SOL002 - VNF Configuration interface

Overview

VNF Configuration interface of ETSI NFV SOL002 IMPORTANT: Please note that this file might be not aligned to the current version of the ETSI Group Specification it refers to. In case of discrepancies the published ETSI Group Specification takes precedence. Please report bugs to https://forge.etsi.org/bugzilla/buglist.cgi?component=Nfv-Openapis&list_id=61&product=NFV&resolution=---

Version information

Version : 1.1.0

License information

License : ETSI Forge copyright notice
License URL : https://forge.etsi.org/etsi-forge-copyright-notice.txt
Terms of service : null

URI scheme

BasePath : /vnfconfig/v1
Schemes : HTTP, HTTPS

Consumes

- `application/json`

Produces

- `application/json`

External Docs

Description : ETSI GS NFV-SOL 002 V2.4.1
URL : http://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/002/02.04.01_60/gs_NFV-SOL002v020401p.pdf

Paths

Read VNF/VNFC configuration from VNF.

```
GET /configuration
```

Description

The client can use this method to read configuration information about a VNF instance and/or its VNFC instances.

Responses

HTTP Code	Description	Schema
200	OK Configuration information about a VNF instance was read successfully. The response body shall contain a representation of the configuration resource.	Response 200
400	Bad Request If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or a syntactically incorrect payload body), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. — If the request contains a malformed access token, the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. — If there is an application error related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond with this response code.The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 400

HTTP Code	Description	Schema
401	Unauthorized If the request contains no access token even though one is required, or if the request contains an authorization token that is invalid (e.g. expired or revoked), the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. **Headers** : Content-Type (string) : The MIME type of the body of the response. WWW-Authenticate (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 401
403	Forbidden If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 403
404	Not Found If the API producer did not find a current representation for the resource addressed by the URI passed in the request, or is not willing to disclose that one exists, it shall respond with this response code. The "ProblemDetails" structure may be provided, including in the "detail" attribute information about the source of the problem, e.g. a wrong resource URI variable. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 404
405	Method Not Allowed If a particular HTTP method is not supported for a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 405

HTTP Code	Description	Schema
406	Not Acceptable If the "Accept" HTTP header does not contain at least one name of a content type that is acceptable to the API producer, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 406
422	Unprocessable Entity If the payload body of a request contains syntactically correct data (e.g. well-formed JSON) but the data cannot be processed (e.g. because it fails validation against a schema), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. NOTE 2: This error response code is only applicable for methods that have a request body. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 422
500	Internal Server Error If there is an application error not related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond withthis response code. The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 500
503	Service Unavailable If the API producer encounters an internal overload situation of itself or of a system it relies on, it should respond with this response code, following the provisions in IETF RFC 7231 [13] for the use of the "Retry-After" HTTP header and for the alternative to refuse the connection. The "ProblemDetails" structure may be omitted. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 503

Response 200

Name	Description	Schema
vnfConfigurationData *required*	This type represents configuration parameters of a VNF instance.	vnfConfigurationData

4

Name	Description	Schema
vnfcConfigurationData *optional*	Configuration parameters of the VNFC instances.	< vnfcConfigurationData > array

vnfConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
extCpConfig *optional*	Configuration parameters for the external CPs of the VNF instance.	< extCpConfig > array
vnfSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

extCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer

Name	Description	Schema
useDynamicAddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

vnfcConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
intCpConfig *optional*	Configuration parameters for the internal CPs of the VNFC instance.	< intCpConfig > array
vnfcInstanceId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfcSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

intCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicAddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

Response 400

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 401

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 403

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 404

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 405

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 406

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 422

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 500

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 503

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Modify VNF/VNFC configuration.

```
PATCH /configuration
```

Description

This method sets or modifies a configuration resource.

Parameters

Type	Name	Description	Schema
Body	**configModific ations** *required*	The parameter for the configuration modification.	configModifications

configModifications

Name	Description	Schema
vnfConfigurationData *optional*	This type represents configuration parameters of a VNF instance.	vnfConfigurationData
vnfcConfigurationData *optional*	Modifications to configuration data for certain VNFC instances. See NOTE 1 and NOTE 2. If present, the modifications of the "vnfcConfigurationData" attribute shall follow these provisions: Modifying an attribute that is an array of objects of type "VnfcConfigurationData". Assumptions: 1) "oldList" is the "VnfcConfigurationData" array to be modified and "newList" is the "VnfcConfigurationData" array that contains the changes. 2) "oldEntry" is an entry in "oldList" and "newEntry" is an entry in "newList". 3) A "newEntry" has a "corresponding entry" if there exists an "oldEntry" that has the same content of the "vnfcInstanceId" attribute as the "newEntry"; a "newEntry" has no corresponding entry if no such "oldEntry" exists. 4) In any array of "VnfcConfigurationData" structures, the content of "vnfcInstanceId" is unique (i.e. there shall be no two entries with the same content of "vnfcInstanceId"). Provisions: 1) For each "newEntry" in "newList" that has no corresponding entry in "oldList", the "oldList" array shall be modified by adding that "newEntry". ```\n 2) For each "newEntry" in "newList" that has a\n corresponding "oldEntry" in "oldList",\n the value of "oldEntry" shall be updated\n with the value of "newEntry" according to\n the rules of JSON Merge PATCH (see IETF RFC\n 7396).\n```	< vnfcConfigurationData > array

vnfConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
extCpConfig *optional*	Configuration parameters for the external CPs of the VNF instance.	< extCpConfig > array

Name	Description	Schema
vnfSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

extCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicAddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)

Name	Description	Schema
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

vnfcConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
intCpConfig *optional*	Configuration parameters for the internal CPs of the VNFC instance.	< intCpConfig > array
vnfcInstanceId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfcSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

intCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicAddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

Responses

HTTP Code	Description	Schema
200	OK The request was accepted and completed. The response body shall contain the parameters of the configuration modification that was applied to the configuration resource.	Response 200

HTTP Code	Description	Schema
400	Bad Request If the request is malformed or syntactically incorrect (e.g. if the request URI contains incorrect query parameters or a syntactically incorrect payload body), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. — If the request contains a malformed access token, the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. — If there is an application error related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond with this response code.The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 400
401	Unauthorized If the request contains no access token even though one is required, or if the request contains an authorization token that is invalid (e.g. expired or revoked), the API producer should respond with this response. The details of the error shall be returned in the WWW-Authenticate HTTP header, as defined in IETF RFC 6750 and IETF RFC 7235. The ProblemDetails structure may be provided. **Headers** : `Content-Type` (string) : The MIME type of the body of the response. `WWW-Authenticate` (string) : Challenge if the corresponding HTTP request has not provided authorization, or error details if the corresponding HTTP request has provided an invalid authorization token.	Response 401

HTTP Code	Description	Schema
403	Forbidden If the API consumer is not allowed to perform a particular request to a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided. It should include in the "detail" attribute information about the source of the problem, and may indicate how to solve it. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 403
404	Not Found If the API producer did not find a current representation for the resource addressed by the URI passed in the request, or is not willing to disclose that one exists, it shall respond with this response code. The "ProblemDetails" structure may be provided, including in the "detail" attribute information about the source of the problem, e.g. a wrong resource URI variable. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 404
405	Method Not Allowed If a particular HTTP method is not supported for a particular resource, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 405
406	Not Acceptable If the "Accept" HTTP header does not contain at least one name of a content type that is acceptable to the API producer, the API producer shall respond with this response code. The "ProblemDetails" structure may be omitted in that case. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 406
409	Conflict Another request is in progress that prohibits the fulfilment of the current request, or the current resource state is inconsistent with the request. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 409

HTTP Code	Description	Schema
412	Precondition Failed A precondition given in an HTTP request header is not fulfilled. Typically, this is due to an ETag mismatch, indicating that the resource was modified by another entity. The response body should contain a ProblemDetails structure, in which the "detail" attribute should convey more information about the error. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 412
416	Requested Range Not Satisfiable This code is returned if the requested byte range in the Range HTTP header is not present in the requested resource. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 416
422	Unprocessable Entity If the payload body of a request contains syntactically correct data (e.g. well-formed JSON) but the data cannot be processed (e.g. because it fails validation against a schema), the API producer shall respond with this response code. The "ProblemDetails" structure shall be provided, and should include in the "detail" attribute more information about the source of the problem. NOTE 2: This error response code is only applicable for methods that have a request body. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 422
500	Internal Server Error If there is an application error not related to the client's input that cannot be easily mapped to any other HTTP response code ("catch all error"), the API producer shall respond withthis response code. The "ProblemDetails" structure shall be provided, and shall include in the "detail" attribute more information about the source of the problem. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 500
503	Service Unavailable If the API producer encounters an internal overload situation of itself or of a system it relies on, it should respond with this response code, following the provisions in IETF RFC 7231 [13] for the use of the "Retry-After" HTTP header and for the alternative to refuse the connection. The "ProblemDetails" structure may be omitted. **Headers** : Content-Type (string) : The MIME type of the body of the response.	Response 503

Response 200

Name	Description	Schema
vnfConfigurationData *optional*	This type represents configuration parameters of a VNF instance.	vnfConfigurationData
vnfcConfigurationData *optional*	Modifications to configuration data for certain VNFC instances. See NOTE 1 and NOTE 2. If present, the modifications of the "vnfcConfigurationData" attribute shall follow these provisions: Modifying an attribute that is an array of objects of type "VnfcConfigurationData". Assumptions: 1) "oldList" is the "VnfcConfigurationData" array to be modified and "newList" is the "VnfcConfigurationData" array that contains the changes. 2) "oldEntry" is an entry in "oldList" and "newEntry" is an entry in "newList". 3) A "newEntry" has a "corresponding entry" if there exists an "oldEntry" that has the same content of the "vnfcInstanceId" attribute as the "newEntry"; a "newEntry" has no corresponding entry if no such "oldEntry" exists. 4) In any array of "VnfcConfigurationData" structures, the content of "vnfcInstanceId" is unique (i.e. there shall be no two entries with the same content of "vnfcInstanceId"). Provisions: 1) For each "newEntry" in "newList" that has no corresponding entry in "oldList", the "oldList" array shall be modified by adding that "newEntry". ```\n 2) For each "newEntry" in "newList" that has a\n corresponding "oldEntry" in "oldList",\n the value of "oldEntry" shall be updated\n with the value of "newEntry" according to\n the rules of JSON Merge PATCH (see IETF RFC\n 7396).\n```	< vnfcConfigurationData > array

vnfConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
extCpConfig *optional*	Configuration parameters for the external CPs of the VNF instance.	< extCpConfig > array

Name	Description	Schema
vnfSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

extCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicAddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)

Name	Description	Schema
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

vnfcConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
intCpConfig *optional*	Configuration parameters for the internal CPs of the VNFC instance.	< intCpConfig > array
vnfcInstanceId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfcSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

intCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicA ddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

Response 400

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 401

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 403

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 404

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 405

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 406

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 409

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 412

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 416

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer

Name	Description	Schema
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 422

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string

Name	Description	Schema
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 500

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Response 503

Name	Description	Schema
detail *required*	A human-readable explanation specific to this occurrence of the problem.	string

Name	Description	Schema
instance *optional*	A URI reference that identifies the specific occurrence of the problem. It may yield further information if dereferenced.	string (URI)
status *required*	The HTTP status code for this occurrence of the problem. The HTTP status code ([RFC7231], Section 6) generated by the origin server for this occurrence of the problem.	integer
title *optional*	A short, human-readable summary of the problem type. It should not change from occurrence to occurrence of the problem, except for purposes of localization. If type is given and other than "about:blank", this attribute shall also be provided. A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see [RFC7231], Section 3.4).	string
type *optional*	A URI reference according to IETF RFC 3986 [5] that identifies the problem type. It is encouraged that the URI provides human-readable documentation for the problem (e.g. using HTML) when dereferenced. When this member is not present, its value is assumed to be "about:blank".	string (URI)

Definitions

VnfConfigModifications

This type represents request parameters for the "Set Configuration" operation. * NOTE 1: At least one of "vnfConfigurationData" and "vnfcConfigurationData" shall be present. * NOTE 2: The VnfcConfiguration data type can only be used to modify the configuration of existing VNFC instances.

Name	Description	Schema
vnfConfigurationData *optional*	This type represents configuration parameters of a VNF instance.	vnfConfigurationData

Name	Description	Schema

vnfcConfigura tionData *optional*	Modifications to configuration data for certain VNFC instances. See NOTE 1 and NOTE 2. If present, the modifications of the "vnfcConfigurationData" attribute shall follow these provisions: Modifying an attribute that is an array of objects of type "VnfcConfigurationData". Assumptions: 1) "oldList" is the "VnfcConfigurationData" array to be modified and "newList" is the "VnfcConfigurationData" array that contains the changes. 2) "oldEntry" is an entry in "oldList" and "newEntry" is an entry in "newList". 3) A "newEntry" has a "corresponding entry" if there exists an "oldEntry" that has the same content of the "vnfcInstanceId" attribute as the "newEntry"; a "newEntry" has no corresponding entry if no such "oldEntry" exists. 4) In any array of "VnfcConfigurationData" structures, the content of "vnfcInstanceId" is unique (i.e. there shall be no two entries with the same content of "vnfcInstanceId"). Provisions: 1) For each "newEntry" in "newList" that has no corresponding entry in "oldList", the "oldList" array shall be modified by adding that "newEntry".
	
	``
	    2) For each "newEntry" in "newList" that has a
	  corresponding "oldEntry" in "oldList",
	      the value of "oldEntry" shall be updated
	  with the value of "newEntry" according to
	      the rules of JSON Merge PATCH (see IETF RFC
	  7396).
	``
		< vnfcConfigurationD ata > array

vnfConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
extCpConfig *optional*	Configuration parameters for the external CPs of the VNF instance.	< extCpConfig > array

Name	Description	Schema
vnfSpecificDa ta *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

extCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicA ddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)

Name	Description	Schema
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

vnfcConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
intCpConfig *optional*	Configuration parameters for the internal CPs of the VNFC instance.	< intCpConfig > array
vnfcInstanceId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfcSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

intCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicA ddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

VnfConfiguration

This type represents configuration parameters of a VNF instance and its VNFC instances.

Name	Description	Schema
vnfConfigurat ionData *required*	This type represents configuration parameters of a VNF instance.	vnfConfigurationDat a
vnfcConfigura tionData *optional*	Configuration parameters of the VNFC instances.	< vnfcConfigurationD ata > array

vnfConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
extCpConfig *optional*	Configuration parameters for the external CPs of the VNF instance.	< extCpConfig > array
vnfSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

extCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicAddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)

vnfcConfigurationData

Name	Description	Schema
dhcpServer *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
intCpConfig *optional*	Configuration parameters for the internal CPs of the VNFC instance.	< intCpConfig > array
vnfcInstanceId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
vnfcSpecificData *optional*	This type represents a list of key-value pairs. The order of the pairs in the list is not significant. In JSON, a set of key-value pairs is represented as an object. It shall comply with the provisions defined in clause 4 of IETF RFC 7159.	object

intCpConfig

Name	Description	Schema
addresses *required*	Network address and port assigned to the CP.	< addresses > array
cpId *required*	An identifier that is unique for the respective type within a VNF instance, but may not be globally unique.	string
cpdId *required*	An identifier that is unique within a VNF descriptor.	string

addresses

Name	Description	Schema
address *optional*	Network address that has been configured on the CP. See NOTE 1.	address
port *optional*	The port assigned to the CP instance (e.g. IP port number, Ethernet port number, etc.).	integer
useDynamicA ddress *optional*	Set to true if an address shall be assigned dynamically. Otherwise set to false. The default value shall be false. See NOTE 1.	boolean

address

Name	Description	Schema
ipAddress *optional*	An IPV4 or IPV6 address. Representation: In case of an IPV4 address, string that consists of four decimal integers separated by dots, each integer ranging from 0 to 255. In case of an IPV6 address, string that consists of groups of zero to four hexadecimal digits, separated by colons.	string (IP)
macAddress *optional*	A MAC address. Representation: string that consists of groups of two hexadecimal digits, separated by hyphens or colons.	string (MAC)