Программное обеспечение «Базис.Cloud». Руководство пользователя

RU.НРФЛ.00004-01.94.01

Москва

12/14/2022

Содержание

1 Требования назначения

Бесперебойная работа дата-центра имеет решающее значение для безотказной работы бизнес-приложений. Меры по повышению надежности (безотказности) инфраструктуры должны проявляться в обеспечении доступа к данным в любое время, на всём протяжении эксплуатации облачной инфраструктуры.
На бесперебойность влияет множество эксплуатационных факторов – расчетные показатели качества и надежности, а также потребительские свойства.
Показатели качества и надежности, также как потребительские свойства, определяются преимущественно *референсной архитектурой* ПО.
Дополнительно может применяться комплекс организационно-технических мер, нацеленных на повышение операционной эффективности в центре обработки данных (серия стандартов ИСО/МЭК 30134). В качестве ключевых показателей экономической эффективности могут быть зафиксированы:

- эффективность использования электроэнергии;
- коэффициент использования ИТ-оборудования для серверов.

Ниже описаны требования, определяющие использование ПО облачной платформы по назначению:

- показатели качества и надежности;
- модель обслуживания облака;
- потребительские свойства платформы.
- потребительские свойства платформы.

1.1 Показатели качества и надежности

Надежность инфраструктуры принято оценивать по следующим критериям:
– доступность (Availability): дата-центры должны обеспечивать доступность запрашиваемой информации;
– безопасность (Security): в дата-центре должны быть установлены правила, процедуры и надлежащая интеграция ключевых компонентов, с целью предотвращения несанкционированного доступа к информации;
– масштабируемость (Scalability): развитие бизнеса требует развертывания большого количества серверов, частой установки новых приложений и создания многочисленных баз данных;
– производительность (Performance): все основные компоненты дата-центра должны обеспечивать оптимальную производительность, в соответствии с необходимыми уровнями обслуживания;
– целостность данных (Data integrity): могут применяться такие механизмы, как коды коррекции ошибок или биты контроля четности, гарантирующие хранение и извлечение данных точно в таком же виде, в котором они были получены;
– объем (Capacity): для эффективного хранения и обработки большого количества данных операции дата-центра требуют соответствующих ресурсов;
– управляемость (Manageability): дата-центр должен обеспечивать простое и интегрированное управление всеми своими компонентами.

Отказоустойчивый кластер (сокр. кластер) – группа серверов, взаимодействие между которыми спроектировано в соответствии с методом обеспечения высокой доступности (Availability). Кластер гарантирует минимальное время простоя за счёт аппаратной избыточности.
Критерий масштабируемости (Scalability) показывает, что имеющиеся ресурсы дата-центра должны позволять инфраструктуре расширяться, в соответствии с растущими потребностями Заказчика (облачной инфраструктуры). При этом расширение не должно отрицательно сказываться на осуществлении бизнес-операций (т.н. обеспечение непрерывности бизнес-процессов).
При увеличении требований к объему (Capacity) дата-центр должен предоставлять дополнительный объем без ущерба для доступности данных или, в крайнем случае, с минимальным ущербом. Объемом можно управлять путем перераспределения существующих или добавления новых ресурсов.
Хорошая управляемость (Manageability) может достигаться путем автоматизации рутинных операций и снижения роли человека (ручного управления) при выполнении типовых заданий.

1.2 Модель обслуживания облака

Для предоставления услуг выбрана модель обслуживания – IaaS, которая соответствует *частному облаку*, построенному преимущественно программными средствами из состава ПО OpenStack и программных продуктов BASIS.
Частное облако (англ. private cloud) – инфраструктура, предназначенная для использования одной организацией, включающей несколько потребителей (например, подразделений одной организации), возможно также клиентами и подрядчиками данной организации.
Частное облако может находиться в собственности, управлении и эксплуатации как самой организации, так и третьей стороны (или какой-либо их комбинации), и оно может физически существовать как

внутри, так и вне юрисдикции владельца.

1.3 Потребительские свойства платформы

Основными *потребительскими свойствами* платформы, используемой для накопления и обработки данных, являются объем (данных) и мощность (количество виртуальных процессорных ядер). Качество предоставления услуг опирается на высокую производительность систем и подсистем передачи данных.
Вычислительные узлы обеспечивают функционирование гипервизора и *горизонтальное масштабирование* (введение новых ВУ в эксплуатацию). Распределение гипервизоров на большое число ВУ позволяет регулировать и перераспределять нагрузку, связанную с использованием ресурсов (вычислительной мощности, памяти).
Управляющие узлы обеспечивают физическое резервирование друг друга, а также несут функции контроля, распределения и перераспределения ресурсов ОП. Они могут быть кластеризованы, с целью повышения *отказоустойчивости* облака.
Под каждый отдельно взятый проект выбирается определенный производитель того или иного вида аппаратного обеспечения. Унификация заказных спецификаций, реализуемых в виде ПАК, позволит снизить издержки на обслуживании (стоимости владения инфраструктурой).
Используя веб-интерфейс, администратор может выполнять манипуляции по формированию IT-ландшафта, в рамках назначенного ему проекта и выделенных IaaS-ресурсов.

2 Условия применения

Администратор – должностное лицо, служебная деятельность которого связана с использованием программных продуктов BASIS и стороннего ПО, используемого при создании среды функционирования (ОС, Python3, OpenStack и др.).
Администратор системы может приступать к своей работе после того как облачная инфраструктура введена в эксплуатацию. Для этого потребуется выполнить вход в интерфейс управления, используя выданные *реквизиты доступа*.

2.1 Квалификация администратора

Для выполнения задач по сопровождению облачной платформы, построенной на основе Базис.Cloud, администратор должен иметь опыт работы, связанный с системным администрированием серверного оборудования, а также понимать основные принципы резервного копирования и восстановления данных.
Администратор облачной платформы должен обладать навыками работы с операционными системами Linux, базовыми знаниями в следующих предметных областях информатики:
– технологии виртуализации (VMware, HyperV и т.п.);
– облачные технологии и референсные архитектуры (OpenStack, Open vSwitch).
Администратор должен уметь использовать множество утилит и инструментов облачной платформы с целью:

- контроля работоспособности облачной платформы (проверки основного функционала);
- проверки работоспособности отдельных системных служб (ОС Linux);
- конфигурирования виртуальных сервисов облачной платформы;
- резервного копирования и восстановления виртуальных машин.

Администратор облачной платформы должен обладать навыками работы с операционными системами Linux, базовыми знаниями в следующих предметных областях информатики:

– технологии виртуализации (VMware, HyperV и т.п.);

– облачные технологии и референсные архитектуры (OpenStack, Open vSwitch).

Желательно, чтобы администратор обладал практическим опытом работы с программными продуктами, выполненными на базе OpenStack.
В части, касающейся эксплуатации BASIS Workplace, администратор должен понимать теоретические основы виртуализации гостевых ОС и обладать пракическими навыками в следующих областях IT:

– протоколы удаленного доступа и средства управления виртуальными машинами;

– установка операционных систем Windows/Linux, настройка сетевого интерфейса, веб-браузера;

– основы администрирования инфраструктуры BASIS и управление VDI (см. стр. 8) проектом.

Администратор, закрепленный за обслуживаением инфраструктуры виртуальных рабочих столов (Workplace), должен уверенно ориентироваться в вопросах, связанных с обслуживанием персональных компьютеров и тонких клиентов, в том числе – подключением средств ввода и вывода информации (клавиатура, манипулятор типа мышь, сенсорный коврик/экран, дисплей, принтер), мультимедийных устройств (звуковые колонки, проекторы и т.п.), средств локального хранения данных (карты памяти, USB-флеш), средств фильтрации и блоков бесперебойного питания.
Деятельность администратора регулируется отделом информационной безопасности и/или внутренними нормативными документами, связанными с обеспечением безопасности данных и конфиденциальности коммерческой информации.

2.2 Инструменты администратора

Деятельность облачного администратора или администратора VDI не ограничена использованием одного компьютера (АРМ). В зависимости от характера возникающих задач администратор может использовать различные виды СВТ: от персонального компьютера (ноутбука) с установленной операционной системой Linux до тонкого клиента, с помощью которого пользователь VDI осуществляет подключение к VDI машине.
На СВТ, используемом администратором, должен быть установлен веб-браузер, поддерживаемый операционной системой (Windows, Ubuntu, CentOS и др.). Кроме того, должно быть установлено ПО, позволяющее осуществлять безопасное подключение к управляющим/вычислительным узлам инфраструктуры, а также к вспомогательным виртуальным машинам, если таковые интегрированы в облако БАЗИС для определенных (сервисно-профилактических) нужд.
Веб-браузер позволяет использовать веб-интерфейс, предоставляемый модулем TIONIX.Dashboard. Рекомендуемые к использованию веб-браузеры:

- Google Chrome;

- Firefox.

2.3 Удаленный доступ к инфраструктуре

Удаленный доступ к инфраструктуре должен обеспечивать безопасные технологии приёма и передачи данных. Допускается использование программных средств удаленного доступа, не допускающих компроментации облачной инфраструктуры (сохранение логинов/паролей в незашифрованном виде и т.п.). Также могут быть выбраны спецсредства, прошедшие проверку ИБ.

Внимание

Рекомендуется выбирать сертифицированные программные средства, включенные в единый реестр ПО (производимого в РФ).

При настройке удаленного доступа к облаку следует использовать SSH или организовывать дополнительные сетевые каналы, использующие VPN. Для доступа к виртуальным машинам и виртуальным рабочим столам (VDI) также должны применяться безопасные каналы или терминальные протоколы, поддерживающие *сквозное шифрование*.
После того как администратор закончил работу в веб-браузере любого из СВТ, не закрепленного лично за ним, он обязан принять меры по удалению любых сохраненных учетных данных, связанных с доступом к средствам управления или аппаратным компонентам облака (имена учетных записей, пароли к ним и т.п.).

2.4 Безопасный доступ к информации

Необходимо соблюдать *меры предосторожности и правила ИБ*, установленные в рамках отдела и/или организации. Администратор должен быть бдительным при выполнении авторизации с чужого рабочего места (ТК), так как некоторые веб-браузеры сохраняют вводимые пароли через куки или другими способами.
Записные или электронные книги, равно как и данный документ, не должны находиться без присмотра в помещениях общего пользования.
Если АРМ администратора оборудован АПМДЗ, то электронные ключи должны храниться в сейфе или сдаваться под охрану, в с соответствии с действующими на предприятии должностными инструкциями по информационной безопасности.
В конце рабочей смены все персональные компьютеры и СВТ, закрепленные за администратором, должны быть переданы по смене, с соответствующей отметкой в журнале технической эксплуатации, или заблокированы и заперты в специальном помещении, в зависимости от принятых на предприятии организационных мероприятий и политик безопасности.

Важно

Не допускается случайная или основанная на личном доверии передача третьим лицам учетных данных, смарт-карт, электронных ключей и других средств, разрешающих полный или частичный доступ к информации об объектах инфраструктуры.

2.5 Рассылка данных статистического учета

Установка модуля TIONIX.PointMeter в составе облачной платформы является обязательной (см. документ Инструкция по развертыванию ПО Базис.vCore).
Для рассылки может потребоваться дополнительное выделение почтового ящика и изменение в настройке расписания рассылки.

Информация

Если в веб-браузере (Linux/Ubuntu) возникли ошибки или нарекания на функционал модуля PointMeter, такие как ограничение в функциональности, следует:

- очистить кэш веб-содержимого и проверить локальную файловую систему на целостность;
- удалить пакет веб-браузера (firefox), обновить репозиторий и установить пакет заново;
- рассмотреть использование иного веб-браузера.

2.6 Техническое обслуживание, ремонт, профилактика

Техническое обслуживание и ремонт средств вычислительной техники, коммутационного оборудования и систем хранения данных, а также источников бесперебойного питания осуществляются на основе паспортов и руководств по (сервисному) обслуживанию, соответствующих моделям устройств. Документы предоставляются предприятиями-изготовителями или вендорами.

Персонал, осуществляющий диагностику, техобслуживание или ремонт оборудования облачной платформы, должен пройти *инструктаж по технике безопасности* и обязан слаженно взаимодействовать с администратором, ответственным за эксплуатацию облачной инфраструктуры.

Администратор обязан вести *журнал эксплуатации* облачной инфраструктуры, оформлять все существенные события, начиная с момента завершения ПНР и фиксации приема-сдачи платформы в эксплуатацию.

Плановые *профилактические работы* должны быть тщательно спланированы. Эксплуатирующая организация должна составить «дорожную карту», содержащую комплекс и график проведения профилактических мероприятий, не противоречащих нормальной эксплуатации ПО.

Должна быть выполнена оценка рисков отказа оборудования или компонентов эксплуатации и выявлен список для возможной оперативной замены (ЗИП). Рекомендуется имитация и отработка вероятных ситуаций на стенде, отдельно от оборудования «продакшен». Это необходимо, чтобы *аварийные ситуации*, связанные с отключением служб или нод, не влияли на качество услуг, предоставляемых инфраструктурой VDI и/или бизнес-приложениями, помещенными в облако (виртуальный ЦОД).

3 Интерфейс управления

Интерфейс пользователя – один из основных инструментов управления инфраструктурой виртуального ЦОД, имеющийся в распоряжении администратора облачной платформы, построенной с применением ПО «Базис.Cloud».
Интерфейс управления (Dashboard) формируется в среде веб-браузера и обеспечивает графическое отображение инструментов управления объектами облачной инфраструктуры, такими как: проекты, виртуальные машины, диски, сети, образы, метаданные, пользователи и т.п.
После выполнения входа в интерфейс для администратора становятся доступными веб-инструменты, реализующие *графический интерфейс управления* облачной инфраструктурой. Доступность веб-инструментов зависит от прав, назначаемых на основе **ролевой модели**.

3.1 Вход в интерфейс управления

Для входа в облачное хранилище (авторизации) администратор системы должен выполнить следующие действия:

1. Ввести в адресной строке браузера ссылку: http://<доменное имя или ip>/dashboard

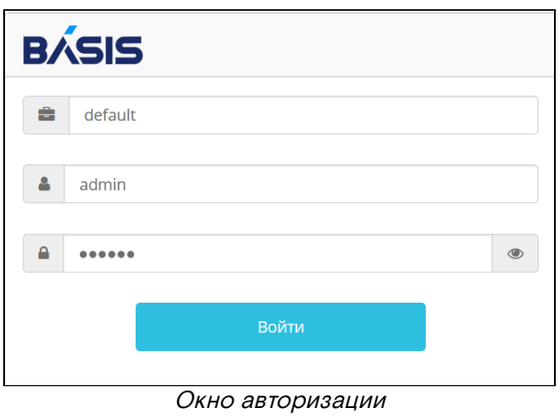
   где доменное имя или ip – выбирается в соответствии с используемой платформой.
   После нажатия <Enter> и разрешения доступа по IP (через доменное имя) откроется окно авторизации.

2. Заполнить учетные данные пользователя:

- Домен;
- Логин;
- Пароль.

По умолчанию используется домен с названием Default.

Нажать на кнопку [Войти].



Окно авторизации

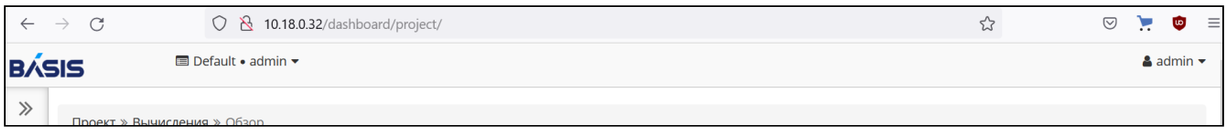
В случае неверно введенных учетных данных в верхней части окна (под заголовком) будет отображено сообщение (на розовом фоне): *Invalid credentials*. После успешной авторизации будет отображено окно интерфейса управления ПО.

> **Внимание**
>
> В случае, если предъявляются частные требования к использованию безопасного протокола HTTPS, необходимо предпринять ряд мер по обеспечению УУ (контроллеров) дополнительными пакетами ПО и настройками конфигурации ПО. Безопасный вход осуществляется также, как описано выше, но с явным указанием на использование протокола.

3.2 Веб-интерфейс

Веб-интерфейс, предоставляемый модулем TIONIX.Dashboard, содержит множество типовых способов представления информации (веб-окна, веб-поля, веб-таблицы) и органов управления (веб-кнопки, веб-меню, веб-списки и т.п.).

> **Внимание**
>
> Далее органы управления будут упоминаться без префикса: кнопки, списки и т.д.



Окно веб-интерфейса Базис.Cloud

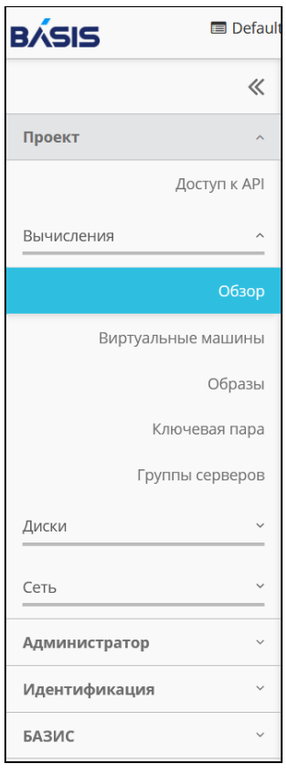
Главная страница веб-интерфейса отображает:

- полосу авторизации (вверху справа);
- панель навигации (веб-меню);
- рабочую область (пространство справа от панели навигации под полосой авторизации).

Для выбора действия, ассоциированного с органом управления, используется клик мышью, после того как курсор мыши наведен над ним.
Структура разделов меню зависит от прав, назначенных пользователю, который выполнил вход в систему.

3.2.1 Навигационная структура

Навигация в графическом интерфейсе управления происходит при помощи мыши. Достаточно навести курсор мыши на элемент меню, кнопку и кликнуть на нем. Отображение в рабочей области изменится исходя из выбранного контекста.
Интерфейс управления представлен в виде меню – навигационной панели – и обладает вложенностью структуры. Ниже поясняются *основы навигации* по этой структуре.



Панель навигации Базис.Cloud (веб-меню)

На показан верхний уровень структуры, содержащий элементы главного меню – *разделы*:
– Проект;
– Администратор;
– Идентификация;
– Базис.

Примечание

Для наглядности раскрыт один из элементов (Идентификация), содержащий подчиненные элементы.

Важно

Раздел меню "Администратор" доступен только в том случае, если вход выполнен от имени пользователя, наделенного полномочиями администратора.

Для указаний по навигации принято сокращенно указывать набор действий, ведущий к определенному элементу меню. Например, полный список меню "Идентификация" может быть представлен следующим образом:

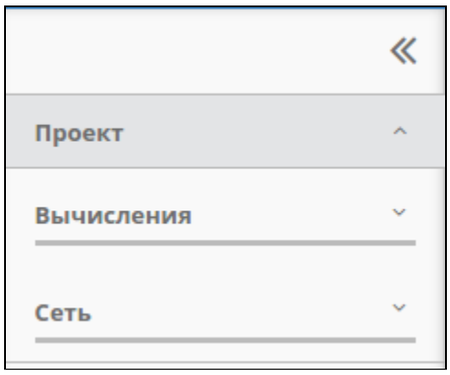
- Идентификация >> Проекты;
- Идентификация >> Пользователи;
- Идентификация >> Группы;
- Идентификация >> Роли.

Примечание

В некоторых документах может встречаться устаревший, иерархический способ представления навигации:
- Идентификация/Проекты
- Идентификация/Пользователи
- и т.д.

3.2.2 Вложенные меню (подразделы)

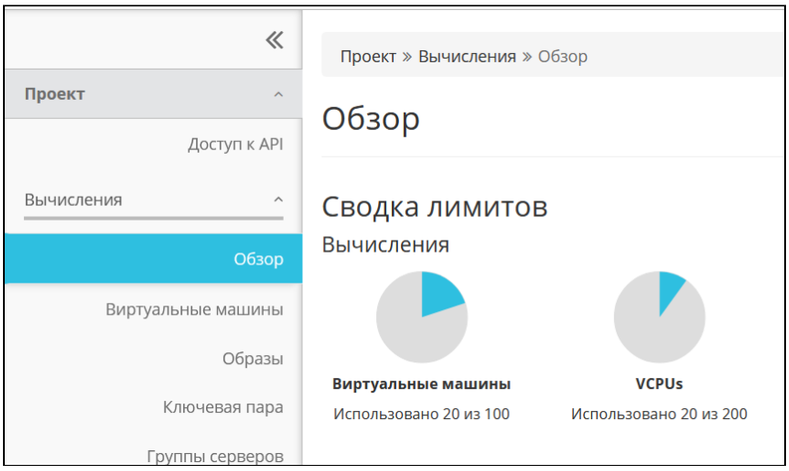
Если раздел (меню) панели навигации имеет более глубокий уровень вложенности, то под текстовым описанием такого веб-элемента отображается серая горизонтальная полоска (рисунок "*Отображение уровней вложенности меню*").
Сами элементы меню принято называть *подразделами* (Вычисления, Сеть).



Отображение уровней вложенности меню

3.2.3 Вкладки и контекст

Определенный контекст, подлежащий открытию – навигации, с указанием его названия в кавычках, будет называться *вкладкой*.



*Контекст интерфейса управления (вкладка **Обзор**)*

После раскрытия (кликом мыши) и выбора в левом верхнем углу рабочей области крупным шрифтом будет отображено текстовая строка, которая определяет текущий контекст. *Контекст* определяет доступность для пользователя тех или иных операций, в зависимости от полномочий.

> **Примечание**
>
> Полномочия определены на основе ролевой модели, заложенной в ПО Базис.Cloud.
> Последний открытый контекст веб-интерфейса запоминается как сеанс работы, который автоматически будет открыт после следующей процедуры авторизации.

3.2.4 Отображение механизмов управления и фильтрации

В рамках графического интерфейса повсюду используется унифицированное отображение фильтров по спискам перечисленных объектов, например – проекты.

Проекты (контекст раздела "Идентификация")

Использование фильтров позволяет сузить круг поиска и ограничить визуальное представление в заданном фильтром контексте.

Инструмент добавления объекта из общего списка доступных объектов

Для добавления в участники проекта пользователя, выбираемого из множества всех доступных пользователей, следует выполнить следующие действия:

1. Выбрать пользователя для добавления в участники проекта.

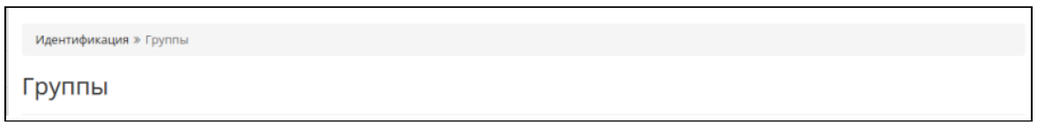
Колонка списка «Все пользователи».

2. Нажать на кнопку со значком «+».

Пользователь переместится в список колонки «Участники проекта».

3. Воспользоваться выпадающим меню с ролями пользователя и присвоить ему роль.

4. Нажать на кнопку «Сохранить».



Пример добавления в проект объекта (пользователя)

Примечание

Аналогичный инструмент используется для добавления в проект других объектов.

Механизм фильтрации списков

Для запуска механизма фильтрации списков объектов, выводимых в рабочей области, используется панель управления Фильтр. С её помощью можно выполнить следующие действия:

1. нажатие на кнопку Имя XXXX открывает выпадающее меню со списком параметров фильтрации.
2. выбор параметра.
3. заполнение поля значения параметра.

Фильтрация списков виртуальных машин и сетей может быть использована администратором для ограничения вывода информации в видимой части окна, отображающего рабочую область, в зависимости от контекста, выбранного с помощью навигационной панели: фильтрация списка имен виртуальных машин, фильтрация списка имен сетей и т.д.

Инструмент сортировки

Все параметры, по которым производится сортировка имеют значок раскрывающегося списка, справа от поля параметра. Для сортировки по параметру необходимо кликнуть на этом значке мышью.

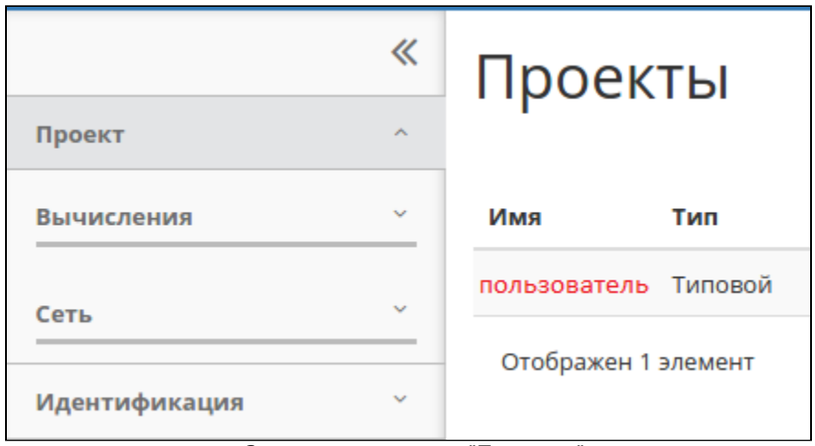
3.3 Структура и элементы меню

Если пользователь обладает правами администратора, то для него будут доступны все элементы меню (разделы, подразделы, вкладки), то есть панель навигации будет отображаться **без ограничений**.
В случае ограничения прав (обычный пользователь инфраструктуры) часть разделов, вкладок и действий будет недоступна.

> **Примечание**
>
> Более подробная информация о влиянии разграничений доступа на отображение элементов веб-интерфейса приводится в конце главы.

3.3.1 Проект

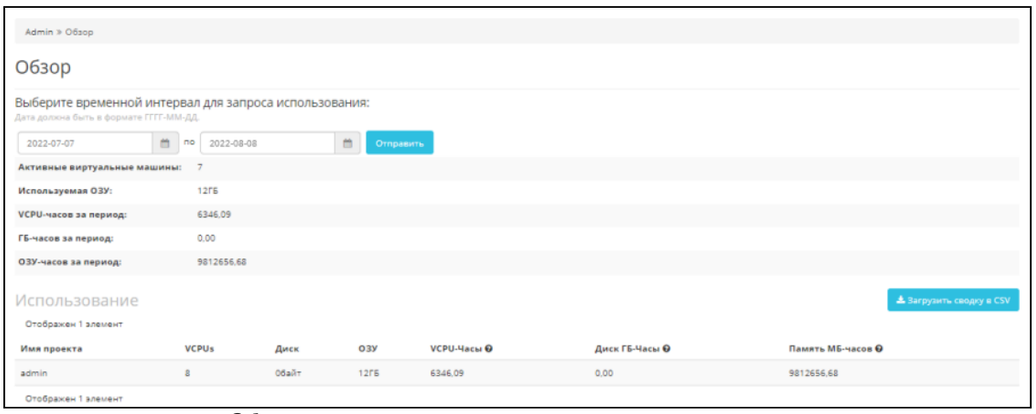
Ниже показана структура раздела "Проект".



Структура раздела "Проекты"

3.3.2 Администратор

Раздел интерфейса управления может быть открыт для обзора переходом по ссылке: http://<IP_контроллера>/dashboard/admin/
Общая сводка об использовании (физических) ресурсов проектами открывается переходом Администратор >> Обзор.



Обзор сводки использования ресурсов проектами

Для администратора также доступны подразделы, отображающие выделенные ресурсы облачной инфраструктуры, такие как:

- Вычисления (http://<IP_контроллера>/dashboard/admin/hypervisors/);
- Диски (http://<IP_контроллера>/dashboard/admin/volumes/);
- Сеть (http://<IP_контроллера>/dashboard/admin/networks/);
- Система (http://<IP_контроллера>/dashboard/admin/info/).

Примечание

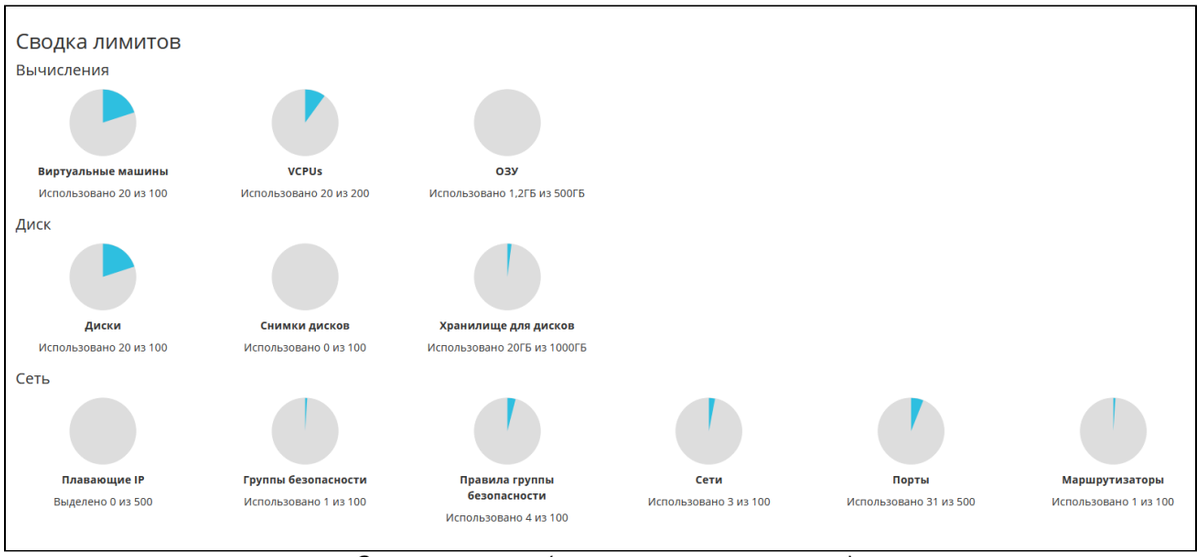
В скобках указаны прямые адресные ссылки (URL) на разделы

Вычисления – раскрывает такие ресурсы инфраструктуры (COMPUTE) как: гипервизоры (общая сводка, список), аггрегаты узлов, виртуальные машины (OpenStack instances), типы инстансов (OpenStack flavours), образы (OpenStack images).

Диски – поэлементно отображается информация о распределении дисков в пространстве хранения данных (STORAGE).

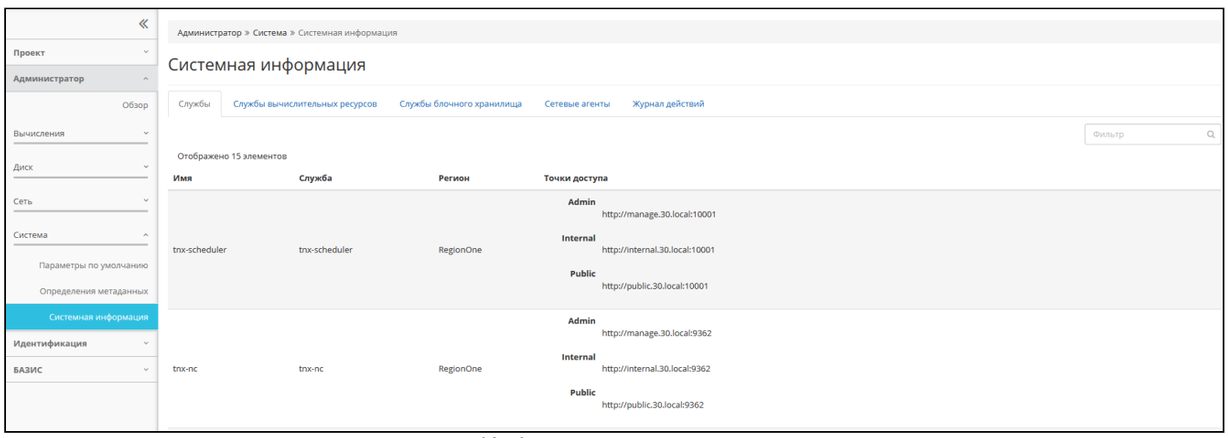
Сеть – поэлементно отображается информация о распределении и текущем состоянии сетевой инфраструктуры (NETWORK), маршрутизаторах и плавающих IP-адресах.

Система – отображаются элементы квот, характеризуемых как параметры по-умолчанию, определения метаданных и системная информация. При переходе Проект >> Вычисления >> Обзор отображается сводка лимитов и задействованных ресурсов облачной инфраструктуры.



Сводка лимитов (использование ресурсов)

Кроме сводки лимитов, внизу (в виде списка) выводится сводка об использовании выделенных ресурсов (проекта), в том числе – время наработки (в часах) виртуальных машин. Запрос использования может быть обновлен, с указанием определенного временного интервала (по умолчанию установлена выборка за сутки). После изменения интервала следует нажать кнопку \[Отправить\] и дождаться обработки запроса с (автоматическим) обновлением страницы. Системная информация содержит подробные сведения о службах OpenStack, в том числе – точках доступа (Admin/Internal/Public).



Информация о системе

Отображаются следующие списки элементов:

- состав служб, функционирующих на контроллере (УУ/control);
- состав служб вычислительных ресурсов (ВУ/compute);
- состав служб блочного хранилища;
- состав сетевых агентов;
- журнал действий.

3.3.3 Идентификация

Раздел содержит несколько вкладок: Домены, Проекты, Пользователи, Группы, Роли, Доступ для приложений.
Список проектов (http://<IP_контроллера>/dashboard/identity/) отображает полный список проектов, созданный в облаке.

Внимание

Данный список доступен, если был выполнен вход с *правами администратора* (пользователь с ролями в домене и проекте – admin:admin или cloud admin).Для других пользователей (с ограниченными правами) будет отображаться только одна строка, описывающая следующие атрибуты:

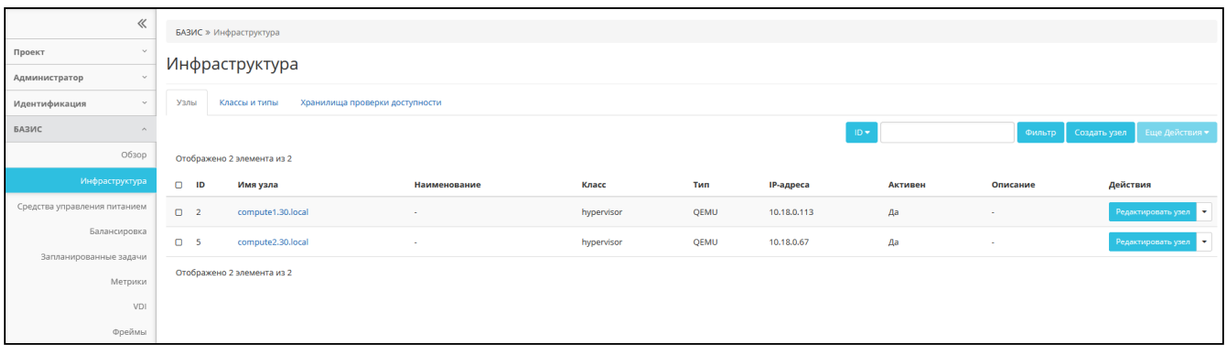
- Имя (проекта);
- Тип (проекта);
- Описание (смысловое назначение проекта);
- ID проекта;
- Имя домена;
- Активен (статус проекта);
- Действия.

Примечание

Выполнение действий над проектом доступно только аминистратору. Для пользователей с ограниченными правами действия не отображаются.

3.3.4 БАЗИС

Меню БАЗИС позволяет администратору получить подробную информацию о составе облачной инфраструктуры (Обзор). Кроме того, администратор может выполнить самодиагностику, а также донастройку инфраструктуры, полученной в результате ввода в эксплуатацию.
Список узлов инфраструктуры полезен при выполнении инвентаризации оборудования – серверных нод (УУ и ВУ), находящихся в настоящий момент в эксплуатации. Он также может быть использован при выполнении сервисно-профилактических работ.



Список узлов инфраструктуры

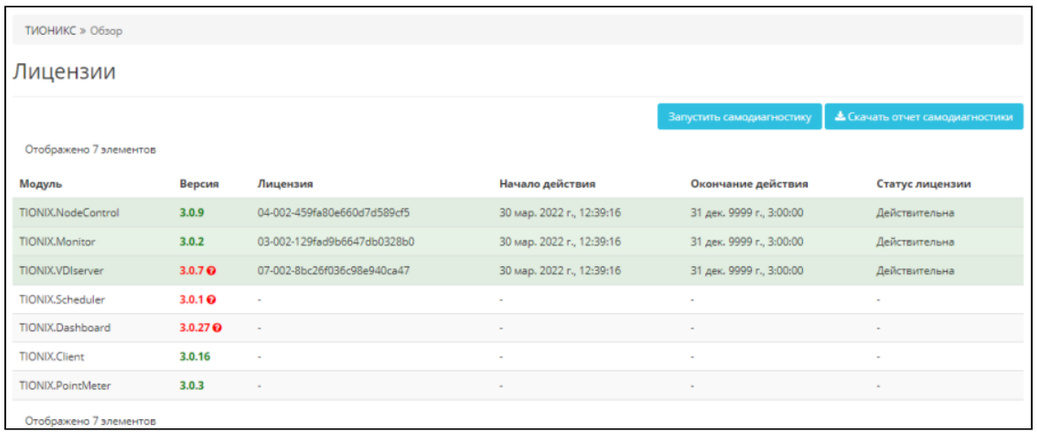
Примечание

Описание операций, связанных с управлением объектами инфраструктуры при помощи веб-интерфейса, подробно изложено в Руководстве администратора.
Ниже представлено краткое изложение возможностей, которые предоставляет веб-интерфейс для управления инфраструктурой.

Внимание

Полное раскрытие возможностей зависит от полноты настроек ПО Базис.Cloud, а также от ролей, назначенных на (авторизованного) пользователя.

Обзор



Обзор доступных лицензий на модули TIONIX

Примечание

Инструкция по обновлению (коммерческих) лицензий изложена в документе Руководство по интеграции ПО Базис.Cloud.
При отсутствии установленного модуля его версия не отображается; выводится соответствующее сообщение.

Средства управления питанием

Используется для настройки способов управления питанием вычислительных узлов инфраструктуры с целью централизации управления инфраструктурой. Может быть полезно как для инженеру по внедрению, так и для облачного администратора, для повышения эффективности труда, при эксплуатации большого количества физических нод.

Балансировка

Используется для просмотра резервов вычислительных узлов и использования вычислительных ресурсов (ОЗУ, vCPU, Диск) гипервизорами. Данные параметры контролируются *балансировщиком нагрузки*, обеспечивающим равномерное распределение вычислительной нагрузки (на ВУ и СХД).

Запланированные задачи

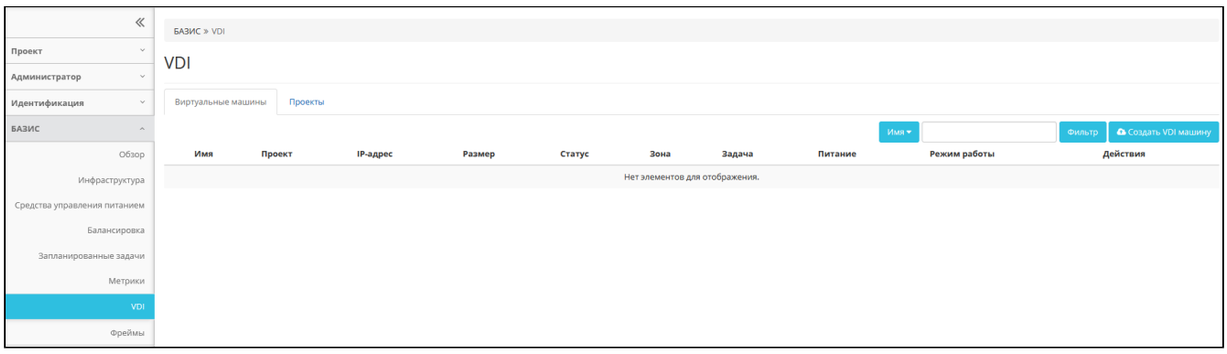
Используется для просмотра всех запланированных задач (действий над виртуальными машинами). Быстрый переход – по ссылке: http://<IP_облака>/dashboard/tionix/scheduler/

Метрики

Используется для просмотра статистики загрузки виртуальных машин. Метрики собирает служба OpenStack – Ceilometer. Быстрый переход – по ссылке: http://<IP_облака>/dashboard/tionix/metrics/

VDI

Основное назначение подраздела «VDI», расположенного в разделе БАЗИС – предоставление возможностей (функций) управления инфраструктурой виртуальных рабочих столов (далее – инфраструктурой VDI). Он не предназначен для непосредственного доступа к VDI машине, но позволяет выполнять операции по созданию проекта и пула виртуальных машин, используемого в рамках (выбранного) проекта.



БАЗИС >> VDI

Примечание

Возможности управления инфраструктурой VDI достаточно обширны; описание решения Базис.WorkPlace и практические способы выполнения операций вынесены в отдельный комплект эксплуатационных документов.

Фреймы

URL-ссылки, которыми наиболее часто пользуется администратор, могут быть сохранены здесь. Фреймы формируют вкладки с присвоенными именами, указанными в поле **Имя** (перед сохранением).
Например, удобно сохранить фреймы, позволяющие быстро переместиться в личный кабинет, используемый системой управления облачной платформой или на страницы документации.
Могут быть указаны как URL веб-страниц, так и ссылки на документы, открываемые с помощью веб-браузера (из которого выполнен вход).

3.4 Разграничение прав доступа (домен, проект)

Ниже представлена таблица, показывающая зависимость отображения элементов управления в графическом интерфейсе от *комбинации роли* в домене и в текущем проекте.

Примечание

Отображение элементов основано на объектно-ролевой модели, реализованной в БД OpenStack.
Комбинация строится из пары ролей, каждая из которых может принимать следующие значения:

- cloud admin (облачный администратор, ответственный за управление НОП);
- admin (администратор инфраструктуры);
- user (пользователь инфраструктуры).

Зависимость прав от ролей в домене и в проекте

Роль в домене	Роль в проекте	Права

user	user	В Разделе «Проект» недоступны некоторые действия: • для запланированных задач – «Повторить задачу»; • над QoS политиками – во вкладке «Сети» - «Сетевые сервисы QoS» (создание, редактирование, удаление), кроме подключения к объектам текущего проекта. Раздел «Администратор» – недоступен. В разделе «Идентификация» отображаются только проекты, в которых состоит текущий пользователь. Действия над проектами недоступны (кроме переключения); В разделе «БАЗИС» отображаются вкладки: • «Обзор»; • «Запланированные задачи»; • «Метрики»; • «VDI»; • «Фреймы». Особенности: • отображаются только задачи над виртуальными машинами и дисками проектов, в которых состоит пользователь; • для задач недоступно действие «Повторить задачу». Во вкладке «VDI» отображаются только доступные для пользователя VDI проекты и их машины. Действия над VDI проектами – недоступны. Доступны следующие действия над VDI-машинами: • создание; • редактирование; • удаление; • архивирование / разархивирование; • клонирование; • назначение пользователя (только себя) VDI машине; • перестройка, постановка на паузу / снятие с паузы; • выключение. Планирование действий над VDI-машинами – недоступно.

user	admin	В Разделе «Проект» - «Сети» - «Сетевые сервисы QoS» недоступны действия над QoS политиками (создание, редактирование, удаление), кроме подключения к объектам текущего проекта. Раздел «Администратор» доступен (доступны только объекты проектов, в которых состоит текущий пользователей). В разделе «Идентификация» отображаются только проекты, в которых состоит текущий пользователь. Действия над проектами недоступны (кроме переключения и планирования); В разделе «БАЗИС» отображаются вкладки: • «Обзор»; • «Запланированные задачи» (отображаются только задачи над виртуальными машинами и дисками проектов, в которых состоит пользователь); • «Метрики»; • «VDI»; • «Фреймы». Во вкладке «VDI» отображаются только доступные для пользователя VDI проекты и их машины. Доступно планирование действий над VDI проектами, в которых состоит пользователь. Для VDI машин доступно: • создание; • редактирование; • удаление; • архивирование / разархивирование; • клонирование; • назначение пользователя (только себя) VDI машине; • перестройка, постановка на паузу / снятие с паузы; • выключение. Планирование действий над VDI машинами – недоступно.

admin	user	Раздел «Администратор» – недоступен. В разделе «Идентификация» отображаются только домен, в котором состоит текущий пользователь. Во вкладке «Проекты» над проектами других доменов недоступно планирование запуска виртуальных машин. В разделе «БАЗИС» отображаются вкладки: • «Обзор»; • «Запланированные задачи»; • «Метрики»; • «VDI»; • «Фреймы». Во вкладке «VDI» отображаются только доступные для пользователя VDI проекты и их машины. Для VDI машин доступно: • создание; • редактирование; • удаление; • архивирование / разархивирование; • клонирование; • назначение пользователя (только себя) VDI машине; • перестройка, постановка на паузу / снятие с паузы; • выключение.
admin	admin	Раздел «Администратор» – доступен. В разделе «Идентификация» во вкладке «Проекты» над проектами других доменов недоступно планирование запуска виртуальных машин. В разделе «БАЗИС» отображаются все вкладки, кроме вкладки «SDS». Во вкладке «VDI» отображаются все VDI проекты и машины. Доступны все действия над проектами и VDI машинами.
cloud admin	user	Раздел «Администратор» недоступен. В разделе «Идентификация», во вкладке «Проекты», над проектами других доменов недоступно планирование запуска виртуальных машин. В разделе «БАЗИС» отображаются вкладки: • Обзор; • Запланированные задачи; • Метрики; • VDI; • Фреймы. Во вкладке «VDI» отображаются только доступные для пользователя VDI проекты и их машины. Для VDI машин доступно: • создание; • редактирование; • удаление; • архивирование / разархивирование; • клонирование; • назначение пользователя (только себя) VDI машине; • перестройка, постановка на паузу / снятие с паузы; • выключение.

cloud admin	admin	Раздел «Администратор» – доступен. Раздел «Идентификация» доступен в полном объеме. В разделе «БАЗИС» отображаются все вкладки. Во вкладке «VDI» отображаются все VDI проекты и машины. Доступны все действия над проектами и VDI машинами.

3.5 Скачивание отчета PointMeter

Перейдите на страницу обзора инфраструктуры – БАЗИС. Выберите интересующий период и нажмите кнопку [Скачать отчет по баллам]. Отчет по баллам представлен архивом, содержащим файл в формате электронной таблицы (MS Excel), т.н. «открытый отчет» (для пользователя Dashboard, администратора).

Внимание

Если попытаться скачать отчет, не загрузив предварительно публичный ключ, то система оповестит сообщением, что в нее следует загрузить публичный ключ. При нажатии на кнопку \ [Информация\] появляется модальное окно с информацией о лицензии, настройках почтового сервера, текущем расписании отправки и статусе последней отправки.

4 Управление ресурсами облачной инфраструктуры

ПО Базис.Cloud опирается на службы OpenStack (компоненты ПО), обеспечивающие управление выделенными для облачной инфраструктуры ресурсами при выполнении операций, связанных с виртуализацией объектов (диски, сети, машины).

Компонент OpenStack Neutron ([Network Service) обслуживает объектные абстракции сетей, подсетей и маршрутизаторов. Каждая абстракция имеет функциональность которая отражает (mimics) свою физическую сущность (physical counterpart): сети содержат подсети, а маршрутизаторы перенаправляют сетевой трафик между различными подсетями и сетями.

Компонент OpenStack Glance управляет статическими образами дисков, содержащими исполняемый код, а также операционную среду. С него начинается исполнение вычислительной рабочей нагрузки.

Компонент OpenStack Nova управляет исполняющимися экземплярами – инстансами (виртуальными машинами). В некоторых случаях может применяться OpenStack Watcher – гибкий и масштабируемый сервис оптимизации ресурсов, предназначенный для мульти-тенантных облачных инфраструктур.

4.1 Облачные сети

Первоочередная задача при конструировании инфраструктуры виртуального ЦОД, построенного по референсной архитектуре ПО Базис.Cloud – создание пользовательских сетей с определенной топологией. Подробная инструкция изложена ниже.

Настройку внешних сетей производит администратор облачной платформы. Изначально, будет доступна системная внешняя сеть – «VLAN External», с маршрутизатором router_<id вашего заказа>, определяющим точку подключения внутренней инфраструктуры к внешней сети.

Различают следующие типы внешней сети:

- Местный (local) – сеть, которая ограничена только одним вычислительным узлом (данный тип сети практически не используется).
- Flat – сеть, представляющая из себя один L2 широковещательный домен.
- Vlan – VLAN Gre и VXLAN сети – тип overlay-сетей, которые также используются для изоляции широковещательных доменов, но работают поверх L3 сетей.

Созданные и настроенные внешние сети будут доступны администратору проекта и будут отражаться в списке сетей.

4.1.1 Сетевая топология

Для построения собственной инфраструктуры нужно создать *пользовательскую сеть* и, при необходимости, подключить её к внешней сети – через *маршрутизатор*.

Создание сети

Переход: Сеть >> Сети.
Список параметров, заполняемых на этапе добавления сети:

- Имя сети – необязательное поле (при пустом значении имя генерируется автоматически);
- Разрешить Admin State - активация «Admin State»;
- Общая (флаг) – при выборе этого флага сеть становится общедоступной;
- Создать подсеть (флаг) – при выборе появляется возможность добавления подсети с заданными параметрами;
- Возможные зоны доступности – перечень зон доступности.

Создание подсети

Переход: Сеть >> Подсети.

Список параметров, заполняемых на этапе добавления подсети:

- Имя подсети – необязательное поле (при пустом значении имя генерируется автоматически);
- Сетевой адрес – адрес сети в формате CIDR (см. стр. 25);
- Версия IP – версия протокола IP. Доступные версии:
  - IPv4;
  - IPv6.
- IP шлюза – IP-адрес шлюза;

Запретить шлюз (флаг) – при выборе шлюз становится неактивным.

Детали подсети:

- Разрешить DHCP (флаг) – при выборе разрешается использование DHCP агентов.
- Выделение пулов – список выделенных IP-адресов пула;
- Сервера DNS – список IP-адресов DNS серверов;
- Маршруты узла – дополнительные маршруты для узлов.

Создание маршрутизатора

Переход: Сеть >> Маршрутизаторы.

1. Нажать на кнопку «Создать маршрутизатор».
2. Назначить произвольное имя, в качестве внешней сети выбрать: external_float_185.171.13.192/27
3. Нажать на активное имя маршрутизатора; произойдет автоматическая переадресация в интерфейс настройки.
4. Во вкладке «Интерфейсы» нажмите на кнопку [Добавить интерфейс].
5. Нажмите на кнопку [Создать маршрутизатор] для подтверждения операции. В списке маршрутизаторов появится строка с новым роутером.

4.1.2 Плавающий IP

Для привязки «Плавающего IP» к виртуальной машине необходимо обеспечить наличие в инфраструктуре следующих объектов (OpenStack):

- сеть;
- подсеть;
- маршрутизатор.

Привязка плавающего IP

Для назначения «Плавающий IP» виртуальной машине необходимо подключить маршрутизатор к созданной подсети и внешней сети.

1. Создайте виртуальную машину с интерфейсом в ранее созданной подсети (internal-net-1).
2. Во вкладке «Вычисления >> Виртуальные машины» из выпадающего меню выберите «Привязать плавающий IP».

В открывшемся окне «Управление назначением плавающих IP» выберите доступный IP-адрес и нажмите на кнопку [Назначить]. Если в выпадающем списке нет IP адресов, следует нажать на плюс и, в новом окне, выбрать Пул: external_floating_185.171.13.192/27

3. Нажмите на кнопку «Выделить IP».

После выделения IP-адреса, выберите из выпадающего меню «Порт для назначения» соответствующий порт и нажмите на кнопку «Назначить».

Управление плавающими IP

В списке виртуальных машин, в поле IP-адрес, отображается информация о *плавающем IP*.
Если необходимо переназначить Плавающий IP другой ВМ, то: сначала, во вкладке «Виртуальные машины», из выпадающего меню следует выбрать действие – «Отсоединить плавающий IP», затем – выполнить операцию привязки плавающего IP адреса (см. выше).

4.2 Облачные хранилища

Метод, на основе которого обеспечен доступ к хранилищу, реализован и предоставляется посредством драйвера блочного хранилища Cinder или несколькими драйверами, в случае конфигурации мульти-бэкэнда (multi-backend). Существует несколько вариантов реализации драйверов хранилищ, такие как NAS/SAN, NFS, iSCSI, Ceph и др.
Служба блочного хранилища – Cinder – компонент платформы OpenStack, который предоставляет доступ к блочным устройствам хранилища (по сети). Тома становятся доступными для гостевых ОС инстансов (виртуальных машин), независимо от того, какой ВУ/гипервизор предоставляет вычислительные ресурсы.

Примечание

Также Cinder позволяет управлять снимками дисков и типами дисков.

Более подробная информация по настройке бэкэнда хранилища с опорой на предусмотренные референсной архитектурой варианты отражена в документе Руководство по интеграции ПО Базис.Cloud.
Существует ещё один способ организации доступа к блочным устройствам хранилища – Ceph – распределенная отказоустойчивая система хранения, массивно масштабируемая и высокопроизводительная. Ceph является проектом с открытым исходным кодом, который предоставляет унифицированные программно-определяемые решения для хранения данных.
Архитектурные особенности распределенной отказоустойчивой системы хранения данных Ceph подробно рассмотрены в официальной документации.

Примечание

Краткий обзор архитектуры и практические инструкции по интеграции Ceph в инфраструктуру ПО BASIS изложены в эксплуатационных документах:
- Руководство архитектора;
- Руководство по интеграции.

4.2.1 Управление образами

Перед началом работы вам необходимо подготовить инфраструктуру для дальнейшего использования. Начните с образов, возможны следующие варианты:

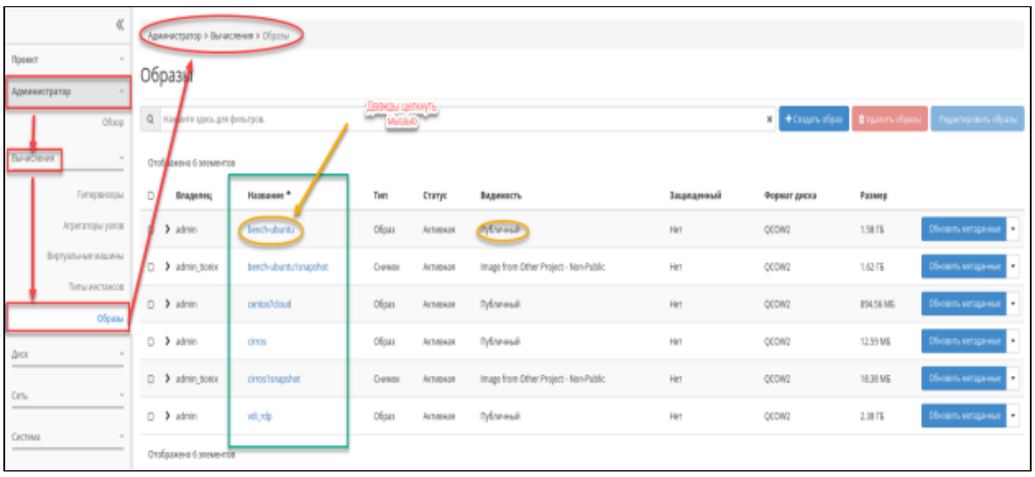
- использовать предустановленные в системе образы;
- загрузить свои образы, заготовленные на съемных индивидуальных носителях;
- скачать по ссылке из сети Интернет.

Использование предустановленных в системе образов

Для использования предустановленного образа (CirrOS) необходимо выполнить следующие действия:

1. Откройте в веб-браузере окно интерфейса управления (Dashboard).

С помощью навигационной панели перейдите: Администратор >> Вычисления >> Образы.



Переход Администратор >> Вычисления >> Образы

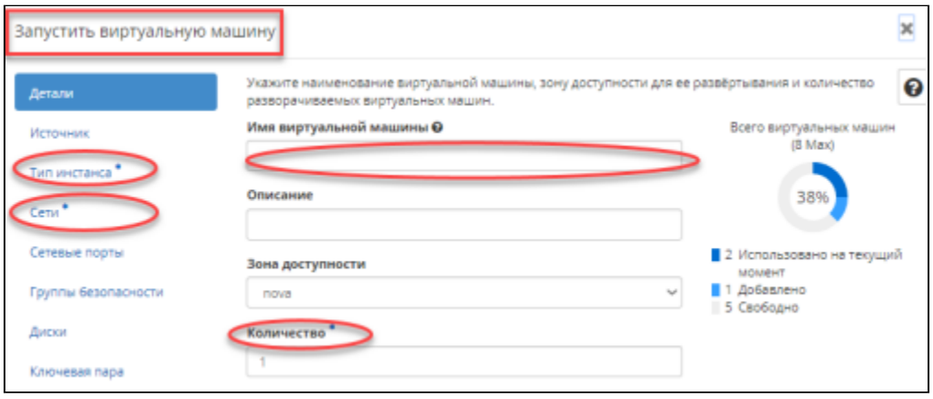
В открывшейся вкладке (Образы) отображается весь список предустановленных образов.

1. Выберите нужный образ в колонке списка *Название* и кликните имя образа.

Отобразится окно с детальной информацией (о предустановленном образе). В правом верхнем углу располагается кнопка [Запуск].

2. Для создания и запуска виртуальной машины с выбранным образом необходимо нажать на кнопку [Запуск].

3. В открывшемся окне Запустить виртуальную машину во внутренней вкладке Детали (по умолчанию), требуется заполнить имя ВМ и задать количество создаваемых машин.



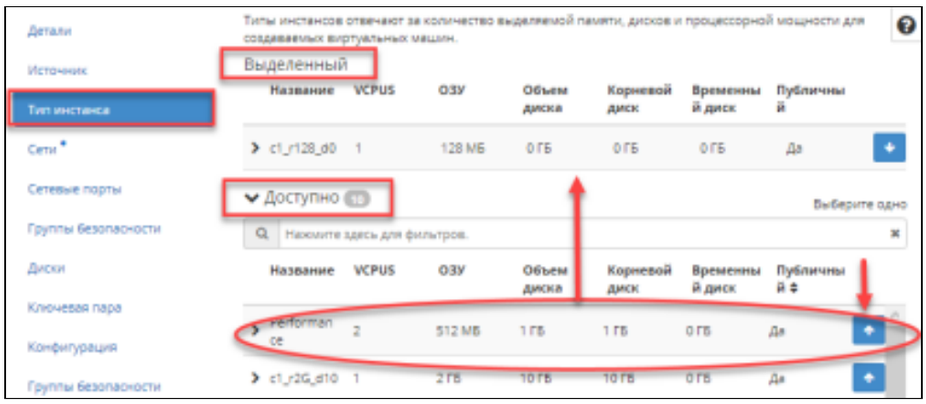
Окно создания виртуальной машины (ввод данных).

Примечание

Поля, обязательные к заполнению помечены знаком «звездочка» ⭐.

4. Перейти во вкладку «Тип инстанса» и выбрать нужный тип инстанса в разделе «Доступно».

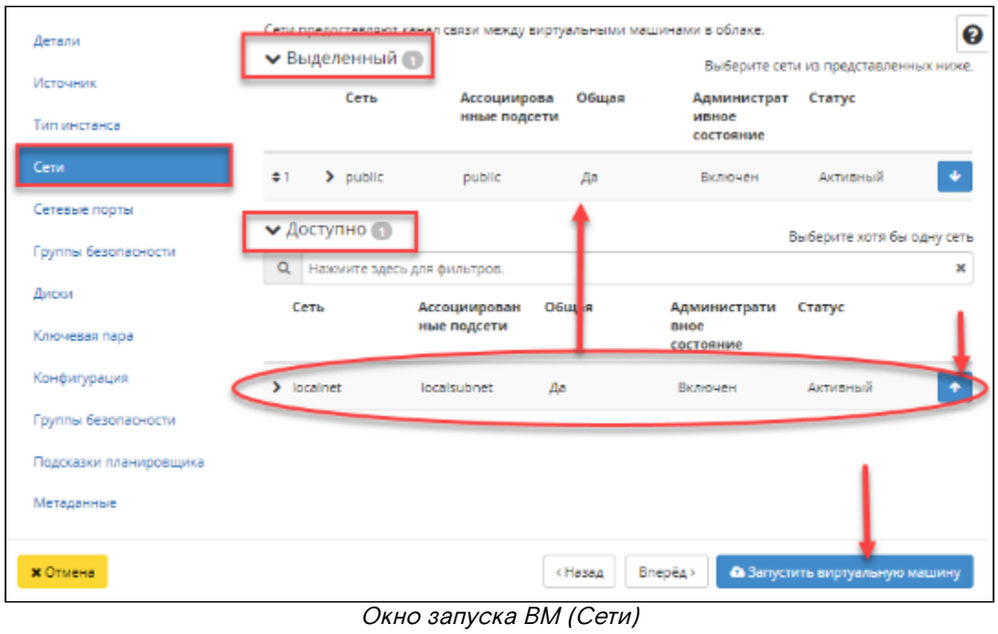
5. Нажмите на знак «стрелочки» в конце строки выбранного типа инстанса. Значение переместится в раздел «Выделенный».



Окно запуска ВМ (Тип инстанса)

6. Перейти во вкладку «Сети». Далее выбрать нужную сеть в разделе «Доступно».

7. Нажмите на знак «стрелочки» в конце строки выбранной сети. Значение переместится в раздел «Выделенный».



Окно запуска ВМ (Сети)

8. Нажать на кнопку [Запустить виртуальную машину].

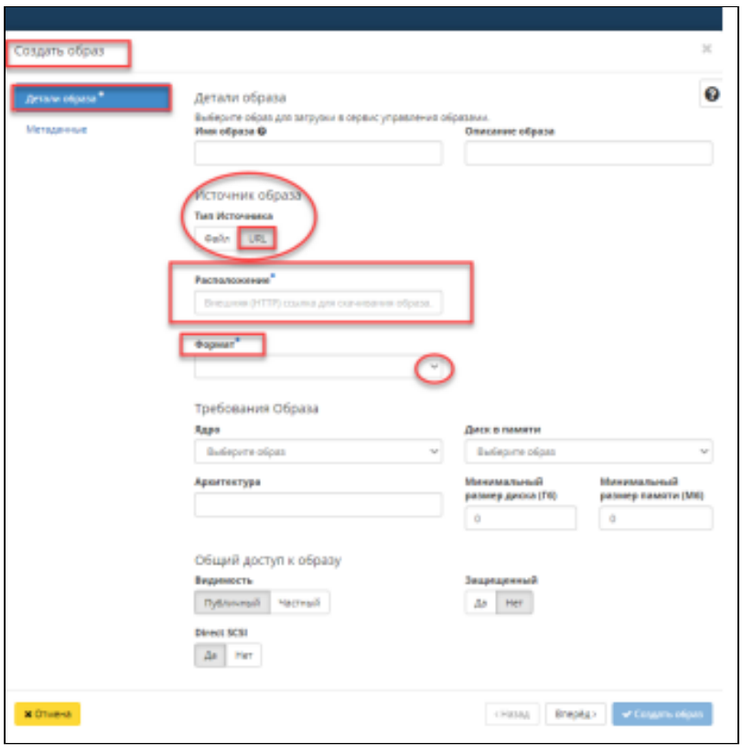
При успешном создании ВМ в правом углу экрана отобразится диагностическое сообщение «Экземпляр/ы запущен/ы».

Загрузка образа способом скачивания по ссылке из сети Интернет

Выполните следующие действия:

1. Перейдите: Администратор >> Вычисления >> Образы.
2. Нажмите на кнопку [Создать образ].

Откроется окно «Создать образ» на внутренней вкладке «Детали» образа.



Окно создания образа. Детали образа (URL)

3. Заполните параметры мастера окна создания образа:

Имя образа – необязательное поле, при пустом значении имя генерируется автоматически;
Описание образа – необязательный параметр;
Тип источника – выбор типа источника загрузки:
Файл;
URL.
Расположение – внешний адрес загрузки (HTTP);
Файл – внутренний адрес образа, который локально расположен в системе;
Формат (обязательный параметр) – выбор формата образа из перечня доступных.
Ядро – выбор ядра образа.
Использоваться могут только образы отдельных форматов, при отсутствии которых поле не отображается;
Диск в памяти – выбор диска из памяти;
Архитектура – архитектура образа;
Минимальный размер диска – требуется для загрузки образа (по умолчанию - 0 ГБ);
Минимальный размер памяти – требуется для загрузки образа (по умолчанию 0 МБ);
Видимость – видимость образа;
Доступные значения:

- Публичный;
- Частный.

Защищенный – защищенность образа;
Доступные значения:

- Да;
- Нет.

Direct SCSI – активация режима Direct SCSI.
Доступные значения:

- Да;
- Нет.

Метаданные – параметры метаданных образа.

4. В опции Источник образа – Тип источника – выберите опцию URL.

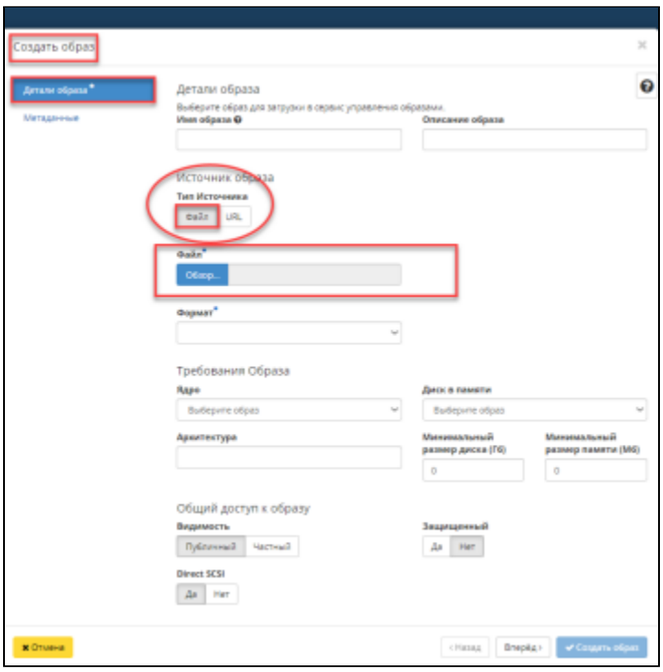
После заполнения всех обязательных параметров кнопка [Создать образ] станет активной. Нажать на кнопку [Создать образ]. В списке доступных образов появится вновь загруженный из Интернета образ.

Загрузка образа со стационарного/съемного индивидуального носителя

Алгоритм по загрузке образа со стационарного (съемного) индивидуального носителя аналогичен вышеописанной процедуре, за исключением пункта выбора источника образа – Тип источника.

1. Перейти: Администратор >> Вычисления >> Образы.
2. Нажать на кнопку Создать образ.

На внутренней вкладке Детали образа откроется окно Создать образ.

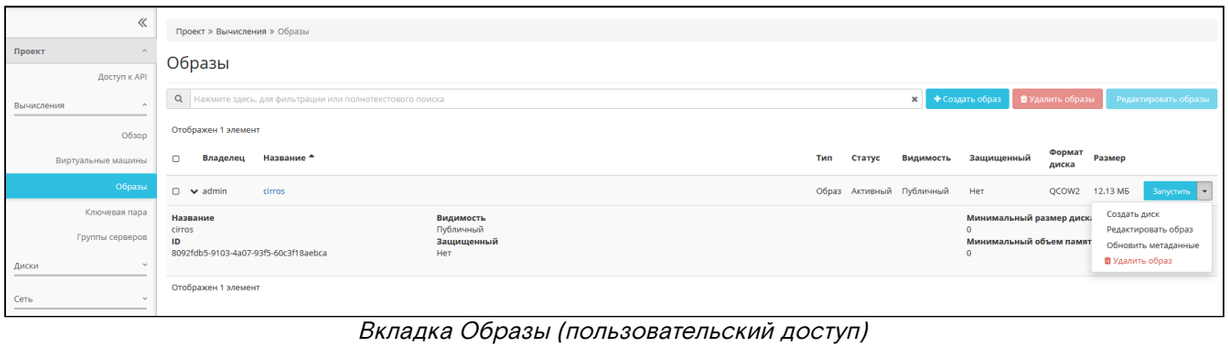


Окно создания образа. Детали образа (Файл)

3. В опции Источник образа – Тип источника необходимо выбрать опцию Файл.

После заполнения всех обязательных параметров кнопка [Создать образ] станет активной.

Нажать на кнопку [Создать образ]. В списке доступных образов появится вновь загруженный из сети Интернет образ. Действия с образами доступны для учетной записи с правами пользователя. Перейдите: Проект >> Вычисления >> Образы.



Вкладка Образы (пользовательский доступ)

Запуск виртуальной машины из загруженного образа

1. В строке выбранного образа нажать на кнопку [Запустить].
2. В открывшемся окне «Запустить виртуальную машину во внутренней вкладке «Детали» (по умолчанию).

Заполнить имя ВМ и задать количество создаваемых машин.

> **Примечание**
>
> Параметры обязательные к заполнению помечены знаком «звездочка» ⭐.

4. Перейти во вкладку «Тип инстанса» и выбрать нужный тип инстанса в разделе Доступно.

Нажать на знак «стрелочки» в конце строки выбранного типа инстанса. Значение переместится в раздел Выделенный.

5. Перейти во вкладку «Сети» и выбрать нужную сеть в разделе Доступно.

Нажать на знак стрелочки в конце строки выбранной сети. Значение переместится в раздел Выделенный.

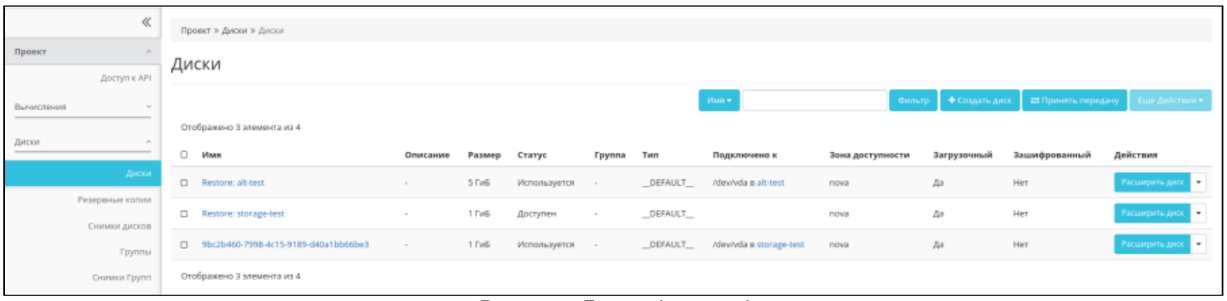
6. Нажать на кнопку [Запустить виртуальную машину].

При успешном создании в правом углу экрана отобразится диагностическое сообщение: *Экземпляр/ы запущен/ы*.

4.2.2 Управление дисками

> **Внимание**
>
> Выполнение действий, связанных с управлением дисками, доступны только для учетной записи, наделенной правами (ролью) администратора.

Перейдите на страницу: Проект >> Диски >> Диски



Вкладка Диски (проект)

Для администратора доступны следующие операции:

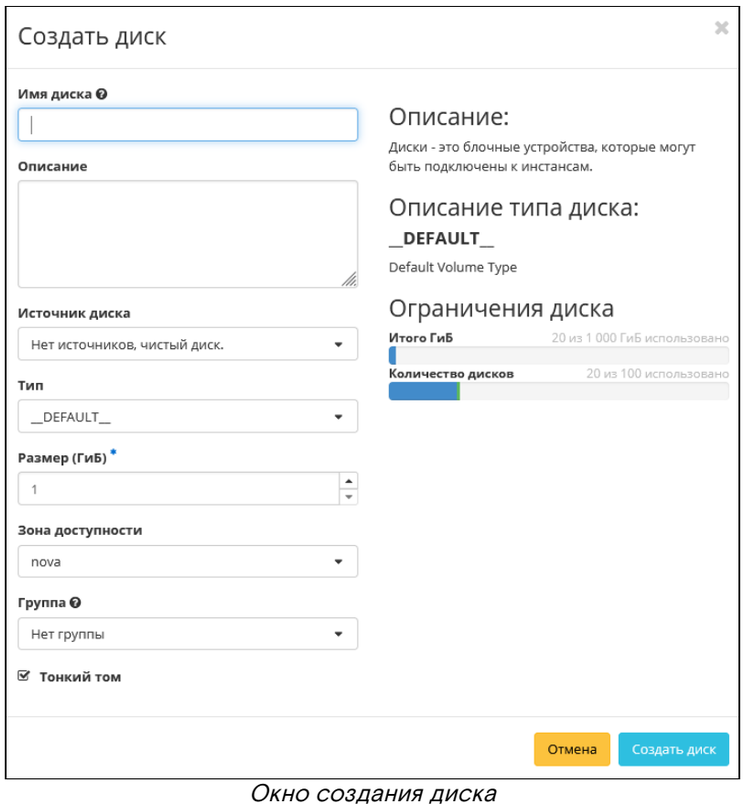
- создание диска;
- создание образа диска;
- загрузка диска на образ;
- расширение диска;
- изменение типа диска;
- удаление диска;
- миграция диска или логического тома;
- создание резервной копии диска и восстановление из неё;
- передача и приём передачи диска.

> **Внимание**
>
> Чтобы были доступны операции создания резервной копии диска и восстановления диска из резервной копии, необходимо, чтобы была включена подситема хранения резервных копий.

> **Примечание**
>
> Часть операций, таких как расширение, передача и прием передачи диска, подробно изложены в документе Руководстве администратора. Там же изложено описание выполнения планирования действий над диском.

Создание диска

В общем списке всех дисков на панели управления нажмите на кнопку [Создать диск]



Окно создания диска

В открывшемся (одноименном) окне, укажите параметры:

- Имя диска – необязательное поле (при пустом значении имя генерируется автоматически);
- Описание – необязательный параметр;
- Источник диска – выбор типа источника загрузки;
- Тип – выбор готового шаблона диска (редактирование типа описано во вкладке Типы дисков);
- Размер – объем памяти диска, в гигабайтах;
- Зона доступности – выбор осуществляется, исходя из потребности в тех или иных ресурсах;
- Тонкий том (флаг) – при выборе флага задействуется технология «Thin provisioning». Технология позволяет использовать свободное пространство диска для других нужд проекта.

Нажмите на кнопку [Создать диск] для подтверждения операции, после чего корректно созданный диск отобразится в общем списке.

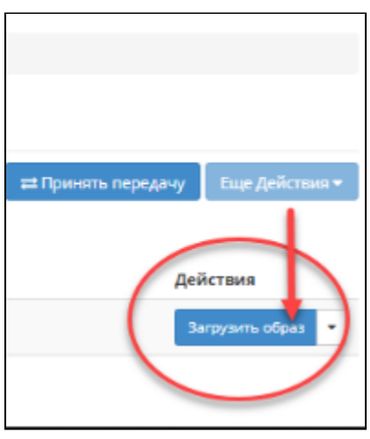
Примечание

По завершении создания диска, может понадобиться время на окончательную настройку всех параметров. В конечном итоге, диск отображается со статусом Доступен

Загрузка образа диска

Загрузка диска в службу образов осуществляется в виде файла образа. Образы дисков создаются с помощью утилиты QEMU disk image. Данная операция эквивалентна выполнению команды OpenStack: cinder upload-to-image{*}.

Нажмите на кнопку [Загрузить образ] для загрузки диска в службу образов.

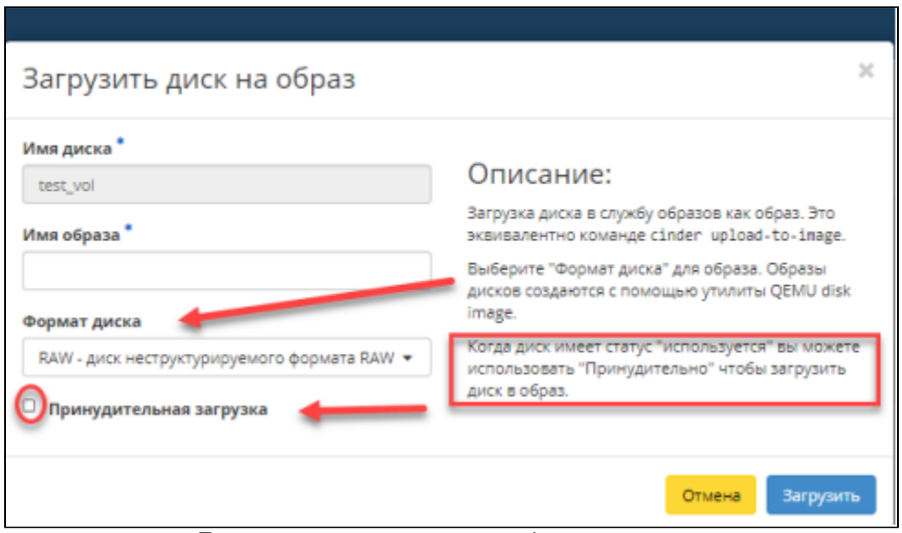


Загрузка образа (через службу образов)

Действие можно произвести как со страницы с общим списком дисков, так и во вкладках с детальной информацией о диске.

Загрузка диска на образ

Задайте имя и формат образа.



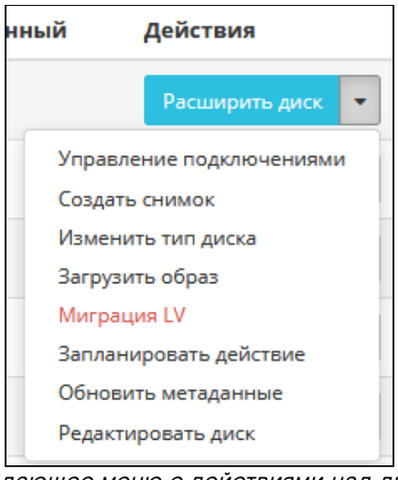
Перечень поддерживаемых форматов диска

Созданный образ отображается во вкладке «Образы» со статусом *Активный*.

Изменение типа диска

Данное действие позволяет редактировать тип и правила миграции для выбранного диска. Выполнить изменение типа можно со страницы с общим списком всех дисков.

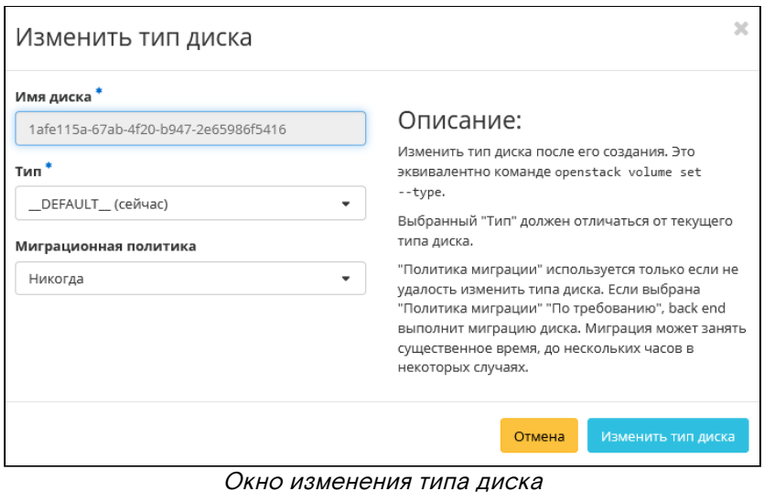
Нажмите на знак раскрывающегося списка кнопки [Загрузить образ] и выберите действие – Изменить тип диска.



Выпадающее меню с действиями над диском

В открывшемся окне задайте необходимые параметры:

- Тип;
- Миграционная политика:
  - Никогда;
  - По требованию.



Окно изменения типа диска

Подтвердите выполнение действия, для этого нажмите на кнопку [Изменить тип диска]. Параметр типа диска будет изменен в строке общего списка дисков.

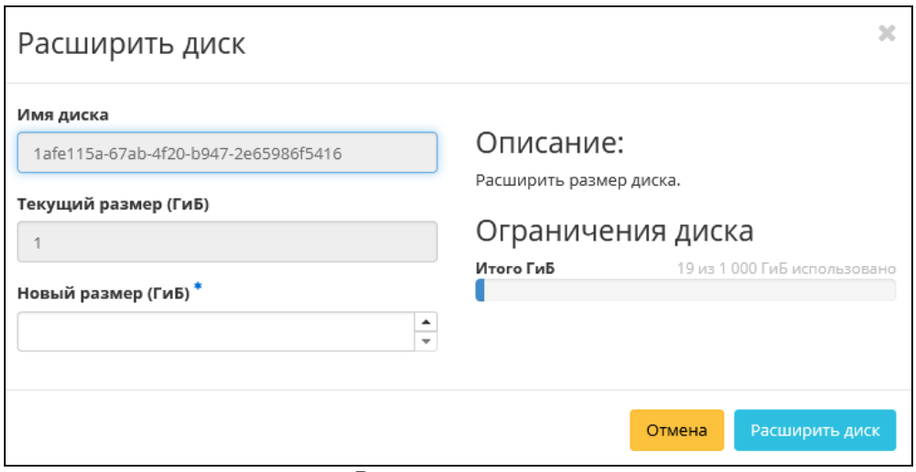
Расширение диска

Размер выбранного диска можно изменить в пределах выделенной квоты на проект. Изменять размер можно как у не подключенного к виртуальной машине диска со статусом *Доступен*, так и у подключенного со статусом *Используется*.

Примечание

Действие можно произвести как со страницы с общим списком дисков, так и во вкладках с детальной информацией о диске.

Активируйте действие «Расширить диск» из меню кнопки [Загрузить образ]. В открывшемся окне задайте необходимый размер диска.



Расширение диска

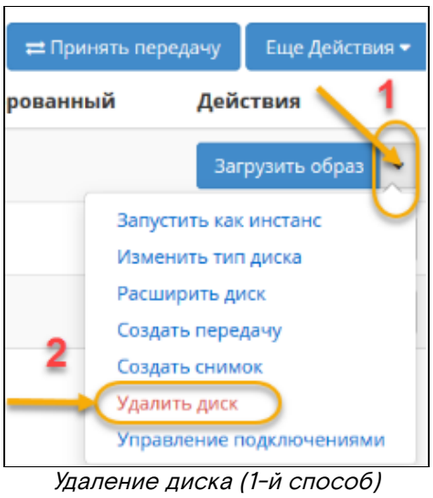
Подтвердите изменение размера диска – нажмите на кнопку [Расширить диск] (в окне изменения размера диска).

Удаление диска

Осуществить удаление диска можно двумя способами.

Способ 1

Нажмите на знак раскрывающегося списка кнопки [Загрузить образ] и выберите – Удалить диск. Отобразится запрос на подтверждение удаления диска.

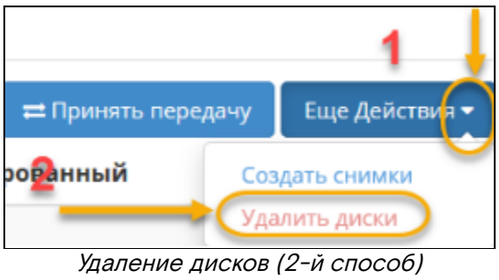


Удаление диска (1-й способ)

Нажмите на кнопку [Удалить]. Появится диагностическое сообщение об успешном удалении диска.

Способ 2

Предварительно, в списке дисков должен быть выбран диск (для удаления). Нажмите на знак раскрывающегося списка кнопки [Еще Действия] – выберите – Удалить диск. Отобразится запрос на подтверждение удаления диска.



Удаление дисков (2-й способ)

Подтвердите удаление, нажав на кнопку [Удалить].

Миграция диска

Миграция диска позволяет производить перенос объема данных и типа выбранного тома на свободный узел или в свободное хранилище. Это действие также может использоваться для эвакуации с проблемного узла или хранилища.

Внимание

Выполняется с правами администратора

Важно

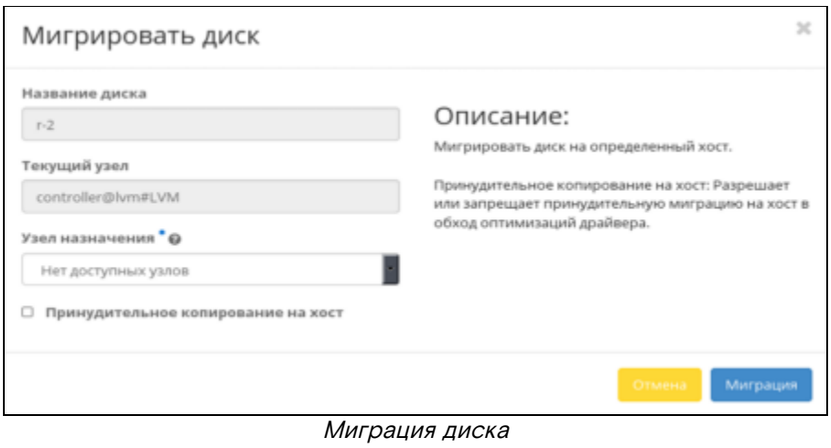
Для успешной миграции диска необходимо выполнение следующих требований:

- наличие прав доступа к диску;
- наличие свободных ресурсов памяти;
- поддержка типа диска;
- отсутствие подключенных виртуальных машин;
- отсутствие ранее созданных снимков диска.

Перейдите: Администратор >> Диски >> Диски.

Выберите – Мигрировать диск.

В открывшемся окне выберите нужный узел.



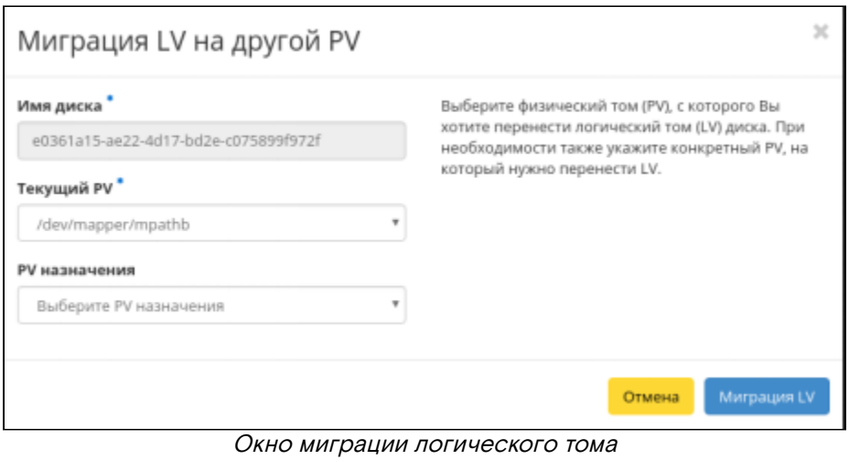
Миграция диска

При необходимости переноса диска без процесса оптимизации драйверов включите флаг Принудительное копирование на узел. Нажмите на кнопку [Миграция].

Миграция логического тома

Предоставляет возможность переноса данных логического тома в активной системе с текущего физического тома на выбранный.

Выберите необходимый диск (логический том). Из раскрывающегося списка кнопки [Запланировать действие] (Загрузить образ) выберите – Миграция LV.



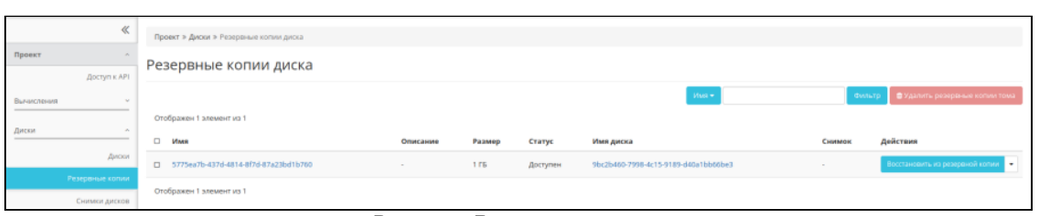
Окно миграции логического тома

Выберите физический том из списка доступных для переноса. Нажмите кнопку [Миграция LV] для подтверждения операции.

> **Важно**
>
> Указание физического тома для переноса необязательно. В этом случае перенос осуществится на автоматически выбранный.

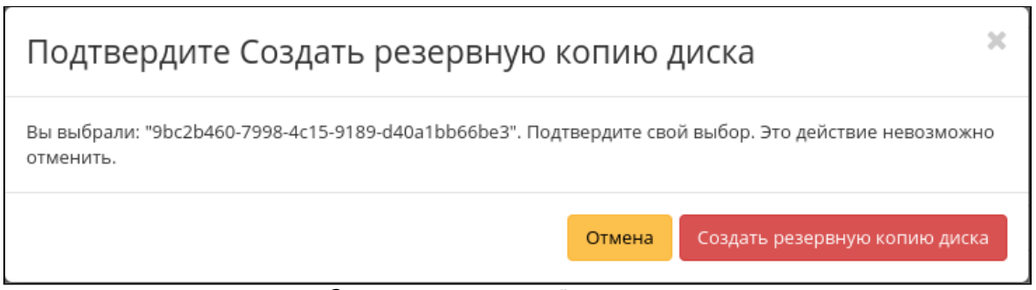
Создание резервной копии диска

Перейдите: Проект >> Диски >> Диски.



Вкладка Резервные копии

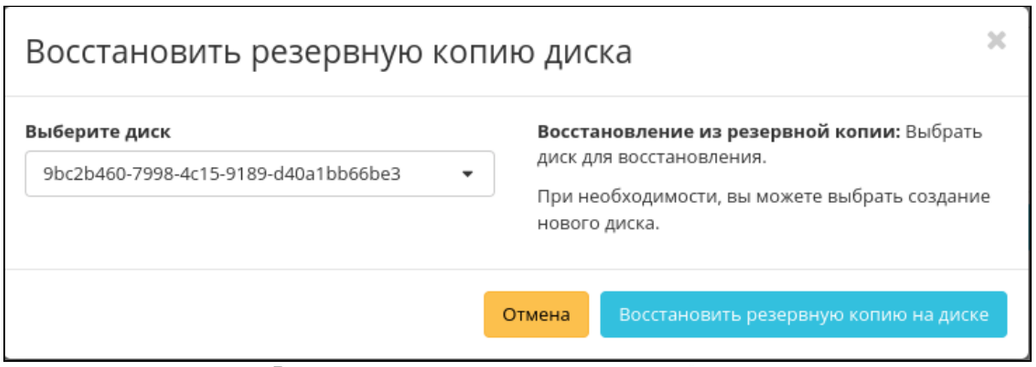
Из контекстного меню справа выберите действие – Создать резервную копию диска. Откроется диалог (proj_backup-disk), в котором достаточно подтвердить выполнение операции.



Создание резервной копии диска

Восстановление диска из резервной копии

Перейдите: Проект >> Диски >> Резервные копии.
Из контекстного меню справа выберите действие – Восстановить из резервной копии. Откроется диалог (proj_disk-restore), в котором достаточно подтвердить выполнение операции.



Восстановление диска из резервной копии

4.2.3 Подсистема хранения резервных копий

Для хранения резервных копий может быть настроена подсистема NFS Backstore. Backstore – термин, применяемый в области телекоммуникаций, который означает: хранение данных на «жестком диске», не требующее оперативного доступа.
NFS – сетевой протокол (сетевая файловая система), позволяющий обращаться к файлам и каталогам, расположенным на удалённых компьютерах (как если бы эти файлы и каталоги были локальными). С помощью NFS отдельно размещенные компьютеры могут разделять дисковое пространство – данные общего пользования хранятся на некотором узле сети и доступны для других компьютеров (клиентов), подключенных к этой сети.

Включение подсистемы

Важно

Убедитесь, что на УУ установлены пакеты **nfs-kernel-server** и **nfs-common**, обеспечивающие API и утилиты, необходимые для взаимодействия с сервером NFS. Оба эти пакета используются как сервером так и клиентом NFS.

Перейдите во вкладку резервного копирования графического интерфейса управления.

Отредактируйте файл local_settings, содержащий настройки Dashboard:

```
vi /etc/openstack-dashboard/local_settings
```

Файл должен содержать следующий параметр:

```
OPENSTACK_CINDER_FEATURES =
{                   „enable_backup": True, }
```

Перезапустите службу веб-сервера, для вступления изменений в силу и активации службы (Backup-подсистемы). Выполните команду:

```
systemctl restart httpd
```

Настройка NFS-сервера

Создайте папку в домашней директории:

```
mkdir /home/cinder-backup
```

Настройки экспорта NFS-сервера хранятся в файле /etc/exports. Откройте текстовый редактор и отредактируйте файл:

```
vi /etc/exports
```

Файл должен содержать строку:

```
/home/cinder-backup *(rw,sync,no_root_squash,no_all_squash)
```

Смысловое значение параметров подключения ФС в точку монтирования /home/cinder-backup:

- **rw** – доступ на чтение и запись (может принимать значение ro – только чтение);
- **sync** – синхронный режим доступа (может принимать обратное значение – **async**):
  - sync - указывает, что сервер должен отвечать на запросы только после записи на диск изменений, выполненных этими запросами;
  - async - указывает серверу не ждать записи информации на диск, что повышает производительность, но понижает надежность, т.к. в случае обрыва соединения или отказа оборудования возможна потеря данных.
- **no_root_squash** – запрет подмены uid/gid для суперпользователя (root);

  По умолчанию пользователь root на клиентской машине не будет иметь доступа к разделяемой директории сервера. Этой опцией снимается это ограничение;
- **all_squash / no_all_squash** – установка подмены идентификатора от всех пользователей.

Варианты использования:

- all_squash: подмена запросов от ВСЕХ пользователей (не только root) на анонимного uid/gid, либо на пользователя, заданного в параметре anonuid/anongid (используется обычно для публичного экспорта директорий);
- no_all_squash: запрет подмены uid/gid для от всех пользователей.

Запустите системную службу nfs-server:

```
systemctl start nfs-server
```

Включите автозагрузку службы (при старте ОС):

```
systemctl enable nfs-server
```

Настройка конфигурации Cinder

Отредактируйте секцию [DEFAULT] конфигурационный файл /etc/cinder/cinder.conf так, чтобы он содержал строки:

```
\[DEFAULT\] backup_driver = cinder.backup.drivers.nfs.NFSBackupDriver backup_file_size =
199983104 backup_container = None backup_enable_progress_timer = False
backup_mount_attempts = 3 backup_mount_options = 'vers=3' backup_sha_block_size_bytes =
32768 backup_share = <IP-адрес_хранилища>:/home/cinder-backup
```

Перезапустите все службы OpenStack, имеющие прямое отношение к Cinder:

```
systemctl restart openstack-cinder-*
```

Просмотрите файл журнала (логи службы, связанные с резервным копированием):

```
less /var/log/cinder/backup.log
```

4.3 Облачные вычисления

Nova управляет «фабрикой» облачных вычислений и, соответственно, является базовым компонентом инфраструктурных сервисов облачной платформы. Кроме того, Nova является самым сложным компонентом OpenStack, что объясняется преимущественно такими причинами, как высокая степень распределенности и большое количество процессов.

Glance отвечает за регистрацию экземпляров виртуальных машин – инстансов, ведение списков ВМ и извлечение образов по запросам службы Nova или облачного администратора.
При запуске экземпляра ВМ производится идентификация и специфицирование шаблонов виртуального оборудования (типов инстансов). Эти шаблоны описывают конфигурацию вычислительных ресурсов (виртуальные процессоры), ресурсов памяти и ресурсов хранения (жесткие диски), которые должны быть назначены экземплярам ВМ (инстансам).

Затем Nova планирует исполнение запрошенного экземпляра посредством приписывания его к определенному вычислительному узлу (в терминологии OpenStack — хост). Каждый узел вычислительного кластера регулярно представляет отчет процессу *tonova-scheduler* – о своем состоянии и возможностях. Процесс использует эти данные для оптимизации распределения ресурсов.

Примечание

Шаблоном выделения ресурсов для ВМ является flavor, в терминологии OpenStack.

Примечание

При установке платформы OpenStack по умолчанию предоставляется несколько flavor-шаблонов, конфигурация которых может быть изменена.

QoS (англ. Quality of Service - «качество обслуживания») – технология предоставления различным классам объектов различных приоритетов в обслуживании.

IOPS – один из ключевых параметров, используемый при измерении производительности систем хранения данных, жестких дисков (НЖМД), твердотельных дисков (SSD) и сетевых хранилищ данных (SAN).
QoS представлен в архитектуре OpenStack как сервисный плагин, отстраненный от остального кода службы Neutron и распределенный по нескольким уровням. QoS расширяет функциональность сключевых ресурсов (порты, сети) без использования «подмесов», унаследованных от плагинов, однако взаимодействует через драйвер расширения ML2 extension.

Миграция останавливает работу виртуальной машины и переносит ее на другой гипервизор (подробности – см. ниже). Процесс миграции виртуальной машины, обслуживаемой с помощью ОП BASIS, следует понимать как серию действий:

- изменение статуса машины с «Активна» на «Отключена»;
- перенос на определенный пользователем вычислительный узел;
- смена статуса машины на «Активна».

Общие особенности выполнения процесса миграции:

- виртуальная машина останавливается на время переноса;
- невозможность выбора вычислительного узла для переноса;
- выполнение миграции не подразумевает использование общего хранилища;
- миграция может происходить достаточно долго;
- Openstack сам выбирает вычислительный узел для миграции, исходя из свободных ресурсов;
- миграция работает с любым типом виртуальных машин.

4.3.1 Дисковые операции в платформе виртуализации

Дисковые операции выполняются в платформе виртуализации, подачей верхнеуровневых команд со стороны интерфейса управления.
Ниже представлены две наиболее общеупотребимых операции:

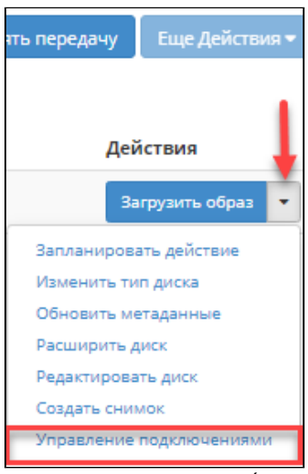
- запуск виртуальной машины из диска;
- подключение диска к виртуальной машине.

Запуск виртуальной машины из диска

Со страницы с общим списком дисков можно производить запуск виртуальной машины на основе выбранного диска.
Подробное описание процедуры создания виртуальной машины описано в официальной документации, на странице с описанием вкладки Виртуальные машины.

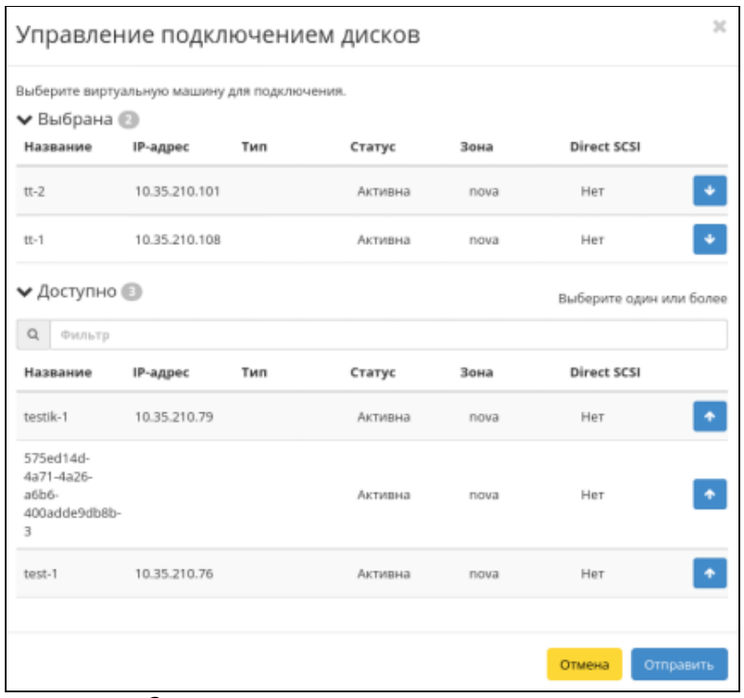
Подключение диска к виртуальной машине

Данное действие можно произвести со страницы с общим списком дисков. При наличии виртуальных машин в проекте можно осуществить подключение выбранного диска к виртуальной машине.



Управление подключениями (диска к ВМ)

Из выпадающего меню кнопки [Загрузить образ] (напротив выбранного диска) выберите – «Управление подключениями». В открывшемся окне, из раздела Доступно, выберите виртуальные машины, к которым нужно подключить выбранный диск. Нажмите на знак стрелочки в строках ВМ. Выбранные машины переместятся в раздел Выбрана.



Окно управления подключением дисков

Также предусмотрено обратное действие по отключению диска от ВМ, путем перевода из раздела Выбрана в раздел Доступно.

В разделе Доступно ВМ можно выбрать по одному из параметров:

- Название;
- IP-адрес;
- Тип;
- Статус;
- Зона;
- Direct SCSI.

При подключении диска к машине с Direct SCSI отправка SCSI команд диску будет производиться напрямую, в обход гипервизора. В зависимости от типа диска, существует возможность подключить диск к нескольким машинам, но только для определенных типов диска. В системе предусмотрена настройка типа диска с функцией множественного подключения (к нескольким виртуальным машинам). Для завершения подключения/ отключения ВМ от данного диска нажмите кнопку [Отправить].

4.3.2 Масштабирование

Операция по добавлению ВУ (Compute-ноды) в облачную инфраструктуру относится к *горизонтальному масштабированию* и выражается в увеличении количества ресурсов, доступных к использованию платформой виртуализации.

Начальные условия, необходимые для добавления дополнительного ВУ в облачную инфраструктуру:

1. Настроена **поддержка виртуализации** (служба гипервизора).
2. Настроен **репозиторий** (ОС, OpenStack, BASIS).
3. Настроена **сеть 10.x.x.x** .

Данные условия предъявляются в контексте конкретного аппаратного средства, вводимого в эксплуатацию в качестве вычислительного узла.
Под настроенной сетью подразумевается ряд фактов:

- установлен Open vSwitch;
- вычислительный узел доступен по инфраструктурной сети;
- создан мост.

Подробные инструкции по установке ОС и настройке вычислительного узла (ВУ) изложены ниже.

Удалите (упакуйте с помощью **xz**) все .repo-файлы, расположенные в директории /etc/yum.repos.d, кроме файла с настройкой доступа к репозиториям BASIS, если у инфраструктурных узлов нет выхода в сеть Интернет.

Примечания

Дополнительная информация о подготовке нового узла – настройке репозиториев различными способами, настройке сети – изложена в документе Инструкция по развертыванию ПО Базис.Cloud (Подготовка вычислительных узлов, Настройка репозиториев, Приложение).
В случае использования локального репозитория необходимо указывать локальные IP-адреса.
Дополнительный материал может быть найден в Интернет, на сайте официальной документации OpenStack для платформы **Victoria** .

Внимание

Чтобы упростить задачу настройки репозиториев, скопируйте файл tionix.repo с любого работающего ВУ, из директории /etc/yum.repos.d, на добавляемый узел, в директорию с тем же путём (названием).
Используйте безопасное копирование – **scp**, во избежание компроментации узла.

После того как выполнены основные условия, следует подготовить ВУ к использованию СХД, а также настроить дополнительные сетевые сервисы. Для вступления внесённых изменений в силу потребуется выполнить перезапуск служб.
Для проверки присутствия гипервизора в инфраструктуре авторизуйтесь в интерфейсе управления (TIONIX.Dashboard) с правами администратора (admin) и перейдите: Администратор >> Вычисления >> Гипервизоры

Добавленный ВУ с уникальным именем (compute_N) должен присутствовать в списке. Перейдите к просмотру деталей гипервизора (Обзор), при необходимости.

Подготовка сетевого имени

Вычислительному узлу необходимо присвоить уникальное (сетевое) имя, выполните две команды:

```
hostname hostnamectl
```

```
set-hostname compute_N
```

Первая команда выводит текущее имя хоста (хранится в файле /etc/hostname), а вторая корректно назначает новое имя (compute_N).
Чтобы применить изменение имени хоста, потребуется перезагрузка системы. Выполните одну из приведенных ниже команд:

init 6 или # systemctl reboot или # shutdown -r

Внимание

После выполнения процедуры смены имени узла рекомендуется убедиться, что изменения вступили в силу (выполните команду **hostname** после перезагрузки).

Подготовка узла

Необходимо проверить, что поддержка аппаратной виртуализации включена (модуль ядра kvm_intel активен). Выполните команду:

```
lsmod | grep kvm
```

Примечание

Поддержка аппаратного расширения виртуализации (Intel VT), как правило, включается из BIOS на самом начальном этапе конфигурирования ВУ.

После включения ВУ и загрузки операционной системы переведите SELinux в режим «Permissive» [6| id30]. Для этого выполните команду:

```
setenforce 0
```

Укажите режим работы – отредактируйте конфигурационный файл /etc/selinux/config:

```
SELINUX=permissive
```

Выполнить редактирование можно одной командой (с помощью строкового редактора):

```
sed -i s/^SELINUX=.*$/SELINUX=permissive/ /etc/selinux/config
```

Для вступления изменений в силу перезапустите компьютер:

```
reboot
```

После перезагрузки убедитесь, что система защиты SELinux функционирует в Permissive-режиме:

```
getenforce
```

Должно быть выведено сообщение *Permissive{*}. Отключите службу сетевого фильтра firewalld:

```
systemctl stop firewalld systemctl disable firewalld
```

Установка OvS

Из командной оболочки (bash) выполните команду установки программного пакета:

```
dnf install openvswitch
```

Продуктивные сети должны подаваться на интерфейс таким же образом, как это настроено в облаке. Например, если сеть 192.168.x.x подана в облаке с помощью VLAN, то на новом хосте настройку необходимо выполнить аналогично предыдущим настройкам ВУ. Файл /etc/sysconfig/network-scripts прописывается по аналогии с другими хостами (ВУ). Интерфейс продуктивных сетей должен быть добавлен в мост OvS. После внесения настроек перезапустите сетевые службы:

```
systemctl network restart
```

Чтобы новый ресурс резолвился, необходимо дополнить файл статического разрешения имен (/etc/hosts) информацией о новом узле (IP-адрес и алиас), на всех хостах (control/compute) облака. Скопируйте этот файл из работающего ВУ в развертываемый (целевой) ВУ; используйте утилиту безопасного копирования *scp* или доступную программу клиента SCP.

Важно

Все узлы в облаке должны резолвиться по DNS (динамически) или с помощью алиасов, прописанных в /etc/hosts (статически).

Установка агентов

Установите на подготавливаемую (Compute-) ноду службы Nova и Neutron, содержащие агенты Nova Compute Agent и Neutron Open vSwitch Agent. Выполните команду:

```
dnf install -y openstack-nova-compute openstack-neutron-openvswitch
```

Примечание

Кроме указанных пакетом, необходимые зависимости будут также установлены (из репозитория).

После установки пакетов необходимо активировать работу агента [TIONIX.Agent]. Для этого потребуется указать на него в конфигурационном файле Nova (/etc/nova/nova.conf), так, чтобы агент TIONIX использовался в качестве Compute-драйвера. Кроме того, требуется точное указание типа используемой системы виртуализации (прописать KVM в секции *\[libvirt\]{*}). Пример готовой настройки /etc/nova/nova.conf:

```
\[DEFAULT\] compute_driver = tionix_agent.virt.driver.TnxLibvirtDriver monkey_patch =
True monkey_patch_modules = nova.virt.driver:tionix_agent.virt.monkey_patch \[libvirt\]
virt_type = kvm
```

Настройка конфигурации сетевых служб

Скопируйте конфигурационные файлы из ранее настроенного ВУ:

- /etc/nova/nova.conf;
- /etc/nova/nova-compute.conf;
- /etc/neutron/neutron.conf;
- /etc/neutron/plugins/ml2/openvswitch_agent.ini.

Исправьте параметр в конфигурации Nova (/etc/nova/nova.conf), указывающий на IP-адрес:

```
\[DEFAULT\] # IP-хоста инфраструктурной сети my_ip = ...
```

Исправьте секцию ovs в конфигурационном файле openvswitch_agent.ini:

```
\[ovs\] local_ip = 10.x.x.x bridge_mappings = external:br-vlan
```

Параметр **local_ip** используется для туннеллирования VXLAN приватных сетей (обычно - инфраструктурной сети).

Параметр **bridge-mappings** сопоставляет физические сети и сети Neutron для работы сетей VLAN и Flat.

В данном примере:

- external -- название физической сети в настройках свойств виртуальной сети Neutron;
- br-vlan – название OvS-моста в операционной системе.

Название (физической) сети указывается при создании сети из Dashboard. Если в конкретном случае OvS-мост не собирается с помощью изменений в конфигурационных файлах, то можно собрать мост вручную. Создайте мостовой провайдер OvS (br-provider):

```
ovs-vsctl add-br br-provider
```

Выполните команду аггрегации двух интерфейсов в мост:

```
ovs-vsctl add-bond br-provider bond0 eno1 eno2 lacp=active
```

где _eno1_ и _eno2_ – название физических интерфейсов (определяемых в ОС).

Установка TIONIX.Agent

Установите модуль TIONIX.Agent:

```
dnf install -y python-consul python-tionix_agent
```

Настройки агента производятся в файле конфигурации модуля – /etc/tionix/agent.conf.

Настройка службы гипервизора

Включите режим «прослушивания» для службы гипервизора. Отредактируйте файл /etc/sysconfig/libvirtd так, чтобы он содержал строку:

```
LIBVIRTD_ARGS="--listen"
```

Отредактируйте файл /etc/libvirt/libvirtd.conf так, чтобы он содержал строки:

```
listen_tls = 0 listen_tcp = 1 auth_tcp = "none"
```

Альтернативно, скопируйте файл конфигурации службы гипервизора с любого работающего ВУ на вводимый в эксплуатацию (вычислительный) узел, используя команду *scp{*}:

```
scp /etc/libvrt/libvirtd.conf tionix@IP-адрес:/etc/libvirt/
```

Пропишите новые SSH-ключи для Nova. Это необходимо для живой миграции и операции изменения размеров инстанса (flavour resize). Предоставьте пользователю Nova возможность входа в систему, выполните команды:

```
usermod -s /bin/bash nova # su nova
```

Используя утилиту безопасного копирования, рекурсивно скопируйте директорию /var/lib/nova/.ssh/ с любого из рабочих ВУ в директорию /var/lib/nova/, расположенную на целевом ВУ:

```
ssh -C tionix@<исходный_ВУ> scp -rp /var/lib/nova/.ssh <цель>
```

где:
<цель> – полный путь к узлу и директории, в которую копируется исходное содержимое директории с ключами (IP-адрес:/var/lib/nova/).

После копирования назначьте права (nova:nova) на всё содержимое скопированной директории:

```
chown nova:nova -R /var/lib/nova/.ssh
```

Проверьте доступ к другому ВУ (без запрашивания пароля):

```
su - nova ssh <Compute-нода> exit
```

Подготовка СХД

Если используются эфемерные диски, то необходимо с помощью утилиты **gdisk** разметить LUN с СХД как один раздел.

Примечание

Если используются multipath, то должна быть настроена соответствующая системная служба (multipathd).

Создайте файловую систему (ФС) на разделе, выполнив команду:

```
mkfs.xfs /dev/mapper/mpathdb-disk1
```

Примонтируйте хранилище в директорию /var/lib/nova/instances и добавьте следующую запись в файл /etc/fstab:

```
/dev/mapper/mpathdb-disk1 /mpathb xfs defaults 1 1
```

Внимание

Необходимо быть очень внимательным при внесении новой или изменении старой информации о способах монтирования файловых систем. Неверные изменения в файле /etc/fstab могут заблокировать нормальную загрузку ОС.

Пропишите права с помощью команды:

```
chown nova:nova -R /var/lib/nova/instances
```

Настройка дополнительных сетевых сервисов

В случае изоляции сетей между УУ (контроллером) и ВУ необходимо доустановить на Compute-ноду агенты: DHCP; metadata; L3. Выполните команду:

```
dnf install -y openstack-neutron-common openstack-neutron
```

Скопируйте конфигурационные файлы (из любого доступного рабочего ВУ):

- /etc/neutron/dhcp_agent.ini;
- /etc/neutron/metadata_agent.ini;
- /etc/neutron/l3_agent.ini.

Запустите сетевые службы Neutron:

```
systemctl restart neutron-dhcp-agent neutron-metadata-agent neutron-l3-agent
```

Добавьте сетевые службы в автозагрузку:

```
systemctl enable neutron-dhcp-agent neutron-metadata-agent neutron-l3-agent
```

Перезапуск служб

По завершении всех вышеописанных настроек на ВУ необходимо выполнить перезапуск служб OpenStack и ПО Базис.Cloud.
Выполните команды:

```
systemctl restart libvirtd openstack-nova-compute neutron-openvswitch-agent tionix-agent
```

Включите автозагрузку служб:

```
systemctl enable libvirtd openstack-nova-compute neutron-openvswitch-agent tionix-agent
```

Примечание

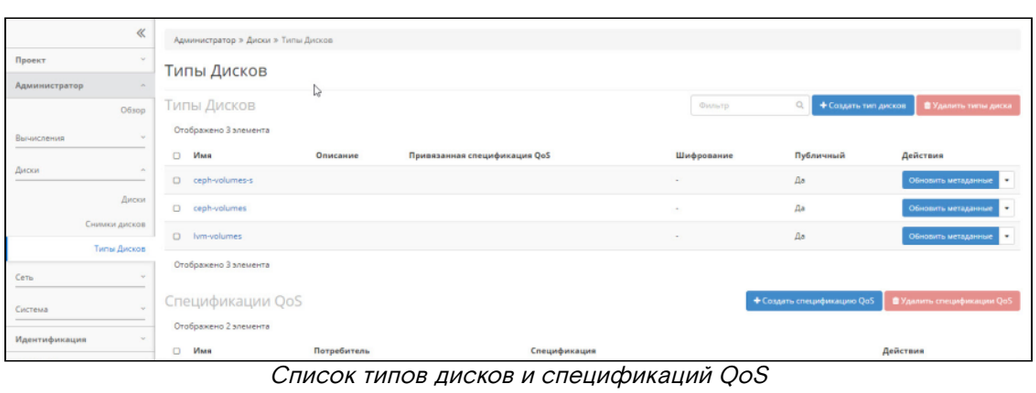
Перечисленные службы обеспечивают работу взаимосвязанных подсистем облачной платформы: гипервизора, Nova, агента OvS и агента TIONIX.

4.3.3 Настройка спецификации QoS

Для управления дисками ВМ и хранилищем данных (СХД) настраивается спецификация QoS. Она используется для ограничения IOPS, которое вводится для того, чтобы одна ВМ не использовала всю производительность СХД.
Параметр IOPS характеризует количество операций ввода/вывода. Параметр равен скорости, деленной на размер блока при выполнении операции. По сути – это количество блоков, которое успевает считаться или записаться на носитель. Чем больше размер блока, тем меньше кусков, из которых состоит файл, и тем меньше будет IOPS, так как на чтение куска большего размера будет затрачиваться больше времени. Значит, для определения IOPS надо знать скорость и размер блока, задействованного на выполнение операции чтения/записи.

Модель и спецификация

Модель качества обслуживания будет расставлять приоритеты, как указано правилами в спецификации QoS. Данная модель реализована в облачной платформе OpenStack.
Создавать и настраивать спецификации QoS можно с помощью вкладки «Типы дисков», расположенной в разделе «Администратор», подразделе «Диски».



Список типов дисков и спецификаций QoS

Администратору облачной инфраструктуры позволено создание типа диска и спецификации QoS.
Тип диска – это метка, которая может быть выбрана при создании диска. Обычно она характеризует диск по какому-либо критерию, например: «Производительное», «SSD», «Архивное» и т.д.
Спецификация QoS, в данном случае, может быть связана с типами дисков. Это используется для отображения набора возможностей QoS, запрошенных владельцем диска. Привязанная спецификация QoS определяет требуемые уровни QoS, которые интерпретируются системой.

Также, у каждого объекта QoS имеется значение, определяющее область применения. Выделены три области:

- фронтенд (Nova Compute Service);
- бэкенд (Cinder driver);
- и то, и другое (фронтенд и бэкэнд вместе).

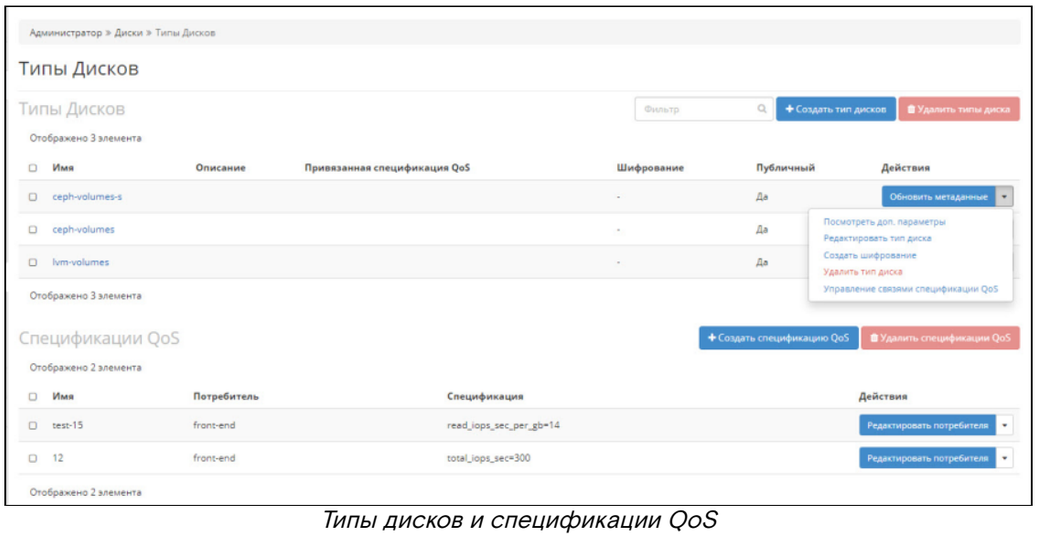
Параметры настройки спецификации QoS:

- Имя спецификации: может быть произвольно задано администратором;
- Потребитель: указывается один из трех видов (областей применения);
- Спецификация: спецификация QoS.

Спецификации QoS могут быть связаны с типами дисков. Для этого используется соответствующая вкладка подраздела «Диски», отображающая набор возможностей QoS, запрошенных владельцем диска.

Управление спецификацией QoS

Действия доступны для выполнения относительно одного выбранного типа диска или спецификаций QoS. Следует выбрать нужное действие в поле «Действия» соответствующей записи в списке и в окне с подробной информацией.



Типы дисков и спецификации QoS

Для спецификаций QoS, в зависимости от статуса, доступны следующие действия:

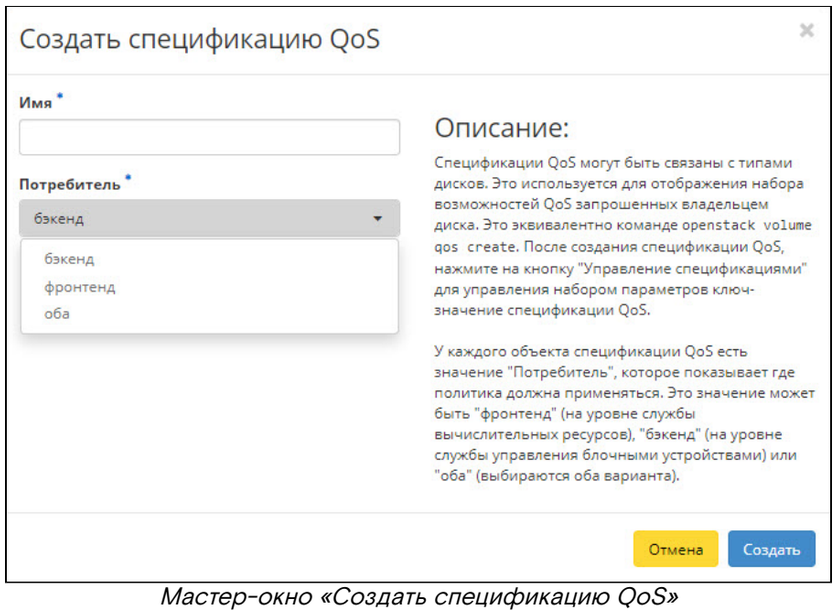
1. Создать спецификацию QoS. Создание спецификации, с указанием имени спецификации и ее потребителя.
2. Управление параметрами. Управление набором параметров ключ-значение спецификации QoS.
3. Редактировать потребителя. Изменение значения «потребитель» в спецификации QoS.
4. Удалить спецификацию QoS. Удаление спецификации QoS.

Также, действия можно запустить в отношении группы предварительно выбранных спецификаций QoS. Для этого необходимо отметить нужные объекты и выбрать групповое действие:

- создание спецификации QoS;
- управление связями спецификации QoS;
- управление параметрами спецификации QoS;
- изменение параметров потребителя.

Создание спецификации

С помощью кнопки [Создать спецификацию QoS] откройте мастер-окно.



Мастер-окно «Создать спецификацию QoS»

Это действие, вызываемое из графического интерфейса управления, эквивалентно команде клиента, выполняемой на контроллере: openstack volume qos create

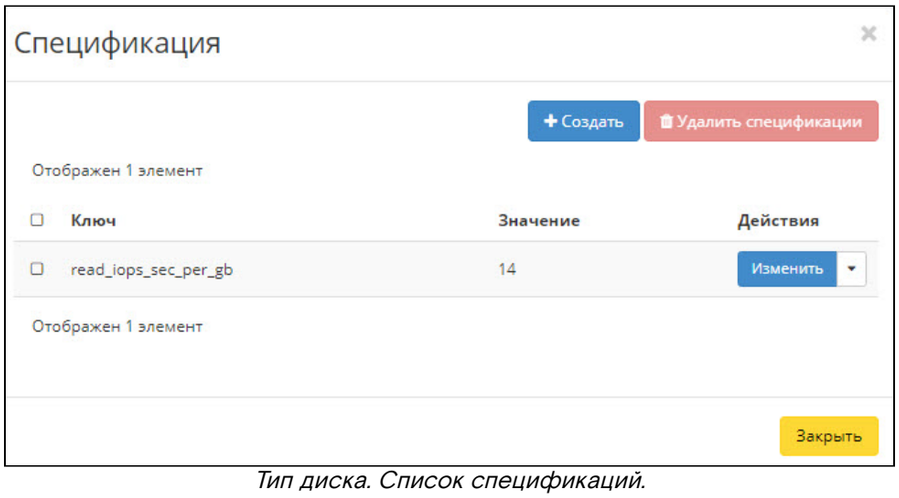
Управление связями спецификации

После создания спецификации QoS нажмите на кнопку [Управление спецификациями] для определения набора параметров ключ-значение спецификации QoS. У каждого объекта спецификации QoS есть значение «Потребитель», которое показывает где политика должна применяться. Это значение может быть одним из:

- front-end: «фронтенд» на уровне службы вычислительных ресурсов);
- back-end: «бэкенд» (на уровне службы управления блочными устройствами);
- both: «оба» (выбираются оба варианта).

Управление параметрами спецификации

Во вкладке «Типы диска» выберите редактируемую спецификацию (секция «Спецификации QOS»). Затем, из контекстного меню кнопки [Редактировать потребителя], выберите – «Управление параметрами». Откроется окно «Спецификация». В нем можно как отредактировать уже заданное значение, так и создать новое. Это позволяет добавлять, изменять или удалять пары «ключ-значение».

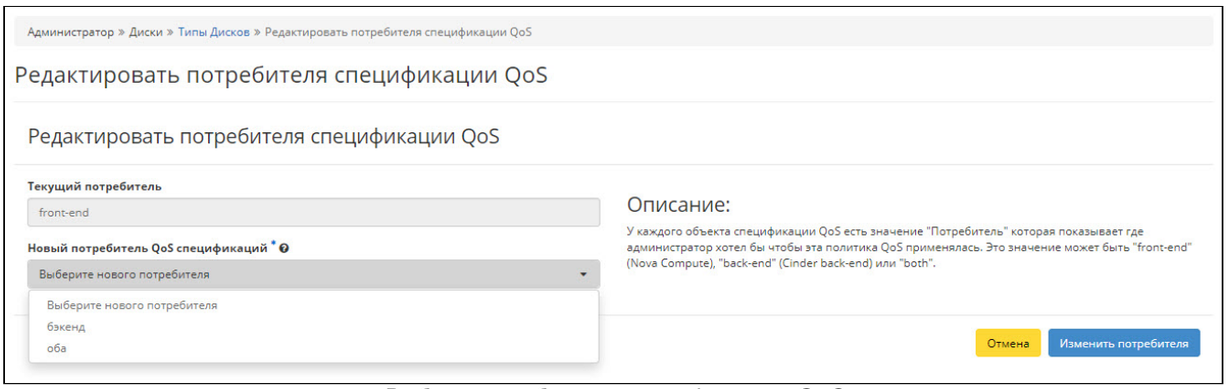


Тип диска. Список спецификаций.

Для ограничения IOPS создается правило спецификации QoS. Создается новая пара ключ-значение «spec» для спецификации QoS «Имя». Допустимые имена ключей ожидаются в спецификациях QoS. Допустимыми значениями для ключа являются: «minIOPS», «maxIOPS» и «burst IOPS».

Редактирование (изменение параметров) потребителя

Во вкладке «Типы диска», в секции «Спецификации QOS» нажмите кнопку [Редактировать потребителя]. В открывшемся мастер-окне выполните необходимые действия и сохраните изменения.



Выбор потребителя спецификации QoS

4.4 Отказоустойчивый кластер (управления)

Кластер – группа серверов, спроектированная в соответствии с методиками обеспечения высокой доступности и гарантирующая минимальное время простоя за счёт конфигурации, содержащей аппаратную избыточность.
Pacemaker – менеджер ресурсов кластера, который позволяет использовать службы и объекты в рамках одного кластера, состоящего из двух или более кластерных нод. Далее такой кластер будет подразумеваться под термином – PCS.
В «классической» архитектуре облачной платформы для управления *отказоустойчивым кластером*, состоящим из нескольких управляющих узлов (контроллеров OpenStack), используется PCS. Данное решение заложено в автоматизированный сценарий развертывания ОП BASIS версии 3.0 (и выше).
Ниже перечислены функциональные возможности Pacemaker:

- позволяет находить и устранять сбои на уровне узлов и служб;
- не зависит от подсистемы хранения (общий накопитель данных не обязателен);
- не зависит от типов ресурсов;
- поддерживает STONITH (Shoot-The-Other-Node-In-The-Head);
- поддерживает кворумные и ресурсозависимые кластеры любого размера;
- поддерживает практически любую избыточную конфигурацию;
- может автоматически реплицировать конфигурационный файл на все узлы кластера (не надо править все вручную);
- можно задать порядок запуска ресурсов, а также их совместимость на одном узле;
- поддерживает расширенные типы ресурсов;
- имеет единую кластерную оболочку CRM с поддержкой скриптов.

Основное назначение PCS, применяемого в ПО Базис.Cloud:

- управление системными службами (сервисами);
- мониторинг состояния служб OpenStack/BASIS и их перезапуск (в случае отказа).

Примечание

Суть технологии STONITH: вышедший из строя узел изолируется и запросы к нему не поступают до тех пор, пока узел не отправит сообщение о том, что он снова находится в рабочем состоянии.

Поддержка расширенных типов ресурсов заключается в следующем: клоны, ресурс которых запущен на множестве узлов, и дополнительные состояния (master/slave и подобное) актуальны для СУБД (MySQL, MariaDB и др.).

Примечание

Сервисы cinder-volume и nova-conductor запускаются в единственном экземпляре, в силу особенностей их работы. Они привязаны только к мастеру Pacemaker.

4.4.1 Ресурсные группы

Сервисы разнесены по так называемым *ресурсным группам*. Для просмотра полного списка ресурсных групп выполните команду на контроллере (master):

```
pcs status --full
```

В выводе утилиты будут отображены все ресурсные группы, настроенные для кластера высокой доступности.
Для кластера, применяемого в рамках ОП BASIS, типовой состав ресурсных групп выглядит следующим образом:

- **Resource group: VIP-group`** – представлена тремя интерфейсами (managment, internal, public), а также сервисом Galera, запущенным на трех контроллерах (master, control2, control3);
- **Resource group: pcs-tnx-nc** – представлены службы BASIS (NodeControl);
- **Resource group: pcs-os-single** – представлены службы OpenStack, а также компоненты кластера (HAproxy, httpd, memcached).

Для просмотра детальной информации о каждой ресурсной группе выполните команду:

```
pcs resource
```

```
Resource Group: vip_group
    management_net_vip (ocf::heartbeat:IPaddr2):       Started ssztnx-ctrl1
Master/Slave Set: p_galera-master [p_galera]
    Masters: [ ssztnx-ctrl1 ssztnx-ctrl2 ssztnx-ctrl3 ]
Master/Slave Set: p_redis-master [p_redis]
    Masters: [ ssztnx-ctrl1 ]
    Slaves: [ ssztnx-ctrl2 ssztnx-ctrl3 ]
Resource Group: pcs_tnx_nc
    p_tionix-node-control-node-tracker (systemd:tionix-node-control-node-tracker):     Started ssztnx-ctrl1
    p_tionix-node-control-node-syncer  (systemd:tionix-node-control-node-syncer):      Started ssztnx-ctrl1
Clone Set: pcs_tnx-clone [pcs_tnx]
    Started: [ ssztnx-ctrl1 ssztnx-ctrl2 ssztnx-ctrl3 ]
Resource Group: pcs_os_single
    p_openstack-nova-conductor (systemd:openstack-nova-conductor):     Started ssztnx-ctrl1
    p_openstack-cinder-volume  (systemd:openstack-cinder-volume):      Started ssztnx-ctrl1
Clone Set: pcs_os-clone [pcs_os]
    Started: [ ssztnx-ctrl1 ssztnx-ctrl2 ssztnx-ctrl3 ]
Clone Set: p_haproxy-clone [p_haproxy]
    Started: [ ssztnx-ctrl1 ssztnx-ctrl2 ssztnx-ctrl3 ]
Clone Set: p_httpd-clone [p_httpd]
    Started: [ ssztnx-ctrl1 ssztnx-ctrl2 ssztnx-ctrl3 ]
Clone Set: p_memcached-clone [p_memcached]
    Started: [ ssztnx-ctrl1 ssztnx-ctrl2 ssztnx-ctrl3 ]
```

Ресурсные группы (листинг)

4.4.2 Настройка инфраструктуры

Предварительная настройка кластера требует настройки преобразования (доменных) имен в IP-адреса, а в некоторых случаях может требоваться обратное преобразование (для разрешения коллизий в работе служб).
Механизм может быть реализован через DNS-сервер (предпочтительный вариант), либо через файл(ы) настройки сетевой конфигурации, содержащей статические адреса и имена – /etc/hosts.
Ниже показано использование варианта с настройкой hosts, который можно использовать не только при отсутствии DNS-серверов, но также при привязке основного имени к нескольким IP-адресам сервера, что может вызвать проблемы при работе Pacemaker.
Внесите в файл(ы) сетевой конфигурации хоста – /etc/hosts – записи для всех серверных нод инфраструктуры:

```
IP_нода1 centos1
IP_нода2 centos2
и т.д.
```

Примечание

Назначение серверных нод не может быть очевидным из файла.

4.4.3 Смена адреса VIP

Могут возникать ситуации, при которых требуется высвободить занятый IP-адрес, назначенный как VIP (точка входа в облачный контроллер). Для смены адреса VIP потребуется выполнить *специальный скрипт* (`change_VIP.sh`), на всех УУ стенда (исходный текст – в конце раздела).

Для изменения VIP-адреса в Pacemaker подключитесь к root@control1 и выполните команду:

```
pcs resource update management_net_vip ip=NEW_IP
```

Проверьте, что в `/etc/` не осталось файлов конфигураций, содержащих старый VIP-адрес, и появился новый (root@control1):

```
grep -rnw '/etc/' -e 'OLD_IP'

# root@control1
grep -rnw '/etc/' -e 'NEW_IP'
```

Возможные проблемы

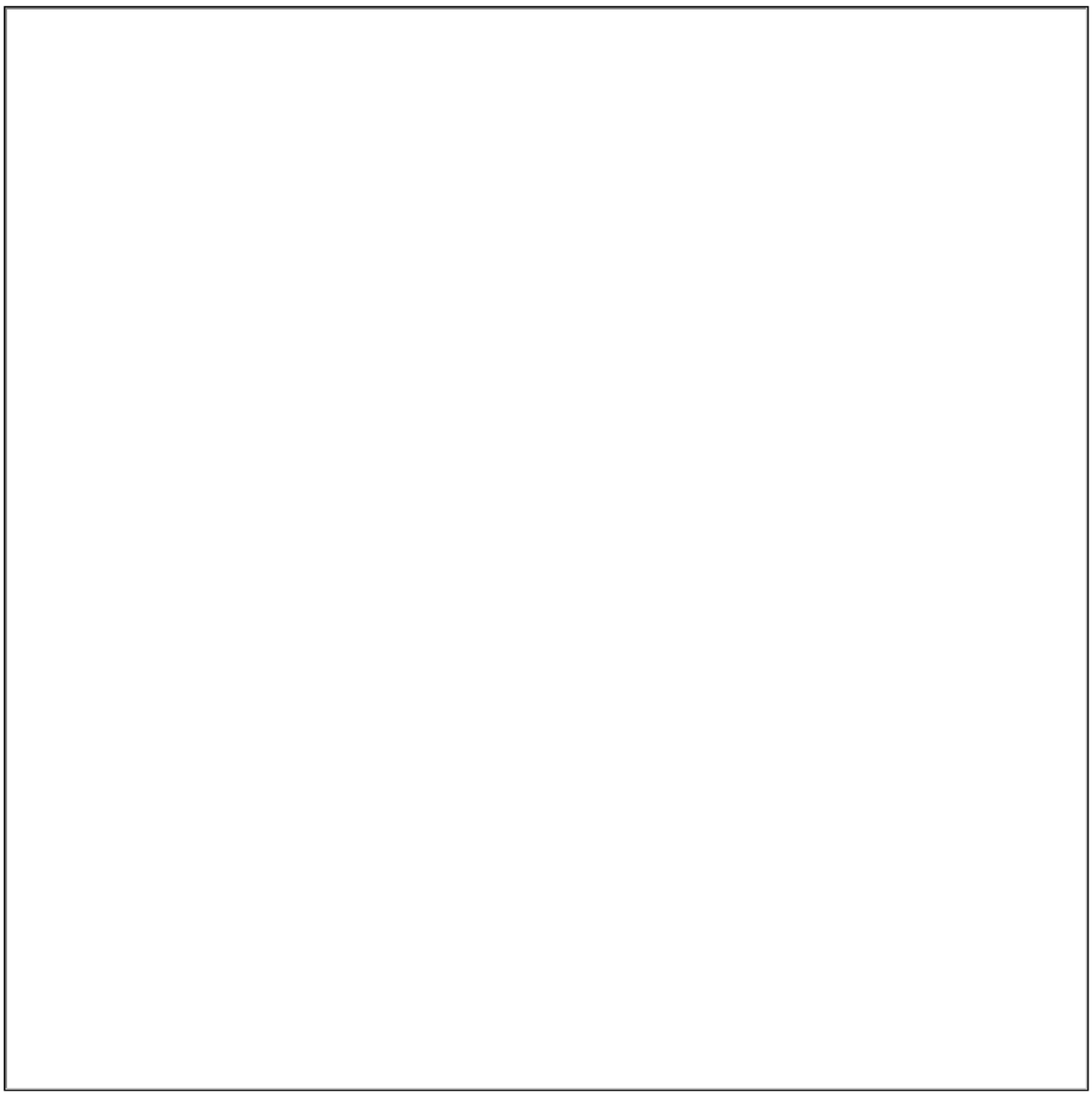
Не отвечает служба Nova.

В логе `/var/log/nova/nova-api.log` ошибка:

> «SQL connection failed. -3 attempts left.: oslo_db.exception.DBConnectionError: (pymysql.err.OperationalError) (2003, «Can’t connect to MySQL server on „10.55.13.9“ ([Errno 113] EHOSTUNREACH)»)

Решение:

подключиться к БД (nova-api) и заменить следующее значение в таблице `cell_mappings`.



Замена (в таблице БД cell_mappings)

Исходный текст скрипта (change_VIP.sh)

```
#!/bin/bash


#old_ip=$1
#new_ip=$2

read -p "Введите старый VIP-адрес: " old_ip
read -p "Введите новый VIP-адрес: " new_ip

###files=(/etc/hosts /etc/haproxy/haproxy.cfg)
files=(/etc/hosts /etc/haproxy/haproxy.cfg /etc/keystone/keystone.conf /etc/glance/
glance-api.conf \
/etc/cinder/cinder.conf /etc/neutron/neutron.conf /etc/nova/nova.conf \
/etc/placement/placement.conf /etc/gnocchi/gnocchi.conf /etc/heat/heat.conf \
/etc/tionix/tionix.yaml /etc/tionix/vdi_server.yaml /etc/sysconfig/memcached \
/etc/openstack-dashboard/local_settings /etc/ceilometer/ceilometer.conf \
/etc/glance/glance-registry.conf)

## Проверка того, что IPv4-адреса введены корректно
echo -e "\nПроверка введенных адресов на корректность: \n"
for ip in $old_ip $new_ip; do

if expr "$ip" : '[0-9][0-9]*\.[0-9][0-9]*\.[0-9][0-9]*\.[0-9][0-9]*$' >/dev/null; then
IFS=.
set $ip
for quad in 1 2 3 4; do
if eval [ \$$quad -gt 255 ]; then
echo "Проверка IPv4 адреса прошла неуспешно - ($ip)"
exit 1
fi
done

echo "Проверка IPv4 адреса прошла успешно - ($ip)"

else

echo "Проверка IPv4 адреса прошла неуспешно - ($ip)"
exit 1
fi

done

## Замена IPv4-адреса в файлах
IFS=' '
for file in ${files[@]}; do

echo -e "\nИзменение файла: ${file}"
sed -i "s/${old_ip}/${new_ip}/gi" $file || break
echo -e "${file} был изменен"

done
```

5 Мониторинг и телеметрия

ОП BASIS реализует функции мониторинга прозрачно для пользователя, при этом предоставляет администратору необходимые сведения о статусах ресурсов, интегрированных в «виртуальный ЦОД».

> **Примечание**
>
> Доступ к сведениям обеспечен для администратора, который выполнил авторизацию через веб-интерфейс.

Метрика программного обеспечения (англ. software metric) – мера, позволяющая получить численное значение некоторого свойства ПО или его спецификаций.

Мониторинг – сбор метрик и представление этих метрик в удобном виде (таблицы, графики, шкалы, уведомления, отчёты). Концептуально, процесс мониторинга может быть изображен так:

Сбор метрик >> Преобразование >> Представление

Система мониторинга – это инструмент анализа фактов и событий, которые происходят/происходили в системе с течением времени. Без понимания значения собранных данных использование *системы мониторинга* является бессмысленным.

Ниже приведена информация, позволяющая администратору облачной инфраструктуры получить общее представление о возможностях системы мониторинга, интегрированной в ПО Базис.Cloud.

Дополнительно, приведена информация, позволяющая выстроить проприетарную систему мониторинга облачной инфраструктуры, учитывающую различные *точки входа* OpenStack.

> **Примечание**
>
> Инструкция по (автоматизированному) развертыванию системы мониторинга и дополнительные сведения о настройке изложены в документе `Руководство по интеграции ПО Базис.Cloud`.

5.1 Модуль Grafana

Grafana – гибко настраиваемое программное средство (инструмент технического мониторинга) и широко используемое для визуализации и анализа данных, как во внутренних, так и во внешних проектах.

При помощи встроенных в Grafana возможностей может осуществляться мониторинг таких объектов облачной инфраструктуры как:

- гипервизор;
- виртуальные машины;
- VDI сервер и сессии.

Например, в детальной информации о гипервизоре отображается следующий набор метрик (телеметрических данных):

- период непрерывной работы гипервизора;
- общее количество ядер процессоров;
- загрузка процессоров на текущий момент, [%];

- средняя загрузка процессоров за последнии 5 минут, [%];
- средняя загрузка процессоров за последнии 15 минут, [%];
- общий объем доступной оперативной памяти, [Gb];
- использование оперативной памяти на текущий момент, [%];
- общий объем доступной SWAP памяти, [Gb];
- использование SWAP памяти на текущий момент, [%];
- общий объем доступной памяти на корневой файловой системе, [Gb];
- процент использование дискового пространства на корневой файловой системе, [%];
- график загрузки CPU;
- график загрузки RAM;
- график утилизации дисковой подсистемы;
- график IOPS дисковой подсистемы;
- график Read-Write.

Если мониторинг осуществляется для виртуальных машин, то для них могут быть получены следующие метрики:

- общее количество виртуальных ядер;
- общее количество виртуальной оперативной памяти;
- общий объем виртуальных дисков;
- график использования vCPU;
- график использования vRAM;
- график Disk Read;
- график Disk Write;
- график Network TX;
- график Network RX.

Примечание

Свод графиков, выстроенных в единое представление, называется дашбордом (dashboard). Установка Graphana может быть выполнена при помощи ПО контейнеризации Docker

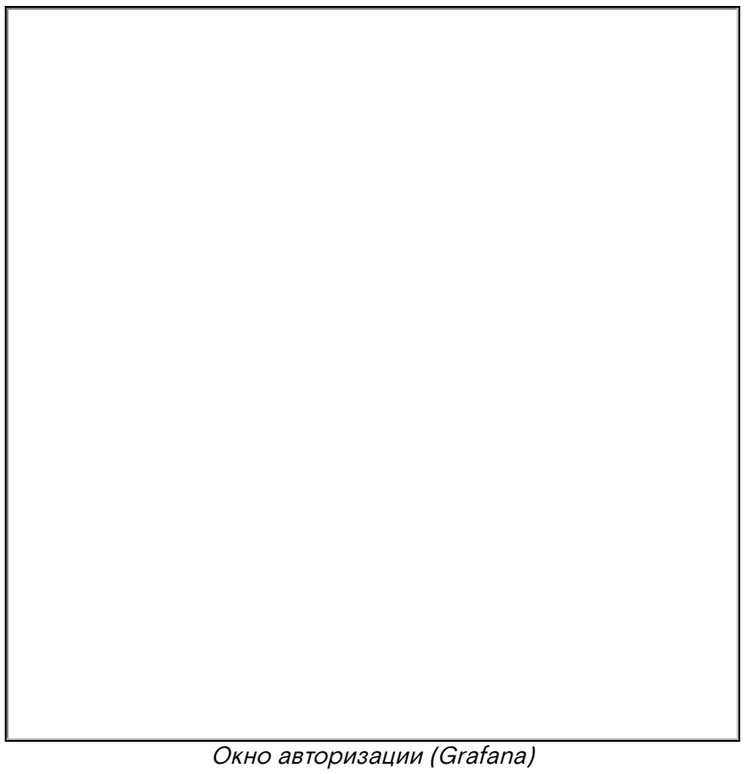
5.1.1 Веб-интерфейс Grafana

Доступ к наблюдению метрик, собираемых службой телеметрии, осуществляется с помощью средств веб-администрирования, встроенных в Grafana.

Администратору облачной инфраструктуры для выполнения входа в Grafana потребуются следующие реквизиты доступа:

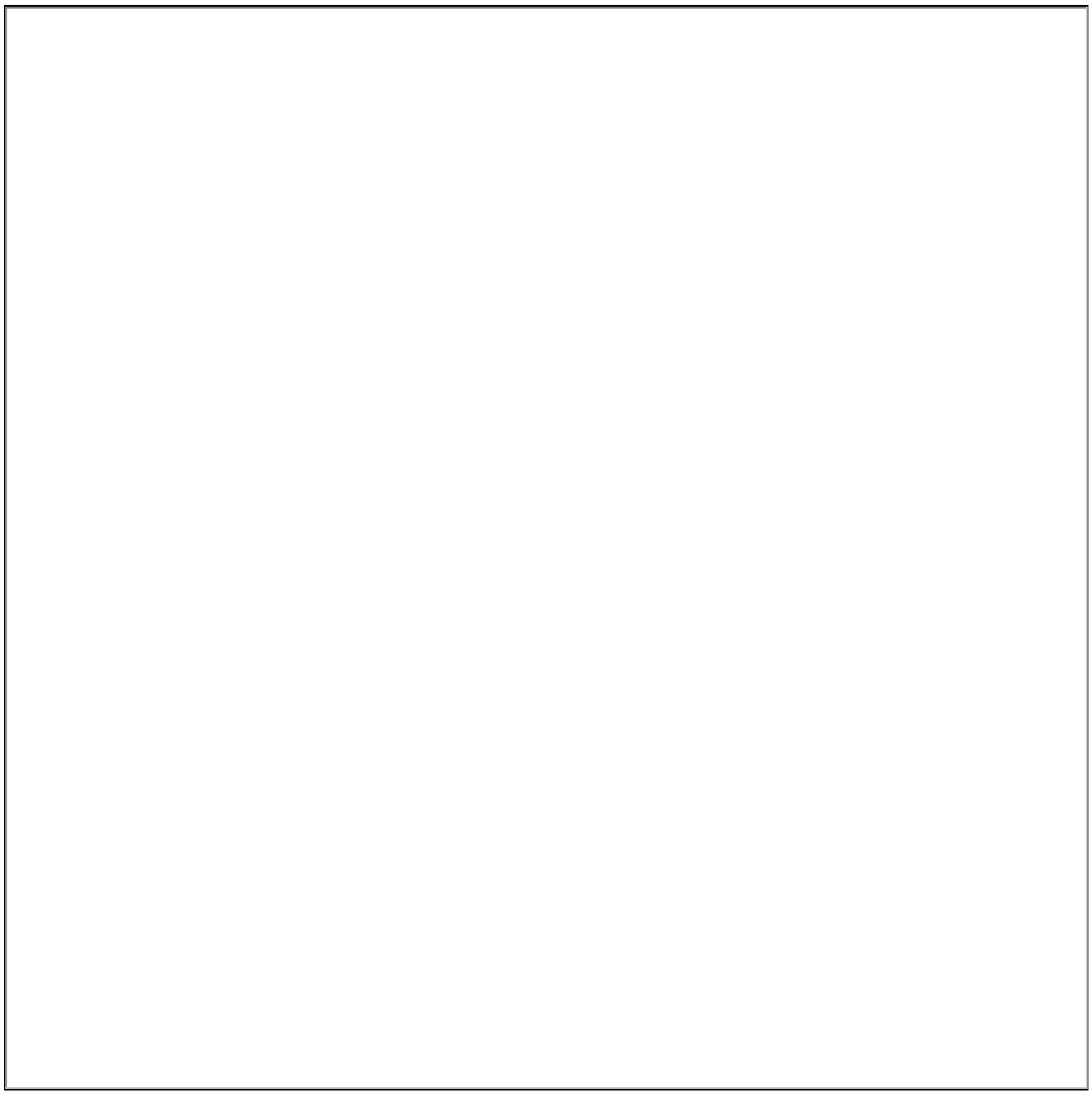
1. http://<IP_узла_мониторинга>:3000/
2. Имя пользователя (для подключения к серверу мониторинга).
3. Пароль (для входа в WebUI).

Реквизиты вводятся в окне авторизаци , открытом в веб-браузере.



Окно авторизации (Grafana)

После успешной авторизации будет выведено сообщение (на зелёном фоне) и произведена инициализация/отображение графического интерфейса в том же окне, из которого выполнен вход.

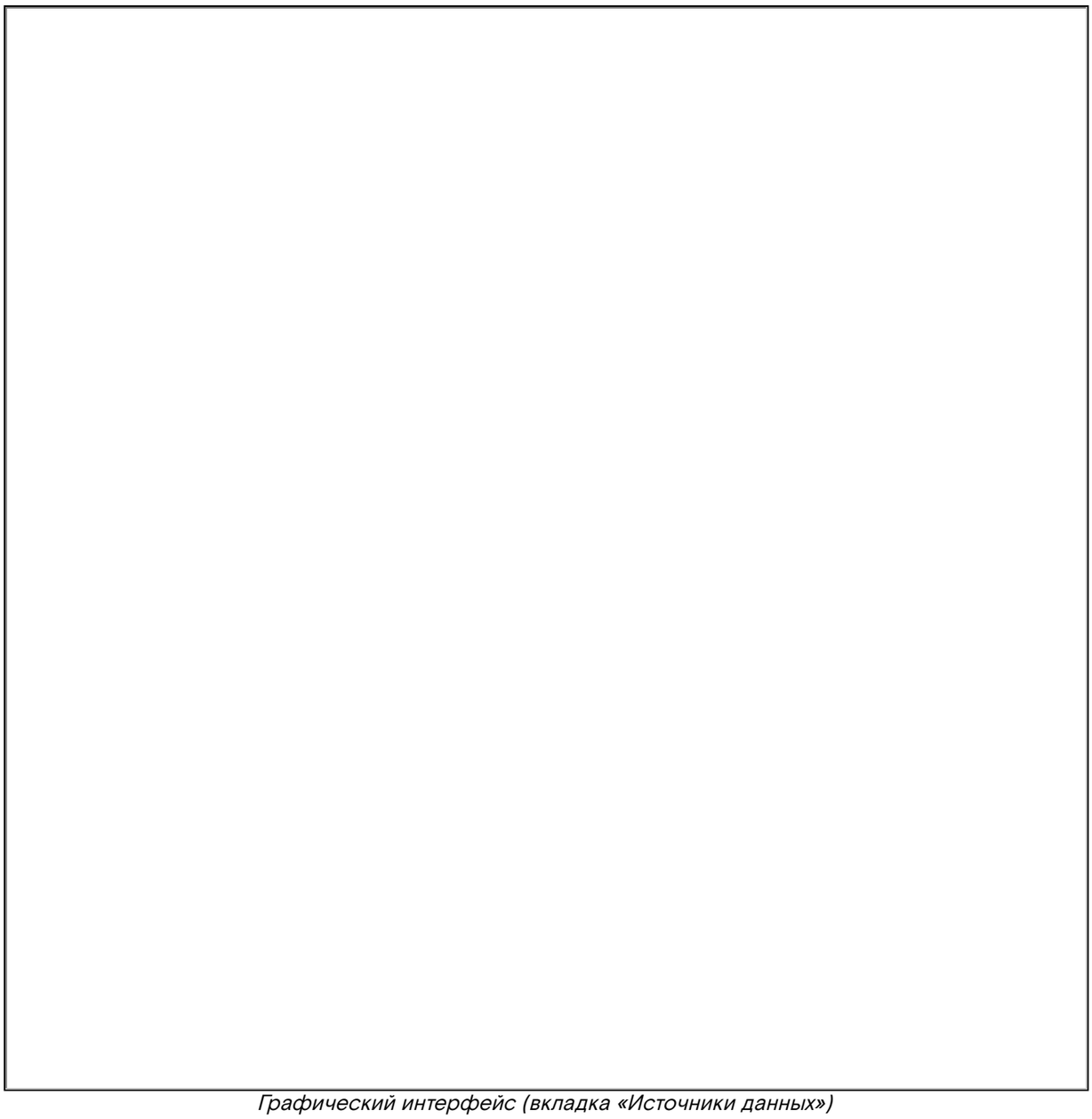


Графический интерфейс (Grafana)

5.1.2 Источники данных

Создание дашборда невозможно без указания источника данных, поэтому в случае установки Grafana вручную («с нуля») должен быть создан как минимум один источник. Для этого выполните перечисленные ниже действия.

Выберите из меню пункт – Data Sources – откроется список предварительно сконфигурированных источников данных.



Графический интерфейс (вкладка «Источники данных»)

Кликните кнопку [Add data source], чтобы просмотреть список всех поддерживаемых типов источников данны .

Наведите курсор на интересующий источник данных и кликните мышью по кнопке [Select].

Сконфигурируйте источник данных, следуя инструкциям касательно этого источника.

Следует отметить, что существует способ ограничить права доступа пользователей к источникам данных, описываемый в официальной документации.

> **Внимание**
>
> По умолчанию, источники данных инфраструктуры могут быть запрошены любым пользователем. Например, пользователь с ролью Viewer может выполнить *любой запрос к источнику*, а не только существующие запросы дашбордов, к которым они относятся.

Чтобы включить разрешения доступа к источнику данных, перейдите: Configuration > Data Sources.

Выберите источник данных, для которого необходимо включить разрешения (права доступа). На вкладке «Permissions» кникните кнопку [Enable].

5.1.3 Панели и дашборды (визуализация)

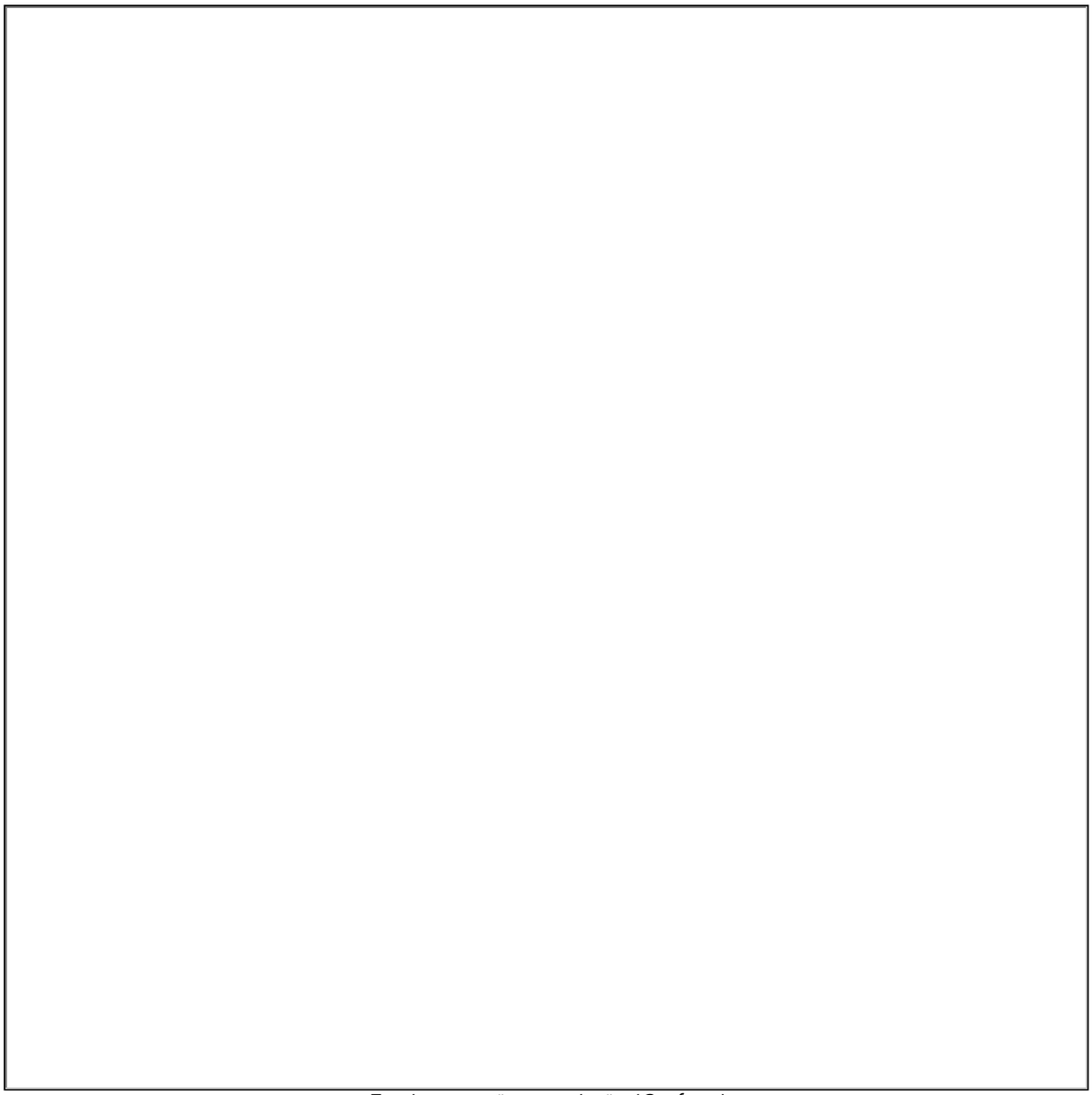
Панели и дашборды – основные сущности представлений внутри Grafana. Каждый дашборд состоит из *набора панелей*. Для создания нового дашборда следует перейти в представление (Dashboards) и нажать кнопку [New Dashboard].

Затем следует добавить на дашборд новую **панель**. В Grafana доступны предустановленные панели, которые можно сразу начать использовать. По умолчанию, представлены следующие типы панелей:

- Graph — панель с графиками с возможностью комбинировать несколько метрик на одной панели;
- Stat (ранее SingleStat) — панель с одиночным графиком и возможностью отображения моментального значения метрики;
- Gauge — панель в формате спидометра, есть возможность ограничить верхнее значение на шкале;
- Bar Gauge — панель с возможностью отображения нескольких метрик на вертикальной гистограмме;
- Table — панель с представлением в виде таблицы, на которой можно отображать значения нескольких метрик;
- Text — панель для отображения произвольного текста (подписи);
- Heatmap — панель для отображения тепловой карты значений метрик;
- Alert list — панель для отображения событий из внешних систем;
- Dashboard list — комбинированная панель для отображения дашбордов, добавленных в избранное;
- News — панель для отображения новостной ленты из внешних источников;
- Logs — панель для отображения строчек лога, которые собираются одной из внешних систем;
- Zabbix problems — панель для отображения событий из системы мониторинга Zabbix.

Далее, выберите в выпадающем меню источник данных (например – Zabbix) и укажите группу, хост, приложение и элемент данных. Если всё выполнено правильно, то на графике появятся данные, полученные из указанного источника.

При создании новой панели, в правой части экрана (в разделе Visualization) есть возможность выбрать *тип визуализации* .



Графический интерфейс (Grafana)

Каждая панель каждого дашборда в Grafana обладает собственным набором настроек. Настройки первой панели открываются из выпадающего меню, а настройки второй могут быть открыты нажатием на пиктограмму «шестеренка», расположенную в верхнем правом углу экрана.

5.1.4 Импорт и экспорт дашборда

В Grafana доступен функционал импорта и экспорта дашборда.

Для импорта следует перейти в представление Dashboards и нажать кнопку [Import]. Доступны две опции импорта:

- загрузка произвольного JSON-файла;
- импорт готовых дашбордов.

Во втором случае – для импорта готового дашборда из репозитория – следует указать `ID` импортируемого дашборда.

Экспорта дашборда активируется нажатием на специальную иконку, расположенную в верхней части экрана. Далее, потребуется нажать [Save to file] – дашборд будет сохранен в файл формата JSON.

5.1.5 Конфигурационный файл

Для «тонкой» настройки Grafana потребуется модифицировать конфигурационный файл (по умолчанию – `/etc/grafana/grafana.ini`). В нём хранятся параметры, которые при необходимости можно изменять.

Основные секции с настроечными параметрами:

```
[paths]

[server]

[database]

[security]

[users]

[session]

[analytics]

[dashboards.json]
```

5.2 Мониторинг облачных сервисов

Для выполнения JSON-запросов прежде всего необходимо получить аутентификационный токен. Для этого следует выполнить команду:

```
curl -v -H 'Content-Type: application/json' -d '{"auth":{"identity":{"methods":
["password"],"password":{"user":{"name":"admin","domain":
{"name":"default"},"password":"<password>"}}}}}' http://manage.tionix.loc:35357/v3/auth/
tokens
```

Токен содержится в заголовке ответа `X-Subject-Token`. Используя этот токен, могут быть выполнены другие запросы, осуществляющие проверку *облачных сервисов*.

С помощью полученного токена осуществляется проверка работоспособности следующих служб OpenStack:

- Nova (Compute Service);
- агенты Neutron (Network Service);
- Cinder (Block storage Service).

5.2.1 Проверка сервисов Nova

```
curl -H ‘X-Auth-Token: <token>’ http://<IP-адрес_облачного_контроллера>:8774/v2.1/os-
services
```

Ответом будет список сервисов, неисправными являются те, у которых параметр `state == down`, в то время как параметр `status == enabled`.

5.2.2 Проверка состояния агентов Neutron

```
curl -H 'X-Auth-Token: <token>' http://<IP-адрес_облачного_контроллера>:9696/v2.0/agents
```

Ответом будет список агентов, неисправными являются те, у которых параметр `alive == false`, в то время как параметр `admin_state_up == true`.

5.2.3 Проверка сервисов Cinder

```
curl -H 'X-Auth-Token: <token>' http://<IP-адрес_облачного_контроллера>:8776/v3/
<projectid>/os-services
```

Ответом будет список сервисов.

Неисправными считаются те сервисы, у которых параметр `state == down`, в то время как параметр `status == enabled`.

Идентификатор проекта *projectid* может быть получен при помощи запроса:

```
curl -H 'X-Auth-Token: <token>' http://<IP-адрес_облачного_контроллера>:35357/v3/projects
```

5.3 Мониторинговые запросы

Ниже приведены **мониторинговые запросы**, которые могут быть выполнены как из командной строки (Linux), так и с помощью произвольно построенной *внешней системы мониторинга* .

Примечание

Способ программного анализа ответа на запрос (объекта JSON[1]) не предоставляется в рамках эксплуатационной документации.

5.3.1 Запрос к службе TIONIX.NodeControl

```
curl -H 'X-Auth-Token: <token>' http://<IP-адрес_облачного_контроллера>:9362/v1/info
```

Ответом должен быть HTTP код 200 и объект JSON с информацией о модуле.

5.3.2 Запрос к службе TIONIX.Monitor

```
curl -H 'X-Auth-Token: <token>' http://<IP-адрес_облачного_контроллера>:9363/v1/info
```

Ответом должен быть HTTP код 200 и объект JSON с информацией о модуле.

5.3.3 Запрос к службе TIONIX.VDIserver

```
curl -H 'X-Auth-Token: <token>' http://<IP-адрес_облачного_контроллера>:9364/v1/info
```

Ответом должен быть HTTP код 200 и объект JSON с информацией о модуле.

5.4 Подключение внешних систем мониторинга

Для подключения внешних систем мониторинга необходима подробная информация о сетевых службах и портах, используемых при взаимодействии между службами OpenStack и модулями BASIS.

1 https://handbook.basistech.ru/CP/Setup/index.html#term-JSON

Построение системы мониторинга облачной инфраструктуры должно быть основано на владении подробной информацией о внутренних процессах взаимодействия между отдельными компонентами ПО, а также подсистемами обслуживания (Control, Network, Compute).

Большинство служб OpenStack и ключевые модули BASIS реализуют модель сетевого взаимодействия, основанную на использовании (сетевых) протоколов, в частности – REST . Этот протокол может быть использован для seamless-интеграции системы мониторинга, формирующей запросы, отправляемые к тем или иным службам/модуля .

Обмен данными в REST происходит с помощью методов HTTP: GET, POST, PUT, DELETE.

Ниже приведена информация, позволяющая организовать «сторонний» мониторинг подсистем, образующих:

– кластер, состоящий из одного и более управляющих узлов (контроллеров OpenStack);

– вычислительный пул, состоящий из нескольких вычислительных узлов, управляемых при помощи кластера.

5.4.1 Сервисы управляющих узлов

В каждом из контроллеров OpenStack – управляющих узлов кластера – должны работать определенные системные службы (сервисы). Полный список (перечень) приведен ниже .

Перечень сервисов облачного контроллера

№	Название системной службы
1	corosync
4	haproxy
5	Httpd 2[2]
6	memcached
7	neutron-dhcp-agent
8	neutron-l3-agent
9	neutron-metadata-agent
10	neutron-openvswitch-agent
11	neutron-server
14	openstack-cinder-api
15	openstack-cinder-scheduler
16	openstack-cinder-volume
17	openstack-glance-api
18	openstack-glance-registry
19	openstack-nova-api
20	openstack-nova-conductor

2 https://handbook.basistech.ru/CP/Operator/MONI_External.html#id15

№	Название системной службы
21	openstack-nova-consoleauth
22	openstack-nova-novncproxy
23	openstack-nova-scheduler
24	pacemaker
25	pcsd
26	rabbitmq-server
27	tionix-journal-api
28	tionix-journal-keystone-listener
29	tionix-journal-listener
30	tionix-journal-nova-listener
31	tionix-monitor-api
32	tionix-monitor-nova-listener
33	tionix-monitor-tionix-listener
34	tionix-node-control-agent
35	tionix-node-control-api
36	tionix-node-control-drs-trigger
37	tionix-node-control-node-syncer
38	tionix-node-control-node-tracker
39	tionix-node-control-nova-listener
40	tionix-node-control-worker
41	tionix-scheduler-beat
42	tionix-scheduler-worker
43	tionix-vdi-broker-api
44	tionix-vdi-keystone-listener
45	tionix-vdi-neutron-listener
46	tionix-vdi-nova-listener

№	Название системной службы
47	tionix-vdi-project-syncer
48	tionix-vdi-server-api
49	tionix-vdi-worker

5.4.2 Сервисы вычислительных узлов

В каждом вычислительном узле облачной инфраструктуры должны работать системные службы, обеспечивающие нормальное функционирование и доступность гипервизора. Перечень сервисов приведен ниже .

Перечень сервисов вычислительных узлов облачной инфраструктуры

№	Название программного пакета
1	libvirtd
2	neutron-openvswitch-agent
3	openstack-ceilometer-compute
4	openstack-nova-compute
5	tionix-agent

5.4.3 Перечень портов для мониторинга

С целью обнаружения потенциально аварийных ситуаций необходимо регулярно проводить проверку доступности сетевых портов контроллера. Перечень представлен ниже .

Перечень сетевых портов контроллера

Название порта	Номер порта
mysql	3306
memcached	11211
rabbitmq	5672
keystone	5000
keystone	35357
glance-api	9292
glance-registry	9191
neutron-server	9696
nova-api	8775
nova-api	8774

Название порта	Номер порта
nova-placement	8778
cinder-api	8776
gnocchi-api	8041
tionix-journal-api	9360
tionix-monitor-api	9363
tionix-node-control-api	9362
tionix-scheduler-beat	10001
tionix-vdi-broker-api	9365
tionix-vdi-server-api	9364

Регулярность проверки (доступности) сетевых портов не регламентирована. Если не указано иного, то частота проверки «один раз в час» будет достаточной для того, чтобы организовать оперативное реагирование.

Внимание

Чрезмерно частая проверка сетевых портов может повлечь за собой неустойчивость работы зависимых служб!

5.4.4 Проверка статусов служб

Некоторые службы OpenStack позволяют выполнение HTTP-запроса, осуществляющего так называемую «проверку здоровья»:

- Keystone: http://manage.<домен>:35357/healthcheck;
- Gnocchi: http://internal.<домен>:8041/healthcheck;
- Glance API: http://internal.<домен>:9292/healthcheck;
- Glance Registry: http://internal.<домен>:9191/healthcheck.

При успешной работе службы возвращается HTTP-код – 200.

5.4.5 Извлечение данных телеметрии

Служба телеметрии (Ceilometer) архитектурно реализована агентами. Некоторые модули телеметрии совмещают функциональность сбора данных, хранения сэмплов в БД, или предоставляют службу API, обрабатывающую *входящие запросы*.

Принципы работы службы изложены в официальной документации.

Агент вычислений (ceilometer-agent-compute) выполняется на каждом ВУ облачной инфраструктуры и отрабатывает (polls) статистику использования ресурсов. Фактически, `ceilometer-polling` – the polling agent – запускается с параметром `--polling-namespace compute`.

Центральный агент (ceilometer-agent-central) выполняется на центральном сервере обслуживания, чтобы отрабатывать (poll) статистику использования ресурсов для тех ресурсов, которые не связаны с инстансами или ВУ. Множество агентов может запускаться с целью *горизонтального масштабирования* (телеметрической) службы. Фактически, таким агентом является `ceilometer-polling`, запускаемый с параметром `--polling-namespace central`.

Агент уведомлений (ceilometer-agent-notification) выполняется на одном или нескольких центральных серверах обслуживания и потребляет сообщения, помещенные в очередь(и) сообщений, чтобы создавать (to build) данные событий и метрик. После чего данные публикуются в определенных

(целевых) местах. По умолчанию, данные отправляются в Gnocchi (базу данных временных рядов), рекомендуемую для эффективного хранения и статистического анализа телеметрических данных.

Все перечисленные выше агенты взаимодуйствуют посредством шины сообщений OpenStack. Данные телеметрии спроектированы таким образом, чтобы они могли публиковаться в различные конечные точки (endpoints), а уже там накапливаться и анализироваться.

Для извлечения данных телеметрии программным способом может использоваться телеметрическое API 5[3], реализованное в виде стандартного протокола обмена REST API. Собранные сэмплы и связанная информация извлекается в виде списка метрик, определений тревог и т.д.

Ссылка на телеметрическое API URL может быть получена из сервисного каталога, предоставленного службой идентификации (OpenStack Identity), доступной (which is populated) втечение процесса инсталляции. Доступ к API для получения метрических данных требует предоставления действующего токена и подходящих прав 6[4].

Кроме того, поддерживаются бэкэнды тревог:

- MySQL;
- PostgreSQL.

Для получения событий поддерживаются следующие бэкэнды:

- ElasticSearch;
- MongoDB;
- MySQL;
- PostgreSQL;
- HBase.

3 https://handbook.basistech.ru/CP/Operator/MONI_External.html#id18
4 https://handbook.basistech.ru/CP/Operator/MONI_External.html#id19

6 Автоматическое конфигурирование ОС и оркестрация

User-data – блок данных, который пользователь может указать при запуске ВМ. ВМ может получить доступ к этим данным через службу метаданных или диск конфигурации. Обычно user-data используется для передачи сценария оболочки, запускаемого экземпляром при загрузке.

Например, одним из приложений, использующих пользовательские данные, является `cloud-init` – пакет с открытым исходным кодом, доступный в различных дистрибутивах Linux и других ОС. Он обеспечивает раннюю инициализацию ВМ в облачной платформе.

6.1 Использование user-data при ручном создании ВМ

Важно

Должны использоваться только подготовленные облачные образы, с установленным `cloud-init`. Информацию и подготовке облачных образов можно получить из документа `Руководство по миграции инфраструктуры`.

Примечание

Файл формата YAML критичен к отступам текста при описании параметров.
Рекомендуется использовать отступ в 4 пробела относительно объявления группы.

Доступно несколько вариантов передачи user-data ВМ:

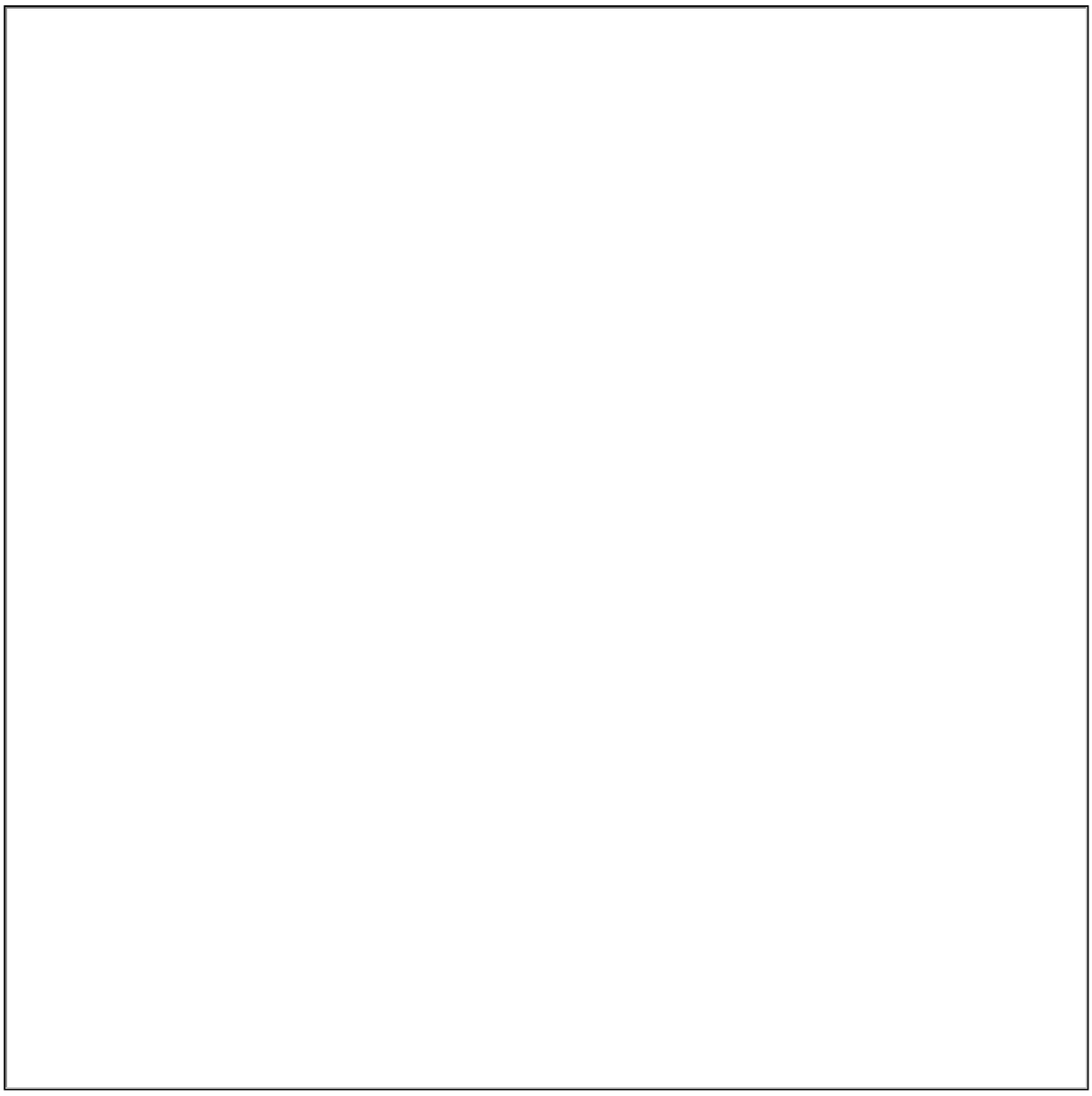
1. Помещение user-data в локальный файл и передача через параметр „-user-data <user-data-file>“, при создании ВМ с помощью CLI:

```
openstack server create --image cloudimage --flavor flavor-1 \ --key-name key1 --nic
net-id=external --user-data user-data.yaml VM_INSTANCE
```

где:

- cloudimage – наименование образа системы;
- flavor-1 – тип инстанса;
- user-data.yaml – файл с user-data в формате YAML;
- VM_INSTANCE – наименование создаваемой ВМ;
- net-id=external – наименование подключаемой сети;
- key1 – наименование ключевой пары.

2. При использовании Dashboard можно передать user-data через поле «Сценарий настройки», поместив содержание файла с user-data в формате YAML .



Передача user-data через поле «Сценарий настройки»

6.1.1 Использование скриптов

С помощью user-data также можно передавать *скрипты*, предназначенные для выполнения в ВМ. В начало скрипта, руководствуясь данными таблицы , введите директиву, определяющую системную оболочку (используемую для выполнения скрипта).

Описание скриптов

Оболочка выполнения	**Директива**	**Описание**
CMD	rem cmd	Будет выполнен cmd.exe скрипт
PowerShell	#ps1 или #ps1_sysnative	Будет выполнен powershell скрипт для Windows 64bit
	#ps1_x86	Будет выполнен powershell скрипт для Windows 32bit
Bash	#!/bin/bash	Будет выполнен bash скрипт
Python	#!/usr/bin/env python	Будет выполнен python скрипт

Пример скрипта для Bash (Linux):

```
#!/bin/bash
hostnamectl set-hostname control1
timedatectl set-timezone Europe/Moscow
echo "nameserver 8.8.8.8" > /etc/resolv.conf
apt-get update
apt-get upgrade
```

6.1.2 Назначение пароля пользователю

Пример назначения пароля пользователю Linux:

```
#cloud-config
groups:
- cloud-group
users:
- default
- name: cloud-user
primary_group: cloud-user
groups: wheel
lock_passwd: false
plain_text_passwd: 45697845
```

где:

- `45697845` – пароль;
- `cloud-user` – имя пользователя.

Пример установки пароля пользователю для Windows:

```
#ps1_sysnative
$password=convertto-securestring "45697845" -asplaintext -force
New-LocalUser "cloud-user" -Password $password
Add-LocalGroupMember -SID "S-1-5-32-544" -Member "cloud-user"
Restart-Computer
```

6.1.3 Примеры часто используемых операций

Настройка временной зоны в системе:

```
#cloud-config
timezone: Europe/Moscow
```

Просмотр списка доступных временных зон из консоли:

```
timedatectl list-timezones
```

Установка пакета `ansible`:

```
#cloud-config
packages:
- ansible
```

Установка пакета `chronyd`, запуск сервиса и добавление в автозагрузку:

```
#cloud-config
packages:
- chronyd
runcmd:
- [ systemctl, daemon-reload ]
- [ systemctl, enable, chronyd.service ]
- [ systemctl, start, chronyd.service ]
```

6.2 Использование встроенного оркестратора Heat

Heat – сервис оркестрации составных облачных приложений с использованием шаблона через собственный REST API OpenStack. Heat обеспечивает оркестровку на основе шаблонов путем выполнения соответствующих вызовов API для запуска ВМ в среде OpenStack.

Шаблоны Heat описывают инфраструктуру в текстовых файлах, которые удобно корректировать под конкретные задачи, и которыми можно управлять с помощью инструментов контроля версий.

Шаблоны позволяют создавать различные типов ресурсов OpenStack (например, ВМ, плавающие IP-адреса, тома, группы безопасности, пользователей и т. д.).

Важно

Программное обеспечение проекта Heat связано с другими компонентами OpenStack.
Одной из важных функций является возможность передачи user-data с помощью шаблонов сервиса Heat. Информацию по использованию user-data можно получить выше.

6.2.1 Установка оркестратора Heat

Установка сервиса осуществляется с помощью плейбуков Ansible. О подготовке рабочего окружения можно прочитать в документе Инструкция по развертыванию.

Установка выполняется с помощью сценария `deploy.yml` с использованием тега „Heat“:

```
ansible-playbook deploy.yml -t heat -vv
```

Сервисы Openstack Heat:

- openstack-heat-api
- openstack-heat-api-cfn
- openstack-heat-engine

Внимание

В случае если в архитектуре используется менеджер ресурсов Pacemaker, рекомендуется добавить сервисы Heat в группу к остальным сервисам Openstack.
Пример команд для добавления сервисов в Pacemaker:

```
pcs resource create p_openstack-heat-api systemd:openstack-heat-api meta
interleave=true --group pcs_os op monitor interval=15s
pcs resource create p_openstack-heat-api-cfn systemd:openstack-heat-api-cfn meta
interleave=true --group pcs_os op monitor interval=15s
pcs resource create p_openstack-heat-engine systemd:openstack-heat-engine meta
interleave=true --group pcs_os op monitor interval=15s
```

6.2.2 Структура шаблонов

Шаблоны оркестратора Heat имеют формат YAML и структуру, описанную ниже:

heat_template_version: 2016-10-14 – параметр, имеющий значение от 2013-05-23 (или более поздней даты), указывает, что документ YAML является шаблоном оркестратора Heat указанной версии;

description – раздел, в котором указывается описание шаблона;

Этот раздел является необязательным и при необходимости может быть пропущен.

parameter_groups – в этом разделе можно указать, как следует группировать входные параметры и в каком порядке вводить параметры;

Этот раздел является необязательным и при необходимости может быть пропущен.

parameters – в этом разделе можно указать входные параметры, которые должны быть заданы при обработке шаблона;

Этот раздел является необязательным и может быть пропущен, если ввод данных не требуется.

resources – в этом разделе задаются ресурсы, используемые при обработке шаблона;

Этот раздел является обязательным для любого шаблона.

outputs – этот раздел позволяет указать исходящие параметры, доступные пользователям после создания экземпляров шаблона;

Этот раздел является необязательным и может быть пропущен, если выходные значения не требуются.

conditions – раздел включает в себя условия, которые можно использовать как ограничения при создании ресурса или при определении свойств. Они могут быть связаны с ресурсами и свойствами ресурсов в разделе «Ресурсы», а также могут быть связаны с выходными данными в разделе «Выходные данные».

Этот раздел является необязательным и может быть пропущен, если не требуется выполнение условий.

Важно

Наиболее подробную информацию можно узнать в дашборде:

- по функциям определенной версии шаблона: во вкладке Оркестрация > Версии шаблонов
- по типам ресурсов (атрибуты, свойства): во вкладке Оркестрация > Типы ресурсов.

6.2.3 Создание стека в OpenStack CLI

Для создания стека требуется выполнить следующую команду в OpenStack CLI:

```
openstack stack create -e <environment file name>  --parameter "<parameter>=<value>" -t
<template name> <stack name>
```

Например, для создания стека по шаблону `basic-stack.yaml` с названием **basic-stack** требуется выполнить команду:

```
openstack stack create -e env.yaml --parameter
"server_image=cirros;server_network=external" -t basic-stack.yaml basic-stack
```

Данная команда возвращает вывод следующего вида:

```
+---------------------+--------------------------------------+
| Field | Value |
+---------------------+--------------------------------------+
| id | 6be269ec-d22f-4cf0-bf04-df71cc6dcf75 |
| stack_name | basic-stack |
| description | No description |
| creation_time | 2021-011-12T15:58:20Z |
| updated_time | None |
| stack_status | CREATE_IN_PROGRESS |
| stack_status_reason | Stack CREATE started |
+---------------------+--------------------------------------+
```

Для удобства можно использовать bash-скрипт, запускающий создание стека:

```
#!/bin/bash
set -e
### deploy instance
openstack stack create CIRROS \
--template cirros.yaml \
--parameter server_name=cirros-server \
--parameter server_network=external \
--parameter server_image=cirros \
--parameter server_flavor=1x1x0 \
--parameter key_name=key-1
```

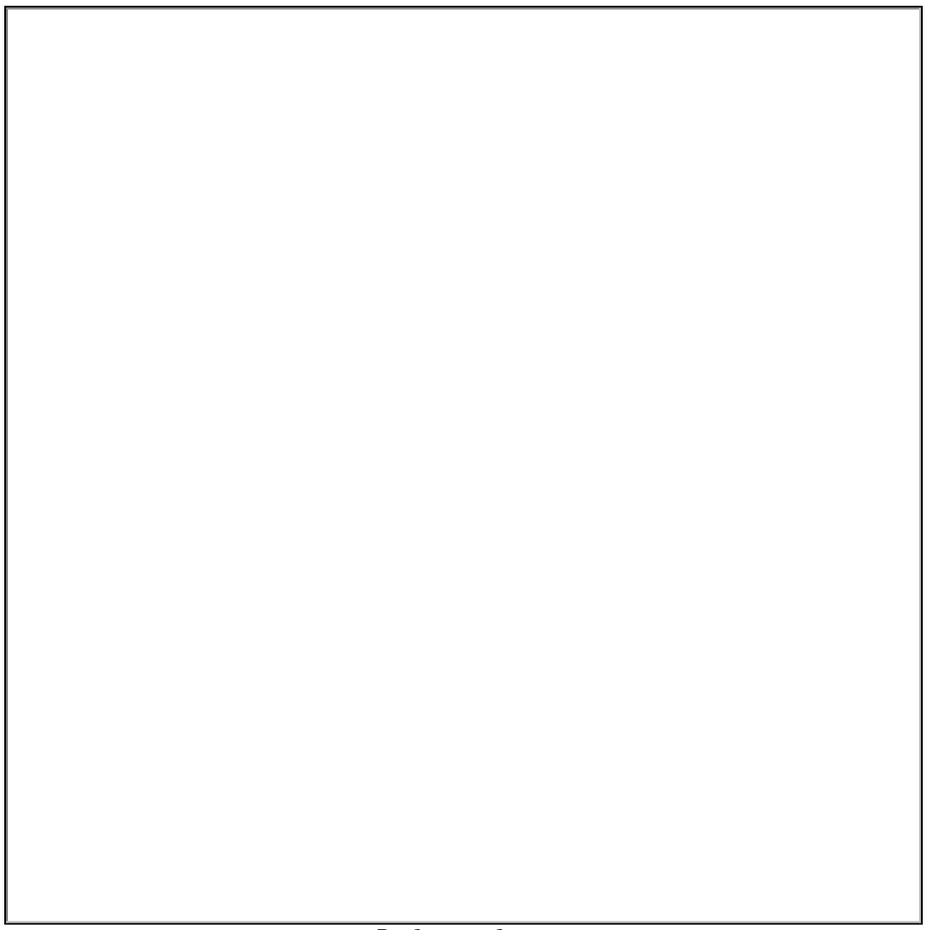
6.2.4 Создание стека в Dashboard

Для создания стека требуется выполнить ряд действий:

1. Перейти: Проект >> Оркестрация >> Стеки .

Стеки

2. Нажать на кнопку **Запустить стек** .



Выбор шаблона

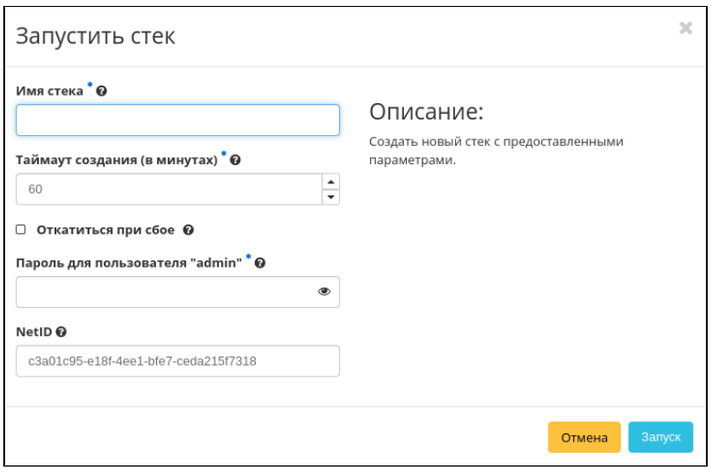
3. Выбрать источник шаблона и источник среды:

Доступные *источники шаблона*:

- Файл;
- Непосредственный ввод;
- Адрес.

Доступные *источники среды*:

- Файл;
- Непосредственный ввод.



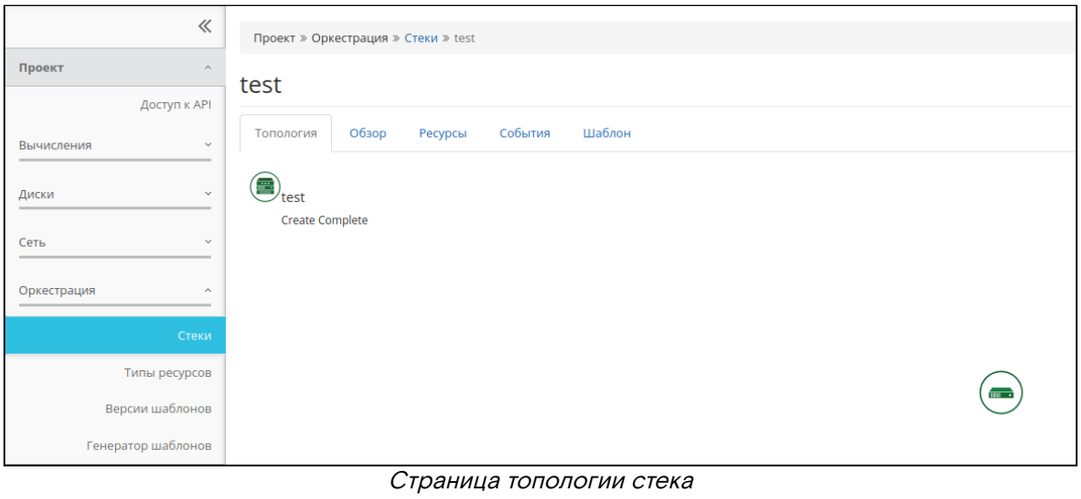
Запуск стека

4. Ввести дополнительные параметры .

Дополнительные параметры:

- Имя стека;
- Таймаут создания стека;
- Определение разрешения на откат при сбое процесса создания или обновления;
- Пароль для пользователя с учетной записью Admin (требуется для выполнения операций в жизненном цикле стека);
- Дополнительные переменные, определяемые шаблоном, например `NetID`.

5. После успешного создания стека осуществится переход на страницу топологии данного стека .



Страница топологии стека

6.3 Примеры шаблонов

Образец самого простого шаблона может содержать только определение ресурсов с использованием только предопределенных свойств (вместе с обязательным тегом версии шаблона Heat).

Например, приведенный ниже шаблон можно использовать для простого развертывания одного *вычислительного экземпляра*.

6.3.1 Минимальный шаблон создания ВМ

```
heat_template_version: 2015-04-30

description: Simple template to deploy a single compute instance

resources:
my_instance:
type: OS::Nova::Server
properties:
key_name: my_key
image: cirros
```

```
flavor: m1.small
networks:
- network: external
```

6.3.2 Минимальный шаблон ВМ с использованием user-data

```
heat_template_version: 2015-04-30
description: Simple template to deploy a single compute instance
resources:
my_instance:
type: OS::Nova::Server
properties:
key_name: my_key
image: cirros
flavor: m1.small
networks:
- network: external
user_data: |
#!/bin/sh
echo "Hello, World!"
user_data_format: RAW
```

6.3.3 Создание ВМ с Cinder диском и передачей user-data

```
heat_template_version: 2013-05-23
description: An example Heat Orchestration Template (HOT).
parameters:
key_name:
type: string
description: Name of a KeyPair to enable SSH access to the instance
default: key1
instance_type:
type: string
description: Instance type for deploy
default: m1.medium
image_id:
type: string
description: Name or ID of the image to use.
default: cirros
inst_vol_size:
type: number
description: The size of the Cinder volume Instance 1
default: 20
instance_name:
type: string
description: Instance name
default: cirros1
resources:
Instance:
type: OS::Nova::Server
properties:
block_device_mapping: [{ device_name: "vda", volume_id : { get_resource: volume_root },
delete_on_termination : "true"}]
name: { get_param: instance_name }
flavor: { get_param: instance_type }
key_name: { get_param: key_name
networks:
- network: external
user_data: |
#cloud-config
timezone: Europe/Moscow
```

```
user_data_format: RAW
volume_root:
type: OS::Cinder::Volume
properties:
image: { get_param: image_id }
size: { get_param: inst_vol_size }
```

6.3.4 Создание ВМ с дополнительным Cinder-диском

Примечание

Cinder-диск монтируеется в `/tmp` и передаётся в user-data.

```
heat_template_version: 2013-05-23

description: An example Heat Orchestration Template (HOT).

parameters:
network:
type: string
description: Network
default: 204a9ff6-14b1-40ea-a23d-f90b47cc0244

key_name:
type: string
description: Name of a KeyPair to enable SSH access to the instance
default: key1

instance_type:
type: string
description: Instance type for deploy
default: m1.medium

image_id:
type: string
description: Name or ID of the image to use.
default: cirros

inst_vol_size:
type: number
description: The size of the Cinder volume Instance 1
default: 20

instance_name:
type: string
description: Instance name
default: cirros1

resources:
Instance:
type: OS::Nova::Server
properties:
block_device_mapping_v2:
- device_name: vda
boot_index: 0
volume_id : { get_resource: volume_root }
delete_on_termination : true
- device_name: vdb
boot_index: 1
volume_id : { get_resource: volume_tmp }
delete_on_termination : true
name: { get_param: instance_name }
flavor: { get_param: instance_type }
key_name: { get_param: key_name }
networks:
```

```
- network: { get_param: network }
user_data: |
#!/bin/sh
apt-get update
apt-get -y upgrade
mkfs.ext4 /dev/vdb
mount /dev/vdb /tmp
user_data_format: RAW
volume_root:
type: OS::Cinder::Volume
properties:
image: { get_param: image_id }
size: { get_param: inst_vol_size }

volume_tmp:
type: OS::Cinder::Volume
properties:
size: 10
```

6.4 Пример с разделением на разные файлы: шаблон, переменные и user-data

Команда для создания стека выглядит следующим образом:

```
openstack stack create -e env.yaml -t basic-stack.yaml basic-stack
```

Пример env.yaml:

```
parameters:
network_id: external
key_name: external
instance_type: 4x4x0
image_id: ubuntu-18.04-server-cloudimg-i386.img
inst_vol_size: 20
instance_name: default
```

Пример basic-stack.yaml:

```
.. literalincude:: basic-stack.yaml
```

Пример user-data.yaml:

```
#!/bin/bash
mkfs.ext4 /dev/vdb
mount /dev/vdb /tmp
apt-get update
apt-get -y upgrade
```

7 Резервное копирование и восстановление

Disaster & Recovery – аварийное резервное копирование и восстановление объектов виртуальной инфраструктуры.

Использование функциональности Freezer начинается с интерфейса пользователя, встроенного в модуль TIONIX.Dashboard, в виде панели управления, состоящей из разделов/подразделов и вкладок .

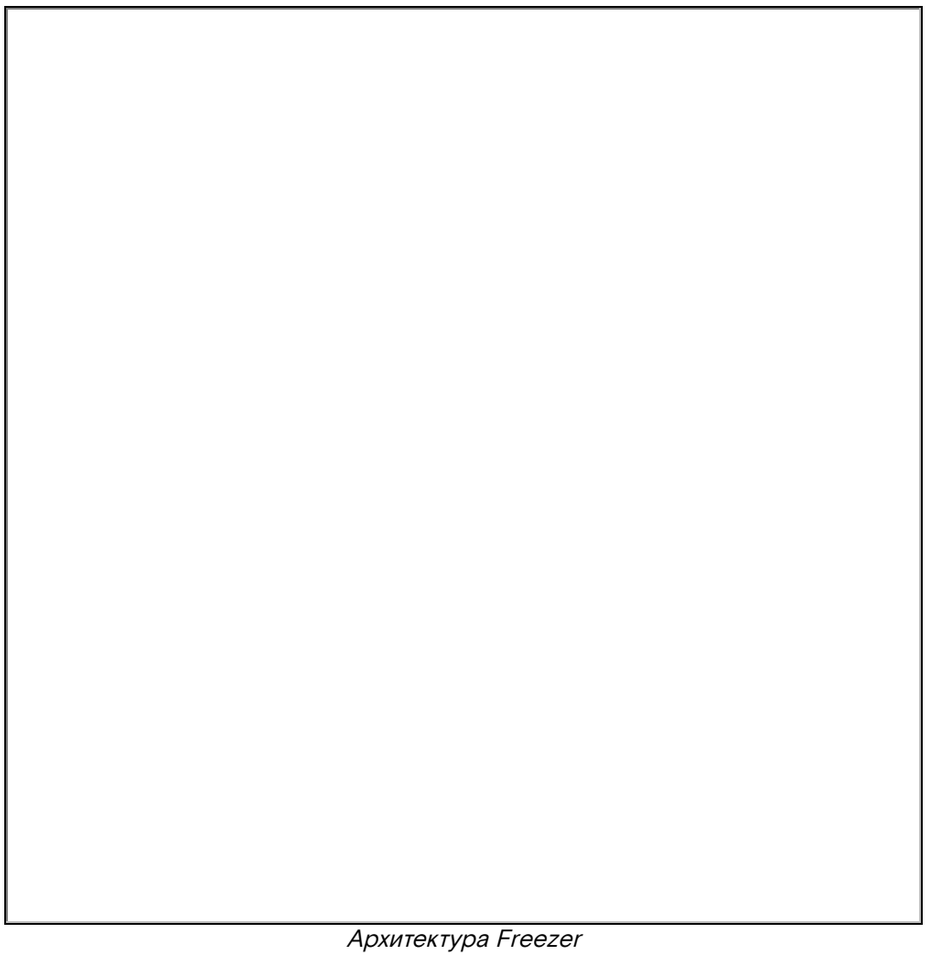
Процесс бэкапа состоит из действий (actions) и заданий (jobs), которые способны объединить несколько действий (например, нескольких ВМ) в одну запланированную работу, помогая компоновать действия – в соответствии с их заданиями . Поэтому, прежде всего необходимо выполнить создание действий, а затем уже можно приступать к формированию заданий. Действия бывают двух типов: **backup** и **restore** (резервное копирование и восстановление).

Исходные требования к подготовке инфраструктуры ОП:

- служба образов Glance настроена на использование бэкэнда Cinder (интеграция);
- настроенное объектное хранилище (Swift, NFS или FTP-хранилище);
- сконфигурированная и запущенная служба Freezer API на управляющих узлах;
- настроенное окружение для авторизации в Keystone.

7.1 Архитектура Freezer

Freezer – инструмент подсистемы резервного копирования (далее – РК), интегрированный в инфраструктуру ПО Базис.Cloud. Он состоит из нескольких функциональных компонентов (`Freezer_Arch`).



Архитектура Freezer

Функцией службы `Freezer API` является создание и обработка информации об элементах в одном из выбранных бэкендов (Elasticsearch или SQLAlchemy). Также, планировщик `Freezer Scheduler` использует `Freezer API` для получения информации о заданиях, которые дожны быть запланированы для выполнения (осуществляется с помощью агента).

`Freezer Agent` – основной «движок» подсистемы резервного копирования; он выполняет задания по резервному копированию данных, которые получает либо от планировщика заданий (Freezer Scheduler), либо напрямую – от утилиты **freezer-agent** (из командной строки).

Планировщик заданий выполняет обращение к службе Freezer API с определенной периодичностью. Основная его цель – сбор, обновление, удаление данных о запланированных задачах, а также передача их на исполнение – в «движок» подсистемы РК (Freezer Agent).

7.2 Подготовка Freezer к использованию

Для Freezer могут быть использованы два способа подготовки СУБД/БД к использованию компонентами Freezer:

- вариант 1: настройка Elasticsearch (желательно использовать);
- вариант 2: настройка MariaDb (не рекомендуется к использованию).

Далее, после настройки СУБД/БД, следует соблюдать порядок установки компонентов Freezer: сначала выполняется установка, настройка и проверка доступности Freezer API; затем устанавливаются планировщик заданий и агент (Freezer Sheduler/Agent); наконец, устанавливается `Freezer Web UI`.

Компоненты Freezer Scheduler/Agent работают в связке друг с другом и отвечают за создание бэкапов непосредственно на узлах.

Freezer Web UI – надстройка интерфейса управления (OpenStack Horizon), позволяющая выполнять администрирование службы Backup aaS с помощью web-интерфейса. Надстройка совместима с TIONIX.Dashboard.

Важно

Freezer Scheduler и Freezer Agent должны быть установлены вместе – на один управляющий узел.

7.2.1 Вариант 1: настройка Elasticsearch (рекомендовано)

Подключитесь к УУ с правами суперпользователя (root@controller) и выполните установку необходимых компонентов.

Сначала установите программные пакеты:

```
dnf install java elasticsearch
```

Затем отредактируйте файл конфигурации `/etc/elasticsearch/elasticsearch.yml` в соотвествии с примерами ниже:

Кластерная реализация Elasticsearch

`/etc/elasticsearch/elasticsearch.yml`:

```
cluster.name: es-tionix-cluster
cluster.initial_master_nodes: ["control1", "control2", "control3"]
node.name: control1
node.master: true
node.data: true
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: control1
http.port: 9200
discovery.zen.ping.unicast.hosts: ["control1", "control2", "control3"]
discovery.zen.minimum_master_nodes: 2
node.max_local_storage_nodes: 1
```

Примечание

Данная конфигурация должна быть распространена на все ноды кластера. Обратите внимание на параметры node.name[5] и network.host, которые должны быть изменены в соотвествии с именем ноды, на которой они указываются.

Одиночная реализация Elasticsearch

`/etc/elasticsearch/elasticsearch.yml`:

```
node.name: control1
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: control1
http.port: 9200
```

Выполните запуск системной службы на всех УУ отказоустойчивого кластера:

```
systemctl enable elasticsearch
systemctl start elasticsearch
```

Проверьте статус кластера с помощью команды-запроса:

```
curl -XGET 'http://control1:9200/_cluster/state?pretty'
```

Ответ должен содержать JSON-файл, в котором описано состояние кластера, в том числе – список нод, участвующих в нём.

Для проверки одиночной реализации используйте команду-запрос:

```
curl -XGET 'http://control1:9200/?pretty'
```

Добавьте в файл конфигурации `/etc/haproxy/haproxy.cfg` следующие строки:

```
frontend elastic-cluster
bind <VIP_ADDR>:9200
mode http
```

5 http://node.name

```
default_backend elastic-cluster_backend

backend elastic-cluster_backend
mode http
option forwardfor
balance source
option httpclose
server control1 <NODE_IP_ADDR>:9200 weight 1 check inter 1000 rise 5 fall 1
server control2 <NODE_IP_ADDR>:9200 weight 1 check inter 1000 rise 5 fall 1
server control3 <NODE_IP_ADDR>:9200 weight 1 check inter 1000 rise 5 fall 1
```

Обратите внимание на значения VIP_ADDR и NODE_IP_ADDR. Их нужно указать в соотвествии с сетевой адресацией, которая используется в платформе.

Перезапустите сервис HAProxy:

```
pcs resource restart p_haproxy
```

В случае одиночной реализации (вне кластера Pacemaker):

```
systemctl restart haproxy
```

Проверьте доступность командой:

```
curl -XGET 'http://<VIP_ADDR>:9200/_cluster/state?pretty'
```

Создайте ресурс в Pacemaker:

```
pcs resource create elasticsearch systemd:elasticsearch op monitor timeout="15s"
interval="15s" clone
```

7.2.2 Вариант 2: настройка MariaDb (не рекомендуется)

Чтобы создать базу данных, выполните следующие действия:

1. Используйте клиент СУБД для подключения к серверу БД от имени пользователя root (root@controller).

```
mysql -u root -p

или

mariadb -u root -p
```

Появится интерактивный запрос клиента на ввод (SQL-)команд

MariaDB (none)>

2. Создайте базу данных с именем **freezer**, выполнив SQL-запрос:

```
CREATE DATABASE freezer;
```

3. Предоставьте доступ к БД Freezer, выполнив SQL-запрос:

```
GRANT ALL PRIVILEGES ON freezer.* TO 'freezer'@'localhost' IDENTIFIED BY
'FREEZER_DBPASS';
GRANT ALL PRIVILEGES ON freezer.* TO 'freezer'@'%' IDENTIFIED BY 'FREEZER_DBPASS';
```

В приведенном выше примере SQL-запроса следует заменить „FREEZER_DBPASS“ соответствующим паролем на доступ к БД.

4. Настройте административное окружение для доступа к ресурсам Openstack с помощью CLI-команд и выполните настройку сервиса.

Подключитесь к УУ с правами суперпользователя:

```
ssh root@controller
```

Выполните нижеописанные действия (на УУ).

- Создайте пользователя freezer:

```
openstack user create --domain default --password-promt freezer
```

После выполнения команды необходимо задать соответствующий пароль для пользователя в интерактивном режиме.

- Предоставьте роль администратора пользователю freezer:

```
openstack role add --project service --user freezer admin
```

Обратите внимание, что команда не предоставляет вывода какой-либо информации при корректном выполнении.

Примечание

Впоследствии, аналогичной командой необходимо давать пользователю freezer доступ к тем проектам, для которых будет выполняться бэкап.

- Создайте службу OpenStack с названием `freezer`:

```
openstack service create --name freezer --description "Freezer Backup Service" backup
```

- Создайте новые эндпоинты для сервиса резервного копирования:

```
openstack endpoint create backup --region RegionOne admin http://http://controller:9090
openstack endpoint create backup --region RegionOne internal http://http://
controller:9090
openstack endpoint create backup --region RegionOne public http://http://controller:9090
```

7.2.3 Установка и настройка Freezer API

Подключитесь к УУ (root@controller) и установите пакет (с зависимостями):

```
dnf install freezer-api
```

Примечание

Вместе с пакетом будут также установлены зависимости, необходимые для работы Freezer API. Актуальные версии пакетов скачиваются из репозитория `tionix-extras`, который был настроен ранее (для установки модулей TIONIX).

Внимание

После установки и настройки Freezer API рекомендуется проверить доступность службы по заданным эндпоинтам.

Настройка конфигурации при использовании бэкэнда Elasticsearch

В случае настроенного бэкенда в Elasticsearch укажите в конфигурационном файле (`/etc/freezer/freezer-api.conf`) следующие параметры:

```
[storage]
backend = elasticsearch
driver = elasticsearch
```

```
[elasticsearch]
hosts = http://127.0.0.1:9200
number_of_replicas = 1
index = freezer
```

Настройка конфигурации при использовании бэкэнда MariaDb (опционально)

Если настроен бэкэнд MariaDB, то для доступа к БД в файле `/etc/freezer/freezer-api.conf` параметры доступа должны быть такими:

```
[storage]
backend = sqlalchemy
driver = sqlalchemy

[database]
connection = mysql+pymysql://freezer:FREEZER_DBPASS@controller/freezer
```

Внимание

Следует изменить „FREEZER_DBPASS“ на пароль для доступа к БД, который был задан ранее.

Настройка конфигурации авторизации

В секции [DEFAULT] конфигурационного файла (`freezer-api.conf`) укажите путь к файлу, в который будут записываться логи API:

```
[DEFAULT]
...
log_file = /var/log/freezer/freezer-api.log
```

Дополнительно, создайте или отредактируйте секцию [keystone_authtoken] в соответсвии с примером:

```
[keystone_authtoken]
...
www_authenticate_uri = http://controller:5000/v3
auth_url = http://controller:5000/v3
auth_type = password
project_domain_id = default
user_domain_id = default
project_name = service
username = freezer
password = <пароль_пользователя_FREEZER>
```

Примечание

Измените значение <пароль_пользователя_FREEZER> на соответствующее – пароль пользователя freezer, который был задан ранее.

Дополнительные параметры настройки конфигурации

Дополните файл конфигурации (`freezer-api.conf`) следующими разделами, если таковые отсутствуют:

```
[oslo_middleware]
enable_proxy_headers_parsing = True

[paste_deploy]
config_file = /etc/freezer/freezer-paste.ini
```

Инициализация БД и запуск службы

Для первичной инициализации БД выполните следующую команду:

```
freezer-manage db sync
```

Выполните ряд команд, необходимых для запуска системной службы `freezer-api.service`:

```
systemctl daemon-reload
systemctl enable freezer-api.service
systemctl start freezer-api.service
```

7.2.4 Проверка доступности API

Подключитесь к УУ (root@controller) и выполните выпуск токена для авторизации в Keystone. Выполните команду:

```
openstack token issue
```

Будет выведен ответ в виде таблицы с информацией о новом токене. Скопируйте в буфер обмена значение параметра `id`. Пример ответа:

expires,	«2022-07-25T10:08:14+0000»
id,	«длинная_алфавитно-цифровая_последовательность»
project_id,	«fdcc2b89b84145af9ae5d90bd4cec347»
user_id,	«fabd301c08bf45588bf4161d907cf209»

Выполните экспорт переменной, содержащей `id` токена (скопированного из предыдущего вывода):

```
export my_token=<длинная_алфавитно-цифровая_последовательность>
```

Выполните HTTP-запрос на адрес и порт, на котором должен работать сервис `Freezer API`:

```
curl -X GET -H "X-Auth-Token: $my_token" http://controller:9090/v2 | jq
```

После выполнения команды ответ от API приходит в формате JSON.

7.2.5 Установка и настройка Scheduler/Agent

Подключитесь к УУ (**ssh root@controller**) и выполните установку необходимых компонент:

```
dnf install freezer python-freezerclient
```

Создайте или отредактируйте файл конфигурации планировщика заданий – `/etc/freezer/freezer-scheduler.conf` – в соответствии с примером (см. ниже). В секции [DEFAULT] необходимо настроить авторизацию службы Freezer:

```
[DEFAULT]
...
os_auth_url = http://controller:5000/v3
os_username = freezer
os_password = <пароль_служебного_пользователя>
os_project_name = service
```

```
os_project_domain_name = default
os_user_domain_name = default
os_backup_endpoint = http://controller:9090/
```

Авторизация выполняется путём выполнения REST API запроса к Keystone.

> **Внимание**
>
> Переменная „os_password“ должна содержать пароль, который был указан при создании служебного пользователя freezer.

Дополните секцию [DEFAULT] следующими строками:

```
jobs_dir = /etc/freezer/scheduler/conf.d
log_file = /var/log/freezer/freezer-scheduler.log
```

Выполните запуск systemd-сервиса (`freezer-scheduler.service`):

```
systemctl daemon-reload
systemctl enable freezer-scheduler.service
systemctl start freezer-scheduler.service
```

Убедитесь, что Freezer Scheduler зарегистрировал ноду в Freezer API с помощью CLI-клиента freezerclient и данной команды:

```
freezer client --list
```

Пример ответа (root@controller):

Client ID	uuid	hostname	description
хэш-код_controller	5cac43214ec64991842b61966c6067a0	controller	None

7.2.6 Установка Freezer Web UI

Установка необходимых компонентов осуществляется после подключения к УУ **ssh root@controller**):

```
dnf install freezer-web-ui
```

Перейдите в директорию с установленным приложением и выполните копирование файла активации панели в файловую структуру Horizon (Openstack Dashboard):

```
cp /usr/lib/python3.6/site-packages/disaster_recovery/enabled/_5050_freezer.py /usr/
share/openstack-dashboard/openstack_dashboard/enabled/
```

Выполните перезапуск веб-сервера:

```
systemctl restart httpd
```

Выполните вход интерфейс управления, и убедитесь в появлении панели Freezer (Аварийное восстановление, Disater&Recovery).

7.3 Проверка работоспособности подсистемы резервного копирования

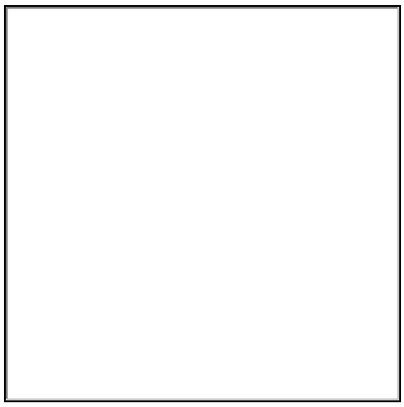
Выполните вход в интерфейс управления (Dashboard) и перейдите:

Аварийное восстановление (локализованный на русский язык)

или

Disaster & Recovery (английский язык интерфейса)

Используя интерфейс пользователя (`Freezer_UI`), настройте задание, содержащее ряд действий (создание РК, восстановление и удаление).



Интерфейс пользователя подсистемы РК (Freezer UI)

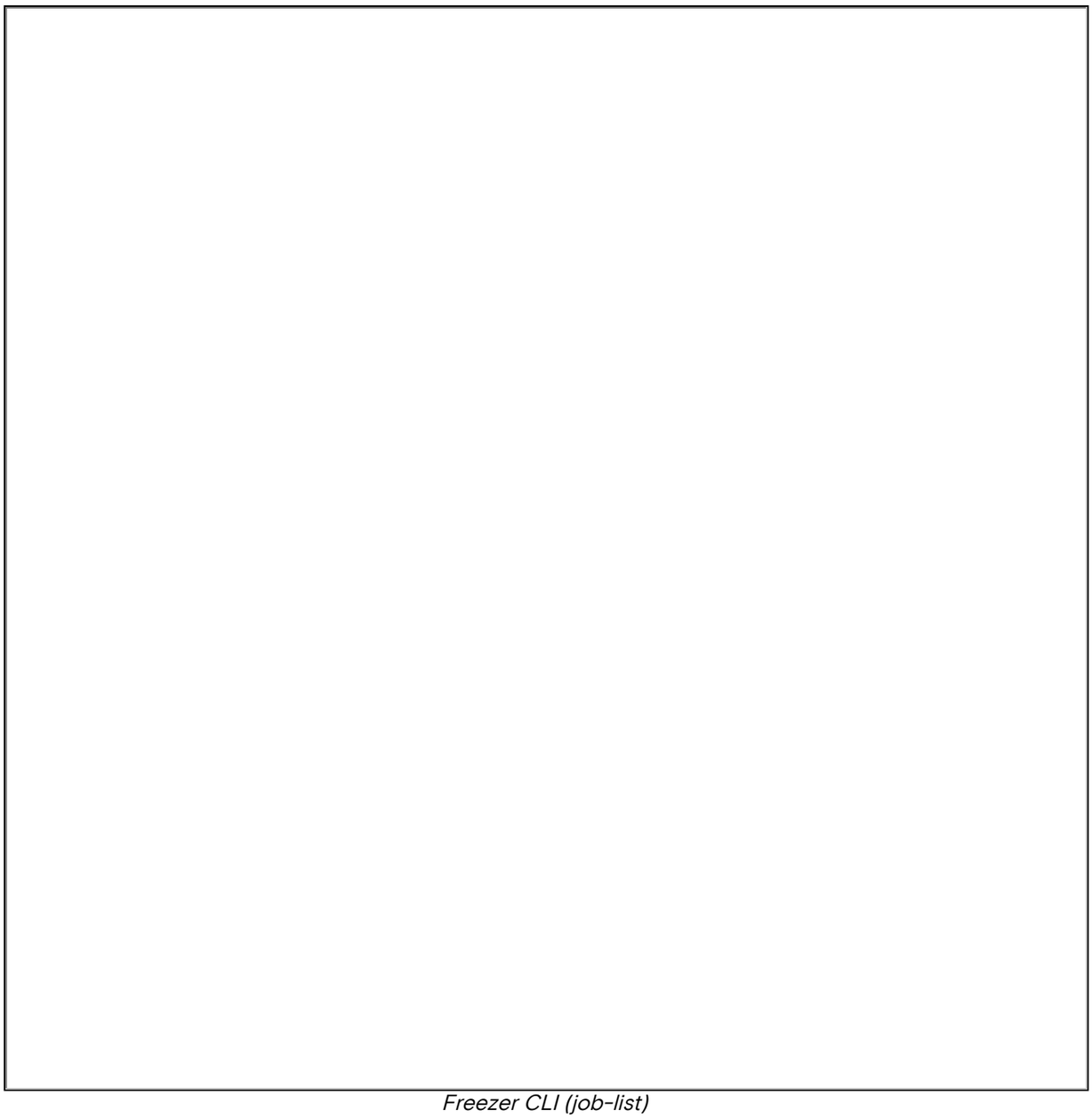
Просмотрите информацию о задании .

7.4 Использование Freezer

Для получения списка заданий, а также информации об их статусе выполните команду:

```
freezer job-list
```

Вывод команды показан ниже на `Freezer_JobList`.



Freezer CLI (job-list)

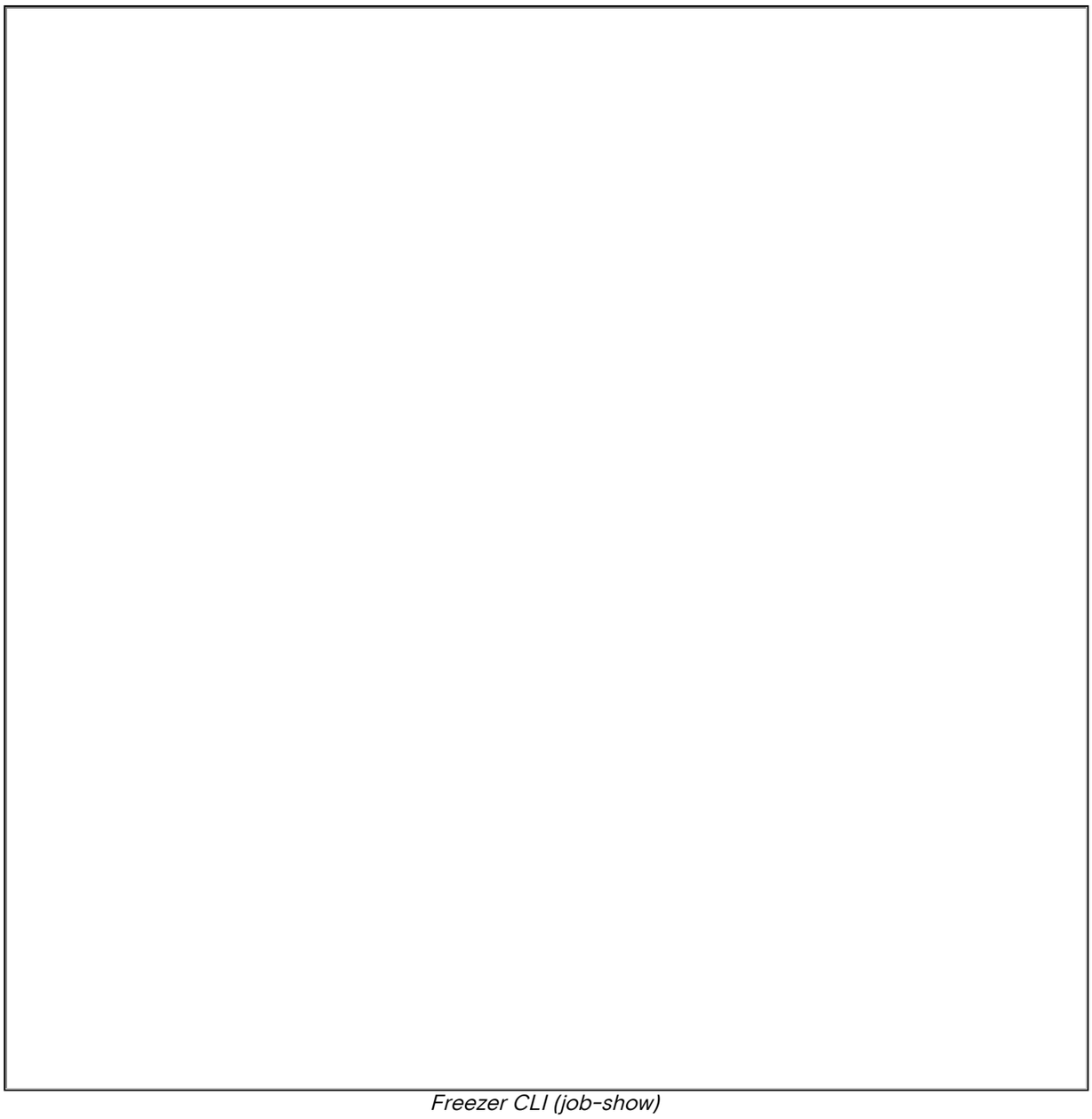
Ангалогично, вывод списка заданий может быть выдолнен в соответствующем разделе интерфейса управления (Dashboard).

7.4.1 Вывод информации о задании

Для уточнения информации о конкретном задании выполните команду:

```
freezer job-show <JOB_ID>
```

Пример вывода показан на `Freezer_JobShow`.



Freezer CLI (job-show)

7.4.2 Формирование задания с помощью Freezer CLI

Для создания элементов производится посредством импорта файлов формата JSON, описывающих элемент.

Создание задания для бэкапа Nova-машин с помощью клиента выполняется командой:

```
freezer job-create --file nova-job.json --client controller
```

Пример файла задания:

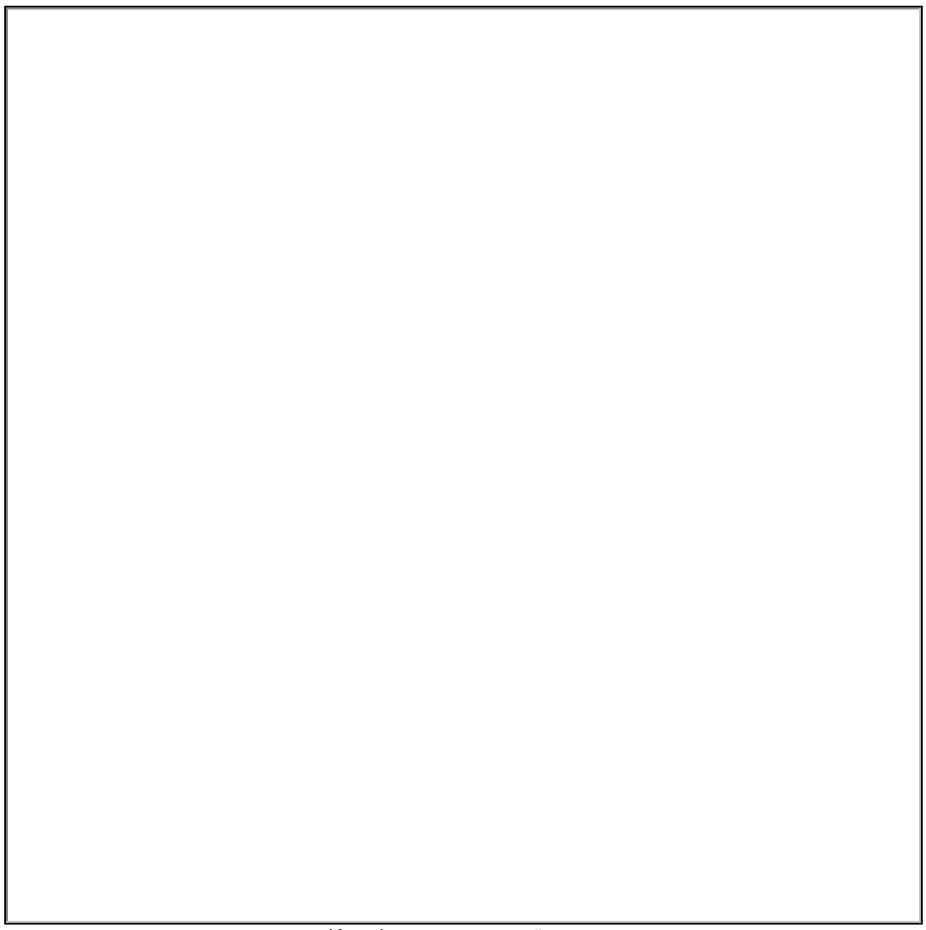
```
{
"description":"nova-backup", ## имя job
"job_actions":[
{
"max_retries":5,
"max_retries_interval":6,
"freezer_action":{
"action":"backup",
"backup_name":"nova-backup-251f85e3-80ad-48b4-a49b-0d0e0fe70d91", ### имя бэкапа состоит из ID машины, которую надо забэкапить
"compression":"gzip",
"container":"/tmp/freezer",
"engine_name":"nova",
"mode":"nova",
```

```
"no_incremental":"True",
"nova_inst_id":"251f85e3-80ad-48b4-a49b-0d0e0fe70d91", ### ID машины для бэкапа
"storage":"local"
}
}
]
}
```

7.4.3 Создание действия по резервному копированию ВМ

Перейдите к разделу панели управления – **Аварийное восстановление** (Disaster Recovery), раскройте вкладку **Резервное копирование и восстановление** и выберите раздел **Действия** (Actions).

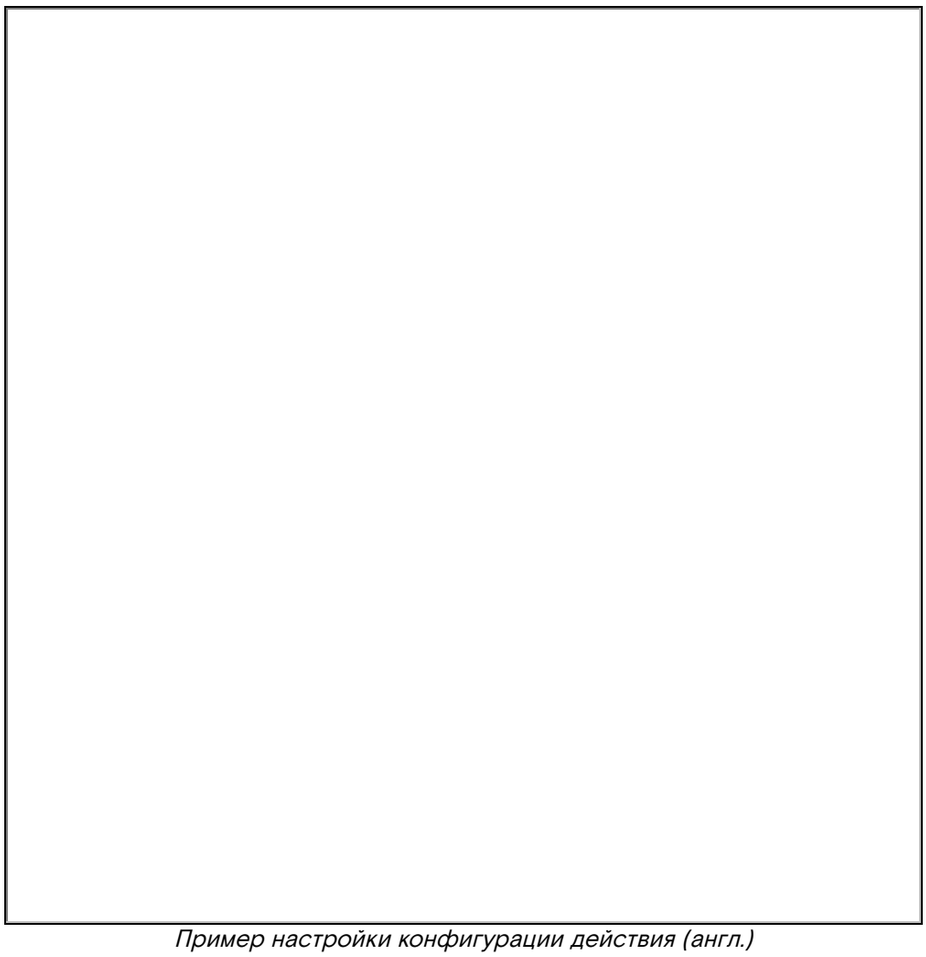
Будет открыта страница со списком действий и кнопок управления ими. Нажмите кнопку [Создать действие] (Create Action) – откроется диалог (`Action_config`).



Конфигурация действия

Заполните параметры конфигурации действия (обязательные поля **Название действия**, **Действие**, **Хранилище**, **Название движка**, **Исходный файл/директория**).

Пример настройки действия по резервному копированию ВМ:

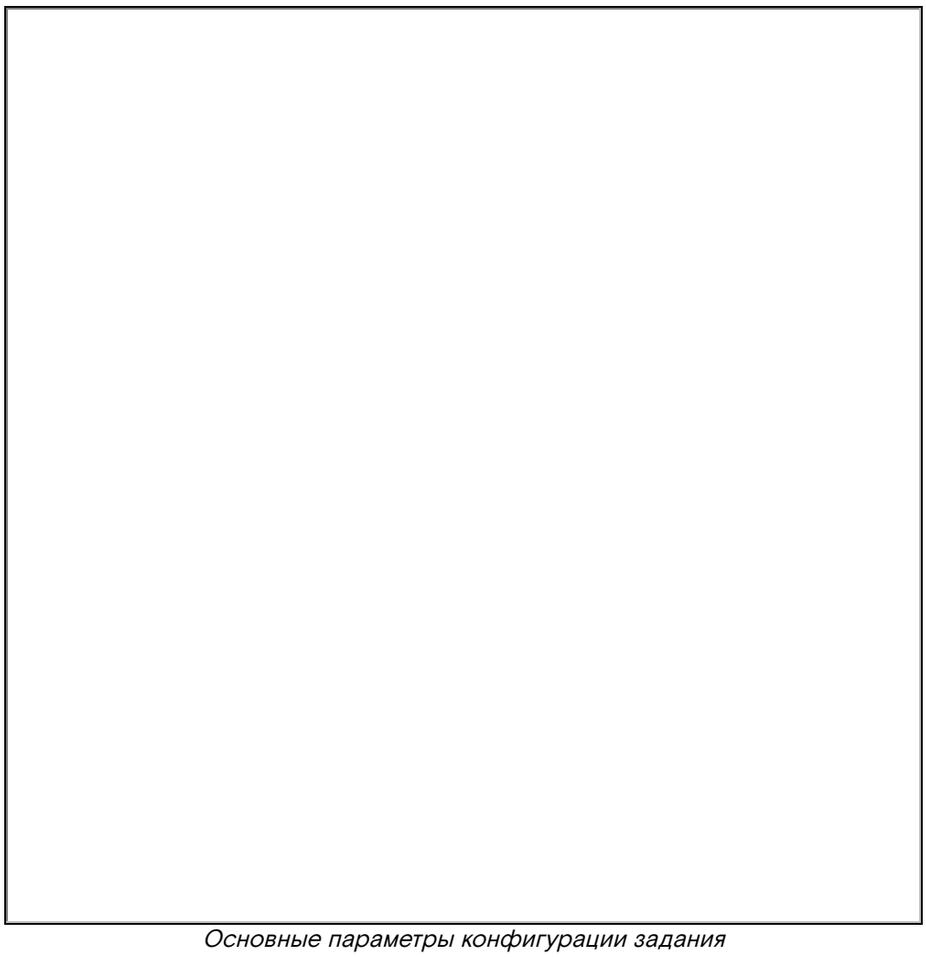


Пример настройки конфигурации действия (англ.)

7.4.4 Создание задания для резервного копирования ВМ

Перейдите ко вкладке **Задания** (Jobs) раздела Аварийное восстановление (Disaster Recovery). Отобразится страница со списком всех имеющихся заданий (см. `Freezer_JobList`).

Нажмите кнопку [Создать задание] (Create Job). Откроется диалог настройки конфигурации задания (`Job_config`).



Основные параметры конфигурации задания

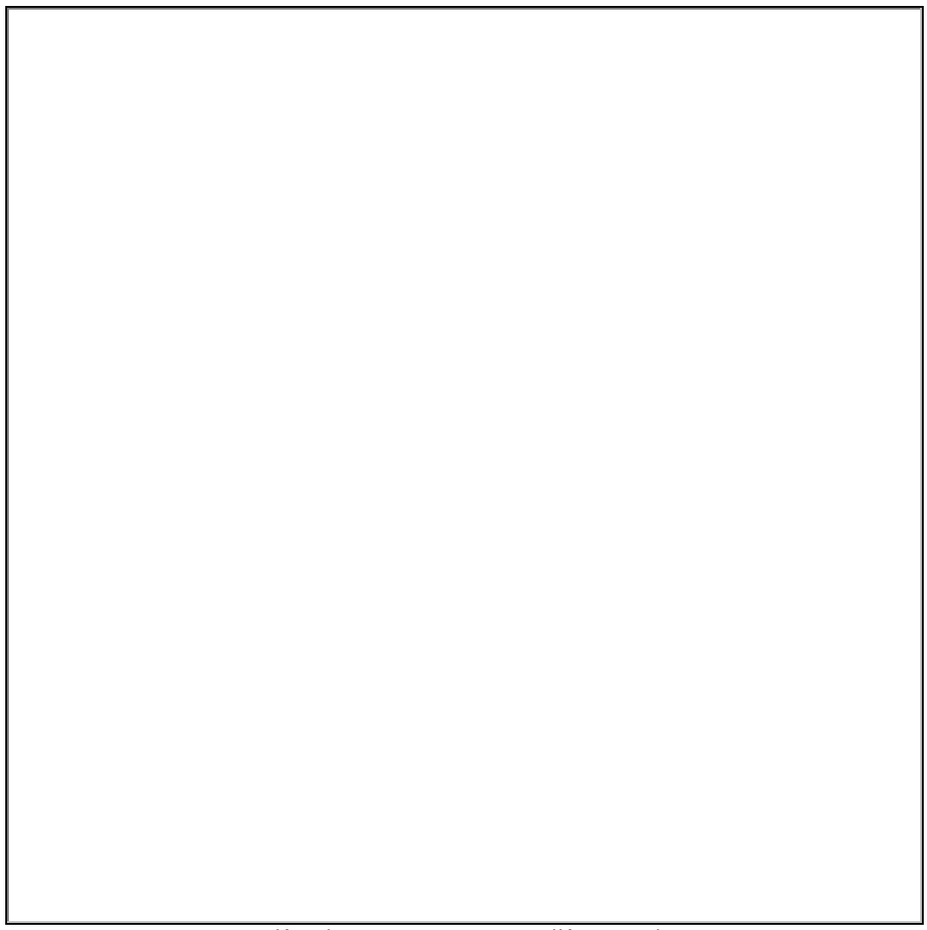
Заполните параметры конфигурации (обязательное поле – **Название задания**, остальные опционально – **Время начала**, **Ед.изм. интервала**, **Дата и время окончания**).

В следующей вкладке необходимо выбрать клиента (из списка доступных), на котором будет выполняться задание.

Примечание

Клиенты регистрируются в API с помощью `freezer-scheduler`.

Перетащите ранее созданные действия слева направо (Selected Actions), для выполнения их в рамках задания (`Job_config_client`).



Конфигурация задания (Клиенты)

8 Миграция (перенос) виртуальных машин

Процессы миграции виртуальных машин относятся к управлению ресурсами облачной инфраструктуры, в частности – вычислительными.

Принято различать несколько видов миграции:

- живая миграция;
- холодная миграция ;
- экстренная миграция .

Живая миграция – перенос виртуальной машины, находящейся в состоянии «Активна», на определенный (пользователем) вычислительный узел. Фактически, в процессе выполнения живой миграции в ОП BASIS виртуальная машина переносится с одного физического сервера на другой (с ноды на ноду), без прекращения ее работы и остановки сервисов.

Особенности живой миграции:

- перенос виртуальной машины происходит без её выключения;
- на время переноса виртуальная машина приостанавливается (статус suspend);
- возможет выбор целевого вычислительного узла (для переноса ВМ);
- миграция окончится ошибкой, если на целевом ВУ недостаточно ресурсов.

Холодная миграция – процесс миграции (гостевой) системы, которая была отключена или временно отключена. Это отличается от практики живой миграции (горячей миграции), когда миграция происходит без останова работы .

Механизм миграции может регулироваться библиотекой `libvirt`, например, с помощью опции „libvirt_cpu_mode“. Эта опция отвечает за то, чтобы модели процессоров исходного хоста и хоста назначения были совместимы, что также влияет на *выбор хоста назначения*. Подробнее – см. по ссылке: http://lists.openstack.org/pipermail/openstack-dev/2016-January/083275.html («Deprecating the live_migration_flag and block_migration_flag config options»)

В `libvirt` есть и другие опции, влияющие на *автоматический выбор хоста*. Отдельные параметры миграции прописаны в файле `/etc/libvirt/qemu.conf`. Такие как, диапазон портов входящих миграций, время отклика и количество обращений во время миграций peer-to-peer (от узла к узлу).

Примечание

Инструкции по выполнению операций миграции вычислительного узла изложены в документе `Руководство администратора ПО Базис.Cloud`.

8.1 Холодная миграция и автоэвакуация

Под *эвакуацией* следует понимать процесс ручного или автоматического переноса виртуальной машины (принудительной миграции), в случае выявления проблем на том вычислительном узле, ресурсы

которого предоставлены гипервизором . Перенос осуществляется на другие доступные вычислительные узлы, но только при наличии свободных ресурсов .

Обычная эвакуация подразумевает плановый перенос ВМ с приостановкой её работы . Она также может называться *холодной миграцией* и может выполняться как в штатном так и сервисном режимах работы ОП .

Экстренная (автоматическая) эвакуация происходит автоматически, если разрешена; при этом используется гипервизор, назначенный резервным и имеющий достаточно свободных ресурсов . Кроме того, необходимо настроить должным образом и привязать к гипервизору устройство управления питанием , а на контроллер(ы) установить соответствующий пакет поддержки (IMPI) или другие утилиты (поддержки протоколов обмена), соответственно типу устройств – модулей, установленных в серверные компьютеры.

8.1.1 Эвакуация (вручную)

Для ручной миграции виртуальной машины необходимо подключиться к вычислительному узлу, которым обслуживается эта машина. Перейдите в директорию, в которой расположены файлы образов (виртуальных дисков) – `/var/lib/nova/instances/$UUID`.

Остановите эвакуируемую виртуальную машину. Выполните команду:

```
nova stop $VM_UUID
```

Убедитесь, достаточное ли количество свободных ресурсов имеется на вычислительном узле, на который будет осуществляться эвакуация (перенос ВМ). Также необходимо убедится, что на том и другом гипервизоре присутствует образ, из которого была создана машина. Проверить это можно следующим образом:

1. Перейдите в каталог с виртуальной машиной:

```
cd /var/lib/nova-compute/instances/UUID/
```

2. Выполните команду:

```
qemu-img info disk
```

Пример вывода данной команды:

```
image: disk

file format: qcow2
virtual size: 32G (34359738368 bytes)
disk size: 1.3G
cluster_size: 65536
backing file: /var/lib/nova/instances/_base/d00[...]61
Format specific information:
compat: 1.1
lazy refcounts: false
```

В данном выводе нас интересует строка „backing file“ с именем диска.

3. Синхронизируйте образы на исходном и целевом гипервизорах:

```
rsync -r --progress /var/lib/nova/instances/_base/d00[...]61 -e ssh $new_node:/var/lib/
nova/instances/_base/d00[...]61
```

4. Выставите необходимые права на скопированный файл:

```
chown nova:nova /var/lib/nova/instances/_base/d00[...]61
```

Необходимо убедится, что на узле, куда будет осуществляться перенос ВМ – в директории `/var/lib/nova-compute/instances/$UUID` – имеется эвакуируемая виртуальная машина. Если ее (ВМ) нет, то необходимо скопировать ее командой:

```
rsync -r --progress $VM_UUID -e ssh $node_name:/var/lib/nova-compute/instances/
```

5. Установите необходимые разрешения для скопированных файлов:

```
chown nova:nova /var/lib/nova-compute/instances/$VM_UUID
chown nova:nova /var/lib/nova-compute/instances/$VM_UUID/disk.info
chown nova:nova /var/lib/nova-compute/instances/libvirt.xml

chown libvirt:kvm /var/lib/nova-compute/instances/$VM_UUID/console.log
chown libvirt:kvm /var/lib/nova-compute/instances/$VM_UUID/disk
chown libvirt:kvm /var/lib/nova-compute/instances/$VM_UUID/disk.config
```

Далее, необходимо внести изменения в базу данных. Выполните на УУ, с помощью клиента MySQL/MariaDb, команду:

```
mysql nova

update instances set node='$new_node', host='$new_node' where uuid='$VM_UUID';
```

Проверьте, что виртуальная машина успешно перемещена на другой гипервизор и запустите её:

```
nova show;
nova start $VM_UUID
```

8.1.2 Автоэвакуация

Модуль `TIONIX.NodeControl` при использовании функции автоэвакуации может воспользоваться возможностями управления питанием вычислительных узлов для их временного вывода из эксплуатации. Он может регистрировать *устройства питания*, которые затем могут быть привязаны к конкретному вычислительному узлу.

Механизм авто-эвакуации использует функционал:

- службы NodeControl, которая функционирует на УУ (в кластере управления);
- средств поддержки протоколов управления питанием (IPMI, SSH и т.д.).

> **Внимание**
>
> Перед включением функции **автоэвакуации** убедитесь, что все службы и средства поддержки протоколов управления питанием, требуемые для нормальной работы, настроены и корректно работают.

Для нормальной работы механизма **автоэвакуации**, алгоритм которого изложен в разделе "Автоэвакуация", должны быть соблюдены следующие условия:

1. настроен `shared-storage` (СХД с Cinder тоже подходит);
2. настроен модуль `TIONIX.NodeControl`;
3. заведен доступ к узлам через сеть IPMI;
4. настроено хранилище проверки доступности (ХПД).

При выходе из строя гипервизора, функционирующего на ВУ, будет включаться запасной ВУ и вводиться один из доступных резервов . Если резервных гипервизоров не найдено, то замена не произойдет; об этом будет сообщено в лог-файле модуля TIONIX.NodeControl.

Функционал модуля `TIONIX.NodeControl` содержит *консольную утилиту* – **openstack** (Интерфейс командной строки (клиента) OpenStack), которая позволяет создать *хранилище*, используемое для проверки доступности ВУ и гипервизоров, управляемых при помощи службы Nova.

> **Внимание**
>
> Чтобы модуль `TIONIX.NodeControl` мог работать с хранилищем проверки доступности, сначала требуется создать **общее хранилище**.

8.2 Живая миграция

В процессе живой миграции участвуют два работающих гипервизора – исходный, из которого должен быть осуществлен перенос и целевой – на который будет производиться перенос .

Ниже изложена инструкция по выполнению живой миграции. Дополнительный материал об использовании миграции на уровне хранилища изложен в разделе "Миграция логического тома".

Дополнительная информация по живой (блочной) миграции доступна по ссылке: https://docs.openstack.org/nova/victoria/admin/live-migration-usage.html

Подробная диаграмма последовательности вызовов доступна на сайте официальной документации OpenStack Nova.

8.2.1 Проверка связанности

Прежде всего, следует начать проверки IP-связанности вычислительных узлов, на которых функционируют исходный и целевой гипервизоры. Подключитесь к ВУ с исходным гипервизором и выполните команду:

```
# ping compute-opt
```

Далее, необходимо запустить ВМ и определить, на каком из доступных ВУ она работает. Выполните однотипные команды:

```
nova hypervisor-servers compute.test.local

+----+------+---------------+---------------------+
| ID | Name | Hypervisor ID | Hypervisor Hostname |
+----+------+---------------+---------------------+
+----+------+---------------+---------------------+

nova hypervisor-servers compute-opt.test.local

+------------------+-------------------+---------------+------------------------+
| ID | Name | Hypervisor ID | Hypervisor Hostname |
+------------------+-------------------+---------------+------------------------+
| 9e319540-9a67-.. | instance-000000df | 2 | compute-opt.test.local |
+------------------+-------------------+---------------+------------------------+
```

Из вывода команд очевидно, что виртуальная машина работает на узле `compute-opt`.

Дальше следует определить, какой flavor использован для этой ВМ. Выполните команду:

```
nova show 9e319540-9a67-4563-9aad-132c64faa1b1 | grep flavor
| flavor | m2.tiny (98cb36fb-3541-415b-9835-bfc7e73546e3) |
```

8.2.2 Проверка доступности ресурсов

Проверьте, достаточно ли ресурсов на том узле, который содержит целевой гипервизор, выполнив команду:

```
$ nova host-describe compute.test.local

+--------------------+------------+-------+-----------+-----------+
| HOST | PROJECT | cpu | memory_mb | disk_gb |
+--------------------+------------+-------+-----------+-----------+
| compute.test.local | (total) | 4 | 3952 | 49 |
| compute.test.local | (used_now) | 0 | 512 | 0 |
| compute.test.local | (used_max) | 0 | 0 | 0 |
+--------------------+------------+-------+-----------+-----------+
```

Если разница между общим количестом и использованием каждого из ресурсов (cpu, memory_mb, disk_gb) позволяет размещение ВМ, обслуживаемой исходным гипервизором, то можно продолжить.

8.2.3 Подготовка конфигурации (гипервизора)

Однако, прежде чем запускать на выполнение команду миграции, необходимо разрешить демону *libvirtd* «слушать» входящие подключения (по сети). На том и другом гипервизорах необходимо добавить опцию,

отвечающую за строку запуска демона, в файл конфигурации (системной службы) `/etc/sysconfig/libvirtd`: LIBVIRTD_ARGS=»–listen»

Далее, в конфигурационном файле `/etc/libvirt/libvirtd.conf` необходимо разрешить выполнение подключений без аутентификации и шифрования:

```
listen_tls = 0
listen_tcp = 1
auth_tcp = "none"
```

Альтернативой последней настройке может быть использование сертификатов или Kerberos.

Выполните рестарт системной службы libvirtd (перезапуск гипервизора) на всех вычислительных узлах, для вступления в силу изменений в конфигурации:

```
systemctl restart libvirtd
```

Последний шаг настройки – исправление флагов миграции. Делается это также на всех вычислительных узлах. Выполните (с правами root) команду:

```
crudini --set /etc/nova/nova.conf DEFAULT block_migration_flag
VIR_MIGRATE_UNDEFINE_SOURCE,VIR_MIGRATE_PEER2PEER,VIR_MIGRATE_LIVE
```

Изменение во флагах (службы Nova) необходимо, поскольку флаги по умолчанию включают в себя TUNELLED, который не работает с обновленным кодом NBD (Network Block Device) в QEMU.

Для применения изменений необходимо перезапустить службу вычислений (Nova Compute). Выполните команнду:

```
systemctl restart openstack-nova-compute.service
```

8.2.4 Запуск процесса миграции

Выполните подготовку окружения (пройдите аутентификацию):

```
$ source keystonerc_admin
```

Теперь можно выполнить команду живой миграции, за исполнение которой отвечает служба Nova. При этом обязательно указать опцию --block-migrate, которая определяет миграцию без общего дискового хранилища:

```
$ nova live-migration --block-migrate 9e319540-9a67-4563-9aad-132c64faa1b1
compute.test.local
```

где

`compute.test.local` – ВУ, на котором функционирует целевой гипервизор.

Во время процесса миграции на узле-источнике можно следить за ходом выполнения, при помощи команды **virsh domjobinfo**:

virsh domjobinfo instance-000000e3

8.2.5 Проверка результата миграции

При помощи команды **nova show** проверяется, на каком узле работает ВМ. Используйте эту команду до и после миграции, чтобы уточнить результат выполнения операции (живой) миграции:

```
$ nova show <vm_UUID> | grep hypervisor

| OS-EXT-SRV-ATTR:hypervisor_hostname | compute-opt.test.local |
```

```
$ nova live-migration --block-migrate <vm_UUID>

$ nova show <vm_UUID> | grep hypervisor

| OS-EXT-SRV-ATTR:hypervisor_hostname | compute.test.local |
```

Различие в принадлежности ВМ с идентификатором *vm_UUID* к гипервизору говорит о том, что миграция выполнена успешно.

В журнале должно появиться сообщение следующего вида:

> <Дата_и_Время> 887 INFO nova.compute.manager [req-a09dc90b-8b69-4e15-8dee-f96ac7d197c3 6310882678344e8183f2d7e886088293 8cc74ebe8da94fe0a7ac6cf54f31b420 - - -] [instance: <vm_UUID>] Migrating instance to compute.test.local finished successfully.

8.2.6 Нештатные ситуации

В случае использования ОС CentOS и дистрибутива RDO живой миграции не будет видно, но будет получена ошибка, отраженная в журнале nova-compute:

> 2015-12-05 23:07:49.391 31600 ERROR nova.virt.libvirt.driver [req-4986043d-abf4-40c4-85d4-7c6b3d235986 6310882678344e8183f2d7e886088293 8cc74ebe8da94fe0a7ac6cf54f31b420 - - -] [instance: 7c505c55-69a8-493c-a119-cba65d58e3bb] Live Migration failure: internal error: unable to execute QEMU command „migrate“: this feature or command is not currently supported

Это происходит потому, что в CentOS пакет `qemu-kvm` собран без поддержки ряда функций, включая те, что необходимы для живой миграции без общего хранилища. Чтобы обойти проблему, необходимо заменить пакет (qemu-kvm), подключив репозиторий `oVirt` (в котором хранится собранный пакет, содержащий необходимый функционал).

Выполните команды:

```
# yum -y install http://resources.ovirt.org/pub/yum-repo/ovirt-release36.rpm
# yum -y install qemu-kvm-ev
```

В результате, будут заменены следующие пакеты: *qemu-kvm*, *qemu-kvm-common* и *qemu-img*.

8.2.7 Живая миграция с использованием двух хранилищ

В ПО Базис.Cloud реализована возможность миграции блочных и файловых дисков между системами хранения, без останова работы виртуальных машин («на-лету»). В результате выполнения такого рода живой миграции (между устройствами разных типов) исходное стройство будет удалено автоматически, освободив при этом дисковое пространство, занимаемое ранее (виртуальным диском).

> **Внимание**
>
> В демонстрируемом кейсе платформа использует сразу два типа устройств хранения – SATA и SAS. Устройства расположены на разных физических хранилищах (type1 и type2, соответсвенно).

Создайте виртуальную машину, корневой диск которой будет размещен на системе хранения второго типа (SAS). Диск не является эфмерным, т.е. он обслуживается как *блочное*, а не файловое устройство. На него следует установить образ гостевой ОС, после чего следует запустить виртуальную машину с этого (виртуального) диска.

После того как ВМ создана (с СХД второго типа) и активна, следует запустить процесс миграции диска ВМ, с одного хранилища на другое. В процессе миграции будет создана копия исходного объекта на системе хранения первого типа (type1), и произведена синхронизация состояния данных объектов.

Откройте консоль ВМ и убедитесь, что она все это время работает (активна) и доступна. Дождитесь синхронизации устройств хранения, пока не станет видно, что система начала подключать новый диск к гостевой ОС (система все еще работает с СХД второго типа).

Проверьте, зарегистрировано ли это устройство внутри операционной системы ? Должен быть виден диск `/dev/vda1` и его параметры. Этот диск – корневой диск ВМ; с ним постоянно идет внутрисистемное взаимодействие.

Когда процесс миграции завершится, будет видно, что виртуальная машина работает уже с СХД первого типа, т.е. «переехала» (с одного хралища на другое).

Проверьте, что произошло внутри ВМ ?.. Должно быть очевидно, что новый диск полностью идентичем предыдущему, а гостевая ОС не обнаружила подмены хранилища. Проверьте состояние (гостевой) системы и убедитесь, что она нормально функционирует.

8.3 Миграция между однотипными платформами

Подготовительный этап затрагивает только данные виртуальных машин, хранящихся в исходной системе виртуализации в виде *образов*. Формат хранения образа остается неизменным ввиду того, что миграция ВМ осуществляется между однотипными платформами (Базис.Cloud).

Начальные условия:

1. Наличие плана миграции, содержащего основные сведения об исходной инфраструктуре и перечень виртуальных машин, которые предстоит мигрировать.
2. Наличие АРМ администратора с установленным системным и общесистемным ПО:
   - ОС Linux;
   - клиент службы удаленного доступа SSH;
   - служба гипервизора libvirt;
   - служба клиента OpenStack;
   - веб-браузер, совместимый с TIONIX.Dashboard.

> **Внимание**
>
> Для нормальной совместимости рекомендуется использование рекомендованного базового дистрибутива ОС Linux, устанавливаемого на управляющие и вычислительные узлы инфраструктуры (см. документ `Руководство архитектора ПО Базис.Cloud`).

8.3.1 Планирование операций по миграции

Носитель, используемый для сохранения исходного образа ВМ в виде файла, должен иметь свободное место объемом не менее того, что отображается в свойствах ВМ интерфейса управления (TIONIX.Dashboard).

Производится калькуляция исходных ресурсов, подготовка и заполнение таблицы с параметрами исходных образов виртуальных машин (ВМ). Требования к ресурсам (оперативной памяти, внешней памяти и количеству виртуальных процессорных ядер) исходной и целевой платформ должны соотноситься в части тех ВМ, которые планируется мигрировать.

Кратковременный вывод ВМ из эксплуатации в исходной облачной инфраструктуре обязателен. Это гарантирует целостность виртуального диска, выгружаемого в файл образа.

Исходная инфраструктура (платформа)

Исходная облачная инфраструктура – среда с поддержкой виртуализации, построенная на основе гипервизора и СПО OpenStack. Далее будет называться *исходной платформой*.

Облачная платформа BASIS поддерживает функционирование множества гостевых операционных систем Windows, серверных (Server) и настольных (выпуски Windows 7/8/10), а также большинства дистрибутивов Linux: CentOS, RHEL, Fedora, Debian, Ubuntu.

Администратор облачной платформы может выполнять любые действия, связанные с обслуживанием VDI (инфраструктуры виртуальных рабочих столов), в том числе действия, связанные с управлением проектом и объектами инфраструктуры: сетями/подсетями, образами, виртуальными машинами, пользователями.

Целевая инфраструктура (платформа)

Целевая облачная инфраструктура, также как исходная (облачная) инфраструктура, построена на базе программного продукта Базис.Cloud (далее будет называться *целевой платформой*).

Целевая платформа обладает высокой обратной совместимостью с исходной платформой, а также унифицированным графическим интерфейсом управления, основанным на веб-технологиях.

> **Примечание**
>
> Незначительные изменения в ПО, влияющие на улучшение потребительских качеств инфраструктуры, могут появляться с новыми релизами.

8.3.2 Подготовка доступа к исходной и целевой платформам

На исходной платформе выполнить следующие действия:

1. Подключиться к первому контроллеру кластера (control1) с помощью SSH.

   Подключение выполняется из АРМ администратора, с помощью команды:

```
ssh root@control1
```

После успешного ввода пароля и авторизации на стороне контроллера будет выводиться следующее приглашение к вводу команд:

```
[root@control1 openstack]#
```

2. Открыть для редактирования файл настройки окружения `admin-openrc`.

Файл находится в папке `/root`. Пример содержания файла:

```
export OS_PROJECT_DOMAIN_NAME=default
export OS_USER_DOMAIN_NAME=default
export OS_PROJECT_NAME=<название_проекта>
export OS_USERNAME=<имя_пользователя>
export OS_PASSWORD=<пароль_пользователя>
export OS_AUTH_URL=http://controller:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
export OS_VOLUME_API_VERSION=2
export OS_AUTH_TYPE=password
export DB_SUPERUSER_PASSWORD=<пароль_администратора_БД>
export RABBIT_PASSWORD=<пароль_контроллера_шины_сообщений>
```

Примечание

В строке `export OS_USER_DOMAIN_NAME=` должно быть указано *имя домена*, в который будет осуществляться импорт.

3. Выполнить файл настройки окружения «от источника»:

```
source /root/admin-openrc
```

Примечание

Рекомендуется убедиться, что настройки окружения, необходимые для нормальной работы OpenStack, вступили в силу. Выполните команду:

```
env | grep OS_
```

4. Настройки окружения, используемые для доступа к исходной платформе, скопировать в файл окружения целевой платформы (`admin-openrc-remote`):

```
cd /root && cp admin-openrc admin-openrc-remote
```

После копирования отредактировать копию таким образом, чтобы корректно перенастроилось окружение на доступ к целевому облаку из того же контроллера (исходного облака).

Примечание

Копия нужна для смены настроек окружения, позволяющей администратору выполнять все операции по выгрузке и загрузке облака с одного контроллера.

Переключение между исходной и целевой инфраструктурами производится, не выходя из оболочки (root@control). При этом используется один и тот же файл образа (.img-формата), сохраненный *локально* (на контроллере исходной платформы).

На целевой платформе отредактировать файл настройки окружения `admin-openrc`. Строка настройки (export OS_USER_DOMAIN_NAME=...) должна содержать имя домена, в который будет осуществляться импорт (загрузка образа).

8.3.3 Импорт образа из исходной платформы «БАЗИС»

> **Примечание**
>
> Имя образа, предназначенного для выгрузки, может быть точно определено с помощью графического интерфейса управления облачной инфраструктурой (TIONIX.Dashboard).

На исходной платформе, из которой импортируется ВМ пользователя, выполните следующие действия:

1. Откройте для редактирования проект, из которого будет импортироваться инстанс (настройки ВМ).
2. Найдите имя образа, указанного в проекте.

   На УУ следует выполнить команду:

```
openstack image list
```

   Пример вывода списка доступных образов (в консоль):

```
| af9de22e-4391-4b99-b03f-f9bd5706aa7a | win2016-trial | active |
| cb938347-e10a-4765-9f7e-1f920de3fac8 | win2019-20200804 | active |
| c5e7fe77-9594-4d7d-9347-d4bdf0c9a845 | windows7-import-no-scsi | active |
```

3. Найти имя/ID образа, указанного в проекте, выполнив команду:

```
openstack image show <название_образа>
```

   Пример вывода детальной информации по указанному образу:

```
| name | win10-20200622 |
| owner | 79f4dbda960441599672cb68938a858f |
| properties | hw_disk_bus='scsi', hw_qemu_guest_agent='yes', |
| | hw _scsi_model='virtio-scsi', hw_video_model='qxl', |
| | hw_vif_multiqueue_enabled='true', os_distro='windows', |
| | os_require_quiesce='yes', os_type='windows' |
| protected | False |
```

4. Зафиксируйте данные о свойствах образа на бумажном/электронном носителе.

> **Примечание**
>
> Зафиксированные данные properties потребуются далее (при импорте образа).

5. Выполните дальнейшие действия:

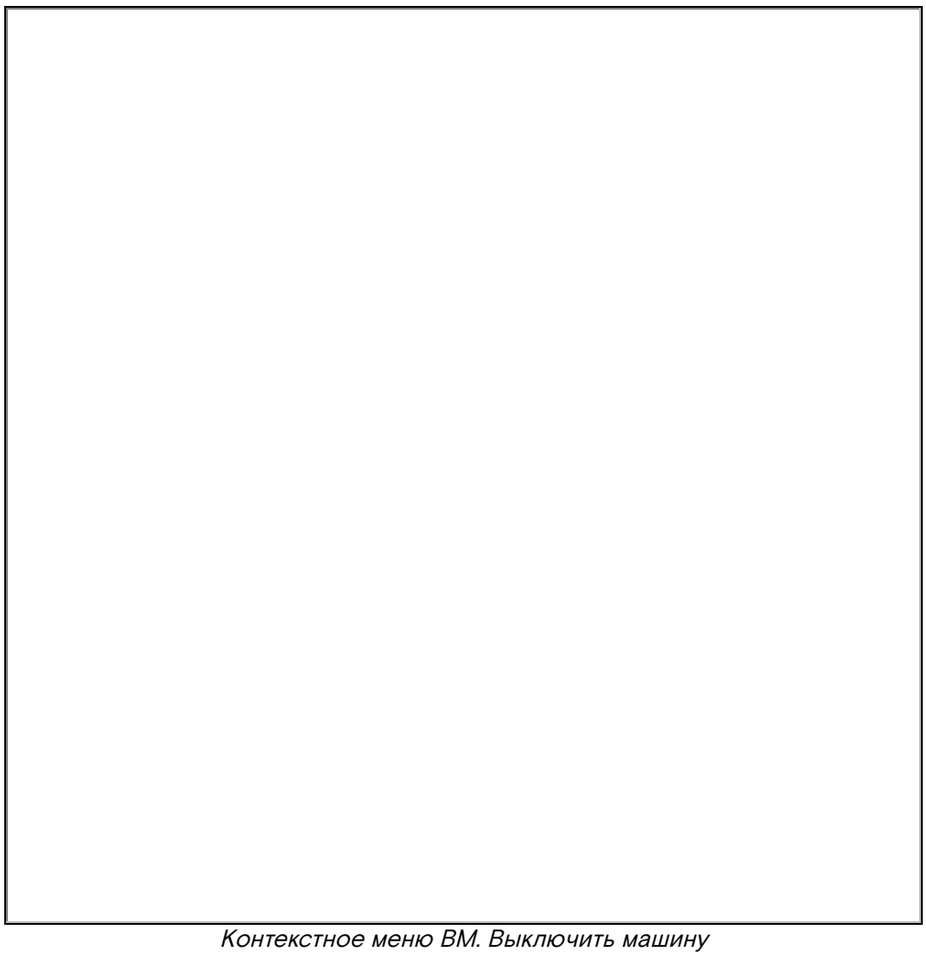
   a. остановите работу виртуальной машины;
   b. сохраните образ виртуальной машины;
   c. выгрузите образ из облака в файл, который необходимо перенести на другую (целевую) облачную платформу.

Подробное описание действий по останову ВМ, сохранению и выгрузке образа из облака (в виде img-файла) приведено ниже.

> **Важно**
>
> Чтобы снизить время простоя пользователя, рекомендуется после успешного сохранения образа ВМ сразу же вернуть её в работу.

Останов работы ВМ

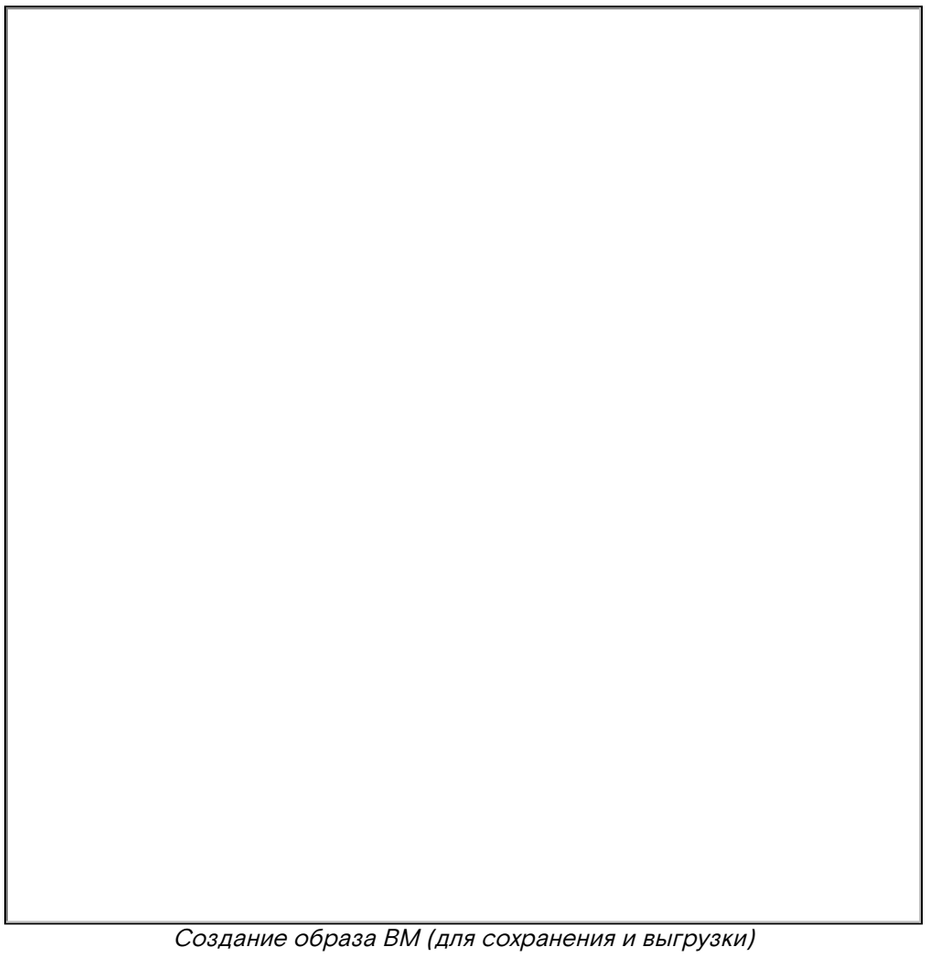
Необходимо остановить работу той ВМ, которой назначен конкретный пользователь и которая предназначена для импорта. Для этого в интерфейсе управления (TIONIX.Dashboard) для выбранной машины из выпадающего списка доступных действий следует кликнуть «Выключить машину» (`pic1-MOV`).



Контекстное меню ВМ. Выключить машину

Сохранение образа ВМ

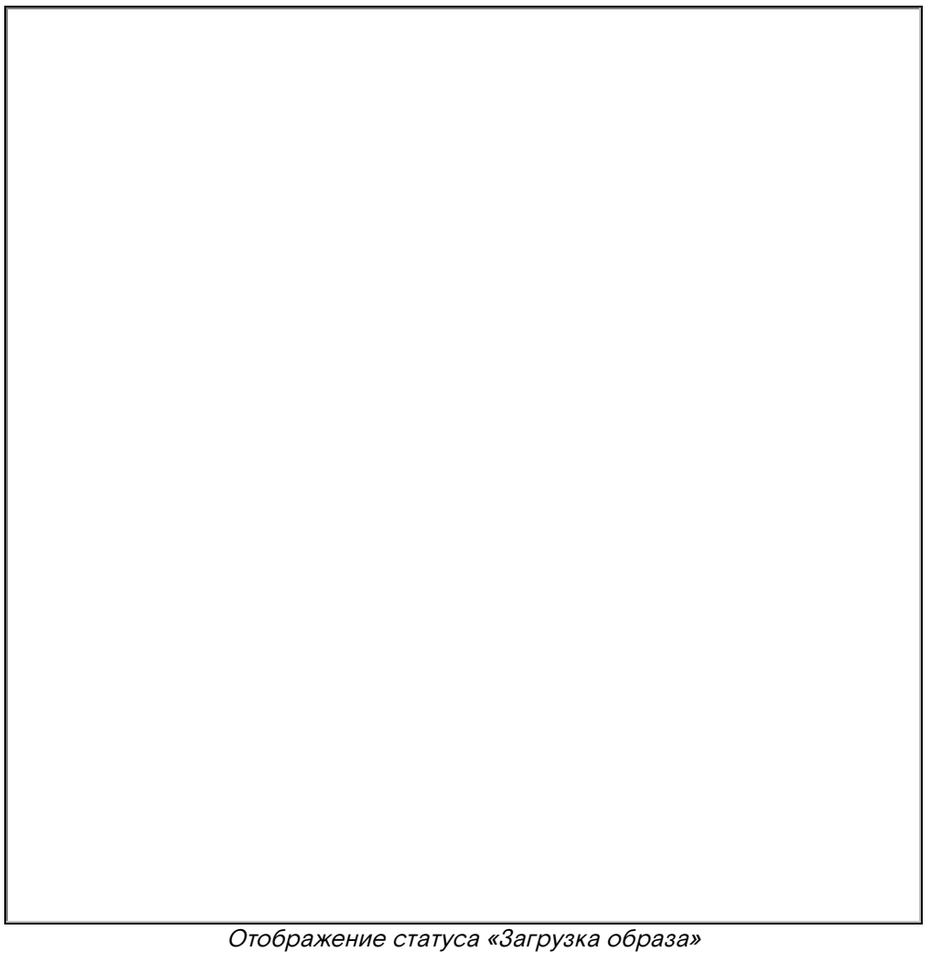
Для сохранения образа машины в интерфейсе управления перейдите: Администратор >> Создать образ



Создание образа ВМ (для сохранения и выгрузки)

Длительность выполнения операции создания образа зависит от размера создаваемого образа (например – ВМ `vdi_win7-18` на `pic2-MOV`), а также от производительности используемых подсистем хранения данных.

Необходимо дождаться завершения операции. В процессе выполнения операции будет отображаться статус «Загрузка образа» (`pic3-MOV`).



Отображение статуса «Загрузка образа»

После завершения операции создания (загрузки) образа выполните команду (на УУ):

```
# openstack image list
```

Образ, сохраненный в хранилище OpenStack, появится в списке, как показано ниже:

```
| ID | Name | Status |
| 08e7fae5-1294-4d1f-80e1-786fb69b9e5a | vdi-win7-16_2020-10-26_14:23:21 | active |
| 007c817c-9a6f-4d58-831d-2e3f795dfd9b | win7ent20200625-raw | active |
| 41835e8d-d268-4673-89b0-06750e76f3e8 | win8-VDI-20200618-raw | active |
| 1338f463-7fd8-4ceb-a282-e2d43a58a4d9 | winserver16dc-compact | active |
```

Например:

- имя созданного образа – `vdi-win7-16_2020-10-26_14:23:21`;
- идентификатор образа (ID) – `08e7fae5-1294-4d1f-80e1-786fb69b9e5a`.

Возвращение ВМ в работу

После завершения предыдущей операции (создания образа) включите ВМ, чтобы пользователь смог продолжить работу, а не ожидал, пока завершится весь процесс миграции.

Выгрузка образа (из облака)

С помощью консоли выполните команду OpenStack, чтобы образ сохранить в файл:

```
# glance image-download --file vdi-win7-...img <ID-образа>
```

По завершении операции выгрузки, которая займет некоторое время, файл образа в формате img с именем `vdi-win7-…` будет храниться в локальной файловой системе того контроллера (УУ), из которого была выполнена данная операция.

Примечание

При необходимости, файл образа диска может быть сконвертирован из RAW-формата в формат QCOW2. Это позволит ускорить процесс загрузки ВМ, путём снижения нагрузки на сетевую инфраструктуру облака, связывающую вычислительный узел (Compute) с хранилищем данных (Storage).

8.3.4 Загрузка образа на целевую платформу

После того как образ виртуального диска выгружен из исходной платформы (доступен в виде файла), можно приступить к интеграции этого образа в подсистему обслуживания образов целевой платформы.

Также как в исходной платформе, используется командная строка (Linux/OpenStack) и веб-интерфейс управления .

Примечания

Фактически, миграция ВМ из исходной платформы осуществляется путем подстановки выгруженного образа виртуального диска, содержащегося в img-файле, в новую ВМ, созданную на целевой платформе.
Проверку доступности к управляющему узлу целевой инфраструктуры можно выполнить командой:

```
ssh root@IP-контроллера
```

Внимание

На стороне контроллера должен быть настроен должным образом SSH-сервер, обслуживающий запросы SSH-клиента.

На целевой платформе выполните следующую последовательность операций:

1. Настройте рабочее окружение для выполнения команд OpenStack (CLI):

```
source /root/admin-openrc-remote
```

2. Загрузите img-файл с исходным образом в службу образов (OpenStack Glance) целевой платформы.

   Выполните следующую команду:

```
# openstack image create \
--disk-format raw --container-format bare \
--public --file <название_образа>.img \
--property hw_vif_multiqueue_enabled=true \
--property hw_scsi_model=virtio-scsi
--property hw_disk_bus=scsi \
--property hw_qemu_guest_agent=yes \
--property hw_video_model=qxl \
--property os_require_quiesce=yes \
--property os_distro=windows \
--property os_type=windows \
<ИМЯ_ОБРАЗА>
```

   где<ИМЯ_ОБРАЗА> – имя создаваемого образа на целевой платформе.

Примечания

В качестве параметров для опции –property используйте параметры вывода команды **openstack image show**.
С подробным описанием используемых свойств (атрибутов), представленных парами ключей после опций propery, можно ознакомиться, перейдя по ссылке:
https://docs.openstack.org/glance/rocky/admin/useful-image-properties.html

8.3.5 Подключение образа к целевой виртуальной машине

Войдите в интерфейс управления (TIONIX.Dashboard) целевой платформы. Вход (в Dashboard) выполняется с АРМ администратора, в веб-браузере, по ссылке: http://<IP-адрес_контроллера>/dashboard/project/instances/

После ввода реквизитов, позволяющих выполнять административные действия над проектом, откроется интерфейс управления облачной инфраструктурой.

Выполните следующие действия:

1. Если не указано название проекта, следует указать название по-умолчанию (default).
2. Выберите и откройте для редактирования VDI-проект, в который будет добавлена ВМ.
3. Замените существующий в проекте образ на новый образ (созданный с помощью Glance).

Примечание

При успешном завершении операции создания образа его статус должен быть – «Активный».

4. Создайте одну новую виртуальную машину в проекте (default).

Перейдите: Проект » Вычисления » Виртуальные машины

Откройте веб-диалог (мастер) создания виртуальной машины, открываемый кликом кнопки [Создать машину].

5. Проверьте созданную ВМ с присоединенным образом на функциональное соответствие с исходным образцом.

После проверки оставьте ВМ в активном состоянии, чтобы пользователь смог незамедлительно приступить к работе в среде Рабочего стола.

6. Для открытия доступа по протоколу RDP перейдите в раздел «Группы безопасности» и добавьте необходимый порт (протокол TCP).

8.3.6 Дополнительные рекомендации

Чтобы загрузка ВМ происходила быстрее, т.е. ожидание пользователя при запросах на подключение к Рабочему столу неактивной машины было минимальным, исходный img-файл образа может быть сконвертирован в формат QCOW2 – сразу же после выгрузки из службы хранения Glance. Такой образ принято называть *оптимизированным*.

Конвертация (оптимизация) образа выполняется с помощью команды:

```
qemu-img convert -O qcow2 -c <vdi-win7_from>.img <vdi-win7_to>.qcow2
```

где

<vdi-win7_from>.img – образ, выгруженный из исходной платформы;

<vdi-win7_to>.qcow2 – образ, загружаемый на целевую платформу.

Загрузка образа с помощью клиента OpenStack потребует указания для параметра `--disk-format` соответствующего значения (qcow2):

```
# openstack image create \
--disk-format qcow2 \
--container-format bare \
--public --file <название_образа>.qcow2 \
--property hw_vif_multiqueue_enabled=true \
--property hw_scsi_model=virtio-scsi \
--property hw_disk_bus=scsi \
--property hw_qemu_guest_agent=yes \
--property hw_video_model=qxl \
--property os_require_quiesce=yes \
--property os_distro=windows \
```

```
--property os_type=windows \
<ИМЯ_ОБРАЗА>
```

9 Автоматическая эвакуация

Модуль `TIONIX.NodeControl` при использовании функции автоэвакуации может воспользоваться возможностями управления питанием вычислительных узлов для их временного вывода из эксплуатации . Он может регистрировать *устройства питания*, которые затем могут быть привязаны к конкретному вычислительному узлу.

При использовании функции автоэвакуации модулем TIONIX.NodeControl используется дополнительный механизм *проверки узлов*, который полагается не только на информацию, предоставляемую службой вычислений (Nova). Необходимо сначала настроить хранилище статусов – проверки доступности , а затем привязать его к вычислительному узлу.

Механизм авто-эвакуации использует функционал:

- службы NodeControl, которая функционирует на УУ (в кластере управления);
- средств поддержки протоколов управления питанием (IPMI, SSH и т.д.).

> **Внимание**
>
> Перед включением функции **автоэвакуации** убедитесь, что все службы и средства поддержки протоколов управления питанием, требуемые для нормальной работы, настроены и корректно работают.

Необходимо настроить должным образом и привязать к гипервизору устройство управления питанием , а на контроллер(ы) установить соответствующий пакет поддержки (IMPI) или другие утилиты (поддержки протоколов обмена), соответственно типу устройств – модулей, установленных в серверные компьютеры.

Для нормальной работы механизма **автоэвакуации**, алгоритм которого изложен ниже , должны быть соблюдены следующие условия:

1. настроен `shared-storage` (СХД с Cinder тоже подходит);
2. настроен модуль `TIONIX.NodeControl`;
3. заведен доступ к узлам через сеть IPMI;
4. настроено хранилище проверки доступности (ХПД).

9.1 Алгоритм авто-эвакуации

При смене статуса гипервизора на «down» службой NodeControl отрабатывается следующий алгоритм:

1. Определяется число потерянных ВУ, которое рассчитывается как число всех ВУ, находящихся в статусе «down», за исключением резервных;
2. Если число потерянных ВУ превышает число, указанное в параметре „MAX_DOWN_HOSTS“, то алгоритм прерывается;
3. Если не выставлен параметр „ALLOW_HOST_AUTO_POWER_OFF“, то выполняется **автоэвакуация** с включением *резервного ВУ*:

    a. Выполняется поиск резервного ВУ и его включение;
    b. Выполняется функция эвакуации всех виртуальных машин, находящихся на ВУ.
        - Выполняется функция эвакуации всех виртуальных машин, находящихся на ВУ.

4. Иначе (параметр „ALLOW_HOST_AUTO_POWER_OFF“ выставлен) выполняются следующие действия:

    a. Выполняется поиск на гипервизоре всех виртуальных машин, находящихся в статусе «Active»;
    b. Выполняется перезапуск ВУ узла с ожиданием таймаута, указанного в параметре „HOST_RESTART_TIMEOUT“;
    c. Опрашивается статус ВУ:
        - если статус – «up», то выполняется перезагрузка всех виртуальных машин вычислительного узла, которые находились в статусе «Active»;
        - если статус – «down», то выполняется автоэвакуация с включением резервного ВУ.

9.2 Резервный гипервизор

Модуль TIONIX.NodeControl, функционирующий на контроллере (УУ), дает возможность назначить резервные гипервизоры (из списка доступных). Это обеспечит «прозрачное» восстановление работоспособности ВМ, если на обслуживающем её ВУ обнаружен отказ.

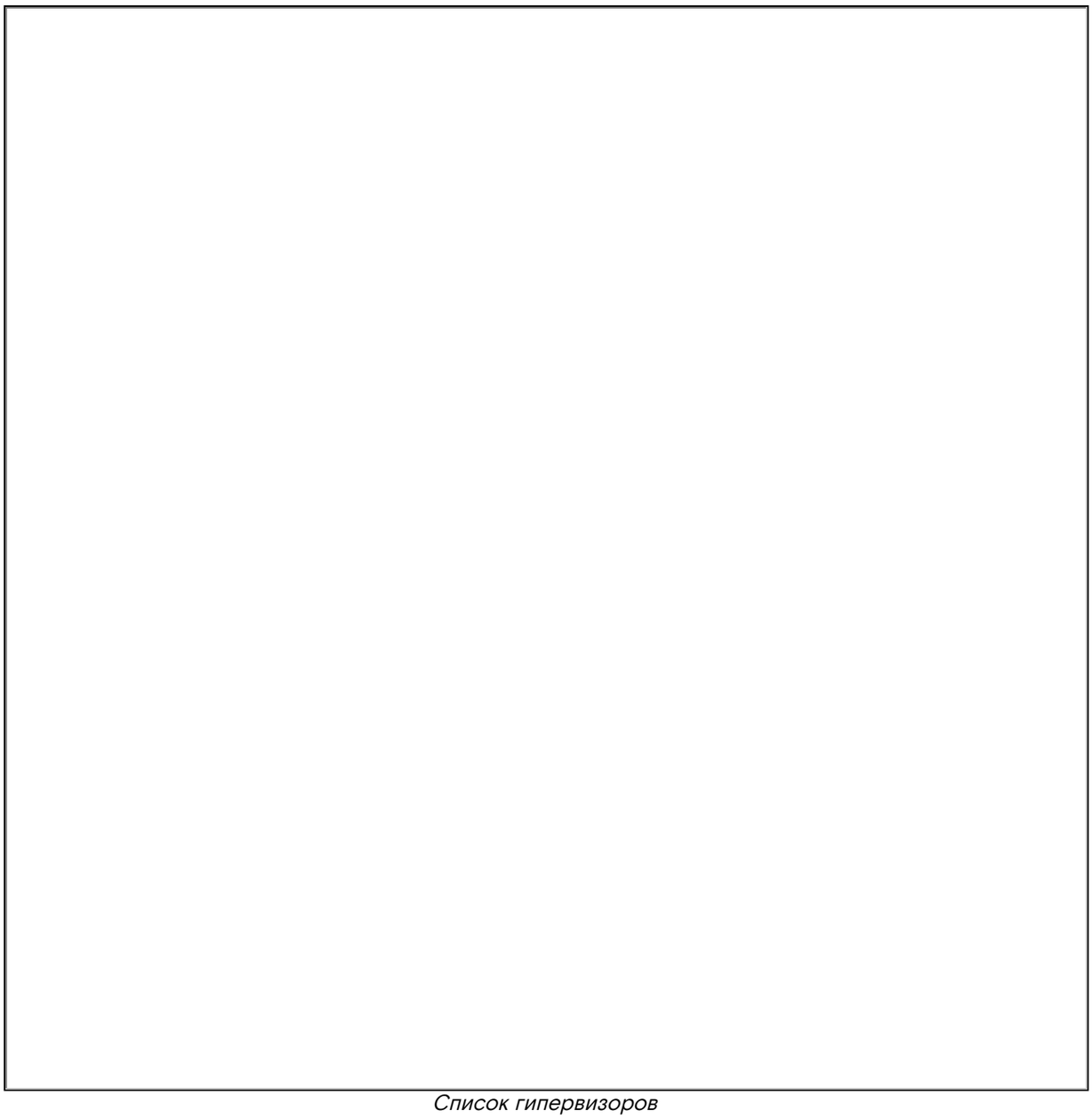
Гипервизор относится к резервным, если он (соответствующий ему ВУ):

- выставлен и помечен как резервный;
- имеет порт управления питанием;
- выключен по питанию (через устройство управления питанием).

При выходе из строя гипервизора, функционирующего на ВУ, будет включаться запасной ВУ – ввод одного из доступных резервов . Если резервных гипервизоров не найдено, то замена не произойдет; об этом будет сообщено в лог-файле модуля (TIONIX.NodeControl).

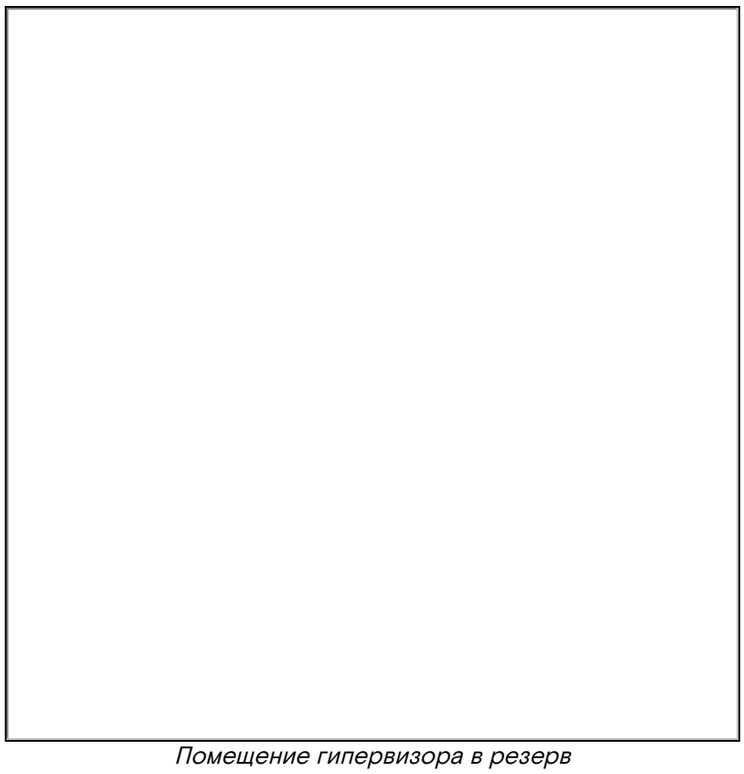
Назначить резервный гипервизор можно посредством интерфейса управления . Перейдите: Администратор >> Вычисления >> Гипервизоры

Будет отображена сводка по гипервизору, а в горизонтальной вкладке «Гипервизор» – список вычислительных узлов (`Compute_lst`).



Список гипервизоров

Пользуясь контекстным меню справа, примените к выбранному гипервизору действие – «Поместить в резерв» или выставите для него флаг «резерв».



Помещение гипервизора в резерв

Внимание

Должно быть предварительно настроено средство управления питанием и сопоставлено с ВУ, назначаемым в резерв.

Примечание

Подробная инструкция о выполнении операции перевода ВУ в резерв изложена в программной документации.

9.3 Параметры настройки NodeControl

Настройка работы автоэвакуации производится путем изменения конфигурационного файла модуля TIONIX.NodeControl (`/etc/tionix/node_control.yaml`).

Важно

После внесения изменений в конфигурационный файл требуется перезапуск системной службы (NodeControl).

9.4 Средства управления питанием

NodeControl при использовании функции *автоэвакуации* может воспользоваться возможностями управления питанием вычислительных узлов, для их временного вывода из эксплуатации. Для этого NodeControl регистрирует *устройства питания*, которые затем могут быть привязаны к вычислительному узлу (т.н. операция назначения средства управления питанием определенному гипервизору).

Для работы требуется предварительно выполнить настройку инфраструктуры, обеспечивающую поддержку операций, связанных с управлением питания ВУ:

- службы NodeControl (УУ);
- протоколы для управления питанием (IPMI, SSH и так далее).

Перед включением этой функции убедитесь, что требуемые сервисы настроены и корректно работают.

9.4.1 Функциональные возможности

Утилиты управления питанием вычислительных узлов позволяют:

- определить и предоставить (по запросу) информацию о соответствии вычислительного узла и контактной площадки устройства управления питанием;
- назначить резервные вычислительные узлы, которые включаются в случае выхода из строя задействованных (замена аварийного вычислительного узла);
- управлять питанием по адресу контактных площадок и устройств управления питанием;
- управлять питанием по именам вычислительных узлов.

9.4.2 Поддерживаемые типы устройств

Для управления питанием используется аппаратно-программный комплекс. Аппаратная часть представлена устройством управления питанием, как правило – промышленным контроллером (модулем). Например: ICPDAS, DAEnetIP2 и др. Программная часть состоит из клиента (инструментальной утилиты), позволяющего удаленно подключаться к устройству и выполнять обмен по поддерживаемому *протоколу управления*.

Реализована поддержка следующих устройств управления питанием:

- устройства на основе платы DAEnetIP2, использующей протокол SNMP;
- устройства ICP DAS ET-7067, использующие открытый протокол MODBUS;
- устройства с поддержкой технологии AMT;
- устройства с поддержкой интерфейса IPMI ;
- виртуальные устройства, использующие протокол SSH (для управления гипервизором).

Модуль обеспечивает следующие возможности:

- получение информации о состоянии питания портов устройства;
- управление состоянием портов устройства (включение-выключение);
- запуск «мягкого» выключения для устройств, реализующих режимы ACPI.

9.4.3 Инициализация средства управления питанием

Основная команда, выполняемая с целью инициализации устройства – средства управления питанием серверного компьютера – из командной строки (OpenStack CLI):

```
openstack tnx power init
```

Примечание

Для нормальной работы клиента (**openstack**) должно быть настроено окружение.

Команда работает в интерактивном режиме и состоит из следующих вопросов:

- тип устройства;
- тип коммуникационного (сетевого) протокола;
- IP-адрес или доменное имя устройства управления;
- сетевой порт устройства (от 1 до 65535);
- имя пользователя для устройства управления;
- пароль пользователя для устройства управления;
- имя устройства управления в NodeControl.

В зависимости от типа добавляемого устройства настройка будет отличаться.

Для получения списка устройств питания выполните команду:

```
openstack tnx power list
```

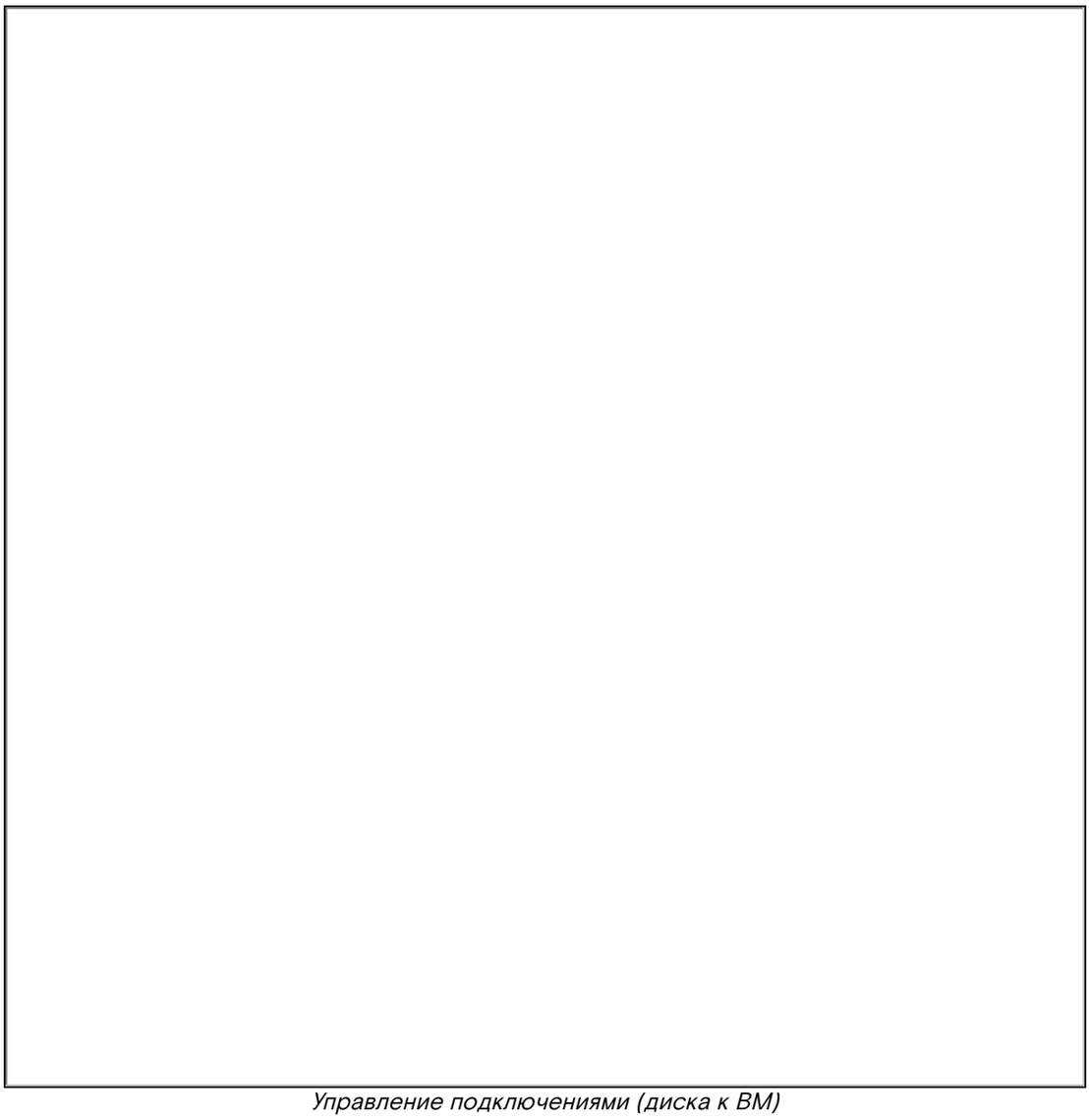
Команда manage позволяет управлять устройствами питания:

```
openstack tnx power manage
```

9.4.4 Назначение средства управления питанием

Войдите в интерфейс управления (Dashboard) и выполните переход по меню:БАЗИС >> Средство управления питанием.

Выберите действие – Добавить новое средство управления питанием (`Power_control`).



Управление подключениями (диска к ВМ)

Далее перейдите в Администратор >> Вычисления >> Гипервизоры и назначьте **средство управления питанием** выбранному гипервизору (ВУ).

Примечание

В качестве инструмента управления питанием виртуальных машин применяется утилита `Virtual BMC`

Выполните команду:

```
virsh list | grep compute
```

Будет получен построчный вывод состояния виртуальных машин:

```
1279 os-tcp-queens-compute1 running
1280 os-tcp-queens-compute2 running

…
```

9.5 Общее хранилище

Общее хранилище можно реализовать следующими способами:

- хранилище с использованием GlusterFS (рекомендуемое);
- хранилище формируется с помощью NFS;
- используя встроенные средства подключенной СХД.

Примечание

Возможно использование локального общего хранилища Cinder[6].

9.5.1 Создание хранилища с использованием GlusterFS

Для обеспечения отказоустойчивости кластер GlusterFS создается на всех управляющих и вычислительных узлах, после чего файловая система может быть доступна на localhost каждого узла.

В первую очередь, требуется выполнить подключение репозитория (Gluster-9):

```
dnf install https://repo.tionix.ru/centos/8-stream/storage/x86_64/gluster-9/centos-release-gluster9-tionix-1.0-2.el8.noarch.rpm
```

Чтобы избежать конфликта пакетов при установке, требуется зафиксировать версию пакета `glusterfs-selinux`. Выполните команды:

```
dnf install python3-dnf-plugin-versionlock
dnf versionlock add glusterfs-selinux-0.1.0
```

Установите пакет `glusterfs-server` на всех управляющих и вычислительных узлах:

```
dnf install glusterfs-server
```

Запустите и добавьте в автозагрузку сервис `glusterd.service`. На всех узлах, управляющих и вычислительных, выполните команду:

```
systemctl enable --now glusterd.service
```

Инициализация кластера (GlusterFS)

Инициализируйте кластер(с VIP-узла), состоящий из управляющих и вычислительных узлов инфраструктуры. Данное действие требуется произвести для каждого узла кластера, кроме VIP:

```
gluster peer probe <hostname>
```

где:

<hostname> – имя узла добавляемого в кластер.

Создание общего хранилища (GlusterFS)

Создание общего хранилища для всех узлов осуществляется с VIP-узла:

```
gluster volume create <volume_name> replica <count> <hostname>:<storage_path> force
```

где:

- <volume_name> – имя хранилища GlusterFS;
- replica <count> – количество узлов GlusterFS (количество управляющих и вычислительных узлов инфраструктуры);
- <hostname> – имя узла в кластере;
- <storage_path> – путь к хранилищу на узле;
- force – опция позваляющая создать хранилище на корневом диске.

Например, создание реплицированного хранилища может быть выполнено с помощью команды:

6 https://docs.tionix.ru/3.0/developer_guide/tionix-node-control-docs/tionix_node_control/management_cinder_backend.html

```
gluster volume create replicated replica 6 control1:/mnt/status-storage control2:/mnt/
status-storage control3:/mnt/status-storage compute1:/mnt/status-storage compute2:/mnt/
status-storage compute3:/mnt/status-storage force
```

Запуск и подключение общего хранилища (GlusterFS)

Требуется запустить созданное хранилище:

```
gluster volume start <volume_name>
```

где:

<volume_name> - имя хранилища GlusterFS

Смонтируйте файловую систему на каждом из узлов, в папку, указанную при создании ХПД:

```
mount -t glusterfs <hostname>:/<volume_name> <status_storage_path>
```

где:

<hostname> – имя узла в кластере, возможно использования localhost; <volume_name> – имя хранилища GlusterFS; <status_storage_path> – папка, указанная при создании ХПД (например `/etc/tionix/statestore`).

Например:

```
mount -t glusterfs localhost:/mnt/status-storage /etc/tionix/statestore
```

Для автоматического монтирования файловой системы при перезагрузке сервера потребуется добавить строку в файл `/etc/fstab`:

localhost:/mnt/status-storage[7] /etc/tionix/statestore glusterfs defaults,_netdev 0 0

Проверка состояния GlusterFS

Для проверки состояния кластера исспользуется команда:

```
gluster peer status
```

Пример вывода:

```
Number of Peers: 6

Hostname: control2
Uuid: 7720c78a-6217-4287-990e-491c87a9c1f4
State: Peer in Cluster (Connected)

Hostname: control3
Uuid: 894948b1-b85e-4977-9fd9-68945b9fb94d
State: Peer in Cluster (Connected)

Hostname: compute1
Uuid: cb747fc9-18b3-4f01-b1d5-f68295306b5d
State: Peer in Cluster (Connected)

Hostname: compute2
Uuid: 569b49ff-cb0e-4151-8d06-0486fe7f666e
State: Peer in Cluster (Connected)

Hostname: compute3
Uuid: 9c125b82-79e7-413a-aa1b-cffacd3b5aa1
State: Peer in Cluster (Connected)

Hostname: storage1
Uuid: f135f126-e51a-42be-9231-212350dd4d25
```

7 http://localhost/mnt/status-storage

```
State: Peer in Cluster (Connected)
```

Для проверки состояния хранилища используется команда:

```
gluster volume info <volume-name>
```

где:

<volume_name> – имя хранилища GlusterFS.

Пример вывода:

```
Volume Name: tnx_storage
Type: Replicate
Volume ID: 710266e1-85b6-45ad-be99-4792d48e1865
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 7 = 7
Transport-type: tcp
Bricks:
Brick1: control1:/etc/tionix/.statestore
Brick2: control2:/etc/tionix/.statestore
Brick3: control3:/etc/tionix/.statestore
Brick4: compute1:/etc/tionix/.statestore
Brick5: compute2:/etc/tionix/.statestore
Brick6: compute3:/etc/tionix/.statestore
Brick7: storage1:/etc/tionix/.statestore
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

9.5.2 Добавление узла в хранилище GlusterFS

Перед добавлением узла в хранилище требуется настроить сервис `glusterd.service` и добавить новый узел в кластер – к существующему хранилищу. Используйте вызов в следующем формате:

```
gluster volume add-brick <volume_name> replica <count> <hostname>:<storage_path> force
```

где:

- <volume_name> – имя хранилища GlusterFS;
- replica <count> – общее количество узлов GlusterFS, равно количеству управляющих и вычислительных узлов;
- <hostname> – имя узла в кластере;
- <storage_path> – путь к хранилищу на узле;
- force – опция, позваляющая создать хранилище на корневом диске.

Например, конечная команда добавления узла может выглядеть так:

```
gluster volume add-brick tnx_storage replica 7 storage1:/etc/tionix/.statestore force
```

9.5.3 Удаление хранилища GlusterFS

> **Внимание**
>
> Перед удалением требуется отмонтировать все узлы от удаляемого хранилища.

Перед удалением хранилища его сначала требуется остановить. Выполните команды:

```
gluster volume stop <volume_name>

gluster volume delete <volume_name>
```

где:

<volume_name> – имя хранилища GlusterFS.

9.5.4 Пример настройки NFS-хранилища

В приведенном ниже примере будет использована существующая директория NFS, подготовленная для Glance.

Подготовка управляющего узла

На контроллере (УУ) выполните проверку использования дискового пространства:

```
[root@control1 ~]# df -h

Файловая система Размер Использовано Дост Использовано% Смонтировано в
10.16.31.195:/mnt/nfs-share 493G 224G 244G 48% /var/lib/glance/images
```

Создайте папку для хранения статусов ВМ (vm-stats):

```
# mkdir /var/lib/glance/images/vm-stats/
```

Подготовка вычислительного узла

9.6 Хранилище проверки доступности (статусов ВМ)

Для настройки хранилища проверки доступности (статусов ВМ) необходимо создать директорию, как на контроллерах, так и на вычислительных узлах. Например, директория может быть создана на сервере NFS или GlusterFS, функционирующим в *общем хранилище* .

> **Примечание**
>
> Операции, выполняемые над хранилищем проверки доступности (сокр. ХПД), изложены в программной документации[8].

> **Внимание**
>
> Для нормальной работы хранилища статусов ВМ на каждый ВУ инфраструктуры должен быть установлен и настроен модуль `TIONIX.Agent`.

Путь к директории могжет быть произвольным, однако в качестве умолчания рекомендуется использовать `/etc/tionix/statestore`. Если хранилищ несколько, то следует создать для них отдельные каталоги (внутри `statestore`).

На всех УУ и ВУ создайте каталог с этим путем и укажите tionix в качестве имени пользователя системы и группы:

```
mkdir /etc/tionix/statestore
chown tionix:tionix /etc/tionix/statestore
```

На этом предварительная подготовка хранилища (статусов ВМ) завершена.

9.6.1 Использование CLI

Зайдите в окружение контроллера облачной платформы по SSH и настройте окружение, необходимое для работы клиента и выполнения команд CLI (**openstack**):

```
ssh root@controller
source ${HOME}/admin-openrc.sh
```

Формат вызова операций над ХПД, осуществляемых с помощью клиента:

```
openstack tnx storage <operation>
```

где:

<operation> – тип выполняемой операции:

- list: вывод списка ХПД;

8 https://docs.tionix.ru/3.0/developer_guide/tionix-node-control-docs/tionix_node_control/availability_storages.html

- create: создание ХПД;
- update: изменение ХПД;
- delete: удаление ХПД.

Для вывода списка ХПД выполняется команда:

```
openstack tnx storage list
```

Для получения уточненной информации (содержит пути к ВУ/УУ):

```
openstack tnx storage list --detail
```

Примечание

Те же действия могут быть выполнены с помощью интерфейса управления (TIONIX.Dashboard). Используйте быстрый переход:
http:/[9]/<IP-облака>/dashboard/tionix/infrastructure/

Для назначения ХПД узлу и снятия назначения с узла используйте команды:

```
openstack tnx storage assign <id> --storages <storage_id>
openstack tnx storage assign <id> --storages ...
openstack tnx storage unassign <id> --storage <storage_id>
```

После опции `--nodes` вместо троеточия необходимо подставить список идентификаторных номеров ХПД, которые необходимо назначить (разделены пробелами).

Создание хранилища

Первым делом создаётся хранилище. Используйте команду:

```
openstack tnx storage create <NAME> /path_to/statestore/dir_on_compute /path_to/
statestore/dir_on_controller
```

где:

- <NAME> – имя хранилища состояний ВМ (проверки доступности);
- /path_to/statestore/dir_on_compute – абсолютный путь в файловой системе ОС, функционирующей на ВУ; путь, где предполагается хранить данные состояния;
- /path_to/statestore/dir_on_controller – абсолютный путь в файловой системе ОС, запущенная в УУ и где предполагается хранить данные состояния узлов.

Важно

Для всех УУ и ВУ указываемый путь должен быть одинаков.

В данном примере конечная команда выглядит так:

```
openstack tnx storage create default /etc/tionix/statestore /etc/tionix/statestore
```

То есть, абсолютные пути к директориям хранилища на УУ и ВУ – одинаковы.

В результате выполнения команды будет получен табличный вывод следующего вида:

Field	Value
Storage ID	2
Storage Name	default
Path for compute	/etc/tionix/statestore

9 https://conf.tionix.ru/http:

Path for controller	/etc/tionix/statestore

Привязка хранилища к ВУ

Для привязки хранилища к ВУ используется отдельная команда assign со следующим форматом вызова:

```
openstack tnx storage assign <STORAGE_ID> --nodes <NODE_ID>
```

где:

- <STORAGE_ID> – ID хранилища статусов (в выводе создания - значение Storage ID);
- <NODE_ID> – ID гипервизора вычислительного узла, можно указать несколько через пробел.

⚠ Не используйте ID службы вычислений `nova-compute`!
Важно также не менять порядок синтаксиса: сначала всегда указывается ID хранилища, а затем – ID гипервизора.

Пример конечной команды:

```
openstack tnx storage assign 2 --nodes 1 2 3
```

В случае успешного выполнения команды будет выведено сообщение:

```
Nodes have been assigned.
```

Для определения ID гипервизоров можно воспользоваться командой:

```
mysql -u tionix --password=<password> --database=tionix_node_control --execute="SELECT
id, name FROM nodes"
```

где:

- <password> – пароль пользователя tionix.

Пример табличного вывода, выводимого в результате:

id	name
4 7 1	compute1.tionix.loc compute2.tionix.loc compute3.tionix.loc

Отвязка хранилища от ВУ

Для отвязки хранилища от ВУ используется команда unassign, со следующим форматом вызова:

```
openstack tnx storage unassign <STORAGE_ID> –node <NODE_ID>
```

где:

- <STORAGE_ID> – ID хранилища статусов (значение Storage ID, полученное при создании);
- <NODE_ID> – ID гипервизора вычислительного узла, к которому привязано хранилище (можно указать только один узел).

⚠ **Внимание**

Не используйте ID службы вычислений `nova-compute`!

Пример конечной команды:

```
openstack tnx storage unassign 2 --node 2
```

В случае успеха будет выведено следующее сообщение:

```
Node has been unassigned.
```

Удаление хранилища

Для удаления хранилища (отвязанного от ВУ) используется команда delete, со следующим форматом вызова:

```
openstack tnx storage delete STORAGE_ID
```

где:

- <STORAGE_ID> – ID хранилища статусов (значение Storage ID, полученное при создании).

Важно

Перед удалением убедитесь в том, что хранилище отвязано ото всех вычислительных узлов.

Пример конечной команды:

```
openstack tnx storage delete 2
```

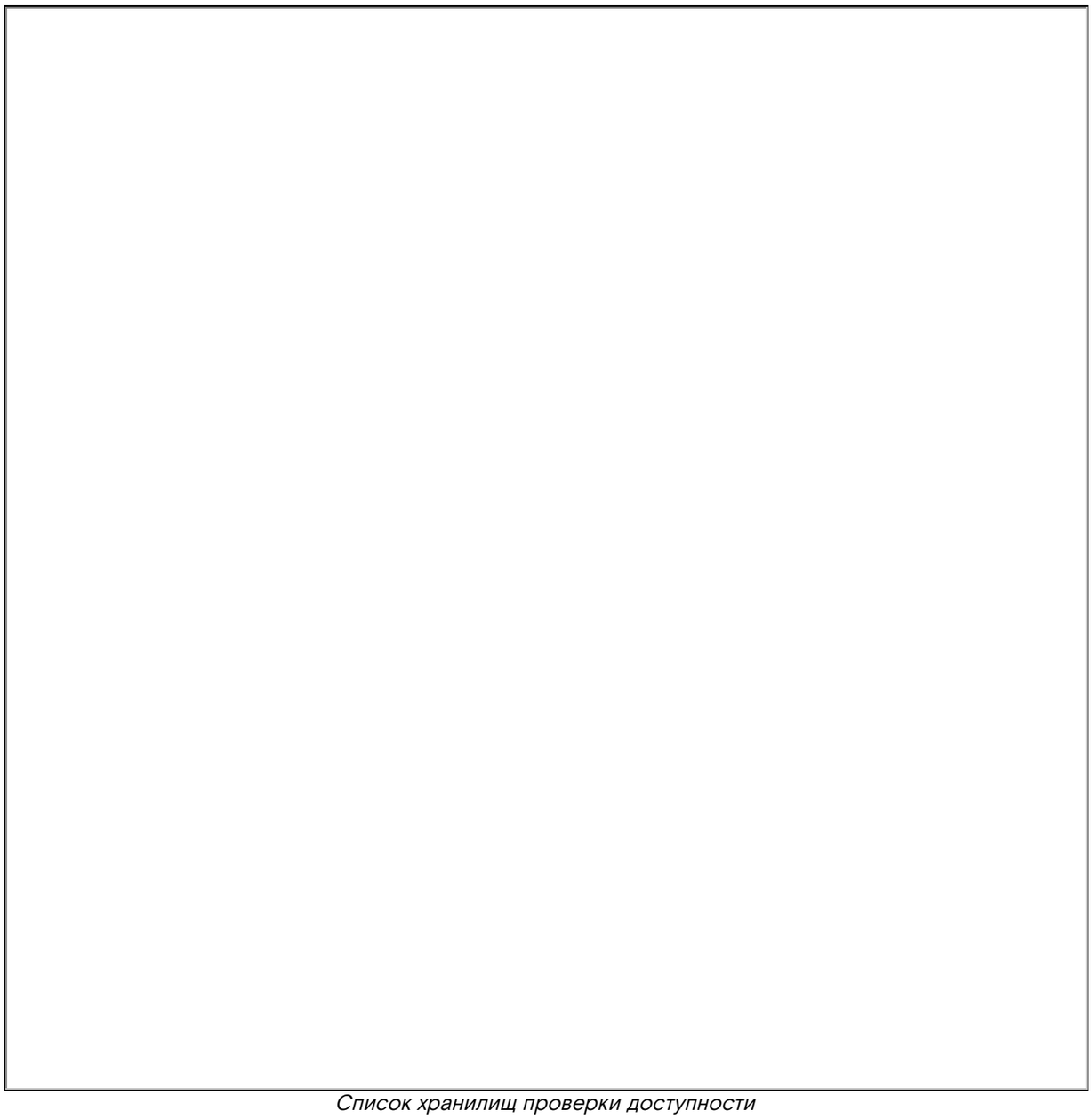
В случае успеха будет выведено следующее сообщение:

Storage with id «1» has been deleted

9.6.2 Использование интерфейса управления

Вывод списка хранилищ

Для получения списка доступных хранилищ проверки доступности перейдите: БАЗИС >> Инфраструктура >> Хранилище проверки доступности



Список хранилищ проверки доступности

Важно

Вкладка доступна только пользователю с правами администратора.

В списке представлена следующая информация:

Путь для вычислительного узла – наименование средства управления питанием, присваивается при создании (редактируется в общем списке); Путь для контроллера – тип средства, задается при создании.

Для списка доступны инструменты сортировки и фильтрации. Поля сортируются по возрастанию и убыванию. Инструмент фильтрации работает по наименованию любого из полей, допустим неполный ввод имени.

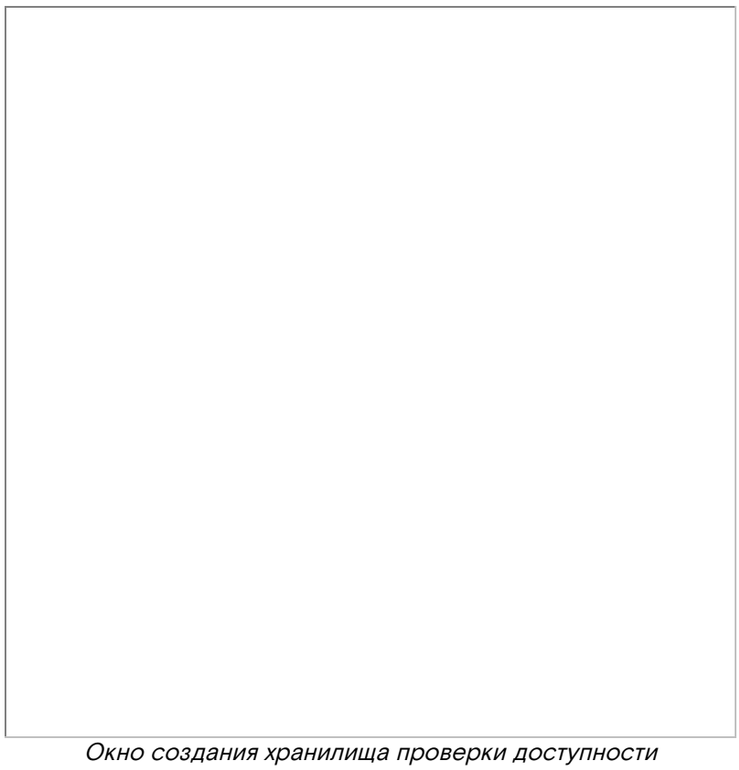
Возможные действия на вкладке:

- Редактировать хранилище
- Изменение параметров хранилища проверки доступности
- Удалить хранилище
- Удаление хранилища проверки доступности.

Создание хранилища проверки доступности

Перейдите: БАЗИС >> Инфраструктура >> Хранилище проверки доступности

Вызовите действие – «Создать хранилище»; откроется окно создания хранилища проверки доступности (StatS_new).



Окно создания хранилища проверки доступности

В открывшемся окне укажите необходимые параметры хранилища проверки доступности. Подробное описание параметров представлено в таблице ниже.

Имя	**Наименование создаваемого хранилища проверки доступности.**
Путь для вычислительного узла	Путь до хранилища проверки доступности на вычислительных узлах.
Путь для контроллера	Путь до хранилища проверки доступности на контроллерах.

Примечание.* – обозначение обязательных для заполнения полей.

Подтвердите операцию создания ХПД нажатием кнопки [Создать хранилище проверки доступности]. Корректно созданное хранилище проверки доступности отобразится в общем списке.

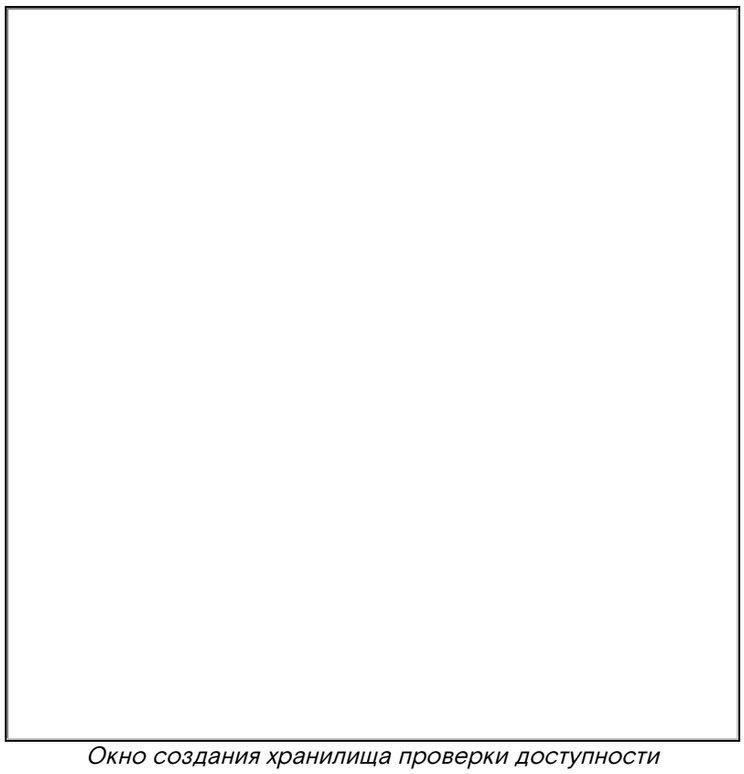
В противном случае система вернется в окно создания, с указанием причин невозможности его создания.

Назначение хранилища проверки доступности на гипервизоры

Перейдите: БАЗИС >> Инфраструктура >> Хранилище проверки доступности

Вызовите действие – «Назначить на гипервизоры».

В открывшемся окне (`StatS_assign`), путем переноса из левого в правый столбец, добавьте гипервизоры для хранилища проверки доступности.



Окно создания хранилища проверки доступности

Завершите процедуру создания нажатием кнопки [Сохранить].