Co-funded by the European Union

D3.2 Cloud infrastructure beta

Due date M9: 31/10/2025

Dissemination level: Public

Author:
Ignacio Lamata Martínez (EGI)

HISTORY OF CHANGES			
Version	Date	Author	Comments
0.1	14/10/2025	Ignacio Lamata Martinez (EGI)	First draft
0.2	21/10/2025	Frederik Temmermans (imec), Valentina Bachi (PHC)	Internal review
0.3	23/10/2025	Ignacio Lamata Martinez (EGI)	Final version
1.0	30/10/2025	Valentina Bachi (PHC)	Submitted version

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This document accompanies the Deliverable *D3.2 Cloud infrastructure beta* in order to provide documentation on the implemented beta version of the project's cloud infrastructure. It outlines the basic design principles, the different systems involved and the operational processes currently in place as part of the beta deployment, aiming to ensure an initial infrastructure as it evolves toward production readiness.

The infrastructure is illustrated in a showcase video, accessible at:
https://www.youtube.com/watch?v=UiTvOg4S078

The beta cloud infrastructure was developed in alignment with the project's core objectives of scalability, security, and cost-efficiency, enabling flexibility around the resources needed by the different tools. The deployment is based on container technology, automated provisioning, dynamic resource management, and continuous deployment pipelines. To secure the access to all underlying systems and the tools' features that require some user authorisation, the authentication and authorisation layer has been prepared accordingly.

While the current beta infrastructure is not final, it provides a stable foundation intended to support ongoing testing, feedback gathering, and iterative improvement.

1. INTRODUCTION

The EUreka3D-XR project develops various XR tools that will enrich the existing ecosystem of the EUreka3D Data Hub[1], the cloud solution available to Cultural Heritage Institutions to store, manage, share and publish their 3D cultural collections in the data space for cultural heritage. These XR tools aim to facilitate the reuse of 3D and 2D digital cultural content by Cultural Heritage Institutions in compelling XR storytelling and, as such, engage their onsite and online audiences. The tools are: AR TOUR BUILDER, AR TOUR EXPERIENCE, AI 3D BUILDER, 3D XR STUDIO and AVATAR BUILDER. They are being developed by three different teams of developers and tested in three main pilot scenarios of Cultural Heritage, as documented in deliverables *D3.1 Technical Requirements*[2] and *D2.1 Pilot Specification and Planning*[3]. These tools require servers to execute and auxiliary systems to operate, which make up the **cloud infrastructure**.

The general overview provided in Figure 1.1 shows that the EUreka3D-XR tools execute on dedicated project infrastructure that is accessed by three different actors:

- The **Cultural Heritage Community** accesses the different tools to use them.
- The "**editors**" are a special type of users that are authenticated and authorised to perform restricted actions in the system.
- The **EUreka3D-XR developers and operators** create the tools and deploy them in the infrastructure.

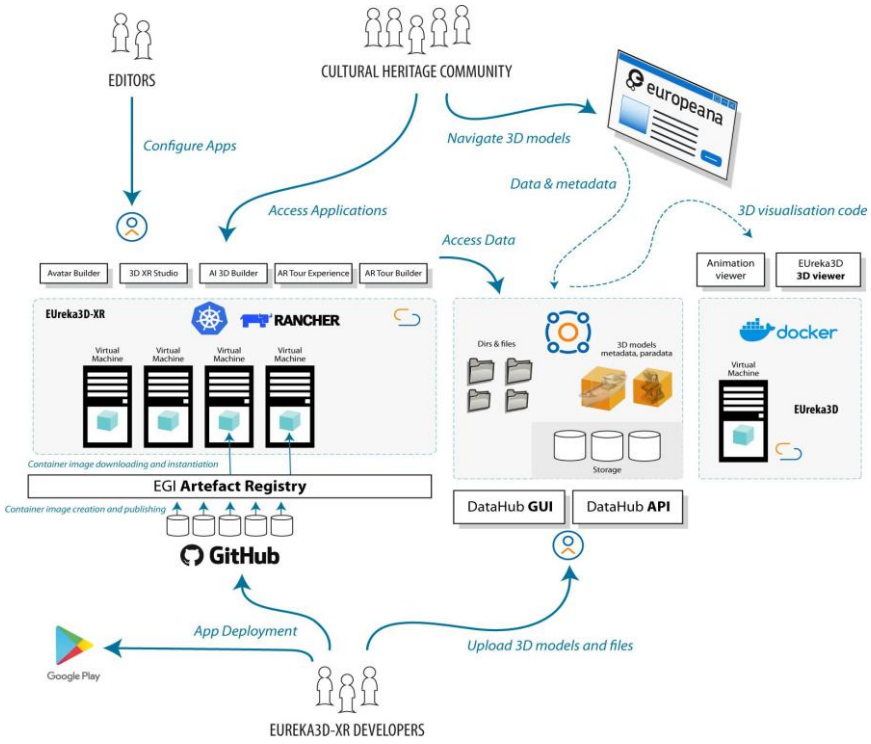

Figure 1.1: General view of the EUreka3D Data Hub ecosystem.

[1] https://eureka3d.eu/eureka3d-data-hub/ The user manual to access the EUreka3D Data Hub can be found at https://go.egi.eu/eureka3d_handbook
[2] https://eureka3d.eu/wp-content/uploads/2025/07/EUreka3D-XR-D3.1-Technical-Requirements.pdf
[3] https://eureka3d.eu/wp-content/uploads/2025/07/EUreka3D-XR-D2.1-Pilot-specification-and-planning.pdf

The infrastructure is a dynamic part of the system, which has to be rescaled according to the project's computational needs. The servers provide the computation layer required by the tools, which are packaged and deployed in *containers*. Containers offer a full ecosystem, but in essence allow the application of different Operating System features to pack software and all its dependencies together and run them in isolation.

A dynamic development environment is necessary to enhance the efficiency of the software development lifecycle. This is supported by Continuous Integration and Continuous Deployment (CI/CD) pipelines, which enable automation for rapid, iterative software delivery. Combined with the scalability and portability of containers, this approach enables reproducible builds and more efficient use of resources. Figure 1.2 illustrates the deployment process, from the time the developers make a software update until the time the software is "compiled" or packed and deployed in the final infrastructure.

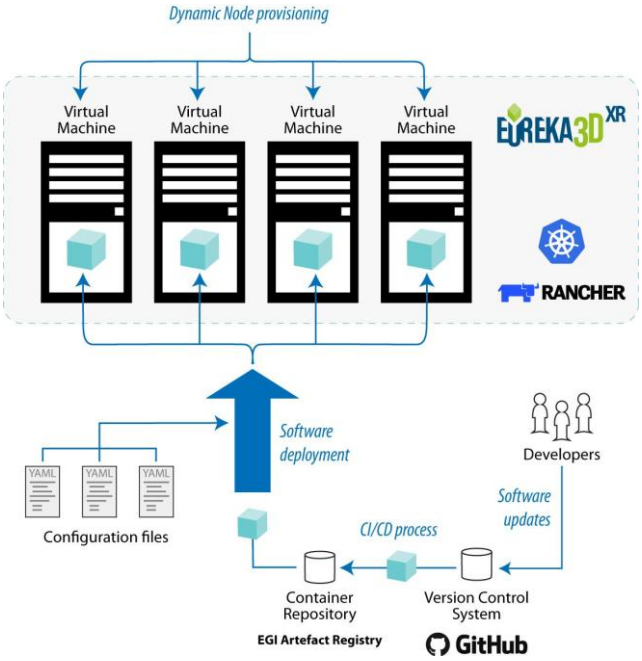

Figure 1.2: Continuous deployment process

Both the tools and the underlying infrastructure in which they run require secure access. Different measures like the authentication and authorisation provided by Check-in[4], the use of TLS certificates for Web applications and network access controls, provide some layers of security, which are essential to protect critical components, prevent unauthorised access, and maintain the integrity and confidentiality of the data.

[4] EGI Check-in is a solution for identity and access management. Detailed information can be found in EUreka3D's Deliverable D3.2 "The EUreka3D AAI architecture", available at https://eureka3d.eu/wp-content/uploads/2024/11/EUreka3D-D3.2-V1.0.pdf

2. Overview of the Initial Infrastructure

The initial beta infrastructure consists of the following main elements:

- The **authentication and authorisation infrastructure**, described in Section 2.1.
- The **tool publication process**, supported by the *source code* and the *EGI Artefact Repository*, discussed in Section 2.2.
- The **tool deployment process**, supported by the *EGI Artefact Repository* and the *EGI Cloud Container Compute*, explained in Section 2.3.

2.1 The Authentication and Authorisation Infrastructure

A system has been put in place to meet the requirements described in *D3.1 Technical Requirements*, mainly AAI-01, AAI-02, AAI-03 and AAI-04. Some groups have been created to facilitate the implementation of AAI-05 by the tools.

The organisation of the project's users is done through a **Virtual Organisation** (VO)[5], which also provides the basis for assigning user permissions and applying authorisation rules. EUreka3D-XR uses two different Virtual Organisations, as graphically shown in Figure 2.1:

- **culturalheritage.vo.egi.eu**, which is the VO used in EUreka3D to organise the different CHIs. This is still used for the data management conducted in EGI DataHub.
- **eureka3d-xr.vo.egi.eu**, which is a VO specific for the project that is mainly focused on permissions related to the infrastructure and the tools.

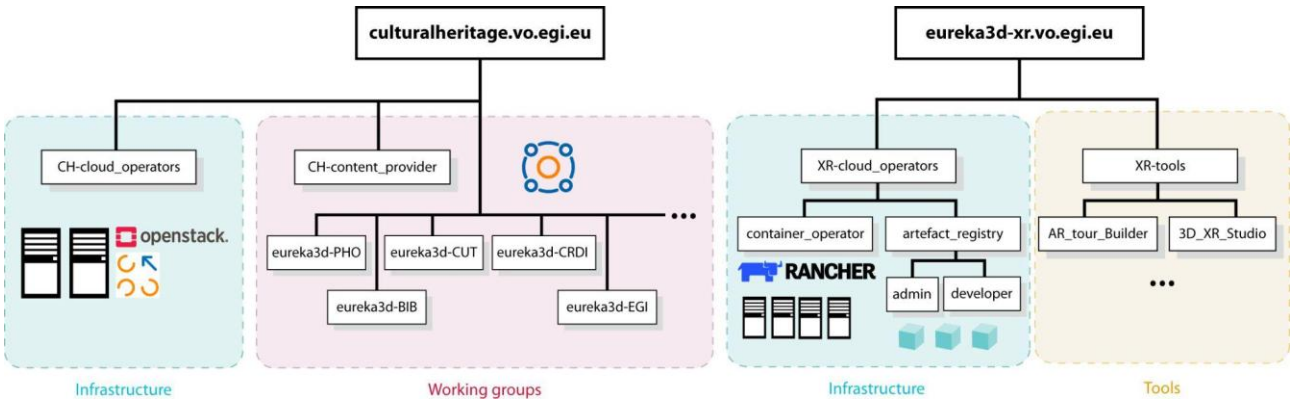

Figure 2.1: Virtual Organisations in EUreka3D-XR

EUreka3D-XR VO is mainly divided into the users that manage infrastructure components and the users related to the XR tools. Testing will decide whether this latter group should stay in the specific EUreka3D-XR VO or should be moved to the *culturalheritage* VO.

[5] https://docs.egi.eu/users/aai/check-in/vos/

2.2 TOOL PUBLICATION PROCESS

The tools are expected to run in *containers*, a technology that allows developers to run and pack software together with all their dependencies. These containers are created from container images that are published in a repository. In EUreka3D-XR, this repository is provided by the **EGI Artefact Registry**[6]. The artefacts remain private but accessible for authorised users.

In order to publish the artefacts, automation has been implemented through a CI/CD pipeline (see Figure 2.2). Developers use GitHub[7] (a version control system) to upload code, with each tool having its own dedicated code repository[8]. Once they commit code changes, the code is automatically "compiled" and packaged in a container image by the pipeline. Finally, this image is automatically published into the EGI Artefact Registry, which makes it available for deployment. The entire process is automatically triggered by changes in the tools' code, so that the latest changes are continuously integrated and deployed with minimal manual effort.

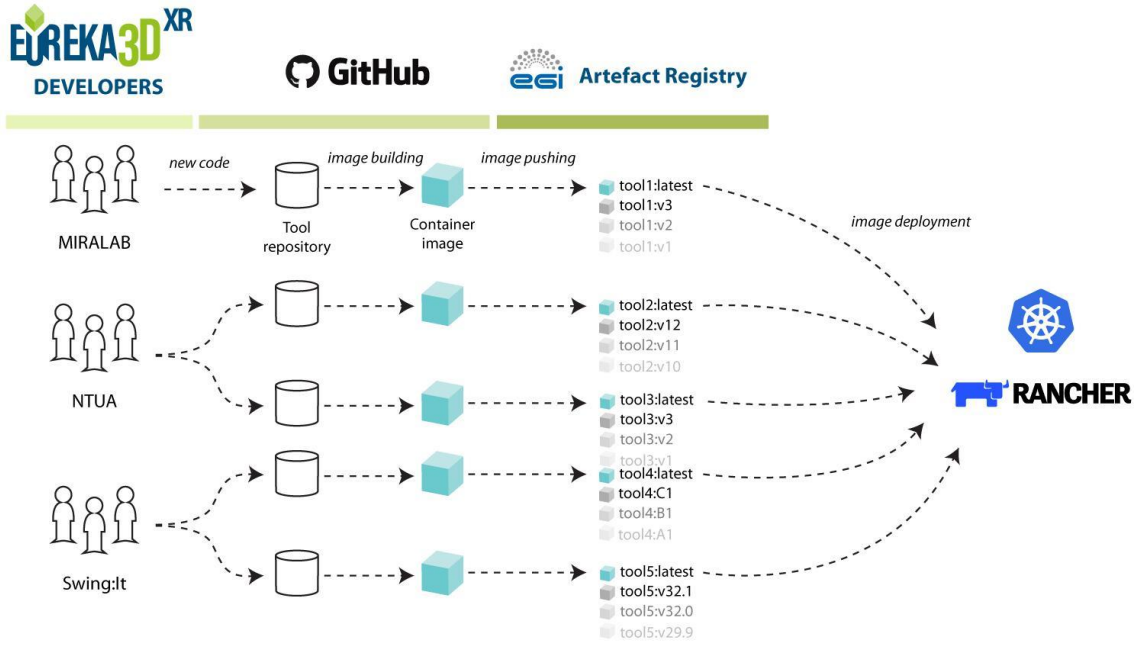

Figure 2.2: CI/CD pipeline from developers to deployment

2.3 TOOL DEPLOYMENT PROCESS

The deployment process depends on the images created during the publication process, but it is independent of it. The container images produced by the source code are deployed in *Kubernetes*, which is an application

[6] https://registry.egi.eu/
[7] https://github.com/EUreka3D-XR
[8] https://github.com/orgs/EUreka3D-XR/repositories

to run and manage containers. Kubernetes operates in clusters, which are a set of nodes or machines whose number can be dynamically adapted to application demand.

A Kubernetes cluster has been allocated in *Rancher*[9], part of the **EGI Cloud Container Compute**[10]. The initial cluster has 16 vCPU and 30 GB of memory divided into four nodes. This is intentionally low to serve as a starting point, in which no resources are wasted and the tools' deployment can be tested. The specifications will be increased in the coming weeks.

Two different environments have been foreseen for deployment: Staging and Production. Changes to the tools will be reflected in the EGI Artefact Repository, which can be detected by the cluster to trigger an automatic deployment in the Staging environment. The transition to Production is initially planned to be triggered manually.

The design of the deployment phase is one of the most demanding, as it involves various aspects that must be taken into account, such as the deployment and configuration of the tools in the different environments, the Dynamic DNS, the request and configuration of TLS certificates, the network routing, the provisioning of volumes and persistent storage, the security aspects, etc.

In summary, the deployment process has been carefully designed to ensure flexibility, efficiency, and scalability, implementing mechanisms to validate tool implementations in a controlled environment before moving to Production. This approach not only ensures the stability of the deployed tools but also provides a solid foundation for scaling and adapting future updates and functionalities as the project evolves.

[9] https://containers.egi.eu/
[10] https://www.egi.eu/service/cloud-container-compute/

CONCLUSIONS

This deliverable provides information about the initial (beta) version of the infrastructure and how it is used by the tools.

The infrastructure is illustrated in a showcase video, accessible at:
https://www.youtube.com/watch?v=UiTvOg4S078

This document offered context on the environment used in the project, including technologies such as Kubernetes, Rancher, containers, version control systems and so on. Instead of merely outlining the initial hardware specifications, the document has presented a comprehensive overview of the infrastructure by structuring it around three key pillars: the authentication and authorisation, the publication of software artefacts, and their subsequent deployment.

The beta infrastructure that has been implemented is the starting point for the project's fundamental support infrastructure. As it is a dynamic set of components, it will adapt to the requirements of the tools throughout the project's lifetime. The final infrastructure will be documented in the updated deliverable *D3.3 Cloud infrastructure final release* (due by M15).