ETSI GS NFV-SOL 014 V3.6.1 (2022-01)

Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; YAML data model specification for descriptor-based virtualised resource 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-SOL014ed361

Keywords
management, model, NFV

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 specifies a set of YAML-based data models for descriptor-based virtualised resource management fulfilling the requirements concerning the input and output information exchanged over the virtualised resource management interfaces specified in the ETSI GS NFV-IFA 005 [1], and the ETSI GS NFV-IFA 006 [2]. The present document focuses on data models used in the virtualised resource descriptors for the Virtualised Compute interfaces, Virtualised Network interfaces and Virtualised Storage interfaces, which are used to perform orchestration and lifecycle management for consumable virtualised resources comprised of compute, network and storage. The present document also focuses on data models used in the virtualised resource descriptors for the Virtualised Resources Change Notification interfaces and Virtualised Resources Fault Management interfaces. Other virtualised resource management interfaces, as well as data models for information specified in ETSI GS NFV-IFA 011 [i.5] and ETSI GS NFV-IFA 014 [i.4], are out of the scope of the present document.

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 005: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".

[2] ETSI GS NFV-IFA 006: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification".

[3] Void.

[4] "YAML Ain't Markup Language (YAML™) Version 1.2", 3rd Edition. Oren Ben-Kiki, Clark Evans, Ingy döt Net.

NOTE: Available at http://www.yaml.org/spec/1.2/spec.html.

[5] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

NOTE: Available at https://tools.ietf.org/html/rfc8259.

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

[7] JSON Schema.

NOTE: Available at https://json-schema.org/.

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

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 GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.2] Heat Orchestration Template (HOT) specification.

NOTE: Available at https://docs.openstack.org/heat/latest/template_guide/hot_spec.html.

[i.3] Openstack®-heat - Orchestration service APIs.

NOTE 1: Available at https://docs.openstack.org/api-ref/orchestration/.

NOTE 2: The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. ETSI is not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

[i.4] ETSI GS NFV-IFA 014: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Network Service Templates Specification".

[i.5] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; VNF Descriptor and Packaging Specification".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

3.2 Symbols

Void.

3.3 Abbreviations

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

JSON JavaScript Object Notation
YAML YAML Ain't Markup Language

4 General aspects

4.1 Overview

The present document defines the data model for the following interfaces used over the Vi-Vnfm and Or-Vi reference point, using YAML [4] as a data-serialization language:

- Virtualised Compute interfaces.
- Virtualised Network interfaces.
- Virtualised Storage interfaces.
- Virtualised Resources Change Notification interfaces.
- Virtualised Resources Fault Management interfaces.

The design of the data model for the above interfaces is based on the information model and requirements defined in ETSI GS NFV-IFA 005 [1] and ETSI GS NFV-IFA 006 [2]. Protocols that use these data models are out of the scope of the present version of the present document.

In clause 4, general aspects are specified that apply to multiple data model on the Vi-Vnfm and Or-Vi reference point. The present document defines data models for input and output parameters derived from the above-mentioned information model. The data instances are used as input and output parameters specified in a virtualised resource descriptor, e.g. a HOT [i.2]. As an alternative, output parameters can also be obtained from an API provided by the template system of the underlying VIM implementation, e.g. the HEAT API [i.3], and be mapped to the data model defined in the present document.

In the subsequent clauses, the data model of the parameters to be used in virtualised resource descriptors as input and output for the individual interfaces are specified. Annex A provides examples of the use of the input and output parameters using HOT [i.2].

4.2 Definition of input and output parameters in YAML

4.2.1 Introduction

Clause 4.2 specifies the types and section definitions in YAML that are applicable for the present document, in particular, for the declaration of the input and output parameters.

4.2.2 Input parameters syntax definition

The set of parameters that are used as input to an operation for which a corresponding template is defined shall be prefixed by a tag named "nfv" and shall comply with the following YAML syntax definition:

```
nfv:
  <parameter_name>:
    type: <the type of parameter>
    description: <description of the parameter>
    default: <default value of the parameter>
    enum:
      - <enumerated values 1>
      - <enumerated values 2>
    ...
  <parameter_name_N>:
    ...
```

Where applicable, then name of a structured input parameter ends with the string "Data" (e.g. subnetData). A description of the syntax definition fields for declaring an input parameter follows. The fields shall comply with the provisions set out in Table 4.2.2-1.

Table 4.2.2-1: Input parameters syntax definition

Field	Required	Description
nfv	yes	The tag emphasizes a group of parameters defined in the present document.
<parameter_name_1>	yes	The name of the first parameter.
<parameter_name_N>	no	The name of the last parameter.
type	yes	The type of each parameter. It shall be a simple data type as defined in clause 4.4.2 or structured data types in clause 4.4.3.
description	yes	A human readable description for each parameter.
default	no	A default value for each parameter.
enum	no	A set of enumerated values for a parameter to restrict the value. It is applicable to parameters of type string or number.

4.2.3 Output parameters syntax definition

If a set of output parameters of an operation is defined in a template, these parameters shall comply with the following YAML [4] syntax definition:

```
  <parameter_name>: value
    description: <description of the parameter>
    type: <type>
```

Where applicable, then name of a structured output parameter ends with the string "Info" (e.g. nfvSubnetInfo). A description of the syntax definition fields for declaring an output parameter follows. The fields shall comply with the provisions set out in Table 4.2.3-1.

Table 4.2.3-1: Output parameters syntax definition

Field	Required	Description
parameter_name	yes	The name of the parameter, which shall start with the prefix "nfv".
type	yes	The type of the parameter.
description	yes	A human readable description for the parameter.

4.3 Definition of output parameters as mapping to an API

The present document defines the set of attributes for each output parameter in the data model in clauses 6, 7 and 8. Besides providing the output parameters that are defined in the data model using the output parameters facility of a template (e.g. parameters in the "outputs" section of a HOT [i.2]), it is also possible to obtain these parameters via VIM-levels APIs such as (such as the HEAT API [i.3]). In the latter case, the output parameters of a VIM-level API can be mapped to the data model for the output parameters defined in the present document. Taking this approach can offer performance advantages in case many resources are required to be managed by the same template. The choice of the mapping of a parameter to a template output parameter, or to a VIM-level API is a deployment decision outside the scope of the present document.

4.4 Common data types

4.4.1 Introduction

Clause 4.4 specifies the common data types that are used for declaring the parameters and grammar elements throughout the present document.

4.4.2 Simple data types

The present document uses the following simple data types as defined in Table 4.4.2-1. In order to accommodate tags with a broader meaning, the YAML specification recommends JSON schema [7] to be supported as an option. JSON schema is commonly supported by modern computing languages. Virtualised resource descriptors complying with the present document shall comply with the YAML v1.2 [4] and JSON schema [7] specifications.

Table 4.4.2-1: Simple data types

Type name	Description	Example(s)
String	A string as defined in YAML v1.2 [4].	"a string"
Number	A number as defined in IETF RFC 8259 [5] referred in JSON Schema [7].	"23", "-1.023E3"
Boolean	A data type that can take the following values: true, false. The type is defined in JSON Schema [7] and referred in YAML v1.2 [4].	"true", "false"

4.4.3 Structured data types

Following the format stated with the label of "nfv" in Table 4.4.3-1, individual structured data type is represented in the present document using ">" recursively as inline definition.

Table 4.4.3-1: Input or Output data model for {parameter name}

Parameter Name and Attributes	Type	Description
{parameter name}	{object, array}	Type of the parameter
{description}	-	Description of the parameter
{attribute}	{attribute type}	Type of {attribute}
>{sub attribute}	{sub attribute type in the attribute}	Type of {sub attribute}

`object` in JSON schema [7] is a type representing mapping from "keys" to "values". The syntax of `object` for parameter definition is represented with the following definition:

```
{parameter name}:
  description: <description of the parameter>
  type: object
  required:
    - {1st mandatory attribute}
    - {2nd mandatory attribute}
    - …
  properties:
  {1st attribute}:
      type: e.g. object
      properties:
        {sub attribute}
  {2nd attribute}:
   …
```


`array` in JSON schema [7] is a type representing an ordered list of elements. The syntax of `array` for parameter definition is represented with the following definition:

```
{parameter name}:
  description: <description of the parameter>
  type: array
  minItems: {lower bound of cardinality}
  maxItems: {upper bound of cardinality}
  items:
  - type: e.g. object
    properties:
      {sub attribute}
```

5 Common data model

5.1 Description

This clause specifies data models for input and output parameters commonly used in different resource management.

5.2 Parameters to be used as input

5.2.1 Parameter: reservationId

The parameter used when pointing to a virtualised compute, network or storage resource shall follow the indications provided in Table 5.2.1-1.

Table 5.2.1-1: Input data model for reservationId

Parameter Name and Attributes	Type	Description
reservationId	String	Identifier of the resource reservation applicable to this virtualised resource management operation

The syntax of the reservationId shall comply with the following definition:

```
reservationId:
  type: string
  description: >
    Identifier of the resource reservation applicable to this virtualised resource
    management operation
  default: ""
```


5.2.2 Parameter: resourceGroupId

The parameter used when pointing to a logical grouping of virtual resources assigned to a tenant shall follow the indications provided in Table 5.2.2-1.

Table 5.2.2-1: Input data model for resourceGroupId

Parameter Name and Attributes	Type	Description
resourceGroupId	String	Unique identifier of the "infrastructure resource group", logical grouping of virtual resources assigned to a tenant within an Infrastructure Domain

The syntax of the resourceGroupId shall comply with the following definition:

```
resourceGroupId:
  description: >
    The identifier of the infrastructure resource group, logical grouping of virtual
    resources assigned to a tenant within an Infrastructure Domain of this
    virtualised resource management operation
  type: string
  default: ""
```

5.2.3 Parameter: groupName

The parameter used when giving a group name of a virtualised compute, network or storage resource affinity or anti-affinity constraints group to be created shall follow the indications provided in Table 5.2.3-1.

Table 5.2.3-1: Input data model for groupName

Parameter Name and Attributes	Type	Description
groupName	String	Name of the group, given by the consumer

The syntax of the groupName shall comply with the following definition:

```
groupName:
  type: string
  description: >
    Name of the group, given by the consumer
  default: ""
```

5.2.4 Parameter: typeOfAffinityOrAntiAffinityConstraints

The parameter used when indicating whether this is an affinity or anti-affinity group for virtualised compute, network or storage resources shall follow the indications provided in Table 5.2.4-1.

Table 5.2.4-1: Input data model for `typeOfAffinityOrAntiAffinityConstraints`

Parameter Name and Attributes	Type	Description
typeOfAffinityOrAntiAffinityConstraints	String	Indicates whether this is an affinity or anti-affinity group

The syntax of the `typeOfAffinityOrAntiAffinityConstraints` shall comply with the following definition:

```
typeOfAffinityOrAntiAffinityConstraints:
  description: >
    Indicates whether this is an affinity or anti-affinity group.
  type: string
  enum:
    - affinity
    - anti-affinity
```

5.2.5 Parameter: stackName

The parameter used when pointing to a stack of virtual resources defined by a descriptor shall follow the indications provided in Table 5.2.5-1.

Table 5.2.5-1: Input data model for stackName

Parameter Name and Attributes	Type	Description
stackName	String	Name of the stack, given by the consumer

The syntax of the stackName shall comply with the following definition:

```
stackName:
  type: string
  description: >
    Name of the stack, given by the consumer
  default: ""
```

5.3 Parameters to be used as output

None.

6 Data model for Virtualised Compute Management

6.1 Description

This clause specifies data models for input and output parameters for Virtualised Compute Management.

6.2 Parameters to be used as input

6.2.1 Parameter: computeName

The parameter used when providing a name for a virtualised compute resource to be allocated shall follow the indications provided in Table 6.2.1-1.

Table 6.2.1-1: Input data model for computeName

Parameter Name and Attributes	Type	Description
computeName	String	Name for a virtualised compute resource to be allocated

The syntax of the computeName shall comply with the following definition:

```
computeName:
  type: string
  description: >
    Name provided by the consumer for the virtualised compute resource to
    allocate
  default: ""
```

6.2.2 Parameter: computeFlavourId

The parameter used when providing an identifier of the Compute Flavour for a virtualised compute resource to be allocated shall follow the indications provided in Table 6.2.2-1.

Table 6.2.2-1: Input data model for computeFlavourId

Parameter Name and Attributes	Type	Description
computeFlavourId	String	Identifier of the Compute Flavour that provides information about the particular memory, CPU and disk resources for virtualised compute resource to allocate

The syntax of the computeFlavourId shall comply with the following definition:

```
computeFlavourId:
  type: string
  description: >
    Identifier of the Compute Flavour that provides information about the particular
    memory, CPU and disk resources for virtualised compute resource to allocate
  default: ""
```

6.2.3 Parameter: vcImageId

The parameter used when providing an identifier of the virtualisation container software image for a virtualised compute resource to be allocated shall follow the indications provided in Table 6.2.3-1.

Table 6.2.3-1: Input data model for vcImageId

Parameter Name and Attributes	Type	Description
vcImageId	String	Identifier of the virtualisation container software image

The syntax of the vcImageId shall comply with the following definition:

```
vcImageId:
  type: string
  description: >
    Identifier of the virtualisation container software image
  default: ""
```