

openstack / keystone Public

<> Code Pull requests 1 Actions Security and quality Insights

⚠ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Commit 8735009



vishvananda authored and yolandacanonical committed on Nov 22, 2012

Ensures User is member of tenant in ec2 validation

It is possible that a user is no longer a member of a tenant when they attempt to use an ec2 token. This checks to make sure that the user still has at least one valid role in the tenant before authenticating them. This should automatically work for the s3 version as well since it is a subclass.

Fixes bug 1064914

Change-Id: Ieb237bae936a7b00ce7ba4d4c59aec6c7a69ec21

· essex-eol

1 parent [ddb4019](#) commit 8735009

1 file changed

+13 -10

↑ Top

Filter files...

keystone/contrib/ec2

core.py

Search within code

keystone/contrib/ec2/core.py

```
@@ -161,9 +161,19 @@ def authenticate(self, context, credentials=None,
```

```

161 161         context=context,
162 162         user_id=creds_ref['user_id'])
163 163     metadata_ref = self.identity_api.get_metadata(
164 164         -         context=context,
165 165         -         user_id=user_ref['id'],
166 166         -         tenant_id=tenant_ref['id'])
167 167
168 168     +         context=context,
169 169     +         user_id=user_ref['id'],
170 170     +         tenant_id=tenant_ref['id'])
171 171
172 172     +     # TODO(termie): optimize this call at some point and put it into the
173 173     +     #         the return for metadata
174 174     +     # fill out the roles in the metadata
175 175     +     roles = metadata_ref.get('roles', [])
176 176     +     if not roles:
177 177     +         raise exception.Unauthorized(message='User not valid for tenant.')
178 178     +     roles_ref = [self.identity_api.get_role(context, role_id)
179 179     +                 for role_id in roles]
180 180
181 181     catalog_ref = self.catalog_api.get_catalog(
182 182         context=context,
183 183         user_id=user_ref['id'],
184 184
185 185     @@ -176,13 +186,6 @@ def authenticate(self, context, credentials=None,
186 186         tenant=tenant_ref,
187 187         metadata=metadata_ref))
188 188
189 189     -     # TODO(termie): optimize this call at some point and put it into the
190 190     -     #         the return for metadata
191 191     -     # fill out the roles in the metadata
192 192     -     roles_ref = []
193 193     -     for role_id in metadata_ref.get('roles', []):
194 194     -         roles_ref.append(self.identity_api.get_role(context, role_id))
195 195     -
196 196     # TODO(termie): make this a util function or something
197 197     # TODO(termie): i don't think the ec2 middleware currently expects a
198 198     #         full return, but it contains a note saying that it

```

Comments 0



Please [sign in](#) to comment.