Report #1: Cloud Migration & Environment Stand-Up Assessment

Bryan Daniel, Justin Cockrell, Dylan Hagy
University of South Carolina Aiken
CSCI A591-PC
Dr. Ali AlSabeh
Spring 2026

Scenario: Southriver Software Solutions — Role-Based Access Control Hardening in an OpenStack Private Cloud Environment

Prerequisites

- Basic Linux command-line familiarity
- Introductory knowledge of virtualization and cloud concepts
- Familiarity with OpenStack identity, compute, and networking terminology
- Access to the provided DevStack-based OpenStack environment
- Snapshot or persistent instance of the NetLab OpenStack deployment

Learning Objectives

By the end of this lab, students will be able to:

- Analyze a baseline OpenStack environment that relies on default Keystone roles [6]
- Identify weaknesses introduced by overly permissive role assignments
- Design and implement group-based identity structures using custom roles [1], [3], [4]
- Enforce least-privilege access using service-level RBAC policy files [7], [9]
- Validate security controls by testing authorized and unauthorized actions

Background

Southriver Software Solutions is in the process of transitioning from developer-managed infrastructure toward a centralized private cloud platform built on OpenStack. The environment was initially deployed to support rapid experimentation and proof-of-concept workloads, prioritizing functionality over formal access controls or governance.

As a result, the cloud relies heavily on default Keystone roles and direct user-to-project assignments [2]. While this configuration allows the environment to function, it provides limited separation of duties and introduces unnecessary risk as additional users and teams begin sharing the platform.

To support future growth and security initiatives, Southriver must first address how identity, access, and authorization are structured within the cloud [1], [5]. Rather than immediately deploying restrictive controls, the organization has chosen to document the baseline configuration, redesign identity assignments, and enforce access boundaries in a controlled, verifiable manner.

Business Scenario

Southriver Software Solutions has engaged a consulting team to assess and remediate role-based access control (RBAC) weaknesses within its persistent OpenStack environment. Although the cloud is operational, access to compute resources is currently governed by default roles that do not reflect organizational job functions or least-privilege principles.

The goal of this engagement is to transition the environment from a permissive baseline toward a structured RBAC model using groups, custom roles, and service-level policy enforcement. This effort establishes a secure foundation that can be expanded upon in later engagements involving networking segmentation, logging, and advanced security controls.

Mission Overview

As members of Southriver's consulting team, your mission in this lab is to incrementally harden access control within the OpenStack environment. Specifically, you will:

- Document the baseline RBAC configuration and its shortcomings
- Redesign identity assignments using groups and custom roles
- Enforce role-based permissions at the compute service level [7], [9]
- Validate that security controls prevent unauthorized actions

Each section of the lab builds upon the previous one, demonstrating how identity design and policy enforcement work together to implement least-privilege access in OpenStack.

Section 0: Baseline RBAC Configuration (Default and Over-Permissive)

Before implementing any changes, it is essential to understand the environment's current access model. In this section, the consulting team documents the baseline Keystone configuration, where users are assigned default roles directly within projects [2], [6].

This baseline reflects a common early-stage deployment pattern in which roles such as member and reader are used broadly without enforcing meaningful separation of duties [6]. While functional, this approach allows users to perform actions outside their intended responsibilities.

Section 0 – Quick Assessment of the Environment

- Inventory existing projects, users, roles, and role assignments
- Identify use of default Keystone roles across tenant projects
- Observe compute access behavior under the baseline configuration
- Document over-permission and lack of role granularity

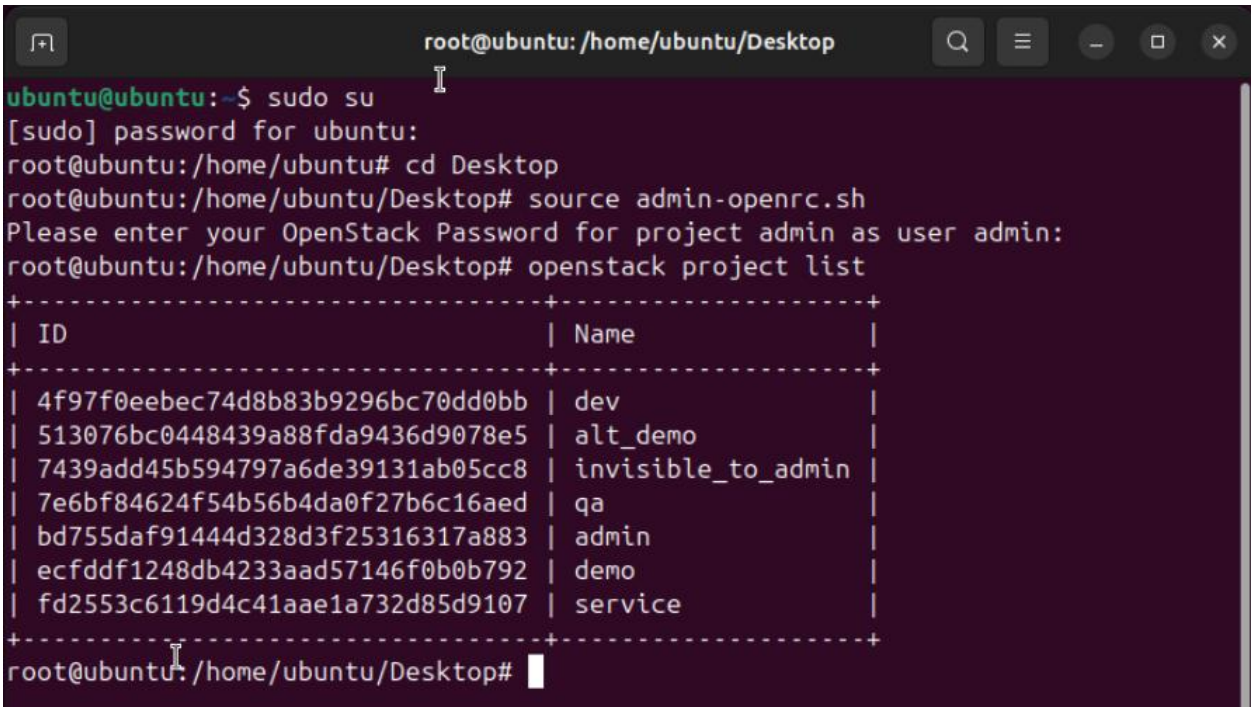
Purpose of This Section

The purpose of Section 0 is to document the existing access control posture of the OpenStack environment prior to any hardening. This establishes a measurable baseline and highlights the risks introduced by reliance on default Keystone roles. No configuration changes are made in this section.

0.1 Inventory of Projects, Users, and Roles

List existing projects

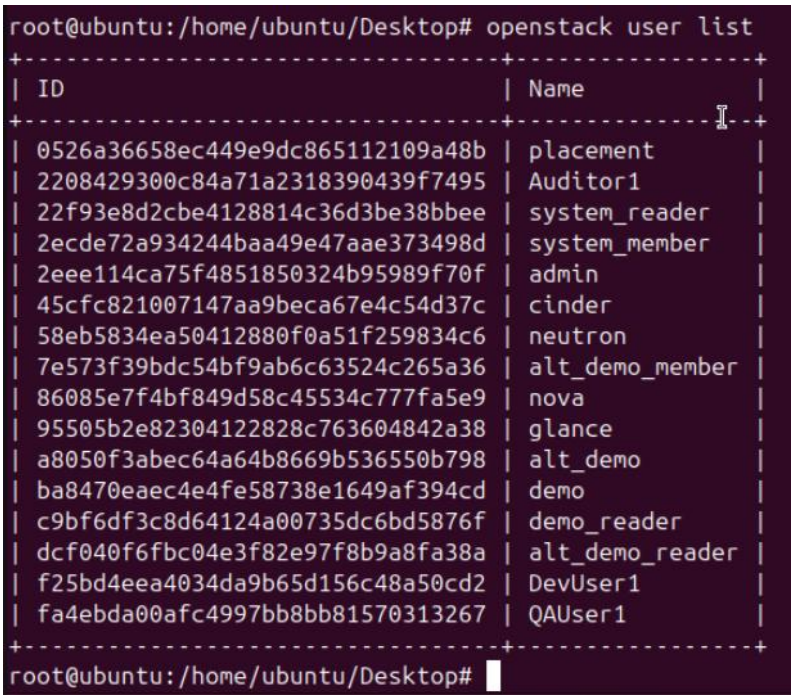
Step 1: Open a terminal with root privileges and source the admin-openrc.sh file to authenticate as the administrative user. Once authenticated, list all OpenStack projects to establish a baseline view of the current environment. The root password for this environment is 'password', and all OpenStack user passwords are 'secret'.



The three projects we will be working with are the dev, qa, and admin projects.

List users

Step 2: Next, list all existing users to identify the accounts currently defined in the environment.



There are many Users here, but we will be working with the following: admin, Auditor1, DevUser1, and QAUser1.

List available roles

Step 3: Now, list all existing roles to understand the different permissions that can be assigned to users and projects.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role list
+----------------------------------+---------------+
| ID                               | Name          |
+----------------------------------+---------------+
| 0d37478aaf6547ca9431a61875ac4581 | service       |
| 2bbb2fc206bb4097bd9a9584433b3251 | reader        |
| 2c88f95e06554c44ad4c61e238b6d830 | member        |
| 525553ab65ab4822a2cfa7b16e5e849a | manager       |
| 7f62fa9295294cfba8540fedf74a8478 | ResellerAdmin |
| 8dba605dda2f4199bac4e12478c0979e | anotherrole   |
| b6cebb4885fe4b678d2159db040ce296 | admin         |
+----------------------------------+---------------+
root@ubuntu:/home/ubuntu/Desktop#
```

It appears that only default roles are available to assign. Let's note this for later.

0.2 Baseline Role Assignments by Project

Role assignments in the dev and qa projects

Step 4: Next, we will observe what roles are assigned in each of the three main projects. First, let's look at the dev project.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project dev -
-names
+--------+-------------+-------+-------------+--------+--------+-----------+
| Role   | User        | Group | Project     | Domain | System | Inherited |
+--------+-------------+-------+-------------+--------+--------+-----------+
| reader | Auditor1@De |       | dev@Default |        |        | False     |
|        | fault       |       |             |        |        |           |
| admin  | admin@Defau |       | dev@Default |        |        | False     |
|        | lt          |       |             |        |        |           |
| member | DevUser1@De |       | dev@Default |        |        | False     |
|        | fault       |       |             |        |        |           |
+--------+-------------+-------+-------------+--------+--------+-----------+
```

For this project, Auditor1 has read-only access, and DevUser1 has member-level access. Devs should be allowed to create/delete instances, so this is not an issue so far.

Step 5: Now, let's observe the role list for the qa project.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project qa --
names
+--------+-----------------+-------+-----------+--------+--------+-----------+
| Role   | User            | Group | Project   | Domain | System | Inherited |
+--------+-----------------+-------+-----------+--------+--------+-----------+
| reader | Auditor1@Default |       | qa@Default |        |        | False     |
| admin  | admin@Default    |       | qa@Default |        |        | False     |
| member | QAUser1@Default  |       | qa@Default |        |        | False     |
+--------+-----------------+-------+-----------+--------+--------+-----------+
root@ubuntu:/home/ubuntu/Desktop#
```

For the qa project, Auditor1 has read-only access, which isn't an issue. However, QAUser1 should only be allowed to reboot an instance if needed.

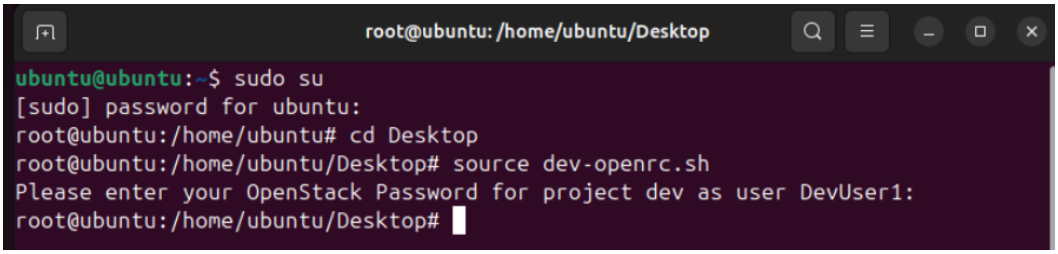
0.3 Behavioral Observation: Default Role Capabilities

At this stage, authorization is enforced only at the identity layer. No service-level policy restrictions have been applied [7], [9].

To observe baseline behavior, authenticate as a project member and enumerate compute resources.

Member role behavior (dev project)

Step 6: Close the current terminal, and repeat the initial steps from section 0.1 to authenticate as a dev user with the dev-openrc.sh file on the desktop.



You are now authenticated as DevUser1.

Step 7: Next, create an instance as DevUser1.

```
root@ubuntu:/home/ubuntu/Desktop# openstack server create --image cirros-0.6.3-x
86_64-disk --flavor m1.tiny --network shared dev-test-instance
+-------------------------------------+--------------------------------------+
| Field                               | Value                                |
+-------------------------------------+--------------------------------------+
| OS-DCF:diskConfig                   | MANUAL                               |
| OS-EXT-AZ:availability_zone         | None                                 |
| OS-EXT-SRV-ATTR:host                | None                                 |
| OS-EXT-SRV-ATTR:hostname            | dev-test-instance                    |
| OS-EXT-SRV-ATTR:hypervisor_hostname | None                                 |
| OS-EXT-SRV-ATTR:instance_name       | None                                 |
| OS-EXT-SRV-ATTR:kernel_id           | None                                 |
| OS-EXT-SRV-ATTR:launch_index        | None                                 |
| OS-EXT-SRV-ATTR:ramdisk_id          | None                                 |
| OS-EXT-SRV-ATTR:reservation_id      | None                                 |
| OS-EXT-SRV-ATTR:root_device_name    | None                                 |
| OS-EXT-SRV-ATTR:user_data           | None                                 |
| OS-EXT-STS:power_state              | N/A                                  |
| OS-EXT-STS:task_state               | scheduling                           |
| OS-EXT-STS:vm_state                 | building                             |
```

Creating the instance worked. Since QAUser1 has the same role, they can also create instances like DevUser1.

Step 8: Now, view the server list, delete the active instance as DevUser1, and then view the server list again.

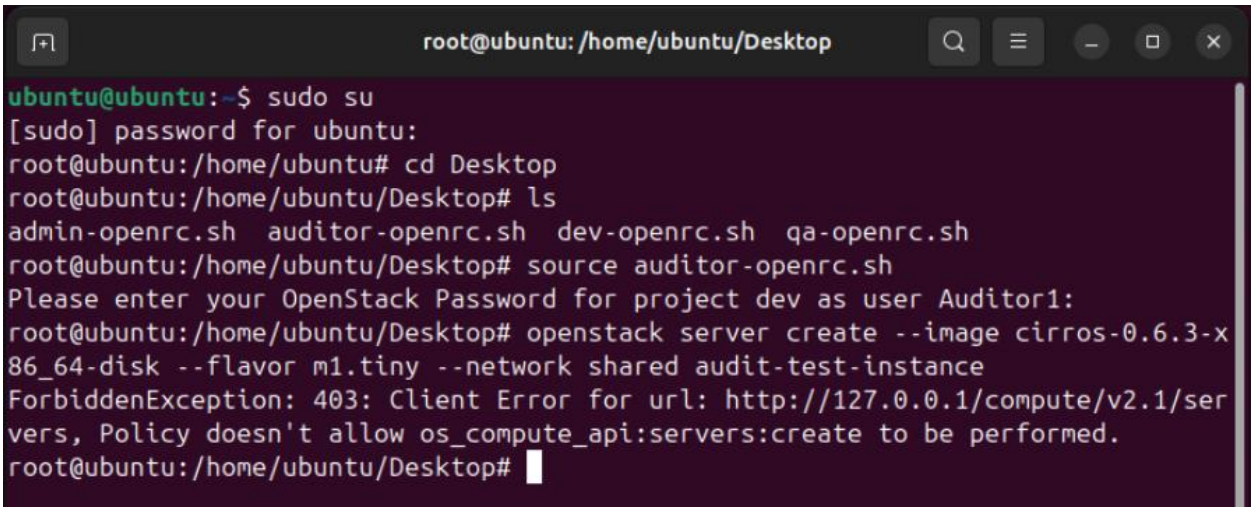
```
root@ubuntu:/home/ubuntu/Desktop# openstack server list
+--------------+--------------+--------+--------------+--------------+---------+
| ID           | Name         | Status | Networks     | Image        | Flavor  |
+--------------+--------------+--------+--------------+--------------+---------+
| b3816835-    | dev-test-    | ACTIVE | shared=192.1 | cirros-      | m1.tiny |
| 8625-46b2-   | instance     |        | 68.233.187   | 0.6.3-       |         |
| 83b9-        |              |        |              | x86_64-disk  |         |
| a424bedf6b36 |              |        |              |              |         |
+--------------+--------------+--------+--------------+--------------+---------+
root@ubuntu:/home/ubuntu/Desktop# openstack server delete dev-test-instance
root@ubuntu:/home/ubuntu/Desktop# openstack server list

root@ubuntu:/home/ubuntu/Desktop#
```

As you can see, members have full access to create/delete instances.

Reader role behavior (auditor)

Step 9: Now, repeat the same steps above to authenticate as Auditor1 to see if the reader role can create an instance in the dev project.



Readers are not allowed to create instances. They are only allowed to view resources.

0.4 Identified Weaknesses in Baseline Configuration

Based on the observations above, the following weaknesses are present:

- Default roles (member, reader) are coarse-grained [6]
- Users are assigned roles directly, without groups [2], [3]
- No alignment exists between roles and organizational job functions
- The member role permits actions beyond least-privilege expectations [6], [8]

Section 0 Outcome

The baseline environment is functional but overly permissive. This section establishes a clear starting point against which RBAC hardening improvements can be measured.

Section 1: Identity Refactoring with Groups and Custom Roles

With the baseline documented, the next step is to redesign how identity and access are structured. In this section, the consulting team introduces custom roles aligned with job functions and replaces direct user-to-project assignments with group-based RBAC [2], [3].

Default role assignments are removed, and users are instead granted access through group membership. This improves clarity, scalability, and auditability while preparing the environment for fine-grained service-level policy enforcement in Section 2 [7], [9].

Section 1 – Quick Assessment of the Goal

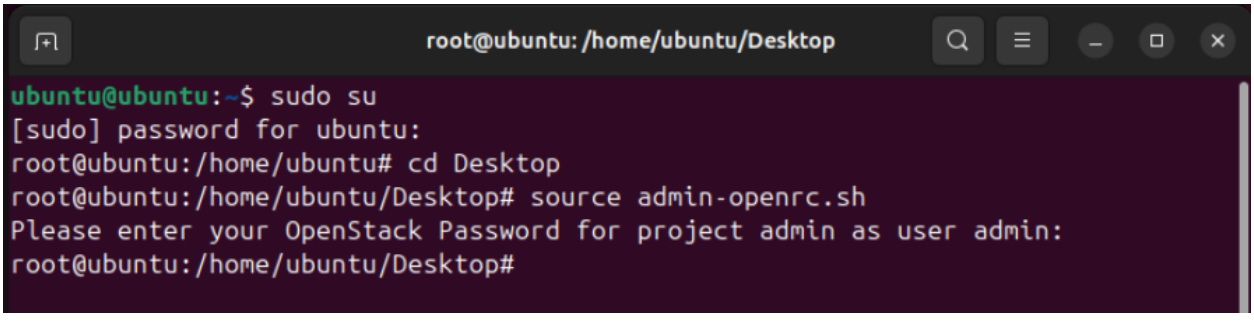
- Create custom job-function roles: dev_compute, qa_compute, audit_read
- Create Keystone groups: DevTeam, QATeam, AuditTeam
- Add users to groups
- Remove direct default role assignments (member, reader)
- Assign custom roles to groups per project
- Verify effective access is now group-based

Purpose of This Section

The purpose of Section 1 is to migrate from direct user-based role assignments using default roles to a group-driven RBAC model using custom roles. No Nova policy enforcement is applied yet; the goal is to restructure identity cleanly in Keystone [1].

1.1 Authenticate as Admin

Step 1: First, begin by opening a terminal with root access and sourcing the admin-openrc.sh file to authenticate as the admin in OpenStack.



You now have administrative privileges in OpenStack.

1.2 Create Custom Roles (Job-Function Roles)

Step 2: Next, we create three custom roles: dev_compute, qa_compute, and audit_read. These roles represent distinct compute-level responsibilities and will later be mapped to Nova policy rules to enforce separation of duties. The roles will be assigned to user groups in subsequent steps, rather than directly to individual users.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role create dev_compute
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| id          | 20dd4f94259549b39a030d4a47b21887 |
| name        | dev_compute                      |
| domain_id   | None                             |
| description | None                             |
+-------------+----------------------------------+
root@ubuntu:/home/ubuntu/Desktop# openstack role create qa_compute
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| id          | 4577276bcc824386b2189c61e7c0d70e |
| name        | qa_compute                       |
| domain_id   | None                             |
| description | None                             |
+-------------+----------------------------------+
root@ubuntu:/home/ubuntu/Desktop# openstack role create audit_read
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| id          | 1dea663380214c78ac6299578e9aa902 |
| name        | audit_read                       |
| domain_id   | None                             |
| description | None                             |
+-------------+----------------------------------+
root@ubuntu:/home/ubuntu/Desktop#
```

The three roles have been created and are ready to be assigned to their respective groups.

Step 3: Next, verify that the roles were created. The output should contain audit_read, dev_compute, and qa_compute.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role list | egrep -i "dev_compute|qa
_compute|audit_read"
| 1dea663380214c78ac6299578e9aa902 | audit_read    |
| 20dd4f94259549b39a030d4a47b21887 | dev_compute   |
| 4577276bcc824386b2189c61e7c0d70e | qa_compute    |
root@ubuntu:/home/ubuntu/Desktop#
```

Use the egrep command to view only the roles you created.

1.3 Create Keystone Groups

Step 4: Next, we create three Keystone groups: DevTeam, QATeam, and AuditTeam. These groups are used to organize users by function and serve as attachment points for role assignments. By assigning roles to groups rather than individual users, the environment enforces consistent access controls and simplifies ongoing identity management.

```
root@ubuntu:/home/ubuntu/Desktop# openstack group create DevTeam
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | default                          |
| id          | 3fb0e57f09bf47abbfc7ef42d52dd5de |
| name        | DevTeam                          |
+-------------+----------------------------------+
root@ubuntu:/home/ubuntu/Desktop# openstack group create QATeam
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | default                          |
| id          | 8ac9308152e442bc9392fedbddb68512 |
| name        | QATeam                           |
+-------------+----------------------------------+
root@ubuntu:/home/ubuntu/Desktop# openstack group create AuditTeam
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | default                          |
| id          | 094bcd0dd06545e99eb2056f8ce31298 |
| name        | AuditTeam                        |
+-------------+----------------------------------+
```

The three Keystone groups have been created and are ready to have roles attached.

Step 5: Verify that the groups were created successfully. The output should include DevTeam, QATeam, and AuditTeam.

```
root@ubuntu:/home/ubuntu/Desktop# openstack group list | egrep -i "DevTeam|QATea
m|AuditTeam"
| 094bcd0dd06545e99eb2056f8ce31298 | AuditTeam |
| 3fb0e57f09bf47abbfc7ef42d52dd5de | DevTeam   |
| 8ac9308152e442bc9392fedbddb68512 | QATeam    |
root@ubuntu:/home/ubuntu/Desktop#
```

With only three groups, the egrep command is not necessary, but it helps in an enterprise environment, where there may be many more groups to filter through.

1.4 Add Users to Groups

Step 6: Next, users are added to their respective Keystone groups based on functional responsibility. This step establishes group membership only. No permissions are granted until roles are later assigned to these groups within specific projects.

```
root@ubuntu:/home/ubuntu/Desktop# openstack group add user DevTeam DevUser1
root@ubuntu:/home/ubuntu/Desktop# openstack group add user QATeam QAUser1
root@ubuntu:/home/ubuntu/Desktop# openstack group add user AuditTeam Auditor1
root@ubuntu:/home/ubuntu/Desktop#
```

Even though there is no success message, the users are now added to their respective groups.

Step 7: Verify that each user has been successfully added to the appropriate group. Each command returns a success message if the user is a member of the specified group.

```
root@ubuntu:/home/ubuntu/Desktop# openstack group contains user DevTeam DevUser1
DevUser1 in group DevTeam
root@ubuntu:/home/ubuntu/Desktop# openstack group contains user QATeam QAUser1
QAUser1 in group QATeam
root@ubuntu:/home/ubuntu/Desktop# openstack group contains user AuditTeam Audito
r1
Auditor1 in group AuditTeam
root@ubuntu:/home/ubuntu/Desktop#
```

All three users were successfully added to the proper group.

1.5 Remove Default Direct Role Assignments

Step 8: Before modifying role assignments, document the current baseline configuration for reference. This captures the default direct user-to-role bindings that will be removed in this step.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project dev -
-names
+--------+-------------+-------+-------------+--------+--------+-----------+
| Role   | User        | Group | Project     | Domain | System | Inherited |
+--------+-------------+-------+-------------+--------+--------+-----------+
| reader | Auditor1@De |       | dev@Default |        |        | False     |
|        | fault       |       |             |        |        |           |
| admin  | admin@Defau |       | dev@Default |        |        | False     |
|        | lt          |       |             |        |        |           |
| member | DevUser1@De |       | dev@Default |        |        | False     |
|        | fault       |       |             |        |        |           |
+--------+-------------+-------+-------------+--------+--------+-----------+
```

The current role assignment list for the dev project.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project qa --
names
+--------+-----------------+-------+-----------+--------+--------+-----------+
| Role   | User            | Group | Project   | Domain | System | Inherited |
+--------+-----------------+-------+-----------+--------+--------+-----------+
| reader | Auditor1@Default |       | qa@Default |        |        | False     |
| admin  | admin@Default    |       | qa@Default |        |        | False     |
| member | QAUser1@Default  |       | qa@Default |        |        | False     |
+--------+-----------------+-------+-----------+--------+--------+-----------+
root@ubuntu:/home/ubuntu/Desktop#
```

The current role assignment list for the qa project.

Step 9: Next, remove default roles that were assigned directly to users. Administrative role assignments are left intact.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role remove --project dev --user Dev
User1 member
root@ubuntu:/home/ubuntu/Desktop# openstack role remove --project dev --user Aud
itor1 reader
root@ubuntu:/home/ubuntu/Desktop#
```

Now, only the admin user has administrative privileges in the dev project.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role remove --project qa --user QAUs
er1 member
root@ubuntu:/home/ubuntu/Desktop# openstack role remove --project qa --user Audi
tor1 reader
root@ubuntu:/home/ubuntu/Desktop#
```

The qa project roles are now set up the same as the dev project.

Step 10: Re-list role assignments to confirm that baseline user assignments have been removed and only administrative access remains.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project dev -
-names
+-------+--------------+-------+-------------+--------+--------+-----------+
| Role  | User         | Group | Project     | Domain | System | Inherited |
+-------+--------------+-------+-------------+--------+--------+-----------+
| admin | admin@Default |       | dev@Default |        |        | False     |
+-------+--------------+-------+-------------+--------+--------+-----------+
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project qa --
names
+-------+--------------+-------+------------+--------+--------+-----------+
| Role  | User         | Group | Project    | Domain | System | Inherited |
+-------+--------------+-------+------------+--------+--------+-----------+
| admin | admin@Default |       | qa@Default |        |        | False     |
+-------+--------------+-------+------------+--------+--------+-----------+
root@ubuntu:/home/ubuntu/Desktop#
```

Only admin has a role in both projects as planned.

1.6 Assign Custom Roles to Groups (Project Scoped)

Step 11: With baseline user assignments removed, custom roles are now assigned to groups within their respective projects. This enforces group-based RBAC and restores required access through controlled role bindings.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role add --project dev --group DevTe
am dev_compute
root@ubuntu:/home/ubuntu/Desktop# openstack role add --project qa --group QATeam
 qa_compute
root@ubuntu:/home/ubuntu/Desktop# openstack role add --project dev --group Audit
Team audit_read
root@ubuntu:/home/ubuntu/Desktop# openstack role add --project qa --group AuditT
eam audit_read
root@ubuntu:/home/ubuntu/Desktop#
```

The custom roles are now assigned to the proper Keystone groups.

Step 12: Verify that roles are now granted via group membership rather than direct user assignment.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project dev -
-names
+-----------+-----------+-----------+-----------+--------+--------+-----------+
| Role      | User      | Group     | Project   | Domain | System | Inherited |
+-----------+-----------+-----------+-----------+--------+--------+-----------+
| admin     | admin@Def |           | dev@Defau |        |        | False     |
|           | ault      |           | lt        |        |        |           |
| audit_rea |           | AuditTeam | dev@Defau |        |        | False     |
| d         |           | @Default  | lt        |        |        |           |
| dev_compu |           | DevTeam@D | dev@Defau |        |        | False     |
| te        |           | efault    | lt        |        |        |           |
+-----------+-----------+-----------+-----------+--------+--------+-----------+
```

The dev project now enforces group-based role assignments using custom RBAC roles.

```
root@ubuntu:/home/ubuntu/Desktop# openstack role assignment list --project qa --
names
+-----------+-----------+-----------+-----------+--------+--------+-----------+
| Role      | User      | Group     | Project   | Domain | System | Inherited |
+-----------+-----------+-----------+-----------+--------+--------+-----------+
| admin     | admin@Def |           | qa@Defaul |        |        | False     |
|           | ault      |           | t         |        |        |           |
| audit_rea |           | AuditTeam | qa@Defaul |        |        | False     |
| d         |           | @Default  | t         |        |        |           |
| qa_comput |           | QATeam@De | qa@Defaul |        |        | False     |
| e         |           | fault     | t         |        |        |           |
+-----------+-----------+-----------+-----------+--------+--------+-----------+
root@ubuntu:/home/ubuntu/Desktop#
```

The qa project also enforces group-based role assignments using custom RBAC roles.

Section 1 Outcome

At the conclusion of this section:

- Default roles (member, reader) no longer define access for your lab users
- Access is granted via group membership
- Job-function roles are now present (dev_compute, qa_compute, audit_read)
- You are ready to bind those roles to real permissions in Nova (Section 2)

Section 2: Enforcing Role-Based Permissions via Service Policy Files

Identity structure alone does not restrict what actions users may perform within a project. OpenStack enforces authorization at the service layer through policy configuration files that map roles to API actions [7], [9].

In this section, the consulting team configures the Nova policy file to enforce least privilege for compute operations using the custom roles created in Section 1.

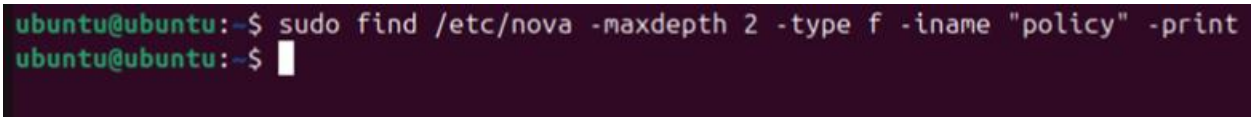
Purpose of This Section

The purpose of Section 2 is to ensure Nova enforces compute permissions based on [7], [8]:

- dev_compute (full instance lifecycle)
- qa_compute (limited lifecycle: reboot; read access)
- audit_read (read-only)

2.1 Locate Nova Policy File (Do this first)

Step 1: Search for existing policy files



Step 2: Examine Nova directory structure

```
ubuntu@ubuntu:~$ sudo ls -la /etc/nova
total 48
drwxr-xr-x   3 stack root   4096 Sep 12 19:47 .
drwxr-xr-x 172 root  root  12288 Sep 12 19:48 ..
-rw-r--r--   1 stack stack  3037 Sep 12 19:43 api-paste.ini
-rw-r--r--   1 stack stack   417 Sep 12 19:43 nova-api-uwsgi.ini
-rw-r--r--   1 stack stack  1299 Sep 12 19:46 nova_cell1.conf
-rw-r--r--   1 stack stack  3659 Sep 12 19:46 nova.conf
-rw-r--r--   1 stack stack  3835 Sep 12 19:47 nova-cpu.conf
-rw-r--r--   1 stack stack   373 Sep 12 19:43 nova-metadata-uwsgi.ini
-rw-r--r--   1 root  root   1114 Sep 12 19:43 rootwrap.conf
drwxr-xr-x   2 root  root   4096 Sep 12 19:43 rootwrap.d
ubuntu@ubuntu:~$
```

Step 3: Check current Nova configuration

```
ubuntu@ubuntu:~$ sudo grep -i policy /etc/nova/nova.conf
[oslo_policy]
ubuntu@ubuntu:~$
```

2.2 Creating Custom Policy Rules

In this section, we will be creating a YAML file that tells Nova exactly which roles can perform which compute operations.

Step 4: Create the policy file

```
ubuntu@ubuntu:~$ sudo nano /etc/nova/policy.yaml
```

Step 5: Add the policy rules

```
# --- Southriver RBAC Lab Policy Overrides ---

# Read/list/show servers (allow all three roles to observe)
"os_compute_api:servers:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:servers:detail": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:servers:show": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-simple-tenant-usage:show": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:images:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:images:detail": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:flavors:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:flavors:detail": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-flavor-extra-specs:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-flavor-extra-specs:show": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:server-groups:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:servers:show:flavor-extra-specs": "role:admin or role:dev_compute or role:qa_compute or role:audit_read
"os_compute_api:os-flavor-extra-specs:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-extended-server-attributes": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:servers:show:host_status": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:servers:show:host_status:unknown-only": "role:admin or role:dev_compute or role:qa_compute or role:audi
"os_compute_api:os-services:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-services:detail": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-services:list": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-hypervisors:list-detail" : "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_compute_api:os-hypervisors:index": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
"os_comute_api:os-hypervisors:show": "role:admin or role:dev_compute or role:qa_compute or role:audit_read"
```

```
# Create instances (dev only)
"os_compute_api:servers:create": "role:admin or role:dev_compute"
"os_compute_api:servers:create:attach_volume": "role:admin or role:dev_compute"
"os_compute_api:servers:create:attach_network": "role:admin or role:dev_compute"
# Delete instances (dev only)
"os_compute_api:servers:delete": "role:admin or role:dev_compute"

# Reboot instances (dev + QA)
"os_compute_api:servers:reboot": "role:dev_compute or role:qa_compute"
```

Step 6: Set proper permissions

```
ubuntu@ubuntu:~$ sudo chmod 644 /etc/nova/policy.yaml
ubuntu@ubuntu:~$
```

2.3 Configuring Nova to Use Custom Policies

Here we are telling Nova to load our custom policy file instead of using the defaults

Step 7: Backup current configuration

```
ubuntu@ubuntu:~$ sudo cp /etc/nova/nova.conf /etc/nova/nova.conf.backup
ubuntu@ubuntu:~$
```

Step 8: Add policy configuration

```
ubuntu@ubuntu:~$ sudo tee -a /etc/nova/nova.conf << EOF
> [oslo_policy]
> policy_file = /etc/nova/policy.yaml
> EOF
[oslo_policy]
policy_file = /etc/nova/policy.yaml
ubuntu@ubuntu:~$
```

Step 9: Verify configuration

```
ubuntu@ubuntu:~$ sudo tail -5 /etc/nova/nova.conf

[os_vif_ovs]
isolate_vif = False
[oslo_policy]
policy_file = /etc/nova/policy.yaml
ubuntu@ubuntu:~$
```

2.4 Restarting Services and Verifying Policy Loading

In this section, we will make Nova reload its configuration and confirm that our policies are active.

Step 10: Restart Nova services

```
ubuntu@ubuntu:~$ sudo systemctl restart devstack@n-api.service
ubuntu@ubuntu:~$ sudo systemctl restart devstack@n-cpu.service
ubuntu@ubuntu:~$ sudo systemctl restart devstack@n-sch.service
```

Step 11: Verify services are running

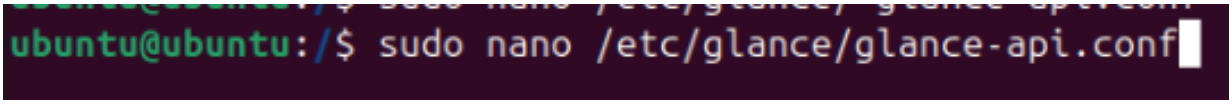
```
ubuntu@ubuntu:~$ sudo systemctl status devstack@n-api.service
● devstack@n-api.service - Devstack devstack@n-api.service
     Loaded: loaded (/etc/systemd/system/devstack@n-api.service; enabled; preset: enabled)
     Active: active (running) since Wed 2026-01-28 21:09:32 EST; 2min 31s ago
   Main PID: 10897 (uwsgi)
     Status: "uWSGI is ready"
      Tasks: 3 (limit: 9200)
     Memory: 252.8M (peak: 253.0M)
        CPU: 14.588s
     CGroup: /system.slice/system-devstack.slice/devstack@n-api.service
             ├─10897 "nova-apiuWSGI master"
             ├─10900 "nova-apiuWSGI worker 1"
             └─10901 "nova-apiuWSGI worker 2"
```

Step 12: Check policy loading

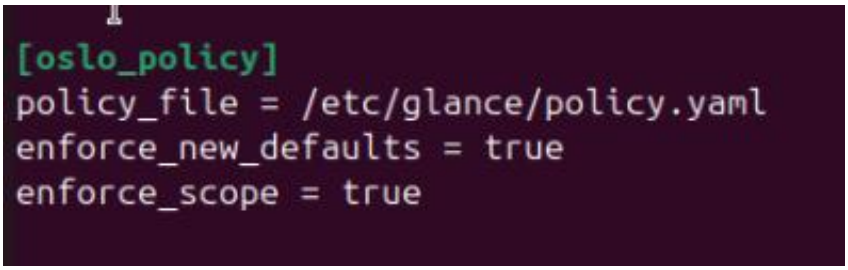
When grepping policy within devstack@n-api.service, [oslo_policy] has successfully been applied.

2.5 Update Glance Policy File and Restart Service

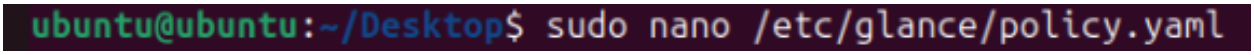
Step 13: Edit the glance-api.conf file



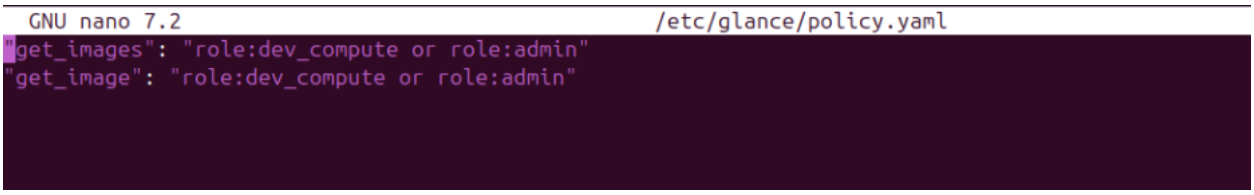
Step 14: Set the policy file, and ensure that enforce_new_defaults and enforce_scope are set to true



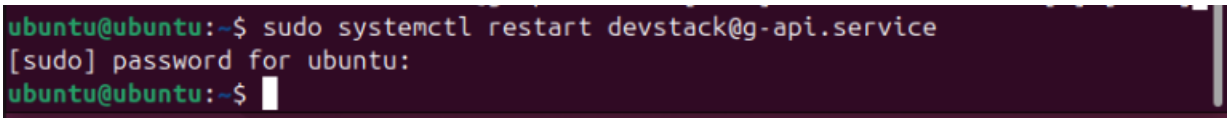
Step 15: Create the Glance policy.yaml file



Step 16: Add the below lines to the created policy file to allow the dev_compute and admin roles to retrieve the details of an image



Step 17: Restart the Glance service



Section 2 Outcome

At the end of this section:

- Nova now enforces compute permissions based on your custom roles
- Identity design (Keystone) is now “real” because it drives allowed/denied actions in compute
- Services restarted successfully without errors
- Logs show policy file loading without syntax errors

Section 3: Validation and Security Verification

The final section validates that RBAC controls behave as intended. The consulting team performs authorized and unauthorized actions to confirm policy enforcement prevents privilege misuse while allowing legitimate workflows [7], [9].

Purpose of This Section

The purpose of Section 3 is to produce measurable evidence that:

- Dev can perform full instance lifecycle actions
- QA can perform limited actions (reboot) but not create/delete
- Auditors can view but not modify

3.1 Authenticate Without Sourcing Scripts (Inline env method)

Step 1: Authenticate as DevUser1 in the dev project:

```
ubuntu@ubuntu:~$ export OS_AUTH_URL=http://127.0.0.1/identity \
> export OS_USERNAME=DevUser1 \
> export OS_PASSWORD=secret \
> export OS_PROJECT_NAME=dev \
> export OS_USER_DOMAIN_NAME=default \
> export OS_PROJECT_DOMAIN_NAME=default \
> export OS_IDENTITY_API_VERSION=3
```

Step 2: Verify token:

```
ubuntu@ubuntu:~/Desktop$ openstack token issue
+------------+-----------------------------------------------------------------+
| Field      | Value                                                           |
+------------+-----------------------------------------------------------------+
| expires    | 2026-01-30T02:09:03+0000                                        |
| id         | gAAAAABpfASvvufXQvDAFVOXVISoDZ-                                 |
|            | E5VIXI1HSeTw9ceancSRTzGPWUMqdMysanI24ffbtckelaT00iJUY6AMgmXub-M |
|            | YngfY9MPqC046CA64io_IvU7YiyLCIt6n_BSU9MwzVU26W0nJaA0wUG3swt9Jcp |
|            | dV93xKtU93sIcnkvYtgVyYpwDQ                                      |
| project_id | 4f97f0eebec74d8b83b9296bc70dd0bb                                |
| user_id    | f25bd4eea4034da9b65d156c48a50cd2                                |
+------------+-----------------------------------------------------------------+
```

3.2 Authorized Tests (DevUser1)

Step 3: List images/flavors (environment visibility):

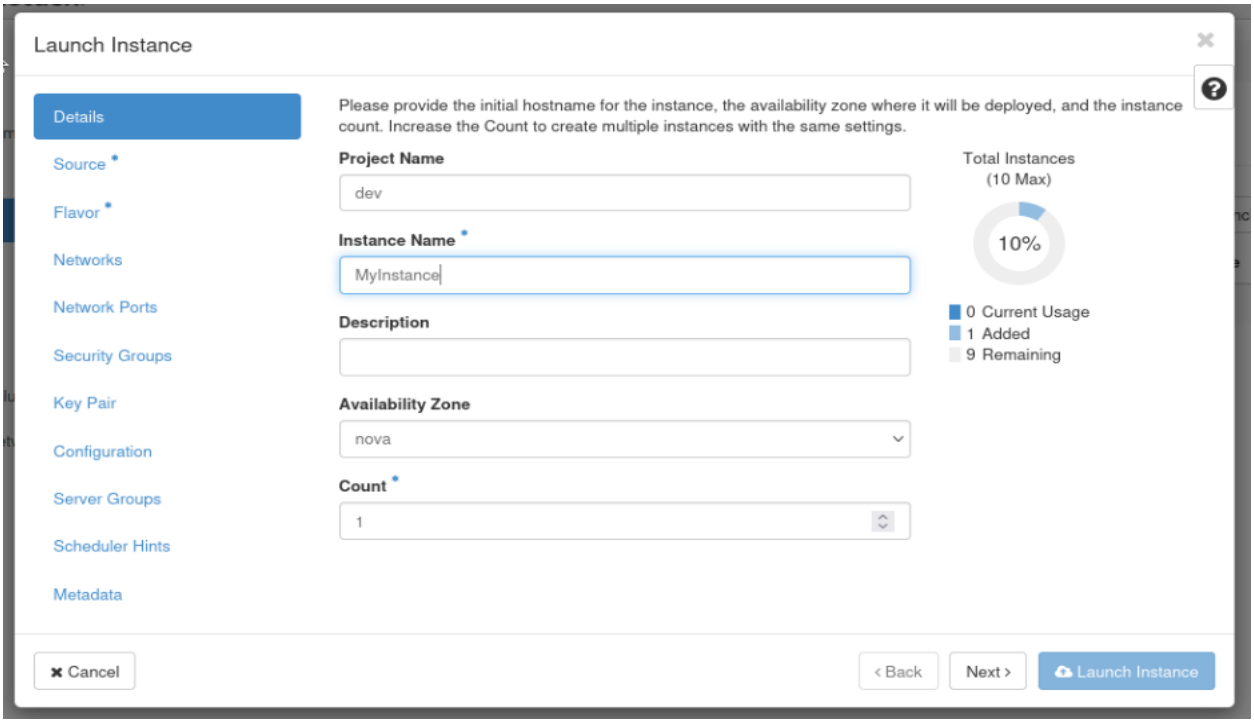
```
ubuntu@ubuntu:~/Desktop$ openstack image list
+--------------------------------------+--------------------------+--------+
| ID                                   | Name                     | Status |
+--------------------------------------+--------------------------+--------+
| efad4afe-b2ce-4f4b-ac7c-9c301efc537a | cirros-0.6.3-x86_64-disk | active |
+--------------------------------------+--------------------------+--------+
```

```
ubuntu@ubuntu:~/Desktop$ openstack flavor list
+----+-----------+-------+------+-----------+-------+-----------+
| ID | Name      |   RAM | Disk | Ephemeral | VCPUs | Is Public |
+----+-----------+-------+------+-----------+-------+-----------+
| 1  | m1.tiny   |   512 |    1 |         0 |     1 | True      |
| 2  | m1.small  |  2048 |   20 |         0 |     1 | True      |
| 3  | m1.medium |  4096 |   40 |         0 |     2 | True      |
| 4  | m1.large  |  8192 |   80 |         0 |     4 | True      |
| 42 | m1.nano   |   192 |    1 |         0 |     1 | True      |
| 5  | m1.xlarge | 16384 |  160 |         0 |     8 | True      |
| 84 | m1.micro  |   256 |    1 |         0 |     1 | True      |
| c1 | cirros256 |   256 |    1 |         0 |     1 | True      |
| d1 | ds512M    |   512 |    5 |         0 |     1 | True      |
| d2 | ds1G      |  1024 |   10 |         0 |     1 | True      |
| d3 | ds2G      |  2048 |   10 |         0 |     2 | True      |
| d4 | ds4G      |  4096 |   20 |         0 |     4 | True      |
+----+-----------+-------+------+-----------+-------+-----------+
```

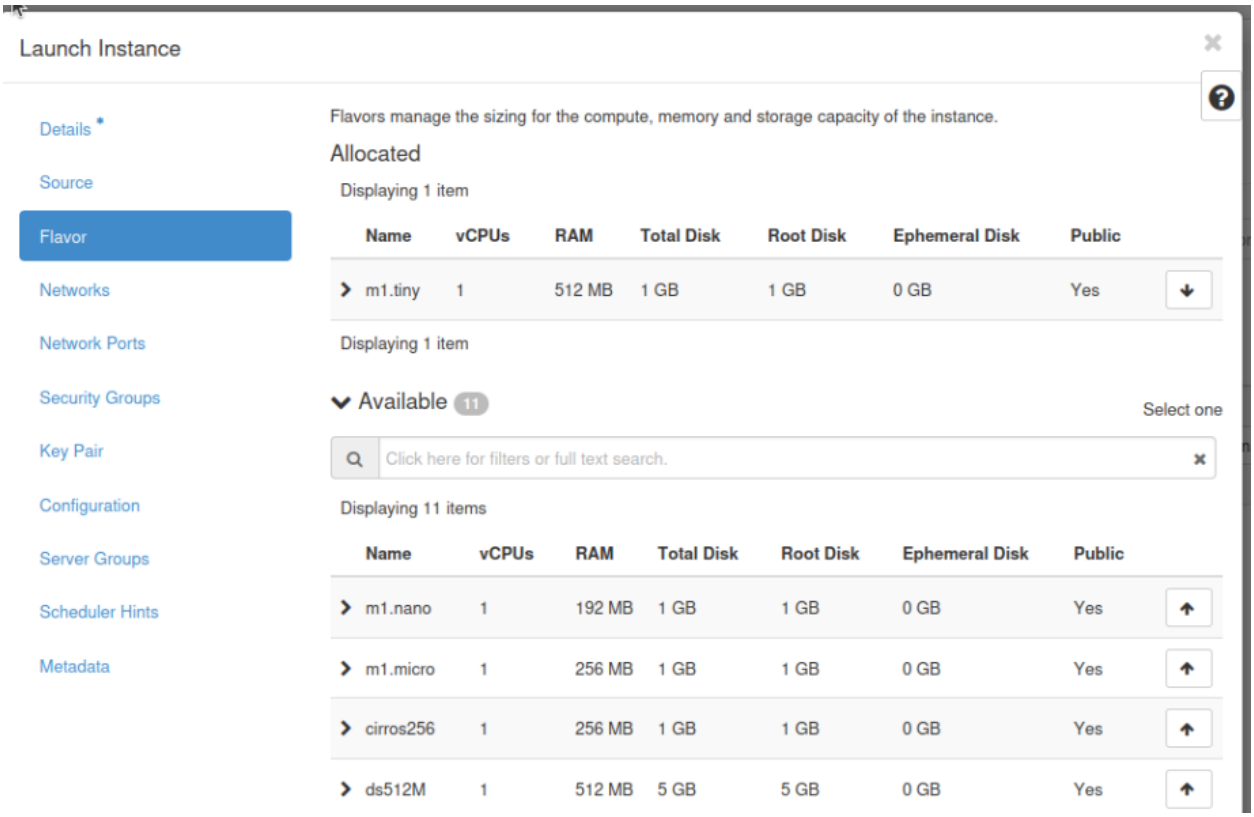
Step 4: List all OpenStack networks:

```
ubuntu@ubuntu:~/Desktop$ openstack network list
+--------------------------------------+--------+--------------------------------------+
| ID                                   | Name   | Subnets                              |
+--------------------------------------+--------+--------------------------------------+
| 24353f0c-c770-4e7a-abd7-196dcb3bc476 | public | 0fd39408-8388-411b-9d1a-f564cfd67967,|
|                                      |        | 95515952-8b36-4743-924d-21b00e17e63c |
| 398b5405-9388-4b39-9937-8e855b33c735 | shared | fd117f77-351a-4a10-a454-beb9919beca3 |
+--------------------------------------+--------+--------------------------------------+
```

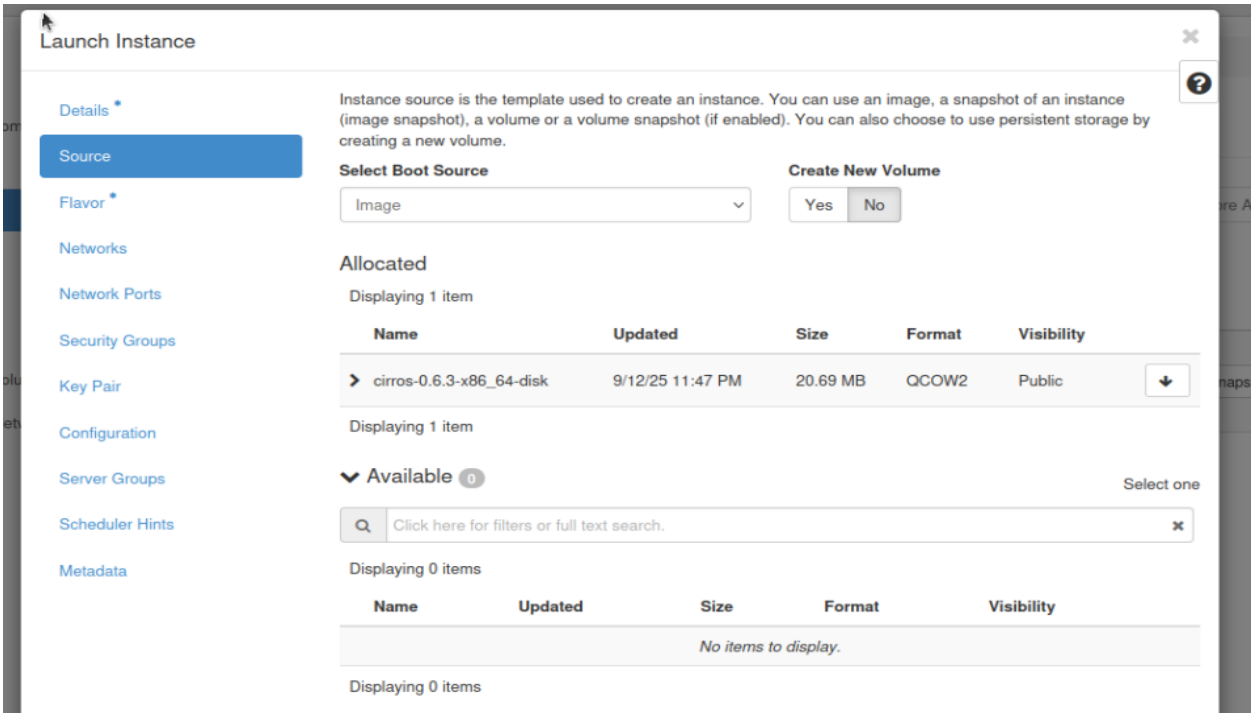
Step 5: Use Horizon to create a new instance with no volume



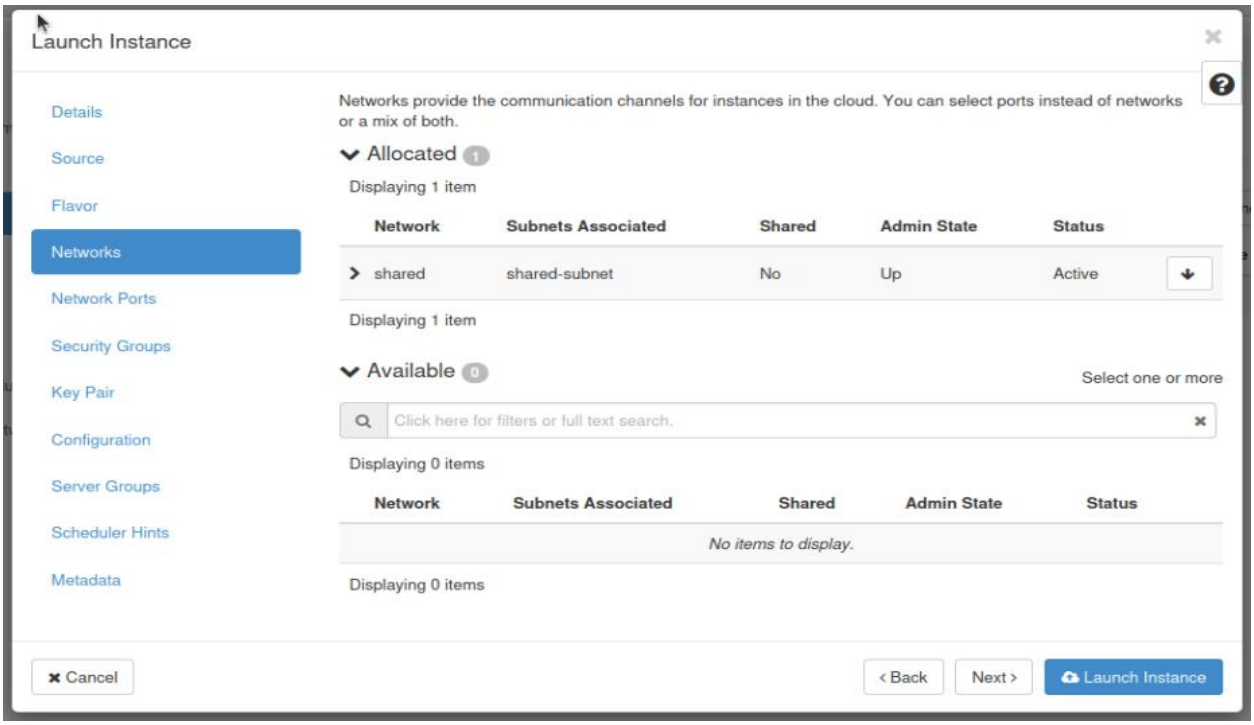
Enter a Name of your Choosing



Under Flavor, select M1.Tiny



Under source, ensure you do not create a volume



Accept the default network (Shared)

Step 6: Confirm the new instance exists using terminal:

```
ubuntu@ubuntu:~/Desktop$ openstack server list
+--------------------------------------+------------+--------+----------------------+--------------------------+---------+
| ID                                   | Name       | Status | Networks             | Image                    | Flavor  |
+--------------------------------------+------------+--------+----------------------+--------------------------+---------+
| afe4032f-db67-4793-90e2-1a02bca7b42d | MyInstance | ACTIVE | shared=192.168.233.26 | cirros-0.6.3-x86_64-disk | m1.tiny |
+--------------------------------------+------------+--------+----------------------+--------------------------+---------+
```

The instance is created and permissions have been verified

Step 7: Delete the instance (Hint: use the command “openstack server delete DevTest1”):

No output is returned

3.3 Unauthorized Tests (QAUser1)

Step 8: Switch identity to qa (either overwrite exports or use a subshell):

```
ubuntu@ubuntu:~$ export OS_USERNAME=QAUser1 \
> export OS_PASSWORD=secret \
> export OS_PROJECT_NAME=qa
```

Step 8: Verify the environment variables are set properly:

```
ubuntu@ubuntu:~/Desktop$ openstack token issue
+------------+----------------------------------------------------------------------------------------------+
| Field      | Value                                                                                        |
+------------+----------------------------------------------------------------------------------------------+
| expires    | 2026-01-30T03:27:52+0000                                                                     |
| id         | gAAAAABpfBcoKBJCrUJLLn4VvT7jwHrJSlJzxCehfRlKYHJ7B7YoIIT_c9t6JMXZHjGSsByv031k71qamC |
|            | GbioUYk0tgzVnHYp3SI1jqJm-                                                                    |
|            | hFTuYp7cJJbczmniBJhLhfDe9qGfnPf_o4AUneZWxFFEvKOeumnmpX8MkZZ2JzKsMvJT9yw01Ud8                 |
| project_id | 7e6bf84624f54b56b4da0f27b6c16aed                                                             |
| user_id    | fa4ebda00afc4997bb8bb81570313267                                                             |
+------------+----------------------------------------------------------------------------------------------+
```

We have refreshed our token to ensure it is under the correct account

Step 9: List all servers in the qa project:

```
ubuntu@ubuntu:~/Desktop$ openstack server list
+--------------------------------------+------+--------+----------+--------------------------+---------+
| ID                                   | Name | Status | Networks | Image                    | Flavor  |
+--------------------------------------+------+--------+----------+--------------------------+---------+
| e860a440-66e3-4b0d-b5dd-3b827930973e | Test | BUILD  |          | N/A (booted from volume) | m1.nano |
+--------------------------------------+------+--------+----------+--------------------------+---------+
```

Step 10: Attempt to create a new instance as a QAUser1 (Hint: use the command "openstack server create –flavor m1.tiny --image cirros-0.6.3-x86_65-disk –network public DevTest1"):

```
ubuntu@ubuntu:~/Desktop$ openstack server create --flavor m1.tiny  --image cirros-0.6.3-x86_65-disk
 --network public DevTest1
ForbiddenException: 403: Client Error for url: http://127.0.0.1/image/v2/images?name=cirros-0.6.3-x
86_65-disk, You are not authorized to complete get_images action.<br /><br />
```

Document the error. This should be a 403 Policy not authorized style failure.

Step 11: Attempt to delete the test server:

```
ubuntu@ubuntu:~/Desktop$ openstack server delete Test
ForbiddenException: 403: Client Error for url: http://127.0.0.1/compute/v2.1/servers/e860a440-66
4b0d-b5dd-3b827930973e, Policy doesn't allow os_compute_api:servers:delete to be performed.
```

3.4 Unauthorized Tests (Auditor1)

Step 12: Switch to Auditor1. Keep project as dev first:

```
ubuntu@ubuntu:~$ export OS_USERNAME=Auditor1 \
> export OS_PASSWORD=secret \
> export OS_PROJECT_NAME=dev
```

Step 13: Verify you are authenticated as Auditor1:

```
ubuntu@ubuntu:~/Desktop$ openstack token issue
+------------+-----------------------------------------------------------------------------------------+
| Field      | Value                                                                                   |
+------------+-----------------------------------------------------------------------------------------+
| expires    | 2026-01-30T03:48:29+0000                                                                |
| id         | gAAAAABpfBv9Hv4n-                                                                       |
|            | fMqIxNMhPQbZKXHdi9jiKDUxIC__xERn72Yq3B9LvtWNxfYZtBkynlaKBl7Jiv5uNA1jXlC7CDAMn5p6Vv |
|            | H2O3Via5-Qtk2_LYk0bmoU7W1PzbNo9x_xBt7Hp1MC-                                             |
|            | Or6a_t1xkrJNm_SLWcx0zu4kTTNpXVB3C331ryxWk                                                |
| project_id | 4f97f0eebec74d8b83b9296bc70dd0bb                                                        |
| user_id    | 2208429300c84a71a2318390439f7495                                                        |
+------------+-----------------------------------------------------------------------------------------+
```

Step 14: List all servers in the dev project:

```
ubuntu@ubuntu:~/Desktop$ openstack server list
+--------------------------+------------+--------+----------------------+--------------------------+---------+
| ID                       | Name       | Status | Networks             | Image                    | Flavor  |
+--------------------------+------------+--------+----------------------+--------------------------+---------+
| afe4032f-db67-4793-90e2- | MyInstance | ACTIVE | shared=192.168.233.26 | cirros-0.6.3-x86_64-disk | m1.tiny |
| 1a02bca7b42d             |            |        |                      |                          |         |
+--------------------------+------------+--------+----------------------+--------------------------+---------+
```

You will see the server we created under the dev account

Step 15: Attempt to create an instance as Auditor1:

```
ubuntu@ubuntu:~/Desktop$ openstack server create --flavor m1.tiny --image cirros-0.6.3-x86_65-disk
--network shared AuditTest
ForbiddenException: 403: Client Error for url: http://127.0.0.1/image/v2/images?name=cirros-0.6.3-x
86_65-disk, You are not authorized to complete get_images action.<br /><br />
```

As expected, the attempt to create an instance was blocked.

3.5 Separation of Duties Verification

Document that:

- Dev role can create/delete/reboot in dev
- QA role cannot create/delete but can reboot.
- Auditor role cannot create/modify anywhere; can view resources

Section 3 Outcome

The validation confirms the RBAC hardening was successful:

- Unauthorized actions are denied by Nova policy enforcement
- Authorized workflows remain functional
- Separation of duties exists across development, QA, and audit roles

References

[1] OpenStack, “Keystone Identity Concepts,” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/keystone/latest/admin/identity-concepts.html. [Accessed: Jan. 30, 2026].

[2] OpenStack, “Role Assignments,” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/keystone/latest/admin/identity-concepts.html#role-assignments. [Accessed: Jan. 30, 2026].

[3] OpenStack, “Groups,” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/keystone/latest/admin/identity-concepts.html#groups. [Accessed: Jan. 30, 2026].

[4] OpenStack, “Roles,” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/keystone/latest/admin/identity-concepts.html#roles. [Accessed: Jan. 30, 2026].

[5] OpenStack, “Authorization Scopes (Project, Domain, System),” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/keystone/latest/admin/identity-concepts.html#authorization-scopes. [Accessed: Jan. 30, 2026].

[6] OpenStack, “Service API Protection and Default Roles,” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/keystone/latest/admin/service-api-protection.html. [Accessed: Jan. 30, 2026].

[7] OpenStack, “Nova Policy Configuration,” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/nova/latest/configuration/policy.html. [Accessed: Jan. 30, 2026].

[8] OpenStack, “Default Policies for Compute (Nova),” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/nova/latest/configuration/policy.html#default-policies. [Accessed: Jan. 30, 2026].

[9] OpenStack, “oslo.policy: Policy Enforcement Library,” *OpenStack Documentation*, 2024. [Online]. Available: https://docs.openstack.org/oslo.policy/latest/. [Accessed: Jan. 30, 2026].