



NGSOC

Next Generation Security Operations Centres

Next Generation Security Operation Centres

D5.2 – Hands-on educational platform			
Report Identifier:	D5.2		
Work-package:	WP5	Task:	T5.2
Responsible Partner:	EDGR	Version Number:	1.0
Due Date:	30/06/2026	Document Date:	30/05/2026
Distribution Security:	PUB	Deliverable Type:	OTHER
Keywords:			
Project website: https://ng-soc.eu/			

Document History

Version	Content & Changes	Issue Date
0.1	Document created	17/03/2026
0.2	Document sent for review	22/06/2026
0.3	Document reviewed	24/6/2026
0.4	Document reviewed	25/6/2026
0.5	Reviews are combined	29/06/2026
0.6	Sent for Quality Assurance	29/06/2026
1.0	Quality Assurance and Submission	30/06/2026

Quality Control

	Name	Organisation	Date
Editor	Georgios Chatzipapas	European Dynamics S.A (EDGR)	22/06/2026
Peer review 1	Andreas Menegatos	University of Piraeus (UPRC)	24/6/2026
Peer review 2	Jose-Maria Romero	FUNITEC (FUNI)	25/6/2026
Authorised by (Technical Coordinator)	Vasileios Mavroeidis	CYEN	30/06/2026
Authorised by (Quality Manager)	Fryni Pechlidou	European Dynamics S.A	30/06/2026
Submitted by (Project Coordinator)	Apostolos Gkletos	European Dynamics S.A	30/06/2026

Legal Disclaimer

NG-SOC is an EU project funded by the Digital Europe Programme (DIGITAL) under grant agreement No. 101145874. The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any specific purpose. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. The NG-SOC Consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

Copyright notice

© Copyright by the NG-SOC Consortium

This document contains information that is protected by copyright. All Rights Reserved. No part of this work covered by copyright hereon may be reproduced or used in any form or by any means without the permission of the copyright holders.

Table of Contents

Table of Contents	4
List of Figures	6
List of Tables	7
Abbreviations	8
Executive Summary.....	10
1 Introduction	11
1.1 Overview	11
1.2 Relation to other tasks and deliverables	11
1.3 Deliverable Structure	12
2 Component Overview	13
2.1 Operational Role	13
2.2 Supported Use Cases	13
3 Functional Specifications	14
3.1 Core Functionalities	14
3.2 Input Data Model.....	15
3.3 Output Data Model	16
3.4 User Roles and Access Control.....	17
4 Technical Architecture	18
4.1 Internal Architecture.....	18
4.2 Technology Stack	18
4.3 Deployment.....	19
4.4 Performance	20
5 Interfaces and Integration	21
5.1 Internal NG-SOC Interface	21
5.2 External Interfaces	22
6 Security Considerations	24
6.1 Authentication and Authorization	24
6.2 Data Protection	24
7 Limitations and Future Enhancements	26
7.1 Current Limitations	26

7.2	Planned Enhancements	26
8	Conclusion.....	28
	Appendix A – Technical Detail	29
	Appendix B – Traceability Matrix.....	31
	Appendix C – Asset Specification	32
	REFERENCES.....	34

List of Figures

Figure 1 High-Level Architecture[1] 15

Figure 2 Training Scenario Instances Overviews..... 16

Figure 3 CyberRangeCZ (KYPO) Training Platform Data Flow 17

Figure 4 Topology of a Scenario..... 22

Figure 5 Welcome Screen 29

Figure 6 Login Screen 30

Figure 7 Access Code Entry 30

List of Tables

Table 1 Inputs from related Deliverables..... 11

Table 2 Platform Technology Stack by Layer 18

Table 3 Traceability Matrix 31

Table 4 CyberRangeCZ Asset Specification 32

Abbreviations

Acronym	Description
EU	European Union
GA	Grant Agreement
WP	Work Package
API	Application Programming Interface
AWS	Amazon Web Services
CACAO	Collaborative Automated Course of Action Operations
CICMS	Collaborative Incident Case Management System
CTF	Capture the Flag
CTI	Cyber Threat Intelligence
CTI-SS	Cyber Threat Intelligence Sharing System
ECSF	European Cybersecurity Skills Framework
GDPR	General Data Protection Regulation
HA	High Availability
JWT	JSON Web Token
KVM	Kernel-based Virtual Machine
NG-SIEM	Next Generation Security Information and Event Management
NG-SOAR	Next Generation Security Orchestration, Automation and Response
OIDC	OpenID Connect
QEMU	Quick Emulator
RBAC	Role-Based Access Control
REST	Representational State Transfer
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation and Response
SOC	Security Operations Centre
SSO	Single Sign-On
STIX	Structured Threat Information Expression

TAXII	Trusted Automated Exchange of Intelligence Information
TLS	Transport Layer Security
UI	User Interface
VM	Virtual Machine
YAML	YAML Ain't Markup Language

Executive Summary

SOC analysts learn effectively when they can investigate, test and respond to realistic incidents in a safe environment. Deliverable D5.2 - Hands-on educational platform presents the Hands-on Educational Platform developed under Task 5.2 of the NG-SOC project (Grant Agreement 101145874, Digital Europe Programme) - a scalable, fully isolated cyber range that provides analysts, incident responders, and cybersecurity educators with such a controlled environment. The platform serves the project's three pilot sectors: financial (CaixaBank), energy (ELES/INFO), and academic (CYNET).

The platform is built on the CyberRangeCZ (KYPO) open-source distribution [1], version v2025.03 [4] - the successor to the CyberRangeCZ (KYPO) Platform developed at Masaryk University. CyberRangeCZ follows a container-based microservices architecture deployed on Kubernetes, with OpenStack as the cloud backend for sandbox provisioning. Each training session runs inside a fully isolated virtual network where trainees interact with realistic operating systems and network topologies through a web browser, with no risk to production infrastructure.

The platform supports three types of training. Linear scenarios guide every trainee through the same tasks in a fixed order. Adaptive scenarios allow the platform's Smart Assistant to choose the next task based on how the trainee is performing. Capture-the-Flag exercises provide automated scoring for challenge-based training. NG-SOC components run inside the sandboxes, so trainees practise playbook authoring, log analysis, and incident response on the same tools they will operate in production. A BSides CTF provided an initial validation of the deployment under realistic load conditions with 15 concurrent sandboxes, each containing 9–10 VMs, sustained on approximately 490 GB of RAM with no performance degradation. Authentication is handled via Keycloak (OIDC), and a four-tier role model (Administrator, Trainer, Trainee, Designer) governs access.

During the remaining project period, the main activity will be the population of the platform with the sector-specific exercise scenarios produced under D5.3 - Open cybersecurity training and exercise scenarios. Together with the platform, these scenarios complete the WP5 capacity-building objectives and feed into the WP6 integration and validation phase.

1 Introduction

1.1 Overview

Effective cybersecurity training requires realistic and controlled environments where analysts can practise attack-and-defence scenarios without risk to operational systems. This deliverable reports on the work performed under Task 5.2 (Hands-on educational platform) of Work Package 5 (Capacity Building). It describes the selection, deployment, and configuration of the CyberRangeCZ Platform [1] as the hands-on training engine for the NG-SOC project. CyberRangeCZ is an open-source cyber range platform that builds on the KYPO Cyber Range Platform developed at Masaryk University [3].

The platform provides isolated, on-demand network environments where SOC analysts, cybersecurity educators, and students can engage in realistic attack-and-defence exercises without risk to production systems. OpenStack provisions the virtual infrastructure, including machines, routers, and network segments, while Ansible automates the installation of operating systems, tools, and vulnerable services inside each sandbox. Trainee progress is tracked in real time through flag submissions, timed tasks, and automated scoring. The deployment uses CyberRangeCZ Platform [1] version v2025.03, running in its default open-source configuration with no project-specific modifications to the source code. This deliberate choice keeps the NG-SOC deployment aligned with upstream releases and positions it to benefit directly from community-driven improvements.

Through this deployment, Task 5.2 has delivered a scalable and reusable hands-on educational platform for designing, delivering, and running cybersecurity training activities. The work described in this deliverable therefore demonstrates the full and timely completion of the Task 5.2 objectives.

1.2 Relation to other tasks and deliverables

D5.2 - Hands-on educational platform connects the project requirements, architecture, skills development activities and training content. It builds on the requirements and architectural decisions defined in WP2, the skills development strategy produced under T5.1, and the exercise scenarios developed under T5.3. In turn, it provides the runtime environment that WP6 uses for integration testing and validation. Table 1 summarises the main project inputs relevant to D5.2.

Table 1 Inputs from related Deliverables

Deliverable	Description	Relevance to D5.2
D2.3	User and system requirements for secure digital infrastructures	Defines requirements that guide the platform capabilities and expected use cases
D2.4	NG-SOC Architecture	Provides architectural decisions and integration points relevant to the platform deployment
D5.1	Skills development strategy report	Provides the training and capacity-building context addressed by the platform

D5.3	Open cybersecurity training and exercise scenarios	Provides the sector-specific training scenarios to be hosted and executed through the platform
------	--	--

1.3 Deliverable Structure

The deliverable is structured in eight chapters:

- **Chapter 1** introduces the deliverable, its scope, and its relation to other project outputs.
- **Chapter 2** provides a component overview and describes the operational role and supported use cases of the platform.
- **Chapter 3** specifies the functional capabilities, including input/output data models and user roles.
- **Chapter 4** details the technical architecture, technology stack, deployment approach, and performance considerations.
- **Chapter 5** describes the interfaces with other NG-SOC components and external systems.
- **Chapter 6** addresses security considerations, including authentication and data protection.
- **Chapter 7** discusses current limitations and planned enhancements.
- **Chapter 8** draws conclusions and maps the work to the project roadmap.

2 Component Overview

2.1 Operational Role

The Hands-on Educational Platform serves as the practical training engine of NG-SOC. Built on the CyberRangeCZ Platform (v2025.03) [1], it creates fully virtualised network environments including routers, firewalls, and vulnerable services, where trainees interact with real operating systems and real tools under controlled conditions. Training traffic remains inside the sandbox, ensuring that production systems are not exposed. Within the NG-SOC architecture, the platform fulfils four operational roles:

- **Skill Validation:** Providing a controlled environment to test the proficiency of analysts in using the NG-SIEM, NG-SOAR, and CTI-SS, PenTest, CICMS, DRME components deployed inside the sandbox.
- **Adversary Simulation:** Running multi-stage attack scenarios, including Red vs. Blue team exercises, where trainees must detect, investigate, and respond to live threats within an isolated network.
- **Curriculum Delivery:** Hosting interactive labs and Capture-The-Flag (CTF) exercises aligned with the training goals defined in D5.1 [7] and mapped to the European Cybersecurity Skills Framework (ECSF) [8].
- **Infrastructure Sandboxing:** Allowing security implementers to test new detection rules, playbook configurations, or network topologies in a virtualised replica of their pilot infrastructure before deploying changes to production.

2.2 Supported Use Cases

The three NG-SOC pilots operate in different sectors and at different maturity levels, so the platform supports each one differently. The use cases below describe how each pilot uses the cyber range.

- **Pilot 1 - Financial sector (CaixaBank) and Pilot 3 (ELES/INFORMATIKA):** The platform gives professional SOC analysts an isolated environment to rehearse incident response against sector-specific threats. CaixaBank uses it for automated Red Team assessments and analyst training delivered through the CyberRangeCZ training module, exercising detection and response to ransomware, malware, data-leak, and business-continuity scenarios. ELES/INFORMATIKA runs intelligence-driven threat scenarios in a controlled testbed that replicates the transmission-grid environment, enabling response speed to be tested without connecting to live systems. Validated NG-SOAR playbooks move from the sandbox to production for pilots that operate live infrastructure.
- **Pilot 2 - Academic sector (CYNET):** This is the primary hands-on training use case. The platform provides infrastructure for university students to practise network defence, participate in structured CTF competitions, and complete ECSF-aligned lab exercises[8]. This pilot also serves as a testbed for the training scenarios developed under D5.3.

3 Functional Specifications

3.1 Core Functionalities

The CyberRangeCZ Platform provides a set of core functionalities that map directly to the user and system requirements specified in D2.3 - User and system requirements [5] for secure digital infrastructures. This section describes each functional area, the data models consumed and produced by the platform, and the role-based access control model that governs user permissions. Figure 1 illustrates the high-level architecture, consisting of a front-end portal backed by containerised microservices that orchestrate sandbox provisioning on OpenStack.

1. **Isolated Sandboxing:** Creation of complex network topologies on OpenStack, isolated from production networks and configured to prevent unintended external exposure. Each sandbox instantiates a set of virtual machines (VMs), routers, and internal networks defined in YAML/JSON topology files. During a BSides CTF exercise hosted on the NG-SOC CyberRangeCZ instance, 15 concurrent sandboxes, each comprising 9–10 VMs, ran simultaneously without observed performance degradation.
2. **Scenario Management:** A web-based portal for trainers to design, deploy, and monitor multi-stage training scenarios. Scenario definitions are version-controlled in Git repositories and loaded by the platform at deployment time. The v2025.03 [4] release adds GitHub repository integration, enabling collaborative scenario development through standard Git workflows.
3. **Automated Assessment:** Real-time tracking of trainee progress through flag submission, in a CTF-style format, and automated evaluation of response actions. The platform supports both linear training, based on a fixed sequence of tasks, and adaptive training, based on branching paths according to trainee performance. The Smart Assistant provides decision logic for adaptive scenarios.
4. **NG-SOC Component Integration:** The training sandboxes include deployed instances of NG-SIEM and NG-SOAR, allowing trainees to practise log analysis, alert triaging, and playbook execution within the same toolset used in production. This addresses D2.3 [5] requirement Req-006, which concerns the operation of NG-SOAR within CyberRangeCZ,
5. **User Progress Tracking:** The platform records trainee answers, completion times, and scores through the Answers Storage and Training Feedback microservices. Separately, an OpenEdX instance hosts the theoretical training curriculum, which is aligned with the skills development strategy defined in D5.1 [7]. The two systems operate independently.

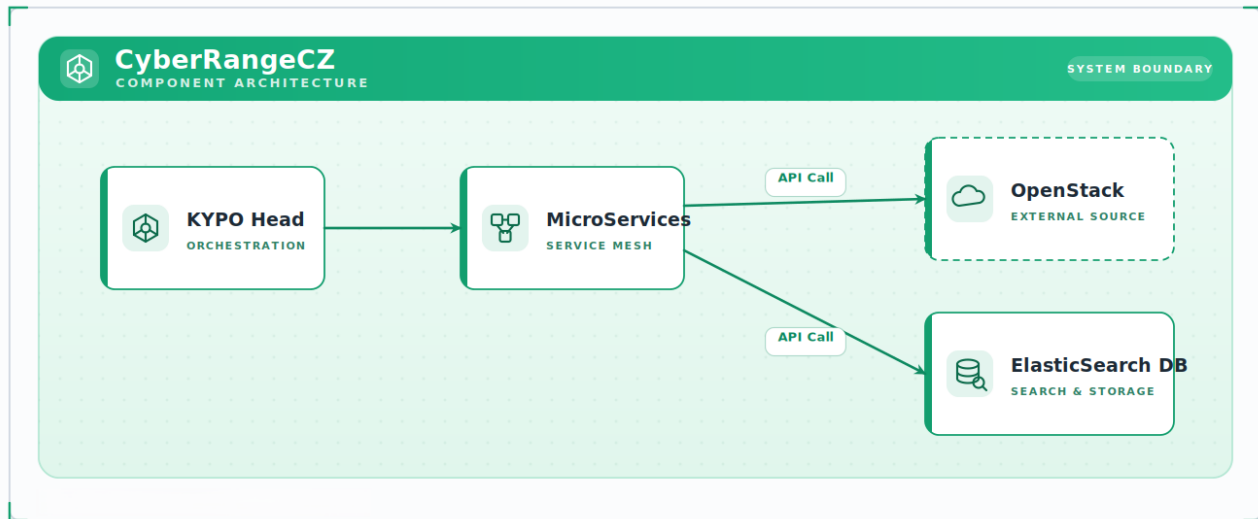


Figure 1 High-Level Architecture[1]

3.2 Input Data Model

Every exercise begins with a set of input artefacts that define the sandbox topology, the services and tools deployed inside it, and the way trainees' performance is assessed. The platform consumes four categories of input:

- **Scenario definitions:** Sandbox topology files written in YAML/JSON specify the virtual machines, networks, routers, and their interconnections. Each file acts as a declarative blueprint that lists the VM image to boot, the IP addressing scheme, the routing rules, and the firewall posture. Ansible playbooks and Packer templates then define the content installed on each VM, including operating systems, tools, vulnerable services, and pre-positioned artefacts for trainees to discover.
- **Training definitions:** JSON-based training definitions describe the sequence of tasks, expected flags, hints, and scoring rules. A linear training definition enforces a fixed task order. An adaptive definition includes branching conditions evaluated by the Smart Assistant at runtime. These files are loaded from a Git repository, either local or GitHub-based, configured during deployment.
- **Training curricula:** The skill development strategy and learning objectives produced in D5.1 [7] inform scenario design and trainee assessment criteria.
- **Baseline infrastructure templates:** OpenStack images and flavours that mirror the pilot environments, including Pilot 1 banking infrastructure, Pilot 2 academic lab, and Pilot 3 energy SCADA network, serve as the base layer for sandbox provisioning.

Title	Start Time	End Time	Expires in	Training Definition	Last Edit By	Pool	Pool Size	Access Token	Actions
LMI-1a-2: Incident Response and Forensics	22 Jun, 2028 10:59	21 Aug, 2028 10:59	49days	Subcase 1a - Instructor-led training of incidentresponse-ops (LMI)		Pool 77	2 (1 Free)	inst-1a-2-3430	[Icons]
LMI-1b: Risk Management	22 Jun, 2028 10:57	31 Aug, 2028 10:57	68days	1.1 Risk Management and threat modeling		Pool 78	2 (0 Free)	inst-1b-1000	[Icons]
PUCC-Sub Case 2b: Network Vulnerability Identification Training	21 Jun, 2028 10:55	31 Jul, 2028 10:55	37days	PUCC-Sub Case 2b: Network Vulnerability Identification Training		Pool 81	1 (0 Free)	inst-1b-1000	[Icons]
LMI Sub Case 1a-2: Incident Response and Forensics	17 Jun, 2028 17:02	27 Jun, 2028 17:02	10days	MD-500 Learning Module 8 - Incident Response and Forensics (Sub Case 1a)		Pool 87	1 (0 Free)	inst-1a-3-1017	[Icons]
PUCC-Sub Case 2a: Phishing Attack Training Scenario	8 Jun, 2028 10:08	27 Jun, 2028 10:08	19days	PUCC-Sub Case 2a: Phishing Attack Training Scenario		Pool 82	1 (0 Free)	inst-1a-3096	[Icons]
Junior test	9 Jun, 2028 18:13	12 Jun, 2028 18:13	expired	LMI0 Penetration Tools		Pool 81	2 (0 Free)	inst-1a-4447	[Icons]
Test Logging	9 Jun, 2028 10:12	1 Jul, 2028 10:12	1days	LMI0 Penetration Tools		Pool 78	1 (0 Free)	inst-1a-1418	[Icons]

Figure 2 Training Scenario Instances Overviews

3.3 Output Data Model

Although each sandbox is destroyed after the training session ends, the platform retains structured data about the activity that occurred during the session. Figure 3 summarises the end-to-end data flow, from content ingestion to sandbox delivery. The platform produces four categories of output:

- **Trainee performance data:** Per-trainee records of flag submissions, timestamps, scores, and hints used, stored in the Answers Storage service and queryable through the platform API.
- **Training Feedback reports:** Aggregated session-level reports that summarise class performance, identify skill gaps, and highlight areas requiring additional training. Trainers can review these reports through the web portal.
- **Session logs:** System and network logs generated inside each sandbox during an exercise. These logs are available for post-exercise forensic review, enabling trainees to reconstruct attack timelines and evaluate response actions.
- **MITRE ATT&CK mapping:** The platform's MITRE Technique service tags training tasks against ATT&CK techniques, providing a structured view of which adversary behaviours are covered by the completed training activities.

Tool Diagram CyberRangeCZ Training Platform

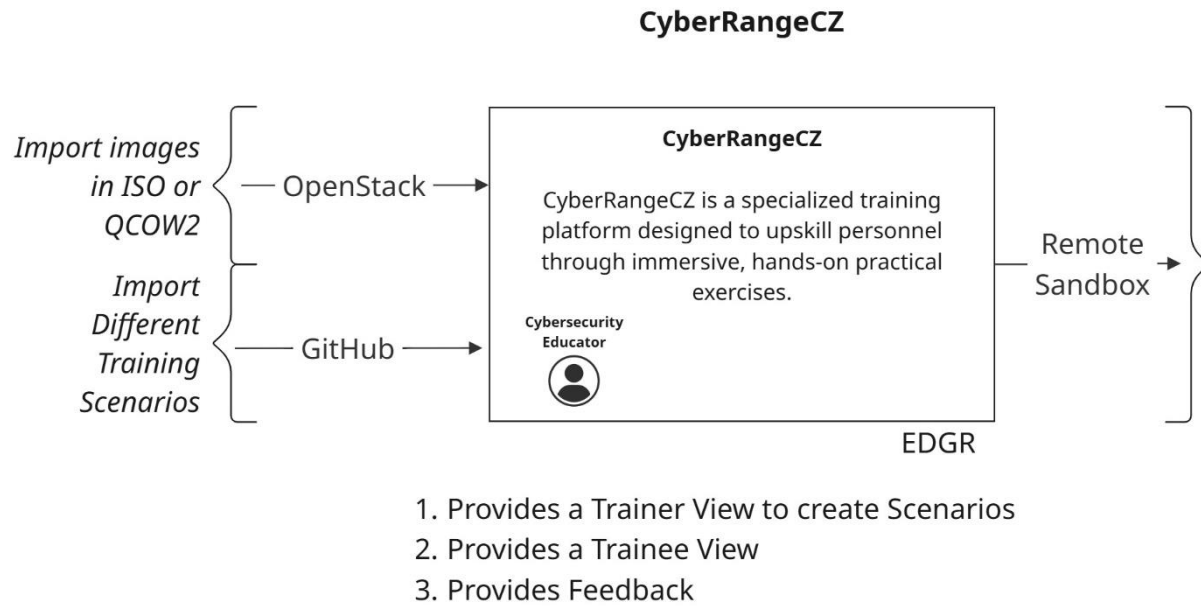


Figure 3 CyberRangeCZ (KYPO) Training Platform Data Flow

3.4 User Roles and Access Control

The CyberRangeCZ Platform implements a role-based access control (RBAC) model based on user roles and permissions. In the NG-SOC deployment, a local Keycloak instance serves as the identity provider, managing user identities and issuing OpenID Connect authentication tokens to the platform. The platform then enforces access control based on the role claims included in these tokens.

The platform defines the following principal roles:

- **Administrator:** Full access to all platform functions, including user management, sandbox pool configuration, and system settings.
- **Trainer (Organizer):** Creates and manages training definitions, provisions sandbox pools, monitors trainee progress in real time, and reviews post-exercise reports.
- **Trainee:** Accesses assigned training runs, interacts with the sandbox environment through a web-based console, submits flags, and views personal scores and feedback.
- **Designer:** Authors and maintains sandbox definitions (topology files, Ansible playbooks, Packer templates) in the connected Git repositories.

Role assignment is managed in Keycloak and propagated to the platform through OIDC tokens. Each API request is validated against the role claims included in the bearer token, ensuring that trainees cannot access trainer functions and that sandbox management endpoints are restricted to authorised personnel.

4 Technical Architecture

4.1 Internal Architecture

The CyberRangeCZ Platform follows a microservices architecture deployed on Kubernetes. Each service runs as an independent container, communicates with other platform components primarily through REST APIs, and can be scaled, updated or replaced with limited impact on the rest of the system. At a conceptual level, the system is organised into three layers:

- **CyberRangeCZ Portal:** The front-end application layer. Built in Angular 18, it exposes the web interface through which users authenticate, access training definitions, manage sandboxes, monitor trainee progress and interact with training exercises. The Portal communicates with the platform backend through REST API clients and mediates access to the microservices that provide the main platform functions for trainees, trainers and administrators.
- **Backend Microservices:** A set of containerized services that implement the platform's core business logic. Key services include the Sandbox Service, which manages the lifecycle management of emulated environments, the Linear Training and Adaptive Training services which orchestrate exercise and scoring, Answers Storage, Training Feedback, User and Group management, and MITRE Technique mapping. Adaptive training functionality is further supported by a Smart Assistant component, which provides decision logic for adjusting training paths based on trainee progress and exercise outcomes.
- **Cloud Infrastructure (OpenStack):** The workload infrastructure layer that provisions the actual compute, storage, and networking resources for sandbox environments. The Sandbox Service interacts with the configured cloud backend to create isolated virtual networks, routers, and virtual machines for each exercise session. As of v2025.03 [4], CyberRangeCZ supports both OpenStack and AWS as cloud backends. The NG-SOC deployment, however, uses an OpenStack-based private cloud hosted at EDGR premises.

Each sandbox is created as an isolated environment with its own virtual network and set of virtual machines. This isolation separates training environments from one another and prevents interference between concurrent exercise sessions. Platform-level services, such as user management, training definitions, training results, feedback, and logs, are maintained separately from ephemeral sandbox resources.

4.2 Technology Stack

The platform is built entirely on open-source components. No proprietary runtime licences are required for the core development, and the principal components can be inspected, forked, and self-hosted. Table 2 summarizes the main technologies used in each layer.

Table 2 Platform Technology Stack by Layer

Layer	Technology	Role
Front-end	Angular 18, TypeScript	Portal UI and single-page application framework
Backend services	Java / Spring Boot	Microservice business logic and REST APIs implementation

Container orchestration	Kubernetes, Helm	Service deployment, scaling, and lifecycle management
Infrastructure-as-Code	Terraform (OpenTofu), Ansible	Cloud resource provisioning and VM image configuration
Cloud provider	OpenStack	Compute, networking, and storage for sandbox VMs
Virtualization	KVM / QEMU	Hypervisor layer for full-fledged virtual machines
VM image building	Packer	Automated creation of sandbox VM images
Identity provider	Keycloak (OIDC)	Authentication and single sign-on
Source control	Git, GitHub (from v2025.03)	Sandbox definition versioning and collaboration

4.3 Deployment

A fresh CyberRangeCZ installation starts from the single Helm chart repository (devops-helm) and the Terraform configuration (devops-tf-deployment). Helm packages every microservice into versioned charts, while Terraform (via OpenTofu) compatible configurations are used to provision the underlying cloud resources including networks, security groups, base images, and the Kubernetes node pool itself. The platform services run on a Kubernetes cluster inside the selected cloud provider environment (OpenStack). A clean deployment can be torn down and rebuilt from the infrastructure configuration without manual reconfiguration of individual services.

For the NG-SOC project, the platform runs on a dedicated server hosted at EDGR premises, provisioned as specified in D2.4 - NG-SOC Architecture [6]: 64 CPU cores, 1 TB RAM, 4 TB storage, and 4 network interfaces. This server hosts the OpenStack private cloud on which CyberRangeCZ (KYPO) operates.

The minimum hardware footprint for a single evaluation instance is 2 CPU cores, 16 GB RAM, and 40 GB storage. This configuration provides a baseline environment for installation, testing, and validation. Larger training events and production-grade exercises require capacity to scale according to the number of concurrent sandboxes, the number of virtual machines per sandbox, VM image sizes, allocated memory, storage use, networking requirements, and the resource profile of each exercise. The maximum usable capacity is governed through OpenStack tenant quotas.

Scalability is achieved at two levels. At the platform level, additional Kubernetes worker nodes can be added to host more backend service replicas. At the infrastructure level, the OpenStack tenant quotas control how many concurrent sandbox VMs the infrastructure can sustain. Each sandbox is an isolated virtual network with its own set of VMs. Destroying a sandbox after a session releases the associated compute, storage and network resources.

High availability is not built into the default open-source deployment. For the NG-SOC pilots, resilience is supported through infrastructure-level operational measures, including regular VM snapshots, backups, and recovery procedures. Where the underlying OpenStack deployment provides native high-availability capabilities, such as shared storage or live migration, these may further improve operational resilience. However, such capabilities depend on the actual physical and logical topology of the deployment. A full high-availability topology with replicated Kubernetes control planes, clustered databases and replicated storage is possible but exceeds the current project scope.

4.4 Performance

Platform performance is governed primarily by the cloud provider's capacity and the complexity of the sandbox definitions. The main factors affecting performance are the number of virtual machines per sandbox, the size and type of VM images, the memory and vCPU allocation per VM, the number of concurrent sandbox requests, and the duration of configuration tasks executed through Ansible.

The key metrics are:

- **Sandbox provisioning time:** Creating a sandbox involves provisioning virtual machines, configuring networks, and running Ansible playbooks. For smaller or typical sandbox definitions, sandbox creation usually takes between 5 and 15 minutes, depending on the number of virtual machines, image sizes, and configuration complexity. Larger exercises may require longer, especially when several sandboxes are requested at the same time. The Sandbox Service queues requests and processes them in parallel up to the concurrency limits of the OpenStack environment.
- **Concurrent sandbox capacity:** The maximum number of simultaneous sandboxes is determined by the cloud tenant's vCPU, RAM, storage, networking, and floating-IP quotas. A typical exercise sandbox for one trainee team comprises 5 to 10 virtual machines. A training event with 20 teams may therefore require capacity for approximately 100 to 200 virtual machines running concurrently. The actual capacity depends on the resource profile of each VM, and the quotas assigned to the relevant OpenStack tenant.
- **Portal response time:** The Angular front-end and backend REST APIs are expected to respond within standard web-application latency under normal load. Page loads and API calls should typically remain within sub-second to low-second response times, depending on network conditions, backend load, and database performance. During sandbox provisioning peaks, the main performance load is placed on the cloud infrastructure and orchestration components rather than on the Portal itself.
- **Resource reclamation:** Sandboxes are destroyed after each session, releasing the allocated compute, storage, and network resources. No persistent sandbox VM state is retained after a session unless it is explicitly preserved for analysis or debugging. Platform-level data, including users, training definitions, answers, feedback, logs, configuration data, and base VM images, remains stored in the relevant platform services and storage systems.

A concrete benchmark was recorded during the BSides Bratislava event CTF event held from the 8th of May until the 16th of May and hosted on the NG-SOC CyberRangeCZ (KYPO) instance. The exercise deployed 15 concurrent sandboxes, each containing 9 to 10 virtual machines with varying specifications, including different operating-system images, memory allocations, and installed toolsets. The OpenStack cluster required approximately 490 GB of RAM to sustain all sandboxes simultaneously, with roughly 140 virtual machines running in parallel. No performance degradation was observed on the Portal or backend services during the event.

End-to-end provisioning of all 15 sandboxes, including VM boot, network wiring, and Ansible playbook execution, completed within a 2-to-4-hour operational window. This figure refers to the bulk provisioning of all sandboxes for the event and should therefore be distinguished from the typical provisioning time reported for smaller or individual sandbox deployments. The total duration may vary depending on sandbox complexity, image sizes, infrastructure scheduling, quota limits, and post-boot configuration tasks.

5 Interfaces and Integration

5.1 Internal NG-SOC Interface

The CyberRangeCZ (KYPO) Platform serves as the main hands-on training environment for NG-SOC. It hosts, co-deploys, or coordinates with seven NG-SOC components, depending on the technical requirements of each training scenario. Some components are instantiated directly inside sandbox environments, some provide data or artefacts to exercises, and others are coordinated at the instructional level rather than through direct system integration. Each integration supports a distinct training objective.

NG-SOAR: NG-SOAR instances are deployed inside the CyberRangeCZ sandboxes so that trainees can create, test, and execute CACAO playbooks without affecting production systems. This supports Req-006, sub-requirement 6, which states that NG-SOAR components must be fully operational in KYPO for training, emulation, and testing purposes. Playbook exchange via STIX 2.1 Course of Action objects is supported through the co-deployed OpenCTI instance.

NG-SIEM: Replicated NG-SIEM instances run inside each sandbox, and ingest logs generated by the scenario's virtual machines. Trainees use these instances to practice log analysis, alert triage, and correlation rule authoring against live data produced within the exercise environment. This supports Req-022, which requires a secure virtualized infrastructure for hands-on training exercises.

OpenEdX: The OpenEdX learning management system hosts the theoretical training component, including presentations, video lectures, quizzes, and reading material produced under T5.3 and delivered through D5.3 - Open Cybersecurity Training and Exercise Scenarios. CyberRangeCZ (KYPO) and OpenEdX are independent platforms. They maintain separate user databases, authentication flows, and training records, and there is no automated data exchange between them. Trainees access each platform separately, while trainers coordinate the sequencing of theoretical modules and hands-on labs through the training design. At the M30 milestone, this coupling is realised at the instructional and process level only and there is no automated, system-level integration between the two platforms.

CTI-SS: The Cyber Threat Intelligence Sharing System provides threat context for sandbox scenarios, including indicators of compromise, adversary tactics, techniques, and procedures, and STIX/TAXII-formatted intelligence. This allows exercises to include realistic and up-to-date threat data instead of relying only on static datasets. Depending on the scenario design, this intelligence can be used to seed detection tasks, enrich investigation workflows, or support trainee decision-making during incident-response exercises..

PenTest: The semi-automated penetration testing platform described in D3.2 is deployed inside CyberRangeCZ (KYPO) sandboxes in training-oriented configurations. It runs against emulated targets within the sandbox so that trainees can perform reconnaissance, web-application scanning, vulnerability assessment, and assisted exploitation in a controlled environment. OpenVAS and a sandbox-local Neo4j instance are hosted inside each Kali VM, which helps prevent concurrent trainees from interfering with one another or with central NG-SOC infrastructure. The platform can also provide engagement artefacts, such as Nmap scan results, exploitation attempts, and CALDERA campaign traces, as JSON outputs for use in repeatable detection-and-response scenarios.

DRME: The Dynamic Risk Management Engine is delivered as a containerized stack (Docker / Docker Compose), and can be instantiated within isolated CyberRangeCZ alongside other NG-SOC components. Within a training scenario, trainees interact primarily with the DRAS analyst dashboard to ingest vulnerability findings, penetration-testing output, and threat-intelligence reports and to map the resulting CVEs and threats in the exercise environment. The Risk Modeling Tool (RMT) then computes the corresponding risk posture and supports threat prioritisation based on projected business and technical impact. The engine consumes raw technical findings (scan results, penetration-testing data, CTI) together with asset inventories, and produces prioritized risk scores, residual-risk evaluations, and mitigation recommendations that trainees act upon during the exercise. This reproduces the operational pilot workflows specified in D4.2 - e.g., automated vulnerability assessment and prioritization (Sub Case 1a) and live incident impact and residual-risk evaluation (Sub Cases 1b/3b) - enabling hands-on practice of dynamic risk-management procedures in a safe, repeatable environment.

CICMS: A dedicated instance of the Collaborative Incident Case Management System described in D4.5 is deployed into the CyberRangeCZ training sandboxes to support collaborative incident-handling exercises (Pilot 2, Subcase 2c - Malware Scenario). During an exercise, trainees use CICMS as a shared case-management workspace. For example, one trainee may examine forensic artifacts while another queries the CTI Sharing System for matching IoCs and MITRE ATT&CK techniques. Actions, findings and decisions are recorded on the case timeline. The completed case can then be reused as a replayable record for post-exercise debriefing and as supporting training material. This usage supports the CICMS multi-organization collaboration model defined under Task 4.5, allowing trainees from different teams to share findings through the case interface while respecting the access scope assigned in the exercise.

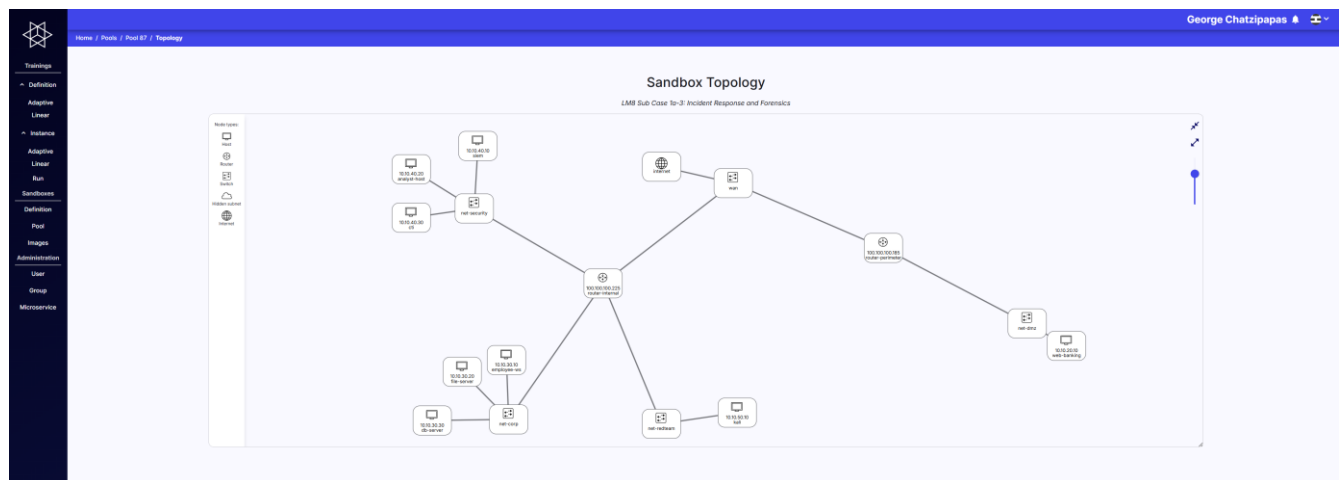


Figure 4 Topology of a Scenario

5.2 External Interfaces

Beyond the NG-SOC component integrations described above, the CyberRangeCZ relies on a limited number of external interfaces for deployment, sandbox lifecycle management, source-code access and authentication.

Cloud Provider APIs: The Sandbox Service interacts with the configured cloud provider during sandbox lifecycle operations, including sandbox creation, monitoring, and destruction. In the NG-SOC deployment, this interface is provided by OpenStack services such as Nova, Neutron, Glance, and Heat. These APIs are used to provision

and manage the Compute, networking, image, and orchestration resources required by sandbox environments. Although CyberRangeCZ supports additional cloud backends, the NG-SOC instance uses OpenStack.

Git Repositories: Sandbox definitions, including YAML topology files, Ansible playbooks, and Packer templates are stored in Git repositories. As of v2025.03, both GitLab and GitHub repositories are supported. The platform retrieves the relevant definitions during sandbox creation. No write-back to the source repositories is performed as part of the standard sandbox lifecycle workflow.

Identity Provider (OIDC): The platform delegates authentication to an external OpenID Connect provider. In the NG-SOC deployment, this role is fulfilled by Keycloak. Since the authentication mechanism is based on OpenID Connect, the platform can be configured to integrate with other OIDC-compliant identity providers, enabling federation with institutional identity-management systems where required.

An important architectural boundary is that the platform itself does not directly integrate with external log sources, external CTI platforms, or third-party security controls at the platform level. Threat intelligence, log feeds, and security-control interactions are introduced inside individual sandboxes as part of the scenario content. This keeps the functional external interface of the core platform limited to cloud-provider APIs, Git repositories, and the configured identity provider.

6 Security Considerations

6.1 Authentication and Authorization

A platform designed to host offensive exercises must enforce strict access boundaries - a trainee who can reach another team's sandbox, or worse the management plane, defeats the purpose of isolation. The CyberRangeCZ Platform delegates authentication to an external OIDC identity provider. In the NG-SOC deployment, this role is fulfilled by a Keycloak instance acting as the central authentication broker. Security is enforced through a combination of identity management, token-based API access, role-based authorization, and sandbox-level network isolation.

- **Single Sign-On (SSO):** Users authenticate once through Keycloak and receive a JSON Web Token (JWT) that grants access to all platform microservices without re-authentication. Keycloak can support federation with institutional identity providers (SAML, LDAP, or OIDC-based systems), allowing pilot organizations to reuse their existing credentials.
- **API Security:** REST API endpoints exposed by the backend microservices require a valid JWT bearer token. Tokens are issued by Keycloak with a configurable expiry and contain role claims that the backend services enforce.
- **Role Enforcement:** The four-tier RBAC model described in Section 3.4 (Administrator, Trainer, Trainee, Designer) is implemented through Keycloak realm roles. Each API call checks the caller's role against the required permission for the requested resource. Role assignments are managed centrally in Keycloak and propagated to the platform via token claims.
- **Sandbox Isolation:** Each sandbox runs on its own isolated virtual network within the cloud provider. No network path exists between sandboxes belonging to different training sessions, and sandbox networks have no outbound route to the internet or the host infrastructure's management network. This isolation is enforced at the cloud provider level (OpenStack security groups and network namespaces).

6.2 Data Protection

Data handled by the platform falls into two distinct categories with different lifecycles and risk profiles. The first category is platform operational data such as user accounts, training definitions, assessment results, logs and configuration data which persists across sessions. The second category is sandbox ephemeral data such as VM disks, network traffic generated during exercises, and temporary files, which is intended to exist only for the duration of a training session unless explicitly preserved for analysis, replay or assessment purposes.

Encryption in Transit: Communication between the users' browser and the Portal is protected through TLS. Access to sandbox VM consoles is provided through secure web-based connections. Internal communication between platform services should be protected according to the deployment configuration, for example through cluster-level network controls, TLS termination, or service-to-service encryption where enabled.

Encryption at Rest: Protection of data at rest depends on the storage layer used by the deployment. In the NG-SOC OpenStack environment, Cinder volumes can be configured with LUKS-based encryption where required. Training definitions stored in Git repositories are protected through repository-level access controls. Platform

databases and persistent storage should be protected through the applicable backup, access-control, and encryption mechanisms of the deployment environment.

Data Retention: Sandbox environments are ephemeral by design. Virtual machines, disks, and network resources are destroyed when a training session ends. Sandbox VM state is therefore not retained after cleanup unless specific artefacts are intentionally exported or preserved for debugging, replay, assessment, or post-exercise analysis. Platform-level data, including assessment scores, training progress, feedback, and audit-relevant metadata, is retained for the duration required by the training programme and can be purged according to the applicable data-retention policy.

Logging: Platform events and user actions are logged internally within the CyberRangeCZ platform stack hosted on the EDGR OpenStack infrastructure. These logs record operational and audit metadata, such as who started a sandbox, when it was started, which training definition was used, and what scoring outcomes were produced. Logs are not intended to capture the full content of sandbox virtual machines, although scenario-specific artefacts or selected outputs may be collected where required for scoring, replay, or post-exercise debriefing.

Compliance: The platform is designed to minimise the amount of personal data it processes. In normal operation, this is limited primarily to identity attributes received from the OIDC provider, such as user display names, email addresses, and role information. The platform itself is not intended to ingest sensitive pilot data, such as financial records or energy-grid telemetry. Where such data is required for a scenario, it should be represented through synthetic, anonymised, or sandbox-contained datasets and handled within the isolated exercise environment. This architecture supports data-minimization principles consistent with GDPR.

7 Limitations and Future Enhancements

No deployment is without constraints. This section summarizes the known limitations of the current CyberRangeCZ v2025.03 deployment and identifies the corresponding mitigations or planned enhancements for the remaining project period.

7.1 Current Limitations

Default open-source deployment: The NG-SOC deployment uses the public CyberRangeCZ platform without project-specific modifications to the core platform. This approach ensures maintainability, reproducibility and alignment with the upstream release cycle. However, functionality that is not available in the open-source distribution must either be contributed upstream, developed as a local extension or addressed through scenario-level design.

OpenEdX integration: CyberRangeCZ and OpenEdX operate as separate platforms with independent user management and training records. At the M30 milestone, trainees access each system individually, while trainers coordinate the sequencing of theoretical modules and hands-on labs through instructional design rather than automated system-level linking.

Sandbox provisioning latency: Creating complex sandbox topologies, especially those involving 10 or more VMs and multiple network segments, can take 10–15 minutes or longer depending on image size, configuration complexity, and infrastructure load. For large-scale training events with many concurrent teams, this startup delay is managed through advance scheduling, staggered launches, and, where appropriate, pre-provisioning of sandbox environments .

High availability: The default Helm deployment runs a single replica of each microservice. A node or pod failure interrupts the platform until Kubernetes reschedules the workload. This is acceptable for scheduled training events, which are supervised.

Scenario authoring complexity: Building a new training scenario requires familiarity with infrastructure-as-code practices, including YAML topology definitions, Ansible playbooks, Packer templates, and Git-based version control. There is currently no graphical scenario editor. This can increase the entry barrier for trainers who are domain experts, such as SOC analysts or energy-sector engineers, but who are not experienced infrastructure-as-code practitioners. The limitation is mitigated through reusable scenario templates, documentation, and collaboration between domain experts and technical scenario authors.

7.2 Planned Enhancements

Adaptive training improvements: The Smart Assistant and adaptive training logic will be further extended to support more granular training paths based on trainee performance, exercise outcomes, and coverage of relevant MITRE ATT&CK techniques.

Scenario library growth: Under T5.3 and D5.3, additional sector-specific training scenarios will be developed for the financial, energy, and academic pilots. These scenarios will expand the reusable training catalogue and provide pilot-specific hands-on exercises aligned with the NG-SOC use cases.

Operational benchmarking: Additional performance measurements will be collected during pilot activities, including sandbox provisioning times, concurrent sandbox capacity, resource consumption, and recovery procedures. These measurements will support more accurate sizing guidance for future deployments.

Improved integration guidance: Further documentation will be prepared to support the integration of NG-SOC components into CyberRangeCZ (KYPO) sandboxes. This will include reusable deployment patterns, configuration guidance, and lessons learned from pilot exercises.

8 Conclusion

Task 5.2 sets out to deliver a scalable, open-source platform for hands-on cybersecurity training. This objective has been achieved through the deployment of a CyberRangeCZ (KYPO) instance based on the open source v2025.03 distribution. That platform was selected for its microservices architecture, cloud-native deployment model, open-source availability, and active community stewardship. It has been deployed on the project's OpenStack infrastructure managed by EDGR and validated under real exercise conditions, including a BSides CTF event that sustained 15 concurrent sandboxes, approximately 140 virtual machines, and around 490 GB of RAM usage.

The principal accomplishments against the Grant Agreement scope are:

- **Platform deployment:** A fully operational CyberRangeCZ instance has been deployed on the project's OpenStack infrastructure, managed by EDGR. The platform provisions isolated sandbox environments in which trainees work with full-fledged operating systems and realistic network topologies, while keeping training activities separated from production systems.
- **NG-SOC component integration:** NG-SOAR and NG-SIEM instances run inside CyberRange sandboxes, enabling trainees to practise playbook authoring, log analysis, alert triage and incident response workflows using tools aligned with the NG-SOC operational stack. This supports the requirements captured in D2.3 [5], including Req-006 and Req-022.
- **Training modalities:** The platform supports linear and adaptive training paths, Capture-the-Flag exercises, automated scoring, trainee monitoring and post-exercise feedback. Adaptive training capabilities, including the Smart Assistant module, support dynamic training paths based on trainee performance and exercise outcomes.
- **Open-source alignment:** By adopting the public CyberRangeCZ release without project-specific forks to the core platform, the NG-SOC deployment benefits from upstream maintainability and future improvements, including the AWS backend support and GitHub integration introduced in v2025.03 [4]. This also supports the project's sustainability objective of delivering an open-source platform for European cybersecurity capacity building.

Mapped against the five WP5 objectives in the Grant Agreement, the platform contributes directly to the delivery of an open-source virtual learning environment (Objective 2), supports competence assessment through automated scoring and trainee analytics (Objective 3), enables hands-on exercises and monitoring through the Trainer dashboard and Training Feedback service (Objective 4), and runs on lightweight container-based Kubernetes infrastructure (Objective 5). The skills development strategy (Objective 1) addressed in D5.1 [7] provides input to scenario design and training-path development.

One area remains open for the final project phase. The sector-specific training scenarios produced under T5.3 and D5.3 will populate the platform with exercises tailored to the financial, energy, and academic pilots. Together with the operational platform, these scenarios complete the capacity-building toolchain that WP5 is designed to deliver and provide input to the WP6 integration testing and the broader NG-SOC validation programme.

Appendix A – Technical Detail

The CyberRangeCZ (KYPO) Platform is documented through the upstream open-source project. The following resources provide technical details beyond the scope of this deliverable:

- **Platform documentation** (<https://docs.platform.cyberrange.cz/>):
The documentation covers platform concepts, installation guidance, OpenStack requirements, training-definition authoring, sandbox management and user workflows.
- **Source code** (<https://github.com/CyberRangeCZ>):
The CyberRangeCZ GitHub organisation contains more than 100 repositories covering the frontend, backend services, Helm charts, Terraform/OpenTofu deployment configurations, Ansible roles, VM image builders, and supporting libraries. Core repositories are published as open-source software under the MIT licence [2].
- **Deployment entry point** ([https://github.com/CyberRangeCZ \(KYPO\)/devops-helm](https://github.com/CyberRangeCZ (KYPO)/devops-helm)):
The devops-helm repository provides the Helm charts used to deploy the platform services to a Kubernetes cluster.
- **Release notes for v2025.3** (<https://www.cyberrange.cz/insights/2025-03/>):
The release notes [4] describe the AWS backend support, GitHub integration for sandbox definitions, Angular 18 migration, and user-interface improvements included in the version deployed for NG-SOC.

The following screenshots are taken from the NG-SOC CyberRangeCZ (KYPO) deployment and illustrate the platform's user-facing interfaces:

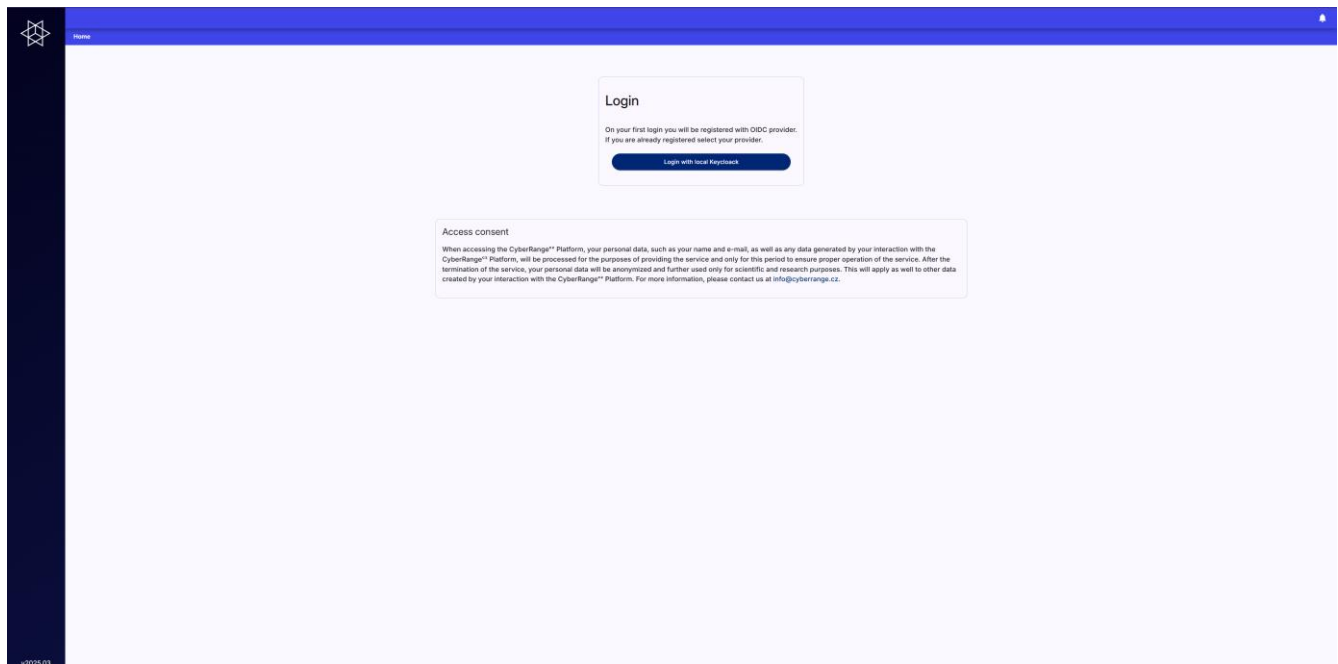


Figure 5 Welcome Screen

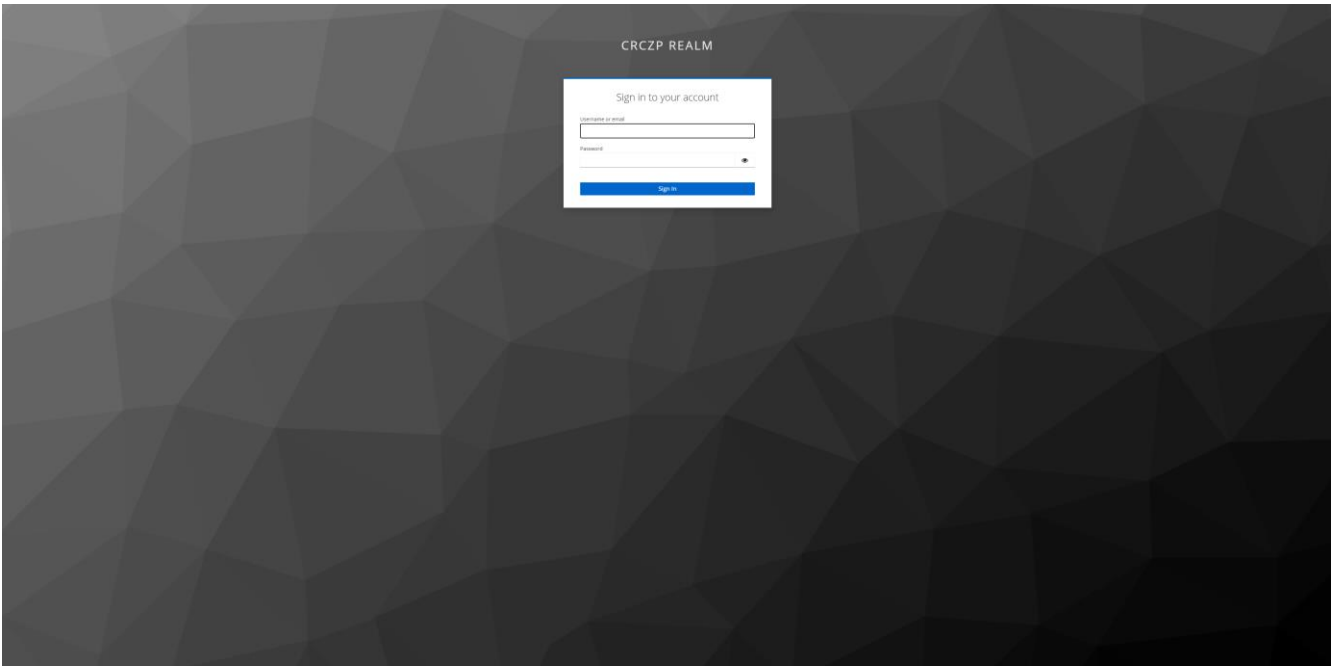


Figure 6 Login Screen

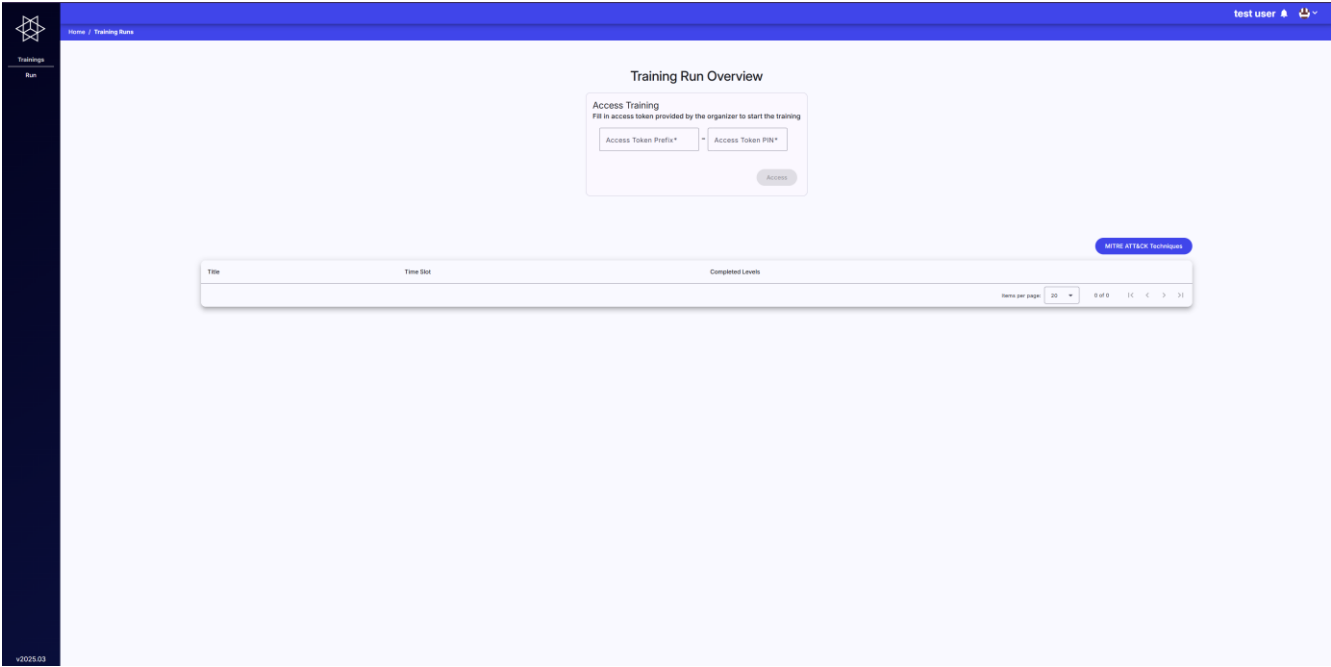


Figure 7 Access Code Entry

Appendix B – Traceability Matrix

The table below maps the D2.3 user and system requirements [5] relevant to the Hands-on Educational Platform to the features implemented or addressed in this deliverable.

Table 3 Traceability Matrix

Req. ID	Requirement Name	Implemented Feature in D5.2
UReq-12	Modern Cybersecurity Training	CyberRangeCZ (KYPO) provides hands-on training through educational material and exercises. Linear and adaptive training paths, CTF exercises, and automated scoring are supported. Scenario definitions are versioned in Git to support updates and reuse.
UReq-32	Accessible Documentation	Training material and platform documentation are accessible through the CyberRangeCZ (KYPO) Portal and the upstream documentation. Theoretical courses, presentations, and quizzes are hosted on OpenEdX as a standalone platform, as described in Section 5.1
Req-006	NG-SOAR in CyberRangeCZ (KYPO)	NG-SOAR instances are deployed inside CyberRange sandboxes for training, emulation, and testing. Supports CACAO playbook execution and playbook exchange through STIX 2.1 Course of Action objects, as described in Section 5.1
Req-022	Secure Virtualized Environment	Isolated OpenStack sandbox environments provide the infrastructure for hands-on training exercises. Each sandbox runs on a separate virtual network with no connectivity to production systems (Sections 4.1, 6.1).
Req-026	Provide Theoretical Training	OpenEdX hosts theoretical training content (presentations, videos, quizzes). Trainees access theoretical and hands-on training through separate entry points (Section 5.1).
UReq-10	Attack Surface Mapping	Sandbox topologies model realistic attack surfaces. Trainees map and analyse these environments during exercises using the deployed NG-SOC tools.
UReq-33	Friendly and Intuitive User Interface	The CyberRangeCZ (KYPO) Portal provides web-based access to platform functions, including training access, sandbox management, trainee monitoring, and exercise feedback. The v2025.03 release includes a refreshed user interface.

Appendix C – Asset Specification

Table 4 CyberRangeCZ Asset Specification

Asset Overview		
Asset(s) name(s):	CyberRangeCZ Hands-On Training Platform	
Created by:	ED	
Document version:	1.0	
Asset Description	The continuous training, education and awareness module aims to train, educate and up-skill the involved within security domain, personnel. The module comprises a platform that provides self-paced training courses and a Cyber range platform for hands-on training scenarios without the risk of compromising systems in production.	
Asset Requirements		
Hardware requirements:	Single Server 16+ Cores 128+ GB RAM 1+ TB Storage 2+ Network Interfaces	
Locality	ED Premises	
Distributed environment	Private Cloud	
Type and size of input data	The OpenStack can import images of different OS (Linux distros, windows) and different types such as iso and qcow2. The CyberRange can import various OS images to be used for different training scenarios	
Data processing	On-demand	
Type and size of output data	Type: GUI OpenStack configuration presented in the GUI Training scenarios definition and execution results presented in the KYPO GUI	
Interfacing (I/O)	Input: SSH connection from the user	Output: Remote user access to the training sandbox
Interactions	OpenStack provides the platform for the KYPO to run and execute the training scenarios	
Software dependencies/libraries required:	OpenStack, OpenStack client, python, docker, ansible, pipenv, kubectl, helm, terraform, Kubernetes,	
Software license:	Open source.	
User Interaction:	Both the OpenStack and CyberRangeCZ provide a user interface	
Additional Information		
Relevant blocks from conceptual architecture	Continuous training, education, and awareness	

Functionalities for MVP (M30)	A trainer will construct and execute hands-on scenarios The trainee will be able to execute these scenarios and enhance their Cybersecurity related knowledge. Feedback will be available both to the trainer and to the trainee.
Notes and Issues:	None.

REFERENCES

- [1] CyberRangeCZ Platform Documentation. CyberSecurity Hub. Retrieved from <https://docs.platform.cyberrange.cz/>
- [2] CyberRangeCZ GitHub Repository. CyberSecurity Hub. Retrieved from <https://github.com/CyberRangeCZ>
- [3] Vykopal, J., Ošlejšek, R., Čeleda, P., Vizvary, M., and Tovarňák, D. (2017). " KYPO Cyber Range: Design and Use Cases". In *Proceedings of the 12th International Conference on Software Technologies (ICSOFT 2017)*, Volume 1, pp. 310–321. DOI: <https://doi.org/10.5220/0006428203100321>
- [4] CyberRangeCZ Platform v2025.03 Release Notes (2025). "AWS support and GitHub integration in CyberRengeCZ Platform 2025.03". Retrieved from <https://www.cyberrange.cz/insights/2025-03/>
- [5] NG-SOC Project Consortium (2024). *D2.3 User and System Requirements for Secure Digital Infrastructures, v1.0*.
- [6] NG-SOC Project Consortium (2024). *D2.4 NG-SOC Architecture, v1.0*.
- [7] NG-SOC Project Consortium (2024). *D5.1 Skills Development Strategy Report*. Retrieved from: https://ng-soc.eu/article/NG-SOC_D5.1_20v1.0.pdf
- [8] ENISA (2022). *European Cybersecurity Skills Framework (ECSF)*. European Union Agency for Cybersecurity. Retrieved from <https://www.enisa.europa.eu/publications/european-cybersecurity-skills-framework-ecsf>



NGSOC

Next Generation Security Operations Centres



Co-funded by
the European Union



ECCC
EUROPEAN CYBERSECURITY
COMPETENCE CENTRE

This project has received funding from the European Union's Digital Europe Programme (DIGITAL) under grant agreement No 101145874

