**VM Lifecycle management in vRTM**

**Various VM status in vRTM :**

1. **Cancelled VM :** VM whose launch is denied by vRTM. In vRTM it’s status represented by 0.
2. **Started VM :**  VM which is running state on openstack. In vRTM it’s status represented by 1.
3. **Stopped VM :** VM which is stopped state on openstack. In vRTM it’s status represented by 2.
4. **Deleted VM :** VM which is deleted on openstack. In vRTM it’s status represented by 3.

**Clean-up interval** and **age of VM with different VM status** can be configured in vRTM configuration file. Clean-up interval is interval after which vRTM should check for; is there any VM whose entry should be removed from vRTM table and it’s related file from trustreport directory.

VM related files is stored at location **/var/lib/nova/trustreports/UUID\_OF\_VM/.** Location **/var/lib/nova/** can be configured in vRTM configuration file.

Testcases :

|  |  |
| --- | --- |
| Test Case | Expected Output |
| **Cancelled VM** ( VM verification fails and Launch policy is ME) | entry for Cancelled VM in vRTM table and related files should present **trustreport/VM\_UUID** dir for time specified or field **cancelled\_vm\_max\_age** in vRTM.cfg. After specified time it should be removed from both places. |
| **Started VM** | Neither the vm entry from vRTM table nor related files such as reports should be removed. |
| **Stopped VM** | entry for Stopped VMs in vRTM table and related files should present **trustreport/VM\_UUID** dir and vRTM should not remove these VM entries from its table.  And vRTM should not generate report for stopped VMs. |
| **Deleted VM** | entry for Deleted VM in vRTM table and related files from **trustreport/VM\_UUID** should be removed immediately. |

VM entry removal from vRTM can be seen in vRTM log file **/var/log/vrtm.log** file and it’s related files can be seen under **/var/lib/nova/trustreports/VM\_UUID/.** VM\_UUID is UUID of VM.

If vRTM doesn’t have any entry of VMs’ in cancel state it should not try to remove entries from its table.

**Equivalent Option for windows:**

vRTM will spawn a thread which will continuously monitor, on interval specified in vRTM.cfg, the states of VM in Started(running) or stopped state, so that if VM are deleted then its entry should be removed from vRTM table. State change from stopped to start is not required because when every time VM will transit from stopped to start it will be verified again and its status will updated automatically.

**States of VM in vRTM**

CANCELLED

STOPPED

DELETED

STARTED

**States of VM on Hyper-V:**

Verification passed

Verification fails

Thread discovers

VM is stopped

On restart request

Verification Request

to vRTM

Thread discovers

VM is deleted

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| **Unknown**  0 | The state of the element could not be determined. |
| **Other**  1 |  |
| **Enabled**  2 | The element is running. |
| **Disabled**  3 | The element is turned off. |
| **Shutting Down**  4 | The element is in the process of going to a Disabled state. |
| **Not Applicable**  5 | The element does not support being enabled or disabled. |
| **Enabled but Offline**  6 | The element might be completing commands, and it will drop any new requests. |
| **In Test**  7 | The element is in a test state. |
| **Deferred**  8 | The element might be completing commands, but it will queue any new requests. |
| **Quiesce**  9 | The element is enabled but in a restricted mode. The behavior of the element is similar to the Enabled state (2), but it processes only a restricted set of commands. All other requests are queued. |
| **Starting**  10 | The element is in the process of going to an Enabled state (2). New requests are queued. |

Thread makes WMI call to Hyper-V WMI provider(V2) (available by default from server 2012) to get list of VM on hyper-v. From these list it takes VM state and map it to vRTM state like this:

if VM is in **Enabled(2), Enabled but Offline(6), Deferred(8) or Quiesce(9)** on Hyper-V, it will be mapped to **started or running** state of vRTM, else it will be mapped to **stopped** state of vRTM. If **VM entry is not found**, it will be mapped to **deleted** state of vRTM and its entry from table will be removed immediately.

In case of report generation if VM status is started, it makes WMI call to confirm its latest status and then generates report if it finds VM is actually running, otherwise it will not generate report.

**Configurations in vRTM :**

**Configuration File location : /opt/vrtm/configuration/vRTM.cfg**

**Configurations description :**

IP of vRTM :

rpcore\_ip=127.0.0.1

Port of vRTM :

rpcore\_port=16005

Max Thread limit of vRTM :

max\_thread\_limit=63

Root Directory of vRTM :

vrtm\_root=../

Trust Report Directory :

trust\_report\_dir=/var/lib/nova/trustreports/

Clean-up interval is interval after which vRTM should check for; is there any VM whose entry should be removed from vRTM table and it’s related file from trustreport directory.

#VM Entry cleanup interval in seconds

Sleep time of cleanup thread:

entry\_cleanup\_interval=30

#entries maximum age in seconds

Time after which cancelled VM entry should be removed from vRTM table.

Max Life of cancelled VM :

cancelled\_vm\_max\_age=120

**Log Files locations :**

vrtm\_proxy Log location : /var/log/vrtm/vrtm\_proxy.log

vrtm\_listener Log location : /var/log/vrtm/vrtm\_listener.log

vrtmcore Log location : /var/log/vrtm/vrtm.log

**vRTM’s Log Properties File location :**

vrtmcore and vrtm\_listener log properties file : /opt/vrtm/configuration/vrtm\_log.properties

vrtm\_proxy log properties file : /opt/vrtm/configuration/vrtm\_proxylog.properties