GEEKPEEK.net

LINUX

Linux Cluster Part 3 – Manage Cluster Nodes and Resources

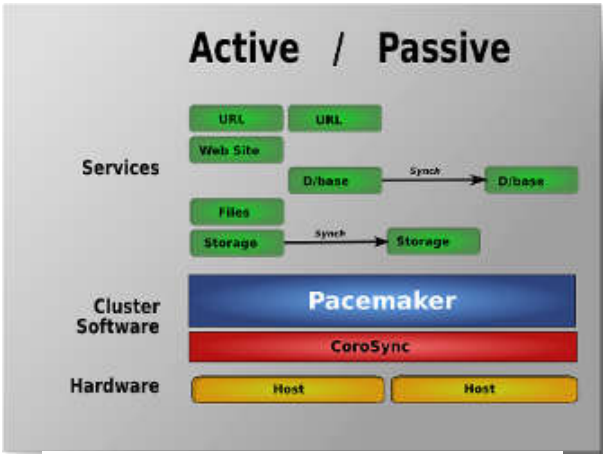
November 13, 2013, 16:34 10 Comments

This is the third part of my "Linux Cluster" posts:

Linux Cluster Part 1 – Install Corosync and Pacemaker on CentOS 6 – Learn how to install Corosync and Pacemaker on CentOS 6

Linux Cluster Part 2 – Adding and Deleting Cluster Resources – Learn how to add and delete Linux Cluster Resources and how to use CRM Shell

Linux Cluster Part 3 – Manage Cluster Nodes and Resources – Learn how to manage Linux Cluster Resources (resource constraints – group, order, colocation, …) and Learn how to manage Linux Cluster Nodes (maintenance mode, standby mode, …).



Cluster Nodes

In Linux Cluster Part 3 post we will continue to manage our Linux Cluster Nodes and Resources from Linux

Cluster Part 2!

Pre-configured **resources are ClusterIP and Apache** on **nodes foo1.geekpeek.net and foo2.geekpeek.net**.

1. Cluster Node Management

CRM Shell is also used for Linux Cluster node management using "*crm node*" commands.

The following examples cover the **basic Linux Cluster node management commands i usually use**. Additional help is available by executing "*crm node help*" command! Take note, that all changes made with "crm node" commands are saved as Linux Cluster Node attributes – if we want to remove it we must run "*crm node attribute nodename delete attribute*".

- List Cluster Nodes – Lists the Linux Cluster nodes – "*crm node list*"

```
[root@foo1 ~]# crm node list
foo1.geekpeek.net: normal
foo2.geekpeek.net: normal
```

- Maintenance Mode – Puts Linux Cluster node in maintenance mode – "*crm node maintenance nodename*"

```
[root@foo1 ~]# crm node maintenance foo1.geekpeek.net
[root@foo1 ~]# crm node status
<nodes>
  <node id="foo1.geekpeek.net" uname="foo1.geekpeek.net">
    <instance_attributes id="nodes-foo1.geekpeek.net">
      <nvpair id="nodes-foo1.geekpeek.net-maintenance"
name="maintenance" value="on"/>
    </instance_attributes>
  </node>
  <node id="foo2.geekpeek.net" uname="foo2.geekpeek.net"/>
</nodes>
```

Once we put a Linux Cluster node into a Maintenance Mode we need to run "*crm node ready nodename*" to get it back online!

- Ready Mode – Returns Linux Cluster node from maintenance mode – "*crm node ready nodename*"

```
[root@foo1 ~]# crm node ready foo1.geekpeek.net
[root@foo1 ~]# crm node status
<nodes>
  <node id="foo1.geekpeek.net" uname="foo1.geekpeek.net">
    <instance_attributes id="nodes-foo1.geekpeek.net">
      <nvpair id="nodes-foo1.geekpeek.net-maintenance"
name="maintenance" value="off"/>
    </instance_attributes>
  </node>
  <node id="foo2.geekpeek.net" uname="foo2.geekpeek.net"/>
</nodes>
```

- Show/Delete/Set Node Attribute – Shows/Deletes/Sets the desired attributes set on Linux Cluster node – "*crm node attribute nodename show/delete/set attribute*"

```
[root@foo1 ~]# crm node attribute foo1.geekpeek.net delete maintenance
Deleted nodes attribute: id=nodes-foo1.geekpeek.net-maintenance
name=maintenance
[root@foo1 ~]# crm node status
<nodes>
  <node id="foo1.geekpeek.net" uname="foo1.geekpeek.net">
    <instance_attributes id="nodes-foo1.geekpeek.net"/>
  </node>
  <node id="foo2.geekpeek.net" uname="foo2.geekpeek.net"/>
</nodes>
```

- Standby Mode – Puts the Linux Cluster node into a Standby mode – "*crm node standby nodename*"

```
[root@foo1 ~]# crm node standby foo1.geekpeek.net
[root@foo1 ~]# crm node status
<nodes>
  <node id="foo1.geekpeek.net" uname="foo1.geekpeek.net">
    <instance_attributes id="nodes-foo1.geekpeek.net">
      <nvpair id="nodes-foo1.geekpeek.net-standby" name="standby"
value="on"/>
    </instance_attributes>
  </node>
  <node id="foo2.geekpeek.net" uname="foo2.geekpeek.net"/>
</nodes>
```

- Online Mode – Returns Linux Cluster node to Online mode from Standby – "*crm node online nodename*"

```
[root@foo1 ~]# crm node online foo1.geekpeek.net
[root@foo1 ~]# crm node status
<nodes>
  <node id="foo1.geekpeek.net" uname="foo1.geekpeek.net">
    <instance_attributes id="nodes-foo1.geekpeek.net">
      <nvpair id="nodes-foo1.geekpeek.net-standby" name="standby"
value="off"/>
    </instance_attributes>
  </node>
  <node id="foo2.geekpeek.net" uname="foo2.geekpeek.net"/>
</nodes>
```

2. Cluster Resource Management

CRM Shell is used for Linux Cluster management. We can use "*crm configure*" with "*group, order, location, colocation, ...*" parameters and "*crm resource*" with "*start, stop, status, migrate, cleanup, ...*".

The following examples cover the **basic Linux Cluster resource management commands you might find useful**. Additional help is available by executing "*crm configure help*" or "*crm resource help*" command.

Our current Linux Cluster resource configuration is:

```
[root@foo1 ~]# crm configure show
node foo1.geekpeek.net
node foo2.geekpeek.net
primitive Apache ocf:heartbeat:apache
    params configfile="/etc/httpd/conf/httpd.conf"
    op monitor interval="30s"
    op start timeout="40s" interval="0"
    op stop timeout="60s" interval="0"
primitive ClusterIP ocf:heartbeat:IPaddr2
    params ip="192.168.1.150" cidr_netmask="24"
    op monitor interval="30s"
property $id="cib-bootstrap-options"
    dc-version="1.1.10-1.el6_4.4-368c726"
    cluster-infrastructure="classic openais (with plugin)"
    expected-quorum-votes="2"
    stonith-enabled="false"
    last-lrm-refresh="1383902488"
```

- Group Linux Cluster Resources "*crm configure group groupname resource1 resource2"*

Group your Linux Cluster resources and start/stop and manage your resource group with one single command.

```
[root@foo1 ~]# crm configure group HTTP-GROUP ClusterIP Apache
[root@foo1 ~]# crm configure show
 node foo1.geekpeek.net
 node foo2.geekpeek.net
 primitive Apache ocf:heartbeat:apache
 params configfile="/etc/httpd/conf/httpd.conf"
 op monitor interval="30s"
 op start timeout="40s" interval="0"
 op stop timeout="60s" interval="0"
```

```
op stop timeout="60s" interval="0"
primitive ClusterIP ocf:heartbeat:IPaddr2
params ip="192.168.1.150" cidr_netmask="24"
op monitor interval="30s"
group HTTP-GROUP ClusterIP Apache
property $id="cib-bootstrap-options"
dc-version="1.1.10-1.el6_4.4-368c726"
cluster-infrastructure="classic openais (with plugin)"
expected-quorum-votes="2"
stonith-enabled="false"
last-lrm-refresh="1383902488"
```

In this example we created a resource group called HTTP-GROUP with ClusterIP and Apache resources. We can now manage all our grouped resources by starting, stopping and managing HTTP-GROUP group resource.

- Linux Cluster Resources Start/Stop Order "*crm configure order ordername inf: resource1 resource2:start*"

With this command we can configure start and stop order of our Linux Cluster resources.

```
[root@foo1 ~]# crm configure order ClusterIP-before-Apache inf:
ClusterIP Apache:start
[root@foo1 ~]# crm configure show
node foo1.geekpeek.net
node foo2.geekpeek.net
primitive Apache ocf:heartbeat:apache
    params configfile="/etc/httpd/conf/httpd.conf"
    op monitor interval="30s"
    op start timeout="40s" interval="0"
    op stop timeout="60s" interval="0"
primitive ClusterIP ocf:heartbeat:IPaddr2
```

```
    params ip="192.168.1.150" cidr_netmask="24"
    op monitor interval="30s"
order ClusterIP-before-Apache inf: ClusterIP Apache:start
property $id="cib-bootstrap-options"
    dc-version="1.1.10-1.el6_4.4-368c726"
    cluster-infrastructure="classic openais (with plugin)"
    expected-quorum-votes="2"
    stonith-enabled="false"
    last-lrm-refresh="1383902488"
```

In this example we configured the start and stop order of our ClusterIP and Apache resources. As configured, ClusterIP resource will start first and only then Apache resource can be started. When stopping, Apache resource will be stopped and only then ClusterIP resource can be stopped too.

- Linux Cluster Resources Colocation "*crm configure colocation colocationname inf: resource1 resource2*"

We can configure Linux Cluster resources colocation. Like said we colocate the desired resources and make sure we always run desired resources on the same node at all time.

```
[root@foo1 ~]# crm configure colocation IP-with-APACHE inf: ClusterIP
Apache
[root@foo1 ~]# crm configure show
node foo1.geekpeek.net
node foo2.geekpeek.net
primitive Apache ocf:heartbeat:apache
    params configfile="/etc/httpd/conf/httpd.conf"
    op monitor interval="30s"
    op start timeout="40s" interval="0"
    op stop timeout="60s" interval="0"
primitive ClusterIP ocf:heartbeat:IPaddr2
    params ip="192.168.1.150" cidr_netmask="24"
```

```
    params ip="192.168.1.190" cidr_netmask="24"
    op monitor interval="30s"
group HTTP-GROUP ClusterIP Apache
colocation IP-with-APACHE inf: ClusterIP Apache
order ClusterIP-before-Apache inf: ClusterIP Apache:start
property $id="cib-bootstrap-options"
    dc-version="1.1.10-1.el6_4.4-368c726"
    cluster-infrastructure="classic openais (with plugin)"
    expected-quorum-votes="2"
    stonith-enabled="false"
    last-lrm-refresh="1384349363"
```

In this example, we configured colocation for ClusterIP and Apache resources. ClusterIP and Apache resources will always be started and running together, on the same Linux Cluster node.

- Linux Cluster Resources Prefered Location "*crm configure location locationname resource score: clusternode*"

We can configure a prefered location for our Linux Cluster resources or resource groups. We must always set the location score – ositive values indicate the resource should run on this node. Negative values indicate the resource should not run on this node.

```
[root@foo1 ~]# crm configure location HTTP-GROUP-prefer-FOO1 HTTP-GROUP
50: foo1.geekpeek.net
[root@foo1 ~]# crm configure show
node foo1.geekpeek.net
node foo2.geekpeek.net
primitive Apache ocf:heartbeat:apache
    params configfile="/etc/httpd/conf/httpd.conf"
    op monitor interval="30s"
    op start timeout="40s" interval="0"
    op stop timeout="60s" interval="0"
```

```
primitive ClusterIP ocf:heartbeat:IPaddr2
    params ip="192.168.61.150" cidr_netmask="24"
    op monitor interval="30s"
group HTTP-GROUP ClusterIP Apache
location HTTP-GROUP-prefer-FOO1 HTTP-GROUP 50: foo1.geekpeek.net
colocation IP-with-APACHE inf: ClusterIP Apache
order ClusterIP-before-Apache inf: ClusterIP Apache:start
property $id="cib-bootstrap-options"
    dc-version="1.1.10-1.el6_4.4-368c726"
    cluster-infrastructure="classic openais (with plugin)"
    expected-quorum-votes="2"
    stonith-enabled="false"
    last-lrm-refresh="1384349363"
```

In this example we configured the prefered location of HTTP-GROUP resource group. By configuring score 50, HTTP-GROUP will prefer to run on foo1.geekpeek.net node but will still in case of foo1 failure move to foo2.geekpeek.net. When foo2 recovers, HTTP-GROUP will move back to prefered foo1.geekpeek.net.

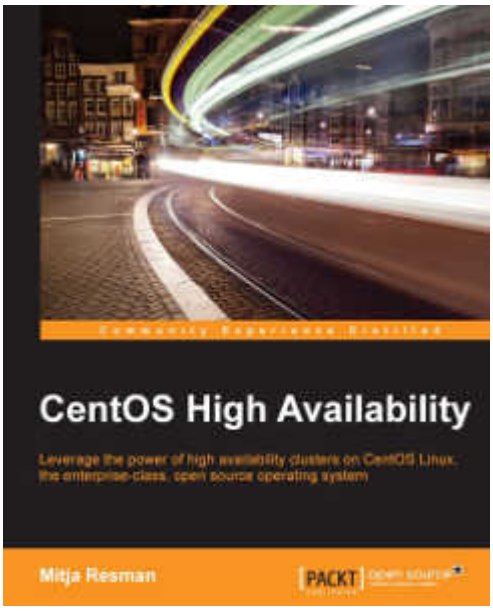
Checking the status of our Linux Cluster Nodes and Resources:

```
[root@foo1 ~]# crm status
Last updated: Wed Nov 13 15:30:45 2013
Last change: Wed Nov 13 15:01:06 2013 via cibadmin on foo1.geekpeek.net
Stack: classic openais (with plugin)
Current DC: foo2.geekpeek.net - partition with quorum
Version: 1.1.10-1.el6_4.4-368c726
2 Nodes configured, 2 expected votes
2 Resources configured

Online: [ foo1.geekpeek.net foo2.geekpeek.net ]

 Resource Group: HTTP-GROUP
     ClusterIP    (ocf::heartbeat:IPaddr2):    Started foo1.geekpeek.net
     Apache    (ocf::heartbeat:apache):    Started foo1.geekpeek.net
```

Here's my latest book about High Availability on CentOS Linux



Tagged with: centos6 cluster corosync pacemaker

Mitch Mod ➚ Md Shariful Islam Saful • 5 years ago

Hi Md Shariful Islam Saful, sure you can add java program or sh script as a service but you should implement a start, stop and monitoring init script to go with. This is not a hard task so should be nor problems. Regards, Mitch

^ | v • Reply • Share ›

canoy • 5 years ago

Mitch -- I'm tasked to add Apache next week. I was wondering if I can add Apache to an existing resource group (using OCF as your example on part 2) without "op start" and "op stop" to the Apache resource property. It will still have the "op monitor timeout" in place (60s).

Can you clarify what "op start" and "op stop" are for...?

This Apache resource will be added as the last item in the existing group. I'm assuming Apache will be the last resource to stop, but first to start. Or should it be 2nd to the last (VIP stops last) and starts first.?

^ | v • Reply • Share ›

Andrej Trobentar • 5 years ago

TOP articles, keep posting Mitch, thanks!

^ | v • Reply • Share ›

Mitch ➚ Andrej Trobentar • 5 years ago

Thanks for support Andrej!

^ | v • Reply • Share ›

Slayer • 6 years ago

Nice Post!!!! 2 thumbs up

^ | v • Reply • Share ›

Mitch ➚ Slayer • 6 years ago

Hey Slayer! Thanks for your support, appreciate it! Regards, Mitch

^ | v • Reply • Share ›

david • 6 years ago

Nice article! Keep on writing about clusters!

^ | v • Reply • Share ›

Mitch ➚ david • 6 years ago

Thanks for your support David! Will definitely write some more posts on cluster topic! Regards, Mitch

^ | v • Reply • Share ›

2 comments • 4 years ago

DC — GReat!! thanks <3

1 comment • 5 years ago

scootbakerr — This brings a players gaming experience to a whole new level and it is all in the palm of their

PostgreSQL Synchronous Streaming Database Replication

6 comments • 4 years ago

Van Nguyen — The slave is now read only. How do I change it to read and write?

ZTE Axon Elite 4G International Edition

2 comments • 4 years ago

Michael gearbest.com — this machine is on huge discount, $70, coupon code is ZTEAXONEU.