

vRealize Suite Lifecycle Manager 1.3 Programming Guide

vRealize Suite 2017



vmware®

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2018 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

vRealize Suite Lifecycle Manager Programming Guide	5
1 Overview of the vRealize Suite Lifecycle Manager REST API	6
Using the vRealize Suite Lifecycle Manager REST API	6
About the API Use Cases	7
2 Request an Authentication Token	8
3 Working with Data Centers	10
Create a Data Center	10
Add a vCenter	11
Update a vCenter Server	12
View Data Center Details	12
4 Getting Version Numbers	14
View the vRealize Suite Lifecycle Manager Version Number	14
View the vRealize Suite Lifecycle Manager API Version	15
5 Configuring vRealize Suite Lifecycle Manager Settings	16
Manage vRealize Suite Lifecycle Manager Common Configuration Settings	17
Add an Active Directory Server	17
Remove an Active Directory Server	18
Add a VMware Identity Manager	19
Update VMware Identity Manager Password Settings	20
Add a MyVMware Account	20
Update My VMware Credentials	21
6 Working with Environments	23
Pre-validating Before Creating a vRealize Suite Environment	24
Create an Environment	35
View an Environment and Details	52
Add a Product to an Existing Environment	53
Add a Node	54
Viewing Products and Product Details	56
Pre-Validate and Upgrade Products in Your vRealize Suite Environment	60
Collect Environment Logs	71
Configuration Drift	72

7 Working with Nodes 79

[View All Nodes and Node IDs 79](#)

[View Details for a Node 80](#)

[View Node Types for a Product 82](#)

[Node Types and Parameters 83](#)

8 Working with Requests 94

[Monitor a Request 94](#)

[Retry a Request 95](#)

vRealize Suite Lifecycle Manager Programming Guide

The *vRealize Suite Lifecycle Manager Programming Guide* provides information about the vRealize Suite Lifecycle Manager REST APIs, including how to use the REST API services and resources, create HTTP bearer tokens for authentication and authorization, and construct REST API service calls.

Intended Audience

This information is intended for administrators and programmers who want to configure and manage vRealize Suite Lifecycle Manager programmatically using the vRealize Suite Lifecycle Manager REST API. The guide focuses on common use cases. For related information about all available REST API services, see the vRealize Suite Lifecycle Manager API reference documentation, which is installed with vRealize Suite Lifecycle Manager at the following URL:

```
https://vRSLCMhost/api
```

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to <http://www.vmware.com/support/pubs>.

Overview of the vRealize Suite Lifecycle Manager REST API

1

You can perform vRealize Suite Lifecycle Manager functions programmatically by using REST API service calls. The API follows the REST style and is available to all licensed users.

- [Using the vRealize Suite Lifecycle Manager REST API](#)

To make vRealize Suite Lifecycle Manager REST API service calls, you can use a browser application or an HTTP client application to send requests and review responses.

- [About the API Use Cases](#)

Although the vRealize Suite Lifecycle Manager API reference contains a menu that lists all REST API service calls, it does not document use cases. The *vRealize Suite Lifecycle Manager Programming Guide* provides frequently used use cases.

Using the vRealize Suite Lifecycle Manager REST API

To make vRealize Suite Lifecycle Manager REST API service calls, you can use a browser application or an HTTP client application to send requests and review responses.

REST Client Programs

Any client application that can send HTTPS requests is an appropriate tool for developing REST applications with the vRealize Suite Lifecycle Manager API. The following open-source applications are commonly used:

- cURL. <http://curl.haxx.se>
- Postman application. <http://www.getpostman.com>

About the API Reference

The vRealize Suite Lifecycle Manager API reference lists all REST API service calls. It is provided as a Swagger document that is installed with vRealize Suite Lifecycle Manager at the following URL:

```
https://$LCM/api
```

\$LCM denotes an instance of vRealize Suite Lifecycle Manager.

About the API Use Cases

Although the vRealize Suite Lifecycle Manager API reference contains a menu that lists all REST API service calls, it does not document use cases. The *vRealize Suite Lifecycle Manager Programming Guide* provides frequently used use cases.

The following REST API use cases provide the prerequisite, command-line options, and format to help you perform various vRealize Suite Lifecycle Manager functions, such as creating an environment.

- [Request an Authentication Token](#)
- [Chapter 3 Working with Data Centers](#)
- [Chapter 4 Getting Version Numbers](#)
- [Configuring vRealize Suite Lifecycle Manager Settings](#)
- [Working With Environments](#)
- [Working With Nodes](#)
- [Chapter 8 Working with Requests](#)

Request an Authentication Token

2

You use a token to authenticate a vRealize Suite Lifecycle Manager API request.

All vRealize Suite Lifecycle Manager API requests require a valid authentication token in the header `x-xenon-auth-token`. To obtain a token, the login request supplies the user credentials in a form that Basic HTTP authentication requires. In this example, the user is logging in to a vRealize Suite Lifecycle Manager instance with URL `https://LCM-Hostname/`

Prerequisites

- Secure a channel between the web browser and the vRealize Suite Lifecycle Manager server. Open a browser and enter the URL such as:

```
https://LCM-Hostname/
```

The system warns that your connection is not private. Click through to confirm the security exception and establish an SSL handshake.

- Verify that you can log in to vRealize Suite Lifecycle Manager using the applicable credentials.

Procedure

- 1 Enter the command to request the HTTP bearer token.

```
curl 'https://LCM-Hostname/lcm/api/v1/login' \  
-H 'content-type: application/json' \  
-H 'accept: application/json' \  
--data-binary '${\n  "username": "admin@localhost",\n  "password": "VMware"\n}' \  
-k
```

The `-k` flag is included to verify the certificate.

2 Examine the response.

A successful request returns an HTTP bearer token that you include in subsequent API requests.

```
{
  "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ4biIsInN1YiI6Ii9jb3JlL2F1dGh6L3VzZXJzL3ZMQ01BZG1pb  
iIsImV4cCI6MTUwNzE4ODQwM30.dAcsjxo3qNkZfLaxwsGKHu1_5MqdEmJUfmNf6zQqpZ0"
}
```

3 For convenience, store the token in a variable.

```
export token="EXAMPLE-TOKEN-TEXT"
```

Working with Data Centers

You can use the vRealize Suite Lifecycle Manager API to create data centers and view data center details.

This chapter includes the following topics:

- [Create a Data Center](#)
- [Add a vCenter](#)
- [Update a vCenter Server](#)
- [View Data Center Details](#)

Create a Data Center

Create a data center to back vRealize Suite environments in vRealize Suite Lifecycle Manager.

Prerequisites

Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Log in to vRealize Suite Lifecycle Manager.

```
curl POST "https://LCM-Hostname/lcm/api/v1/login" \
```

- 2 Submit a request for a new data center with parameters inline.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/action/create/datacenter" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{"datacenterName": "$name",
    "city": "$city",
    "country": "$country",
    "latitude": "$latitude",
    "longitude": "$longitude",
    "state": "$state"}'
```

- 3 Examine the response to verify that the request is successful.
- 4 (Optional) View data center details.

```
curl GET "https://LCM-Hostname/lcm/api/v1/view/datacenter?datacenterId={datacenterId}"
```

Example: Create a Data Center Response

When the data center is successfully completed, vRealize Suite Lifecycle Manager returns a response similar to the following:

```
{
  "id": "$DataCenterID",
  "type": null,
  "state": null,
  "status": "SUCCESS",
  "isRetriable": null,
  "retryParameters": null
}
```

Add a vCenter

Add a vCenter to a Data Center before using that vCenter to create a private cloud environment.

Prerequisites

- Verify that you have a valid authentication token that matches your login credentials.
- Verify that you have added a data center and the request has completed successfully. See [Create a Data Center](#).
- Ensure that you have the vCenter fully qualified domain name, user name, and password.

Procedure

- 1 Log in to vRealize Suite Lifecycle Manager.

```
curl POST "https://LCM-Hostname/lcm/api/v1/login" \
```

- 2 Submit a request to add a vCenter to vRealize Suite Lifecycle Manager with parameters inline.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/action/add/vc" \
  -H "accept: application/json" \
  -H "x-xenon-auth-token: $token" \
  -H "content-type: application/json" \
  -d '{
    "datacenterName": "$DataCenterName",
    "vCenterName": "&vCenterFQDN",
    "userName": "admin@localhost",
    "password": "$adminPassword!",
    "type": 3}'
```

If the vCenter is successfully added, vRealize Suite Lifecycle Manager returns the following response:

```
200 Operation successful
```

Update a vCenter Server

If the password is changed for a vCenter Server that exists in a vRealize Suite Lifecycle Manager data center, you must update the password in vRealize Suite Lifecycle Manager and trigger data collection.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Update the password for an existing vCenter Server. The request also triggers vCenter Server data collection.

```
curl -X PATCH "https://LCM-Hostname/lcm/api/v1/action/add/vc" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{"datacenterName": "<datacenter in LCM>",
    "vCenterName": "<vcenterhost>",
    "userName": "<vCenter_username>",
    "password": "<vCenter_Password>",
    "type": 3}'
```

- 2 Examine the response.

The output of the request includes request ID of type VC_DATA_COLLECTION that you can use to check the progress of the data collection.

- id = Request ID of VC_DATA_COLLECTION request
- type = VC_DATA_COLLECTION
- status = Status of the request

View Data Center Details

To view details about all vCenter servers associated with a data center, you get the data center ID then request the details for that ID.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Request the name and ID of the data center in vRealize Suite Lifecycle Manager.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/datacenter" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

- 2 Examine the response for the request.

The output of the request includes:

- id = Data Center ID
- name = Data Center Name

- 3 Use the Data Center ID to request details about the data center.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/datacenter?datacenterId=<Datacenter-ID>" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

- 4 Examine the response to view information about all vCenter servers for the specified data center.

The output of the request includes:

- dataCenterName = Data Center Name
- vCenters = Details for all vCenter servers in the data center.

Getting Version Numbers

You can use the API to get version numbers for an instance of vRealize Suite Lifecycle Manager and its API. When making API calls, the versions of the vRealize Suite Lifecycle Manager the installed instance and the API must match.

This chapter includes the following topics:

- [View the vRealize Suite Lifecycle Manager Version Number](#)
- [View the vRealize Suite Lifecycle Manager API Version](#)

View the vRealize Suite Lifecycle Manager Version Number

POST /lcm/api/v1/lcmversion returns the vRealize Suite Lifecycle Manager version number and build number.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example requests the version number and build number for this instance of vRealize Suite Lifecycle Manager.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/lcmversion" \  
  -H "accept: application/json" \  
  -H "x-xenon-auth-token: $token" \  
  -H "content-type: application/json" \  
  -H "x-xenon-auth-token: $token"
```

JSON Output

The following JSON output is returned with the vRealize Suite Lifecycle Manager version number and build number when the operation completes successfully.

```
200 Operation successful
```

View the vRealize Suite Lifecycle Manager API Version

View the vRealize Suite Lifecycle Manager API version.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Request the API version.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/version" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

- 2 Examine the response for the request.

The API version is value of the property name version in the response.

```
{  
  "version": "1.0.0"  
}
```

Configuring vRealize Suite Lifecycle Manager Settings

5

You can modify settings for vRealize Suite Lifecycle Manager, such as VMware Identity Manager and MyVMware settings.

- [Manage vRealize Suite Lifecycle Manager Common Configuration Settings](#)

POST /lcm/api/v1/settings modifies vRealize Suite Lifecycle Manager common configuration settings.

- [Add an Active Directory Server](#)

POST /lcm/api/v1/settings/add/ad adds an active directory server to vRealize Suite Lifecycle Manager.

- [Remove an Active Directory Server](#)

DELETE /lcm/api/v1/settings/add/ad removes an active directory server from vRealize Suite Lifecycle Manager.

- [Add a VMware Identity Manager](#)

/lcm/api/v1/settings/add/vidm adds a VMware Identity Manager (vIDM) to vRealize Suite Lifecycle Manager.

- [Update VMware Identity Manager Password Settings](#)

If the VMware Identity Manager (vIDM) admin password is changed outside of vRealize Suite Lifecycle Manager, you can use a PATCH request to update the password in vRealize Suite Lifecycle Manager.

- [Add a MyVMware Account](#)

POST /lcm/api/v1/settings/add/myvmware adds your MyVMware account to vRealize Suite Lifecycle Manager, which enables vRealize Suite Lifecycle Manager to download product OVAs through MyVMware.

- [Update My VMware Credentials](#)

If the MyVMware credentials registered in vRealize Suite Lifecycle Manager change, you can use a PATCH request to update the password in vRealize Suite Lifecycle Manager.

Manage vRealize Suite Lifecycle Manager Common Configuration Settings

POST `/lcm/api/v1/settings` modifies vRealize Suite Lifecycle Manager common configuration settings.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example sets new passwords for admin, root, and SSH user, enables SSH and telemetry, and sets the vRealize Suite Lifecycle Manager host name.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/settings" \
  -H "accept: application/json" \
  -H "x-xenon-auth-token: $token" \
  -H "content-type: application/json" \
  -d '{
    "adminPassword": "VMware1!",
    "rootPassword": "VMware1!",
    "sshuserPassword": "VMware1!",
    "sshEnabled": true,
    "telemetryEnabled": true
  }'
```

JSON Output

The following JSON output is returned when the operation completes successfully.

```
200 Operation successful
```

Add an Active Directory Server

POST `/lcm/api/v1/settings/add/ad` adds an active directory server to vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.

- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example adds an active directory server to vRealize Suite Lifecycle Manager.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/settings/add/ad" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{
    "adName": "domain.local",
    "baseDN": "OU=Admins,DC=sqa,DC=local",
    "bindDN": "CN=testuser,OU=Admins,DC=domain,DC=local",
    "groupDN": "DC=domain,DC=local",
    "bindPassword": "vmware",
    "uberAdmin": "testuser@domain.local"
}'
```

JSON Output

The following JSON output is returned when the operation completes successfully.

```
200 Operation successful
```

Remove an Active Directory Server

DELETE /lcm/api/v1/settings/add/ad removes an active directory server from vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that VMware Identity Manager is not registered with vRealize Suite Lifecycle Manager.

curl Command

The following example removes an active directory server from vRealize Suite Lifecycle Manager.

```
curl -X DELETE "https://LCM-Hostname/lcm/api/v1/settings/add/ad" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
```

JSON Output

The following JSON output is returned when the operation completes successfully.

```
200 Operation successful
```

Add a VMware Identity Manager

`/lcm/api/v1/settings/add/vidm` adds a VMware Identity Manager (vIDM) to vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that you have an existing VMware Identity Manager version 2.9.2.
- Verify that you have added an active directory server to vRealize Suite Lifecycle Manager. See [Add an Active Directory Server](#).

curl Command

The following example adds and enables a vIDM for all users.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/settings/add/vidm" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{
  "hostname": "$vIDMhostFQDN",
  "adminUserName": "$adminUserName",
  "adminPassword": "$adminPassword",
  "vidmCloudAdminGroup": "ALL_USERS",
  "enabled": true}'
```

JSON Output

The following JSON output is returned when the operation completes successfully.

```
200 Operation successful
```

Update VMware Identity Manager Password Settings

If the VMware Identity Manager (vIDM) admin password is changed outside of vRealize Suite Lifecycle Manager, you can use a PATCH request to update the password in vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Submit a request with the new admin password to update the vIDM admin password in vRealize Suite Lifecycle Manager.

```
curl -X PATCH "https://LCM-Hostname/lcm/api/v1/settings/add/vidm/vidmsettings" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{"adminPassword": "<new_admin_password>"}
```

Where *new_admin_password* is the updated admin password.

- 2 Examine the response to see the ID and the status of the request.
 - id = ID of the PATCH operation
 - status=Status of the request

Add a MyVMware Account

POST `/lcm/api/v1/settings/add/myvmware` adds your MyVMware account to vRealize Suite Lifecycle Manager, which enables vRealize Suite Lifecycle Manager to download product OVAs through MyVMware.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that you have valid MyVMware credentials.

curl Command

The following example adds a MyVMware account with the specified user name and password.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/settings/add/myvmware" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{
  "username": "$myVMwareUserName",
  "password": "$myVMwarePassword"}'
```

JSON Output

The following JSON output is returned when the operation completes successfully.

```
200 Operation successful
```

Update My VMware Credentials

If the MyVMware credentials registered in vRealize Suite Lifecycle Manager change, you can use a PATCH request to update the password in vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Submit a request to get the MyVMware ID.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/settings/add/myvmware" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{"username": "myVMW_username", "password": "myVMW_password"}'
```

Where *myVMW_username* and *my_VMW_password* are the MyVMware user details.

- 2 Examine the response for the *myVMware_ID*.

- 3 Using the *myVMware_ID*, submit a request to update the MyVMware credential with the new MyVMware password.

```
curl -X PATCH "https://LCM-Hostname/lcm/api/v1/settings/add/myvmware/<myVMware_ID>" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{"username": "myVMW_username", "password": "new_myVMW_password" }'
```

- 4 Examine the response to see the ID and the status of the request.
 - id = ID of the PATCH operation
 - status=Status of the request

Working with Environments

You can use the REST API to create and monitor vRealize Suite environments.

Environments allow you to install, upgrade, and manage vRealize Suite products as unit. You can build your environment using either product-based or solution-based installation, or you can import an existing vRealize Suite environment to manage it with vRealize Suite Lifecycle Manager.

- [Pre-validating Before Creating a vRealize Suite Environment](#)

You can use the vRealize Suite Lifecycle Manager API to pre-validate deployment inputs, infrastructure, and products before creating or importing a vRealize Suite environment.

- [Create an Environment](#)

POST /lcm/api/v1/action/create/environment creates a vRealize Suite environment.

- [View an Environment and Details](#)

You can list the details of an environment and its nodes in a vRealize Suite environment managed by vRealize Suite Lifecycle Manager.

- [Add a Product to an Existing Environment](#)

POST /lcm/api/v1/action/add/product?id={environmentId} adds a vRealize Suite product to an existing environment.

- [Add a Node](#)

POST /lcm/api/v1/action/addNode scales out the environment by adding a single node to a product in the environment. To add multiple nodes, make a separate request for each node.

- [Viewing Products and Product Details](#)

You can use the vRealize Suite Lifecycle Manager API to view products and the details of products installed in your vRealize Suite environment.

- [Pre-Validate and Upgrade Products in Your vRealize Suite Environment](#)

You can use the vRealize Suite Lifecycle Manager API to upgrade products installed in your vRealize Suite environment. Before upgrading, you pre-validate the upgrade request. If the pre-validation check indicates no issues with the upgrade, it is safe to proceed with the upgrade.

- [Collect Environment Logs](#)

POST /lcm/api/v1/collectlogs/environment collects log files for all vRealize Suite Lifecycle Manager environments that you can use for troubleshooting.

■ Configuration Drift

Configuration drift shows the changes in product configuration over time and allows you to revert a product to an earlier configuration state.

Pre-validating Before Creating a vRealize Suite Environment

You can use the vRealize Suite Lifecycle Manager API to pre-validate deployment inputs, infrastructure, and products before creating or importing a vRealize Suite environment.

Pre-validate Environment Request

Before creating a vRealize Suite environment, you pre-validate the create environment request. You use the output from the pre-validation request to obtain a pre-validation report.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Run a request to pre-validate your environment.

The following input example includes a snippet for the vRealize Log Insight product and amended code sections for other products.

```
curl -X POST "https://LCM-HostName/lcm/api/v1/action/prevalidate/create/environment" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d "{
  "requestId": "null",
  "environmentId": "",
  "infrastructure": {
    "sourceLink": "",
    "properties": {
      "bindPassword": "",
      "dataCenterName": "Datacenter-Name",
      "vcHostname": "vCenter Server Host name",
      "environmentId": "",
      "masterVidmAdminUserName": "",
      "netmask": "Network-Netmask",
      "environmentName": "vRealize Suite Large Advance",
      "clusterName": "VC-Datacenter-Name#VC-Cluster-Name",
      "enableTelemetry": "true",
      "dnsServers": "Network-DNS-Server-IPs",
      "diskFormat": "Thin",
```



```

        "baseDN": "",
        "vcPassword": "vCenter Server Password",
        "defaultPassword": "vRealize Suite Products password",
        "adminEmail": "cloud admin email address",
        "adName": "",
        "certificateChain": "-----BEGIN CERTIFICATE-----\nMIIG3zC\\...\\n-----END CERTIFICATE-----",
        "masterKeyPassphrase": "",
        "datastoreName": "VC Datastore Name",
        "masterVidmAdminPassword": "",
        "masterVidmEnabled": "",
        "uberAdmin": "",
        "license": "",
        "privateKey": "-----BEGIN RSA PRIVATE KEY-----\nMIIEpQIBAAKCA\\...\\n-----END RSA PRIVATE
KEY-----",
        "bindDN": "",
        "vmNetwork": "Network-Portgroup",
        "masterPrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nMIIEpQIBAAKCA\\...\\n-----END RSA PRIVATE
KEY-----",
        "masterVidmHostName": "",
        "groupDN": "",
        "masterVidmCloudAdminGroup": "",
        "vcUsername": "vCenter Server Username",
        "domain": "",
        "acceptEULA": "true",
        "keyPassphrase": "",
        "gateway": "Network-Gateway",
        "searchpath": "",
        "masterCertificateChain": "-----BEGIN CERTIFICATE-----\nMIIG3zC\\...\\n-----END CERTIFICATE-----"
    }
},
"encoded": false,
"products": [
    {
        "id": "vrli",
        "version": "4.5.0",
        "clusterVIP": [],
        "properties": {
            "vrliClusterVips": "vRLI-Cluster-IP#vRLI-Cluster-IP-Hostname"
        },
    },
    "nodes": [
        {
            "sourceLink": "",
            "type": "vrli-master",
            "properties": {
                "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
                "vrliLicenseKey": "",
                "installerLocation": "/data/productlinks/vrli/4.5.0/install/vrli.ova",
                "dns": "Network-DNS-Server-IPs",
                "ipAddress": "vRLI-IP",
                "vCenterHost": "vCenter Server Host name",
                "storage": "VC Datastore Name",
                "userName": "vCenter Server Username",
                "masterVidmAdminPassword": "",
                "uberAdmin": "",
                "network": "Network-Portgroup",
            },
        },
    ],

```

```

        "masterVidmEnabled": "",
        "vrliAdminEmail": "",
        "hostname": "vRLI-Hostname",
        "password": "vCenter Server Password",
        "masterVidmHostName": "",
        "masterVidmAdminUserName": "",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vrli-master",
        "diskFormat": "Thin",
        "vrliClusterVips": "vRLI-Cluster-IP#vRLI-Cluster-IP-Hostname",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
},
{
    "sourceLink": "",
    "type": "vrli-worker",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "installerLocation": "/data/productlinks/vrli/4.5.0/install/vrli.ova",
        "ipAddress": "",
        "dns": "Network-DNS-Server-IPs",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vrli-worker-01",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "vrli-worker",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "installerLocation": "/data/productlinks/vrli/4.5.0/install/vrli.ova",
        "ipAddress": "",
        "dns": "Network-DNS-Server-IPs",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vrli-worker-02",
        "diskFormat": "Thin",

```

```

        "gateway": "Network-Gateway",
        "searchpath": ""
    }
}
],
{
    "id": "vra",
    "version": "7.3.0",
    "clusterVIP": [
        {
            "type": "vra",
            "hostname": "vRA-LoadBalancer-Hostname",
            "ipAddress": "vRA-LoadBalancer-IP"
        },
        {
            "type": "iaas-web",
            "hostname": "IaaS-Web-LoadBalancer-Hostname",
            "ipAddress": "IaaS-Web-LoadBalancer-IP"
        },
        {
            "type": "iaas-manager",
            "hostname": "IaaS-Manager-Service-LoadBalancer-Hostname",
            "ipAddress": "IaaS-Manager-Service-LoadBalancer-IP"
        }
    ],
    "properties": {
        "windowsPassword": "",
        "windowsUsername": ""
    },
    "nodes": [
        ...
    ]
},
{
    "id": "vrbc",
    "version": "7.3.0",
    "clusterVIP": [],
    "properties": {
        "currency": "USD - US Dollar"
    },
    "nodes": [
        ...
    ]
},
{
    "id": "vrops",
    "version": "6.6.1",
    "clusterVIP": [],
    "properties": {
        "ntpServerIP": "NTP Server IP"
    },
    "nodes": [
        ...
    ]
}

```

```
    ]
  }
]
}"
```

- 2 Examine the response to track the request.

The output includes:

- id = Environment Pre-validate Request ID
- status = Environment Pre-validate Status

- 3 (Optional) Monitor the status of the request with the Environment Pre-validate Request ID.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/request/status/<Environment-Prevalidate-Request-ID>" -
H "accept: application/json" -H "x-xenon-auth-token: $token"
```

What to do next

Use the Environment Pre-validate Request ID to retrieve a pre-validation report of your environment. See [Retrieve Pre-validation Report for Environment Creation](#).

Retrieve Pre-validation Report for Environment Creation

After completing a pre-validation request, you retrieve the pre-validation report to check for errors.

A pre-validation report provides:

- Input data validation alerts such as missing input or incorrect input formats
- Infrastructure validation alerts such as vCenter Server input checks, a reused VM name, already used IP addresses, and other checks related to the vCenter Server
- Product validation alerts

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that you have the Environment Pre-validate Request ID from [Pre-validate Environment Request](#).

Procedure

- 1 Retrieve the pre-validation report .

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/prevalidation/report?requestId=<Request-ID>" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token"
```

Request-ID is the Environment Pre-validate Request ID.

2 Check for the key word FAILED in the request response.

The following snippet is from an example response.

```
{
  "rootValidations": [
    {
      "id": null,
      "checkName": "Data validation",
      "status": "FAILED",
      "recommendations": null,
      "resultDescription": null,
      "elementType": null,
      "childElements": [
        {
          "id": null,
          "checkName": "Input schema",
          "status": "PASSED",
          "recommendations": null,
          "resultDescription": null,
          "elementType": null,
          "childElements": [
            {
              "id": null,
              "checkName": "Input schema validation",
              "status": "PASSED",
              "recommendations": null,
              "resultDescription": "Success. all the schema validations passed",
              "elementType": null,
              "childElements": null,
              "checktype": "ERROR"
            }
          ],
          "checktype": null
        },
        {
          "id": null,
          "checkName": "Input format",
          "status": "FAILED",
          "recommendations": null,
          "resultDescription": null,
          "elementType": null,
          "childElements": [
            {
              "id": null,
              "checkName": "Infrastructure property check",
              "status": "FAILED",
              "recommendations": [
                "Make sure that provided field values are valid"
              ],
              "resultDescription": "<searchpath> is undefined or empty",
              "elementType": null,
              "childElements": null,
              "checktype": "ERROR"
            }
          ],
          "checktype": null
        }
      ]
    }
  ],
  "checktype": null
}
```

```

{
  "id": null,
  "checkName": "Infrastructure property check",
  "status": "FAILED",
  "recommendations": [
    "Make sure that provided field values are valid"
  ],
  "resultDescription": "<domain> is undefined or empty",
  "elementType": null,
  "childElements": null,
  "checktype": "ERROR"
},
...
{
  "id": null,
  "checkName": "Infrastructure property validation",
  "status": "FAILED",
  "recommendations": [
    "Make sure that provided field values are valid"
  ],
  "resultDescription": "<netmask> is invalid\"
  \"elementType\": null,
  \"childElements\": null,
  \"checktype\": \"ERROR\"
},
...
{
  \"id\": null,
  \"checkName\": \"Product id and version check\",
  \"status\": \"FAILED\",
  \"recommendations\": [
    \"Make sure that provided field values are valid\"
  ],
  \"resultDescription\": \"Product is unrecognized with product id vrli and product
version 4.5.0\",
  \"elementType\": null,
  \"childElements\": null,
  \"checktype\": \"ERROR\"
},
...
{
  \"id\": null
  \"checkName\": \"Product property validation for vrops\",
  \"status\": \"FAILED\",
  \"recommendations\": [
    \"Make sure that provided field values are valid\"
  ],
  \"resultDescription\": \"<timeSyncMode> is undefined or empty\",
  \"elementType\": null,
  \"childElements\": null,
  \"checktype\": \"ERROR\"
},
...
{
  \"id\": null,

```

```

    "checkName": "Node property check for vrops:S1-master",
    "status": "FAILED",
    "recommendations": [
        "Make sure that provided field values are valid"
    ],
    "resultDescription": "<netmask> is invalid",
    "elementType": null
    "childElements": null,
    "checktype": "ERROR"
},
{
    "id": null,
    "checkName": "Node property check for vrops:S1-master",
    "status": "FAILED",
    "recommendations": [
        "Make sure that provided field values are valid"
    ],
    "resultDescription": "<gateway> is invalid",
    "elementType": null
    "childElements": null,
    "checktype": "ERROR"
},
...
{
    "id": null,
    "checkName": "Node property check for vrops:S1-replica",
    "status": "FAILED",
    "recommendations": [
        "Make sure that provided field values are valid"
    ],
    "resultDescription": "<ipAddress> is required",
    "elementType": null
    "childElements": null,
    "checktype": "ERROR"
},
...
{
    "id": null,
    "checkName": "Node property check for vrops:S1-data-01",
    "status": "FAILED",
    "recommendations": [
        "Make sure that provided field values are valid"
    ],
    "resultDescription": "<ipAddress> is required",
    "elementType": null
    "childElements": null,
    "checktype": "ERROR"
},
{
    "id": null,
    "checkName": "Node property check for vrops:S1-data-02",
    "status": "FAILED",
    "recommendations": [
        "Make sure that provided field values are valid"
    ],
    "resultDescription": "<ipAddress> is required",
    "elementType": null
    "childElements": null,
    "checktype": "ERROR"
},
{
    "id": null,
    "checkName": "Node property check for vrops:S1-data-03",
    "status": "FAILED",
    "recommendations": [
        "Make sure that provided field values are valid"
    ],
    "resultDescription": "<ipAddress> is required",
    "elementType": null
    "childElements": null,
    "checktype": "ERROR"
},
...
}

```

```

        "resultDescription": "<hostname> is invalid",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    ...
    {
        "id": null,
        "checkName": "Node property check for vrops:S1-data-03",
        "status": "FAILED",
        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<ipAddress> is required",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    {
        "id": null,
        "checkName": "Node property check for vrops:S1-data-04",
        "status": "FAILED",
        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<hostname> is invalid",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    ...
    {
        "id": null,
        "checkName": "Node property check for vrops:S1-data-05",
        "status": "FAILED",
        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<ipAddress> is required",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    {
        "id": null,
        "checkName": "Node property check for vrops:S1-data-06",
        "status": "FAILED",
        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<hostname> is invalid",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    },

```



```

...
{
  "id": null,
  "checkName": "Node property check for vrops:S1-data-07",
  "status": "FAILED",
  "recommendations": [
    "Make sure that provided field values are valid"
  ],
  "resultDescription": "<ipAddress> is required",
  "elementType": null
  "childElements": null,
  "checktype": "ERROR"
},
{
  "id": null,
  "checkName": "Node property check for vrops:S1-data-08",
  "status": "FAILED",
  "recommendations": [
    "Make sure that provided field values are valid"
  ],
  "resultDescription": "<hostname> is invalid",
  "elementType": null
  "childElements": null,
  "checktype": "ERROR"
},
...
{
  "id": null,
  "checkName": "Node property check for vrops:S1-data-09",
  "status": "FAILED",
  "recommendations": [
    "Make sure that provided field values are valid"
  ],
  "resultDescription": "<ipAddress> is required",
  "elementType": null
  "childElements": null,
  "checktype": "ERROR"
},
{
  "id": null,
  "checkName": "Node property check for vrops:S1-data-10",
  "status": "FAILED",
  "recommendations": [
    "Make sure that provided field values are valid"
  ],
  "resultDescription": "<hostname> is invalid",
  "elementType": null
  "childElements": null,
  "checktype": "ERROR"
},
...
{
  "id": null,
  "checkName": "Node property check for vrops:S1-data-11",
  "status": "FAILED",

```

```

        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<ipAddress> is required",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    {
        "id": null,
        "checkName": "Node property check for vrops:S1-data-12",
        "status": "FAILED",
        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<hostname> is invalid",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    ...
    {
        "id": null,
        "checkName": "Node property check for vrops:S1-remotecollector-01",
        "status": "FAILED",
        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<deployOption> is required",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    {
        "id": null,
        "checkName": "Node property check for vrops:S1-vrops-remotecollector-02",
        "status": "FAILED",
        "recommendations": [
            "Make sure that provided field values are valid"
        ],
        "resultDescription": "<hostname> is invalid",
        "elementType": null
        "childElements": null,
        "checktype": "ERROR"
    },
    ...
    ],
    "checktype": null
}
],
"checktype": null
}
],
"manualValidations": null,
"overallStatus": false,

```

```

    "requestState": null,
    "lastUpdateTimeMillis": 0,
    "startTimeMillis": 0,
    "requestId": null
  }

```

What to do next

If the report indicates errors in the environment request, correct the JSON input spec with the correct parameter values and run the pre-validation request again. See [Pre-validate Environment Request](#).

Create an Environment

POST /lcm/api/v1/action/create/environment creates a vRealize Suite environment.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Configure OVA settings for the vRealize Suite to install.
- Verify that you have added a data center and the request has completed successfully. See [Create a Data Center](#).
- Verify that you have added a vCenter to the data center and the request has completed successfully. See [Add a vCenter](#).

curl Command

The following example creates a vRealize Suite environment with all available products and nodes.

```

curl -X POST "https://LCM-Hostname/lcm/api/v1/action/create/environment" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{
  "requestId": "null",
  "environmentId": "",
  "infrastructure": {
    "sourceLink": "",
    "properties": {
      "bindPassword": "",
      "dataCenterName": "Datacenter-Name",
      "vcHostname": "vCenter Server Host name",
      "environmentId": "",
      "masterVidmAdminUserName": "",
      "netmask": "Network-Netmask",
      "environmentName": "vRealize Suite Large Advance",

```

```

        "clusterName": "VC-Datacenter-Name#VC-Cluster-Name",
        "enableTelemetry": "true",
        "dnsServers": "Network-DNS-Server-IPs",
        "diskFormat": "Thin",
        "baseDN": "",
        "vcPassword": "vCenter Server Password",
        "defaultPassword": "vRealize Suite Products password",
        "adminEmail": "cloud admin email address",
        "adName": "",
        "certificateChain": "-----BEGIN
CERTIFICATE-----\nMIIG3zC\nCZImiZPyLQG8GRYFbG9jYwWxEzARBgoJkiaJk/IsZAEZ\nFnNxYS1\n-----END
CERTIFICATE-----",
        "masterKeyPassphrase": "",
        "datastoreName": "VC Datastore Name",
        "masterVidmAdminPassword": "",
        "masterVidmEnabled": "",
        "uberAdmin": "",
        "license": "",
        "privateKey": "-----BEGIN RSA PRIVATE
KEY-----\nMIIEpQIBAAKCA\n4Hf/7hp59x\nGGlHmsQAidubXQdBMGgxiBGZz/6cEoUs3+EWTIg0pLt3a78yCkK9wLZ/U=\n-----
END RSA PRIVATE KEY-----",
        "bindDN": "",
        "vmNetwork": "Network-Portgroup",
        "masterPrivateKey": "-----BEGIN RSA PRIVATE
KEY-----\nMIIEpQIBAAKCA\n4Hf/7hp59x\nGGlHmsQAidubXQdBMGgxiBGZz/6cEoUs3+EWTIg0pLt3a78yCkK9wLZ/U=\n-----
END RSA PRIVATE KEY-----",
        "masterVidmHostName": "",
        "groupDN": "",
        "masterVidmCloudAdminGroup": "",
        "vcUsername": "vCenter Server Username",
        "domain": "",
        "acceptEULA": "true",
        "keyPassphrase": "",
        "gateway": "Network-Gateway",
        "searchpath": "",
        "masterCertificateChain": "-----BEGIN
CERTIFICATE-----\nMIIG3zC\nCZImiZPyLQG8GRYFbG9jYwWxEzARBgoJkiaJk/IsZAEZ\nFnNxYS1\n-----END
CERTIFICATE-----"
    }
},
    "encoded": false,
    "products": [
        {
            "id": "vrli",
            "version": "4.5.0",
            "clusterVIP": [],
            "properties": {
                "vrliClusterVips": "vrli-Cluster-IP#vrli-Cluster-IP-Hostname"
            }
        },
        {
            "sourceLink": "",
            "type": "vrli-master",
            "properties": {
                "cluster": "VC-Datacenter-Name#VC-Cluster-Name",

```

```

        "vrliLicenseKey": "",
        "installerLocation": "/data/productlinks/vrli/4.5.0/install/vrli.ova",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "vRLI-IP",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "masterVidmAdminPassword": "",
        "uberAdmin": "",
        "network": "Network-Portgroup",
        "masterVidmEnabled": "",
        "vrliAdminEmail": "",
        "hostname": "vRLI-Hostname",
        "password": "vCenter Server Password",
        "masterVidmHostName": "",
        "masterVidmAdminUserName": "",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vrli-master",
        "diskFormat": "Thin",
        "vrliClusterVips": "vRLI-Cluster-IP#vRLI-Cluster-IP-Hostname",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
},
{
    "sourceLink": "",
    "type": "vrli-worker",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "installerLocation": "/data/productlinks/vrli/4.5.0/install/vrli.ova",
        "ipAddress": "",
        "dns": "Network-DNS-Server-IPs",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vrli-worker-01",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "vrli-worker",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "installerLocation": "/data/productlinks/vrli/4.5.0/install/vrli.ova",
        "ipAddress": "",
        "dns": "Network-DNS-Server-IPs",

```

```

        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vrli-worker-02",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
}
],
},
{
    "id": "vra",
    "version": "7.3.0",
    "clusterVIP": [
        {
            "type": "vra",
            "hostname": "vRA-LoadBalancer-Hostname",
            "ipAddress": "vRA-LoadBalancer-IP"
        },
        {
            "type": "iaas-web",
            "hostname": "IaaS-Web-LoadBalancer-Hostname",
            "ipAddress": "IaaS-Web-LoadBalancer-IP"
        },
        {
            "type": "iaas-manager",
            "hostname": "IaaS-Manager-Service-LoadBalancer-Hostname",
            "ipAddress": "IaaS-Manager-Service-LoadBalancer-IP"
        }
    ],
    "properties": {
        "windowsPassword": "",
        "windowsUsername": ""
    },
    "nodes": [
        {
            "sourceLink": "",
            "type": "vra-server-primary",
            "properties": {
                "vidmVraDisabledAdvanced": "",
                "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
                "certificateChain": "",
                "installerLocation": "/data/productlinks/vra/7.3.0/install/vra.ova",
                "dns": "Network-DNS-Server-IPs",
                "ipAddress": "vRA-IP",
                "vCenterHost": "vCenter Server Host name",
                "storage": "VC Datastore Name",
                "userName": "vCenter Server Username",
                "network": "Network-Portgroup",

```

```

        "vidmPassword": "",
        "privateKey": "",
        "licenseKey": "",
        "hostname": "vRA-Hostname",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vra-server-primary",
        "keyPassphrase": "",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "vra-server-secondary",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "installerLocation": "/data/productlinks/vra/7.3.0/install/vra.ova",
        "certificateChain": "",
        "ipAddress": "",
        "dns": "Network-DNS-Server-IPs",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "vidmPassword": "",
        "privateKey": "",
        "password": "vCenter Server Password",
        "hostname": "",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-vra-server-secondary",
        "keyPassphrase": "",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
},
{
    "sourceLink": "",
    "type": "db",
    "properties": {
        "useWindowsAuthentication": "true",
        "useExistingDatabase": "false",
        "password": "vCenter Server Password",
        "hostname": "",
        "databaseName": "iaas",
        "name": "db",
        "databaseUserName": "",
        "databasePassword": "",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username"
    }
},

```

```

{
  "sourceLink": "",
  "type": "iaas-web",
  "properties": {
    "webPassword": "",
    "vidmPassword": "",
    "hostname": "",
    "password": "vCenter Server Password",
    "ipAddress": "",
    "name": "iaas-web-01",
    "webUserName": "",
    "vCenterHost": "vCenter Server Host name",
    "userName": "vCenter Server Username"
  }
},
{
  "sourceLink": "",
  "type": "iaas-web",
  "properties": {
    "vidmPassword": "",
    "webPassword": "",
    "password": "vCenter Server Password",
    "hostname": "",
    "ipAddress": "",
    "name": "iaas-web-02",
    "vCenterHost": "vCenter Server Host name",
    "webUserName": "",
    "userName": "vCenter Server Username"
  }
},
{
  "sourceLink": "",
  "type": "iaas-manager-active",
  "properties": {
    "password": "vCenter Server Password",
    "hostname": "",
    "msUserName": "",
    "msPassword": "",
    "name": "iaas-manager-active",
    "ipAddress": "",
    "vCenterHost": "vCenter Server Host name",
    "userName": "vCenter Server Username"
  }
},
{
  "sourceLink": "",
  "type": "iaas-manager-passive",
  "properties": {
    "password": "vCenter Server Password",
    "hostname": "",
    "msUserName": "",
    "msPassword": "",
    "name": "iaas-manager-passive",
    "ipAddress": "",
    "vCenterHost": "vCenter Server Host name",

```



```

        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-orchestrator",
    "properties": {
        "demPassword": "",
        "hostname": "",
        "password": "vCenter Server Password",
        "ipAddress": "",
        "name": "Demorchestrator-02",
        "demUserName": "",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-orchestrator",
    "properties": {
        "demPassword": "",
        "password": "vCenter Server Password",
        "hostname": "",
        "name": "Demorchestrator-01",
        "ipAddress": "",
        "demUserName": "",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-worker",
    "properties": {
        "demPassword": "",
        "password": "vCenter Server Password",
        "hostname": "",
        "name": "Demworker-01-2",
        "ipAddress": "",
        "demUserName": "",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-worker",
    "properties": {
        "demPassword": "",
        "password": "vCenter Server Password",
        "hostname": "",
        "name": "Demworker-01-3",
        "ipAddress": "",
        "demUserName": "",

```

```

        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-worker",
    "properties": {
        "demPassword": "",
        "hostname": "",
        "password": "vCenter Server Password",
        "ipAddress": "",
        "name": "Demworker-02-2",
        "demUserName": "",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-worker",
    "properties": {
        "demPassword": "",
        "hostname": "",
        "password": "vCenter Server Password",
        "name": "Demworker-02-3",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "demUserName": "",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-worker",
    "properties": {
        "demPassword": "",
        "password": "vCenter Server Password",
        "hostname": "",
        "name": "Demworker-01-1",
        "ipAddress": "",
        "demUserName": "",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "iaas-dem-worker",
    "properties": {
        "demPassword": "",
        "hostname": "",
        "password": "vCenter Server Password",
        "name": "Demworker-02-1",
        "ipAddress": "",

```

```

        "vCenterHost": "vCenter Server Host name",
        "demUserName": "",
        "userName": "vCenter Server Username"
    }
},
{
    "sourceLink": "",
    "type": "proxy-agent-vsphere",
    "properties": {
        "password": "vCenter Server Password",
        "hostname": "",
        "agentUserName": "",
        "name": "VCAgent-1",
        "ipAddress": "",
        "agentName": "VCAgent-1",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username",
        "vsphereEndpointName": "VCEndpoint-1",
        "agentPassword": ""
    }
},
{
    "sourceLink": "",
    "type": "proxy-agent-vsphere",
    "properties": {
        "hostname": "",
        "password": "vCenter Server Password",
        "agentUserName": "",
        "ipAddress": "",
        "name": "VCAgent-2",
        "agentName": "VCAgent-2",
        "vCenterHost": "vCenter Server Host name",
        "userName": "vCenter Server Username",
        "vsphereEndpointName": "VCEndpoint-2",
        "agentPassword": ""
    }
}
]
},
{
    "id": "vrbc",
    "version": "7.3.0",
    "clusterVIP": [],
    "properties": {
        "currency": "USD - US Dollar"
    },
    "nodes": [
        {
            "sourceLink": "",
            "type": "vrbc-server",
            "properties": {
                "ssoPassword": "",
                "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
                "vrbcTelemetryEnabled": "true",
                "installerLocation": "/data/productlinks/vrbc/7.3.0/install/vrbc.ova",

```

```

        "tenantPassword": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "password": "vCenter Server Password",
        "hostname": "vRB-Hostname",
        "masterVidmAdminUserName": "",
        "netmask": "Network-Netmask",
        "diskFormat": "Thin",
        "vrbCurrency": "USD - US Dollar",
        "certificateChain": "",
        "ipAddress": "vRB-IP",
        "dns": "Network-DNS-Server-IPs",
        "vrbLicenseKey": "",
        "userName": "vCenter Server Username",
        "masterVidmAdminPassword": "",
        "uberAdmin": "",
        "masterVidmEnabled": "",
        "isTelemetryEnable": "true",
        "privateKey": "",
        "masterVidmHostName": "",
        "domain": "",
        "name": "S1-vrb-server",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
},
{
    "sourceLink": "",
    "type": "vrb-collector",
    "properties": {
        "ssoPassword": "",
        "cluster": "LCM-DC#LCM-Cluster-02",
        "vrbTelemetryEnabled": "true",
        "installerLocation": "/data/productlinks/vrbc/7.3.0/install/vrbc.ova",
        "tenantPassword": "",
        "vCenterHost": "lcm-vc.",
        "storage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "masterVidmAdminUserName": "",
        "netmask": "Network-Netmask",
        "diskFormat": "Thin",
        "vrbCurrency": "USD - US Dollar",
        "sshEnabled": "True",
        "ipAddress": "",
        "dns": "Network-DNS-Server-IPs",
        "vrbLicenseKey": "",
        "userName": "vCenter Server Username",
        "masterVidmAdminPassword": "",
        "isTelemetryEnable": "true",
        "masterVidmEnabled": "true",
        "uberAdmin": "",
        "masterVidmHostName": "",
    }
}

```

```

        "domain": "",
        "name": "S1-vrb-collector",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
}
],
{
    "id": "vrops",
    "version": "6.6.1",
    "clusterVIP": [],
    "properties": {
        "ntpServerIP": "NTP Server IP"
    },
    "nodes": [
        {
            "sourceLink": "",
            "type": "master",
            "properties": {
                "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
                "installerLocation": "/data/productlinks/vrops/6.6.1/install/vrops.ova",
                "vCenterHost": "vCenter Server Host name",
                "storage": "VC Datastore Name",
                "network": "Network-Portgroup",
                "password": "vCenter Server Password",
                "hostname": "vROPS-Hostname",
                "masterVidmAdminUserName": "",
                "netmask": "Network-Netmask",
                "diskFormat": "Thin",
                "ntpServer": "NTP Server IP",
                "address": "vROPS-IP",
                "certificateChain": "",
                "dns": "Network-DNS-Server-IPs",
                "ipAddress": "vROPS-IP",
                "userName": "vCenter Server Username",
                "masterVidmAdminPassword": "",
                "masterVidmEnabled": "false",
                "license": "",
                "privateKey": "",
                "masterVidmHostName": "",
                "domain": "",
                "name": "S1-master",
                "keyPassphrase": "",
                "gateway": "Network-Gateway",
                "searchpath": ""
            }
        },
        {
            "sourceLink": "",
            "type": "replica",
            "properties": {
                "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
                "address": "",
                "ipAddress": "",

```

```

        "dns": "Network-DNS-Server-IPs",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-replica",
        "diskFormat": "Thin",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
},
{
    "sourceLink": "",
    "type": "data",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "extendedStorage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-data-01",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "data",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "extendedStorage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",

```

```

        "name": "S1-data-02",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "data",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "extendedStorage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-data-03",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "data",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "extendedStorage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-data-04",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "data",

```

```

    "properties": {
      "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
      "address": "",
      "dns": "Network-DNS-Server-IPs",
      "ipAddress": "",
      "vCenterHost": "vCenter Server Host name",
      "storage": "VC Datastore Name",
      "userName": "vCenter Server Username",
      "extendedStorage": "VC Datastore Name",
      "network": "Network-Portgroup",
      "hostname": "",
      "password": "vCenter Server Password",
      "netmask": "Network-Netmask",
      "domain": "",
      "name": "S1-data-05",
      "diskFormat": "Thin",
      "gateway": "Network-Gateway",
      "searchpath": ""
    }
  },
  {
    "sourceLink": "",
    "type": "data",
    "properties": {
      "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
      "address": "",
      "dns": "Network-DNS-Server-IPs",
      "ipAddress": "",
      "vCenterHost": "vCenter Server Host name",
      "storage": "VC Datastore Name",
      "userName": "vCenter Server Username",
      "extendedStorage": "VC Datastore Name",
      "network": "Network-Portgroup",
      "hostname": "",
      "password": "vCenter Server Password",
      "netmask": "Network-Netmask",
      "domain": "",
      "name": "S1-data-06",
      "diskFormat": "Thin",
      "gateway": "Network-Gateway",
      "searchpath": ""
    }
  },
  {
    "sourceLink": "",
    "type": "data",
    "properties": {
      "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
      "address": "",
      "dns": "Network-DNS-Server-IPs",
      "ipAddress": "",
      "vCenterHost": "vCenter Server Host name",
      "storage": "VC Datastore Name",
      "userName": "vCenter Server Username",
      "extendedStorage": "VC Datastore Name",

```



```

        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-data-07",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "data",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "extendedStorage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-data-08",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "data",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "extendedStorage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-data-09",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
}

```

```

    }
  },
  {
    "sourceLink": "",
    "type": "data",
    "properties": {
      "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
      "address": "",
      "dns": "Network-DNS-Server-IPs",
      "ipAddress": "",
      "vCenterHost": "vCenter Server Host name",
      "storage": "VC Datastore Name",
      "userName": "vCenter Server Username",
      "extendedStorage": "VC Datastore Name",
      "network": "Network-Portgroup",
      "hostname": "",
      "password": "vCenter Server Password",
      "netmask": "Network-Netmask",
      "domain": "",
      "name": "S1-data-10",
      "diskFormat": "Thin",
      "gateway": "Network-Gateway",
      "searchpath": ""
    }
  },
  {
    "sourceLink": "",
    "type": "data",
    "properties": {
      "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
      "address": "",
      "dns": "Network-DNS-Server-IPs",
      "ipAddress": "",
      "vCenterHost": "vCenter Server Host name",
      "storage": "VC Datastore Name",
      "userName": "vCenter Server Username",
      "extendedStorage": "VC Datastore Name",
      "network": "Network-Portgroup",
      "hostname": "",
      "password": "vCenter Server Password",
      "netmask": "Network-Netmask",
      "domain": "",
      "name": "S1-data-11",
      "diskFormat": "Thin",
      "gateway": "Network-Gateway",
      "searchpath": ""
    }
  },
  {
    "sourceLink": "",
    "type": "data",
    "properties": {
      "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
      "address": "",
      "dns": "Network-DNS-Server-IPs",

```

```

        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "extendedStorage": "VC Datastore Name",
        "network": "Network-Portgroup",
        "hostname": "",
        "password": "vCenter Server Password",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-data-12",
        "diskFormat": "Thin",
        "gateway": "Network-Gateway",
        "searchpath": ""
    }
},
{
    "sourceLink": "",
    "type": "remotecollector",
    "properties": {
        "cluster": "VC-Datacenter-Name#VC-Cluster-Name",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "password": "vCenter Server Password",
        "hostname": "",
        "netmask": "Network-Netmask",
        "domain": "",
        "name": "S1-remotecollector-01",
        "ntpServer": "NTP Server IP",
        "diskFormat": "Thin",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
},
{
    "sourceLink": "",
    "type": "remotecollector",
    "properties": {
        "cluster": "LCM-DC#LCM-Cluster-02",
        "address": "",
        "dns": "Network-DNS-Server-IPs",
        "ipAddress": "",
        "vCenterHost": "vCenter Server Host name",
        "storage": "VC Datastore Name",
        "userName": "vCenter Server Username",
        "network": "Network-Portgroup",
        "password": "vCenter Server Password",
        "hostname": "",
        "netmask": "Network-Netmask",
        "domain": "",

```

```

        "name": "S1-vrops-remotecollector-02",
        "ntpServer": "NTP Server IP",
        "diskFormat": "Thin",
        "searchpath": "",
        "gateway": "Network-Gateway"
    }
}
]
}
]
}'

```

JSON Output

When the operation completes successfully, the JSON output includes:

- id = Create environment request ID
- status = Create environment status

View an Environment and Details

You can list the details of an environment and its nodes in a vRealize Suite environment managed by vRealize Suite Lifecycle Manager.

Use the list to verify that vRealize Suite Lifecycle Manager deployed the environment correctly and to plan for a future expansion of your environment.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use `POST /lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 List all products and nodes in your environment.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/environment" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

- 2 Examine the response.

The output of the request includes:

- id = Environment ID
- name = Environment Name

3 Using the Environment ID, get all the environment details.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/view/environment?environmentId=<Environment-ID>" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token"
```

4 Examine the response.

The output of the request includes:

- name = Environment Name
- products = List of products and component details present in the environment

Add a Product to an Existing Environment

POST /lcm/api/v1/action/add/product?id={environmentId} adds a vRealize Suite product to an existing environment.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example adds vRealize Business for Cloud to the specified environment.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/action/add/product?id=78b45f299cbdc2755585cc2bb88f8" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{
  "id": "vrbc",
  "version": "7.3.0",
  "clusterVIP": [],
  "properties": {
    "currency": "USD - US Dollar"
  },
  "nodes": [
    {
      "sourceLink": "",
      "type": "vrbc-server",
      "properties": {
        "esxHost": "$esxHostFQDN",
        "guestFullName": "SUSE Linux Enterprise 12 (64-bit)",
        "ipAddress": "$ipAddress",
        "cpu": "4",
        "vCenterHost": "$vCenterHostFQDN",
        "ssoTenant": "vsphere.local",
```

```

        "storage": "FC-LUN-15 TB",
        "userName": "Administrator@vsphere.local",
        "productName": "vRealize Business for Cloud",
        "network": "infra-traffic-1006",
        "isTelemetryEnable": "true",
        "hostname": "$hostname",
        "password": "$password",
        "productVersion": "7.2.1.10029 Build 5101870",
        "powerState": "poweredOn",
        "cafeHost": "$cafeHost",
        "memoryInMb": "8192",
        "name": "S2-vrb-server",
        "vrbRootPassword": "$vrbRootPassword"
    }
}
]
}'

```

JSON Output

When the operation completes successfully, the JSON output includes:

- id = Add a product request ID
- status = Add a product status

Add a Node

POST /lcm/api/v1/action/addNode scales out the environment by adding a single node to a product in the environment. To add multiple nodes, make a separate request for each node.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Each node request requires the following parameters:

- name
- ipAddress
- hostname

See [Node Types and Parameters](#) for a full list of parameters for each node type.

The following node types cannot be scaled:

- vra-server-primary
- db

- vrli-master
- iaas-manager-active
- master
- managementagent
- replica

curl Command

The following example adds a vrli-worker node to the environment.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/action/addNode" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d '{
  "environmentId": "8c7e3ea7e945f475558855d497c68",
  "dataCenterId": "Data Center Name",
  "productId": "vrli",
  "version": "4.3.0",
  "node": {
    "type": "vrli-worker",
    "sourceLink": "",
    "properties": {
      "name": "VM Name",
      "cluster": "",
      "ipAddress": "1.1.1.1",
      "dns": "",
      "vCenterHost": "hostname",
      "storage": "",
      "network": "",
      "hostname": "hostname",
      "netmask": "",
      "domain": "",
      "diskFormat": "",
      "searchpath": "",
      "gateway": "",
      "deployOption": "",
      "userName": "vCenter Username",
      "password": "password"
    }
  }
}'
```

JSON Output

When the operation completes successfully, the JSON output includes:

- id = Add a node request ID
- status = Add a node status

Viewing Products and Product Details

You can use the vRealize Suite Lifecycle Manager API to view products and the details of products installed in your vRealize Suite environment.

List All Products

You can list all the vRealize Suite products managed by vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 List information for all the products managed by vRealize Suite Lifecycle Manager.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/product" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

- 2 Examine the response.

The output of the request includes the following information:

- name = product name
- id = product ID
- version = product version

For example, the following response lists information about multiple versions of the vRealize Suite products vRealize Automation, vRealize Log Insight, and vRealize Business for Cloud.

```
[
  {
    "name": "vra",
    "id": "d7ca8663e452e27556df33e7c14eb",
    "version": "7.3.1"
  },
  {
    "name": "vrli",
    "id": "d7ca8663e452e27556991e197f92a",
    "version": "4.5.0"
  },
  {
    "name": "vra",
    "id": "d7ca8663e452e27556a3a36abe2a7",
    "version": "7.4.0"
  },
]
```



```
{
  "name": "vrli",
  "id": "d7ca8663e452e27556992923fd611",
  "version": "4.5.1"
}
...
{
  "name": "vrbc",
  "id": "d7ca8663e452e27556a3a36abe2a9",
  "version": "7.3.1"
},
]
```

List All Products in an Environment

You can list all the environments managed by vRealize Suite Lifecycle Manager and get information about the products in a specific environment.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 List all environments managed by vRealize Suite Lifecycle Manager.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/environment" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

- 2 Examine the response.

The output of the request includes:

- id = Environment ID
- name = Environment Name

- 3 Using the Environment ID, list information for all the products installed the environment.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/view/product?environmentId=<Environment-ID>" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token"
```

- 4 Examine the response.

The output of the request includes the following information:

- name = product name
- id = product ID

- version = product version

For example, the following response lists information about vRealize Automation installed in the environment.

```
{
  "name": "vra",
  "id": "d7ca8663e452e27556df33e7c14eb",
  "version": "7.3.1"
}
```

View Product Details

You can list the details of a vRealize Suite product managed by vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 List information for all the products managed by vRealize Suite Lifecycle Manager.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/product" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

- 2 Examine the response.

The output of the request includes the following information:

- name = product name
- id = product ID
- version = product version

For example, the following response lists information about the vRealize Suite products vRealize Automation and vRealize Business for Cloud.

```
[
  {
    "name": "vra",
    "id": "d7ca8663e452e27556df33e7c14eb",
    "version": "7.3.1"
  },
  ...
  {
    "name": "vrbc",
```

```

    "id": "d7ca8663e452e27556a3a36abe2a9",
    "version": "7.3.1"
  },
]

```

- 3 Using the product ID for vRealize Automation, list the product details.

```

curl -X GET "https://LCM-Hostname/lcm/api/v1/view/product?productId=<Product-ID>" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token"

```

- 4 The response provides details about the products installed.

For example, the following response lists details about vRealize Automation.

```

{
  "properties": [
    {
      "name": "windowsPassword",
      "value": "s2enc~hMNeyQK75zHfj4ASa432PA=="
    },
    {
      "name": "windowsUsername",
      "value": "sqa.sampaths.admin"
    }
  ],
  "clusterVIP": [
    {
      "type": "vra",
      "hostname": "lcm-vrs-48-232.sqa.local",
      "ipAddress": ""
    },
    {
      "type": "iaas-web",
      "hostname": "lcm-vrs-48-233.sqa.local",
      "ipAddress": ""
    },
    {
      "type": "iaas-manager",
      "hostname": "lcm-vrs-48-178.sqa.local",
      "ipAddress": ""
    }
  ],
  "nodes": [
    {
      "name": "iaas-manager-passive",
      "id": "d7ca8663e452e27556df33e7c20da",
      ...
    }
  ],
  "properties": [
    {
      "name": "cluster",
      "value": "LCM-Cluster"
    },
  ],
  {

```

```

        "name": "password",
        "value": "s2enc~yeJ77TXNsX9xtWK3RKRNCw=="
    },
    {
        "name": "esxHost",
        "value": ""
    },
    ...
    {
        "name": "demRole",
        "value": "DemWorker"
    }
]
}

```

Pre-Validate and Upgrade Products in Your vRealize Suite Environment

You can use the vRealize Suite Lifecycle Manager API to upgrade products installed in your vRealize Suite environment. Before upgrading, you pre-validate the upgrade request. If the pre-validation check indicates no issues with the upgrade, it is safe to proceed with the upgrade.

Procedure

1 [Create Product Upgrade Request](#)

You create a product upgrade request using the vRealize Suite Lifecycle Manager API . Then you use the output from the request to pre-validate and upgrade any vRealize Suite product deployed in your environment.

2 [Assessing vRealize Operations Manager Before Upgrading](#)

If vRealize Operations Manager is deployed in your vRealize Suite environment, run an assessment to check for pre-version 6.7 metrics. After running the assessment, you patch the request for the vRealize Operations Manager upgrade.

3 [Pre-Validate Product Upgrade](#)

Before upgrading a product in your vRealize Suite environment, you can pre-validate the product upgrade request to identify issues that can possibly cause an upgrade failure.

4 [Retrieve Pre-Validation Report for Product Upgrade](#)

After a pre-validation request finishes successfully, you can retrieve a pre-validation report. Any checks with status FAILED indicate potential problems with the product upgrade. Before proceeding with the upgrade, you should address the issues and pre-validate the upgrade request again.

5 [Patch Product Upgrade Request Then Upgrade](#)

Using a single PATCH request, you can pre-validate the upgrade request for a product deployed in your vRealize Suite environment then submit an upgrade.

Create Product Upgrade Request

You create a product upgrade request using the vRealize Suite Lifecycle Manager API . Then you use the output from the request to pre-validate and upgrade any vRealize Suite product deployed in your environment.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Create a request to upgrade a product in your environment.

```
curl -X POST "https://LCM-HostName/lcm/api/v1/action/upgrade/product" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d "{
  "environmentId": "<Environment-ID>",
  "productType": "<ProductType>",
  "productVersion": "<Product-Version>",
}"
```

Table 6-1. Input Parameters for Upgrade Product Request

Parameter	Description
environmentId	Environment ID is in the response to the request for a list of all environments managed by vRealize Suite Lifecycle Manager. See List All Products in an Environment .
productType	Supported vRealize Suite product name in vRealize Suite Lifecycle Manager. For example: vra , vrbc , vrops , vrli , or vrni .
productVersion	vRealize Suite product version post upgrade. For example, if upgrading from version x.x.x to version y.y.y, the value is y.y.y.

- 2 Examine the response to track the request.

The output includes:

- id = Product Upgrade Request ID
- status = Product Upgrade Request Status

3 (Optional) Monitor the status of the request with the Product Upgrade Request ID.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/request/status/<Product-Update-Request-ID>" -H
"accept: application/json" -H "x-xenon-auth-token: $token"
```

An upgrade request is created and moves to INITIATED status.

What to do next

Use the Product Upgrade Request ID to:

- [Pre-Validate Product Upgrade](#)
- [Retrieve Pre-Validation Report for Product Upgrade](#)
- [Patch Product Upgrade Request Then Upgrade](#)

Assessing vRealize Operations Manager Before Upgrading

If vRealize Operations Manager is deployed in your vRealize Suite environment, run an assessment to check for pre-version 6.7 metrics. After running the assessment, you patch the request for the vRealize Operations Manager upgrade.

Assess Readiness for a vRealize Operations Manager 6.7 Upgrade

If you plan to upgrade vRealize Operations Manager to version 6.7 or later, first check the output content of your existing vRealize Operations Manager deployment.

vRealize Operations Manager 6.7 or later enables different metrics than in earlier versions. Before upgrading, you use the Pre-Upgrade Assessment Tool to scan your vRealize Operations Manager output content for discontinued or disabled metrics.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Request an assessment check for a vRealize Operations Manager deployment in your vRealize Suite environment.

```
curl -X POST "https://LCM-HostName/lcm/api/v1/action/assessment/product" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d "{
  "environmentId": "<Environment-ID>",
  "requestId": "<request-ID>"
}"
```

Table 6-2. Input Parameters for Assessment Request

Parameter	Description
environmentId	Environment ID is in the response to the request for a list of all environments managed by vRealize Suite. See List All Products in an Environment
requestId	ID received from the response of the call POST /action/upgrade/product. See Create Product Upgrade Request .

2 Examine the response.

The output includes:

- id: Assessment Check Request ID
- status: Status of the request

3 Monitor the status of the request with the Assessment Check Request ID.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/request/status/{Assessment-Check-Request-ID}" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json"
```

4 Once the assessment check reaches COMPLETED status, request the assessment report.

```
curl -X GET "https://LCM-HostName/lcm/api/request/{requestId}" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json"
```

Where the *requestId* is from the response of the call POST /action/upgrade/product. See [Create Product Upgrade Request](#).

5 Examine the response to obtain the value for resultText under the resultData property.

The following snippet is from an example response.

```
{
  "requestId": "d7ca8663e452e275572731276c3a3",
  "requestState": "COMPLETED",
  "requestSource": "d7ca8663e452e27556991da97e3ba",
  "requestData": "{...}",
  "requestDataType": "com.vmware.vrealize.lcm.nxui.document.GuiRequest",
  "executionPath": "{...}",
  "requestStatus": "{...}",
  "requestType": "APUAT_CHECK",
  "resultData": "[
    {
      "vmid": "c4de1537-c9bf-4632-bea6-14c262c9f119",
      "createdOn": 1533213518038,
      "lastUpdatedOn": 1533213518038,
      "version": 0,
```

```

    "sourceOfTheRequest": "c13ca5c1-ad39-4997-a129-d5e4e5dcd9d1",
    "resultKey": "apuatReport",
    "resultType": "java.lang.String",
    "resultText": ""/data/APUAT/APUAT_BY_LCM_d7ca8663e452e27556991da97e3ba/report/index.html"",
    "eventId": "8febbdf6-3080-4dcb-a456-8f4495d0d18e",
    "machineInstanceId": "0d27e506-8d73-4a68-9964-9e0080f2c6ab"
  }],
  "documentVersion": 17
  "documentKind": "com:vmware:vrealize:lcm:common:documents:nxrequest:GenericRequest",
  "documentSelfLink": "/lcm/api/request/d7ca8663e452e275572731276c3a3",
  "documentUpdateTimeMicros": 1533213521931004,
  "documentUpdateAction": "PATCH",
  "documentExpirationTimeMicros": 0,
  "documentAuthPrincipalLink": "/core/authz/system-user"
}

```

What to do next

Use the value for `resultText` to view the assessment report from the appliance console. To obtain the same report from a web browser, replace `/data/` with `https://LCM-hostname:4443/` in the URL. To access the web link, you must provide UI administrator credentials.

The assessment report identifies the vRealize Operations Manager output content such as dashboards, alerts, and reports that will be affected by the upgrade. For more information about the upgrade assessment tool and how to act on information provided as output, see the VMware Knowledge Base article: <https://kb.vmware.com/s/article/53545>.

Set Assessment Check for vRealize Operations Manager 6.7 or Later

After reviewing the report produced by the assessment check, you patch the upgrade request to indicate that you have viewed the results.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

- 1 Patch the vRealize Operations Manager upgrade request.

```

curl -X PATCH "https://LCM-HostName/lcm/api/v1/action/assessment/product" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d "{

```



```

    "environmentId": "<Environment-ID>",
    "requestId": "<request-ID>"
    "viewed": "<is-viewed>"
  }"

```

Table 6-3. Input Parameters to Patch vRealize Operations Manager Upgrade Request

Parameter	Description
environmentId	Environment ID is in the response to the request for a list of all environments managed by vRealize Suite. See List All Products in an Environment
requestId	ID received from the response of the call POST /action/upgrade/product. See Create Product Upgrade Request .
viewed	If you reviewed the report from Assess Readiness for a vRealize Operations Manager 6.7 Upgrade and determined that the discontinued or disabled metrics will have no impact on your vRealize Operations Manager 6.7 deployment, set the value to True . Otherwise, set the value to False to prevent an upgrade to vRealize Operations Manager 6.7.

2 Examine the response.

The output of the request includes:

- id = Product upgrade request ID, provided as input.
- status = Patch request status

Pre-Validate Product Upgrade

Before upgrading a product in your vRealize Suite environment, you can pre-validate the product upgrade request to identify issues that can possibly cause an upgrade failure.

Pre-validation is an optional step that ensures a smooth product upgrade using vRealize Suite Lifecycle Manager. To pre-validate as part of the upgrade process, you can set the `preValidate` toggle in the PATCH /action/upgrade/product request. See [Patch Product Upgrade Request Then Upgrade](#).

Pre-validation is particularly useful when upgrading from an older version of a product or in a production environment where a clean upgrade is critical. However, a best practice is to pre-validate every product upgrade request.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

1 Pre-validate your product upgrade request.

```
curl -X POST "https://LCM-HostName/lcm/api/v1/action/prevalidate/upgrade/product" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d "{
  "environmentId": "<Environment-ID>",
  "productType": "<ProductType>",
  "repositoryType": "URL",
  "repositoryUrl": "<Product-Upgrade-File-URL>",
  "productVersion": "<Product-Version>",
  "requestId": "<request-ID>",
  "preValidate": "<Prevalidate-True-or-False>",
  "requestState": "SUBMITTED",
}"
```

Table 6-4. Input Parameters for Product Upgrade Request Pre-Validation

Parameter	Description
environmentId	Environment ID is in the response to the request for a list of all environments managed by vRealize Suite Lifecycle Manager. See List All Products in an Environment
productType	Supported vRealize Suite product name in vRealize Suite Lifecycle Manager. For example: vra , vrbc , vrops , vrli , or vrni .
repositoryType	Repository Type can be URL or lcmrepository
repositoryUrl	The URL points to the upgrade binary file. For example: <code>https://<LCM-IP>:4443/productlinks/vrli/4.6.0/upgrade/<UpgradeFileName></code>
productVersion	vRealize Suite product version post upgrade. For example, if upgrading from version x.x.x to version y.y.y, the value is y.y.y.
requestId	ID received from the response of the call POST /action/upgrade/product. See Create Product Upgrade Request .
preValidate	Option to toggle the pre-validation before actual upgrade flow: true or false .
requestState	Request state to patch the existing state. Default is SUBMITTED .

2 Examine the response to track the request.

The output includes:

- id = Product Update Request ID provided as input

- status = Request status

3 (Optional) Monitor the status of the request with the Product Update Request ID.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/request/status/<Product-Update-Request-ID>" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

Retrieve Pre-Validation Report for Product Upgrade

After a pre-validation request finishes successfully, you can retrieve a pre-validation report. Any checks with status FAILED indicate potential problems with the product upgrade. Before proceeding with the upgrade, you should address the issues and pre-validate the upgrade request again.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that you have the Product ID for the product you want to upgrade and the Environment ID for the environment where the product is to be upgraded. To obtain the Environment ID and Product ID, see the responses to requests in [List All Products in an Environment](#).

Procedure

- ◆ Retrieve the pre-validation report.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/view/prevalidation/report/upgrade/product?environmentId={Environment-ID}&productId={Product-ID}&requestId={Request-ID}" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token"
```

- *Environment-ID* identifies the environment where the product to be upgraded is deployed.
- *Product-ID* identifies the product to upgrade.
- *Request-ID* is the Product Upgrade Request ID. See [Create Product Upgrade Request](#).

The report includes two sections with the `checkName` parameter. These sections display output for Data Validation and Product Validation.

The following report is the response for a pre-validation check for a product upgrade from vRealize Log Insight 4.5.1 to vRealize Log Insight 4.6.0.

```
{
  "rootValidations": [
    {
      "checkName": "Data validation",
      "status": "PASSED",
      "elementType": "GROUP",
      "childElements": [
```

```

    {
      "checkName": "Version support from LCM.",
      "checkType": "ERROR",
      "status": "PASSED",
      "resultDescription": "Check if the selected version is supported by LCM policy.",
      "elementType": "CHECK"
    },
    {
      "checkName": "File format check.",
      "checkType": "ERROR",
      "status": "PASSED",
      "resultDescription": "Check if a file with valid format is selected.",
      "elementType": "CHECK"
    }
  ]
},
{
  "checkName": "vRealize Log Insight Validations",
  "status": "PASSED",
  "elementType": "GROUP",
  "childElements": [
    {
      "checkName": "SSH check.",
      "checkType": "ERROR",
      "status": "PASSED",
      "resultDescription": "Check if SSH is enabled.",
      "elementType": "CHECK"
    },
    {
      "checkName": "Valid IP/FQDN (master) for upgrade.",
      "checkType": "ERROR",
      "status": "FAILED",
      "resultDescription": "An upgrade should be triggered with master node's IP/FQDN. Here
IP/FQDN used is wdc-interop29-67.sqa.local.",
      "elementType": "CHECK"
    },
    {
      "checkName": "Cluster setup check for vRLI.",
      "checkType": "ERROR",
      "status": "PASSED",
      "resultDescription": "Either have a standalone cluster setup or have minimum 3 to a maximum
12 nodes for a valid cluster setup.",
      "elementType": "CHECK"
    },
    {
      "checkName": "All nodes should be connected.",
      "checkType": "ERROR",
      "status": "PASSED",
      "resultDescription": "All nodes connection status should be either CONNECTED or MAINTENANCE
for upgrade.",
      "elementType": "CHECK"
    },
    {
      "checkName": "NTP settings check.",
      "checkType": "WARNING",

```

```

        "status": "PASSED",
        "resultDescription": "Check if NTP is configured with the same NTP servers on all the
nodes.",
        "elementType": "CHECK"
    },
    {
        "checkName": "Version check for upgrade.",
        "checkType": "ERROR",
        "status": "PASSED",
        "resultDescription": "Current version is 4.5.0, Version to upgrade from file 4.6.0",
        "elementType": "CHECK"
    },
    {
        "checkName": "Disk space check on the root filesystem.",
        "checkType": "ERROR",
        "status": "PASSED",
        "resultDescription": "Disk space on the root file system should be more than 1 GB. Present
space is 12GB.",
        "elementType": "CHECK"
    },
    {
        "checkName": "/storage/var disk space check.",
        "checkType": "WARNING",
        "status": "PASSED",
        "resultDescription": "/storage/var should not be full. Currently its usage is 40 percent.",
        "elementType": "CHECK"
    },
    {
        "checkName": "TLS check.",
        "checkType": "WARNING",
        "status": "PASSED",
        "resultDescription": "Check if TLSv1 and TLSv1.1 is enabled.",
        "elementType": "CHECK"
    }
]
},
"overallStatus": true,
"requestState": "COMPLETED",
"lastUpdateTimeMillis": 1532515260057,
"startTimeMillis": 1532515060839
}

```

This example shows a failure in the check for Valid IP/FQDN (master) for upgrade. Follow the instructions in the result description to correct the problem, run the pre-validation request, and review the report again to verify that all checks passed before performing the product upgrade.

Patch Product Upgrade Request Then Upgrade

Using a single PATCH request, you can pre-validate the upgrade request for a product deployed in your vRealize Suite environment then submit an upgrade.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- If you are planning to upgrade vRealize Operations Manager, verify that you have completed the assessment check. See [Assessing vRealize Operations Manager Before Upgrading](#).

Procedure

- 1 Upgrade a product in your environment.

```
curl -X PATCH "https://LCM-HostName/lcm/api/v1/action/upgrade/product" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d "{
  "environmentId": "<Environment-ID>",
  "productType": "<ProductType>",
  "repositoryType": "URL",
  "repositoryUrl": "<Product-Upgrade-File-URL>",
  "productVersion": "<Product-Version>",
  "requestId": "<request-ID>",
  "preValidate": "<Prevalidate-True-or-False>",
  "requestState": "<Request-State-Post-Patching>",
}"
```

Table 6-5. Input Parameters to Patch Product Upgrade Request

Parameter	Description
environmentId	Environment ID is in the response to the request for a list of all environments managed by vRealize Suite Lifecycle Manager. See List All Products in an Environment .
productType	Supported vRealize Suite product name in vRealize Suite Lifecycle Manager. For example: vra , vrbc , vrops , vrli , or vrni .
repositoryType	Repository Type can be URL or lcmrepository .
repositoryUrl	The URL points to the upgrade binary file. For example: <code>https://<LCM-IP>:4443/productlinks/vrli/4.6.0/upgrade/<UpgradeFileName></code> .
productVersion	vRealize Suite product version post upgrade. For example, if upgrading from version x.x.x to version y.y.y, the value is y.y.y.
requestId	ID received from the response of the call POST <code>/action/upgrade/product</code> . See Create Product Upgrade Request .

Table 6-5. Input Parameters to Patch Product Upgrade Request (Continued)

Parameter	Description
preValidate	Option to pre-validate the upgrade request before starting the upgrade: true or false .
requestState	Request state to patch the existing state. Default is SUBMITTED .

Note If `preValidate=true` and the pre-validation check fails, the patch is unsuccessful and the upgrade process does not start. If you assess that the failure to pre-validate is not critical to the operation of the product, set `preValidate=false` so that the upgrade will run.

2 Examine the response to track the request.

The output includes:

- `id` = Product Update Request ID provided as input.
- `status` = Request status

3 (Optional) Monitor the status of the request with the Product Update Request ID.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/request/status/<Product-Update-Request-ID>" -H
"accept: application/json" -H "x-xenon-auth-token: $token"
```

Collect Environment Logs

POST `/lcm/api/v1/collectlogs/environment` collects log files for all vRealize Suite Lifecycle Manager environments that you can use for troubleshooting.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example collects a support bundle of logs for the environment.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/collectlogs/environment" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
```

JSON Output

The following JSON output is returned based on the command input.

```
{
  "requestId": "78b45f299cbdc2755589c4ee37bf8"
}
```

Configuration Drift

Configuration drift shows the changes in product configuration over time and allows you to revert a product to an earlier configuration state.

Create a Configuration Drift Baseline

POST /lcm/api/v1/drift/lcmtemplate captures a product's configuration parameters at a given time as a baseline for configuration drift reports.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example saves the current vRealize Operations Manager product configuration as the vRealize Operations Manager configuration drift baseline.

```
curl -X POST "https://LCM-Hostname/lcm/api/v1/drift/lcmtemplate" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-d '{
  "environment": "Sample Environment Name",
  "product": "vrops"
}'
```

JSON Output

The following JSON output is returned when the product baseline is successfully saved.

```
200    Operation successful
```

Retrieve a Configuration Drift Baseline

GET /lcm/api/v1/drift/lcmtemplate retrieves the configuration drift baseline for a product.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example retrieves the vRealize Operations Manager configuration drift baseline.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/drift/lcmtemplate" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-d '{
    "environment": "Sample Environment Name",
    "product": "vrops"
}'
```

JSON Output

The following JSON output is returned when the product baseline is successfully retrieved.

```
200    Operation successful
```

Retrieve a Configuration Drift Report

GET `/lcm/api/v1/drift/report` retrieves a configuration drift report that shows the changes in a product's current configuration compared to the product's configuration drift baseline.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example retrieves the configuration drift report for vRealize Operations Manager.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/drift/report" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
-d '{
    "environment": "Sample Environment Name",
    "product": "vrops"
}'
```

JSON Output

The following JSON output is returned with the configuration drift report when the product baseline is successfully retrieved.

```
200      Operation successful
```

Remediating Configuration Drift

As the configuration of products installed in your vRealize Suite environment change over time, you can use the vRealize Suite Lifecycle Manager API to remediate the changes.

Trigger a Remediation

If a remediation report indicates a configuration drift in the vRealize Suite environment, you can use vRealize Suite Lifecycle Manager to trigger a remediation.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that remediation is required. See [Retrieve a Remediation Report](#).

Procedure

- 1 Trigger the remediation request.

```
curl -X POST "https://LCM-HostName/lcm/api/v1/remediation" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

The body of the JSON input specifies the product configuration that you want to restore such as, vRealize Automation, vRealize Operations Manager, vRealize Log Insight, or vRealize Business for Cloud. For a remediation to occur, the `isRemediationEnabled` parameter value must be set to `true` as in the following input example to remediate a change in the DNS server for vRealize Automation version 7.3.

```
{
  "updateTime": "2018-06-28T11:09:28.632Z[Etc/UTC]",
  "productType": "vra-7.3.0",
  "productNames": [
    "764b17cc6f57875569146ef4e5a2"
  ],
  "nodes": [
    {
      "type": "vRealize Automation",
      "nodeIdField": "APPLICATION",
```

```

    "driftedProducts": [],
    "categories": [
      {
        "categoryType": "Tenant",
        "categoryId": "xxyyzz",
        "isRetry": false,
        "isRemediationEnabled": true,
        "isGroupRemediation": true,
        "driftedProducts": [],
        "property": [
          {
            ...
            "name": "DNS",
            "isRemediationEnabled": true,
            "isRetry": false,
            "values": [
              {
                "productName": "TEMPLATE",
                "value": "10.141.66.213,10.118.183.252"
              },
              {
                "productName": "764b17cc6f57875569146ef4e5a2",
                "value": "10.118.183.252"
              }
            ],
            "remediationCandidate": "TEMPLATE",
          },
          ...
        ]
      }
    ]
  }
}

```

2 Examine the response to verify whether the operation was successful.

Retrieve a Remediation Report

If a change occurs in a vRealize Suite product following a configuration drift, you can get a remediation report.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use `POST /lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

Procedure

1 Retrieve a remediation report.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/remediation" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

The following input is optional:

- environment = environment name
- requestId = request ID
- product = product type

2 To identify any changes, examine the response. A remediation report is only available if a configuration drift in the vRealize Suite environment has occurred.

In the following code snippet from a response body, "driftedProducts":

["764b17cc6f57875569146ef4e5a2"] shows a change in vRealize Automation version 7.3.

```
{
  "updateTime": "2018-06-28T11:09:28.632Z[Etc/UTC]",
  "isRetry": false,
  "productType": "vra-7.3.0",
  "productNames": [
    "764b17cc6f57875569146ef4e5a2"
  ],
  "nodes": [
    {
      ...
    },
    {
      "type": "vRealize Automation Cafe Virtual Machine",
      "nodeIdField": "vra2-va.sqa.local",
      "driftedProducts": [
        "764b17cc6f57875569146ef4e5a2"
      ],
      "categories": [
        {
          ...
        },
        {
          "categoryType": "vRealize Automation VM Network Information",
          "categoryId": "10.158.129.11",
          "isRetry": false,
          "isRemediationEnabled": true,
          "driftedProducts": [
            "764b17cc6f57875569146ef4e5a2"
          ],
          "property": [
            {
              "name": "ipaddress",
              "isRemediationEnabled": false,
              "isRetry": false,
```

```

        "values": [
            {
                "productName": "TEMPLATE",
                "value": "10.158.129.11"
            },
            {
                "productName": "764b17cc6f57875569146ef4e5a2",
                "value": "10.158.129.11"
            }
        ]
    },
    {
        "name": "ethernet",
        "isRemediationEnabled": false,
        "isRetry": false,
        "values": [
            {
                "productName": "TEMPLATE",
                "value": "eth0"
            },
            {
                "productName": "764b17cc6f57875569146ef4e5a2",
                "value": "eth0"
            }
        ]
    },
    {
        ...
    },
    {
        "name": "passphrase",
        "isRemediationEnabled": true,
        "isRetry": false,
        "values": [
            {
                "productName": "TEMPLATE",
                "value": ""
            },
            {
                "productName": "764b17cc6f57875569146ef4e5a2",
                "value": ""
            }
        ]
    }
],
    "subCategory": []
},
]
},
{
    "type": "vRealize Automation Cafe Virtual Machine",
    "nodeIdField": "vra2-web.sqa.local",
    "driftedProducts": [],
    "categories": [
        {

```

```
        ...  
      }  
    ]  
  }  
],  
"documentVersion": 0,  
"documentUpdateTimeMicros": 0,  
"documentExpirationTimeMicros": 0  
}
```

What to do next

If a change has occurred, you can trigger a drift remediation. See [Trigger a Remediation](#).

Working with Nodes

You can add nodes to vRealize Suite products to scale out your environment.

Each vRealize Suite product managed by vRealize Suite Lifecycle Manager includes nodes, or components, that provide functionality for the products. When you install vRealize Suite products using vRealize Suite Lifecycle Manager, you select which nodes to install with each product. As the size or needs of your environment grow, you can scale out the environment by adding additional nodes.

- [View All Nodes and Node IDs](#)

GET `/lcm/api/v1/view/node` displays the nodes and node IDs managed by vRealize Suite Lifecycle Manager.

- [View Details for a Node](#)

GET `/lcm/api/v1/view/node?nodeID=$nodeID` displays details for the specified node.

- [View Node Types for a Product](#)

GET `/lcm/api/v1/productinfo?productId=$product&version=$productVersionNumber$queryFor=listOfComponents` displays the node types for the specified product and product version.

- [Node Types and Parameters](#)

vRealize Suite Lifecycle Manager supports the following node types and parameters.

View All Nodes and Node IDs

GET `/lcm/api/v1/view/node` displays the nodes and node IDs managed by vRealize Suite Lifecycle Manager.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example requests details for all nodes.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/view/node" \
  -H "accept: application/json" \
  -H "x-xenon-auth-token: $token" \
```

JSON Output

The following JSON output is returned based on the command input.

```
[
  {
    "name": "vra-server-primary",
    "id": "78b45f299cbdc2755585ce13bab8"
  },
  {
    "name": "managementagent",
    "id": "78b45f299cbdc2755585ce13babd9"
  },
  {
    "name": "iaas-web",
    "id": "78b45f299cbdc2755585ce13babd7"
  },
  {
    "name": "iaas-dem-worker",
    "id": "78b45f299cbdc2755585ce13bb3ae"
  },
  {
    "name": "proxy-agent-vmware",
    "id": "78b45f299cbdc2755585ce13bb3a8"
  }
]
```

View Details for a Node

GET /lcm/api/v1/view/node?nodeID=\$nodeID displays details for the specified node.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

- Verify that you have the node ID for the node. See [View All Nodes and Node IDs](#).

curl Command

The following example requests details for a data node.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/view/node?nodeID=a3b2e69ac10ad27555e68c9cc89ab" \
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
```

JSON Output

The following JSON output is returned based on the command input.

```
{
  "name": "data",
  "id": "a3b2e69ac10ad27555e68c9cc89ab",
  "properties": [
    {
      "name": "ipAddress",
      "value": "10.000.000.000"
    },
    {
      "name": "dns",
      "value": "10.000.000.001,10.000.000.002"
    },
    {
      "name": "hostname",
      "value": "host.example.local"
    },
    {
      "name": "domain",
      "value": "example.local"
    },
    {
      "name": "netmask",
      "value": "255.255.255.0"
    },
    {
      "name": "diskFormat",
      "value": "Thin"
    },
    {
      "name": "name",
      "value": "data-host"
    },
    {
      "name": "storage",
      "value": "FC_LUN_1TB_1"
    },
    {
      "name": "cluster",
      "value": "CLUSTER#SS_MGMT_Cluster_TB1"
    }
  ]
}
```

```

    },
    {
      "name": "vCenterHost",
      "value": "vcenter.example.local"
    },
    {
      "name": "network",
      "value": "NTWK_121"
    },
    {
      "name": "userName",
      "value": "Administrator@vsphere.local"
    },
    {
      "name": "searchpath",
      "value": "example.local"
    },
    {
      "name": "gateway",
      "value": "10.000.000.003"
    },
    {
      "name": "address",
      "value": "10.000.000.000"
    },
    {
      "name": "password",
      "value": "password"
    },
    {
      "name": "rootPassword",
      "value": "password"
    }
  ]
}

```

View Node Types for a Product

GET /lcm/api/v1/productinfo?productId=*\$product*&version=*\$productVersionNumber*
 \$queryFor=listOfComponents displays the node types for the specified product and product version.

Prerequisites

Satisfy the following conditions before performing any tasks for this use case.

- Use POST /lcm/api/v1/login to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).

curl Command

The following example requests node types for vRealize Automation 7.2.

```
curl -X GET "https://LCM-Hostname/lcm/api/v1/productinfo?
productId=vra&version=7.2$queryFor=listOfComponents"
-H "accept: application/json" \
-H "x-xenon-auth-token: $token" \
```

JSON Output

The following JSON output is returned based on the command input.

```
[{"db","iaas-dem-orchestrator","iaas-dem-worker","iaas-manager-active","iaas-manager-passive","iaas-
web","managementagent","proxy-agent-vsphere","vra-server-primary","vra-server-secondary"]
```

Node Types and Parameters

vRealize Suite Lifecycle Manager supports the following node types and parameters.

vrli-worker

Parameter	Description
name	Specifies the virtual machine name
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
ipAddress	Specifies the IP address for the virtual machine.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
domain	Specifies the domain name.
vrliRootPassword	Specifies the root password for the vRealize Log Insight appliance.
diskFormat	Specifies whether the disk format is thick or thin.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.
deployOption	Specifies the deployment option: small, medium, or large.

Parameter	Description
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.

vrli-master

Parameter	Description
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
vrliLicenseKey	Specifies the vRealize Log Insight license key.
installerLocation	Specifies the file path location where the OVA is stored.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
ipAddress	Specifies the virtual machine IP address.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the vCenter datastore name.
userName	Specifies the vCenter user name.
uberAdmin	Specifies the vIDM user to be assigned the product administrator role post-deployment.
network	Specifies the network name.
masterVidmEnabled	Specifies whether to configure vRealize Log Insight with vRealize Suite Lifecycle Manager vIDM. Set to <code>true</code> or <code>false</code> .
vrliAdminEmail	Specifies the vRealize Log Insight administrator email address.
hostname	Specifies the host name for the virtual machine.
password	Specifies the vCenter password.
netmask	Specifies the netmask IP.
domain	Specifies the domain name.
name	Specifies the virtual machine name.
diskFormat	Specifies whether the disk format is thick or thin.
vrliClusterVips	Specifies the vRealize Log Insight cluster vIPS in the format <i>vRLI-Cluster-IP#vRLI-Cluster-IP-Hostname</i> .
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.

vrb-collector

Parameter	Description
name	Specifies the virtual machine name.
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
vrbTelemetryEnabled	Flag to enable telemetry in vRealize Business for Cloud.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
diskFormat	Specifies whether the disk format is thick or thin.
vrbCurrency	Specifies the vRealize Business for Cloud currency format.
sshEnabled	Specifies whether SSH is enabled in sshEnabled. Set to true or false.
ipAddress	Specifies the IP address for the virtual machine.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
vrbLicenseKey	Specifies the vRealize Business for Cloud license key.
isTelemetryEnable	Specifies whether telemetry is enabled in vRealize Business for Cloud. Set to true or false.
domain	Specifies the domain name.
vrbRootPassword	Specifies the root password for the vRealize Business for Cloud appliance.
uberAdmin	Specifies the vIDM user to be assigned the product administrator role post-deployment.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.

vrb-server

Parameter	Description
name	Specifies the virtual machine name.
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
vrbTelemetryEnabled	Flag to enable telemetry in vRealize Business for Cloud.

Parameter	Description
tenantPassword	Specifies the password for the tenant user who has the business management administrator role in vRealize Automation. Used for vRB-vRA registration.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
tenantUser	Specifies the user name for the tenant user who has the business management administrator role in vRealize Automation. Used for vRB-vRA registration.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
diskFormat	Specifies whether the disk format is thick or thin.
vrbcurrency	Specifies the vRealize Business for Cloud currency format.
sshEnabled	Specifies whether SSH is enabled in sshEnabled. Set to true or false.
ipAddress	Specifies the IP address for the virtual machine.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
vrblicenseKey	Specifies the vRealize Business for Cloud license key.
isTelemetryEnable	Specifies whether telemetry is enabled in vRealize Business for Cloud. Set to true or false.
domain	Specifies the domain name.
vrbrRootPassword	Specifies the root password for the vRealize Business for Cloud appliance.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.
isStandalone	Specifies whether vRealize Business for Cloud is provisioned as a standalone deployment. Set to true or false.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.
uberAdmin	Specifies the vIDM user to be assigned the product administrator role post-deployment.

iass-dem-orchestrator

Parameter	Description
name	Specifies the virtual machine name
ipAddress	Specifies the IP address for the virtual machine.
vCenterHost	Specifies the vCenter host name.

Parameter	Description
demUserName	Specifies the DEM user name in <i>domain\username</i> format. If not provided, the default Windows user name given at the product level is used.
demPassword	Specifies the DEM password. If not provided, the default Windows password set at the product level is used.
hostname	Specifies the host name for the node.
installationPath	Specifies the Windows file path location where this component is installed.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.

iaas-manager-passive

Parameter	Description
name	Specifies the virtual machine name
msUserName	Specifies the IaaS service user name in <i>domain\username</i> format. If not provided, the default Windows user name given at the product level is used.
msPassword	Specifies the IaaS service user password. If not provided, the default Windows password set at the product level is used.
ipAddress	Specifies the IP address for the virtual machine.
hostname	Specifies the host name for the node.
installationPath	Specifies the Windows file path location where this component is installed.
vCenterHost	Specifies the vCenter host name.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.

iaas-web

Parameter	Description
name	Specifies the virtual machine name
webUserName	Specifies the IaaS service user name in <i>domain\username</i> format. If not provided, the default Windows user name given at the product level is used.
webPassword	Specifies the IaaS service user password. If not provided, the default Windows password set at the product level is used.
ipAddress	Specifies the IP address for the virtual machine.
hostname	Specifies the host name for the node.

Parameter	Description
installationPath	Specifies the Windows file path location where this component is installed.
vCenterHost	Specifies the vCenter host name.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.

proxy-agent-vsphere

Parameter	Details
name	Specifies the virtual machine name
agentUserName	Specifies the IaaS service user name in <i>domain\username</i> format. If not provided, the default Windows user name given at the product level is used.
ipAddress	Specifies the IP address for the virtual machine.
vCenterHost	Specifies the vCenter host name.
agentPassword	Specifies the IaaS service user password. If not provided, the default Windows password set at the product level is used.
hostname	Specifies the host name for the node.
installationPath	Specifies the Windows file path location where this component is installed.
vsphereEndpointName	Specifies the vCenter endpoint name.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.

vra-server-secondary

Parameter	Description
name	Specifies the virtual machine name
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
sshEnabled	Specifies whether SSH is enabled on the appliance. Set to <i>true</i> or <i>false</i> .
ipAddress	Specifies the IP address for the virtual machine.

Parameter	Description
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
domain	Specifies the domain name.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.

vra-server-primary

Parameter	Description
vidmVraDisabledAdvanced	Federates vRealize Automation internal vIDM with vRealize Suite Lifecycle Manager vIDM. Set <code>true</code> to disable or <code>false</code> to enable.
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
installerLocation	Specifies the file path location where the OVA is stored.
ipAddress	Specifies the IP address for the virtual machine.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
userName	Specifies the vCenter administrator user name.
network	Specifies the network name.
vidmPassword	Specifies the vRealize Automation default SSO administrator password.
licenseKey	Specifies the vRealize Automation license key.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
password	Specifies the vCenter administrator password.
name	Specifies the virtual machine name
domain	Specifies the domain name.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.

db

Parameter	Description
useWindowsAuthentication	Specifies whether to use Windows authentication for IaaS DB. Set to <code>true</code> or <code>false</code> .
databaseUserName	Specifies the SA user name. Used in case Windows authentication is set to <code>false</code> .
databasePassword	Specifies the SA password. Used in case Windows authentication is set to <code>false</code> .
useExistingDatabase	Specifies whether to use an existing database within the machine. Set to <code>true</code> or <code>false</code> .
password	Specifies the vCenter administrator password.
hostname	Specifies the host name for the node.
databaseName	Specifies the IaaS database name.
name	Specifies the virtual machine name
ipAddress	Specifies the IP address for the virtual machine.
vCenterHost	Specifies the vCenter host name.
userName	Specifies the vCenter administrator user name.

iaas-manager-active

Parameter	Description
password	Specifies the vCenter administrator password.
hostname	Specifies the host name for the node.
msUserName	Specifies the IaaS service user name in <code>domain\username</code> format. If not provided, the default Windows user name given at the product level is used.
msPassword	Specifies the IaaS service user password. If not provided, the default Windows password set at the product level is used.
name	Specifies the virtual machine name
ipAddress	Specifies the IP address for the virtual machine.
vCenterHost	Specifies the vCenter host name.
userName	Specifies the vCenter administrator user name.

master

Parameter	Description
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
installerLocation	Specifies the file path location where the OVA is stored.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
password	Specifies the vCenter administrator password.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
diskFormat	Specifies whether the disk format is thick or thin.
ntpServer	Specifies the NTP server IP address.
ipAddress	Specifies the IP address for the virtual machine.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
userName	Specifies the vCenter administrator user name.
masterVidmEnabled	Specifies whether to register with vRealize Suite Lifecycle Manager vIDM. Set <code>true</code> or <code>false</code> .
license	Specifies the license key.
name	Specifies the virtual machine name
domain	Specifies the domain name.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.

replica

Parameter	Description
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
ipAddress	Specifies the IP address for the virtual machine.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
userName	Specifies the vCenter administrator user name.

Parameter	Description
password	Specifies the vCenter administrator password.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
name	Specifies the virtual machine name
domain	Specifies the domain name.
diskFormat	Specifies whether the disk format is thick or thin.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.

data

Parameter	Description
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
ipAddress	Specifies the IP address for the virtual machine.
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
userName	Specifies the vCenter administrator user name.
extendedStorage	Specifies the vCenter datastore name for extended storage.
password	Specifies the vCenter administrator password.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
name	Specifies the virtual machine name
domain	Specifies the domain name.
diskFormat	Specifies whether the disk format is thick or thin.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.

remotecollector

Parameter	Description
cluster	Specifies the cluster name in vCenter in the following format: <i>data_center_name#vcenter_cluster_name</i>
ipAddress	Specifies the IP address for the virtual machine.

Parameter	Description
dns	Specifies the DNS IP address. Use a comma to separate multiple addresses.
vCenterHost	Specifies the vCenter host name.
storage	Specifies the datastore name.
network	Specifies the network name.
userName	Specifies the vCenter administrator user name.
password	Specifies the vCenter administrator password.
hostname	Specifies the host name for the node.
netmask	Specifies the netmask IP address.
name	Specifies the virtual machine name
domain	Specifies the domain name.
diskFormat	Specifies whether the disk format is thick or thin.
ntpServer	Specifies the NTP server IP address.
searchpath	Specifies the DNS search path.
gateway	Specifies the gateway IP address.

Working with Requests

You use the vRealize Suite Lifecycle Manager request API to monitor requests. In case of failure, you can also retry requests.

Types of requests include:

- Creating an environment
- Creating a product snapshot
- Installing vRealize Suite Lifecycle Manager upgrades
- Adding and managing content from Marketplace
- Configuring My VMware settings

This chapter includes the following topics:

- [Monitor a Request](#)
- [Retry a Request](#)

Monitor a Request

To monitor the status of a vRealize Suite Lifecycle Manager request, you provide a request ID.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that you have obtained the request ID from a response to [Create an Environment](#), [Add a Product to an Existing Environment](#), [Add a Node](#), or [Create Product Upgrade Request](#).

Procedure

- 1 Get the status of a request with a request ID.

```
curl -X GET "https://LCM-HostName/lcm/api/v1/request/status/<Request-ID>" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```

2 Examine the response for the status request.

The output of the request includes:

- id = Request ID
- type = Request Type
- state = Request State
- status = Request Status
- isRetriable = shows where the request is retrievable in case of any failures.
- retryParameters = provides the retry parameters in case the retry is possible by modifying the input parameters.

Retry a Request

If a request fails, you can retry the request to create an environment, upgrade vRealize Suite Lifecycle Manager, or download from my.vmware.com.

Prerequisites

Satisfy the following conditions before performing the tasks for this use case.

- Use POST `/lcm/api/v1/login` to log in to vRealize Suite Lifecycle Manager.
- Verify that you have a valid authentication token that matches your login credentials. See [Request an Authentication Token](#).
- Verify that you have obtained the request ID from a response to [Create an Environment](#), [Add a Product to an Existing Environment](#), [Add a Node](#), or [Create Product Upgrade Request](#).

Procedure

1 Retry a request with a request ID.

```
curl -X POST "https://LCM-HostName/lcm/api/v1/request/retry/<Request-ID>" \
-H "accept: application/json" -H "x-xenon-auth-token: $token" \
-H "content-type: application/json" \
-d "[
  {
    "eventId": "<Retry-Event-ID>",
    "messageId": "<Retry-Message-ID>",
    "message": "Error Message",
    "properties": {
      "retrykey": "newvalue"
    }
  }
]"
```

Where the request body input parameters such as eventId, messageId, message, and properties with the retrykey list are retry parameters listed in the response to a status request. See [Monitor a Request](#).

2 Examine the response to track the retry request.

The output of the request includes:

- requestId = Retry Request ID
- requestState = Retry Request State
- requestType = Retry Request Type

3 (Optional) Monitor the status of the request with the Retry Request ID from [Step 2](#).

```
curl -X GET "https://LCM-HostName/lcm/api/v1/request/status/<Retry-Request-ID>" -H "accept: application/json" -H "x-xenon-auth-token: $token"
```