General: How can I configure Isilon's DNS and NFS with Bright?

How can I configure Isilon's DNS and NFS with Bright?

DNS setup for use with Isilon SmartConnect [1]

The DNS running on the Bright head node can be configured in two ways to work with the DNS of the Isilon SmartConnect:

1.If the DNS on Bright head node has nothing to do with the Isilon storage domain.

A forward zone which points to the Isilon DNS should then be defined in /etc/named.conf.include so that it doesn't get overwritten by CMDaemon:

```
zone "isilon.bcm.local" IN {

      type forward;

      forwarders {10.240.0.254;};

}
```

2.If the DNS on the Bright head node manages the parent domain of the Isilon storage domain.

The DNS on the head node should then be configured to delegate queries for isilon.bcm.local by adding the the following entries in the include file of the relevant zone:

```
# cat /var/named/bcm.local.zone.include

[...]

sc-isilon.bcm.local. IN A 10.240.0.254

isilon.bcm.local. IN NS sc-isilon.bcm.local.
```

General: How can I configure Isilon's DNS and NFS with Bright?

[...]

Best Practices for Isilon NFS client settings [2]

1. Use NFS over TCP, not UDP

If using UDP, then if the network speed of the client and server are not identical, dropped packets may cause performance degradation when the faster system sends data to the slower system. For better performance and resilience, use the Transmission Control Protocol (TCP) instead.

2. Use NFS version 3 or 4

NFS version 2 should not be used for mounting because it uses UDP and does not support files larger than 2 GB. NFS version 3 is the standard for NFS and is the most stable and mature version of the NFS protocol. NFS version 4 is still considered new, but is becoming more popular, and is also relatively stable. Version 4 however does not support failover because it is stateful.

3. Use async mounts from the client

Using sync as a client mount option causes all write operations to be synchronous, which usually results in very poor write performance. Sync mounts should only be used in exceptional circumstances where a client program does not specify synchronous writes itself, but relies on that behavior.

4. Configure read and write RPC sizes to be as large as possible

In general, you should configure the read and write remote procedure call (RPC) sizes (rsize, wsize) to be as large as possible. On OneFS 5.0 and later, the server-side limits are:

- rsize: 128 KB
- wsize: 512 KB

The client implementation might limit you or default to lower values. For more information about the client implementation, consult your client documentation.

5. Use "hard" mounts

On most clients, it is recommended to use the hard and intr mount options so that input/output (I/O) errors are retried indefinitely, but can still be interrupted by signals. The hard option retries I/O errors indefinitely; the intr option allows signals to interrupt. In the event that an NFS client cannot access the mounted resource, a hard-mounted client continues to request access to the resource until the request is satisfied.

6. Use the lock mount option to ensure that clients can obtain exclusive access to the files when needed.

The OneFS implementation uses advisory locks, as opposed to mandatory locks, so all clients must be configured to use locking to ensure consistent locking behavior.

Recommended client settings

Modern Linux distributions support NFS read/write block sizes up to 1 MB, and will automatically negotiate the optimal block size with the NFS server. The negotiated values are ideal for properly configured high-performance, low-latency networks. To limit the negotiated read/write record size for all clients, the following sysctls can be set on the cluster:

Like in most operating systems, the default maximum network buffer sizes are too small. The network buffer size parameters must be set on the client before mounting NFS exports.

Recommended network buffer settings for 1 GbE Ethernet connections:

```
echo "262144" > /proc/sys/net/core/wmem_max
```

```
echo "262144" > /proc/sys/net/core/wmem_default
```

```
echo "262144" > /proc/sys/net/core/rmem_max
```

```
echo "262144" > /proc/sys/net/core/rmem_default
```

For a boot-time change, /etc/sysctl.conf should be changed as follows:

```
net.core.wmem_max = 262144
```

```
net.core.wmem_default = 262144
```

```
net.core.rmem_max = 262144
```

```
net.core.rmem_default = 262144
```

Recommended network buffer settings for 10 GbE connected clients by setting these sysctls:

```
net.core.wmem_max = 16777216
```

```
net.core.wmem_default = 524287
```

```
net.core.rmem_max = 16777216
```

```
net.core.rmem_default = 524287
```

General: How can I configure Isilon's DNS and NFS with Bright?

Recommended mount settings for 10 GbE Ethernet connections:

```
mount -t nfs -o nfsvers=3,tcp,rw,hard,intr,timeo=600,retrans=2,isicluster:/ifs /mnt
```

If this is a non-public server, you can turn off TCP timestamps (to reduce CPU processing) by setting this sysctl:

```
net.ipv4.tcp_timestamps=0
```

Make sure window scaling is enabled by setting this sysctl:

```
net.ipv4.tcp_window_scaling=1
```

Set the number of task request slots for RPC calls to 16 by setting this sysctl:

```
sunrpc.tcp_slot_table_entries=16
```

To make these changes permanent, follow the instructions for editing the /etc/sysctl.conf file in *Configuring sysctls and making sysctl changes persist through node and cluster reboots and upgrades,* http://bit.ly/kb-89232.

Adding metric collection for Isilon

Bright provides a metric collection script for Isilon. The script connects to *SmartConnect* and collects the the metrics. It is recommended to add *SmartConnect* as a generic device in CMDaemon and associate the metrics with it using the procedure below.

1. Create the generic device.

General: How can I configure Isilon's DNS and NFS with Bright?

```
# cmsh
% device add generic sc-isilon
% set ip 10.20.10.205
```

% set userdefinedresources isilon   (for Bright 8.0+)

```
% commit
```

2. Set the username and the password of the user privileged to view the statistics in /cm/local/apps/cmd/scripts/metrics/configfiles/sample_isilon.conf. This user should be added on Isilon.

cat /cm/local/apps/cmd/scripts/metrics/configfiles/isilon.conf
#!/usr/bin/perl

```
our $isilon_auth =
{ "username" => 'root',
"password" => 'Password01',
"services" => ["platform","namespace"]
};
our $port = 8080;
our $isilon_host = "sc-isilon";
```

If using Bright prior to version 8.0, use step 3.a:

3.a Add the metric collection script.

% monitoring metrics

```
% add isilon

% set command /cm/local/apps/cmd/scripts/metrics/sample_isilon

% set samplingmethod sampleonheadnode

% set classofmetric prototype

% set timeout 30

% commit
```

If using Bright version 8.0 and later, use step 3.b:

3.b Add the metric collection script.

% monitoring setup

% add collection isilon

% set script /cm/local/apps/cmd/scripts/metrics/sample_isilon

% nodeexecutionfilters

% active

% exit

% executionmultiplexers

% add resource isilon

% set resources isilon

% commit

4. Add the metrics to the sc-isilon device.

```
# cmsh
% monitoring setup metricconf sc-isilon
% add isilon
% set samplinginterval 300
% commit
```

5. If the script authenticates successfully, you should see a message like the one below.

Wed Mar 8 22:36:03 2017 [notice] sc-isilon: Summary for initializing metric collections. Added 3103 new metrics, added 3103 metrics to monitoring configuration

General: How can I configure Isilon's DNS and NFS with Bright?

For details type: events details 128

References

1. *Isilon OneFS: How to create a UNIX-based BIND DNS setup for use with SmartConnect,* https://support.emc.com/kb/89758
2. *OneFS: Best practices for NFS client settings* http://bit.ly/kb-90041

Unique solution ID: #1237
Author: Frank Furter
Last update: 2018-03-15 13:02