Программное обеспечение «1Stack (1Стек)»

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЕЙ С РОЛЯМИ «АДМИНИСТРАТОР ОБЛАКА», «АДМИНИСТРАТОР», «УЧАСТНИК ПРОЕКТА»

(новая редакция)

2024

Содержание

1 Управление ролями

В платформе 1Stack используется механизм управления доступом к ее ресурсам на основе ролей (role-based access control, RBAC).

Роли (roles) описывают привилегии, которые могут выполняться пользователями. По умолчанию есть две предустановленные роли:

1. Роль участника, связанного с проектом (member).
2. Роль администратора для предоставления возможности администрирования пользователям, которые не являются администраторами.

> В сервисе Identity (keystone) также есть роль читателя (reader), которая отображается в списках ролей. Не используйте роль читателя, поскольку она не интегрирована в другие проекты 1Stack и предоставляет непоследовательные разрешения в различных сервисах.

Также можно создавать кастомные роли для рабочей среды.

1.1 Понимание роли администратора платформы 1Stack

При назначении роли администратора пользователь получает разрешение на просмотр, изменение, создание или удаление любого ресурса в любом проекте. Он может создавать общие ресурсы, доступные для всех проектов (например, общедоступные образы или сети провайдеров).

Пользователь с ролью администратора также может создавать или удалять аккаунты пользователей и управлять ролями.

Проект, в котором назначается пользователю роль администратора, является проектом по умолчанию, в котором выполняется команда **openstack**. Например, в проекте development будет создана сеть с именем **internal-network**, если администратор в проекте с названием **development** выполняет следующую команду:

```
$ openstack network create internal-network
```

Администратор может создавать внутреннюю сеть в любом проекте при помощи параметра **--project**:

```
$ openstack network create internal-network --project testing
```

1.2 Просмотр ролей через интерфейс командной строки (CLI)

Администратор может просматривать информацию о существующих ролях.

Порядок действий:

1. Перечислите доступные предустановленные роли:

```
$ openstack role list
+----------------------------------+------------------+
| ID                               | Name             |
+----------------------------------+------------------+
| 01d92614cd224a589bdf3b171afc5488 | admin            |
| 034e4620ed3d45969dfe8992af001514 | member           |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin    |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user  |
| cfea5760d9c948e7b362abc1d06e557f | reader           |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator    |
| ef3d3f510a474d6c860b4098ad658a29 | service          |
+----------------------------------+------------------+
```

2. Просмотрите информацию о конкретной роли:

```
$ openstack role show admin
```

Пример

```
$ openstack role show admin
+-----------+----------------------------------+
| Field     | Value                            |
+-----------+----------------------------------+
| domain_id | None                             |
| id        | 01d92614cd224a589bdf3b171afc5488 |
| name      | admin                            |
+-----------+----------------------------------+
```

1.3 Создание и назначение ролей через интерфейс командной строки (CLI)

Администратор может создавать и управлять ролями при помощи клиента сервиса идентификации (keystone), с использованием набора команд, описанных далее.

Каждое развертывание платформы 1Stack должно включать как минимум один проект, одного пользователя и одну роль, связанные друг с другом.

Можно назначить пользователей более чем на один проект. Чтобы назначить пользователей на несколько проектов, создайте роль и назначьте ее паре «пользователь-проект».

> Для выбора пользователей, ролей или проектов можно использовать как название (имя), так и ID.

Порядок действий

1. Создайте новую роль:

```
$ openstack role create <role_name>
```

2. Чтобы назначить пользователя на проект, сначала найдите пользователя, роль и названия или ID проектов при помощи следующих команд:
   - openstack user list;
   - openstack role list;
   - openstack project list.
3. Назначьте роль паре **пользователь-проект**.

```
$ openstack role add <role_name> --user <user_name> --project <project_name>
```

Пример далее демонстрирует назначение роли администратора пользователю-администратору в демопроекте:

```
$ openstack role add admin --user admin --project demo
```

4. Проверьте назначение роли пользователю-администратору:

```
$ openstack role assignment list --user <user_name> --project <project_name> --names
```

В следующем примере показана проверка того, что пользователь-администратор назначен на демопроект с ролью администратора:

```
$ openstack role assignment list --user admin --project demo --names
+-------+---------------+-------+--------------+--------+--------+-----------+
| Role  | User          | Group | Project      | Domain | System | Inherited |
+-------+---------------+-------+--------------+--------+--------+-----------+
| admin | admin@Default |       | demo@Default |        |        | False     |
+-------+---------------+-------+--------------+--------+--------+-----------+
```

1.4 Создание неявных ролей (implied roles)

Сервис Identity (keystone) обеспечивает контроль доступа, подтверждая, что пользователю назначена определенная роль. В сервисе Identity используются назначения неявных ролей (implied roles). Если администратор в явном виде назначает пользователю конкретную роль, можно также неявным образом назначить дополнительные роли.

Просмотр неявных ролей по умолчанию в 1Stack:

```
$ openstack implied role list
+----------------------------------+------------------+----------------------------------
-+--------------------+
| Prior Role ID                    | Prior Role Name | Implied Role ID
| Implied Role Name |
+----------------------------------+------------------+----------------------------------
-+--------------------+
| 54454217f38247e5a2131c8a47138d32 | admin            | b59703369e194123b5c77dad60d11a25
| member             |
| b59703369e194123b5c77dad60d11a25 | member           | 382761de4a9c4414b6f8950f8580897c
| reader             |
+----------------------------------+------------------+----------------------------------
-+--------------------+
```

В сервисе Identity (keystone) также добавлена роль читателя (reader), которая отображается в списках ролей. Не используйте роль читателя, поскольку она не интегрирована в другие сервисы 1Stack и предоставляет непоследовательные разрешения в различных сервисах.

Роль с разрешениями более высокого уровня подразумевает разрешения для роли с меньшим количеством разрешений.

В неявных ролях по умолчанию роль администратора (admin) подразумевает участника (member), а роль участника – читателя (reader). При использовании неявных ролей назначения ролей пользователю обрабатываются кумулятивно, таким образом, пользователь наследует подчиненные роли.

Используя кастомные (настраиваемые) роли, администратор может создавать неявные (подразумеваемые) ассоциации.

Созданная новая роль будет иметь такие же политики доступа, как и роль участника по умолчанию.

Порядок действий

- Используйте следующую команду, чтобы установить роль, которая подразумевает другую роль:

```
$ openstack implied role create manager --implied-role poweruser
+------------+----------------------------------+
| Field      | Value                            |
+------------+----------------------------------+
| implies    | ab0b966e0e5e411f8d8b0cc6c26fefd1 |
| prior_role | 880761f64bff4e4a8923efda73923b7a |
+------------+----------------------------------+
```

Проверка

- Получите список всех неявных ролей:

```
$ openstack implied role list
+----------------------------------+-----------------+----------------------------------
-+--------------------+
| Prior Role ID                    | Prior Role Name | Implied Role ID
| Implied Role Name |
+----------------------------------+-----------------+----------------------------------
-+--------------------+
| 54454217f38247e5a2131c8a47138d32 | admin           | b59703369e194123b5c77dad60d11a25
| member           |
| 880761f64bff4e4a8923efda73923b7a | manager         | ab0b966e0e5e411f8d8b0cc6c26fefd1
| poweruser        |
| b59703369e194123b5c77dad60d11a25 | member          | 382761de4a9c4414b6f8950f8580897c
| reader           |
```

Если неявная ассоциация выполнена по ошибке, изменения можно отменить:

```
$ openstack implied role delete manager --implied-role poweruser
```

2 Управление группами

Администратор может использовать группы (groups) сервиса Identity (keystone) для назначения одинаковых разрешений большому количеству пользовательских аккаунтов.

2.1 Настройка групп через интерфейс командной строки (CLI)

Создайте группу и назначьте разрешения этой группе. Участники группы наследуют разрешения, назначенные группе:

1. Создайте группу grp-Auditors:

```
$ openstack group create grp-Auditors
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | default                          |
| id          | 2a4856fc242142a4aa7c02d28edfdfff |
| name        | grp-Auditors                     |
+-------------+----------------------------------+
```

2. Просмотрите список групп keystone:

```
$ openstack group list --long
+----------------------------------+--------------+-----------+-------------+
| ID                               | Name         | Domain ID | Description |
+----------------------------------+--------------+-----------+-------------+
| 2a4856fc242142a4aa7c02d28edfdfff | grp-Auditors | default   |             |
+----------------------------------+--------------+-----------+-------------+
```

3. Предоставьте группе grp-Auditors разрешение на доступ к демопроекту, используя роль участника:

```
$ openstack role add member --group grp-Auditors --project demo
```

4. Добавьте существующего пользователя user1 в группу grp-Auditors:

```
$ openstack group add user grp-Auditors user1
```

Пользователь user1 добавлен в группу grp-Auditors.

5. Подтвердите, что пользователь user1 является участником группы grp-Auditors:

```
$ openstack group contains user grp-Auditors user1
```

Пользователь user1 находится в группе grp-Auditors.

6. Проверьте действующие разрешения, предоставленные пользователю user1:

```
$ openstack role assignment list --effective --user user1
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
| Role                             | User                             | Group | Project
| Domain | Inherited |
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
| 9fe2ff9ee4384b1894a90878d3e92bab | 3fefe5b4f6c948e6959d1feaef4822f2 |       |
0ce36252e2fb4ea8983bed2a568fa832 |        | False     |
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
```

2.2 Настройка групп при помощи панели управления (Dashboard)

Панель управления (Dashboard) может использоваться для управления участием в группах (group membership) keystone. При этом для назначения ролей в группе необходимо использовать командную строку.

2.2.1Создание группы

1. Войдите в панель управления как пользователь с привилегиями администратора.
2. Выберите **Identity** > **Groups**.
3. Нажмите на **+Create Group**.
4. Введите имя и описание группы.
5. Нажмите на **Create Group**.

2.2.2 Управление участием в группах (group membership)

Панель управления можно использовать для управления участием в группах keystone:

1. Войдите в панель управления как пользователь с привилегиями администратора.
2. Выберите **Identity** > **Groups**.
3. Нажмите на **Manage Members** для группы, в которую требуется внести изменения.
4. Добавьте пользователя в группу при помощи **Add users**. Если нужно удалить пользователя, отметьте соответствующее поле и нажмите на **Remove users**.

3 Управление квотами

Администратор облака может устанавливать и управлять квотами (quota) проекта.

Для каждого проекта выделяются ресурсы, а пользователям проекта предоставляется доступ к их использованию. Это позволяет использовать одно облако для нескольких проектов, при этом их разрешения и ресурсы не будут мешать друг другу.

Набор квот ресурсов настраивается предварительно при создании нового проекта. Квоты включают определенное количество виртуальных процессоров (vCPU), инстансов, оперативной памяти и плавающих IP-адресов, которые могут быть назначены проектам.

Квоты могут применяться на уровне проекта, а также на уровне пользователя проекта. С помощью панели управления администратор может устанавливать или изменять вычислительные квоты (Compute) и квоты хранилищ (Block Storage) для новых и существующих проектов.

3.1 Просмотр квот вычислительных ресурсов (Compute) для пользователя

Порядок действий

- Запустите следующую команду, чтобы указать установленные на данный момент значения квоты для пользователя:

```
$ nova quota-show --user [USER] --tenant [TENANT]
```

Пример

```
$ nova quota-show --user demoUser --tenant demo
+-----------------------------+-------+
| Quota                       | Limit |
+-----------------------------+-------+
| instances                   | 10    |
| cores                       | 20    |
| ram                         | 51200 |
| floating_ips                | 5     |
| fixed_ips                   | -1    |
| metadata_items              | 128   |
| injected_files              | 5     |
| injected_file_content_bytes | 10240 |
| injected_file_path_bytes    | 255   |
| key_pairs                   | 100   |
| security_groups             | 10    |
| security_group_rules        | 20    |
| server_groups               | 10    |
| server_group_members        | 10    |
+-----------------------------+-------+
```

3.2 Обновление квот вычислительных ресурсов для пользователя

Порядок действий

- Запустите следующие команды, чтобы обновить определенное значение квоты:

```
$ nova quota-update --user [USER] --[QUOTA_NAME] [QUOTA_VALUE] [TENANT]
$ nova quota-show --user [USER] --tenant [TENANT]
```

Пример

```
$ nova quota-update --user demoUser --floating-ips 10 demo
$ nova quota-show --user demoUser --tenant demo
+------------------------------+-------+
| Quota                        | Limit |
+------------------------------+-------+
| instances                    | 10    |
| cores                        | 20    |
| ram                          | 51200 |
| floating_ips                 | 10    |
| ...                          |       |
+------------------------------+-------+
```

4 Управление проектами

Администратор облака может создавать проекты и управлять ими. Проект (project) – это пул виртуальных совместных ресурсов, где можно назначать пользователей и группы 1Stack.

Для каждого проекта можно настраивать квоту общих виртуальных ресурсов. 1Stack позволяет создавать несколько проектов таким образом, что их разрешения и ресурсы не будут мешать друг другу.

Пользователи могут быть связаны более чем с одним проектом. Каждому пользователю должна быть назначена роль для каждого проекта, в который он добавлен.

4.1 Создание проекта

Создайте проект, добавьте в него пользователей и установите ограничения по ресурсам для этого проекта:

1. Войдите в панель управления как пользователь с привилегиями администратора.
2. Выберите **Identity** > **Projects**.
3. Нажмите Create Project.
4. На вкладке **Project Information** введите название и описание проекта. По умолчанию будет проставлен флажок **Enabled**.
5. На вкладке **Project Members** добавьте участников в проект из списка **All Users**.
6. На вкладке **Quotas** укажите ограничения по ресурсам для проекта.
7. Нажмите на **Create Project**.

4.2 Внесение изменений в проект

Проект можно редактировать, менять его название или описание, активировать или временно деактивировать его, а также обновлять список участников проекта:

1. Войдите в панель управления как пользователь с привилегиями администратора.
2. Выберите **Identity** > **Projects**.
3. В колонке проекта **Actions** нажмите на стрелку, затем на **Edit Project**.
4. В окне **Edit Project** внесите обновления в проект: измените его название или описание, активируйте или временно деактивируйте проект.
5. На вкладке **Project Members** добавьте участников в проект или при необходимости удалите их.
6. Нажмите **Save**.

Кнопка-флажок **Enabled** будет выбрана по умолчанию.
Для временной дективации проекта снимите флажок **Enabled**.
Для активации неактивного проекта проставьте флажок **Enabled**.

4.3 Удаление проекта

Выполните следующие действия:

1. Войдите в панель управления как пользователь с административными привилегиями.
2. Выберите **Identity** > **Projects**.
3. Выберите проект, который нужно удалить.
4. Нажмите на **Delete Projects**. При этом откроется окно **Confirm Delete Projects**.
5. Нажмите на **Delete Projects** для подтвержения действия.

Проект будет удален, а любые пользователи, связанные с ним, отсоединены.

4.4 Обновление квот проекта

Квоты (quotas) – это ограничения ресурсов, установленные по каждому проекту для оптимизации облачных ресурсов.

Можно установить квоты, если нужно предотвратить исчерпание ресурсов проекта без уведомления.

Квоты можно применять на уровне проекта, а также на уровне пользователей проекта.

Выполните следующие действия:

1. Войдите в панель управления как пользователь с привилегиями администратора.
2. Выберите **Identity** > **Projects**.
3. В колонке проекта **Actions** нажмите на стрелку, затем на **Modify Quotas**.
4. На вкладке **Quota** измените квоты проекта, как требуется.
5. Нажмите на **Save**.

На данный момент поддержка квот вложенных проектов пока не поддерживается, поэтому необходимо управлять квотами индивидуально по проектам и подпроектам.

4.5 Внесение изменений в активный проект

Установите проект как активный, чтобы использовать панель управления для взаимодействия с объектами в проекте.

Чтобы установить проект как активный, администратор должен быть участником проекта. Кроме того, пользователь должен быть участником более одного проекта, чтобы активировать опцию **Set as Active Project**.

Нельзя установить отключенный проект как активный, пока он не будет включен повторно.

Выполните следующие действия:

1. Войдите в панель управления как пользователь с административными привилегиями.
2. Выберите **Identity** > **Projects**.
3. В колонке проекта **Actions** нажмите на стрелку, затем на **Set as Active Project**.
4. Альтернативно, как пользователь без прав администратора, в столбце **Actions** нажмите **Set as Active Project**.

4.6 Иерархии проекта

Администратор может включать проекты в другие проекты при помощи многоарендной архитектуры (multitenancy) в сервисе Identity (keystone). Многоарендность позволяет подпроектам наследовать назначенные роли из родительского проекта.

4.6.1Создание иерархических проектов и подпроектов

Администратор может реализовать иерархическую многопользовательскую среду (Hierarchical Multitenancy, HMT), используя домены и проекты сервиса Keystone.

Создайте новый домен, а затем проект в рамках этого домена. Далее в проект также можно добавить подпроекты. Можно также «повысить» пользователя до администратора подпроекта, добавив ему роль администратора в этом подпроекте.

> Структура Hierarchical Multitenancy, используемая сервисом Keystone, в настоящее время не представлена на панели управления.

Порядок действий

1. Создайте новый домен с названием corp:

```
$ openstack domain create corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | True                             |
| id          | 69436408fdcb44ab9e111691f8e9216d |
| name        | corp                             |
+-------------+----------------------------------+
```

2. Создайте родительский проект private-cloud в рамках домена corp:

```
$ openstack project create private-cloud --domain corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | 69436408fdcb44ab9e111691f8e9216d |
| enabled     | True                             |
| id          | c50d5cf4fe2e4929b98af5abdec3fd64 |
| is_domain   | False                            |
| name        | private-cloud                    |
| parent_id   | 69436408fdcb44ab9e111691f8e9216d |
+-------------+----------------------------------+
```

3. Создайте подпроект dev в родительском проекте private-cloud и укажите домен corp:

```
$ openstack project create dev --parent private-cloud --domain corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | 69436408fdcb44ab9e111691f8e9216d |
| enabled     | True                             |
| id          | 11fccd8369824baa9fc87cf01023fd87 |
```

```
| is_domain   | False                            |
| name        | dev                              |
| parent_id   | c50d5cf4fe2e4929b98af5abdec3fd64 |
+-------------+----------------------------------+
```

4. 4 Создайте еще один подпроект с названием qa:

```
$ openstack project create qa --parent private-cloud --domain corp
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | 69436408fdcb44ab9e111691f8e9216d |
| enabled     | True                             |
| id          | b4f1d6f59ddf413fa040f062a0234871 |
| is_domain   | False                            |
| name        | qa                               |
| parent_id   | c50d5cf4fe2e4929b98af5abdec3fd64 |
+-------------+----------------------------------+
```

4.6.2 Настройка доступа к иерархическим проектам

По умолчанию в созданном новом проекте нет назначенных ролей.

При назначении разрешений ролям в родительском проекте можно включить флажок --inherited. В этом случае подпроекты будут наследовать назначенные разрешения от родительского проекта. Например, пользователь с ролью администратора в родительском проекте также получит доступ администратора к подпроектам.

Предоставление доступа пользователям:

1. Просмотрите текущие разрешения, назначенные проекту:

```
$ openstack role assignment list --project private-cloud
```

2. Просмотрите существующие роли:

```
$ openstack role list
+----------------------------------+------------------+
| ID                               | Name             |
+----------------------------------+------------------+
| 01d92614cd224a589bdf3b171afc5488 | admin            |
| 034e4620ed3d45969dfe8992af001514 | member           |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin    |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user  |
| cfea5760d9c948e7b362abc1d06e557f | reader           |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator    |
| ef3d3f510a474d6c860b4098ad658a29 | service          |
+----------------------------------+------------------+
```

3. Предоставьте пользовательскому аккаунту user1 доступ к проекту private-cloud:

```
$ openstack role add --user user1 --user-domain corp --project private-cloud member
```

4. Повторно выполните эту команду, используя флажок **--inherited**. В результате пользователь user1 также получит доступ к подпроектам **private-cloud**, которые унаследовали назначенную роль:

```
$ openstack role add --user user1 --user-domain corp --project private-cloud member --
inherited
```

5. Проверьте результат обновления разрешений:

```
$ openstack role assignment list --effective --user user1 --user-domain corp
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
| Role                             | User                             | Group | Project
| Domain | Inherited |
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
c50d5cf4fe2e4929b98af5abdec3fd64 |        | False     |
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
11fccd8369824baa9fc87cf01023fd87 |        | True      |
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
b4f1d6f59ddf413fa040f062a0234871 |        | True      |
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
```

Пользователь user1 унаследовал доступ к проектам **qa** и **dev**. Кроме того, к родительскому проекту был применен флажок **--inherited**, поэтому пользователь user1 также получит доступ ко всем подпроектам, которые будут созданы позже.

4.6.3 Удаление доступа пользователей

Явные и унаследованные разрешения необходимо удалять раздельно:

1. Удалите пользователя из явно назначенной роли:

```
$ openstack role remove --user user1 --project private-cloud member
```

2. Проверьте результат изменений. Обратите внимание, что унаследованные разрешения по-прежнему сохраняются:

```
$ openstack role assignment list --effective --user user1 --user-domain corp
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
| Role                             | User                             | Group | Project
| Domain | Inherited |
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
11fccd8369824baa9fc87cf01023fd87 |        | True      |
| 034e4620ed3d45969dfe8992af001514 | 10b5b34df21d485ca044433818d134be |       |
b4f1d6f59ddf413fa040f062a0234871 |        | True      |
+----------------------------------+----------------------------------+-------+--------
--------------------------+--------+-----------+
```

3. Удалите унаследованные разрешения:

```
$ openstack role remove --user user1 --project private-cloud member –inherited
```

4. Проверьте результат изменений. Унаследованные разрешения были удалены:

```
$ openstack role assignment list --effective --user user1 --user-domain corp
```

4.7 Обзор проекта Reseller

Цель проекта Reseller – создание иерархии доменов. Домены позволят рассмотреть возможность перепродажи частей облака, при этом поддомен представляет собой полностью готовое облако. Эта работа была разделена на этапы, описание этапа 1 приведено далее.

Этап 1 проекта Reseller

Reseller (этап 1) – это расширение иерархической многоарендности (hierarchical multitenancy, HMT).

Раньше домены сервиса Keystone изначально предназначались для использования в качестве контейнеров для хранения пользователей и проектов, с собственной таблицей в базе данных. В результате – домены более не хранятся в собственной таблице и объединены с таблицей проекта:

- Домен теперь является типом проекта, который отличается флажком **is_domain**.
- Домен представляет собой проект верхнего уровня в иерархии проектов: домены являются корнем в иерархии проектов.
- Программные интерфейсы (API) обновлены для создания и загрузки доменов при помощи свойств проектов.

Выполните следующие действия:

1. Создайте новый домен – создайте проект с флажком **is_domain**, установленным в **true**.
2. Создайте список проектов-доменов: получите данные проектов, включающие параметр запроса **is_domain**.

Этап 1 не позволяет создавать иерархию доменов. Это означает, что поддомены еще недоступны. Кроме того, это не меняет сферу применения токенов и не включает в себя поддержку иерархии, необходимую для других проектов (не keystone).

4.8 Управление безопасностью проекта

Группы безопасности (security groups) – это наборы правил IP-фильтрации, которые можно назначить инстансам проекта и которые определяют сетевой доступ к инстансу.

Группы безопасности зависят от проекта. Участники проекта могут редактировать правила по умолчанию для своей группы безопасности и добавлять новые наборы правил.

Все проекты имеют группу безопасности по умолчанию, которая применяется к любому инстансу, не имеющему другой определенной группы безопасности. Если не менять значения

по умолчанию, эта группа безопасности будет блокировать весь входящий трафик и пропускать только исходящий трафик с инстанса.

Не удаляйте группу безопасности по умолчанию, не создав группы, разрешающие необходимый исходящий трафик. Например, если инстансы используют DHCP и метаданные, то инстансу необходимы правила группы безопасности, разрешающие исходящий трафик к DHCP-серверу и агенту метаданных.

4.8.1Создание группы безопасности посредством веб-интерфейса

Создайте группу безопасности для настройки правил безопасности. Например, можно включить трафик ICMP или отключить HTTP-запросы.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Access & Security**.
2. На вкладке **Security Groups** нажмите на **Create Security Group**.
3. Введите имя и описание группы и нажмите на **Create Security Group**.

4.8.2 Добавление правила группы безопасности

По умолчанию правила (security group rule) для новой группы предоставляют только исходящий доступ. Чтобы предоставить дополнительный доступ, необходимо добавить новые правила.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Access & Security**.
2. На вкладке **Security Groups** нажмите на **Manage Rules** той группы, в которую нужно внести изменения.
3. Нажмите на **Add Rule**, чтобы добавить новое правило.
4. Укажите значения правила и нажмите **Add**.

Следующие поля правил являются обязательными:

Rule

Rule type. Если указать шаблон правила (например, SSH), его поля будут заполняться автоматически:

- TCP: Как правило, используется для обмена данными между системами и конечным пользователем.
- UDP: Обычно используется для обмена данными между системами, особенно на уровне приложений.
- ICMP: Обычно используется сетевыми устройствами, например, маршрутизаторами, для отправки сообщений об ошибках или мониторинге.

Direction

Ingress (входящий) или **Egress** (исходящий).

Open Port

Для правил TCP или UDP, **Port** или **Port Range** (единичный порт или диапазон портов) для открытия:

- Для диапазона портов введите значение порта в поле **From Port** или **To Port**.
- Для одного порта введите значение порта в поле **Port**.

Type

Тип правил ICMP строго в диапазоне: 1:255.

Code

Код для правил ICMP строго в диапазоне: 1:255'.

Remote

Источник трафика для этого правила:

- CIDR (бесклассовая междоменная маршрутизация): блок IP-адресов, который ограничивает доступ к IP-адресам внутри блока. Введите CIDR в поле Source.
- Группа безопасности: исходная группа, которая позволяет любому инстансу группы получать доступ к любому другому инстансу группы.

4.8.3 Удаление правила группы безопасности

Ненужные правила группы безопасности можно удалить.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Access & Security**.
2. На вкладке **Security Groups** нажмите **Manage Rules** для группы безопасности.
3. Выберите правило группы безопасности и нажмите **Delete Rule**.
4. Еще раз нажмите **Delete Rule**.

Действие удаления отменить нельзя.

4.8.4 Удаление группы безпасности

Ненужные группы безопасности можно удалить.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Access & Security**.
2. На вкладке **Security Groups** выберите группу и нажмите **Delete Security Groups**.
3. Нажмите **Delete Security Groups**.

Действие удаления отменить нельзя.

5 Управление доменами

Домены (domain) сервиса Identity – это дополнительные пространства имен, которые можно создать в keystone.

Используйте домены keystone для разделения пользователей, групп и проектов. Также можно настроить эти отдельные домены для авторизации пользователей в различных средах LDAP или Active Directory.

> Сервис Identity включает встроенный домен с названием Default. Рекомендуется зарезервировать этот домен только для учетных записей сервисов, а для учетных записей пользователей создать отдельный домен.

5.1 Просмотр списка доменов

Список доменов можно просмотреть при помощи команды openstack domain list:

```
$ openstack domain list
+----------------------------------+------------------+---------+--------------------+
| ID                               | Name             | Enabled | Description        |
+----------------------------------+------------------+---------+--------------------+
| 3abefa6f32c14db9a9703bf5ce6863e1 | TestDomain       | True    |                    |
| 69436408fdcb44ab9e111691f8e9216d | corp             | True    |                    |
| a4f61a8feb8d4253b260054c6aa41adb | federated_domain | True    |                    |
| default                          | Default          | True    | The default domain |
+----------------------------------+------------------+---------+--------------------+
```

5.2 Просмотр информации о домене

Информацию о домене можно просмотреть при помощи команды openstack domain show:

```
$ openstack domain show TestDomain
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| enabled     | True                             |
| id          | 3abefa6f32c14db9a9703bf5ce6863e1 |
| name        | TestDomain                       |
+-------------+----------------------------------+
```

6 Учетные данные приложения

Учетные данные приложения (Application Credentials) позволяют исключить использование учетных данных пользователя в файлах конфигурации. Вместо этого пользователь создает учетные данные приложения, которые получают делегированный доступ к одному проекту и имеют свой собственный секретный ключ.

Пользователь также может ограничить делегирование привилегии одной ролью в этом проекте. Это позволяет придерживаться принципа наименьших привилегий, когда авторизованный сервис получает доступ только к одному проекту и роли, которые необходимы для его работы, а не ко всем проектам и ролям.

Можно использовать эту методологию для использования API, не раскрывая пользовательские учетные данные, а приложения могут проходить авторизацию в сервисе Keystone, не требуя встроенных учетных данных пользователя.

Учетные данные приложения можно использовать для создания токенов и настройки параметров keystone_authtoken для приложений. Эти варианты применения описаны в следующих разделах.

> Учетные данные приложения зависят от акаунта пользователя, создавшего их, и прекращают свое действие, если эта учетная запись будет удалена или потеряет доступ к соответствующей роли.

6.1 Создание токенов при помощи учетных данных приложения

Учетные данные приложения доступны пользователям как функция самообслуживания на панели управления. Далее показано, как пользователь может создать учетные данные приложения и использовать их для создания токена (token).

1. Создайте тестовый проект и тестовые пользовательские аккаунты:
   а) Создайте проект с названием AppCreds:

```
$ openstack project create AppCreds
```

   б) Создайте пользователя с именем AppCredsUser:

```
$ openstack user create --project AppCreds --password-prompt AppCredsUser
```

   в) Предоставьте пользователю AppCredsUser доступ к роли участника проекта AppCreds:

```
$ openstack role add --user AppCredsUser --project AppCreds member
```

2. Зайдите в панель управления как AppCredsUser и создайте учетные данные приложения:

Overview → Identity → Application Credentials → +Create Application Credential.

Убедитесь, что загружено содержимое файла clouds.yaml, поскольку к нему невозможно будет получить доступ повторно после закрытия всплывающего окна с названием Your Application Credential.

3. Создайте файл с названием /home/stack/.config/openstack/clouds.yaml при помощи интерфейса командной строки (CLI) и введите содержимое файла clouds.yaml:

```
# This is a clouds.yaml file, which can be used by OpenStack tools as a source
# of configuration on how to connect to a cloud. If this is your only cloud,
# just put this file in ~/.config/openstack/clouds.yaml and tools like
# python-openstackclient will just work with no further config. (You will need
# to add your password to the auth section)
# If you have more than one cloud account, add the cloud entry to the clouds
# section of your existing file and you can refer to them by name with
# OS_CLOUD=openstack or --os-cloud=openstack
clouds:
  openstack:
    auth:
      auth_url: http://10.0.0.10:5000/v3
      application_credential_id: "6d141f23732b498e99db8186136c611b"
      application_credential_secret: "<example secret value>"
    region_name: "regionOne"
    interface: "public"
    identity_api_version: 3
    auth_type: "v3applicationcredential"
```

Приведённые значения являются примером.

4. Используйте учетные данные приложения для создания токена. Пользователь, применяющий следующую команду, не должен быть зарегистрирован как определенный пользователь и должен находиться в той же директории, что и его файл clouds.yaml:

```
[stack@undercloud-0 openstack]$ openstack --os-cloud=openstack token issue
+------------+------------------------------------------------------------------------
------------------------------------------------------------+
| Field      | Value
|
+------------+------------------------------------------------------------------------
------------------------------------------------------------+
| expires    | 2018-08-29T05:37:29+0000
|
| id         | gAAAAABbhiMJ4TxxFlTMdsYJpfStsGotPrns0lnpvJq9ILtdi-
NKqisWBeNiJlUXwmnoGQDh2CMyK9OeTsuEXnJNmFfKjxiHWmcQVYzAhMKo6_QMUtu_Qm6mtpzYYHBrUGboa_
|            | Ay0LBuFDtsjtgtvJ-r8G3TsJMowbKF-yo--O_XLhERU_QQVl3hl8zmMRdmLh_P9Cbhuolt
|
| project_id | 1a74eabbf05c41baadd716179bb9e1da
|
| user_id    | ef679eeddfd14f8b86becfd7e1dc84f2
|
+------------+------------------------------------------------------------------------
------------------------------------------------------------+
```

Если отображается ошибка, похожая на *init() got an unexpected keyword argument 'application_credential_secret'*, возможно, по-прежнему используются предыдущие учетные данные. Чтобы создать новую среду, выполните команду sudo su - stack.

6.2 Интеграция учетных данных приложения с приложениями

Учетные данные приложения (Application Credentials) можно использовать для авторизации приложений в keystone.

При применении учетных данных приложения настройки keystone_authtoken используют v3applicationcredential в качестве типа аутентификации и содержат учетные данные, которые пользователь получает при создании учетных данных.

Введите следующие значения:

1. **application_credential_secret**: секретный ключ учетных данных.
2. **application_credential_id**: ID учетных данных приложения.
3. (Опционально) **application_credential_name**: этот параметр можно применять, если используется учетная запись приложения с названием, а не с идентификатором.

Например:

```
[keystone_authtoken]
auth_url = http://10.0.0.10:5000/v3
auth_type = v3applicationcredential
application_credential_id = "6cb5fa6a13184e6fab65ba2108adf50c"
application_credential_secret = "<example password>"
```

6.3 Управление учетными данными приложения

Для создания и удаления учетных данных приложения можно использовать командную строку.

Подкоманда create создает учетные данные приложения на основе учетной записи. Например, при создании учетной записи от имени пользователя-администратора учетным данным будут предоставлены те же роли:

```
$ openstack application credential create --description "App Creds - All roles"
AppCredsUser
+--------------+-------------------------------------------------------------------------
-----------------+
| Field        | Value
|
+--------------+-------------------------------------------------------------------------
-----------------+
| description  | App Creds - All roles
|
| expires_at   | None
|
| id           | fc17651c2c114fd6813f86fdbb430053
|
| name         | AppCredsUser
|
| project_id   | 507663d0cfe244f8bc0694e6ed54d886
|
| roles        | member reader admin
|
| secret       | fVnqa6I_XeRDDkmQnB5lx361W1jHtOtw3ci_mf_tOID-09MrPAzkU7mv-
by8ykEhEa1QLPFJLNV4cS2Roo9lOg |
| unrestricted | False
|
```

Использование параметра **--unrestricted** позволяет учетным данным приложения создавать и удалять другие учетные данные приложения и делегировать права. Это потенциально опасное поведение, оно деактивировано по умолчанию. Нельзя использовать параметр **--unrestricted** в сочетании с другими правилами доступа.

По умолчанию итоговое ролевое членство включает все роли, назначенные учетной записи, с которой созданы учетные данные. Можно ограничить состав ролей, делегировав доступ только определенной роли:

```
$ openstack application credential create --description "App Creds - Member" --role
member AppCredsUser
+--------------+-----------------------------------------------------------------------
-----------------+
| Field        | Value
|
+--------------+-----------------------------------------------------------------------
-----------------+
| description  | App Creds - Member
|
| expires_at   | None
|
| id           | e21e7f4b578240f79814085a169c9a44
|
| name         | AppCredsUser
|
| project_id   | 507663d0cfe244f8bc0694e6ed54d886
|
| roles        | member
|
| secret       |
XCLVUTYIreFhpMqLVB5XXovs_z9JdoZWpdwrkaG1qi5GQcmBMUFG7cN2htzMlFe5T5mdPsnf5JMNbu0Ih-4aCg
|
| unrestricted | False
|
+--------------+-----------------------------------------------------------------------
-----------------+
```

Чтобы удалить учетные данные приложения, укажите:

```
$ openstack application credential delete AppCredsUser
```

6.4 Замена учетных данных приложения

Учетные данные приложения привязаны к создавшей их учетной записи пользователя и становятся недействительными при удалении этой учетной записи или потере доступа пользователя к делегированной роли. Поэтому, возможно, понадобится создать новые учетные данные приложения.

Замена существующих учетных данных файлами конфигурации

Обновите учетные данные, назначенные приложению (при помощи файла конфигурации):

1. Создайте новый набор учетных данных.

2. Добавьте новые учетные данные в файл конфигурации приложения, заменив существующие.
3. Перезапустите сервис приложения, чтобы применить изменения.
4. Если необходимо, удалите старую учетную запись приложения.

Замена существующих учетных данных приложения файлами clouds.yaml

Замените существующие учетные данные, которые использует файл clouds.yaml:

Например, если clouds.yaml содержит учетную запись **AppCred1** с истекающим сроком действия:

1. Создайте учетные данные с названием **AppCred2**.
2. Добавьте новый **AppCred2** в файл clouds.yaml, удалив конфигурацию **AppCred1**.
3. Сгенерируйте токен с помощью clouds.yaml, чтобы убедиться, что учетные данные работают ожидаемым образом.

7 Создание инстанса

Для создания инстанса (виртуальной машины, instance) необходима доступность других компонентов 1Stack, таких как шаблон инстанса, источник загрузки образа, сеть, пара ключей и группа безопасности. Эти компоненты используются при создании инстанса и недоступны по умолчанию.

При создании инстанса пользователь выбирает:

- источник загрузки с загрузочной операционной системой, необходимой для инстанса;
- шаблон с профилем оборудования, который необходим для инстанса;
- сеть, к которой нужно подключить инстанс;
- любое нужное дополнительное хранилище, например, тома данных (volume) и временное хранилище (ephemeral).

При создании инстанса сервис Compute (nova) использует имя, которое присвоено инстансу, для создания действительного имени хоста для инстанса в сервисе метаданных и на диске конфигурации. Сервис Compute меняет пробелы (' ') и подчеркивания (_) в имени инстанса на дефис (-) для создания корректной метки DNS. Сервис Compute также меняет точки (.) на дефис (-), если имя инстанса заканчивается точкой, за которой следует одна или несколько цифр.

Требования

1. Нужный образ инстанса (image) или том (volume) доступны в качестве источника загрузки.
2. Доступен шаблон, где указано необходимое количество процессоров, память и емкость хранилища. Настройки шаблона должны соответствовать минимальным требованиям к размеру диска и памяти, указанным выбранным вами образом, в противном случае запуск инстанса будет неуспешным.
3. Доступна требуемая сеть.

7.1 Создание инстанса из образа

Пользователь может создать инстанс, используя образ инстанса (image) в качестве источника загрузки.

Порядок действий

1. Получите имя или ID шаблона с профилем оборудования, который необходим для инстанса:

```
$ openstack flavor list
```

Выберите шаблон достаточного размера для загрузки образа, в противном случае загрузка инстанса будет неуспешной.

2. Получите имя или ID образа, который имеет профиль ПО, необходимый для инстанса:

```
$ openstack image list
```

Если нужный образ недоступен, можно загрузить или создать новый образ:

3. Получите имя или ID сети, к которой нужно подключить инстанс:

```
$ openstack network list
```

4. Создайте инстанс:

```
$ openstack server create --flavor <flavor> --image <image> --network <network> --wait
myInstanceFromImage
```

- Поменяйте <flavor> на имя или ID шаблона, которые были получены при выполнении шага 1.
- Поменяйте <image> на имя или ID образа, которые были получены при выполнении шага 2.
- Поменяйте <network> на имя или ID сети, которые были получены при выполнении шага 3.

Опцию **--network** можно использовать несколько раз, если нужно подключить инстанс к нескольким сетям.

7.2 Создание инстанса из загрузочного тома

Пользователь может создать инстанс, используя загрузочный том (bootable volume) в качестве источника загрузки.

Загрузите инстанс с загрузочного тома, если нужно повысить доступность данных инстанса при сбое.

> При использовании тома блочного хранения для данных диска инстанса, том блочного хранения остается неизменным при любых пересборках инстанса, в том числе при использовании нового образа, который требует создания нового диска.

Порядок действий

1. Получите имя или ID образа с профилем ПО, необходимым для инстанса:

```
$ openstack image list
```

Если нужный образ недоступен, можно загрузить или создать новый образ.

2. Создайте загрузочный том из образа:

```
$ openstack volume create --image <image> --size <size_gb> --bootable myBootableVolume
```

- Поменяйте <image> на имя или ID образа для записи на том, полученные при выполнении шага 1.
- Поменяйте <size_gb> на размер диска в ГБ.

3. Получите имя или ID шаблона с профилем оборудования, который необходим для инстанса:

```
$ openstack flavor list
```

4. Получите имя или ID сети, к которой нужно подключить инстанс:

```
$ openstack network list
```

5. Создайте инстанс с загрузочным томом:

```
$ openstack server create --flavor <flavor> --volume myBootableVolume --network
<network> --wait myInstanceFromVolume
```

- Поменяйте <flavor> на имя или ID шаблона, который был получен при выполнении шага 3.
- Поменяйте <network> на имя или ID сети, который был получен при выполнении шага 4.

Опцию **--network** можно использовать несколько раз, если нужно подключить инстанс к нескольким сетям.

7.3 Создание инстанса с сетевым интерфейсом SR-IOV

Для создания инстанса с единым корневым сетевым интерфейсом виртуализации ввода-вывода (SR-IOV) нужно создать необходимый порт SR-IOV.

Порядок действий

1. Получите имя или ID шаблона с профилем аппаратного оборудования, который необходим для инстанса:

```
$ openstack flavor list
```

Для успешной загрузки образа выберите шаблон достаточного размера, в противном случае произойдет сбой загрузки.

Можно указать политику привязки (affinity) NUMA, которая будет применяться к инстансу для транзитных устройств PCI и интерфейсов SR-IOV – для этого выбрать шаблон с нужной политикой. Если выбрать шаблон с политикой привязки (affinity) NUMA, то используемый образ должен иметь либо ту же политику сходства NUMA, либо не иметь политики сходства NUMA.

2. Получите имя или ID образа с профилем ПО, необходимым для инстанса:

```
$ openstack image list
```

Если нужный образ недоступен, можно загрузить или создать новый образ.

Можно указать политику привязки (affinity) NUMA, которая будет применяться к инстансу для транзитных устройств PCI и интерфейсов SR-IOV – для этого выбрать шаблон с нужной политикой. Если выбрать шаблон с политикой привязки (affinity) NUMA, то используемый образ должен иметь либо ту же политику сходства NUMA, либо не иметь политики сходства NUMA.

3. Получите имя или ID сети, к которой нужно подключить инстанс:

```
$ openstack network list
```

4. Создайте тип порта, необходимый для интерфейса SR-IOV:

```
$ openstack port create --network <network> --vnic-type <vnic_type> mySriovPort
```

- Поменяйте <network> на имя или ID сети, полученные при выполнении шага 3.
- Поменяйте <vnic_type> на одно из значений:
  - direct: Создает порт виртуальной функции SR-IOV (VF);
  - direct-physical: Создает порт физической функции (PF) SR-IOV.

5. Создайте инстанс:

```
$ openstack server create --flavor <flavor> --image <image> --port <port> --wait
mySriovInstance
```

- Поменяйте <flavor> на имя или ID шаблона, полученные при выполнении шага 1.
- Поменяйте <image> на имя или ID образа, полученные при выполнении шага 2.
- Поменяйте <port> на имя или ID порта, созданного при выполнении шага 4.

8 Создание инстанса с гарантированной минимальной пропускной способностью QoS

Пользователь может создавать инстансы, для которых необходима гарантированная минимальная пропускная способность, используя политику качества обслуживания (Quality of service, QoS).

Политики QoS с правилом гарантированной минимальной пропускной способности назначаются портам в конкретной физической сети.

При создании инстанса, использующего настроенный порт, сервис планирования Compute выбирает для инстанса хост, который удовлетворяет такому запросу.

Перед выбором хоста для развертывания инстанса сервис Compute проверяет сервис Placement на наличие объема полосы пропускания, зарезервированной другими инстансами на каждом физическом интерфейсе.

Ограничения

- Политику QoS с гарантированной минимальной пропускной способностью можно назначить только при создании нового инстанса. Это нельзя сделать для уже запущенных инстансов, поскольку сервис вычислительных ресурсов обновляет информацию об использовании ресурсов для размещенного инстанса только при операции создания или перемещения. Это означает, что минимальная пропускная способность, доступная для инстанса, не может быть гарантирована.
- Нельзя выполнить «живую» миграцию, выгрузить инстанс на диск (unshelve) или переместить с сервера инстанс (evacuate), использующий порт с запросами на ресурсы, например политику QoS с гарантированной минимальной пропускной способностью. Запустите следующую команду, чтобы проверить наличие запросов на ресурсы у порта:

```
$ openstack port show <port_name/port_id>
```

Предварительные требования

- Доступна QoS-политика с правилом минимальной пропускной способности.

Порядок действий

1. Получите список доступных политик QoS:

```
(overcloud)$ openstack network qos policy list
+--------------------------------------+---------+--------+---------+---------+
| ID                                   | Name    | Shared | Default | Project |
+--------------------------------------+---------+--------+---------+---------+
| 6d771447-3cf4-4ef1-b613-945e990fa59f | policy2 | True   | False   |
ba4de51bf7694228a350dd22b7a3dc24 |
| 78a24462-e3c1-4e66-a042-71131a7daed5 | policy1 | True   | False   |
ba4de51bf7694228a350dd22b7a3dc24 |
| b80acc64-4fc2-41f2-a346-520d7cfe0e2b | policy0 | True   | False   |
ba4de51bf7694228a350dd22b7a3dc24 |
+--------------------------------------+---------+--------+---------+---------+
```

2. Проверьте правила каждой доступной политики и определите, какая из них имеет необходимую минимальную пропускную способность:

```
(overcloud)$ openstack network qos policy show policy0
+-------------------+--------------------------------------------------------------------------------------+
| Field             | Value                                                                                |
+-------------------+--------------------------------------------------------------------------------------+
| description       |                                                                                      |
| id                | b80acc64-4fc2-41f2-a346-520d7cfe0e2b                                                 |
| is_default        | False                                                                                |
| location          | cloud='', project.domain_id=, project.domain_name='Default',                         |
                    | project.id=ba4de51bf7694228a350dd22b7a3dc24, project.name=admin, region_name=regionOne |
                    | , zone=                                                                              |
| name              | policy0                                                                              |
| project_id        | ba4de51bf7694228a350dd22b7a3dc24                                                     |
| rules             | [{min_kbps: 100000, direction: egress, id: d46218fe-9218-4e96-952b-                  |
                    | 9f45a5cb3b3c, qos_policy_id: b80acc64-4fc2-41f2-a346-520d7cfe0e2b, type:              |
                    | minimum_bandwidth},                                                                  |
                    | {min_kbps: 100000, direction: ingress, id: 1202c4e3-a03a-464c-80d5-                  |
                    | 0bf90bb74c9d, qos_policy_id: b80acc64-4fc2-41f2-a346-520d7cfe0e2b, type:              |
                    | minimum_bandwidth}]                                                                  |
| shared            | True                                                                                 |
| tags              | []                                                                                   |
+-------------------+--------------------------------------------------------------------------------------+
```

3. Создайте порт из соответствующей политики:

```
(overcloud)$ openstack port create port-normal-qos --network net0 --qos-policy policy0
```

4. Создайте инстанс, указав порт сетевой карты (NIC) для использования

```
$ openstack server create --flavor cirros256 --image cirros-0.3.5-x86_64-disk --nic
port-id=port-normal-qos --wait qos_instance
```

Состояние вывода **ACTIVE** означает, что инстанс создан успешно на хосте, который может обеспечить запрашиваемую гарантированную минимальную пропускную способность.

8.1 Удаление QoS с гарантированной минимальной пропускной способностью с инстанса

Если нужно снять ограничение политики QoS с гарантированной минимальной пропускной способностью с инстанса, интерфейс можно отключить.

Порядок действий

- Для отключения интерфейса введите команду:

```
$ openstack server remove port <vm_name|vm_id> <port_name|port_id>
```

9 Обновление инстанса

Пользователь может добавлять и удалять дополнительные ресурсы из запущенных инстансов, например постоянное хранилище томов, сетевой интерфейс или общедоступный IP-адрес. Можно также обновлять метаданные инстанса и группы безопасности, к которым принадлежит инстанс.

9.1 Подключение сети к инстансу

К работающему инстансу можно подключить сеть.

При подключении сети к инстансу сервис вычислительных ресурсов Compute создает в сети порт для этого инстанса.

Используйте сеть для подключения сетевого интерфейса к инстансу, если нужно использовать группу безопасности по умолчанию и если в сети есть только одна подсеть.

Порядок действий

1. Определите доступные сети и отметьте имя или идентификатор сети, которую нужно подключить к инстансу:

```
(overcloud)$ openstack network list
```

2. Если нужная сеть недоступна, создайте новую сеть:

```
(overcloud)$ openstack network create <network>
```

3. Подключите сеть к инстансу.

```
$ openstack server add network <instance> <network>
```

- Поменяйте <instance> на имя или ID инстанса, к которому нужно подключить сеть.
- Поменяйте <network> на имя или ID сети, которую нужно подключить к инстансу.

9.2 Отключениие сети от инстанса

Пользователь может отключить сеть от инстанса.

При отключении сети отключаются все сетевые порты. Если инстанс имеет несколько портов в сети и нужно отключить только один из них, выполните порядок действий, описанный в разделе Отключение порта от инстанса.

Порядок действий

1. Определите сеть, подключенную к инстансу:

```
(overcloud)$ openstack server show <instance>
```

2. Отключите сеть от инстанса:

```
$ openstack server remove network <instance> <network>
```

- Поменяйте <instance> на имя или ID инстанса, от которого нужно отсоединить сеть.
- Поменяйте <network> на имя или ID сети, которую нужно отключить от инстанса.

9.3 Подключение порта к инстансу

Можно подключить сетевой интерфейс к работающему инстансу, используя порт. Единовременно можно подключить порт только к одному инстансу.

Используйте порт для подключения сетевого интерфейса к инстансу, если нужно использовать настраиваемую группу безопасности или если в сети есть несколько подсетей.

При подключении сетевого интерфейса при помощи сети порт создается автоматически.

К инстансу нельзя подключить порт с сетевой картой SR-IOV vNIC или порт с политикой QoS с гарантированной минимальной пропускной способностью.

Порядок действий

1. Определите доступные порты и отметьте имя или идентификатор порта, который нужно подключить к инстансу:

```
(overcloud)$ openstack port list
```

Если нужный порт недоступен, создайте новый порт:

```
(overcloud)$ openstack port create --network <network> <port>
```

- Поменяйте <network> на имя или ID сети, в которой нужно создать порт.
- Поменяйте <port> на имя или ID порта, который нужно подключить к инстансу.

2. Подключите порт к инстансу:

```
$ openstack server add port <instance> <port>
```

- Поменяйте <instance> на имя или ID инстанса, к которому надо подключить порт.
- Поменяйте <port> на имя или ID порта, который надо подключить к инстансу.

9.4 Отключение порта от инстанса

Порт можно отключить от инстанса.

Порядок действий

1. Определите порт, подключенный к инстансу:

```
(overcloud)$ openstack server show <instance>
```

2. Отключите порт от инстанса:

```
$ openstack server remove port <instance> <port>
```

- Поменяйте <instance> на имя или ID инстанса, от которого нужно отключить порт.
- Поменяйте <port> на имя или ID порта, который нужно отключить от инстанса.

9.5 Подключение тома к инстансу

Можно подключить том к инстансу для постоянного хранения.

Единовременно можно подключить том только к одному инстансу, за исключением случаев, когда он был настроен как том с множественным подключением.

Предварительные требования

- Для подключения тома с множественным подключением переменная среды OS_COMPUTE_API_VERSION установлена на 2.60 или более позднюю версию.

Порядок действий

1. Определите доступные тома. Или отметьте имя или ID диска, который нужно подключить к инстансу:

```
(overcloud)$ openstack volume list
```

2. Подключите том к инстансу:

```
$ openstack server add volume <instance> <volume>
```

- Поменяйте <instance> на имя или ID инстанса, к которому нужно подключить том.
- Поменяйте <volume> на имя или ID тома, который нужно подключить к инстансу.

Если при выполнении команды возникает указанная ниже ошибка, это означает, что том, который был выбран для подключения к инстансу, имеет множественное подключение. Поэтому необходимо использовать Compute API 2.60 или более поздней версии:
Multiattach volumes are only supported starting with compute API version 2.60. (HTTP 400) (Request-ID: req-3a969c31-e360-4c79-a403-75cc6053c9e5)
Можно установить переменную среды OS_COMPUTE_API_VERSION=2.72 либо добавить аргумент --os-compute-api-version при добавлении тома к инстансу:
$ openstack --os-compute-api-version 2.72 server add volume <instance> <volume>

Укажите версию --os-compute-api-version 2.20 или выше, чтобы добавить том к инстансу со статусом **SHELVED** или **SHELVED_OFFLOADED**.

3. Подтвердите, что том подключен к инстансу или инстансам:

```
$ openstack volume show <volume>
```

Поменяйте <volume> на имя или ID тома для отображения.
Пример выдачи:

```
+--------------------------------------+----------------+--------+------+--------------
--------------------------------------------------------+
| ID                                   | Name           | Status | Size | Attached to
|
+--------------------------------------+----------------+--------+------+--------------
--------------------------------------------------------+
| f3fb92f6-c77b-429f-871d-65b1e3afa750 | volMultiattach | in-use |   50 | Attached to
instance1 on /dev/vdb Attached to instance2 on /dev/vdb  |
+--------------------------------------+----------------+--------+------+--------------
--------------------------------------------------------+
```

9.6 Просмотр томов, подключенных к инстансу

Тома, подсоединенные к определенному инстансу, можно просмотреть.

Предварительные требования

- Используется python-openstackclient 5.5.0.

Порядок действий

Перечислите все тома, подключенные к инстансу:

```
$ openstack server volume list <instance>
+----------------------+----------+----------------------+-----------------------+
| ID                   | Device   | Server ID            | Volume ID             |
+----------------------+----------+----------------------+-----------------------+
| 1f9dcb02-9a20-4a4b- | /dev/vda | ab96b635-1e63-4487- | 1f9dcb02-9a20-4a4b-9f |
| 9f25-c7846a1ce9e8   |          | a85c-854197cd537b   | 25-c7846a1ce9e8       |
+----------------------+----------+----------------------+-----------------------+
```

9.7 Отключение тома от инстанса

Том можно отключить от инстанса.

При отключении сети отключаются все сетевые порты. Если у инстанса есть несколько портов в сети и нужно отключить только один из них, выполните действия, описанные в разделе Отключение порта от инстанса.

Порядок действий

1. Определите том, подключенный к инстансу:

```
(overcloud)$ openstack server show <instance>
```

2. Отключите том от инстанса:

```
$ openstack server remove volume <instance> <volume>
```

- Поменяйте <instance> на имя или ID инстанса, от которого нужно отключить том.
- Поменяйте <volume> на имя или ID тома, который нужно отключить от инстанса.

Укажите --os-compute-api-version 2.20 или выше, чтобы отключить том от инстанса со статусом **SHELVED** или **SHELVED_OFFLOADED**.

10 Предоставление публичного доступа к инстансу

Новые инстансы автоматически получают порт с фиксированным IP-адресом в сети, которой назначен инстанс. Этот IP-адрес является частным и непрерывно связан с инстансом до его удаления. Фиксированный IP-адрес используется для коммуникации между инстансами.

Пользователь может подключить общедоступный инстанс непосредственно к внешней сети совместного использования, где инстансу напрямую назначается общедоступный IP-адрес. Это удобно, если работа выполняется в частном облаке.

Также можно предоставить публичный доступ к инстансу через сеть проекта, имеющую маршрутизируемое соединение с сетью внешнего провайдера. Этот метод предпочтителен при работе в общедоступном облаке или когда общедоступные IP-адреса ограничены. Для предоставления публичного доступа через сеть проекта она должна быть подключена к маршрутизатору со шлюзом, настроенным на внешнюю сеть. Чтобы внешний трафик достиг инстанса, пользователь облака должен связать плавающий IP-адрес с инстансом.

Для обеспечения доступа к инстансу и из него, независимо от подключения к общей внешней сети либо маршрутизируемой сети провайдера, необходимо настроить правила группы безопасности для нужных протоколов, таких как SSH, ICMP или HTTP. При создании также необходимо передать пару ключей инстансу, чтобы иметь возможность получать к нему доступ удаленно.

Предварительные требования

- Во внешней сети имеется подсеть для предоставления плавающих IP-адресов.
- Сеть проекта подключена к маршрутизатору, внешняя сеть которого настроена в качестве шлюза.

10.1 Защита доступа к инстансу при помощи групп безопасности и пар ключей

Группы безопасности (security groups) – это наборы правил IP-фильтрации, которые контролируют доступ из сети и по протоколам к инстансам и от них, например ICMP, позволяющий проверять связь с инстансом, и SSH, позволяющий подключаться к инстансу.

Правила группы безопасности применяются ко всем инстансам в проекте.

Во всех проектах есть группа безопасности по умолчанию с названием default, которая используется, если пользователь не указал указал группу безопасности для своих инстансов.

По умолчанию группа безопасности разрешает весь исходящий и запрещает весь входящий трафик из любого источника, кроме инстансов из этой же группы безопасности.

Пользователь может добавить правила в группу безопасности по умолчанию либо создать новую группу безопасности для своего проекта.

К инстансу можно применять одну или несколько групп безопасности в процессе его создания. Чтобы применить группу безопасности к работающему инстансу, примените группу безопасности к порту, который подключен к инстансу.

Пары ключей (key pairs) – это учетные данные SSH или x509, которые внедряются в инстанс при запуске для обеспечения удаленного доступа к нему. Можно создавать новые пары ключей

в 1Stack или импортировать существующие пары ключей. У каждого пользователя должна быть хотя бы одна пара ключей. Пара ключей может использоваться для нескольких инстансов.

> Пользователи проекта не могут совместно использовать пары ключей, поскольку каждая пара ключей принадлежит отдельному пользователю, который создал или импортировал ее, а не проекту.

10.1.1 Создание группы безопасности

Можно создать новую группу безопасности (security group) и применить ее к инстансам и портам в проекте.

Порядок действий

1. Опционально: чтобы убедиться, что нужной группы безопасности нет, просмотрите доступные группы безопасности и соответствующие правила:

```
$ openstack security group list
$ openstack security group rule list <sec_group>
```

   - Поменяйте <sec_group> на имя или ID группы безопасности, которая была включена в список доступных групп.

2. Создайте свою группу безопасности:

```
$ openstack security group create mySecGroup
```

3. Добавьте правила созданной группе безопасности:

```
$ openstack security group rule create --protocol <protocol> \
[--dst-port <port-range>] \
[--remote-ip <ip-address> | --remote-group <group>] \
[--ingress | --egress] mySecGroup
```

   - Поменяйте <protocol> на имя протокола, которому нужно разрешить связь с инстансами пользователя.
   - Опционально: поменяйте <port-range> на порт назначения или диапазон портов, который будет открыт для протокола. Необходимо для IP-протоколов TCP, UDP и SCTP. Установите значение 1, чтобы разрешить все порты для указанного протокола.
   - Опционально: можно разрешить доступ только с указанных IP-адресов, используя **--remote-ip** для указания блока удаленных IP-адресов, или **--remote-group**, чтобы указать, что правило применимо только к пакетам от интерфейсов, которые являются участником удаленной группы. При использовании **--remote-ip** поменяйте <ip-address> на блок удаленных IP-адресов. Можно использовать нотацию CIDR. При использовании **--remote-group** поменяйте <group> на имя или ID существующей группы безопасности.

     Если не указана ни одна из опций, доступ будет разрешен ко всем адресам, которые составляют диапазон удаленного IP-доступа по умолчанию (IPv4 default: 0.0.0.0/0; IPv6 default: ::/0).

- Укажите направление сетевого трафика, к которому применяется правило протокола: входящий (ingress) или исходящий (egress). При отсутствии указаний по умолчанию используется ingress.

4. Повторите шаг 3 для создания правила для всех протоколов, которым нужно разрешить доступ к инстансам пользователя.

В примере показано создание правила для разрешения SSH-подключений к инстансам в группе безопасности mySecGroup:

```
$ openstack security group rule create --protocol tcp --dst-port 22 mySecGroup
```

10.1.2 Обновление правил группы безопасности

Обновлять правила можно для любой группы безопасности, к которой имеется доступ.

Порядок действий

1. Получите имя или ID группы безопасности, для которой нужно обновить правила:

```
$ openstack security group list
```

2. Определите правила, которые необходимо применить к группе безопасности.
3. Добавьте правила к группе безопасности:

```
$ openstack security group rule create --protocol <protocol> \
[--dst-port <port-range>] \
[--remote-ip <ip-address> | --remote-group <group>] \
[--ingress | --egress] <group_name>
```

- Поменяйте <protocol> на имя протокола, который нужно разрешить для связи с инстансами.
- Опционально: поменяйте <port-range> на порт назначения или диапазон портов, который будет открыт для протокола. Требуется для IP-протоколов TCP, UDP и SCTP. Установите значение 1, чтобы разрешить все порты для указанного протокола.
- Опционально: можно разрешить доступ только с указанных IP-адресов, используя **--remote-ip** для указания блока удаленных IP-адресов, или **--remote-group**, чтобы указать, что правило применимо только к пакетам от интерфейсов, которые являются участником удаленной группы. При использовании **--remote-ip** поменяйте <ip-address> на блок удаленных IP-адресов. Можно использовать нотацию CIDR. При использовании **--remote-group** поменяйте <group> на имя или ID существующей группы безопасности.

  Если не указана ни одна из опций, доступ будет разрешен ко всем адресам, которые составляют диапазон удаленного IP-доступа по умолчанию (IPv4 default: 0.0.0.0/0; IPv6 default: ::/0).
- Укажите направление сетевого трафика, к которому применяется правило протокола: входящий (ingress) или исходящий (egress). При отсутствии указания по умолчанию используется ingress.

- Поменяйте <group_name> на имя или ID группы безопасности, к которой нужно применить правило.

4. Повторите шаг 3 для создания правил для всех протоколов, которым нужно разрешить доступ к инстансам пользователя.

В примере показано создание правила для разрешения SSH-подключений к инстансам в группе безопасности mySecGroup:

```
$ openstack security group rule create --protocol tcp --dst-port 22 mySecGroup
```

10.1.3 Удаление правил группы безопасности

Правила можно удалять из группы безопасности.

Порядок действий

1. Определите группу безопасности, к которой применяются правила:

```
$ openstack security group list
```

2. Получите ID правил, связанных с группой безопасности:

```
$ openstack security group show <sec-group>
```

3. Удалите правило или правила:

```
$ openstack security group rule delete <rule> [<rule> ...]
```

Поменяйте <rule> на ID правила, которое нужно удалить. Можно удалять сразу несколько правил. Для этого укажите списком идентификаторы правил для удаления и разделите их пробелами.

10.1.4 Добавление группы безопасности на порт

Группа безопасности по умолчанию применяется к инстансам, для которых не указана альтернативная группа безопасности. Альтернативную группу безопасности можно применить к порту работающего инстанса.

Порядок действий

1. Определите порт инстанса, к которому нужно применить группу безопасности:

```
$ openstack port list --server myInstancewithSSH
```

2. Примените группу безопасности к порту:

```
$ openstack port set --security-group <sec_group> <port>
```

Поменяйте <sec_group> на имя или ID группы безопасности, которую нужно применить к порту работающего инстанса пользователя.

Опцию **--security-group** можно использовать неоднократно, если нужно применить несколько групп безопасности.

10.1.5 Удаление группы безопасности с порта

Для удаления группы безопасности из порта, необходимо прежде всего удалить все группы безопасности, а затем повторно добавить те группы безопасности, которые нужно оставить назначенными для порта.

Порядок действий

1. Получите список всех групп безопасности, связанных с портом, и отметьте идентификаторы групп безопасности, которые нужно оставить связанными с портом:

```
$ openstack port show <port>
```

2. Удалите все группы безопасности, связанные с портом:

```
$ openstack port set --no-security-group <port>
```

3. Повторно примените группы безопасности к порту:

```
$ openstack port set --security-group <sec_group> <port>
```

Поменяйте <sec_group> на ID группы безопасности, которую нужно повторно применить к порту работающего инстанса пользователя.

Опцию **--security-group** можно использовать неоднократно, если нужно применить несколько групп безопасности.

10.1.6 Удаление группы безопасности

Удалять можно только те группы безопасности, которые не связаны ни с одним портом.

Порядок действий

1. Получите имя или ID группы безопасности, которую нужно удалить:

```
$ openstack security group list
```

2. Получите список доступных портов:

```
$ openstack port list
```

3. Проверьте каждый порт на наличие связанной с ним группы безопасности:

```
$ openstack port show <port-uuid> -c security_group_ids
```

Если группа безопасности, которую нужно удалить, связана с каким-либо портом, предварительно удалите ее из порта.

4. Удалите группу безопасности:

```
$ openstack port set --security-group <sec_group> <port>
```

Поменяйте <group> на идентификатор группы, которую нужно удалить.

Можно удалять сразу несколько групп. Для этого укажите списком идентификаторы групп для удаления и разделите их пробелами.

10.1.7 Генерация новой пары SSH-ключей

Можно сгенерировать новую пару SSH-ключей (SSH key pair) для использования в проекте пользователя.

Используйте сертификат x509 для создания пары ключей для инстанса Windows.

Порядок действий

1. Создайте пару ключей и сохраните закрытый ключ в локальной директории .ssh.

```
$ openstack keypair create <keypair> > ~/.ssh/<keypair>.pem
```

Поменяйте <keypair> на имя новой пары ключей.

2. Защитите закрытый ключ:

```
$ chmod 600 ~/.ssh/<keypair>.pem
```

10.1.8 Импорт существующей пары SSH-ключей

Пользователь может импортировать SSH-ключ в свой проект, созданный вне платформы 1Stack, предоставив файл открытого ключа при создании новой пары ключей.

Порядок действий

1. Создайте пару ключей из существующего файла ключей и сохраните закрытый ключ в локальной директории .ssh:
   - Для импорта пары ключей из существующего файла открытого ключа, введите следующую команду:

```
$ openstack keypair create --public-key ~/.ssh/<public_key>.pub
   <keypair> > ~/.ssh/<keypair>.pem
```

   - Поменяйте <public_key> на имя файла открытого ключа, который нужно использовать для создания пары ключей.
   - Поменяйте <keypair> на имя новой пары ключей.

- Чтобы импортировать пару ключей из существующего файла закрытого ключа, введите следующую команду:

```
$ openstack keypair create --private-key ~/.ssh/<private_key>
   <keypair> > ~/.ssh/<keypair>.pem
```

- Поменяйте <private_key> на имя файла открытого ключа, который нужно использовать для создания пары ключей.
- Поменяйте <keypair> на имя новой пары ключей.

2. Защитите закрытый ключ:

```
$ chmod 600 ~/.ssh/<keypair>.pem
```

10.2 Назначение плавающего IP-адреса инстансу

Можно назначить инстансу общедоступный плавающий IP-адрес (floating IP) для связи с сетями за пределами облака, включая сеть Интернет.

Администратор облака настраивает доступный пул плавающих IP-адресов для внешней сети. Пользователь может назначить плавающий IP-адрес из этого пула своему проекту, а затем связать плавающий IP-адрес со своим инстансом.

Проекты имеют ограниченную квоту плавающих IP-адресов, которые могут использоваться инстансами в проекте, по умолчанию – 50. По этой причине следует освобождать IP-адреса для повторного использования, когда они больше не нужны.

Предварительные требования

- Инстанс находится во внешней сети или в сети проекта, подключенной к маршрутизатору, внешняя сеть которого настроена в качестве шлюза.
- Внешняя сеть, к которой будет подключаться инстанс, содержит подсеть для предоставления плавающих IP-адресов.

Порядок действий

1. Проверьте плавающие IP-адреса, выделенные для текущего проекта:

```
$ openstack floating ip list
```

- Если свободные плавающие IP-адреса, которые нужно использовать, отсутствуют, выделите плавающий IP-адрес для конкретного проекта из пула внешней сети:

```
$ openstack floating ip create <provider-network>
```

- Поменяйте <provider-network> на имя или ID внешней сети, которую нужно использовать для предоставления внешнего доступа.

По умолчанию плавающий IP-адрес выделяется из пула внешней сети случайным образом. Администратор облака может использовать опцию --floating-ip-address для выделения определенного плавающего IP-адреса из внешней сети.

2. Назначьте плавающий IP-адрес инстансу:

```
$ openstack server add floating ip [--fixed-ip-address <ip_address>] <instance>
<floating_ip>
```

- Поменяйте <instance> на имя или ID инстанса, к которому надо предоставить открытый доступ.
- Поменяйте <floating_ip> на плавающий IP-адресом, который нужно назначить инстансу.
- Опционально: поменяйте <ip_address> на IP-адрес интерфейса, к которому нужно привязать плавающий IP-адрес. По умолчанию это вызовет присоединение плавающего IP-адеса к первому порту.

3. Проверьте, что плавающий IP-адрес назначен инстансу:

```
$ openstack server show <instance>
```

10.3 Отвязка плавающего IP-адреса от инстанса

Когда публичный доступ к инстансу более не нужен, отвяжите его от инстанса и верните в пул распределения.

Порядок действий

1. Отвяжите плавающий IP-адрес от инстанса:

```
$ openstack server remove floating ip <instance> <ip_address>
```

- Поменяйте <instance> на имя или ID инстанса, к которому нужно закрыть публичный доступ.
- Поменяйте <floating_ip> на IP-адрес, назначенный инстансу.

2. Верните плавающий IP-адрес обратно в пул распределения.

```
$ openstack floating ip delete <ip_address>
```

3. Подтвердите, что плавающий IP-адрес удален и больше не доступен для назначения:

```
$ openstack floating ip list
```

10.4 Создание инстанса с SSH-доступом

Можно предоставить SSH-доступ к инстансу, указав пару ключей при его создании.

Пары ключей – это учетные данные SSH или x509, которые внедряются в инстанс при запуске. В каждом проекте должна быть как минимум одна пара ключей. Пара ключей принадлежит индивидуально пользователю, а не проекту.

> Нельзя связать пару ключей с инстансом уже после создания инстанса.

Предварительные требования

- Доступна пара ключей, которую пользователь может использовать для SSH доступа к своему инстансу.
- Сеть, в которой планируется создать инстанс, – это внешняя сеть или сеть проекта, которая подключена к маршрутизатору, имеющему внешнюю сеть. В сети проекта этот маршрутизатор должен быть указан в качестве шлюза.
- Внешняя сеть, к которой подключается инстанс, включает подсеть для предоставления плавающих IP-адресов.
- Группой безопасности разрешен доступ по SSH к инстансам.
- Образ, на котором основан инстанс, содержит пакет cloud-init для внедрения открытого ключа SSH в инстанс.
- Доступен плавающий IP-ключ для назначения инстансу.

Порядок действий

1. Получите имя или ID шаблона с профилем аппаратного оборудования, которое необходимо для инстанса:

```
$ openstack flavor list
```

> Выберите шаблон достаточного размера для успешной загрузки образа, в противном случае произойдет сбой загрузки инстанса.

2. Получите имя или ID образа с профилем ПО, который необходим для инстанса:

```
$ openstack image list
```

Если нужный образ недоступен, можно скачать или создать новый образ.

3. Получите имя или ID сети, к которой нужно подключить инстанс:

```
$ openstack network list
```

4. Получите имя пары ключей, которая будет использоваться для удаленного доступа к инстансу.

```
$ openstack keypair list
```

5. Создайте инстанс с доступом по SSH:

```
$ openstack server create --flavor <flavor> --image <image> --network <network> \
  [--security-group <secgroup>] --key-name <keypair> --wait myInstancewithSSH
```

- Поменяйте <flavor> на имя или ID шаблона инстанса, полученные при выполнении шага 1.
- Поменяйте <image> на имя или ID образа, полученные при выполнении шага 2.
- Поменяйте <network> на имя или ID сети, полученные при выполнении шага 3.

  Опцию **--network** можно использовать несколько раз, если нужно подключить инстанс к нескольким сетям.
- Опционально: группа безопасности по умолчанию применяется к инстансам, для которых не указана альтернативная группа безопасности. Пользователь может применить альтернативную группу безопасности непосредственно к инстансу при создании или к порту работающего инстанса.

  Используйте параметр **--security-group**, чтобы указать альтернативную группу безопасности при создании инстанса.
- Поменяйте <keypair> на имя или ID пары ключей, полученные при выполнении шага 4.

6. Назначьте инстансу плавающий IP-адрес:

```
$ openstack server add floating ip myInstancewithSSH <floating_ip>
```

- Поменяйте <floating_ip> на плавающий IP-адрес, который нужно назначить инстансу.

7. Используйте автоматически созданную учетную запись облачного пользователя, чтобы убедиться, можно войти в инстанс при помощи SSH:

```
$ ssh -i ~/.ssh/<keypair>.pem cloud-user@<floatingIP>
```

11 Подключение к инстансу

Пользователь может получить доступ к инстансу из внешнего местоположения по отношению к облаку с помощью удаленной оболочки типа SSH или WinRM, если протокол был разрешен в правилах группы безопасности инстанса. Также можно подключиться напрямую к консоли инстанса для отладки даже при сбоях сетевого подключения.

Если пользователь не предоставил инстансу пару ключей или не назначил группу безопасности для инстанса, то получить доступ к инстансу можно только изнутри облака через VNC-консоль. Отправить ping-запрос нельзя.

11.1 Доступ к консоли инстанса

К VNC-консоли (VNC console) инстанса можно подключиться напрямую – для этого ввести ее URL-адрес в браузере.

Порядок действий

1. Для отображения URL-адреса VNC-консоли инстанса, введите команду:

```
$ openstack console url show <vm_name>
+-------+-----------------------------------------------------+
| Field | Value                                               |
+-------+-----------------------------------------------------+
| type  | novnc                                               |
| url   | http://172.25.250.50:6080/vnc_auto.html?token=      |
|       | 962dfd71-f047-43d3-89a5-13cb88261eb9                |
+-------+-----------------------------------------------------+
```

2. Чтобы подключиться напрямую к консоли VNC, введите отображаемый URL-адрес в браузере.

11.2 Вход в инстанс

В общедоступные инстансы можно войти удаленно.

Предварительные требования

- У пользователя есть сертификат пары ключей для инстанса. Сертификат загружается при создании пары ключей. Если пользователь не создавал пару ключей самостоятельно, нужно обратиться к администратору.
- Инстанс настроен как общедоступный.
- У пользователя есть учетная запись пользователя облака.

Порядок действий

1. Получите плавающий IP-адрес инстанса, в который нужно зайти:

```
$ openstack server show <instance>
```

2. Поменяйте <instance> на имя или ID инстанса, к которому нужно подключиться.
3. Используйте созданный автоматически аккаунт облачного пользователя для входа в инстанс:

```
$ ssh -i ~/.ssh/<keypair>.pem cloud-user@<floatingIP>
```

- Поменяйте <keypair> на имя пары ключей.
- Поменяйте <floating_ip> на плавающий IP-адрес инстанса.

Для авторизации в инстансе без плавающего IP-адреса можно использовать команду:
$ openstack server ssh --login cloud-user --identity ~/.ssh/<keypair>.pem --private <instance>

- Поменяйте <keypair> на имя пары ключей.
- Поменяйте <instance> на имя или ID инстанса, к которому надо подключиться.

12 Управление инстансом

Пользователь может выполнять операции управления инстансом, например, изменять размер инстанса или выключать инстанс, очищать ресурсы.

12.1 Изменение размера инстанса

Необходимость в изменении размера инстанса возникает, если нужно увеличить или уменьшить количество памяти или процессоров (CPU) инстанса.

Чтобы изменить размер инстанса, выберите новый шаблон инстанса с нужной емкостью. При изменении размера инстанса осуществляется его пересборка и перезапуск.

Порядок действий

1. Получите имя или ID инстанса, размер которого нужно изменить:

```
$ openstack server list
```

2. Получите имя или ID шаблона, который будет использоваться для изменения размера инстанса:

```
$ openstack flavor list
```

3. Измените размер инстанса:

```
$ openstack server resize --flavor <flavor> --wait <instance>
```

- Поменяйте <flavor> на имя или ID шаблона, полученные при выполнении шага 2.
- Поменяйте <instance> на имя или ID инстанса, размер которого изменяется.

Изменение размера может занять некоторое время. Операционная система инстанса выполняет регулируемое завершение работы перед отключением инстанса и изменением его размера. В это время статус инстанса – **RESIZE**:

```
$ openstack server list
+----------------------+---------------+--------+-----------------------------------+
| ID                   | Name          | Status | Networks                          |
+----------------------+---------------+--------+-----------------------------------+
| 67bc9a9a-5928-47c... | myCirrosServer | RESIZE | admin_internal_net=192.168.111.139|
+----------------------+---------------+--------+-----------------------------------+
```

4. После завершения изменения размера статус инстанса поменяется на **VERIFY_RESIZE**. Затем необходимо подтвердить либо отменить изменение размера:
   - Чтобы подтвердить изменение размера, введите команду:

```
$ openstack server resize confirm <instance>
```

   - Чтобы отменить изменение размера, введите команду:

```
$ openstack server resize revert <instance>
```

Инстанс вернется к исходному шаблону, а статус поменяется на **ACTIVE**.

Облако может быть настроено на автоматическое подтверждение изменения размера инстанса, если изменения не подтверждены или не отменены в течение установленного периода времени.

12.2 Создание снимка инстанса

Снимок (snapshot) – это образ, фиксирующий состояние рабочего диска инстанса.

Пользователь может сделать снимок инстанса, чтобы использовать его в качестве шаблона при создании новых инстансов.

Снимки позволяют создавать новые инстансы из другого инстанса и восстанавливать состояние инстанса.

Если удалить инстанс, на котором основан снимок, останется возможность использовать снимок для создания нового инстанса в том же состоянии, что и снимок.

Порядок действий

1. Получите имя или ID инстанса, с которого необходимо сделать снимок:

```
$ openstack server list
```

2. Создайте снимок:

```
$ openstack server image create --name <image_name> <instance>
```

   - Поменяйте <image_name> на имя нового изображения снимка.
   - Поменяйте <instance> на имя или ID инстанса, с которого нужно сделать снимок.

3. Опционально: чтобы обеспечить консистентное состояние диска, когда снимок инстанса используется как шаблон для создания новых инстансов, включите гостевой агент QEMU и укажите, что файловая система должна быть приостановлена при обработке снимка, добавив в образ снимка следующие метаданные:

```
$ openstack image set --property hw_qemu_guest_agent=yes --property
os_require_quiesce=yes <image_name>
```

QEMU Guest Agent – это фоновый процесс, который помогает приложениям управления выполнять команды на уровне ОС инстанса. Активация этого агента добавляет к инстансу еще одно устройство, которое потребляет слот PCI, и ограничивает количество других устройств, которые можно выделить для инстанса. Также при этом инстансы Windows отображают предупреждение о неизвестном аппаратном устройстве.

12.3 Восстановление инстанса

В экстренной ситуации, например при системном сбое или отказе доступа, можно перевести инстанс в режим восстановления (rescue mode). При этом инстанс выключается, перезагружается с новым диском инстанса и монтирует оригинальный диск инстанса и диск конфигурации (config drive) в качестве тома на перезагруженном инстансе. Можно подключиться к перезагруженному инстансу и просмотреть оригинальный диск инстанса для восстановления системы и данных.

> Нельзя восстановить инстанс, загруженный с тома диска (volume).

Порядок действий

1. Восстановите инстанс:

```
$ openstack server rescue [--image <image>] <instance>
```

   - Опционально: по умолчанию инстанс загружается из аварийного образа, предоставленного администратором облака, или из новой копии исходного образа инстанса. Используйте параметр **--image**, чтобы указать альтернативный образ, который будет использоваться при перезагрузке инстанса в режиме восстановления.
   - Поменяйте <instance> на имя или ID инстанса, который нужно восстановить.
2. Подключитесь к восстановленному инстансу, чтобы устранить проблему.
3. Перезагрузите инстанс с обычного загрузочного диска:

```
$ openstack server unrescue <instance>
```

12.4 Выключение инстанса и очистка ресурсов ЦПУ и памяти

Неиспользуемый инстанс, который не планируется удалять, можно выключить с очисткой ресурсов ЦПУ и памяти (shelve). При таком выключении инстанса сохраняются его данные и выделенные ресурсы, но его память очищается.

В зависимости от конфигурации облака выключенные инстансы переводятся в состояние **SHELVED_OFFLOADED** сразу либо после заданной временной задержки.

При выключении инстанса с очисткой ЦПУ и памяти сервис вычислительных ресурсов (Compute) создает моментальный снимок, фиксирующий состояние инстанса, и присваивает ему имя в формате <инстанс>-shelved. Снимок удаляется при включении инстанса или при его удалении.

Если выключенный инстанс более не нужен, его можно удалить.

Также одновременно можно выключить с очисткой ресурсов несколько инстансов.

Порядок действий

1. Получите имя или ID инстанса или инстансов, которые нужно выключить (shelve):

```
$ openstack server list
```

2. Выключите инстанс или инстансы с очисткой ЦПУ и памяти:

```
$ openstack server shelve <instance> [<instance> ...]
```

Поменяйте <instance> на имя или ID инстанса, который нужно выключить. При необходимости можно указать несколько выключаемых инстансов.

3. Убедитесь, что инстанс выключен.

```
$ openstack server list
```

Выключенные инстансы имеют статус **SHELVED_OFFLOADED**.

12.5 Операции управления инстансом

После создания инстансапользователь может выполнять следующие операции управления:

Операция	**Описание**	**Команда**
Остановить инстанс	Останавливает инстанс	*openstack server stop*
Запустить инстанс	Запускает остановленный инстанс	*openstack server start*
Поставить работу инстанса на паузу	Мгновенно приостанавливает работу инстанса. Состояние инстанса хранится в памяти (ОЗУ). Приостановленный инстанс продолжает работу в замороженном состоянии. Подтверждать действие приостановки не требуется	*openstack server pause*
Возобновить работу приостановленного инстанса	Мгновенно возобновляет работу приостановленного инстанса. Подтвердить действие возобновления не требуется	*openstack server unpause*
Перевести работающий инстанс в состояние ожидания	Немедленно переводит инстанс в состояние ожидания. Состояние инстанса хранится на диске инстанса. Подтверждать действие перевода в состояние ожидания не требуется	*openstack server suspend*
Возобновить работу приостановленного инстанса	Мгновенно возобновляет работу инстанса в состоянии ожидания. Состояние инстанса хранится на диске инстанса. Подтверждать действие возобновления работы не требуется	*openstack server resume*
Удалить инстанс	Удаляет инстанс навсегда.	*openstack server delete*

Операция	Описание	Команда
	Подтверждать действие по уничтожению не требуется. Удаленные инстансы невозможно восстановить, кроме случаев, когда в облаке настроено обратимое удаление. **Примечание** Удаление инстанса не приводит к удалению подключенных к нему томов. Подключенные тома необходимо удалить отдельно	
Редактировать данные инстанса	Пользователь может использовать метаданные инстанса, чтобы указать свойства инстанса	openstack server set --property <key=value> [--property <key=value>] <instance>
Добавить группы безопасности	Добавляет указанную группу безопасности к инстансу	*openstack server add security group*
Удалить группы безопасности	Удаляет указанную группу безопасности для инстанса	*openstack remove security group*
Произвести аварийное восстановление инстанса	В чрезвычайной ситуации, например, при сбое системы или отказе доступа, можно перевести инстанс в режим аварийного восстановления. При этом инстанс отключается, а корневой диск подключается к временному серверу. Чтобы восстановить систему и данные пользователь может подключиться к временному серверу. Также можно перезагрузить работающий инстанс в режим восстановления. Например, такая операция может быть необходима при повреждении файловой системы инстанса. **Примечание** Выполнять аварийное восстановление инстанса, загруженного с тома, нельзя	*openstack server rescue*
Восстановить инстанс после аварийного восстановления	Восстанавливает инстанс после аварийного восстановления	*openstack server unrescue*
Просмотреть журналы инстанса	Просмотр последнего раздела журнала консоли инстанса	*openstack console log show*
Выключить инстанс и очистить ресурсы ЦПУ и памяти	Если выключить инстанс с очисткой ресурсов, будут сохранены его данные и распределение ресурсов, при этом память инстанса очистится. В зависимости от конфигурации облака выключенные инстансы переводятся в состояние **SHELVED_OFFLOADED** сразу либо после заданной временной задержки. Когда инстанс находится в состоянии **SHELVED_OFFLOADED**, его данные и выделение	*openstack server shelve*

Операция	Описание	Команда
	ресурсов удаляются. Состояние инстанса хранится на диске инстанса. Если экземпляр был загружен с тома, он мгновенно переходит в состояние **SHELVED_OFFLOADED**. Подтверждать действие по выключению с очисткой ресурсов не требуется	
Включить инстанс обратно	Восстанавливает инстанс, используя образ диска выключенного инстанса.	*openstack server unshelve*
Блокировать инстанс	Блокирует инстанс, чтобы запретить пользователям, не являющимся администраторами, выполнять действия на инстансе	*openstack server lock* *openstack server unlock*
Мягкая перезагрузка инстанса	Мягко останавливает и перезапускает инстанс. При мягкой перезагрузке выполняется попытка мягко завершить все процессы перед перезапуском экземпляра. При перезагрузке инстанса по умолчанию происходит мягкая перезагрузка	*openstack server reboot --soft <server>*
Жесткая перезагрузка инстанса	Остановка и перезапуск инстанса. При жесткой перезагрузке питание инстанса отключается, а затем снова включается	*openstack server reboot --hard <server>*
Пересборка инстанса	Используйте новые параметры образа и разделения диска для пересборки инстанса, включая выключение инстанса, создание нового образа и перезагрузку. Используйте эту опцию при возникновении проблем с операционной системой вместо прерывания работы инстанса и его повторного запуска	*openstack server rebuild*

13 Создание кастомизированного инстанса

Пользователи облака могут указать дополнительные данные для использования при запуске инстанса, например сценарий оболочки, который инстанс запускает при загрузке.

Пользователь облака может применять следующие методы для передачи данных в инстансы:

Данные пользователя (User Data)

Используйте для включения инструкций в команду запуска инстанса для выполнения cloud-init.

Метаданные инстанса (Instance metadata)

Список пар «ключ-значение», которые можно указать при создании или обновлении инстанса.

Пользователь может получить доступ к дополнительным данным, передаваемым в инстанс, используя диск конфигурации или сервис метаданных.

Диск конфигурации (Config drive)

Пользователь может подключать диск конфигурации к инстансу при загрузке.

Диск конфигурации может использоваться инстансом как диск, доступный только для чтения. Инстанс может подключать этот диск и считывать с него файлы.

Диск конфигурации можно использовать в качестве источника информации **cloud-init**.

Диски конфигурации удобны в сочетании с **cloud-init** для создания сервера, а также если нужно передать большие файлы своему инстансу. Например, можно настроить **cloud-init** для автоматического подключения диска конфигурации и запуска сценариев установки при начальной загрузке инстанса.

Диски конфигурации создаются с меткой тома **config-2** и подключаются к инстансу при его загрузке. Содержимое любых дополнительных файлов, передаваемых на диск конфигурации, добавляется в файл **user_data** в каталоге openstack/{version}/ диска конфигурации. Из этого файла cloud-init извлекает пользовательские данные.

Сервис метаданных (Metadata service)

Предоставляет REST API для получения данных, относящихся к инстансу. Инстансы получают доступ к этому сервису по адресу 169.254.169.254 или fe80::a9fe:a9fe.

cloud-init может использовать диск конфигурации, а также сервис метаданных для добавления дополнительных данных при настройке инстанса.

Пакет **cloud-init** поддерживает несколько форматов ввода данных. Скрипты оболочки и формат **cloud-config** являются наиболее распространенными входными форматами:

- сценарии оболочки: объявление данных начинается с #! или Content-Type: text/x-shellscript. Сценарии оболочки вызываются в процессе загрузки в последнюю очередь.
- формат **cloud-config**: объявление данных начинается с #cloud-config или Content-Type: text/cloud-config. Файлы **cloud-config** должны иметь корректный формат YAML, чтобы их можно было анализировать и исполнять с помощью **cloud-init**.

cloud-init имеет максимальный размер 16384 байт для пользовательских данных, передаваемых в инстанс. Ограничние по размеру изменить нельзя: используйте сжатие gzip, если нужно превысить ограничение размера.

Данные, специфические для вендора (Vendor-specific data)

Администратор 1Stack также может передавать данные инстансам при их создании. Эти данные могут быть не видны пользователю облака, например криптографический токен, который регистрирует инстанс в Active Directory.

Администратор 1Stack использует функцию vendordata для передачи данных в инстансы. Конфигурация vendordata доступна только для чтения и находится в одном из следующих файлов:

1. /openstack/{version}/vendor_data.json.
2. /openstack/{version}/vendor_data2.json.

Пользователь может просмотреть эти файлы с помощью сервиса метаданных или с диска конфигурации своего инстанса. Чтобы получить доступ к файлам через сервис метаданных, отправьте запрос GET по адресу:

http://169.254.169.254/openstack/{version}/vendor_data.json

или

http://169.254.169.254/openstack/{version}/vendor_data2.json.

13.1 Кастомизация инстанса при помощи данных пользователя

С помощью данных пользователя можно включать инструкции в команду запуска.

cloud-init выполняет эти команды для кастомизации инстанса на последнем этапе загрузки.

Порядок действий

1. Создайте файл с инструкциями для **cloud-init**, например, сценарий bash, который инсталлирует и активирует веб-сервер на инстансе:

```
$ cat /home/scripts/install_httpd
#!/bin/bash
yum -y install httpd python-psycopg2
systemctl enable httpd –now
```

2. Запустите инстанс с опцией **--user-data** для передачи сценария bash.

```
$ openstack server create --image rhel8 --flavor default --nic net-id=web-server-
network --security-group default \
  --key-name web-server-keypair --user-data /home/scripts/install_httpd --wait web-
server-instance
```

3. Когда инстанс находится в активном состоянии, подключите плавающий IP-адрес.

```
$ openstack floating ip create web-server-network
$ openstack server add floating ip web-server-instance 172.25.250.123
```

4. Войдите в инстанс при помощи SSH:

```
$ ssh -i ~/.ssh/web-server-keypair cloud-user@172.25.250.123
```

5. Убедитесь, что настройка прошла успешно. Например, чтобы проверить, что веб-сервер установлен и включен, введите следующую команду:

```
$ curl http://localhost | grep Test
<title>Test Page for the Apache HTTP Server on MSVSphere</title>
<h1>MSVSphere <strong>Test Page</strong></h1>
```

6. Проверьте файл /var/log/cloud-init.log на предмет релевантных сообщений, например, выполнен ли **cloud-init**.

```
$ sudo less /var/log/cloud-init.log
...output omitted...
...util.py[DEBUG]: Cloud-init v. 0.7.9 finished at Sat, 23 Jun 2018 02:26:02 +0000.
Datasource DataSourceOpenStack [net,ver=2].  Up 21.25 seconds
```

13.2 Кастомизация инстанса при помощи метаданных

Метаданные инстанса можно использовать для указания свойств инстанса в команде запуска инстанса.

Порядок действий

1. Запустите инстанс при помощи опции **--property <key=value>**. Например, чтобы отметить инстанс как веб-сервер, установите следующее свойство:

```
$ openstack server create --image rhel8 --flavor default --property role=webservers --
wait web-server-instance
```

2. Опционально: добавьте дополнительное свойство к инстансу после его создания, например:

```
$ openstack server set --property region=emea --wait web-server-instance
```

13.3 Кастомизация инстанса при помощи диска конфигурации

Пользователь может создать диск конфигурации (Config Drive) для инстанса, который будет подключаться в процессе загрузки инстанса. Содержимое можно передать на диск конфигурации, чтобы сделать его доступным для инстанса.

Порядок действий

1. Активируйте диск конфигурации и укажите файл с содержимым, которое нужно сделать доступным на диске конфигурации. Например, следующая команда создает новый инстанс с именем **config-drive-instance** и подключает диск конфигурации с содержимым файла **my-user-data.txt**:

```
(overcloud)$ openstack server create --flavor m1.tiny --config-drive true --user-data
./my-user-data.txt \
             --image cirros config-drive-instance
```

Эта команда создает диск конфигурации с меткой тома **config-2**, который прикрепляется к инстансу при его загрузке, и добавляет содержимое **my-user-data.txt** в файл **user_data** в каталог диска конфигурации openstack/{version}/.

2. Войдите в инстанс
3. Подключите диск конфигурации:
   - Если операционная система инстанса использует udev:

```
# mkdir -p /mnt/config
# mount /dev/disk/by-label/config-2 /mnt/config
```

   - Если ОС инстанса не использует udev, необходимо сначала определить блочное устройство, соответствующее диску конфигурации:

```
# blkid -t LABEL="config-2" -o device /dev/vdb
# mkdir -p /mnt/config
# mount /dev/vdb /mnt/config
```

14 Управление сетями проекта

Сети проекта (networks) помогают изолировать сетевой трафик для облачных вычислений. Шаги по созданию сети проекта включают планирование и создание сети и добавление подсетей и маршрутизаторов.

14.1 Планирование сетей VLAN

При планировании развертывания платформы 1Stack следует начать с нескольких подсетей и выделить из них отдельные IP-адреса.

Если используется несколько подсетей, можно разделить трафик между системами на сетях VLAN. Например, в идеале трафик управления или API не должен находиться в той же сети, что и системы, обслуживающие веб-трафик.

Трафик между сетями VLAN проходит через маршрутизатор, где пользователь может установить межсетевые экраны для управления потоком трафика.

Необходимо планировать VLAN-сети в рамках общего плана, который включает изоляцию трафика, высокую доступность и использование IP-адресов для различных типов виртуальных сетевых ресурсов в развертывании.

Максимальное количество VLAN в одной сети или в одном агенте OVS для сервера сети – 4094.
В ситуациях, когда требуется превысить максимальное количество VLAN, можно создать несколько сетей провайдера (сети VXLAN) и несколько серверов сети – по одному на сеть.
Каждый сервер может содержать до 4094 частных сетей.

14.2 Создание сети

Создайте сеть так, чтобы инстансы могли взаимодействовать друг с другом и получать IP-адреса с помощью DHCP.

При создании сетей нужно иметь в виду, что сети могут содержать несколько подсетей. Это важно, если планируется разместить в одной сети совершенно разные системы и требуется обеспечить определенную изоляцию между ними.

Например, можно указать, что в одной подсети должен присутствовать только трафик веб-сервера, а трафик базы данных проходить через другую подсеть. Подсети изолированы друг от друга, и любой инстанс, который будет взаимодействовать с другой подсетью, должен направлять трафик через маршрутизатор.

Рассмотрите возможность размещения систем, которым необходим большой объем трафика, в одной подсети, чтобы им не требовалась маршрутизация и впоследствии можно было избежать задержек и нагрузки.

1. На панели управления выберите **Project** > **Network** > **Networks.**
2. Нажмите **+Create Network** и укажите следующие значения:

Поле	Описание
Network Name (Имя сети)	Имя, описывающее роль, которую будет выполнять сеть. Если сеть интегрируется с внешней VLAN, рассмотрите возможность добавления номера идентификатора VLAN к имени. Например, если веб-серверы HTTP размещаются в подсети webservers_122, то тегом VLAN будет 122. Или можно использовать internal-only, если нужно сохранить конфиденциальность сетевого трафика и не интегрировать сеть с внешней сетью
Admin State (Состояние)	Контролирует, доступна ли сеть в любой момент. Используйте это поле для создания сети в состоянии **Down** («Отключено»), когда она логически присутствует, но неактивна. Это полезно, если не требуется сразу вводить сеть в производство
Create Subnet (Создать подсеть)	Определяет, нужно ли создавать подсеть. Например, возможно, создавать подсеть не нужно, если планируется сохранять эту сеть как заполнитель без сетевого подключения

3. Нажмите кнопку **Next** и укажите следующие значения на вкладке **Subnet**:

Поле	Описание
Subnet Name (имя подсети)	Введите описательное имя подсети
Network Address (Адрес сети)	Введите адрес в формате CIDR, который содержит диапазон IP-адресов и маску подсети в одном значении. Чтобы определить адрес, рассчитайте количество битов, замаскированных в маске подсети, и добавьте это значение к диапазону IP-адресов. Например, в маске подсети 255.255.255.0 – 24 бита маски. Чтобы использовать эту маску с диапазоном адресов IPv4 192.168.122.0, укажите адрес 192.168.122.0/24
IP Version (Версия IP)	Указывает версию интернет-протокола, где допустимые типы – IPv4 или IPv6. Диапазон IP-адресов в поле **Network Address** (Адрес сети) должен соответствовать выбранной версии
Gateway IP (IP шлюза)	IP-адрес интерфейса маршрутизатора шлюза по умолчанию. Этот адрес является маршрутом по умолчанию для маршрутизации любого трафика, предназначенного для внешнего расположения, и должен быть в диапазоне, указанном в поле **Network Address**. Например, если сетевой адрес CIDR – 192.168.122.0/24, то шлюзом по умолчанию, скорее всего, будет 192.168.122.1
Disable Gateway (Отключение шлюза)	Отключает передачу и изолирует подсеть

4. Нажмите **Next** и укажите параметры DHCP:
   - **Enable DHCP** – включает службы DHCP для этой подсети. DHCP можно использовать для автоматизации распространения настроек IP на инстансы пользователя.
   - **IPv6 Address** – режимы настройки. Если создается сеть IPv6, необходимо указать способ выделения адресов IPv6 и дополнительную информацию:

- **No Options Specified** – выберите эту опцию, если нужно установить IP-адреса вручную или используете для выделения адресов метод, не поддерживающий 1Stack.
- **SLAAC** (Stateless Address Autoconfiguration). Инстансы генерируют адреса IPv6 на основе сообщений Router Advertisement (RA), отправленных с маршрутизатора Networking. Используйте эту конфигурацию для создания подсети Networking с параметром **ra_mode**, установленным на slaac, и **address_mode**, установленным на slaac.
- **DHCPv6 stateful** – инстансы получают адреса IPv6, а также дополнительные параметры (например, DNS) от сервиса Networking DHCPv6. Используйте эту конфигурацию, чтобы создать подсеть с параметром **ra_mode**, установленным на **dhcpv6-stateful**, и **address_mode**, установленным на **dhcpv6-stateful**.
- **DHCPv6 stateless** – инстансы генерируют адреса IPv6 на основе сообщений Router Advertisement (RA), отправленных с маршрутизатора Networking. Дополнительные опции (например, DNS) выделяются из сервиса Networking DHCPv6. Используйте эту конфигурацию для создания подсети с параметром **ra_mode**, установленным на **dhcpv6-stateless**, и **address _mode**, установленным на **dhcpv6-stateless**.

- **Пулы распределения** – диапазон IP-адресов, которые будет назначать DHCP. Например, значение 192.168.22.100,192.168.22.150 рассматривает все исходящие адреса в этом диапазоне как доступные для выделения.
- **DNS Name Servers** – IP-адреса DNS-серверов, доступных в сети. DHCP распределяет эти адреса инстансам для разрешения имени.

  Опыт показывает, что не стоит размещать стратегические сервисы, такие как DNS, в облаке. Например, если в облаке размещен DNS и облако становится неработоспособным, DNS становится недоступным, а облачные компоненты не смогут выполнять поиск друг друга.

- **Host Routes** – статические маршруты хостов. Сначала укажите сеть назначения в формате CIDR, а затем – маршрут по умолчанию, который нужно использовать для маршрутизации (например, 192.168.23.0/24, 10.1.31.1). Укажите это значение, если требуется распределить статические маршруты по инстансам.

5. Нажмите **Create**.

Просматривать сеть целиком можно на вкладке **Networks**. Если нужно изменить какие-либо параметры, нажмите **Edit**. При создании инстансов можно настроить их на использование подсети, тогда они получат любые указанные параметры DHCP.

14.3 Работа с подсетями

Используйте подсети (subnets), чтобы предоставить инстансам сетевое соединение.

Каждый инстанс назначается подсети в процессе создания инстанса, поэтому важно учитывать правильное размещение инстансов, чтобы наилучшим образом учесть требования к их подключению.

Создавать подсети можно только в уже существующих сетях. Следует помнить, что сети проектов в Networking могут содержать несколько подсетей. Это нужно иметь в виду, если планируется разместить в одной сети совершенно разные системы и обеспечить определенную изоляцию между ними.

Например, можно указать, что в одной подсети присутствует только трафик веб-сервера, а трафик базы данных проходит через другую подсеть.

Подсети изолированы друг от друга, и трафик инстанса, которому требуется взаимодействовать с другой подсетью, должен направляться маршрутизатором. Таким образом можно уменьшить стевую задержку и нагрузку, сгруппировав в одной подсети системы, между которыми требуется большой объем трафика.

14.4 Создание подсети

Для создания подсети выполните следующие действия:

1. На панели управления выберите **Project** > **Network** > **Networks** и нажмите на имя сети в **Networks**.
2. Нажмите **Create Subnet** и укажите следующие значения:

Поле	**Описание**
Subnet Name (Имя подсети)	Описательное имя подсети
Network Address (Адрес сети)	Адрес в формате CIDR, включающий диапазон IP-адресов и маску подсети в одном значении. Чтобы определить адрес CIDR, рассчитайте количество битов, замаскированных в маске подсети, и добавьте это значение к диапазону IP-адресов. Например, маска подсети 255.255.255.0 содержит 24 маскированных бита. Чтобы использовать эту маску с диапазоном адресов IPv4 192.168.122.0, укажите адрес 192.168.122.0/24
IP Version (IP версии)	Версия интернет-протокола с допустимыми типами IPv4 или IPv6. Диапазон IP-адресов в поле **Network Address** «**Сетевой адрес**» должен соответствовать выбранной версии протокола
Gateway IP (IP шлюза)	IP-адрес интерфейса маршрутизатора шлюза по умолчанию. Этот адрес является маршрутом по умолчанию для маршрутизации любого трафика, направленного во внешнее местоложение, и должен находиться в диапазоне, указанном пользователем в поле **Network Address**. Например, если сетевой адрес CIDR – 192.168.122.0/24, то шлюзом по умолчанию, скорее всего, будет 192.168.122.1
Disable Gateway (Отключение шлюза)	Отключает пересылку и изолирует подсеть

3. Нажмите **Next**, чтобы указать опции **DHCP**:

- **Enable DHCP** – включает службы DHCP для этой подсети. DHCP можно использовать для автоматизации распространения настроек IP на инстансы пользователя.

- **IPv6 Address** – режимы конфигурации. При создании сети IPv6 необходимо указать способ выделения адресов IPv6 и дополнительную информацию:
  - **No Options Specified** – выберите эту опцию, если нужно установить IP-адреса вручную или используете для выделения адресов метод, не поддерживающий 1Stack.
  - **SLAAC** (Stateless Address Autoconfiguration). Инстансы генерируют адреса IPv6 на основе сообщений Router Advertisement (RA), отправленных с маршрутизатора Networking. Используйте эту конфигурацию для создания подсети Networking с параметром **ra_mode**, установленным на slaac, и **address_mode**, установленным на slaac.
  - **DHCPv6 stateful** – инстансы получают адреса IPv6, а также дополнительные параметры (например, DNS) от сервиса Networking DHCPv6. Используйте эту конфигурацию, чтобы создать подсеть с параметром **ra_mode**, установленным на **dhcpv6-stateful**, и **address_mode**, установленным на **dhcpv6-stateful**.
  - **DHCPv6 stateless** – инстансы генерируют адреса IPv6 на основе сообщений Router Advertisement (RA), отправленных с маршрутизатора Networking. Дополнительные опции (например, DNS) выделяются из сервиса Networking DHCPv6. Используйте эту конфигурацию для создания подсети с параметром **ra_mode**, установленным на **dhcpv6-stateless**, и **address _mode**, установленным на **dhcpv6-stateless**.
- **Allocation Pools** – диапазон IP-адресов, которые будет назначать DHCP. Например, значение 192.168.22.100,192.168.22.150 указывает на то, что все исходящие адреса в этом диапазоне доступны для выделения.
- **DNS Name Servers** – IP-адреса DNS-серверов, доступных в сети. DHCP распределяет эти адреса по инстансам для разрешения имен.
- **Host Routes** – статические маршруты хостов. Сначала укажите сеть назначения в формате CIDR, а затем – маршрут по умолчанию, который нужно использовать для маршрутизации (например, 192.168.23.0/24, 10.1.31.1). Укажите это значение, если требуется распределить статические маршруты по инстансам.

4. Нажмите **Create**.

Просмотреть подсеть можно в списке **Subnets**. Если нужно изменить какие-либо параметры, нажмите **Edit**. При создании инстансов можно настроить их на использование подсети, тогда они получат любые указанные параметры DHCP.

14.5 Добавление маршрутизатора

Сервис Networking предоставляет маршрутизацию с использованием виртуального маршрутизатора (router) на основе SDN.

Маршрутизаторы необходимы для связи инстансов с внешними подсетями, в том числе в физической сети. Маршрутизаторы и подсети подключаются с помощью интерфейсов, для каждой подсети требуется собственный интерфейс к маршрутизатору.

Шлюз маршрутизатора по умолчанию определяет маршрут по умолчанию для любого трафика, получаемого маршрутизатором. Его сеть обычно настроена на маршрутизацию трафика во внешнюю физическую сеть с использованием виртуального моста.

Чтобы создать маршрутизатор:

1. На панели управления выберите **Project** > **Network** > **Routers** и нажмите **Create Router**.
2. Введите описательное имя нового роутера и нажмите **Create router**.
3. Нажмите **Set Gateway** рядом с записью нового маршрутизатора в списке **Routers.**
4. В списке **External Network** укажите сеть, которая должна получать трафик, предназначенный для внешнего размещения.
5. Нажмите **Set Gateway**.

После добавления маршрутизатора необходимо настроить все созданные подсети для отправки трафика через этот маршрутизатор. Для этого создайте интерфейсы между подсетью и маршрутизатором.

Маршруты по умолчанию для подсетей не должны перезаписываться.
Когда маршрут по умолчанию для подсети удаляется, агент L3 автоматически удаляет соответствующий маршрут в пространстве имен маршрутизатора, а сетевой трафик не может передаваться в связанную подсеть и из нее.
Если существующий маршрут пространства имен маршрутизатора был удален, чтобы устранить эту проблему, выполните следующие действия:

1) Отвяжите все плавающие IP-адреса в подсети.
2) Отключите маршрутизатор от подсети.
3) Повторно подключите маршрутизатор к подсети.
4) Повторно подключите все плавающие IP-адреса.

14.6 Очистка всех ресурсов и удаление проекта

Используйте команду **openstack project purge** для удаления всех ресурсов, принадлежащих конкретному проекту, а также для удаления проекта.

Например, чтобы очистить ресурсы проекта **test-project**, а затем удалить проект, укажите:

```
# openstack project list
+----------------------------------+--------------+
| ID                               | Name         |
+----------------------------------+--------------+
| 02e501908c5b438dbc73536c10c9aac0 | test-project |
| 519e6344f82e4c079c8e2eabb690023b | services     |
| 80bf5732752a41128e612fe615c886c6 | demo         |
| 98a2f53c20ce4d50a40dac4a38016c69 | admin        |
+----------------------------------+--------------+
# openstack project purge --project 02e501908c5b438dbc73536c10c9aac0
```

14.7 Удаление маршрутизатора

Маршрутизатор можно удалить при отсутствии подключенных к нему интерфейсов. Чтобы удалить маршрутизатор и его интерфейсы:

1. На панели управления выберите **Project** > **Network** > **Routers** и нажмите на имя маршрутизатора, который нужно удалить.
2. Выберите интерфейсы типа **Internal Interface** и нажмите **Delete Interfaces**.
3. В списке маршрутизаторов **Routers** выберите маршрутизатор и нажмите **Delete Routers**.

14.8 Удаление подсети

Подсеть, которая больше не используется, можно удалить. Однако если какие-либо инстансы по-прежнему настроены на использование подсети, удаление не будет выполнено, а на панели управления появится сообщение об ошибке.

Для удаления определенной подсети из сети выполните следующие действия:

1. На панели управления выберите **Project** > **Network** > **Networks**.
2. Нажмите на имя сети.
3. Выберите подсеть и нажмите **Delete Subnets**.

14.9 Удаление сети

В некоторых случаях возникает необходимость удалить ранее созданную сеть, например, при обслуживании или в процессе вывода из эксплуатации. Предварительно необходимо удалить или отключить все интерфейсы, где сеть все еще используется.

Чтобы удалить сеть в проекте вместе со всеми зависимыми интерфейсами, выполните следующие действия:

1. На панели управления выберите **Project** > **Network** > **Networks**.
2. Удалите все интерфейсы маршрутизатора, связанные с подсетями целевой сети. Чтобы удалить интерфейс, найдите ID-номер удаляемой сети – для этого нажмите на ее имя в списке сетей и посмотрите поле идентификатора. Во всех связанных с сетью подсетях это значение будет отображено в поле **Network ID**.
3. Зайдите в **Project** > **Network** > **Routers**, нажмите на имя виртуального маршрутизатора в списке маршрутизаторов **Routers** и найдите интерфейс, связанный с удаляемой подсетью.

Отличить эту подсеть от других подсетей можно по IP-адресу, который использовался в качестве IP-адреса шлюза. Дополнительно можно проверить отличие, убедившись, что сетевой идентификатор интерфейса совпадает с идентификатором, который отмечен на предыдущем шаге.

4. Нажмите на кнопку **Delete Interface** удаляемого интерфейса.
5. Выберите **Project** > **Network** > **Networks** и нажмите на имя сети.
6. Нажмите на кнопку **Delete Subnet** подсети, которую нужно удалить.

> Если на этом этапе по-прежнему не удается удалить подсеть, убедитесь, что она уже не используется каким-либо инстансом/инстансами.

7. Выберите **Project** > **Network** > **Networks** и сеть, которую нужно удалить.
8. Нажмите **Delete Networks**.

15 Настройка параметров максимального модуля передачи данных (MTU)

15.1 Обзор максимального модуля передачи данных (MTU)

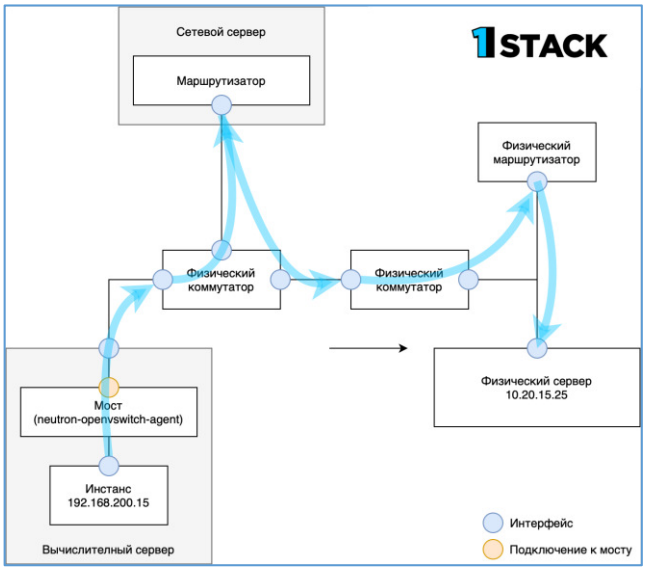
Сервис Networking может рассчитать максимально возможный размер максимальной единицы передачи (Maximum Transmission Unit, MTU), который будет гарантированно применяться к инстансам.

Значение MTU обозначает максимальный объем данных, который может передать один сетевой пакет; это число является переменным в зависимости от наиболее подходящего размера для приложения. Например, для общих ресурсов NFS может потребоваться иной размер MTU, чем для приложения VoIP.

Для просмотра максимально возможных значений MTU, которые рассчитывает Networking можно использовать команду **openstack network show <network_name>**.
net-mtu – это расширение Neutron API, которого нет в некоторых реализациях.
Требуемое значение MTU может быть анонсировано клиентам DHCPv4 для автоматической настройки, если это поддерживается инстансом, а также клиентам IPv6 через пакеты Router Advertisement (RA).
Для отправки сообщений маршрутизатора сеть должна быть подключена к маршрутизатору.

Необходимо последовательно настроить параметры MTU от начала до конца. Это означает, что установочный параметр MTU должен быть одинаковым в каждой точке прохождения пакета, включая виртуальную машину, инфраструктуру виртуальной сети, физическую сеть и целевой сервер.

На приведенной ниже диаграмме кружки обозначают различные точки, где значение MTU необходимо корректировать для трафика между инстансом и физическим сервером. Нужно изменить значение MTU для того интерфейса, который обрабатывает сетевой трафик, чтобы разместить пакеты определенного размера MTU. Это необходимо, если трафик проходит от инстанса 192.168.200.15 к физическому серверу 10.20.15.25:



Несогласованные значения MTU могут привести к ряду проблем в сети, чаще всего – к случайной потере пакетов, в результате которой произойдет разрыв соединения и снизится производительность сети.

Подобные проблемы устранить сложно, поскольку необходимо идентифицировать и проверить каждую возможную точку сети и убедиться, что она имеет правильное значение MTU.

15.2 Настройка параметров максимального модуля передачи данных в Undercloud

В этом примере показано, как настроить MTU при помощи шаблонов конфигурации NIC. Необходимо установить MTU на мосту, соединении (если применимо), интерфейсе (ах) и VLAN-сети (сетях):

```
- type: ovs_bridge
  name: br-isolated
  use_dhcp: false
  mtu: 9000    # <--- Set MTU
  members:
    - type: ovs_bond
      name: bond1
      mtu: 9000    # <--- Set MTU
      ovs_options: {get_param: BondInterfaceOvsOptions}
      members:
        - type: interface
          name: ens15f0
          mtu: 9000    # <--- Set MTU
          primary: true
        - type: interface
          name: enp131s0f0
          mtu: 9000    # <--- Set MTU
    - type: vlan
      device: bond1
      vlan_id: {get_param: InternalApiNetworkVlanID}
      mtu: 9000    # <--- Set MTU
      addresses:
      - ip_netmask: {get_param: InternalApiIpSubnet}
    - type: vlan
      device: bond1
      mtu: 9000    # <--- Set MTU
      vlan_id: {get_param: TenantNetworkVlanID}
      addresses:
      - ip_netmask: {get_param: TenantIpSubnet}
```

15.3 Проверка полученного MTU

Пользователь может просмотреть рассчитанное значение максимального модуля передачи данных, которое является максимально возможным значением MTU для инстансов. Используйте рассчитанное значение MTU для настройки всех интерфейсов, участвующих в пути сетевого трафика:

```
# openstack network show <network>
```

16 Настройка политик RBAC в Networking

16.1 Обзор политик RBAC

Политики управления доступом на основе ролей (Role-based access control, RBAC) в Networking обеспечивают детальное управление сетями совместного использования. В Networking задействована таблица RBAC для управления совместным использованием сетей между проектами. Это позволяет администратору контролировать, каким проектам предоставляется разрешение на подключение инстансов к сети.

В результате администраторы облака могут запретить некоторым проектам создавать сети и разрешить им подключаться к уже существующим сетям, соответствующим их проекту.

16.2 Создание политик RBAC

В примере ниже показан порядок действий при работе с политикой управления доступом на основе ролей (RBAC) для предоставления проекту доступа к сети совместного использования:

1. Просмотрите список доступных сетей:

```
# openstack network list
+--------------------------------------+-------------+--------------------------------
----------------------+
| id                                   | name        | subnets
|
+--------------------------------------+-------------+--------------------------------
----------------------+
| fa9bb72f-b81a-4572-9c7f-7237e5fcabd3 | web-servers | 20512ffe-ad56-4bb4-b064-
2cb18fecc923 192.168.200.0/24 |
| bcc16b34-e33e-445b-9fde-dd491817a48a | private     | 7fe4a05a-4b81-4a59-8c47-
82c965b0e050 10.0.0.0/24      |
| 9b2f4feb-fee8-43da-bb99-032e4aaf3f85 | public      | 2318dc3b-cff0-43fc-9489-
7d4cf48aaab9 172.24.4.224/28  |
+--------------------------------------+-------------+--------------------------------
----------------------+
```

2. Просмотрите список проектов:

```
# openstack project list
+----------------------------------+----------+
| ID                               | Name     |
+----------------------------------+----------+
| 4b0b98f8c6c040f38ba4f7146e8680f5 | auditors |
| 519e6344f82e4c079c8e2eabb690023b | services |
| 80bf5732752a41128e612fe615c886c6 | demo     |
| 98a2f53c20ce4d50a40dac4a38016c69 | admin    |
+----------------------------------+----------+
```

3. Создайте запись RBAC для сети веб-серверов, предоставляющую доступ к проекту auditors (4b0b98f8c6c040f38ba4f7146e8680f5):

```
# openstack network rbac create --type network --target-project
4b0b98f8c6c040f38ba4f7146e8680f5 --action access_as_shared web-servers
Created a new rbac_policy:
+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| action         | access_as_shared                     |
| id             | 314004d0-2261-4d5e-bda7-0181fcf40709 |
| object_id      | fa9bb72f-b81a-4572-9c7f-7237e5fcabd3 |
| object_type    | network                              |
| target_project | 4b0b98f8c6c040f38ba4f7146e8680f5     |
| project_id     | 98a2f53c20ce4d50a40dac4a38016c69     |
+----------------+--------------------------------------+
```

В результате пользователи проекта auditors смогут подключать инстансы к сети web-servers.

16.3 Проверка политик RBAC

Выполните следующие действия:

1. Запустите команду **openstack network rbac**, чтобы получить ID существующих политик управления доступом на основе ролей (RBAC):

```
# openstack network list
+--------------------------------------+-------------+--------------------------------
----------------------+
| id                                   | name        | subnets
|
+--------------------------------------+-------------+--------------------------------
----------------------+
| fa9bb72f-b81a-4572-9c7f-7237e5fcabd3 | web-servers | 20512ffe-ad56-4bb4-b064-
2cb18fecc923 192.168.200.0/24 |
| bcc16b34-e33e-445b-9fde-dd491817a48a | private     | 7fe4a05a-4b81-4a59-8c47-
82c965b0e050 10.0.0.0/24      |
| 9b2f4feb-fee8-43da-bb99-032e4aaf3f85 | public      | 2318dc3b-cff0-43fc-9489-
7d4cf48aaab9 172.24.4.224/28  |
+--------------------------------------+-------------+--------------------------------
----------------------+
```

2. Запустите команду **openstack network rbac-show**, чтобы посмотреть детали конкретной записи RBAC:

```
# openstack project list
+----------------------------------+----------+
| ID                               | Name     |
+----------------------------------+----------+
| 4b0b98f8c6c040f38ba4f7146e8680f5 | auditors |
| 519e6344f82e4c079c8e2eabb690023b | services |
| 80bf5732752a41128e612fe615c886c6 | demo     |
| 98a2f53c20ce4d50a40dac4a38016c69 | admin    |
+----------------------------------+----------+
```

16.4 Удаление политик RBAC

Выполните следующие действия:

1. Запустите команду **openstack network rbac list**, чтобы получить ID существующих политик управления доступом на основе ролей (RBAC):

```
# openstack network rbac list
+--------------------------------------+-------------+--------------------------------
-----+
| id                                   | object_type | object_id
|
+--------------------------------------+-------------+--------------------------------
-----+
| 314004d0-2261-4d5e-bda7-0181fcf40709 | network     | fa9bb72f-b81a-4572-9c7f-
7237e5fcabd3 |
| bbab1cf9-edc5-47f9-aee3-a413bd582c0a | network     | 9b2f4feb-fee8-43da-bb99-
032e4aaf3f85 |
+--------------------------------------+-------------+--------------------------------
-----+
```

2. Запустите команду **openstack network rbac delete**, чтобы удалить RBAC, используя его идентификатор:

```
# openstack network rbac delete 314004d0-2261-4d5e-bda7-0181fcf40709
Deleted rbac_policy: 314004d0-2261-4d5e-bda7-0181fcf40709
```

16.5 Предоставление доступа к политике RBAC внешним сетям

Администратор может предоставить доступ к политике управления доступом на основе ролей (RBAC) внешним сетям (с подключенными интерфейсами шлюза), используя параметр **--action access_as_external**.

Выполните действия, описанные далее, чтобы создать RBAC для сети веб-серверов и предоставить доступ к проекту engineering project (c717f263785d4679b16a122516247deb):

- Создайте новую политику RBAC при помощи опции **--action access_as_external**:

```
# openstack network rbac create --type network --target-project
c717f263785d4679b16a122516247deb --action access_as_external web-servers
 Created a new rbac_policy:
+----------------+--------------------------------------+
| Field          | Value                                |
+----------------+--------------------------------------+
| action         | access_as_external                   |
| id             | ddef112a-c092-4ac1-8914-c714a3d3ba08 |
| object_id      | 6e437ff0-d20f-4483-b627-c3749399bdca |
| object_type    | network                              |
| target_project | c717f263785d4679b16a122516247deb     |
| project_id     | c717f263785d4679b16a122516247deb     |
+----------------+--------------------------------------+
```

В результате пользователи проекта engineering смогут просматривать сеть или подключать к ней инстансы:

```
$ openstack network list
+--------------------------------------+-------------+--------------------------------
--------------------+
| id                                   | name        | subnets
|
+--------------------------------------+-------------+--------------------------------
--------------------+
| 6e437ff0-d20f-4483-b627-c3749399bdca | web-servers | fa273245-1eff-4830-b40c-
57eaeac9b904 192.168.10.0/24 |
+--------------------------------------+-------------+--------------------------------
--------------------+
```

17 Настройка разрешенных пар адресов

17.1 Обзор разрешенных пар адресов

Разрешенная пара адресов (allowed address pair) – это результат действий администратора, когда он определяет конкретный MAC-адрес, IP-адрес или их оба, и разрешает сетевому трафику проходить через порт независимо от подсети.

Определив разрешенные пары адресов, можно использовать такие протоколы, как протокол резервирования виртуального маршрутизатора (Virtual Router Redundancy Protocol, VRRP), которые перемещают IP-адрес между двумя инстансами виртуальных машин, чтобы обеспечить быструю обработку отказа.

Разрешенные пары адресов определяются с помощью команды командной строки **openstack port** клиента 1Stack.

Имейте в виду, что не следует использовать группу безопасности по умолчанию с более широким диапазоном IP-адресов в разрешенной паре адресов. Это может создать возможность для одного порта обходить группы безопасности для всех остальных портов этой же сети.
Например, эта команда затрагивает все порты сети и обходит все группы безопасности:
openstack port set --allowed-address mac-address=3e:37:09:4b,ip-address=0.0.0.0/0
9e67d44eab334f07bf82fa1b17d824b6

С помощью сетевого драйвера механизма ML2/OVN можно создавать VIP-адреса. Однако IP-адрес, назначенный связанному порту с помощью allowed_address_pairs, должен совпдать с IP-адресом виртуального порта (/32).
Если же используется IP-адрес в формате CIDR для **allowed_address_pairs** привязанного порта, переадресация портов не настраивается на стороне сервера, и происходит сбой трафика для любого IP-адреса в CIDR, ожидающего достижения связанного IP-порта.

17.2 Создание порта и разрешение одной пары адресов

Создание порта с разрешенной парой адресов позволяет сетевому трафику проходить через порт независимо от подсети.

Не используйте группу безопасности по умолчанию с более широким диапазоном IP-адресов в разрешенной паре адресов. Это может позволить одному порту обходить группы безопасности для всех остальных портов в той же сети.

Порядок действий

- Создайте порт и разрешите одну пару адресов при помощи команды:

```
# openstack port create --network <network> --allowed-address mac-address=<mac-address>,ip-address=<ip_cidr> <port_name>
```

17.3 Добавление разрешенных пар адресов

Чтобы сетевой трафик мог проходить через порт независимо от подсети, можно добавить на порт разрешенную пару адресов.

Не используйте группу безопасности по умолчанию с более широким диапазоном IP-адресов в разрешенной паре адресов. Это может позволить одному порту обходить группы безопасности для всех остальных портов в той же сети.

Порядок действий

- Для добавления разрешенных пар адресов используйте следующую команду:

```
openstack port set --allowed-address mac-address=<mac_address>,ip-address=<ip_cidr>
<port>
```

Нельзя установить пару allowed-address, соответствующую mac_address и ip_address порта. Это связано с тем, что такая настройка не имеет эффекта, поскольку трафику, соответствующему mac_address и ip_address, уже разрешено проходить через порт.

18 Управление образами

Сервис Image (glance) обеспечивает сервисы обнаружения, регистрации и сохранения образов (images) дисков и серверов. Он предоставляет возможность копировать или делать снимки образа сервера и мгновенно сохранять их.

Сохраненные образы можно использовать в качестве шаблона для быстрого ввода в эксплуатацию новых серверов в сравнении с установкой серверной операционной системы и индивидуальной настройкой сервисов.

18.1 Загрузка образа

1. На панели управления выберите **Project** > **Compute** > **Images**.
2. Нажмите **Create Image**.
3. Укажите значения и нажмите **Create Image**.

Опции образа

Поле	Описание
Name	Имя образа. В рамках проекта имя должно быть уникальным
Description	Краткое описание образа
Image Source	Источник образа: **Image Location** или **Image File**. Следующее поле отображается в зависимости от выбора
Image Location или **Image File**	▪ Выберите опцию **Image Location**, чтобы указать URL расположения образа. ▪ Выберите опцию **Image File**, чтобы загрузить образ с локального диска
Format	Формат образа (например, **qcow2**)
Architecture	Архитектура образа. Например, используйте i686 для архитектуры 32 бит или x86_64 для архитектуры 64 бит
Minimum Disk (GB)	Минимальный размер диска, необходимый для загрузки образа. Если в этом поле ничего не указано, значение по умолчанию – 0 (минимум отсутствует)
Minimum RAM (MB)	Минимальный размер памяти, необходимый для загрузки образа. Если в этом поле ничего не указано, значение по умолчанию – 0 (минимум отсутствует)
Public	При выборе этой опции образ становится общедоступным для всех пользователей, имеющих доступ к проекту
Protected	Этот выбор разрешает удалять этот образ только пользователям с определенными разрешениями

Если загрузка образа прошла успешно, его статус принимает значение **active**. т. е. образ становится доступным для использования.

Обратите внимание, что сервис **Image** может работать даже с крупными образами, загрузка которых занимает много времени – больше, чем время жизненного цикла токена сервиса **Identity**, который использовался при инициализации загрузки. Это связано с тем, что сервис **Image** сначала создает доверенное соединение с сервисом **Identity** таким образом, что новый токен можно получить и использовать после завершения загрузки и обновления статуса образа.

Можно также использовать команду **glance image-create** с опцией **--property** для загрузки образа.
В командной строке доступны также дополнительные значения.

18.2 Обновление образа

1. На панели управления выберите **Project** > **Compute** > **Images**.
2. Нажмите в списке на **Edit Image**.

Параметр **Edit Image** доступен только при авторизации пользователя в качестве администратора.
При входе в систему в качестве демопользователя доступны опции **Launch an instance** или **Create Volume**.

3. Обновите поля и нажмите **Update Image**. Можно обновить следующие значения:

- имя, описание,
- ID ядра,
- ID виртуального диска,
- архитектура,
- формат,
- минимальный диск,
- минимальный объем оперативной памяти,
- общедоступный, защищенный.

4. Нажмите на раскрывающееся меню и выберите опцию **Update Metadata**.
5. В левом столбце приведены определения метаданных из каталога метаданных сервиса **Image** (**Image Service Metadata Catalog**). Выберите **Other**, чтобы добавить метаданные с выбранным ключом, и нажмите **Save**.
6. Укажите метаданные – добавьте элементы из левого столбца в правый столбец.

Можно также использовать команду **glance image-update** с опцией **--property** для обновления образа.
Дополнительные значения доступны в командной строке.

18.3 Импорт образа

Образы можно импортировать в сервис **Image** (glance) одним из двух методов:

1. При помощи **web-download** для импорта орбраза с URI.
2. При помощи **glance-direct** для импорта образ из локальной файловой системы.

По умолчанию включен метод **web-download**.

Методы импорта настраивает администратор облака. Пользователь мржет запустить команду **glance import-info**, чтобы просмотреть списк доступных опций импорта.

18.3.1 Импорт образа с удаленной URI

Для копирования образа с удаленной URI можно использовать метод **web-download**.

1. Создайте образ и укажите URI образа для импорта:

```
$ openstack image create --container-format <container-format> --disk-format <disk-
format> \
  --size <size> --location <image-url> <image-name>
```

- Поменяйте <CONTAINER FORMAT> на формат контейнера, который нужно установить для образа (**none, ami, ari, aki, bare, ovf, ova, docker**).
- Поменяйте <DISK-FORMAT> на формат диска, который нужно установить для образа (**none, ami, ari, aki, vhd, vhdx, vmdk, raw, qcow2, vdi, iso, ploop**).
- Поменяйте <IMAGE-NAME> на описательное имя образа.
- Поменяйте <URI> на URI образа.

2. Проверить доступность образа можно с помощью команды **openstack image show <IMAGE_ID>**.

- Поменяйте <IMAGE_ID> на ID, который был указан при создании образа.

Метод веб-загрузки (web download) сервиса **Image** использует двухэтапный процесс импорта. Этот метод:

- создает запись образа;
- извлекает образ по указанной URI.

Опционально: возможна фильтрация URI по спискам запрещенных и разрешенных ссылок.

Плагин **Image Property Injection** может внедрять свойства метаданных в образ. Внедренные свойства определяют, на каких вычислительных серверах запущены инстансы образа.

18.3.2 Импорт образа с локального диска

Метод **glance-direct** создает запись образа, которая генерирует его ID.

После загрузки образа в сервис **Image** с локального диска он сохраняется в промежуточной области и становится активным после прохождения всех настроенных проверок.

Для метода **glance-direct** необходима промежуточная область совместного использования при работе в конфигурации высокой доступности (high-availability, HA).

Загрузка образа методом **glance-direct** может завершиться неуспешно в среде высокой доступности (HA), если отсутствует общая области подготовки.
В среде HA active-active («активный-активный») вызовы API распределяются по серверам управления сервиса Image.
Вызов API загрузки может быть отправлен на сервер управления, отличный от вызова API для загрузки образа.

Для импорта образа из локального источника можно использовать команду **openstack image create**:

```
$ openstack image create --container-format <container-format> --disk-format <disk-format> \
  --size <size> --file <file> <image-name>
```

- Поменяйте <CONTAINER FORMAT>, <DISK-FORMAT>, <NAME> и <FILE> на значения, релевантные для образа.

После перемещения образа из промежуточной области в основное хранилище образ появится в списке. Однако для того, чтобы изображение стало активным, может потребоваться некоторое время.

Проверить доступность образа можно с помощью команды **openstack image show <IMAGE_ID>**.

- Поменяйте <IMAGE_ID> на ID, который использовался при создании образа.

18.4 Удаление образа

Выполните следующие действия:

1. На панели управления выберите **Project** > **Compute** > **Images**.
2. Выберите образ и нажмите **Delete Images**.

19 Управление томами

Облачным хранилищем можно управлять с помощью панели управления 1Stack или клиентов командной строки: большинство процедур можно выполнять, используя любой из этих методов. Однако некоторые более сложные действия можно выполнить только в командной строке.

Создайте и настройте тома как основной тип постоянного хранилища для инстансов сервиса вычислительных ресурсов (Compute) в облачном хранилище. Создавайте тома, присоединяйте тома к инстансам, редактируйте тома и изменяйте их размеры, а также изменяйте владельцев томов.

19.1 Создание томов

Создавайте тома, чтобы обеспечить постоянное хранилище для инстансов, которые нужно запускать при помощи сервиса вычислительных ресурсов (nova) в **Overcloud**.

Максимальное количество томов, которое можно создать для проекта по умолчанию – 10.

Предварительные требования

- Установка **Undercloud** выполнена успешно.
- Развертывание **Overcloud** выполнено успешно.
- Имеется доступ к панели управления (horizon) 1Stack.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Volumes**.
2. Нажмите **Create Volume** и внесите изменения в следующие поля:

Поле	Описание
Volume name	Имя тома
Description	Опционально, короткое описание тома
Type	Если имеется несколько хранилищ Block Storage, для этого поля можно выбрать конкретное хранилище
Size (GB)	Размер тома (в гигабайтах). Если нужно создать зашифрованный том из незашифрованного образа, необходимо убедиться, что размер тома больше размера образа, чтобы данные шифрования не урезали данные тома
Availability Zone	Зоны доступности (группы логических серверов), наряду с агрегатами хостов, являются распространенным методом разделения (сегрегации) ресурсов в 1Stack. Зоны доступности определяются в процессе установки

3. Укажите источник тома:

Источник	Описание
No source, empty volume	Том пустой и не содержит файловой системы или таблицы разделов

Источник	Описание
Snapshot	Использовать существующий снимок в качестве источника тома. При выборе этой опции откроется новый список **Use snapshot as a source** (Использовать снимок в качестве источника), снимок можно будет выбрать в списке. Если нужно создать новый том из снимка зашифрованного тома, необходимо убедиться, что новый том больше старого не менее чем на 1 ГБ
Image	Использовать существующий образ в качестве источника тома. При выборе этой опции откроется новый список **Use snapshot as a source** (Использовать снимок в качестве источника), образ можно выбрать в списке
Volume	Использовать существующий том в качестве источника тома. При выборе этой опции откроется новый список **Use snapshot as a source**. В списке можно будет выбрать том

4. Нажмите **Create Volume**. После создания тома его имя появится в таблице **Volumes.**

   В дальнейшем можно будет изменить тип тома.

19.2 Редактирование имени или описания тома

Редактировать имена и описания можно из панели управления.

Предварительные требования

- Установка **Undercloud** выполнена успешно.
- Развертывание **Overcloud** выполнено успешно.
- Имеется доступ к панели управления 1Stack.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Volumes**.
2. Нажмите кнопку тома **Edit Volume**.
3. Внесите изменения в имя или описание тома.
4. Нажмите **Edit Volume**, чтобы сохранить изменения.

Чтобы создать зашифрованный том, сначала необходимо настроить тип тома специально для шифрования.
Помимо этого необходимо настроить оба сервиса (**Compute** и **Block Storage**) для использования одного и того же статического ключа.

19.3 Изменение размера (расширение) тома

Изменение размера томов увеличивает их емкость.

Возможность изменения размера используемого тома поддерживается, однако зависит от драйвера.
Поддерживается RBD. Нельзя расширить используемые тома с несколькими подключениями (multi-attach volumes).

Предварительные требования

- Установка **Undercloud** выполнена успешно.

Порядок действий

1. Получите список томов, чтобы получить идентификатор ID тома, который нужно расширить:

```
$ openstack volume list
```

2. Для изменения размера тома выполните следующие команды, чтобы указать корректную микроверсию API, затем передайте ID тома и новый размер (значение, превышающее предыдущее) в качестве параметров:

```
$ openstack volume set <volume ID> <size>
```

Поменяйте <volume ID>, and <size> на нужные значения. Например

```
$ openstack volume set 573e024d-5235-49ce-8332-be1576d323f8 --size 10
```

19.4 Удаление тома

Для удаления ненужных томов используйте панель управления.

> Том невозможно удалить при наличии у него существующих снимков.

Предварительные требования

- Установка **Undercloud** выполнена успешно.
- Развертывание **Overcloud** выполнено успешно.
- Имеется доступ к панели управления 1Stack.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Volumes**.
2. В таблице **Volumes** выберите тома, которые нужно удалить.
3. Нажмите **Delete Volumes**.

19.5 Подключение тома

Инстансы могут использовать том для постоянного хранения. Том можно подключить только к одному экземпляру единовременно.

Предварительные требования

- Установка **Undercloud** выполнена успешно.
- Развертывание **Overcloud** выполнено успешно.
- Имеется доступ к панели управления 1Stack.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Volumes**.
2. Выберите действие **Edit Attachments**. Если том не подключен к инстансу, отобразится раскрывающийся список **Attach To Instance**.
3. В списке **Attach To Instance** выберите инстанс, к которому нужно подключить том.
4. Нажмите Attach Volume.

19.6 Отключение тома

Инстансы могут использовать том для постоянного хранения. Том можно присоединить только к одному инстансу единовременно.

Предварительные требования

- Установка **Undercloud** выполнена успешно.
- Развертывание **Overcloud** выполнено успешно.
- Имеется доступ к панели управления 1Stack.

Порядок действий

1. На панели управления выберите **Project** > **Compute** > **Volumes**.
2. Выберите действие **Manage Attachments** (управление вложениями тома). Если том подключен к инстансу, имя инстанса отобразится в таблице **Attachments**.
3. Нажмите **Detach Volume** в этом и следующем диалоговом окне.

Масштабирование серверов Overcloud

Если после создания **Overcloud** нужно добавить или удалить серверы, необходимо обновить **Overcloud**.

Не используйте функцию удаления сервера **openstack** для удаления серверов из **Overcloud**. Следуйте инструкциям, приведенным в этом разделе, чтобы правильно удалять и заменять серверы.

Перед масштабированием или удалением сервера **Overcloud**, убедитесь, что физические серверы не находятся в режиме обслуживания.

Перед масштабированием **Overcloud**, убедитесь, что у имеется как минимум 10 ГБ свободного места. Это свободное пространство необходимо для преобразования и кеширования образов в процессе подготовки сервера.

Используйте информацию, приведенную в таблице ниже, чтобы определить поддержку масштабирования серверов каждого типа:

Тип сервера	Масштабировать с увеличением?	Масштабировать с уменьшением?	Примечание
Маршрутизатор	Нет	Нет	Можно заменить серверы управления, используя порядок действий, описанный в разделе Замена серверов управления
Compute	Да	Да	

20 Масштабирование серверов Overcloud

Если после создания **Overcloud** нужно добавить или удалить серверы, необходимо обновить **Overcloud**.

Не используйте функцию удаления сервера **openstack** для удаления серверов из **Overcloud**. Следуйте инструкциям, приведенным в этом разделе, чтобы правильно удалять и заменять серверы.

Перед масштабированием или удалением сервера **Overcloud**, убедитесь, что физические серверы не находятся в режиме обслуживания.

Перед масштабированием **Overcloud**, убедитесь, что у имеется как минимум 10 ГБ свободного места. Это свободное пространство необходимо для преобразования и кеширования образов в процессе подготовки сервера.

Используйте информацию, приведенную в таблице ниже, чтобы определить поддержку масштабирования серверов каждого типа:

Тип сервера	Масштабировать с увеличением?	Масштабировать с уменьшением?	Примечание
Маршрутизатор	Нет	Нет	Можно заменить серверы управления, используя порядок действий, описанный в разделе Замена серверов управления
Compute	Да	Да	

20.1 Добавление серверов в Overcloud

Выполните следующие действия, чтобы добавить серверы в пул серверов **Undercloud**.

Порядок действий

1. Создайте новый JSON-файл с названием **newnodes.json**, содержащий сведения о новом сервере, который нужно зарегистрировать:

```
{
  "nodes":[
    {
        "mac":[
            "dd:dd:dd:dd:dd:dd"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
```

```
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.168.24.207"
    },
    {
        "mac":[
            "ee:ee:ee:ee:ee:ee"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.168.24.208"
    }
  ]
}
```

2. Зарегистрируйте новые серверы:

```
$ source ~/stackrc
(undercloud) $ openstack overcloud node import newnodes.json
```

3. После регистрации новых серверов запустите процесс интроспекции для каждого из них:

```
(undercloud) $ openstack overcloud node introspect [NODE UUID] --provide
```

Этот процесс определяет и оценивает аппаратные характеристики серверов.

4. Настройте свойства образа для сервера:

```
(undercloud) $ openstack overcloud node configure [NODE UUID]
```

20.2 Увеличение количества серверов для ролей

Чтобы масштабировать серверы **Overcloud** для определенной роли, например, вычислительного сервера, выполните следующие действия.

Порядок действий

1. Пометьте каждый новый сервер меткой (tag) с нужной ролью. Например, чтобы отметить сервер ролью **Compute**, укажите следующую команду:

```
(undercloud) $ openstack baremetal node set --property
capabilities='profile:compute,boot_option:local' [NODE UUID]
```

2. Чтобы масштабировать **Overcloud**, необходимо отредактировать файл среды, содержащий реальное количество серверов, и повторно развернуть **Overcloud**. Например, чтобы масштабировать **Overcloud** до пяти вычислительных серверов, отредактируйте параметр ComputeCount:

```
parameter_defaults:
  ...
  ComputeCount: 5
  ...
```

3. Повторно запустите команду развертывания с обновленным файлом (в примере имеет имя **node-info.yaml**):

```
(undercloud) $ openstack overcloud deploy --templates -e /home/stack/templates/node-info.yaml [OTHER_OPTIONS]
```

Убедитесь, что включены все файлы и параметры среды из изначального создания **Overcloud**. Сюда входят те же параметры масштабирования для серверов, которые не являются вычислительными серверами.

4. Дождитесь завершения операции развертывания.

20.3 Удаление или замена вычислительного сервера

В некоторых ситуациях необходимо удалить вычислительный сервер (Compute) из **Overcloud**, например, если нужно заменить проблемный вычислительный сервер.

При удалении вычислительного сервера его индекс добавляется по умолчанию в список запрещенных (denylist), чтобы предотвратить повторное использование индекса в операциях горизонтального масштабирования.

После удаления сервера из развертывания **Overcloud** удаленный вычислительный сервер можно заменить.

Предварительные требования

- Отключите вычислительный сервис на серверах, которые нужно удалить, чтобы сервис не планировал новые инстансы на этом сервере.
- Чтобы подтвердить, что вычислительный сервис отключен, используйте следующую команду:

```
(overcloud)$ openstack compute service list
```

- Если вычислительный сервис не отключен, отключите его:

```
(overcloud)$ openstack compute service set <hostname> nova-compute
   --disable
```

Используйте параметр **--disable-reason**, чтобы добавить краткое объяснение причины отключения сервиса. Это может понадобиться, если планируется повторное развертывание сервиса вычислительных ресурсов.

- Рабочие нагрузки на вычислительных серверах были перенесены на другие вычислительные серверы.
- Если включен инстанс высокой доступности, выберите один из следующих вариантов:
    1. Если вычислительный сервер доступен, авторизуйтесь на нем как пользователь **root** и полностью завершите работы с помощью команды reboot.
    2. Если вычислительный сервер недоступен, авторизуйтесь на сервере управления как пользователь **root**, отключите устройство STONITH для вычислительного сервера и завершите работу физического сервера:

[root@controller-0 ~]# pcs stonith disable <stonith_resource_name>
[stack@undercloud ~]$ source stackrc
[stack@undercloud ~]$ openstack baremetal node power off <UUID>

1. Активируйте исходную конфигурацию **Undercloud**:

```
(overcloud)$ source ~/stackrc
```

2. Определите UUID набора ресурсов в **Overcloud**:

```
(undercloud)$ openstack stack list
```

3. Определите UUID или имена хостов вычислительных серверов, которые нужно удалить:

```
(undercloud)$ openstack server list
```

4. Опционально: запустите команду развертывания **Overcloud** с параметром **--update-plan-only**, чтобы обновить планы при помощи последних конфигураций из шаблонов. Это гарантирует актуальность конфигурации **Overcloud** перед удалением любых вычислительных серверов:

```
$ openstack overcloud deploy --update-plan-only --templates  \
  -e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
  -e /home/stack/templates/network-environment.yaml \
  -e /home/stack/templates/storage-environment.yaml \
  -e /home/stack/templates/rhel-registration/environment-rhel-registration.yaml \
  [-e |...]
```

Этот шаг необходим, если список запрещенных серверов **Overcloud** был обновлен.

5. Удалите вычислительные серверы из stack:

```
$ openstack overcloud node delete --stack <overcloud> <node_1> ... [node_n]
```

- Поменяйте <overcloud> на имя или UUID настроенного имени набора управляемых ресурсов (stack) **Overcloud**.
- Поменяйте <node_1> и опционально все серверы на [node_n] на имя хоста вычислительного сервиса или UUID вычислительных серверов, которые нужно удалить. Не используйте комбинацию UUID и имен хостов. Используйте только UUIDs либо только имена хостов.

Если сервер уже был выключен, эта команда возвращает сообщение о предупреждении (WARNING):

Ansible failed, check log at /var/lib/mistral/overcloud/ansible.log
WARNING: Scale-down configuration error. Manual cleanup of some actions may be necessary. Continuing with node removal.

Это сообщение можно проигнорировать.

6. Дождитесь удаления вычислительных серверов.
7. Удалите сетевые агенты для каждого удаленного сервера:

```
(overcloud)$ for AGENT in $(openstack network agent list --host <scaled_down_node> -c
ID -f value); do openstack network agent delete $AGENT ; done
```

8. Проверьте состояние набора ресурсов (stack) **Overcloud** после завершения удаления сервера:

```
(undercloud)$ openstack stack list
```

Результат:

Статус	Описание
UPDATE_COMPLETE	Операция удаления завершена успешно
UPDATE_FAILED	Операция удаления завершена неуспешно. Распространенная причина неудачной операции удаления – недоступный интерфейс IPMI на сервере, который нужно удалить. Если операция удаления завершается неудачно, необходимо удалить вычислительный сервер вручную

9. Если инстанс высокой доступности включен, выполните следующие действия:
   - Очистите ресурсы **Pacemaker** для сервера:

```
$ sudo pcs resource delete <scaled_down_node>
$ sudo cibadmin -o nodes --delete --xml-text '<node id="<scaled_down_node>"/>'
$ sudo cibadmin -o fencing-topology --delete --xml-text '<fencing-level
target="<scaled_down_node>"/>'
$ sudo cibadmin -o status --delete --xml-text '<node_state id="<scaled_down_node>"/>'
$ sudo cibadmin -o status --delete-all --xml-text '<node id="<scaled_down_node>"/>' --
force
```

   - Удалите устройство STONITH для сервера:

```
$ sudo pcs stonith delete <device-name>
```

10. Если замена удаленных вычислительных серверов в **Overcloud** не нужна, уменьшите параметр **ComputeCount** в файле среды с количеством серверов. Как правило, этот файл называется **node-info.yaml**. Например, уменьшите количество серверов с четырех до трех, если удален один сервер:

```
parameter_defaults:
  ...
  ComputeCount: 3
```

Уменьшение количества серверов гарантирует, что **Undercloud** не будет предоставлять новые серверы при запуске развертывания **openstack overcloud**.

20.4 Удаление вычислительного сервера вручную

Если команда удаления сервера **openstack overcloud** завершилась неудачно из-за недоступности сервера, необходимо завершить удаление вычислительного сервера из **Overcloud** вручную.

Предварительные требования

- Удаление или замена вычислительного сервера со статусом **UPDATE_FAILED**.

Порядок действий

1. Определите UUID набора ресурсов (stack) **Overcloud**:

```
(undercloud)$ openstack stack list
```

2. Определите UUID сервера, который нужно удалить вручную:

```
(undercloud)$ openstack baremetal node list
```

3. Переведите сервер, который нужно удалить, в режим обслуживания:

```
(undercloud)$ openstack baremetal node maintenance set <node_uuid>
```

4. Дождитесь синхронизации состояния вычислительного сервиса с сервисом Bare Metal. Это может занять до четырех минут.
5. Примените конфигурацию **Overcloud**:

```
(undercloud)$ source ~/overcloudrc
```

6. Удалите сетевые агенты для удаленного сервера:

```
(overcloud)$ for AGENT in $(openstack network agent list --host <scaled_down_node> -c
ID -f value) ; \do openstack network agent delete $AGENT ; done
```

Поменяйте <scaled_down_node> на имя сервера, который нужно удалить.

7. Убедитесь, что сервис вычислительных ресурсов (Compute) отключен на удаленном сервере в **Overcloud**, чтобы предотвратить внесение сервером новых инстансов в расписание:

```
(overcloud)$ openstack compute service list
```

8. Если сервис вычислительных ресурсов не отключен, отключите его:

```
(overcloud)$ openstack compute service set <hostname> nova-compute --disable
```

Используйте параметр **--disable-reason**, чтобы добавить краткое объяснение причины отключения сервиса. Это может понадобиться, если планируется повторное развертывание сервиса вычислительных ресурсов.

9. Удалите сервис вычислительных ресурсов (Compute) с сервера:

```
(overcloud)$ openstack compute service delete <service_id>
```

10. Удалите удаленный сервис вычислительных ресурсов в качестве поставщика ресурсов с сервиса Placement:

```
(overcloud)$ openstack resource provider list
(overcloud)$ openstack resource provider delete <uuid>
```

11. Примените конфигурацию **Undercloud**:

```
(overcloud)$ source ~/stackrc
```

12. Удалите сервер вычислительных ресурсов из stack:

```
(undercloud)$ openstack overcloud node delete --stack <overcloud> <node>
```

- Поменяйте <overcloud> на имя или UUID набора ресурсов (stack) в **Overcloud**.
- Поменяйте <node> на имя хоста сервиса вычислительных ресурсов или UUID сервера вычислительных ресурсов, который нужно удалить.

Если сервер уже был выключен, эта команда возвращает сообщение о предупреждении (WARNING):

Ansible failed, check log at /var/lib/mistral/overcloud/ansible.log
WARNING: Scale-down configuration error. Manual cleanup of some actions may be necessary. Continuing with node removal.

Это сообщение можно проигнорировать.

13. Дождитесь удаления сервера **Overcloud**.
14. Проверьте состояние набора ресурсов (stack) **Overcloud** после завершения удаления сервера:

```
(undercloud)$ openstack stack list
```

Результат:

Статус	Описание
UPDATE_COMPLETE	Операция удаления завершена успешно
UPDATE_FAILED	Операция удаления завершена неуспешно. Если сервер **Overcloud** не удается удалить в режиме обслуживания, проблема может быть связана с оборудованием

15. Если инстанс высокой доступности включен, выполните следующие действия:
    - Очистите ресурсы **Pacemaker** для сервера:

```
$ sudo pcs resource delete <scaled_down_node>
$ sudo cibadmin -o nodes --delete --xml-text '<node id="<scaled_down_node>"/>'
$ sudo cibadmin -o fencing-topology --delete --xml-text '<fencing-level
target="<scaled_down_node>"/>'
$ sudo cibadmin -o status --delete --xml-text '<node_state id="<scaled_down_node>"/>'
$ sudo cibadmin -o status --delete-all --xml-text '<node id="<scaled_down_node>"/>' -
force
```

    - Удалите устройство STONITH для сервера:

```
$ sudo pcs stonith delete <device-name>
```

16. Если замена удаленного сервера вычислительных ресурсов в **Overcloud** не нужна, уменьшите параметр **ComputeCount** в файле среды с количеством серверов. Как правило, этот файл называется **node-info.yaml**. Например, уменьшите количество серверов с четырех до трех, если удален один сервер:

```
parameter_defaults:
  ...
  ComputeCount: 3
  ...
```

Уменьшение количества серверов гарантирует, что **Undercloud** не будет предоставлять новые серверы при запуске развертывания **openstack overcloud**.

20.5 Замена удаленного сервера

Чтобы заменить удаленный вычислительный сервер при развертывании **Overcloud**, можно зарегистрировать и проверить новый сервер вычислительных ресурсов или повторно добавить удаленный. Необходимо также настроить **Overcloud** для предоставления сервера.

Порядок действий

1. Опционально: чтобы повторно использовать индекс удаленного сервера вычислительных ресурсов, настройте для роли параметры **RemovalPoliciesMode** и **RemovalPolicies** для замены запрещенного списка при удалении вычислительного сервера:

```
parameter_defaults:
  <RoleName>RemovalPoliciesMode: update
  <RoleName>RemovalPolicies: [{'resource_list': []}]
```

2. Поменяйте удаленный вычислительный сервер:
   - Чтобы добавить новый вычислительный сервер, зарегистрируйте, сделайте интроспекцию и отметьте новый сервер, чтобы подготовить его к инициализации.
   - Чтобы добавить сервер вычислительных ресурсов, который был удален вручную, выведите его из режима обслуживания:

```
(undercloud)$ openstack baremetal node maintenance unset
   <node_uuid>
```

3. Повторно запустите команду развертывания **openstack overcloud**, которая применялась для развертывания существующего **Overcloud**.
4. Дождитесь завершения процесса развертывания.
5. Подтвердите, что **Undercloud** успешно зарегистрировал новый сервер вычислительных ресурсов:

```
(undercloud)$ openstack baremetal node list
```

6. Если на шаге 1 были выполнены действия по установке режима **RemovalPoliciesMode** для обновляемой роли, то необходимо сбросить этот режим до значения по умолчанию (append), чтобы добавить индекс сервера вычислительных ресурсов в имеющийся список запретов при удалении сервера вычислительных ресурсов:

```
parameter_defaults:
  <RoleName>RemovalPoliciesMode: append
```

7. Повторно запустите команду развертывания **openstack overcloud**, которая использовалась для развертывания существующего **Overcloud**.

20.6 Сохранение имен хостов при замене серверов, использующих предсказуемые IP-адреса и HostNameMap

Если выполнена настройка **Overcloud** на использование предсказуемых IP-адресов и HostNameMap (для сопоставления на основе Heat имен хостов с именами хостов предварительно подготовленных серверов), то необходимо настроить **Overcloud** для сопоставления нового замещающего сервера с IP-адресом и именем хоста.

Порядок действий

1. Войдите в **Undercloud** как пользователь **stack**.
2. Примените файл stackrc:

```
$ source ~/stackrc
```

3. Получите **physical_resource_id** и **removed_rsrc_list** для ресурса, который нужно заменить:

```
(undercloud)$ openstack stack resource show <stack> <role>
```

- Поменяйте <stack> на настроенное имя набора управляемых ресурсов (stack), к которому принадлежит ресурс, например, **Overcloud**.
- Поменяйте <role> на имя роли, для которой нужно заменить сервер, например, **Compute**.

Пример вывода:

```
+------------------------+--------------------------------------------------------------------+
| Field                  | Value                                                              |
+------------------------+--------------------------------------------------------------------+
| attributes             | {u'attributes': None, u'refs': None, u'refs_map': None,            |
|                        | u'removed_rsrc_list': [u'2', u'3']}                                |
| creation_time          | 2017-09-05T09:10:42Z                                               |
| description            |                                                                    |
| links                  | [{u'href': u'http://192.168.24.1:8004/v1/bd9e6da805594de9          |
|                        | 8d4a1d3a3ee874dd/stacks/overcloud/1c7810c4-8a1e-                   |
|                        | 4d61-a5d8-9f964915d503/resources/Compute', u'rel':                 |
|                        | u'self'}, {u'href': u'http://192.168.24.1:8004/v1/bd9e6da          |
|                        | 805594de98d4a1d3a3ee874dd/stacks/overcloud/1c7810c4-8a1e-          |
|                        | 4d61-a5d8-9f964915d503', u'rel': u'stack'}, {u'href': u'h          |
|                        | ttp://192.168.24.1:8004/v1/bd9e6da805594de98d4a1d3a3ee874          |
|                        | dd/stacks/overcloud-Compute-zkjccox63svg/7632fb0b-                 |
|                        | 80b1-42b3-9ea7-6114c89adc29', u'rel': u'nested'}]                  |
| logical_resource_id    | Compute                                                            |
| physical_resource_id   | 7632fb0b-80b1-42b3-9ea7-6114c89adc29                               |
| required_by            | [u'AllNodesDeploySteps',                                           |
|                        | u'ComputeAllNodesValidationDeployment',                            |
|                        | u'AllNodesExtraConfig', u'ComputeIpListMap',                       |
|                        | u'ComputeHostsDeployment', u'UpdateWorkflow',                      |
|                        | u'ComputeSshKnownHostsDeployment', u'hostsConfig',                 |
|                        | u'SshKnownHostsConfig', u'ComputeAllNodesDeployment']              |
| resource_name          | Compute                                                            |
| resource_status        | CREATE_COMPLETE                                                    |
| resource_status_reason | state changed                                                      |
| resource_type          | OS::Heat::ResourceGroup                                            |
| updated_time           | 2017-09-05T09:10:42Z                                               |
+------------------------+--------------------------------------------------------------------+
```

В списке **removed_rsrc_list** перечислены индексы серверов, которые уже удалены для ресурса.

4. Получите имя ресурса (resource_name), чтобы определить максимальный индекс, примененный Heat к серверу для этого ресурса:

```
(undercloud)$ openstack stack resource list <physical_resource_id>
```

Поменяйте <physical_resource_id> на ID, который получен на шаге 2.

5. Используйте **resource_name** и **removed_rsrc_list**, чтобы определить следующий индекс, который Heat применит к новому серверу:
   - Если **removed_rsrc_list** пустой, то следующим индексом будет (current_maximum_index) + 1.
   - Если **removed_rsrc_list** включает значение (current_maximum_index) + 1, то следующим индексом будет следующий доступный индекс.
6. Получите ID замещающего физического сервера:

```
(undercloud)$ openstack baremetal node list
```

7. Обновите capability заменяемого сервера новым индексом:

```
$ openstack baremetal node set --property capabilities='node:<role>-
<index>,boot_option:local' <node>
```

- Поменяйте <role> на имя роли, на которую нужно заменить сервер, например, compute.
- Поменяйте <index> на индекс, вычисленный на шаге 5.
- Поменяйте <node> на ID физического сервера.

Планировщик **Compute** использует возможности сервера для его сопоставления при развертывании.

8. Назначьте имя хоста новому серверу – добавьте индекс в конфигурацию **HostnameMap**, например:

```
parameter_defaults:
  ControllerSchedulerHints:
    'capabilities:node': 'controller-%index%'
  ComputeSchedulerHints:
    'capabilities:node': 'compute-%index%'
  HostnameMap:
    overcloud-controller-0: overcloud-controller-prod-123-0
    overcloud-controller-1: overcloud-controller-prod-456-0
    overcloud-controller-2: overcloud-controller-prod-789-0
    overcloud-controller-3: overcloud-controller-prod-456-0
    overcloud-compute-0: overcloud-compute-prod-abc-0
    overcloud-compute-3: overcloud-compute-prod-abc-3
    overcloud-compute-8: overcloud-compute-prod-abc-3
....
```

Не удаляйте настройку (сопоставление) для удаленного сервера из HostnameMap.

9. Добавьте IP-адрес заменяемого сервера в конец каждого списка сетевых IP-адресов в файле сопоставления сетевых IP-адресов **ips-from-pool-all.yaml**.

   В приведенном ниже примере IP-адрес для нового индекса **overcloud-controller-3** добавлен в конец списка IP-адресов для каждой сети ControllerIPs, и ему назначен тот же IP-адрес, что и у **overcloud-controller-1**, поскольку он заменяет **overcloud- controller-1**.

   IP-адрес для нового индекса **overcloud-compute-8** также добавлен в конец списка IP-адресов для каждой сети ComputeIPs, и ему назначен тот же IP-адрес, что и у индекса, который он заменяет – **overcloud-compute-3**:

```
parameter_defaults:
  ControllerIPs:
    ...
    internal_api:
      - 192.168.1.10
      - 192.168.1.11
      - 192.168.1.12
      - 192.168.1.11
    ...
    storage:
      - 192.168.2.10
      - 192.168.2.11
      - 192.168.2.12
      - 192.168.2.11
    ...
```

```
ComputeIPs:
  ...
  internal_api:
    - 172.17.0.10
    - 172.17.0.11
    - 172.17.0.11
  ...
  storage:
    - 172.17.0.10
    - 172.17.0.11
    - 172.17.0.11
  ...
```

21 Замена серверов управления

В определенных обстоятельствах сервер управления в кластере высокой доступности может выйти из строя. В этом случае необходимо удалить сервер из кластера и заменить его новым сервером управления.

Выполните действия, описанные в этом разделе, для замены сервера управления.

Процесс замены сервера управления включает в себя запуск команды развертывания **openstack overcloud** для обновления **Overcloud** запросом на замену сервера управления.

> Описанный далее в подразделах порядок действий применим только к средам высокой доступности. Не используйте его, если имеется только один сервер управления.

21.1 Подготовка к замене сервера управления

До замены сервера управления **Overcloud** важно проверить актуальное состояние среды платформы 1Stack. Эта проверка поможет избежать осложнений в процессе замены сервера управления.

Используйте следующий список предварительных проверок, чтобы определить, безопасно ли выполнять замену сервера управления. Запустите все команды для этих проверок в **Undercloud**.

Порядок действий

1. Проверьте актуальный статус набора ресурсов (stack) **Overcloud** в **Undercloud**:

```
$ source stackrc
(undercloud)$ openstack stack list --nested
```

Набор ресурсов (stack) в overcloud и его последующие дочерние наборы ресурсов должны иметь статус **CREATE_COMPLETE** либо **UPDATE_COMPLETE**.

2. Установите инструменты клиента базы данных:

```
(undercloud)$ sudo dnf -y install mariadb
```

3. Настройте доступ привилегированного пользователя (root) к базе данных:

```
(undercloud)$ sudo cp /var/lib/config-data/puppet-generated/mysql/root/.my.cnf /root/.
```

4. Выполните резервное копирование баз данных **Undercloud**:

```
(undercloud)$ mkdir /home/stack/backup
(undercloud)$ sudo mysqldump --all-databases --quick --single-transaction | gzip >
/home/stack/backup/dump_db_undercloud.sql.gz
```

5. Убедитесь, что в **Undercloud** есть 10 ГБ свободного места для кеширования и изменения образа при выделении нового сервера:

```
(undercloud)$ df –h
```

6. Если IP-адрес для нового сервера управления используется повторно, убедитесь, что удален порт, который использовался старым сервером управления:

```
(undercloud)$ openstack port delete <port>
```

7. Проверьте статус **Pacemaker** на работающих серверах управления. Например, если 192.168.0.47 – IP-адрес работающего сервера управления, используйте следующую команду для просмотра статуса **Pacemaker**:

```
(undercloud)$ ssh heat-admin@192.168.0.47 'sudo pcs status'
```

В выдаче показаны все сервисы, работающие на существующих серверах и остановленные на сервере, который вышел из строя.

8. Проверьте следующие параметры на каждом сервере кластера **Overcloud MariaDB**:
   - **wsrep_local_state_comment**: Synced;
   - **wsrep_cluster_size**: 2.

Используйте следующую команду, чтобы проверить эти параметры на каждом работающем сервере управления. В этом примере IP-адреса сервера управления – 192.168.0.47 и 192.168.0.46:

```
(undercloud)$ for i in 192.168.24.6 192.168.24.7; do echo "*** $i ***" ; ssh heat-
admin@$i "sudo podman exec \$(sudo podman ps --filter name=galera-bundle -q) mysql -e
\"SHOW STATUS LIKE 'wsrep_local_state_comment'; SHOW STATUS LIKE
'wsrep_cluster_size';\""; done
```

9. Проверьте статус **RabbitMQ**. Например, если 192.168.0.47 – IP-адрес работающего сервера управления, используйте следующую команду для просмотра статуса **RabbitMQ**:

```
(undercloud)$ ssh heat-admin@192.168.0.47 "sudo podman exec \$(sudo podman ps -f
name=rabbitmq-bundle -q) rabbitmqctl cluster_status"
```

Ключ **Running_nodes** должен отображать только два доступных сервера, а не неисправный сервер.

10. Если ограждение (fencing) включено, отключите его. Например, если 192.168.0.47 - это IP-адрес работающего сервера управления, используйте следующую команду для проверки статуса ограждения:

```
(undercloud)$ ssh heat-admin@192.168.0.47 "sudo pcs property show stonith-enabled"
```

11. Выполните следующую команду для отключения ограждения:

```
(undercloud)$ ssh heat-admin@192.168.0.47 "sudo pcs property set stonith-enabled=false"
```

12. Убедитесь, что вычислительные сервисы активны на сервере **Undercloud**:

```
(undercloud)$ openstack hypervisor list
```

В выдаче все серверы, не находящиеся в режиме обслуживания, должны отображаться как работающие.

13. Убедитесь, что все контейнеры **Undercloud** запущены:

```
(undercloud)$ sudo podman ps
```

14. 1Остановите все контейнеры nova_*, работающие на вышедшем из строя сервере управления:

```
[root@controller-0 ~]$ sudo systemctl stop tripleo_nova_api.service
[root@controller-0 ~]$ sudo systemctl stop tripleo_nova_api_cron.service
[root@controller-0 ~]$ sudo systemctl stop tripleo_nova_compute.service
[root@controller-0 ~]$ sudo systemctl stop tripleo_nova_conductor.service
[root@controller-0 ~]$ sudo systemctl stop tripleo_nova_metadata.service
[root@controller-0 ~]$ sudo systemctl stop tripleo_nova_placement.service
[root@controller-0 ~]$ sudo systemctl stop tripleo_nova_scheduler.service
```

21.2 Подготовка кластера к замене сервера управления

Перед заменого старого сервера необходимо сначала убедиться, что на нем не работает **Pacemaker**, а затем удалить его из кластера **Pacemaker**.

Порядок действий

1. Чтобы просмотреть список IP-адресов серверов управления, запустите следующую команду:

```
(undercloud) $ openstack server list -c Name -c Networks
+------------------------+------------------------+
| Name                   | Networks               |
+------------------------+------------------------+
| overcloud-compute-0    | ctlplane=192.168.0.44 |
| overcloud-controller-0 | ctlplane=192.168.0.47 |
| overcloud-controller-1 | ctlplane=192.168.0.45 |
| overcloud-controller-2 | ctlplane=192.168.0.46 |
+------------------------+------------------------+
```

2. Если старый сервер еще доступен, авторизуйтесь на одном из остальных серверов и остановите **Pacemaker** на старом сервере. В этом примере показана его остановка на **overcloud-controller-1**:

```
(undercloud) $ ssh heat-admin@192.168.0.47 "sudo pcs status | grep -w Online | grep -w
overcloud-controller-1"(undercloud) $ ssh heat-admin@192.168.0.47 "sudo pcs cluster
stop overcloud-controller-1"
```

Если старый сервер физически недоступен или остановлен, выполнять предыдущую операцию не нужно, поскольку **Pacemaker** на этом сервере уже остановлен.

3. После остановки **Pacemaker** на старом сервере удалите старый сервер из кластера **Pacemaker**. Пример команды, при которой выполняется вход в **overcloud-controller-0** для удаления **overcloud-controller-1**:

```
(undercloud) $ ssh heat-admin@192.168.0.47 "sudo pcs cluster node remove overcloud-controller-1"
```

Если сервер, который нужно заменить, недоступен (например, из-за сбоя оборудования), запустите команду **pcs** с дополнительными параметрами **--skip-offline** и **--force**, чтобы принудительно удалить сервер из кластера:

```
(undercloud) $ ssh heat-admin@192.168.0.47 "sudo pcs cluster node remove overcloud-controller-1 --skip-offline --force"
```

4. После удаления старого сервера из кластера **Pacemaker**, удалите сервер из списка известных хостов в **Pacemaker**:

```
(undercloud) $ ssh heat-admin@192.168.0.47 "sudo pcs host deauth overcloud-controller-1"
```

Эту команду можно выполнять независимо от доступности сервера.

5. Чтобы гарантировать, что новый сервер управления использует корректное устройство ограждения STONITH после замены, удалите старые устройства с сервера следующей командой:

```
(undercloud) $ ssh heat-admin@192.168.0.47 "sudo pcs stonith delete <stonith_resource_name>"
```

6. Поменяйте <stonith_resource_name> на имя ресурса STONITH, которое соответствует старому серверу. Имя ресурса использует формат <resource_agent>-<host_mac>. Агент ресурса и MAC-адрес хоста можно найти в разделе **FencingConfig** файла **fencing.yaml**.
7. База данных **Overcloud** должна продолжать работу во время действий по замене. Чтобы **Pacemaker** не останавливал **Galera** во время этой процедуры, выберите работающий сервер управления и укажите следующую команду в **Undercloud** с IP-адресом сервера управления:

```
(undercloud) $ ssh heat-admin@192.168.0.47 "sudo pcs resource unmanage galera-bundle"
```

21.3 Замена сервера управления

Чтобы заменить сервер управления, укажите индекс сервера, который нужно заменить.

- Если сервер виртуальный, определите сервер, где находится неисправный диск, и восстановите диск из резервной копии. Убедитесь, что MAC-адрес сетевого адаптера, используемого для загрузки PXE на неисправном сервере, остался прежним после замены диска.

- Если сервер представляет собой физический сервер, замените диск, подготовьте новый диск с конфигурацией **Overcloud** и выполните интроспекцию сервера на новом оборудовании.
- Если сервер входит в кластер высокой доступности с отключением от кластера (fencing), возможно, потребуется восстановить серверы **Galera** отдельно.

Выполните следующие действия, чтобы заменить сервер **overcloud-controller-1** на сервер **overcloud-controller-3**. Идентификатор (ID) сервера **overcloud-controller-3** – 75b25e9a-948d-424a-9b3b-f0ef70a6eacf.

Чтобы заменить сервер существующим физическим сервером, включите режим обслуживания на исходящем сервере, чтобы **Undecloud** не выполнял автоматическую повторную подготовку сервера.

Порядок действий

1. Загрузите файл **stackrc**:

```
$ source ~/stackrc
```

2. Определите индекс сервера **overcloud-controller-1**:

```
$ INSTANCE=$(openstack server list --name overcloud-controller-1 -f value -c ID)
```

3. Определите физический сервер, связанный с инстансом:

```
$ NODE=$(openstack baremetal node list -f csv --quote minimal | grep $INSTANCE | cut -
f1 -d,)
```

4. Переведите сервер в режим обслуживания:

```
$ openstack baremetal node maintenance set $NODE
```

5. Если сервер управления виртуальный, укажите следующую команду на хосте управления, чтобы заменить виртуальный диск из резервной копии:

```
$ cp <VIRTUAL_DISK_BACKUP> /var/lib/libvirt/images/<VIRTUAL_DISK>
```

6. Поменяйте <VIRTUAL_DISK_BACKUP> на путь к резервной копии неисправного виртуального диска, а <VIRTUAL_DISK> на имя виртуального диска, который нужно заменить.

   Если резервной копии исходящего сервера нет, необходимо использовать новый виртуализированный сервер.

   Если сервер управления – это физический сервер, выполните следующие действия, чтобы заменить диск новым физическим диском:

   - Поменяйте физический жесткий диск или твердотельный накопитель.
   - Подготовьте сервер с той же конфигурацией, как у неисправного сервера.

7. Получите список несвязанных серверов и определите ID нового сервера:

```
$ openstack baremetal node list --unassociated
```

8. Пометьте новый сервер профилем управления:

```
(undercloud) $ openstack baremetal node set --property
capabilities='profile:control,boot_option:local' 75b25e9a-948d-424a-9b3b-f0ef70a6eacf
```

21.4 Сохранение имен хостов при замене серверов управления, использующих предсказуемые IP-адреса и HostNameMap.0

Если **Overcloud** был настроен для использования предсказуемых IP-адресов, а HostNameMap – для сопоставления имен хостов на основе Heat с именами хостов предварительно подготовленных серверов, то **Overcloud** необходимо настроить для сопоставления нового индекса сервера замены с IP-адресом и именем хоста.

Порядок действий

1. Войдите в **Undercloud** как пользователь **stack**.
2. Загрузите файл **stackrc**:

```
$ source ~/stackrc
```

3. Получите **physical_resource_id** и **removed_rsrc_list** для ресурса, который нужно заменить:

```
(undercloud)$ openstack stack resource show <stack> <role>
```

- Поменяйте <stack> на имя набора управляемых ресурсов (stack), к которому принадлежит ресурс, например, overcloud.
- Поменяйте <role> на имя роли, для которой нужно заменить сервер, например, compute.

Пример:

```
+------------------------+--------------------------------------------------------------+
| Field                  | Value                                                        |
+------------------------+--------------------------------------------------------------+
| attributes             | {u'attributes': None, u'refs': None, u'refs_map': None,      |
|                        | u'removed_rsrc_list': [u'2', u'3']}                          |
| creation_time          | 2017-09-05T09:10:42Z                                         |
| description            |                                                              |
| links                  | [{u'href': u'http://192.168.24.1:8004/v1/bd9e6da805594de9    |
|                        | 8d4a1d3a3ee874dd/stacks/overcloud/1c7810c4-8a1e-             |
|                        | 4d61-a5d8-9f964915d503/resources/Compute', u'rel':           |
|                        | u'self'}, {u'href': u'http://192.168.24.1:8004/v1/bd9e6da    |
|                        | 805594de98d4a1d3a3ee874dd/stacks/overcloud/1c7810c4-8a1e-    |
|                        | 4d61-a5d8-9f964915d503', u'rel': u'stack'}, {u'href': u'h    |
|                        | ttp://192.168.24.1:8004/v1/bd9e6da805594de98d4a1d3a3ee874    |
|                        | dd/stacks/overcloud-Compute-zkjccox63svg/7632fb0b-           |
|                        | 80b1-42b3-9ea7-6114c89adc29', u'rel': u'nested'}]            |
| logical_resource_id    | Compute                                                      |
| physical_resource_id   | 7632fb0b-80b1-42b3-9ea7-6114c89adc29                         |
| required_by            | [u'AllNodesDeploySteps',                                     |
|                        | u'ComputeAllNodesValidationDeployment',                      |
|                        | u'AllNodesExtraConfig', u'ComputeIpListMap',                 |
|                        | u'ComputeHostsDeployment', u'UpdateWorkflow',                |
|                        | u'ComputeSshKnownHostsDeployment', u'hostsConfig',           |
|                        | u'SshKnownHostsConfig', u'ComputeAllNodesDeployment']        |
| resource_name          | Compute                                                      |
| resource_status        | CREATE_COMPLETE                                              |
```

```
| resource_status_reason | state changed                                                  |
| resource_type          | OS::Heat::ResourceGroup                                        |
| updated_time           | 2017-09-05T09:10:42Z                                           |
+------------------------+----------------------------------------------------------------+
```

4. Получите **resource_name**, чтобы определить максимальный индекс, который был применен сервисом Heat к серверу для этого ресурса:

```
(undercloud)$ openstack stack resource list <physical_resource_id>
```

5. Поменяйте <physical_resource_id> на ID, который был получен на шаге 2. Используйте **resource_name** и **removed_rsrc_list**, чтобы определить следующий индекс, который сервис Heat применит к новому серверу:
   - Если **removed_rsrc_list** пуст, следующим индексом будет (current_maximum_index) + 1.
   - Если **removed_rsrc_list** включает значение (current_maximum_index) + 1, то следующим индексом будет следующий доступный индекс.
6. Получите ID заменяемого физического сервера:

```
(undercloud)$ openstack baremetal node list
```

7. Обновите возможности capability заменяемого сервера новым индексом:

```
openstack baremetal node set --property capabilities='node:<role>-
<index>,boot_option:local' <node>
```

   - Поменяйте <role> на имя роли, для которой нужно заменить сервер, например, compute.
   - Поменяйте <index> индексом, рассчитанным на шаге 5.
   - Поменяйте <node> на ID физического сервера.

Планировщик **Compute** (scheduler) использует возможности сервера для сопоставления сервера при развертывании.

8. Назначьте имя хоста для нового сервера, добавив индекс в конфигурацию HostnameMap, например:

```
parameter_defaults:
  ControllerSchedulerHints:
    'capabilities:node': 'controller-%index%'
  ComputeSchedulerHints:
    'capabilities:node': 'compute-%index%'
  HostnameMap:
    overcloud-controller-0: overcloud-controller-prod-123-0
    overcloud-controller-1: overcloud-controller-prod-456-0
    overcloud-controller-2: overcloud-controller-prod-789-0
    overcloud-controller-3: overcloud-controller-prod-456-0
    overcloud-compute-0: overcloud-compute-prod-abc-0
    overcloud-compute-3: overcloud-compute-prod-abc-3
    overcloud-compute-8: overcloud-compute-prod-abc-3
    ....
```

Не удаляйте настройку (mapping) удаленного сервера из HostnameMap.

9. Добавьте IP-адрес заменяемого сервера в конец каждого списка сетевых IP-адресов в файле сопоставления сетевых IP-адресов **ips-from-pool-all.yaml**.

   В следующем примере IP-адрес для нового индекса **overcloud-controller-3** добавлен в конце списка IP-адресов для каждой сети ControllerIPs, и ему назначен тот же IP-адрес, что и у **overcloud-controller-1**, поскольку он заменяет **overcloud- controller-1.**
   IP-адрес для нового индекса **overcloud-compute-8** также добавлен в конце списка IP-адресов для каждой сети ComputeIPs, и ему назначен тот же IP-адрес, что и у индекса, который он заменяет – **overcloud-compute-3**:

```
parameter_defaults:
  ControllerIPs:
    ...
    internal_api:
      - 192.168.1.10
      - 192.168.1.11
      - 192.168.1.12
      - 192.168.1.11
    ...
    storage:
      - 192.168.2.10
      - 192.168.2.11
      - 192.168.2.12
      - 192.168.2.11
    ...
  ComputeIPs:
    ...
    internal_api:
      - 172.17.0.10
      - 172.17.0.11
      - 172.17.0.11
    ...
    storage:
      - 172.17.0.10
      - 172.17.0.11
      - 172.17.0.11
    ...
```

21.5 Инициация замены сервера управления

Выполните следующие действия, чтобы удалить старый сервер управления и заменить его новым сервером управления.

Порядок действий

1. Определите UUID сервера управления, который нужно удалить, и сохраните его в переменной <NODEID>. Убедитесь, что <node_name> будет меняться на имя сервера, который нужно удалить:

```
(undercloud)[stack@undercloud ~]$ NODEID=$(openstack server list -f value -c ID --name
<node_name>)
```

2. Чтобы определить ID ресурса Heat, укажите следующую команду:

```
$ openstack stack resource show overcloud ControllerServers -f json -c attributes | jq
--arg NODEID "$NODEID" -c '.attributes.value | keys[] as $k | if .[$k] == $NODEID then
"Node index \($k) for \(.[$k])" else empty end'
```

3. Создайте файл среды **~/templates/remove-controller.yaml** и добавьте индекс сервера управления, который нужно удалить:

```
parameters:
  ControllerRemovalPolicies:
    [{'resource_list': ['<node_index>']}]
```

4. Введите команду развертывания **Overcloud** и добавьте файл среды **remove-controller.yaml** и любые другие файлы среды, относящиеся к среде пользователя:

```
(undercloud) $ openstack overcloud deploy --templates \
    -e /home/stack/templates/remove-controller.yaml \
    [OTHER OPTIONS]
```

   - Включите **-e ~/templates/remove-controller.yaml** только для этого примера команды развертывания. Удалите этот файл среды из последующих операций развертывания.
   - Включите **~/templates/bootstrap-controller.yaml**, если нужно заменить сервер управления начальной загрузки и сохранить имя сервера.

5. **Undercloud** удаляет старый сервер, создает новый сервер и обновляет набор ресурсов (stack) **Overcloud**. Проверить статус набора ресурсов (stack) **Overcloud** можно следующей командой:

```
(undercloud) $ openstack stack list --nested
```

6. После завершения команды развертывания **Undercloud** покажет, что старый сервер заменен новым:

```
(undercloud) $ openstack server list -c Name -c Networks
+------------------------+------------------------+
| Name                   | Networks               |
+------------------------+------------------------+
| overcloud-compute-0    | ctlplane=192.168.0.44  |
| overcloud-controller-0 | ctlplane=192.168.0.47  |
| overcloud-controller-2 | ctlplane=192.168.0.46  |
| overcloud-controller-3 | ctlplane=192.168.0.48  |
+------------------------+------------------------+
```

Теперь на новом сервере размещены работающие сервисы области управления.

21.6 Очистка ресурсов после замены сервера управления

После замены сервера выполните следующие действия, чтобы завершить работу кластера управления.

Порядок действий

1. Войдите в сервер управления.
2. Включите управление **Pacemaker** кластером **Galera** и запустите **Galera** на новом сервере:

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs resource refresh galera-bundle
[heat-admin@overcloud-controller-0 ~]$ sudo pcs resource manage galera-bundle
```

3. Выполните финальную проверку статуса, чтобы убедиться, что сервисы работают корректно:

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs status
```

В случае сбоя в работе какого-либо сервиса, используйте команду обновления ресурсов **pcs**, чтобы устранить и перезапустить отказавшие сервисы.

4. Выйдите из **Undercloud**:

```
[heat-admin@overcloud-controller-0 ~]$ exit
```

5. Примените файл **overcloudrc** для взаимодействия с **Undercloud**:

```
$ source ~/overcloudrc
```

6. Проверьте сетевые агенты в облачной среде:

```
(overcloud) $ openstack network agent list
```

7. Если появятся какие-либо агенты для старого сервера, удалите их:

```
(overcloud) $ for AGENT in $(openstack network agent list --host overcloud-controller-
1.localdomain -c ID -f value) ; do openstack network agent delete $AGENT ; done
```

8. При необходимости добавьте маршрутизатор к хосту агента L3 на новом сервере. Используйте следующий пример команды, чтобы добавить маршрутизатор с именем **r1** к агенту L3 при помощи UUID 2d1c1dc1-d9d4-4fa9-b2c8-f29cd1a649d4:

```
(overcloud) $ openstack network agent add router --l3 2d1c1dc1-d9d4-4fa9-b2c8-
f29cd1a649d4 r1
```

9. Очистите сервисы блочных хранилищ (cinder).
   - Получите список блочных хранилищ:

```
(overcloud) $ openstack volume service list
```

   - Войдите на сервер управления, подключитесь к контейнеру cinder-api и используйте команду удаления сервиса cinder-manage service remove, чтобы удалить оставшиеся сервисы:

```
[heat-admin@overcloud-controller-0 ~]$ sudo podman exec -it cinder_api cinder-manage
service remove cinder-backup <host>
[heat-admin@overcloud-controller-0 ~]$ sudo podman exec -it cinder_api cinder-manage
service remove cinder-scheduler <host>
```

10. Очистите кластер RabbitMQ.
    - Войдите на сервер управления.
    - Используйте команду **podman exec** для запуска bash и проверьте статус кластера RabbitMQ:

```
[heat-admin@overcloud-controller-0 ~]$ podman exec -it rabbitmq-bundle-podman-0 bash
[heat-admin@overcloud-controller-0 ~]$ rabbitmqctl cluster_status
```

    - Используйте команду **rabbitmqctl**, чтобы «забыть» замененный сервер управления:

```
[heat-admin@overcloud-controller-0 ~]$ rabbitmqctl forget_cluster_node <node_name>
```

11. Если сервер управления начальной загрузки заменен, то после замены необходимо удалить файл среды **~/templates/bootstrap-controller.yaml** или удалить параметры **pacemaker_short_bootstrap_node_name** и **mysql_short_bootstrap_node_name** из существующего файла среды. Этот шаг предотвращает попытку **Undercloud** переопределить имя сервера управления при последующих заменах.

22 Перезагрузка серверов

Возможно, придется перезагрузить серверы в **Undercloud** и **Overcloud**. Используйте следующий порядок действий, чтобы понять, как перезагрузить серверы разных типов:

- Если нужно перезагрузить все серверы одной роли, рекомендуется перезагружать каждый сервер по отдельности. Если же это сделать одновременно для всех серверов в роли, то во время операции перезагрузки может произойти простой сервиса.
- Если нужно перезагрузить все серверы в среде платформы 1Stack, перезагрузите серверы последовательно в указанном порядке:

1. Перезагрузите сервер **Undercloud**.
2. Перезагрузите серверы управления.
3. Перезагрузите автономные серверы **Ceph MON** (опционально).
4. Перезагрузите серверы хранилища **Ceph** (опционально).
5. Перезагрузите серверы сервиса **Object Storage** (опционально).
6. Перезагрузите вычислительные серверы.

22.1 Перезагрузка сервера Undercloud

Перезагрузите сервер **Undercloud.**

Порядок действий

1. Войдите в **Undercloud** как пользователь **stack**.
2. Перезагрузите **Undercloud**:

```
$ sudo reboot
```

3. Дождитесь загрузки сервера.

22.2 Перезагрузка серверов управления

Перезагрузите серверы управления.

Порядок действий

1. Войдите на сервер, который нужно перезагрузить.
2. Опционально: если сервер использует ресурсы **Pacemaker**, остановите кластер:

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs cluster stop
```

3. Перезагрузите сервер:

```
[heat-admin@overcloud-controller-0 ~]$ sudo reboot
```

4. Дождитесь окончания загрузки сервера.

Проверка

Проверьте, что сервисы включены.

- Если сервер использует сервисы **Pacemaker**, проверьте, что сервер снова присоединился к кластеру:

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs status
```

- Если сервер использует сервисы **Systemd**, проверьте, что все сервисы включены:

```
[heat-admin@overcloud-controller-0 ~]$ sudo systemctl status
```

- Если сервер использует контейнерные сервисы, проверьте, что все контейнеры на сервере активны:

```
[heat-admin@overcloud-controller-0 ~]$ sudo podman ps
```

22.3 Перезагрузка вычислительных серверов

Чтобы обеспечить минимальное время простоя инстансов в среде 1Stack, нужно при мигрции инстансов с вычислительного сервера выполнить описанные ниже действия.

> Если инстансы с исходного вычислительного сервера на другой вычислительный сервер перенесены не будут, они могут быть перезапущены на исходном сервере, что может привести к сбою обновления. Это связано с известной проблемой, связанной с изменениями в **Podman** и сервисе **libvirt**

Рабочий процесс миграции инстансов

1. Решите, следует ли переносить инстансы на другой вычислительный сервер, перед перезагрузкой сервера.
2. Выберите и отключите вычислительный сервер, который нужно перезагрузить, чтобы он не подготавливал новые инстансы.
3. Перенесите инстансы на другой вычислительный сервер.
4. Перезагрузите пустой вычислительный сервер.
5. Включите пустой вычислительный сервер.

Предварительные требования

- Перед перезагрузкой сервера необходимо решить, следует ли переносить инстансы на другой вычислительный сервер в процессе перезагрузки.

Ознакомьтесь со списком ограничений миграции, которые могут возникнуть при переносе инстансов между вычислительными серверами.

Если перенести инстансы нельзя, можно установить следующие основные параметры шаблона для управления состоянием инстансов после перезагрузки вычислительного сервера:

NovaResumeGuestsStateOnHostBoot

Определяет, следует ли возвращать инстансы в то же состояние на вычислительном сервере после перезагрузки. Если установлено значение **False**, инстансы остаются выключенными, поэтому их необходимо запустить вручную. **False** является значением по умолчанию.

NovaResumeGuestsShutdownTimeout

Определяет количество секунд ожидания до завершения работы инстанса перед перезагрузкой. Не рекомендуется устанавливать нулевое значение. Значение по умолчанию равно 300.

Порядок действий

1. Войдите в **Undercloud** как пользователь **stack**.
2. Получите список всех вычислительных серверов и их UUID:

```
$ source ~/stackrc
(undercloud) $ openstack server list --name compute
```

Определите UUID вычислительного сервера, который нужно перезагрузить.

3. Из **Undercloud** выберите вычислительный сервер (Compute) и отключите его:

```
$ source ~/overcloudrc
(overcloud) $ openstack compute service list
(overcloud) $ openstack compute service set <hostname> nova-compute --disable
```

4. Получите список всех инстансов на вычислительном сервере:

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

5. Опционально: если принято решение перенести инстансы на другой вычислительный сервер, выполните следующие действия:

a) Используйте одну из команд:

- Чтобы перенести инстанс на другой хост, укажите:

```
(overcloud) $ openstack server migrate <instance_id> --live
   <target_host> --wait
```

- Разрешите **nova-scheduler** автоматически выбрать целевой хост:

```
(overcloud) $ nova live-migration <instance_id>
```

- Выполните динамическую миграцию всех инстансов одновременно:

```
$ nova host-evacuate-live <hostname>
```

Команда **nova** может вызвать предупреждения об устаревании, которое можно игнорировать.

б) Дождитесь окончания миграции.

в) Подтвердите, что миграция прошла успешно:

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

г) Продолжите миграцию инстансов до их полного отсутствия на вычислительном сервере.

6. Войдите на вычислительный сервер и перезагрузите сервер:

```
[heat-admin@overcloud-compute-0 ~]$ sudo reboot
```

7. Дождитесь загрузки сервера.
8. Повторно активируйте вычислительный сервер:

```
$ source ~/overcloudrc
(overcloud) $ openstack compute service set <hostname> nova-compute –enable
```

9. Проверьте, что вычислительный сервер включен:

```
(overcloud) $ openstack compute service list
```

23 Отключение и запуск Undercloud и Overcloud

Для обслуживания **Undercloud** и **Overcloud** необходимо выключить и запустить серверы **Undercloud** и **Overcloud** в определенном порядке, чтобы свести проблемы к минимуму при запуске **Overcloud**.

Предварительные требования

- **Undercloud** и **Overcloud** находятся в рабочем состоянии.

23.1 Порядок завершения работы Undercloud и Overcloud

Для завершения работы 1Stack, необходимо остановить работу **Overcloud** и **Undercloud** в следующем порядке:

1. Выключите инстансы на вычислительных серверах **Overcloud.**
2. Выключите вычислительные серверы.
3. Остановите все сервисы высокой доступности и платформы 1Stack на серверах управления.
4. Выключите серверы **Ceph Storage** (опционально).
5. Выключите серверы управления.
6. Выключите **Undercloud.**

23.2 Отключение инстансов на вычислительных серверах Overcloud

В процессе завершения работы среды 1Stack остановите работу всех инстансов на вычислительных серверах перед завершением работы этих серверов.

Предварительные требования

- **Overcloud** с активными сервисами вычислительных ресурсов.

Порядок действий

1. Войдите в **Undercloud** как пользователь **stack**.
2. Создайте файл учетных данных для **Overcloud**:

```
$ source ~/overcloudrc
```

3. Просмотрите инстансы, запущенные в **Overcloud**:

```
$ openstack server list --all-projects
```

4. Остановите работу каждого инстанса в **Overcloud**:

```
$ openstack server stop <INSTANCE>
```

Все инстансы в **Overcloud** должны быть остановлены.

23.3 Отключение вычислительных серверов

В процессе завершения работы среды 1Stack авторизуйтесь и остановите работу каждого вычислительного сервера.

Предварительные требования

- Все инстансы на вычислительных серверах выключены.

Порядок действий

1. Авторизуйтесь как привилегированный пользователь root на вычислительном сервере.
2. Выключите сервер:

```
# poweroff
```

3. Выполните эти действия для каждого вычислительного сервера.

23.4 Завершение работы сервисов на серверах управления

В процессе завершения работы среды 1Stack остановите сервисы на серверах управления перед выключением серверов (включая сервисы **Pacemaker** и **systemd**).

Предварительные требования

- **Overcloud** с активными сервисами **Pacemaker.**

Порядок действий

1. Авторизуйтесь как пользователь **root** на сервере управления.
2. Остановите кластер **Pacemaker**.

```
# pcs cluster stop --all
```

Эта команда останавливает кластер на всех серверах.

3. Дождитесь прекращения работы сервисов **Pacemaker** и проверьте, что сервисы остановлены.
   - Проверьте статус **Pacemaker**:

```
# pcs status
```

   - Проверьте, что ни один сервис **Pacemaker** не работает в **Podman**:

```
# podman ps --filter "name=.-bundle."
```

4. Остановите сервисы 1Stack:

```
# systemctl stop 'tripleo*'_
```

5. Дождитесь остановки сервисов и проверьте, что сервисы более не работают в **Podman**:

```
# podman ps
```

23.5 Отключение серверов управления

В процессе завершения работы среды 1Stack войдите в систему и завершите работу каждого сервера управления.

Предварительные требования

Должны быть остановлены:

- кластер **Pacemaker**;
- все сервисы 1Stack на серверах управления.

Порядок действий

1. Авторизуйтесь как пользователь **root** на сервере управления.
2. Завершите работу сервера:

```
# poweroff
```

3. Выполните эти действия для каждого сервера управления.

23.6 Отключение Undercloud

В процессе завершения работы среды 1Stack войдите на сервер **Undercloud** и отключите **Undercloud**.

Предварительные требования

- Работающий **Undercloud.**

Порядок действий

1. Авторизуйтесь в **Undercloud** как пользователь **stack**.
2. Завершите работу **Undercloud**:

```
$ sudo poweroff
```

23.7 Проведение работ по обслуживанию системы

После полного отключения **Undercloud** и **Overcloud** выполните любые работы по обслуживанию используемых систем, а затем запустите **Undercloud** и **Overcloud**.

23.8 Порядок запуска Undercloud и Overcloud

Для запуска среды 1Stack запустите **Undercloud** и **Overcloud** в следующем порядке:

1. Запустите **Undercloud.**
2. Запустите серверы управления.
3. Запустите серверы **Ceph Storage** (опционально),
4. Запустите вычислительные серверы.
5. Запустите инстансы на вычислительных серверах в **Overcloud.**

23.9 Запуск Undercloud

В процессе запуска среды 1Stack включите сервер Undercloud, войдите в **Undercloud** и проверьте сервисы **Undercloud**.

Предварительные требования

- Выключенный **Undercloud.**

Порядок действий

Включите **Undercloud** и дождитесь загрузки **Undercloud**.

Проверка

1. Авторизуйтесь в **Undercloud** как пользователь **stack**.
2. Проверьте сервисы в **Undercloud**:

```
$ systemctl list-units 'tripleo*'_
```

3. Создайте файл учетных данных для **Undercloud** и запустите команду валидации, чтобы убедиться, что все сервисы и контейнеры активны и работоспособны:

```
$ source stackrc$ openstack tripleo validator run --validation service-status --limit
undercloud
```

23.10 Запуск серверов управления

В процессе запуска среды 1Stack включите каждый сервер управления и проверьте на нем сервисы, не относящиеся к **Pacemaker**.

Предварительные требования

- Выключенные серверы управления.

Порядок действий

Включите каждый сервер управления.

Проверка

1. Авторизуйтесь на каждом сервере управления как пользователь **root**.
2. Проверьте сервисы на сервере управления:

```
$ systemctl -t service
```

Работают только сервисы, не основанные на **Pacemaker**.

3. Дождитесь запуска сервисов **Pacemaker** и убедитесь, что сервисы запущены:

```
$ pcs status
```

Если в среде используется инстанс высокой доступности, ресурсы **Pacemaker** запустятся только после запуска вычислиельных серверов или выполнения операции снятия ограничения (unfence) вручную с помощью команды **pcs stonith submit <compute_node>**.
Эту команду необходимо выполнить на каждом вычислительном сервере, использующем инстанс высокой доступности.

23.11 Запуск вычислительных серверов

В процессе запуска среды 1Stack включите каждый вычислительный сервер и проверьте сервисы на сервере.

Предварительные требования

- Выключенные вычислительные серверы.

Порядок действий

Включите каждый вычислительный сервер.

Проверка

1. Авторизуйтесь на каждом вычислительном сервере как пользователь **root**.
2. Проверьте сервисы на вычислительном сервере:

```
$ systemctl -t service
```

23.12 Запуск инстансов на вычислительных серверах Overcloud

В процессе запуска среды 1Stack запустите инстансы на вычислительных серверах.

Предварительные требования

- Активный **Overcloud** с активными серверами.

Порядок действий

1. Войдите в **Undercloud** как пользователь **stack**.
2. Примените файл с учетными данными для **Overcloud**:

```
$ source ~/overcloudrc
```

3. Просмотрите запущенные инстансы в **Overcloud**:

```
$ openstack server list --all-projects
```

4. Запустите инстанс в **Overcloud**:

```
$ openstack server start <INSTANCE>
```

24 Операции интроспекции

В некоторых ситуациях может потребоваться интроспекция вне стандартного рабочего процесса развертывания **Overcloud**. Например, если нужно выполнить интроспекцию новых серверов или обновить данные интоспекции после замены оборудования на существующих неиспользуемых серверах.

24.1 Выполнение интроспекции отдельного сервера

Чтобы выполнить одиночную интроспекцию на доступном сервере, переведите сервер в режим управления и выполните интроспекцию.

Порядок действий

1. Приведите все серверы в управляемое состояние:

```
(undercloud) $ openstack baremetal node manage [NODE UUID]
```

2. Проведите интроспекцию:

```
(undercloud) $ openstack overcloud node introspect [NODE UUID] --provide
```

После завершения интроспекции сервер перейдет в доступное состояние.

24.2 Выполнение интроспекции сервера после первоначальной интроспекции

После первоначальной интроспекции все серверы переходят в состояние доступности благодаря опции **--provide**.

Чтобы выполнить интроспекцию на всех серверах после первоначальной интроспекции, переведите сервер в режим управления и выполните интроспекцию.

Порядок действий

1. Приведите все серверы в управляемое состояние:

```
(undercloud) $ for node in $(openstack baremetal node list --fields uuid -f value) ; do
openstack baremetal node manage $node ; done
```

2. Запустите команду массовой интроспекции

```
(undercloud) $ openstack overcloud node introspect --all-manageable --provide
```

После завершения интроспекции все серверы перейдут в доступное состояние.

24.3 Выполнение интроспекции сети для получения информации об интерфейсе

Сетевая интроспекция извлекает данные протокола обнаружения канального уровня (link layer discovery protocol, LLDP) из сетевых коммутаторов.

Следующие команды показывают подмножество информации LLDP для всех интерфейсов на сервере или полную информацию для конкретного сервера и интерфейса. Это может быть полезно при устранении неполадок. **Undercloud** активирует сбор данных LLDP по умолчанию.

Порядок действий

1. Чтобы получить список интерфейсов на сервере, запустите команду:

```
(undercloud) $ openstack baremetal introspection interface list [NODE UUID]
```

Например:

```
(undercloud) $ openstack baremetal introspection interface list c89397b7-a326-41a0-
907d-79f8b86c7cd9
+-----------+-------------------+------------------------+-------------------+---------
-------+
| Interface | MAC Address       | Switch Port VLAN IDs   | Switch Chassis ID | Switch
Port ID |
+-----------+-------------------+------------------------+-------------------+---------
-------+
| p2p2      | 00:0a:f7:79:93:19 | [103, 102, 18, 20, 42] | 64:64:9b:31:12:00 | 510
|
| p2p1      | 00:0a:f7:79:93:18 | [101]                  | 64:64:9b:31:12:00 | 507
|
| em1       | c8:1f:66:c7:e8:2f | [162]                  | 08:81:f4:a6:b3:80 | 515
|
| em2       | c8:1f:66:c7:e8:30 | [182, 183]             | 08:81:f4:a6:b3:80 | 559
|
+-----------+-------------------+------------------------+-------------------+---------
-------+
```

2. Чтобы просмотреть данные интерфейса и информацию о порте коммутатора, запустите команду:

```
(undercloud) $ openstack baremetal introspection interface show [NODE UUID] [INTERFACE]
```

Например:

```
(undercloud) $ openstack baremetal introspection interface show c89397b7-a326-41a0-
907d-79f8b86c7cd9 p2p1
+--------------------------------------+-------------------------------------------------
-----------------------------------------------------------------+
| Field                                | Value
|
+--------------------------------------+-------------------------------------------------
-----------------------------------------------------------------+
| interface                            | p2p1
|
| mac                                  | 00:0a:f7:79:93:18
|
| node_ident                           | c89397b7-a326-41a0-907d-79f8b86c7cd9
|
```

```
| switch_capabilities_enabled          | [u'Bridge', u'Router']
|
| switch_capabilities_support          | [u'Bridge', u'Router']
|
| switch_chassis_id                    | 64:64:9b:31:12:00
|
| switch_port_autonegotiation_enabled  | True
|
| switch_port_autonegotiation_support  | True
|
| switch_port_description              | ge-0/0/2.0
|
| switch_port_id                       | 507
|
| switch_port_link_aggregation_enabled | False
|
| switch_port_link_aggregation_id      | 0
|
| switch_port_link_aggregation_support | True
|
| switch_port_management_vlan_id       | None
|
| switch_port_mau_type                 | Unknown
|
| switch_port_mtu                      | 1514
|
| switch_port_physical_capabilities    | [u'1000BASE-T fdx', u'100BASE-TX fdx',
u'100BASE-TX hdx', u'10BASE-T fdx', u'10BASE-T hdx', u'Asym and Sym PAUSE fdx'] |
| switch_port_protocol_vlan_enabled    | None
|
| switch_port_protocol_vlan_ids        | None
|
| switch_port_protocol_vlan_support    | None
|
| switch_port_untagged_vlan_id         | 101
|
| switch_port_vlan_ids                 | [101]
|
| switch_port_vlans                    | [{u'name': u'RHOS13-PXE', u'id': 101}]
|
| switch_protocol_identities           | None
|
| switch_system_name                   | rhos-compute-node-sw1
|
+--------------------------------------+-----------------------------------------------
-----------------------------------------------------------------------+
```

24.4 Получение данных интроспекции аппаратного оборудования

Функция **hardware-inspection-extras** сервиса **Bare Metal** включена по умолчанию, ее можно использовать для получения сведений об оборудовании для конфигурации **Overcloud**.

Например, сборщик **numa_topology** является частью функции **hardware-inspection-extras** и содержит следующую информацию для каждого сервера **NUMA**:

- ОЗУ (в килобайтах);
- физические ядра ЦП и их родственные потоки;
- сетевые адаптеры, связанные с сервером **NUMA**.

Порядок действий

Чтобы получить указанную выше информацию, поменяйте <UUID> на UUID физического сервера для выполнения следующей команды:

```
# openstack baremetal introspection data save <UUID> | jq .numa_topology
```

В примере показана полученная информация NUMA для физического сервера:


```
{
  "cpus": [
    {
      "cpu": 1,
      "thread_siblings": [
        1,
        17
      ],
      "numa_node": 0
    },
    {
      "cpu": 2,
      "thread_siblings": [
        10,
        26
      ],
      "numa_node": 1
    },
    {
      "cpu": 0,
      "thread_siblings": [
        0,
        16
      ],
      "numa_node": 0
    },
    {
      "cpu": 5,
      "thread_siblings": [
        13,
        29
      ],
      "numa_node": 1
    },
    {
      "cpu": 7,
      "thread_siblings": [
        15,
        31
      ],
      "numa_node": 1
    },
    {
      "cpu": 7,
      "thread_siblings": [
        7,
        23
      ],
      "numa_node": 0
    },
    {
      "cpu": 1,
      "thread_siblings": [
        9,
        25
      ],
      "numa_node": 1
```

```
    },
    {
      "cpu": 6,
      "thread_siblings": [
        6,
        22
      ],
      "numa_node": 0
    },
    {
      "cpu": 3,
      "thread_siblings": [
        11,
        27
      ],
      "numa_node": 1
    },
    {
      "cpu": 5,
      "thread_siblings": [
        5,
        21
      ],
      "numa_node": 0
    },
    {
      "cpu": 4,
      "thread_siblings": [
        12,
        28
      ],
      "numa_node": 1
    },
    {
      "cpu": 4,
      "thread_siblings": [
        4,
        20
      ],
      "numa_node": 0
    },
    {
      "cpu": 0,
      "thread_siblings": [
        8,
        24
      ],
      "numa_node": 1
    },
    {
      "cpu": 6,
      "thread_siblings": [
        14,
        30
      ],
      "numa_node": 1
    },
    {
      "cpu": 3,
      "thread_siblings": [
        3,
        19
      ],
      "numa_node": 0
    },
    {
      "cpu": 2,
      "thread_siblings": [
```

```
          2,
          18
        ],
        "numa_node": 0
      }
    ],
    "ram": [
      {
        "size_kb": 66980172,
        "numa_node": 0
      },
      {
        "size_kb": 67108864,
        "numa_node": 1
      }
    ],
    "nics": [
      {
        "name": "ens3f1",
        "numa_node": 1
      },
      {
        "name": "ens3f0",
        "numa_node": 1
      },
      {
        "name": "ens2f0",
        "numa_node": 0
      },
      {
        "name": "ens2f1",
        "numa_node": 0
      },
      {
        "name": "ens1f1",
        "numa_node": 0
      },
      {
        "name": "ens1f0",
        "numa_node": 0
      },
      {
        "name": "eno4",
        "numa_node": 0
      },
      {
        "name": "eno1",
        "numa_node": 0
      },
      {
        "name": "eno3",
        "numa_node": 0
      },
      {
        "name": "eno2",
        "numa_node": 0
      }
    ]
  }
```

25 TLS в Overcloud

При использовании решения центра сертификации (certification authority, CA) доступны: продление сертификатов, списки отзыва сертификатов (CRL) и средства криптографии, принятые в отрасли.

Предварительные требования

- Установлен пакет **openssl-perl**.
- Имеется сертификат SSL/TLS.

25.1 Обновление сертификатов TLS вручную

Выполните следующие действия, если используются собственные сертификаты TLS, которые не создаются автоматически в процессе TLS everywhere (TLS-e).

Порядок действий

1. Отредактируйте шаблоны Heat:
   - Отредактируйте файл **enable-tls.yaml** и обновите параметры **SSLCertificate**, **SSLKey** и **SSLIntermediateCertificate**.
   - Если центр сертификации изменился, отредактируйте файл **inject-trust-anchor-hiera.yaml** и обновите параметр **CAMap**.
2. Повторно запустите команду развертывания:

```
$ openstack overcloud deploy --templates \
[...]
-e /home/stack/templates/enable-tls.yaml \
-e ~/templates/custom-domain.yaml \
-e ~/templates/inject-trust-anchor-hiera.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/ssl/tls-endpoints-public-
dns.yaml
```

3. На **Undercloud** запустите следующую команду для сервера управления:

```
ssh heat-admin@<controller> sudo podman \
restart $(podman ps --format="{{.Names}}" | grep -w -E 'haproxy(-bundle-.*-[0-9]+)?')
```

26 Резервное копирование сервера Undercloud

Для резервного копирования сервера **Undercloud:** настройте сервер резервного копирования, установите инструмент Relax-and-Recover на сервере **Undercloud** и создайте образ резервной копии. Создавать резервные копии можно в рамках регулярного обслуживания среды.

Кроме того, перед обновлениями или модернизацией необходимо создать резервную копию сервера **Undercloud**.

Резервные копии можно использовать для восстановления сервера **Undercloud** до его предыдущего состояния в случаях, когда в процессе обновления или модернизации возникают ошибки.

26.1 Поддерживаемые форматы и протоколы резервного копирования

В процессе резервного копирования и восстановления **Undercloud** используется инструмент с открытым исходным кодом **Relax-and-Recover (ReaR)** для создания и восстановления загрузочных резервных образов. **ReaR** написан на Bash и поддерживает многие форматов образов и транспортных протоколы.

Форматы и протоколы резервного копирования, которые поддерживает платформа 1Stack при использовании ReaR для резервного копирования и восстановления **Undercloud** и области управления:

Форматы загрузочных носителей:

- ISO.

Протоколы передачи файлов:

- SFTP,
- NFS.

26.2 Настройка места хранения резервных копий

Перед созданием резервной копии серверов управления настройте место хранения резервной копии в файле среды **bar-vars.yaml**. В этом файле хранятся параметры «ключ-значение», которые нужно передать при выполнении резервного копирования.

Порядок действий

1. В файле **bar-vars.yaml** настройте место хранения резервных копий:
   - Если используется NFS-сервер, добавьте параметр **tripeo_backup_and_restore_server** и установите в качестве значения IP-адрес NFS-сервера:

```
tripleo_backup_and_restore_server: <ip_address>
```

По умолчанию значение параметра **tripeo_backup_and_restore_server** – 192.168.24.1.

- Если используется SFTP-сервер, добавьте параметр **tripeo_backup_and_restore_output_url** и задайте значения URL-адреса и учетных данных SFTP-сервера:

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

2. Поменяйте <user>, <password> и <backup_node> на URL-адрес и учетные данные резервного сервера.

26.3 Установка и настройка NFS-сервера на сервере резервного копирования

Новый сервер NFS можно установить и настроить для хранения файла резервной копии.

Для установки и настройки сервера NFS на сервере резервного копирования: создайте файл инвентаризации, создайте ключ SSH и запустите команду резервного копирования **Undercloud** с параметрами сервера NFS.

- Если сервер NFS или SFTP уже установлен и настроен, выполнять приведенный ниже порядок действий не требуется. Введите информацию о сервере при настройке **ReaR** на сервере, резервную копию которого нужно создать.
- По умолчанию параметр IP-адреса **Relax** and **Recover (ReaR)** для сервера NFS – 192.168.24.1. Чтобы установить значение IP-адреса, соответствующее среде, необходимо добавить параметр **tripeo_backup_and_restore_server**.

Порядок действий

1. На сервере **Undercloud** получите учетные данные **Undercloud**:

```
[stack@undercloud ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

2. На сервере **Undercloud** создайте файл инвентаризации для сервера резервного копирования:

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF'> ~/nfs-inventory.yaml
[BackupNode]
<backup_node> ansible_host=<ip_address> ansible_user=<user>
EOF
```

Поменяйте <ip_address> и <user> на значения, применимые к среде.

3. Скопируйте публичный ключ SSH с сервера **Undercloud** на сервер резервного копирования.

```
(undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <backup_node>
```

Поменяйте <backup_node> на путь и имя сервера резервного копирования.

4. Настройте сервер NFS на сервере резервного копирования:

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-nfs --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/nfs-inventory.yaml
```

26.4 Установка ReaR на сервере Undercloud

Перед созданием резервной копии сервера **Undercloud** установите и настройте **Relax and Recover (ReaR)** на **Undercloud**.

Предварительные требования

- На сервере резервного копирования установлен и настроен NFS- или SFTP-сервер.

Порядок действий

1. Получите учетные данные Undercloud с сервера **Undercloud:**

```
[stack@undercloud ~]$ source stackrc
```

Если используется настраиваемое имя **stack**, добавьте параметр **--stack** <stack_name> к команде **tripeo-ansible-inventory**.

2. Если это еще не сделано, создайте файл инвентаризации и используйте команду **tripleo-ansible-inventory** для создания статического файла инвентаризации, содержащего хосты и переменные для всех серверов **Overcloud**:

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

3. Установите **ReaR** на сервере **Undercloud**:

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-rear --extra-
vars /home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
```

4. Если система использует загрузчик **UEFI**, выполните следующие действия на сервере **Undercloud**.
   - Установите следующие инструменты:

```
$ sudo dnf install dosfstools efibootmgr
```

   - Активируйте резервное копирование UEFI в файле конфигурации **ReaR** (расположен в /etc/rear/local.conf) – поменяйте значение 0 параметра **USING_UEFI_BOOTLOADER** на значение 1.

26.5 Создание резервной копии базы данных серверов Undercloud

Резервные копии базы данных **Undercloud** можно опционально включить в регулярный график резервного копирования для дополнительной безопасности данных.

Полная резервная копия сервера **Undercloud** включает резервную копию базы данных сервера **Undercloud**.

Порядок действий

Создайте резервную копию базы данных серверов **Undercloud**:

```
openstack undercloud backup --db-only
```

Резервный файл базы данных хранится в /home/stack.

Имя резервного файла: **openstack-backup-mysql-<timestamp>.sql**.

26.6 Настройка интерфейсов Open vSwitch (OVS) для резервного копирования

Если в среде используется коммутатор Open vSwitch (OVS), то перед созданием резервной копии **Undercloud** или серверов управления необходимо вручную настроить интерфейсы **OVS**.

В процессе восстановления эта информация используется для восстановления сетевых интерфейсов.

Порядок действий

1. В файле **/etc/rear/local.conf** добавьте параметр **NETWORKING_PREPARATION_COMMANDS** в следующем формате:

```
NETWORKING_PREPARATION_COMMANDS=('<command_1>' '<command_2>' ...')
```

2. Поменяйте <command_1> и <command_2> на команды, которые настраивают имена сетевых интерфейсов или IP-адреса.

   Например, можно добавить команду **ip link add br-ctlplane type bridge**, чтобы настроить имя моста управления, или добавить команду **ip link set eth0 up**, чтобы задать имя интерфейса. В зависимости от конфигурации сети к параметру можно добавить дополнительные команды.

26.7 Создание резервной копии сервера Undercloud

Чтобы создать резервную копию сервера **Undercloud**, используйте команду **openstack undercloud backup**. Затем можно использовать резервную копию для восстановления сервера **Undercloud** до предыдущего состояния в случае, если сервер поврежден или недоступен.

Резервная копия сервера **Undercloud** включает в себя резервную копию базы данных, работающей на сервере **Undercloud**.

Предварительные требования

- На резервном сервере установлен и настроен NFS или SFTP.
- На сервере **Undercloud** установлен **ReaR**.
- Для сетевых интерфейсов используется коммутатор **OVS**.

Порядок действий

1. Зайдите в **Undercloud** как пользователь **stack**.
2. Получите пароль MySQL:

```
[stack@undercloud ~]$ PASSWORD=$(sudo /bin/hiera -c /etc/puppet/hiera.yaml
mysql::server::root_password)
```

3. Создайте резервную копию базы данных сервера **Undercloud**:

```
[stack@undercloud ~]$ sudo podman exec mysql bash -c "mysqldump -uroot -p$PASSWORD --
opt --all-databases" | sudo tee /root/undercloud-all-databases.sql
```

4. Получите учетные данные **Undercloud**:

```
[stack@undercloud ~]$ source stackrc
```

5. Если файл инвентаризации создан не был, создайте его при помощи команды **tripeo-ansible-inventory**. Это будет статический файл инвентаризации, содержащий хосты и переменные для всех серверов **Overcloud**:

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

6. Создайте резервную копию сервера **Undercloud**:

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --inventory
/home/stack/tripleo-inventory.yaml
```

26.8 Планирование резервного копирования серверов Undercloud с помощью cron

Резервное копирование серверов **Undercloud** можно запланировать с помощью **ReaR**, используя роль резервного копирования и восстановления (**backup-and-restore role**) Ansible.

Журналы можно просмотреть в каталоге /var/log/rear-cron.

Предварительные требования

- На сервере резервного копирования установлен и настроен NFS- или SFTP-сервер.
- **ReaR** установлен на серверах **Undercloud** и серверах управления.
- Для хранения резервной копии в соответствующей директории имеется достаточно свободного пространства на диске.

Порядок действий

1. Чтобы запланировать резервное копирование серверов управления по расписанию, выполните следующую команду (расписание по умолчанию – в полночь по воскресеньям):

```
$ openstack undercloud backup --cron
```

2. Опционально: настройте запланированное резервное копирование в соответствии с развертыванием:
   - Чтобы изменить расписание резервного копирования по умолчанию, укажите иное расписание cron в параметре **tripeo_backup_and_restore_cron**.

```
openstack undercloud backup --cron --extra-
   vars'{"tripleo_backup_and_restore_cron": "0 0 * * 0"}'
```

   - Чтобы определить дополнительные параметры, которые добавляются к команде резервного копирования, когда cron запускает резервное копирование по расписанию, передайте параметр **tripeo_backup_and_restore_cron_extra** в команду резервного копирования:

```
openstack undercloud backup -cron --extra-vars
   '{"tripleo_backup_and_restore_cron_extra":"-extra-vars bar-
   vars.yaml --inventory /home/stack/tripleo-inventory.yaml"}'
```

   - Чтобы изменить пользователя по умолчанию, выполняющего резервное копирование, передайте параметр **tripeo_backup_and_restore_cron_user** в команду резервного копирования:

```
openstack undercloud backup --cron --extra-vars
   '{"tripleo_backup_and_restore_cron_user": "root"}
```

27 Резервное копирование серверов управления

Для резервного копирования серверов управления настройте сервер резервного копирования, установите инструмент **Relax-and-Recover** на серверах управления и создайте образ резервной копии. Создавать резервные копии можно при регулярном обслуживании среды.

27.1 Поддерживаемые форматы и протоколы резервного копирования0

В процессе резервного копирования и восстановления **Undercloud** используется инструмент с открытым исходным кодом **Relax-and-Recover (ReaR)** для создания и восстановления загружаемых образов резервных копий. **ReaR** написан на Bash и поддерживает многие форматы образов и транспортных протоколов.

Форматы и протоколы резервного копирования, которые поддерживает платформа 1Stack при использовании **ReaR** для резервного копирования и восстановления **Undercloud** и области управления:

Форматы загрузочных носителей:

- ISO.

Протоколы передачи файлов:

- SFTP,
- NFS.

27.2 Установка и настройка сервера NFS на резервном сервере

Можно установить и настроить новый сервер NFS для хранения файла резервной копии. Чтобы установить и настроить сервер NFS на резервном сервере, создайте файл инвентаризации и ключ SSH и запустите команду резервного копирования **openstack undercloud** с параметрами сервера NFS.

- Если сервер NFS или SFTP уже установлен и настроен, выполнять приведенный ниже порядок действий не требуется. Введите информацию о сервере при настройке **ReaR** на сервере, резервную копию которого нужно создать.
- По умолчанию параметр IP-адреса **Relax** and **Recover (ReaR)** для сервера NFS – 192.168.24.1. Чтобы установить значение IP-адреса, соответствующее среде, необходимо добавить параметр **tripeo_backup_and_restore_server**.

Порядок действий

1. На сервере **Undercloud** примените учетные данные **Undercloud**:

```
[stack@undercloud ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

2. На сервере **Undercloud** создайте файл инвентаризации для сервера резервного копирования:

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF'> ~/nfs-inventory.yaml
[BackupNode]
<backup_node> ansible_host=<ip_address> ansible_user=<user>
EOF
```

Поменяйте <ip_address> и <user> на значения, применимые для среды.

3. Скопируйте публичный ключ SSH с сервера **Undercloud** на сервер резервного копирования:

```
(undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <backup_node>
```

Поменяйте <backup_node> на путь и имя сервера резервного копирования.

4. Настройте сервер NFS на сервере резервного копирования:

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-nfs --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/nfs-inventory.yaml
```

27.3 Установка ReaR на серверах управления

Перед созданием резервной копии сервера управления **Overcloud** установите и настройте **Relax and Recover (ReaR)** на каждом сервере управления.

Из-за существующей проблемы резервное копирование **ReaR** серверов **Overcloud** продолжается даже в том случае, если сервер управления не работает.
Перед запуском резервного копирования **ReaR**, убедитесь, что все серверы управления работают.

Предварительные требования

- На резервном сервере установлен и настроен сервер NFS или SFTP.

Порядок действий

1. На сервере **Undercloud** примените учетные данные **Undercloud**:

```
[stack@undercloud ~]$ source stackrc
```

2. Если это еще не сделано, создайте файл инвентаризации и используйте команду **tripleo-ansible-inventory** для создания статического файла инвентаризации, содержащего хосты и переменные для всех серверов **Overcloud**:

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

3. Установите **ReaR** на серверах управления:

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup --setup-rear --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
```

4. В файле **bar-vars.yaml** настройте место хранения резервной копии:
   - Если установлен и настроен собственный NFS-сервер, добавьте параметр **tripeo_backup_and_restore_server** и установите в качестве значения IP-адрес NFS-сервера:

```
tripleo_backup_and_restore_server: <ip_address>
```

     По умолчанию значение параметра **tripleo_backup_and_restore_server** – 192.168.24.1.
   - Если используется SFTP-сервер, добавьте параметр **tripeo_backup_and_restore_output_url** и задайте значения URL и учетных данных SFTP-сервера:

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

     Поменяйте <user>, <password> и <backup_node> на URL и учетные данные сервера резервного копирования.
5. Если в системе используется загрузчик **UEFI**, выполните следующие действия на серверах управления:
   - Установите следующие инструменты:

```
$ sudo dnf install dosfstools efibootmgr
```

   - Активируйте резервное копирование UEFI в файле конфигурации **ReaR** (расположен в /etc/rear/local.conf) – поменяйте значение 0 параметра **USING_UEFI_BOOTLOADER** на значение 1.

27.4 Настройка Open vSwitch (OVS) для резервного копирования

Если в среде используется коммутатор Open vSwitch (**OVS**), то перед созданием резервной копии серверов **Undercloud** или серверов управления необходимо настроить вручную интерфейсы **OVS**.

В процессе восстановления эта информация используется для восстановления сетевых интерфейсов.

Порядок действий

1. В **файле /etc/rear/local.conf** добавьте параметр **NETWORKING_PREPARATION_COMMANDS** в следующем формате:

```
NETWORKING_PREPARATION_COMMANDS=('<command_1>' '<command_2>' ...')
```

2. Поменяйте <command_1> и <command_2> на команды, которые настраивают имена сетевых интерфейсов или IP-адреса.

   Например, можно добавить команду **ip link add br-ctlplane type bridge**, чтобы настроить имя моста управления, или добавить команду **ip link set eth0 up**, чтобы задать имя интерфейса. В зависимости от конфигурации сети к параметру можно добавить дополнительные команды.

27.5 Создание резервной копии серверов управления

Чтобы создать резервную копию серверов управления, используйте команду **openstack overcloud backup**. Затем можно использовать резервную копию для восстановления серверов управления до их предыдущего состояния в случае, если серверы повреждены или недоступны.

Резервное копирование серверов управления включает резервное копирование базы данных, которая работает на этих серверах.

Предварительные требования

- На сервере резервного копирования установлен и настроен NFS- или SFTP-сервер.
- На серверах управления установлен **ReaR**.
- Для сетевых интерфейсов используется коммутатор **OVS**.

Порядок действий

1. Найдите раздел **config-drive** на каждом сервере управления:

```
[stack@undercloud ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    253:0    0   55G  0 disk
├─vda1 253:1    0    1M  0 part
├─vda2 253:2    0  100M  0 part /boot/efi
└─vda3 253:3    0 54.9G  0 part /
```

   Раздел **config-drive** – это раздел размером 1 МБ, который не смонтирован.

2. На каждом сервере управления создайте резервную копию раздела **config-drive** каждого сервера от имени пользователя **root**:

```
[root@controller-x ~]# dd if=<config_drive_partition> of=/mnt/config-drive
```

   Поменяйте <config_drive_partition> на имя раздела config-drive, который был найден на шаге 1.

3. На сервере **Undercloud** примените учетные данные **Undercloud**:

```
[stack@undercloud ~]$ source stackrc
```

4. Если файл инвентаризации создан не был, создайте его при помощи команды **tripeo-ansible-inventory.** Это будет статический файл инвентаризации, содержащий хосты и переменные для всех серверов **Overcloud**:

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

5. Создайте резервную копию серверов управления:

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup --inventory
/home/stack/tripleo-inventory.yaml
```

Процесс резервного копирования выполняется последовательно на каждом сервере управления, не нарушая при этом работу сервиса в среде.

27.6 Планирование резервного копирования серверов управления с помощью cron

Запланировать резервное копирование серверов управления можно с помощью **ReaR**, используя роль **backup-and-restore** в Ansible.

Журналы можно просмотреть в каталоге /var/log/rear-cron.

Предварительные требования

- На сервере резервного копирования установлен и настроен NFS- или SFTP-сервер.
- **ReaR** установлен на серверах **Undercloud** и серверах управления.
- Для хранения резервной копии в соответствующей директории имеется достаточно свободного пространства на диске.

Порядок действий

1. Чтобы запланировать резервное копирование серверов управления по расписанию, выполните следующую команду (расписание по умолчанию – в полночь по воскресеньям):

```
openstack overcloud backup --cron
```

2. Опционально: настройте резервное копирование по расписанию в соответствии с развертыванием:
   - Чтобы изменить расписание резервного копирования по умолчанию, передайте другое расписание cron в параметре **tripeo_backup_and_restore_cron**:

```
openstack overcloud backup --cron --extra-
   vars'{"tripleo_backup_and_restore_cron": "0 0 * * 0"}'
```

- Чтобы определить дополнительные параметры, которые добавляются к команде резервного копирования, когда cron запускает резервное копирование по расписанию, передайте параметр **tripeo_backup_and_restore_cron_extra** в команду резервного копирования:

```
openstack overcloud backup --cron --extra-vars
    '{"tripleo_backup_and_restore_cron_extra":"--extra-vars bar-
    vars.yaml --inventory /home/stack/tripleo-inventory.yaml"}'
```

- Чтобы изменить пользователя по умолчанию, выполняющего резервное копирование, передайте параметр **tripeo_backup_and_restore_cron_user** в команду резервного копирования:

```
openstack overcloud backup --cron --extra-vars
    '{"tripleo_backup_and_restore_cron_user": "root"}
```

28 Восстановление серверов Undercloud и серверов управления

Если серверы управления **Undercloud** или **Overcloud** повреждены или если возникает ошибка, серверы можно восстановить из резервной копии до предыдущего состояния.

Если в процессе восстановления не удается автоматически восстановить кластер **Galera**, можно восстановить эти компоненты вручную.

28.1 Восстановление сервера Undercloud

Восстановить сервер **Undercloud** в предыдущее состояние можно с использованием резервного ISO-образа, созданного с помощью **ReaR**.

Резервные ISO-образы находятся на сервере резервного копирования. Запишите загрузочный ISO-образ на DVD-диск или загрузите его на сервер **Undercloud** через удаленный доступ Integrated Lights-Out (iLO).

Предварительные требования

- Создана резервная копия сервера **Undercloud**.
- Имеется доступ к серверу резервного копирования.
- Если используется коммутатор **OVS** для сетевых интерфейсов, имеется доступ к информации о конфигурации сети, которая была указана для параметра **NETWORKING_PREPARATION_COMMANDS**.

Порядок действий

1. Выключите сервер **Undercloud**. Убедитесь, что сервер **Undercloud** отключен полностью.
2. Загрузите сервер **Undercloud** с резервным ISO-образом.
3. Когда отобразится меню загрузки **Relax-and-Recover**, выберите **ReaR** <undercloud_node>. Поменяйте <undercloud_node> на имя сервера **Undercloud**.

> Если система использует **UEFI**, выберите опцию **Relax-and-Recover** (no Secure Boot).

4. Авторизуйтесь как пользователь **root** и восстановите сервер:

   Отобразится следующее сообщение:

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <undercloud_node>:~ # rear recover
```

Когда процесс восстановления сервера **Undercloud** завершится, в консоли отобразится следующее сообщение:

Finished recovering your systemExiting rear recoverRunning exit tasks

5. Отключите сервер:

```
RESCUE <undercloud_node>:~ # poweroff
```

При загрузке сервер возобновит свое предыдущее состояние.

28.2 Восстановление серверов управления

Восстановить серверы управления в их предыдущее состояние можно с использованием резервного ISO-образа, созданного с помощью **ReaR**.

Чтобы восстановить область управления, необходимо восстановить все серверы управления, чтобы обеспечить согласованность состояния.

Резервные ISO-образы находятся на сервере резервного копирования. Запишите загрузочный ISO-образ на DVD-диск или загрузите его на сервер **Undercloud** через удаленный доступ Integrated Lights-Out (iLO).

Предварительные требования

- Создана резервная копия сервера управления.
- Имеется доступ к серверу резервного копирования.
- Если используется коммутатор **OVS** для сетевых интерфейсов, имеется доступ к информации о конфигурации сети, которая была указана в параметре **NETWORKING_PREPARATION_COMMANDS**.

Порядок действий

1. Выключите каждый сервер управления. Убедитесь, что серверы управления отключены полностью.
2. Загрузите каждый сервер управления с помощью соответствующего резервного образа ISO.
3. При появлении меню загрузки **Relax-and-Recover**, на каждом сервере управления выберите **Recover** <control_plane_node>. Поменяйте <control_plane_node> на имя соответствующего сервера управления.

> Если система использует **UEFI**, выберите опцию **Relax-and-Recover** (no Secure Boot).

4. Авторизуйтесь как пользователь **root** на каждом сервере управления и восстановите сервер:

   Отобразится следующее сообщение:

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <control_plane_node>:~ # rear recover
```

Когда процесс восстановления сервера управления будет завершен, на консоли отобразится сообщение:

Finished recovering your system
Exiting rear recover
Running exit tasks

5. Когда консоль командной строки станет доступна, восстановите раздел **config-drive** каждого сервера управления:

```
# once completed, restore the config-drive partition (which is ISO9660)
RESCUE <control_plane_node>:~ $ dd if=/mnt/local/mnt/config-drive
of=<config_drive_partition>
```

6. Выключите сервер:

```
RESCUE <control_plane_node>:~ # poweroff
```

7. Установите последовательность загрузки на обычное загрузочное устройство. При загрузке сервер возобновит свое предыдущее состояние.
8. Чтобы убедиться в корректной работе сервисов, проверьте состояние pacemaker. Войдите на сервер управления как пользователь **root** и введите следующую команду:

```
# pcs status
```

9. Чтобы просмотреть статус **Overcloud**, используйте **OpenStack Integration Test Suite** (tempest).

Исправление возможных ошибок

- Удалите аварийные оповещения ресурсов, отображаемые статусом **pcs** – укажите следующую команду:

```
# pcs resource clean
```

- Удалите ошибки действия ограждения STONITH, отображаемые **pcs** – укажите следующие команды:

```
# pcs resource clean
# pcs stonith history cleanup
```

28.3 Восстановление кластера Galera вручную

Если кластер **Galera** не восстанавливается, как указано ниже в разделе «Порядок действий» по восстановлению, необходимо восстановить его вручную.

> В соответствии с разделом «Порядок действий», необходимо выполнить определенные действия на одном сервере управления. Убедитесь, что вы выполняете эти действия на том же сервере управления при выполнении «Порядка действий».

Порядок действий

1. На Controller-0 получите виртуальный IP-адрес кластера **Galera**:

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql_vip
```

2. Отключите соединения с базой данных через виртуальный IP на всех серверах управления:

```
$ sudo iptables -I INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

3. На Controller-0 получите пароль **root** MySQL.

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql::server::root_password
```

4. На Controller-0 переведите ресурс **Galera** в неуправляемый режим:

```
$ sudo pcs resource unmanage galera-bundle
```

5. Остановите контейнеры MySQL на всех серверах управления:

```
$ sudo podman container stop $(sudo podman container ls --all --format ".Names" --
filter=name=galera-bundle)
```

6. Переместите директорию на все серверы управления:

```
$ sudo mv /var/lib/mysql /var/lib/mysql-save
```

7. Создайте новую директорию /var/lib/mysq на всех серверах управления:

```
$ sudo mkdir /var/lib/mysql
$ sudo chown 42434:42434 /var/lib/mysql
$ sudo chcon -t container_file_t /var/lib/mysql
$ sudo chmod 0755 /var/lib/mysql
$ sudo chcon -r object_r /var/lib/mysql
$ sudo chcon -u system_u /var/lib/mysql
```

8. Запустите контейнеры MySQL на всех серверах управления:

```
$ sudo podman container start $(sudo podman container ls --all --format "{{ .Names }}"
--filter=name=galera-bundle)
```

9. Создайте базу данных MySQL на всех серверах управления:

```
$ sudo podman exec -i $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mysql_install_db --datadir=/var/lib/mysql --
user=mysql --log_error=/var/log/mysql/mysql_init.log"
```

10. Запустите базу данных на всех серверах управления:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mysqld_safe --skip-networking --wsrep-on=OFF --log-
error=/var/log/mysql/mysql_safe.log" &
```

11. Переместите файл конфигурации **.my.cnf Galera** на все серверы управления:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mv /root/.my.cnf /root/.my.cnf.bck"
```

12. Сбросьте пароль **root Galera** на всех серверах управления:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mysql -uroot -e'use mysql;update user set
password=PASSWORD(\"$ROOTPASSWORD\")where User=\"root\";flush privileges;'"
```

13. Восстановите файл конфигурации **.my.cnf Galera** внутри контейнера **Galera** на всех серверах управления:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mv /root/.my.cnf.bck /root/.my.cnf"
```

14. На Controller-0 скопируйте файлы резервной базы данных на /var/lib/MySQL:

```
$ sudo cp $BACKUP_FILE /var/lib/mysql$ sudo cp $BACKUP_GRANT_FILE /var/lib/mysql
```

Путь к этим файлам /home/heat-admin/.

15. На Controller-0 восстановите базу данных MySQL:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
\"/var/lib/mysql/$BACKUP_FILE\" "
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
\"/var/lib/mysql/$BACKUP_GRANT_FILE\" "
```

16. Выключите базы данных на всех серверах управления:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "mysqladmin shutdown"
```

17. На Controller-0 запустите сервер начальной загрузки.

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) \
/usr/bin/mysqld_safe --pid-file=/var/run/mysql/mysqld.pid --
socket=/var/lib/mysql/mysql.sock --datadir=/var/lib/mysql \
--log-error=/var/log/mysql/mysql_cluster.log --user=mysql --open-files-limit=16384 --
wsrep-cluster-address=gcomm:// &
```

18. Проверка: на Controller-0, проверьте статус кластера:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "clustercheck"
```

Убедитесь, что отображается следующее сообщение: «Galera cluster node is synced», в противном случае потребуется заново создать сервер.

19. На Controller-0 получите адрес кластера из конфигурации:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "grep wsrep_cluster_address
/etc/my.cnf.d/galera.cnf" | awk '{print $3}'
```

20. На каждом из оставшихся серверов управления запустите базу данных и проверьте кластер:

- Запустите базу данных:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) /usr/bin/mysqld_safe --pid-file=/var/run/mysql/mysqld.pid --
```

```
socket=/var/lib/mysql/mysql.sock --datadir=/var/lib/mysql --log-
error=/var/log/mysql/mysql_cluster.log --user=mysql --open-files-limit=16384 --wsrep-
cluster-address=$CLUSTER_ADDRESS &
```

- Проверьте статус кластера MYSQL:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) bash -c "clustercheck"
```

Убедитесь, что отображается следующее сообщение: «Galera cluster node is synced», в противном случае потребуется заново создать сервер.

21. Остановите контейнер MySQL на всех серверах управления:

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) \/usr/bin/mysqladmin -u root shutdown
```

22. На всех серверах управления удалите следующее правило межсетевого экрана, чтобы разрешить подключения к базе данных через виртуальный IP-адрес:

```
$ sudo iptables -D INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

23. Перезапустите контейнер MySQL на всех серверах управления.

```
$ sudo podman container restart $(sudo podman container ls --all --format "{{ .Names
}}" --filter=name=galera-bundle)
```

24. Перезапустите контейнер кластерной проверки на всех серверах управления.

```
$ sudo podman container restart $(sudo podman container ls --all --format "{{ .Names
}}" --filter=name=clustercheck)
```

25. На Controller-0 переведите ресурс **Galera** в управляемый режим:

```
$ sudo pcs resource manage galera-bundle
```

Проверка

- Чтобы убедиться в корректной работе сервисов, проверьте состояние pacemaker:

```
$ sudo pcs status
```

- Чтобы просмотреть статус **Overcloud**, используйте **OpenStack Integration Test Suite** (tempest).
- Чтобы посмотреть, имеется ли проблема с конкретным сервером, проверьте состояние кластера с помощью кластерной проверки:

```
$ sudo podman exec clustercheck /usr/bin/clustercheck
```

28.4 Восстановление базы данных сервера Undercloud вручную

Если база данных Undercloud не восстанавливается в рамках процесса восстановления **Undercloud**, вы можете восстановить базу данных вручную. Вы можете восстановить базу данных только в том случае, если ранее вы создали отдельную резервную копию базы данных.

Предварительные требования

- Создана отдельная резервная копия базы данных **Undercloud**.

Порядок действий

1. Войдите на север **Undercloud** как пользователь **root**.
2. Остановите все сервисы tripleo:

```
[root@undercloud ~]# systemctl  stop  tripleo_*
```

3. Убедитесь, что на сервере не работают контейнеры – укажите следующую команду:

```
[root@undercloud ~]# podman ps
```

4. Если какие-либо контейнеры запущены, укажите следующую команду для их остановки:

```
[root@undercloud ~]# podman stop <container_name>
```

5. Создайте резервную копию директории /var/lib/mysql, а затем директорию:

```
[root@undercloud ~]# cp -a /var/lib/mysql /var/lib/mysql_bck
[root@undercloud ~]# rm -rf /var/lib/mysql
```

6. Воссоздайте директорию базы данных и установите атрибуты **SELinux** для новой директории:

```
[root@undercloud ~]# mkdir /var/lib/mysql
[root@undercloud ~]# chown 42434:42434 /var/lib/mysql
[root@undercloud ~]# chmod 0755 /var/lib/mysql
[root@undercloud ~]# chcon -t container_file_t /var/lib/mysql
[root@undercloud ~]# chcon -r object_r /var/lib/mysql
[root@undercloud ~]# chcon -u system_u /var/lib/mysql
```

7. Создайте локальный тег для образа **mariadb**. Поменяйте <image_id> и <undercloud.ctlplane.example.com> на значения, применимые в среде:

```
[root@undercloud ~]# podman images | grep mariadb
<undercloud.ctlplane.example.com>:8787/1Stack/centos-binary-
mariadb:pcmklatest                    16.2_20210322.1   <image_id>   3 weeks ago   718
MB
[root@undercloud ~]# podman tag <image_id> mariadb
[root@undercloud ~]# podman images | grep maria
localhost/mariadb   latest  <image_id>   3 weeks ago   718 MB
<undercloud.ctlplane.example.com>:8787/1Stack/centos-binary-
mariadb:pcmklatest    16.2_20210322.1    <image_id>  3 weeks ago 718 MB
```

8. Инициализируйте каталог /var/lib/mysql с контейнером:

```
[root@undercloud ~]# podman run --net=host -v /var/lib/mysql:/var/lib/mysql
localhost/mariadb mysql_install_db --datadir=/var/lib/mysql --user=mysql
```

9. Скопируйте файл резервной копии базы данных, который нужно импортировать в базу данных:

```
[root@undercloud ~]# cp /root/undercloud-all-databases.sql /var/lib/mysql
```

10. Запустите сервис базы данных для импорта данных:

```
[root@undercloud ~]# podman run --net=host -dt -v /var/lib/mysql:/var/lib/mysql
localhost/mariadb  /usr/libexec/mysqld
```

11. Импортируйте данные и настройте параметр **max_allowed_packet**:

- Войдите в контейнер и настройте его:

```
[root@undercloud ~]# podman exec -it <container_id> /bin/bash
()[mysql@5a4e429c6f40 /]$ mysql -u root -e "set global max_allowed_packet =
1073741824;"
()[mysql@5a4e429c6f40 /]$ mysql -u root < /var/lib/mysql/undercloud-all-databases.sql
()[mysql@5a4e429c6f40 /]$ mysql -u root -e 'flush privileges'
()[mysql@5a4e429c6f40 /]$ exit
```

- Остановите контейнер:

```
[root@undercloud ~]# podman stop <container_id>
```

- Убедитесь, что ни один контейнер не запущен:

```
[root@undercloud ~]# podman ps
CONTAINER ID  IMAGE  COMMAND  CREATED  STATUS  PORTS  NAMES
[root@undercloud ~]#
```

12. Перезапустите все сервисы **tripleo**:

```
[root@undercloud ~]# systemctl start multi-user.target
```

Термины, сокращения и определения

Термин	Определение
API	Application Programming Interface. Программный интерфейс
CA	Certification Authority. Центр сертификации, удостоверяющий центр
Ceph	Распределенная система хранения данных
CIDR	Classless Inter-Domain Routing. Бесклассовая междоменная маршрутизация. Метод IP-адресации, позволяющий гибко управлять пространством IP-адресов
CLI	Command Line Interface. Интерфейс командной строки
CRL	Certificate Revocation List. Список отзыва сертификатов
DHCP	Dynamic Host Configuration Protocol. Протокол динамического конфигурирования узлов
DNS	Domain Name System. Система доменных имен
HA	High Availability. Высокая доступность
HMT	Hierarchical Multitenancy. Иерархическая многопользовательская среда
HTTP	HyperText Transfer Protocol. Протокол передачи гипертекста
ICMP	Internet Control Message Protocol. Протокол межсетевых управляющих сообщений. Передает сообщения об ошибках и других исключительных ситуациях, которые возникают при передаче данных
ID	Идентификатор
iLO	Integrated Lights-Out. Инструмент удаленного управления серверами
IP	Интернет-протокол
IPMI	Intelligent Platform Management Interface. Интеллектуальный интерфейс управления платформой
ISO-образ	Архивный файл, который содержит идентичную копию (образ) данных
1Stack (1Стек)	Облачная платформа виртуализации. Представляет собой программное решение для управления виртуализированными ресурсами: ▪ процессорной мощности и оперативной памяти физических серверов; ▪ сети передачи данных; ▪ систем хранения данных и предоставления пользователю этих ресурсов в пределах выделенной квоты, с поддержкой виртуальной и физической изоляции ресурсов между различными пользователями
JSON	JavaScript Object Notation. Текстовый формат обмена данными, основанный на JavaScript
LDAP	Lightweight Directory Access Protocol. Протокол быстрого доступа к каталогам

Термин	Определение
LLDP	Link Layer Discovery Protocol. Протокол канального уровня, который позволяет сетевым устройствам анонсировать в сеть информацию о себе и о своих возможностях, а также собирать информацию о соседних устройствах
MAC-адрес	Media Access Control Address. Уникальный идентификатор. Присваивается каждому сетевому оборудованию
MariaDB	Реляционная система управления базами данных с открытым исходным кодом и бесплатной лицензией
MTU	Maximum Transmission Unit. Максимальная единица передачи
MySQL	Система управления базами данных
NFS	Network File System. Сетевая файловая система
NIC	Network Interface Card. Сетевая интерфейсная карта
NUMA	Non-Uniform Memory Access. Архитектура организации компьютерной памяти. Используется в мультипроцессорных системах
OpenLDAP	OpenLDAP. Протокол облегченного доступа к каталогам с открытым исходным кодом
OVN	Open Virtual Network. Платформа сетевой виртуализации, которая отделяет физическую топологию сети от логической
OVS	Open vSwitch. Многоуровневый программный коммутатор. Используется для работы в качестве виртуального коммутатора в средах виртуальных машин
PCI	Peripheral Component Interconnect. Локальная компьютерная шина для подключения аппаратных устройств к компьютеру
PF	Порт физической функции
PXE	Preboot eXecution Environment. Среда для загрузки компьютера с помощью сетевой карты
QEMU	Программа с открытым исходным кодом для эмуляции аппаратного обеспечения различных платформ
QoS	Quality of Service. Набор технологических решений для оптимизации сетевого трафика с помощью назначаемых приоритетов передачи информации
RA	Router Advertisement. Сообщение, инициируемое маршрутизатором
RabbitMQ	Программный брокер сообщений
RADOS	Reliable Autonomic Distributed Object Store. Распределенная объектная система хранения данных файловой системы Ceph
RBAC	Role-Based Access Control. Управление доступом на основе ролей
RBD	RADOS Block Device. Программное обеспечение с открытым исходным кодом для хранения данных на основе блочного устройства в распределенных системах хранения

Термин	Определение
REST API	Способ доступа к веб-сервисам без какой-либо обработки
SCTP	Stream Control Transmission Protocol. Протокол транспортного уровня в компьютерных сетях
SDN	Software-Defined Networking. Программно-определяемые сети
SELinux	Security-Enhanced Linux. Система контроля доступа Linux
SFTP	Secure File Transfer Protocol. Протокол безопасной передачи файлов через сеть
SLAAC	Stateless Address Autoconfiguration. Автоматическая конфигурация
SR-IOV	Single Root Input/Output Virtualization. Виртуализация ввода-вывода. Применяется для виртуализации ресурсов ввода-вывода для отдельных серверов
SSH	Secure Shell. Безопасная оболочка – сетевой протокол прикладного уровня. Позволяет удаленно управлять операционной системой и туннелировать TCP-соединения
SSL	Secure Sockets Layer. Протокол безопасности, который создает зашифрованное соединение между веб-сервером и веб-браузером
STONITH	Техника ограждения (изоляции) вышедшего из строя узла кластера
TCP	Transmission Control Protocol. Протокол управления передачей данных
TLS	Transport Layer Security. Криптографический протокол обеспечения безопасной передачи данных
UDP	User Datagram Protocol. Сетевой протокол транспортного уровня. Использует IP для передачи данных от одного устройства к другому. Данные (датаграммы), которые вносятся в пакет UDP, включают порты назначения, источник, контрольную сумму и длину пакета
UEFI	Unified Extensible Firmware Interface. Унифицированный расширяемый интерфейс встроенного программного обеспечения. Инициализирует оборудование при включении системы и передает управление загрузчику или ядру операционной системы
URI	Uniform Resource Identifier. Унифицированный идентификатор ресурса
UUID	Универсально уникальный идентификатор. 128-битная метка, используемая для идентификации информации
vCPU	Virtual Central Processing Unit. Виртуализированный вариант физического CPU – центральные блоки управления в виртуальных машинах и облачных средах
VF	Virtual Function. Виртуальная функция
VIP	Virtual IP address. Виртуальный IP-адрес – компонент сетевой и интернет-инфраструктуры, который обеспечивает балансировку нагрузки, высокую доступность и эффективное распределение ресурсов в вычислительной среде. Это уникальная числовая метка, присвоенная виртуальной машине или службе, а не физическому устройству

Термин	Определение
VLAN	Virtual Local Area Network. Виртуальная локальная сеть
VNC	Virtual Network Computing. Метод удаленного доступа к рабочему столу компьютера по сети
vNIC	Virtual Network Interface Card. Виртуальный сетевой интерфейс, основанный на физических сетевых картах узла
VoIP	Voice over Internet Protocol. Технология передачи голосовых сообщений в локальных сетях или в сети Интернет с использованием протокола IP
VRRP	Virtual Router Redundancy Protocol. Сетевой протокол, предназначенный для увеличения доступности маршрутизаторов, которые выполняют роль шлюза по умолчанию
VXLAN	Virtual Extensible LAN. Технология сетевой виртуализации для решения проблем масштабируемости в больших системах облачных вычислений
WinRM	Windows Remote Management. Технология Microsoft, которая позволяет удаленно управлять компьютерами Windows через защищенное соединение
YAML	Yet Another Markup Language. Формат сериализации данных. Используется при управлении конфигурацией, а также для хранения данных в структурированном формате
ОЗУ	Оперативная память
ОС	Операционная система
ПО	Программное обеспечение
ЦПУ	Центральный процессор