ETSI GS NFV-SOL 018 V4.3.1 (2022-09)

Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; Profiling specification of protocol and data model solutions for OS Container management and orchestration

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/NFV-SOL018

Keywords
container, NFV, orchestration, protocol, service, virtualisation

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document provides a mapping of the NFV object model for OS container management and orchestration to managed objects of Kubernetes® and Helm™ as specified by the CNCF® along with a specification of a mapping between a common set of input parameters (e.g. derived from VNFD/NSD and/or NFV-MANO RESTful APIs) and output parameters associated to the management and orchestration of the referred managed objects. It profiles the reference Kubernetes® API as NFV protocol and data model solution for OS container management and orchestration. It profiles the reference Helm™ documentation as NFV protocol and data model solution for management of OS container workload based on an MCIOP. It profiles the reference OCI™ Distribution Specification API (which is based on the Docker™ registry API) as NFV protocol and data model solution for OS container image management. The latest published versions of the reference Kubernetes® API, Helm™ documentation and OCI™ Distribution Specification API are profiled against the requirements on the functions and the management service interfaces of the Container Infrastructure Service Management (CISM) and Container Image Registry (CIR) functions as specified in ETSI GS NFV-IFA 040 [2] and ETSI GS NFV-IFA 010 [1].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Functional requirements specification".

[2] ETSI GS NFV-IFA 040: "Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification".

[3] Kubernetes® API v1.24.

NOTE: Available at https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.24/.

[4] Helm™ CLI v3.9.0.

NOTE: Available at https://helm.sh/docs/helm/helm/.

[5] OCI™ Distribution Specification v1.0.0.

NOTE: Available at https://github.com/opencontainers/distribution-spec/releases/tag/v1.0.0.

[6] OCI™ Image Format Specification v1.0.1.

NOTE: Available at https://github.com/opencontainers/image-spec/releases/tag/v1.0.1.

[7] Helm™ charts v3.9.0.

NOTE: Available at https://helm.sh/docs/topics/charts/.

[8] Kubernetes® reference documentation, API Access Control.

NOTE: Available at https://kubernetes.io/docs/reference/access-authn-authz/.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GR NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.2] ETSI GS NFV-SOL 003: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[i.3] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; NFV descriptors based on TOSCA specification".

[i.4] ETSI NFV Release Documentation.

NOTE: Available at https://docbox.etsi.org/ISG/NFV/Open/Other/ReleaseDocumentation.

[i.5] ETSI GS NFV-SOL 013: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".

[i.6] IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

NOTE: Available at https://tools.ietf.org/html/rfc7235.

[i.7] ETSI GS NFV-SOL 002: "Network Functions Virtualisation (NFV) Release 4; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point".

[i.8] Open Container Initiative Charter v1.3.

NOTE: Available at https://github.com/opencontainers/tob/blob/main/CHARTER.md.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GR NFV 003 [i.1] and the following apply:

Compute MCIO: MCIO which declarative descriptor specifies compute infrastructure resource requests

Network MCIO: MCIO which declarative descriptor specifies network infrastructure resource requests

Storage MCIO: MCIO which declarative descriptor specifies storage infrastructure resource requests

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GR NFV 003 [i.1] apply.

4 Overview of protocols and data models for OS Container management and orchestration

4.1 Summary of ETSI GS NFV-IFA 040

ETSI GS NFV-IFA 040 [2] specifies the requirements on the following services to be provided by the CISM, and the requirements on the management services interfaces to expose these services to other NFV-MANO functional entities and/or external entities outside NFV-MANO:

1) OS container workload management service
2) OS container compute management service
3) OS container storage management service
4) OS container network management service
5) OS container configuration management service

The OS container workload management service interface is produced by the CISM to enable consumers to request lifecycle management operations on containerized workloads based on a MCIOP and to query information on containerized workloads based on a MCIOP.

The OS container compute/storage/network management service interfaces are produced by the CISM to enable consumers to request management operations on Compute/Storage/Network MCIOs and to query information on Compute/Storage/Network MCIOs.

The OS container configuration management service interface is produced by the CISM to enable consumers to request management operations on MCIO configurations and policies for MCIOs and to query information on those respectively. The OS container configuration management service interface also enables consumers to request management operations on namespaces and namespace quota.

ETSI GS NFV-IFA 040 [2] also specifies the requirements on the OS container image management service to be provided by the CIR, and the requirements on the OS container image management service interface to be exposed to other NFV-MANO functional entities and/or external entities outside NFV-MANO. The OS container image management service interface is produced by the CIR to enable consumers to request adding and deleting OS container images to/from the CIR and to query information about OS container images in the CIR.

4.2 Profiled protocol and data model solutions

4.2.1 Kubernetes® API

4.2.1.1 Introduction

This clause provides an overview over the Kubernetes® API [3] which is profiled to the requirements on the OS container management service interfaces exposed by the CISM as specified in ETSI GS NFV-IFA 040 [2]. The overview covers the high level API structure concerning the grouping of the managed resource objects into resource categories as well as the generic concepts for the data model of the managed resource objects.

4.2.1.2 API structure

The Kubernetes® API [3] managed objects represent a concrete instance of a resource type on the CIS cluster. Kubernetes® leverages standard RESTful terminology to describe the API concepts:

- A **resource type** is the name used in the URL.
- All resource types have a representation in JSON (their object schema) which is called a **kind.**
- A list of instances of a resource type is known as a **collection.**

- A single instance of a resource type is called a **resource,** and also usually represents an **object.**

All resource types are either scoped by the CIS cluster (e.g. `/apis/GROUP/VERSION/*`) or to a namespace (e.g. `/apis/GROUP/VERSION/namespaces/NAMESPACE/*`).

Standard HTTP methods POST, PUT, PATCH, and DELETE support single resources only. These methods with single resource support have no support for submitting multiple resources together in an ordered or unordered list or transaction.

The Kubernetes® API [3] supports read and write operations on the Kubernetes® resource objects via a Kubernetes® API endpoint. Kubernetes® differentiates the following categories of resource objects managed via their APIs:

- **Workloads:** objects used to manage and run OS containers on the CIS cluster.
- **Discovery & Loadbalancing:** objects used to inter-connect the workloads into externally accessible, load-balanced services.
- **Configuration & Storage:** objects used to inject initialization data into the containerized applications, and to persist data that is external to the OS containers.
- **Cluster:** objects define how the CIS cluster itself is configured.
- **Metadata:** objects used to configure the behaviour of other resources within the CIS cluster.

A mapping of the individual Kubernetes® managed resource objects to the NFV object model is provided in clause 5 of the present document.

4.2.1.3 Data model concepts

The Kubernetes® resource objects are modelled with individual object schemas. All resource objects typically have 3 components:

- **Resource ObjectMeta:** The metadata about the resource object, such as its name, type, API version, annotations, and labels. This schema, which is common to all resource types, contains fields that may be updated both by the external user and the CIS system.
- **ResourceSpec:** Defined by the external user and describes the desired state of the system concerning the resource object. Specified when creating or modifying a resource object is requested.
- **ResourceStatus:** Provided by the CIS system and represents the current state of the system concerning the resource object.

4.2.2 Helm™ CLI

4.2.2.1 Introduction

This clause provides an overview over the Helm™ CLI [4] which is profiled to the requirements on the OS container workload management service interface exposed by the CISM as specified in ETSI GS NFV-IFA 040 [2]. Helm™ is a tool for managing OS container workloads deployed on Kubernetes® CIS clusters based on MCIOPs called Helm™ charts. The overview covers the high level CLI structure concerning the main operations as well as the generic concepts for the data model of the MCIOP and the managed runtime objects.

For Helm™, there are three important concepts:

1) The **Helm™ chart** is a bundle of information necessary to create an instance of an OS container workload deployed on Kubernetes® CIS clusters.
2) The **Helm™ config** contains configuration information that can be merged into a packaged Helm™ chart to create a releasable object.
3) A **Helm™ release** is a running instance of an OS container workload based on a Helm™ chart, combined with a specific config.

4.2.2.2 CLI structure

The Helm™ CLI provides commands for the following, common actions:

- Install OS container workloads based on Helm™ charts
- Get information on existing Helm™ releases and their runtime details
- Upgrade a Helm™ release to a new version of a Helm™ chart
- Roll-back a Helm™ release to a previous Helm™ release version
- Uninstall a Helm™ release of a Helm™ chart, removing all of the resources associated with the last Helm™ release
- Display the status information of a Helm™ release

All Helm™ CLI commands do have the generic synopsis:

```
helm [COMMAND] [RELEASE] [CHART] [flags]
```

Variations and details for the specific syntaxes of the commands are described in the corresponding profiling clauses of the present document.

4.2.2.3 Data model concepts

4.2.2.3.1 Helm™ chart file structure

A Helm™ chart is organized as a collection of files inside of a directory. The directory name is the name of the chart (without versioning information). Inside of this directory, Helm™ will expect a structure that matches this:

```
chartname/
  Chart.yaml                  # A YAML file containing information about the Helm™ chart
  LICENSE                     # A plain text file containing the license for the
                              # Helm™ chart
  README.md                   # A human-readable README file
  values.yaml                 # The default configuration values for this Helm™ chart
  values.schema.json          # A JSON Schema for imposing a structure on the
                              # values.yaml file
  charts/                     # A directory containing any Helm™ charts upon which this
                              # Helm™ chart depends.
  crds/                       # Custom Resource Definitions
  templates/                  # A directory of templates that, when combined with values,
                              # will generate valid Kubernetes® manifest files.
  templates/NOTES.txt         # A plain text file containing short usage notes
```

Helm™ chart reserves the use of the `charts/`, `crds/`, and `templates/` directories, and of the listed file names. Other files will be left as they are.

NOTE: The intent of this list is to provide an overview of the Helm™ chart structure. It is not intended to convey a mandatory or optional presence of each of the structure elements.

4.2.2.3.2 Helm™ release objects

The Helm™ release object is one of the built-in objects of Helm™. The data model describing the properties of a Helm™ release consists of the following main elements:

- **Name**: the name of the Helm™ release
- **Info:** the deployment dates, status information and notes associated to a Helm™ release
- **Chart:** the Helm™ chart that the Helm™ release is based upon
- **Config:** the set of extra values that have been added to the Helm™ chart
- **Manifest:** the string representation of the rendered Helm™ template

- **Hooks:** all of the hooks declared for this Helm™ release
- **Version:** the revision of the Helm™ release
- **Namespace:** the Kubernetes® namespace of the Helm™ release
- **Labels:** the labels of the Helm™ release

4.2.3 OCI™ Distribution Specification API

4.2.3.1 Introduction

This clause provides an overview over the OCI™ Distribution Specification [5] which is profiled to the requirements on the OS container image management service interface exposed by the CIR as specified in ETSI GS NFV-IFA 040 [2]. The OCI™ Distribution Specification is based on the earlier published Docker™ Registry HTTP API v2. The overview covers the high-level API structure as well as the generic concepts for the data model of the managed resource objects.

4.2.3.2 API structure

The OCI™ Distribution Specification [5] is the protocol to facilitate the management of OS container images which are stored in a CIR. The API endpoints are prefixed by the API version and the repository name:

```
/{VERSION}/{repository_name}/
```

Additional information on the utilization of the repository name is provided in clause B.2.1 of the present document.

The OCI™ Distribution Specification [5] supports read and write operations on the CIR resource objects via the OCI™ Distribution Specification API endpoints. The OCI™ Distribution Specification differentiates the following resource objects managed via their resource endpoints:

- **tag:** A custom identifier of an OS container image
- **manifest:** a JSON document which specifies an artifact of an OS container image
- **blob:** the binary form of content that is stored by a registry, addressable by a digest

4.2.3.3 Data model concepts

The artifacts to be managed via the OCI™ Distribution Specification [5] are components of an OCI™ image, which are specified in the OCI™ Image Format Specification [6]. An OCI™ image consists of the following components:

- **Image Manifest:** a document describing the components that make up a OS container image
- **Image Index:** an annotated index of image manifests
- **Filesystem Layer:** a changeset that describes a OS container's filesystem
- **Image Configuration:** a document determining layer ordering and configuration of the OS container image suitable for translation into a OS container runtime bundle.

5 NFV object model mapping to profiled solution objects

5.1 Managed Container Infrastructure Objects

5.1.1 Compute MCIOs

Selected Kubernetes® resource objects of the Workloads category are identified to map to the Compute MCIO type of the NFV object model, see clauses 5.2.1 and 6.4 in ETSI GS NFV-IFA 040 [2]. Table 5.1.1-1 lists the Kubernetes® resource objects which are mapped to the NFV objects of the Compute MCIO type.

Table 5.1.1-1: Kubernetes® resource objects mapped to NFV objects of Compute MCIO type

Kubernetes® resource object kind	Kubernetes® resource URI	Kubernetes® resource object description
CronJob	/apis/batch/v1/namespaces/{namespace}/cronjobs	Represents a recurring task that runs to completion and then stops.
DaemonSet	/apis/apps/v1/namespaces/{namespace}/daemonsets	Defines a set of Pods that provide local facilities of CIS cluster nodes.
Deployment	/apis/apps/v1/namespaces/{namespace}/deployments	Represents a stateless application workload.
Job	/apis/batch/v1/namespaces/{namespace}/jobs	Represents a one-off task that runs to completion and then stops.
Pod	/api/v1/namespaces/{namespace}/pods	Represents the smallest deployable unit of an application workload as a group of one or more OS containers.
ReplicaSet	/apis/apps/v1/namespaces/{namespace}/replicasets	Represents a stable set of stateless application workloads.
StatefulSet	/apis/apps/v1/namespaces/{namespace}/statefulsets	Represents a stateful application workload.

NOTE: The Kubernetes® "Container" resource object is not identified and mapped as individual Compute MCIO because it is always managed within the context of a Kubernetes® "Pod" resource object.

5.1.2 Storage MCIOs

Selected Kubernetes® resource objects of the Storage category are identified to map to the Storage MCIO type of the NFV object model, see clauses 5.2.1 and 6.5 in ETSI GS NFV-IFA 040 [2]. Table 5.1.2-1 lists the Kubernetes® resource objects which are mapped to the NFV objects of the Storage MCIO type.

Table 5.1.2-1: Kubernetes® resource objects mapped to NFV objects of Storage MCIO type

Kubernetes® resource object kind	Kubernetes® resource URI	Kubernetes® resource object description
PersistentVolumeClaim	/api/v1/namespaces/{namespace}/persistentvolumeclaims	Represents requests for persistent storage resources.

5.1.3 Network MCIOs

Selected Kubernetes® resource objects of the Discovery & Loadbalancing category are identified to map to the Network MCIO type of the NFV object model, see clauses 5.2.1 and 6.6 in ETSI GS NFV-IFA 040 [2]. Table 5.1.3-1 lists the Kubernetes® resource objects which are mapped to the NFV objects of the Network MCIO type.

Table 5.1.3-1: Kubernetes® resource objects mapped to NFV objects of Network MCIO type

Kubernetes® resource object kind	Kubernetes® resource URI	Kubernetes® resource object description
Endpoints	/api/v1/namespaces/{namespace}/endpoints	Represent a group of network addresses with a common set of ports.
EndpointSlice	/apis/discovery.k8s.io/v1/namespaces/{namespace}/endpointslices	References to a set of network endpoints.
Ingress	/apis/networking.k8s.io/v1/namespaces/{namespace}/ingresses	Represents an external access routed to one or more Service resource objects.
Service	/api/v1/namespaces/{namespace}/services	Represents a stable IP endpoint loadbalanced across multiple application workload replicas.

5.1.4 MCIO configurations

Selected Kubernetes® resource objects of the Configuration and Metadata categories are identified to map to the MCIO configurations of the NFV object model, see clauses 5.2.1 and 6.7 in ETSI GS NFV-IFA 040 [2]. Table 5.1.4-1 lists the Kubernetes® resource objects which are mapped to the NFV objects of MCIO configurations.

Table 5.1.4-1: Kubernetes® resource objects mapped to NFV objects of MCIO configurations

Kubernetes® resource object kind	Kubernetes® resource URI	Kubernetes® resource object description
ConfigMap	/api/v1/namespaces/{namespace}/configmaps	Stores non-confidential configuration data for application workloads.
Secret	/api/v1/namespaces/{namespace}/secrets	Stores confidential configuration data for application workloads.
CustomResourceDefinition	/apis/apiextensions.k8s.io/v1/customresourcedefinitions	Defines custom resources as API extensions.

5.1.5 MCIO policies

Selected Kubernetes® resource objects of the Metadata and Cluster categories are identified to map to the MCIO policies of the NFV object model, see clauses 5.2.1 and 6.7 in ETSI GS NFV-IFA 040 [2]. Table 5.1.5-1 lists the Kubernetes® resource objects which are mapped to the NFV objects of MCIO policies.

Table 5.1.5-1: Kubernetes® resource objects mapped to NFV objects of MCIO policies

Kubernetes® resource object kind	Kubernetes® resource URI	Kubernetes® resource object description
PodDisruptionBudget	/apis/policy/v1/namespaces/{namespace}/poddisruptionbudgets	Policy to control the number of Pods of a workload application allowed to be evicted.
NetworkPolicy	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies	Policy to control the traffic flow to/from workload applications.

5.2 Managed Container Infrastructure Object Packages

The Helm™ chart [7] is identified to map to the Managed Container Infrastructure Object Package of the NFV object model, see clauses 5.2.2 and 6.3 in ETSI GS NFV-IFA 040 [2].

5.3 Namespace

The Kubernetes® resource object Namespace of the Cluster category is identified to map to the Namespace of the NFV object model, see clauses 5.2.3 and 6.7 in ETSI GS NFV-IFA 040 [2]. Table 5.3-1 lists the Kubernetes® resource object which is mapped to the NFV objects of the Namespace type.

Table 5.3-1: Kubernetes® resource object mapped to NFV objects of Namespace type

Kubernetes® resource object kind	Kubernetes® resource URI	Kubernetes® resource object description
Namespace	/api/v1/namespaces	Represents the logical grouping of resources, identifiers, policies and authorizations.

5.4 Namespace quota

The Kubernetes® resource object ResourceQuota of the Cluster category is identified to map to the Namespace quota of the NFV object model, see clauses 5.2.4 and 6.7 in ETSI GS NFV-IFA 040 [2]. Table 5.4-1 lists the Kubernetes® resource object which is mapped to the NFV objects of the Namespace quota type.

Table 5.4-1: Kubernetes® resource object mapped to NFV objects of Namespace quota type

Kubernetes® resource object kind	Kubernetes® resource URI	Kubernetes® resource object description
ResourceQuota	/api/v1/namespaces/{namespace}/resourcequotas	Represents constraints that limit aggregate resource consumption per namespace.

5.5 OS container image

The OCI™ Image [6] is identified to map to the OS container image of the NFV object model, see clauses 5.2.5 and 7.3 in ETSI GS NFV-IFA 040 [2].

6 Input/Output parameter mapping between NFV data model and profiled solution data models

6.1 Introduction

The CIS and CISM have the capability to perform the necessary OS container infrastructure resources management by using declarative descriptors of MCIOs [2]. These descriptors are included in MCIOPs and used by the CISM to perform the orchestration and lifecycle management of a CIS cluster resources when allocating, updating, querying or terminating OS container infrastructure resources for the MCIOs realizing the VNF.

NOTE: The use of MCIOP enables the consumer of the CISM to perform lifecycle management of a container infrastructure objects in a declarative manner.

For realizing the lifecycle management of MCIOs, the interactions between the consumer and the producer of OS container workload management services (i.e. the CISM) are based on:

- interface operations;
- MCIOPs; and
- values assignments to specified input and output parameters.

Figure 6.1-1 illustrates the concept of interoperability between the CISM and its consumer based on MCIOPs. Interface operations (refer to point (1) API call in Figure 6.1-1) enable the consumer to request actions to fulfil the management capabilities specified in clause 6.3 of ETSI GS NFV-IFA 040 [2].

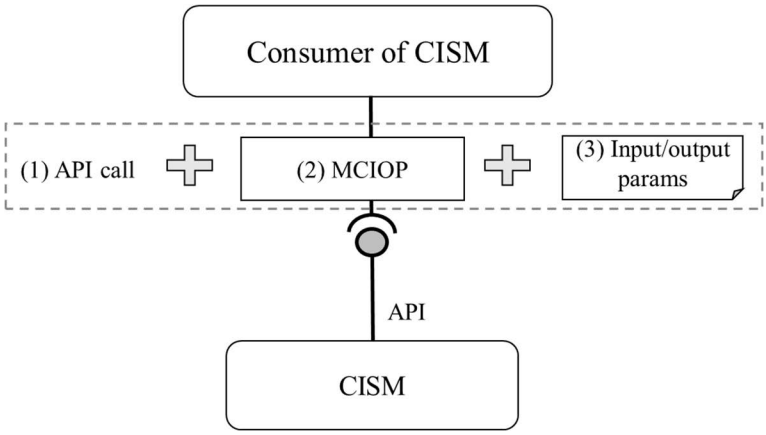

Figure 6.1-1: Concept of CISM OS container infrastructure resources management based on MCIOP

The MCIOPs (refer to point (2) in Figure 6.1-1) associate parameters and requirements to different types of MCIOs. The information in these MCIOs and their composition, together with the input/output parameters, determine how the containerized VNF (or the MCIOP) is deployed. The lifecycle management of MCIOs is executed by the CISM by orchestrating more granular management of OS container infrastructure resources based on the information contained in the MCIOP and additional value assignments provided as input/output parameters.

The OS container workload management services process the corresponding input parameter values that are provided by the consumer (see point (3) in Figure 6.1-1). Likewise, the OS container workload management services return output parameters as a result of the deployment of MCIO according to the requested interface operation. The input and output parameter to be signalled on the interface of the CISM shall be compliant to the input and output parameter mapping principles as specified in the present document.

The input and output data may be provided as:

- arguments or payload in the API calls;
- defined in configuration/value files, which are passed as arguments or payload in the API calls; or
- values that are fixed (i.e. design-time value) in MCIOPs.

6.2 Input parameters to CISM APIs

6.2.1 Framework

6.2.1.1 Introduction

This clause describes a framework to specify the handling of input parameters to APIs which are exposed by a CISM.

The specification concerning input parameters to the CISM API is stipulated based on the two following regards:

- mapping to elements of the NFV data models (i.e. VNFD data model and/or data model used in ETSI GS NFV-SOL 002 [i.7] and ETSI GS NFV-SOL 003 [i.2] interfaces); and

- conveying parameters values.

The first one is to specify mapping between attributes of the CISM APIs and the NFV data models. The second one is to specify how to convey the attribute values to the CISM APIs.

The CISM is required to expose management service interfaces on different abstraction levels. One abstraction level are the MCIOPs, the other abstraction level are the MCIOs. The profiled solutions of the Helm™ CLI and the Kubernetes® API are based on the same Kubernetes® object model, while their interfaces expose different methods to manage it. Therefore, conveying the parameter values is specified separately from mapping to the NFV data models.

6.2.1.2 Mapping to NFV data models

Table 6.2.1.2-1 describes the principle of mapping API objects to NFV data models. Regarding respective API objects, similar tables are described. The first row of the table describes a target API object and the subsequent rows describe fields of the target API object. The symbol ">" is used to represent the structure of API objects and fields. In case of the second row of the table, it is described that the field corresponding to the row belongs to the API object described in the first row.

Table 6.2.1.2-1: Principles of CISM API object parameter mapping to NFV data models

API object or field	Mapped NFV data element/attribute	Description
{Kind Version Group}	(Not applicable)	Indicates a target API object, e.g. "Container v1 core" in the column "API object or field". In this description, parent API objects of the target API object are listed. For instance: • "PodSpec v1 core"
>{Field}	{attribute in information element}	Indicates a field of the target API object, e.g. "image" in the column "API object or field". In the column "Mapped NFV data element/attribute", it is specified what NFV data model is mapped to the field, e.g. "swImage in SwImageDesc". In this description, information related to the field is explained. All fields of the target API object are not necessarily described.

6.2.1.3 Conveying parameters

In case of using Helm™ for the OS container workload management service interface by the CISM, there are two ways to convey input parameters to the CISM APIs. Run-time information is conveyed by overriding the default values of the "values.yaml" configuration of the Helm™ chart and design-time information is conveyed by configuration files included in the Helm™ charts (see clause 4.2.2.3.1).

In case of using the Kubernetes® API for the OS container management service interfaces exposed by the CISM, the way to convey input parameters to the CISM APIs is that the CISM API consumer invokes the REST API calls with JSON based payload which is converted from API object or field of Kubernetes®. The API structure is described in clause 4.2.1.2.

6.2.2 Workloads

6.2.2.1 Mapping to NFV data models

This clause describes the mapping regarding Workloads, which is categorized in clause 4.2.1.2.

Table 6.2.2.1-1 indicates CISM API object parameter mapping to NFV data models related to Deployment.

Table 6.2.2.1-1: CISM API object parameter mapping to NFV data models related to Deployment

API object or field	Mapped NFV data element/attribute	Description
Deployment v1 apps	(Not applicable)	A resource object to deploy VNFC instances.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Deployment. The subsequent name and namespace are fields of the child API object.
>>name	"name" in Vdu.OsContainerDeployableUnit (ETSI GS NFV-SOL 001 [i.3], clause 6.8.13)	Indicates name of the Deployment.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the Deployment is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>spec	(Not applicable)	Describes specification of the desired behaviour of the Deployment. The field corresponds to a child API object of the Deployment, and the child API object is specified by DeploymentSpec v1 apps.

Table 6.2.2.1-2 indicates CISM API object parameter mapping to NFV data models related to DeploymentSpec.

Table 6.2.2.1-2: CISM API object parameter mapping to NFV data models related to DeploymentSpec

API object or field	Mapped NFV data element/attribute	Description
DeploymentSpec v1 apps	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • Deployment v1 apps
>replicas	"number_of_instances" in VduLevel (ETSI GS NFV-SOL 001 [i.3], clause 6.2.19) of levels in VduInstantiationLevels (ETSI GS NFV-SOL 001 [i.3], clause 6.10.2) used in VNF instantiation and scale to level operations.	Indicates the number of desired VNFC instances to be instantiated or after performing Scale VNF to Level operation within a deployment flavour.
	Equal to a numerical value computed as current replica value plus (in case of scaling out) or minus (in case of scaling in) the "number_of_instances" in VduLevel (ETSI GS NFV-SOL 001 [i.3], clause 6.2.19) of deltas in VduScalingAspecDeltas (ETSI GS NFV-SOL 001 [i.3], clause 6.10.6)	Indicates the number of desired VNFC instances after performing Scale VNF operation within a deployment flavour.
>template	(Not applicable)	Describes the VNFC instances that will be created. The field corresponds to a child API object of the DeploymentSpec, and the child API object is specified by PodTemplateSpec v1 core.

Table 6.2.2.1-3 indicates CISM API object parameter mapping to NFV data models related to PodTemplateSpec.

Table 6.2.2.1-3: CISM API object parameter mapping to NFV data models related to PodTemplateSpec

API object or field	Mapped NFV data element/attribute	Description
PodTemplateSpec v1 core	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following ones: • DeploymentSpec v1 apps • StatefulSetSpec v1 apps
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the PodTemplateSpec. The subsequent namespace and annotations are fields of the child API object. See note.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the VNFC instance is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>>annotations	(Not applicable)	Describes unstructured key value maps stored with a resource that may be set by external tools to store and retrieve arbitrary metadata.
>>> k8s.cni.cncf.io/networks	resourceId in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates a list of identifiers of NetworkAttachmentDefinition representing secondary container cluster external networks which are attached to a Pod deployed based on the PodTemplateSpec.
>spec	(Not applicable)	Describes specification of the desired behaviour of the VNFC instance. The field corresponds to a child API object of the PodTemplateSpec, and the child API object is specified by PodSpec v1 core.
NOTE: The name is automatically generated based on name of Deployment or StatefulSet and thereby there is no need to specify the mapping to NFV data models.		

Table 6.2.2.1-4 indicates CISM API object parameter mapping to NFV data models related to PodSpec.

Table 6.2.2.1-4: CISM API object parameter mapping to NFV data models related to PodSpec

API object or field	Mapped NFV data element/attribute	Description
PodSpec v1 core	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • PodTemplateSpec v1 core
> affinity	(Not applicable)	Describes the Pod scheduling constraints. The field corresponds to a child API object of the "PodSpec", and the child API object is specified by "Affinity v1 core".
>> nodeAffinity. requiredDuringSchedulingIgnoredDuringExecution. nodeSelectorTerms. matchExpressions	(Not applicable)	Describes CIS node selector matching expressions, specifying the CIS node affinity scheduling rules for the Pod. The field corresponds to a nested child API object of the "Affinity", and the nested child API object is specified by "NodeSelectorRequirement v1 core". The subsequent "key" and "values" are fields of the nested child API object. See note 1 and note 2.
>>> key	Key sub-string from value of mcioConstraints in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Specifies the CIS node label key that the selector applies to.
>>> values	Value sub-string from value of mcioConstraints in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Specifies the CIS node label values that the selector applies to.
>> podAffinity. requiredDuringSchedulingIgnoredDuringExecution.podAffinityTerm	(Not applicable)	Describes Pod affinity scheduling rules. The field corresponds to a nested child API object of the "Affinity", and the nested child API object is specified by "PodAffinityTerm v1 core". The subsequent "topologyKey" is a field of the nested child API object.
>>> topologyKey	Value of mcioConstraints in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Specifies the CIS node label key where the Pod can be co-located according to the affinity rules.
>> podAntiAffinity. requiredDuringSchedulingIgnoredDuringExecution.podAffinityTerm	(Not applicable)	Describes Pod anti-affinity scheduling rules. The field corresponds to a nested child API object of the "Affinity", and the nested child API object is specified by "PodAffinityTerm v1 core". The subsequent "topologyKey" is a field of the nested child API object.
>>> topologyKey	Value of mcioConstraints in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Specifies the CIS node label key where the Pod can not be co-located according to the anti-affinity rules.
>containers	List of Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Describes list of OS containers belonging to the VNFC instance. The field corresponds to a child API object of the "PodSpec", and the child API object is specified by "Container v1 core".

NOTE 1: The "NodeAffinity v1 core" API object includes the placement control capability provided by the "nodeSelector" API field. Therefore, there is no need to map the "nodeSelector" API field to a NFV data element/attribute.

NOTE 2: In case the GrantInfo for the ResourceDefinition associated to a PodSpec contains more than one mcioContstraints element, each mcioConstraint element is mapped to one "matchExpressions" API field element.

Table 6.2.2.1-5 indicates CISM API object parameter mapping to NFV data models related to Container.

Table 6.2.2.1-5: CISM API object parameter mapping to NFV data models related to Container

API object or field	Mapped NFV data element/attribute	Description
Container v1 core	Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	The API object is not directly utilized and is only called as a child API object of the following one: • PodSpec v1 core
>image	Compound value of the followings: – cirConnectionInfo in Grant (ETSI GS NFV-SOL 003 [i.2], clause 9.5.2.3) – name in SwImage (ETSI GS NFV-SOL 001 [i.3], clause 6.3.1) – version in SwImage (ETSI GS NFV-SOL 001 [i.3], clause 6.3.1) (Clause B.2.1 in the present document provides additional explanation on how to consist of the above information.)	Indicates name of an OS container image.
>name	name in Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Indicate name of an OS container.
>resources	(Not applicable)	Describes compute resources required by this OS container. It cannot be updated. The field corresponds to a child API object of the "Container". The subsequent "limits" and "requests" are fields of the child API object.
>>limits	(Not applicable)	Describes the maximum amount of compute resources allowed for this OS container.
>>>cpu	cpu_resource_limit in Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Indicates the maximum amount of cpu allowed for this OS container.
>>>memory	memory_resource_limit in Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Indicates the maximum amount of memory allowed for this OS container.
>>>ephemeral-storage	ephemeral_storage_resource_limit in Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Indicates the maximum amount of local ephemeral storage allowed for this OS container.
>>>hugepages-<size>	requested_size in HugePages (ETSI GS NFV-SOL 001 [i.3], clause 6.2.71) <size>, which is part of the field, is equal to hugepage_size in HugePages (ETSI GS NFV-SOL 001 [i.3], clause 6.2.71)	Indicates the total of the hugepages requested for this OS container, which the OS container can maximally use. The size for the hugepages is indicated by <size>.
>>requests	(Not applicable)	Describes the minimum amount of compute resources required for this OS container.
>>>cpu	requested_cpu_resource in Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Indicates the minimum amount of cpu required for this OS container.
>>>memory	requested_memory_resource in Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Indicates the minimum amount of memory required for this OS container.
>>>ephemeral-storage	requested_ephemeral_storage_resources in Vdu.OsContainer (ETSI GS NFV-SOL 001 [i.3], clause 6.8.12)	Indicates the minimum amount of local ephemeral storage required for this OS container.

Table 6.2.2.1-6 indicates CISM API object parameter mapping to NFV data models related to StatefulSet.

Table 6.2.2.1-6: CISM API object parameter mapping to NFV data models related to StatefulSet

API object or field	Mapped NFV data element/attribute	Description
StatefulSet v1 apps	(Not applicable)	A resource object to deploy VNFC instances with virtual storage resources. See note.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the StatefulSet. The subsequent name and namespace are fields of the child API object.
>>name	name in Vdu.OsContainerDeployableUnit (ETSI GS NFV-SOL 001 [i.3], clause 6.8.13)	Indicates name of the StatefulSet.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the StatefulSet is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations
>spec	(Not applicable)	Describes specification of the desired behaviour of the StatefulSet. The field corresponds to a child API object of the StatefulSet, and the child API object is specified by StatefuleSetSpec v1 apps.

Table 6.2.2.1-7 indicates CISM API object parameter mapping to NFV data models related to StatefulSetSpec.

Table 6.2.2.1-7: CISM API object parameter mapping to NFV data models related to StatefulSetSpec

API object or field	Mapped NFV data element/attribute	Description
StatefulSetSpec v1 apps	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • StatefulSet v1 apps
>replicas	"number_of_instances" in VduLevel (ETSI GS NFV-SOL 001 [i.3], clause 6.2.19) of levels in VduInstantiationLevels (ETSI GS NFV-SOL 001 [i.3], clause 6.10.2) used in VNF instantiation and scale to level operations.	Indicates the number of desired VNFC instances to be instantiated or after performing Scale VNF to Level operation within a deployment flavour.
	Equal to a numerical value computed as current replica value plus (in case of scaling out) or minus (in case of scaling in) the "number_of_instances" in VduLevel (ETSI GS NFV-SOL 001 [i.3], clause 6.2.19) of deltas in VduScalingAspecDeltas (ETSI GS NFV-SOL 001 [i.3], clause 6.10.6)	Indicates the number of desired VNFC instances after performing Scale VNF operation within a deployment flavour.
>template	(Not applicable)	Describes the VNFC instances that will be created. The field corresponds to a child API object of the StatefulSetSpec, and the child API object is specified by PodTemplateSpec v1 core.
>volumeClaimTemplates	(Not applicable)	Describes list of claims that VNFC instances are allowed to reference. The field corresponds to a child API object of the StatefulSetSpec, and the child API object is specified by PersistentVolumeClaim v1 core.

6.2.3 Discovery and Loadbalancing

6.2.3.1 Mapping to NFV data models

This clause describes the mapping regarding discovery and loadbalancing, which is categorized in clause 4.2.1.2.

Table 6.2.3.1-1 indicates CISM API object parameter mapping to NFV data models related to different types of Service, namely:

- LoadBalancer;
- NodePort.

When the VirtualCp models an ingress, it uses a service of type LoadBalancer or of type NodePort to enable access from outside of the CIS cluster.

NOTE 1: The Kubernetes® ClusterIP Service is excluded because the Service is used only for communication within the CIS cluster. In the present document, VirtualCp, which is generally mappable to Service, is assumed to be used without regard to whether the communication is inside or outside the CIS cluster. Therefore, the ClusterIP Service is not mapped to any object of the ETSI NFV data model.

In case some specific field or API object is applicable to certain types of Service, the "Description" column indicates such applicability.

Table 6.2.3.1-1: CISM API object parameter mapping to NFV data models related to Service with external IP

API object or field	Mapped NFV data element/attribute	Description
Service v1 core	(Not applicable)	A resource object to access/expose VNFC instances from within the cluster or outside the container cluster.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Service. The subsequent name and namespace are fields of the child API object.
>>name	Name of the following node type: VirtualCp (ETSI GS NFV-SOL 001 [i.3], clause 6.8.15)	Indicates name of the Service.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the Service is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>spec	(Not applicable)	Describes specification of the desired behaviour of the Service. The field corresponds to a child API object of the Service, and the child API object is specified by ServiceSpec v1 apps.

Table 6.2.3.1-2 indicates CISM API object parameter mapping to NFV data models related to ServiceSpec.

Table 6.2.3.1-2: CISM API object parameter mapping to NFV data models related to ServiceSpec

API object or field	Mapped NFV data element/attribute	Description
ServiceSpec v1 core	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • Service v1 core
>externalIPs	To be defined	Describes the Service external IP exposed by VirtualCP to access the VNFC instances. See note 2.
>loadbalancerIP	"loadBalancerIp" in VirtualCpAddressData data type in clause 4.4.1.10d of ETSI GS NFV-SOL 003 [i.2]	Describes the loadbalancer IP exposed by VirtualCP to access the VNFC instances. This attribute is only applicable to LoadBalancer type Service. See note 1.
>ports	"portData" in "AdditionalServiceData" (ETSI GS NFV-SOL 001 [i.3], clause 6.2.66) of the VirtualCP node type (ETSI GS NFV-SOL 001 [i.3], clause 6.8.15).	Describes the Service port exposed by VirtualCP to access VNFC instances. The field corresponds to a child API object of the ServiceSpec, and the child API object is specified by ServicePort v1 core.
NOTE 1: The use of the LoadBalancerIP assumes that the CIS cluster is set up to be able to configure an external load balancer. NOTE 2: externalIPs is not supported in this version of the present document.		

Table 6.2.3.1-3 indicates CISM API object parameter mapping to NFV data models related to ServicePort.

Table 6.2.3.1-3: CISM API object parameter mapping to NFV data models related to ServicePort

API object or field	Mapped NFV data element/attribute	Description
ServicePort v1 core	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following ones: ServiceSpec v1 core
>name	"name" in "ServicePortData" type (ETSI GS NFV-SOL 001 [i.3], clause 6.2.65)	Indicates name of the port.
>protocol	"protocol" in "ServicePortData" type (ETSI GS NFV-SOL 001 [i.3], clause 6.2.65)	Indicates the L4 protocol of the port.
>port	"port" in "ServicePortData" type (ETSI GS NFV-SOL 001 [i.3], clause 6.2.65)	Indicates numeric value of port. This attribute is not applicable to NodePort Service.

NOTE 2: The present document assumes that the port number for NodePort is automatically assigned by CISM.

Table 6.2.3.1-4 indicates CISM API object parameter mapping to NFV data models related to Ingress.

Table 6.2.3.1-4: CISM API object parameter mapping to NFV data models related to Ingress

API object or field	Mapped NFV data element/attribute	Description
Ingress v1 networking.k8s.io	(Not applicable)	A resource object to access/expose VNFC instances from within the cluster or outside the container cluster.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Service. The subsequent name and namespace are fields of the child API object.
>>name	Name of the following node type: VirtualCp (ETSI GS NFV-SOL 001 [i.3], clause 6.8.15)	Indicates name of the Ingress.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the Ingress is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>spec	(Not applicable)	Describes specification of the desired behaviour of the Ingress. The field corresponds to a child API object of the Ingress, and the child API object is specified by IngressSpec v1 networking.

Table 6.2.3.1-5 indicates CISM API object parameter mapping to NFV data models related to IngressSpec.

Table 6.2.3.1-5: CISM API object parameter mapping to NFV data models related to IngressSpec

API object or field	Mapped NFV data element/attribute	Description
IngressSpec v1 networking	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • Ingress v1 networking.k8s.io
>rules	(Not applicable)	Describes the list of host rules used to configure the Ingress. The field corresponds to a child API object of the IngressSpec, and the child API object is specified by IngressRule v1 networking.k8s.io.

Table 6.2.3.1-6 indicates CISM API object parameter mapping to NFV data models related to IngressRule.

Table 6.2.3.1-6: CISM API object parameter mapping to NFV data models related to IngressRule

API object or field	Mapped NFV data element/attribute	Description
IngressRule v1 networking.k8s.io.	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • IngressSpec v1 networking
>host	serviceData in AdditionalServiceData data type in ETSI GS NFV-SOL 001 [i.3], clause 6.2.66	Host is the fully qualified domain name of a network host.
>http	"protocol" in "ServicePortData" type (ETSI GS NFV-SOL 001 [i.3], clause 6.2.65)	Describes the access type i.e. http or https. For https protocol, tls certificate needs to be specified at run time. The field corresponds to a child API object of the IngressRule, and the child API object is specified by HTTPIngressRuleValue v1 networking.k8s.io.

Table 6.2.3.1-7 indicates CISM API object parameter mapping to NFV data models related to HTTPIngressRuleValue.

Table 6.2.3.1-7: CISM API object parameter mapping to NFV data models related to HTTPIngressRuleValue

API object or field	Mapped NFV data element/attribute	Description
HTTPIngressRuleValue v1 networking.k8s.io.	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • IngressRule v1 networking.k8s.io.
>paths	(Not applicable)	A collection of paths that map requests to backends. The field corresponds to a child API object of the HTTPIngressRuleValue, and the child API object is specified by HTTPIngressPath v1 networking.k8s.io.

Table 6.2.3.1-8 indicates CISM API object parameter mapping to NFV data models related to HTTPIngressPath.

Table 6.2.3.1-8: CISM API object parameter mapping to NFV data models related to HTTPIngressPath

API object or field	Mapped NFV data element/attribute	Description
HTTPIngressPath v1 networking.k8s.io.	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • HTTPIngressRuleValue v1 networking.k8s.io
>path	serviceData in AdditionalServiceData data type in ETSI GS NFV-SOL 001 [i.3], clause 6.2.66	Path is matched against the path of an incoming request.

NOTE 3: The input parameter mapping of the Kubernetes® Ingress resource object is not supported by this version of the present document.

6.2.4 Configuration & Storage

6.2.4.1 Mapping to NFV data models

This clause describes the mapping regarding Configuration & Storage, which is categorized in clause 4.2.1.2.

Table 6.2.4.1-1 indicates CISM API object parameter mapping to NFV data models related to PersistentVolumeClaim.

Table 6.2.4.1-1: CISM API object parameter mapping to NFV data models related to PersistentVolumeClaim

API object or field	Mapped NFV data element/attribute	Description
PersistentVolumeClaim v1 core	(Not applicable)	A resource object to claim features of volume to be associated with VNFC instances. If per_vnfc_instance in VirtualBlockStorage or VirtualFileStorage is "false" then the API object is directly utilized and if not, then the API object is called as a child API object of the following one: • StatefulSetSpec v1 apps
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the PersistentVolumeClaim. The subsequent name and namespace are fields of the child API object.
>>name	Name of either of the following node types: VirtualBlockStorage (ETSI GS NFV-SOL 001 [i.3], clause 6.8.4), or VirtualFileStorage (ETSI GS NFV-SOL 001 [i.3], clause 6.8.6)	Indicates name of the PersistentVolumeClaim.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the PersistentVolumeClaim is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>spec	(Not applicable)	Describes specification of the desired behaviour of the PersistentVolumeClaim. The field corresponds to a child API object of the PersistentVolumeClaim, and the child API object is specified by PersistentVolumeClaimSpec v1 apps.

Table 6.2.4.1-2 indicates CISM API object parameter mapping to NFV data models related to PersistentVolumeClaimSpec.

Table 6.2.4.1-2: CISM API object parameter mapping to NFV data models related to PersistentVolumeClaimSpec

API object or field	Mapped NFV data element/attribute	Description
PersistentVolumeClaimSpec v1 core	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • PersistentVolumeClaim v1 core
>resources	(Not applicable)	Describes storage resources to be required by the VNF instance. The field corresponds to a child API object of the "PersistentVolumeClaimSpec". The subsequent "request" is a field of the child API object.
>>requests	(Not applicable)	Describes the minimum amount of storage resources required by the VNF instance.
>>>storage	virtual_block_storage_data.size_of_storage in VirtualBlockStorage (ETSI GS NFV-SOL 001 [i.3], clause 6.8.4) or virtual_file_storage_data.size_of_storage in VirtualFileStorage (ETSI GS NFV-SOL 001 [i.3], clause 6.8.6)	Indicates the minimum amount of storage required by the PersistentVolumeClaim.
>storageClassName	storageClassName in StorageAsset (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.13)	Indicates the name of the StorageClass required by the PersistentVolumeClaim.
>volumeMode	Corresponding to either of the following node types: VirtualBlockStorage (ETSI GS NFV-SOL 001 [i.3], clause 6.8.4), or VirtualFileStorage (ETSI GS NFV-SOL 001 [i.3], clause 6.8.6)	Indicates what type of volume is required by the PersistentVolumeClaim. If the node type is VirtualBlockStorage then the value is "Block", and if the node type is VirtualFileStorage then the value is "Filesystem".

Table 6.2.4.1-3 indicates CISM API object parameter mapping to NFV data models related to ConfigMap.

Table 6.2.4.1-3: CISM API object parameter mapping to NFV data models related to ConfigMap

API object or field	Mapped NFV data element/attribute	Description
ConfigMap v1 core	(Not applicable)	A resource object to contain unconfidential configuration data for the relevant VNFC/VNF.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the ConfigMap. The subsequent name and namespace are fields of the child API object.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the ConfigMap is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>data	A value of the key-value pairs in "data" equals the value in the key-value pairs of "vnfConfigurableProperties" in the following: • InstantiateVnfRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.4), • ChangeVnfFlavourRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.7), • ChangeCurrentVnfPkgRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.11a), or • VnfInfoModificationRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.12) or equals the value in the key-value pairs of "additionalParam" in the following: • InstantiateVnfRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause • ScaleVnfRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.5), • ScaleVnfToLevelRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.6), • ChangeCurrentVnfPkgRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.11a), or • ChangeVnfFlavourRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.7)	Indicates a list of key-value pairs which represent unconfidential configuration data to be consumed by VNFC/VNF. See note.
NOTE: The present document assumes that "data" can be updated while existing resource objects consume the "data". Thus, resource objects which are created after updating (e.g. Pod newly created by scaling out, Pod recreated by auto-healing of Deployment, etc.) are expected to obtain configuration different from the existing resource objects. In addition, if the "immutable" field of the ConfigMap is set to true, it is incompatible with the above assumption and an error is expected to occur in case that the "data" is attempted to be updated.		

Table 6.2.4.1-4 indicates CISM API object parameter mapping to NFV data models related to Secret.

Table 6.2.4.1-4: CISM API object parameter mapping to NFV data models related to Secret

API object or field	Mapped NFV data element/attribute	Description
Secret v1 core	(Not applicable)	A resource object to contain confidential configuration data for the relevant VNFC/VNF. The present resource object may be used in case that the definition of the properties of some of the data types derived from the data types specified in ETSI GS NFV-SOL 001 [i.3] includes the following metadata: • sensitive: "true"
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Secret. The subsequent name and namespace are fields of the child API object.
>>namespace	"containerNamespace" in GrantInfo (ETSI GS NFV-SOL 003 [i.2], clause 9.5.3.3)	Indicates namespace to which the Secret is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>data	A value of the key-value pairs in "data" equals the value of the key-value pairs of "vnfConfigurableProperties" in the following: • InstantiateVnfRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.4), • ChangeVnfFlavourRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.7), • ChangeCurrentVnfPkgRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.11a), or • VnfInfoModificationRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.12) or equals the value in the key-value pairs of "additionalParam" in the following: • InstantiateVnfRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.4), • ScaleVnfRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.5), • ScaleVnfToLevelRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.6), • ChangeVnfFlavourRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.7), or • ChangeCurrentVnfPkgRequest (ETSI GS NFV-SOL 002 [i.7]/ETSI GS NFV-SOL 003 [i.2], clause 5.5.2.11a).	Indicates a list of key-value pairs which represent confidential configuration data to be consumed by VNFC/VNF. See notes 1 and 2.

API object or field	Mapped NFV data element/attribute	Description
NOTE 1: The present document assumes that "data" can be updated while existing resource objects consume the "data". Thus, resource objects which are created after updating (e.g. Pod newly created by scaling out, Pod recreated by auto-healing of Deployment, etc.) are expected to obtain configuration different from the existing resource objects. In addition, if the "immutable" field of the Secret is set to true, it is incompatible with the above assumption and an error is expected to occur in case that the "data" is attempted to be updated. NOTE 2: The format of the values of "data" is Base64. When the values are passed to instances of resource objects (e.g. Pod), the CISM will provide decoded data.		

6.3 Output parameters from CISM APIs

6.3.1 Framework

6.3.1.1 Introduction

Clause 6.3.1 describes the framework to specify the handling of output parameters from APIs which are exposed by a CISM.

Differently from the input parameters described in clause 6.2, the specification concerning output parameters from the CISM API is stipulated based on the following aspect:

- Mapping to elements of runtime information NFV data models (i.e. data model of information used and/or expected on the interfaces).

The difference with respect to the input parameter framework comes from the fact that there is only one abstraction level to convey the attribute values from the CISM API, which are the MCIOs.

NOTE: Indeed, precisely there are two ways to convey the attribute values: one is to use Helm™ CLI and the other is to natively use Kubernetes® API, but in either way the values are shown as objects of Kubernetes®. Therefore, as long as the output parameters are concerned, the abstraction level is only one.

6.3.1.2 Mapping to NFV data models

The mapping adopts the same way as the input parameters described in clause 6.2. For more details, refer to clause 6.2.1.2.

6.3.2 Workload

6.3.2.1 Mapping to NFV data models

This clause describes the mapping regarding Workloads, which are categorized in clause 4.2.1.2.

Table 6.3.2.1-1 indicates CISM API object parameter mapping to NFV data models related to Deployment.

Table 6.3.2.1-1: CISM API object parameter mapping to NFV data models related to Deployment

API object or field	Mapped NFV data element/attribute	Description
Deployment v1 apps	(Not applicable)	A resource object to deploy VNFC instances.
>kind	mcioType in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30).	Represents the Deployment object type.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Deployment. The subsequent name and namespace are fields of the child API object.
>>name	mcioName in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30).	Indicates name of the Deployment.
>>namespace	mcioNamespace in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30)	Indicates namespace to which the Deployment is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>status	(Not applicable)	Describes actual status of the Deployment. The field corresponds to a child API object of the Deployment, and the child API object is specified by DeploymentStatus v1 apps.

Table 6.3.2.1-2 indicates CISM API object parameter mapping to NFV data models related to DeploymentStatus.

Table 6.3.2.1-2: CISM API object parameter mapping to NFV data models related to DeploymentStatus

API object or field	Mapped NFV data element/attribute	Description
DeploymentStatus v1 apps	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • Deployment v1 apps
>availableReplicas	availableInstances in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30)	Indicates the number of available VNFC instances instantiated by the Deployment.
>replicas	desiredInstances in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30)	Indicates the number of actual VNFC instances instantiated by the Deployment.

Table 6.3.2.1-3 indicates CISM API object parameter mapping to NFV data models related to Pod. Clause B.3.1 provides the detail of mapping regarding Pod name.

Table 6.3.2.1-3: CISM API object parameter mapping to NFV data models related to Pod

API object or field	Mapped NFV data element/attribute	Description
Pod v1 core	(Not applicable)	A resource object corresponding to each VNFC instance.
>kind	vimLevelResourceType in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Represents the Pod object type.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Pod. The subsequent name, namespace and annotations are fields of the child API object.
>>name	resourceId in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates name of the Pod.
>>namespace	container_namespace in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates namespace to which the Pod is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>>annotations	(Not applicable)	Describes unstructured key value maps stored with a resource that may be set by external tools to store and retrieve arbitrary metadata.
>>> k8s.cni.cncf.io/networks	resourceId in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates a list of identifiers of NetworkAttachmentDefinition representing secondary container cluster external networks which are attached to the Pod.

Table 6.3.2.1-4 indicates CISM API object parameter mapping to NFV data models related to StatefulSet.

Table 6.3.2.1-4: CISM API object parameter mapping to NFV data models related to StatefulSet

API object or field	Mapped NFV data element/attribute	Description
StatefulSet v1 apps	(Not applicable)	A resource object to deploy VNFC instances.
>kind	mcioType in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30).	Represents the StatefulSet object type.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the StatefulSet. The subsequent name and namespace are fields of the child API object.
>>name	mcioName in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30).	Indicates name of the StatefulSet.
>>namespace	mcioNamespace in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30)	Indicates namespace to which the StatefulSet is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>status	(Not applicable)	Describes actual status of the StatefulSet. The field corresponds to a child API object of the StatefulSet, and the child API object is specified by StatefulSetStatus v1 apps.

Table 6.3.2.1-5 indicates CISM API object parameter mapping to NFV data models related to StatefulSetStatus.

Table 6.3.2.1-5: CISM API object parameter mapping to NFV data models related to StatefulSetStatus

API object or field	Mapped NFV data element/attribute	Description
StatefulSetStatus v1 apps	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • StatefulSet v1 apps
>availableReplicas	availableInstances in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30)	Indicates the number of available VNFC instances instantiated by the StatefulSet.
>replicas	desiredInstances in McioInfo data type (ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.24 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.30)	Indicates the number of actual VNFC instances instantiated by the StatefulSet.

6.3.3 Discovery and Loadbalancing

6.3.3.1 Mapping to NFV data models

This clause describes the mapping regarding Discovery and Loadbalancing, which are categorized in clause 4.2.1.2.

Table 6.3.3.1-1 indicates CISM API object parameter mapping to NFV data models related to Service.

Table 6.3.3.1-1: CISM API object parameter mapping to NFV data models related to Service

API object or field	Mapped NFV data element/attribute	Description
Service v1 core	(Not applicable)	A resource object to access/expose VNFC instances from within the cluster or outside the container cluster.
>kind	vimLevelResourceType in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Represents the Service object type.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Service. The subsequent name and namespace are fields of the child API object.
>>name	resourceId in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates name of the Service.
>>namespace	container_namespace in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates namespace to which the Service is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>spec.ports.nodePort	port in ServicePortInfo data type (ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.33 and ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.27).	Indicates the port number of the nodePort of the deployed Service, which is automatically assigned by the CISM.
>status	(Not applicable)	Describes actual status of the Service. The field corresponds to a child API object of the Service, and the child API object is specified by ServiceStatus v1 core.

Table 6.3.3.1-2 indicates CISM API object parameter mapping to NFV data models related to ServiceStatus.

Table 6.3.3.1-2: CISM API object parameter mapping to NFV data models related to ServiceStatus

API object or field	Mapped NFV data element/attribute	Description
ServiceStatus v1 core	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • Service v1 core
>loadBalancer	(Not applicable)	Describes a list containing ingress points for the Service deployed with the type LoadBalancer. The field corresponds to a child API object of the ServiceStatus, and the child API object is specified by LoadBalancerStatus v1 core.

Table 6.3.3.1-3 indicates CISM API object parameter mapping to NFV data models related to Ingress.

Table 6.3.3.1-3: CISM API object parameter mapping to NFV data models related to Ingress

API object or field	Mapped NFV data element/attribute	Description
Ingress v1 networking.k8s.io	(Not applicable)	A resource object to access/expose VNFC instances from within the cluster or outside the container cluster.
>kind	vimLevelResourceType in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Represents the Ingress object type.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the Ingress. The subsequent name and namespace are fields of the child API object.
>>name	resourceId in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates name of the Ingress.
>>namespace	container_namespace in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates namespace to which the Ingress is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.
>status	(Not applicable)	Describes actual status of the Ingress. The field corresponds to a child API object of the Ingress, and the child API object is specified by IngressStatus v1 networking.

Table 6.3.3.1-4 indicates CISM API object parameter mapping to NFV data models related to IngressStatus.

Table 6.3.3.1-4: CISM API object parameter mapping to NFV data models related to IngressStatus

API object or field	Mapped NFV data element/attribute	Description
IngressStatus v1 networking.k8s.io	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following one: • Ingress v1 networking.k8s.io
>loadBalancer	(Not applicable)	Describes a list containing ingress points for the Ingress. The field corresponds to a child API object of the IngressStatus, and the child API object is specified by LoadBalancerStatus v1 core.

Table 6.3.3.1-5 indicates CISM API object parameter mapping to NFV data models related to LoadBalancerStatus.

Table 6.3.3.1-5: CISM API object parameter mapping to NFV data models related to LoadBalancerStatus

API object or field	Mapped NFV data element/attribute	Description
LoadBalancerStatus v1 core	(Not applicable)	The API object is not directly utilized and is only called as a child API object of the following ones: • ServiceStatus v1 core • IngressStatus v1 networking.k8s.io
>ingress	(Not applicable)	Describes a list containing ingress points for the Service deployed with the type LoadBalancer or the Ingress.
>>ip	loadBalancerIp in VirtualCpAddressInfo data type (ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.15b and ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.10a)	Indicates IP address which is set for load-balancer ingress points for the deployed Service or the Ingress.
>>ports	(Not applicable)	Describes a list of ports for the deployed Service or the Ingress.
>>>port	port in ServicePortInfo data type (ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.33 and ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.27).	Indicates the port number of the port of the deployed Service or the Ingress.
>>>protocol	protocol in ServicePortInfo data type (ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.33 and ETSI GS NFV-SOL 003 [i.2], clause 5.5.3.27).	Indicates the protocol of the port of the deployed Service or the Ingress.

6.3.4 Configuration & Storage

6.3.4.1 Mapping to NFV data models

This clause describes the mapping regarding Configuration & Storage, which is categorized in clause 4.2.1.2.

Table 6.3.4.1-1 indicates CISM API object parameter mapping to NFV data models related to PersistentVolumeClaim.

Table 6.3.4.1-1: CISM API object parameter mapping to NFV data models related to PersistentVolumeClaim

API object or field	Mapped NFV data element/attribute	Description
PersistentVolumeClaim v1 core	(Not applicable)	A resource object to claim features of volume to be associated with VNFC instances. If per_vnfc_instance in VirtualBlockStorage or VirtualFileStorage is "false" then the API object is directly utilized and if not, then the API object is called as a child API object of the following one: • StatefulSetSpec v1 apps
>kind	vimLevelResourceType in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], 5.5.3.13)	Represents the PersistentVolumeClaim object type.
>metadata	(Not applicable)	Describes standard object metadata. The field corresponds to a child API object of the PersistentVolumeClaim. The subsequent name and namespace are fields of the child API object.
>>name	resourceId in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates name of the PersistentVolumeClaim.
>>namespace	container_namespace in ResourceHandle data type (ETSI GS NFV-SOL 003 [i.2], clause 4.4.1.7 and ETSI GS NFV-SOL 002 [i.7], clause 5.5.3.13)	Indicates namespace to which the PersistentVolumeClaim is applied. The field defines the name of the logical grouping of resources, identifiers, policies and authorizations.

7 OS container workload management service interface

7.1 Description

This interface allows the CLI consumer to invoke OS container workload management operations based on a MCIOP towards the CLI producer. The Helm™ chart [7] is identified as MCIOP in clause 5.2 of the present document.

The operations provided through this interface are:

- Instantiate containerized workload based on a MCIOP
- Modify containerized workload based on a modified MCIOP
- Terminate containerized workload based on a MCIOP
- Query information about containerized workload based on a MCIOP

7.2 CLI version

Helm™ applies semantic versioning to its CLI [4] and chart [7] specifications, indicated by {MAJOR}.{MINOR}.{PATCH} version elements. The CLI { MAJOR } version for the profiled Helm™ CLI [4] for OS container workload management operations based on a MCIOP shall be set to "3". Details on the Helm™ CLI structure are specified in clause 4.2.2.2 of the present document.

7.3 Sequence diagrams (informative)

7.3.1 Flow of instantiating a containerized workload based on a MCIOP

This clause describes a sequence for instantiating a containerized workload based on a MCIOP.

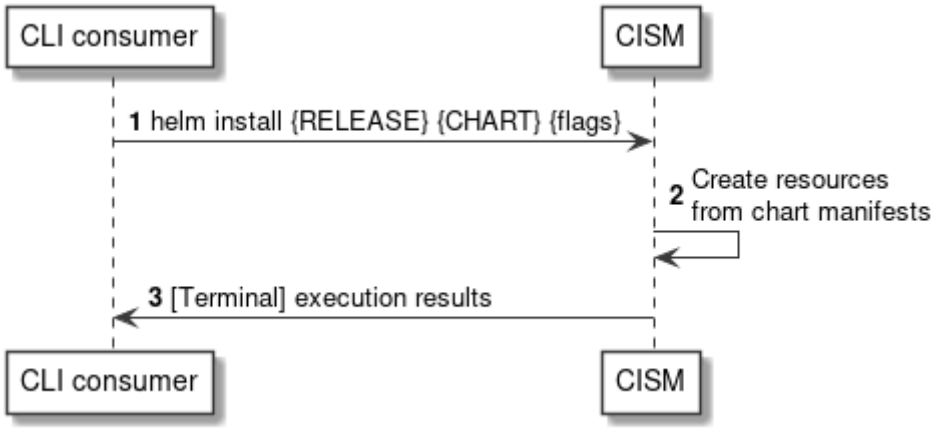

Figure 7.3.1-1: Flow of containerized workload instantiation

The instantiation of a containerized workload based on a MCIOP, as illustrated in Figure 7.3.1-1, consists of the following steps.

Precondition: None.

1) The CLI consumer sends a "helm install" command, including the requested Helm™ release name as {RELEASE}, a reference to the MCIOP as {CHART} and optional command {flags}.

2) The CISM creates the Kubernetes® resources, based on the resource manifests included in the referenced MCIOP.

3) The CISM returns the command execution results in the CLI terminal.

Postcondition: Upon successful completion, the containerized workload based on a MCIOP has been instantiated.

Error handling: In case of failure, appropriate error information is provided in the CLI terminal.

7.3.2 Flow of modifying a containerized workload based on a MCIOP via upgrade

This clause describes a sequence for modifying a containerized workload based on a MCIOP via upgrade.

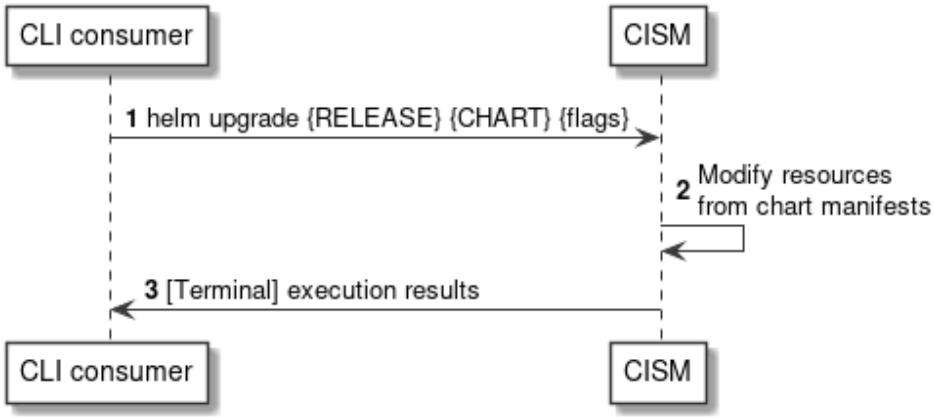

Figure 7.3.2-1: Flow of containerized workload modification via upgrade

The modification of a containerized workload based on a MCIOP via upgrade, as illustrated in Figure 7.3.2-1, consists of the following steps.

Precondition: The Helm™ release to be modified has been instantiated.

1) The CLI consumer sends a "helm upgrade" command, including the name of the Helm™ release to be modified as {RELEASE}, a reference to the modified MCIOP as {CHART} and optional command {flags}.

2) The CISM modifies the Kubernetes® resources, based on the resource manifests included in the referenced modified MCIOP.

3) The CISM returns the command execution results in the CLI terminal.

Postcondition: Upon successful completion, the containerized workload based on a MCIOP has been modified.

Error handling: In case of failure, appropriate error information is provided in the CLI terminal.

7.3.3 Flow of modifying a containerized workload based on a MCIOP via rollback

This clause describes a sequence for modifying a containerized workload based on a MCIOP via rollback.

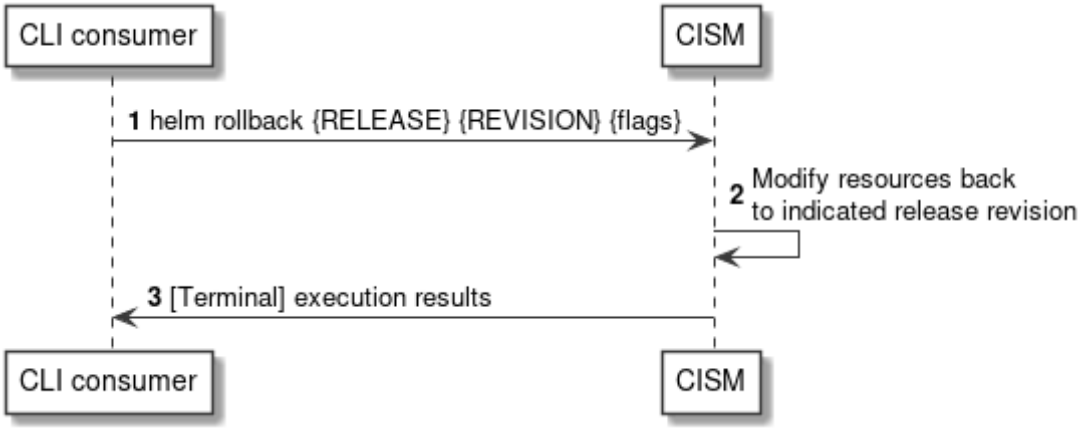

Figure 7.3.3-1: Flow of containerized workload modification via rollback

The modification of a containerized workload based on a MCIOP via rollback, as illustrated in Figure 7.3.3-1, consists of the following steps.

Precondition: The Helm™ release to be modified has been instantiated.

1) The CLI consumer sends a "helm rollback" command, including the name of the Helm™ release to be modified as {RELEASE}, the target Helm™ release revision {REVISION} and optional command {flags}.

2) The CISM modifies the Kubernetes® resources by changing them back to the desired state of the indicated Helm™ release revision.

3) The CISM returns the command execution results in the CLI terminal.

Postcondition: Upon successful completion, the containerized workload based on a MCIOP has been modified.

Error handling: In case of failure, appropriate error information is provided in the CLI terminal.

7.3.4 Flow of terminating a containerized workload based on a MCIOP

This clause describes a sequence for terminating a containerized workload based on a MCIOP.

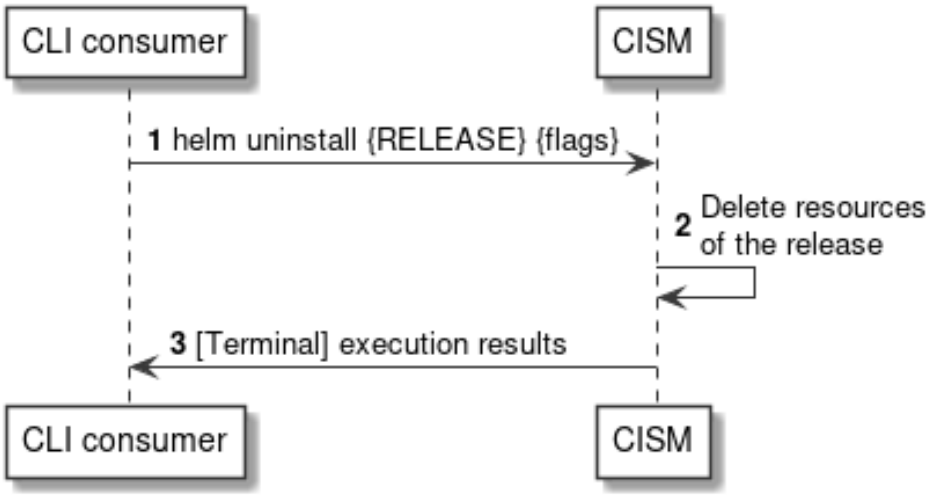

Figure 7.3.4-1: Flow of containerized workload termination

The termination of a containerized workload based on a MCIOP, as illustrated in Figure 7.3.4-1, consists of the following steps.

Precondition: The HelmTM release to be terminated has been instantiated.

1) The CLI consumer sends a "helm uninstall" command, including the name of the Helm™ release to be terminated as {RELEASE} and optional command {flags}.

2) The CISM deletes the Kubernetes® resources constituting the Helm™ release to be terminated.

3) The CISM returns the command execution results in the CLI terminal.

Postcondition: Upon successful completion, the containerized workload based on a MCIOP has been terminated.

Error handling: In case of failure, appropriate error information is provided in the CLI terminal.

7.3.5 Flow of querying information about a containerized workload based on a MCIOP

This clause describes a sequence for querying information about a containerized workload based on a MCIOP.

Figure 7.3.5-1: Flow of containerized workload information querying

The querying information about a containerized workload based on a MCIOP, as illustrated in Figure 7.3.5-1, consists of the following steps.

Precondition: The Helm™ release to be queried has been instantiated.

1) The CLI consumer sends a "helm status" command, including the name of the Helm™ release to be queried as {RELEASE} and optional command {flags}.

2) The CISM returns the Helm™ release status information in the CLI terminal.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the CLI terminal.

7.4 Operations

7.4.1 Introduction

This clause profiles the operations provided by the OS container workload management service interface.

7.4.2 Operation: Helm™ install

This operation represents the Helm™ CLI command "helm install", which instantiates a containerized workload based on a MCIOP.

Table 7.4.2-1 provides the profiling of the "helm install" command against the OS container workload management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The command parameters and responses of the individual command are described in the "helm install" command specification of the profiled Helm™ CLI [4].

Table 7.4.2-1: "helm install" command profiling against OS container workload management service interface requirements

CLI command	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
helm install {RELEASE} {CHART} {flags}	Instantiate a containerized workload based on a MCIOP.	CismWkldMgt.001

7.4.3 Operation: Helm™ upgrade

This operation represents the Helm™ CLI command "helm upgrade", which modifies a containerized workload based on a MCIOP via upgrade.

Table 7.4.3-1 provides the profiling of the "helm upgrade" command against the OS container workload management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The command parameters and responses of the individual command are described in the "helm upgrade" command specification of the profiled Helm™ CLI [4].

Table 7.4.3-1: "helm upgrade" command profiling against OS container workload management service interface requirements

CLI command	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
helm upgrade {RELEASE} {CHART} {flags}	Modify a containerized workload based on a MCIOP via upgrade.	CismWkldMgt.003

7.4.4 Operation: Helm™ rollback

This operation represents the Helm™ CLI command "helm rollback", which modifies a containerized workload based on a MCIOP via rollback.

Table 7.4.4-1 provides the profiling of the "helm rollback" command against the OS container workload management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The command parameters and responses of the individual command are described in the "helm rollback" command specification of the profiled Helm™ CLI [4].

Table 7.4.4-1: "helm rollback" command profiling against OS container workload management service interface requirements

CLI command	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
helm rollback {RELEASE} {REVISION} {flags}	Modify a containerized workload based on a MCIOP via rollback.	CismWkldMgt.003

7.4.5 Operation: Helm™ uninstall

This operation represents the Helm™ CLI command "helm uninstall", which terminates a containerized workload based on a MCIOP.

Table 7.4.5-1 provides the profiling of the "helm uninstall" command against the OS container workload management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The command parameters and responses of the individual command are described in the "helm uninstall" command specification of the profiled Helm™ CLI [4].

Table 7.4.5-1: "helm uninstall" command profiling against OS container workload management service interface requirements

CLI command	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
helm uninstall {RELEASE} {flags}	Terminate a containerized workload based on a MCIOP.	CismWkldMgt.004

7.4.6 Operation: Helm™ status

This operation represents the Helm™ CLI command "helm status", which queries information about a containerized workload based on a MCIOP.

Table 7.4.6-1 provides the profiling of the "helm status" command against the OS container workload management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The command parameters and responses of the individual command are described in the "helm status" command specification of the profiled Helm™ CLI [4].

Table 7.4.6-1: "helm status" command profiling against OS container workload management service interface requirements

CLI command	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
helm status {RELEASE} {flags}	Query information about a containerized workload based on a MCIOP.	CismWkldMgt.002

7.5 Data model

The command and response data structures of the OS container workload management service interface are defined in the respective "helm" command specifications of the profiled Helm™ CLI [4].

7.6 Additional feature profiling

Helm™ provides additional features which are not exposed via the CLI and corresponding operations. Instead, they are provided as functional capabilities.

Table 7.6-1 provides the profiling of the Helm™ additional features against the OS container workload management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

Table 7.6-1: Additional Helm™ feature profiling against OS container workload management service interface requirements

Feature	Description	Requirement identifier from ETSI GS NFV-IFA 040 [2]
Helm™ CLI [4] access control	Access control for CLI execution via authorization for dedicated users on the compute nodes which are hosting the CLI client.	CismWkldMgt.006
Kubernetes® API Access Control [8]	The Helm™ CLI client is authenticated and authorized to access the Kubernetes® API to realize the OS container workload management towards the Kubernetes® CIS cluster.	CismWkldMgt.006

8 OS container compute management service interface

8.1 Description

This interface allows the API consumer to invoke Compute MCIO management operations towards the API producer. Kubernetes® resource objects identified as NFV objects of the Compute MCIO type are listed in clause 5.1.1 of the present document.

The operations provided through this interface are:

- Create Compute MCIO
- Modify the desired state of Compute MCIO
- Modify the actual state of Compute MCIO
- Replace Compute MCIO
- Delete Compute MCIO
- Get information about the desired and actual state of Compute MCIO
- List Compute MCIOs

8.2 API version

The API {VERSION} for the profiled Kubernetes® API [3] for Kubernetes® resource objects identified as Compute MCIOs shall be set to "v1". Details on the Kubernetes® API structure are specified in clause 4.2.1.2 of the present document.

The corresponding Kubernetes® API roots are specified as:

- /api/v1
- /apis/apps/v1
- /apis/batch/v1

8.3 Resource structure and methods

Figures 8.3-1, 8.3-2 and 8.3-3 show the overall resource URI structures for the profiled Kubernetes® API [3] for the OS container compute management service interface.

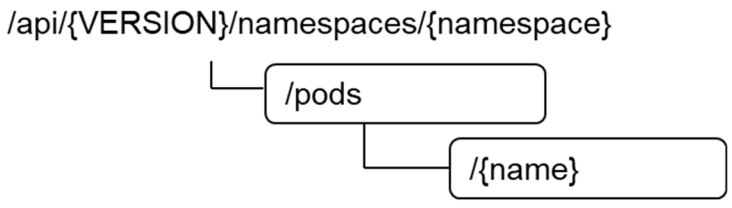

Figure 8.3-1: Resource URI structure of Pod resource object for the OS container compute management service interface

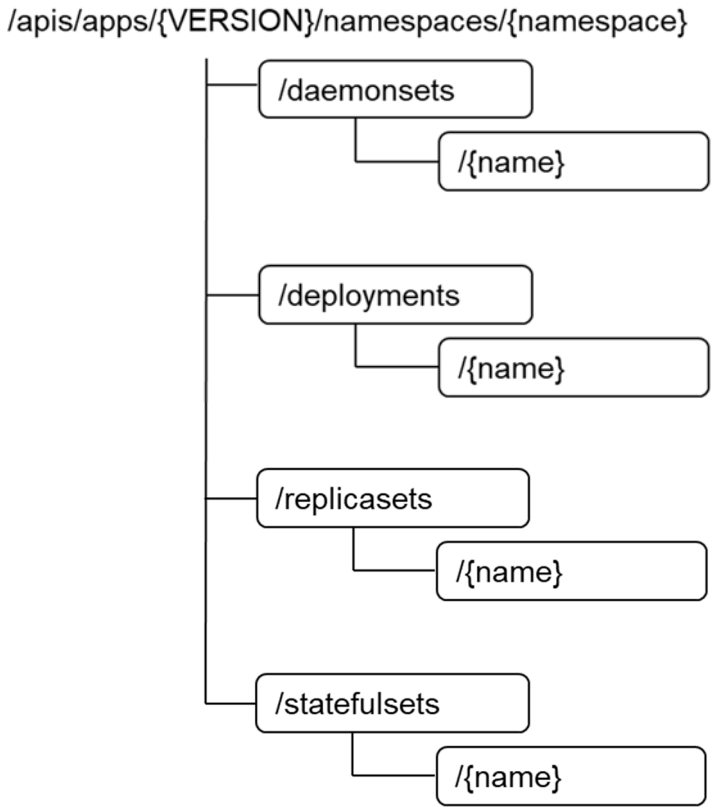

Figure 8.3-2: Resource URI structure of DaemonSet, Deployment, ReplicaSet, StatefulSet resource objects for the OS container compute management service interface

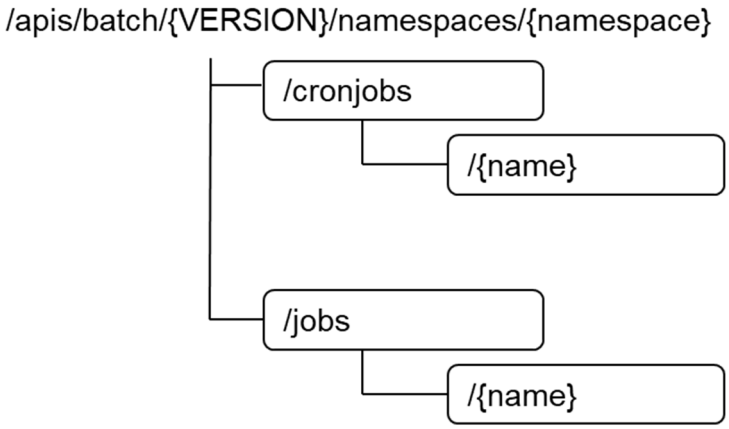

Figure 8.3-3: Resource URI structure of CronJob, Job resource objects for the OS container compute management service interface

Table 8.3-1 lists the individual resources defined, and the applicable HTTP methods.

The CISM shall support responding to requests for all HTTP methods on the resources in Table 8.3-1 that are marked as "M" (mandatory) in the "Cat" column.

Table 8.3-1: Resources and methods overview of the OS container compute management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Pod	/pods	GET	M	List multiple Pod instances.
		POST	M	Create a new "Individual Pod instance" resource.

Resource name	Resource URI	HTTP Method	Cat	Meaning
Individual Pod instance	/pods/{name}	GET	M	Get information about the desired and actual state of an "Individual Pod instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual Pod instance" resource.
		PUT	M	Replace an "Individual Pod instance" resource.
		DELETE	M	Delete an "Individual Pod instance" resource.
DaemonSet	/deamonsets	GET	M	List multiple DaemonSet instances.
		POST	M	Create a new "Individual DaemonSet instance" resource.
Individual DaemonSet instance	/deamonsets/{name}	GET	M	Get information about the desired and actual state of an "Individual DaemonSet instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual DaemonSet instance" resource.
		PUT	M	Replace an "Individual DaemonSet instance" resource.
		DELETE	M	Delete an "Individual DaemonSet instance" resource.
Deployment	/deployments	GET	M	List multiple Deployment instances.
		POST	M	Create a new "Individual Deployment instance" resource.
Individual Deployment instance	/deployments/{name}	GET	M	Get information about the desired and actual state of an "Individual Deployment instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual Deployment instance" resource.
		PUT	M	Replace an "Individual Deployment instance" resource.
		DELETE	M	Delete an "Individual Deployment instance" resource.
ReplicaSet	/replicasets	GET	M	List multiple ReplicaSet instances.
		POST	M	Create a new "Individual ReplicaSet instance" resource.
Individual ReplicaSet instance	/replicasets/{name}	GET	M	Get information about the desired and actual state of an "Individual ReplicaSet instance" resource.

Resource name	Resource URI	HTTP Method	Cat	Meaning
		PATCH	M	Modify the desired or actual state of an "Individual ReplicaSet instance" resource.
		PUT	M	Replace an "Individual ReplicaSet instance" resource.
		DELETE	M	Delete an "Individual ReplicaSet instance" resource.
StatefulSet	/statefulsets	GET	M	List multiple StatefulSet instances.
		POST	M	Create a new "Individual StatefulSet instance" resource.
Individual StatefulSet instance	/statefulsets/{name}	GET	M	Get information about the desired and actual state of an "Individual StatefulSet instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual StatefulSet instance" resource.
		PUT	M	Replace an "Individual StatefulSet instance" resource.
		DELETE	M	Delete an "Individual StatefulSet instance" resource.
CronJob	/cronjobs	GET	M	List multiple CronJob instances.
		POST	M	Create a new "Individual CronJob instance" resource.
Individual CronJob instance	/cronjobs/{name}	GET	M	Get information about the desired and actual state of an "Individual CronJob instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual CronJob instance" resource.
		PUT	M	Replace an "Individual CronJob instance" resource.
		DELETE	M	Delete an "Individual CronJob instance" resource.
Job	/jobs	GET	M	List multiple Job instances.
		POST	M	Create a new "Individual Job instance" resource.
Individual Job instance	/jobs/{name}	GET	M	Get information about the desired and actual state of an "Individual Job instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual Job instance" resource.
		PUT	M	Replace an "Individual Job instance" resource.
		DELETE	M	Delete an "Individual Job instance" resource.

8.4 Sequence diagrams (informative)

8.4.1 Introduction

The sequence diagrams provided in the subsequent sub-clauses are generalized so that they apply to all Kubernetes® resource objects identified as Compute MCIOs. The diagrams and their description contain placeholders indicated as <Compute MCIO> which need to be replaced by the applicable resource name as listed in clause 8.3.

8.4.2 Flow of creating a Compute MCIO

This clause describes a sequence for creating an individual Compute MCIO resource.

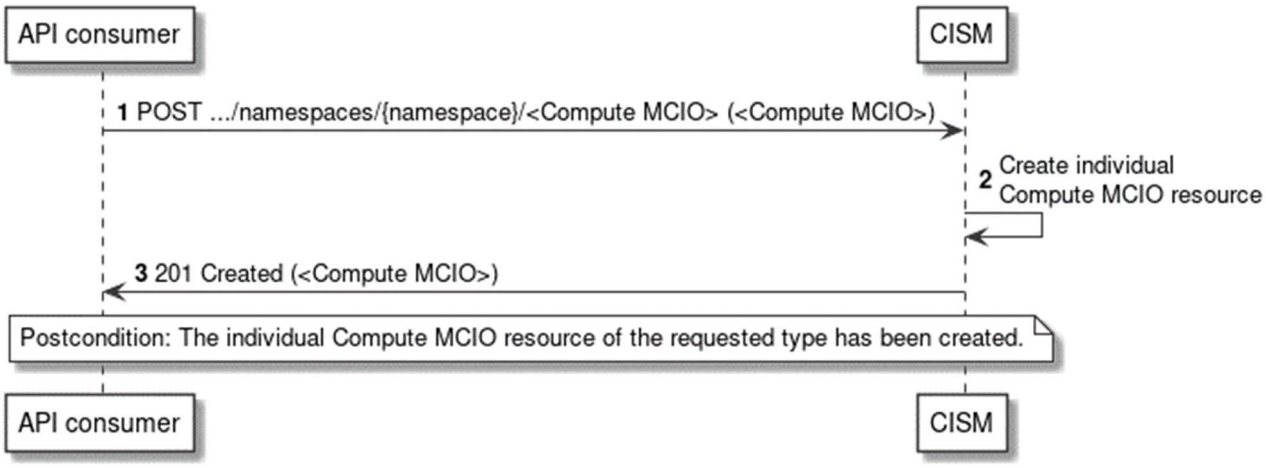

Figure 8.4.2-1: Flow of Compute MCIO creation

The creation of a Compute MCIO resource, as illustrated in Figure 8.4.2-1, consists of the following steps.

Precondition: None.

1) The API consumer sends a POST request to the <Compute MCIO> resource with the appropriate namespace in the URI, including the data structure of the declarative descriptor of the respective Kubernetes® resource object in the payload body.

2) The CISM creates an individual Compute MCIO resource.

3) The CISM returns a "201 Created" response to the API consumer and includes in the payload body a representation of the created <Compute MCIO> resource.

Postcondition: Upon successful completion, the individual Compute MCIO resource has been created.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.3 Flow of modifying a Compute MCIO

This clause describes a sequence for modifying the desired or actual state of an individual Compute MCIO resource.

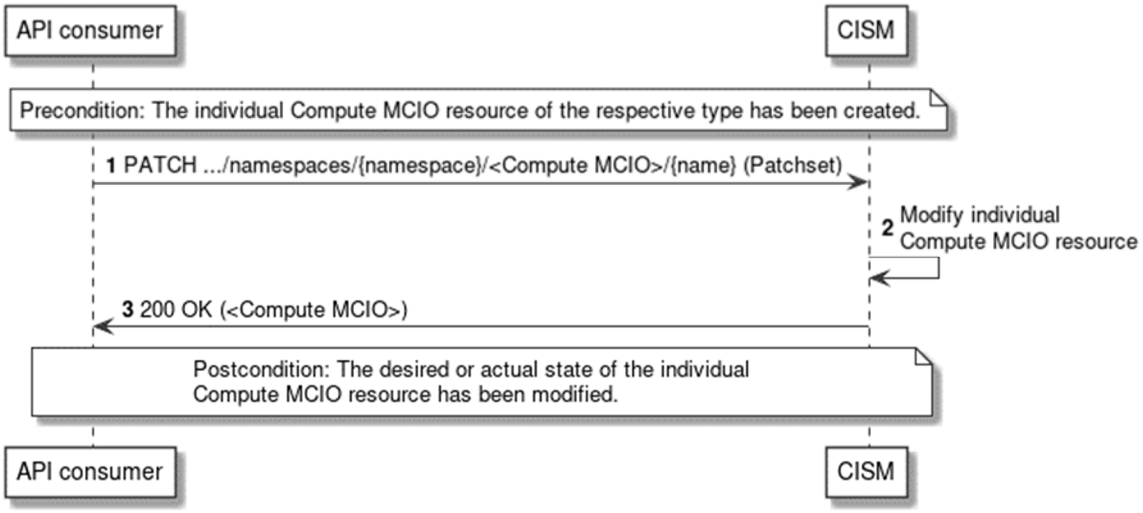

Figure 8.4.3-1: Flow of Compute MCIO modification

The modification of the desired or actual state of an individual Compute MCIO resource, as illustrated in Figure 8.4.3-1, consists of the following steps.

Precondition: The individual Compute MCIO resource of the respective type has been created.

1) The API consumer sends a PATCH request to the individual <Compute MCIO> resource identified by its name with the appropriate namespace in the URI, including the data structure representing the Patchset with the properties of the desired or actual state to be modified in the payload body.

2) The CISM modifies the individual Compute MCIO resource.

3) The CISM returns a "200 OK" response to the API consumer and includes in the payload body a representation of the modified <Compute MCIO> resource.

Postcondition: Upon successful completion, the desired or actual state of the individual Compute MCIO resource has been modified.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.4 Flow of replacing a Compute MCIO

This clause describes a sequence for replacing an individual Compute MCIO resource.

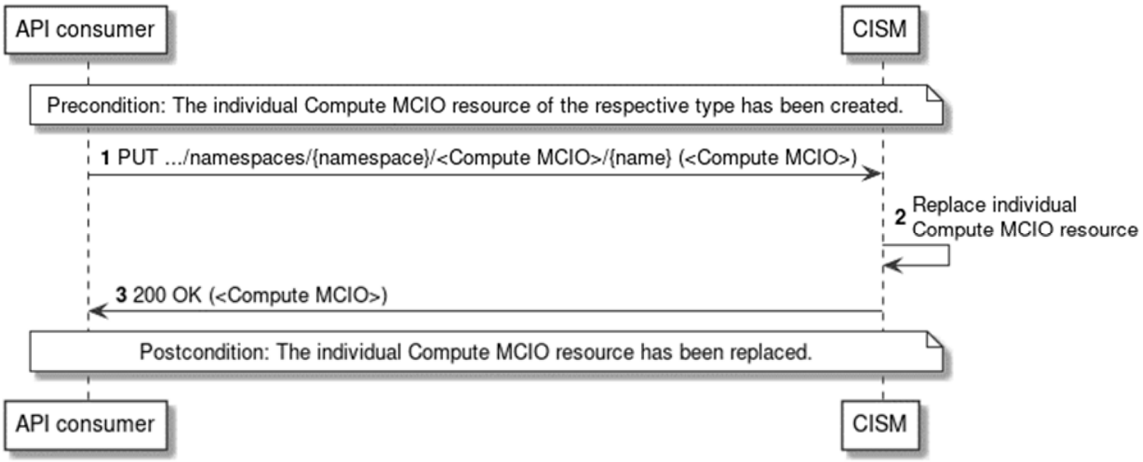

Figure 8.4.4-1: Flow of Compute MCIO replacement

The replacement of an individual Compute MCIO resource, as illustrated in Figure 8.4.4-1, consists of the following steps.

Precondition: The individual Compute MCIO resource of the respective type has been created.

1) The API consumer sends a PUT request to the individual <Compute MCIO> resource identified by its name with the appropriate namespace in the URI, including the data structure of the replacing declarative descriptor of the respective Kubernetes® resource in the payload body.

2) The CISM replaces the individual Compute MCIO resource.

3) The CISM returns a "200 OK" response to the API consumer and includes in the payload body a representation of the replacing <Compute MCIO> resource.

Postcondition: Upon successful completion, the individual Compute MCIO resource has been replaced.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.5 Flow of deleting a Compute MCIO

This clause describes a sequence for deleting an individual Compute MCIO resource.

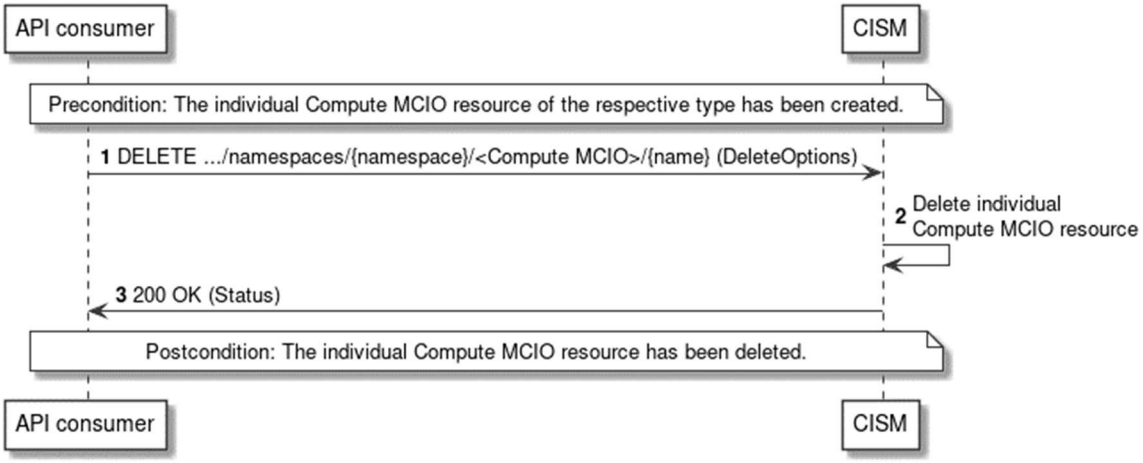

Figure 8.4.5-1: Flow of Compute MCIO deletion

The deletion of an individual Compute MCIO resource, as illustrated in Figure 8.4.5-1, consists of the following steps.

Precondition: The individual Compute MCIO resource of the respective type has been created.

1) The API consumer sends a DELETE request to the individual <Compute MCIO> resource identified by its name with the appropriate namespace in the URI, including the representation of the deletion options in the payload body.

2) The CISM deletes the individual Compute MCIO resource.

3) The CISM returns a "200 OK" response to the API consumer and includes in the payload body a representation of the status details of the operation.

Postcondition: Upon successful completion, the individual Compute MCIO resource has been deleted.

Error handling: In case of failure, appropriate error information is provided in the response.

8.4.6 Flow of listing/getting Compute MCIO information

This clause describes the sequences for listing multiple Compute MCIO resources and getting information about the desired and actual state of an individual Compute MCIO resource.

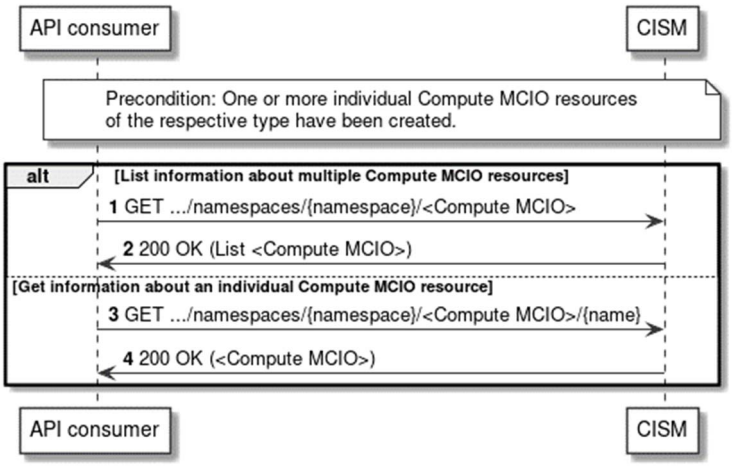

Figure 8.4.6-1: Flow of listing/getting Compute MCIO information

The listing or getting information about one or more Compute MCIO resources, as illustrated in Figure 8.4.6-1, consists of the following steps.

Precondition: One or more individual Compute MCIO resources of the respective type have been created.

1) If the API consumer intends to list multiple Compute MCIO resources, it sends a GET request to the <Compute MCIO> resource with the appropriate namespace in the URI.

2) The CISM returns a "200 OK" response to the API consumer and includes in the payload body a list with one or more representations of the individual <Compute MCIO> resources created within the specified namespace.

3) If the API consumer intends to get the desired and actual state of an individual Compute MCIO resource, it sends a GET request to the individual <Compute MCIO> resource identified by its name with the appropriate namespace in the URI.

4) The CISM returns a "200 OK" response to the API consumer and includes in the payload body a representation of the individual <Compute MCIO> resource.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

8.5 Resources

8.5.1 Introduction

This clause profiles all the resources and methods provided by the OS container compute management service interface.

8.5.2 Resource: Pod

This resource represents the Kubernetes® resource object of a Pod, which is the smallest deployable unit of an application workload as a group of one or more OS containers.

Table 8.5.2-1 provides the profiling of the supported Pod resource methods against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Pod resource object specification of the profiled Kubernetes® API [3].

Table 8.5.2-1: Pod resource methods profiling against OS container compute management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/pods	GET	List multiple Pod instances. Request notifications in the event of changes to Pod resource objects.	CismCompMgt.007 CismCompMgt.008
	POST	Create a new "Individual Pod instance" resource.	CismCompMgt.001
/pods/{name}	GET	Get information about the desired and actual state of an "Individual Pod instance" resource.	CismCompMgt.006
	PATCH	Modify the desired or actual state of an "Individual Pod instance" resource.	CismCompMgt.002 CismCompMgt.003
	PUT	Replace an "Individual Pod instance" resource.	CismCompMgt.004
	DELETE	Delete an "Individual Pod instance" resource.	CismCompMgt.005

8.5.3 Resource: DaemonSet

This resource represents the Kubernetes® resource object of a DaemonSet, which defines a set of Pods that provide local facilities of CIS cluster nodes.

Table 8.5.3-1 provides the profiling of the supported DaemonSet resource methods against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® DaemonSet resource object specification of the profiled Kubernetes® API [3].

Table 8.5.3-1: DaemonSet resource methods profiling against OS container compute management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/deamonsets	GET	List multiple DaemonSet instances. Request notifications in the event of changes to DaemonSet resource objects.	CismCompMgt.007 CismCompMgt.008
	POST	Create a new "Individual DaemonSet instance" resource.	CismCompMgt.001
/deamonsets/{name}	GET	Get information about the desired and actual state of an "Individual DaemonSet instance" resource.	CismCompMgt.006
	PATCH	Modify the desired or actual state of an "Individual DaemonSet instance" resource.	CismCompMgt.002 CismCompMgt.003
	PUT	Replace an "Individual DaemonSet instance" resource.	CismCompMgt.004
	DELETE	Delete an "Individual DaemonSet instance" resource.	CismCompMgt.005

8.5.4 Resource: Deployment

This resource represents the Kubernetes® resource object of a Deployment, which is a stateless application workload.

Table 8.5.4-1 provides the profiling of the supported Deployment resource methods against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Deployment resource object specification of the profiled Kubernetes® API [3].

Table 8.5.4-1: Deployment resource methods profiling against OS container compute management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/deployments	GET	List multiple Deployment instances. Request notifications in the event of changes to Deployment resource objects.	CismCompMgt.007 CismCompMgt.008
	POST	Create a new "Individual Deployment instance" resource.	CismCompMgt.001
/deployments/{name}	GET	Get information about the desired and actual state of an "Individual Deployment instance" resource.	CismCompMgt.006
	PATCH	Modify the desired or actual state of an "Individual Deployment instance" resource.	CismCompMgt.002 CismCompMgt.003
	PUT	Replace an "Individual Deployment instance" resource.	CismCompMgt.004
	DELETE	Delete an "Individual Deployment instance" resource.	CismCompMgt.005

8.5.5 Resource: ReplicaSet

This resource represents the Kubernetes® resource object of a ReplicaSet, which is a stable set of stateless application workloads.

Table 8.5.5-1 provides the profiling of the supported ReplicaSet resource methods against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® ReplicaSet resource object specification of the profiled Kubernetes® API [3].

Table 8.5.5-1: ReplicaSet resource methods profiling against OS container compute management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/replicasets	GET	List multiple ReplicaSet instances. Request notifications in the event of changes to ReplicaSet resource objects.	CismCompMgt.007 CismCompMgt.008
	POST	Create a new "Individual ReplicaSet instance" resource.	CismCompMgt.001
/replicasets/{name}	GET	Get information about the desired and actual state of an "Individual ReplicaSet instance" resource.	CismCompMgt.006
	PATCH	Modify the desired or actual state of an "Individual ReplicaSet instance" resource.	CismCompMgt.002 CismCompMgt.003
	PUT	Replace an "Individual ReplicaSet instance" resource.	CismCompMgt.004
	DELETE	Delete an "Individual ReplicaSet instance" resource.	CismCompMgt.005

8.5.6 Resource: StatefulSet

This resource represents the Kubernetes® resource object of a StatefulSet, which is a stateful application workload.

Table 8.5.6-1 provides the profiling of the supported StatefulSet resource methods against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® StatefulSet resource object specification of the profiled Kubernetes® API [3].

Table 8.5.6-1: StatefulSet resource methods profiling against OS container compute management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/statefulsets	GET	List multiple StatefulSet instances. Request notifications in the event of changes to StatefulSet resource objects.	CismCompMgt.007 CismCompMgt.008
	POST	Create a new "Individual StatefulSet instance" resource.	CismCompMgt.001
/statefulsets/{name}	GET	Get information about the desired and actual state of an "Individual StatefulSet instance" resource.	CismCompMgt.006
	PATCH	Modify the desired or actual state of an "Individual StatefulSet instance" resource.	CismCompMgt.002 CismCompMgt.003
	PUT	Replace an "Individual StatefulSet instance" resource.	CismCompMgt.004
	DELETE	Delete an "Individual StatefulSet instance" resource.	CismCompMgt.005

8.5.7 Resource: CronJob

This resource represents the Kubernetes® resource object of a CronJob, which is a recurring task that runs to completion and then stops.

Table 8.5.7-1 provides the profiling of the supported CronJob resource methods against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® CronJob resource object specification of the profiled Kubernetes® API [3].

Table 8.5.7-1: CronJob resource methods profiling against OS container compute management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/cronjobs	GET	List multiple CronJob instances. Request notifications in the event of changes to CronJob resource objects.	CismCompMgt.007 CismCompMgt.008
	POST	Create a new "Individual CronJob instance" resource.	CismCompMgt.001
/cronjobs/{name}	GET	Get information about the desired and actual state of an "Individual CronJob instance" resource.	CismCompMgt.006
	PATCH	Modify the desired or actual state of an "Individual CronJob instance" resource.	CismCompMgt.002 CismCompMgt.003
	PUT	Replace an "Individual CronJob instance" resource.	CismCompMgt.004
	DELETE	Delete an "Individual CronJob instance" resource.	CismCompMgt.005

8.5.8 Resource: Job

This resource represents the Kubernetes® resource object of a Job, which is a one-off task that runs to completion and then stops.

Table 8.5.8-1 provides the profiling of the supported Job resource methods against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Job resource object specification of the profiled Kubernetes® API [3].

Table 8.5.8-1: Job resource methods profiling against OS container compute management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/jobs	GET	List multiple Job instances. Request notifications in the event of changes to Job resource objects.	CismCompMgt.007 CismCompMgt.008
	POST	Create a new "Individual Job instance" resource.	CismCompMgt.001
/jobs/{name}	GET	Get information about the desired and actual state of an "Individual Job instance" resource.	CismCompMgt.006
	PATCH	Modify the desired or actual state of an "Individual Job instance" resource.	CismCompMgt.002 CismCompMgt.003
	PUT	Replace an "Individual Job instance" resource.	CismCompMgt.004
	DELETE	Delete an "Individual Job instance" resource.	CismCompMgt.005

8.6 Data model

The request and response data structures of the OS container compute management service interface are defined in the respective Compute MCIO Kubernetes® resource object specifications of the profiled Kubernetes® API [3].

8.7 Additional feature profiling

The Kubernetes® API provides additional features which are not exposed via resource objects and corresponding methods. Instead, they are provided as functional capabilities.

Table 8.7-1 provides the profiling of the Kubernetes® API additional features against the OS container compute management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

Table 8.7-1: Additional Kubernetes® API feature profiling against OS container compute management service interface requirements

API feature	Description	Requirement identifier from ETSI GS NFV-IFA 040 [2]
Kubernetes® API Access Control [8]	Users and Kubernetes® service accounts can be authenticated and authorized for API access.	CismCompMgt.009

9 OS container storage management service interface

9.1 Description

This interface allows the API consumer to invoke Storage MCIO management operations towards the API producer. Kubernetes® resource objects identified as NFV objects of the Storage MCIO type are listed in clause 5.1.2 of the present document.

The operations provided through this interface are:

- Create Storage MCIO
- Modify the desired state of Storage MCIO
- Modify the actual state of Storage MCIO
- Replace Storage MCIO
- Delete Storage MCIO
- Get information about the desired and actual state of Storage MCIO
- List Storage MCIOs

9.2 API version

The API {VERSION} for the profiled Kubernetes® API [3] for Kubernetes® resource objects identified as Storage MCIOs shall be set to "v1". Details on the Kubernetes® API structure are specified in clause 4.2.1.2 of the present document.

The corresponding Kubernetes® API root is specified as:

- /api/v1

9.3 Resource structure and methods

Figure 9.3-1 shows the overall resource URI structures for the profiled Kubernetes® API [3] for the OS container storage management service interface.

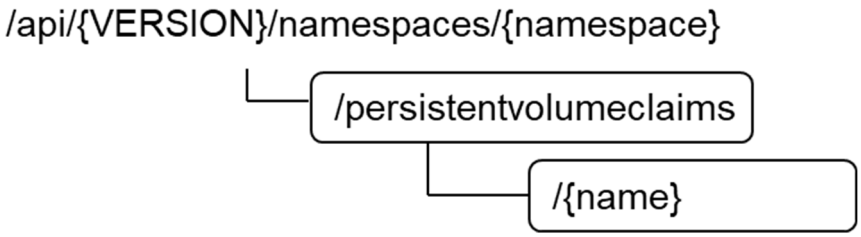

Figure 9.3-1: Resource URI structure of PersistentVolumeClaim resource object for the OS container storage management service interface

Table 9.3-1 lists the individual resources defined, and the applicable HTTP methods.

The CISM shall support responding to requests for all HTTP methods on the resources in Table 9.3-1 that are marked as "M" (mandatory) in the "Cat" column.

Table 9.3-1: Resources and methods overview of the OS container storage management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
PersistentVolumeClaim	/persistentvolumeclaims	GET	M	List multiple PersistentVolumeClaim instances.
		POST	M	Create a new "Individual PersistentVolumeClaim instance" resource.
Individual PersistentVolumeClaim instance	/persistentvolumeclaims/{name}	GET	M	Get information about the desired and actual state of an "Individual PersistentVolumeClaim instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual PersistentVolumeClaim instance" resource.
		PUT	M	Replace an "Individual PersistentVolumeClaim instance" resource.
		DELETE	M	Delete an "Individual PersistentVolumeClaim instance" resource.

9.4 Sequence diagrams (informative)

The sequence diagrams provided in clause 8.4 are generalized so that they apply to all Kubernetes® resource objects identified as Compute MCIOs. For Kubernetes® resource objects identified as Storage MCIOs, in principle the same flows apply as depicted in the sequence diagrams of clause 8.4. The diagrams and their description contain placeholders indicated as <Compute MCIO> which need to be replaced by the applicable resource name as listed in clause 9.3.

9.5 Resources

9.5.1 Introduction

This clause profiles all the resources and methods provided by the OS container storage management service interface.

9.5.2 Resource: PersistentVolumeClaim

This resource represents the Kubernetes® resource object of a PersistentVolumeClaim, which is a request for a persistent storage resource.

Table 9.5.2-1 provides the profiling of the supported PersistentVolumeClaim resource methods against the OS container storage management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® PersistentVolumeClaim resource object specification of the profiled Kubernetes® API [3].

Table 9.5.2-1: PersistentVolumeClaim resource methods profiling against OS container storage management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/persistentvolumeclaims	GET	List multiple PersistentVolumeClaim instances. Request notifications in the event of changes to PersistentVolumeClaim resource objects.	CismStrgMgt.007 CismStrgMgt.008
	POST	Create a new "Individual PersistentVolumeClaim instance" resource.	CismStrgMgt.001
/persistentvolumeclaims/{name}	GET	Get information about the desired and actual state of an "Individual PersistentVolumeClaim instance" resource.	CismStrgMgt.006
	PATCH	Modify the desired or actual state of an "Individual PersistentVolumeClaim instance" resource.	CismStrgMgt.002 CismStrgMgt.003
	PUT	Replace an "Individual PersistentVolumeClaim instance" resource.	CismStrgMgt.004
	DELETE	Delete an "Individual PersistentVolumeClaim instance" resource.	CismStrgMgt.005

9.6 Data model

The request and response data structures of the OS container storage management service interface are defined in the respective Storage MCIO Kubernetes® resource object specifications of the profiled Kubernetes® API [3].

9.7 Additional feature profiling

The Kubernetes® API provides additional features which are not exposed via resource objects and corresponding methods. Instead, they are provided as functional capabilities.

Table 9.7-1 provides the profiling of the Kubernetes® API additional features against the OS container storage management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

Table 9.7-1: Additional Kubernetes® API feature profiling against OS container storage management service interface requirements

API feature	Description	Requirement identifier from ETSI GS NFV-IFA 040 [2]
Kubernetes® API Access Control [8]	Users and Kubernetes® service accounts can be authenticated and authorized for API access.	CismStrgMgt.009

10 OS container network management service interface

10.1 Description

This interface allows the API consumer to invoke Network MCIO management operations towards the API producer. Kubernetes® resource objects identified as NFV objects of the Network MCIO type are listed in clause 5.1.3 of the present document.

The operations provided through this interface are:

- Create Network MCIO
- Modify the desired state of Network MCIO
- Modify the actual state of Network MCIO
- Replace Network MCIO
- Delete Network MCIO
- Get information about the desired and actual state of Network MCIO
- List Network MCIOs

10.2 API version

The API {VERSION} for the profiled Kubernetes® API [3] for Kubernetes® resource objects identified as Network MCIOs shall be set to "v1". Details on the Kubernetes® API structure are specified in clause 4.2.1.2 of the present document.

The corresponding Kubernetes® API roots are specified as:

- /api/v1
- /apis/discovery.k8s.io/v1
- /apis/networking.k8s.io/v1

10.3 Resource structure and methods

Figures 10.3-1, 10.3-2 and 10.3-3 show the overall resource URI structures for the profiled Kubernetes® API [3] for the OS container network management service interface.

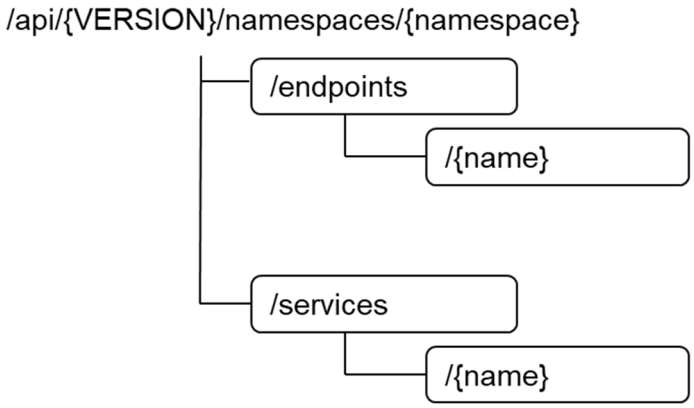

Figure 10.3-1: Resource URI structure of Endpoints and Service resource objects for the OS container network management service interface

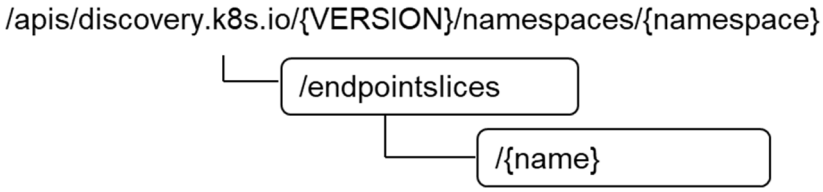

Figure 10.3-2: Resource URI structure of EndpointSlice resource object for the OS container network management service interface

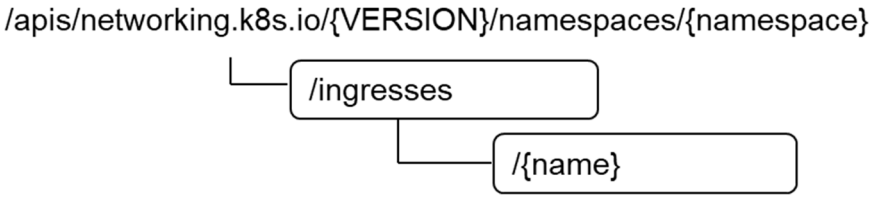

Figure 10.3-3: Resource URI structure of Ingress resource object for the OS container network management service interface

Table 10.3-1 lists the individual resources defined, and the applicable HTTP methods.

The CISM shall support responding to requests for all HTTP methods on the resources in Table 10.3-1 that are marked as "M" (mandatory) in the "Cat" column.

Table 10.3-1: Resources and methods overview of the OS container network management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Endpoints	/endpoints	GET	M	List multiple Endpoints instances.
		POST	M	Create a new "Individual Endpoints instance" resource.
Individual Endpoints instance	/endpoints/{name}	GET	M	Get information about the desired and actual state of an "Individual Endpoints instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual Endpoints instance" resource.

Resource name	Resource URI	HTTP Method	Cat	Meaning
		PUT	M	Replace an "Individual Endpoints instance" resource.
		DELETE	M	Delete an "Individual Endpoints instance" resource.
Service	/services	GET	M	List multiple Service instances.
		POST	M	Create a new "Individual Service instance" resource.
Individual Service instance	/services/{name}	GET	M	Get information about the desired and actual state of an "Individual Service instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual Service instance" resource.
		PUT	M	Replace an "Individual Service instance" resource.
		DELETE	M	Delete an "Individual Service instance" resource.
EndpointSlice	/endpointslices	GET	M	List multiple EndpointSlice instances.
		POST	M	Create a new "Individual EndpointSlice instance" resource.
Individual EndpointSlice instance	/endpointslices/{name}	GET	M	Get information about the desired and actual state of an "Individual EndpointSlice instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual EndpointSlice instance" resource.
		PUT	M	Replace an "Individual EndpointSlice instance" resource.
		DELETE	M	Delete an "Individual EndpointSlice instance" resource.
Ingress	/ingresses	GET	M	List multiple Ingress instances.
		POST	M	Create a new "Individual Ingress instance" resource.
Individual Ingress instance	/ingresses/{name}	GET	M	Get information about the desired and actual state of an "Individual Ingress instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual Ingress instance" resource.
		PUT	M	Replace an "Individual Ingress instance" resource.
		DELETE	M	Delete an "Individual Ingress instance" resource.

10.4 Sequence diagrams (informative)

The sequence diagrams provided in clause 8.4 are generalized so that they apply to all Kubernetes® resource objects identified as Compute MCIOs. For Kubernetes® resource objects identified as Network MCIOs, in principle the same flows apply as depicted in the sequence diagrams of clause 8.4. The diagrams and their description contain placeholders indicated as <Compute MCIO> which need to be replaced by the applicable resource name as listed in clause 10.3.

10.5 Resources

10.5.1 Introduction

This clause profiles all the resources and methods provided by the OS container network management service interface.

10.5.2 Resource: Endpoints

This resource represents the Kubernetes® resource object of Endpoints, which is a group of network addresses with a common set of ports.

Table 10.5.2-1 provides the profiling of the supported Endpoints resource methods against the OS container network management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Endpoints resource object specification of the profiled Kubernetes® API [3].

Table 10.5.2-1: Endpoints resource methods profiling against OS container network management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/endpoints	GET	List multiple Endpoints instances. Request notifications in the event of changes to Endpoints resource objects.	CismNetwMgt.007 CismNetwMgt.008
	POST	Create a new "Individual Endpoints instance" resource.	CismNetwMgt.001
/endpoints/{name}	GET	Get information about the desired and actual state of an "Individual Endpoints instance" resource.	CismNetwMgt.006
	PATCH	Modify the desired or actual state of an "Individual Endpoints instance" resource.	CismNetwMgt.002 CismNetwMgt.003
	PUT	Replace an "Individual Endpoints instance" resource.	CismNetwMgt.004
	DELETE	Delete an "Individual Endpoints instance" resource.	CismNetwMgt.005

10.5.3 Resource: Service

This resource represents the Kubernetes® resource object of a Service, which is a stable IP endpoint loadbalanced across multiple application workload replicas.

Table 10.5.3-1 provides the profiling of the supported Service resource methods against the OS container network management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Service resource object specification of the profiled Kubernetes® API [3].

Table 10.5.3-1: Service resource methods profiling against OS container network management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/services	GET	List multiple Service instances. Request notifications in the event of changes to Service resource objects.	CismNetwMgt.007 CismNetwMgt.008
	POST	Create a new "Individual Service instance" resource.	CismNetwMgt.001
/services/{name}	GET	Get information about the desired and actual state of an "Individual Service instance" resource.	CismNetwMgt.006
	PATCH	Modify the desired or actual state of an "Individual Service instance" resource.	CismNetwMgt.002 CismNetwMgt.003
	PUT	Replace an "Individual Service instance" resource.	CismNetwMgt.004
	DELETE	Delete an "Individual Service instance" resource.	CismNetwMgt.005

10.5.4 Resource: EndpointSlice

This resource represents the Kubernetes® resource object of a EndpointSlice, which is a set of network endpoints.

Table 10.5.4-1 provides the profiling of the supported EndpointSlice resource methods against the OS container network management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® EndpointSlice resource object specification of the profiled Kubernetes® API [3].

Table 10.5.4-1: EndpointSlice resource methods profiling against OS container network management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/endpointslices	GET	List multiple EndpointSlice instances. Request notifications in the event of changes to EndpointSlice resource objects.	CismNetwMgt.007 CismNetwMgt.008
	POST	Create a new "Individual EndpointSlice instance" resource.	CismNetwMgt.001
/endpointslices/{name}	GET	Get information about the desired and actual state of an "Individual EndpointSlice instance" resource.	CismNetwMgt.006
	PATCH	Modify the desired or actual state of an "Individual EndpointSlice instance" resource.	CismNetwMgt.002 CismNetwMgt.003
	PUT	Replace an "Individual EndpointSlice instance" resource.	CismNetwMgt.004
	DELETE	Delete an "Individual EndpointSlice instance" resource.	CismNetwMgt.005

10.5.5 Resource: Ingress

This resource represents the Kubernetes® resource object of an Ingress, which is an external access routed to one or more Service resource objects.

Table 10.5.5-1 provides the profiling of the supported Ingress resource methods against the OS container network management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Ingress resource object specification of the profiled Kubernetes® API [3].

Table 10.5.5-1: Ingress resource methods profiling against OS container network management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/ingresses	GET	List multiple Ingress instances. Request notifications in the event of changes to Ingress resource objects.	CismNetwMgt.007 CismNetwMgt.008
	POST	Create a new "Individual Ingress instance" resource.	CismNetwMgt.001
/ingresses/{name}	GET	Get information about the desired and actual state of an "Individual Ingress instance" resource.	CismNetwMgt.006
	PATCH	Modify the desired or actual state of an "Individual Ingress instance" resource.	CismNetwMgt.002 CismNetwMgt.003
	PUT	Replace an "Individual Ingress instance" resource.	CismNetwMgt.004
	DELETE	Delete an "Individual Ingress instance" resource.	CismNetwMgt.005

10.6 Data model

The request and response data structures of the OS container network management service interface are defined in the respective Network MCIO Kubernetes® resource object specifications of the profiled Kubernetes® API [3].

10.7 Additional feature profiling

The Kubernetes® API provides additional features which are not exposed via resource objects and corresponding methods. Instead, they are provided as functional capabilities.

Table 10.7-1 provides the profiling of the Kubernetes® API additional features against the OS container network management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

Table 10.7-1: Additional Kubernetes® API feature profiling against OS container network management service interface requirements

API feature	Description	Requirement identifier from ETSI GS NFV-IFA 040 [2]
Kubernetes® API Access Control [8]	Users and Kubernetes® service accounts can be authenticated and authorized for API access.	CismNetwMgt.009

11 OS container configuration management service interface

11.1 Description

This interface allows the API consumer to invoke management operations towards the API producer on:

- MCIO configurations;
- MCIO policies;
- namespaces; and
- namespace quotas.

Kubernetes® resource objects identified as NFV objects of the MCIO configurations type are listed in clause 5.1.4 of the present document. Kubernetes® resource objects identified as NFV objects of the MCIO policies type are listed in clause 5.1.5 of the present document. Kubernetes® resource objects identified as NFV objects of the namespaces are listed in clause 5.3 of the present document and Kubernetes® resource objects identified as NFV objects of the namespace quotas are listed in clause 5.4 of the present document.

The operations provided through this interface are:

- Create MCIO configurations
- Modify the desired state of MCIO configurations
- Modify the actual state of MCIO configurations
- Replace MCIO configurations
- Delete MCIO configurations
- Get information about the desired and actual state of MCIO configurations
- List MCIO configurations
- Create MCIO policies
- Modify the desired state of MCIO policies
- Modify the actual state of MCIO policies
- Replace MCIO policies
- Delete MCIO policies
- Get information about the desired and actual state of MCIO policies
- List MCIO policies
- Create namespace
- Get information about namespaces
- Delete namespaces
- Create namespace quota
- Get information about namespace quotas
- Modify namespace quota

- Delete namespace quota

11.2 API version

The API {VERSION} for the profiled Kubernetes® API [3] for Kubernetes® resource objects identified as MCIO configurations or MCIO policies shall be set to "v1". Details on the Kubernetes® API structure are specified in clause 4.2.1.2 of the present document.

The corresponding Kubernetes® API roots are specified as:

- /api/v1
- /apis/apiextensions.k8s.io/v1
- /apis/policy/v1
- /apis/networking.k8s.io/v1

11.3 Resource structure and methods

Figures 11.3-1, 11.3-2, 11.3-3, 11.3-4, 11.3-5 and 11.3-6 show the overall resource URI structures for the profiled Kubernetes® API [3] for the OS container configuration management service interface.

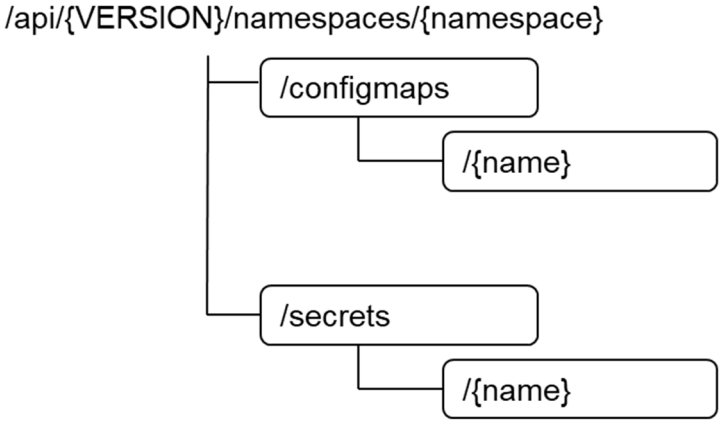

Figure 11.3-1: Resource URI structure of ConfigMap and Secret resource objects for the OS container configuration management service interface

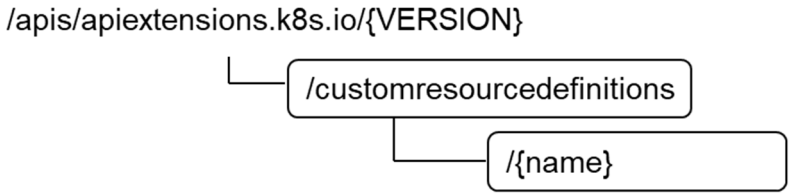

Figure 11.3-2: Resource URI structure of CustomResourceDefinition resource object for the OS container configuration management service interface

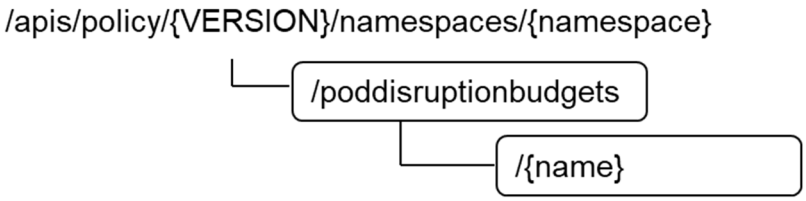

Figure 11.3-3: Resource URI structure of PodDisruptionBudget resource object for the OS container configuration management service interface

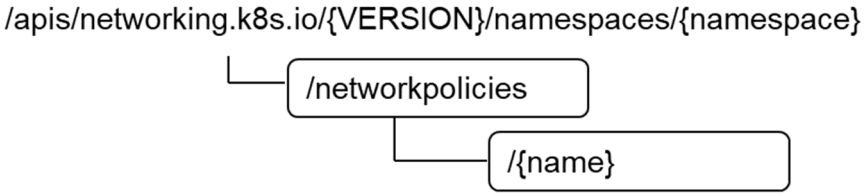

Figure 11.3-4: Resource URI structure of NetworkPolicy resource object for the OS container configuration management service interface

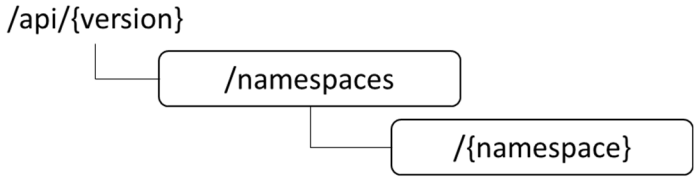

Figure 11.3-5: Resource URI structure of Namespace resource object for the OS container configuration management service interface

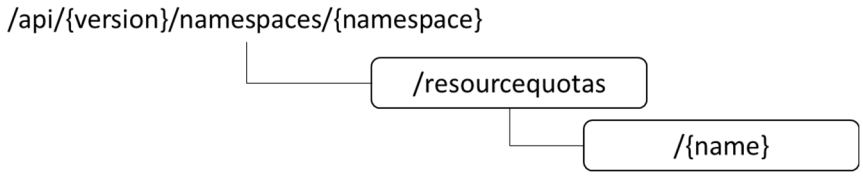

Figure 11.3-6: Resource URI structure of ResourceQuotas object for the OS container configuration management service interface

Table 11.3-1 lists the individual resources defined, and the applicable HTTP methods.

The CISM shall support responding to requests for all HTTP methods on the resources in Table 11.3-1 that are marked as "M" (mandatory) in the "Cat" column.

Table 11.3-1: Resources and methods overview of the OS container configuration management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
ConfigMap	/configmaps	GET	M	List multiple ConfigMap instances.
		POST	M	Create a new "Individual ConfigMap instance" resource.
Individual ConfigMap instance	/configmaps/{name}	GET	M	Get information about the desired and actual state of an "Individual ConfigMap instance" resource.

Resource name	Resource URI	HTTP Method	Cat	Meaning
		PATCH	M	Modify the desired or actual state of an "Individual ConfigMap instance" resource.
		PUT	M	Replace an "Individual ConfigMap instance" resource.
		DELETE	M	Delete an "Individual ConfigMap instance" resource.
Secret	/secrets	GET	M	List multiple Secret instances.
		POST	M	Create a new "Individual Secret instance" resource.
Individual Secret instance	/secrets/{name}	GET	M	Get information about the desired and actual state of an "Individual Secret instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual Secret instance" resource.
		PUT	M	Replace an "Individual Secret instance" resource.
		DELETE	M	Delete an "Individual Secret instance" resource.
CustomResource Definition	/customresourcedefinitions	GET	M	List multiple CustomResourceDefinition instances.
		POST	M	Create a new "Individual CustomResourceDefinition instance" resource.
Individual CustomResource Definition instance	/customresourcedefinitions/{name}	GET	M	Get information about the desired and actual state of an "Individual CustomResourceDefinition instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual CustomResourceDefinition instance" resource.
		PUT	M	Replace an "Individual CustomResourceDefinition instance" resource.
		DELETE	M	Delete an "Individual CustomResourceDefinition instance" resource.
PodDisruptionBudget	/poddisruptionbudgets	GET	M	List multiple PodDisruptionBudget instances.
		POST	M	Create a new "Individual PodDisruptionBudget instance" resource.
Individual PodDisruptionBudget instance	/poddisruptionbudgets/{name}	GET	M	Get information about the desired and actual state of an "Individual PodDisruptionBudget instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual PodDisruptionBudget instance" resource.
		PUT	M	Replace an "Individual PodDisruptionBudget instance" resource.

Resource name	Resource URI	HTTP Method	Cat	Meaning
		DELETE	M	Delete an "Individual PodDisruptionBudget instance" resource.
NetworkPolicy	/networkpolicies	GET	M	List multiple NetworkPolicy instances.
		POST	M	Create a new "Individual NetworkPolicy instance" resource.
Individual NetworkPolicy instance	/networkpolicies/{name}	GET	M	Get information about the desired and actual state of an "Individual NetworkPolicy instance" resource.
		PATCH	M	Modify the desired or actual state of an "Individual NetworkPolicy instance" resource.
		PUT	M	Replace an "Individual NetworkPolicy instance" resource.
		DELETE	M	Delete an "Individual NetworkPolicy instance" resource.
Namespace	/namespaces	GET	M	List multiple namespace resources.
		POST	M	Create a new "Individual Namespace instance" resource.
Individual Namespace instance	/namespaces/{namespace}	GET	M	Get information about the specified "Individual Namespace instance" resource.
		DELETE	M	Delete an "Individual Namespace instance" resource.
ResourceQuota	/resourcequotas	GET	M	Get information about ResourceQuota resources.
		POST	M	Create a new "Individual ResourceQuota instance" resource.
Individual ResourceQuota instance	/resourcequotas/{name}	GET	M	Get information about the specified "Individual ResourceQuota instance" resource.
		PATCH	M	Modify an "Individual ResourceQuota instance" resource.
		PUT	M	Replace an "Individual ResourceQuota instance" resource.
		DELETE	M	Delete an "Individual ResourceQuota instance" resource.

11.4 Sequence diagrams (informative)

The sequence diagrams provided in clause 8.4 are generalized so that they apply to all Kubernetes® resource objects identified as Compute MCIOs. For Kubernetes® resource objects identified as MCIO configurations or MCIO policies, in principle the same flows apply as depicted in the sequence diagrams of clause 8.4. The diagrams and their description contain placeholders indicated as <Compute MCIO> which need to be replaced by the applicable resource name as listed in clause 11.3.

11.5 Resources

11.5.1 Introduction

This clause profiles all the resources and methods provided by the OS container configuration management service interface.

11.5.2 Resource: ConfigMap

This resource represents the Kubernetes® resource object of ConfigMap, which stores non-confidential configuration data for application workloads.

Table 11.5.2-1 provides the profiling of the supported ConfigMap resource methods against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® ConfigMap resource object specification of the profiled Kubernetes® API [3].

Table 11.5.2-1: ConfigMap resource methods profiling against OS container configuration management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/configmaps	GET	List multiple ConfigMap instances. Request notifications in the event of changes to ConfigMap resource objects.	CismCfgMgt.013 CismCfgMgt.014
	POST	Create a new "Individual ConfigMap instance" resource.	CismCfgMgt.008
/configmaps/{name}	GET	Get information about the desired and actual state of an "Individual ConfigMap instance" resource.	CismCfgMgt.012
	PATCH	Modify the desired or actual state of an "Individual ConfigMap instance" resource.	CismCfgMgt.009
	PUT	Replace an "Individual ConfigMap instance" resource.	CismCfgMgt.010
	DELETE	Delete an "Individual ConfigMap instance" resource.	CismCfgMgt.011

11.5.3 Resource: Secret

This resource represents the Kubernetes® resource object of a Secret, which stores confidential configuration data for application workloads.

Table 11.5.3-1 provides the profiling of the supported Secret resource methods against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Secret resource object specification of the profiled Kubernetes® API [3].

Table 11.5.3-1: Secret resource methods profiling against OS container configuration management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/secrets	GET	List multiple Secret instances. Request notifications in the event of changes to Secret resource objects.	CismCfgMgt.013 CismCfgMgt.014
	POST	Create a new "Individual Secret instance" resource.	CismCfgMgt.008
/secrets/{name}	GET	Get information about the desired and actual state of an "Individual Secret instance" resource.	CismCfgMgt.012
	PATCH	Modify the desired or actual state of an "Individual Secret instance" resource.	CismCfgMgt.009
	PUT	Replace an "Individual Secret instance" resource.	CismCfgMgt.010
	DELETE	Delete an "Individual Secret instance" resource.	CismCfgMgt.011

11.5.4 Resource: CustomResourceDefinition

This resource represents the Kubernetes® resource object of a CustomResourceDefinition, which defines custom resources as API extensions.

Table 11.5.4-1 provides the profiling of the supported CustomResourceDefinition resource methods against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® CustomResourceDefinition resource object specification of the profiled Kubernetes® API [3].

Table 11.5.4-1: CustomResourceDefinition resource methods profiling against OS container configuration management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/customresourcedefinitions	GET	List multiple CustomResourceDefinition instances. Request notifications in the event of changes to CustomResourceDefinition resource objects.	CismCfgMgt.013 CismCfgMgt.014
	POST	Create a new "Individual CustomResourceDefinition instance" resource.	CismCfgMgt.008
/customresourcedefinitions/{name}	GET	Get information about the desired and actual state of an "Individual CustomResourceDefinition instance" resource.	CismCfgMgt.012
	PATCH	Modify the desired or actual state of an "Individual CustomResourceDefinition instance" resource.	CismCfgMgt.009
	PUT	Replace an "Individual CustomResourceDefinition instance" resource.	CismCfgMgt.010
	DELETE	Delete an "Individual CustomResourceDefinition instance" resource.	CismCfgMgt.011

11.5.5 Resource: PodDisruptionBudget

This resource represents the Kubernetes® resource object of an PodDisruptionBudget, which is a policy to control the number of Pods of a workload application allowed to be evicted.

Table 11.5.5-1 provides the profiling of the supported PodDisruptionBudget resource methods against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® PodDisruptionBudget resource object specification of the profiled Kubernetes® API [3].

Table 11.5.5-1: PodDisruptionBudget resource methods profiling against OS container configuration management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/poddisruptionbudgets	GET	List multiple PodDisruptionBudget instances. Request notifications in the event of changes to PodDisruptionBudget resource objects.	CismCfgMgt.020 CismCfgMgt.021
	POST	Create a new "Individual PodDisruptionBudget instance" resource.	CismCfgMgt.015
/poddisruptionbudgets/{name}	GET	Get information about the desired and actual state of an "Individual PodDisruptionBudget instance" resource.	CismCfgMgt.019
	PATCH	Modify the desired or actual state of an "Individual PodDisruptionBudget instance" resource.	CismCfgMgt.016
	PUT	Replace an "Individual PodDisruptionBudget instance" resource.	CismCfgMgt.017
	DELETE	Delete an "Individual PodDisruptionBudget instance" resource.	CismCfgMgt.018

11.5.6 Resource: NetworkPolicy

This resource represents the Kubernetes® resource object of an NetworkPolicy, which is a policy to control the traffic flow to/from workload applications.

Table 11.5.6-1 provides the profiling of the supported NetworkPolicy resource methods against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® NetworkPolicy resource object specification of the profiled Kubernetes® API [3].

Table 11.5.6-1: NetworkPolicy resource methods profiling against OS container configuration management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/networkpolicies	GET	List multiple NetworkPolicy instances. Request notifications in the event of changes to NetworkPolicy resource objects.	CismCfgMgt.020 CismCfgMgt.021
	POST	Create a new "Individual NetworkPolicy instance" resource.	CismCfgMgt.015
/networkpolicies/{name}	GET	Get information about the desired and actual state of an "Individual NetworkPolicy instance" resource.	CismCfgMgt.019
	PATCH	Modify the desired or actual state of an "Individual NetworkPolicy instance" resource.	CismCfgMgt.016
	PUT	Replace an "Individual NetworkPolicy instance" resource.	CismCfgMgt.017
	DELETE	Delete an "Individual NetworkPolicy instance" resource.	CismCfgMgt.018

11.5.7 Resource: Namespace

This resource represents the Kubernetes® resource object of a namespace, which provides the logical grouping of resources, identifiers, policies and authorizations.

Table 11.5.7-1 provides the profiling of the supported Namespace resource methods against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® Namespace resource object specification of the profiled Kubernetes® API [3].

Table 11.5.7-1: Namespace resource methods profiling against OS container configuration management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/namespaces	GET	List multiple Namespaces. Request notifications in the event of changes to Namespace objects.	CismCfgMgt.002
	POST	Create a new "Individual Namespace instance" resource.	CismCfgMgt.001
/namespaces/{namespace}	GET	Get information about an "Individual Namespace instance" resource.	CismCfgMgt.002
	DELETE	Delete an "Individual Namespace instance" resource.	CismCfgMgt.003

11.5.8 Resource: NamespaceQuota

This resource represents the Kubernetes® ResourceQuota object, which maps to NFV namespace quota and represents constraints that limit aggregate resource consumption per namespace.

Table 11.5.8-1 provides the profiling of the supported ResourceQuota resource methods against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the Kubernetes® ResourceQuota resource object specification of the profiled Kubernetes® API [3].

Table 11.5.8-1: ResourceQuota resource methods profiling against OS container configuration management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/resourcequotas	GET	List multiple ResourceQuota instances. Request notifications in the event of changes to ResourceQuota objects.	CismCfgMgt.005 CismCfgMgt.022
	POST	Create a new "Individual ResourceQuota instance" resource.	CismCfgMgt.004
/resourcequotas/{name}	GET	Get information about an "Individual ResouceQuota instance" resource.	CismCfgMgt.005
	PATCH	Modify an "Individual ResourceQuota instance" resource.	CismCfgMgt.006
	PUT	Replace an "Individual ResourceQuota instance" resource.	CismCfgMgt.006
	DELETE	Delete an "Individual ResourceQuota instance" resource.	CismCfgMgt.007

11.6 Data model

The request and response data structures of the OS container configuration management service interface are defined in the respective Kubernetes® resource object specifications of the profiled Kubernetes® API [3].

11.7 Additional feature profiling

The Kubernetes® API provides additional features which are not exposed via resource objects and corresponding methods. Instead, they are provided as functional capabilities.

Table 11.7-1 provides the profiling of the Kubernetes® API additional features against the OS container configuration management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

Table 11.7-1: Additional Kubernetes® API feature profiling against OS container configuration management service interface requirements

API feature	Description	Requirement identifier from ETSI GS NFV-IFA 040 [2]
Kubernetes® API Access Control [8]	Users and Kubernetes® service accounts can be authenticated and authorized for API access.	CismCfgMgt.023

12 OS container image management service interface

12.1 Description

This interface allows the API consumer to invoke OS container image management operations towards the API producer. The OCI™ Image [6] is identified to map to the OS container image of the NFV object model and consists of the resource objects as described in clause 4.2.3.2 of the present document.

The operations provided through this interface are:

- Push OS container image constituents
- Delete OS container image constituents
- Discover OS container images

12.2 API version

The API {VERSION} for the profiled OCI™ Distribution Specification [5] for OCI™ Image resource objects shall be set to "v2". Details on the OCI™ Distribution Specification API structure are specified in clause 4.2.3.2 of the present document.

The corresponding OCI™ Distribution Specification API root is specified as:

- /v2/{repository_name}

12.3 Resource structure and methods

Figure 12.3-1 shows the overall resource URI structure for the profiled OCI™ Distribution Specification API [5] for the OS container image management service interface.

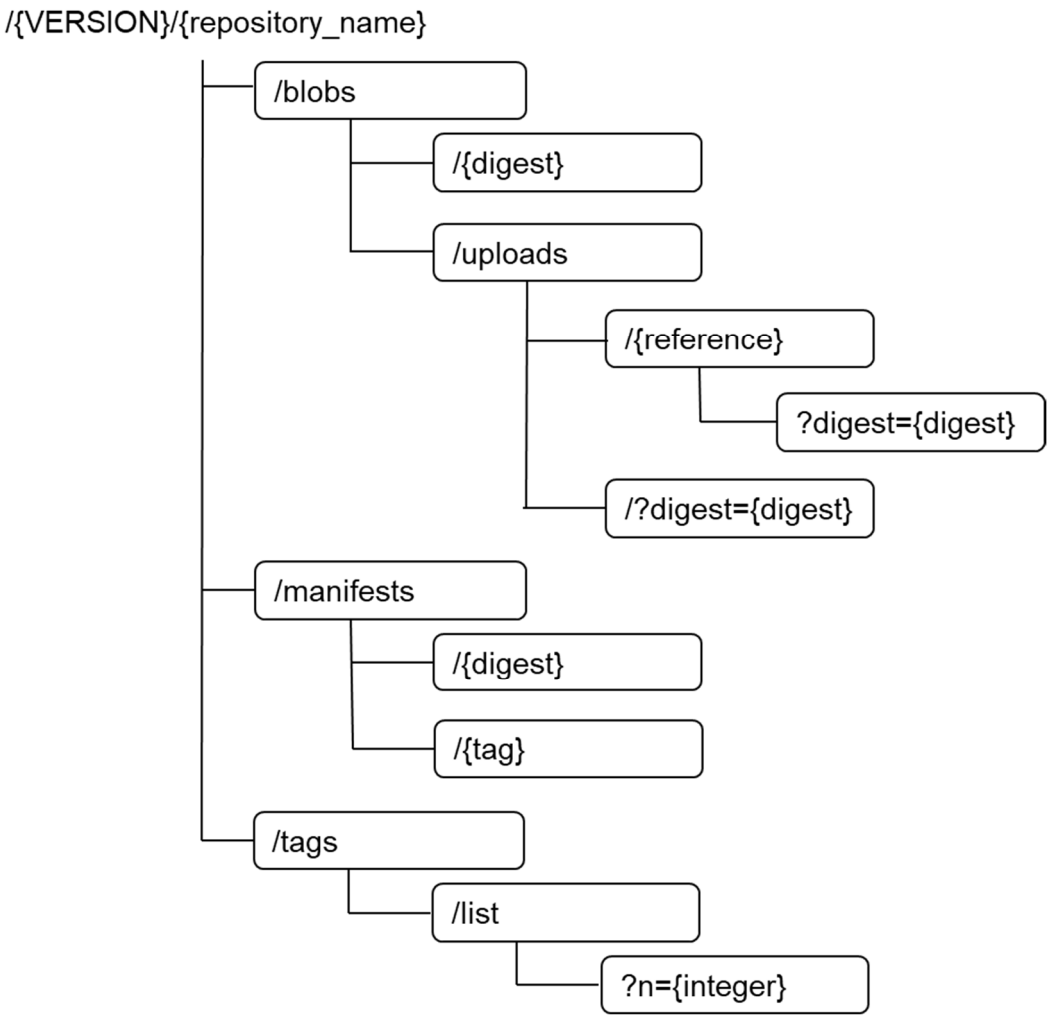

Figure 12.3-1: Resource URI structure for the OS container image management service interface

Table 12.3-1 lists the individual resources defined, and the applicable HTTP methods.

The CIR shall support responding to requests for all HTTP methods on the resources in Table 12.3-1 that are marked as "M" (mandatory) in the "Cat" column.

Table 12.3-1: Resources and methods overview of the OS container image management service interface

Resource name	Resource URI	HTTP Method	Cat	Meaning
Individual Blob instance	/blobs/{digest}	DELETE	M	Delete an "Individual Blob instance" resource.
	/blobs/uploads	POST	M	Obtain an upload location for an "Individual Blob instance" resource.
	/blobs/uploads/?digest={digest}	POST	M	Push an "Individual Blob instance" resource monolithically.
	/blobs/uploads/{reference}	PATCH	M	Push an "Individual Blob instance" resource chunk to an upload location.
	/blobs/uploads/{reference}?digest={digest}	PUT	M	Push an "Individual Blob instance" resource monolithically to an upload location.
Individual Manifest instance	/manifests/{digest}	PUT	M	Push an "Individual Manifest instance" resource with digest reference.
		DELETE	M	Delete an "Individual Manifest instance" resource.
	/manifests/{tag}	PUT	M	Push an "Individual Manifest instance" resource with tag reference.
OS container image tag	/manifests/{tag}	DELETE	M	Delete an OS container image tag.
	/tags/list	GET	M	Discover all OS container image tags stored in this repository, identified by their tags.
	/tags/list?n={integer}	GET	M	Discover a specified number of OS container image tags stored in this repository, identified by their tags.

12.4 Sequence diagrams (informative)

12.4.1 Flow of pushing OS container image constituent blob

This clause describes the sequences for pushing an individual OS container image blob resource to the CIR.

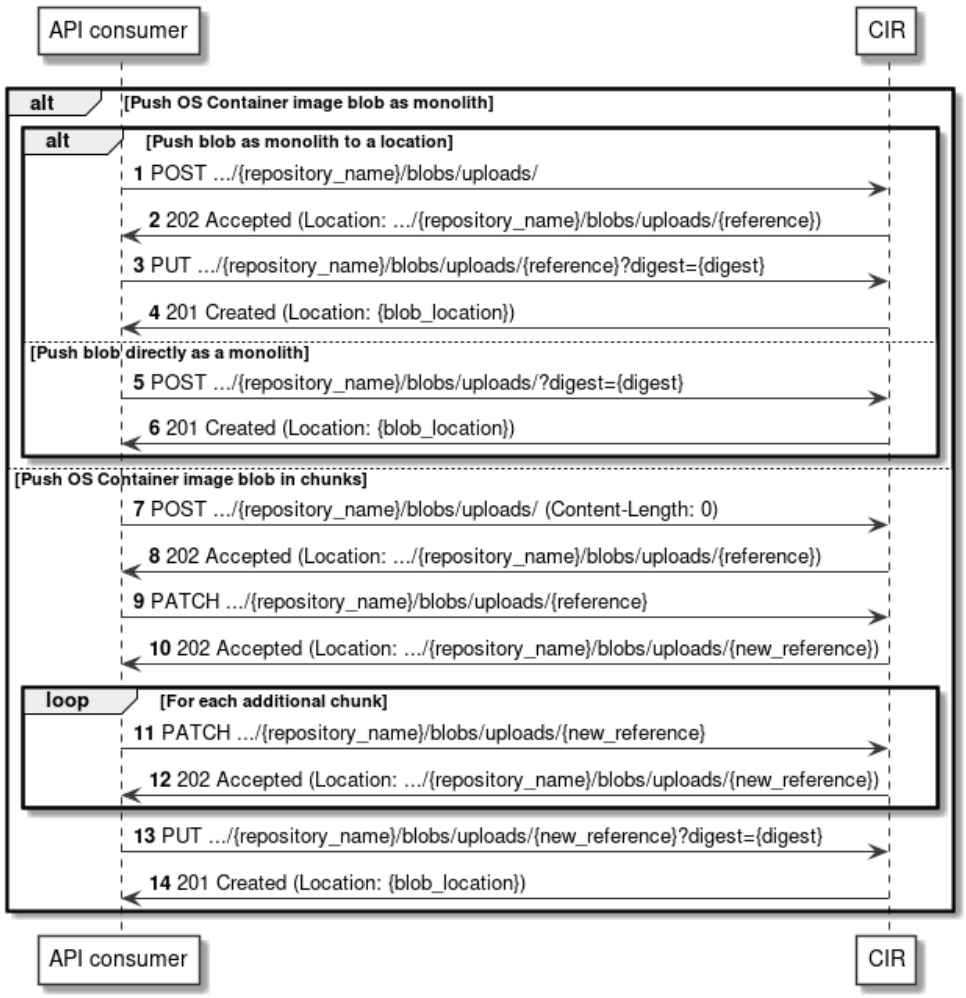

Figure 12.4.1-1: Flows of pushing an OS container image constituent blob

The pushing of an OS container image constituent blob, as illustrated in Figure 12.4.1-1, consists of the following steps.

Precondition: None.

Alternative 1: Push OS container image constituent blob as monolith to a location

1) The API consumer sends a POST request to the blob resource with the appropriate image repository name in the URI.

2) The CIR returns a "202 Accepted" response to the API consumer and includes in the "Location" header the URL of the upload target location for the blob.

3) The API consumer sends a PUT request to the received upload target location with the digest of the blob to be uploaded in the URI. The payload of the request contains the byte-stream of the blob.

4) The CIR returns a "201 Created" response to the API consumer and includes in the "Location" header the URL of the location for the blob.

Alternative 2: Push OS container image constituent blob directly as monolith

5) The API consumer sends a POST request to the blob resource with the appropriate image repository name and the digest of the blob to be uploaded in the URI.

6) The CIR returns a "201 Created" response to the API consumer and includes in the "Location" header the URL of the location for the blob.

Alternative 3: Push OS container image constituent blob in chunks

7) The API consumer sends a POST request to the blob resource with the appropriate image repository name in the URI and indicates a "Content-length" header with value "0".

8) The CIR returns a "202 Accepted" response to the API consumer and includes in the "Location" header the URL of the upload target location for the blob chunk.

9) The API consumer sends a PATCH request to the received upload target location. The payload of the request contains the byte-stream of the blob chunk.

10) The CIR returns a "202 Accepted" response to the API consumer and includes in the "Location" header the URL of a new upload target location for the next blob chunk.

Repeat steps 11) and 12) for each additional blob chunk to be uploaded:

11) The API consumer sends a PATCH request to the received new upload target location. The payload of the request contains the byte-stream of the blob chunk.

12) The CIR returns a "202 Accepted" response to the API consumer and includes in the "Location" header the URL of a new upload target location for the next blob chunk.

13) The API consumer sends a PUT request to the received upload target location with the digest of the complete blob uploaded in the URI.

14) The CIR returns a "201 Created" response to the API consumer and includes in the "Location" header the URL of the location for the blob.

Postcondition: Upon successful completion, the OS container image constituent blob has been stored in the CIR.

Error handling: In case of failure, appropriate error information is provided in the response.

12.4.2 Flow of pushing OS container image constituent manifest

This clause describes the sequences for pushing an individual OS container image manifest resource to the CIR.

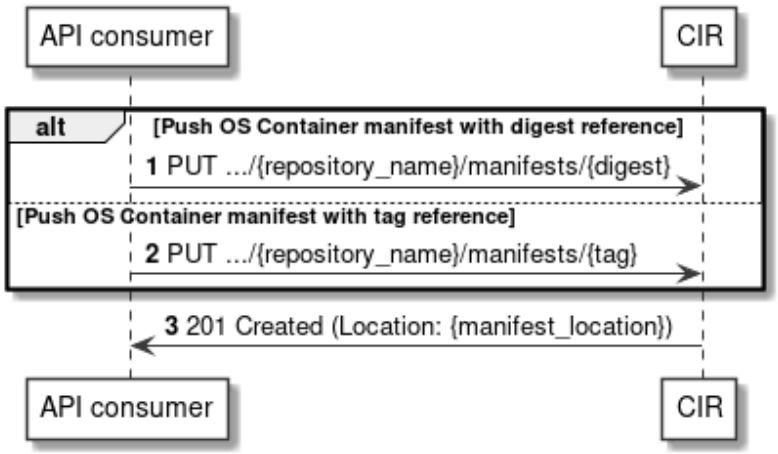

Figure 12.4.2-1: Flows of pushing an OS container image constituent manifest

The pushing of an OS container image constituent manifest, as illustrated in Figure 12.4.2-1, consists of the following steps.

Precondition: None.

Alternative 1: Push OS container image constituent manifest with digest reference

1) The API consumer sends a PUT request to the manifest resource with the appropriate image repository name and with the digest of the manifest to be uploaded in the URI.

Alternative 2: Push OS container image constituent manifest with tag reference

2) The API consumer sends a PUT request to the manifest resource with the appropriate image repository name and with the tag of the manifest to be uploaded in the URI.

3) The CIR returns a "201 Created" response to the API consumer and includes in the "Location" header the URL of the location for the manifest.

Postcondition: Upon successful completion, the OS container image constituent manifest has been stored in the CIR.

Error handling: In case of failure, appropriate error information is provided in the response.

12.4.3 Flow of deleting OS container image constituent blob

This clause describes the sequences for deleting an individual OS container image blob resource from the CIR.

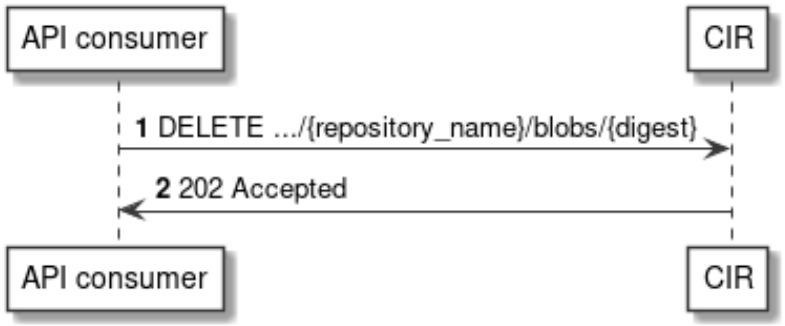

Figure 12.4.3-1: Flows of deleting an OS container image constituent blob

The deletion of an OS container image constituent blob, as illustrated in Figure 12.4.3-1, consists of the following steps.

Precondition: The individual OS container image blob resource is stored in the CIR.

1) The API consumer sends a DELETE request to the blob resource with the appropriate image repository name and the digest of the blob to be deleted in the URI.

2) The CIR returns a "202 Accepted" response to the API consumer.

Postcondition: Upon successful completion, the OS container image constituent blob has been deleted from the CIR.

Error handling: In case of failure, appropriate error information is provided in the response.

12.4.4 Flow of deleting OS container image constituent manifest

This clause describes the sequences for deleting an individual OS container image manifest resource from the CIR.

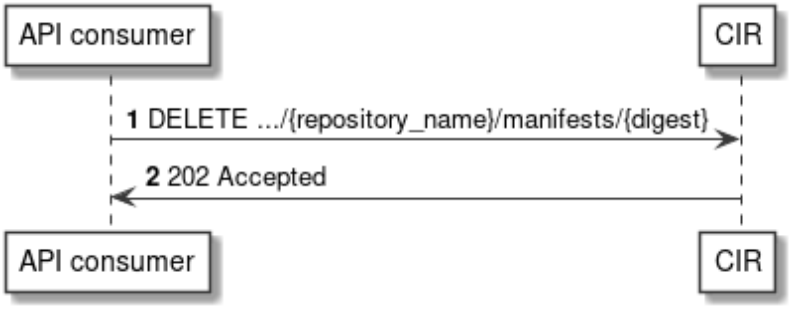

Figure 12.4.4-1: Flows of deleting an OS container image constituent manifest

The deletion of an OS container image constituent manifest, as illustrated in Figure 12.4.4-1, consists of the following steps.

Precondition: The individual OS container image manifest resource is stored in the CIR.

1) The API consumer sends a DELETE request to the manifest resource with the appropriate image repository name and the digest of the manifest to be deleted in the URI.

2) The CIR returns a "202 Accepted" response to the API consumer.

Postcondition: Upon successful completion, the OS container image constituent manifest has been deleted from the CIR.

Error handling: In case of failure, appropriate error information is provided in the response.

12.4.5 Flow of deleting OS container image tag

This clause describes the sequences for deleting an individual OS container image tag from the CIR.

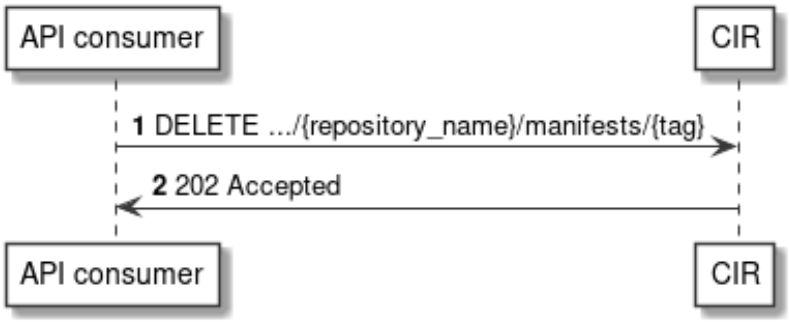

Figure 12.4.5-1: Flows of deleting an OS container image tag

The deletion of an OS container image tag, as illustrated in Figure 12.4.5-1, consists of the following steps.

Precondition: The individual OS container image tag is stored in the CIR.

1) The API consumer sends a DELETE request to the manifest resource with the appropriate image repository name and the tag to be deleted in the URI.

2) The CIR returns a "202 Accepted" response to the API consumer.

Postcondition: Upon successful completion, the OS container image tag has been deleted from the CIR.

Error handling: In case of failure, appropriate error information is provided in the response.

12.4.6 Flow of discovering OS container images

This clause describes the sequences for discovering OS container images which are stored in the CIR.

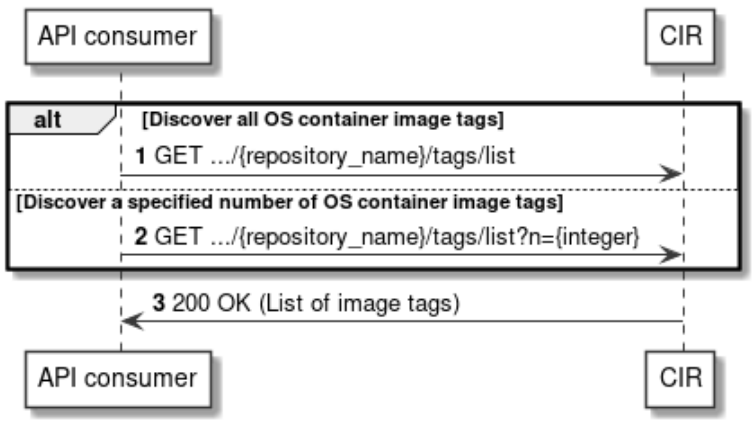

Figure 12.4.6-1: Flows of discovering OS container images

The discovery of OS container images which are stored in a CIR, as illustrated in Figure 12.4.6-1, consists of the following steps.

Precondition: none.

Alternative 1: Discover all OS container image tags

1) The API consumer sends a GET request to the tags list resource with the appropriate image repository name in the URI.

Alternative 2: Discover a specified number of OS container image tags

2) The API consumer sends a GET request to the tags list resource with the appropriate image repository name and the wanted number of tags to be received in the URI.

3) The CIR returns a "200 OK" response to the API consumer and includes in the payload body a list of the tags of the OS container images stored in the CIR.

Postcondition: None.

Error handling: In case of failure, appropriate error information is provided in the response.

12.5 Resources

12.5.1 Introduction

This clause profiles all the resources and methods provided by the OS container image management service interface.

12.5.2 Resource: Blob

This resource represents the OCI™ Distribution Specification resource object of a Blob, which is the binary form of an OS container image constituent that is stored by a registry, addressable by a digest.

Table 12.5.2-1 provides the profiling of the supported Blob resource methods against the OS container image management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the OCI™ Blob resource object specification of the profiled OCI™ Distribution Specification API [5].

Table 12.5.2-1: Blob resource methods profiling against OS container image management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/blobs/{digest}	DELETE	Delete an "Individual Blob instance" resource.	CirImgMgt.002
/blobs/uploads	POST	Obtain an upload location for an "Individual Blob instance" resource.	
/blobs/uploads/?digest={digest}	POST	Push an "Individual Blob instance" resource monolithically.	CirImgMgt.001
/blobs/uploads/{reference}	PATCH	Push an "Individual Blob instance" resource chunk to an upload location.	CirImgMgt.001
/blobs/uploads/{reference}?digest={digest}	PUT	Push an "Individual Blob instance" resource monolithically to an upload location.	CirImgMgt.001

12.5.3 Resource: Manifest

This resource represents the OCI™ Distribution Specification resource object of a Manifest, which is an OS container image constituent JSON document which specifies an artifact of an OS container image.

Table 12.5.3-1 provides the profiling of the supported Manifest resource methods against the OS container image management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the OCI™ Manifest resource object specification of the profiled OCI™ Distribution Specification API [5].

Table 12.5.3-1: Manifest resource methods profiling against OS container image management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/manifests/{digest}	PUT	Push an "Individual Manifest instance" resource with digest reference.	CirImgMgt.001
	DELETE	Delete an "Individual Manifest instance" resource.	CirImgMgt.002
/manifests/{tag}	PUT	Push an "Individual Manifest instance" resource with tag reference.	CirImgMgt.001

12.5.4 Resource: OS container image tag

This resource represents the OCI™ Distribution Specification resource object of an OS container image tag, which is a custom identifier of an OS container image that is stored by a registry.

Table 12.5.4-1 provides the profiling of the supported OS container image tag resource methods against the OS container image management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

The URI query parameters, request and response bodies, and response codes of the individual resource methods are described in the OCI™ OS container image tag resource object specification of the profiled OCI™ Distribution Specification API [5].

Table 12.5.4-1: OS container image tag resource methods profiling against OS container image management service interface requirements

Resource URI	HTTP Method	Meaning	Requirement identifier from ETSI GS NFV-IFA 040 [2]
/manifests/{tag}	DELETE	Delete an OS container image tag.	CirImgMgt.002
/tags/list	GET	Discover all OS container image tags stored in this repository, identified by their tags.	CirImgMgt.003
/tags/list?n={integer}	GET	Discover a specified number of OS container image tags stored in this repository, identified by their tags.	CirImgMgt.003

12.6 Data model

The request and response data structures of the OS container image management service interface are defined in the respective OCI™ resource object specifications of the profiled OCI™ Distribution Specification API [5] and the referenced OCI™ Image Format Specification [6].

12.7 Additional feature profiling

The OCI™ Distribution Specification API provides additional features which are not exposed via resource objects and corresponding methods. Instead, they are provided as functional capabilities.

Table 12.7-1 provides the profiling of the OCI™ Distribution Specification API additional features against the OS container image management service interface requirements as specified in ETSI GS NFV-IFA 040 [2].

Table 12.7-1: Additional OCI™ Distribution Specification API feature profiling against OS container image management service interface requirements

API feature	Description	Requirement identifier from ETSI GS NFV-IFA 040 [2]
OCI™ Distribution Specification [5], access control details in referenced appendix 1.	Realizations of the OCI™ Distribution Specification are required to provide an access controller which supports IETF RFC 7235 [i.6] compliant authorization headers.	CirImgMgt.005

Annex A (informative): Integration of the profiled solutions into the NFV-MANO framework

A.1 Concepts and evaluation

A.1.1 Releases of ETSI NFV own defined protocol and data modeling

New versions of ETSI NFV specifications, including the specifications defining protocol and data models such as the NFV-MANO RESTful APIs specified in ETSI GS NFV-SOL 003 [i.2] and data models for NFV descriptors based on TOSCA specified in ETSI GS NFV-SOL 001 [i.3], are typically released every 6 months. The release of new versions of specifications in ETSI NFV is documented in the ETSI NFV Release Description documentation made available on the ETSI NFV Open Area [i.4], which specifically refers within an NFV Release, which specification is updated and what version.

In terms of NFV-MANO RESTful APIs, another aspect is about the versioning of the specifications and APIs:

- ETSI NFV specification for NFV-MANO RESTful APIs: all deliverables, both drafts and published ones, have an associated version.
- RESTful NFV-MANO APIs: each API specified in the ETSI NFV specification is versioned. Information and guidelines about versioning of NFV-MANO RESTful APIs is provided in clause 9 of ETSI GS NFV-SOL 013 [i.5].

The two versions are independent. For instance, the specification deliverable containing an NFV-MANO RESTful API can be updated, while the version of the API specified in the deliverable remains the same because none of the changes have affected the actual resources and payloads exposed on the API.

Clause 9 of ETSI GS NFV-SOL 013 [i.5] specifies the meaning of the versioning pattern used as well as the rules for updating the API versions based on the changes/updates that are introduced into the respective API. In particular, non-backward compatible changes are represented by updates on the major digit of the version, such as updating from "v1" to "v2".

In terms of data model version of ETSI GS NFV-SOL 001 [i.3], annex B specifies the meaning of the template_version used in the type definitions files as well as the rules for updating the version number based on the changes in the type definitions.

A.1.2 Releases of referenced open source solutions

Kubernetes® and Helm™ are de-facto open source solutions which are frequently updated. The minor versions are expected to be released every three or four months, and the patch releases are provided for maintenance almost every month for approximately one year.

The present document profiles only stable Kubernetes® APIs; in case of the unstable APIs, "alpha" or "beta" is added to the version indicator, but in case of stable APIs, there is only the version indicator, e.g. "v1".

The stable API means that the fields of the API object is not removed within a major version of Kubernetes® even if the fields can be marked as deprecated. At the same time, within a major version, updates can be performed to add new attributes introduced by an enhancement. Typically, every minor version release includes lots of enhancements without any change on the version indicator of the API.

The update of Helm™ is aligned with the update of Kubernetes®. Therefore, the software component(s) hosting Helm™ CLI is (are) updated if a Kubernetes® cluster in which APIs are consumed by the software component(s) is updated; even if the software component(s) and the cluster are intended to be treated as different entities and separately updated, update-cycle coupling between them could happen depending on implementation.

The present document also profiles OCI™ Specification APIs. According to the OCI™ Charter [i.8], non-backward compatible changes are not recommended to be introduced and the release cadence depends upon the input of the community in terms of feature requests, security vulnerabilities and bug fixes.

A.1.3 Versioning correlation between referenced open source solutions, the present document and other NFV-MANO specifications

In the present annex, the frequent update of referenced open source solutions impact onto NFV-MANO is considered, e.g. whether the enhancements added to the referenced open source solution APIs can impact on NFV-MANO. The present document can provide the way of such an investigation because clause 6 provides the mappings between NFV-MANO related data objects (as specified in NFV-MANO RESTful APIs and NFV descriptors) and Kubernetes® API objects.

Based on the clause 6 mappings, it can be determined that:

- if there is any enhancement that impacts on NFV-MANO, a new mapping entry is expected to be added in the specified mapping tables; and
- if there is no new data model element to be reflected in the mapping tables due to updates and enhancements of Kubernetes®, this does not result in any update on the rest of NFV-MANO specifications.

NOTE: The present document does not provide guidelines or rules on how the profiled and referenced solutions can be adopted in a way that facilitates their integration with NFV-MANO framework solutions.

Annex B (informative): Mapping between NFV-MANO and Kubernetes®

B.1 Overview

This annex provides detail explanation on the mapping between NFV-MANO data models and Kubernetes® resource objects.

B.2 Detail explanation on Input parameter mapping

B.2.1 Images

In actual telecom deployment environments, CIR is likely to be privately deployed yet accessible for retrieving the OS container software images. For this purpose, on top of just an image name and its version, a registry hostname and a port number of CIR are also included in a value to specify the attribute "image" of Container v1 core in Kubernetes®. More precisely, the image name may include a repository name. Therefore, the value, which is finally passed as the "image", consists of multiple parts which are individually derived from different sources as described in Table 6.2.2.1-5. Figure B.2.1-1 illustrates the overview of handling information related to "image".

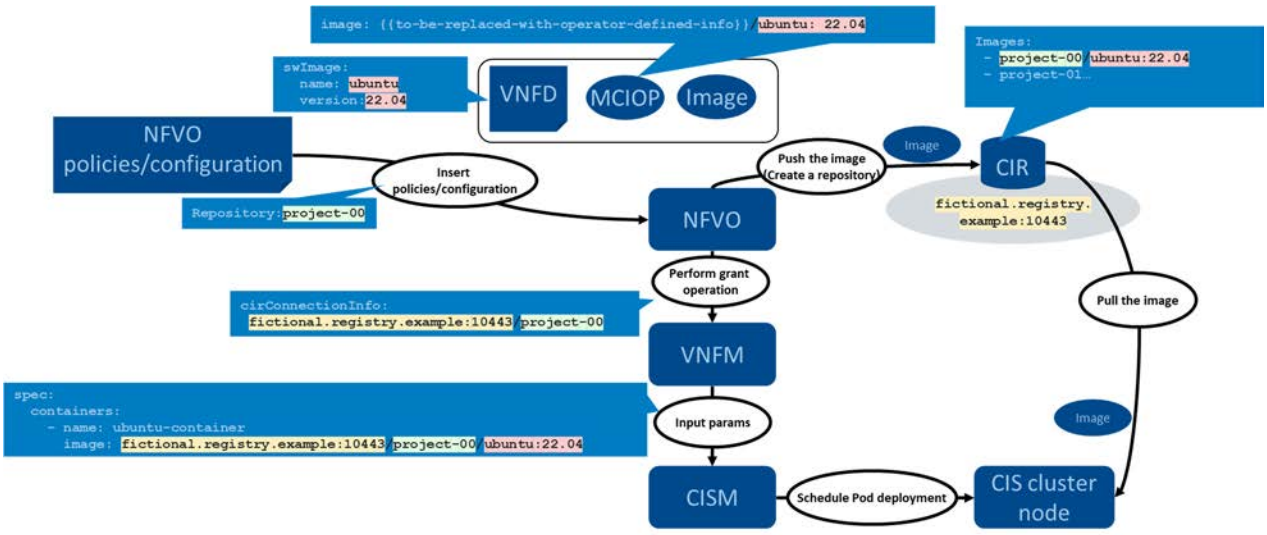

Figure B.2.1-1: Overview of handling information related to "image"

B.2.2 Affinity and anti-affinity

There are mainly following patterns on affinity/anti-affinity:

- Node affinity for VNFC/VNF
- Affinity and anti-affinity for intra/inter-VNFC

NOTE: In terms of affinity and anti-affinity rules, the present document does not specify the support to express affinities and anti-affinities between workloads (realizing VNFCs) deployed into different namespaces.

Figure B.2.2-1 illustrates an example of node affinity for VNFC/VNF. The Pod to be deployed based on the Deployment is expected to run on a CIS cluster node which has the capabilities: DPDK, SSD and NUMA. In the present specification, capabilities described in mcioConstraints are assumed to be evaluated in AND condition; therefore, the capabilities will be listed under one entry of matchExpressions as seen in the figure.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: with-node-affinity
spec:
  replicas: 1
  selector:
    matchLabels:
      app: xyz
  template:
    metadata:
      labels:
        app: xyz
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: nicAccel
                operator: In
                values:
                - dpdk
              - key: dsiktype
                operator: In
                values:
                - ssd
              - key: logicalNode
                operator: In
                values:
                - NUMA
      containers:
        - name: with-node-affinity
          image: k8s.gcr.io/pause:2.0
```

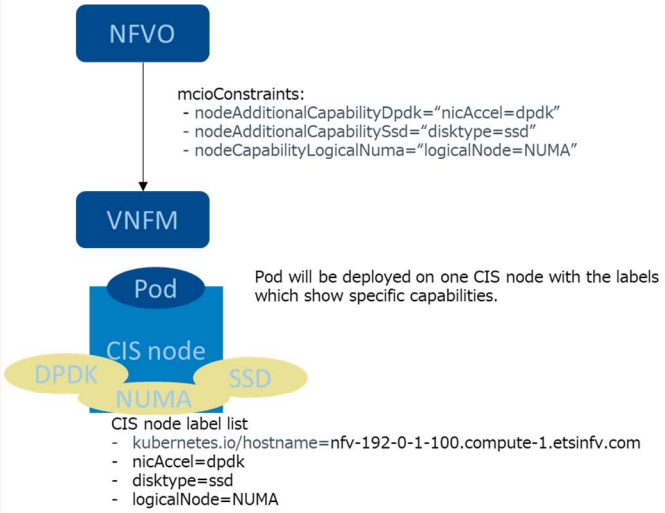

Table B.2.2-1: An example of node affinity for VNFC/VNF

Figure B.2.2-2 illustrates an example of affinity and anti-affinity for intra-VNFC. The three Pods to be deployed based on the Deployment is expected to run on the same CIS cluster node. If there are multiple scopes of affinity/anti-affinity, there will be as many entries described under requiredDuringSchedulingIgnoredDuringExecution as the scopes.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: with-pod-affinity
spec:
  selector:
    matchLabels:
      app: xyz
  replicas: 3
  template:
    metadata:
      labels:
        app: xyz
    spec:
      affinity:
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchExpressions:
              - key: app
                operator: In
                values:
                - xyz
            topologyKey: kubernetes.io/hostname
      containers:
      - name: with-pod-affinity
        image: k8s.gcr.io/pause:2.0
```

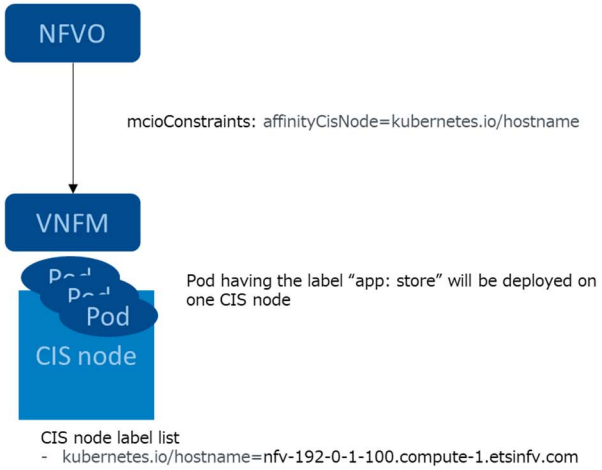

Table B.2.2-2: An example of affinity and anti-affinity for intra-VNFC

B.3 Detail explanation on Output parameter mapping

B.3.1 Pod name

To map information of Kubernetes® resource objects, especially Pod name, to NFV data objects, a consumer of Kubernetes® (i.e. VNFM) is expected to perform mapping update after specific triggers.

The below patterns are assumed as specific triggers:

a) Newly deploying of Deployment or StatefulSet into a target namespace.

b) Detection on (re)creation of resource objects, e.g. Pod, ReplicaSet, etc. in a target namespace:

b.1) By scheduled polling status of resource objects.

b.2) By watch mechanism set up to monitor status of resource objects.

After the specific triggers, mapping update is expected to be performed as the following:

NOTE 1: Here it is assumed that one Deployment or StatefulSet is created; in case of Deployment, the Deployment will create one ReplicaSet; the ReplicaSet or the StatefulSet will create multiple Pods based on the number of "replicas".

1) Send a request for a list of Pods in a target namespace (with a label selector based on each pattern if possible. See note 3, which is described in step 1) of clause 8.4.6.

EXAMPLE 1:

GET `/api/v1/namespaces/namespace-for-nfv/pods`

2) Receive a response for the list of the Pods.

EXAMPLE 2:

Pod names are "`app1-9456bbbf9-77ddh`", "`app1-9456bbbf9-cx94d`", "`app1-9456bbbf9-j9ckb`", "`app2-0`", "`app2-1`" and "`app2-2`".

3) Identify a relevant ReplicaSet or StatefulSet by referencing Pod's metadata.ownerReferences and if it is StatefulSet, go to step 7).

EXAMPLE 3:

i. Pod `app1-9456bbbf9-{77ddh, cx94d or j9ckb}`'s metadata.ownerReferences show "`kind: ReplicaSet`" and "`name: app1-9456bbbf9`".

ii. Pod `app2-{0,1 or 2}`'s metadata.ownerReferences show "`kind: StatefulSet`" and "`name: app2`".

4) Send a request to get an individual resource of ReplicaSet, which is described in step 3) of clause 8.4.6.

EXAMPLE 4:

GET `/api/v1/namespaces/namespace-for-nfv/replicasets/app1-9456bbbf9`

5) Receive a response for the individual resource of the ReplicaSet.

6) Identify a relevant Deployment by referencing ReplicaSet's metadata.ownerReferences.

EXAMPLE 5:

iii. ReplicaSet `app1-9456bbbf9`'s metadata.ownerReferences show "`kind: Deployment`" and "`name: app1`".

7) Send a request to get an individual resource of Deployment or StatefulSet, which is described in step 3) of clause 8.4.6.

EXAMPLE 6:

iv. GET `/api/v1/namespaces/namespace-for-nfv/deployments/app1`

v. GET `/api/v1/namespaces/namespace-for-nfv/statefulsets/app2`

8) Receive a response for the individual resource of the Deployment or the StatefulSet.

EXAMPLE 7:

vi.
```
{
 "kind": "Deployment",
 "status": {
 "availableReplicas": 3,
 "replicas": 3,
…
}
```

vii.
```
{
 "kind": "StatefulSet",
 "status": {
 "availableReplicas": 3,
 "replicas": 4,
…
}
```

9) Identify a relevant VDU and relevant VNFC instances.

EXAMPLE 8:

viii. The name in Vdu.OsContainerDeployableUnit in VDU A is app1; therefore, Deployment `app1` is associated with VNFC instances based on VDU A.

ix. The name in Vdu.OsContainerDeployableUnit in VDU B is app2; therefore, StatefulSet `app2` is associated with VNFC instances based on VDU B.

10) Map relevant information of the Deployment or the StatefulSet to McioInfo data type.

EXAMPLE 9:

x. "mcioType: Deployment", "mcioName: app1", "mcioNamespace: namespace-for-nfv", "availableInstances: 3" and "desiredInstances: 3".

xi. "mcioType: StatefulSet", "mcioName: app2" and "mcioNamespace: namespace-for-nfv", "availableInstances: 3" and "desiredInstances: 4".

11) Map Pod names to resourceIds of ResourceHandle data type (see note 2).

EXAMPLE 10:

xii. "`app1-9456bbbf9-77ddh`", "`app1-9456bbbf9-cx94d`" and "`app1-9456bbbf9-j9ckb`" are set as resourceIds.

xiii. "`app2-0`", "`app2-1`" and "`app2-2`" are set as resourceIds.

NOTE 2: Association of a specific instance of a Pod in a Deployment/StatefulSet to a specific instance of a VNFC is done by the VNFM. There are two types of association: the one is "Free" association (i.e. any Pod can be associated to any VNFC instance) which assumes that all Pod replicas in a Deployment/StatefulSet are all the same and therefore, VNFM can freely associate any of these to a VNFC instance and the other is "Guided" association which assumes that some specific metadata/characteristics of the Pod (e.g. its IP address) can be used to associated to a specific VNFC instance.

NOTE 3: According to the current specification of Kubernetes®, the field selector mechanism cannot support wildcards; the consumer cannot query Pods name by a field selector with regex, e.g. "metadata.name=nginx-*". Instead, if metadata of PodTemplateSpec is set up with the label of the parent object name, it is possible to query Pods name by a label selector, e.g. "app=nginx". Therefore, if the label selector can be assumed, an efficient filter can be performed at step 1); otherwise, it is necessary to get all Pods' name in the target namespace.
In case of Trigger A, the label selector is retrieved from a name of a resource object deployed into the target namespace; in case of Trigger B-1, determination of the label selector is dependent on e.g. the range of the polling (e.g. each kind of resource object in each namespace, each Deployment, each Pod, etc.); in case of Trigger B-2, the label selector is determined by parsing the name of the resource objects.

Annex (informative): Change History

Date	Version	Information about changes
January 2021	0.0.1	First version providing the document skeleton and scope.
April 2021	0.1.0	Implementing contributions: NFVSOL(21)000139r7 - SOL018 Clause 6.1 Introduction NFVSOL(21)000264 - SOL018-Clause 4.1-Summary of ETSI GS NFV-IFA 040 NFVSOL(21)000278r1 - SOL018-Clause 4.2.1-Overview profiled solution Kubernetes Table of content updated to reflect new content Minor editorial corrections to the new content
June 2021	0.2.0	Implementing contributions: NFVSOL(21)000356r2 - SOL018-Clause 4.2.2-Overview profiled solution Helm NFVSOL(21)000322r4 - SOL018 Clause 6.2 Input param framework Table of content updated to reflect new content
July 2021	0.3.0	Implementing contributions: NFVSOL(21)000390 - SOL018-Change the reference for the profiled solution for the CIR API NFVSOL(21)000392r1 - SOL018-Clause 4.2.3-Overview profiled solution OCI Distribution Specification Table of content updated to reflect new content
August 2021	0.4.0	Implementing contributions: NFVSOL(21)000405r3 - SOL018 Clause 6.2.2 workload NFVSOL(21)000431r1 - SOL018-Clause 5.1 MCIO type classifications Table of content updated to reflect new content
September 2021	0.5.0	Implementing contributions: NFVSOL(21)000449r1 - SOL018 Clause 6.2.4 config and storage for PVC NFVSOL(21)000460 - SOL018-Clause 5 NFV object model classifications Table of content updated to reflect new content
October 2021	0.6.0	Implementing contributions: NFVSOL(21)000471r1 - SOL018-Clause 6.2.2 Pod placement constraints NFVSOL(21)000532r2 - SOL018 Clause 6.2.3 Discovery and Loadbalancing with externalIP Table of content updated to reflect new content
November 2021	0.7.0	Implementing contributions: NFVSOL(21)000573 - SOL018 Clause 6.3 Framework NFVSOL(21)000580 - SOL018 Clause 6.3.2 Workload NFVSOL(21)000581 - SOL018 Resolve EN for hugepages NFVSOL(21)000592r1 - SOL018 Clause 6.2.3 Discovery and Loadbalancing with Ingress Text formatting according to ETSI styles. Table of content updated to reflect new content.
December 2021	0.8.0	Implementing contributions: NFVSOL(21)000572r2- SOL018 Clause 6.2.4 config and storage for CM and Secret NFVSOL(21)000591r3- SOL018 Clause 6.2.3 Discovery and Loadbalancing with LoadBalancer and NodePort service NFVSOL(21)000657r1- SOL018 Clause 6.3.3 Discovery and Loadbalancing Text formatting according to ETSI styles, including all editor's note converted into new ETSI style for EN. Table of content updated to reflect new content.
February 2022	0.9.0	Implementing contributions: NFVSOL(22)000015 - SOL018-Clause 8.1-Description compute management service interface NFVSOL(22)000016 - SOL018-Clause 8.3-Resource structure compute management service interface NFVSOL(22)000042 - SOL018 Clause 6 Resolve ENs Text formatting according to ETSI styles. Table of content updated to reflect new content.

Date	Version	Information about changes
March 2022	0.10.0	Implementing contributions: NFVSOL(22)000051 - SOL018-Clause 8.4.2-Flow description create Compute MCIO NFVSOL(22)000062r1- SOL018-Clause 8.4.3-Flow description modify Compute MCIO NFVSOL(22)000063r2- SOL018-Clause 8.4.4-Flow description replace Compute MCIO NFVSOL(22)000064r1- SOL018-Clause 8.4.5-Flow description delete Compute MCIO NFVSOL(22)000065- SOL018-Clause 8.4.5-Flow description get Compute MCIO information NFVSOL(22)000080r1- SOL018 Clause 6 Resolve ENs on image NFVSOL(22)000081- SOL018 Clause 6 Resolve ENs on namespace NFVSOL(22)000082r2- SOL018 Clause 6 Resolve ENs on affinity NFVSOL(22)000083r1- SOL018 Clause 6 Resolve ENs on deployment Text formatting according to ETSI styles. Table of content updated to reflect new content.
April 2022	0.11.0	Implementing contributions: NFVSOL(21)000357r4 - SOL018 Clause 6.2 Conveying param framework NFVSOL(22)000119 - SOL018-Clause 8.5-Resources Compute management service interface NFVSOL(22)000124 - SOL018-Clause 8.6-Data model Compute management service interface NFVSOL(22)000125 - SOL018-Clause 9.1-Description storage management service interface NFVSOL(22)000129 - SOL018-Clause 9.3-Resource structure storage management service interface NFVSOL(22)000144r1 - SOL018-Clause 9.5-Resources Storage management service interface NFVSOL(22)000145 - SOL018-Clause 10.1-Description network management service interface NFVSOL(22)000146 - SOL018-Clause 10.3-Resource structure network management service interface NFVSOL(22)000147 - SOL018-Clause 10.5-Resources Network management service interface Update of Disclaimer page according to new ETSI GS template Text formatting according to ETSI styles. Table of content updated to reflect new content.
April 2022	0.12.0	Implementing contributions: NFVSOL(21)000658r1 - SOL018 Clause 6.3.4 Configuration and Storage NFVSOL(22)000156r1 - SOL018 Clause 6 Resolve ENs on Host NFVSOL(22)000157r1 - SOL018 Clause 6 Resolve ENs on Path NFVSOL(22)000166 - SOL018-Clause 11.1-Description configuration management service interface NFVSOL(22)000167 - SOL018-Clause 11.3-Resource structure configuration management service interface NFVSOL(22)000168 - SOL018-Clause 11.5-Resources configuration management service interface NFVSOL(22)000176r1 - SOL018 Clarification on Supported cases of Affinity NFVSOL(22)000191 - SOL018-Clause 4.2.3-Refinements and resolve EN NFVSOL(22)000203 - SOL018 Clause 6.2.4 EN resolution NFVSOL(22)000205 - SOL018-Clause 12.1-Description image management service interface NFVSOL(22)000206 - SOL018-Clause 12.3-Resource structure image management service interface NFVSOL(22)000207 - SOL018-Clause 12.4.1-Flow descriptions push OS container constituents NFVSOL(22)000208 - SOL018-Clause 12.4.3-Flow descriptions delete OS container constituents NFVSOL(22)000209 - SOL018-Clause 12.4.6-Flow descriptions discover OS container images NFVSOL(22)000221 - SOL018 Clause 6.3.2 ResourceHandle for Pod Text formatting according to ETSI styles. Table of content updated to reflect new content.

Date	Version	Information about changes
May 2022	0.13.0	Implementing contributions: NFVSOL(22)000158r2 - SOL018 Clause 6 Resolve ENs on LoadBalancerIP NFVSOL(22)000084r1 - SOL018 Clause 6 Resolve ENs on service NFVSOL(22)000210 - SOL018-Clause 12.5-Resources image management service interface NFVSOL(22)000226r1 - SOL018 Clause 6 Resolve ENs on new type of services NFVSOL(22)000236 - SOL018-Clause 7.1-Description workload management service interface NFVSOL(22)000237 - SOL018-Clause 7.3.1-Flow description instantiate workload NFVSOL(22)000238 - SOL018-Clause 7.3.2-Flow description modify workload NFVSOL(22)000239 - SOL018-Clause 7.3.4-Flow description terminate workload NFVSOL(22)000240 - SOL018-Clause 7.3.5-Flow description query workload information NFVSOL(22)000241 - SOL018-Clause 7.4-Operations workload management service interface NFVSOL(22)000242 - SOL018 Clause 6.3.3 ResourceHandle for Service NFVSOL(22)000243 - SOL018 Clause 6.3.2 ResourceHandle for StatefulSet Text formatting according to ETSI styles. Table of content updated to reflect new content.
June 2022	0.14.0	Implementing contributions: NFVSOL(22)000250 - SOL018 Clause 6.3.3 ResourceHandle for Ingress NFVSOL(22)000251 - SOL018 Clause 6 ENs related to conveyed parameters NFVSOL(22)000252 - SOL018 Clause 6.2.3 EN for nodePort NFVSOL(22)000253 - SOL018 Annex A NFVSOL(22)000290 - SOL018 Some Editorials NFVSOL(22)000291r4 - SOL018 Add Management of Namespaces Text formatting according to ETSI styles. Table of content updated to reflect new content.
June 2022	0.15.0	Implementing contributions: NFVSOL(22)000285r3 - SOL018 Clause 6 Resolve ENs on Relationships between Services NFVSOL(22)000310r1 - SOL018 Clause 6 Resolve ENs on ExternalIP's NFVSOL(22)000316 - SOL018-Clause 8.7-Additional features compute management service interface NFVSOL(22)000317 - SOL018-Clause 9.7-Additional features storage management service interface NFVSOL(22)000318 - SOL018-Clause 10.7-Additional features network management service interface NFVSOL(22)000319 - SOL018-Clause 11.7-Additional features configuration management service interface NFVSOL(22)000320r1 - SOL018-Clause 12.7-Additional features image management service interface NFVSOL(22)000321 - SOL018-Clause 7.6-Additional features workload management service interface Text formatting according to ETSI styles. Table of content updated to reflect new content.
July 2022	0.16.0	Implementing contributions: NFVSOL(22)000329r1 - SOL018-Clause 6.2.3.1-Remove resolved editors note NFVSOL(22)000330 - SOL018- Clause 2.1-Update profiled solution versions Text formatting according to ETSI styles. Table of content updated to reflect new content.
August 2022	0.17.0	Implementing contributions: NFVSOL(22)000356r2 - SOL018 review clause 4.2 profiled solutions definition improvement NFVSOL(22)000357r1 - SOL018 Missing mapping for NAD NFVSOL(22)000358r1 - SOL018 Amendment of mapping for LoadBalancerIp NFVSOL(22)000360r1 - SOL018 resolving editor note related to SOL001 versioning NFVSOL(22)000363r1 - SOL018 clarify input/output mapping solution NFVSOL(22)000366 - SOL018 Resolve remaining ENs in Annex A Text formatting according to ETSI styles. Table of content updated to reflect new content.

History

Document history		
V4.3.1	September 2022	Publication