



Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; YAML data model specification for descriptor-based virtualised resource management

ETSI GS NFV-SOL 014 V5.2.1 (2024-12)

<https://standards.iteh.ai/catalog/standards/etsi/5b9d8bdd-98bb-4465-a935-227c2dc6d4e1/etsi-gs-nfv-sol-014-v5-2-1-2024-12>

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-SOL014ed521

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

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2024.
All rights reserved.

Contents

Intellectual Property Rights	6
Foreword.....	6
Modal verbs terminology.....	6
1 Scope	7
2 References	7
2.1 Normative references	7
2.2 Informative references.....	8
3 Definition of terms, symbols and abbreviations.....	8
3.1 Terms.....	8
3.2 Symbols.....	8
3.3 Abbreviations	8
4 General aspects.....	9
4.1 Overview	9
4.2 Definition of input and output parameters in YAML	9
4.2.1 Introduction.....	9
4.2.2 Input parameters syntax definition.....	9
4.2.3 Output parameters syntax definition	10
4.3 Definition of output parameters as mapping to an API	10
4.4 Common data types	10
4.4.1 Introduction.....	10
4.4.2 Simple data types	11
4.4.3 Structured data types.....	11
5 Common data model	12
5.1 Description	12
5.2 Parameters to be used as input.....	12
5.2.1 Parameter: reservationId	12
5.2.2 Parameter: resourceGroupId	12
5.2.3 Parameter: groupName	12
5.2.4 Parameter: typeOfAffinityOrAntiAffinityConstraints	13
5.2.5 Parameter: stackName	13
5.2.6 Parameter: startTime	13
5.2.7 Parameter: endTime	14
5.2.8 Parameter: expiryTime	14
5.3 Parameters to be used as output.....	14
6 Data model for Virtualised Compute Management.....	14
6.1 Description	14
6.2 Parameters to be used as input.....	14
6.2.1 Parameter: computeName.....	14
6.2.2 Parameter: computeFlavourId.....	15
6.2.3 Parameter: vcImageId	15
6.2.4 Parameter: locationConstraints	15
6.2.5 Parameter: affinityOrAntiAffinityConstraintsForCompute	16
6.2.6 Parameter: interfaceData.....	17
6.2.7 Parameter: computeId	18
6.2.8 Parameter: networkInterfaceNew	18
6.2.9 Parameter: networkInterfaceUpdate	19
6.2.10 Parameter: flavour.....	21
6.2.11 Parameter: userData	23
6.2.12 Parameter: computePoolReservation	24
6.2.13 Parameter: minAmount	25
6.2.14 Parameter: maxAmount	26
6.2.15 Parameter: computeHostProperties.....	26
6.2.16 Parameter: callbackUriForComputeHostCapacityChangeNotify	26

6.2.17	Parameter: inputFilterForComputeHostCapacityChangeNotify	27
6.2.18	Parameter: changeIdForComputeHostCapacityChangeNotify	28
6.2.19	Parameter: zoneIdForComputeHostCapacityChangeNotify	28
6.2.20	Parameter: resourceDescriptorForComputeHostCapacityChangeNotify	28
6.2.21	Parameter: capacityInformationForComputeHostCapacityChangeNotify	29
6.2.22	Parameter: existingVirtualComputeResourcesTermination	29
6.3	Parameters to be used as output	30
6.3.1	Parameter: nfVComputeInfo	30
7	Data model for Virtualised Network Management	35
7.1	Description	35
7.2	Parameters to be used as input	35
7.2.1	Parameter: networkResourceName	35
7.2.2	Parameter: networkResourceType	36
7.2.3	Parameter: typeNetworkData	36
7.2.4	Parameter: typeNetworkPortData	38
7.2.5	Parameter: typeSubnetData	39
7.2.6	Parameter: affinityOrAntiAffinityConstraintsForNetwork	40
7.2.7	Void	41
7.2.8	Parameter: locationConstraintsForNetwork	41
7.2.9	Parameter: queryNetworkFilter	42
7.2.10	Parameter: networkResourceId	42
7.2.11	Parameter: updateNetworkData	42
7.2.12	Parameter: updateSubnetData	44
7.2.13	Parameter: updateNetworkPort	46
7.2.14	Parameter: scopeOfAffinityOrAntiAffinityConstraintForNetwork	46
7.2.15	Parameter: typeRoutingResourceData	47
7.2.16	Parameter: mirroringJobName	48
7.2.17	Parameter: descriptionForDataFlowMirroringJob	48
7.2.18	Parameter: collectorDetails	49
7.2.19	Parameter: dataFlowDetails	49
7.3	Parameters to be used as output	50
7.3.1	Parameter: nfVNetworkInfo	50
7.3.2	Parameter: nfVSubnetInfo	51
7.3.3	Parameter: nfVNetworkPortInfo	52
7.3.4	Parameter: nfVRoutingResourceInfo	54
7.3.5	Parameter: nfVMirroringJob	55
8	Data model for Virtualised Storage Management	56
8.1	Description	56
8.2	Parameters to be used as input	56
8.2.1	Parameter: storageName	56
8.2.2	Parameter: affinityOrAntiAffinityConstraintsForStorage	56
8.2.3	Parameter: storageData	58
8.2.4	Parameter: updateStorageData	58
8.2.5	Parameter: storageOperation	59
8.2.6	Parameter: newSize	59
8.2.7	Parameter: scopeOfAffinityOrAntiAffinityConstraintsForStorage	60
8.3	Parameters to be used as output	60
8.3.1	Parameter: nfVStorageInfo	60
9	Data model for Virtualised Resources Change Notification	61
9.1	Description	61
9.2	Parameters to be used as input	62
9.2.1	Parameter: callbackUriForChangeNotify	62
9.2.2	Parameter: inputFilter	62
9.2.3	Parameter: changeId	62
9.2.4	Parameter: virtualisedResourceId	63
9.2.5	Parameter: virtualisedResourceGroupId	63
9.2.6	Parameter: endOfChange	63
9.2.7	Parameter: changeTime	64
9.2.8	Parameter: vimId	64
9.2.9	Parameter: changeType	64

9.2.10	Parameter: changedResourceData	65
9.3	Parameters to be used as output.....	65
10	Data model for Virtualised Resources Fault Management.....	65
10.1	Description	65
10.2	Parameters to be used as input.....	65
10.2.1	Parameter: callbackUriForFaultNotify	65
10.2.2	Parameter: filter	65
10.2.3	Parameter: alarm.....	66
10.3	Parameters to be used as output.....	68
Annex A (informative):	Examples using OpenStack® Heat Orchestration Template.....	69
A.1	Introduction	69
A.2	Overview	69
A.2.1	Introduction	69
A.2.2	Template structure	69
A.3	Examples	69
A.3.1	Example#1: Allocate Virtualised Compute Resource operation	69
A.3.2	Example#2: Allocate Virtualised Network Resource operation.....	75
A.3.3	Example#3: Allocate Virtualised Storage Resource operation.....	81
A.3.4	Example#4: Create Compute Flavour operation	84
A.3.5	Example#5: API mapping of output parameters for Allocate Virtualised Storage Resource operation.....	87
A.3.6	Example#6: OpenStack Heat API sequence.....	87
A.3.7	Example#7: Virtualised Resources Change Notification Interface Subscribe operation.....	89
A.3.8	Example#8: Virtualised Resources Fault Management Interface Subscribe operation	92
A.3.9	Example#9: Create Compute Resource Reservation operation	95
A.3.10	Example#10: Create Compute Host Reservation operation	97
A.4	Complex templates	98
Annex B (informative):	Explanations of concepts.....	99
B.1	Introduction	99
B.2	Concept of descriptor-based virtualised resource management	99
Annex C (informative):	Change history	101
History		102

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](https://standards.itsc.ai).

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 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 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 005](#): "Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".
- [2] [ETSI GS NFV-IFA 006](#): "Network Functions Virtualisation (NFV) Release 5; 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.
- [5] [IETF RFC 8259](#): "The JavaScript Object Notation (JSON) Data Interchange Format".
- [6] [ETSI GS NFV-SOL 001](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; NFV descriptors based on TOSCA specification".
- [7] [JSON Schema](#).
- [8] [ETSI GS NFV-SOL 013](#): "Network Functions Virtualisation (NFV) Release 5; Protocols and Data Models; Specification of common aspects for RESTful NFV MANO APIs".
- [9] [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] [Heat Orchestration Template \(HOT\) specification](#).

[i.3] [Openstack®-heat - Orchestration Service APIs](#).

NOTE: 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 5; Management and Orchestration; Network Service Templates Specification".

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

(<https://standards.iteh.ai>)

3 Definition of terms, symbols and abbreviations

3.1 Terms

ETSI GS NFV-SOL 014 V5.2.1 (2024-12)

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:

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-level 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 inlined 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
<code>typeOfAffinityOrAntiAffinityConstraints</code>	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
<code>stackName</code>	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: ""
```

5.2.6 Parameter: `startTime`

The parameter used when giving a date and time when the consumption of the resources starts shall follow the indications provided in Table 5.2.6-1.

Table 5.2.6-1: Input data model for `startTime`

Parameter Name and Attributes	Type	Description
<code>startTime</code>	String	Specifies when the consumption of the resources starts.

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

```
startTime:
  type: string
  description: >
    Specifies when the consumption of the resources starts
  default: ""
```

5.2.7 Parameter: endTime

The parameter used when giving a date and time when the reservation ends shall follow the indications provided in Table 5.2.7-1.

Table 5.2.7-1: Input data model for endTime

Parameter Name and Attributes	Type	Description
endTime	String	Specifies when the reservation ends (when the issuer of the request expects that the resources will no longer be needed) and used by the VIM to schedule the reservation.

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

```
endTime:
  type: string
  description: >
    Specifies when the reservation ends (when the issuer of the request expects that the resources
    will no longer be needed) and used by the VIM to schedule the reservation
  default: ""
```

5.2.8 Parameter: expiryTime

The parameter used when giving a date and time when the VIM can release the reservation in case no allocation request against this reservation was made shall follow the indications provided in Table 5.2.8-1.

Table 5.2.8-1: Input data model for expiryTime

Parameter Name and Attributes	Type	Description
expiryTime	String	Specifies when the VIM can release the reservation in case no allocation request against this reservation was made.

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

```
expiryTime:
  type: string
  description: >
    Specifies when the VIM can release the reservation in case no allocation request against this
    reservation was made
  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.