1. Technological architecture

Technological architecture

Technology Viewpoint INNOSOC Common

Technology Viewpoint Asset Inventory

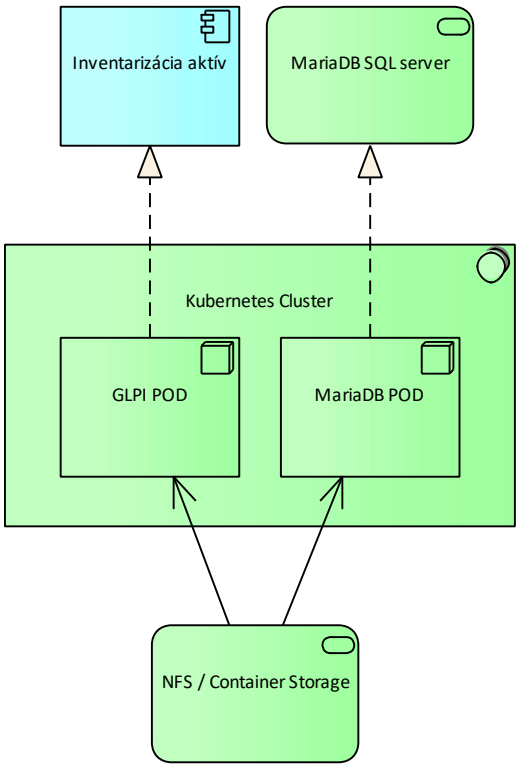


Figure 1 : Technology Usage Viewpoint Asset Inventory alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 1 represents the use of the database directly within the module setup. The design envisages a containerized version of the SQL server.

Description of the elements used:

Asset inventory

Asset inventory is a fundamental pillar of every security solution. It enables the SOC team to obtain an overview of all devices, system entities, applications and virtual or cloud resources that make up the organization's infrastructure. In the SOCaaS environment, it plays an even more significant role, as it enables centralized management, categorization, monitoring and tracking of the

security status of the infrastructure across different customers.

NFS / Container Storage
Represents the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

MariaDB SQL server
Represents the technology service of the relational MariaDB SLQ database.

GLPI POD
Represents the GLPI POD for asset inventory.

MariaDB POD
Represents the MariaDB POD for the relational SQL DB.

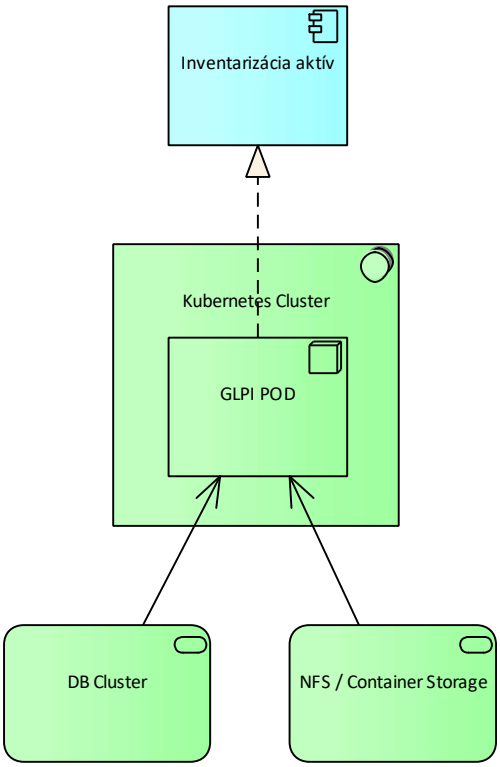


Figure 2 : Technology Usage Viewpoint Asset Inventory alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 2 represents the use of the database directly in the form of a centralized DB cluster.

Description of the elements used:

Asset inventory

Asset inventory is a fundamental pillar of every security solution. It enables the SOC team to obtain an overview of all devices, system entities, applications and virtual or cloud resources that make up the organization's infrastructure. In the SOCaaS environment, it plays an even more significant role, as it enables centralized management, categorization, monitoring and tracking of the security status of the infrastructure across different customers.

DB Cluster
Represents the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

GLPI POD
Represents the GLPI POD for asset inventory.

Technology Viewpoint Kubernetes

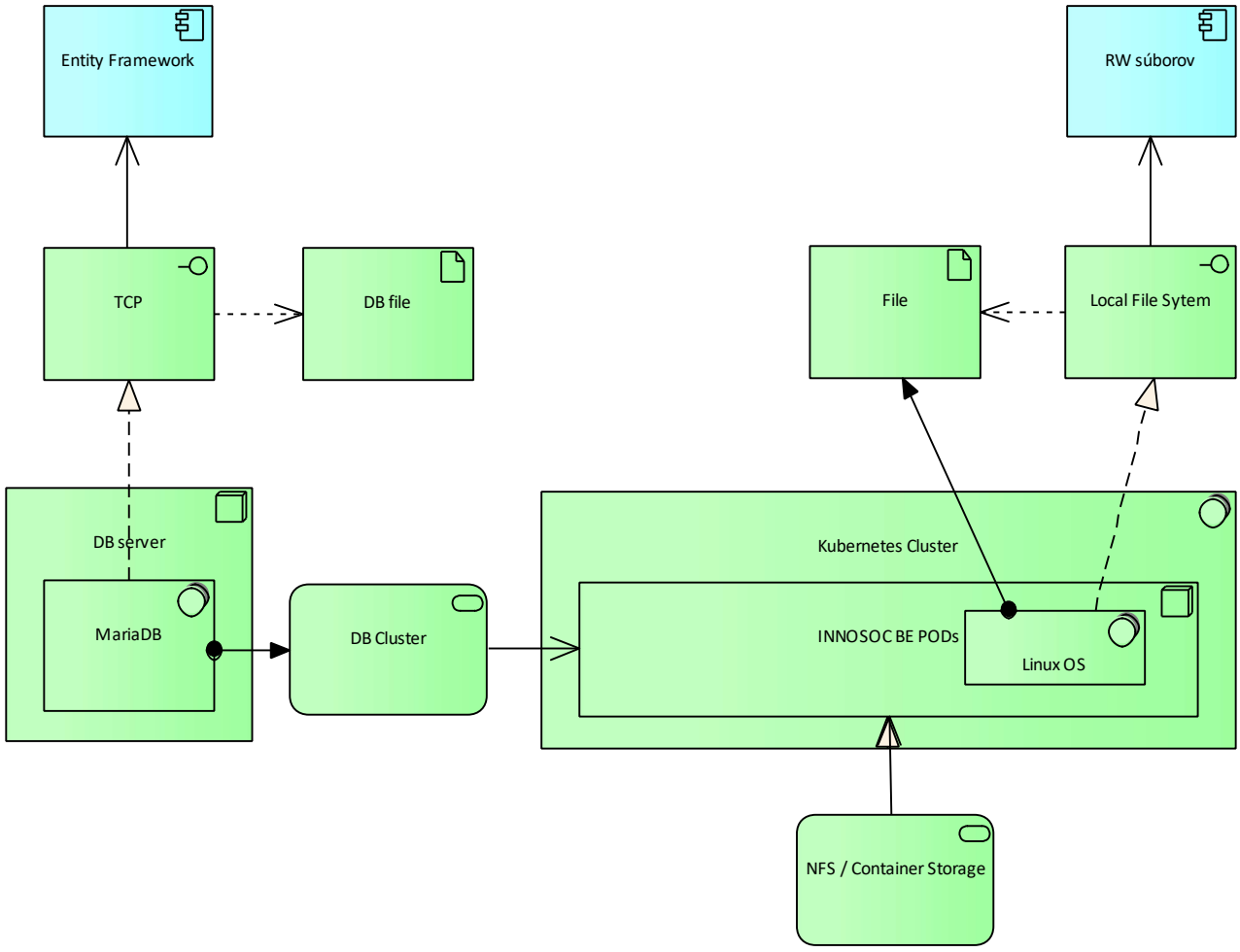


Figure 3 : Technology Usage Viewpoint CNM BE, DB and File System

The diagram represents and describes in more detail the use of the persistence layer for the needs of the BIT NC and CNM BE module by means of its Data layer component, which handles work with PostgreSQL and the Local File System.

Description of the elements used:

DB Cluster
Represents the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

DB file
DB file - a file representing the MariaDB database file

DB server
Represents the MariaDB server

File
File - a file representing data such as attachments, images, documents and others

Entity Framework
Represents the Entity Framework application module

File RW
Represents the RW module for files

Local File Sytem
Local file system providing work with files, implemented by the operating system under which the BE module is run

TCP
Represents client-server communication using the TCP protocol

Kubernetes Cluster
Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

Linux OS

MariaDB
Represents the MariaDB relational database

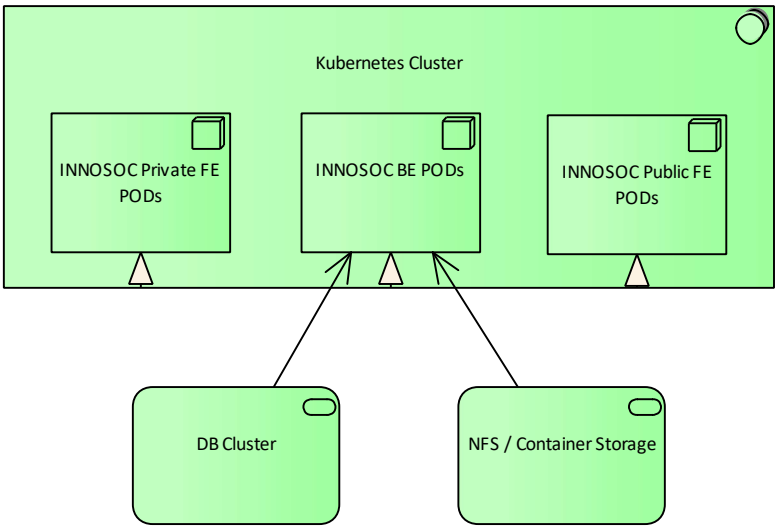


Figure 4 : Technology Usage Viewpoint CNM and Containers maping

The diagram shows the assignment of the individual PODs to the corresponding application components.

Description of the elements used:

DB Cluster
Represents the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

Kubernetes Cluster
Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

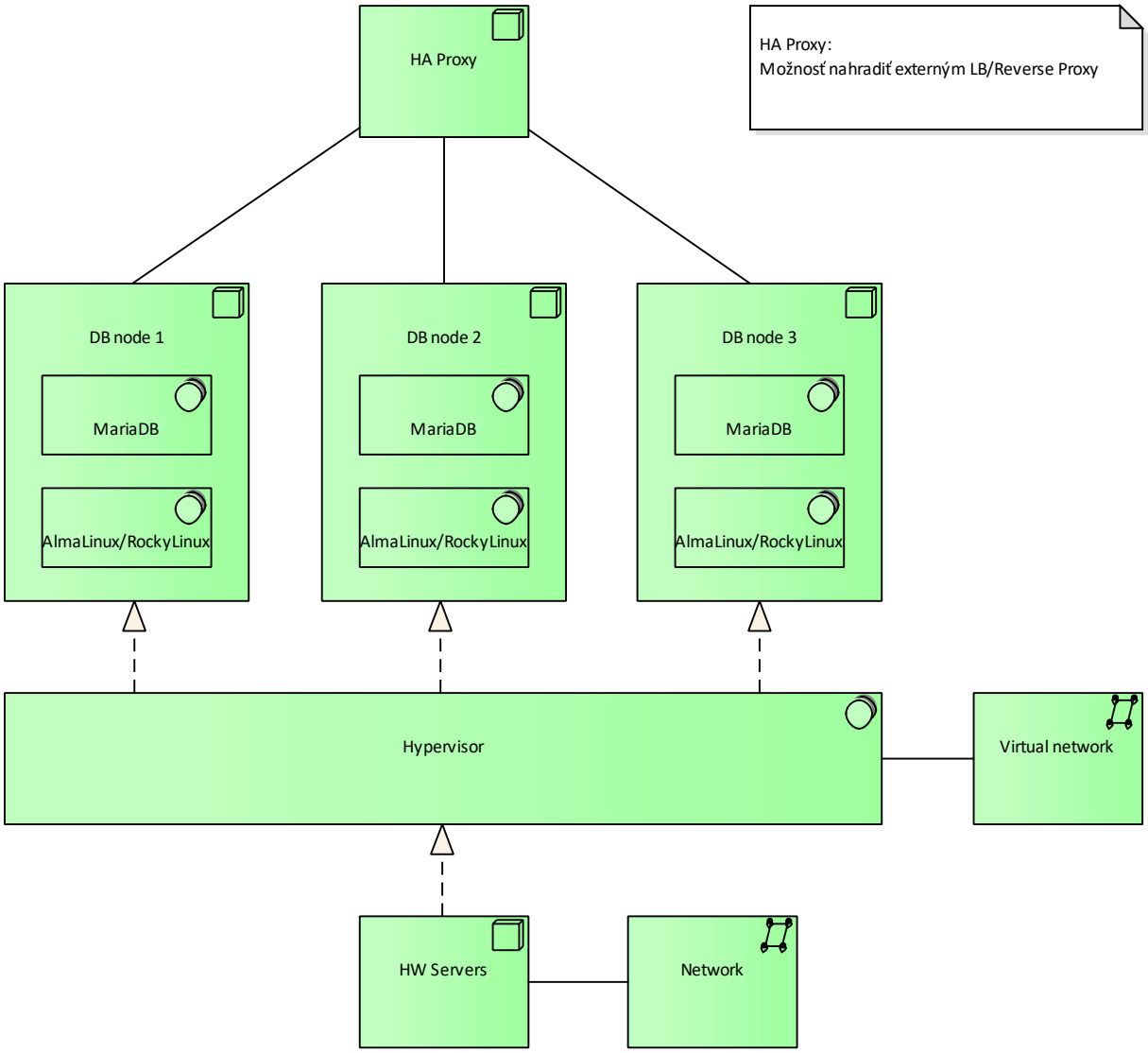


Figure 5 : Technology Viewpoint DB Cluster

The diagram shows the technological architecture of the PostgreSQL Cluster.

Description of the elements used:

DB node 1
Represents Node 2 of the MariaDB Galera Cluster

DB node 2
Represents Node 2 of the MariaDB Galera Cluster

DB node 3
Represents Node 3 of the MariaDB Galera Cluster

HA Proxy
Represents an internal reverse proxy built on HA Proxy

HW Servers
Represents the HW server infrastructure

Hypervisor

Represents a Type 1 or Type 2 hypervisor

Network
Represents the network/subnet of the HW infrastructure

Virtual network
Represents the virtual network/subnet of the hypervisor

Note
HA Proxy:
Can be replaced by an external LB/Reverse Proxy

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

MariaDB
Represents the MariaDB relational database

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

MariaDB
Represents the MariaDB relational database

AlmaLinux/RockyLinux
Represents the AlmaLinux/RockyLinux Linux distribution

MariaDB
Represents the MariaDB relational database

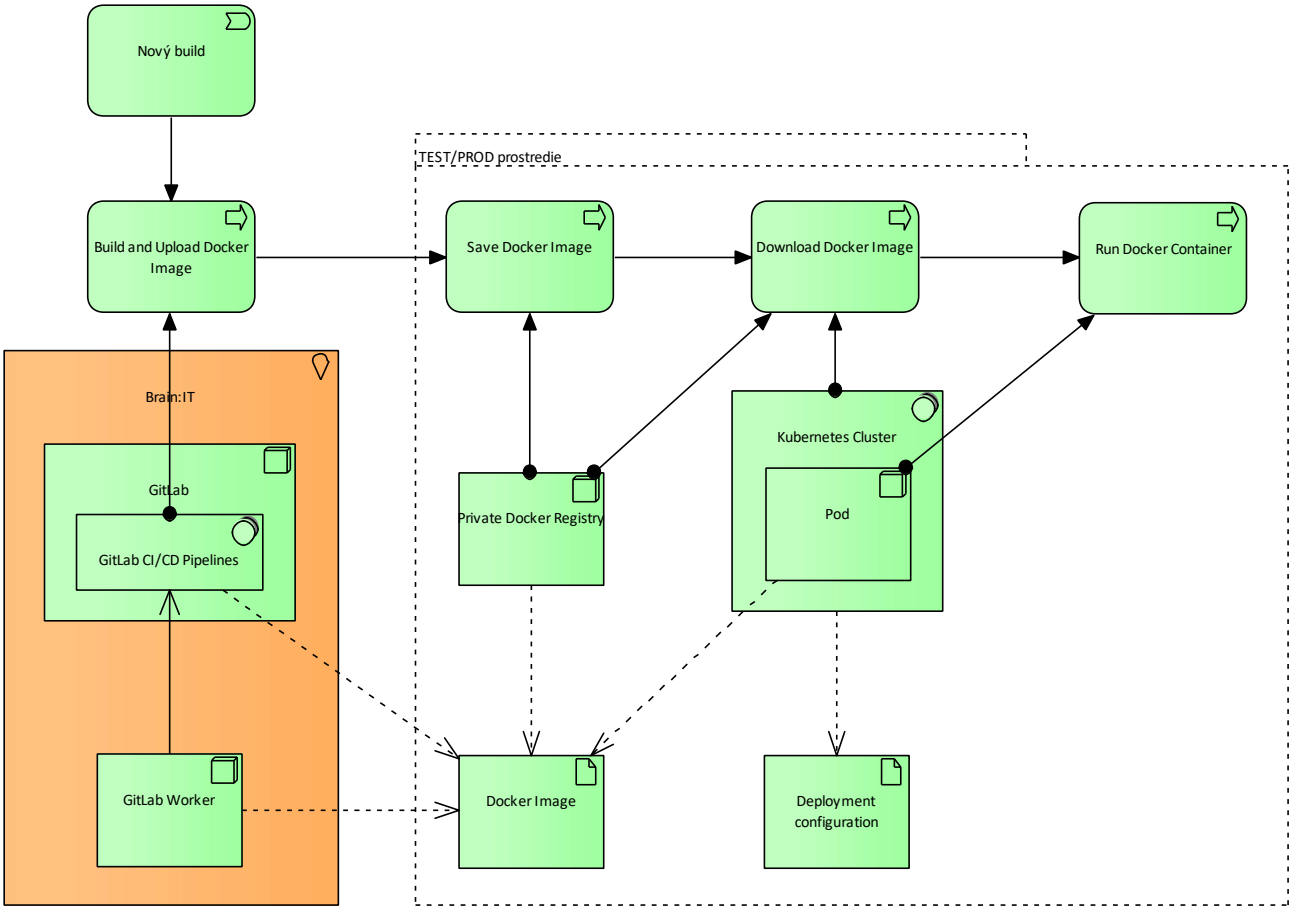


Figure 6 : Technology Viewpoint Development CI/CD TEST/PROD

The diagram shows the process of building and deploying a new version of a package in the form of a Docker image to the TEST/PROD environment. The diagram is applicable to the case of in-house packages; for other existing ones, the public DockerHub source is used.

Description of the elements used:

Build and Upload Docker Image
Build process and upload of the Docker Image to the Docker Registry

Download Docker Image
Process of storing the Docker Image in the Docker Registry

New build
Launch of a new build process

Run Docker Container
Process of launching a Docker Container

Save Docker Image
Process of storing the Docker Image in the Docker Registry

TEST/PROD environment
Represents the internal DEV and TEST environment

Brain:IT
Brain:IT site

Private Docker Registry
Represents a private Docker Registry

Deployment configuration
Representation of the Kubernetes Deployment artefact.

Docker Image
Representation of the Docker Image artefact.

GitLab
Represents a GitLab instance.

GitLab Worker
Represents a GitLab Worker instance

Kubernetes Cluster
Represents the Kubernetes Cluster

GitLab CI/CD Pipelines
GitLab CI/CD Pipelines are a series of tasks providing compilation, tests, configuration, deployment and other activities within GitLab.

Pod
Represents a Pod instance in which a new Docker Image is run in the form of a Docker Container

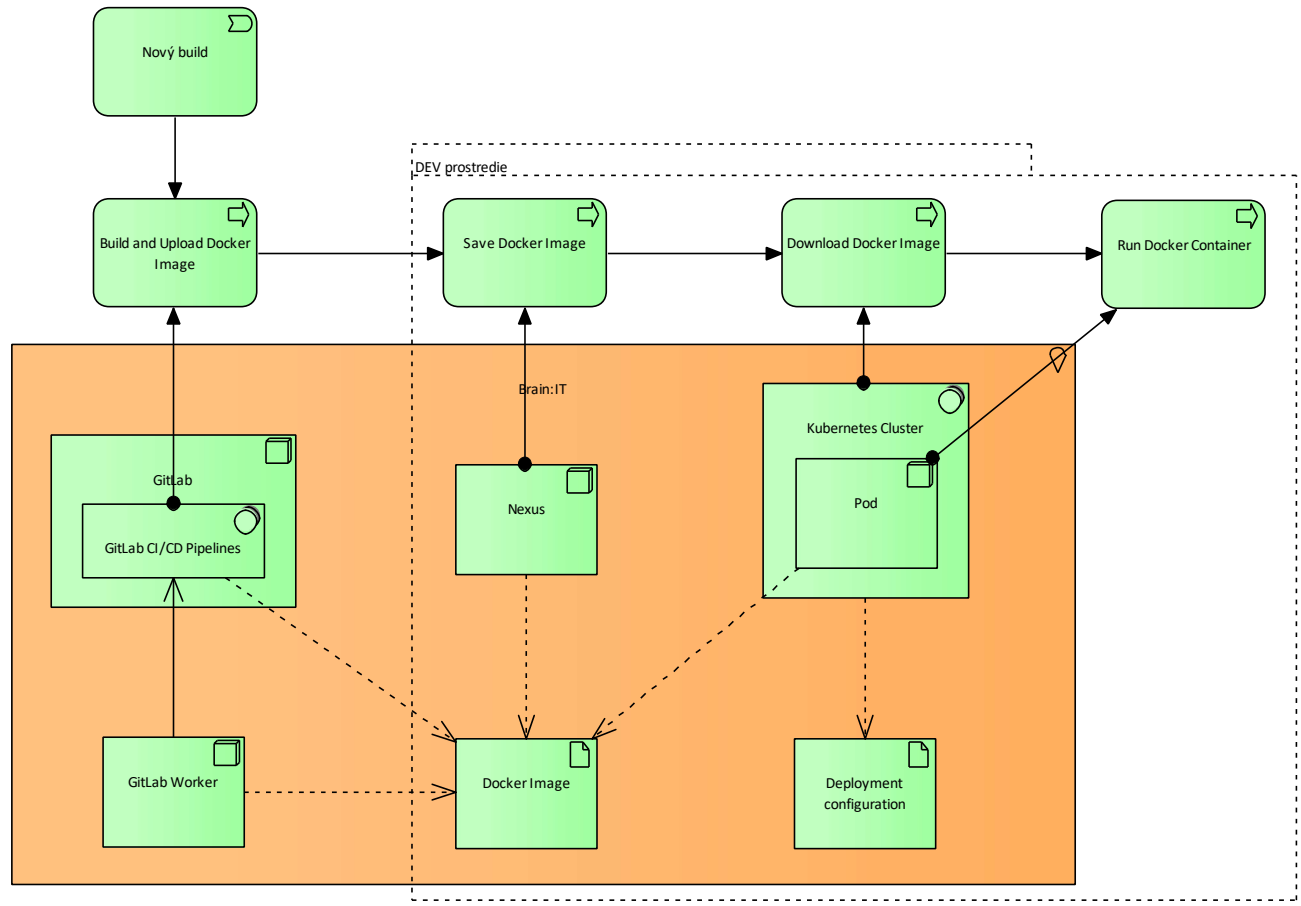


Figure 7 : Technology Viewpoint Developmnent CI/CD DEV

The diagram shows the process of building and deploying a new version of a package in the form of a Docker image to the DEV environment. The diagram is applicable to the case of in-house packages; for other existing ones, the public DockerHub source is used.

Description of the elements used:

Build and Upload Docker Image
Build process and upload of the Docker Image to the Docker Registry

Download Docker Image
Process of storing the Docker Image in the Docker Registry

New build
Launch of a new build process

Run Docker Container
Process of launching a Docker Container

Save Docker Image
Process of storing the Docker Image in the Docker Registry

DEV environment
Represents the internal DEV and TEST environment

Brain:IT
Brain:IT site

Deployment configuration
Representation of the Kubernetes Deployment artefact.

Docker Image
Representation of the Docker Image artefact.

GitLab
Represents a GitLab instance.

GitLab Worker
Represents a GitLab Worker instance

Kubernetes Cluster
Represents the Kubernetes Cluster

Nexus
Represents a private Docker Registry

GitLab CI/CD Pipelines
GitLab CI/CD Pipelines are a series of tasks providing compilation, tests, configuration, deployment and other activities within GitLab.

Pod
Represents a Pod instance in which a new Docker Image is run in the form of a Docker Container

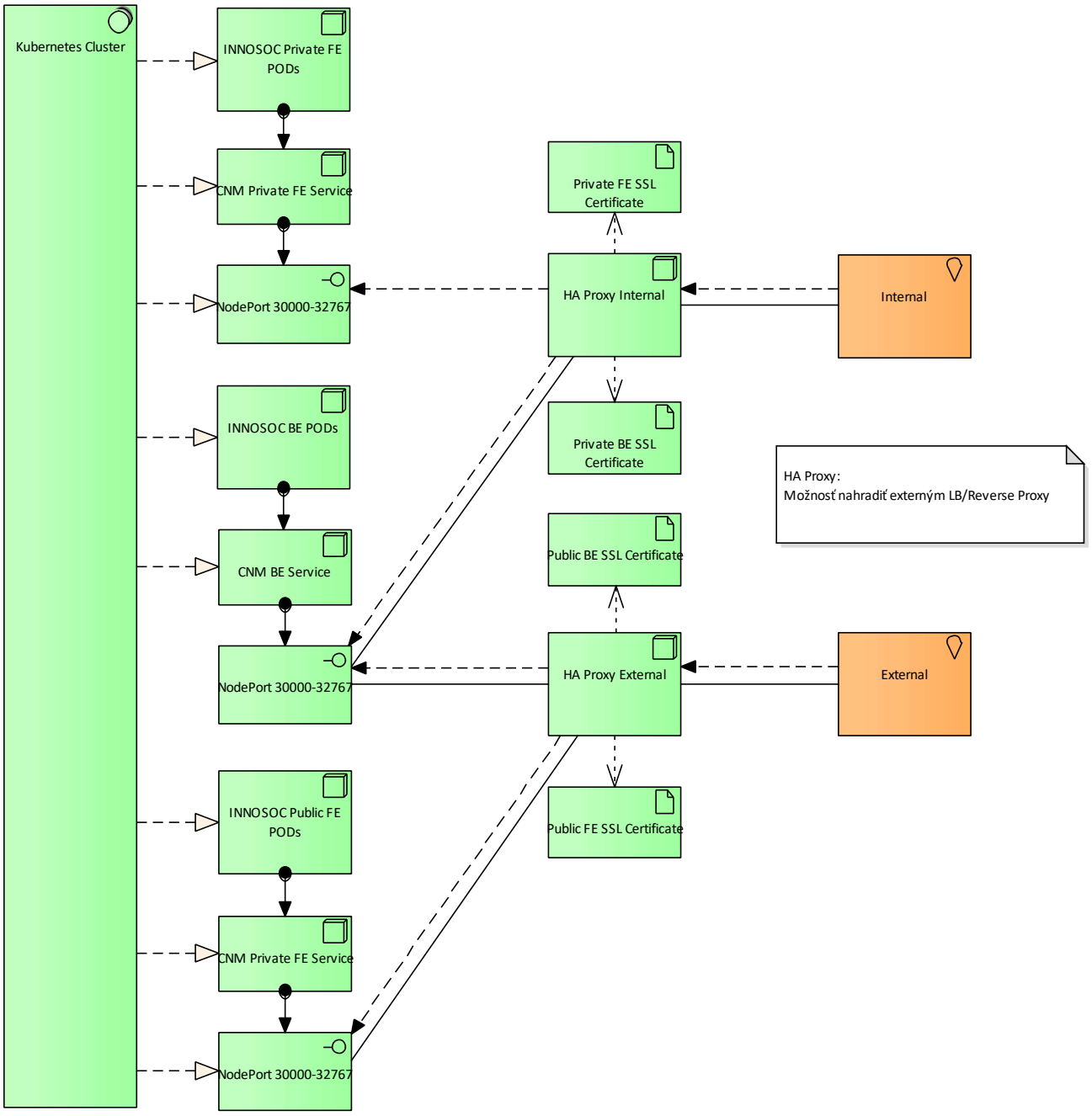


Figure 8 : Technology Viewpoint Interconnect

The diagram shows the deployment and interconnection of exposed Kubernetes services and the LB/Reverse Proxy for two zones.

Description of the elements used:

CNM BE Service
Represents CNM BE Service

CNM Private FE Service
Represents the CNM FE Private Service

CNM Private FE Service
Represents the CNM FE Private Service

External
Represents the external Internet environment

HA Proxy External
Represents an external reverse proxy built on HA Proxy

HA Proxy Internal
Represents an internal reverse proxy built on HA Proxy

Internal
Represents the internal BRAIN:IT environment

Private BE SSL Certificate
Represents a valid private SSL certificate for HTTPS purposes

Private FE SSL Certificate
Represents a valid SSL certificate for HTTPS purposes

Public BE SSL Certificate
Represents a valid public SSL certificate for HTTPS purposes

Public FE SSL Certificate
Represents a valid public SSL certificate for HTTPS purposes

NodePort 30000-32767
Represents NodePort-type routing from the range 30000-32767

NodePort 30000-32767
Represents NodePort-type routing from the range 30000-32767

NodePort 30000-32767
Represents NodePort-type routing from the range 30000-32767

Note
HA Proxy:
Can be replaced by an external LB/Reverse Proxy

Kubernetes Cluster
Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

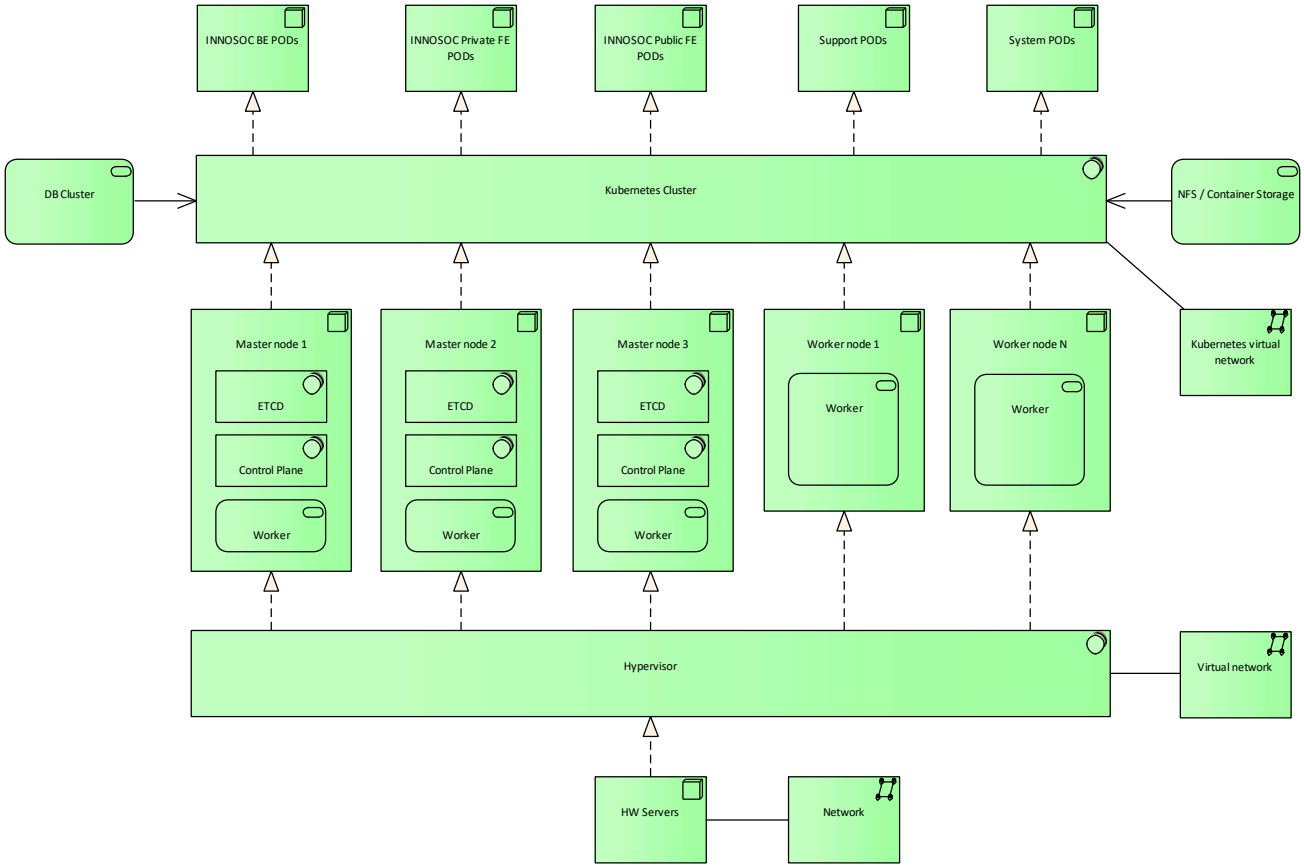


Figure 9 : Technology Viewpoint Kubernetes Cluster

Represents the role and service of the Worker node .

Description of the elements used:

DB Cluster
Represents the clustered relational database service

NFS / Container Storage
Represents the NFS service or another Container Storage

HW Servers
Represents the HW server infrastructure

Hypervisor
Represents a Type 1 or Type 2 hypervisor

Kubernetes virtual network
Represents the virtual network/subnet of the Kubernetes Cluster

Master node 1
Represents the Master node of the Kubernetes cluster

Master node 2
Represents the Master node of the Kubernetes cluster

Master node 3
Represents the Master node of the Kubernetes cluster

Network
Represents the network/subnet of the HW infrastructure

Support PODs
Represents the PODs of supporting services

System PODs
Represents the PODs of system services

Virtual network
Represents the virtual network/subnet of the hypervisor

Worker node 1
Represents a Worker node of the Kubernetes cluster

Worker node N
Represents a Worker node of the Kubernetes cluster

Kubernetes Cluster
Represents the Kubernetes Cluster

INNOSOC BE PODs
Represents the CNM BE PODs

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

Worker
Represents the role and service of the Worker node

Control Plane
Represents the Control Plane for managing the Kubernetes cluster

ETCD
Represents the ETCD database

Worker
Represents the role and service of the Worker node

Control Plane
Represents the Control Plane for managing the Kubernetes cluster

ETCD
Represents the ETCD database

Worker

Control Plane
Represents the Control Plane for managing the Kubernetes cluster

ETCD
Represents the ETCD database

Worker

Worker

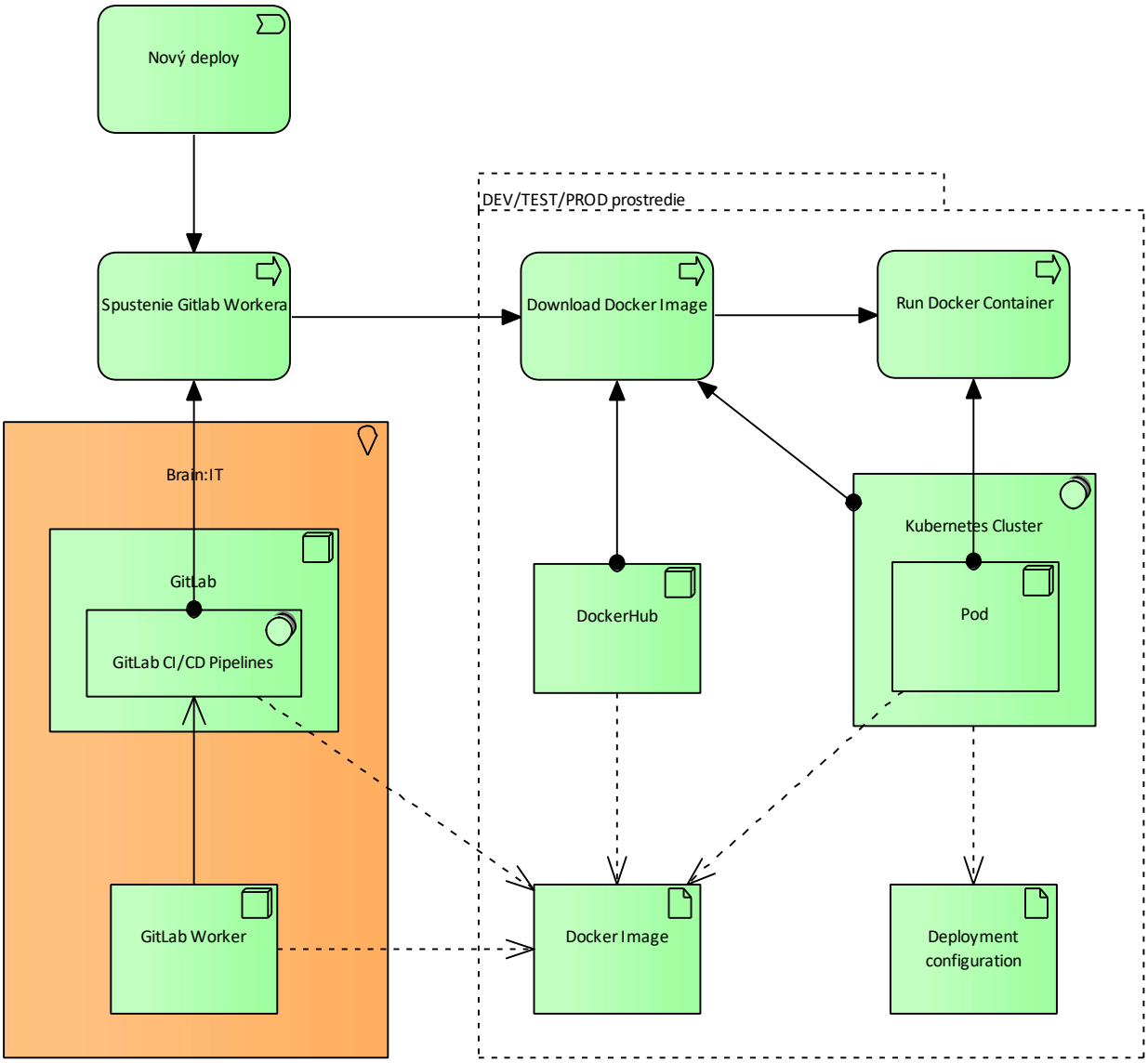


Figure 10 : Technology Viewpoint standard CI/CD DEV/TEST/PROD

Description of the elements used:

Download Docker Image
Process of storing the Docker Image in the Docker Registry

New deploy
Launch of a new deploy process

Run Docker Container

Process of launching a Docker Container

Starting the Gitlab Worker
process of launching the Gitlam Worker to deploy a new version of the package.

DEV/TEST/PROD environment
Represents the internal DEV and TEST environment

Brain:IT
Brain:IT site

DockerHub
Represents the public Docker Registry DockerHub

Deployment configuration
Representation of the Kubernetes Deployment artefact.

Docker Image
Representation of the Docker Image artefact.

GitLab
Represents a GitLab instance.

GitLab Worker
Represents a GitLab Worker instance

Kubernetes Cluster
Represents the Kubernetes Cluster

GitLab CI/CD Pipelines
GitLab CI/CD Pipelines are a series of tasks providing compilation, tests, configuration, deployment and other activities within GitLab.

Pod
Represents a Pod instance in which a new Docker Image is run in the form of a Docker Container

Technology Viewpoint OpenStack

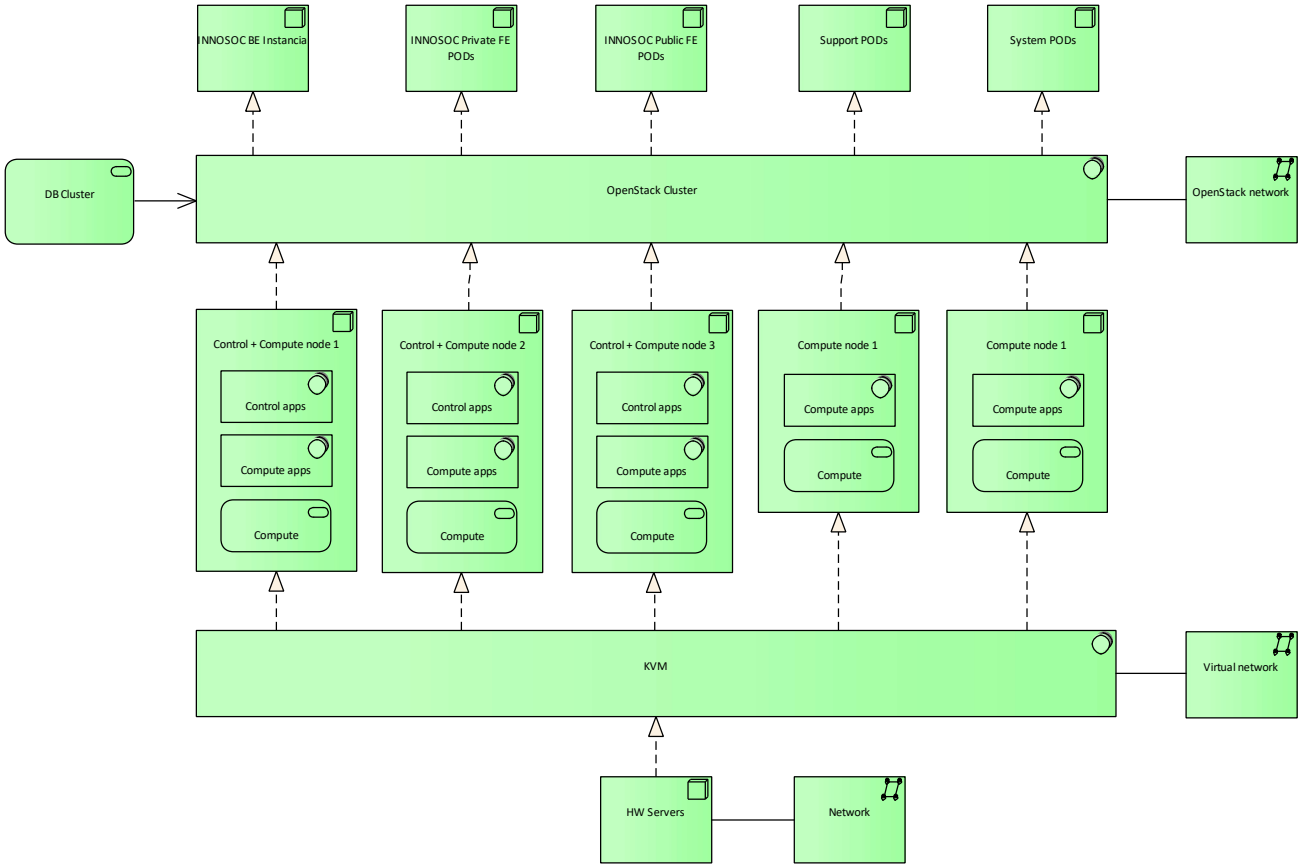


Figure 11 : Technology Viewpoint OpenStack Cluster

Description of the elements used:

DB Cluster
Represents the clustered relational database service

Compute node 1
Represents a Compute node of the OpenStack cluster

Compute node 1
Represents a Compute node of the OpenStack cluster

Control + Compute node 1
Represents Control and Compute node 1 of the OpenStack cluster

Control + Compute node 2
Represents Control and Compute node 2 of the OpenStack cluster

Control + Compute node 3
Represents Control and Compute node 3 of the OpenStack cluster

HW Servers
Represents the HW server infrastructure

KVM
Represents a type 1 KVM hypervisor.

Network
Represents the network/subnet of the HW infrastructure

OpenStack Cluster
Represents the OpenStack Cluster

OpenStack network
Represents the network/subnet of the OpenStack cluster, specifically:
Corporate network
OpenStack Tenant Data Network
OpenStack Internal Admin / API Subnet

Support PODs
Represents the PODs of supporting services

System PODs
Represents the PODs of system services

Virtual network
Represents the virtual network/subnet of the hypervisor

INNOSOC BE Instance
Represents an OpenStack VM instance or a container

INNOSOC Private FE PODs
Represents the CNM FE Private PODs

INNOSOC Public FE PODs
Represents the CNM FE Public PODs

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Control apps
Represents the set of applications for the Control node required for managing the cluster

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Control apps
Represents the set of applications for the Control node required for managing the cluster

Compute
Represents the role and service of the Compute node

Compute apps

Represents the set of Compute applications

Control apps
Represents the set of applications for the Control node required for managing the cluster

Compute
Represents the role and service of the Compute node

Compute apps
Represents the set of Compute applications

Technology Viewpoint OpenStack External

Technology Viewpoint Reporting

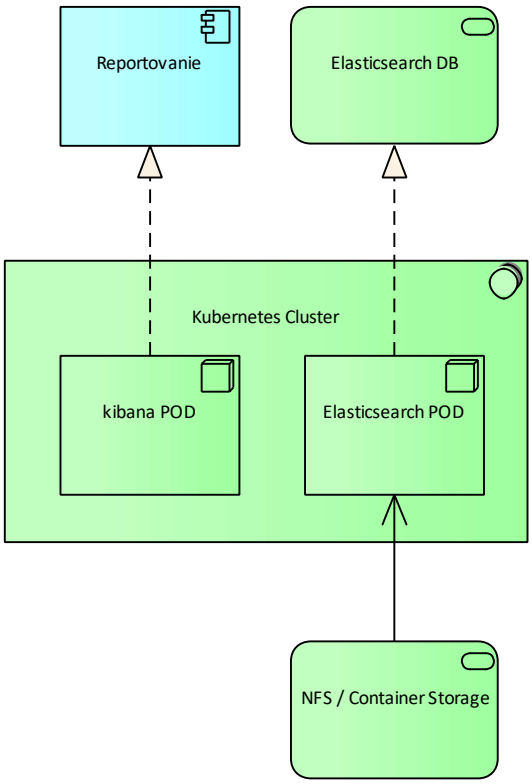


Figure 12 : Technology Usage Viewpoint Reporting alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages a containerized version of the NoSQL database for Elasticsearch.

Description of the elements used:

Reporting

The reporting component within the SOCaaS architecture serves as the central means for aggregating, visualizing and distributing security information. Its role is to provide both regular and ad hoc outputs on the state of cybersecurity for various target groups – from SOC analysts, through team leaders, to the management of the organization or clients. Such a reporting system must cover:

- Generation of metrics for incidents, alerts, SLAs and KPIs,
- Overviews of trends over time and assessment of the performance of security measures,
- Adaptation of content according to the level of detail (technical vs. strategic),
- Automation and scheduling of the distribution of outputs,
- Retrospective analysis and archiving of historical data.

NFS / Container Storage

Represents the NFS service or another Container Storage

Kubernetes Cluster

Represents the Kubernetes Cluster

Elasticsearch DB

Represents the technology service of the NoSQL Elasticsearch database.

Elasticsearch POD

Represents the Elasticsearch POD for the NoSQL database.

kibana POD

Represents the Kibana POD for Reporting.

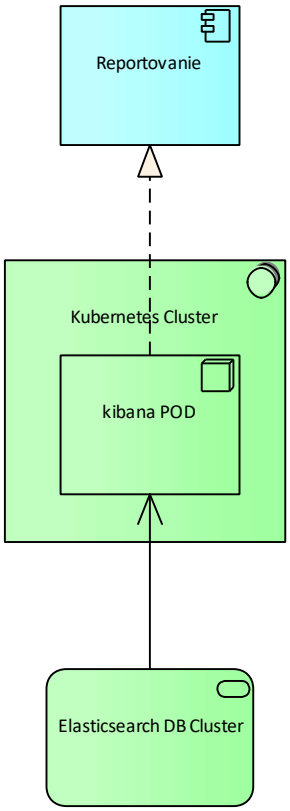


Figure 13 : Technology Usage Viewpoint Reporting alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 2 represents the use of the Elasticsearch database directly in the form of a centralized DB cluster.

Description of the elements used:

Reporting

The reporting component within the SOCaaS architecture serves as the central means for aggregating, visualizing and distributing security information. Its role is to provide both regular and ad hoc outputs on the state of cybersecurity for various target groups – from SOC analysts, through team leaders, to the management of the organization or clients. Such a reporting system must cover:

- Generation of metrics for incidents, alerts, SLAs and KPIs,
- Overviews of trends over time and assessment of the performance of security measures,
- Adaptation of content according to the level of detail (technical vs. strategic),
- Automation and scheduling of the distribution of outputs,
- Retrospective analysis and archiving of historical data.

Kubernetes Cluster

Represents the Kubernetes Cluster

Elasticsearch DB Cluster

Represents the technology service of the NoSQL Elasticsearch database in the form of a cluster.

kibana POD
Represents the Kibana POD for Reporting.

Technology Viewpoint Vulnerability Scanner

Technology Viewpoint Ticketing system / Incident registration and management

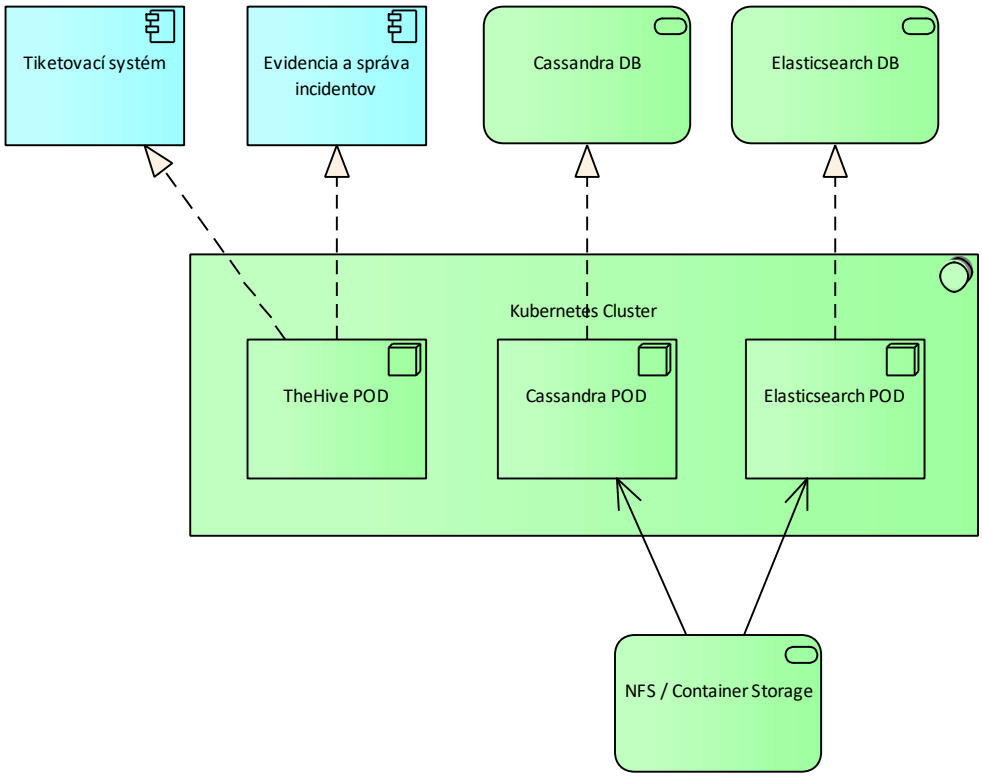


Figure 14 : Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 1 represents the use of the Elasticsearch database directly within the module setup. The design envisages a containerized version of the NoSQL databases for Cassandra and Elasticsearch.

Description of the elements used:

Incident recording and management

Incident recording and management is a core element of every security operation (SOC). This subcomponent ensures systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A high-quality recording system must enable teams to collaborate in a coordinated manner, document steps, record related artefacts (observables) and link incidents
with relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, auditing, multi-tenancy and the possibility of retrospective investigation of incidents.

Ticketing system

The ticketing system serves as the central communication and organizational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

NFS / Container Storage

Represents the NFS service or another Container Storage

Kubernetes Cluster

Represents the Kubernetes Cluster

Cassandra DB

Represents the technology service of the NoSQL Cassandra database.

Elasticsearch DB

Represents the technology service of the NoSQL Elasticsearch database.

Cassandra POD

Represents the Apache Cassandra POD for the NoSQL database.

Elasticsearch POD

Represents the Elasticsearch POD for the NoSQL database.

TheHive POD

Represents the TheHive POD for asset inventory.

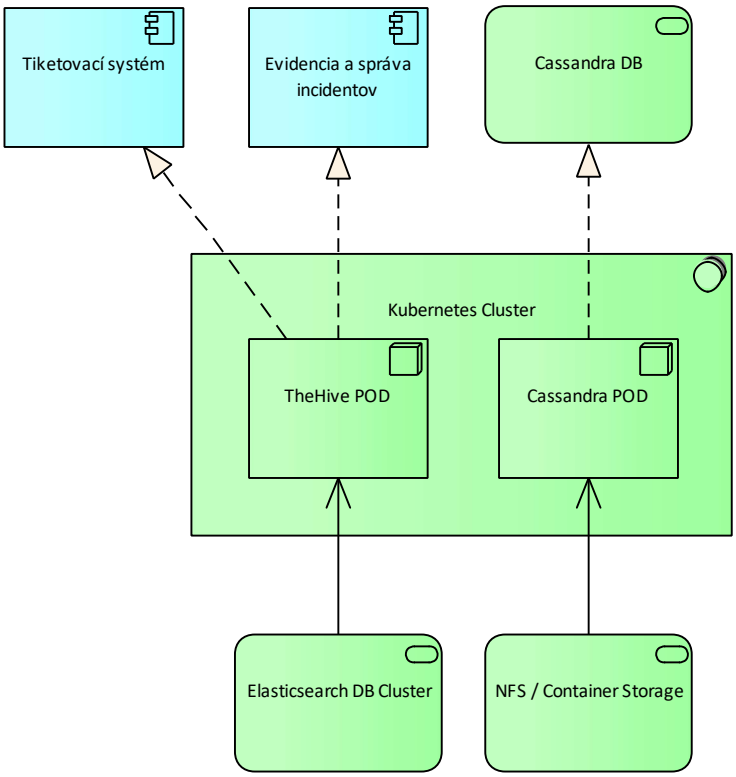


Figure 15 : Technology Usage Viewpoint Ticketing System / Incident Recording and Management alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 2 represents the use of the Elasticsearch database directly in the form of a centralized DB cluster. The design envisages a containerized version of the NoSQL database for Cassandra.

Description of the elements used:

Incident recording and management

Incident recording and management is a core element of every security operation (SOC). This subcomponent ensures systematic documentation, tracking and management of security events and incidents throughout their life cycle – from detection through analysis to closure. A high-quality recording system must enable teams to collaborate in a coordinated manner, document steps, record related artefacts (observables) and link incidents
with relevant threats, systems or historical cases. Ideally, it also supports advanced data enrichment, auditing, multi-tenancy and the possibility of retrospective investigation of incidents.

Ticketing system

The ticketing system serves as the central communication and organizational channel for the management of security incidents throughout their entire life cycle. It enables the creation, assignment and tracking of tasks and communication between analysts, while supporting audit, SLA monitoring, escalation and notifications.

NFS / Container Storage

Represents the NFS service or another Container Storage

Kubernetes Cluster

Represents the Kubernetes Cluster

Cassandra DB
Represents the technology service of the NoSQL Cassandra database.

Elasticsearch DB Cluster
Represents the technology service of the NoSQL Elasticsearch database in the form of a cluster.

Cassandra POD
Represents the Apache Cassandra POD for the NoSQL database.

TheHive POD
Represents the TheHive POD for asset inventory.

Technology Viewpoint Knowledge Portal for the SOC Team

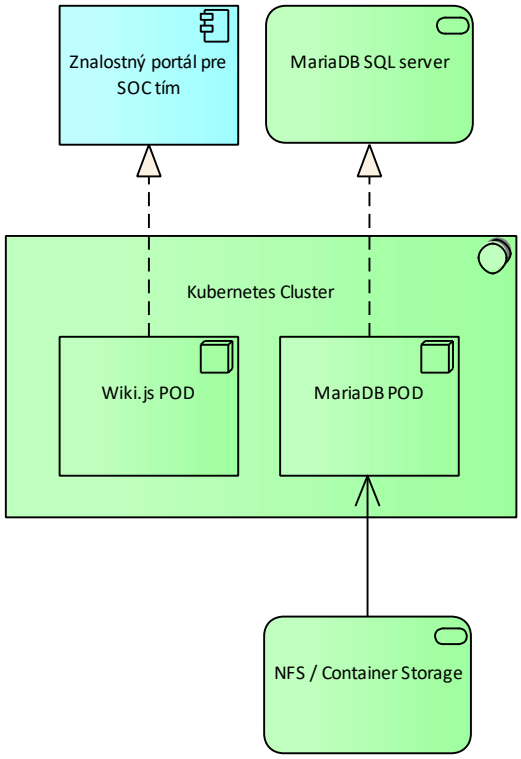


Figure 16 : Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 1

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 1 represents the use of the database directly within the module setup. The design envisages a

containerized version of the SQL server.

Description of the elements used:

Knowledge portal for the SOC team

A knowledge portal for the SOC team is an important building block of every modern security operations centre. It serves as a central repository of know-how, playbooks, procedures, recommendations, analytical guides and often also historical lessons learned from incidents. The knowledge portal:

- Increases the efficiency and consistency of incident response,
- Enables rapid onboarding of new team members,
- Supports the standardization of processes and methodologies,
- Enables the team to retain and share proven procedures and its own analytical experience,
- Strengthens the ability to learn collectively from incidents and to build internal "threat intel" knowledge.

NFS / Container Storage

Represents the NFS service or another Container Storage

Kubernetes Cluster

Represents the Kubernetes Cluster

MariaDB SQL server

Represents the technology service of the relational MariaDB SLQ database.

MariaDB POD

Represents the MariaDB POD for the relational SQL DB.

Wiki.js POD

Represents the Wiki.js POD for the Knowledge portal for the SOC team.

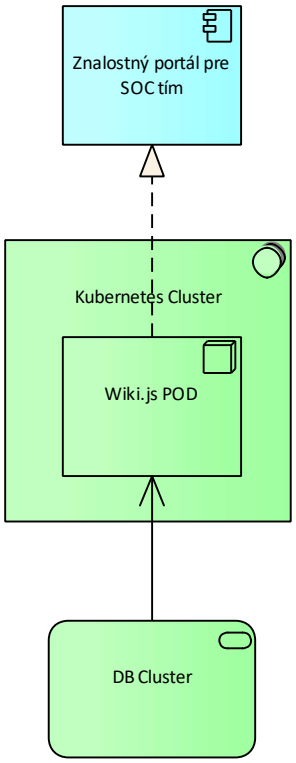


Figure 17 : Technology Usage Viewpoint Knowledge Portal for the SOC Team alt 2

The diagram shows a simplified assignment of the individual PODs to the corresponding application components.
Alt 2 represents the use of the database directly in the form of a centralized DB cluster.

Description of the elements used:

Knowledge portal for the SOC team

A knowledge portal for the SOC team is an important building block of every modern security operations centre. It serves as a central repository of know-how, playbooks, procedures, recommendations, analytical guides and often also historical lessons learned from incidents. The knowledge portal:

- Increases the efficiency and consistency of incident response,
- Enables rapid onboarding of new team members,
- Supports the standardization of processes and methodologies,
- Enables the team to retain and share proven procedures and its own analytical experience,
- Strengthens the ability to learn collectively from incidents and to build internal "threat intel" knowledge.

DB Cluster

Represents the clustered relational database service

Kubernetes Cluster

Represents the Kubernetes Cluster

Wiki.js POD

Represents the Wiki.js POD for the Knowledge portal for the SOC team.