

Dell Redfish Life Cycle Events streaming from iDRAC

Abstract

Configuration and inventory changes in DELL PowerEdge servers is an ever-mutable process till the system comes to end of life. Capturing such changes helps IT administrators to manage hardware and software efficiently. This white paper helps in subscribing to Redfish resource-related configuration and inventory changes and receive them as Redfish Life Cycle Events.

June 2021

Revisions

Date	Description
June 2021	Initial release

Acknowledgments

Author: Mahesh Babu R

Support: Venkataragavan V

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

© 2021 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

Table of contents

Revisions.....	2
Acknowledgements.....	Error! Bookmark not defined.
Executive summary.....	4
1 Redfish Life Cycle Events Overview	5
1.1 Terms and Definitions.....	6
1.2 Prerequisites.....	6
2 Configuring Redfish Life Cycle Events.....	7
2.1 Configuring RLCE using subscription method (HTTP Push)	7
2.2 Configuring RLCE through SSE	11
2.3 Supported Event Types and Report format.....	14
2.4 Feature constraints and limitations.....	15
2.5 Troubleshooting and Tips.....	15
2.6 Best practices	16
2.7 Key Links	16
3 Conclusion.....	17

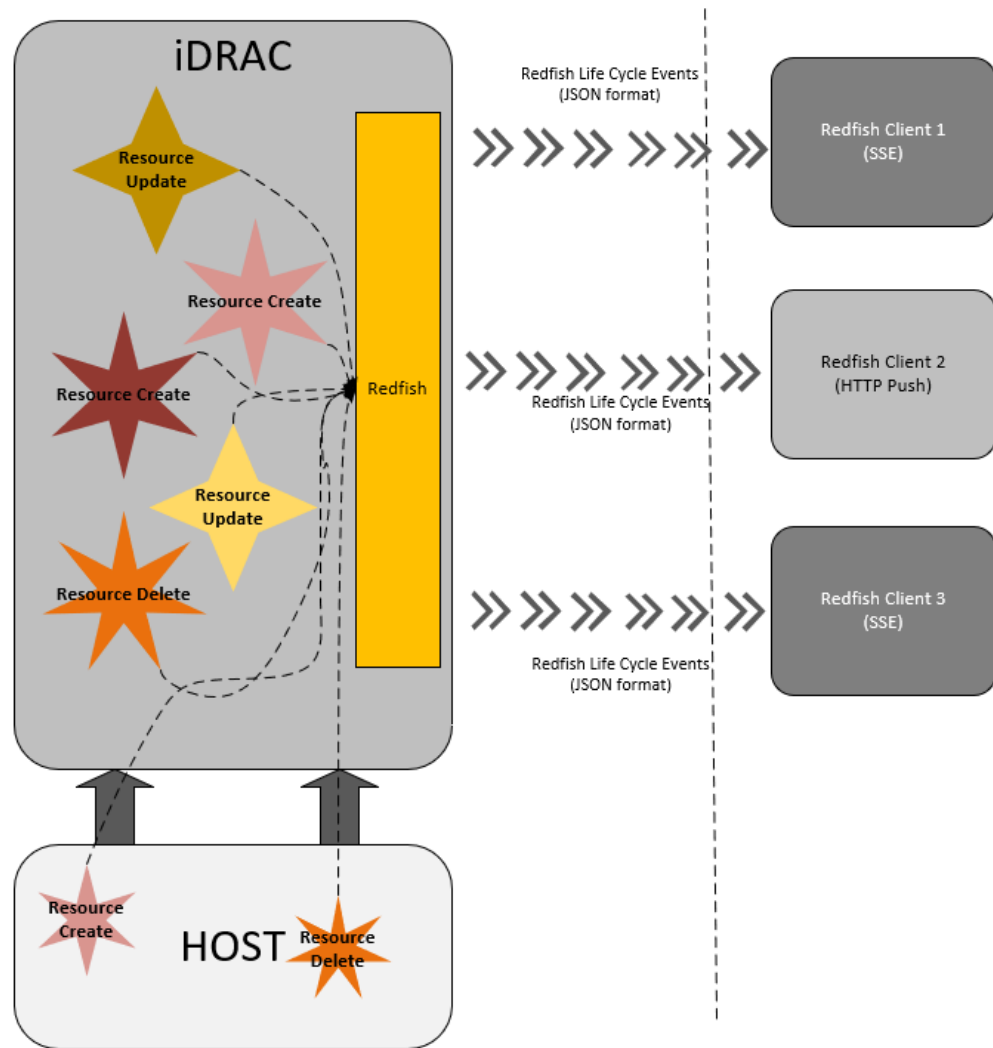
Executive summary

Redfish APIs have been widely adopted for configuring, monitoring, and managing servers. Effective from iDRAC9 version 5.00.00.00, IT managers can receive hardware and software change events uninterrupted and in real time through Redfish event subscriptions. This would track both internally and externally initiated changes such as create, delete and update of about 200+ Redfish resources. The events are streamed to user-defined clients, created either through SSE or HTTP subscription mechanism and would complement the existing alert-based events. The reports can be parsed using comprehensive analytic tools for enhanced diagnosis and outcomes.

1 Redfish Life Cycle Events Overview

Redfish Life Cycle Events streaming is an extension to existing Redfish eventing service that can capture the changes to inventory, configurations, and monitoring aspects of a server in real time. Beginning with iDRAC9 version 5.00.00.00, Redfish clients can subscribe to hardware and software-related changes over channels supporting SSE or HTTP push subscriptions. This paper includes steps and methods for subscribing Redfish Life Cycle Events (RLCE).

The following diagram shows the high-level flow of Redfish Life Cycle Events streaming to Redfish clients.



1.1 Terms and Definitions

Redfish Life Cycle Event (RLCE) – An event reported as per DMTF Redfish event specification when subscriptions are created with Event Type as Other.

SSE – Server Sent Events allow clients to receive Redfish life cycle events on creating subscription over a web interface.

1.2 Prerequisites

Redfish Life Cycle Events can be subscribed by following event subscription guidelines with no additional iDRAC licensing requirement, from iDRAC9 version 5.00.00.00 onwards.

2 Configuring Redfish Life Cycle Events

Clients can subscribe to Redfish life cycle events by creating event subscriptions as defined by DMTF Redfish event subscription service. iDRAC9 supports two styles of subscriptions, that are SSE and HTTP Push based. By default, this feature does not require any additional iDRAC license and only requires Redfish service to be enabled in order to receive all types of Redfish events i.e. alerts and life cycle events.

To differentiate between the Redfish alerts and life cycle logs, subscriptions are created with “EventTypes” as “Other”. Configuring events can be further streamlined by listing small subsets of hardware or software components, that can be queried through resource types i.e. storage, network, sessions, jobs etc. Clients can also configure events on scenarios of Redfish resources being created, deleted, and updated.

2.1 Configuring RLCE using subscription method (HTTP Push)

Subscription for Redfish life cycle events, through HTTP push mechanism is created on performing a POST operation with necessary key-value payload consisting of Destination, Context, and Protocol as mandatory properties. With this the client gets to receive all changes in the system, that have a Redfish resource mapping. Filter-based subscription can be created with payload like OriginResources and MessageId, which is in addition to mandatory properties.

Schema - <https://redfish.dmtf.org/schemas/v1/EventDestination.xml>

Subscription for all events -

The screenshot displays a REST client interface for a POST request to the endpoint `https://192.168.1.213/redfish/v1/EventService/Subscriptions`. The request body is a JSON object with the following structure:

```

1 {
2   "Destination": "https://192.168.1.213:5001",
3   "Context": "root",
4   "Protocol": "Redfish",
5   "EventFormatType": "Event",
6   "EventTypes": [
7     "Other"
8   ],
9   "ResourceTypes": [
10    "Job", "Drive"
11  ]
12 }

```

The response is a 201 Created status with the following headers:

```

Date: Thu, 17 Jun 2021 06:10:10 GMT - 4d 11h
Server: Apache
OData-Version: 4.0
Access-Control-Allow-Origin: *
Cache-Control: no-cache

```

The response body is a JSON array of messages:

```

{
  "@Message.ExtendedInfo": [
    {
      "Message": "The resource has been created successfully",
      "MessageArgs": [],
      "MessageArgs@odata.count": 0,
      "MessageId": "..."
    },
    {
      "Message": "A new resource is successfully created.",
      "MessageArgs": [],
      "MessageArgs@odata.count": 0,
      "MessageId": "IDF..."
    }
  ]
}

```

Output: Subscription for all events -

201 Created

```
HEADERS @ pretty < BODY @
Date: Thu, 17 Jun 2021 05:35:03 GMT +10h 34m
Server: Apache
OData-Version: 4.0
Access-Control-Allow-Origin: *
Cache-Control: no-cache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=63072000; includeSubdomains; preload
Content-Type: application/json; charset=utf-8
Location: /redfish/v1/EventService/Subscriptions/c4609568-cf2d-11eb-a919-509a4ca815bc
Content-Length: 1 kilobytes
Keep-Alive: timeout=60, max=100
Connection: Keep-Alive
COMPLETE REQUEST HEADERS

@Message.ExtendedInfo: [
  {
    Message: "The resource has been created successfully",
    MessageArgs: [],
    MessageArgs@odata.count: 0,
    MessageId: "Base.1.0.Created",
    RelatedProperties: [],
    RelatedProperties@odata.count: 0,
    Resolution: "None",
    Severity: "OK"
  },
  {
    Message: "A new resource is successfully created.",
    MessageArgs: [],
    MessageArgs@odata.count: 0,
    MessageId: "IDRAC.2.5.SYS414",
    RelatedProperties: [],
    RelatedProperties@odata.count: 0,
    Resolution: "No response action is required.",
    Severity: "Informational"
  }
],
@odata.context: "/redfish/v1/$metadata#EventDestination.EventDestination",
@odata.id: "/redfish/v1/EventService/Subscriptions/c4609568-cf2d-11eb-a919-509a4ca815bc",
@odata.type: "#EventDestination.v1_0_0.EventDestination",
Actions: {
  #EventDestination.ResumeSubscription: {
    target: "/redfish/v1/EventService/Subscriptions/c4609568-cf2d-11eb-a919-509a4ca815bc/Actions/EventDestination.ResumeSubscription"
  }
},
Context: "root",
DeliveryRetryPolicy: "RetryForever",
Description: "Event Subscription Details",
Destination: "https://17.1.0.01",
EventFormatType: "Event",
EventTypes: [
  "Other"
],
EventTypes@odata.count: 1,
HttpHeaders: [],
HttpHeaders@odata.count: 0,
Id: "c4609568-cf2d-11eb-a919-509a4ca815bc",
Name: "EventSubscription c4609568-cf2d-11eb-a919-509a4ca815bc",
Protocol: "Redfish",
Status: {
  Health: "OK",
  HealthRollup: "OK",
  State: "Enabled"
},
SubscriptionType: "RedfishEvent"
}
Top Bottom Collapse Open
```


Subscription with filtering: OriginResources -

METHOD: POST | SCHEME // HOST [: PORT] | PATH [? * QUERY] | length: 60 byte(s)

URL: https://redfish/v1/EventService/Subscriptions

HEADERS:

- Authorization: Basic cm9vdDpjYVWx2aW4=
- OdataVersion: 4.0
- Content-Type: application/json

BODY:

```
1 {
2   "Destination": "https://7.19 31:5001",
3   "Context": "root",
4   "Protocol": "Redfish",
5   "EventFormatType": "Event",
6   "EventTypes": [
7     "Other"
8   ],
9   "OriginResources": [
10    {
11      "@odata.id": "/redfish/v1/Systems/System.Embedded.1"
12    }
13  ]
14 }
```

Response | Cache Detected - Elapsed Time: 1.27s

201 Created

HEADERS:

- Date: Thu, 17 Jun 2021 06:05:02 GMT +10h 34m
- Server: Apache
- OData-Version: 4.0
- Access-Control-Allow-Origin: *
- Cache-Control: no-cache
- X-Frame-Options: DENY
- Strict-Transport-Security: max-age=63072000; includeSubDomains; preload

BODY:

```
{
  "@Message.ExtendedInfo": [
    {
      "Message": "The resource has been created successfully",
      "MessageArgs": [],
      "MessageArgs@odata.count": 0,
      "MessageId": "Base.1.8.Created",
      "MessageSeverity": "Information",
      "MessageText": "The resource has been created successfully."
    },
    {
      "Message": "A new resource is successfully created.",
      "MessageArgs": [],
      "MessageArgs@odata.count": 0,
      "MessageId": "IDRAC.2.5.SYS414",
      "MessageSeverity": "Information",
      "MessageText": "A new resource is successfully created."
    }
  ],
  "@odata.context": "/redfish/v1/$metadata#EventDestination.EventDestination",
  "@odata.id": "/redfish/v1/EventService/Subscriptions/f4c07b54-cf31-11eb-a8ac-509a4ca815bc",
  "@odata.type": "#EventDestination.v1_9_0.EventDestination",
  "EventDestination": {
    "Destination": "https://7.19 31:5001",
    "Context": "root",
    "Protocol": "Redfish",
    "EventFormatType": "Event",
    "EventTypes": [
      "Other"
    ],
    "OriginResources": [
      {
        "@odata.id": "/redfish/v1/Systems/System.Embedded.1"
      }
    ]
  }
}
```

Subscription with filtering: ResourceTypes –

METHOD: POST
SCHEME: https://
HOST: .97.132.5001
PATH: /redfish/v1/EventService/Subscriptions

HEADERS:
Authorization: Basic cm9vdDpjYWx2aW4=
OdataVersion: 4.0
Content-Type: application/json

BODY:
1 {
2 "Destination": "https://.97.132.5001",
3 "Context": "root",
4 "Protocol": "Redfish",
5 "EventFormatType": "Event",
6 "EventTypes": [
7 "Other"
8],
9 "ResourceTypes": [
10 "Job", "Drive"
11]
12 }

Response
201 Created

HEADERS:
Date: Thu, 17 Jun 2021 06:10:10 GMT +10h 34m
Server: Apache
OData-Version: 4.0
Access-Control-Allow-Origin: *

BODY:
{
 "@Message.ExtendedInfo": [{
 "Message": "The resource has been created successfully",
 "MessageArgs": [],
 "MessageArgs@odata.count": 0,
 "@odata.context": "/redfish/v1/\$metadata#EventDestination.EventDestination",
 "@odata.id": "/redfish/v1/EventService/Subscriptions/ac1ca4e4-cf32-11eb-91ab-509a4ca815bc",
 "@odata.type": "#EventDestination.v1_9_0.EventDestination",
 "Actions": {
 "#EventDestination.ResumeSubscription": {
 "target": "/redfish/v1/EventService/Subscriptions/ac1ca4e4-cf32-11eb-91ab-509a4ca815bc"
 }
 },
 "Context": "root",
 "DeliveryRetryPolicy": "RetryForever",
 "Description": "Event Subscription Details",
 "Destination": "https://.97.132.5001",
 "EventFormatType": "Event",
 "EventTypes": ["Other"],
 "EventTypes@odata.count": 1,
 "HttpHeaders": [],
 "HttpHeaders@odata.count": 0,
 "Id": "ac1ca4e4-cf32-11eb-91ab-509a4ca815bc",
 "Name": "EventSubscription ac1ca4e4-cf32-11eb-91ab-509a4ca815bc",
 "Protocol": "Redfish",
 "ResourceTypes": ["Job", "Drive"],
 }]
}

Subscription Collection -

METHOD: GET | SCHEME // HOST [":" PORT] [PATH ["?" QUERY]]

https:// /redfish/v1/EventService/Subscriptions

QUERY PARAMETERS

HEADERS

- Authorization: Basic cm9vdDpjYWx2aW4=
- OdataVersion: 4.0

Form | BODY

XHR does not allow payloads for GET request.

Response

200 OK

HEADERS

```

Date: Thu, 17 Jun 2021 06:14:59 GMT +10h 34m
Server: Apache
Allow: POST
Link: </redfish/v1/Schemas/EventDestinationCollection.json>;rel=describedby
Odata-Version: 4.0
Access-Control-Allow-Origin: *
Cache-Control: no-cache
X-Frame-Options: DENY
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 233 bytes
Keep-Alive: timeout=60, max=100
Connection: Keep-Alive
Content-Type: application/json;odata.metadata=minimal; charset=utf-8
    
```

COMPLETE REQUEST HEADERS

prettify | BODY

```

{
  "@odata.context": "/redfish/v1/$metadata#EventDestinationCollection.EventDestinationCollection",
  "@odata.id": "/redfish/v1/EventService/Subscriptions",
  "@odata.type": "#EventDestinationCollection.EventDestinationCollection",
  Description: "List of Event subscriptions",
  Members: [
    {
      "@odata.id": "/redfish/v1/EventService/Subscriptions/ac1ca4e4-cf32-11eb-91ab-509a4ca815bc"
    },
    {
      "@odata.id": "/redfish/v1/EventService/Subscriptions/f4c07b54-cf31-11eb-a8ac-509a4ca815bc"
    }
  ],
  Members@odata.count: 2,
  Name: "Event Subscriptions Collection"
}
    
```

lines nums

2.2 Configuring RLCE through SSE

Clients can subscribe to Redfish life cycle events with subscription created over an SSE channel, by performing GET on the SSE URI. Events generated on iDRAC during a five second period are grouped and sent to the client as a single RLCE, as per Redfish event message SSE stream. Every RLCE can have an array of events confined to hardware and/or software Redfish resource. Resource and URI-specific SSE subscriptions are also supported by adding filter logic on ResourceTypes or OriginResources respectively. A maximum of 10 entities can be considered per subscription.

Connection can be terminated by either client or iDRAC event service. If no events are sent for more than an hour, then the connection with the client would be closed from the event service endpoint. On closure of connection, the subscription entry is removed from the collection of subscriptions. A maximum of 20 subscriptions can be created, including both SSE or HTTP PUSH based and also legacy subscriptions containing alerts.

Redfish Uri for SSE streaming –

All reports -

[https:// <iDRAC_IP>/redfish/v1/SSE?\\$filter=\(EventType eq 'Other'\)](https://<iDRAC_IP>/redfish/v1/SSE?$filter=(EventType eq 'Other'))

```
← → × Not secure | 1 | /redfish/v1/SSE?$filter=(EventType%20eq%20%27Other%27)
Apps JSONLint - The JSO... White Papers - Tech... Storage Controllers List of PowerEdge R... kronos IP Camera WiFi not... Redfish Schema Ind... settings GitHub - DMTF/Red... Reslet Client Tightness in Hamstr... LAAS

id: 11244567
data: [{"@odata.context": "/redfish/v1/$metadata#Event.Event", "@odata.type": "#Event.v1_3_0.Event", "Events": [{"EventId": "17296629", "EventTimestamp": "2021-06-17T01:22:00-05:00", "EventType": "Other", "MemberId": "0", "MessageId": "ResourceEvent.1.0.ResourceCreated", "OriginOfCondition": {"@odata.id": "/redfish/v1/EventService/Subscriptions/46b0bbf98a0fc4182c89d677ba42f6d9"}}, {"Events@odata.count": 1, "Id": "11244567", "Name": "Redfish LifeCycle Events"}]}]

id: 11244568
data: [{"@odata.context": "/redfish/v1/$metadata#Event.Event", "@odata.type": "#Event.v1_3_0.Event", "Events": [{"EventId": "17296630", "EventTimestamp": "2021-06-17T01:22:02-05:00", "EventType": "Other", "MemberId": "0", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-2-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-2-1"}}, {"EventId": "17296631", "EventTimestamp": "2021-06-17T01:22:02-05:00", "EventType": "Other", "MemberId": "1", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-2-1"}}, {"EventId": "17296632", "EventTimestamp": "2021-06-17T01:22:02-05:00", "EventType": "Other", "MemberId": "2", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-1-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-1-1"}}, {"EventId": "17296633", "EventTimestamp": "2021-06-17T01:22:02-05:00", "EventType": "Other", "MemberId": "3", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-1-1"}}, {"Events@odata.count": 4, "Id": "11244568", "Name": "Redfish LifeCycle Events"}]}]

id: 11244569
data: [{"@odata.context": "/redfish/v1/$metadata#Event.Event", "@odata.type": "#Event.v1_3_0.Event", "Events": [{"EventId": "17296634", "EventTimestamp": "2021-06-17T01:22:04-05:00", "EventType": "Other", "MemberId": "0", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-4-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-4-1"}}, {"EventId": "17296635", "EventTimestamp": "2021-06-17T01:22:04-05:00", "EventType": "Other", "MemberId": "1", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-4-1"}}, {"EventId": "17296636", "EventTimestamp": "2021-06-17T01:22:04-05:00", "EventType": "Other", "MemberId": "2", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-3-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-3-1"}}, {"EventId": "17296637", "EventTimestamp": "2021-06-17T01:22:04-05:00", "EventType": "Other", "MemberId": "3", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-3-1"}}, {"Events@odata.count": 4, "Id": "11244569", "Name": "Redfish LifeCycle Events"}]}]

id: 11244570
data: [{"@odata.context": "/redfish/v1/$metadata#Event.Event", "@odata.type": "#Event.v1_3_0.Event", "Events": [{"EventId": "17296638", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "0", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-4-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-4-1"}}, {"EventId": "17296639", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "1", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-4-1"}}, {"EventId": "17296640", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "2", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-3-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-3-1"}}, {"EventId": "17296641", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "3", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-2-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-2-1"}}, {"EventId": "17296642", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "4", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Chassis/System.Embedded.1/Power"}}, {"EventId": "17296643", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "5", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-2-1"}}, {"EventId": "17296644", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "6", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-1-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-1-1"}}, {"EventId": "17296645", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "7", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-3-1"}}, {"EventId": "17296646", "EventTimestamp": "2021-06-17T01:22:08-05:00", "EventType": "Other", "MemberId": "8", "MessageId": "ResourceEvent.1.0.ResourceChanged", "OriginOfCondition": {"@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-1-1"}}, {"Events@odata.count": 9, "Id": "11244570", "Name": "Redfish LifeCycle Events"}]}]
```

OriginResource based report –

[https://<iDRAC_IP>/redfish/v1/SSE?\\$filter=\(EventType%20eq%20%27Other%27\)%20and%20\(OriginResource%20eq%20%27redfish/v1/Systems/System.Embedded.1%27\)](https://<iDRAC_IP>/redfish/v1/SSE?$filter=(EventType%20eq%20%27Other%27)%20and%20(OriginResource%20eq%20%27redfish/v1/Systems/System.Embedded.1%27))

2.3 Supported Event Types and Report format

Event service on receiving a “GET” request from an SSE client, creates an entry of the SSE request in the subscription collection and starts streaming Redfish life cycle events as per subscription criteria. The message that is received on the client contains the fields defined by SSE protocol: **Id** and **Data**, with **Data** as part of the payload contains events as per DMTF Redfish Event specification.

Clients that subscribe through HTTP push method, receive Redfish life cycle events as defined in the specification with properties: **Id**, **Name**, and **Events**. **Events** property would capture all the events happening during the defined period and is represented through **EventId**, **EventTimestamp**, **EventType**, **MessageId**, and **OriginOfCondition**.

Schema - <https://redfish.dmtf.org/schemas/v1/Event.xml>

Sample Output –

```
{
  "@odata.context": "/redfish/v1/$metadata#Event.Event",
  "@odata.type": "#Event.v1_3_0.Event",
  "Events": [
    {
      "EventId": "17817475",
      "EventTimestamp": "2021-06-20T14:18:04-05:00",
      "EventType": "Other",
      "MemberId": "0",
      "MessageId": "ResourceEvent.1.0.ResourceChanged",
      "OriginOfCondition": {
        "@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-4-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-4-1"
      }
    },
    {
      "EventId": "17817476",
      "EventTimestamp": "2021-06-20T14:18:04-05:00",
      "EventType": "Other",
      "MemberId": "1",
      "MessageId": "ResourceEvent.1.0.ResourceChanged",
      "OriginOfCondition": {
        "@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-4-1"
      }
    },
    {
      "EventId": "17817477",
      "EventTimestamp": "2021-06-20T14:18:04-05:00",
      "EventType": "Other",
      "MemberId": "2",
      "MessageId": "ResourceEvent.1.0.ResourceChanged",
      "OriginOfCondition": {
        "@odata.id": "/redfish/v1/Systems/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-3-1/Oem/Dell/DellNICPortMetrics/NIC.Integrated.1-3-1"
      }
    }
  ],
  "Events@odata.count": 3,
  "Id": "11330840",
  "Name": "Redfish LifeCycle Events"
}
```

To differentiate between legacy alert subscription and Redfish life cycle event subscription and to maintain the backward compatibility, following deviations from schema is taken:

- **EventType** = 'Other' for Redfish Life Cycle Events. If the EventType is not specified in the subscription, service considers the subscription as EventType = Alert.

- **ResourceTypes** , **MessageId's** , **OriginResources** are only supported on EventType = Other.
- **ResourceTypes**: This property is useful when a client is interested in a subset of resource types as an array (e.g. Drive, Network, Job, Session, etc). If the generated event's resource type matches any of the ResourceTypes listed in the subscription, the event is sent to the listener. When an empty array is passed in the ResourceTypes, events from all resources are sent to the event listener. In case of a no match, the event is discarded (maximum array elements is 10). Full list of ResourceTypes can be derived out of schema names available at - <https://redfish.dmtf.org/schemas/v1/>, barring collection and nonresource schemas.
- **OriginResources**: This property is used when a client is interested in specific resource instances. Client specifies a list of resource URIs when creating a subscription. When events are generated, it is mapped to a resource URI (i.e. OriginOfCondition property in the examples). When the generated event's resource URI matches any of the resource URIs specified in the subscription, the event is sent to the listener. When an empty array is passed in the ResourceTypes, events from all resources are sent to the event listener. If there is no match, the event is discarded (maximum array elements are 10).
- **MessageIds**: Of the various messages defined in ResourceEvent and EEMI registries, following are the Message Ids that are supported on subscriptions:
 - ResourceCreated: Indicates creation of a resource or addition of new devices.
 - ResourceChanged: Displays one or more resource properties that were updated. Changes in Resource Status (Resource.Status) property are excluded from this event.
 - ResourceRemoved: Indicates deletion of a resource or removal of a device.

2.4 Feature constraints and limitations

List of features that are not supported in iDRAC9 version 5.00.00.00 and may be supported in a future release -

- Support for IncludeOriginOfCondition=true.
- Combination of ResourceTypes and MessageId's or OriginResources and MessageId's.
- RegistryPrefixes, SubordinateResources in a subscription.
- Other MessageIds barring ResourceCreated, ResourceChanged, Resource Removed.

List of features not supported or supported with limitations -

- All types of updates related to resource collections are not supported.
- Updates related to static resources like Registries and resource definitions are not supported.
- LastEventId in SSE is not supported.
- MessageId's are supported only for life cycle events subscription.
- Subscriptions with more than 10 ResourceTypes or OriginResources.
- Destination field only supports HTTPS protocol.

2.5 Troubleshooting and Tips

If Redfish life cycle events are not displayed for a client through an HTTP PUSH based subscription, then check if Destination property contains a valid address or not. This can also be confirmed by referring to iDRAC Lifecycle logs collection.

2.6 Best practices

- It is recommended to create subscriptions with specific resource or URI, as it allows easier processing of reports, especially during a host reboot scenario.

2.7 Redfish Events: Alerts / Telemetry / Life cycle events

Feature	Alerts	Telemetry	Life cycle events
System Anomalies	✓	✗	✗
Metrics – Time series data	✗	✓	✗
Resource - Updates/Actions	✗	*✓	✓
Triggers	✗	✓	✗
Supports HTTP Push and SSE Redfish clients	✓	✓	✓
Supported iDRAC Licenses	Express, Enterprise, Datacenter	OMEA Advanced, Datacenter	Express, Enterprise, Datacenter

* -- Partial support.

2.8 Key Links

Telemetry -

- [Telemetry Streaming with iDRAC9 — What you Need to Get Started \(dell.com\)](#)

Redfish legacy events (Alerts) –

- [Implementation of the DMTF Redfish API on Dell EMC PowerEdge Servers](#)
- [Redfish API Reference Guide](#)

Redfish Event Listener –

- <https://github.com/DMTF/Redfish-Event-Listener>

3 Conclusion

Redfish life cycle events provides Dell PowerEdge servers the much-needed solution to capture real-time configuration and inventory changes. With one-to-many solution of collection and streaming workflow, a client would need a solution setup to analyze inventory and configuration changes of the system to adapt and respond to an ever-changing data center management landscape.