Managing VNF Snapshot

Managing VNF Snapshots

A snapshot is a mechanism that allows the creation of a new image on OpenStack from a running Instance. VNF Snapshots mainly serves two purposes:

- As a backup mechanism: Save the main disk of the instance to an image and later boot a new instance from this image with saved data.
- As a templating mechanism: Customize a base image and save it to use as a template for new instances.

The full lifecycle of a VNF snapshot can be managed using ETSI-defined APIs.

Notes and Limitations:

Before using the ETSI APIs for VNF Snapshots, it is important to understand the following points:

- There are no changes required for the VNF Descriptor files to use VNF snapshots.Snapshot functionality exists for VNFs deployed against an OpenStack VIM. If a snapshot is attempted for a VNF deployed on a non-OpenStack VIM such as CVIM or VMWare, then the appropriate error message is generated.
- As per ETSI specifications, the API root is only available under the new, "v2" URL, that is http://192.168.201.33:8250/or_vnfm/vnflcm/v2/vnf_snapshots for SOL003 APIs or http://192.168.201.33:8250/ve_vnfm/vnflcm/v2/vnf_snapshots for SOL002 APIs.
- If a VNF uses one or more volumes that are either VNF-managed volumes or out-of-band volumes, then a resultant snapshot of the VNF results in image and volume snapshot resources generated on OpenStack.
- Deletion of a VNF within ETSI does not trigger deletion of any previous snapshots taken of the VNF. Therefore, delete the VNF Snapshots before deletion of the parent VNF.

API Resources for Snapshot Management:

Create, Query, Revert and Delete the VNF Snapshots using the ETSI APIs.

VNF Snapshot Creation:

The creation of a snapshot with the associated resources generated on OpenStack is a two-step process:

- Creating a snapshot resource

- Creating the snapshot given a snapshot resource ID and existing VNF Instance ID

API Execution

The following shows the operations, sample payloads, and the API responses using Linux curl as a client, executing the APIs on the ESC VM that is a local host itself:

- Create a snapshot resource - note the returned "id" value

```
[admin@host]$ curl -s --user 'admin:*******' -X POST --data {} -H
'Content-Type:application/json' http://localhost:8250/or_vnfm/vnflcm/v2/vnf_snapshots  |
python -m json.tool
{
    "_links": {
        "self": {
            "href":
"http://localhost:8250/or_vnfm/vnflcm/v2/vnf_snapshots/fc7f055c-a541-4801-9295-299ce806763f"

        }
    },
    "id": "fc7f055c-a541-4801-9295-299ce806763f"
}
```

- Create the snapshot given snapshot resource ID and an existing VNF Instance ID

```
[admin@host]$ cat create_snapshot.json
{
    "vnfSnapshotInfoId": "fc7f055c-a541-4801-9295-299ce806763f"
}

[admin@host]$ curl -s --user 'admin:*******' -X POST --data @create_snapshot.json  -H
'Content-Type:application/json'
http://localhost:8250/or_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5/create_snapshot
```

The payload must contain the VNF Snapshot ID from the earlier operation, and the VNF Instance ID in the URL must refer to an INSTANTIATED VNF.

Error Conditions:

- An error returns if the VNF Snapshot ID or the VNF Instance ID are invalid.
- OpenStack-specific errors return if the OpenStack is unreachable or if the resource quotas exceed.
- The ETSI services rely on all other ESC services to operate, otherwise there are connectivity-related errors.

Notifications Generated:

There are no notifications generated when the snapshot resource is created.

When the snapshot is created on OpenStack, three notifications are generated for the three operational states namely: STARTING, PROCESSING, and COMPLETED as shown:

```
{
    "vnfInstanceId": "c9cdf5c8-3681-4641-ba7e-df40539815b5",
    "timeStamp": "2022-07-20T15:08:43.089Z",
    "isAutomaticInvocation": false,
    "notificationType": "VnfLcmOperationOccurrenceNotification",
    "operationState": "STARTING",
    "notificationStatus": "START",
    "vnfLcmOpOccId": "ecbbdc92-a38a-4aed-bc7c-acf0df1a5b92",
```

```
    "_links": {
        "vnfInstance": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5"

        },
        "vnfLcmOpOcc": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/vnf_lcm_op_occs/ecbbdc92-a38a-4aed-bc7c-acf0df1a5b92"

        },
        "subscription": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/subscriptions/900c511f-27e7-4819-aa8d-1fae527caa85"

        }
    },
    "subscriptionId": "900c511f-27e7-4819-aa8d-1fae527caa85",
    "operation": "CREATE_SNAPSHOT",
    "id": "640804b1-2564-4020-af72-16b70d6ac83d"
}

{
    "vnfInstanceId": "c9cdf5c8-3681-4641-ba7e-df40539815b5",
    "timeStamp": "2022-07-20T15:08:43.798Z",
    "isAutomaticInvocation": false,
    "notificationType": "VnfLcmOperationOccurrenceNotification",
    "operationState": "PROCESSING",
    "notificationStatus": "START",
    "vnfLcmOpOccId": "ecbbdc92-a38a-4aed-bc7c-acf0df1a5b92",
    "_links": {
        "vnfInstance": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5"

        },
        "vnfLcmOpOcc": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/vnf_lcm_op_occs/ecbbdc92-a38a-4aed-bc7c-acf0df1a5b92"

        },
        "subscription": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/subscriptions/900c511f-27e7-4819-aa8d-1fae527caa85"

        }
    },
    "subscriptionId": "900c511f-27e7-4819-aa8d-1fae527caa85",
    "operation": "CREATE_SNAPSHOT",
    "id": "6907ac6f-41e4-4bb6-9d31-83f9e809b933"
}

{
    "vnfInstanceId": "c9cdf5c8-3681-4641-ba7e-df40539815b5",
    "timeStamp": "2022-07-20T15:09:02.773Z",
    "isAutomaticInvocation": false,
    "notificationType": "VnfLcmOperationOccurrenceNotification",
    "operationState": "COMPLETED",
    "notificationStatus": "RESULT",
    "vnfLcmOpOccId": "ecbbdc92-a38a-4aed-bc7c-acf0df1a5b92",
    "_links": {
        "vnfInstance": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5"
```

```
        },
        "vnfLcmOpOcc": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/vnf_lcm_op_occs/ecbbdc92-a38a-4aed-bc7c-acf0df1a5b92"

        },
        "subscription": {
            "href":
"https://192.168.10.50:8251/or_vnfm/vnflcm/v2/subscriptions/900c511f-27e7-4819-aa8d-1fae527caa85"

        }
    },
    "subscriptionId": "900c511f-27e7-4819-aa8d-1fae527caa85",
    "operation": "CREATE_SNAPSHOT",
    "id": "de25c769-4264-4fa3-a61f-2aae960c6b60"
}
```


OpenStack Resources Generated:

Upon successful completion of the operation and receiving the final notification, the following resources are created in OpenStack:

IMAGE

Create an image for every VM within the VNF. For example, if the VNF contains two VDUs, then two images are created in OpenStack.

The images have the name of the auto-generated VNFC Snapshot, a UUID-type value. For example

```
[admin@host]$ openstack image list
+--------------------------------------+--------------------------------------------------+--------+
| ID                                   | Name
 | Status |
+--------------------------------------+--------------------------------------------------+--------+
| 92e144ae-24fc-49a5-8622-bb224f1e55cd | eac61a66-51d2-47dd-b8f4-289f38203eff
 | active |
```

Note Note: Find both the image ID and its UUID-like name in the VNF Snapshot query output, explained in the “Query VNF Snapshot” section

VOLUME SNAPSHOT:

Create a volume snapshot for every volume within the VNF. For example, if the VNF contains two VDUs within two volumes each, then four volume snapshots are created in OpenStack.

The volume snapshots have the name of the auto-generated VNFC Snapshot which is a UUID type value prepended by “snapshot for “. For example:

```
[admin@host]$ openstack volume snapshot list
+--------------------------------------+-----------------------------------------+-----------+---------+------+
| ID                                   | Name
 | Description | Status    | Size |
+--------------------------------------+-----------------------------------------+-----------+---------+------+
| 503c348d-94f1-4351-85ec-686b4a21589c | snapshot for eac61a66-51d2-47dd-b8f4-289f38203eff
 | None        | available |    1 |
```

Note Find both the volume snapshot ID and the UUID portion of its name in the VNF Snapshot query output, explained in the “Query VNF Snapshot” section

Query VNF Snapshot:

Use these two main queries to return ETSI VNF Snapshot information:

- Query all VNF Snapshots
- Query a specific VNF Snapshot

API Execution

The following shows both these operations and the API responses, using Linux curl as a client, executing the APIs on the ESC VM that is, the localhost itself:

- Query all VNF Snapshots - an array is returned

```
[admin@host]$ curl -s --user 'admin*******' -X GET -H 'Content-Type:application/json'
http://localhost:8250/or_vnfm/vnflcm/v2/vnf_snapshots | python -m json.tool
[
    {
        "_links": {
            "self": {
                "href":
"http://localhost:8250/or_vnfm/vnflcm/v2/vnf_snapshots/fc7f055c-a541-4801-9295-299ce806763f"

            },
            "takenFrom": {
                "href":
"http://localhost:8250/or_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5"

            }
        },
        "id": "fc7f055c-a541-4801-9295-299ce806763f",
        "vnfSnapshot": {
            "creationFinishedAt": "2022-07-20T15:09:02.588Z",
            "creationStartedAt": "2022-07-20T15:08:43.966Z",
            "id": "0e61b4f8-b347-4d48-80e1-b7a1d28196ef",
            "vnfInstanceId": "c9cdf5c8-3681-4641-ba7e-df40539815b5",
            "vnfdId": "9fb7e4ee-2db1-4aef-bc62-98a2d35d1fa0"
        }
    }
]
```

- Query a specific VNF Snapshot - a single snapshot is returned

```
[admin@host]$ curl -s --user 'admin:cisco123' -X GET -H 'Content-Type:application/json'
http://localhost:8250/or_vnfm/vnflcm/v2/vnf_snapshots/fc7f055c-a541-4801-9295-299ce806763f
 | python -m json.tool
{
    "_links": {
        "self": {
            "href":
"http://localhost:8250/or_vnfm/vnflcm/v2/vnf_snapshots/fc7f055c-a541-4801-9295-299ce806763f"

        },
        "takenFrom": {
            "href":
```

```
"http://localhost:8250/or_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5"

        }
    },
    "id": "fc7f055c-a541-4801-9295-299ce806763f",  <!-- THE VNF SNAPSHOT ID -->
    "vnfSnapshot": {
        "creationFinishedAt": "2022-07-20T15:09:02.588Z",
        "creationStartedAt": "2022-07-20T15:08:43.966Z",
        "id": "0e61b4f8-b347-4d48-80e1-b7a1d28196ef",
        "vnfInstance": {
            "id": "c9cdf5c8-3681-4641-ba7e-df40539815b5",
            "instantiatedVnfInfo": {

<!-- Data deleted as identical to the output from a VNF Instance query -->

        },
        "vnfInstanceId": "c9cdf5c8-3681-4641-ba7e-df40539815b5",  <!-- THE VNF INSTANCE ID
 -->
        "vnfcSnapshots": [
            {
                "computeSnapshotResource": {
                    "resourceId": "92e144ae-24fc-49a5-8622-bb224f1e55cd"  <!-- THE IMAGE
ID -->
                },
                "creationFinishedAt": "2022-07-20T15:09:02.588Z",
                "creationStartedAt": "2022-07-20T15:08:43.966Z",
                "id": "eac61a66-51d2-47dd-b8f4-289f38203eff",   <!-- THE IMAGE NAME AND
VOLUME SNAPSHOT NAME -->
                "storageSnapshotResources": [
                    {
                        "storageResourceId": "res-cfd9a704-0cae-43e2-9880-0b1ba41f2615",
                        "storageSnapshotResource": {
                            "resourceId": "503c348d-94f1-4351-85ec-686b4a21589c"  <!-- THE
 VOLUME SNAPSHOT ID -->
                        }
                    }
                ],
                "vnfcInstanceId": "res-9f5401e3-0129-4657-8ef7-18da424fd369",  <!-- NEEDED
 IF USING THE SOL002 API -->
                "vnfcResourceInfoId": "res-9f5401e3-0129-4657-8ef7-18da424fd369"
            },
        ],
        "vnfdId": "9fb7e4ee-2db1-4aef-bc62-98a2d35d1fa0"
    }
}
```

Reverting to VNF Snapshot using SOL002 or SOL003 APIs:

Users can perform a revert to VNF snapshot lifecycle management operation to return to a previous version of the VNF.

API Execution:

The following shows a sample payload to revert to VNF snapshot:

- Revert to VNF snapshot giving snapshot resource id and the VNF Instance ID

```
[admin@host]$ cat revert_snapshot.json
{
    "vnfSnapshotInfoId": "fc7f055c-a541-4801-9295-299ce806763f"
}

[admin@host]$ curl -s --user 'admin:*******' -X POST --data @revert_snapshot.json -H
```

```
'Content-Type:application/json'
http://localhost:8250/or_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5/revert_to_snapshot
```

Note The SOL002 API root uses ve_vnfm, not or_vnfm.

The following sample payload, restricts the revert to a single VNFC

SOL002:

```
{
    "vnfSnapshotInfoId": "fc7f055c-a541-4801-9295-299ce806763f",
    "vnfcInstanceId": "res-9f5401e3-0129-4657-8ef7-18da424fd369",
    "vnfcSnapshotInfoId": "eac61a66-51d2-47dd-b8f4-289f38203eff"
}
```

SOL003 using Additional Parameters:

```
{
    "vnfSnapshotInfoId": "fc7f055c-a541-4801-9295-299ce806763f",
    "additionalParams": {
        "vnfcInstanceId": "res-9f5401e3-0129-4657-8ef7-18da424fd369",
        "vnfcSnapshotInfoId": "eac61a66-51d2-47dd-b8f4-289f38203eff"
    }
}
```

Notifications Generated:

When the snapshot is reverted on OpenStack, three notifications are generated for the three operational states namely: STARTING, PROCESSING, and COMPLETED.

Notes and Limitations:

- Revert to snapshot does not currently support snapshots of VM without a bootable volume.
- It is not possible to revert a snapshot with OOB volumes.
- OpenStack prevents the deletion of volumes if they have a volume snapshot, ESC attempts to delete the volumes during the revert but these are left on the VIM.

VNF Snapshot Deletion:

Deleting a VNF snapshot involves the single URL as shown:

```
[admin@host]$ curl --user 'admin:********' -X DELETE
http://localhost:8250/or_vnfm/vnflcm/v2/vnf_snapshots/fc7f055c-a541-4801-9295-299ce806763f
```

Note The VNF Snapshot Deletion is a synchronous operation, that is, the API call does not return until the entire workflow in ESC has finished. Deletion takes some time if the VNF Snapshot has to delete multiple VDUs and volumes.

Error Conditions

- A suitable error returns if the VNF Snapshot ID is invalid.
- OpenStack-specific errors return if the OpenStack is unreachable or if the resource quotas exceed.

- The ETSI services rely on all other ESC services to operate, otherwise there are connectivity-related errors.

Notifications Generated:

As per the ETSI Specification, no notifications are generated for a VNF Snapshot delete operation due to its synchronous nature.

Creating a VNF Snapshot using SOL002 APIs:

Create a VNF Snapshot using the SOL002 API, which allows the specification of an individual VNFC ID.

Individual VDUs (VNFCs) within a VNF have snapshots created for them as opposed to taking a VNF Snapshot of the entire VNF.

```
[admin@host]$ cat create_snapshot.json
{
    "vnfSnapshotInfoId": "fc7f055c-a541-4801-9295-299ce806763f",
    "additionalParams": {
        "vnfcInstanceId": "res-9f5401e3-0129-4657-8ef7-18da424fd369"
    }
}
[admin@host]$ curl --user 'admin:********' -X POST --data @create_snapshot.json -H
'Content-Type:application/json'
http://localhost:8250/ve_vnfm/vnflcm/v2/vnf_instances/c9cdf5c8-3681-4641-ba7e-df40539815b5/create_snapshot
```

Note The SOL002 API root uses *ve_vnfm*, not *or_vnfm*.

Error Conditions:

- An error returns if the VNF Snapshot ID, the VNF Instance ID, or the VNFC Instance ID are invalid.
- OpenStack-specific errors return if the OpenStack is unreachable or if the resource quotas exceed.
- The ETSI services rely on all other ESC services to operate, otherwise there are connectivity-related errors.

Notifications generated:

The identical notifications for the SOL003 VNF Snapshot Create are generated.