

Configuring: How do I make regular nodes return an FQDN using 'hostname -f'? Or resolve the hostnames to IPs on the external network?

How do I make regular nodes return an FQDN using 'hostname -f'? Or resolve the hostnames to IPs on the external network?

There are two ways to achieve this goal. For the second way there are two variations.

Make sure that you go through the "Additional Considerations" section at the end of the article (they apply to both solutions).

Note: please refer to Administrator Manual if you're looking to enable FQDN resolution (to an external IP) on the head nodes. The information is in the appendix on CMDaemon configuration file directives. The name of the directive is "ResolveToExternalName".

Brief outline of the end results introduced by the two solutions:

Results of implementing solution #1:

- 'hostname -f' executed on regular nodes returns the external FQDN of the regular node (e.g. nodeXXX.bcmcluster.customer.com)
- the short hostnames (nodeXXX) of all regular nodes (those with a network interface configured on the external network) resolve to their respective IPs on the external network.

Results of implementing solution #2:

- 'hostname -f' executed on regular nodes returns the external FQDN of the regular node (e.g. nodeXXX.bcmcluster.customer.com)
- no changes in the DNS-based hostname resolution (regular nodes continue to resolve themselves in the default manner, i.e.. using the IPs on the local network.)

Detailed description of the two solutions:

Details For Solution #1

Configuring: How do I make regular nodes return an FQDN using 'hostname -f'? Or resolve the hostnames to IPs on the external network?

The steps below result in the following two changes in the cluster's behaviour:

- 'hostname -f' executed on regular nodes will return the external FQDN of the regular node (e.g. nodeXXX.bcmcluster.customer.com)
- the short hostnames (nodeXXX) of all regular nodes (those with a network interface configured on the external network) will resolve to their respective IPs on the external network.

The steps:

1. Create a new node category for the compute nodes connected to the external network (not necessary if all of the compute nodes will be connected to the external network)
2. The administrator must set (for example) **bcmcluster.customer.com** as the domain name of the external network. The extra subdomain visible in the example above ("bcmcluster") is required. If that subdomain is missing, then the compute nodes within the cluster (those consulting the head node as the primary name server) will not be able to resolve hostnames on the external network. This is because the name server running on the head node cannot be informed of the existence of the external network (and will, unsuccessfully, attempt to resolve the hostname on the customer.com network).
3. Determine a new list of search domains for the compute nodes connected to the external network:
 - get the current search domains written in the '/etc/resolv.conf' on an already running regular node
 - change the order of those search domains so that the external domain (example from (item #1) **bcmcluster.customer.com**) will be the first searchdomain in the list
 - set this new search domain ordering in the property "searchdomains" for the category used by the compute nodes that are to have a network interface on the external network
4. After changing the 'externalnet' network type from external to internal (**don't do this yet, that's covered in a following step**) the headnode's '/etc/shorewall/masq' will be rewritten by the CMDaemon (the entries in the CMDaemon-generated section in this file will be removed) and the 'Shorewall' service will be restarted. This change will prohibit external network traffic to be routed back to the compute nodes via the headnode. To fix this, the entries which are normally written by the CMDaemon to the mentioned CMDaemon-generated section should be copied outside of this section (duplicated) before changing the type of the network to internal in the following step. This way, after changing the network type, the CMDaemon will remove the entries in the CMDaemon generated section, but the copy of the entries will persist outside of the CMDaemon-generated section.

Configuring: How do I make regular nodes return an FQDN using 'hostname -f'? Or resolve the hostnames to IPs on the external network?

Entries in question should look similar to the following:
eth0 10.141.0.0/16

5. The administrator must change the network 'type' of the external network from 'external' to 'internal'
6. Commit all changes and reboot the compute node
7. **Optional:** in order for the headnode to resolve the regular node's short hostname to its respective FQDN external domain hostnames/IPs, an additional change in the searchdomain order on the headnode would be required (again, make the (example) bcmcluster.customer.com the first searchdomain on the list of search domains).

These steps result in this behavior:

```
[root@node001 ~]# hostname -f  
node001.bcmcluster.customer.com
```

```
[root@node001 ~]# nslookup node001  
Server: 10.141.255.254  
Address: 10.141.255.254#53  
Name: node001.bcmcluster.customer.com  
Address: 10.150.150.1
```

```
[root@node001 ~]# nslookup node002  
Server: 10.141.255.254  
Address: 10.141.255.254#53  
Name: node002.bcmcluster.customer.com  
Address: 10.150.150.2
```

```
[root@node001 ~]# ping node001  
PING node001.bcmcluster.customer.com (10.150.150.1) 56(84) bytes of data.  
[root@node001 ~]# ping node002  
PING node002.bcmcluster.customer.com (10.150.150.2) 56(84) bytes of data.
```

Details For Solution #2

This solution should be used if the administrator wants the ONLY change on the cluster to be that 'hostname -f' returns the FQDN on the regular node. I.e. the administrator wants to keep the existing behaviour of DNS-based resolution of the regular node's short hostnames (e.g. 'nslookup node001' performed on node002 will continue to resolve to node001.cm.cluster / internal IP).

Aside: in this solution the FQDN can also be 'node001.customer.com' (i.e. there's no need for the extra subdomain)

In this case, only '/etc/hosts' needs to be modified. This is done on the regular nodes that are going to have the modified 'hostname -f' behavior. For example, for node001:

Configuring: How do I make regular nodes return an FQDN using 'hostname -f'? Or resolve the hostnames to IPs on the external network?

```
<nodes-external-ip> node001.customer.com node001
```

This line can be appended to regular nodes using a finalize script (see the Admin manual for details on finalize scripts).

The result is the following behavior:

```
[root@node001 ~]# hostname -f
node001.customer.com
-or (depending on the /etc/hosts entry)-
node001.bcmcluster.customer.com
```

```
[root@node001 ~]# nslookup node001
Server: 10.141.255.254
Address: 10.141.255.254#53
Name: node001.cm.cluster
Address: 10.141.0.1
```

```
[root@node001 ~]# nslookup node002
Server: 10.141.255.254
Address: 10.141.255.254#53
Name: node002.cm.cluster
Address: 10.141.0.2
```

The default `gethostbyname()` system call, as used by, for example, `ping`, uses the order set in `/etc/nsswith.conf`. A small side-effect of the configuration of `/etc/hosts` on the regular nodes as given in the preceding is that for applications that resolve the node's own short hostname to IP using the order of the default system call (i.e. first query the `/etc/hosts`, only afterwards query the DNS server), the node's own short hostname will resolve to the node's external IP:

```
[root@node001 ~]# ping node001
PING node001.customer.com (10.150.150.1) 56(84) bytes of data.
```

However, resolving the short hostname of any other nodes from the same node will resolve to that node's respective internal IP:

```
[root@HEADNODE ~]# ping node001
PING node001.cm.cluster (10.141.0.1) 56(84) bytes of data.
```

```
[root@node001 ~]# ping node002
PING node002.cm.cluster (10.141.0.2) 56(84) bytes of data.
```

```
[root@node001 ~]# hostname -f
node001.customer.com
```

The side-effect can be changed by putting the node's internal IP in the previously-mentioned line in `/etc/hosts`, instead of the external IP, like this:

Configuring: How do I make regular nodes return an FQDN using 'hostname -f'? Or resolve the hostnames to IPs on the external network?

```
<nodes-internal-ip> node001.bcmcluster.customer.com node001
```

In this case the hostname resolution of the node's own short hostname done using the "gethostbyname()" syscall will return the external FQDN, but the resolved IP will be the IP on the internal network:

```
[root@node001 ~]# ping node001
```

```
PING node001.customer.com (10.141.0.1) 56(84) bytes of data. # previously this was '10.150.150.1'
```

```
[root@HEADNODE ~]# ping node001
```

```
PING node001.cm.cluster (10.141.0.1) 56(84) bytes of data.
```

```
[root@node001 ~]# ping node002
```

```
PING node002.cm.cluster (10.141.0.2) 56(84) bytes of data.
```

```
[root@node001 ~]# hostname -f
```

```
node001.customer.com
```

Additional considerations (applies to both solutions):

Changing the default gateway on compute nodes:

Optionally, one can consider configuring compute nodes to use a different default gateway (the one on the external network). This way traffic aimed from the compute nodes at the external network/Internet will not be routed via the headnode. The default gateway property can be found in the node's category settings.

Dealing with an external DHCP server.

If the interface of the compute node which is connected to the external network obtains its IP via DHCP, the 'dhclient-script' (run during system boot, i.e. after the node installer) will overwrite the '/etc/resolv.conf' written previously by the node installer during the node installation process. One way to inhibit this behaviour is by using hooks (this approach is not covered in this article). As an alternative to hooks, one may write out a custom '/etc/dhcp/dhclient.conf' file from within the finalize script in the node installer which will influence the behaviour of the 'dhclient-script'. What follows is an example of the content written to the '/etc/dhcp/dhclient.conf' file:

```
prepend domain-name "domain1 domain2 domain3";
```

```
prepend domain-name-servers 10.141.255.254, 8.8.8.8;
```

the domain1, domain2, domain3 should be the domains which would normally end up in the '/etc/resolv.conf' file (i.e. the ones configured for the node's category). the IP '10.141.255.254' in this example is the example of the headnode, the IP '10.10.1.1' is an example IP of a third part DNS server.

Note, that a slightly different behaviour can be achieved by using the "supersede" keyword

Configuring: How do I make regular nodes return an FQDN using 'hostname -f'? Or resolve the hostnames to IPs on the external network?

instead of 'prepend'. The 'supersede' keyword will make the 'dhclient-script' overwrite the configuration received from the external dhcp server with whatever was specified in the 'dhclient.conf'.

Unique solution ID: #1091

Author: Piotr Wachowicz

Last update: 2013-05-13 11:16