

ООО «ТИОНИКС»

УТВЕРЖДЕН

RU.НРФЛ.00004 – 01 31 01-ЛУ

Эксплуатационная документация

Описание применения
облачной платформы «ТИОНИКС»
(TIONIX Cloud Platform)

RU.НРФЛ.00004 – 01 31 01-ЛУ

Листов 30

Москва, 2020

Аннотация

Программное обеспечение «Облачная платформа ТИОНИКС» — «TIONIX Cloud Platform» (далее - ПО «Тионикс», Облачная платформа) предназначено для создания инфраструктурных облачных сервисов и облачных хранилищ на основе комплекса проектов программного обеспечения с открытым кодом Openstack.

ПО «Тионикс» состоит из программных модулей (далее -- модулей Тионикс), которые обеспечивают:

- управление и контроль аппаратной средой;
- безопасность доступа к облачной среде виртуализации.

Использование технологии виртуализации позволяет оптимизировать организационные процессы, связанные с управлением IT-инфраструктурой, помещенной в облако.

Содержание

1 Общие сведения.....	5
1.1 Обозначение и наименование.....	5
1.2 Назначение ПО.....	5
1.3 Основные функциональные возможности.....	5
1.4 Используемые технологии.....	7
2 Структура и состав.....	10
2.1 Описание логической структуры.....	10
2.2 Перечень модулей.....	10
3 СПО OpenStack и ПО TIONIX.....	12
3.1 Используемые компоненты OpenStack.....	12
3.2 Функциональные связи компонентов OpenStack.....	14
3.3 Взаимодействие TIONIX с OpenStack.....	15
3.4 Сетевая инфраструктура OpenStack.....	16
3.4.1 Модель OpenStack Networking.....	16
3.4.2 Плагины.....	17
3.4.3 Архитектура Neutron.....	17
3.5 Вычислительная инфраструктура OpenStack.....	18
3.6 Инфраструктура хранения образов.....	19
4 Функциональное описание модулей.....	21
4.1 TIONIX.Autosetup.....	21
4.2 TIONIX.Client.....	21
4.3 TIONIX.NodeControl.....	21
4.4 TIONIX.Scheduler.....	22
4.5 TIONIX.Dashboard.....	23
4.6. TIONIX.Monitor.....	23

4.6 TIONIX.Agent.....	24
4.7 TIONIX.VDI*.....	24
4.7.1 VDIserver.....	25
4.7.2 VDIclient.....	25
5 Лицензирование.....	27
Термины, сокращения и определения.....	28

1 Общие сведения

1.1 Обозначение и наименование

Программное обеспечение для облачной платформы – ПО «Облачная платформа ТИОНИКС» (англ. Tionix Cloud Platform). Далее может применяться сокращенный термин – платформа ТИОНИКС или ОП.

1.2 Назначение ПО

ПО «Облачная платформа ТИОНИКС» предназначено для создания инфраструктурных облачных сервисов и облачных хранилищ, с применением компонентов технологического программного обеспечения с открытым исходным кодом (Open Source).

Модель предоставления инфраструктуры – IaaS – означает, что контроль со стороны поставщика услуги (облачного оператора) за устанавливаемым внутри облака программным обеспечением не предусмотрен. Контролируется только физическая и виртуальная составляющие инфраструктуры: управляющие и вычислительные узлы (УУ и ВУ), а также – сеть.

На базе [облачной платформы](#) может быть организована услуга «Виртуальный ЦОД» (VDC) – предоставление виртуализированных ресурсов ЦОД для конечных потребителей ИТ-услуг.

Облачная платформа как услуга позволяет потребителю услуги внедрять в инфраструктуру VDC высоконадежные (отказоустойчивые) ИТ-решения, необходимые для выполнения коммерческих или производственных задач, а также задач, связанных с территориально-административным управлением.

Использование технологии виртуализации и ПО «Виртуальные рабочие столы ТИОНИКС» для развертывания [VDI-решений](#) позволяет оптимизировать управление ИТ-инфраструктурой, связанное с многочисленными задачами системного администрирования в корпоративной среде.

1.3 Основные функциональные возможности

Облачная платформа виртуального дата-центра (VDC), организованная с применением облачной платформы, позволяет организовать объекты, составляющие

основу систем виртуализации, в единое целое – облачную инфраструктуру.

Принято различать следующие классы объектов, из которых построена облачная инфраструктура:

1. Сетевое взаимодействие (NETWORK).
2. Хранение данных (STORAGE).
3. Вычислительные мощности (COMPUTE).
4. Кластер управления (CONTROL).

Чтобы построить необходимые взаимосвязи между функциональными блоками виртуального ЦОД, относящимися к вышеперечисленным классам, платформа TIONIX предоставляет в распоряжение административные инструменты для создания объектов облачной инфраструктуры, выстраивания взаимосвязей между ними.

Платформа ТИОНИКС включает в себя проприетарные программные модули, которые обеспечивают:

- управление и контроль УУ и ВУ;
- безопасный доступ к облачным ресурсам (виртуальным машинам);
- создание и управление средой виртуализации рабочих столов.

Используя компоненты и инструментарий OpenStack [\[1\]](#), настраиваются инфраструктурные процессы, обеспечивающие идентификацию, аутентификацию, оркестрацию. Прямое влияние инструментов OpenStack на бизнес-логику облачных приложений исключено, не считая изменений в организации сетевого взаимодействия между объектами инфраструктуры.

Интерфейс пользователя и администратора облачной инфраструктуры доступен через веб-браузер, установленный на ПК или ТК.

Администратор или пользователь может использовать веб-доступ к виртуальным машинам, используя консоль. Кроме того, для пользователя могут быть доступны виртуальные рабочие столы (VDI). Доступ контролируется посредством брокера, проверяющего реквизиты доступа перед тем, как разрешить удаленное подключение с применением поддерживаемого терминального протокола.

Для настольных виртуальных машин поддерживается автоматизация входа в AD. Это позволяет администраторам использовать групповые политики для обеспечения единообразия настройки пользовательской рабочей среды.

Для улучшения таких эксплуатационных показателей как доступность и отказоустойчивость применяются технологии кластеризации? встроенные в референсную архитектуру «виртуального ЦОД» – эталонную модель развертывания облака.

1.4 Используемые технологии

В основе функциональности ОП лежит технология OpenStack – комплекс проектов свободного ПО, предоставляющих архитектурно полную модель функционально-интегрированных компонентов. Программное обеспечение OpenStack свободно используется для создания, развертывания и масштабирования безопасных и устойчивых публичных или частных облаков. В качестве среды функционирования могут быть использованы различные дистрибутивы Linux [2].

Большинство используемого при построении облачной инфраструктуры ПО интегрируется через библиотеки поддержки языка Python [3] на общесистемном уровне. Фактически, исполнительная система языка Python представляет собой технологию распределения рантайма на множество узлов.

Практически все службы OpenStack для хранения служебной информации используют реляционные базы данных. В качестве СУБД, поддерживающих SQL, могут использоваться широко распространенные OpenSource-проекты, однако в рамках текущей реализации VDC и референсной архитектуры используется MySQL или MariaDb [4].

Работа модуля OpenStack Horizon (Dashboard), позволяющего контролировать выделение и распределение ресурсов и выполнять операции по управлению облачной инфраструктурой, основана на использовании веб-фреймворка Django [5].

Архитектура OpenStack спроектирована таким образом, чтобы доступное физическое оборудование могло быть включено в платформу частного, публичного или гибридного облака, которое включает в себя:

- постоянное хранилище блочного уровня (Block-level Storage);
- ядро развертывания виртуальных машин и хранилище образов (Image Storage);
- механизмы аутентификации и авторизации (Identification);
- интегрированную сеть (Networking);
- распределенное объектное хранилище (Object Storage).

NETWORK

Для распределения потоков обмена информацией по сетевым каналам

используется открытая технология программно-определяемых сетей – коммутатор Open vSwitch [6] (далее – OVS). Коммутатор VLAN (Virtual Local Area Network), построенный на основе этой технологии, является логической структурой и может быть гибко подстроен под любое архитектурное решение. Интеграция ПО коммутатора OVS и OpenStack образуют единую, гибко модифицируемую интегрированную сеть.

STORAGE

Сетевой доступ к системам хранения данных осуществляется с помощью семейства драйверов (Tionix Storage Driver):

- драйвер Cinder: предоставляет возможность для работы с блочными устройствами Cinder на общем хранилище LVM без использования iSCSI;
- драйвер ML2: инфраструктура, позволяющая OpenStack Networking использовать множество сетевых технологий 2-го уровня, не зависящих от поставщика.

ML2 упрощает добавление поддержки новых сетевых технологий (2-го уровня) и обеспечивает динамическое предоставление VLAN/VXLAN на коммутаторах в среде OpenStack.

COMPUTE

Для распределения вычислительных ресурсов на уровне отдельных узлов используется виртуальная машина на основе ядра (KVM) (гипервизор), а также libvirt API [7]. Гипервизор тесно интегрирован с коммутатором Open vSwitch, что позволяет строить изолированные каналы управления виртуализацией, на которые не могут повлиять бизнес- приложения, развертываемые в инфраструктуре (IaaS-типа).

CONTROL

Сбой сервера управления (управляющий узел/контроллер) приведёт к тому, что поддерживаемые им приложения или сетевые сервисы будут недоступны до восстановления его работоспособности, выполняемого системным администратором или инженером.

Отказоустойчивая кластеризация исправляет эту ситуацию, перезапуская приложения на другой системе без вмешательства администратора (в случае обнаружения ошибок в аппаратном или программном обеспечении). Процесс перезапуска известен как *аварийное переключение*: ПО кластеризации может настроить узел перед запуском приложения на нём (например, импортировать и установить соответствующие файловые системы или перезапустить некоторые приложения).

Сноски

- [1] <https://ru.wikipedia.org/wiki/OpenStack>
- [2] <https://ru.wikipedia.org/wiki/Linux>
- [3] <https://www.python.org>
- [4] <https://dev.mysql.com/doc/>, <https://mariadb.com/docs/reference/>
- [5] <https://ru.wikipedia.org/wiki/Django>
- [6] https://ru.wikipedia.org/wiki/Open_vSwitch, <https://ru.wikipedia.org/wiki/VLAN>
- [7] <https://ru.wikipedia.org/wiki/Libvirt>

2 Структура и состав

Комплектность поставки [ПО «Тионикс»](#) зависит от потребностей Заказчика – облачного оператора.

Инструменты управления облачной инфраструктурой поставляются в виде TIONIX-модулей, развертываемых в окружении OpenStack.

2.1 Описание логической структуры

Отдельные функциональные компоненты программного комплекса реализованы в виде программных модулей ([Таблица 2.1](#)).

Логическая структура предполагает деление модульного ПО и других решений на следующие группы:

- ПО для создания виртуальной облачной платформы ([VDC](#));
- ПО для создания инфраструктур виртуальных Рабочих столов ([VDI](#));
- ПО для защиты средств виртуализации ([TVS](#));
- ПО для защиты виртуальных рабочих мест ([VDI Security](#)).

2.2 Перечень модулей

Ниже в табличном виде представлен перечень основных модулей, образующих программный комплекс – облачную платформу ТИОНИКС.

Наименования модулей (TIONIX.*) отличаются от наименований компонентов OpenStack ([Раздел 3.1](#)), определяющих базовые функциональные возможности управления облачной инфраструктурой.

Таблица 2.1 Перечень и характеристика модулей ПО «Тионикс»

№	Название модуля	Назначение (характеристика)	Группа
1.	TIONIX.NodeController	модуль, расширяющий функции платформы OpenStack, связанные с управлением аппаратными ресурсами облачной инфраструктуры – вычислительными узлами.	VDC

2.	TIONIX.Dashboard	модуль, расширяющий функции OpenStack Horizon; дополняет стандартный функционал графическими инструментами управления платформой TIONIX (администрирования).	—
3.	TIONIX.Monitor	модуль, расширяющий функции платформы OpenStack, связанные с получением статистических данных об основных характеристиках производительности виртуальных машин.	VDC
4.	TIONIX.Scheduler	модуль, обеспечивающий отложенный запуск функций остальных модулей TIONIX или платформы OpenStack.	VDC
5.	TIONIX.VDIclient	модуль, предоставляющий интерфейс десктопного клиента для работы в среде рабочего стола VDI машины.	VDI
6.	TIONIX.VDIserver	модуль управления виртуальными рабочими столами (VDI).	VDI
7.	TIONIX.Autosetup	модуль, обеспечивающий автоматическую установку и настройку модулей TIONIX, а также их обновление и удаление.	VDC
8.	TIONIX.Client	служебный модуль, необходим для предоставления доступа к функциональности модулей TIONIX.	VDC
9.	TIONIX.Agent	модуль, предоставляющий сервис для непосредственного управления узлами через RPC-вызовы.	VDC

3 СПО OpenStack и ПО TIONIX

СПО OpenStack имеет логический дизайн архитектуры, охватывающий практически все аспекты информационных технологий, связанные со сбором, накоплением и обработкой данных. Конечному пользователю предоставляются утилиты для работы из командной строки, а также – графические средства управления облачными ресурсами [1].

Все ключевые службы OpenStack обращаются в службу идентификации прежде чем позволить выполнение операций, затребованных пользователем/администратором. При этом используется единый программный интерфейс – OpenStack Identity API [2].

Сетевое взаимодействие между компонентами OpenStack подчинено единому концепт-дизайну [3].

ПО TIONIX в разной мере использует функциональность СПО Openstack на уровне компонентов [4].

3.1 Используемые компоненты OpenStack

Компоненты OpenStack имеют собственные имена, ассоциированные с проектами:

- Nova — контроллер вычислительных ресурсов;
- Glance — библиотека образов виртуальных машин;
- Cinder — служба работы с блочными устройствами хранения данных;
- Keystone — служба идентификации;
- Neutron — «подключение к сети как услуга» между интерфейсами устройств (vNIC), управляемыми другими сервисами OpenStack;
- Horizon — графический интерфейс администрирования;
- Heat — оркестратор;
- Ceilometer — средства сбора, нормализации и трансформации данных (данные предоставляются службами OpenStack);
- Swift — облачное файловое хранилище.

Назначение служб (компонентов) Openstack:

Neutron ([Network Service](#)) – предоставляет Сеть как сервис для остальных служб OpenStack и виртуальных машин. Включает в себя пользовательское API для определения

сетей и присоединения к ним VM. Служба функционально расширяема с помощью плагинов;

Nova ([Compute Service](#)) – отвечает за создание, запуск, перезапуск, остановку виртуальных машин, и т.д. Компонент позволяет осуществлять управление вычислительными ресурсами облака и может работать с различными технологиями и системами виртуализации (гипервизорами) на уровне операционной системы;

Glance ([Image Service](#)) – позволяет обнаруживать, регистрировать и извлекать образы виртуальных машин;

Cinder ([Block Storage Service](#)) – блочное хранилище;

Keystone ([Identity Service](#)) – служба идентификации интегрирует управление аутентификацией, авторизацию и каталог служб. Другие службы OpenStack используют службу идентификации для проверки подлинности пользователей и обнаружения других служб (в рамках развертывания);

Ceilometer – инструмент для сбора различных статистических данных в облаке OpenStack. Основной целью проекта является мониторинг нагрузки и измерения потребления ресурсов клиентами;

Swift ([Object Storage Service](#)) – это полностью распределенное «безграничное» хранилище, которое характеризуется отказоустойчивостью и высокой надежностью.

3.2 Функциональные связи компонентов OpenStack

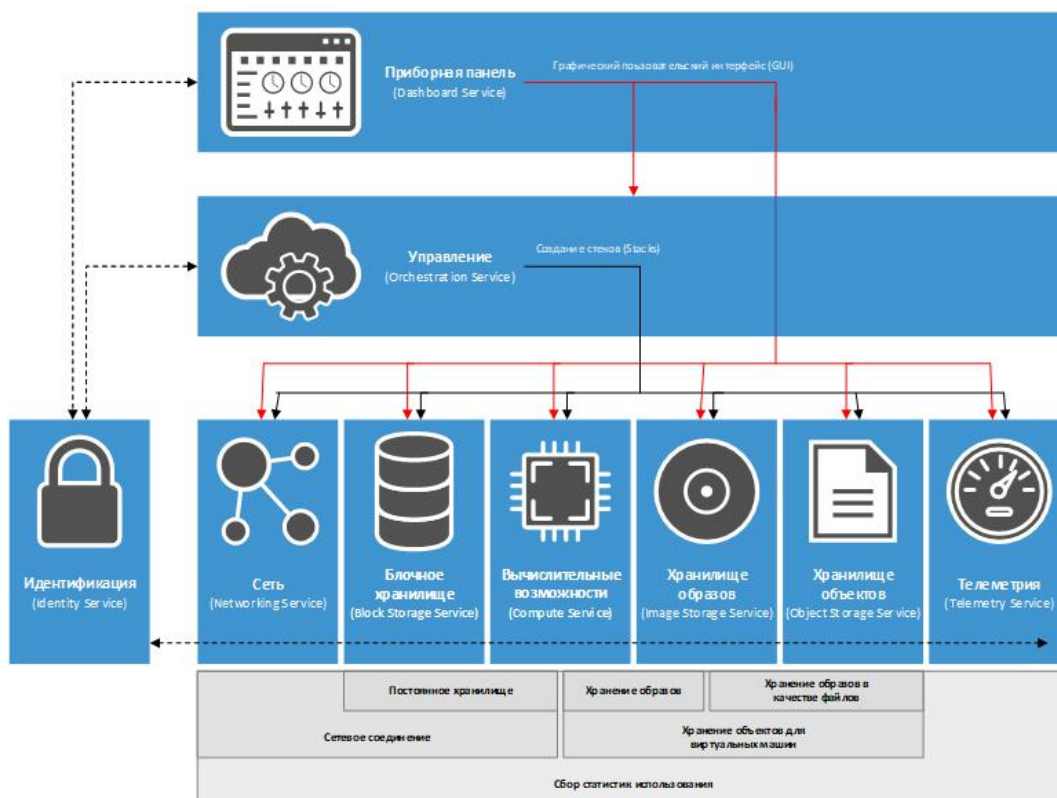


Рис. 3.1 Архитектура облачной платформы OpenStack

При создании облачной инфраструктуры (Рис. 3.1) могут, в различной мере, использоваться функции следующих компонентов OpenStack [5]:

- панель управления (Dashboard Service, [Horizon](#));
- служба идентификации (Identity Service, [Keystone](#));
- сетевая служба (Networking Service, [Neutron](#));
- вычислительная служба (Compute Service, [Nova](#));
- служба хранилищ образов (Image Storage Service, [Glance](#));
- служба блочных хранилищ (Block Storage Service, [Cinder](#));
- управление/оркестрация (Orchestration Service, [Heat](#));
- хранилище объектов (Object Storage Service, [Swift](#)).
- служба телеметрии (Telemetry Service, [Ceilometer](#)).

Примечание. Названия служб OpenStack указаны со ссылками на веб-ресурсы.

3.3 Взаимодействие TIONIX с OpenStack

Функциональность модулей TIONIX основана на использовании программных интерфейсов СПО OpenStack [6]. Прежде чем обратиться к какой-либо службе, модуль выполняет запрос на аутентификацию (OpenStack Identity) с указанием реквизитов. В ответ модуль получает токен.

Управляющие функции выполняют модули NodeControl и Scheduler. [TIONIX.NodeControl](#) – ключевой модуль облачной платформы. Он обеспечивает централизованное управление аппаратными и виртуальными ресурсами облачной инфраструктуры. [TIONIX.Scheduler](#) – планировщик работы вычислительных узлов – является вспомогательным.

В повседневной работе администратора облачной инфраструктуры возникает необходимость вызова определенного функционала OpenStack из командной строки. Такое взаимодействие обеспечивается с помощью так называемого клиента – Openstack Client [7].

Ниже представлена таблица взаимосвязей между модулями облачной платформы «Тионикс» и компонентами OpenStack.

Таблица 3.1 Взаимосвязи между модулями ОП и компонентами OpenStack

№	Модуль ПО «Тионикс»	Компоненты OpenStack
	TIONIX.NodeControl	Neutron, Nova
	TIONIX.Dashboard	Cinder, Keystone, Neutron, Nova, Glance
	TIONIX.Monitor	Ceilometer, Gnocchi
	TIONIX.Scheduler	Cinder, Keystone, Nova
	TIONIX.Autosetup	Horizon
	TIONIX.Client	Horizon
	VDIclient	–
	VDIserver	–
	TIONIX.Agent	–

3.4 Сетевая инфраструктура OpenStack

Если задействовать существующую (физическую) сетевую инфраструктуру для выделения IP-адресов и для передачи данных между узлами, то в мультитенантной среде, содержащей несколько арендаторов, это приведет к тому, что имеющаяся система управления сетью будет неспособна эффективно и надежно изолировать трафик между пользователями облачной инфраструктуры.

В этом случае облачная среда нуждается в построении тщательно продуманного стека сетевого управления, который обрабатывал бы все связанные с сетью запросы. Для решения такого рода задач создан абстрактный уровень под названием OpenStack Networking и поддерживающий компонент – сетевая служба (Networking Service). Эта служба, называемая Neutron [8], содержит ассортимент плагинов, обеспечивающих интеграцию с другими сетевыми сервисами.

3.4.1 Модель OpenStack Networking

OpenStack Networking базируется на простой модели абстракций, используемой для описания сетевых ресурсов (виртуальные сети, подсети, порты).

Сеть представляет собой изолированный сегмент уровня 2 (L2), аналогичный виртуальной локальной сети (VLAN) в мире физических сетей.

Подсеть – это совокупность IP-адресов (v4 или v6) и ассоциированных с ними конфигураций. IP-адреса из этого пула платформа OpenStack может назначать виртуальным машинам. Каждая подсеть специфицируется как CIDR-диапазон IP-адресов и должен быть ассоциирована с какой-либо сетью. Помимо подсети, арендатор может при желании указать шлюз, список DNS-серверов и набор маршрутов хостов. Все экземпляры виртуальных машин в этой подсети автоматически унаследуют эту конфигурацию.

Порт – это точка подключения к виртуальному коммутатору. Экземпляр виртуальной машины способен подключить свой сетевой адаптер к виртуальной сети через порт. После создания порта он получает фиксированный IP-адрес от одной из указанных подсетей. После высвобождения порта все выделенные ему IP-адреса также высвобождаются и возвращаются в пул адресов. Кроме того, платформа OpenStack позволяет задавать MAC-адреса, которые должен использовать интерфейс.

OpenStack Networking позволяет создавать плагины [9], обеспечивающие поддержку расширенных сетевых возможностей, таких как L2/L3-туннелирование и сквозная поддержка качества обслуживания (QoS). Сторонние поставщики могут

создавать, кроме собственных плагинов, различные сетевые сервисы (балансировщики нагрузки, виртуальные частные сети, брандмауэры и т. д.), включаемые в сети арендаторов OpenStack [10].

3.4.2 Плагины

Архитектура, содержащая плагины, предлагает администратору облачной среды высокую степень гибкости при настройке сетевой конфигурации.

Выбор определенного плагина осуществляется администратором или IT-архитектором, который способен оценить предлагаемые возможности и согласовать их с требованиями конкретной установки (референсной архитектуры).

Некоторые плагины OpenStack Networking могут использовать базовые механизмы Linux (IP-таблицы и VLAN-сети).

3.4.3 Архитектура Neutron

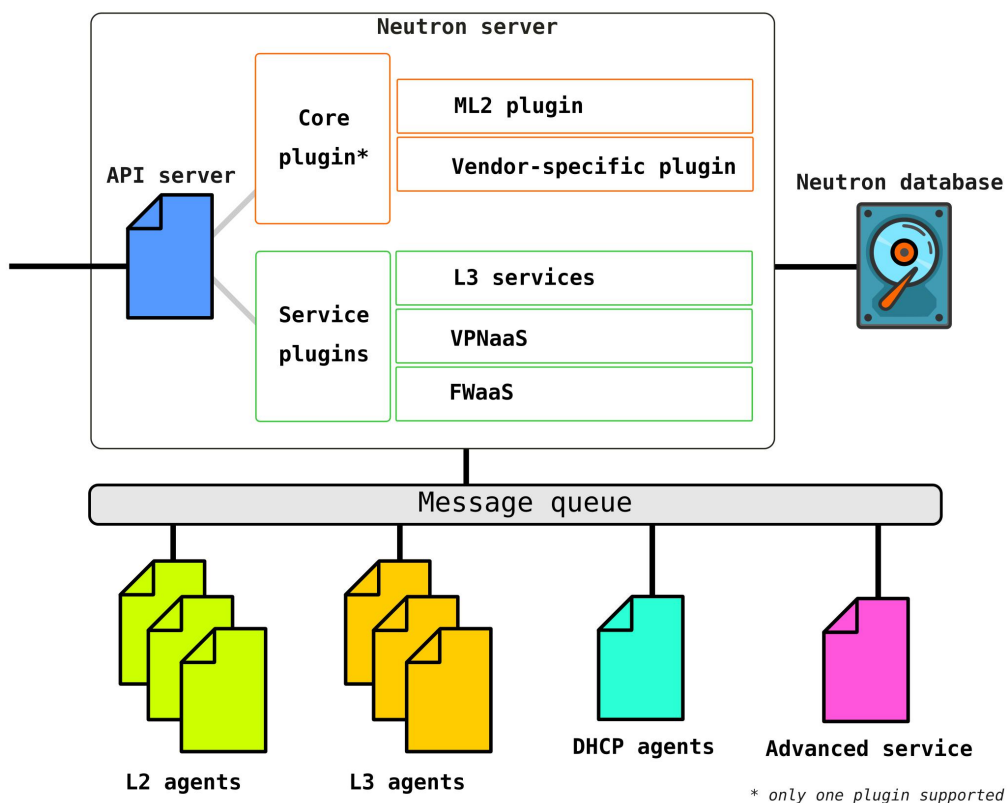


Рис. 3.2 Архитектура сетевой службы Neutron

neutron-server – это основной серверный процесс OpenStack Networking. ПО сервера реализовано в виде демона (Core plugin* на Рис. 3.2), перенаправляющего запросы из API OpenStack Networking в сконфигурированный плагин (ML2 или Vendor-specific).

Кроме сервера, в состав OpenStack Networking включены три агента (L2/L3/DHCP), которые могут взаимодействовать с основным процессом. Взаимодействие происходит через очередь сообщений (Message queue) или через стандартный API-интерфейс OpenStack Networking.

Агент **neutron-dhcp-agent** – предоставляет DHCP-сервисы всем сетям арендатора.

Агент **neutron-l3-agent** – поддерживает функциональность L3/NAT, чтобы обеспечить виртуальным машинам (в сетях арендаторов) доступ к внешней сети.

Необязательный агент, определяемый плагином (**neutron-*-agent**), выполняет конфигурирование локального виртуального коммутатора на каждом гипервизоре.

Интеграция с компонентом OpenStack Compute (Nova) является более специфичной. При запуске виртуального экземпляра службой Nova производится связывание с компонентом OpenStack Networking. Цель – включения каждого интерфейса виртуальной сети в соответствующий порт.

3.5 Вычислительная инфраструктура OpenStack

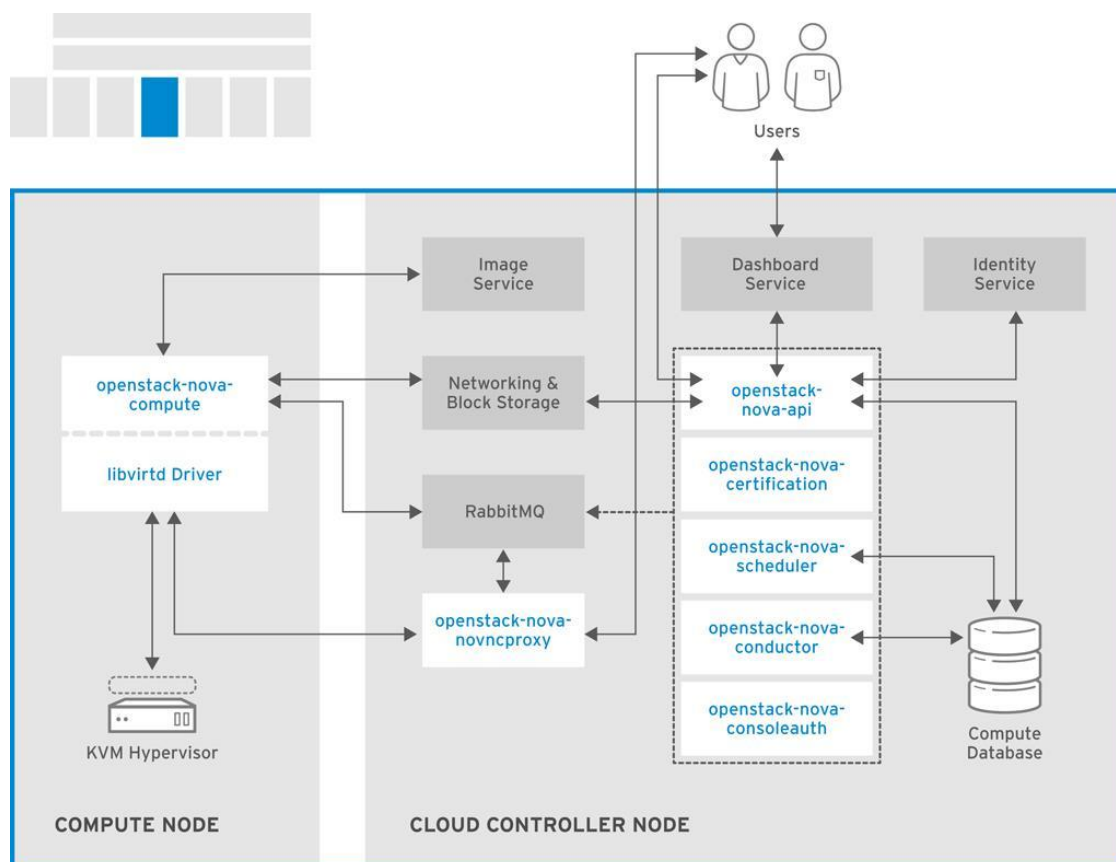


Рис. 3.3 Принцип организации вычислительной инфраструктуры (OpenStack)

Управление вычислительными ресурсами реализовано с помощью гипервизора (KVM Hypervisor на [Рис. 3.3](#)).

Служба Nova (Compute Service) обеспечивает управление экземплярами виртуальных машин, обращаясь к гипервизору за выполнением API-команд: запуск VM; останов VM; и т.п.

OpenStack Nova состоит из следующих компонентов (системных служб):

openstack-nova-api – отвечает за обработку пользовательских вызовов API;

openstack-nova-scheduler – планировщик, получает из очереди запросы на запуск VM и выбирает узел для запуска;

openstack-nova-conductor – выступает в качестве посредника между базой данных и nova-compute, позволяет осуществлять горизонтальное масштабирование;

openstack-nova-novncproxy – выступает в роли VNC--прокси и позволяет подключаться к консоли VM при помощи веб-браузера;

openstack-nova-consoleauth – отвечает за авторизацию для предыдущего сервиса;

openstack-nova-placement-api – отвечает за отслеживание списка ресурсов и их использование;

openstack-nova-compute – демон, управляющий виртуальными машинами через API гипервизора.

Примечание. Демон openstack-nova-compute, как правило, запускается на вычислительных узлах, где располагается гипервизор.

Компоненты OpenStack Compute (Nova) взаимодействуют со следующими инфраструктурными компонентами кластера:

- брокером сообщений RabbitMQ и СУБД MySQL;
- сервисом кэширования данных в оперативной памяти (memcached).

Выбор гипервизора для запуска VM осуществляется планировщиком [\[11\]](#), согласно весам узлов и после применения фильтров, таких как: необходимый объем оперативной памяти, определенная зона доступности, и т.п.

3.6 Инфраструктура хранения образов

Компонент Glance управляет образами в кластере OpenStack, но не отвечает за их

фактическое хранение. Glance обеспечивает абстрагирование нескольких технологий хранения, в диапазоне от простых файловых систем до систем хранения объектов, таких как проект OpenStack Object Storage (проект Swift).

Помимо реальных образов дисков, Glance отвечает за хранение метаданных и сведений о состоянии, описывающих образы.

OpenStack [Image Store](#) – это центральный репозиторий виртуальных образов. Он позволяет пользователям и другим проектам сохранять как публичные, так и частные образы, к которым эти пользователи/проекты могут обращаться (с целью запуска экземпляров).

Пользователи/проекты могут запросить список доступных образов и получить их конфигурационную информацию, а затем использовать образы для запуска *экземпляров Nova* (инстансов). Кроме того, можно сделать моментальные снимки исполняющихся экземпляров, с целью создания резервных копий виртуальных машин и сохранения их (промежуточных) состояний.

Сноски

- [1] <https://docs.openstack.org/arch-design/design.html>
- [2] <https://docs.openstack.org/ocata/config-reference/identity.html>
- [3] <https://docs.openstack.org/arch-design/design-networking/design-networking-concepts.html>
- [4] <https://www.openstack.org/software/project-navigator/openstack-components>
- [5] <https://docs.openstack.org/ocata/ru/install-guide-rdo/common/get-started-conceptual-architecture.html>
- [6] <https://docs.openstack.org/api-quick-start/api-quick-start.html>
- [7] <https://docs.openstack.org/python-openstackclient/queens/cli/commands.html>
- [8] <https://ru.bmstu.wiki/OpenStack#Neutron>
- [9] <https://docs.openstack.org/devstack/latest/plugins.html>

4 Функциональное описание модулей

Модульная архитектура ПО TIONIX позволяет адаптировать типовое решение под частные нужды Заказчика, развертывая рабочую платформу (VDC) на предоставленной площадке (ЦОД) по готовому сценарию, учитывающему референсную архитектуру типовой облачной инфраструктуры.

Ниже приведено текстовое описание потребительских свойств, присущим каждому из модулей TIONIX в-отдельности. Полный перечень модулей приведен в таблице [Перечень и характеристика модулей ПО «Тионикс»](#) (см. [описание структуры и состава](#)).

4.1 TIONIX.Autosetup

Установка TIONIX-модулей может производиться с помощью утилиты **tnx_autosetup**, которая становится доступна на управляющем узле (контроллере) сразу после установки данного модуля.

Для ознакомления с порядком установки модулей следует перейти [по ссылке](#).

4.2 TIONIX.Client

Модуль расширяет возможности консольной утилиты **openstack** дополнительными командами. Данный модуль необходим для работы всех остальных модулей TIONIX и должен быть заранее установлен и настроен.

Для получения списка доступных команд следует вызвать справку об использовании:

```
openstack tnx --help
```

4.3 TIONIX.NodeControl

Модуль TIONIX.NodeControl предоставляет доступ к вычислительным узлам, используемым виртуальными машинами (ВМ) облачной инфраструктуры. Кроме того, модуль обеспечивает необходимое взаимодействие с системой бесперебойного питания.

Модуль использует БД, реализующую модель данных OpenStack [1], с которой может взаимодействовать служба [openstack-nova-compute](#) (Nova, Compute Service), которая сосредоточена на таких функциях как:

- обработка запросов на создание VM;
- соединение VM с внешним миром;
- контроль за работоспособностью VM;
- распределение нагрузки на физические машины и каналы связи;
- детерминированная реакция на сбои.

Модуль TIONIX.NodeControl позволяет осуществлять:

- назначение расширенных атрибутов для вычислительного узла (инвентарный номер, локация);
- управление PXE-образами вычислительных узлов;
- мониторинг состояния вычислительных узлов в реальном времени и запуск автоматической эвакуации;
- создание и управление резервными узлами;
- действия над программно-определяемыми хранилищами (SDS) и блоками, в случае настроенной системы Ceph;
- сбор информации о блочных хранилищах Cinder и управление локальным общим хранилищем.

Модуль содержит консольные утилиты, интегрируемые в окружение управляющего узла:

- tnx-node-control-api;
- tnx-node-control-node-syncer;
- tnx-node-control-node-tracker;
- tnx-node-control-worker;
- tnx-node-control-nova-listener;
- tnx-node-control-drs-trigger;
- tnx-node-control-storage-syncer.

4.4 TIONIX.Scheduler

С помощью модуля TIONIX.Scheduler – планировщика задач – другие модули TIONIX могут выполнять запуск определенных задач (в том числе – по расписанию). Непосредственно сами задачи выполняет модуль [TIONIX.NodeControl](#).

Задачи выстраиваются в очередь, имеют определенную периодичность запуска,

различные приоритеты запуска. Планирование задач доступно для модулей:

- [TIONIX.Client](#);
- [TIONIX.Dashboard](#);
- [TIONIX.NodeControl](#);
- [TIONIX.VDIserver](#);

4.5 TIONIX.Dashboard

Модуль TIONIX.Dashboard характеризуется как «приборная панель» [2], которая (по умолчанию) предоставляет доступ к:

- проектам, организуемым в облаке;
- функциям администрирования;
- настройке параметров идентификации;
- параметрам среды (ТИОНИКС).

Кроме того, веб интерфейс модуля позволяет осуществлять консольный доступ (просмотр) к VDI-машинам. Это возможно только при установленном и настроенном модуле TIONIX.VDIserver ([Раздел 4.8.1](#)).

4.6. TIONIX.Monitor

Модуль TIONIX.Monitor осуществляет предоставление статистики о работе виртуальных машин, в виде фактических показателей:

- процента использования центрального процессора;
- процента использования оперативной памяти;
- количества запросов на чтение/запись с диска;
- количества запросов на прием/отправку пакетов (по сети);

Модуль позволяет осуществлять интеграцию с некоторыми внешними системами.

Мониторинг состояния виртуальных машин предназначен для обеспечения отказоустойчивости и позволяет:

1. Производить восстановление работы виртуальных машин.
2. Безопасно вывести из эксплуатации вычислительный узел (выключить питание).

Восстановление производится после поступления сигнала о недоступности или некорректности работы ВМ.

Перед выключением осуществляется перенос всех виртуальных машин, расположенных на выключаемом вычислительном узле, на другие доступные ВУ.

4.6 TIONIX.Agent

Модуль TIONIX.Agent устанавливается на вычислительные узлы и необходим для корректной работы следующего функционала:

- включения и выключения режима динамического конфигурирования компонентов (DCC) на вычислительных узлах;
- включения и выключения механизма SNMP на вычислительном узле;
- включения и выключения доступа к гипервизору (по протоколу SSH).

4.7 TIONIX.VDI*

VDI – это инфраструктура виртуальных рабочих столов (программное решение), в рамках которой организовывается персональный виртуальный Рабочий стол, закрепленный за конкретным пользователем [3].

Виртуализация рабочих мест – концепция, в которой данные любого корпоративного сотрудника хранятся централизованно, а АРМ сотрудника состоит из виртуальной машины (вместо ПК) и подключаемый к ней «тонкий клиент» (микрокомпьютер) [4]. Эта концепция востребована для распределения бизнес-приложений, при этом на жестком диске отдельно взятого АРМ не хранятся бизнес-данные.

Администратор сервера создает виртуальное рабочее место с отдельным набором приложений, программ, документов и доступов и это всё хранится на сервере в (виртуальном) ЦОД. Подключение АРМ сотрудника к серверу выполняется через сеть и аппаратную прослойку – «тонкий клиент».

Предусмотрено хранение (в облаке) типового виртуального рабочего стола, созданного на основе «золотого образа».

Допускается поочередное предоставление одного виртуального рабочего места нескольким пользователям.

Функциональные возможности VDI:

- поддержка популярных пользовательских (гостевых) операционных систем;
- поддержка удаленной работы через тонкие клиенты;

- поддержка кэширования (на стороне клиента);
- поддержка работы с несколькими дисплеями;
- поддержка работы с USB-устройствами;
- поддержка периферийных устройств (принтеры, сканеры);
- интеграция с Microsoft Active Directory (AD/LDAP).

Благодаря AD/LDAP администратор может разворачивать программное обеспечение на множестве виртуальных узлов через групповые политики или посредством System Center Configuration Manager, устанавливать обновления ОС (Windows), прикладного и серверного ПО на всех компьютерах в сети, используя Службу обновления Windows Server [5].

4.7.1 VDIserver

VDI-машина – виртуальная машина, хранящаяся вместе с образом гостевой ОС в облаке. Она содержит среду Рабочего стола и некоторое количество учетных записей, позволяющих выполнять вход и осуществлять определенные операции в установленных и настроенных приложениях, с использованием графического интерфейса пользователя «тонкого клиента».

Модуль TIONIX.VDIserver предоставляет следующие способы доступа к VDI-машине:

1. Веб-доступ из браузера ([TIONIX.Dashboard](#)).
2. Доступ при помощи [клиента VDI](#).

Подробные сведения о настройке и эксплуатации инфраструктуры VDI изложены в документе Руководство по эксплуатации. ПО TIONIX VDI.

4.7.2 VDIclient

Для доступа к виртуальному рабочему столу (VDI-машине) в ПК или ТК устанавливается и запускается ПО клиента VDI, подробные сведения об использовании которого изложены в документе Руководство пользователя. ПО TIONIX VDI.

Модуль [TIONIX.VDIclient](#), содержащий ПО клиента, является нелицензируемым компонентом и взаимодействует с модулем сервера VDI, функционирующим в облачной инфраструктуре ([TIONIX.VDIserver](#)).

С помощью текстового или графического интерфейса VDI-клиента выполняются:

- авторизация пользователя (с выданными реквизитами доступа или смарт-картой);
- подключение к удаленному рабочему столу (по терминальному протоколу).

Если с пользователем сопоставлено несколько проектов, то перед подключением происходит запрос выбора (одного из возможных вариантов).

Сноски

- [1] <https://docs.openstack.org/shade/latest/user/model.html>
- [2] <https://www.openstack.org/software/releases/queens/components/horizon>
- [3] <https://www.vmware.com/ru/topics/glossary/content/virtual-desktop-infrastructure-vdi.html>
- [4] <https://selectel.ru/blog/vdi-technology-review/>
- [5] https://ru.m.wikipedia.org/wiki/Active_Directory

5 Лицензирование

Лицензирование модулей – это технология, определяющая легальное использование и распространение программного комплекса TIONIX, защищенного авторским правом (Свидетельство о государственной регистрации программы для ЭВМ №2018617974 от 5 июля 2018 г.).

Лицензия разрешает получателю использовать функциональность модуля в полной мере, в течение определенного срока действия лицензии. По истечении срока (действия лицензии) функционал модуля ограничивается, но информация и данные, созданные в течение этого срока, сохраняются.

Для восстановления функциональности (лицензируемого) модуля достаточно установить актуализированную лицензию.

Термины, сокращения и определения

Термин	Определение
АРМ	автоматизированное рабочее место
БД	база данных
ВМ	виртуальная машина
ВУ	вычислительный узел
ЛВС	локальная вычислительная сеть
ОП	облачная платформа
ПО	программное обеспечение
СВТ	средство вычислительной техники
СПО	свободное программное обеспечение
СУБД	система управления базами данных
УУ	управляющий узел
ЦОД	центр обработки данных, дата-центр
AD	(англ. Active Directory) LDAP-совместимая реализация службы каталогов корпорации Microsoft для операционных систем семейства Windows Server
API	(англ. Application Program Interface) набор интерфейсов вызова, посредством которых программное приложение «общается» с операционной системой (ОС) и другими программами
CIDR	(англ. Classless Inter-Domain Routing) метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя жёсткие рамки классовой адресации. Использование этого метода позволяет экономно использовать ограниченный ресурс IP-адресов, поскольку возможно применение различных масок подсетей к различным подсетям
DHCP	(англ. Dynamic Host Configuration Protocol) Протокол, реализованный по модели «клиент-сервер», в которой DHCP-клиенты запрашивают у DHCP-сервера конфигурационные данные, такие как: IP-адрес; маршрут по умолчанию; адрес одного или более серверов DNS

Термин	Определение
DNS	(англ. Domain Name System) компьютерная распределённая система для получения информации о доменах. Часто используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах (для работы сетевых протоколов в домене)
IaaS	инфраструктура как услуга (технология предоставления ЦОД-ресурсов)
iSCSI	(англ. Internet Small Computer System Interface) – протокол, основанный на TCP/IP и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами
IT	информационные технологии
LAN	(англ. Local Area Network) локальная вычислительная сеть
LVM	(англ. Logical Volume Manager) подсистема нижнего уровня операционных систем Linux, позволяющая использовать разные области одного жёсткого диска и/или области с разных жёстких дисков как один (логический) том
OSI	сетевая (эталонная) модель взаимодействия открытых систем
RPC	(англ. Remote Procedure Call) класс технологий, позволяющих компьютерным программам вызывать функции или процедуры в другом адресном пространстве (на удалённых компьютерах)
SDS	(англ. Software Defined Storage) программное решение, обеспечивающее создание сети хранения данных на неспециализированном оборудовании массового класса – группе серверных узлов архитектуры x86-64 под управлением операционных систем общего назначения (FreeBSD, Linux, Windows)
SNMP	протокол простого сетевого управления
SQL	стандартизованный язык запросов (к СУБД)
SSH	протокол безопасного (удаленного) доступа к консоли
VDC	виртуальный дата-центр, разновидность ЦОД, формируемого в виде IaaS (облачной инфраструктуры)
VDI	технология, позволяющая создавать виртуальную ИТ-инфраструктуру Рабочих столов (множество виртуальных машин) на базе одного СВТ серверного типа

Термин	Определение
VNC	система удалённого доступа к рабочему столу компьютера, использующая RFB (Remote FrameBuffer) – простой протокол удаленного доступа к графическому интерфейсу пользователя
VLAN	(англ. Virtual Local Area Network) «виртуальная» локальная компьютерная сеть, наделенная теми же свойствами, что и физическая ЛВС. VLAN позволяет образовывать из конечных станций (нодов) сгруппированные структуры, даже если они не находятся в одной физической сети (территориально распределены)
VXLAN	(англ. Virtual eXtensible LAN) технология сетевой виртуализации, созданной для решения проблем масштабируемости в больших системах облачных вычислений

