

```
diff --git a/cinder/context.py b/cinder/context.py
index b43fc9c..72a1915 100644
```

a/cinder/context.py

```
:
14 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
15 #...License for the specific language governing
   permissions and limitations
16 #...under the License.
17 # Copyright (c) Glenn M. Gobeili
18
19 """RequestContext: context for requests that persist
   through all of cinder."""
```

```
diff --git a/cinder/tests/test_nfs.py b/cinder/tests/test_nfs.py
index 4f346fe..a5edd68 100644
```

a/cinder/tests/test_nfs.py

```
:
12 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
13 #...License for the specific language governing
   permissions and limitations
14 #...under the License.
15 # Copyright (c) Glenn M. Gobeili
16 """Unit tests for the NFS driver module."""
17
18 import errno
19
20
101 def test_set_rw_permissions_for_all_without_nfs_secure_files(self):
102     (mox, drv) = self._mox, self._driver
103
104     self.configuration.nfs_secure_files = False
105
106     mox.StubOutWithMock(drv, '_execute')
107     drv._execute('chmod', 'ugo+rw', '/path',
108                 run_as_root=False)
109
110
111 def test_set_rw_permissions_for_all_with_nfs_secure_files(self):
112     (mox, drv) = self._mox, self._driver
113
114     self.configuration.nfs_secure_files = True
115
116     mox.StubOutWithMock(drv, '_execute')
117     drv._execute('chmod', '660', '/path',
118                 run_as_root=False)
119
120
121
402 mox = self._mox
403 drv = self._driver
404 self.configuration.nfs_shares_config = self.
   TEST_SHARES_CONFIG_FILE
405 self.configuration.nfs_secure_files =
   'Autodetect'
406 mox.StubOutWithMock(os.path, 'exists')
407 os.path.exists(self.TEST_SHARES_CONFIG_FILE).
   AndReturn(True)
408 mox.StubOutWithMock(drv, '_execute')
```

```
diff --git a/cinder/tests/test_volume.py b/cinder/tests/test_volume.py
index 59bd234..b883849 100644
```

a/cinder/tests/test_volume.py

```
:
13 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
14 #...License for the specific language governing
   permissions and limitations
15 #...under the License.
16
17 # Copyright (c) Glenn M. Gobeili
18
19 Tests for Volume Code.
```

```
diff --git a/cinder/volume/drivers/nfs.py b/cinder/volume/drivers/nfs.py
index 2dd9d3e..d5bc1d2 100644
```

a/cinder/volume/drivers/nfs.py

```
:
12 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
13 #...License for the specific language governing
   permissions and limitations
14 #...under the License.
```

b/cinder/context.py

```
:
14 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
15 #...License for the specific language governing
   permissions and limitations
16 #...under the License.
17
18 """RequestContext: context for requests that persist
   through all of cinder."""
```

b/cinder/tests/test_nfs.py

```
:
12 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
13 #...License for the specific language governing
   permissions and limitations
14 #...under the License.
15
16 """Unit tests for the NFS driver module."""
17
18 import errno
19
20
100 def test_set_rw_permissions_for_all_without_nfs_secure_files(self):
101     (mox, drv) = self._mox, self._driver
102
103     self.configuration.nfs_secure_files = 'false'
104
105     mox.StubOutWithMock(drv, '_execute')
106     drv._execute('chmod', 'ugo+rw', '/path',
107                 run_as_root=False)
108
109
110 def test_set_rw_permissions_for_all_with_nfs_secure_files(self):
111     (mox, drv) = self._mox, self._driver
112
113     self.configuration.nfs_secure_files = 'true'
114
115     mox.StubOutWithMock(drv, '_execute')
116     drv._execute('chmod', '660', '/path',
117                 run_as_root=False)
118
119
120
401 mox = self._mox
402 drv = self._driver
403 self.configuration.nfs_shares_config = self.
   TEST_SHARES_CONFIG_FILE
404 self.configuration.nfs_secure_files =
   'autodetect'
405 mox.StubOutWithMock(os.path, 'exists')
406 os.path.exists(self.TEST_SHARES_CONFIG_FILE).
   AndReturn(True)
407 mox.StubOutWithMock(drv, '_execute')
```

b/cinder/tests/test_volume.py

```
:
13 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
14 #...License for the specific language governing
   permissions and limitations
15 #...under the License.
16
17 Tests for Volume Code.
```

b/cinder/volume/drivers/nfs.py

```
:
12 #...WARRANTIES OR CONDITIONS OF ANY KIND, either
   express or implied. See the
13 #...License for the specific language governing
   permissions and limitations
14 #...under the License.
```

```

15 #.Copyright.(c).Glenn.M.Gobeli
16
17 import errno
18 import os
19
203
204 .....:parm.path:the volume file path.
205 .....""
206 .....if self.configuration.nfs_secure_files== 'True':
207 .....permissions= '660'
208 .....LOG.debug('Filepath %s is being set with
209 permissions: %s' %
210 .....(path, permissions))
211 .....else:
212 .....permissions= 'ugo+rw'
213 .....LOG.warn('Filepath %s is being set with open
214 permissions: %s' %
215 .....(path, permissions))
216
217
218 self._execute('chmod', permissions, path,
219 run_as_root=False)
220
221
222 .....LOG.error(msg)
223 .....raise exception.NfsException(msg)
224
225 .....if self.configuration.nfs_secure_files.lower()
226 == 'autodetect':
227 .....self.configuration.nfs_secure_files=
228 'Autodetect'
229 .....elif self.configuration.nfs_secure_files.lower()
230 == 'true':
231 .....self.configuration.nfs_secure_files= 'True'
232 .....elif self.configuration.nfs_secure_files.lower()
233 == 'false':
234 .....self.configuration.nfs_secure_files= 'False'
235 .....else:
236 .....msg= ("NFS config 'nfs_secure_files=%s'
237 invalid..Must be ""
238 ..... "Autodetect, True, or False") % \
239 .....self.configuration.nfs_secure_files
240 .....LOG.error(msg)
241 .....raise exception.NfsException(msg)
242
243
244 def set_nfs_security_indication(self,
245 is_new_cinder_install):
246 .....""Determine the setting to use for
247 nfs_secure_files.
248
249 .....Value of nfs_secure_files is updated if it is
250 currently 'Autodetect'
251
252 .....to either True or False based upon if this is a
253 new Cinder
254 .....installation.
255
256 .....:parm.is_new_cinder_install: bool indication of
257 new Cinder install
258
259
260
261 self._ensure_shares_mounted()
262
263
264 .....if self.configuration.nfs_secure_files==
265 'Autodetect':
266 .....file_name= '.nfsSecureFileDetector'
267 .....mount_point= self._get_mount_point_for_
268 share(
269 .....self._mounted_shares[0])
270 .....file_path= '/%s/%s' % (mount_point,
271 file_name)
272 .....if os.path.exists(file_path):
273 .....self.configuration.nfs_secure_files=
274 'True'
275 .....LOG.debug('New Cinder installation
276 detector file found.')
277 .....else:
278 .....if is_new_cinder_install:
279 .....self.configuration.nfs_secure_files
280 = 'True'

```

```

15
16 import errno
17 import os
18
203
204 .....:parm.path:the volume file path.
205 .....""
206 .....if self.configuration.nfs_secure_files==
207 'true':
208 .....permissions= '660'
209 .....LOG.debug('Filepath %s is being set with
210 permissions: %s' %
211 .....(path, permissions))
212 .....else:
213 .....permissions= 'ugo+rw'
214 .....err_parms= {'path': path, 'perm':
215 permissions}
216 .....LOG.warn(_('%(path)s is being set with open
217 permissions: '
218 .....'%(perm)s') % err_parms)
219
220 self._execute('chmod', permissions, path,
221 run_as_root=False)
222
223
224 .....LOG.error(msg)
225 .....raise exception.NfsException(msg)
226
227 .....if (self.configuration.nfs_secure_files.lower()
228 == 'autodetect' or
229 .....self.configuration.
230 nfs_secure_files.lower() == 'true' or
231 .....self.configuration.nfs_secure_files.
232 lower() == 'false'):
233 .....self.configuration.nfs_secure_files
234 = \
235 .....self.configuration.
236 nfs_secure_files.lower()
237 .....else:
238 .....msg= ("NFS config 'nfs_secure_files=%s'
239 invalid..Must be ""
240 ..... "autodetect, true, or false") % \
241 .....self.configuration.nfs_secure_files
242 .....LOG.error(msg)
243 .....raise exception.NfsException(msg)
244
245
246 def set_nfs_security_indication(self,
247 is_new_cinder_install):
248 .....""Determine the setting to use for
249 nfs_secure_files.
250
251 .....Value of nfs_secure_files is updated if it is
252 currently 'autodetect'
253
254 .....to either true or false based upon if this is a
255 new Cinder
256 .....installation.
257
258 .....:parm.is_new_cinder_install: bool indication of
259 new Cinder install
260
261
262
263 self._ensure_shares_mounted()
264
265
266 .....if self.configuration.nfs_secure_files==
267 'autodetect':
268 .....file_name= '.nfsSecureFileDetector'
269 .....mount_point= self._get_mount_point_for_
270 share(
271 .....self._mounted_shares[0])
272 .....file_path= '/%s/%s' % (mount_point,
273 file_name)
274 .....if os.path.exists(file_path):
275 .....self.configuration.nfs_secure_files=
276 'true'
277 .....LOG.debug('New Cinder installation
278 detector file found.')
279 .....else:
280 .....if is_new_cinder_install:
281 .....self.configuration.nfs_secure_files
282 = 'true'

```

8/21/2014	HTML Diff
<pre> 535try: 536fh=.open(file_path, 'w') 537 secure_files_usage.\n') : 545LOG.error(_('Failed to created nfs_secure_files.' 546 format(err)) 547else: 548 =. 'False' 549 550LOG.debug('NFS variable nfs_secure_files setting is: %s' .%) 551self.configuration.nfs_secure_files) 552 553if self.configuration.nfs_secure_files ==. 554doc_html =. "http://docs.openstack.org/admin- guide-cloud/content".\ 555"/nfs_backend.html" 556LOG.warn(_("The NFS file permissions mode will be 666 (allowing." diff --git a/cinder/volume/manager.py b/cinder/volume/manager.py index 178c774..3e6345c 100644 a/cinder/volume/manager.py : 13 #...WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the 14 #...License for the specific language governing permissions and limitations 15 #...under the License. 16 # 17 #Copyright (c) Glenn M. Gobeil 18 """ 19 Volume manager manages creating, attaching, detaching, and persistent storage. 20 </pre>	<pre> 534try: 535fh=.open(file_path, 'w') 536 secure_files_usage.\n') : 544LOG.error(_('Failed to created nfs_secure_files.' 545 % format(err)) 546else: 547 =. 'false' 548 549LOG.debug('NFS variable nfs_secure_files setting is: %s' .%) 550self.configuration.nfs_secure_files) 551 552if self.configuration.nfs_secure_files ==. 553doc_html =. "http://docs.openstack.org/admin- guide-cloud/content".\ 554"/nfs_backend.html" 555LOG.warn(_("The NFS file permissions mode will be 666 (allowing." diff --git b/cinder/volume/manager.py : 13 #...WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the 14 #...License for the specific language governing permissions and limitations 15 #...under the License. 16 """ 17 Volume manager manages creating, attaching, detaching, and persistent storage. 18 </pre>

Modified at 21.08.2014. HTML formatting created by [diff2html](http://diff2html.com/).