

# Common Integration Platform User Manual



**SPHINX**

A Universal Cyber Security Toolkit for  
Health-Care Industry



# Table of contents

- 1 Introduction..... 3
- 2 Existing Infrastructures ..... 3
  - 2.1 VMs..... 3
- 3 Purpose/Installed Tools..... 3
- 4 Application Deployment..... 4
  - 4.1 Service Manager ..... 4
  - 4.2 Kubernetes API Service..... 5
  - 4.3 CST ..... 5
  - 4.4 Other SPHINX toolkit ..... 5

# List of tables

- Table 1 ICOM - VMs..... 3





# 1 Introduction

The Common Integration Platform is the framework that we use in order to run all the SPHINX applications. As such, the CIP should be understood as an enabler for the other SHPINX components to run and it does not present special interest for most of the SPHINX end users. In light of the above, this document is of interest primarily –if not exclusively- for IT administrators, responsible for deploying / maintaining the platform on top of which the various SPHINX applications run, and not for the broader set of end users (e.g. medical personnel, other hospital employees).

## 2 Existing Infrastructures

We have set up 4 clusters of physical / virtual machines for the needs of the project.

### 2.1 VMs

Table 1 ICOM - VMs

Location	Master	Node1	Node2	DNS name
ICOM_test	146.124.106.170	146.124.106.171		Sphinx-kubernetes.intracom-telecom.com
ICOM_prod	146.124.106.181	146.124.106.182	146.124.106.183	Sphinx-toolkit.intracom-telecom.com
DYPE5	10.10.2.60	10.10.2.61	10.10.2.62	
HESE	172.17.67.1	172.17.67.2	172.17.67.3	

## 3 Purpose/Installed Tools

The cluster ICOM\_test is used for testing purposes.

Installed tools:

- Kafka
- Ingress
- Service Manager
- Kubernetes API Service
- SPHINX Components for testing

The cluster ICOM\_prod is for the deployment of SPHINX components [subset of the full SPHINX Toolkit].

Installed tools:





- Kafka
- Ingress
- Local Storage Provisioner
- Service Manager
- Kubernetes API Service
- Prometheus
- Kuberhealthy
- SPHINX Components

The cluster DYPE5 is for the deployment of SPHINX components [subset of the full SPHINX Toolkit].

- Installed tools:
- Kafka
- Ingress
- Local Storage Provisioner
- Prometheus
- Kuberhealthy
- SPHINX Components

The cluster on HESE is for deployment of the E-care Platform and a subset of the full SPHINX Toolkit.

## 4 Application Deployment

### 4.1 Service Manager

The Service Manager (SM) component is a generic component of the SPHINX Universal Cyber Security Toolkit used for service management, authentication and authorization purposes and SSO functionalities for end users. It keeps the registry of the different SPHINX components of the SPHINX Universal Cyber Security Toolkit and system services required for the operation of the SPHINX Universal Cyber Security Toolkit and acts as an authorization server for the needs of the direct interaction among SPHINX components, their interaction through Kafka or the interaction among end users and the SPHINX Universal Cyber Security Toolkit.

All files to deploy the Service Manager can be found on the Gitlab server's [repo](#). The instructions below refer to the deployment of the Service Manager to a Kubernetes environment.

To deploy the Service Manager to the Kubernetes environment you have to be able to run the below commands in the master of the cluster you want to deploy it.

```
kubectl apply -f deployment.yml
```

After the deployment run the below command and when it finishes the deployment is ready.

```
kubectl wait pod --l app=service-manager --for=condition=Ready --timeout=120s
```





## 4.2 Kubernetes API Service

The Kubernetes API (K-API) service is a tailored service that makes use of the Kubernetes API to provide an interface for the deployment, lifecycle management and monitoring of the SPHINX components on the cluster.

All files to deploy the Kubernetes API Service can be found on the Gitlab server's [repo](#). The instructions below refer to the deployment of the Kubernetes API service to a Kubernetes environment.

To deploy the Kubernetes API service to the Kubernetes environment you have to be able to run the below commands in the master of the cluster you want to deploy it.

```
kubectl apply -f deployment.yml
```

After the deployment run the below command and when it finishes the deployment is ready

```
kubectl wait pod -l app=kube-service --for=condition=Ready --timeout=120s
```

## 4.3 CST

All files to deploy the Kubernetes API Service can be found on the Gitlab server's [repo](#).

The instructions below refer to the deployment of the CST to a Kubernetes environment.

To deploy the CST to the Kubernetes environment you have to be able to run the below commands in the master of the cluster you want to deploy it.

```
kubectl apply -f k8sdeployment.yml
```

After the deployment run the below command and when it finishes the deployment is ready.

```
kubectl wait pod -l app=cst --for=condition=Ready --timeout=120s
```

## 4.4 Other SPHINX toolkit

To deploy other SPHINX components please refer to the CST User Manual.

