

# RIFT.ware™ version 8.1.1.0.114767

Release Notes  
June 2020

## Notice

---

The information and descriptions contained herein embody confidential and proprietary information that is the property of RIFT, Inc. Such information and descriptions may not be copied, reproduced, disclosed to others, published or used, in whole or in part, for any purpose other than that for which it is being made available without the express prior written permission of RIFT, Inc.

Nothing contained herein shall be considered a commitment by RIFT to develop or deliver such functionality at any time. RIFT reserves the right to change, modify, or delete any item(s) at any time for any reason.

## Table of Contents

---

|                                                                                           |           |
|-------------------------------------------------------------------------------------------|-----------|
| <b>Notice.....</b>                                                                        | <b>2</b>  |
| <b>RIFT.ware 8.1.1.0 Release Notes .....</b>                                              | <b>4</b>  |
| <b>New and Changed Features for 8.1.1 .....</b>                                           | <b>5</b>  |
| <b>Dynamic Editing to a Network Service with CNFs.....</b>                                | <b>6</b>  |
| UI Enhancements .....                                                                     | 6         |
| API Enhancements .....                                                                    | 20        |
| <b>Enhancements to Dynamic Editing a Network Service with VNFs.....</b>                   | <b>23</b> |
| API Enhancements .....                                                                    | 23        |
| <b>Improvements to the Update a Network Service Workflow .....</b>                        | <b>25</b> |
| API Changes.....                                                                          | 25        |
| <b>Support for Adding Dependent VNFDs and CNFDs.....</b>                                  | <b>26</b> |
| Create Dependent VNFs or CNFs in the UI.....                                              | 26        |
| <b>Support for Healing a CNFD based Network Service (Kubernetes VIM) .....</b>            | <b>28</b> |
| Heal a Service Running with CNFs in the UI .....                                          | 28        |
| API Enhancements .....                                                                    | 30        |
| <b>Support for Helm Charts when Instantiating RIFT.ware using Kubernetes as a VIM....</b> | <b>31</b> |
| CaaS Account Updates in the UI.....                                                       | 31        |
| Direct Onboarding of HELM Charts.....                                                     | 34        |
| API Enhancements .....                                                                    | 34        |
| Model Enhancements .....                                                                  | 34        |
| <b>Fixed Issues in RIFT.ware 8.1.1.0 .....</b>                                            | <b>35</b> |
| <b>Known Issues in RIFT.ware 8.1.1.0.....</b>                                             | <b>38</b> |

## RIFT.ware 8.1.1.0 Release Notes

---

This guide describes the RIFT.ware 8.1.1 release, including new features, fixed and known issues, with their workarounds.

## New and Changed Features for 8.1.1

---

RIFT.ware version 8.1.1 introduces enhancements to improve management.

| Feature                                                                           | PFR and JIRAs              |
|-----------------------------------------------------------------------------------|----------------------------|
| Dynamic Editing to a Network Service with CNFs                                    | <a href="#">RIFT-25980</a> |
| Enhancements to Dynamic Editing a Network Service with VNFs                       | <a href="#">RIFT-25980</a> |
| Improvements to the Update a Network Service Workflow                             | <a href="#">RIFT-25980</a> |
| Support for a Heal Attempt when instantiating RIFT.ware using Kubernetes as a VIM | <a href="#">RIFT-28324</a> |
| Support for Helm Charts when Instantiating RIFT.ware using Kubernetes as a VIM    | <a href="#">RIFT-28416</a> |

## Dynamic Editing to a Network Service with CNFs

In release 8.1.1, it is now possible to update a service in the **RUNNING** or **FAILED-TEMP** state with CNFs. An operator can add a new CNF to a running NS and remove a CNF from an NS. VNFs and CNFs are both supported under the same API update-network-service-v2. See [update-network-service-v2](#) for additional API information.

```
https://<launchpad_ip>:8008/api/operations/update-network-service-v2
```

- [UI Enhancements](#)
  - [Insert a CNF into a Running or Failed-Temp Service](#)
  - [Remove a CNF from a Running or Failed-Temp Service](#)
- [API Enhancements](#)

### UI Enhancements

1. On the **Launchpad** menu, click **SERVICES > FULL LIST** to open the list of NS instances.

SERVICES ^

FULL LIST

2. Click  next to the NS that you want to see in more detail. The **VIEWPORT** screen opens.

| SERVICES <span>Created - 0</span> <span>Running - 2</span> <span>Terminated - 0</span> <span>Failed - 0</span> |                                                                                     |                   |        |               |                     |                |         |
|----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------|--------|---------------|---------------------|----------------|---------|
| <div>  </div>               |                                                                                     |                   |        |               |                     |                |         |
|                                                                                                                |                                                                                     | ID                | NAME   | NSD           | STATUS              | INSTANTIATED   | UP TIME |
|                                                                                                                |                                                                                     |                   |        |               |                     |                |         |
|                                                                                                                |  | 8ed4156d-0670-... | Sample | ping_pong_nsd | Running - config... | 34 minutes ago | 34m:44s |

3. Click  to open the Service Update widget and provide the select-update details:

SAMPLE running 6h:29m





VIEWPORT
DATACENTERS
MANAGEMENT INTERFACES
HISTORY
TOPOLOGY
CONFIGURATION


Sample NSR

2 CNFR

NS Data
Configuration
Virtual Links

MONITORING PARAMETERS

|                       |                        |                        |                       |
|-----------------------|------------------------|------------------------|-----------------------|
| ping-request-tx-count | ping-response-tx-count | ping-response-rx-count | ping-request-rx-count |
| 8795                  | 8982                   | 8766                   | 8982                  |

EPA-PARAMS
NO EPA PARAMS CONFIGURED

4. In the **Update Operation Choice** drop-down, either choose **instantiate-cnf** or **remove-cnf**.

Service Update

select-update

Update Operation Choice

instantiate-cnf

instantiate-cnf

remove-cnf

add-vl

remove-vl

CNFD to insert.

Member

Id for CNF.

UPDATE

CLOSE

### Insert a CNF into a Running or Failed-Temp Service

The steps and screenshots in this section are based on the ping pong sample application. This specific workflow is determined by the selected CNFD and NSD associated with the running service.

1. After choosing **instantiate-cnf**, additional fields appear. Provide the select-update details to insert a CNF into a Running or Failed-Temp Service:

**Service Update**

select-update scripts Connections initial-service-p...termini

select-update

**Update Operation Choice**  
instantiate-cnf  
Select service update operation.

**CNFD**  
ping\_cnfd-1.0

**ping\_cnfd**  
ping\_cnfd 1.0  
Ping CNF  
RIFT

CNFD to insert.

**Member**  
ping\_cnfd  
Id for CNF.

[Next >>](#)

UPDATE CLOSE

```
{instantiate-cnf: Object}
instantiate-cnf: Object
member-id-ref: "ping_cnfd"
cnfd-id-ref: "55a7f7ba-0efc-4051-91fa-68509c772740"
```

- **CNFD:** CNFD to insert. The options include the CNFDs from the project's catalog.
- **Member:** ID for the CNF. This is the member-id-ref field.

**Note:** The YAML configuration appears on the right side of the screen for each section.

2. Click **Next >>** to provide the following details in the Scripts step to identify which scripts that you want to add to the service:

## Service Update

The screenshot shows the 'Service Update' dialog box with the 'scripts' tab selected. The 'File Path' field contains 'scripts/init\_config.py'. The 'External Url' section has a 'BROWSE' button and a 'Url' input field. The 'Username' and 'Password' fields are empty. The 'Overwrite' dropdown is set to 'FALSE'. The JSON preview on the right shows the configuration for the script, including the 'file-path' and 'overwrite' settings. The 'UPDATE' and 'CLOSE' buttons are at the bottom.

- **File Path 1:** File including the relative path in the package. After you type the file name, additional fields appear.

**Note:** You must specify the script folder in this field.

- **External URL:** **BROWSE** or type the URL: URL to download. You can either browse or type your URL and add the script to the subfolder that you defined in the File path field.
- **Username:** Username if the url uses authentication.
- **Password:** Password if the url uses authentication.
- **Overwrite:** Overwrite the username and password in this field by choosing False (True or False)

**Note:** If you have additional scripts, then type the path in File Path 2 and the same fields will appear for the subsequent scripts.

3. Click **Next >>** to provide details in the Connections step:

## Service Update

The screenshot shows the 'Service Update' dialog box with the 'Connections' tab selected. The left pane lists VLDs and their connection points. The right pane displays a JSON configuration for the service update.

**Connections Section:**

- Macvlan1**: ping\_cnfd/cp0
- Mgmt VLD**: ping\_cnfd/cp1
- VLD 3**: Select...

**JSON Configuration (Right Pane):**

```
{
  "instantiate-cnfd": {
    "member-id-ref": "ping_cnfd",
    "cnfd-id-ref": "55a7f7ba-0efc-4051-91fa-68509c772740",
    "connection-point": [
      {
        "name": "ping_cnfd/cp1",
        "vld-id-ref": "mgmt_vld"
      },
      {
        "name": "ping_cnfd/cp0",
        "vld-id-ref": "macvlan1"
      }
    ],
    "files-to-upload": [
      {
        "id": 1,
        "file-path": "scripts/init_config.py",
        "overwrite": "false"
      }
    ]
  }
}
```

Buttons at the bottom: UPDATE, CLOSE.

The fields listed in the Connections section are the VLDs in the NSD. If you add connections, then the connections points are linked to a VL.

**Note:** When you choose a connection point in the first drop-down, it will not appear in the second drop-down section.

- Click **Next >>** to provide the following details in the initial-service-primitive step:

**Note:** This step is optional.

## Service Update

select-update scripts Connections initial-service-p... terminate

initial-service-primitive

**Name** 1

Sample

Name of the configuration primitive.

**Seq**

1

Sequence number for the configuration primitive.

**User Defined Script**

start\_traffic.py

Script file to be executed.

**Parameter**

List of parameter settings. Enter as name=value.

**Name** 2

<<Previous Next>>

UPDATE CLOSE

```
{
  "instantiate-cnf": {
    "member-id-ref": "ping_cnf",
    "cnfd-id-ref": "55a7f7ba-0efc-4051-91fa-68509c772740",
    "connection-point": [
      {
        "name": "ping_cnf/cp0",
        "vld-id-ref": "macvlan1"
      },
      {
        "name": "ping_cnf/cp1",
        "vld-id-ref": "mgmt_vld"
      }
    ],
    "initial-service-primitive": [
      {
        "name": "Sample",
        "seq": 1,
        "user-defined-script": "start_traffic.py",
        "files-to-upload": [
          {
            "id": 1,
            "file-path": "scripts/init_config.py",
            "overwrite": "false"
          }
        ]
      }
    ]
  }
}
```

- **Name 1:** Name of the configuration primitive. After you type the configuration name, additional fields appear.
- **Seq:** Sequence number for the configuration primitive.
- **User Defined Script:** Select the script file to be executed from the drop-down menu. The options include:
  - script files that are already included in the NSD.
  - any script files added using this update process (file path scripts)
- **Parameter:** List of parameter settings (param=value).

**Note:** If you have additional initial-service-primitive, then type the name in Name 2 and the same fields will appear for the subsequent initial-service-primitives.

5. Click **Next >>** to provide details in the terminate-service-primitive, add-nf-service-primitive and remove-nf-service-primitive steps. These steps include all same details as the initial-service-primitive step.

**Note:** These steps are optional.

initial-service-p... terminate-servi... add-nf-service... remove-nf-servi... |

6. Click **Next >>** to provide the following details in the Placement step:

## Service Update

initial-service-p... terminate-servi... add-nf-service... remove-nf-servi...

Placement

**Datacenter**

BM4

Datacenter to use when instantiating the VNF.

<<Previous Next>>

UPDATE CLOSE

```
{
  "instantiate-cnfd": {
    "member-id-ref": "ping_cnfd",
    "cnfd-id-ref": "55a7f7ba-0efc-4051-91fa-68509c772740",
    "connection-point": [
      {
        "name": "ping_cnfd/cp0",
        "vld-id-ref": "macvlan1"
      },
      {
        "name": "ping_cnfd/cp1",
        "vld-id-ref": "mgmt_vld"
      }
    ],
    "initial-service-primitive": [
      {
        "name": "Sample",
        "seq": 1,
        "user-defined-script": "start_traffic.py"
      }
    ],
    "terminate-service-primitive": [
      {
        "name": "Sample",
        "seq": 1
      }
    ],
    "add-nf-service-primitive": [
      {
        "name": "Sample",
        "seq": 1
      }
    ],
    "remove-nf-service-primitive": [
      {
        "name": "Sample",
        "seq": 1
      }
    ],
    "files-to-upload": [
      {
        "id": 1,
        "file-path": "scripts/init_config.py",
        "overwrite": false
      }
    ]
  }
}
```

- **Datacenter:** A datacenter select box appears if there is more than one available datacenter to pick in this option. The options in this section are dependent on the configuration of the selected datacenter. You may be able to specify a region, an availability zone and host aggregate meta-data.

**Note:** This step only appears if the CNFD had defined a placement group. The options in this step can also change slightly depending on your selected datacenter.

7. Click **Next >>** to provide the following details in the config-input step:

## Service Update

```

{
  "instantiate-cnf": {
    "member-id-ref": "ping_cnf",
    "cnfd-id-ref": "55a7f7ba-0efc-4051-91fa-68509c772740",
    "connection-point": [
      {
        "name": "ping_cnf/cp0",
        "vld-id-ref": "macvlan1"
      },
      {
        "name": "ping_cnf/cp1",
        "vld-id-ref": "mgmt_vld"
      }
    ],
    "initial-service-primitive": [
      {
        "name": "Sample",
        "seq": 1,
        "user-defined-script": "start_traffic.py"
      }
    ],
    "terminate-service-primitive": [
      {
        "name": "Sample",
        "seq": 1
      }
    ],
    "add-nf-service-primitive": [
      {
        "name": "Sample",
        "seq": 1
      }
    ],
    "remove-nf-service-primitive": [
      {
        "name": "Sample",
        "seq": 1
      }
    ],
    "deployment-profile": {
      "input-parameter": [
        {
          "xpath": "/cnfd[ping_cnf]/name",
          "id": 1,
          "file-path": "scripts/init_config.py",
          "overwrite": "false"
        }
      ]
    }
  }
}

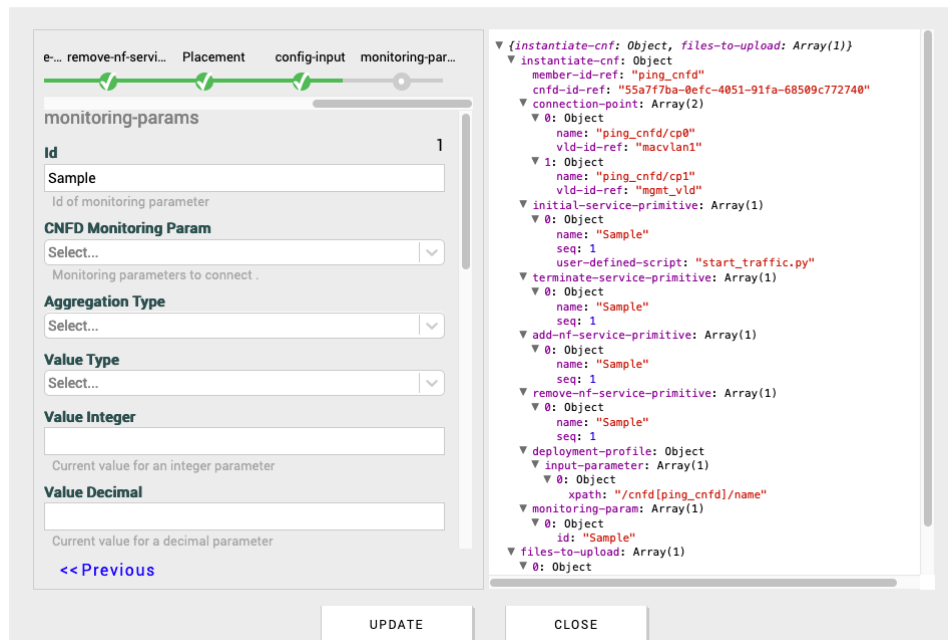
```

- **Xpath 1:** Select the CNFD property to set from the drop-down menu. After you choose the xpath, additional fields appear. The xpath options are dependent on the definition of the CNF that you are adding to your NS. These options are the same as the options available if you configure an input-parameter-xpath in the NSD designer.
- **Value:** Value to set on property.

**Note:** If you have additional xpaths, then choose the xpath in the Xpath 2 drop-down menu and the same fields will appear for the subsequent xpaths.

8. Click **Next >>** to provide the following details in the monitoring-params step:

## Service Update



The dialog shows a progress bar with four steps: 'e... remove-nf-servi...', 'Placement', 'config-input', and 'monitoring-par...'. The 'monitoring-par...' step is active. The left pane contains the 'monitoring-params' form with fields for 'Id' (Sample), 'CNFD Monitoring Param' (Select...), 'Aggregation Type' (Select...), 'Value Type' (Select...), 'Value Integer' (empty), and 'Value Decimal' (empty). The right pane shows a JSON configuration snippet. At the bottom are 'UPDATE' and 'CLOSE' buttons.

monitoring-params

Id 1

Sample

Id of monitoring parameter

CNFD Monitoring Param

Select...

Monitoring parameters to connect .

Aggregation Type

Select...

Value Type

Select...

Value Integer

Current value for an integer parameter

Value Decimal

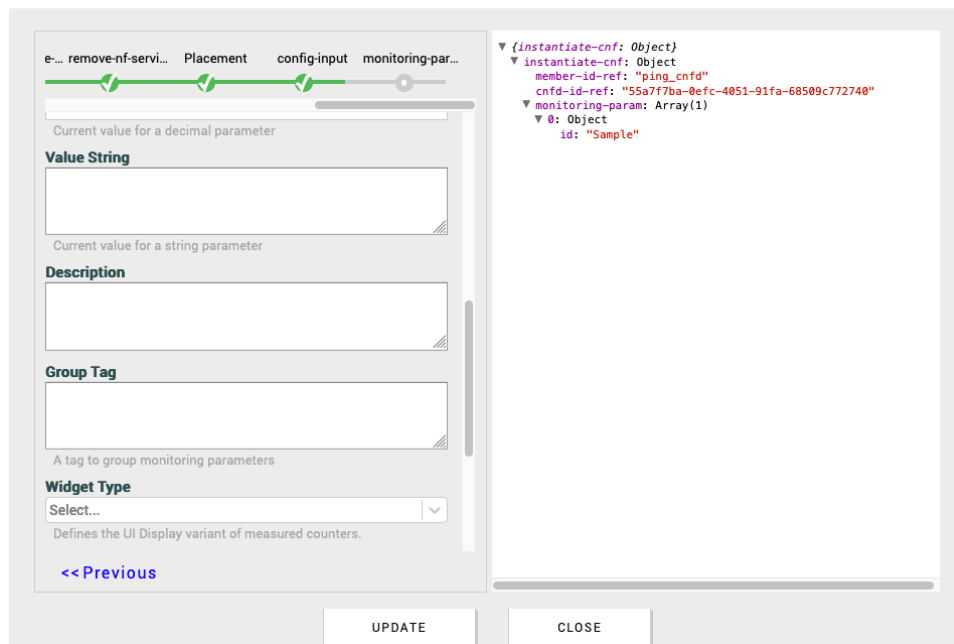
Current value for a decimal parameter

<< Previous

UPDATE CLOSE

```
{
  "instantiate-cnfd": Object,
  "files-to-upload": Array(1)
}
instantiate-cnfd: Object
  member-id-ref: "ping_cnfd"
  cnfd-id-ref: "55a7f7ba-0efc-4051-91fa-68509c772740"
  connection-point: Array(2)
    0: Object
      name: "ping_cnfd/cp0"
      vld-id-ref: "macvlan1"
    1: Object
      name: "ping_cnfd/cp1"
      vld-id-ref: "mgmt_vld"
  initial-service-primitive: Array(1)
    0: Object
      name: "Sample"
      seq: 1
      user-defined-script: "start_traffic.py"
  terminate-service-primitive: Array(1)
    0: Object
      name: "Sample"
      seq: 1
  add-nf-service-primitive: Array(1)
    0: Object
      name: "Sample"
      seq: 1
  remove-nf-service-primitive: Array(1)
    0: Object
      name: "Sample"
      seq: 1
  deployment-profile: Object
    input-parameter: Array(1)
      0: Object
        xpath: "/cnfd[ping_cnfd]/name"
  monitoring-param: Array(1)
    0: Object
      id: "Sample"
  files-to-upload: Array(1)
    0: Object
```

## Service Update



The dialog shows a progress bar with four steps: 'e... remove-nf-servi...', 'Placement', 'config-input', and 'monitoring-par...'. The 'monitoring-par...' step is active. The left pane contains the 'monitoring-param' form with fields for 'Value String' (empty), 'Description' (empty), 'Group Tag' (empty), and 'Widget Type' (Select...). The right pane shows a JSON configuration snippet. At the bottom are 'UPDATE' and 'CLOSE' buttons.

Value String

Current value for a decimal parameter

Description

Group Tag

A tag to group monitoring parameters

Widget Type

Select...

Defines the UI Display variant of measured counters.

<< Previous

UPDATE CLOSE

```
{
  "instantiate-cnfd": Object
}
instantiate-cnfd: Object
  member-id-ref: "ping_cnfd"
  cnfd-id-ref: "55a7f7ba-0efc-4051-91fa-68509c772740"
  monitoring-param: Array(1)
    0: Object
      id: "Sample"
```

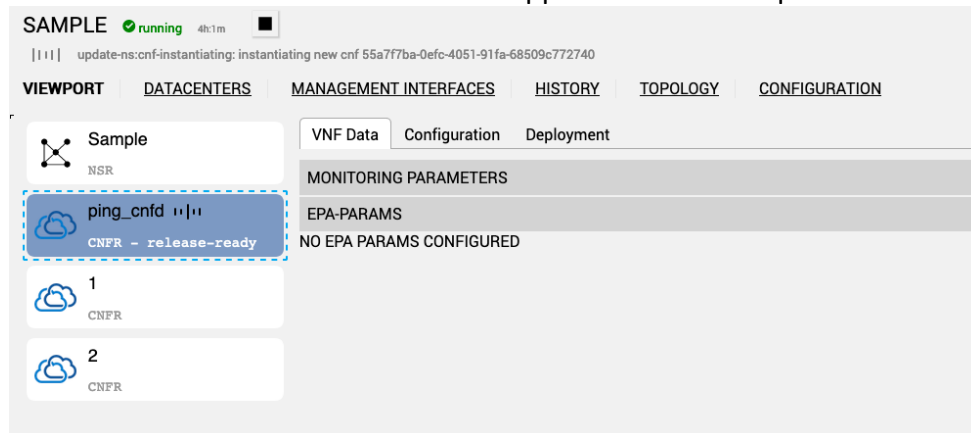
## Service Update

- **Id 1:** Id of monitoring parameter. After you type the ID, additional fields appear.
- **CNFD Monitoring Param:** Select the CNFD monitoring parameters to connect from the drop-down menu.
- **Aggregation Type:** Select the aggregation type from the drop-down menu. The options include:
  - AVERAGE
  - MINIMUM
  - MAXIMM
  - COUNT
  - SUM
- **Value Type:** Select the value type from the drop-down menu. The options include:
  - INT
  - DECIMAL
  - STRING
- **Value Integer:** Current value for an integer parameter.
- **Value Decimal:** Current value for a decimal parameter.
- **Value String:** Current value for a string parameter.
- **Description:** Type the description for the monitoring parameter.
- **Group Tag:** A tag to group monitoring parameters.
- **Widget Type:** Select the widget type to define the UI display variant of measured counters. The options include:
  - COUNTER
  - GAUGE
  - TEXTBOX

- SLIDER
- HISTOGRAM
- BAR
- **Units:** Measured Counter Units (e.g., Packets, Kbps, Mbps, etc.)

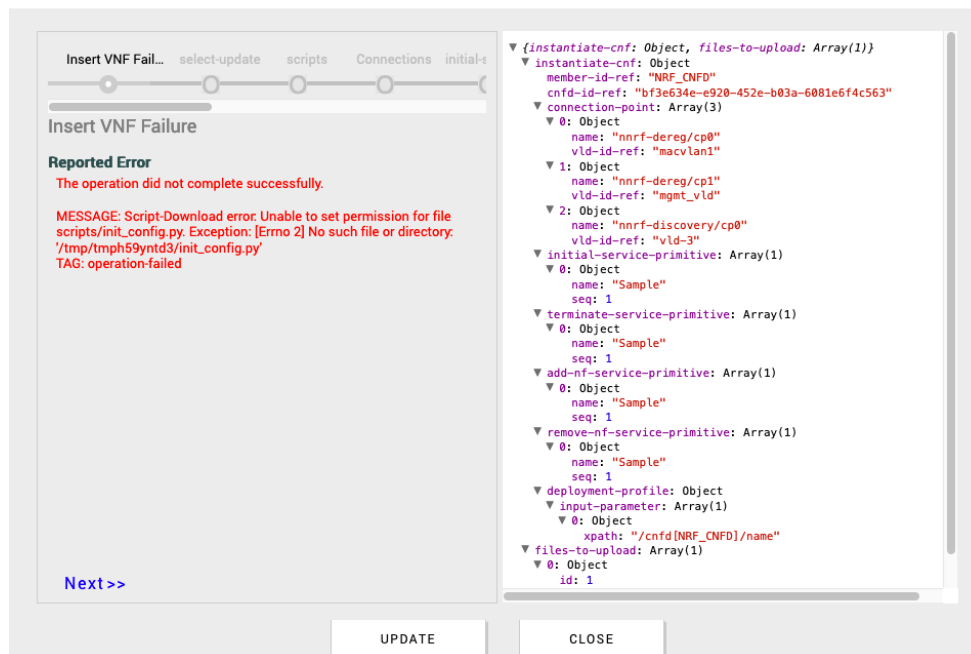
**Note:** If you have additional monitoring parameter IDs, then type the ID in the ID 2 field and the same fields will appear for the subsequent monitoring parameters.

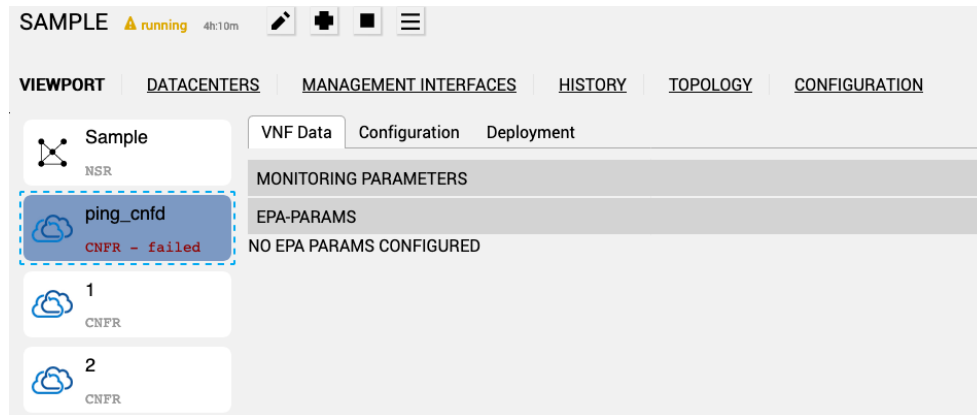
9. Click **UPDATE**. The Service → Viewport shows the progress of the heal operation under the service name. The new CNF appears on the Viewport screen.



If the **instantiate-cnf** attempt fails, then an Insert Failure error message appears in the Service Update widget. The CNF will also indicate that the attempt failed on the Viewport screen.

#### Service Update

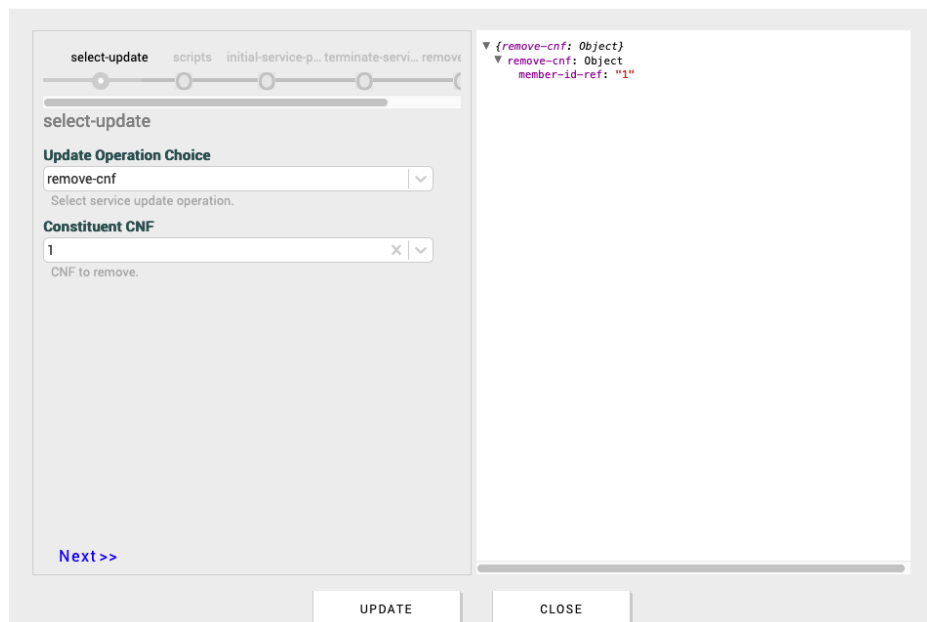




### Remove a CNF from a Running or Failed-Temp Service

1. Choose **remove-cnfd** and then click on which **Constituent CNF** to remove. The constituent CNF options include the CNFs that are currently part of the running Service.

#### Service Update



2. Click **Next >>** to provide the following details in the Scripts step to identify which scripts that you want to remove from the service:

## Service Update

select-update scripts initial-service-p... terminate-servi... remove

scripts

| File Path              | 1 |
|------------------------|---|
| scripts/init_config.py |   |

Package asset destination, include relative path and file name

External Url

BROWSE

Url

Url to download

Username

username if the url uses authentication

Password

password if the url uses authentication

Overwrite

FALSE

<<Previous Next>>

UPDATE CLOSE

```

{remove-cnf: Object, files-to-upload: Array(1)}
  remove-cnf: Object
    member-id-ref: "1"
  files-to-upload: Array(1)
    0: Object
      id: 1
      file-path: "scripts/init_config.py"
      overwrite: "false"
  
```

- **File Path 1:** File including the relative path in the package. After you type the file name, additional fields appear.

**Note:** You must specify the script folder in this field.

- **External URL:** **BROWSE** or type the URL: URL to download. You can either browse or type your URL and add the script to the subfolder that you defined in the File path field.
- **Username:** Username if the url uses authentication.
- **Password:** Password if the url uses authentication.
- **Overwrite:** Overwrite the username and password in this field by choosing False (True or False)

**Note:** If you have additional scripts, then type the path in File Path 2 and the same fields will appear for the subsequent scripts.

3. Click **Next >>** to provide the following details in the initial-service-primitive step:

**Note:** This step is optional.

## Service Update

```

{remove-cnf: Object, files-to-upload: Array(1)}
  remove-cnf: Object
    member-id-ref: "1"
  initial-service-primitive: Array(1)
    0: Object
      name: "Sample"
      seq: 1
      files-to-upload: Array(1)
        0: Object
          id: 1
          file-path: "scripts/init_config.py"
          overwrite: "false"
  
```

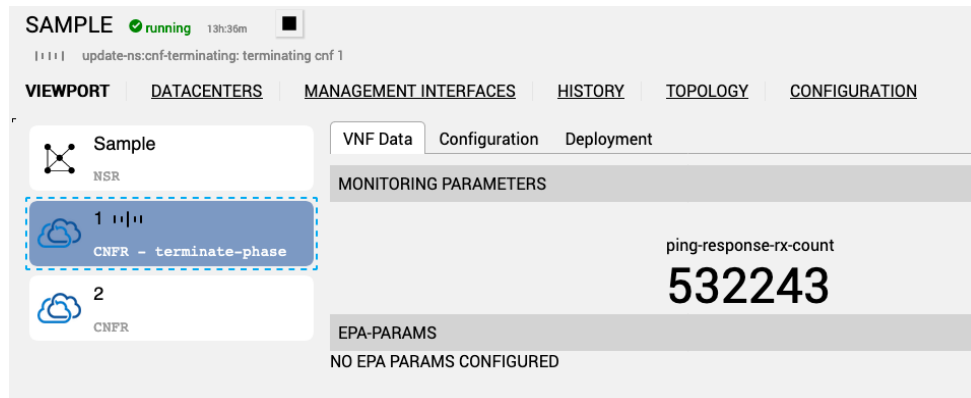
- **Name 1:** Name of the configuration primitive. After you type the configuration name, additional fields appear.
- **Seq:** Sequence number for the configuration primitive.
- **User Defined Script:** Select the script file to be executed.
- **Parameter:** List of parameter settings (param=value).

**Note:** If you have additional initial-service-primitive, then type the name in Name 2 and the same fields will appear for the subsequent initial-service-primitives.

4. Click **Next >>** to provide the details in the terminate-service-primitive and remove-nf-service-primitive steps. These steps include all same details as the initial-service-primitive step.

**Note:** These steps are optional.

5. Click **UPDATE**. The Service → Viewport shows the progress of the update operation under the service name. The CNF no longer appears on the Viewport screen.



If the **remove-cnf** attempt fails, then an Insert Failure error message appears in the Service Update widget. The CNF will also indicate that the attempt failed on the Viewport screen.

## API Enhancements

The payload examples for adding or removing a CNF to a running service are different than the payload examples for adding or removing a VNF.

### Add a CNF to a running NS Payload Example

```
{
  "input": {
    "project-name": "default",
    "nsr-id-ref": "f0eee244-eb4d-4711-921f-a0fdafa90b76",
    "ns-update-type": "instantiate-cnf",
    "files-to-upload": [
      {
        "id": 1,
        "external-url":
          "http://tau.eng.riftio.com/~username/cnfd/start_traffic_member-id-ping_cnfd-3.py",
        "file-path": "scripts/start_traffic_member-id-ping_cnfd-3.py"
      },
      {
        "id": 2,
        "external-url":
          "http://tau.eng.riftio.com/~username/cnfd/test.py",
        "file-path": "scripts/test.py"
      }
    ],
    "instantiate-cnf": {
      "member-id-ref": "3",
      "cnfd-id-ref": "55a7f7ba-0efc-4051-91fa-68509c772740",
      "connection-point": [
        {
          "name": "ping_cnfd/cp1",
          "vld-id-ref": "macvlan1"
        },
        {
          "name": "ping_cnfd/cp0",
          "vld-id-ref": "mgmt_vld"
        }
      ]
    }
  }
}
```

```

    ],
    "add-nf-service-primitive": [
      {
        "name": "enabling-ping-member-id-3",
        "seq": 1,
        "user-defined-script": "start_traffic_member-id-
          ping_cnfd-3.py",
        "parameter": [
          {
            "name": "param1",
            "value": "value1"
          },
          {
            "name": "param2",
            "value": "value2"
          }
        ]
      }
    ],
    "deployment-profile": {
      "datacenter": "bm3",
      "input-parameter": [
        {
          "xpath": "/cnfd[3]/config-data/values-file-
url",
          "value":
"http://tau.eng.riftio.com/~username/cnfd/ping-values.yaml"
        }
      ]
    }
  }
}

```

### Remove a CNF to a running NS Payload Example

```

{
  "input": {
    "project-name": "default",
    "nsr-id-ref": "f0eee244-eb4d-4711-921f-a0fdafa90b76",
    "ns-update-type": "remove-cnfd",
    "files-to-upload": [
      {
        "id": 1,
        "external-url":
"http://tau.eng.riftio.com/~username/cnfd/test.py",
        "file-path": "scripts/test.py"
      }
    ],
    "remove-cnfd": {
      "member-id-ref": "3",
      "remove-nf-service-primitive": [
        {
          "name": "remove-cnfd-primitive-sample",
          "seq": 1,
          "user-defined-script": "test.py",
          "parameter": [

```

```
{
  {
    "name": "param1",
    "value": "value1"
  }
}
```

## Enhancements to Dynamic Editing a Network Service with VNFs

### API Enhancements

The following changes have been made to the nsr:update-network-service-v2 in the NSD Data model (nsd:nsr).

```
https://<launchpad_ip>:8008/api/operations/update-network-service-v2
```

- ns-update-type remove-vnfd **changed to** ns-update-type remove-vnf.
- vnfd-id-ref **field for** remove-vnf **is deprecated**.
- add-vnf-service-primitive **changed to** add-nf-service-primitive.
- remove-vnf-service-primitive **changed to** remove-nf-service-primitive.
- deployment-profile **moved inside the** instantiate-vnf **container**.

The payload examples for adding or removing a VNF to a running service changed in release 8.1.1.

### Add a VNF to a Network Service Updated Payload Example

```
{
  "input": {
    "project-name": "default",
    "nsr-id-ref": "f40d2670-1e18-4fa6-a41b-9eae28f3760c",
    "ns-update-type": "instantiate-vnf",
    "instantiate-vnf": {
      "member-vnf-index-ref": "7",
      "vnfd-id-ref": "eantc_testvnffw_vnfd",
      "connection-point": [
        {
          "name": "client",
          "vld-id-ref": "client"
        },
        {
          "name": "server",
          "vld-id-ref": "SERVER"
        },
        {
          "name": "management",
          "vld-id-ref": "management"
        }
      ],
      "deployment-profile": {
        "datacenter": "ES2",
        "placement-group-maps": [
          {
            "placement-group-ref": "pg-ovs",
            "cloud-type": "openstack",
            "openstack-input": {
              "zone": "nova"
            }
          }
        ]
      }
    }
  }
}
```

```
    }  
  }  
}
```

**Remove a VNF to a Network Service Updated Payload Example**

```
{  
  "input": {  
    "project-name": "default",  
    "nsr-id-ref": "4819b83d-7612-4142-be31-0d7d9968b229",  
    "ns-update-type": "remove-vnf",  
    "remove-vnf": {  
      "member-vnf-index-ref": "3"  
    }  
  }  
}
```

## Improvements to the Update a Network Service Workflow

### API Changes

The following changes have been made to the `nsr:update-network-service-v2` in the NSD Data model (`nsd:nsr`) in release 8.1.1.

```
https://<launchpad_ip>:8008/api/operations/update-network-service-v2
```

- An operator can update a Network Service when the NS is either in the **RUNNING** or **FAILED-TEMP** state.
- An operator can provide `add-nf-service-primitive` primitives to a service and they are executed once the VNF or CNF is brought up successfully. See [Insert a CNF into a Running or Failed-Temp Service](#).
- An operator can provide `remove-nf-service-primitive` primitives and they are executed before initiating VNF or CNF termination. See [Remove a CNF into a Running or Failed-Temp Service](#).
- An operator can use the `deployment-file` or `input-parameter` fields to override values in a VNFD or CNFD when adding them to an NS. For example, a user can update an image in the `vdu:image` for VNFDs or a user can provide a values file for CNFDs.
- The `files-to-upload` field includes the user script details that need to be downloaded and copied to the RVR which can later be referenced through the following primitives:
  - `initial-service-primitive`
  - `terminate-service-primitive`
  - `add-nf-service-primitive`
  - `remove-nf-service-primitive`

See [update-network-service-v2](#) for additional API information.

### Response for update-network-service-v2 API

The response for the `update-network-service-v2` API is the same for the following operations: `instantiate-cnf`, `instantiate-vnf`, `remove-cnf` and `remove-vnf`.

**Note:** There is no change in the response for `instantiate-vnf` or `remove-vnf`.

```
{
  "output": {
    "transaction-status": "init",
    "status": "success",
    "transaction-id": "9ad6dc8d-2135-42ef-86b9-b97c45409520"
  }
}
```

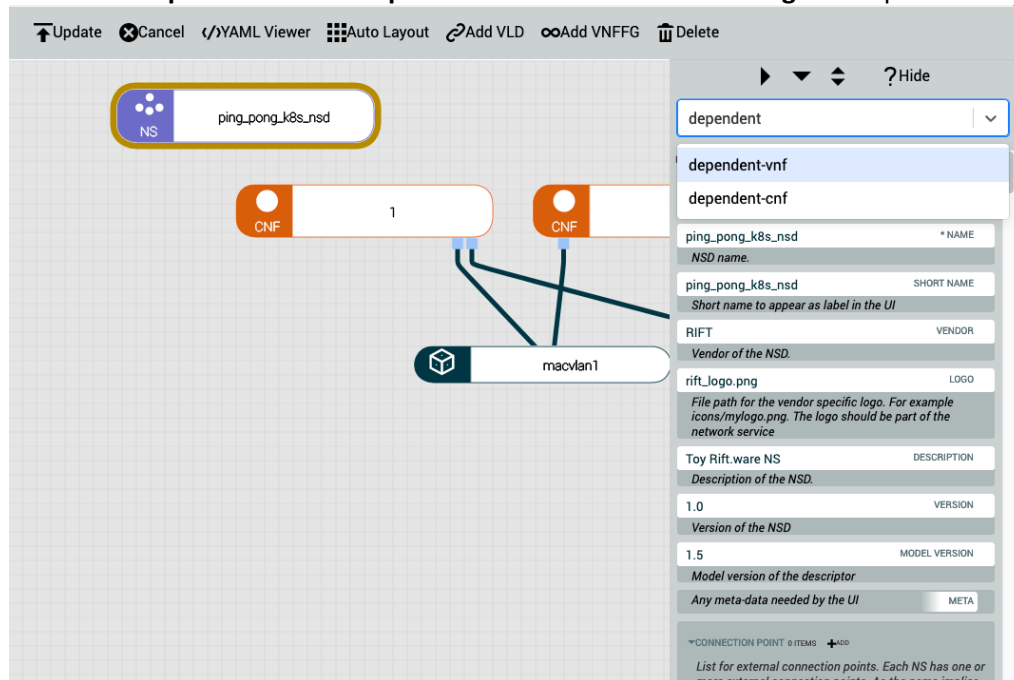
## Support for Adding Dependent VNFDs and CNFDs

In release 8.1.1, it is possible to specify the dependency flow of VNFs and CNFs in the NSD descriptor. An operator can create a VNF or CNF that is dependent on another VNF or CNF in the NSD. While instantiating the NSD, the dependent VNF or CNF is created prior to creating the original VNF or CNF.

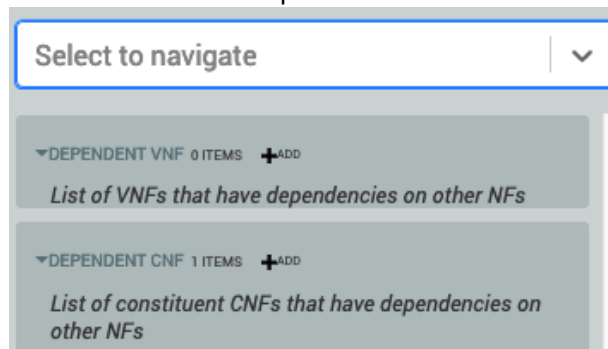
For example, an operator creates VNF-A that is dependent on VNF-B. During the instantiation process, VNF-B is created prior to VNF-A.

### Create Dependent VNFs or CNFs in the UI

1. On the **Launchpad** menu, click **SERVICES** and select the appropriate network service from the catalog and click [🔗](#).
2. Search for **dependent-vnf** or **dependent-cnf** in the **Select to navigate** drop down menu.



3. Click **+ADD** to add a Dependent VNF or CNF.



- **Dependent VNF:** Select the **MEMBER VNF INDEX REF** from the drop-down menu.
- **Dependent CNF:** Select the **MEMBER ID REF** from the drop-down menu.

▼DEPENDENT VNF 1  REMOVE  COPY

**Member Vnf Index Ref** \* MEMBER VNF INDEX REF ▼

*Reference to vnfd within constituent-vnfd list*

▼CNF DEPENDENCY 1 ITEMS  ADD

*List of CNF dependencies.*

▼CNF DEPENDENCY 1  REMOVE  COPY

**Member Id Ref** \* MEMBER ID REF ▼

*Dependee member CNF index*

▼VNF DEPENDENCY 0 ITEMS  ADD

*List of VNF dependencies.*

▼DEPENDENT CNF 1 ITEMS  ADD

*List of constituent CNFs that have dependencies on other NFs*

▼DEPENDENT CNF 1  REMOVE  COPY

**1** \* MEMBER ID REF ▼

*Reference to cnfd within constituent-cnfd list*

▼CNF DEPENDENCY 0 ITEMS  ADD

*List of CNF dependencies.*

▼VNF DEPENDENCY 0 ITEMS  ADD

*List of VNF dependencies.*

4. Repeat step 3 to add multiple dependencies.
5. Click  Update.

## Support for Healing a CNFD based Network Service (Kubernetes VIM)

In release 8.1.1, an operator can restart or reconfigure Network Services based on CNF (using Kubernetes as VIM) using the heal feature in Launchpad. It is possible to heal a CNFD by attempting to update the Kubernetes release using values data or recovering from external failures.

- [Heal a Service Running with CNFs in the UI](#)
- [API Enhancements](#)

### Heal a Service Running with CNFs in the UI

There is now a **Heal** button in the **SERVICES** section in the UI for a running service that uses Kubernetes as a VIM.

1. On the **Launchpad** menu, click **SERVICES**.
2. Click the  button next to a specific service to start the heal process.

If you open a running NS, then click the  button at the top of the screen. The **Heal Service** screen appears.

**Heal Service**

SELECT OPERATION

Restart

SELECT OPERATION TYPE

Update Release

SELECT CNFS

CNFR LIST

VALUES FILE

BROWSE

Url

USERNAME

PASSWORD

HEAL

CANCEL

3. Choose **Restart** or **Reconfigure** in the **SELECT OPERATION** drop down menu to attempt the heal.
  - **Restart:** This heal option attempts to restart or update a container on a running NS.
  - **Reconfigure:** This heal option attempts to reconfigure the container. It does not restart the container. This heal option means that only pre-heal primitives/post-heal primitives are run on the service.

4. Choose the operation type in the **SELECT OPERATION TYPE** field.
  - **Update Release:** This option updates the release using the values file with different values.
  - **Reinstall Release:** This option uninstalls the current release and installs it again. This attempt assumes that there are things to be cleaned up on the Kubernetes cluster prior to installing the release again.

**Note:** Reinstall Release is only available when you choose the Restart option in step 3.

5. Click on **CNFR LIST** in the **SELECT CNFS** field and the possible cnfs appear. Choose the appropriate cnfs.
6. If you want to add a new values files while attempt to heal a service using the restart operation, then you can upload it in the **VALUE FILES** field. The values file allows an operator to update environment variables and configuration information. After uploading the file and attempting the heal, you'll restart a cnf with a new set of parameters.

**Note:** Reinstall Release is only available when you choose the Restart option in step 3.

7. If the value file requires authentication, then you can add the **USERNAME** and **PASSWORD** in the appropriate fields.

**Note:** Reinstall Release is only available when you choose the Restart option in step 3.

8. Click **HEAL**.
9. During the heal attempt process, the Service → Viewport shows the progress of the heal operation under the service name.

**Note:** If the CNF heal attempt fails, then it is possible to recover the instance by attempting to retry and rollback at the NSR level for all NS operations. See the [Create and Instantiate a Network Service](#) section.

The progress of the CNF Heal will no longer appear once the NS instantiation is successful.

After performing a heal action, the UI now shows the status as Success or Failure in the **HISTORY** tab. See the [Notifications](#) and [Event Viewer](#) sections for more details about a specific event.

See [NS and VNF Heal Support](#) for more information on a heal operation for CNFDs.

## API Enhancements

It is now possible to use API calls to heal a network service running with CNFs.

```
https://<launchpad_ip>:8008/api/operations/heal-network-service-v2
```

An operator can use the heal-netowrk-service-v2 API to heal a network service that is running with CNFs. When attempting to heal a CNF service, the payload example and response for this API is different from when healing a VNF service.

See [nsr:heal-network-service-v2](#) API.

The following is the payload example for Healing a Service Running with CNFs.

```
{
  "input": {
    "project-name": "default",
    "nsr-id-ref": "b3ccf457-1b91-4aef-acd4-c128efc7f511",
    "cnfr": { "id-ref": "9479c638-46e9-42e8-9a14-
      "id-ref": "35055892-8b0d-40f6-8c0e-f346fbcfc664",
      "operation": "restart",
      "op-type": "update-release",
      "values-file": "http://tau.eng.riftio.com/user/ping-
        values.yaml"
    }
  }
}
```

After calling this API with the correct parameters, you should receive a response in the following format.

```
{
  "output": {
    "transaction-status": "init",
    "status": "success",
    "transaction-id": "cc88843a-35eb-457b-a4e9-6b1668bd7a8c"
  }
}
```

**Note:** An operator can use the **transaction-id** in the `async-transaction-command` in the `nsr.yang` model to check the status of the heal operation.

## Support for Helm Charts when Instantiating RIFT.ware using Kubernetes as a VIM

Helm is a Kubernetes application and package manager. RIFT.ware uses helm charts to power CNF deployments to provide Kubernetes as a VIM. In this release, RIFT.ware now supports Helm version 3. The Helm version 3 no longer requires a tiller server to be installed on the Kubernetes cluster. Also, RIFT.ware now validates helm charts prior to CNFD generation.

For Helm version 3 updates, see <https://v3.helm.sh/docs/faq/#changes-since-helm-2>.

See [Kubernetes Documentation](#) for additional information.

- [CaaS Account Updates in the UI](#)
- [Direct Onboarding of Helm Charts \(Both v2 and v3\)](#)
- [API Enhancements](#)
- [Model Enhancements](#)

### CaaS Account Updates in the UI

There is a change to the fields required to fill out when an operator creates a CaaS Account. This change is applicable to the API and the UI. When creating an account, steps 1-4 are the same as the previous release. In step 5, it is no longer necessary to include the **TILLER IP** or **TILLER PORT**. The payload for CaaS account creation only includes the KUBE CONFIG.

1. Sign-in to the Launchpad Dashboard menu and click **CONNECTORS**.
2. Under **ACCOUNT TYPE**, choose **CaaS** and click **CREATE ACCOUNT**.
3. Add the unique **CaaS Account Name\*** name for the existing CaaS account you are linking to.
4. Type the **CNF INSTANCE TIMEOUT (DEFAULT 300 SECONDS)**. Maximum time allocated for resource instantiation in the CaaS account. Default is 300.
5. Under **SELECT ACCOUNT TYPE**, click Kubernetes.
6. Provide the **KUBE CONFIG** information.

#### [Organizing Cluster Access Using kubeconfig Files](#)

The **KUBE CONFIG** is the Kubernetes configuration file in YAML format. This file must be provided by the operator.

The **TILLER IP** and **TILLER PORT** fields are deprecated from the CaaS Account creation process in 8.1.1.

ACCOUNT

NAME \*

CNF INSTANCE TIMEOUT (SECONDS)

SELECT ACCOUNT TYPE


Kubernetes


Mock  
Kubernetes


ROBIN  
Robin

Account Details

KUBE CONFIG \*

CANCEL

SAVE

### Sample Kube Config

apiVersion: v1 clusters: - cluster: certificate-authority-data:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0
ICQURBTKJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRX
SmwKY201bGRHVnpNQjRYRFRJd01ESXhNREUyTVRRd05Wb1hEVE13TURJd056
RTJNVFF3TIZvd0ZURVRNqkVHQTFRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0F
TSXdEUVlIKS29aSWWh2Y05BUUVCQlFBRGdnRVBIBRENDQVFvQ2dnRUJBT3lx
CjN
yOHlZVmlRL01Ua043UEJ3Y3FBMmVFQjF6T2UveVpuSExaSjlsVWhtdmN3YWtrO
HNmY05NelhqQVpZOGVsMGcKWUdRVThBRXRubHhEbTduOGZlQlRBTmVtbGsy
bEplTW1HQVlGVGxrNkxsWldjZ0tQcS9sMWNweFNQaDkzclVscApXNnBtTXgvQl
M0aTFYOE1XV2NQY2h1c0MwSVE0MlpOYk9kMko4cDRsRk9QZnZpd1JlZG5OUjk
zZGY1eXVJQjcwCnRhekdzYTRSOUg5djdRSzQ0Y2hvaFRUck4vQVg4WXIrWetVRS
81dFFPNk9OSk1TUmf3MWI1WERWK215SXgrS0IKaEJzdk5uZ2dTL2UwbXd5Ny9E
TzNVcFNkZlBTGRKYzRucmlTcm1IM2kvU1hNVy9jTVdmV2VXNW1ZbFlhMGRsT
Qp6QzJvKzEwcTRiQ21PUy91ZEZzQ0F3RUFBU1qTUNFd0RnWURWUjBQOVFIL
0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0RRWUpLb1pJaHZjT
kFRRUxCUUFEZ2dFQkFERVBVc0tweWRlOHNjb0hyWjdLdi9SRmVnQmwKZVI0N
VprYkw2cDJsV2dWaUdCbUsxYmEycnhFbko3S1pxQU1kUE4ybJ2Y1Q1dTBSRTRY
STNjUGNobE9SYVJrWQpPbVhYdTmZRTczYzlwVXdkKcHZDTXQ0NzVPSVlSRHR
0bXFpSVYzZzkvMUEzR2crMlZFdENZdVp5VHFrNGtOa1NpCnJLcGJYWEYzcmE1c
VExwINuVEZCb2VrbXdmeGNvWGtXVmVzQjNTRjFoZmlFNGlCbGVsTU4rdFAwe
kVuYmxYdkEKZXA0N3lPZWZkSnpNQnB4L2s3OGZiNHJVZGxXbktTWXdJdERTV
WpxNFVxTEJtTmJmaHl3MXVhVXdJb0ZsL2YyZQpIa0lGM2FaR3p3YThqVTV2TFN
```

MUTNlcytiUFJXSfJiNTB4bGg5MjFWSEINQUphaEpZNmVwK1F0TExiST0KLS0tLS  
1FTkQgQ0VSVElGSUNBVEUtlS0tLQo= server: https://10.66.4.35:6443 name:  
cluster.local contexts: - context: cluster: cluster.local user: kubernetes-admin name:  
kubernetes-admin@cluster.local current-context: kubernetes-admin@cluster.local kind:  
Config preferences: {} users: - name: kubernetes-admin user: client-certificate-data:  
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUM4akNDQWRxZ0F3SUJBZ0I  
JSUk2dEo1THk5YXN3RFFZSkvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVU  
KQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TURBeU1UQXhOakUwTURWYUZ3  
MHINVEF5TURReE5qRTBNRFphTURReApGekFWQmdOVkJBb1REbk41YzNSbGJU  
cHRZWE4wWlhKek1Sa3dGd1IEVlFRREV4QnJkV0psY201bGRHVnpMV0ZrCmJXb  
HVNSUICSWpBTkNa3Foa2IHOXcwQkFRRUZBQU9DQVE4QU1JSUJDZ0tDQVFF  
QXlCRUc2Vk8wOWd6ZnVBL1UKeWpYUXU4aE5sRTdFL0cyK2F5TXg0WmtpYUJo  
U3NIRkVqWVMrU0ljZkFqaG1FbWVRTIFUYmI5emFDcmNXWTFTWgpkWndkZGR  
tcWpSTnJpSzdmYlduWXJQRlZPRmc4aWIBN3ZzOWZEKzRLUGNyMINBUkRjUE9t  
N3NxSDBYU9i0aGJ3CIU1T0xaVmx1bnNaamRtZ3ZjdkxobGRWeUdIQjZObE54UWIO  
NTB2VTEyaU1vWi9UdURGRUs4U1A3YnVUakF1VUYKMkRGYzhVbUZRRWIWZz  
h4cVI6N2E4bWhtQzFZZ2JEa2JoMmRjcnBIMzhTRk9obU5yRFQvbmVYNnFST2Z5Vj  
dkQQp1NVZGdV3OVZuQW5vQTVrQXdwZjdnWGNJVXhEbHhhalY1MXNZMWU  
zT3dqWUMvOU5kQzk1NWZ0QS9zL2JSZG1XCjd4cElxd0lEQVFBQm95Y3dKVEFP  
QmdOVkhROEJBZjhFQkFNQ0JhQXdFd1lEVlIwbEJBd3dDZ1lJS3dZQkRJVUgKQXd  
Jd0RRWUpLb1pJaHZjTkFRRUxCUUFEZ2dFQkFGWngvRWtuMm1kZERlN2xtR1B  
DL0tvcGp3UWN5RGRSK0JrQQpaZU9wamFWNkdzVUVPUEVJcjZsU3d6U0JGK29B  
Ky9TT0d5SFJGMGlyR3EzM1J5d1JCUDdLNTZqd3pEdFhPQzRkCnQwZUtBM2IDS  
W4vT2FMSithNERnRGZ1RExDQTZsNUl0MzFSOWFITk9QVi9LumpYSGNLc2dXSjdo  
NXQ3YzdrUkgKSEpRaytaWF11NTZlcXYyU09WQ0I3NUFMV1VsOFIEcHAxWnFvO  
G9IZHVLcnkxRmgvNlJOWjM2a1hFSgglV1hoeQpod3VFafZ1c1d5d2dTR25mMjdmS  
VJKTm9Oc2lmdFJHQ1cwWDdLMk5sc0tlY1lld1JuLzl0VWU3Q1d3ZkprNjR2CkFmW  
VpPbkFXcVlsRU52c2Q4WjRrbE1XUWptUXdJaHRQQTVCUVFrY1E4TnlRQS8wTW  
VaTT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtlS0tLQo= client-key-data:  
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSU1Fb3dJQkFBS0NBUEV  
BeUJFRzZWTzA5Z3pmdUEvVXlqWFF1OGhObEU3RS9HMiheU14NFpraWFCaFNz  
ZUZFCmpZUytTSWNmQWpobUVtZVFOUVRiYjl6YUNyY1dZMVNaZFP3ZGRkbX  
FqUk5yaUs3ZmJXbllYUEZWT0ZnOGlpQTcKdnM5ZkQrNEtQY3IyU0FSRGNQT203c  
3FIMFhSL3RoYndVNU9MWlZsdW5zWmpkbWd2Y3ZMaGxkVnlHSEI2TmxOeApRY  
jQ1MHZVMTJpTW9aL1R1REZFSzhSUDdidVRqQXVVRjJERmM4VW1GUUVpVmc  
4eHFZeJdhOG1obUMxWWdiRGtiCmgyZGNycEgzOFNGT2htTnJEVC9uZVg2cVJPZn  
lWN2RBdTVWRnVZNzlwbkFub0E1a0F3cGY3Z1hjSVV4RGx4YWoKVjUxc1kxZTN  
Pd2pZQy85TmRDOTU1ZnRBL3MvYlJkbVc3eHBjCXdJREFRQUJBb0lCQUQzbnk5OS  
S9Pa0RmbnVGZQoxS0owb0U5YUhhOY0V3R0t5dWphQVAXRGtod2JhYjh2bVBJWgt  
DdFI2S3BnUFIxWHVnV3BUkhlS1NGVUViWUx5CmpFRWR1dWZhSDdGakR1VE  
41d2Y5dGpOY3dOZWNSYThmWmgwV0lQM1cwNGhuSW1rS0dLdjhWK3lPOGhO  
WERxVVMKMSsya08zOXllbzNhS0xCY2NFSUs4TTNocTFudjlxN2VkN1p2aEFkc1IB  
S1FNeUZyU2ZzSWZUM0liUzcrN200Vwo0WjF6d3diLzJmWjQ2WDJCQllNbmRxUFp  
FeElhT0o1YkdvdWl3d1pyZWlYUINCslk1UVNLTGI4clNiRW9pOUdVCjhLbzZmd3B  
MZVM1K2VCMTRKam5BS3MxbG1PdW1UUVoxZ05RSXdqT3IrVGMymnN1dWZD  
UGJMQmpsRnE3RHJHYnkKSi9WRmtRRUNnWUVBKzA5Mk9udlY5TDloUjdualhN

HpZRk1wMzU0S1MzWjBLcGsvQVJVNHVCa3hZWUNUb3JGdgpGNHkyRjI2TXRQ  
 TzlPU3F1MDdlbHh4SGF1NUFhU0NHVGlkdIjleFFsa2ZTQW9CUFNHTFhYZHhwc05  
 ZakNYSWErCitJOG9aaTFiQ054SXhtY2dYdXNNemp0UEp6d2RjQ2g4UHB4dlVMTS  
 9OT080UGZpeU1GYTg0V3NDZ1lFQXk4ekUKYi9OaisvZjV2ZEFPTjEzTE5ORTThFb  
 XpscFJIM05EaVpLNnFKdm56TzFGeWVXZUIRSD11bHdKSFRFZnV1UjdtAp1QUUp  
 KeTlyZXRqMWIESldocUJ5MlIxQXpsdHRMcVv3S2Z4M3FWSIZCeXZ3M25zTW9Q  
 TEZWZTZPZ04zZWZYamp4CnA5eFNod1hZRXJobDZmMkFsZmlvcDVnUnVQemow  
 RjlUZjluM3hjRUNnWUFTYzhjdWpDRXRrMC9GUUhUZG16alMKZmNpNVRwVmE  
 3UnVpZfJZWE1rT3o0TVdyYnhGb0JMY3ZkM2wzUnZxMTNwK3FMdVfMvDRDd2  
 1UTUNTcFpqYlhPegp6NXdWK1dpNHlzY1crZDJYU3VMRE1BRjUvTXlVbG93M0cr  
 dlBkTTBxWFhaS2V3LzVhRERNZzdaUUh5M2FLbDdWcKnnOXIVWGIybnpITCt4S  
 zZVZVFVZVFLQmdFR2lGbmSVnlu2VJZUtvNmx0Q1c1ZkhBdUxDb0lZd1JZT1RE  
 WGwKM0NwK28xVjg1bnBuOVcxdEhYcGcvQ3JFZFNJZ1NEUUVIS09ORUJuOHRz  
 RDZ4MjhPb0lXalZObjJrZTFaMGQRUwpVTThncW1tbXhIcmF1dkNnNDdqSHYyZGR  
 uUE9KUHpvaUdHbm5sZWloZmlEWFJUd0hNcm1XbmM2SGt3NmVSSlBlCkxJK0JB  
 b0dCQVBCL05qaGVJK1REWVFLbmXpb1JldmllQWNLTs9id1d5Y3NHQ3J2Nnl2N2  
 p6Y2phNnZ5REJkdFYKM0c0cGtpUFdDdGNxUm9xZ0tIY3FON1FoWkdaS1hrSjJldy9  
 zYWVCNzFuYzBFQnJod0kwb1NSOXdvVEphVXhkNngpzajR4MVdvVUpLVDIVTIJ0  
 NzREaXY4aEhtaGNCOFVKV3dGMU1RdHNINUIKRIEzSGNKY3p5Ci0tLS0tRU5EIF  
 JTQSBQUkiWQVRFIetFWS0tLS0tCg==

## Direct Onboarding of HELM Charts

RIFT allows direct onboarding of HELM charts to create CNFDs. In previous releases, helm chart validation was done during instantiation. In release 8.1.1, RIFT.ware now validates helm charts prior to CNFD generation. Invalid helm charts will now fail during onboarding and proper limiting errors are reported.

## API Enhancements

The `tiller-ip` and `tiller-port` fields are removed from the CaaS Account API. The `kube-config` field still remains for the CaaS Account API.

## Model Enhancements

In release 8.1.1, RIFT.ware no longer supports remote helm charts.

- Deprecated the `source-type` field containing the enums `local` and `repo` in the CNFD Data Model (`cnfd:cnfd-catalog`).
- The CNF now only uses the `source-location` field to refer to the name of the local helm chart in the CNFD.

## Fixed Issues in RIFT.ware 8.1.1.0

| Support Tickets | Title                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RIFT-24343      | Cpu stays high for rwmain tasklet due to msgbroker.     | <p>In some instances when many packages were uploaded or instantiations were done, the main rwmain process started using 100% CPU.</p> <p>In this scenario, the CPU usage was high because the lib dispatch library started subscribing for more threads to the DTS router queue and each thread kept polling out of the event loop.</p> <p>This issue is resolved.</p>                                                             |
| RIFT-27451      | Instantiation variables not correctly applied to VNFD   | <p>An operator had an NSD with an instantiation variable associated with input-parameter-xpath entries that point to input-variables entries created on the ping vnfd. After instantiating the NS, the variables were not applied to the targets. If the operator changes the usage of the instantiation variables, then the results were different and unpredictable.</p> <p>This issue is resolved.</p>                           |
| RIFT-27452      | Datacenter instantiation variable not correctly applied | <p>An operator had an NSD with an instantiation variable used to set the datacenter (xpath/datacenter). After instantiating the NS, the variable was not applied. The following message appeared:</p> <p>"operational-status-details": "Datacenter None not present"</p> <p>Instantiation variable substitution for a datacenter was not happening when more than one cloud account was present.</p> <p>This issue is resolved.</p> |
| RIFT-28683      | LCM worker communication error with flavor 16/64/500    | <p>When lots of LCM transactions were performed concurrently on a high load, there was a chance that the Python threads were blocked.</p>                                                                                                                                                                                                                                                                                           |

| Support Tickets | Title                                                                | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |                                                                      | <p>This is a libc (version 2.27) pthread library issue in the condition variable implementation.</p> <p>For more details, see <a href="https://sourceware.org/bugzilla/show_bug.cgi?id=25847">https://sourceware.org/bugzilla/show_bug.cgi?id=25847</a>.</p> <p>This made the affected LCM workers unresponsive and no further LCM actions could be scheduled on those workers.</p> <p>This issue is resolved.</p> |
| RIFT-28883      | NS Instantiation failed on using instantiation-variable.             | <p>While performing an audit discovery using instantiation variables, the NS adoption was failing and the following error message appeared:</p> <p>Instantiating vnfr failed: VM Orig1-NxdnTWHwZ-iovdu0 is not found for discovery.</p> <p>Previously, instantiation-variable-support was not working properly while trying to provide input variables in an attached csv file.</p> <p>This issue is resolved.</p> |
| RIFT-28892      | Handle terminate when NS is in scaling or update state               | <p>If a RIFT.ware NS request was terminated when a NS was scaling (in or out) or in the update process, then the request was denied with the following error response: <b>Cannot terminate while NS is in &lt;state name&gt; state.</b></p> <p>In release 8.1.1, it is now possible to terminate an NS in the scaling or updating state.</p> <p>This issue is resolved.</p>                                        |
| RIFT-29003      | Job details logs in Service Primitives are cluttered and unreadable. | <p>While an operator was attempting a scale operation, service primitives were failing. The operator tried to find the root cause by opening the job-details. It was not in a readable format.</p> <p>The log files generated as part of execution of configuration scripts can now be</p>                                                                                                                         |

| Support Tickets          | Title                           | Description                                                                                                                                                                                                                                                                                       |
|--------------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          |                                 | downloaded through the log-file link as part of the config job data.<br>This issue is resolved.                                                                                                                                                                                                   |
| RIFT-29646<br>RIFT-29632 | HA CGN update logic has changed | In release 8.1.1, the Configuration Generation Number (CGN) displayed on the <b>REDUDANCY: STATUS</b> UI page no longer reflects the NS artefact additions and deletions during LCM actions. Syncing does occur in the backend using the package ID but the changes are not reflected in the CGN. |

## Known Issues in RIFT.ware 8.1.1.0

---

| Support Tickets | Title                                                                | Description                                                                                                                                                                               | Impact                                                                                                                               | Workaround                                                                                                                                                                                                                                                                       |
|-----------------|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RIFT-13715      | Confd configuration transaction abort results in inconsistent state. | Confd can abort configuration change transactions due to its own internal reasons. RIFT.ware cannot undo the changes (on an ABORT form Confd) because it is already internally committed. | This could result in a data mismatch between that is there in the configuration database and what is known to the RIFT.ware backend. | This issue mostly occurs when a user makes successive config changes without any idle time in between the modifications. Any test or automation script using RIFT REST APIs to complete configuration changes must have a delay between two successive config change operations. |
| RIFT-19249      | RBAC API to Update user's project/MANO roles has changed.            | RBAC API to Update user's project/MANO roles has changed. This information needs to be updated in the User Documentation.                                                                 | An older REST API cannot be used to change the User's Project/MANO roles and might break existing scripts.                           | Use the Launchpad UI to change a user's Project/MANO role, or use the New API.                                                                                                                                                                                                   |
| RIFT-21978      | Password is displayed in plain text in event logs.                   | A password is displayed in plain text in the events log.                                                                                                                                  | A password is displayed in plain text in the events log.                                                                             | N/A                                                                                                                                                                                                                                                                              |
| RIFT-21923      | TC_CONCURRENT_NS_TERMINATE: PackageDeleteError                       | When multiple descriptor (NSD/VNFD) packages are deleted concurrently using an API, the config data in the Confd might be missing for a brief period for a few packages.                  | The config data in the Confd might be missing for a brief period for a few packages.                                                 | When using an API to delete multiple package entries, do not send multiple DELETE requests concurrently. Wait for a request to complete before firing another request.                                                                                                           |

| Support Tickets | Title                                                                           | Description                                                                                                                                                                                  | Impact                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Workaround                                                             |
|-----------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| RIFT-23621      | Introduce a new field in VNFM account page for RIFT.ware<-->VNFM handshake URL. | This is a request to add a new field in the VNFM accounts page that the Operator so the operator can populate the URL for handshaking. If the URL field is empty, handshake is not required. | RIFT.ware uses a GET call to /vnf_instances to validate VNFM Account. If a SVNFM account does not support GET on /vnf_instances, then the VNFM Account validation may Fail. The UI shows the account status as failed. If the SVNFM account supports notification, then RIFT.ware sends a subscription request next as part of the validation. This is successfully and the VNFM account status is successful. Therefore, a SVNFM account which doesn't support GET call on /vnf_instances, then the VNFM Account status will appear to fail. This will not block any further operations such as instantiation and termination. | N/A                                                                    |
| RIFT-23760      | Terminating one service with the same name on any Launchpad                     | This issue occurs because all services on all Launchpads have the same                                                                                                                       | If a user terminates any of the services on either of the network-services,                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Use a unique NSR name if an organization is using multiple Launchpads. |

| Support Tickets | Title                                                                       | Description                                                                                                                                                                                                                                 | Impact                                                                                                                                                                      | Workaround                                                                                                                                          |
|-----------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | interrupts existing services on all Launchpads.                             | UniqueID which is the InstanceID on ES2. When a user terminates a service, then terminates Instance on ES2 which causes issues for all existing services because they have the same UniqueID.                                               | this will make any other services orphan.                                                                                                                                   | Ensure that an OpenStack tenant is controlled by a single Launchpad.<br><b>Note:</b> There are no plans to change this behavior.                    |
| RIFT-24706      | Adopted network-service fails after HA failover.                            | A user instantiated and NS with a vim-network-name provided for both the mgmt-network and ping-pong-vld. During instantiation, NS failed at the VL-init stage.                                                                              | N/A                                                                                                                                                                         | If user is going to configure the vim-network-name in VLDs using input variable xpaths, then the ip-profile-ref value in the NSD should not be set. |
| RIFT-24872      | Updating ipv4 to ipv6 fails in the case of FIXED IP ADDRESS and vice versa. | A user modified the 'FIXED IP ADDRESS' field and then added an IPv4 address in the descriptor details pane for a VDU. Next the user updated the configuration to IPV6 and clicked the 'Update' button. An error message appeared on the UI. | This issue only occurs when there are two IP addresses in interface configurations (IPv4 or IPv6) and the user tries to change one address to one with a different version. | Delete the existing address and add one new address with different a version.                                                                       |
| RIFT-24992      | Running AWS NS fails after failover to a new LP.                            | A user created an NS on Launchpad 1. The service is in the running state for tiny descriptors with an AWS VM account. The user set up an HA pairing with another Launchpad of a different version.                                          | N/A                                                                                                                                                                         | Terminate the NS and re-instantiate in the second LP.                                                                                               |

| Support Tickets | Title                                                                                                                    | Description                                                                                                                                                                                                                                                                                                                         | Impact                                                                                                                                        | Workaround |
|-----------------|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|------------|
|                 |                                                                                                                          | Failover occurred to Launchpad 2; the NS was that running in the Launchpad 1 failed in instantiate in Launchpad 2.                                                                                                                                                                                                                  |                                                                                                                                               |            |
| RIFT-25495      | OpenId stats publisher overwrites old data on restart                                                                    | After restating Launchpad, the stats corresponding to XPath rw-openidc-provider:openidc-provider-state/rw-openidc-provider:statistics are overwritten.                                                                                                                                                                              | After restarting Launchpad, the openidc stats are reset rather than aggregated.                                                               | N/A        |
| RIFT-25610      | AWS - NS stuck indefinitely in VNF-INIT phase when using a scaling-group and constituent VNF's start-by-default = False. | A user tried to instantiate a simple NS with one VNF, a single VDU (for an Ubuntu VM), and a single VLD. The NSD had a scaling group and the constituent VNFD's start-by-default value is False. When the NS is instantiated, it is stuck forever in the VNF-INIT phase. There are no errors seen in the Event Logs or in rift.log. | No VNF is started by default. Since the min-instance-count for scaling group is 1, the VNF is supposed to be scaled out once NS instantiates. | N/A        |
| RIFT-25902      | Discovery gets triggered even on failed account due to non-reachable network status.                                     | A user created a new VIM account to OpenStack. If the network is down and the cloud setup is not reachable, then when a user attempts discovery it fails but                                                                                                                                                                        | Misleading status of VIM account on UI.                                                                                                       | N/A        |

| Support Tickets | Title                                                                             | Description                                                                                                                                                                                    | Impact                                                                                                                   | Workaround                                                             |
|-----------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
|                 |                                                                                   | the process is still triggered.                                                                                                                                                                |                                                                                                                          |                                                                        |
| RIFT-26498      | Creation of custom "User" in the VDUs via RIFT.ware isn't working.                | In the NS instantiation page, Launchpad provides the option to create new users in the VDUs. This functionality isn't working correctly.                                                       | A user is not able to add users to VMS from the Launchpad instantiation screen.                                          | N/A                                                                    |
| RIFT-26409      | AWS alarm was failed to create after one of the instance (running) failure in AWS | RIFT supports AWS alarms (SNS notifications). These alarms help in monitoring the state of the VM (VDU). Upon failure of the instance in the AWS, an SNS notification appears.                 | After the instance failure from the AWS console, the NS should fail in Launchpad. This issue is not occurring correctly. | A user must depend on the monitoring params to check the state of VMs. |
| RIFT-26498      | Creation of custom "User" in the VDUs via RIFT.ware isn't working.                | In the NS instantiation page, Launchpad provides the option to create new users in the VDUs. This is not working correctly.                                                                    | It is possible to add users to VMs from the Launchpad instantiation screen.                                              | N/A                                                                    |
| RIFT-26567      | Placement groups: transaction failure prints 5 traps<br>riftLPUserSessionEnd      | An operator loaded ping pong scaling descriptors and then deleted placement groups from the NSD an VNFDs. Next, the user created and instantiated the NS. An error message and 5 traps appear. | Duplicate traps are generated.                                                                                           | N/A                                                                    |
| RIFT-26409      | AWS alarm was failed to create after one of the instance (running) failure in AWS | RIFT supports AWS alarms (SNS notifications). These alarms help in monitoring the state                                                                                                        | After the instance failure from the AWS console, the NS should fail in Launchpad. This                                   | A user must depend on the monitoring params to check the state of VMs. |

| Support Tickets | Title                                                                                                | Description                                                                                                                                                                                                                                                             | Impact                                                                                                                                                          | Workaround                                                                           |
|-----------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|                 |                                                                                                      | of the VM (VDU). Upon failure of the instance in the AWS, an SNS notification appears.                                                                                                                                                                                  | issue is not occurring correctly.                                                                                                                               |                                                                                      |
| RIFT-27186      | Service instantiation fails with incorrect timeout error                                             | RIFT.ware was not capturing the real reason why pods failed. Pods only retry in the case of failure, so it was not possible to say that pod creation was in a Failure state.                                                                                            | Instantiation fails in this scenario.                                                                                                                           | View the cluster event logs in Launchpad.                                            |
| RIFT-27241      | Instantiating ping-pong fails after ping VNF descriptor is modified and it is impossible to tell why | If a user choses the management interface type in a VNFD as VDU or Connection Point and the corresponding VDU/CP is not selected in the next input, then no error appears until the user attempts to instantiate. The configuration fails in the instantiation attempt. | If a user chooses the mgmt-interface type but no ID is selected, then RIFT.ware considers both the type and the ID as empty.                                    | If a user configures the NS properly, then an issue will not occur.                  |
| RIFT-28142      | Create service progress did not complete but service is created. Cannot instantiate.                 | An operator attempted to instantiate a Network Service and the screen froze. However, the Services tab on the UI lists the service as Created.                                                                                                                          | This issue occurs when a Datacenter account is created before project creation is complete. It might happen when using APIs, but it is not likely using the UI. | Create another project and wait for 30 seconds before creating a Datacenter account. |

| Support Tickets | Title                                                | Description                                                                                                                                                                                                                                                                                                                                   | Impact                                                                                                                     | Workaround                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RIFT-28449      | Dependency issue in cloud accounts, config, operdata | If an operator has tens or hundreds of accounts/datacenters configured under a single project which are not being used by any NS, then it is necessary to delete all the cloud accounts prior to deleting that one project.                                                                                                                   | An error message appears if you do not delete all the cloud accounts prior to deleting the project                         | Delete all cloud accounts prior to deleting the project.                                                                                                                                                                            |
| RIFT-28519      | Config agent account broken                          | An operator went to the connector plugin in the UI and created a config agent account with random credentials. After 5 to 10 minutes, the account did not have a connection status or a message. Also, the field 'secret' from the UI is not present in the API response when a user queries config-agent. Lastly, the oper-state is missing. | RIFT is the only config agent that is working correctly. Other VNFDs and NSDs need to use RIFT as the configuration agent. | RIFT is the only config agent that is working correctly. Therefore, configuration scripts need to comply with RIFT.                                                                                                                 |
| RIFT-28834      | Overwrite existing package option is not available   | An operator onboarded a package in the UI. An operator might want to onboard the same package again and overwrite the existing package. The option to overwrite the existing package is missing from the UI.                                                                                                                                  | It is not possible to overwrite an existing package in the UI.                                                             | If a package is not in use, then it can be deleted from the catalog and then onboarded again. To onboard an updated version of a package, ensure that it has a new unique id. Note: When using Launchpad to copy a package, the new |

| Support Tickets | Title                                                                                                              | Description                                                                                                                                                                                                                                            | Impact                                                                                                                                                                     | Workaround                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |                                                                                                                    |                                                                                                                                                                                                                                                        |                                                                                                                                                                            | package is automatically given a new ID. The package with the new ID can be onboarded and it will co-exist with the current package.                                                                |
| RIFT-29355      | NS Instantiation stuck at INIT stage while using instantiation variable file                                       | An operator added a datacenter and then uploaded ping pong descriptors. In the Create NS wizard, a user attached the instantiation variable file and clicked YES on the instantiate radio button. The results is the NS being stuck in the INIT stage. | Using import-instantiation-variables-v2 rpc is currently not supported. NS will get stuck at the INIT stage and a user will eventually have to terminate or delete the NS. | Do not select the Instantiate radio-button to Yes. It is still possible to create a NS by attaching the Instantiation variable file and then later perform instantiation actions on the created NS. |
| RIFT-29543      | CNFD service instantiation failing with error release monitoring failed: 'str' object has no attribute 'type_yang' | CNFD service instantiation fails when a helm-chart-based Launchpad is setup in a non-default namespace and uses the chart to launch CNFs in other namespaces. The issue is not seen with standalone Launchpads.                                        | NS instantiations fail with the following error message: Release monitoring failed.                                                                                        | Use kube-config with a explicitly defined namespace.                                                                                                                                                |
| RIFT-29601      | Brownfield Discovery with Scaling descriptors creating fresh new instance on OpenStack.                            | An operator attempts to validate Brownfield Discovery. The NSD without a scaling group is fully discovered but the NSD with a scaling                                                                                                                  | Discovery does not work correctly. While attempting discovery, a new instance is created instead of discovering the                                                        | Set NSD/constituent-vnfd/start-by-default to true.                                                                                                                                                  |

| Support Tickets | Title                                                                              | Description                                                                                                                                                                                                                                                              | Impact                                                                                                                              | Workaround                                                                          |
|-----------------|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                 |                                                                                    | group is only partially discovered.                                                                                                                                                                                                                                      | existing scaling instance in the VIM.                                                                                               |                                                                                     |
| RIFT-29695      | Resource Naming feature not working                                                | The cloud resource naming feature is not working correctly in this release.                                                                                                                                                                                              | If a user uploads the resource naming rules and tries to instantiate the NS, it will fail at the NF instantiation stage.            | Do not use this feature in release 8.1.1.                                           |
| RIFT-29396      | LP-based CNF orchestration does not work when restricted to non-default namespaces | RIFT.ware expects a full-cluster-access Kube-config and uses this config to create a net-attach-def CRD in the default namespace used for networking. If the Kube-config provided by the user does not have access to the "default" namespace, then instantiation fails. | Launchpad based CNF orchestration does not work when the CaaS Account restricts kubernetes cluster usage to non-default namespaces. | Kubernetes cluster administrator has to provide access cluster-wide access for now. |