

FreeNAS - Bug #30405

Fix traceback in S3 Service

03/20/2018 06:14 PM - Erin Clark

Status: Done	
Priority: Expected	
Assignee: Vladimir Vinogradenko	
Category: Middleware	
Target version: 11.2-BETA1	
Seen in: Master - FreeNAS Nightlies	Needs Merging: No
Severity: New	Needs Automation: No
Reason for Closing:	Support Suite Ticket: n/a
Reason for Blocked:	Hardware Configuration:
Needs QA: No	ChangeLog Required: No
Needs Doc: No	
Description I am seeing the traceback in the attached screenshot in the S3 service when I go in there for the first time (perhaps I am missing some data it is expecting)?	

Associated revisions

Revision f5615595 - 03/23/2018 08:09 AM - Vladimir Vinogradenko

fix(middleware): Insert rows in ConfigService if none was found

Ticket: #30405

Revision b21647d5 - 03/23/2018 08:19 AM - Vladimir Vinogradenko

fix(middleware): Insert rows in ConfigService if none was found

Ticket: #30405

Revision d30eb775 - 03/23/2018 10:58 AM - Vladimir Vinogradenko

fix(middleware): Insert rows in ConfigService if none was found

Ticket: #30405

Revision a18fa0ea - 03/26/2018 11:24 AM - Vladimir Vinogradenko

fix(middleware): Insert rows in ConfigService if none was found

Ticket: #30405

History

#1 - 03/20/2018 07:31 PM - Vaibhav Chauhan

- Assignee changed from Vaibhav Chauhan to William Grzybowski

I think this issue is related to the middleware. Reassigning to William for verification.

```
root@freenas-vb:~ # midclt call s3.config
list index out of range
Traceback (most recent call last):
  File "/usr/local/lib/python3.6/site-packages/middlewared/main.py", line 152, in call_method
    result = await self.middleware.call_method(self, message)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/main.py", line 958, in call_method
    return await self._call(message['method'], serviceobj, methodobj, params, app=app)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/main.py", line 915, in _call
    return await methodobj(*args)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/schema.py", line 502, in nf
    return await f(*args, **kwargs)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/service.py", line 197, in config
    'prefix': self._config.datastore_prefix
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/main.py", line 962, in call
    return await self._call(name, serviceobj, methodobj, params, pipes=pipes)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/main.py", line 915, in _call
    return await methodobj(*args)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/schema.py", line 502, in nf
    return await f(*args, **kwargs)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/plugins/datastore.py", line 190, in config
    return await self.query(name, None, options)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/schema.py", line 502, in nf
    return await f(*args, **kwargs)
  File "/usr/local/lib/python3.6/asyncio/coroutines.py", line 109, in __next__
    return self.gen.send(None)
  File "/usr/local/lib/python3.6/site-packages/middlewared/plugins/datastore.py", line 176, in query
    return result[0]
IndexError: list index out of range
```

#2 - 03/21/2018 05:06 AM - William Grzybowski

- *Category changed from GUI (new) to Middleware*
- *Assignee changed from William Grzybowski to Vladimir Vinogradenko*

Vladimir, what do you think? What is your opinion in making ConfigService create an entry of none has been found?

The entry should be there on database creation but we cant guarantee that everytime.

#3 - 03/21/2018 05:46 AM - Vladimir Vinogradenko

- *Status changed from Not Started to Blocked*
- *Reason for Blocked set to Waiting for feedback*

Vladimir, what do you think? What is your opinion in making ConfigService create an entry of none has been found?

Negative.

The entry should be there on database creation but we cant guarantee that everytime.

Why can't we guarantee it? This entry is part of database schema.

This is bad: https://github.com/freenas/freenas/blob/master/gui/services/migrations/0002_add_s3_service.py#L18

I can't imagine why an Exception can occur there. Do you have any idea?

#4 - 03/21/2018 05:56 AM - William Grzybowski

Vladimir Vinogradenko wrote:

Vladimir, what do you think? What is your opinion in making ConfigService create an entry of none has been found?

Negative.

The entry should be there on database creation but we cant guarantee that everytime.

Why can't we guarantee it? This entry is part of database schema.

Its part of making the system somewhat self-recoverable. If for any reason an entry in the database vanishes the system will stop working? sqlite3 is not known to be the most resilient database out there.

This is bad: https://github.com/freenas/freenas/blob/master/gui/services/migrations/0002_add_s3_service.py#L18

I can't imagine why an Exception can occur there. Do you have any idea?

Thats not S3 table, thats for Services table.

#5 - 03/21/2018 06:17 AM - Vladimir Vinogradenko

Its part of making the system somewhat self-recoverable. If for any reason an entry in the database vanishes the system will stop working? sqlite3 is not known to be the most resilient database out there.

Did you see any real cases when this happened? One row vanished and the rest of the database stayed alive.

Having rows being auto-created can lead to obscure bugs. By the way, this is one of them:

<https://github.com/freenas/freenas/blob/master/gui/services/views.py#L345>

#6 - 03/21/2018 06:24 AM - William Grzybowski

Vladimir Vinogradenko wrote:

Its part of making the system somewhat self-recoverable. If for any reason an entry in the database vanishes the system will stop working? sqlite3 is not known to be the most resilient database out there.

Did you see any real cases when this happened? One row vanished and the rest of the database stayed alive.

Having rows being auto-created can lead to obscure bugs. By the way, this is one of them:

<https://github.com/freenas/freenas/blob/master/gui/services/views.py#L345>

Yes, we have seen a few cases like these in the past, reason we did the above in some cases. We figured it would be best to try to recover than fill our bug tracking with errors like these.
These could have been mitigated by now with newer version of sqlite3 though. It does not help we are accessing a file database from multiple threads and processes.

#7 - 03/21/2018 06:25 AM - Vladimir Vinogradenko

Okay, you've convinced me.

#8 - 03/22/2018 08:20 AM - Dru Lavigne

- Status changed from *Blocked* to *Not Started*
- Reason for *Blocked* deleted (*Waiting for feedback*)

#9 - 03/27/2018 12:25 AM - Vladimir Vinogradenko

- Status changed from *Not Started* to *In Progress*

#10 - 03/27/2018 12:26 AM - Vladimir Vinogradenko

- Status changed from *In Progress* to *Done*

#11 - 03/27/2018 03:46 AM - Dru Lavigne

- Subject changed from *Traceback in S3 Service* to *Fix traceback in S3 Service*
- Needs Doc changed from *Yes* to *No*
- Needs Merging changed from *Yes* to *No*

Master PR: <https://github.com/freenas/freenas/pull/1012>

#12 - 05/02/2018 05:42 AM - Dru Lavigne

- Status changed from *Done* to *Ready for Testing*

#13 - 06/21/2018 10:07 AM - Joe Maloney

- Status changed from *Ready for Testing* to *Passed Testing*
- Severity set to *New*

#14 - 06/21/2018 10:18 AM - Dru Lavigne

- Status changed from *Passed Testing* to *Done*
- Needs QA changed from *Yes* to *No*

Files

screenshot-20180320-181203.jpg	44.2 KB	03/21/2018	Erin Clark
--------------------------------	---------	------------	------------