DZS Cloud Orchestration version 9.1.0.0.120877

Release Notes

August 2021

DZS AMERICAS
Global Headquarters &
Regional Headquarters
Plano, TX, USA
info@dzsi.com
www.dszi.com/contact-us/

Table of Contents

DZS Cloud Orchestration 9.1 Release Notes

This guide describes the DZS Cloud Orchestration 9.1 release, including new features, fixed and known issues, with their workarounds.

New and Changed Features for 9.1

DZS Cloud Orchestration version 9.1 introduces enhancements to improve management.

Feature	PFRs and JIRAs
CNF Orchestration Enhancements	PFR-679 PFR-674 RIFT-31911 RIFT-31801
Copy or Export a Project Config from one Project to Another Project or Remote Launchpad	PFR-625 PFR-677 RIFT-29545
DZS Cloud Orchestration Disaster Recovery Support Enhancements	PFR-625 RIFT-31604
Geographic Redundancy Scalability Improvements	PFR-663 RIFT-31057 RIFT-31776
LDAP Plugin Support	RIFT-32861
Mapping Service Elements to Existing Resources Improvements	PFR-680 RIFT-31912 RIFT-31482
Network Slicing UI Enhancements	PFR-678 PFR-673 RIFT-31632
Restructured DZS Cloud Orchestration Launchpad UI	PFR-625 PFR-658 RIFT-29268
SOL 001 VNF Indicators Support in VNFD	PFR-652 RIFT-31908
Upgrade Process Changes for a Geographically Paired System	N/A

Feature	PFRs and JIRAs
Validate NS Resource Requirements	PFR-616 RIFT-26179

CNF Orchestration Enhancements

Release 9.1 introduces several enhancements to the process of onboarding and deploying CNFs. This feature also provides support for Helm charts with remote repositories.

Update CNFD from Helm Chart

Prior to release 9.1, Launchpad did not have an option to update the CNFD components such as microservices, connection points, input-parameters, etc while updating a Helm Chart in the CNFD. In previous releases, an operator had to manually update the CNFD or onboard the chart separately and then readd additional components such as monitoring parameters. This release introduces an option to regenerate the impacted components of the CNFD with an RPC or by using the DZS Cloud Orchestration UI.

In previous releases, the values file specified in the CNFD config-data was not used if there was a values files present in the values directory of the descriptor. In release 9.1, the values files specified in the config data is used first when instantiating CNF in a Kubernetes Cluster. If there is no values file specified in the config data, then the values file in the values directory of the CNFD is used to install Launchpad. For details on how to create and edit a values file, see the helm Values Files documentation.

Update a CNFD using a Values File

Release 9.1 introduces the following changes. If a values file (URL) is:

- passed as part of update-cnfd-from-chart RPC invocation, then this value file is used to generate the Helm template and then updated as the values/Values.yaml in CNFD.

  Note: If the values file used for update CNFD RPC contains reference to input variables which are required for Helm template creation, this would fail. If the input variable reference is part of multi-line config, then update CNFD will pass.

- not passed as part of the CNFD and values file is present as /values/Values.yaml for the CNFD, it is used to generate the Helm template.
- not passed or present in /values/Values.yaml, then the values file in the Helm chart is used. This will be updated as /values/Values.yaml in the CNFD.

The values file is not updated with the input-variable or input-parameter values for generating the Helm chart. The tags for all dependencies are set to True to enable all the depend charts in the Helm chart.

Note: If a connection point is removed or renamed in the CNFD and is referred in an NSD which is onboarded in the same project, then the update CNFD RPC fails.

Instantiate a Network Service with CNFs using a Values File

Values File Specified in the CNFD

If a values file is specified in the CNFD config-data and is accessible, then this values file is used and the values file in values directory is ignored. In this case, only the input-parameter-xpath with run time parameters starting with /nsr and /vlr (not all the input-parameter-xpath) are updated in the values file for this CNF.

If there are any entries in the values file with '{{ input-variable.<name> }}', then these values are updated from input variables in the CNFD.

Values File Not Specified in the CNFD

If the values file is not specified in the CNFD config-data and a values file is present in the values directory of the CNFD package, then the latter is used. In this case, all the input-parameter specified in the CNFD are applied to the values file as well as the processing of entries with '{{ input-variable.<name> }}'.

If values file is not specified in CNFD config-data nor present in /values, then instantiation fails.

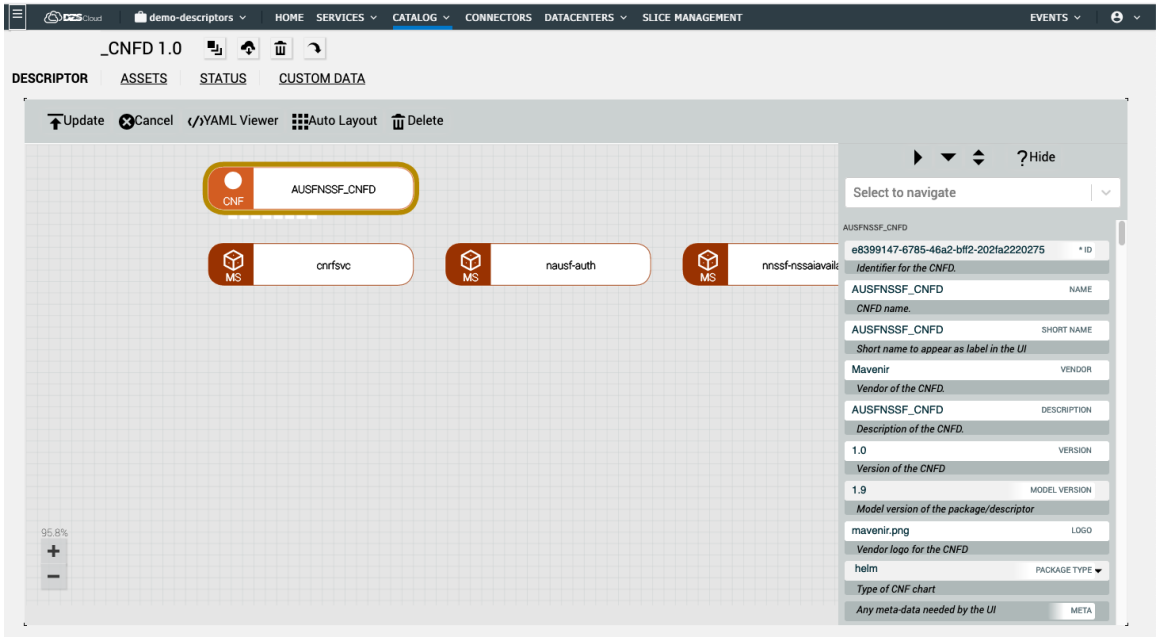
CNF Scaling

If input parameters are marked to scale, then these will also be updated in the values file if the values file is defined in the config-data.

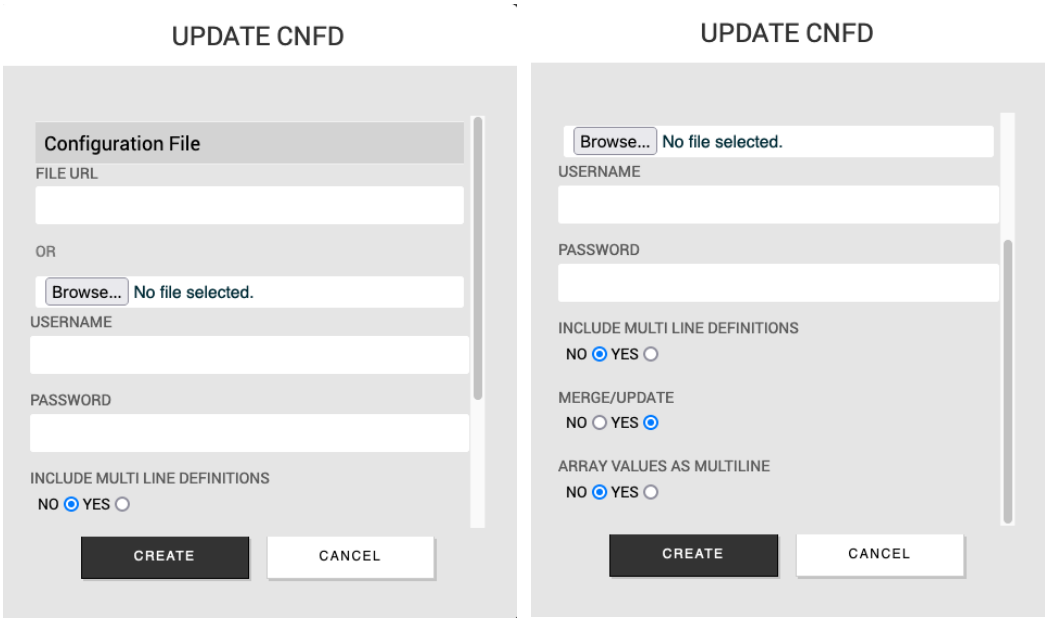
UI Enhancements

In release 9.1, an operator can now update a CNFD option in the UI.

1. On the **Launchpad** NFVO menu, click **CATALOG** > **CNFD LIST** and then click to select the appropriate CNFD descriptor from the catalog. Double-click the descriptor.



2. Click to open the **Update CNFD** wizard and provide the following details:



- **Configuration File**: Configuration file URL. You can either type your URL or browse to the file. An operator can use an external value file (or external URL) for input parameters instead of the ones in the CNFD.
- **USERNAME**: Username if the url uses authentication.
- **PASSWORD**: Password if the url uses authentication.
- **INCLUDE MULTI LINE DEFINITIONS**: NO or YES (Default: NO) Generate input parameters for multi-line value or not.
- **MEREGE/UPDATE**: NO or YES (Default: YES) Option to merge or update new value variables or rebuild the input parameter list. Merge input parameters do not remove existing input parameters. **ARRAY VALUES AS MULTI LINE**: No or YES (Default: NO) If an operator clicks YES, then array inputs are considered multi-line entries instead of individual entries for input parameters.

3. Click **Create**.

Model Changes

RPC Changes

update-cnfd-from-chart RPC

An operator can now update a CNFD using the below RPC.

```
  rpc update-cnfd-from-chart {
    description "Update parts of the CNFD based on the chart referred
in it.";

    input {
      uses manotypes:rpc-project-name;
```

```
      leaf cnfd-id {
        description "ID of the CNFD to update";
        type leafref {
          path "/rw-project:project[rw-project:name=current()/../
          project-name]/" +
          "project-cnfd:cnfd-catalog/project-cnfd:cnfd/project-
           cnfd:id";
        }
      }

      uses update-cnfd-params;
    }

    output {
      uses task-id;

      uses manotypes:nats-rpc-output;
    }
  }
}
}
  }
```

cnfd:http-endpoint:port

Prior to release 9.1, an operator had to add a monitoring server service port when configuring monitoring server monitoring parameters in a CNFD (normally it is port 30000). If the value did not match the cluster configuration, the monitoring parameters are not displayed or loaded after instantiation.

Since this port is a cluster and datacenter level attribute, it should be defined in that datacenter and not in the CNFD. If the port is empty in the CNFD:http-endpoint, then the value for the port is taken from the monitoring server configured in the datacenter.

Note: This presumes that all NFs launched on that datacenter and the Launchpad managing them are using a common monitoring server instance.

Support for Custom Network Configuration using a Multus CNI plugin

Release 8.3.1 introduced support for networks created by the Multus CNI (containerized network interface) plugins to support the static IPAM plugin. The Multus CNI plugin enables attaching multiple network interfaces to pods for Kubernetes using different plugins. Release 9.1 introduces support for passing a CNI plugin config directly so that the new plugins can be supported.

For additional information on the cnitool and the multi CNI plugin, see:

- **CNI**: https://www.cni.dev/docs/cnitool/
- **Multus CNI**: https://github.com/intel/multus-cni

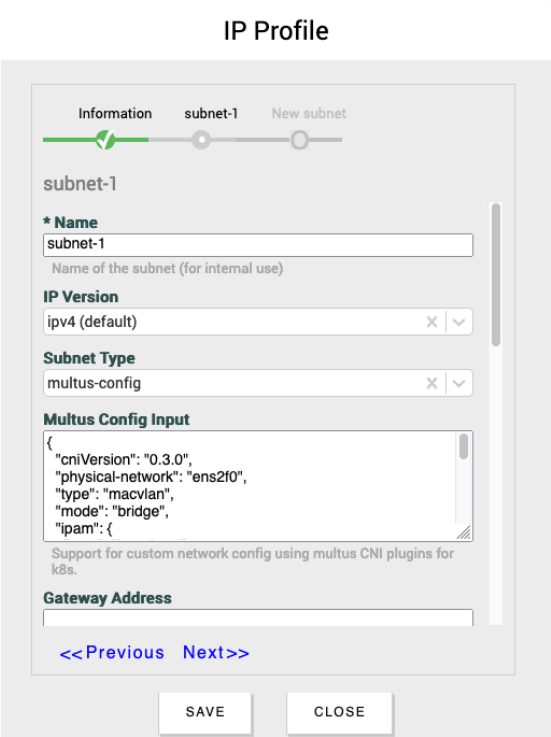
UI Enhancements

An operator can now add a multus-config plugin to ip profiles. Refer to Profile Configuration for details on how to set up IP Profiles.

IP PROFILES

In the **Subnet Type** field, an operator can now choose multus-config.

1. Select multus-config and then an additional **Multus Config Input** field appears.
   - **Subnet Type**: Choose multus-config.
   - **Multus Config Input**: Type the multus config input. This is the support for custom network config using multus CNI plugins for k8s.
2. Click **SAVE**.



Model Changes

This release introduces a new multus-config-input field in the ip-profile definition in the rw-cloud:subnet in the RW Project Data Model.

- `mutlus-config-input`: Support for custom network config using multus CNI plugins for k8s.

> **Note:** If the multus-config-input field is populated, then the other field options are ignored.

Post Validation of a Deployed CNF

In release 9.1, an operator can verify that a CNF is deployed correctly. Launchpad only updates the values passed to the Helm Chart for deployment. This feature now validates if the values passed for deployment are the same values that are in the deployment by using 'helm get values <release-name>'. If any differences are found, then a warning event is generated. See the example below.

This warning will not update the CNF record status to Failed.

Warning events


```
2021-04-20 12:49:39,269 ERROR {"metadata": {"event_time": "2021-04-20
12:49:39.269456", "message": "Found item changes:
{'dictionary_item_added': [root['key1']],\n 'dictionary_item_removed':
[root['global'], root['macvlanNetName'], root['macvlanNetNamespace'],
root['metrics'], root['service']]}", "event_module": "lcm-worker-log",
"event_name": "internal-error", "filename": "compare_desc.py",
"linenumber": 86, "severity": "error", "crit_info": "true", "version":
"9.1.0.0.5", "system_id": "b554c694-5c3f-43b7-82a0-7137cb7d47b3",
"description": "Internal Error"}, "attributes": {"worker_id": "lcm-
worker-001", "component_name": "RW.LcmWorker.001", "cnfr_id":
"c67ba622-f3e0-4011-a198-a7c8f00934e4", "cnfr_name": "de-pj--pong-
zw4wewktc-0-0-1", "cnfd_id": "e35db28a-94b4-4500-8950-e5e6dccd229a",
"cnfd_name": "pong_cnfd", "datacenter": "bm4", "nsr_id": "c8b92cb4-
23a8-44da-8a34-fbc5b2e0c572", "project_name": "default", "nsr_name":
"pj-pp", "user_id": "admin@system", "worker_name": "RW.LcmWorker.001",
"worker_type": "lcm-worker"}, "debug_details": []}
2021-04-20 12:49:39,270 WARNING {"metadata": {"event_time": "2021-04-20
12:49:39.270109", "message": "Error comparing provided and current
values for release", "event_module": "lcm-worker-log", "event_name":
"generic-warning", "filename": "rwnsm_cnfr.py", "linenumber": 1263,
"severity": "warning", "crit_info": "true", "version": "9.1.0.0.5",
"system_id": "b554c694-5c3f-43b7-82a0-7137cb7d47b3"}, "attributes":
{"worker_id": "lcm-worker-001", "component_name": "RW.LcmWorker.001",
"cnfr_id": "c67ba622-f3e0-4011-a198-a7c8f00934e4", "cnfr_name": "de-pj-
-pong-zw4wewktc-0-0-1", "cnfd_id": "e35db28a-94b4-4500-8950-
e5e6dccd229a", "cnfd_name": "pong_cnfd", "datacenter": "bm4", "nsr_id":
"c8b92cb4-23a8-44da-8a34-fbc5b2e0c572", "project_name": "default",
"nsr_name": "pj-pp", "user_id": "admin@system", "worker_name":
"RW.LcmWorker.001", "worker_type": "lcm-worker"}, "debug_details":
[{"type": "rw-log:python-exception", "details": ["Traceback (most
recent call last):\n", "  File
\"/localdisk/pjoseph/rel910/.build/ub18_debug/install/usr/rift/usr/lib6
4/python3.6/site-packages/rift/nsms/rwnsm_riftnsm/rwnsm_cnfr.py\", line
1258, in compare_values\n    CompareDescShell.compare_dicts(cur_values,
exp_values, rwpylog=self.rwpylog)\n", "  File
\"/localdisk/pjoseph/rel910/.build/ub18_debug/install/usr/rift/usr/lib6
4/python3.6/site-packages/rift/mano/utils/compare_desc.py\", line 87,
in compare_dicts\n    raise ValueError(msg)\n", "ValueError: Found item
changes: {'dictionary_item_added': [root['key1']],\n
'dictionary_item_removed': [root['global'], root['macvlanNetName'],
root['macvlanNetNamespace'], root['metrics'], root['service']]}\n"]},
{"type": "rw-log:python-exception", "details": ["Traceback (most recent
```

```
call last):\n", "  File
\"/localdisk/pjoseph/rel910/.build/ub18_debug/install/usr/rift/usr/lib6
4/python3.6/site-packages/rift/nsms/rwnsm_riftnsm/rwnsm_cnfr.py\", line
1258, in compare_values\n    CompareDescShell.compare_dicts(cur_values,
exp_values, rwpylog=self.rwpylog)\n", "  File
\"/localdisk/pjoseph/rel910/.build/ub18_debug/install/usr/rift/usr/lib6
4/python3.6/site-packages/rift/mano/utils/compare_desc.py\", line 87,
in compare_dicts\n    raise ValueError(msg)\n", "ValueError: Found item
changes: {'dictionary_item_added': [root['key1']],\n
'dictionary_item_removed': [root['global'], root['macvlanNetName'],
root['macvlanNetNamespace'], root['metrics'], root['service']]}\n"]}]}
2021-04-20 12:49:39,270 WARNING {"metadata": {"event_time": "2021-04-20
12:49:39.270667", "message": "The values used for in release do not
match the provided values", "event_module": "lcm-worker-log",
"event_name": "generic-warning", "filename": "rwnsm_cnfr.py",
"linenumber": 1523, "severity": "warning", "crit_info": "true",
"version": "9.1.0.0.5", "system_id": "b554c694-5c3f-43b7-82a0-
7137cb7d47b3"}, "attributes": {"worker_id": "lcm-worker-001",
"component_name": "RW.LcmWorker.001", "cnfr_id": "c67ba622-f3e0-4011-
a198-a7c8f00934e4", "cnfr_name": "de-pj--pong-zw4wewktc-0-0-1",
"cnfd_id": "e35db28a-94b4-4500-8950-e5e6dccd229a", "cnfd_name":
"pong_cnfd", "datacenter": "bm4", "nsr_id": "c8b92cb4-23a8-44da-8a34-
fbc5b2e0c572", "project_name": "default", "nsr_name": "pj-pp",
"user_id": "admin@system", "worker_name": "RW.LcmWorker.001",
"worker_type": "lcm-worker"}, "debug_details": []
```


Pre instantiation with Config and Service Primitives

Some the Helm charts require the installation of a set of Custom Resource Definitions (CRDs) prior to instantiation. In release 9.1, this is handled in the Helm archive with a specific script and by introducing a new set of pre instantiation CNF or VNF config or service primitives.

UI Enhancements

Service primitives can now be configured in the NSD under pre-instantiation-service-primitive. Pre instantiation config or service primitives will run before CNF or VNF instantiation. If the CNF or VNF fails during instantiation due to the below operations, then an operator can retry the instantiation.

Config primitive failure

- cnfr/vnfr opstatus: **init**
- cnfr/vnfr config status: **failed**
- NSR opstatus: **failed if part of instantiation**
- NSR config status: **failed**

Service primitive failure

- NSR opstatus: **init**
- NSR config status: **failed**

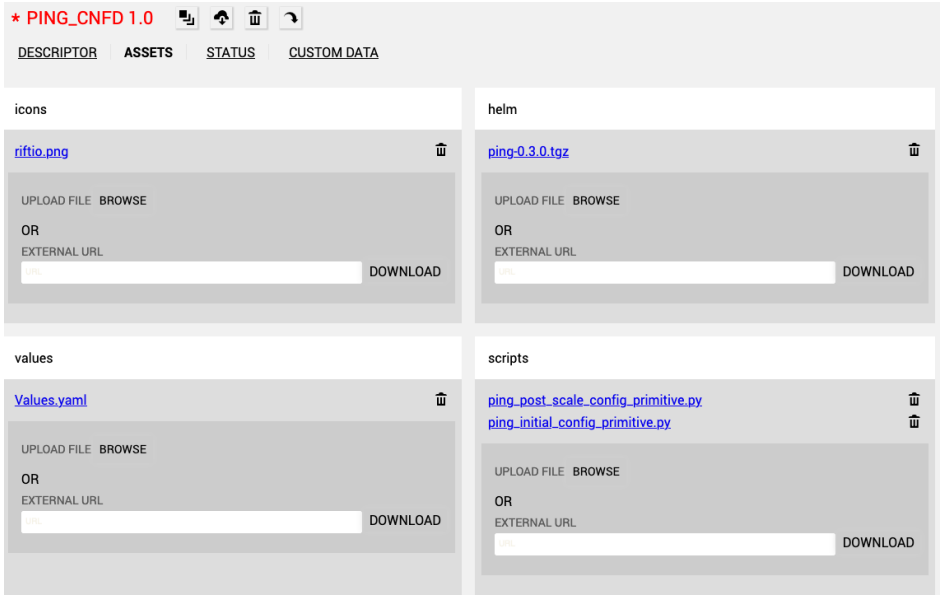
Configure Config Primitives

Config primitives can be configured in the CNFD or VNFD under pre-instantiation-config-primitive.

1. On the **Launchpad** menu, click **CATALOG > CNFD LIST** and select the appropriate CNFD descriptor from the catalog. Double-click the descriptor.

   The **PING_CNFD** or **PONG_CNFD** screen opens.

2. Click the **ASSETS** tab.
3. In the scripts card, upload the script or add the external URL.



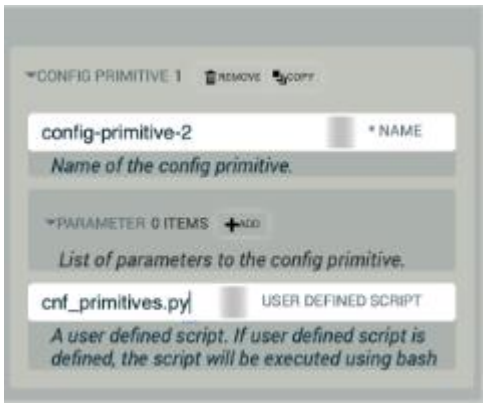
4. Return to the **DESCRIPTOR** tab and add the details in the descriptor section:

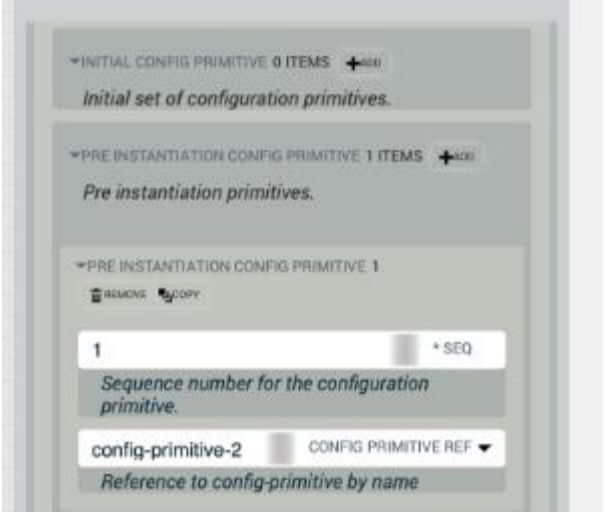
   CONFIG PRIMITIVE

   - **NAME:** Name of the config primitive.

   PARAMETER: List of parameters of the config primitive.

   - **USER DEFINED SCRIPT:** A user defined script. If a user script is defined, then a script will be executed using bash.





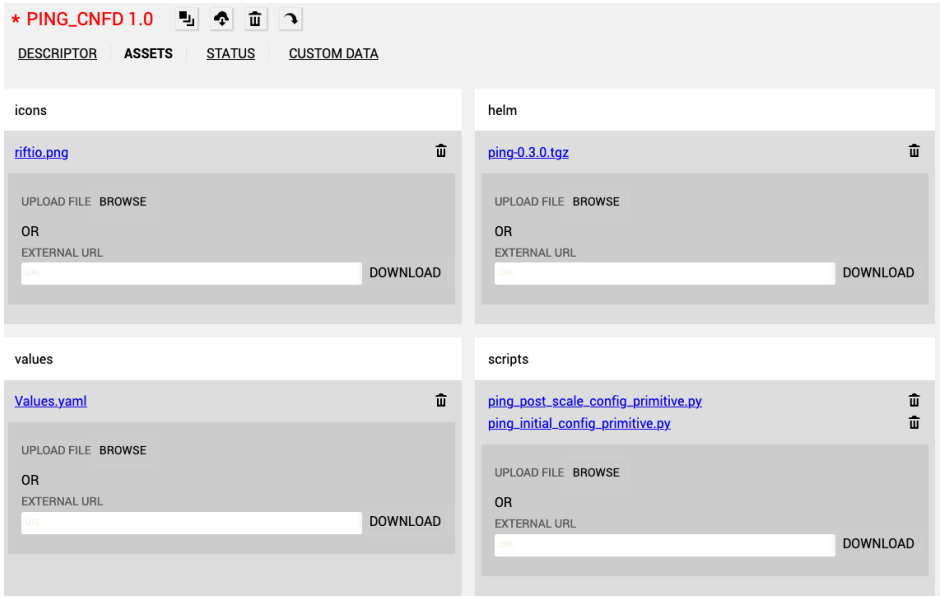
5. Click Update.

Configure Service Primitives

Service Primitive can be configured in the NSD under pre-instantiation-service-primitive,

1. On the **Launchpad** menu, click **CATALOG > CNFD LIST** and select the appropriate CNFD descriptor from the catalog. Double-click the descriptor.

   The **PING_CNFD** or **PONG_CNFD** screen opens.
2. Click the **ASSETS** tab.
3. In the scripts card, upload the script or add the external URL.



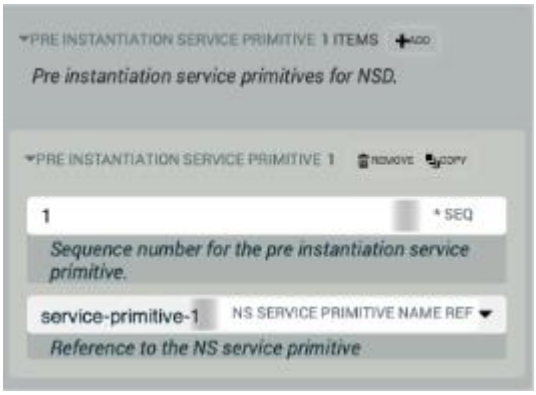
4. Return to the **DESCRIPTOR** tab and add the details in the descriptor section:

   SERVICE PRIMITIVE: Network service level service primitives

- **NAME:** Name of the service primitive.

PRE INSTANTIATION SERVICE PRIMITIVE: Pre instantiation service primitives for NSD.

- **SEQ:** Sequence number for the pre instantiation service primitive.
- **NS SERVICE PRIMITIVE NAME REF:** Reference to the NS service primitive.



5. Click Update.

Model Changes

This release introduces new pre-instantiation-service-primitive and pre-instantiation-config-primitive fields in the NSD Data Model, CNFD Data Model and VNFD Data Model.

NSD Data Model

- `seq`: Sequence number for the pre instantiation service primitive.
- `ns-service-primitive-name-ref`: Reference to the NS service primitive

CNFD Data Model and VNFD Data Model

- `seq`: Sequence number for the configuration primitive.
- `config-primitive-ref`: Reference to config-primitive by name.

Copy or Export a Project Config from one Project to Another Project or Remote Launchpad

This feature allows a super admin operator to copy or export a configuration from one project to another project or a remote Launchpad. The supported config elements are:

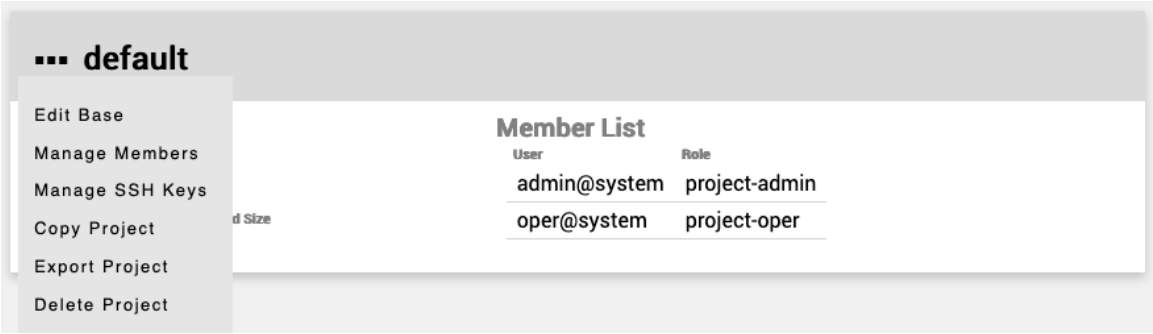
- Packages
- Cloud accounts
- Connector accounts
- SSH Keys

This feature includes the following UI Enhancements, RPC Model Changes and API Changes.

UI Enhancements

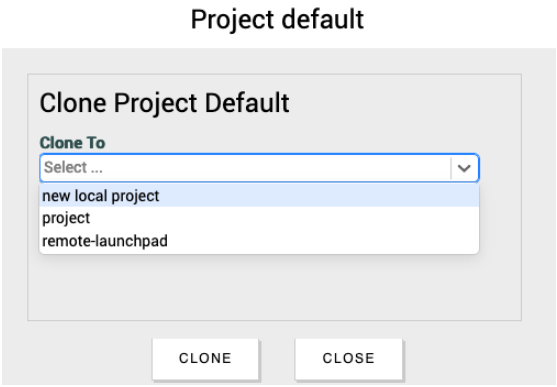
Choose **LAUNCHPAD** > **ADMIN-TOOLS** to display the projects tab. See Restructured DZS Cloud Orchestration Launchpad UI for changes in release 9.1.0.

Click ▪▪▪ next to a project to display the available menu options. In the list, an operator can choose Clone Project or Export Project.



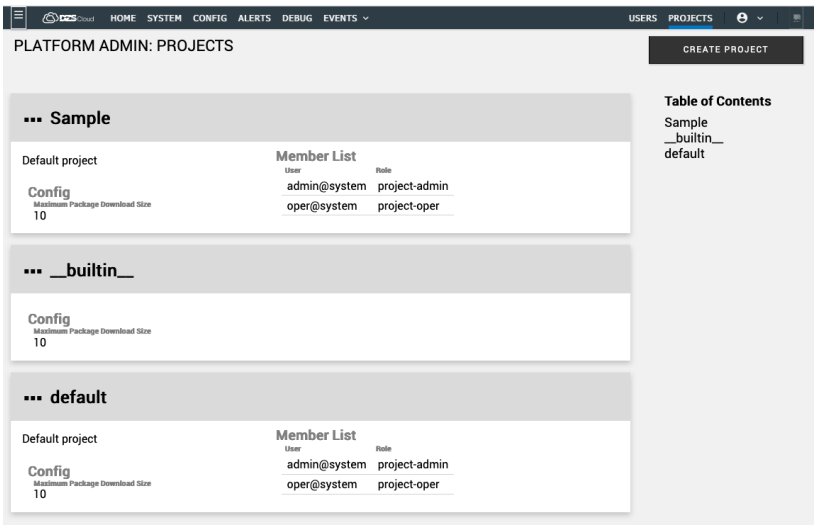
Clone Project

1. Click **Clone Project** and then choose where to copy the project to.



2. In the **Clone To** drop-down menu, select where you want to clone your project to:
    - **new local project**
    - **project**
    - **remote-launchpad**
3. If you select new local project or project, then a ***Local Project** drop-down menu appears. Select the Name of the project and proceed to step 5.
4. If you select remote-launchpad, then provide the following information for that Launchpad:
    - ***Url**: URL to the remote launchpad
    - ***Remote Project**: Name of the project - If the project does not exist, a new project with the given name will be created.
    - ***Remote User**: User name. This field is auto prefilled with admin.
    - ***Password**: This field is auto prefilled with the password.
5. Click **CLONE**.

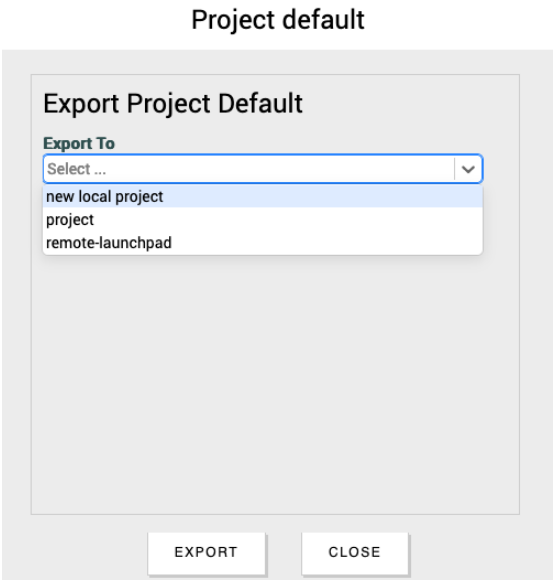
The below screen shot shows a copy of the default project to a new local project (Sample).



Export Project

Note: When exporting a package NSD, the constituent VNFDs and CNFDS are also exported.

1. Click **Export Project** and then choose where to export the project to.



2. In the **Export To** drop-down menu, select where you want to export your project to:
    - **new local project**
    - **project**
    - **remote-launchpad**
3. If you select new local project or project, then provide the following information for that project and then proceed to step 5.

Note: In some cases, entries will not be exported for accounts and keypairs.

- ***Local Project**: Name of the new project to be created.
- **Export Key Pairs**: Export all key-pairs in a project (True or False)
- **Export Packages**: Export packages in a project (True or False)
- **Export Cloud Accounts**: Export Cloud Accounts in a project (True or False)
- **Export Connectors**: Export Connectors in a project (True or False)
- **Export Csmf**: Export CSMF in a project (True or False)

Note: The **Export Csmf** config element is not supported in 9.1.0.

4. If you select remote-launchpad, then provide the following information for that Launchpad:
   - ***Url**: URL to the remote launchpad
   - ***Remote Project**: Name of the project - If the project does not exist, a new project with the given name will be created.
   - ***Remote User**: User name. This field is auto prefilled with admin.
   - ***Password**: Password. This field is auto prefilled with the password.
   - **Export Key Pairs**: Export all key-pairs in a project. (True or False)
   - **Export Packages**: Export packages in a project. (True or False)
   - **Export Cloud Accounts**: Export Cloud Accounts in a project. (True or False)
   - **Export Connectors**: Export Connectors in a project. (True or False)
   - **Export Csmf**: Export CSMF in a project. (True or False)

   Note: The **Export Csmf** config element is not supported in 9.1.0.

5. Click **EXPORT**.

Note: If a target project is not created, then the package export will fail. If an operator selects a:

New local project: Then an attempt is made to create project and that fails. Therefore, the export also fails.

Remote Launchpad: and the operator enters a remote project name but the project does not exist, then an attempt is made to create it. If that attempt fails, then the export also fails.

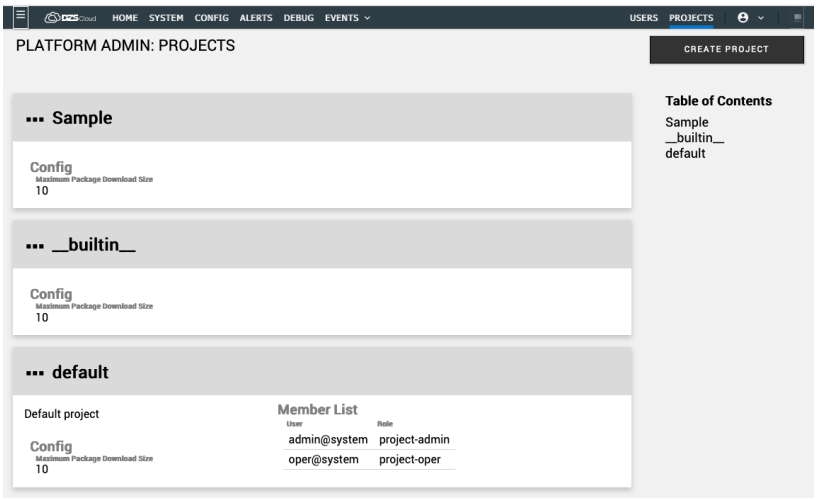
6. If a target project is not created in the below scenario, then the package export will fail. If an operator selects a:
   - **New local project**: An attempt is made to create project and that fails, then the export also fails.

- **Remote Launchpad**: The operator enters a remote project name. If the project does not exist, then an attempt is made to create it. If that attempt fails, then the export also fails.

The below screen shot shows an export of the default project to a new local project (Sample). The following are all marked True in the exported project:

- **Export Key Pairs**
- **Export Packages**
- **Export Cloud Accounts**
- **Export Connectors**
- **Export Csmf**

> **Note:** The **Export Csmf** config element is not supported in 9.1.0.



RPC Model Changes

The RPC definition is added in IM submodule. There is a new file mano/rw-export-config.yang change for this feature.

API Examples

{REST_HOST}/api/operations/export-project-config API is used for bulk copying config. The following are the JSON PDU examples used in the POST request.

Export all supported config to a local project

```
{
    "input": {
        "project-name": "default",
        "local-project": "local_project",
        "export-all": true
    }
}
```

Export all supported config to a remote project

```
{
    "input": {
        "project-name": "default",
        "url": "https://10.64.210.31:8008",
        "remote-project": "remote_project",
        "remote-user": "admin",
        "password": "admin",
        "export-all": true
    }
}
```

Export selected config to local project

```
{
    "input": {
        "project-name": "default",
        "local-project": "local_project",
        "export-packages": {
            "export-packages": {
                "vnfd": [
                    {"vnfd-id": "024a1b5c-2ad4-11eb-80af-
02420a40d20a"},
                    {"vnfd-id": "024aa248-2ad4-11eb-80af-02420a40d20a"}
                ],
                "nsd": [
                    {"nsd-id": "024aec8a-2ad4-11eb-80af-02420a40d20a"}
                ]
            }
        },
        "export-key-pairs": {
            "key-name": [{
                "key-name": "key-1"
            }]
        },
        "export-cloud-accounts": {
            "cloud-account": [{
                "cloud-account": "smarakka"
            }]
        },
        "export-config-agent-accounts": {
            "cfg-agent-account": [{
                "cfg-agent-account": "cfg1"
            }]
        },
        "export-misc-connectors": {
            "misc-conn-account": [{
                "misc-conn-account": "misc1"
            }]
        },
        "export-oss-bss-accounts": {
            "oss-bss-account": [{
                "oss-bss-account": "ossbss1"
            }]
        },
        "export-remote-lp-accounts": {
```

```
            "remote-lp-account": [{
                "remote-lp-account": "rlp1"
            }]
        },
        "export-ro-accounts": {
            "ro-account": [{
                "ro-account": "ro1"
            }]
        },
        "export-sdn-accounts": {
            "sdn-account": [{
                "sdn-account": "sdn1"
            }]
        },
        "export-vnfm-accounts": {
            "vnfm-account": [{
                "vnfm-account": "vnfm1"
            }]
        }
    }
}
```

Export selected config to remote project

```
{
    "input": {
        "project-name": "default",
        "url": "https://10.64.210.31:8008",
        "remote-project": "remote_project",
        "remote-user": "admin",
        "password": "admin",
        "export-packages": {
            "export-packages": {
                "vnfd": [
                    {"vnfd-id": "024a1b5c-2ad4-11eb-80af-
02420a40d20a"},
                    {"vnfd-id": "024aa248-2ad4-11eb-80af-02420a40d20a"}
                ],
                "nsd": [
                    {"nsd-id": "024aec8a-2ad4-11eb-80af-02420a40d20a"}
                ]
            }
        },
        "export-cloud-accounts": {
            "cloud-account": [{
                "cloud-account": "smarakka_bm4"
            }]
        },
        "export-key-pairs": {
            "key-name": [{
                "key-name": "key-1"
            }]
        },
        "export-config-agent-accounts": {
            "cfg-agent-account": [{
                "cfg-agent-account": "cfg2"
            },
```

```
            {
                "cfg-agent-account": "cfg3"
            }]
        },
        "export-misc-connectors": {
            "misc-conn-account": [{
                "misc-conn-account": "misc2"
            }]
        },
        "export-oss-bss-accounts": {
            "oss-bss-account": [{
                "oss-bss-account": "ossbss2"
            }]
        },
        "export-remote-lp-accounts": {
            "remote-lp-account": [{
                "remote-lp-account": "rlp2"
            }]
        },
        "export-ro-accounts": {
            "ro-account": [{
                "ro-account": "ro2"
            }]
        },
        "export-sdn-accounts": {
            "sdn-account": [{
                "sdn-account": "sdn2"
            }]
        },
        "export-vnfm-accounts": {
            "vnfm-account": [{
                "vnfm-account": "vnfm2"
            }]
        }
    }
}
```

Export All of packages, cloud & VNFM accounts and key pairs to local project

```
{
    "input": {
        "project-name": "default",
        "local-project": "local_project",
        "export-packages": {
            "export-all": true
        },
        "export-key-pairs": {
            "export-all": true
        },
        "export-cloud-accounts": {
            "export-all": true
        },
        "export-vnfm-accounts": {
            "export-all": true
        }
    }
}
```

Export project level config

When exporting selectively from project config, the following fields can be populated:

- **export-project-config**: Used to export top level project configuration. Currently handles only "rw-package:maximum-package-download-size"
- **export-user-roles**: Can be exported as a whole. or selectively.
- **export-event-map-list**: Can be exported as a whole. or selectively.

In the case that a user or umb account is not configured in the remote lp, then the corresponding entry in the export request will be skipped.

```
{
    "input": {
        "project-name": "source",
        "export-project-config": true,
        "url": "https://10.64.210.6:8008",
        "remote-project": "destination_kafka_selection_4",
        "remote-user": "admin",
        "password": "admin",
        "export-user-roles": {
            "export-all": true
        },
        "export-event-map-list": {
            "export-all": true
        }
    }
}


{
    "input": {
        "project-name": "source",
        "export-project-config": true,
        "url": "https://10.64.210.6:8008",
        "remote-project": "destination_kafka_selection_4",
        "remote-user": "admin",
        "password": "admin",
        "export-event-map-list": {
            "event-topic-map": [{
                "event": "rw-umb-mgr:ns-instance-change"
            }, {
                "event": "rw-umb-mgr:vnfd-change"
            }]
        }
    }
}


{
    "input": {
        "project-name": "source",
        "export-project-config": false,
        "url": "https://10.64.210.6:8008",
```

```
        "remote-project": "destination",
        "remote-user": "admin",
        "password": "admin",
        "export-user-roles": {
            "user": [{
                "user-name": "admin",
                "user-domain": "system"
            }, {
                "user-name": "ami",
                "user-domain": "systemm"
            }, {
                "user-name": "amia",
                "user-domain": "system"
            }, {
                "user-name": "ambu",
                "user-domain": "system"
            }, {
                "user-name": "amia",
                "user-domain": "systemm"
            }, {
                "user-name": "ami",
                "user-domain": "system"
            }]
        }
    }
}
```

DZS Cloud Orchestration Disaster Recovery Support Enhancements

Release 9.1 introduces enhancements to the DZS Cloud Orchestration disaster recovery process. In this release, the UI now displays the progress status of an export job. An operator can now also download a log file that collects the exported or imported related logs. The progress percentage and the link for the file are available as part of launchpad-export-state (Operational Data).

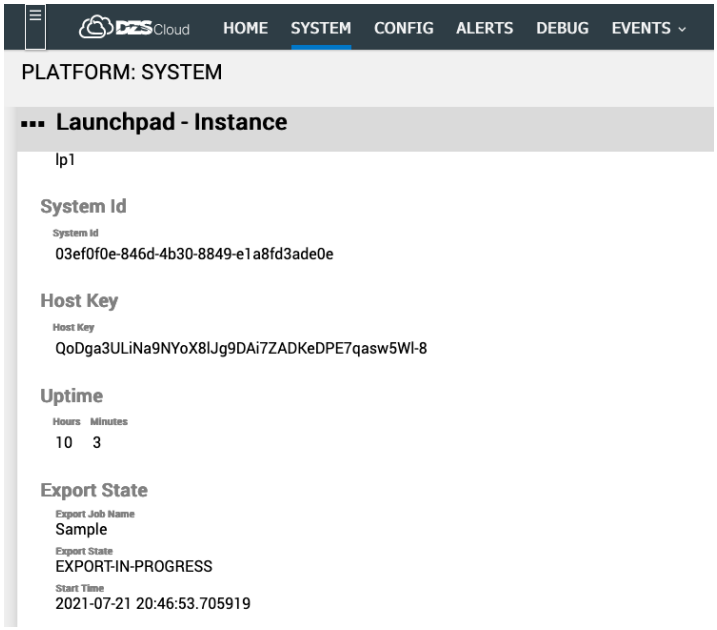
Prior to release 9.1.0, there was a limitation when importing an exported archive. The important Launchpad release needed to match the archive release version. In release 9.1.0, an operator can now import an archived file to an upgrade launchpad release version.

UI Changes

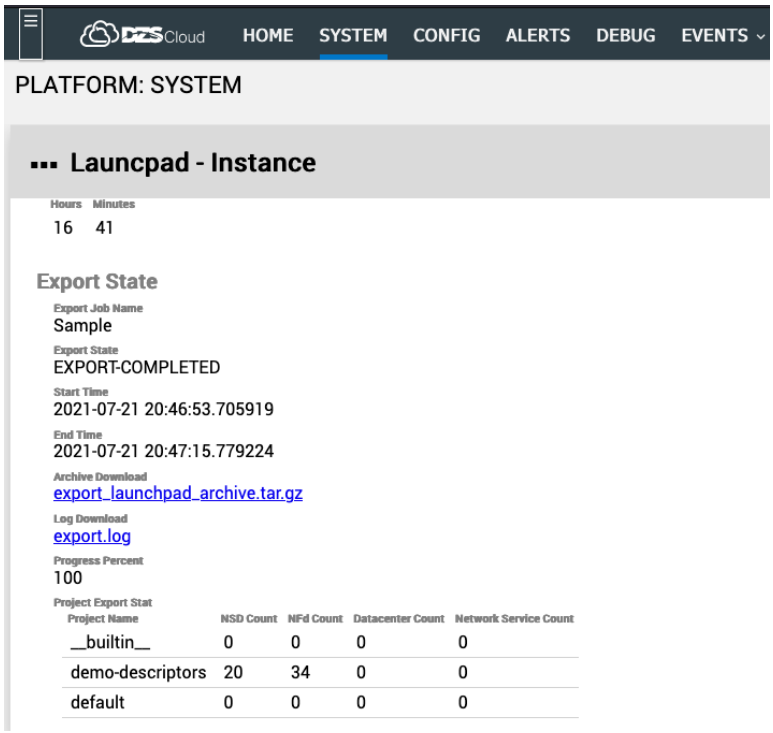
1. On the **Launchpad** Dashboard, click **PLATFORM** > **SYSTEM**.
2. On the **Launchpad Instance** card, click **Export Launchpad State** card to export an archive file that can be used to recover from a system down condition.

> **Note:** An operator must be a super user with super-admin roles to open this URL.

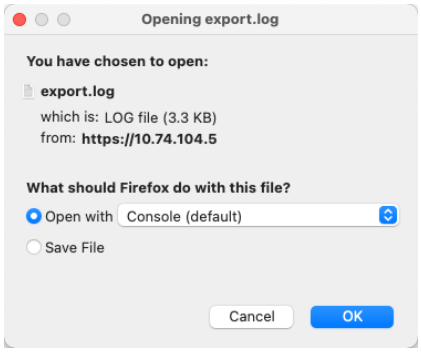
The UI now displays the Export State. In the example below, the status is **EXPORT-IN-PROGRESS**.



Once the export is complete, the **Launchpad – Instance** screen includes a link to download the export log. In the example below, the Log Download file is export.log. The UI now displays additional statistics related to the export archive. The metrics include the NSD Count, NFd Count, Datacenter Count and Network Service Count for each export project.

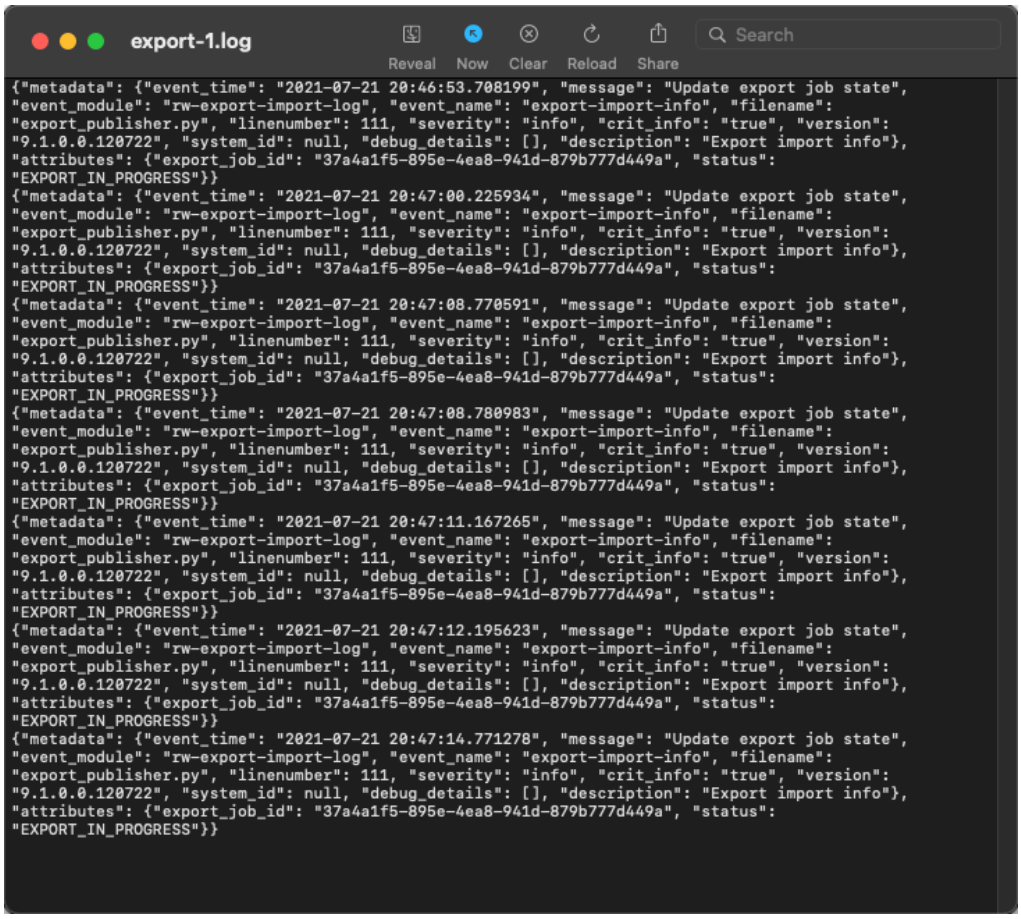


3. Click export.log.



An operator can either save the file or open it with a terminal console.

The example below shows an operator choosing to open the log with the console terminal.



SNMP Traps

DZS Cloud Orchestration now sends alerts as SNMPv3 traps when an export job is successfully completed or fails.

- riftLPExportLPSuccess: Trap notification for successfully completed export job.
- riftLPExportLPFail: Trap notification for failed export job.

Trap Name	OID	Severity	NOC Action	Additional Fields	Description and action to be taken
riftLPExportLPSuccess	1.3.6.1.4.1.53030.1.2.57	Informational	No Action	riftSeverity riftMessage	Trap notification for successfully completed launchpad export job.
riftLPExportLPFail	1.3.6.1.4.1.53030.1.2.58	Informational	Contact Operations	riftSeverity riftMessage	Trap notification for failed launchpad export job.

Model Changes

The following API checks the status of the export and it now includes checks for the progress percentage and the link for an export/import file. This API existed in previous releases but the changes in bold are active in this release.

launchpad-export-state (Operational Data)

```
   list launchpad-export-state {
     description "Launchpad export job details and status";
     config false;
     key "export-job-id";
     leaf export-job-id {
       description "Unique Id created for the export job";
       type yang:uuid;
     }
     leaf export-job-name {
       description "The export job name";
       type string;
     }
     leaf export-metadata {
       description "The metadata added by user to identify or define a
particular export job";
       type string;
     }
     leaf export-state {
       description "Provides the current state of export job";
       type enumeration {
         enum NONE;
         enum EXPORT-IN-PROGRESS;
         enum EXPORT-COMPLETED;
         enum EXPORT-CANCELLED;
         enum EXPORT-ERROR;
       }
     }
     leaf start-time {
       description "Start time of launchpad export job";
       type rwt:date-and-time;
     }
```

```
    leaf end-time {
      description "Finish time of launchpad export job";
      type rwt:date-and-time;
    }
    leaf exported-launchpad-version {
      description "Launchpad version and build details during export";
      type string;
    }
    leaf current-state-detail {
      description "Detailed error description or current job state
detail.";
      type string;
    }
    leaf archive-download {
      description "Fully qualified filesystem path of the archive";
      type string;
    }
    leaf log-download {
      description "Fully qualified filesystem path to download export
logs";
      type string;
    }
    leaf progress-percent {
      description "Progress of data export { percentage }";
      type uint16 {
        range "0 .. 100";
      }
    }
    list project-export-stat {
      description "Project level statistics during export";
      key project-name;
      leaf project-name {
        description "Name of the project";
        type string {
          length "1..255";
        }
      }
      leaf nsd-count {
        description "Number of nsd's configured in the project";
        type uint32;
      }
      leaf nfd-count {
        description "Number of nfd's configured in the project";
        type uint32;
      }
      leaf datacenter-count {
        description "Number of datacenters configured in the project";
        type uint32;
      }
      leaf network-service-count {
        description "Number of network services running in the
project";
        type uint32;
      }
    }
  }
```

Geographic Redundancy Scalability Improvements

This release introduces improvements to the time it takes to recover a large number of Network Service (NS) instances to reload during a Geographic Redundancy (GR) switchover or a Launchpad restart. Prior to release 9.1, it took a while to recover all the NS instances in a longevity setup. An NS is now lazily restored on an external event from the VIM or the Network Service. A recovery process begins due to the following external events:

- A user requested action to either reload the operation or through Lifecycle Management Action on a Network Service.
- VIM alarm
- Policy Action
- Background Restore Task

Geographic Redundancy Scalability Improvements include:

- UI Enhancements
  - Start a Network Service Restore
  - View the Restore Status in the UI
  - Configure High Availability Enhancements
- Model Changes

UI Enhancements

A user can now initiate a NS restore explicitly by clicking on the new restore-ns action from UI. An NS is also restored implicitly by performing any LCM action on an NS, this occurs automatically without any manual input from the user. The UI now also displays the count of Network Services that are being restored after a system restart or failover. This information can appears on the Launchpad Dashboard.

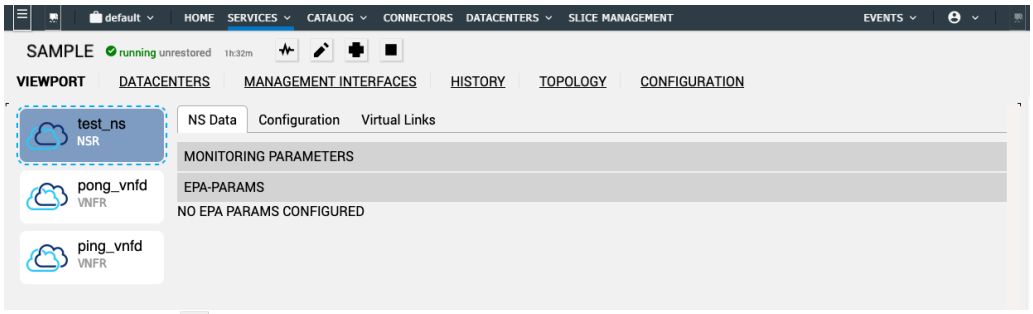
The UI now displays the recovery status of an NS instance. The UI shows the restore status in the Viewport tab.

- Start a Network System Restore
- View the Restore Status in the UI
- Configure High Availability Enhancements

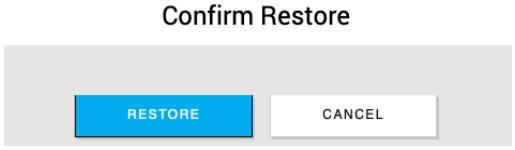
Start a Network Service Restore

This feature allows an operator to restore a Network Service explicitly from UI. The status of the Network Service restore progress appears on the Viewport screen.

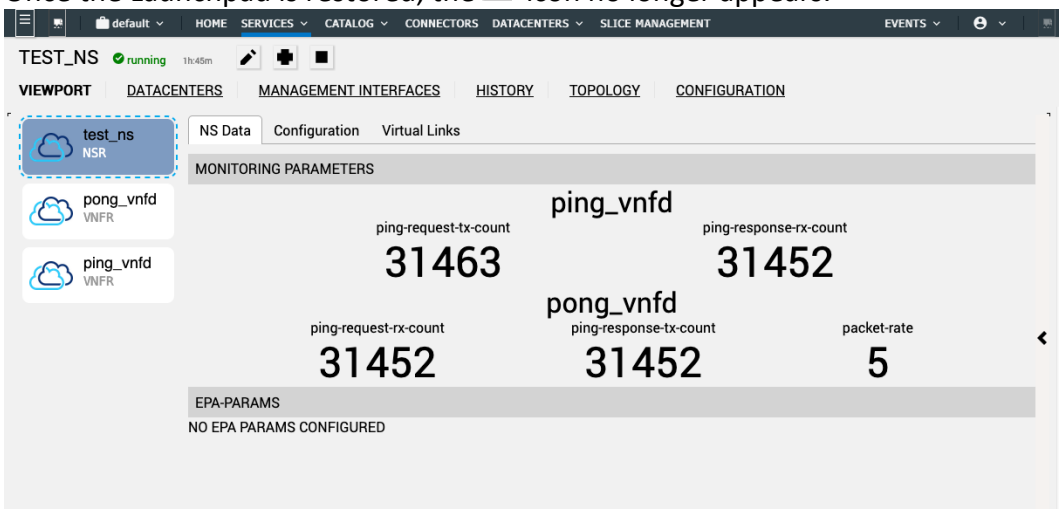
1. On the Launchpad menu, click **SERVICES** > **SERVICE LIST** to open the list of NS instances.
2. Click next to the NS that you want to see in more details. The **VIEWPORT** screen opens.
3. If Launchpad is in the unrestored state, then click to restore the Launchpad.



4. After clicking , a confirmation box appears. Click **RESTORE** and a progress window appears.



5. Once the Launchpad is restored, the icon no longer appears.

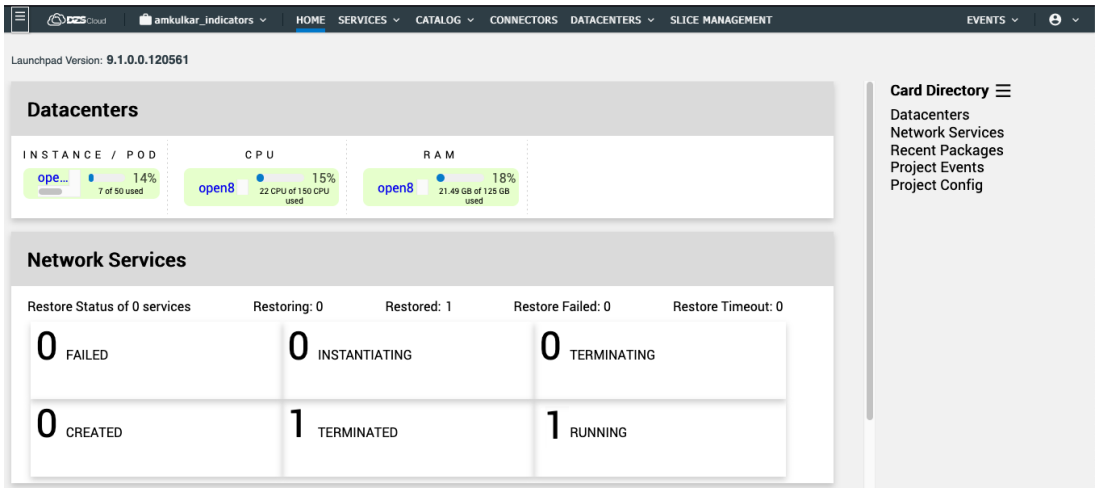


View the Restore Status in the UI

An operator can view the status of a system restore in the UI.

Choose the **NFVO** view in the App Menu and click the **HOME** tab. The Network Services card displays the following restore metrics:

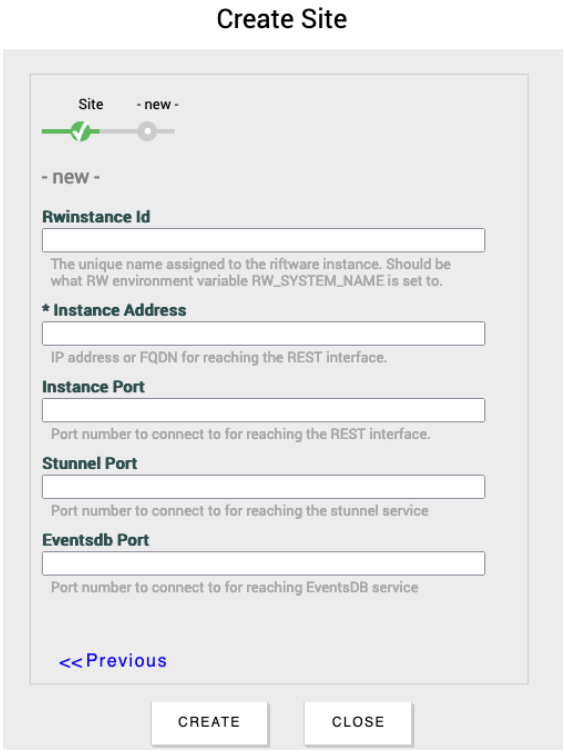
- Restore Status of ‘x’ services
- Restoring: ‘x’
- Restored: ‘x’
- Restore Failed: ‘x’
- Restore Timeout: ‘x’



Configure High Availability Enhancements

This feature include changes to `redundancy-config` and `redundancy-state` to reflect the correct usage of the floating-ip field. Previously, the floating-ip node was under `/redundancy-config/site/rw-instances` and `/redundancy-state/config-state` xpaths. Release 9.1.0 introduces four new nodes: instance-address, instance-port, stunnel port and Eventsdb Port. This change allows an operator to pair a site using a non default REST port. See Model Changes for more information.

1. On the Launchpad menu, click **ADMIN-TOOLS**> **SYSTEM** to add or manage at site.
2. On the **Redundancy** card, click ▪▪▪ and choose **Add site** or **Manage 'siteX'**.
3. On the **Create Site** or **Update siteX** window, provide details for the site (fields with * are required).
   - *** Site Name**: The name of the redundancy site. Set the name to the value set for **SYSTEM NAME** configured in the **Set up a System Name for each Launchpad** section in Configure DZS Cloud Orchestration for Geographic Redundancy.
   - **Site ID**: The ID of the site.



4. Click **Next >>** and provide the instance-address and instance-port site details.
   - *** Instance Address**: IP address or FQDN for reaching the REST interface.
   - **Instance Port**: Port number to connect to the REST interface.
   - **Stunnel Port**: Port number to connect to the stunnel service.
   - **Eventsdb Port**: Port number to connect to the EventsDB service.
5. Provide the remaining site details and click **CREATE** or **UPDATE**.

Model Changes

This feature includes model changes to the `redundancy-config` and `redundancy-state` to reflect the correct usage of the field floating-ip. Previously the floating-ip node was under `/redundancy-config/site/rw-instances` and `/redundancy-state/config-state` xpaths.

In release 9.1.0, there are four new nodes: instance-address, instance-port, stunnel port and Eventsdb Port. See Configure High Availability Enhancements for the UI changes.

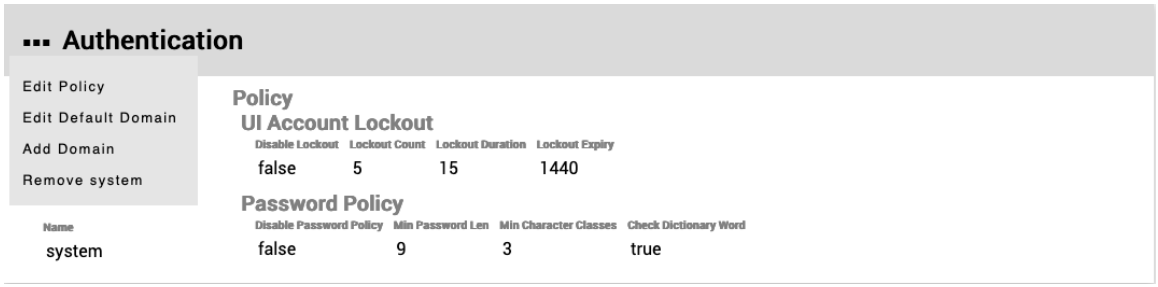
```
        leaf instance-address {
          description "IP address or FQDN for reaching the REST
interface";
          type ip-fqdn;
        }

        leaf instance-port {
          description "Port number to connect to for reaching the REST
interface";
```

```
          type rwt:port-number;
        }

leaf stunnel-port {
          description "Port number to connect to for reaching the
stunnel service";
          type rwt:port-number;
          default 8014;
        }

        leaf eventsdb-port {
          description "Port number to connect to for reaching EventsDB
service";
          type rwt:port-number;
          default 8006;
        }
type uint32;
      }
    }
  }
```

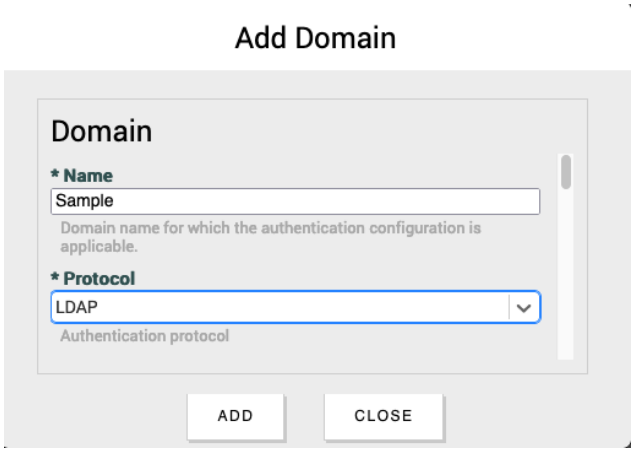
LDAP Plugin Support

This feature introduces a new authentication protocol: Lightweight Directory Access Protocol (LDAP) in release 9.1.0.

1. On the Launchpad Dashboard menu, click **ADMIN-TOOLS**> **CONFIG** to add an authentication domain.
2. On the **Authentication** card, click ▪▪▪ and choose **Add Domain**.



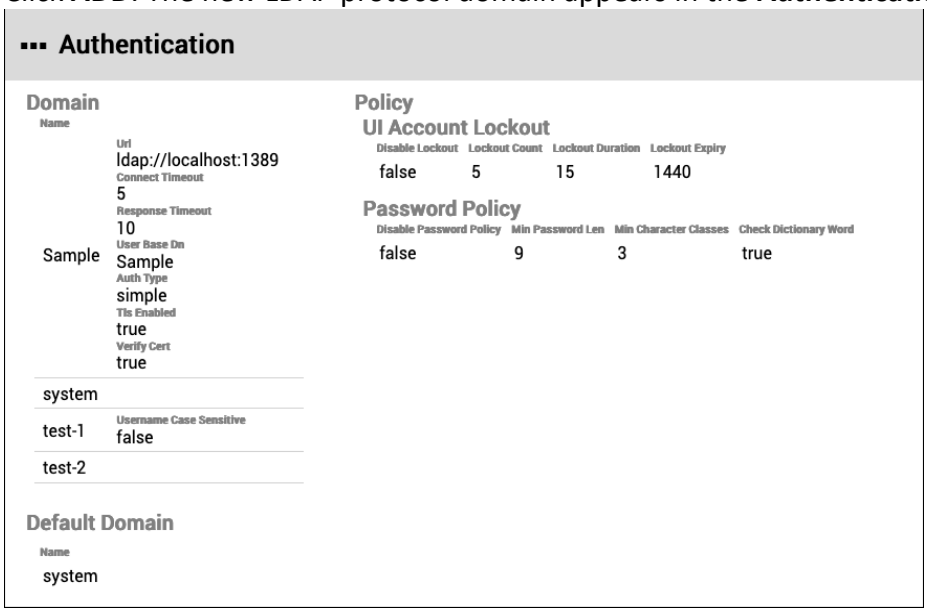
3. On the **Add Domain** window, provide the following details:
   - ***Name**: A value is required. Domain name for which the authentication configuration is applicable.
   - ***Protocol**: Authentication Protocol. In the Protocol dropdown, choose LDAP. After choosing LDAP, there are new fields that an operator can populate for that protocol.



   - ***Server**: URL or host-port.
   - ***URL**: URL for connecting to the LDAP server. If you select URL in the server, then a URL drop-down field appears.
   - **Host-port**: If you select host-port in the server, then a host and port drop-down fields appear.
   - ***Host**: IP address or the fully qualified domain name of the LDAP server

- **Port**: LDAP server's port. Default port value used will be based on TLS setting./nPort number 1389 is used by default for non TLS and 1636 by default for TLS.
- **Connect Timeout**: The connection timeout to use in seconds.
- **Response Timeout**: The response timeout in seconds.
- ***User Base Dn**: Base DN under which the user configurations are present. cn=user-name would get added in front of the provided base DN while authenticating. Example base DN "ou=users,dc=example,dc=org". For active directory, this will be DOMAIN.
- **Auth Type**: LDAP authentication protocol to be used for user authentication. Choose simple or ntlm.
- **Tls Enabled**: Initiates a TLS connection. The server CA certificate must be configured in CA bundle. Choose True or False.
- **Verify Cert**: Verify the certificate sent by the server. By default certificate verification is enabled. Choose True or False.

4. Click **ADD**. The new LDAP protocol domain appears in the **Authentication** card.



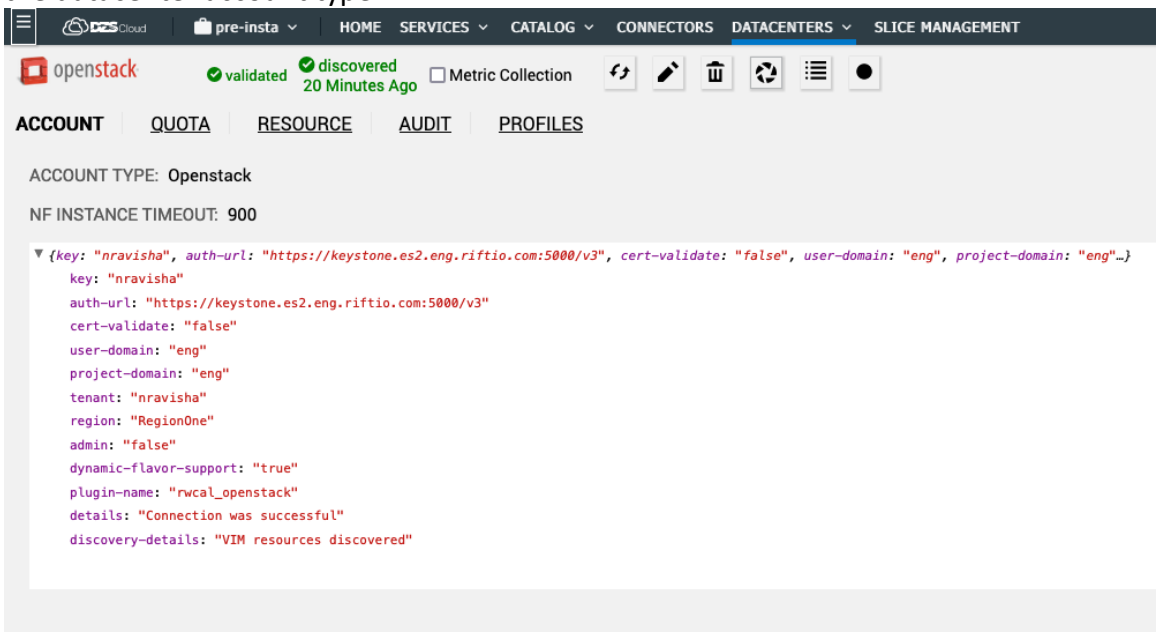
Mapping Service Elements to Existing Resources Improvements

This release includes improvements to the way an operator can map existing resources to accounts and during instantiation. See Map Service Elements to Existing Resources and Discover Account Resources in DZS Cloud RO. This feature introduces new capabilities to help an operator create VNF/CNF descriptors from a running service. An operator can also create segment profiles for a configured datacenter from the running services to help resolve the constructed NS descriptor segment profile reference to a running network service when instantiating a service.

UI Enhancements

This release introduces two new buttons in on the Launchpad NFVO > Datacenters tab after completing discovery. Refer to Discover Account Resources in DZS Cloud RO for details on how to run discovery on an account.

After running discovery, there are two new buttons in the header of the Datacenter tab to generate segment profiles and generate a cnfd or vnfd depending on the datacenter account type.



Generate Segment Profile

1. Click to open the **Generate Segment Profiles** wizard and provide the following details (fields with * are required):
   Network
   - **Filter**: Filter string for network.
   - *** Network**: List of Networks from which to create segment profiles.
   - **Segment Profile Name**: Prefix for generated segment profile name.
   - **External Access**: External access for the network. Will be neglected if the network is not shared. (Default: None)
2. Click **Generate**.

Generate CNFD or VNFD

1. Click to open the **Generate CNFD/VNFD** wizard and provide the following details (fields with * are required):
   Server
   - **Filter**: Filter string for server.
   - **CNFD/VNFD Name**: Name for the cnfd/vnfd generated from datacenter resources.
   - *** Server**: List of Servers from which to create cnfd/vnfd.
2. Click **Generate**.

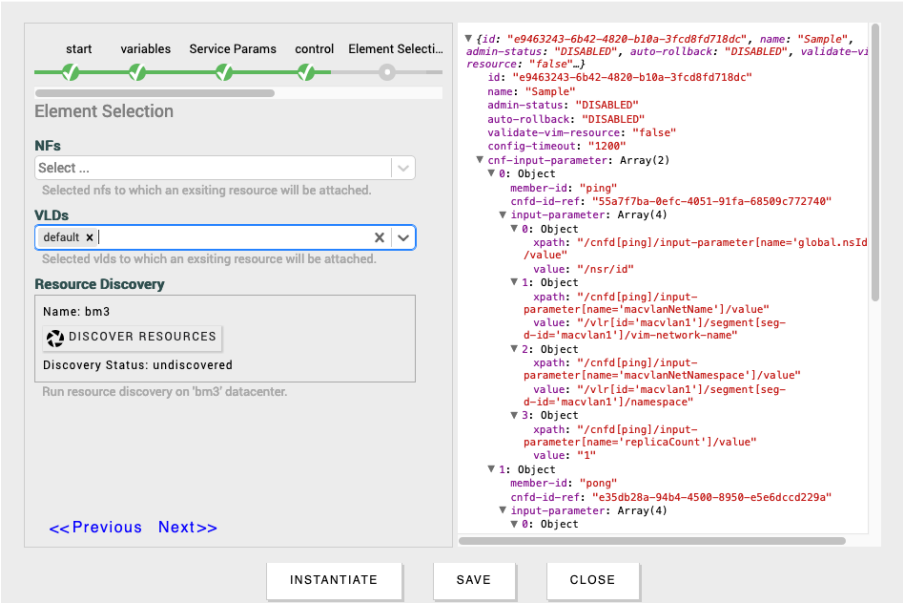
VLDs Section Changes

This release includes improvements to the way an operator can map existing resources during instantiation. See Map Service Elements to Existing Resources for information on how to add existing resources into a running NS during instantiation.

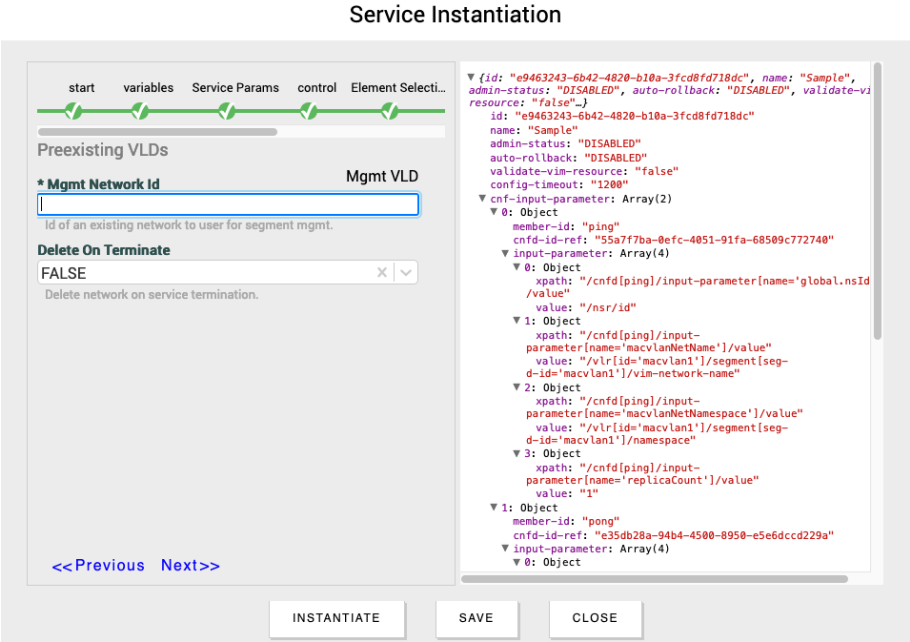
A user can now select VLDs when mapping existing resources in the instantiation wizard. If an operator selects the VLDs, then a new step appears to add Preexisting VLDs.

VLDs

Service Instantiation



Preexisting VLDs



Model Changes

An operator can now create a cnfd or vnfd from cloud resources and create datacenter profiles from discovered networks using the below RPCs.

`https://<launchpad_ip/fqdn>/api/operations/create-nfd-from-resources`

```
{
    "input":
    {
    "project-name": "test52",
    "datacenter": "Auto_OS_Autobot",
    "vnfd-name": "discovery_vnfd_pong",
    "server": [{
    "server-id": "bf775fe9-bd31-488c-90a3-6b3d740f6e8e"
    }]
}
}
```

`https://<launchpad_ip/fqdn>/api/operations/generate-datacenter-profiles`

```
{
   "input":{
      "project-name":"test58_02",
      "cloud-account":"Auto_OS_Autobot",
      "network":[
         {
            "network-id":"459d0de9-0813-4493-90ac-f11d91ec9690",
            "segment-profile-name":"Mgmt-profile"
         },
         {
            "network-id":"3fba4d46-775f-467f-8999-98bd72e5ad03",
            "segment-profile-name":"Discovery-data-link"
```

```
            }
        ]
    }


}
```

Network Slicing UI Enhancements

Note: The Network Slicing UI Enhancements documentation for release 9.1 is still in progress.

Restructured DZS Cloud Orchestration Launchpad UI

In this release, the Launchpad UI has been restructured to be more efficient and useful to all operators. There are a number of enhancements, improvements and new functionalities related to Launchpad Administration.

8.3 DZS Cloud Orchestration UI Top Menu section displays the structure of the UI in release 8.3. The next section walks you through the new structure in release 9.1.

- 8.3 DZS Cloud Orchestration UI Top Menu
  - default
  - HOME
  - SERVICES
  - CATALOG
  - CONNECTORS
  - EVENTS
  - ADMINISTARTION
  - admin
- 9.1 DZS Cloud Orchestration UI Top Menu
  - Table of Contents
  - NFVO
    - default
    - HOME
    - SERVICES
    - CATALOG
    - CONNECTORS
    - DATACENTERS
    - SLICE MANAGEMENT
    - EVENTS
    - admin
  - PLATFORM
    - HOME
    - SYSTEM
    - CONFIG
    - ALERTS
    - DEBUG
    - EVENTS
    - USERS
    - PROJECTS
    - admin
  - METRICS

8.3 DZS Cloud Orchestration UI Top Menu

Note: In the above screen shot, the default project is currently the active project and admin is the current logged in user name.

- default
- HOME
- SERVICES
- CATALOG
- CONNECTORS
- EVENTS
- ADMINISTARTION
- admin

Click **default** to open a drop-down menu displays a list of projects associated to the logged in user (admin).

Note: This project does not include all the projects on the Launchpad installation.

An operator can view the following widgets on the **HOME** tab:

- Datacenters
- Network Services
- Events
- System CPU
- System Memory
- System Disk Utilization
- Rest Metrics
- RO Accounts
- Packages
- Redundancy Info
- Resource Usage
- RIFT.ware Licensing

An operator can view the service list or service order list on the **SERVICES** tab.

- SERVICE LIST
- SERVICE ORDER LIST

An operator can add an NSD, VNFD, CNFD, Specification or NEST on the **CATALOG** tab.

- NSD LIST
- VNFD LIST
- CNFD LIST
- SPECIFICATION LIST
- NEST LIST

CONNECTORS

An operator can create and modify the following account types on Launchpad on the **CONNECTORS** tab.

- Resource Orchestration
- Config Agent
- SDN
- Remote Launchpad
- VNFM
- Secure API Endpoint
- Misc

EVENTS

An operator can access events from Launchpad in the following areas on the **EVENTS** tab:

- NOTICE AND ABOVE
- PLATFORM
- HIGH AVAILABLITY
- LCM MONITORING
- SYSTEM ALARMS
- VIM ALARMS
- RBAC & AUTHENTICATION
- SECURITY
- TRAPS/NOTIFICATION
- REST
- LICENSE
- CONNECTOR
- LCM
- PACKAGE
- USER AUDIT
- SLICING

ADMINISTRATION

An operator can view the following sections related to Launchpad Administration on the **ADMINISTRATION** tab:

- PROJECT MANAGEMENT
- MESSAGE BUS
- SLICE MANAGEMENT

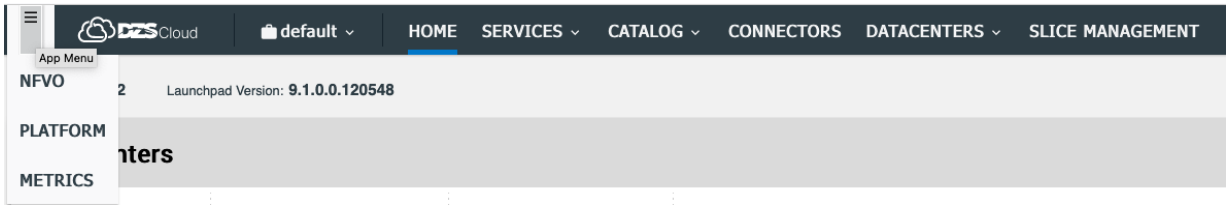
- SSH KEYS
- USER MANAGEMENT
- SYSTEM
- REDUDANCY
- CONFIGURATION
- ABOUT

Note: admin is the default username using the default username/password credentials (admin/admin). After an operator creates a new user and then logs into Launchpad with the new username and password credentials, then the new user appears at the top right corner of the menu.

- MY PROFILE
- LOGOUT

9.1 DZS Cloud Orchestration UI Top Menu

In release 9.1, there is a new App Menu at the top left-hand corner of the UI screen.

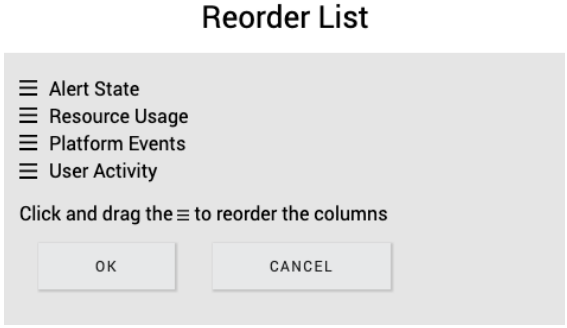


The UI now separates NFVO operations (**NFVO**), administrative configuration and metrics (**PLATFORM**) and additional statistics (**METRICS**) into three different sections.

- NFVO
- PLATFORM
- METRICS

Table of Contents

This release introduces a new Table of Contents on the right side of the UI with links to each card. Click ☰ to the right of **Card Directory** to open a Reorder List window. In the Reorder List window, click and drag ☰ next to a specific card to change the order of appearance in the UI. After rearranging the cards, click **OK**.



NFVO

Choose the **NFVO** view in the App Menu to display and configure NFVO metrics and functions.

- default
- HOME
- SERVICES
- CATALOG
- CONNECTORS
- DATACENTERS
- SLICE MANAGEMENT
- EVENTS
- admin

Click **default** to open a drop-down menu displays a list of projects associated to the logged in user (admin).

Note: This project does not include all the projects on the Launchpad installation.

The **HOME** tab displays the following NFVO metrics are in the **NFVO** tab:

- **Datacenters**: This card appeared on the **HOME** tab in release 8.3.
- **Network Services**: This card appeared on the **HOME** tab in release 8.3.
- **Recent Packages**: This card was previously titled Packages and it appeared on the **HOME** tab in release 8.3.
- **Project Events**: This card was previously titled Events and it appeared on the **HOME** tab in release 8.3.
- **Project Config**: This is a new card in release 9.1. In release 8.3, this information is included on the **ADMINISTRATION** > **PROJECT MANAGEMENT** tab.

There is no change to the **SERVICES** tab in release 9.1.

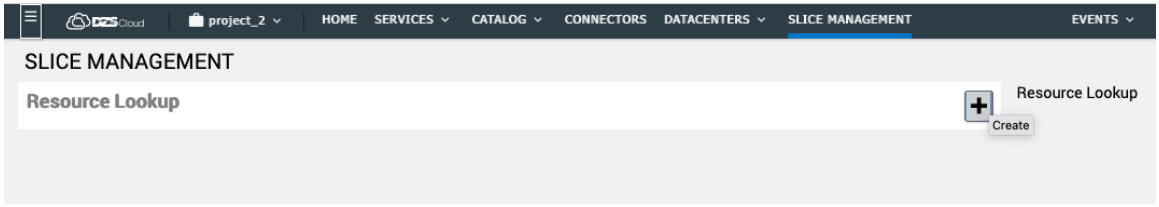
There is no change to the **CATALOG** tab in release 9.1.

There is no change to the **CONNECTORS** tab in release 9.1.

There is no change to the **DATACENTERS** tab in release 9.1.

SLICE MANAGEMENT

An operator can configure the dbLookup Resource Selector type on the **SLICE MANAGEMENT**. In release 8.3, this option is included in the **ADMINISTRATION** > **SLICE MANAGEMENT** tab. See Network Slicing Enhancements.



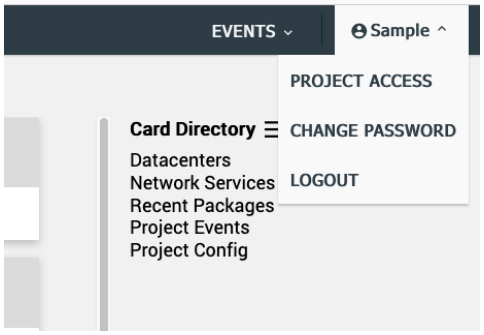
EVENTS

An operator can access events from Launchpad in the following areas on the **EVENTS** tab. In release 8.3, all of the events were listed in the **EVENTS** tab. The events are now broken into **ADMIN-TOOLS** → **EVENTS** and **NFVO** → **EVENTS**.

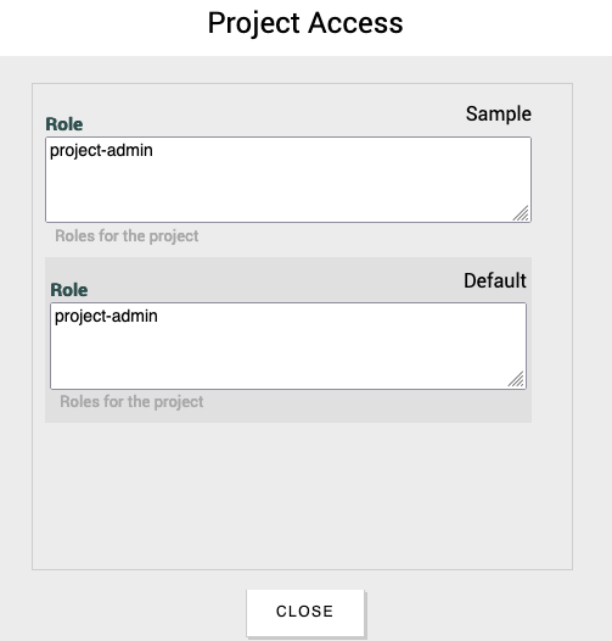
- NOTICE AND ABOVE
- LCM MONITORING
- VIM ALARMS
- CONNECTOR
- LCM
- PACKAGE
- USER AUDIT
- SLICING

admin

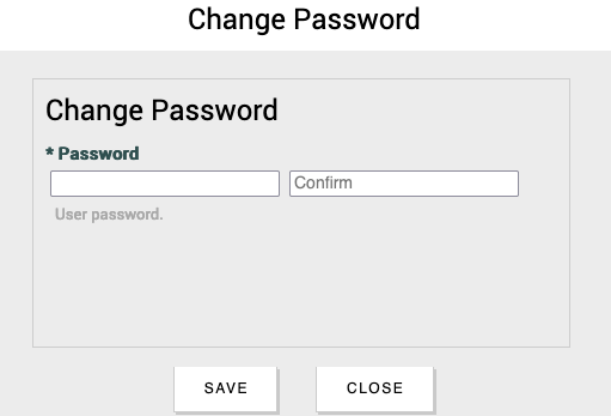
Note: admin is the default username using the default username/password credentials (admin/admin). After an operator creates a new user and then logs into Launchpad with the new username and password credentials, then the new user appears at the top right corner of the menu.



- **PROJECT ACCESS:** This section displays the projects that the logged in user is a member of. This section does not appear for a super-admin user since super-admins have access to all projects by default.



- **CHANGE PASSWORD**



- **LOGOUT**

PLATFORM

Choose the **PLATFORM** view in the App Menu to display and configure administrative metrics and functions.

- HOME
- SYSTEM
- CONFIG
- ALERTS
- DEBUG
- EVENTS
- USERS
- PROJECTS

- admin

HOME

An operator can view the following cards on the **HOME** tab:

- **Alert State**: This is a new card in release 9.1.
- **Resource Usage**: This card appeared on the **HOME** tab in release 8.3.
- **Platform Events**: This card was previously titled Events and it appeared on the **HOME** tab in release 8.3. An operator can set up platform events alerts in this card.
- **User Activity**: This card appeared on the **HOME** tab in release 8.3.

SYSTEM

An operator can access the following cards on the **SYSTEM** tab:

- **Launchpad – Instance**: This information appeared in the **ADMINISTRATION** > **ABOUT** section in release 8.3. In this card, an operator can:
  - Edit instance information
  - Restart UI Service
  - Restart Launchpad service
  - Export Launchpad state
  - Prepare Launchpad recovery
  - Delete All imports
  - Refresh
- **Redundancy**: This information appeared in the **ADMINISTRATION** > **REDUNDANCY** section in release 8.3. In this card, an operator can:
  - Edit Base
  - Edit Polling
  - Add Site
  - Manage ‘siteX’: When an operator adds a new site, a new manage siteX option (X is the site name) appears in the Redundancy menu. A user can configure, reconfigure or delete a site.
  - Execute Command:
    - SHUTDOWN
    - REBOOT
    - ERASE
    - FORCE_SYNC
  - Refresh

Note: If the Launchpad is paired to another Launchpad, then an In sync icon appears on the right side of the Redundancy card.

- **Message Bus** - This information appeared in the **ADMINISTRATION** > **MESSAGE BUS** section in release 8.3. In this card, an operator can:
  - Edit Publishing

    - Edit event Topic Map
    - Add Account:
        - Create
        - Delete
        - Modify
    - Refresh Status
- **About** - This information appeared in the **ADMINISTRATION** > **ABOUT** section in release 8.3. In this card, an operator can view the:
    - Version
        - Version
        - Build SHA
        - Build Date
    - Foss Info
    - Copyright

CONFIG

An operator can configure the following options on the **CONFIG** tab:

- **License**: This card appeared on the **HOME** tab in release 8.3. In this card, an operator can:
    - Add License: Add a license using the host key from the Launchpad on **SYSTEM → Launchpad – Instance**.
    - Remove *xxxxxx* (a license)
- **OpenIDC Providers**: This information appeared in the **ADMINISTRATION** > **CONFIGURATION** > **openidc-provider-config** section in release 8.3. In this card, an operator can:
    - Edit Basics: Update OpenIDC Providers
    - Add Client: Create an OpenIDC Client
    - Manage 'X': When an operator adds a new client, a new Manage X option (X is the client name) appears in the OpenIDC Providers menu. A user can configure, reconfigure or delete a client.
- **Environment Variables**: This information appeared in the **ADMINISTRATION** > **CONFIGURATION** > environment-variables (20) section in release 8.3. In this card, an operator can edit environment variables.
- **SNMP Traps**: This information appeared in the **ADMINISTRATION** > **CONFIGURATION** > **snmp-trap** section in release 8.3. In this card, an operator can:
    - Edit USM User Table: Update User MIB
    - Edit SNMP Target Addr Table: Update Target MIB
    - Edit VACM View Tree Family Table: Update View Based ACM MIB
    - Test Trap: Send a test trap
- **Authentication** – This information appeared in the **ADMINISTRATION** > **CONFIGURATION** > **authentication-config** section in release 8.3. In this card, an operator can:

    - Edit Policy: An operator can create a user by choosing 'disable password policy' as true and verify no authentication condition is required. An operator can also make change to the password length policy and verify that it is satisfied. This is related to login attempts with an invalid password.
    - Edit Default Domain: An operator can create and delete one or more SAML domains. An operator can also delete the default domain.
    - Add Domain: An operator can add a domain. The available domain protocols are SAML, WebClient, System and LDAP. LDAP is a new domain in release 9.1.0. See LDAP Plugin Support (add link).
    - Manage 'X': When an operator adds a new domain, a new Manage X option (X is the domain name) appears in the Authentication menu. A user can configure, reconfigure or delete a domain.
    - Remove system: An operator can remove an added domain system since there is no configuration for a system domain.
- **REST Conf** – This information appeared in the **ADMINISTRATION** > **CONFIGURATION** > **rwrestconf-configuration** section in release 8.3. In this card, an operator can:
    - Add RWrestconf Configuration
    - Edit RWrestconf Configuration
- **Certificates** – This information appeared in the **ADMINISTRATION** > **CONFIGURATION** > **certificate-config** section in release 8.3. In this card, an operator can:
    - Add, Edit or Remove Certificate
    - Add, Edit or Remove CA Bundle
- **Resource Naming** – This information appeared in the **ADMINISTRATION** > **CONFIGURATION** > **resource-naming** section in release 8.3. In this card, an operator can:
    - Edit Resource Naming rules

ALERTS

An operator can check the following metrics on the **ALERTS** tab. In all of these cards, an operator can add or configure alerts.

- **CPU Usage**: This card was previously titled System CPU and it appeared on the **HOME** tab in release 8.3. This section displays an alert when the Launchpad core CPU usage crosses a specified threshold.
- **Memory Usage**: This card appeared on the **HOME** tab in release 8.3. This section displays an alert when the Launchpad core memory usage crosses a specified threshold.
- **Disk Usage**: This card was previously titled System Disk Utilization and it appeared on the **HOME** tab in release 8.3. This section displays an alert when the Launchpad core disk usage crosses a specified threshold.

- **Rest Metrics**: This card appeared on the **HOME** tab in release 8.3. This section displays an alert when the RW.REST requests per second crosses a specified threshold.
- **License**: This card was previously titled DZS Cloud Orchestration (RIFT.ware) License and it appeared on the **HOME** tab in release 8.3. This section displays an alert when the usage crosses a specified percent of the licensed limit.
- **Active Auth Sessions**: This is a new card in release 9.1. This section displays an alert when the number of active sessions crosses threshold.

DEBUG

An operator can check the following metrics on the **DEBUG** tab:

- **System Logs** – In this card, an operator can:
  - Generate Logs
  - Refresh Log Detail
- **Logging Events** – In this card, an operator can:
  - Edit Default Severity
  - Edit Deny
  - Edit Allow
- **Logging Console** – In this card, an operator can:
  - Enable Logging: Enable Configuration
  - Edit Filter: Update Console Filter
- **Logging Sink** – In this card, an operator can:
  - Add sink: Create Logging Sink
  - Manage rift.log
  - Edit rift.log Filter
- **Logging Events Database** – In this card an operator can edit the logging events database.
- **Component Info** – In this card, an operator can refresh this information.
- **POD Info** – In this card, an operator can refresh this information.

EVENTS

An operator can access events from Launchpad in the following areas on the **EVENTS** tab. In release 8.3, all of the events were listed in the **EVENTS** tab. The events are now broken into **ADMIN-TOOLS → EVENTS** and **NFVO → EVENTS**.

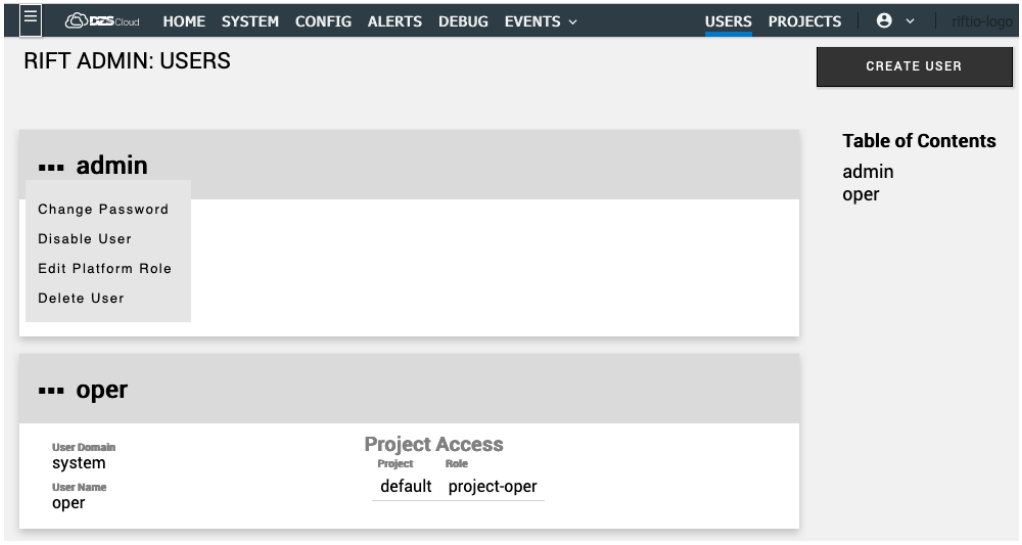
- **PLATFORM**
- **HIGH AVAILABILITY**
- **SYSTEM ALARMS**
- **RBAC & AUTHENTICATION**
- **SECURITY**
- **TRAPS/NOTIFICATIONS**
- **REST**
- **LICENSE**

USERS

An operator can create or delete a user on the **USERS** tab. It also displays all of the users on the Launchpad. The default options are admin and oper.

Click CREATE USER to create a user. After creating a user, click ▪▪▪ next to the user name to:

- **Change Password**
- **Disable User**
- **Edit Platform Role**
  - super-admin
  - platform-admin
  - platform-oper
  - project-list
- **Delete User**



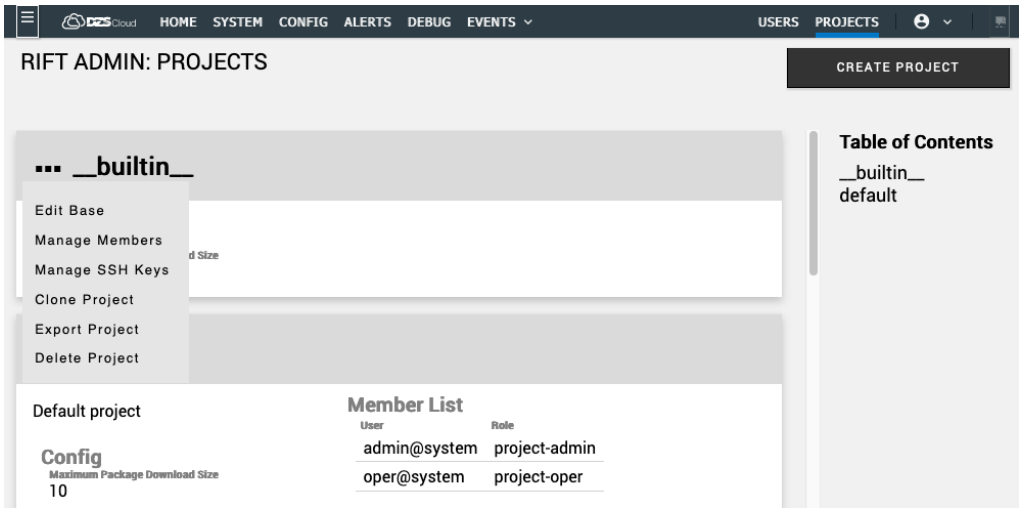
PROJECTS

An operator can create, modify, clone, export, delete and configure a project on the **PROJECTS** tab. It also displays all of the projects on the Launchpad. The default options are __builtin__ and default.

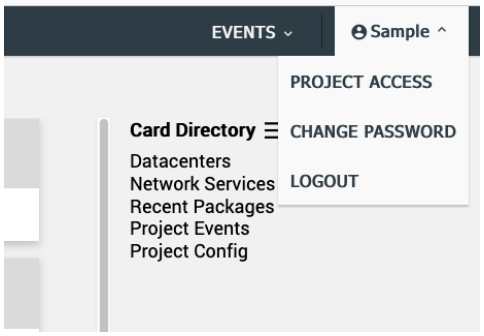
Click CREATE PROJECT to create a project. After creating a project, click ▪▪▪ next to the name to:

- **Edit Base**
- **Manage Members**
  - Assign a user to a project with roles
  - Delete the assigned user
  - Modify the roles assigned to a user
- **Manage SSH Keys**

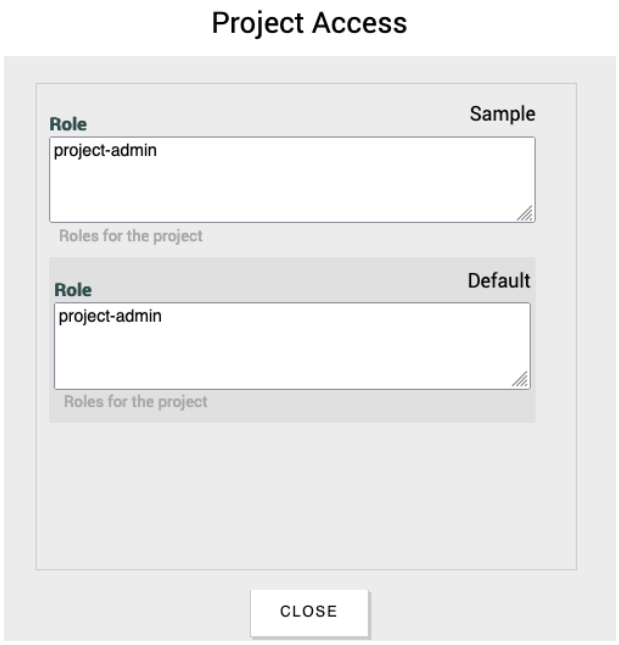
- **Clone Project**
  - New local Project
  - Project
  - Remote Launchpad
- **Export Project**
  - New local Project
  - Project
  - Remote Launchpad
- **Delete Project**



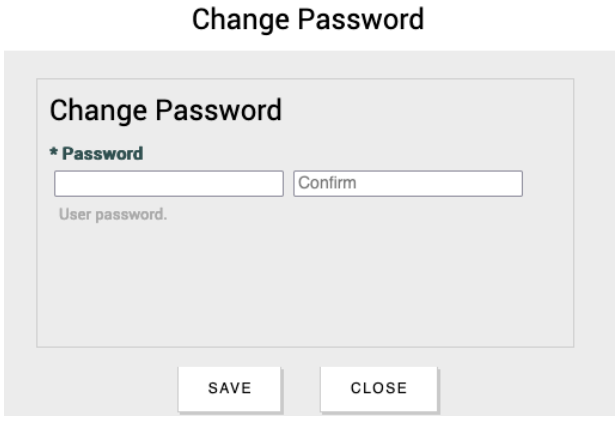
Note: admin is the default username using the default username/password credentials (admin/admin). An operator can create a new user and then logs into Launchpad with the new username and password credentials, then the new user appears at the top right corner of the menu.



- **PROJECT ACCESS:** This section displays the projects that the logged in user is a member of. This section does not appear for a super-admin user since super-admins have access to all projects by default.



- **CHANGE PASSWORD**



- **LOGOUT**

METRICS

Click the **METRICS** link to open DZS Cloud Orchestration metrics. Prior to release 8.3, this section was titled **GRAFANA** in the UI.

Note: The **METRICS** option only appears for a super-admin user and when Launchpad is installed on Kubernetes.

SOL 001 VNF Indicators Support in VNFD

DZS Cloud Orchestration now supports SOL 001 indicators in Launchpad. The Launchpad UI includes a new Indicator card on the Network Services tab.

This feature allows for asynchronous notification of VNF specific information from the VNFM. An operator must define an attribute value in a VNF Node type for a VNF indicators. The value of the VNF indicator can be impacted every time a notification is received.

This information only appears after an operator configures a VNFM to send indicators to the required project on the Launchpad. See Add a VNFM Account to create an OR-VNFM account.

Note: SOL 001 VNF Indicators are not supported when an operator is mapping service elements to existing resources.

- Define VNFD Attributes for SOL 001 VNF Indicators
- View SOL 001 VNFM Indicators in the UI
- Model Changes

Define VNFD Attributes for SOL 001 VNF Indicators

An operator must define the attribute values in a VNF node type for the VNF indicators.

1. On the **Launchpad** menu, click **CATALOG > VNFD LIST** and select the appropriate VNFD descriptor from the catalog. Double-click the descriptor.

   The **PING_VNFD** or **PONG_CNFD** screen opens.

2. Navigate to the Policy Definition section, provide the following information:

   POLICY

   - **NAME**: Name of this policy.

   SCALING GROUP

   - **INSTANCE ID**: Default: 1

   SCALING GROUP NAME

   - **ENABLED**: Whether this policy is enabled or not (TRUE OR FALSE)
   - **COOLDOWN TIME**: The duration in seconds after a policy action has been triggered, for which there will be no further operation. Default: 120.

   CRITIERIA: List of conditions to be met for triggering an operation.

   - **NAME**: Name for the criteria.
   - **CRITERIA TYPE**: INDICATOR
   - **INDICATOR**: Criteria based on NS monitoring parameters

- **INDICATOR NAME**: Name of the indicator.
- **COMP OPERATOR**: Operator to compare the values. Default: GT.
  - EQ (Equal to)
  - GT (Greater than)
  - LT (Lesser than)
  - GE (Greater than or Equal to)
  - LE (Lesser than or Equal to)
  - NE (Not Equal)
- **THRESHOLD VALUE**: Threshold value to compare. Default: 0

LCM ACTION: Actions to be triggered when criteria is met.

- **SEQ**: Sequence in which the actions are to be executed. The lowest sequence is executed first. Default is 1.
- **CRITICAL**: This attribute determines if a failure of this action is critical or not. If true, then the action fails and does not continue with subsequent ones. If false, then the action continues with the next action even if this one fails. Default: False.

ACTION TYPE: Choose scaling-action.

- Primitive
- scaling-action
- healing-action

- **SCALING ACTION**: Scaling action to be executed when criteria are met.
- **SCALING TYPE**: Scaling action is scale out or scale in.
  - scale-out
  - scale-in

3. Click 

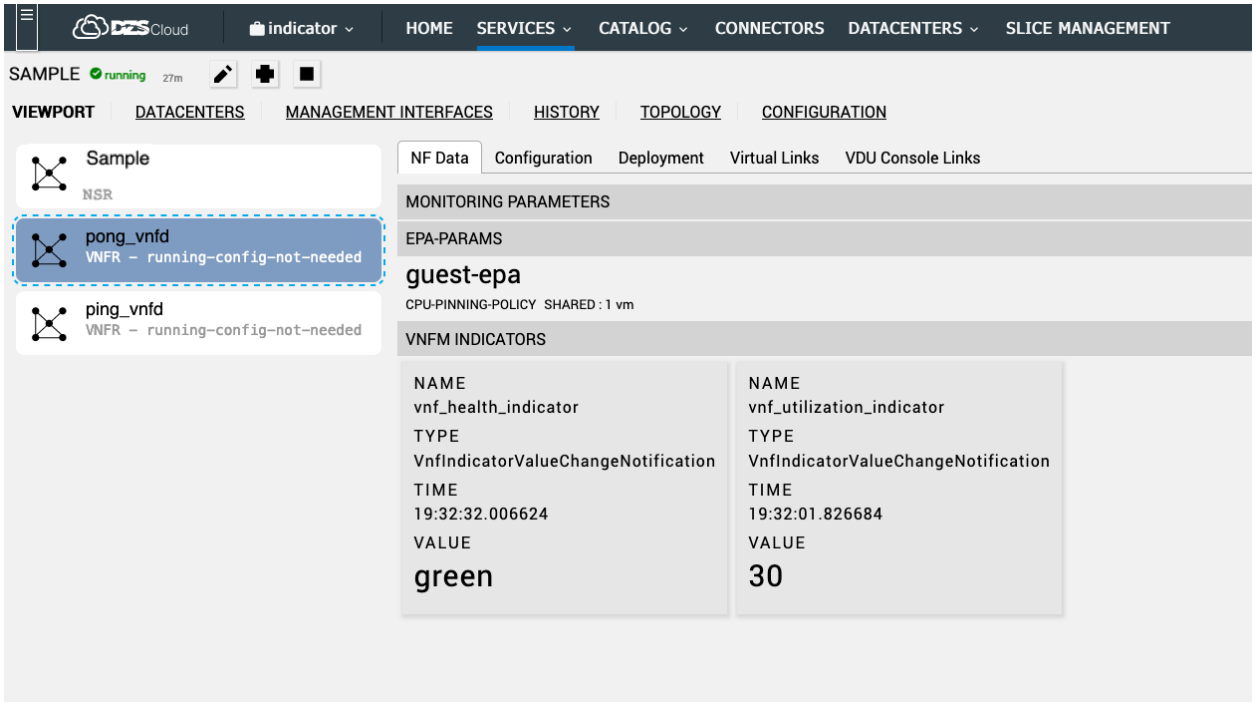
.
4. Click **INSTANTIATE** or **SAVE** the Service.
   The Network Services screen includes a new indicator card displaying the SOL VNF Indicators.

View SOL 001 VNFM Indicators in the UI

After defining the attribute values in a VNF node type for the VNF indicators, then a new indicator card appears on the Network Services tab.

The **VNFM INDICATORS** tab displays:

- NAME
- TYPE
- TIME
- VALUE



Model Changes

Added a new indicator type field type in the vnfr-catalog Data Model. This field includes the following variables:

project → vnfr-catalog → vnfr → indicator

Name	Type	Description	Is ID TYPE?
id	string	Id of indicator from VNFM	Yes
name	string	Name of indicator from VNFM	No
type	string		No
value	string	Indicator value	No
notification-type	string	For example, VnfIndicatorValueChangeNotification	No
time-stamp	string	Sent by VNFM, time at which indicator was generated	No
subscription-id	string	Subscription id coz of which this indicator was sent	No
vnf-instance-id	uuid	Identifier for the VNFM Instance id.	No

Added a new policy type field type in the vnfd-catalog Data Model. This field includes the following variables:

vnfd-catalog → vnfd → policy

Name	Type	Description
name (key)	string (length 1..63)	Name of this policy.
scaling-group	list	
enabled	boolean	Whether the policy is enabled or not.
cooldown-time	uint32	The duration in seconds after a policy action has been triggered, for which there will be no further operation.
criteria	list	
lcm-action	list	

scaling-group

Name	Type	Description
instance-id	string	
scaling-group-name	leafref (path: "../../../vnf-scaling-group-descriptor/name")	

criteria

Name	Type	Description
name	string (length 1..63)	Name of the criteria
criteria-type	choice	CHOICES: **indicator**

indicator

Name	Type	Description
indicator-name	string	
comp-operator	enum (ET, GE etc)	Operator to compare the values
threshold-value	string	Threshold value to compare

lcm-action

Name	Type	Description
seq	uint32	Sequence in which the actions are to be executed. The lowest sequence is executed first.
critical	boolean	

action-type	choice	CHOICES: scaling-action, enum, either scale out or scale in to be executed healing-action, enum, restart or reconfigure primitive, leafref (path: "../../../../vnf-configuration/config-primitive/name"), Primitives to run

Added a new indicator type field type in the vnfd-catalog Data Model. This field includes the following variables:

vnfd-catalog → vnfd → indicator

Name	Type	Description
id (key)	string	
description	string	
type	string	
constraints	enum	This determines if the indicator value from VNFM is valid
values	string	valid_values For example: 1, 2, 3; in_range For example: 1:3
interface	string	Mapped field to SOL001 indicator interface
name	string	Name of indicator sent by VNFM
source	string	Source of indicator creation

Upgrade Process Changes for a Geographically Paired System

Release 9.1.0 introduces changes the upgrade process for a Geographically paired system when upgrading from 8.3.1 to 9.1.0. There is no change to the upgrade process for a standalone system

Note: In this procedure, the Launchpad nodes are called Launchpad-1 and Launchpad-2.

- Break the Geographically Paired Systems
- Export Launchpad to Backup Data
- Upgrade Launchpad-1 to Release 9.1
- Validate the Configurations and the Network Service
- Install Launchpad-2 Release 9.1
- Configure Geographic Redundancy on Paired Launchpads
- Configure the Proxy Ports in the Geographic Redundancy Configuration

Break the Geographically Paired Systems

Break the Geographically paired systems by removing all the geographic redundancy configuration.

1. Delete the Geographic Site configuration from Launchpad-1. Remove both sites. See High Availability with Geographic Redundancy.
2. Delete the licenses from Launchpad-2and Launchpad-1. See License Management.
3. Delete the openid configuration from Launchpad-2and Launchpad-1. See Configure DZS Cloud Orchestration for Geographic Redundancy.
4. Uninstall Launchpad-2. See DZS Cloud Orchestration Installation.

Export Launchpad to Backup Data

Next, perform the Launchpad export to back up all the data. See Disaster Recovery Support.

Note: Keep one of the Launchpad nodes to complete the upgrade and uninstall the other node. In this example, the operator can keep Launchpad-1 and uninstall Launchpad-2.

Upgrade Launchpad-1 to Release 9.1

Upgrade Launchpad-1 to 9.1 using the following command. See DZS Cloud Orchestration Installation.

```
helm upgrade <release-name> <chart-path> --set
launchpad.upgradeFrom=8.3.1.3.119771 -f <values-file>
```

Validate the Configurations and the Network Service

Manually validate all the configurations such as catalogs, all Network Services and Datacenters are back up. Verify that there are no errors in the Launchpad.

Install Launchpad-2 to Release 9.1

Install Launchpad-2 with the new 9.1 image. See DZS Cloud Orchestration Installation.

Configure Geographic Redundancy on Paired Launchpads

Configure Geographic Redundancy on both paired Launchpads.

1. Add the system config in lp-2-new (Launchpad-2 new). See High Availability with Geographic Redundancy.
2. Add licenses of both Launchpad nodes in both Launchpad nodes. See License Management.
3. Add openid config of both nodes in both nodes. See Configure DZS Cloud Orchestration for Geographic Redundancy.
4. Add site config of both Launchpad nodes in both Launchpad nodes. While adding the site, make sure to add the **proxy port** details. See High Availability with Geographic Redundancy.
    - Instance Port is for Launchpad (default 443)
    - Stunnel Port is for REdis Sync (default 8014)
    - Eventsdb port is for Mongo(default 8006)
5. Add the federation id under the Geographic Redundancy config in both Launchpad nodes.
6. Send the PAIR **command to lp-2-new** so that it comes up as STANDBY and gets all the config from Launchpad-1. See High Availability with Geographic Redundancy.

Configure the Proxy Ports in the Geographic Redundancy Configuration

DZS Cloud Orchestration Northbound Integration User Guide

This guide shows how an application can integrate with DZS Cloud Orchestration on the northbound side. Please only distribute the PDF version of the guide.

IMPORTANT Please send any changes directly to Christine Crudele . If you need to make a change to the document, then post an updated version here with your initials at the end of the file name and rev up the version (i.e.: 8.3.0-dzs-cloud-orchestration-northbound-APIs-v1.3.5_CC.docx). Include your changes in the comments. See Northbound APIs - updates, etc for additional information.

This guide consists of 2 fonts. Calibri is the main font and Courier New is used for all code. The Calibri font size changes for the headings and the body. If you find a discrepancy regarding the font, then please contact Christine Crudele.

Version	File
Version 8.3.0	8.3.0-dzs-cloud-orchestration-northbound-APIs.pdf
Version 8.1.0	8.1.0-riftware-northbound-APIs-v1.3.13.pdf
Version 7.2.2	DRAFT version only for this release. Priority was 8.1.0 Northbound API guide. 7.2.2-riftware-northbound-APIs_V1.docx
Version 7.2	7.2.0-RIFT.ware-Northbound-APIs.pdf
Version 6.2	Asynchronous NS APIs.pdf

Northbound APIs

CONNECTORS AND DATACENTER

- Add a New Datacenter (VIM) Account
- Add a New VNFM Account
- Add a New CaaS Account
- Query the CaaS Account Validation Status
- Get VIM State of a Single Account
- Get the State of all VIM Accounts in a Project
- Get VNFM State of a Single Account
- Get the State of all VNFM Accounts in a Project
- Get CaaS Account State
- Get the State of all CaaS Accounts in a Project

Configure the proxy ports in the Geographic Redundancy configuration. From release 9.1 onwards, an operator can configure Geographic redundancy on paired launchpads even if it is sitting behind a proxy with well-known ports mapped to different port numbers.

Validate NS Resource Requirements

This feature allows a user to verify resources against the VIM before Instantiating a Network Service with CNFs or VNFs.

Note: It is not possible to validate NS resource requirements for VMware VCD accounts in release 9.1.

Validate NS Resource Requirements with VIM before Instantiation

In release 9.1, an operator can verify resources against the VIM before instantiating, updating or scaling out a Network Service with CNFs or VNFs in the DZS Cloud Orchestration UI.

Note: The workflow is specific to the NSD from which the service is created. The below workflow shows the process for a new Network Service instantiation. An operator can also verify resources against the VIM when updating an NS or scaling out an NS. See Add or Remove a VNF to a Running or Failed-Temp Network Service and Add or Remove a CNF to a Running or Failed-Temp Network Service.

1. On the **Launchpad NFVO** menu, click **SERVICES**.
2. Click next to the NS in the Service page for an NS that has not been instantiated and the current configuration of the service appears.
   This page also includes the button to start the instantiation process for a service or a delete a service.

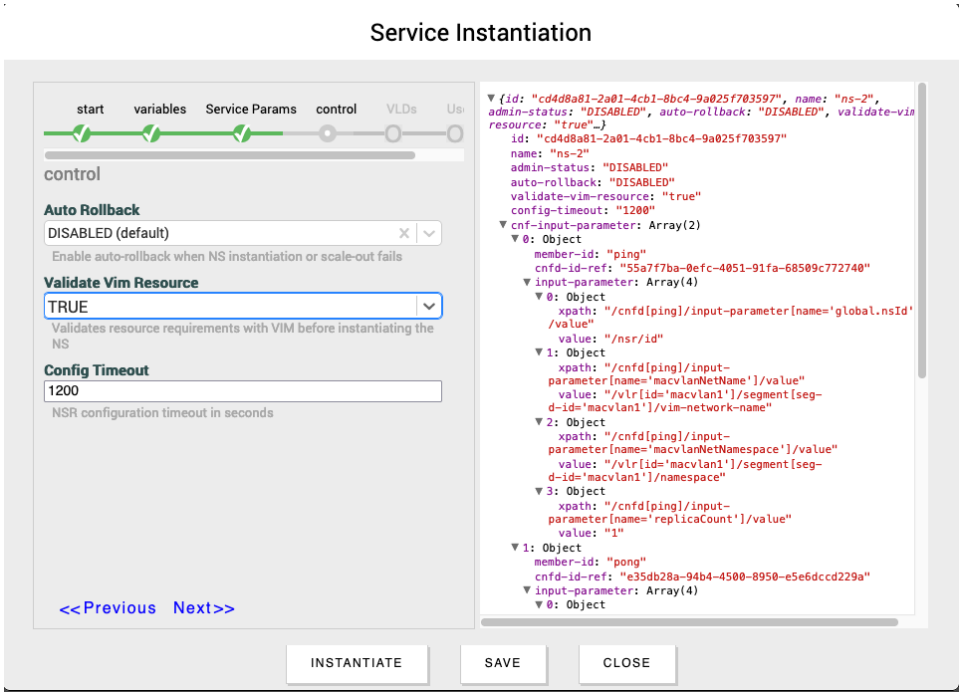
3. Click to open the Service Instantiation screen. Fill in the appropriate details for the specific service.

   Note: You cannot navigate forward without filling in required fields (*). The **Next >>** button will not function if the previous required field isn't populated. The Save button allows you to save your NSR config content and edit it later if you are not ready to Instantiate.

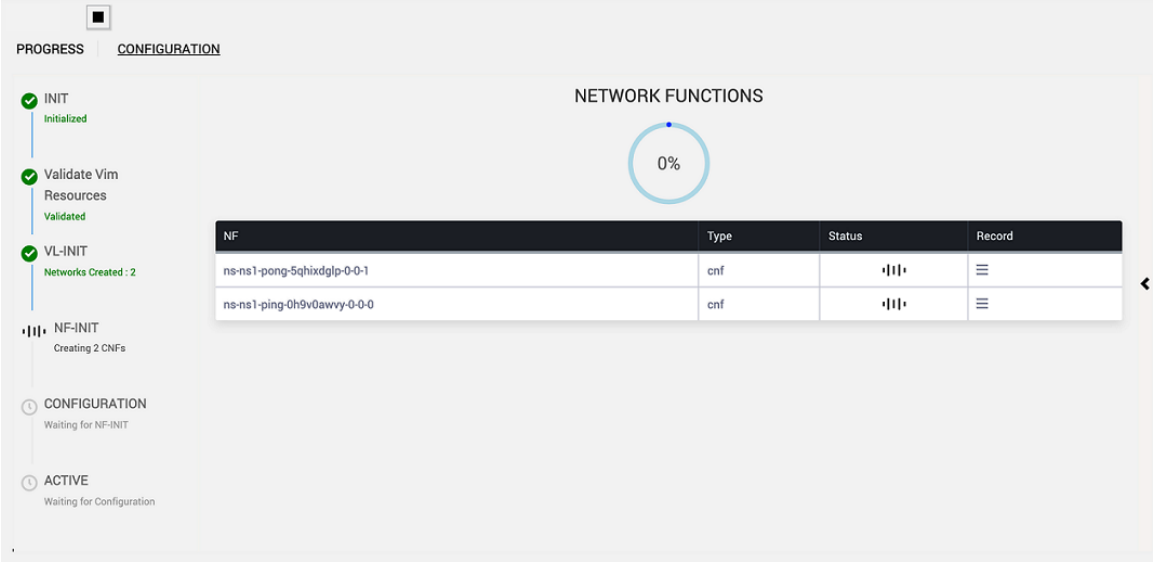
4. Provide the necessary details in the start and Service Params Service Instantiation steps.

Note: The steps in the process can change depending on the way an NS is configured For example, if there are instantiation-variable or input-parameter-xpath entries, then there is a variable step.

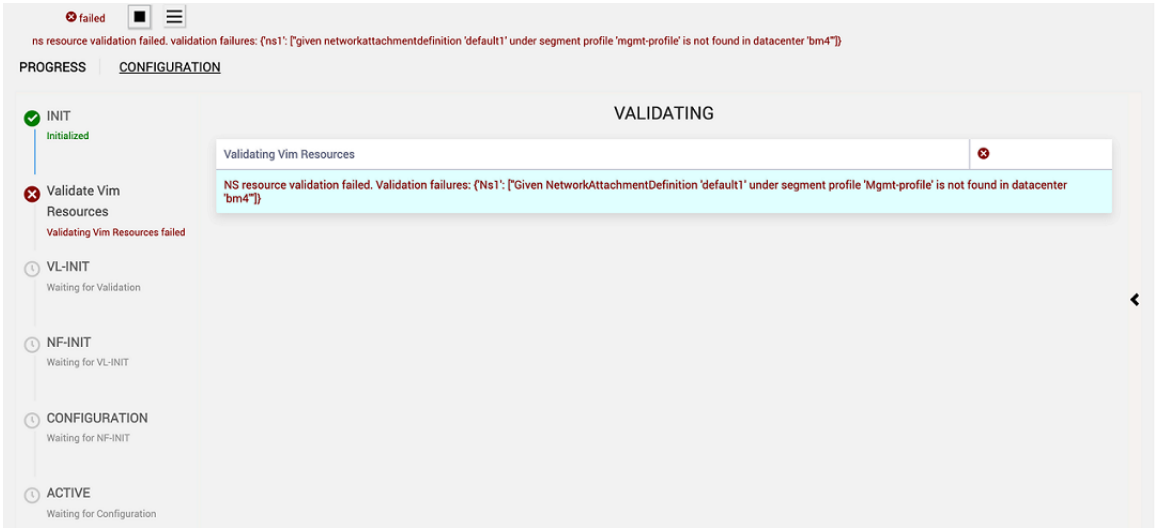
5. Click **Next >>** to provide the following details in the control step. In this step, there is a new Validate Vim Resource option. The default value for this field is False. Click True to verify resources against the VIM before instantiating a NS, updating an NS or scaling out an NS.
   - **Auto Rollback**: Enable auto-rollback when NS instantiation or scale-out fails.
   - **Validate Vim Resource**: Validates resource requirements with VIM before Instantiating the NS.
   - **Config Timeout**: NSR configuration timeout in seconds.



6. Click **Next >>** to provide details in the remainder of the Service Instantiation steps.
7. Carefully check your entries. Click **INSTANTIATE** to start the instantiation of the service or click **SAVE** if you are not ready to instantiate. If an operator clicks true to verify resources against the VIM, then the Network Services progress bar will show an additional step for **Validate VIM Resources** after the **INIT** phase during instantiation.



If an operator clicks true to verify resources against the VIM during the instantiation process and the resources and data passed are not valid, then the instantiation will fail.



Fixed Issues in DZS Cloud Orchestration 9.1.0.0

Support Tickets	Title	Description
RIFT-23916	Master: HA- Re-Paring of active-node after failover leaves the nodes in split brain condition.	While an operator was setting up HA pairing of Launchpad nodes prior to release 9.1, it was possible to accidentally trigger the PAIR operations even after an initial successful pairing. In this release, the commands shown or available to a user from the UI are based on the health status of the Launchpad nodes in the HA pair. After an initial successful pairing operation, the PAIR command is not visible in the UI which prevents any “double pairing” situations. This issue is resolved.
RIFT-29543	CNFD service instantiation failing with error release monitoring failed: 'str' object has no attribute 'type_yang'	CNFD service instantiation failed when a helm-chart-based Launchpad was setup in a non-default namespace and used the chart to launch CNFs in other namespaces. The issue was not seen with standalone Launchpads. Now CNFs can be instantiated by Launchpad in a k8s even though the same namespace is not in the target datacenter. This issue is resolved.
RIFT-30849	[k8s based LP] Requires namespace same as release name manually created in cluster, when service is instantiated from k8s based LP on different bare metal cluster	If launchpad was deployed in a custom namespace (For example: cluster_ns_1) in Kubernetes cluster, then this custom namespace (cluster_ns_1) was used by default by Launchpad to instantiate Network Service resources in Kubernetes cluster. The Network Service failed. Now CNFs can be instantiated by Launchpad in a k8s even though the same namespace is not in the target datacenter. This issue is resolved.
RIFT-31602	cnfd:chart-info:interface:connection-point-ref doesn't seem to do anything	Launchpad does not support updating the helm chart in a CNFD through the composer UI. After updating the helm chart outside Launchpad, an operator can upload the

Support Tickets	Title	Description
		modified chart as an asset, update the chart information in the CNFD and then click the update CNFD. This will update the CNFD to show the new chart structure. This issue is resolved.
RIFT-31732	Discovered VDU instances are marked as init under VDU operational-status.	A Network Service which was created using Sol003 based VNF Brownfield Discovery showed VNFR:VDUR status as 'init'. The status should have been 'running' instead of 'init'. VDU status is now updated to the running state once Brownfield Discovery is done for the VNF. This issue is resolved.
RIFT-31872	Retry fails for cnfd service even if physical network is corrected to proper value in BM datacenter	A network attachment was created in a cluster with the wrong physical network during the VL init stage. The cluster did not validate it and as a result the cluster returned a successful response and the VL creation was successful. The NF init stage failed while trying to attach a pod with an invalid network. Since the network is already created, the retry will not try creating the network again because k8s already indicated that the network creation is successful. This is the same case with NSD and NFDs. The updated changes are not added a NSD or NFD after a Network Service is created for a retry attempt. This issue is resolved.
RIFT-32304	Pass LP IP as part of input params in primitives	Enhanced the input that is passed to the script for primitive execution. Now the script knows if Launchpad is running in Kubernetes and it also identifies the related parameters. Also, enhanced the rw-config-util. Added the following properties for rw-config-util: • namespace: Get the namespace if LP is running in k8s

Support Tickets	Title	Description
		• pod_name: Get the pod name if LP is running in k8s • lp_service_name: Get the service name for DZS Cloud Orchestration Launchpad • is_in_k8s: Check if LP is in k8s or not In addition, added the helper method: • send_lp_request: Send request to DZS Cloud Orchestration Launchpad and return the response data. This issue is resolved.
RIFT-33141	NFVO page shows error after project deletion	An operator created a project and added an OpenStack account to the Datacenter in the created project. Next, the operator deleted the project without deleting the added cloud account. Then, the operator opened the NFVO page and the UI displayed an error message. The error message disappeared after 5 to 10 seconds. This issue is resolved.
RIFT-33242	CNFD's generated from discovered data for BFD, throw helm chart name mistmatch error with names other than 'pingchart' or 'pongchart'	A CNFD was generated from discovered data. Then, the operator provided a name other than 'pingchart' or 'pongchart' and the service ended up in helm chart name mistmatch error during instantiation. This issue is resolved. See Mapping Service Elements to Existing Resources Improvements section for UI and Model changes for the create-nfd-from-resources RPC changes.

Known Issues in DZS Cloud Orchestration 9.1.0.0

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. DZS Cloud Orchestration 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 DZS Cloud Orchestration 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 DZS Cloud Orchestration REST APIs to complete configuration changes must have a delay between two successive config change operations.
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.
RIFT-23621	Introduce a new field in VNFM account page for DZS Cloud Orchestration<-->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,	DZS Cloud Orchestration uses a GET call to /vnf_instances to validate VNFM Account. If a SVNFM account does not support GET	N/A

Support Tickets	Title	Description	Impact	Workaround
		handshake is not required.	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 DZS Cloud Orchestration 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.	
RIFT-23760	Terminating one service with the same name on any Launchpad interrupts existing services on all Launchpads.	This issue occurs because all services on all Launchpads have the same UniqueID which is the InstanceID on ES2. When a user terminates a service, then terminates Instance on ES2 which causes issues for all	If a user terminates any of the services on either of the network-services, this will make any other services orphan.	Use a unique NSR name if an organization is using multiple Launchpads. Ensure that an OpenStack tenant is controlled by a single Launchpad. **Note**: There are no plans to change this behavior.

Support Tickets	Title	Description	Impact	Workaround
		existing services because they have the same UniqueID.		
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-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 the process is still triggered.	Misleading status of VIM account on UI.	N/A
RIFT-26409	NS in launchpad didn't fail after alarm triggered in AWS for running instance	DZS 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 DZS Cloud	Launchpad provides the option to create custom user accounts in VMs from DZS Cloud Orchestration	It is not possible to create custom user accounts in VMs with DZS Cloud Orchestration. User	Once the VM is running, user accounts can be created by logging into the VM and

Support Tickets	Title	Description	Impact	Workaround
	Orchestration isn't working.	using cloud-init. This is not working correctly.	accounts must be created directly in the VM after logging into Launchpad.	using the operator interface.
RIFT-26567	Placement groups: transaction failure prints 5 traps 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	DZS Cloud Orchestration 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 is not occurring correctly.	A user must depend on the monitoring params to check the state of VMs.
RIFT-27186	Service instantiation/scale-out fails with incorrect timeout error if invalid input given	DZS Cloud Orchestration 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	If a user choses the management interface type in a VNFD as VDU or Connection Point and	If a user chooses the mgmt-interface type but no ID is selected, then DZS Cloud Orchestration	If a user configures the NS properly, then an issue will not occur.

Support Tickets	Title	Description	Impact	Workaround
	impossible to tell why	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.	considers both the type and the ID as empty.	
RIFT-32998	Discovered VIM Resources data isn't synced between the pairs.	The data discovered through the Discovery mechanism does not persist after a restart or failover. The data can be retrieved by running the discovery operation when required.	The discovered data should not be reused for any operation done after a certain period since the data would be stale. An operator must run discovery again.	Trigger the discovery process again.
RIFT-31236	AWS EKS | Service discovery information is not getting populated in /config	When an operator installs DZS Cloud Orchestration using helm with "--generate-name" option, instead of explicitly specifying the release name or when the provided release name has "launchpad" as part of the name. This results in an incorrect handling of service discovery in a DZS Cloud Orchestration prelaunch container for Launchpad.	Launchpad does not start or starts up incorrectly.	Do not use the --generate-- name option which generates the release name with "launchpad" string in it. Do not set the release name with "launchpad" as part of the string when setting the release name explicitly