OpenECOMP Application Controller (APPC) API Guide

Revision **Version 1.0.0**

Revision Date **1 February 2017**

Table of Contents

1 Introduction

This guide describes the APPC API that allows you to manage and control the life cycle of controlled virtual network functions (VNFs).

Target Audience

This document is intended for an advanced technical audience, such as the engineers or architects who need to use this guide to develop an interfacing application. The guide assumes a knowledge of the Open Enhanced Control, Orchestration, Management and Policy (OpenECOMP) components and features, and familiarity with JSON notation.

Conventions

Convention	Description

`Monospace`	This font indicates sample codes, screenshots, or elements. For example:
	`"contact": {`
	`  "contactType": "USER",`
	`  "source": "app1",`
	`}`

Italics	▪ Emphasizes a point or denotes new terms at the place where they are defined in the text. ▪ Indicates an external book title reference.

2 Life Cycle Management Commands

APPC receives commands from external OpenECOMP components, such as MSO, DCAE, or the Portal, to manage the life cycle of virtual applications and their components.

A virtual application is composed of the following layers of network technology:

- Service
- Virtual Network Function (VNF)
- Virtual Network Function Component (VNFC)
- Virtual Machine (VM)

A Life Cycle Management (LCM) command may affect one or more of these layers. For example, a `Terminate` command applies to all four layers, whereas a `Scale` command may not apply to the VM layer.

An LCM command is sent as a request to the APPC using an HTTP POST request, or in a message on OpenECOMP's bus (DMaaP). For each request, the APPC returns one or more responses:

- An **asynchronous** command, which is sent as an authorized and valid request, results in at least two discrete response events:
  - an accept response, to indicate that the request is accepted for processing
  - a final response to indicate the status and outcome of the request processing

  An unauthorized or invalid request results in a single ERROR response.
- A **synchronous** command, such as Lock or Unlock, results in a single response that is either SUCCESS or ERROR.

NOTE: For both asynchronous or synchronous commands, the first response is always returned using the same transport that the initial action used. For example, if the action request was via DMaaP (such as when it originates from DCAE), then the response is also via DMaaP. However, if the request was via a direct HTTP call, the response is similarly a synchronous HTTP response.

Message Format on the OpenECOMP Bus (DMaaP)

The APPC can receive an LCM request as a message on the OpenECOMP Bus (DMaaP) and returns one or more responses for each request.

The life cycle management interaction with APPC is conducted through OpenECOMP's event handling components, DMaaP, that expect to receive messages in the following particular format.

Request / Response Message

The LCM request or response is wrapped in the following message structure:

```
{
    "cambria.partition" : "<TOPIC>",
    "rpc-name" : "<RPC_NAME>",
    "body" : <RPC_SPECIFIC_BODY>
}
```

Table 1 Request / Response Message Fields

Field	Description	Required?
cambria. partition	Indicates the specific topic partition that the message is intended for. For example: ▪ For incoming messages, this value should be `APPC`. ▪ For outgoing messages, this value should be `MSO`.	No
rpc-name	The target Remote Processing Call (RPC) name which should match the LCM command name. For example: `modify-config`, `health-check`.	Yes

Field	Description	Required?

body	Contains the input or output LCM command content, which is either the request or response body. The body field format is identical to the equivalent HTTP Rest API command based on the specific RPC name. For example:
	`{`
	`    "input" : {`
	`    "common-header" : {...}`
	`    "action" : "ModifyConfig",`
	`    "action-identifiers" : {...},`
	`    "payload": "..."`
	`}`
		Yes

NOTE: For previous versions of the OpenECOMP message format, see Revision History.

Generic Request Format

This section describes the generic request format applicable to a general LCM command.

NOTE: Refer to specific LCM commands (on page 17) to see the format for that command. The specific implementation may overwrite the generic request structure or include extra data.

The request format is applicable for both POST HTTP API and for the message body received via the OpenECOMP bus.

LCM Request

The LCM request comprises a common header and a section containing the details of the LCM action.

The LCM request conforms to the following structure:

```
{
 "input": {
  "common-header": {
   "timestamp": "<TIMESTAMP>",
   "api-ver": "<API_VERSION>",
   "originator-id": "<ECOMP_SYSTEM_ID>",
   "request-id": "<ECOMP_REQUEST_ID>",
   "sub-request-id": "<ECOMP_SUBREQUEST_ID>",
   "flags": {
    "mode": "<EXCLUSIVE|NORMAL>",
    "force": "<TRUE|FALSE>",
    "ttl": "<TTL_VALUE>"
   }
  },
  "action": "<COMMAND_ACTION>",
  "action-identifiers": {
   "service-instance-id": "<ECOMP_SERVICE_INSTANCE_ID>",
   "vnf-id": "<ECOMP_VNF_ID>",
   "vnfc-name": "<ECOMP_VNFC_NAME>",
   "vserver-id": "VSERVER_ID"
  },
  ["payload": "<PAYLOAD>"]

  [ADDITIONAL PARAMETERS]
 }
```

Table 2 LCM Request Fields

Field	**Description**	**Required?**
input	The block that defines the details of the input to the command processing. Contains the `common-header` details.	Yes
common-header	The block that contains the generic details about a request.	Yes
timestamp	The time of the request, in ISO 8601 format, ZULU offset. For example: `2016-08-03T08:50:18.97Z`.	Yes
api-ver	Identifies the API version, in X.YY format, where X denotes the major version increased with each APPC release, and YY is the minor release version. For example: ▪ `2.00` for APIs added for OpenECOMP APPC 1.0.0	Yes

Field	Description	Required?
originator-id	An identifier of the calling system limited to a length of 40 characters. It can be used for addressing purposes, such as to return an asynchronous response to the correct destination, in particular where there are multiple consumers of APP-C APIs.	Yes
request-id	The UUID for the request ID, limited to a length of 40 characters. The unique OSS/BSS identifier for the request ID that triggers the current LCM action. Multiple API calls can be made with the same request-id. The request-id is stored throughout the operations performed during a single request.	Yes
sub-request-id	Uniquely identifies a specific LCM or control action, limited to a length of 40 characters. Persists throughout the life cycle of a single request.	No
flags	Generic flags that apply to all LCM actions: ▪ "MODE" : • "EXCLUSIVE" - accept no queued requests on this VNF while processing, or • "NORMAL" - queue other requests until complete ▪ "FORCE" : "TRUE"|"FALSE" - run action even if target is in an unstable state (for example, if VNF is busy processing another LCM command or if a previous command failed and VNF was indicated as not STABLE), or not. The specific behavior of forced actions varies, but implies cancellation of the previous action and an override by the new action. The default value is FALSE. ▪ "TTL" : <0....N> - The timeout value for the action to run, between action received by APPC and action initiated. If no TTL value provided, the default/configurable TTL value is to be used.	No

Field	Description	Required?
action	The action to be taken by APP-C, for example: Test, Start, Terminate. ***NOTE:*** *The specific value for the action parameter is provided for each* command *(on page 17).*	Yes
action-identifiers	A block containing the action arguments. These are used to specify the object upon which APP-C LCM command is to operate.	
service-instance-id	Identifies a specific service instance that the command refers to. When multiple APP-C instances are used and applied to a subset of services, this will become significant. The field is mandatory when the vnf-id is empty.	No
vnf-id	Identifies the VNF to which this action is to be applied (vnf-id uniquely identifies the service-instance referred to). Note that some actions are applied to multiple VNFs in the same service. When this is the case, vnf-id may be left out, but service-instance-id must appear. The field is mandatory when service-instance-id is empty.	No
vnfc-name	Identifies the VNFC to which this action is to be applied. Some actions apply only to a component within a VNF (for example, RESTART is sometimes applied only to a VM). In such cases, the name of the VNFC is used to search for the component within the VNF.	No
vserver-id	Identifies a specific VM within the given service or VNF to which this action is to be applied.	No

Field	Description	Required?
payload	An action-specific open-format field. The payload can be any valid JSON string value. JSON escape characters need to be added when an inner JSON string is included within the payload, for example: `"{\" vnf -host- ip -address\": \"<VNF-HOST-IP-ADDRESS>\"}"`. The payload decouples action parameters from the APPCinternal mechanism. Only DGs are able to parse the payload or act upon its contents. Note that not all LCM commands need have a payload. ***NOTE:*** *The specific value for the payload parameter, where relevant, is provided for the* command *(on page 17).*	No

Generic Response Format

This section describes the generic response format applicable to a general LCM command.

NOTE: Refer to specific LCM commands (on page 17) to see the format for that command. The specific implementation may overwrite the generic response structure or include extra data.

The response format is applicable for both POST HTTP API and for the message body received via the OpenECOMP bus.

LCM Response

The LCM response comprises a common header and a section containing the payload and action details.

The LCM response conforms to the following structure:

```
{
  "output": {
    "status": {
      "code": <RESULT_CODE>,
      "message": "<RESULT_MESSAGE>"
    },
    "common-header": {
      "api-ver": "<API_VERSION>",
      "request-id": "<ECOMP_REQUEST_ID>",
      "originator-id": "<ECOMP_SYSTEM_ID>",
      "sub-request-id": "<ECOMP_SUBREQUEST_ID>",
      "timestamp": "2016-08-08T23:09:00.11Z",
      "flags": {
        "ttl": <TTL_VALUE>,
        "force": "<TRUE|FALSE>",
        "mode": "<EXCLUSIVE|NORMAL>"
      }
    },
    ["payload": "<PAYLOAD>"]

    [ADDITIONAL PARAMETERS]
  }
}
```

Table 3 LCM Response Fields

Field	Description	Required?
output	The block that defines the details of the output of the command processing. Contains the `common-header` details.	Yes
common-header	The block that contains the generic details about a request.	Yes
timestamp	The time of the request, in ISO 8601 format, ZULU offset. For example: `2016-08-03T08:50:18.97Z`.	Yes
api-ver	Identifies the API version, in X.YY format, where X denotes the major version increased with each APPC release, and YY is the minor release version. For example: `2.00` for APIs added for OpenECOMP APPC 1.0.0	Yes

Field	Description	Required?
originator-id	An identifier of the calling system limited to a length of 40 characters. It can be used for addressing purposes, such as to return an asynchronous response to the correct destination, in particular where there are multiple consumers of APP-C APIs.	Yes
request-id	The UUID for the request ID, limited to a length of 40 characters. The unique OSS/BSS identifier for the request ID that triggers the current LCM action. Multiple API calls can be made with the same request-id. The request-id is stored throughout the operations performed during a single request.	Yes
sub-request-id	Uniquely identifies a specific LCM or control action, limited to a length of 40 characters. Persists throughout the life cycle of a single request.	No
status	The status describes the outcome of the command processing. Contains a `code` and a `message` providing success or failure details. ***NOTE:*** *See* Status Codes *(on page 10) for code values.*	Yes
code	A unique pre-defined value that identifies the exact nature of the success or failure status.	No
message	The description of the success or failure status.	No

Status Codes

The status code is returned in the response message as the `code` parameter, and the description as the `message` parameter

The different responses are categorized as follows:

ACCEPTED

Request is valid and accepted for processing.

ERROR

Request invalid or incomplete.

REJECT

Request rejected during processing due to invalid data, such as an unsupported command or a non-existent service-instance-id.

SUCCESS

Request is valid and completes successfully.

FAILURE

The request processing resulted in failure.

A FAILURE response is always returned asynchronously via DMaaP.

PARTIAL SUCCESS

The request processing resulted in partial success where at least one step in a longer process completed successfully.

A PARTIAL SUCCESS response is always returned asynchronously via DMaaP.

FAILURE

The request processing resulted in partial failure.

A PARTIAL FAILURE response is always returned asynchronously via DMaaP.

Table 4 Status Codes

Category	**Code**	**Message / Description**
ACCEPTED	100	ACCEPTED - Request accepted.
ERROR	200	UNEXPECTED ERROR - ${detailedErrorMsg}.
REJECT	300	REJECTED - ${detailedErrorMsg}
	301	INVALID INPUT PARAMETER -${detailedErrorMsg}

Category	Code	Message / Description
	302	MISSING MANDATORY PARAMETER - Parameter ${paramName} is missing
	303	REQUEST PARSING FAILED - ${detailedErrorMsg}
	304	NO TRANSITION DEFINE - No Transition Defined for ${actionName} action and ${currentState} state
	305	ACTION NOT SUPPORTED - ${actionName} action is not supported
	306	VNF NOT FOUND - VNF with ID ${vnfId} was not found
	307	DG WORKFLOW NOT FOUND - No DG workflow found for the combination of ${dgModule} module ${dgName} name and ${dgVersion} version
	308	WORKFLOW NOT FOUND - No workflow found for VNF type ${vnfTypeVersion} and ${actionName} action
	309	UNSTABLE VNF - VNF ${vnfId} is not stable to accept the command
	310	LOCKING FAILURE -${detailedErrorMsg}
	311	EXPIREDREQUEST. The request processing time exceeded the maximum available time.
	312	DUPLICATEREQUEST. The request already exists.
SUCCESS	400	The request was processed successfully.
FAILURE	401	DG FAILURE - ${ detailedErrorMsg }
	402	NO TRANSITION DEFINE - No Transition Defined for ${ actionName } action and ${ currentState } state
	403	UPDATE_AAI_FAILURE - failed to update AAI. ${ errorMsg }
	404	EXPIRED REQUEST FAILURE - failed during processing because TTL expired
	405	UNEXPECTED FAILURE - ${ detailedErrorMsg }
	406	UNSTABLE VNF FAILURE - VNF ${vnfId} is not stable to accept the command

Category	Code	Message / Description
PARTIAL SUCCESS	500	PARTIAL SUCCESS
PARTIAL FAILURE	501-599	PARTIAL FAILURE

Malformed Message Response

A malformed message is an invalid request based on its YANG scheme specification. APPC rejects malformed requests as implemented by ODL infrastructure level.

Response Format for Malformed Requests

```
{
  "errors": {
    "error": [
      {
        "error-type": "protocol",
        "error-tag": "malformed-message",
        "error-message": "<ERROR-MESSAGE>",
        "error-info": "<ERROR-INFO>"
      }
    ]
  }
}
```

Example Response

```
{
  "errors": {
    "error": [
      {
        "error-type": "protocol",
        "error-tag": "malformed-message",
        "error-message": "Error parsing input: Invalid value 'Stopp' for enum
 type. Allowed values are: [Sync, Audit, Stop, Terminate]",
        "error-info": "java.lang.IllegalArgumentException: Invalid value 'Stopp'
 for enum type. Allowed values are: [Sync, Audit, Stop, Terminate]..."
      }
    ]
  }
}
```

3 API Scope

Defines the level at which the LCM command operates for the current release of APPC.

Commands, or actions, can be performed at one or more of the following scope levels:

Service

Commands can be applied at the level of a specific service instance. This may include multiple VNFs, using a `service-instance-id`.

VNF

Commands can be applied at the level of a specific VNF instance. This may include multiple VNFCs, using a `vnf-id`.

VNFC

Commands can be applied at the level of a specific VNFC instance. This may include multiple VMs, using a `vnfc-name`.

VM

Commands can be applied at the level of a specific VM instance, using a `vserver-id`.

NOTE: To view the details of the API scope for a particular release, see Chapter 5, Revision History (on page 34).

Table 5 Command Scope

Command	Service	VNF	VNFC	VM
Audit	No	Yes	No	No
CheckLock	No	Yes	No	No
HealthCheck	No	Yes	No	No
LiveUpgrade	No	Yes	No	No
Lock	No	Yes	No	No
Migrate	No	No	No	Yes
ModifyConfig	No	Yes	No	No
Rebuild	No	No	No	Yes
Restart	No	No	No	Yes
Rollback	No	Yes	No	No
SoftwareUpload	No	Yes	No	No
Stop	No	Yes	No	No
Sync	No	Yes	No	No
Terminate	No	Yes	No	No
Test	No	Yes	No	No
Unlock	No	Yes	No	No

4 LCM Commands

Audit

The Audit command compares the configuration of the VNF associated with the current request against the configuration that is stored in APPC's configuration store.

A successful Audit means that the current VNF configuration matches the APPC stored configuration.

A failed Audit indicates that the request configuration is different from the stored configuration.

This command can be applied to any VNF type. The only restriction is that a particular VNF should be able to support the interface for Reading Configuration using existing adapters and use the following protocols: CLI, RestConf and XML.

The Audit action does not require any payload parameters.

NOTE: Audit does not return a payload containing details of the comparison, only the Success/Failure status.

Target URL	/restconf /operations/ appc-provider-lcm:audit
Action Parameter	Audit
Additional Parameters	None
Payload Parameters	None
Revision History	None.

CheckLock

The CheckLock command returns true if the specified VNF is locked, false if not.

A CheckLock command is deemed successful if the processing completes without error, whether the VNF is locked or not. The command returns only a single response with a final status.

The APPC also locks the target VNF during any command processing, so a VNF can have a locked status even if no Lock command has been explicitly called.

The CheckLock command returns a specific response structure that extends the default LCM response.

The CheckLock action does not require any payload parameters.

Target URL	/restconf/operations/appc-provider-lcm:checklock
Action Parameter	CheckLock
Additional Parameters	None
Payload Parameters	None
Revision History	None.

CheckLock Response

The CheckLock command returns a customized version of the LCM response.

The CheckLock response conforms to the standard response format (on page 8), but has the following additional field.

Table 6 Additional Parameters

Parameter	**Description**	**Requir**	**Example**
locked	"TRUE"|"FALSE" - returns TRUE if the specified VNF is locked, otherwise FALSE.	No	`"locked": "<TRUE|FALSE>"`

CheckLock Response Format

```
{
  "output": {
    "status": {
      "code": <RESULT_CODE>,
      "message": "<RESULT_MESSAGE>"
    },
    "common-header": {
      "api-ver": "<API_VERSION>",
      "request-id": "<ECOMP_REQUEST_ID>",
      "originator-id": "<ECOMP_SYSTEM_ID>",
      "sub-request-id": "<ECOMP_SUBREQUEST_ID>",
      "timestamp": "2016-08-08T23:09:00.11Z",
      "flags": {
        "ttl": <TTL_VALUE>,
        "force": "<TRUE|FALSE>",
        "mode": "<EXCLUSIVE|NORMAL>"
      }
    },
    "locked": "<TRUE|FALSE>"
  }
```

HealthCheck

This command runs a VNF health check and returns the result.

A health check is VNF-specific. For a complex VNF, APPC initiates further subordinate health checks.

HealthCheck is a VNF level command which interrogates the VNF in order to determine the health of the VNF and the VNFCs. The HealthCheck will be implemented differently for each VNF.

Target URL	/restconf/operations/appc-provider-lcm:health-check
Action Parameter	HealthCheck
Additional Parameters	None
Payload Parameters	vnf-host-ip-address (on page 20)
Revision History	None.

Table 7 Payload Parameters

Paramete	Description	Requir	Example
vnf-host-ip-address	The host IP where the check will be performed.	Yes	`"payload": "{\"vnf-host-ip-address\": \"10.222.22.2\"}"`

LiveUpgrade

The LiveUpgrade LCM action upgrades the target VNF to a new version without interrupting VNF operation.

A successful upgrade returns a success status.

A failed upgrade returns a failure code and the failure messages in the response payload block.

The payload includes the IP of the location that hosts the new software version installer file and the new software version.

Connections or operations that are active at the time of the LiveUpgrade action request will not be interrupted by the action and, therefore, the action may take a significant amount of time to run.

A LiveUpgrade is defined as non-disruptive; it is the responsibility of the VNF to handle disruptions if they occur.

Target URL	/restconf/operations/appc-provider-lcm:live-upgrade
Action Parameter	LiveUpgrade
Additional Parameters	None
Payload Parameters	upgrade-version (on page 21), vnf-host-ip-address (on page 21)
Revision History	None. New in 1610

Table 8 Payload Parameters

Parameter	Description	Required?	Example
upgrade-version	The host IP where the check will be performed.	Yes	`"payload": "{\"upgrade-version\": \"3.1\", \"vnf-host-ip-address\": \"10.222.22.2\"}"`
vnf-host-ip-address	The IP address that hosts the new software version installer file and the new software version.	Yes	

Lock

Use the Lock command to ensure exclusive access during a series of critical LCM commands.

The Lock action will return a successful result if the VNF is not already locked or if it was locked with the same `request-id`, otherwise the action returns a response with a reject status code.

Lock is a command intended for APPC and does not execute an actual VNF command. Instead, lock will ensure that OpenECOMP is granted exclusive access to the VNF.

When a VNF is locked, any subsequent sequential commands with same `request-id` will be accepted. Commands associated with other `request-id`s will be rejected.

The Lock command returns only one final response with the status of the request processing.

The APPC also locks the target VNF during any command processing. If a lock action is then requested on that VNF, it will be rejected because the VNF was already locked, even though no actual lock command was explicitly invoked.

Target URL	/restconf/operations/appc-provider-lcm:lock
Action Parameter	Lock
Additional Parameters	None
Payload Parameters	None
Revision History	None.

Migrate

Migrates a running target VFC from its current AIC host to another.

A destination AIC node will be selected by relying on AIC internal rules to migrate.

Migrate calls an AIC command in order to perform the operation.

Migrate suspends the guest virtual machine, and moves an image of the guest virtual machine's memory to the destination host physical machine. The guest virtual machine is then resumed on the destination host physical machine and the memory that it used on the source host physical machine is freed.

A successful Migrate action returns a success response and the new AIC node identity in the response payload block.

A failed Migrate action returns a failure and the failure messages in the response payload block.

NOTE: AIC refers to and the command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:migrate
Action Parameter	Migrate

Additional Parameters	None
Payload Parameters	vm-id (on page 23), identity-url (on page 23), tenant-id (on page 23)

Table 9 Payload Parameters

Parameter	Description	Required?	Example
vm-id	The unique identifier (UUID) of the resource. For backwards-compatibility, this can be the self-link URL of the VM.	Yes	`"payload": "{\"vm-id\": \"<VM-ID>\", \"identity-url\": \"<IDENTITY-URL>\", \"tenant-id\": \"<TENANT-ID>\"}"`
identity-url	The identity url used to access the resource	No	
tenant-id	The id of the provider tenant that owns the resource	No	

ModifyConfig

Use the ModifyConfig command when a full configuration cycle is either not required or is considered too costly.

The ModifyConfig LCM action affects only a subset of the total configuration data of a VNF. The set of configuration parameters to be affected is a subset of the total configuration data of the target VNF type. The payload block must contain the configuration parameters to be modified and their values.

The list of parameters that can be modified with this command is transparent to APPC and will be specific for each VNF type.

A successful modify returns a success response.

A failed modify returns a failure response and the specific failure messages in the response payload block.

Target URL	/restconf/operations/appc-provider-lcm:modify-config
Action Parameter	ModifyConfig
Additional Parameters	None
Payload Parameters	configuration-file-name, vnf-host-ip-address (on page 24)

Table 10 Payload Parameters

Parameter	Description	Required?	Example
configuration-file-name	The name of the configuration file that needs to be processed.	Yes	`"payload": "{\"configuration-file-name\": \"<CONFIGURATION-FILE-NAME>\", \"vnf-host-ip-address\": \"<VNF-HOST-IP-ADDRESS>\"}"`
vnf-host-ip-address	The host IP address used for running the configuration file.	Yes	

Rebuild

Recreates a target VFC instance to a known, stable state.

Rebuild calls an AIC command immediately and therefore does not expect any prerequisite operations to be performed, such as shutting off a VM.

A successful rebuild returns a success response and the rebuild details in the response payload block. A failed rebuild returns a failure and the failure messages in the response payload block.

NOTE: AIC refers to and the command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:rebuild
Action Parameter	Rebuild
Additional Parameters	None
Payload Parameters	vm-id (on page 23), identity-url (on page 23), tenant-id (on page 23)

Table 11 Payload Parameters

Parameter	Description	Required?	Example

vm-id	The unique identifier (UUID) of the resource. For backwards-compatibility, this can be the self-link URL of the VM.	Yes	`"payload":`
			`    "{\"vm-id\": \"<VM-ID>`
			`\",`
			` \"identity-url\":`
			` \"<IDENTITY-URL>\",`
			` \"tenant-id\": \"<TENANT-`
			`ID>\"}"`

identity-url	The identity url used to access the resource	No	
tenant-id	The id of the provider tenant that owns the resource	No	

Restart

Use the Restart command to restart a VNF or VM.

NOTE: The command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:restart

Action Parameter	Restart
Additional Parameters	None
Payload Parameters	vm-id (on page 23), identity-url (on page 23), tenant-id (on page 23)

Table 12 Payload Parameters

Parameter	Description	Required?	Example
vm-id	The unique identifier (UUID) of the resource. For backwards-compatibility, this can be the self-link URL of the VM.	Yes	`"payload": "{\"vm-id\": \"<VM-ID>\", \"identity-url\": \"<IDENTITY-URL>\", \"tenant-id\": \"<TENANT-ID>\"}"`
identity-url	The identity url used to access the resource	No	
tenant-id	The id of the provider tenant that owns the resource	No	

Rollback

Sets a VNF to the previous version of the configuration without explicitly invoking the configuration set name.

This command is used when the configuration was successful, but the health-check was not.

A successful rollback returns a success status when the restart process has completed.

A failed or a partially failed (for a complex VNF) rollback returns a failure and the failure messages in the response payload block.

This command can be applied to any VNF type. The only restriction is that the particular VNF should be built based on the generic heap stack.

Target URL	/restconf/operations/appc-provider-lcm:rollback
Action Parameter	Rollback
Additional Parameters	identity-url (on page 27), snapshot-id (on page 27)
Payload Parameters	None
Revision History	None.

Table 13 Additional Parameters

Parameter	Description	Required?	Example
identity-url	The identity URL used to access the resource. Enables the default AIC identity configured in APPC to be overridden.	No	`"identity-url" : "http://10.147.249.50:5000/v2.0)",`
snapshot-id	The unique identifier created by the Snapshot (on page 27) command.	Yes	`"snapshot-id" : "193838382",`

Snapshot

Creates a snapshot of a VNF, or VM.

The Snapshot command returns a customized response containing a reference to the newly created snapshot instance if the action is successful.

This command can be applied to any VNF type. The only restriction is that the particular VNF should be built based on the generic heap stack.

NOTE: The command implementation is based on Openstack functionality. For further details, see http://developer.openstack.org/api-ref/compute/.

Target URL	/restconf/operations/appc-provider-lcm:snapshot
Action Parameter	Snapshot
Additional Parameters	None
Payload Parameters	None

Snapshot Response

The Snapshot command returns an extended version of the LCM response.

The Snapshot response conforms to the standard response format (on page 8), but has the following additional field.

Table 14 Additional Parameters

Parameter	Description	Requir	Example
snapshot-id	The snapshot identifier created by AIC. This identifier will be returned only in the final success response returned via DMaaP.	No	`"snapshot-id": "<SNAPSHOT_ID>"`

SoftwareUpload

This LCM command uploads the file that contains a new software version to the target VNF.

This command is used in the context of a workflow that upgrades a VNF. The management of the file and versions is performed in a workflow outside the scope of APPC.

Target URL	/restconf/operations/appc-provider-lcm:software-upload
Action Parameter	SoftwareUpload
Additional Parameters	None
Payload Parameters	vnf-host-ip-address, source-file-url
Revision History	None.

Table 15 Payload Parameters

Parameter	Description	Requir	Example
source-file-url	The URL location of the source file.	Yes	`"payload": "{\"source-file-url\": \"<SOURCE-FILE-URL>\", \"vnf-host-ip-address \": \"<VNF-HOST-IP-ADDRESS> \"}"`
vnf-host-ip-address	The IP address that hosts the new software version installer file and the new software version.	Yes	

Stop

Stop a target VNF or VNFC.

A successful stop returns a success response. For a multi-component stop to be considered successful, all component stop actions must succeed.

A failed stop returns a failure and the failure messages in the response payload block.

Target URL	/restconf/operations/appc-provider-lcm:stop
Action Parameter	Stop
Additional Parameters	None
Payload Parameters	AICIdentity

Table 16 Payload Parameters

Parameter	Description	Required?	Example
AICIdentity	Specify a value for this parameter to override of the default AIC identify configured in APPC.	Yes	`"payload":"{\"AICIdentity \": \"http://10.147.249.50:5000/ v2.0\"}"`

Sync

The Sync action updates the current configuration in the APPC store with the running configuration from the device.

A successful Sync returns a success status.

A failed Sync returns a failure response status and failure messages in the response payload block.

This command can be applied to any VNF type. The only restriction is that a particular VNF should be able to support the interface for Reading Configuration using existing adapters and use the following protocols: CLI, RestConf and XML.

Target URL	/restconf/operations/appc-provider-lcm:sync
Action Parameter	Sync
Additional Parameters	None
Payload Parameters	None
Revision History	None.

Terminate

Terminate a target VNF and release its resources (possibly gracefully).

Specific scripts can be run before termination by placing them under the Terminate life cycle event.

All configuration files related to the target VNF are deleted.

The resources of a terminated VNF that are not managed by APPC, such as those handled by SDNC or other components, are not handled and remain the responsibility of their respective managing functions.

A successful Terminate action returns a success response. For a multi-component terminate to be considered successful, all component Terminate actions must also succeed.

A failed Terminate returns a failure status and the failure messages in the response payload block.

NOTE: A partially successful Terminate may leave a VNF in an indeterminate state.

Target URL	/restconf/operations/appc-provider-lcm:terminate
Action Parameter	Terminate
Additional Parameters	None
Payload Parameters	None

Test

The Test LCM action checks a target VNF or VNFC for correct operation.

The functionality of the Test LCM action involves should involve more than a HealthCheck , it should provide a means for launching a test transaction and determining if the transaction completed successfully or not. A transaction launcher microservice will have to be supplied by the VNF and called by APPC.

A successful test returns a success and the results of the test in the payload block. A failed test returns a failure and specific failure messages in the payload block

Target URL	/restconf/operations/appc-provider-lcm:test
Action Parameter	Test
Additional Parameters	None
Payload Parameters	vnf-host-ip-address
Revision History	None. New in 1610

Table 17 Payload Parameters

Paramete	Description	Requir	Example
vnf-host-ip-address	The host IP where the check will be performed.	Yes	`"payload": "{\"vnf-host-ip-address\": \"10.222.22.2\"}"`

Unlock

Run the Unlock command to release the lock on a VNF and allow other clients to perform LCM commands on that VNF.

Unlock is a command intended for APPC and does not execute an actual VNF command. Instead, unlock will release the VNF from the exclusive access held by OpenECOMP.

The Unlock command will result in success if the VNF successfully unlocked or if it was already unlocked, otherwise commands will be rejected.

The Unlock command will only return success if the VNF was locked with same request-id (on page 6).

The Unlock command returns only one final response with the status of the request processing.

The APPC also locks the target VNF during any command processing. If an Unlock action is then requested on that VNF with a different request-id, it will be rejected because the VNF is already locked for another process, even though no actual lock command was explicitly invoked.

Target URL	/restconf/operations/appc-provider-lcm:unlock
Action Parameter	Unlock
Additional Parameters	None
Payload Parameters	None
Revision History	None.

Appendix

Contains reference material relevant to this document.

Document History

Lists the tracked changes of the document versions.

Table 29 Document Version Tracking

Version	Date	Author	Change
