ETSI GS NFV-SOL 020 V5.2.1 (2025-03)

Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Specification of protocols and data models for Container Infrastructure Service Cluster Management

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
RGS/NFV-SOL020ed521

Keywords
container, data models, MANO, NFV, protocol

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 IPR online database.

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™**, **LTE™** and **5G™** logo 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 specifies the service interfaces to fulfil the functional requirements of the container infrastructure service cluster management (CIS Cluster Management, or CCM in abbreviation) as specified by ETSI GS NFV-IFA 036 [1]. It analyses the NFV object model for CIS cluster management and the managed objects of the referenced open-source solutions (e.g. Kubernetes® cluster API) and provide a mapping between them. Based on the analysis results, the referenced solutions are profiled for specifying the CIS cluster lifecycle, configuration, performance, fault and security management service interfaces.

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 in the ETSI docbox.

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 036: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for container cluster management and orchestration specification".

[2] Kubernetes® Cluster API v1.6.

[3] Kubernetes® API v1.28.

[4] ETSI GS NFV-IFA 040: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification".

[5] Kubernetes®: "API Access Control".

[6] Kubernetes®: "Auditing".

[7] ETSI GS NFV-IFA 045: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Faults and alarms modelling specification".

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 002: "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point".

[i.3] ETSI GS NFV-SOL 003: "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point".

[i.4] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Functional requirements specification".

[i.5] OpenStack® Tacker: "Virtualized Network Function Lifecycle Management Interface (VNF LCM)".

NOTE: Based on ETSI GS NFV-SOL 003 V3.3.1.

[i.6] OpenStack® Tacker: "Virtualized Network Function Performance Management Interface (VNF PM)".

NOTE: Based on ETSI GS NFV-SOL 003 V3.3.1.

[i.7] OpenStack® Tacker: "Virtualized Network Function Fault Management Interface (VNF FM)".

NOTE: Based on ETSI GS NFV-SOL 003 V3.3.1.

[i.8] Kubernetes® documentation: "Kubeadm".

[i.9] OpenStack®: "Tacker User Guide".

[i.10] ETSI GS NFV-SOL 014: "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; YAML data model specification for descriptor-based virtualised resource management".

[i.11] Kubernetes® Node Status.

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] apply.

3.2 Symbols

Void.

3.3 Abbreviations

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

NOTE: An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in ETSI GR NFV 003 [i.1].

CCM CIS Cluster Management
RBAC Role Based Access Control

4 Overview

4.1 Summary of ETSI GS NFV-IFA 036

ETSI GS NFV-IFA 036 [1] specifies the requirements on CIS cluster management service interfaces provided by the CCM function:

1) CIS cluster lifecycle management.

2) CIS cluster configuration management.

3) CIS cluster performance management.

4) CIS cluster fault management.

5) CIS cluster security management.

Lifecycle management of CIS clusters produced by the CCM includes creating, modifying, deleting and querying CIS clusters, as well as sending notifications about changes related to CIS clusters when consumer's subscribed events occur. CIS clusters can be based on virtualised or physical compute resources. CCM is able to manage virtual, bare-metal and hybrid CIS clusters.

Configuration management for the CIS cluster is produced by the CCM to its consumers during the lifecycle of the CIS cluster. The configuration management operations include transferring and applying configuration information to the CIS cluster, querying the configuration of the CIS cluster and subscribing/notifying the events in case of the change of CIS cluster configurations.

The CIS cluster performance management is produced by the CCM to enable its consumers to collect performance information on a given CIS cluster related object instance. The performance information results from performance information of the virtualised resources that is collected from the VIM or performance information of bare-metal resources that is collected from a management entity managing physical resources, and mapped by the CCM to this CIS cluster related object instance.

The CIS cluster fault management is produced by the CCM to enable its consumers to collect CIS cluster fault information. Virtualised resource alarms or physical resource alarms collected by the CCM will be filtered and correlated by the CCM and mapped to the corresponding CIS cluster related object instances (e.g. CIS cluster node, storage or network), resulting in alarms on corresponding CIS cluster.

Security management for the CIS cluster is produced by the CCM to its consumers. CCM is able to guarantee secure communication among CIS cluster nodes, authenticate and authorize invocating CISM capabilities from external and/or internal entities of CIS cluster by using configuration files and declarative descriptors representing Role Based Access Control (RBAC) related information, enable and configure the auditing of CIS cluster nodes, and configure encryption of confidential data services in the CIS cluster.

4.2 Summary of ETSI GS NFV-IFA 010

ETSI GS NFV-IFA 010 [i.4] specifies that the following statements on the scope of the CCM function, part of NFV-MANO, applies to all CCM related requirements:

- The CCM is responsible for the deployment, monitoring and lifecycle management of CIS clusters, the lifecycle management of CISM instances of the CIS cluster, and allocation of instantiated infrastructure resources for CIS cluster nodes.

- The CCM is responsible for the lifecycle management of MCCO for a CIS cluster, interacting directly with the CIS cluster nodes or via the CISM.

- The CCM exposes corresponding APIs to its consumers and translates incoming request into operations which are enforced towards the required actions to the CIS cluster.

- The CCM is further responsible to use information from declarative descriptors and configuration files of the CIS cluster.

The functional requirements on the CCM function are grouped into requirements per CCM management service:

- CIS cluster lifecycle;
- resource;
- configuration;
- descriptor;
- fault;
- performance; and
- capacity management.

According to the ETSI GS NFV-IFA 010 [i.4], more detailed information about the CCM services and the CIS cluster NFV object model is provided in ETSI GS NFV-IFA 036 [1].

4.3 Profiled protocol and data model solution: Kubernetes® Cluster API

4.3.1 Introduction

This clause provides an overview over the Kubernetes® Cluster API (Cluster API, or CAPI in abbreviation) which is profiled against the requirements of the CCM service interfaces as specified by ETSI GS NFV-IFA 036 [1]. The overview covers the high-level API structure as well as the concepts for the data model of the managed resource objects.

4.3.2 API structure

Kubernetes® Cluster API [2] is a Kubernetes® sub-project focused on providing declarative APIs and tooling to simplify provisioning, upgrading, and operating multiple Kubernetes® clusters. Started by the Kubernetes® Special Interest Group (SIG) Cluster Lifecycle as a Kubernetes® add-on project, Cluster API also uses Kubernetes®-style APIs and follows Kubernetes® basic patterns to leverage 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/*`).

The Kubernetes® API [3] supports read and write operations on the Kubernetes® resource objects via a Kubernetes® API endpoint. A **custom resource** is an extension of the Kubernetes® API that is not necessarily available in a default Kubernetes® installation. The Cluster API project defines a series of custom resource objects to automate cluster lifecycle management.

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

A mapping of the Cluster API managed custom resource objects to the NFV object models is provided in clause 5 of the present document.

4.3.3 Data model concepts

Following Kubernetes® [3] pattern for resource objects, the Cluster API managed custom resource objects are modelled with individual object schemas. All custom resource objects defined in Cluster API project typically have three components:

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

5 NFV object models mapping to profiled solution objects

5.1 CIS cluster object

Selected Kubernetes® Cluster API custom resource object is identified to map to the CIS cluster object of the NFV object model, see clauses 4.2 and 5.1 in ETSI GS NFV-IFA 036 [1]. Table 5.1-1 lists the Kubernetes® Cluster API custom resource objects which is mapped to the NFV CIS cluster object type.

Table 5.1-1: CAPI custom resource object mapped to NFV CIS cluster object

CAPI custom resource object kind	CAPI custom resource URI	CAPI custom resource object description
Cluster	/apis/cluster.x-k8s.io/v1beta1/clusters	Represents a Kubernetes® cluster with necessary configuration parameters. A Kubernetes® cluster is a grouping of Nodes. Kubernetes® could orchestrate containerized workload on these Nodes.

5.2 CIS cluster node object

Selected Kubernetes® Cluster API custom resource objects are identified to map to the CIS cluster node object of the NFV object model, see clauses 4.2 and 5.1 in ETSI GS NFV-IFA 036 [1]. Table 5.2-1 lists the Kubernetes® Cluster API custom resource objects which are mapped to the NFV CIS cluster node object type.

Table 5.2-1: CAPI custom resource objects mapped to NFV CIS cluster node object

CAPI custom resource object kind	CAPI custom resource URI	CAPI custom resource object description
Machine	/apis/cluster.x-k8s.io/v1beta1/machines	Represents an infrastructure component hosting a Kubernetes® Node (for example, a Virtual Machine (VM) or a Bare-Metal (BM) server).
MachineDeployment	/apis/cluster.x-k8s.io/v1beta1/machinedeployments	Represents a set of Machines
MachineSet	/apis/cluster.x-k8s.io/v1beta1/machinesets	Represents a stable set of Machines
MachinePool	/apis/cluster.x-k8s.io/v1beta1/machinepools	Similar to MachineDeployment, see clause 6.5.6 for specific differences.
KubeadmConfig	/apis/ bootstrap.cluster.x-k8s.io/v1beta1/kubeadmconfigs	Represents bootstrap configuration of Kubernetes® Node, which is bootstrap by Kubeadm. (see note)
NOTE: In CAPI, Kubeadm is not the only way for providing bootstrap configuration. Providers can offer their own types of bootstrap configurations instead of using the default CAPI custom resource object, KubeadmConfig. Other types of bootstrap configurations may depend on specific deployment environments.		

5.3 CISM instance object

Selected Kubernetes® Cluster API custom resource object is identified to map to the CISM instance object of the NFV object model, see clauses 4.2 and 5.1 in ETSI GS NFV-IFA 036 [1].

The CISM function offers OS container management services as described in ETSI GS NFV-IFA 040 [4]. A CIS cluster has one or more instances of the CISM which manages workloads deployed in its CIS cluster. The CISM instance is also responsible for managing the CIS instances. A CIS cluster has at least one CIS cluster node hosting a CIS instance and one CISM instance for the management [1].

In Kubernetes® Cluster API, the **control plane** is mapped to the NFV CISM instance object. The control plane in CAPI is a set of components that serve the Kubernetes® API [2]. In CAPI, Kubeadm [i.8] is used to bootstrap the control plane in the default method [2]. Table 5.3-1 lists the Kubernetes® Cluster API custom resource object which is mapped to the NFV CISM instance object type.

Table 5.3-1: CAPI custom resource object mapped to NFV CISM instance object

CAPI custom resource object kind	CAPI custom resource URI	CAPI custom resource object description
KubeadmControlPlane	/apis/controlplane.cluster.x-k8s.io/v1beta1/ kubeadmcontrolplanes	Represents a set of components that serve the Kubernetes® API, which is bootstrapped by Kubeadm. (see note)
KubeadmConfig	/apis/ bootstrap.cluster.x-k8s.io/v1beta1/kubeadmconfigs	Represents bootstrap configuration of Kubeadm control plane.
NOTE: In CAPI, Kubeadm is not the only way for bootstrapping the control plane. Providers could provide their own kind of control plane instead of the default CAPI custom resource object KubeadmControlPlane. Other kind of control plane (than KubeadmControlPlane) may have dependency on specific deployment environment.		

6 CIS cluster lifecycle management service interface

6.1 Description

This interface allows the API consumer to invoke CIS cluster lifecycle management operations towards the API producer. Kubernetes® Cluster API custom resource objects identified as CIS cluster related NFV objects (e.g. CIS cluster object, CIS cluster node object, and CISM instance object) are listed in clause 5 of the present document.

The operations provided through this interface are:

- Create CIS cluster.
- Query information about CIS cluster.
- Modify CIS cluster.
- Delete CIS cluster.
- Send notifications in the event of changes to a CIS cluster.

NOTE: ETSI GS NFV-IFA 036 [1] specifies that the CCM requests the NFVO to grant the process of resource management. This granting operation is not supported by the profiled solution CAPI, which is recognized as a gap in the analysis in the present document.

6.2 API version

The API {VERSION} for the profiled Kubernetes® Cluster API [2] custom resource objects identified as CIS cluster related NFV objects shall be set to "v1beta1". Details on the API structure are specified in clause 4.3.2 of the present document.

The corresponding Kubernetes® Cluster API roots are specified as:

/apis/cluster.x-k8s.io/v1beta1

/apis/controlplane.cluster.x-k8s.io/v1beta1

6.3 Resource structure and methods

Figures 6.3-1, 6.3-2, 6.3-3 and 6.3-4 show the overall resource URI structures for the profiled Kubernetes® Cluster API [2] for the CIS cluster lifecycle management service interface.

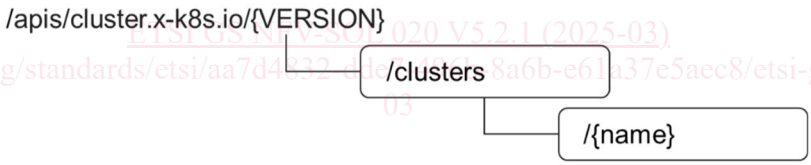

Figure 6.3-1: Resource URI structure of Cluster resource object for the CIS cluster lifecycle management service interface

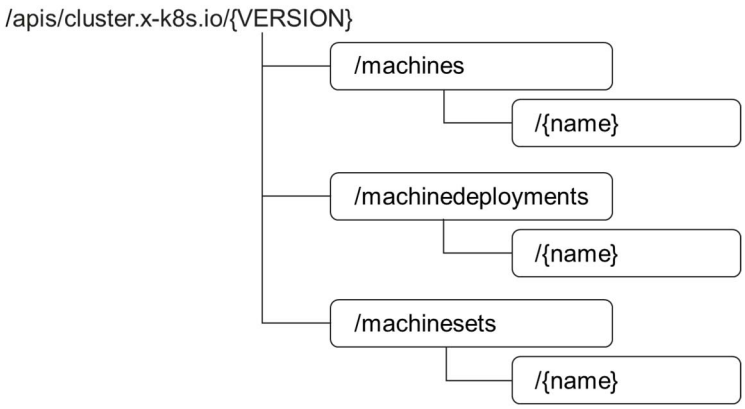

Figure 6.3-2: Resource URI structure of Machine, MachineDeployment and MachineSet resource objects for the CIS cluster lifecycle management service interface

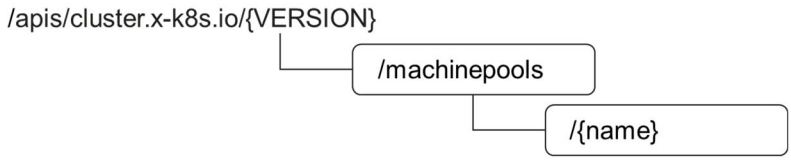

Figure 6.3-3: Resource URI structure of MachinePool resource objects for the CIS cluster lifecycle management service interface

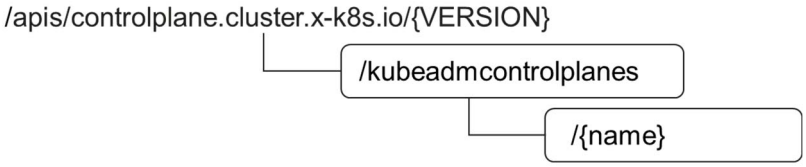

Figure 6.3-4: Resource URI structure of KubeadmControlPlane resource object for the CIS cluster lifecycle management service interface

Table 6.3-1 lists the individual resources defined, and the applicable HTTP methods.

The CCM supports responding to requests for all HTTP methods on the resources in Table 6.3-1 that are marked as "M" (mandatory) in the "Cat" column.

Table 6.3-1: Resources and methods overview of the CIS cluster lifecycle management service interface

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

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