
CPU	Central Processing Unit
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
DoW	Description of Work, i.e. the GATEKEEPER proposal document
DC	Data Centre
DDI	DNS, DHCP, and IPAM
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
GK	GATEKEEPER
iSCSI	Internet Small Computer Systems Interface
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPAM	IP Address Management
I/O	Input/Output
IT	Information Technology
IRF	Intelligent Resilient Framework
KVM	Kernel-based Virtual Machine
LDAP	Lightweight Directory Access Protocol
NIC	Network Interface Card
OS	Operating System
OSS	Open Source Software
OWASP	Open Web Application Security Project

PKI	Public Key Infrastructure
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RHEL	Red Hat Enterprise Linux
SCA	Software Composition Analysis
SMTP	Simple Mail Transfer Protocol
ToC	Table of Content
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network

Table of contents

TABLE OF CONTENTS.....	7
LIST OF TABLES	8
LIST OF FIGURES	9
1 INTRODUCTION.....	10
2 GATEKEEPER DATA CENTRE INFRASTRUCTURE	11
2.1 PROVISION AND SETUP OF CLOUD SERVICES	11
2.1.1 Data Centre Layout.....	11
2.1.2 Security Organizational Measures.....	15
2.1.3 Security Technical Measures.....	20
2.2 CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY	23
2.2.1 CI/CD Strategy & Architecture	23
2.2.2 CI/CD Tools.....	26
2.3 CONTAINERIZATION TECHNIQUES	28
2.3.1 Introduction to Containers: Hypervisor vs Container.....	28
2.3.2 Characteristics of Containers	29
2.3.3 Containers and DevOps.....	31
2.3.4 Containers and Micro-services.....	31
2.3.5 Docker.....	32
2.3.6 Container orchestration.....	33
3 GATEKEEPER USER MANUALS	37
3.1 GATEKEEPER DATA CENTRE INFRASTRUCTURE	37
3.1.1 Data Centre Access User Manual.....	37
3.2 CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY	37
3.3 CONTAINERIZATION TECHNIQUES	38
3.3.1 Container Workload Orchestration - OKD User Manual.....	38
4 CONCLUSIONS	39
5 REFERENCES	40
APPENDIX A ANNEXES.....	42
APPENDIX B USER REGISTRATION FORM TEXT.....	43
APPENDIX C USER REGISTRATION FORM FIELDS.....	44
APPENDIX D INTERNAL HPE PORTAL HOME PAGE	46

List of tables

TABLE 1: LIST OF ACRONYMS	5
TABLE 2: DATA CENTRE USER ACCESS ROLES.....	19

List of figures

FIGURE 1 – DC PHYSICAL LAYOUT	12
FIGURE 2 – DC LOGICAL LAYOUT	13
FIGURE 3 – DC RECOGNITION PLATE FOR EU H2020	15
FIGURE 4 – PREFACE OF THE USER REGISTRATION FORM	17
FIGURE 5 – FIELDS OF THE USER REGISTRATION FORM	18
FIGURE 6 – INTERNAL HPE PORTAL.....	19
FIGURE 7 – INTERNAL CHANGE PASSWORD SERVICE	21
FIGURE 8 – VPN / AUTHENTICATION / AUTHORIZATION DIAGRAM	22
FIGURE 9 – LOG MANAGEMENT SERVICE.....	22
FIGURE 10 – SECDEVOPS CI/CD PRACTICE.....	24
FIGURE 11 – CI/CD STAGES.....	25
FIGURE 12 – HYPERVISOR VS CONTAINER	29
FIGURE 13 – CONTAINER FILESYSTEM ISOLATION	30
FIGURE 14 – IMAGE LAYERS OF A CONTAINER	31
FIGURE 15 – MONOLITHIC VS MICRO-SERVICES (MS) BASED APPLICATION.....	32
FIGURE 16 – DOCKER SWARM.....	34
FIGURE 17 – KUBERNETES.....	35
FIGURE 18 – SEGREGATION FOR GATEKEEPER PILOTS	36
FIGURE 19 – GATEKEEPER-WP4-GK_DATA_CENTRE_ACCESS DOCUMENT ToC	37
FIGURE 20 – OKD WEBINAR FOR GK PARTNERS.....	38

1 Introduction

As outlined in the DOW for WP4, in particular T4.1 and starting from the requirements collected in T3.1, informed by user requirements collected in T2.3, together with the plans of the pilots to use the platform identified in T6.2, the requirements of the technology providers gathered in T3.4 (and reported in D3.4 which have been used to define the capacity needs), and the components descriptions of WP4 and WP5, this deliverable aims to define the overall GATEKEEPER infrastructure architecture, describing the components and services that build the HPE GK Data Centre.

The GATEKEEPER infrastructure at the time of this deliverable (M18) provides i) the *"provision and setup of high-performance cloud services"*, ii) the preliminary work on *"definition of the continuous integration and continuous delivery pipelines"*, and iii) the initial setup of *"the containers clustering techniques and orchestration mechanisms"*. In fact, the GATEKEEPER infrastructure is going to host the GATEKEEPER platform, including the components delivered in WP4 such as the Thing Management System (output work of T4.2, and preliminary described in D4.2 at M12), the Data Federation Framework (output work of T4.4, and preliminary described in D4.4 at M15), and the GATEKEEPER Trust Authority (output work of T4.4, and preliminary described in D4.5 at M12) that together constitute the Core Platform of the GATEKEEPER overall architecture as outlined in deliverables D3.2.x series of WP3.

This task results have been organized following the structure described below:

Section 1 – it this is introduction.

Section 2 – focuses on describing the Data Centre setup, with the provided tools and services.

Section 3 - reports the user manuals useful for the GATEKEEPER partners to access and work with the Data Centre tools and services.

Section 4 - shows the conclusions and future work.

Section 5 - is the references.

Finally, **Appendices** show the collateral material (annexes) and additional information.

2 GATEKEEPER Data Centre Infrastructure

This document describes the Data Centre Infrastructure where resides the GATEKEEPER Platform with available services and security measures. Here partners can find information and tools for accessing and working in the environment.

The whole datacentre and all its components have been purchased new for the GATEKEEPER project, as it is all the set up and configuration activities.

All hardware equipment is composed by devices manufactured by Hewlett Packard Enterprise (HPE) group. Internet connectivity is provided by a Telecom Provider equipment (company name not reported for privacy and security reasons).

Software resources include HPE software for *infrastructure management* (HPE Oneview), *network* and *storage devices*. For all other components (operating system, virtualization platform and services applications) GATEKEEPER takes advantage of Open Source software except for Big Data and Analytics platform that will be implemented with HPE Ezmeral framework (this latter is however part of T4.3 activities and will be reported at M24 in D4.3).

Operating system and virtualization software are chosen from Open Source projects specifically tailored to enterprise environments. In fact, they come from Red Hat upstream projects and were selected for their stability, security, and long-term support. The general guideline for our Open Source products selection is the possibility, in case of need or opportunity, to migrate them into the correspondent commercial version releases, where specific customer's requirements can be met (e.g. service level agreements and support contracts).

The choice of both hardware equipment and software tools have been a trade-off between security and reliability of advanced hardware technologies (commercial off-the-shelf HPE products), the associated costs, and the openness of the software services.

2.1 Provision and Setup of Cloud Services

This section describes the provision and the setup of the Cloud Services offered to the GATEKEEPER project by HPE partner hosted at its Italian premises.

2.1.1 Data Centre Layout

In this section we describe the configuration of the Infrastructure, where the GATEKEEPER resources reside.

GATEKEEPER Infrastructure is located in a Data Centre hosted in HPE Offices in Rome, Italy. Infrastructure in this Data Centre is organized in separate blocks: one block for general services and for Access/Security/Network/Services management (called Service Block) and one specific Project block that includes also mass storage services (split in Compute and Storage Blocks). GATEKEEPER is hosted in this latter as it is described in the coming sections.

This design follows the industry best practice to have general Data Centre services (Service Block) separated from the business services that we are going to deliver (Compute and Storage Blocks). For example, this allows having system administrators working on DC services that does not need to have access to the infrastructure hosting the GATEKEEPER resources and data, thus improving the overall security of the solution.

Next we describe the GATEKEEPER Infrastructure layout based on component type.

2.1.1.1 Physical and Logical layout

The Physical Layout of the DC is represented in the Figure 1 diagram:

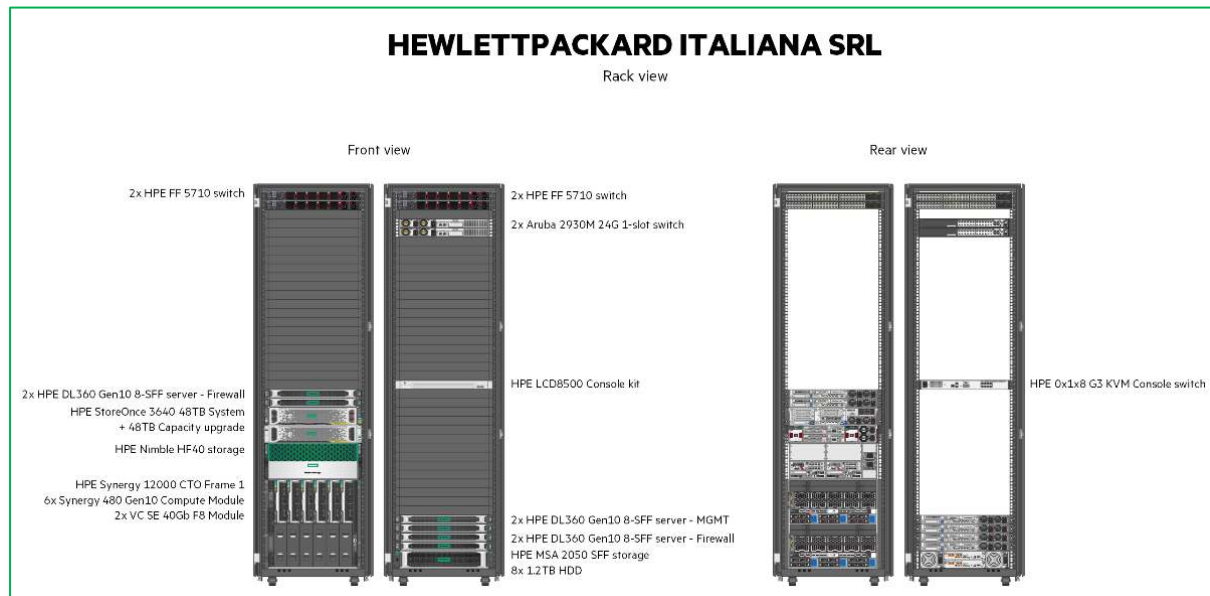


Figure 1 – DC Physical Layout

As you can see in Figure 1, there are two Racks shown both in front and rear view. If you consider the front view, on right side, you can see the resources for general services of the Data Centre (i.e. the Service Block), while on left side, there are the specific resources for GATEKEEPER (i.e. the Compute and Storage Blocks).

All hardware components are redounded to guarantee high-availability, by using software clustering techniques or hardware redundancy (e.g. RAID disks, dual NICs, dual power supplies, stacked switches, etc.). This follows both software specific guidelines (e.g., hypervisor's clustering¹), and the industry HPE product best practices, such as Intelligent Resilient Framework (IRF) technology [1].

¹ oVirt Clusters, https://www.ovirt.org/documentation/administration_guide/#chap-Clusters

Figure 2 shows the Logical Layout where you can see on the right side Service Block components, that host general Data Centre services, and on the left side the GATEKEEPER Block components (Compute and Storage Blocks).

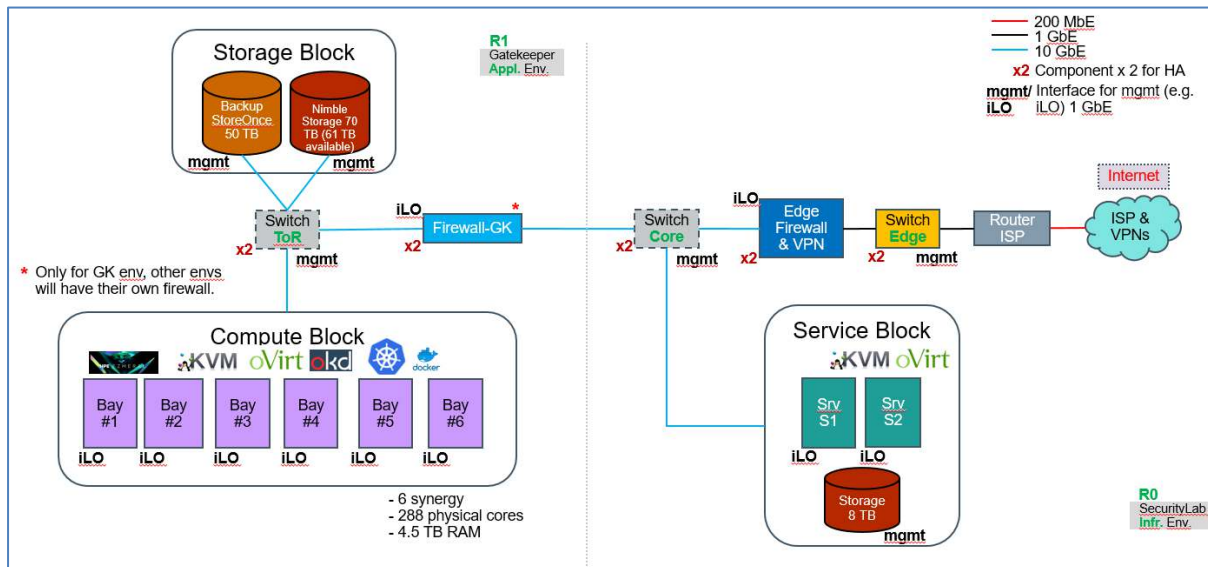


Figure 2 – DC Logical Layout

2.1.1.2 Service Block

Service Block includes all resources needed for DC general services:

- Secure Access:
 - Internet connectivity
 - VPN
 - Edge (perimeter) and Core Firewall
- Identity Management
 - Centralized authentication services based on LDAP technology
- Log Management
 - Tracing of user activities for compliance and security purposes
- Internal Certification Authority (CA)
 - Manage lifecycle of digital certificates for secure communication and identification
- Backup
 - Provide resiliency of the infrastructure by creating copies of software installation and configuration that can be used to recover in case of need (e.g., hardware failures, human errors, security incidents)
- DDI Services
 - Provides DNS, DHCP, and IPAM - IP Address Management services, necessary for the correct functioning of the DC network.
- Monitoring
 - Control of equipment and services health (service to be implemented yet)

2.1.1.3 Compute Block

GATEKEEPER compute resources consists of 6 HPE Synergy servers [1]. Computational power consists in 288 physical CPU cores (576 virtual CPU cores) and 4.5 TB RAM (volatile memory).

These servers have been configured by creating an oVirt [7] cluster that provides computing virtualization services; the underlining operating system is CentOS Linux [6], the Open Source platform derived from Red Hat Enterprise Linux (RHEL). This cluster hosts the following services that will power the GATEKEEPER platform:

OKD [4], the Open Source distribution of Red Hat OpenShift [5], for container orchestration based on the Kubernetes [7] engine;

HPE Ezmeral [8] for Big Data services.

GATEKEEPER services and application will be implemented as computer containers on this compute block. The most widespread technology for containers is based on Docker [9] tools.

2.1.1.4 Storage Block

Mass storage resources are provided by an HPE Nimble Storage [10] connected to the DC network via the iSCSI protocol. Storage has two controllers (for redundancy) and disks for about 70 TB of capacity. HPE Nimble Storage is configured with data encryption, so all data at-rest is kept in encrypted form.

For Backup services GATEKEEPER takes advantage of an HPE StoreOnce appliance that has 50 TB of capacity available for the backup needs of GATEKEEPER environment and its working data.

2.1.1.5 Network and Firewalls services

In addition to the Service, Compute and Storage Blocks, the GK Data Centre makes use of additional equipment to manage the network communication and to secure the DC access.

GATEKEEPER network is realized with two HPE Networking Switches [12] that allow connections between Compute and Storage Block's components and Service Block. GATEKEEPER connection to DC external services is realized with Service Block intermediation and control.

Network is designed with separate VLANs for different scopes (management, application, operation, etc.). In this way we segregate the different network traffic types.

For the whole infrastructure, the network devices are configured with the following specifications:

- IEEE 802.1Q [13] protocol to support **VLAN tagging** on the Ethernet network;
- **Link Aggregation Control Protocol** (LACP – IEEE 802.3ad [14]) to manage the bundling of several physical network ports together to form a single logical channel. In particular this increases the network resiliency in case of hardware faults.

Service Block has two servers [15], which host Firewall services for DC access. These firewalls host also the VPN Service that controls and manages users and application secure access to services and applications.

Compute Block has two servers, who hosts Firewall services specific for GATEKEEPER protection requirements.

2.1.2 Security Organizational Measures

Data Centre access is controlled and managed with these diverse organizational measures:

- Physical access;
- Registration and partner validation process for users;
- Roles definition and assignment for different kind of users.

2.1.2.1 Physical Access

Physical access to Data Centre is allowed to specific HPE personnel authorized by HPE Project Manager and Security/Facility Manager of HPE site. Authorised HPE people have a personal recognition device that can open DC door. All DC access are logged and authorisation list is revised every three months as demanded by HPE internal security policies.

Figure 3 is the design of the plate that is going to be exposed near the DC door to recognize the project and EU funding:

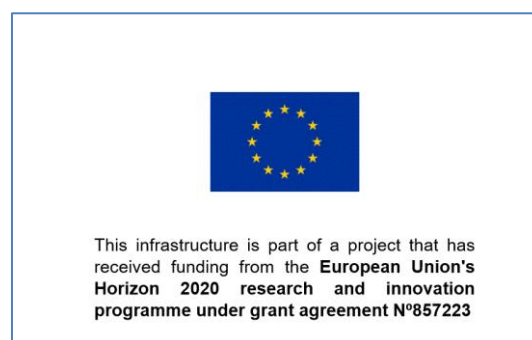


Figure 3 – DC Recognition Plate for EU H2020

2.1.2.2 User Registration Process

For screening the DC user access, it has been defined a process of request, validation and provision of a user registration. Process includes de-provision in case of revocation of user access.

Users are mainly of two kinds:

- HPE administrators;
- GATEKEEPER partner users.

For HPE administrators, who works on infrastructure and management, user authorisations is established by the HPE Project Manager. For all other users, request and validation is controlled by a nominated contact person for each partner and HPE Project Manager of GATEKEEPER.

Request for user registration to DC services is made by partners filing a registration form for each user. HPE validates the form and asks, with an email, confirmation for user and role(s). Then HPE proceeds with the creation of the user account for the specific user's role.

In case of dismissing a role from a user, end of project/role, user or partner request, HPE can disable and/or delete the user. Every 6 months the list of user accounts belonging to each partner is sent to the partner's reference contact point and to the GK coordination team. User accounts that are not confirmed within 2 weeks are disabled and they will no longer be able to access the GK Data Centre. Partner's reference contact point is also asked to proactively notify HPE in a timely manner about any change in their working groups that affects the right of their users to access the GK Data Centre. This will allow HPE to revoke the no longer required account to avoid possible unauthorized access.

Figure 4 shows the preface of the registration form, where the user is informed about the rules of HPE Data Centre access, where also Privacy concerns are addressed (form and text has been approved by HPE Legal Office):

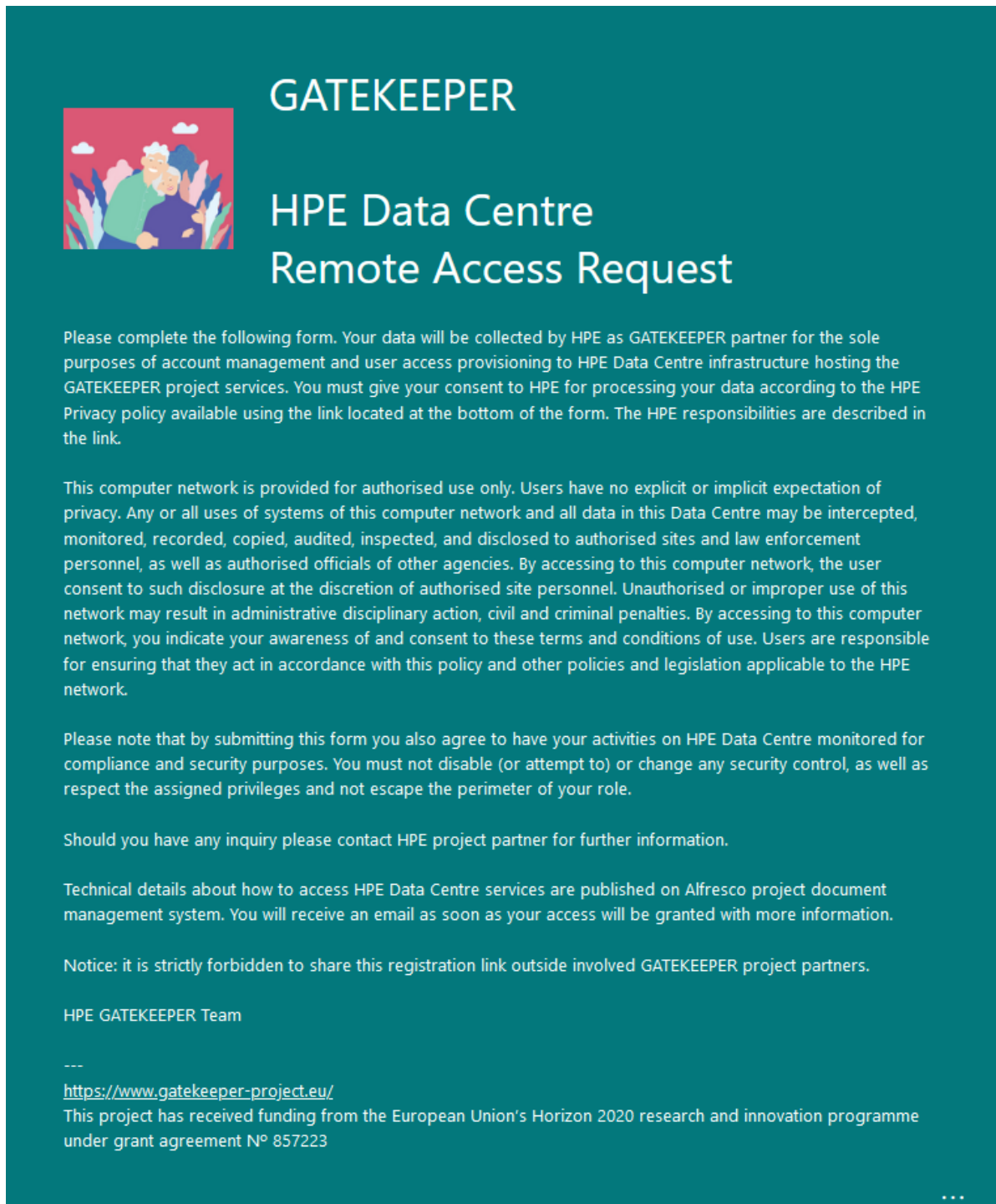
The image shows a registration form preface on a teal background. At the top left is a small illustration of two people, one with a beard and a hat, and another with a beard, standing in front of stylized plants. To the right of this illustration, the word 'GATEKEEPER' is written in large, white, sans-serif capital letters. Below it, 'HPE Data Centre' and 'Remote Access Request' are written in a slightly smaller, white, sans-serif font. The main body of the form contains several paragraphs of white text. The first paragraph explains that the user's data will be collected by HPE as a GATEKEEPER partner for account management and user access provisioning. The second paragraph states that the computer network is for authorized use only and that users have no expectation of privacy; it lists various uses of the network and data, and mentions that unauthorized use may result in disciplinary action. The third paragraph notes that by submitting the form, users agree to have their activities monitored for compliance and security. The fourth paragraph provides contact information for inquiries. The fifth paragraph mentions that technical details are available on the Alfresco project document management system. The sixth paragraph is a notice forbidding the sharing of the registration link. The seventh paragraph identifies the HPE GATEKEEPER Team. The eighth paragraph includes a URL: <https://www.gatekeeper-project.eu/>. The ninth paragraph states that the project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 857223. At the bottom right, there are three small white dots indicating that the form continues.

Figure 4 – Preface of the User Registration Form

For the sake of readability and accessibility, the full text is reported in Appendix B.

The Figure 5 shows the data collection form fields:

* Required

1. First Name *

Please insert your name and optional middle name

Enter your answer

2. Surname *

Please insert your surname

Enter your answer

3. E-Mail Address *

Please specify your e-mail address (e.g. user@example.org). This e-mail address will be used only for communications about HPE data centre. Only valid GATEKEEPER partner e-mail address will be accepted.

Enter your answer

4. Organisation Name *

Please specify your organisation:

Select your answer

5. Your role in the environment *

Notice: your role selection will undergo validation check. Roles explanation follow.

Project User:

- GK-User: access to gatekeeper platform, pilot and other project services

Note: this is not the end user (e.g., patient). GK-User is a regular project partner member

Developer Roles:

- GK-Developer: access to development environment and tools (OKD)
- GK-Data-Scientist: access to big data environment and tools (HPE Ezmeral)

Developers access common services (Gitlab, Jenkins, etc.)

☐ GK-User

☐ GK-Developer

☐ GK-Data-Scientist

6. I have read and I accept the HPE Privacy policy for personal data collection available at the URL: https://www.hpe.com/emea_europe/en/legal/privacy.html *

☐ Yes

You can print a copy of your answer after you submit

Figure 5 – Fields of the User Registration Form

For the sake of readability and accessibility, the full text is reported in Appendix C.

2.1.2.3 User Roles

For GATEKEEPER DC services, HPE has established specific roles as indicated in Table 2:

Table 2: Data Centre User Access Roles

Role	Description
GK-User	Can access the GATEKEEPER platform, pilot and other project services Note: this is not the end user (e.g., patient). GK-User is a regular project partner member.
GK-Developer	Can access the development environment and tools (OKD) where the GATEKEEPER platform is developed.
GK-Data-Scientist	Can access the big data platform environment and tools (HPE Ezmeral) to perform data analytics services execution.

Based on request and expected user activity on the GK platform, HPE assigns specific privileges to the user to allow accessing and using only the required services.

When the user access the HPE GK Data Centre, s/he is welcomed by an internal HPE Portal that provides some hints about the available services (note: some content has been removed due to its security sensitiveness and link is available only internally):

HPE SECLAB

This page provides helpful links for HPE SECLAB made for [GATEKEEPER](#).

Identity Management

Go here to manage your user account.

Access Password Change:

You can change your password or reset it if it expires.

SECLAB Internal Certification Authority

Import the HPE SECLAB into your browser or other services. Download the SECLAB Internal CA to trust the provided services:

For Developers

Tools provided for authorised developers only.

Access OKD Console:

OKD is a distribution of Kubernetes optimized for continuous application development and multi-tenant deployment.

Access HPE Ezmeral for Big Data Console: [soon available](#)

Software platform designed to run both cloud-native and non-cloud native applications in containers. More [info](#).

Important notice

This computer network is provided for authorised use only. Users have no explicit or implicit expectation of privacy. Any or all uses of systems of this computer network and all data in this Data Centre may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to authorised sites and law enforcement personnel, as well as authorised officials of other agencies. By accessing to this computer network, the user consent to such disclosure at the discretion of authorised site personnel. Unauthorised or improper use of this network may result in administrative disciplinary action, civil and criminal penalties. By accessing to this computer network, you indicate your awareness of and consent to these terms and conditions of use. Users are responsible for ensuring that they act in accordance with this policy and other policies and legislation applicable to the HPE network.

About this Project

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857223




(c) Copyright 2021 Hewlett Packard Enterprise Development Company, L.P. Valid agreement required.

Figure 6 – Internal HPE Portal

For the sake of readability and accessibility, the full text of Figure 6 is reported in Appendix C.

2.1.3 Security Technical Measures

Several technical security measures have been implemented to protect the Data Centre infrastructure where the GATEKEEPER platform will run. This process is still ongoing, because these measures are frequently assessed, refined and improved over time, as the global security landscape evolves.

Note: several technical details are not included in this document, because of the sensitivity of such security information. This is mainly due to the Public scope of this document (i.e., Dissemination Level = PU), as specified in the DoW.

2.1.3.1 Secure Access

As mentioned in the previous sections, access to DC is secured via a **VPN connection**. Users must use a VPN client to connect to the DC. The VPN client is based on open security standards and is freely available for numerous platforms, such as desktop operating environments (Microsoft Windows, Apple macOS, Linux) and smartphones (Google Android and Apple iOS). There are VPN clients both open source or commercially supported offered by the market. More details are available in the confidential Annex document (Appendix A).

VPN establishes a secure and encrypted tunnel, which is a communication channel between the user's operating system and the DC VPN server. Data exchange cannot be intercepted in clear or tampered with, and identities of both parties (user and VPN server) are assured thanks to the use of state-of-the-art Public Key Infrastructure (PKI) mechanisms.

At network level, the DC employs stateful network-based firewall services, which track sessions of network connections through them and is able to allow or deny access to specific IP protocol and services. The VPN service is provided by the **Edge Firewall** equipment that protects the outer perimeter. The **Firewall-GK** further regulates the traffic flows towards the GATEKEEPER platform. Edge Firewall and Firewall-GK are shown in Figure 2 – DC Logical Layout.

In addition, at host level, servers and virtual machines use host-based firewall services to further increase the perimeter protection.

Firewalls are the first line-of-defence, but are only one of the components of the defence-in-depth protection strategy and zero-trust model that is in use.

2.1.3.2 Identity Management

User identities are centralized on an LDAP repository in the DC infrastructure. This means that accounts are managed in a central location which ease both the user provisioning and de-provisioning. Access is subject to a two-factor authentication (2FA), which is a form of strong authentication that uses something that only the user knows (a password) and something the user has (a token). Only combining these two pieces of information allows the user to successfully prove its identity to the DC infrastructure and access the environment. The token is a One-Time-Password (OTP) code that is generated by a software application that runs typically on the user's smartphone. The OTP code has a short validity period and changes very frequently over time (in the order of seconds or minutes).

2FA is used to access both the VPN connection and all the available DC services, including OKD and HPE Ezmeral.

As part of the User Registration Process (2.1.2.2), the user is notified of an initial password with a short validity period, which must be changed as soon as possible. HPE developed a service to allow the password change that follows the mandated Password Policy guidelines about password complexity, validity, and so.

Figure 7 shows a screenshot of the internal HPE Portal that provides this service:

Change Password
This service allows you to update your Identity Management password

Change Password
Fill out this form

Username:

Password:

New password:

Retype new password:

Otp code:

Submit the request

Messages
Using the Otp Token?
Make sure you input the numeric code into the Otp code field, otherwise request will fail. This is **mandatory** if you have an Otp token.

Important notice
This computer network is provided for authorised use only. Users have no explicit or implicit expectation of privacy. Any or all uses of systems of this computer network and all data in this Data Centre may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to authorised sites and law enforcement personnel, as well as authorised officials of other agencies. By accessing to this computer network, the user consent to such disclosure at the discretion of authorised site personnel. Unauthorised or improper use of this network may result in administrative disciplinary action, civil and criminal penalties. By accessing to this computer network, you indicate your awareness of and consent to these terms and conditions of use. Users are responsible for ensuring that they act in accordance with this policy and other policies and legislation applicable to the HPE network.

About this Project
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857223

GATEKEEPER
Hewlett Packard Enterprise

(c) Copyright 2021 Hewlett Packard Enterprise Development Company, L.P. Valid agreement required.

Figure 7 – Internal Change Password service

It is worth mentioning that user Authentication is not enough to access the DC services. As explained in Section 2.1.2.3, users have specific roles assigned which enable them to access only the tools they need. This is part of the user Authorization procedures that are in place in the DC.

The diagram in Figure 8 depicts graphically the sequence of VPN Secure Access (Section 2.1.3.1) that requires the Two-Factor Authentication (Section 2.1.3.2) as user identification and then the Role-based Authorization (Section 2.1.2.3) that grants specific access rights.

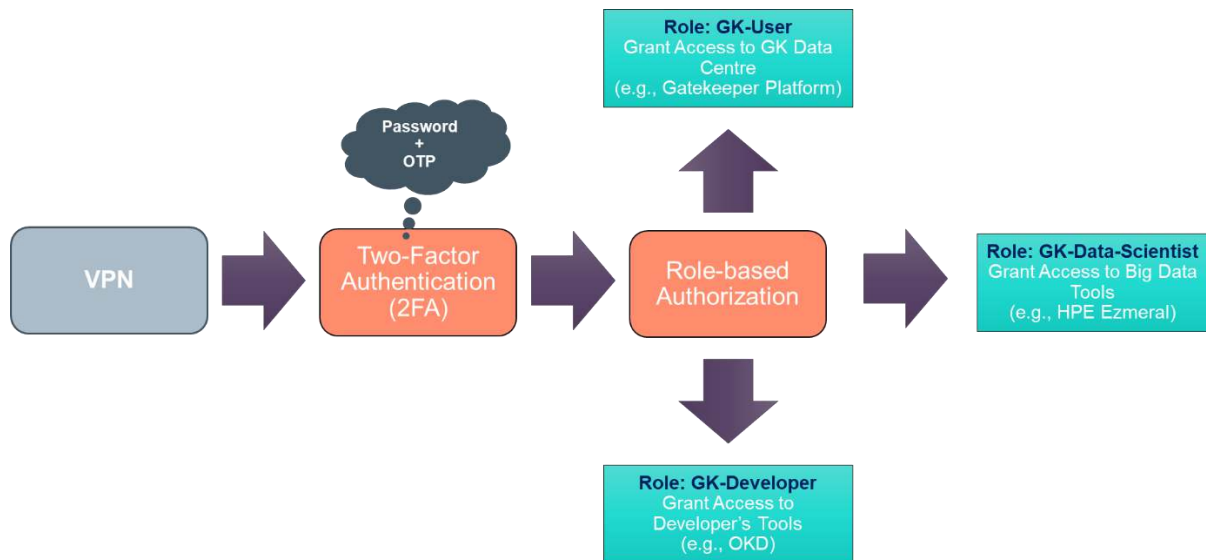


Figure 8 – VPN / Authentication / Authorization diagram

2.1.3.3 Log Management

As you might have seen in the “**Important notice**” reported on the HPE Portal screenshot in Figure 6 – Internal HPE Portal, all user activities is subject to monitoring for security and compliance needs. The Log Management services collects such information and stores log data historically for specific retention periods. Data is kept for investigation purposes in case of security incidents and for compliance reasons related to Privacy regulation.

This service is based on the Open Source Graylog tool and provides capabilities for data search and analysis, helps recognising security attack patterns and provides reporting functionalities.

The screenshot in Figure 9 shows an overview of (anonymized) user authentications collected from the DC activity:

timestamp	event_source_product	event_log_name	event_action	event_outcome	user_id	source_ip	hostname	log_source_address
2021-03-19 18:21:24 +01:00	hpe ilo	syslog	authentication	success				1934
2021-03-19 18:21:23 +01:00	hpe ilo	syslog	authentication	success				1934
2021-03-19 18:21:22 +01:00	hpe ilo	syslog	authentication	success				1937
2021-03-19 18:21:21 +01:00	hpe ilo	syslog	authentication	success				1947
2021-03-19 18:21:20 +01:00	hpe ilo	syslog	authentication	success				1924

Figure 9 – Log Management service

The Log Management service needs to be extended in the next months to be able to recognise attack patterns from the collected log data.

2.1.3.4 Internal Certification Authority

Network communication in the DC is always secured via encrypted connections. For this reason, we implemented a Certification Authority (CA) recognized internally to the DC. The CA is trusted on all the hosts and provides digital certificates for securing all the services that provides web (https) access.

2.1.3.5 Backup

The Backup service is crucial for the resiliency of the whole DC and its services. It provides the ability to recover after hardware failures, security incidents, or human mistakes. We started implementing Backup strategy and solution based on the Open Source tool called Bareos [16]. This enables to save system files, project data, assets configuration and others artefacts such in a way it is possible to quickly restore them in case of need. Bareos has currently been integrated with HPE StoreOnce solution, to store data files and virtual machines snapshots. Integration with all the DC services is undergoing.

2.1.3.6 Additional measures

There are several additional measures that are in use or under evaluation to be implemented during the lifespan of the GK project, also contributed or suggested by Consortium's partners and others are expected to be assessed as more feedbacks are received. In particular, we propose a list below, scheduled to be completed in the next months, with their current status:

- **Web Proxy:** access to outer Internet is mediated and regulated by a Web Proxy and no direct Internet access is allowed from internal hosts. Web Proxy can be extended with URL filtering to forbid access to blacklisted or malicious web sites. This might be a counter-measure to block access in case of some forms of DC internal attacks (e.g., botnet, ransomware, etc.).
- **SMTP:** possibility to send emails to Internet recipients via the SMTP protocol is regulated by a centralised SMTP service. SMTP service can be extended with anti-virus and anti-spam capabilities to block such kind of service misuse.
- **Intrusion Detection System (IDS):** certain traffic both internally to the DC and from the Internet might contain malicious payloads, such as malformed network packets, SQL injection attacks, and others. This component can be setup to detect or prevent such attacks.

Also patch management, in particular related to fix security issues, is a planned practice in the DC. We schedule period reviews to assess the status of available fixes and to plan their installation.

2.2 Continuous Integration and Continuous Delivery

2.2.1 CI/CD Strategy & Architecture

DevOps is a set of practices to build, test and release code in small frequent steps. In this process, all the actors involved collaborate with each other in order to continuously release the software in a shorter time. This practice makes it possible to achieve the set objectives thanks to the rapid feedback sent by the customer or user who is able to evaluate the entire application in real time. We will focus on the concept of **SecDevOps** which is characterized by a **"Security by Design"** approach. In fact, it happens more and more often that development and operations activities are accompanied by security ones, with the clear objective of defending not only the infrastructure of the app or program

under construction, but also on offering a high level of protection for the data and personal information of users who will use them. From an “operational” point of view, this safety-centered approach can be different from the standard one, going in some cases, to upset the development logics that would have been adopted instead. From a purely “finalistic” point of view, however, nothing changes. In fact, the goal of the entire team made up of the various security, development and operation teams is to create applications in the shortest possible time, always guaranteeing high quality and security standards. This goal can be achieved through the right use of the so-called Continuous Integration and Continuous Delivery practice.

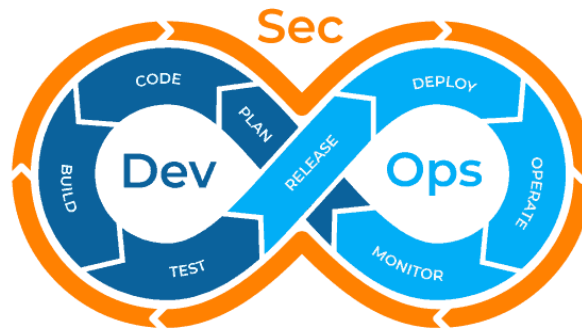


Figure 10 – SecDevOps CI/CD practice

The CI/CD pipeline concept comes from the need for different teams (developers and operational people) to be able to collaborate actively and quickly on the entire development of the project in order to eliminate waiting times due to slow communication between two or more working groups. The term CI/CD is made up of two keywords: **Continuous Integration** and **Continuous Delivery** or Deployment. With the term of continuous integration, we mean the idea of frequently integrating the programming source code distributed among the different teams to spot any problems that do not fulfil the stated acceptance criteria as soon as it is committed into the system. For successful continuous integration, new code changes are regularly compiled, tested, and merged into a shared repository, thus solving the problem of conflicts between the many branches of an application under development.

The principle of Continuous Integration relies in achieving certain prerequisites:

- **Use Version Control:** every part of the project like code, tests, scripts and anything else needed to the application must be checked into a single version control repository;
- **Check-In Regularly:** ensures that the changes between two versions are minimal. This way can avoid errors and bugs that could break the software build;
- **Create a Comprehensive automated test suite:** to achieve this purpose it is necessary to implement a correct testing process to provide confidence that the application is actually working. There are three main types of tests: *unit tests*, *component tests* and *acceptance tests*. Unit tests are written to test the behaviour of small pieces of the application. Component tests are used to verify the behaviour of some application components. Acceptance tests verify that the dictated acceptance criteria have been met. These three sets of tests, combined, should provide an extremely high-level of confidence that any introduced change has not broken existing functionality;
- **Keep the build and test process short:** the process of compiling and testing the entire application should take a few minutes at the most.

Moving on to the term "CD", it takes on two distinct but related meanings. The first is **Continuous Delivery** and identifies the process by which changes made by a developer to the application are automatically tested for bugs and uploaded to a shared repository. This solution was proposed to address the problem of poor visibility and communication between development and operational teams. In this way it is possible to guarantee a continuous distribution by limiting manual interventions as much as possible and to have a base code always ready to be distributed to a production environment.

The other possible meaning for "CD" is **Continuous Deployment**. It refers to the automatic release of changes made (i.e., the **software artefacts**) by the developer from the repository to production, where they become available to end-users and customers. In this way, all those manual processes that would slow down the distribution of applications are eliminated. Thanks to this method, the application can be available to the customer after a few minutes of writing the code by the developer and it will then be possible to receive feedback sent by users quickly and constantly.

Returning to the general concept of the CI/CD pipeline, a deployment pipeline is, in essence, an automated implementation of the application's build, deploy, test, and release process. Each organization may have minimal differences from an implementation point of view due to a different business context, but the fundamental principle behind the pipeline concept is universal for everyone. It is possible to divide the various phases into distinct subsets. The main stages of the pipeline that needs be implemented in this process include:

- **Development:** in this phase the team of developers, in our case GK developer's partners, write the source code, script and anything else needed to the application;
- **Source Control:** in this phase the various versions of the application are tracked. This way when any change occur, you can still access the previous revision;
- **Build:** in this phase the application code is created by compiling the sources;
- **Test:** in this phase the code is tested. Here testing automation can save time and effort and be replicable across different builds.
- **Release:** in this phase the final application artefacts are made available on a repository;
- **Deployment:** at this stage the code is deployed from the repository to the operating environment.

The picture in Figure 11 illustrates the flows between these stages:

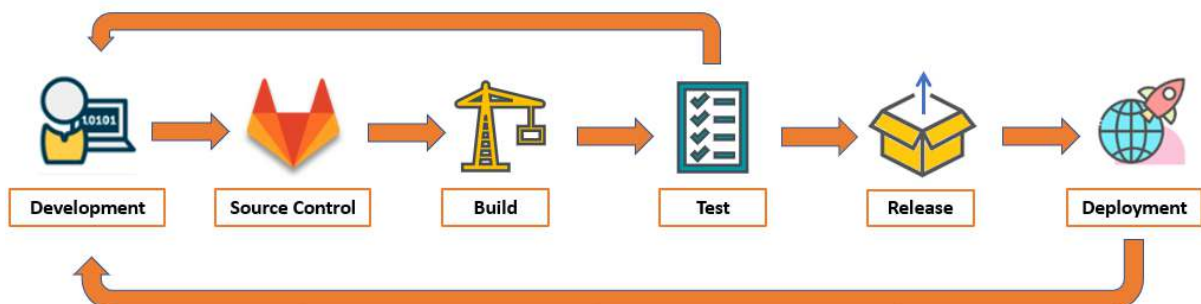


Figure 11 – CI/CD stages

Of course, this process is not bulletproof, because still few issues and defects might reach the deployment stage and be put in production. However, the operational people can use the "channel" setup with developers to open defects and provide feedbacks in a win-win virtuous cycle (the back arrows in Figure 11).

2.2.2 CI/CD Tools

We propose hereafter a selection of tools we will use to build the CI/CD pipeline for the GK project. The choice was based on experience of the consortium partners, tools used in other EU funded projects or internal R&D factories, the fact of being open source, and on public reviews and ratings (e.g., Synopsys Black Duck Open Hub site [17]). Below we summary the major features and capabilities of the selected toolset.

Code & Version Control: GitLab Community Edition (CE)



GitLab CE is an open source software to collaborate on source code development. GitLab offers a git [18] version control repository management, code reviews, issue tracking, activity feeds and wikis. We do plan to have it connected with the GK DC internal LDAP service to enable authentication and authorization of developers and partners. Apart from source code repository, we will use it for tracking software defects and enhancement thanks to its build-in issue tracking mechanism.

It has strong established and mature codebase maintained by a very large development team and few high vulnerabilities reported during the years as shown on OpenHub. It uses an open source MIT License that is commercially friendly.

Link: <https://gitlab.com/gitlab-org/gitlab-foss/>

Artefact Repository: Nexus Repository OSS



Nexus OSS is an open source repository that supports many artefact formats, including Java binary artefacts and Docker images. With the Nexus tool integration, pipelines in the CI/CD can publish and retrieve versioned components and their dependencies by using a central repository.

Nexus OSS has a broad support for many tools:

- Store, manage, and distribute artefacts in Java, Docker, and more, either fetched from external sources or generated from the continuous integration pipeline, such as Java binaries and containers images;
- Support the Java ecosystem, including Maven² and Gradle³ build tools;
- Compatible with popular tools like Eclipse⁴, IntelliJ⁵, Jenkins (see next), and more.

It is release under the open source Eclipse Public License 1.0.

Link: <https://www.sonatype.com/nexus/repository-oss>

² Maven, software project management tool, <https://maven.apache.org>

³ Gradle, software build tool, <https://gradle.org>

⁴ Eclipse IDE, Integrated Development Environment, <https://www.eclipse.org/eclipseide>

⁵ JetBrains IntelliJ IDEA, Integrated Development Environment, <https://www.jetbrains.com/idea/>

Continuous Integration: Jenkins



Jenkins is a continuous integration server, allowing to automatically monitor source repositories, build software, run tests and deploying software. Through the installation of plugins, Jenkins integrates with a huge set of tools in the continuous integration and continuous delivery toolchain, including GitLab CE and Nexus OSS. It has several dashboards for controlling the status of the unit and integration tests (e.g., JUnit compatible) and dashboards for visualising the status of the quality and security tests performed on code artefacts.

Jenkins is made available via an open source MIT License.

Link: <https://www.jenkins.io>

On top of the tools just mentioned, that constitutes the backbone of the CI/CD, we have selected few others that helps the GK platform to be continuously assessed for obtaining a high degree of assurance and possibly be free of security vulnerabilities. In particular we selected a couple of open source tools that perform static application security testing, though static code analysis techniques. Others will be evaluated during the next period of activities.

OWASP Dependency-Check



OWASP Dependency-Check is a Software Composition Analysis (SCA) tool that attempts to detect publicly disclosed vulnerabilities contained within the dependencies of a developed software artefact. The tool extracts the Common Platform Enumeration (CPE) identifiers of the third-parties software libraries that the developed component depends on. It then uses such information to match against the Common Vulnerabilities and Exposures (CVE) public database to retrieve, when available, its security vulnerabilities. Dependency-Check integrates natively with Jenkins via a dedicated plug-in. It was conceived in 2012 based on the *OWASP risk #9 Using Components with Known Vulnerabilities*, part of the OWASP Top Ten project⁶, that was identified as a major issue of concern in that period. It is released with the Apache Software License 2.0 as open source code.

Link: <https://owasp.org/www-project-dependency-check/>

SpotBugs



SpotBugs uses static source code analysis to look for bugs in Java code. SpotBugs checks for more than 400 bug patterns and can be used as a standalone tool or via several integrations, including the continuous integration builds with Jenkins. The bug patterns database of SpotBugs can be extended with a plugins mechanism and the open source community has released a security-oriented plugin called **Find Security Bugs**. Find Security Bugs detects over 138 different types of security vulnerabilities, including the risks from OWASP Top Ten and from the Common Weakness Enumeration (CWE) initiative. Both SpotBugs and

⁶ Open Web Application Security Project Top Ten Project, <https://owasp.org/www-project-top-ten/>

Find Security Bugs are released under the terms of the GNU Lesser General Public License.

Link: <https://spotbugs.github.io/> - <https://find-sec-bugs.github.io/>

2.3 Containerization Techniques

The GK platform is planned to be deployed by using a virtualization technique called containerization. The containers will run on top of virtual machines (VMs) running in the oVirt virtualization platform, as reported in Section 2.1.1.3. This combination of VMs and containers allows a flexible use of DC compute resources, operating at the different abstractions levels of virtualization. As you will learn in the following sections, containers work perfectly well with the CI/CD pipeline and allow easy packaging of applications, helping Developers and Operational people (DevOps) working together more efficiently. The next sections illustrate such techniques, the difference between VMs and containers, and the available tools.

2.3.1 Introduction to Containers: Hypervisor vs Container

The **Hypervisor** and the **Container** are *virtualization mechanisms* that each perform an abstraction at different levels in the computer hardware / software stack of an operating environment:

- The **Hypervisor** positions itself above the hardware and performs an abstraction of the hardware components (CPU, RAM, storage and network) of a **Computer Server** in order to make them available as virtual resources to the “virtual computer machines” defined on it (called **Guest OS**). In particular, modern hypervisors (KVM [19], ESXi [20], Xen [21]) perform abstraction by sharing hardware resources (CPU and RAM) between virtual machines. On top of Guest OS, it runs the **Applications** with their required software **Binaries** and **Libraries**.
- The **Container** is located above the operating system (called **Host OS**) and performs an abstraction on the operating system itself: in particular, the OS kernel is shared among a series of isolated instances capable of encapsulating both **Applications** and its **Binaries** and **Libraries** within them. Each instance sees resources (filesystem, memory, processes and devices) as if they were entirely dedicated to it.

Figure 12 graphically depicts these concepts:

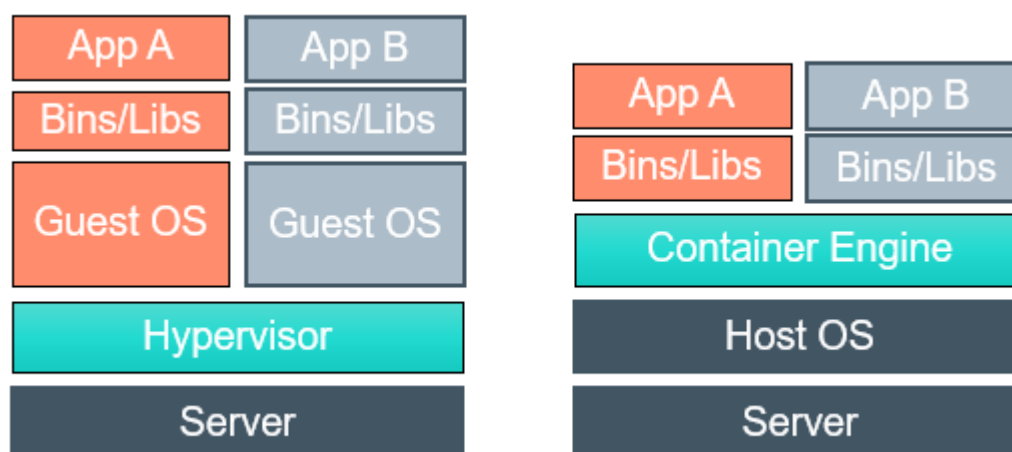


Figure 12 – Hypervisor vs Container

Although hypervisor and container are both virtualization technologies, the objectives achieved by these technologies are different from each other:

- Virtualization technology via hypervisor aims to totally emulate a new environment: each virtual machine (Guest OS) can run a different operating system;
- Container technology aims to make applications portable and self-contained. The virtualized environment is the same for all instances and is the same as the operating system on which the containers “run”.

2.3.1.1 Brief History of Containers

For decades Unix systems have used the “chroot” [22] command to provide a form of filesystem isolation. Since 1998, FreeBSD has been using the “jail” utility [22] which creates a “sandbox” for processes. A first concrete example of container is offered in 2001 by the Solaris operating system feature called “Zones” [23].

In 2001, Parallels Inc. (formerly SWsoft) released the Virtuozzo [24] container technology which it will then make Open Source in 2005 under the name OpenVZ [25].

Google begins to develop “Cgroups” in 2006 [26] for the Linux kernel and begins to move all its infrastructure to containers.

In 2008 the Linux Containers (LXC) [27] project was born which combines together Cgroups, kernel namespaces and the chroot.

In 2013, Docker [9] packages all the above tools to provide a complete solution for creating and deploying containers.

2.3.1.2 Container: the metaphor of freight transport

The freight transport industry has undergone a revolution with the introduction of intermodal containers: these containers have standard dimensions all over the world and all the machinery for handling and transporting containers are designed and built according to these containers. The benefit of this standardization is represented by the fact that the transport industry abstracts from the type of goods to be transported and can focus on the handling and storage of containers regardless of the content.

By extending the benefits of intermodal containers to IT, applications become the transported goods and software developers focus on application development activities without worrying about differences in execution environments and dependencies.

2.3.2 Characteristics of Containers

This section describes the most important characteristics and advantages of the containerization techniques.

2.3.2.1 Isolation and management of resources

In order to ensure the isolation of OS resources (filesystems, memory, processes and devices) between the host system and the containers running on it, the OS kernel implements a mechanism known as a **namespace**. Namespaces ensure that each OS process has its own vision of the system, different from that of the other OS processes. The resources are therefore effectively shared, but each container sees resources as if they were entirely dedicated to it. For instance, namespaces are the feature we use to segregate different operating environments (development/testing/production) using the same resources, as well as segregate the GK pilots that require such kind of data separation.

In order to manage access to OS resources, containers use another OS kernel mechanism known as **cgroups** (short for *control groups*). Control groups allows both to restrict access to CPU, memory and I/O for some containers, and to make sure that they can access them with specific assigned priority with respect to others.

2.3.2.2 Advantages of file system isolation

Containers allow to obtain a complete isolation between the filesystem of the application running on the container and the Host OS filesystem that runs the container itself.

In this model there is a filesystem linked to the context of the container and a filesystem linked to the context of the Host OS: a containerized application residing in the `"/usr/bin/"` filesystem path that refers to libraries in `"/lib"` can modify the libraries in `"/lib"` without affecting the correct functioning of the Host OS.

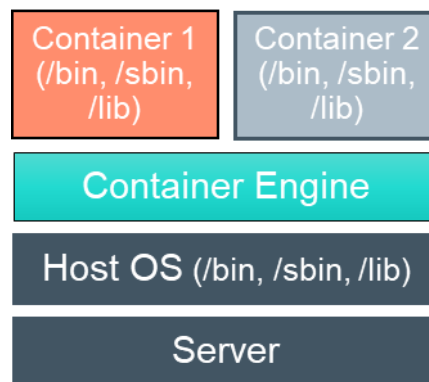


Figure 13 – Container filesystem isolation

2.3.2.3 Other Advantages of using Containers

There are several other advantages offered by Containers. We report here the most relevant.

- **Simplified development:** a container effectively *packages the application* into a single component which can be deployed and configured with a single command, so developers do not need to worry about any configuration of the execution environment;
- **Increased availability:** a developer can host myriads of containers without heavily impacting the hardware resources of the computer itself since containers turn out a very light-weight form of virtualization;
- **Faster start-up times:** the container can be started in a time that is much shorter than the one needed to start a virtual machine because you do not need to start a Guest OS first;
- **Portability and Consistency:** this is probably the most important advantage this technology is able to offer thanks to its format that allows execution on different hosts without changes.

2.3.2.4 Container Images

A container is created by instantiating an **image**. We can think of an image as a static model used to create containers. Images usually start with a base filesystem and add changes to the filesystem and related execution parameters in sorted read-only layers. Unlike a typical Operating System installation, an image usually contains only what is

strictly necessary to run the application. The images are stateless and immutable, and they are the starting point for containers.

When images are instantiated in a container, a **read-write layer** is added to the top of the image. This combination of *read-only layers followed by a read-write layer* is known as a **Union File System**⁷. When a change is made to an existing file in a started container, the file is copied from the read-only space within the read-write layer in which the changes are applied. The version in the read-write layer hides the original file, but does not remove it. Changes in the read-write layer exist only within a single container instance. When a container is deleted all changes are lost unless measures are taken to safeguard them.

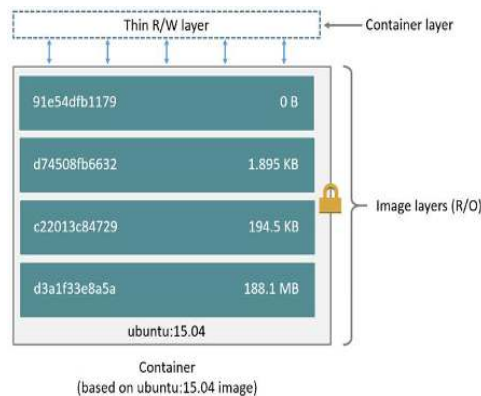


Figure 14 – Image Layers of a Container

The images used to run the containers can be created (by the administrator or the developer) or downloaded from a **Registry**, which can be public or private (see 2.3.5).

2.3.3 Containers and DevOps

DevOps is the contraction of the terms “Development” and “Operation”, as already mentioned in Section 2.2. The goal is to bring these two key functions closer together to accelerate the innovation cycles of applications by improving their quality.

Container technology allows you to package applications and isolate them together with the entire runtime environment - that is, with all the files needed to run. In this way, it is possible to quickly transfer the applications contained in the packages between different environments (development, test, production), while maintaining full functionality.

Containers make it easier for development and operations teams to collaborate with the ability to define responsibilities: developers can focus on application development, while operations teams focus on infrastructure.

2.3.4 Containers and Micro-services

Micro-services represent an architectural computer programming style by which an application is built no longer as a monolithic object but as a collection of small autonomous and independent services, which can be implemented separately and which

⁷ From the **union mounting**, “a way of combining multiple directories into one that appears to contain their combined contents”. Ref. https://en.wikipedia.org/wiki/Union_mount

communicate with other micro-services through an exchange of messages via the computer network.



Figure 15 – Monolithic vs micro-services (ms) based application

The main advantages of micro-services are:

- **Reusability:** each micro-service can be reused by other applications;
- **Scalability:** if necessary, only the distressed micro-service is scaled and not the entire application;
- **Deployment:** an application can be deployed both on the same machine and on multiple machines depending on the performance requirements.

The micro-services architecture blends easily with containerization techniques, where each service can be easily packaged as a container.

2.3.5 Docker

The Docker [9] project starts from the need to implement container management platform through an open source engine. Docker is a set of technologies that cooperate in order to standardize the container format and have a clear and orderly toolchain, therefore efficient. It was originally conceived for the Linux OS, but later evolved to support others as well (e.g., Microsoft Windows and Apple macOS).

The elements that make up the Docker ecosystem are as follows:

Docker engine, the daemon that resides on the Host OS and accepts commands from a client, be it a command line utility or an API call. This daemon communicates with the Linux OS Kernel through the *libcontainer* library, also part of the project.

Registry, a platform available over the network and that provides an area to upload, download and share images of the various containers. Docker's public registry is called "Docker Hub"⁸ and it is cloud-based.

Docker Swarm, a Docker's native clustering solution: transforms a pool of Docker hosts into a single Docker host that acts as a high-available service.

Docker Compose, a tool for defining and running a multi-container Docker application. It uses a specification document to configure the application services; therefore, with a

⁸ Docker Hub, <https://hub.docker.com/>

single command it is possible to create and start all the services defined on such configuration file.

2.3.5.1 Data Persistence and Docker Volume

The container's read-write image layer is an ephemeral storage space and when the container is destroyed, that storage space disappears.

To run *stateful applications* in containers (e.g., a database that needs to store information), we need a persistent storage space - that is, one that survives the disappearance of the container. Persistent storage in a container can be implemented through the **Docker Volume** mechanism. This is a portion of the storage space, outside the container filesystem, which by default is on the host running the Docker engine. Usually, it is a directory of the host filesystem, managed entirely by Docker, which can be mounted by more than one container and whose contents can be modified by both the container and the host.

2.3.5.2 Security of the container images

In its default configuration, Docker uses the Docker Hub, but other public registries can be setup where various communities publish their images with their personal customizations. From a security point of view, this practice can put systems at risk as the images uploaded to the public registry are not subjected to accurate checks on their actual content, can contain security vulnerabilities or even malware software. This is why it is important to perform a screening of the available images, as well as a continuous assessment of their security posture to spot vulnerabilities that were not initially present in the image but happens to be discovered.

For example, the Red Hat Docker distribution (OpenShift [5]) adopts the best practice to configure the vendor's registry as the default registry (registry.access.redhat.com). All Docker images in this registry are tested and certified by Red Hat itself.

2.3.6 Container orchestration

Enterprise applications take advantage of the containerization concept, but at the same time requires ways to manage the whole containers lifecycle, including the possibility to automatically scale containerized applications up and down, monitoring containers, replace failed containers, rollout software updates or new configuration. Tools to manage, scale, and maintain containerized applications are called **container orchestrators**.

We refer here to some of the most popular orchestration tools:

- Docker Swarm;
- Kubernetes.

Although each of them can perform the same task, the ways in which they carry out the tasks differ from each other. In fact, each solution offers a specific level of abstraction, operational approach (how the orchestration is managed and how it integrates with other services) and is recommended for specific use cases (containers, services, various small or large-scale workloads, etc.). It must also be assessed whether the framework can easily integrate / flank our working approach or whether we must necessarily adopt a "new operating philosophy".

2.3.6.1 Docker Swarm

Swarm [28] is the proposal developed by Docker that easily adapts to environments where you already work with Docker containers. The solution is ideal for testing operations or for

the deployment of small-scale applications. "Swarm" is characterized by a series of components responsible for carrying out specific tasks, in detail:

Manager. They take care of distributing the tasks among the various clusters. A manager is in charge of orchestrating the *worker nodes* that make up the Swarm cluster.

Worker. They run containers that have been assigned by a manager.

Services. An interface for a particular set of running Docker containers.

Task. Individual containers running images and commands required by a particular service.

Key-value store. Services (e.g., etcd [29]) that take care of archiving the status of the Swarm and providing visibility to the service.

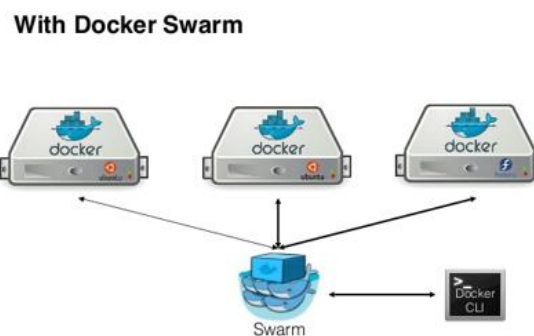


Figure 16 – Docker Swarm

2.3.6.2 Kubernetes

It is the open source project developed by Google [7]. Kubernetes is recommended for medium to large clusters and complex applications. Compared to Swarm, the learning curve is much steeper - but those who learn how it works will be rewarded by the flexibility / modularity of the solution and the ability to manage very large-scale deployments.

A Kubernetes cluster consists of:

Master. It deals with managing API calls, assigning workloads and supervising the configuration of states.

Worker. Servers running workloads or other items not located in the Master.

Pod. Computational units consisting of one or more *containers* which have been deployed on the same host. They take care of performing tasks and have a single IP address.

Services. Front-end and pod load balancer, provide a floating IP address to access the pods running the service.

Replication controller. They are in charge of supervising a certain number of copies of the required pods.

Label. Tags used to identify pods, replication controllers, and services.

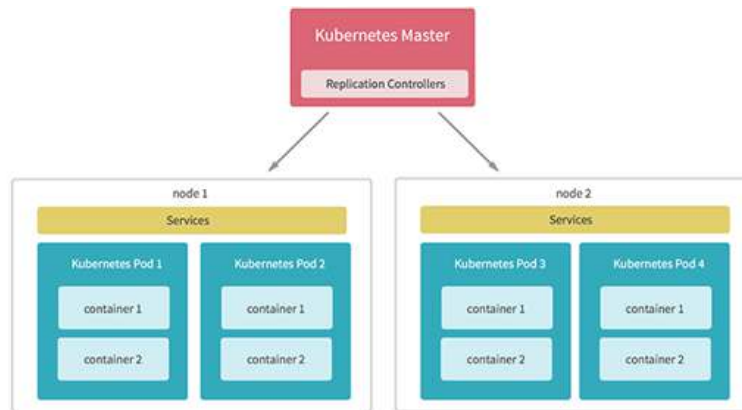


Figure 17 – Kubernetes

2.3.6.3 OpenShift

OpenShift [5] is an excellent container orchestrator. It is based on Kubernetes, which we already mentioned as an open source product created by Google from a long experience of using and managing containers in high-performance production environments.

The Red Hat OpenShift platform compared to similar container orchestrators, has a fundamental advantage: it allows you to define a complete CI/CD workflow, which starts from the source code and images descriptor code to arrive at the deployment of containerized applications.

OpenShift is built from Kubernetes, but uses several other open source projects and technologies such as Jenkins [30], a very popular CI/CD server, and "Source 2 Image" [31], a technology that allows you to pass from the source code of an application to the containerized application.

OpenShift also provides various user interfaces (web-based or command line, to allow better automation), APIs that give the possibility to manage every aspect of the platform and a series of runtimes to facilitate the creation of containerized applications. OpenShift, has available a built-in catalogue of languages and servers to build complex distributed systems and micro-service applications in a simple way.

It also has many tools to facilitate collaboration between users by minimizing unwanted interference. The platform is *multi-tenancy*⁹ and equipped with an articulated system for managing users, roles and permissions. Access to resources can be controlled at an extremely fine level of granularity, and the use of any resource can be restricted.

OpenShift is the ideal platform to optimize collaboration between developers and systems engineers. OpenShift allows developers to use independently the tools needed to build sophisticated applications and build and deploy workflows; systems engineers have full control of the platform, resources and application execution environment. Finally, the system engineers can immediately access any security fixes published by Red Hat and instantly apply them to containers running on the platform.

⁹ Multi-tenancy is a single application instance that can be used by different users (the *tenants*), where each tenant have dedicated resources and segregated data, effectively isolating each of them from the others.

2.3.6.4 OKD

OKD [4] is a Kubernetes distribution optimized for continuous application development and multi-tenant deployment. It is the open source version of OpenShift, which is in turn commercially distributed by Red Hat. OKD adds developer-centric and operations-centric tools in addition to Kubernetes to enable rapid application development, easy deployment and scalability, and long-term lifecycle maintenance for small and large teams. OKD is a sister Kubernetes distribution of Red Hat OpenShift.

OKD incorporates Kubernetes and extends it with security and other integrated concepts. OKD is also referred to as Origin on GitHub¹⁰ and in the documentation.

Among the orchestration methods described above, the GK project selected to use OKD as container orchestrator.

OKD builds upon the concept of namespaces and cgroups (described in Section 2.3.2.1) and can bundle together several computational units (Pods) to guarantee the confinement of different workload, e.g., segregating the different GK Pilots resources and data such in a way that they cannot communicate or overlap. The picture in Figure 18 graphically depicts this concept:

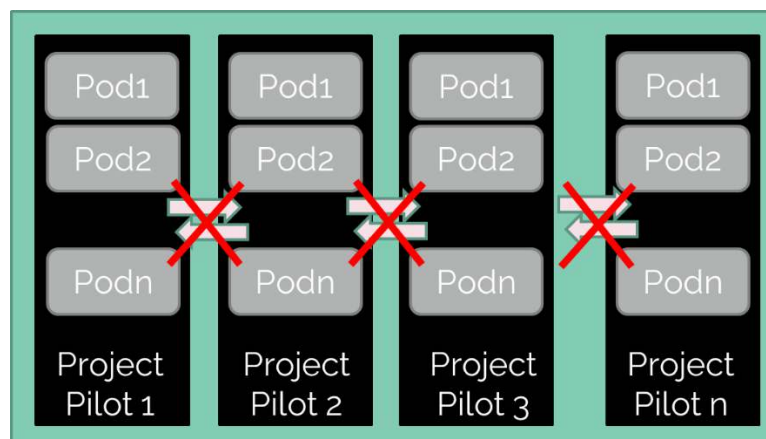


Figure 18 – Segregation for GATEKEEPER Pilots

2.3.6.5 OKD vs OpenShift

There is basically no difference between OpenShift Enterprise 4 and OKD 4 as with most Red Hat products, the upstream version is essentially a free version with no support or specific Service Level Agreements (SLAs).

OKD version is free to use and includes most of the features of its commercial product. For this reason we have chosen to orchestrate the containers on the OKD platform. This allows a GK partner to switch to the commercial and paid-supported Red Hat OpenShift variant without losing the investment made during the GK platform development.

¹⁰ OKD public source code repository, <https://github.com/openshift/okd>

3 GATEKEEPER User Manuals

3.1 GATEKEEPER Data Centre Infrastructure

3.1.1 Data Centre Access User Manual

As part of this deliverable, we release as an Annex the **confidential document** that describes in detail the operating procedure GK partners' users must follow to connect and use the HPE GK Data Centre. We report in Figure 19 the index of the document.

Table of contents

TABLE OF CONTENTS.....	4
LIST OF TABLES	5
LIST OF FIGURES	6
INTRODUCTION	7
1 HOW TO CONNECT TO HPE GK DATA CENTRE	8
1.1 QUICK START GUIDE - FOR EXPERIENCED USERS.....	8
1.2 DETAILED PROCEDURE	8
1.2.1 Activate your OTP token	8
1.2.2 Install OpenVPN.....	15
1.2.3 Access the HPE Portal	20
2 TRUST HPE GK DATA CENTRE WEB RESOURCES.....	25
2.1 USING CHROME BROWSER ON WINDOWS	26
CONCLUSIONS.....	31
REFERENCES	32
APPENDIX A HPE GK CERTIFICATION AUTHORITY.....	33

Figure 19 – GATEKEEPER-WP4-GK_Data_Centre_Access document ToC

The document guides the GATEKEEPER partner users through the steps of getting the necessary software to setup the VPN connection to the Data Centre, the activation of the multi-factor authentication mechanism (2FA), the access to the entry point where the GK services are available. Part of this information has already been discussed in Section 2.1.2 and 2.1.3.

This document is marked as confidential because it contains Data Centre details that must not be made public.

3.2 Continuous Integration and Continuous Delivery

Since the implementation activities related to the CI/CD task have not yet been delivered, this user manual will come at a later stage and documented in the next version of this deliverable.

3.3 Containerization Techniques

3.3.1 Container Workload Orchestration - OKD User Manual

To use this platform for GK application development and hosting needs there are three common ways we can use OKD as a user:

- Command line utility;
- Web console;
- IDE integration.

In the attached manual, we describe how to use the web console with the basics of command line use.

We report in Figure 20 the agenda of the webinar HPE delivered to the GK partners:

WP4 T4.1: CONTAINER WORKLOAD ORCHESTRATION OKD FOR GATEKEEPER PLATFORM – WEBINAR AGENDA



Introduction to OKD	Deployment creation
Platform access prerequisites	Creating a PVC
Pre-requisites for users of the OKD platform	Creating a POD
Access to the platform	Service creation
Description of the web console	Routes creation
Description of the "Administration Cluster" interface	Config Maps creation
Access to the "Developer" interface	Secret creation
Description of the "Developer" interface	Using the CLI
Access to a specific project	
Example of creating a MySQL database	



2

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 857223



Figure 20 – OKD Webinar for GK partners

4 Conclusions

This deliverable includes the description of the effort spent in the setup of the GATEKEEPER Data Centre hosted by HPE, from its hardware equipment, to the access and software services provided to the GATEKEEPER consortium. A specific accent has been posed to the security mechanisms that have been put in place or are planning to be setup in the next period of activities. Some services are still under construction, like the CI/CD platform, other have been deployed and are ready to be used by the partners for hosting the GATEKEEPER platform.

Next steps include the tune of the currently delivered services, once they become generally available after the release of this deliverable. Also, the T4.1 partners will review the feedback coming from the GK consortium to enhance the user experience and the security assurance level.

As a companion of this document, deliverable D4.3, expected at M24, will complement the released DC infrastructure with the Big Data tools to serve the needs of Data Scientists and, in particular of WP3, WP5, and WP6.

5 References

- [1] Flex Fabric Virtual Technology Configuration, https://support.hpe.com/hpesc/public/docDisplay?docId=a00050312en_us
- [2] HPE Synergy - composable bladed infrastructure, https://www.hpe.com/emea_europe/en/integrated-systems/synergy.html
- [3] oVirt - free open-source virtualization solution for enterprise, <https://www.ovirt.org>
- [4] OKD - community distribution of Kubernetes, <https://www.okd.io>
- [5] Red Hat OpenShift - hybrid cloud platform, <https://www.openshift.com>
- [6] CentOS, Community Enterprise Operating System, <https://www.centos.org>
- [7] Kubernetes - production-grade container orchestration, <https://kubernetes.io>
- [8] HPE Ezmeral, Run, manage, control and secure the apps, data and IT that run your business—from edge to cloud, <https://www.hpe.com/ezmeral>
- [9] Docker, Open Source containers, <https://www.docker.com/community/open-source>
- [10] HPE Nimble, storage solution, <https://www.hpe.com/us/en/storage/nimble.html>
- [11] HPE StoreOnce, backup solution, <https://www.hpe.com/us/en/storage/storeonce.html>
- [12] HPE Networking Switches, network solution, https://www.hpe.com/emea_europe/en/networking/switches.html
- [13] IEEE 802.1Q, Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks, https://standards.ieee.org/standard/802_1Q-2018.html
- [14] IEEE 802.3, Standard for Information Technology - Local and Metropolitan Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications-Aggregation of Multiple Link Segments, https://standards.ieee.org/standard/802_3ad-2000.html
- [15] HPE Proliant DL Servers, rack-optimized secure industry-standard servers, <https://www.hpe.com/us/en/servers/proliant-dl-servers.html>
- [16] Bareos, Open Source Data Protection, <https://www.bareos.org>
- [17] Synopsis Black Duck Open Hub, <https://www.openhub.net>
- [18] Git, distributed version control, <https://git-scm.com>
- [19] Linux KVM, Kernel-based Virtual Machine, <https://www.linux-kvm.org>
- [20] VMware ESXi, Bare Metal Hypervisor, <https://www.vmware.com>
- [21] Xen, Type-1 Virtual Machine, <https://xenproject.org>
- [22] FreeBSD Chroot Jails, https://docs.freebsd.org/en_US.ISO8859-1/books/handbook/jails.html
- [23] Solaris Zones, Oracle Solaris Zones Introduction, https://docs.oracle.com/cd/E36784_01/html/E36848/zones.intro-1.html
- [24] Virtuozzo Containers, product retired, <https://www.virtuozzo.com/support/all-products/virtuozzo-containers.html>
- [25] OpenVZ, Open source container-based virtualization for Linux, <https://openvz.org/>
- [26] Cgroups (previously process containers), <https://lwn.net/Articles/236038/>
- [27] Linux Containers (LXC) project, Infrastructure for container projects, <https://linuxcontainers.org>
- [28] Docker Swarm, Swarm mode key concepts, <https://docs.docker.com/engine/swarm/key-concepts>
- [29] Etcd, a distributed, reliable key-value store for the most critical data of a distributed system, <https://etcd.io/>
- [30] Jenkins, free and open source automation server, <https://www.jenkins.io/>
- [31] Source-To-Image (S2I), toolkit and workflow for building reproducible container images from source code, <https://github.com/openshift/source-to-image>

Appendix A Annexes

The following documents are released as confidential documents restricted only for members of the consortium (including the Commission Services):

- GATEKEEPER-WP4-GK_Data_Centre_Access_HPE.docx
- GATEKEEPER-WP4-GK_OKD_Usage_Webinar_HPE.pptx

Appendix B User Registration Form Text

For the sake of readability and accessibility, we report the full text of the User Registration Form shown in Figure 4 – Preface of the User Registration Form.

GATEKEEPER

HPE Data Centre

Remote Access Request

Please complete the following form. Your data will be collected by HPE as GATEKEEPER partner for the sole purposes of account management and user access provisioning to HPE Data Centre infrastructure hosting the GATEKEEPER project services. You must give your consent to HPE for processing your data according to the HPE Privacy policy available using the link located at the bottom of the form. The HPE responsibilities are described in the link.

This computer network is provided for authorised use only. Users have no explicit or implicit expectation of privacy. Any or all uses of systems of this computer network and all data in this Data Centre may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to authorised sites and law enforcement personnel, as well as authorised officials of other agencies. By accessing to this computer network, the user consent to such disclosure at the discretion of authorised site personnel. Unauthorised or improper use of this network may result in administrative disciplinary action, civil and criminal penalties. By accessing to this computer network, you indicate your awareness of and consent to these terms and conditions of use. Users are responsible for ensuring that they act in accordance with this policy and other policies and legislation applicable to the HPE network.

Please note that by submitting this form you also agree to have your activities on HPE Data Centre monitored for compliance and security purposes. You must not disable (or attempt to) or change any security control, as well as respect the assigned privileges and not escape the perimeter of your role.

Should you have any inquiry please contact HPE project partner for further information.

Technical details about how to access HPE Data Centre services are published on Alfresco project document management system. You will receive an email as soon as your access will be granted with more information.

Notice: it is strictly forbidden to share this registration link outside involved GATEKEEPER project partners.

HPE GATEKEEPER Team

<https://www.gatekeeper-project.eu/>

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 857223

Appendix C User Registration Form Fields

For the sake of readability and accessibility, we report the fields of the User Registration Form shown in Figure 5 – Fields of the User Registration Form.

* Required

1.First Name *

Please insert your name and optional middle name

2.Surname *

Please insert your surname

3.E-Mail Address *

Please specify your e-mail address (e.g., user@example.org). This e-mail address will be used only for communications about HPE data centre. Only valid GATEKEEPER partner e-mail address will be accepted.

4.Organisation Name *

*Please specify your organisation: **here we show a drop down list of consortium partners' acronyms***

5.Your role in the environment *

Notice: your role selection will undergo validation check. Roles explanation follow.

Project User:

- GK-User: access to gatekeeper platform, pilot and other project services

Note: this is not the end user (e.g., patient). GK-User is a regular project partner member

Developer Roles:

- GK-Developer: access to development environment and tools (OKD)

- GK-Data-Scientist: access to big data environment and tools (HPE Ezmeral)

Developers access common services (Gitlab, Jenkins, etc.)

- ☐ GK-User
- ☐ GK-Developer
- ☐ GK-Data-Scientist

6.I have read and I accept the HPE Privacy policy for personal data collection available at the URL:
https://www.hpe.com/emea_europe/en/legal/privacy.html *

☐ Yes

You can print a copy of your answer after you submit

[Submit]

Appendix D Internal HPE Portal home page

For the sake of readability and accessibility, we report the text of the Internal HPE Portal home page shown in Figure 6 – Internal HPE Portal.

HPE SECLAB	
This page provides helpful links for HPE SECLAB made for GATEKEEPER.	
Identity Management Go here to manage your user account. Access Password Change: <i>[internal link]</i> You can change your password or reset it if it expires. SECLAB Internal Certification Authority Import the HPE SECLAB into your browser or other services. Download the SECLAB Internal CA to trust the provided services: <i>[internal link]</i>	For Developers Tools provided for authorised developers only. Access OKD Console: <i>[internal link]</i> OKD is a distribution of Kubernetes optimized for continuous application development and multi-tenant deployment. Access HPE Ezmeral for Big Data Console: <i>soon available</i> Software platform designed to run both cloud-native and non-cloud native applications in containers. More info.
Important notice This computer network is provided for authorised use only. Users have no explicit or implicit expectation of privacy. Any or all uses of systems of this computer network and all data in this Data Centre may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to authorised sites and law enforcement personnel, as well as authorised officials of other agencies. By accessing to this computer network, the user consent to such disclosure at the discretion of authorised site personnel. Unauthorised or improper use of this network may result in administrative disciplinary action, civil and criminal penalties. By accessing to this computer network, you indicate your awareness of and consent to these terms and conditions of use. Users are	About this Project This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857223 [GATEKEEPER] [Hewlett Packard Enterprise - HPE] (c) Copyright 2021 Hewlett Packard Enterprise Development Company, L.P. Valid agreement required.

responsible for ensuring that they act in accordance with this policy and other policies and legislation applicable to the HPE network.	
---	--