

# EOSC Technical Specification

## Cloud IaaS Container Management

|                        |   |
|------------------------|---|
| <b>Technical Area:</b> | <b>Cloud compute, containers and orchestration</b>  |
| <b>Version:</b>        | 1.0   |
| <b>Status:</b>         | FINAL   |
| <b>Document Link:</b>  | <a href="https://confluence.egi.eu/pages/viewpage.action?pageId=52598373">https://confluence.egi.eu/pages/viewpage.action?pageId=52598373</a> |

### COPYRIGHT NOTICE



This work by Parties of the EOSC-hub Consortium is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). The EOSC-hub project is co-funded by the European Union Horizon 2020 programme under grant number 777536.

### DELIVERY SLIP



DSC-HUB RECEIVES FUNDING FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO. 777536.

| <i>Date</i>          | <i>Name</i>  | <i>Partner/Activity</i>         |
|----------------------|--|---------------------------------|
| <b>From:</b>         | Enol Fernandez   | EGI.eu                          |
| <b>Moderated by:</b> | Enol Fernandez   | EGI.eu                          |
| <b>Reviewed by:</b>  | Marica Antonacci<br>Marcin Plociennik<br>Mark Van Sanden<br>Marcus Hardt | INFN<br>PSNC<br>SurfSARA<br>KIT |
| <b>Approved by:</b>  | TCOM   |                                 |

## DOCUMENT LOG

| <i>Issue</i> | <i>Date</i> | <i>Comment</i>  | <i>Author</i>  |
|--------------|-------------|---|----------------|
| V1.0         | 2020-05-15  | First version taking comments from reviewers of draft | Enol Fernández |

## TERMINOLOGY

<https://wiki.eosc-hub.eu/display/EOSC/EOSC-hub+Glossary>

| <i>Terminology/Acronym</i> | <i>Definition</i> |
|----------------------------|-------------------|
|                            |                   |
|                            |                   |
|                            |                   |
|                            |                   |

## Table of Contents

|   |  |
|---|--|
| Introduction.....   | 2  |
| Adopted standards.....  | 2  |
| High-level Service Architecture.....                                    | 4  |
| Interoperability guidelines .....                                       | 4  |
| Technical interoperability guidelines .....                             | 4  |
| Policy interoperability guidelines .....                                | <b>Errore. Il segnalibro non è definito.</b> |
| Examples of solutions implementing this specification .....             | 5  |
| Procedure to integrate a service with the EOSC Hub <core service> ..... | 5  |

## Introduction

Services of Cloud IaaS Container Management provide on-demand API-based management of container-based applications. These services support the (Automated) Orchestration of container-

based applications which manage the deployment of a complete lifecycle of the containers that compose an application into a set of computing resources.

## Adopted standards

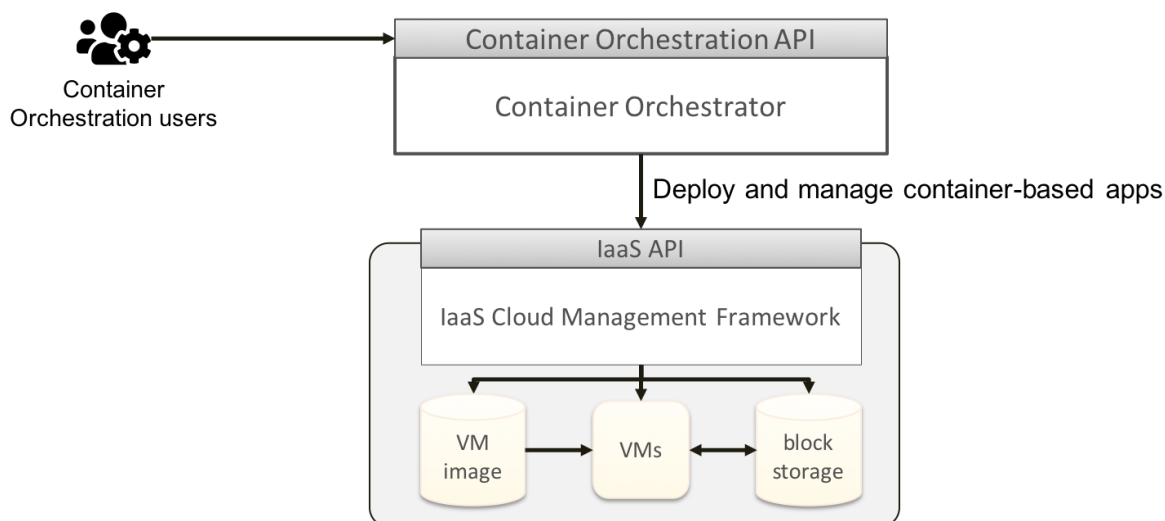
| Standard                       | Short description  | References                                   |
|--------------------------------|--|--|
| OCI                            | OCI contains two specifications: the Runtime Specification (runtime-spec) and the Image Specification (image-spec). The Runtime Specification outlines how to run a “filesystem bundle” that is unpacked on disk. At a high-level an OCI implementation would download an “OCI Image” then unpack that image into an “OCI Runtime filesystem” bundle. At this point the “OCI Runtime Bundle” would be run by an “OCI Runtime”. | <a href="#">OCI</a>                          |
| Singularity Image Format (SIF) | SIF is the image format used by Singularity  | <a href="#">SIF reference implementation</a> |

There are several container orchestration systems, each of them with their own not interoperable API. Kubernetes has gained a lot of momentum in the recent years and can be considered the main tool in this area. The table below lists some of these tools, but it’s not meant to be an exhaustive list.

| Protocol/API | Short description  | References                        |
|--------------|--|-----------------------------------|
| Kubernetes   | Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications   | <a href="#">kubernetes</a>        |
| Docker Swarm | Cluster management and orchestration features embedded in the Docker Engine  | <a href="#">Docker Swarm mode</a> |
| Mesos        | Apache Mesos abstracts CPU, memory, storage, and other compute resources away from machines (physical or virtual), enabling fault-tolerant and elastic distributed systems to easily be built and run effectively.             | <a href="#">Apache Mesos</a>      |
| AWS ECS      | Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container orchestration service that supports Docker containers and allows you to easily run and scale containerized applications on AWS. | <a href="#">AWS ECS</a>           |
| AWS Fargate  | AWS Fargate is a compute engine for Amazon ECS that allows you to run containers without having to manage servers or clusters.   | <a href="#">AWS Fargate</a>       |

## High-level Service Architecture

IaaS Container Orchestration services allow users to manage applications that are composed by containers. The container orchestrator manages a set of bare-metal or IaaS Cloud resources where the containers are scheduled. The system also manages the containers lifecycle, associated storage for containers, provides networking among the application containers and exposes those as services to external applications, and scales up and down the deployments as needed. Other features may be also provided.



## Interoperability guidelines

### Technical interoperability guidelines

Interoperable service in this category should:

- Support OCI image and runtime specs for container execution.
- Provide access to users authenticated with one of the EOSC-hub AAI federated identity protocols (OpenID Connect and/or SAML)

If the service can manage underlying IaaS resources automatically, it should support main IaaS VM Management systems in the EOSC-hub (OpenStack mostly)

There are several non-compatible container orchestrators available, there are no guidelines for the APIs of those currently.

## Examples of solutions implementing this specification

EOSC-hub services:

- [EGI Cloud Container Compute](#)

Other services:

- Kubernetes based: [AWS EKS](#), [GCP GKE](#), [Azure AKS](#)
- Other: [AWS ECS](#), [AWS Fargate](#)

## Procedure to integrate a service with the EOSC Hub Cloud Container Compute

Most container management systems can be deployed on any IaaS system with a commonly supported API. There is no specific procedure for the integration besides providing these IaaS APIs accessible. The EGI Cloud Compute providers is an example of IaaS federation that can support container management system. Documentation on joining the EGI Cloud Compute is available at <https://docs.egi.eu/providers/fedcloud/>.