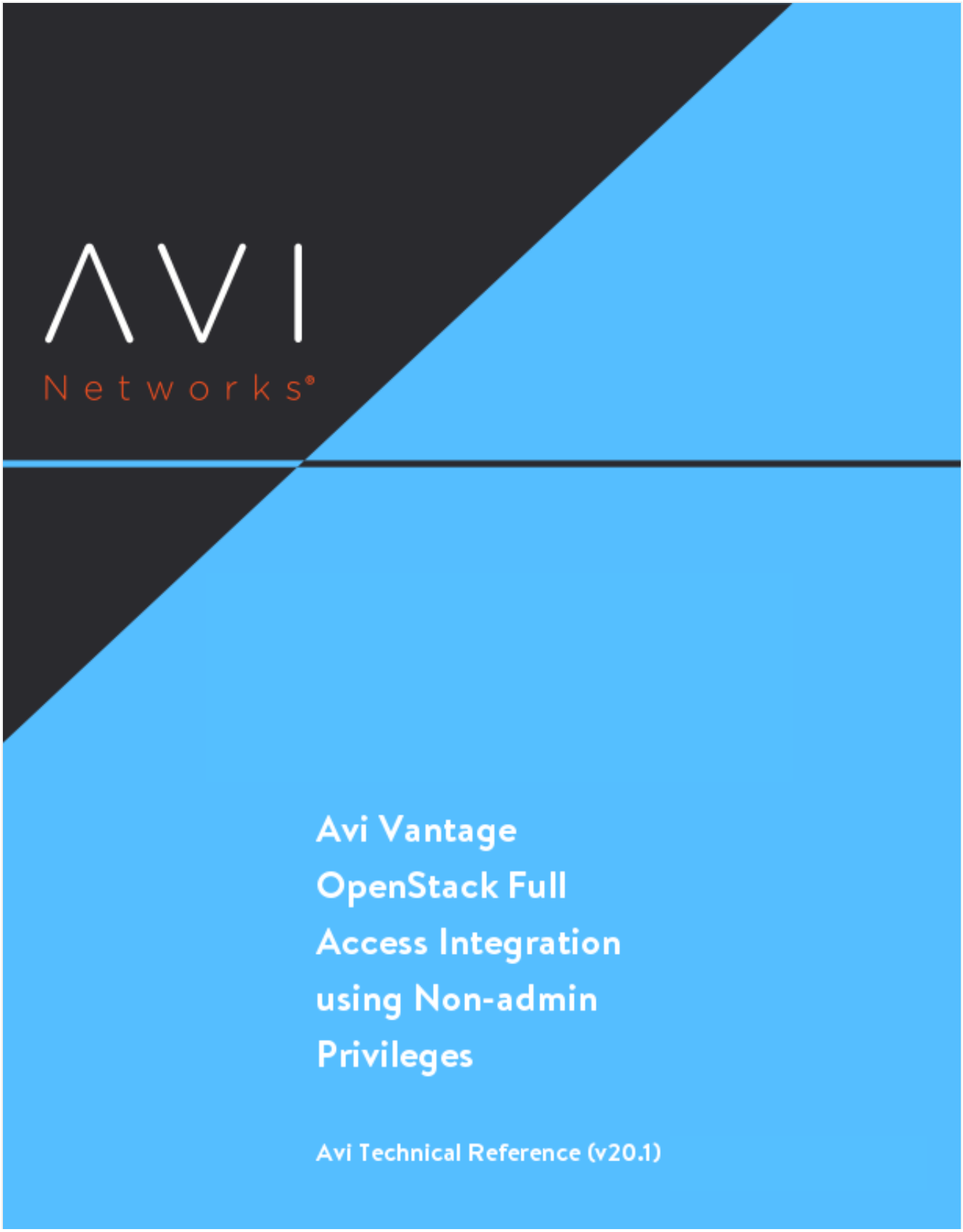

Avi Vantage OpenStack Full Access Integration using Non-admin Privileges

Avi Technical Reference (v20.1)

Avi Vantage OpenStack Full Access Integration using Non-admin Privileges

view online

Overview

This guide explains the integration process of OpenStack full access using non-admin privileges.

You require admin privileges to deploy Avi Solution in Provider mode on OpenStack. Admin privileges provides system-wide access to OpenStack resources like Tenants (Projects), Users, Roles, Networks, Routers, Ports etc., thereby enabling the Avi - OpenStack integration to work seamlessly. The cloud admin or the operator does not have to do any additional configuration in OpenStack.

Note: This is the recommended way to integrate Avi Vantage with OpenStack.

However, for some deployments, admin privileges will be disabled as per business or IT policies. In such cases, you will not be able to deploy and use Avi Vantage with Provider mode. This guide outlines the OpenStack services policy changes, and constraints on OpenStack network resources in order to deploy and use Avi Vantage in Provider mode without giving it admin privileges.

Creating Admin User without Admin Privileges

For OpenStack to work without admin privileges, OpenStack admin needs to create Avi admin user, an Avi admin role, and a tenant in OpenStack to be used in Avi cloud configuration. Existing tenant can also be used.

You can use admin user configured in Avi OpenStack cloud (called Avi Admin user) for establishing initial connection, authentication with keystone, and accessing projects, deployed regions and networks for initial setup. Once initial setup is complete, you can use this to import tenants from OpenStack, and access resources in various tenants. Ensure that Avi admin user has enough privileges to read the Tenants (Projects), Users, Roles, and list the role assignments for users in tenants.

Admin tenant in Avi OpenStack cloud (called Avi Admin tenant) is the tenant in which Avi will spin up the Service Engines in Provider mode. Ensure that the admin user has enough permissions to upload the Service Engine images, list the flavors, and spin up the VMs in networks created or shared in the Avi Admin tenant.

Following are the steps to create Avi admin user in OpenStack:

1. Create aviadmin role in OpenStack.
2. Create user with name aviuser in OpenStack.
3. Create project with name avilbaas in OpenStack.
4. Assign aviuser user, aviadmin role in avilbaas tenant.
5. Add aviuser user, aviadmin role to other projects where Avi Load balancer service is required.
6. For non-default domain, use the desired domain while creating the Avi Admin user and Avi Admin tenant.

Note: The role name should be aviadmin (like admin role in OpenStack) to ensure Avi Controller use these credentials to deploy in Provider mode with SE in provider context. However, the user name and project can be anything. By convention OpenStack has pre-defined role names like admin, member, reader etc. so there is a pre-defined Avi admin role name, aviadmin.

The following are the OpenStack commands for above steps:

1. Run the following command using admin credentials.

```
source admin-openrc.sh
```

2. Create aviuser to be used as Avi admin user in Avi OpenStack Cloud.

```
config
$ openstack user create --password avi123 --description 'Avi Admin user' aviuser
```

3. Create aviadmin role.

```
$ openstack role create aviadmin
```

4. Create avilbaas project to be used as Avi Admin Tenant in Avi OpenStack.

```
Cloud config
$ openstack project create --description 'Avi LBaaS Project' avilbaas
```

5. Assign aviuser aviadmin role in avilbaas project.

```
$ openstack role add --project avilbaas --user aviuser aviadmin
```

6. Add aviadmin role in demo tenant (all other tenants).

```
$ openstack role add --project demo --user aviuser aviadmin
```

7. Verify the roles assigned to aviuser in all tenants.

```
$ openstack role assignment list --user aviuser --names
```

OpenStack Policy Changes

Avi Controller interacts with Keystone, Nova, Neutron, and Glance OpenStack service. Following is a list of policy changes required per service:

Keystone Policy Changes

The following are the steps to interact with Keystone policy change:

1. Identify policy file from keystone.conf file. For instance,

```
[oslo_policy]
   policy_file = keystone.policy.yaml
```

2. Define aviadmin role in keystone policy file (for instance, keystone.policy.yaml)

```
"aviadmin_role": "role:aviadmin"
```

3. Give access to the following APIs for aviadmin_role (for instance, in keystone.policy.yaml)

```
Show domain details
   GET  /v3/domains/{domain_id}
   Intended scope(s): system
   "identity:get_domain": "rule:admin_required or token.project.domain.id:%(target.domain.id)s or rule:aviadmin_role"

   List domains
   GET  /v3/domains
   Intended scope(s): system
   "identity:list_domains": "rule:admin_required or rule:aviadmin_role"



   List endpoints
   GET  /v3/endpoints
   Intended scope(s): system
   "identity:list_endpoints": "rule:admin_required or rule:aviadmin_role"

   List projects
   GET  /v3/projects
   Intended scope(s): system
   "identity:list_projects": "rule:admin_required or rule:aviadmin_role"

   Show role details
   GET  /v3/roles/{role_id}
   HEAD  /v3/roles/{role_id}
   Intended scope(s): system
   "identity:get_role": "rule:admin_required or rule:aviadmin_role"

   List roles
   GET  /v3/roles
   HEAD  /v3/roles
   Intended scope(s): system
   "identity:list_roles": "rule:admin_required or rule:aviadmin_role"

   List role assignments
   GET  /v3/role_assignments
   HEAD  /v3/role_assignments
   Intended scope(s): system
   "identity:list_role_assignments": "rule:admin_required or rule:aviadmin_role"

   List services
   GET  /v3/services
   Intended scope(s): system
```

```
    "identity:list_services": "rule:admin_required or rule:aviadmin_role"

    Show user details
    GET  /v3/users/{user_id}
    HEAD  /v3/users/{user_id}
    "identity:get_user": "rule:admin_or_owner or rule:aviadmin_role"

    List users
    GET  /v3/users
    HEAD  /v3/users
    Intended scope(s): system
    "identity:list_users": "rule:admin_required or rule:aviadmin_role"
```

4. Restart Keystone service using the following command:

```
apache2ctl restart
```

Nova Policy Changes

The following are the steps to interact with Nova policy change:

1. Identify policy file from nova.conf file. For instance,

```
[oslo_policy]
   policy_file = nova.policy.yaml
```

2. Define aviadmin role in nova policy file (for instance, nova.policy.yaml)

```
"aviadmin_role": "role:aviadmin"
```

3. Give access to following APIs for aviadmin_role (for instance, in nova.policy.yaml):

```
List all aggregates
GET  /os-aggregates
"os_compute_api:os-aggregates:index": "rule:admin_api or rule:aviadmin_role"

Show details for an aggregate
GET  /os-aggregates/{aggregate_id}
"os_compute_api:os-aggregates:show": "rule:admin_api or rule:aviadmin_role"

List availability zone information without host information
GET  /os-availability-zone
"os_compute_api:os-availability-zone:list": "rule:admin_or_owner or rule:aviadmin_role"

List detailed availability zone information with host information
GET  /os-availability-zone/detail
```

```
"os_compute_api:os-availability-zone:detail": "rule:admin_api or rule:aviadmin_role"

List available extensions and show information for an extension by
alias
GET  /extensions
GET  /extensions/{alias}
"os_compute_api:extensions": "rule:admin_or_owner or rule:aviadmin_role"

It also allows access to the full list of tenants that have access
to a flavor via an os-flavor-access API.
GET  /flavors/{flavor_id}/os-flavor-access
GET  /flavors/detail
GET  /flavors/{flavor_id}
POST  /flavors
PUT  /flavors/{flavor_id}
"os_compute_api:os-flavor-access": "rule:admin_or_owner or rule:aviadmin_role"

List all servers
GET  /servers
"os_compute_api:servers:index": "rule:admin_or_owner or rule:aviadmin_role"

List all servers with detailed information
GET  /servers/detail
"os_compute_api:servers:detail": "rule:admin_or_owner or rule:aviadmin_role"

List all servers for all projects
GET  /servers
"os_compute_api:servers:index:get_all_tenants": "rule:admin_api or rule:aviadmin_role"

List all servers with detailed information for all projects
GET  /servers/detail
"os_compute_api:servers:detail:get_all_tenants": "rule:admin_api or rule:aviadmin_role"
```

4. Restart nova-api service:

```
service nova-api restart
```

Neutron Policy Changes

The following are the steps to interact with Neutron policy change:

1. Identify policy file from neutron.conf file. For instance,

```
[oslo_policy]
policy_file = neutron.policy.json
```

2. Define aviadmin role in neutron policy file (for instance, neutron.policy.json)

```
"aviadmin_role": "role:aviadmin",
```

3. Give access to following APIs for aviadmin_role (for instance, in neutron.policy.json):

```
"create_port:device_owner": "not rule:network_device or rule:context_is_advsvc or rule:admin_or_network_owner o

"create_port:fixed_ips": "rule:context_is_advsvc or rule:admin_or_network_owner or rule:aviadmin_role",
"create_port:fixed_ips:ip_address": "rule:context_is_advsvc or rule:admin_or_network_owner or rule:aviadmin_rol
"create_port:fixed_ips:subnet_id": "rule:context_is_advsvc or rule:admin_or_network_owner or rule:shared or rul
"create_port:allowed_address_pairs": "rule:admin_or_network_owner or rule:aviadmin_role",
"get_port": "rule:context_is_advsvc or rule:admin_owner_or_network_owner or rule:aviadmin_role",
"update_port:fixed_ips": "rule:context_is_advsvc or rule:admin_or_network_owner or rule:aviadmin_role",
"update_port:fixed_ips:ip_address": "rule:context_is_advsvc or rule:admin_or_network_owner or rule:aviadmin_rol
"update_port:fixed_ips:subnet_id": "rule:context_is_advsvc or rule:admin_or_network_owner or rule:shared or rul
"update_port:allowed_address_pairs": "rule:admin_or_network_owner or rule:aviadmin_role ",
"get_agent": "rule:admin_only or rule:aviadmin_role",
"get_l3-agents": "rule:admin_only or rule:aviadmin_role",
"get_loadbalancer-agent": "rule:admin_only or rule:aviadmin_role",
"get_loadbalancer-pools": "rule:admin_only or rule:aviadmin_role",
"get_agent-loadbalancers": "rule:admin_only or rule:aviadmin_role",
"get_loadbalancer-hosting-agent": "rule:admin_only or rule:aviadmin_role",
"create_floatingip:floating_ip_address": "rule:admin_only or rule:aviadmin_role",
"get_floatingip": "rule:admin_or_owner or rule:aviadmin_role",
```

4. Restart neutron-server service using following command:

```
service neutron-server restart
```

Configuring Cloud

On Avi Controller, configure the cloud with aviuser credentials, and avilbaas as admin tenant. In OpenStack role mapping, map OpenStack aviadmin role to Avi System-Admin or Tenant-Admin role.

Note: You can set map_admin_to_cloud_admin to True. This will map avilbaas tenant to admin tenant in Avi, and any action taken in admin tenant in Avi will reflect in avilbaas tenant.

Configuration Example:

In this example, all OpenStack roles are given Tenant-Admin role.

```
[admin:avi-controller]: show cloud Default-Cloud
+-----------------------------+-----------------------------------------+
| Field                       | Value                                   |
+-----------------------------+-----------------------------------------+
| uuid                        | cloud-4db84437-f236-41cb-996f-11e450976744 |
| name                        | Default-Cloud                           |
| vtype                       | CLOUD_OPENSTACK                         |
```

```
| openstack_configuration      |                                          |
|   username                   | aviuser                                  |
|   password                   | [sensitive]                              |
|   admin_tenant               | avilbaas                                 |
|   mgmt_network_name          | mgmt                                     |
|   privilege                  | WRITE_ACCESS                             |
|   use_keystone_auth          | True                                     |
|   region                     | RegionOne                                |
|   hypervisor                 | KVM                                      |
|   tenant_se                  | False                                    |
|   import_keystone_tenants    | True                                     |
|   anti_affinity              | True                                     |
|   port_security              | False                                    |
|   security_groups            | False                                    |
|   allowed_address_pairs      | True                                     |
|   free_floatingips           | False                                    |
|   img_format                 | OS_IMG_FMT_AUTO                          |
|   use_admin_url              | True                                     |
|   role_mapping[1]            |                                          |
|     os_role                  | *                                        |
|     avi_role                 | Tenant-Admin                             |
|   use_internal_endpoints     | False                                    |
|   config_drive               | True                                     |
|   auth_url                   | http://10.10.32.213:5000/v3              |
|   insecure                   | False                                    |
|   external_networks          | False                                    |
|   neutron_rbac               | True                                     |
|   map_admin_to_cloudadmin    | True                                     |
|   nuage_port                 | 8443                                     |
|   contrail_plugin            | False                                    |
|   name_owner                 | True                                     |
|   use_nuagevip               | False                                    |
|   nuage_virtualip            | False                                    |
|   contrail_disable_policy    | False                                    |
| apic_mode                    | False                                    |
| dhcp_enabled                 | True                                     |
| mtu                          | 1500 bytes                               |
| prefer_static_routes         | False                                    |
| enable_vip_static_routes     | False                                    |
| license_type                 | LIC_CORES                                |
| state_based_dns_registration | True                                     |
| ip6_autocfg_enabled          | True                                     |
| tenant_ref                   | admin                                    |
| license_tier                 | ENTERPRISE_18                            |
| autoscale_polling_interval   | 60 seconds                               |
+------------------------------+------------------------------------------+
```

Requirements for OpenStack Keystone/Nova/Neutron Resources

The following are the requirements for OpenStack resources:

1. Avi Controllers should be in avilbaas tenant for cluster VIP to work. Without admin privileges, Avi Controller will not be able to look into other tenants for controller and it expects the Controllers and SE to be in same tenant (avilbaas tenant).

2. Flavors ? Avi Controller will be able to use only public flavors or the flavors accessible to avilbaas tenant. Share the flavor to avilbaas tenant in order to use the flavor for SE. After sharing the flavor, set the instance_flavor option in SE Group.

3. Add aviadmin role in all tenants ? You need to have an admin role for integration to work.

    a. Avi admin user will not be able to create ports in tenant networks, without having an admin role. This cannot be changed via Neutron policy. To fix this issue, the CC agent will use the tenant scoped client to create VS VIP port.

    b. Without having an admin role or without being network owner, Avi admin user will not able to add allowed-address-pair entry to the port created in tenant network. This will cause VS placement to fail. This can be fixed using policy changes. To fix this, you can add aviadmin role to Avi admin user in the tenants, and make policy changes as mentioned above in Neutron Policy Changes section.

4. Tenant Networks ?

    a. When deploying in provider mode, tenant networks must be shared with Avi admin tenant (avilbaas tenant in example). Without this, Avi Controller will not be able to connect Service Engine to the networks. For instance,

```
neutron rbac-create --target-tenant [avilbaas-tenant-uuid] --action access_as_shared --type network [network-uu
```

    b. When deploying in non-provider mode (dedicated SE mode, SE in tenant context), management network should be shared in all tenants that you want to deploy Avi LB. Tenant networks used for creating Virtual Service VIP or backend pool members need not be shared with avilbaas tenant.

5. FIP allocation ? The routers connecting the tenant networks to provider networks must be created in the tenant. If the routers are created in admin tenant (or any other tenant), Avi controller will not be able to access them and identify which provider networks to use for FIP allocation.

    Note: For using floating IP feature, having router in different tenant will not work even if you manually pass the floating network/subnet UUID in VS VIP request. This will not work because the access is limited to tenant in which VS is being created, and if Avi Controller cannot see the VIP network connected to a router, it will not be able to see the FIP network as well. FIP networks are derived from the tenant router?s interfaces to provider networks, and requires the routers to be the tenant hosting the VIP networks.

Upgrade Feature

The existing deployments are using admin credentials with admin role. The user will re-configure the cloud to use new credentials with aviadmin role.

Case 1:

OpenStack admin tenant is used as Avi admin tenant ?

You ned to add aviadmin role to the Avi admin user in admin tenant, and share all the tenant networks with admin tenant. Upgrading to use this feature will not be disruptive.

Case 2:

OpenStack non-admin tenant is used as Avi admin tenant ?

In this case, aviadmin role is added to Avi admin user in this tenant. This tenant is configured as Avi admin tenant in Avi Cloud config. All the tenant networks need to be shared with this tenant. Upgrading to use this feature will not be disruptive.

Case 3:

Changing the Avi admin tenant in cloud ?

This will be disruptive for some instances, since Avi admin tenant will be used to spin up the SEs. Configuring different Avi admin tenant would mean using the other tenant to host all the SEs. This is disruptive, and all the VSes need to be disabled /enabled in a maintenance window.