Программное обеспечение «Базис.Cloud». Руководство по эксплуатации

RU.НРФЛ.00004-01.97.01

Москва

12/14/2022

Содержание

1 Аннотация

Настоящий документ предназначен для технического администратора ПО и содержит инструкции по выполнению работ, необходимых для эксплуатации ПО.

2 Назначение руководства

Настоящее руководство по техническому обслуживанию содержит инструкции по выполнению следующих работ:

- сопровождение и обслуживание ПО;
- диагностику, локализацию и устранение проблем.

3 Перечень эксплуатационных документов

Дополнительно к настоящему документу технические администраторы должны использовать следующие документы:

- «ПО «Базис.Cloud». Руководство по установке. RU.НРФЛ.00006-01.96.01;
- «ПО «Базис.Cloud». Руководство администратора RU.НРФЛ.00006-01.95.01.

4 Идентификационные данные документа

Идентификационные данные ПО	**Программа для ЭВМ «Базис.Cloud»**
Название документа	«ПО «Базис.Cloud». Руководство по эксплуатации»
Обозначение документа	RU.НРФЛ.00006-01.97.01
Автор документа	ООО «БАЗИС»

5 Требования к составу и квалификации обслуживающего персонала

Системный инженер – должностное лицо, служебная деятельность которого обеспечивает качественную и безопасную эксплуатацию оборудования ЦОД или виртуального ЦОД – облачой платформы, построенной на основе ПО «Базис.Cloud» – после внедрения (ввода в эксплуатацию).

Администратор ОП – должностное лицо, служебная деятельность которого связана с эксплуатацией программных продуктов и стороннего ПО, используемого при создании среды функционирования: ОС Linux, Python3, OpenStack и др.

Системный инженер должен иметь навыки проектирования или настройки аппаратных и программных конфигураций компьютерных сетей, обслуживания локальных вычислительных сетей. Кроме того, он может быть ответственен за организацию защиты информации и производить установку антивирусов и другого программного обеспечения, обновление ПО. Полезным будет также навык анализа затрат на системное обслуживание, составление отчетов и поиск способов оптимизации расходов.

Оперативный персонал (системный инженер), осуществляющий манипуляции с оборудованием на площадке, должен иметь допуск к эксплуатации электроустановок до 1000В. Категория допуска должна быть согласована со службами эксплуатации ЦОД.

Обычными задачами системного администратора, в зависимости от инфраструктуры, являются контроль работы компьютерных программ и устранение ошибок в их работе, разовая диагностика/ремонт ПК и другой офисной техники.

Системный (облачный) администратор должен уметь использовать множество утилит и инструментов администрирования 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.

Для выполнения задач по сопровождению облачной платформы, построенной на основе ПО «Базис.Cloud», необходимо иметь опыт работы, связанный с системным администрированием серверного оборудования, а также понимать основные принципы резервного копирования и восстановления данных.

Деятельность системного инженера регулируется и контролируется отделом информационной безопасности, а также внутренними регламентами предприятия, нацеленными на обеспечение безопасности данных и соблюдение конфиденциальности.

6 Описание и работа ПО

6.1 Назначение ПО

Программное обеспечение «Базис.Cloud» (далее ПО «Базис.Cloud») предназначено для создания инфраструктурных облачных сервисов и облачных хранилищ на основе комплекса проектов программного обеспечения с открытым кодом Openstack.

ПО «Базис.Cloud» включает в себя модули «TIONIX», которые обеспечивают:

- управление и контроль аппаратной средой;
- управление виртуальной средой для размещения виртуального ЦОД;
- создание и управление средой виртуальных рабочих мест (VDI);
- безопасность доступа к облачной среде виртуализации.

Использование технологии виртуализации и управления виртуальными рабочими столами (VDI) позволяет оптимизировать управление IT- структурой.

6.2 Структура ПО

ПО «Базис.Cloud» на уровне компонентов использует функциональность СПО Openstack. 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.

При создании облачной инфраструктуры ПО «Базис.Cloud» в различной мере использует функции следующих компонентов OpenStack:

- панель управления (Dashboard – 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);
- служба телеметрии (Telemetry Service – Ceilometer).

Для интеграции ПО «Базис.Cloud» и СПО «OpenStack» используется функциональность модулей TIONIX основаная на использовании программных интерфейсов OpenStack. Прежде чем обратиться к какой-либо службе, модуль выполняет запрос на аутентификацию (OpenStack Identity) с указанием реквизитов. В ответ, модуль (служба TIONIX) 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.

В таблице 2 представлены взаимосвязи между программными модулями ПО «Базис.Cloud» и компонентами OpenStack.

Таблица 2 - Взаимосвязи между программными модулями ПО «Базис.Cloud» и компонентами OpenStack

№	Программный модуль	Компоненты OpenStack
1	TIONIX.NodeControl	Neutron, Nova
2	TIONIX.Dashboard	Cinder, Keystone, Neutron, Nova, Glance
3	TIONIX.Monitor	Ceilometer, Gnocchi
4	TIONIX.Scheduler	Cinder, Keystone, Nova

№	Программный модуль	Компоненты OpenStack
5	TIONIX.Client	Horizon
6	TIONIX.Agent	--

6.3 Функциональные возможности

6.3.1 Функциональное описание модулей

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

Ниже приведено текстовое описание потребительских свойств, присущих каждому из модулей ПО «Базис.Cloud» в отдельности.

Для получения доступа к функциональности ключевых модулей применяется лицензирование.

Client

Модуль TIONIX.Client необходим для работы всех остальных модулей TIONIX и должен быть установлен и настроен в первую очередь.

Функциональность модуля включает:

- REST API (доступен через токен);
- консольные утилиты;
- дополнения к интеграции с LDAP;
- исправления к интеграции с Cinder.

В составе модуля содержится модифицированная реализация identity-драйвера для сервисов LDAP, называемая драйвером tnx_ldap. По умолчанию используется стандартный драйвер LDAP Keystone.

Модуль содержит исправления для Cinder, решающие вопрос живой миграции виртуальной машины при наличии блочных устройств на базе протоколов iSCSI и FibreChannel.

Модуль расширяет возможности консольной утилиты openstack дополнительными командами.

NodeControl

Модуль TIONIX.NodeControl предоставляет доступ к вычислительным узлам (ВУ), используемым виртуальными машинами (ВМ) облачной инфраструктуры. Для взаимодействия с ВУ используется БД, реализующая модель данных OpenStack.

Служба вычислений – openstack-nova-compute – сосредоточена на следующих функциях:

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

TIONIX.NodeControl позволяет осуществлять:

– назначение расширенных атрибутов для вычислительного узла (инвентарный номер, локация);

– управление PXE-образами вычислительных узлов;

– мониторинг состояния вычислительных узлов и запуск автоматической эвакуации;

– создание и управление резервными узлами;

– действия над программно-определяемыми хранилищами (SDS) и блоками, если настроен Ceph;

– сбор информации о блочных хранилищах Cinder и управление локальным общим хранилищем.

Кроме того, модуль обеспечивает связь со средством управления питанием, а также содержит консольные утилиты, интегрируемые в системное 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.

Scheduler

С помощью планировщика задач – модуля TIONIX.Scheduler – другие модули могут выполнять запуск определенных задач, в том числе – по расписанию. Непосредственно сами задачи выполняет модуль TIONIX.NodeControl. Задачи выстраиваются в очередь, имеют определенную периодичность и различные приоритеты запуска.

Планирование задач доступно для модулей:

- Client;
- Dashboard;
- NodeControl;
- VDIserver.

Monitor

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

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, например – с Zabbix.

Мониторинг состояния виртуальных машин предназначен для обеспечения отказоустойчивости и позволяет:

- производить восстановление работы виртуальных машин.
- 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.

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).

PointMeter

Модуль TIONIX.PointMeter предназначен для рассылки данных статистического учета и должен быть обязательно установлен в составе ПО «Базис.Cloud».

Модуль осуществляет сбор статистики об использовании вычислительных ресурсов за отчётный период, которая используется для оплаты использования лицензий ПО «Базис.Cloud» и «Базис.Workplace» по сервисной модели (OPEX). Использование ПО оценивается на основании ежемесячных отчетов потребления.

Модуль привязывается к интерфейсу управления (TIONIX.Dashboard), и настраивается для передачи почтовых сообщений для автоматической рассылки отчетов (E-mail), в соответствии с настроенным расписанием рассылки.

Алгоритм сбора статистики (по умолчанию):

- ежедневно, с периодичностью 1 час система получает данные о выделенных вычислительных ресурсах всех виртуальных машин в рамках проекта. В подсчёте индекса использования ресурсов используется информация о времени жизни всех виртуальных машин внутри проекта (реализация OpenStack);

- в конце отчётного периода суммируются общие показатели использования ресурсов В расчётах используются показатели по каждому проекту, за все время его существования и с учетом времени жизни всех ВМ внутри этого проекта. У модуля должна быть возможность суммирования всех данных по проектам, которые расположены в одном домене и предоставлять отчет по всем доменам облачной платформы.
- на основе собранной информации (использование vCPU, RAM) по всем виртуальным машинам за отчётный период, в разрезе программных продуктов (VDC и VDI), формируется файл отчёта.

Формирование отчёта происходит на основании утилизации (потребления) ресурсов, по проектам или доменам.

Для лицензии VDC учитываются ресурсы, выделенные в рамках классических проектов OpenStack.

Для лицензий VDI используется информация о ресурсах, выделенные в рамках проектов с типами «VDI.Стандартный» и «VDI.Совместный».

Пользователь имеет возможность скачивания отчёта по выбранному периоду и получения файла отчёта по запросу.

Approvie

Модуль TIONIX.Approvie предназначен для расширения функции RBAC платформы OpenStack. Он позволяет использовать сторонний сервис для обработки разрешений доступа к ресурсам платформы.

Архитектура Approvie предусматривает несколько компонентов, устанавливаемых на контроллеры OpenStack в виде системных служб (systemd):

- tionix-keystone-rbac.service – управление правилами HttpCheck для службы Keystone;
- tionix-cinder-rbac.service – управление правилами HttpCheck для службы Cinder;
- tionix-glance-rbac.service – управления правилами HttpCheck для службы Glance;
- tionix-nova-rbac.service – управление правилами HttpCheck для службы Nova;
- tionix-neutron-rbac.service – управление правилами HttpCheck для службы Neutron.

Dashboard

Модуль TIONIX.Dashboard характеризуется как «приборная панель» интерфейса управления, которая предоставляет стандартный функционал графических инструментов, имеющих доступ к:

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

Доступ к функциям управления объектами инфраструктуры основан на объектно-ролевой модели. При этом используется графический интерфейс пользователя, отображаемый через вкладку веб-браузера.

Для доставки контента может использоваться протокол передачи гипертекста – HTTP, а также его защищенная реализация – HTTPS. Протокол HTTPS поддерживается большинством популярных веб-браузеров.

6.4 Требования к облачной платформе Базис.Cloud

6.4.1 Требования к построению кластера

Платформа ВСР в продуктивной среде должна работать с адекватным уровнем отказоустойчивости - параметре, который определяет корректность работы ПО при отказе части компонентов и при их восстановлении. Для повышения уровня отказоустойчивости используются различные средства кластеризации - объединения экземпляров сервисов, запущенных на разных узлах в один метасервис с единой точкой входа. Этот раздел кратко объяснит требования к кластеризируемым сервисам и к инфраструктуре.

Управляющие узлы

Различные сервисы платформы используют свои механизмы кластеризации.

Pacemaker

Pacemaker является основным средством запуска сервисов в режиме отказоустойчивости. В качестве средства синхронизации состояния сервисов используется Corosync.

Для корректной работы Pacemaker требуется:

- Минимальное количество экземпляров в кластере: 3, рекомендуется нечетное количество.

- Необходима настройка протокола STONITH с использованием протокола IPMI (при технической возможности) и статуса сервисов.
- Управление статусом всех сервисов платформы должно производиться через механизмы Pacemaker.
- Рекомендуется дублировать данные Corosync через сеть репликации.

MariaDB

MariaDB содержит встроенную систему кластеризации в режиме Active/Active, которая называется Galera. Galera использует протокол wprep и rsync для репликации данных между узлами. В Pacemaker сервисы MariaDB добавляются в режиме Active/Active.

Для корректной работы Galera требуется:

- MariaDB Galera является сервисом с хранением данных и алгоритмом репликации на основе Raft, поэтому при построении кластера нужно учесть правила кворума.
- Минимальное количество экземпляров в кластере: 3, рекомендуется нечетное количество.
- Для хранения данных БД требуется быстрый локальный носитель на базе твердотельного диска.
- Рекомендуется дублировать данные через сеть репликации.

RabbitMQ

RabbitMQ имеет встроенные средства кластеризации с репликацией данных очередей сообщений. В Pacemaker сервисы MariaDB добавляются в режиме Active/Backup.

Для корректной работы кластера RabbitMQ требуется:

- Минимальное количество экземпляров в кластере: 3, рекомендуется нечётное количество.
- Для полноценной репликации необходимо включить функцию персистентных очередей сообщений (durable queue), которые должны сохраняться на дисках узлов управления и реплицироваться между собой.
- Для хранения данных БД желательно использовать быстрый локальный носитель на базе твердотельного диска.
- Рекомендуется дублировать данные через сеть репликации.

memcached

memcached не имеет встроенных средств кластеризации, добавляется в Pacemaker в режиме Active/Backup без синхронизации данных кэша между инстансами memcached. Принято, что данные кэша не являются важными и их можно терять.

Сервисы OpenStack

- Большинство сервисов OpenStack не хранят свое состояние (речь про состояние самого сервиса, а не про данные облачной платформы, хранимые в БД). Поэтому они должны запускаться в Pacemaker в режиме Active/Active.
- Минимальное количество экземпляров сервисов: 2.
- Конфигурация сервисов OpenStack между узлами кластера должна быть эквивалентной.
- В точках доступа сервисов OpenStack обязательно нужно использовать DNS-имя контроллера облака с резолвингом на виртуальный IP-адрес или динамическое изменение адреса DNS управляющего узла.
- Сервисы cinder-volume и nova-novncproxy должны запускаться в единственном экземпляре, поэтому они должны быть добавлены в Pacemaker в режиме Active/Backup из-за особенностей работы в кластерном окружении.

Модули TIONIX

- Большинство сервисов TIONIX не хранят свое состояние (речь про состояние самого сервиса, а не про данные облачной платформы, хранимые в БД). Поэтому они должны запускаться в Pacemaker в режиме Active/Active.
- Минимальное количество экземпляров сервисов: 2
- Конфигурация сервисов Tionix между узлами кластера должна быть эквивалентной.
- Сервис tionix-node-control-node-sync должен запускаться в единственном экземпляре, поэтому он должен быть добавлен в Pacemaker в режиме Active/Backup из-за особенностей работы в кластерном окружении.

Вычислительные узлы

Сервисы вычислительных узлов должны работать вне кластера управления. Сервис nova-compute должен быть настроен на единый виртуальный адрес кластера.

6.4.2 Требования к вычислительным ресурсам

Серверный комплекс платформы состоит из двух видов узлов в соответствии с их функциональным назначения:

- Управляющие узлы или контроллеры (далее – УУ). Используются для обеспечения вычислительными ресурсами СУ ОП ПВ;
- Вычислительные узлы (далее – ВУ). Используются для предоставления виртуализованных вычислительных ресурсов, в виде экземпляров виртуальных машин, прикладным информационным системам.

Управляющие узлы

Минимальная, рекомендованная конфигурации узлов, определяющая выделение ресурсов для кластера управления, представлены в данной таблице:

Минимальная конфигурация	**Рекомендуемая базовая**
3 физических узла	
2 x CPU sockets (6 CPU cores (x86_64))	2 x CPU sockets (10 CPU cores (x86_64))
64 GB RAM	128 GB RAM
2 x 300GB SSD (DWPD >= 3) RAID1	2 x 300GB SSD (DWPD >= 3) RAID1
2 x 1GbE, 2 x 10GbE ports	2 x 10 GbE, 2 x 10 GbE

Вычислительные узлы

Для определения мощности вычислительного узла (ВУ), подключаемого к вычислительному кластеру (Compute), используются исходные требования к количеству ВМ на один ВУ.
Например, для 20-ти виртуальных машин, размещаемых на одном ВУ, следует использовать наиболее употребляемый шаблон, требующий 2 виртуальных CPU (vCPU) и 4 Гбайт оперативной памяти (RAM).

Линейная калькуляция потребности выделения физических ресурсов:

- 20 x 2 = 40 [vCPU];
- 20 x 4096 = 81920 = 80 GB [RAM].

Коэффициент запаса, предполагающий 20%, выбирается равным 1,2 для обоих параметров вычислительной мощности. Соответственно, суммарный сайзинг, из расчета на один ВУ, составит: 48 vCPU и 96 GB vRAM.

С учетом оверкоммита – высоконагруженного состояния – необходимо поделить количество vCPU на показатель переподписки, например 8 (одно физическое ядро процессора : восемь vCPU). Будет получено количество реальных ядер CPU, необходимое для покрытия среднестатистических потребностей:

48/8 = 6 CPU cores (6 физических ядер процессора)

Исходя из приведенного расчета, достаточно выбрать серверную систему с одним физическим процессором (CPU), содержащим 6 или более (физических) ядер и 96 ГБ RAM.

> ⚠ Современные серверные системы позволяют размещать на системной плате от двух и более физических процессоров, каждый из которых содержит 8-16 процессорных ядер.

Не рекомендуется использовать переподписку для cpu более чем 8.

Использование переподписки по памяти не рекомендуется (1 ГБ RAM : 1ГБ vRAM).

6.4.3 Требование к сети (сегментация)

В архитектуре облачной платформы необходимо использовать несколько физических сегментов сети, которые должны обрабатывать трафик с различным функциональным назначением. Принят следующий список сетей:

- **Сеть управления** (management network). Предназначен для трафика с содержанием команд управления облака и контроля над вычислительными ресурсами.

    - Минимальная скорость интерфейсов сети управления: 1Gbit/s.
    - Интерфейсы сети управления должны иметься на всех узлах облачной платформы.
    - Jumboframe не обязателен.
    - Можно использовать для цели репликации данных кластера управляющих компонентов.
- **Сеть вычислений** (compute network). Используется для обмена трафиком между виртуальными сетями облачной платформы (иными словами, между виртуальными машинами).
    - Минимальная скорость интерфейсов сети вычислений: 10Gbis/s.
    - Интерфейсы сети вычислений должны иметься на всех вычислительных узлах.
    - Необходимо включение Jumbo-кадров, равным 9000 байтов.
- **Сеть хранения** (storage network). Используется для получения доступа к данным ВМ, которые хранятся во внешних системах хранения.
    - Может использовать как Ethernet, так и FibreChannel в качестве протоколов канального уровня.
    - Минимальная скорость интерфейсов сети хранения: 8Gbit/s (FC), 10Gbit/s (Ethernet).
    - Интерфейсы сети хранения должны иметься на всех вычислительных узлах и в узлах хранения, если они являются обычными узлами на ОС на базе ядра Linux.
    - Управление внешней системой хранения должно осуществляться через сеть управления.
    - Необходимо включение Jumbo-кадров, равным 9000 байтов.
- **Сеть VDI** (VDI network). При наличии VDI-функций используется для доступа до VDI-сессий (пока только SPICE).
    - Минимальная скорость интерфейсов сети хранения: 10Gbit/s.
    - Интерфейсы сети VDI должны иметься на вычислительных узлах, которые добавлены в проекты VDI.
    - В этой сети можно включить Jumbo-кадры, равные 9000 байтов, однако необходимо иметь в виду, что включение Jumbo негативно влияет на задержки в работе VDI-протоколов.

Специализированная сеть передачи данных FibreChannel является рекомендуемой, в связи с отсутствием ethernet задержек и специализированности данных сетей для передачи данных.

Тем не менее вполне допустимо использование Ethernet сетей для взаимодействия с хранилищами, при обеспечении достаточной отказоустойчивости и производительности.

6.4.4 Требование к дисковому пространству

В облачной платформе могут быть использованы два вида хранения:

- Блочное устройство Cinder.
- Эфемерные диски - виртуальные диски, расположенные в файловой системе ВУ.

Требования к блочным устройствам Cinder

Использование блочных устройств Cinder является рекомендуемым вариантом для хранения пользовательских данных. Требования к СХД, подключаемых к сервисам Cinder, могут отличаться в зависимости от используемого драйвера.

Общие требования к СХД с Ethernet и FC

- Количество экземпляров СХД должно быть не менее двух. Потеря одного экземпляра СХД не должна приводить к потере доступа к данным.
- СХД должны использовать сеть хранения для доступа к пользовательским данным.
- СХД должны быть доступны по множественным путям сети с использованием протокола Multipath.
- При возможности следует использовать TIONIX Driver, в противном случае необходимо использовать драйвер, предоставленный вендором СХД.
- Необходимо использовать только образы формата RAW для запуска виртуальных машин. При использовании остальных форматов следует предоставить УУ дисковое пространство для конвертации образа в формат RAW.

Общие требования к эфемерным дискам

- Эфемерные диски доступны только при наличии дисков в ВУ. При сетевой загрузке ВУ эфемерные диски недоступны.
- Эфемерные диски не предназначены для долговременного хранения пользовательских данных.
- Необходимо использовать формат QCOW2 или другой формат, поддерживающий дельта-файлы.

6.4.5 Требования к хостовым ОС

Системные пакеты

Дистрибутив	Версия QEMU	Поддержка KVM	Версия libvirt	Версия OVS	Версия RabbitMQ	Версия memcached	Версия MariaDB
Almalinux 8.4	5.0	да	6.0.0	2.12	3.8.0	1.6.0	10.3

Модули TIONIX

Модуль	Версия
NodeControl	⩾3.0
Dashboard	⩾3.0
Monitor	⩾3.0
Scheduler	⩾3.0
Client	⩾3.0
Drivers	⩾3.0
Agent	⩾3.0
PointMeter	⩾3.0

6.5 Перечень поддерживаемого оборудования

Поставляемое решение	Номенклатура оборудования
Базис.Cloud	MB Supermicro MBD-X11SPH-NCTPF-O/Intel Xeon-SC 6138; Сервер Intel® Server BoardS2600WFT/1*Intel Xeon-SC 6138; Сервер Intel® Server BoardS2600WFT/2*Intel Xeon-SC 6138; MB Supermicro MBD-X10SDV-12C-TLN4F+/Intel® Xeon® processor D-1557; MB Supermicro MBD-X10SDV-8C-TLN4F+/Intel® Xeon® processor D-1537; MB Supermicro MBD-X10SDV-TLN4F/X10SDV-8C-TLN4F/Intel® Xeon® processor D-1541; MB Supermicro MBD-X10SDV-TP8F/Intel® Xeon® processor D-1518; MB Supermicro MBD-X10DRI-T/Intel® Xeon® Processor E5-2650v4; MB Supermicro MBD-X10DRU-i+/Intel® Xeon® Processor E5-2650v4; Сервер HUAWEI 2488 V5 /4*Intel(R) Xeon(R) Gold 6148; СХД HP StorageWorks P2000 G3
	Сервер HUAWEI 2488 V5 /4*Intel(R) Xeon(R) Gold 6148; СХД HUAWEI OceanStor 2600V3(2U, Dual Ctrl, AC, 64GB, 2*6*GE,25*2.5; SPE23C0225) с полкой OceanStor 2600 V3 Disk Enclosure
	Сервер Huawei RH1288 V3 2 xIntel(R) Xeon(R) CPU E5-2650 v4 @2.20GHz/ 256 GB/ 2x 600GB HDD/2x 480GB SSD
	MB Supermicro MBD-X11SPH-NCTPF-O/Intel Xeon-SC 6138; Сервер Intel® Server BoardS2600WFT/1*Intel Xeon-SC 6138

6.6 Общие указания

Для реализации функций безопасности среды функционирования ПО «Базис.Cloud» должны выполняться следующие действия:

- необходимо регулярное обновление всех сред функционирования ПО «Базис.Cloud» до актуальных версий с применением всех необходимых патчей безопасности с официальных сайтов разработчиков сред функционирования;
- компоненты операционной системы и сред функционирования ПО «Базис.Cloud» должны быть максимально ограничены. Компоненты, которые не участвуют в функционировании ПО «Базис.Cloud», должны быть отключены;
- должно обеспечиваться предотвращение несанкционированного доступа к идентификаторам и паролям администраторов среды виртуализации, которые необходимы для управления и технической поддержки среды функционирования ПО «Базис.Cloud»;
- необходимо использовать на серверах, где развернута среда функционирования ПО «Базис.Cloud», в качестве средств защиты информации от несанкционированного доступа, сертифицированных ФСТЭК России версий операционных систем с установленными обновлениями или наложенных средств защиты информации, прошедших сертификацию по требованиям безопасности информации в системе сертификации средств защиты информации по требованиям безопасности информации № РОСС RU.0001.01БИ00;
- должна быть обеспечена физическая сохранность серверной платформы с установленным ПО «Базис.Cloud» и исключение возможности физического доступа к ней посторонних лиц;
- каналы передачи данных ПО «Базис.Cloud» должны быть либо расположены в пределах контролируемой зоны и защищены с использованием организационно-технических мер, либо, в случае их выхода за пределы контролируемой зоны, должны быть защищены путем применения средств криптографической защиты информации, сертифицированных в системе сертификации ФСБ России.

6.7 Действия по безопасной установке и настройке

6.7.1 Операционная среда

Администратор ПО должен организовать свое рабочее место таким образом, чтобы можно было устанавливать безопасное подключение к любому узлу облачной инфраструктуры.

Схема сегментации ЛВС, обеспечивающей коммутацию физических и логических объектов облачной инфраструктуры, соответствует выбранной референсной архитектуре, предусматривающей деление средств ВТ на управляющие и вычислительные узлы.

Сеть (аппаратного) управления, подаваемая со стороны ЦОД, использует интерфейс IPMI, гарантирующий высокую доступность средств ВТ серверного типа и СХД, размещенных на одной или нескольких площадках. Дополнительно, каждый из управляющих и вычислительных узлов (хост-систем) оснащен двумя сетевыми интерфейсами с высокой пропускной способностью, используемыми аггрегированным каналом (LACP).

На стороне продуктивной сети, организованной при помощи стека Ethernet коммутаторов, используется разделение информационных потоков с использованием VLAN. Информационные потоки делятся на:

- сеть/сети тенантов;
- внутренняя сеть (Internal API);
- публичная сеть (Public API).

Системное окружение

Чтобы взаимодействовать со службами OpenStack, функционирующими на контроллере (УУ), необходимо настроить окружение. Это касается не только клиента OpenStack, но и утилит, позволяющих облачному администратору взаимодействовать с инфраструктурными ресурсами посредством служб OpenStack.

Состав служб, запускаемых на управляющих узлах, может незначительно отличаться и обусловлен требованиями ПО, используемого для кластеризации облачных ресурсов.

Для настройки окружения, после успешного подключения к УУ необходимо выполнить одну из команд:

```
. /root/admin-openrc.sh
или
source /root/admin-openrc.sh
```

Включение требуемой системной службы ОС, обеспечивающей автозагрузку определенного облачного сервиса (службы OpenStack и др.) на этапе инициализации операционной системы, производится выполнением команды:

```
systemctl enable <системная_служба>
или
systemctl enable <список_системных_служб>
```

Удаленное подключение к контроллеру

Для удаленного подключения к контроллеру OpenStack (УУ) необходимы реквизиты доступа – имя (системного) пользователя и пароль. Если развертывание облака производилось автоматизированно, то имя системного пользователя **tionix** уже создано (на всех узлах инфраструктуры).

Подключение из системной оболочки Linux производится с помощью утилиты ssh, которая поставляется в составе пакета OpenSSH. В строке приглашения к вводу (может выглядеть по-разному) выполните команду:

```
ssh tionix@<доменное_имя>
```

Подключение к серверу OpenSSH из Windows осуществляется с помощью PowerShell . В строке приглашения к вводу (`PS C:\Users\cloud_admin>`) выполните команду:

```
ssh.exe tionix@<доменное_имя>
```

Если не настроена ключевая пара, используемая для осуществления безопасного подключения к контроллеру о обмена по SSH, то будет выведено сообщение вида:

```
tionix@<доменное_имя>: Permission denied (publickey,gssapi-keyex,gssapi-with-mic)
```

Копирование (создание) открытого ключа

Копирование открытого ключа (файла `id_rsa.pub`) рекомендуется выполнить с использованием утилиты `ssh-copy-id`, вызываемой в следующем формате:

```
ssh-copy-id -i $HOME/.ssh/id_rsa.pub user@<serverN>.<domain_name>


где

<domain_name> – доменное имя или IP-адрес;

<serverN> – название сервера, помещенного в облачный домен и распознаваемое через DNS
```

Способ генерации ключа в Windows и Linux практические не отличается. Используется утилита `ssh_keygen`.

Linux:

```
$ ssh_keygen
```

Windows PowerShell:

```
$ ssh_keygen.exe
```

Подключение к облаку

OpenStack поддерживает использование `clouds.yaml` – файла конфигурации, содержащего параметры настройки подключения к облаку. Работу с несколькими облаками можно упростить, сохранив информацию о конфигурации этих облаков в локальном файле.

OpenStack Client использует плагины аутентификации Keystone, поэтому необходимые параметры аутентификации не всегда известны, пока не выбран метод. OpenStack попытается обнаружить пару общих типов аутентификации – на основе аргументов, переданных или найденных в файле конфигурации. Но, если они не являются полными, то будет невозможно узнать, какой тип аутентификации назначен.

Конфигурационные файлы

OpenStack для настройки конфигурации выполняет поиск файла с именем `clouds.yaml`, в следующем порядке:

- просмотр текущей директории (`./`);
- просмотр директории `~/.config/openstack` (локальная конфигурация);
- просмотр файла `/etc/openstack` (глобальная конфигурация).

Первый найденный файл используется для настройки (поиск прекращается).

Переменные среды

Переменные среды (окружения) могут быть установлены для изменения поведения **openstack**. Большинство из них имеют соответствующие параметры командной строки, которые имеют *приоритет* (если установлены).

`OS_CLOUD` – имя конфигурации облака в `clouds.yaml`.

`OS_AUTH_PLUGIN` – плагин аутентификации, используемый при подключении к службе Identity, его версия должна соответствовать версии Identity API.

`OS_AUTH_URL` – URL аутентификации.

`OS_URL` – сервисный URL (при использовании сервисного токена).

`OS_DOMAIN_NAME` – область авторизации на уровне домена (имя или идентификатор).

`OS_PROJECT_NAME` – область проверки подлинности на уровне проекта (имя или идентификатор).

`OS_PROJECT_DOMAIN_NAME` – доменное имя или идентификатор, содержащий проект.

`OS_USERNAME` – аутентификация имени пользователя.

`OS_TOKEN` – аутентифицированный или сервисный токен.

`OS_PASSWORD` – пароль аутентификации.

`OS_USER_DOMAIN_NAME` – доменное имя или ID, содержащий пользователя.

`OS_TRUST_ID` – идентификатор траста для использования в качестве доверенного пользователя.

`OS_DEFAULT_DOMAIN` – идентификатор домена по умолчанию (по умолчанию: «по умолчанию»).

`OS_REGION_NAME` – название региона аутентификации.

`OS_CACERT` – файл пакета сертификатов CA.

`OS_CERT` – файл пакета сертификата клиента.

`OS_KEY` – файл ключа сертификата клиента.

`OS_IDENTITY_API_VERSION` – версия Identity API (по умолчанию: 2.0).

`OS_XXXX_API_VERSION` – дополнительные параметры версии API будут доступны в зависимости от установленных библиотек API.

`OS_INTERFACE` – тип интерфейса (допустимые значения: public, admin и internal).

При переключении на openstackclient из клиентов (указанных в проекте), таких как: nova, neutron и т.д., использовать `OS_INTERFACE` вместо `OS_ENDPOINT_TYPE`.

Методы аутентификации

OpenStack использует *схему аутентификации*, аналогичную CLI проекта OpenStack, с информацией о полномочиях, предоставляемой либо в качестве переменных среды, либо в качестве параметров в командной строке.

Основным отличием является использование «проекта» в названии опций OS_PROJECT_NAME/ OS_PROJECT_ID над старыми именами арендаторов.

```
export OS_AUTH_URL=<url-to-openstack-identity>
export OS_PROJECT_NAME=<project-name>
export OS_USERNAME=<user-name>
export OS_PASSWORD=<password> # (optional)
```

Могут использоваться различные типы подключаемых модулей аутентификации, предоставляемых библиотекой `keystoneclient`. Доступны следующие плагины (по умолчанию):

- token: плагин аутентификации с помощью токена;
- password: плагин аутентификации с использованием имени пользователя и пароля.

Более подробная информация об этих плагинах и их опциях, а также для получения полного списка доступных плагинов приведена в описании библиотеки `keystoneclient`.

Необходимо иметь в виду, что некоторые плагины могут не поддерживать все функции openstack. Например, плагин `v3unscopedsaml` может доставлять только токены с незаданной областью, некоторые команды могут быть недоступны при использовании этого метода аутентификации.

Для использования метода *v3unscopedsaml* необходимо установить пакет `lxml`.

Кроме того, для аутентификации можно использовать служебный токен Keystone, задав параметры –os-token и –os-url (или переменные среды). OS_TOKEN а также OS_URL соответственно). Этот метод имеет приоритет над плагинами аутентификации.

6.8 Включение/выключение оборудования платформы

В разделе изложен порядок запуска в эксплуатацию всех объектов облачной инфраструктуры, включенных в отказоустойчивый кластер.

Исходное состояние – полностью отключенное оборудование на площадке, регламентируемое понятием «технологического перерыва».

Далее приведено описание корректного включения оборудования и запуск сервисов платформы виртуализации после проведения сервисных (профилактических) работ, связанных с ремонтом/заменой оборудования или временного выхода инфраструктуры из строя (пропадание электропитания и т.п.).

После включения оборудования следует произвести экспресс-проверку состояния служб (Самодиагностика платформы).

Включение оборудования инфраструктуры (аппаратных средств) следует выполнять в соответствии с инструкциями по эксплуатации от производителя оборудования.

6.8.1 Включение оборудования (запуск инфраструктурных компонентов)

Процедуру включения оборудования производить в следующем порядке:

- в первую очередь производится включение вычислительных узлов (ВУ);
- в последнюю очередь производится включение устройств управления (УУ).

6.8.2 Проверка состояния технологических сервисов

Для проверки состояния технологических сервисов подключиться к консоли контроллера под учетной записью root и проверить состояние ресурсов отказоустойчивого кластера (PCS).

- Выполнить команду:

```
pcs status
```

- Проверить состояние кластера Galera, выполнив команду:

```
mysql -p -e "SHOW STATUS LIKE'wsrep_cluster_size';"
```

- Проверить состояние кластера очередей сообщений RabbitMQ, выполнив команду:

```
rabbitmqctl cluster_status
```

- Убедиться, что все узлы кластера запущены.
- На всем оборудовании платформы виртуализации проверить наличие синхронизации системного времени, выполнив команду:

```
chronyc sources
```

6.8.3 Проверка состояния служб системы виртуализации

- Из консоли контроллера загрузить профиль переменных пользователя **admin,** выполнив команду:

```
# . admin-openrc.sh
```

- Произвести контроль состояния сервисов службы вычислений Nova, выполнив команду:

```
# openstack compute service list
```

- Произвести контроль состояния агентов сетевой службы Neutron:

```
# openstack network agent list
```

- Произвести проверку состояния службы блочных устройств Cinder, выполнив команду:

```
# openstack volume service list
```

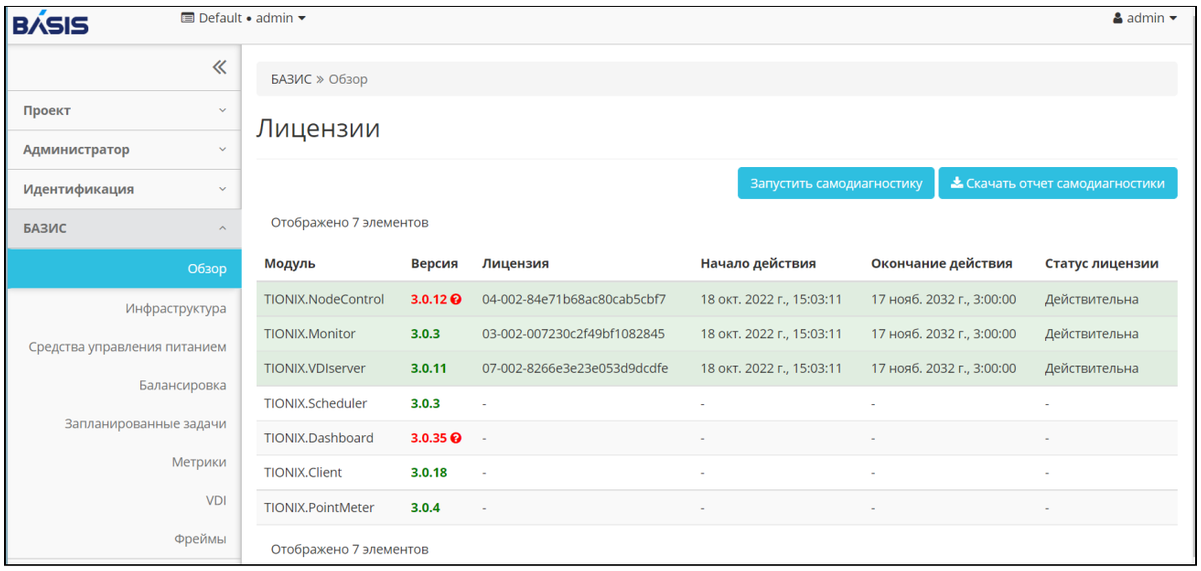
6.8.4 Самодиагностика платформы

Выполнить вход в графический интерфейс управления платформой виртуализации с правами администратора.

Перейти в раздел:

> *БАЗИС >> Обзор*

Запустить процесс самодиагностики.



Скачать отчет самодиагностики и выбрать один из вариантов:

- сохранение отчета в виде файла;
- загрузка отчета в текстовый редактор.

Проанализировать отчет о самодиагностике платформы (модулей TIONIX).

6.8.5 Выключение ОП (с корректным завершением работы служб)

Отключение виртуальных машин, потребляющих вычислительные ресурсы ОП, выполняется в первую очередь. Рекомендуется дождаться статуса завершения работы для каждой из ВМ.

Для отключения виртуальных машин может быть использован механизм *планирования задач* – служба планировщика (модуль TIONIX.Scheduler), обеспечивающая выполнение административных действий в заданное время. Планирование отключения ВМ позволит равномерно распределить нагрузку на подсистемы виртуализации и хранения, так как планировщик использует очередь.

Завершение работы виртуальных машин

Выполнить вход в интерфейс управления с правами администратора (роль - admin).

Перейти в раздел:

> *Проект >> Администратор >> Вычисления >> Виртуальные машины*

Отметить (галочкой) чек-бокс «Все ВМ».

Из контекстного меню «Еще действия» выбрать – «Выключить машину».

Завершение работы средств вычислительной техники

По завершении процесса отключения виртуальных машин необходимо последовательно выполнить удаленное подключение ко всем узлам платформы (сначала – ВУ, затем – УУ) и, команду отключения **poweroff** (с полномочиями суперпользователя).

Процедуру выключения СВТ необходимо производить в следующем порядке:

- первыми выключать ВУ, один за другим (согласно пула IP-адресов);
- в последнюю очередь произвести выключение УУ.

Для того, чтобы сократить количество операций ввода команд, рекомендуется выполнять безопасное подключение с указанием команды выключения, которая должна быть выполнена на удаленном узле:

```
ssh tionix@copmute1 -c "poweroff"

ssh tionix@copmute2 -c "poweroff"


и т.д.
```

6.9 Проверка работоспособности ПО

Так как инфраструктура облачной платформы развернутой на базе ПО "Базис.Cloud" сложно организована, *функциональное тестирование* должно заключаться в проверке работоспособности подсистем и компонент, учитывая все особенности архитектурного построения отказоустойчивого управляющего и масштабируемого вычислительного кластеров.

После завершения развертывания на базе ПО "Базис.Cloud" рекомендуется выполнить запуск **тестов самодиагностики** с указанием опции --errors-only. Будут отражены только те тесты, которые завершились с ошибкой.

В случае возникновения неопределенных обстоятельств в работе ПО (модулей, используемых системных библиотек), служба эксплуатации может эффективно взаимодействовать со службой Технической Поддержки, опираясь на логирование – сохранение информационных сообщений в контрольных точках алгоритмов работы модулей.

6.9.1 Штатный режим функционирования

Ниже перечислены критерии оценки состояния ПО, функционирующего в *штатном режиме*:

- Доступны (по сети) все узлы управляющего кластера.

Всего узлов: 3 шт (проверка через внешний мониторинг).

- Проверка «здоровья» Openstack отрабатывает успешно.
  Запросы «health-check» к службам Keystone, Glance выполняются успешно (код возврата HTTP-запроса – 200).
- Работает вход в Dashboard – окно авторизации обсуживает запросы через точку входа (IP-адрес).
- После входа в дашборд отображаются доступные лицензии в статусе "Действительна" (зеленый цвет).
- Доступна сводка лимитов:

> *Проект >> Вычисления >> Обзор*

> *https://<точка_входа>/dashboard/project/*

Мощность вычислительной инфраструктуры обычно измеряется в количестве vCPU, выделенном для использования в рамках проекта (по подписке). Общее количество vCPU обычно пропорционально количеству ВУ, на которых установлена служба гипервизора, а также количеству процессорных ядер на один отдельно взятый физический процессор (CPU).

Кроме того, сводка лимитов отображает доступный к использованию (по подписке) объем оперативной памяти, отображаемый на физическую память (RAM), установленную на материнские платы вычислительных узлов.

- Доступны гипервизоры (ВУ) в определенном количестве, зафиксированном при вводе в эксплуатацию.

> *https://<точка_входа>/dashboard/tionix/infrastructure/*

- Функционируют виртуальные машины.

Переход:

> *Проект >> Вычисления >> Виртуальные машины*

В зоне доступности (гипервизоров) создается одна и более виртуальных машин, после чего они могут находиться в различных управляемых состояниях (Активна, На паузе, Выключено).

Просмотр виртуальных машин также может быть открыт с помощью URL, заданного в формате:

> *https://<точка_входа>/dashboard/project/instances/*
>
> *https://<точка_входа>/dashboard/admin/instances/*

- К виртуальной машине можно подключиться из Dashboard (консоли ВМ), с использованием встроенной VNC.

6.10 Аварийные ситуации

6.10.1 Действия в случаях обнаружении несанкционированного вмешательства в данные

Несанкционированное вмешательство обнаруживается при помощи протокола нарушений безопасности.

В случаях обнаружения несанкционированного вмешательства в данные, необходимо установить логин пользователя, под которым была произведена аутентификация, затем сменить пароль для этого пользователя и проинформировать пользователя о смене пароля.

6.10.2 Действия в других ситуациях

В других аварийных ситуациях необходимо обратиться в сервисную службу:

Электронный адрес: **support@basistech.ru**

7 Описание функционирования

Описание совместного функционирования технических средств и ПО, описание организации входных и выходных данных, используемых при обслуживании технических средств и описание взаимодействий устройств с ПО приведено в эксплуатационных документах:

«ПО «Базис.Cloud». Руководство по установке. RU.НРФЛ.00004-01.96.01;

«ПО «Базис.Cloud». Руководство администратора. RU.НРФЛ.00004-01.95.01.

8 Перечень. Термины и сокращения

Термин	Определение
LDAP	(англ. Lightweight Directory Access Protocol) - протокол прикладного уровня для доступа к службе каталогов
USB	(англ. Universal Serial Bus) — «универсальная последовательная шина», последовательный интерфейс для подключения периферийных устройств к вычислительной технике
ПО	Программное обеспечение
Пул	Логический объект СХД, объединяющий пространства нескольких физических накопителей в единое пространство хранения данных
СХД	Система хранения данных
ЦОД	Центр обработки данных