

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 lines changed

↑ Top ⚙

Filter files...

keystone/contrib/ec2

core.py

1 file changed +13 -10 lines changed

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

```

## Comments 0



Please [sign in](#) to comment.