A distributed data-mining software platform for extreme data across the compute continuum

D4.4 EXTRACT Platform Compute Continuum Validation

Version 1.0

Documentation Information

Contract Number	101093110
Project Website	www.extract-project.eu
Contractual Deadline	M39, 31 March 2026
Dissemination Level	Public
Nature	Other
Author	IBM Research, Israel (IBM)
Contributors	BINARE, IKERLAN, BSC
Reviewer	LRI
DOI	10.5281/zenodo.19208127
Keywords	EXTRACT Software Platform, Compute Continuum, Data

Change Log

Version	Description Change
V0.1	Initial draft, monitoring overview (IKERLAN)
V0.2	Organization, IBM contributions (IBM)
V0.3	BINARE Contributions
V0.4	BSC Contributions
V0.5	Internal Review
V1.0	Ready to submit

The EXTRACT Project has received funding from the European Union's Horizon Europe programme under grant agreement number 101093110.

Table of Contents

1. Introduction

The final period (up to M36/M39) of the EXTRACT project deals with validating the products of the project, namely putting the entire software stack to functional tests and stress-tests under various deployment conditions. At the basic level, these tests make sure the software operates as designed, especially in the context of the use-cases (i.e., PER use-case and TASKA use-cases). Beyond that, they benchmark the software for properties such as performance and security to validate EXTRACT KPIs as declared in the EXTRACT action documents.

The validation work in this document is focused on Work Package 4 (WP4) – the core compute continuum layer. This is the bottom-most layer of the software stack, providing the foundation on top which the upper layers are deployed.

1.1. Purpose and scope of the deliverable

This deliverable presents the final validation of the EXTRACT compute continuum, as designed and implemented in WP4. It consolidates the outcomes of Task 4.4 – Compute Continuum Validation (M31–M36/M39) and supports the overall evaluation activities of T1.4, focusing on the assessment of Objective O3 and its associated KPIs (KPI3.1, KPI3.2). Building on the requirements and evolving design from D4.1, D4.2 and D4.3.

D4.4 therefore has a multiple purpose:

- To report validation activities executed in the context of the compute continuum layer and any impact on the software stack implementation in that layer.
- To provide quantitative and qualitative evidence that the compute continuum abstraction layer, its orchestration mechanisms, and the supporting data backbone meet the performance, scalability and robustness expectations of the project as defined in O3 and in particular KPI 3.1.
- To demonstrate that the Cybersecurity mechanisms implemented across the continuum provide a sufficient level of protection for data, workflows and ML models, in line with the EXTRACT security requirements and with the intended level of assurance for Objective O3 and in particular KPI 3.2.

1.2. Structure of the document

This document is organised as following. Section 1 is an introduction, providing an overview of the purpose and scope of the compute continuum validation. Section 2 is a recap of the compute continuum layer and main components, and describes main deployments. Section 3 discusses functional validation of components – is a component behaving as expected, and possible code changes/fixes resulting from tests. Section 4 deals with KPI validation. First, it recalls Objective O3 and associated KPIs. Then each KPI has a dedicated subsection, detailing how it was evaluated and main results. Section 5 is summary and conclusions. First, O3 and KPI assessment conclusions, and then lessons learned. Finally, Section 6 provides the standard support data – cited references, acronyms and abbreviations.

2. Recap: EXTRACT Compute Continuum Layer

This section provides a concise overview of the EXTRACT Compute Continuum Layer. It is based on previous documents D4.1, D4.2 and D4.3.

2.1. Multi-Premise Architecture

As detailed in previous documents D4.1, D4.2 and D4.3, EXTRACT tackles the challenge of distributing computation across the compute continuum from edge to cloud. This requires addressing deployment elasticity, security, data access and propagation, and interoperability across multiple premises (edge, HPC, cloud), each potentially consisting of several independent sites.

The chosen architecture defines each EXTRACT instance as a connected, non-empty set of Compute Continuum Sites (CCS), shown in Figure 1. A CCS is essentially a Kubernetes [1] cluster running scalable and universal EXTRACT services, providing a consistent execution environment across all premises. Each EXTRACT instance includes several CCSs, one per premise, with the specific services deployed at each site depending on application needs.

All CCSs within an EXTRACT instance are connected, enabling EXTRACT applications to be automatically managed across premises, with global consistent data access. This is delivered by a cross-CCS backbone consisting of distributed data and orchestration functions. The data backbone is implemented with SkyStore (IBM). The distributed orchestration layer uses COMPSs (BSC). In addition, there are per-CCS common facilities of security (BINARE/BNR) and monitoring (IKERLAN). All these main components are overviewed below.

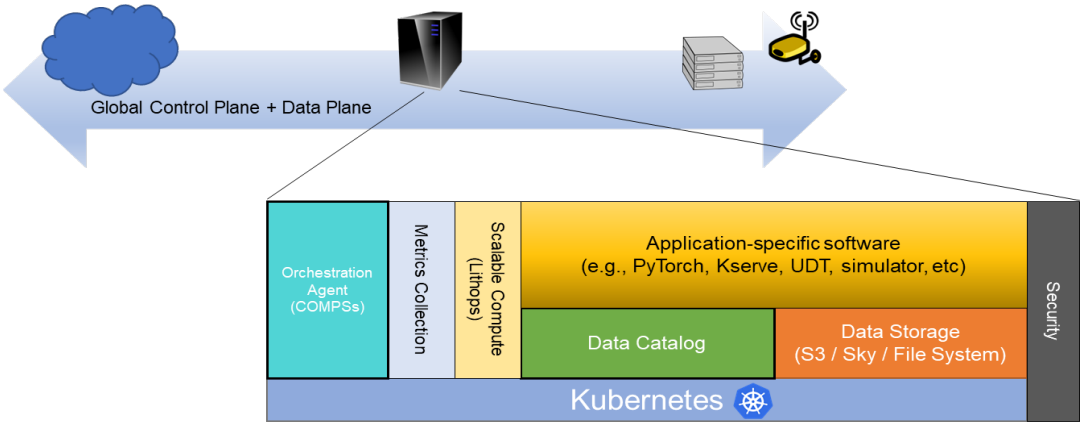


Figure 1. EXTRACT Multi-Premise Architecture

2.2. SkyStore Data Backbone

EXTRACT relies on S3-compatible object storage as the common data substrate, providing a unified interface for storing and accessing data across the platform components. In light of that, EXTRACT data backbone uses SkyStore [2] for uniform, efficient and seamless access to S3 data that is distributed across different concrete S3 clusters. As Figure 2 below shows, clients connect to S3 storage via SkyStore S3 proxy, which further relays the S3 access to any of multiple S3 stores - some on public cloud(s), and some on private premises.

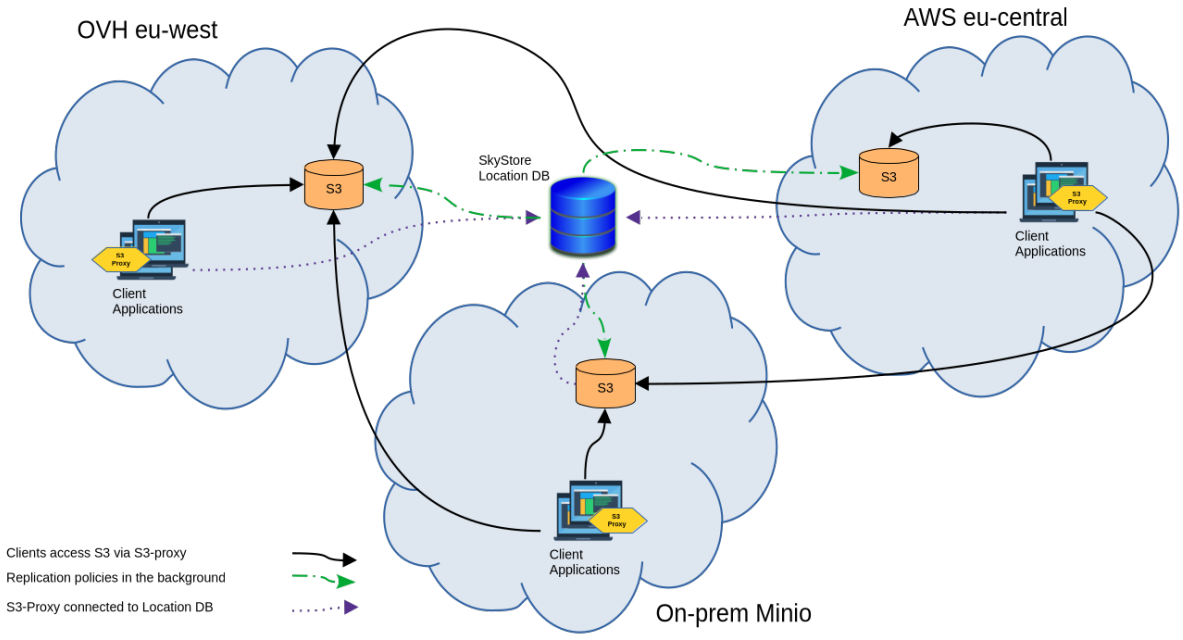


Figure 2. SkyStore Multi-Premise S3

Access efficiency (depending on application requirements) can be obtained by configuring data access and replication policies that are enforced at each S3 proxy transparently to client requests. One important aspect is locality – performance differences between remote S3 stores and local S3 stores (in terms of network topology) can be significant, due to network distance, connectivity issues, bandwidth limitations, etc. SkyStore replication policies allow pulling objects to local S3 store or pushing it to remote S3 stores, as warmup or during repeated access, to mitigate those performance differences. The policies are applied without modifying the global virtual S3 layout of buckets and objects provided by SkyStore, by coordinating operations through its central location DB service.

2.3. COMPSs Orchestration Backbone

The orchestration backbone of the EXTRACT infrastructure is powered by COMP Superscalar (COMPSs), a task-based programming model designed to simplify the development and execution of applications across distributed infrastructures. COMPSs has been adapted to be executed in both, Kubernetes single cluster and multicluster configurations.

Figure 3 below illustrates the resource mapping in a Kubernetes multicluster deployment. In this example, the master and one worker run in the local cluster, while two additional workers operate in the remote cluster. Each component is deployed as a Deployment with an associated Service, all within the same Namespace. RBAC policies are applied to the master, and workers specify hardware resource requests. Services for COMPSs workers from the remote cluster are mirrored into the local cluster with the remote cluster's name appended as a suffix, enabling the COMPSs master to transparently communicate with all workers. Also, the master Service is mirrored in the remote cluster with the local cluster's name appended, so the remote workers can communicate with the master Pod.

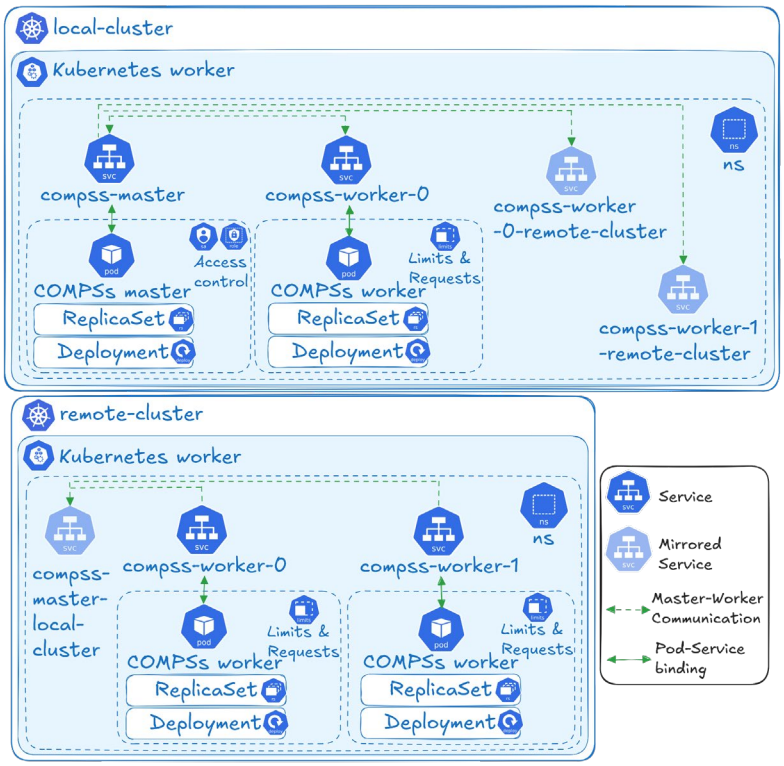


Figure 3. Graphic representation of a COMPSs deployment in Kubernetes multiclusters.

COMPSs allows users to select the tier (edge, cloud, HPC) in which tasks will execute, while the runtime handles parallelism, scheduling, or data dependencies abstracting the underlying infrastructure.

2.4. Security and Privacy

The need for cybersecurity cannot be understated for any modern system and design, and is of paramount importance. At the same time, it is important to understand that there is no such thing as a perfect cybersecurity, and that the cybersecurity is a continuous process rather than an end goal or a milestone. For example, a perfectly-secure system today may become insecure tomorrow even without a single change in it's design, implementation, or deployment – this is due to the fact that new vulnerabilities are always detected in third-party dependencies once believed to be flawless as well as because the tools and techniques used by attackers and ethical hackers always evolve, become better, and find new or more complex bugs (including security bugs).

To ensure security of the compute continuum and EXTRACT Platform, throughout the project we have employed both automated tools, specifically developed scripts, and manual validations/inspections on the following:

- Source code of individual repositories/modules
- Source configurations of individual repositories or deployments
- Docker images associated with individual repositories/modules
- Specific kubernetes deployments (whether OVH, own cloud, and localhost deployments)
- Communications between entities/pods/services within a deployment
- OS-level storage encryption
- TPM-level support
- Generic Security Advisories related to core technologies (docker, kubernetes)

Below we summarize a list of state-of-the-art and industry-standard tools used to verify various security parts mentioned above:

- Trivy
- Syft
- Grype
- Sonar Source (SonarQube Cloud)
- Docker Hub Enterprise Security
- Nmap / Nmap Unleashed
- BurpSuite
- sslscan, tlssled
- lsblk, blkid, cryptsetup
- Binare's FastCVE
- Binare's EXTRACT specific automation scripts/tools assembled into BISETO-CC toolkit

2.5. Monitoring Infrastructure

The EXTRACT monitoring infrastructure, developed by IKERLAN, provides a uniform way to observe the heterogeneous resources that make up the compute continuum, from constrained edge devices to powerful cloud, HPC and kubernetes nodes. Its main goals are to ensure that all nodes operate within their functional and non-functional requirements, and to expose up-to-date information about available compute and storage capacity, energy usage and application behaviour to both operators and the orchestration layer.

At the core of the monitoring stack is Prometheus, used as a time-series database and scraping engine. Each CCS runs a local Prometheus instance that periodically pulls metrics from exporters and instrumented services deployed on its nodes. The exporters expose standard host-level metrics (e.g., CPU load, memory usage, disk and network statistics), as well as accelerator-specific counters when available (e.g. NVIDIA GPU utilisation via dedicated exporters). Application-level metrics, traces and logs can be produced through OpenTelemetry or language-specific client libraries, allowing developers to report domain-specific indicators when needed. All samples are stored as labeled time series, where labels encode, among other things, the CCS, cluster, the node and the application identifiers.

The monitoring platform follows Prometheus's native pull model. Exporters and instrumented services run on each node (or as DaemonSets in Kubernetes), listening on local endpoints. The Prometheus server in the same CCS periodically scrapes these endpoints according to a configurable schedule, typically every few seconds for infrastructure metrics and at longer intervals for slower-changing indicators such as energy counters. This pull-based approach simplifies deployments behind NAT, allows each CCS to control its own scraping cadence, and avoids pushing monitoring traffic from resource-constrained devices unless strictly necessary.

To support the multi-cluster nature of the compute continuum, EXTRACT adopts a federated monitoring model. Each CCS keeps a complete, independent view of its local metrics, while a higher-level Prometheus instance – deployed in the main cloud CCS – aggregates a subset of those metrics from all sites. This is achieved through Prometheus federation and carefully crafted scrape configurations: the global instance periodically fetches pre-aggregated series (for example, per-node and per-service summaries) from each CCS, re-labeling them with cluster identifiers. In this way, the project obtains a consolidated view of the continuum without centralizing all raw samples or overloading the WAN links between premises, in line with the "database federation" approach originally envisioned for the monitoring platform.

Multi-cluster integration relies on three key principles:

- **Federation and Orchestration**: A federated control plane based on Karmada (or any Kubernetes-supported tool) manages policy propagation and deployment across clusters, ensuring consistent installation of Prometheus, Grafana, and exporters in all CCSs. Propagation policies guarantee synchronized updates and configurations without manual intervention, reducing operational complexity.
- **Secure Inter-Cluster Connectivity**: Mechanisms such as Submariner, Skupper, or Linkerd establish secure tunnels between clusters, enabling service exposure even in NATed or isolated environments. Service Export/Import and ClusterLink policies guarantee visibility of federated metrics.
- **Consistent Monitoring Deployment**: Each CCS runs its own Prometheus and Grafana instances for local observability, while the global Prometheus aggregates selected metrics for continuum-wide dashboards.

The monitoring data serves two main consumers. First, the workflow deployment and scheduling mechanisms — especially COMPSs — use it to take informed decisions. A dedicated RESTful API, developed by Ikerlan, exposes selected metrics and derived indicators (such as available CPU capacity or memory pressure in each CCS) in a format suitable for the orchestrator. COMPSs queries this API when planning or adapting executions, enabling placement and scaling strategies that are informed by the real-time state of the continuum rather than static assumptions. Second, the same time-series data is visualised through Grafana dashboards tailored to EXTRACT. Per-CCS and global dashboards provide infrastructure operators with an at-a-glance view of node health, resource utilisation, energy profiles and application-level metrics. This makes it possible to detect abnormal behaviour or capacity issues quickly, correlate them with use-case executions (PER, TASKA-C/D) and, when needed, adjust deployments or repeat experiments during the validation campaigns.

In the context of this deliverable, the monitoring infrastructure is therefore both part of the validation environment and an enabler for the validation itself: it instruments all CCSs in a consistent way, provides a multi-cluster view of the compute continuum, and exposes the metrics required to analyse performance, scalability and resource-efficiency results. The monitoring platform allows hierarchical federation and, with some additional implementations, can enable cross-service federation, ensuring scalability and flexibility across diverse deployment scenarios.

3. Functional Validation

This section describes functional validation activities for the main components of the Compute Continuum Layer, and their respective results.

3.1. SkyStore Functional Validation

As can be expected, functional validation of SkyStore was in place already during the earlier development and integration phases. However, it focused on synthetic scenarios (not necessarily use-case-related). In the last phase, testing SkyStore in use-case application and stress-testing it under different configurations yielded several useful lessons that contributed to improving it.

As of now, with the changes below applied, all functional tests pass, and SkyStore is fully operational, supporting all S3 options of the compute continuum – major clouds, public endpoint and private endpoints.

3.1.1. Test Scope and Procedure

SkyStore provides a standard S3 interface to clients through S3-proxy and a configuration interface to administrators (configuration file). As such, our functional validation revolved around the following S3 functions:

1. Create bucket – creates a named bucket to store objects in. Objects in S3 can only be stored in buckets.
2. Put (upload) object – create an object in a specific bucket by providing a key (label) for the object and body (content, typically a file being uploaded)
3. Get (download) object – retrieve the content of an object with a specific key in a specific bucket, and download it into a local file. Typically we compare the content of the downloaded object with the file it was uploaded from to verify no data corruption.

We also checked other functions such as delete object and delete bucket, but those are of lesser importance for the use-cases.

From a configuration perspective, there are two interesting and relevant dimensions of testing. First, there is the type of S3 region to add to SkyStore. The original SkyStore prototype was supposed to cover major clouds (AWS [3], GCP [4] and Azure [5]) as well as public S3 providers (e.g., OVH [6]). During the use-case tests it turned out that it would do well to support also private S3 (e.g., local Minio [7], or local Ceph [8]). However, tests proved otherwise, and lead to changes as reported further below.

A second configuration dimension of testing that is especially relevant for SkyStore performance, is the replication policy for each S3-proxy. The default policy is "write_local" which means all put object operations are done at the S3 region defined as local, and get object is done from the S3 region the object is stored at. A second policy "copy_on_read" performs background copy of remote objects to local S3 region upon read, and is used to prove the value of locality in performance as discussed above (warmup read or repeated read). Functional testing of the second policy failed in several cases and resulted in additional fixes, as reported below. There is also a beginning of support for a policy "push" for advancing copies of new objects to remote regions in the background, but it will be left for future work, as we benchmark locality through "copy_on_read".

The tests have been conducted through a combination of CLI and API (boto3) clients, which can be found in the "eval" folder of the SkyStore repository [2]. These clients perform the S3 operations described above, and are also used to benchmark performance (see D2.4). A test failure is simply a failure in a client execution during one of the operations. After identifying the failed functional tests, the fixes described below have been applied, and the tests were run again, all passing.

3.1.2. Public S3 Endpoint Support

The SkyStore prototype officially supported only major cloud providers – AWS, GCP and Azure. However, it was possible to leverage the flexible configuration of the AWS S3 client to introduce overrides in order to support any public S3 endpoints. One could override the actual S3 endpoint, the credentials etc. While possible, it didn't always work, and this became painfully apparent when we tried to use OVH (chosen cloud provider for EXTRACT demos) as an S3 region in SkyStore during PER use-case integration testing.

For this reason, we decided to make SkyStore explicitly support all public S3 endpoints. SkyStore supports AWS, GCP and Azure through "provider" abstraction: *aws:*, *gcp:* and *azure:*, respectively. For example, to support the AWS S3 region of eu-west-1 (Dublin, Ireland), we refer to it as *aws:eu-west-1* in SkyStore configuration. It uses AWS credentials, which SkyStore has separate specific support for. Support for GCP and Azure is based on similar concepts.

In order to support any S3 provider with a public endpoint, we created a 4th provider *custom:* . Unlike the existing providers, each "custom" endpoint requires explicit full configuration which is provided explicitly to the boto3 client, not as "override" for another endpoint. This configuration is specified in a new section "custom_endpoints" in the S3-proxy configuration. Figure 3 below is an example for configuring OVH Gravelines S3 as a "custom endpoint" using this new configuration format.

```
{
  "init_regions": [
    "aws:eu-central-1",
    "custom:ovh-eu-west",
  ],
  "client_from_region": "custom:ovh-eu-west",
  "skystore_bucket_prefix": "skystore-extract",
  "policy": "write_local",
  "server_addr": "127.0.0.1",
  "custom_endpoints": {
    "custom:ovh-eu-west": {
      "endpoint_url": "https://s3.gra.cloud.ovh.net",
      "aws_access_key_id": "<OVH S3 access key>",
      "aws_secret_access_key": "<OVH secrect access key>",
      "region": "gra"
    }
  }
}
```

Figure 4: SkyStore S3-proxy Configuration with "custom" S3 endpoint in OVH

We can now use the official S3 configuration of each endpoint as-is in SkyStore, as-is. The code change has been applied and tested in SkyStore and is now working.

3.1.3. Private S3 Endçpoint Support

As discussed above, another issue that has been realized in SkyStore during use-case integration tests is the need to support S3 regions whose endpoint is private, and therefore accessible only from the S3-proxy in the same CCS. This can occur in different scenarios. An edge cluster (e.g, a 5G MEC) may choose to deploy a small S3 locally to cache data before sending it to a remote data center, or cache data that is remote but is often needed. Such a service is intended only to serve the scope of the cluster, not the general public. Another possible scenario is that of a multi-site company or even a "virtual enterprise" [9] that spans one or more private data centers, each with its internal S3 services, as well as possibly some dedicated cloud S3, and need to provide global, consistent and efficient access to S3 data.

Unlike the previous bug-fix, this is more of a design flaw in SkyStore that was fixed. SkyStore design relies on each S3-proxy being able to directly access all the underlying S3 services. This holds true for custom public S3 endpoints, but fails for private ones. We set up a test with one local Minio in one cluster (using the "custom" extension above) and one in another cluster using AWS region. The test failed for second S3-proxy to access objects that were written to the Minio.

To fix this issue, we leveraged the SSH tunnel that connects each S3-proxy to the central SkyStore Location Data service, for port-forwarding. One S3-proxy in the same CCS as the private S3 region forwards the S3 endpoint service to the location data service host at a specific port. Every S3-proxy in other CCS-s forwards the service from data service host to itself using the same port. This allows SkyStore to operate correctly, with each client being able to reach all the underlying S3 services.

The solution implementation turned out to be quite elegant, as a separate new component called "tunneler" sets up all the forwarding outside the core S3-proxy code, and only reconfigures the S3-proxy, replacing unreachable remote IP addresses of other S3 regions with *localhost:port* where the port is the forwarded port for that service.

The only additional complexity is required in case of private DNS, so that the private S3 service is known by DNS name, not IP address, and a wrapper is required to run S3-proxy with DNS addresses redirected to localhost.

Figure 4 below shows an example new configuration with port forwarding, used to add a private local Minio deployment to SkyStore. The new configuration is based on the "custom:" extension (clearly, private S3 endpoints are not part of global cloud providers). A new field called *local_port_fwd* can be added to a custom endpoint and requires that it is forwarded over the tunnel across the central location data to S3-proxies in other CCS, using the specified port.

```
{
  "init_regions": [
    "aws:eu-central-1",
    "custom:minio-local"
  ],
  "client_from_region": "custom:ovh-eu-west",
  "skystore_bucket_prefix": "skystore-extract",
  "policy": "write_local",
  "server_addr": "127.0.0.1",
  "custom_endpoints": {
    "custom:minio-local": {
      "endpoint_url": "http://10.3.86.146:9000",
      "aws_access_key_id": "minio",
      "aws_secret_access_key": "minio123",
      "region": "us-east-1",
      "local_port_fwd": 8012
    }
  }
}
```

Figure 4: SkyStore S3-proxy Configuration with private S3 endpoint forwarded

One important point about using private endpoints and performance. Since the S3-proxy is responsible for forwarding the S3 service through the SSH tunnel, it cannot be scaled in and out, because the forwarded service might break. However, there is an easy solution for that. In the same CCS, create two s3-proxies connected to the same local S3 service. One will be used to handle the service forwarding (has *local_port_fwd* in the configuration), and will not serve clients. The other will be used to serve clients without forwarding (no *local_port_fwd* in the configuration) and can thus be dynamically scaled.

With this fix implemented, together with the previous "custom" fix, SkyStore now has support for all types of S3 regions across the continuum.

3.1.4. Support for copy_on_read Policy

This is a bug-fix that was realized from functional testing. The copy_on_read policy in quite simple by design. When an S3-proxy configured with this policy needs to read an object that is not from the S3 region designated as the S3-proxy's "local" S3 region, then it should both read the object and return it to the client, and initiate a background copy of the object to write it to the local S3 region. Subsequent reads of the object will find (through the metadata service) that there is a local copy and will choose to read it instead of the remote one, resulting in improved performance.

In reality of tests, when using this policy, often either the client read would fail or the background copy would fail. The reason was the implementation that relied on splitting the stream of data being read between two endpoints (client filesystem and local S3) with bugs in synchronizing the rates and no buffering. Instead of trying to fix it, we replaced it with a simple dual-stream implementation, such that each endpoint has its own stream independent of the other. The fix has been successfully applied and tested.

3.2. COMPSs Functional Validation

COMPSs was originally designed for HPC and Grid infrastructures, where it has been widely used to orchestrate complex scientific workflows. However, despite its maturity in traditional HPC environments, there is not a standard way to deploy it in Kubernetes environments.

To fill this gap, we have developed *kubecompss*, a declarative CLI tool that automates the deployment of COMPSs workflows on Kubernetes single and multicluster configurations. Implemented using the Kubernetes Go client library, it accepts parameterized workflow descriptions and dynamically generates Kubernetes resources at deployment time.

3.2.1. Kubernetes single cluster

The Kubernetes resources deployed in single cluster configuration are:

- The COMPSs master runs as a Kubernetes Deployment with an associated Service.
- RBAC policies are applied to grant the master the necessary permissions for workflow management, including workers' graceful termination. A Role, ClusterRole, and ServiceAccount are created.
- Each COMPSs worker runs as an individual Kubernetes Deployment with its own Service. Worker Pods specify resource requests and limits for CPU, GPU, and memory, and mount persistent Volumes to store working directories, ensuring fault tolerance.

The modifications on the COMPSs runtime to allow Kubernetes execution are:

- Workers are registered with each worker's Kubernetes Service name.
- A script that handles COMPSs runtime start parameters and COMPSs master and worker graceful termination is specified as the Master Pod init command.

3.2.2. Kubernetes multiclusters

To enable deployment across the compute continuum, *kubecompss* supports multicluster deployments. This mode deploys COMPSs masters and workers across distinct Kubernetes contexts, leveraging Service Mirroring solutions such as Skupper and Linkerd for secure and transparent Service connectivity. Both solutions have been integrated into the *kubecompss* CLI to automate setup and management.

The modifications on the COMPSs runtime to enable Kubernetes multicluster execution are:

- Local workers are registered with each worker Kubernetes Service name, and remote workers are registered with the Mirrored Service name.
- Remote workers register the master mirrored Service name.

The following code snippet shows how the remote workers register the master with its mirrored Service name, depending if the *REMOTE_CLUSTER_NAME* environment variable exists.

```
String remoteClusterName = System.getenv("REMOTE_CLUSTER_NAME");
if (remoteClusterName != null && !remoteClusterName.isEmpty() &&
name.endsWith(remoteClusterName)) {
    String localClusterName = System.getenv("LOCAL_CLUSTER_NAME");
    if (localClusterName != null && !localClusterName.isEmpty()) {
        masterName = masterName + "-" + localClusterName;
    }
    else {
        LOGGER.debug("LOCAL_CLUSTER_NAME is not set. Using default master
name.");
    }
}
else {
    LOGGER.info("REMOTE_CLUSTER_NAME is not set or worker's name does not end
with it. Using default master name.");
}
```

The experimental environment where *kubecompss* has been tested consists of two independent on-premise Kubernetes clusters representing the HPC tier and the edge tier of the continuum. Both clusters run Kubernetes v1.34, use containerd as the container runtime interface, and rely on Flannel for intra-cluster networking. Multicluster connectivity is enabled through Linkerd edge-25.11.1, with the multicluster extension deployed in hierarchical mode[1]. The HPC-side cluster is registered in Linkerd with the name *hpc*, while the edge-side cluster is registered as *edge*. Under this configuration, Kubernetes Services exported from the HPC cluster are exposed in the edge cluster with the suffix *-hpc*, whereas Services exported from the edge cluster are exposed in the HPC cluster with the suffix *-edge*.

We have tested the PER use-case workflow, with the following configuration: the COMPSs master (*compss-master*) and one worker (*worker-0*) are deployed in the HPC cluster, and an additional worker (*worker-0*) in the edge cluster. Each master and worker is created as an independent Kubernetes Deployment, exposed via a Service for stable networking as required by the COMPSs runtime.

Multicluster connectivity is established using Linkerd's Service Mirroring. The Service for the edge worker is exported and mirrored into the HPC cluster as *worker-0-edge*; the COMPSs master Service is mirrored into the edge cluster as *compss-master-hpc*. This enables the master to transparently discover and register both local and remote workers, and the remote workers to communicate with the master.

[1] https://linkerd.io/2-edge/features/multicluster/\#hierarchical-networks

Figure 5 displays an actual execution trace of the PyCOMPSs workflow, including the training, aggregation, and inference tasks. The trace has been generated with the Extrae instrumentation library and visualized with Paraver performance analysis. Each trace encodes timestamped events corresponding to tasks, worker states, communication, and runtime operations. By representing these events on a timeline, Paraver enables the visual inspection of the execution. Concretely, the Figure shows the Paraver trace from an execution of the workflow with a maximum of two inference runs. On the Y axis, Executor 2 corresponds to the edge cluster worker (Worker Main 2.1.1), and Executor 3 corresponds to the HPC cluster worker (Worker Main 3.1.1). The workflow begins with the map generation task (blue, Executor 3.2.4), but it is not clearly seen because of its small execution time. It is followed by parallel training across all four HPC executors (white). The aggregation task also runs on the HPC cluster (red, Executor 3.2.4). Training and aggregation iterate through four cycles until the first inference task is triggered on the edge cluster (pink, Executor 2.2.1). Training continues alongside aggregation tasks, and after three more cycles, a second inference is triggered on the edge (pink, Executor 2.2.1).

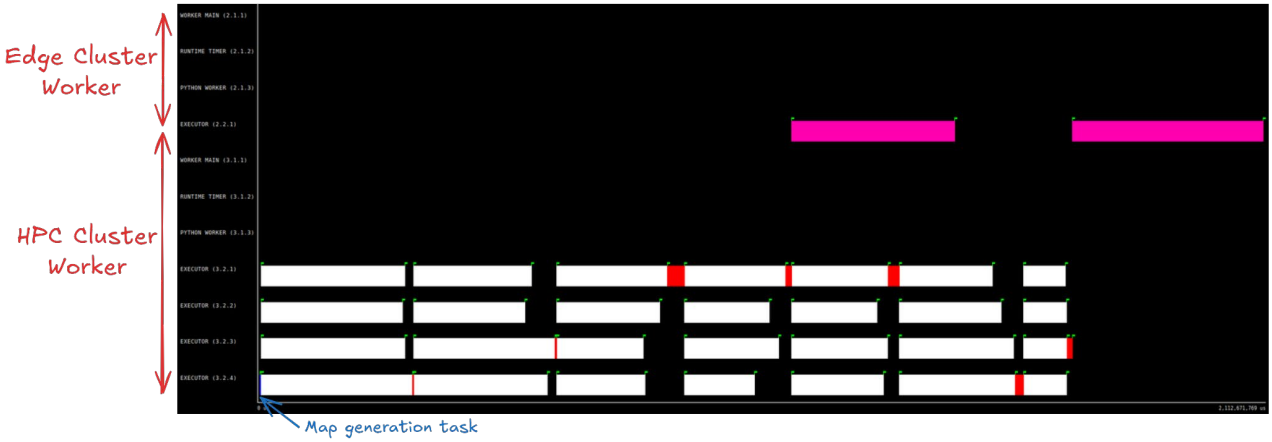


Figure 5. Graphic representation of a COMPSs deployment in Kubernetes multiclusters.

This figure demonstrates that *kubecompss* successfully deploys and orchestrates COMPSs workflows across multiple Kubernetes clusters, enabling coordinated execution between HPC and edge resources in a unified multi-cluster setup.

3.3. Cybersecurity Functional Validation

Below, we detail and summarize the main cybersecurity findings that were identified during the final validation iteration(s). The complete list of modules/repositories that went through the test scope and procedures along with their description is presented in more detail in deliverable D4.5.

It is worth noting again that security (similar to the quality assurance (QA)) is a continuous process and not a finishing frozen milestone. With this in mind, it is expected that at any release (even a project's final output) bugs (including cybersecurity ones) will always most likely be present, and even unavoidable. This however should not invalidate any of the major efforts poured into the overall R&D, integration, deployment, testing, and validation – quite on the contrary, this list of findings, evaluations and observations should serve as input in further security hardening efforts of the individual modules/repositories, and act as a guiding checklist for the organizations/users up-taking and exploiting the results (including EXTRACT partners using these as background IPR in future projects) of the EXTRACT Platform (whether as a whole, or just some of its parts).

3.3.1. Test Scope and Procedures

To ensure security of the compute continuum and EXTRACT Platform, throughout the project we have employed both automated tools, specifically developed scripts, and manual validations/inspections on the following approaches.

3.3.1.1. Secure code validation and scanning

- Source code of individual repositories/modules
- Source configurations of individual repositories or deployments
- Docker images associated with individual repositories/modules
- Specific kubernetes deployments (whether OVH, own cloud, and localhost deployments)

Ansible:

ansible Security Audit Report

Executive Summary

A static security audit of the **ansible** repository was performed on 2026-03-13. The repository contains Ansible playbooks for provisioning and configuring Kubernetes clusters (kubeadm-based) and a monitoring stack (Prometheus, Grafana, Kepler, NVIDIA exporter) on bare-metal or cloud nodes.

19 findings were identified across playbooks, role task files, configuration files, and inventory templates.

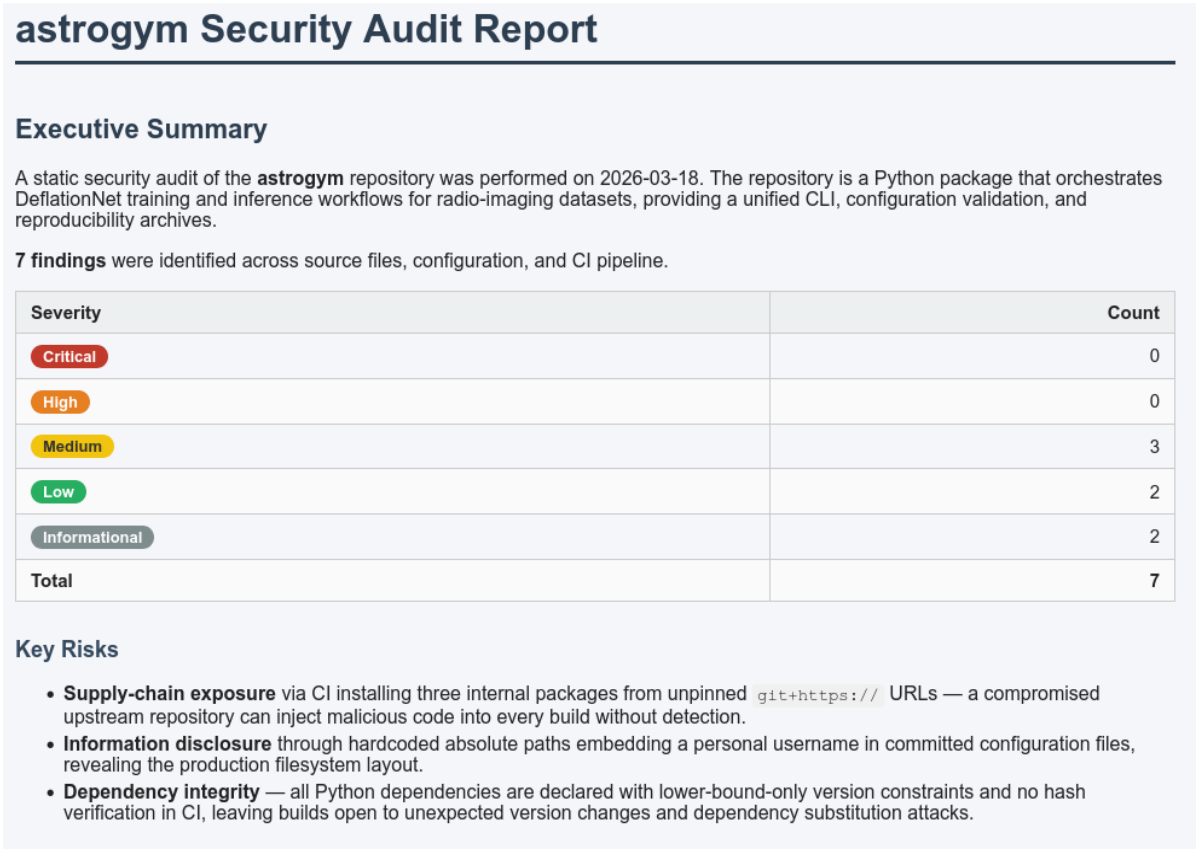
Severity	Count
Critical	4
High	8
Medium	4
Low	2
Informational	1
Total	**19**

Key Risks

- **Hardcoded credentials**: Grafana admin password committed in plaintext; GitLab credential file referenced in `.gitignore`.
- **Privilege escalation**: Join script written with 0777 permissions and executed via shell, allowing local users to inject commands before execution.
- **Kubernetes admin token exposure**: `admin.conf` fetched to world-readable `/tmp` and copied without restrictive permissions.
- **Supply-chain risks**: Flannel installed from `latest` GitHub release; Calico manifests and Helm binary downloaded without checksum verification.
- **Insecure TLS**: Metrics server configured with `--kubelet-insecure-tls`, disabling certificate validation.

No secrets hard-coded or leaked were found in the final repository. Source code security audit report: No results for SBOM (Software Bill of Materials) third-party/supply-chain security audit.

astrogym



astrogym Security Audit Report

Executive Summary

A static security audit of the **astrogym** repository was performed on 2026-03-18. The repository is a Python package that orchestrates DeflationNet training and inference workflows for radio-imaging datasets, providing a unified CLI, configuration validation, and reproducibility archives.

7 findings were identified across source files, configuration, and CI pipeline.

Severity	Count
Critical	0
High	0
Medium	3
Low	2
Informational	2
Total	**7**

Key Risks

- **Supply-chain exposure** via CI installing three internal packages from unpinned `git+https://` URLs — a compromised upstream repository can inject malicious code into every build without detection.
- **Information disclosure** through hardcoded absolute paths embedding a personal username in committed configuration files, revealing the production filesystem layout.
- **Dependency integrity** — all Python dependencies are declared with lower-bound-only version constraints and no hash verification in CI, leaving builds open to unexpected version changes and dependency substitution attacks.

No secrets hard-coded or leaked were found in the final repository. Source code security audit report: SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-18 15:03:15.268814154 +0200 EET m=+0.137926269

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
filelock	CVE-2025-68146	MEDIUM	3.18.0	3.20.1	https://access.redhat.com/security/cve/CVE-2025-68146 https://github.com/tox-dev/filelock https://github.com/tox-dev/filelock/commit/4724d7f8c3393ec1f048c93933e6e3e6ec321f0e Toggle more links
filelock	CVE-2026-22701	MEDIUM	3.18.0	3.20.3	https://access.redhat.com/security/cve/CVE-2026-22701 https://github.com/tox-dev/filelock https://github.com/tox-dev/filelock/commit/255ed068bc85d1ef406e50a135e1459170dd1bf0 Toggle more links
fonttools	CVE-2025-66034	MEDIUM	4.57.0	4.60.2	https://access.redhat.com/security/cve/CVE-2025-66034 https://github.com/fonttools/fonttools https://github.com/fonttools/fonttools/commit/a696d5ba93270d5954f98e7cab5ddca8a02c1e32 Toggle more links
h11	CVE-2025-43859	CRITICAL	0.14.0	0.16.0	https://access.redhat.com/security/cve/CVE-2025-43859 https://github.com/python-hyper/h11 https://github.com/python-hyper/h11/commit/114803a29ce50116dc47951c690ad4892b1a36ed Toggle more links
jupyter-core	CVE-2025-30167	HIGH	5.7.2	5.8.1	https://github.com/jupyter/jupyter_core https://github.com/jupyter/jupyter_core/commit/5e8965600adda6b416692ce7e85ecb2bd814bd52 https://github.com/jupyter/jupyter_core/security/advisories/GHSA-33p9-3p43-82vq Toggle more links
jupyterlab	CVE-2025-59842	LOW	4.4.0	4.4.8	https://access.redhat.com/security/cve/CVE-2025-59842 https://github.com/jupyterlab/jupyterlab https://github.com/jupyterlab/jupyterlab/commit/88ef373039a8cc09f27d3814382a512d9033675c Toggle more links
nbconvert	CVE-2025-53000	HIGH	7.16.6	7.17.0	https://access.redhat.com/security/cve/CVE-2025-53000 https://github.com/jupyter/nbconvert https://github.com/jupyter/nbconvert/blob/4f61702f5c7524d8a3c4ac0d5fc33a6ac2fa36a7/nbconvert/preprocessors/svg2pdf.py#L104 Toggle more links
pillow	CVE-2026-25990	HIGH	11.1.0	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
protobuf	CVE-2026-0994	HIGH	6.32.0	6.33.5, 5.29.6	https://access.redhat.com/errata/RHSA-2026:3095 https://access.redhat.com/security/cve/CVE-2026-0994 https://bugzilla.redhat.com/2432398 Toggle more links
requests	CVE-2024-47081	MEDIUM	2.32.3	2.32.4	http://seclists.org/fulldisclosure/2025/Jun/2 http://www.openwall.com/lists/oss-security/2025/06/03/11 http://www.openwall.com/lists/oss-security/2025/06/03/9 Toggle more links
setuptools	CVE-2025-47273	HIGH	78.1.0	78.1.1	https://access.redhat.com/errata/RHSA-2025:13578 https://access.redhat.com/security/cve/CVE-2025-47273 https://bugzilla.redhat.com/2366982 Toggle more links
tornado	CVE-2025-47287	HIGH	6.4.2	6.5	https://access.redhat.com/errata/RHSA-2025:8136 https://access.redhat.com/security/cve/CVE-2025-47287 https://bugzilla.redhat.com/2366703 Toggle more links
tornado	CVE-2026-31958	HIGH	6.4.2	6.5.5	https://access.redhat.com/security/cve/CVE-2026-31958 https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/119a195e290c43ad2d63a2cf012c29d43d6ed839 Toggle more links
tornado	GHSA-78cv-mqj4-43f7	MEDIUM	6.4.2	6.5.5	https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/24a2d96ea115f663b223887deb0060f13974c104 https://github.com/tornadoweb/tornado/releases/tag/v6.5.5 Toggle more links
urllib3	CVE-2025-66418	HIGH	2.4.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66418 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-66471	HIGH	2.4.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66471 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2026-21441	HIGH	2.4.0	2.6.3	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2026-21441 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-50181	MEDIUM	2.4.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50181 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/f05b1329126d5be6de501f9d1e3e36738bc08857 Toggle more links
urllib3	CVE-2025-50182	MEDIUM	2.4.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50182 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/7eb4a2aafe49a279c29b6d1f0ed0f42e9736194f Toggle more links
werkzeug	CVE-2025-66221	MEDIUM	3.1.3	3.1.4	https://access.redhat.com/security/cve/CVE-2025-66221 https://github.com/pallets/werkzeug https://github.com/pallets/werkzeug/commit/4b833376a45c323a189cd11d2362bcffdb1c0c13 Toggle more links
werkzeug	CVE-2026-21860	MEDIUM	3.1.3	3.1.5	https://github.com/pallets/werkzeug https://github.com/pallets/werkzeug/commit/7ae1d254e04a0c33e241ac1cca4783ce6c875ca3 https://github.com/pallets/werkzeug/security/advisories/GHSA-87hc-h4r5-73f7 Toggle more links
werkzeug	CVE-2026-27199	MEDIUM	3.1.3	3.1.6	https://github.com/pallets/werkzeug https://github.com/pallets/werkzeug/commit/f407712fdc60a09c2b3f4fe7db557703e5d9338d https://github.com/pallets/werkzeug/releases/tag/3.1.6 Toggle more links
No Misconfigurations found					

data-catalogue

Secrets hard-coded or leaked were found in the final repository. Source code security audit report.

data-catalogue Security Audit Report

Executive Summary

A static security audit of the **data-catalogue** repository was performed on 2026-03-13. The repository implements a data provenance and cataloguing library that tracks pipeline execution records in Nuvla (a cloud management platform) via MQTT messaging and direct API calls, and manages S3 object storage.

11 findings were identified across Python source files, configuration, and git history. Additionally, the gitleaks secret scanner detected a live API secret in the git commit history.

Severity	Count
Critical	2
High	4
Medium	4
Informational	1
Total	**11**

Key Risks

- **Leaked Nuvla secret in git history** — commit `04f08b6` contains a live API key/secret pair that must be rotated immediately.
- **API credentials printed to stdout** — `NUVLA_KEY` and `NUVLA_KEY_SECRET` are logged at every startup run.
- **Filter query injection** — multiple Nuvla search filters are constructed with unsanitized user input, enabling data access bypass.
- **MQTT transport in plaintext** — broker connection uses port 1883 without TLS.

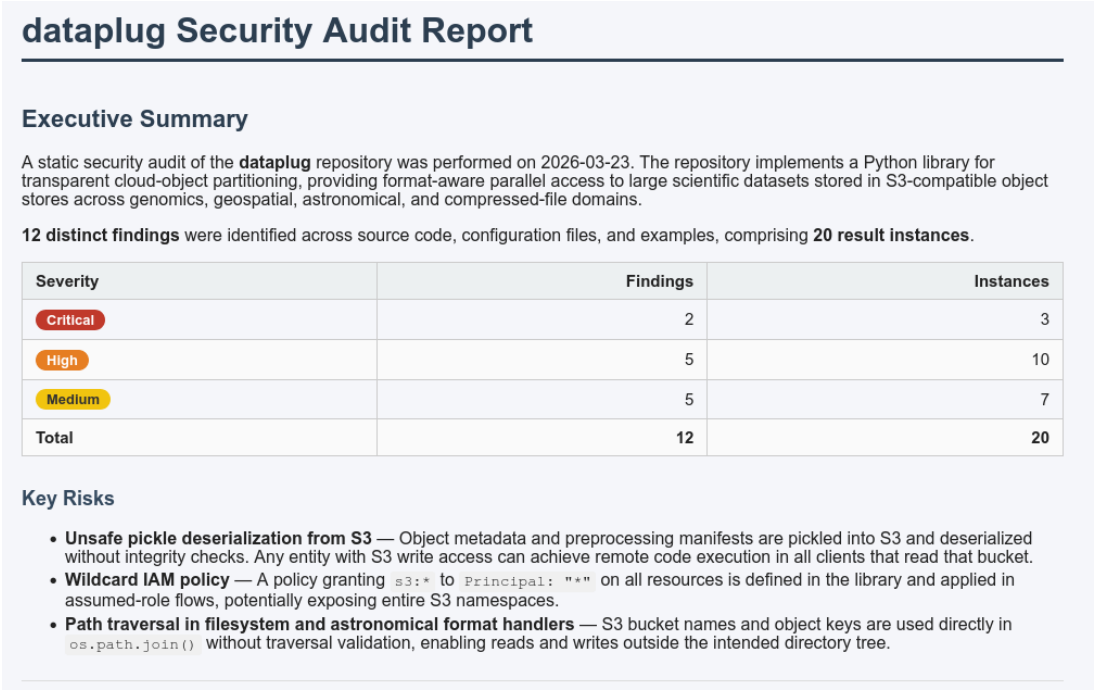
SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-13 14:41:57.502754113 +0200 EET m=+0.076997684

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
urllib3	CVE-2025-66418	HIGH	1.26.20	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66418 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-66471	HIGH	1.26.20	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66471 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2026-21441	HIGH	1.26.20	2.6.3	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2026-21441 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-50181	MEDIUM	1.26.20	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50181 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/f05b1329126d5be6de501f9d1e3e36738bc08857 Toggle more links
urllib3	CVE-2025-66418	HIGH	2.5.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66418 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-66471	HIGH	2.5.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66471 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2026-21441	HIGH	2.5.0	2.6.3	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2026-21441 https://bugzilla.redhat.com/2419455 Toggle more links
No Misconfigurations found					

dataplug

Secrets hard-coded or leaked were found in the final repository. Source code security audit report:



dataplug Security Audit Report

Executive Summary

A static security audit of the **dataplug** repository was performed on 2026-03-23. The repository implements a Python library for transparent cloud-object partitioning, providing format-aware parallel access to large scientific datasets stored in S3-compatible object stores across genomics, geospatial, astronomical, and compressed-file domains.

12 distinct findings were identified across source code, configuration files, and examples, comprising **20 result instances**.

Severity	Findings	Instances
Critical	2	3
High	5	10
Medium	5	7
Total	**12**	**20**

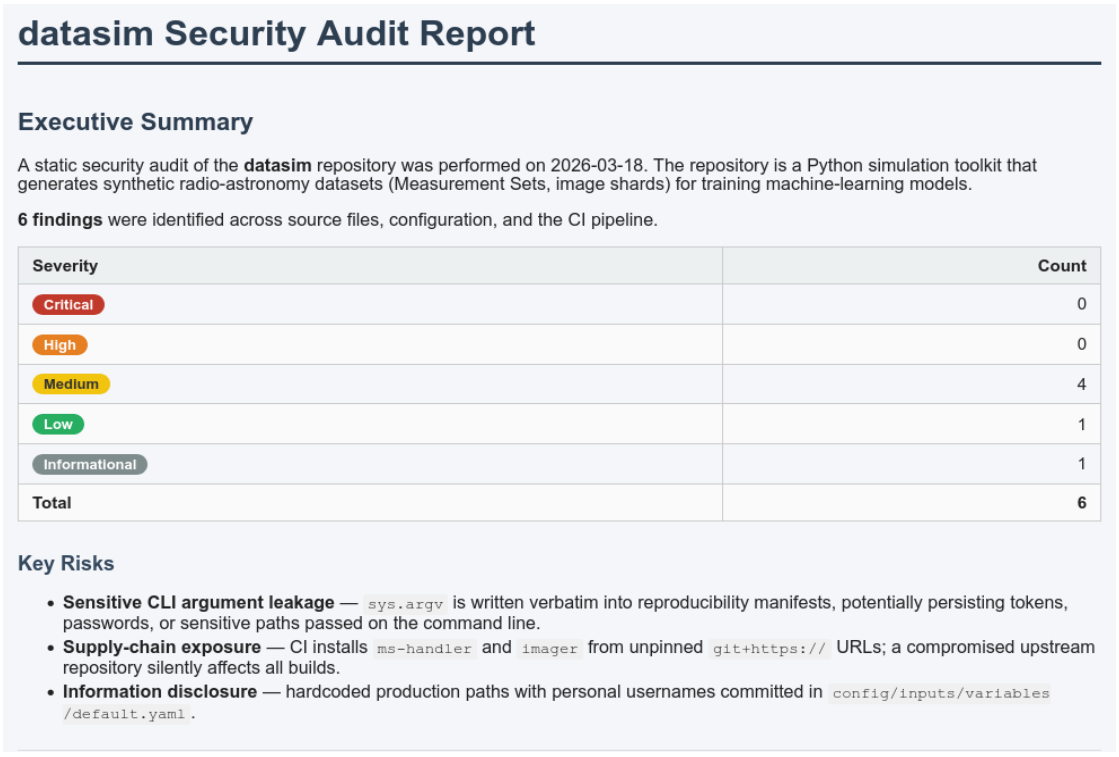
Key Risks

- **Unsafe pickle deserialization from S3** — Object metadata and preprocessing manifests are pickled into S3 and deserialized without integrity checks. Any entity with S3 write access can achieve remote code execution in all clients that read that bucket.
- **Wildcard IAM policy** — A policy granting `s3:*` to `Principal: "*"` on all resources is defined in the library and applied in assumed-role flows, potentially exposing entire S3 namespaces.
- **Path traversal in filesystem and astronomical format handlers** — S3 bucket names and object keys are used directly in `os.path.join()` without traversal validation, enabling reads and writes outside the intended directory tree.

No results for SBOM (Software Bill of Materials) third-party/supply-chain security audit:

datasim

No secrets hard-coded or leaked were found in the final repository. Source code security audit report:



datasim Security Audit Report

Executive Summary

A static security audit of the **datasim** repository was performed on 2026-03-18. The repository is a Python simulation toolkit that generates synthetic radio-astronomy datasets (Measurement Sets, image shards) for training machine-learning models.

6 findings were identified across source files, configuration, and the CI pipeline.

Severity	Count
Critical	0
High	0
Medium	4
Low	1
Informational	1
Total	**6**

Key Risks

- **Sensitive CLI argument leakage** — `sys.argv` is written verbatim into reproducibility manifests, potentially persisting tokens, passwords, or sensitive paths passed on the command line.
- **Supply-chain exposure** — CI installs `ms-handler` and `imager` from unpinned `git+https://` URLs; a compromised upstream repository silently affects all builds.
- **Information disclosure** — hardcoded production paths with personal usernames committed in `config/inputs/variables/default.yaml`.

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-18 15:03:18.957786419 +0200 EET m=+0.069376382

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
fonttools	CVE-2025-66034	MEDIUM	4.60.1	4.60.2	https://access.redhat.com/security/cve/CVE-2025-66034 https://github.com/fonttools/fonttools https://github.com/fonttools/fonttools/commit/a696d5ba93270d5954f98e7cab5ddca8a02c1e32 Toggle more links
nbconvert	CVE-2025-53000	HIGH	7.16.6	7.17.0	https://access.redhat.com/security/cve/CVE-2025-53000 https://github.com/jupyter/nbconvert https://github.com/jupyter/nbconvert/blob/4f61702ff5c7524d8a3c4ac0d5fc33a6ac2fa36a7/nbconvert/preprocessors/svg2pdf.py#L104 Toggle more links
pillow	CVE-2026-25990	HIGH	11.3.0	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
pillow	CVE-2026-25990	HIGH	12.0.0	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
tornado	CVE-2026-31958	HIGH	6.5.2	6.5.5	https://access.redhat.com/security/cve/CVE-2026-31958 https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/119a195e290c43ad2d63a2cf012c29d43d6ed839 Toggle more links
tornado	GHSA-78cv-mqj4-43f7	MEDIUM	6.5.2	6.5.5	https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/24a2d96ea115f663b223887deb0060f13974c104 https://github.com/tornadoweb/tornado/releases/tag/v6.5.5 Toggle more links
urllib3	CVE-2025-66418	HIGH	2.5.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66418 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-66471	HIGH	2.5.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66471 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2026-21441	HIGH	2.5.0	2.6.3	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2026-21441 https://bugzilla.redhat.com/2419455 Toggle more links
No Misconfigurations found					

flexecutor
Secrets hard-coded or leaked were found in the final repository.

Critical Findings

CRITICAL SA-CWE-312-001 CWE-312

AWS Credentials Exposed in S3 Configuration Dictionary

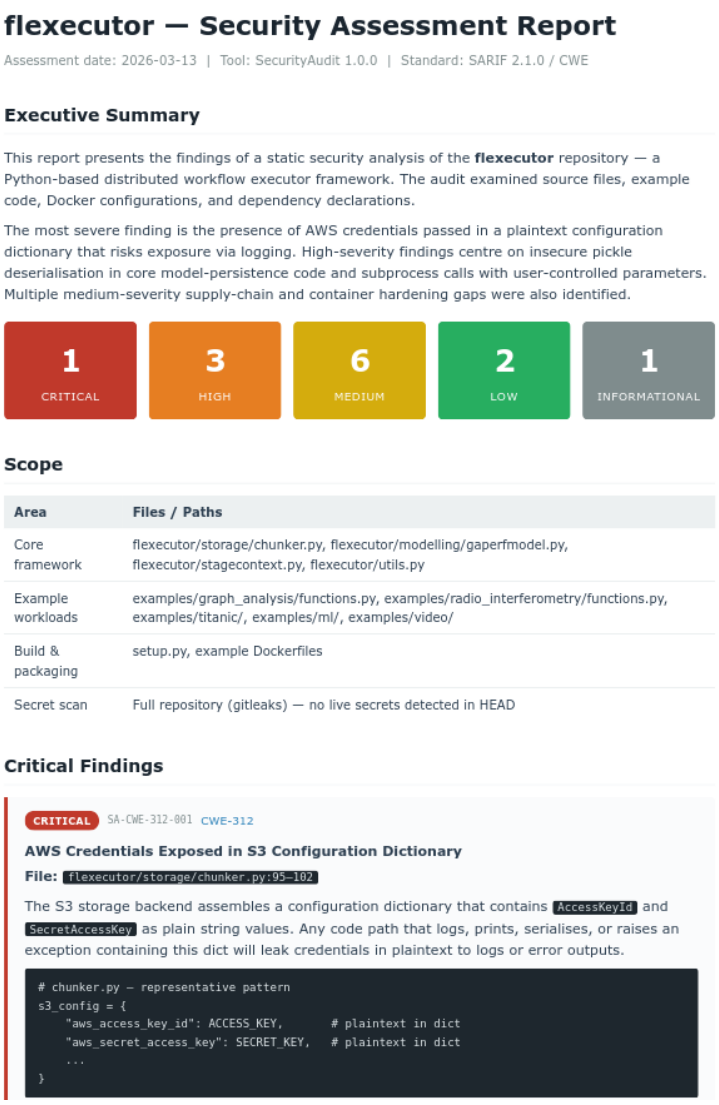
File: `flexecutor/storage/chunker.py:95–102`

The S3 storage backend assembles a configuration dictionary that contains `AccessKeyId` and `SecretAccessKey` as plain string values. Any code path that logs, prints, serialises, or raises an exception containing this dict will leak credentials in plaintext to logs or error outputs.

```
# chunker.py — representative pattern
s3_config = {
    "aws_access_key_id": ACCESS_KEY,       # plaintext in dict
    "aws_secret_access_key": SECRET_KEY,   # plaintext in dict
    ...
}
```

Remediation: Wrap credential values in a redacting wrapper class (override `__repr__` / `__str__` to return `"***"`). Never log the raw config dict. Use AWS IAM instance roles or environment-variable credential providers to avoid handling secrets in application code entirely.

Source code security audit report:



flexecutor — Security Assessment Report

Assessment date: 2026-03-13 | Tool: SecurityAudit 1.0.0 | Standard: SARIF 2.1.0 / CWE

Executive Summary

This report presents the findings of a static security analysis of the **flexecutor** repository — a Python-based distributed workflow executor framework. The audit examined source files, example code, Docker configurations, and dependency declarations.

The most severe finding is the presence of AWS credentials passed in a plaintext configuration dictionary that risks exposure via logging. High-severity findings centre on insecure pickle deserialisation in core model-persistence code and subprocess calls with user-controlled parameters. Multiple medium-severity supply-chain and container hardening gaps were also identified.

1	3	6	2	1
CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

Scope

Area	Files / Paths
Core framework	flexecutor/storage/chunker.py, flexecutor/modelling/gaperfmodel.py, flexecutor/stagecontext.py, flexecutor/utils.py
Example workloads	examples/graph_analysis/functions.py, examples/radio_interferometry/functions.py, examples/titanic/, examples/ml/, examples/video/
Build & packaging	setup.py, example Dockerfiles
Secret scan	Full repository (gitleaks) — no live secrets detected in HEAD

Critical Findings

CRITICAL SA-CWE-312-001 CWE-312

AWS Credentials Exposed in S3 Configuration Dictionary

File: flexecutor/storage/chunker.py:95–102

The S3 storage backend assembles a configuration dictionary that contains AccessKeyId and SecretAccessKey as plain string values. Any code path that logs, prints, serialises, or raises an exception containing this dict will leak credentials in plaintext to logs or error outputs.

```
# chunker.py — representative pattern
s3_config = {
    "aws_access_key_id": ACCESS_KEY,       # plaintext in dict
    "aws_secret_access_key": SECRET_KEY,   # plaintext in dict
    ...
}
```

No results for SBOM (Software Bill of Materials) third-party/supply-chain security audit:

imager

No secrets hard-coded or leaked were found in the final repository. Source code security audit report:

imager Security Audit Report

Executive Summary

A static security audit of the **imager** repository was performed on 2026-03-18. The repository provides radio-astronomy imaging primitives used by the `astrogym` pipeline, including WSClean and Wgridder gridders and the DeflationNet neural network restorer.

5 findings were identified across source code, CI pipeline, and dependency declarations.

Severity	Count
Critical	0
High	0
Medium	3
Low	1
Informational	1
Total	**5**

Key Risks

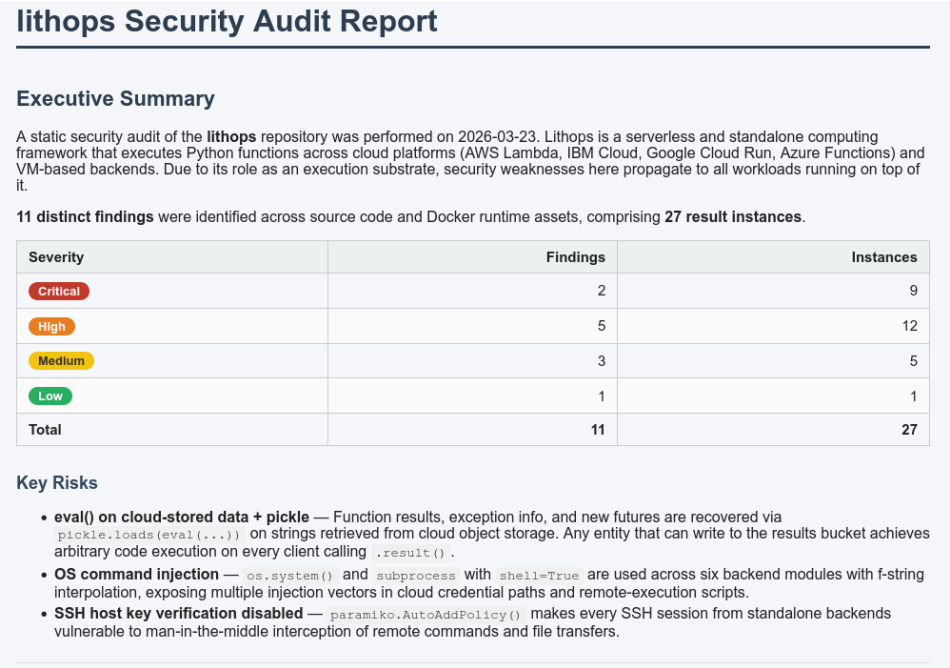
- **Information disclosure** — the WSClean binary path `/cep/lofar/wsclean-v3.6/bin/wsclean` is hardcoded, revealing the production HPC cluster filesystem layout and exact tool version.
- **Supply-chain exposure** — CI installs `ms-handler` from an unpinned git URL without integrity verification.
- **Dependency risk** — all packages use open-ended `>=` constraints with no CVE scanning in CI.

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-18 15:03:21.90233608 +0200 EET m=+0.107317655

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
filelock	CVE-2025-68146	MEDIUM	3.18.0	3.20.1	https://access.redhat.com/security/cve/CVE-2025-68146 https://github.com/tox-dev/filelock https://github.com/tox-dev/filelock/commit/4724d7f8c3393ec1f048c93933e6e3e6ec321f0e Toggle more links
filelock	CVE-2026-22701	MEDIUM	3.18.0	3.20.3	https://access.redhat.com/security/cve/CVE-2026-22701 https://github.com/tox-dev/filelock https://github.com/tox-dev/filelock/commit/255ed068bc85d1ef406e50a135e1459170dd1bf0 Toggle more links
fonttools	CVE-2025-66034	MEDIUM	4.57.0	4.60.2	https://access.redhat.com/security/cve/CVE-2025-66034 https://github.com/fonttools/fonttools https://github.com/fonttools/fonttools/commit/a696d5ba93270d5954f98e7cab5ddca8a02c1e32 Toggle more links
h11	CVE-2025-43859	CRITICAL	0.14.0	0.16.0	https://access.redhat.com/security/cve/CVE-2025-43859 https://github.com/python-hyper/h11 https://github.com/python-hyper/h11/commit/114803a29ce50116dc47951c690ad4892b1a36ed Toggle more links
jupyter-core	CVE-2025-30167	HIGH	5.7.2	5.8.1	https://github.com/jupyter/jupyter_core https://github.com/jupyter/jupyter_core/commit/5e8965600adda6b416692ce7e85ecb2bd814bd52 https://github.com/jupyter/jupyter_core/security/advisories/GHSA-33p9-3p43-82vq Toggle more links
jupyterlab	CVE-2025-59842	LOW	4.4.1	4.4.8	https://access.redhat.com/security/cve/CVE-2025-59842 https://github.com/jupyterlab/jupyterlab https://github.com/jupyterlab/jupyterlab/commit/88ef373039a8cc09f27d3814382a512d9033675c Toggle more links
nbconvert	CVE-2025-53000	HIGH	7.16.6	7.17.0	https://access.redhat.com/security/cve/CVE-2025-53000 https://github.com/jupyter/nbconvert https://github.com/jupyter/nbconvert/blob/4f61702f5c7524d8a3c4ac0d5fc33a6ac2fa36a7/nbconvert/preprocessors/svg2pdf.py#L104 Toggle more links
pillow	CVE-2025-48379	HIGH	11.2.1	11.3.0	https://access.redhat.com/security/cve/CVE-2025-48379 https://github.com/pypa/advisory-database/tree/main/vulns/pillow/PYSEC-2025-61.yaml https://github.com/python-pillow/Pillow Toggle more links
pillow	CVE-2026-25990	HIGH	11.2.1	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
requests	CVE-2024-47081	MEDIUM	2.32.3	2.32.4	http://seclists.org/fulldisclosure/2025/Jun/2 http://www.openwall.com/lists/oss-security/2025/06/03/11 http://www.openwall.com/lists/oss-security/2025/06/03/9 Toggle more links
torch	CVE-2025-3730	MEDIUM	2.7.0	2.8.0	https://github.com/pytorch/pytorch https://github.com/pytorch/pytorch/commit/01f226bfb8f2c343f5c614a6bbf685d91160f3af https://github.com/pytorch/pytorch/issues/150835 Toggle more links
torch	CVE-2025-2953	LOW	2.7.0	2.7.1-rc1	https://access.redhat.com/security/cve/CVE-2025-2953 https://github.com/pytorch/pytorch https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models Toggle more links
tornado	CVE-2025-47287	HIGH	6.4.2	6.5	https://access.redhat.com/errata/RHSA-2025:8136 https://access.redhat.com/security/cve/CVE-2025-47287 https://bugzilla.redhat.com/2366703 Toggle more links
tornado	CVE-2026-31958	HIGH	6.4.2	6.5.5	https://access.redhat.com/security/cve/CVE-2026-31958 https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/119a195e290c43ad2d63a2cf012c29d43d6ed839 Toggle more links
tornado	GHSA-78cv-mqj4-43f7	MEDIUM	6.4.2	6.5.5	https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/24a2d96ea115f663b223887deb0060f13974c104 https://github.com/tornadoweb/tornado/releases/tag/v6.5.5 Toggle more links
urllib3	CVE-2025-66418	HIGH	2.4.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66418 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-66471	HIGH	2.4.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66471 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2026-21441	HIGH	2.4.0	2.6.3	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2026-21441 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-50181	MEDIUM	2.4.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50181 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/f05b1329126d5be6de501f9d1e3e36738bc08857 Toggle more links
urllib3	CVE-2025-50182	MEDIUM	2.4.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50182 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/7eb4a2aafe49a279c29b6d1f0ed0f42e9736194f Toggle more links
No Misconfigurations found					

lithops
Tests/Doc-ONLY secrets hard-coded or leaked were found in the final repository. Source code security audit report.



lithops Security Audit Report

Executive Summary

A static security audit of the **lithops** repository was performed on 2026-03-23. Lithops is a serverless and standalone computing framework that executes Python functions across cloud platforms (AWS Lambda, IBM Cloud, Google Cloud Run, Azure Functions) and VM-based backends. Due to its role as an execution substrate, security weaknesses here propagate to all workloads running on top of it.

11 distinct findings were identified across source code and Docker runtime assets, comprising **27 result instances**.

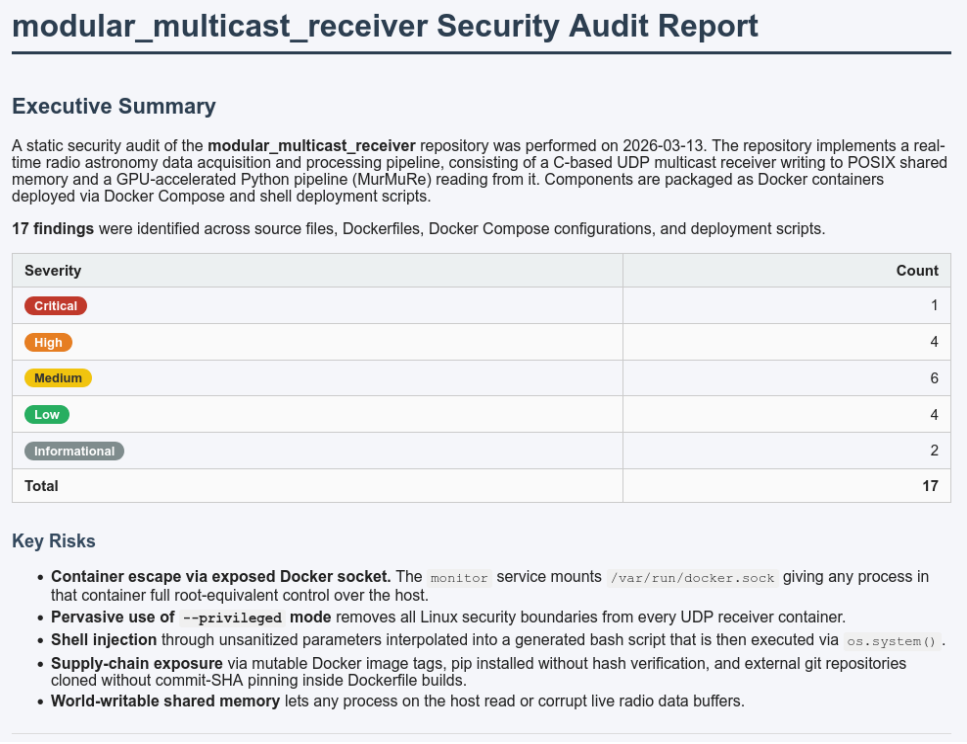
Severity	Findings	Instances
Critical	2	9
High	5	12
Medium	3	5
Low	1	1
Total	11	27

Key Risks

- **eval() on cloud-stored data + pickle** — Function results, exception info, and new futures are recovered via `pickle.loads(eval(...))` on strings retrieved from cloud object storage. Any entity that can write to the results bucket achieves arbitrary code execution on every client calling `.result()`.
- **OS command injection** — `os.system()` and `subprocess` with `shell=True` are used across six backend modules with f-string interpolation, exposing multiple injection vectors in cloud credential paths and remote-execution scripts.
- **SSH host key verification disabled** — `paramiko.AutoAddPolicy()` makes every SSH session from standalone backends vulnerable to man-in-the-middle interception of remote commands and file transfers.

No results for SBOM (Software Bill of Materials) third-party/supply-chain security audit:

modular_multicast_receiver
No secrets hard-coded or leaked were found in the final repository. Source code security audit report:



modular_multicast_receiver Security Audit Report

Executive Summary

A static security audit of the **modular_multicast_receiver** repository was performed on 2026-03-13. The repository implements a real-time radio astronomy data acquisition and processing pipeline, consisting of a C-based UDP multicast receiver writing to POSIX shared memory and a GPU-accelerated Python pipeline (MurMuRe) reading from it. Components are packaged as Docker containers deployed via Docker Compose and shell deployment scripts.

17 findings were identified across source files, Dockerfiles, Docker Compose configurations, and deployment scripts.

Severity	Count
Critical	1
High	4
Medium	6
Low	4
Informational	2
Total	17

Key Risks

- **Container escape via exposed Docker socket.** The `monitor` service mounts `/var/run/docker.sock` giving any process in that container full root-equivalent control over the host.
- **Pervasive use of `--privileged` mode** removes all Linux security boundaries from every UDP receiver container.
- **Shell injection** through unsanitized parameters interpolated into a generated bash script that is then executed via `os.system()`.
- **Supply-chain exposure** via mutable Docker image tags, pip installed without hash verification, and external git repositories cloned without commit-SHA pinning inside Dockerfile builds.
- **World-writable shared memory** lets any process on the host read or corrupt live radio data buffers.

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

modular_multicast_receiver/modular_multicast_receiver_image_docker_latest_sbom.json (alpine 3.23.3) - Trivy Report - 2026-03-13 14:37:05.305066606 +0200 EET m=+0.123726188

alpine

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
zlib	CVE-2026-22184	CRITICAL	1.3.1-r2	1.3.2-r0	https://access.redhat.com/security/cve/CVE-2026-22184 https://github.com/madler/zlib https://github.com/madler/zlib/issues/1142 Toggle more links
zlib	CVE-2026-27171	MEDIUM	1.3.1-r2	1.3.2-r0	https://7asecurity.com/blog/2026/02/zlib-7asecurity-audit/ https://7asecurity.com/reports/pentest-report-zlib-RC1.1.pdf https://access.redhat.com/security/cve/CVE-2026-27171 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
go.opentelemetry.io/otel/sdk	CVE-2026-24051	HIGH	v1.37.0	1.40.0	https://github.com/open-telemetry/opentelemetry-go https://github.com/open-telemetry/opentelemetry-go/commit/d45961bcda453fcbdb6469c22d6e88a1f9970a53 https://github.com/open-telemetry/opentelemetry-go/security/advisories/GHSA-9h8m-3fm2-qjrq Toggle more links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
golang.org/x/net	CVE-2025-22870	MEDIUM	v0.35.0	0.36.0	http://www.openwall.com/lists/oss-security/2025/03/07/2 https://access.redhat.com/security/cve/CVE-2025-22870 https://github.com/golang/go/issues/71984 Toggle more links
golang.org/x/net	CVE-2025-22872	MEDIUM	v0.35.0	0.38.0	https://access.redhat.com/security/cve/CVE-2025-22872 https://github.com/TheDegenerateDev5150/net/commit/e1fcd82abba34df74614020343be8eb1fe65f0d9 https://github.com/advisories/GHSA-vvgc-356p-c3xw Toggle more links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
go.opentelemetry.io/otel/sdk	CVE-2026-24051	HIGH	v1.38.0	1.40.0	https://github.com/open-telemetry/opentelemetry-go https://github.com/open-telemetry/opentelemetry-go/commit/d45961bcda453fcbdb8469c22d6e88a1f9970a53 https://github.com/open-telemetry/opentelemetry-go/security/advisories/GHSA-9h8m-3fm2-qjrq Toggle more links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
github.com/sigstore/sigstore	CVE-2026-24137	MEDIUM	v1.10.0	1.10.4	https://access.redhat.com/security/cve/CVE-2026-24137 https://github.com/sigstore/sigstore https://github.com/sigstore/sigstore/commit/8ec410a2993ea78083acf0e473a85453039486e Toggle more links
go.opentelemetry.io/otel/sdk	CVE-2026-24051	HIGH	v1.38.0	1.40.0	https://github.com/open-telemetry/opentelemetry-go https://github.com/open-telemetry/opentelemetry-go/commit/d45961bcda453fcbdb6469c22d6e88a1f9970a53 https://github.com/open-telemetry/opentelemetry-go/security/advisories/GHSA-9h8m-3fm2-qjrq Toggle more links
stdlib	CVE-2026-25679	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-25679 https://go.dev/cl/752180 https://go.dev/issue/77578 Toggle more links
stdlib	CVE-2026-27142	MEDIUM	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27142 https://go.dev/cl/752081 https://go.dev/issue/77954 Toggle more links
stdlib	CVE-2026-27139	LOW	v1.25.7	1.25.8, 1.26.1	https://access.redhat.com/security/cve/CVE-2026-27139 https://go.dev/cl/749480 https://go.dev/issue/77827 Toggle more links

No Misconfigurations found

monitoring-api

No secrets hard-coded or leaked were found in the final repository. Source code security audit report:

monitoring-api Security Audit Report

Executive Summary

A static security audit of the **monitoring-api** repository was performed on 2026-03-13. The repository implements a FastAPI service that exposes Kubernetes cluster topology and Prometheus metrics (CPU, memory, GPU, power, network) via REST endpoints, packaged as a Docker image for Kubernetes or Docker Compose deployment.

18 findings were identified across Python source files, Dockerfile, and configuration.

Severity	Count
Critical	3
High	6
Medium	5
Low	2
Informational	2
Total	**18**

Key Risks

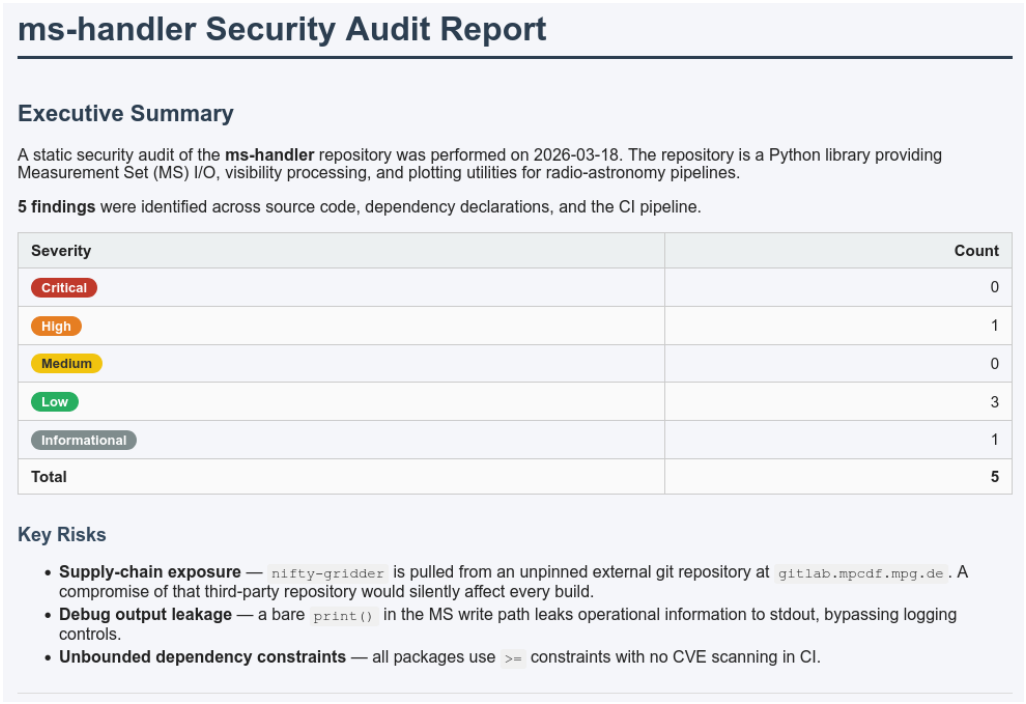
- **No authentication** on any endpoint — Kubernetes node data, GPU status, and raw Prometheus metrics are publicly accessible.
- **Prometheus query injection** via f-string interpolation of user-controlled parameters into PromQL queries.
- **Full exception details** returned to HTTP clients, potentially leaking database connection strings, file paths, or stack traces.
- **Cluster-wide Kubernetes API access** using admin kubeconfig; no RBAC restrictions applied.

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

zlib1g	CVE-2023-45853	CRITICAL	1:1.2.13.dfsg-1		http://www.openwall.com/lists/oss-security/2023/10/20/9 http://www.openwall.com/lists/oss-security/2024/01/24/10 https://access.redhat.com/security/cve/CVE-2023-45853 Toggle more links
zlib1g	CVE-2026-27171	MEDIUM	1:1.2.13.dfsg-1		https://7asecurity.com/blog/2026/02/zlib-7asecurity-audit/ https://7asecurity.com/reports/pentest-report-zlib-RC1.1.pdf https://access.redhat.com/security/cve/CVE-2026-27171 Toggle more links
No Misconfigurations found					
python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
setuptools	CVE-2024-6345	HIGH	65.5.1	70.0.0	https://access.redhat.com/errata/RHSA-2024:6726 https://access.redhat.com/security/cve/CVE-2024-6345 https://bugzilla.redhat.com/2297771 Toggle more links
setuptools	CVE-2025-47273	HIGH	65.5.1	78.1.1	https://access.redhat.com/errata/RHSA-2025:13578 https://access.redhat.com/security/cve/CVE-2025-47273 https://bugzilla.redhat.com/2366982 Toggle more links
starlette	CVE-2025-62727	HIGH	0.41.3	0.49.1	https://access.redhat.com/security/cve/CVE-2025-62727 https://github.com/Kludex/starlette https://github.com/Kludex/starlette/commit/4ea6e22b489ec388d6004cfbca52dd5b147127c5 Toggle more links
starlette	CVE-2025-54121	MEDIUM	0.41.3	0.47.2	https://access.redhat.com/security/cve/CVE-2025-54121 https://github.com/encode/starlette https://github.com/encode/starlette/blob/fa5355442753f794965ae1af0f87f9fec1b9a3de/starlette/datastructures.py#L436C5-L447C14 Toggle more links
wheel	CVE-2026-24049	HIGH	0.45.1	0.46.2	https://access.redhat.com/errata/RHSA-2026:1939 https://access.redhat.com/security/cve/CVE-2026-24049 https://bugzilla.redhat.com/2431959 Toggle more links
No Misconfigurations found					

ms-handler

No secrets hard-coded or leaked were found in the final repository. Source code security audit report:



ms-handler Security Audit Report

Executive Summary

A static security audit of the **ms-handler** repository was performed on 2026-03-18. The repository is a Python library providing Measurement Set (MS) I/O, visibility processing, and plotting utilities for radio-astronomy pipelines.

5 findings were identified across source code, dependency declarations, and the CI pipeline.

Severity	Count
Critical	0
High	1
Medium	0
Low	3
Informational	1
Total	**5**

Key Risks

- **Supply-chain exposure** — `nifty-gridder` is pulled from an unpinned external git repository at `gitlab.mpcdf.mpg.de`. A compromise of that third-party repository would silently affect every build.
- **Debug output leakage** — a bare `print()` in the MS write path leaks operational information to stdout, bypassing logging controls.
- **Unbounded dependency constraints** — all packages use `>=` constraints with no CVE scanning in CI.

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-18 15:03:25.764668658 +0200 EET m=+0.066520067

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
fonttools	CVE-2025-66034	MEDIUM	4.56.0	4.60.2	https://access.redhat.com/security/cve/CVE-2025-66034 https://github.com/fonttools/fonttools https://github.com/fonttools/fonttools/commit/a696d5ba93270d5954f98e7cab5ddca8a02c1e32 Toggle more links
h11	CVE-2025-43859	CRITICAL	0.14.0	0.16.0	https://access.redhat.com/security/cve/CVE-2025-43859 https://github.com/python-hyper/h11 https://github.com/python-hyper/h11/commit/114803a29ce50116dc47951c690ad4892b1a36ed Toggle more links
jupyter-core	CVE-2025-30167	HIGH	5.7.2	5.8.1	https://github.com/jupyter/jupyter_core https://github.com/jupyter/jupyter_core/commit/5e8965600adda6b416692ce7e85ecb2bd814bd52 https://github.com/jupyter/jupyter_core/security/advisories/GHSA-33p9-3p43-82vq Toggle more links
jupyterlab	CVE-2025-59842	LOW	4.3.6	4.4.8	https://access.redhat.com/security/cve/CVE-2025-59842 https://github.com/jupyterlab/jupyterlab https://github.com/jupyterlab/jupyterlab/commit/88ef373039a8cc09f27d3814382a512d9033675c Toggle more links
nbconvert	CVE-2025-53000	HIGH	7.16.6	7.17.0	https://access.redhat.com/security/cve/CVE-2025-53000 https://github.com/jupyter/nbconvert https://github.com/jupyter/nbconvert/blob/4f61702f5c7524d8a3c4ac0d5fc33a6ac2fa36a7/nbconvert/preprocessors/svg2pdf.py#L104 Toggle more links
pillow	CVE-2026-25990	HIGH	11.1.0	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
requests	CVE-2024-47081	MEDIUM	2.32.3	2.32.4	http://seclists.org/fulldisclosure/2025/Jun/2 http://www.openwall.com/lists/oss-security/2025/06/03/11 http://www.openwall.com/lists/oss-security/2025/06/03/9 Toggle more links
setuptools	CVE-2025-47273	HIGH	76.0.0	78.1.1	https://access.redhat.com/errata/RHSA-2025:13578 https://access.redhat.com/security/cve/CVE-2025-47273 https://bugzilla.redhat.com/2366982 Toggle more links
tornado	CVE-2025-47287	HIGH	6.4.2	6.5	https://access.redhat.com/errata/RHSA-2025:8136 https://access.redhat.com/security/cve/CVE-2025-47287 https://bugzilla.redhat.com/2366703 Toggle more links
tornado	CVE-2026-31958	HIGH	6.4.2	6.5.5	https://access.redhat.com/security/cve/CVE-2026-31958 https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/119a195e290c43ad2d63a2cf012c29d43d6ed839 Toggle more links
tornado	GHSA-78cv-mqj4-43f7	MEDIUM	6.4.2	6.5.5	https://github.com/tornadoweb/tornado https://github.com/tornadoweb/tornado/commit/24a2d96ea115f663b223887deb0060f13974c104 https://github.com/tornadoweb/tornado/releases/tag/v6.5.5 Toggle more links
urllib3	CVE-2025-66418	HIGH	2.3.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66418 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-66471	HIGH	2.3.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66471 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2026-21441	HIGH	2.3.0	2.6.3	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2026-21441 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-50181	MEDIUM	2.3.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50181 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/f05b1329126d5be6de501f9d1e3e36738bc08857 Toggle more links
urllib3	CVE-2025-50182	MEDIUM	2.3.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50182 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/7eb4a2aafe49a279c29b6d1f0ed0f42e9736194f Toggle more links
No Misconfigurations found					

per-demo-sept
Secrets hard-coded or leaked were found in the final repository.

[CRIT-1] Hardcoded S3/MinIO Credentials in Shell Script

Field	Value
Rule ID	SA-CWE-798-001
CWE	CWE-798: Use of Hard-coded Credentials
CVSS (estimated)	9.1 (Critical)
File	`notebook/run.sh` lines 1–2

Description

```
export ACCESS_KEY="minio"
export SECRET_KEY="minio123"
```

S3 access and secret keys are committed in plaintext. Anyone who clones or views this repository obtains working credentials. Even if these are currently MinIO test defaults, they establish a dangerous pattern and may reflect production credentials in forks or derivative deployments.

Remediation

Remove credentials from the script. Use a `.env` file excluded by `.gitignore`, a Kubernetes Secret, or a secrets manager (HashiCorp Vault, AWS Secrets Manager):

```
# .env (never committed)
ACCESS_KEY=minio
SECRET_KEY=minio123

# run.sh
source .env || { echo "Create .env with ACCESS_KEY and SECRET_KEY"; exit 1; }
export ACCESS_KEY SECRET_KEY
```

[CRIT-2] Credentials Documented in README

Field	Value
Rule ID	SA-CWE-798-002
CWE	CWE-798: Hard-coded Credentials
CVSS (estimated)	8.5 (Critical)
File	`README.md` lines 115–116

Description

```
export ACCESS_KEY=minio
export SECRET_KEY=minio123
```

Credentials embedded in documentation are indexed by search engines, GitHub Code Search, and cached in browser history. Developers copying the snippet verbatim into production configurations carry the credentials forward.

Remediation

Replace with placeholder values and reference the secrets-management approach documented above:

```
export ACCESS_KEY=<YOUR_MINIO_ACCESS_KEY>
export SECRET_KEY=<YOUR_MINIO_SECRET_KEY>
```

Source code security audit report:

per-demo-sept Security Audit Report

Executive Summary

A static security audit of the **per-demo-sept** repository was performed on 2026-03-23. The repository implements a KServe/Kubernetes-based ML inference service for a branching-node decision model, including custom model server code, Kubernetes manifests, and a Jupyter notebook for edge deployment.

12 distinct findings were identified across source code, configuration files, and Docker assets, comprising **19 result instances**.

Severity	Findings	Instances
Critical	3	6
High	4	8
Medium	4	4
Low	1	1
Total	12	19

Key Risks

- **Hardcoded credentials** — MinIO/S3 access keys are stored in plaintext in both a shell script and the README, making credential theft trivial for anyone with repository access.
- **Unsafe model deserialization** — `torch.load()` is called without `weights_only=True` in three model files, enabling arbitrary code execution if a model file is poisoned.
- **Null-pointer dereference** — The `headers` parameter defaults to `None` in `preprocess()` but is written to unconditionally, crashing the service on every v1 API call.

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-12 20:22:08.447472126 +0200 EET m=+0.107673959

python-pkg

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
torch	CVE-2025-32434	CRITICAL	2.4.1	2.6.0	https://github.com/pypa/advisory-database/tree/main/vulns/torch/PYSEC-2025-41.yaml https://github.com/pytorch/pytorch https://github.com/pytorch/pytorch/commit/8d4b8a920a2172523deb95bf20e8e52d50649c04 Toggle more links
torch	CVE-2025-3730	MEDIUM	2.4.1	2.8.0	https://github.com/pytorch/pytorch https://github.com/pytorch/pytorch/commit/01f226bfb8f2c343f5c614a6bbf685d91160f3af https://github.com/pytorch/pytorch/issues/150835 Toggle more links
torch	CVE-2025-2953	LOW	2.4.1	2.7.1-rc1	https://access.redhat.com/security/cve/CVE-2025-2953 https://github.com/pytorch/pytorch https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models Toggle more links

No Misconfigurations found

pipeline

No secrets hard-coded or leaked were found in the final repository. Source code security audit report:

pipeline Security Audit Report

Executive Summary

A static security audit of the **pipeline** repository was performed on 2026-03-13. The repository implements a scientific data-processing workflow engine (*taskac*) that orchestrates radio-astronomy pipeline steps (DP3 calibration, WSCLEAN imaging) across multiple execution backends (Lithops, COMPSs, Lambda), packaged as Docker images for Kubernetes and Lambda deployment.

11 findings were identified across Python source files, Dockerfiles, and shell scripts.

Severity	Count
Critical	2
High	3
Medium	5
Low	1
Informational	0
Total	11

Key Risks

- **Arbitrary code execution**: `PythonS3Step` downloads Python source from S3 and executes it dynamically — any write access to the bucket yields full pipeline worker compromise.
- **Credential exposure**: GitLab credentials embedded in Docker build layer via shell substitution in git clone URL.
- **Path traversal**: S3 object keys used to derive local file paths without boundary validation.
- **Containers run as root**: No `USER` directive in any Dockerfile.

No results for SBOM (Software Bill of Materials) third-party/supply-chain security audit:

rl-agent

Secrets hard-coded or leaked were found in the final repository. Source code security audit report:

rl-agent — Security Assessment Report

Assessment date: 2026-03-13 | Tool: SecurityAudit 1.0.0 | Standard: SARIF 2.1.0 / CWE

URGENT — Secrets in Git History: Gitleaks detected 5 live secrets committed to git history: a Nuvla API secret, an AWS secret access key, a second Nuvla secret, and an API token. These are accessible via `git log` even though the files appear removed from HEAD. **Rotate all affected credentials immediately** before taking any other action.

Executive Summary

This report presents the findings of a static security analysis of the **rl-agent** repository — a reinforcement-learning agent and simulator framework used for network resource management. The audit examined Python source files, Docker configurations, and dependency declarations, combined with a gitleaks secret scan of the full git history.

Three critical findings were identified: secrets committed to git history, insecure MQTT connections transmitting telemetry in cleartext, and unsafe `torch.load()` usage allowing arbitrary code execution via crafted model files. Five high-severity findings relate to further deserialisation, path traversal, and input validation deficiencies.

3	5	3	0	2
CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

Scope

Area	Files / Paths
Inference & training	inference.py, baseline.py
Simulator core	sim/SimulatorClass.py, sim/Plotter.py, sim/Inference.py
MQTT consumer	sim/Consumer/MQTTDataConsumer.py, sim/Consumer/mqttenv.py
Action decoder	ActionDecoder/extractConverter.py
Build & packaging	Dockerfile, requirements.txt
Secret scan	Full repository (gitleaks) — 5 secrets detected in git history

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-13 14:42:04.336498732 +0200 EET m=+0.095665681

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
fonttools	CVE-2025-66034	MEDIUM	4.57.0	4.60.2	https://access.redhat.com/security/cve/CVE-2025-66034 https://github.com/fonttools/fonttools https://github.com/fonttools/fonttools/commit/a696d5ba93270d5954f98e7cab5ddca8a02c1e32 Toggle more links
geopandas	CVE-2025-69662	HIGH	1.0.1	1.1.2	https://aydinnyunus.github.io/2025/12/27/sql-injection-geopandas https://aydinnyunus.github.io/2025/12/27/sql-injection-geopandas/ https://github.com/geopandas/geopandas Toggle more links
pillow	CVE-2025-48379	HIGH	11.2.1	11.3.0	https://access.redhat.com/security/cve/CVE-2025-48379 https://github.com/pypa/advisory-database/tree/main/vulns/pillow/PYSEC-2025-61.yaml https://github.com/python-pillow/Pillow Toggle more links
pillow	CVE-2026-25990	HIGH	11.2.1	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
requests	CVE-2024-47081	MEDIUM	2.32.3	2.32.4	http://seclists.org/fulldisclosure/2025/Jun/2 http://www.openwall.com/lists/oss-security/2025/06/03/11 http://www.openwall.com/lists/oss-security/2025/06/03/9 Toggle more links
urllib3	CVE-2025-66418	HIGH	2.4.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66418 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-66471	HIGH	2.4.0	2.6.0	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2025-66471 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2026-21441	HIGH	2.4.0	2.6.3	https://access.redhat.com/errata/RHSA-2026:1239 https://access.redhat.com/security/cve/CVE-2026-21441 https://bugzilla.redhat.com/2419455 Toggle more links
urllib3	CVE-2025-50181	MEDIUM	2.4.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50181 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/f05b1329126d5be6de501f9d1e3e36738bc08857 Toggle more links
urllib3	CVE-2025-50182	MEDIUM	2.4.0	2.5.0	https://access.redhat.com/security/cve/CVE-2025-50182 https://github.com/urllib3/urllib3 https://github.com/urllib3/urllib3/commit/7eb4a2aafe49a279c29b6d1f0ed0f42e9736194f Toggle more links
No Misconfigurations found					

skystore

Tests/Doc-ONLY secrets hard-coded or leaked were found in the final repository. (ignoring).Source code security audit report:
SBOM (Software Bill of Materials) third-party/supply-chain security audit:

- Trivy Report - 2026-03-12 20:21:43.859790192 +0200 EET m=+0.122283446

cargo

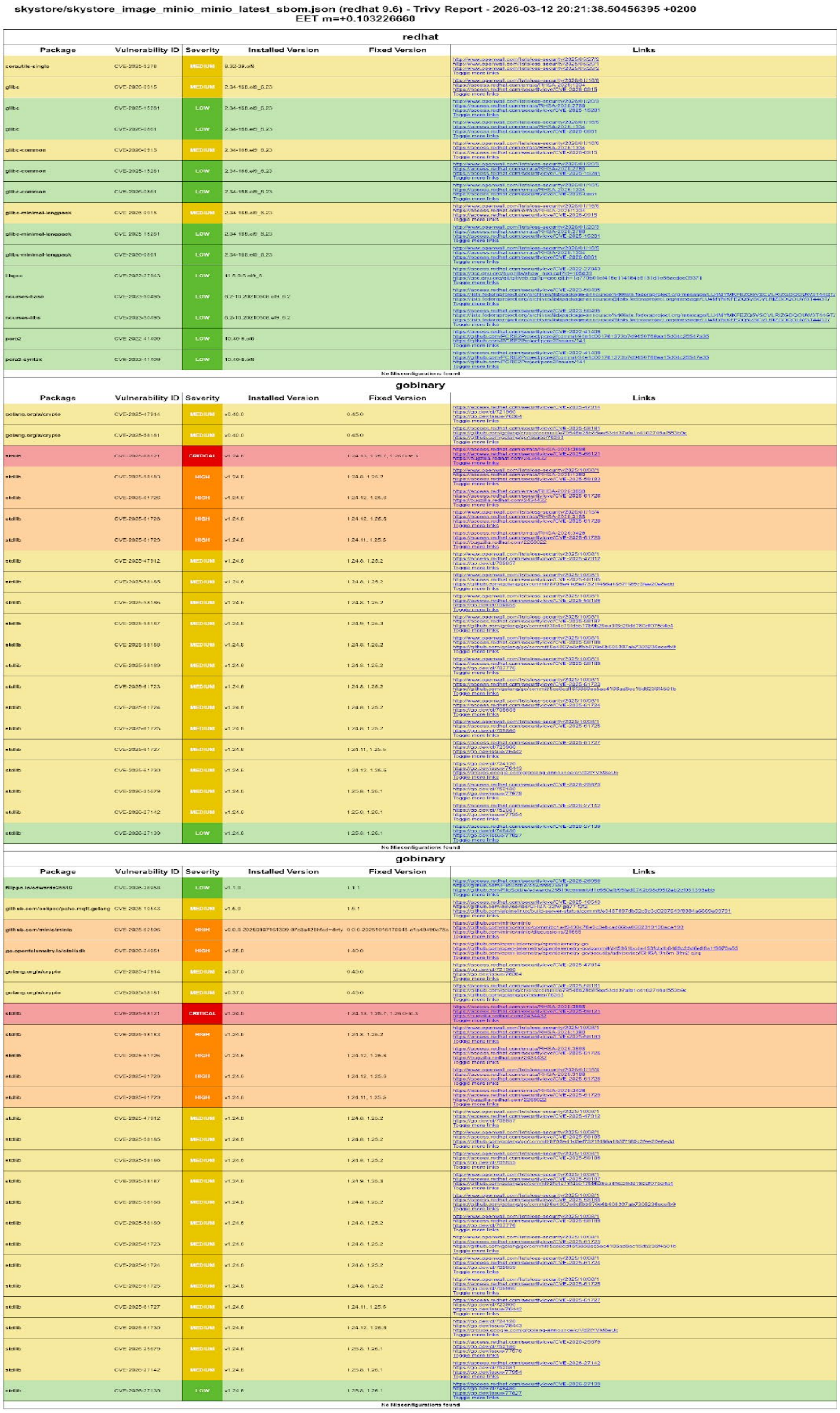
Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
aws-sdk-s3	GHSA-g59m-gf8j-gjf5	LOW	0.28.0	1.112.0	[illegible]
aws-sdk-sso	GHSA-g59m-gf8j-gjf5	LOW	0.28.0	1.89.0	[illegible]
aws-sdk-sts	GHSA-g59m-gf8j-gjf5	LOW	0.28.0	1.91.0	[illegible]
bytes	CVE-2026-25541	MEDIUM	1.4.0	1.11.1	[illegible]
bytes	CVE-2026-25541	MEDIUM	1.5.0	1.11.1	[illegible]
h2	GHSA-8r5v-vm4m-4g25	MEDIUM	0.3.18	0.3.24, 0.4.2	[illegible]
h2	GHSA-c6cp-cfwc-4gcv	MEDIUM	0.3.18	0.3.26, 0.4.4	[illegible]
h2	GHSA-8r5v-vm4m-4g25	MEDIUM	0.3.20	0.3.24, 0.4.2	[illegible]
h2	GHSA-c6cp-cfwc-4gcv	MEDIUM	0.3.20	0.3.26, 0.4.4	[illegible]
idna	CVE-2024-12224	MEDIUM	0.3.0	1.0.0	[illegible]
idna	CVE-2024-12224	MEDIUM	0.4.0	1.0.0	[illegible]
jsonwebtoken	CVE-2026-25537	MEDIUM	8.3.0	10.3.0	[illegible]
mio	CVE-2024-27308	HIGH	0.8.6	0.8.11	[illegible]
mio	CVE-2024-27308	HIGH	0.8.8	0.8.11	[illegible]
openssl	CVE-2023-53159	MEDIUM	0.10.52	0.10.56	[illegible]
openssl	CVE-2025-24898	MEDIUM	0.10.52	0.10.70	[illegible]
openssl	GHSA-4fcv-w3qc-ppgg	MEDIUM	0.10.52	0.10.72	[illegible]
openssl	GHSA-c445-7m23-qmw	MEDIUM	0.10.52	0.10.66	[illegible]
openssl	GHSA-xphf-cx8h-7q9g	MEDIUM	0.10.52	0.10.60	[illegible]
openssl	CVE-2025-24898	MEDIUM	0.10.55	0.10.70	[illegible]
openssl	GHSA-4fcv-w3qc-ppgg	MEDIUM	0.10.55	0.10.72	[illegible]
openssl	GHSA-c445-7m23-qmw	MEDIUM	0.10.55	0.10.66	[illegible]
openssl	GHSA-xphf-cx8h-7q9g	MEDIUM	0.10.55	0.10.60	[illegible]
ring	CVE-2025-4432	MEDIUM	0.16.20	0.17.12	[illegible]
rsa	CVE-2023-49092	MEDIUM	0.6.1		[illegible]
rsa	GHSA-4grx-2x9w-596c	MEDIUM	0.6.1		[illegible]
rsa	CVE-2026-21895	LOW	0.6.1	0.9.10	[illegible]
rustix	CVE-2024-43806	MEDIUM	0.37.19	0.35.15, 0.36.16, 0.37.25, 0.38.19	[illegible]
rustix	CVE-2024-43806	MEDIUM	0.37.22	0.35.15, 0.36.16, 0.37.25, 0.38.19	[illegible]
time	CVE-2020-26235	MEDIUM	0.1.45	0.2.23	[illegible]
time	CVE-2026-25727	MEDIUM	0.3.21	0.3.47	[illegible]
time	CVE-2026-25727	MEDIUM	0.3.22	0.3.47	[illegible]
tokio	GHSA-rr8g-9fpq-6wmg	LOW	1.28.0	1.44.2, 1.38.2, 1.43.1	[illegible]
tokio	GHSA-rr8g-9fpq-6wmg	LOW	1.33.0	1.44.2, 1.38.2, 1.43.1	[illegible]
tracing	GHSA-824-6n29-wn2r	MEDIUM	0.1.38	0.1.40	[illegible]
tracing-subscriber	CVE-2025-58160	LOW	0.3.17	0.3.20	[illegible]
webpki	GHSA-8cv2-6vc6-g2g7	HIGH	0.22.0	0.22.2	[illegible]

No Misconfigurations found

python-pkg

Package	Vulnerability ID	Severity	Installed Version	Fixed Version	Links
certifi	CVE-2024-39689	LOW	2023.7.22	2024.7.4	[illegible]
idna	CVE-2024-3651	MEDIUM	3.4	3.7	[illegible]
requests	CVE-2024-35195	MEDIUM	2.31.0	2.32.0	[illegible]
requests	CVE-2024-47081	MEDIUM	2.31.0	2.32.4	[illegible]
urllib3	CVE-2025-66418	HIGH	2.0.7	2.6.0	[illegible]
urllib3	CVE-2025-66471	HIGH	2.0.7	2.6.0	[illegible]
urllib3	CVE-2026-21441	HIGH	2.0.7	2.6.3	[illegible]
urllib3	CVE-2024-37891	MEDIUM	2.0.7	1.26.19, 2.2.2	[illegible]
urllib3	CVE-2025-50181	MEDIUM	2.0.7	2.5.0	[illegible]

No Misconfigurations found



skystore/skystore_image_minio_minio_latest_sbom.json (redhat 9.6) - Trivy Report - 2026-03-12 20:21:38.50456395 +0200 EET m=+0.103226660

redhat

Package	Vulnerability ID	Severity	Installed Version	Fixed Version
coreutils-single	CVE-2025-5278	MEDIUM	8.32-39.el9	
glibc	CVE-2026-0915	MEDIUM	2.34-168.el9_6.23	
glibc	CVE-2025-15281	LOW	2.34-168.el9_6.23	
glibc	CVE-2026-0861	LOW	2.34-168.el9_6.23	
glibc-common	CVE-2026-0915	MEDIUM	2.34-168.el9_6.23	
glibc-common	CVE-2025-15281	LOW	2.34-168.el9_6.23	
glibc-common	CVE-2026-0861	LOW	2.34-168.el9_6.23	
glibc-minimal-langpack	CVE-2026-0915	MEDIUM	2.34-168.el9_6.23	
glibc-minimal-langpack	CVE-2025-15281	LOW	2.34-168.el9_6.23	
glibc-minimal-langpack	CVE-2026-0861	LOW	2.34-168.el9_6.23	
libgcc	CVE-2022-27943	LOW	11.5.0-5.el9_5	
ncurses-base	CVE-2023-50495	LOW	6.2-10.20210508.el9_6.2	
ncurses-libs	CVE-2023-50495	LOW	6.2-10.20210508.el9_6.2	
pcre2	CVE-2022-41409	LOW	10.40-6.el9	
pcre2-syntax	CVE-2022-41409	LOW	10.40-6.el9	

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version
golang.org/x/crypto	CVE-2025-47914	MEDIUM	v0.40.0	0.45.0
golang.org/x/crypto	CVE-2025-58181	MEDIUM	v0.40.0	0.45.0
stdlib	CVE-2025-68121	CRITICAL	v1.24.6	1.24.13, 1.25.7, 1.26.0-rc.3
stdlib	CVE-2025-58183	HIGH	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61726	HIGH	v1.24.6	1.24.12, 1.25.6
stdlib	CVE-2025-61728	HIGH	v1.24.6	1.24.12, 1.25.6
stdlib	CVE-2025-61729	HIGH	v1.24.6	1.24.11, 1.25.5
stdlib	CVE-2025-47912	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58185	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58186	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58187	MEDIUM	v1.24.6	1.24.9, 1.25.3
stdlib	CVE-2025-58188	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58189	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61723	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61724	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61725	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61727	MEDIUM	v1.24.6	1.24.11, 1.25.5
stdlib	CVE-2025-61730	MEDIUM	v1.24.6	1.24.12, 1.25.6
stdlib	CVE-2026-25679	MEDIUM	v1.24.6	1.25.8, 1.26.1
stdlib	CVE-2026-27142	MEDIUM	v1.24.6	1.25.8, 1.26.1
stdlib	CVE-2026-27139	LOW	v1.24.6	1.25.8, 1.26.1

No Misconfigurations found

gobinary

Package	Vulnerability ID	Severity	Installed Version	Fixed Version
filippo.io/edwards25519	CVE-2026-26958	LOW	v1.1.0	1.1.1
github.com/eclipse/paho.mqtt.golang	CVE-2025-10543	MEDIUM	v1.5.0	1.5.1
github.com/minio/minio	CVE-2025-62506	HIGH	v0.0.0-20250907161309-07c3a429bfed+dirty	0.0.0-20251015170045-e1a49490c78e
go.opentelemetry.io/otel/sdk	CVE-2026-24051	HIGH	v1.35.0	1.40.0
golang.org/x/crypto	CVE-2025-47914	MEDIUM	v0.37.0	0.45.0
golang.org/x/crypto	CVE-2025-58181	MEDIUM	v0.37.0	0.45.0
stdlib	CVE-2025-68121	CRITICAL	v1.24.6	1.24.13, 1.25.7, 1.26.0-rc.3
stdlib	CVE-2025-58183	HIGH	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61726	HIGH	v1.24.6	1.24.12, 1.25.6
stdlib	CVE-2025-61728	HIGH	v1.24.6	1.24.12, 1.25.6
stdlib	CVE-2025-61729	HIGH	v1.24.6	1.24.11, 1.25.5
stdlib	CVE-2025-47912	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58185	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58186	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58187	MEDIUM	v1.24.6	1.24.9, 1.25.3
stdlib	CVE-2025-58188	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-58189	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61723	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61724	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61725	MEDIUM	v1.24.6	1.24.8, 1.25.2
stdlib	CVE-2025-61727	MEDIUM	v1.24.6	1.24.11, 1.25.5
stdlib	CVE-2025-61730	MEDIUM	v1.24.6	1.24.12, 1.25.6
stdlib	CVE-2026-25679	MEDIUM	v1.24.6	1.25.8, 1.26.1
stdlib	CVE-2026-27142	MEDIUM	v1.24.6	1.25.8, 1.26.1
stdlib	CVE-2026-27139	LOW	v1.24.6	1.25.8, 1.26.1

No Misconfigurations found

taska-a1

No secrets hard-coded or leaked were found in the final repository. Source code security audit report:

taska-a1 Security Audit Report

Executive Summary

A static security audit of the **taska-a1** repository was performed on 2026-03-23. The repository implements the SpikeNet pipeline for detecting radio spikes in NenuFAR telescope data using a TensorRT-accelerated neural network, including inference runner, data readers, visualization tools, and model-conversion utilities.

11 distinct findings were identified across source code, configuration files, and container assets, comprising **18 result instances**.

Severity	Findings	Instances
High	2	6
Medium	7	10
Low	2	2
Total	**11**	**18**

Key Risks

- **Unsafe pickle deserialization** — `pickle.load()` is called across four call sites on inference result files. A poisoned result file or directory traversal feeding a malicious pickle into the glob pattern enables arbitrary code execution.
- **Division by zero in Stokes normalization** — `resultV /= resultI` is performed without zero-checking, producing silent `inf` / `NaN` corruption or a crash propagated through the entire analysis pipeline.
- **Hardcoded filesystem paths** — Absolute paths to internal cluster filesystems appear in six source files, disclosing infrastructure layout and making the code non-portable between environments.

SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-12 20:35:08.900037621 +0200 EET m=+0.086254766

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
ray	CVE-2023-48022	CRITICAL	2.36.0		https://access.redhat.com/security/cve/CVE-2023-48022 https://atlas.mitre.org/studies/AML.CS0023 https://bishopfox.com/blog/ray-versions-2-6-3-2-8-0 Toggle more links
ray	CVE-2025-62593	CRITICAL	2.36.0	2.52.0	https://access.redhat.com/security/cve/CVE-2025-62593 https://docs.ray.io/en/releases-2.51.1/ray-security/index.html https://en.wikipedia.org/wiki/Malvertising Toggle more links
ray	CVE-2025-1979	MEDIUM	2.36.0	2.43.0	https://access.redhat.com/security/cve/CVE-2025-1979 https://github.com/pypa/advisory-database/tree/main/vulns/ray/PYSEC-2025-23.yaml https://github.com/ray-project/ray Toggle more links
ray	CVE-2026-27482	MEDIUM	2.36.0	2.54.0	https://github.com/ray-project/ray https://github.com/ray-project/ray/commit/0fda8b824cdc9dc6edd763bb28dfd7d1cc9b02a4 https://github.com/ray-project/ray/pull/60526 Toggle more links
No Misconfigurations found					

taska-a2

No secrets hard-coded or leaked were found in the final repository. Source code security audit report:

taska-a2 Security Audit Report

Executive Summary

A static security audit of the **taska-a2** repository was performed on 2026-03-13. The repository implements a scientific pipeline for detecting millisecond radio S-bursts in NenuFAR and NDA/JunoN telescope data using a Radon-transform–based detection algorithm.

11 distinct findings were identified across source code, configuration, and version-control metadata, comprising **22 result instances**.

Severity	Findings	Instances
Critical	1	6
High	3	4
Medium	3	7
Low	2	3
Informational	2	2
Total	**11**	**22**

Key Risks

- **OS command injection** — User-supplied path strings are passed directly into `os.system()` across six call sites. An attacker who controls the `--output_folder` CLI argument can execute arbitrary commands at the privilege level of the running process.
- **Path traversal** — The same user-supplied paths are concatenated into HDF5 output paths and a log-file `FileHandler` without canonicalization, allowing writes outside the intended directory.
- **Supply-chain exposure** — An unpinned external git submodule and three completely unconstrained Python dependencies create vectors for malicious code injection through dependency confusion or upstream compromise.

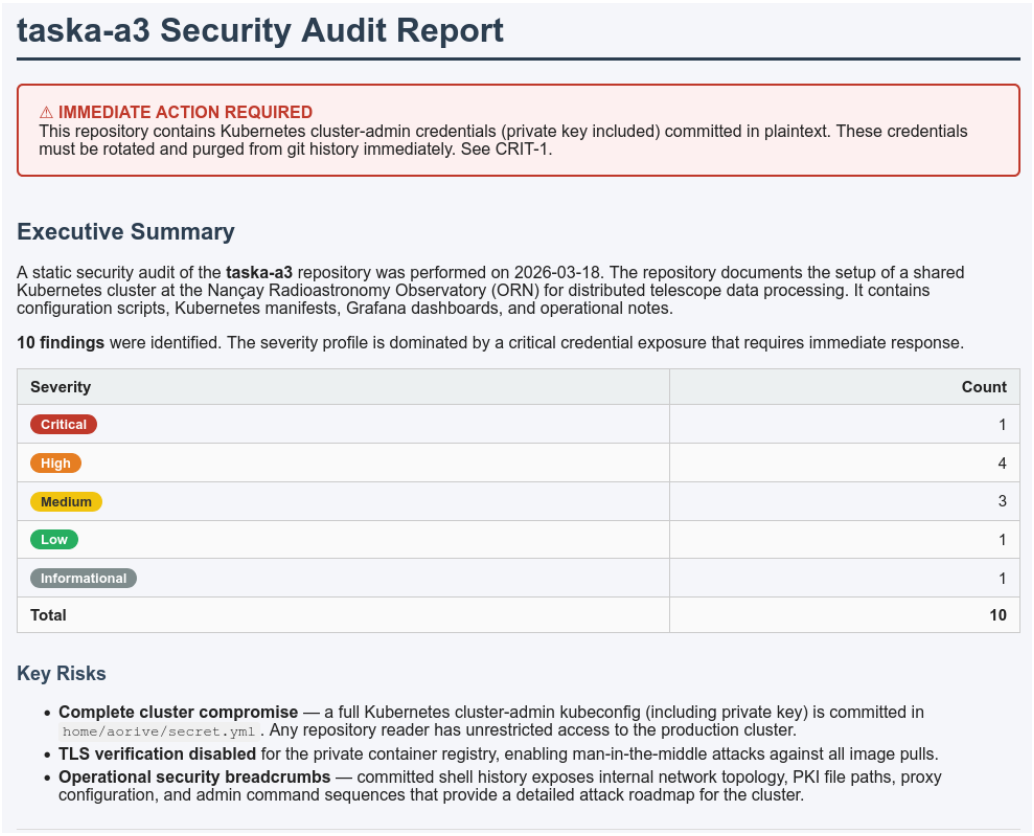
SBOM (Software Bill of Materials) third-party/supply-chain security audit:

Python - Trivy Report - 2026-03-12 20:21:50.659066295 +0200 EET m=+0.106367307

python-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
astropy	CVE-2023-41334	HIGH	5.2.2	5.3.3	https://github.com/astropy/astropy https://github.com/astropy/astropy/blob/9b97d98802ee4f5350a62b681c35d8687ee81d91/astropy/coordinates/transformations.py#L539 https://github.com/astropy/astropy/commit/22057d37b1313f5f5a9b5783df0a091d978dccb5 Toggle more links
fonttools	CVE-2025-66034	MEDIUM	4.57.0	4.60.2	https://access.redhat.com/security/cve/CVE-2025-66034 https://github.com/fonttools/fonttools https://github.com/fonttools/fonttools/commit/a696d5ba93270d5954f98e7cab5ddca8a02c1e32 Toggle more links
pillow	CVE-2026-25990	HIGH	10.4.0	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
pillow	CVE-2025-48379	HIGH	11.2.1	11.3.0	https://access.redhat.com/security/cve/CVE-2025-48379 https://github.com/pypa/advisory-database/tree/main/vulns/pillow/PYSEC-2025-61.yaml https://github.com/python-pillow/Pillow Toggle more links
pillow	CVE-2026-25990	HIGH	11.2.1	12.1.1	http://www.openwall.com/lists/oss-security/2026/02/12/1 https://access.redhat.com/security/cve/CVE-2026-25990 https://github.com/python-pillow/Pillow Toggle more links
No Misconfigurations found					

taska-a3

Secrets hard-coded or leaked were found in the final repository. Source code security audit report:



taska-a3 Security Audit Report

⚠ IMMEDIATE ACTION REQUIRED
This repository contains Kubernetes cluster-admin credentials (private key included) committed in plaintext. These credentials must be rotated and purged from git history immediately. See CRIT-1.

Executive Summary

A static security audit of the **taska-a3** repository was performed on 2026-03-18. The repository documents the setup of a shared Kubernetes cluster at the Nançay Radioastronomy Observatory (ORN) for distributed telescope data processing. It contains configuration scripts, Kubernetes manifests, Grafana dashboards, and operational notes.

10 findings were identified. The severity profile is dominated by a critical credential exposure that requires immediate response.

Severity	Count
Critical	1
High	4
Medium	3
Low	1
Informational	1
Total	**10**

Key Risks

- **Complete cluster compromise** — a full Kubernetes cluster-admin kubeconfig (including private key) is committed in `home/aorive/secret.yml`. Any repository reader has unrestricted access to the production cluster.
- **TLS verification disabled** for the private container registry, enabling man-in-the-middle attacks against all image pulls.
- **Operational security breadcrumbs** — committed shell history exposes internal network topology, PKI file paths, proxy configuration, and admin command sequences that provide a detailed attack roadmap for the cluster.

No results for SBOM (Software Bill of Materials) third-party/supply-chain security audit

3.3.1.1. Secure communications (data-in-transit)

- Communications between entities/pods/services within a deployment

All the access controls and communications between the modules/services/pods/clusters were checked and validated to use secure communication methods (HTTPS, SSH).

Certain modules in their test-modes were using insecure methods (e.g., insecure clear-text MQTT over port 1883), however those were easily fixed to use secure HTTPS MQTT over port 8883, which required minor adjustments to the MQTT server and client configurations.

Moreover, since EXTRACT Platform as a whole is not yet at TRL9 and there is no production-ready domain-name and DNS architecture established, that prevented the use of proper full deployment of complete CA-validated certificate-chains on all HTTPS endpoints. The HTTPS tests passed with self-signed certificates or auto-generated certificates (e.g., linkerd, mTLS), and therefore confirms the communications and data-in-transit is secure against eavesdropping, interception and interference in EXTRACT Platform deployments.

3.3.1.2. Secure data storage (data-at-rest)

- OS-level storage encryption

While Kubernetes abstraction(s) allows for encryption of data-at rest, it covers encryption for resource data stored using the Kubernetes API itself. For example, you can encrypt Secret objects, including the key-value data they contain. "encryption for resource data stored using the Kubernetes API itself" is generally a limiting factor, especially in highly complex and integrated systems such as EXTRACT Platform, and would require reworks on many levels in EXTRACT modules that do not natively support storing data via Kubernetes APIs, and rather use volumes (e.g., persistent) and application-specific encryptions (which still rely on some persistent storage or volumes). However, if one wants to encrypt data in filesystems that are mounted into containers, it instead need to either: use a storage integration that provides encrypted volumes; and/or encrypt the data within your own application.

Therefore, OS-level storage encryption was employed as a standard and transparent means to secure the data-at-rest (e.g., input data, output data, temporary data).

Linux Unified Key Setup (LUKS) is the standard for Linux hard disk encryption, providing full-disk or partition-level security by encrypting data at rest. It utilizes dm-crypt to protect sensitive information, requiring a passphrase at boot, making it ideal for securing laptops, removable media, and swap space against physical theft.

Protection: Secures data by making it inaccessible if the drive is stolen or removed, requiring a passphrase to unlock.

Implementation: Usually implemented during OS installation (e.g., Ubuntu, Fedora) or via the cryptsetup command-line tool.

Key Management: Supports up to 8 different user passphrases/keys per partition, allowing for secure key management.

Compatibility: Works on partitions, raw disks, and logical volumes (LVM).

LUKS (and associated persisten volumes that are encrypted) is also well supported by Kubernetes (LUKS), as well as by major Managed Kubernetes service providers such as OVH (which was also used for EXTRACT Platform deployment/validation/testing).

- TPM-level support

1. Runtime TPM hardenings

A Trusted Platform Module (TPM) is a specialized, hardware-based security chip located on a computer/server that generates, stores, and protects cryptographic keys and secret materials. In a nutshell explanation, it protects hardware from malware and unauthorized tampering by storing encryption keys and other configuration secrets in a highly-secure untamperable storage.

TPMs are generally available (or even mandatory) on most traditional modern computers and servers, as well as modern embedded computers (e.g., Raspberry Pi) and Single-Board-Computers (SBC) (somewhat similar to Raspberry Pi).

To ensure smooth testing, we have assumed and used a TPM-emulator implementation that provides the same TPM abstractions required across all various compute media and environments. This way, all the modules can have abstracted access to the TPM and therefore store/retrieve secrets, keys and password in an uniform and secure manner, independent of the environment where the module is developed, tested, or integrated.

More details about runtime TPM module (TPM 2.0, TPM-emulator) are in deliverable D4.5.

2. Pre-runtime TPM-like hardenings

The TPM is generally a runtime protection. However, code lives not only at runtime, but also in code repositories and it may require additional protections to store sensitive configurations/information, and secret passwords and keys. However, TPM is not usable in those conditions, and therefore we explored the usage of Ansible Vault as a code-repository equivalent of TPMs. In a nutshell, Ansible vault provides a way to encrypt and manage sensitive data such as passwords. Therefore, all runtime secrets of each module of the platform can be securely stored in the code repository using Ansible Vault, then extracted from there during the deployment to a specific implementation or CCS, and subsequently previously explained TPM techniques are used to store the secrets already on the running machines and services.

More details about the pre-runtime TPM-like module (ansible vault) are in deliverable D4.5.

3.3.1.3. **Overall security awareness**

- Generic Security Advisories related to core technologies (docker, kubernetes)

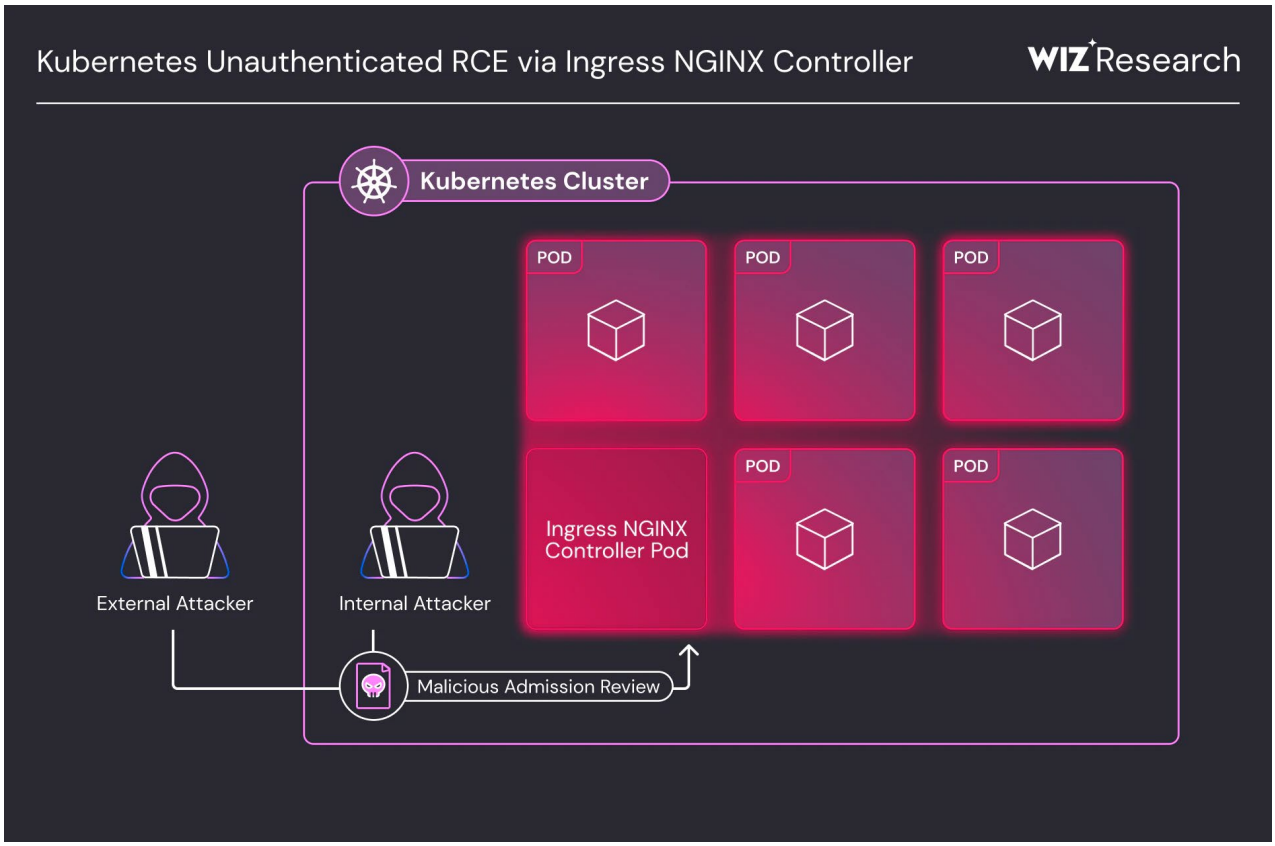
1. CVE-2025-1974: IngressNightmare in Kubernetes: CVE-2025-1974 - 9.8 Critical Unauthenticated Remote Code Execution Vulnerabilities in Ingress NGINX

Ingress NGINX deploys an admission controller within its pod, designed to validate incoming ingress objects before they are deployed. By default, admission controllers are accessible over the network without authentication, making them a highly appealing attack vector.

When the Ingress-NGINX admission controller processes an incoming ingress object, it constructs an NGINX configuration from it and then validates it using the NGINX binary. Our team found a vulnerability in this phase that allows injecting an arbitrary NGINX configuration remotely, by sending a malicious ingress object directly to the admission controller through the network.

During the configuration validation phase, the injected NGINX configuration causes the NGINX validator to execute code, allowing remote code execution (RCE) on the Ingress NGINX Controller's pod.

The admission controller's elevated privileges and unrestricted network accessibility create a critical escalation path. Exploiting this flaw allows an attacker to execute arbitrary code and access all cluster secrets across namespaces, that could lead to complete cluster takeover.



2. Kubernetes Pods Using Default Service Account with Excessive Permissions

A high-severity security vulnerability where Kubernetes pods use the default service account which often has overly broad permissions or access to sensitive cluster resources. Default service accounts are automatically assigned to pods unless explicitly overridden, and they frequently inherit cluster-wide permissions that violate the principle of least privilege. This can lead to privilege escalation, unauthorized access to secrets, and potential cluster compromise.

When Kubernetes pods are deployed without specifying a service account, they automatically use the 'default' service account in their namespace. This default account often has more permissions than necessary, especially in environments where cluster administrators have granted broad RBAC permissions to default service accounts for convenience.

3. Kubernetes nodes/proxy GET Remote Code Execution KEP-2862 (Fine-Grained Kubelet API Authorization) - A Deep-Dive into the Kubernetes nodes/proxy RCE

Kubernetes administrators often grant access to the nodes/proxy resource to service accounts requiring access to data such as Pod metrics and Container logs. As such, Kubernetes monitoring tools commonly require this resource for reading data.

nodes/proxy GET allows command execution when using a connection protocol such as WebSockets. This is due to the Kubelet making authorization decisions based on the initial WebSocket handshake's request without verifying CREATE permissions are present for the Kubelet's /exec endpoint requiring different permissions depending solely on the connection protocol.

The result is anyone with access to a service account assigned nodes/proxy GET that can reach a Node's Kubelet on port 10250 can send information to the /exec endpoint, executing commands in any Pod, including privileged system Pods, potentially leading to a full cluster compromise. Kubernetes AuditPolicy does not log commands executed through a direct connection to the Kubelet's API.

A service account with nothing more than nodes/proxy GET permission can achieve full remote code execution (RCE) in any container on a Kubernetes node. This permission is routinely granted to monitoring tools like Prometheus, Datadog, and Grafana for metrics collection. Depending on the security configuration, this could lead to a full cluster takeover.

Read differently: your Prometheus service account, which you thought could only scrape metrics, can execute arbitrary commands as root into any container on any node it can reach. A permission typically granted for observability becomes a permission for remote code execution.

The attacker now can have RCE as root inside the container. From here, an attacker can:

Extract environment variables (secrets, API keys, database credentials)

Read mounted volumes (potentially including service account tokens)

Make network requests to internal services

Install persistence mechanisms

Attempt container escape via kernel vulnerabilities

4. **In summary**:

We have validated in the affected vanilla kubernetes clusters/deployments that those vulnerabilities are generally possible. Nevertheless, the final documentation and configuration of the EXTRACT Platform makes sure that the vulnerable versions as well as vulnerable configurations are avoided by the end-user of the EXTRACT Platform.

4. Objective and KPI Validation

The focus of validation in this deliverable is objective O3.

4.1. KPI 3.1 Validation

The requirement in O3.1 is to demonstrate that the abstraction layer introduces limited overhead (below 20%) while still exploiting the available computing resources efficiently. This requirement is further detailed in deliverable D3.4. In summary, the results show that the multicluster deployment based on COMPSs reduces the training execution time in the PER workflow from 331.7 s (sequential) to 199.4 s, demonstrating effective exploitation of parallel resources. However, the abstraction layer introduces a noticeable overhead that exceeds the 20% target, reflecting the cost of orchestration and infrastructure abstraction across the compute continuum.

4.2. KPI 3.2 Validation

We have run test-batteries and measured the performance/state indicators with and without each individual security measure turned on. We aggregated and averaged the performance impact measurements across test-batteries, which resulted in the following summary table:

Security measure	**Averaged impact measurements**	**Main impact areas**
Secure code validation and scanning	<20%	☐ Code/config commits/merge pipelines ☐ Pre-deployment pipelines ☐ Docker image building pipelines ☐ Deployment pipelines
Data-at-rest security (e.g., OS-level storage encryption)	< 30%	☐ Storage I/O ☐ General I/O throughput
Data-in-transit security (e.g., HTTPS, secure communications)	< 25%	☐ Network I/O ☐ General I/O throughput
Runtime TPM Trusted Platform Module hardenings (e.g., TPM 2.0, TPM-emulator)	< 25%	☐ Pre-processing requiring secrets interaction with TPM ☐ Critical processing paths/modules requiring "secure state" update/check
Pre-runtime TPM-like hardenings (ansible vault)	< 10%	☐ Code/config commits/merge pipelines ☐ Pre-deployment pipelines ☐ Docker image building pipelines ☐ Deployment pipelines

5. Summary and Conclusions

5.1. KPI Assessment Validation Conclusions

Overall, we can state that the core objective of WP4 (O3) and its KPIs (KPI3.1, KPI3.2) were achieved and fulfilled as planned, using several iterations of implementation, integrations, testing and validation.

5.2. Lessons Learned

The work carried out in T4.4 and across WP4 has yielded several important lessons and recommendations:

- Multi-premise deployment complexity: Network heterogeneity (bandwidth, latency, firewalls, identity-federation constraints) remains a key challenge for seamless continuum deployments. Early joint planning with infrastructure providers and clear CCS blueprints significantly reduce integration effort.
- Abstraction trade-offs: The overhead introduced by the continuum abstraction layer is acceptable but not negligible. For latency-critical workflows, careful placement of components and selective use of cross-premise communication are essential. For throughput-oriented workloads, the benefits of added capacity clearly outweigh the overhead.
- Security vs. Performance: Enabling TLS/mTLS, continuous scanning and additional security controls has a measurable but manageable impact on performance. Designing these mechanisms from the start and treating them as first-class citizens of the architecture proved more effective than retrofitting security later.
- DevSecOps and automation: Automated pipelines for container scanning, SBOM generation and deployment testing greatly increased the reliability and repeatability of deployments. We recommend keeping these pipelines in place beyond the project and extending them to cover future components.
- Standard CCS blueprint: The concept of a Compute Continuum Site (CCS), with a well-defined set of services (Kubernetes, COMPSs, SkyStore client, catalog, monitoring, security stack), proved essential to manage complexity. We recommend adopting the CCS blueprint as the standard deployment unit for future EXTRACT-like platforms.
- Future work and adoption: For adopters of the EXTRACT technology, we recommend:
    - starting from the validated CCS configurations used in this deliverable,
    - re-using the provided monitoring and security setups as defaults, and
    - planning early for multi-premise connectivity and identity-management issues.

These lessons inform not only the final evaluation of WP4, but also the exploitation, standardisation and further research directions for compute continuum platforms beyond EXTRACT.

6. References

1. Kubernetes. https://kubernetes.io/ Retrieved March 21, 2026.

2. SkyStore. https://github.com/gilv/skystore/tree/tunneler Retrieved March 21, 2026.

3. AWS S3. https://aws.amazon.com/s3/ Retrieved March 21, 2026.

4. GCP. Google Cloud Storage. https://cloud.google.com/storage Retrieved March 21, 2026.

5. Azure. Azure Blob Storage. https://azure.microsoft.com/en-us/products/storage/blobs Retrieved March 21, 2026.

6. OVH. OVHcloud Object Storage. https://www.ovhcloud.com/en/public-cloud/object-storage/ Retrieved March 21, 2026.

7. Minio. S3 Compatible Exascale Object Store for AI. https://www.min.io/ Retrieved March 21, 2026.

8. Ceph. Ceph Object Gateway S3 API. https://docs.ceph.com/en/latest/radosgw/s3/ Retrieved March 21, 2026.

9. Virtual Enterprise. https://www.sciencedirect.com/topics/computer-science/virtual-enterprise Retrieved March 21, 2026.