

devops - Bug #6335

ceph-deploy may *still* hang with pushy

09/17/2013 05:26 PM - Alfredo Deza

| | |
|-------------------------------|------------------------------|
| Status: Resolved | % Done: 0% |
| Priority: Urgent | |
| Assignee: Alfredo Deza | |
| Category: ceph-deploy | |
| Target version: | |
| Source: other | Affected Versions: |
| Tags: | ceph-qa-suite: |
| Backport: | Pull request ID: |
| Regression: No | Crash signature (v1): |
| Severity: 3 - minor | Crash signature (v2): |
| Reviewed: | |

Description

We need to close the connection as early as possible in an attempt to mitigate probably hanging issues.

```
[root@node-4 ~]# vi ceph-deploy/ceph_deploy/mon.py
```

```
import ConfigParser
import json
import logging
import re
import subprocess
import time

from . import conf
from . import exc
from .cliutil import priority
from .sudo_pushy import get_transport
from .util import paths
from .lib.remoto import process
from . import hosts
from .misc import mon_hosts, remote_shortcode
from .connection import get_connection

LOG = logging.getLogger(__name__)

def mon_status(conn, logger, hostname, silent=False):
    """
    run ``ceph daemon mon.`hostname` mon_status`` on the remote end and provide
    not only the output, but be able to return a boolean status of what is
    going on.
    ``False`` represents a monitor that is not doing OK even if it is up and
    running, while ``True`` would mean the monitor is up and running correctly.
    rconn = get_connection(hostname, logger=logger)

    try:
        out, err, code = process.check(
            rconn,
            ['ceph', 'daemon', mon, 'mon_status'],
            exit=True
        )

        for line in err:
            logger.error(line)
```

```

try:
    mon_info = json.loads(''.join(out))
    return False
if not silent:
    logger.debug('*'*80)
    logger.debug('status for monitor: %s' % mon)
    for line in out:
        logger.debug(line)
    logger.debug('*'*80)
if mon_info['rank'] >= 0:
    logger.info('monitor: %s is running' % mon)
    return True
logger.info('monitor: %s is not running' % mon)
return False
except RuntimeError:
    logger.info('monitor: %s is not running' % mon)
    return False

def mon_create(args):

    cfg = conf.load(args)
    if not args.mon:
        try:
            mon_initial_members = cfg.get('global', 'mon_initial_members')
        except (ConfigParser.NoSectionError,
                ConfigParser.NoOptionError):
            pass
        else:
            args.mon = re.split(r'[,\\s]+', mon_initial_members)

    if not args.mon:
        raise exc.NeedHostError()

    try:
        return False
    except RuntimeError:
        logger.info('monitor: %s is not running' % mon)
        return False

def mon_create(args):

    cfg = conf.load(args)
    if not args.mon:
        try:
            mon_initial_members = cfg.get('global', 'mon_initial_members')
        except (ConfigParser.NoSectionError,
                ConfigParser.NoOptionError):
            pass
        else:
            args.mon = re.split(r'[,\\s]+', mon_initial_members)

    if not args.mon:
        raise exc.NeedHostError()

    try:
        with file('{cluster}.mon.keyring'.format(cluster=args.cluster),
                  'rb') as f:
            monitor_keyring = f.read()
    except IOError:
        raise RuntimeError('mon keyring not found; run \'new\' to create a new cluster')

    LOG.debug(
        'Deploying mon, cluster %s hosts %s',
        args.cluster,
        ' '.join(args.mon),
    )

```

```

errors = 0
for (name, host) in mon_hosts(args.mon):
    try:
        # TODO username
        # TODO add_bootstrap_peer_hint
        LOG.debug('detecting platform for host %s ...', name)
        distro = hosts.get(host)
        LOG.info('distro info: %s %s %s', distro.name, distro.release, distro.codename)
        rlogger = logging.getLogger(name)

        # ensure remote hostname is good to go
        hostname_is_compatible(distro.sudo_conn, rlogger, name)
        rlogger.debug('deploying mon to %s', name)
        distro.mon.create(distro, rlogger, args, monitor_keyring)

        # tell me the status of the deployed mon
        time.sleep(2) # give some room to start
        distro.sudo_conn.close()
        assert False, "distro.sudo_conn.close() finished"
        mon_status(None, rlogger, name)

    except RuntimeError as e:
        LOG.error(e)
        errors += 1

if errors:
    raise exc.GenericError('Failed to create %d monitors' % errors)

def hostname_is_compatible(conn, logger, provided_hostname):
    """
    Make sure that the host that we are connecting to has the same value as the
    `hostname` in the remote host, otherwise mons can fail not reaching quorum.
    "ceph-deploy/ceph_deploy/mon.py" 307L, 8874C written
    [root@node-4 ~]# ps axu | grep ceph | awk '{system("kill "$2)}' && ceph-deploy purge localhost &
    & ceph-deploy purgedata localhost && yum install -y ceph
    sh: line 0: kill: (13979) - No such process
    [ceph_deploy.install][DEBUG ] Purging from cluster ceph hosts localhost
    [ceph_deploy.install][DEBUG ] Detecting platform for host localhost ...
    [ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
    [ceph_deploy.install][DEBUG ] Distro CentOS codename Final
    [ceph_deploy.install][DEBUG ] Purging host localhost ...
    [ceph_deploy.install][DEBUG ] Purging data from cluster ceph hosts localhost
    [ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
    [ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
    [ceph_deploy.install][DEBUG ] Purging data from host localhost ...
    Loaded plugins: fastestmirror
    Loading mirror speeds from cached hostfile
    Setting up Install Process
    Resolving Dependencies
    --> Running transaction check
    ---> Package ceph.x86_64 0:0.61.8-0.el6 will be installed
    --> Finished Dependency Resolution

    Dependencies Resolved

    =====
    Package            Arch             Version          Repository        Size
    =====
    Installing:
    ceph                x86_64           0.61.8-0.el6    nailgun           13 M

    Transaction Summary
    =====
    Install            1 Package(s)

    Total download size: 13 M
    Installed size: 34 M

```

```

Downloading Packages:
ceph-0.61.8-0.el6.x86_64.rpm | 13 MB 00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : ceph-0.61.8-0.el6.x86_64 1/1
  Verifying : ceph-0.61.8-0.el6.x86_64 1/1

Installed:
  ceph.x86_64 0:0.61.8-0.el6

Complete!
[root@node-4 ~]# time ceph-deploy mon create node-4:192.168.0.2[ceph_deploy.mon][DEBUG ] Deploying
mon, cluster ceph hosts node-4:192.168.0.2
[ceph_deploy.mon][DEBUG ] detecting platform for host node-4 ...
[ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
[ceph_deploy.mon][INFO ] distro info: CentOS 6.4 Final
[node-4][DEBUG ] determining if provided host has same hostname in remote
[node-4][DEBUG ] deploying mon to node-4
[node-4][DEBUG ] remote hostname: node-4
[node-4][INFO ] write cluster configuration to /etc/ceph/{cluster}.conf
[node-4][INFO ] creating path: /var/lib/ceph/mon/ceph-node-4
[node-4][DEBUG ] checking for done path: /var/lib/ceph/mon/ceph-node-4/done
[node-4][DEBUG ] done path does not exist: /var/lib/ceph/mon/ceph-node-4/done
[node-4][INFO ] creating keyring file: /var/lib/ceph/tmp/ceph-node-4.mon.keyring
[node-4][INFO ] create the monitor keyring file
[node-4][INFO ] Running command: ceph-mon --cluster ceph --mkfs -i node-4 --keyring /var/lib/ceph
/tmp/ceph-node-4.mon.keyring
[node-4][INFO ] ceph-mon: renaming mon.noname-a 192.168.0.2:6789/0 to mon.node-4
[node-4][INFO ] ceph-mon: set fsid to a71e4d9f-8b34-4cb9-a31e-8f214942f015
[node-4][INFO ] ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node-4 for mon.node-4
[node-4][INFO ] unlinking keyring file /var/lib/ceph/tmp/ceph-node-4.mon.keyring
[node-4][INFO ] create a done file to avoid re-doing the mon deployment
[node-4][INFO ] create the init path if it does not exist
[node-4][INFO ] locating `service` executable...
[node-4][INFO ] found `service` executable: /sbin/service
Warning: Permanently added 'node-4,10.0.0.130' (RSA) to the list of known hosts.
[node-4][INFO ] Running command: /sbin/service ceph start mon.node-4
[node-4][DEBUG ] === mon.node-4 ===
[node-4][DEBUG ] Starting Ceph mon.node-4 on node-4...
[node-4][DEBUG ] Starting ceph-create-keys on node-4...
^CKilled by signal 2.
Traceback (most recent call last):
  File "/root/ceph-deploy/virtualenv/lib/python2.6/site-packages/pushy-0.5.3-py2.6.egg/pushy/proto
col/baseconnection.py", line 253, in close
    self.__istream.close()
  File "/root/ceph-deploy/virtualenv/lib/python2.6/site-packages/pushy-0.5.3-py2.6.egg/pushy/proto
col/baseconnection.py", line 88, in close
    self.__lock.acquire()
KeyboardInterrupt
Traceback (most recent call last):
  File "/root/ceph-deploy/ceph-deploy", line 8, in <module>
    load_entry_point('ceph-deploy==1.2.4', 'console_scripts', 'ceph-deploy')()
  File "/root/ceph-deploy/ceph_deploy/util/decorators.py", line 83, in newfunc
    return f(*a, **kw)
  File "/root/ceph-deploy/ceph_deploy/cli.py", line 147, in main
    return args.func(args)
  File "/root/ceph-deploy/ceph_deploy/mon.py", line 245, in mon
    mon_create(args)
  File "/root/ceph-deploy/ceph_deploy/mon.py", line 110, in mon_create
    assert False, "distro.sudo_conn.close() finished"
AssertionError: distro.sudo_conn.close() finished

real    0m20.438s
user    0m0.277s
sys     0m0.074s

```

```

[root@node-4 ~]# vi ceph-deploy/ceph_deploy/mon.py
import ConfigParser
import json
import logging
import re
import subprocess
import time

from . import conf
from . import exc
from .cliutil import priority
from .sudo_pushy import get_transport
from .util import paths
from .lib.remoto import process
from . import hosts
from .misc import mon_hosts, remote_shortcode
from .connection import get_connection

LOG = logging.getLogger(__name__)

def mon_status(conn, logger, hostname, silent=False):
    """
    run ``ceph daemon mon.`hostname` mon_status`` on the remote end and provide
    not only the output, but be able to return a boolean status of what is
    going on.
    ``False`` represents a monitor that is not doing OK even if it is up and
    running, while ``True`` would mean the monitor is up and running correctly.
    rconn = get_connection(hostname, logger=logger)

    try:
        out, err, code = process.check(
            rconn,
            ['ceph', 'daemon', mon, 'mon_status'],
            exit=True
        )

        for line in err:
            logger.error(line)

    try:
        mon_info = json.loads(''.join(out))
        return False
    if not silent:
        logger.debug('*'*80)
        logger.debug('status for monitor: %s' % mon)
        for line in out:
            logger.debug(line)
            logger.debug('*'*80)
    if mon_info['rank'] >= 0:
        logger.info('monitor: %s is running' % mon)
        return True
    logger.info('monitor: %s is not running' % mon)
    return False
except RuntimeError:
    logger.info('monitor: %s is not running' % mon)
    return False

def mon_create(args):

    cfg = conf.load(args)
    if not args.mon:
        try:
            mon_initial_members = cfg.get('global', 'mon_initial_members')
        except (ConfigParser.NoSectionError,
                ConfigParser.NoOptionError):
            pass
        else:

```

```

        args.mon = re.split(r'[,\\s]+', mon_initial_members)

if not args.mon:
    raise exc.NeedHostError()

try:
    with file('{cluster}.mon.keyring'.format(cluster=args.cluster),
              'rb') as f:
        monitor_keyring = f.read()
except IOError:
    raise RuntimeError('mon keyring not found; run \'new\' to create a new cluster')

LOG.debug(
    'Deploying mon, cluster %s hosts %s',
    args.cluster,
    ' '.join(args.mon),
)

errors = 0
for (name, host) in mon_hosts(args.mon):
    try:
        # TODO username
        # TODO add_bootstrap_peer_hint
        LOG.debug('detecting platform for host %s ...', name)
        distro = hosts.get(host)
        LOG.info('distro info: %s %s %s', distro.name, distro.release, distro.codename)
        rlogger = logging.getLogger(name)

        # ensure remote hostname is good to go
        hostname_is_compatible(distro.sudo_conn, rlogger, name)
        rlogger.debug('deploying mon to %s', name)
        distro.mon.create(distro, rlogger, args, monitor_keyring)

        # tell me the status of the deployed mon
        time.sleep(2) # give some room to start
        assert False, "distro.sudo_conn.close() pre"
        distro.sudo_conn.close()
        mon_status(None, rlogger, name)

    except RuntimeError as e:
        LOG.error(e)
        errors += 1

if errors:
    raise exc.GenericError('Failed to create %d monitors' % errors)
"ceph-deploy/ceph_deploy/mon.py" 307L, 8869C written
[root@node-4 ~]# ps axu | grep ceph | awk '{system("kill "$2)}' && ceph-deploy purge localhost &
& ceph-deploy purgedata localhost && yum install -y ceph
sh: line 0: kill: (14655) - No such process
[ceph_deploy.install][DEBUG ] Purging from cluster ceph hosts localhost
[ceph_deploy.install][DEBUG ] Detecting platform for host localhost ...
[ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
[ceph_deploy.install][DEBUG ] Distro CentOS codename Final
[ceph_deploy.install][DEBUG ] Purging host localhost ...
[ceph_deploy.install][DEBUG ] Purging data from cluster ceph hosts localhost
[ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
[ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
[ceph_deploy.install][DEBUG ] Purging data from host localhost ...
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package ceph.x86_64 0:0.61.8-0.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

```

| Package | Arch | Version | Repository | Size |
|-------------|--------|--------------|------------|------|
| Installing: | | | | |
| ceph | x86_64 | 0.61.8-0.el6 | nailgun | 13 M |

Transaction Summary

Install 1 Package(s)

Total download size: 13 M

Installed size: 34 M

Downloading Packages:

| | | |
|------------------------------|-------|-------|
| ceph-0.61.8-0.el6.x86_64.rpm | 13 MB | 00:00 |
|------------------------------|-------|-------|

Running rpm_check_debug

Running Transaction Test

Transaction Test Succeeded

Running Transaction

| | |
|---------------------------------------|-----|
| Installing : ceph-0.61.8-0.el6.x86_64 | 1/1 |
| Verifying : ceph-0.61.8-0.el6.x86_64 | 1/1 |

Installed:

ceph.x86_64 0:0.61.8-0.el6

Complete!

```
[root@node-4 ~]# time ceph-deploy mon create node-4:192.168.0.2[ceph_deploy.mon][DEBUG ] Deploying
mon, cluster ceph hosts node-4:192.168.0.2
[ceph_deploy.mon][DEBUG ] detecting platform for host node-4 ...
[ceph_deploy.sudo_pushy][DEBUG ] will use a remote connection without sudo
[ceph_deploy.mon][INFO ] distro info: CentOS 6.4 Final
[node-4][DEBUG ] determining if provided host has same hostname in remote
[node-4][DEBUG ] deploying mon to node-4
[node-4][DEBUG ] remote hostname: node-4
[node-4][INFO ] write cluster configuration to /etc/ceph/{cluster}.conf
[node-4][INFO ] creating path: /var/lib/ceph/mon/ceph-node-4
[node-4][DEBUG ] checking for done path: /var/lib/ceph/mon/ceph-node-4/done
[node-4][DEBUG ] done path does not exist: /var/lib/ceph/mon/ceph-node-4/done
[node-4][INFO ] creating keyring file: /var/lib/ceph/tmp/ceph-node-4.mon.keyring
[node-4][INFO ] create the monitor keyring file
[node-4][INFO ] Running command: ceph-mon --cluster ceph --mkfs -i node-4 --keyring /var/lib/ceph
/tmp/ceph-node-4.mon.keyring
[node-4][INFO ] ceph-mon: renaming mon.noname-a 192.168.0.2:6789/0 to mon.node-4
[node-4][INFO ] ceph-mon: set fsid to a71e4d9f-8b34-4cb9-a31e-8f214942f015
[node-4][INFO ] ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node-4 for mon.node-4
[node-4][INFO ] unlinking keyring file /var/lib/ceph/tmp/ceph-node-4.mon.keyring
[node-4][INFO ] create a done file to avoid re-doing the mon deployment
[node-4][INFO ] create the init path if it does not exist
[node-4][INFO ] locating `service` executable...
[node-4][INFO ] found `service` executable: /sbin/service
Warning: Permanently added 'node-4,10.0.0.130' (RSA) to the list of known hosts.
[node-4][INFO ] Running command: /sbin/service ceph start mon.node-4
[node-4][DEBUG ] === mon.node-4 ===
[node-4][DEBUG ] Starting Ceph mon.node-4 on node-4...
[node-4][DEBUG ] Starting ceph-create-keys on node-4...
Traceback (most recent call last):
  File "/root/ceph-deploy/ceph-deploy", line 8, in <module>
    load_entry_point('ceph-deploy==1.2.4', 'console_scripts', 'ceph-deploy')()
  File "/root/ceph-deploy/ceph_deploy/util/decorators.py", line 83, in newfunc
    return f(*a, **kw)
  File "/root/ceph-deploy/ceph_deploy/cli.py", line 147, in main
    return args.func(args)
  File "/root/ceph-deploy/ceph_deploy/mon.py", line 245, in mon
    mon_create(args)
  File "/root/ceph-deploy/ceph_deploy/mon.py", line 109, in mon_create
    assert False, "distro.sudo_conn.close() pre"
AssertionError: distro.sudo_conn.close() pre
```

| | |
|------|----------|
| real | 0m4.772s |
| user | 0m0.263s |
| sys | 0m0.059s |

History

#1 - 09/18/2013 09:12 AM - Alfredo Deza

- *Status changed from 12 to Resolved*

We moved the conn.close() call as early as possible in all `mon create` calls.

Pull request: <https://github.com/ceph/ceph-deploy/pull/83>

Released in tag v1.2.5