

Table of Contents

Monitoring the OpenNMS Cluster	1
--------------------------------------	---

Monitoring the OpenNMS Cluster

It is very useful to monitor the VMs involved on the cluster solution to be sure they are working fine and be sure they can handle the current load.

The approach for this is creating a new requisition without detectors and one IP policy to avoid discovering IP interfaces. The reason for this is to monitor only what is explicitly defined on the requisition.

The reason for this is because the floating IP (or virtual IP) will be bouncing between the two cluster nodes where *OpenNMS* is running, but the standby node that is not running *OpenNMS* must be monitored.

For this reason the idea is adding a node for the *opennms-cluster* using the *virtual-ip* and the services for the *OpenNMS JVM* and *PostgreSQL*; and then, Adding a node for each machine involved on the cluster to monitor them through SNMP.

Before start, we should configure *Pollerd* and *Collectd* properly.

Edit *poller-configuration.xml*, and replace the service called *Postgres* with the following:

```
<service name="PostgreSQL" interval="300000" user-defined="false" status="on">
  <parameter key="retry" value="1" />
  <parameter key="banner" value="*" />
  <parameter key="port" value="5432" />
  <parameter key="timeout" value="3000" />
</service>
<service name="OpenNMS-PostgreSQL" interval="300000" user-defined="false" status="on">
  <parameter key="retry" value="1" />
  <parameter key="banner" value="*" />
  <parameter key="port" value="9999" />
  <parameter key="timeout" value="3000" />
</service>
...
<monitor service="PostgreSQL"
  class-name="org.opennms.netmgt.poller.monitors.TcpMonitor" />
<monitor service="OpenNMS-PostgreSQL"
  class-name="org.opennms.netmgt.poller.monitors.TcpMonitor" />
```

Edit *collectd-configuration.xml* and change the name of the *PostgreSQL* service to look like the following:

```
<service name="OpenNMS-PostgreSQL" interval="300000" user-defined="false" status="on">
  <parameter key="collection" value="PostgreSQL"/>
  <parameter key="thresholding-enabled" value="true"/>
  <parameter key="driver" value="org.postgresql.Driver"/>
  <parameter key="user" value="opennms"/>
  <parameter key="password" value="opennms"/>
  <parameter key="url" value="jdbc:postgresql://OPENNMS_JDBC_HOSTNAME:9999/opennms"/>
</service>
...
<collector service="OpenNMS-PostgreSQL"
  class-name="org.opennms.netmgt.collectd.JdbcCollector"/>
```

Finally restart *OpenNMS*. Remember to use the cluster service to perform the restart and avoid using the *opennms* script.

Now, proceed to configure the requisition and start monitoring the cluster.

1. Install *SNMP* on all the machines.

```
[root@onmssrv01 ~]# yum install net-snmp net-snmp-utils -y
```

1. Configure the internal firewall to accept connections on port *UDP 161*, on all the machines.

On *RHEL/CentOS 6*:

Edit the */etc/sysconfig/iptables* file on all the servers, and then add the following rules before the first **REJECT** entry:

```
-A INPUT -p udp --dport 161 -j ACCEPT
```

Then, restart iptables on all the servers:

```
[root@onmssrv01 ~]# service iptables restart
```

On *RHEL/CentOS 7*:

Create a file called */etc/firewalld/services/snmp.xml* on all the servers with the following content:

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>snmp</short>
  <description>SNMP Simple Network Management Protocol</description>
  <port protocol="udp" port="161"/>
</service>
```

Then, execute the following commands on all the servers:

```
[root@onmssrv01 ~]# firewall-cmd --reload
[root@onmssrv01 ~]# firewall-cmd --permanent --add-service=snmp
[root@onmssrv01 ~]# firewall-cmd --add-service=snmp
```

1. Configure *SNMP* by editing */etc/snmp/snmp.conf* and adding the following settings, on all the machines.

```
com2sec onmsUser 192.168.205.0/24 0penNMS!
group onmsGroup v1 onmsUser
group onmsGroup v2c onmsUser
view all included .1 80
access onmsGroup "" any noauth 0 all none none
```

1. Enable and start the *SNMP* service on all the machines.

On *RHEL/CentOS 6*:

```
[root@onmssrv01 ~]# chkconfig snmpd on
[root@onmssrv01 ~]# service snmpd start
```

On *RHEL/CentOS 7*:

```

---
[root@onmssrv01 ~]# systemctl enable snmpd
[root@onmssrv01 ~]# systemctl start snmpd
--

```

1. Configure the *SNMP* community in *OpenNMS*.

From the cluster node on which *OpenNMS* is running, execute the following commands to configure the *SNMP* community:

```

[root@onmssrv01 ~]# for num in 151 152 153 154 155
> do
> /opt/opennms/bin/provision.pl --username admin --password admin snmp set 192.168.205.$num 0penNMS!
version=2c
> done

```

The community string is *0penNMS!*, and the version is *2c*.

1. Create the requisition.

```

[root@onmssrv01 ~]# PROVISION="/opt/opennms/bin/provision.pl --username admin --password admin"
[root@onmssrv01 ~]# $PROVISION requisition add OpenNMS
[root@onmssrv01 ~]# $PROVISION node add OpenNMS onms-cluster onms-cluster
[root@onmssrv01 ~]# $PROVISION interface add OpenNMS onms-cluster 192.168.205.150
[root@onmssrv01 ~]# $PROVISION interface set OpenNMS onms-cluster 192.168.205.150 snmp-primary N
[root@onmssrv01 ~]# $PROVISION service add OpenNMS onms-cluster 192.168.205.150 OpenNMS-JVM
[root@onmssrv01 ~]# $PROVISION service add OpenNMS onms-cluster 192.168.205.150 OpenNMS-PostgreSQL
[root@onmssrv01 ~]# $PROVISION node add OpenNMS onmssrv01 onmssrv01
[root@onmssrv01 ~]# $PROVISION interface add OpenNMS onmssrv01 192.168.205.151
[root@onmssrv01 ~]# $PROVISION interface set OpenNMS onmssrv01 192.168.205.151 snmp-primary P
[root@onmssrv01 ~]# $PROVISION node add OpenNMS onmssrv02 onmssrv02
[root@onmssrv01 ~]# $PROVISION interface add OpenNMS onmssrv02 192.168.205.152
[root@onmssrv01 ~]# $PROVISION interface set OpenNMS onmssrv02 192.168.205.152 snmp-primary P
[root@onmssrv01 ~]# $PROVISION node add OpenNMS pgdbsrv01 pgdbsrv01
[root@onmssrv01 ~]# $PROVISION interface add OpenNMS pgdbsrv01 192.168.205.153
[root@onmssrv01 ~]# $PROVISION interface set OpenNMS pgdbsrv01 192.168.205.153 snmp-primary P
[root@onmssrv01 ~]# $PROVISION node add OpenNMS pgdbsrv02 pgdbsrv02
[root@onmssrv01 ~]# $PROVISION interface add OpenNMS pgdbsrv02 192.168.205.154
[root@onmssrv01 ~]# $PROVISION interface set OpenNMS pgdbsrv02 192.168.205.154 snmp-primary P
[root@onmssrv01 ~]# $PROVISION node add OpenNMS nfssrv01 nfssrv01
[root@onmssrv01 ~]# $PROVISION interface add OpenNMS nfssrv01 192.168.205.155
[root@onmssrv01 ~]# $PROVISION interface set OpenNMS nfssrv01 192.168.205.155 snmp-primary P

```

Here is how the requisition looks like:

```

[root@onmssrv01 ~]# $PROVISION list
* OpenNMS (last updated: 2015-07-30T00:53:13.067Z)
  * nodes:
    * onms_cluster (foreign ID: onms_cluster)
      * building: OpenNMS
      * interfaces:
        * 192.168.205.150 (Virtual)
          * services:
            * OpenNMS-PostgreSQL
            * OpenNMS-JVM
          * SNMP Primary: N
          * Status: 1
    * nfssrv01 (foreign ID: nfssrv01)
      * building: OpenNMS
      * interfaces:
        * 192.168.205.155 (LAN)
          * SNMP Primary: P
          * Status: 1
    * pgdbsrv02 (foreign ID: pgdbsrv02)
      * building: OpenNMS
      * interfaces:
        * 192.168.205.154 (LAN)
          * SNMP Primary: P
          * Status: 1
    * pgdbsrv01 (foreign ID: pgdbsrv01)
      * building: OpenNMS
      * interfaces:
        * 192.168.205.153 (LAN)
          * SNMP Primary: P
          * Status: 1
    * onmssrv02 (foreign ID: onmssrv02)
      * building: OpenNMS
      * interfaces:
        * 192.168.205.152 (LAN)
          * SNMP Primary: P
          * Status: 1
    * onmssrv01 (foreign ID: onmssrv01)
      * building: OpenNMS
      * interfaces:
        * 192.168.205.151 (LAN)
          * SNMP Primary: P
          * Status: 1

```

As you can see, the cluster has `snmp-primary = N` and it has two services: *OpenNMS-JVM* (to monitor the running *JVM* on the active node) and *OpenNMS-PostgreSQL* (to monitor the *PostgreSQL* activity through *pgpool-II*).

1. Create the foreign source definition.

Create an *XML* at `/tmp/OpenNMS.xml` with the following content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<foreign-source xmlns="http://xmlns.opennms.org/xsd/config/foreign-source" name="OpenNMS" date-
stamp="2015-07-30T00:53:24.561Z">
  <scan-interval>1w</scan-interval>
  <detectors>
    <detector name="ICMP" class="org.opennms.netmgt.provision.detector.icmp.IcmpDetector"/>
    <detector name="SNMP" class="org.opennms.netmgt.provision.detector.snmp.SnmpDetector">
      <parameter key="ipMatch" value="192.168.205.151-155"/>
    </detector>
    <detector name="PostgreSQL" class="org.opennms.netmgt.provision.detector.simple.TcpDetector">
      <parameter key="port" value="5432"/>
    </detector>
  </detectors>
  <policies>
    <policy name="NoDiscoveredIPs"
class="org.opennms.netmgt.provision.persist.policies.MatchingIpInterfacePolicy">
      <parameter key="action" value="DO_NOT_PERSIST"/>
      <parameter key="matchBehavior" value="NO_PARAMETERS"/>
    </policy>
  </policies>
</foreign-source>
```

Pay attention to the *SNMP* service. It is excluding the detection for the *VIP* address (i.e. the *SNMP* service will be detected only on the the nodes where their IPs are in the range that starts on 192.168.205.151 and ends on 192.168.205.155).

Then, push it to *OpenNMS*:

```
[root@onmssrv01 ~]# cd /tmp
[root@onmssrv01 ~]# curl -v -d @OpenNMS.xml -u "admin:admin"
http://localhost:8980/opennms/rest/requisitions/foreignSources
```

1. Synchronize the requisition

```
[root@onmssrv01 ~]# $PROVISION requisition import OpenNMS
```

Upgrading *OpenNMS* on a cluster environment

WARNING | this is a work in progress...

In general it is extremely important to disable automatic updates through *YUM*. Any kind of upgrade procedure must be performed by an administrator.

There are several possible upgrade scenarios. Each of them depend on which component you want to upgrade.