



RBAC in Neutron and Tungsten Fabric

Tungsten Fabric Users Group



Who we are

Intro

- we work at CodiLime
 - services, consulting, development, teams
 - SDN/NFV, cloud-native, DevOps
- contacts:
 - Maciek Jagiello maciej.jagiello@codilime.com
 - Jarek Lukow jaroslaw.lukow@codilime.com

What you'll find in this talk

Intro

- CodiLime's R&D work
- Neutron RBAC feature
- use-cases
- Tungsten Fabric implementation
- Neutron plugin API

Our roadmap

RBAC in Neutron and TF

1. **the use-case**
2. Neutron RBAC
3. Tungsten Fabric RBAC
4. Neutron plugin API
5. the integration

codilime[®]
CREATING VALUE

tf tungstenfabric

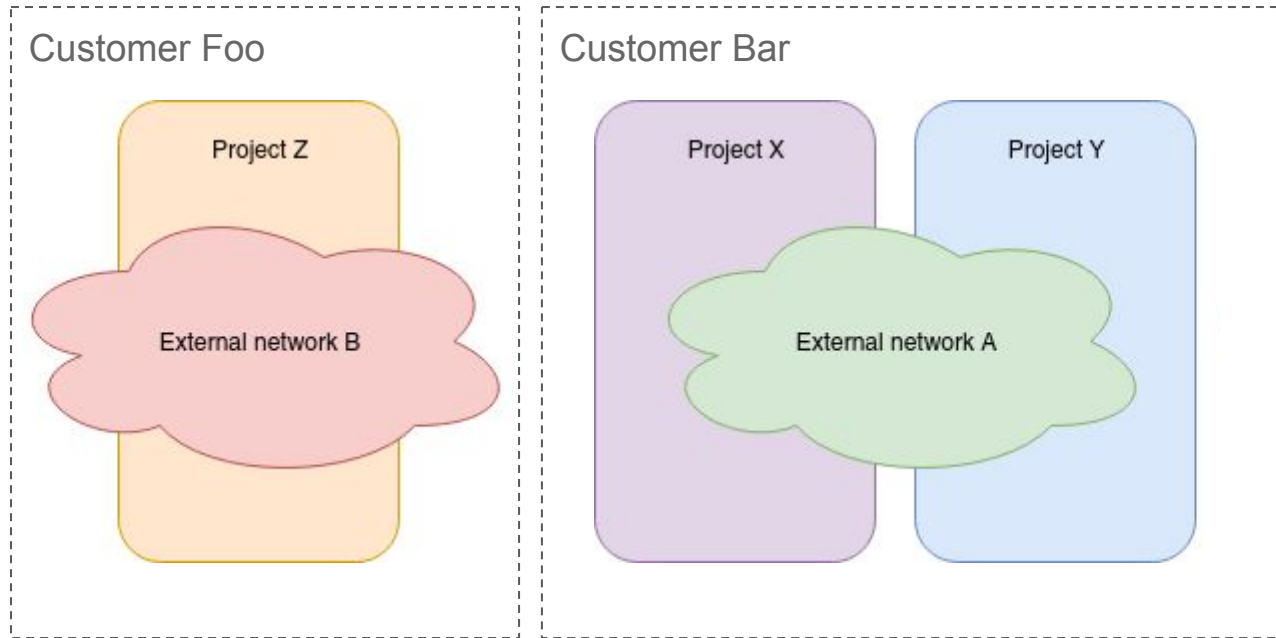
The use-case

Motivation

- multi-tenant public cloud provider service
- floating IP pool assigned to a customer
- customer uses multiple tenants
- additional considerations:
 - operations
 - self-service

The use-case

Motivation



Our roadmap

RBAC in Neutron and TF

1. the use-case
2. **Neutron RBAC**
3. Tungsten Fabric RBAC
4. Neutron plugin API
5. the integration

codilime[®]
CREATING VALUE

tf tungstenfabric

Neutron access modes

RBAC in Neutron

- 'shared' network attribute
- accessible in all projects or single project
- but... there is RBAC

Neutron RBAC: model

RBAC in Neutron

Field	Value
id	afdd5b8d-b6f5-4a15-9817-5231434057be
name	None
project_id	61b7eba037fd41f29cfba757c010faff
target_project_id	target project UUID
action	access_as_external
object_type	network
object_id	network UUID

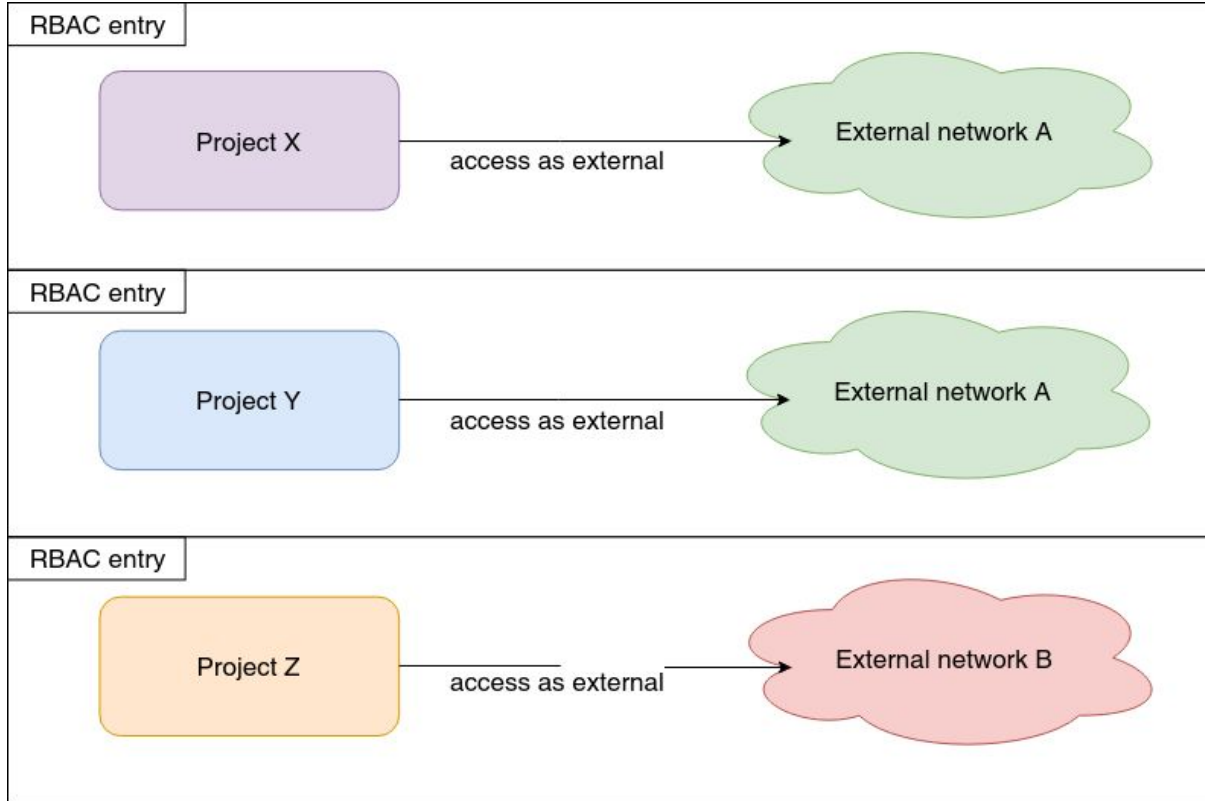
Neutron RBAC: model

RBAC in Neutron

Field	Value
target_project_id	target project UUID
action	access_as_external
object_id	network UUID

Neutron RBAC: model

RBAC in Neutron



Neutron RBAC: configuration

RBAC in Neutron

```
$ openstack network rbac create \  
  --target-project [target_project_id] \  
  --action access_as_external \  
  --type network [network_id]
```

```
+-----+-----+  
| Field          | Value                                     |  
+-----+-----+  
| id             | afdd5b8d-b6f5-4a15-9817-5231434057be  |  
| name          | None                                     |  
| project_id     | 61b7eba037fd41f29cfba757c010faff      |  
| target_project_id | [target_project_id]                   |  
| action         | access_as_external                     |  
| object_type    | network                                 |  
| object_id      | [network_id]                           |  
+-----+-----+
```

Our roadmap

RBAC in Neutron and TF

1. the use-case
2. Neutron RBAC
3. **Tungsten Fabric RBAC**
4. Neutron plugin API
5. the integration

codilime[®]
CREATING VALUE

tf tungstenfabric

Tungsten Fabric RBAC

RBAC in TF

- there is also an RBAC concept in TF
- it has to be enabled globally
- entirely changes the way of dealing with objects and authentication

Tungsten Fabric RBAC: model

RBAC in TF

- each object has an ACL
 - R - read
 - W - create, update
 - X - link, reference

Tungsten Fabric RBAC: configuration

RBAC in TF

/etc/contrail/contrail-api.conf

```
aaa-mode = rbac
```


Tungsten Fabric RBAC: configuration

RBAC in TF

The screenshot displays the 'Edit' dialog for a share in Tungsten Fabric. The breadcrumb navigation shows 'networks > default-domain > admin'. The dialog has three tabs: 'Network', 'Tags', and 'Permissions', with 'Permissions' selected. The 'Owner' field contains the ID '28835727-0978-463a-aed5-338c28694ac5'. Under 'Owner Permissions', there are three buttons: 'Read', 'Write', and 'Refer'. Under 'Global Share Permissions', there are two buttons: 'Read' and 'Refer'. A 'Share List' section is expanded, showing a table with columns 'Project' and 'Permissions'. The table contains one entry: 'demo (3e86269d-8e70-4bdf-9913-ε)' with 'Read' and 'Refer' permissions. At the bottom right, there are 'Cancel' and 'Save' buttons.

networks > default-domain > admin

Edit

Network Tags Permissions

Owner

28835727-0978-463a-aed5-338c28694ac5

Owner Permissions

Read Write Refer

Global Share Permissions

Read Refer

Share List

Project	Permissions
demo (3e86269d-8e70-4bdf-9913-ε)	Read Refer

Cancel Save

Our roadmap

RBAC in Neutron and TF

1. the use-case
2. Neutron RBAC
3. Tungsten Fabric RBAC
- 4. Neutron plugin API**
5. the integration

codilime[®]
CREATING VALUE

tf tungstenfabric

Plugin API background

Neutron plugin API

- core plugin being deprecated in favor of ML2 and mechanism drivers
- well-defined and restricted API
- can connect multiple SDN backends to single Neutron

Neutron plugin configuration

Neutron plugin API

`/etc/neutron/neutron.conf`

```
core_plugin = ml2
service_plugins = opencontrail-router
```

`/etc/neutron/plugins/ml2/ml2_conf.ini`

```
[ml2]
mechanism_drivers = opencontrail
```

Our roadmap

RBAC in Neutron and TF

1. the use-case
2. Neutron RBAC
3. Tungsten Fabric RBAC
4. Neutron plugin API
5. **the integration**

codilime[®]
CREATING VALUE

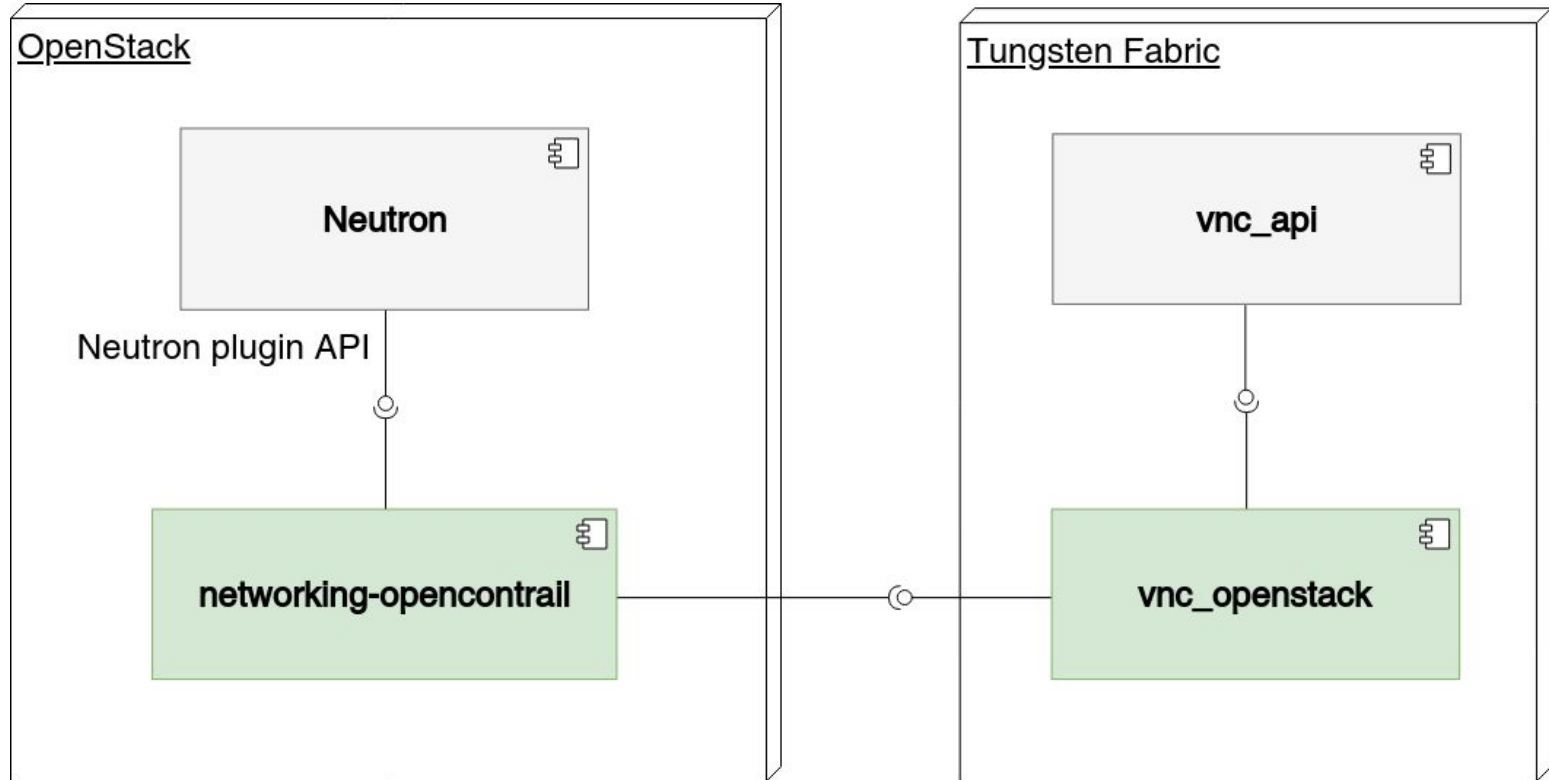
tf tungstenfabric

Example integration plan

- service plugin callbacks
- data flowing to the `vnc_openstack` module without major modifications
- translation being done in `vnc_openstack`
- native calls to `vnc_api`

The integration

Neutron to TF



Wrap-up

Outro

- Neutron has RBAC
- Tungsten has RBAC too
- how to implement integrations using the ML2/service plugin interface

Q&A